Utilizing the Open Movie Database API and Netflix Data for predicting rating of a movie

Amit Mandliya
North Carolina State University
Raleigh, US
amandli@ncsu.edu

Jay Modi
North Carolina State University
Raleigh, US
jmodi3@ncsu.edu

Jaydip Gabani
North Carolina State University
Raleigh, US
jgabani@ncsu.edu

Maharshi Parekh
North Carolina State University
Raleigh, US
mgparekh@ncsu.edu

ABSTRACT

Prediction systems can be utilized to plan and make changes in the features that might lead to a better product. This paper describes the prediction mechanism that we have developed in the field of movies and TV shows. For the data model, the data-set was obtained from The Open Movie Database and Netflix Prize database. The paper also discusses a baseline model that could be applied to the data-set and compares it with the final model described in the paper.

KEYWORDS

Ensemble methods, movie rating prediction, OMDB, Regression, Stacked Generalization

1 INTRODUCTION

Movie ratings prediction system can be used to create movies that most appeals to the user. It can be used to cast the crew, the combination of whom has a greater impact on a movie being liked among the users. The rating prediction system can also be used by the distributors and sellers to identify the sales for a particular movie. The early prediction of movie also needs to have proper weights associated with various features. All of these reasons are the motivating factors behind this project.

We have come up with a model which can predict the average rating a movie will get based on the people who have worked on the movie and the length of a movie etc.

2 BACKGROUND

The project idea is to predict the likeliness of a movie among users based on the previous available data. The data-set is obtained from Open Movie Database (OMDB). OMDB provides the data in the form of JSON format per movie. We have obtained the movie titles and year from Netflix prize data-set. We have then used these titles and years to fetch the data from the API provided by OMDB. The model is trying to predict the movie ratings by finding doing a regression on the dataset with multiple underlying models and combining them as Stacked Generalization method. [5]. We would also be able to find some interesting patterns like most liked movie genre wise, best director-actor pairs, etc. To evaluate the models, we have used Mean Squared Error, Root Mean Squared Error, Absolute Error and R^2 score. [2]

We have gone through some previous years' research papers that have helped us in shaping our thought process for this problem and building the data model around it:

- A paper[3] discussing various techniques and some existing interesting relationships in typical movie rating datasets. It talks in detail about the content-based, collaborative based, and hybrid methods for rating prediction and compares them. We have used the KNN model as our baseline model to predict the use rating. This model is also described in the mentioned paper. The paper discusses how the KNN algorithm is useful in predicting the ratings of a movie and uses cosine similarity among users.
- This paper[4] obtains the data from three sources IMDB, Wikipedia, and Rotten Tomatoes and combines them. It uses different models namely - Linear Regression Model, Support Vector Machine Regression Model, and Logistic regression. It takes various nominal attributes like Actors, Director, Writer, Production-House, Genre, and Numerical attributes like Budget, IMDB Rating, No of Rating, IMDB Votes, Metascore, Tomato Meter, Tomato User Rating, Tomato Reviews, Tomato Fresh, Tomato Rotten into consideration. We have implemented the Support Vector Machine Regression Model by selecting a few of these attributes and combining them with additional features like Director, Rated, etc which were available in our data set.
- An eye opening paper[1] on visualization trends and challenges provided critical insights into the myths, dos, and don'ts of visualization and various important aspects that make a visual more interpretable. Major takeaways include - not everything needs to be visualized, a simple visualization might easily highlight the issues with our data-set, data visualization is not the substitute for data analysis and reasoning thinking. In our project, we have made use of Tableau Desktop to plot our data. Even making use of some newer plots such as tree-maps and circle packing we have found some interesting relations between Genres, Ratings, Actors, and Directors. Possible avenues of research based on attributes such as Writers and Metascores have also been discussed herewith. The paper's analysis of multiple dimension plots using Tableau was tested out thoroughly by trying out the Pearson Correlation plot and by even increasing granularity.

3 METHODS

3.1 Approach

We are using stacked generalization ensemble method. Here is some brief explanation regarding the estimators that we have used, and how is it related to the movies data-set that we are using:

- **Ridge Regressor**: Ridge regression is a way to create a model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity. In our data-set, after converting categorical data to numerical data we had around 15000 columns, which serves as a motivation to use Ridge Regressor.
- **Support Vector Regressor**: The Support Vector Regressor works with notion of creating a hyper-plane that separated the data into classes. It maps a decision boundary which makes the distinction between two or more classes, which is then used for prediction.
- **Random Forest Regressor:** A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the data-set and uses averaging to improve the predictive accuracy and control over-fitting.
- **Stacked Generalization Method:** We used the Ensemble methods which combine the best performing models from the above-mentioned models and ensemble them. Provided by scikit-learn (sklearn) package in python, Stacked generalization is an ensemble technique that blends the prediction from two or more sub-models.
  The movie database has categorical attributes like Actors, Directors, Writers, Production, Content Rating, Genre, and Language. After converting these categorical attributes to vectors for analysis purposes, we applied multiple regression techniques (Lasso, Ridge, Supporting Vector, Random Forest) and the best individually performing regressions were Ridge Regressor, Support Vector Regressor, and Random Forest Regressor. These techniques were the best among all the Regressor but to improve results we decided to combine these techniques with stacked generalizations to get even better results.

We have compared the results of the model that we create with the baseline model - KNN algorithm.
Our approach fulfills the following novelty criteria:

- **Collect a novel dataset** - The dataset of our interest is not readily available and is created using OMDB API and Netflix Prize Dataset as mentioned earlier.
- **Using a dataset to explore a meaningful hypothesis** - We have come up with interesting relationships like yearly trends in the movie ratings, what values of factors lead a movie to be rated better than others.
- **Implementing a novel/complex machine learning technique** - We will be using the Stack generalization method for our data models which is an ensembled method that best combines multiple models.

3.2 Rationale

We are dealing with the prediction problem here, where our goal is to predict the average rating a movie will receive based on the features it has, for example, Genre, Country, Actor, Director, etc. To do this we implemented various models namely - KNN, Lasso Regression, Ridge Regression, Ridge Regression, Support Vector Regressor, Random Forest Regressor. These will serve as estimator models, and some of the better performing models will be combined in our final model as an ensemble model. The motivation behind using multiple models is to see if we can improve the accuracy of a data model by blending predictions provided by multiple models together. Linear Regression model is not useful here because the data-set is complex with many important attributes, and a simple linear regression method might overfit the data since it does not penalize for its choice of weights. We also tried using it just to make sure that this is right and we found that the mean squared error was tremendously high.

We have found that ridge, supporting vector regressor and random forest regressor were individually proving to be a good model for the provided data-set. Setting these models as estimators for the ensemble model will be a good choice.
There is an ensemble regressor provided by sklearn, Voting Regressor, which also combines multiple sub-models, however, it averages out the predictions provided by all the models without considering the bias factor, this is only useful when all the estimators are equally well-performing. These reasons serve as motivation for us to use stacked generalization [5] in our problem since this model best combines the estimators rather than averaging them out.

We practiced data visualization to decide on some good factors for our method implementation, which are discussed below:

- To understand the concept of how a movie rating might be affected by geography figures 1 and 2 were plotted. This helps us grasp that there are a lot of movies being released in numerous countries where the average rating of a movie differs from country to country showing that there might be interesting connections between data points and release-region. Though this is a rough overview of the data-set, this does provide an inkling as to a region-specific audience which might or might not judge your movie with bias.

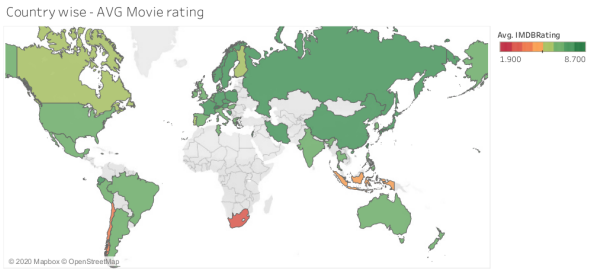

Figure 1: Average Movie Rating across the Countries

- Further interesting insights were obtained when analyzing the relationship between actors and directors across all the movies:

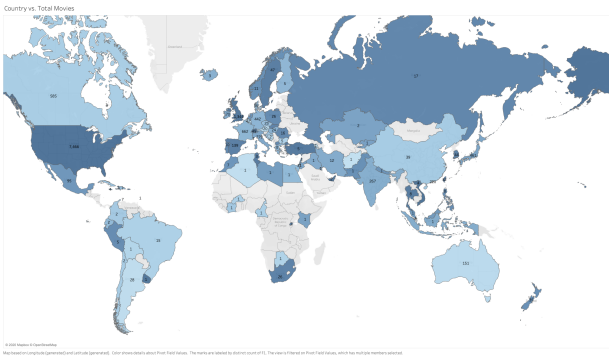

Figure 2: Number of Movies released per country

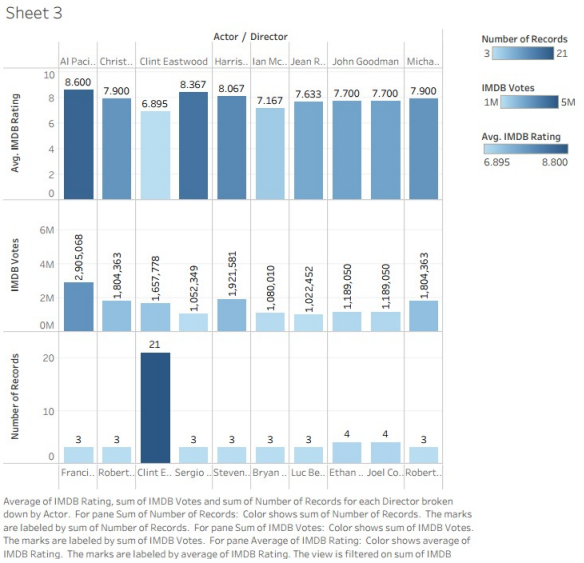

Figure 3: Number of Votes and Ratings for Actors and Directors

Figure 3 shows the pairs of Actors and Directors who've worked at least 3 times and whose movies have garnered at least 1M votes. Figure 4 shows the binned average IMDB rating of a movie with the number of votes accrued. In figure 4, we can see that movies with a higher number of votes have an average rating between 5.8 and 7.4. But for the actor-director pairs in figure 3, the ratings are very high (around 8) despite a high number of votes.

- Connecting the dots, we see that the actor-director pair who've worked together before and have a good history of pleasing the masses tend to have a good blockbuster movie

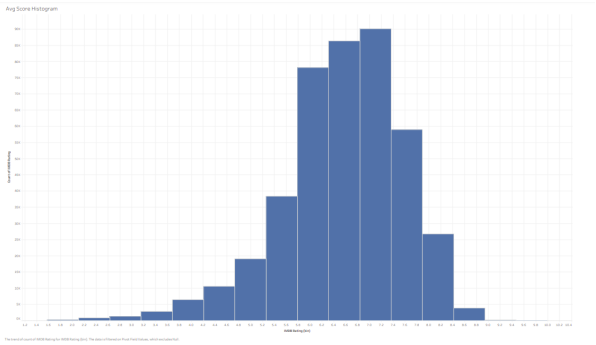

Figure 4: Average IMDB Score Histogram

whether they get a median number of votes or not. Next, we take a look at how different genres of a movie might affect its rating.

- *[Refer to Figure 11 in Appendix for 2-Genre Treemap image.]* Generally, a movie will be associated with 2 or more genres at its launch and it's really interesting to see from the figure that while people appreciate the combination of Drama and Film-Noir the most, they loathe the combination of Music, Musical movies. On an average, people seem to rate movies of Drama and Adventure genres on a higher note than the rest.

4 PLAN AND EXPERIMENT

4.1 Description of the data-set and fetching of data

We have utilized the data-set provided by publicly available Netflix Prize movie data-set. It contains a file with titles of movies along with the year of release. We have then used this data to extract the details for each movie from the Open Movie Database (OMDB). The OMDB data-set provides the details of a movie in JSON format, some of the details with respect to each movie that we were able to fetch are - Director, Genre, Country, Actors, Production, Rated, Runtime, IMDBRating, Box Office, IMDBVotes. We are using the JSON package in python to convert the data and combine all the movie data to CSV format. Overall, our data-set contains around 22,000 rows.

4.2 Data preprocessing

There were attributes like Genre, Director, Actor, Writer - all these had comma-separated values. We have converted the categorical values into numerical values by adding a column for each possible director, actor, writer, the genre of a movie, and setting the value to 0 and 1, where 1 means that this is the right director or any other crew of the movie.

We noticed that there were many duplicate entries for particular movies and we have removed those by filtering out with only one row for each movie title and year. After removing the duplicates, we have around 8k rows in the data-set.

We have removed the rows with missing values of the attributes

that were included in the prediction process.
We have vectorized the categorical data using numpy and sklearn packages and then scaled the whole dataset for the model training.

4.3 Hypotheses

The first and the major hypothesis that we have tried to answer can be stated as what will be the average IMDB rating of a movie given its attributes such as director(s), writer(s), actor(s), genre(s), country, runtime, production house, and language.

Another interesting hypothesis that we tried to investigate is whether there exists any correlation between actors and directors on a movie's ratings.

4.4 Experimental Design

(1) **Prediction algorithms used to evaluate the first hypothesis:**
We have applied below data models to identify which of them predicts the movie ratings with more accuracy. We have taken attributes - Director, Genre, Country, Actors, Production, Rated. We have divided our data-set into testing and training samples to validate by applying 5-fold Cross-Validation. The results for these models are described in the Results section.

- **KNN Algorithm**: We have applied 3-NN, an algorithm to find three nearest similar movies for each movie in the test row. The prediction is made by averaging the ratings of the closest found neighbor.
- **Ridge Regression:** We have used the Ridge method provided by sklearn package in python. The alpha value used for ridge regression is 1 which is the value chosen for ridge regression. We have tried multiple lambda values for this regression, to find the optimum value for this hyperparameter: [0.001, 0.01, 0.05, 0.1, 1, 10]. The scoring technique used was a negative mean squared error.
- **Support Vector Regressor:** We have used SVM class of sklearn for this purpose. The SVR method provided by this class is used to fit the model. The input parameters of this method are smoothness factor, epsilon and regularization parameter, C. We have tried various combinations of these hyperparameters:C takes the values as [0.001,0.1,1,10,15], and epsilon as [0.001,0.01,0.1,0.5,1,10]. We have a combination of each of these values.
- **Random Forest Regressor:** We have used the RandomForestRegressor method provided by sklearn.ensemble package to implement RandomForestRegressor. The input parameters for this regressor are the number of trees, maximum features per tree, and maximum depth. We have used numTrees = [1, 5, 10, 15, 20], maxFeatures = [0.25, 0.5, 0.75, 1] and, maxDepth = [3, 6, 8, 10, 15, 25]. We have found the best combination of these parameters by finding the least CV error among these.
- **Stacked Generalization Method:** We have used StackingRegressor method provided by sklearn.ensemble package to implement the Stacked Generalization Method. The estimators used for the method were RandomForestRegressor with parameters max_depth=10, random_state=0, n_estimators=15, max_features=0.5 - SVR with parameters C = 10, epsilon = 0.5 - Ridge with parameters alpha=0.01. The data-set was split in 80% as test data and 20% as test data. Then we used the fit method of the same package to train our model. After training our model we tested it against our formally decided evaluation matrices.
To test the individual models we are using Cross-Validation of 5 fold to identify the errors.

Here is a figure that explains the model that we are using as final model:

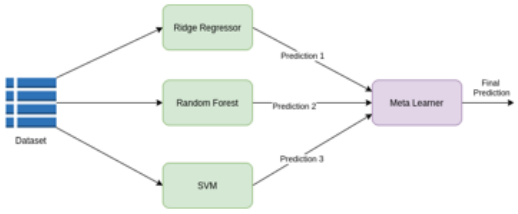

Figure 5: Stacked Generalization

(2) **Evaluation of the second hypothesis:**
To find out how a movie's rating will change over time with an increase in the number of votes, we can use the analysis explained in the 'Rationale' section. As mentioned earlier, figure 4 suggests that movies rating will be around 6-7 as the number of votes increase. But, figure 3 suggests, the rating might stay higher even with the increase in votes if the movie has some famous actor-director duo.
To test our hypothesis, after ensemble methods provide us with output ratings of a test movie having an actor-director pair, we can check whether the predicted rating corresponds to the trends in actors and directors.

5 RESULTS

5.1 Results obtained from the test

The below table includes the best hyperparameter values that we calculated with trying out different combinations of those values.

Technique	Least 5 fold CV Error	Hyperparameters
Ridge Regression	0.49	lambda = 0.01
Support Vector Machine	0.57	epsilon= 0.5, C= 10
Random Forest Regressor	0.38	Num of tree = 10, Max depth = 15, Features = 50%

Here is the visual representation of the accuracy for baseline model:

Figure 6: Accuracy of baseline model

Here is the visual representation of the of the accuracy for the model we used:

Figure 7: Accuracy of final model

We can clearly see from the above two figures that how better is the final model than baseline model. Here is a table that shows the comparison of the baseline model with our final model. We have improved to a good extent, as evident from the R2 score:

Evalution Metrics	KNN, k = 5	Stacked Generalization
Mean Squared Error	1.627	0.585
Root Mean Squared Error	1.275	0.765
Absolute Mean Error (Error Margin = 0.7)	0.524	0.366
Absolute Mean Error (Error Margin = 0.8)	0.462	0.307
Absolute Mean Error (Error Margin = 0.9)	0.409	0.25

R^2 [2] score can be derived with the formula:

$$R^2 = 1 - \frac{MSE(model)}{MSE(baselinemodel)} \quad (1)$$

Calculating the Coefficient of Determination, R^2 score for our models, it comes out to be 0.64.

Here is another plot that explains how the stacking regressor performs better than the individual regressing models.

Figure 8: Comparing individual model's performance with stacking regressor

Tabulated below are the results of Lasso Regression training (most useful when trying to find hidden associations when predicting values). Coupled with the inconclusive results from our Stacked Generalisation Model and less accuracy from the Lasso Regression model, our predictions have a weak confidence level to garner and support any useful association data and inference between Actors and Directors.

Allowed Error	Prediction Accuracy
0.10	10.58
0.30	24.87
0.50	40.21
0.70	55.56
0.90	67.72
Root Mean Squared Error	**0.918**

5.2 Discussion

Discussing the first hypothesis results: The results show the hyper-parameters that we set for our models. And also how did we combine it in the stacked generalization. As it is evident from the figure 8, that stacked generalization helped us in improving the model accuracy than the individual models that we tested out.

We also see from Figure 6 and 7 that our model is performing much better than the baseline model.

The evaluation metrics we used shows that the final model's RMSE value (which is 0.765) is much better than the baseline model's value (which is 1.275).

We can see in Figure 7, that our final model can predict the average ratings of a movie correctly 80 percent of the time with an error margin of 1. It is interesting to note that a movie can have certain

other features like the story-line, or the time of the release like a festive time, or what other movies are releasing at the same time, marketing strategy followed, etc., which might impact the ratings of a movie. These types of features are hard to identify before-hand and hence, could not be used in the dataset for prediction. Still identifying the rating with an error margin of 0.7 to 1 should be a good criterion. This can help distributors and sellers determine a movie likeliness beforehand and plan accordingly.

We were able to combine the models that were described in the prior work and come up to a better model as evident from Figure 8.

Discussing the second hypothesis results: testing ensemble results against the expected values proved to bring out inconclusive results while testing similarly with lasso regression models showed little association between actors and directors. Trying to train and test with limited attributes including only actors, directors, and languages also provided no results.

The reason being, we see this deviation from the expectation because of some hidden factors which were gone unnoticed while performing exploratory analysis on the dataset. Just as there are some hidden nodes in a Bayes Network, there would be some hidden factors that were to correlate to both Actors and Directors. Thus, even though we can say that there does not exist substantial proof of an association between the two, there might be an influencing third-party (maybe Genre or Production house or some unknown attribute such as the Movie Plot). Though disappointing to not find concrete evidence or the relation between those two, it is interesting to see that what is seen is not directly implied.

Futher analyses to find interesting relations between attributes:

- **Rate of change in the number of votes and average rating every 10 years:**
  As seen in figure 9, This time-series visual shows us that as people begin interacting more and begin voicing their opinions about their likes and dislike the movie ratings start averaging out over the years and reach a score around 6. As years progress, we can expect to see an average rating for a movie.

- **Metascores and Writer's Ratings:**
  Finding out the number of writers (Figure 10) in a particular score bin (1-10, size 1) showed that there were a lot of writers whose average movie rating was in 6-8 range, as expected from the average movie ratings trend in Figure 10. Next, plotting the movie Metascore versus the IMDB rating for each writer, we see that there is an interesting corollary between Metascores and ratings for writers, as shown below.

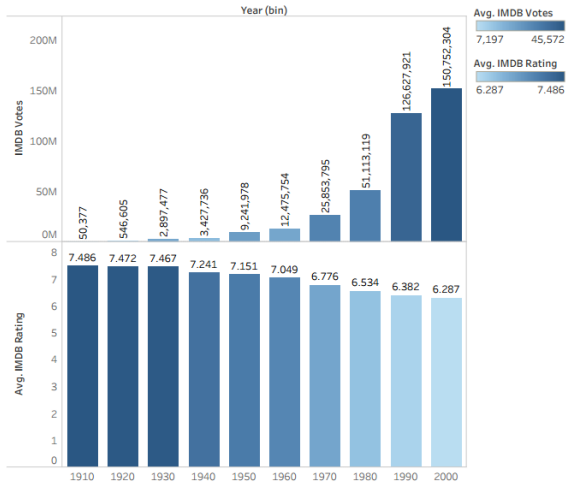

Figure 9: Total Votes and Average Rating Every 10 Years

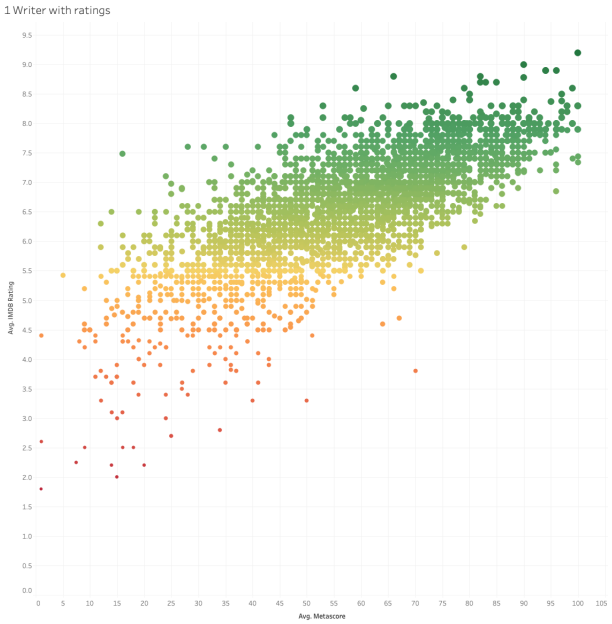

Figure 10: Writers plotted with Movie Metascore and Rating

6 CONCLUSION

There can be several approaches to movie rating prediction. We successfully assimilated the movies dataset and processed it to form clean and workable data. We have applied multiple algorithms for this project and combined the good performing ones with Stacked Generalization techniques. We can now successfully predict a Movie's Rating out of 10 with 80% accuracy. Our model is especially useful when you would like to predict the rating of a Movie when you have a lot of details about the movie.

The data-set was collected from publicly available Netflix Prize data-set and Open Movie Database. We found some interesting insights into the movies from the data-set. Analyzing various methods and attempting our hands at several different routes, this project has enabled us to learn the following:

(1) We see that a large dataset and higher dimensionality requires smart calculations else complex models will become very time-consuming .

(2) To make computations less time intensive, it is very important to choose proper attributes, specially in the case like ours when you have a lot of attributes.

Some exciting features not present in our dataset include storyline and movie plots. Advanced NLP analysis on the storyline might provide unexpected insights and results. Also, the information about movie release time and demographic conditions, if available, can also lead to very interesting outcomes. If these methods and data gets combined with our model, it can really make a difference in the performance of movies.

Overall, this work can be very useful for film producers to decide important factors for their movies and to analyse how to make the movie more successful.

REFERENCES

[1] Guanghui Wang Lidong Wang and Cheryl Ann Alexander. 2015. Big Data and Visualization: Methods, Challenges and Technology Progress. *Digital Technologies* 1, 1 (2015). https://doi.org/10.12691/dt-1-1-7

[2] Divyanshu Mishra. 2019. Regression: An Explanation of Regression Metrics And What Can Go Wrong. *Toward Data Science* (Dec. 2019). https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914

[3] Mladen Mik sa Sini sa Pribil Mladen Marovic, Marko Mihokovi c and Alan Tus. 2011. Automatic movie ratings prediction using machine learning. *IEEE* 54, 2 (May 2011), 1640–1645. http://www.csc.kth.se/~miksa/papers/AutomaticMovieRatingsPrediction_MIPRO.pdf

[4] Sarath Babu PB Lijiya A Nithin VR, Pranav M. 2014. Predicting Movie Success Based on IMDB Data. *International Journal of Data Mining Techniques and Applications* 3 (June 2014). https://doi.org/10.20894/IJBI.105.003.002.004

[5] David H. Wolpert. 1992. Stacked Generalization. *Neural Networks* 5 (1992), 241–259. https://www.sciencedirect.com/science/article/abs/pii/S0893608005800231

REPOSITORY LINK

Please go to below github repository to view the code. Refer to the readme file for instructions on how to run the code.
https://github.com/AmitMandliya/movie-rating-prediction

7 APPENDICES

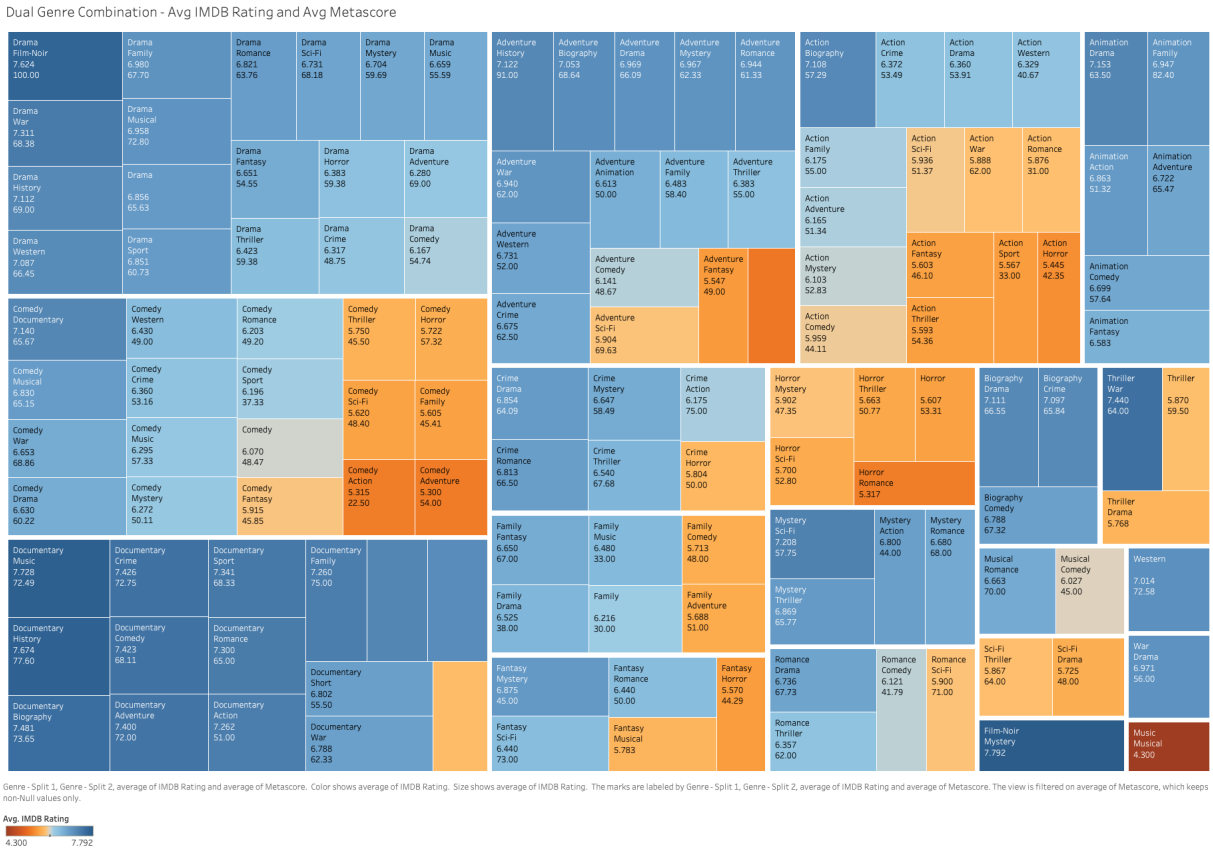

Figure 11: 2 Genre Combination TreeMap showcasing interesting inter-genre relations while also highlighting the IMDB score (out of 10) and Metascore (out of 100).