PREDICTING MOVIE RATINGS AND VIEWER'S SENTIMENT ANALYSIS BASED ON MOVIELENS DATASET

Manisha Chandran, Shwetha Panampilly, Tanmay Sawaji and Vishwas Desai

Abstract

Movie rating is an important element, it is like a summary that reflects the quality of all elements concerning a movie. Our project uses the movielens dataset that contains key attributes like movie titles, genres, ratings, IMDb ID. Our primary goal is to predict ratings of a movie based on key attributes such as box office, budget, and genres and additional attributes like actors will be retrieved by making use of the IMDb API. The secondary goal is to binarize the reviews attribute by applying sentiment analysis which is then trained on a different dataset containing reviews and sentiment. To achieve the secondary goal Naive Bayes classifier algorithm was implemented and for the primary goal, multiple models were tested. Through this project, we intend to predict the performance of the movie and the viewer's sentiment.

Index Terms *— Movielens, IMDb, movies, ratings, regression, deep neural networks*

1. INTRODUCTION

There is no universally accepted way to gauge a movie is good or bad even after its theatrical release. There will be some movies which will have a little to no difference between the critics and the viewers opinion of the movie but some movies will have a stark difference in opinions. Through this project, we aim to gauge or predict a movie's performance via giving out a rating based on the sentiment analysis of the viewer's review.

The task of this project was to a) Binarize the reviews attribute by applying sentiment analysis and b) Predict ratings of a movie based on the Movielens dataset [5]. The motivation behind this project is twofold. Firstly, to create a prototype rating model for online content streaming services. Secondly, this model can be used by viewers and content creators alike to curate movies for their own collection and for consumption respectively.

The project involved implementation of a naive bayesian classifier, a deep neural network, Support vector Regressor and Random Forest Regressor of which the latter models were used to train and test our dataset.

2. PREVIOUS WORK

There have been many papers on this topic using other technologies. The following are the many approaches used for prediction of movie rating

1. The prediction is done by using two regression models: Kernel regression and linear tree [1]

2. Making predictions by using historical values or prerelease elements such as genres, actors, directors etc. [2]

3. The third approach is using an IMDB dataset. Here the data is extracted from data mining algorithms.It is performed in an WEKA environment and confusion matrices are generated [3]

The common drawback in each of the approaches is the dataset and the feature selection in each of the dataset.In our project we address this issue by using multiple datasets.

3. DATA PREPARATION

This study used datasets from multiple sources. The datasets were in comma separated value (CSV) format. The MovieLens dataset consisted of 3 different csv files that were merged by matching the movieId column. Additional attributes were added by writing API scripts to fetch data from IMDb databases. After the dataset was formed, pre-processing such as data cleaning, normalization and encoding was performed. Performing such processing on a dataset will help increase the quality of the dataset, which will improve the accuracy of movie rating prediction done by the ML algorithm. The IMDb 50K reviews dataset contained 50,000 rows and two attributes: reviews and sentiment. The sentiment attribute was binary, it just included 'positive' and 'negative' labels. This dataset did not require any preprocessing, instead the Naive Bayes classifier was written such that it will clean the input data.

3.1 Features

Features are attributes associated with the success of a movie. Multiple features are numerical features, categorical features, and topical features.

3.1.1 Metadata features

Metadata features describe the numerical information about the movie. There were multiple ratings present for the same movie, so we took the average of all the ratings for processing

Feature	Description
(rating, Average Rating)	Rating(or average rating) for a specific movie
budget	Amount taken to make the movie
box_office	Collection for that movie

Table 1: Metadata features

3.1.2 Categorical Features

The text attributes associated with a movie are considered as categorical features. These categorical features consist of categorical attributes. Since genres, actors and directors are text data, transforming them is needed to get value for them.

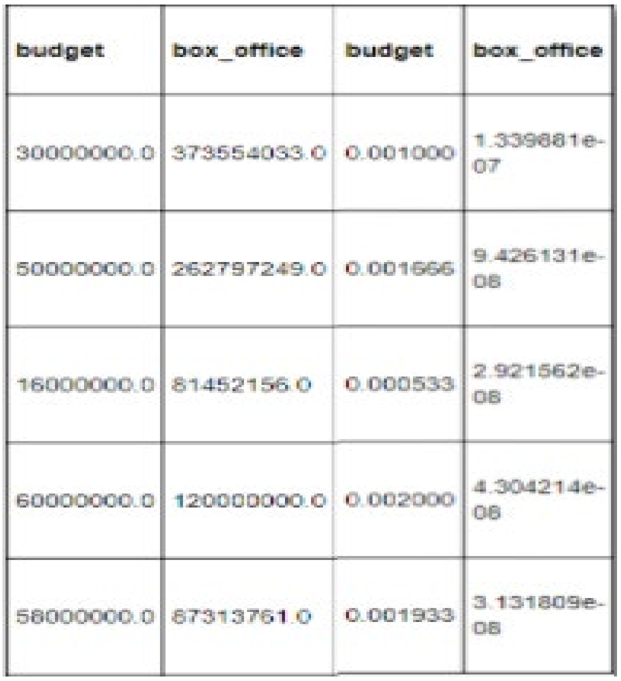
Feature	Description
genres	Describes the genre movie belongs to
director	Person who directed the movie
actors	People who acted in the movie

Table 2: Categorical Data

3.2 Min-Max Scaler

All the numerical data have different scales, hence we need to normalize it so that the algorithm processes them easily. The new normalized data will have values in the same scale that is between 0 and 1. The general formula for a min-max of [0, 1] is given as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$



budget	box_office	budget	box_office
30000000.0	373554033.0	0.001000	1.339881e-07
50000000.0	262797249.0	0.001666	9.426131e-08
16000000.0	81452156.0	0.000533	2.921562e-08
60000000.0	120000000.0	0.002000	4.304214e-08
58000000.0	67313761.0	0.001933	3.131809e-08

Figure 1: Example of normalization

The figure shows the attribute values before and after normalized scaling is applied.

4. NAIVE BAYES CLASSIFIER

4.1 Bag-of-Words Model

A bag-of-words model is a statistical framework used for many machine learning models. A bag-of-words model treats a sentence or any data as a collection of smaller units of data, where the smaller units of data are conditionally independent of each other. In this project, the smaller unit of data is words. The bag-of-words model assumes that the position of a word in the sentence does not affect the sentiment of the sentence; it assumes that the occurrences of the words decide the class of the sentiment. Our naive assumption is that all words in an example are conditionally independent of each other. This assumption, in general, disproves the need for grammar and semantics in a language, which we know to be false; yet this approach works well in practice when the dataset is large enough.

4.2 Cleaning

The dataset chosen for this model is the Imdb 50K reviews dataset, which includes 50,000 reviews of various movies and a ‘positive’ or ‘negative’ sentiment for each review. The Naive Bayes classifier takes every entry as input and removes all punctuation marks and numerical characters from the dataset. All examples are converted to lowercase so that “It” and “it”, for example, are considered to be the same word. Stopwords are words in a language that occur very frequently. For example, for the English language, ‘a’, ‘the’, ‘and’ can be considered as stopwords. Elimination of these words shows an improvement in performance of the model as shown in table.

Model	Accuracy	Precision	Recall
Original Reviews	84.17	0.86	0.80
Eliminating stop words	85.33	0.86	0.83

Table 3: Values after cleaning the data

4.3 Metrics for the Bag-of-Words model

The output of the bag-of-words classifier will be used to create a new attribute that will be used for rating prediction. This implies that the cases of misclassifications must be minimized. In such a situation, the best metrics to evaluate the performance of a model are accuracy, recall and precision. Accuracy of a model shows how many samples are correctly classified. Precision of a model shows out of the positive classified examples how many examples were actually positive class. Recall of a model shows how many actual positive examples were detected out of the total positive examples in the dataset. The formula of these metrics is given as follows:

Precision= TP / (TP + FP)
Accuracy = (TP + TN) / Total samples
Recall= TP/ (TP+ FN)

(TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative)

4.4 Training Split

Determining the training and testing split in a dataset is necessary as if the training set is too large then the model starts to overfit the training data resulting in high variance. On the other hand, if the training data is too small, then the model might underfit, causing a high bias towards a specific class. The training and testing split is decided by training the dataset on various splits and the resulting model is evaluated based on the metrics of accuracy, precision and recall. These graphs were plotted against a test size of 10,000 samples.

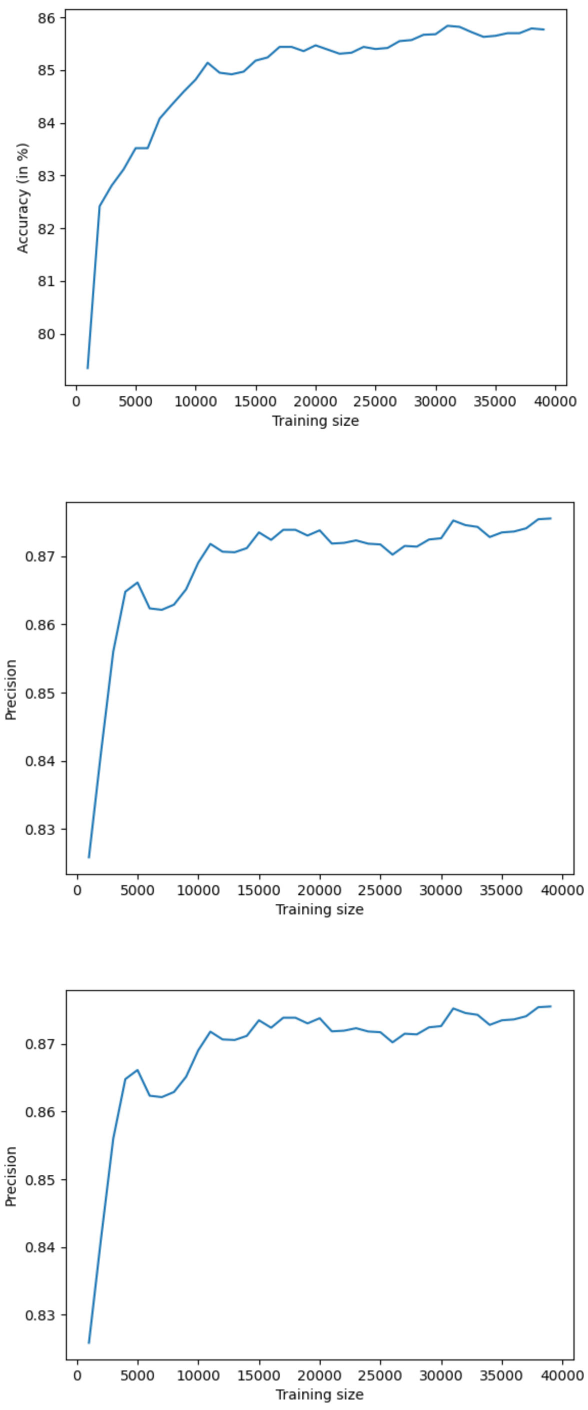


Figure 2: Metric plots for various training sizes

Training Size	Accuracy	Precision	Recall
5000	83.52	0.86	0.79
10000	84.82	0.86	0.82
15000	85.18	0.87	0.82
20000	85.47	0.87	0.82
25000	85.4	0.87	0.83
30000	85.68	0.87	0.83
35000	85.65	0.87	0.835

Table 4: Training split table

4.5 Laplace smoothing:

Laplace smoothing is a smoothing technique that handles the problem of zero probability in Naïve Bayes. A value, alpha, is added to the frequency of every unique word in the dictionary so that no occurrence value is zero. This helps in the calculation of likelihood probabilities as it eliminated the issue of the likelihood being zero. As alpha increases, the likelihood probability moves towards uniform distribution. Finding a suitable value of alpha is done using the metrics accuracy, precision and recall for a training-testing split of 25,000 samples each

Alpha	Accuracy	Precision	Recall
1	85.4	0.87	0.83
2	85.56	0.87	0.83
3	85.6	0.87	0.83
4	85.45	0.87	0.83
5	85.39	0.87	0.83
6	85.37	0.87	0.82
7	85.34	0.87	0.82
8	85.37	0.87	0.82
9	85.32	0.87	0.82
10	85.33	0.87	0.82

Table 5: Laplace Smoothing Table

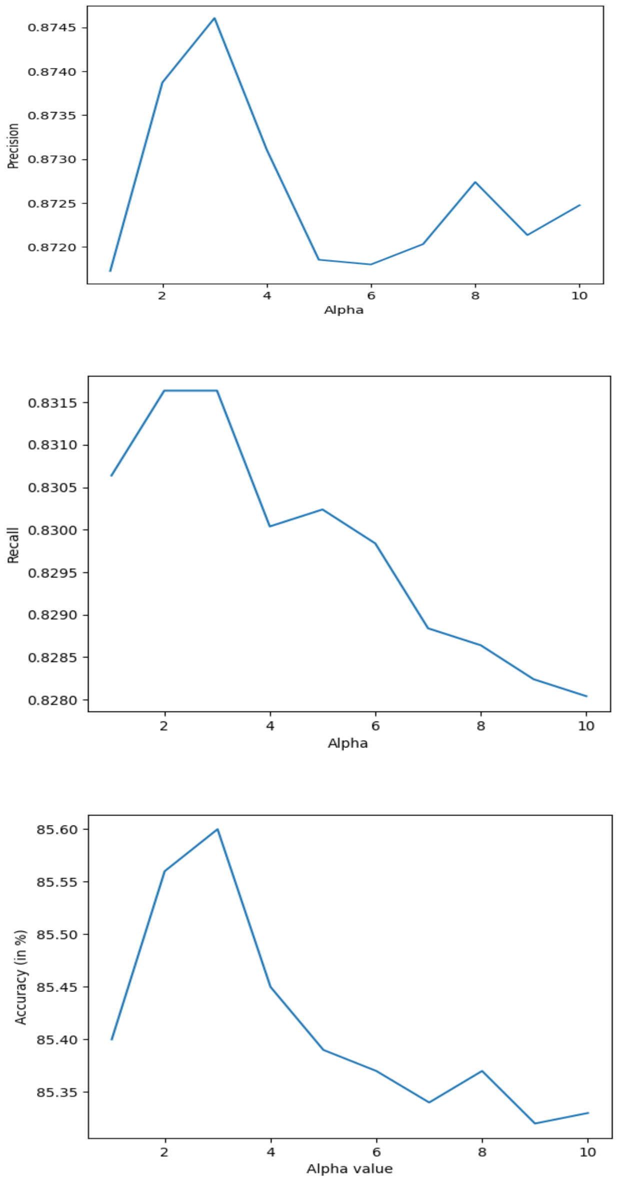


Figure 3: Precision, Accuracy, Recall for different alpha values

5. REGRESSION MODELS

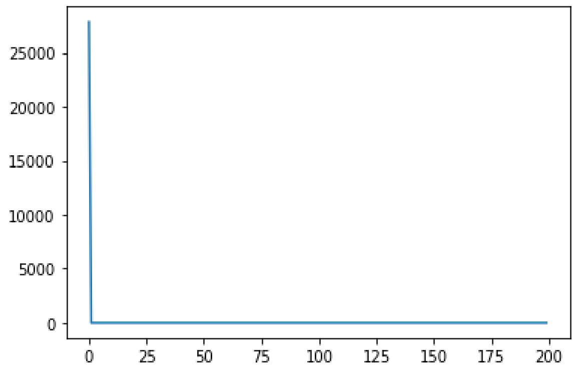
5.1 Deep Neural Network

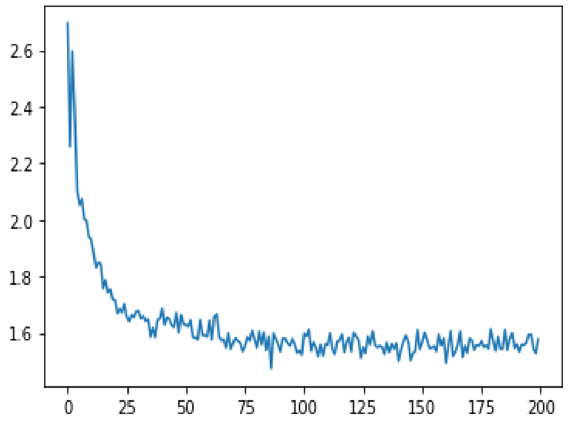
PyTorch is a GPU and CPU optimized tensor library for deep learning. It provides a degree of versatility while also allowing deep learning models to be expressed in a user-friendly manner. The Deep Neural Network approach was preferred over Linear Regression as the initial approach. While preprocessing the data, we discovered that some of the attributes do not have a linear correlation and hence, DNN was chosen. DNN was found to be more stable for the data and was combined with activation functions such as ReLU.

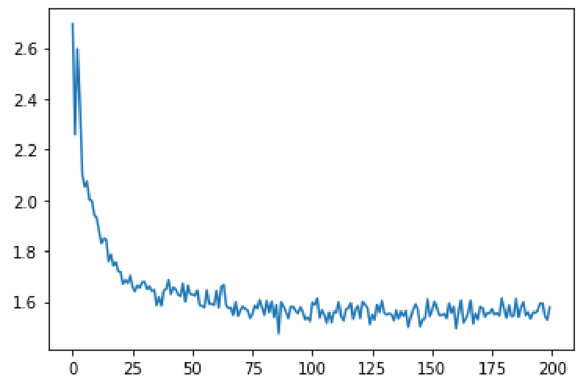
The hyperparameters were tuned and used to evaluate the model. ReLU was the obvious choice as it is less computationally expensive and involves simpler mathematical operations. It takes a real valued number as input and thresholds it at 0.

For a bigger neural network with lots of neurons, having sigmoid or a tanh will use all neurons making the activations dense, which would be a costly move. An ideal implementation of a NN would require fewer neurons making the activations sparse and efficient which ReLU or Rectified Linear Unit Activation Function provides.

For a bigger neural network with lots of neurons, having sigmoid or a tanh will use all neurons making the activations dense, which would be a costly move. An ideal implementation of a NN would require fewer neurons making the activations sparse and efficient which ReLU or Rectified Linear Unit Activation Function provides.







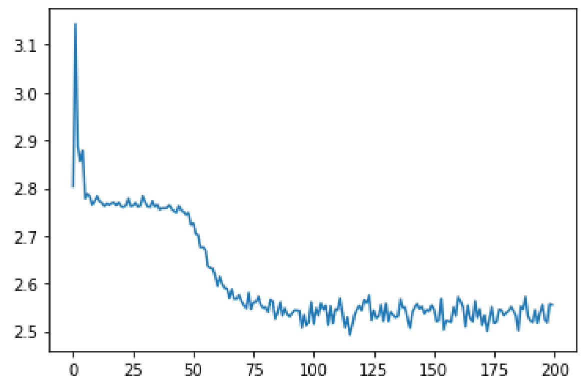


Figure 4: Loss Plots for different hyperparameters

5.2 Support Vector Regressor

SVR allows us to determine how much error is reasonable in our model and will fit the data with an appropriate line (or hyperplane in higher dimensions).SVR is a powerful algorithm that allows us to choose how tolerant we are of errors, both in terms of an acceptable error margin() and in terms of tuning our tolerance for exceeding that acceptable error rate. SVR recognizes the role of non-linearity in the data and offers an effective prediction model. SVR is memory efficient, which means that training the model needs less computational resources. This is due to the immense computational benefits of presenting the solution using a small subset of training points. The relationships between features and labels are non-linear or complex. This is because, in the case of support vector regression, it allows you to transform non-linear relationships to higher-dimensional problems.

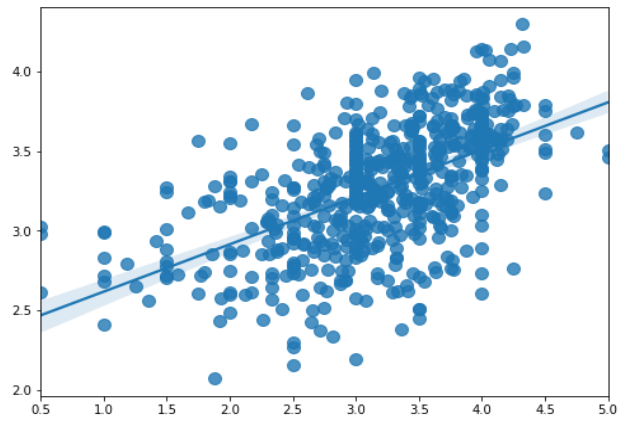


Figure 5: SVR plot

5.3 Random Forest Regressor

A Random Forest is an ensemble technique that uses several decision trees and a technique called Bootstrap and Aggregation, also known as bagging, to perform both regression tasks. Instead of depending on individual decision trees, the basic concept is to combine several decision trees to determine the final output. As a base learning model, Random Forest uses multiple decision trees. Row and function sampling are done at random from the dataset, resulting in sample datasets for each model. Random forest regressor for our dataset gave an r2 score of 0.25 [6]

6. CONCLUSION

In this project, we have seen that we can predict movie ratings based on the attributes of the movie. For this data was merged from multiple sources and processed. Then, we implemented our regression models namely Deep Neural Network, Support Vector Regressor and Random Forest Regressor which yielded an r2 score of -11, 0.20 and 0.25 respectively

With the nature of data, we are training, it is normal to not see an increase in prediction accuracy. People can choose to like or dislike a movie based on various reasons; certain subjective attributes are hard to encode. Due to this , it is not often possible to achieve higher accuracy.

Even though feature selection helped in understanding the influence of certain attributes of a movie with respect to predicting rating, the influence was not significant enough. Intuitively, we know that the artist and directors are very important to predict ratings.
Future work in these lines would be encoding the data in a better way; performing optimal feature selection; choosing the hyper parameters more methodically; selecting hybrid-models to train with better intuition and training the model for longer time for better conversion.

7. REFERENCES

[1] Armstrong, N., & Yoon, K. (1995). *Movie rating prediction*, Technical Report, Carnegie Mellon University.

[2] Abarja, R. A., & Wibowo, A. (2020). Movie Rating Prediction using Convolutional Neural Network based on Historical Values. International Journal.

[3] D.abidin, c. Bostanci, a. Site.(2018) *Movie Rating Prediction with Machine Learning Algorithms on IMDB dataset , ICATCES'18(International Conference on Advanced Technologies, Computer Engineering and Science)*, Safranbolu, Turkey, May 11-13, 2018

[4] Jeffrey Lund., & Yiu-Kai Ng,*Movie Recommendations Using the Deep Learning Approach*, Brigham Young SA University, Utah,84602, USA.

[5] F. Harper and J. Konstan. *The Movielens Datasets: History and Context.* ACM TiiS, 5(4): Article 19, 2016.

[6] Zahabiya Mhowwala.,A. Razia Sulthana , and , Sujala D. Shetty,*Movie Rating Prediction using Ensemble Learning Algorithms,*(IJACSA) International Journal of Advanced Computer Science and Applications, Vol 11, No. 8, 2020,Dubai,United Arab Emirates, 2020

.