Predict Movie Revenue

Group 3

Why?

- Film producers spend millions of dollars to produce a high budget movie, and risk their money on the filmmaking.

- It becomes really important for them to analyse the worth of taking that risk and know the expected target revenue.

Benjamin Et Al. (2015)

- Features prior to release from OMDB API - genre, length, release date, actors and directors, locality.
- Linear regression with revenue buckets. - (not good)
- Naive Bayes classification - 51.6%
- Multinomial support vector machine - 49.4%
- Multinomial logistic regression - 50.9%
- Drawback - Sparsity of data in buckets with the densest bucket containing more than 80% of the data points.

Jason Et Al. (2013)

- Feature set - The Rotten Tomatoes API.
- The feature set was represented in the form of a sparse vector of 0s and 1s.
- Linear regression on the raw data - 30% (margin of 100% error)
- Logistic regression and additional k-means clustering - 52%
- Drawback - Feature set which included every unique actor and director was too large to handle and slowed down the process significantly.

Darim Et Al.

- The data worked on was obtained from the IMDB manually.
- Features - genre, user rating, budget, run-time, MPAA ratings.
- Linear regression using gradient descent - 25.3%.
- Linear buckets were used for logistic regression of variable size which gave the resultant high error of 57.1%.
- Predicted the profitability with an accuracy of 72.4%
- Drawback - some features were unavailable pre-release (user review) and success of actors and directors were not taken into account.

Ramesh Et Al. (2006)

- Data consisted of 834 movies in the time period of 1998-2002 from ShowBiz Data.
- Features - MPAA rating, Competition, Star value, Genre, Technical effects, Sequel and Number of screens.
- A multi-layer perceptron neural network architecture using 2 hidden layers and sigmoid function - 36.9%
- Drawback - The absence of key features such as movie budget and also the linear categorization instead of logarithmic classes.

Dataset

- Famous IMDB Movie database which contains data of more than 5000 movies
- Has 5043 instances with 28 features.

Features

- Title/Name of Movie
- Color represents if movie is Colored or Blank and white
- Number of Critics for Reviews for the Movie
- Movie’s Facebook Page Likes
- Duration of the Movie
- Director’s Name, Director’s Facebook Page Likes

- Actor 1,2,3’s Name, Actor 1,2,3’s Facebook Page Likes
- Gross Revenue Collection
- Genre of Movie
- Number of Voted Users
- Cast’s total Facebook Likes
- Number of Faces on Poster
- Plot Keywords
- Movie’s IMDB Link
- Number of user for Reviews
- Language of Movie, Country in which Movie Released
- Content Rating
- Budget of the Movie
- Title Year
- IMDB Score
- Aspect Ratio.

Feature Selection

Some features will have no significant role in the prediction as all the movies have different value for that variable.

Number of Critics for Reviews for the Movie, Movie’s Facebook Page Likes, Duration of the Movie, Director’s Facebook Page Likes, Actor 3’s Facebook Page Likes, Actor 2’s Facebook Page Likes, Actor 1's Facebook Likes, Genre of Movie, Number of Voted Users, Cast’s total Facebook Likes, Number of user for Reviews, Content Rating, Budget of the Movie, Title Year, IMDB Score, Language.

Genre and Content rating

Genre

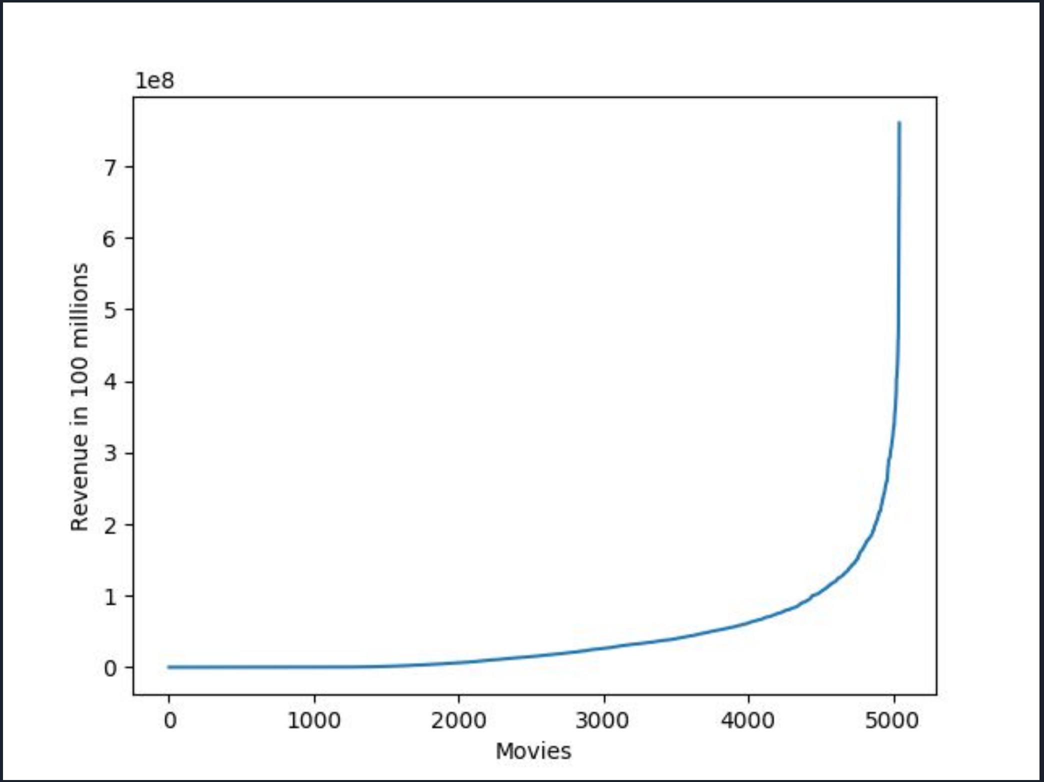
Action, Adventure, Fantasy, Sci-fi, Thriller, Comedy, Family, Horror, War, Animation, Western, Romance, Musical, Documentary, Drama, History,Biography, Mystery, Crime.

Content Rating

General Audiences, Parental Guidance Suggested, Parents Strongly Cautioned, Restricted, Adults Only

Normalization

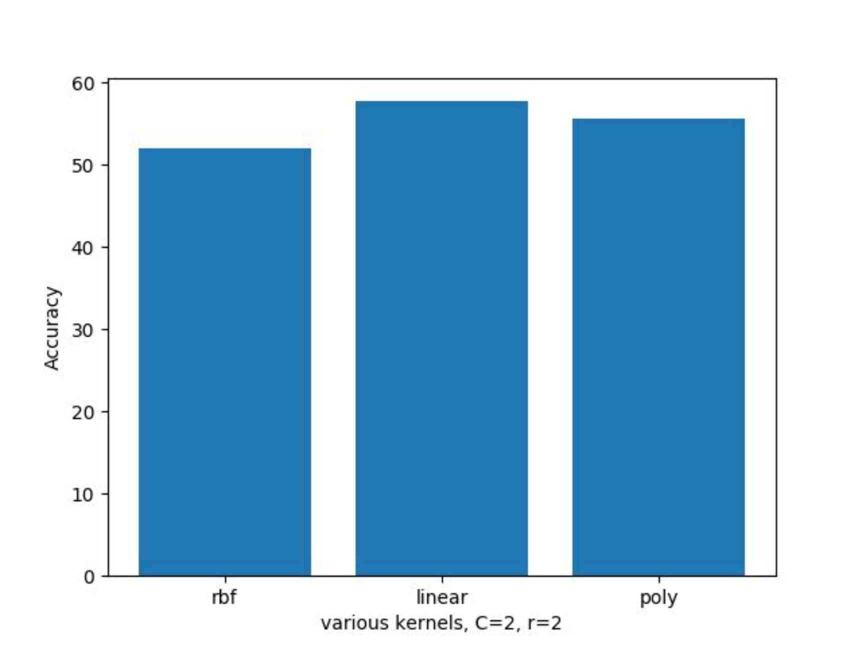
- Features such as Budget, Number of facebook likes, etc. had a larger range of values. We normalized the data such that the values had a mean of 0 and a variance of 1.
- The predicted revenue was classified into logarithmic buckets. We did this by converting the value of the revenue to its logarithm base 10 and rounding down to the nearest integer.

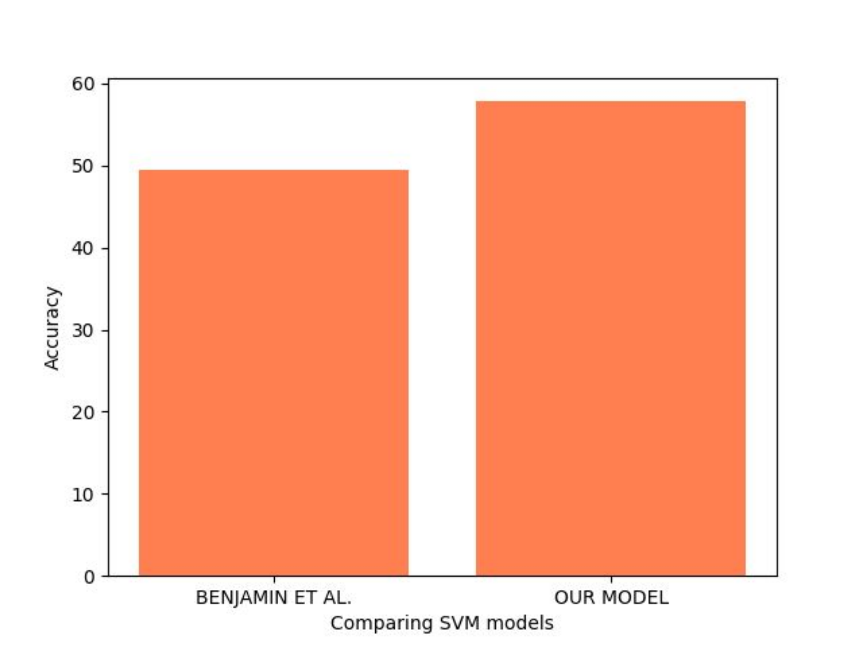


Values from 0 to 8

SVM

- Used for classification into buckets.
- C = 2, gamma = 2.
- Various kernels used - rbf, linear, polynomial.
- Linear gave best results.
- Reason - SVM is very effective in high dimensional spaces and our number of parameters taken into account was significantly high.



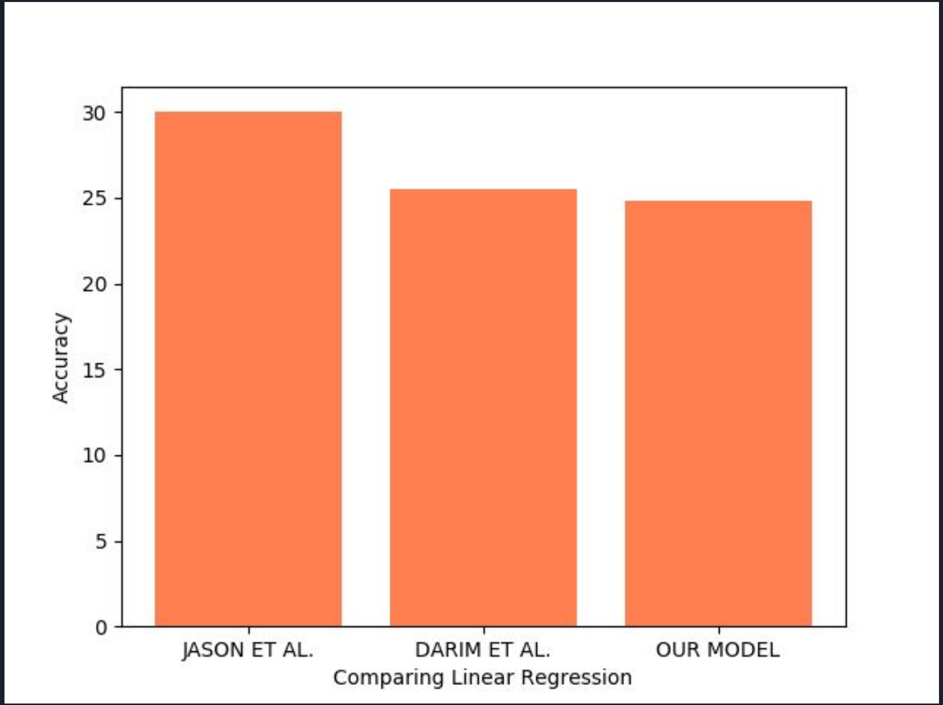


Linear kernel with C=2, gamma=2
Accuracy = 57.8%

Benjamin Et Al. Accuracy = 49.4%
This was because the dataset used by them was insufficient to fill up the logarithmic buckets significantly.

Linear Regression

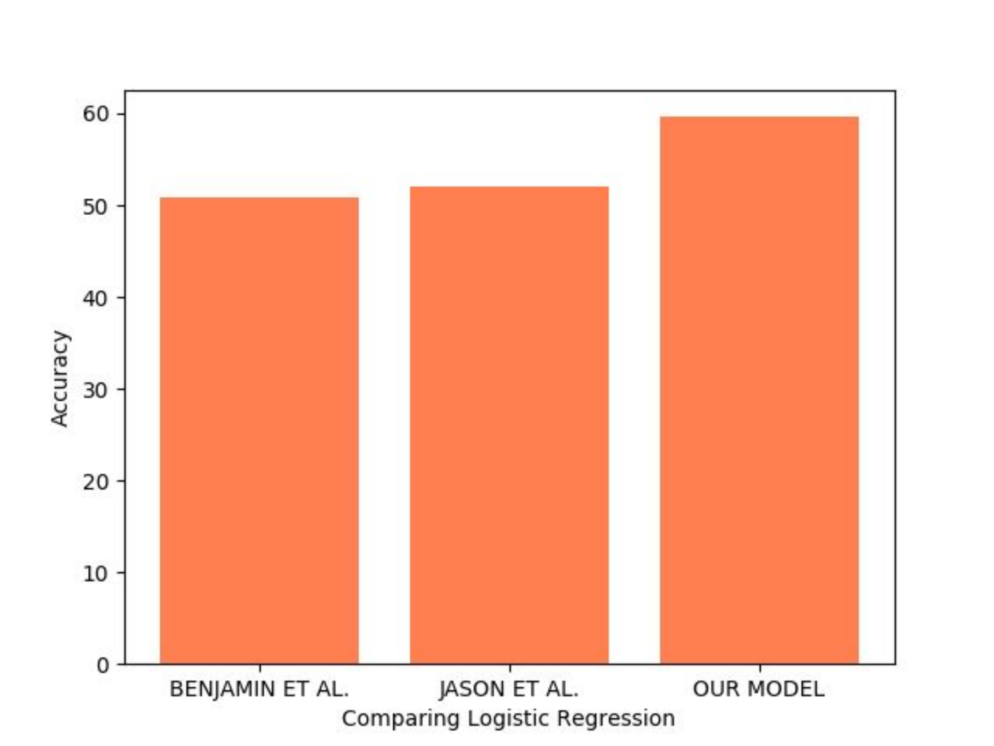
- It fits a linear model to minimize the sum of squares by observed response and predicted response by linear approximation.
- As the feature space was so large in our case, the linear model performed poorly with just 24.9% accuracy.
- Previous works also perform poor on Linear Regression.
- Reason- Overfitting due to large feature set.



Darim Et Al. = 25.3%
Our model = 24.9%

Logistic Regression

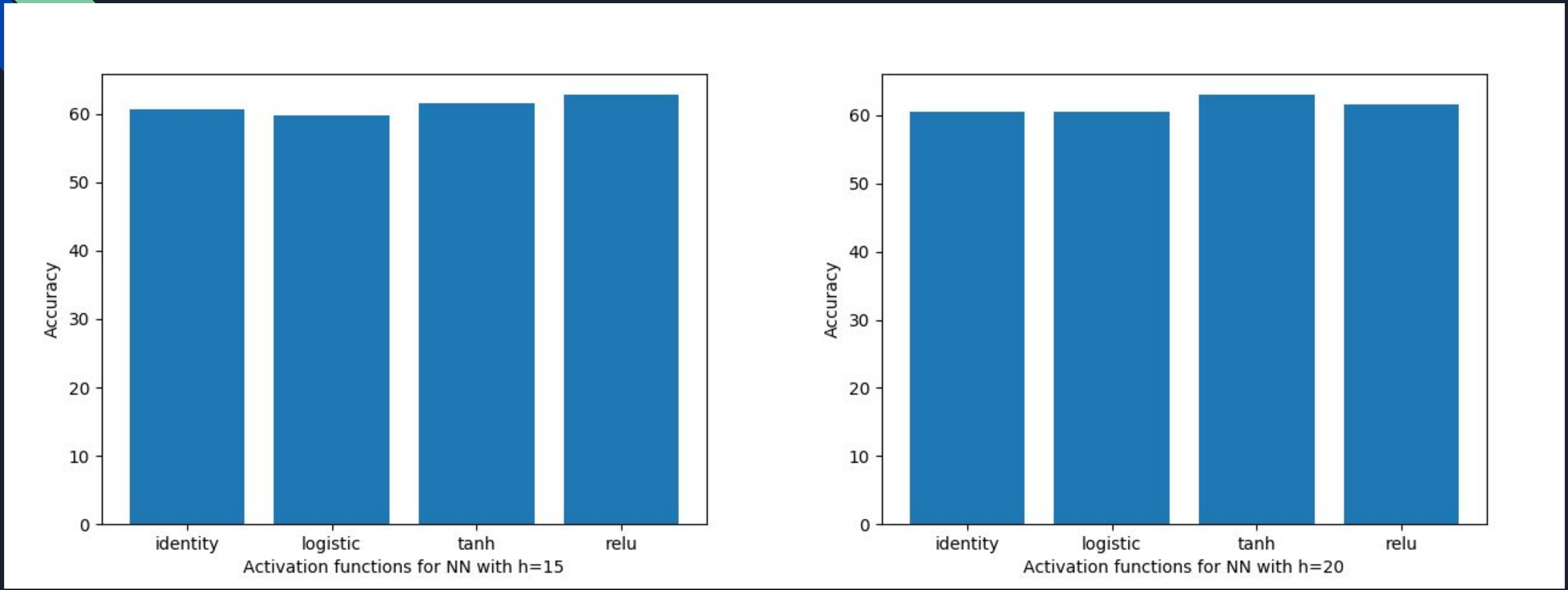
- Our logistic regression model was multinomial which classified the revenues into the 9 predefined classes.
- Logistic regression as expected performed significantly better than linear regression with an accuracy of 59.6%
- Reason - Feature selection was optimal and dataset was abundant.



Benjamin Et Al. = 50.9%
Jason Et Al. = 52%
Our model = 59.6%

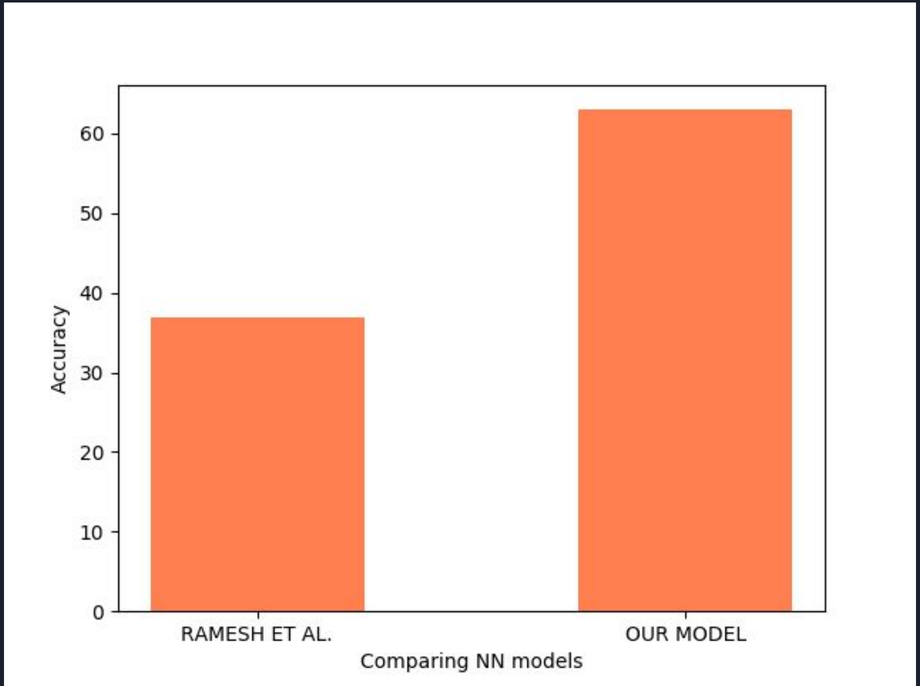
Neural Network

- Used the Multi-layer Perceptron (MLP) that learns a function by training on a dataset. We used it for performing classification on our target labeled from 0-8.
- We implemented MLP with 15 and 20 hidden layers using various activation function - identity, logistic, tanh, relu.
- Reason - It is capable of learning complex non linear models.



With 15 layers ‘relu’ gave 62.8%   With 20 layers ‘tanh’ gave 63%

Iterations = 200



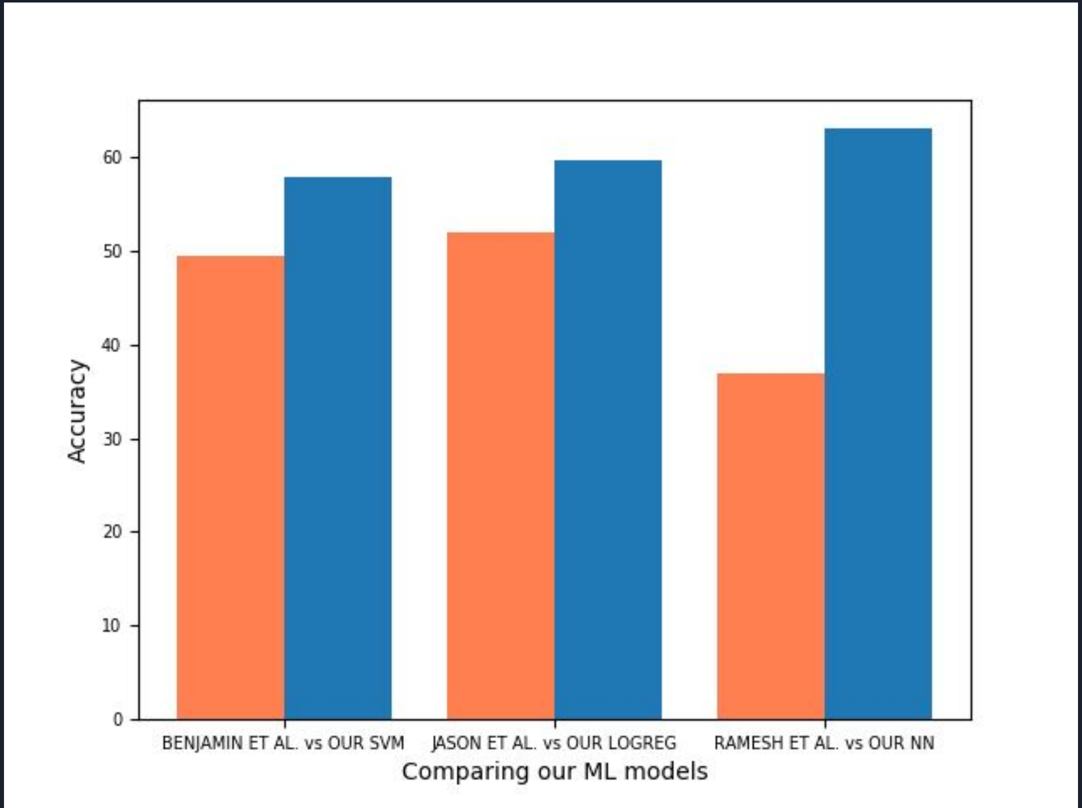
Ramesh Et Al. = 36.9%
Our MLP = 63%

Comparing NN

We got better results because the dataset used by them was insufficient to fill up the logarithmic buckets significantly.

As we had much more features to learn from and also our target buckets were logarithmic rather than linear buckets used by them.

Conclusion



0
10
20
30
40
50
60
Accuracy
BENJAMIN ET AL. vs OUR SVM
JASON ET AL. vs OUR LOGREG
RAMESH ET AL. vs OUR NN
Comparing our ML models

Our SVM model gave better results with accuracy of 57.8% compared to Benjamin et al. with 49.4% as we have larger feature set and SVM is effective in high dimensional spaces.

Our Logistic regression performed better with an accuracy of 59.6% compared to Benjamin, Jason et al. having accuracy around 50%. Our feature selection was optimal and dataset was abundant.

We got best results using MLP model with an accuracy of 63% much better than Ramesh et al., because of using more features and using logarithmic buckets to classify.

In future, we can try to include the textual ratings in the features as the reviews given by critics and audience.

We could try to give variable weightage to the genre rather than binary.

THANKS