# METHODOLOGY DETAILS

## Dataset

The dataset from Kaggle is based on the imdb ratings of movies in a .csv format. It contained 5,000 entries with 28 attributes namely, imdb\_score, budget, gross, director\_name, cast\_total\_facebook\_likes, movie\_name, num\_critic-for\_reviews, num\_user\_for\_reviews etc. The visualization on the dataset provided showed that imdb score of 6.5 having the highest frequencies.

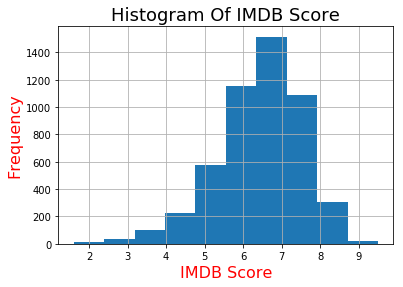


Figure 2: Frequency of IMDB Score of raw dataset

## Data Cleaning

The original dataset [movie\_metadata] had 28 attributes from which 9 attributes were removed for more accurate findings, thus obtaining a cleaned dataset [processed\_df]. Some numeric data having zero or missing values were transformed using their mean values while the missing values of categorical data were classified into a missing category altogether, besides removing the duplicate values. To transform categorical data for some attributes, one-hot encoding and label encoding of scikit-learn library was used as most machine learning algorithms work on numerical data viz. country[USA] was encoded into country\_code[63], content\_rating [PG-13 ] was transformed to content\_rating\_code[ 8] etc.

## Data Pruning

The cleansed data was pruned according to the density found in the num\_user\_for\_reviews v/s imdb\_score visualization. Over a period of several incremental iterations, we selected entries where num\_user\_for\_reviews>20.

## Algorithms

We have picked three machine learning algorithms namely, Logistic Regression, Random Forest and Support Vector Machine to make a comparison of their accuracies on the pruned dataset.

*2.2.1 Logistic Regression.* The algorithm gives adjusted probability of classes and works well on large number of observations. The algorithm was chosen as it mostly have a fast prediction speed as well as training speed. We also found out that the algorithm gave best measure when c-param=1 for its range from 0.001 to 1000 with a ten times increase for each point. The following optimized problem is solved by the L1 regularized logistic regression.

p(y = 1|x; θ) = σ(θx) = 1/( 1 + exp(−θx) )

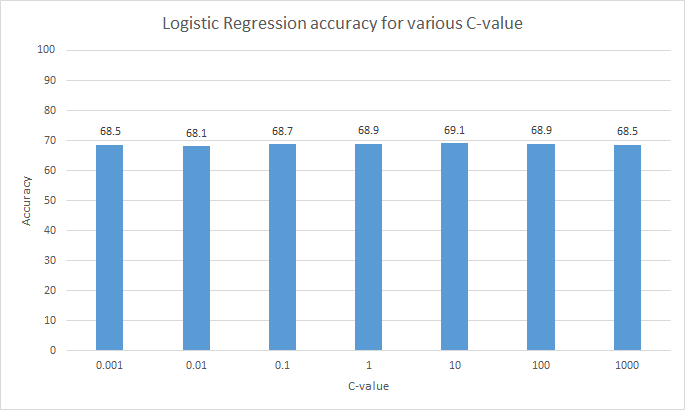


Figure 3: c-param accuracy of logistic regression

*2.2.2 Random Forest.* The choice for this algorithm is due to its effiecient handling of loads of irrelevant features of dataset unless the noise-ratio is high. The n-estimator for random forest for range [10,100,10] was found best for 40 on an unpruned data. Hence, we applied it on the pruned data as well.

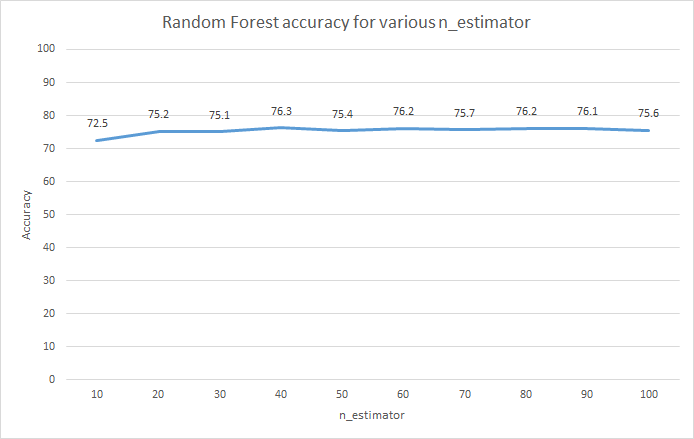


Figure 4: n-Estimator accuracy for random forest

*2.2.3 Support Vector Machine.* In our dataset, at some points there is presence of large margins which eventually help us in generalization. We ran the SVM weights to prevent overfitting on larger margins. For c-param [0.001, 0.01,0.1,10,25,50,1000], SVM was found to perform best for c-param=0.1. On the whole, this helps the test data to perform well as it doesn’t overfit the training data.

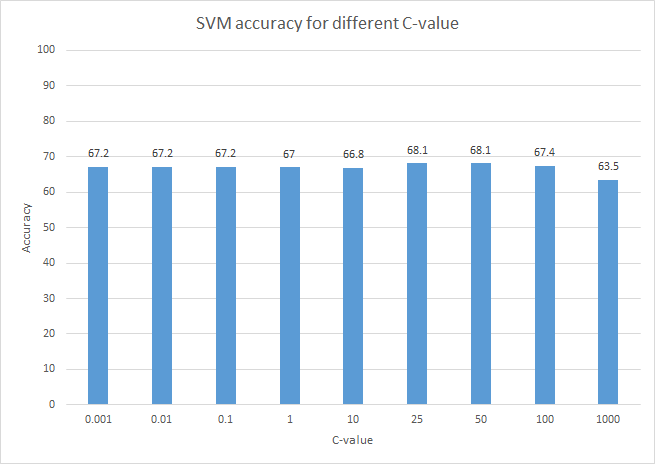


Figure 5: c-value accuracy of SVM

## Evaluation

Out of three divided train-test groups, it was found that a 20% test data would give an able accuracy measure. This was manually tested and eventually we went with the 80:20 split ratio for train-test datasets.

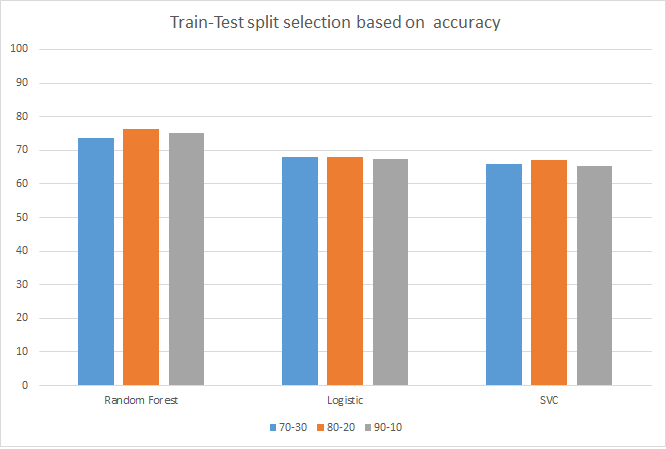
**

Figure 6: Accuracy of train-test dataset split for various algorithms