Prediction of Movie Ratings Using Artificial Neural Networks

Viswa Suhaas Penugonda[1], Bharath Varma Kantheti[1],
Sowmya Reddy Kogilathota Jagirdar[1]
[1]Indiana University Bloomington

***Abstract* – We worked on a dataset which consists of several features such as language, genre, release year, region, ratings and cast member roles. The goal of the project is to predict the rating of a movie, based on the historical statistics of the above-mentioned features. Upon doing sufficient Data pre-processing which included Data Cleaning, Exploratory Data Analysis and Feature Selection, we obtained an appropriate data set structure. Using the dataset, we applied various Machine Learning models and tried several Artificial Neural Network Architectures to find the best model/architecture that sufficiently predicts the rating of a movie released. Some of the machine learning models used include Gradient Boosting Regressor, Support Vector Regressor, Linear Regression, Stochastic Gradient Descent Regressor and the ANN included architectures included a combination of dense layers along with combinations of dense layers and a dropout. An ANN architecture with a combination of dense layers performed best in predicting the ratings using the dataset used.**

***Keywords* – Data Cleaning, Machine Learning, Regression, Artificial Neural Networks.**

I. Introduction

Considering the rise in the use of Over the Top (OTT) platforms due to reasons such as convenience of having a movie experience at home and access to internet everywhere in our day-to-day life's, the need to go to a movie theatre has become something that needs a thought to be given. The ratings are playing a major role in the decision-making process of an average movie goer whether to have a theatre experience or wait for the movie to be released online in an OTT platform. Having a software which can help predict the ratings using the parameters such as the popularity of the cast and crew of the film and their previous movie history along with the other features like the time at which the movie is released, duration of the movie can help decide as to go to a movie theatre to watch the film or not.

Having this as our objective, we tried to implement an architecture which can predict the rating of the newly released movie.

II. Data and Features:

For this project, we used an open- source dataset obtained from a GitHub repository. It consists of 18,166 entries with 14 data columns (features). The dataset includes both categorical and numerical data. Some of the important features include Genre, ratings, role and number of votes. The major is to use these features to obtain an accurate rating of a movie.

III. Data Preprocessing:

It is a crucial step to process the raw data present in the dataset to feed the resulting data into the machine learning model. Doing this helped us obtain usable insights which further helped in fine tuning the model. While performing this process, we did the following:

A. Removed the null values in each of the features in our dataset.

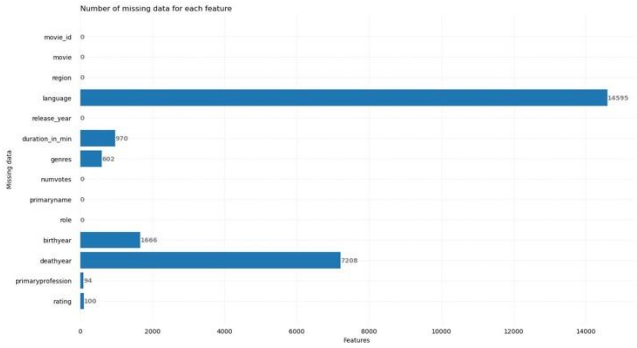

Fig. 1: Number of Null Values present in the features

- The feature 'Language' consists of 14,595 null values. Since the total number of data entries present in the data set is 18,166 it doesn't provide any insights for further processing. Hence it is dropped.
- The null values in features 'birth_year', 'death_year', 'duration_in_mins' are replaced with the mean values of their repective columns.
- The null values in the feature 'Genre' are replaced with the maximum repeated genre (mode).

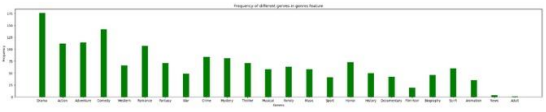

Fig. 2: Frequency of each genre

As seen in the above diagram, the null values are replaced with genres such as 'Adventure', 'Comedy', 'Drama'.

- The null values in the feature 'primaryprofession' are also replaced with the mode of the column.

Advisor: Yuzhen, Ye

B. Generalizing the features in the dataset

- Since the number of unique types of genres are high, we divided the set of genres into main and sub genres. We later generalized the sub- genres into their corresponding main genres.

Main Genre	Sub Genres
Drama	Drama, Romance, Musical, Western, family, Fantasy, Music
Adventure	Adventure, War
Thriller	Crime, Horror, Action, Thriller, Mystery, Sci-Fi, History
comedy	comedy
Documentary	Biography, Documentary

Fig. 3: List of Main- Genres and Sub- Genres

Generalizing helped us reduce the complexity of processing the dataset.

- The feature 'primaryprofession' is also generalized in a similar way as we did with the feature 'Genre'.

C. One- Hot Encoding

Features such as 'role', 'movie', 'region' are one-hot encoded and are later dropped after processing.

D. Features are dropped due to inadequate relevancy

Features such as 'primaryname', 'movie_id' are dropped since each value is uniques to the row of the data entry and does provide any insights for further processing.

E. Categorized the movies based on the number of votes

Votes Class	Range of Votes
A	Votes>=500,000
B	200,000<=Votes<500,000
C	50,000<=Votes<200,000
D	20,000<=Votes<50,000
E	5,000<=Votes<20,000
F	2,000<=Votes<5,000
G	1,000<=Votes<2,000
H	500<=Votes<1,000
I	<500

Fig. 4: Vote Class for the number of votes

We have categorized the movies as shown in the above table based on the number of votes they've obtained to provide us an insight as to how the number of votes affect the ratings of a movie.

IV. Exploratory Data Analysis

We plotted a scatter plot with ratings against duration in minutes of a movie to find if an outlier exists.

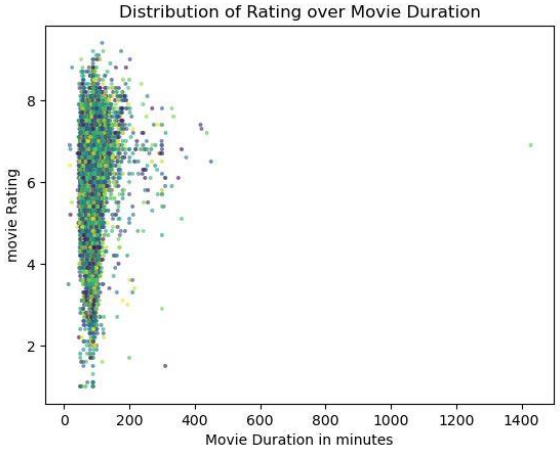

Fig. 5: Scatter plot with ratings against duration of movie

As seen in the above plot, we found an outlier which was later dropped.

We have also plotted a scatter plot with ratings against the number of votes to find if there's an abnormality in the data.

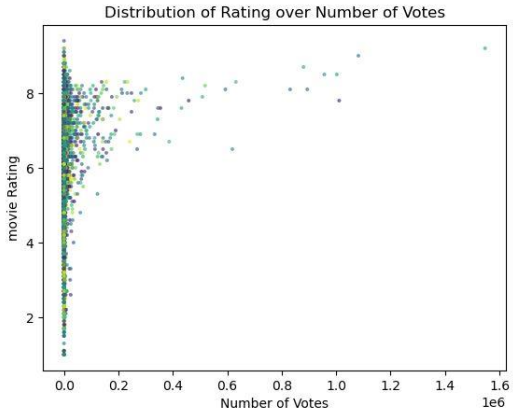

Fig. 6: Scatter plot with ratings against the number of votes

As shown in the above plot, we found outliers, which were later removed for further processing.

V. Normalization:

Since the type and magnitude of every feature in the dataset is different, we've normalized the entire dataset using MinMax scaler to get uniformity, thereby obtained accurate insights as to how the data is.

VI. Feature Selection:

To find the correlation between attributes in the dataset, we plotted heatmap.

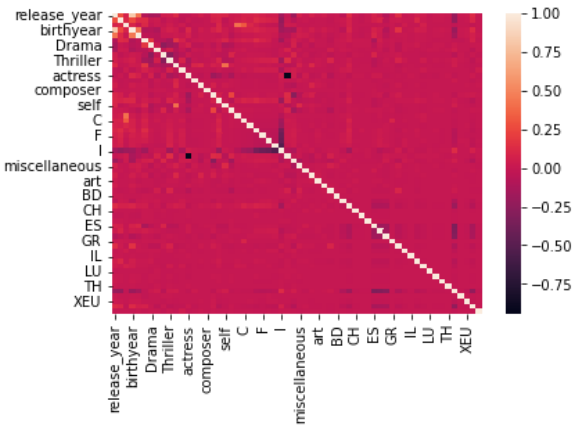

Fig. 7: Heat map to find the correlation

As seen in the above diagram, we obtained a correlation between actor and actress to be 94%, release_year and birthyear to be 83% and classes H and I to 52%. Assuming we have threshold limit of 90%, among the attributes actor and actress, the attribute 'actor' is dropped.

VII. APPROACHES

A. *Gradient Boosting Regressor Model*

By integrating the weak learners or weak predictive models, the gradient boosting algorithm creates an ensemble model. Models can be trained using the gradient boosting approach for both classification and regression problems. The approach known as Gradient Boosting Regression is used to fit the model that forecasts the continuous value. The accuracy that was achieved for this model is 25% and the mean squared error is 0.0125.

B. *Support Vector Regression Model*

A supervised learning model called Support Vector Regression (SVR) can be utilized to carry out both linear and nonlinear regressions. The objective of applying linear regression, is to reduce the difference between the forecast and the actual data. To ensure that the mistakes do not go above the cutoff, Support Vector Regression is used to analyze a data set. This model takes more time than Gradient Boosting Regressor Model and the accuracy achieved is 20% and the mean squared error is 0.01340.

C. *Linear Regression Model*

A model that assumes a linear relationship between the input variables (x) and the single output variable (y) is known as a linear regression. The variable y can be determined more specifically from a linear combination of the input variables (x). The technique is known as simple linear regression when there is only one input variable (x). Multiple linear regression is a term that appears frequently in statistical literature when there are multiple input variables. This model does not follow the trend of the data and it is over-fitting because the r2 score that was achieved is negative.

D. *Stochastic Gradient Descent Model*

An iterative optimization method known as stochastic gradient descent (or SGD) comes close to a smooth, differentiable gradient. SGD estimates the gradient using a randomly chosen portion of the data, whereas the true gradient must be calculated using all the data. The simplicity, computation efficiency, and simplicity of this algorithm are its benefits. Having many parameters to fine-tune, the algorithm's sensitivity to the scale or units of the independent variables, and the need to iterate potentially many times without really knowing whether the final solution is a local or global minima for the thing you are trying to optimize are some of the drawbacks. The accuracy that was achieved for this model is 13% and the mean squared error is 0.0145.

E. *Ridge Regressor Model*

Any data that exhibits multicollinearity can be analyzed using the model tuning technique known as ridge regression. This technique carries out L2 regularization. Predicted values differ much from real values when the problem of multicollinearity arises, least-squares are unbiased, and variances are significant. The accuracy that was achieved for this model is 18% and the mean squared error is 0.0137.

E. *Lasso Regressor Model*

Like linear regression, lasso regression employs the "shrinkage" strategy, which involves reducing the coefficients of determination until they are zero. Regression coefficients as observed in the dataset are provided via linear regression. To prevent overfitting and improve their performance on other datasets, you can reduce or regularize these coefficients using the lasso regression method. When the dataset exhibits strong multicollinearity or when you want to automate feature selection and variable exclusion, this sort of regression is used. The r2 score that was achieved for this model is 0.00875 and the mean squared error is 0.0166.

F. *Random Forest Regressor Model*

With the aid of several decision trees and a method known as Bootstrap and Aggregation, also referred to as bagging, Random Forest is an ensemble methodology capable of handling both regression and classification tasks. This method's fundamental principle is to integrate several decision trees to get the result rather than depending solely on one decision tree. Multiple decision trees serve as the fundamental learning models in Random Forest. We create sample datasets for each model by randomly selecting rows and features from the dataset. This component is known as Bootstrap. The accuracy that was achieved for this model is 25% and the mean squared error is 0.0125.

G. *Adaboost Regressor Model*

A meta-estimator called an AdaBoost regressor starts by fitting one regressor on the original dataset, and then it fits subsequent copies of the regressor on the same dataset with the weights of the instances being changed in accordance with the error of the most recent prediction. Decision trees with a single split, or "decision stumps," are the weak learners in AdaBoost. AdaBoost works by giving cases that are challenging to categorize more weight and instances that are already handled well less weight.

The r2 score that was achieved for this model is 0.0039 and the mean squared error is 0.01676.

H. *Neural Networks Model*

For all the neural network approach we built a regression model using deep learning in Keras. We experimented with different options with the activation function as used in the hidden layers as a rectified linear unit, or ReLU. Its advantages of being nonlinear and its capacity to not stimulate every neuron at once make it the most popular activation function. In layman's words, this indicates that the network is sparse and extremely effective because just a small number of neurons are stimulated at once.

- *Neural Networks + Dense layers:*

This model performed better than all the other models that were tried before. Our loss metric is the mean squared error, and our minimization method is the "Adam" optimizer. The key benefit of the "adam" optimizer is that, unlike with gradient descent, we don't have to specify the learning rate, sparing us the time-consuming chore of optimizing the learning rate for our model. We also provided epochs parameter, which denotes the number of training iterations. We have taken 300 epochs for this model and 705 neurons. The accuracy that was achieved for this model is 89.5% and the mean squared error is 0.0018.

- *Neural Networks +Dropout (0.5) + Dense layers:*

This model performed better than all the other models that were tried before other than the above model. Using the same parameters as the neural networks and dense layer model, we have additionally added a dropout of 0.5 with 300 epochs to check the performance of the model. The accuracy that was achieved was 84% and the mean-squared error is 0.0027.

- *Neural Networks +Dropout (0.2) + Dense layers:*

This model performed better than all the other models that were tried before other than the above model. Using the same parameters as the neural networks and dense layer model, we have additionally added a dropout of 0.5 with 300 epochs to check the performance of the model. The accuracy that was achieved was 88.7% and the mean-squared error is 0.0019.

VIII. Summary of Results

We can observe from the results that the Linear regression model has the least accuracy and artificial neural networks with dense layers performs the best which is close to the neural networks along with the dropout as 0.2. The r2 scores and the mean squared error are shown in the Figure 8.

Linear regression model performs the worst because it overfits the model extremely. As seen in the Table, the artificial neural network model using dense layers has the has the best r2 scores among all the models which is 89.5%.

Model	R2 Score	Mean Squared Error
ANN+ Dense Layers	0.895	0.0018
ANN+ Dropout(0.5)+ Dense Layers	0.84	0.0027
ANN+ Dropout(0.2)+ Dense Layers	0.887	0.0019
Gradient Boosting Regressor	0.25	0.0125
Support Vector Regressor	0.20	0.0134
Linear Regressor	-1.60	2.70
Stochastic Gradient Descent Regressor	0.13	0.0145
Ridge Regressor	0.18	0.0137
Lasso Regressor	0.008	0.0166
Random Forest Regressor	0.25	0.0125
Adaboost Regressor	0.003	0.0167

Fig. 8: Summary of results

The scatter plot in Figure 9 defines the relationship between the dependent variable and independent variable achieved with our Artificial Neural Networks using the dense layer model.

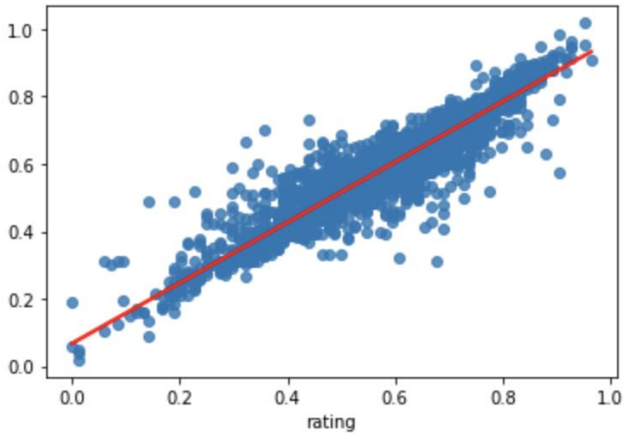

Fig. 9: Scatter plot

IX. Conclusion

The model that we have chosen for the movie rating predictions is using the neural networks using the dense layers based on the results that we have achieved. Using this model, we can forecast the movie rating based on the popularity of the cast and crew and their prior filmography, as well as other factors like the time the movie is released and the length of the movie, which might influence whether viewers choose to watch the movie in a theater. LSTM, Bi-directional LSTM, and transformer-based models like BERT, RoBeRTa were deep learning models that we would have loved to spend more time on if we had had the time. We liked exploring better approaches for data analysis and visualization. Along with a few techniques for data analysis and visualization, we learnt how to compare and use various categorization methods. We are grateful to Professor Yuzhen Ye for providing us with the chance to work with a real-world dataset.

X. References

[1] Lim, Yew Jin, and Yee Whye Teh. "Variational Bayesian approach to movie rating prediction." *Proceedings of KDD cup and workshop*. Vol. 7. 2007.

[2] Li, Xiaoyue, et al. "Research on movie rating prediction algorithms." *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*. IEEE, 2020.

[3] Marović, Mladen, et al. "Automatic movie ratings prediction using machine learning." *2011 Proceedings of the 34th International Convention MIPRO*. IEEE, 2011.

[4] Dhir, Rijul, and Anand Raj. "Movie success prediction using machine learning algorithms and their comparison." *2018 first international conference on secure cyber computing and communication (ICSCCC)*. IEEE, 2018.

[5] Abarja, Rudy Aditya, and Antoni Wibowo. "Movie rating prediction using convolutional neural network based on historical values." *International Journal* 8.5 (2020).

[6] Basu, Somdutta. "Movie rating prediction system based on opinion mining and artificial neural networks." *International Conference on Advanced Computing Networking and Informatics*. Springer, Singapore, 2019.