



Project Report

Book Dataset

Μάθημα: Επιχειρηματική Ευφυΐα και Ανάλυση Μεγάλων Δεδομένων

Ομάδα: 13

Μπούρα Αγγελική, AM: 8190118

Σοφία Σωτηροπούλου, AM: 8190159

Περιεχόμενα

1. Περιγραφή Dataset.....	2
2. Διαδικασία ETL.....	3
Καθαρισμός δεδομένων:.....	3
Δημιουργία και Ανανέωση Διαστάσεων.....	4
Δημιουργία και Ανανέωση Fact Table	5
3. Star Schema	6
4. Κύβος.....	7
5. Dashboards με Power BI	8
6. Μοντέλα Data Mining	12
7. Παράρτημα.....	19

1. Περιγραφή Dataset

Για τις ανάγκες της εργασίας στο πλαίσιο του μαθήματος, αφού ψάξαμε σε σχετικές πηγές ώστε να βρούμε ένα dataset που να μας ενδιαφέρει και ταυτόχρονα να πληροί τις προϋποθέσεις της ανάλυσης, καταλήξαμε στην επιλογή του [Books Dataset](#) από την πλατφόρμα Kaggle. Το dataset αποτελείται από βιβλία που διαβάστηκαν από χρήστες και τις βαθμολογίες τους στο marketplace της Amazon. Τα δεδομένα συλλέχθηκαν από τον Cai-Nicolas Ziegler σε μία αναζήτηση 4 εβδομάδων (August / September 2004) της Book-Crossing κοινότητας με την άδεια του Ron Hornbaker, CTO of Humankind Systems. Το Dataset υπόκειται υπό το license: “CC0 1.0 Universal (CC0 1.0) Public Domain Dedication”.

Αναλυτικότερα, το Books Dataset αποτελείται από τρία αρχεία τύπου CSV. Το ένα αρχείο είναι πίνακας βιβλίων με 271,379 εγγραφές και στήλες το ISBN του βιβλίου, τον τίτλο, τον συγγραφέα, τον εκδότη και την χρονολογία έκδοσης του. Επίσης, υπάρχουν τρεις ακόμα στήλες, η κάθε μία με αντίστοιχο υπερσύνδεσμο που παραπέμπει στις εικόνες του βιβλίου.

Ονόματα στηλών: ("ISBN";"Book-Title";"Book-Author";"Year-Of-Publication";"Publisher";"Image-URL-S";"Image-URL-M";"Image-URL-L")

Το δεύτερο αρχείο απαρτίζεται από έναν πίνακα με 278,859 εγγραφές χρηστών και στήλες τον κωδικό του χρήστη, την τοποθεσία του και σε κάποιες εγγραφές την ηλικία του. Η στήλη τοποθεσία διαθέτει πόλη, περιφέρεια και χώρα. Ωστόσο, σε πολλές εγγραφές τα πρώτα δύο πεδία είτε είναι άγνωστα ή έχουν λανθασμένες εκχωρήσεις. Με αποτέλεσμα, η πιο αξιόπιστη πληροφορία της στήλης να είναι μόνο η χώρα προέλευσης του χρήστη/ αναγνώστη.

Ονόματα στηλών: ("User-ID";"Location";"Age")

Τέλος, υπάρχει το CSV των βαθμολογιών το οποίο περιέχει 1,149,780 εγγραφές σχετικές με την αξιολόγηση των διάφορων βιβλίων από τους χρήστες και τρεις στήλες, το ISBN του βιβλίου που βαθμολογείται, τον κωδικό του χρήστη που βαθμολογεί και την βαθμολογία που έδωσε στο βιβλίο από 0 έως 10. Η τιμή 0 σε μία βαθμολογία σημαίνει πως ο χρήστης έχει διαβάσει/αγοράσει το

βιβλίο αλλά δεν το έχει αξιολογήσει άμεσα. Η άμεση βαθμολογία των βιβλίων παίρνει τιμές από 1 έως 10, υψηλότερες τιμές υποδεικνύουν και υψηλότερη εκτίμηση.

Ονόματα στηλών: ("User-ID";"ISBN";"Book-Rating")

2. Διαδικασία ETL

Προκειμένου να δημιουργήσουμε ένα Data Warehouse όπου θα αποθηκεύονται τα δεδομένα του παραπάνω dataset, προβήκαμε σε μία σειρά ενεργειών ώστε να κάνουμε την εξόρυξη των δεδομένων, να τα καθαρίσουμε από εσφαλμένες ή περιττές εγγραφές και να τα φορτώσουμε στον Microsoft SQL Server στην μορφή που ταιριάζει στο Star Schema που σχεδιάσαμε για την αποθήκη δεδομένων μας. Τα βήματα και οι ενέργειες αυτές συνιστούν την διαδικασία ETL (Extract-Transform-Load-Refresh), την οποία κατασκευάσαμε με την χρήση του εργαλείου MSSQL Integretion Services μέσω της πλατφόρμας Visual Studio 2022 και SQL Server Management Studio 2019.

Το πρώτο βήμα της διαδικασίας ETL είναι το άδειασμα των προσωρινών πινάκων Book_Staging, User_Staging και Rating Staging, στους οποίους εκχωρούμε τα πρωτογενή δεδομένα μας και η μετατροπή του τύπου των στηλών Year-Of-Publication, Age και Book-Rating σε varchar καθώς έτσι καταχωρούνται όλες οι στήλες αρχικά για την αποφυγή σφαλμάτων. Το βήμα αυτό χρειάζεται ώστε να ανανεώνουμε τα δεδομένα του Data Warehouse χωρίς να δημιουργούνται τεχνικά προβλήματα.

Στο δεύτερο βήμα διαβάζουμε όλα τα αρχεία του Book dataset ξεχωριστά χρησιμοποιώντας ως delimiter ";" προκειμένου να αποφύγουμε λάθη κατά την ανάγνωση των αρχείων. Τα λάθη που παρουσιάστηκαν και μας ώθησαν σε αυτήν την επιλογή delimiter οφείλονταν κυρίως σε λάθος εκχωρήσεις ή περίεργους χαρακτήρες και σύμβολα που δεν μπορούσαμε να διαχειριστούμε πριν την καταχώρηση των δεδομένων στην SQL βάση μας. Για τον ίδιο λόγο θεωρούμε πως όλα τα data types των στηλών μας είναι string.

Η ανάγνωση των 3 αρχείων CSV έχει ως προορισμούς 3 staging tables, αντίστοιχα στον SQL Server μας. Ένας προσωρινός πίνακας για τα βιβλία (εξαιρώντας τις στήλες των εικόνων), ένας για τους χρήστες και ένας για τις αξιολογήσεις. Σε αυτό το σημείο, αποφασίσαμε να χρησιμοποιήσουμε OLE DB Destination και όχι SQL Server DB Destination, διότι στην δεύτερη περίπτωση απαιτείται η εκτέλεση του προγράμματος με δικαιώματα διαχειριστή, κάτι που στην συνέχεια μας απέτρεπε από το να χρησιμοποιήσαμε SQL Tasks στο workflow μας.

Καθαρισμός δεδομένων:

Αφού έχουμε εκχωρήσει τα δεδομένα μας από τα αρχεία CSV σε πίνακες στον server μας, προχωράμε στον καθαρισμό τους ώστε να μπορέσουμε στην συνέχεια να τα αξιοποιήσουμε ευκολότερα και αποδοτικότερα στην ανάλυσή μας.

Αρχικά, για κάθε πίνακα από τους τρεις που δημιουργήσαμε αφαιρούμε το σύμβολο " από την αρχή της πρώτης στήλης και από το τέλος της τελευταίας του, αυτή η ανάγκη προκύπτει από το γεγονός ότι επιλέξαμε νωρίτερα ως delimiter το σύμβολο ";".

Στην συνέχεια, για τα δεδομένα των βιβλίων, κρατάμε μόνο τα βιβλία που το ISBN τους αποτελείται από 10 ή 13 ψηφία για να επιβεβαιώσουμε ότι έχουμε μόνο έγκυρους κωδικούς. Για το year πεδίο κρατάμε μόνο όσες εγγραφές αποτελούνται από 4 χαρακτήρες, αφού μετατρέψουμε τον τύπο της στήλης year σε int. Φυσικά, με βάση την τιμή ISBN αφαιρούμε τις διπλο-εγγραφές.

Για τον πίνακα με τους αναγνώστες μετατρέπουμε σε ακέραιο τον τύπο της στήλης age και κρατάμε μόνο τις ηλικίες μεταξύ 6 και 90 που θέσαμε ως ακραίες τιμές για νεότερους και γηραιότερους αναγνώστες. Τις υπόλοιπες τιμές τις κάνουμε null. Τέλος, όσες ηλικίες είναι null εκχωρούμε σε αυτές την μέση ηλικία των αναγνωστών της βάσης μας, ώστε να μπορούμε ακόμα να αξιοποιήσουμε την πληροφορία της προέλευσης των αναγνωστών και κυρίως τις βαθμολογίες που έχουν κάνει. Επιπλέον, για τους σκοπούς της ανάλυσης στρογγυλοποιήσαμε τις ηλικίες στον πρώτο πολλαπλάσιο του 5, δημιουργώντας ηλικιακές ομάδες.

Όσον αφορά στα locations, διορθώνουμε την μορφοποίηση για τις εγγραφές που έχουν null την στήλη age, καθώς η τιμή του location σε αυτήν την περίπτωση είχε αυτή την μορφή "[location]"NULL. Όπως αναφέρθηκε στην περιγραφή του dataset, στο αρχείο Users.csv η στήλη location περιείχε regions, πόλεις ή και χώρες, ωστόσο σε πολλές εγγραφές η μορφοποίηση των δεδομένων είναι λάθος ή μπερδεμένη. Παρατηρήσαμε ότι στην πλειοψηφία των εγγράφων ακόμα και αν δεν αναγράφεται η πόλη, η χώρα βρίσκεται μετά το τελευταίο space. Επομένως, κρατήσαμε αυτό το κομμάτι της στήλης, βγάλαμε τις εγγραφές που αποτελούνταν από ψηφία και όχι χαρακτήρες και όσες είχαν μήκος μικρότερο ή ίσο από δύο χαρακτήρες με εξαίρεση το UK. Στην συνέχεια, από τις εγγραφές που περιέχουν χαρακτήρα σύμβολο κρατούσαμε μόνο αυτές που εμφανίζονται πάνω από 30 φορές. Έπειτα, παρατηρώντας τις πιο δημοφιλείς εγγραφές, όταν κάποια από αυτές επαναλαμβανόταν με άλλη ονομασία, την μετατρέπαμε στην πιο συχνή ονομασία. Για παράδειγμα, το "uk," , "uk" και το "uk." τα μετατρέψαμε σε "United kingdom". Επίσης, επειδή υπήρχαν πολλές εγγραφές που στην θέση της χώρας είχαν αποθηκευτεί πόλεις ή πολιτείες, εάν αυτή η τιμή είχε συχνότητα πάνω από 30 φορές πάλι την μετατρέπαμε στην χώρα που της αντιστοιχεί. Για παράδειγμα, "Washington", "Florida.", "Texas" παίρνουν την τιμή "usa". Επιπλέον, χρησιμοποιώντας regular expressions μετατρέψαμε τις ονομασίες που περιείχαν σύμβολα όπως τόνοι πάλι στην συνηθέστερη ονομασία αυτής της χώρας. Για παράδειγμα, η τιμή México γίνεται Mexico και η τιμή España γίνεται Spain.

Με αυτόν τον τρόπο φτάσαμε στην εμφάνιση 81 διαφορετικών τιμών location από τις οποίες όλες ήταν έγκυρες.

Στο παράρτημα θα παραθέσουμε τα sql scripts που δημιουργήσαμε για τον καθαρισμό.

Δημιουργία και Ανανέωση Διαστάσεων

Αφού τα δεδομένα μας είναι καθαρά προχωράμε στην δημιουργία των διαστάσεων για το star schema μας. Επιλέξαμε να δημιουργήσουμε 4 διαφορετικές διαστάσεις. Έτσι, δημιουργήσαμε στην βάση μας τους πίνακες, Book_Dimension με στήλες book_id (αυξάνεται με κάθε εγγραφή) , isbn,book_title, και year_of_publication που είναι η διάσταση του βιβλίου.

Δημιουργήσαμε έναν πίνακα Author_Dimension με στήλες author_id(αυξάνεται με κάθε εγγραφή) και author_name που αποτελεί την διάσταση του συγγραφέα και έναν πίνακα

Publisher__Dimension με στήλες publisher_id(αυξάνεται με κάθε εγγραφή) και publisher_name για την διάσταση του Εκδότη.

Τέλος, δημιουργήσαμε έναν πίνακα User_Dimension με στήλες user_id(αυξάνεται με κάθε εγγραφή), reader, location και age, που είναι η διάσταση Αναγνώστη (Χρήστης) του σχήματος μας. Χρησιμοποιούμε διαφορετικό id για την διάσταση user από τον κωδικό που πήραμε αρχικά από το αρχείο καθώς μετά τους καθαρισμούς η αλληλουχία των κωδικών έχει αλλάξει και δεν μας διευκολύνει η χρήση του ως ευρετήριο.

Γεμίζουμε τους πίνακες των διαστάσεων Book, Author και Publisher από τον Book Staging πίνακά μας και την διάσταση User από τον User Staging πίνακα.

Δημιουργία και Ανανέωση Fact Table

Μετά τον καθαρισμό δεδομένων και την δημιουργία των διαστάσεων , ενώνουμε με δύο διαδοχικά joins του τρεις καθαρούς(προσωρινούς) staging πίνακες, book, user και rating, έτσι ώστε να διατηρήσουμε μόνο τα βιβλία και του χρήστες για τα οποία έχουμε εγγραφές στον πίνακα των ratings. Αλλά και αντίστροφα για να διατηρήσουμε μόνο τα ratings που αναφέρονται σε βιβλία που διαθέτουμε στην βάση μας. Επίσης, με τον ενιαίο πίνακα γίνεται ευκολότερο το γέμισμα του Fact Table.

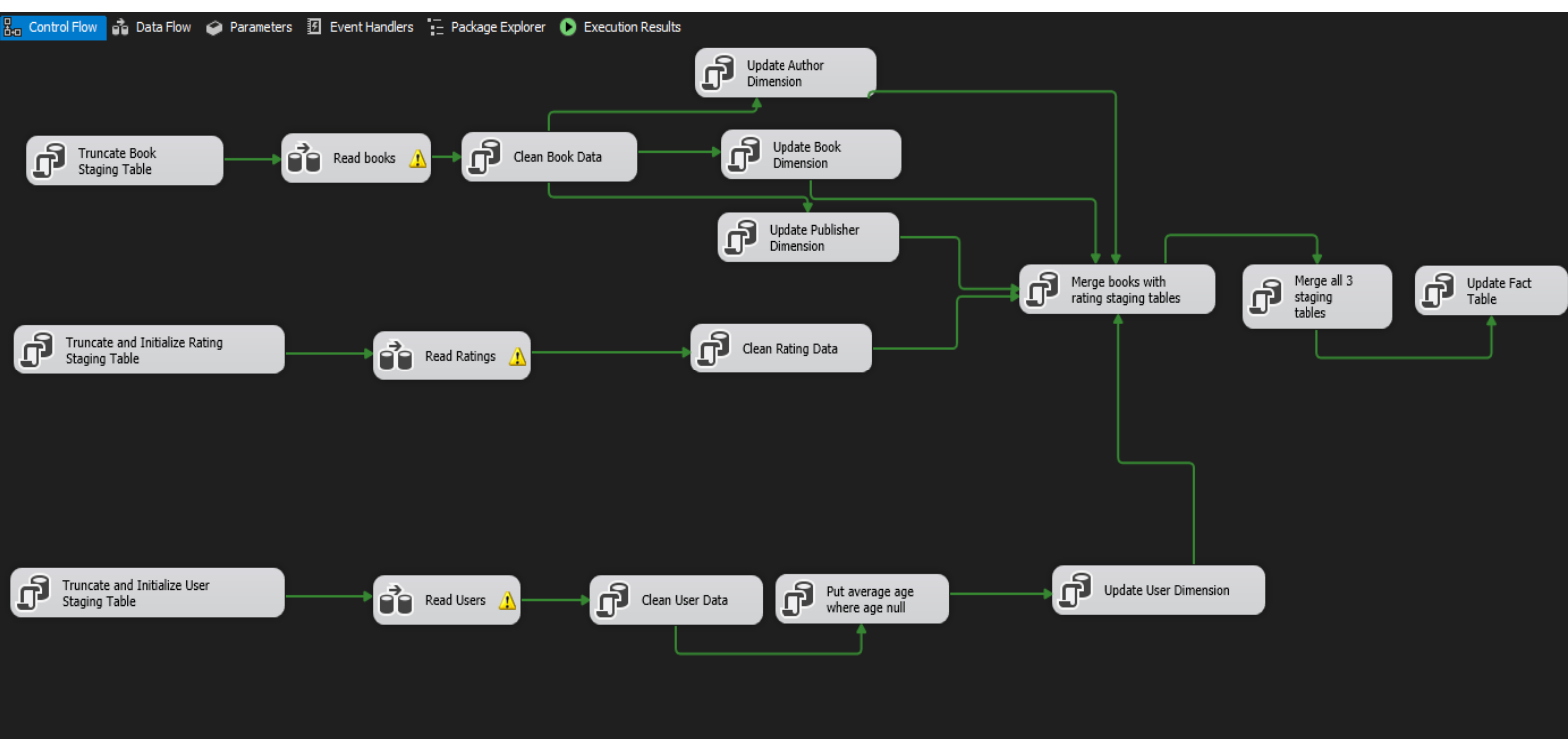
Πριν το γέμισμα του Fact Table, τον δημιουργήσαμε με το SQL Server Management Studio, θέτοντας ως στήλες του τα εξής: Book η οποία έχει ως ξένο κλειδί το book_id από την διάσταση του βιβλίου, Book Author η οποία έχει ως ξένο κλειδί το author_id από την διάσταση του συγγραφέα, Publisher η οποία έχει ως ξένο κλειδί το user_id από την διάσταση του εκδότη, Reader η οποία έχει ως ξένο κλειδί το publisher_id από την διάσταση του χρήστη/αναγνώστη. Τέλος, έχει την στήλη Rating Score η οποία έχει ίδιο τύπο με την στήλη Book_Rating από τον staging πίνακα Ratings, αλλά δεν το ορίζουμε ως ξένο κλειδί.

Αφού ορίσαμε τους τύπους κι τις σχέσεις των στηλών του fact table, προχωρούμε στο γέμισμά του. Το γέμισμα επιτυγχάνεται με την αντιστοίχιση των τιμών του merged staging πίνακα και των πρωτευόντων κλειδιών από τις διαστάσεις μας. Έτσι, τελικά ο fact table μας αποτελείται από 732,242 εγγραφές, από 4 στήλες με ξένα κλειδιά των διαστάσεων και από μια στήλη που δείχνει την μετρική του Rating Score που αντιστοιχεί σε αυτή την εγγραφή.

Η τελική μορφή του Fact Table για τα Ratings είναι ως εξής:

	Book	Book_Author	Publisher	Reader	Rating_Score
1	72921	615	3431	139750	5
2	11141	615	12293	75603	0
3	6171	615	13590	76184	9
4	56907	616	10151	25191	0
5	56919	616	10151	111579	0
6	56907	616	10151	104972	0
7	57069	616	10151	65449	0
8	56960	616	10151	33109	0
9	56907	616	10151	70403	0
10	57046	616	10151	33109	0
11	57101	616	10151	33109	0
12	57046	616	10151	83817	3
13	57046	616	10151	16851	0
14	57069	616	10151	103607	0

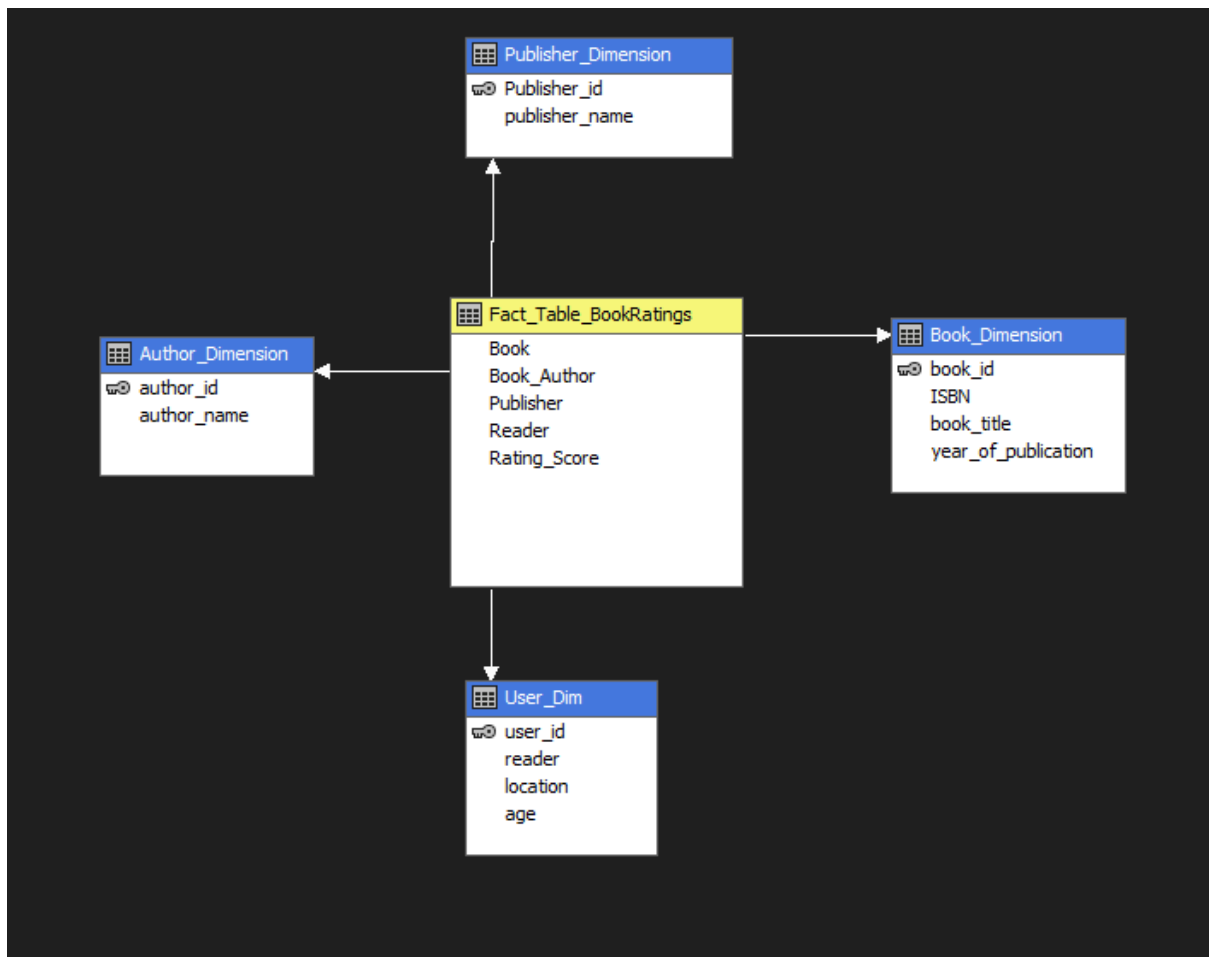
Στην παρακάτω εικόνα φαίνεται το ETL workflow που δημιουργήσαμε.



3. Star Schema

Στην εικόνα φαίνεται το Star Schema που δημιουργήσαμε. Αυτό αποτελείται από τον fact table με τις βαθμολογίες των βιβλίων. Κάθε βαθμολογία βιβλίου χαρακτηρίζεται από το βιβλίο που βαθμολογείται, τον συγγραφέα του βιβλίου, τον εκδότη του βιβλίου, τον αναγνώστη που έκανε την βαθμολογία και το σκορ της βαθμολογίας, το οποίο είναι μετρική με τιμές της κλίμακας από 0 έως 10. Τα πρώτα 4 πεδία που χαρακτηρίζουν μία βαθμολόγηση βιβλίου είναι διαστάσεις.

Η διάσταση του βιβλίου αποτελείται από ένα ακέραιο book id που συμμετέχει ως ξένο κλειδί στο Fact Table, το ISBN, τον τίτλο και τον χρόνο έκδοσης του βιβλίου. Η διάσταση Book Author αποτελείται από ένα ακέραιο author id, ξένο κλειδί στον fact table και το όνομα του author. Παρόμοια, η διάσταση Publisher αποτελείται από έναν ακέραιο publisher id και το όνομα του publisher. Τέλος, η διάσταση User αποτελείται από τον ακέραιο user_id που δανείζει ως ξένο κλειδί στο fact table, τον κωδικό του αναγνώστη όπως ήταν αρχειοθετημένος στα αρχικά δεδομένα, την χώρα προέλευσής του και την ηλικία του.



4. Κύβος

Για την δημιουργία και το process του κύβου για το data warehouse που δημιουργήσαμε εργαστήκαμε πάλι στο Visual Studio με το πακέτο MSSQL Analysis Services. Αρχικά, ορίσαμε ως Data Source την βάση δεδομένων στον server μας, όπου υπάρχουν οι πίνακες των διαστάσεων του σχήματος και ο Fact Table με τα Book Ratings.

Στην συνέχεια, δημιουργήσαμε ένα Data Source View φορτώνοντας τον πίνακα του fact table και τους σχετιζόμενους με αυτόν πίνακες, δηλαδή τους πίνακες των διαστάσεων. Έπειτα, προχωρήσαμε στην δημιουργία ενός κύβου από το Data Source View, ορίζοντας το Fact Table Book Ratings ως το measure group. Ως μετρικές του κύβου διατηρήσαμε το Rating Score και την μετρική Count που δημιουργήθηκε αυτόματα. Ως διαστάσεις ορίσαμε τους πίνακες Book, Author, Publisher και User.

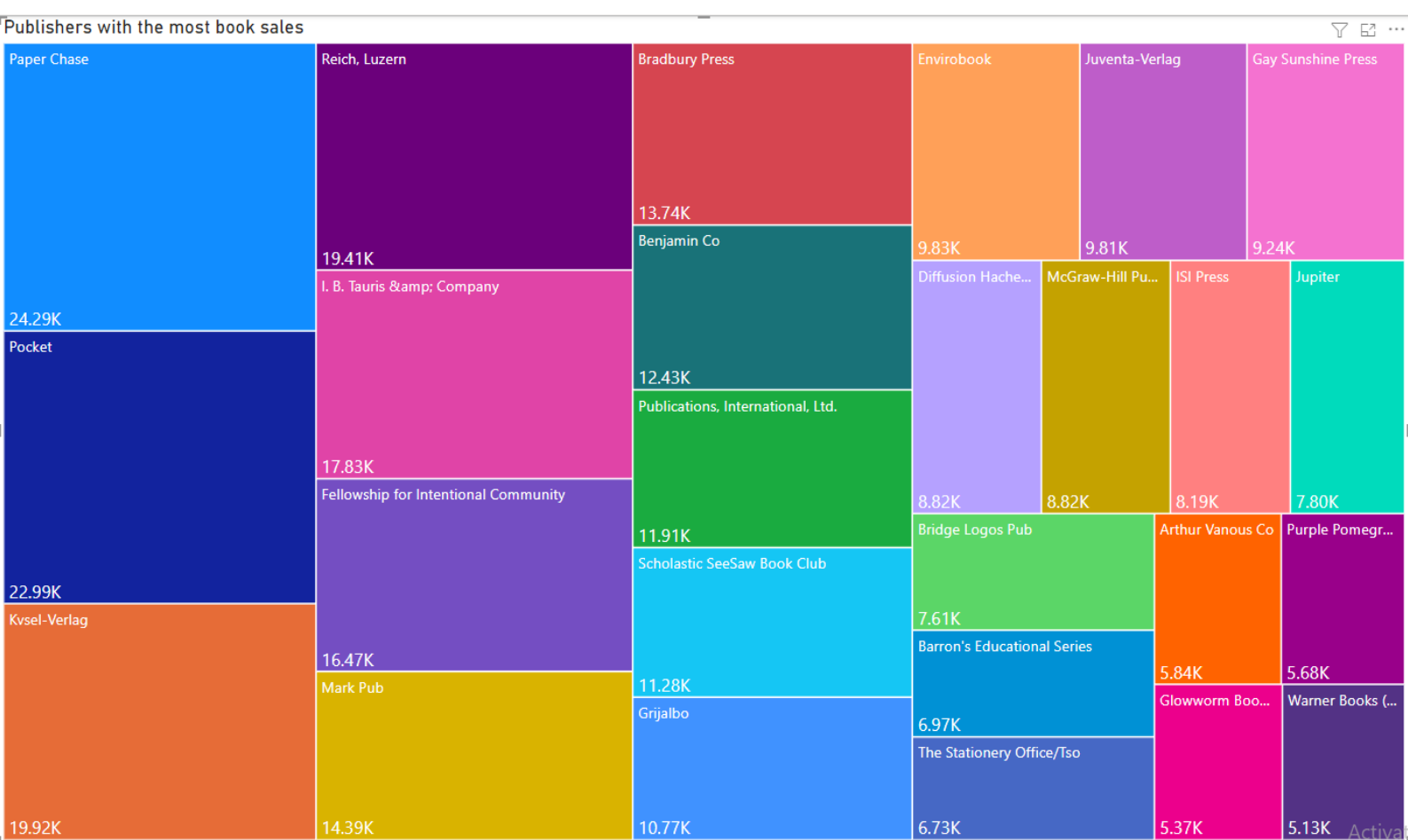
Στην συνέχεια, φροντίσαμε για κάθε διάσταση να φέρουμε στον κύβο εκτός από το πρωτεύον κλειδί της και τα υπόλοιπα γνωρίσματα της προκειμένου να μπορούμε να ερμηνεύσουμε τα αποτελέσματα των υπολογισμών του κύβου και τον κάναμε process να δούμε πως όλα είναι σωστά. Σε αυτό το σημείο μας εμφανίστηκαν κάποια σφάλματα τα οποία δεν είχαμε παρατηρήσει κατά την ETL διαδικασία και είχαμε την δυνατότητα να γυρίσουμε πίσω και να κάνουμε διορθώσεις στα δεδομένα μας πριν την χρήση του κύβου.

Μετά τις διορθώσεις, αφού είδαμε πως ο κύβος γίνεται επιτυχώς process προχωρήσαμε στην δημιουργία νέας μετρικής. Ένα χρήσιμο measure που θέλουμε να γνωρίζουμε για τα δεδομένα που μας απασχολούν ήταν ο μέσος όρος των book ratings, το οποίο δημιουργήσαμε μέσω της ενότητας Calculations κάνοντας την πράξη $[Measures].[Rating\ Score]/[Measures].[Fact\ Table\ Book\ Ratings\ Count]$ και έπειτα ξανά κάναμε process τον κύβο μας προκειμένου να τον αξιοποιήσουμε στην δημιουργία visuals.

5. Dashboards με Power BI

Αφού κάναμε deploy τον κύβο της βάσης μας και δημιουργήσαμε επιπλέον μετρικές που μας ενδιαφέρουν, χρησιμοποιήσαμε το εργαλείο Power BI και αφού το συνδέσαμε με το Analysis Services Model μας δημιουργήσαμε μερικά dashboards με κάποιες ενδιαφέρουσες αναπαραστάσεις των δεδομένων μας.

Στο πρώτο διάγραμμα μπορούμε να δούμε τους εκδότες που είχαν τις περισσότερες πωλήσεις βιβλίων. Συγκεκριμένα, το φίλτρο που χρησιμοποιήσαμε ήταν τα βιβλία τους να έχουν συμπεριληφθεί σε πάνω από 2000 ratings. Το μέγεθος του πλαισίου του κάθε εκδότη είναι ανάλογο του μεγέθους πωλήσεων. Βλέπουμε πως οι κορυφαίοι πέντε εκδότες με το μεγαλύτερο μερίδιο αγοράς στην Amazon ήταν οι Paper Chase, Pocket, Kvsel-Verlag, Reich, Luzen και I.B. Tauris & Company.



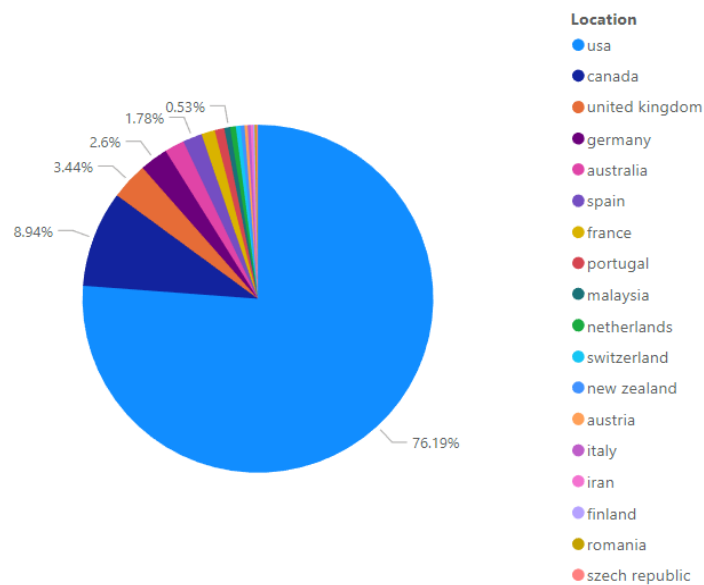
Στο δεύτερο visual χρησιμοποιούμε το ίδιο διάγραμμα για να δείξουμε τους συγγραφείς με τις κορυφαίες πωλήσεις βιβλίων, ενώ βλέπουμε συγκεκριμένα τους συγγραφείς που έχουν πωλήσει πάνω από 2000 βιβλία. Οι τρεις κορυφαίοι συγγραφείς σε πωλήσεις είναι ο Hanley G, William Mason και ο Marcus Laux.



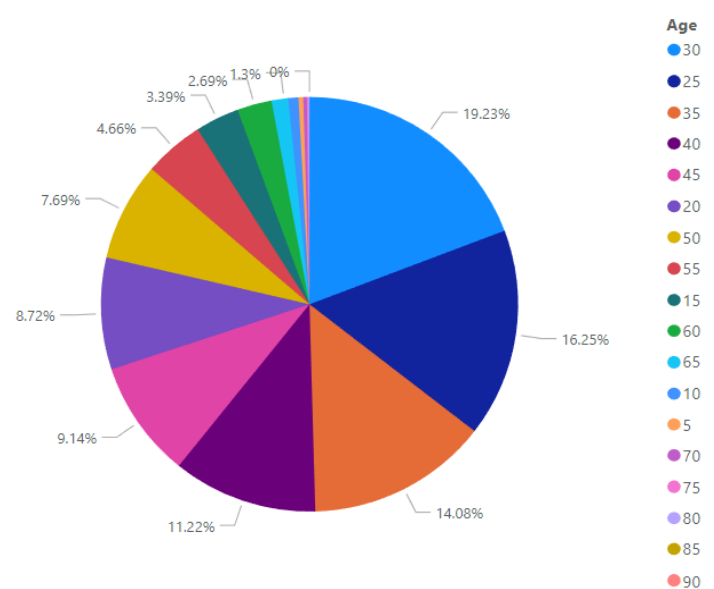
Στην συνέχεια βλέπουμε δύο pie charts με τις αγορές βιβλίων πρώτα ανά χώρα προέλευσης και μετά στο δεύτερο ανά ηλικία. Για το πρώτο chart κρατήσαμε τις χώρες από τις οποίες οι πελάτες έχουν διαβάσει πάνω από 1000 βιβλία αθροιστικά ενώ για το δεύτερο διάγραμμα κρατήσαμε τις ηλικίες που έχουν διαβάσει πάνω από 5000 βιβλία αθροιστικά. Παρατηρούμε πως το συντριπτικά μεγαλύτερο μέρος των αγορών είναι από τις Ηνωμένες Πολιτείες κάτι που δικαιολογείται από το γεγονός πως το Amazon Bookstore είναι πολύ δημοφιλές στις ΗΠΑ σε σύγκριση με άλλες χώρες. Στην συνέχεια, ακολουθεί ο Καναδάς, το Ηνωμένο Βασίλειο, η Αυστραλία και κάποιες ευρωπαϊκές χώρες. Ένα εύλογο και χρήσιμο συμπέρασμα είναι πως το αγοραστικό κοινό έχει την αγγλική ως μητρική γλώσσα.

Όσον αφορά τις ηλικιακές ομάδες βλέπουμε πως οι περισσότερες αγορές βιβλίων έχουν πραγματοποιηθεί από πελάτες που κυμαίνονται σε ηλικίες μεταξύ 25 με 45 χρονών. Οι νεότεροι σε ηλικία κάτω των 25 έχουν κάνει λιγότερες σχετικά αγορές κάτι που υποδεικνύει μία αποστροφή προς την ανάγνωση των νεότερων γενεών. Από την ηλικία 60 και μετά οι αγορές όλο και μειώνονται, κάτι που μπορεί να δικαιολογηθεί από την έλλειψη τεχνολογικών γνώσεων περισσότερο, καθώς μεγαλύτερα ηλικιακές ομάδες δεν χρησιμοποιούν συχνά το διαδίκτυο προκειμένου να κάνουν ηλεκτρονική αγορά από την Amazon.

Books bought by county

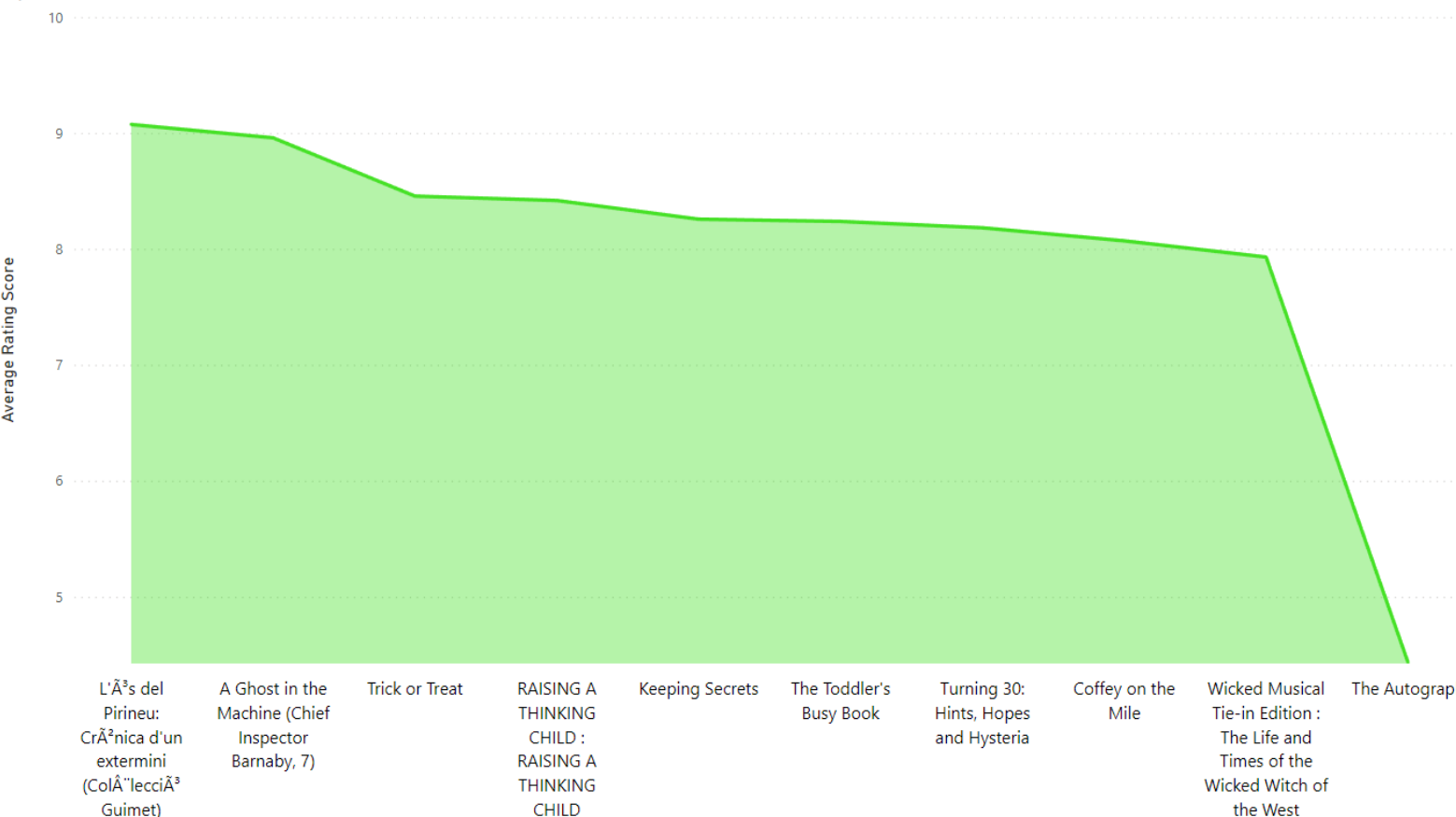


Books bought by Age

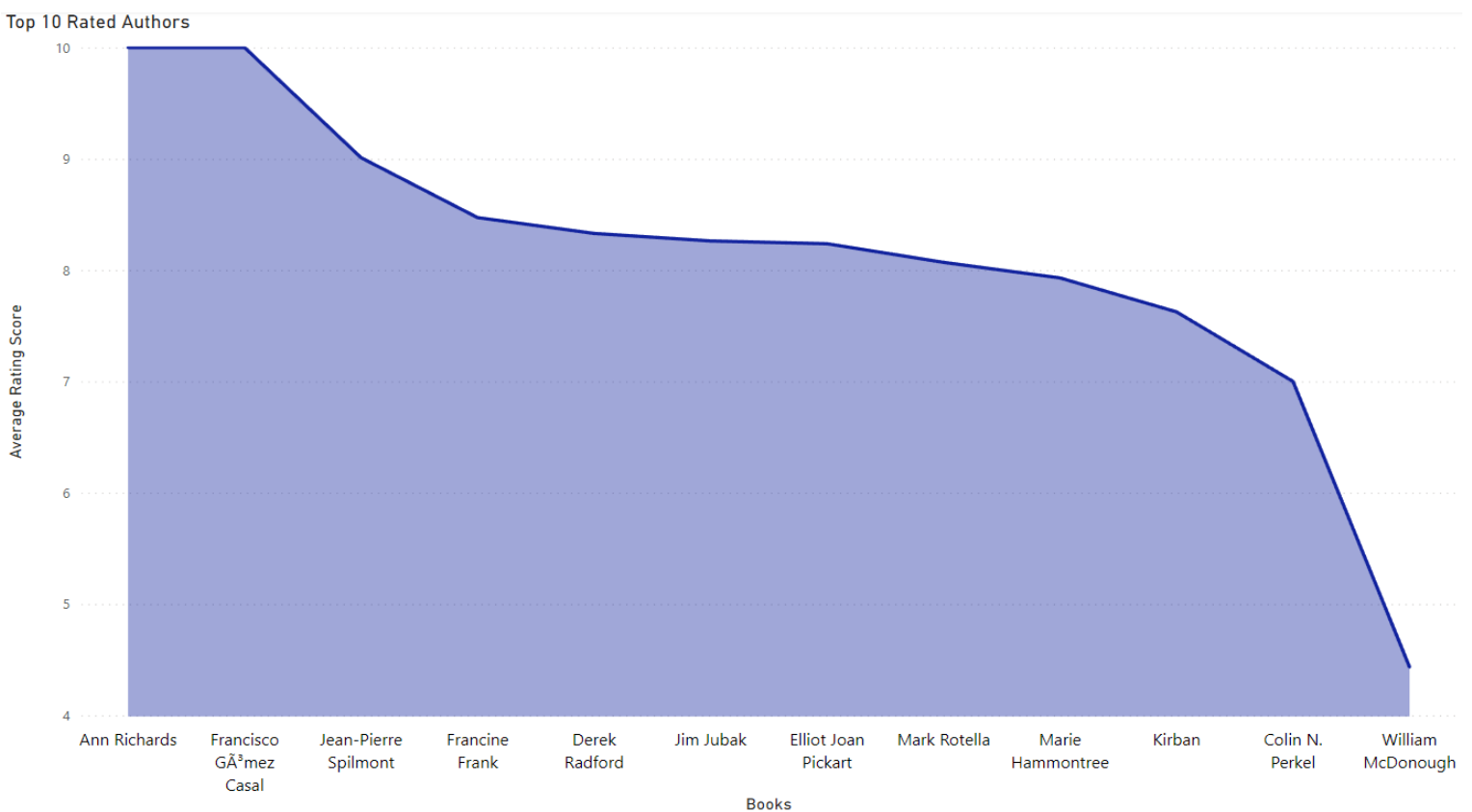


Για τα επόμενα δύο διαγράμματα μας ενδιέφερε η αξιολόγηση των βιβλίων και όχι η αγορά τους, επομένως εξαιρέσαμε τις εγγραφές για τις οποίες το Rating Score ισούται με 0. Στο διάγραμμα που ακολουθεί φαίνονται τα 10 κορυφαία σε αθροιστικές βαθμολογίες βιβλία και ο μέσος όρος. Έχει σημασία η διαλογή των βιβλίων με βάση την αθροιστική βαθμολογία καθώς λαμβάνεται υπόψιν και ο αριθμός των πωλήσεων σε συνδυασμό με τις βαθμολογίες.

Top 10 Rated books

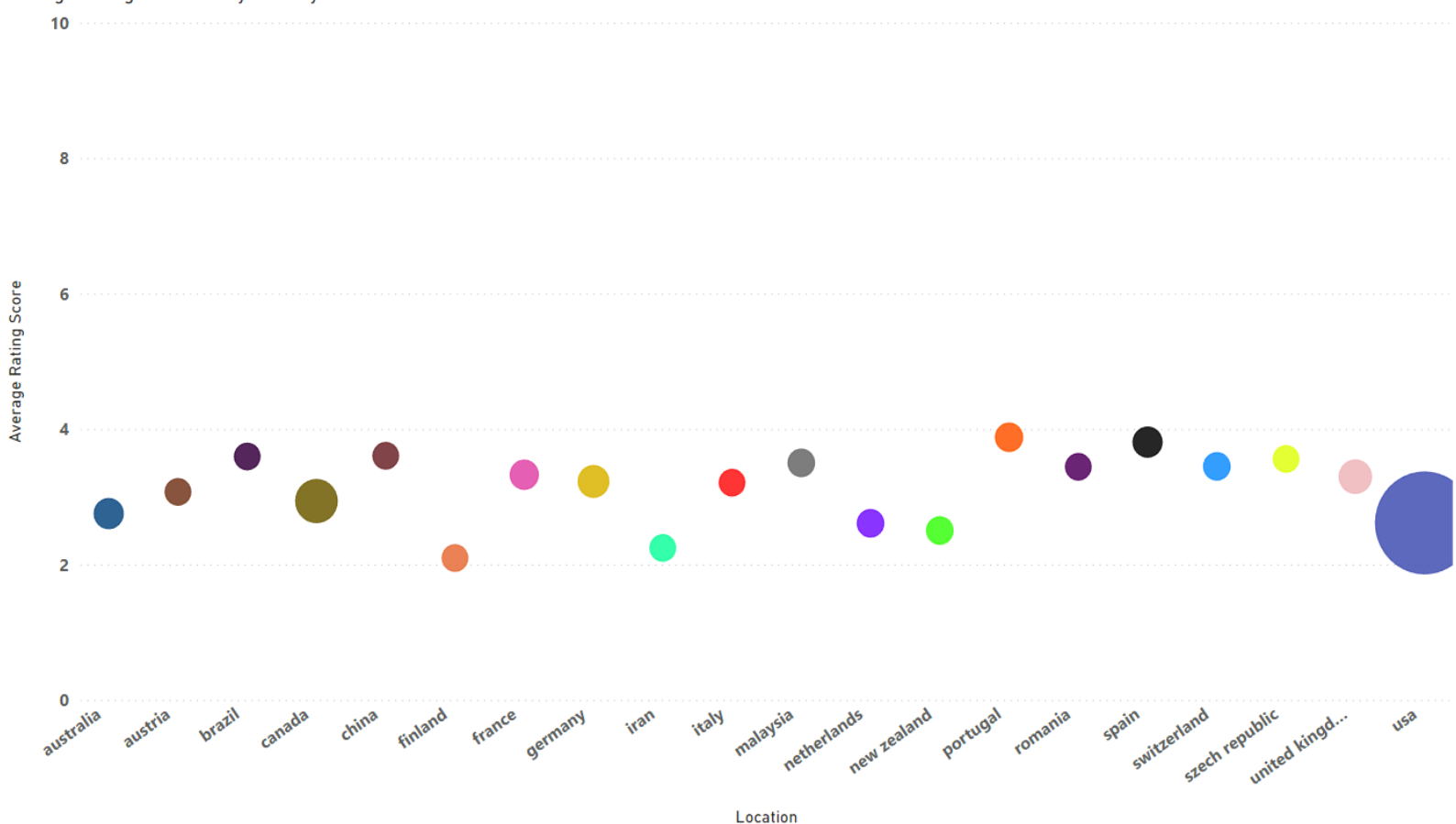


Αντίστοιχα παρουσιάζουμε τους 10 κορυφαίους συγγραφείς και τον μέσο όρο βαθμολογιών τους δίνοντας, πάλι, έμφαση στην αθροιστική βαθμολογία των βιβλίων τους. Αν συγκρίνουμε τα ονόματα σε αυτό το διάγραμμα με εκείνα των συγγραφέων με τις κορυφαίες πωλήσεις δεν υπάρχουν πολλά κοινά αυτό σημαίνει είτε πως οι συγγραφείς με τις περισσότερες πωλήσεις δεν έχουν καλές κριτικές ή ότι πολλοί από αυτούς που αγόρασαν το βιβλίο τους δεν θέλησαν να γράψουν κριτική πάνω σε αυτό.



Το τελευταίο διάγραμμα είναι ένα scatter plot με τις 20 χώρες που αγόρασαν πιο συχνά βιβλία και τον μέσο όρο βαθμολογίας που οι πολίτες τους βάζουν στα βιβλία. Το μέγεθος του σημείου υποδεικνύει το πλήθος αγορών που έχει γίνει από κάθε χώρα, με τις ΗΠΑ να πρωταγωνιστούν όπως είδαμε και στο pie chart προηγουμένως. Η χαμηλή θέση των χωρών στον άξονα του y οφείλεται και στις αγορές βιβλίων χωρίς άμεση αξιολόγηση. Επομένως, μπορούμε να ερμηνεύσουμε την αυστηρότητα των χωρών μόνο εν συνάρτηση του μεγέθους τους και σε σύγκριση με τις υπόλοιπες χώρες.

Average Rating and Sales by Country



6. Μοντέλα Data Mining

Προκειμένου να εξαχθεί χρήσιμη πληροφόρηση από τα δεδομένα δημιουργήθηκαν και αξιοποιήθηκαν τρία μοντέλα data mining. Έτσι, κρίνεται σκόπιμη μία ενδελεχής αναφορά στα μοντέλα καθώς και στα αποτελέσματα τους.

Πρώτο Μοντέλο:

Όσον αφορά στο πρώτο μοντέλο αξιοποιήθηκε ο αλγόριθμος apriori ώστε να πραγματοποιηθεί ανάλυση κανόνων συσχετίσεων. Μέσω της ανάλυσης κανόνων συσχετίσεων εξάγεται πληροφόρηση σχετικά με το ποια προϊόντα διαβάστηκαν συνδυαστικά και καθίστανται ικανή η πρόταση νέων βιβλίων στους χρήστες λαμβάνοντας υπόψιν προηγούμενες αγορές τους. Τα αποτελέσματα της ανάλυσης συσχετίσεων είναι πολύ χρήσιμα καθώς μέσω της πληροφόρησης σχετικά με το ποια βιβλία διαβάζονται συνδυαστικά, οι διασταυρούμενες πωλήσεις μπορούν να

βελτιωθούν, η διάταξη των καταστημάτων μπορεί να αλλάξει έτσι ώστε οι πωλήσεις να αυξάνονται, οι εκπτώσεις και οι προσφορές «1+1» μπορούν να σχεδιαστούν έτσι ώστε να είναι πολύ προσοδοφόρες και οι προωθητικές δραστηριότητες μπορούν να γίνουν πιο στοχευμένες και αποτελεσματικές.

Σχετικά με τον αλγόριθμο Apriori, αυτός χρησιμοποιήθηκε για την πρώτη φάση των κανόνων συσχέτισης και είναι ο πιο δημοφιλής και κλασικός αλγόριθμος. Ο αλγόριθμος αυτός υλοποιείται πολύ εύκολα και εκμεταλλεύεται την ιδιότητα των συχνών itemsets. Πιο συγκεκριμένα, αναζητούνται ισχυρές σχέσεις μεταξύ των προϊόντων και στην συγκεκριμένη περίπτωση μεταξύ τίτλων βιβλίων. Η σπουδαιότητα ενός κανόνα συσχέτισης μπορεί να προσδιοριστεί από 3 παραμέτρους που χρησιμοποιούνται για τον προσδιορισμό της ισχύος του αλγορίθμου.

1. Υποστήριξη (support): είναι το ποσοστό των ομάδων/συνόλων που περιέχουν όλα τα στοιχεία που αναφέρονται σε αυτόν τον κανόνα συσχέτισης ή αλλιώς η πιθανότητα να εμφανιστούν συνδυαστούν αυτά τα προϊόντα
2. Εμπιστοσύνη (confidence): είναι η αναλογία του πλήθους των συνολικών δοσοληψιών που περιέχουν το X και Y ως προς το πλήθος των συνολικών δοσοληψιών που περιέχουν το X. Όπου X εννοούμε την κεφαλή του κανόνα(*antecedent*) και όπου Y εννοούμε τα επακόλουθα(*consequent*). Με άλλα λόγια, η εμπιστοσύνη είναι η εξαρτημένη πιθανότητα.
3. Lift: Είναι η πιθανότητα να εμφανιστούν όλα τα στοιχεία μαζί διαιρούμενη με το γινόμενο του προγενέστερου και του επακόλουθου που εμφανίζονται σαν να είναι ανεξάρτητα το ένα από το άλλο. Το τελευταίο δεν θα μας απασχολήσει τόσο στην τωρινή ανάλυση.

Με δεδομένα αυτά, προχωρήσαμε στην υλοποίηση του αλγορίθμου. Αναφορικά με την υλοποίηση όλες οι διαδικασίες καθώς και το μοντέλο εκτελέστηκαν στην γλώσσα python. Αξιοποιήθηκε η βιβλιοθήκη mlxtend ενώ προκειμένου να υλοποιηθεί το μοντέλο απαιτούνταν απαλοιφή κάποιων δεδομένων, αρκετές πράξεις καθώς και σύνθετοι υπολογισμοί ώστε τα δεδομένα να αποκτήσουν μία μορφή συμβατή με τη μέθοδο apriori της βιβλιοθήκης. Πιο αναλυτικά, σχετικά με τον καθαρισμό των δεδομένων επιλέχθηκε να απαλειφθούν βιβλία που δεν αγοράζονται τόσο συχνά ή ορθότερα έχουν αγοραστεί από πολύ λίγα άτομα. Επίσης, επιλέξαμε να αφαιρέσουμε χρήστες που διάβασαν λιγότερα από τρία βιβλία. Αυτοί οι χρήστες δεν συνεισφέρουν στην ανίχνευση κοινών συσχετίσεων. Τέλος, όσον αφορά στην μορφή έπρεπε να προκύψει ένα πίνακας με γραμμές τους χρήστες και στήλες τους διάφορους τίτλους βιβλίων. Στο πίνακα αυτό για κάθε γραμμή, δηλαδή για κάθε χρήστη, υπάρχει η τιμή True στη στήλη του βιβλίου που έχει διαβάσει και η τιμή False σε όσα βιβλία δεν έχει διαβάσει. Έτσι, προκύπτει ένας πίνακας όπου για κάθε χρήστη υπάρχει True στις στήλες των βιβλίων που έχει διαβάσει και False σε άλλη περίπτωση. Ο πίνακας χρησιμοποιείται προκειμένου να εκτελεστεί ο αλγόριθμος apriori.

Τελευταίο και εξαιρετικά σημαντικό είναι ότι για το μοντέλο της συσχέτισης κανόνων δεν κάναμε απαλοιφή των τιμών με μηδενική αξιολόγηση. Το μηδέν παραπέμπει ότι ένα βιβλίο έχει αγοραστεί από το χρήστη ωστόσο δεν έχει προχωρήσει στην αξιολόγηση του. Στο συγκεκριμένο μοντέλο όπου ερευνάμε συνδυασμούς βιβλίων που αγοράστηκαν και διαβάστηκαν μας ενδιαφέρει αυτή η πληροφορία ακόμα και αν δεν συνοδεύεται από αξιολόγηση.

Κάποια ενδεικτικά στιγμιότυπα:

- Από προεπιλογή, το `apriori` επιστρέφει τους δείκτες των στηλών των στοιχείων, οι οποίοι μπορεί να είναι χρήσιμοι σε επόμενες λειτουργίες, όπως η εξόρυξη κανόνων συσχέτισης. Για καλύτερη αναγνωσιμότητα, μπορούμε να ορίσουμε `use_colnames=True` για να μετατρέψουμε αυτές τις ακέραιες τιμές στα αντίστοιχα ονόματα στοιχείων

```
df = apriori(association_df, min_support=0.01, use_colnames=True, verbose=1, max_len= None, low_memory=False )
df
```

[144] ✓ 1m 2.7s Python

... Processing 204 combinations | Sampling itemset size 6543

	support	itemsets
0	0.027704	(60929)
1	0.042146	(107643)
2	0.028588	(107486)
3	0.038609	(107666)
4	0.042146	(107695)
...
3968	0.010610	(36578, 36067, 130572, 36277, 36120)
3969	0.010610	(36578, 36422, 130572, 36277, 36120)
3970	0.010315	(36578, 36067, 36422, 130572, 36120)
3971	0.011789	(36578, 36067, 36422, 36277, 36120)
3972	0.010021	(36067, 36422, 36721, 36277, 36120)

3973 rows x 2 columns

- Επιλέγω τους κορυφαίους 10 συνδυασμούς

```
df_ar = association_rules(df, metric = "confidence")
final_results = df_ar.sort values(by='confidence').iloc[:10][['antecedents', 'consequents', 'support', 'confidence', 'lift']]
```

[105] ✓ 0.4s

- Τα τελικά αποτελέσματα με το `book_id`, απομένει ένα τελευταίο βήμα να εμφανίσω τα `book_titles`

```
final_results
```

[106] ✓ 0.4s

	antecedents	consequents	support	confidence	lift
123	(82995, 82757, 82974)	(123557)	0.017683	0.800000	22.810084
178	(36120, 36067, 36277)	(130572)	0.016799	0.802811	16.11397
184	(130572, 36422)	(36120, 36277)	0.015620	0.803030	29.616107
193	(130572, 36422)	(36120, 36067)	0.015620	0.803030	30.24242
11	(82996, 82975)	(82758)	0.015620	0.803030	13.555631
51	(95730, 95630)	(117171)	0.012084	0.801922	24.797326
276	(85920, 85441, 85537)	(85454)	0.010905	0.804348	9.158229
278	(85920, 85561, 85876)	(85441)	0.010905	0.804348	8.719336
267	(85920, 85561, 86087)	(85454)	0.010905	0.804348	9.158229
13	(83596, 82974)	(82757)	0.017094	0.805556	21.353516

- Για να εμφανίσω τους τίτλους βιβλίων χρειάζομαι το πίνακα `dimension book_title_dim`

```
con.append(rr['book_title_label'])
continue
print(asc, con)
print("-----> " + 'support ' + str(row['support']) + " & confidence " + str(row['confidence']) + ' ' )
```

[220] ✓ 1.8s Python Python

... ['Harry Potter and the Goblet of Fire (Book 4)', 'Harry Potter and the Chamber of Secrets (Book 2)', 'Harry Potter and the Prisoner of Azkaban (Book 3)'] ['Harry Potter and the Sorcerer's Stone (Book 1)']

-----> support 0.0176834659932803 & confidence 0.8

['Four To Score (A Stephanie Plum Novel)', 'Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)', 'High Five (A Stephanie Plum Novel)'] ['Two for the Dough']

-----> support 0.016799292661361626 & confidence 0.8028169014084506

['Two for the Dough', 'Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)'] ['Four To Score (A Stephanie Plum Novel)', 'High Five (A Stephanie Plum Novel)']

-----> support 0.015620394930739759 & confidence 0.803030303030303

['Two for the Dough', 'Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)'] ['Four To Score (A Stephanie Plum Novel)', 'Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)']

-----> support 0.015620394930739759 & confidence 0.803030303030303

['Harry Potter and the Goblet of Fire (Book 4)', 'Harry Potter and the Prisoner of Azkaban (Book 3)'] ['Harry Potter and the Chamber of Secrets (Book 2)']

-----> support 0.015620394930739759 & confidence 0.803030303030303

['K Is for Killer (Kinsey Millhone Mysteries (Paperback))', 'G Is for Gumshoe (Kinsey Millhone Mysteries (Paperback))'] ['F Is for Fugitive (Kinsey Millhone Mysteries (Paperback))']

-----> support 0.012083701738874153 & confidence 0.803921568627451

['The Rainmaker', 'The Firm', 'The Client'] ['A Time to Kill']

-----> support 0.010904804008252285 & confidence 0.8043478260869565

['The Rainmaker', 'The Pelican Brief', 'The Chamber'] ['The Firm']

-----> support 0.010904804008252285 & confidence 0.8043478260869565

['The Rainmaker', 'The Pelican Brief', 'The Partner'] ['A Time to Kill']

-----> support 0.010904804008252285 & confidence 0.8043478260869565

['Harry Potter and the Order of the Phoenix (Book 5)', 'Harry Potter and the Prisoner of Azkaban (Book 3)'] ['Harry Potter and the Chamber of Secrets (Book 2)']

-----> support 0.017094017094017096 & confidence 0.8055555555555555

Στο παράρτημα θα παραθέσουμε τα αρχεία `python`.

Τα τελικά αποτελέσματα σε μορφή πίνακα:

Antecedents	Consequents	Support	Confidence
1. Seven Up (A Stephanie Plum Novel)			
2. Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)			
3. Two for the Dough			
4. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)	Four To Score (A Stephanie Plum Novel)	0.0103	1.0000
1. A Is for Alibi (Kinsey Millhone Mysteries (Paperback))			
2. F Is for Fugitive (Kinsey Millhone Mysteries (Paperback))	B Is for Burglar (Kinsey Millhone Mysteries (Paperback))	0.0103	1.0000
3. C Is for Corpse (Kinsey Millhone Mysteries (Paperback))			
1. Harry Potter and the Goblet of Fire (Book 4)			
2. Harry Potter and the Sorcerer's Stone (Book 1)			
3. Harry Potter and the Prisoner of Azkaban (Book 3)	Harry Potter and the Chamber of Secrets (Book 2)	0.0177	0.9836
1. Harry Potter and the Order of the Phoenix (Book 5)			
2. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)			
3. Harry Potter and the Prisoner of Azkaban (Book 3)	Harry Potter and the Chamber of Secrets (Book 2)	0.0139	0.9792
1. Harry Potter and the Goblet of Fire (Book 4)			
2. Harry Potter and the Order of the Phoenix (Book 5)			
3. Harry Potter and the Sorcerer's Stone (Book 1)	Harry Potter and the Chamber of Secrets (Book 2)	0.0121	0.9762
1. Seven Up (A Stephanie Plum Novel)			
2. Two for the Dough			
3. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)	Four To Score (A Stephanie Plum Novel)	0.0115	0.9750
1. Harry Potter and the Goblet of Fire (Book 4)			
2. Harry Potter and the Order of the Phoenix (Book 5)			
3. Harry Potter and the Sorcerer's Stone (Book 1)			
4. Harry Potter and the Prisoner of Azkaban (Book 3)	Harry Potter and the Chamber of Secrets (Book 2)	0.0115	0.9750
1. Seven Up (A Stephanie Plum Novel)			
2. Two for the Dough			
3. High Five (A Stephanie Plum Novel)			
4. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)	Four To Score (A Stephanie Plum Novel)	0.0115	0.9750
1. Four To Score (A Stephanie Plum Novel)			
2. Hard Eight : A Stephanie Plum Novel (A Stephanie Plum Novel)			
3. Seven Up (A Stephanie Plum Novel)	High Five (A Stephanie Plum Novel)	0.0103	0.9722
1. A Is for Alibi (Kinsey Millhone Mysteries (Paperback))			
2. F Is for Fugitive (Kinsey Millhone Mysteries (Paperback))	B Is for Burglar (Kinsey Millhone Mysteries (Paperback))	0.0103	0.9722
3. G Is for Gumshoe (Kinsey Millhone Mysteries (Paperback))			

Ανάλυση για το τελευταίο κελί του παραπάνω πίνακα:

Παρατηρούμε ότι όποιος αγοράζει τα βιβλία:

1. *A Is for Alibi (Kinsey Millhone Mysteries (Paperback))*
2. *F Is for Fugitive (Kinsey Millhone Mysteries (Paperback))*
3. *G Is for Gumshoe (Kinsey Millhone Mysteries (Paperback))*

Τότε αγοράζει και το βιβλίο *B Is for Burglar (Kinsey Millhone Mysteries (Paperback))* όπου και βγάζει απόλυτο νόημα.

Πιο αναλυτικά

1. Υποστήριξη 0.010315355: Η πιθανότητα να αγοραστούν συνδυαστικά τα βιβλία
2. Εμπιστοσύνη 0.97222222: Από το σύνολο όλων των αγορών που περιέχουν τα βιβλία *A Is for Alibi (Kinsey Millhone Mysteries (Paperback))* , *F Is for Fugitive (Kinsey Millhone Mysteries (Paperback))* , *G Is for Gumshoe (Kinsey Millhone Mysteries (Paperback))* , το 97.2% των αγορών θα περιέχουν το βιβλίο *B Is for Burglar (Kinsey Millhone Mysteries (Paperback))*

Όμοια ανάλυση γίνεται και για τις υπόλοιπες συσχετίσεις.

Έτσι, με δεδομένες τις συνολικές αγορές των ατόμων σε βιβλία εξάγονται συμπεράσματα για τους συνδυασμούς των βιβλίων. Ένα καθόλου παράξενο στοιχείο που προέκυψε από την ανάλυση και παρουσιάζεται στο πίνακα είναι ότι κάποιος χρήστης που έχει αγοράσει το πρώτο, το τρίτο και το τέταρτο βιβλίο της σειράς του Harry Potter τότε κατά 98% κάποια στιγμή θα αγοράσει και τον δεύτερο τόμο της σειράς Harry Potter. Με άλλα λόγια, η πλειονότητα των χρηστών τείνει να αγοράζει αυτούς τους συνδυασμούς.

(Τα αποτελέσματα δεν αφορούν μία μεμονωμένη συναλλαγή αλλά τις συνολικές αγορές)

Δεύτερο Μοντέλο:

Όσον αφορά στο δεύτερο μοντέλο ακολουθήθηκε παρόμοια διαδικασία, με τη διαφορά ότι θέλουμε να βρούμε συνδυασμούς «ταιριαστών»/αρεστών βιβλίων. Με άλλα λόγια κάποιος μπορεί να προβεί σε μία αγορά που εκ των υστέρων θα θεωρηθεί λαθεμένη. Έτσι, θέλουμε να αποκτήσουμε γνώση που αφορά επιτυχημένους συνδυασμούς βιβλίων. Η πληροφόρηση σχετικά με τις επιτυχημένες ομάδες βιβλίων είναι πολύτιμη καθώς μπορούμε να προτείνουμε στους χρήστες βιβλία που πιθανών να τους αρέσουν. Επομένως, κάνει το παραπάνω μοντέλο πιο εύστοχο εφόσον είναι περισσότερες οι πιθανότητες να μείνουν οι καταναλωτές ικανοποιημένοι από την σύσταση των βιβλίων αλλά ταυτόχρονα πιο περιορισμένο καθώς τα δεδομένα είναι αρκετά λιγότερα σε σχέση με πριν.

Χρησιμοποιήθηκε, και πάλι, ο αλγόριθμος `apriori` ώστε να πραγματοποιηθεί ανάλυση κανόνων συσχετίσεων. Μέσω της ανάλυσης κανόνων συσχετίσεων εξάγεται πληροφόρηση σχετικά με το ποια προϊόντα διαβάστηκαν συνδυαστικά και ταυτόχρονα εκτιμήθηκαν από τους αναγνώστες, καθιστώντας ικανή την πρόταση νέων βιβλίων στους χρήστες. Η διαδικασία, οι πράξεις, οι βιβλιοθήκες της `rgthon` και η υλοποίηση του αλγορίθμου είναι ίδια με το προηγούμενο μοντέλο. Η μόνη διαφορά έγκειται στο γεγονός ότι αφαιρέσαμε βιβλία με αρνητικές αξιολογήσεις(αξιολογήσεις μικρότερες του 6) από τους χρήστες. Τέτοια βιβλία δεν εξυπηρετούν την ουσία του συγκεκριμένου ερωτήματος όπου είναι η ανίχνευση επιτυχημένων/κορυφαίων συνδυασμών βιβλίων.

Κάποια ενδεικτικά στιγμιότυπα:

- Από το πίνακα `final` όπου περιέχει τα τελικά δεδομένα του `Fact Table` έτσι όπως εξήχθησαν από το `Warehouse`, ελέγγω τα βιβλία που διάβασε ο κάθε χρήστης. Με άλλα λόγια ελέγγω τον `Fact Table` και μόλις βρω ότι ο χρήστης έκανε κάποια κριτική για κάποιο βιβλίο πηγαίνω και σημειώνω στο `association_df` ότι ο συγκεκριμένος χρήστης όπου βρίσκεται στη γραμμή `users_dict[row['Reader']]` έχει διαβάσει το βιβλίο που βρίσκεται στη στήλη `row['Book_Title']`. Με αυτό το τρόπο έχω στο `association_df` την πληροφορία ποιος χρήστης διάβασε ποια βιβλία.

```
[11] ✓ 1.9s Python
```

```
for index, row in final.iterrows():
    association_df.at[users_dict[row['Reader']], row['Book']] = True
```

- Από προεπιλογή, το `apriori` επιστρέφει τους δείκτες των στήλων των στοιχείων, οι οποίοι μπορεί να είναι χρήσιμοι σε επόμενες λειτουργίες, όπως η εξόρυξη κανόνων συσχέτισης. Για καλύτερη αναγνωσιμότητα, μπορούμε να ορίσουμε `use_colnames=True` για να μετατρέψουμε αυτές τις ακέραιες τιμές στα αντίστοιχα ονόματα στοιχείων

```
[12] ✓ 0.9s Python
```

```
df = apriori(association_df, min_support=0.005, use_colnames=True, verbose=1, max_len= None, low_memory=False )
```

... Processing 264 combinations | Sampling itemset size 6543

	support	itemsets
0	0.019749	(107643)
1	0.018851	(107802)
2	0.157989	(38052)
3	0.037702	(37073)
4	0.021544	(80686)
...
1266	0.005386	(36835, 36422, 130572, 36277, 36120)
1267	0.006284	(36067, 36835, 36422, 36277, 36120)
1268	0.005386	(36067, 36835, 36422, 130572, 36277)
1269	0.005386	(36067, 36835, 36422, 130572, 36120)

```
[Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)', 'Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)', 'One for the Money (A Stephanie Plum Novel)']
['Four To Score (A Stephanie Plum Novel)', 'High Five (A Stephanie Plum Novel)']
-----> support 0.0062836624775583485 & confidence 1.0
['Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)', 'Two for the Dough', 'Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)'] ['Four To Score (A Stephanie Plum Novel)']
-----> support 0.010771992818671455 & confidence 1.0
['Harry Potter and the Goblet of Fire (Book 4)', 'Harry Potter and the Sorcerer's Stone (Book 1)', 'The Catcher in the Rye'] ['Harry Potter and the Chamber of Secrets (Book 2)']
-----> support 0.0062836624775583485 & confidence 1.0
['One for the Money (Stephanie Plum Novels (Paperback))', 'Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel)', 'Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)'] ['Four To Score (A Stephanie Plum Novel)']
-----> support 0.005385996489335727 & confidence 1.0
['Harry Potter and the Goblet of Fire (Book 4)', 'Harry Potter and the Sorcerer's Stone (Book 1)', 'The Catcher in the Rye'] ['Harry Potter and the Chamber of Azkaban (Book 3)']
-----> support 0.0062836624775583485 & confidence 1.0
```

- Επιλέγω τους κορυφαίους 10 συνδυασμούς

```
[13] ✓ 0.1s Python
```

```
df_ar = association_rules(df, metric = "confidence")
final_results = df_ar.sort_values(by=['confidence'], ascending=False).iloc[:5][['antecedents', 'consequents', 'support', 'confidence', 'lift']]
```

- Εμφανίζω τους κωδικούς βιβλίων των τελικών αποτελεσμάτων. Απομένει να αντικαταστήσω τους κωδικούς με τους τίτλους των βιβλίων

```
[14] ✓ 0.2s Python
```

```
final_results
```

	antecedents	consequents	support	confidence	lift
233	(36067, 36422, 36835)	(36120, 36277)	0.006284	1.0	42.846154
158	(36067, 130572, 36422)	(36120)	0.010772	1.0	25.906977
90	(82995, 123557, 38215)	(82757)	0.006284	1.0	13.925000
154	(8585, 36067, 36422)	(36120)	0.005386	1.0	25.906977
87	(82995, 123557, 38215)	(82974)	0.006284	1.0	12.804598

Τα τελικά αποτελέσματα σε μορφή πίνακα:

Antecedents	Consequents	Support	Confidence
1.Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel) 2.Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel) 3.One for the Money (A Stephanie Plum Novel)	1. Four To Score (A Stephanie Plum Novel) 2. High Five (A Stephanie Plum Novel)	0.0062836624775583485	1.0
1. Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel) 2.Two for the Dough 3. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)	Four To Score (A Stephanie Plum Novel)	0.010771992818671455	1.0
1. Harry Potter and the Goblet of Fire (Book 4) 2. Harry Potter and the Sorcerer's Stone (Book 1) 3.The Catcher in the Rye	Harry Potter and the Chamber of Secrets (Book 2)	0.0062836624775583485	1.0
1. One for the Money (Stephanie Plum Novels (Paperback)) 2. Three To Get Deadly : A Stephanie Plum Novel (A Stephanie Plum Novel) 3. Hot Six : A Stephanie Plum Novel (A Stephanie Plum Novel)	Four To Score (A Stephanie Plum Novel)	0.005385996409335727	1.0
1. Harry Potter and the Goblet of Fire (Book 4) 2. Harry Potter and the Sorcerer's Stone (Book 1) 3. The Catcher in the Rye	Harry Potter and the Prisoner of Azkaban (Book 3)	0.0062836624775583485	1.0

Ανάλυση για το τελευταίο κελί του παραπάνω πίνακα:

Παρατηρούμε ότι όποιος αγοράζει τα βιβλία:

1. Harry Potter and the Goblet of Fire (Book 4)
2. Harry Potter and the Sorcerer's Stone (Book 1)
3. The Catcher in the Rye

Τότε αγοράζει και το βιβλίο Harry Potter and the Prisoner of Azkaban (Book 3) όπου και βγάζει απόλυτο νόημα.

Πιο αναλυτικά

3. Υποστήριξη 0.00628366: Η πιθανότητα να αγοραστούν συνδυαστικά τα βιβλία
4. Εμπιστοσύνη 1: Από το σύνολο των αγορών που περιέχουν τα βιβλία Harry Potter and the Goblet of Fire (Book 4), Harry Potter and the Sorcerer's Stone (Book 1), The Catcher in the Rye, το 100% των αγορών θα περιέχουν το βιβλίο Harry Potter and the Prisoner of Azkaban (Book 3)

Έτσι, εξάγονται χρήσιμα συμπεράσματα σχετικά με τους αγαπημένους συνδυασμούς βιβλίων των αναγνωστών.

Τρίτο Μοντέλο:

Όσον αφορά στο τρίτο μοντέλο πραγματοποιήθηκε μία ομαδοποίηση των αναγνωστών , προκειμένου να εξαχθούν χρήσιμες πληροφορίες σχετικά με τις προτιμήσεις τους, τις οποίες μπορεί να χρησιμοποιήσει ένα ηλεκτρονικό βιβλιοπωλείο σαν την Amazon για να κάνει περισσότερο προσωποποιημένες τις δράσεις marketing των βιβλίων της.

Το κριτήριο για το segmentation του αναγνωστικού κοινού προέκυψε ύστερα από μία κατηγοριοποίηση που εφαρμόσαμε στα βιβλία των δεδομένων μας. Συγκεκριμένα, δημιουργήσαμε κάποιες labels/ετικέτες, μετα-δεδομένα, τις οποίες στην συνέχεια αποδώσαμε στο κάθε βιβλίο. Πιο αναλυτικά, προέκυψαν τρεις κατηγορίες:

1. Ευρέως αρεστά βιβλία: αναφέρεται σε βιβλία που είναι παγκοσμίως αρεστά και αποδεκτά. Παραδείγματα τέτοιων είναι κλασσικά βιβλία, βιβλιογραφίες και λογοτεχνικά αριστουργήματα και πολλά ακόμη. Ως ευρέως αρεστά θα ορίσουμε τα βιβλία με μέσο όρο αξιολογήσεων μεγαλύτερου του 6 και διακύμανση μικρότερη του 1.67
2. Αμφιλεγόμενα: η κατηγορία αυτή αναφέρεται στα βιβλία που έχουν καταφέρει να διχάσουν το κοινό. Βιβλία που κάποιοι τα λάτρεψαν ενώ άλλοι τα απέρριψαν. Στην κατηγορία αυτοί τοποθετούνται όλα τα βιβλία με τυπική απόκλιση μεγαλύτερη του 1.67
3. Ευρέως μη αρεστά: η κατηγορία αυτή αναφέρεται σε βιβλία που δεν κατάφεραν να έχουν απήχηση στο κοινό και συγκέντρωσαν μαζικά αρνητικές αξιολογήσεις. Με άλλα λόγια, στην κατηγορία τοποθετούνται βιβλία με μέσο όρο αξιολογήσεων μικρότερο του 6 και τυπική απόκλιση μικρότερη του 1.67

Τα όρια αυτά, για τον μέσο όρο και την τυπική απόκλιση, επιλέχθηκαν ύστερα από ανάλυση των στατιστικών των βαθμολογιών για βιβλία που έχουν βαθμολογηθεί πάνω από 100 φορές. Σε αυτό το σημείο, είναι εύλογο να προταθεί η απόσυρση βιβλίων που ανήκουν στην 3^η κατηγορία, αφού δεν έχουν ζήτηση από το κοινό, επομένως δεν επιφέρουν κέρδος και το πλήθος των αρνητικών αξιολογήσεων που προκαλούν μπορεί να φθείρει την γενική εικόνα του βιβλιοπωλείου.

Έτσι, αφού κάθε βιβλίο ανήκει σε μία από τις τρεις κατηγορίες, αντιστοιχήσαμε τα βιβλία αυτά στους αναγνώστες που τα έχουν βαθμολογήσει και υπολογίσαμε τον μέσο όρο βαθμολογίας κάθε αναγνώστη ανά κατηγορία βιβλίου. Επιπλέον, υπολογίσαμε την συχνότητα που κάθε χρήστης αγοράζει από κάθε κατηγορία προϊόντων. Όπως, προβλέψαμε ήταν αρκετά σπάνιο φαινόμενο η αγορά βιβλίων της 3^η κατηγορίας, έτσι ήταν εύκολο να την αφαιρέσουμε θέτοντας την συνθήκη: η συχνότητα εμφάνισης κάθε κατηγορίας για κάθε αναγνώστη να είναι μεγαλύτερη του 2.

Έχοντας, λοιπόν στην διάθεση την πληροφορία για τον κάθε χρήστη πόσο συχνά αγοράζει από τις κατηγορίες 1 και 2, και ποιος είναι ο μέσος όρος των αξιολογήσεων που έχει δώσει στα βιβλία της κάθε κατηγορίας, προχωρήσαμε στην εκτέλεση ενός αλγορίθμου συσταδοποίησης, k-means, με k=2, χρησιμοποιώντας το εργαλείο Rapid Miner.

Τα αποτελέσματα της συσταδοποίησης συνεπάγονται πολύτιμη πληροφορία. Αρχικά, γνωρίζουμε για κάθε χρήστη τις προτιμήσεις του περί κλασσικών ή μη κλασσικών βιβλίων. Το τελευταίο είναι εξαιρετικά σημαντικό καθώς επιτρέπει τη γνωριμία με τους αναγνώστες, την κατανόηση των αναγκών τους, την αμεσότερη επικοινωνία και προσωποποιημένη πρόταση βιβλίων. Επίσης, από τα δεδομένα εξάγονται πολλά συμπεράσματα, μεταξύ άλλων, και αυτό της τάσης των ηλικιών προς συγκεκριμένες κατηγορίες. Το παρακάτω διάγραμμα παρουσιάζει μια καμπύλη σύμφωνα με την οποία τα νεαρά άτομα τείνουν προς τα αμφιλεγόμενα βιβλία, στην συνέχεια ηλικιακή ομάδα 35-55 «στρέφεται» προς τα κλασσικά βιβλία και πέρα της ηλικίας των 55 τα άτομα τείνουν να εκτιμούν περισσότερο «αμφιλεγόμενα» βιβλία. Αποτελέσματα πλήρως αναμενόμενα με δεδομένα ότι την ηλικιακή ομάδα των 35-55 χαρακτηρίζει μία σοβαρότητα και ωριμότητα, γεγονός που αναπαρίστανται/απεικονίζεται και στις αγορές βιβλίων.



7. Παράρτημα

SQL Script για καθαρισμό βιβλίων :

```
Enter SQL Query

Use Book_Ratings;
UPDATE [Book_Ratings].[dbo].[Book_Staging_Table]
SET ISBN = SUBSTRING(ISBN,2,len(ISBN));
DELETE FROM Book_Staging_Table WHERE len(ISBN) != 10 AND len(ISBN) != 13;
DELETE FROM Book_Staging_Table WHERE len(Year_Of_Publication) != 4;
DELETE FROM Book_Staging_Table WHERE Publisher = null;
ALTER TABLE Book_Staging_Table
ALTER COLUMN Year_Of_Publication int;

WITH CTE AS(
    SELECT [ISBN], [Book_Title], [Book_Author], [Year_Of_Publication], [Publisher],
    RN = ROW_NUMBER(OVER(PARTITION BY [isbn],[Book_Title] ORDER BY [isbn])
    FROM [dbo].[Book_Staging_Table]
)
DELETE FROM CTE WHERE RN > 1
```

SQL Script για καθαρισμό αναγνωστών:

```
Enter SQL Query

Use Book_Ratings;

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Reader = SUBSTRING(Reader,2,len(Reader)) FROM [Book_Ratings].[dbo].[User_Staging_Table];

UPDATE [Book_Ratings].[dbo].[User_Staging_Table] SET Age=0 FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE len(Age) = 0;

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Age = SUBSTRING(Age,1,len(Age)-1);

ALTER TABLE User_Staging_Table
ALTER COLUMN Age int;

UPDATE [Book_Ratings].[dbo].[User_Staging_Table] SET Age= 0 WHERE Age > 90 OR Age < 6;

update [Book_Ratings].[dbo].[User_Dim]
Set Age = round(Age/5.0)/5

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = RIGHT(Location, CHARINDEX('-',REVERSE(Location))-1) FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE CHARINDEX('-', Location) != 0;

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = SUBSTRING(Location,1,len(Location)-6) FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE CHARINDEX('-', Location) != 0;

DELETE FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE len(Location) <= 2;

DELETE FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE (Location in (select Location FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE (Location LIKE '%[a-zA-Z0-9]%' Group By Location having (count(*) > 90 OR count(*) < 24)));
DELETE FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE (location LIKE '%[0-9]%' );

DELETE FROM [Book_Ratings].[dbo].[User_Staging_Table] WHERE (Location in (SELECT Location FROM [Book_Ratings].[dbo].[User_Staging_Table] Group By Location having (count(*) < 24)));

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'United kingdom' WHERE (Location = 'kingdom' OR Location = 'england' OR Location = 'england. ');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'new zealand' WHERE (Location = 'zealand');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'costa rica' WHERE (Location = 'rica');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'szech republic' WHERE (Location = 'republic');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'china' WHERE (Location = 'kong');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'spain' WHERE (Location = 'espana');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'in lanka' WHERE (Location = 'lanka');

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Location = 'usa' WHERE (Location = 'illinois,' OR Location = 'washington,' OR Location = 'florida,' OR Location = 'massachusetts,' OR Location = 'pennsylvania,' OR Location = 'york,' OR Location = 'california,' OR Location = 'texas,');

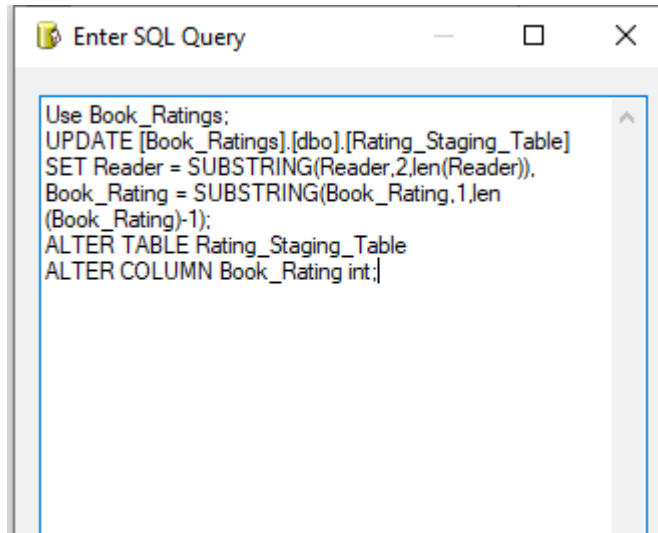
update [Book_Ratings].[dbo].[User_Staging_Table]
set location = 'mexico' where [location] LIKE '%xico'

Activate Windows
Go to Settings to activate Windows
```

```
Enter SQL Query

UPDATE [Book_Ratings].[dbo].[User_Staging_Table]
SET Age = (select AVG(Age) from [Book_Ratings].[dbo].[User_Staging_Table] where Age = 0;
```

SQL Script για καθαρισμό βαθμολογήσεων:



```
Use Book_Ratings;
UPDATE [Book_Ratings].[dbo].[Rating_Staging_Table]
SET Reader = SUBSTRING(Reader,2,len(Reader)),
Book_Rating = SUBSTRING(Book_Rating,1,len
(Book_Rating)-1);
ALTER TABLE Rating_Staging_Table
ALTER COLUMN Book_Rating int;
```

Κώδικας Python

Τα αρχεία κώδικα που παραδόθηκαν:

- association_rule_model1: περιέχει τον κώδικα για την πραγματοποίηση της ανάλυσης κανόνων συσχετίσεων
- association_reviews_model2: περιέχει τον κώδικα για την πραγματοποίηση της ανάλυσης κανόνων συσχετίσεων, αυτή τη φορά λαμβάνοντας υπόψιν μόνο αγαπημένα βιβλία

Δεδομένα:

- fact_table.txt: το αρχείο έχει εξαχθεί από το data warehouse και αντιστοιχεί στα δεδομένα πίνακα fact Table όπως προέκυψαν μετά τη διαδικασία ETL

8. Αναφορές

[Improving Recommendation Lists Through Topic Diversification](#),

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, Georg Lausen; *Proceedings of the 14th International World Wide Web Conference (WWW '05)*, May 10-14, 2005, Chiba, Japan. *To appear*.