­Part 3: Data Mining

The movie data was collected from the IMDB web site which includes columns such as movie title, title year, duration, director name, actor name, user votes, critic reviews, gross, budget and imdb score, etc. The dataset when loaded had around 5000 instances with some missing values. We replaced all the missing values using weka filter function as the missing values were in small percentage with respect to the total instances. As there were around 20 columns we had to do feature engineering i.e. we used weka’s attribute selection (Best First) method to select the important attributes needed for our tasks.

We selected imdb score as a target variable as it was a best fit for measuring the greatness of any movie. As imdb score is a continuous numeric attribute we thought that the numeric prediction algorithms would work better in terms of performance. Classification algorithms wouldn’t work as most of the dataset includes numeric attributes. Similarly, clustering and association finding wasn’t used as our aim was about measuring the effectiveness of movies based on imdb score.

As we began using numeric algorithms like Linear regression for prediction the weka kept crashing. So we reduced to 8 no of attributes. When it comes to measuring the greatness of movie, apart from imdb score, a movie could get high gross through director popularity, actor popularity or country. There were more than 2500+ director names and several actor names which are why our weka kept on crashing when considering these attributes. We even used Discretization and NominaltoBinary filters on these attributes to encode the category values but still weka kept on crashing. Also, we used only one category attribute for e.g. Director Name or Movie Title and rest numerical attributes for building model. But, the Numerical algorithms took more than 10-15 minutes and yet model wasn’t build. The reason was too many category values to handle while training the data. Due to this reason we had to only use country as a nominal attribute (category) and discard the important features like director name, actor names, etc.

Following are the visualizations (golden nuggets/patterns/insights) of the data:-

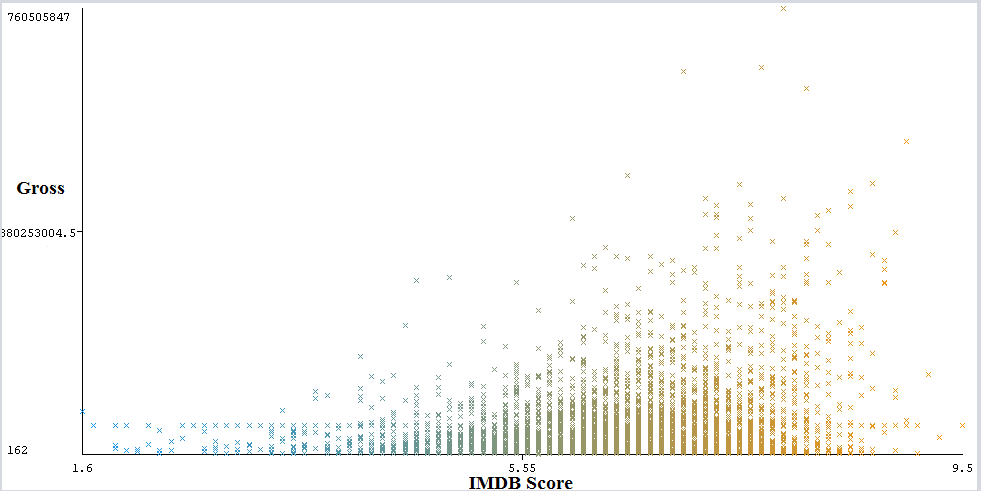


Fig 1

In the Fig 1, we can see that the lines of points are higher in between 6 to 8 IMDB Score. The gross value also increases for the 6 to 8 imdb scores. This means that as the gross value increases the imdb score is expected to be between 6 and 8.

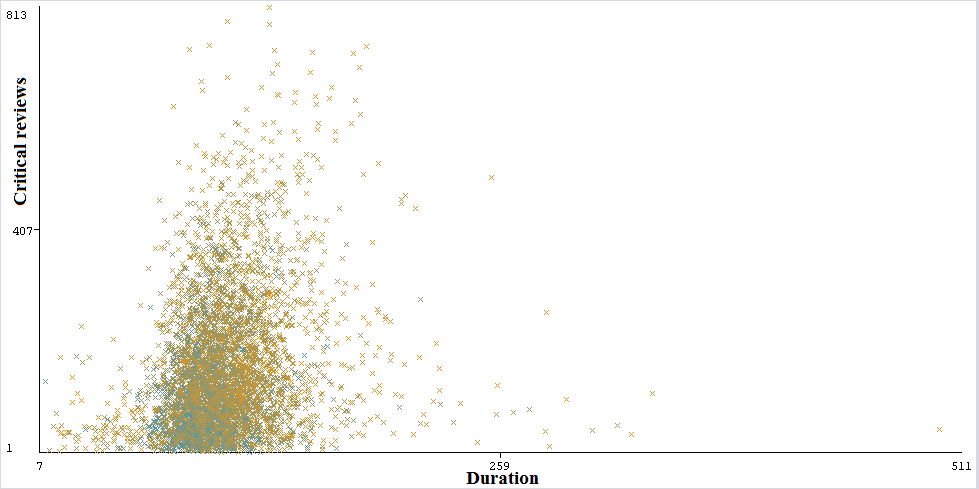


Fig 2

The blue points are IMDB scores between 1 and 6-5 whereas the brown points are IMDB scores between 6 and 9.9. In the Fig 2, we can see that for the duration between 60 and 124 (in minutes) the majority of the points are clustered together. The brown points (6-9.9 imdb score) tends to increase as the numbers of critical reviews increases. This means that if the critical reviews are higher the imdb scores are between 6 and 9.9 for the duration of the movies between 60-124 minutes.

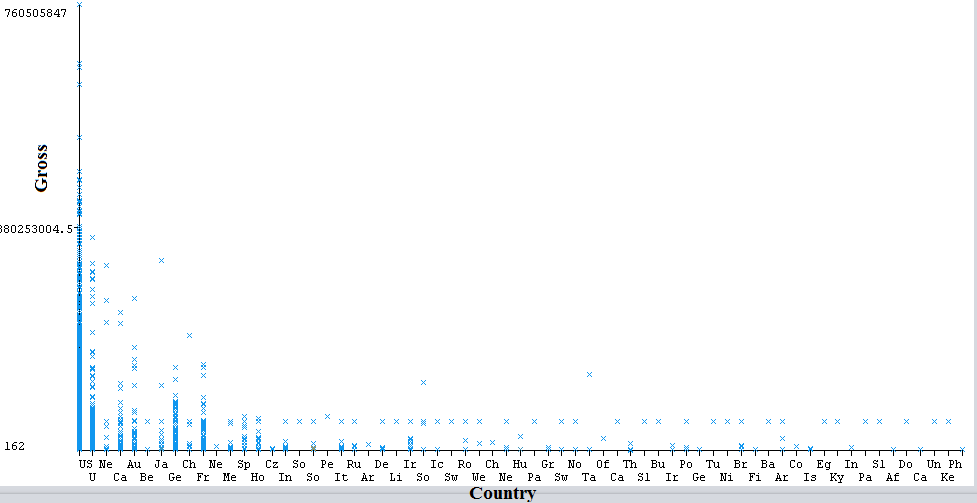


Fig 3

In the Fig 3, we can conclude that, US (United States), U (United Kingdom), Ca (Canada), Au (Australia), Ge (Germany) and Fr (France) are the top 6 countries with higher movie gross (in Dollars) values.

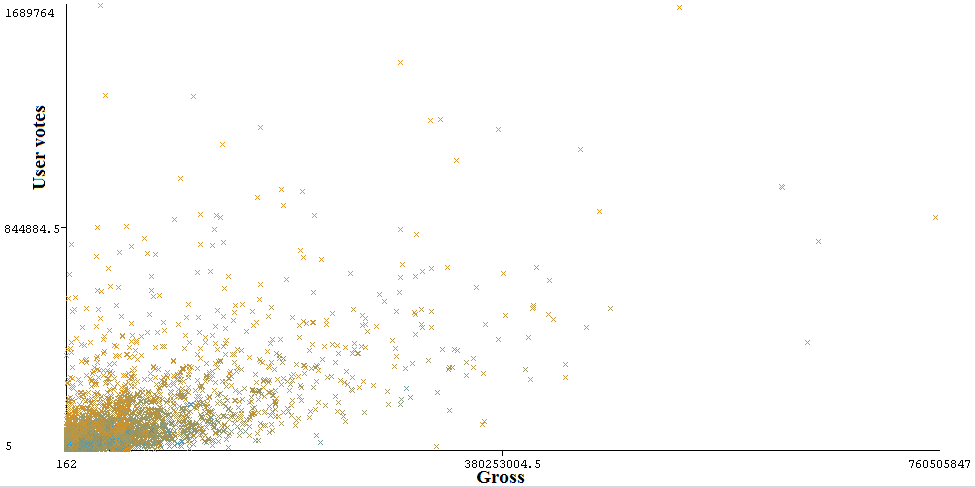


Fig 4

In the Fig 4, we can say that the line of points looks roughly linear. That means that as the gross value increases the User votes tends to increase too. Also, by combining all the visualizations we can conclude that Gross, User votes, Critical reviews and IMDB scores are closely correlated with each other. This also provides evidence that IMDB scores could be used as a target dependent variable since its more interpretable than the values of the other 3 features. Lets move on to predicting the model.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithms** | **Training error (RAE)** | **Cross Validation error (RAE)** | **Overfitting** |
| IBk | 0% | 98.46% | YES |
| Random Forest | 25.67% | 69.35% | YES |
| M5P | 66.71% | 75.46% | YES |
| Linear Regression | 84.97% | 86.33% | NO |

We chose M5P and Linear Regression as both works on predicting Numerical class. In some cases, random forest could be used to classify the target variable. IBk was chosen as it works when more category values are to be used. In the above table, we can see that only Linear Regression algorithm is not overfitting while rests of the algorithms are overfitting. So we chose to perform hyper-parameter tuning on Linear Regression.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Parameters** | **Correlation Coefficient (For Training set )** | **Correlation Coefficient (For Cross Validation)** | **Training error (RAE)** | **Cross Validation Error (RAE)** | **Overfitting** |
| Linear Regression | batchSize = 50 | 0.49 | 0.47 | 84.97% | 86.33% | NO |
| Linear Regression | batchSize = 100 | 0.49 | 0.47 | 84.97% | 86.33% | NO |
| Linear Regression | numDecimalPlaces = 4 | 0.49 | 0.47 | 84.97% | 86.33% | NO |
| Linear Regression | batchSize = 400 | 0.49 | 0.47 | 84.97% | 86.33% | NO |
| Linear Regression | numDecimalPlaces = 20 | 0.49 | 0.47 | 84.97% | 86.33% | NO |

In the above table, we can say that despite changing the parameter values, the difference errors and coefficient values remains constant. We found that only the time taken for each run was different.

Finally, we can conclude that Linear Regression works better in terms of prediction compared to other models. This is because the target variable is a continuous numeric attribute where in general Linear Regression is used to predict the numerical values. As we mentioned earlier, we had to discard important features like Director Name, Actor Name, Movie title for the building an efficient model (avoid weka crash). Also, since the categorical values were too many for these attributes, visualizations in Weka using these attributes couldn’t be interpreted. The model could now only be able to predict score based on a country which includes lot of mislabelled errors.