CMPT 353 Project Report - Movies Analysis

Sherman Chao Wen Chow - 301232684
Ahsan Naveed - 301228556

Introduction

There are three parts to our project, which are comprised of determining the correlation between success criteria, predicting movie genres from plot summaries, and determining the success of a movie from selected features such as genres, country of origin, publication date, cast members, etc.

1 Correlation Between Success Criteria

1.1 Problem

Our initial question was to determine if the various criteria for success, such as critic reviews, audience reviews, and profit/loss correlate with each other.

1.2 Data

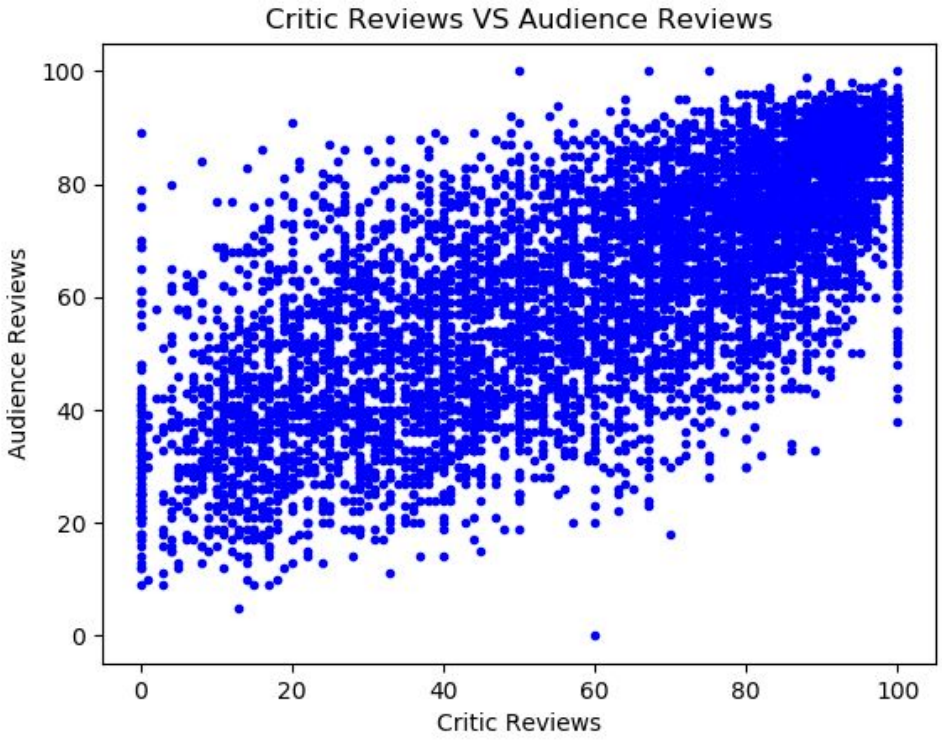
The data that were mainly used was extracted from the Rotten Tomatoes and WikiData data sets. We read these data sets into DataFrames.

After joining the data sets, we made use of only the necessary columns that represented the critic percentage of positive reviews, the audience percentage of who liked the movie, and whether the movie made profit or not. We then separated the data into three DataFrames, each with two columns. These DataFrames were organized into tables that compared ‘critic reviews vs audience reviews’, ‘critic reviews vs profit’, and ‘audience reviews vs profit’. We also removed rows that had empty or invalid values.

1.3 Analysis

The techniques used to analyse the data included the use linregress to compute the correlation coefficient between the following success criteria: ‘critic reviews and audience reviews’, ‘critic reviews and profit’, and ‘audience reviews and profit’. We also plotted a scatterplot of the above data to see if we can observe any obvious relations.

```
Correlation coefficient between critic reviews and audience reviews: 0.7238120931512411
Correlation coefficient between critic reviews and profit/loss: 0.2178798104231512 7
Correlation coefficient between audience reviews and profit/loss: 0.2597119876614688
```



By looking at the correlation coefficients and plotting the data, we can observe that there was a strong correlation between the critic reviews and audience reviews. However, the other criteria we selected for success such as critic reviews, audience reviews, and profit/loss did not correlate as much. This was rather surprising as our initial expectation was for all the success criteria to have strong correlations.

1.3 Limitations

If we had more time, we would have used the data regarding awards from the OMDB data set. Since this data included descriptions of the number of nominations, types of nominations, and awards won, we could have applied regular expressions to extract and categorize the types of awards. After extracting the data, we would have used it as a more detailed criteria for success containing the different types of awards to find some correlation with the other success criteria.

2 Movie Genre Prediction

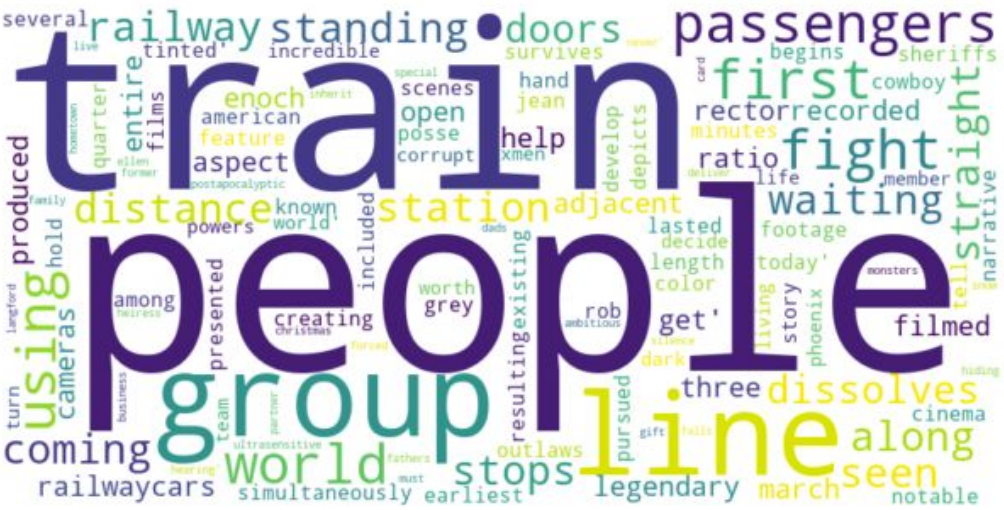
2.1 Problem

Our problem here was to attempt to predict movie genres from plot summaries.

2.2 Data

The data used was from WikiData, OMDB, and Genres. We read these data sets into DataFrames, and used the Genres data set to map the genre codes from OMDB to their corresponding English genre labels.

We first merged the WikiData with the corresponding OMDB plot summaries by their IMDB ID, and converted all the text to lowercase. We cleaned the data by removing rows with no plot summaries, removing all punctuation, and kept only valid words and numerical characters in the text. We then tokenized the plot summaries into individual words and removed NLTK stop words. After obtaining the cleaned plot summaries, we generated a word cloud image of the most common words as shown in the visualization below.



2.3 Multi-Label Classification

Since each movie may have multiple genres, this problem is a multi-label classification using natural language processing. Therefore, we used the tf-idf as calculated by TfidfVectorizer to pick out the important words that appear often in a particular plot, but infrequently in the data set overall. We converted the plot summaries to features by using MultiLabelBinarizer in order to encode genres in a binary label representation. After splitting our data into training and validation sets, we built a Logistic Regression model, and trained it. Since some movies had multiple labels, we also used OneVsRestClassifier for this Binary Relevance problem to train each genre. Afterwards, we made predictions on the validation data with a threshold value of

0.25, and obtained an F1 score of 0.34. This F1 score is rather decent. Observing the image below, showing some of the predicted genres compared to their actual genres, we can conclude that this classification does a decent job at predicting movie genres from their plot summaries.

```
f1 score: 0.34161419576416713

IMDB ID: tt0172684
        Real genre(s): ['drama film', 'romance film', 'coming-of-age story', 'comedy-drama']
        Predicted genre(s): ['drama film']

IMDB ID: tt1007950
        Real genre(s): ['comedy-drama', 'comedy film']
        Predicted genre(s): ['drama film', 'comedy film']

IMDB ID: tt0335438
        Real genre(s): ['buddy film', 'comedy film', 'buddy cop film', 'action film']
        Predicted genre(s): ['drama film', 'comedy film', 'action film', 'crime film']

IMDB ID: tt0405422
        Real genre(s): ['romantic comedy', 'buddy film', 'sex comedy']
        Predicted genre(s): ['drama film', 'comedy film']

IMDB ID: tt0082425
        Real genre(s): ['teen film', 'comedy horror']
        Predicted genre(s): ['drama film']

IMDB ID: tt1333125
        Real genre(s): ['comedy film']
        Predicted genre(s): ['drama film', 'romantic comedy']

IMDB ID: tt2126235
        Real genre(s): ['action film', 'crime film', 'drama film']
        Predicted genre(s): ['action film']

IMDB ID: tt0290334
        Real genre(s): ['action film', 'superhero film', 'speculative fiction film', 'science fiction film']
        Predicted genre(s): ['action film', 'science fiction film']

IMDB ID: tt0115006
        Real genre(s): ['horror film', 'science fiction film', 'monster film']
        Predicted genre(s): ['horror film', 'adventure film', 'science fiction film']

IMDB ID: tt0033152
        Real genre(s): ['fantasy film', 'flashback film', 'family film', 'fairy tale']
        Predicted genre(s): ['romance film', 'drama film', 'fantasy film']
```

2.3 Limitations

If we had more time, we would have tested our problem with different classifiers such as SVM and Naive Bayes to try to obtain a higher F1 score and better overall predictions. We could have also incorporated the publication dates of the movies to observe the trend of words and popular genres throughout the years.

3 Movie Success Prediction

3.1 Problem

Can you predict review scores from other data we have about the movie? Maybe genre determines a lot about the success of a movie? Or maybe the actors?

3.2 Data

The data used was from WikiData and Rotten Tomatoes datasets. We merged these two datasets on 'rotten_tomatoes_id'. After joining these datasets we only extracted the necessary columns for training the model. For 'X_columns' we selected all the columns we got after one-hot-encoding the 'genre' column and for 'y_column' we chose 'audience_percent' as an indicator of a movie's success, because it is the percentage of people who liked the movie. Later we normalized 'audience_percent' to values between '0' to '10' so that our model will not treat values like '70.5' and '70.6' differently and instead use '7' as a representative of both percentages. So, model has a finite amount of values to converge to. We also made sure none of the ratings are abnormal i.e. greater than 10. Overall, this helped reduce our training time. We also dropped the 'null' values from our final dataset before training our model.

3.2 Analysis

3.2.1 Based On Genres

During our analysis for this task we experimented with different models. The task at hand was a regression task, since we were attempting to predict a numerical quantity so we ended up selecting 'MLP Regressor' from 'sklearn' library. We used 'train_test_split' helper function to split our dataset into training and test data. We then use this training data to train our model. We experimented with different numbers of 'hidden_layers' in our model and also with different numbers of neurons per layer. We got the best result with two layers and 357 neurons i.e. model scored 0.229 with those parameters. We further consolidated the configurations of our model by checking online articles [1] and most of them recommended used 2 hidden_layers and each layer with number of neurons equal to the number of features.

3.2.1 Based On Cast

We wanted to see if the 'cast_member' could tell us something more about the success of the movies but after training the model with settings mentioned above we got a lower score compared to what our model spit out after being trained on genre based data. We also considered other features in our dataset in hope to get better scores for our model e.g. 'country of origin', 'publication date' e.t.c. We were not able to cross the genre based

score. We hypothesized that genres were the strongest candidate for the movie's success prediction. Which makes sense and has been confirmed by our research [2].

3.2.2 Limitations

If we had more time we would have tried to combine genres with other features in our dataset and probably used 'GridSearch' for better hyperparameter tuning instead of doing it manually. We also wanted to predict the most popular genre given a year and other features such as 'country_of_origin'. We think this would have been beneficial for our overall analysis.

Project Experience Summary

Sherman Chow

- Extracted and processed movie data to analyse correlations between success criteria
- Cleaned movie plot summaries and extracted features using tf-idf
- Performed multi-label classification in NLP to predict movie genres from plot summaries
- Generated visualizations representing correlation results using a scatter plot and a wordcloud to display word frequencies

Ahsan Naveed

- Wrangled movie dataset to get desired dataframe to feed to model
- Achieved shorter training times on different models by smart feature engineering
- Built strong understanding of handling categorical data with different techniques e.g. 'one-hot-encoding', 'binary-labelizer', 'multi-labelizer' e.t.c.
- Learnt hyperparameter tuning for different ML models e.g. MLP Regressor, LogisticRegression e.t.c.

References:

[1] Categorical Data
[2] Business Insider: Top Movie Genres
[3] Working With Text Data
[4] Multi-Label Classification