



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα

ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ

ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ

ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εμπλουτισμός δεδομένων recommender systems με Linked Open Data

Λουκάς Μάριος Ι. Μοσχάτος

ΕΠΙΒΛΕΠΩΝ: Δρ. Χαράλαμπος Μπράτσας

Τμήμα Μαθηματικών Α.Π.Θ.

Υπό την εποπτεία: Ιωάννη Αντωνίου

Καθηγητής Σ.Θ.Ε. Α.Π.Θ.

Θεσσαλονίκη, Δεκέμβριος 2016



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα

ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ

ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ

ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ



Εμπλουτισμός δεδομένων recommender systems με Linked Open Data

Λουκάς Μάριος Ι. Μοσχάτος

ΕΠΙΒΛΕΠΩΝ: Δρ. Χαράλαμπος Μπράτσας

Τμήμα Μαθηματικών Α.Π.Θ.

Υπό την εποπτεία: Ιωάννη Αντωνίου

Καθηγητής Σ.Θ.Ε. Α.Π.Θ.

Εγκρίθηκε από την Τριμελή Εξεταστική Επιτροπή την 20 δεκεμβρίου 2016

.....
Χ. Μπράτσας

ΕΕΔΙΠΣ.Θ.Ε. Α.Π.Θ.

.....
Ι. Αντωνίου

Καθηγητής Σ.Θ.Ε. Α.Π.Θ.

.....
Π. Μπαμίδης

Επικουρος καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Δεκέμβριος 2016

.....
Λουκάς Μάριος Ι. Μοσχάτος

Πτυχιούχος Μαθηματικών Α.Π.Θ.

Copyright © Λουκάς Μάριος Ι. Μοσχάτος, 2016 Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευτεί ότι εκφράζουν τις επίσημες θέσεις του Α.Π.Θ.

Στους

Γονείς μου απόντες και μη

ΠΕΡΙΛΗΨΗ

Σε αυτή την εργασία περιγράφονται αλγόριθμοι συστάσεων, ο εμπλουτισμός των δεδομένων μέσω της εκμετάλλευσης ανοικτών διασυνδεδεμένων δεδομένων, η δημιουργία ενός τέτοιου αλγορίθμου ώστε να εξεταστεί το όφελος που προκύπτει και τέλος ο σχεδιασμός μίας εφαρμογής για την εύρεση προτεινόμενων ταινιών. Οι αλγόριθμοι συστάσεων μπορούν να έχουν πολλές διαφορετικές εκφάνσεις, με πολλές από τις οποίες ερχόμαστε συχνά σε επαφή χωρίς συνήθως να το αντιλαμβανόμαστε, όπως τους ιστότοπους: Amazon , Netflix, EBay, YouTube, κ.α. Ο λόγος που υπάρχουν τόσες διαφορετικές μηχανές συστάσεων είναι η αύξηση των πωλήσεων και η ευχρηστία το εκάστοτε ηλεκτρονικού καταστήματος ή μίας υπηρεσίας αυξάνοντας έτσι το κέρδος και την επισκεψιμότητα. Όσον αφορά τα δεδομένα που θα χρησιμοποιηθούν στο χτίσιμο του αλγορίθμου, επιλέξαμε να ασχοληθούμε με κινηματογραφικά δεδομένα από τον ιστότοπο Movielens, ο οποίος παρέχει δεδομένα από πραγματικές βαθμολογίες χρηστών αλλά και τους τύπους των ταινιών που βαθμολογήθηκαν. Τα δεδομένα εμπλουτίστηκαν περαιτέρω χρησιμοποιώντας τους πόρους της Wikipedia. Ο ευκολότερος τρόπος για να αντληθούν μαζικά τα δεδομένα είναι μέσω του end-point της DBpedia. Στη συνέχεια τα δεδομένα αποσφαλμάτωθηκαν και μορφοποιήθηκαν ώστε να δημιουργηθεί ένα εύχρηστο dataset για την υλοποίηση του αλγορίθμου. Ακολούθως ελέγξαμε τη διαφορά που προκύπτει στα αποτελέσματα μετά την χρήση των διασυνδεδεμένων δεδομένων . Τέλος σχεδιάσαμε μια εφαρμογή η οποία μας βοηθά να βρούμε όμοιες ταινίες. Η εργασία αυτή δίνει μία βάση για το πώς μπορεί να εκμεταλλευτεί κανείς τα διασυνδεδεμένα δεδομένα σε τέτοια συστήματα.

Λέξεις κλειδιά: Αλγόριθμοι συστάσεων, Διασυνδεδεμένα δεδομένα, DBpedia.

ABSTRACT

In this work we describe recommender system algorithms, the creation of one such algorithm, and the augmentation of data using Open Linked Data. Recommendation system algorithms can be found in many forms, which we usually come across without even noticing, as is with the case of the sites: Amazon , Netflix, EBay, YouTube, and other. The reason there are so many different recommender systems is the increase of sales and the friendliness of the site, thus increasing profit and times of visit. For the data we are going to use for the creation of our recommender, we choose to use cinematography data from the site Movielens, which provides files with real ratings of movies by real users and also the types of each movie. The data was enriched using Wikipedia resources. The easiest way to quickly obtain big data is through the DBpedia end-point by SPARQL querying taking advantage of the ontologies. Next the data was cleaned and formatted, to make it suitable for creating our algorithm, using R and mainly the library “SPARQL” and Open refine. For the creation of the algorithm we once again used R and the very helpful library for this case “recommenderlab”. Finally we built a simple recommender, which will help find movies similar to one we like.

Keywords: Recommender system, Linked data, DBpedia.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	6
ABSTRACT	7
ΠΕΡΙΕΧΟΜΕΝΑ	8
ΕΙΣΑΓΩΓΗ	10
SUMMARY	11
ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ	20
1. RECOMMENDER SYSTEMS	21
1.1 Κλασικά Προβλήματα	26
1.2 Κατάταξη βάση προσωποποίησης	27
1.3 Κατάταξη βάση μεθόδου	30
1.4 Μαθηματικά εργαλεία	34
1.5 Χρήσιμες μετρικές για έλεγχο απόδοσης	38
2. ΜΕΘΟΔΟΛΟΓΙΑ	40
3. ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΕΞΟΡΥΞΗΣ ΚΑΙ ΚΑΘΑΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ	41
3.1 Εξόρυξη δεδομένων	41
3.2 Καθαρισμός	44
3.3 Βιβλιοθήκες της R studio	50
4. ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ, ΔΟΜΗΣΗ ΚΑΙ ΚΑΘΑΡΙΣΜΟΣ	51
4.1 Δεδομένα βαθμολογιών	51
4.2 Δόμηση των δεδομένων	51
4.3 Εξόρυξη δεδομένων	53
4.4 Καθαρισμός δεδομένων	55
5. ΑΛΓΟΡΙΘΜΟΣ ΣΥΣΤΑΣΕΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΤΑΙΝΙΩΝ	56
5.1 Επιλογή αλγορίθμου	56
5.2 Δομή αλγορίθμου και αποτελέσματα	57
6. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ	62
ΣΥΜΠΕΡΑΣΜΑΤΑ και ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΕΥΝΕΣ	63
ΒΙΒΛΙΟΓΡΑΦΙΑ	65
ΧΡΗΣΙΜΟΙ ΣΥΝΔΕΣΜΟΙ	67

ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία ασχολείται με την ανάλυση το μηχανών συστάσεων τον εμπλουτισμό δεδομένων χρησιμοποιώντας ανοικτά διασυνδεδεμένα δεδομένα αλλά και με την δημιουργία μιας τέτοιας μηχανής στην προγραμματιστική γλώσσα **R**. Στόχος μας είναι να εξετάσουμε αν μια διαδικασία εμπλουτισμού δεδομένων μπορεί να βελτιώσει τις επιδώσεις τέτοιων αλγορίθμων, αλλά και να εξεταστεί σε τι επίπεδο βρίσκονται οι ανοικτές πηγές όσο αφορά την ποιότητα και ποσότητα των δεδομένων που είναι διαθέσιμα από ανοικτές πλατφόρμες. Όπως όλες οι μηχανές που κάνουν προβλέψεις έτσι και αυτά τα συστήματα μπορούν να επωφεληθούν όταν η βάση δεδομένων που χρησιμοποιούν εμπλουτιστεί με περισσότερη πληροφορία. Για τα δεδομένα μας επιλέξαμε να χρησιμοποιήσουμε τα δεδομένα που παρέχει το site **MOVIELENS** κρατώντας μόνο τις πληροφορίες για το είδος των ταινιών και βαθμολογιών τους από χρήστες. Η βάση δεδομένων για τον εμπλουτισμό ήταν η **DBPEDIA**. Από το end-point της μέσω ερωτημάτων **SPARQL** μπορούμε να κατεβάσουμε δεδομένα είτε μεμονωμένα είτε μαζικά. Αφού δομηθούν κατάλληλα τα δεδομένα θα ακολουθήσει η περιγραφή ενός item-Based Collaborative filtering αλγορίθμου. Θα συγκριθούν τα αποτελέσματα του με ή χωρίς την ένταξη των δεδομένων που αφορούν ιδιότητες των ταινιών. Τέλος θα περιγράψει η δημιουργία μια εφαρμογής που μας δίνει την δυνατότητα να βρούμε παρόμοιες ταινίες.

SUMMARY

In this master thesis we are going to investigate if data enrichment for Recommender Systems datasets is viable, and to what degree. We will also create simple **RS** algorithms to test if the data added from linked open data can make a significant difference. To understand better recommender systems let's analyze the categories by two variables: personalization and method.

- **Based on personalization**

1. Non personalized

This category consists of systems that create recommendation based only upon trends, popularity and ratings. The users demographics is not at all taken into consideration.

2. Demographic based

The term demographic means that the algorithm has the ability to divide the recommendations by large groups of people. This is usually achieved using cookies. Main criteria are: age, gender, season, hobby preferences and others.

3. Ephemeral

Such systems usually require the user to input information for current interests. These machines are greatly useful in shops that users rarely use. Consider a gift shop. The user can input his friend's preferences and easily find a suitable present.

4. Persistent

Persistent systems match long-term interest of their users. They require a long term relationship with the user to learn their preferences. Thus there is danger

in situation that the user changes preferences as the user's profile is hard to change, and the system designers have to take careful consideration.

- **Based on method**

1. **Non-Personalized Summary statistics**

This category describes systems that use concepts like: Best selling product, hot topic, most popular. The statistics can be fed by external sources such as Google. These are generally easy to implement and do not require a lot of cpu time.

2. **Content-Based Filtering**

Information Filtering

By analyzing the data base we can assume the preferences of each user. Consider a user that picks a lot of action movies the system can realize the preference and make well adjusted suggestions. This idea is also applicable to other item characteristics.

3. **Knowledge-based**

This method requires tracing the users interests in a different way. It is most common to request the user for direct input of that information in the system. This may alienate some user as we have become used to having everything ready for us, yet it alleviates the **Cold-start** problem.

4. **User based Colaborative filtering**

The technique utilizes the creation of user neighborhoods or simply put calculating a user similarity score. The idea is that similar user will have very similar tastes thus creating good recommendations for each other, of course without noticing their neighborhood.

5. Item based Colaborative filtering

Similarly to the user based collaborative filtering, the system calculates item similarity scores using the rating matrix and/or by similar attributes. With these similarities computed we can suggest similar items to a user that we can assume is interested in a group of items. This approach is generally considered to produce marginally worse results compared to the previous method, but it is also somewhat faster and the gains in computation times are usually worth the penalty.

6. Dimensionality reduction

As the title suggests this technique is about reducing the dimensions of the rating matrix between users and items. It can provide acceleration to the computing process. It is also a good way to tackle multipurpose application as no specific tuning is needed. The mathematics used is: **Singular value decomposition, Stochastic gradient descent, Alternating least squares, Principal Component Analysis.**

7. Hybrid recommenders

Many times it is a sound strategy to combine the **ratings** using:

Weighted, switching, Mixed, Feature Combination, Feature Augmentation, Cascade or Meta-level techniques

We also have to mention the mathematical methods used in Recommender systems.

- **Dimensionality reduction**
- **Principal component analysis:** This procedure uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**.
- **Singular Value Decomposition:** This algebra method decomposes the matrix to its singular values. More specifically the target is to solve the equation: $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$

Where $\mathbf{U}$ is an $m \times m$ where $\mathbf{U}$ is a unitary matrix, $\mathbf{\Sigma}$ is a $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{V}$ is an $n \times n$ unitary matrix.

- **Clustering:** This is the method of creating groups of similar user or items.
- **K-Nearest Neighbours:** This technique involves finding the k-nearest neighbours by calculating similarities between users or items. This is a very good method and fairly fast as well. It is thus widely used in **Collaborative filtering**.
- **k-means:** This method has the same goal but it is achieved by repeatedly adjusting the groups until a requirement is met i.e no more changes in the last 2 loops.
- **Artificial Neural Networks:** Neural networks are based on the behavior of the human brain. They use neurons in many levels each feeding the next one. They can identify relationship beyond the linear ones. The level amounts and neuron count have to be adjusted accordingly.
- **Support Vector Machine:** The goal is to calculate hyperplanes to divide the space of the values by maximizing the distance from the hyperplane, thus creating ideal splits.
- **Multinomial logistic regression:** Mostly used in mixed type algorithms with feature combination.
- **Other collaborative filtering:** Algorithms use the rating matrix to calculate missing values for the matrix, as described above.
- **Random forests:** They run many parallel decision trees and then combine the results. They offer good result if used right, they are very flexible and can be used in many applications. Over-fitting is also rare due to their random nature.

Useful metrics

RMSE: Root square mean error or root mean square deviation (RMSD). It is used in regression and generally prediction models. It is punished large deviations from the real value.

$$Rmse = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}$$

MAE: Mean Absolute Error. This measures focuses more on the small deviations. Using both measures is a good way to understand our result more in depth.

$$Rmse = \sqrt{\frac{\sum_{i=1}^n |y_i - x_i|}{n}}$$

MSE: Mean squared error. Virtually works similarly to RMSE.

$$Mse = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$$

Precision: $Precision = \frac{\{relative\} \cap \{received\}}{\{received\}}$

Recall: $Recall = \frac{\{relative\} \cap \{received\}}{\{relative\}}$

F-measure: $F = 2 \cdot \frac{Precision+recall}{Precision+recall}$

Recommender Systems Problems

- **Cold-Start problems:** With the term cold-start we define the difficult task of providing accurate results for first time and new users. There is a difficulty with new items as well but it is generally easier to implement solutions for them. As for the user we can directly demand information to build a profile, yet it is not often used as user don't value such interactions.
- **Diversity:** As with many systems, RSs are practically a trade-off between diversity and accuracy. Some tactics are applying a greater weight to more recent movies, or shuffling the top-n recommendations.
- **Scalability:** A RS needs to be scalable. Meaning it has to perform well in larger datasets, as the rating matrix is dynamic and constantly growing.
- **Nonintrusiveness:** Many RSs demand great involvement from the user or they demand access to personal data. Most users nowadays do not mind giving their data, but many also will not even take the time to provide rating. There are nonintrusive ways around this, such as the time he spent watching a trailer.

RATING DATA AND LINKED OPEN DATA

Data for recommender systems come in many shapes and sizes. For our test we are going to use the dataset provided by MovieLens. We are going to add to that dataset more information from Linked open Data.

Linked data is the term that describes the machine readable formats thus augmenting their usability. In the base of this technology stand ontologies. Ontologies are sets of triples describing concepts and property relationships, either in specific categories or more generic ones. Οι βάσεις πάνω στις οποίες στήνονται τα δεδομένα ονομάζονται οντολογίες. Published Data in the format mentioned above, rank in the top of Tim Berners Lee, star ranking system. To access the data in RDF format, the SPARQL language was created.

From the well know to everyone, Wikipedia and through the DBpedia endpoint. Most of the information in the info-box in Wikipedia can be recalled using RDF technology. We used R to automatically call properties for each movie in the data set, using SPARQL.

SPARQL (SPARQL Protocol and RDF Query Language) this very important technology of the web is basically a language that makes it possible to access data in RDF format.

Algorithms and Testing

In order to test the performance between the two datasets, we will use a simple item based collaborative filtering, that calculates similar items, and produces the predictions for the new ratings.

To incorporate all data in the system we used linear regression to calculate the weights for each similarity matrix calculating by user. We run this by the assumption that the weights should be optimized for the least error possible by user. We normalized once per user so that the sum of weights is equal to zero, and also after calculating the means. Finally we excluded negative weights, and we got the following results:

- director 0.17913764
- Star -
- writer -
- distributor 0.78463883
- genre -
- ratings 0.02497441
- Year -

	f-measure	RMSE	MAE	MSE
Ratings only	9,466887	1,011	1,021	0,808
Combined	4,713576	1,025	1,05	0,816

We can see the error is very close to the original algorithm, but the f-measure is almost halved. This could possibly be due to the fact that all the weight stands on the director, a reason for which is that other properties had many missing values.

It has to be noted that the data retrieved from DBpedia was incomplete. We run the procedure again for the non-missing values of the LOD with the following results:

- director 0.379372318
- Star -
- writer -

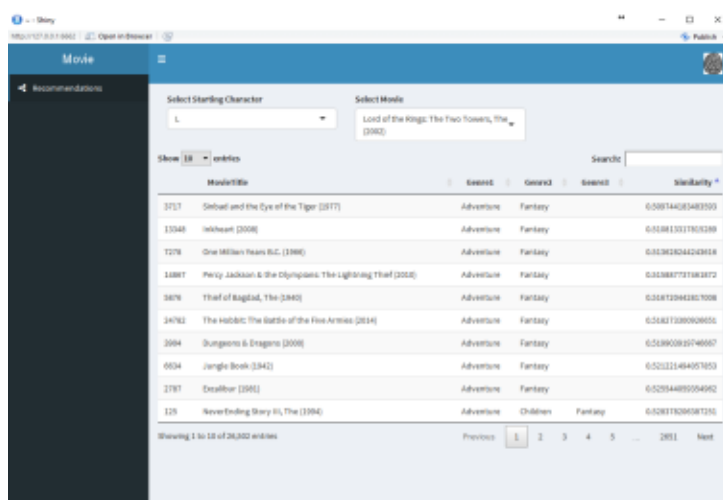
- distributor 0.618277133
- genre 0.002350549
- ratings -
- Year -

The results for all tries:

	f-measure	RMSE	MAE	MSE
Ratings	9,466887	1,011	1,021	0,808
Combined	4,5397350	1,015	1,03	0,811
Combined and common	4,2251655	1,022	1,045	0,816

We got worse results thus for the data we have it is best to use all data even if there are missing values.

Finally we created a simple application that requires the user to input a movie he likes and returns similar movies according to a similarity matrix we created from user ratings and the movie genres.



ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Η βασική ιδέα της εργασίας είναι να προτείνει μια νέα πηγή πληροφορίας για τα Recommender systems και επίσης να εξετάσει κατά πόσο αυτό βοηθάει στην απόδοση των αλγορίθμων τέτοιου είδους. Καθώς οι βάσεις δεδομένων των Recommender systems συνεχώς αυξάνουν σε πλήθος τόσο από χρήστες όσο και από items, είναι λογικό τα συστήματα αυτά να έχουν απώλειες σε απόδοση και κάθε αύξηση αυτής είναι επιθυμητή. Ο πρώτος στόχος είναι να εμπλουτιστούν τα δεδομένα που είναι διαθέσιμα σε εμάς, εξετάζοντας παράλληλα, σε τι επίπεδο είναι δυνατό να συμβεί αυτό, μέσω της εκμετάλλευσης των ανοιχτών διασυνδεδεμένων δεδομένων. Σε δεύτερο στάδιο θα πρέπει να τρέξουμε απλούς αλγόριθμους (item based collaborative filtering) με εμπλουτισμένα αλλά και μη εμπλουτισμένα δεδομένα ώστε να δούμε αν υπάρχει ουσιαστικό νόημα στο εγχείρημα, και να ελέγξουμε αν είναι νωρίς ακόμα να επενδύσει κάποιος χρόνο στην εκμετάλλευση των ανοιχτών δεδομένων. Τέλος θα δούμε πως μπορεί να στηθεί μια σχετική εφαρμογή.

1. RECOMMENDER SYSTEMS

Σε αυτό το κεφάλαιο θα γνωρίσουμε τι είναι ένα Recommender system, θα δούμε προβλήματα που αντιμετωπίζουμε, τους τρόπους που κατατάσσουμε τα Recommender System, μαθηματικές μεθόδους και τέλος ποιες μετρικές χρησιμοποιούνται για να ελεγχθεί η απόδοση τους.

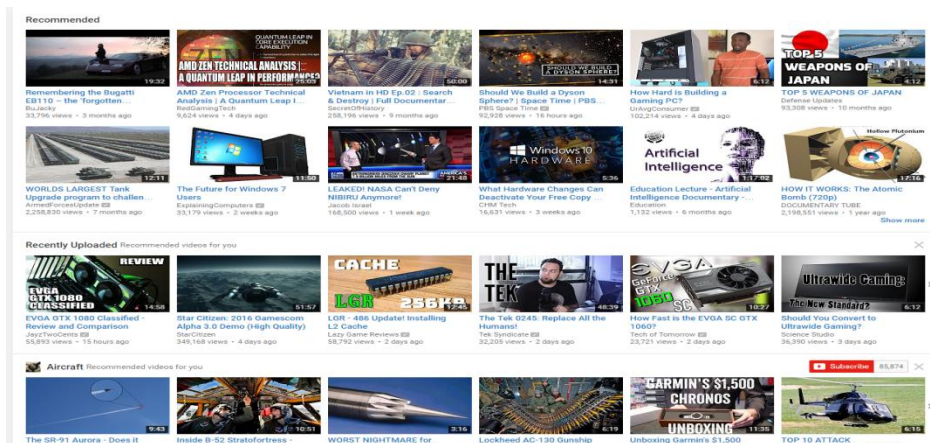
Οι αλγόριθμοι που αποκαλούμε Recommender Systems(συστήματα συστάσεων **RSs**) χρησιμοποιούν διαφορετικές μεθόδους πρόβλεψης και λήψης αποφάσεων, με την τάση να είναι ο συνδυασμός πολλών μεθόδων σε ένα σύστημα, όσον αφορά εταιρίες με μεγάλες βάσεις δεδομένων. Για την καλύτερη κατανόηση του θέματος θα αναφερόμαστε με τον όρο "item" σε ότι μπορεί να προταθεί σε ένα χρήστη "user" και η βαθμολογία ενός User για ένα Item : "Rating".

Η λογική πίσω από τα **RSs** είναι η ανάγκη να βοηθηθεί ένας χρήστης στην επιλογή ενός αντικειμένου, μιας ταινίας ή μιας υπηρεσίας ανάμεσα σε μια πληθώρα παρόμοιων τεμαχίων τι στιγμή που ο ίδιος δεν έχει την επαρκή γνώση, ικανότητα ή χρόνο να ερευνήσει μόνος του για μία νέα επιλογή. Για παράδειγμα ας δούμε το site "NETFLIX". Κάθε χρήστης σύμφωνα με τις προσωπικές του προτιμήσεις λαμβάνει εξειδικευμένες για τον ίδιο προτάσεις ώστε να επιλέξει εύκολα την επόμενη ταινία που θα παρακολουθήσει. Υπάρχουν βέβαια και πιο απλοί τύποι προτάσεων οι οποίοι δεν παρέχουν προσωποποιημένες προτάσεις. Τέτοιου είδους αλγόριθμοι είναι πολύ πιο εύκολο να δημιουργηθούν αλλά σε καμία περίπτωση δεν αγγίζουν την ακρίβεια των προσωποποιημένων, και δεν θα αναφερθούμε εκτεταμένα σε τέτοιες μορφές.

Η ταχεία ανάπτυξη του τομέα ήρθε από την παρατήρηση ότι λειτουργούμε και επιλέγουμε προϊόντα σύμφωνα με το τι έχει προταθεί σε εμάς από τρίτους, αν μπορούσε λοιπόν ένα ηλεκτρονικό κατάστημα να προσφέρει μια τέτοια υπηρεσία θα ήθελε σίγουρα να την αναπτύξει προς τέρψη των πελατών και επισκεπτών του ηλεκτρονικού της καταστήματος. Οι αλγόριθμοι αυτοί συνήθως εκμεταλλεύονται τα στοιχεία άλλων χρηστών πολλές φορές όχι μόνο το τί επέλεξαν αλλά και συγκρίνοντας τα προσωπικά τους στοιχεία. Ίσως προκύπτει λοιπόν το θέμα της ιδιωτικότητας, αλλά παρατηρώντας τα γεγονότα φαίνεται πως οι περισσότεροι προτιμούν την άνεση που προσφέρουν αυτά τα εργαλεία αδιαφορώντας για

τους όρους χρήσης που επιβάλλει η εκάστοτε εταιρία, αλλά και οι ασφάλεια των δεδομένων δεν φαίνεται να είναι ανεπαρκής.

Όπως προείπαμε ο τομέας των συστημάτων συστάσεων παρουσιάζει ταχεία ανάπτυξη τα τελευταία χρόνια, καθώς η ζήτηση είναι μεγάλη και η ταχύτητα αλλά και οι καινοτομίες στο πεδίο των υπολογιστών επιτρέπει σε οργανισμούς και υπηρεσίες να διαθέτουν την απαραίτητη υπολογιστική ισχύ για την υλική υποστήριξη των πολύ εξελιγμένων και ολοένα και πιο απαιτητικών αλγορίθμων. Το γεγονός αυτό είναι φανερό καθώς υπάρχει πληθώρα άρθρων στο διαδίκτυο που ερευνούν το συγκεκριμένο θέμα, με αποκορύφωμα ίσος τον διαγωνισμό που είχε διοργανώσει η υπηρεσία τηλεοπτικών παροχών **NETFLIX** στις 2 Οκτωβρίου 2006, με την πρώτη ομάδα να παρουσιάζει αποτελέσματα μόλις στις 8 του ίδιου μήνα. Αντίστοιχα και το site **YOUTUBE** χρησιμοποιεί την δική του μηχανή η οποία εξελίσσετε αποκλειστικά εν οίκo. Αυτό που όμως ίσως δεν συνειδητοποιούμε εύκολα είναι το γεγονός ότι και απλούστερες τέτοιες μηχανές λειτουργούν σε site πωλήσεων όπως **AMAZON**, **E-BAY**, **ALIEXPRESS** και άλλα. Γίνετε λοιπόν αντιληπτό ότι το εύρος και η πληθώρα συστημάτων συστάσεων μας αναγκάζει να τα κατηγοριοποιήσουμε ανάλογα με τον τρόπο που ενεργούν.



1) Recommendations από το YouTube

Fashion

Electronics

Watches

Health & Beauty

Home & Garden

All Deals



eBay deals

BEST DEALS OF THE WEEK

From trusted global sellers
with shipping to your country

13% OFF

Monoprice 5457
3ft USB 2.0 A
Male to Micro...

EUR 2.75
13% OFF

30% OFF

5FT 1080p HDMI
to Micro HDMI
Male Adapter...

EUR 6.17
30% OFF

56% OFF

5pcs Neon Rope
Light 2 Wire T
Splice Connect...

EUR 6.97
56% OFF

60% OFF

Fosmon 1ft 3ft 6ft
Type C (USB-C)
to Type A (US...

1FT
3FT
6FT
EUR 5.29
60% OFF

SHOP NOW

2) Εδω βλέπουμε ένα μέρος από προωθητικό mail του eBay μετά από αγορές καλωδίων για ηλεκτρονικούς υπολογιστές.



3) Ο αλγόριθμος του Netflix με τις προτάσεις του ταξινομημένες σε κατηγορίες, και με τους πιθανότερους τίτλους να εμφανίζονται από τα αριστερά προς τα δεξιά.

Οι αλγόριθμοι που αποκαλούμε Recommender Systems(συστήματα συστάσεων **RSs**) χρησιμοποιούν διαφορετικές μεθόδους πρόβλεψης και λήψης αποφάσεων, με την τάση να είναι ο συνδυασμός πολλών μεθόδων σε ένα σύστημα, όσον αφορά εταιρίες με μεγάλες βάσεις δεδομένων. Για την καλύτερη κατανόηση του θέματος θα αναφερόμαστε με τον όρο "item" σε ότι μπορεί να προταθεί σε ένα χρήστη "user" και η βαθμολογία ενός User για ένα Item : "Rating".

Η λογική πίσω από τα **RSs** είναι η ανάγκη να βοηθηθεί ένας χρήστης στην επιλογή ενός αντικειμένου, μιας ταινίας ή μιας υπηρεσίας ανάμεσα σε μια πληθώρα παρόμοιων τεμαχίων τι στιγμή που ο ίδιος δεν έχει την επαρκή γνώση ,ικανότητα ή χρόνο να ερευνήσει μόνος του για μία νέα επιλογή. Για παράδειγμα ας δούμε το site "NETFLIX". Κάθε χρήστης σύμφωνα με τις προσωπικές του προτιμήσεις λαμβάνει εξειδικευμένες για τον ίδιο προτάσεις ώστε να επιλέξει εύκολα την επόμενη ταινία που θα παρακολουθήσει. Υπάρχουν βέβαια και ποιο απλοί τύποι προτάσεων οι οποίοι δεν παρέχουν προσωποποιημένες προτάσεις. Τέτοιου είδους αλγόριθμοι

είναι πολύ πιο εύκολο να δημιουργηθούν αλλά σε καμία περίπτωση δεν αγγίζουν την ακρίβεια των προσωποποιημένων, και δεν θα αναφερθούμε εκτεταμένα σε τέτοιες μορφές.

Η ταχεία ανάπτυξη του τομέα ήρθε από την παρατήρηση ότι λειτουργούμε και επιλέγουμε προϊόντα σύμφωνα με το τι έχει προταθεί σε εμάς από τρίτους, αν μπορούσε λοιπόν ένα ηλεκτρονικό κατάστημα να προσφέρει μια τέτοια υπηρεσία θα ήθελε σίγουρα να την αναπτύξει προς τέρψη των πελατών και επισκεπτών του ηλεκτρονικού της καταστήματος. Οι αλγόριθμοι αυτοί συνήθως εκμεταλλεύονται τα στοιχεία άλλων χρηστών πολλές φορές όχι μόνο το τί επέλεξαν αλλά και συγκρίνοντας τα προσωπικά τους στοιχεία. Ίσως προκύπτει λοιπόν το θέμα της ιδιωτικότητας, αλλά παρατηρώντας τα γεγονότα φαίνεται πως οι περισσότεροι προτιμούν την άνεση που προσφέρουν αυτά τα εργαλεία αδιαφορώντας για τους όρους χρήσης που επιβάλει η εκάστοτε εταιρία, αλλά και οι ασφάλεια των δεδομένων δεν φαίνεται να είναι ανεπαρκής.

Στο κεφάλαιο αυτό θα δούμε τι προβλήματα μπορεί να αντιμετωπίσει κανείς, πως κατατάσσονται τα Recommender systems, και τι εργαλεία μπορεί να είναι χρήσιμα σε ένα σχεδιαστή τέτοιων συστημάτων.

Πιο αναλυτικά θα δούμε:

- **Κλασικά Προβλήματα**
- **Κατάταξη βάση προσωποποίησης**
- **Κατάταξη βάση μεθόδου**
- **Μαθηματικά εργαλεία**
- **Χρήσιμες μετρικές για έλεγχο απόδοσης**

1.1 Κλασικά Προβλήματα

Τα Recommender systems παρουσιάζουν κάποια προβλήματα άλλα που αφορούν την διαχείριση δεδομένων ή την απουσία αυτών, και άλλα σχετικά με την απόδοση τους. Εδώ θα δούμε τα διαχρονικά προβλήματα που μπορεί να αντιμετωπίσει κάποιος. Αυτά επιγραμματικά είναι:

1. **Cold-start problems**
2. **Diversity**
3. **Scalability**
4. **Non-intrusiveness**

- **Cold-Start problems:** Αυτός ο όρος αναφέρεται στις περιπτώσεις που έχουμε εισαγωγή ενός νέου item ή user στο σύστημα. Το πρόβλημα που προκύπτει είναι η έλλειψη δεδομένων για το αντικείμενο ή χρήστη με συνέπεια, ο αλγόριθμος να μην είναι σε θέση να προτείνει με καλή ακρίβεια καθώς δεν είναι δυνατό να το κατατάξει σε μία ομάδα με κοινά χαρακτηριστικά. Όταν το πρόβλημα αφορά νέους χρήστες υπάρχει η δυνατότητα να ζητήσουμε από το χρήστη να απαντήσει σε ερωτήσεις ώστε να χτιστεί το προφίλ του ή ακόμα πιο βολικά πλέον να πάρουμε στοιχεία από φίλους μέσω κοινωνικών δικτύων. Ακόμα αυτό μπορεί να επιτευχθεί αναζητώντας τα Cookies που είναι αποθηκευμένα στον προσωπικό του υπολογιστή. Για τα items όμως οι επιλογές είναι διαφορετικές. Αφού συνήθως μπορούμε να έχουμε πληθώρα χαρακτηριστικών είναι αρκετά εύκολο να υπολογιστούν ομοιότητες ή να καταχωρηθούν σε κλάσεις ομοίων τεμαχίων. Επίσης ο εμπλουτισμός των δεδομένων που επιλέγουμε να κάνουμε, τουλάχιστον στην ιδανική του μορφή, θα μπορούσε να δώσει μια εξίσου καλή λύση με το πλεονέκτημα της πλήρους αυτοματοποίησης.
- **Diversity:** Όπως στα περισσότερα προβλήματα έτσι και τα **RSs** πρακτικά είναι ένα trade-off ανάμεσα στην ακρίβεια και την ποικιλία. Για παράδειγμα ένας χρήστης απολάμβανε

για μεγάλο χρονικό διάστημα ταινίες τρόμου, αλλά πλέον προτιμάει ταινίες δράσεις. Ένα πολύ αυστηρό σύστημα που για καιρό πρότεινε ταινίες τρόμου είναι πλέον δύσκολο να αλλάξει το προφίλ του χρήστη. Φυσικά για αυτό το θέμα υπάρχουν λύσεις, όπως να δίνεται βάρος στις πιο πρόσφατες ταινίες που είδε. Μία άλλη καλή τακτική είναι η ανάδευση των n-καλύτερων προτάσεων ώστε να αυξάνεται η ποικιλία των ταινιών.

- **Scalability:** Ένα RS δεν έχει νόημα να δημιουργηθεί απλά για την παρούσα βάση δεδομένων που ζητά την εφαρμογή. Αντίθετα πρέπει να σχεδιαστεί με τρόπο τέτοιο ώστε να μπορεί να αποδίδει μ σωστά, δηλαδή σε λογικούς χρόνους και με μικρά λάθη όσο αυξάνονται οι διαστάσεις των δεδομένων.
- **Non-intrusiveness:** Πολλά RSs έχουν την απαίτηση για αναπληρωφόρηση από τον χρήστη και μεγάλη ανάμειξη του χρήστη. Παρόλο που πλέον οι περισσότεροι χρήστες πλέον δεν έχουν πρόβλημα με την καταγραφή στοιχείων για αυτούς από τις μηχανές, λίγοι είναι αυτοί που έχουν την διάθεση να ασχοληθούν και να βαθμολογήσουν αντικείμενα. Υπάρχουν μη παρεμβατικοί μέθοδοι για αυτό το πρόβλημα όπως για παράδειγμα πόση ώρα παρέμεινε σε ένα trailer μιας ταινίας. Τέτοιες λύσεις φυσικά δεν έχουν την απόδοση ενός συστήματος που τροφοδοτείτε απευθείας από τον χρήστη. Μια πιο αποδοτική σύμβαση είναι να οριστεί το ελάχιστο ποσό ratings που μπορεί να ζητηθεί από τον χρήστη ώστε να χτιστεί ένα ικανοποιητικό προφίλ για αυτό το χρήστη.

1.2 Κατάταξη βάση προσωποποίησης

Για να μπορεί κάποιος να περιγράψει έναν αλγόριθμο θα πρέπει να είναι σε θέση και να τους κατηγοριοποιεί. Η πρώτη κατηγοριοποίηση έχει να κάνει με το επίπεδο το οποίο γνωρίζει ένα σύστημα τους χρήστες του. Τα 4 επίπεδα ξεκινούν από μηδενική γνώση για τον χρήστη έως μια μακροχρόνια συλλογή δεδομένων σχετικά με αυτόν. Επιγραμματικά οι κατηγορίες αυτές είναι:

- Μη προσωποποιημένοι
- Δημογραφικοί
- Εφήμερου χαρακτήρα
- Μόνιμου χαρακτήρα

- **Μη προσωποποιημένοι**

Σε αυτή την κατηγορία εμπίπτουν μηχανές που δημιουργούν προτάσεις βάση γενικών τάσεων, δημοτικότητας, κρητικών και βαθμολογιών. Δε λαμβάνετε λοιπόν καθόλου υπ' όψιν ο χρήστης που θα λάβει τις προτάσεις και το αποτέλεσμα είναι το ίδιο για όλους τους χρήστες. Τέτοια συστήματα μπορεί να βρίσκουν εφαρμογή σε περιοδικά ή καταστήματα που η βάση πελατών και προϊόντων τους δεν είναι επαρκεί για να επενδύσουν σε μία πιο ακριβή τεχνολογία.

- **Δημογραφικοί**

Με τον όρο δημογραφικοί εννοούμε την δυνατότητα του αλγόριθμου να προτείνει σε μεγάλες ομάδες πληθυσμού τα ίδια προϊόντα ή υπηρεσίες. Συνήθως μέσω των γνωστών μας πλέον cookies ή πληροφοριών που είναι συνδεδεμένες με την ηλεκτρονική μας υπογραφή μια ιστοσελίδα μπορεί να είναι ικανή να παραθέσει προϊόντα ανάλογα με την ηλικία, το φύλλο, την εποχή και άλλα κριτήρια βελτιώνοντας κατά πολύ την πιθανότητα μίας αγοράς/συναλλαγής σε σχέση με την προηγούμενη κατηγορία.

- **Εφήμερου χαρακτήρα**

Σε ένα τέτοιο σύστημα μπορεί να ζητηθούν από ένα χρήστη τα τρέχοντα ενδιαφέροντα του ώστε να προτείνει νέα ή δημοφιλή τεμάχια, μια τέτοια υλοποίηση είναι ιδιαίτερα χρήσιμη για παράδειγμα σε ένα κατάστημα δώρων όπου ένας συνδυασμός βαθμολογιών και νεότητας προϊόντος θα ήταν σχεδόν ιδανικός, ή αν εκμεταλλεύεται δεδομένα παλαιότερων προτιμήσεων, γίνεται αποκοπή από ένα χρονικό σημείο και πριν ή απλά αυξάνοντας τα βάρη των τελευταίων επιλογών δίνοντας έτσι την δυνατότητα να αξιολογηθούν αυτόματα οι τελευταίες προτιμήσεις με αποτέλεσμα να είναι πιο ευέλικτο το σύστημα και να προσαρμόζεται πιο εύκολα στα νέα ενδιαφέροντα του χρήστη.

- **Μόνιμου χαρακτήρα**

Στην κατηγορία αυτή είναι η απαραίτητη η διαχρονική συλλογή δεδομένων για κάθε χρήση και η ανάλυση των μακροπρόθεσμων προτιμήσεων του. Αντίθετα λοιπόν με την άνω

κατηγορία το προφίλ του χρήστη είναι αρκετά άκαμπτη στο να αλλάξει χαρακτηριστικά μετά από μεγάλης διάρκειας όμοιων προτιμήσεων και απαιτεί πολύ χρόνο συλλογής δεδομένων για να είναι αποδοτική, παρόλα αυτά εφαρμογές όπως συστήματα σχετικά με ταινίες και βιβλία συνήθως αξιοποιούν καλά τέτοια είδη αλλά συνήθως, είναι επιβεβλημένος ένας συνδυασμό των δύο τακτικών, εκμεταλλευόμενοι συνήθως τα μοντέλα μόνιμου χαρακτήρα με μεγαλύτερο βάρος στους παλαιότερους χρήστες και μικρότερο στους νέους.

Personalization Level

Generic/Non-Personalized: Οι προτάσεις είναι οι ίδιες για όλους τους Χρήστες

Demographic: Οι προτάσεις αλλάζουν ανά δημογραφικό group

Ephemeral: Οι προτάσεις παράγονται μόνο για τα τρέχοντα ενδιαφέροντα

Persistent: Οι προτάσεις δημιουργούνται βάση διαχρονικών προτιμήσεων

1.3 Κατάταξη βάση μεθόδου

Πέρα από την σχέση του χρήστη με το σύστημα όμως πρέπει να έχουμε και μία μέθοδο κατηγοριοποίησης με βάση τον τρόπο λειτουργίας του αλγορίθμου αυτού. Έχει λοιπόν να κάνει με τα διαθέσιμα δεδομένα αλλά κυρίως με τον τρόπο επεξεργασίας αυτών για την εξαγωγή συστάσεων. Στον παρακάτω πίνακα βλέπουμε τις κατηγορίες και υποκατηγορίες που υπάρχουν. Και αμέσως μετά ακολουθεί μια περιγραφή της κάθε μίας.

Methods	
○	Non-Personalized Summary statistics
○	Content-Based Filtering
	Information Filtering
	Knowledge-based
○	Colaborative filtering
	User based or Item based
	Dimensionality reduction
	Hybrid recommenders

○ **Non-Personalized Summary statistics**

Σε αυτή την κατηγορία έχουμε τα μη προσωποποιημένα **RSs**. Χρησιμοποιούν έννοιες όπως Best-seller, περισσότερο διάσημο, καυτό θέμα κ.α. Από εξωτερικές πηγές όπως μηχανές αναζήτησης εκτός καταστήματος για παράδειγμα στατιστικά από την Google. Η αποτελέσματα από τα μέλη της ενεργής κοινότητας δηλαδή περισσότερα αγαπητό αντικείμενο. Τέτοια παραδείγματα μπορούμε να δούμε σε εφαρμογές όπως Tripadvisor ή Billboard. Η επεξεργασία λοιπόν που βαρύνει τον επεξεργαστή είναι ελάχιστη γιατί έχουμε μια μονομερή συγκέντρωση

στατιστικών στοιχείων και δεν απαιτούνται εφαρμογές προβλέψεων που είναι και οι πλέον απαιτητικές.

- **Content-Based Filtering**

- Information Filtering**

Το φιλτράρισμα πληροφοριών μπορεί πολύ απλά να προκύψει ενημερώνοντας την βάση δεδομένων μας με στοιχεία για τα **items** και κάνοντας στατιστικές αναλύσεις στον πίνακα που δημιουργούμε να καθορίσουμε αυτόματα τις προτιμήσεις του κάθε χρήστη. Θα φέρουμε το παράδειγμα ενός χρήστη που προτιμάει διαχρονικά ταινίες δράσεις. Ο χρήστης λοιπόν αυτός αναμένετε να βαθμολογήσει υψηλότερα και συχνότερα τέτοια ήδη ταινιών με αποτέλεσμα με κατάλληλη εκπαίδευση-χρήση αλγορίθμων το σύστημα να αναγνωρίσει το ενδιαφέρον αυτό και να παράγει σχετικές προτάσεις. Το ίδιο φυσικά μπορεί να συμβεί σε κάθε ιδιότητα των **items** που είναι διαθέσιμα στο σύστημα.

- Knowledge-based**

Το φιλτράρισμα με χρήση μιας βάσης γνώσης προϋποθέτει εναλλακτικούς τρόπους ανίχνευσης των ενδιαφερόντων του χρήστη. Κυρίως η μέθοδος είναι η απλή λήψη πληροφοριών απευθείας από τον ίδιο το χρήστη. Αν και πλέον μπορεί να μας ξενίζει να παρέχουμε μόνοι μας πληροφορίες καθώς έχουμε επαναπαυθεί στην αυτοματοποίηση των συστημάτων, εδώ υπάρχει το βασικό πλεονέκτημα της μείωσης έως και εξάλειψης του προβλήματος **Cold-start** όπως περιγράφετε στη βιβλιογραφία. Δηλαδή την αδυναμία των πλήρως αυτόνομων μηχανών να παρέχουν ακριβείς προτάσεις σε νέους χρήστες.

- **Colaborative filtering**

- User based**

Η δημιουργία μιας γειτονιάς χρηστών με παρόμοιες προτιμήσεις ή η εύρεση χρηστών που θα μπορούσε ο εκάστοτε χρήστης να εμπιστευτεί για τις μελλοντικές επιλογές του

χαρακτηρίζεται ως συνεργατικό φιλτράρισμα χρήστη με χρήστη. Ο σκοπός είναι να χρησιμοποιήσουμε την άποψη των βέλτιστα όμοιων προς τον χρήστη "κοντινών του ομοιών" ελπίζοντας σε μία πολύ καλή αποτύπωση των πιθανών επιλογών από το σύστημα είτε επιλέγοντας χρήστες από το κοινωνικό του περίγυρο. Ένα τέτοιου είδους σύστημα μπορεί να είναι πολύ αποδοτικό σε καταστάσεις όπου ο χρήστης υπό εξέταση, έχει ήδη ένα ικανοποιητικά μεγάλο αριθμό βαθμολογιών **Ratings** ώστε το μηχάνημα να βρει την βέλτιστη γειτονιά. Τέλος είναι η πιο χρονοβόρα υπολογίστηκα τεχνική αλλά παράλληλα και η συνηθέστερα πιο αποδοτική.

-Item based

Εφαρμόζοντας την τεχνική τεμάχιο-τεμάχιο (**item-item**) η τακτική είναι η ακόλουθη: προϋπολογίζετε η ομοιότητα των τεμαχίων μέσω των βαθμολογιών που έλαβαν (**ratings**) ή και μέσω δεδομένων χαρακτηριστικών τους. Εν συνεχεία επιλέγονται όμοια με αυτό για προβολή στον χρήστη. Μια άλλη οπτική μπορεί να δημιουργεί προτάσεις μέσω κοινών αγορών από χρήστες. Για παράδειγμα ο χρήστης επιλέγει ένα αντικείμενο. Βρίσκοντας στην ουσία τα ποιο δημοφιλή αντικείμενα ανάμεσα στους χρήστες που έχουν επίσης επιλέξει το ίδιο αντικείμενο η μηχανή προβάλλει τα αντικείμενα με σειρά δημοτικότητας παλαιότητας ή ένα συνδυασμό σχετικών χαρακτηριστικών. Είναι γενικά παραδεκτό από την βιβλιογραφία ότι ενώ η μέθοδος **user-user** δημιουργούν τις πιο εύστοχες προβλέψεις-επιλογές, η μέθοδος **item-item** έπεται με μικρή διαφορά αλλά κερδίζει δυσανάλογα περισσότερη ταχύτητα.

-Dimensionality reduction

Όπως εύκολα αντιλαμβανόμαστε από τον τίτλο η τεχνική είναι η μείωση των διαστάσεων του πίνακα των **ratings** ανάμεσα σε **users** και **items**. Με αυτή την μέθοδο μπορούμε να επιταχύνουμε την δημιουργία των αποτελεσμάτων μας. Επίσης είναι μια καλή επιλογή για ένα σύστημα γενικής χρήσης καθώς δεν απαιτούν ρύθμιση των παραμέτρων και μπορούν να εφαρμοστούν εύκολα σε πολλαπλές περιπτώσεις. Οι μαθηματικές μέθοδοι που

βρίσκουν εφαρμογή είναι: **Singular value decompositon, Stochastic gradient descent, Alternating least squares, Principal Component Analysis.**

-Hybrid recommenders

Πολλές φορές είναι μια καλή πρακτική ο συνδυασμός των μεθόδων είτε συνδυάζοντας προβλέψεις βαθμολογιών **ratings** με:

- 1** σταθμισμένους μέσους : συνδυάζοντας τα αποτελέσματα υπολογίζοντας τους σταθμισμένους μέσους των προβλέψεων
- 2** επιλογή ενός : Επιλέγοντας τον κατάλληλο αλγόριθμο για κάθε περίπτωση ανάλογα τις ιδιότητες της απαιτούμενης πρόβλεψης
- 3** Μικτή επιλογή : Παρουσιάζοντας προβλέψεις από πολλούς αλγόριθμους
- 4** Αύξηση χαρακτηριστικών : Οι αλγόριθμοι χρησιμοποιούνται εν σειρά παράγοντας ο καθένας τις παραμέτρους εισαγωγής του επόμενου
- 5** αλληλουχία : Οι αλγόριθμοι με σειρά προτεραιότητας δημιουργούν προτάσεις με τους τελευταίους να εισάγουν τις δικές τους κάπου ανάμεσα στην υψηλής προτεραιότητας
- 6** Συνδυασμός ιδιοτήτων : συνδυάζονται διάφορα χαρακτηριστικά από διάφορες κατηγορίες σε ένα ενιαίο συνήθως πιο περίπλοκο συνδυασμό σε ένα αλγόριθμο
- 7** Μετα-επίπεδο : Οι αλγόριθμοι χρησιμοποιούνται εν σειρά παράγοντας μοντέλα για επεξεργασία από τον επόμενο, παρόμοια με την μέθοδο αύξησης χαρακτηριστικών.

1.4 Μαθηματικά εργαλεία

Εδώ θα αναφερθούμε επιγραμματικά σε μαθηματικές μεθόδους που μπορεί κανείς να συναντήσει στην βιβλιογραφία, σχετικά με το τι έχει χρησιμοποιηθεί στα Recommender systems.

- **Μείωση διαστάσεων:** Είναι μια τακτική που ακολουθείτε συχνά καθώς οι πίνακες σε αυτές τι εφαρμογές είναι πολύ αραιοί.
- **Principal component analysis:** Η στατιστική διαδικασία χρησιμοποιεί τον ορθογώνιο μετασχηματισμό ώστε να δημιουργηθούν κύρια μέρη σε ένα νέο σύστημα συντεταγμένων τοποθετώντας το μέρη σε φθίνουσα σειρά βάση της διασποράς τους. Ο τρόπος εφαρμογής για ταχύτερη επεξεργασία χρησιμοποιεί την μέθοδο της συνδιασποράς. Υπολογίζεται ο εμπειρικός μέσος οι αποκλίσεις από τους μέσους για κάθε γραμμή του πίνακα. Στη συνέχεια υπολογίζεται ο πίνακας συνδιασποράς και οι ιδιοτιμές και τα ιδιοδιανύσματα, και ο μετασχηματισμός του πίνακα.
- **Singular Value Decomposition:** Η ανάλυση πίνακα σε ιδιάζουσες τιμές είναι η μέθοδος της αλγεβρας που σκοπό έχει την ανάλυση του πίνακα σε ιδιάζουσες τιμές. Για την ακρίβεια αναζητούμε την ακόλουθη παραγοντοποίηση: $M = U \Sigma V^*$

όπου U είναι ένας $m \times m$ πραγματικός ή μιγαδικός ορθομοναδιαίος πίνακας, Σ ένας $m \times n$ ορθογώνιος διαγώνιος πίνακας με μη αρνητικές τιμές στην διαγώνιο και V^* (ο συζυγής ανάστροφος του V , ή απλά ο ανάστροφος του V αν ο V είναι πραγματικός) ένας $n \times n$ πραγματικός ή μιγαδικός ορθομοναδιαίος πίνακας. Τα διαγώνια στοιχεία $\Sigma_{i,i}$ του Σ είναι γνωστά ως ιδιάζουσες τιμές του M . Οι m στήλες του U και οι n στήλες του V ονομάζονται αριστερά-ιδιάζοντα διανύσματα και δεξιά-ιδιάζοντα διανύσματα του M , αντίστοιχα

- **Clustering:** Συσταδοποίηση ή ομαδοποίηση είναι η διαδικασία κατά την οποία χωρίζουμε τα δεδομένα μας σε ομάδες χαρακτηριστικών ή ατόμων που μπορούν λογικά να καταταγούν σε διαφορετικές ομάδες.
- **K-Nearest Neighbours:** Πρακτικά δημιουργούμε έναν αλγόριθμο που για κάθε χρήστη βρίσκει τους k -κοντινότερους γείτονες με κάποια μέτρα ομοιότητας. Η μέθοδος αυτή παρουσιάζει πολύ καλά αποτελέσματα και μάλιστα σε πολύ μικρό χρόνο. Είναι η πλέον αποδεκτή μέθοδος σε εφαρμογές **Collaborative filtering**.
- **k-means:** Παρόμοια μέθοδος που λειτουργεί επαναληπτικά δημιουργώντας ομάδες μέχρι να ικανοποιηθεί το κριτήριο που μπορεί να είναι η επιθυμητή ακρίβεια ή ένας σταθερός αριθμός επαναλήψεων.
- **Artificial Neural Networks:** Τα νευρωνικά δίκτυα με βάση την συμπεριφορά του ανθρώπινου εγκεφάλου χρησιμοποιούν νευρώνες σε επίπεδα με κάθε επίπεδο να παράγει τα δεδομένα για τις επόμενες στρώσεις νευρώνων. Το βασικό τους πλεονέκτημα είναι ότι μπορούν να χειριστούν σχέσεις πέραν των γραμμικών. Το πλήθος των νευρών και των επιπέδων τους πρέπει να είναι ο ιδανικός. Πρέπει να εφαρμόζετε προσεκτικά καθώς το πρόβλημα του over-fitting μπορεί να παράγει έναν μη ευέλικτο αλγόριθμο.
- **Support Vector Machine:** Ο σκοπός του αλγορίθμου είναι να υπολογίσει υπερεπίπεδα ώστε να χωρίσει τον χώρο των τιμών με βάση την μεγιστοποίηση της απόστασης των τιμών από το υπερεπίπεδο. Η ιδέα είναι οι υπόχωροι να είναι όσο το δυνατό καλύτερα διαχωρισμένοι ώστε να έχουμε ένα βέλτιστο διαχωρισμό.
- **Multinomial logistic regression:** Σε αλγορίθμους κυρίως της κατηγορίας των μικτών με συνδυασμό ιδιοτήτων που ενσωματώνουν δηλαδή πολλά χαρακτηριστικά σε ένα μοντέλο.
- **Other collaborative filtering:** Οι αλγόριθμοι εδώ μπορούν να υπολογίσουν προβλεπόμενα **ratings** με βάση τις υπάρχουσες τιμές στον πίνακα μας. Κάποιες μέθοδοι που αναφέρθηκαν ανωτέρω θα αναφερθούν και εδώ αναλύοντας περαιτέρω σημαντικά στοιχεία τους.

- **Random forests:** Τυχαία δάση. Δημιουργούνται όπως είναι αντιληπτό ένα πλήθος δέντρων επιλογών (decision trees) τα οποία δουλεύουν παράλληλα και ανεξάρτητα μεταξύ τους. Τα πλεονεκτήματα που προσφέρουν είναι πολύ καλές προβλέψεις με μικρές αποκλίσεις, ευελιξία στην βελτιστοποίηση, χωρίς εμφανή κίνδυνο **over-fitting** λόγω της "τυχαίας" φύσης τους, καλούς επεξεργαστικούς χρόνους αναλογιζόμενοι την απόδοση, πολύ εύκολη εφαρμογή σε πληθώρα εφαρμογών με εμπλουτισμένα στοιχεία (πέραν δηλαδή του στοιχειώδους collaborative filtering). Η λειτουργία τους περιγράφεται ως ακολούθως: 1) Επιλογή αριθμού παράλληλα εκτελέσιμων δέντρων. 2) Επιλογή πλήθους μεταβλητών για κάθε δέντρο 3) Επιλογή ποσοστού δείγματος για την δημιουργία του μοντέλου (επιταχύνει την διαδικασία με μικρό κόστος αποκλίσεων σε μεγάλα datasets) 4) Βελτιστοποίηση του διαχωρισμού των μεταβλητών (αυτοματοποιημένη διαδικασία) 5) Δημιουργία δέντρων με βάση χρήση στατιστικών μεθόδων όπως **linear regression, logistic regression, poisson regression**, και άλλα. 6) Συγκέντρωση αποτελεσμάτων με ψηφοφορία των δέντρων.
- **Artificial neural networks:** Τα νευρωνικά δίκτυα μπορούν και αυτά να χρησιμοποιηθούν ανάλογα. Να χτιστούν δηλαδή με τρόπο τέτοιο ώστε να δημιουργούν ένα μοντέλο για κάθε γραμμή ή στήλη του πίνακα. Ένα άλλο πλεονέκτημα είναι η δυνατότητα να επεξεργαστούν παράλληλα και χαρακτηριστικά πέραν της ομοιότητας είτε σε ένα ενιαίο πλαίσιο είτε με ξεχωριστά cluster νευρώνων δίνοντας την δυνατότητα σε έμπειρους προγραμματιστές να κατασκευάσουν πολύ αποδοτικούς αλγόριθμους κατά περίπτωση.
- **K-Nearest Neighbours:** Μπορεί να αναφέραμε τον συγκεκριμένο αλγόριθμο στο τμήμα σχετικά με την ομαδοποίηση παρόλα αυτά ο τρόπος χρήσης δύσκολα θυμίζει την άνω μέθοδο. Ας δούμε λοιπόν πώς λειτουργεί. Υπολογίζονται τα μέτρα ομοιότητας (similarity measures) ανάμεσα στις γραμμές ή στήλες του πίνακα για user-user ή item-item αντίστοιχα. Κρατάμε τα k-κοντινότερα για κάθε γραμμή ή στήλη και χρησιμοποιώντας τα μέτρα βγάζουμε την πρόβλεψη ως σταθμισμένο μέσο των αντιστοίχων βαθμολογιών. Η κύρια διαφοροποίηση ανάμεσα στους αλγόριθμους τέτοιου είδους είναι, πέραν του μεγέθους της γειτονίας του χρήστη ή του αντικειμένου, το μέτρο της μεταξύ τους ομοιότητας που θα επιλεγεί. Η τακτική είναι η επιλογή ενός μέτρου απόστασης και η

μετατροπή σε μέτρο ομοιότητας σύμφωνα με τον τύπο: $sim(x, y) = \frac{1}{1+r_2(x, y)}$ ή κάποια άλλη κανονικοποίηση.

Κλασικές επιλογές είναι :

1. Η απόσταση **Hamming**, η απόσταση είναι ίση με των αριθμό των διαφορετικών στοιχείων ανάμεσα σε δύο δυαδικές ακολουθίες ίσου μήκους. Ένα παράδειγμα: πίνακας με χρήστες και ταινίες που παρακολούθησαν.

2. **Jaccard Distance**: Ορίζετε το σύνολο των διαφορετικών τιμών προς το μήκος των δυο διανυσμάτων. Πολύ γρήγορη μέθοδος όμοια με την απόσταση hamming σε βαθμολογίες με μεγάλες ακρίβειες συνήθως χάνει πληροφορία.

3. **Cosine similarity**: Συνημίτονο γωνίας διανυσμάτων που δίνεται από τον τύπο: $sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$

4. Η **Ευκλείδεια απόσταση**, Ορίζεται η ρίζα του αθροίσματος του τετραγώνου των διαφορών των αντιστοιχών τιμών δύο διανυσμάτων.

$$r_2(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}$$

5. Η **Minkowski**: $d_{Mk} = \sqrt[p]{\sum_{i=1}^d |P_i - Q_i|^p}$

με ίσως πιο συνήθη την υποκατηγορία της **City block**: $d_{cb} = \sum_{i=1}^d |P_i - Q_i|$

6. **Pearson's Correlation**: Το γνωστό μέτρο όπως χρησιμοποιείτε στην στατιστική: $corr(x, y) = \frac{Cov(x, y)}{\sigma_x \sigma_y}$

7. Πέρα από αυτά τα μέτρα ομοιότητάς τα οποία είναι από τα πλέον συνηθισμένα, υπάρχουν πάρα πολλά άλλα και η παράθεση τους δεν θα είχε νόημα. Για περισσότερα βιβλιογραφία [1].

1.5 Χρήσιμες μετρικές για έλεγχο απόδοσης

Για να ελέγξουμε την απόδοση του συστήματός μας είναι δόκιμο να χρησιμοποιούμε κάποιες μετρικές που έχει νόημα η εφαρμογή τους, και χρησιμοποιούνται διαχρονικά για την μέτρηση της απόδοσης ενός Recommender system. Αυτές είναι:

- RMSE
- MAE
- MSE
- Precision
- Recall
- F-measure

RMSE: Root square mean error or root mean square deviation (RMSD). Τετραγωνική ρίζα του μέσου των τετραγώνων των διαφορών. Κλασσικά χρησιμοποιείτε στη παλινδρόμηση αλλά και γενικά σε μοντέλα πρόβλεψης. Θα έλεγε κάνεις η πιο διάσημη μετρική. Το χαρακτηριστικό της είναι ότι τιμορεί σε μεγάλο βαθμό τις μεγάλες αποκλείσεις.

$$Rmse = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}$$

MAE: Mean Absolute Error. Τετραγωνική ρίζα του μέσου των απολύτων των διαφορών. Το μέτρο αυτό τιμωρεί περισσότερο της μικρές διαφορές (μικρότερες του 1). Πολλές φορές λοιπόν είναι καλό να υπολογίζονται και τα δυο μέτρα ώστε να έχουμε μια καλύτερη εικόνα για το αποτέλεσμα.

$$Rmse = \sqrt{\frac{\sum_{i=1}^n |y_i - x_i|}{n}}$$

MSE: Mean squared error. Μέσο τετράγωνο διαφορών. Σχεδόν το ίδιο με το RMSE.

$$Mse = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$$

Precision: Ακρίβεια, Δηλαδή το ποσοστό των σχετικών αποτελεσμάτων σε σχέση με τα ληφθέντα ή καλύτερα τα σωστά αποτελέσματα προς το σύνολο τους.

$$Precision = \frac{\{\sigmaχετικά\} \cap \{\lambdaηφθέντα\}}{\{\lambdaηφθέντα\}}$$

Recall: Ανάκληση, Το ποσοστό των ληφθέντων αποτελεσμάτων προς το επιθυμητό δηλαδή το πλήθος των σωστών αποτελεσμάτων προς το αναμενόμενο πλήθος.

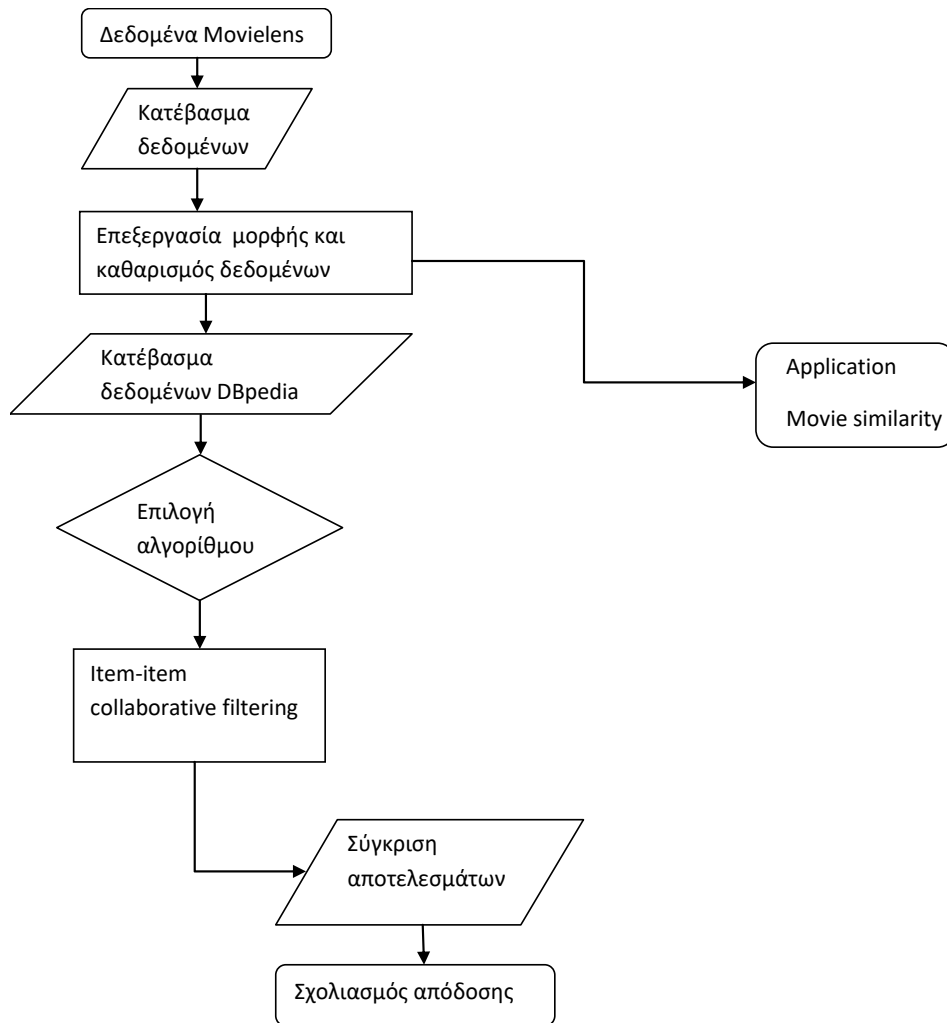
$$Recall = \frac{\{\sigmaχετικά\} \cap \{\lambdaηφθέντα\}}{\{\sigmaχετικά\}}$$

F-measure: Είναι ένας συνδυασμός των μετρικών precision και recall. Υπό μία σχετικά αφηρημένη έννοια ένας σταθμισμένος μέσος αυτών.

$$F = 2 \cdot \frac{Precision + recall}{Precision \cdot recall}$$

2. ΜΕΘΟΔΟΛΟΓΙΑ

Εδώ θα δούμε την μεθοδολογία που ακολουθήθηκε στην εργασία όπως φαίνεται από το διάγραμμα ροής.



4) Διάγραμμα ροής της εκπόνησης της εργασίας

Όπως φαίνεται στο διάγραμμα επιλέξαμε τα δεδομένα από το Movielens. Διαμορφώθηκαν κατάλληλα και σύμφωνα με αυτά κατεβάσαμε επιπλέον στοιχεία από την DBpedia. Ακολουθεί η επιλογή αλγορίθμου ή δημιουργία του και τέλος συγκρίνουμε την διαφορά απόδοσης χρησιμοποιώντας στοιχεία των ταινιών αλλά και χωρίς αυτά. Παράλληλα σχεδιάζουμε μια εφαρμογή που μας προτείνει ταινίες με βάση μια ταινία που μας αρέσει.

3. ΛΟΓΙΣΜΙΚΑ ΚΑΙ ΕΡΓΑΛΕΙΑ ΕΞΟΡΥΞΗΣ ΚΑΙ ΚΑΘΑΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ

Εδώ περιγράφουμε τα εργαλεία και που χρησιμοποιήθηκαν αυτά.

- Εξόρυξης (SPARQL)
- Καθαρισμού(Open Refine, R-Rstudio)
- Βιβλιοθήκες της R studio

3.1 Εξόρυξη δεδομένων

Linked data είναι ο όρος που περιγράφει μια μέθοδο δημοσιοποίησης δομημένων δεδομένων με τρόπο τέτοιο ώστε να είναι μηχαναγνώσιμα και συνδεδεμένα ώστε να αυξάνετε η χρηστική τους αξία. Οι τεχνολογίες που αξιοποιούνται σε αυτή τη μορφή δεδομένων το HTTP (Hypertext Transfer Protocol) και τα URIs (Uniform Resource Identifiers), επεκταμένα ώστε να υποστηρίζουν την μηχαναγνώσιμη φύση τους. Οι βάσεις πάνω στις οποίες στήνονται τα δεδομένα ονομάζονται οντολογίες. Είναι πρακτικά ένα δίκτυο εννοιών που συνδέονται βάση ιδιοτήτων. Επίσης τα δεδομένα που δημοσιεύονται υπό το πρότυπο αυτό, λαμβάνουν τη μέγιστη δυνατή βαθμολογία, στη κλίμακα που έχει προτείνει ο Tim Berners Lee.

Η δομή των διασυνδεδεμένων δεδομένων μοντελοποιείται σε μορφή τριπλετών. Μια λογική και εύκολα κατανοητή μορφή με "Υποκείμενο, Κατηγορούμενο, Αντικείμενο". Για την αναζήτηση και ανάκτηση των δεδομένων έχει δημιουργηθεί η γλώσσα SPARQL. Σύμφωνα με την λογική των RDF δεδομένων έτσι και η SPARQL έχει στηθεί και χρησιμοποιείται με ανάλογο ύφος.

SPARQL (SPARQL Protocol and RDF Query Language) αυτή η πολύ σημαντική τεχνολογία του web είναι κατά βάση μία γλώσσα που καθιστά δυνατή τη πλοήγηση σε δεδομένα τύπου Rdf. Η SPARQL μπορεί να ανάκτηση και να διαχειρίζεται τα δεδομένα στους γράφους RDF. Στην ουσία θέτονται ερωτήματα σε SPARQL endpoint και μετά λαμβάνει τις απαντήσεις, σε διάφορες μορφές αποτελεσμάτων.

Μέσω της SPARQL μπορούν να απαντηθούν τεσσάρων ειδών ερωτήσεις. Πρώτο είδος τέτοιας ερώτησης είναι το SELECT, τα δεδομένα που λαμβάνουμε από το endpoint είναι μορφής πίνακα. Δεύτερον στο CONSTRUCT τα αποτελέσματα παίρνουν την μορφή RDF. Τρίτον, στο ASK τα αποτελέσματα που έχουμε είναι γραμμένα ως αληθή ή ψευδή. Και τέλος, στο DESCRIBE δίνει ένα RDF στο οποίο αναγράφονται οι πηγές στις οποίες βρέθηκαν.

Προκειμένου τα αποτελέσματα να είναι πιο συγκεκριμένα για κάθε ερώτημα που τίθεται, χρησιμοποιείται η εντολή WHERE, λόγω της αναζήτησης με την χρήση των patterns. Η μορφή της ερώτησης για να γράφει παίρνει την το σύμβολο "?" ή "\$" , όμως στην περίπτωση της χρήσης prefixes τα σύμβολα εισάγονται μέσα στο ερώτημα, με το τρόπο αυτό δεν χρησιμοποιούνται ολόκληρα τα namespace της οντολογίας.

Τα στάδια της SPARQL ερώτησής είναι πέντε. Πρώτο στάδιο, είναι οι δηλώσεις των προθεμάτων, δηλαδή οι συντομεύσεις που ορίζονται για τα URIs. Δεύτερο στάδιο είναι ο ορισμός της βάσης των δεδομένων, που δείχνει ποια δεδομένα θα δέχονται ερωτήματα. Τρίτο στάδιο, ο ορισμός των αποτελεσμάτων, είναι οι πληροφορίες οι οποίες επιστρέφουν από το ερώτημα που τίθεται. Τέταρτο στάδιο είναι ο κορμός της ερώτησης, είναι το κυριότερο κομμάτι και αυτό επειδή θέτεται στο κομμάτι αυτό η ερώτηση του χρήστη βασισμένο στην βάση των δεδομένων. Πέμπτο στάδιο και το τελικό είναι οι τροποποιητές του ερωτήματος, στην οποία εντάσσονται οι εντολές οι οποίες αποκόπτουν, ταξινομούν και τακτοποιούν τα αποτελέσματα.

Τις εντολές που προαναφέραμε, στην χρήση της SPARQL, θα τις δούμε τώρα πιο αναλυτικά:

- PREFIX: συνδέεται με τρία ή τέσσερα γράμματα και ένα URI για το οποίο και δημιουργεί συντόμευση. Η αναζήτηση από τον χρήστη γίνεται από τα γράμματα και όχι από τον σύνδεσμο.
- FROM :πάντοτε είναι συνδεδεμένο με ένα URI στο οποίο περιέχονται οι βάση δεδομένων και σύμφωνα με αυτό δημιουργείτε το ερώτημα. Το endpoint της DBpedia, πλέον παραλείπεται , εφόσον η Wikipedia αποτελεί την βάση δεδομένων.
- SELECT : την συνοδεύουν οι μεταβλητές του πίνακα αποτελεσμάτων. Στην εντολή SELECT DISTINCT διαγράφονται οι διπλοεγγραφές οι οποίες μπορούν να προκύψουν έως αυτό το βήμα.
- WHERE : μετά τα άγκιστρα, με τα περιεχόμενα όλων των συνδέσεων των μεταβλητών που ορίζονται για τα δεδομένα των συνολικών δεδομένων που μελετά από το χρήστη, εφαρμόζεται η εντολή αυτή. Στην βασική δομή έχουμε την σύνδεση των δύο μεταβλητών με μια ιδιότητα της οντολογίας, όσον αφορά τα πιο πολύπλοκα ερωτήματα χρησιμοποιούνται περισσότερες ιδιότητες. Στην περίπτωση που είναι χρήσιμα για τον χρήστη να πάρει πιο περιορισμένα αποτελέσματα τότε η εντολή FILTER είναι αυτή που θα χρησιμοποιήσει.
- ORDER BY : είναι μια από τις κυριότερες εντολές που χρησιμοποιούνται για το στάδιο στο οποίο τακτοποιούνται τα αποτελέσματα των διαφόρων εντολών. Η εντολή αυτή συνοδεύεται από μία ή περισσότερες από τις ορισμένες μεταβλητές, από την οποία θα προκύψει η ταξινόμηση τους. Η αύξουσα σειρά θα χρησιμοποιηθεί ταυτόχρονα με την ASC. Η φθίνουσα σειρά θα χρησιμοποιηθεί ταυτόχρονα με την DESC. Μια ακόμα εντολή που είναι αρκετά σημαντική είναι η LIMIT και τη συνοδεύει ένας αριθμός, και αυτό που κάνει είναι να περιορίζει το πλήθος των αποτελεσμάτων που εμφανίζονται στο αριθμό που την συνοδεύει.

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#) | [iSPARQL](#)

Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

select distinct ?Concept where {[] a ?Concept} LIMIT 100

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format: HTML

Execution timeout: 30000 milliseconds (values less than 1000 are ignored)

Options: ☒ Strict checking of void variables ☐ Log debug info at the end of output (has no effect on some queries and output formats)

(The result can only be sent back to browser, not saved on the server, see [details](#))

Run Query Reset

5) DBpedia End-point

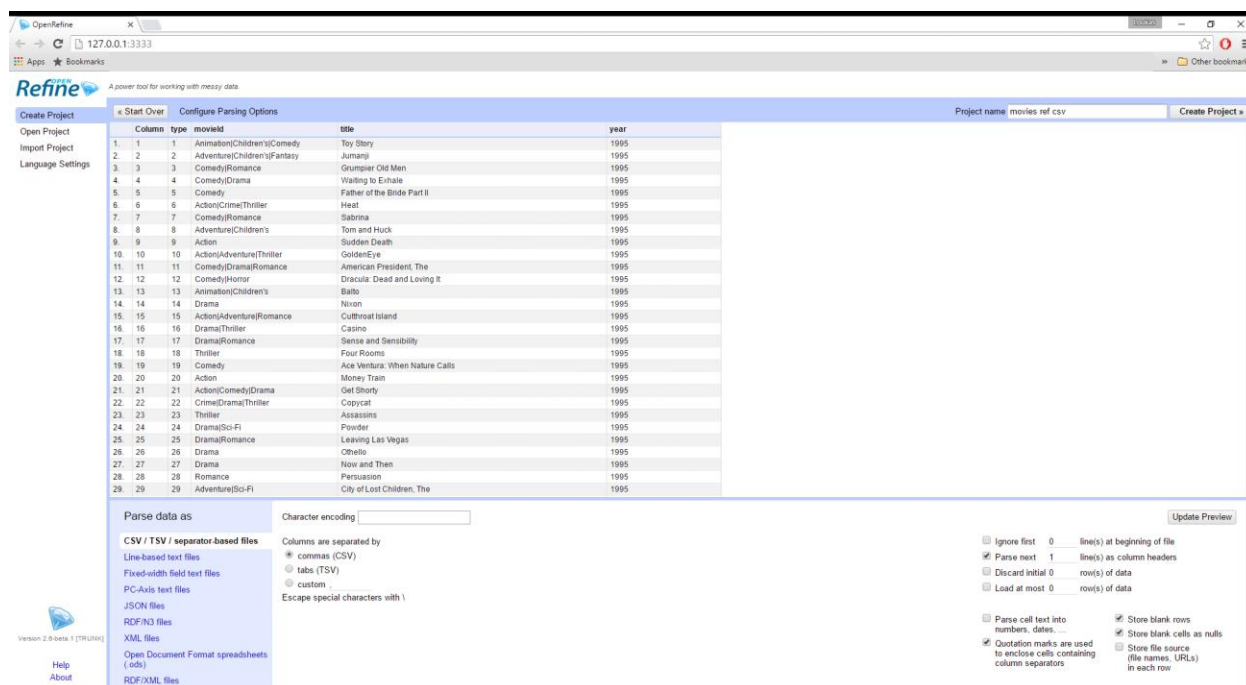
Οι μορφές που θα εμφανίζονται τα αποτελέσματα στον χρήστη θα μπορούν να είναι διαφορετικές, τέτοιες μορφές είναι XML, JSON, CSV/TSV, RDF και HTML. Σε μορφή JSON και XML και σε μορφή πίνακα ανάγνωσης (προεπιλέγοντας Browse) εμφανίζονται τα αποτελέσματα για το endpoint της DBpedia.

3.2 Καθαρισμός

- **Open Refine:** Είναι ένα πολύ χρήσιμο εργαλείο για να διαχειριστεί κανείς την μορφή των δεδομένων, να καθαρίσει ατέλειες, να ομογενοποιήσει διαφορετικά δεδομένα και να κάνει εύκολα μαζικές αλλαγές σε αυτά. Από την πρώτη διαθέσιμη έκδοσή του για γενική χρήση με το όνομα "Google Refine" δημοσιογράφοι δεδομένων, αναλυτές δεδομένων, hactivists και άλλων η χρήση του άρχισε να εξαπλώνεται καθώς η ευκολία

χρήσης του και η δυναμική του ήταν πρωτοφανής και ξεκίνησε η διάδοση του από άτομο σε άτομο σε συνέδρια και ομιλίες. Το πρόγραμμα επίσης διατίθεται δωρεάν ακόμα ένας λόγος λοιπόν για την επιτυχία του. Να σημειωθεί επίσης ότι η ανάπτυξη του εργαλείου έπαψε να υποστηρίζεται από την GOOGLE από τον Οκτώβριο του 2012, και πλέον αναπτύσσεται από εθελοντές. Ωφέλειες που παρέχονται στους χρήστες είναι:

- Η εισαγωγή δεδομένων στην εφαρμογή επιτρέπει δημοφιλή αρχεία όπως CSV, xls, xml αλλά είναι εύκολο να διαχειριστεί και ειδικές μορφές όπως αρχεία με ειδικούς διαχωριστές δίνοντας πάντα στο χρήστη μία διαθέσιμη προεπισκόπηση ώστε να είναι σίγουρος για το αποτέλεσμα.
- Φυσικά κλασικές εντολές όπως διαγραφή στοιχείων, σειρών και στηλών, απλή μεταβολή στοιχείων, αλλά και ταξινόμηση κατά στήλες είναι διαθέσιμες. Εκεί που ξεχωρίζει το Open Refine είναι σε πιο εξελιγμένες επιλογές όπως: 1)ομαδοποίηση παρατηρήσεων με βοήθεια αλγορίθμων που μπορούν να βοηθήσουν να εντοπίσουμε μικρά λάθη σε κελιά χαρακτήρων όπως εισαγωγή περισσότερων κενών. 2) διαχείριση κελιών με πολλαπλές τιμές. Παράδειγμα κάποιος καταχωρεί τηλεφωνικούς αριθμούς στο αρχείο πελατών του αλλά κάποιοι πελάτες παρέχουν πάνω από ένα αριθμούς.
- Πέρα από αυτά όμως κάποιος μπορεί να κάνει αλλαγές στα δεδομένα του χρησιμοποιώντας της γλώσσα General Refine Expression Language (GREL). Για παράδειγμα αν επιθυμούμε να κάνουμε ένα πιο εξεζητημένο φιλτράρισμα στα δεδομένα μας με πολλαπλά φίλτρα ή και εξόδους τιμών μπορούμε να κάνουμε χρήση αυτής της εύκολα κατανοήσιμης γλώσσας.



6) Στην άνω φωτογραφία βλέπουμε το OpenRefine

Πιο αναλυτικά όταν ο χρήστης θα επιλέξει το μενού τότε θα του εμφανιστούν οι περισσότερες επιλογές. Οι ενέργειες τις οποίες μπορεί να κάνει ο χρήστης είναι να ταξινομήσει τα στοιχεία με βάση την στήλη, μπορεί να προσθέτει και να αφαιρεί στήλες, επίσης μπορεί να κάνει αναζήτηση εγγράφων, να κάνει φιλτράρισμα στα αποτελέσματα. Τέλος αυτό που μπορεί να κάνει ο χρήστης είναι να ενοποιήσει τα στοιχεία του με τα ήδη υπάρχοντα στο διαδίκτυο, με τον τρόπο αυτό αυξάνει τα στοιχεία του με νέα στοιχεία.

Η εντολή sort αποτελεί την πιο απλοϊκή εντολή, διότι με αυτήν δίνεται η δυνατότητα στον χρήστη να κάνει μια ταξινόμηση των δεδομένων του είτε σε αύξουσα είτε σε φθίνουσα τόσο αριθμητική όσο και αλφαβητική σειρά. Στην περίπτωση που υπάρχουν στα δεδομένα κάποια κενά κελιά ή κάποια στοιχεία που είναι λανθασμένα τότε αυτά μεταφέρονται απ-ο την θέση τους είτε στην αρχή είτε στο τέλος των δεδομένων και αυτό ανάλογα με την επιλογή του χρήστη.

Η είναι text filter είναι μια ακόμα εντολή που εφαρμόζεται εύκολα. Σε αυτήν την εντολή υπάρχει ένα παράθυρο το οποίο βρίσκεται στο αριστερό παράθυρο της εφαρμογής, και εκεί

είναι που ο χρήστης πληκτρολογεί τους χαρακτήρες που θέλει και παράλληλα στο Open Refine θα εμφανίζονται οι εγγραφές με το περιεχόμενο τους χαρακτήρες που έγραψε ο χρήστης στο αριστερό παράθυρο.

Υπάρχει η κατηγορία view η οποία περιέχει τέσσερις επιλογές:

- 1) Η κατάργηση της στήλης.
- 2) Η κατάργηση όλων των στηλών εκτός από μια.
- 3) Η κατάργηση όλων των στηλών είτε από αριστερά είτε από δεξιά, εκτός από μια στήλη.
- 4) Η επιλογή All με την ταξινόμηση των στηλών συνδυάζει το ολικό καθαρισμό των δεδομένων από πληροφορίες που δεν είναι χρήσιμες για την μελέτη.

Το σημείο στο οποίο διαφέρει το Open Refine από τις εφαρμογές των υπολογιστικών φύλλων είναι η επιλογή της προστιθέμενης αξίας.

Ο χρήστης μπορεί να εφαρμόσει την κατηγορία facet προκειμένου να πάρει όλα τα δεδομένα και να τα κατηγοριοποιήσει ανάλογα με την ποσότητα εμφάνισης τους. Τέτοιες κατηγορίες μπορούν να ενωθούν δημιουργώντας μια συγχωνευμένη κατηγορία, όμως μπορεί να τις αφήσει ως έχουν και να κάνει διάφορες επεξεργασίες σε κάθε κατηγορία ξεχωριστά. Μια πολύ χρησιμοποιημένη επιλογή είναι το text facet, η οποία επιλογή εμφανίζει σε ένα αριστερό παράθυρο στην εφαρμογή τις διαφορετικές εγγραφές που περιέχονται στις στήλες που επιλέχτηκαν να μελετηθούν. Τέτοιες εγγραφές ακολουθούνται από έναν αριθμό που αντιπροσωπεύει την ποσότητα εμφάνισης του στοιχείου αυτού στην στήλη που μελετάτε.

Η χρήση αυτής της εφαρμογής είναι αρκετά σημαντική εφόσον πολλές εγγραφές μπορούν να περιέχουν ίδια στοιχεία, όμως η διατύπωση τους θα διαφέρει και για τον λόγο αυτό ο υπολογιστής να το αντιλαμβάνεται ως κάτι διαφορετικό. Με την χρήση του Open Refine θα μπορεί να γίνει ο εντοπισμός όλων τέτοιων στοιχείων και να κάνει μια ολική επεξεργασία τους προκειμένου να είναι εκχωρημένες με την ίδια μορφή. στην συνέχεια μπορεί να χρησιμοποιηθεί και για φιλτράρισμα, διότι ο χρήστης μπορεί να επιλέξει την κατηγορία που

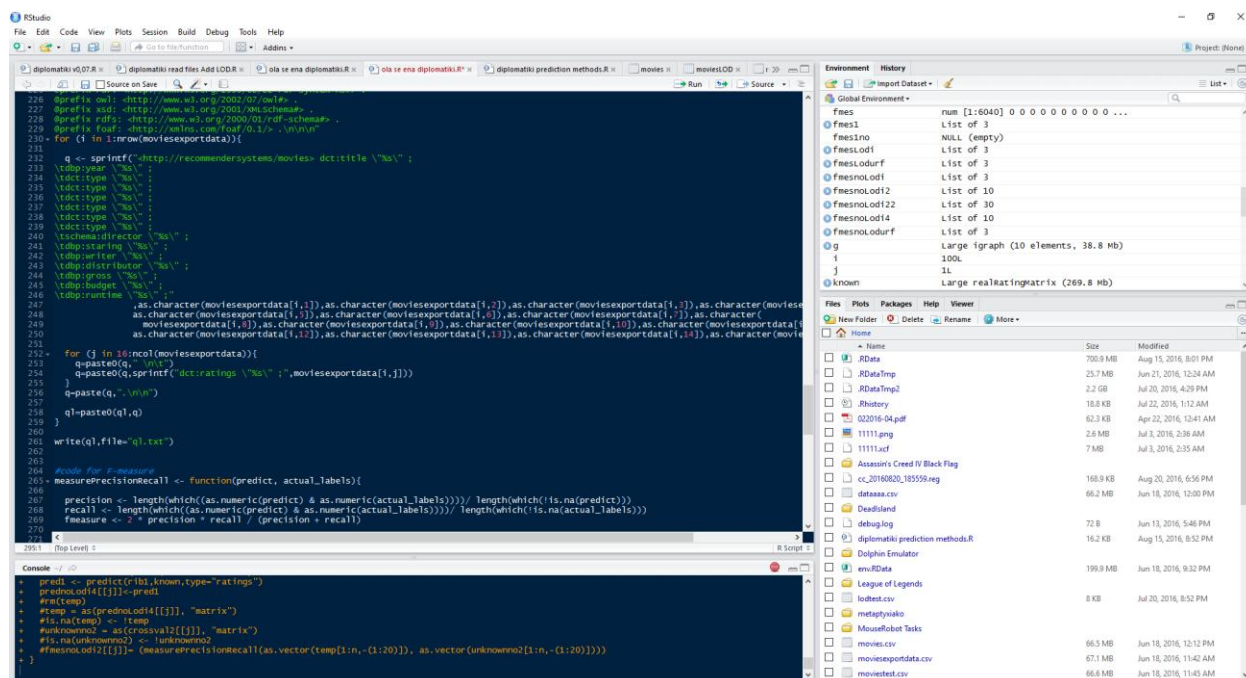
τον ενδιαφέρει και από εκεί να μελετήσει τις σειρές στην οποία υπάρχει η εγγραφή που προεπέλεξε για να την διαχωρίσει από τις άλλες εγγραφές.

Οι περισσότερες ασυνέχειες που μπορούν να υπάρχουν είναι μικρές στο μέγεθος, τέτοιου όπως και το κενό που υπάρχει στο τέλος της εγγραφής ή όπως είναι η διάφορα που έχουν τα κεφάλαια γράμματα από τα πεζά, γενικά τόσο μικρό μέγεθος έχουν. Το Edit cells οδηγεί στο Transform ή Edit cells το οποίο με την σειρά του οδηγεί στο Common transforms όπου ο χρήστης βρίσκεται σε θέση να επεξεργάζεται της ατέλειες μαζικά. Ο χρήστης έχει την δυνατότητα να αλλάξει μια στήλη που περιέχει πεζά γράμματα σε κεφάλαια και το ανάποδο, ακόμα έχει την δυνατότητα να διαγράψει τα κενά που βρίσκονταν στην σειρά ανάμεσα στις λέξεις, επιπλέον και αυτά τα κενά που είναι στην αρχή και στο τέλος των εγγράφων που εισχώρησε, τέλος μπορεί να αλλάξει τα δεδομένα την μια κατηγορία σε μια άλλη.

Ενώ πολλά από αυτά θα μπορούσαν να γίνουν απευθείας από το προγραμματιστικό περιβάλλον της R η φιλικότητα του εργαλείου και άμεση προεπισκόπηση που παρέχει, επιτάχυνε κατά πολύ τις εργασίες που θέλαμε να κάνουμε. Ακόμα και όταν η R φαινόταν καλύτερη επιλογή για κάποια μαζική μεταβολή στοιχείων το Open Refine βοήθησε στην καλύτερη κατανόηση του προβλήματος με την ευκολία των έτοιμων φίλτρων που παρείχε.

- **R studio:** Η εφαρμογή αυτή δεν είναι πάρα ένα νέο UI (Περιβάλλον) για την

αύξηση της παραγωγικότητας. Τα πλεονεκτήματα που προσφέρει είναι πλήρη χρήση της R με δυνατότητα μερικής, δυστυχώς, επισκόπησης των datatable και πινάκων και εμφάνιση των ενεργών μεταβλητών που είναι καταχωρημένες στη προσωρινή μνήμη με πλήρη περιγραφή των ανωτέρω, πράγμα που διευκολύνει την χρήση διαφορετικών πακέτων, καθώς, στα πακέτα που χρησιμοποιούμε λόγο το ότι προορίζονται για επεξεργασία πολύ μεγάλων αρχείων δεδομένων, εκμεταλλεύονται μορφές πινάκων που εξυπηρετούν την καταχώριση αραιών πινάκων ως είναι σύνηθες στα Recommender Systems. Επίσης η εγκατάσταση πακέτων δεν απαιτεί την συγγραφή κώδικα αλλά μπορεί να γίνει από τον αντίστοιχο πίνακα ελέγχου. Ομοίως και η εμφάνιση βοήθειας έχει ενσωματωθεί με παρόμοια λογική.



7) Στην άνω φωτογραφία βλέπουμε το περιβάλλον του Rstudio

R: Η στατιστική προγραμματιστική γλώσσα R είναι ένα πολύ καλό εργαλείο για επεξεργασία και ανάλυση δεδομένων. Είναι μια γλώσσα της οποίας οι χρήστες συνήθως βασίζονται σε βιβλιοθήκες οι οποίες παρέχουν εντολές, πολλές φορές γραμμένες σε άλλες γλώσσες κυρίως c, c++ αλλά και fortran. Η βάση της R έχει πολλές εντολές με τις οποίες κάποιος μπορεί να δημιουργήσει πολλούς αλγόριθμους επεξεργασίας. Το μειονέκτημα της όμως είναι ότι λόγω του τρόπου με τον οποίο εκτελούνται οι επαναληπτικές διαδικασίες στη R οι λούπες μπορεί να επιβαρύνουν τον χρόνο επεξεργασίας δραματικά κάνοντας τη χρήση τους σε μεγάλα datasets χρονοβόρα και ασύμφορη. Εκεί έρχονται να βοηθήσουν οι βιβλιοθήκες της R επιταχύνοντας δραματικά τους χρόνους επεξεργασίας. Ένα ακόμα ζήτημα που έρχονται να λύσουν τα πακέτα είναι η αδυναμία παράλληλης επεξεργασίας καθώς αυτό δεν έχει προβλεφθεί από την βασική έκδοση της R. Το πρόβλημα λύνετε με βιβλιοθήκες που δίνουν στον προγραμματιστή την δυνατότητα να επιμερίσει ημιαυτόματα τις διεργασίες. Εντυπωσιακή είναι επίσης η εισαγωγή βαρέως παραλληλισμένων διαδικασιών που επιταχύνονται με την εκμετάλλευση των πολλών πύρινων των καρτών γραφικών γνωστής εταιρίας επιταχύνοντας έτσι διαδικασίες όπως η εκτέλεση ενός εξαιρετικά περίπλοκου νευρωνικού δικτύου. Για έμπειρους προγραμματιστές δίνεται η δυνατότητα να γράψουν ένα κομμάτι κώδικα σε άλλες γλώσσες και με χρήση

δυναμικά συνδεδεμένων βιβλιοθηκών (dll) να γράψουν ταχύτατες επαναληπτικές διαδικασίες και να της προσαρμόσουν σε ένα κώδικα.

3.3 Βιβλιοθήκες της R studio

- Shiny: Η βιβλιοθήκη Shiny μας επιτρέπει να δημιουργήσουμε εφαρμογές μέσα από το προγραμματιστικό περιβάλλον της R. Μέσω των εντολών που προσφέρει είναι δυνατό να χτίσουμε ένα user Interface ώστε να γίνει μια απεικόνιση δεδομένων, αναζήτηση σε αυτά ή ότι σχετικό μπορεί να απαιτείτε. Για την χρήση της εφαρμογής όμως πρέπει να δημιουργηθεί και ο server τις shiny, από όπου και αντλεί δεδομένα για την απεικόνιση των δεδομένων.
- Recommenderlab: Η βιβλιοθήκη αυτή προσφέρει ένα απλό recommendation algorithm και βοηθά στη δημιουργία νέων αλγόριθμων από τον χρήστη αλλά ακόμα πιο βασικό είναι η παροχή των κατάλληλων στατιστικών για την αξιολόγηση των αλγορίθμων, με τα στατιστικά Mean Average Error (MAE) και Root Mean Square Error (RMSE).
- igraph: Περιέχει εντολές για Γράφους όπως, δημιουργία πίνακα συνδέσεων, μέτρα κεντρικότητας, λίστα ακμών και άλλα. Ιδιαίτερα χρήσιμη για μετατροπή ανάμεσα στο είδος του πίνακα που περιγράφει το δίκτυο.
- Metrics: Όπως καταλαβαίνουμε από το όνομα πρόκειται για ένα πακέτο μετρικών που βοηθά στον υπολογισμό λαθών
- Splitstackshape: Ένα πολύ δυνατό πακέτο για την διαμόρφωση κελιών με διαχωρισμό των μεταβλητών χαρακτήρων με κριτήρια που ορίζει ο χρήστης και γενικά χρήσιμο στην αναδιαμόρφωση των δεδομένων.
- SPARQL: Μας δίνει την δυνατότητα να εκτελούμε sparql ερωτήματα από την κονσόλα της R σε οποιοδήποτε end-point. Πολύ χρήσιμο για επιλεκτικές, επαναληπτικές αναζητήσεις δεδομένων.

4. ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ, ΔΟΜΗΣΗ ΚΑΙ ΚΑΘΑΡΙΣΜΟΣ

4.1 Δεδομένα βαθμολογιών

Τα δεδομένα των βαθμολογιών για ένα recommender system μπορεί να είναι σε διάφορες μορφές: εμπλουτισμένα με χαρακτηριστικά χρηστών, τεμαχίων, σε δυαδική 0-1 μορφή σε βαθμολογίες 0-5. Για την εργασία αυτή επιλέχθηκαν τα ratings πλήθους ενός εκατομμυρίου και 20 εκατομμυρίων εμπλουτισμένα με το είδος ταινιών από το site movielens. Το συγκεκριμένο site αναπτύχθηκε από την ομάδα του groupLens και πρακτικά είναι ένα project για την έρευνα στα recommender systems. Το groupLens είναι το ερευνητικό εργαστήριο του πανεπιστημίου της Μινεσότα το οποίο ασχολείται με recommender systems, online communities, mobile and ubiquitous technologies, digital libraries, and local geographic information systems. Τα δεδομένα που παρέχονται είναι χωρισμένα βάση του πλήθους των βαθμολογιών σε τέσσερις κατηγορίες: 100 χιλιάδων, 1ος, 10, και 20 εκατομμυρίων αντίστοιχα. Πέραν των βαθμολογιών παρέχονται χαρακτηριστικά ταινιών και χρηστών. Τα τελευταία δεν θα χρησιμοποιηθούν. Τα δεδομένα αυτά προέρχονται από την διαδικτυακή πλατφόρμα που διατηρεί και αναπτύσσει η ομάδα αυτή, ένα Recommender system, όπου ο χρήστης μπορεί να επωφεληθεί από την χρήση αυτού συστήματος που του προσφέρει προτάσεις ταινιών σχετικές με τα ενδιαφέροντά του, και παράλληλα οι ερευνητές κερδίζουν μια βάση δεδομένων με την οποία μπορούν να αναπτύξουν τις μεθόδους τους.

4.2 Δόμηση των δεδομένων

Τα δεδομένα που επιλέχθηκαν έρχονται σε μορφή edgelist καθώς είναι η πιο συμφέρουσα μέθοδος για την αποθήκευση τους. Τα δεδομένα των Recommender systems είναι πάρα πολύ αραιά και η αποθήκευση σε μορφή πίνακα βαθμολογιών θα απαιτούσε πολλαπλάσιο χώρο αποθήκευσης. Για να εργαστούμε όμως σε οποιονδήποτε αλγόριθμο, πρέπει να δημιουργηθεί ο πίνακας Βαθμολογιών. Αυτό θα γίνει, με την βοήθεια της R-studio.

Υπάρχουν δύο τρόποι να μορφοποιήσουμε τα δεδομένα. Μπορούμε να γράψουμε τον δικό μας κώδικα με την παρακάτω λογική:

1. Δημιουργία διανύσματος μοναδικών ταινιών και χρηστών
2. Δημιουργία κενού πίνακα με τα άνω διανύσματα
3. Συμπλήρωση του πίνακα με αντιστοίχιση στον αρχικό πίνακα

Ή δεύτερη επιλογή είναι να εκμεταλλευτούμε το πακέτο Igraph. Με το πακέτο αυτό μπορούμε να:

1. δημιουργήσουμε ένα γράφο από τον πίνακα ακμών
2. Από τον γράφο μπορούμε να λάβουμε τον πίνακα συσχετίσεων
3. και να κρατήσουμε τις μοναδικές ταινίες στις γραμμές και τους χρήστες στις στήλες

έτσι καταλήγουμε από την αρχική μορφή του πίνακα ακμών στην τελική μορφή του πίνακα βαθμολογιών:

	userId	movielid	rating
1	1	2	3.5
2	1	29	3.5
3	1	32	3.5
4	1	47	3.5
5	1	50	3.5
6	1	112	3.5
7	1	151	4.0
8	1	223	4.0
9	1	253	4.0
10	1	260	4.0
11	1	293	4.0
12	1	296	4.0
13	1	318	4.0
14	1	337	3.5
15	1	367	3.5
16	1	541	4.0
17	1	589	3.5
18	1	593	3.5

	u1	u2	u3	u4	u5	u6	u7	u8	u9
1	5	NA	NA	NA	NA	4	NA	4	5
2	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA	NA	NA	3	NA
5	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	2	NA	4	NA	NA
7	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	NA	NA	NA	NA	NA	NA	NA	4	NA
15	NA	NA	NA	NA	NA	NA	NA	NA	NA
16	NA	NA	NA	NA	3	NA	NA	4	4
17	NA	NA	NA	NA	NA	4	NA	4	NA
18	NA	NA	NA	NA	NA	NA	NA	NA	NA

8) Πίνακας βαθμολογιών

4.3 Εξόρυξη δεδομένων

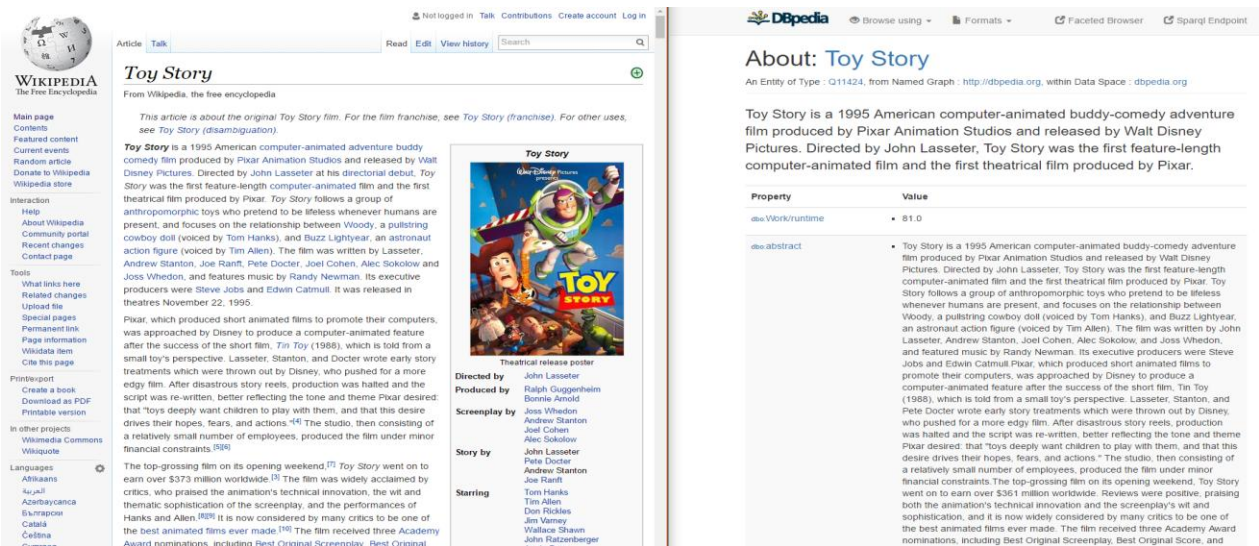
Linked data είναι ο όρος που περιγράφει μια μέθοδο δημοσιοποίησης δομημένων δεδομένων με τρόπο τέτοιο ώστε να είναι μηχαναγνώσιμα και συνδεδεμένα ώστε να αυξάνετε η χρηστική τους αξία. Οι τεχνολογίες που αξιοποιούνται σε αυτή τη μορφή δεδομένων το HTTP (Hypertext Transfer Protocol) και τα URIs (Uniform Resource Identifiers), επεκταμένα ώστε να υποστηρίζουν την μηχαναγνώσιμη φύση τους. Οι βάσεις πάνω στις οποίες στήνονται τα δεδομένα ονομάζονται οντολογίες. Είναι πρακτικά ένα δίκτυο εννοιών που συνδέονται βάση ιδιοτήτων. Επίσης τα δεδομένα που δημοσιεύονται υπό το πρότυπο αυτό, λαμβάνουν τη μέγιστη δυνατή βαθμολογία, στη κλίμακα που έχει προτείνει ο Tim Berners Lee.

Η δομή των διασυνδεδεμένων δεδομένων μοντελοποιείται σε μορφή τριπλετών. Μια λογική και εύκολα κατανοητή μορφή με "Υποκείμενο, Κατηγορούμενο, Αντικείμενο". Για την αναζήτηση και ανάκτηση των δεδομένων έχει δημιουργηθεί η γλώσσα SPARQL. Σύμφωνα με την λογική των RDF δεδομένων έτσι και η SPARQL έχει στηθεί και χρησιμοποιείται με ανάλογο ύφος.

Θα ασχοληθούμε τώρα με την περιγραφή του εμπλουτισμού των δεδομένων ώστε με την βοήθεια παραδειγμάτων να γίνει καλύτερη παρουσίαση της μορφής των δεδομένων.

Από την πασίγνωστη πλέων Wikipedia θα δούμε πως είναι δυνατό να αντλήσουμε δεδομένα για μια συγκεκριμένη ταινία από το SPARQL end-point που είναι διαθέσιμο από το θυγατρικό site της: DBpedia. Η Wikipedia ως η εγκυκλοπαιδεία που βασίζεται στη γνώση των πολλών διαθέτει δεδομένα για πάρα πολλά θέματα, για την εργασία αυτή όμως θα περιοριστούμε στις ταινίες.

Ας δούμε λοιπόν το κομμάτι της σελίδας που θα προσφέρει τις πληροφορίες για την πρώτη ταινία που εμφανίζεται στο dataset των βαθμολογιών.



The image shows two side-by-side web pages. On the left is the Wikipedia article for 'Toy Story', which includes a detailed introduction, a list of characters, and a section on the film's production. On the right is the DBpedia page for 'Toy Story', which provides a structured overview of the film, including its release date, director, and a table of properties and values.

Wikipedia Article: Toy Story

This article is about the original *Toy Story* film. For the film franchise, see *Toy Story (franchise)*. For other uses, see *Toy Story (disambiguation)*.

Toy Story is a 1995 American computer-animated adventure buddy comedy film produced by Pixar Animation Studios and released by Walt Disney Pictures. Directed by John Lasseter at his directorial debut, *Toy Story* was the first feature-length computer-animated film and the first theatrical film produced by Pixar. *Toy Story* follows a group of anthropomorphic toys who pretend to be lifeless whenever humans are present, and focuses on the relationship between Woody, a pullstring cowboy doll (voiced by Tom Hanks), and Buzz Lightyear, an astronaut action figure (voiced by Tim Allen). The film was written by Lasseter, Andrew Stanton, Joe Ranft, Pete Docter, Joel Cohen, Alec Sokolow and Joss Whedon, and features music by Randy Newman. Its executive producers were Steve Jobs and Edwin Catmull. It was released in theatres November 22, 1995.

Pixar, which produced short animated films to promote their computers, was approached by Disney to produce a computer-animated feature after the success of the short film, *Tin Toy* (1988), which is told from a small toy's perspective. Lasseter, Stanton, and Docter wrote early story treatments which were thrown out by Disney, who pushed for a more edgy film. After disastrous story reels, production was halted and the script was re-written, better reflecting the tone and theme Pixar desired: that "toys deeply want children to play with them, and that this desire drives their hopes, fears, and actions."⁴⁶ The studio, then consisting of a relatively small number of employees, produced the film under minor financial constraints.^[98]

The top-grossing film on its opening weekend,^[7] *Toy Story* went on to earn over \$373 million worldwide.^[3] The film was widely acclaimed by critics, who praised the animation's technical innovation, the wit and thematic sophistication of the screenplay, and the performances of Hanks and Allen.^[99] It is now considered by many critics to be one of the best animated films ever made.^[98] The film received three Academy Award nominations, including Best Original Screenplay, Best Original

DBpedia Page: About: Toy Story

An Entity of Type : Q11424, from Named Graph : <http://dbpedia.org>, within Data Space : dbpedia.org

Toy Story is a 1995 American computer-animated buddy-comedy adventure film produced by Pixar Animation Studios and released by Walt Disney Pictures. Directed by John Lasseter, Toy Story was the first feature-length computer-animated film and the first theatrical film produced by Pixar.

Property	Value
dis:Work/runtime	81.0
dis:abstract	Toy Story is a 1995 American computer-animated buddy-comedy adventure film produced by Pixar Animation Studios and released by Walt Disney Pictures. Directed by John Lasseter, Toy Story was the first feature-length computer-animated film and the first theatrical film produced by Pixar. Toy Story follows a group of anthropomorphic toys who pretend to be lifeless whenever humans are present, and focuses on the relationship between Woody, a pullstring cowboy doll (voiced by Tom Hanks), and Buzz Lightyear, an astronaut action figure (voiced by Tim Allen). The film was written by John Lasseter, Andrew Stanton, Joel Cohen, Alec Sokolow, and Joss Whedon, and featured music by Randy Newman. Its executive producers were Steve Jobs and Edwin Catmull. Pixar, which produced short animated films to promote their computers, was approached by Disney to produce a computer-animated feature after the success of the short film, <i>Tin Toy</i> (1988), which is told from a small toy's perspective. Lasseter, Stanton, and Pete Docter wrote early story treatments which were thrown out by Disney, who pushed for a more edgy film. After disastrous story reels, production was halted and the script was re-written, better reflecting the tone and theme Pixar desired: that "toys deeply want children to play with them, and that this desire drives their hopes, fears, and actions." The studio, then consisting of a relatively small number of employees, produced the film under minor financial constraints. The top-grossing film on its opening weekend, <i>Toy Story</i> went on to earn over \$361 million worldwide. Reviews were positive, praising both the animation's technical innovation and the screenplay's wit and sophistication, and it is now widely considered by many critics to be one of the best animated films ever made. The film received three Academy Award nominations, including Best Original Screenplay, Best Original Score, and

9) Βλέπουμε τις δυο αντίστοιχες ιστοσελίδες. Αριστερά η Wikipedia και δεξιά η DBpedia με αναλυτικά στοιχεία για τα διασυνδεδεμένα διαθέσιμα δεδομένα

Τα χαρακτηριστικά που βρίσκονται εντός του infobox όπως: σκηνοθέτης, παραγωγός, σεναριογράφος, πρωταγωνιστές, εταιρίες παραγωγής και διανομής, ημερομηνία κυκλοφορίας, διάρκεια, γλώσσα, budget, box office και άλλα, είναι γενικά διαθέσιμα σε μορφή RDF από την DBpedia. Θα πλοηγηθούμε λοιπόν στη διεύθυνση dbpedia.org/page/Toy_Story. Απλά προσθέτουμε το τελευταίο κομμάτι του συνδέσμου της wikipedia στο σύνδεσμο: dbpedia.org/page/.

Στην σελίδα αυτή βλέπουμε όλα τα δεδομένα και τις αντίστοιχες ιδιότητες από τις οντολογίες που τα περιγράφουν. Για κάποιον που ψάχνει δεδομένα είναι πολύ βολικό να μπορεί εύκολα να βρει τις ιδιότητες και τις οντολογίες για να τρέξει το ερώτημα του στο endpoint της DBpedia.

Για τον αλγόριθμο μας επιλέξαμε τα εξής χαρακτηριστικά:

- Σκηνοθέτης
- Παραγωγός
- Σεναριογράφος
- πρωταγωνιστής

Καθώς είναι προτιμότερο να επιλεγθούν δεδομένα μόνο για τις ταινίες που περιέχονται στο αρχικό dataset, είναι πιο εύκολο ο αλγόριθμος να στηθεί στην R είναι εύκολο να κάνουμε την λήψη των στοιχείων και αντιστοίχηση μέσω της R, 'προγραμματιστικά' χωρίς διάφορα από αν επιλέγαμε να γράψουμε στο end-point της DBpedia.

4.4 Καθαρισμός δεδομένων

Για να γίνει η άντληση των δεδομένων πρέπει πρώτα τα ονόματα των ταινιών να είναι στην σωστή μορφή. Η λίστα των ονομάτων όταν την κατεβάσαμε μαζί με τα δεδομένα των βαθμολογιών, περιείχε στην ίδια στήλη την χρονολογία της ταινίας, ταινίες που ο τίτλος αρχίζει με το άρθρο "The" ή "A" βρισκόταν στο τέλος, καθώς και να ελεγχθούν για πιθανά "λάθη" που μπορεί να προκύψουν κατά την καταγραφή τους. Παραδείγματα:

Glass Shield, The (1994)	The Glass Shield
Black Sunday (La Maschera Del Demonio) (1960)	Black Sunday
League of Their Own, A (1992)	A League of Their Own
"American President, The (1995)"	The American President
"MisΓ©rables, Les (1995)"	Les Miserables

Για προβλήματα που δεν εμφανίζονται τακτικά όπως λάθος χαρακτήρες μέσω του Open Refine είμαστε αναγκασμένοι να κάνουμε την αναζήτηση χειροκίνητα. Αυτό είναι πιο εύκολο να γίνει μέσω το Open Refine καθώς μας δίνει πολύ καλή δυνατότητα εποπτείας στα δεδομένα μας.

Για προβλήματα που παρουσιάζονται συχνά και προβλέψιμα όπως τα άρθρα στο τέλος του τίτλου από φασίσαμε να κάνουμε την αλλαγή μέσω της R. Για τις χρονολογίες έγινε διαχωρισμός σε νέα στήλη. Για τα άρθρα αναζητώντας τις εκφράσεις ", The" και ", A" αντίστοιχα και διορθώνοντας αφαιρώντας τους 5 ή 3 χαρακτήρες από το τέλος και προσθέτοντας τους στην αρχή.

5. ΑΛΓΟΡΙΘΜΟΣ ΣΥΣΤΑΣΕΩΝ ΚΙΝΗΜΑΤΟΓΡΑΦΙΚΩΝ ΤΑΙΝΙΩΝ

Σε αυτή την παράγραφο περιγράφουμε τον αλγόριθμο που χρησιμοποιήσαμε με σκοπό να ελέγξουμε την απόδοση του αλγορίθμου και την επίπτωση των δεδομένων της DBpedia σε αυτόν. Σχολιάζουμε τα αποτελέσματα. Και περιγράφουμε την λογική της εφαρμογής που στήσαμε.

- **Επιλογή αλγορίθμου**
- **Δομή αλγορίθμου και αποτελέσματα**

5.1 Επιλογή αλγορίθμου

Item based collaborative filtering: Επιλέξαμε αυτόν τον αλγόριθμο καθώς μπορούμε εύκολα να αναμίξουμε δεδομένα που αφορούν τις ταινίες, και επίσης όπως φαίνεται και από την βιβλιογραφία είναι από τους πιο κοινόχρηστους αλγορίθμους για την εφαρμογή μας, καθώς είναι σχετικά γρήγορος και τα αποτελέσματα που δίνει είναι ικανοποιητικά. Επίσης παρόμοια προσπάθεια έχει γίνει στο paper: [18] **Feature Weighting in Content Based Recommendation System Using Social Network Analysis**. Η βιβλιοθήκη Recommenderlab προσφέρει την δυνατότητα για Item Based Collaborative filtering, User based collaborative filtering, Popularity, Random, Rerecommend, SVD, SVDF. Επίσης μπορεί να παράγει τα RATINGS ή απευθείας την Top-N λίστα. Με την βοήθεια της βιβλιοθήκης Recommenderlab και δικού μας κώδικα αργότερα, τρέξαμε τον αλγόριθμο για απλό item-based collaborative filtering και με προσθήκη δεδομένων από την Dbpedia αλλά και χωρίς. Αρχικά χρησιμοποιήσαμε αυτή, αλλά ήταν απαραίτητο να γράψουμε τον δικό μας κώδικα ώστε να τον τροποποιούμε κατάλληλα για τις προσθήκες που θέλουμε να κάνουμε, ώστε να χρησιμοποιήσουμε περεταίρω δεδομένα για τις προβλέψεις που θέλουμε να κάνουμε.

5.2 Δομή αλγορίθμου και αποτελέσματα

Πιο αναλυτικά σε ένα αλγόριθμο item-based υπολογίζονται οι αποστάσεις μεταξύ των ταινιών. Η απόσταση επιλέξαμε να υπολογίζεται ως το συνημίτονο μεταξύ των ταινιών, καθώς στην βιβλιογραφία φαίνεται να είναι η πιο κοινή μέθοδος και αποδίδει καλύτερα [16,17] σε πλήθος περιπτώσεων. Κανονικοποιούνται σε μορφή ομοιότητας. Και τέλος υπολογίζονται οι βαθμολογίες ώστε να κάνουμε την πρόβλεψη για τις τιμές που λείπουν από των πίνακα των βαθμολογιών. Ο τρόπος που γίνεται ο έλεγχος είναι με cross-validation με 10 επαναλήψεις αφαιρώντας κάθε φορά το ένα δέκατο των διαθέσιμων βαθμολογιών.

Παράδειγμα Item-Based collaborative :

Movies Users	Toy story	Finding Nemo	Aladin	Shreck
Tom	4	2	NA	5
George	NA	NA	4	3
Martha	NA	5	NA	2
John	3	5	2	NA

$$\text{distance (Toy Story, Finding Nemo)} = 1 - \frac{4 \times 2 + 3 \times 5}{\sqrt{4^2 + 3^2} \times \sqrt{2^2 + 5^2}} = 0.83$$

$$\text{similarity} = \frac{1}{1 + \text{distance}} = 0.5464$$

Ομοίως υπολογίζονται όλες οι ομοιότητες, και από αυτές με πολλαπλασιασμό των ομοιοτήτων με τις στήλες μπορούμε να βρούμε τις προβλέψεις για τις βαθμολογίες όλων των χρηστών για

όλες τις ταινίες. $R_{i,j} = \frac{\sum_y (R_{y,j} \times sim_{i,y})}{\sum_y (sim_{i,y})} \quad y \forall R_{y,j} \neq NA$

Για να συνδυάσουμε τα δεδομένα βαθμολογιών με τα ανοιχτά δεδομένα, πατώντας σε μία λογική παρόμοια με το προαναφερθέν `parser`, ακολουθούμε την εξής λογική:

1. Υπολογίζουμε τους πίνακες ομοιότητας για τα ακόλουθα χαρακτηριστήκα με τις αντίστοιχες μεθόδους:

- Σκηνοθέτης Αν ίδιος =1 αλλιώς =0
- Πρωταγωνιστής Αν ίδιος =1 αλλιώς =0
- Συγγραφέας Αν ίδιος =1 αλλιώς =0
- Διανομέας Αν ίδιος =1 αλλιώς =0
- Γένος Minkowski p=1
- Έτος Minkowski p=1
- Βαθμολογίες Απόσταση συνημίτονου

2. Κάνουμε γραμμική παλινδρόμηση υπολογίζοντας τις βαθμολογίες που έχουμε ανά χρήστη, θέλοντας να μειώσουμε την απόσταση από την πραγματική βαθμολογία. Συνδιάζουμε με βάρη όλες τις ομοιότητες, με την μεταβλητή να είναι το βάρος που

πρέπει να λάβει η κάθε ιδιότητα για τον υπολογισμό της βαθμολογίας ως πρόβλεψη. Υπολογίζονται τα βάρη. Κανωνικοποιούμε ώστε το άθροισμα τους να είναι ίσο με τη μονάδα. Και τέλος βγαίνει ο μέσος όρος τους, απορρίπτουμε τους πίνακες που τους αναλογεί αρνητική τιμή και ξανά, γίνεται κανωνικοποίηση για τα βάρη που έμειναν ώστε το άθροισμα τους να είναι και πάλι ίσο με την μονάδα.

Τα βάρη που προέκυψαν από την διαδικασία αυτή είναι τα ακόλουθα:

- Σκηνοθέτης 0.17913764
- Πρωταγωνιστής -
- Συγγραφέας -
- Διανομέας 0.78463883
- Γένος -
- Βαθμολογίες 0.02497441
- Έτος 0.1224912

Σε αυτό το σημείο πρέπει να επισημάνουμε το γεγονός ότι όσο αφορά τα ανοιχτά δεδομένα για πολλές από τις ταινίες δεν υπήρχαν διαθέσιμα όλα τα δεδομένα που θελήσαμε να κατεβάσουμε. Συγκεκριμένα είχαμε συλλέξει δεδομένα σε ποσοστό για τα ακόλουθα:

- Πρωταγωνιστής 81,35462 %
- Σκηνοθέτης 100 %
- Συγγραφέας 70,10044 %
- Διανομέας 82,79681 %

Είναι λοιπόν πολύ πιθανό τα νούμερα που πήραμε να απέχουν από αυτά που θα έπρεπε να λάβουμε.

3. Πολλαπλασιάζουμε τον κάθε πίνακα με το βάρος που του αναλογεί και τους προσθέτουμε ώστε να χρησιμοποιήσουμε το αποτέλεσμα αυτό ως τον πίνακα που θα κάνουμε τις προβλέψεις μας.

Αφού ολοκληρωθούν οι άνω διαδικασίες χρησιμοποιούμε την ίδια διαδικασία για την πρόβλεψη των βαθμολογιών όπως θα ήταν σε ένα απλό item based collaborative filtering αλγόριθμο αλλά με το νέο πίνακα ομοιότητας που κατασκευάσαμε.

Αποτελέσματα αλγορίθμου:

	f-measure	RMSE	MAE	MSE
Ratings	9,466887	1,011	1,021	0,808
Combined	4,5397350	1,015	1,03	0,811

Βλέπουμε πως όσο αφορά στο σφάλμα ο νέος αλγόριθμος είναι πολύ κοντά με τον παλιό, οριακά χειρότερα όμως. Εκεί που βλέπουμε το πρόβλημα είναι ότι μας μείωσε στο μισό περίπου το f-measure. Πρακτικά τις επιτυχίες στην πρόβλεψη των 10 κορυφαίων ταινιών ανά χρήστη. Αυτό πιθανότατα οφείλεται στο ότι το βάρος έπεσε αποκλειστικά σχεδόν στο ποιος είναι ο σκηνοθέτης της ταινίας.

Στην συνέχεια τρέξαμε την ίδια διαδικασία υπολογισμού των βαρών, αυτή την φορά χωρίς να λαμβάνονται υπόψη τα στοιχεία που απουσιάζουν. Τα βάρη διαμορφώθηκαν ως εξής:

- Σκηνοθέτης 0.379372318
- Πρωταγωνιστής -
- Συγγραφέας -
- Διανομέας 0.618277133
- Γένος 0.002350549
- Βαθμολογίες -

- Έτος -

Και τα αποτελέσματα για όλες τις δοκιμές:

	f-measure	RMSE	MAE	MSE
Ratings	9,466887	1,011	1,021	0,808
Combined	4,5397350	1,015	1,03	0,811
Combined and common	4,2251655	1,022	1,045	0,816

Όμοια με την προηγούμενη δοκιμή το βάρος δεν πέφτει στις βαθμολογίες. Τώρα ο διανομέας λαμβάνει μειωμένο βάρος και αυξάνεται του σκηνοθέτη. Όλα τα αποτελέσματα είναι οριακά χειρότερα.

6. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

Για να παρουσιάσουμε και μία πιο κατανοητή εικόνα για το πώς θα μπορούσε να μοιάζει ένα Recommender system, φτιάξαμε μία απλή εφαρμογή knowledge based, εφήμερου χαρακτήρα. Ο χρήστης μπορεί να αναζητήσει κάποια ταινία που του αρέσει και αντλώντας στοιχεία από έναν πίνακα ομοιότητας, η εφαρμογή, να του παρουσιάσει τις κοντινότερες ταινίες με αυτήν. Ο πίνακας που φτιάξαμε, κατασκευάστηκε από τον πίνακα των βαθμολογιών και έναν πίνακα περιγραφής του είδους των ταινιών. Υπολογίσαμε την απόσταση συνημιτόνου ανάμεσα στις ταινίες. Κανονικοποιήσαμε σε ομοιότητα και βγάλαμε τον μέσο όρο αυτών. Δυστυχώς, το μέγεθος των πινάκων μας ανάγκασε να κρατήσουμε ένα μικρό ποσοστό των διαθέσιμων δεδομένων, όσων αφορά στον πίνακα των βαθμολογιών. Λόγο του μεγέθους των πινάκων που προκύπτουν είχαμε πολλά προβλήματα με την μνήμη Ram η οποία τελείωνε πολύ γρήγορα σε προσπάθειες να χρησιμοποιήσουμε τους πίνακες ως έχουν. Οι μορφές των αραιών πινάκων που προσφέρει η R δοκιμάστηκαν επανειλημμένα σαν λύση αλλά η δυσκολία που παρουσιάζουν στην επεξεργασία τους μας οδήγησε σε αυτή την επιλογή της αποκοπής.

Movie

Recommendations

Select Starting Character: L

Select Movie: Lord of the Rings: The Return of the King, The (2003)

Show 10 entries

Search:

	MovieTitle	Genre1	Genre2	Genre3	Distance
10458	Beowulf & Grendel (2005)	Action	Adventure	Drama	0.508272123084461
12127	Seeker: The Dark Is Rising, The (2007)	Action	Adventure	Drama	0.512890507084935
2345	Mighty Joe Young (1998)	Action	Adventure	Drama	0.514891023517533
12351	D-War (Dragon Wars) (2007)	Action	Adventure	Drama	0.517927081467486
2064	Clan of the Cave Bear, The (1986)	Adventure	Drama	Fantasy	0.519270097674354
15035	Clash of the Titans (2010)	Action	Adventure	Drama	0.520100983820388
4890	Sheena (1984)	Action	Adventure	Fantasy	0.520330803964742
86	White Squall (1996)	Action	Adventure	Drama	0.521914592487738
9176	Merlin (1998)	Action	Adventure	Drama	0.524358424839264
2808	Plunkett & MacCleane (1999)	Action	Adventure	Drama	0.524388719333165

Showing 1 to 10 of 26,502 entries

Previous 1 2 3 4 5 ... 2651 Next

10) Εφαρμογή knowledge based, εφήμερου χαρακτήρα

ΣΥΜΠΕΡΑΣΜΑΤΑ και ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΕΥΝΕΣ

Τέλος παραθέτουμε τις εντυπώσεις για τον αλγόριθμο που στήθηκε, την χρήση των linked data από DBpedia, και καταγράφουμε ιδέες για μελλοντική δουλειά.

- **Σχόλια σχετικά με τα δεδομένα τα βάρη και τα αποτελέσματα.**
- **Linked Data or Not**
- **Ιδέες για μελλοντική εργασία**

Σχόλια σχετικά με τα δεδομένα, τα βάρη και τα αποτελέσματα: Παρατηρούμε από τα βάρη που προέκυψαν, ότι ιδιαίτερα σημαντικό κριτήριο για την επιλογή μίας ταινίας είναι ο διανομέας του έργου, και σκηνοθέτης. Πρακτικά ήταν τα μοναδικά θα έλεγε κανείς κριτήρια που χρησιμοποιήθηκαν στον αλγόριθμο μας, τα αποτελέσματα του οποίου είναι αρκετά κοντά σε αυτά του αρχικού, δηλαδή χρησιμοποιώντας μόνο τις βαθμολογίες. Αν αναλογιστούμε μάλιστα την έλλειψη αρκετών στοιχείων, φαίνεται αρκετά πιθανό να μπορεί κάποιος να το χρησιμοποιεί γενικότερα ως βασικά κριτήρια στους αλγορίθμους του. Εδώ όμως δοκιμάζοντας να βγάλουμε αποτέλεσμα από τις ταινίες, για τις οποίες έχουμε πλήρη στοιχεία, το βάρος αλλάζει και πέφτει περισσότερο στον σκηνοθέτη. Άρα για να εξετάσουμε σωστά την περίπτωση της κατανομής των βαρών, θα πρέπει να επεκταθεί αυτή η μελέτη και σε άλλα datasets, ώστε υπολογίζοντας τα βάρη από κάθε σύνολο δεδομένων να δημιουργηθεί μία λίστα με την σημαντικότητα των χαρακτηριστικών των ταινιών, όσο αφορά τα Recommender systems, καθώς αυτό μπορεί να διαφέρει όπως φαίνεται στο paper που αναφέραμε. Έχοντας λοιπόν μία πληθώρα από διαφορετικά δεδομένα είναι πιο σίγουρο ότι θα φτάσουμε κοντύτερα στον πραγματικό μέσο όρο των βαρών που πρέπει να θεωρήσει κανείς στο σύστημα του, αφού και τα δεδομένα μπορεί να είναι επηρεασμένα από το εκάστοτε Recommender system που χρησιμοποιείτε από τους χρήστες του.

Linked Data or Not: Για να δούμε, αν, η προσπάθεια αυτή αξίζει, και πραγματικά ενισχύει την ικανότητα ενός Recommender πρέπει να γίνει μια σύγκριση στο ίδιο αλγόριθμο. Για να απαντήσουμε οριστικά στο ερώτημα θα πρέπει να γίνει μια εκτενέστερη μελέτη όπως αναφέραμε και πριν. Αλλά το αποτέλεσμα που λάβαμε μόνο από την χρήση δύο ιδιοτήτων της

ταινίας φαίνεται πως με μεγάλη βεβαιότητα μπορούν να βοηθήσουν. Όπως φαίνεται επίσης για να υπάρξει η πλήρης ποσότητα δεδομένων που μπορεί κανείς εύκολα να αντλήσει από πηγές Linked open data, είναι απαραίτητο να υπάρξει μια πιο ξεκάθαρη και ομοιόμορφη δόμηση των δεδομένων αυτών. Παρατηρήσαμε ότι στην DBpedia ο χαρακτηρισμός των ιδιοτήτων μπορεί να διαφέρει κάποιες φορές, κάνοντας την μαζική άντληση τους δύσκολη. Αυτό είναι βέβαια λογικό καθώς η τεχνολογία αυτή βρίσκεται υπό συνεχή και ταχεία εξέλιξη. Καθώς τα δεδομένα ξανακατέβηκαν από την DBpedia πριν την ολοκλήρωση της εργασίας με λιγότερα κενά. Ίσως, όμως για κάποιους που επιθυμούν να κάνουν πιο βαθειά ανάλυση με περισσότερα στοιχεία, να είναι απαραίτητες και άλλες μέθοδοι όπως η αναζήτηση λέξεων κλειδιών, web scraping όπου και αν αυτό επιτρέπεται ή άλλες μεθόδους εμπλουτισμού. Παρόλα αυτά η ευκολία και οργάνωση των δεδομένων που υπόσχεται αυτή η τεχνολογία δείχνει πως αξίζει ο χρόνος που καταναλώνεται στην εξέλιξη της, καθώς είναι εύκολο το στήσιμο ενός dataset που μπορεί να ενδιαφέρει κάποιον ερευνητή, μόνο και μόνο από διασυνδεδεμένα δεδομένα.

Ιδέες για μελλοντική εργασία: Σε συνέχιση της αναζήτησης καλύτερων αποτελεσμάτων πρέπει να εξερευνηθούν και άλλοι τρόποι συνδυασμού των δεδομένων όπως άλλες λογικές για την διαχείριση των βαρών των πινάκων ομοιότητας. Ακόμα μπορούμε να εξετάσουμε και άλλους αλγόριθμους όπως Random forests και να μελετηθεί η επίδραση των επιπλέον δεδομένων σε αυτούς. Η εργασία αυτή μπορεί επίσης, να αποτελέσει μια καλή βάση ώστε να συνδυαστούν τα δεδομένα από βαθμολογίες ταινιών με δεδομένα που αφορούν τους παραγωγούς των ταινιών. Σε συνδυασμό με την άλλη εργασία πάνω στα ανοιχτά κινηματογραφικά δεδομένα που πραγματοποιήθηκε στα πλαίσια του μεταπτυχιακού[19], θα μπορούσαμε να ερευνήσουμε σχέσεις κέρδους και βαθμολογιών, κέρδους και είδους ταινιών κ.α. Όλα αυτά πιθανό θα μπορούσαν να οπτικοποιηθούν μέσω του πακέτου Shiny.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] S.-H. Cha. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions: International journal of mathematical models and methods in applied sciences, Issue 4, Volume 1, pages 300-307, January 2007,
- [2] S. Debnath, N. Ganguly and P. Mitra. Feature Weighting in Content Based Recommendation System Using Social Network Analysis. Proceedings of the 17th international conference on World Wide Web, pages 1041-1042, April 2008.
- [3] M. Hahsler, recommenderlab: A Framework for Developing and Testing Recommendation Algorithms, February 2015.
- [4] R. Mirizzi, T. Di Noia, A. Ragone, V. Claudio Ostuni and E. Di Sciascio. *Movie recommendation with DBpedia*, January 2012.
- [5] H. Zhang, F. Min and S. Wang. *A random Forest Approach to Model-based Recommendation*. Journal of information & Computational Science, pages 5341-5348, October 2014.
- [6] J. Bobadilla, F. Ortega, A. Hernando and J. Bernal. *A collaborative filtering approach to mitigate the new user cold start problem*. Knowledge-Based Systems, volume 26, pages 225–238, February 2012.
- [7] Y. Koren, R. Bell and C. Volinsky, *Matrix factorization techniques for recommender systems*. Yahoo Research Robert Bell and Chris Volinsky, AT&T Labs—Research, August 2009.
- [8] L. Kidzinski. *Statistical foundations of recommender systems*. University of Warsaw Faculty of Mathematics, Informatics and Mechanics. September 2011.
- [9] X. Amatriain, A. Jaimes, N. Oliver and J. M. Pujol. Data Mining Methods for Recommender Systems, Recommender Systems Handbook, pages 39-71, October 2010.
- [10] P. Symeonidis. *Content-based Dimensionality Reduction for Recommender Systems*. Data Analysis, Machine Learning and Applications, pages 619-626, March 2007.

- [11] T. Di Noia , R. Mirizzi , V. Claudio Ostuni and, D. Romito. *Exploiting the Web of Data in Model-based Recommender Systems*, Proceedings of the sixth ACM conference on Recommender systems, pages 253-256, September 2012.
- [12] T. Di Noia , I. Cantador and V. Claudio Ostuni. *Linked Open Data-enabled Recommender Systems: ESWC 2014 Challenge on Book Recommendation*. Volume 475 of the series Communications in Computer and information Science, pages 129-143, October 2014.
- [13] V. Claudio Ostuni, T. Di Noia and E. Di Sciascio, R. Mirizz. *Top-N Recommendations from Implicit Feedback leveraging Linked Open Data*. Proceedings of the 7th ACM conference on Recommender systems, pages 85-92, October 2013.
- [14] R. Verborgh, M. De Wilde. Using OpenRefine. January 2013.
- [15] G. Linden, B. Smith, and J. York. *Amazon.com Recommendations Item-to-Item Collaborative Filtering*. IEEE internet computing, Volume 7, pages 76-80, January 2003.
- [16] I. Cantador, A. Bellogín , D. Vallet, *Content-based Recommendation in Social Tagging Systems*, Proceedings of the fourth ACM conference on Recommender systems, pages 237-240, September 2010.
- [17] G. Karypis, *Evaluation of Item-Based Top-N Recommendation Algorithms*, ACM Transaction systems, Volume 22 Issue 1, January 2004, pages 143-177, October 2001.
- [18] S. Debnath, N. Ganguly and P. Mitra. *Feature Weighting in Content Based Recommendation System Using Social Network Analysis*, WWW '08 Proceedings of the 17th international conference on World Wide Web, pages 1041-1042, April 2008.
- [19] N. Livanos, C. Bratsas, S. Karampatakis and I. Antoniou. Knowledge Networks and Statistical Analysis of Cinematography Linked Data. *CEUR Workshop Proceedings*, vol 1695, September 2016.

ΧΡΗΣΙΜΟΙ ΣΥΝΔΕΣΜΟΙ

<https://en.wikipedia.org/>

<http://wiki.dbpedia.org/>

<http://www.linkedmdb.org/snorql/>

<https://cran.r-project.org/>

<https://www.rstudio.com/>

<http://openrefine.org/>

<http://www.netflixprize.com/>

<https://movielens.org/>

<http://challenges.2014.eswc-conferences.org/index.php/RecSys>