1. **Introduction**

People have recognized the importance of data and information from a very early age. Having access to timely and comprehensive data and being able to gain insights from them have been proven extremely valuable in not only business and technology researches, but also many other fields. Therefore, data management has always been an important skill. In today’s era of information explosion, the potential available data is experiencing exponential growth. A 2000 study (Gray and Shenoy, pp.9) concluded that, with data stores grows huge, our biggest challenge will be “*to make it easy to access and manage them… automating all the tasks of data organization, accesses and protection*”. Data Engineering, as the practice designing and building systems for collecting, storing, and analysing data at scale (Coursera, 2021), is therefore of significant value.

One particular industry that demands more attention to data engineering is the film industry. Ever since the first tickets to “moving pictures” was sold by Lumière brothers in 1895, films has been not only a charming dream-making art, but also a overall immensely profitable tool. According to Forbes, the film industry has managed to break $100 billion for the first time in 2019, to which the box office income contributed nearly 50% (Escandon, 2020). However, with the continuous development of modern film technology and the improvement of the public aesthetic requirements, the production cost of film has been rising year by year. As a result, the risks that film distributors and producers have to take on have also largely increased.

This report aims to help with such problem with data engineering skills by collecting data on the IMDb top 250 rated films from multiple sources and build a data processing system for future utilities. Although the top 250 rated film list from IMDb has been used by many analysts, it is still worth looking at since IMDb is still the most popular film rating website around the world so that their data is less likely to be biased. Moreover, most researched using the top 250 list primarily focus on the rating and public praise of the films. While in this report, combined with data from Box Office Mojo, a IMDb Pro service website that records the box office income, the films are being examined from a more economic and business perspective.

In this report, web scraping by BeautifulSoup and regular expression are used for data mining from relevant websites. The collected data are then further cleaned and transformed for analyse and storage. A querying system based on PostgreSQL is then constructed, which can be used by film distributors for strategy and decision making reference. Besides, machine learning pipeline and models are also trained for predictive analytics and converted into APIs by Flask, allowing easier usage for non data analysts in the future.

1. **Data Collection**

Firstly, all libraries and packages used in this report are imported at once.

In terms of data processing, Apache Spark is installed at the very beginning to utilize in-memory caching, and optimize query execution.

Note that in the exported pdf version, the line of spark installing is commented out after it is successfully installed for the connivence of reading because the output of this cell is extremely long (taking up over 10 pages) and does not provide much useful information. In the notebook version, this cell is run as normal.

As mentioned above, the dataset is contains two parts, the basic film information from IMDb top 250 list, and box office income data from Mojo.

* 1. **General Film Information from IMDb**

The source for the top 250 films is: <http://www.imdb.com/chart/top>. As demonstrated in the screenshot below, the page displays only the rank, title, year of first release and the IMDb rating score. But looking at the html files behind the page, it also contains the information of “crew names”, which includes the name of the director and two leading actors and/or actresses.

When choosing the data source, another “IMDb Top 1000 movies” (available at https://www.imdb.com/search/title/?count=100&groups=top\_1000&sort=user\_rating) list was also noticeable (see screenshot attached below). From the appearance, this list contains a few more details on each film including genre, duration and brief introduction. However, this list seems to be more real-time updated, which results in very new films which has only less than 1/100 votes of the others making high positions. The list in therefore highly likely biased. Therefore, the top 250 list is used instead. And the missing information will be retrieved from other sources later.

More importantly, whichever list used, the data mining step is similar. Simply by changing the url link and a few tags in the web scraping code used below, it can then be applied to other similar IMDb pages.

The data scraping in this report is mostly done by using BeautifulSoup, a widely used and recognised library for pulling data out of HTML and XML files in Python. Although it does have the flaw of running relatively slower when performing certain heavy data tasks, the problem can be overcome with the assistance of the multithreading concept. And compared to other web scraping libraries such as Selenium and Scrapy, BeautifulSoup is much more user-friendly and it caters to efficient and comprehensive documentation (Arsalan, 2021).

For this project, the basic information, unique link (serving as the ID number of each film on IMDb database), crew information (as mentioned above), and the rating scores of each film among the top 250 are first extracted from the web page using BeautifulSoup and response. Title, year and ranking place are then further extracted from basic information. Crew information are split into director and leading cast. For each film, a dictionary is created for the aforementioned information. Finally, using the title as index, lists for each kind of data are generated and converted into a pandas data frame.

The code used in this section are inspired by and partially cited from @Priyank181 on GeeksforGeeks.org (Priyank181, 2021).

* 1. **Box Office Income Data from MOJO**

As discussed above, one of the main objectives of this report is to provide a more business related point of view using data for film companies. Therefore, a more important dataset is the box office income data. The source used here is Box Office Mojo (<https://www.boxofficemojo.com/>). Because it is a service provided also by IMDb, they share the same database for film information. Therefore, using the unique link for each film extracted from previous step, it is very easy to navigate to the detailed box office income page for each film in the data frame.

Take “*The Shawshank Redemption*” as an example, inserting its unique link “*/title/tt0111161/*” into the url link, the detailed information page will show up. As the screenshot below shows, this page includes not only the domestic, international and worldwide box office income data, but also the information missing from the previous top 250 list such as genre, duration and introduction. Other details such as distributor and release date are also included.

To gather these data, BeautifulSoup is used again as before. Besides, since the html files behind Mojo is written differently than the top 250 list, regular expression is also used this time.

It is worth noticing that, all the money related data on this page including three box office numbers, opening income and the budget all share a span tag with the exactly same class: “money”. Therefore, they are collected together as a list names earnings, and then using indexing to extract different income separately. The intention at first was to extract all 5 numbers for each film. However, due to realistic limitations, a large number of films among the 250 are missing one or more of the 5 numbers, mostly the “domestic opening” and the “budget”. This may have be caused by the incomplete information disclosure of film companies. Therefore, the opening and budget columns are dropped in this report. Another problem is that, this top 250 includes some very old films which even date back to early 1920s. Some of these old films have even less than three numbers in the earnings list as they were only released domestically and thus missing international income as well. The solution for that is to constrain the length of the earnings list for each film. Only those with more than three earning numbers are further processed for now. While the others will be manually fixed later in the data transformation section.

Apart from the income data, genre, duration and distributor of each film are also extracted using regular expression. A small number of films have more then one labels in genre column. Only the main genre (the first) is considered. Similarly, for a few of those missing the distributor information, they are filled with zeros instead.

All the data collected from this section are merged into the basic information data frame from before after a few initial cleaning. For example, genre “*Drama*”, “*Western*”, “*Horror*” and “*Comedy*” are tagged slightly different from other genres in html files and needs manually dropping extra punctuations. Moreover, the unit of all three income columns are changed from dollars to thousands of dollars because otherwise, some of their incomes will be too large to write into PostgreSQL database in later steps.

Lastly, the pandas data frame are converted to parquet format for storage after adjusting the data type of each column. The rational for using parquet format will be further explained in section 6.

1. **Data Transformation**

Before the dataset can be further analysed and write into databases, some data cleaning and transformation are needed.

* 1. **Duration Time**

Firstly, the duration column needs to be changed. Currently, for each film the duration is a string including numbers and words of “hr” and “min”. It is hard to process time related data in that format. Two transformation options are: turning this column into timestamps, or calculate the duration in mins. Considering the traditions in the film industry, represent running time of a film in mins are more sensible.

The idea was to add 60 times the hour digit to the mins. However, a few rows of films are having special durations, less than one hour (leads to no hour digit) or running for exactly whole hours (results in no minute digits). Therefore, before they can be transformed into minutes, such special ones need to be located and manually fixed by adding zeros to relevant digits.

After that, the “hr” and “min” are dropped and replaced by only a colon in between as the indicator of string split in the next step. Finally the string is split and hours are calculated into mins.

* 1. **Income Proportions**

For any industry, understanding the market and client segments are always one of the top priorities. Films, as an entertainment methods that largely based on culture, can receive very various feedbacks from different areas. Therefore, besides the gross income numbers, the proportion of domestic and international box office are also valuable information, especially for those considering opening overseas markets.

With domestic, international and worldwide incomes data being collected already, the calculation of percentage appears to very simple. However, as mentioned above in section 2.2, there are potential problems with these three columns, and some films are still missing relevant data. Therefore, after dividing both domestic and international incomes by worldwide incomes, an extra “check” column is created. Theoretically, since worldwide incomes are made up by only two parts, adding up the dom\_pct and int\_pct should get results of 1. However, as some films missing one or more money related data from the beginning, there are several problems causing by column mismatch. For example, for some films which has no international incomes, their openings or budget might be mistaken for worldwide income. Such rows are located by filtering “check” column equals to one and then manually fixed.

Similarly, for those old films has only domestic incomes as discussed above, their income columns were filled with zeros at the beginning. Here they are also found and completed manually.

After sorting out the three income columns, the two percentages are re-calculated and the add-up-to-one check is also re-run. This time, apart from the 11 films which are missing all income data for realistic reasons, almost all rows have a check value of 1. But there are still a few films having check number of between 0.85 and 0.99 while all three income numbers are correct. Taking a closer look at them, they are all very popular IPs distributed by Walt Disney or Twentieth Century Fox, which was acquired by Disney in 2020 (BBC News, 2020). A reasonable explanation for this is that another part of the worldwide income of these films comes from the Disney plus streaming media platform, which are therefore not included in box office income.

Finally, the “check” column is dropped since it’s no longer useful.

* 1. **Rankings**

Lastly, a few more rankings are added apart from the initial ratings ranking from IMDb list.

The added columns includes ranking by duration time and ranking by worldwide box office income. Moreover, since the genre is a very important feature in the film industry. The genre of a film sometimes can largely influence it’s style, audience segment and box office performance. Therefore, the in-genre rankings of all the factors mentioned above including the ratings place, duration and worldwide income, are all generated separately as well.

Lastly, the difference of the “place” column and the “income\_rank” column for each film is calculated as the “rank\_diff”. This metric is intended to detect the gap between a film’s economical performance (box office incomes) and public recognition (ratings). If the rank\_diff value is positive, it indicates that despite being highly praised, this film is not very profitable, or at least not as profitable as it should be. On the other hand, if the rank\_diff value is negative, the film might be commercially successful, but it’s reviews does not match it’s profits. The larger the number is, the stronger the imbalance is. And such imbalance, on either side, can both be bad news for film companies. A extreme mismatching of ratings and incomes may lead to financial losses or reputation damages.

* 1. **Final Cleaning**

At last, the data frame is checked for missing values. And the ten missing values found are filled with zeros. The order of the columns are also adjusted for the connivence of reading. The final version of the film information data frame is then converted into parquet format for storage.

1. **Machine Learning with API**

A machine learning pipeline is built in this section for model training and data deployment purpose in thus section. The model trained here is a widely used logistic regression model. The machine learning model is then further converted to be served with API to improve the usability of this approach by others in the future.

* 1. **Data Pre-processing**

Before training the machine learning model, data pre processing is conducted to generate suitable dependent and independent variables.

Columns that are not suitable for logistic regression such as director and cast are removed. Although they can be transformed into dummy variables, it would result in too many X variables with no significance for the model. Besides, all the ranking variables are also removed as they are highly correlated with rating and will cause multicollinearity. Only the worldwide income column is retained for the same reason.

Distributor and genre are converted into dummy variables. The distributors are divided into two groups, top 10 distributors, (*Warner Bros., Walt Disney Studios Motion Pictures, United Artists, Paramount Pictures, Universal Pictures, Columbia Pictures, Twentieth Century Fox, Miramax, Metro-Goldwyn-Mayer (MGM) and Sony Pictures Classics*) whose films make up nearly 65% of the list. The variable will be assigned the value 1 if the film is produced by one of the top 10 distributors and 0 if not.

Next, a binary dummy variable is created for each genre. Considering there are only very few films in the five most least seen genres (*Animation, Horror, Mystery, Western and Film-Noir*), they are combined into one category called “other\_genre”.

At first a linear regression was fitted, trying to predict the exact worldwide income of each film. However, since the information are not comprehensive enough, the error of prediction was large. Therefore, a binary logistic regression to predict whether the film will be a commercially success is trained instead. The standard of “success” is defined by whether the worldwide income of the film reaches the median of the worldwide income of all 250 films.

The new data frame for the machine learning pipeline is defined as df\_ml, and it is also converted into parquet format for storage.

* 1. **Machine Learning Model**

The logistic regression model is trained by scikit learning library. Train and test sets are split at 80%. The model is fitted with the train set and used to generate predicted outcome for the test set. The model performance is measured by the prediction accuracy. Although 250 rows of data seem to be not very sufficient, The model does show an accuracy of nearly 65% for the test set. It is reasonable to believe that given more data available, the model can perform better.

* 1. **Serve with API**

The more important task is to convert and serve the above machine learning model with API. By applying an API first approach, cross-language applications of the machine learning models will be largely simplified and thus improve the general usability of the models.

The joblib package from scikit learn is first used to save the model by serialization. After that, Flask is used to create the API. One small flaw here is that jupyter notebook is not very suitable for running such APIs and the “use\_reloader” needs to be manually set to false.

1. **PostgreSQL Schema and Database**
   1. **Schema Structure**

In order to store and manage the data collected in PostgreSQL database, a schema is designed for this research. The .sql file is attached in the zip file uploaded with this report. And the structure of the schema is demonstrated as the figure below.

* 1. **Prepare Subset Tables**

Before writing the data into PostgreSQL database, the subset tables designed in the schema are all prepared separately and stored as parquet files.

* 1. **Write into PostgreSQL Database**

The schema is written into the PostgreSQL database by a line of shell script and the selected data are all written into the database using spark.write.jdbc.

* 1. **SQL Query Example**

As discussed above, one of the objectives of this report is to build a data querying system which can help gain insights to the current film industry. After writing the collected data into database, such purpose can be achieved by performing SQL queries. This section gives one example of using SQL query for data manipulation and understanding.

The example here is to calculate the average performance of films distributed by different companies. The mean value of each distributor’s films’ rating rankings, worldwide box office income rankings and the differentials between the two values are calculated. Using the aforementioned rule to interpret the “rank\_diff” value, sorting the output by “average rank\_diff” value, it is clear that the top companies tend to produce highly recognised but less profitable films, while others are the opposite. Such cross comparison is very useful for a film company to identify its value proposition and position in the industry.

Other queries are also available and valuable. For instance, the companies can check their films’ performance by genre, which allows them to understand their strength and weakness. Or the database can also be used to test the public recognition and commercial value of different directors and actors/actresses.

1. **Data Storage Control and Lineage**
   1. **Data Storage**

In this report, all datasets collected are converted to the parquet format for storage in order to make data more tractable and manageable. Apache Parquet is an open-source, column-oriented data storage format. Another similar high quality data storage format is avro. However, since avro is more row-oriented, parquet is used in this report. In parquet format, values are clustered by columns. Therefore it can not only provide efficient data compression and encoding schemes with enhanced performance, but also properly sort and retrieve row order. All these aforementioned advantages of parquet format prove to be useful for large scale data storage, which improves the usability of the system built in this report even with larger data volume.

Moreover, as introduced in section 5, all datasets are eventually divided into eleven different tables and wrote into PostgreSQL database for efficient operation and maintenance.

* 1. **Source Version Control**

Although this is an individual project, good source version control is still important as it helps to organise and keep track of the development of codes and files. Git and GitHub are used for source version control by pushing commits. When version of files updated, new commits will replace the old ones.

The link of GitHub repository is: https://github.com/HermioneSun0321/DataEngineering

* 1. **Data Lineage**

Data lineage is the process of understanding, recording, and visualizing data as it flows from data sources to consumption throughout the project, including all the data transformations. It helps to track errors, lower risks and ensures accuracy. It is a process that is particularly important in large scale data manipulating and teamwork collaborations (Imperva, 2021).

In this project, Data Version Control (DVC) is used for data lineage just like in previous group assignments. After each data collection and transformation throughout the project, check points are made. Screenshots of the auto script for running the process is demonstrated below. The original script files are also included in the zip folder uploaded with the final report.

* 1. **Automate Terraform with GitHub Actions**

Terraform with GitHub actions are also set up for automation.

1. **Conclusion**
   1. **Main Takeaways**

This report started with gathering basic information on the IMDb top 250 rated films using web scraping with BeautifulSoup. Box office income data for the 250 films are also gathered from Mojo, another source. Data cleaning and transformations carried out to prepare for analysis. For the purpose of viewing the top rated films from a commercial perspective, a machine learning predictive models is built and converted to API for the convince of future usage by others. Last but not least, a systematic querying structure is designed and constructed on collected data and wrote into PostgreSQL database. During the entire process, uniformed data storage formats are utilised, data lineage process is also conducted to ensure the data accuracy.

In sum, this report managed to provide an approach to gain insights into the business value creation of the film industry by data engineering skills.

* 1. **Limitations and Future Steps**

Admittedly, there are certain limitations to this project. Firstly, 250 films do not make a very sufficient dataset for analysing, model training or pattern discovering. And the top 250 films list from IMDb covers only highly rated films, which, even from a commercial perspective, are potentially biased as they are more likely to earn more box office incomes. Moreover, due to time and space limitations, the top user reviews data collected are not presented in this report. Given more time and space, they can be used to carry out a sentiment analysis to examine the public opinion on each film.

For future steps, more data from different aspects, especially from professional film industry point of views, needs to be collected. Finally, the data system constructed in this report can be generalised to other entertainment business such as TV shows and video streaming platforms.

**Reference**

Arsalan, 2021. [online] Available at: <https://it-s.com/best-web-scraping-library-scrapy-vs-beautiful-soup-vs-selenium/> [Accessed 20 April 2022].

BBC News. 2020. Disney ends the historic 20th Century Fox brand. [online] Available at: <https://www.bbc.co.uk/news/business-53747270> [Accessed 21 April 2022].

Coursera. 2021. [online] Available at: <https://www.coursera.org/articles/what-does-a-data-engineer-do-and-how-do-i-become-one> [Accessed 19 April 2022].

Escandon, R., 2020. *The Film Industry Made A Record-Breaking $100 Billion Last Year*. [online] Forbes. Available at: <https://www.forbes.com/sites/rosaescandon/2020/03/12/the-film-industry-made-a-record-breaking-100-billion-last-year/?sh=10843e0f34cd> [Accessed 20 April 2022].

Gray, J. and Shenoy, P., 2000, February. Rules of thumb in data engineering. In *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)* (pp. 3-10). IEEE.

IMDb. n.d. Top 250 Movies. [online] Available at: <https://www.imdb.com/chart/top> [Accessed 20 April 2022].

Imperva. 2021. *What is Data Lineage | Examples of Tools and Techniques | Imperva*. [online] Available at: <https://www.imperva.com/learn/data-security/data-lineage/#:~:text=Data%20lineage%20is%20the%20process,%2C%20what%20changed%2C%20and%20why.> [Accessed 23 April 2022].

Priyank181, 2021. *Scrape IMDB movie rating and details using Python - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/scrape-imdb-movie-rating-and-details-using-python/> [Accessed 20 April 2022].