1. **Introduction**

Since last few decades movies have become one of the best sources of the entertainment. With the advancement of technology, the way the films are produced has also changed. Starting with black and white films in the 90’s to 4D films in today’s world there has been a lot of advancement in the Movie Industry. But even after years of experience, the producers after investing millions of dollars, they still live in stress thinking of whether their movies will we be successful at the box office or not. Therefore, the main aim for our project is to predict whether the film will be successful at the box office or not based on features given below:

1. Number of critics for review - Film critic is a person who writes or publishes a review of a film from either an artistic or entertainment point of view. This predictor tells the number of critics who reviewed the film.
2. Duration - In motion picture terminology, duration is the length of a feature film
3. Director FB likes – The number of likes received by the profile or page of the director on Facebook.
4. Actor FB likes - The number of likes received by the profile or page of the Actor on Facebook.
5. Gross-Grossis the gross box office earnings of a movie in U.S. dollars
6. Number of voted users – The people who watched the movie and voted on IMDb website for the movie
7. Cast total FB likes - The total number of likes received by pages or profiles of each member of the film cast on Facebook.
8. Face number in poster – The number of faces of the actors displayed in the film poster
9. Number of users for review – The number of people who provided the review of the film.
10. The budget of a movie refers to the amount of money spent on the production of the film
11. Aspect ratio - Aspect ratio refers to how the image appears on the screen based on how it was shot–the ratio of width (horizontal or top) to height (vertical or side) of a film frame, image, or screen.
12. Movie FB likes - The number of likes received by the page of the Film on Facebook.

Based on these features, we want to predict the IMDB scores of new movies to be produced.

This model will help the directors to get idea of whether the films they are making will be successful or not and what they can include in the films in order to increase their profits from the movies and help the Movie Industry.

2. **Data Cleaning**

This is the first step in any data analysis project. In this pre-processing step, we analyze and organize data to make good strategic decisions. In our dataset, we have approximately 5000 data points with a few missing values and 28 features of which half were redundant. After cleaning the data: deleting the null cells (since we had the luxury of a big and comprehensive dataset with independent datapoints) and deleting the redundant categorical features (like names of directors, names of actors, movies’ year of release); our dataset comprised of approximately 3800 data points & 14 features. To make sure we did not have any more null values for our dataset, we performed the command isnull() to check for missing values. We found that we no longer had any missing values, so we could proceed forward safely.

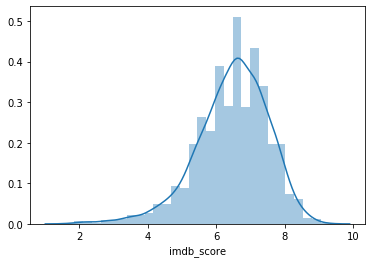
Since IMDb score was initially a quantitative variable so we needed to convert it to categorical variable for logistic regression and neural network classification models. Hence, we decided to divide it into two categories: all IMDb scores of 7 and above were given a value of 1 (movie is supposed to be a hit if it has a score of 7 or above) and those below 7 were given value of 0 (movie is supposed to be a flop if it has a score of below 7).

Since our data was not grouped to summarize a large dataset, we did not perform any bin generation.

We performed standardization of the data, in order to conform one standard throughout the dataset. We used the technique of Min-Max scaler to standardize the features’ dataset.

3. **Preliminary Analysis**

For preliminary analysis, we plotted the distribution of imdb\_score and we observe that the distribution of our response variable, i.e., imdb\_score is fairly normal with a slight shift towards the right from the mean which means the average of the values will be a little towards the right of the normal average.



We see from the scatterplot matrix that the data is fairly distributed throughout all the sets of features and there is random distribution of almost all the data points among various features and the response variable (IMDb score) of the dataset without showing any signs of a very strong linear correlation.

Referring to the correlation matrix, we can infer the following

All the features show small correlation values among themselves. A few of them show some signs of a correlation and those interactions are as presented below:  
- Response-Predictor: Almost all features show signs of weak positive correlation except facenumber\_in\_poster which shows a very weak negative correlation.  
- Predictor-Predictor: All features are weakly correlated, but num\_critic\_for\_reviews and movie\_facebook\_likes (0.7), num\_voted\_users and num\_user\_for\_reviews (0.77) and actor\_1\_facebook\_likes and cast\_total\_facebook\_likes (0.9)shows significant sign of positive correlation.

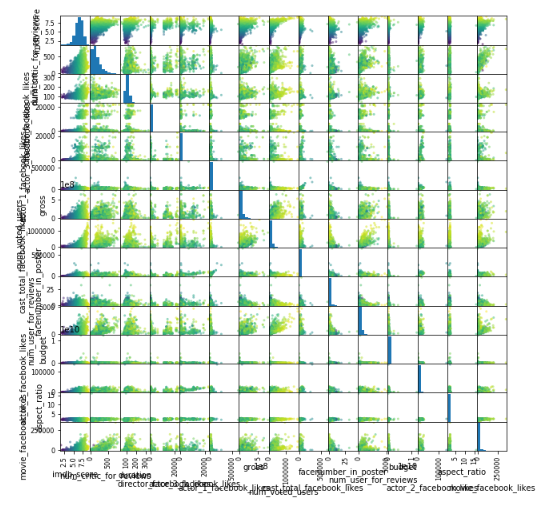
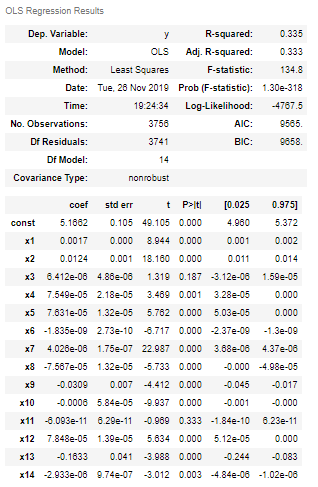
Potential Complications: The potential complications in our project could be:

i) Lack of Diverse Subject Content Experts: We know how to deal with the dataset and gain meaningful insight from them. On the contrary, we have little to no knowledge about the actors, movies, their gross profit and various other dynamics and plays involved in the movie industry.

ii) Using Inferior/Poor Quality Data: If poor quality data is utilized, we won’t get proper insight into the dataset. So in order to predict 100% accuracy of the model, we need to have adequate & acceptable dataset. We have our dataset from a trusted source but we can never be sure about the quality of data until we have performed necessary preprocessing and preliminary analysis on them.

Referring to the ANOVA table for multivariate model below, all the features have p-values less than the alpha (0.10) except director\_facebook likes & budget which have p-values of 0.183 & 0.336 respectively. Thus, we fail to reject null hypothesis for these two variables and conclude that director\_facebook likes & budget don’t have significant interaction with IMDb score.

Also, after checking the p-values of the univariate model, we notice that all p-values are significant (less than alpha=0.10) and thus, all these predictors are significant. From scatter plot matrix, correlation matrix and ANOVA table, we can see that there are no significant multicollinearity issues in our dataset.



4. **Modelling Techniques**

The models selected to fit the data are as follows:

1. Logistic Regression

2. Linear Regression

3. Neural Networks (Regressor and Classifier)

Linear Regression

When we fit this model on the data, we get the training score 0.349 and the test score 0.305. This model explains almost 35% of the variance of the training data set and 30% of the test data set.

Hence, this model is not a good fit for the data set.

When we regularize (alpha = 20) and fit this model on the data, we get the training score of 0.264 and the test score of 0.24. This model explains almost 26% of the variance of the training data set and 24% of the test data set.

We can see from above that after regularization, the accuracy of our model decreases for both testing and training.

Hence, we can conclude that this model is not a good fit for the data set.

When we fit the linear regression model (without regularization), we get the following parameters (coefficients of features) for our feature set:

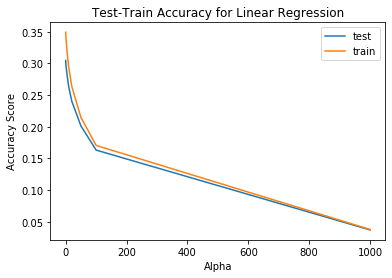
1.4 =num\_critic\_for\_reviews;3.9 = Duration; 0.14 = Director\_fb\_likes; 1.2 = actor\_3\_facebook\_likes; 39.03 = actor\_1\_facebook\_likes; -1.34 = gross; 6.76 = num\_voted\_users; -39.42 = cast\_total\_facebook\_likes; -1.35 = facenumber\_in\_poster; -2.99 = num\_users\_for\_reviews; -0.74 = budget; 8.6 = actor\_2\_facebook\_likes; -5.3 = aspect\_ratio; -0.88 = movie\_facebook\_likes

When we fit the regularized linear model, we selected the hyperparameter (alpha value) as 20 and for optimal hyperparameter selection, we tested for various hyperparameter values and got accuracy as follows for them [Notation: hyperparameter (training accuracy, testing accuracy)]:

0 (0.35, 0.30); 1 (0.34, 0.30); 5 (0.32, 0.28); 10 (0.29, 0.26); 20 (0.26, 0.24); 50 (0.21, 0.20); 100 (0.17, 0.16); 1000 (0.04, 0.04)

When we plot the graph for training and testing accuracy for our data set VS the hyper parameters, we observe that as we increase the hyper parameter values, the accuracy of the testing and training data in the model decreases hence by increasing the hyper parameter values, we the increase bias and decrease the variance and overall accuracy of our model.

Hence, we will choose the hyperparameter with low value such that it has low bias and high variance and accuracy. In this we will take 1 as the optimal hyperparameter.



Mathematical Representation of our Model (without regularization):

Imdb\_score = 5.5 + 1.46 \*(num\_critic\_for\_reviews)+ 3.94 \* (Duration) + 0.14 \* (Director\_fb\_likes) + 1.32 \* (actor\_3\_facebook\_likes) + 41.3 \* (actor\_1\_facebook\_likes) + (-1.34) \* (gross) + 6.82 \* (num\_voted\_users) + (-41.58) \* (cast\_total\_facebook\_likes) + (-1.40) \* (facenumber\_in\_poster) + (-3.13) \* (num\_users\_for\_reviews) + (-0.29) \* budget + 8.89 \* (actor\_2\_facebook\_likes) + (-0.59) \* (aspect\_ratio) + (-0.90) \* (movie\_facebook\_likes)

Neural Net Regressor

When we fit this model on the data, we get the training score 0.47 and the test score 0.40. This model explains almost 47% of the variance of the training data set and 40% of the test data set.

Hence, this model is also not a very good fit for the data set.

When we regularize (alpha = 1) and fit this model on the data, we get the training score of 0.48 and the test score of 0.42. This model explains almost 48% of the variance of the training data set and 42% of the test data set.

We can see from above that after regularization, the accuracy of our model increases for both testing and training as there is a local maximum for accuracy for this alpha value.

Hence, we can conclude that this model is a better fit for the data set when we compared to the normal neural net regressor model without any penalty on the weights. But overall, this model also fails to explain our dataset properly.

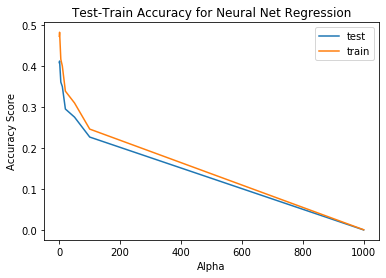
When we fit the neural net regressor model, we get the parameters in the form of 1. weights of the different layers (2 layers for our model) and nodes (100 nodes per layer in our model) of the neural network regressor model; and 2. The final coefficients of features for our feature set.

When we fit the regularized neural net regressor model, we selected various hyperparameter (alpha value) for optimal hyperparameter selection [Notation: hyperparameter (training accuracy, testing accuracy)]:

0 (0.47, 0.40); 1 (0.48, 0.42); 5 (0.42, 0.36); 10 (0.40, 0.35); 20 (0.34, 0.29); 50 (0.31, 0.27); 100 (0.24, 0.23); 1000 (-2.66, 0.00)

When we plot the graph for training and testing accuracy for our data set vs the hyper parameters, we observe that as we increase the hyper parameter values, the accuracy of the testing and training data in the model decreases hence by increasing the hyper parameter values, we the increase bias and decrease the variance and overall accuracy of our model.

Hence, we will choose the hyperparameter with low value such that it has low bias and high variance and accuracy. In this we will take 1 as the optimal hyperparameter.

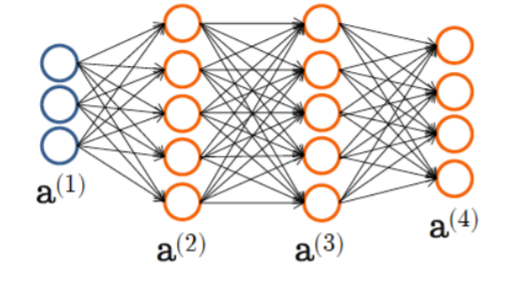


Mathematical Representation of our Model (without regularization):

f(z) = f(b + Ʃxw) (activation function = relu function)

where x = input to neuron; w = weights; b = bias

Since we have two hidden layers with a total of 100 nodes in each layer, our neural network will look something like this:



a(1) = input values of the features for first hidden layer

z(2) = Θ(1)a(1)

a(2) = g(z(2))

z(3) = Θ(2)a(2)

a(3) = g(z(3))

z(4) = Θ(3)a(3)

a(4) = g(z(4)) = Quantitative Output (ŷ)

Logistic Regression

When we fit this model on the data, we get the training score 0.77 and the test score 0.76. This model explains almost 77% of the variance of the training data set and 76% of the test data set.

Hence, this model is a good fit for the data set.

We see a significant increase in our accuracy score largely due to the fact that it is easier for the model to classify and distinguish features into two categories rather that predict a different and particular quantitative score for different feature set. So, to predict a movies’ success, we will select a classification model rather than regression model as this model gives comparatively higher accuracy than the regression models.

We can see from the confusion matrix of logistic classifier that we have a decent average F-1 score and accuracy of 0.76. So, we can say that non-regularized logistic regression does a good job to explain our dataset.

When we regularize (alpha = 20) and fit this model on the data, we get the training score of 0.205 and the test score of 0.175. This model explains almost 20% of the variance of the training data set and 17% of the test data set which is a significant drop from the non-regularized model’s accuracy of 77% and 76% respectively.

We can see from above that after regularization, the accuracy of our model decreases significantly for both testing and training.

Hence, we can conclude that this model is not a good fit for the data set.

When we fit the logistic regression model (without regularization), we get the following parameters (coefficients of features) for our feature set:

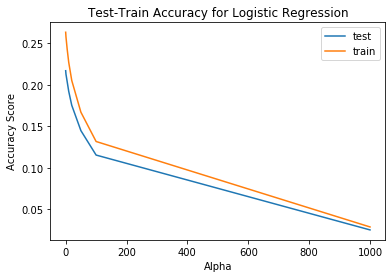
1.7 =num\_critic\_for\_reviews;5.2 = Duration; 1.03 = Director\_fb\_likes; -0.85 = actor\_3\_facebook\_likes; 0.22 = actor\_1\_facebook\_likes; -2.6 = gross; 8.24 = num\_voted\_users; -0.06 = cast\_total\_facebook\_likes; -1.77 = facenumber\_in\_poster; 0.7 = num\_users\_for\_reviews; -0.5 = budget; -0.4 = actor\_2\_facebook\_likes; -1.7 = aspect\_ratio; 1.91 = movie\_facebook\_likes

When we fit the regularized linear model, we selected the hyperparameter (alpha value) as 20 and for optimal hyperparameter selection, we tested for various hyperparameter values and got accuracy as follows for them [Notation: hyperparameter (training accuracy, testing accuracy)]:

0 (0.26, 0.21); 1 (0.257, 0.21); 5 (0.24, 0.20); 10 (0.23, 0.19); 20 (0.20, 0.17); 50 (0.17, 0.14); 100 (0.13, 0.11); 1000 (0.03, 0.02)

When we plot the graph for training and testing accuracy for our data set vs the hyper parameters, we observe that as we increase the hyper parameter values, the accuracy of the testing and training data in the model decreases hence by increasing the hyper parameter values, we the increase bias and decrease the variance and overall accuracy of our model.

Hence, we will choose the hyperparameter with low value such that it has low bias and high variance and accuracy. In this we will take 1 as the optimal hyperparameter.



Mathematical Representation of our Model (without regularization):

Imdb\_score = 1/[1+exp{-(-1.94 + 1.77 \*(num\_critic\_for\_reviews)+ 5.17 \* (Duration) + 1.03 \* (Director\_fb\_likes) + (-0.85) \* (actor\_3\_facebook\_likes) + 0.22 \* (actor\_1\_facebook\_likes) + (-2.61) \* (gross) + 8.24 \* (num\_voted\_users) + (-0.06) \* (cast\_total\_facebook\_likes) + (-1.77) \* (facenumber\_in\_poster) + 0.68 \* (num\_users\_for\_reviews) + (-0.49) \* budget + (-0.39) \* (actor\_2\_facebook\_likes) + (-1.70) \* (aspect\_ratio) + (-1.91) \* (movie\_facebook\_likes))}]

Neural Net Classifier

When we fit this model on the data, we get the training score 0.82 and the test score 0.77. This model explains almost 82% of the variance of the training data set and 77% of the test data set.

Hence, this model is a very good fit for the data set.

When we regularize (alpha = 5) and fit this model on the data, we get the training score of 0.81 and the test score of 0.77. This model explains almost 81% of the variance of the training data set and 77% of the test data set.

We can see from above that after regularization, the accuracy of our model only slightly decreases for testing and training.

We can see from the confusion matrix of logistic classifier that we have a decent average F-1 score and accuracy of 0.77. So we can say that non-regularized logistic regression does a good job to explain our dataset.

Hence, we can conclude that this model a good fit for the data set.

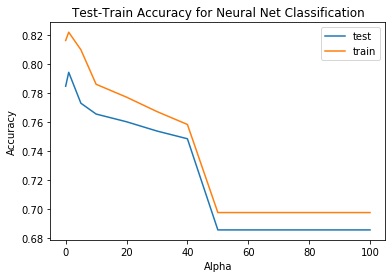
When we fit the neural net regressor model, we get the parameters in the form of 1. weights of the different layers (2 layers for our model) and nodes (100 nodes per layer in our model) of the neural network regressor model; and 2. The final coefficients of features for our feature set.

When we fit the regularized linear model, we selected various hyperparameter (alpha value) for optimal hyperparameter selection [Notation: hyperparameter (training accuracy, testing accuracy)]:

0 (0.82, 0.77); 1 (0.81, 0.79); 5 (0.80, 0.78); 10 (0.79, 0.76); 20 (0.78, 0.76); 30 (0.77, 0.75); 40 (0.76, 0.75); 50 (0.70, 0.68); 60 (0.70, 0.68); 70 (0.70, 0.68); 80 (0.70, 0.68); 90 (0.70, 0.68); 100 (0.70, 0.68)

When we plot the graph for training and testing accuracy for our data set vs the hyper parameters, we observe that as we increase the hyper parameter values, the accuracy of the testing and training data in the model first decreases and then reaches a constant value after alpha value of around 50. Hence by increasing the hyper parameter values, we the increase bias and decrease the variance and accuracy of our model.

Hence, we will choose the hyperparameter with low value such that it has low bias and high variance and accuracy. In this we will take 1 as the optimal hyperparameter as this has the highest accuracy for both test and train due to occurrence of a local maxima in the graph.

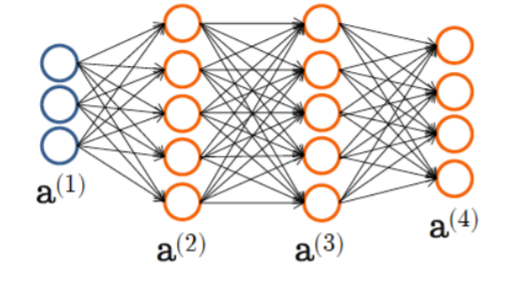


Mathematical Representation of our Model (without regularization):

f(z) = f(b + Ʃxw) (activation function = relu function)

where x = input to neuron; w = weights; b = bias

Since we have two hidden layers with a total of 100 nodes in each layer, our neural network will look something like this:



a(1) = input values of the features for first hidden layer

z(2) = Θ(1)a(1)

a(2) = g(z(2))

z(3) = Θ(2)a(2)

a(3) = g(z(3))

z(4) = Θ(3)a(3)

a(4) = g(z(4)) = Categorical Output (ŷ)

5. **Model Selection**

After adding a polynomial feature of deg = 3, we obtained a negative R-square value which tells us that adding a polynomial deg = 3 doesn’t do a good job explaining the fit of our dataset rather the mean line explains our dataset in a better way in this case. Hence, we may decrease the bias for our model, but we are increasing the variance and decreasing the accuracy of our overall model by adding polynomial features of deg = 3.

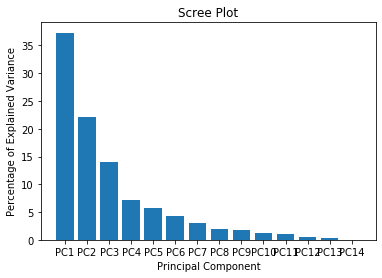
After performing regularization for all the fitted models (linear regression, logistic regression, neural net regression and classifier) we found that the two potentially best models for our dataset are Neural Network-Classification & Neural Network- Regression as we have observed the highest training and testing accuracy scores for these two models. Among these two models we found that, neural network classification has a higher accuracy mean cross validation score and hence that’s the best model.

In our dataset, as the alpha value increases, cross validation score decreases which is not desirable. So small hyperparameter values would be preferable here which is either 0 or 1. We chose an alpha value of 1, where we get a mean cross validation score of 0.787.

The best overall model for our dataset is regularized neural network-classification because it gives a better cross validation scores as compared to all other previously fitted models and also has a better accuracy for training & testing data. The optimal hyperparameter selected for this model is taken as 1 as this gives a good accuracy along with a good cross validation score.

6. **Model Validation**

For dimension reduction, we applied principle component analysis for our model. When we applied PCA we observed from the Scree Plot that almost 37% of our explained variance of our feature data was explained by the PC1 component, 22% of the explained variance was explained by the PC2 component, 14% of the explained variance was explained by the PC3 component, 7% by PC4, 5% by PC5 and 4% by PC6. So, to fit our final model, we consider only 6 principle components to be significant and hence, the reduced feature set is concatenated into 6 principle components. These 6 principle components together explain about 90% of the explained variance in our feature dataset.



When we fit these reduced feature set into our previously obtained best model (neural net classification) with optimal hyperparameter (alpha) of 1 and observe the accuracy of 77% for our training data and approximately 75% accuracy for our testing data.

7. **Conclusion**

For our best model, we consider the neural net classification model with optimal hyperparameter of 1 and with a reduced feature set as even with reduced accuracy than the original model (without applying PCA) from 82% & 79% for training and testing respectively to 77% & 75% for training and testing respectively we are still able to capture majority of the explained variance of our dataset and through this reduced and significant features we have a model with accuracy close to our original model.