Ανάλυση καλαθιού σουπερμάρκετ

Μάιος 2021

Πανεπιστήμιο Θεσσαλίας

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα:Εξόρυξη Δεδομένων

Εαρινό εξάμηνο 2020-2021

Μέλη:Βασίλης Θείου Κοκάρας Μενέλαος

Εργασία με θέμα:Ανάλυση καλαθιού σουπερμάρκετ

Εισαγωγή

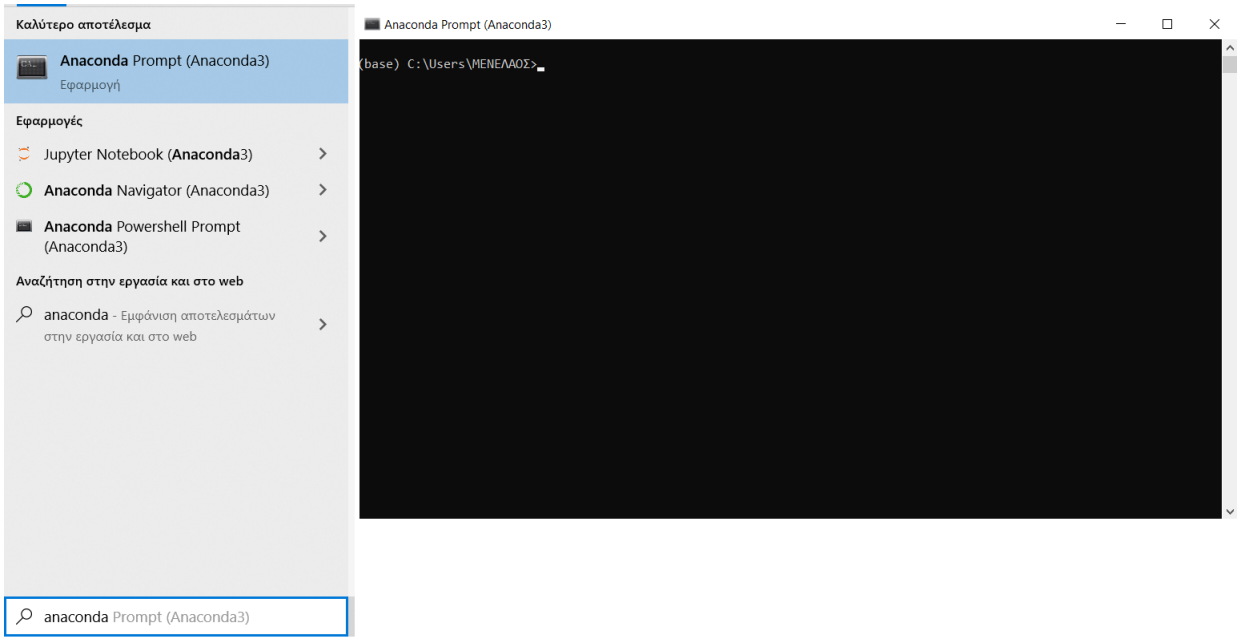
Στο συγκεκριμένο αρχείο δίνονται οδηγίες για την δημιουργία μιας εφαρμογής που αναλύει μια σειρά αγορών σε ένα σουπερμάρκετ και βγάζει χρήσιμα συμπεράσματα όπως δημοφιλή προιόντα και συσχετίσεις μεταξύ προιοντών.Τα συμπεράσματα αυτά μπορούν να γίνουν ωφέλιμα για το σούπερμαρκετ αφού κάνουν πιο αποτελεσματικές τις διαφημίσεις καθώς και τον τρόπο που είναι κατανεμημένα τα προιόντα στα ράφια κάτι που πιθανότατα θα προκαλέσει αύξηση των πωλήσεων.Παρακάτω περιγράφονται οι οδηγίες εγκατάστασης των εργαλείων που χρησιμοποιήθηκαν και η δόμη του προγράμματος.Επίσης υπάρχουν διάφορες δοκιμαστικές εκτελέσεις και μια αξιολόγηση των αποτελεσμάτων.

Εγκατάσταση του περιβάλλοντος

Η συγκεκριμένη εφαρμογή έγινε σε περιβάλλον anaconda και ο κώδικας γράφτηκε σε python σε Spyder IDE.Για την εγκατάσταση των παραπάνω υπάρχει ένας σύνδεσμος στο τέλος του αρχείου.Όσον αφορά τις βιβλιοθήκες οι περισσότερες είναι ήδη προεγκατεστημένες.Ωστόσο χρειάζονται αλλες δύο:Η 1)mlxtend που περιέχει τους αλγορίθμους για την ανάλυση των αγορών και 2) pysimplegui για το

γραφικό περιβάλλον.Για την εγκατάσταση αυτών χρειάζονται τα παρακάτω βήματα:

1)Άνοιγμα του anaconda prompt κάνωντας δεξί κλικ και εκτέλεση με δικαιώματα διαχειριστή.



2)Στο prompt πληκτρολογήστε τις παρακάτω εντολές:

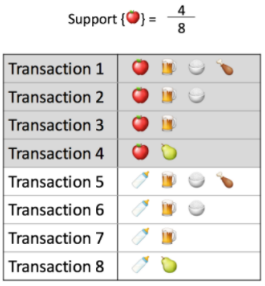
α) pip install pysimplegui
και
β) pip install mlxtend

3)Για το άνοιγμα του spyder χρειάζεται απλά ¨Αναζήτηση' και αριστερό κλικ στο πρόγραμμα.

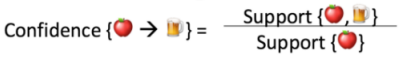
Θεωρία

Όπως αναφέρθηκε παραπάνω ο σκοπός της εφαρμογής είναι να βρει δημοφιλή αντικείμενα και συσχετίσεις μεταξύ αυτών των αντικειμένων.Αυτό γίνεται με την χρήση association rules και την χρήση τουapriori αλγορίθμου.Ο ορισμός της ανάλυσης association rules είναι:μία τεχνική για να ανακαλύψεις τον τρόπο με τον οποίο κάποια αντικείμενα συνδέονται μεταξύ τους.Υπάρχουν οι παρακάτω τρόποι για να μετρήσει κάποιος την συσχέτιση μεταξύ των αντικειμένων:
1)**Support**:Αυτό μας λέει πόσο δημοφιλές είναι ένα αντικείμενο.Για παράδειγμα

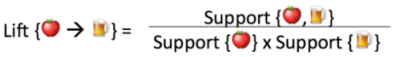
στον παρακάτω πίνακα το support του μήλου είναι $\frac{4}{8}$.Επίσης μπορούμε να έχουμε και support για περισσότερα από ένα αντικείμενα.Για παράδειγμα το support του {apple,bear,rice} είναι ίσο με $\frac{2}{8}$.



2)**Confidence**:Αυτό μας λέει πόσο πίθανο είναι να αγοραστεί ένα αντικείμενο Υ όταν αγοράζεται ένα αντικείμενο Χ.Ο τρόπος υπολογισμού φαίνεται στην παρακάτω εικόνα:(Ο όρος {Χ ⇒ Υ} συμβολίζει αγορά του Υ οταν αγοράζεται το Χ.)



3)**Lift**:Αυτό μας λέει πόσο πιθανό είναι να αγοραστεί ένα αντικείμενο Υ όταν αγοράζεται ένα αντικείμενο Χ ελέγχωντας παράλληλα την δημοφιλία του αντικειμένου Υ.Αν το **Lift** είναι μεγαλύτερο του 1 είναι πιθανό να αγοραστεί το Υ αν αγοραστεί το Χ,αν είναι μικρότερο του 1 δεν είναι πιθανό να αγοραστεί το Υ αν αγοραστεί το Χ και τέλος αν είναι ίσο με 1 δεν υπάρχει κάποια συσχέτιση.Ο τρόπος υπολογισμού φαίνεται στην παρακάτω εικόνα:

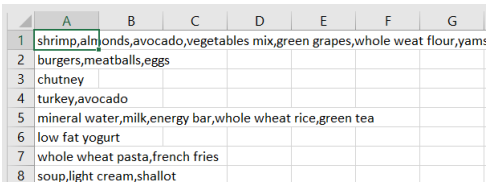


Στην εφαρμογή,γίνεται χρήση μόνο του **support** για την δημοφιλία και του **lift** για την συσχέτιση.

Apriori Algorithm:Ο σκοπος του συγκεκριμένου αλγορίθμου είναι να μειώσει τα σετ που πρέπει να εξετάσουμε.Αυτο γίνεται με την λογική ότι αν ένα αντικειμενο δεν είναι συχνό(έχει μικρό support) και τα σετ του δεν είναι σύχνα.Για παράδειγμα αν η μπύρα δεν είναι συχνή αγορά τότε και η αγορά μπύρα μαζι με πίτσα δεν είναι συχνή.Άρα δε χρειάζεται να ληφθεί υπόψιν.Για περισσότερες λεπτομέρειες πάνω στην θεωρία υπάρχουν στο τέλος οι αντίστοιχοι σύνδεσμοι.

Ανάπτυξη της εφαρμογής

Σε αυτό το κομμάτι θα αναλυθεί το τεχνικό κομμάτι της δημιουργίας της εφαρμογής χωρίς να γινει εκτεταμένη αναφορά στον κώδικα.Το πρώτο πράγμα που πρέπει να γίνει είναι να βρούμε ένα σετ δεδομένων και να το μεταλλάξουμε έτσι ώστε να γίνει ιδανικό για τη χρήση του στις συναρτήσεις που χρειαζόμαστε.Η επέκταση του σετ πρέπει να είναι .csv ωστόσο αυτό αλλάζει εύκολα.Το σημαντικό κομμάτι είναι η δομή του αρχείου η οποία είναι η παρακάτω(κάθε γραμμή να έχει το κάθε προιον χωρισμένο με κόμμα):



	A
1	shrimp,almonds,avocado,vegetables mix,green grapes,whole weat flour,yams
2	burgers,meatballs,eggs
3	chutney
4	turkey,avocado
5	mineral water,milk,energy bar,whole wheat rice,green tea
6	low fat yogurt
7	whole wheat pasta,french fries
8	soup,light cream,shallot

Στη συνέχεια διαβάζοντας τα δεδομένα,πρέπει να τα αποθηκεύσουμε στον παρακάτω τύπο δεδομένων και σε παρόμοια μορφή:

```
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]
```

Ύστερα,κάνοντας χρήση του TransactionEncoder απο την βιβλιοθήκη mlxtend καθώς και της συνάρτησης DataFrame από την βιβλιοθήκη pandas τα δεδομένα μας καταλήγουν σε αυτή την μορφή:

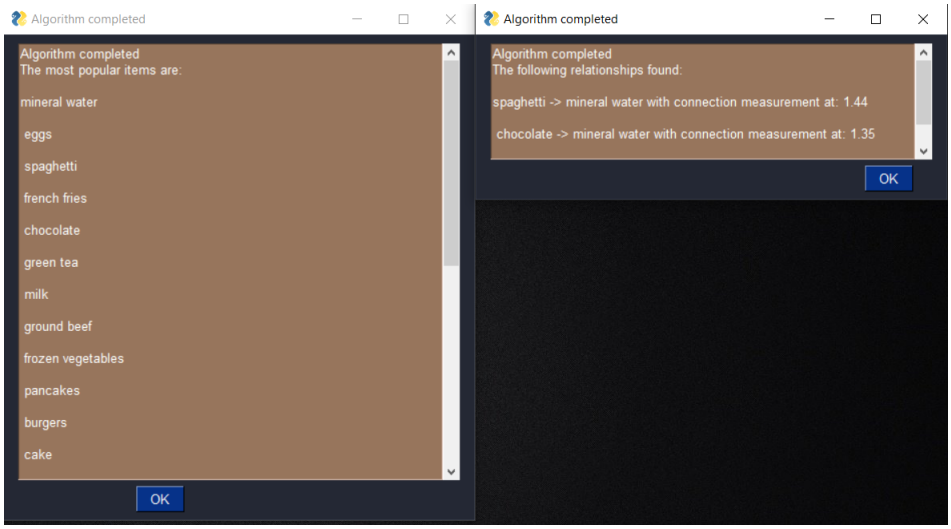
	Apple	Bananas	Beer	Chicken	Milk	Rice
0	True	False	True	True	False	True
1	True	False	True	False	False	True
2	True	False	True	False	False	False
3	True	True	False	False	False	False
4	False	False	True	True	True	True
5	False	False	True	False	True	True
6	False	False	True	False	True	False
7	True	True	False	False	False	False

Τώρα,ειμαστε έτοιμοι για το δεύτερο κομμάτι που είναι η κλήση των δύο βασικών συναρτήσεων.Αυτές είναι οι: **apriori,assosiaction_rules**.Οι κλήσεις τους στον κώδικα της εφαρμογής είναι αυτές:
frequent_itemsets = apriori(df, min_support=0.03, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
Στην συνάρτηση apriori το df είναι το σετ δεδομένων και το min_support είναι το ελάχιστο support που θα μιλήσουμε για την τιμή του αργότερα.Όσον αφορά την συνάρτηση assosiaction_rules είναι αυτη που παράγει το lift για κάθε συνδιασμό αντικειμένων.Όπως φαίνεται έχει επιλεγεί ως το metric το lift με ελάχιστη τιμή το 1 αφού οπως αναφέρθηκε στην θεωρία τότε μόνο έχουμε πιθανή συσχέτιση.

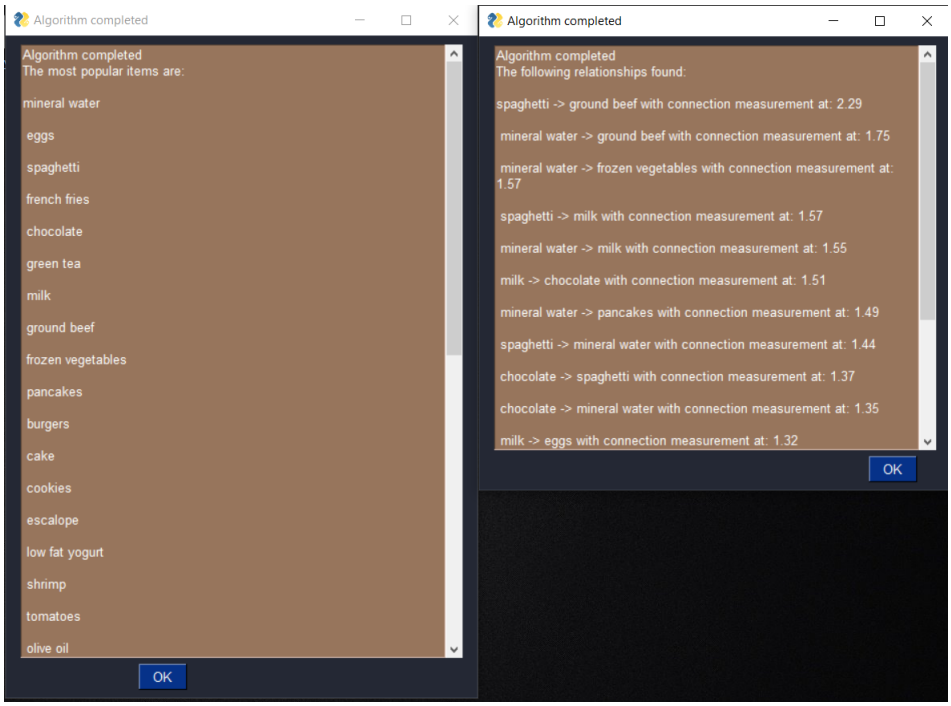
Το τελικό κομμάτι είναι η δημιουργία του γραφικού περιβάλλοντος.Αύτο έγινε με την χρήση της βιβλιοθήκης PySimpleGUI.Μέσω αυτού επιτυγχάνεται η ευκολία του χρήστη να επιλέξει αρχείο από το filesystem.Δίνονται επίσης οι επιλογές στον χρήστη να επιλέξει τι θέλει να εμφανιστεί(δημοφιλία και συσχέτιση).Τέλος,τα αποτελέσματα που πήραμε απο τις συναρτήσεις εκτυπώνονται στα παράθυρα που εμφανίζονται.Για να γίνει αυτό,χρειάζονται κάποια προεργασία ώστε να αλλάξει ο τύπος δεδομένων τους,να είναι πιο φιλικά προς στον χρήστη,να εμφανίζονται από την μεγαλύτερη τιμή προς την μικρότερη και να μην υπάρχουν duplicates.

Αξιολόγηση αποτελεσμάτων

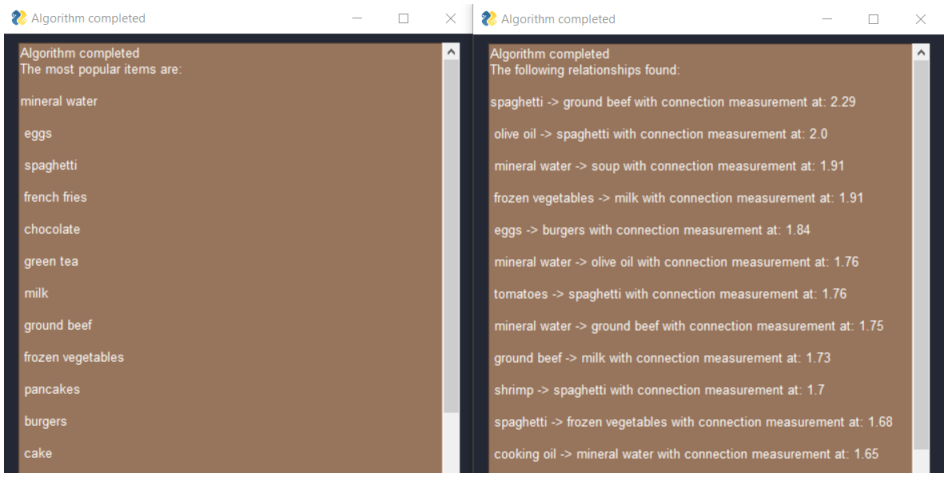
Παρόλο που τα αποτελέσματα είναι αρκετά ξεκάθαρα,πολύ σημαντικη είναι και η αξιολόγηση τους ώστε να πάρθουν σώστες αποφάσεις για τα προιόντα.Όπως προαναφέρθηκε,στην συνάρτηση apriori το min_support καθορίζει αν τα αποτελέσματα μας είναι χρήσιμα ή όχι.Δυστυχώς δεν υπάρχει κάποιος αξιοσημείωτος τρόπος για να βρούμε αυτή την τιμή πέρα απο την εμπειρία και το αντίστοιχο dataset.Αν η τιμή αυτά είναι υπερβολικά μεγάλη πιθανότατα να μην έχουμε αποτέλεσμα.Στη συνέχεια ας εξετάσουμε την παρακάτω εκτέλεση με support = 0.05.



Όπως φαινεται,παρόλο που έχουμε αποτελέσματα οι συσχετίσεις που προκύπτουν δε μας είναι ιδιαίτερα χρήσιμες.Αυτό διότι το mineral water όπως φαίνεται στο αριστερό παράθυρο είναι το πιο διάσημο προιόν και αρκετά διάσημα είναι και τα spaghetti,chocolate.Άρα είναι λογικό να υπάρχει κάποια συσχέτιση αφού όλα τα προιόντα αγοράζονται τακτικά σε γενικές γραμμές.
Ας μειώσουμε την τιμή του support σε 0.03.Τα αποτελέσματα είναι τα παρακάτω:



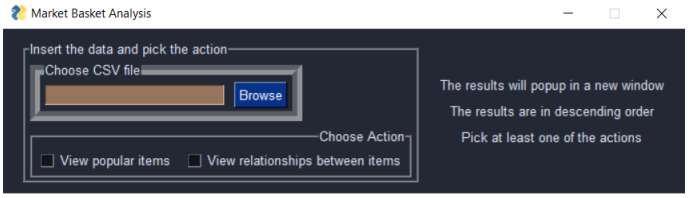
Μπορείτε να δείτε ότι πλέον έχουμε πολλές συσχετίσεις μεταξύ αντικειμένων.Παρόλα αυτά,ακομη υπάρχουν συσχετίσεις που δε μας δίνουν κάποια ιδιατέρη πληροφορία όπως του mineral water.Ωστόσο,η μεγαλύτερη συσχέτιση είναι αυτή μεταξύ του ground beef και του spaghetti.Παρόλο που είναι αρκετά γνώστα και τα δύο η τίμη συσχέτισης είναι αρκετα υψηλή σε σχέση με τις υπόλοιπες.Άρα ίσως χρειάζεται να ληφθεί υπόψιν ως συσχέτιση.
Ας μειώσουμε την τιμή του support σε 0.02:



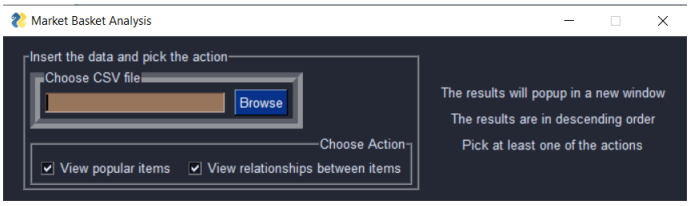
Βλέπουμε,ότι έχουμε αρκετα χρησιμα αποτελέσματα Όπως spaghetti με olive oil και mineral water με soup.
Να σημειωθεί τέλος ότι η μείωση του support δεν συνεπάγεται με καλύτερα αποτελέσματα.Αν το support ενός αντικειμένου είναι υπερβολικά μικρό το lift αυξάνεται σύμφωνα με τον τύπο που δώθηκε στην θεωρία.
Καταλήγουμε στο συμπέρασμα,ότι για να αξιολογήσουμε αν μία συσχέτιση είναι χρήσιμη η όχι πρέπει να γνωρίζουμε και την δημοφιλία των αντικειμένων γιατί επηρεάζουν την τιμή της συσχέτισης.

Εκτέλεση του προγράμματος

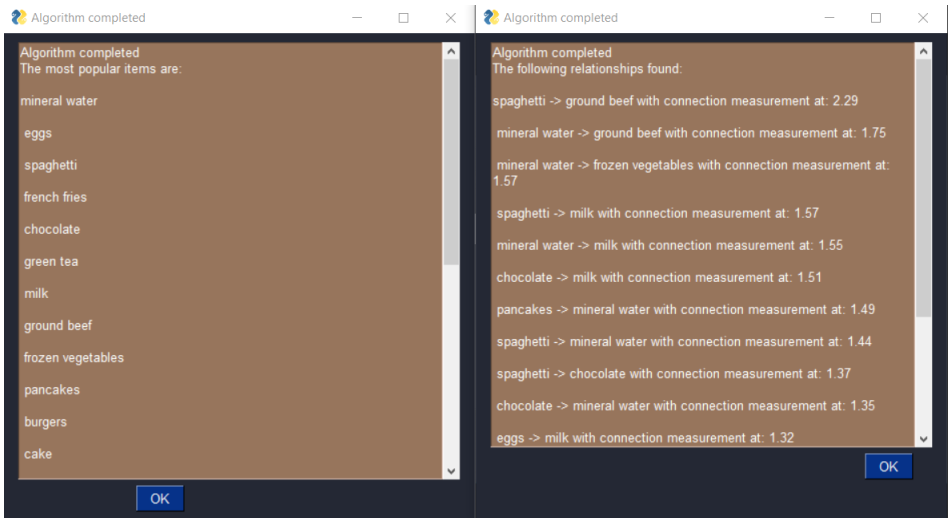
Παρακάτω θα δούμε ενα παραδείγμα εκτέλεσης της εφαρμογής.Το interface της εφαρμογής είναι αυτό:



Στη συνέχεια διαλέγουμε τουλάχιστον μία απο τις επιλογές.Στο συγκεκριμένο παράδειγμα επιλέγουμε και τις δύο.



Στην συνέχεια πατώντας το κουμπί Browse επιλέγουμε ένα CSV αρχείο.Τα αποτελέσματα που προκύπτουν είναι τα παρακάτω:



Όπως φαίνεται στα αριστερά υπάρχουν τα πιο δημοφιλή αντικείμενα(έχοντας ως βάση μια συγκεκριμένη τιμή δημοφιλίας) και δεξιά διάφορες συσχετίσεις μεταξύ των προιόντων.

Πηγές

Association rules theory:
https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

Apriori and Assocation rules in Python:
http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules
http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/

PySimpleGui docs:
https://pysimplegui.readthedocs.io/en/latest/

Dataset:
https://www.kaggle.com/sindraanthony9985/marketing-data-for-a-supermarket-in-united-states