An Investigation in predictive model of Rating of movies

Group 1: Jinshun Su, Maeshal Hijazi, and Chunbo Wang

1 Introduction

This study is aimed to draw a prediction of the rating of a movie based on its result input. With this model applied to commercial movies shall one be able to give an estimation on the rating those movies may generate. Considering the study's predictive nature, a regression analysis is to be expected. Consequently, the tasks lie in finding out the potential factors that contribute to the rating of a movie and building a regression model that predicts the outcome (rating) as satisfactory as possible when compared to the targets of test inputs.

To better analyze the correlation between properties and the rating of a movie, a preprocessing stage was performed and recorded in the second part of the paper. Among the given properties, the ones that give out the most accurate outputs were recognized as inputs that contribute to the rating. Then, the dataset was split into training and testing groups for generating and assessing a predictive model with the method of supervised learning. As a result, the output values were compared to target values and the difference was set to error used to assess the model.

The rest of this report is organized as follows. Section 2 introduces the description of dataset and preprocessing of data. Section 3 describes the regression analysis algorithm. The numerical results and discussions are provided in section 4, and finally section 5 concludes the project.

2 Data Preprocessing

The selected Dataset consists of about 1000 variables (movies) along with Genre, Description, Director, Actors, Runtime, Rating, Votes and Metascore of each individual. Properties were selected as input (Votes, Revenue and Metascore) and output (Rating). The csv file was read via **pandas** library. By browsing the dataset, numerous missing values were found in input, so imputation was applied to fill in those values with mean value of that column.

Since all variables were numerical values, **arrays** was used to store all data and carry operations. And as a general approach, the values were standardized using **sklearn.standardizer** to convert into a scoop of 0 to 1.

As for inputs, Runtime, Rating, Votes and Metascore were chosen as candidates. To verify the reliability of these inputs, hypothesis tests were chosen from **statsmodels.stats**. The null hypothesis was set to be no correlation between input(x) and output(y) while the alternative being there is correlation; With the significance of 0.05, all inputs were tested to be valid.

3 Regression Analysis

In the beginning of the analysis, each training input candidate was fit to a linear model with the training output and the regression was plotted; nonetheless, all cases were assessed to be non-linear according plots. Thus, **MLPREGRESSOR** was ingratiated in model training. And the test group was input into the model, giving results for error analysis.

For the purpose of optimizing the model, different combinations of input variables were designed and tested for building the model and comparing the errors in result. First, 2 variables were chosen for building the model. Based on combinatorics, there were 3 combinations that could compose to inputs(x1,x2),(x2,x3),(x1,x3). Each result was recorded for computing accuracy. Then, all 3 variables were set as inputs, and the result recorded and compared to all others.

4 Numerical Results

In order to check correlation between input features and target label, we did the t-test for input features and target label and obtain the p-value of each input, which is shown in Table 1. According to the Table 1, the p-values of all input features are lower than 0.05. Hence, runtime, votes, and metascore have a significant impact on response of rating. Next, we train and test the model with different number and combination of input features. Based on our dataset, we randomly select 250 samples as the test group.

Table 1: P-value for each input feature

	Runtime	**Votes**	**Metascore**
Rating	0	5.42e-178	0

4.1 Case I

In this case, we consider runtime and votes as input features. And the variation of output values and error for test sample is presented in Figure 1 and 2.

4.2 Case II

In this case, we consider runtime and metascore as input features. And the variation of output values and error for test sample is presented in Figure 3 and 4.

4.3 Case III

In this case, we consider votes and metascore as input features. And the variation of output values and error for test sample is presented in Figure 5 and 6.

4.4 Case IV

In this case, we consider runtime, votes, and metascore as input features. And the variation of output values and error for test sample is presented in Figure 7 and 8.

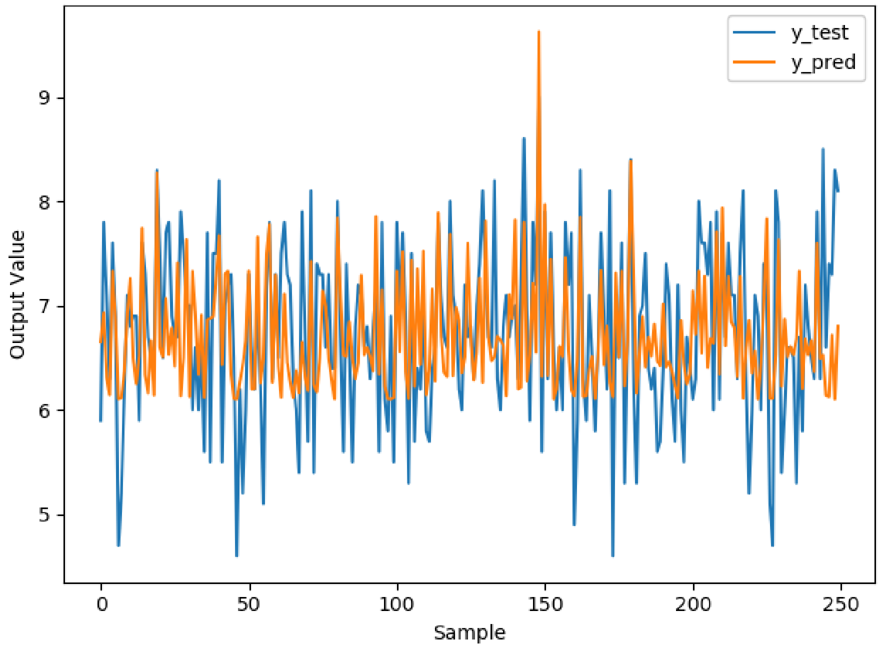

Figure 1: The output values of prediction and test group in case I

4.5 Discussion

To verify the efficiency of the regression for prediction, we obtain values of mean squared error in different cases, which is shown in Table 2. According to 2, Case IV has lowest value of mean squared error. Therefore, the performance of prediction for rating of rating is the best while considering three input features including runtime, votes, and metascore.

Table 2: Values of mean squared error in different case

	Case I	**Case II**	**Case III**	**Case IV**
Mean squared value	0.586	0.589	0.518	0.469

5 Conclusion

Based on the Numerical result from the proposed model, it was found that a strong correlation between a movie's properties(runtime, metascore and votes) and its rating exists: while all three inputs were used to fit the model, the resultant mean square error

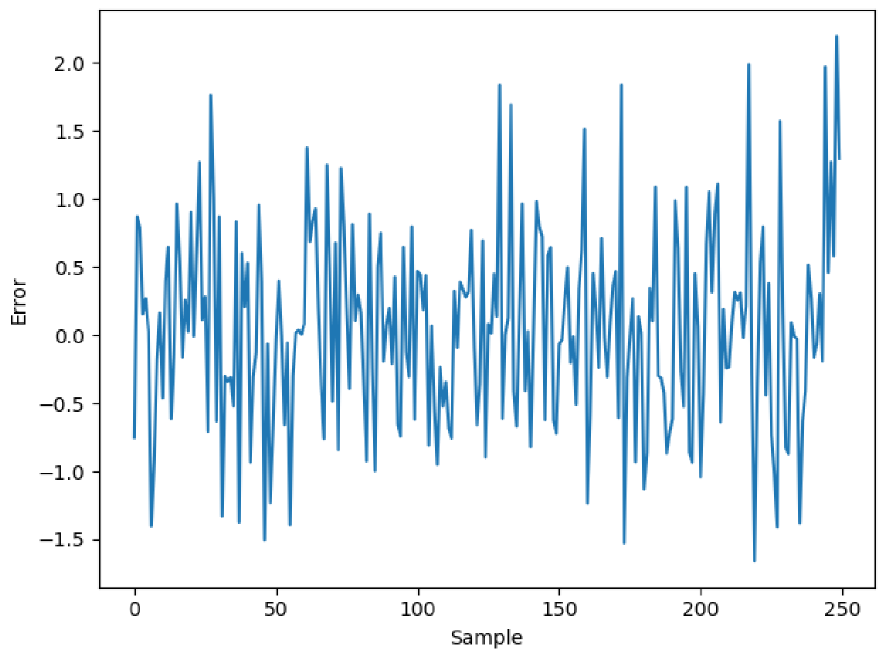

Figure 2: The output values of error in case I

between target and test output was only 0.469, the smallest among all cases. Therefore, it was concluded that the model was relatively satisfactory when all properties(runtime, metascore and votes) are considered as factors to contribute to the rating of a movie. And the built model via **MLPREGRESSOR** was able to produce efficient prediction according to the extent of match between prediction and target in Plot 6. Beyond the scope of this study is the optimization of this built model, which requires extensive search in more data as the training group, and maybe other advanced algorithms.

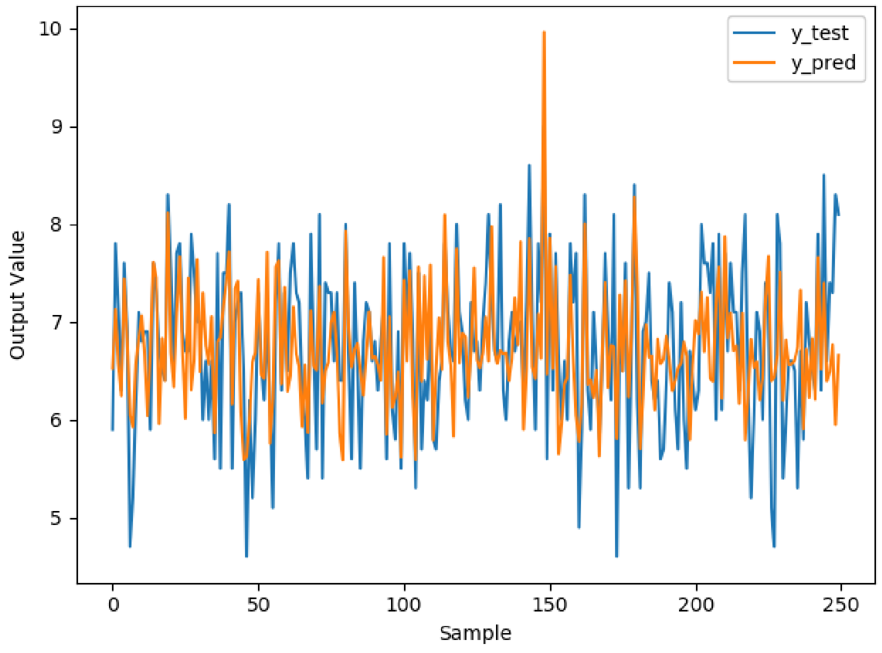

Figure 3: The output values of prediction and test group in case II

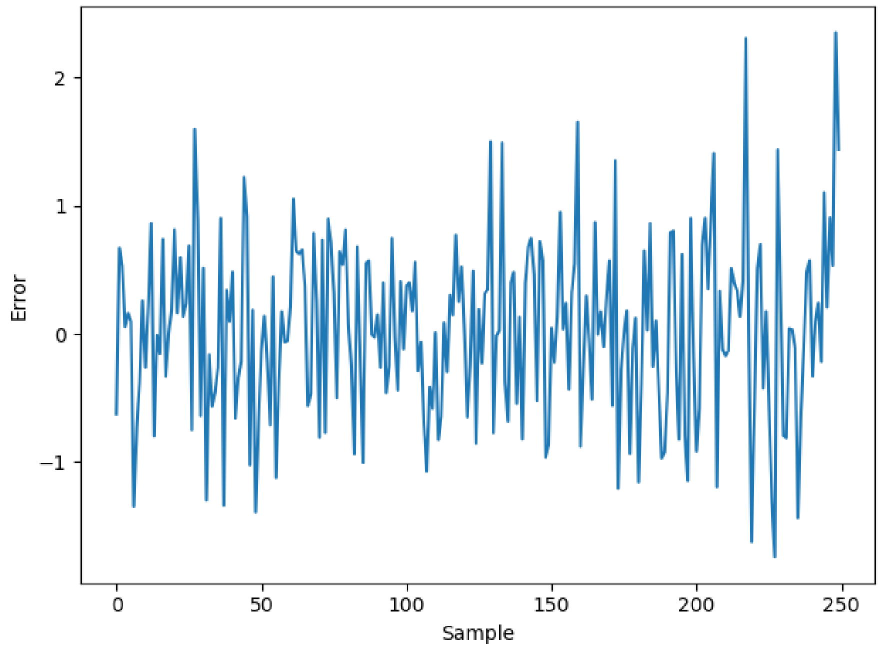

Figure 4: The output values of error in case II

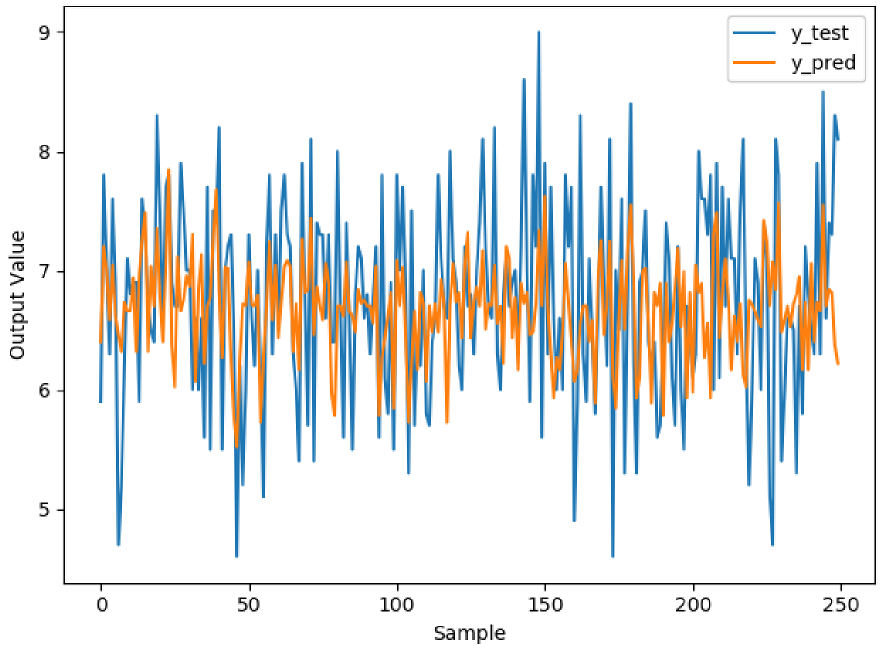

Figure 5: The output values of prediction and test group in case III

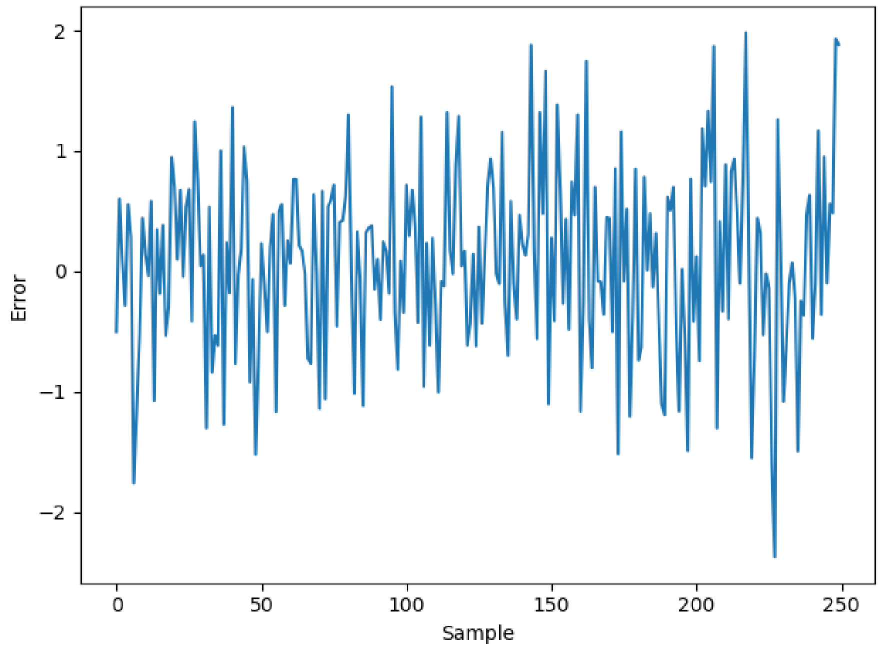

Figure 6: The output values of error in case III

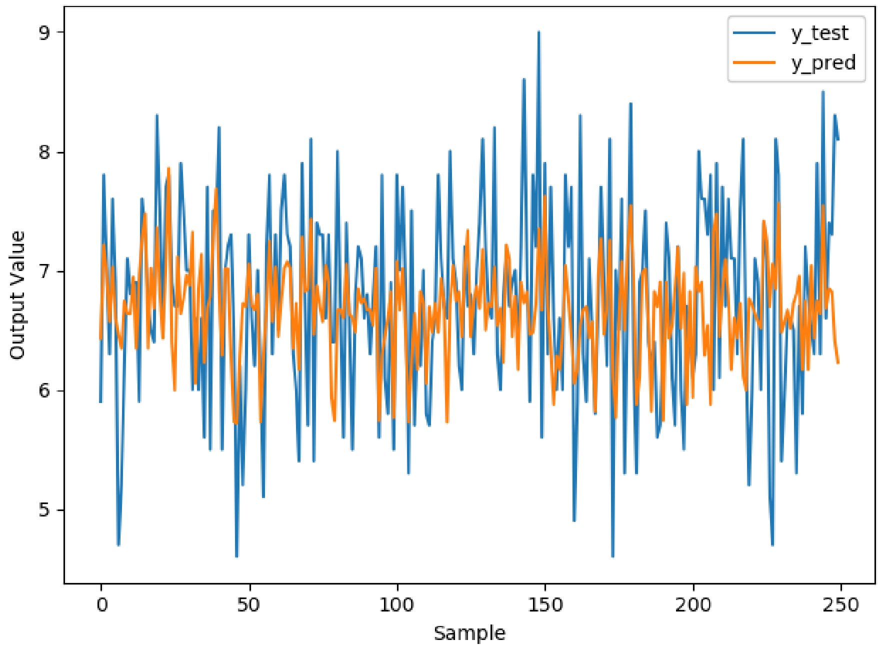

Figure 7: The output values of prediction and test group in case IV

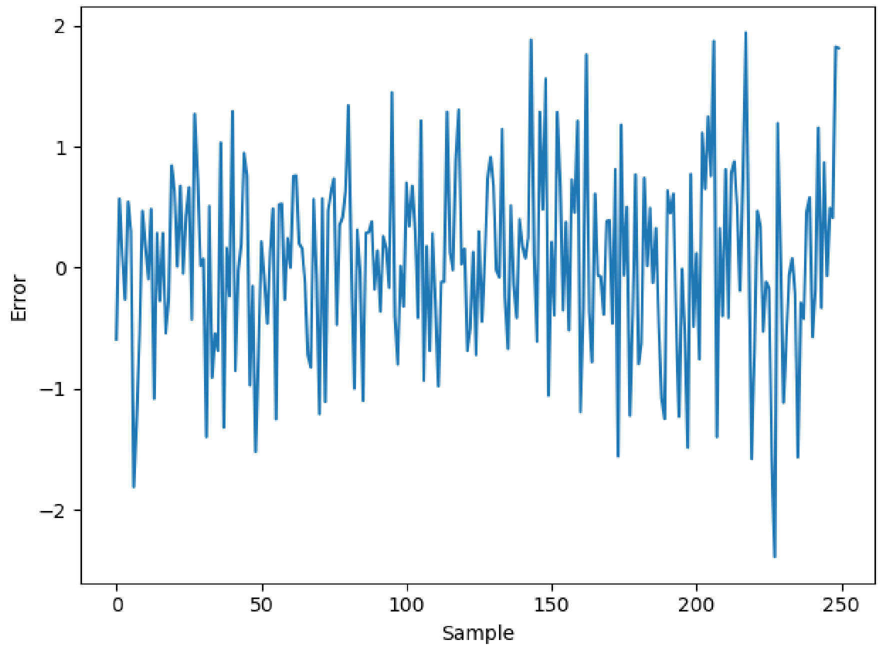

Figure 8: The output values of error in case IV