UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE POLITICHE,
ECONOMICHE E SOCIALI

ALGORITHMS FOR MASSIVE DATA MODULE

MARKET BASKET ANALYSIS ON MOVIES AND ACTORS

ANDREA IERARDI
STUDENT'S ID: 960188

DECEMBER 22, 2021

MASTER IN DATA SCIENCE AND ECONOMICS

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Contents

List of Figures

Abstract

The project aim is the implementation of a system for finding frequent itemsets (aka market-basket analysis) and analyzing the IMDB dataset. In particular, the analysis considers movies as baskets and actors as items and the application of the Frequent Pattern Growth Algorithm for identification frequent item sets and association rules.

1 Introduction - Dataset

1.1 Choosen Dataset

The choosen dataset is the «IMDB» and is published on Kaggle, under IMDb non-commercial licensing. Each dataset is contained in a gzipped, tab-separated-values (TSV) formatted file in the UTF-8 character set. The first line in each file contains headers that describe what is in each column. A '/N' is used to denote that a particular field is missing or null for that title/name. It is divided as :

- *title.akas.tsv.gz* : Contains information for titles for each region.
- *title.basics.tsv.gz*: Contains information for titles.
- *title.principals.tsv.gz*: Contains the principal cast/crew for titles.
- *title.ratings.tsv.gz* – Contains the IMDb rating and votes information for titles
- *name.basics.tsv.gz* – Contains information for names.

1.2 Data Cleaning

Data Cleaning and selection techniques are applied on the dataset. In particular, from the original data has been selected only person with role as actor and actresses and movies with type "movies" since it also contains other category of cinematographic creations. For what concert the data pre-processing, no tecniques have been applied to the initial dataset.

1.3 Data Inspection

An exploration of dataset has been done. In particolar it can be divided in two parts:

- Movies Inspection
- Actors Inspection

1.3.1 Movies Inspection

For what concert the movies, we initially focus on the minimun and maximum average rating. Movies are rating starting from a minimum vote equal to 0 to a maximum average of 10.
We also focus on obtaining the ratings distributions dividing it in different bins. The average rating distribution is concentrate majorly in the rate of 5-9.

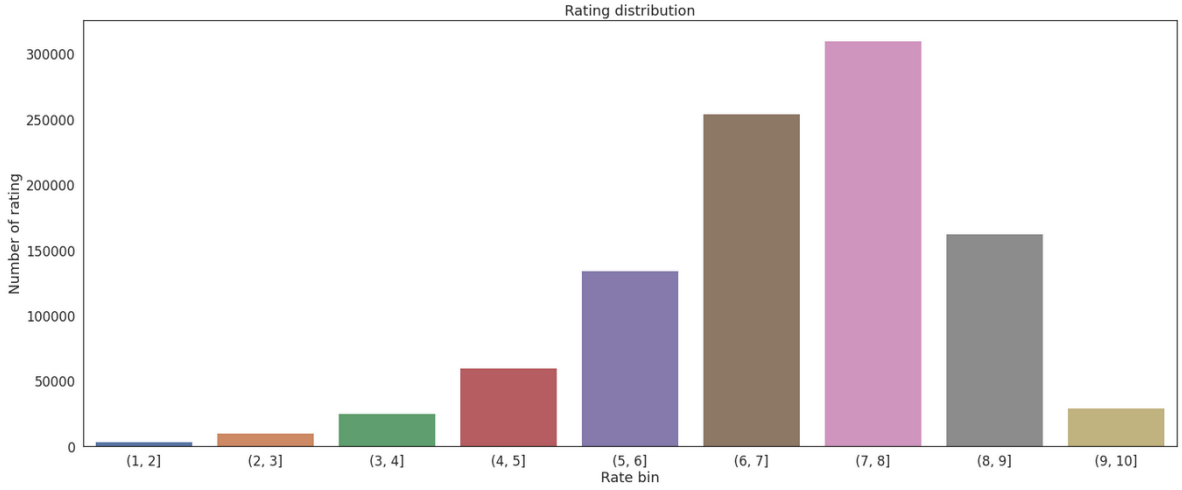

Figure 1.1: Movies bins distribution

Concentrating the ratings in a more general bins [1-4], [4-7], [7-10] we get the an average rating distribution concentrate on the 7-10 bin

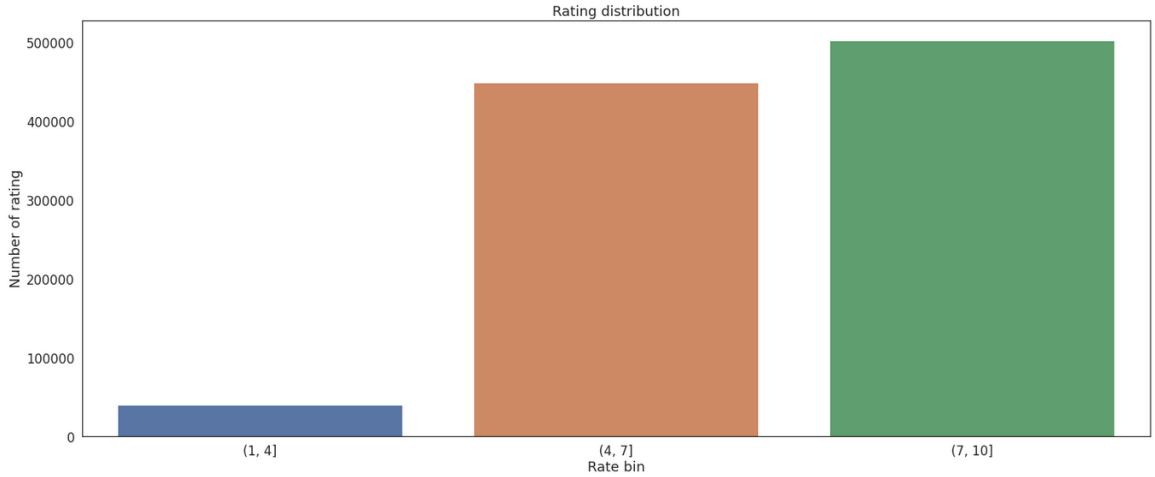

Figure 1.2: Movies bins distribution less bins

Then we focus on the movie genres distributions. From the Figure 1.3 we can see the top ten most common movie genres.

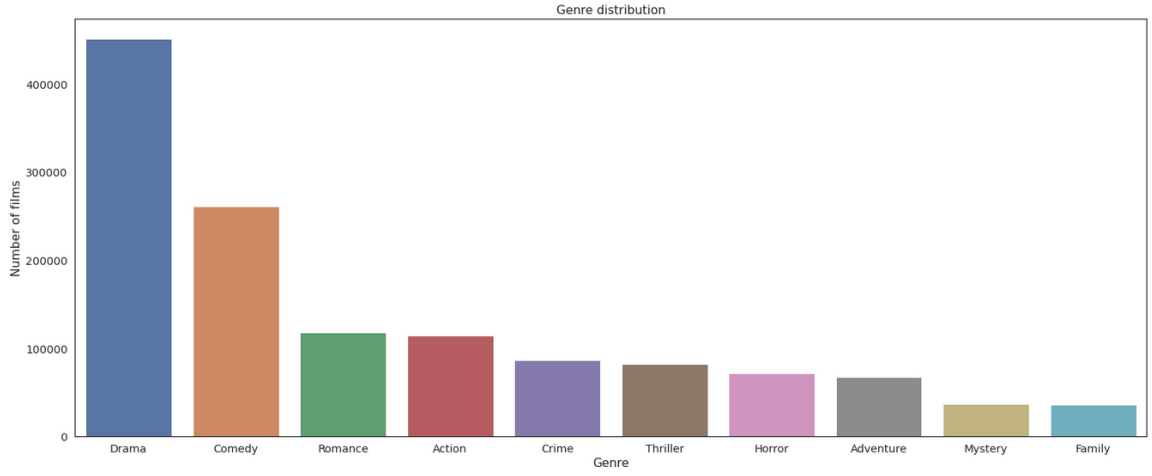

Figure 1.3: Movie genres distribution

From Figure 1.4 we get the pie distribution for all the movie genres. A more refined information than the top-ten one.

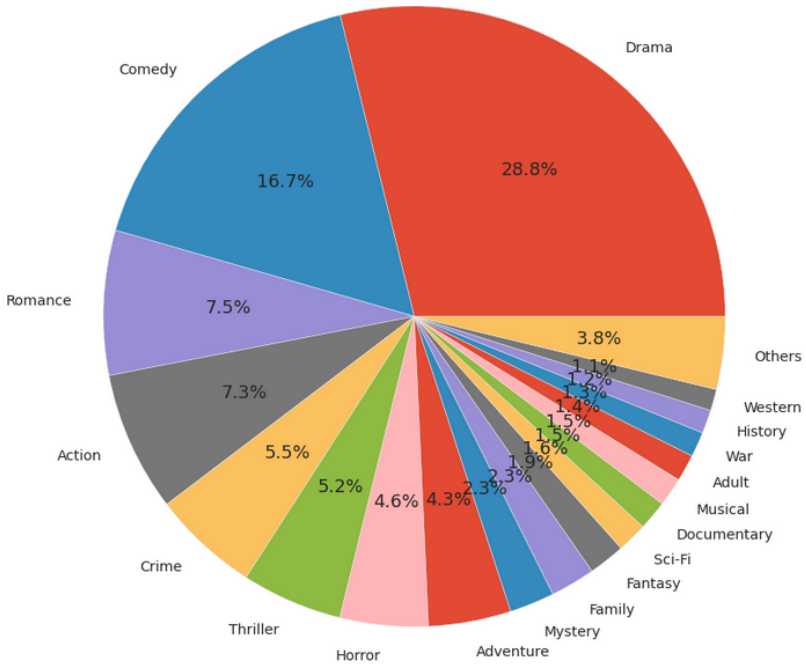

Figure 1.4: Pie plot of movies genre distribution

1.3.2 Actors Inspection

For what concert the actors information we inspect which actor played in more films and the movies with the greater number of actor playing. In Figure 1.5 we can see that Hamlet is the film with the major number of actor played.

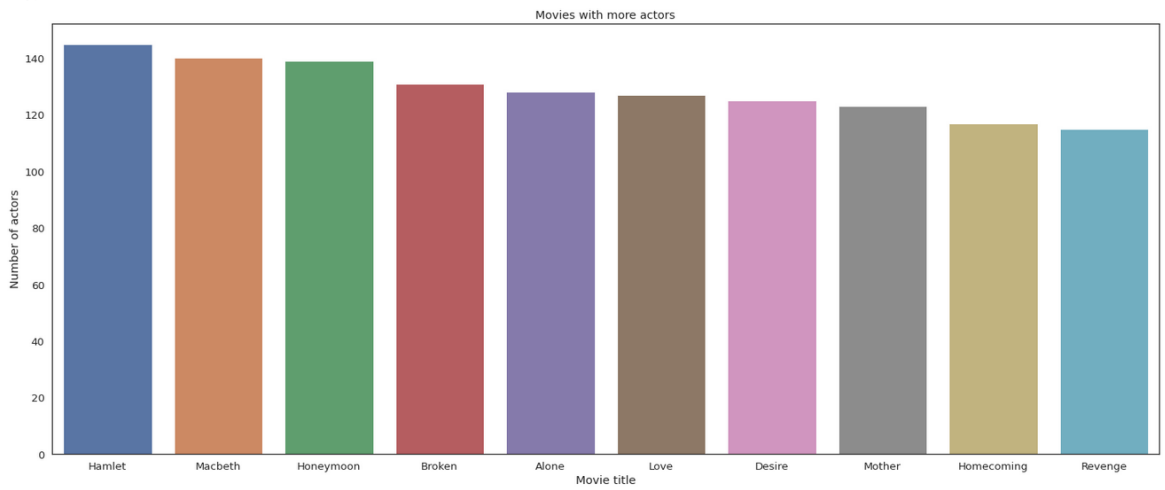

Figure 1.5: Movies with more actors

In Figure 1.6 is possible to see that Brahmanandam is the actor who played in the major number of movies.

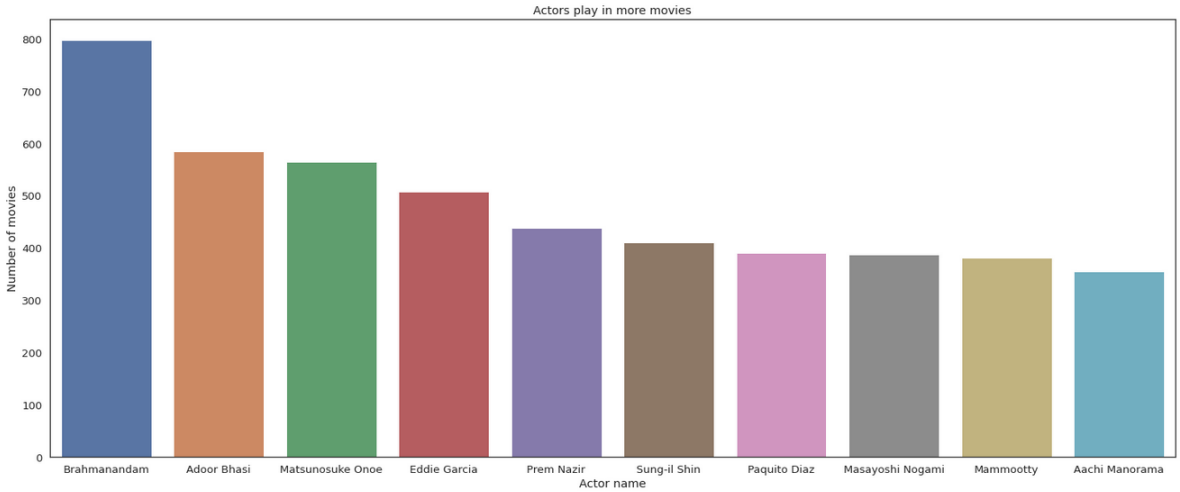

Figure 1.6: Actors who played in the major number of movies

We focus then in the age distribution of the actors. In particular some outliers have been found with a birth Year of 4 which is impossible. So after filtering for a minimum of birth Year greater than 22 and lesser than 2022 we get the actor age distributions in bins. We get a distribution of birth concentrated mostly in the year span of 1950-1975.

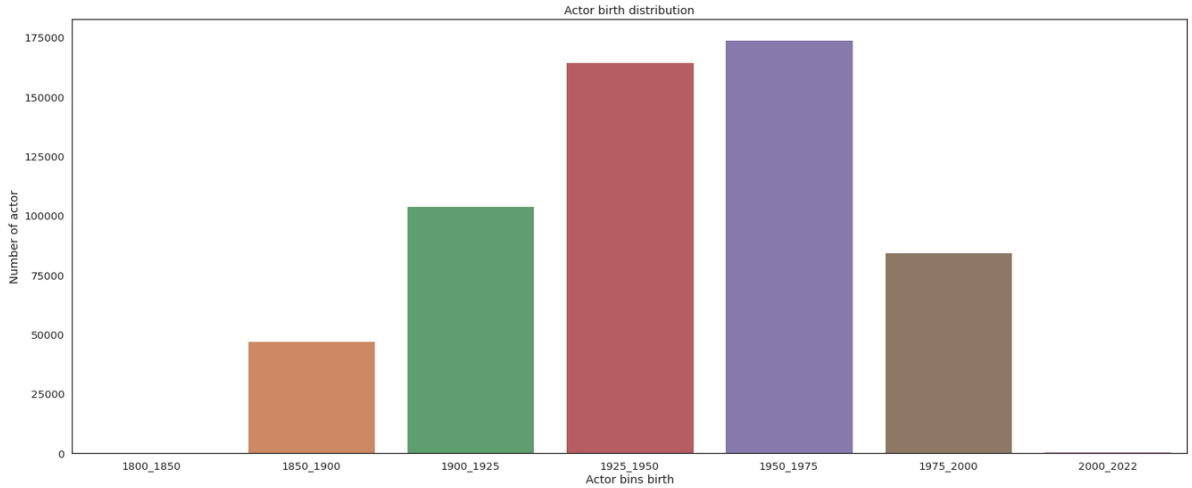

Figure 1.7: Actors birth age distribution in bins

Then we try to inspect the death year distribution.

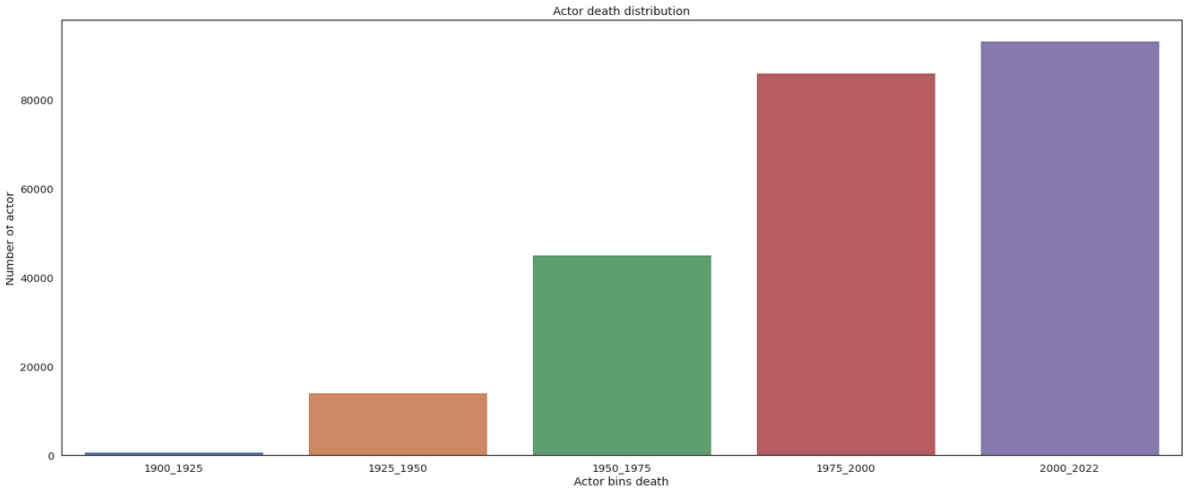

Figure 1.8: Actors death age distribution in bins

We get a distribution of deaths mostly in the 2000-2022 years span.

2 Experimental Methods

2.1 Baskets creation

From the cleaned dataset, baskets of movies containing different actors playing have been created. For this scope, the movies have been grouped by the movies ID to then aggregate each actor playing in that film and finally get the baskets dataset (Figure 2.1)

	tconst	items
0	tt0004272	[Wilbur Higby, Francis Ford, Harry Schumm, Ern...
1	tt0004336	[Marguerite Snow, Frank Farrington, Florence L...
2	tt0005209	[DeWolf Hopper Sr., Julia Faye, Fay Tincher, R...
3	tt0006204	[George Periolat, Richard Bennett, Rhoda Lewis...
4	tt0006489	[Dorothy Dalton, William S. Hart, Enid Markey,...
...	...	...
214171	tt9738388	[Tabla Nani, Chandan Achar, Pranaya Murthy, Su...
214172	tt9750864	[Niousha Alipour, Azadeh Nobahari, Zhila Shahi...
214173	tt9828428	[Kailash, Raashi Khanna, Vinoth Kishan, Aarthi...
214174	tt9848968	[Luci Hare, Michele Hine, Robyn Paterson, Hann...
214175	tt9870726	[Farhad Aeesh, Mahoor Alvand, Behnoosh Tabatab...

214176 rows × 2 columns

Figure 2.1: Basket

2.2 Algorithms

2.2.1 Frequent Patten Growth Algorithm

The Frequent Patten(FP) Growth Algorithm, is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree).
The FP Growth Algorithm has been used to obtain the frequent item sets and the association rule changing the minimum support parameter.

2.2.2 Scaling solution

All the algorithms and solution applied in this project have been implemented using PySpark primitives. In particular, this holds also for the data loading, exploration and the implementation of the algorithms. In this way even changing the dataset size, the resulting operation and inspection can be executed without the consumption of all the system RAM.

3 Results and Discussion

3.1 FP growth Algorithm results

Figure 3.1 shows the most frequent items resulting from FP Growth application with a minimum support set to 0.00005.

items	freq
[Brahmanandam]	412
[Mohanlal]	321
[Mammootty]	312
[Mithun Chakraborty]	298
[Cüneyt Arkin]	281
[Sridevi]	242
[Dharmendra]	237
[Ron Jeremy]	228
[Tom Byron]	227
[Jagathi Sreekumar]	218
[Mohan Babu]	217
[Rajkumar]	212
[Ashok Kumar]	198

Figure 3.1: Frequent Items

Figure 3.2 shows generated association rules generated from the FPGrowth algorithm. Moreover, information about the confidence, lift and the support.

antecedent	consequent	confidence	lift	support
[Jerry Lewis]	[Dean Martin]	0.30434782608695654	1330.2857142857144	6.536680113551472E-5
[Kyle Rea, Robert...	[G. Larry Butler]	1.0	6490.181818181818	5.135962946361871E-5
[Dimitris Papamic...	[Aliki Vougiouklaki]	0.3902439024390244	1899.5654101995565	7.470491558344539E-5
[Sudhir Joshi]	[Laxmikant Berde]	0.5217391304347826	1470.3157894736842	5.602868668758404...
[Tom Tyler]	[Bob Steele]	0.18421052631578946	334.35147190008917	6.536680113551472E-5
[Ric Lutze]	[Rene Bond]	0.4492753623188406	1688.140350877193	1.447407739429254...
[Nandamuri Balakr...	[Vijayshanti]	0.17333333333333334	608.5875409836066	6.069774391154938E-5
[Nandamuri Balakr...	[Brahmanandam]	0.24	124.76271844660194	8.404303003137606E-5
[Vishnuvardhan]	[B.S. Dwarakish]	0.2	1647.5076923076924	5.135962946361871E-5
[Nobuyo Ôyama, Mi...	[Kazuya Tatekabe]	0.7272727272727273	9162.609625668449	7.470491558344539E-5
[Nobuyo Ôyama, Mi...	[Kaneta Kimotsuki]	0.9545454545454546	7301.454545454545	9.805020170327207E-5
[Kamal El-Shinnawi]	[Ismail Yassin]	0.13580246913580246	338.2049956933678	5.135962946361871E-5
[Kamal El-Shinnawi]	[Shadia]	0.1728395061728395	616.9679012345679	6.536680113551472E-5
[Bradford Hill, D...	[Jason Barker]	0.8235294117647058	8819.011764705881	6.536680113551472E-5
[Bradford Hill, D...	[Robert Axelrod]	0.6470588235294118	6599.260504201681	5.135962946361871E-5
[Bradford Hill, D...	[Donald F. Glut]	1.0	9735.272727272726	7.937397280741073E-5
[Bradford Hill, D...	[Marieve Herington]	1.0	10198.857142857143	7.937397280741073E-5
[Aliki Vougiouklaki]	[Dimitris Papamic...	0.3636363636363636	1899.5654101995565	7.470491558344539E-5
[Mona Geijer-Falk...	[John Elfström]	1.0	5491.692307692308	5.135962946361871E-5
[Meena Kumari]	[Ashok Kumar]	0.19047619047619047	206.03751803751803	5.602868668758404...

Figure 3.2: Association Rule

From Figure 3.3 is possible to see how the confidence is distributed amount the association rules. The mean confidence is about 0.42 and ranges from 0.03 to 1.

```
assoc_rules.confidence.describe()
```

```
count    2530.000000
mean        0.418687
std         0.337238
min         0.026699
25%         0.128793
50%         0.281651
75%         0.750000
max         1.000000
```

Figure 3.3: Distribution of Confidence in association rule

From Figure 3.4 is possible to see how the support is distributed amount the association rules. The mean support is about 0.000074 and ranges from 0.000051 to 0.000346.

```
assoc_rules.support.describe()
```

```
count    2530.000000
mean        0.000074
std         0.000030
min         0.000051
25%         0.000056
50%         0.000065
75%         0.000079
max         0.000346
```

Figure 3.4: Distribution of support in association rules

3.2 FP growth Algorithm association rules

Different minimum support values has been used for generating association rules. To better display the result a graph representation has been used. In particular, has been consider each couple antecedent and consequent as a graph edge.
Figure 3.5 show how the association rule appears using a minimum support of 0.0003 as input to the FP Algorithm. There are only 2 couple of node connected.

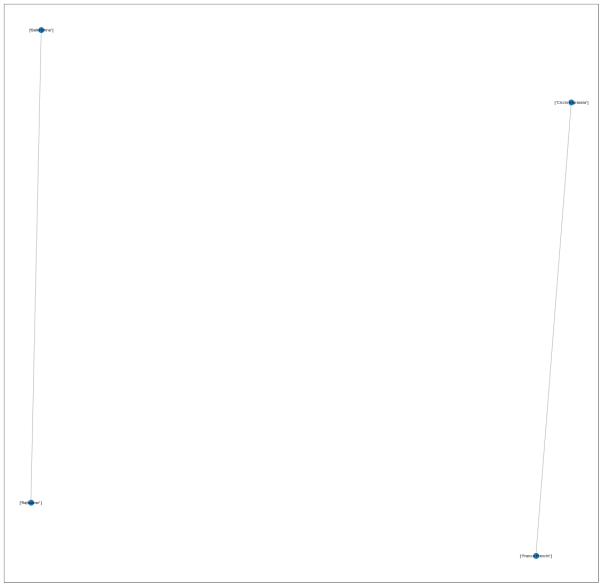

Figure 3.5: Association Rule Graph with minimum support 0.0003

Figure 3.6 shows how are connected the different couples using a minimum support of 0.0002. There are nodes: 22 and links:11

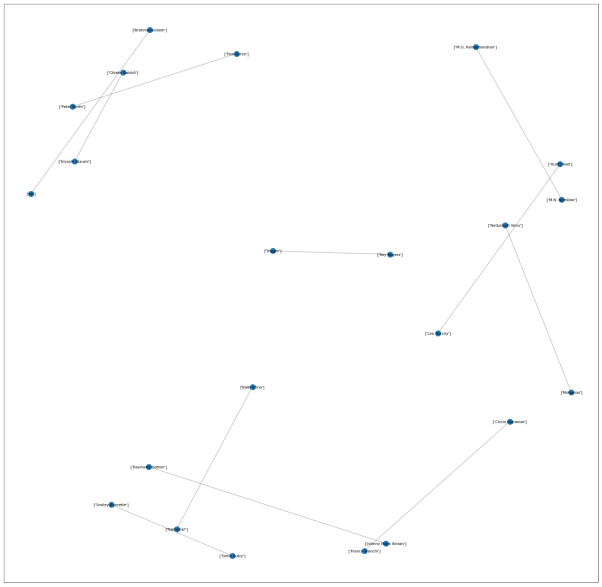

Figure 3.6: Association Rule Graph with minimum support 0.0002

Figure 3.7 shows how are connected the different couples using a minimum support of 0.0001. There are 174 nodes and 156.

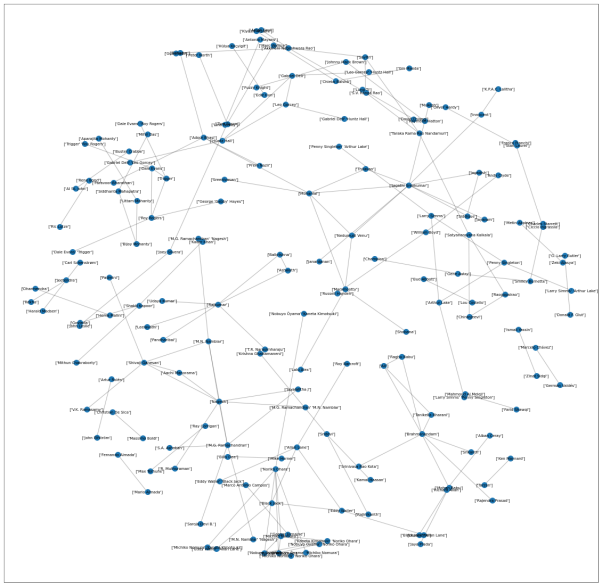

Figure 3.7: Association Rule Graph with minimum support 0.0001

Figure 3.8 shows how are connected the different couples using a minimum support of 0.00005. There are 1061 nodes and 1583 links.

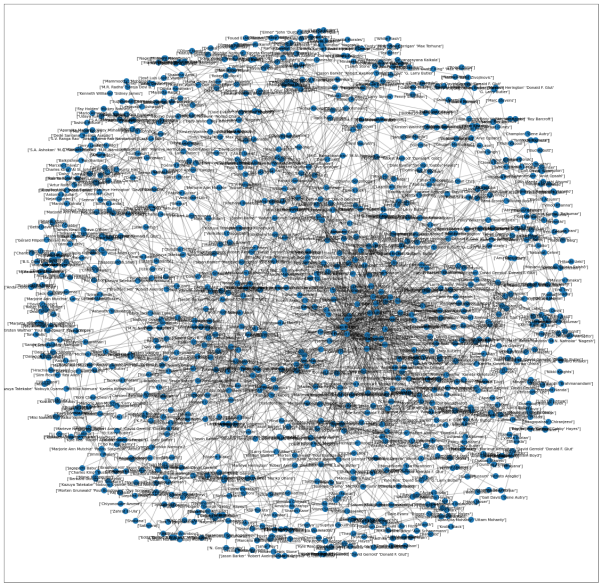

Figure 3.8: Association Rule Graph with minimum support 0.00005

From the previous figure it is possible to know that as the minimum support decrease, the number of nodes will increase. Figure 3.9 displays how the distribution of links increases inversely proportional to the minimum support value. For Support values greater than 0.0003 we obtain empty association rules.

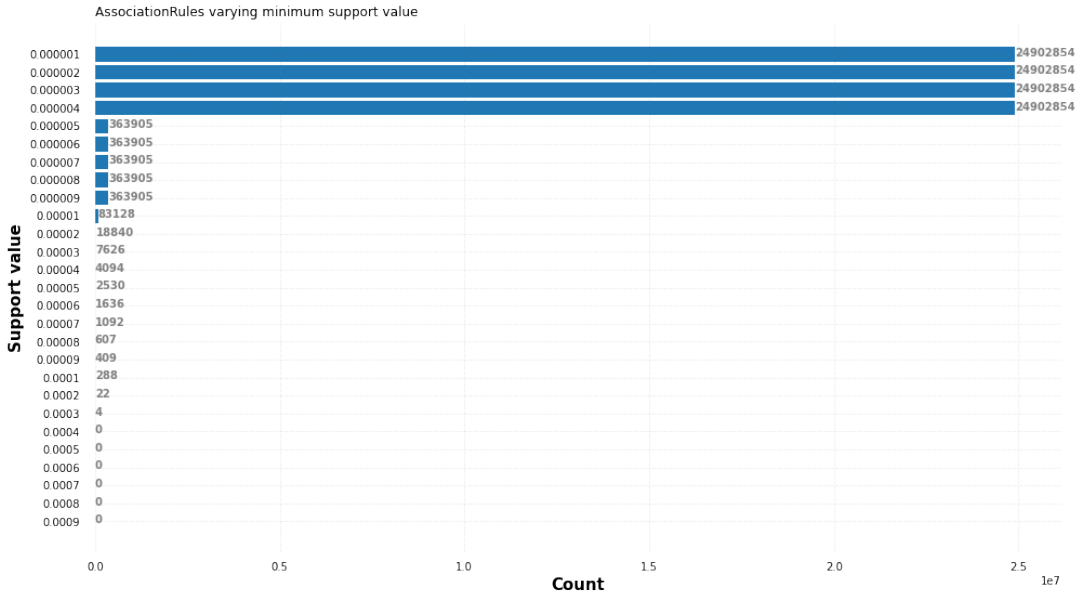

Figure 3.9: Distribution of antecedent and consequent nodes increasing minimum support value

4 | Conclusions

The focus of the study is the implementation of a system for finding frequent itemsets and analyzing the IMDB dataset. The dataset from a IMDB Kaggle competition have been cleaned and inspected resulting in a deep analysis on movies and actors. In particular, movies result having average rating distribution is concentrate majorly in the rate of 7-10 and a major class of movies with Drama and Comedy genres. The movies with most actors playing are Hamlet, Macbeth and Honeymoon, while Brahmanandam results to be the actor who played in the major number of movies. Then, a Basket creation and definition is perfomed for running the Frequent Patten Growth Algorithm. In this way it is possible to define the most frequent items in the baskets and the association rules. In the end a comparison is applied to see how the distribution of antecedent and consequent items changed varying minimum support value. Moreover, a visualization it using Networks Graph is obtained. We demonstrate (Figure 3.9) that the distribution of links increases inversely proportional with respect to the minimum support value.