IMDB MOVIE RATING PREDICTION SYSTEM

PROJECT FOR COURSE DS-GA-1001:

INTRODUCTION TO DATA SCIENCE

PROFESSOR: Brian d’Alessandro

PRESENTED BY:

Shixin Li

Kashyap Uppuluri

Li-Hao Liu

Rohit Wankhede

1. BUSINESS UNDERSTANDING:

Movies have always formed an integral part of recreation and culture amongst human beings. The worldwide market for movies is huge and is a major source of revenue and income as well. According to recent estimates, the global box office returns for movies released all over the world is around $ 39.4 billion. Close to 5000 movies were released worldwide. Most movies have a relatively high value of production cost; hence it is imperative to come up with ways and means to maximize the revenue for every movie released. One way of doing this is to try and gauge public sentiment about a movie before its actual release. This sort of feedback can help producers to get a sense of the kind of response their movie will get. Before the advent of the Information age this sort of feedback was gained through conventional wisdom and empirical results. In today's times there is plenty of scope for the use of data science to solve such problems.

This paper details the predictive analysis on the IMDb data, which contains records of movies from all over the world along with information associated with every movie such as title, genre, gross income, language, country of origin. Based on votes provided by users, a rating is assigned to every movie.

The goal of this paper is to identify the correlation between the various attributes of a movie with the movie rating. For example, is there any correlation between the budget of a movie with its rating i.e. does a high budget movie have a higher rating than a low budget movie. Also whether movies from a particular genre are more popular than others. This can then be used to predict the rating of a movie, if it were to be listed on the IMDB website.

2. IMDb:

Launched in 1990, as of November 2015 IMDB has approximately 3.5 million titles (including episodes) and 6.8 million personalities in its database. There are close to 64 million registered users on IMDB. IMDB registered users can rate every movie in the website (on a scale of 1 to 10). Users can rate a movie how many ever times they want, but each rating will overwrite the previous one.

3. DATA UNDERSTANDING:

The data for analysis has been taken from the IMDb website. The data was distributed over 49 text files. One of the primary data engineering tasks was to extract relevant data from the files and collate it onto a single dataframe.

The website [1] consists of files that contain the data pertaining to the movies on IMDb. These files are stored in plain text format with a '.list' extension. Each file contains data associated with different facets associated with a movie. These are also highly interrelated with one file referencing the other file for certain information. The files contain the data about: actor, actresses, complete cast and crew, companies, genre, keywords, language, locations, runtimes, release date among many more attributes associated with filmmaking apart from the no. of votes cast and the rating assigned to the movie.

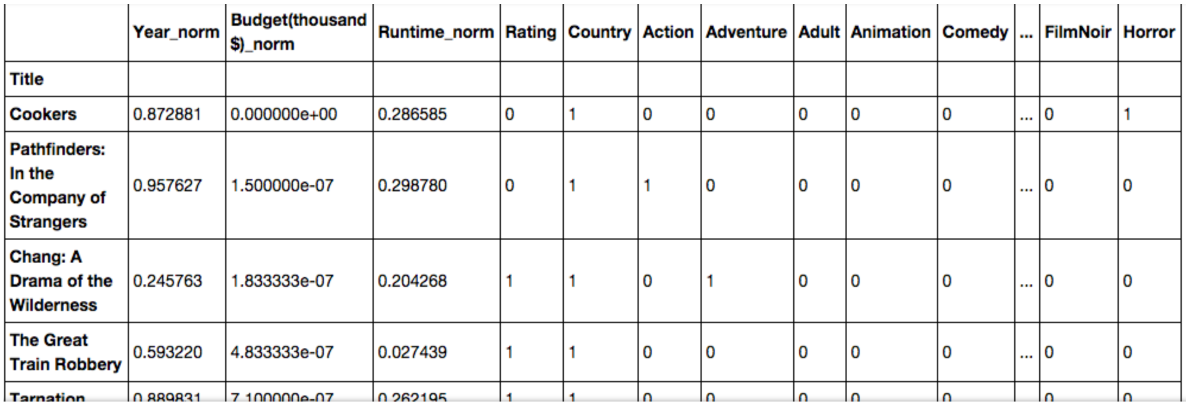
4. DATA PREPARATION:

The process begins with the study of each of these attributes and identification of the attributes that work best for our analysis. Many of the attributes like aka-names, aka-titles, file sizes, quotes among more were dropped on account of not adding any value to the analysis that we plan to undertake. After substantive research on their relevance and feasibility of implementation we decided to use the following features: Year of release, Country the movie is based, Runtime, Genre, and Budget involved.

The dependent variable is the rating of the movie as featured on IMDb website. We are converting into a binary variable by factoring it on the basis of 'Rating 7 or higher'.

These independent features help in explaining how the tastes of the customers have changed over the years, help calculate the net present value of money from the years past, movies from which industry (Country) score a higher rating, and how the amount of budget spent helps in shaping user perceptions and impact the rating.

Substantive data modeling has been performed to extract the relevant data from the files and into a dataframe. In order to cater specifically to the American market, the movies from other Countries have been marked International and budgets converted to a dollar value. Furthermore, these have been converted to binary indicating ‘1’ if movie is based in USA and a ‘0’ if International. The values of ‘Budget’ for movies from different countries were each in different currencies. In order to standardize the results, these values were placed in thousands of USD ($)[2]. For the Release year, as the values for the rest of the columns such as genre and country are binary, the data was normalized such that its value lies between 0-1.A larger value indicates that the movie is more new whilst, a smaller movie indicates that the movie is relatively older. We have created dummy variables in order to accommodate the values from ‘Genre’. All values have been normalized in order to get a stable convergence and distance parameter when using a distance function in k-Nearest Neighbors. The final dataframe consists of around 33,000 rows along with their respective features.



	Year_norm	Budget(thousand $)_norm	Runtime_norm	Rating	Country	Action	Adventure	Adult	Animation	Comedy	...	FilmNoir	Horror
Title													
Cookers	0.872881	0.000000e+00	0.286585	0	1	0	0	0	0	0	...	0	1
Pathfinders: In the Company of Strangers	0.957627	1.500000e-07	0.298780	0	1	1	0	0	0	0	...	0	0
Chang: A Drama of the Wilderness	0.245763	1.833333e-07	0.204268	1	1	0	1	0	0	0	...	0	0
The Great Train Robbery	0.593220	4.833333e-07	0.027439	1	1	0	0	0	0	0	...	0	0
Tarnation	0.889831	7.100000e-07	0.262195	1	1	0	0	0	0	0		0	0

Fig [1]: A snapshot of the final dataframe

5. FEATURE IMPORTANCE:

In order to get a sense of the importance of features that regard to predicting the rating of a movie, we can plot a graph for feature importance. The feature importance are assigned on the basis that which features increase entropy to the maximum and contribute to maximizing information content.

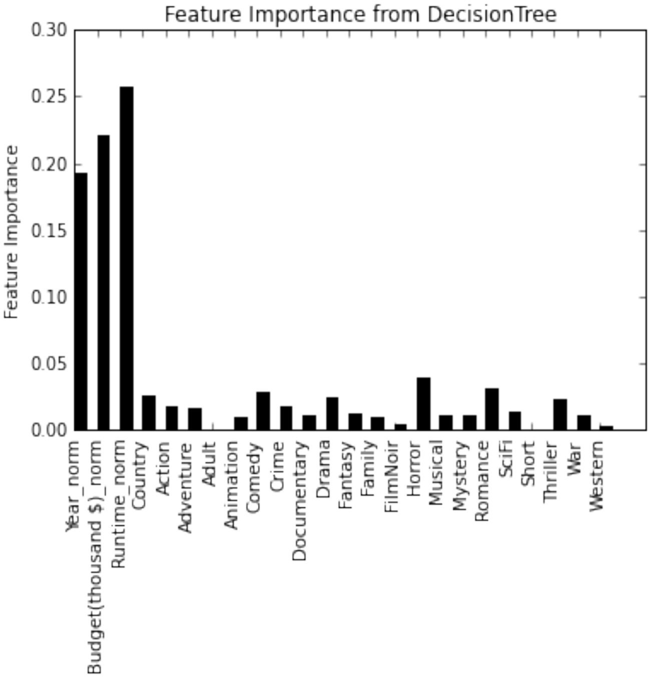


Fig [2]: Graph explaining Feature Importance

6. MODELING:

The dataframe is split into two parts. The first part contains all movies except movies in 2015, and is used for modeling. Another part only contains movies that released in 2015, and is used for further evaluation. The following models are implemented and compared on the basis of various metrics discussed in the further sections:

6.1 DECISION TREE CLASSIFIER:

Decision tree learning uses a decision tree as its predictive model, where it focuses on identifying an item's target value based on its features. It focuses on splitting the parent node into the purest possible child nodes. It uses the concept of entropy and maximizing information gain in order to achieve this. The complexity of a decision tree is dependent on the depth of the tree, the size of the internal node that can be split and the minimum number of leaf instances that can be in a leaf.

6.2 SUPPORT VECTOR MACHINES:

Support Vector Machines (SVM) are supervised learning models used in classification and regression analysis. They perform classification tasks by constructing hyper planes in a multidimensional space that separates classes of different labels. The SVM focuses on choosing a hyper plane that is maximally far from as many points as possible. SVMs have high geometric interpretation and are easily extendable to nonlinear decision surfaces.

6.3 NAIVE BAYES CLASSIFIER:

Naive Bayes Classifiers work on classification tasks by applying Bayes theorem with strong independence assumptions between the features rather than the actual distribution of the features. It works by assigning an item to a class, based on the class that returns the highest probability with regards to the state of the item. Based on the assumption regarding the distribution of the features, we have Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. In this paper, Bernoulli Naive Bayes has been implemented as we are dealing with binary data [3].

6.4 K–NEAREST NEIGHBORS:

K-Nearest Neighbors is an algorithm used for classification in which the input consists of the k closest training examples in the feature space. In kNN classification, an object is assigned to the class most common amongst its k nearest neighbors. If k = 1, the object is simply assigned to the class of its nearest neighbor. It is useful to assign weights to the neighbors, so that near neighbors have a greater effect on the final classification than distant neighbors. Choosing an optimal value for k is a critical component of this algorithm. The optimal value for k depends on the data, generally larger values of k reduce the effect of noise but make the boundaries between classes less distinct and vice versa for smaller values for binary classification problems, it is advisable to select an odd value for k to avoid the case of a tied scenario.

6.5 LOGISTIC REGRESSION:

Logistic Regression works on estimating the probability that an object will lie in a particular class using an inverse logit function as its model. Logistic Regression has strong statistical properties and is designed to calculate the conditional probability that an object will belong to a particular class. They calculate the weights of the features that generalize the model to generate posteriori probability values for an object to be classified.

Results obtained by applying the above models to the prediction task at hand. The ROC curve was generated for every model as it gives a comprehensive summary of how well a classifier works. The Area under the curve (AUC) is indicative of the efficiency of a particular classification.

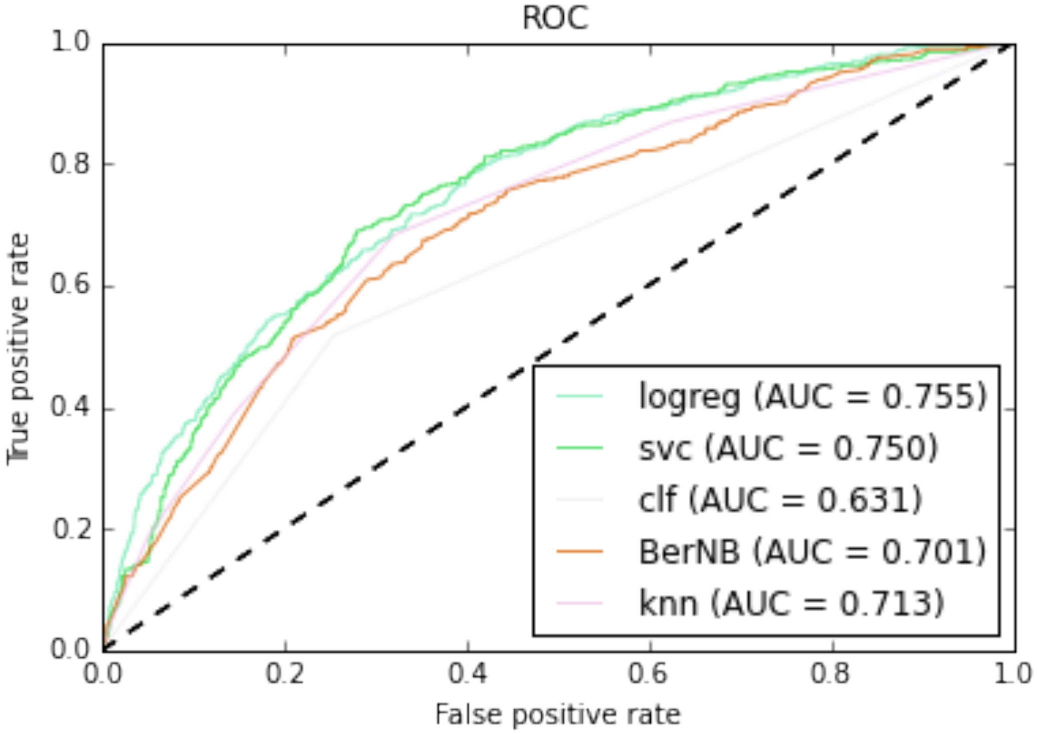


Fig [3]: ROC Curve for all five algorithms

7. EVALUATION:

Based on the default Hyperparameters for every model, it is observed that the Logistic Regression returns the highest value for the AUC. But this cannot be the only basis on which to choose or reject a particular algorithm. We have to vary the Hyperparameters for every model and calculate the AUC for every combination, as there is a possibility that a larger value for the AUC can be obtained. The dataset is divided into a training set and a validation set, once the Hyperparameters for a model are decided; the model is run on the test set. This is performed using Cross Validation.

7.1 DECISION TREE CLASSIFIER:

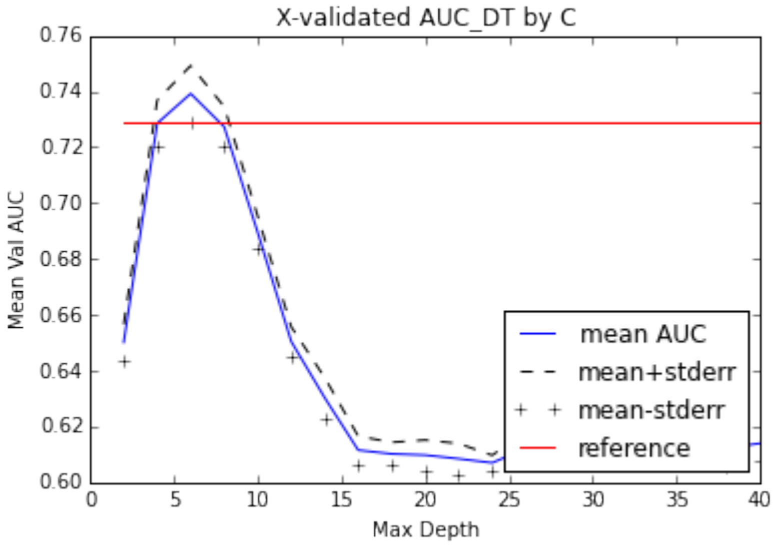


Fig [4]: X-Validated AUC for Decision Tree Classifier by Hyperparameter Max Depth

For Decision Tree, the max_depth parameter was adjusted. As we can see, the AUC improves at first and then decreases, as the Hyperparameters are varied but there is an added level of complexity with each level as well. A maximum AUC is reached when the max depth is set at 6.

7.2 SUPPORT VECTOR MACHINES:

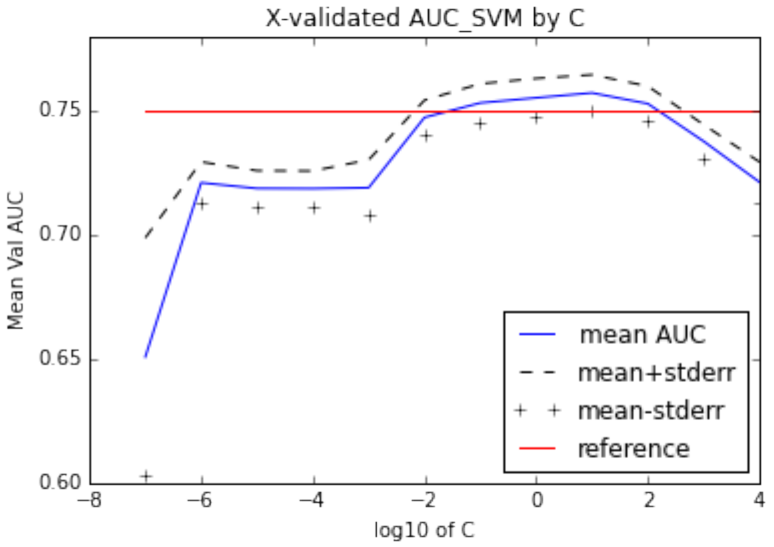


Fig [5]: X-Validated AUC for Support Vector Machines by Hyperparameter C

In the case of SVM, the penalty parameter of the error term (C) is varied and the AUC is calculated at every point. K-fold cross validation with 6 folds is performed in this case.

7.3 NAIVE BAYES CLASSIFIER:

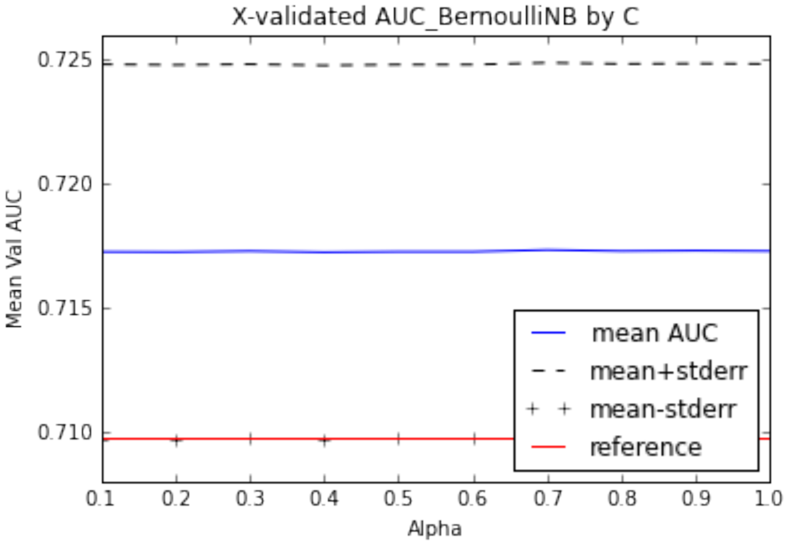


Fig [6]: X-Validated AUC for Naive Bayes Classifier by Hyperparameter Alpha

For the Naive Bayes Classifier, the Hyperparameter alpha is varied from 0.1 to 1.0, to observe the effects on the efficiency of the classifier. It is observed that the AUC remains constant for the Naive Bayes Classifier even as alpha is varied.

7.4 K-NEAREST NEIGHBORS:

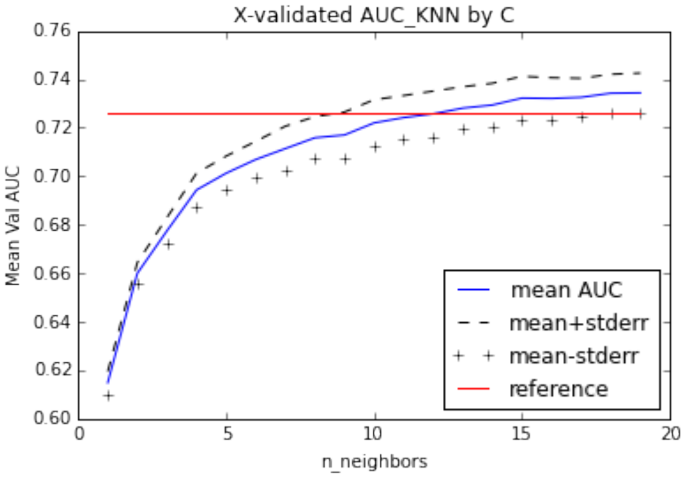


Fig [7]: X-Validated AUC K - nearest neighbors by Hyperparameter k

In the case of K - Nearest Neighbors, the most crucial component is the no of neighbors that contribute towards making a decision. As the no of neighbors is increased the AUC increases and tends to flatten out after a certain stage.

7.5 LOGISTIC REGRESSION:

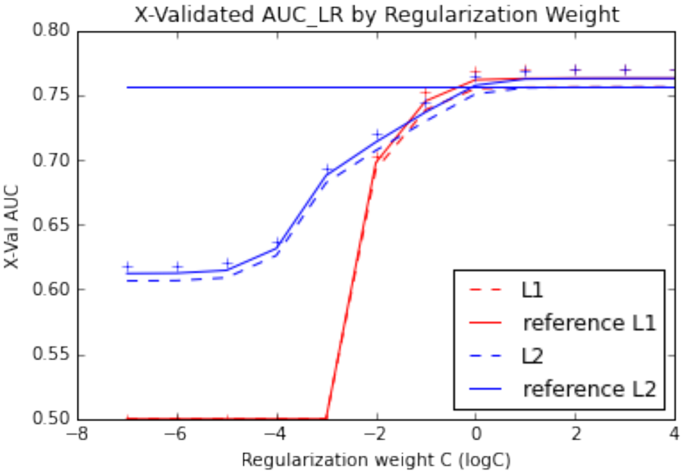


Fig [8]: X-Validated AUC for logistic regression by Hyperparameter C

For Logistic Regression, the regularization weight parameter C is varied, and the AUC is calculated at every point. It is observed that the AUC increases at first, reaches a maximum threshold value and then tends to flatten out and remain constant.

As mentioned in prior sections, we evaluate the models one final time by predicting the outcomes on the movies released in 2015; the models have not been exposed to this data earlier. This final assessment can help us simulate the behavior in a production environment and help assess the performance of each model. The results were:

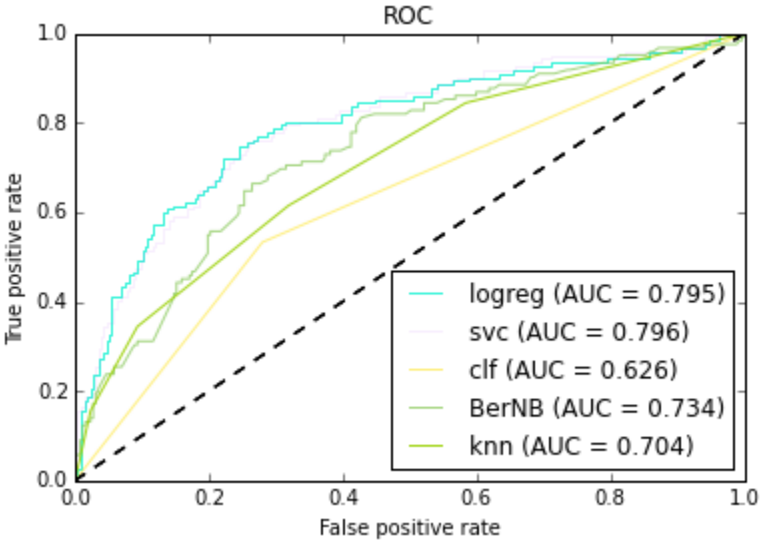


Fig [9] ROC Curve for the data on 2015 Movies

8. DEPLOYMENT:

The current model can be deployed as a web page that automatically populates the independent variables based on the 'Title' and outputs a rating of whether the upcoming movie will have a rating of 7 or higher. We have chosen to use the Logistic Regression for our model as it's fairly easy to implement and highly scalable. Additionally, it is good at dealing with correlated variables and provides good probabilistic implementation that makes it a good first step for this modeling.

9. REFERENCES:

[1] ftp://ftp.fu-berlin.de/pub/misc/movies/database/

[2] https://docs.python.org/2/library/locale.html

[3] http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html