**Data Collection:**

First we had to get a list of all the movies released from 2011 to 2015. So,we generated a query in the "cinemalytics.com" api to get all the available data of the movies for the required time period in JSON format. Then we extracted the movie title and IMDbId. We used this list of IMDbId's to make successive requests in the "omdbapi.com" to get some more data of these movies. Then we filtered the movies on the basis of their budget,Origin (Indian) and movie length (short films were excluded). This was done because data for short films was inadequate and movies with very low budget would lead to huge variance and affect model formation. In the end we were left with a dataset of 150 Bollywood movies.

Then we started collecting the Youtube data (trailer views,likes,dislikes). We extracted the trailer links from the dataset and successively downloaded the youtube pages. From these pages we extracted the number of views,likes and dislikes and fed them into the dataset.

**Feature Modification:**

We had to convert all the features into numerical form so that it could be used to train the model. We started of with the actor and director rating. We got the rating, on a scale of 5 , of all the actors and directors in our database from reliable sources like IMDb itself. Then the highest rating among the actors in a movie was considered as the actor rating of that movie. Simmilarly for director rating. Then the Budget Feature of the movie was taken as the Budget (in INR) divided by 100 million. For Movie Length Feature the movie's length in minutes was considered. Youtube views were taken as it is. For Genre, total of 18 genres in the database were labelled from 1 to 18. After some graphical analysis the genres were labelled agained such that a genre with higher label showed greater IMDb rating in general ( like Biography was labelled 15). Then for a movie the highest genre label was considered as its Genre Feature.

**Feature Analysis:**

We had to see how the individual features were related to the IMDb rating of the movie. So, we plotted graphs of IMDb Rating vs Feature to get an idea about the relations or patterns. This was done using python and its libraries scipy and matplotlib. As a result of this graphical analysis we found that Director Rating showed proportional behaviour with IMDb rating. Actor Rating too showed somewhat proportional behaviour. Genre feature too showed a pattern for genres labelled 10 or more. However, for genres labelled less than 9 there was no such pattern observed. Other features too showed disappointing results as there was no such pattern or fixed behaviour observed.

\*\*\*\*\*\*\*\*\*\*\*\*\*images\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Polynomial Regression:**

After studying the graphs we realized that we could find a polynomial curve to fit the data. This curve gave more accurate results for Actor Rating Feature and Director Rating Feature. Also, for Genre label from 10 to 18 the polynomial fit the data quite well. Hence, it was decided to form a simple mathematical model combining the three feature (Actor Rating and Director Rating and Genre label) polynomials. The dataset was divided into a training dataset of 120 movies and testing dataset of 30 movies.

Fitting Polynomials:

Director Rating Feature: R(d)=0.535\*d^2 - 2.939\*d + 8.21

Actor Rating Feature: R(a)=0.495\*a^2 - 2.547\*a + 8.705

Genre Label Feature: R(g)=-0.015\*g^2 + 0.585\*g + 1.703 (for g>9)

Proposed Model:

R(d,a,g)=M1\*(0.535\*d^2 - 2.939\*d + 8.21) + M2\*(0.495\*a^2 - 2.547\*a + 8.705) + M3\*(-0.015\*g^2+0.585\*g+1.703) (for g>9)

R(d,a)=M4\*(0.535\*d^2 - 2.939\*d + 8.21) + M5\*(0.495\*a^2 - 2.547\*a + 8.705) (for g<=9)

d=Director Rating

a=Actor Rating

g=Genre Label (1 to 18)

Mi=Parameters

M1+M2+M3=1 & M4+M5=1

R=ImdbRating

From Experimantal analysis:-

M1=0.1

M2=0.2

M3=0.7

M4=0.3

M5=0.7

**Fuzzy Logic System:**

We decided to use different algorithms to design the model. As a result we tried using Fuzzy Logic Systems. For this algorithm also we used only three features (Actor Rating and Director Rating and Genre label), since we were aware of their behaviour against the IMDb rating.This algorithm was implemented using 'skfuzzy' python library. The following Membership Functions, finalized after testing on the dataset, were used.

\*\*\*\*\*\*\*\*\*\*\*\*\*images\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The Fuzzy Rules were initially formed on the basis of intelligent guessing. Later on after lots of experiments, keeping in mind to reduce the maximum and average error, these rules were changed to get the best results.

**SVR (Support Vector Regresssion):**

This was another algorithm used to see if it could get us a better model with less error. All the features which we decided upon earlier (Actor and Director Rating,Budget,Movie Length,Genre,Youtube Views) were used for training this algorithm. The python library used for its implementation was 'sklearn'.

**Neural Networks:**

Neural Networks is probably the most widely used and preffered Machine Learning algorithm. Hence, we decided to use this algorithm. Once again all the features (Actor and Director Rating,Budget,Movie Length,Genre,Youtube Views) were used to train this algorithm. Using the python library 'pybrain' we formed a network of 3 layers:

Input Layer of 6 nodes

Hidden Layer of 6 nodes

Output layer of 1 node

The dataset was divided in a ratio of 1:5 for testing and training respectively. The network was trained using a Backpropagation trainer for a few epochs(cycles) aiming for convergence. The testing was done a few times to get the best results.

**Results and Error Analysis:**

Polynomial Regression:

An average error of 1.5 was recorded when three features namely Actor,Director Rating and Genre was considered for predicting the values. A maximum error of 3.5 was observed. The average error is not too good so that we can consider this model fullproof, still it can give us close values. The main problem though is a maximum error of 3.5 which means this model cannot be completed trusted. This is a result of the fact that even though a movie having excellent star cast and top director it may fail to impress the audience considering the storyline,screenplay etc.

Neural Networks:

Considered to be the most widely used and preffered machine learning algorithm, Neural Networks gave us the best results. All the 6 features namely star cast,director,movie runtime,budget,youtube trailer views,genre were used as the input to the model. This model considering 6 nodes in input layer,6 nodes in hidden layer and 1 node in output layer gave us an average error of 0.87. This is quite better than the previous model, making it more apt for all cases. The maximum error problem still persists though, the reasons being the same.

Fuzzy Logic System:

This algorithm gave us an average error of 1.58, which is poor compared to the other two algorithms we used. Also this algorithm gave a maximum error of 4.5 which is quite vague. Hence, this model is completely unsuitable and was rejected.

SVR (Support Vector Regression):

This model gave us an average error of 1.23, which is better than the Polynomial Regression model. However, this model gave a maximum error of 4 on the test data. This makes this model a little less dependable. Though in many cases it gave a minimal error this model cannot be used for prediction since it gave few large (>2) errors as well.

**Conclusion:**

Various Machine Learning algorithms were used to combine the featues and predict the IMDb rating of Bollywood movies. The 4 algorithms which were used are Polynomial Regression, Neural Networks, Fuzzy Logic, SVMs. Minimizing the the average and maximum error was the main objective while using different algorithms.

Finally it can be concluded that Neural Networks provided the best prediction model for this set of data and feature set. Although there are some irregularities which give rise to high error for some samples. This is probably due lack of features like storyline or maybe the movie was completely different than what even the public expected it to be. From these experiments it is clear that for some movies the IMDb rating can be predicted quite accurately while for others it can be quite vague. The best way though is using the Neural Networks model.

**Future Work:**

More features can be added like writers,public/critics reviews available pre-release,combination of features such as actor and genre (few actors are quite successful in a particular genre) etc. This way a more dependable model may be generated which gives little or no error for all the movies.