LetterBoxd Film Analysis

Genre/Theme/Production

Clayton Brittan
CSPB-4502
CU Bouler
clbr8512@colorado.edu

Problem Statement:

The film industry is constantly evolving and with it the viewing public's tastes. Because of this it is becoming more important to understand these trends both as filmmakers and viewers. By understanding the underlying factors that influence a film's success we can determine if any particular factor plays a greater role in determining a film's success. This project aims to do just that, by studying the data available from LetterBoxd, a crowd sourced film review site, we can make predictions of certain films.

This project aims to take in a wide variety of features including actors, directors, studios, themes, genres, duration, and train a model on the data to predict a film's quality. After selecting our features and training our model we can use it to try to predict the quality of certain films.

The reasoning behind this project is to reveal the underlying roots that propel a certain film towards greatness. Perhaps a certain actor's performance carried the film, or perhaps the themes matched the genre, or maybe it was the perfect combination of funny and sad and not too long. If any of these trends are true it would be present in our model. This way we can see if a certain actor outperforms others, or if certain combinations of themes in a film are preferred given a certain genre. One question I want to study is how the viewing public adapts to film runtime, if it affects the film's score at all?

By creating such a modeling tool we can use it to not only predict a film's quality based on its features, but also find similar movies to ones you like. As well as determine which actors, producers and genres are most popular and successful.

Literature Survey:

This dataset has also been used by other data scientist to create a mapping of which genres are most popular by country:

(https://www.kaggle.com/code/suntzunami/what-the-world-be-watchin).

This project simply took films based on producing countries and plotted them against viewing country data. This project did not look at the underlying trends within the films nor the actors.

Proposed Work:

Prior to starting the data analysis, the dataset is split into several csv files, each containing relevant information. To collect all these files and place them into one usable environment we will rely on a database or csv merging, whichever proves to be faster and more efficient. Luckily, each file includes the film name so we will be able to merge the CSVs based on the film's name. Any films that are missing entries in this merger will be removed due to incomplete information. This will save us time in the data processing step as we won't have to worry about null or zero values. Once all of our data is collected into a single source we can begin dividing it into smaller more manageable subsections. Initially I would like to divide the films based on country or country of production, as this will separate most films into easily identifiable categories. Then these groupings can be further split into genres, themes, producers, studios, whichever is needed to analyze the data. Once the data has been sectioned we can begin analyzing the correlations between the features and look to select the highest correlated features. By optimizing our feature selection we can ensure the accuracy and efficiency of our model. After selecting the appropriate features we can build our model on those features with our target variable being the movie's score. The film's score is rated 1-10 and is an average score of thousands of user votes. Once we have built our model we can use it to predict film scores. We can use a training set of data to train the model and determine its accuracy. Once the model has been built we can optimize the model's hyperparameters to try to increase the accuracy of the model as much as we can. This involves changing the individual parameters of the model and rebuilding several times to find which ones produce the best results. This data will also need to be stored for graphing purposes in the future. With our optimized model we can then use it to predict how we think a film will do.

Data set:

https://www.kaggle.com/datasets/gsimonx37/letterboxd/data

The dataset includes a substantial amount of information including movie posters for all 900k+ films. The total file size is 24GB, however since most of that is from movie posters we will not be using those. The dataset is further divided into CSV files and for our purposes we will be mainly using the movies, themes and genres csv files.

The dataset contains entries for every film posted on LetterBoxd and their accumulated score out of 5, as well as all relevant actor, director, producer, genre, release, language, country of origin, etc information. The dataset is split into multiple CSV files that each contain unique features and the movie title. The dataset itself comprises 1.7 million movie entries including films from every country in the world. Each CSV file contains scope centric information for example; geographical location of production, production crew, studio is contained in one CSV while Genre, Theme, and Setting information is contained in another. This means that the full dataset is spread over multiple files and to access all the information at once requires merging.

Evaluation Methods:

Our primary forms of validations are MSE and RMSE, as well as cross-validation. The MSE and RMSE are great measurements for determining the accuracy of predicted models. By subtracting the difference between what we expect vs our actual data we can get a numerical measure of our inaccuracy and try to minimize that measure. The RMSE provides a normalized version of the measurements as RMSE instead measures the standard deviation difference between the expected and actual. By training the data on a partitioned version of the dataset we can utilize cross validation to analyze our predictive performance. CV (cross validation) is sometimes a better evaluator as it can prove how good a model is at predicting new data. As that is one of the intended uses of the model it is valuable to utilize CV in our evaluation. Based on the efficacy of both of these methods, by implementing them both we can ensure our model is robust and accurate.

Tools:

The primary tool we will be using is Python, as it is the driving programming language behind the project, it is also very versatile and has great data analyzing and modeling capabilities. We will also be utilizing python libraries such as Pandas which is a very prominent data manipulation and analysis tool in Python; as well as NumPy for our mathematical calculations. Sciki-Learn includes many of the statistical models that we will be using so it is an essential tool for this project, as without it we don't have a model. We will also be using a variety of plotting tools such as MatplotLib, Seaborn, and PyPlot. Each excel in certain aspects and fall short in others so they will be used when necessary. SQL databases could be used depending on how the data processing step goes and what is required. The whole project is hosted using Git and we will be using it for project tracking and branch management. Finally we will be using Jupyter notebooks as a presentation tool to store a lot of the visualizations and modeling. By utilizing all these tools we can efficiently perform all the tasks laid out and ideally create a predictive model that is highly efficient and accurate. However with these tools, we can also allow for reproducibility, scalability and all around success.

Milestones:

The primary milestones for this project are:

- Data Wrangling - 3/20/2024
- Data Processing - 3/24/2024
- Feature Selection - 3/29/2024
- Model Building - 4/2/2024
- Hyperparameter Optimization - 4/7/2024
- Analyze Model Data - 4/14/2024
- Build Report & Presentation - 4/20/2024
- Progress Report - 4/22/2024
- Rebuild Model on Full Data - 4/26/2024
- Final Report - 5/2/2024
- Presentation - 5/2/2024
- Evaluation - 5/2/2024