ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
Εργαστήριο Συστημάτων Υποστήριξης Αποφάσεων

Ανάπτυξη Μεθοδολογίας και Συστήματος Προσωποποιημένων Συστάσεων Ηλεκτρονικών Παιχνιδιών

ΠΑΠΑΔΗΜΗΤΡΙΟΥ ΘΟΔΩΡΗΣ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Νικόλαος Ματσατσίνης, Καθηγητής (επιβλέπων)

Μ. Δούμπος, Καθηγητής

Σ. Τσαφαράκης, Επίκ. Καθηγητής

Χανιά, Δεκέμβριος 2018

Πίνακας Περιεχομένων

Πίνακας Πινάκων

Πίνακας Διαγραμμάτων

Πίνακας Σχημάτων

Περίληψη

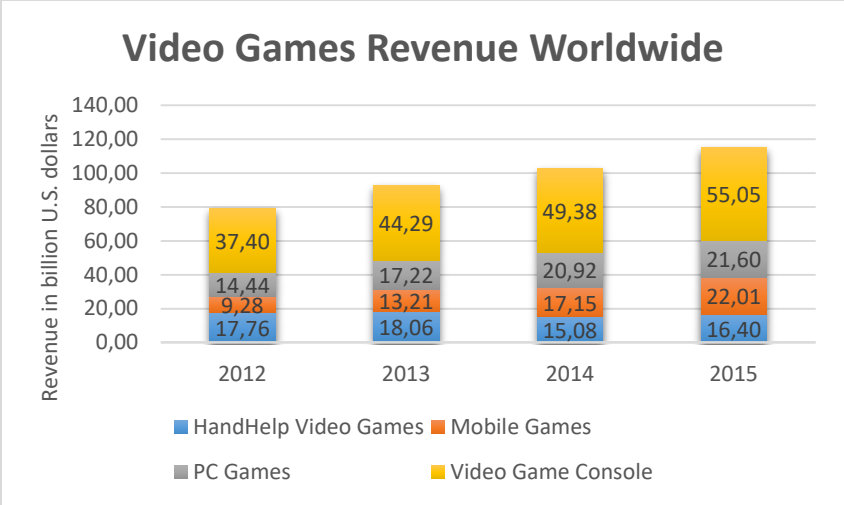
Η εργασία αυτή αφορά την ανάπτυξη μιας πολυκριτήριας μεθοδολογίας και ενός συστήματος συστάσεων, το οποίο βασιζόμενο στο προφίλ και τις απαιτήσεις των χρηστών (παικτών ηλεκτρονικών παιχνιδιών) και λαμβάνοντας υπόψη του τα χαρακτηριστικά των παιχνιδιών, θα αξιολογεί και θα προτείνει εκείνα τα ηλεκτρονικά παιχνίδια(video games) που θα του ταιριάζουν. Για την υλοποίηση του συστήματος έγινε χρήση του Steam Web Api για την εξαγωγή στοιχείων για τους χρήστες που είχαν το προφίλ τους δημόσιο. Παράλληλα αντλήθηκαν πληροφορίες για τα ηλεκτρονικά παιχνίδια και από άλλες ιστοσελίδες. Τα δεδομένα αυτά χρησιμοποιήθηκαν για τη δημιουργία των προφίλ των χρηστών και των παιχνιδιών και εν συνεχεία στη δημιουργία των σχετικών πολυκριτήριων πινάκων. Έγινε ανάλυση του δείγματος και εξόρυξη δεδομένων με χρήση τεχνικών συσταδοποίησης και ταξινόμησης. Με χρήση της Utastar και του αλγορίθμου k-means δημιουργείται μια λίστα προτεινόμενων παιχνιδιών για κάθε χρήστη της πλατφόρμας.

Κεφάλαιο 1: Εισαγωγή

1.1 Παρουσίαση προβλήματος

Η βιομηχανία των ηλεκτρονικών παιχνιδιών τα τελευταία χρόνια έχει μεγάλη άνθηση καθώς και μεγάλους τζίρους (Διάγραμμα 1). Τα συναντάμε από τα κλασικά video games στις κονσόλες, στον υπολογιστή αλλά πλέον και στις ταμπλέτες και τα κινητά. Είναι σύνηθες οι χρήστες να μην μπορούν να διαλέξουν ένα παιχνίδι της αρεσκείας τους λόγω του μεγάλου όγκου τίτλων, οπότε καταφεύγουν είτε στα πιο δημοφιλή είτε σε κάποιο που τους σύστησε κάποιος φίλος τους. Σε αυτή την περίπτωση είναι που ένα ολοκληρωμένο σύστημα συστάσεων θα τους προτείνει τις καλύτερες λύσεις ανάλογα με τις προτιμήσεις τους.

Στόχος της εργασίας είναι η ανάπτυξη μεθοδολογίας και συστήματος προσωποποιημένων συστάσεων ηλεκτρονικών παιχνιδιών. Για την υλοποίηση χρησιμοποιήσαμε πληροφορίες και δεδομένα από την πλατφόρμα του Steam καθώς 1) είναι ευρέως γνωστή στην κοινότητα 2) έχει μεγάλο αριθμό χρηστών και παιχνιδιών 3) έχουμε πρόσβαση σε στατιστικά και δεδομένα. Βέβαια, το ίδιο το steam έχει το δικό του σύστημα συστάσεων το οποίο όμως βάσει κριτικών από παίχτες αποδεικνύεται πως δεν πληροί τις απαιτήσεις τους καθώς προτείνει παιχνίδια τα οποία οι χρήστες έχουν ήδη παίξει ή που τελικά δεν τους ενδιαφέρουν. Όπως επίσης υπάρχουν και άλλα συστήματα από τρίτους τα οποία όμως χρησιμοποιούν συνήθως μόνο ένα κριτήριο για την λήψη απόφασης. Με έναν μέσο όρο 6 εκατομμυρίων συνδεδεμένων χρηστών στο Steam καθημερινά δημιουργείται η ανάγκη για την δημιουργία ενός συστήματος το οποίο χρησιμοποιεί πολλαπλά κριτήρια στην είσοδο για να έχει πιο αντιπροσωπευτικό αποτέλεσμα στην έξοδο.



Διάγραμμα 1 Έσοδα βιομηχανίας ηλεκτρονικών παιχνιδιών Πηγή: https://www.statista.com/

1.2 Δομή εργασίας

Χρησιμοποιώντας εργαλεία όπως το Steam Web Api αλλά και με τεχνικές αλιεύματος πληροφοριών από το internet (crawlers), συγκεντρώσαμε σε διάστημα 5 μηνών δεδομένα που αφορούν τους χρήστες αλλά και τα παιχνίδια. Με την προϋπόθεση ότι οι πληροφορίες ήταν δημόσιες(public) μπορούσαμε να ξέρουμε για κάθε χρήστη την χώρα προέλευσης, την ημέρα δημιουργίας του λογαριασμού, την τελευταία φορά που έπαιξε, ποια παιχνίδια έχει, πόσο έχει παίξει κάθε παιχνίδι, πόσοι έχουν κάθε παιχνίδι και διάφορες άλλες πληροφορίες. Για την διαδικασία αυτή χρειάστηκε να παράγουμε τυχαίους αριθμούς που πιθανώς αντιστοιχούσαν σε κάποιο αριθμό χρήστη και ζητάγαμε τις πληροφορίες του από τις ιστοσελίδες με την χρήση επαναληπτικού αλγορίθμου. Στην συνέχεια ακολούθησε φιλτράρισμα αυτών έτσι ώστε να απομονώσουμε μόνο τους πραγματικούς χρήστες.

Με στόχο να ανακαλύψουμε ομάδες χρήστων με κοινά χαρακτηριστικά, φιλτράραμε, διαλέξαμε, αναλύσαμε, και δημιουργήσαμε νέα γνώση με σκοπό να κατανοήσουμε ποια είναι αυτά τα στοιχεία που έχουν βαρύτητα στην επιλογή ενός παιχνιδιού από τον παίκτη.

Ακολούθησε ανάλυση του δείγματος με στατιστικά μέτρα με αποτέλεσμα να κατανοήσουμε την συμπεριφορά των παικτών παγκοσμίως ως προς την κατανάλωση, τον χρόνο που αφιερώνει αλλά και τις προτιμήσεις σε είδη παιχνιδιών. Στην συνέχεια πραγματοποιήσαμε εξόρυξη δεδομένων με χρήση του προγράμματος Weka. Τέλος, εφαρμόσαμε στους τελικούς πίνακες α) χρήστη, β) παιχνιδιών, συσταδοποίηση (clustering) με τον αλγόριθμο k-means και ταξινόμηση (classification) υπό την μορφή δέντρων αποφάσεων(decision trees) με τον αλγόριθμο j48.

Αφού ολοκληρώθηκε η διαδικασία ανακάλυψης γνώσης προχωρήσαμε στην δημιουργία κριτηρίων τα οποία αποτελούνται από συνδυαστικά υποκριτήρια των πινάκων χρήστη και παιχνιδιών. Ο αποφασίζων καλείται να αξιολογήσει τα κριτήρια και με την τεχνική ανάθεσης εργασίας σε πράκτορα[12], γίνεται η αντιστοίχιση της εργασίας(παιχνίδι) στον πράκτορα(αποφασίζων) ανάλογα με τα εκάστοτε βάρη. Η τελική απόφαση βασίζεται στα κριτήρια αξιολόγησης που χρησιμοποιεί ο αποφασίζων. Ένα πολύ σημαντικό χαρακτηριστικό αυτού του μοντέλου είναι η επανεκτίμηση των χαρακτηριστικών των παραγόντων κάθε φορά που εκχωρείται μια εργασία. Καταλήγοντας, δημιουργείται ο τελικός πολυκριτήριος πίνακας με χρήση της UTASTAR, όπου φαίνεται και η τελική κατάταξη προτιμήσεων για το χρήστη.

Κεφάλαιο 2: Μεθοδολογικό Πλαίσιο

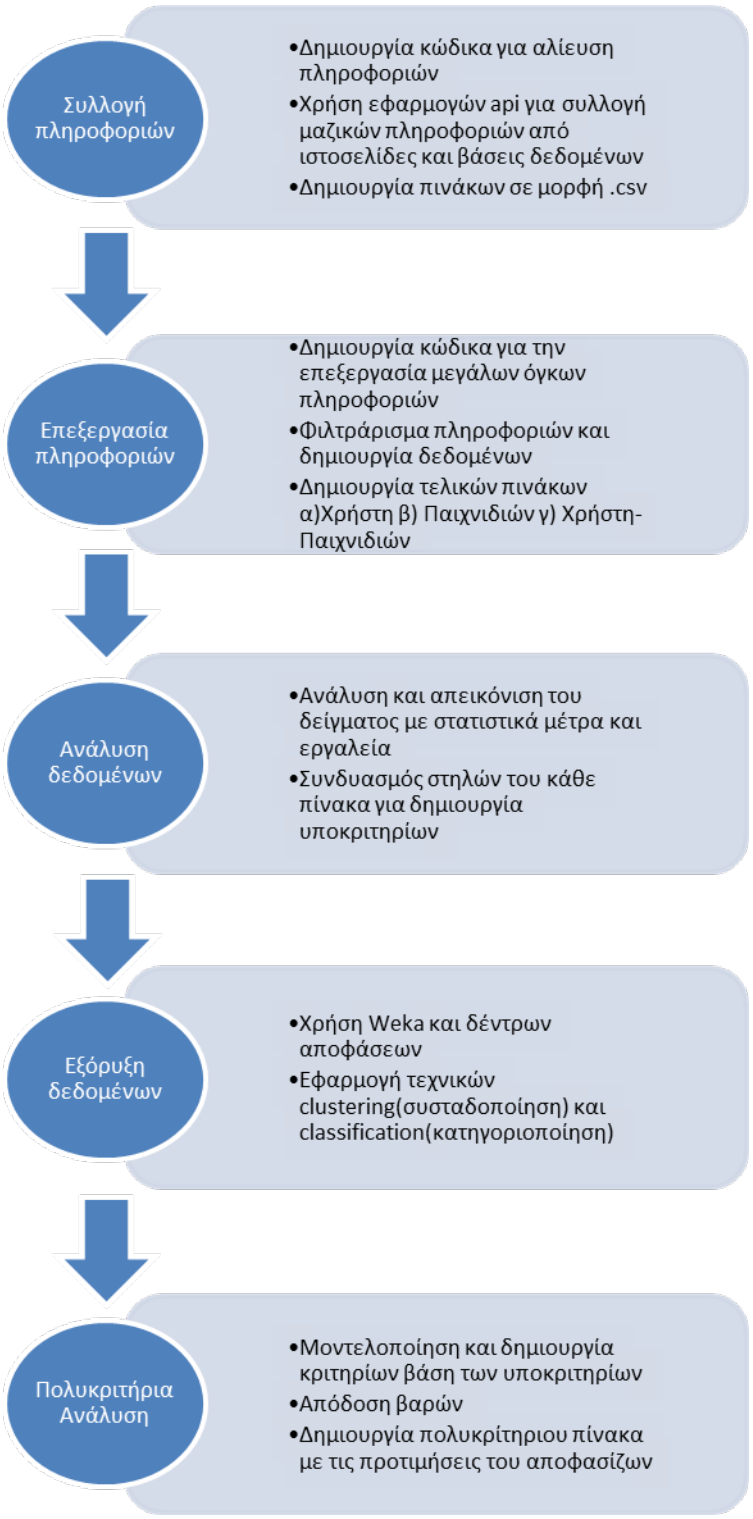
Ο τελικός σκοπός μας είναι να δημιουργήσουμε ένα πίνακα για κάθε χρήστη όπου θα περιλαμβάνει όλες τις εναλλακτικές, δηλαδή παιχνίδια, τα βάρη του χρήστη σε κάθε κριτήριο και την σειρά προτίμησης του χρήστη έτσι ώστε να τρέξουμε τον αλγόριθμο της Utastar και να βγάλουμε χρησιμότητες σε κάθε εναλλακτική αλλά και τα βάρη των κριτηρίων για κάθε χρήστη. Βέβαια για να το αντιμετωπίσουμε αυτό θα πρέπει να δημιουργήσουμε όλη αυτήν την πληροφορία από ακατέργαστα δεδομένα.

Το πρώτο σκέλος λοιπόν αφορά την συλλογή δεδομένων, για την διαδικασία αυτή απαιτήθηκε δημιουργία κατάλληλου κώδικα έτσι ώστε να εξορύξουμε τα δεδομένα και στην συνέχεια να τα αποθηκεύσουμε και να τα αναλύσουμε. Ο λόγος που χρειαζόμαστε την ανάλυση των δεδομένων είναι έτσι ώστε να καταλάβουμε την συμπεριφορά των παικτών και να δημιουργήσουμε κριτήρια βάση των χαρακτηριστικών από τους πίνακες. Έπρεπε να κατανοήσουμε ποια είναι αυτά τα χαρακτηριστικά τα οποία έχουν επίδραση στις επιλογές των παικτών. Έτσι λοιπόν, μετά την ανάλυση των δεδομένων με κλασσικά στατιστικά μέτρα, όπως μέση τιμή, μέγιστο-ελάχιστο αλλά και έλεγχος συγκεντρώσεων, οδηγηθήκαμε στην εξόρυξη δεδομένων. Με την συσταδοποίηση αναγνωρίσαμε συστάδες με κοινά χαρακτηριστικά ενώ με την κατηγοριοποίηση παρατηρήσαμε ποια χαρακτηριστικά αποτελούν μεγαλύτερο βαθμό σημαντικότητας στους παίκτες.

Με τις πληροφορίες αυτές μπορέσαμε να δημιουργήσουμε 4 κριτήρια τα οποία τα συνθέσαμε από χαρακτηριστικά των πινάκων που είχαμε συλλέξει. Για παράδειγμα, δημιουργήσαμε το κριτήριο Ποικιλία το οποίο αποτελούταν από το συνδυασμό χαρακτηριστικών Αριθμός παιχνιδιών χρήστη-Κάλυψη κατηγοριών παιχνιδιού και κάλυψη κατηγοριών χρήστη (κεφάλαιο 6.2).

Χρησιμοποιώντας τους ευρετικούς πίνακες κάθε κριτηρίου, δημιουργείται ένας πολυκριτήριος πίνακας για κάθε χρήστη όπου περιέχει όλες τις εναλλακτικές, τα κριτήρια, τα βάρη που αποδώθηκαν στα κριτήρια βάση των ευρετικών πινάκων και την σειρά προτίμησης.

Στο παρακάτω διάγραμμα διατυπώνεται η ακολουθία των διαδικασιών και η μεθοδολογία που εφαρμόστηκε σε κάθε βήμα.

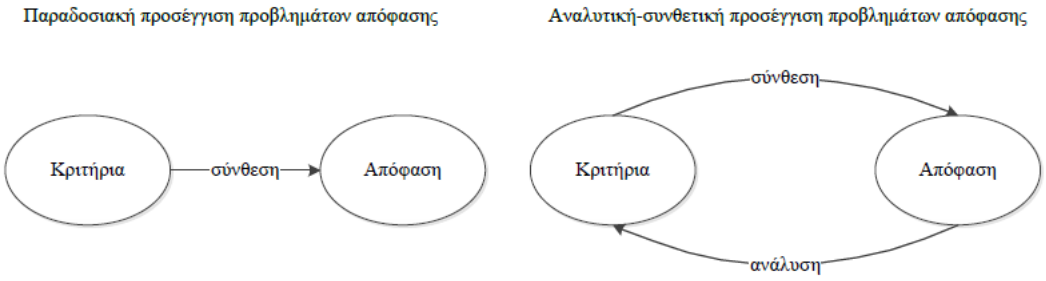


Διάγραμμα 2 Διαδικασία μεθοδολογίας

2.1 Αναλυτικά μοντέλα αποφάσεων. Οι μέθοδοι UTA και UTASTAR.

Γενικό μεθοδολογικό πλαίσιο

Τα μοντέλα της πολυκριτήριας ανάλυσης στην πλειονότητα τους βασίζονται στις αρχές της γραμμικότητας και της αιτιότητας, δηλαδή στη λογική ότι η απόφαση καθορίζεται από τα κριτήρια (συνθετική προσέγγιση-aggregation approach). Η αναλυτική-συνθετική προσέγγιση (aggregation-disaggregation approach), δέχεται ότι η απόφαση και τα κριτήρια επιδέχονται προοδευτική επεξεργασία αλληλοδομούμενα μέσα στο χρόνο, όπως αυτό φαίνεται στο σχήμα Η.1 (Jacquet-Lagrèze and Siskos,1982; Siskos et al., 2005).



Σχήμα 1 Παραδοσιακή και αναλυτική-συνθετική προσέγγιση προβλημάτων απόφασης (Πηγή: Σίσκος, 2008)

Η αναλυτική-συνθετική ή απλά αναλυτική προσέγγιση (disaggregation approach) εστιάζεται στη συσχέτιση των πραγματικών δεδομένων απόφασης και του μοντέλου απόφασης, έτσι ώστε να επιτυγχάνεται η μεγαλύτερη δυνατή συμβατότητα μοντέλου-αποφασίζοντος. Δηλαδή, στις μεθόδους της συγκεκριμένης προσέγγισης εκτιμώνται ή συμπεραίνονται οι παράμετροι εκείνες του ενός μοντέλου απόφασης οι οποίες επιτρέπουν την βέλτιστη ανασύσταση μιας απόφασης. Ουσιαστικά, πρόκειται για το γνωστό στους στατιστικολόγους παράδειγμα της επαγωγής-inference paradigm (Σίσκος, 2008).

Για τη διατύπωση της ολικής προτίμησης ενός αποφασίζοντος, οι Jacquet-Lagrèze and Siskos (1982) τονίζουν την αναγκαιότητα ύπαρξης ενός συνόλου δράσεων αναφοράς A_R (reference actions), το οποίο μπορεί να είναι:

- Ένα σύνολο προγενέστερων δράσεων (A_R: past actions)

- Ένα υποσύνολο των πραγματικών δράσεων του προβλήματος, ιδιαίτερα όταν το σύνολο Α είναι αρκετά μεγάλο
- Ένα σύνολο εικονικών δράσεων (fictitious actions), το οποίο μπορεί να αξιολογηθεί με ευκολία από τον αποφασίζοντα, ώστε αυτός να εκφράσει τις ολικές του προτιμήσεις (A_R: fictitious actions).

Σε καθεμία από τις παραπάνω περιπτώσεις, ζητείται από τον αποφασίζοντα να εξωτερικεύσει ή/και επιβεβαιώσει τις ολικές προτιμήσεις του στο σύνολο αναφοράς A_R, λαμβάνοντας υπόψη τις επιδόσεις των δράσεων αναφοράς σε όλα τα κριτήρια.

Η μέθοδος UTA

Η μέθοδος UTA (UTilités Additives) προτάθηκε από τους Jacquet-Lagrèze and Siskos το 1982 (Siskos et al., 2005). Στόχος της μεθόδου είναι η εκτίμηση μιας ή περισσότερων προσθετικών συναρτήσεων αξίας από μία προδιάταξη ενός συνόλου αναφοράς A_R, την οποία έχει διατυπώσει ο αποφασίζων. Η μέθοδος χρησιμοποιεί ειδικές τεχνικές γραμμικού προγραμματισμού για την εκτίμηση των συναρτήσεων αξίας, έτσι ώστε η κατάταξη που αποκτάται μέσω αυτών να είναι όσο πιο συμβατή γίνεται με την αρχική προδιάταξη (ή αρχική προτιμησιακή προδιάταξη) που έχει διατυπώσει ο αποφασίζων.

Το μοντέλο σύνθεσης των κριτηρίων (μοντέλο απόφασης) στη μέθοδο UTA είναι μία προσθετική συνάρτηση αξίας (additive value function) της ακόλουθης μορφής:

$$u(g) = \sum_{i=1}^{n} u_i(g_i^*) \qquad \text{(H.1')}$$

Υπό τους περιορισμούς κανονικοποίησης:

$$\begin{cases} \sum_{i=1}^{n} u_i(g_i^*) = 1, \\ u_i(g_i^*) = 0 \; \forall \; i = 1,2,\dots,n \end{cases} \qquad \text{(H.2')}$$

Όπου u_i, i=1,2,...,n είναι αύξουσες συναρτήσεις των g_i, οι οποίες συνήθως αναφέρονται ως περιθώριες ή μερικές συναρτήσεις αξίας (Marginal value functions).

Η ύπαρξη ενός τέτοιου μοντέλου προϋποθέτει την προτιμησιακή ανεξαρτησία των κριτηρίων (preferential independence) για τον αποφασίζοντα. Η ιδιότητα της συνέπειας ή μονοτονίας θα πρέπει να ισχύει, τόσο για τις περιθώριες όσο και για την ολική συνάρτηση αξίας. Στην τελευταία περίπτωση θα πρέπει να ισχύουν οι ακόλουθες ιδιότητες:

u[g(a)] > u[g(b)] ⇔ a > b (για την περίπτωση προτίμησης)

(H.3')

u[g(a)] = u[g(b)] ⇔ a ~ b (για την περίπτωση αδιαφορίας)

Χρησιμοποιώντας το προσθετικό μοντέλο (H.1') – (H.2') και λαμβάνοντας υπόψη τις σχέσεις προτίμησης (H.3'), η αξία κάθε εναλλακτικής a ∈ A_R μπορεί να γραφεί ως εξής:

$$u'[g(\mathrm{a})] = \sum_{i=1}^{n} u_i\left(g_i\ (\mathrm{a})\right) + \sigma(\mathrm{a})\ \forall\ \mathrm{a} \in A_R \qquad \text{(H.4')}$$

Όπου σ(a) είναι το ενδεχόμενο σφάλμα (μοναδικό σφάλμα στη UTA σε αντιδιαστολή με την UTASTAR που εισάγει 2 σφάλματα) σε σχέση με το u'[g(a)].

Για την εκτίμηση των αντίστοιχων περιθωρίων (μερικών) συναρτήσεων αξίας σε μια γραμμική κατά τμήματα μορφή, οι Jacquet-Lagrèze and Siskos προτείνουν τη χρήση της γραμμικής παρεμβολής. Έτσι, για κάθε κριτήριο, το διάστημα [g_{i*},g_i*] χωρίζεται σε (α_i-1) ίσα διαστήματα και τα τελικά σημεία g_i^j δίνονται από την σχέση:

$$g_i^j = g_{i*} + \frac{j-1}{a_i-1}(g_i^* - g_{i*})\ \forall\ j = 1,2,\dots,a_i \qquad \text{(H.5')}$$

Η περιθώρια (μερική) αξία μιας εναλλακτικής a υπολογίζεται με χρήση γραμμικής παρεμβολής, ως εξής:

$$u_i[g_i(\mathrm{a})] = u_i\left(g_i^j\right) + \frac{g_i(\mathrm{a})-g_i^j}{g^{j+1}-g_i^j}\left[u_i\left(g_i^{j+1}\right) - u_i\left(g_i^j\right)\ \gamma\iota\alpha\ g_i\ (\mathrm{a}) \in [g_i^j, g_i^{j+1}]\right. \qquad \text{(H.6')}$$

Η μέθοδος UTASTAR

Η μέθοδος UTASTAR προτάθηκε από τους Siskos and Yannacopoulos (1985) και αποτελεί μια βελτιωμένη έκδοση της πρωτότυπης μεθόδου UTA. Στην αρχική έκδοση της μεθόδου UTA (Jacquet-Lagrèze and Siskos,1982), για κάθε δράση a $\in$ A_R ορίζεται ένα μοναδικό σφάλμα σ(a) ενώ στην βελτιωμένη έκδοση της μεθόδου ορίζονται δύο σφάλματα που οδηγούν σε καλύτερα αποτελέσματα.

Ειδικότερα, για την εκτίμηση των περιθωρίων (μερικών) συναρτήσεων αξίας, οι κλίμακες μέτρησης κάθε κριτηρίου διακριτοποιούνται σε ένα σύνολο σημείων ως εξής:

$$G_i = \{g_{i*} = g_i^1, g_i^2, \dots, g_i^l, \dots, g_i^{ai} = g_i^*\}$$

Επίσης, το σύνολο αναφοράς A_R =$\{a_1, a_2, \dots, a_k\}$ «ανακατατάσσεται» με τέτοιο τρόπο, ώστε οι δράσεις να είναι διατεταγμένες σε μια σειρά προτίμησης, δηλαδή η a_1 αποτελεί την κεφαλή και η a_k την ουρά της κατάταξης. Δεδομένου ότι η συγκεκριμένη κατάταξη έχει τη μορφή μιας προδιάταξης R, για κάθε ζεύγος διαδοχικών δράσεων (a_j, a_{j+1}) ισχύει, είτε $\alpha_j > a_{j+1}$ (προτίμηση) είτε $\alpha_j \sim a_{j+1}$ (αδιαφορία).

Στη μέθοδο UTASTAR οι Siskos and Yannacopoulos εισάγουν μία διπλή θετικής συνάρτηση σφάλματος και έτσι ο τύπος Η.4’ γίνεται :

$$u'[g(\mathrm{a})] = \sum_{i=1}^{n} u_i\left(g_i\ (\mathrm{a})\right) - \sigma^+(a) + \sigma^-(a)\ \forall\ \mathrm{a} \in A_R \qquad \text{(H.7')}$$

Όπου σ^+ και σ^- είναι τα σφάλματα υποεκτίμησης και υπερεκτίμησης αντίστοιχα.

Επιπρόσθετα, μια άλλη σημαντική τροποποίηση αφορά τους περιορισμούς μονοτονίας των κριτηρίων, οι οποίοι μοντελοποιούνται με την βοήθεια των ακόλουθων μετασχηματισμών των μεταβλητών:

$$w_{ij} = u_i\left(g_i^{j+1}\right) - u_i\left(g_i^j\right) \geq 0 \quad \forall\ i = 1,2,\dots,n\ \kappa\alpha\iota\ j = 1,2,\dots,a_i - 1 \qquad \text{(H.8')}$$

Κατά αυτόν τον τρόπο, οι συνθήκες μονοτονίας του τύπου

$$u_i\left(g_i^{j+1}\right) - u_i\left(g_i^j\right) \geq\ s_i\ \ \forall\ i = 1,2,\dots,n\ \kappa\alpha\iota\ j = 1,2,\dots,a_i - 1$$

μπορούν να αντικατασταθούν από περιορισμούς μη αρνητικότητας των μεταβλητών w_{ij}.

Η μέθοδος UTASTAR, όπως και η UTA, χρησιμοποιεί ειδικές τεχνικές γραμμικού προγραμματισμού για την εκτίμηση των συναρτήσεων αξίας, έτσι ώστε η κατάταξη που αποκτάται μέσω αυτών να είναι όσο πιο συμβατή γίνεται με την αρχική προδιάταξη που έχει που διατυπώσει ο αποφασίζων. Τα βήματα που ακολουθεί είναι τα παρακάτω:

Βήμα 1.

Η ολική αξία των δράσεων του συνόλου αναφοράς $u[g(a_k)]$, k= 1,2,…,m εκφράζεται αρχικά ως συνάρτηση των περιθώριων (μερικών) αξιών $u_i(g_i)$ και στην συνέχεια των μεταβλητών w_{ij}, :

$$\begin{cases} u_i(g_i^1) = 0 \ \ \forall\, i = 1,2,\dots,n \\ u_i(g_i^j) = \sum_{i=1}^{j-1} w_{ij} \ \ \forall\, i = 1,2,\dots,n \ \kappa\alpha\iota \ j = 2,3,\dots,a_i - 1 \end{cases} \qquad \text{(H.9')}$$

Βήμα 2.

Εισάγονται 2 συναρτήσεις σφάλματος σ^+ και σ^- στο A_R γράφοντας για κάθε ζεύγος διαδοχικών δράσεων στην προδιάταξη τις αναλυτικές εκφράσεις:

$$\Delta(a_k, a_{k+1}) = u\ \ [g(a_k)] - \sigma^+(a_k) + \sigma^-(a_k) - u\ \ [g(a_{k+1})] + \sigma^+(a_{k+1}) - \sigma^-(a_{k+1})$$

Βήμα 3.

Επίλυση του ακόλουθου γ.π.

Minimize $\quad z = \sum_{k=1}^{m} [\,\sigma^+(a_k) + \sigma^-(a_k)]$

Υπό το σύνολο των περιορισμών

$$\begin{cases} \Delta(a_k, a_{k+1}) \geq \delta \ \ \varepsilon\acute{\alpha}\nu\ a_k > a_{k+1} \\ \Delta(a_k, a_{k+1}) = 0 \ \varepsilon\acute{\alpha}\nu\ a_k \sim a_{k+1} \end{cases} \quad \forall\, k \qquad \text{(H.10')}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{a_i} w_{ij} = 1$$

$$w_{ij} \geq 0, \ \ \sigma^+(a_k) \geq 0, \ \ \sigma^-(a_k) \geq 0 \quad \forall i, j \ \kappa\alpha\iota \ k$$

Βήμα 4.

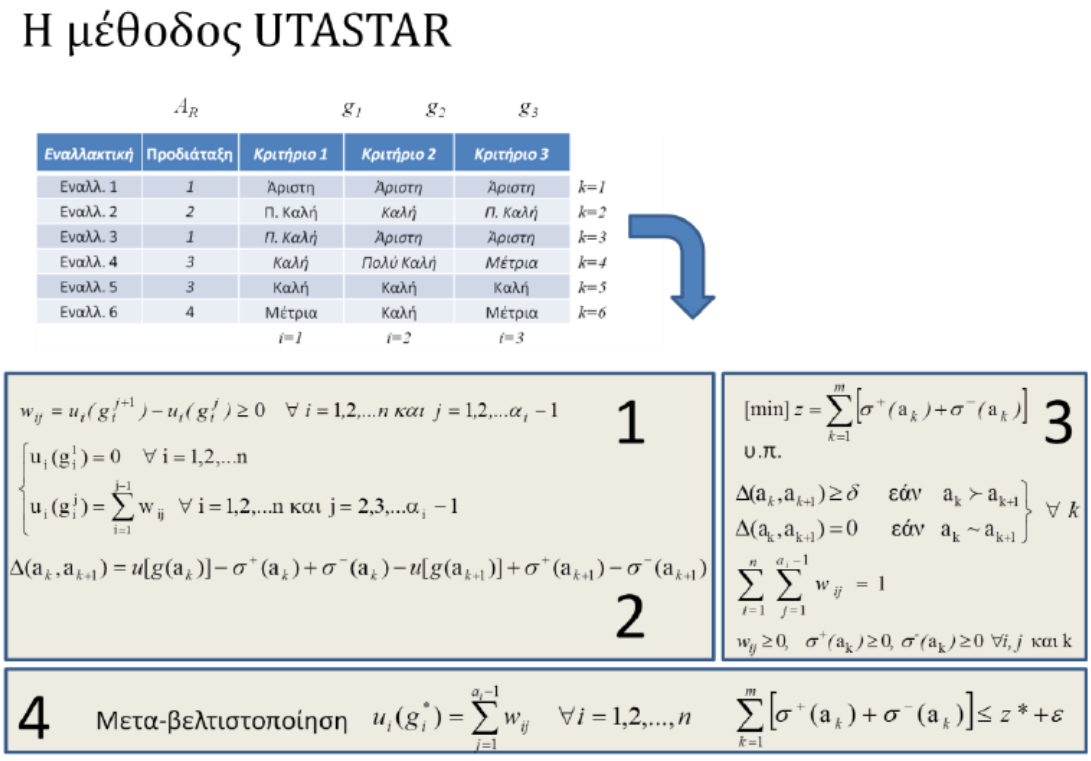
Ελέγχεται η ύπαρξη πολλαπλών βέλτιστων ή ημιβέλτιστων λύσεων στο γ.π. (H.10') υπολογίζοντας το βαρύκεντρο των προσθετικών συναρτήσεων αξίας που μεγιστοποιούν τις ακόλουθες αντικειμενικές συναρτήσεις:

$$u_i(g_i^*) = \sum_{j=1}^{a_i - 1} w_{ij} \quad \forall i = 1,2,\dots,n$$

Στο υπερπολύεδρο των περιορισμών του γ.π. (Η.10') που περιορίζεται από τον επόμενο νέο περιορισμό:

$$\sum_{k=1}^{m} [\sigma^+(a_k) + \sigma^-(a_k)] \leq z^* + \varepsilon$$

Όπου z* η βέλτιστη λύση του γ.π. στο βήμα 3 και ε ένας πολύ μικρός θετικός αριθμός ή μηδέν.



Σχήμα 2 Lakiotaki, Kleanthi. "An Integrated Recommender System Based on Multi-Criteria Decision Analysis and Data Analysis Methods: Methodology, Implementation and Evaluation." December (2010)

2.2 Εξόρυξη δεδομένων

Η εκρηκτική αύξηση των δεδομένων έχει σαν αποτέλεσμα να έχουμε πλέον στη διάθεσή μας τεράστιους όγκους δεδομένων του επιπέδου των terabytes και petabytes.

Το πρόβλημα αυτό, λύθηκε μέσω της ανάπτυξης μεθόδων και τεχνικών αναζήτησης γνώσης μέσα από τεράστιους όγκους δεδομένων. Έτσι, προέκυψε η εξόρυξη δεδομένων ή η ανακάλυψη γνώσης από βάσεις δεδομένων (data mining or knowledge discovery from data).

Εξόρυξη γνώσης είναι ένα σύνολο αποδοτικών τεχνικών για να αναλύσουμε πολύ μεγάλες συλλογές από δεδομένα και να εξάγουμε χρήσιμες πληροφορίες –γνώσεις από αυτά.

Κύριες πηγές αυτών των άφθονων δεδομένων είναι:

- Οι επιχειρήσεις,
- Το διαδίκτυο,
- Το ηλεκτρονικό εμπόριο και οι ηλεκτρονικές συναλλαγές & δοσοληψίες, ...
- Η επιστημονική έρευνα και τα αποτελέσματά της όπως από την υγεία, το μάρκετινγκ, την τηλεπισκόπηση, τη βιοπληροφορική, τις εικόνες, τα video, ειδήσεις, ψηφιακές φωτογραφίες, social media, κλπ.

Η εξόρυξη δεδομένων ή η ανακάλυψη γνώσης από βάσεις δεδομένων (data mining or knowledge discovery from data) αφορά την εξαγωγή ενδιαφέρουσας, μη τετριμμένης, προηγούμενα άγνωστης και πιθανά χρήσιμων προτύπων ή γνώσης από την τεράστιους όγκους δεδομένων.

Η εξόρυξη γνώσης από δεδομένα (Data Mining) αφορά τη χρήση αλγορίθμων για την εξαγωγή πληροφοριών και προτύπων που εξάγονται μέσω της διαδικασίας ανακάλυψης γνώσης (Knowledge Discovery in Databases - KDD).

Η ανακάλυψη γνώσης από δεδομένα (Knowledge Discovery in Databases – KDD) είναι η διαδικασία ανεύρεσης χρήσιμων πληροφοριών και προτύπων σε δεδομένα.

Με την εξόρυξη δεδομένων, δημιουργούμε πρότυπα και μοντέλα τα οποία είναι

- Έγκυρα: Κρατάμε τα νέα δεδομένα με κάποια βεβαιότητα
- Χρήσιμα: Θα πρέπει να είναι δυνατόν να δουλέψουμε με αυτά
- Απροσδόκητα: Μη προφανή στο σύστημα

- Κατανοητά: Οι άνθρωποι θα πρέπει να είναι σε θέση να ερμηνεύσουν το πρότυπο

Τα βήματα που ακολουθούνται για την ανακάλυψη νέας γνώσης, την εξόρυξη της από ακατέργαστα δεδομένα είναι:

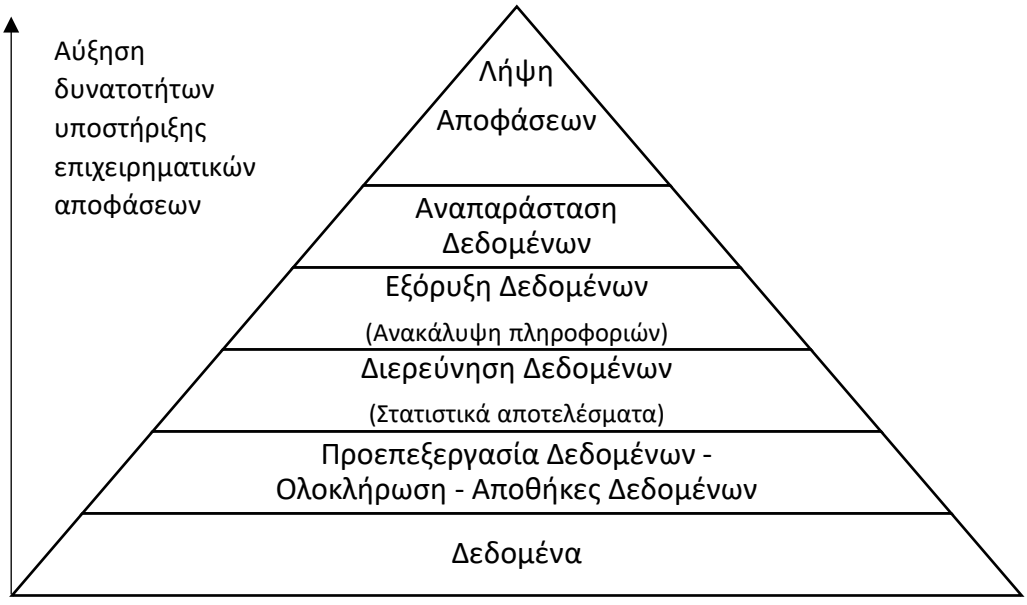
Επιλογή: Στο στάδιο αυτό συγκεντρώνονται, τα απαραίτητα για την ανάλυση, δεδομένα από διάφορες και ετερογενείς πηγές (βάσεις δεδομένων, διαδίκτυο, κλπ) και δημιουργούνται οι κατάλληλες βάσεις δεδομένων.

Προεπεξεργασία: Γίνεται έλεγχος των δεδομένων που έχουν αποθηκευτεί για λάθη καταχώρησης, για το αν λείπουν δεδομένα, για διαφορετικούς τύπους δεδομένων, κλπ

Μετατροπές: Τα δεδομένα μας πρέπει να έχουν κοινή μορφή και για αυτό ίσως χρειαστούν κάποιες μετατροπές. Έτσι μπορεί να γίνουν μετατροπές δεδομένων από μια κατηγορία σε μια άλλη (να ανακωδικοποιηθούν), να μειωθούν οι διαστάσεις τους, κοκ.

Εξόρυξη γνώσης από δεδομένα: Στο στάδιο αυτό εφαρμόζονται οι κατάλληλοι αλγόριθμοι ώστε εξαχθούν τα κατάλληλα αποτελέσματα.

Ερμηνεία & Αξιολόγηση: Παρουσίαση των αποτελεσμάτων με κατάλληλο τρόπο ώστε να γίνουν κατανοητά και χρήσιμα.



Σχήμα 3 Διαδικασία ανακάλυψης γνώσης, Ματσατσίνης (2010, p. 1008).

Όσο υψηλότερα φτάνει κάποιος στην πυραμίδα τόσες περισσότερες ουσιώδης πληροφορίες έχει. Ο αναλυτής θα φτάσει μέχρι την αναπαράσταση των δεδομένων και στην συνέχεια ο αποφασίζον καλείται να αποφασίσει με βάση τις πληροφορίες που έχει.

Τα δεδομένα των πραγματικών εφαρμογών μπορεί να είναι ασυνεπή, ελλιπή και/ή να περιέχουν θορύβους. Αυτό σημαίνει ότι, αν τα προς επεξεργασία δεδομένα δεν είναι καλής ποιότητας τότε και τα αποτελέσματα της εξόρυξης γνώσης από αυτά δεν θα είναι ποιοτικά.

Τα προβλήματα οφείλονται κυρίως σε προβλήματα καταχώρησης δεδομένων, μετάδοσης δεδομένων ή συλλογής δεδομένων. Περιέχουν σφάλματα, ακραίες τιμές ή θορύβους, διπλές εγγραφές, ελλιπή ή ελλείποντα δεδομένα (τιμές χαρακτηριστικών), αντιφάσεις σε δεδομένα (ασυμφωνίες σε ονομασίες, κωδικούς κλπ).

Άμα τα δεδομένα που έχουμε δεν έχουν τύχει προεπεξεργασίας, τίθενται διάφορες προβληματικές όπως:

- Η λήψη αποφάσεων τίθεται σε κίνδυνο.
- Μπορεί να μην είναι αξιόπιστη η απόφαση.
- Έχουμε καλύτερες πιθανότητες να ανακαλύψουμε χρήσιμες γνώσεις, όταν τα δεδομένα είναι καθαρά.

2.2.1 Λειτουργία ταξινόμησης- Classification

Η λειτουργία ταξινόμησης είναι η διαδικασία της κατάταξης (ή χαρτογράφησης) ενός δεδομένου μέσα σε μία από πολλές προκαθορισμένες κλάσεις. Στο πεδίο του Web, ενδιαφέρον παρουσιάζει η ανάπτυξη ενός προφίλ για χρήστες που ανήκουν σε μία συγκεκριμένη κλάση ή κατηγορία. Αυτή η διαδικασία απαιτεί εξαγωγή και επιλογή των χαρακτηριστικών εκείνων που περιγράφουν καλύτερα τις ιδιότητες μίας συγκεκριμένης κλάσης ή κατηγορίας.

Το classification μπορεί να επιτευχθεί χρησιμοποιώντας αλγορίθμους επαγωγής που υπάγονται στην ευρύτερη κατηγορία μάθησης με επίβλεψη. Τέτοιοι είναι decision tree classifiers (κατηγοριοποίηση με δένδρα αποφάσεων), naive Bayesian classifiers (κατηγοριοποίηση με βάση τον απλό Bayes), k-means (βλ. Ενότητα 2.2.3) , μηχανές Διανυσμάτων Υποστήριξης (support vector machines), νευρωνικά δίκτυα, ταξινόμηση βασισμένη σε κανόνες, ταξινόμηση βασισμένη σε πρότυπα, λογιστική παλινδρόμηση, και άλλοι. Τα νέα δεδομένα κατηγοριοποιούνται με βάση το σύνολο εκπαίδευσης.

Τυπικές εφαρμογές εφαρμόζονται σε ανίχνευση απάτης σε πιστωτικές κάρτες, απευθείας μάρκετινγκ, ταξινόμηση αστεριών, ασθενειών, ιστοσελίδων, κα.

2.2.2 Λειτουργία Συσταδοποίησης – Clustering

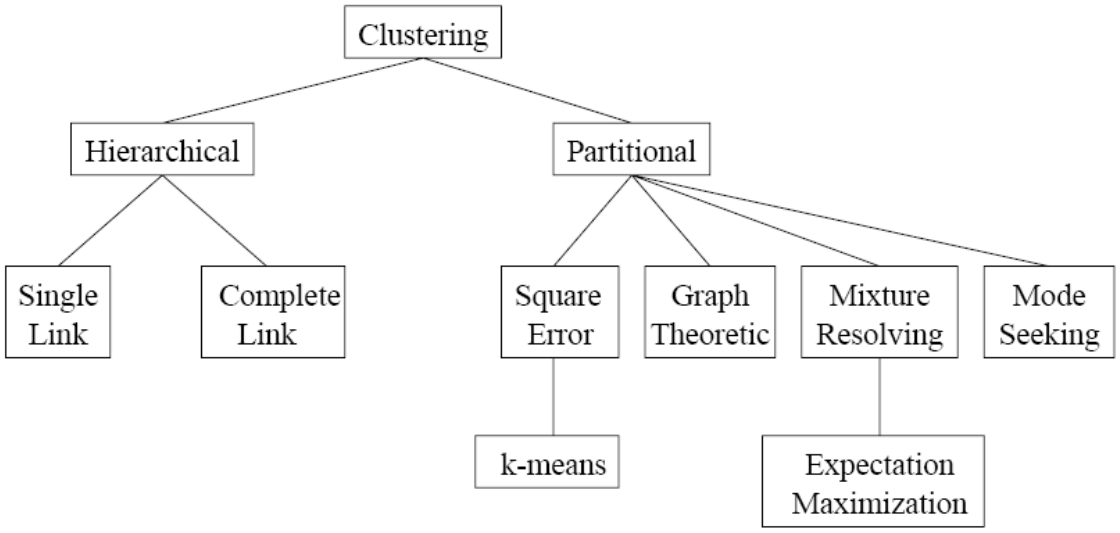
Ένας απλός ορισμός για την ομαδοποίηση ή συσταδοποίηση (clustering): ομαδοποίηση ονομάζεται η διαδικασία που οργανώνει πρότυπα (παρατηρήσεις, δεδομένα ή διανύσματα χαρακτηριστικών) σε ομάδες (συστάδες-clusters), όπου τα μέλη μιας ομάδας είναι παρόμοια μεταξύ τους σύμφωνα με κάποιο κριτήριο. Σκοπός είναι να προσδιοριστούν οι ομάδες που ανήκουν διάφορες ποσότητες δεδομένων, με βάση κάποια κριτήρια ομοιογένειας. Η τεχνική της ομαδοποίησης υπάγεται στην ευρύτερη κατηγορία των τεχνικών μάθησης χωρίς επίβλεψη. Η διαφορά της ομαδοποίησης δεδομένων (data clustering) από την ταξινόμηση δεδομένων (data classification) είναι ότι, στην ταξινόμηση οι ομάδες στις οποίες θα τοποθετηθούν τα δεδομένα είναι προκαθορισμένες. Αυτό σημαίνει, ότι είναι εκ των προτέρων γνωστός ο αριθμός των ομάδων, τα ονόματα και οι ταυτότητες τους. Είναι και αυτό ένα σύστημα μάθησης μιας και οι ετικέτες που δίνονται από τα διαθέσιμα πρότυπα χρησιμοποιούνται ώστε να μάθει το σύστημα ταξινόμησης την περιγραφή κάθε κλάσης και να είναι σε θέση να ταξινομήσει ένα νέο πρότυπο. Αντίθετα, στην ομαδοποίηση δεδομένων τονίζεται ιδιαίτερα ότι οι ομάδες δεν προϋπάρχουν αλλά αποφασίζονται από τον αλγόριθμο κατά δυναμικό τρόπο. Στην ομαδοποίηση δεδομένων δηλαδή, υπάρχει ένα σύνολο δεδομένων το οποίο πρέπει να διαχειριστεί ώστε από αυτό να προκύψουν δυναμικά οι ομάδες (είναι δηλαδή data driven). Σκοπός είναι να δημιουργηθούν ομάδες, που η καθεμία από αυτές θα συγκεντρώνει ομοιογενή στοιχεία. Κάθε μία από αυτές τις ομάδες διατηρεί ένα κέντρο, συνήθως το πιο κεντρικό στοιχείο της.

Στόχος είναι η μεγιστοποίηση ενδο-συσταδικής ομοιότητας & ελαχιστοποίηση της δια-συσταδικής ομοιότητας

Συνήθως το σύνολο δεδομένων εισόδου χωρίζεται σε:

- ένα σύνολο εκπαίδευσης (training set), και
- ένα σύνολο ελέγχου (test set)

Το σύνολο εκπαίδευσης χρησιμοποιείται για να κατασκευαστεί το μοντέλο, ενώ το σύνολο ελέγχου για την επικύρωση του μοντέλου.



Σχήμα 4 Κατηγοροποίηση τεχνικών συσταδοποίησης-clustering, Galpin, Ixent et. al. (2018)

2.2.3 K-means

Διαχωριστικός αλγόριθμος ο οποίος κατασκευάζει Κ συστάδες (είσοδος στο πρόβλημα). Κάθε συστάδα συσχετίζεται με ένα κεντρικό σημείο (centroid). Κάθε σημείο συσχετίζεται με την κοντινότερη σε αυτό συστάδα (Κ συστάδες – Κ κεντρικά σημεία). Ο αριθμός των ομάδων, Κ, είναι είσοδος στον αλγόριθμο.

Δεδομένου k, ο αλγόριθμος k-means υλοποιείται σε τέσσερα στάδια:

1. Διαχωρισμός αντικειμένων σε k μη κενά υποσύνολα
2. Υπολογίζει τα κεντρικά σημεία (centroids) που θα είναι τα κέντρα βάρους των συστάδων του τρέχοντος διαχωρισμού (το κέντρο βάρους είναι το κέντρο, δηλαδή το μέσο σημείο της συστάδας)
3. Ανέθεσε κάθε αντικείμενο στη συστάδα με το κοντινότερο κεντρικό σημείο
4. Πηγαίνετε πίσω στο βήμα 2. Η διαδικασία σταματά όταν η ανάθεση ολοκληρωθεί

Ο διαμεριστικός αλγόριθμος k-means είναι ένας από τους πιο απλούς και δημοφιλέστερους αλγορίθμους ομαδοποίησης που ανήκουν στην ευρύτερη κατηγορία των τεχνικών μάθησης χωρίς επίβλεψη. Ο αλγόριθμος αυτός είναι δημοφιλής εξαιτίας της απλότητας της υλοποίησης του και της

γραμμικής πολυπλοκότητας του η οποία είναι της τάξης n (O(n)), όπου n το σύνολο των στοιχείων. Η διαδικασία της ομαδοποίησης ενός συνόλου δεδομένων με βάση τον k-means είναι εύκολη, αρκεί να είναι εκ των προτέρων καθορισμένος ο αριθμός (k) των clusters (ομάδων) που θα προκύψουν. Η κύρια ιδέα είναι να προσδιοριστούν αρχικά k centroids (κεντροειδή), ένα για κάθε cluster. Αυτά τα αρχικά centroids πρέπει να επιλεγούν με επιδέξιο τρόπο, γιατί διαφορετικές αρχικές θέσεις για τα centroids δίνουν διαφορετικά αποτελέσματα. Δηλαδή, η αρχική θέση των centroids επηρεάζει το αποτέλεσμα που θα δώσει ο αλγόριθμος. Έτσι, συχνά θεωρείται καλύτερη η επιλογή εκείνων των centroids ώστε να απέχουν μεταξύ τους όσο περισσότερο γίνεται. Το επόμενο βήμα είναι επιλογή κάθε στοιχείου από το σύνολο δεδομένων και συσχέτιση του με το κοντινότερο σε αυτό centroid. Όταν αυτό γίνει για όλα τα στοιχεία του συνόλου δεδομένων, το πρώτο βήμα έχει ολοκληρωθεί και μία πρώτη και «πρόχειρη» ομαδοποίηση έχει ήδη προκύψει. Στη συνέχεια, απαιτείται να υπολογιστούν ξανά k νέα centroids, τα οποία θα αποτελούν το κέντρο βάρους για κάθε ένα cluster που προέκυψε από το προηγούμενο βήμα. Αφού λοιπόν οριστούν τα νέα k centroids, ακολουθεί και πάλι η ίδια διαδικασία ανάθεσης καθενός από τα στοιχεία του συνόλου δεδομένων στο κοντινότερο με αυτό, νέο πλέον, centroid. Έτσι, γίνεται μια επανάληψη της ίδιας διαδικασίας. Αποτέλεσμα αυτής της επανάληψης είναι ότι σε κάθε βήμα τα centroids αλλάζουν θέση (ορίζονται νέα) και τα στοιχεία ανατίθενται στο κατάλληλο cluster κάθε φορά με βάση το κοντινότερο centroid. Όταν σε κάποια επανάληψη δεν σημειωθούν αντιμεταθέσεις στοιχείων, τότε τερματίζει η εκτέλεση του αλγορίθμου. Το αποτέλεσμα που προκύπτει είναι η ομαδοποίηση του συνόλου δεδομένων σε k clusters.

Ο αλγόριθμος στοχεύει να ελαχιστοποιήσει μία αντικειμενική συνάρτηση, την λεγόμενη συνάρτηση τετραγωνικού λάθους που ορίζεται ως εξής: $J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^j - c_j\right\|^2$, $όπου\ \left\|x_i^j - c_j\right\|^2$ είναι ένα μέτρο απόστασης που χρησιμοποιείται για να μετρά την απόσταση κάθε στοιχείου x_i^j από το centroid c_j του κάθε cluster. Όπου n το σύνολο των στοιχείων του συνόλου δεδομένων.

Ο παρακάτω πίνακας δείχνει συνοπτικά τα βήματα του αλγορίθμου k-means:

Είσοδος:

$D = \{x_1, x_2, \ldots, x_n\}$ // Σύνολο στοιχείων

k // Αριθμός επιθυμητών clusters

Έξοδος:

k // Σύνολο clusters

k-Means αλγόριθμος:

Ανέθεσε τιμές στα αρχικά centroids $C_1, C_2, \ldots, C_k$;

Επανέλαβε

Ανέθεσε κάθε X_i στο cluster με του οποίου το centroid η απόσταση είναι η μικρότερη;

Υπολόγισε νέα centroids για κάθε cluster;

Μέχρι να συναντηθεί το κριτήριο σύγκλισης;

Σχήμα 5 Βήματα αλγορίθμου k-means

Αν και μπορεί να αποδειχθεί ότι ο αλγόριθμος πάντα τερματίζει, αξίζει να τονιστεί ότι δεν καταφέρνει πάντα να βρίσκει τη βέλτιστη λύση. Ο αλγόριθμος επηρεάζεται σημαντικά από τα αρχικά centroids. Για αυτό πολλές φορές συνίσταται η εκτέλεση του πολλές φορές μέχρι να μειωθεί η επίδραση αυτή.

Έστω ότι υπάρχουν n διανύσματα τα $x_1, x_2, \ldots, x_n$ και όλα είναι της ίδια διάστασης. Ακόμη είναι γνωστό ότι όλα εμπίπτουν σε k συμπαγή clusters, για k<n. Έστω m_i είναι το μέσο διάνυσμα του i cluster. Εφόσον τα clusters είναι σαφώς διαχωρισμένα μεταξύ τους, μπορεί να χρησιμοποιηθεί σαν μέτρο απόστασης μεταξύ των στοιχείων η Ευκλείδεια απόσταση ή και άλλα δημοφιλή μέτρα απόστασης, που έχουν αναλυθεί σε προηγούμενο κεφάλαιο. Αυτό σημαίνει ότι σε κάθε βήμα θα λέγεται: το στοιχείο x ανήκει στο cluster i, εάν η Ευκλείδεια απόσταση του από το centroid του i cluster είναι η μικρότερη σε σχέση με όλες τις άλλες αποστάσεις του από τα centroids των άλλων clusters. Έτσι βρίσκονται οι Ευκλείδειες αποστάσεις για όλα τα στοιχεία και κάθε ένα από αυτά ανατίθεται στο cluster από του οποίου το centroid απέχει λιγότερο (δηλαδή η Ευκλείδεια απόσταση είναι η μικρότερη). Στην συνέχεια υπολογίζονται τα νέα centroids και μετά πάλι οι Ευκλείδειες αποστάσεις όλων των στοιχείων για τα νέα centroids. Γίνονται οι κατάλληλες μετακινήσεις στοιχείων και η ίδια διαδικασία επαναλαμβάνεται μέχρι κανένα στοιχείο να μην μετακινείται σε άλλο cluster, δηλαδή τα clusters να μένουν αμετάβλητα.

2.3 Υφιστάμενη κατάσταση

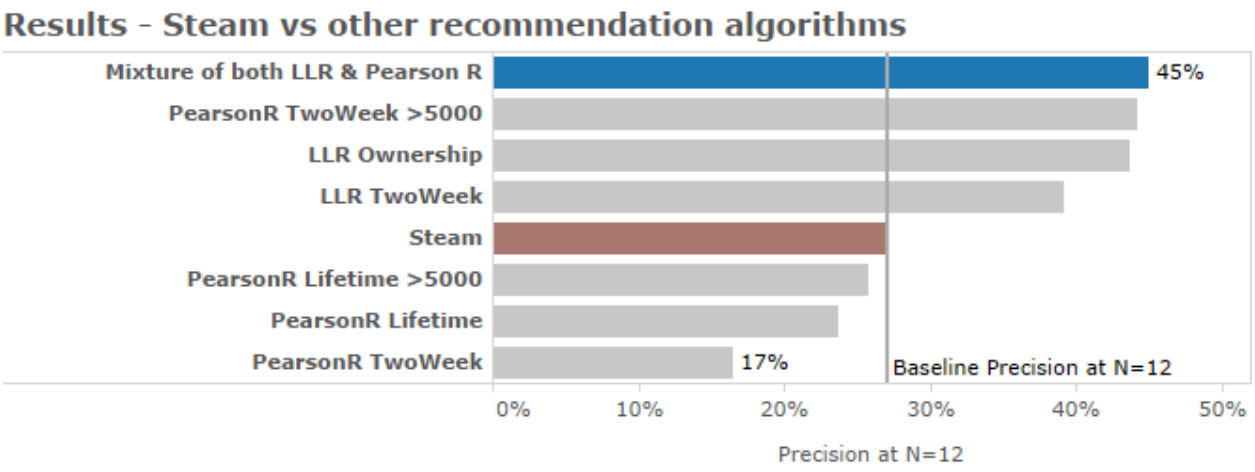
Οι (Sifa, Bauckhage, & Drachen, 2014a) χρησιμοποίησαν 2 μονοκριτηριακές μεθόδους, Factor Oriented και Neighborhood Oriented models, προτείνοντας στον χρήστη ένα L πιθανό αριθμό παιχνιδιών βάση του ιστορικού τους σε απόλυτο χρόνο παιχνιδιού στα παιχνίδια που είχαν στην κατοχή τους. Ενώ από την δουλειά των (Sifa, Bauckhage, & Drachen, 2014b) παρατηρείται ότι ο χρόνος που δαπανάται στα παιχνίδια από τους χρήστες μπορεί να μοντελοποιηθεί με μια κατανομή Weibull. Η Lakiotaki (2010) κατασκεύασε ένα σύστημα προσωποποιημένων συστάσεων για ταινίες. Εξάγοντας δεδομένα από χρήστες δημιούργησε ένα πολυκριτήριο πίνακα. Στην συνέχεια με χρήση της Utastar και του αλγορίθμου Disaggregation-Aggregation (Y. Siskos et al., 2005) απέδωσε ειδικά βάρη στα κριτήρια τα οποία χρησιμοποίησε στον αλγόριθμο k-means έπειτα από μία ομαδοποίηση (clustering) για να αποδώσει την τελική σύσταση ταινιών στον χρήστη. Το αποτέλεσμα ήταν η δημιουργία ενός προγράμματος που ονομάζεται UtaRec System. Τέλος, οι Matsatsinis & Delias (2003) εισαγάγανε το AgentAllocator κατά το οποίο πολλές ασχολίες διανέμονται στους πράκτορες, ανάλογα με τα βάρη και τα κριτήρια που έχει θέσει ο αποφασίζων.

Ακόμα οι (Sifa, Bauckhage, Drachen, 2014) με μία έρευνα παρατήρησαν ότι τα παιχνίδια χωρίζονται στις παρακάτω κατηγορίες, με ένα μεγάλο ποσοστό 23.3% να είναι ανώνυμες μάρκες που δεν έχουν προβληθεί αρκετά στο κοινό.

Πίνακας 1 Κατανομή συχνοτήτων ηλεκτρονικών παιχνιδιών (Sifa, Bauckage, Drachen,2014)

Archetype	Common Game Types	Representation
z1	FPS, Action and Indie	44.1%
z2	F2P RPG and Adventure	10.2%
z3	Adventure and Point & Click	22.4%
z4	AAA Games	23.3%

Σε ένα αντίστοιχο πρόβλημα έχει δώσει λύση ο δημιουργός [20] της ιστοσελίδας Kevin Wong, ο οποίος βλέποντας την ανάγκη για εξατομικευμένες και αντικειμενικές συστάσεις παιχνιδιών, δημιούργησε αυτήν την πλατφόρμα. Συλλέγοντας και εκείνος πληροφορίες από το Steam Web API και το online store της Valve βασίστηκε δε κυρίως σε 2 αλγόριθμους επιλέγοντας κάθε φορά ένα κριτήριο για την έρευνα του. Συγκεκριμένα χρησιμοποίησε την μέθοδο Pearson R Correlation και την Log-Likelihood Ratio έχοντας ως κριτήρια το συνολικό χρόνο παιχνιδιών, το συνολικό χρόνο παιχνιδιών τις τελευταίες 2 εβδομάδες , τα παιχνίδια που κατέχει ο χρήστης και τέλος τα παιχνίδια που έπαιξε τις τελευταίες 2 εβδομάδες. Ο αλγόριθμος που έδωσε τα καλύτερα αποτελέσματα ήταν ένα μίγμα των δύο μεθόδων βασισμένο στα παιχνίδια που έπαιξε τις τελευταίες 2 εβδομάδες ο χρήστης.



Διάγραμμα 3 Αποδοτικότητα αλγορίθμου συστάσεων του Steam με άλλους προτεινόμενους, Kevin Wong (2014)

Κεφάλαιο 3: Συλλογή Δεδομένων

3.1 Περιβάλλον του Steam Web Api

Πληροφορίες

Για την εργασία χρησιμοποιήσαμε δεδομένα για χρήστες του Steam. Το Steam (https://store.steampowered.com/) αποτελεί μία ολοκληρωμένη πλατφόρμα διανομής παιχνιδιών με επιπρόσθετες λειτουργίες όπως κοινωνική δικτύωση κ.α. Επιτρέπει στους χρήστες που διαθέτουν λογαριασμό να διαχειρίζονται βιβλιοθήκες με παιχνίδια που ανήκουν στην εταιρεία (Valve) είτε τίτλους από τρίτους.

Η συλλογή των δεδομένων έγινε αρχικά με χρήση του Steam Web Api (https://steamcommunity.com/dev) , το οποίο αποτελεί ένα εργαλείο της Valve. Διανέμεται δωρεάν σε προγραμματιστές και με την χρήση ενός ειδικού κλειδιού μπορεί κάποιος να στέλνει εντολές με ερωτήματα και να του επιστρέφει τα δεδομένα. Σχετικά με την λειτουργία του Steam web Api, οι αιτήσεις που μπορείς να κάνεις στο σύστημα περιορίζονται στις 100.000 API calls κάθε μέρα και επιστρέφονται σε JSON format(JavaScript Object Notation). Η μορφή JSON αποτελεί ένα σύστημα μετάδοσης πληροφοριών σε μορφή διανυσμάτων κυριώς στο προγραμματισμό ιστοσελίδων και είναι ανεξάρτητα από τη προγραμματιστικές γλώσσες.

Τέλος υπάρχουν κάποιες προβληματικές οι οποίες διαμορφώνουν και τις πληροφορίες που μπορούμε να συλλέξουμε.

1) Η λειτουργία του API ξεκίνησε το 2009 οπότε δεν έχουμε πληροφορίες και στατιστικά πριν από το 2009 παρόλο που ο πρώτος χρήστης εγγράφηκε το 2003, οπότε για παιχνίδια που παίζονταν πριν το 2009 κυρίως δεν θα έχουμε σωστά αποτελέσματα.

2) Δεν υπάρχουν πληροφορίες για αν το Steam κρατάει στατιστικά ακόμα και όταν ο χρήστης δεν είναι Online και παίζει κάποιο παιχνίδι που δεν χρειάζεται σύνδεση στο Internet.

3) Υπάρχουν κατηγορίες από παίχτες ότι ο συνολικός χρόνος παιχνιδιού δεν ταυτίζεται με τον χρόνο που δίνει ως αποτέλεσμα το API και αυτό αποδίδεται ίσως στο ότι κάποια παιχνίδια χρησιμοποιούν διαφορετικές πλατφόρμες. Για παράδειγμα, μπορείς να κατεβάσεις και να παίξεις το παιχνίδι μέσα από το Steam αλλά μπορείς να πας και στην ιστοσελίδα του παιχνιδιού και να κατεβάσεις την δικιά τους πλατφόρμα

4) Το Steam δίνει την δυνατότητα στον χρήστη να επιλέξει τον τύπο του λογαριασμού του.

- Public (προεπιλογή)
- Friends Only (Τα στοιχεία του είναι ορατά μόνο σε φίλους)
- Private (Τα στοιχεία του φαίνονται μόνον στον ίδιο)

Οπότε εμείς έχουμε πληροφορίες μόνο για τους χρήστες που έχουν το προφίλ τους δημόσιο.

Στην συνέχεια πήραμε δεδομένα από την ιστοσελίδα http://store.steampowered.com/, τα οποία κυρίως συσχετίζονταν με τις εφαρμογές- παιχνίδια και αφορούσαν το είδος του παιχνιδιού(Action, Strategy …), συνολικές βαθμολογίες χρηστών (global rankings), τιμές και άλλα.

Για την εξαγωγή των παραπάνω δεδομένων χρησιμοποιήσαμε την γλώσσα Python καθώς και διάφορες βιβλιοθήκες. Ένα ακόμα πολύ ενδιαφέρον στοιχείο ήταν ότι ο τρόπος λειτουργίας του Steam Web Api προϋποθέτει ότι ο χρήστης θα πληκτρολογήσει τον μοναδικό του User ID που έχει στο Steam οπότε βρήκαμε τον τρόπο με τον οποίο παράγονται τα User id και ελέγχαμε αν υπήρχαν πληροφορίες για το εκάστοτε Steam id αλλιώς δεν το χρησιμοποιούσαμε.

3.1.1 Steam ID

Το Steam ID είναι ένας μοναδικός αριθμός που αντιπροσωπεύει κάθε χρήστη στην κοινότητα του Steam. Αποτελείται από 4 στοιχεία και καθένα από αυτά υποδηλώνει συγκεκριμένες ιδιότητες του λογαριασμού. Συγκεκριμένα αποτελείται από :

- Universe – από ποιο σύστημα προέρχεται ο λογαριασμός, συνήθως είναι Public
- Account Type – τι τύπος λογαριασμού είναι
- Instance
- Account ID

Η αναπαράσταση του με γράμματα ακολουθεί το μοτίβο '[C:U:A]' or '[C:U:A:I]' ανάλογα με τον τύπο του Steam ID

- C είναι ένας χαρακτήρας που υποδηλώνει το τύπο του λογαριασμού ή και μία μίξη με το Instance που ανήκει
- U είναι το Universe

- A είναι το Account ID.
- I είναι το Instance ID. Εάν δεν υπάρχει τότε έχει σαν προεπιλεγμένη τιμή το ‘C’

Όταν απεικονίζεται σε προγράμματα τότε χρησιμοποιείται μία μορφή 64-bit.

- Τα πρώτα 32 bits περιλαμβάνουν το Account ID.
- Τα επόμενα 20 bits περιλαμβάνουν το Instance του λογαριασμού. Για ατομικούς λογαριασμούς συνήθως είναι 1.
- Τα επόμενα 4 bits περιλαμβάνουν το Account Type.
- Τα τελευταία 8 bits περιλαμβάνουν το Universe στο οποίο ανήκει ο λογαριασμός..

Υπάρχουν 4 Universe για τους λογαριασμούς.

Πίνακας 2 Τύπος σύμπαντος (Universe) λογαριασμών Steam

Universe ID	Type
0	Invalid
1	Public
2	Beta
3	Internal
4	Dev

Λαμβάνοντας υπόψη τα παραπάνω στοιχεία, ένα Steam ID μπορεί να μετατραπεί στην 64-bit μορφή του ως εξής:
((*Universe* << 56) | (*Account Type* << 52) | (*Instance* << 32) | *Account ID*)

Παράδειγμα που λειτουργεί:

- Universe: Public (1)
- Account Type: Clan (7)
- Instance: 0
- Account ID: 4
- 64-bit integer value: 103582791429521412

3.2 Διαδικασία Συλλογής

Αρχικά πραγματοποιήθηκε η συλλογή δεδομένων. Για την διαδικασία αυτή χρησιμοποιήσαμε το Steam Web Api (https://steamcommunity.com/dev) το οποίο είναι ένα εργαλείο που διαθέτει δωρεάν σε προγραμματιστές η Steam. Το Steam Web Api δέχεται σαν όρισμα το UserID, δηλαδή τον μοναδικό κωδικό που αντιστοιχεί σε κάθε χρήστη, οπότε παράξαμε τυχαία id's και ελέγχαμε αν αντιστοιχούν σε πραγματικό χρήστη. Στην συνέχεια, συλλέξαμε γενικές πληροφορίες για τους χρήστες όπως κατάσταση προφίλ (δημόσιο, ιδιωτικό, μόνο για φίλους), χώρα, ημερομηνία δημιουργίας, ημερομηνία τελευταίας αποσύνδεσης.

Πίνακας 3 Γενικές πληροφορίες χρηστών

	01.Uid	02.ComVisibility	03.Personastate	04.Country	05.State	06.City	07.Created	08.Logout
0	76561193671398447	3	1	nan	nan	nan	08-05-04 05:35	03-11-15 15:31
1	76561193671398449	3	0	nan	nan	nan	08-05-04 05:35	01-05-13 09:38
2	76561193671398453	3	0	nan	nan	nan	08-05-04 05:36	03-11-15 18:15
3	76561193671398458	3	3	AU	TAS	4978	09-05-04 09:31	01-11-15 09:04
4	76561193671398470	3	0	FJ	3	nan	09-05-04 09:32	02-11-15 03:09
5	76561193671398471	3	1	CA	QC	4657	08-05-04 05:37	03-11-15 08:31
6	76561193671398479	3	0	nan	nan	nan	08-05-04 05:38	20-05-08 19:41
7	76561193671398482	1	0	nan	nan	nan	nan	03-11-15 00:44
8	76561193671398486	3	0	nan	nan	nan	09-05-04 09:33	29-08-06 22:50
9	76561193671398494	3	3	US	CA	189	09-05-04 09:34	03-11-15 13:42

Όπως φαίνεται στον πίνακα 3, η συλλογή περιέχει τον κωδικό χρήστη(Uid), κωδικοποιήση σχετικά με είδος ασφάλειας προφίλ(δημόσιο ή ιδιωτικό)(ComVisibility), την χώρα, την πόλη, την ημερομηνία που δημιουργήθηκε το προφίλ(Created) και τέλος την τελευταία αποσύνδεση του.

Σε όλους μας τους πίνακες παρατηρήθηκε να μην μπορούμε να συλλέξουμε κάποια συγκεκριμένα δεδομένα λόγω απόκρυψης απο τον χρήστη, ή λόγω ελλειπής ενημέρωσης ή επειδή ήταν μηδενικά, αυτά τα δεδομένα τα απεικονίσαμε με το "nan" έτσι ώστε κατά την κωδικοποίηση να μπορούν να αναγνωρίζονται από το πρόγραμμα.

Μέσα από αυτούς τους χρήστες επιλέξαμε μόνο αυτούς που έχουν το προφίλ τους δημόσιο και που έχουν αποσυνδεθεί τελευταία φορά μετά από 01/2010. Χρησιμοποιήσαμε λοιπόν αυτό το πρώτο δείγμα και μετά από αυτήν την διαλογή δημιουργήσαμε μια πρώτη βάση δεδομένων (σχήμα 6).

Το αρχικό δείγμα που χρησιμοποιήσαμε για αυτή την διαδικασία περιείχε 371.755 χρήστες, έπειτα αφού αποκλείσαμε αυτούς με ιδιωτικό προφίλ είχαμε 284.851 χρήστες από τους οποίους μπορούσαμε να συλλέξουμε πληροφορίες για όλες τους τις δραστηριοτητές. Βέβαια παρατηρήσαμε ότι υπήρχαν

πολλοί λογαριασμοί ανενεργοί και σε συνδυασμό με το οτι το API ξεκίνησε να λειτουργει το 2009 αποκλείσαμε και όλους αυτούς που είχαν κάνει τελευταία αποσύνδεση πριν το 2010.

Σχήμα 6 Δημιουργία φίλτρου για δημόσια και ενεργά προφίλ

Έπειτα συλλέξαμε περισσότερες πληροφορίες που αφορούν χρήστη- παιχνίδι οι οποίες περιείχαν στοιχεία όπως τον κωδικό του χρήστη, τα παιχνίδια που διαθέτει και πόσο έχει παίξει το καθένα συνολικά αλλά και τις τελευταίες 2 εβδομάδες (πίνακας 4). Στο κάθε παιχνίδι αντιστοιχεί ένας κωδικός (appid). Στο παρακάτω πίνακα έχουμε απεικονίσει ένα ευανάγνωστο παράδειγμα για το πως λαμβάνουμε τα δεδομένα. Στην πρώτη στήλη φαίνεται επαναλαμβανόμενα ο κωδικός χρήστη(uid) και στην συνέχεια φαίνονται κωδικοί παιχνιδιών(appid), από εδώ λοιπόν φαίνεται πως ο συγκεκριμένος χρήστης έχει στην κατοχή του όλα αυτά τα παιχνίδια και στις αμέσως επόμενες στήλες φαίνεται πόσο χρόνο έχει δαπανήσει τις τελευταίες 2 εβδομάδες και συνολικά, αντίστοιχα.

Πίνακας 4 Πίνακας Χρήστη-Παιχνιδιών

UserID	No	appid	playtime_2weeks	playtime_forever
76561193665898439	0	10	nan	0
76561193665898439	1	20	nan	0
76561193665898439	2	30	nan	0
76561193665898439	3	40	nan	0
76561193665898439	4	50	nan	0
76561193665898439	5	60	nan	0
76561193665898439	6	70	nan	0
76561193665898439	7	130	nan	0
76561193665898439	8	220	nan	0
76561193665898439	9	240	nan	17091
76561193665898439	10	320	nan	0
76561193665898439	11	340	nan	0
76561193665898439	12	2100	nan	761
76561193665898439	13	2130	nan	0
76561193665898439	14	550	nan	152

Τέλος δημιουργήσαμε ένα πίνακα παιχνιδιών του οποίου συλλέξαμε πληροφορίες από την ιστοσελίδα του store της Steam καθώς το Steam Web Api δεν διέθετε τέτοιες πληροφορίες. Ο πίνακας αυτός περιείχε τον κωδικό του παιχνιδιού, την τιμή, τις κατηγορίες στις οποίες ανήκει, ημερομηνία έναρξης και ένα συνολικό βαθμό από το metacritic.

Πίνακας 5 Πίνακας παιχνιδιών

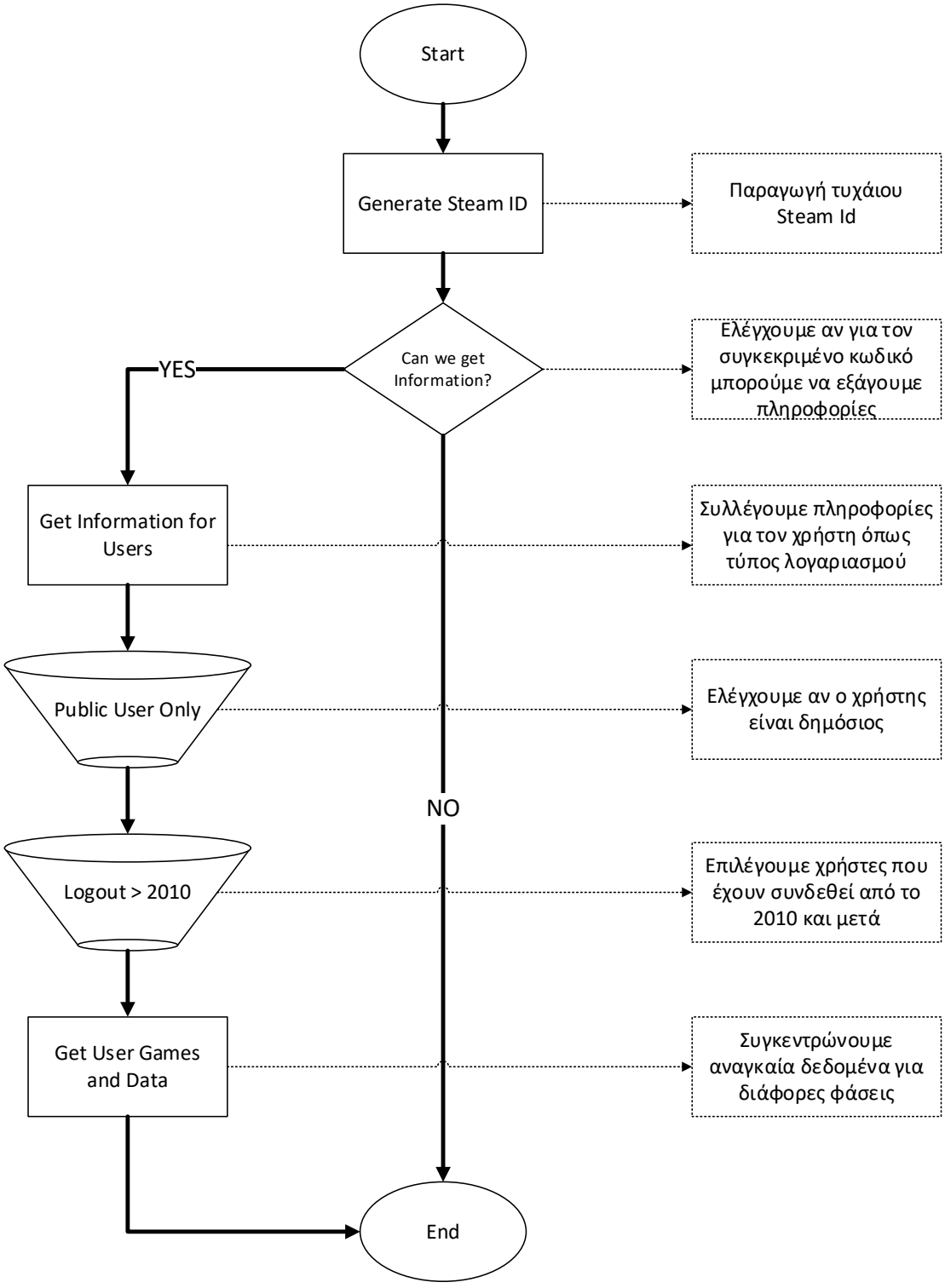
appid	currency	final_price	genres	initial_price	is_free	metacritic	release_date
10.0	EUR	799.0	[Action]	799	0	88	01-Nov-00
30.0	EUR	399.0	[Action]	399	0	79	01-May-03
130.0	EUR	399.0	[Action]	399	0	71	01-Jun-01
220.0	EUR	174.0	[Action]	699	0	96	16-Nov-04
240.0	EUR	1999.0	[Action]	1999	0	88	01-Nov-04
2800.0	EUR	499.0	[Strategy]	499	0	72	21-Jul-06
2850.0	EUR	449.0	[Simulation, Strategy]	449	0	nan	08-Oct-10
17410.0	EUR	999.0	[Action, Adventure]	999	0	81	14-Jan-09
97000.0	EUR	699.0	[Indie, Casual]	699	0	72	17-Jun-11
271640.0	EUR	899.0	[Action, Indie]	899	0	nan	20-Feb-14

Για να πραγματοποιήσουμε αναλύσεις οι οποίες αφορούν στατιστικά στοιχεία του δείγματος και της καταναλωτικής συμπεριφοράς, δημιουργήσαμε ένα φίλτρο έτσι ώστε να δημιουργηθεί μια δοκιμαστική βάση. Κρατήσαμε μόνο τους παίχτες που έχουν παίξει τουλάχιστον ένα παιχνίδι τις

τελευταίες 2 εβδομάδες και στην συνέχεια επιλέξαμε τυχαία 50.000 χρήστες (Εικόνα 1). Με αυτή την βάση που δημιουργήσαμε, εξετάσαμε χρονικές μεταβολές και αλλαγές στην συμπεριφορά των χρηστών.

Εικόνα 1 Δημιουργία φίλτρου δοκιμαστικής βάσης

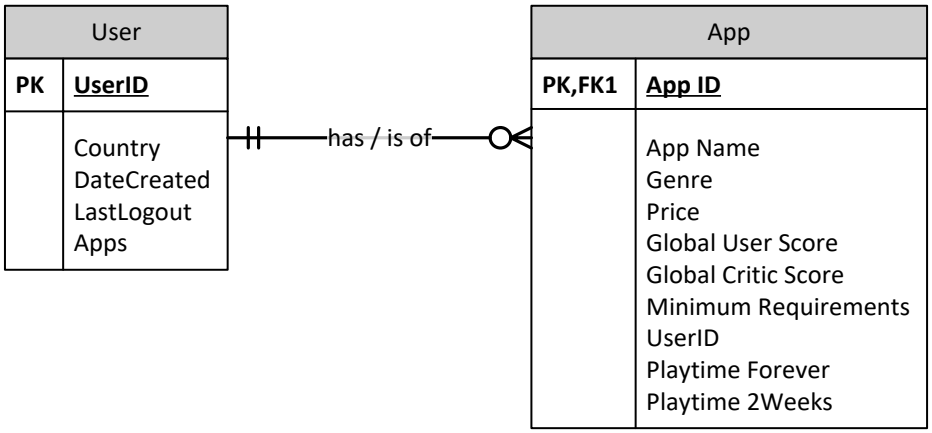
Στο επόμενο διάγραμμα (Σχήμα 7) παρατίθεται συνοπτικά η διαδικασία σε μορφή διαγράμματος.



Σχήμα 7 Διαδικασία συλλογής δεδομένων

3.2.1 Παραδοχές

- Το Steam Web Api το οποίο χρησιμοποιήσαμε για να εξάγουμε δεδομένα ξεκίνησε την καταγραφή μετά το 2009, οπότε δεν υπάρχουν πληροφορίες πριν από τότε.
- Συλλέξαμε δεδομένα για χρήστες που έχουν το προφίλ τους δημόσιο (Public) και έχουν πραγματοποιήσει την τελευταία τους είσοδο μετά το 2010.
- Οι πληροφορίες για τους χρήστες αφορούν την δεδομένη στιγμή που τις συλλέξαμε, οπότε πιθανές αλλαγές δεν διατυπώνονται.
- Η τιμή των παιχνιδιών είναι δυναμική μεταβλητή και υπόκειται σε αλλαγές, π.χ προσφορές, πακέτα. Εμείς χρησιμοποιήσαμε την τιμή που έδινε το store την συγκεκριμένη στιγμή.
- Η χώρα προέλευσης του παίχτη είναι διψήφιος κωδικός , π.χ GR, US
- Το είδος του παιχνιδιού (Genre) το έχουμε πάρει από την επίσημη σελίδα του παιχνιδιού στο steam και κάθε παιχνίδι μπορεί να ανήκει σε μία και παραπάνω κατηγορίες.



Σχήμα 8 Σχηματική αναπαράσταση από Microsoft Visio της βάσης δεδομένων μας

3.3 Δεδομένα

Παρακάτω παρατίθεται πίνακας με τις μεταβλητές που έχουμε στην διάθεση μας και μπορούμε να χρησιμοποιήσουμε.

Πίνακας 6 Μεταβλητές όπως εξήχθησαν από την συλλογή

Variable	Type	Measurement Unit	Example
User ID	Number	-	76561193665498400
Country	Text Code	-	US
Date Created	Time	YYYY-MM-DD HH:MM:SS	2003-09-12 18:07:20
Last Logout	Time	YYYY-MM-DD HH:MM:SS	2015-07-08 15:21:09
App Name	Text	-	Dota 2
App ID	Number	-	570
Playtime Forever	Number	Minutes	2808
Playtime 2Weeks	Number	Minutes	60
Genre	Text	-	Action, Strategy
Minimum PC Requirements	Text	-	OS: Windows 7 , Processor: Dual core from Intel or AMD at 2.8 GHz , Memory: 2gb RAM, DirectX: Version 9.0c
Price	Number	$	Free
Global User Score	Number 1 decimal	Scale of 0-10	6.4
Global Critic Score	Number 0 decimal	Scale of 0-100	90

3.3.1 Ορισμοί

Για καλύτερη σαφήνεια, οι μεταβλητές του παραπάνω πίνακα επεξηγούνται παρακάτω:

- User ID : Μοναδικός κωδικός χρήστη
- Country : Κωδικός χώρας χρήστη, 2-character ISO
- Date Created : Η ημερομηνία που δημιουργήθηκε ο λογαριασμός
- Last Logout : Η τελευταία φορά που συνδέθηκε ο χρήστης στην πλατφόρμα
- App name : Λίστα με τα ονόματα των παιχνιδιών που έχει ο χρήστης
- App id : Λίστα με τους κωδικούς παιχνιδιών που έχει ο χρήστης
- Playtime Forever : Ο συνολικός χρόνος που έχει δαπανήσει ο χρήστης στο παιχνίδι, σε λεπτά.
- Playtime 2Weeks : Ο συνολικός χρόνος σε λεπτά για τις τελευταίες 2 εβδομάδες
- Genre : Κατηγορίες παιχνιδιών
- Minimum Computer Requirements : Μια λίστα με τα απαραίτητα χαρακτηριστικά υπολογιστή για να λειτουργήσει το παιχνίδι σωστά.
- Price : Η τιμή του παιχνιδιού σε δολλάρια
- Global Critic Score : Συγκεντρώνουν δείγματα από έμπιστους κριτικούς και εφαρμόζουν σταθμισμένο μέσο βασιζόμενοι στην ποιότητα και την αναγνωρισιμότητα των κριτικών τους και το αποτέλεσμα είναι ένας αριθμός από το 0-100

Σημείωση1: Κάποια παιχνίδια, λιγότερο δημοφιλή δεν έχουν αρκετές κριτικές με αποτέλεσμα να μην έχουμε το global score, δηλαδή την κατάταξη τους.

Σημείωση2: Στην κατηγορία playtime 2Weeks υπάρχουν πολλές εγγραφές με ‘nan’ το οποίο υποδηλώνει πως το παιχνίδι δεν έχει παιχτεί τις τελευταίες 2 εβδομάδες.

Η συλλογή γινόταν ανά χρήστη και ο πίνακας που δημιουργήθηκε περιείχε όλα τα δεδομένα που αφορούν τους χρήστες, δηλαδή όλα τα παιχνίδια καθώς και όλες τις τιμές σε κάθε χαρακτηριστικό που αναφέρεται πιο πάνω. Οπότε δημιουργήθηκε ένας πίνακας με όλους τους χρήστες αλλά και όλες τις εναλλακτικές(παιχνίδια) που κατέχει ο καθένας, με αποτέλεσμα να δημιουργηθεί ένας πίνακας με 13 εκατομμύρια καταχωρίσεις.

Κεφάλαιο 4: Ανάλυση δεδομένων

Το αποτέλεσμα της συλλογής δεδομένων ήταν πολλές ακατέργαστες πληροφορίες. Είχαμε πίνακες με 13 εκατομμύρια γραμμές και βάσεις τις οποίες αδυνατούσαν να ανοίξουν γνωστά προγράμματα όπως Microsoft Excel καθώς ξεπερνούσαν τον όγκο επιτρεπτών κελιών. Συγκεντρώσαμε και αποθηκεύσαμε σε αρχεία κειμένου(.csv) πληροφορίες για χρήστες και παιχνίδια. Λόγω του μεγέθους των πληροφοριών δεν μπορούσαμε να επεξεργαστούμε τα δεδομένα με τυπικές μεθόδους, όπως το excel, για παράδειγμα μια και ο πίνακας χρηστών παιχνιδιών του δείγματος περιείχε 13 εκατομμύρια εγγραφές. Οπότε με χρήση της προγραμματιστικής γλώσσας Python(https://www.python.org) και του στατιστικού πακέτου Pandas, πραγματοποιήσαμε αναλύσεις και μετατροπές στα δεδομένα μας.

Τα αποτελέσματα ενός πειράματος συνήθως δημιουργούν ένα μεγάλο αριθμό δεδομένων. Κρίνεται λοιπόν αναγκαία η εύρεση διαδικασιών, με τις οποίες τα αποτελέσματα αυτά μπορθούν να οργανωθούν και να παρουσιαστούν με απλό και εύληπτο τρόπο. Αυτός ακριβώς είναι και ο στόχος των περιγραφικών στατιστικών δεικτών, δηλαδή να παρέχουν μμεθόδους που απλοποιούν και διευκολύνουν την οργάνωση και παρουσίαση των αποτελεσμάτων.

Η περιγραφική στατιστική (descriptive statistics), περιλαμβάνει μεθόδους για την οργάνωση, απλοποίηση και συνοπτική παρουσίαση των δεδομένων. Αν και υπάρχουν πολλές τεχνικές που ανήκουν σε αυτήν την κατηγορία, η πιο διαδεδομένη είναι ο υπολογισμός της μέσης τιμής (mean) και της τυπικής απόκλισης (standard deviation). Ωστόσο στους πίνακες παρακάτω φαίνεται ακόμα η ελάχιστη τιμή, η μέγιστη καθώς και ο διαχωρισμός του δείγματος σε 4 ίσα ποσοστημόρια (τεταρτημόρια), 0%-25% , 25%-50% , 50%-75%, 75%-100%.

Τα ποσοστημόρια αποτελούν γενίκευση της έννοιας της διαμέσου και βοηθούν στην πληρέστερη περιγραφή της θέσης της κατανομής παρατηρήσεων. Το ποσοστημόριο είναι η τιμή x, για την οποία ισχύει ότι: το α% των παρατηρήσεων είναι μικρότερες από αυτή και το υπόλοιπο (1-α)% των παρατηρήσεων είναι μεγαλύτερες από αυτή.

4.1 Στατιστικά Μέτρα

Πίνακας Χρήστη

Πίνακας 7Πίνακας χρήστη μετά από επεξεργασία

UserID	GamesCount	ForeverMean	Weeks2Mean	Weeks2Sum	PriceMean	PriceSum	GenCover
76561193665450065	53	2197.113208	13.830189	733	951.413043	43765	9
76561193665450076	189	1563.068783	22.671958	4285	1470.397661	251438	10
76561193665450079	85	1537.764706	10.658824	906	1327.211268	94232	11
76561193665450084	32	1686.718750	0.500000	16	1183.040000	29576	3
76561193665450089	86	502.058140	16.569767	1425	1531.506494	117926	11

Στο συγκεκριμένο πίνακα(Πίνακας 7) έχουμε συγκεντρώσει όλους τις χρήστες και έχουμε προσθέσει με την σειρά τα εξής:

- GamesCount: αριθμό των παιχνιδιών που κατέχουν
- ForeverMean: τον χρόνο που αφιερώνουν κατά μέσο όρο σε ένα παιχνίδι
- Weeks2Mean: τον χρόνο που αφιερώνουν κατά μέσο όρο σε ένα παιχνίδι τις τελευταίες 2 εβδομάδες
- Weeks2Sum: τον συνολικό χρόνο που δαπάνησαν παίζοντας τις τελευταίες 2 εβδομάδες
- PriceMean: την μέση τιμή των παιχνιδιών που κατέχουν
- PriceSum: το συνολικό ποσό χρημάτων που έχουν διαθέσει
- GenCover: τον αριθμό διαφορετικών κατηγοριών στις οποίες παίζουν

Για παράδειγμα, όπως φαίνεται στον πίνακα, ο χρήστης με κωδικό: 76561193665450065, κατέχει 53 τίτλους παιχνιδιών, παίζει κατά μέσο όρο 2197 λεπτά (36 ώρες) το κάθε παιχνίδι, τις τελευταίες 2 εβδομάδες έπαιξε κάθε παιχνίδι κατά μέσο όρο 13 λεπτά και συνολικά έπαιξε 733 λεπτά. Το μέσο κόστος των παιχνιδιών του είναι 9,51 € και συνολικά έχει δαπανήσει 437,65 €. Τέλος, έχει παίξει 9 διαφορετικές κατηγορίες παιχνιδιών.

Πίνακας 8 Στατιστικά μέτρα για τον τελικό πίνακα χρήστη

#	Αριθμός Παιχνιδιών	Χρόνος παιξίματος	Μέσος Χρόνος παιξίματος 2 εβδομάδων	Συνολικός Χρόνος 2 εβδομάδων	Μέση Δαπάνη	Κάλυψη κατηγορίων	Δαπάνες
count	219,336.00	219,336.00	219,336.00	219,336.00	219,336.00	219,336.00	219,336.00
mean	50.96	1,341.35	12.82	469.84	865.96	5.62	52,574.07
std	105.76	2,784.19	99.63	3,148.32	360.57	4.52	112,293.03
min	1.00	0.00	0.00	0.00	0.00	1.00	0.00
25%	12.00	127.85	0.00	0.00	556.29	1.00	6,989.00
50%	18.00	519.49	0.00	0.00	759.13	5.00	12,987.00
75%	45.00	1,423.06	1.97	129.00	1,154.12	10.00	46,367.00
max	5,373.00	161,609.43	13,119.40	593,212.00	5,499.00	22.00	4,428,180.00

Στον πίνακα 8 φαίνονται τα χαρακτηριστικά του πίνακα και τα στατιστικά τους μέτρα όπως προέκυψαν για το σύνολο του δείγματος των χρηστών. Ενδιαφέρον προκαλεί ότι κάτα μέσον όρο οι χρήστες έχουν 50.96 παιχνίδια παίξει, επίσης κατά μέσον όρο ένας παίκτης δαπανά 865€ για την αγορά των παιχνιδιών του.

Πίνακας Παιχνιδιών

Πίνακας 9 Πίνακας παιχνιδιών μετά από επεξεργασία

appid	final_price	metacritic	sumgenres	DaysFromRelease	UsersCount	weeks2Mean	weeks2Sum
10	799	88	1	5522	219062	994.627022	4549424
20	399	NaN	1	6102	218627	613.397380	140468
30	399	79	1	4611	218624	574.483696	211410
40	399	NaN	1	5310	218617	1315.317073	53928
50	399	NaN	1	5888	218628	988.474227	95882

Σε αυτό τον πίνακα βρίσκονται όλα τα παιχνίδια που βρίσκονται στην πλατφόρμα του Steam. Εμείς συλλέξαμε πληροφορίες που αφορούν αποκλειστικά το παιχνίδι αλλά προσθέσαμε και άλλες πληροφορίες οι οποίες προήλθαν από το δείγμα μας, όπως χρόνος παιξίματος του παιχνιδιού από τους παίχτες του δείγματος(Πίνακας 9).

Αναλυτικά, τα χαρακτηριστικά που έχει αυτός ο πίνακας είναι ο κωδικός του παιχνιδιού(appid), η τελική τιμή(final price), η βαθμολογία του στο metacritic(metacritic), ο αριθμός κατηγοριών στον οποίο

ανήκει(sumgenres), πόσες μέρες είναι στην αγορά(daysfromrelease), πόσοι χρήστες από το δείγμα μας το παίζουν(UsersCount), πόσο παίχτηκε κατά μέσο όρο τις 2 τελευταίες εβδομάδες(weeks2Mean) και πόσο παίχτηκε συνολικά τις 2 τελευταίες εβδομάδες(weeks2Sum).

Για παράδειγμα το παιχνίδι με κωδικό 10, ονομάζεται Counter-Strike. Η τιμή του είναι στα 7,99€, έχει βαθμολογία 88 στα 100, ανήκει μόνο στην κατηγορία Action και η ηλικία του είναι 5.522 ημέρες. Από τους χρήστες στο δείγμα μας το είχαν οι 219.062, ο μέσος όρος που παίχτηκε της τελευταίες 2 εβδομάδες ήταν 994 λεπτά και στο σύνολο 4.549.424 λεπτά.

Πίνακας 10 Στατιστικά μέτρα για τον τελικό πίνακα παιχνιδιών

#	Κάλυψη κατηγοριών	Τιμή	Βαθμολογία	Χρόνος 2 εβδομάδων	Ηλικία	Χρήστες	Μέσος χρόνος 2 εβδομάδων
count	6,703.00	6,608.00	1,870.00	4,325.00	6,644.00	6,701.00	4,325.00
mean	2.32	1,051.53	72.07	23,461.86	784.70	1,523.36	232.39
std	1.29	1,044.73	11.00	495,949.60	836.64	8,981.13	774.30
min	0.00	0.00	20.00	1.00	-366.00	1.00	1.00
25%	1.00	499.00	66.00	108.00	210.00	30.00	38.00
50%	2.00	899.00	73.00	509.00	473.50	152.00	109.00
75%	3.00	1,399.00	80.00	2,613.00	1,008.25	726.00	198.05
max	11.00	19,900.00	96.00	31,326,988.00	6,742.00	219,062.00	17,751.00

Ενδιαφέρον αποτελεί ότι ένα παιχνίδι κατά μέσον όρο δεν ανήκει σε μόνο μια κατηγορία. Δηλαδή τα παιχνίδια δεν χαρακτηρίζονται από ένα γένος π.χ. περιπέτειας αλλά τείνουν να έχουν 2 ή περισσότερα γένη όπως για παράδειγμα περιπέτειας, οικογενειακό, δράσης. Αυτό δυσκολεύει την κατηγοριοποίηση τους καθώς οι συνδυασμοί δημιουργούν αρκετή πολυπλοκότητα στη σύγκριση τους.

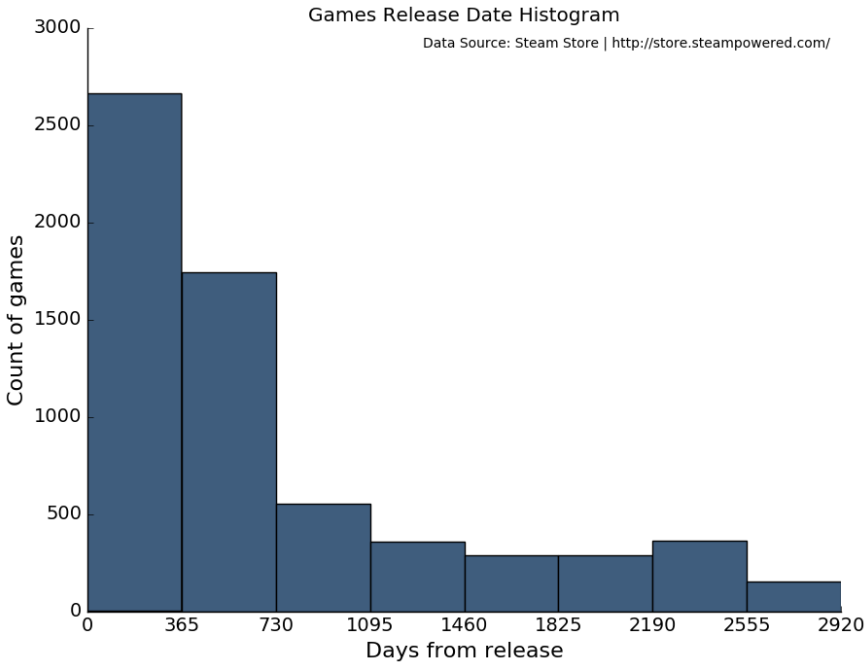
4.2 Γραφική απεικόνιση

Τα δεδομένα ενός πίνακα συχνοτήτων μπορούν να παρασταθούν γραφικά με ένα ραβδόγραμμα, όπου κάθε ράβδος παρουσιάζει τη συχνότητα (ή αθροιστική συχνότητα ή σχετική συχνότητα) για κάθε τιμή της μεταβλητής. Σε αυτή την υποενότητα επιλέχθηκαν κάποια ραβδογράμματα τα οποία παρατηρήσαμε ότι ακολουθούσαν ενδιαφέρουσες μεταβολές. Τα στοιχεία προήλθαν από την βάση δεδομένων μας και αφορούν τόσο τους παίχτες όσο και τα παιχνίδια. Οι πληροφορίες εξήχθησαν είτε από το Steam Web Api είτε από την σελίδα store.steampowered.com με crawlers που κατασκευάσαμε. Για την ανάγκη αυτή χρειάστηκε να δημιουργήσουμε κατάλληλους επαναλαμβανόμενους αλγορίθμους οι οποίοι με την μέθοδο της δοκιμής και επανεξέτασης συλλέγανε πληροφορίες. Αυτό σημαίνει ότι έπρεπε για κάθε καταχώριση που θέλαμε να εξετάσουμε είχαμε κατασκευάσει έναν κώδικα ο οποίος χτύπαγε μια ιστοσελίδα και έπαιρνε τις πληροφορίες από τις μεταβλητές που εμείς του ζητήσαμε. Αυτή η διαδικασία πέρα από χρονοβώρα ήταν και αρκετά δοκιμαστική καθώς έπρεπε να κατανοήσουμε το τρόπο που είχε δομηθεί το κάθε σύστημα έτσι ώστε να μπορούμε να τραβήξουμε τις πληροφορίες που θέλαμε.

Αφού έγινε η συλλογή και η διαλογή τους, έπρεπε να επεξεργαστούμε αυτό τον μεγάλο όγκο που προαναφέρθηκε σε προηγούμενη ενότητα με καινοτόμους τρόπους μιας και τα δεδομένα ξεπερνούσαν την διαθέσιμη υπολογιστική που διαθέταμε. Σε αυτό βοήθησε η δημιουργία έξυπνων αλγορίθμων, κώδικα, στους οποίους καταφέραμε να επεξεργαζόμαστε κόμματια του πίνακα(chunks) και όχι όλο τον πίνκα μαζί με αποτέλεσμα να χρειαζόμαστε λιγότερη ισχύ αλλά πολλές επαναλήψεις.

Εδώ χρησιμοποιήσαμε στατιστικά εργαλεία και βιβλιοθήκες όπως pandas (https://pandas.pydata.org/), matplotlib (https://matplotlib.org/), numpy (www.numpy.org/) και scipy (https://www.scipy.org/)

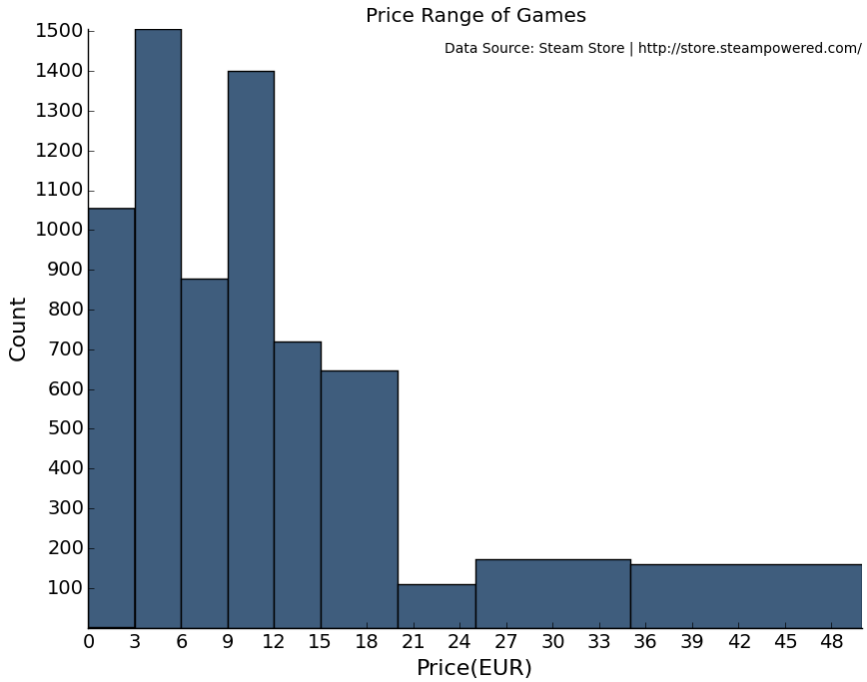
Ηλικία παιχνιδιών



Διάγραμμα 4 Ραβδόγραμμα ηλικίας παιχνιδιών

Το παραπάνω διάγραμμα αφορά την προσέλευση παιχνιδιών στην αγορά τα τελευταία χρόνια. Παρατηρούμε ότι κάθε χρόνο προστίθενται περισσότεροι τίτλοι, το οποίο έμμεσα αποδεικνύει την μεγάλη ζήτηση της αγοράς. Συγκεκριμένα μόνο τον τελευταίο χρόνο προστέθηκαν πάνω από 2500 νέα παιχνίδια.

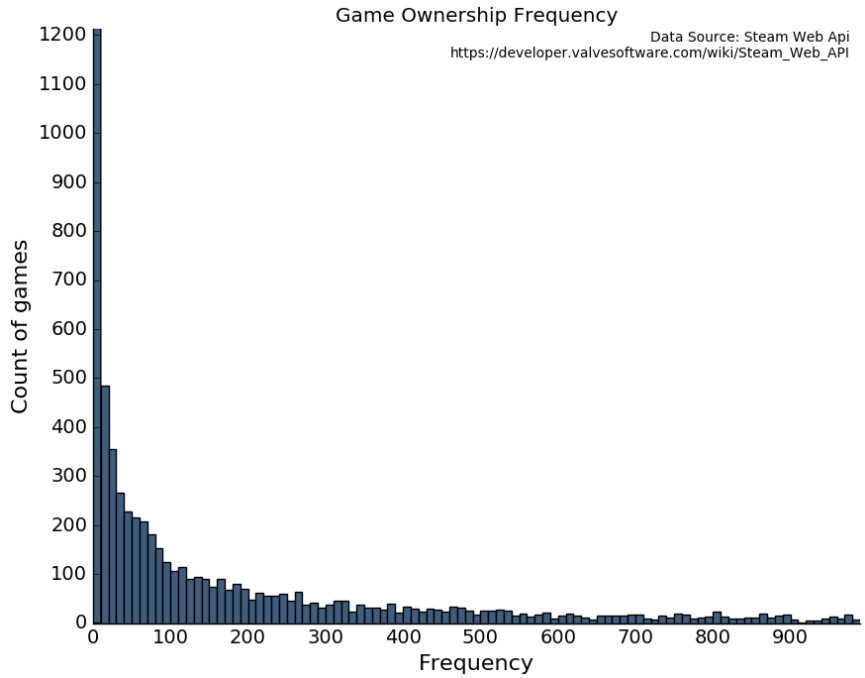
Τιμές παιχνιδιών



Διάγραμμα 5 Ραβδόγραμμα τιμών παιχνιδιών

Στο παραπάνω διάγραμμα έχει γίνει ομαδοποίηση των παιχνιδιών ανάλογα με την τιμή τους και αφορά μόνο παιχνίδια που είναι επί πληρωμή. Όσον αφορά τα δωρεάν παιχνίδια καταλαμβάνουν μόλις το 7% του συνόλου. Επίσης όπως φαίνεται και στο διάγραμμα παρατηρείται μεγάλος αριθμός παιχνιδιών με τιμές στα πεδία 3-6 και 9-12 και πιο συγκεκριμένα παρατηρήθηκαν στους πίνακες τα ποσά κοντά στα 5 και 10 ευρώ.

Κατοχή παιχνιδιών



Διάγραμμα 6Ραβδόγραμμα κατοχής παιχνιδιών

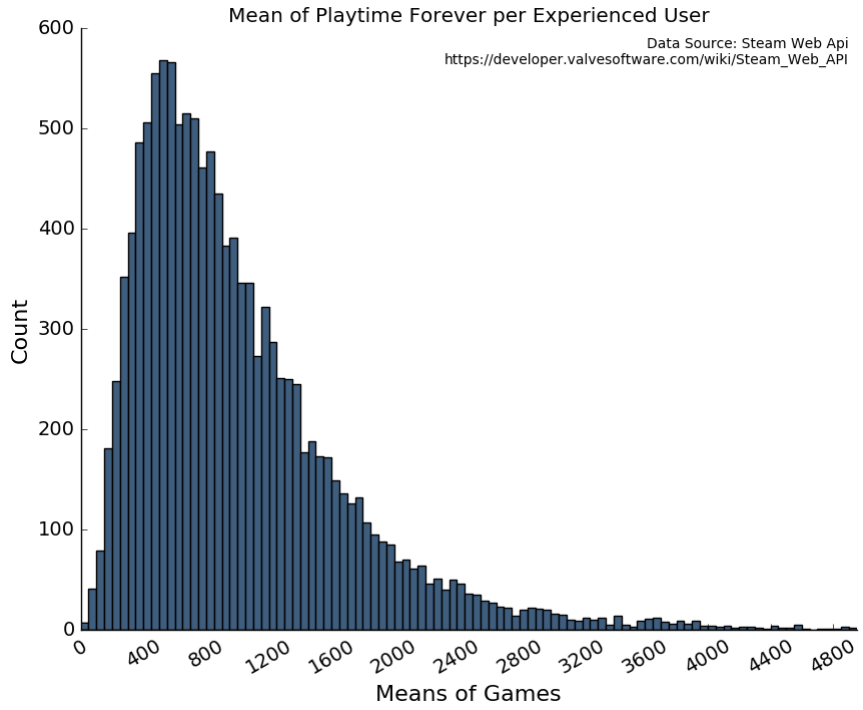
Αυτό το διάγραμμα παρουσιάζει την συχνότητα παιχνιδιών που εμφανίζεται ανά χρήστη. Δηλαδή πόσα παιχνίδια έχει ο χρήστης κατά μέσο όρο. Στον X άξονα βρίσκεται η συχνότητα εμφάνισης σε απόλυτο αριθμό και στον Y άξονα τον αριθμό των παιχνιδιών που ανήκουν στην συγκεκριμένη κατηγορία συχνότητας. Κάθε μπάρα στο ιστόγραμμα καλύπτει εύρος 10 μονάδων.

Με χρήση της εντολής mquantiles από την βιβλιοθήκη scipy, εφαρμόσαμε ομαδοποίηση με 3 συστάδες με στόχο να εμφανίζεται ίσος αριθμός παιχνιδιών σε κάθε μια. Το αποτέλεσμα ήταν τα πεδία :

[0 - 25], (25 – 140], (140 - 689]

Παρότι φαίνεται καθαρά στο διάγραμμα, συνυπολογίζοντας και αυτή την πληροφορία καταλαβαίνουμε ότι τα περισσότερα παιχνίδια δεν είναι διαδεδομένα και οι μεγάλοι τίτλοι είναι ελάχιστοι. Οπότε υπάρχει μεγάλο περιθώριο για εφαρμογή μιας μεθοδολογίας συστάσεων.

Μέσος όρος παιξίματος



Διάγραμμα 7 Ραβδόγραμμα μέσου χρόνου παιξίματος

Το παραπάνω διάγραμμα αφορά μόνο έμπειρους παίχτες με την έννοια ότι έχουν στη κατοχή τους τουλάχιστον 12 παιχνίδια. Στον άξονα Χ εμφανίζεται συχνότητα εμφάνισης χρηστών με κοινό κριτήριο τον μέσο χρόνο που αφιερώνει ο χρήστης σε ένα παιχνίδι και στον άξονα Υ το άθροισμα αυτών των χρηστών.

Στην συσταδοποίηση με 3 συστάδες που έγινε προέκυψε το ακόλουθο αποτέλεσμα:

[526.9 836.8 1298.7]

Σε συνεργασία και με το διάγραμμα φαίνεται πως οι τιμές βρίσκονται κυρίως μεταξύ 400 και 1600 δηλαδή ο χρήστης δαπανά από 6.5 μέχρι 26 ώρες ανά παιχνίδι.

Κεφάλαιο 5: Εξόρυξη Δεδομένων

5.1 Παρουσίαση Weka

Το WEKA(Waikato Environment for Knowledge Analysis)(https://www.cs.waikato.ac.nz/ml/weka/) είναι ένα περιβάλλον ανάπτυξης εφαρμογών μηχανικής μάθησης και εξόρυξης γνώσης, το οποίο αναπτύχθηκε από ερευνητές στο πανεπιστήμιο του Waikato στην Νέα Ζηλανδία. Επίσης είναι γραμμένο σε Java, τρέχει σχεδόν σε κάθε πλατφόρμα και έχει ελεγχθεί στα λειτουργικά συστήματα των Windows, Linux και Macintosh. Χρησιμοποιείται για έρευνα, εκπαίδευση και άλλες εφαρμογές και διαθέτει μια συλλογή από αλγορίθμους machine learning που μπορούν να κληθούν αμέσως από το περιβάλλον διεπαφής (GUI) ή από το δικό μας κώδικα Java. Το WEKA παρέχει εκτενή υποστήριξη για ολόκληρη την διαδικασία data mining συμπεριλαμβανομένης την προετοιμασία των δεδομένων εισόδου, την εκτίμηση στατιστικών σχημάτων εκμάθησης και την εικονική απεικόνιση των δεδομένων εισόδου καθώς και το αποτέλεσμα της εκάστοτε εκμάθησης. Ως πρόγραμμα το οποίο διαθέτει μια μεγάλη ποικιλία αλγορίθμων εκμάθησης περιλαμβάνει και ένα ευρύ φάσμα από εργαλεία προεπεξεργασίας δεδομένων. Σε αυτό το περιεκτικό εργαλείο υπάρχει πρόσβαση μέσω μιας κοινής διεπαφής έτσι ώστε οι χρήστες του να μπορούν να συγκρίνουν διαφορετικές μεθόδους και να προσδιορίσουν ποια από αυτές είναι η πιο κατάλληλη για το εκάστοτε πρόβλημα.

Η μέθοδος αποθήκευσης στοιχείων στο πρόγραμμα WEKA γίνεται συνήθως με ένα αρχείο .arff το οποίο περιέχει όλα τα δεδομένα τα οποία πρέπει να χρησιμοποιηθούν και γράφονται και διαβάζονται μέσω ενός απλού επεξεργαστή κειμένου. Ένα αρχείο .arff αποτελείται από την λίστα των περιπτώσεων (instances) που διαθέτουμε, και οι τιμές των χαρακτηριστικών για κάθε περίπτωση διαχωρίζονται από διαχωριστικό το οποίο δηλώνεται (π.χ κόμματα ή εσοχές).

5.1 Clustering

Αφού αναλύσαμε τα δεδομένα μας και έχουμε αποκτήσει νέες πληροφορίες , είναι τώρα η στιγμή που πρέπει να συνθέσουμε και να δημιουργήσουμε τα κριτήρια μας. Πριν φτάσουμε όμως σε αυτό το βήμα, χρησιμοποιούμε μεθόδους εξόρυξης δεδομένων έτσι ώστε να καταλάβουμε ποια στοιχεία είναι πιο σημαντικά για τους παίκτες (Διάγραμμα 2). Με την μέθοδο της συσταδοποίησης (clustering) εντοπίζουμε κοινές ομάδες-συστάδες στο δείγμα μας οι οποίες τείνουν να έχουν κάποια κοινά χαρακτηριστικά.

Στην βιβλιογραφία συναντάει κανείς πολλούς τρόπους για την εκτίμηση του αποτελέσματος της ομαδοποίησης ενός αλγορίθμου, που απαιτεί να είναι προκαθορισμένος ο αριθμός των clusters (k). Έχουν γίνει αξιοσημείωτες προσπάθειες προς αυτή την κατεύθυνση, ώστε να ξεπεραστεί σε μεγάλο βαθμό το πρόβλημα για το ποιο είναι το καλύτερο k. Μια πολύ γνωστή τεχνική είναι να συγκριθούν τα αποτελέσματα του διαμεριστικού αλγορίθμου ομαδοποίησης με τα αντίστοιχα ενός ιεραρχικού αλγορίθμου και μέσω του δενδρογράμματος να προκύψει το καλύτερο k. Ακόμη, γνωστή τεχνική αποτελεί η χρήση δεικτών εκτίμησης που αναλόγως την τιμή λαμβάνουν μπορεί να προκύψει συμπέρασμα για το k. Άλλος τρόπος που προτείνεται από μερικούς είναι να γίνει απεικόνιση των clusters και με την εφαρμογή διάφορων εργαλείων να βρεθεί το κατάλληλο k. Τέλος διάφορες προσπάθειες που έγιναν για την εκτίμηση του k, εμπλέκουν την έννοια της ευστάθειας. Αξίζει δε να αναφερθεί, ότι χρήσιμα συμπεράσματα για το k έχουν προκύψει και από προσπάθειες βελτίωσης διάφορων προβλημάτων που εμφανίζουν κάποιοι αλγόριθμοι. Ωστόσο όλες οι παραπάνω μέθοδοι δεν κατάφεραν να επιλύσουν το πρόβλημα, δηλαδή να μην προκαθορίζεται το k. Ο λόγος είναι ότι οι παραπάνω τεχνικές δεν μπορούν να εφαρμοστούν παντού και πάντα γιατί εμφανίζουν αδυναμίες σε κάποιες περιπτώσεις. Έτσι για παράδειγμα υπάρχουν δείκτες που έχουν μεγάλο υπολογιστικό κόστος κάτι που είναι καλό να αποφεύγεται κυρίως σε μεγάλα σύνολα δεδομένων. Άλλοι δείκτες πάλι δεν είναι κατάλληλοι όταν τα clusters έχουν κατά πολύ διαφορετικό αριθμό μελών. Και γενικά προβλήματα εμφανίζουν και οι άλλες τεχνικές γιατί οι περισσότερες προέκυψαν για να λύσουν το πρόβλημα κάτω από κάποιες συνθήκες και έγιναν κάποιες παραδοχές. Έτσι η έρευνα για την προσπάθεια εύρεσης του καλύτερου k με τρόπο που να γίνεται ευρέως αποδεκτό συνεχίζεται.

Με χρήση του προγράμματος Weka, επιλέξαμε Cluster-> Choose-> Simple K-means .Οπότε εφαρμόζοντας τον αλγόριθμο k-means κάναμε προσπάθειες με 3, 4 και 5 συστάδες αντίστοιχα και παρατηρήσαμε ότι τα αποτελέσματα με k=3 είχαμε την μεγαλύτερη διαφάνεια και ευστάθεια. Παρακάτω παρατίθενται τα αποτελέσματα:

Πίνακας χρήστη

Πίνακας 11 Αποτελέσματα συσταδοποίησης πίνακα χρήστη

```
Final cluster centroids:
                            Cluster#
Attribute       Full Data           0           1           2
                 (219336)    (107190)     (51399)     (60747)
==========================================================
GamesCount         50.9611     12.5732     32.6514    134.1899
ForeverMean      1341.355    1093.4321   1856.2652   1343.1491
2WeeksMean         12.8208      7.2656     20.0876     16.4746
2WeeksSum         469.8394     85.6814    468.3838     1148.93
PriceMean         865.9628    615.7102      983.88   1207.7699
PriceSum         52574.072   7826.1446  31769.2042 149136.5315
GenCover           5.6196      1.4919      7.1586     11.6009


Clustered Instances

0      107190 ( 49%)
1       51399 ( 23%)
2       60747 ( 28%)
```

Η πρώτη ομάδα που δημιουργήθηκε από την συσταδοποίηση καταλάμβανε το μεγαλύτερο ποσοστό (49%) και αφορούσε παίχτες οι οποίοι παίζανε πολύ στοχευμένες κατηγορίες (GenCover<2), δηλαδή δεν είχαν μεγάλη ποικιλία στις κατηγορίες που παίζανε. Ακόμα παρατηρήθηκε ότι είχανε τον μικρότερο αριθμό παιχνιδιών στην κατοχή τους αλλά επίσης παίζανε και κατά μέσο όρο σημαντικά λιγότερα λεπτά. Η δεύτερη ομάδα φαίνεται να έχουν σαν κύριο γνώρισμα την αφοσίωση στα παιχνίδια, με την έννοια ότι παίζουν κατά μέσο όρο περισσότερα λεπτά από τις 2 ομάδες, συνολικά αλλά και τις τελευταίες 2 εβδομάδες. Όσο για την τελευταία κατηγορία, φαίνεται ότι έχουν πάρα πολλά παιχνίδια στην κατοχή τους και εκτείνονται σε πολλές διαφορετικές κατηγορίες, με αποτέλεσμα να ξοδεύουν και αρκετά χρήματα σε σχέση με τους άλλους.

Πίνακας παιχνιδιών

Πίνακας 12 Αποτελέσματα συσταδοποίησης πίνακα παιχνιδιών

```
Final cluster centroids:
                                Cluster#
Attribute          Full Data          0          1          2
                      (6703)     (5127)      (337)     (1239)
=============================================================
final_price        1051.5374  1068.4748  1241.5448   929.7694
metacritic           72.0754    72.5211    54.5816    74.9892
DaysFromRelease     784.7095   411.7071   974.2692    2276.64
UsersCount         1523.3698   603.1714   978.2938  5479.4211
2weeksMean          232.3953   228.7767   231.5134   247.6091
2weeksSum         23461.8608  18588.303  9722.9528 47365.6079


Clustered Instances

0      5127 ( 76%)
1       337 (  5%)
2      1239 ( 18%)
```

Στα παιχνίδια, η συσταδοποίηση εμφάνισε μία κυρίαρχη ομάδα με 76% η οποία περιλαμβάνει παιχνίδια που έχουν κυκλοφορήσει τον τελευταίο ενάμιση χρόνο περίπου. Μεγάλες διαφορές δεν παρατηρήθηκαν πέρα από το γεγονός ότι η πρώτη κατηγορία είχε κατά μέσο όρο τους λιγότερους παίχτες και η Τρίτη κατηγορία με διαφορά τους περισσότερους (5.479) .

5.2 Classification

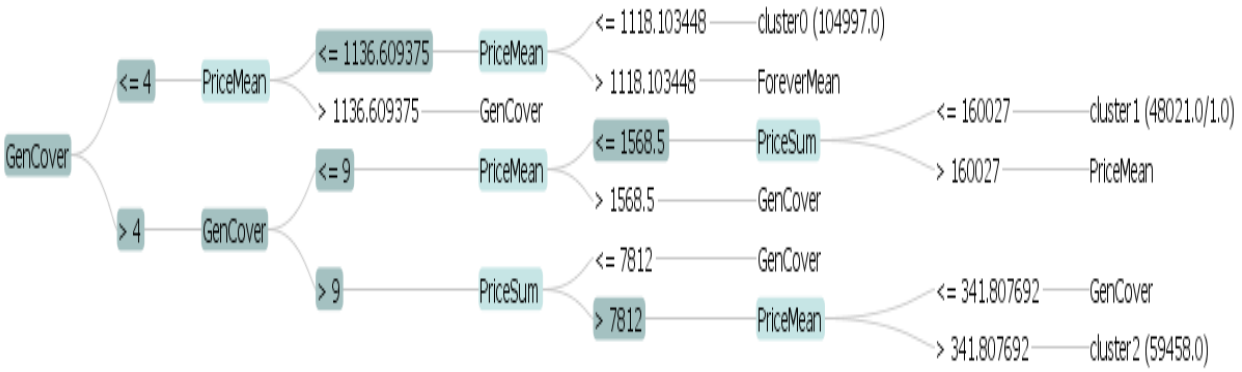
Τα δέντρα απόφασης αποτελούν μια από τις ποιο βασικές μεθόδους πρόβλεψης και ταξινόμησης και αποτελούν μια επαγωγική διαδικασία. Αναπαριστούν κανόνες και είναι αρκετά διαδεδομένα καθώς ο μηχανισμός της διαδικασίας απόφασης είναι αρκετά εμφανής και επιτρέπει αρκετά καλή ανάλυση της γνώσης την οποία λαμβάνουμε. Τα κύρια χαρακτηριστικά του είναι οι κόμβοι, τα κλαδιά και τα φύλλα. Σύμφωνα με έναν ορισμό, ένα δέντρο απόφασης είναι ένα δέντρο με τις ακόλουθες ιδιότητες:

- Κάθε εσωτερικός κόμβος ονοματίζεται με το όνομα ενός χαρακτηριστικού x
- Κάθε κλαδί / σύνδεση ονοματίζεται με ένα κατηγόρημα που μπορεί να εφαρμοστεί στο χαρακτηριστικό που αποτελεί το όνομα του κόμβου
- Κάθε φύλλο ονοματίζεται με το όνομα μιας κλάσης

Οι κόμβοι του δέντρου αφορούν τον έλεγχο ενός συγκεκριμένου χαρακτηριστικού. Τα κλαδιά περιέχουν συνθήκες σύγκρισης (κυρίως ανισότητες) της τιμής που λαμβάνει το συγκεκριμένο χαρακτηριστικό με μια άλλη τιμή η οποία και θα προσδιορίσει σε ποιο κόμβο ¨ παιδί¨ θα συνεχίσουμε την αναζήτηση ώστε να φτάσουμε στην κλάση την οποία και θέλουμε να προβλέψουμε και αναπαρίσταται στα φύλλα του δέντρου.

Για την ταξινόμηση προσθέσαμε στα δεδομένα που έιχαμε ήδη στους πίνακες μια νέα στήλη, όπου αναφερόταν η συστάδα στην οποία άνηκε η κάθε καταχώρηση. Επιλέξαμε τον αλγόριθμο j48 στο WEKA και με χρήση ενός πρόσθετου plug in (prefuse tree) για καλύτερη απεικόνιση δημιουργήθηκαν τα παρακάτω δέντρα για τις συστάδες που είχαμε βρει από την συσταδοποίηση.

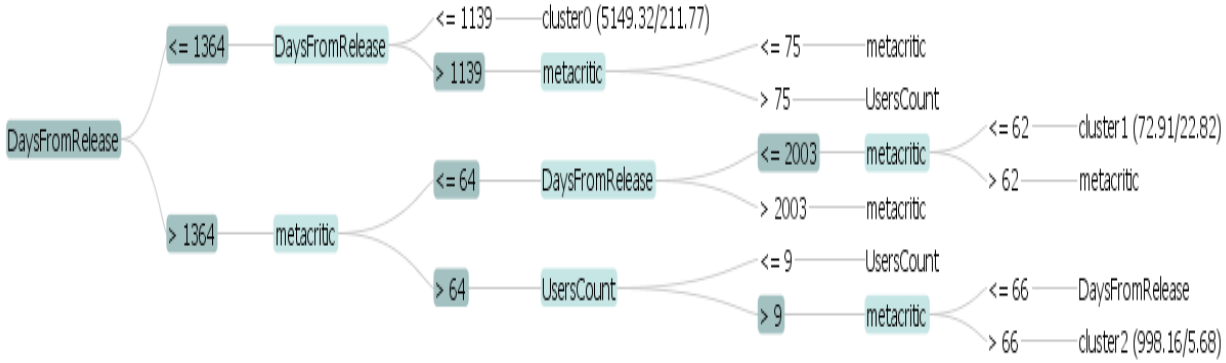
Πίνακας χρήστη



Σχήμα 9 Απεικόνιση με δένδρα αποφάσεων- Ταξινόμηση πίνακα χρήστη

Αφού πραγματοποιήσαμε την συσταδοποίηση στους πίνακες, προσθέσαμε μία στήλη στα δεδομένα όπου αφορούσε την συστάδα στην οποία ανήκαν και εφαρμόσαμε ταξινομήση. Το αποτέλεσμα μας έδειξε (Σχήμα 9)ότι η κάλυψη κατηγοριών είναι το χαρακτηριστικό με την μεγαλύτερη επιρροή.

Πίνακας παιχνιδιών



Σχήμα 10 Απεικόνιση με δένδρα αποφάσεων- Ταξινόμηση πίνακα χρήστη

Παρομοίως για τον πίνακα των παιχνιδιών, ο διαχωρισμός ήταν αρκετά ξεκάθαρος. Το πιο ισχυρό χαρακτηριστικό(Σχήμα 10) ήταν αυτό της ηλικίας και συνέχισε η βαθμολογία από το metacritic.

Κεφάλαιο 6: Πολυκριτήρια Ανάλυση

Τα συστήματα πολλαπλών πρακτόρων στην Τεχνητή Νοημοσύνη περιλαμβάνουν πολύ συχνά προβλήματα κατανομής εργασιών. Τα προβλήματα αυτά είναι δύσκολο να διαμορφωθούν και ως εκ τούτου, θεωρείται επίσης δύσκολη η εύρεση ενός βέλτιστου σχεδίου κατανομής. Το Agent Allocator είναι μία εύκολη στη χρήση, ανεξαρτήτως πλατφόρμας εφαρμογή, η οποία υλοποιεί μία πολυκριτήρια μέθοδο για την υποστήριξη της απόφασης της κατανομής εργασιών. Το άτομο που λαμβάνει την απόφαση είναι σε θέση να διαμορφώσει το πρόβλημα (σύμφωνα με την πολιτική του) μέσω των εισερχόμενων διαλόγων και να παρουσιάσει την τελική λύση που προτείνεται από το σύστημα.

Αντιμετωπίζοντας το πρόβλημα με παρόμοια οπτική, όπου οι πράκτορες είναι οι παίκτες και οι εργασίες είναι τα παιχνίδια. Προσπαθήσαμε να μοντελοποιήσουμε νέα κριτήρια και να εφαρμόσουμε την UTASTAR στον πολυκριτήριο που δημιουργείται.

6.1 Μοντελοποίηση Σύστασης

Δημιουργήσαμε τέσσερα κριτήρια τα οποία αποτελούταν από τα χαρακτηριστικά των πινάκων χρήστη και παιχνιδιού. Η ανάλυση του δείγματος και η εξόρυξη γνώσης στα προηγούμενα βήματα μας οδήγησε στην επιλογή των χαρακτηριστικών για την δημιουργία των κριτηριών. Αυτά τα κριτήρια θα είναι οι στήλες του πολυκριτήριου πίνακα για κάθε χρήστη.

Πίνακας 13 Μοντελοποίηση κριτηρίων

Ποικιλία	Εμπειρία	Δαπάνες	Χρήση
Αριθμός παιχνιδιών - Κάλυψη κατηγοριών	Μέσος χρόνος παιχνιδιών χρήστη- Ηλικία παιχνιδιού	μέσο κόστος παιχνιδιών- κόστος παιχνιδιού	Μέσος χρόνος παιχνιδιών χρήστη
Κάλυψη κατηγοριών χρήστη	Αριθμός παιχνιδιών χρήστη- Αριθμός παιχτών		Χρόνος παιξίματος 2 εβδομάδων χρήστη- Μέσος χρόνος παιξίματος 2 εβδομάδων

6.2 Απόδοση βαρών

Ο αποφασίζον καλείται να αποδώσει βάρη στα κριτήρια ανάλογα με τις προσωπικές του εκτιμήσεις. Στόχος του μοντέλου είναι να υποβοηθήσει τον αποφασίζον.

Οι πίνακες αυτοί (Lookup Tables) χρησιμοποιούνται για την δημιουργία του πρώτου πίνακα για τη σύνθεση του γραμμικού προβλήματος. Περιέχουν όλες τις εναλλακτικές(παιχνίδια) και τα βάρη που αντιστοιχούν στα κριτήρια με την προτίμηση του αποφασίζων. Συνδυάσαμε χαρακτηριστικά από τους πίνακες Χρήστη, Παιχνιδιών και Χρήστη-Παιχνιδιών βάση των τεχνικών συσταδοποίησης και κατηγοριοποίησης έτσι ώστε να παράγουμε τα κριτήρια Ποικιλία, Εμπειρία, Δαπάνες, Χρήση.

Παρακάτω φαίνεται ένα παράδειγμα με τιμές που προσθέσαμε στα ανάλογα πεδία.

Ποικιλία

<table>
<tr><td rowspan="6">Ποικιλία</td><td colspan="4">Αριθμός παιχνιδιών</td></tr>
<tr><td rowspan="4">Κάλυψη κατηγοριών παιχνιδιού</td><td></td><td>Χαμηλός (<=13)</td><td>Κανονικός (<=30)</td><td>Μεγάλος (>30)</td></tr>
<tr><td>Χαμηλή (<=1)</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Κανονική (<=2)</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Μεγάλη (>2)</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td colspan="4">Κάλυψη κατηγοριών χρήστη</td></tr>
<tr><td colspan="2">Μικρή (<=3)</td><td colspan="2">Ικανοποιητική (<=9)</td><td>Μεγάλη (>9)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">3</td><td>5</td></tr>
</table>

Πίνακας 14 Απόδοση βαρών από αποφασίζων στο κριτήριο Ποικιλία

Εμπειρία

Εμπειρία					
	Μέσος χρόνος παιχνιδιών χρήστη				
	Ηλικία παιχνιδιού		Χαμηλός *(<=180)*	Κανονικός *(<=720)*	Μεγάλος *(>720)*
		Νέο *(<=360)*	1	2	3
		Κανονικό (<=720)	2	3	4
		Παλιό *(>720)*	3	4	5
	Αριθμός παιχνιδιών χρήστη				
	Αριθμός παικτών παιχνιδιού		Χαμηλός *(<=13)*	Κανονικός *(<=30)*	Μεγάλος *(>30)*
		Χαμηλός *(<=50)*	1	2	3
		Κανονικός *(<=350)*	2	3	4
		Μεγάλος *(>350)*	3	4	5

Πίνακας 15 Απόδοση βαρών από αποφασίζων στο κριτήριο Εμπειρία

Δαπάνες

Δαπάνες					
	Μέσο κόστος παιχνιδιών				
	Κόστος παιχνιδιού		Χαμηλό *(<=4,99)*	Κανονικό *(<=9,99)*	Μεγάλο *(>9,99)*
		Χαμηλό *(<=4,99)*	5	3	1
		Κανονικό *(<=9,99)*	3	5	3
		Μεγάλο *(>9,99)*	0	3	5

Πίνακας 16Απόδοση βαρών από αποφασίζων στο κριτήριο Δαπάνες

Χρήση

Χρήση					
	Μέσος χρόνος παιχνιδιών χρήστη				
	Χαμηλός *(<=180)*	Κανονικός *(<=720)*			Μεγάλος *(>720)*
	1	3			5
	Χρόνος παιξίματος 2 εβδομάδων χρήστη				
	Μέσος χρόνος παιξίματος 2 εβδομάδων		Μη Ικανοποιητικός *(<=5)*	Ικανοποιητικός *(<=360)*	Μεγάλος *(>360)*
		Χαμηλός *(<=40)*	1	2	3
		Κανονικός *(<=110)*	2	3	4
		Μεγάλος *(>110)*	3	4	5

Πίνακας 17Απόδοση βαρών από αποφασίζων στο κριτήριο Χρήση

Στόχος μας είναι να δημιουργήσουμε ένα πολυκριτήριο πίνακα για τον κάθε χρήστη με όλες τις εναλλακτικές(παιχνίδια) και τα βάρη σε κάθε κριτήριο, τα οποία προκύπτουν από τους πίνακες 14,15,16,17. Έτσι λοιπόν δημιουργήσαμε έναν κώδικα ο οποίος δέχεται σαν είσοδο α. Πίνακας παιχνιδιών, β. Πίνακας χρηστών-παιχνιδιών, γ.Πίνακας χρηστών. Αυτοί οι πίνακες φιλτράρονται μέσα από τους ευρετικούς πίνακες 14,15,16,17 και δίνονται τα βάρη για κάθε εναλλακτική στο κάθε κριτήριο.

Παρακάτω παρουσιάζονται κομμάτια από κάθε πίνακα για καλύτερη κατανόηση των δεδομένων.

UserID	GamesCount	ForeverMean	2WeeksMean	2WeeksSum	PriceMean	PriceSum	GenCover
76561193665450065	53	2197.11	13.83	733.00	951.41	43765	9
76561193665450076	189	1563.07	22.67	4285.00	1470.40	251438	10
76561193665450079	85	1537.76	10.66	906.00	1327.21	94232	11
76561193665450084	32	1686.72	0.50	16.00	1183.04	29576	3
76561193665450089	86	502.06	16.57	1425.00	1531.51	117926	11

Πίνακας 18 Κομμάτι πίνακα χρηστών για δημιουργία πολυκριτήριου πίνακα χρήστη

Στον πίνακα 18 έχουμε ένα παράδειγμα από την είσοδο δεδομένων χρηστών. Αυτός ο πίνακας περιέχει τους χρήστες (UserID), τον αριθμό παιχνιδιών που κατέχουν(GamesCount), το μέσο χρόνο που παίζουν

κάθε παιχνίδι(ForeverMean), το μέσο χρόνο που έπαιξαν τις τελευταίες 2 εβδομάδες κάθε παιχνίδι (2WeeksMean), το σύνολο χρόνου που δαπάνησαν τις τελευταίες 2 εβδομάδες σε παιχνίδια (2WeeksSum), την μέση τιμή που ξοδεύουν ανά παιχνίδι(PriceMean), το σύνολο που ξοδέψανε συνολικά στα παιχνίδια(PriceSum) και τέλος το σύνολο διαφορετικών κατηγοριών παιχνιδιών που έχουν παίξει (GenCover).

UserID	appid	sumgenres
76561193665450065	10	1
76561193665450065	20	1
76561193665450065	30	1
76561193665450065	40	1
76561193665450065	50	1

Πίνακας 19 Κομμάτι πίνακα χρηστών-παιχνιδιών για δημιουργία πολυκριτήριου πίνακα χρήστη

Στον πίνακα 19 παρουσιάζεται ένα μέρος του πίνακα χρηστών παιχνιδιών ο οποίος περιέχει όλους τους χρήστες και όλα τα παιχνίδια που κατέχουν. Συγκεκριμένα η πρώτη στήλη περιέχει τον χρήστη (UserID, ο οποίος επαναλαμβάνεται μέχρι να τελειώσουν τα παιχνίδια του και μετά πάει στον επόμενο), τα παιχνίδια που κατέχει ο χρήστης (appid), και σε ποια κατηγορία ανήκει το παιχνίδι(sumgenres).

appid	final_price	metacritic	sumgenres	DaysFromRelease	UsersCount	2weeksMean	2weeksSum
10	799	88	1	5522	219062	994.6270223	4549424
30	399	79	1	4611	218624	574.4836957	211410
70	699	96	1	6246	218656	672.625	242145
80	799	65	1	4306	59827	652.9867841	148228
130	399	71	1	5310	218629	990.2142857	69315

Πίνακας 20 Κομμάτι πίνακα παιχνιδιών για δημιουργία πολυκριτήριου πίνακα χρήστη

Ο πίνακας 20 περιέχει όλα τα διαθέσιμα παιχνίδια (appid), την τιμή τους(final_price), την βαθμολογία τους(metacritic), την κατηγορία-ες στην οποία ανήκουν (sumgenres), το πόσες μέρες είναι σε κυκλοφορία (DaysFromRelease), πόσοι χρήστες το έχουν στην κατοχή τους (UserCount), πόσο παίχτηκε τις τελευταίες 2 εβδομάδες κάτα μέσο όρο(2weeksMean) και πόσο παίχτηκε τις τελευταίες 2 εβδομάδες συνολικά (2weeksSum).

6.3 Πολυκριτήριος Πίνακας

Ο πολυκριτήριος πίνακας αφορά έναν παίχτη και περιέχει όλα τα παιχνίδια που έχει στην κατοχή του. Η βαθμολογία του κάθε κριτηρίου για το κάθε παιχνίδι υπολογίζεται από την εξίσωση του. Για παράδειγμα στο κριτήριο Δαπάνες, έστω ότι ο χρήστης έχει μέσο κόστος παιχνιδιών Υψηλό και το κόστος παιχνιδιού είναι Μεσσαίο, τότε η βαθμολογία του κριτηρίου για τον συγκεκριμένο χρήστη στο συγκεκριμένο παιχνίδι, στο συγκεκριμένο κριτήριο είναι 3. Όπως δηλαδή προκύπτει από τους ευρετικούς πίνακες που αναλύσαμε στο 6.2. Στο τέλος προσθέτουμε και την τελευταία στήλη η οποία περιέχει την κατάταξη του παιχνιδιού από τους χρήστες, με αύξουσα κλίμακα από 0-100.

Παρακάτω παρουσιάζεται ένας παράδειγμα πολυκριτήριου πίνακα για έναν χρήστη αφού ολοκληρώθηκε όλη η διαδικασία απόδοσης βαρών. Ο πίνακας αυτός είναι η είσοδος για την εφαρμογή της μεθόδου Utastar στην επόμενη παράγραφο.

Πίνακας 21 Παράδειγμα πολυκριτήριου πίνακα

UserID	AppID	Poikilia	Empeiria	Dapanes	Xrisi	Global_score
76561193665450065	10	3	9	3	7	88
	20	3	9	5	7	NaN
	30	3	9	5	7	79
	40	3	9	5	7	NaN
	50	3	9	5	7	NaN
	60	3	9	5	7	NaN
	70	3	9	3	7	96
	130	3	9	5	7	71
	80	3	9	3	7	65
	220	3	9	3	7	96
	240	3	9	0	7	88
	280	3	9	5	7	NaN
	300	3	9	3	7	80
	320	3	9	5	7	NaN
	340	3	9	5	7	NaN
	360	3	9	5	7	NaN
	380	3	9	3	7	87
	2100	5	9	5	7	72
	2270	3	9	5	7	NaN
	9000	3	9	5	3	NaN
	9010	3	9	3	5	88
	400	3	9	3	7	90
	420	3	9	3	7	90
	13210	3	9	0	5	83
	500	3	9	0	7	89
	17480	3	9	3	7	82
	8980	5	9	0	7	81
	10180	3	9	0	7	86
	41500	3	9	0	7	83
	550	3	9	0	7	89
	9900	7	9	5	7	76
	42700	3	9	0	7	81
	48220	5	9	0	7	77
	620	5	9	0	7	95
	57900	3	9	0	5	54
	42680	3	9	0	7	NaN
	730	3	9	0	7	83
	113420	7	9	5	7	71
	200210	7	9	5	7	82
	209870	5	9	5	7	75
	226320	7	9	5	7	81
	282440	3	7	3	7	NaN
	290930	7	5	5	7	NaN
	304050	7	5	5	7	NaN
	289650	5	7	0	7	NaN
	339610	7	5	5	7	NaN

Όπως φαίνεται στον Πίνακα 21, ο αρχικός πολυκριτήριος για τον χρήστη με UserId: 76561193665450065 περιέχει όλες τις εναλλακτικές(παιχνίδια, appid), τις εκτιμήσεις των εναλλακτικών σε κάθε κριτήριο όπως προέκυψε από τους συνδυασμένους πίνακες απαιτήσεων – χαρακτηριστικών 14-17 καθώς και μια στήλη με κατάταξη από την συνολική βαθμολογία του παιχνιδιού.

6.4 Utastar

Για να υλοποιήσουμε την λύση μας στο μοντέλο της Utastar χρειαζόμαστε ως εισόδους τον πολυκριτήριο πίνακα (πίνακας 21), ένα πίνακα με τα χαρακτηριστικά των κριτηρίων (πίνακας 22) και τις σταθερές delta=0.005 και epsilon=0.0001.

Cri/attributes	Monotonicity	Type	Worst	Best	a
Poikilia	0	0	3	15	5
Empeiria	0	0	3	15	5
Dapanes	0	0	0	5	5
Xrhsh	0	0	3	15	5

Πίνακας 22 Πίνακας χαρακτηριστικών κριτηρίων

Ο πίνακας 21 είναι αυτός που χρησιμοποιούμε στον παράδειγμα μας. Η πρώτη στήλη περιέχει τα κριτήρια που έχουμε δημιουργήσει. Η δεύτερη στήλη αφορά την μονοτονία των κριτιρίων, 1 ή 0 για αύξουσα ή φθίνουσα αντιστοίχως. Η τρίτη στήλη αφορά τον τύπο, 1 ή 0 για ποιοτικό ή συνεχές αντίστοιχα. Η επόμενες δύο στήλες Worst και Best περιέχουν τις χειρότερες και τις καλύτερες, αντίστοιχα, δυνατές τιμές που μπορούν να πάρουν τα κριτήρια. Τέλος το a, είναι ένα στοιχείο το οποίο εισάγουμε εμείς ως αποφασίζων και αφορά στο πόσα διαστήματα θα γίνει ο χωρισμός. Εάν ο τύπος του κριτηρίου ήταν 1, δηλαδή συνεχές, θα έπρεπε το a να είναι ίσο με το μήκος της κλίμακας, δηλαδή εάν οι δυνατές τιμές ήταν eg[1 2 3 4 5] => a=5.

Εισάγοντας τα δεδομένα ξεκινάμε με το πρώτο στάδιο τις UTASTAR, να αποσαφηνίσουμε τις χρησιμότητες του κάθε κριτηρίου. Για να το κάνουμε αυτό όμως αρχικά επιλέγουμε τις εξής κλίμακες:

```
              0     1    2      3     4
Poikilia   3.0  6.00  9.0  12.00  15.0
Empeiria   3.0  6.00  9.0  12.00  15.0
Dapanes    0.0  1.25  2.5   3.75   5.0
Xrisi      3.0  6.00  9.0  12.00  15.0
```

Εικόνα 2 Κλίμακες κριτήριων Utastar

Στην συνέχεια για την επίλυση του γραμμικού προβήμματος ακολουθείται ο αλγόριθμος της UTASTAR(Παράρτημα 1) και έχουμε το ακόλουθο αποτέλεσμα για τον συγκεκριμένο χρήστη.

Παρακάτω περιέχονται η ολική χρησιμότητα κάθε εναλλακτικής, μερικές χρησιμότητες κριτηρίων, καθώς και τα βάρη του μοντέλου για τον χρήστη.

appid	Global utilities	appid	Global utilities	appid	Global utilities
10	0.01497	**48000**	0.018283333	**247660**	0.003313333
20	0.104945	**57300**	0.00497	**15320**	0.01497
70	0.01497	**204060**	0.018283333	**233130**	0.003313333
130	0.104945	**107100**	0.00497	**255520**	0.003313333
80	0.01497	**209830**	0.01994	**221910**	0.003313333
280	0.104945	**200010**	0.018283333	**227300**	0.003313333
300	0.01497	**50300**	0.003313333	**265930**	0.01994
320	0.104945	**201790**	0.00497	**211820**	0.00497
360	0.104945	**20530**	0.01497	**268870**	0.00497
2100	0.108258333	**20550**	0.018283333	**4500**	0.003313333
3920	0.01497	**202170**	0.003313333	**20510**	0.018283333
6200	0.104945	**211260**	0.018283333	**41700**	0.003313333
6860	0.01497	**4920**	0.01994	**271240**	0.003313333
1700	0.104945	**49520**	0.003313333	**280220**	0.00497
400	0.01497	**200710**	0.00497	**282400**	0.018283333
17410	0.018283333	**219150**	0.018283333	**236430**	0.003313333
23310	0.01497	**205100**	0.003313333	**287700**	0.003313333
35420	0.104945	**223470**	0.01994	**294860**	0.00497
25890	0.01497	**219890**	0.003313333	**296830**	0.00497
40700	0.018283333	**230410**	0.108258333	**301520**	0.109915
8190	0.003313333	**203160**	0.003313333	**40400**	0.018283333
33900	0.01994	**233450**	0.00497	**274170**	0.003313333
33930	0.00497	**113020**	0.00497	**332200**	0.00497
47780	0.01497	**222730**	0.01994	**249050**	0.01994
92800	0.018283333	**206190**	0.01994	**346110**	0.00497
91600	0.01994	**241600**	0.00497	**350070**	0.01994
620	0.003313333	**206420**	0.003313333	**255710**	0.003313333
22100	0.018283333	**225080**	0.00497	**341940**	0.003313333
105600	0.01994	**239030**	0.018283333	**391540**	0.018283333
107200	0.01994	**248820**	0.01994		
57690	0.003313333	**218620**	0.003313333		
113200	0.109915	**236850**	0.003313333		
65800	0.00497	**250260**	0.003313333		
41070	0.003313333	**250700**	0.003313333		
2820	0.00497	**250760**	0.00497		
201310	0.018283333	**231310**	0.018283333		
3830	0.01497	**249130**	0.003313333		

Πίνακας 23 Αποτελέσματα Utastar για έναν χρήστη. Ολικές χρησιμότητες παιχνιδιών

Από τον πίνακα 23 καταλαβαίνουμε ποια παιχνίδια είναι αυτά τα οποία ταιριάζουν περισσότερο στον χρήστη. Όσο πιο κοντά στην μονάδα τόσο το καλύτερο.

	Poikilia	Empeiria	Dapanes	Xrisi
0	0	0	0	0
1	0.00497	0	0.01497	0
2	0.00497	0	0.01497	0
3	0.449995	0.22253	0.01497	0.22253
4	0.449995	0.22253	0.104945	0.22253

Πίνακας 24 Αποτελέσματα UTASTAR για έναν χρήστη. Μερικές χρησιμότητες κριτηρίων.

	Model Weights
Poikilia	0.449995
Empeiria	0.22253
Dapanes	0.104945
Xrisi	0.22253

Πίνακας 25 Αποτελέσματα UTASTAR για έναν χρήστη. Βάρη μοντέλου.

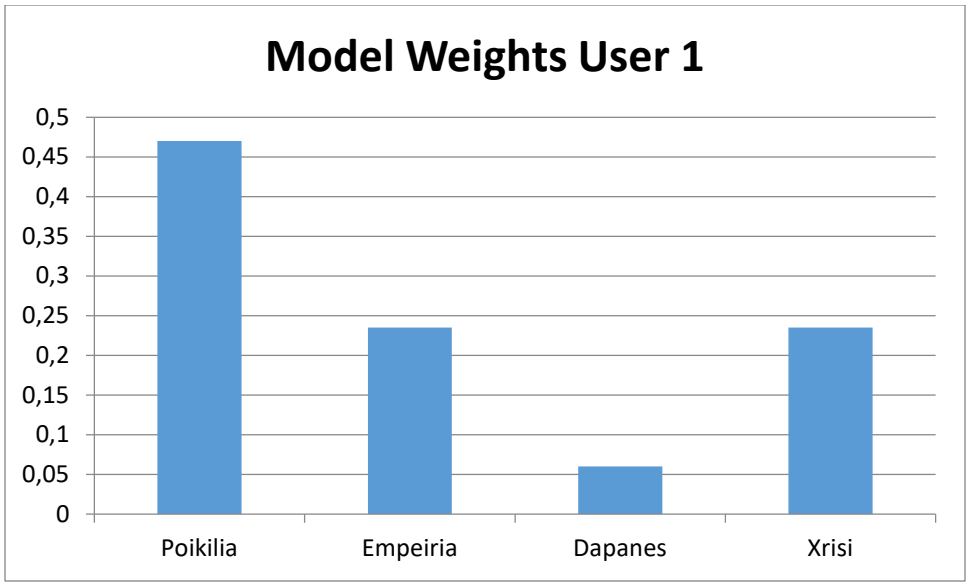
Στον συγκεκριμένο παράδειγμα, όπως προκύπτει από τους πίνακες 24 και 25, ο χρήστης δίδει μεγάλυτερη έμφαση στο κριτήριο Ποικιλία, τον ενδιαφέρει δηλαδή να δοκιμάζει πολλά παιχνίδια και σε διαφορετικές κατηγορίες.

Παρακάτω παρατίθενται άλλα δύο παραδείγματα από άλλους χρήστες όσον αφορά τα βάρη των κριτηρίων. Παρατηρείται διαφορά στις τιμές, το οποίο σημαίνει οτι ο αλγόριθμος δουλεύει με επιτυχία.

Όπως φαίνεται και στον Πίνακα 26, ο πρώτος χρήστης έχει μεγάλη προτίμηση στο κριτήριο Ποικιλία δηλαδή προτιμάει να δοκιμάζει διαφορετικά παιχνίδια αλλά έπειτα υπάρχει και μία ισόβαθμη σχεδόν προτίμηση στα κριτήρια Εμπειρία και Χρήση . Αυτό αναπαριστάται και στο διάγραμμα

Criteria	Model Weights User 1
Poikilia	0.469975
Empeiria	0.235
Dapanes	0.060025
Xrisi	0.235

Πίνακας 26 Αποτελέσματα-Βάρη Κριτηριών για 1ο χρήστη

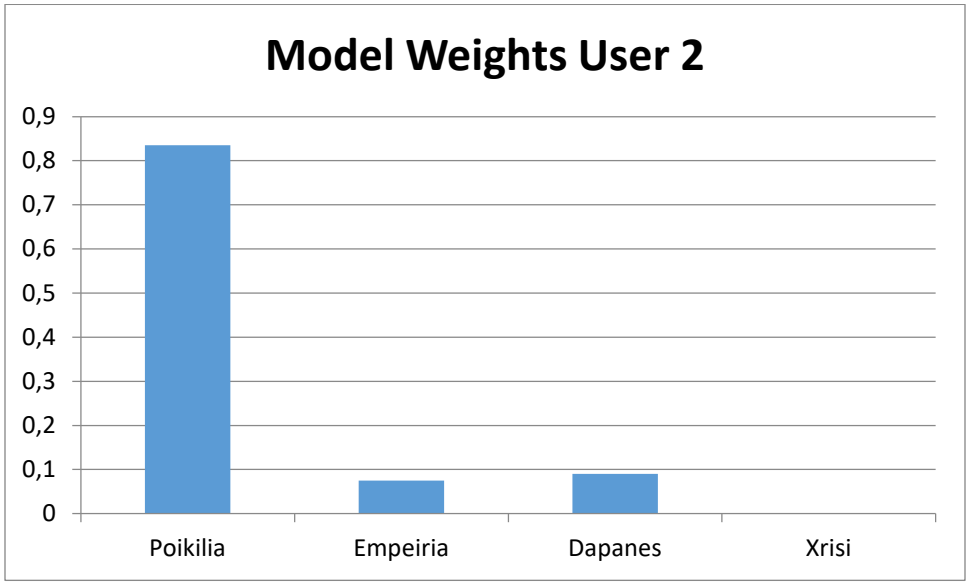


Διάγραμμα 8 Αποτελέσματα-Βάρη Κριτηρίων για 1ο χρήστη

Ο δεύτερος χρήστης ωστόσο φαίνεται να έχει μόνο ένα κριτήριο σημαντικό, το οποίο είναι η ποικιλία, και δεν δίνει καθόλου βάρος στα υπόλοιπα κριτήρια(Πίνακας 27).

Criteria	Model Weights User 2
Poikilia	0.8353
Empeiria	0.0747
Dapanes	0.09
Xrisi	0

Πίνακας 27 Αποτελέσματα-Βάρη Κριτηριών για 2ο χρήστη

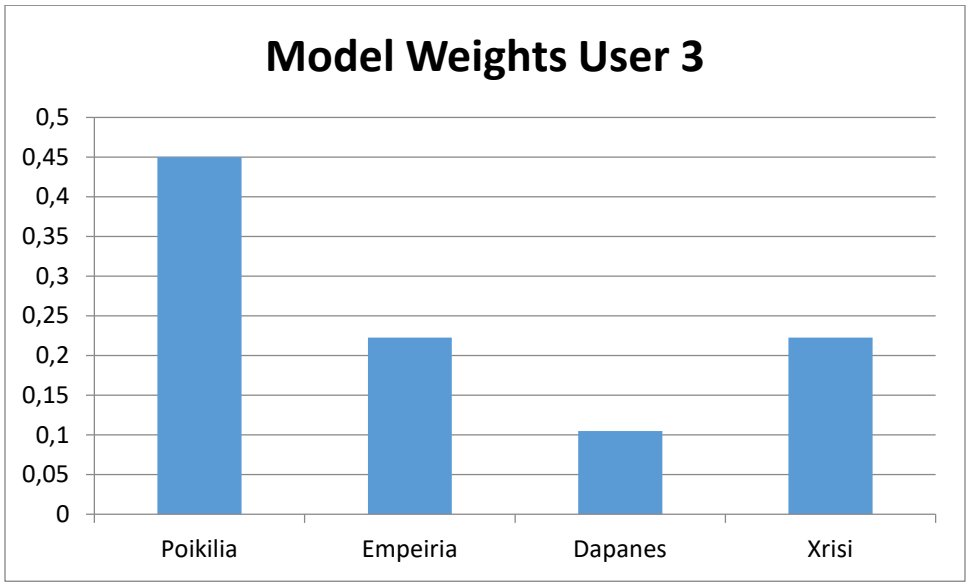


Διάγραμμα 9 Αποτελέσματα-Βάρη Κριτηριών για 2ο χρήστη

Τέλος, ο 3ος χρήστης παρουσιάζει σχεδόν ίδια αποτελέσματα με τον 1ο χρήστη(πίνακας 28) προτίμηση στο κριτήριο Ποικιλία και στην συνέχεια ισόβαθμη σχεδόν προτίμηση στα κριτήρια Εμπειρία και Χρήση.

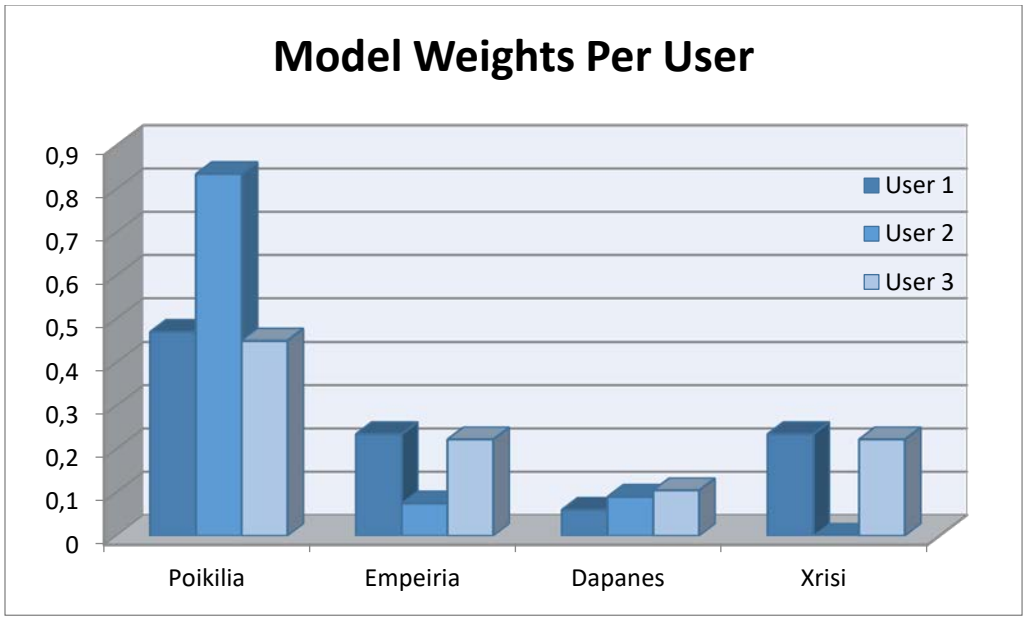
Criteria	Model Weights User 3
Poikilia	0.449995
Empeiria	0.22253
Dapanes	0.104945
Xrisi	0.22253

Πίνακας 28 Αποτελέσματα-Βάρη Κριτηριών για 3ο χρήστη



Διάγραμμα 10Αποτελέσματα-Βάρη Κριτηριών για 3ο χρήστη

Συγκεντρωτικά βλέπουμε πως ενώ κάθε χρήστης έχει διαφορετικές προτιμήσεις, σε μεγάλη κλίμακα θα υπάρχουν χρήστες οι οποίοι ανήκουν στις ίδιες συστάδες (Πίνακας 29).



Πίνακας 29 Συγκεντρωτική απεικόνιση βαρών για 3 χρήστες

Καταλήγοντας, παρατηρούμε ότι θα μπορούσε να υπάρξει συσχέτιση μεταξύ των παιχτών ανάλογα με τα χαρακτηριστικά τους και τις αποδόσεις τους σε κάθε κριτήριο. Δηλαδή παίκτες που έχουν παρόμοιες βαθμολογίες στα κριτήρια, δηλαδή ανήκουν στις ίδιες συστάδες, θα μπορούσαν να μοιράζονται και κοινό απόθεμα εναλλακτικών. Δηλαδή μέσα στην ίδια συστάδα-ομάδα παιχτών να προτείνονται τα παιχνίδια που δεν έχει παίξει κάποιος χρήστης άλλα είναι δημοφιλή στην συστάδα του.

Συμπεράσματα

Η διαδικασία της συλλογής δεδομένων ήταν μια απαιτητική εργασία καθώς χρειάστηκε να βρούμε πρωτότυπους τρόπους για να συλλέξουμε τις πληροφορίες που χρειαζόμασταν. Βέβαια ο τεράστιος όγκος δεδομένων οδήγησε σε δυσκολία ανάλυσης τους καθώς με τους παραδοσιακοί τρόπους δεν ήταν εφικτή η επεξεργασία τους. Βέβαια τα αποτελέσματα της ανάλυσης φάνηκαν πολύ ενδιαφέρον καθώς είχαμε καθαρή αντίληψη της καταναλωτικής συμπεριφοράς των παικτών.

Όσον αφορά την πολυκριτήρια ανάλυση υπήρχε πρόβλημα με τα δεδομένα εισόδου καθώς δεν υπήρχε ξεκάθαρη κατάταξη από τους παίκτες αλλά χρησιμοποιήθηκε η βαθμολογία του παιχνιδιού για την προτίμηση τους. Με αποτέλεσμα, για παράδειγμα, παιχνίδια τα οποία είχαν ξεκάθαρη σχέση υπεροχής ως προς τα κριτήρια να έχουν ίση βαθμολογία από το χρήστη. Αυτό μας οδήγησε σε πολύ χαμήλους δείκτες συσχέτισης t του kendall.

Ωστόσο φαίνεται από όλη την εργασία ότι υπάρχει ενδιαφέρον για ένα τέτοιο μοντέλο και είναι πιθανώς εφικτό με αλλάγη παραμέτρων και βελτίωση δεδομένων εισόδου.

Για μελλοντική εργασία προτείνεται βελτιστοποίηση του κώδικα συλλογής δεδομένων από το διαδίκτυο έτσι ώστε να λαμβάνονται τα δεδομένα στην μορφή που απαιτείται, με μικρότερη υπολογιστική ισχύ αλλά και να συλλέγουμε μόνο αυτά τα οποία είναι απαραίτητα για την διαδικασία. Ακόμα προτείνεται να χρησιμοποιηθεί κάποιο άλλο μέτρο κατάταξης στον πολυκριτήριο πίνακα καθώς το metacritic score φάνηκε αναξιόπιστο λόγω κενών εγγραφών(nan) και μη πραγματικής προτίμησης χρήστη.

Παράρτημα

1. Κώδικας δημιουργίας πολυκριτήριου πίνακα σε python 2.7

```
import pandas as pd
import numpy as np
import sys, traceback
import matplotlib.pyplot as plt
pd.set_option('display.height', 1000)
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

apps_df = pd.read_csv('AppCriteriaFinal.csv', sep='\t')
apps_df=apps_df.set_index(['appid'])
apps_df=apps_df.drop(['Unnamed: 0'],1)
apps_df = apps_df.rename(columns={'2weeksMean': 'weeks2Mean', '2weeksSum': 'weeks2Sum'})
users_df = pd.read_csv('UserCriteriaTraining.csv', sep='\t')
users_df = users_df.set_index(['UserID'])
users_df = users_df.rename(columns={'2WeeksMean': 'Weeks2Mean', '2WeeksSum': 'Weeks2Sum'})
users_df=users_df.drop(['Unnamed: 0'],1)
user_app = pd.read_csv('ConcatOwnedTrainingF.csv', sep='\t', usecols=['UserID','appid','sumgenres'])
user_app = user_app.set_index(['UserID','appid'])
us=users_df
us=us.reset_index()

print(apps_df.head())
print(users_df.head())
print(user_app.head())

def making_criteria(list_user,list_app):
    ##  Poikilia
  #sub1
```

```
if list_user.GamesCount.any()<=13:
    if list_app.sumgenres <=1:
        sub_kalipsi1 = 1
    elif (list_app.sumgenres > 1) and (list_app.sumgenres<=2):
        sub_kalipsi1 = 2
    else:
        sub_kalipsi1 = 3
elif list_user.GamesCount.any()>13 and list_user.GamesCount.any()<=30:
    if list_app.sumgenres <=1:
        sub_kalipsi1 = 2
    elif list_app.sumgenres>1 and list_app.sumgenres<=2:
        sub_kalipsi1 = 3
    else:
        sub_kalipsi1 = 4
else:
    if list_app.sumgenres <=1:
        sub_kalipsi1 = 3
    elif list_app.sumgenres>1 and list_app.sumgenres<=2:
        sub_kalipsi1 = 4
    else:
        sub_kalipsi1 = 5
#sub2
if list_user.GenCover.any() <=3:
    sub_kalipsi2 = 1
elif list_user.GenCover.any()>3 and list_user.GenCover.any()<=9:
    sub_kalipsi2 = 3
else:
    sub_kalipsi2 = 5

criterion_poikilia= 2*sub_kalipsi1+1*sub_kalipsi2
```

```
##   Empeiria
#sub1
if list_user.ForeverMean.any()<=180:
    if list_app.DaysFromRelease <=360:
        sub_empeiria1 = 1
    elif (list_app.DaysFromRelease > 360) and (list_app.DaysFromRelease<=720):
        sub_empeiria1 = 2
    else:
        sub_empeiria1 = 3
elif list_user.ForeverMean.any()>180 and list_user.ForeverMean.any()<=720:
    if list_app.DaysFromRelease <=360:
        sub_empeiria1 = 2
    elif list_app.DaysFromRelease>360 and list_app.DaysFromRelease<=720:
        sub_empeiria1 = 3
    else:
        sub_empeiria1 = 4
else:
    if list_app.DaysFromRelease <=360:
        sub_empeiria1 = 3
    elif list_app.DaysFromRelease>360 and list_app.DaysFromRelease<=720:
        sub_empeiria1 = 4
    else:
        sub_empeiria1 = 5

#sub2
if list_user.GamesCount.any()<=13:
    if list_app.UsersCount <=50:
        sub_empeiria2 = 1
    elif (list_app.UsersCount > 50) and (list_app.UsersCount<=350):
        sub_empeiria2 = 2
    else:
        sub_empeiria2 = 3
elif list_user.GamesCount.any()>13 and list_user.GamesCount.any()<=30:
```

```
    if list_app.UsersCount <=50:
        sub_empeiria2 = 2
    elif list_app.UsersCount>50 and list_app.UsersCount<=350:
        sub_empeiria2 = 3
    else:
        sub_empeiria2 = 4
else:
    if list_app.UsersCount <=50:
        sub_empeiria2 = 3
    elif list_app.UsersCount>50 and list_app.UsersCount<=350:
        sub_empeiria2 = 4
    else:
        sub_empeiria2 = 5

criterion_empeiria = 2*sub_empeiria1+1*sub_empeiria2

##Dapanes
if list_user.PriceMean.any()<=499.0:
    if list_app.final_price <=499.0:
        sub_dapanes1 = 5
    elif (list_app.final_price > 499.0) and (list_app.final_price<=999.0):
        sub_dapanes1 = 3
    else:
        sub_dapanes1 = 0
elif list_user.PriceMean.any()>499.0 and list_user.PriceMean.any()<=999.0:
    if list_app.final_price <=499.0:
        sub_dapanes1 = 3
    elif list_app.final_price>499.0 and list_app.final_price<=999.0:
        sub_dapanes1 = 5
    else:
        sub_dapanes1 = 3
else:
    if list_app.final_price <=499.0:
```

```
        sub_dapanes1 = 1
    elif list_app.final_price>499.0 and list_app.final_price<=999.0:
        sub_dapanes1 = 3
    else:
        sub_dapanes1 = 5

criterion_dapanes=sub_dapanes1

##Xrhsh
#sub1
if list_user.ForeverMean.any() <=180.0:
    sub_xrisi1 = 1
elif list_user.ForeverMean.any()>180.0 and list_user.ForeverMean.any()<=720.0:
    sub_xrisi1 = 3
else:
    sub_xrisi1 = 5

if list_user.Weeks2Sum.any()<=5:
    if list_app.weeks2Mean <=40.0:
        sub_xrisi2 = 1
    elif (list_app.weeks2Mean > 40.0) and (list_app.weeks2Mean<=110.0):
        sub_xrisi2 = 2
    else:
        sub_xrisi2 = 3
elif list_user.Weeks2Sum.any()>5 and list_user.Weeks2Sum.any()<=360:
    if list_app.weeks2Mean <=40.0:
        sub_xrisi2 = 2
    elif list_app.weeks2Mean>40.0 and list_app.weeks2Mean<=110.0:
        sub_xrisi2 = 3
    else:
        sub_xrisi2 = 4
else:
    if list_app.weeks2Mean <=40.0:
```

```
            sub_xrisi2 = 3
        elif list_app.weeks2Mean>40.0 and list_app.weeks2Mean<=110.0:
            sub_xrisi2 = 4
        else:
            sub_xrisi2 = 5

    criterion_xrisi= 1*sub_xrisi1 +2*sub_xrisi2

    return [criterion_poikilia, criterion_empeiria, criterion_dapanes, criterion_xrisi, list_app.metacritic]

def multi_matrix(user1,users_df,apps_df,user_app):
    list_tuples=[]
    apps=user_app.loc[user1]
    list_user=users_df.loc[user1]
    k=0
    for index, row in apps.iterrows():
        app = index
        try:
            list_app=apps_df.loc[app]
            test=making_criteria(users_df,list_app)
            tup=(user1,app)
            list_tuples.append(tup)
            if k==0:
                c = np.array(test)
            else:
                a = np.array(test)
                c = np.vstack((c,a))

            k=k+1

        except:
            exc_type, exc_value, exc_traceback = sys.exc_info()
            #print exc_type, exc_value, exc_traceback
```

```
    index = pd.MultiIndex.from_tuples(list_tuples, names=['UserID', 'AppID'])
    s = pd.DataFrame(c, index=index, columns=['Poikilia','Empeiria','Dapanes','Xrisi', 'Global_score'])
    return s

k=0
for user in us.UserID:
    print k
    if k==0:
        mult=multi_matrix(user,users_df,apps_df,user_app)
    elif k==1:
        break
    else:
        mult1=multi_matrix(user,users_df,apps_df,user_app)
        mult = pd.concat([mult, mult1])



    k=k+1
    print k

print(mult)
df1 = mult[['Poikilia','Empeiria','Dapanes','Xrisi','Global_score']]
df1.to_csv(' user_mult_matrix.csv')
```

Βιβλιογραφία

1. Siskos. "UTA Methods." Multiple criteria decision analysis: State of the art surveys (2005): 297-344.
2. Siskos, Jacquet-Lagreze and. "Assesing a set of additive utility functions for multicriteria decision-making, the UTA method." European Journal of Operational Research (1982): 151-164.
3. Siskos, Y., E. Grigoroudis, N.F. Matsatsinis (2005), UTA methods, in: J. Figueira, S. Greco, M. Ehrgott (eds.), Multiple Criteria Decision Analysis, - State of the Art - Surveys, International Series in Operations Research and Management Science, pp. 297-344, Springer.
4. Yannacopoulos, Siskos and. "An ordinal regression method for building additive value functions." Investagacao Operacional (1985): 39-53.
5. Σίσκος. Γραμμικός Προγραμματισμός. Αθήνα: Εκδόσεις Νέων Τεχνολογιών, 1998.
6. —. Μοντέλα Αποφάσεων. Μεθοδολογία Επιχειρησιακής Έρευνας. Θεωρία Πολυκριτήριας Ανάλυσης. Εφαρμογές σε Επιχειρήσεις και Οργανισμούς. Αθήνα: Εκδόσεις Νέων Τεχνολογιών, 2008.
7. Adomavicius, Gediminas, Nikos Manouselis, and Youngok Kwon. “Multi-Criteria Recommender Systems.” Recommender systems … Mcdm (2011): 769–803. Web.
8. Chambers, Chris et al. “Measurement-Based Characterization of a Collection of On-Line Games.” IMC ’05 Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (2005): 1–14. Web.
9. Drachen, Anders, and Matthias Schubert. “Spatial Game Analytics.” Game Analytics (2013): 365–402. Web.
10. Lakiotaki, Kleanthi. “An Integrated Recommender System Based on Multi-Criteria Decision Analysis and Data Analysis Methods: Methodology, Implementation and Evaluation.” December (2010): PhD Dissertation, Technical University of Crete.
11. Lim, Chong-U, and D. Fox Harrell. “Modeling Player Preferences in Avatar Customization Using Social Network Data.” Proceedings of IEEE Conference on Computational Intelligence in Games 2 (2013): 153–160. Print.
12. Matsatsinis, Nikolaos, and Pavlos Delias. “AgentAllocator : An Agent-Based Multi-Criteria Desicion System for Task Allocation.” (2003): n. pag. Print.
13. Sifa, Rafet, Christian Bauckhage, and Anders Drachen. “Archetypal Game Recommender Systems.” September (2014): 8–10. Web.

14. ---. “The Playtime Principle: Large-Scale Cross-Games Interest Modeling.” 2014 IEEE Conference on Computational Intelligence and Games (2014): 1–8. Web.
15. Srivastava, Jaideep, et al. "Web usage mining: Discovery and applications of usage patterns from web data." ACM SIGKDD Explorations Newsletter 1.2 (2000): 12-23.
16. Vakali, Athena, Jaroslav Pokorný, and Theodore Dalamagas. "An overview of web data clustering practices." Current Trends in Database Technology-EDBT 2004 Workshops. Springer Berlin Heidelberg, 2004.
17. Mobasher, Bamshad, et al. "Integrating web usage and content mining for more effective personalization." Electronic commerce and web technologies. Springer Berlin Heidelberg, 2000. 165-176.
18. Cooley, Robert, Bamshad Mobasher, and Jaideep Srivastava. "Web mining: Information and pattern discovery on the world wide web." Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on. IEEE, 1997.
19. Halgamuge, Saman K., and Lipo Wang, eds. Classification and clustering for knowledge discovery. Vol. 4. Springer Science & Business Media, 2005.
20. Wong, Kevin. "Methodology." NextVideoGame.com. 2014. Web.
21. Galpin, Ixent & Gray, Alasdair & A A Fernandes, Alvaro & W Paton, Norman & Kotsifakos, Alexis & Kotsakos, Dimitris & Gunopulos, Dimitrios. (2018). Data Requirements, Data Management and Analysis Issues, and Query-Based Functionalities.
22. Ματσατσίνης, Νικόλαος Φ. Συστήματα υποστήριξης αποφάσεων / Νικόλαος Ματσατσίνης. - 1η έκδ. - Αθήνα : Εκδόσεις Νέων Τεχνολογιών, 2010. - 1008σ.
23. Galpin, Ixent & Gray, Alasdair & A A Fernandes, Alvaro & W Paton, Norman & Kotsifakos, Alexis & Kotsakos, Dimitris & Gunopulos, Dimitrios. (2018). Data Requirements, Data Management and Analysis Issues, and Query-Based Functionalities.
24. Lakiotaki, K., N. Matsatsinis, A. Tsoukias (2011), Multi-Criteria User Profiling in Recommender Systems, IEEE Intelligent Systems, vol. 26, no.2, pp. 64 – 76.