ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τεχνικές εξαγωγής πληροφορίας από βιολογικά δεδομένα και χρήση Νευρωνικών Δικτύων ως ταξινομητές για ιατρική διάγνωση

Διπλωματική Εργασία

Κωνσταντίνα Κορμπή

Χανιά, 2019

Εξεταστική Επιτροπή :

Καθ. Μιχάλης Ζερβάκης,

Καθ. Γεώργιος Σταυρακάκης,

Δρ. Ελευθερία Σεργάκη

TECHNICAL UNIVERSITY OF CRETE

Techniques for biological data feature aggregation for medical diagnosis and use of Neural Networks as classifiers.

Diploma Thesis

Konstantina Kormpi

Chania, 2019

Thesis Committee:

Prof. Michael Zervakis,

Prof. Georgios Stavrakakis,

Dr. Eleftheria Sergaki

Abstract

Cancer is a global problem as it is described in the World Cancer Report. Today's technology can give approaches that reveal the cellular and molecular level of cancer. In a cancer disease sample such a cell biopsy to be processed, thousands of genes at a time can be subjected simultaneously for analysis in a single chip, called Microarray.

Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to "learn" from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex proteomic and genomic measurements. As a result, machine learning is frequently used in cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. This latter approach is particularly interesting as it is part of a growing trend towards personalized, predictive medicine.

Our goal was, firstly, to construct a framework for statistical analysis, description and visualization of real biological data and secondly, build a predictive model for binary classification of cancer based on machine learning algorithms and feature selection techniques. We use six algorithms of supervised machine learning such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Naïve Bayes (NB) and Linear Support Vector Machines (SVM) to be tested in different datasets of Cervical, Breast, Acute Myeloid Leukemia and Pancreatic cancer, publicly available on Gene Expression Omnibus platform.

During the learning procedure, the data were split to validation and train sets. The train set, is used in 5-fold cross-validation for three different scenarios: on primary data, on standardized data, and finally on standardized data that have been transformed by the dimensionality reduction technique of Principal Component Analysis (PCA) and other feature reduction techniques. Finally we compare the results and use the validation dataset to evaluate our models' predictions on unseen data.

We end up with prediction accuracy: 100% of models trained with LR, NB and SVM on Cervical dataset, 90% of models built with LDA on Breast dataset, 95.4% of models trained with NB on AML dataset and 94.4% trained with LR Pancreatic dataset, respectfully. During the procedure, we compare the results of 5-fold cross-validation on each step and finally we estimate more evaluation metrics such as precision, sensitivity, f1-score and ROC curves, in order to extract useful insights.

Keywords: cancer diagnosis, predictive model, machine learning, microarrays, gene expression, feature selection techniques, dimensionality reduction, Logistic Regression, Linear Discriminant Analysis, k-Nearest Neighbors, Classification And Regression Trees, Naïve Bayes, Support Vector Machines, Principal Component Analysis.

Περίληψη

Η Παγκόσμια Έκθεση για τον Καρκίνο περιγράφει την ασθένεια του καρκίνου σαν ένα παγκόσιο πρόβλημα. Η σημερινή τεχνολογία μπορεί να μας δώσει προσεγγίσεις που αποκαλύπτουν τον καρκίνο σε κυτταρικό και μοριακό επίπεδο. Σε ένα δείγμα καρκινικής νόσου όπως μια βιοψία κυττάρων, χιλιάδες γονίδια κάθε φορά μπορούν να υποβληθούν σε ανάλυση με την τεχνολογία μικροσυστοιχιών. Οι μικροσυστοιχίες βοηθούν στην ταυτόχρονη ανάλυση των προφίλ γονιδιακής έκφρασης ενός μεγάλου αριθμού γονιδίων σε ένα μόνο πείραμα. Η κατανόηση των προτύπων γονιδιακής έκφρασης μπορεί να βοηθήσει στη διάγνωση και διάκριση διαφόρων τύπων καρκίνου. Η μηχανική μάθηση είναι ένας κλάδος της τεχνιτής νοημοσύνης που χρησιμοποιεί μια ποικιλία τεχνικών στατιστικής, πιθανοτήτων και βελτιστοποίησης που επιτρέπουν στους υπολογιστές να "μαθαίνουν" από παλιά παραδείγματα και να ανιχνεύουν μορφές που είναι δύσκολο να διακρίνουν από μεγάλα, θορυβώδη ή σύνθετα σύνολα δεδομένων. Αυτή η ικανότητα είναι ιδιαίτερα κατάλληλη για ιατρικές εφαρμογές, ειδικά εκείνες που εξαρτώνται από σύνθετες πρωτεϊνικές και γονιδιακές μετρήσεις. Ως αποτέλεσμα, η μηχανική μάθηση χρησιμοποιείται συχνά στη διάγνωση και στον εντοπισμό του καρκίνου. Πιο πρόσφατα η μηχανική μάθηση έχει εφαρμοστεί στην πρόγνωση καρκίνου. Αυτή η τελευταία προσέγγιση είναι ιδιαίτερα ενδιαφέρουσα, καθώς αποτελεί μέρος μιας αυξανόμενης τάσης της προγνωστικής ιατρικής.

Καταρχήν, ο στόχος μας ήταν να επεξεργαστουμε πραγματικά βιολογικά δεδομένα κάνοντας μια στατιστική ανάλυση, περιγραφή και οπτικοποίηση και στη συνέχεια να εκπαιδεύσουμε μοντέλο προβλέψεων για δυαδική ταξινόμηση του καρκίνου, βασισμένο σε αλγόριθμους μηχανικής μάθησης και τεχνικές εξαγωγής γνωρισμάτων. Χρησιμοποιούμε έξι αλγόριθμους μηχανικής μάθησης εποπτείας, όπως Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Naïve Bayes (NB) και Linear Support Vector Machines (SVM) σε διαφορετικά δεδομένα γονιδιακής έκφρασης για τον καρκίνο του τραχήλο, του μαστού, της οξείας μυελοειδούς λευχαιμίας και του καρκίνου στο πάγκρεας, όλα διαθέσιμα δημοσίως στην πλατφόρμα GEO.

Κατά τη διάρκεια της διαδικασίας, τα δεδομένα χωρίστηκαν τυχαία σε δεδομένα εκπαίδευσης (train set) των αλγορίθμων και σε δεδομένα για τις τελικές προβλέψεις (validation set). Το train set χρησιμοποείται με τη μέθοδο 5-fold cross-validation για τρία διαφορετικά σενάρια: στα αρχικά δεδομένα, σε δεδομένα που έχουν επεξεργαστεί με την μέθοδο standardization και τελικά σε επεξεργασμένα δεδομένα που έχουν μετασχηματιστεί από τεχνικές εξαγωγής και μείωσης γνωρισμάτων όπως Principal Component Analysis (PCA). Στο τέλος αφού εκπαιδεύσουμε τα μοντέλα, χρησιμοποιούμε το validation set για αξιολογήσουμε την απόδοση των μοντέλων μας στις προβλέψεις.

Καταλήγουμε να έχουμε ποσοστά ακρίβειας (accuracy) : 100% στα μοντέλα που εκπαιδευτηκαν με LR, NB και SVM στα δεδομένα γονιδιακής έκφρασης του καρκίνου του τραχήλου της μήτρας, 90% στα μοντέλα που εκπαιδευτηκαν με LDA στα δεδομένα γονιδιακής έκφρασης του καρκίνου του μαστού, 95,4% στα μοντέλα που εκπαιδευτηκαν με NB στα δεδομένα γονιδιακής έκφρασης της οξείας μυελοειδούς λευχαιμίας και 94,4% στα μοντέλα που εκπαιδευτηκαν με LR στα δεδομένα γονιδιακής έκφρασης του καρκίνου στο πάγκρεας. Επιπλέον, κατά τη διάρκεια της διαδικασίας εξετάζουμε τα μοντέλα μας για να αξιολογήσουμε περαιτέρω μετρήσεις ταξινόμησης όπως η ακρίβεια (precision), η ευαισθησία (sensitivity) , η βαθμολογία f1 (f1-score) και οι καμπύλες ROC. Τέλος, συγκρίνουμε τα αποτελέσματα του 5-fold cross-validation σε κάθε σενάριο, προκειμένου να εξάγουμε χρήσιμες γνώσεις.

Λέξεις κλειδιά: διάγνωση καρκίνου, μοντέλο προβλέψεων, μηχανική μάθηση, μικροσυστοιχίες, γονιδιακη έκφραση, τεχνικές εξαγωγής γνωρισμάτων και μείωσης διαστάσεων, Logistic Regression, Linear Discriminant Analysis, k-Nearest Neighbors, Classification And Regression Trees, Naïve Bayes, Support Vector Machines.

ACKNOWLEDGEMENTS

First of all, from Technical University of Crete, I would like to thank, my thesis co-supervisor, Dr. Eleftheria Sergaki for the opportunity to work on this subject, the guidance and the knowledge she offered me throughout this work and beyond that.

Additionally, I need to express my gratitude to my teacher Prof. Michael Zervakis for my thesis supervision and to Prof. Georgios Stavrakakis for his membership to my thesis committee.

Furthermore, I would like to thank my old and new friends for the inspiration, motivation and moral support.

Last but more importantly, I would like to thank my parents, Dimitrios and Jenny, and my sister Poppy for their support in every way through these past years. Nothing would be accomplished without them.

Contents

List of Figures

1. INTRODUCTION

1.1 Problem

The number of patients diagnosed with cancer is increasing rapidly. (World cancer report, 2014.) The World Cancer Report described cancer as a global problem and projected an increase to 20 million new cases by 2025. Currently, cancer diagnosis is practiced by using image processing techniques, blood analysis and biopsies. Cancer is caused by the accumulation of excessive amount of damaged cells. (A. TAşçi, 2017) There are approaches in technology that reveals the cellular and molecular level of cancer. In a cancer disease sample such a cell biopsy to be processed, thousands of genes at a time can be subjected for analysis in a single chip called microarray. Microarrays are microscopic slides that contain ordered series of samples of DNA (Deoxyribonucleic acids), RNA (Ribonucleic acids), protein, or tissue and others. (Wong) Gene expression provides the information of how active a gene is. Microarray is one of the widely used measurement methods for gene expression. Gene expression values obtained by microarrays can be employed in cancer diagnosis and the classification of cancer types. (Venugopal Mikkilineni, 2004)

Microarray chip helps the simultaneous analysis of gene expression profiles of a large number of genes in a single experiment. Understanding gene expression pattern can help to diagnose and distinguish different type of cancer. (Gregory Piatetsky-Shapiro, 2003) Generally, Microarray datasets have high number of features (ranges from 2000 to 30000) compared to the samples size (mostly less than 150) and this is called “curse of dimensionality”. (Anil Jain, 1997) So, microarray analysis brings an exciting field of study for Machine Learning researchers. In addition to this, noise and variability of the data make this domain more exciting. (Saeys Yvan, 2007) (C.ArunKumar, 2017)

Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to “learn” from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex proteomic and genomic measurements. As a result, machine learning is frequently used in cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. This latter approach is particularly interesting as it is part of a growing trend towards personalized, predictive medicine.

According to the latest PubMed statistics, more than 1500 papers have been published on the subject of machine learning and cancer. However, the vast majority of these papers are concerned with using machine learning methods to identify, classify, detect, or distinguish tumors and other malignancies. In other words machine learning has been used primarily as an aid to cancer diagnosis and detection (McCarthy et al. 2004). It has only been relatively recently that cancer researchers have attempted to

apply machine learning towards cancer prediction and prognosis. As a consequence the body of literature in the field of machine learning and cancer prediction/prognosis is relatively small.

The fundamental goals of cancer prediction and prognosis are distinct from the goals of cancer detection and diagnosis. In cancer prediction/prognosis one is concerned with three predictive foci: 1) the prediction of cancer susceptibility (i.e. risk assessment); 2) the prediction of cancer recurrence and 3) the prediction of cancer survivability. In the first case, one is trying to predict the likelihood of developing a type of cancer prior to the occurrence of the disease. In the second case one is trying to predict the likelihood of redeveloping cancer after to the apparent resolution of the disease. In the third case one is trying to predict an outcome (life expectancy, survivability, progression, tumor-drug sensitivity) after the diagnosis of the disease. In the latter two situations the success of the prognostic prediction is obviously dependent, in part, on the success or quality of the diagnosis. However a disease prognosis can only come after a medical diagnosis and a prognostic prediction must take into account more than just a simple diagnosis (Hagerty et al. 2005). (Joseph A. Cruz)

The challenge of cancer classification using microarrays is the application of model based selection and prediction algorithm that will classify the cancer genes using gene expression data. The computation time, classification accuracy, and its biological relevance in the cancer classification is still in question. (Wong) Machine learning based prediction of clinical outcomes can be used for appropriate decision making and can lead to better patient care. ML is also a great advantage over traditional statistical models including high power and accuracy to predict disease. To our knowledge, there is no specific algorithm that performs better for the prediction model. (Md. Mohaimenul Islam)

1.2 Goals of thesis

The main goal of this thesis is to build an efficient, robust and accurate predictive model for binary classification of cancer and healthy gene expression data. With this aim, this study explores and examines different machine learning techniques on different cancer datasets and retrieves useful insights.

This work uses four binary microarray datasets of Cervical, Breast, Acute Myeloid Leukemia and Pancreatic Cancer which were retrieved by the Gene Expression Omnibus platform (Gene Expression Omnibus: NCBI gene expression and hybridization array data repository). These datasets are high dimensional with different number of samples each, and include cases with balanced and imbalanced classes worthy of further examination in order to derive useful information.

A statistical analysis of the four datasets and their features is necessary in order to determine the process and the characteristics of gene expression levels' information. The machine learning procedure, which was followed for building a predictive model, includes the evaluation of six different well-known classification algorithms of supervised learning. Logistic Regression, Linear Discriminant Analysis, k-Nearest Neighbors, Classification and Regression Trees, Naïve Bayes and Linear Support Vector Machines shape the algorithm group which this thesis use for further analysis.

On the other hand on the, we aim not only to build a model but to contribute also by analyzing different scenarios with these algorithms. For these purposes and knowing the small number of the samples, cross-validation is chosen to evaluate the performance of the models during the procedure. Three different scenarios are going to be examined in order to derive insights.

Firstly, we intent to check the behavior of the six algorithms in the primary data. Secondly, as data transformation is a necessary step on machine learning, we will reevaluate them in order to compare the results. Finally, we intent to use feature selection techniques, like Principal Component Analysis for feature dimensionality reduction, and reevaluate the models.

Concluding, in this study we aim to build a predictive machine learning model trained on real biological data and also contribute by submitting the results and the insights respectfully during to the whole procedure.

2. THEORETICAL BACKGROUND

2.1 BIOLOGY

2.1.1 Cells

All living things are made of cells: small, membrane-enclosed units filled with a concentrated aqueous solution of chemicals and endowed with the extraordinary ability to create copies of themselves by growing and dividing in two. The simplest forms of life are solitary cells. Higher organisms, including ourselves, are communities of cells derived by growth and division from a single founder cell: each animal, plant, or fungus is a vast colony of individual cells that perform specialized functions coordinated by intricate systems of communication.

There are a multitude of specific chemical transformations that not only provide the energy needed by a cell, but also coordinate all of the events and activities within that cell. The life process involves a wide array of molecules ranging from water to small organic compounds (e.g., fatty acids and sugars), and macromolecules (DNA, proteins, and polysaccharides) that define the structure of the cells. Macromolecules control and govern most of the activities of life.

Deoxyribonucleic acid (DNA) molecules store information about the structure of macromolecules, allowing them to be made precisely according to cells' specifications and needs. DNA is a very stable molecule that forms the "blueprint" of an organism. The DNA structure encodes information as a sequence of chemically linked molecules that can be read by the cellular machinery and guides the construction of the linear arrangements of protein building blocks, which eventually fold to form functional proteins. Molecular biology deals with how information is stored and converted to all the components and interactions that make up a living organism. (LEE) (Alberts B. H.)

2.1.1.1 Procaryotic and Eucaryotic cells

Of all the types of cells revealed by the microscope, bacteria have the simplest structure and come closest to showing us life stripped down to its essentials. Indeed, a bacterium contains essentially no organelles—not even a nucleus to hold its DNA. This property—the presence or absence of a nucleus—is used as the basis for a simple but fundamental classification of all living things. Organisms whose cells have a nucleus are called **eucaryotes**. Organisms whose cells do not have a nucleus are called **procaryotes**. The terms "bacterium" and "procaryote" are often used interchangeably, although we shall see that the category of procaryotes also includes another class of cells, the archaea (singular archaeon), which are so remotely related to bacteria that they are given a separate name. (Alberts B. H.)

2.1.1.1.1 Procaryotes

Procaryotes are typically spherical, rod like, or corkscrew-shaped, and small- just a few micrometers long, although there are some giant species as much as 100 times longer than this. They often have a tough protective coat, called a cell wall, surrounding the plasma membrane, which encloses a single compartment containing the cytoplasm and the DNA. In the electron microscope, the cell interior typically appears as a matrix of varying texture without any obvious organized internal structure. The cells reproduce quickly by dividing in two.

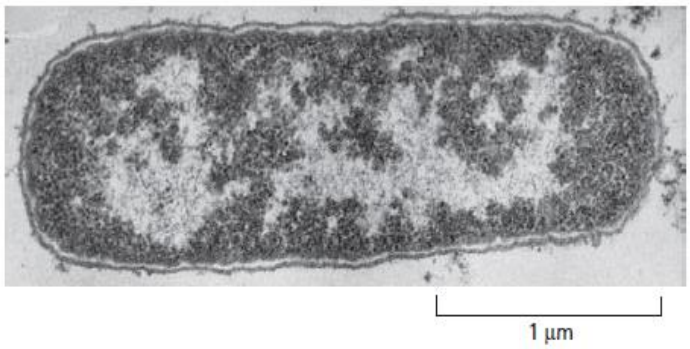


Figure 2-1 **The bacterium Escherichia coli The bacterium Escherichia coli (E. coli)** is understood more thoroughly than any other living organism. An electron micrograph of a longitudinal section is show here; the cell's Dna is concentrated in the lightly stained region.

(Courtesy of e. Kellenberger.) (Alberts B. H.)

Most procaryotes live as single-celled organisms, although some join together to form chains, clusters, or other organized multicellular structures. In shape and structure, procaryotes may seem simple and limited, but in terms of chemistry, they are the most diverse and inventive class of cells. These creatures exploit an enormous range of habitats, from hot puddles of volcanic mud to the interiors of other living cells, and they vastly outnumber other living organisms on Earth. Some are aerobic, using oxygen to oxidize food molecules; some are strictly anaerobic and are killed by the slightest exposure to oxygen. Mitochondria-the organelles that generate energy for the eucaryotic cell-are thought to have evolved from aerobic bacteria that took to living inside the anaerobic ancestors of today's eucaryotic cells. Thus our own oxygen-based metabolism can be regarded as a product of the activities of bacterial cells. (Alberts B. H.)

2.1.1.1.2 Eucaryotes

Eucaryotic cells, in general, are bigger and more elaborate than bacteria and archaea. Some live independent lives as single-celled organisms, such as amoebae and yeasts; others live in multicellular assemblies. All of the more complex multicellular organisms-including plants, animals, and fungi-are formed from eucaryotic cells. By definition, all eucaryotic cells have a nucleus. But possession of a nucleus goes hand-in-hand with possession of a variety of other organelles, subcellular structures that perform specialized functions. Most of these are likewise common to all eucaryotic organisms. (Alberts B. H.)

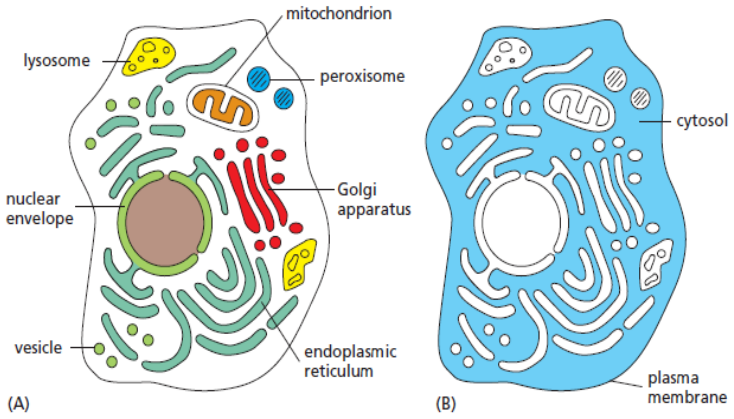


Figure 2-2 **Membrane-enclosed organelles are distributed throughout the cytoplasm**. (A) a variety of membrane enclosed compartment exist within eucaryotic cells, each specialized to perform a different function. (B) the rest of the cell, excluding all these organelles, is called the cytosol (colored blue). (Alberts B. H.)

We will now take a look at the main organelles found in eucaryotic cells from the point of view of their functions.

Lysosomes are small, irregularly shaped organelles in which intracellular digestion occurs, releasing nutrients from food particles and breaking down unwanted molecules for recycling or excretion.

And **peroxisomes** are small, membrane-enclosed vesicles that provide a contained environment for reactions in which hydrogen peroxide, a dangerously reactive chemical, is generated and degraded.

Membranes also form many different types of small **vesicles** involved in the transport of materials between one membrane-enclosed organelle and another.

If we were to strip the plasma membrane from a eucaryotic cell and then remove all of its membrane-enclosed organelles, including nucleus, endoplasmic reticulum, Golgi apparatus, mitochondria, chloroplasts, and so on, we would be left with the **cytosol**. In other words, the cytosol is the part of the cytoplasm that is not partitioned off within intracellular membranes. In most cells, the cytosol is the largest single compartment. It contains a host of large and small molecules, crowded together so closely that it behaves more like a water-based gel than a liquid solution. The cytosol is the site of many chemical reactions that are fundamental to the cell's existence.

The early steps in the breakdown of nutrient molecules take place in the cytosol, for example, and it is here that the cell performs one of its key synthetic processes—the manufacture of proteins. **Ribosomes**, the molecular machines that make the protein molecules, are visible with the electron microscope as small particles in the cytosol, often attached to the cytosolic face of the endoplasmic.

(Alberts B. H.)

2.1.1.1.2.1 Golgi apparatus

Stacks of flattened membrane-enclosed sacs constitute the **Golgi apparatus**, which receives and often chemically modifies the molecules made in the endoplasmic reticulum and then directs them to the exterior of the cell or to various locations inside the cell. (Alberts B. H.)

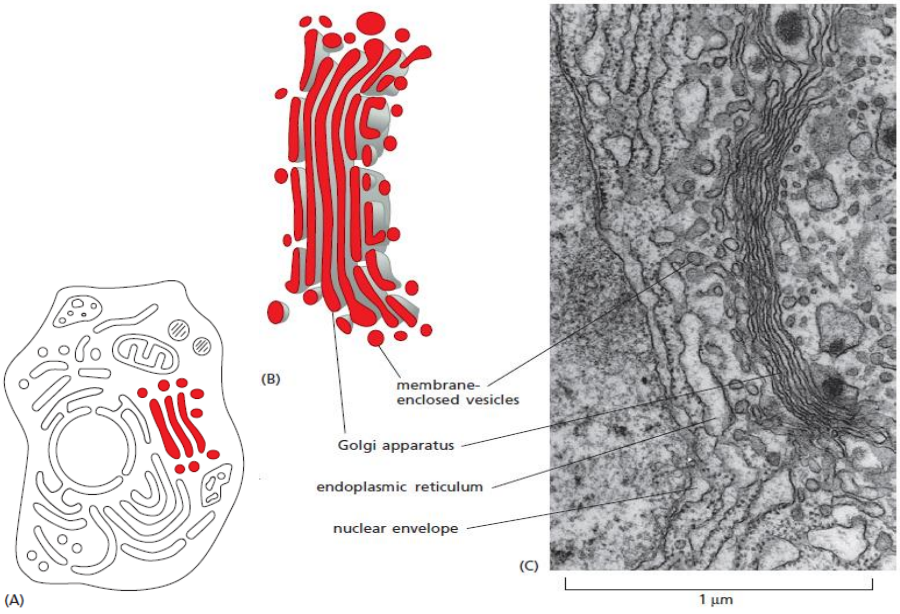


Figure 2-3 **The Golgi apparatus resembles a stack of flattened discs**. This organelle is just visible under the light microscope but is often inconspicuous. the Golgi apparatus is involved in the synthesis and packaging of molecules destined to be secreted from the cell, as well as in the routing of newly synthesized proteins to the correct cellular compartment. (A) Schematic diagram of an animal cell with the Golgi apparatus colored red. (B) Drawing of the Golgi apparatus reconstructed from electron microscope images. the organelle is composed of flattened sacs of membrane stacked in layers. Many small vesicles are seen nearby; some of these have pinched off from the Golgi stack, while others are destined to fuse with it. Only one stack is shown here, but several can be present in a cell. (C) electron micrograph of the Golgi apparatus from a typical animal cell. (C, courtesy of Brij J. Gupta.) (Alberts B. H.)

2.1.1.1.2.2 Mitochondria

Mitochondria are present in essentially all eucaryotic cells, and they are among the most conspicuous organelles in the cytoplasm mitochondria are generators of chemical energy for the cell. They harness the energy from the oxidation of food molecules, such as sugars, to produce adenosine triphosphate, or ATP—the basic chemical fuel that powers most of the cell's activities. Because the mitochondrion consumes oxygen and releases carbon dioxide in the course of this activity, the entire process is called cellular respiration—essentially, breathing on a cellular level. (Alberts B. H.)

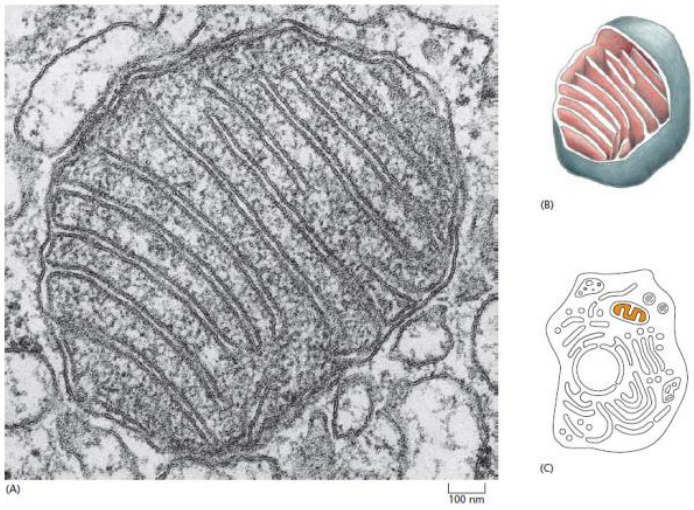


Figure 2-4 **Mitochondria have a distinctive structure**.
(A) an electron micrograph of a cross-section of a mitochondrion reveals the extensive folding of the inner membrane. (B) this three-dimensional representation of the arrangement of the mitochondrial membranes shows the smooth outer membrane and the highly convoluted inner membrane. the inner membrane contains most of the proteins responsible for cellular respiration, and it is highly folded to provide a large surface area for this activity. (C) In this schematic cell, the interior space of the mitochondrion is colored. (a, courtesy of Daniel S. Friend.) (Alberts B. H.)

2.1.1.1.2.3 Endoplasmic Reticulum

The **endoplasmic reticulum** (ER)—an irregular maze of interconnected spaces enclosed by a membrane - is the site where most cell membrane components, as well as materials destined for export from the cell, are made. (Alberts B. H.)

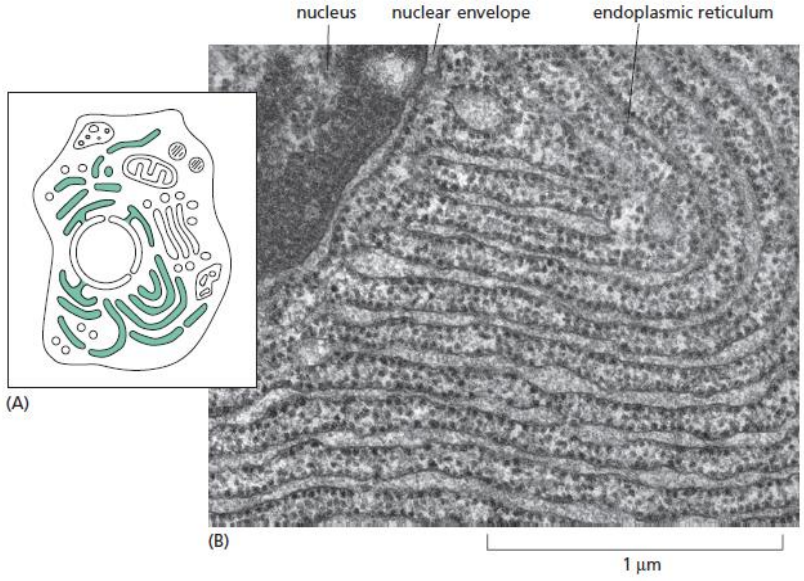


Figure 2-5 **Many cellular components are produced in the endoplasmic reticulum**.
(A) Schematic diagram of an animal cell shows the endoplasmic reticulum in green.
(B) electron micrograph of a thin section of a mammalian pancreatic cell shows a small part of the endoplasmic reticulum (er), of which there are vast tracts in this cell type, which is specialized for protein secretion. note that the er is continuous with the membrane of the nuclear envelope. the black particles studding the particular region of the er shown here are ribosomes—the molecular assemblies that perform protein synthesis. Because of its appearance, ribosome-coated er is often called "rough er." (B, courtesy of Lelio Orci.) (Alberts B. H.)

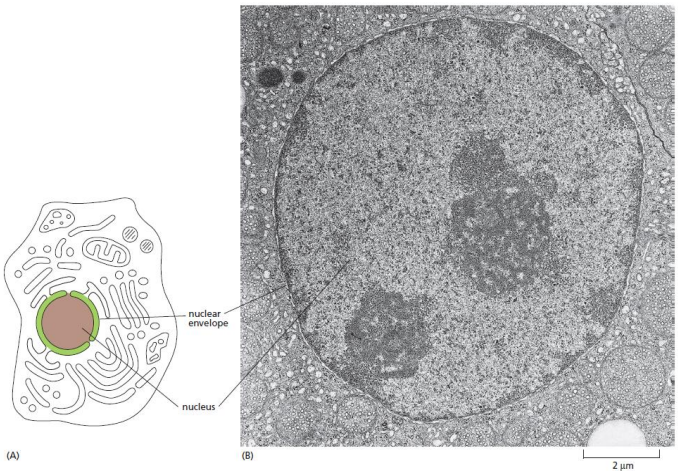


Figure 2-6 **The nucleus contains most of the DNA in a eucaryotic cell.** (A) In this drawing of a typical animal cell—complete with its extensive system of membrane-enclosed organelles—the nucleus is colored brown, the nuclear envelope is green, and the cytoplasm (the interior of the cell outside the nucleus) is white. (B) an electron micrograph of a nucleus in a mammalian cell. Individual chromosomes are not visible because the Dna is dispersed as fine threads throughout the nucleus at this stage of the cell's growth. (B, courtesy of Daniel S. Friend.) (Alberts B. H.)

2.1.1.1.2.4 Nucleus

The **nucleus** is usually the most prominent organelle in a eucaryotic. It is enclosed within two concentric membranes that form the nuclear envelope, and it contains molecules of **DNA**—extremely long polymers that encode the genetic information of the organism. In the light microscope, these giant DNA molecules become visible as individual chromosomes when they become more compact as a cell prepares to divide into two daughter cells. DNA also stores the genetic information in procaryotic cells; these cells lack a distinct nucleus not because they lack DNA, but because they do not keep their DNA inside a nuclear envelope, segregated from the rest of the cell contents. (Alberts B. H.)

Figure 2-7 **Chromosomes become visible when a cell is about to divide**. As a eucaryotic cell prepares to divide, its DNA becomes compacted or condensed into threadlike chromosomes that can be distinguished in the light microscope. The photographs show three successive steps in this process in a cultured cell from a newt's lung. (Courtesy of Conly L. rieder.) (Alberts B. H.)

2.1.2 DeoxyriboNucleic Acid

A molecule of deoxyribonucleic acid (DNA) consists of two long polynucleotide chains. Each of these DNA chains, or DNA strands, is composed of four types of nucleotide subunits, and the two chains are held together by hydrogen bonds between the base portions of the nucleotides.

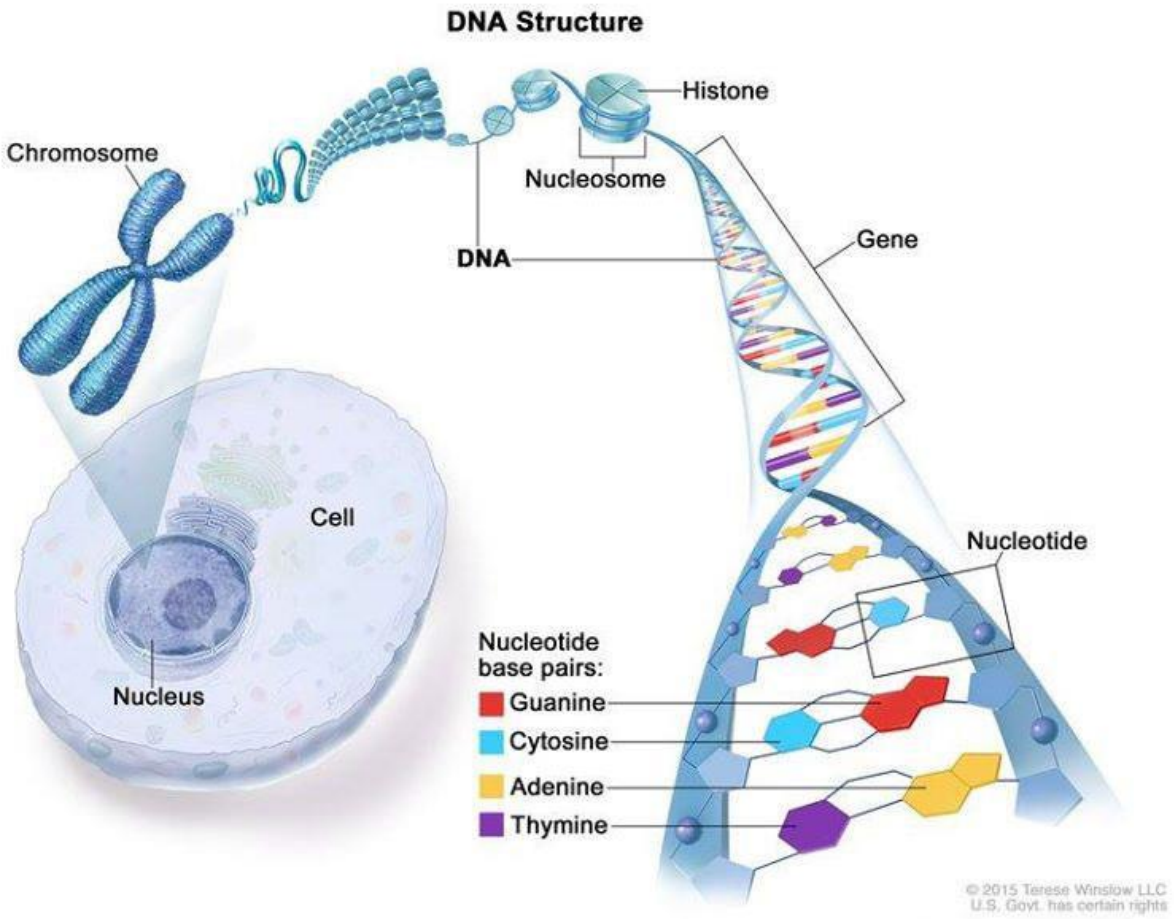


Figure 2-8

Cancer is caused by certain changes to genes, the basic physical units of inheritance. Genes are arranged in long strands of tightly packed DNA called chromosomes.

Credit: Terese Winslow

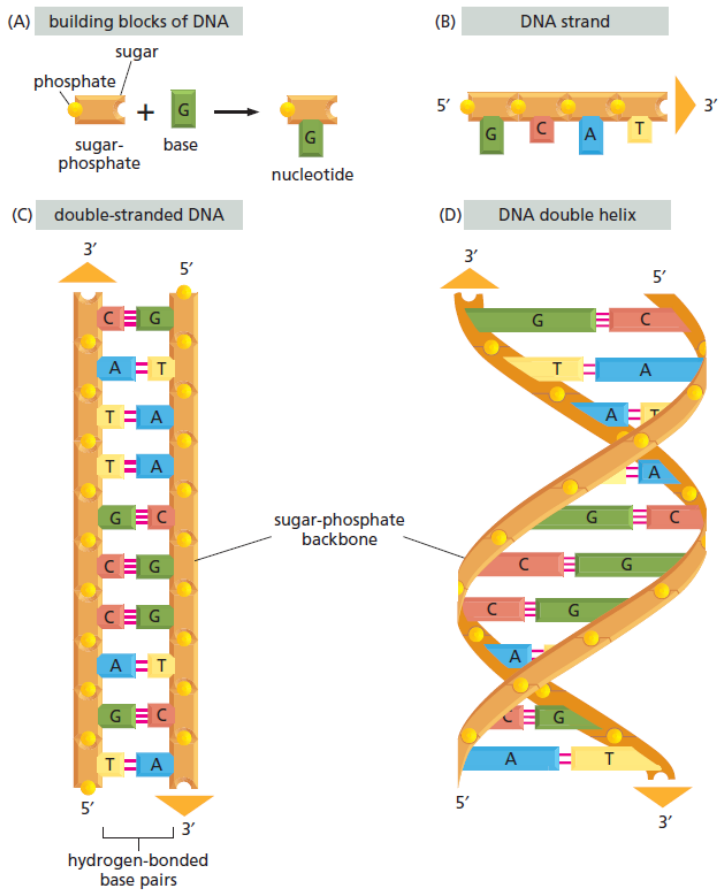


Figure 2-9 **DNA is made of four nucleotide building blocks**. (A) each nucleotide is composed of a sugar– phosphate covalently linked to a base.(B) the nucleotides are covalently linked together into polynucleotide chains, with a sugar–phosphate backbone from which the bases (a, C, G, and t) extend. (C) a DNa molecule is composed of two polynucleotide chains (DNa strands) held together by hydrogen bonds between the paired bases. the arrows on the DNa strands indicate the polarities of the two strands, which run antiparallel to each other in the DNa molecule. (D) although the DNa is shown straightened out in (C), in reality, it is wound into a double helix, as shown here. (Alberts B. H.)

Nucleotides

Nucleotides are composed of a five-carbon sugar to which are attached one or more phosphate groups and a nitrogen-containing base. For the nucleotides in DNA, the sugar is deoxyribose attached to a single phosphate group (hence the name deoxyribonucleic acid); the base may be either adenine (A), cytosine (C), guanine (G), or thymine (T). The nucleotides are covalently linked together in a chain through the sugars and phosphates, which thus form a “backbone” of alternating sugar–phosphate–sugar. Because it is only the base that differs in each of the four types of subunits, each polynucleotide chain in DNA can be thought of as a necklace (the backbone) strung with four types of beads (the four bases A, C, G, and T). These same symbols (A, C, G, and T) are also commonly used to denote the four different nucleotides—that is, the bases with their attached sugar and phosphate groups.

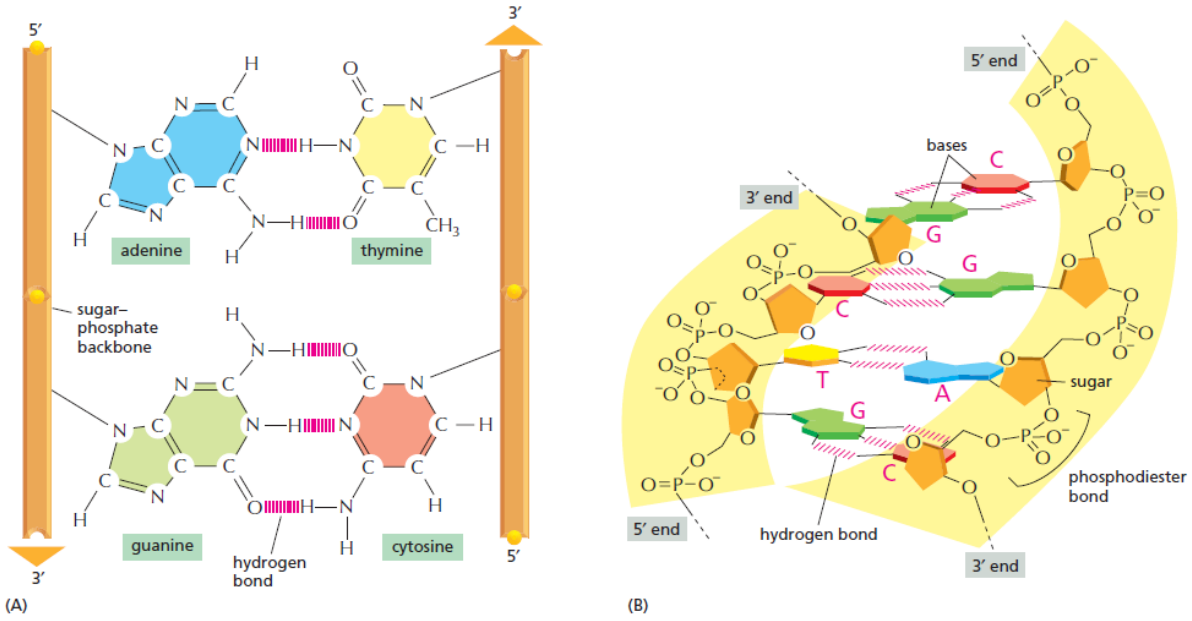


Figure 2-10 **The two strands of the DNA double helix are held together by hydrogen bonds between complementary base pairs**. (A) the shapes and chemical structure of the bases allow hydrogen bonds to form efficiently only between a and t and between G and C, where atoms that are able to form hydrogen bonds can be brought close together without perturbing the double helix. two hydrogen bonds form between A and T, whereas three form between G and C. the bases can pair in this way only if the two polynucleotide chains that contain them are antiparallel—that is, oriented in opposite polarities. (B) A short section of the double helix viewed from its side. Four base pairs are shown. The nucleotides are linked together covalently by phosphodiester bonds through the 3′-hydroxyl (–OH) group of one sugar and the 5′-phosphate ($–PO_4$) of the next. This linkage gives each polynucleotide strand a chemical polarity; that is, its two ends are chemically different. The 3′ end carries an unlinked –oh group attached to the 3′ position on the sugar ring; the 5′ end carries a free phosphate group attached to the 5′ position on the sugar ring. (Alberts B. H.)

The way in which the nucleotide subunits are linked together gives a DNA strand a chemical **polarity.** If we imagine that each nucleotide has a knob (the phosphate) and a hole, each chain, formed by interlocking knobs with holes, will have all of its subunits lined up in the **same orientation**. Moreover, the two ends of the chain can be easily distinguished, as one will have a hole (the 3′ hydroxyl) and the other a knob (the 5′ phosphate). This polarity in a DNA chain is indicated by referring to one end as the 3′ end and the other as the 5′ end. This convention is based on the details of the chemical linkage between the nucleotide subunits.

The two polynucleotide chains in the DNA double helix are held together by hydrogen-bonding between the bases on the different strands. All the bases are therefore on the inside of the helix, with the sugar–phosphate backbones on the outside. The bases do not pair at random, however: A always pairs with T, and G always pairs with C.

In each case, a bulkier two-ring base is paired with a single-ring base (a pyrimidine). Each purine–pyrimidine pair is called a base pair, and this complementary base-pairing enables the base pairs to be packed in the energetically most favorable arrangement in the interior of the double helix. In this arrangement, each base pair is of similar width, thus holding the sugar–phosphate back-bones an equal distance apart along the DNA molecule. The members of each base pair can fit together within the double helix because the two strands of the helix run antiparallel to each other—that is, they are oriented with opposite polarities.

DNA encodes information in the order, or sequence, of the nucleotides along each strand. Each base—A, C, T, or G—can be considered as a letter in a four-letter alphabet that is used to spell out biological messages in the chemical structure of the DNA. Organisms differ from one another because their respective DNA molecules have different **nucleotide sequences** and, consequently, carry different biological messages. (Alberts B. H.)

2.1.2.1 Chromosomes

In eucaryotic cells, very long double-stranded DNA molecules are packaged into structures called **chromosomes**, which not only fit readily inside the nucleus but can be easily apportioned between the two daughter cells at each cell division. The complex task of packaging DNA is accomplished by specialized proteins that bind to and fold the DNA, generating a series of coils and loops that provide increasingly higher levels of organization and prevent the DNA from becoming an unmanageable tangle. Amazingly, the DNA is compacted in a way that allows it to remain accessible to all of the enzymes and other proteins that replicate it, repair it, and direct the expression of its genes.

In eucaryotes, such as ourselves, the DNA in the nucleus is distributed among a set of different chromosomes. The human genome, for example, contains approximately 3.2×10^9 nucleotides parceled out into 24 chromosomes. Each chromosome consists of a single, enormously long, linear DNA molecule associated with proteins that fold and pack the fine thread of DNA into a more compact structure. The complex of DNA and protein is

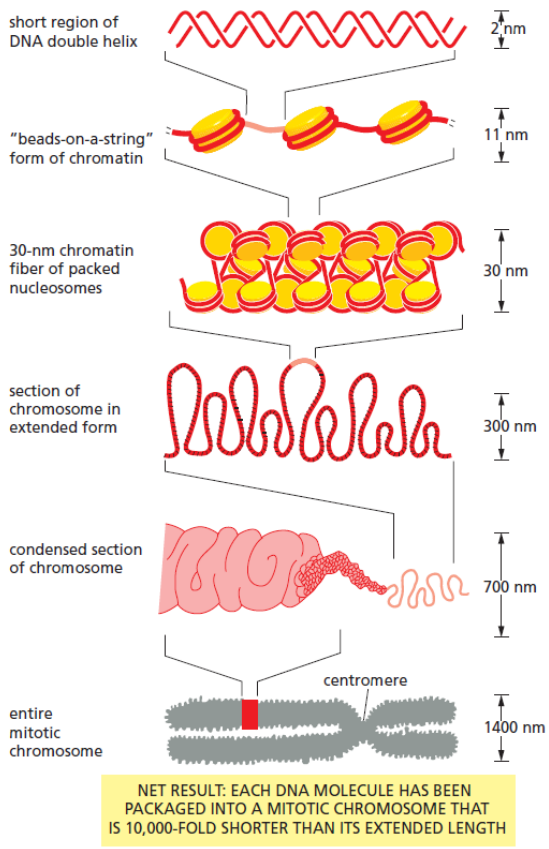


Figure 2-11 **DNA packing occurs on several levels in chromosomes**. This schematic drawing shows some of the levels thought to give rise to the highly condensed mitotic chromosome. (Alberts B. H.)

called chromatin. In addition to the proteins involved in packaging the DNA, chromosomes are also associated with many other proteins involved in gene expression, DNA replication, and DNA repair.

In addition to being different sizes, human chromosomes can be distinguished from one another by a variety of techniques.

Each chromosome can be “painted” a different color using sets of chromosome-specific DNA molecules coupled to different fluorescent dyes. This involves the technique of DNA hybridization. The most important function of chromosomes is to carry the genes—the functional units of heredity. (Alberts B. H.)

2.1.3 RiboNucleic Acid

Once the structure of DNA (deoxyribonucleic acid) had been determined in the early 1950s, it became clear that the hereditary information in cells is encoded in DNA’s sequence of nucleotides. Even before the DNA code had been broken, it was known that the information contained in genes somehow directed the synthesis of proteins. Proteins are the principal constituents of cells and determine not only their structure but also their functions. We have encountered some of the thousands of different kinds of proteins that cells can make. The properties and function of a protein molecule are determined by the linear order—the sequence—of the different amino acid subunits in its polypeptide chain: each type of protein has its own unique amino acid sequence, and this sequence dictates how the chain will fold to give a molecule with a distinctive shape and chemistry. The genetic instructions carried by DNA must therefore specify the amino acid sequences of proteins.

DNA does not direct protein synthesis itself, but acts rather like a manager, delegating the various tasks to a team of workers. When a particular protein is needed by the cell, the nucleotide sequence of the appropriate section of an immensely long DNA molecule in a chromosome is first copied into another type of nucleic acid—**RNA (ribonucleic acid)**. These RNA copies of short segments of the DNA are then used to direct the synthesis of the protein. Many thousands of these conversions from DNA to protein occur each second in every cell in our bodies.

Like DNA, **RNA** is a linear polymer made of four different types of nucleotide subunits linked together by phosphodiester bonds. It differs from DNA chemically in two respects: (1) the nucleotides in RNA are ribonucleotides—that is, they contain the sugar ribose (hence the name ribonucleic acid) rather than deoxyribose; (2) although, like DNA, RNA contains the bases adenine (A), guanine (G), and cytosine (C), it contains uracil (U) instead of the thymine (T) found in DNA. Because U, like T, can base-pair by hydrogen-bonding with A, the complementary base-pairing properties described for DNA in Chapter 5 apply also to RNA.

Although their chemical differences are small, DNA and RNA differ quite dramatically in overall structure. Whereas DNA always occurs in cells as a double-stranded helix, RNA is single-stranded. This difference has important functional consequences. Because an RNA chain is single-stranded, it can fold up into a variety of shapes, just as a polypeptide chain folds up to form the final shape of a protein; double-stranded DNA cannot fold in this fashion. The ability to fold into a complex three dimensional shape allows RNA to carry out functions in cells in addition to conveying information between DNA and protein.

FROM DNA TO RNA

The first step a cell takes in reading out one of its many thousands of genes is to copy the nucleotide sequence of that gene into RNA. The process is called **transcription** because the information, though copied into another chemical form, is still written in essentially the same language—the language of nucleotides.

Transcription produces RNA Complementary to One strand of DNA

All of the RNA in a cell is made by transcription, a process that has certain similarities to DNA replication. Transcription begins with the opening and unwinding of a small portion of the DNA double helix to expose the bases on each DNA strand. One of the two strands of the DNA double helix then acts as a template for the synthesis of RNA. Ribonucleotides are added, one by one, to the growing RNA chain, and as in DNA replication, the nucleotide sequence of the RNA chain is determined by complementary base-pairing with the DNA template. When a good match is made, the incoming ribonucleotide is covalently linked to the growing RNA chain in an enzymatically catalyzed reaction. The RNA chain produced by transcription—the **transcript**—is therefore elongated one nucleotide at a time and has a nucleotide sequence exactly complementary to the strand of DNA used as the template.

Several Types of RNA are produced in Cells

The vast majority of genes carried in a cell's DNA specify the amino acid sequence of proteins, and the RNA molecules that are copied from these genes (and that ultimately direct the synthesis of proteins) are collectively called **messenger RNA** (**mRNA**). In eucaryotes, each mRNA typically carries information transcribed from just one gene, coding for a single protein; in bacteria, a set of adjacent genes is often transcribed as a single mRNA that therefore carries the information for several different proteins.

The final product of other genes, however, is the RNA itself. These, non-messenger RNAs, like proteins, serve as regulatory, structural, and enzymatic components of cells, and they play key parts in translating the genetic message into protein. **Ribosomal RNA** (**rRNA**) forms the core of the ribosomes, on which mRNA is translated into protein, and **transfer RNA** (**tRNA**) forms the adaptors that select amino acids and hold them in place on a ribosome for their incorporation into protein. Other small RNAs, called **microRNAs** (**miRNAs**), serve as key regulators of eucaryotic gene expression. (Alberts B. H.)

The sequence of **messenger RNA** is complementary to the sequence of the bottom strand of DNA and is identical to the top strand of DNA, except for the replacement of T with U. A messenger RNA includes a sequence of nucleotides that corresponds to the sequence of amino acids in the protein. This part of the nucleic acid is called the **coding region**. Because mRNA is an exact copy of the DNA coding regions, mRNA analysis can be used to identify polymorphisms in coding regions of DNA. A **polymorphism** is a DNA region for which nucleotide sequence variants exist in a population of organisms. Such variations can sometimes explain the occurrence of a disease or enzyme deficiency within a population. Hence, a considerable effort has been put into trying to identify such variations.

Microarray technology can be used both in the identification of polymorphisms and in the diagnosis of polymorphism-related disease. In eukaryotic cells, the initial pre-mRNA transcription product can be many times longer than needed for translation into protein. At the end of a eukaryotic gene, there is a

regulatory region to which various proteins bind, causing the gene to be transcribed at the right time and in the right amount. A region at the end of the gene contains a sequence encoding the termination of transcription. In the genes of many eukaryotes, the protein-encoding sequence is interrupted by varying numbers of segments called **introns**. The coding sequence segments interrupted by the introns are called **exons**. Introns are removed in the **splicing** process to generate the final mature mRNA ready to be translated by the protein synthesis machinery. (LEE)

FROM RNA TO PROTEIN

In contrast, the conversion of the information in RNA into protein represents a **translation** of the information into another language that uses quite different symbols. Because there are only 4 different nucleotides in mRNA but 20 different types of amino acids in a protein, this translation cannot be accounted for by a direct one-to-one correspondence between a nucleotide in RNA and an amino acid in protein. The rules by which the nucleotide sequence of a gene, through the medium of mRNA, is translated into the amino acid sequence of a protein are known as the genetic code.

2.1.4 Central dogma of molecular biology

The conversion of genotype to phenotype requires information stored in DNA to be converted to protein. The nature of information flow in cells was first described by **Francis Crick as the central dogma of molecular biology.** Information passes in one direction from the gene (DNA) to an RNA copy of the gene, and the RNA copy directs the sequential assembly of a chain of amino acids into a protein. Stated briefly,

DNA → RNA → protein

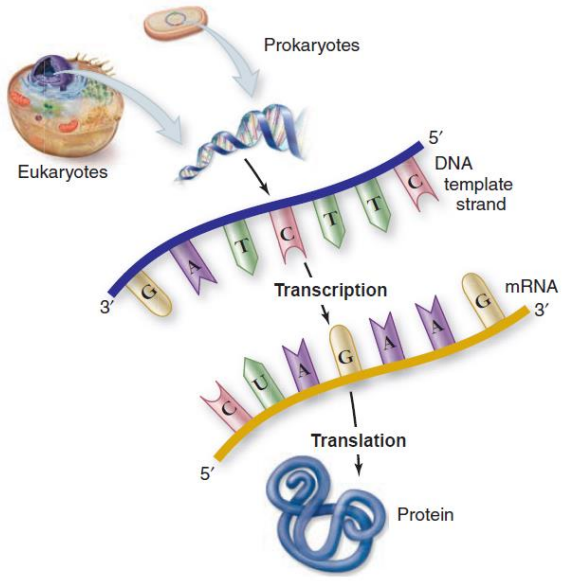


Figure 2-12 **The central dogma of molecular biology.** DNA is transcribed to make mRNA, which is translated to make a protein. (Biology -- 9th ed.)

The central dogma provides an intellectual framework that describes information flow in biological systems. We call the DNA-to-RNA step transcription because it produces an exact copy of the DNA, much as a legal transcription contains the exact words of a court proceeding. The RNA-to-protein step is termed translation because it requires translating from the nucleic acid to the protein "languages." Since the original formulation of the central dogma, a class of viruses called retroviruses was discovered that can convert their RNA genome into a DNA copy, using the viral enzyme reverse transcriptase. This conversion violates the direction of information flow of the central dogma, and the discovery forced an updating of the possible flow of information to include this "reverse" flow from RNA to DNA. (Biology -- 9th ed.)

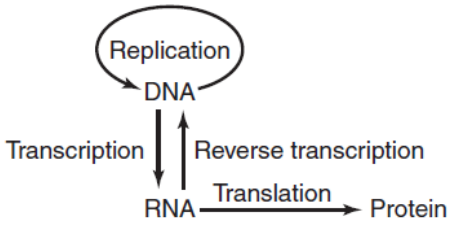


Figure 2-13 **The updated direction of information flow of the central dogma**

2.1.5 Genes

Genes are the units of the DNA sequence that control the identifiable hereditary traits of an organism. A **gene** can be defined as a segment of DNA that specifies a functional RNA. The total set of genes carried by an individual or a cell is called its **genome**. The genome defines the genetic construction of an organism or cell, or the **genotype**. The **phenotype**, on the other hand, is the total set of characteristics displayed by an organism under a particular set of environmental factors. The outward appearance of an organism (phenotype) may or may not directly reflect the genes that are present (genotype). Today the complete genome sequences of several species are known, including several bacteria, yeasts, and humans. With microarray technology we can study the expression of all the genes in an organism simultaneously. Such genome-wide studies will help to uncover and decipher cellular processes from a completely new perspective. (LEE)

2.1.6 The Genetic Code

The sequence of nucleotides in DNA is important not because of its structure, but because it codes for the sequence of amino acids that dictate the structure of a protein with a defined function, be it structural or catalytic. The relationship between a sequence of DNA and the sequence of the corresponding protein is called the genetic code. The genetic code is read in groups of three nucleotides, or codons, each of which represents one amino acid. Because each position in the three nucleotide codon could be one of the four bases A, C, G, and T, there are a total of $4 \times 4 \times 4 = 64$ possible different codons, each representing an amino acid or a signal to terminate translation. As there are only 20 common amino acids, several different codons can code for the same amino acid (the genetic code is said to be degenerate due to this many-to-one relationship). Since the genetic code is read in non-overlapping triplets, there are three possible ways of translating any nucleotide sequence into a protein, depending on the starting point. These are called **reading frames**. A reading frame that starts with a special initiation codon (AUG-methionine) and extends through a series of codons representing amino acids until it ends at one of three termination codons (UAA, UAG, UGA) can potentially be translated into a protein and is called an **open reading frame** (ORF). A long open reading frame is unlikely to exist by chance. The identification of a lengthy open reading frame is strong evidence that the sequence is translated into protein in that frame. An open reading frame for which no protein product has been identified is sometimes called an **unidentified reading frame** (URF).

2.1.7 Gene Expression and Microarrays

Gene expression is the process by which mRNA, and eventually protein, is synthesized from the DNA template of each gene. The first stage of this process is **transcription**, when an RNA copy of one strand of the DNA is produced. In eukaryotes it is followed by RNA **splicing**, during which the introns are cut out of the primary transcript and a mature mRNA is made. As part of the maturation process, a tail of adenine nucleotides is added to the 3' end of the mRNA. This poly A tail can vary greatly in length and is believed to stabilize the mRNA molecule. Transcription and splicing of RNA occur in the nucleus. The next stage of gene expression is the **translation** of the mRNA into protein. This occurs in the cytoplasm. In the process of gene expression RNA provides not only the essential substrate (mRNA) but also components of the protein synthesis apparatus (tRNA, rRNA).

Some protein-encoding genes are transcribed more or less constantly; they are sometimes called **housekeeping genes** and are always needed for basic reactions. Other genes may be rendered unreadable or, to suit the functions of the organism, readable only at particular moments and under particular external conditions. The signal that masks or unmasks a gene may come from outside the cell; for example, from a nutrient or a hormone. Special **regulatory sequences** in the DNA dictate whether a gene will respond to the signals, and they in turn affect the transcription of the protein-encoding gene. Understanding which genes are expressed under which condition gives invaluable information about the biological processes in the cell. The power of microarray technology lies in its ability to measure the expression of thousands of genes simultaneously. (LEE)

2.1.8 Hybridization

The specific base pairing of nucleic acids is the foundation of microarray technology. The specific pairing of an artificial DNA sequence probe with its biological counterpart allows for exact identification of the sought-after unique sequence or gene.

Because of the base-pairing arrangments, the two strands of DNA can separate and re-form very quickly under physiological conditions that disrupt the hydrogen bonds between the bases but are much too mild to pose any threat to the covalent bonds in the backbone of the DNA. The process of strand separation is called **denaturation** or **melting**. Because of the complementarity of the base pairs, the two separated complementary strands can be re-formed into a double helix (the two strands are then said to be annealed). This process is called **renaturation**. The technique of renaturation can be extended to allow any two complementary nucleic acid sequences to anneal with each other to form a duplex structure.

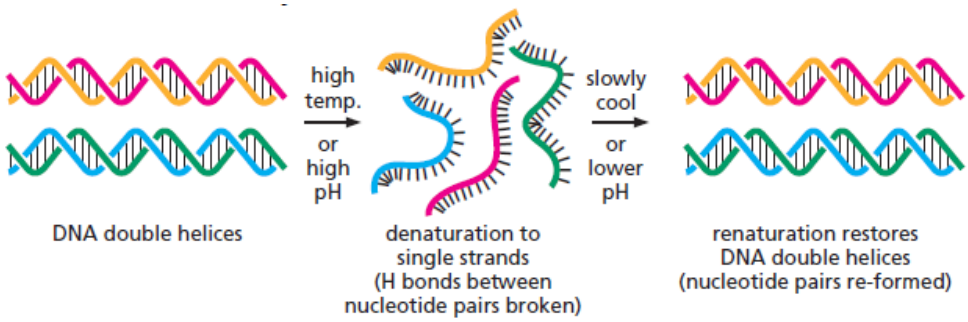


Figure 2-14 **A molecule of DNA can undergo denaturation and renaturation (hybridization).** For hybridization to occur, the two single strands must have complementary nucleotide sequences that allow base-pairing. In this example, the red and orange strands are complementary to each other, and the blue and green strands are complementary to each other. (Alberts B. H.)

Hybridization is the biochemical method on which DNA microarray technology is based. Nucleic acid sequences can be compared in terms of complementarity that is determined by the rules for base pairing. In a perfect duplex of DNA, the strands are precisely complementary. It is possible to measure complementarity because the denaturation of DNA is reversible under appropriate conditions. Detecting and identifying nucleic acid (DNA, mRNA) with a labeled cDNA probe that is complementary to it is an application of nucleic acid hybridization. DNA microarrays utilize hybridization reactions between single-stranded fluorescent dye-labeled nucleic acids to be interrogated and single-stranded sequences immobilized on the chip surface. (LEE)

2.1.9 Complementary DNA (cDNA)

Complementary DNA (cDNA) is used in recombinant DNA technology. cDNA is complementary to a given mRNA and is usually made by the enzyme reverse transcriptase, first discovered in retroviruses. Reverse transcription allows a mature mRNA to be retrieved as cDNA without the interruption of non-coding introns. The coexistence of mRNA and cDNA establishes the general principle that information in the form of either type of nucleic acid sequence can be converted into the other type. In microarray technology the process of reverse transcription is frequently used to incorporate fluorescent dyes into cDNA complementary to the mRNA transcripts. (LEE)

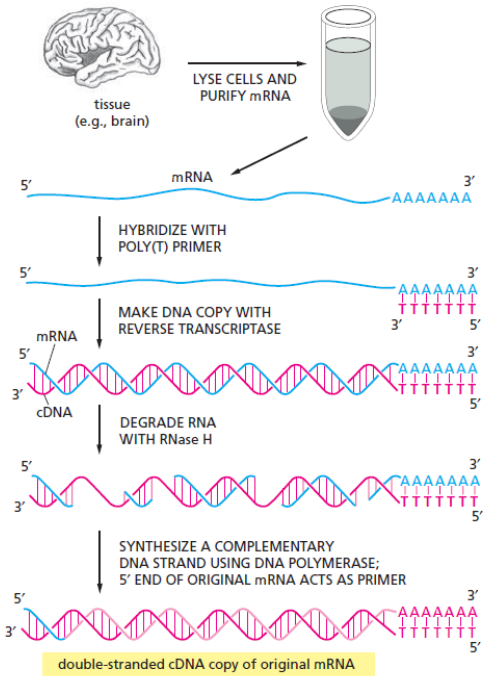


Figure 2-15 **Complementary DNA (cDNA) can be prepared from mRNA.** Total mRNa is extracted from a particular tissue, and DNA copies (cDNa) of the mRNa molecules are produced by the enzyme reverse transcriptase. For simplicity, the copying of just one of these mRNas into cDNa is illustrated here. (Alberts B. H.)

2.1.10 PCR

Now that so many genome sequences are available, genes can be cloned directly without the need to construct DNA libraries first. A technique called **the polymerase chain reaction (PCR)** makes this rapid cloning possible. PCR allows the DNA from a selected region of a genome to be amplified a billion fold, effectively "purifying" this DNA away from the remainder of the genome. Two sets of DNA oligonucleotides, chosen to flank the desired nucleotide sequence of the gene, are synthesized by chemical methods. These oligonucleotides are then used to prime DNA synthesis on single strands generated by heating the DNA from the entire genome. The newly synthesized DNA is produced in a reaction catalyzed **in vitro** by a purified DNA polymerase, and the primers remain at the 5' ends of the final DNA fragments that are made.

Nothing special is produced in the first cycle of DNA synthesis; the power of the PCR method is revealed only after repeated rounds of DNA synthesis. Every cycle doubles the amount of DNA synthesized in the previous cycle. Because each cycle requires a brief heat treatment to separate the two strands of the template DNA double helix, the technique requires the use of a special DNA polymerase, isolated from a thermophilic bacterium, that is stable at much higher temperatures than normal, so that it is not denatured by the repeated heat treatments. With each round of DNA synthesis, the newly generated fragments serve as templates in their turn, and within a few cycles the predominant product is a single species of DNA fragment whose length corresponds to the distance between the two original primers.

In practice, 20 30 cycles of reaction are required for effective DNA amplification, with the products of each cycle serving as the DNA templates for the next hence the term polymerase "chain reaction." A single cycle requires only about 5 minutes, and the entire procedure can be easily automated. PCR thereby makes possible the "cell-free molecular cloning" of a DNA fragment in a few hours, compared with the several days required for standard cloning procedures. This technique is now used routinely to clone DNA from genes of interest directly starting either from genomic DNA or from mRNA isolated from cells. The PCR method is extremely sensitive; it can detect a single DNA molecule in a sample. Trace amounts of RNA can be analyzed in the same way by first transcribing them into DNA with reverse transcriptase. The PCR cloning technique has largely replaced Southern blotting for the diagnosis of genetic diseases and for the detection of low levels of viral infection. It also has great promise in forensic medicine as a means of analyzing minute traces of blood or other tissues even as little as a single cell and identifying the person from whom they came by his or her genetic "fingerprint". (Alberts W. J.)

2.2 MICROARRAY TECHNOLOGY

A DNA microarray consists of a solid surface, usually a microscope slide, onto which DNA molecules have been chemically bonded. The purpose of a microarray is to detect the presence and abundance of labeled nucleic acids in a biological sample, which will hybridize to the DNA on the array via Watson–Crick duplex formation, and which can be detected via the label. In the majority of microarray experiments, the labeled nucleic acids are derived from the mRNA of a sample or tissue, and so the microarray measures gene expression. The power of a microarray is that there may be many thousands of different DNA molecules bonded to an array, and so it is possible to measure the expression of many thousands of genes simultaneously.

Also, comparing healthy and diseased cells can yield vital information on the causes of diseases. Microarrays have been successfully applied to several biological problems and, as arrays become more easily available to researchers, the popularity of these kinds of experiments will increase. The demand for good statistical analysis regimens and tools tailored for microarray data analysis will increase as the popularity of microarrays grows. The future will likely bring many new microarray applications, each with its own demands for specialized statistical analysis.

In order to analyze any experimental data correctly, it is fundamental to understand the experiments that generated the data. Microarray experiments contain many steps, each with its individual noise and variation. The final result may be affected by any of the steps in the process. Good experimental design and careful statistical analysis are required for successful interpretation of microarray data. (Stekel) (LEE)

Microarray technology has evolved from Ed Southern's insight that labeled nucleic acid molecules could be used to identify nucleic acid molecules attached to a solid support. Hybridization methods, such as Southern and Northern blots, colony hybridizations, and dot blots, have long been used to identify and quantify nucleic acids in biological samples. These methods traditionally attempt to identify and measure only one gene or transcript at a time.

Hybridization methods have evolved from these early membrane-based, radioactive detection embodiments to highly parallel quantitative methods using fluorescence detection. Some key innovations have made it possible to develop techniques that analyze hundreds or thousands of hybridizations in parallel. The first was the use of non-porous solid supports, such as nylon filters or glass slides, which facilitate miniaturization. The second was the development of methods for spatial synthesis and robotic spotting of oligonucleotides and cDNAs on a very small scale. These methods have made it possible to generate arrays with very high densities of DNA, allowing tens of thousands of genes to be represented in areas smaller than standard glass microscope slides. In fact, today it is technically possible to generate arrays of probes representing all the genes of a genome on a single slide. Finally, improvements in fluorescent labeling of nucleic acids, fluorescent-based detection, and image processing have improved the accuracy of microarrays.

Before describing the process of generating and using microarrays in more detail, a clarification of the nomenclature is needed. At least two nomenclature systems currently exist in the literature for referring to DNA hybridization partners. There is no general consensus on the usage of the terms **probe** and

target, and researchers have used these two terms interchangeably in a number of publications. With respect to the nucleic acids whose entwining represents the hybridization reaction, the identity of one is defined as it is tethered to the solid phase, making up the microarray itself. The identity of the other is revealed by hybridization. Nature Genetics14 and Duggan et al.15 adopted the nomenclature that the tethered nucleic acids spotted on the array are the probes, and the fluor-tagged cDNAs from a complex mRNA mixture extracted from cells are the targets. (LEE)

2.2.1 The Technology behind DNA Microarrays

When DNA microarrays are used for measuring the concentration of messenger RNA in living cells, a **probe** of one DNA strand that matches a particular messenger RNA in the cell is used. The concentration of a particular messenger is a result of expression of its corresponding gene, so this application is often referred to as **expression analysis**. When different probes matching all messenger RNAs in a cell are used, a snapshot of the total messenger RNA pool of a living cell or tissue can be obtained. This is often referred to as an **expression profile** because it reflects the expression of every single measured gene at that particular moment. Expression profile is also sometimes used to describe the expression of a single gene over a number of conditions.

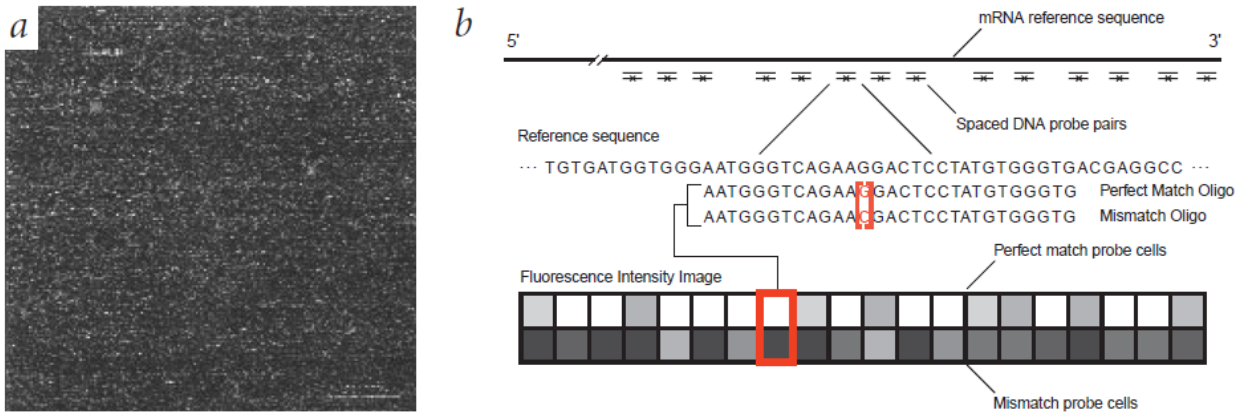


Figure 2-16 **The Affymetrix GeneChip technology**.
Gene expression monitoring with oligonucleotide arrays.
a, A single 1.28´1.28 cm array containing probe sets for approximately 40,000 human genes and ESTs. This array contains features smaller than 22´22 mm and only four probe pairs per gene or EST.
b, Expression probe and array design. Oligonucleotide probes are chosen based on uniqueness criteria and composition design rules. For eukaryotic organisms, probes are chosen typically from the 3´ end of the gene or transcript (nearer to the poly(A) tail) to reduce problems that may arise from the use of partially degraded mRNA. The use of the PM minus MM differences averaged across a set of probes greatly reduces the contribution of background and cross-hybridization and increases the quantitative accuracy and reproducibility of the measurement. (Robert J. Lipshutz)

Expression analysis can also be performed by a method called **serial analysis of gene expression** (**SAGE**). Instead of using microarrays, SAGE relies on traditional DNA sequencing to identify and enumerate the messenger RNAs in a cell.

Another traditional application of DNA microarrays is to detect mutation in specific genes. The massively parallel nature of DNA microarrays allows the simultaneous screening of many, if not all, possible mutations within a single gene. This is referred to as **genotyping**.

The treatment of array data does not depend so much on the technology used to gather the data as it depends on the application in question. For expression analysis the field has been dominated in the past by two major technologies. The Affymetrix, Inc. GeneChip system uses prefabricated oligonucleotide chips. Custom-made chips use a robot to spot cDNA, oligonucleotides, or PCR products on a glass slide or membrane. More recently, several new technologies have entered the market. (Knudsen, 2006)

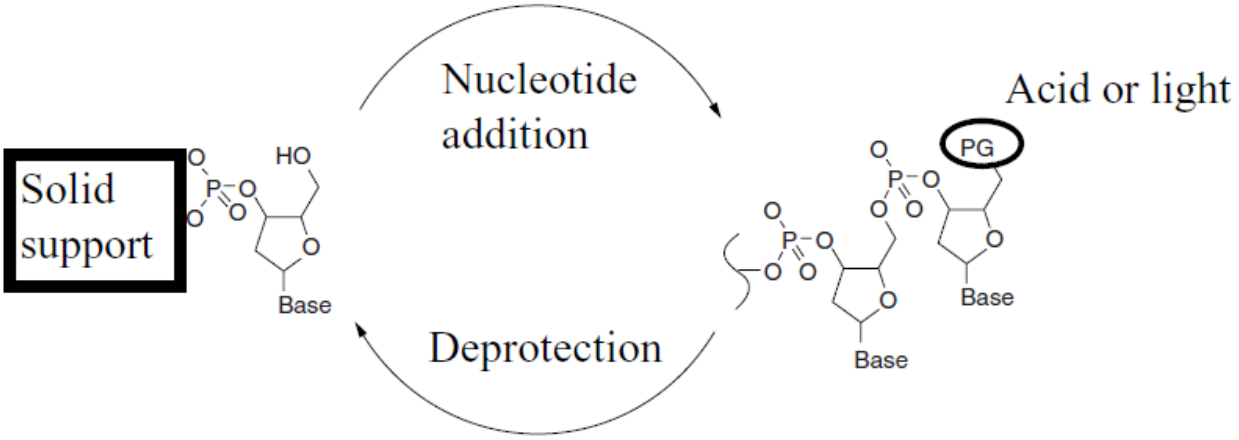


Figure 2-17 **In-situ synthesis of oligonucleotides**. The oligonucleotides are built on the glass array one base at a time. At each step, the base is added via the reaction between the hydroxyl group 5of the terminal base and the phosphate group of the next base. There is a protective group on the 5 of the base being added, which prevents the addition of more than one base at each step. Following addition, there is a deprotection step at which the protective group is converted to a hydroxyl group to allow addition of the next base.

2.2.2 Spotted cDNA Arrays

In spotted cDNA arrays full-length cDNA clones or expressed sequence tag (EST) libraries are robotically spotted and immobilized on the support. Many laboratories already have cDNA libraries, so generation of these arrays requires only investment in the robotic equipment to spot, or array, the cDNA. Spotted cDNA arrays have an advantage over other types of arrays in that unknown sequences can be spotted. Thus, for organisms for which no or only limited genome sequence information is available, spotted cDNA microarrays are the only choice for genome-wide transcriptional profiling. (LEE)

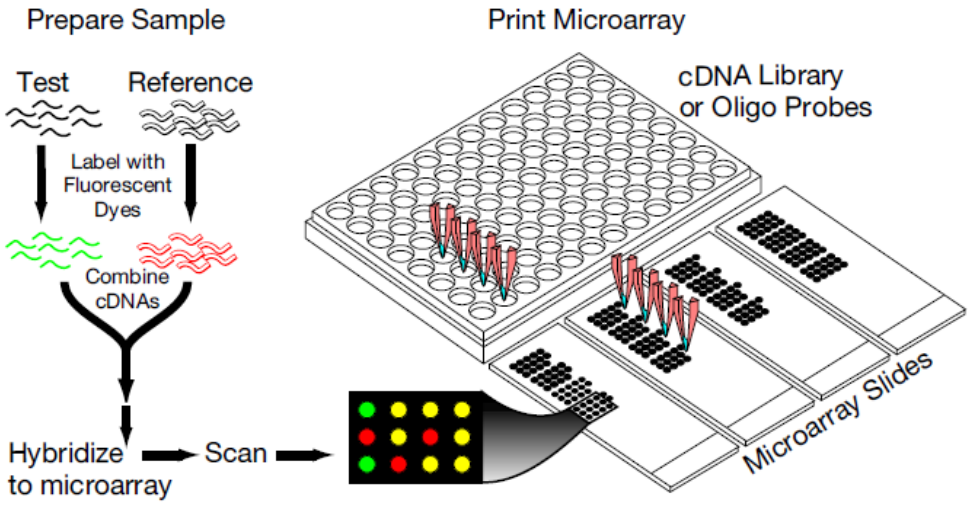


Figure 2-18 **The spotted array technology**. A robot is used to transfer probes in solution from a microtiter plate to a glass slide where they are dried. Extracted mRNA from cells is converted to cDNA and labeled fluorescently. Reference sample is labeled red and test sample is labeled green. After mixing, they are hybridized to the probes on the glass slide. After washing away unhybridized material, the chip is scanned with a confocal laser and the image is analyzed by computer. (Knudsen, 2006)

2.2.3 Spotted Oligonucleotide Arrays

This is the technology by which the first microarrays were manufactured. The array is made using a spotting robot via three main steps:

1. Making the DNA probes to put on the array
2. Spotting the DNA onto the glass surface of the array with the spotting robot
3. Post spotting processing of the glass slide

There are three main types of spotted array, which can be subdivided in two ways: by the type of DNA probe, or by the attachment chemistry of the probe to the glass. The DNA probes used on a spotted array can either be polymerase chain reaction (PCR) products or oligonucleotides. In the first case, highly parallel PCR is used to amplify DNA from a clone library, and the amplified DNA is purified. In the second case, DNA oligonucleotides are presynthesised for use on the array. (Stekel, 2003)

2.2.4 In-Situ Oligonucleotide Arrays

In-situ oligonucleotide arrays were developed by Fodor et al.16 and Affymetrix, Inc. In-situ oligonucleotide arrays use a combination of photolithography and solid-phase oligonucleotide chemistry to synthesize short oligonucleotide probes (25-mer oligos) directly on the solid support surface. The number of oligonucleotides (50,000 probes per 1.28 square centimeters) on a chip manufactured by this method vastly exceeds what can be achieved by spotting solution robotically. This takes place by covalent reaction between the 5' hydroxyl group of the sugar of the last nucleotide to be attached and the phosphate group of the next nucleotide. Each nucleotide added to the oligonucleotide on the glass has a protective group on its 5' position to prevent the addition of more than one base during each round of synthesis. The protective group is then converted to a hydroxyl group either with acid or with light before the next round of synthesis.

Affymetrix Inc. has chosen to utilize this advantage to construct an array with several oligonucleotide probes and cross-hybridization controls for each target gene. However, the researcher has little, if any, control over what probes are used on pre-manufactured arrays like the Affymetrix GeneChip arrays. On the other hand, comparison of results between different laboratories is facilitated by the use of products from a common manufacturer.

For in-situ oligonucleotide arrays, the test and reference samples (or the treatment and control samples) are hybridized separately on different chips. In contrast, for either spotted cDNA arrays or spotted oligonucleotide arrays, a test and a reference sample labeled with two different fluorescent dyes are commonly simultaneously hybridized on the same arrays. This difference affects how microarray data generated with single-color or two-color arrays are analyzed. (Stekel, 2003) (LEE)

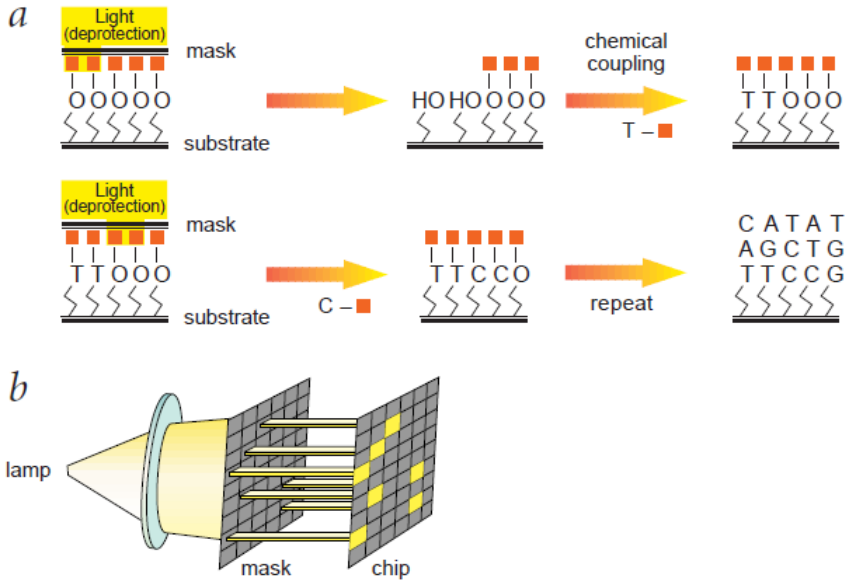


Figure 2-19 **Affymetrix technology.** Affymetrix arrays are manufactured using in-situ synthesis with a light-mediated deprotection step. During each round of synthesis, a single base is added to appropriate parts of the array. A mask is used to direct light to the appropriate regions of the array so that the base is added to the correct features. Each step requires a different mask. The masks are expensive to produce, but once made, it is straightforward to use them to manufacture a large number of identical arrays. (Reproduced with Permission from Affymetrix Inc.) (Stekel, 2003)

a. Light directed oligonucleotide synthesis. A solid support is derivatized with a covalent linker molecule terminated with a photolabile protecting group. Light is directed through a mask to deprotect and activate selected sites, and protected nucleotides couple to the activated sites. The process is repeated, activating different sets of sites and coupling different bases allowing arbitrary DNA probes to be constructed at each site.

b. Schematic representation of the lamp, mask and array. (Robert J. Lipshutz)

2.2.5 Affymetrix GeneChip Technology

Affymetrix uses equipment similar to that which is used for making silicon chips for computers, and thus allows mass production of very large chips at reasonable cost. Where computer chips are made by creating masks that control a **photolithographic** process for removal or deposition of silicon material on the chip surface, Affymetrix uses masks to control synthesis of oligonucleotides on the surface of a chip. The standard phosphoramidite method for synthesis of oligonucleotides has been modified to allow light control of the individual steps. The masks control the synthesis of several hundred thousand squares, each containing many copies of an oligo. So the result is several hundred thousand different oligos, each of them present in millions of copies.

That large number of oligos, up to 25 nucleotides long, has turned out to be very useful as an experimental tool to replace all experimental detection procedures that in the past relied on using oligonucleotides: Southern, Northern, and dot blotting as well as sequence specific probing and mutation detection.

For expression analysis, up to 40 oligos are used for the detection of each gene. Affymetrix has chosen a region of each gene that (presumably) has the least similarity to other genes. From this region 11 to 20 oligos are chosen as perfect match (PM) oligos (i.e., perfectly complementary to the mRNA of that gene). In addition, they have generated 11 to 20 mismatch (MM) oligos, which are identical to the PM oligos except for the central position 13, where one nucleotide has been changed to its complementary nucleotide. Affymetrix claims that the MM oligos will be able to detect nonspecific and background hybridization, which is important for quantifying weakly expressed mRNAs. However, for weakly expressed mRNAs where the signal-to-noise ratio is smallest, subtracting mismatch from perfect match adds considerably to the noise in the data (Schadt et al., 2000). That is because subtracting one noisy signal from another noisy signal yields a third signal with even more noise.

The hybridization of each oligo to its target depends on its sequence. All 11 to 20 PM oligos for each gene have a different sequence, so the hybridization will not be uniform. That is of limited consequence as long as we wish to detect only changes in mRNA concentration between experiments.

To detect hybridization of a target mRNA by a probe on the chip, we need to label the target mRNA with a fluorochrome. The steps from cell to chip usually are as follows:

- Extract total RNA from cell (usually using TRIzol from Invitrogen or RNeasy from QIAGEN).
- Separate mRNA from other RNA using poly-T column (optional).
- Convert mRNA to cDNA using reverse transcriptase and a poly-T primer.
- Amplify resulting cDNA using T7 RNA polymerase in the presence of biotin-UTP and biotin-CTP, so each cDNA will yield 50 to 100 copies of biotin-labeled cRNA.
- Incubate cRNA at 94 degrees Celsius in fragmentation buffer to produce cRNA fragments of length 35 to 200 nucleotides.
- Hybridize to chip and wash away non hybridized material.
- Stain hybridized biotin-labeled cRNA with streptavidin-phycoerythrin and wash.
- Scan chip in confocal laser scanner (optional).
- Amplify the signal on the chip with goat IgG and biotinylated antibody.
- Scan chip in scanner.

Usually, 5 to 10 µg of total RNA are required for the procedure. But new improvements to the cDNA synthesis protocols reduce the required amount to 100 ng. If two cycles of cDNA synthesis and cRNA synthesis are performed, the detection limit can be reduced to 2 ng of total RNA (Baugh et al., 2001). MessageAmp kits from Ambion allow up to 1000 times amplification in a single round of T7 polymerase amplification. (Knudsen, 2006)

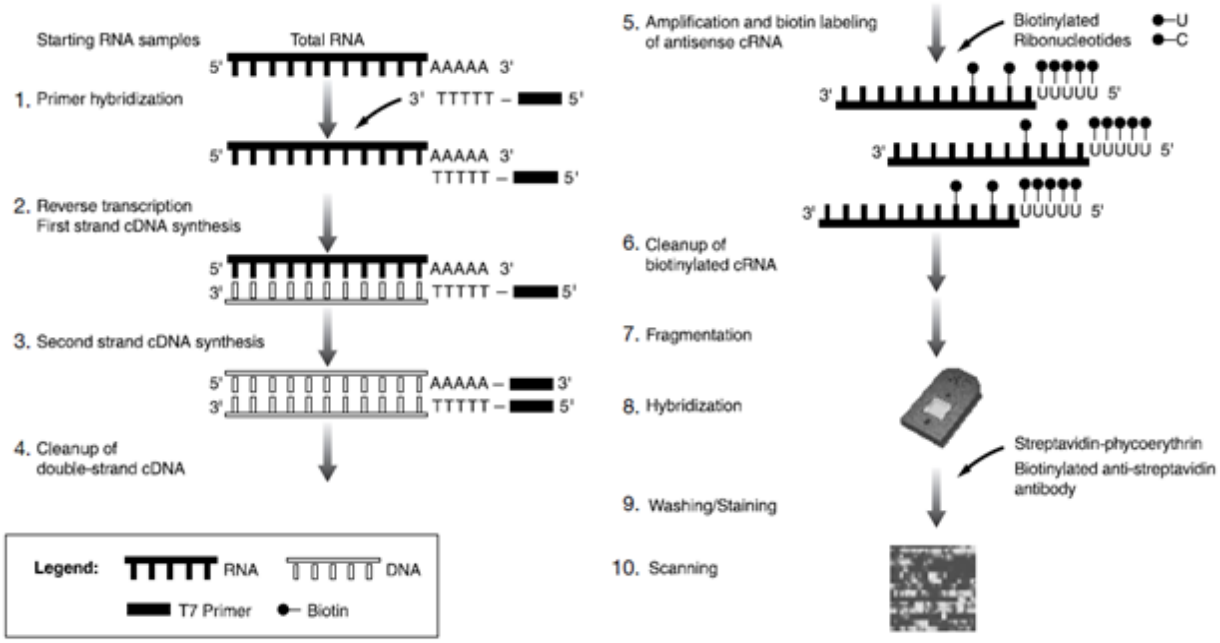


Figure 2-20 **Preparation of sample for GeneChip arrays**. Messenger RNA is extracted from the cell and converted to cDNA. It then undergoes an amplification and labeling step before fragmentation and hybridization to 25-mer oligos on the surface of the chip. After washing of unhybridized material, the chip is scanned in a confocal laser scanner and the image is analyzed by computer. (Image courtesy of Affymetrix)

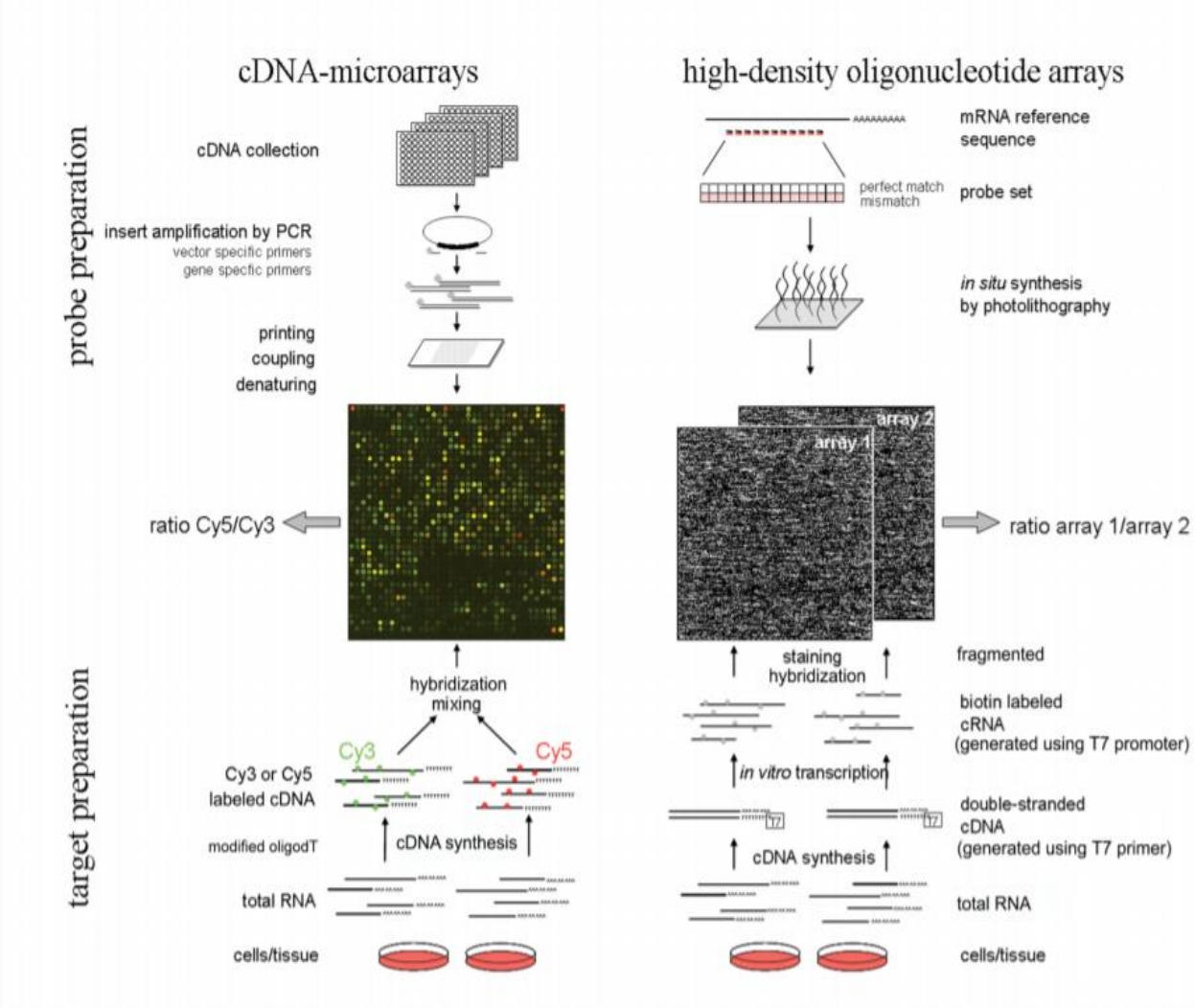


Figure 2-21 **Schematic overview of spotted cDNA microarrays and high-density oligonucleotide arrays.** cDNA microarrays : Array preparation: inserts from cDNA collections or libraries are amplified and the PCR products printed at specified sites on glass slides using high-precision arraying robots. These probes are attached by chemical linkers. Target preparation: RNA from 2 different tissues or cell populations is used to synthesize cDNA in the presence of nucleotides labeled with 2 different fluorescent dyes (eg: Cy3 and Cy5). Both samples are mixed in a small volume of hybridization buffer and hybridized to the array, resulting in competitive binding of differentially labeled cDNAs to the corresponding array elements. High resolution confocal fluorescence scanning of the array with two different wavelengths corresponding to the dyes used provides relative signal intensities and ratios of mRNA abundance for the genes represented on the array. High-density oligonucleotide microarrays : Array preparation: sequences of 16-20 short oligonucleotides (typically 25mer) are chosen from the mRNA reference sequence of each gene, often representing the unique part of the transcript. Light-directed, in situ oligonucleotide synthesis is used to generate high- density probe arrays containing over 300,000 individual elements. Target preparation: Total RNA from different tissues or cell populations is used to generate cDNA carrying a transcriptional start site for T7 DNA polymerase. During IVT, biotin-labeled nucleotides are incorporated into the synthesized cRNA molecules which is then fragmented. Each target sample is hybridized to a separate probe array and target binding is detected by staining with a fluorescent dye coupled to streptavidin. Signal intensities of probe array element sets on different arrays are used to calculate relative mRNA abundance for the genes represented on the array. Modified and reprinted with permission from Nature Cell Biology (Vol. 3, No. 8, pp. E190-E195) Copyright ©2001 Macmillan Publishers Limited. 262 (The microarray: Potential aplications for ophthalmic research)

2.3 CANCER

We pay a price for having bodies that can renew and repair themselves. The delicately adjusted mechanisms that control these processes can go wrong, leading to catastrophic disruption of the body's structure. Foremost among the diseases of tissue renewal is **cancer**, which stands alongside infectious illness, malnutrition, war, and heart disease as a major cause of death among humans. In Europe and North America, for example, one in four of us will die of cancer.

Cancers arise from violations of the basic rules of social cell behavior. To make sense of the origins and progress of the disease, and to devise treatments, we have to draw upon almost every part of our knowledge of how cells work and interact in tissues. Conversely, much of what we know about cell and tissue biology has been discovered as a byproduct of cancer research.

Cancer is due to failures of the mechanisms that usually control the growth and proliferation of cells. During normal development and throughout adult life, intricate genetic control systems regulate the balance between cell birth and death in response to growth signals, growth-inhibiting signals, and death signals. Cell birth and death rates determine adult body size, and the rate of growth in reaching that size. In some adult tissues, cell proliferation occurs continuously as a constant tissue-renewal strategy. Intestinal epithelial cells, for instance, live for just a few days before they die and are replaced; certain white blood cells are replaced as rapidly, and skin cells commonly survive for only 2–4 weeks before being shed. The cells in many adult tissues, however, normally do not proliferate except during healing processes. Such stable cells (e.g., hepatocytes, heart muscle cells, neurons) can remain functional for long periods or even the entire lifetime of an organism.

The losses of cellular regulation that give rise to most or all cases of cancer are due to genetic damage. **Mutations** in two broad classes of genes have been implicated in the onset of cancer: **proto-oncogenes** and **tumor-suppressor genes**. Proto-oncogenes are activated to become oncogenes by mutations that cause the gene to be excessively active in growth promotion. Either increased gene expression or production of a hyperactive product will do it. Tumor-suppressor genes normally restrain growth, so damage to them allows inappropriate growth. Many of the genes in both classes encode proteins that help regulate cell birth (i.e., entry into and progression through the cell cycle) or cell death by **apoptosis**; others encode proteins that participate in repairing damaged DNA. Cancer commonly results from mutations that arise during a lifetime's exposure to carcinogens, which include certain chemicals and ultraviolet radiation. Cancer-causing mutations occur mostly in somatic cells, not in the germ-line cells, and somatic cell mutations are not passed on to the next generation. In contrast, certain inherited mutations, which are carried in the germ line, increase the probability that cancer will occur at some time. In a destructive partnership, somatic mutations can combine with inherited mutations to cause cancer.

Thus the cancer-forming process, called **oncogenesis** or **tumorigenesis**, is an interplay between genetics and the environment. Most cancers arise after genes are altered by carcinogens or by errors in the copying and repair of genes. Even if the genetic damage occurs only in one somatic cell, division of this cell will transmit the damage to the daughter cells, giving rise to a **clone** of altered cells. Rarely, however, does mutation in a single gene lead to the onset of cancer. More typically, a series of mutations in multiple genes creates a progressively more rapidly proliferating cell type that escapes

normal growth restraints, creating an opportunity for additional mutations. Eventually the clone of cells grows into a **tumor**. In some cases cells from the primary tumor migrate to new sites (metastasis), forming secondary tumors that often have the greatest health impact.

Metastasis is a complex process with many steps. Invasion of new tissues is nonrandom, depending on the nature of both the metastasizing cell and the invaded tissue. Metastasis is facilitated if the tumor cells produce growth and angiogenesis factors (blood vessel growth inducers). Motile, invasive, aggregating, deformable cells are most dangerous. Tissues under attack are most vulnerable if they produce growth factors and readily grow new vasculature. They are more resistant if they produce anti-proliferative factors, inhibitors of proteolytic enzymes, and anti-angiogenesis factors.

Research on the genetic foundations of a particular type of cancer often begins by identifying one or more genes that are mutationally altered in tumor cells. Subsequently it is important to learn whether an altered gene is a contributing cause for the tumor, or an irrelevant side event. Such investigations usually employ multiple approaches: epidemiological comparisons of the frequency with which the genetic change is associated with a type of tumor, tests of the growth properties of cells in culture that have the particular mutation, and the testing of mouse models of the disease to see if the mutation can be causally implicated. A more sophisticated analysis is possible when the altered gene is known to encode a component of a particular molecular pathway (e.g., an intracellular signaling pathway). In this case it is possible to alter other components of the same pathway and see whether the same type of cancer arises.

Because the multiple mutations that lead to formation of a tumor may require many years to accumulate, most cancers develop later in life. The occurrence of cancer after the age of reproduction may be one reason that evolutionary restraints have not done more to suppress cancer. The requirement for multiple mutations also lowers the frequency of cancer compared with what it would be if tumorigenesis were triggered by a single mutation. However, huge numbers of cells are, in essence, mutagenized and tested for altered growth during our lifetimes, a sort of evolutionary selection for cells that proliferate. Fortunately the tumor itself is not inherited.

Tumors arise with great frequency, especially in older individuals, but most pose little risk to their host because they are localized and of small size. We call such tumors **benign**; an example is warts, a benign skin tumor. The cells composing benign tumors closely resemble, and may function like, normal cells. The cell-adhesion molecules that hold tissues together keep benign tumor cells, like normal cells, localized to the tissues where they originate. A fibrous capsule usually delineates the extent of a benign tumor and makes it an easy target for a surgeon. Benign tumors become serious medical problems only if their sheer bulk interferes with normal functions or if they secrete excess amounts of biologically active substances like hormones. Acromegaly, the overgrowth of head, hands, and feet, for example, can occur when a benign pituitary tumor causes overproduction of growth hormone. In contrast, cells composing a **malignant** tumor, or **cancer**, usually grow and divide more rapidly than normal, fail to die at the normal rate (e.g., chronic lymphocytic leukemia, a tumor of white blood cells), or invade nearby tissue without a significant change in their proliferation rate (e.g., less harmful tumors of glial cells). Some malignant tumors, such as those in the ovary or breast, remain localized and encapsulated, at least for a time. When these tumors progress, the cells invade surrounding tissues, get into the body's circulatory system, and establish secondary areas of proliferation, a process called metastasis. Most

malignant cells eventually acquire the ability to **metastasize**. Thus the major characteristics that differentiate metastatic (or malignant) tumors from benign ones are their invasiveness and spread. (Alberts B. H.) (Lodish)

2.4 MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. (expertsystem.com)

The name machine learning was coined in 1959 by Arthur Samuel (Some Studies in Machine Learning Using the Game of Checkers, 1959). Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field

***Definition**: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E (Mitchell, 1997)*

Generally, there are four types of Machine Learning Algorithms:

- **Supervised learning** refers to any machine learning process that learns a function from an input type to an output type using data comprising examples that have both input and output values. Two typical examples of supervised learning are classification learning and regression. In these cases, the output types are respectively categorical (the classes) and numeric. Supervised learning stands in contrast to unsupervised learning, which seeks to learn structure in data, and to reinforcement learning in which sequential decision-making policies are learned from reward with no examples of “correct” behavior.
- **Unsupervised learning** refers to any machine learning process that seeks to learn structure in the absence of either an identified output or feedback. Three typical examples of unsupervised learning are clustering, association rules, and self-organizing maps.
- **Reinforcement learning** describes a large class of learning problems characteristic of autonomous agents interacting in an environment: sequential decision-making problems with delayed reward. Reinforcement-learning algorithms seek to learn a policy (mapping from states to actions) that maximizes the reward received over time. Unlike in supervised learning problems, in reinforcement learning problems, there are no labeled examples of correct and incorrect behavior. However, unlike unsupervised learning problems, a reward signal can be perceived.
- **Semi-supervised learning** uses both labeled and unlabeled data to perform an otherwise supervised learning or unsupervised learning task.

(Sammut C., 2010)

2.4.1 Classification

In common usage, the word classification means to put things into categories, group them together in some useful way. If we are screening for a disease, we would group people into those with the disease and those without. We, as humans, usually do this because things in a group, called a class in machine learning, share common characteristics. If we know the class of something, we know a lot about it. In machine learning, the term classification is most commonly associated with a particular type of learning where examples of one or more classes, labeled with the name of the class, are given to the learning algorithm.

The input data for a classification task is a collection of records. Each record, also known as an instance or example, is characterized by a tuple (x,y), where x is the attribute set and y is a special attribute, designated as the class label (also known as category or target attribute). The class label, on the other hand, must be a discrete attribute. This is a key characteristic that distinguishes classification from regression, a predictive modeling task in which g is a continuous attribute. (Pang Ning Tan)

Also (Pang Ning Tan) defines **classification** in his book (Introduction to Data Mining.) as the task of learning a target function / that maps each attribute set x to one of the predefined class labels y.

The target function is also known informally as a **classification model**. A classification model is useful for the following purposes.

- **Descriptive Modeling** A classification model can serve as an explanatory tool to distinguish between objects of different classes.
- **Predictive Modeling** A classification model can also be used to predict the class label of unknown records. A classification model can be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown record.

A classification rule is an IF-THEN rule. The condition of the rule (the rule body or antecedent) typically consists of a conjunction of Boolean terms, each one constituting a constraint that needs to be satisfied by an example. If all constraints are satisfied, the rule is said to fire, and the example is said to be covered by the rule. The rule head (also called the consequent or conclusion) consists of a single class value, which is predicted in case the rule fires. This is in contrast to association rules, which allow multiple features in the head. (Sammut C., 2010)

2.4.2 Binary Classification

Binary classification problems (Duda et al. 2001) consider assigning an individual to one of two categories, by measuring a series of attributes. An example is medical diagnosis for a single medical condition (say disease vs. no disease) based on a battery of tests. (Science Direct)

There are many influential binary classification methods such as kernel methods (Hofmann et al., 2008), ensemble methods (Polikar, 2006), and deep learning methods (Bengio, 2009). Support vector machine (SVM) (Vapnik, 1999) is a classical kernel method. Ensemble methods include boosting (Freund and Schapire, 1997; Friedman et al., 2000) and random forest (RF) (Breiman, 2001). Deep learning methods are based on artificial neural networks (ANNs) (Bishop and et al., 1995). (Science Direct)

2.4.3 Classification Algorithms

There are a very large number of classification algorithms. The most common are separated in linear and non-linear. A simple way of representing the output from machine learning is a linear model, the output of which is just the sum of the attribute values, except that weights are applied to each attribute before adding them together. The trick is to come up with good values for the weights-ones that make the model's output match the desired output. Here, the output and the inputs-attribute values-are all numeric. Linear models can also be applied to binary classification problems. In this case, the line produced by the model separates the two classes: It defines where the decision changes from one class value to the other. Such a line is often referred to as the decision boundary. (Ian H. Witten)

2.4.3.1 Logistic Regression

Statisticians use the word regression for the process of predicting a numeric quantity, and regression model is another term for this kind of linear model.

Logistic regression provides a mechanism for applying the techniques of linear regression to classification problems. It utilizes a linear regression model of the form

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + +\beta_n x$$

where x_1 to x_n represent the values of the n attributes and β_0 to β_n represent weights. This model is mapped onto the interval $[0,1]$ using $P(c_0|x_1 \dots x_n) = \frac{1}{1+e^{-z}}$ where c_0 represents class 0.

2.4.3.2 Linear Discriminant Analysis

A discriminant is a function that takes an input variable x and outputs a class label y for it. A linear discriminant is a discriminant that uses a linear function of the input variables and more generally a linear function of some vector function of the input variables $f(x)$. This entry focuses on one such linear discriminant function called Fisher's linear discriminant. Fisher's discriminant works by finding a projection of input variables to a lower dimensional space while maintaining a class separability property. (Sammut C., 2010)

Given N observed training data points $\{(x_i, y_i)\}_{i=1}^{N}$ where $y_i \in \{1, \dots, \Omega\}$ is the label for an input variable $x_i \in \mathbb{R}^d$, our task is to find the underlying discriminant function, $f: \mathbb{R}^d \rightarrow \{1, \dots, \Omega\}$. The linear discriminant seeks a projection of d-dimensional input onto a line in the direction of $w \in \mathbb{R}^d$, such that

$$f(x) = w^T x$$

Fisher's criterion maximizes a large separation between the projected class means while simultaneously minimizing a variance within each class. (Sammut C., 2010)

Learning LDA Models

LDA makes some simplifying assumptions about our data:

- That our data is Gaussian, that each variable is is shaped like a bell curve when plotted.
- That each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.

With these assumptions, the LDA model estimates the mean and variance from our data for each class. It is easy to think about this in the univariate (single input variable) case with two classes.

The mean (mu) value of each input (x) for each class (k) can be estimated in the normal way by dividing the sum of values by the total number of values.

$mu_k = 1/n_k * sum(x)$

Where mu_k is the mean value of x for the class k, n_k is the number of instances with class k. The variance is calculated across all classes as the average squared difference of each value from the mean.

$\sigma^2 = 1 / (n-K) * sum((x - mu)^2)$

Where σ^2 is the variance across all inputs (x), n is the number of instances, K is the number of classes and mu is the mean for input x.

Making Predictions with LDA

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class that gets the highest probability is the output class and a prediction is made.

The model uses Bayes Theorem to estimate the probabilities. Briefly Bayes' Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:

$$P(Y = x|X = x) = (PI_k * f_k(x)) / sum(PI_k * f_k(x))$$

Where PI_k refers to the base probability of each class (k) observed in your training data (e.g. 0.5 for a 50-50 split in a two class problem). In Bayes' Theorem this is called the prior probability.

$$PI_k = n_k/n$$

The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminate function and the class is calculated as having the largest value will be the output classification (y):

$$D_k(x) = x * (mu_k/ \sigma^2) - (mu_k^2/(2 * \sigma^2)) + ln(PI_k)$$

$D_k(x)$ is the discriminate function for class k given input x, the mu_k, σ^2 and PI_k are all estimated from your data. (J.Brownlee)

2.4.3.3 SVM

A support vector machine (SVM) is a supervised learning technique that has proven useful in classification problems encountered in working with microarray data. In the simplest case of two class classification, SVMs find a **hyperplane** that separates the two classes of data with as wide a **margin** as possible. This leads to good generalization accuracy on unseen data and supports specialized optimization methods that allow SVM to learn from a large amount of data.

SVM has a stronger mathematical basis than some machine learning methods such as neural networks and is closely related to some well-established theories in statistics. As a linear model, it not only tries to correctly classify the training data but also maximizes the margin for better generalization performance. This formulation leads to a separating hyperplane that depends only on the (usually small fraction of) data points that lie on the margin, which are called support vectors. Hence the whole algorithm is called support vector machine. In addition, since real-world data analysis problems often involve nonlinear dependencies, SVMs can be easily extended to model such nonlinearity by means of positive semi-definite kernels. Moreover, SVMs can be trained via quadratic programming, which (a) makes theoretical analysis easier and (b) provides much convenience in designing efficient solvers that scale for large datasets. Finally, when applied to real-world data, SVMs often deliver state of- the-art performance in accuracy, flexibility, robustness, and efficiency. (C., 2010)

2.4.3.3.1 Linearly Separable Binary Classification

Theory

We have L training points, where each input x_i has *D* attributes (i.e. is of dimensionality *D*) and is in one of two classes $y_i = -1 \; or \; +1$, i.e our training data is of the form:

$$\{x_i, y_i\} \quad where \quad i = 1 \dots L, \quad y_i \in \{-1,1\}, \quad x \in \mathbb{R}^D \tag{1.0}$$

Here we assume the data is linearly separable, meaning that we can draw a line on a graph of x_1 vs x_2 separating the two classes when $D = 2$ and a hyperplane on graphs of $x_1, x_2 \dots x_D$ for when $D > 2$.

This hyperplane can be described by $\boldsymbol{w} \cdot x + b = 0$ where:

- $\boldsymbol{w}$ is normal to the hyperplane.
- $\frac{b}{\|\boldsymbol{w}\|}$ is the perpendicular distance from the hyperplane to the origin.

Support Vectors are the examples closest to the separating hyperplane and the aim of Support Vector Machines (SVM) is to orientate this hyperplane in such a way as to be as far as possible from the closest members of both classes.

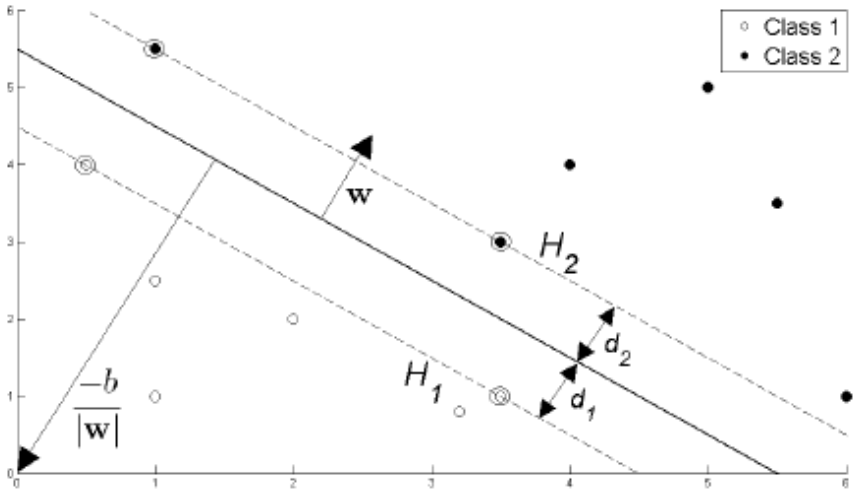


Figure 2-22 **SVM Hyperplane through two linearly separable classes**

Referring to Figure 22, implementing a SVM boils down to selecting the variables $\boldsymbol{w}$ and b so that our training data can be described by:

$$x_\iota \cdot \boldsymbol{w} + b \geq +1 \quad for \quad y_\iota = +1 \tag{1.1}$$

$$x_\iota \cdot \boldsymbol{w} + b \leq -1 \quad for \quad y_\iota = -1 \tag{1.2}$$

These equations can be combined into: $y_\iota(x_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \; \forall_i$ (1.3)

If we now just consider the points that lie closest to the separating hyper plane, i.e. the Support Vectors (shown in circles in the diagram), then the two planes H1 and H2 that these points lie on can be described by:

$$x_\iota \cdot \boldsymbol{w} + b = +1 \quad for \quad H_1 \tag{1.4}$$

$$x_\iota \cdot \boldsymbol{w} + b = -1 \quad for \quad H_2 \tag{1.5}$$

Referring to Figure 21, we define d_1 as being the distance from H_1 to the hyperplane and d_2 from H2 to it. The hyperplane's equidistance from H1 and H2 means that d1 = d2 - a quantity known as the SVM's margin. In order to orientate the hyperplane to be as far from the Support Vectors as possible, we need to maximize this margin.

Simple vector geometry shows that the margin is equal to $\frac{1}{\|\boldsymbol{w}\|}$ and maximizing it subject to the constraint in is equivalent to finding:

$min\|\boldsymbol{w}\|$ such that $y_\iota(x_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \; \forall_i$

Minimizing $\|\boldsymbol{w}\|$ is equivalent to minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2$ and the use of this term makes it possible to perform Quadratic Programming (QP) optimization later on. We therefore need to find:

$$min \; \tfrac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad y_\iota(x_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \; \forall_i \tag{1.6}$$

In order to cater for the constraints in this minimization, we need to allocate them Lagrange multipliers $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha_i} \geq 0 \; \forall_{\boldsymbol{i}}$ **:**

$$L_p \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \boldsymbol{\alpha}[\, y_i(x_i \cdot \boldsymbol{w} + b) - 1 \;\forall_i] \tag{1.7}$$

$$\equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{L} \boldsymbol{\alpha_i}[\, y_i(x_i \cdot \boldsymbol{w} + b) - 1] \tag{1.8}$$

$$\equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{L} \boldsymbol{\alpha_i} y_i(x_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{L} \boldsymbol{\alpha_i} \tag{1.9}$$

We wish to find the $\boldsymbol{w}$ and b which minimizes, and the $\boldsymbol{\alpha}$ which maximizes (whilst keeping $\boldsymbol{\alpha_i} \geq 0\ \forall_{\boldsymbol{i}}$). We can do this by differentiating L_P with respect to $\boldsymbol{w}$ and b and setting the derivatives to zero:

$$\frac{\partial Lp}{\partial w}=0 \rightarrow \quad w = \sum_{i=1}^{L} \boldsymbol{\alpha_i} y_i x_i \tag{1.10}$$

$$\frac{\partial Lp}{\partial b}=0 \rightarrow \sum_{i=1}^{L} \boldsymbol{\alpha_i} y_i x_i = 0 \tag{1.11}$$

Substituting (1.10) and (1.11) into (1.9) gives a new formulation which, being dependent on $\boldsymbol{\alpha}$, we need to maximize:

$$L_D \equiv \sum_{i=1}^{L} \boldsymbol{\alpha_i} - \frac{1}{2}\sum_{i,j} \boldsymbol{\alpha_i}\, \boldsymbol{\alpha_j} y_i y_j x_i \cdot x_j \; s.t. \boldsymbol{\alpha_i} \geq 0\ \forall_{\boldsymbol{i}}, \sum_{i=1}^{L} \boldsymbol{\alpha_i} \boldsymbol{y_i} = 0 \tag{1.12}$$

$$\equiv \sum_{i=1}^{L} \boldsymbol{\alpha_i} - \frac{1}{2}\sum_{i,j} \boldsymbol{\alpha_i} H_{ij} \boldsymbol{\alpha_j} \;\; where\; H_{ij} \equiv y_i y_j x_i \cdot x_j \tag{1.13}$$

$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\boldsymbol{a}^T \mathbf{H} \boldsymbol{a} \;\; s.t. \boldsymbol{\alpha_i} \geq 0\ \forall_{\boldsymbol{i}}, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{1.14}$$

This new formulation L_D is referred to as the **Dual** form of the **Primary** L_p . It is worth noting that the Dual form requires only the dot product of each input vector $x_i$to be calculated, this is important for the Kernel Trick.

Having moved from minimizing L_p to maximizing L_D, we need to find:

$$\max_{\alpha} \left[\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\boldsymbol{a}^T \mathbf{H} \boldsymbol{a}\right] \quad \text{s.t.} \quad \boldsymbol{\alpha_i} \geq 0\ \forall_{\boldsymbol{i}} \quad \text{and} \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{1.15}$$

This is a convex quadratic optimization problem, and we run a QP solver which will return $\boldsymbol{\alpha}$ and from (1.10) will give us $\boldsymbol{w}$. What remains is to calculate b.

Any data point satisfying (1.11) which is a Support Vector x_s will have the form:

$$y_s(x_s \cdot w + b) = 1$$

Substituting in (1.10):

$$y_s\left(\sum_{m \epsilon S} a_m y_m\, x_m \cdot x_s + b\right) = 1$$

Where *S* denotes the set of indices of the Support Vectors. *S* is determined by finding the indices *i* where $\boldsymbol{\alpha_i} > 0$. Multiplying through by y_s and then using $y_s^2 = 1$ from (1.1) and (1.2):

$$y_s^2\left(\sum_{m \epsilon S} a_m y_m\, x_m \cdot x_s + b\right) = y_s$$

$$b = y_s - \sum_{m \epsilon S} a_m y_m\, x_m \cdot x_s$$

Instead of using an arbitrary Support Vector x_s, it is better to take an average over all of the Support Vectors in *S*:

$$b = \frac{1}{N_s}\sum_{s\, \epsilon\, S}(y_s - \sum_{m \epsilon S} a_m y_m\, x_m \cdot x_s) \tag{1.16}$$

We now have the variables w and b that define our separating hyperplane's optimal orientation and hence our Support Vector Machine.

Application

In order to use an SVM to solve a linearly separable, binary classification problem we need to:

- Create **H**, where $H_{ij} \equiv y_i y_j x_i \cdot x_j$.
- Find α so that $\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\boldsymbol{a}^T \mathbf{H} \boldsymbol{a}$ is maximized, subject to the constraints
  $\boldsymbol{a_i} \geq 0 \; \forall_{\boldsymbol{i}}$ and $\sum_{i=1}^{L} \alpha_i y_i = 0$
  This is done using a QP solver.
- Calculate $w = \sum_{i=1}^{L} \boldsymbol{\alpha_i y_i x_i}$.
- Determine the set of Support Vectors *S* by finding the indices such that $\boldsymbol{a_i} > 0$.
- Calculate $b = \frac{1}{N_s} \sum_{s \, \epsilon \, S} (y_s - \sum_{m \epsilon S} a_m y_m \, x_m \cdot x_s)$
- Each new point x' is classified by evaluating $y' = \text{sgn}(\text{w} \cdot x' + \text{b})$.

2.4.3.3.2 Binary Classification for Data that is not Fully Linearly Separable

Theory

In order to extend the SVM methodology to handle data that is not fully linearly separable, we relax the constraints for (1.1) and (1.2) slightly to allow for misclassi_ed points. This is done by introducing a positive slack variable $\xi_i, \; i = 1, \ldots . L$:

$x_\iota \cdot \boldsymbol{w} + b \geq +1 - \xi_i \quad for \quad y_\iota = +1$ (2.1)

$x_\iota \cdot \boldsymbol{w} + b \geq -1 + \xi_i \quad for \quad y_\iota = -1$ (2.2)

$\xi_i \geq 0 \forall i$ (2.3)

Which can be combined into:

$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0$ where $\xi_i \geq 0 \forall i$ (2.4)

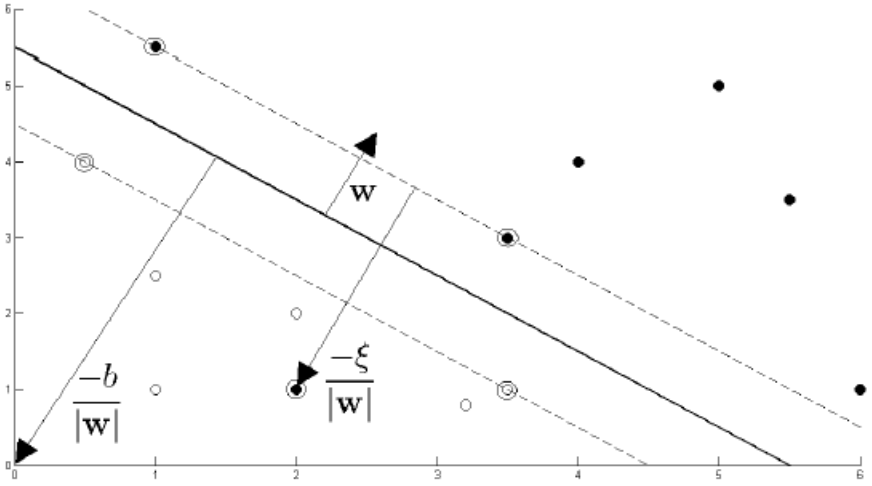


Figure 2-23 **SVM Hyperplane through two non-linearly separable classes**

In this **soft margin** SVM, data points on the incorrect side of the margin boundary have a penalty that increases with the distance from it. As we are trying to reduce the number of misclassifications, a sensible way to adapt our objective function (1.6) from previously, is to find:

$$\min \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{L}\xi_i \quad \text{s.t.} \quad y_i(x_i \cdot w + b) - 1 + \xi_i \ \geq 0 \ \ \forall i \tag{2.5}$$

Where the parameter *C* controls the trade-off between the slack variable penalty and the size of the margin. Reformulating as a Lagrangian, which as before we need to minimize with respect to $\boldsymbol{w}$, *b* and ξ_i and maximize with respect to $\boldsymbol{\alpha}$ (where $a_i \geq 0$, $\mu_i \geq 0 \ \forall_i$):

$$L_p \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 + \boldsymbol{C}\sum_{\boldsymbol{i=1}}^{\boldsymbol{L}}\boldsymbol{\xi_i} - - \sum_{i=1}^{L}\alpha_i[y_i(x_i \cdot w \ + \ b) - 1 + \xi_i] + \sum_{i=1}^{L}\mu_i \ \xi_i \tag{2.6}$$

Differentiating with respect to w, b and ξ_i and setting the derivatives to zero:

$$\frac{\partial Lp}{\partial w}=0 \ \rightarrow \ \ w = \sum_{i=1}^{L}\boldsymbol{\alpha_i} y_i x_i \tag{2.7}$$

$$\frac{\partial Lp}{\partial b}=0 \ \rightarrow \ \sum_{i=1}^{L}\boldsymbol{\alpha_i} y_i=0 \tag{2.8}$$

$$\frac{\partial Lp}{\partial \xi_i}=0 \ \rightarrow C = \alpha_i + \mu_i \tag{2.9}$$

Substituting these in, L_D has the same form as (1.14) before. However (2.9) together with $\mu_i \geq 0 \ \forall_i$, implies that $\boldsymbol{\alpha} \geq C$. We therefore need to _nd:

$$\max_{\alpha}\left[\sum_{i=1}^{L}\alpha_i - \frac{1}{2}\boldsymbol{a}^T\mathbf{H}\boldsymbol{a}\right] \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C \ \forall_i \quad \text{and} \quad \sum_{i=1}^{L}\alpha_i y_i = 0 \tag{2.10}$$

b is then calculated in the same way as in (1.6) before, though in this instance the set of Support Vectors used to calculate *b* is determined by finding the indices *i* where $0 \leq \alpha_i \leq C$.

Application

In order to use an SVM to solve a binary classification for data that is not fully linearly separable we need to:

- Create **H**, where $H_{ij} \equiv y_i y_j x_i \cdot x_j$.
- Choose how significantly misclassifications should be treated, by selecting a suitable value for the parameter *C*.
- Find α so that $\sum_{i=1}^{L}\alpha_i - \frac{1}{2}\boldsymbol{a}^T\mathbf{H}\boldsymbol{a}$ is maximized, subject to the constraints
  $0 \leq \alpha_i \leq C \ \forall_i \quad \text{and} \quad \sum_{i=1}^{L}\alpha_i y_i = 0$
  This is done using a QP solver.
- Calculate $w = \sum_{i=1}^{L}\boldsymbol{\alpha_i y_i x_i}$.
- Determine the set of Support Vectors *S* by finding the indices such that $0 \leq \alpha_i \leq C$.
- Calculate $b = \frac{1}{N_s}\sum_{s \ \epsilon \ S}(y_s - \sum_{m\epsilon S} a_m y_m \, x_m \cdot x_s)$
- Each new point x' is classified by evaluating $y' = \ \text{sgn}(\text{w} \cdot x' \ + \ \text{b})$.

(Fletcher, 2008)

2.4.3.4 KNN

The k-nearest neighbor classifier (kNN) is based on the Euclidean distance between a test sample and the specified training samples. A test sample x is assigned to the class x of its nearest neighbor, where m_i is a nearest neighbor to x if the distance.

$$D(\mathbf{m}i, \mathbf{x}) = \min j \{ D(\mathbf{m}i, \mathbf{x}) \}$$

Where $D(\mathbf{m}i, \mathbf{x}) = ||\mathbf{m}i, \mathbf{x}||$ is the Euclidean distance. The k-nearest neighbors to x are identified and the decision rule is D(**x**→ω) to assign sample **x** to the class ω which is the most popular among the k nearest training samples.

The class of nearest-neighbor methods can be viewed as direct estimates of this conditional expectation, but we have seen that they can fail in at least two ways:

• if the dimension of the input space is high, the nearest neighbors need not be close to the target point, and can result in large errors;

• if special structure is known to exist, this can be used to reduce both the bias and the variance of the estimates.

We anticipate using other classes of models for f(x), in many cases specifically designed to overcome the dimensionality problems, and here we discuss a framework for incorporating them into the prediction problem. Nearest neighbors are useful in many machine learning and data mining tasks, such as classification, anomaly detection and motif discovery and in more general tasks such as spell checking, vector quantization, plagiarism detection, web search, and recommender systems. The naive method to find the nearest neighbor to a point q requires a linear scan of all objects in **i**n a data collection **M**. (Sammut C., 2010) (Hastie) (Leif E. Peterson, 2008)

2.4.3.5 NB

Naıve Bayes is a simple learning algorithm that utilizes Bayes' rule together with a strong assumption that the attributes are conditionally independent given the class. While this independence assumption is often violated in practice, naıve Bayes nonetheless often delivers competitive classification accuracy. Coupled with its computational efficiency and many other desirable features, this leads to naıve Bayes being widely applied in practice.

Naıve Bayes provides a mechanism for using the information in sample data to estimate the posterior probability $P(y|\mathbf{x})$ of each class *y* given an object **x**. Once we have such estimates, we can use them for classification or other decision support applications.

Naıve Bayes' features include the following:

- *Computational efficiency*: training time is linear with respect to both the number of training examples and the number of attributes, and classification time is linear with respect to the number of attributes and unaffected by the number of training examples.
- *Low variance*: because naıve Bayes does not directly fit the posterior distribution, it has low variance, albeit at the cost of high bias.
- *Incremental learning*: naıve Bayes operates from estimates of low-order probabilities that are derived from the training data. These can readily be updated as new training data are acquired.
- *Direct prediction of posterior probabilities*.
- *Robustness in the face of noise*: naıve Bayes always uses all attributes for all predictions and hence is relatively insensitive to noise in the examples to be classified. Because it uses probabilities, it is also relatively insensitive to noise in the training data.
- *Robustness in the face of missing values*: because naıve Bayes always uses all attributes for all predictions, if one attribute value is missing, information from other attributes is still used, resulting in graceful degradation in performance. It is also relatively insensitive to missing attribute values in the training data due to its probabilistic framework.

Structure of Learning System

Naıve Bayes is a form of Bayesian network classifier based on Bayes' rule:

$$P(y|\mathbf{x}) = P(y)P(\mathbf{x}|y)P(\mathbf{x}) \quad (1)$$

together with an assumption that the attributes are conditionally independent given the class. For attribute-value data, this assumption entitles

$$P(y|\mathbf{x}) = \prod_{i=1}^{n} P(xi|y) \quad (2)$$

where xi is the value of the i th attribute in x and n is the number of attributes:

$$P(\mathbf{x}) = \prod_{i=1}^{k} P(ci)P(\mathbf{x}|ci) \quad (3)$$

where *k* is the number of classes and c_i is the *i*th class. Thus, (1) can be calculated by normalizing the numerators of the right-hand side of the equation. The resulting classifier uses a linear model, equivalent to that used by logistic regression, differing only in the manner in which the parameters are chosen.

For categorical attributes, the required probabilities $P(y)$ and $P(xi|y)$ are normally derived from frequency counts stored in arrays whose values are calculated by a single pass through the training data

at training time. These arrays can be updated as new data are acquired, supporting incremental learning. Probability estimates are usually derived from the frequency counts using smoothing functions such as the Laplace estimate or an m-estimate.

(Sammut C., 2010)

2.4.3.6 CART

The induction of decision trees is one of the oldest and most popular techniques for learning discriminatory models, which has been developed independently in the statistical (Breiman et al. 1984; Kass 1980) and machine learning (Hunt et al. 1966; Quinlan 1983, 1986) communities. A decision tree is a tree-structured classification model, which is easy to understand, even by non expert users, and can be efficiently induced from data.

A decision tree is a largely used non-parametric effective machine learning modeling technique for regression and classification problems. To find solutions a decision tree makes sequential, hierarchical decision about the outcomes variable based on the predictor data.

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

The Understanding Level of Decision Tree algorithm is so easy as compared to classification algorithm.

In Decision tree algorithm we solve our problem in tree representation. Each internal node of the tree corresponds to an attributes. Each leaf node corresponds to a Class Label.

In decision tree for predicting a class label for a record we start from the root of the tree. We compare the value of the root attribute with record's attribute on the basis of comparison. We follow the branch corresponding to that values & jump to the next node. We continue comparing our record's attribute value with other internal nodes of the tree until we reach a leaf node.

Learning Algorithm

Decision trees are learned in a top-down fashion, with an algorithm known as **top-down induction of decision trees** (TDIDT), **recursive partitioning**, or **divide-and-conquer** learning. The algorithm selects the best attribute for the root of the tree, splits the set of examples into disjoint sets, and adds corresponding nodes and branches to the tree. The simplest splitting criterion is for discrete attributes, where each test has the form $t \leftarrow (A = u)$where *u* is one possible value of the chosen attribute *A.* The corresponding set S_t contains all training examples for which the attribute *A* has the value *t*. This can be easily adapted to numerical attributes, where one typically uses binary splits of the form $t \leftarrow (A = u_t)$, which indicate whether the attribute's value is above or below a certain threshold value u_t. Alternatively, one can transform the data beforehand using a discretization algorithm.

After splitting the dataset according to the selected attribute, the procedure is recursively applied to each of the resulting datasets. If a set contains only examples from the same class, or if no further splitting is possible (e.g., because all possible splits have already been exhausted or all remaining splits will have the same outcome for all examples), the corresponding node is turned into a leaf node and labeled with the respective class. For all other sets, an interior node is added and associated with the

best splitting attribute for the corresponding set as described above. Hence, the dataset is successively partitioned into non overlapping, smaller datasets until each set only contains examples of the same class (a so-called pure node). Eventually, a pure node can always be found via successive partitions unless the training data contains two identical but contradictory examples, i.e., examples with the same feature values but different class values. (Medium.com) (C., 2010)

CART Algorithm.

function TDIDT(*S*)
Input: S, a set of labeled examples.
Tree = new empty node
if all examples have the same class c
or no further splitting is possible
then // new leaf
LABEL(*Tree*) = *c*
else // new decision node
(*A, T*) = FINDBESTSPLIT(*S*)
for each test $t \in T$ **do**
S_t = all examples that satisfy *t*
$Node_t$ = TDIDT(S_t)
ADDEDGE(Tree $\xrightarrow{t} Node_t$)
endfor
endif
return *Tree*

2.4.4 Classification metrics

In practice, a binary classifier […] can make two types of errors: it can incorrectly assign an individual who defaults to the no default category, or it can incorrectly assign an individual who does not default to the default category. It is often of interest to determine which of these two types of errors are being made. A confusion matrix […] is a convenient way to display this information. (Gareth James, 2013)

- A **false negative** is an example of positive class that has been incorrectly classified as negative.
- A **false positive** is an example of a negative class that has been incorrectly classified as positive.
- **True negatives** are the negative examples that are correctly classified by a classification model.
- **True positives** are the positive examples that are correctly classified by a classification model.

2.4.4.1 Confusion Matrix

A confusion matrix summarizes the classification performance of a classifier with respect to some test data. It is a two-dimensional matrix, indexed in one dimension by the true class of an object and in the other by the class that the classifier assigns.

A special case of the confusion matrix is often utilized with two classes; one designated the positive class and the other the negative class. In this context, the four cells of the matrix are designated as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

(Sammut C., 2010)

	Predicted Positive	**Predicted Negative**		
Actual Positive (True)	True Positive (TP)	False Negative (FN)	Sensitivity or Recall TP / (TP + FN)	
Actual Negative (False)	False Positive (FP)	True Negative (TN)	Specificity TN / (TN + FP)	
	Precision TP/(TP+FP)	Negative Predictive value TN/(TN+FN)	$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$	$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$

Table 1-Confusion Matrix

2.4.4.2 Sensitivity and Specificity

Sensitivity and specificity are two measures used together in some domains to measure the predictive performance of a classification model or a diagnostic test. For example, to measure the effectiveness of a diagnostic test in the medical domain, sensitivity measures the fraction of people with disease (i.e., positive examples) who have a positive test result; and specificity measures the fraction of people without disease (i.e., negative examples) who have a negative test result. They are defined with

reference to a special case of the confusion matrix, with two classes; one designated the positive class and the other the negative class, as indicated in Table 1.

Sensitivity is equivalent to **recall**, sometimes also is called true **positive rate**.

Specificity is sometimes also called true **negative rate**.

They are defined as follows:

$$Sensitivity = TP / (TP + FN)$$

$$Specificity = TN / (TN + FP)$$

Instead of two measures, they are sometimes combined to provide a single measure of predictive performance as follows:

$$Sensitivity\ x\ Specificity = TP * TN / [\ (TP + FN) * (TN + FP)\ ]$$

(C., 2010)

2.4.4.3 Precision

Precision is defined as the ratio of true positives (TP) and the total number of positives predicted by a model. This is defined with reference to a special case of the confusion matrix, with two classes: one designated the positive class and the other the negative class, as indicated in Table 1. Precision can then be defined in terms of true positives and false positives (FP) as follows.

$$Precision = TP/(TP + FP)$$

(C., 2010)

2.4.4.4 F_1-Score

It is often possible to construct baseline models that maximize one metric but not the other. For example, a model that declares every record to be the positive class will have a perfect recall, but very poor precision. Conversely, a model that assigns a positive class to every test record that matches one of the positive records in the training set has very high precision, but low recall. Building a model that maximizes both precision and recall is the key challenge of classification algorithms. Precision and recall can be summarized into another metric known as the F1 measure.

$$F_1 = \frac{2\,recall\,precision}{recall+precision} = \frac{2 \times TP}{2 \times TP+FP+FN}$$

In principle, F_1 represents a harmonic mean between recall and precision, i.e.

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}}$$

The harmonic mean of two numbers z and gr tends to be closer to the smaller of the two numbers. Hence, a high value of F1-measure ensures that both precision and recall are reasonably high. (Pang Ning Tan, 2006)

2.4.4.5 Accuracy

Accuracy refers to a measure of the degree to which the predictions of a model matches the reality being modeled. Accuracy is also used as a statistical measure of how well a binary classification test correctly identifies or excludes a condition. That is, the accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. The formula for quantifying binary accuracy is:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

(wikipedia.org) (C., 2010)

2.4.4.6 ROC Curve

During the past four decades, ROC analysis has become a popular method for evaluating the accuracy of medical diagnostic systems. The most desirable property of ROC analysis is that the accuracy indices derived from this technique are not distorted by fluctuations caused by the use of arbitrarily chosen decision criteria or cut-offs. In other words, the indices of accuracy are not influenced by the decision criterion (i.e. the tendency of a reader or observer to choose a specific threshold on the separator variable) and/or to consider the prior probability of the "signal". The derived summary measure of accuracy, such as the area under the curve (AUC) determines the inherent ability of the test to discriminate between the diseased and healthy populations. Using this as a measure of a diagnostic performance, one can compare individual tests or judge whether the various combination of tests (e.g. combination of imaging techniques or combination of readers) can improve diagnostic accuracy.

ROC analysis is used in clinical epidemiology to quantify how accurately medical diagnostic tests (or systems) can discriminate between two patient states, typically referred to as "diseased" and "non-diseased". An ROC curve is based on the notion of a "separator" scale, on which results for the diseased and non-diseased form a pair of overlapping distributions. The complete separation of the two underlying distributions implies a perfectly discriminating test while complete overlap implies no discrimination.

The receiver operating characteristics (ROC) curve is a two-dimensional graph in which the TPR represents the y-axis and FPR is the x-axis. The ROC curve has been used to evaluate many systems such as diagnostic systems, medical decision-making systems, and machine learning systems. It is used to make a balance between the benefits, i.e., true positives, and costs, i.e., false positives. Any classifier that has discrete outputs such as decision trees is designed to produce only a class decision, i.e., a decision for each testing sample, and hence it generates only one confusion matrix which in turn corresponds to one point into the ROC space. However, there are many methods that were introduced for generating full ROC curve from a classifier instead of only a single point such as using class proportions or using some combinations of scoring and voting. On the other hand, in continuous output classifiers such as the Naive Bayes classifier, the output is represented by a numeric value, i.e., score, which represents the degree to which a sample belongs to a specific class. The ROC curve is generated by changing the threshold on the confidence score; hence, each threshold generates only one point in the ROC curve.

Figure 22 shows an example of the ROC curve. As shown, there are four important points in the ROC curve. The point A, in the lower left corner (0,0) represents a classifier where there is no positive classification, while all negative samples are correctly classified and hence TPR=0 and FPR=0 The point C, in the top right corner (1,1), represents a classifier where all positive samples are correctly classified, while the negative samples are misclassified. The point D in the lower right corner (1,0) represents a classifier where all positive and negative samples are misclassified. The point B in the upper left corner (0,1) represents a classifier where all positive and negative samples are correctly classified; thus, this point represents the perfect classification or the Ideal operating point. Figure 22 shows the perfect classification performance. It is the green curve which rises vertically from (0,0) to (0,1) and then horizontally to (1,1). This curve reflects that the classifier perfectly ranked the positive samples relative to the negative samples. A point in the ROC space is better than all other points that are in the southeast, i.e., the points that have lower TPR, higher FPR, or both. Therefore, any classifier appears in the lower right triangle performs worse than the classifier appears in the upper left triangle. A point in the ROC space is better than all other points that are in the southeast, i.e., the points that have lower TPR, higher FPR, or botղ. Therefore, any classifier appears in the lower right triangle performs worse than the classifier appears in the upper left triangle. (Tharwat, 2018) (INDRAYAN, 2011) (Hajian-Tilaki, 2013)

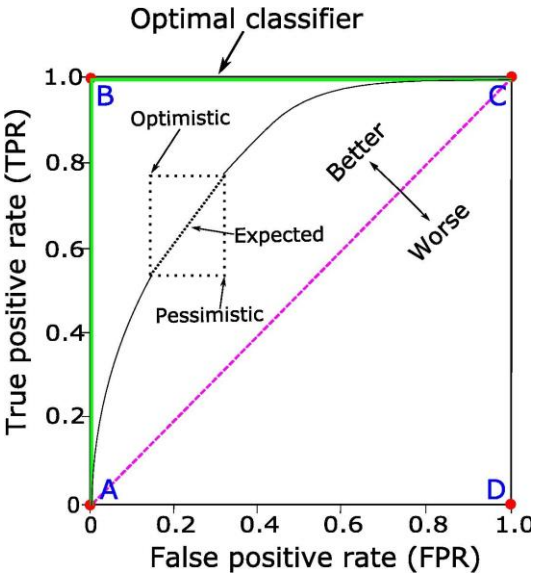


Figure 2-24 A basic ROC curve showing important points, and the optimistic, pessimistic and expected ROC segments for equally scored samples. (Tharwat, 2018)

The AUC Statistic

The most important statistic associated with ROC curves is the *area under (ROC) curve* or *AUC*. Since the curve is located in the unit square, we have $0 \leq AUC \leq 1$. $AUC=1$ is achieved if the classifier scores every positive higher than every negative; $AUC= 0$ is achieved if every negative is scored higher than every positive. $AUC=1/2$ is obtained in a range of different scenarios, including: (i) the classifier assigns the same score to all test examples, whether positive or negative, and thus the ROC curve is the ascending diagonal; (ii) the per-class score distributions are similar, which results in an ROC curve close (but not identical) to the ascending diagonal; and (iii) the classifier gives half of a particular class the highest scores and the other half the lowest scores. Notice that, although a classifier with *AUC* close to one half is often said to perform randomly, there is nothing random in the third classifier: rather, its excellent performance on some of the examples is counter balanced by its very poor performance on some others (Sometimes a linear rescaling $2 \cdot AUC - 1$ called the *Gini coefficient* is preferred, which has a related use in the assessment of income or wealth distributions using Lorenz curves: a Gini coefficient close to 0 means that income is approximately evenly distributed. Notice that this Gini coefficient is often called the Gini index, but should not be confused with the impurity measure used in decision tree learning). *AUC* has a very useful statistical interpretation: it is the expectation that a (uniformly) randomly drawn

positive receives a higher score than a randomly drawn negative. It is a normalized version of the *Wilcoxon-Mann-Whitney sum of ranks test*, which tests the null hypothesis that two samples of ordinal measurements are drawn from a single distribution. The "sum of ranks" epithet refers to one method to compute this statistic, which is to assign each test example an integer rank according to decreasing score (the highest scoring example gets rank 1, the next gets rank 2, etc.); sum up the ranks of the n^- negatives, which we want to be high; and subtract $\sum_{i=1}^{n^-} i = \frac{n^-(n^-+1)}{2}$ to achieve 0 if all negatives are ranked first. The *AUC* statistic is then obtained by normalizing by the number of pairs of one positive and one negative, n^+n^-. There are several other ways to calculate *AUC*, for instance, we can calculate, for each negative, how many positives precede it, which basically is a column wise calculation and yields an alternative view of *AUC* as the expected true positive rate if the operating point is chosen just before a randomly drawn negative. (C., 2010)

2.4.5 Learning Procedure

2.4.5.1 Algorithm Evaluation

Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting[1]. To avoid it, it is common practice when performing a supervised machine learning experiment to hold out part of the available data as a **test set**. Note that the word "experiment" is not intended to denote academic use only, because even in commercial settings machine learning usually starts out experimentally. Here is a flowchart of typical cross validation workflow in model training.

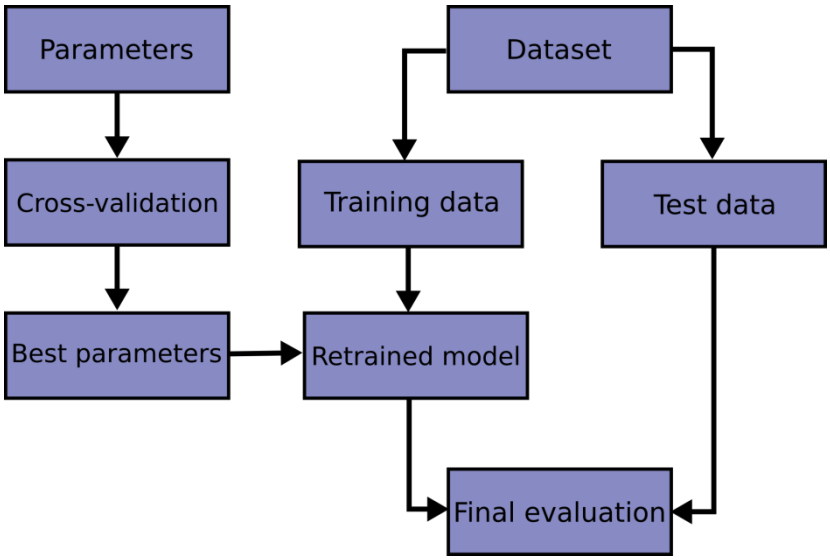


Figure 2-25 Cross Validation workflow in model training flowchart

When evaluating different settings ("hyperparameters") for estimators, such as the C setting that must be manually set for an SVM, there is still a risk of overfitting[1] on the test set because the parameters can be tweaked until the estimator performs optimally. This way, knowledge about the test set can "leak" into the model and evaluation metrics no longer report on generalization performance. To solve this problem, yet another part of the dataset can be held out as a so-called "validation set": training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set.

However, by partitioning the available data into three sets, we drastically reduce the number of samples which can be used for learning the model, and the results can depend on a particular random choice for the pair of (train, validation) sets.

A solution to this problem is a procedure called cross-validation (CV for short). A test set should still be held out for final evaluation, but the validation set is no longer needed when doing CV. In the basic approach, called *k*-fold CV, the training set is split into *k* smaller sets (other approaches are described below, but generally follow the same principles). The following procedure is followed for each of the *k* "folds". (Scikit-learn: Machine Learning in Python, 2011) (API design for machine learning software: experiences from the scikit-learn project, 2013)

2.4.5.1.1 Hold Out Evaluation Dataset

Holdout evaluation is an approach to out-of-sample evaluation whereby the available data are partitioned into a training set and a test set. The test set is thus out-of-sample data and is sometimes called the **holdout set** or **holdout data**. The purpose of holdout evaluation is to test a model on different data to that from which it is learned. This provides less biased estimate of learning performance than in-sample evaluation. In **repeated holdout evaluation**, repeated holdout evaluation experiments are performed, each time with a different partition of the data, to create a distribution of training and test sets with which an algorithm is assessed. (C., 2010)

[1] A model *overfits* the training data when it describes features that arise from noise or variance in the data, rather than the underlying distribution from which the data were drawn. Overfitting usually leads to loss of accuracy on out-of-sample data.

2.4.5.1.2 K-Fold Cross Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called ***k*** that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds. (An Introduction to Statistical Learning, 2013)

It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of **hyperparameters**. A failure to perform these operations within the loop may result in data leakage and an optimistic estimate of the model skill.

Despite the best efforts of statistical methodologists, users frequently invalidate their results by inadvertently peeking at the test data. (Artificial Intelligence: A Modern Approach (3rd Edition), 2009.)

The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error.

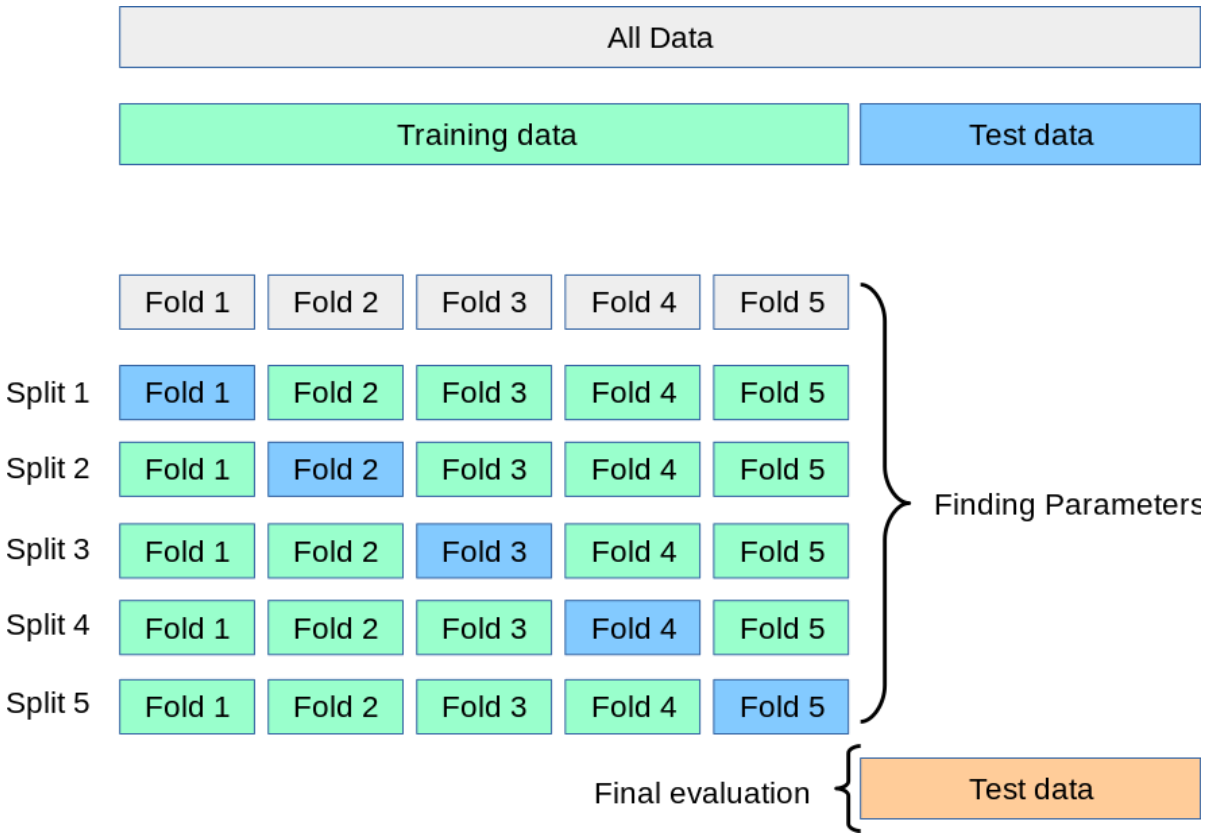


Figure 2-26 **5-Fold Cross Validation**

Configuration of k

The k value must be chosen carefully for data sample.

A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).

Three common tactics for choosing a value for k are as follows:

- **Representative**: The value for k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.
- **k=10**: The value for k is fixed to 10, a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance.
- **k=n**: The value for k is fixed to n, where n is the size of the dataset to give each test sample an opportunity to be used in the hold out dataset. This approach is called leave-one-out cross-validation.

The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller (Applied Predictive Modeling, 2013)

A value of k=10 is very common in the field of applied machine learning, and is recommend if you are struggling to choose a value for your dataset.To summarize, there is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. (An Introduction to Statistical Learning, 2013)If a value for k is chosen that does not evenly split the data sample, then one group will contain a remainder of the examples. It is preferable to split the data sample into k groups with the same number of samples, such that the sample of model skill scores are all equivalent.

(Jason, 2018)

2.4.5.2 Data Transformation

It is frequently necessary to transform data from one representation to another. There are many reasons for changing representations:

- **To generate symmetric distributions instead of the original skewed distributions**.
- **Transformation improves visualization** of data that might be tightly clustered relative to a few outliers.
- Data are transformed to achieve **better interpretability.**
- Transformations are often used to **improve the compatibility of the data with assumptions underlying a modeling process**, for example, to linearize (straighten) the relation between two variables whose relationship is nonlinear. Some of the data mining algorithms require the relationship between data to be linear.

Different types of transformation will be referred whereby each data point $\boldsymbol{x_i}$ is replaced with a **transformed value** $\boldsymbol{y_i = f(x_i)}$, where $\boldsymbol{f}$ f is the transformation function. Many techniques are applied for data transformation. Each technique has its own purpose and dependency on the nature of data. Some of the major transformations are discussed below. (C., 2010)

2.4.5.2.1 Normalization

Min-max normalization projects the original range of data onto a new range. Very common normalization intervals are [0, 1] and [-1, 1]. This normalization method is very useful when we apply a machine learning or data mining approach that utilizes distance. For example, in k-nearest neighbor methods, using un-normalized values might cause attributes whose values have greater magnitudes to dominate over other attributes. Therefore, normalization aims to standardize magnitudes across variables. A useful application for min-max scaling is image processing where pixel intensities have to be normalized to fit within a certain range (i.e., 0–255 for the RGB color range). Also, typical neural network algorithms (ANN) require data that is on a 0–1 scale. Normalization provides the same range of values for each of the inputs to the model.

2.4.5.2.2 Standardization

Z-score normalization (also referred to as standardization) is a normalization method that transforms not only the data magnitude but also the dispersion. Some data mining methods are based on the assumption that data follow a certain distribution. For example, methods such as logistic regression, SVM, and neural network when using gradient descent/ascent optimization methods assume data follow a Gaussian distribution. Otherwise, the approaches will be ill conditioned and might not guarantee a stable convergence of weight and biases. Other approaches such as linear discriminant analysis (LDA), principal component analysis (PCA), and kernel principal component analysis require features to be on the same scale to find directions that maximize the variance (under the constraints that those directions/eigenvectors/principal components are orthogonal). Z-score normalization overcomes the problem of variables with different units as it transforms variables so that they are centered on 0 with a standard deviation of 1.

2.4.5.3 Dimensionality Reduction

Every data object in a computer is represented and stored as a set of features, for example, color, price, dimensions, and so on. Instead of the term *features*, one can interchangeably use the term **dimensions** because an object with n features can also be represented as a multidimensional point in an n-dimensional space. Therefore, dimensionality reduction (dR) refers to the process of mapping an n-dimensional point into a lower k-dimensional space. This operation reduces the size for representing and storing an object or a dataset in general; hence, dimensionality reduction can be seen as a method for data compression. In addition, this process promotes data visualization, particularly when objects are mapped onto two or three dimensions. Finally, in the context of classification, dimensionality reduction can be a useful tool for (a) making tractable classification schemes that are superlinear with respect to dimensionality tractable, (b) reducing the variance of classifiers that are plagued by large variance in higher dimensionalities, and (c) removing the noise that may be present, thus boosting classification accuracy.

Genomic Microarray Data is usually short and fat data – high dimensionality with a small sample size, which poses a great challenge for computational techniques. Their dimensionality can be up to tens of thousands of genes, while their sample sizes can only be several hundreds. Furthermore, additional experimental complications like noise and variability render the analysis of microarray data an exciting domain. Because of these issues, various feature selection algorithms are adopted to reduce the dimensionality and remove noise in microarray data analysis.

There are many techniques for dimensionality reduction. The objective of these techniques is to appropriately select the k dimensions (and also the number k) so that the important characteristics of the original object are retained. For example, when performing dimensionality reduction on an image, e.g., using a wavelet based technique, the desirable outcome is that the difference between the original and the final images is almost imperceptible. When performing dimensionality reduction not on a single object, but on a dataset, an additional requirement is that the relationship between the objects in the original space be preserved. This is particularly important for reasons of classification and visualization in the new space.

Two important categories of dimensionality reduction techniques exist:

• **Feature selection** techniques, in which only the most important or descriptive features/dimensions are retained, and the rest are discarded. More details on such techniques can be found under the entry Feature Selection

• **Feature projection** methodologies, which project the existing features onto different dimensions or axes. The aim here is, again, to find those new data axes that retain the dataset structure and preserve its variance as closely as possible

2.4.5.3.1 Feature Selection

Feature selection, as a dimensionality reduction technique, aims to choose a small subset of the relevant features from the original ones by removing irrelevant, redundant, or noisy features. Feature selection usually leads to better learning performance, i.e., higher learning accuracy, lower computational cost, and better model interpretability. Generally speaking, irrelevant features are features that cannot help discriminate samples from different classes (supervised) or clusters (unsupervised). Removing irrelevant features will not affect learning performance. In fact, the removal of irrelevant features may help learn a better model, as irrelevant features may confuse the learning system and cause memory and computation inefficiency.

A redundant feature is a feature that implies the copresence of another feature. Individually, each redundant feature is relevant, but removal of one of them will not affect the learning performance. A noisy feature is a type of relevant feature. However, due to the noise introduced during the data collection process or because of the nature of this feature, a noisy feature may not be so relevant to the learning or mining task. It can discriminate a part of the points from the two classes and may confuse the learning model for the overlapping points (Noisy features are very subtle. One feature may be a noisy feature itself. However, in some cases, when two or more noisy features can complement each other to distinguish samples from different classes, they may be selected together to benefit the learning model.)

In many real-world applications, such as data mining, machine learning, computer vision, and bioinformatics, we need to deal with high dimensional data. In the past 30 years, the dimensionality of the data involved in these areas has increased explosively. The huge number of high-dimensional data has presented serious challenges to existing learning methods. First, due to the large number of features and relatively small number of training samples, a learning model tends to **overfit**, and their learning performance degenerates. Data with high dimensionality not only degenerates many algorithms' performance due to the **curse of dimensionality** and the existence of irrelevant, redundant, and noisy dimensions, it also significantly increases the time and memory requirement of the algorithms. Second, storing and processing such amounts of high-dimensional data become a challenge. Dimensionality reduction is one of the most popular techniques to reduce dimensionality and can be categorized into feature extraction and feature selection. Both feature extraction and feature selection are capable of improving performance, lowering computational complexity, building better generalization models, and decreasing required storage. Feature extraction maps the original feature space to a new feature space with lower dimensionality by combining the original feature space. Therefore, further analysis of new features is problematic since there is no physical meaning for the transformed features obtained from

feature extraction. In contrast, feature selection selects a subset of features from the original feature set. Therefore, feature selection keeps the actual meaning of each selected feature, which makes it superior in terms of feature readability and interpretability.

Structure of the Learning System

From the perspective of label availability, feature selection methods can be broadly classified into supervised, unsupervised, and semi-supervised methods. In terms of different selection strategies, feature selection can be categorized as filter, wrapper, and embedded models.

- **Supervised feature selection** is usually used for classification tasks. The availability of the class labels allows supervised feature selection algorithms to effectively select discriminative features to distinguish samples from different classes. A general framework of supervised feature selection is shown in Figure 25. Features are first generated from training data. Instead of using all the data to train the supervised learning model, supervised feature selection will first select a subset of features and then process the data with the selected features to the learning model. The feature selection phase will use the label information and the characteristics of the data, such as information gain or Gini index, to select relevant features. The final selected features, as well as with the label information, are used to train a classifier, which can be used for prediction.
- **Unsupervised feature selection** is usually used for clustering tasks. A general framework of unsupervised feature selection is very similar to supervised feature selection, except that there's no label information involved in the feature selection phase and the model learning phase. Without label information to define feature relevance, unsupervised feature selection relies on another alternative criterion during the feature selection phase. One commonly used criterion chooses features that can best preserve the manifold structure of the original data. Another frequently used method is to seek cluster indicators through clustering algorithms and then transform the unsupervised feature selection into a supervised framework. There are two different ways to use this method. One way is to seek cluster indicators and simultaneously perform the supervised feature selection within one unified framework. The other way is to first seek cluster indicators, then to perform feature selection to remove or select certain features, and finally to repeat these two steps iteratively until certain criterion is met. In addition, certain supervised feature selection criterion can still be used with some modification.
- **Semi-supervised feature selection** is usually used when a small portion of the data is labeled. When such data is given to perform feature selection, both supervised and unsupervised feature selection might not be the best choice. Supervised feature selection might not be able to select relevant features because the labeled data is insufficient to represent the distribution of the features. Unsupervised feature selection will not use the label information, while label information can give some discriminative information to select relevant features. Semi-supervised feature selection, which takes advantage of both labeled data and unlabeled data, is a better choice to handle partially labeled data. The general framework of semi-supervised feature selection is the same as that of supervised feature selection, except that data is partially labeled. Most of the existing semi-supervised feature selection algorithms rely on the construction of the similarity matrix and select features that best fit the similarity matrix. Both

the label information and the similarity measure of the labeled and unlabeled data are used to construct the similarity matrix so that label information can provide discriminative information to select relevant features, while unlabeled data provide complementary information.

- **Filter Models** For filter models, features are selected based on the characteristics of the data without utilizing learning algorithms. This approach is very efficient. However, it doesn't consider the bias and heuristics of the learning algorithms. Thus, it may miss features that are relevant for the target learning algorithm. A filter algorithm usually consists of two steps. In the first step, features are ranked based on certain criterion. In the second step, features with the highest rankings are chosen. A lot of ranking criteria, which measures different characteristics of the features, are proposed: the ability to effectively separate samples from different classes by considering between class variance and within class variance, the dependence between the feature and the class label, the correlation between feature class and feature-feature, the ability to preserve the manifold structure, the mutual information between the features, and so on.
- **Wrapper Models** The major disadvantage of the filter approach is that it totally ignores the effects of the selected feature subset on the performance of the clustering or classification algorithm. The optimal feature subset should depend on the specific biases and heuristics of the learning algorithms. Based on this assumption, wrapper models use a specific learning algorithm to evaluate the quality of the selected features. The feature search component will produce a set of features based on certain search strategies. The feature evaluation component will then use the predefined learning algorithm to evaluate the performance, which will be returned to the feature search component for the next iteration of feature subset selection. The feature set with the best performance will be chosen as the final set. The search space for m features is $O(2^m)$. To avoid exhaustive search, a wide range of search strategies can be used, including hill-climbing, best-first, branch-and-bound, and genetic algorithms.
- **Embedded Models** Filter models are computationally efficient, but totally ignore the biases of the learning algorithm. Compared with filter models, wrapper models obtain better predictive accuracy estimates, since they take into account the biases of the learning algorithms. However, wrapper models are very computationally expensive. Embedded models are a tradeoff between the two models by embedding the feature selection into the model construction. Thus, embedded models take advantage of both filter models and wrapper models: (1) they are far less computationally intensive than wrapper methods, since they don't need to run the learning models many times to evaluate the features, and (2) they include the interaction with the learning model. The biggest difference between wrapper models and embedded models is that wrapper models first train learning models using the candidate features and then perform feature selection by evaluating features using the learning model, while embedded models select features during the process of model construction to perform feature selection without further evaluation of the features. (C., 2010)

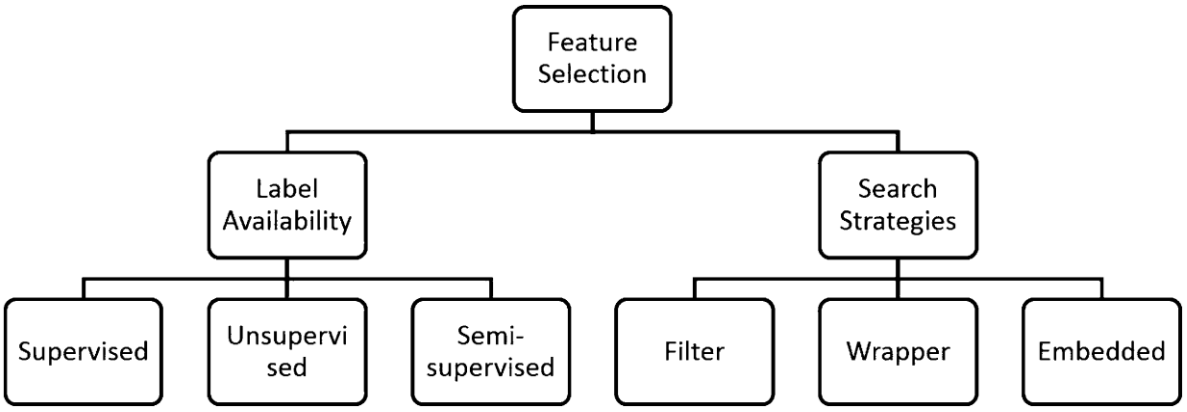


Figure 2-28 **Feature selection categories**

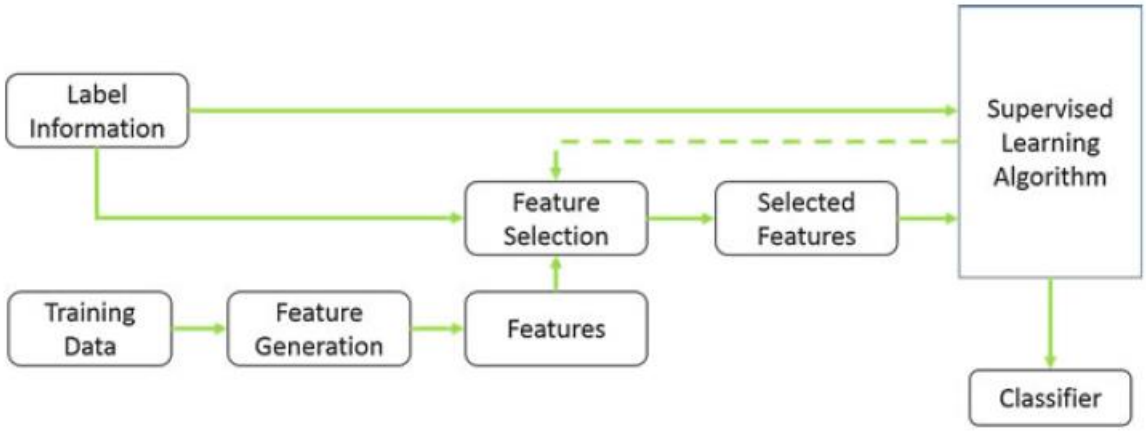


Figure 2-27 **Feature Selection.** A general framework of supervised feature selection.

2.4.5.3.2 Feature Projection

Feature projection techniques typically exploit the correlations between the various data dimensions, with the goal of creating dimensions/axes that are uncorrelated and sufficiently describe the data. One of the most popular dimensionality reduction techniques is *principal component analysis* or PCA. It attempts to discover those axes (or components) onto which the data can be projected while maintaining the original correlation between the dimensions.

PCA uses the Euclidean distance as the measure of dissimilarity among objects. The first principal component (or axis) indicates the direction of maximum variance in the original dimensions. The second component shows the direction of the next highest variance

2.4.5.3.2.1 PCA

PCA is defined as an orthogonal linear transformation with the property that it transforms the data into a new coordinate system, such that the projection of the data on the first coordinate has the greatest variance among all projections on a line, the projection of the data on the second coordinate has the second greatest variance, and so on. Let *X* denote the data matrix, with each point written as a column vector in *X*, and modified so that *X* has empirical mean zero (i.e., the mean vector is subtracted from each data point). Then the eigenvectors of the matrix *XX T* are the coordinates of the new system. To reduce the dimensionality, keep only the eigenvectors corresponding to the largest few eigenvalues. Principal components analysis (PCA) produces a low-dimensional representation of a data set. It finds a sequence of linear combinations of the variables that have maximal variance and are mutually uncorrelated. Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization. (Diego Galar, 2017) (C., 2010)

Sort Definitions for PCA steps

2.4.5.3.2.1.1 Variance

It is a measure of the variability or it simply measures how spread the data set is. Mathematically, it is the average squared deviation from the mean score. The following formula is used to compute variance $var(x)$.

$$var(x) = \frac{\sum(x_i - \bar{x})^2}{N}$$

2.4.5.3.2.1.2 Covariance

Covariance: It is a measure of the extent to which corresponding elements from two sets of ordered data move in the same direction. Formula is shown denoted by $cov(x, y)$ as the covariance of *x* and *y*.

$$cov(x, y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N}$$

Here, x_i is the value of x in i_{th} dimension. x bar and y bar denote the corresponding mean values.

One way to observe the covariance is how interrelated two data sets are.

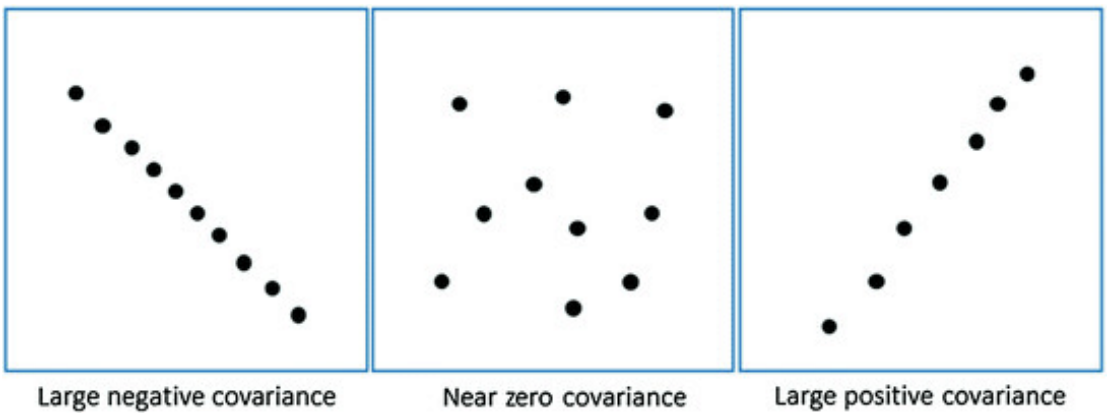


Figure 2-29 **Covariance**

Positive covariance means X and Y are positively related i.e. as X increases Y also increases. Negative covariance depicts the exact opposite relation. However zero covariance means X and Y are not related.

Since we try to find the patterns among the data sets so we want the data to be spread out across each dimension. Also, we want the dimensions to be independent. Such that if data has high covariance when represented in some *n* number of dimensions then we replace those dimensions with **linear combination** of those n dimensions. Now that data will only be dependent on linear combination of those related n dimensions. *(related = have high covariance)*

PCA finds a new set of dimensions (or a set of basis of views) such that all the dimensions are orthogonal (and hence linearly independent) and ranked according to the variance of data along them. It means more important principle axis occurs first. *(more important = more variance/more spread out data)*

How does PCA work:

1. Calculate the covariance matrix *X* of data points.
2. Calculate eigen vectors and corresponding eigen values.
3. Sort the eigen vectors according to their eigen values in decreasing order.
4. Choose first k eigen vectors and that will be the new k dimensions.
5. Transform the original n dimensional data points into k dimensions.

2.4.5.3.2.1.3 Eigenvalues-Eigenvectors

Eigenvalues/vectors are instrumental to understanding electrical circuits, mechanical systems, ecology and even Google's PageRank algorithm. To begin, let $\boldsymbol{v}$ be a vector (shown as a point) and $\boldsymbol{A}$ be a matrix with columns $\boldsymbol{\alpha_1}$ and $\boldsymbol{\alpha_2}$ (shown as arrows). If we multiply $\boldsymbol{v}$ by $\boldsymbol{A}$, then $\boldsymbol{A}$ sends $\boldsymbol{v}$ to a new vector $\boldsymbol{Av}$.

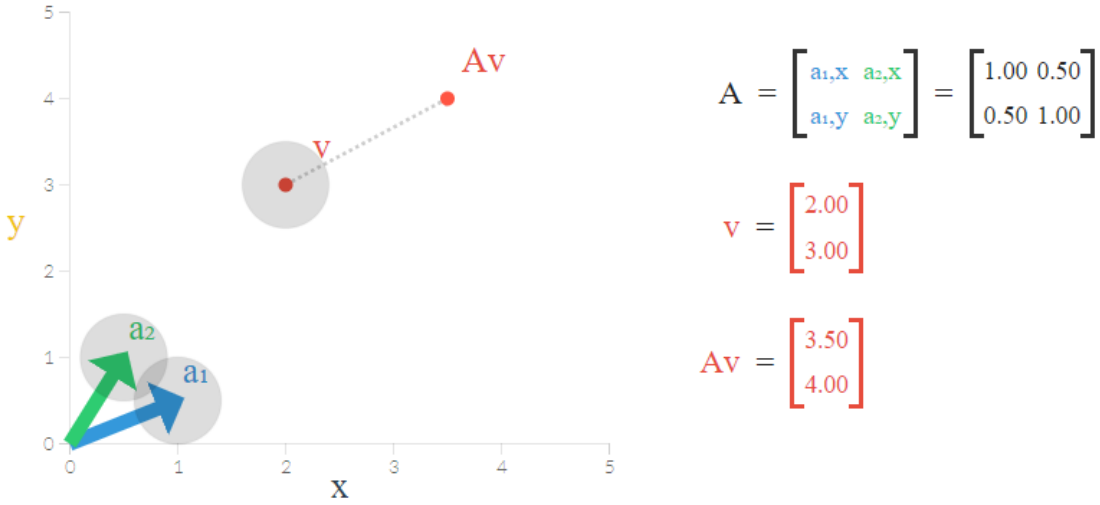
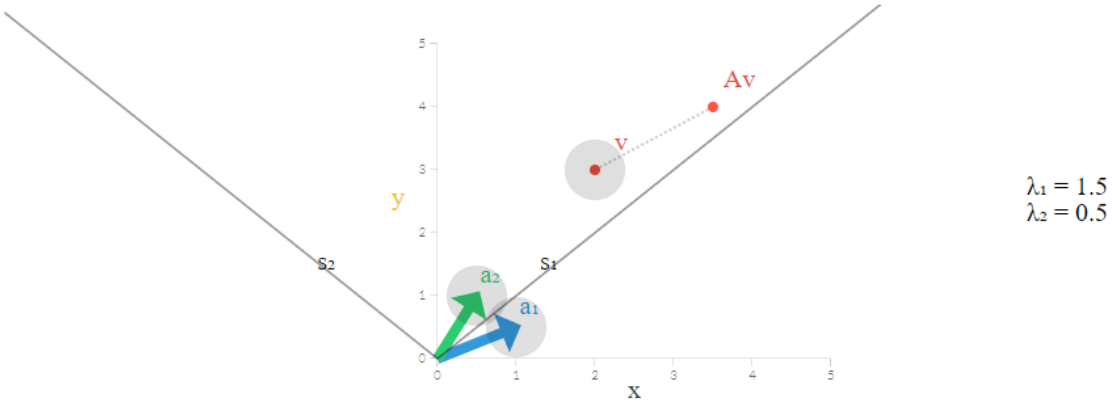


Figure 2-30 **EigenValues-EigenVectors example**

If a line can be drew through the three points (0,0), $\boldsymbol{v}$ and $\boldsymbol{Av}$, then $\boldsymbol{Av}$ is just $\boldsymbol{v}$ multiplied by a number $\boldsymbol{\lambda}$; that is, $\boldsymbol{Av} = \boldsymbol{\lambda v}$. In this case, $\boldsymbol{\lambda}$ is called an **eigenvalue** and $\boldsymbol{v}$ an **eigenvector**. For example, here (1,2) is an eigvector and 5 an eigenvalue.

$$Av = \begin{pmatrix} 1 & 2 \\ 8 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 5 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \lambda v.$$

Below, change the columns of $\boldsymbol{A}$ and drag $\boldsymbol{v}$ to be an eigenvector. Note three facts: First, every point on the same line as an eigenvector is an eigenvector. Those lines are **eigenspaces**, and each has an associated eigenvalue. Second, if you place $\boldsymbol{v}$ on an eigenspace (either s_1 or s_2) with associated eigenvalue $\boldsymbol{\lambda} < 1$, then $\boldsymbol{Av}$ is closer to (0,0) than v $\boldsymbol{v}$; but when $\boldsymbol{\lambda} > 1$, it's farther. Third, both eigenspaces depend on both columns of $\boldsymbol{A}$: it is not as though $\boldsymbol{\alpha_1}$ only affects s_1.



$$[Covariance\ matrix] \cdot [Eigenvector] = [eigenvalue] \cdot [Eigenvector]$$

2.4.5.3.2.1.4 Covariance Matrix

As variance and covariance are defined, we shall look into what a ***Covariance matrix*** is.

$$\begin{bmatrix} V_a & C_{a,b} & C_{a,c} & C_{a,d} & C_{a,e} \\ C_{a,b} & V_b & C_{b,c} & C_{b,d} & C_{b,e} \\ C_{a,c} & C_{b,c} & V_c & C_{c,d} & C_{c,e} \\ C_{a,d} & C_{b,d} & C_{c,d} & V_d & C_{d,e} \\ C_{a,e} & C_{b,e} & C_{c,e} & C_{d,e} & V_e \end{bmatrix}$$

A covariance matrix of some data set in 4 dimensions a,b,c,d.

V_a *: variance along dimension a*

$C_{a,b}$ *: Covariance along dimension a and b*

If we have a matrix X of m*n dimension such that it holds n data points of m dimensions, then covariance matrix can be calculated as

$$C_x = \frac{1}{n-1}(X - \bar{X})(X - \bar{X})^T \quad , X^T = Transpose\ of\ X$$

It is important to note that the covariance matrix contains; variance of dimensions as the main diagonal elements, covariance of dimensions as the off diagonal elements. Also, covariance matrix is symmetric.

As, it's mentioned earlier data need to be spread out i.e. it should have high variance along dimensions. Also we want to remove correlated dimensions i.e. covariance among the dimensions should be zero (they should be linearly independent). Therefore, our covariance matrix should have; large numbers as the main diagonal elements, zero values as the off diagonal elements. We call it a **diagonal matrix**. So, the original data have to be transformed to points such that their covariance is a diagonal matrix. The process of transforming a matrix to diagonal matrix is called **diagonalization**.

This defines the goal of PCA:

1. Find linearly independent dimensions (or basis of views) which can losslessly represent the data points.
2. Those newly found dimensions should allow to predict/reconstruct the original dimensions. The reconstruction/projection error should be minimized.

Projection error: Suppose we have to transform a 2 dimensional representation of data points to a one dimensional representation. So we will basically try to find a straight line and project data points on them. (A straight line is one dimensional). There are many possibilities to select the straight line, two of them are:

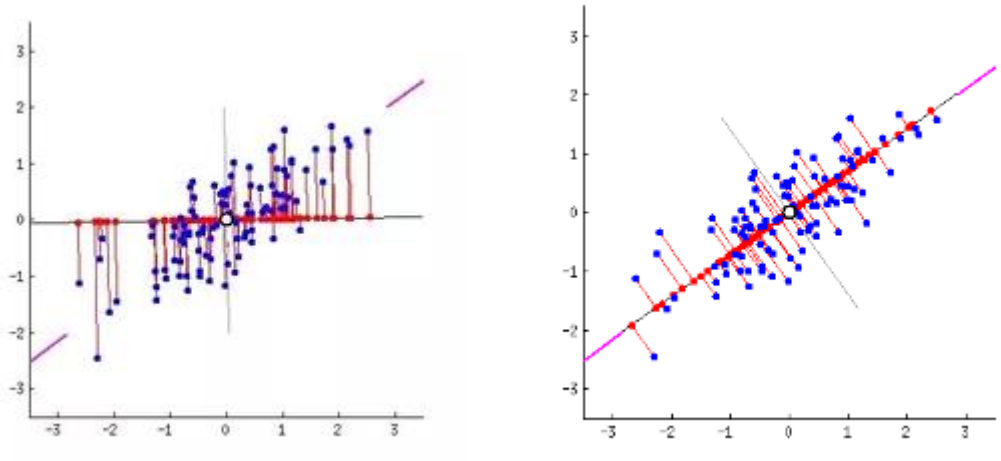


Figure 2-31 **PCA. Principal Axis.**

Magenta line will be our new dimension. The red lines (connecting the projection of blue points on magenta line) i.e. the perpendicular distance of each data point from the straight line is the projection error. Sum of the error of all data points will be the total projection error. Our new data points will be the projections (red points) of those original blue data points. As we can see we have transformed 2 dimensional data points to one dimensional data points by projection them on 1 dimensional space i.e. a straight line. That magenta straight line is called ***principal axis***. Since we are projecting to a single dimension, we have only one principal axis. Clearly, Second choice of straight line is better because: The projection error is less than that in the first case, newly projected red points are more widely spread out than the first case. i.e. more variance. The above mentioned two points are related i.e. if we minimize the reconstruction error, the variance will increase.

Now the original data points need to be transformed such that the covariance matrix of transformed data points is a diagonal matrix.

$$C_x = covariance\ matrix\ of\ original\ data\ set\ X$$
$$C_y = covariance\ matrix\ of\ transformed\ data\ set\ Y$$
$$such\ that,$$
$$Y = PX$$
$$For\ simplicity,\ we\ discard\ the\ mean\ term\ and\ assume\ the$$
$$data\ to\ be\ centered.\ i.e.\ X = (X - \bar{X})$$
$$So,\ C_x = \frac{1}{n}XX^T$$
$$C_y = \frac{1}{n}YY^T$$
$$= \frac{1}{n}(PX)(PX)^T$$
$$= \frac{1}{n}PXX^TP^T$$
$$= P(\frac{1}{n}XX^T)P^T$$
$$= PC_xP^T$$

Here's the trick: If we find the matrix of eigen vectors of C_x and use that as P (P is used for transforming X to Y, see the image above), then C_y (covariance of transformed points) will be a diagonal matrix. Hence Y will be the set of new/transformed data points. Now, if we want to transform points to k dimensions then we will select first k eigen vectors of the matrix C_x (sorted decreasingly according to eigen values) and form a matrix with them and use them as P.

So, for m dimensional original n data points then

$$X : m * nP : k * m$$
$$Y = PX : (k * m)(m * n) = (k * n)$$

Hence, our new transformed matrix has n data points having k dimensions.

(Kumar, 2018)

3. DATA DESCRIPTION & S/W IMPLEMENTATION

3.1 DATA PLATFORM

The **National Center for Biotechnology Information** (**NCBI**) is part of the United **States National Library of Medicine** (**NLM**), a branch of the **National Institutes of Health** (**NIH**). The NCBI houses a series of databases relevant to biotechnology and biomedicine and is an important resource for bioinformatics tools and services. Major databases include GenBank for DNA sequences and PubMed, a bibliographic database for the biomedical literature. Other databases include the NCBI Epigenomics database. All these databases are available online through the Entrez search engine.

The **Gene Expression Omnibus** (**GEO**) is a public repository supported by the American National Center for Biotechnology Information (NCBI) at the American National Library of Medicine (NLM) that accepts raw and processed data with written descriptions of experimental design, sample attributes, and methodology for studies of high-throughput gene expression and genomics, also archives and freely distributes comprehensive sets of microarray, next-generation sequencing, and other forms of high-throughput functional genomic data submitted by the scientific community. In addition to data storage, a collection of web-based interfaces and applications are available to help users query and download the studies and gene expression patterns stored in GEO.

GEO was designed around the common features of most of the high-throughput and parallel molecular abundance-measuring technologies in use today. These include data generated from microarray and high-throughput sequence technologies, for example:

- Gene expression profiling by microarray or next-generation sequencing
- Non-coding RNA profiling by microarray or next-generation sequencing
- Chromatin immunoprecipitation (ChIP) profiling by microarray or next-generation sequencing
- Genome methylation profiling by microarray or next-generation sequencing
- High-throughput RT-PCR
- Genome variation profiling by array (arrayCGH)
- SNP arrays
- Serial Analysis of Gene Expression (SAGE)
- Protein arrays

The GEO database has a flexible and open design that is responsive to developing trends.

GEO requires raw data, processed data and metadata. Raw data facilitates the unambiguous interpretation of the data and potential verification of conclusions. For microarray data, raw data may be supplied either within the Sample record data tables or as external supplementary data files, e.g., Affymetrix CEL. For high-throughput sequencing, GEO brokers the complete set of raw data files, e.g., FASTQ, to the SRA database on your behalf.

3.1.1 DATA TYPES

Processed sequence data files: GEO hosts processed sequence data files, which are linked at the bottom of Sample and/or Series records as supplementary files. Requirements for processed data files are not yet fully standardized and will depend on the nature of the study, but data typically include genome tracks or expression counts.

***Raw sequence data files*:** Raw data are loaded to NCBI's Sequence Read Archive (SRA) database. Use the SRA Run Selector to list and select Runs to be downloaded or analyzed with the SRA Toolkit.

GEO DataSets is a *study-level* database which users can search for studies relevant to their interests. The database stores descriptions of all original submitter-supplied records, as well as curated DataSets.

Geo Database Organization	
Platform	Platform records are supplied by submitters **A Platform record is composed of a summary description of the array or sequencer and, for array-based Platforms, a data table defining the array template.Each Platform record is assigned a unique and stable GEO accession number (GPLxxx). A Platform may reference many Samples that have been submitted by multiple submitters.**
Sample	**Sample records are supplied by submitters** A Sample record describes the conditions under which an individual Sample was handled, the manipulations it underwent, and the abundance measurement of each element derived from it. Each Sample record is assigned a unique and stable GEO accession number (GSMxxx). A Sample entity must reference only one Platform and may be included in multiple Series
Series	**Series records are supplied by submitters** A Series record links together a group of related Samples and provides a focal point and description of the whole study. Series records may also contain tables describing extracted data, summary conclusions, or analyses. Each Series record is assigned a unique and stable GEO accession number (GSExxx).
DataSet	**DataSet records are assembled by GEO curators** As explained above, A GEO Series record is an original submitter-supplied record that summarizes an experiment. These data are reassembled by GEO staff into GEO Dataset records (GDSxxx). A DataSet represents a curated collection of biologically and statistically comparable GEO Samples and forms the basis of GEO's suite of data display and analysis tools. Samples within a DataSet refer to the same Platform, that is, they share a common set of array elements. Value measurements for each Sample within a DataSet are assumed to be calculated in an equivalent manner, that is, considerations such as background processing and normalization are consistent across the DataSet. Information reflecting experimental factors is provided through DataSet subsets. Both Series and DataSets are searchable using the GEO DataSets interface, but only

	DataSets form the basis of GEO's advanced data display and analysis tools including gene expression profile charts and DataSet clusters. Not all submitted data are suitable for DataSet assembly and we are experiencing a backlog in DataSet creation, so not all Series have corresponding DataSet record(s).
Profile	**Profiles are derived from DataSets** A Profile consists of the expression measurements for an individual gene across all Samples in a DataSet. Profiles can be searched using the GEO Profiles interface.

3.1.2 Download GEO data

All GEO data[2] can be downloaded in various formats using a variety of mechanisms. A popular method for downloading data for specific studies is to download directly from Series pages. At the bottom of each Series page, there is a banner with the text "Download family" under which there are links for downloading the data for that Series in 3 different formats:

1. ***SOFT formatted family*** file(s) is a link for downloading all of the Series, Sample and Platform data in a single SOFT formatted file. SOFT is an acronym that stands for "Simple Omnibus Format in Text" and formats the data as line-based, plain text.
2. ***MINiML formatted family*** file(s) is a link for downloading all of the Series, Sample, and Platform data in MiNIML formatted files. MiNIML is an acronym that stands for MIAME Notation in Markup Language, and formats the data as XML with separate data tables. MINiML is essentially an XML rendering of SOFT format.
3. ***Series Matrix*** File(s) is a link for downloading a tab-delimited value-matrix table generated from the "VALUE" column of each Sample record, headed by Sample and Series metadata. This format is convenient for uploading into data programs such as Microsoft Excel or R.

The Series page also contains links to any supplementary files associated with the Series and a link to a tar archive of all supplementary files provided with the Samples, typically raw data files (see Note 10). If only a subset of the supplementary files are required there is an option to customize the set of files in the tar archive by clicking the word "custom" on same line as "GSExxx_RAW.tar". Clicking the "custom" button expands the page to include a list of all Sample supplementary files in the Series with check boxes to select the desired files. Once the boxes next to the needed files have been selected, pressing "Download" initiates the download of a tar archive containing only the selected files. Additional options

[2] https://www.ncbi.nlm.nih.gov/sites/GDSbrowser/

for downloading data, including downloading specific portions of records, or programmatic approaches are described at http://www.ncbi.nlm.nih.gov/geo/info/download.html. (Barrett, 2016)

3.2 DATASETS

Sample type	RNA
Extracted molecule	total RNA
Extraction protocol	• RNA extraction was performed as Affymetrix GeneChip expression technical manual (Affymetrix, Inc., Santa Clara, CA). Briefly, total RNA samples were extracted, followed by measurement of the A2260/a280 ratio with at least 1.8 for pure RNA. Quality of the RNA was checked by an Agilent 2100 Bioanalyzer. The Bioanalyzer gel profile exhibited a 28S band that is 2 times more intense than 18S ribosomal RNA. The quality of RNA was assessed by agarose gel electrophoresis. • Total RNA was isolated from laser-capture microdissected tissue using the Picopure RNA isolation kit from Arcturus.
Label	biotin
Label protocol	• Labeling was performed according to Affymetrix Gene Chip technical manual • TRIzol extraction • A dual-round amplification procedure was performed on 100 nanograms total RNAusing the MessageAMP aRNA kit from Ambion. In the second round, biotin-labeled cRNA was generated from the double-stranded cDNA template using a nucleotide mix that contained biotinylated CTP and UTP (Enzo RNA Transcript Labeling Kit; Enzo Diagnostics, Farmingdale, NY). The biotinylated cRNA was purified using RNeasy affinity columns (Qiagen, Valencia, MD). • Target was labeled with Enzo BioArray High Yield RNA Trascript Labeling Kit (Enzo Life TEchnologies, Farmingdale, NY) according to manufacturer's protocol.
Hybridization protocol	• The targets for Affymetrix DNA microarray analysis were prepared according to the manufacturer's instructions. Biotin-labeled cRNA, produced by in vitro transcription, was fragmented and hybridized to Affymetrix GeneChip Human Genome U133 Plus 2.0 Arrays at 45°C for 16 hr and then washed and stained using the GeneChip Fluidics. • For each GeneChip, 20 micrograms of the labeled product was fragmented in 40 mM Tris-acetate, pH 8.1, 100mM KOAc, 30mM MgOAc, for 35 minutes at 94 degrees-Celsius, to an average size of 35 to 200 bases. 15 micrograms of this fragmented, biotinylated cRNA, along with hybridization controls supplied by the manufacturer (Affymetrix), were hybridized to the arrays for 16 hours at 45 degrees-Celsius and 60 rpm. Arrays were washed and stained according to the standard Antibody Amplification for Eukaryotic Targets protocol (Affymetrix)
Scan protocol	• The arrays were scanned by a GeneArray Scanner and patterns of hybridization detected as light emitted from the fluorescent reporter groups incorporated into the target and hybridized to oligonucleotide probes. • The stained GeneChip arrays were scanned at 488 nm using an Affymetrix Gene Chip Scanner 3000 (Affymetrix, Santa Clara, CA).
Data	The data were analyzed with Agilent Gene Spring GX 7.3 version using

Processing	Affymetrix default analysis settings and GC-RMA as normalization method. • .DAT files were generated using GCOS 1.2.1 software (Affymetrix). CEL files were generated using MAS 5.0 softare (Affymetrix) with target signals for probe sets scaled to 500. Log2 expression values for individual probe sets were generated from .CEL files via robust multi-array average (gcRMA).

Table 2 Data Information from https://www.ncbi.nlm.nih.gov/geo/

For the purposes of this study, 4 datasets have been obtained from Gene Expression Omnibus (GEO) database. The data which were chosen contain both healthy and cancer samples for the classification.

1. **GDS4102-Pancreatic Tumor and Normal tissue samples.**
   Analysis of tumor tissue and normal tissue in pancreatic cancer samples. The fresh frozen samples were obtained during surgical procedures. The cell types include: bone marrow, peripheral blood, bone marrow CD34plus and PBSC CD34plus. This experiment consists of 36 tumor samples and 16 normal samples. A total of 52 samples, with 54,613 gene expression levels for each sample, which run on Platform GPL96 Affymetrix [Human Genome U133 Plus 2.0 Array] GeneChip® which Technology type is in situ oligonucleotide.
2. **GDS3233-Cervical cancer tumorigenesis**
   Analysis of cervical cancer (CC) primary tumors and cell lines. A total of 52 samples were included in this study, which include 33 primary tumors, 9 cell lines, and 24 normal cervical epithelium with 14,062 gene expression levels for each sample. The gene expression profiles in cervical cancer run on Platform GPL96 Affymetrix [Human Genome U133A Array] GeneChip® which Technology type is in situ oligonucleotide.
3. **GDS3139-Breast cancer: histologically normal breast epithelium**
   Analysis of histologically normal breast epithelia of breast cancer patients. Results provide insight into the molecular abnormalities in normal appearing breast epithelium in breast cancer and the roles these abnormalities play in carcinogenesis. 29 samples from histologically normal microdissected breast epithelium are included in this series. 14 samples are from epithelium adjacent to a breast tumor, 15 samples were obtained from patients undergoing reduction mammoplasty without apparent breast cancer. Each sample has 22,283 gene expression levels. The gene expression data run on Platform GPL96 Affymetrix [Human Genome U133A Array] GeneChip® which Technology type is in situ oligonucleotide.
4. **GDS3057-Acute myeloid leukemia**
   Leukemic blasts from 26 acute myeloid leukemia (AML) patients with normal hematopoietic cells at a variety of different stages of maturation from 38 healthy donors. Results provide insight into the possible clinical significance of those genes with AML-specific expression changes. Each sample has 22,283 gene expression levels run on Platform GPL96 Affymetrix [Human Genome U133A Array] GeneChip® which Technology type is in situ oligonucleotide.

Cancer Type	Dataset	Reference Series	Platform	Total Samples	Healthy Samples	Cancer Samples	Features
Pancreatic	GDS4102	GSE16515	GPL570	52	16	36	54,613
Cervical	GDS3233	GSE9750	GPL96	52	24	28	14,062
Breast	GDS3139	GSE9574	GPL96	29	15	14	22,283
Acute Myeloid Leukemia	GDS3057	GSE9476	GPL96	64	38	26	22,283

Table 3 **Overview of Study Datasets**

In our study, we interest in supervised classification of different cancer cases. In order to do so, we have assigned the **class**, in which each sample belongs to, as a separate binary feature with name “class”. So we created **labels** for each sample depending on its disease state. The patient state is declared in the annotation note following the .SOFT file we downloaded. For healthy donors the “class” value is zero (0) and for cancer patients the “class” value is one (1). This process has been done in Microsoft Excel, where we can open in tabular view files of big data (like .SOFT files) with thousands of features and transform them into .csv files, which are the most common and manageable dataset file for machine learning.

	ID_REF	GSM414924	GSM414925	GSM414926	GSM414927	GSM414929	GSM414931	GSM414933	GSM414935	GSM414936	GSM414937	GSM414939	GSM414941	GSM414943	GSM414944	GSM414945	GSM414946	GSM414948	GSM414949	GSM414950	GSM
1	1552563_a_at	14.75	15.29	18.91	14.43	35.53	10.83	21.64	8.32	5.89	9.19	8.28	10.54	9.22	10.58	15.33	15.74	9.96	12.21	8.39	
2	1552829_at	12.73	13.6	12.01	10.93	11.7	10.18	10.04	10.04	7.04	10.06	8.53	9.98	13.2	15.98	10.44	20.31	10.14	11.97	23.07	
3	1552867_at	208.3	90.75	196.2	82.05	79.95	88.91	45.41	79.05	44.55	51.94	92.43	100.1	98.77	68.28	89.68	52.29	69.12	83.71	103.2	
4	1552961_at	15.45	14.74	12.85	17.31	13.67	12.8	13.76	12.73	10.83	11.67	12.27	13.05	12.14	13.27	19.46	12.62	12.64	18.49	11.23	
5	1552974_at	9.91	10.38	11.69	15.52	10.07	10.55	13.42	9.66	8.76	9.31	10.06	9.69	10.24	10.31	10.9	9.16	10.23	10.81	8.96	
6	1552975_x_at	9.81	9.4	11.94	15.75	9.75	10.85	11.33	9.94	8.97	10.32	10.16	8.78	10.24	10.22	11.08	10.13	10.26	10.98	8.72	
7	1553069_at	4.07	4.05	4.11	4.31	4.09	4.4	4.14	3.88	3.99	3.94	4	4.02	4.01	4.32	4.12	3.91	3.92	4.02	3.93	
8	1553083_at	4.05	3.99	4.08	4.38	4	5.22	4.17	3.99	3.9	3.94	3.91	5.35	4.02	4	6.44	3.94	4.03	4.05	3.86	
9	1553275_s_at	29.9	42.67	62.87	8.79	26.19	73.62	60.68	82.97	27.22	51.09	69.74	66.92	42.27	25.51	19.47	76.08	47.64	101.3	42.4	
10	1553354_a_at	7.99	8.33	8.47	9.75	9.03	8.74	8.58	7.56	10.75	8.44	8.28	8.99	8	10.8	14.12	8.89	8.62	8.74	8.87	
11	1553355_at	4.2	3.87	4.02	6.01	3.98	4.04	4.86	4.25	4.01	4.02	3.85	3.88	3.95	4.07	9.05	4.13	3.96	4.45	3.93	
12	1553356_at	4.5	4.46	4.63	4.94	4.53	4.52	4.8	4.45	4.43	4.42	4.42	4.39	4.46	4.63	4.63	4.42	4.52	4.58	4.45	
13	1553372_at	8.96	10.71	11.21	15.97	9.54	9.63	22.15	10.15	8.6	9.28	10.45	10.46	10.41	10.74	11.5	10.3	9.3	10.88	9.32	
14	1553398_at	5.09	4.78	4.96	5.82	4.85	5.03	5.49	4.86	4.66	4.83	4.83	4.66	4.82	4.99	4.98	4.82	4.94	4.93	4.7	
15	1553439_at	4.64	3.95	4.59	4.91	4.55	4.68	4.71	4.5	5.23	4.58	4.59	4.51	4.67	4.55	4.69	4.57	4.64	4.68	4.44	
16	1553456_at	4.89	4.85	5.02	5.4	4.9	4.97	5.69	4.87	4.73	4.86	4.76	4.8	4.89	4.91	5.04	5.58	5.02	4.94	4.69	
17	1553462_at	11.66	9.58	11.58	18.91	10.58	10.75	10.99	9.3	9.43	11.18	9.99	9.68	10.85	10.27	11.22	10.28	9.59	10.9	10.12	
18	1553475_at	5.98	5.7	6.07	6.69	5.73	5.96	6.11	5.75	5.45	5.78	5.68	5.62	6.52	6.64	6.1	5.79	6.05	6.09	5.53	
19	1553498_at	10.25	9.77	10.43	11	10.32	11.4	10.38	9.87	9.81	9.93	9.94	9.69	10.18	10.25	15.82	9.76	9.74	10.29	10.11	
20	1553546_at	6.9	6.41	7.2	9.8	7.03	7.19	7.27	6.8	6.5	6.71	6.8	6.53	7.01	7.03	7.22	6.74	6.97	7.09	6.58	
21	1553547_at	5.05	4.94	5.11	6.5	5.06	7.89	5	4.96	4.97	4.97	4.87	4.91	5.01	5.06	5.19	4.92	5.02	5.08	4.95	
22	1553881_at	4.42	4.4	4.73	5.94	4.4	4.54	4.52	4.36	4.4	4.58	4.38	4.62	4.4	4.42	4.48	4.46	4.41	4.47	4.18	
23	1554007_at	111.7	129.5	192.8	385.6	125.7	82.87	118.6	442	59.58	241.9	83.36	249.6	75.14	41.57	48.07	269	76.14	304.4	244.8	
24	1554232_a_at	4.66	4.59	4.75	5.26	4.65	4.73	4.77	4.68	4.53	4.59	4.57	4.51	4.7	4.71	4.92	4.66	4.68	4.73	4.54	
25	1554281_at	7.89	7.77	8.51	11.16	11.81	8.57	8.58	7.48	8.39	7.4	7.78	8.14	8.71	8.24	8.53	8.19	7.89	8.2	8.12	
26	1554372_at	7.45	6.92	7.04	8.3	7.52	7.04	7.69	7.46	6.85	7.11	8.5	6.66	7.66	7.44	6.81	7.42	8.66	7.49	7.4	
27	1554374_at	5.36	4.65	5.42	5.47	5.34	5.42	6.85	5.32	5.24	5.28	5.17	5.35	5.37	5.38	5.46	5.27	5.41	5.45	4.73	
28	1554404_a_at	4.25	4.19	4.27	4.5	4.21	4.23	4.33	4.17	4.14	4.18	4.17	4.11	4.21	4.22	4.28	4.64	4.21	4.25	4.16	
29	A1BG	15.49	15.38	15.84	28.54	15.09	16.66	17.39	16.05	15.27	18.92	15.5	18.06	13.98	10.12	15.62	15.05	15.68	15.44	15.39	
30	A1BG-AS1	18.61	14.83	16.28	15.09	12.03	15.04	12.58	14.74	11.45	11.85	11.49	15.29	12.1	12.47	10.42	11.41	10.87	14.64	12.03	
31	A1CF	137.7	25.295	86.555	197.2	41.61	10.615	10.19	77.26	68.49	27.95	11.57	38.485	15.51	11.115	641.05	27.165	20.83	59.065	47.565	
32	A2M	803.89	1043.475	985.575	1469.04	1536.03	1079.77	254.325	2235.1	1114.325	1063.06	1341.355	970.485	1391.88	406.255	4622.42	1291.1	1197.915	1666.15	1909.985	1
33	A2M-AS1	13.62	17.75	21.43	22.56	19.59	7.28	5.8	10.26	8.71	18.56	12.47	10.47	19.73	22.57	13.56	23.71	10.6	28.85	30.13	

	ID_REF	GSM414924	GSM414925	GSM414926	GSM414927	GSM414929	GSM414931	GSM414933	GSM414935	GSM414936	GSM414937	GSM414939	GSM414941	GSM414943	GSM414944	GSM414945	GSM414946	GSM414948	GSM414949	GSM414950	GSM
54582	65635_at	158.1	218.3	223.7	175.9	185.8	183.1	170.2	128.1	152.8	138.4	142.7	158.7	157.8	204.8	97.24	179.1	172.9	216.8	152.3	
54583	65718_at	70.41	77.24	75.18	47.91	52.69	106.8	43.39	128.6	79.51	69.64	100.5	92.06	92.42	70.54	38.58	114.5	85.83	82.89	131.6	
54584	65770_at	124.9	123.5	121.3	86.26	116	130.8	132.7	133.6	92.52	120.1	139.7	119.3	141.6	157.8	96.67	117.3	170.2	121.3	134.5	
54585	65884_at	106	150.4	145.8	67.51	114.9	102.2	181.2	137.1	114.2	132.4	124.4	102.6	96.3	138.5	65.38	111.3	93.08	108.3	111.6	
54586	66053_at	34.18	67.44	35.21	23.84	22.91	36.89	24	28.73	34.34	26.6	34.94	22	30.58	32.9	22.57	44.18	27.7	14.06	17.48	
54587	71933_at	10.81	13.18	10.3	25.43	9.39	10.57	12.76	10.3	9.62	10.32	8.82	11.88	10.52	12.32	12.76	10.99	11.28	16.79	13.63	
54588	74694_s_at	216.1	190.1	211.1	139.2	175.3	155.8	151.5	140.8	112.6	160.9	199.4	130.1	141.2	99.45	190	141.3	162.7	126.4	139.2	
54589	76897_s_at	27.55	23.76	15.65	19.81	35.1	22.31	24.92	22.62	47.77	63.35	33.03	23.45	27.08	47.69	20.02	27.52	38.03	48.53	59.73	
54590	77508_r_at	62.67	43.15	47.63	55.87	45.12	48.73	54.61	40.68	46.5	38.88	51.01	39.9	42.99	45.54	48.3	43.31	45.46	49.72	43.36	
54591	78047_s_at	162.1	134.1	153.7	163.9	130.3	207.8	314.2	131.4	153.7	173.3	141.5	163.8	152.3	102.4	138	148.9	186.7	149.9	163.2	
54592	78330_at	6.31	6.44	6.47	6.66	6.17	7.75	6.51	6.21	6.18	6.48	6.19	6.13	6.52	6.23	8.66	6.08	6.28	6.37	6.12	
54593	78383_at	65.79	55.2	41.67	111.3	57.98	49.12	152.8	53.09	59.03	65.57	59.75	50.17	67.38	53.01	56.83	66.1	46.45	77.95	70.9	
54594	78495_at	52.99	53.51	73.85	49.32	58.97	52.18	39.75	65.8	59.67	53.59	51.59	62.9	62.23	143.1	93.46	82.54	78.23	61.21	69.58	
54595	79005_at	57.15	37.2	69.61	77.31	69.65	73.94	65.04	49.81	59.95	65.8	73.48	56.74	63.8	69.79	75.49	58.01	62.8	62.72	53.19	
54596	81737_at	79.29	114.5	174.1	80.85	101.6	75.93	64.34	44.95	80.42	70.44	82.59	86.88	51.17	112.5	67.39	85.75	45.48	166.6	95.66	
54597	81811_at	121.7	122.5	137.5	73.81	93.42	61.48	111.2	71.39	73.33	50.67	49.58	78.7	50.01	78.25	94.98	141.5	71.58	73.85	88.11	
54598	823_at	49.83	196.2	186.3	33.96	31.95	45.48	110.8	189.2	52.71	173.6	94.25	142.9	59.65	63.19	37.41	54.27	56.31	64.95	65.68	
54599	87100_at	103.2	133.2	172.8	73.3	233.5	144.2	30.66	35.6	86.37	81.25	64.41	84.23	41.25	16.75	33.22	122.4	264.4	81.03	50.22	
54600	89476_r_at	115.7	87.86	114.3	108.7	106.7	112.6	103.3	89.66	75.98	107.4	104.5	89.24	84.4	90.46	94.38	86.14	90.05	73.45	88.42	
54601	89948_at	33.57	30	34.13	34.74	40.04	48.89	63.72	33.22	36.73	28.6	34.95	33.96	50.05	29.13	44.96	29.16	29.45	35.19	33.26	
54602	89977_at	7.37	11.83	7.96	11.05	8.2	8.37	8.19	9.26	23.28	14.14	11.84	11.09	10.57	8.61	8.59	8.82	8.63	19.94	13.15	
54603	90265_at	824.1	405.2	853.1	211.4	470.8	394.1	928.2	372.9	326.5	363.8	445.2	570.5	414	106	260	582.5	318.6	140.5	269.4	
54604	90610_at	130.2	141.3	157.8	115.4	112.5	128.8	185.4	152.4	189.9	143.7	154.6	136.1	184.7	106.7	115.4	144.4	132.3	114.8	146.4	
54605	91580_at	6.38	6.05	6.63	8	6.38	6.56	6.74	6.23	6.15	6.38	6.24	6.42	6.43	6.43	6.71	6.15	6.45	6.56	6	
54606	91617_at	69.29	71.19	96.83	47.52	53.65	46.49	34.6	48.27	52.48	41.99	61.11	37.42	38.93	48.11	76.35	94.37	51.42	51.12	47.12	
54607	91682_at	7.63	6.84	6.77	9.54	7.92	8.57	8.02	7.14	8.44	7.45	7.61	7.1	7.97	7.77	8.08	6.81	7.87	9.54	7.35	
54608	91684_g_at	40.37	35.79	47.45	34.41	61.66	62.88	65.55	45.36	47.23	44.43	48.67	44.99	38.94	35.41	43.43	62.95	51.08	26	35.83	
54609	91703_at	78.23	114.2	95.1	17.48	66.33	69.09	40.74	76.52	55.15	106.3	119.2	74.22	62.37	106.9	17.42	81.3	126.2	42.8	87.17	
54610	91816_f_at	26.75	28.67	31.02	5.88	13.99	42.05	25.39	25.96	19.04	23.39	52.47	33.6	44.77	5.4	109.2	24.44	45.52	34.89	35.97	
54611	91826_at	557.4	2390	718.3	95	268.1	870	212.7	243.2	198.4	418.8	579.9	387.8	524.3	34.4	10.15	995.3	587.6	298.5	306.8	
54612	91920_at	8.35	8.94	9.77	9.22	9.3	10.54	13.55	8.94	11.95	8.37	10.02	9.35	9.65	10.72	9.47	9.61	9.42	9.28	8.78	
54613	91952_at	20.1	22.03	26.59	18.56	20.43	26.46	26.15	23.77	21.24	25.43	29.71	17.62	27.85	24.63	33.95	30.33	25.13	22.65	28.37	
54614	class	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Figure 3-1 **A tabular view of gene expression levels for GDS4102-Pancreatic tissue samples**

The first column titled "ID_REF" contains the gene symbols, our features. As we can see, the last one is the label for each sample with the name class. The first row contains the Donors' number.

The GDS4102 data shape is 52 x 54614 (Samples x Features)

3.3 SOFTWARE

3.3.1 Python

There are a lot of choices in order to apply machine learning techniques like Matlab and R. For our study, we chose the programming language Python (version 3.7). Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. Directly from the platform and without involving DevOps, data scientists can develop and deploy AI and machine learning models rapidly into production. In its environment it contains a variety of application. Here, we develop our program in Jupyter Notebook and JupyterLab.

Project Jupyter is a non-profit, open-source project, born out of the IPython Project in 2014 as it evolved to support interactive data science and scientific computing across all programming languages. Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the modified BSD license. Jupyter is developed in the open on GitHub, through the consensus of the Jupyter community. (Jupyter.org) (anaconda.com/why-anaconda/) (python.org/about/) (Kuhlman, 2013)

3.3.2 Python Libraries

Python provides us with a variety of scientific libraries for data mining and machine learning tasks. In this study we have used the followings:

- Scipy (version 1.3.1) is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data. With SciPy, an interactive Python session becomes a data-processing and system-prototyping environment rivaling systems, such as MATLAB, IDL, Octave, R-Lab, and SciLab. (scipy)
- Numpy: (version 1.17.2) is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array, object sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

- Matplotlib: (version 3.0.3) is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

- Pandas: (version 0.24.2) pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

- Sklearn: (version 0.21.3) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. (Buitinck et al.) (wiki/Scikit-learn) (scikit-learn.org)

4. PROPOSED EXPERIMENT

All the figures are Cell results from 4 different Notebooks run in Jupyter Notebook.

4.1 Preprocess

A necessary step in machine learning is data preprocess. Before data can be analyzed, they must be organized into an appropriate form. Data preparation is the process of manipulating and organizing data prior to analysis. Data preparation is typically an iterative process of manipulating raw data, which is often unstructured and messy, into a more structured and useful form that is ready for further analysis. The whole preparation process consists of a series of major activities (or tasks) including data profiling, cleansing, integration, and transformation. (C., 2010)

In this study, we have loaded the .csv files with the help of pandas read_csv() function and **DataFrame structure**. In some of the cases, we had to transpose the matrix in order to get a shape in the form $(samples\ x\ feautres)$ where samples are in rows and their features in columns. The next step is to drop duplicate values and find the NA tabs, which were filled the median of the data.

4.2 Analyze Data

4.2.1 Peek on Data

We can take a look at our DataFrame by printing it. A useful information to keep here is that the gene expression datasets have a small number of examples and a large number of features.

Data Table:

ID_REF	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	\
GSM241999	3088.9	107.2	203.2	775.3	42.6	393.2	241.3	
GSM242000	2377.7	140.7	95.1	870.9	58.1	679.9	135.0	
GSM242001	2348.5	37.0	91.2	886.5	70.5	417.6	155.0	
GSM242002	2078.9	84.5	144.4	853.8	88.2	632.0	139.4	
GSM242003	2335.8	456.7	448.3	1491.9	463.0	781.2	278.5	
GSM242004	2955.6	95.1	28.0	906.2	85.0	378.2	237.2	
GSM242005	2774.5	69.5	205.7	1060.9	100.4	547.1	249.3	
GSM242006	2978.1	98.5	103.4	791.8	40.3	421.1	48.7	
GSM242007	2963.9	59.9	168.4	954.2	43.3	811.6	166.4	
GSM242008	1932.5	31.2	150.2	840.8	75.4	346.3	141.0	
GSM242009	3435.7	159.0	243.4	897.5	87.9	571.4	208.7	
GSM242010	1963.7	82.0	106.6	705.0	42.5	480.8	75.9	
GSM242011	2619.0	28.7	166.0	1068.3	169.0	533.0	149.8	
GSM242012	2461.4	26.7	82.6	942.3	71.8	630.2	186.0	
GSM242013	2776.9	86.7	162.7	843.1	51.6	778.1	176.3	
GSM242014	2709.0	204.8	147.8	908.1	77.5	405.7	145.1	
GSM242015	2749.9	105.9	101.2	1014.0	66.9	595.7	221.6	
GSM242016	4122.9	68.3	123.8	1747.4	101.7	1452.0	155.6	
GSM242017	2367.4	9.9	98.4	793.5	59.6	594.4	137.5	
GSM242018	2763.2	20.0	194.3	925.4	85.9	783.5	111.7	
GSM242019	4260.7	37.4	278.6	1273.0	65.8	1345.2	191.0	
GSM242020	2612.5	119.3	186.0	806.2	84.3	647.5	192.7	
GSM242021	2992.5	107.6	167.9	776.6	66.4	686.4	172.5	
GSM242022	3128.5	88.6	139.0	925.7	27.6	628.2	219.5	
GSM242023	3310.1	8.8	141.6	1090.3	39.9	483.0	133.1	
GSM242024	2252.8	39.4	81.7	872.6	45.5	396.4	114.8	
GSM242025	3110.9	56.0	113.6	709.4	56.7	526.1	152.3	
GSM242026	2405.2	83.8	85.8	760.5	88.4	400.9	163.2	
GSM242027	2390.3	115.4	73.6	709.2	59.3	606.7	128.6	

ID_REF	1320_at	1405_i_at	1431_at	...	AFFX-r2-Hs28SrRNA-M_at	\
GSM241999	20.5	59.3	84.5	...	190.6	
GSM242000	17.9	23.8	64.9	...	215.8	
GSM242001	52.4	4.6	61.0	...	184.0	
GSM242002	57.2	22.0	111.8	...	195.7	
GSM242003	369.4	80.6	218.1	...	302.7	
GSM242004	66.5	8.2	53.6	...	240.3	
GSM242005	14.5	34.6	155.7	...	235.6	
GSM242006	95.4	10.9	53.0	...	197.0	
GSM242007	10.2	65.0	11.8	...	207.7	
GSM242008	10.6	26.3	98.3	...	163.2	
GSM242009	18.0	26.1	45.0	...	391.8	
GSM242010	14.6	9.8	33.0	...	169.1	
GSM242011	15.1	23.7	108.2	...	75.0	
GSM242012	15.8	71.3	58.7	...	225.3	
GSM242013	11.6	12.7	65.1	...	271.2	
GSM242014	70.7	15.5	74.2	...	181.0	
GSM242015	81.6	80.2	141.0	...	162.1	
GSM242016	89.6	152.0	29.1	...	245.5	
GSM242017	12.2	36.5	70.3	...	132.4	
GSM242018	45.6	8.9	100.0	...	144.3	
GSM242019	21.4	273.4	91.8	...	242.7	
GSM242020	88.4	25.4	78.6	...	223.3	
GSM242021	12.0	34.9	67.9	...	231.4	
GSM242022	9.7	142.7	95.4	...	1283.9	
GSM242023	20.3	224.2	9.3	...	244.6	
GSM242024	7.2	30.7	68.2	...	143.0	
GSM242025	5.1	106.5	67.8	...	80.1	
GSM242026	5.5	14.0	71.6	...	74.0	
GSM242027	7.1	16.5	71.3	...	103.8	

Figure 4-1 (A) **Peek at values of Breast Cancer Dataset GDS3139 as an example**

ID_REF	AFFX-r2-P1-cre-3_at	AFFX-r2-P1-cre-5_at	AFFX-ThrX-3_at	\
GSM241999	83742.3	60663.9	23.2	
GSM242000	66514.9	48880.5	15.1	
GSM242001	37405.6	26240.8	5.6	
GSM242002	38944.8	25854.4	7.9	
GSM242003	194545.0	136124.0	61.4	
GSM242004	78297.7	63057.6	29.4	
GSM242005	64253.2	39705.3	10.8	
GSM242006	36108.7	23632.5	7.7	
GSM242007	54110.2	37729.5	7.1	
GSM242008	35295.0	25760.2	10.4	
GSM242009	51636.5	39749.5	13.8	
GSM242010	36381.1	26691.4	12.5	
GSM242011	90633.1	62803.2	20.8	
GSM242012	38732.8	27646.0	11.3	
GSM242013	17673.1	13351.7	7.6	
GSM242014	50426.2	27174.8	13.7	
GSM242015	50774.5	40469.6	12.2	
GSM242016	66559.5	40492.6	22.7	
GSM242017	40490.0	26865.2	8.4	
GSM242018	49626.4	37274.6	6.1	
GSM242019	39053.5	29820.4	6.9	
GSM242020	24362.3	16816.0	5.6	
GSM242021	15121.2	11495.4	9.1	
GSM242022	30895.9	25656.7	8.9	
GSM242023	41387.2	40486.3	8.7	
GSM242024	26677.6	23233.5	5.2	
GSM242025	47513.6	38717.0	4.7	
GSM242026	48758.0	36827.2	9.9	
GSM242027	26728.2	20174.2	7.5	

ID_REF	AFFX-TrpnX-M_at	class
GSM241999	5.8	0.0
GSM242000	4.0	0.0
GSM242001	4.8	0.0
GSM242002	4.6	0.0
GSM242003	31.4	0.0
GSM242004	7.5	0.0
GSM242005	6.5	0.0
GSM242006	3.2	0.0
GSM242007	3.2	0.0
GSM242008	2.5	0.0
GSM242009	10.6	0.0
GSM242010	2.9	0.0
GSM242011	8.5	0.0
GSM242012	3.3	0.0
GSM242013	3.0	0.0
GSM242014	3.1	1.0
GSM242015	5.0	1.0
GSM242016	4.4	1.0
GSM242017	3.1	1.0
GSM242018	6.4	1.0
GSM242019	3.5	1.0
GSM242020	3.9	1.0
GSM242021	3.0	1.0
GSM242022	2.9	1.0
GSM242023	7.4	1.0
GSM242024	2.6	1.0
GSM242025	4.7	1.0
GSM242026	45.5	1.0
GSM242027	3.0	1.0

Figure 4-2 (B) **Peek at values of Breast Cancer Dataset GDS3139 as an example**

4.2.2 Data Dimensions

The results are listed in rows then columns. In this study we have chosen 4 different datasets depending on their shape. We cover cases with big number of samples and also features, big number of samples but fewer features, small number of samples with also small number of features and finally big number of samples but small number of features.

```
data shape:
 (52, 54614)
```

Figure 4-3 **Pancreatic Cancer Dataset GDS4102 shape**

```
data shape:
 (52, 14063)
```

Figure 4-4 **Cervical Cancer Dataset GDS3233 shape**

```
data shape:
 (64, 22284)
```

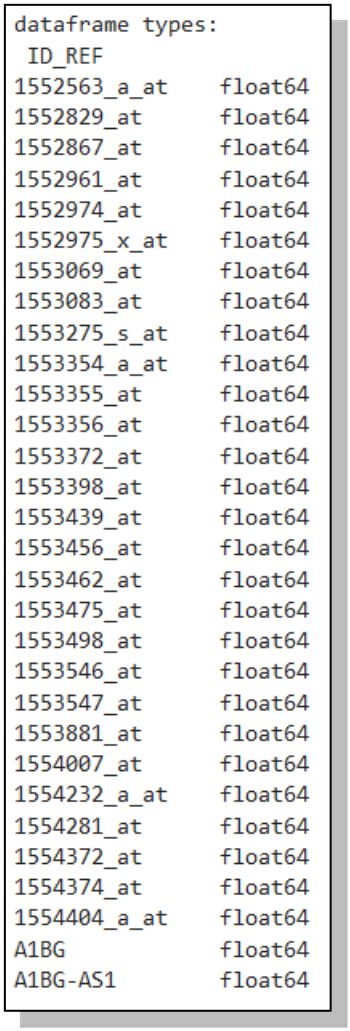
Figure 4-5 **AML Cancer Dataset GDS3057 shape**

```
data shape:
 (29, 22284)
```

Figure 4-6 **Breast Cancer Dataset GDS3139 shape**

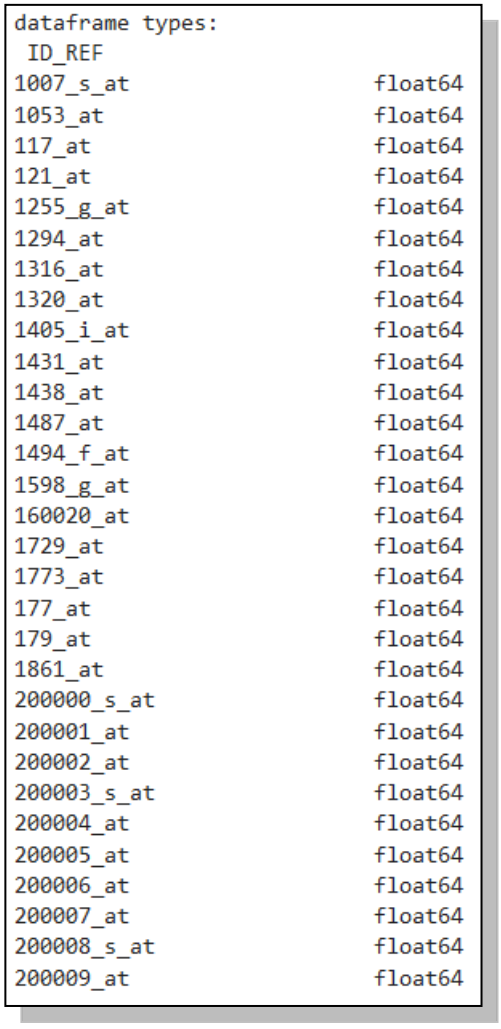
4.2.3 Attribute Data Type

The type of each attribute is important. Strings may need to be converted to floating point values or integers to represent categorical or ordinal values. We can get an idea of the types of attributes by peeking at the raw data, as above. (Brownlee) In our experiment, all the features are **float64** type.



```
dataframe types:
 ID_REF
1552563_a_at    float64
1552829_at      float64
1552867_at      float64
1552961_at      float64
1552974_at      float64
1552975_x_at    float64
1553069_at      float64
1553083_at      float64
1553275_s_at    float64
1553354_a_at    float64
1553355_at      float64
1553356_at      float64
1553372_at      float64
1553398_at      float64
1553439_at      float64
1553456_at      float64
1553462_at      float64
1553475_at      float64
1553498_at      float64
1553546_at      float64
1553547_at      float64
1553881_at      float64
1554007_at      float64
1554232_a_at    float64
1554281_at      float64
1554372_at      float64
1554374_at      float64
1554404_a_at    float64
A1BG            float64
A1BG-AS1        float64
```

Figure 4-8 **First example of Data Type Attributes on GDS4102.** The first column shows the names of features (genes). The second column is each feature's type.



```
dataframe types:
 ID_REF
1007_s_at               float64
1053_at                 float64
117_at                  float64
121_at                  float64
1255_g_at               float64
1294_at                 float64
1316_at                 float64
1320_at                 float64
1405_i_at               float64
1431_at                 float64
1438_at                 float64
1487_at                 float64
1494_f_at               float64
1598_g_at               float64
160020_at               float64
1729_at                 float64
1773_at                 float64
177_at                  float64
179_at                  float64
1861_at                 float64
200000_s_at             float64
200001_at               float64
200002_at               float64
200003_s_at             float64
200004_at               float64
200005_at               float64
200006_at               float64
200007_at               float64
200008_s_at             float64
200009_at               float64
```

Figure 4-7 **Second example of Data Type Attributes on GDS3139.** The first column shows the names of features (genes). The second column is each feature's type.

4.2.4 Descriptive Statistics

Descriptive statistics can give us great insight into the properties of each attribute. Often you can create more summaries than you have time to review. The describe () function on the Pandas DataFrame lists 8 statistical properties of each attribute. (Brownlee) They are:

- **Count.** The count of rows (samples)
- **Mean or average**. Symbolically, if we have a data set consisting of the values $a_1, a_2, \ldots, a_n$ then the arithmetic mean $\boldsymbol{\mu}$ is defined by the formula: $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} a_n = \frac{a_1+a_2+\cdots+a_n}{n}$ (wiki/Arithmetic_mean)
- **Standard Deviation** is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the values are spread out over a wider range. The formula is: $= \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$. Let X be a random variable with mean value μ: $[X] = \mu$. Here the operator E denotes the average or expected value of X. Then the standard deviation of X is the quantity $\sigma = \sqrt{E[(X-\mu)^2]}$ (wiki/Standard_deviation)
- **Minimum and Maximum Value**
- **25th 50th and 75th Percentile**. A percentile is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations fall. (wiki/Percentile)

statistic descriptions:

ID_REF	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	1405_i_at	\
count	29.000	29.000	29.000	29.000	29.000	29.000	29.000	29.000	29.000	
mean	2767.828	90.641	149.397	937.945	83.279	619.083	166.472	43.659	56.562	
std	551.028	84.048	79.364	229.819	77.954	254.566	51.461	69.187	66.109	
min	1932.500	8.800	28.000	705.000	27.600	346.300	48.700	5.100	4.600	
25%	2377.700	37.400	98.400	793.500	51.600	421.100	137.500	11.600	15.500	
50%	2749.900	83.800	141.600	886.500	66.900	594.400	155.600	17.900	26.300	
75%	2992.500	107.200	168.400	954.200	85.900	679.900	192.700	57.200	71.300	
max	4260.700	456.700	448.300	1747.400	463.000	1452.000	278.500	369.400	273.400	

ID_REF	1431_at	...	AFFX-r2-Hs28SrRNA-M_at	AFFX-r2-P1-cre-3_at	AFFX-r2-P1-cre-5_at	\
count	29.000	...	29.000	29.000	29.000	
mean	77.903	...	233.003	51125.797	37013.586	
std	42.430	...	213.595	33210.059	23248.909	
min	9.300	...	74.000	15121.200	11495.400	
25%	58.700	...	162.100	36108.700	25760.200	
50%	70.300	...	197.000	41387.200	29820.400	
75%	95.400	...	240.300	54110.200	40469.600	
max	218.100	...	1283.900	194545.000	136124.000	

ID_REF	AFFX-ThrX-3_at	AFFX-ThrX-5_at	AFFX-ThrX-M_at	AFFX-TrpnX-3_at	AFFX-TrpnX-5_at	\
count	29.000	29.000	29.000	29.000	29.000	
mean	12.903	13.128	17.897	17.621	20.976	
std	11.092	17.248	14.956	15.698	17.731	
min	4.700	2.800	2.300	2.700	3.700	
25%	7.500	5.200	5.700	4.800	7.700	
50%	9.100	7.400	12.300	11.700	11.800	
75%	13.700	10.700	24.400	27.000	33.100	
max	61.400	91.200	62.300	57.000	58.300	

ID_REF	AFFX-TrpnX-M_at	class
count	29.000	29.000
mean	6.907	0.483
std	9.155	0.509
min	2.500	0.000
25%	3.100	0.000
50%	4.000	0.000
75%	6.400	1.000
max	45.500	1.000

Figure 4-9 **Statistical Description of Features (genes) on Breast Cancer Dataset GDS3139**

```
statistic descriptions:
         class     A1CF       A2M   A4GALT    A4GNT  AA053967  AA129909  AA243143  AA308853  \
count   52.000   52.000    52.000   52.000   52.000    52.000    52.000    52.000    52.000
mean     0.538  182.037  1349.606   62.915   92.277    11.002    14.838   700.444  1062.887
std      0.503  108.315   911.957   22.874   33.291    10.136    28.588   426.551   556.755
min      0.000    8.000   282.700   10.900   38.100     0.900     0.600   110.800   393.000
25%      0.000   85.100   670.400   47.825   69.775     3.700     3.850   412.175   699.500
50%      1.000  147.050  1125.500   63.850   90.450     6.600     8.100   620.050   973.850
75%      1.000  255.300  1806.625   76.850  108.900    13.425    15.225   871.725  1340.450
max      1.000  447.900  3986.700  111.400  207.800    40.400   201.300  1895.500  3035.000

        AA365670  ...  222296_at  222299_x_at  222323_at  222332_at  222338_x_at  222339_x_at  \
count     52.000  ...     52.000       52.000     52.000     52.000       52.000       52.000
mean     101.975  ...     15.742        6.950     24.987     13.831      311.167       61.531
std       51.189  ...     17.936        8.811     18.844     12.744       98.418       89.461
min       12.100  ...      3.000        0.100      4.100      1.100      182.700        3.800
25%       65.275  ...      6.000        1.000     11.225      3.800      252.525        8.800
50%       91.400  ...     10.250        3.250     15.400      8.850      290.250       36.350
75%      134.150  ...     17.925        9.050     37.350     23.575      341.850       69.600
max      247.400  ...     96.200       34.900     82.800     51.300      650.000      502.300

        222358_x_at  222370_x_at  222371_at  91682_at
count        52.000       52.000     52.000    52.000
mean        118.277      104.079    148.104   122.154
std          74.363       55.957    102.163    57.641
min          31.700       37.900     40.400    27.800
25%          66.925       69.100     85.350    77.750
50%          97.650       98.300    126.650   115.400
75%         151.225      116.250    193.150   151.425
max         439.000      363.100    670.100   275.600
```

Figure 4-10 **Statistical Description of Features (genes) on Cervical Cancer Dataset GDS3233**

```
statistic descriptions:
 ID_REF  1552563_a_at  1552829_at  1552867_at  1552961_at  1552974_at  1552975_x_at  1553069_at  \
count         52.000       52.000      52.000      52.000      52.000        52.000      52.000
mean          12.006       12.844      94.002      15.807      12.234        12.498       4.163
std            4.787        5.161      48.038       5.750       4.287         4.704       0.227
min            5.890        7.040      33.860      10.830       8.660         8.720       3.880
25%            9.197       10.120      68.933      12.238       9.683         9.908       4.018
50%           10.535       10.915      82.065      12.950      10.345        10.420       4.090
75%           14.510       13.563      99.102      17.595      13.455        13.610       4.280
max           35.530       36.930     250.600      36.560      30.450        34.580       4.960

ID_REF  1553083_at  1553275_s_at  1553354_a_at  ...  91580_at  91617_at  91682_at  91684_g_at  \
count       52.000        52.000        52.000  ...    52.000    52.000    52.000      52.000
mean         4.220        41.602        10.106  ...     7.031    49.137     8.790      40.594
std          0.452        24.827         2.785  ...     1.275    14.803     2.052      10.390
min          3.860         6.000         7.230  ...     5.880    23.260     6.770      23.300
25%          3.968        25.308         8.550  ...     6.310    39.492     7.570      34.590
50%          4.050        40.375         8.940  ...     6.470    46.320     8.105      38.615
75%          4.380        59.068        10.450  ...     7.102    52.753     9.390      45.785
max          6.440       101.300        19.630  ...    12.410    96.830    16.880      68.550

ID_REF  91703_at  91816_f_at  91826_at  91920_at  91952_at   class
count     52.000      52.000    52.000    52.000    52.000  52.000
mean      71.051      29.299   315.782    10.044    29.862   0.692
std       41.890      25.337   388.354     1.530    10.124   0.466
min        7.560       4.340    10.150     7.700    12.360   0.000
25%       39.160      12.652    75.060     9.175    22.807   0.000
50%       71.655      24.915   205.550     9.800    28.110   1.000
75%      104.800      35.028   397.725    10.297    36.225   1.000
max      159.000     132.900  2390.000    15.170    58.470   1.000
```

Figure 4-11 **Statistical Description of Features (genes) on Pancreatic Cancer Dataset GDS4102**

```
statistic descriptions:
 ID_REF  1007_s_at  1053_at  117_at  121_at  1255_g_at  1294_at  1316_at  1320_at  1405_i_at  \
count      64.000   64.000  64.000  64.000     64.000   64.000   64.000   64.000     64.000
mean        3.143    6.601   6.019   7.287      2.250    7.779    4.688    2.789      7.122
std         0.211    0.510   2.308   0.328      0.074    0.573    0.250    0.132      2.373
min         2.739    5.707   3.924   6.666      2.069    6.707    4.184    2.483      3.125
25%         3.016    6.309   4.256   7.094      2.214    7.411    4.568    2.712      4.803
50%         3.105    6.500   4.840   7.224      2.247    7.690    4.651    2.777      7.399
75%         3.228    6.811   7.488   7.529      2.287    8.134    4.732    2.870      9.471
max         4.190    8.143  12.358   8.221      2.512    9.531    5.479    3.174     11.425

ID_REF  1431_at  ...  AFFX-r2-Hs28SrRNA-M_at  AFFX-r2-P1-cre-3_at  AFFX-r2-P1-cre-5_at  \
count    64.000  ...                  64.000               64.000               64.000
mean      3.181  ...                   5.638               14.475               13.837
std       0.196  ...                   0.489                0.350                0.353
min       2.829  ...                   3.858               13.791               13.248
25%       3.058  ...                   5.584               14.216               13.607
50%       3.170  ...                   5.680               14.410               13.790
75%       3.278  ...                   5.919               14.734               14.054
max       4.040  ...                   6.507               15.617               14.881

ID_REF  AFFX-ThrX-3_at  AFFX-ThrX-5_at  AFFX-ThrX-M_at  AFFX-TrpnX-3_at  AFFX-TrpnX-5_at  \
count           64.000          64.000          64.000           64.000           64.000
mean             2.355           3.351           2.991            3.026            2.357
std              0.076           0.101           0.130            0.152            0.080
min              2.146           3.164           2.597            2.651            2.144
25%              2.311           3.300           2.920            2.925            2.308
50%              2.350           3.339           2.991            3.029            2.349
75%              2.388           3.398           3.059            3.103            2.389
max              2.658           3.717           3.307            3.404            2.561

ID_REF  AFFX-TrpnX-M_at   class
count           64.000  64.000
mean             3.051   0.406
std              0.106   0.495
min              2.855   0.000
25%              2.994   0.000
50%              3.049   0.000
75%              3.099   1.000
max              3.352   1.000
```

Figure 4-12 **Statistical Description of Features (genes) on AML Cancer Dataset GDS3057**

4.2.5 Class Distribution

On classification problems we need to know how balanced the class values are. Highly imbalanced problems (a lot more observations for one class than another) are common and may need special handling in the data preparation stage of our project. (Brownlee)

```
 class
0.0    16
1.0    36
```

Figure 4-15 **Class Distribution of Pancreatic Cancer Dataset GDS4102.**

16 healthy donors are labeled with zero value in "class" and 36 patients are labeled with one value.

```
 class
0.0    15
1.0    14
```

Figure 4-14 **Class Distribution of Breast Cancer Dataset GDS3139.**

15 healthy donors are labeled with zero value in "class" and 14 patients are labeled with one value.

```
 class
0.0    38
1.0    26
```

Figure 4-13 **Class Distribution of AML Cancer Dataset GDS3057.**

38 healthy donors are labeled with zero value in "class" and 26 patients are labeled with one value.

```
 class
0    24
1    28
```

Figure 4-16 **Class Distribution of Cervical Cancer Dataset GDS3233.**

24 healthy donors are labeled with zero value in "class" and 28 patients are labeled with one value.

As we can observe here, we have datasets with almost equal number of sample in the two classes (Cervical and Breast Cancer Datasets) and we have the imbalanced distributions with almost double cancer versus the healthy samples like Pancreatic Dataset or otherwise like AML Dataset.

4.2.6 Skew of Univariate Distributions

Skew refers to a distribution that is assumed Gaussian (normal or bell curve) that is shifted or squashed in one direction or another. Many machine learning algorithms assume a Gaussian distribution. Knowing that an attribute has a skew may allow us to perform data preparation to correct the skew and later improve the accuracy of your models. (Brownlee)

skew	
ID_REF	
1552563_a_at	2.703
1552829_at	2.582
1552867_at	1.867
1552961_at	1.915
1552974_at	2.272
1552975_x_at	2.636
1553069_at	1.498
1553083_at	3.014
1553275_s_at	0.535
1553354_a_at	2.232
1553355_at	3.197
1553356_at	1.514
1553372_at	3.008
1553398_at	1.723
1553439_at	2.605
1553456_at	1.114
1553462_at	2.312
1553475_at	2.636
1553498_at	2.611
1553546_at	2.011
1553547_at	5.357
1553881_at	1.797
1554007_at	0.710
1554232_a_at	1.752
1554281_at	1.831
1554372_at	2.207
1554374_at	0.928
1554404_a_at	1.533
A1BG	2.335
A1BG-AS1	1.098
...	
65884_at	0.436
66053_at	1.774
71933_at	4.490
74694_s_at	0.022
76897_s_at	0.955
77508_r_at	1.900
78047_s_at	1.994
78330_at	3.128
78383_at	2.098
78495_at	1.855
79005_at	2.370
81737_at	1.048
81811_at	0.988
823_at	2.459
87100_at	1.137
89476_r_at	0.766
89948_at	1.603
89977_at	2.918
90265_at	1.375
90610_at	2.902
91580_at	2.317
91617_at	1.382
91682_at	2.308
91684_g_at	0.864
91703_at	0.065
91816_f_at	2.263
91826_at	3.304
91920_at	1.523
91952_at	0.794
class	-0.858

Figure 4-17 **Features Skew of Pancreatic Cancer Dataset GDS4102**

skew	
ID_REF	
1007_s_at	2.075
1053_at	1.023
117_at	1.098
121_at	0.536
1255_g_at	0.405
1294_at	0.440
1316_at	1.122
1320_at	0.383
1405_i_at	-0.066
1431_at	1.376
1438_at	6.129
1487_at	0.740
1494_f_at	2.430
1598_g_at	0.869
160020_at	0.984
1729_at	0.523
1773_at	3.853
177_at	1.216
179_at	1.225
1861_at	2.615
200000_s_at	-0.419
200001_at	0.358
200002_at	-1.351
200003_s_at	-1.886
200004_at	0.466
200005_at	-0.299
200006_at	-1.259
200007_at	-0.254
200008_s_at	-0.542
200009_at	-0.531
...	
AFFX-r2-Bs-lys-5_at	0.584
AFFX-r2-Bs-lys-M_at	0.019
AFFX-r2-Bs-phe-3_at	2.163
AFFX-r2-Bs-phe-5_at	-0.297
AFFX-r2-Bs-phe-M_at	1.265
AFFX-r2-Bs-thr-3_s_at	0.328
AFFX-r2-Bs-thr-5_s_at	2.596
AFFX-r2-Bs-thr-M_s_at	-0.129
AFFX-r2-Ec-bioB-3_at	0.237
AFFX-r2-Ec-bioB-5_at	0.451
AFFX-r2-Ec-bioB-M_at	0.576
AFFX-r2-Ec-bioC-3_at	0.249
AFFX-r2-Ec-bioC-5_at	0.609
AFFX-r2-Ec-bioD-3_at	0.375
AFFX-r2-Ec-bioD-5_at	0.343
AFFX-r2-Hs18SrRNA-3_s_at	0.064
AFFX-r2-Hs18SrRNA-5_at	0.764
AFFX-r2-Hs18SrRNA-M_x_at	1.531
AFFX-r2-Hs28SrRNA-3_at	-0.253
AFFX-r2-Hs28SrRNA-5_at	-0.007
AFFX-r2-Hs28SrRNA-M_at	-1.168
AFFX-r2-P1-cre-3_at	0.633
AFFX-r2-P1-cre-5_at	0.491
AFFX-ThrX-3_at	0.883
AFFX-ThrX-5_at	0.970
AFFX-ThrX-M_at	-0.061
AFFX-TrpnX-3_at	0.181
AFFX-TrpnX-5_at	0.372
AFFX-TrpnX-M_at	0.418

Figure 4-18 **Features Skew of AML Cancer Dataset GDS3057**

skew	
ID_REF	
1007_s_at	1.019
1053_at	3.125
117_at	2.015
121_at	2.161
1255_g_at	4.429
1294_at	2.049
1316_at	0.084
1320_at	3.984
1405_i_at	2.054
1431_at	1.340
1438_at	2.018
1487_at	-0.263
1494_f_at	2.328
1598_g_at	0.942
160020_at	3.480
1729_at	1.254
1773_at	0.475
177_at	2.135
179_at	1.768
1861_at	0.628
200000_s_at	0.772
200001_at	0.012
200002_at	2.527
200003_s_at	2.990
200004_at	-0.957
200005_at	1.629
200006_at	-0.091
200007_at	0.055
200008_s_at	1.050
200009_at	0.607
...	
AFFX-r2-Bs-lys-5_at	1.554
AFFX-r2-Bs-lys-M_at	0.847
AFFX-r2-Bs-phe-3_at	1.453
AFFX-r2-Bs-phe-5_at	1.925
AFFX-r2-Bs-phe-M_at	1.369
AFFX-r2-Bs-thr-3_s_at	2.239
AFFX-r2-Bs-thr-5_s_at	1.042
AFFX-r2-Bs-thr-M_s_at	0.859
AFFX-r2-Ec-bioB-3_at	3.759
AFFX-r2-Ec-bioB-5_at	3.463
AFFX-r2-Ec-bioB-M_at	3.115
AFFX-r2-Ec-bioC-3_at	2.337
AFFX-r2-Ec-bioC-5_at	1.848
AFFX-r2-Ec-bioD-3_at	3.491
AFFX-r2-Ec-bioD-5_at	2.950
AFFX-r2-Hs18SrRNA-3_s_at	1.732
AFFX-r2-Hs18SrRNA-5_at	1.858
AFFX-r2-Hs18SrRNA-M_x_at	1.069
AFFX-r2-Hs28SrRNA-3_at	4.557
AFFX-r2-Hs28SrRNA-5_at	0.959
AFFX-r2-Hs28SrRNA-M_at	4.517
AFFX-r2-P1-cre-3_at	3.052
AFFX-r2-P1-cre-5_at	2.948
AFFX-ThrX-3_at	3.307
AFFX-ThrX-5_at	3.704
AFFX-ThrX-M_at	1.338
AFFX-TrpnX-3_at	1.170
AFFX-TrpnX-5_at	0.999
AFFX-TrpnX-M_at	3.567
class	0.073

Figure 4-19 **Features Skew of Breast Cancer Dataset GDS3139**

skew			
class	-0.159	...	
A1CF	0.649	ZWILCH	1.197
A2M	1.307	ZWINT	1.239
A4GALT	-0.119	ZXDB	1.387
A4GNT	0.979	ZXDC	1.327
AA053967	1.478	ZYX	1.418
AA129909	5.748	ZZEF1	-0.260
AA243143	1.228	ZZZ3	0.716
AA308853	1.671	221798_x_at	-0.147
AA365670	0.793	221852_at	2.602
AA393940	1.676	221929_at	0.752
AA447740	3.381	221955_at	1.307
AA502643	0.296	221965_at	1.658
AA521034	1.248	222050_at	1.619
AA621286	1.505	222097_at	0.671
AA732995	2.218	222098_s_at	0.848
AA853175	1.556	222178_s_at	1.937
AA923707	1.105	222229_x_at	0.841
AAAS	0.890	222254_at	3.415
AACS	0.426	222288_at	1.091
AADAC	5.051	222295_x_at	1.248
AAGAB	1.263	222296_at	3.168
AAK1	2.010	222299_x_at	1.708
AAMDC	1.049	222323_at	1.057
AAMP	0.327	222332_at	1.062
AANAT	4.119	222338_x_at	1.555
AAR2	0.820	222339_x_at	3.224
AARS	1.218	222358_x_at	2.034
AASDHPPT	0.270	222370_x_at	2.343
AASS	2.256	222371_at	2.767
		91682_at	0.652

Figure 4-20 **Features Skew of Cervical Cancer Dataset GDS3233**

As we can conclude the features distribution are skewed. For models like Logistic Regression that assumes Gaussian distribution we need to generate a normal symmetric distribution. For fixing this, we can apply a Data Transformation technique like **Standardization** (Z-score normalization) overcomes the problem of variables with different units as it transforms variables so that they are centered on 0 with a standard deviation of 1.

4.3 Data Visualization

4.3.1 Histograms

A fast way to get an idea of the distribution of each attribute is to look at histograms. Histograms group data into bins and provide you a count of the number of observations in each bin. From the shape of the bins you can quickly get a feeling for whether an attribute is Gaussian, skewed or even has an exponential distribution. It can also help you see possible outliers. (Brownlee)

As we took a first idea of the statistical analysis of skew on each feature, now we will visualize some features histogram of our datasets. Due to the large computational power it takes, in order to produce them we used Orange3. It is open source machine learning and data visualization tool provided in Anaconda that has an interactive graphical environment for machine learning. (Orange3)

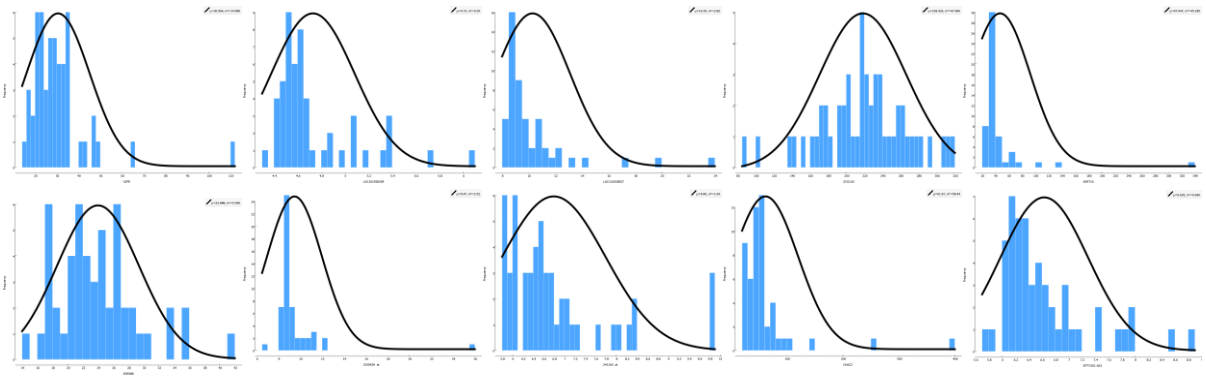


Figure 4-21 **Histogram Distributions of 10 different features on Pancreatic Cancer Dataset GDS4102**

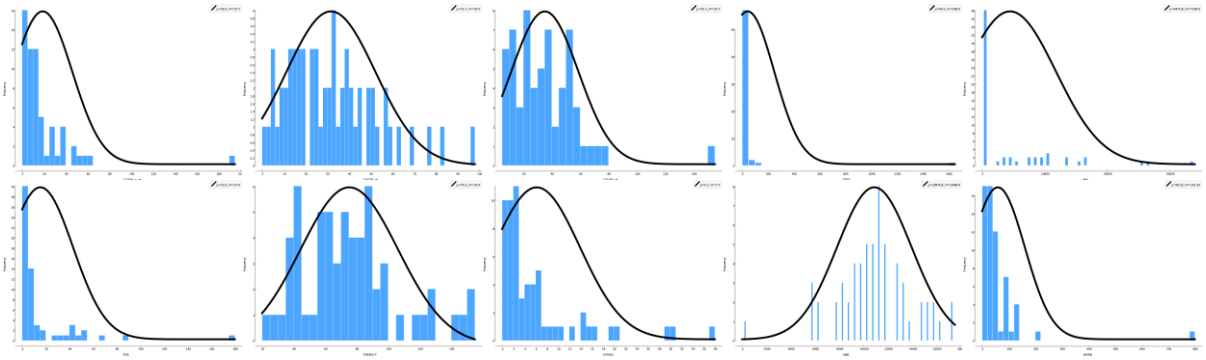


Figure 4-22 **Histogram Distributions of 10 different features on Cervical Cancer Dataset GDS3233**

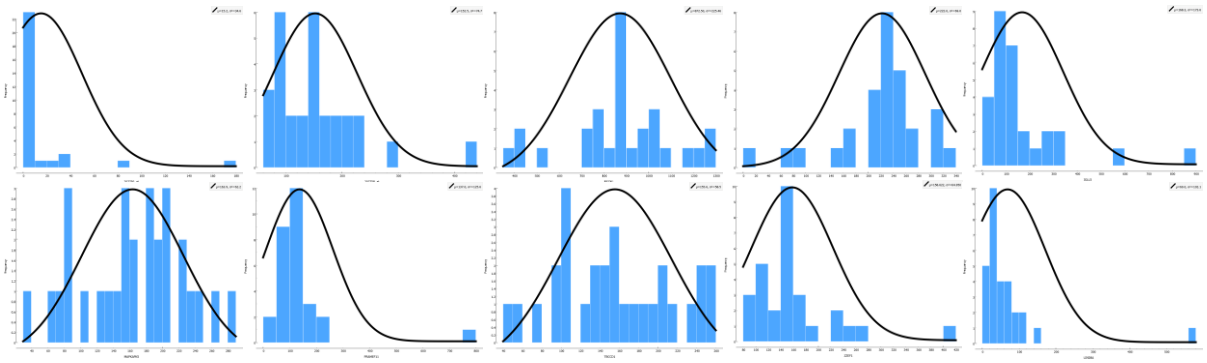


Figure 4-23 **Histogram Distributions of 10 different features on Breast Cancer Dataset GDS3139**

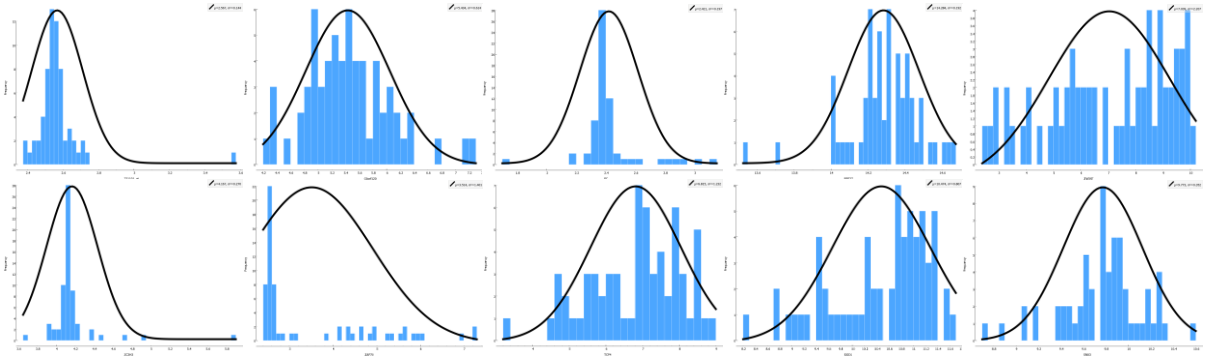


Figure 4-24 **Histogram Distributions of 10 different features on AML Cancer Dataset GDS3057**

As we can observe from the histograms the data features distribution differs. In most cased the data are skewed right or left like assuming an exponential distribution. Some others but only few of them in total are symmetric assuming a Gaussian or nearly Gaussian distribution. This information must be noted, because many machine learning techniques assume a Gaussian univariate distribution on the input variables. So, this leads us to apply a data transformation technique like Standardization, which transform attributes to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

4.3.2 Correlation Matrix Plot

Correlation refers to the relationship between two variables and how they may or may not change together. The most common method for calculating correlation is Pearson's Correlation Coefficient that assumes a normal distribution of the attributes involved. A correlation of -1 or 1 shows a full negative or positive correlation respectively. Whereas a value of 0 shows no correlation at all. Some machine learning algorithms like linear and logistic regression can suffer poor performance if there are highly correlated attributes in your dataset. The matrix lists all attributes across the top and down the side, to give correlation between all pairs of attributes (twice, because the matrix is symmetrical). We can see the diagonal line through the matrix from the top left to bottom right corners of the matrix shows perfect correlation of each attribute with itself. (Brownlee)

This task needed high RAM space in order to find thousands of data correlation. To run this part of code, we used the Google Colaboratory. It is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. With Colaboratory we can write and execute code, save and share our analyses, and access powerful computing resources, all for free from our browser. (Google Colaboratory)

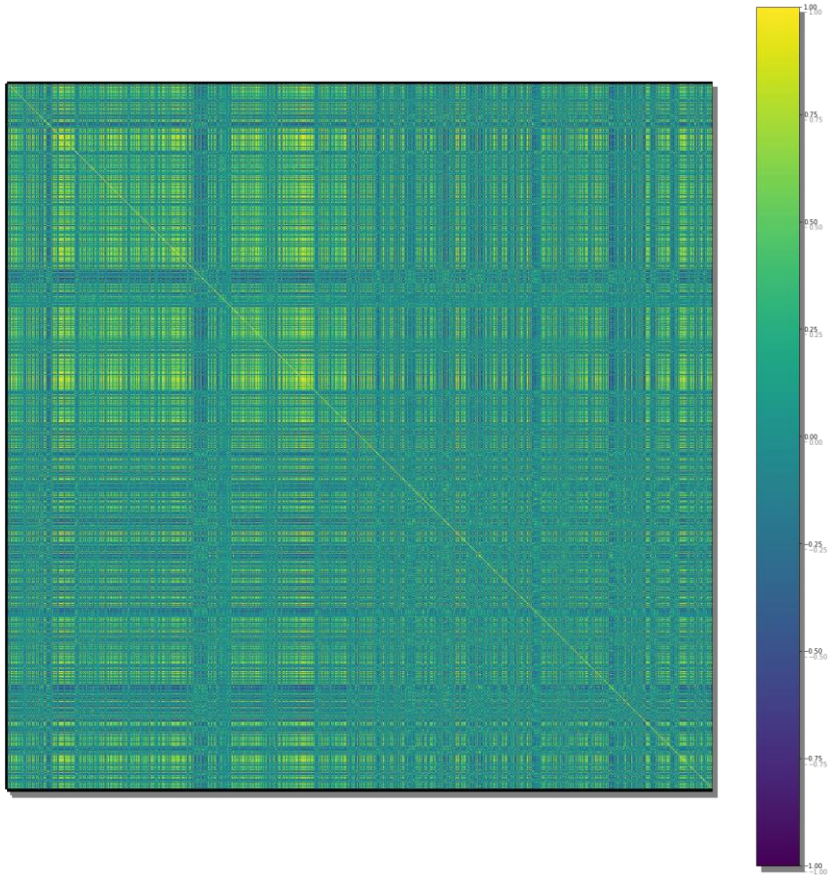


Figure 4-25 **Pearson's Correlation Plot of Pancreatic Cancer Dataset GDS4102 with 54614 features**

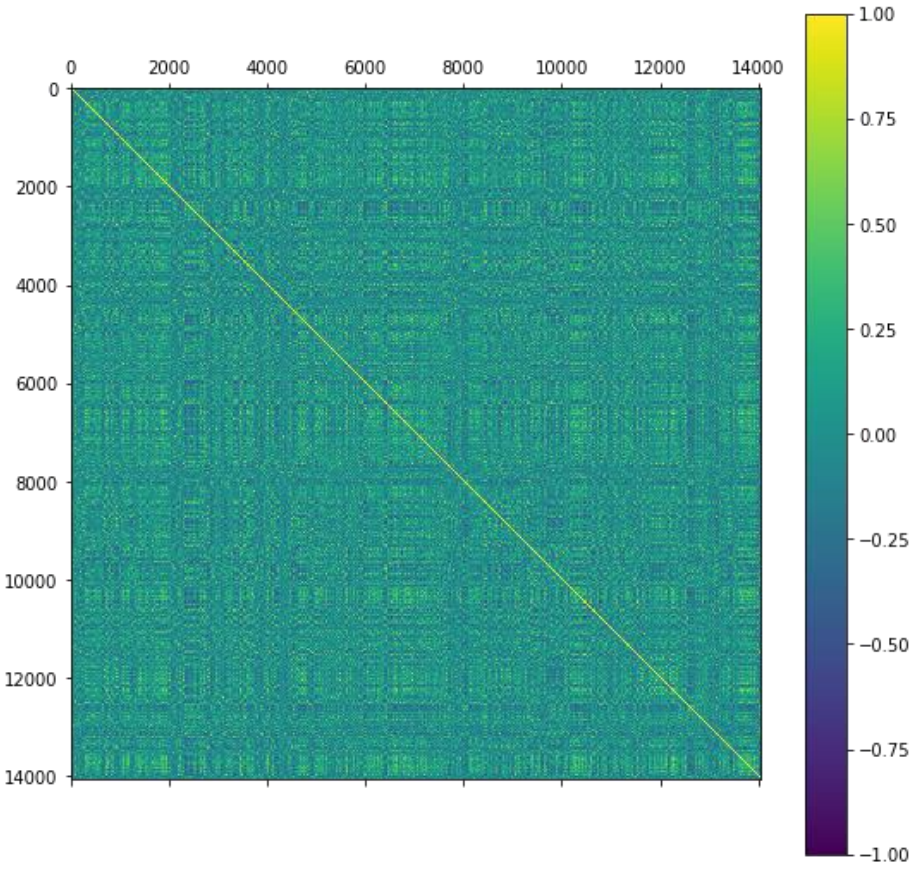


Figure 4-26 **Pearson's Correlation Plot of Cervical Cancer Dataset GDS3233 with 14063 features**

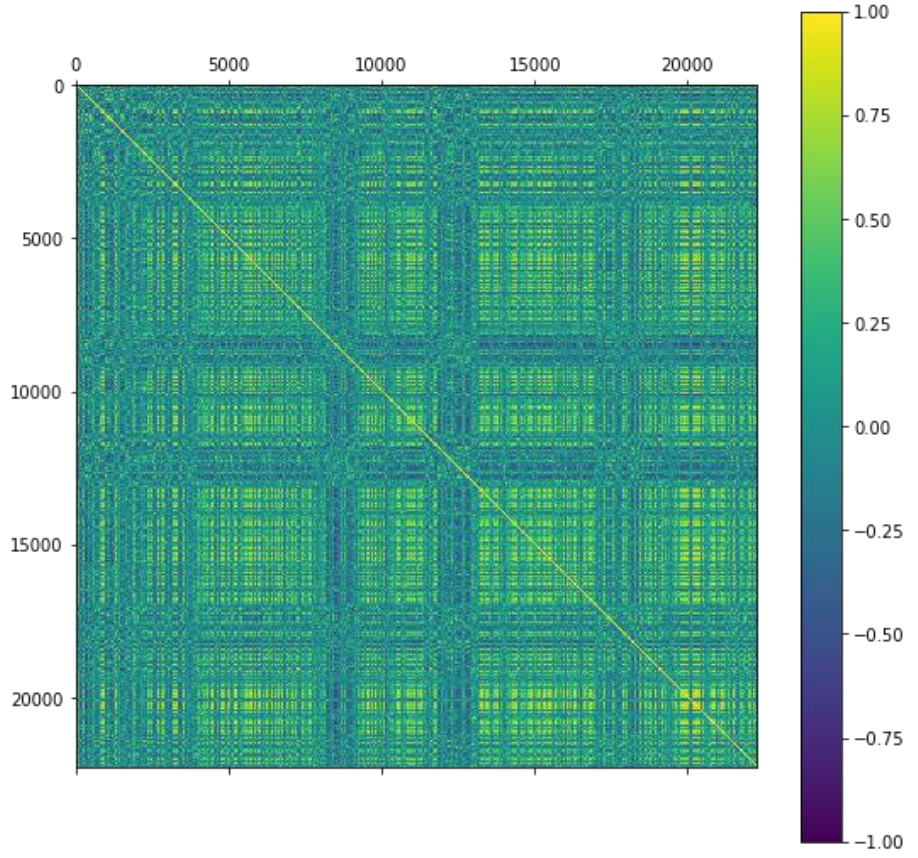


Figure 4-27 **Pearson's Correlation Plot of Breast Cancer Dataset GDS3139 with 14063 features**

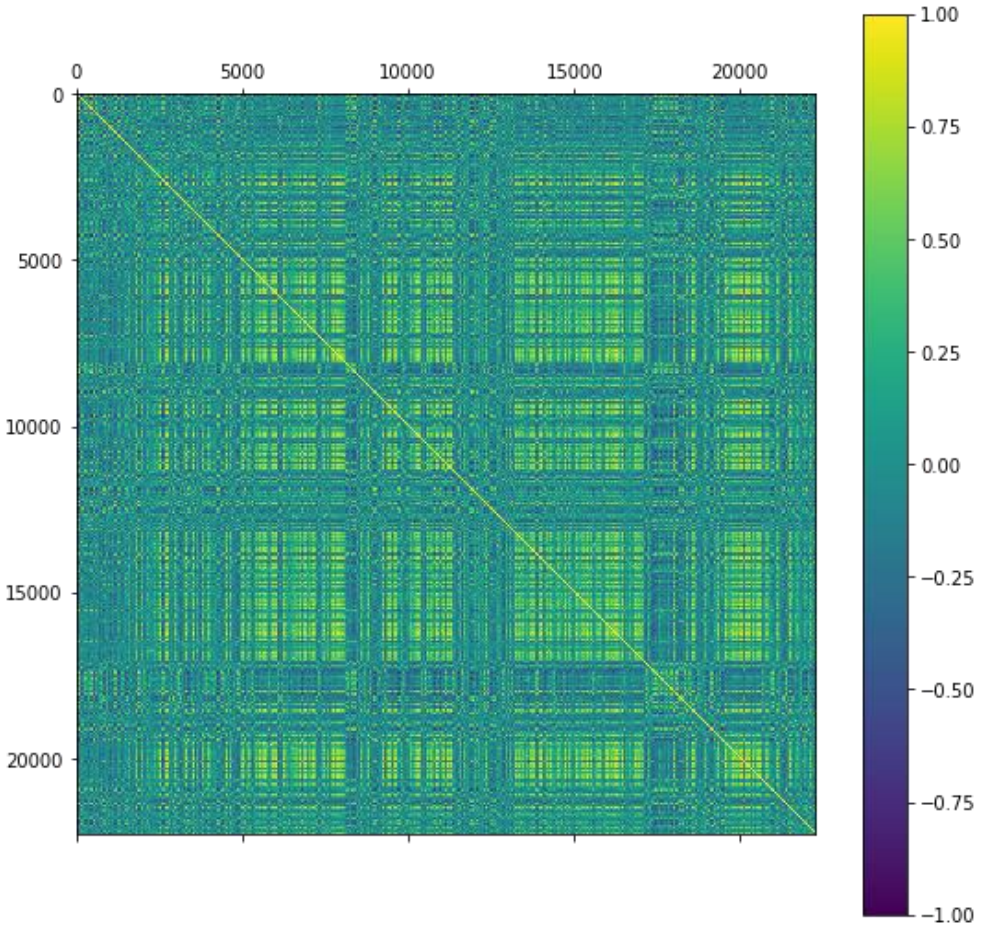


Figure 4-28 **Pearson's Correlation Plot of AML Cancer Dataset GDS3057 with 22284 features**

4.4 Data Analysis and Preparation

4.4.1 Validation Dataset

As it is mentioned on second chapter, it is a good idea to use a validation hold-out set. This is a sample of the data that we hold back from our analysis and modeling. We use it right at the end of our project to confirm the accuracy of our final model. It is a smoke test that we can use to see if we messed up and to give us confidence on our estimates of accuracy on unseen data. (Brownlee) In our study the 4 datasets will be split randomly in **train** and **validation** data with **67%** and **33%** of total samples each. X array is the data values (gene expression levels) and the Y array is the binary label (cancer or healthy class) for the supervised classification process.

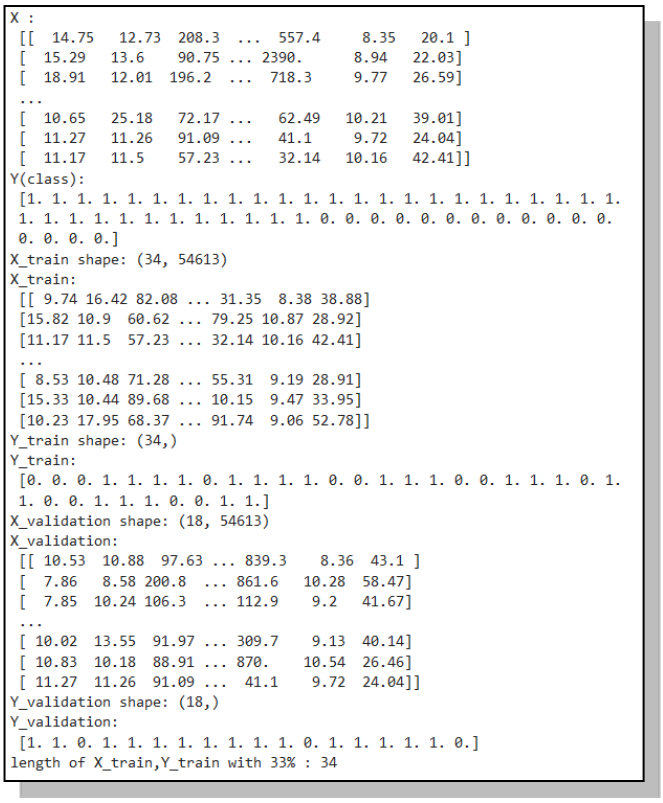


Figure 4-30 **GDS4102 split**
X_array stores data values Y_array stores the labels
X_train (34 x 54,613) for modeling Y_train (34 x 1)
X_validation (18 x 54,613) held out data for predictions and final evaluation Y_validation (18 x 1)

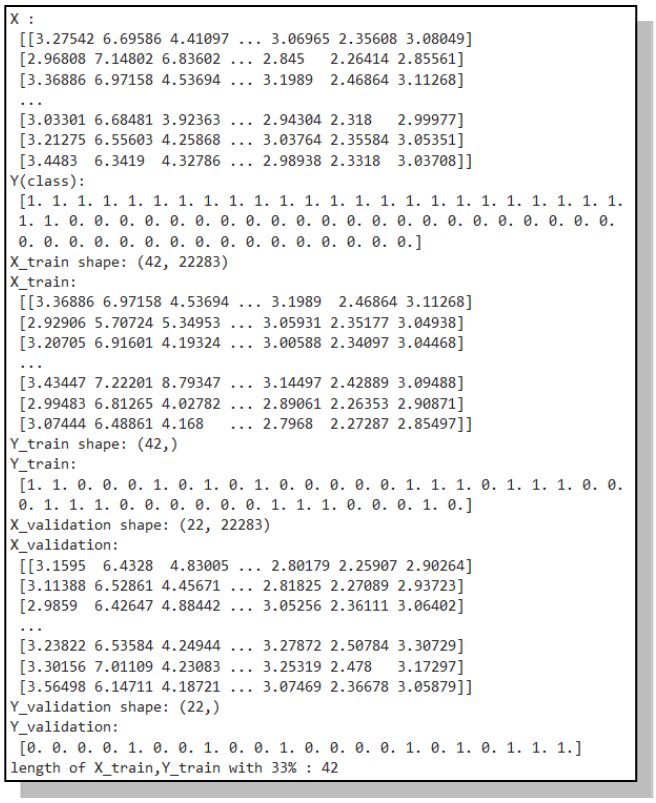
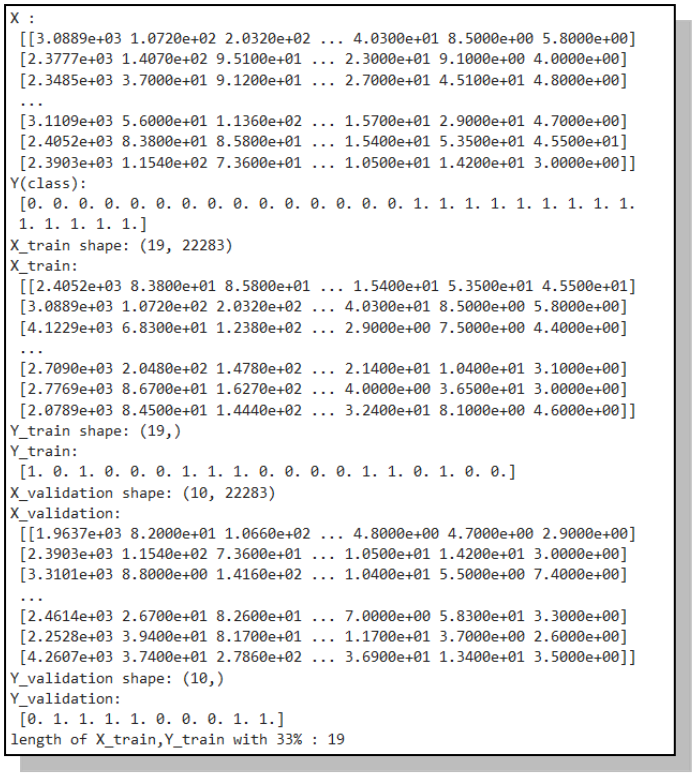
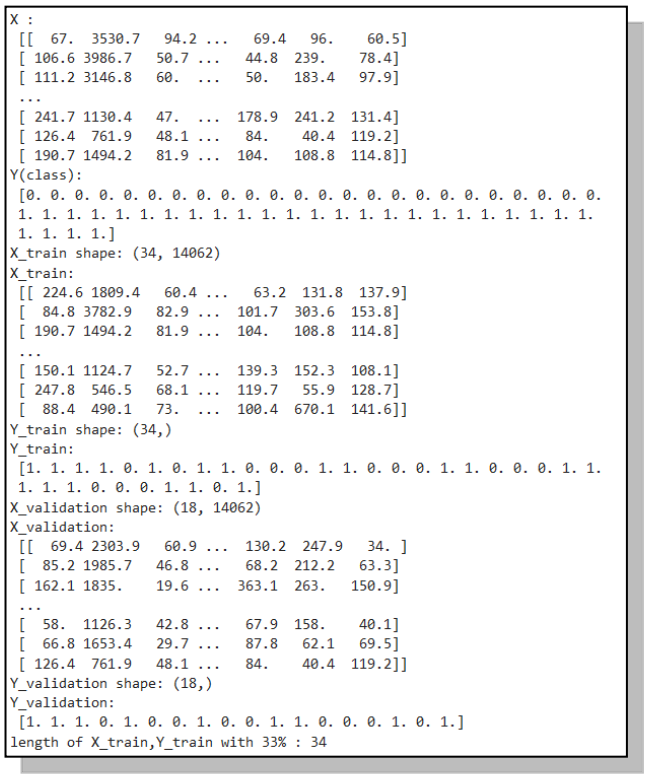


Figure 4-29 **GDS3057 split**
X_array stores data values Y_array stores the labels
X_train (42 x 22,283) for modeling Y_train (42 x 1)
X_validation (22 x 22,283) held out data for predictions and final evaluation Y_validation (22 x 1)



```
X :
 [[3.0889e+03 1.0720e+02 2.0320e+02 ... 4.0300e+01 8.5000e+00 5.8000e+00]
 [2.3777e+03 1.4070e+02 9.5100e+01 ... 2.3000e+01 9.1000e+00 4.0000e+00]
 [2.3485e+03 3.7000e+01 9.1200e+01 ... 2.7000e+01 4.5100e+01 4.8000e+00]
 ...
 [3.1109e+03 5.6000e+01 1.1360e+02 ... 1.5700e+01 2.9000e+01 4.7000e+00]
 [2.4052e+03 8.3800e+01 8.5800e+01 ... 1.5400e+01 5.3500e+01 4.5500e+01]
 [2.3903e+03 1.1540e+02 7.3600e+01 ... 1.0500e+01 1.4200e+01 3.0000e+00]]
Y(class):
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1.]
X_train shape: (19, 22283)
X_train:
 [[2.4052e+03 8.3800e+01 8.5800e+01 ... 1.5400e+01 5.3500e+01 4.5500e+01]
 [3.0889e+03 1.0720e+02 2.0320e+02 ... 4.0300e+01 8.5000e+00 5.8000e+00]
 [4.1229e+03 6.8300e+01 1.2380e+02 ... 2.9000e+00 7.5000e+00 4.4000e+00]
 ...
 [2.7090e+03 2.0480e+02 1.4780e+02 ... 2.1400e+01 1.0400e+01 3.1000e+00]
 [2.7769e+03 8.6700e+01 1.6270e+02 ... 4.0000e+00 3.6500e+01 3.0000e+00]
 [2.0789e+03 8.4500e+01 1.4440e+02 ... 3.2400e+01 8.1000e+00 4.6000e+00]]
Y_train shape: (19,)
Y_train:
 [1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0.]
X_validation shape: (10, 22283)
X_validation:
 [[1.9637e+03 8.2000e+01 1.0660e+02 ... 4.8000e+00 4.7000e+00 2.9000e+00]
 [2.3903e+03 1.1540e+02 7.3600e+01 ... 1.0500e+01 1.4200e+01 3.0000e+00]
 [3.3101e+03 8.8000e+00 1.4160e+02 ... 1.0400e+01 5.5000e+00 7.4000e+00]
 ...
 [2.4614e+03 2.6700e+01 8.2600e+01 ... 7.0000e+00 5.8300e+01 3.3000e+00]
 [2.2528e+03 3.9400e+01 8.1700e+01 ... 1.1700e+01 3.7000e+00 2.6000e+00]
 [4.2607e+03 3.7400e+01 2.7860e+02 ... 3.6900e+01 1.3400e+01 3.5000e+00]]
Y_validation shape: (10,)
Y_validation:
 [0. 1. 1. 1. 1. 0. 0. 0. 1. 1.]
length of X_train,Y_train with 33% : 19
```

Figure 4-32 **GDS3139 split**
X_array stores data values Y_array stores the labels
X_train (19 x 22,283) for modeling Y_train (19 x 1)
X_validation (10 x 22,283) held out data for predictions and final evaluation Y_validation (10 x 1)



```
X :
 [[  67.  3530.7   94.2 ...   69.4   96.    60.5]
 [ 106.6 3986.7   50.7 ...   44.8  239.    78.4]
 [ 111.2 3146.8   60.  ...   50.   183.4   97.9]
 ...
 [ 241.7 1130.4   47.  ...  178.9  241.2  131.4]
 [ 126.4  761.9   48.1 ...   84.    40.4  119.2]
 [ 190.7 1494.2   81.9 ...  104.   108.8  114.8]]
Y(class):
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1.]
X_train shape: (34, 14062)
X_train:
 [[ 224.6 1809.4   60.4 ...   63.2  131.8  137.9]
 [  84.8 3782.9   82.9 ...  101.7  303.6  153.8]
 [ 190.7 1494.2   81.9 ...  104.   108.8  114.8]
 ...
 [ 150.1 1124.7   52.7 ...  139.3  152.3  108.1]
 [ 247.8  546.5   68.1 ...  119.7   55.9  128.7]
 [  88.4  490.1   73.  ...  100.4  670.1  141.6]]
Y_train shape: (34,)
Y_train:
 [1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1.
 1. 1. 1. 0. 0. 0. 1. 1. 0. 1.]
X_validation shape: (18, 14062)
X_validation:
 [[  69.4 2303.9   60.9 ...  130.2  247.9   34. ]
 [  85.2 1985.7   46.8 ...   68.2  212.2   63.3]
 [ 162.1 1835.    19.6 ...  363.1  263.   150.9]
 ...
 [  58.  1126.3   42.8 ...   67.9  158.    40.1]
 [  66.8 1653.4   29.7 ...   87.8   62.1   69.5]
 [ 126.4  761.9   48.1 ...   84.    40.4  119.2]]
Y_validation shape: (18,)
Y_validation:
 [1. 1. 1. 0. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1.]
length of X_train,Y_train with 33% : 34
```

Figure 4-31 **GDS3233 split**
X_array stores data values Y_array stores the labels
X_train (34 x 14,062) for modeling Y_train (34 x 1)
X_validation (18 x 14,062) held out data for predictions and final evaluation Y_validation (18 x 1)

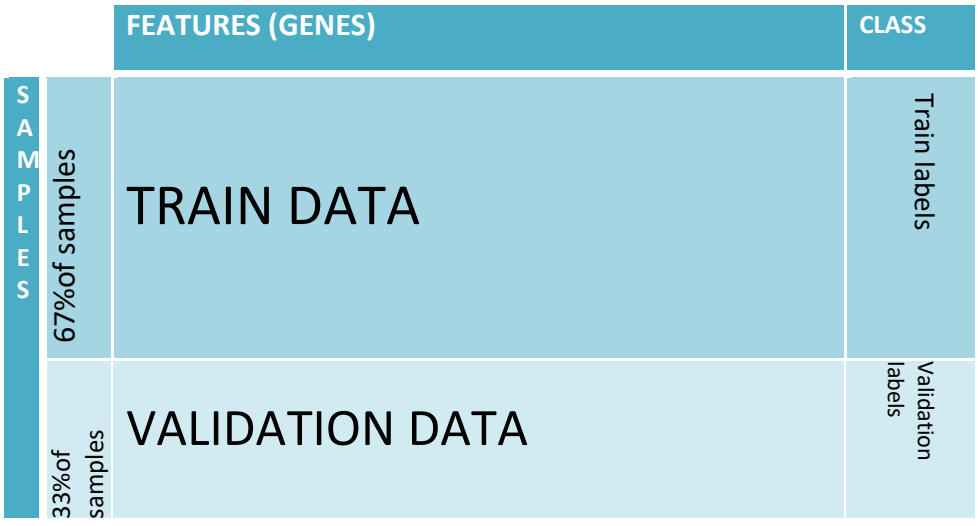


Table 4 **Overview of Datasets Split on Train and Validation Data**

4.5 Implementation of different Machine Learning Algorithms: Training and Evaluation

In this section, we evaluate the performance of different Machine Learning Algorithms in 3 scenarios: on baseline level, on standardized data and last, with the use of PCA for feature extraction on standardized data. The evaluation is made in two stages. The most important evaluation metric is accuracy which gives us a first insight on how correct our models are in this binary classification problem. First, we measure the accuracy with **k-fold Cross-Validation** on the train dataset to see a first picture on our models. The second and last stage is to **make predictions** on unseen data, using the held out validation dataset. As a result, it sums up to a final accuracy score, a confusion matrix and a classification report.

The Classification Algorithms

In order to find the best performance on our data we have to experiment with different algorithms. Both linear and non-linear algorithms are selected for this problem. The algorithms all use default tuning parameters. The suite of six algorithms is

- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- Classification and Regression Trees (CART)
- Linear Support Vector Machines (SVM)
- Gaussian Naive Bayes (NB)
- k-Nearest Neighbors (KNN)

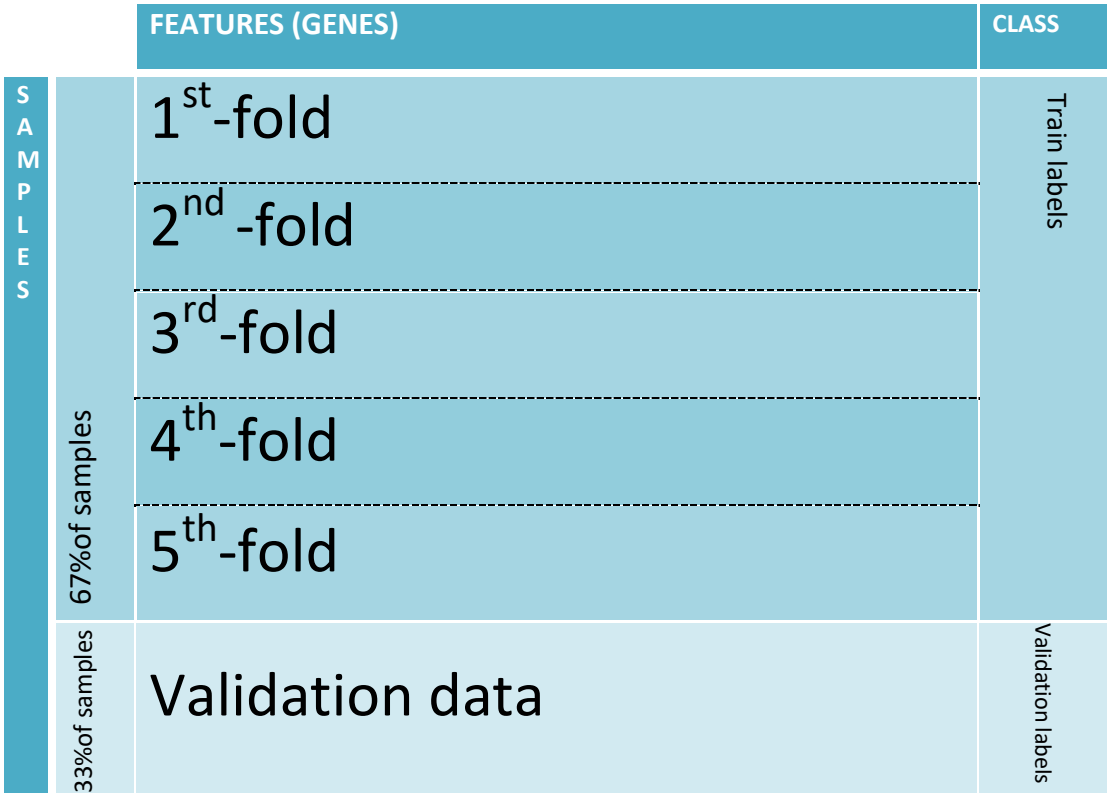


Table 5 **Overview of Datasets Structure with 5-Fold Cross-Validation and Split on Train and Validation Data**

4.6 Algorithm Evaluation: Baseline

4.6.1 Cross-Validation Results

The choice of k must allow the size of each test partition to be large enough to be a reasonable sample of the problem, whilst allowing enough repetitions of the train-test evaluation of the algorithm to provide a fair estimate of the algorithms performance on unseen data. For modest sized datasets in the thousands or tens of thousands of records, k values of 3, 5 and 10 are common. (Brownlee)

In our data we use 5-fold cross-validation. We compare the algorithms, by displaying the mean and standard deviation of accuracy, on training dataset with 5-fold cross-validation, for each algorithm as we calculate it and collect the results for use later.

```
LR Accuracy: 0.9380952380952381 (  0.0761904761904762 )
LDA Accuracy: 0.8523809523809524 (  0.09085135251589958 )
KNN Accuracy: 0.9142857142857143 (  0.06998542122237654 )
CART Accuracy: 0.8428571428571429 (  0.18294640678379567 )
NB Accuracy: 0.9095238095238095 (  0.07438333024673005 )
SVM Accuracy: 0.9095238095238095 (  0.07438333024673005 )
```

Figure 4-33 **Cervical Cancer Dataset GDS3233 5-fold Cross-Validation Accuracy Results**

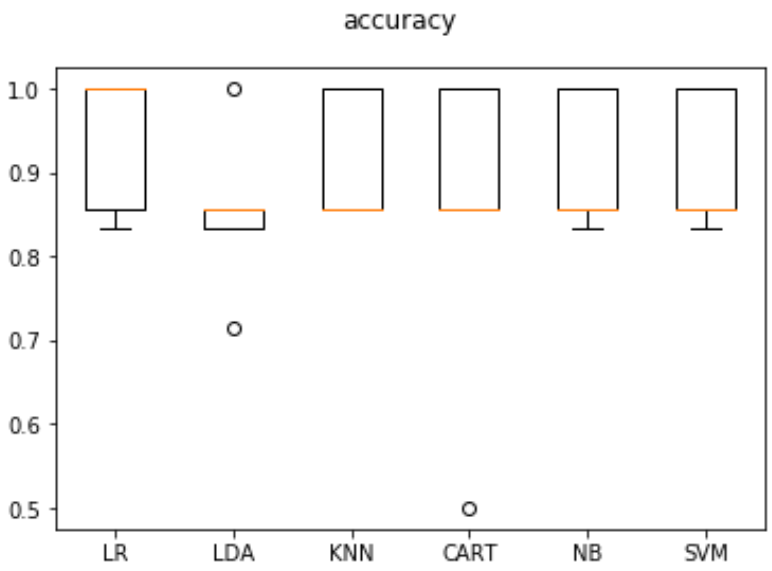


Figure 4-34 **Cervical Cancer Dataset GDS3233**
Graphical representation of 5-fold Cross-Validation Accuracy Results using box and whisker plots

```
LR Accuracy: 0.5666666666666667 (  0.27588242262078083 )
LDA Accuracy: 0.5166666666666667 (  0.13333333333333336 )
KNN Accuracy: 0.3 (  0.18708286933869708 )
CART Accuracy: 0.5 (  0.3535533905932738 )
NB Accuracy: 0.5666666666666667 (  0.16158932858054434 )
SVM Accuracy: 0.6333333333333333 (  0.2505549396395485 )
```

Figure 4-35 **Breast Cancer Dataset GDS3139 5-fold Cross-Validation Accuracy Results**

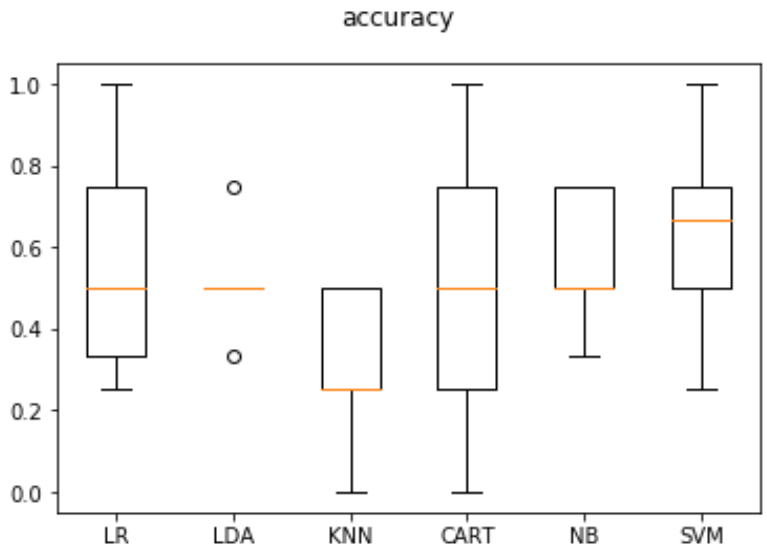


Figure 4-36 **Breast Cancer Dataset GDS3139**
Graphical representation of 5-fold Cross-Validation Accuracy Results using box and whisker plots

```
LR Accuracy: 0.975 (  0.049999999999999996 )
LDA Accuracy: 0.8055555555555556 (  0.12576923802968634 )
KNN Accuracy: 0.9027777777777779 (  0.09296222517045283 )
CART Accuracy: 0.9055555555555556 (  0.047628967220784024 )
NB Accuracy: 0.975 (  0.049999999999999996 )
SVM Accuracy: 0.975 (  0.049999999999999996 )
```

Figure 4-37 **AML Cancer Dataset GDS3057 5-fold Cross-Validation Accuracy Results**

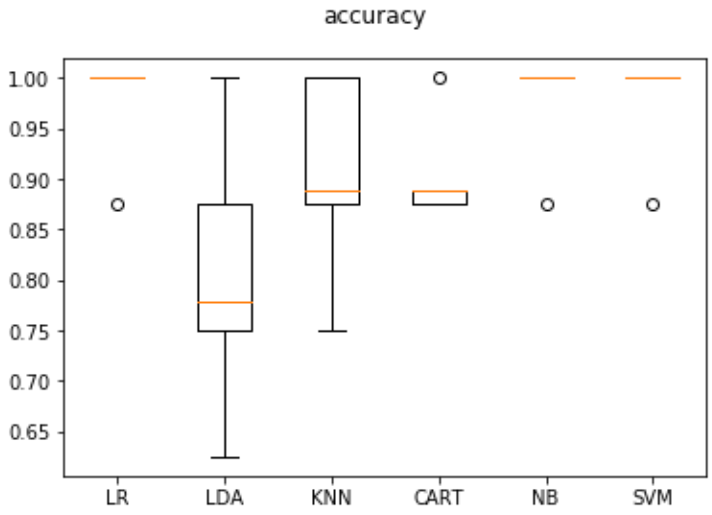


Figure 4-38 **AML Cancer Dataset GDS3057**
Graphical representation of 5-fold Cross-Validation Accuracy Results using box and whisker plots

```
LR Accuracy: 0.9095238095238095 (  0.07438333024673005 )
LDA Accuracy: 0.8476190476190476 (  0.13932132226978852 )
KNN Accuracy: 0.7857142857142858 (  0.1690308509457033 )
CART Accuracy: 0.8523809523809524 (  0.15678169174432693 )
NB Accuracy: 0.7904761904761904 (  0.08302664654363187 )
SVM Accuracy: 0.880952380952381 (  0.060233860193683424 )
```

Figure 4-39 **Pancreatic Cancer Dataset GDS4102 5-fold Cross-Validation Accuracy Results**

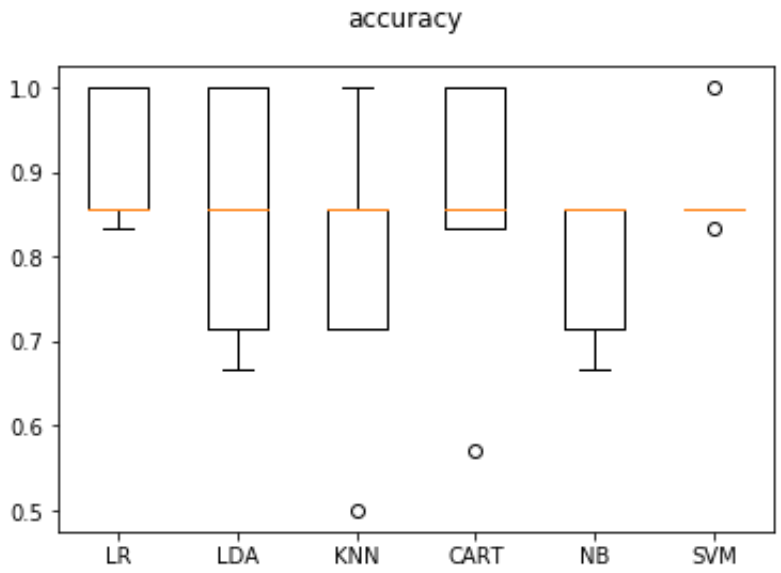


Figure 4-40 **Pancreatic Cancer Dataset GDS4102**
Graphical representation of 5-fold Cross-Validation Accuracy Results using box and whisker plots

These were the mean accuracy values. We check the distribution of accuracy values calculated across cross-validation folds. From a quick look, we observe that Logistic Regression and Linear SVM gives us the best accuracy scores with low variance.

The results on Breast Dataset are surprisingly bad, comparing to others. However we have to take into consideration that the Breast Cancer Dataset is the dataset with the fewest samples (29 samples, 15 healthy and 14 cancer), only with 19 sample left after split, for cross validation.

	Cervical	Breast	AML	Pancreatic
LR	93.8	56.6	97.5	90.9
LDA	85.2	51.6	80.5	84.7
KNN	91.4	30.0	90.2	78.5
CART	84.2	50.0	90.5	85.2
NB	90.9	56.6	97.5	79.0
SVM	90.9	63.3	97.5	88.0

Table 6 **Overview of 5-fold Cross-Validation Accuracy Results (%)**

4.7 Algorithm Evaluation: Standardized Data

Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1. It is most suitable for techniques that assume a Gaussian distribution in the input variables and work better with rescaled data, such as linear regression, logistic regression and linear discriminate analysis. (Brownlee)

In order to avoid a negative impact on algorithms' skill, due to raw data's differing distribution, now we revaluate them with a standardized copy of the training dataset. Also, a useful technique that scikit-learn library provide us in order to avoid data leakage is pipelines. In this stage, we give pipelines the scaler and the algorithm and after we test the model with 5-fold Cross-Validation.

4.7.1 Cross-Validation Results

```
ScaledLR Accuracy: 0.9380952380952381 (  0.0761904761904762 )
ScaledLDA Accuracy: 0.8523809523809524 (  0.09085135251589958 )
ScaledKNN Accuracy: 0.9142857142857143 (  0.11428571428571428 )
ScaledCART Accuracy: 0.8476190476190476 (  0.13932132226978852 )
ScaledNB Accuracy: 0.9095238095238095 (  0.07438333024673005 )
ScaledSVM Accuracy: 0.9380952380952381 (  0.0761904761904762 )
```

Figure 4-41 **Cervical Cancer Dataset GDS3233 5-fold Cross-Validation Accuracy Results on Standardized Data**

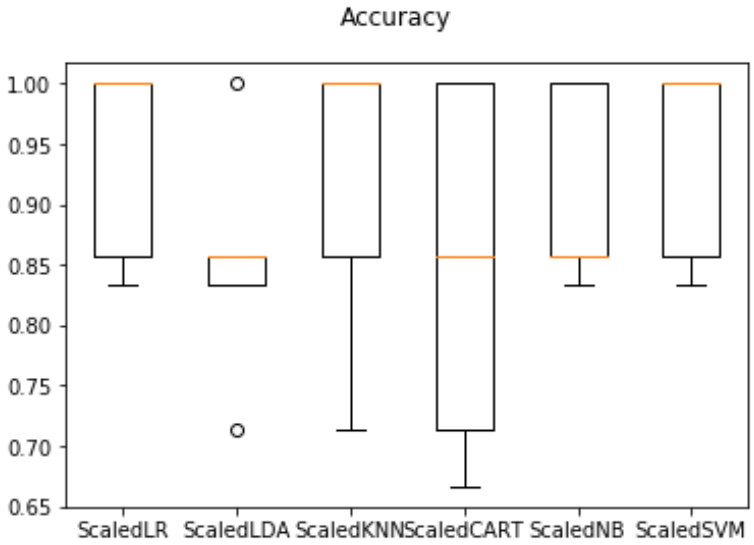


Figure 4-42 **Cervical Cancer Dataset GDS3233**
Graphical representation of 5-fold Cross-Validation Accuracy Results on Standardized Data using box and whisker plots

```
ScaledLR Accuracy: 0.4666666666666667 (  0.17159383568311665 )
ScaledLDA Accuracy: 0.5166666666666667 (  0.13333333333333336 )
ScaledKNN Accuracy: 0.4833333333333333 (  0.2603416558635552 )
ScaledCART Accuracy: 0.5166666666666667 (  0.20682789409984761 )
ScaledNB Accuracy: 0.5166666666666667 (  0.13333333333333336 )
ScaledSVM Accuracy: 0.5166666666666667 (  0.2603416558635551 )
```

Figure 4-43 **Breast Cancer Dataset GDS3139 5-fold Cross-Validation Accuracy Results on Standardized Data**

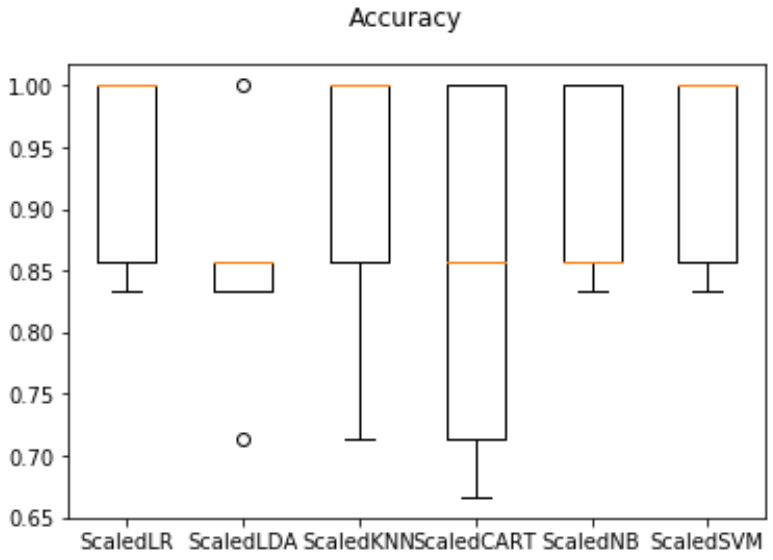


Figure 4-44 **Breast Cancer Dataset GDS3139**

Graphical representation of 5-fold Cross-Validation Accuracy Results on Standardized Data using box and whisker plots

```
ScaledLR Accuracy: 0.975 (  0.049999999999999996 )
ScaledLDA Accuracy: 0.8055555555555556 (  0.12576923802968634 )
ScaledKNN Accuracy: 0.7861111111111111 (  0.04614791034954486 )
ScaledCART Accuracy: 0.8805555555555555 (  0.07934920476158722 )
ScaledNB Accuracy: 0.975 (  0.049999999999999996 )
ScaledSVM Accuracy: 0.95 (  0.06123724356957946 )
```

Figure 4-45 **AML Cancer Dataset GDS3057 5-fold Cross-Validation Accuracy Results on Standardized Data**

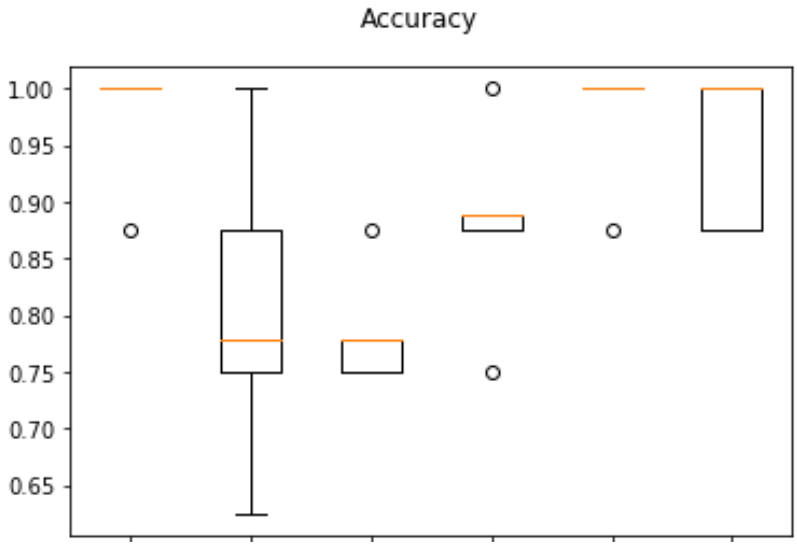


Figure 4-46 **AML Cancer Dataset GDS3057**

Graphical representation of 5-fold Cross-Validation Accuracy Results on Standardized Data using box and whisker plots

```
ScaledLR Accuracy: 0.8428571428571429 (  0.18294640678379567 )
ScaledLDA Accuracy: 0.8476190476190476 (  0.13932132226978852 )
ScaledKNN Accuracy: 0.7571428571428571 (  0.139970842444753 )
ScaledCART Accuracy: 0.9142857142857143 (  0.11428571428571428 )
ScaledNB Accuracy: 0.7047619047619047 (  0.01904761904761907 )
ScaledSVM Accuracy: 0.8761904761904763 (  0.12270570215928693 )
```

Figure 4-47 **Pancreatic Cancer Dataset GDS4102 5-fold Cross-Validation Accuracy Results on Standardized Data**

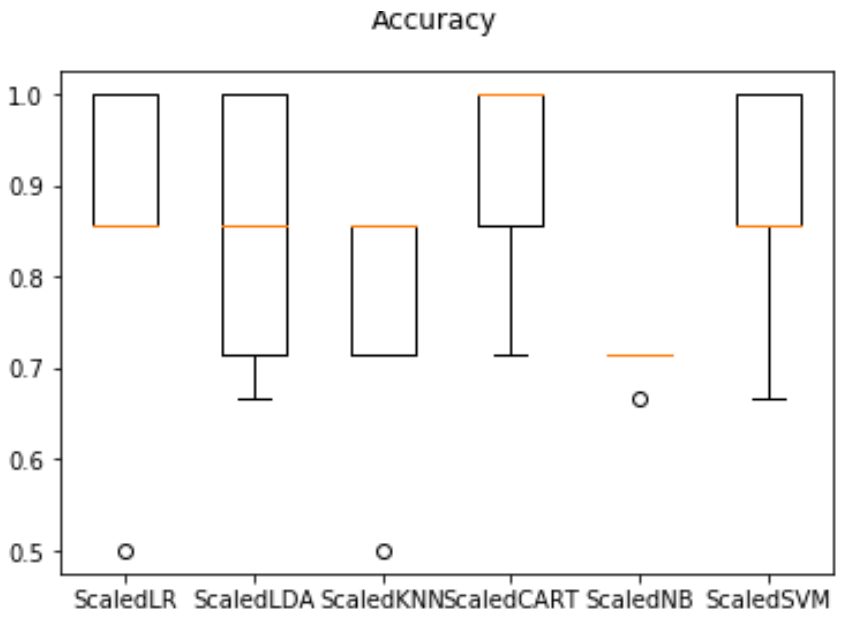


Figure 4-48 **AML Cancer Dataset GDS3057**
Graphical representation of 5-fold Cross-Validation Accuracy Results on Standardized Data using box and whisker plots

	Cervical		Breast		AML		Pancreatic	
	Primary	Scaled	Primary	Scaled	Primary	Scaled	Primary	Scaled
LR	93.8	93.8	56.6	46.6	97.5	97.5	90.9	84.2
LDA	85.2	85.2	51.6	51.6	80.5	80.5	84.7	84.7
KNN	91.4	91.4	30.0	48.3	90.2	78.6	78.5	75.7
CART	84.2	84.7	50.0	51.6	90.5	88.0	85.2	91.4
NB	90.9	90.9	56.6	51.6	97.5	97.5	79.0	70.4
SVM	90.9	93.8	63.3	51.6	97.5	95.0	88.0	87.6

Table 7 **Overview of 5-fold Cross-Validation Accuracy Results (%) before and after Data Standardization**

4.8 Algorithm Evaluation: Feature Reduction on Standardized Data

The data features that are used to train the machine learning models have a huge influence on the performance we can achieve. Irrelevant or partially relevant features can negatively impact model performance.

Feature selection is a process where you automatically select those features in the data that contribute most to the prediction variable or output in which you are interested. Having irrelevant features in our data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression. Three benefits of performing feature selection before modeling your data are:

- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modeling accuracy improves.
- **Reduces Training Time**: Less data means that algorithms train faster. (Brownlee)

4.8.1 PCA

Principal Component Analysis (or PCA) uses linear algebra to transform the dataset into a **compressed form**. Generally this is called a data reduction technique. A property of PCA is that you can choose the number of dimensions or principal components in the transformed result. (Brownlee)

In this stage of our study, we find the how many components to use by selecting **95% of total variance of the Train Set**. After we use PCA, as a part of pipelines, on standardized data and reevaluate our models.

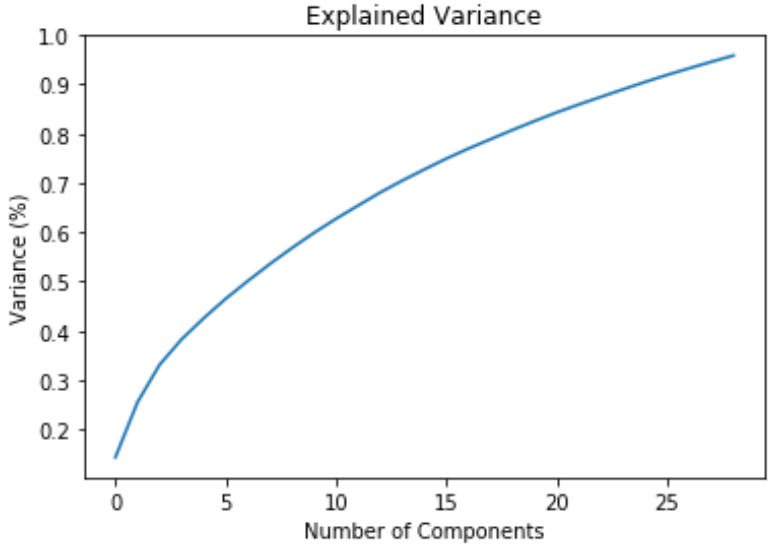


Figure 4-49 **Cervical Cancer Dataset GDS3233 Explained Variance**
X axis: Number of PCA Components, Y axis: Variance (%)

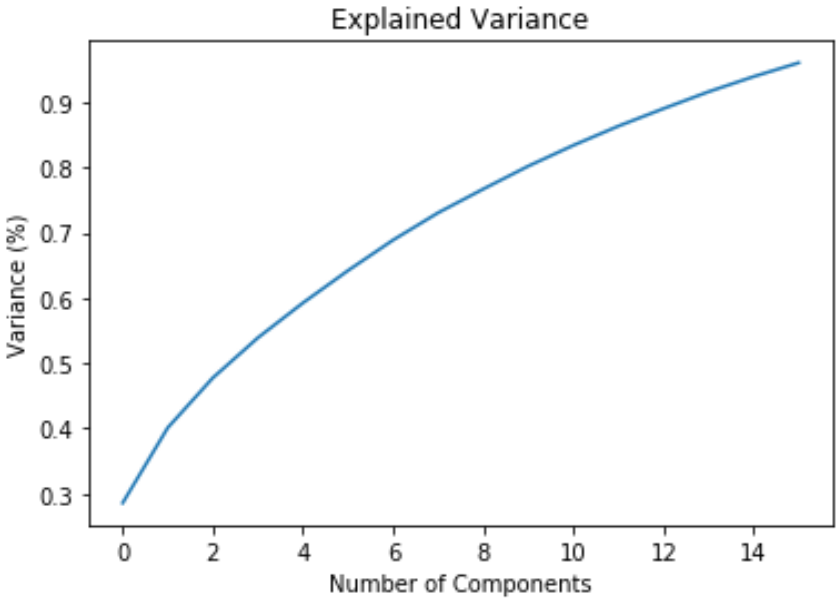


Figure 4-50 **Breast Cancer Dataset GDS3139 Explained Variance**
X axis: Number of PCA Components, Y axis: Variance (%)

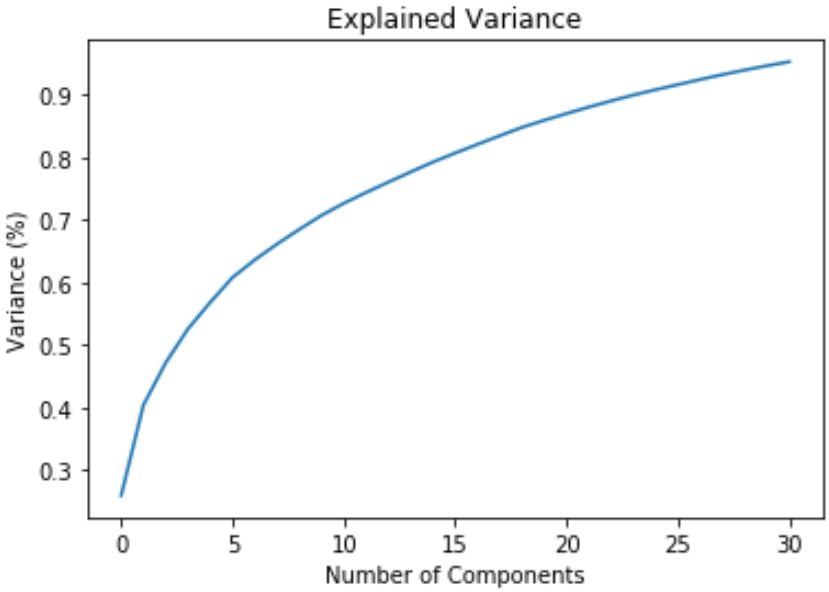


Figure 4-51 **AML Cancer Dataset GDS3057 Explained Variance**
X axis: Number of PCA Components, Y axis: Variance (%)

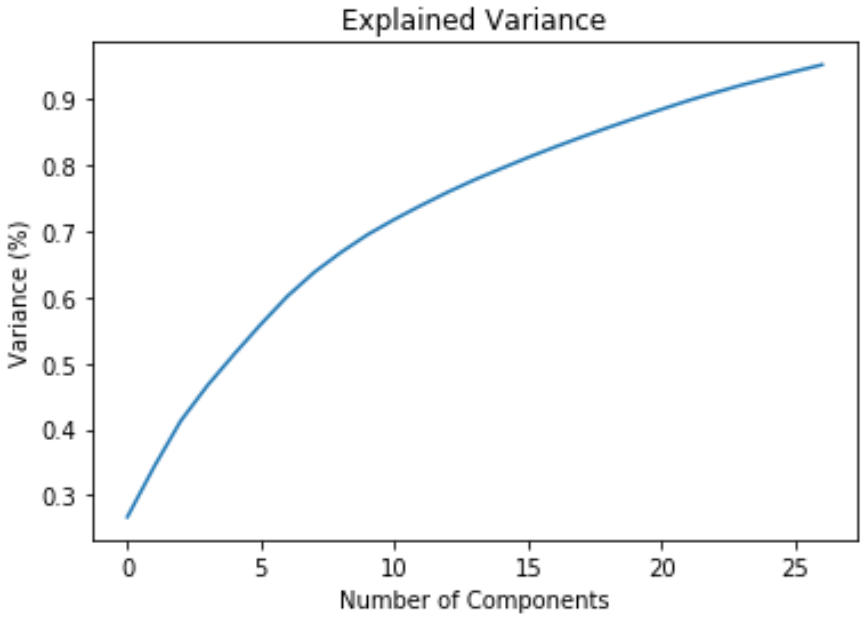


Figure 4-52 **Pancreatic Cancer Dataset GDS4102 Explained Variance**
X axis: Number of PCA Components, Y axis: Variance (%)

The threshold of variance for transforming the train set features into PCA components was set for the 95%. In each dataset, the train set was transformed into a dataset whose features now are compressed and their information represents the 95% of variance information of the previous dataset.

	original shape:	**transformed shape:**
Cervical	(34, 14062)	(34, 20)
Breast	(19, 22283)	(19, 12)
AML	(42, 22283)	(42, 27)
Pancreatic	(34, 54613)	(34, 16)

Table 8 **Train Datasets features transformation with PCA Components**

(a) Original shape of Train Dataset (Samples, Features)
(b) Transformed shape of Train Dataset (Samples, Components)

4.8.2 Cross-Validation Results

There are standard workflows in applied machine learning. Standard because they overcome common problems like **data leakage** in your test harness. Python scikit-learn provides a **Pipeline** utility to help automate machine learning workflows. Pipelines work by allowing for a linear sequence of data transforms to be chained together culminating in a modeling process that can be evaluated. The goal is to ensure that all of the steps in the pipeline are constrained to the data available for the evaluation, such as the training dataset or each fold of the cross-validation procedure.

In the final stage of the study, we build the pipelines with transformed features and the algorithms. The pipeline provides a handy tool called the **FeatureUnion** which allows the results of multiple feature selection and extraction procedures to be combined into a larger dataset on which a model can be trained. Importantly, all the feature extraction and the feature union occurs within each fold of the cross-validation procedure. (Brownlee)

Our study is finalized with pipelines in the steps below:

1. Data Standaridization
2. Feature Extraction with Principal Component Analysis.
3. Feature Extraction with Statistical Selection. Select features according to the k highest scores with SelectKBest() function. It scores the features using a function (in this case f_classif) and then removes all but the k highest scoring features. The score function refers to ANOVA F-value for the classification. (wiki/F-test)

$$F = \frac{explained\ variance}{unexplained\ variance} = \frac{\sum_{i=1}^{K} n_i (\bar{Y}_i - \bar{Y})^2 / (K-1)}{\sum_{i=1}^{k}\sum_{j=1}^{n_i} n_i (Y_{ij} - \bar{Y}_i)^2 / (N-K)}$$

4. Feature Union. Concatenates results of multiple transformer objects.
5. Learn the six algorithms.

The pipeline is then evaluated using 5-fold cross-validation.

```
LR: Accuracy: 0.9166666666666666 (  0.12909944487358058 )
LDA: Accuracy: 0.8833333333333332 (  0.1452966314513558 )
KNN: Accuracy: 0.925 (  0.11456439237389601 )
CART: Accuracy: 0.9166666666666666 (  0.12909944487358058 )
NB: Accuracy: 0.95 (  0.09999999999999999 )
SVM: Accuracy: 0.95 (  0.09999999999999999 )
```

Figure 4-53 **Cervical Cancer Dataset GDS3233 5-fold Cross-Validation Accuracy Results after PCA on Standardized Data**

```
LR  Accuracy: 0.5 (  0.31622776601683794 )
LDA  Accuracy: 0.55 (  0.35 )
KNN  Accuracy: 0.25 (  0.33541019662496846 )
CART Accuracy: 0.5 (  0.31622776601683794 )
NB Accuracy: 0.45 (  0.26925824035672524 )
SVM Accuracy: 0.5 (  0.3872983346207417 )
```

Figure 4-54 **Breast Cancer Dataset GDS3139 5-fold Cross-Validation Accuracy Results after PCA on Standardized Data**

```
CART Accuracy: 0.975 (  0.075 )
NB Accuracy: 0.95 (  0.09999999999999999 )
SVM Accuracy: 0.975 (  0.075 )
```

Figure 4-55 **AML Cancer Dataset GDS3057 5-fold Cross-Validation Accuracy Results after PCA on Standardized Data**

```
LR  Accuracy: 0.9083333333333332 (  0.1416666666666667 )
LDA  Accuracy: 0.8333333333333334 (  0.22360679774997896 )
KNN  Accuracy: 0.75 (  0.2886751345948129 )
CART Accuracy: 0.9083333333333332 (  0.1416666666666667 )
NB Accuracy: 0.65 (  0.31797973380564853 )
SVM Accuracy: 0.8749999999999998 (  0.1547847968417226 )
```

Figure 4-56 **Pancreatic Cancer Dataset GDS4102 5-fold Cross-Validation Accuracy Results after PCA on Standardized Data**

	Cervical			**Breast**			**AML**			**Pancreatic**		
	Primary	**Scaled**	**Feature Reduction**	**Primary**	**Scaled**	**Feature Reduction**	**Primary**	**Scaled**	**Feature Reduction**	**Primary**	**Scaled**	**Feature Reduction**
LR	93.8	93.8	91.6	56.6	46.6	50.0	97.5	97.5	97.5	90.9	84.2	90.8
LDA	85.2	85.2	88.3	51.6	51.6	55.0	80.5	80.5	88.0	84.7	84.7	83.3
KNN	91.4	91.4	92.5	30.0	48.3	25.0	90.2	78.6	78.4	78.5	75.7	75.0
CART	84.2	84.7	91.6	50.0	51.6	50.0	90.5	88.0	97.5	85.2	91.4	90.8
NB	90.9	90.9	95.0	56.6	51.6	45.0	97.5	97.5	95.0	79.0	70.4	65.0
SVM	90.9	93.8	95.0	63.3	51.6	50.0	97.5	95.0	97.5	88.0	87.6	87.4

Table 9 **Overview of 5-fold Cross-Validation Accuracy Results (%) on three different scenarios**

(c) Not scaled data (b) Standardized Data (c) with PCA feature extraction on standardized data

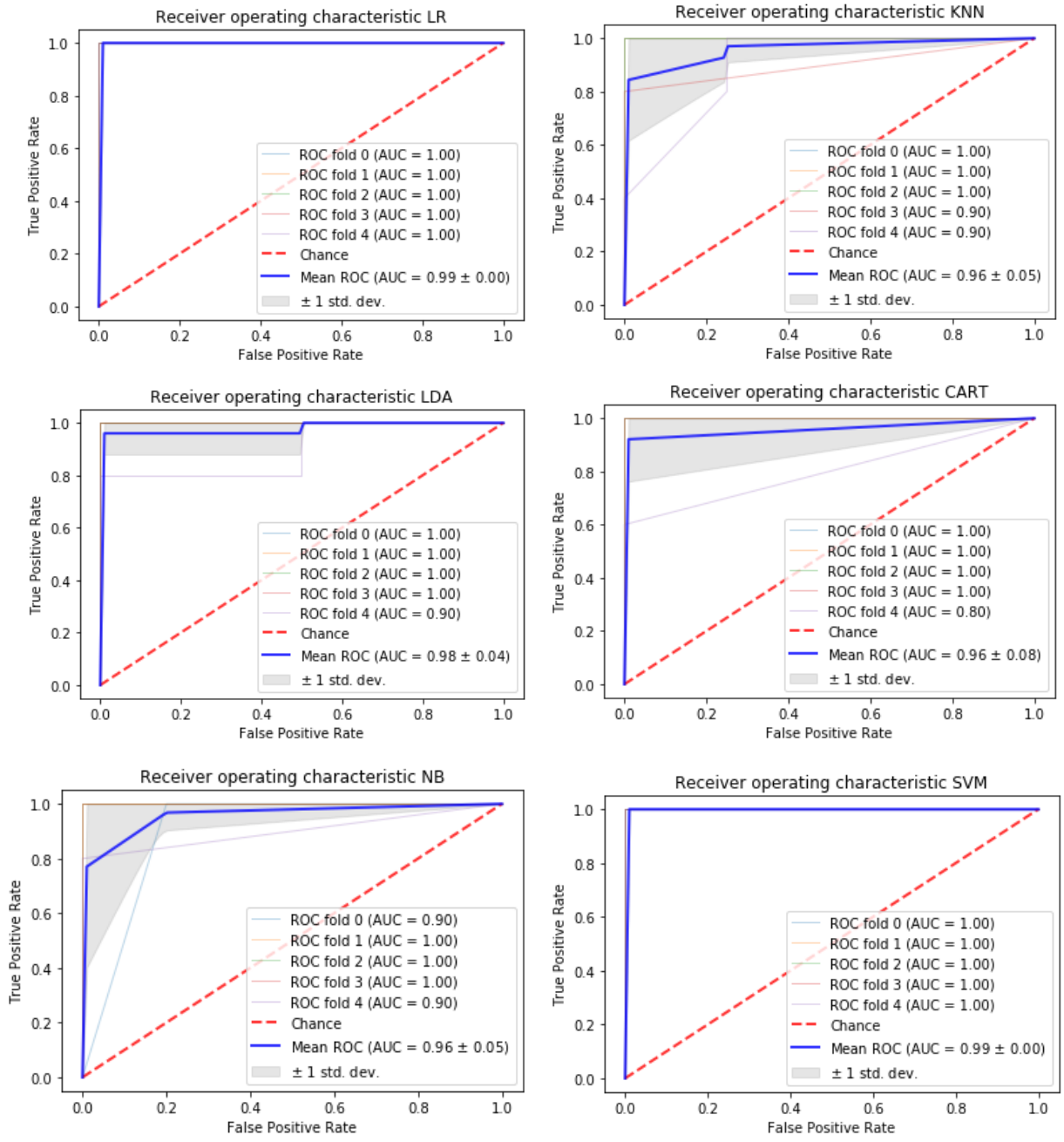


Figure 4-57 **ROC Curve of 5-fold cross-validation of 6 Classification models on Cervical Cancer Dataset GDS3233**

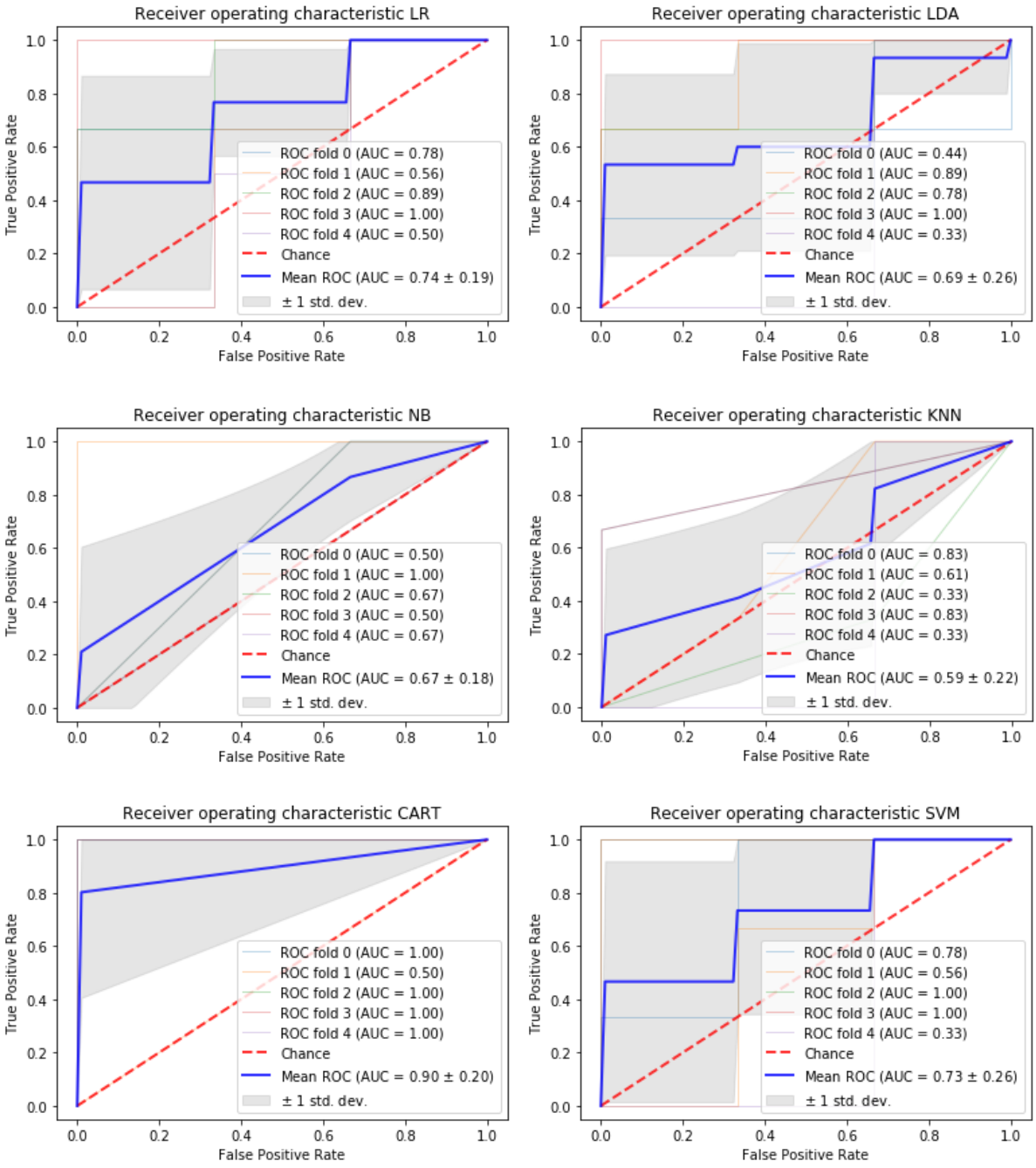


Figure 4-58 **ROC Curve of 5-fold cross-validation of 6 Classification models on Breast Cancer Dataset GDS3139**

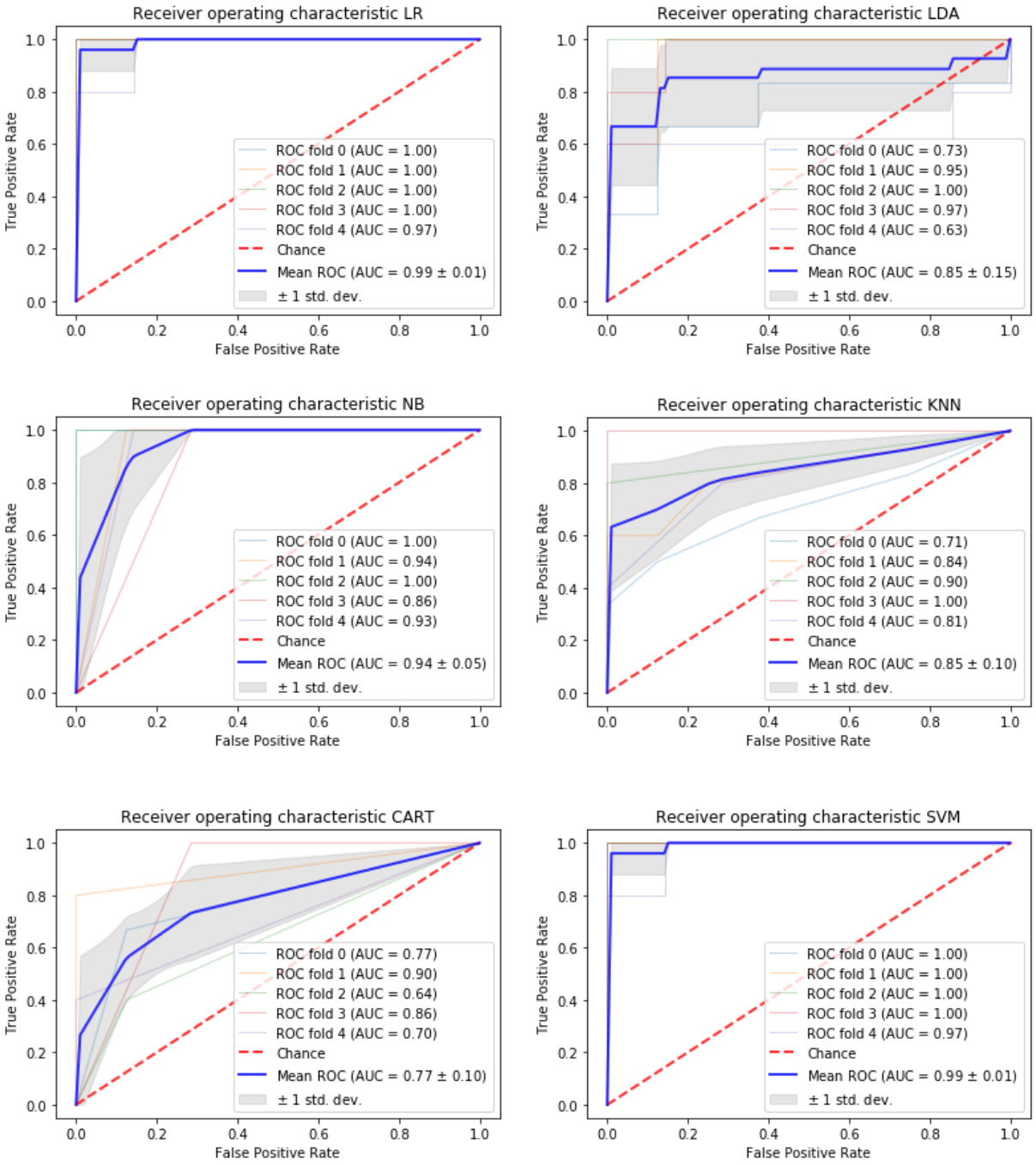


Figure 4-59 **ROC Curve of 5-fold cross-validation of 6 Classification models on AML Cancer Dataset GDS3057**

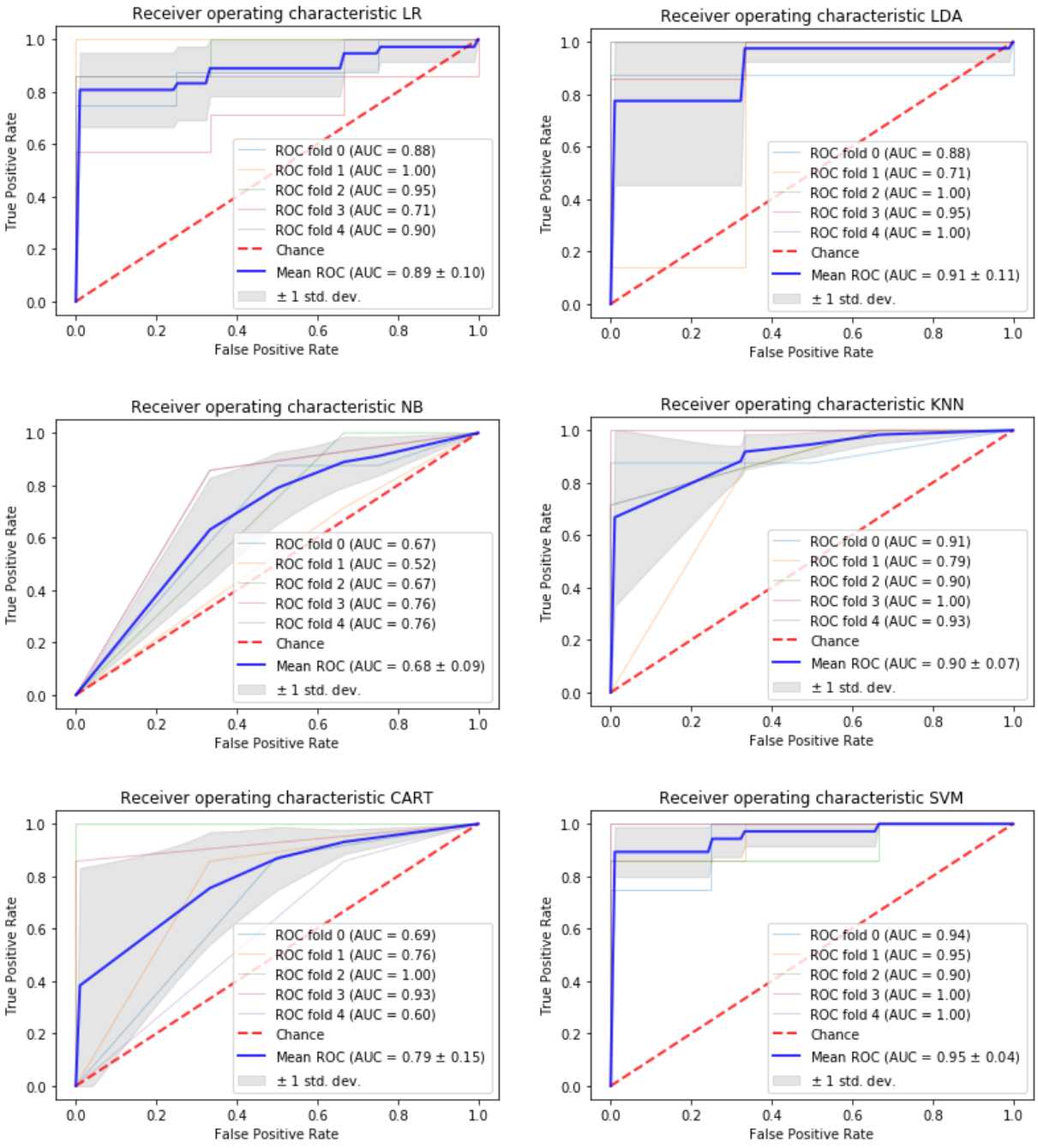


Figure 4-60 **ROC Curve of 5-fold cross-validation of 6 Classification models on Pancreatic Cancer Dataset GDS4102**

4.8.3 Predictions Results

Unlike statistics, where models are used to understand data, predictive modeling is laser focused on developing models that make the most accurate predictions at the expense of explaining why predictions are made. Why can't we prepare your machine learning algorithm on our training dataset and use predictions from this same dataset to evaluate performance? The simple answer is **overfitting**. Imagine an algorithm that remembers every observation it is shown during training. If we evaluated our machine learning algorithm on the same dataset used to train the algorithm, then an algorithm like this would have a perfect score on the training dataset. But the predictions it made on new data would be terrible. We must evaluate our machine learning algorithms on data that is not used to train the algorithm. (Brownlee)

The last stage of our study is to check the performance of our models on unseen data. We will finalize the models by training them on the entire training dataset and make predictions for the hold-out validation dataset to confirm our findings.

Confusion matrix here has biological meaning too. Considering how we have assigned the classes healthy in 0 (Positives) and cancer with 1 (Negatives), the True Negatives (TN) in confusion matrix play the most important role for medical diagnosis, as they predict the cases of cancer.

```
LR: 1.0
[[9 0]
 [0 9]]
LR
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00         9
         1.0       1.00      1.00      1.00         9

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18

LDA: 0.7222222222222222
[[4 5]
 [0 9]]
LDA
              precision    recall  f1-score   support

         0.0       1.00      0.44      0.62         9
         1.0       0.64      1.00      0.78         9

    accuracy                           0.72        18
   macro avg       0.82      0.72      0.70        18
weighted avg       0.82      0.72      0.70        18

KNN: 0.9444444444444444
[[8 1]
 [0 9]]
KNN
              precision    recall  f1-score   support

         0.0       1.00      0.89      0.94         9
         1.0       0.90      1.00      0.95         9

    accuracy                           0.94        18
   macro avg       0.95      0.94      0.94        18
weighted avg       0.95      0.94      0.94        18
```

```
CART: 0.7777777777777778
[[7 2]
 [2 7]]
CART
              precision    recall  f1-score   support

         0.0       0.78      0.78      0.78         9
         1.0       0.78      0.78      0.78         9

    accuracy                           0.78        18
   macro avg       0.78      0.78      0.78        18
weighted avg       0.78      0.78      0.78        18

NB: 1.0
[[9 0]
 [0 9]]
NB
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00         9
         1.0       1.00      1.00      1.00         9

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18

SVM: 1.0
[[9 0]
 [0 9]]
SVM
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00         9
         1.0       1.00      1.00      1.00         9

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18
```

Figure 4-61 **Predictions Results of 6 Classification models of Cervical Cancer Dataset GDS3233**
(a) Accuracy of predictions on Validation Data (b) Confusion matrix (c) Classification Report

```
LR: 0.6
[[2 2]
 [2 4]]
LR
              precision    recall  f1-score   support

         0.0       0.50      0.50      0.50         4
         1.0       0.67      0.67      0.67         6

    accuracy                           0.60        10
   macro avg       0.58      0.58      0.58        10
weighted avg       0.60      0.60      0.60        10

LDA: 0.9
[[4 0]
 [1 5]]
LDA
              precision    recall  f1-score   support

         0.0       0.80      1.00      0.89         4
         1.0       1.00      0.83      0.91         6

    accuracy                           0.90        10
   macro avg       0.90      0.92      0.90        10
weighted avg       0.92      0.90      0.90        10

KNN: 0.6
[[1 3]
 [1 5]]
KNN
              precision    recall  f1-score   support

         0.0       0.50      0.25      0.33         4
         1.0       0.62      0.83      0.71         6

    accuracy                           0.60        10
   macro avg       0.56      0.54      0.52        10
weighted avg       0.57      0.60      0.56        10
```

```
CART: 0.7
[[3 1]
 [2 4]]
CART
              precision    recall  f1-score   support

         0.0       0.60      0.75      0.67         4
         1.0       0.80      0.67      0.73         6

    accuracy                           0.70        10
   macro avg       0.70      0.71      0.70        10
weighted avg       0.72      0.70      0.70        10

NB: 0.6
[[0 4]
 [0 6]]
NB
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00         4
         1.0       0.60      1.00      0.75         6

    accuracy                           0.60        10
   macro avg       0.30      0.50      0.37        10
weighted avg       0.36      0.60      0.45        10

SVM: 0.8
[[3 1]
 [1 5]]
SVM
              precision    recall  f1-score   support

         0.0       0.75      0.75      0.75         4
         1.0       0.83      0.83      0.83         6

    accuracy                           0.80        10
   macro avg       0.79      0.79      0.79        10
weighted avg       0.80      0.80      0.80        10
```

Figure 4-62 **Predictions Results of 6 Classification models of Breast Cancer Dataset GDS3139**
(a) Accuracy of predictions on Validation Data (b) Confusion matrix (c) Classification Report

```
LR: 0.9090909090909091
[[14  0]
 [ 2  6]]
LR
              precision    recall  f1-score   support

         0.0       0.88      1.00      0.93        14
         1.0       1.00      0.75      0.86         8

    accuracy                           0.91        22
   macro avg       0.94      0.88      0.90        22
weighted avg       0.92      0.91      0.91        22

LDA: 0.8636363636363636
[[13  1]
 [ 2  6]]
LDA
              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90        14
         1.0       0.86      0.75      0.80         8

    accuracy                           0.86        22
   macro avg       0.86      0.84      0.85        22
weighted avg       0.86      0.86      0.86        22

KNN: 0.6818181818181818
[[14  0]
 [ 7  1]]
KNN
              precision    recall  f1-score   support

         0.0       0.67      1.00      0.80        14
         1.0       1.00      0.12      0.22         8

    accuracy                           0.68        22
   macro avg       0.83      0.56      0.51        22
weighted avg       0.79      0.68      0.59        22
```

```
CART: 0.8181818181818182
[[14  0]
 [ 4  4]]
CART
              precision    recall  f1-score   support

         0.0       0.78      1.00      0.88        14
         1.0       1.00      0.50      0.67         8

    accuracy                           0.82        22
   macro avg       0.89      0.75      0.77        22
weighted avg       0.86      0.82      0.80        22

NB: 0.9545454545454546
[[13  1]
 [ 0  8]]
NB
              precision    recall  f1-score   support

         0.0       1.00      0.93      0.96        14
         1.0       0.89      1.00      0.94         8

    accuracy                           0.95        22
   macro avg       0.94      0.96      0.95        22
weighted avg       0.96      0.95      0.96        22

SVM: 0.9090909090909091
[[14  0]
 [ 2  6]]
SVM
              precision    recall  f1-score   support

         0.0       0.88      1.00      0.93        14
         1.0       1.00      0.75      0.86         8

    accuracy                           0.91        22
   macro avg       0.94      0.88      0.90        22
weighted avg       0.92      0.91      0.91        22
```

Figure 4-63 **Predictions Results of 6 Classification models of AML Cancer Dataset GDS3057**
Accuracy of predictions on Validation Data (b) Confusion matrix (c) Classification Report

```
LR: 0.9444444444444444
[[ 3  0]
 [ 1 14]]
LR
              precision    recall  f1-score   support

         0.0       0.75      1.00      0.86         3
         1.0       1.00      0.93      0.97        15

    accuracy                           0.94        18
   macro avg       0.88      0.97      0.91        18
weighted avg       0.96      0.94      0.95        18

LDA: 0.8888888888888888
[[ 2  1]
 [ 1 14]]
LDA
              precision    recall  f1-score   support

         0.0       0.67      0.67      0.67         3
         1.0       0.93      0.93      0.93        15

    accuracy                           0.89        18
   macro avg       0.80      0.80      0.80        18
weighted avg       0.89      0.89      0.89        18

KNN: 0.8333333333333334
[[ 2  1]
 [ 2 13]]
KNN
              precision    recall  f1-score   support

         0.0       0.50      0.67      0.57         3
         1.0       0.93      0.87      0.90        15

    accuracy                           0.83        18
   macro avg       0.71      0.77      0.73        18
weighted avg       0.86      0.83      0.84        18
```

```
CART: 0.7777777777777778
[[ 2  1]
 [ 3 12]]
CART
              precision    recall  f1-score   support

         0.0       0.40      0.67      0.50         3
         1.0       0.92      0.80      0.86        15

    accuracy                           0.78        18
   macro avg       0.66      0.73      0.68        18
weighted avg       0.84      0.78      0.80        18

NB: 0.8888888888888888
[[ 2  1]
 [ 1 14]]
NB
              precision    recall  f1-score   support

         0.0       0.67      0.67      0.67         3
         1.0       0.93      0.93      0.93        15

    accuracy                           0.89        18
   macro avg       0.80      0.80      0.80        18
weighted avg       0.89      0.89      0.89        18

SVM: 0.8888888888888888
[[ 2  1]
 [ 1 14]]
SVM
              precision    recall  f1-score   support

         0.0       0.67      0.67      0.67         3
         1.0       0.93      0.93      0.93        15

    accuracy                           0.89        18
   macro avg       0.80      0.80      0.80        18
weighted avg       0.89      0.89      0.89        18
```

Figure 4-64 **Predictions Results of 6 Classification models of Pancreatic Cancer Dataset GDS4102**
(a) Accuracy of predictions on Validation Data (b) Confusion matrix (c) Classification Report

	Cervical		Breast		AML		Pancreatic	
	Cross-Validation	Predictions	Cross-Validation	Predictions	Cross-Validation	Predictions	Cross-Validation	Predictions
LR	91.6	100.0 9 0 0 9	50.0	60.0 2 2 2 4	97.5	90.0 14 0 2 6	90.8	94.4 3 0 1 14
LDA	88.3	72.2 4 5 0 9	55.0	90.0 4 0 1 5	88.0	86.3 13 1 2 6	83.3	88.8 2 1 1 14
KNN	92.5	94.4 8 1 0 9	25.0	60.0 1 3 1 5	78.4	68.1 14 0 7 1	75.0	83.3 2 1 2 13
CART	91.6	77.7 7 2 2 7	50.0	70.0 3 1 2 4	97.5	81.8 14 0 4 4	90.8	77.7 2 1 3 12
NB	95.0	100.0 9 0 0 9	45.0	60.0 0 4 0 6	95.0	95.4 13 1 0 8	65.0	88.8 2 1 1 14
SVM	95.0	100.0 9 0 0 9	50.0	80.0 3 1 1 5	97.5	90.9 14 0 2 6	87.4	88.8 2 1 1 14

Table 9 **Summarization of basic study results of Accuracy (%) and Confusion Matrix.**

(a) In cross-validation columns we see the accuracy score of 5-fold cross-validation on train data at the 3rd scenario.

(b) In Predictions columns we see the accuracy score and confusion matrix of the 6 models on unseen data (validation dataset).

4.9 Final Model

In our study, finally, we picked the most accurate model considering its performance in predictions and metrics like confusion matrix and classification report. After picking the right model for each dataset we can save and load the model using the Joblib library. The Joblib library is part of the SciPy ecosystem and provides utilities for pipelining Python jobs. It provides utilities for saving and loading Python objects that make use of NumPy data structures, efficiently.

Running the Jupyter Notebook saves the model to file as finalized model.sav and also creates one file for each NumPy array in the model. After the model is loaded an estimate of the accuracy of the model on unseen data is reported (Brownlee).

- In case of Cervival Cancer Classification dataset GDS3233 we can pick models between Logistic Regression, Naïve Bayes and Linear SVM algorithms, with final accuracy on unseen data at 100%.
- In case of Breast Cancer Classification dataset GDS3139 we picked the model trained with Linear Discriminant Analysis algorithm with final accuracy on unseen data at 90%.
- In case of AML Cancer Classification dataset GDS3057 we picked the model trained with Naïve Bayes algorithm with final accuracy on unseen data at 95.4%.
- In case of Pancreatic Cancer Classification dataset GDS4102 we picked the model trained with Logistic Regression algorithm with final accuracy on unseen data at 94.4%.

5. CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

The aim of our thesis is to make a robust and efficient binary classification predictive model, which applies in gene expression datasets with high accuracy, for medical diagnosis of cancer. Through the whole procedure, useful results came along with interesting thoughts and outcomes.

Firstly, in order to have a spherical view on the problem, we chose to conclude 4 different cancer datasets of 4 different cancers (Cervical, Breast, AML and Pancreatic) which were retrieved from different tissues or cell lines. A main characteristic of these datasets is that they are high dimensional (thousands of gene expression levels), but the number of samples is significantly lower (tens of people). These 4 different datasets have different shapes (samples x features) as well: The Cervical dataset: 52 x 14063, the AML dataset: 64 x 22284, the Breast dataset: 29 x 22284, the Pancreatic dataset: 52 x 54614

So, we examine big and smaller datasets comparing the one with another. For example the AML dataset has twice samples than the Breast dataset with the same amount of features. Also, in case of Cervical cancer we have the same number of sample with the Pancreatic or AML cancer dataset, but under the half of features. An interesting insight will come up if we see how different algorithms will behavior in different shaped datasets.

Another criterion is how the data distribute into classes. In this thesis we examine imbalanced and balanced cases.

a) Cervical and Breast cancer dataset have balanced cases of cancer and healthy samples in different proportions each. For Cervical 24 healthy and 28 cancer and for Breast 15 healthy and 14 cancer samples.
b) AML dataset has more cases of healthy samples than cancer samples. (38 healthy, 26 cancer)
c) Pancreatic has almost double cancer samples than healthy. (16 healthy, 36 cancer)

Continuing, a very useful result that came up from feature statistics and visualization was that the most data are skewed right or left, like assuming an exponential distribution, some others, but only few of them in total, are symmetric assuming a Gaussian or nearly Gaussian distribution. Also, it is shown that datasets like AML, Breast and Pancreatic Cancer have high correlated features and might cause issues in models trained with linear algorithms. For algorithms like Linear Regression and Linear Discriminant Analysis which assume a Gaussian distribution in the input variables we have first to rescale the data.

The first results from cross-validation accuracy scores **Table 6** gives us a first impression on how the 6 algorithms behave on different datasets and how they classify the data. The smallest dataset (Breast) with the fewest samples gives the worst scores in the evaluation. On the other hand the algorithms give the best performance on the dataset with the biggest number of samples (AML). Also, if we examine the results from the algorithms view, we can observe that the Logistic Regression has the highest scores in all the datasets. Also 10/24 of scores are higher than 90% in accuracy and 16/24 are higher than 80%.

Proceeding in the next stage of the study, we have the results from cross-validation accuracy scores on the standardized data. In **Table 7** we compare the old results with the new and we can find interesting insights. To begin with from datasets view, the Cervical cancer dataset has the same results, except of CART and SVM algorithms whose accuracy is now higher. In Breast cancer dataset the results of cross-validation on standardized data showed same performance for LDA, lower for LR, NB and SVM, but higher for KNN and CART. In AML cancer dataset LR, LDA and NB had the same results, but KNN, CART and SVM had lower performance than before. Finally in Pancreatic cancer Dataset LDA performance is the same, CART's is higher and LR, KNN, NB, SVM got lower. On the other hand, from the algorithms view, LDA's performance stayed the same by the data transformation in all the cancer datasets and we noted high scores like 97.5% with LR and NB on AML cancer dataset.

In the third step, where we apply PCA and feature extraction techniques on standardized data, useful insights came up in **Table 8**. In case of Cervical cancer all the results scaled up, except of LR. In case of Breast cancer all the results scaled down, except of LDA. In AML case all results scaled up, except of NB which was 2.5% lower and finally, in Pancreatic cancer the results scaled down except of LR.

The last step of our study was to evaluate our models' predictions on unseen data and compare them to the last cross-validation' results, as it seems in **Table 9**. At first sight, the accuracy score of all the models in Breast cancer is significantly higher on unseen data than in cross-validation results. After, in Cervical cancer three models with LR, NB and SVM scores 100% accuracy on unseen data and the models with LDA(72.2%) and KNN (94.4%) scores 9/9 cancer samples correctly as it seems on confusion matrix. In AML cancer the models scored lower accuracy on unseen data, except of NB which also classified correctly all the cancer cases. At last, in Pancreatic cancer case models scored higher except CART. 9/24 cases scored higher than 90% accuracy on predictions and

Concluding, the best scores achieved in the Cervical cancer dataset which has 52 samples with 14,063 features, the biggest amount of samples with the fewest features, comparing to the other datasets. The poorest scores were given by the Breast cancer dataset, which has 29 samples with 22,284, the smallest dataset of them all. From the other hand Logistic Regression performed with high evaluation scores through all the datasets and Naïve Bayes classified all the cancer samples correctly in 3/4 datasets.

Finally, we end this thesis by proposing the models which were built, along with the each step's results for more observation and study in the case of supervised cancer classification and prediction. Also we propose, respectfully any procedure that occurred and can contribute in the classification and prediction of Cervical, Breast, Acute Myeloid Leukemia and Pancreatic cancer research.

5.2 FUTURE WORK

Microarrays of gene expression levels challenge nowadays computer science to develop procedures in order to process and extract knowledge from them. In the field of cancer classification different approaches of machine learning can be applied in order to identify patterns, develop efficient and accurate model for classification and finally medical diagnosis. On the other hand, in the big data era of computational biology, machine learning can progress as well.

In this thesis, we studied different, proposed ways from literature, with the intention of building models for cancer classification. The three steps sum up to building models among the best performance of 6 different algorithms' on primary data, on transformed data and finally on dimensionally reduced data.

The framework that was followed in this thesis can be applied with different techniques in each step. For future studies, more classification algorithms can be examined, with different data transformation and feature selection techniques in different step or combination.

To continue this thesis, a proposal is to improve performance with Ensembles methods that can boost the accuracy scores. Bagging, Boosting and Majority Voting are the most proposed methods in order to combine different models' predictions. Another step that can be added in this thesis is Algorithm Tuning. It can be considered as the last step of model finalizing in the applied machine learning procedure. On a given specific problem, machine learning models' behavior can be tuned with the purpose of finding the best combination of parameters.

We can extend this study, by using different classifier such as the Artificial Neural Networks which are related with cancer classification. The findings in that study can be compared with the finding in this thesis. Another approach could be to test different datasets in same way of this examination. Different cases of cancer classification or different shaped datasets could be lead to a better view on the problem and could extract different and important insights.

Finally as it is massively referred in the literature, machine learning is a field that requires practice of different experimental methodologies.

Bibliography

(n.d.). Retrieved from Gene Expression Omnibus: NCBI gene expression and hybridization array data repository: https://www.ncbi.nlm.nih.gov/geo/

(2009.). Artificial Intelligence: A Modern Approach (3rd Edition).

(2013). An Introduction to Statistical Learning.

(2013). An Introduction to Statistical Learning.

(2013). Applied Predictive Modeling.

A. TAşçi, T. İ. (2017). "A comparison of feature selection algorithms for cancer classification through gene expression data: Leukemia case". *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, (pp. 1352-1354). Bursa.

al., B. e. (2013). *API design for machine learning software: experiences from the scikit-learn project.*

al., P. e. (2011). Scikit-learn: Machine Learning in Python. JMLR 12.

Alberts, B. H. (n.d.). *ESSENTIAL CELL BIOLOGY.*

Alberts, W. J. (n.d.). *Molecular Biology of the Cell.*

Ali, A. (n.d.). *Medium.com*. Retrieved 2019, from https://medium.com/machine-learning-researcher/decision-tree-algorithm-in-machine-learning-248fb7de819e

anaconda.com/why-anaconda/. (n.d.). Retrieved from https://www.anaconda.com/why-anaconda/

Anil Jain, D. Z. (1997). Feature Selection: Evaluation, Application, and Small. *IEEE transactions on pattern analysis and machine intelligence* , 153-158.

Arthur, S. (1959). *Some Studies in Machine Learning Using the Game of Checkers.* IBM Journal of Research and Development.

Barrett, E. C. (2016). *Methods Mol Biol.*

Brownlee, J. (n.d.). *Machine Learning Mastery with Python, Machine Learning Mastery*. Retrieved 2019, from https://machinelearningmastery.com/machine-learning-with-python/

Buitinck et al., 2. (n.d.). *API design for machine learning software: experiences from the scikit-learn project.*

C., S. (2010). *Encyclopedia of Machine Learning and Data Mining / edited by Claude Sammut, Geoffrey I. Webb.* Boston, MA: Springer Science+Business Media, LLC.

C.ArunKumar, S. M. (2017). *A Comparative Performance Evaluation of Supervised Feature Selection Algorithms on Microarray Data.*

Diego Galar, U. K. (2017). *eMaintenance.*

expertsystem.com. (n.d.). Retrieved 2019, from https://expertsystem.com/machine-learning-definition/

F., C. (1970). Central dogma of molecular biology. *Nature*.

Fitzmaurice C, D. D.-L. (2015). The global burden of cancer 2013.

Fletcher, T. (2008). *Support Vector Machines Explained.* www.cs.ucl.ac.uk/staff/T.Fletcher.

Gareth James, D. W. (2013). An Introduction to Statistical Learning. New York: Springer.

Google Colaboratory . (n.d.). Retrieved from https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=GJBs_flRovLc

Gregory Piatetsky-Shapiro, P. T. (2003). Microarray Data Mining: Facing the Challenges . *ACM SIGKDD Explorations Newsletter*.

Hajian-Tilaki, K. (2013). *Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation.*

Hastie. (n.d.). Elements.

Ian H. Witten, E. F. (2011). Data mining: Practical Machine Learning Tools and Techniques. In W. I. H.), *Data mining: Practical Machine Learning Tools and Techniques.*

INDRAYAN, R. K. (2011). *Receiver Operating Characteristic (ROC) Curve for Medical Researchers.*

J.Brownlee. (n.d.). Retrieved from https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/

Jason, B. (2018). *A Gentle Introduction to k-fold Cross-Validation*. Retrieved 2019, from https://machinelearningmastery.com/: https://machinelearningmastery.com/k-fold-cross-validation/

Joseph A. Cruz, D. S. (n.d.). *Applications of Machine Learning in Cancer Prediction and Prognosis.* Canada T6G 2E8: Departments of Biological Science and Computing Science, University of Alberta Edmonton, AB,.

Jupyter.org. (n.d.). Retrieved from https://jupyter.org/about

Knudsen, S. (2006). *Cancer Diagnostics with DNA Microarrays.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Kuhlman, D. (2013). *A Python Book: Beginning Python, Advanced.*

Kumar, R. (2018). *Understanding Principal Component Analysis*. Retrieved 2019, from https://medium.com/: https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0

LEE, M.-L. T. (n.d.). *ANALYSIS OF MICROARRAY GENE EXPRESSION DATA.* NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW: KLUWER ACADEMIC PUBLISHERS.

Leif E. Peterson, M. A. (2008). Machine learning-based receiver operating characteristic curves for crisp and fuzzy classification DNA microarrays in cancer research. *International Journal of Approximate Reasoning*.

Lodish, B. M. (n.d.). *Molecular Cell Biology.*

Md. Mohaimenul Islam, T. N. (n.d.). *Machine Learning Models of Breast Cancer Risk Prediction.* doi: https://doi.org/10.1101/723304.

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Orange3. (n.d.). Retrieved from https://orange.biolab.si/

Pang Ning Tan, M. S. (2006). *Introduction to Data Mining.* Addison – Wesley Companion Book Site.

Pei H, L. L. (n.d.). FKBP51 affects cancer cell response to chemotherapy by negatively regulating Akt. *Cancer Cell 2009 Sep 8;16(3):259-66.* , PMID: 19732725.

Peter H. Raven, e. a. (n.d.). *Biology -- 9th ed.*

python.org/about/. (n.d.). Retrieved from https://www.python.org/about/

Robert J. Lipshutz, S. P. (1999, january). High density synthetic oligonucleotide arrays. *Nature America Inc.*

Saeys Yvan, e. a. (2007). A review of feature selection techniques in bioinformatics. *Oxford Univ Press*, 2507-2517.

Sammut C., W. G. (2010). *Encyclopedia of Machine Learning and Data Mining / edited by Claude Sammut, Geoffrey I. Webb.* Boston, MA: Springer Science+Business Media, LLC.

Science Direct. (n.d.). Retrieved 2019, from https://www.sciencedirect.com/topics/computer-science/binary-classification

scikit-learn.org. (n.d.). Retrieved from https://scikit-learn.org/stable/

scipy. (n.d.). Retrieved from https://scipy.github.io/devdocs/tutorial/general.html

Stekel, D. (2003). *Microarray Bioinformatics.* Cambridge, UK: Cambridge University Press.

Stewart, B. &. (2014.). *World cancer report.* World.

Tharwat, A. (2018). Classification assessment methods https://doi.org/10.1016/j.aci.2018.08.003. *Applied Computing and Informatics*.

The microarray: Potential aplications for ophthalmic research. (n.d.). Retrieved 11 3, 2019, from ResearchGate: https://www.researchgate.net/figure/Schematic-overview-of-spotted-cDNA-microarrays-and-high-density-oligonucleotide-arrays_fig1_11249590

Venugopal Mikkilineni, R. D. (2004). Digital Quantitative Measurements of Gene Expression. *Biotechnology and bioengineering*, pp. 117-124.

wiki/Arithmetic_mean. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Arithmetic_mean

wiki/F-test. (n.d.). Retrieved from https://en.wikipedia.org/wiki/F-test

wiki/Percentile. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Percentile

wiki/Scikit-learn. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Scikit-learn

wiki/Standard_deviation. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Standard_deviation

wikipedia.org. (n.d.). Retrieved 2019, from https://en.wikipedia.org/wiki/Accuracy_and_precision

Wong, G. (. (n.d.). Introduction. In Minna Laine. DNA Microarray data analysis. *Helsinki: CSC- Scientific computing Inc.*, (15-24). .

Zhao Youcai, H. S. (2017). *in Pollution Control and Resource Recovery.*