CMPT 353 Project

Simon Fraser University
Michael Fischer, Ilia Fatemi

Table of Content

Report Overview

In this report we will be looking at how different factors affect a movie's profitability so that we can make informed decisions on which types of movies to create. Alongside this we will be creating a model to see how accurately we can predict a movie's revenue so we can gauge how well our future movies will do.

Data Gathering, Cleaning, and Integration

Overview

The primary datasets were downloaded from the IMDb and additional information was fetched using The Movie Database (TMDb) API. The goal was to compile a complete database containing various attributes of movies, including budget and revenue information.

Data Source

The primary datasets were downloaded from the IMDb and additional information was fetched using The Movie Database (TMDb) API. The goal was to compile a complete database containing various attributes of movies, including budget and revenue information.

IMDb Database

The following compressed files were downloaded from the IMDb dataset page: IMDB Dataset. Information about the column names and data structure is available at the IMDb Developer page: IMDb Developer.

1. **title.basics.tsv.gz**: This file contains information for title, primary title, original title, start year, end year, runtime, and genres.
2. **title.akas.tsv.gz**: This file contains alternative names for titles, along with attributes such as region, language, and type.
3. **title.ratings.tsv.gz**: This file contains the IMDb rating and number of votes for titles.

These files were not included in the GitHub repository due to its large size.

TMDb API

Since the IMDb contained id's for each movie, we were able to get additional details like the budget and revenue for each movie fetched using the API. Documentation for this API is linked here: TMDb API Documentation. An authorization token was required to access the API, which is included in the headers in the file `fetch_movies.py` for the API requests.

Data Cleaning and Integration

The following steps were performed to clean and integrate the data:

1. **Reading the IMDb Data**:
    a. The `*.tsv.gz` files were read into pandas `DataFrame`.

2. **Merging DataFrame**:
    a. The `title_basics_df` and `title_akas_df` were merged on `tconst` and `titleId`.
    b. The resulting DataFrame was then merged with `title_ratings_df` on `tconst`.
3. **Filtering Data**:
    a. The merged DataFrame was filtered to only include movies in the US-region with a start year of 1860 or later.
    b. Only `titleType` of 'movie' was retained.
    c. Duplicates based on `primaryTitle` were removed to ensure unique entries.
4. **Dropping Irrelevant Columns**:
    a. Columns that were not necessary for the final analysis were dropped.
    b. These columns include: `filtered_df.drop(columns=['titleId', 'language', 'originalTitle', 'isAdult', 'ordering', 'title', 'attributes', 'types', 'isOriginalTitle', 'region', 'endYear', 'titleType'])`
5. **Fetching Budget and Revenue Data**:
    a. For each movie, budget and revenue details were fetched using the TMDb API.
    b. The API responses were integrated into the Dataframe as new columns.
    ```
    filtered_df[['budget', 'revenue']] =
    filtered_df.apply(lambda id:
    movies.getDetailsById(id['tconst']), axis=1,
    result_type='expand')
    ```
6. **Saving the Dataset**:
    a. The cleaned dataset was saved to a CSV file as `movies_dataset_complete.csv` and ready for analysis.
7. **Additional Cleaning for Profits Dataset:**
    a. Movies with 0 or nan values in their budget or revenue were removed.
    b. A profit category was created which is the movie's revenue - budget
    c. An additional cleaned dataset with the movies profits was saved to a CSV file called `movie_dataset_cleaned_with_profits.csv` .

Dataset Results

After gathering, cleaning, and integrating the data from IMDb and TMDb, we managed to reduce the data down to 156K rows and 8295 rows to our complete and profits dataset respectively. The complete code used for this process is provided in the `fetch_movies.py` file, which demonstrates all the steps in detail.

Factors Affecting Movie's Profits

Overview

There are many different factors that influence a movie's profits. We looked at a few factors which are how movie's run time, rating, genre and release year affected their profits.

Data Processing

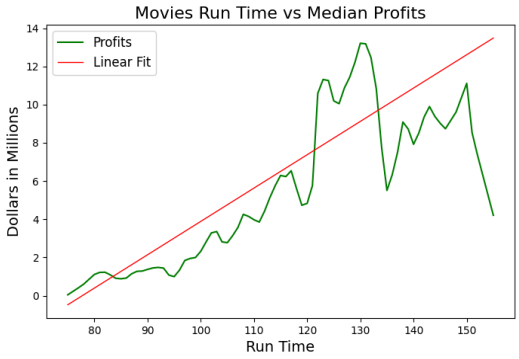
We grouped our data according to their runtime, rating, genre and release year and took the median profits for each. Below is an example of what the command looked like:

```
grouped = temp.groupby('runtimeMinutes')['profit'].median().reset_index()
grouped = temp.groupby('runtimeMinutes').agg(
    median_profit = pd.NamedAgg(column='profit', aggfunc='median'),
    entries = pd.NamedAgg(column='profit', aggfunc='count'),
).reset_index()
```

Results and Findings

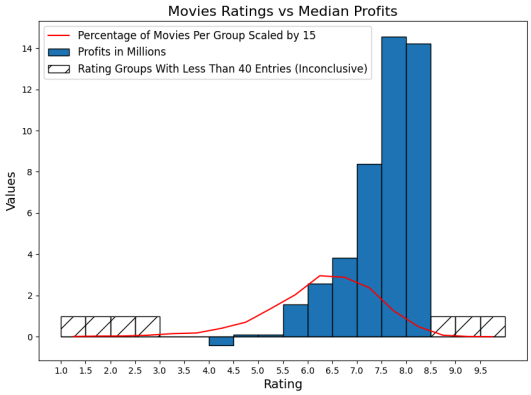
Run Time vs Median Profits

The profitability of movies seemed to have a positive correlation with run time with a correlation coefficient r of 0.66. The most profitable movies seem to be those between 121 and 150 minutes which is close to double the median profit of all movies at $3.4 million at 105 minutes.



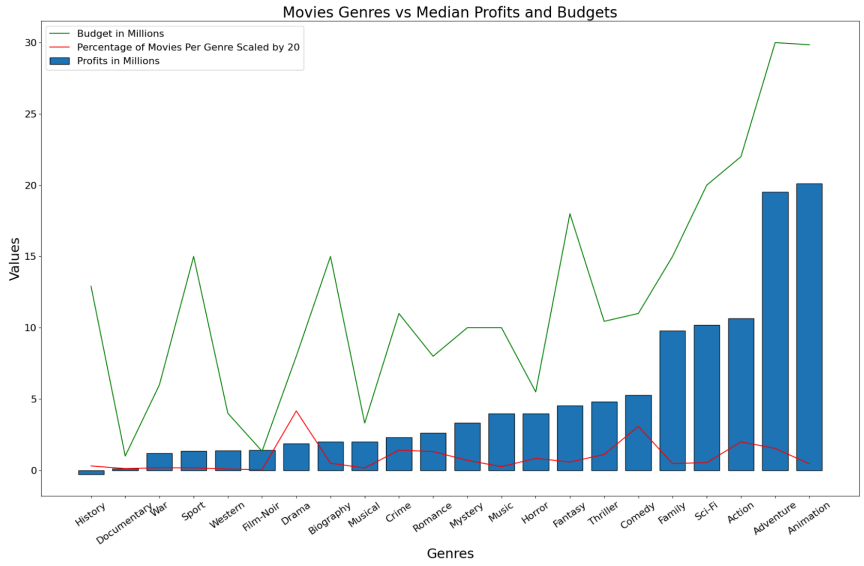
Ratings vs Profits

The profits of movies have a positive correlation to ratings which gets stronger once you pass a rating of 5.5 stars. The mean rating of all movies is 6.37 and if we take the + or - standard deviation σ from this we get that approximately 68% of movies fall between the ratings 5.29 and 7.47.



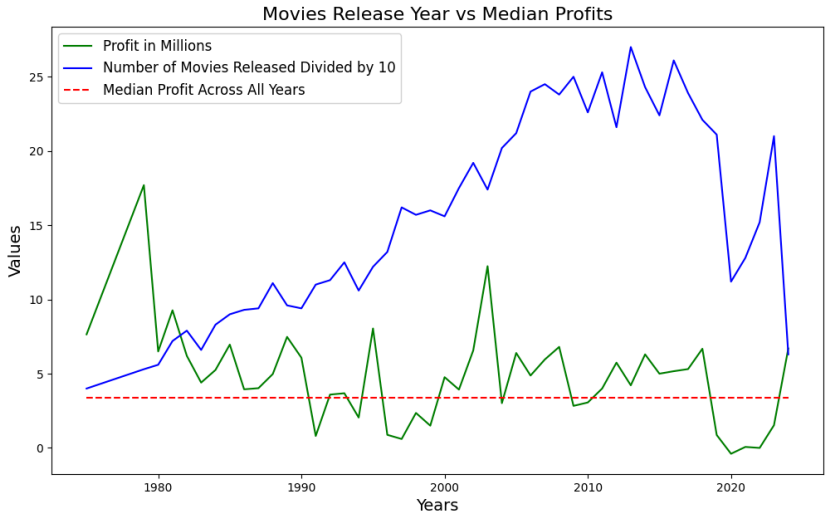
Genres vs Profits and Budgets

Adventure and Animation movies are the most profitable genres, however they also require the largest budget. Genres that have budgets with high peaks with relatively the same profits as its neighbors should be avoided to reduce risk. Some noteworthy genres that should be avoided are Fantasy, Biography, Sports and History.



Release Year

The median profits of movies have a large amount of variance year to year. We can use the median profit over all of the years of $3.4 million as a nice base-line to compare each year to. Covid had a tremendously negative effect on the movie industry which is evident by how the most recent years have fallen far below base-line. Noteworthy is the fact that the only years with negative median profits are 2020 and 2022. The data does show an upward trend however as 2023 had a positive median value of $1.5 million and 2024 is currently at $6.7 million which is well above base-line. Given the current trajectory we think it is a good time to make movies.



Limitations and Challenges

After grouping the data we found some groups did not have many entries and including them would have skewed our findings. For the runtime group we omitted those with less than 20 entries and imputed 5 data points. Noteworthy of which is 153 and 154 which contributed to the graph's downward trend at the end which seems counterintuitive.

For the ratings, genre and release year groups we omitted those with less than 40 entries. We increased our threshold to 40 as these were larger groupings and seemed like a reasonable number given our dataset's size.

We would have liked to have more data in some of our groups as it would be nice to see what the profits are on the lower and upper bounds of the ratings graph.

Predicting Movie Revenue

Overview

We used Machine Learning to predict the revenue of movies using Random Forest Regressor. We used a dataset that consists of features such as the year of release, runtime, genres, avg rating, number of votes and budget. The goal was to create a predictive model that can estimate the revenue base on the features mentioned.

Data Processing

To ensure that our dataset was clean and suitable for modeling, we converted non-numeric values to numeric values in the `runtimeMinutes` column and used `coerced` as a function argument to set invalid data to `NaN`, then dropped rows with any missing values.

```
# Convert runtimeMinutes to numeric, coerce errors to handle non-numeric values
movie_data['runtimeMinutes'] = pd.to_numeric(movie_data['runtimeMinutes'], errors='coerce')

# Drop rows with missing values
movie_data = movie_data.dropna()
```

Some of the data in the `genres` column contained multiple genres for each movie, separated by commas. Encoding was used to transform the categorical data into multiple binary columns, one for each genre. This technique allowed us to include genres into our regression model.

We used `MinMaxScalar` from scikit-learn to ensure that all features were on standard scale and so we normalized all numerical features.

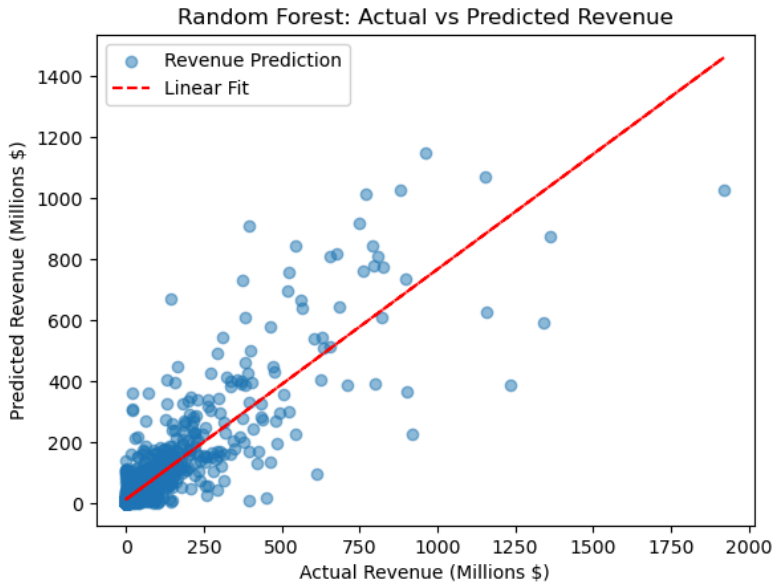
Most of the movie revenues that we encountered were in the high millions or over billions, we divided all the revenue by 1 million to scale it for better readability of the data.

Our dataset included movies from well before the year 2000, many of which had incomplete data in key columns such as `runtimeMinutes`, `budget`, and `revenue`, often recorded as zeros. To prevent these inaccuracies from introducing sensitivity to outliers and skewing the predictions, we excluded these rows from the dataset.

Results and Findings

Actual VS Predicted Revenue

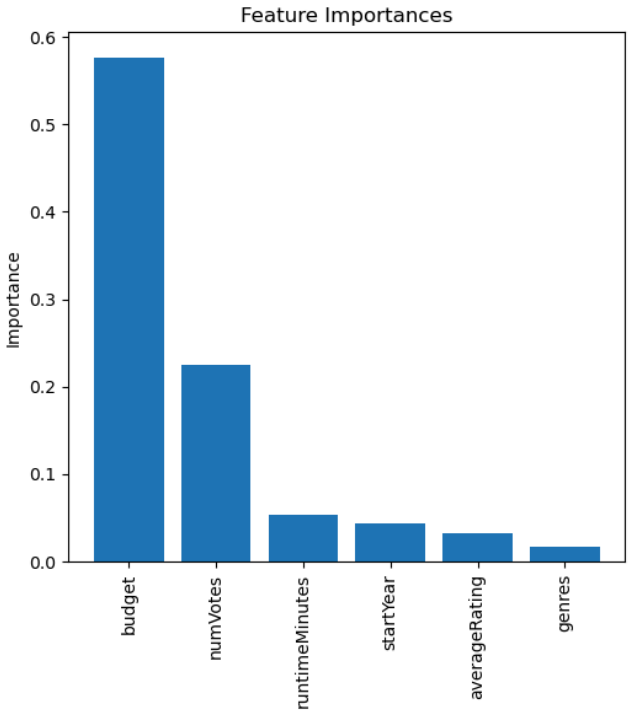
The model has a score of 96.6% accuracy on the training data which is an indication of a good learning of the data. Analyzing the score on the test data, we get a score of approximately 74.2% on unseen data, which indicates that the model shows somewhat a strong performance in predicting movie revenues. The accuracy is lower on the test data than the training data which is an indication of overfitting.



The Random Forest model shows promising results in learning from training data by achieving a high accuracy. However, when evaluating test data, the model achieves a much lower accuracy indicating a reduction in performance on unseen data. While there is a trend with predictions aligning with actual values, there are instances where the prediction departs significantly from the actual revenue, specially in the higher revenue. This concludes that there is room for improvement.

Feature Importance

The features importance was determined using the `feature_importance_` attribute of the Random Forest Regressor. Features with importance less than zero were excluded to simplify the analysis.



This graph presents an analysis of the feature importance derived from the Random Forest model. As we can see, the budget of a movie is the most critical factor influencing the revenue predictions with a ~0.60 importance. The number of votes is also a significant predictor of its revenue. Thereafter we have the runtime of the movie, year released, average rating, and genres.

Limitations and Challenges

While we had basic features such as year of release, runtime, average rating, number of votes, budget, and genres, there are many other factors that influence the revenue of a movie. As we saw in the feature importance graph, the budget is the most influential factor in determining the movie revenue, thus, other features such as marketing money spent could possibly help with more accurate predictions. Other features such as directors, cast and production company could definitely improve the models predictions.

Due to time and API constraints, we were not able to gather more features as mentioned above for our datasets. Some other challenges were tuning the RandomForestRegressor parameters to find the appropriate prediction score.

Project Experience Summary

Ilia Fatemi

- Researched and found open source datasets to use for project (IMDb datasets)
- Integrated TMDb APIs with the IMDb dataset to gather additional fields for each movie.
- Cleaned dataset after gathering the data from the API
- Created a pipeline model that includes Random Forest Regressor and transforms each feature using the Min Max Scalar to normalize the data.
- Analysis on Actual Vs Predicted Revenue to see how the model did on predictions.
- Analysis on Feature Importance to find features with significant impacts on prediction

Michael Fischer

- Grouped quantitative data taking their median profit and number of entries per group
- Grouped categorical data that contained multiple categories per entry.
- Cleaned groups by removing those that had less than a certain threshold of entries.
- Imputed data by using a k-nearest neighbor classifier.
- Created Graphs on factors that affect a movie's profits.
- Did analysis by taking the mean, median, standard deviation, and correlation coefficient of variables.