# CSCI B 565 - Data Mining - Project Report

# Movie Rating Prediction

Shardul Dabhane(sdabhane@iu.edu), Mohit Dalvi(msdalvi@iu.edu)

2nd May, 2022

## Abstract

**The IMDB rating of a movie can be considered a metric of success. If a movie has a good IMDB rating, the chances of the movie having critical and commercial success are very high. Hence, it is very important to have a framework that will help us predict the IMDB rating of a movie based on parameters like genre, the number of votes it has received on IMDB, the directors, and the writers of the movie as well. This study aims at using various regression models on two IMDB movie datasets: one from the official IMDB website and one from Kaggle, and comparing the results of these models on these datasets. We will be using models like Linear Regression, Random Forest, Ridge Regression, KNN Regression, and XGBoost Regression.**

## Keywords

Regression, IMDB, Data Mining, Movie Rating Prediction, Ensemble Methods, Machine Learning

## 1. Introduction

In today's world, the movie industry is a multi-billion dollar industry. Thousands of people, from the actors and director to the people working on lighting and sound and even catering, work tirelessly in collaboration to create a movie. Hence, it is very important that a movie is successful both critically and commercially. IMDB Ratings is one of the best metrics to check the success of a movie with the general audience. Unlike Rotten Tomatoes or Metacritic, you don't need to be a certified critic to rate a movie on IMDB. The rise of Regression models and Machine Learning techniques can help in predicting the IMDB rating of a movie. There are many factors that influence the rating of a movie like the genre of the movie, the budget, names of the cast and crew, release date, and many more. The popularity of a movie on the internet also influences the IMDB rating of a movie. A popular movie might get high ratings even with poor critical acclaim but a critically acclaimed movie might also get bad ratings on IMDB due to the movie not resonating with the audience. We used all these factors in our regression models to make the predictions. Previous research done using Regression for IMDB rating predictions shows that the accuracy achieved with these types of datasets is relatively low and it is difficult to achieve high accuracy.

The most important part of any project related to rating prediction is the dataset that we use. We applied a variety of regression models to two datasets: the larger dataset from IMDB [2] and the subset of the IMDB dataset on Kaggle[3]. The larger dataset from IMDB is a combination of 3 large tsv files: the first file contains attributes describing the name of the movie or a piece of media, a boolean attribute to depict if a movie is made for adults or not, the runtime of the movie, genres of the movie, the type of title or media it is and the year of the movie. The second file contains the crew of the movie, which is the director and the writers of the movie. Both the directors and the writers are depicted with an array of constants. The third file contains the number of people who rated the movie or the number of votes the movie has received on IMDB and the actual average rating of the movie. Each of the files contains an attribute called "tconst", which is an alphanumeric unique identifier for a movie. This large dataset contains over records. The smaller dataset of movies from Kaggle has 12 attributes and 1000 records. It contains movies released from 2006 to 2016. It contains attributes like the name of the actors, directors, genres, runtime of the movie, Metascore, revenue earned in the millions in the USA box office, description, number of votes earned, and the rating. We used these attributes to predict the average

IMDB rating of a movie, after performing some data preprocessing, feature selection, and data analysis.

## 2. Methods

We followed the flow of the model as described in figure 9. Before we performed EDA and Data Preprocessing on the large dataset, we needed to create it first. We read each file and joined the columns using “tconst” as a unique identifier to join the columns. Once we joined the columns, we removed the extra rows from the dataset. We then also remove all the other types of titles from the dataset like “short”, “movie”, “video”, “tvSeries” and only kept the rows with “movie” as the type of title. We then performed one hot encoding

After creating the larger dataset, we performed EDA on this dataset. We created the scatter plot for the number of votes received vs the rating of the movie

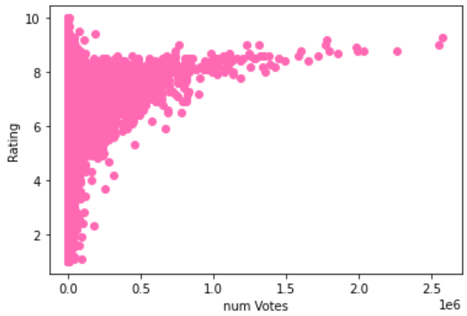


Fig 1: Number of Votes vs Rating Scatter Plot

As you can see, very few movies have received more than 100000 votes. But if the movie has received more votes, chances are that the movie has a high rating. We also visualized the number of votes each movie received, by using a bar graph to plot the number of votes each movie has received in the form of a horizontal bar graph. We used bins to set the range for the number of votes in each bar.

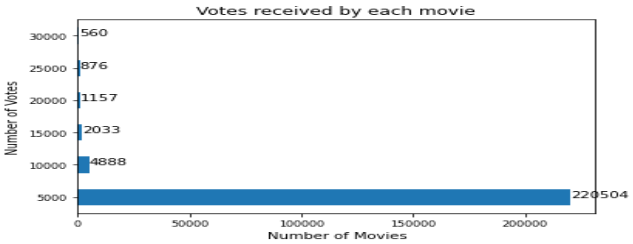


Fig 2: Number of Movies vs Number of Votes Bar Plot

We observe that most of the movies have received 5000 or less than 5000 votes. We also created a pie chart to describe the percentage of movies belonging to each genre.

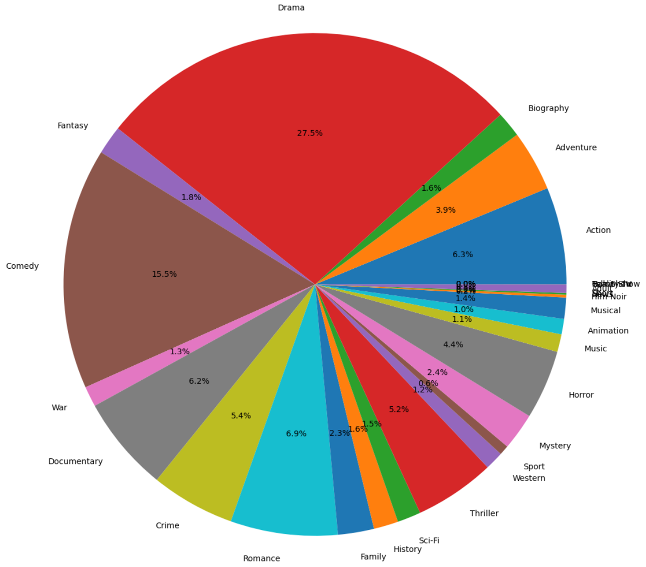


Fig 3: Pie Chart for the percentage of each genre in the dataset

As we can see, drama is the most dominant genre with 27.5% of the movies belonging to the drama genre. After drama, comedy is the most dominant genre with 15.5%. We then used one-hot encoding to separate the genres as individual attributes. With this information, we calculated the correlation of the attributes and kept the ones with the most positive and negative correlation. Both ‘Drama’ and ‘Comedy’ attributes were selected as the relevant attributes. We then used data imputation to replace the “NaN” values with 0.

Similarly for the Kaggle dataset, we performed EDA and Data Preprocessing. We drew the plot of the number of movies released each year in the form of a bar graph.

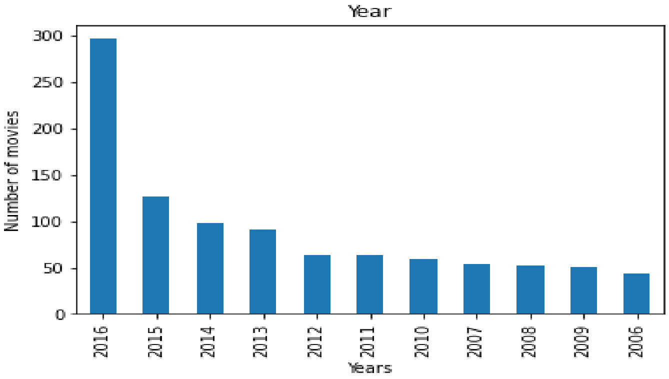


Fig 4: Bar graph for the number of movies released per year

Most of the movies released are from 2016, although there are enough movies released from each year in the dataset. After finding the correlation matrix, we found that the IMDB rating of a movie and its Metascore(critic score) are highly correlated. We plotted the relationship between Metascore and IMDB rating using a Scatter plot.

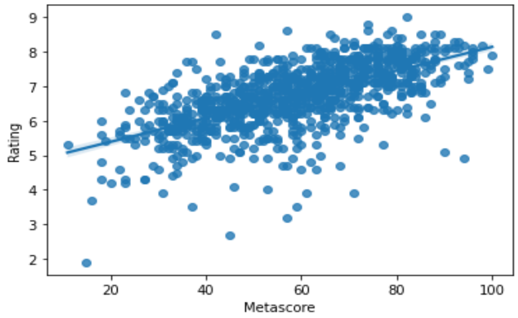


Fig 5: Scatter Plot of Metascore vs IMDB rating

Using EDA, we can also find the highest rating for each genre.

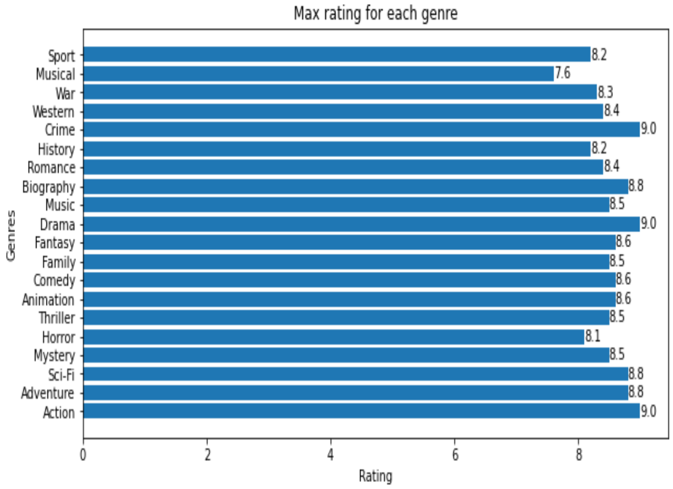


Fig 6: Horizontal bar graph for highest IMDB rating per genre

As we can see from the bar graph, the Action, Crime, and Drama genres have the highest rating because the highest-rated movie in the dataset is “The Dark Knight” with a 9.0 rating. Its genres are Action, Crime, and Drama. The musical has the lowest highest rating out of all genres with a value of 7.6. We can also plot the average rating for each genre using a bar graph.

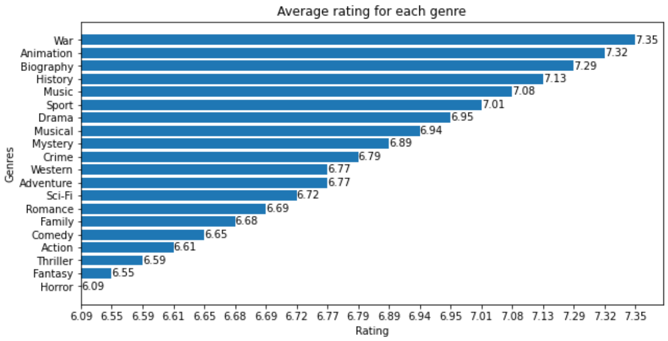


Fig 7: Horizontal bar graph for average IMDB rating per genre

The average rating for the War genre is the highest with a value of 7.35. The Horror genre has the lowest average rating of 6.09. Based on this graph, we were curious to see how many movies belong to each genre. Just like for the large dataset, we created a pie chart depicting the percentages of each movie genre for the Kaggle dataset as well.

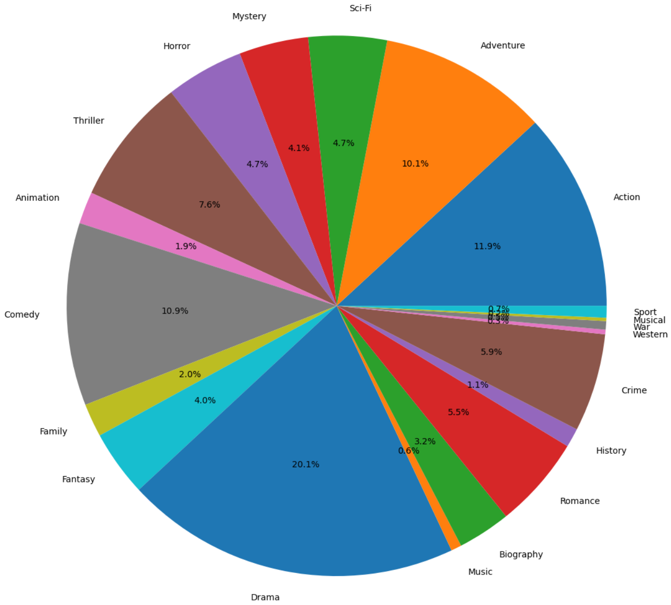


Fig 8: Pie Chart for the percentage of each genre in the dataset.

As seen from the pie chart, here too, Drama is the dominant genre with 20.1% of the records belonging to the genre. Action has the second-highest percentage with 11.9%. We then used one-hot encoding to separate the genres as individual attributes. With this information, we calculated the correlation of the attributes and kept the ones with the most positive and negative correlation. Only Drama was selected out of Drama and Action genres as one of the relevant attributes. We then used data imputation to replace the “NaN” values with 0.

After performing EDA and data preprocessing on both datasets, we performed various Regression models on our datasets. Our goal is to reduce the error metrics like mean absolute error, mean square error, and root mean square error.

1. **Linear Regression**.: A very simple, linear model where we assume that there is a linear relationship between the dependent and independent variables. We predict the value of dependent(target) variables based on one or more independent variables. Primarily it checks that a set of predictor variables does a good job in predicting an outcome variable. It takes into account the significant attributes that have the most influence on the target variable. The equation for Linear Regression is as follows:

   $$Y = w_1X_1 + w_2X_2 + w_3X_3+......w_nX_n$$

   where, $X_1$, $X_2$,...$X_n$ are independent variables and $w_1$, $w_2$, …..$w_n$ represent the corresponding weights.

2. **K-Nearest Regression**: According to (Teixeira-Pinto, 2021)[4], it is a non-parametric method that uses the relationship between the independent variables and the average of the observations belonging to the same neighbourhood. KNN regression uses the same distance functions as KNN classification. Distances used for K-Nearest Regressor are Euclidean, Manhattan, and Minkowski.

3. **Random Forest Regression**: Random forest regression is an ensemble learning technique. In ensemble learning, you take multiple algorithms or the same algorithm multiple times and put together a model that's more powerful than the original. In Random Forest Regression, we use a meta estimator that fits multiple decision trees on various sub-samples of the dataset and performs averaging to improve the predictive accuracy and avoid overfitting. This prediction is more accurate as it uses the predictions made by multiple decision trees. The algorithm is stable as any changes in the dataset will impact individual trees but not the group of trees or "forest" as a whole.

4. **XGBoost Regression**: XGBoost stands for extreme gradient boosting. It is a very powerful ensemble learning method that uses base learners and combines all their predictions together. The bad predictions from all base learners would get canceled giving us a much better result. XGBoost minimizes the regularized L1 and L2 objective functions and combines a convex loss function and a penalty term for model complexity. It was developed in 2016 by Tianqi Chen[2] to work on billion examples using far few resources.

5. **Ridge Regression**: It is a method to estimate the coefficients of multiple-regression models where the independent variables are highly correlated. We use it to "shrink" the parameters(weights) of the model so that the predictions are less sensitive to any change in the input. Ridge regression is also called L2 regularization. Its equation is:

   $$\mathrm{argmin}_w(||y-X_w||_2\textasciicircum 2)+\alpha||w||_2\textasciicircum 2$$

   $X_w$ is the set of independent variables and y is the dependent variable. Alpha is a coefficient of the tuning parameter, which is a shrinkage penalty term. w is used to denote the weights.

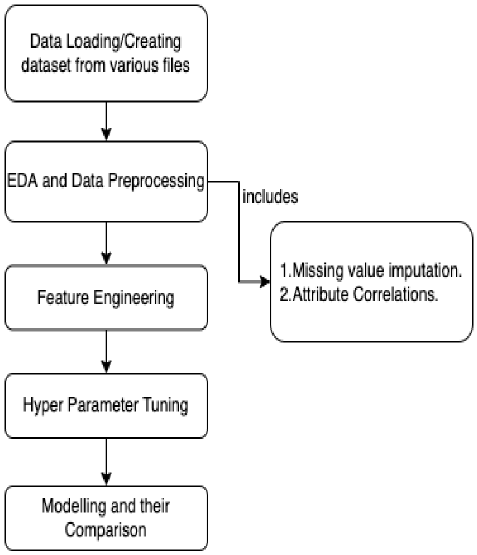
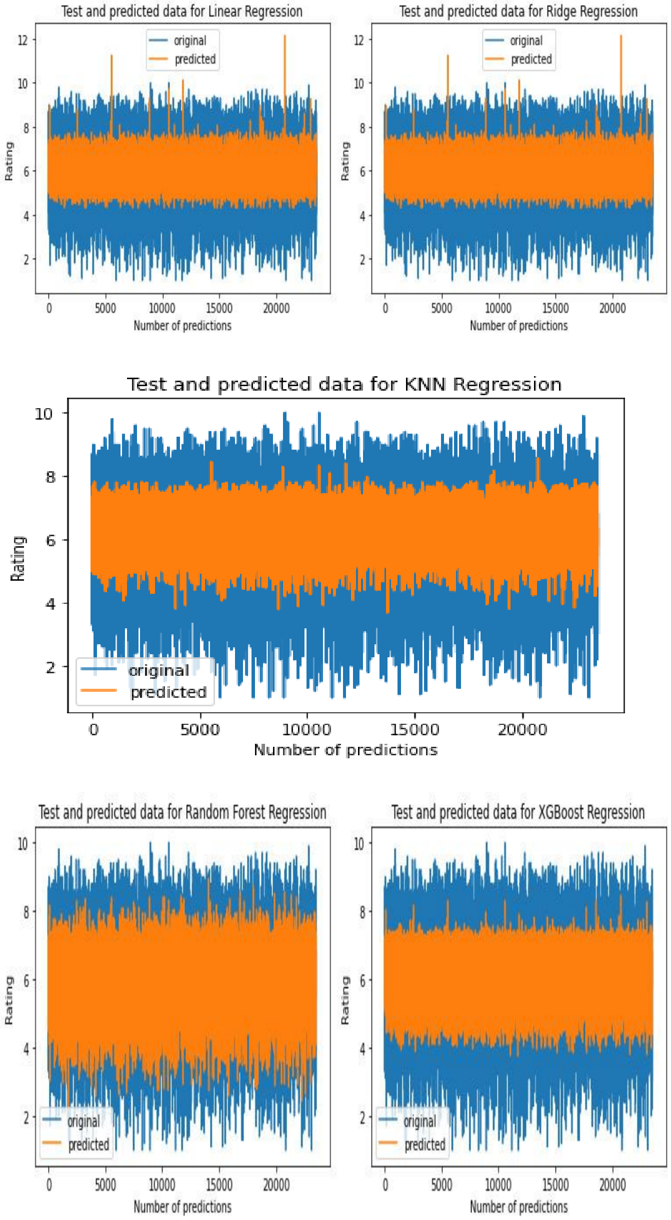

Data Loading/Creating dataset from various files
EDA and Data Preprocessing
includes
1.Missing value imputation.
2.Attribute Correlations.
Feature Engineering
Hyper Parameter Tuning
Modelling and their Comparison

Fig 9: Flow of Model

## 3. Results and Discussion

We obtained the following results on applying Linear Regression, K - Nearest Regressor, Random Forest Regressor, and Xgboost to the preprocessed data. The models predicted the value of movie ratings based on the values of the independent attributes of the data. We plotted the predicted and actual values of both the IMDB and Kaggle datasets.


Test and predicted data for Linear Regression

Test and predicted data for Ridge Regression

Test and predicted data for KNN Regression

Test and predicted data for Random Forest Regression

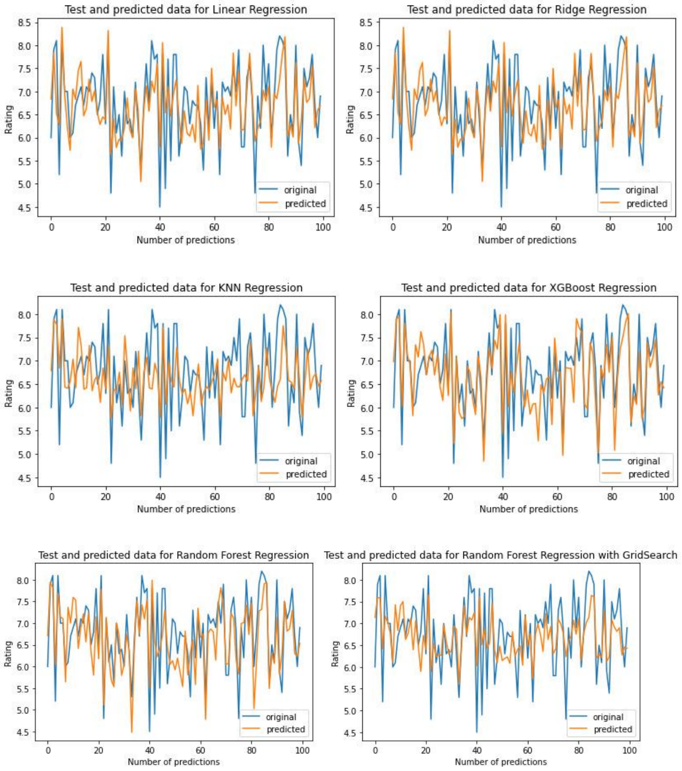
Test and predicted data for XGBoost Regression

Fig 10 : Predicted vs Observed data of Large IMDB Dataset

Table 1: Error function values for Large IMDB Dataset

| (IMDB Large Dataset) | Linear Regression | Ridge Regression | K-Nearest Regression | Random Forest Regression | XGboost Regression |
|---|---|---|---|---|---|
| **MSE** | 1.425 | 1.425 | 1.483 | 1.488 | 0.895 |
| **MAE** | 0.922 | 0.922 | 0.944 | 0.941 | 1.348 |
| **RMSE** | 1.193 | 1.193 | 1.217 | 1.220 | 1.161 |

For the large dataset, we used various splits and the best results came with 90% data for training and 10% data for testing. We used the models from Python on the dataset and got the results. We have plotted the test and predicted data and it is showing high values of the error metrics. We have also created a table that conveys the Mean Square Error(MSE), MAE(Mean Absolute Error), and the Root Mean Square Error (RMSE) that are obtained when these models are tested on the test data.


Test and predicted data for Linear Regression

Test and predicted data for Ridge Regression

Test and predicted data for KNN Regression

Test and predicted data for XGBoost Regression

Test and predicted data for Random Forest Regression

Test and predicted data for Random Forest Regression with GridSearch

Fig 11: Predicted vs Observed data of Large IMDB Dataset

Table 2: Error function values for Kaggle IMDB Dataset

| (Kaggle Dataset) | Linear Regress-ion | Ridge Regression | K-Nearest Regression | Random Forest Regress-ion | Random Forest Regress-ion with GridSearch | XGboost Regress-ion |
|---|---|---|---|---|---|---|
| **MSE** | 0.3488 | 0.3486 | 0.576 | 0.374 | 0.404 | 0.278 |
| **MAE** | 0.483 | 0.483 | 0.636 | 0.480 | 0.504 | 0.413 |
| **RMSE** | 0.5905 | 0.5904 | 0.759 | 0.611 | 0.636 | 0.527 |

For the Kaggle Dataset, we used the same 90:10 split and used the models from Python to fit on the data. The results from this dataset are much better as we get low values for all the error functions which indicate that we get a better model. The "Revenue (Millions)" attribute in the dataset could have contributed to better results for this dataset.

## 4. Conclusion and Future Scope.

We observed that the attributes in the smaller dataset were more correlated with the target than the attributes in the larger dataset and hence the accuracy of the models was more for the smaller dataset. Additional attributes like the revenue of the movie and critics' scores of the movie may have helped in getting low error function values. The rating problem is complicated due to the wide range of values possible for the IMDB rating and we need better attributes for predictions, whether it's through feature engineering or by data preprocessing.

We also understood that choosing the right dataset is very important since a lot depends on the data to make accurate predictions.

To improve accuracy it is possible to do further feature engineering with some attributes. On the larger dataset accuracy can be increased by applying hyperparameter tuning techniques like GridSearchCV. We can also look into Neural Networks as one of the methods to use to predict the IMDB rating of a movie. We can also use cross-validation for all models to obtain better results in the future.

## References:

[1] Tan, Steinbach, Kumar, Karpatne, "Introduction to Data Mining 2nd Edition"

[2] IMDb.com.(n.d.).IMDb. May 2, 2022. https://www.imdb.com/interfaces/

[3] IMDB data from 2006 to 2016. (2017, June 26). Kaggle. https://www.kaggle.com/datasets/PromptCloudHQ/imdb-data

[4] Teixeira-Pinto, A. (2021, August 2). Machine Learning for Biostatistics. 2 K-nearest Neighbours Regression. Retrieved May 2, 2022, from https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html

[5] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, pp. 785–94. arXiv.org, https://doi.org/10.1145/2939672.2939785.

[6] Awan, M. J., Khan, R. A., Nobanee, H., Yasin, A., Anwar, S. M., Naseem, U., & Singh, V. P. (2021). A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. Electronics, 10(10), 1215. https://doi.org/10.3390/electronics10101215.

[7] Mhowwala, Z., Razia, A., & D., S. (2020). Movie Rating Prediction using Ensemble Learning Algorithms. International Journal of Advanced Computer Science and Applications, 11(8). https://doi.org/10.14569/ijacsa.2020.0110849

[8] B. Çizmeci and Ş. G. Ögüdücü, "Predicting IMDb Ratings of Pre-release Movies with Factorization Machines Using Social Media," 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 173-178, DOI: 10.1109/UBMK.2018.8566661.

[9] A. Bhave, H. Kulkarni, V. Biramane and P. Kosamkar, "Role of different factors in predicting movie success," 2015 International Conference on Pervasive Computing (ICPC), 2015, pp. 1-4, DOI: 10.1109/PERVASIVE.2015.7087152.

[10] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[11] Bristi, Warda & Zaman, Zakia & Sultana, Nishat. (2019). Predicting IMDb Rating of Movies by Machine Learning Techniques. 1-5. 10.1109/ICCCNT45670.2019.8944604.

[12] Jeffrey Ericson, Jesse Grodman; "A Predictor for Movie Success" ; Stanford University, 201