The Application of Hadoop on Online Streaming Service Provider Industry

TAN CHUN KIT MAN181057

MATTHEW VOON YONG HUI MAN181059

MUHAMMAD ARIEF IRSYADI MAN181065

MUHAMMAD SHAHRUL NADZMI MAN181051

UNIVERSITY TEKNOLOGI MALAYSIA

TABLE OF CONTENTS

Chapter 1 INTRODUCTION 3

1.1 Introduction 3

1.2 Background of Study 3

1.3 Problem Statement 4

1.4 Research Objective 6

Chapter 2 LITERATURE REVIEW 7

2.1 Introduction to Literature Review 7

2.2 Hadoop 7

2.3 HDFS 8

2.4 Spark 8

2.5 Machine Learning and Movie Recommendation 9

Chapter 3 PROJECT IMPLEMENTATION AND RESULT 10

3.0 Summary 10

3.1.0 Dashboard and Descriptive Analytics 10

3.1.1 Data Descriptions 11

3.1.2 Data Preprocessing 12

3.1.3 Data Collection 15

3.1.4 Descriptive analysis and Dashboard 17

3.2.0 Machine Learning and Movie Recommendation 20

3.2.1 Ensuring Data is uploaded into HDFS 21

3.2.2 Coding 21

3.2.3 Pseudo-User 0 and Its Persona 22

3.2.4 Deployment and Result 23

CHAPTER 4 CONCLUSION 26

4.0 Summary 26

4.1 Conclusion 26

REFERENCES 27

APPENDICES 28

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION**

Internet had created a wave of revolutions around the globe and the process of digitization is still continuing to impact various industries. Throughout the year of digitization, the process and product that enabling companies and businesses to ‘stay online’ are becoming more matured and integrated. These products included Social Networks, Semantic Web, embedded systems, Internet of things (IoT), virtualization technologies, and cloud computing (Khayer, Talukder, Bao, & Hossain, 2020). Inevitably, the process of digitization has generated tons of information and data which requires a huge volume of storage, let alone the processing power needed to process and refine these amounts of information into insights.

Many tools have been deployed for the purpose of storing, extracting and processing these huge amounts of information generated, namely the big data. The example of these tools included Google BigQuery, Snowflake, Qubole, Amazon EMR, Hadoop and more (‘g2.com’, 2020). Each of these are taking a similar yet distinctive (in some way) approach in coping with the big data. Among these tools, Hadoop is one of the most widely adopted and well-known platform mainly due to its huge ecosystem of toolkit available and open-source nature. Company such as British Airways, Expedia, IBM and Microsoft adoption of Hadoop further justify its functionality.

This paper explores the Hadoop ecosystem and its usage in big data. An online movie streaming industry (online movie provider) has been selected due to its substantial growth and big data related business problem. Throughout the exploration of the Hadoop ecosystem, solutions were coined and deployed with the hope of solving the related business problem in a practical way.

* 1. **BACKGROUND OF STUDY**

The online movie provider such as Netflix, Amazon Prime Video, Hulu and HBONOW had been growing sophisticatedly recently. This is mainly due to the widespread use of internet and its gradually lower cost of adaptation. Furthermore, almost every online movie provider continuously to market their platform as a low-cost alternative comparing to traditional method such as heading to a cinema or buying a DVD copy of the movie. Given the described phenomenon, it isn’t surprised that these online movie providers have been extremely profitable. Taking Netflix as an example, according to Hoang (2020), Netflix generated nearly 5.77 billion USD in revenue in the quarter 1 of 2020, an equivalent of 27.6% yearly growth. Furthermore, it is expected by 2024, Netflix’s share price could reach 570 USD per share after recently it reached an all-time high at 433.83 USD per share.

Apart from Netflix, which was initiated as an online streaming service, other traditional company such as HBO are now adopting online streaming too. While there are many reasons why video / movie provider or any other marketer are becoming more cling to internet related and enabled services, these reasons can be summarized to 2 point of views as suggested by Khayer, Talukder, Bao, & Hossain (2020):

a. Tremendous market competition

b. Dramatically changing business environment

* 1. **PROBLEM STATEMENT**

Online streaming service provider has its advantages when comparing to traditional movie / video provider:

a. No geographical boundaries

b. Smaller space requirement when it comes to stocking up the media files (only 1 source file for duplication upon purchase)

While having such advantages, many challenges has been faced by the online streaming service provider in advancing and improving their sales and revenue.

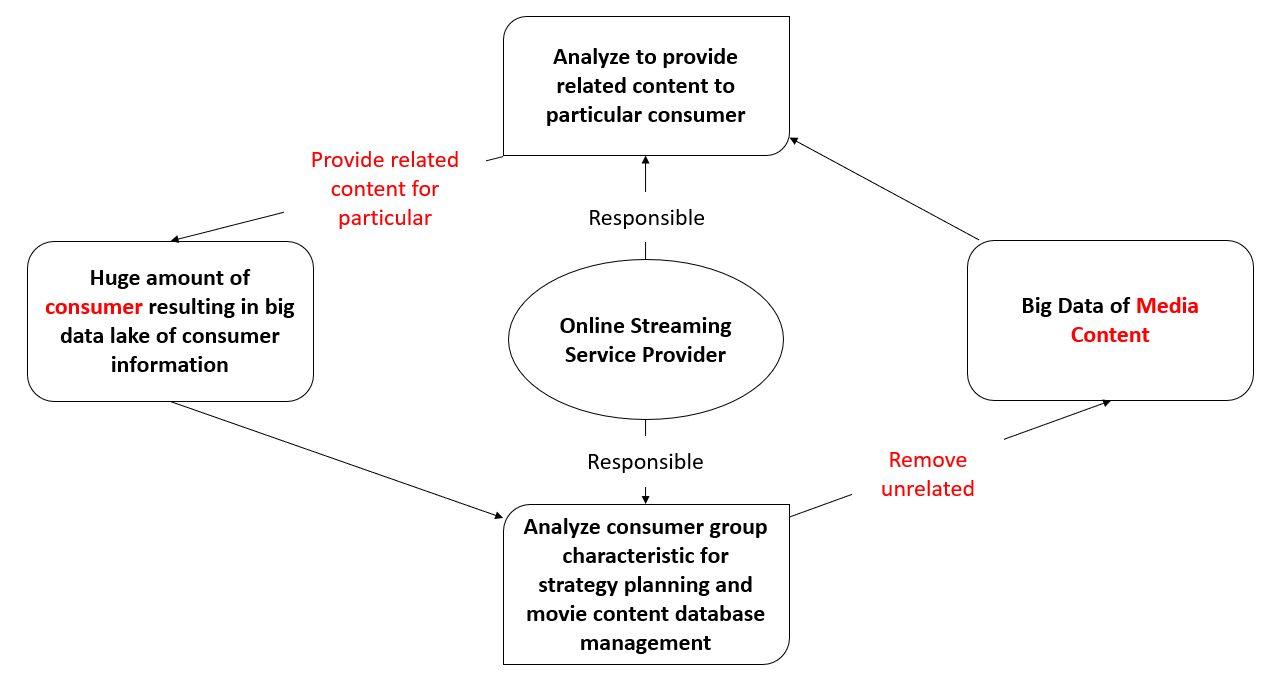


Figure 1.1 Online streaming service provider’s business objectives and challenges

Firstly, the enormous amount of consumer results in service provider’s inability to analyze and understand consumer needs through a traditional analytical manner. The barrier created by internet itself surely has created a thin mist between the service provider and consumer.

Secondly, today media contents are creating at the speed faster than ever. Movies are going on screen almost every single day and the streaming provider couldn’t acquire and sell every copy of it. While on the consumer perspective, users now have to spend plenty of time in searching movies that they are interested. Hence, helping users to find resources that they want rapidly has become an important requirement and most importantly, having customer to look into a long list of uninterested movies is surely a discouragement toward a buying behavior. Figure 1.1 summarizes the challenges being faced by online streaming service providers.

Streaming service provider must resolve these issues in order to further improve their revenue and reduce consumer dissatisfaction. Utilizing Hadoop’s big data capability in storing and processing the data may be helpful in solving the business problem.

* 1. **RESEARCH OBJECTIVES**

In order to cater with the business problem, the following research objectives are defined to serve as a guidance throughout the entire research project.

RO1: To develop a descriptive dashboard that describe and summarize consumer characteristics and movie related information.

RO2: To develop a movie recommendation toolkit based on machine learning approach.

The research question is not explicitly defined given the business problem has been discussed and defined clearly in problem statement, which links directly to the research objective of the project.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION AND PURPOSE OF LITERATURE REVIEW**

Over the years, the importance of managing and making use of Big Data is increasing exponentially. All organizations need to make use of the data available for making decisions, rather than making our own assumptions. Thumb rule may not be relevant enough for the decision making process, since the available data may be producing a better result later on. In this paper, the decision-making process for movie recommendation (Jain, 2016). Which all of the data from every past viewer is taken, and then the analytics result will produce the recommendation for the next movies to be watched. Big Data will just be a waste if it is not being used correctly. In order to make use of 100% of the data’s potential, the implementation of current big data analytical tools and technologies is needed. In order to be able to fully understand and utilize these tools, the following subchapter reviews the related technology in terms of its usage and function.

**2.2 HADOOP**

Apache Hadoop works in an open source platform which is currently popular in the analytics market. The function equipped in Hadoop includes data storage, processing, and analyzing. Since it was released in 2011, the technology has become a great success in the Big Data market, due to its reliability on analyzing Big Data in a short period of time. Although, the speed of processing mostly depends on the computer’s memory (RAM). The Hadoop system is a big step towards helping and makes the analytical process easier to handle. The system is, however, not able to work alone and requires support from other systems for storing, and processing data. Cloud Computing, which was complimenting the big data solutions, had the power to store and provide resources for processing the data. Hence, a new term called Big Data Cloud was introduced (Bhathal, 2018).

**2.3 HDFS**

As stated by IBM, Hadoop Distributed File System (HDFS) is a distributed file system, which has the capabilities to handle a big dataset in a hardware. It is a key part of Hadoop technology, because it does provide reliable data managing functions and can be a great support for a big data analytics application later on (IBM, 2019). In a study done by Yang and Yecies, 2016, they have proven that HDFS is a great system for collecting, distributing, storing and processing data. They prove it as they run the data capturing process for their study for analyzing a set of data related to ‘movie review’. Their process of data crawler module is collecting and distributing data via asynchronous scraping crawler, and then upload and store everything on HDFS. From there, another process is being done easier (Yang & Yecies, 2016). Besides that, the benefit of using HDFS mechanism may increase the profits of the organization involved (Narendra, 2016), (in our case the movie industry) since the data uploading speed using HDFS is quite amazing when it comes to a Big Data type (Rajurkar, 2015). Other than that, HDFS also has the function of providing a mapper that will convert the dataset value pairs, that will go through some process by a reducer, and produce the result we desired (Gupta, 2016). Again in (Gupta, 2016), the proposed work is starting with separating the dataset used for sentiment analysis, which have the input of different movies and reviews to them. They load them into HDFS, which then converts the ratings (string) into the desired key/value, and lastly being put into their algorithm.

**2.4 Spark**

Apache Spark is a system used for handling a big data workload. This includes data distribution and processing in the open source. Spark makes use of the in-memory caching and optimized query performance for fast queries of any size of data. In a simple word, Spark is considered to be a fast and general engine, for big data processing. To produce a much faster result, Sparks make use of the device’s memory (RAM) (Rajurkar, 2015). The component included in Apache Spark Core includes SQL, Streaming, Machine Learning (MLlib), and GraphX (Graph). Furthermore, all of the features that exist within Spark is fast enough to handle the volume, variety, velocity, and veracity of big data. Spark is also said to be better than Map Reduce function, as Spark is rich with SQL series. In conclusion, Spark is one of the most flexible, and fast systems for handling Big Data, with the help of the libraries (TensorFlow, PyTorch, and SciKit-Learn) (Joseph, 2020).

A study done by (Gonzalez, 2019), the researcher did a paper on making a movie recommendation system by using SQL in Spark. Here, Spark is proven to be able to handle more than 30,000 of data (Movie reviews). The hustle is when they need to study on the big data characteristics and to implement the Spark with Scala on the Hadoop ecosystem is always installing and integrating all the tools from these frameworks. The analytics flows done by the researcher are data import, extraction, recommender system (using Machine Learning), and visualization (Python) (Gonzalez, 2019).

**2.5 MACHINE LEARNING AND MOVIE RECOMMENDATION**

Nowadays, the new generation teen is keen on watching movies online, rather than watching it on television. In the process of picking the movie they want to watch, a recommendation segment which will give less hurdle for the watcher to pick their next movie. Recommendation system provides the facility to understand a person's taste and find new, desirable content for them automatically based on the pattern between their likes and rating of different items. Although people's tastes vary, they do follow patterns (Gupta, 2016). People tend to like things that are similar to other things they like as well as other similar behavioural people. Sometimes these types of patterns can be related with the relevancy of items. On the other hand, the system could figure out what items are similar to what they already liked, again by looking at others apparent preferences. The movie recommendation system is proposed for large amounts of data available on the web in the form of ratings, reviews, opinions, complaints, remarks, feedback, and comments about any item (product, event, individual and services) using Hadoop Framework. Depending on the taste of the person a list of movies would be recommended to him (Kadam, 2016).

**CHAPTER 3**

**PROJECT IMPLEMENTATION AND RESULT**

**3.0 SUMMARY**

This chapter described the process of implementation as well as the gathered results divided into 2 section:

3.1 Dashboard and Descriptive Analytics

3.2 Machine learning and Movie Recommendation

The detail process is recorded from data pre-processing to result and findings. Furthermore, toolkit and applications that are employed are discussed too.

**3.1.0 DASHBOARD AND DESCRIPTIVE ANALYTICS**

This part of the project is implemented according to the flow as illustrated as in figure 3.1. Figure 3.1 provided the data flowchart across the employed applications and tools.

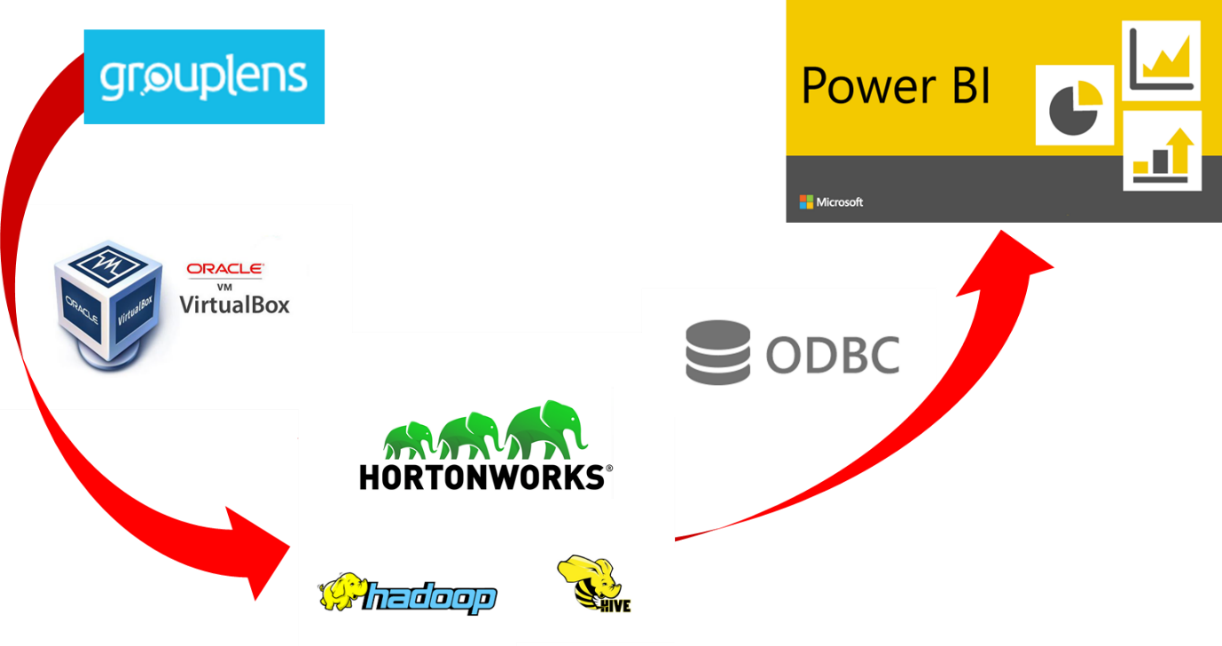
****

Figure 3.1 Data flowchart across the applications

**3.1.1 DATA DESCRIPTION**

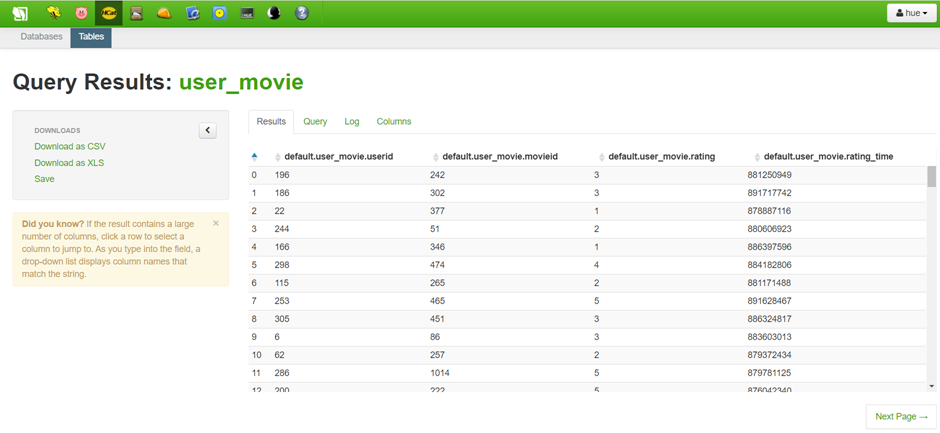
The MovieLens dataset that used for descriptive analysis is extracted from the website hosted by GroupLens. There are a number of datasets that are available for recommendation research. Amongst them, the MovieLens dataset is probably the one of the more popular ones. MovieLens is a non-commercial web-based movie recommender system. It is created in 1997 and run by GroupLens, a research lab at the University of Minnesota, in order to gather movie rating data for research purposes. MovieLens data has been critical for several research studies including personalized recommendation and social psychology. Several versions are available in GroupLens website. We will use the MovieLens 100K dataset [Herlocker et al., 1999]. This dataset is comprised of 100,000 ratings, ranging from 1 to 5 stars, from 943 users on 1682 movies. It has been cleaned up so that each user has rated at least 20 movies. Some simple demographic information such as age, gender, genres for the users and items are also available. There are many files in the folder, a detailed description for each file as shown in the table 3.1.

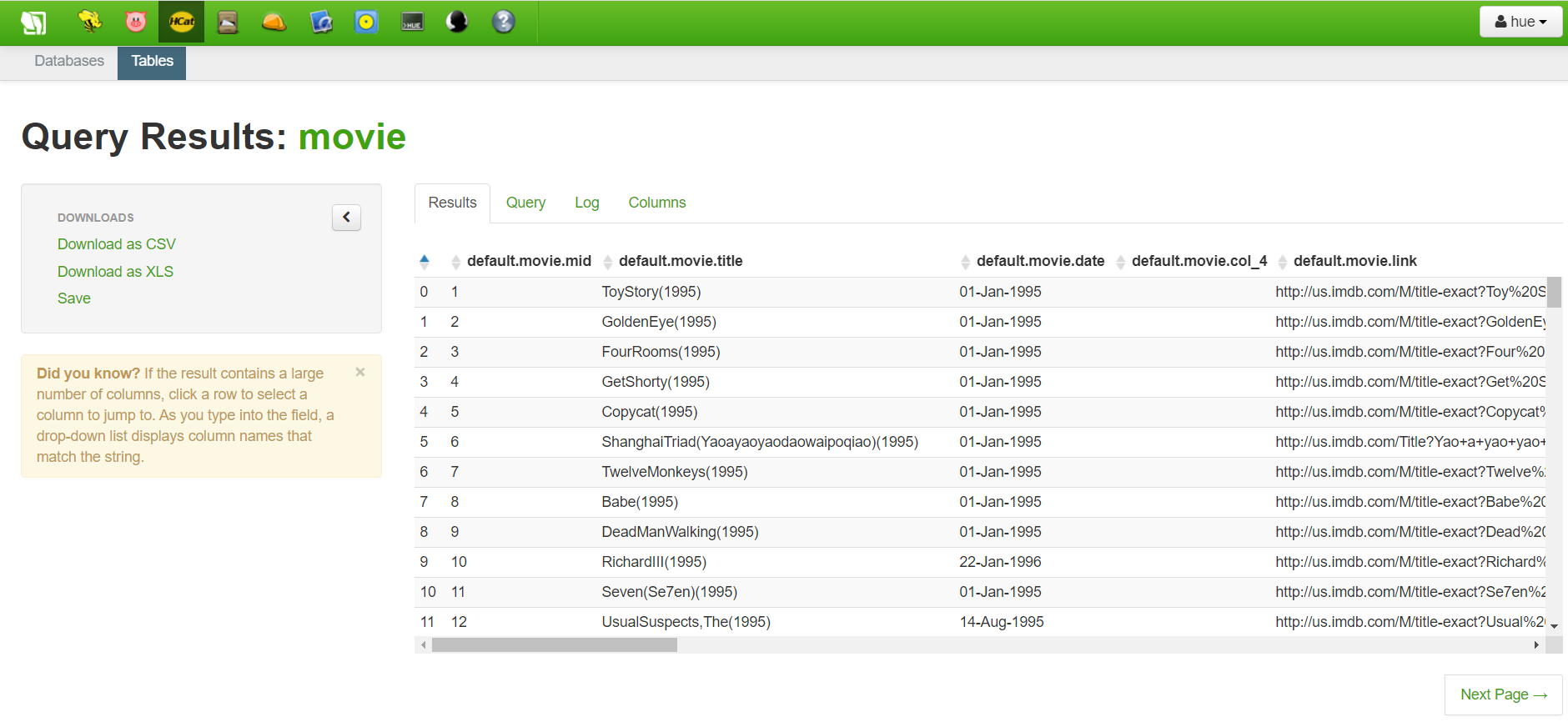
Table 3.1 Files, data attributes and data types

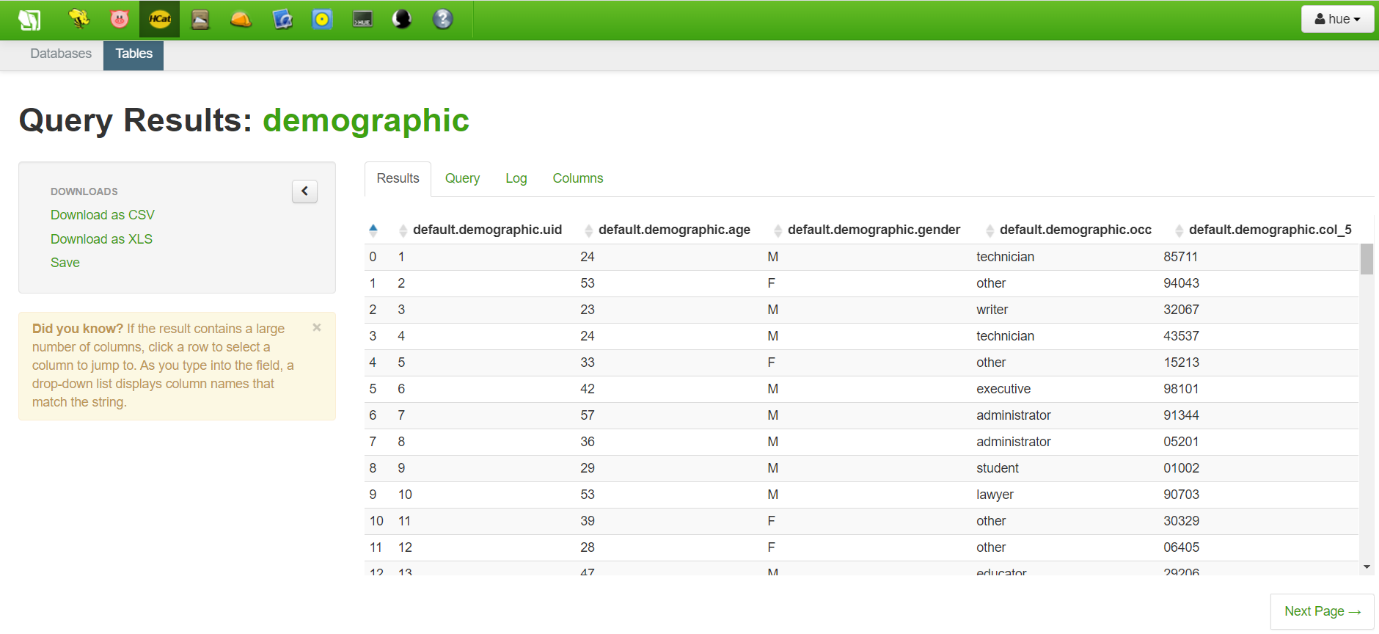
|  |  |  |
| --- | --- | --- |
| **Files** | **Attributes** | **Data Types** |
| u.data | user id | Integer |
| item id | Integer |
| rating | Integer |
| timestamp | Timestamp |
| u.item | movie id | Integer |
| movie title | String |
| release date | Date |
| video release date | Date |
| IMDB url | String |
| unknown | Integer |
| Action | Integer |
| Adventure | Integer |
| Animation | Integer |
| Children’s | Integer |
| Comedy | Integer |
| Crime | Integer |
| Documentary | Integer |
| Drama | Integer |
| Fantasy | Integer |
| Film-Noir | Integer |
| Horror | Integer |
| Musical | Integer |
| Mystery | Integer |
| Romance | Integer |
| Sci-Fi | Integer |
| Thriller | Integer |
| War | Integer |
| Western | Integer |
| u.user | user id | Integer |
| age | Integer |
| gender | String |
| occupation | String |
| zip code | Integer |

**3.1.2 DATA PRE-PROCESSING**

The dataset is pre-process using Apache Hadoop components known as Apache HCatalog and Apache Pig. First of all, the u.data, u.item and u.user which extracted from the ml-100k folder is uploaded to the HCatalog portal and assign as new tables with new attribute name and data type as shown in Figure 3.2, 3.3 and 3.4.

Figure 3.2 New Table of u.data

Figure 3.3 New table of u.time

Figure 3.4 New table of u.user

Next, the combination of three new tables as one is executed using Query Editor in Apache Hive with the command as illustrated in Figure 3.5 and 3.6. Figure 3.5 represent the command used to join the new table of **u.data** and **u.time** and at the same time created new table named as ml100k\_1. On the other side, figure 3.6 represent the command used to join the new created table of ml100k\_1 and the new table of **u.user**. At last, the final table that containing all data from the three initial tables is created to proceed with data analysis as shown in figure 3.7.

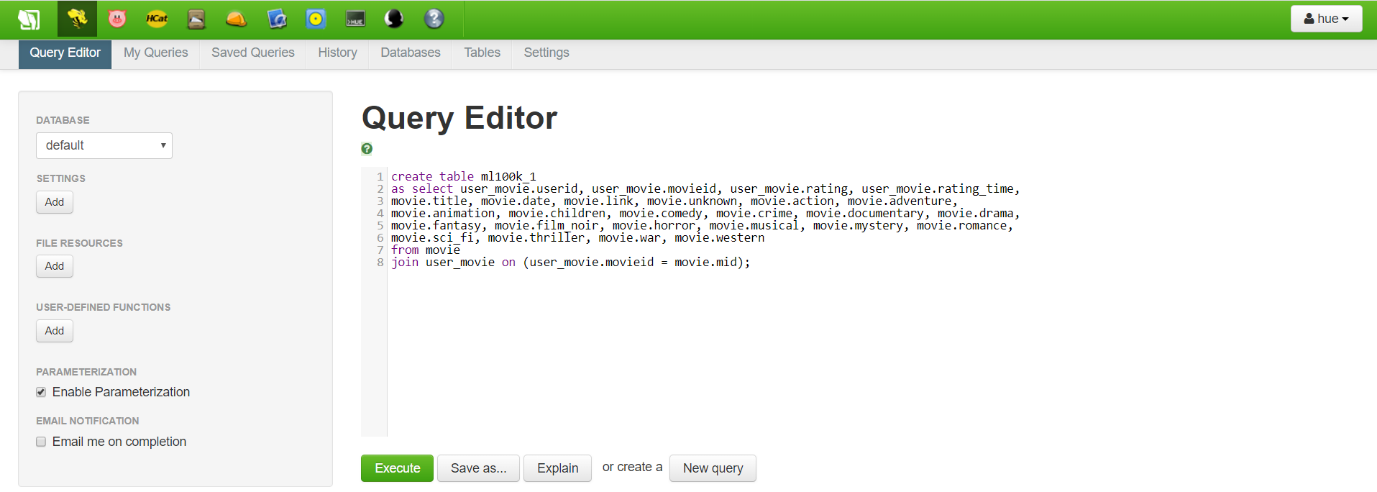


Figure 3.5 Command for joining new table of u.data and u.time

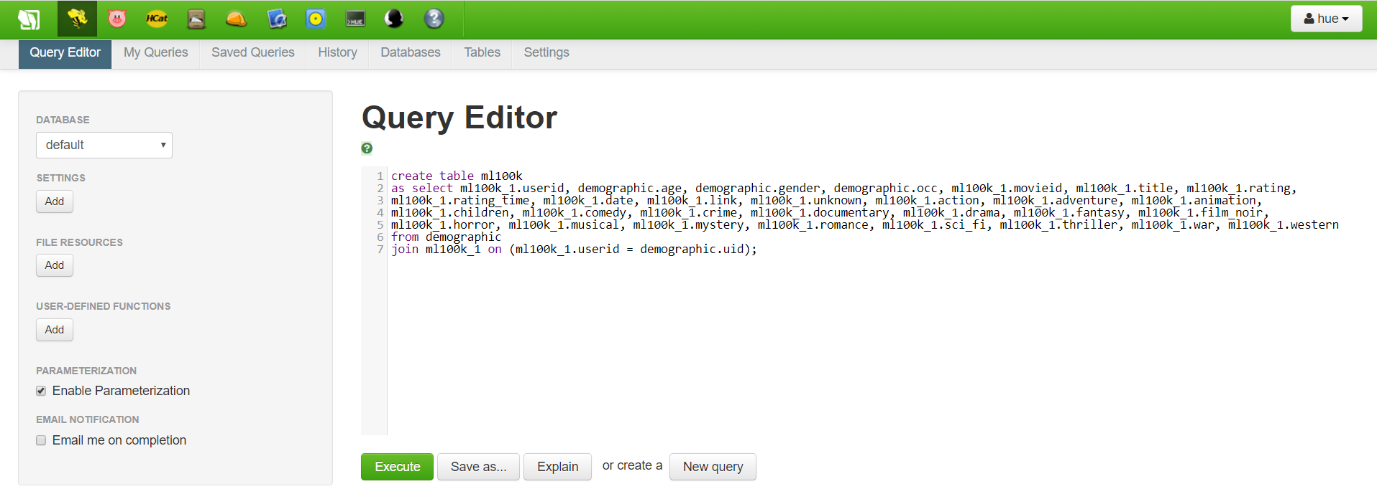


Figure 3.6 Command for joining new created table of ml100k\_1 and new table of u.user

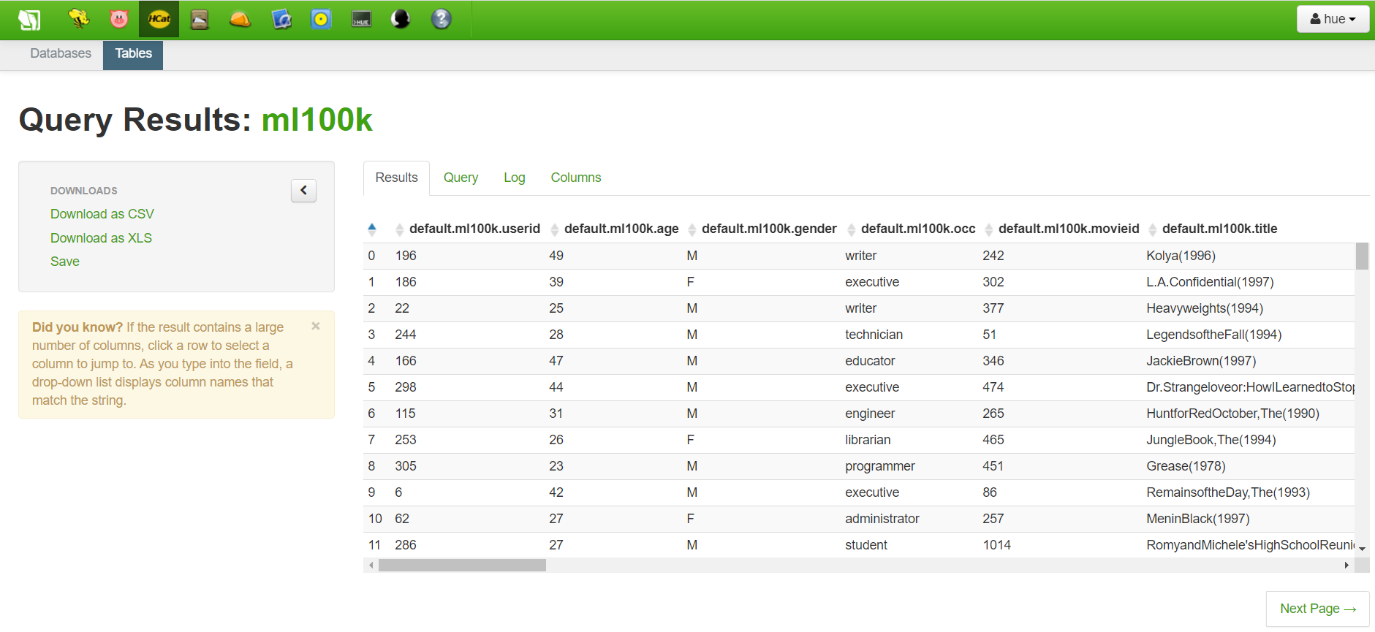


Figure 3.7 Figure 6 Final table of ml100k

**3.1.3 DATA COLLECTION**

After pre-processing the dataset and creating final table of ml100k, the data from the final table is exported to Power BI using ODBC driver which act as a standard application programming interface (API) for accessing database management systems. However, it is required to give permission for allowing access data from Hortonworks as indicated in figure 3.8. Thus, the dataset is able to loaded and used in Power BI as shown in figure 3.9.

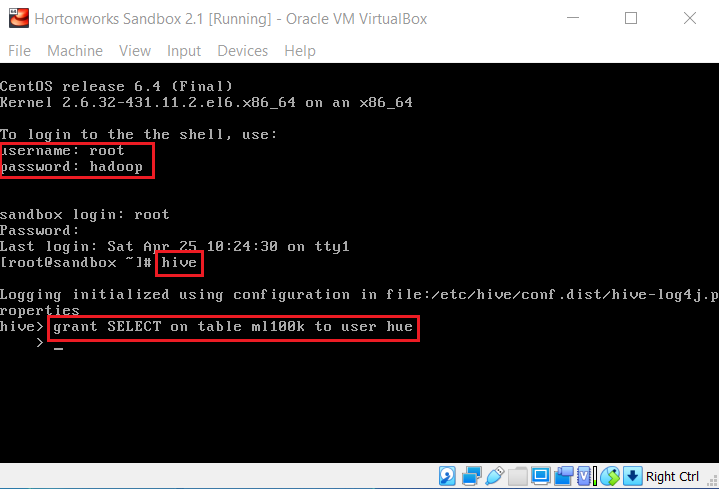


Figure 3.8 Command for access data from Hortonworks

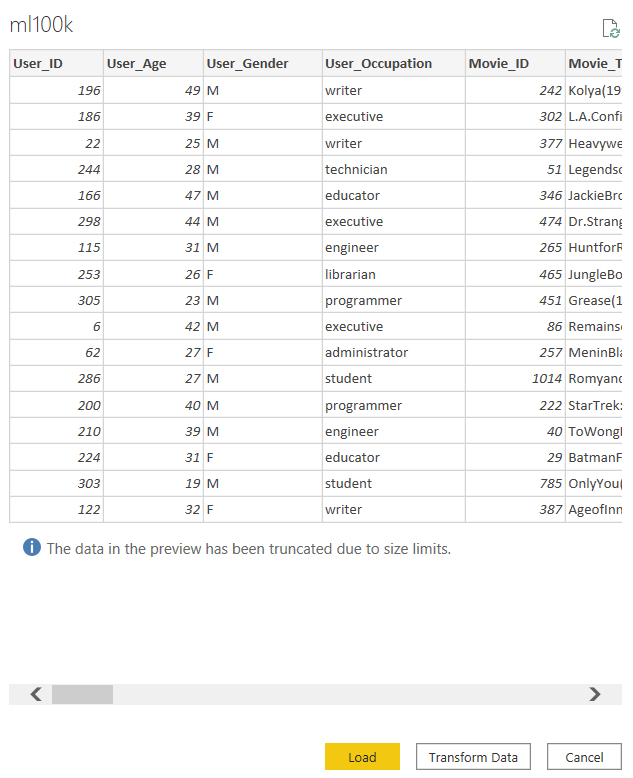


Figure 3.9 Data loading using Power BI

**3.1.4 DESCRIPTIVE ANALYSIS AND DASHBOARD**

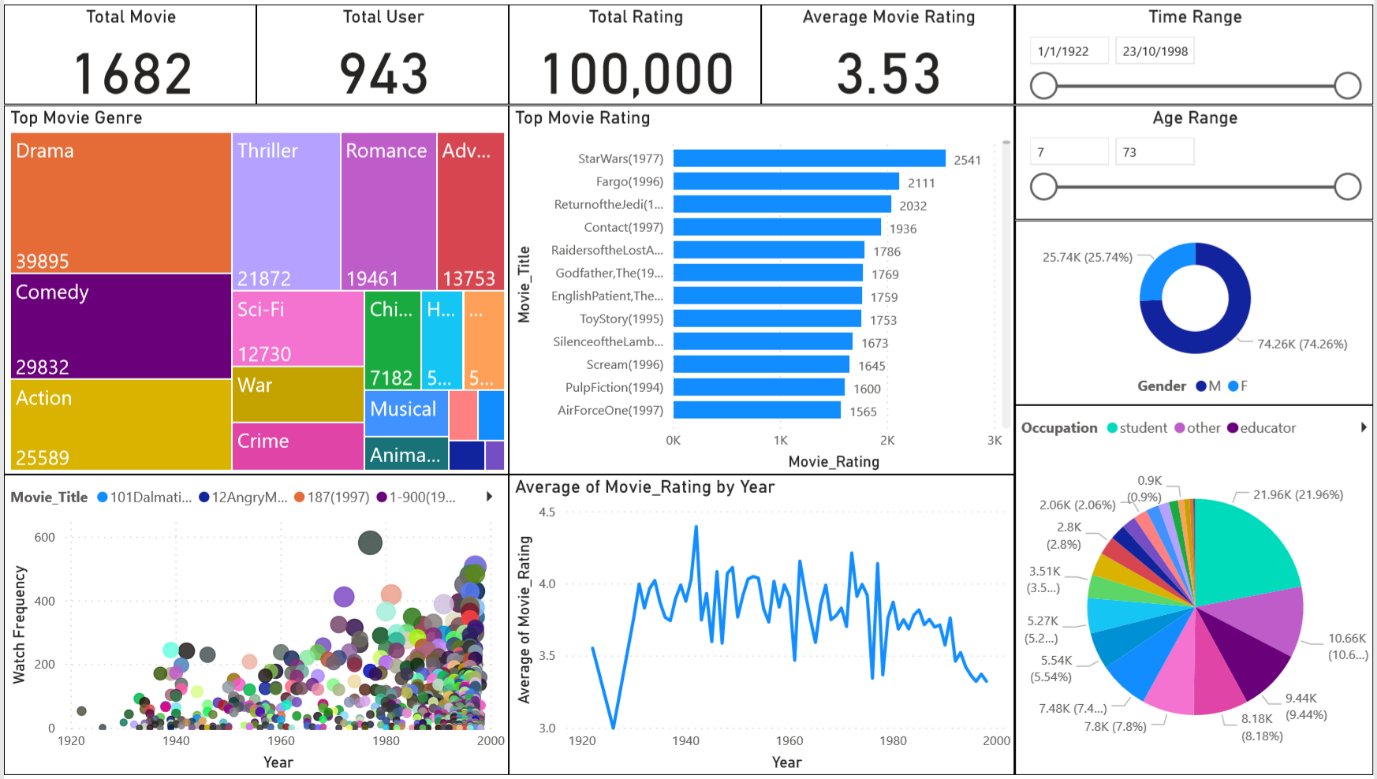


Figure 3.10 Power BI dashboard for MovieLens dataset (default)

From figure 3.10, it shows the total amount of 1682 movies, 943 users, 100,000 user’s rating, average movie rating of 3.53, bar chart of top movie genre, tree map of top movie rating, donut chart of user’s gender and pie chart of user’s occupation for the respective time frame and range of user’s age. Hence, all the values will change according to the time range and age range. This will help the user to visualize the trend of each aspect with different time frame and age range. For example, the user able to visualize the total movie release, total user, total rating, average movie rating, top movie genre, top movie rating, distribution of user’s gender and occupation from year 1993 to 1998 as shown in figure 3.11 and even with specific age range of 63 to 73 based on specific user’s occupation as demonstrated in figure 3.12.

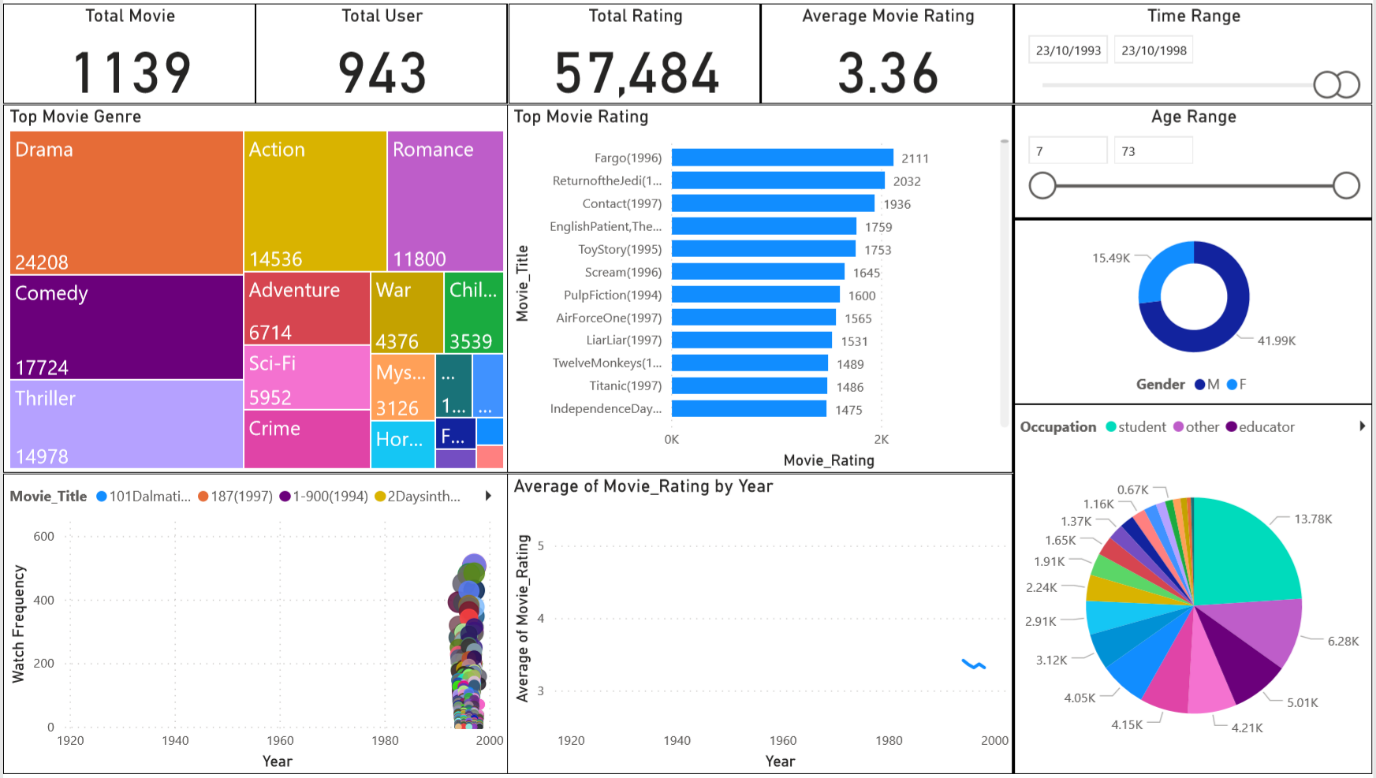


Figure 3.11 Power BI dashboard for MovieLens dataset (year 1993 to 1998)

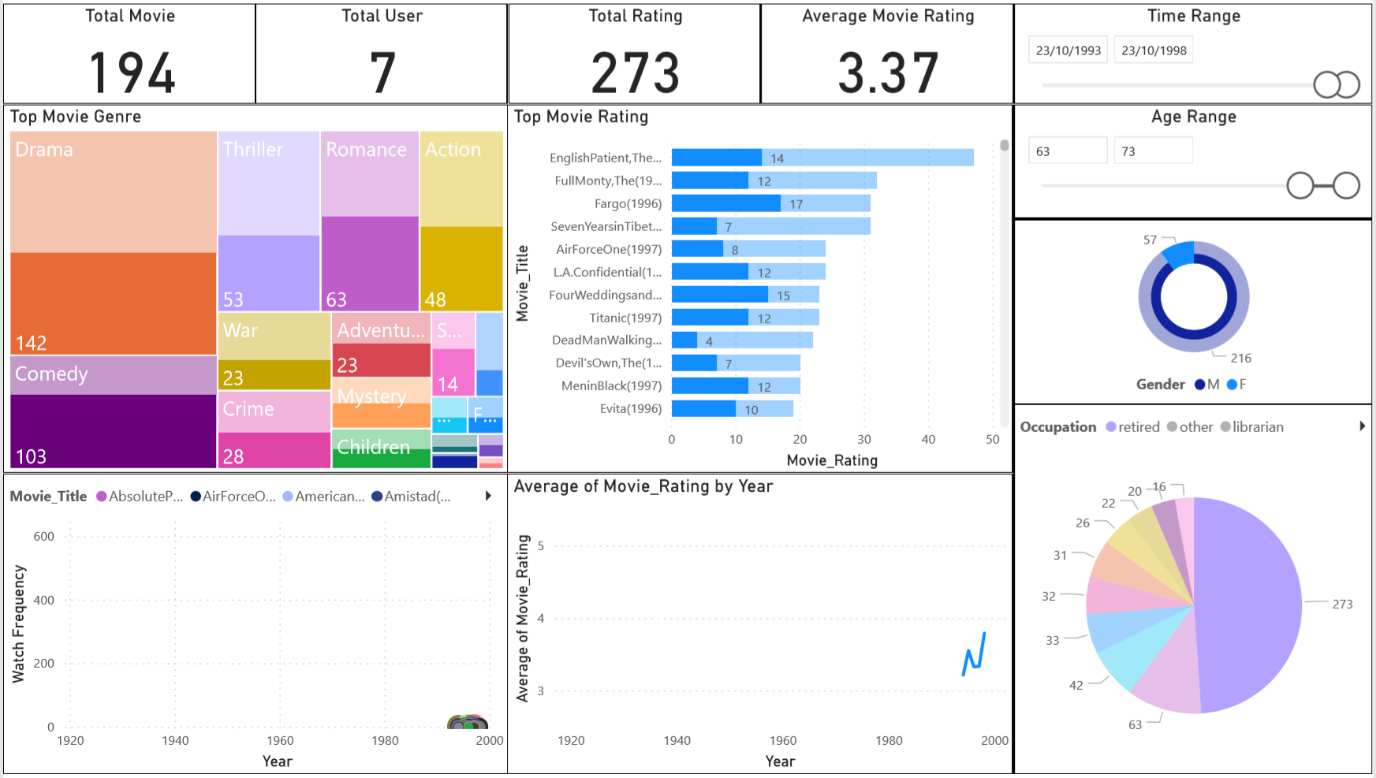


Figure 3.12 Power BI dashboard for MovieLens dataset (year 1993 to 1998 with age range from 63 to 73)

Besides that, the dashboard also provides the movie watch frequency by year from the scatter chart and average of movie rating by year from the line chart. This allow user to find out the most frequent movie viewed by user and highest average of movie rating. At the same time, it helps user to analyze what lead to the most frequent movie viewed by user and highest average of movie rating. For instance, by comparing figure 3.13 and 3.14 which figure 3.13 present the most frequent watch movie of year 1977 while figure 3.14 present most frequent watch movie of year 1997. Although there is gap of 20 years but both resulting the same output. Both figures indicated that the movie with highest view is mostly viewed by male user and also the user with student occupation. Moreover, the movie rating is not influence by the movie watch frequency according to both figures.

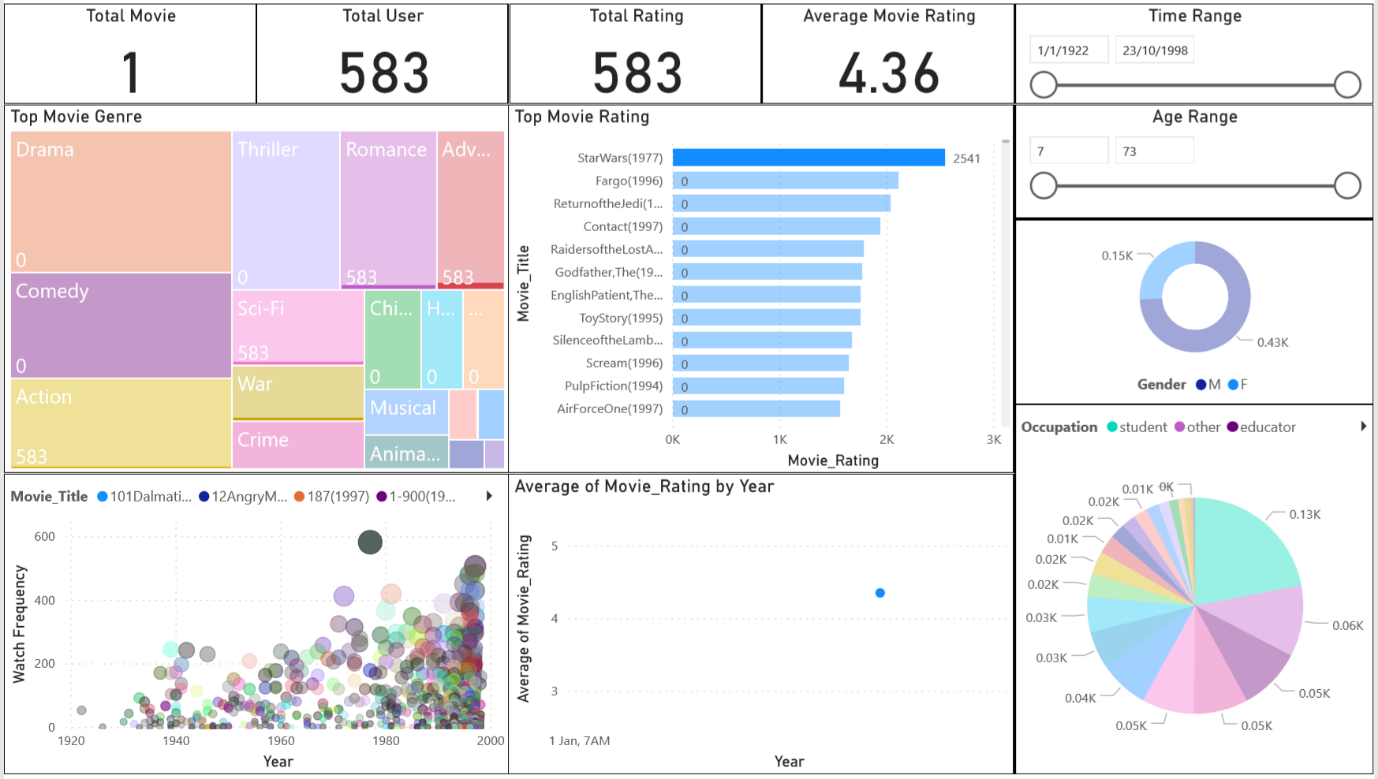


Figure 3.13 Power BI dashboard for MovieLens dataset (most frequent watch movie of year 1977)

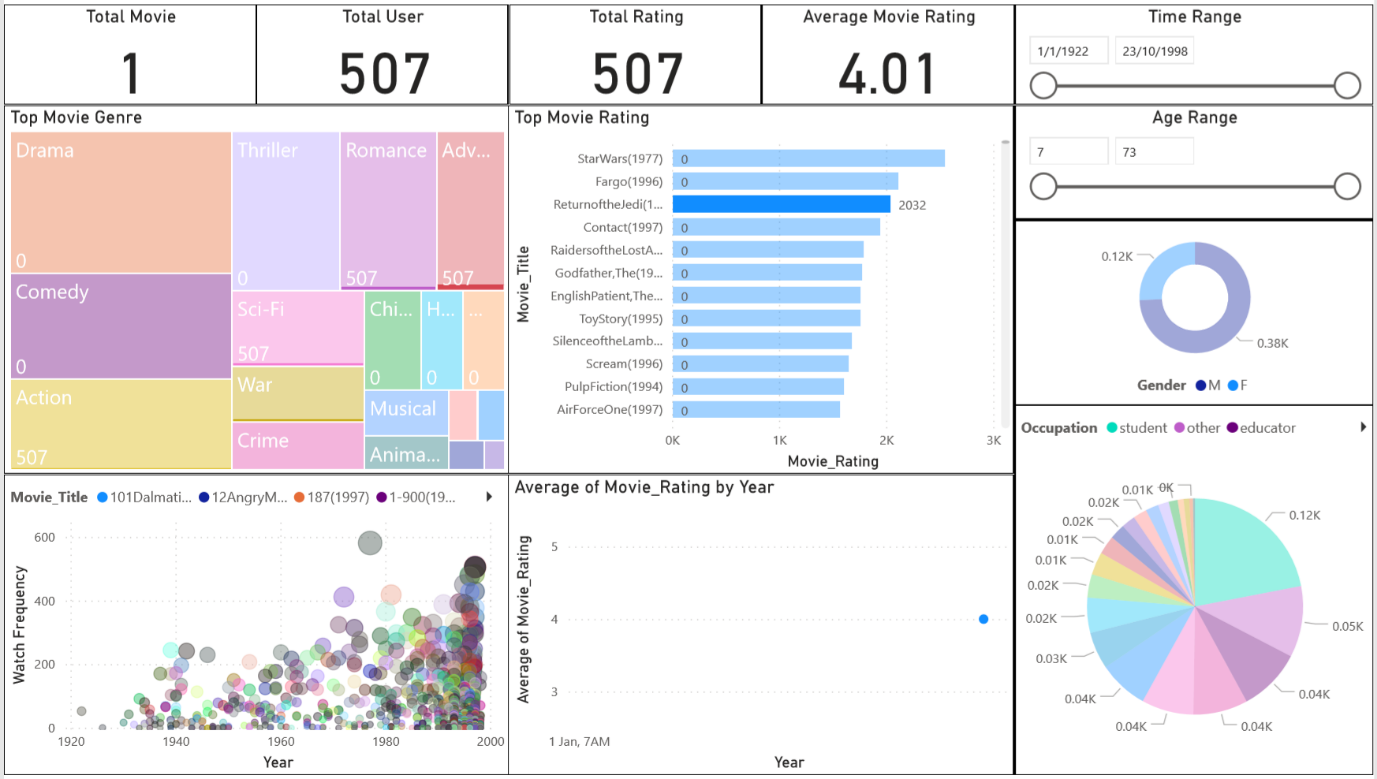


Figure 3.14 Power BI dashboard for MovieLens dataset (most frequent watch movie of year 1997)

**3.2.0 MACHINE LEARNING AND MOVIE RECOMMENDATION**

Apart from the descriptive analytics and visualization, This project looks into the machine learning paradigm of the Hadoop’s ecosystem. It is well known that online movie/video streaming sites provides movie/video recommendation and Machine Learning is the backbone of such feature. This section of the project uses Spark2 to deploy a machine learning technique in providing a movie recommendation toward a pseudo-user (a made-up user). As described earlier in the problem statement, online streaming service provider often lack of the information regarding the consumer group. In order to simulate such environment and test the machine learning algorithm capability, all demographic variables are omitted in this section of the project. Only user ID, movie ID and rating are used in the following section as most of the time, streaming provider only get to know the userID (individual identifier), movie they watched and the rating they provided. It is important to examine if machine learning could be functional under such tough environment.

The ALS machine learning algorithm is known to be able to work nicely under circumstances where a well define layers and predictors is absent. This characteristic indeed matched the situation of giving online content recommendation where streamer / streaming company often lack of detail information regarding user such as their demographic information or movie genre interest. Furthermore, the labelling and tagging of movie itself can easily become overwhelming as a movie could contains multiple genre. The boundaries and differences between movies aren’t really distinctive and discrete, laying any predictor built on top of that powerless. On the other side, method such as ALS focus on the similarities between 1 user and another, assuming users who are similar in terms of rating history should rate similarly for the unwatched other movie, resulting in a clean and functional method.

**3.2.1 ENSURING DATA IS UPLOADED INTO HDFS**

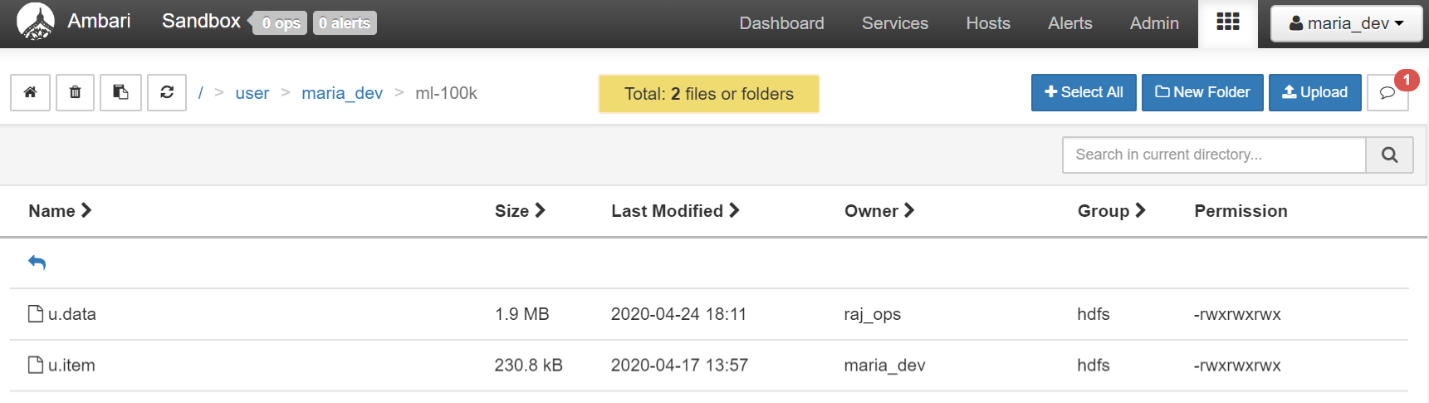


Figure 3.15 HDFS file view

The required datasets (namely u.data and u.item) are uploaded into HDFS located in the ml-100k directory prior to the machine learning as shown in figure 3.15. All required files are located in the HDFS to simulate an authentic cloud machine cluster environment. These files will later be used (called by coding) in the machine learning python coding.

**3.2.2 CODING**

The coding is separated into 4 main parts:

1. **Import relevant packages**

Pyspark is the main package used within the coding as it bridges the connection between Python and Apache Sparrk2. Apart from that, subprocess were used for certain file-reading related codes. Some process wasn’t available to HDP 2.5 natively and hence ‘pip install’ was used to fill up the missing packages.

1. **Define important user-defined-functions**

Two user-defined-functions were designated which will be used in the main coding.

|  |  |
| --- | --- |
| Function | Purpose |
| loadMovieNames | To load and transform u.item file into a structured format |
| PARSEINPUT | To convert u.data file into a structured format |

Table: User-Defined-Functions

1. **Main ML process**

The main ML process covers most of the coding designated. These process involve load and transform the u.item and u.data files in multiple way and several times in order to extract a structured data frame or exporting a list of useful information for later programming. The structured data frame is later act as an input toward the machine learning function (namely ALS).

1. **Cleaning ML result**

The cleansing of the ML result focuses on removing the movie that duplicates with the user 0 rated movie. The logic behind this is that a video / movie recommendation system should recommend new contents to the user instead of the watched one.

1. **Output/print result**

From the command prompt perspective, only 2 outputs are designated. The first output will print the targeted user (user0 in this case)’s watched movie and the respective rating. The second output will print the list of movies that is recommended by the machine learning algorithm.

**3.2.3 PSEUDO-USER 0 AND ITS PERSONA**

A user 0 is created in order to serve as the baseline of evaluation. There are 6 pseudo-entries for user 0 are shown in table 3.2:

Table 3.2 The 6 pseudo-entries for user 0

|  |  |  |
| --- | --- | --- |
| Movie ID | Movie Name | Rating |
| 1 | Toy Story (1995) | 5 |
| 95 | Aladdin (1992) | 5 |
| 71 | The Lion King (1994) | 5 |
| 750 | Amisted (1997) | 1 |
| 22 | Braveheart (1995) | 1 |
| 28 | Apollo 13 (1995) | 1 |

The characteristics or persona of the fabricated user 0 is straight-forward: A user who likes cartoon movie and dislike historical movie. These entries will be compared with other user’s movie ratings, resulting in a list of movies that is rating highly among user who have similar rating output as user 0, excluded the movies that is within the user 0 rated list as described in (d.)

**3.2.4 DEPLOYMENT AND RESULT**

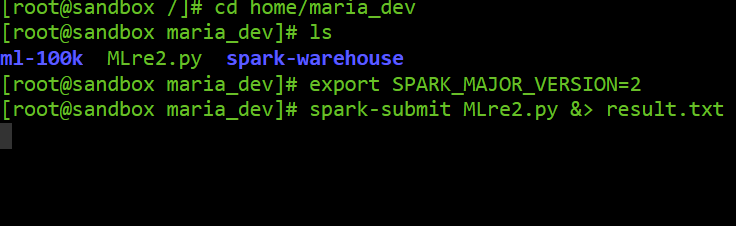


Figure 3.16 Linux commands on deployment

Figure 3.16 the process of deploying the ALS Machine learning python file of the HDP 2.5 command prompt (Linux RedHat based). Firstly, user root was used to logged into the command prompt shell and the directory is switched to the directory containing the MLre2.py machine learning Python file. Then, a command of changing default SPARK version to 2 is used in order to allow the system to use SPARK 2 when interacting using spark-submit. Lastly, the spark-submit command was used to initiate the machine learning process and the result is outputted to results.txt in the same directory. Alternatively, the output can be printed in the command prompt as the result too as shown in figure 3.18. The file named results.txt containing the printed version of the output file can be view in appendix 2.

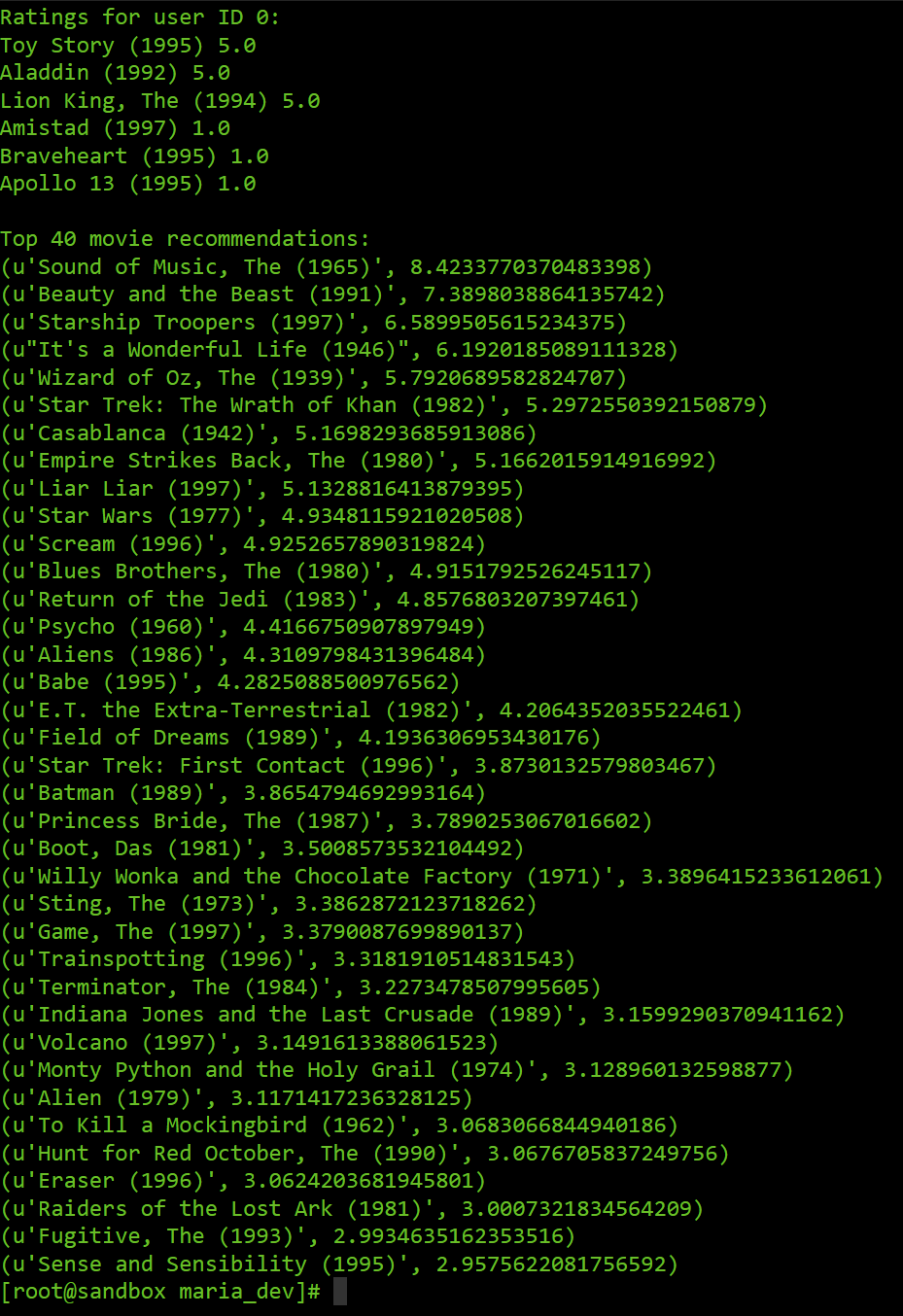


Figure 3.17 Part of the machine learning output printed on the Linux shell

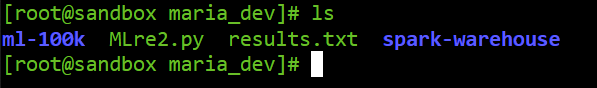


Figure 3.18 Example of stored output in file named results.txt

Based on the output of the result as shown in figure 3.17, most of the highly recommended titles are cartoon movie such as Beauty and the Beast, The Wizard of Oz and Casablanca. Along with the cartoon movie recommendation, SciFi (Science-Fiction) movie seems to be highly populated along the entire list, suggesting user who prefer cartoon movie are liking SciFi movies. Across the list, no historical / documentation related movies were recommended besides a few drama movies listed at the lowest section of the list. Further analysis explains the phenomenon can be due to the supply of cartoon movie back then wasn’t saturated, resulting user in the 90s era tends to combine watching experience with other fantasy genre (e.g. science fiction and magic fantasy movies). Nevertheless, this group of users who are similar to the fabricated user 0 doesn’t seems to be in any liking of historical and documentation related movies. Other movies genre such as horror, comedy, adventure and action doesn’t seems to have any significant theme besides some exception along the list, suggesting a distinctive user-0 related group that has their own unique movie preference back in 90s. Lastly, while they are many optimization can be done and even more algorithm selections that may achieve higher accuracy, the ALS machine learning technique’s result shows that ALS is a reliable and functional technique in the given domain and more importantly, being supported by Apache SPARK 2 under the Hadoop ecosystem would enable the possibility of parallel processing and file distribution. Future study may focus on exploring the SPARK2 supported machine learning algorithm in optimizing the search result.

**CHAPTER 4**

**CONCLUSION**

**4.0 SUMMARY**

This paper explores the online streaming service provider industry and the challenges they are facing. Research objectives were coined to guide the research process. Literature regarding the related tool and application that are involved in the development of solutions are reviewed to ensure highest level of understanding toward the features and functions of the toolkits. Furthermore, this research’s outputs included a dashboard, descriptive analytics and a movie recommendation tool utilizing ALS machine learning method.

**4.1 CONCLUSION**

As the conclusion of this research project, Hadoop as advertised, is a comprehensive platform in storing as well as processing big data. It HDFS framework of file storing prevent data lost due to hardware issues. Data processing tool that built on top of MapReduce and YARN such as SPARK 2 is able to execute complicated coding such as a machine learning algorithm and output result in a flexible way. The machine learning algorithm provided in this paper is exploratory, further work and research can be done in order to polish and optimize the result as well as evaluating the result in a statistical manner.

**REFERENCES**

Gupta, P. (2016). Rating based mechanism to contrast abnormal posts on movies reviews using MapReduce paradigm. *International Conference on Information and Communication Technologies, 5*, 7-9.

IBM. (2019). *What is HDFS?* IBM.

Jain, A. (2016). Movie Analytics for Effectiveness. *International Journal of Rough Sets and Data Analysis, 3*(2), 83-85.

Joseph, R. (2020). *What is Spark?* Kearny: Chartio.

Kadam, S. (2016). Big data analytics-recommendation system with Hadoop Framework. *International Conference on Inventive Computation Technologies (ICICT)*, 1-5.

Khayer, A., Talukder, M. S., Bao, Y., & Hossain, M. N. (2020). Cloud computing adoption and its impact on SMEs’ performance for cloud supported operations: A dual-stage analytical approach. *Technology in Society*, *60*, 101225. https://doi.org/10.1016/j.techsoc.2019.101225

Li, J., Xu, W., Wan, W., & Sun, J. (2018). Movie recommendation based on bridging movie feature and user interest. *Journal of Computational Science*, *26*, 128–134. https://doi.org/10.1016/j.jocs.2018.03.009

Narendra, B. (2016). Sentiment Analysis on Movie Reviews. *Intelligent Systems and Applications, 8*, 66-70.

Rajurkar, G. D. (2015). A speedy data uploading approach using Hadoop. *International Conference on Computing Communication Control and Automation*.

Wei, S., Zheng, X., Chen, D., & Chen, C. (2016). A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications*, *18*, 83–94. https://doi.org/10.1016/j.elerap.2016.01.003

Yang, J., & Yecies, B. (2016). Mining Chinese social media UGC: a big-data framework for analyzing Douban movie reviews. *Journal of Big Data, 3*, 19.

**APPENDICES**

**APPENDIX 1: Link to Complete Files**

Google Drive Link that contains complete working file:

https://drive.google.com/drive/folders/1\_2\_7xasB15CkKIOf\_lZ6ZyY-W85UUvTb?usp=sharing

**Appendix 2: Content of result.txt**

SPARK\_MAJOR\_VERSION is set to 2, using Spark2

/usr/hdp/2.5.0.0-1245/spark2/python/lib/pyspark.zip/pyspark/sql/context.py:477: DeprecationWarning: HiveContext is deprecated in Spark 2.0.0. Please use SparkSession.builder.enableHiveSupport().getOrCreate() instead.

/usr/hdp/2.5.0.0-1245/spark2/python/lib/pyspark.zip/pyspark/sql/context.py:477: DeprecationWarning: HiveContext is deprecated in Spark 2.0.0. Please use SparkSession.builder.enableHiveSupport().getOrCreate() instead.

Ratings for user ID 0:

Toy Story (1995) 5.0

Aladdin (1992) 5.0

Lion King, The (1994) 5.0

Amistad (1997) 1.0

Braveheart (1995) 1.0

Apollo 13 (1995) 1.0

Top 40 movie recommendations:

[Stage 51:==========> (6 + 4) / 10][Stage 55:> (0 + 0) / 10][Stage 55:===========> (2 + 3) / 10][Stage 55:============================> (5 + 3) / 10][Stage 53:==================> (66 + 4) / 200][Stage 53:=========================> (93 + 4) / 200][Stage 53:==================================> (129 + 4) / 200][Stage 53:==========================================> (158 + 4) / 200][Stage 53:===================================================> (191 + 4) / 200][Stage 56:=======> (28 + 4) / 200][Stage 56:===============> (55 + 4) / 200][Stage 56:=======================> (85 + 4) / 200][Stage 56:============================> (107 + 4) / 200][Stage 56:======================================> (144 + 4) / 200][Stage 56:===============================================> (176 + 4) / 200][Stage 57:====================> (75 + 6) / 200][Stage 57:=========================> (93 + 4) / 200][Stage 57:===================================> (130 + 4) / 200][Stage 57:============================================> (165 + 4) / 200][Stage 57:=============================================> (167 + 4) / 200] [Stage 77:===============> (58 + 4) / 200][Stage 77:===================> (72 + 4) / 200][Stage 77:=======================> (85 + 5) / 200][Stage 77:===========================> (102 + 4) / 200][Stage 77:==================================> (127 + 4) / 200][Stage 77:==========================================> (159 + 4) / 200][Stage 77:===================================================> (190 + 4) / 200] (u'Sound of Music, The (1965)', 8.4233770370483398)

(u'Beauty and the Beast (1991)', 7.3898038864135742)

(u'Starship Troopers (1997)', 6.5899505615234375)

(u"It's a Wonderful Life (1946)", 6.1920185089111328)

(u'Wizard of Oz, The (1939)', 5.7920689582824707)

(u'Star Trek: The Wrath of Khan (1982)', 5.2972550392150879)

(u'Casablanca (1942)', 5.1698293685913086)

(u'Empire Strikes Back, The (1980)', 5.1662015914916992)

(u'Liar Liar (1997)', 5.1328816413879395)

(u'Star Wars (1977)', 4.9348115921020508)

(u'Scream (1996)', 4.9252657890319824)

(u'Blues Brothers, The (1980)', 4.9151792526245117)

(u'Return of the Jedi (1983)', 4.8576803207397461)

(u'Psycho (1960)', 4.4166750907897949)

(u'Aliens (1986)', 4.3109798431396484)

(u'Babe (1995)', 4.2825088500976562)

(u'E.T. the Extra-Terrestrial (1982)', 4.2064352035522461)

(u'Field of Dreams (1989)', 4.1936306953430176)

(u'Star Trek: First Contact (1996)', 3.8730132579803467)

(u'Batman (1989)', 3.8654794692993164)

(u'Princess Bride, The (1987)', 3.7890253067016602)

(u'Boot, Das (1981)', 3.5008573532104492)

(u'Willy Wonka and the Chocolate Factory (1971)', 3.3896415233612061)

(u'Sting, The (1973)', 3.3862872123718262)

(u'Game, The (1997)', 3.3790087699890137)

(u'Trainspotting (1996)', 3.3181910514831543)

(u'Terminator, The (1984)', 3.2273478507995605)

(u'Indiana Jones and the Last Crusade (1989)', 3.1599290370941162)

(u'Volcano (1997)', 3.1491613388061523)

(u'Monty Python and the Holy Grail (1974)', 3.128960132598877)

(u'Alien (1979)', 3.1171417236328125)

(u'To Kill a Mockingbird (1962)', 3.0683066844940186)

(u'Hunt for Red October, The (1990)', 3.0676705837249756)

(u'Eraser (1996)', 3.0624203681945801)

(u'Raiders of the Lost Ark (1981)', 3.0007321834564209)

(u'Fugitive, The (1993)', 2.9934635162353516)

(u'Sense and Sensibility (1995)', 2.9575622081756592