Midterm Report (Kaggle Competition)

my kaggle username is yanjie99

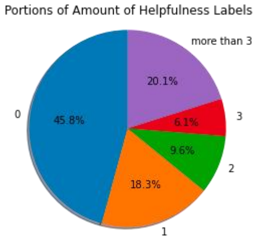
Preliminary Analysis/Exploration

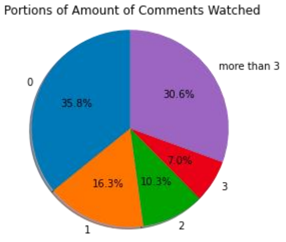
It seems that you are working on a project to analyze user reviews from Amazon Movie Reviews and predict the star rating score using various features such as user/product identifiers, helpfulness ratings, timestamp, review summary, and review text. By doing so, you aim to estimate the popularity and reputation of movies and potentially improve recommendation systems in this field.

first thing I do is importing libraries, such as missingno and hstack. and then, I import the train data. After that, I am going to discover some significant properties among the data by carrying out an exploratory data analysis.

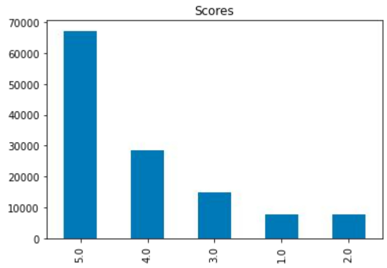
The original data set contains 9 columns, and column "Id" serves as the unique identifier of each row entry which in other words, is irrelevant to the modeling. I also get the user data and found that most active user among the dataset has no more than 3000 records in it. Compared to the total amount of over 1 million records, this is not a very significant portion. Therefore we have to worry about whether a certain user is able to make too big an influence to the entire prediction. This is the same in terms of the column "ProductId" because its most frequently seen entry has no more than 3000 records either.

I have also analyzed the "HelpfulnessNumerator" and "HelpfulnessDenominator" features and found that no single value accounts for more than 50% of the total. Therefore, you believe that none of the values are too dominant or overwhelming to significantly impact the performance of the predictor.





And for the score data,　I have noticed that commenters tend to give a high rating of 5.0/5.0 to movies, which could cause an issue when implementing machine learning. Specifically, the classifier may predict an unlabeled movie to have a rating of 5.0/5.0, which could lead to biased results. Therefore, you plan to address this potential issue in the following chapters to avoid it from happening.



I also an analyze the summary and text, in this step, I have generated word cloud images for both the "Summary" and "Text" features to visually analyze the most frequently occurring words. The purpose of this was to identify any potentially valuable information that could be used in the analysis. I have observed that there is not much overlap between the word clouds generated for the "Summary" and "Text" features. This is likely due to the fact that the summary is more concise and contains less information compared to the text. However, you have concluded that since there are few common words, there is not a strong correlation between the two features. As a result, you have decided to use both the "Summary" and "Text" features as factors in your predictor.

I have done the Missing Value Detection, As the overview suggests, there are missing values inside our data set. So it is necessary to research their properties and make sure no distortions caused by them. Except "Score" (our projection of this competition which contains test data), it seems that there almost no missing values insides other properties. According to the following heatmap, the missing values in other columns only exist in the property "Summary" and "Text", while neither of each is strongly correlated to one another. This makes the problem simple because I can just apply a simple imputation on them without considering other situations.

Data Preprocessing

first thing is Downsize the data whose score is 5.0/5.0, as I previously demonstrated, there are many instances of a score of 5.0/5.0 in the dataset. However, to prevent potential side effects of this (i.e., the predictor being biased towards predicting a score of 5.0/5.0 for unknown records), you have decided to limit the scale of the dataset by removing instances with a score of 5.0/.

One-hot Encoding to Categorical Features, I have chosen one-hot encoding as the method for digitalizing categorical features. The reason for this decision is that label encoding assigns a unique numerical value to each identifier based on their order of appearance, which conflicts with the fact that identifiers are not measurable features. The disadvantage of one-hot encoding is that it requires more memory compared to label encoding. However, you plan to demonstrate how to handle this potential memory issue.

Standardize the numerical features, Standardization helps to reduce the influence of outliers and converge faster. Calculate TF-IDF and Vectorize, This is the baseline step. By vectorizing the text values, we can access to more information about this dataset so that a higher accuracy is obtained. Concatenate all the features, make dense series to a sparse matrix. stack all the sparse matrices above acording to their row indices , Sparse matrix helps solve the memory consuming trouble of the one-hot encoding.

After that , I have split out the train set and test set,This is a visualization of my sparse matrix. However, even if it is called "Sparse", it is not sparse at all from the perspective of maths because the elements within the matrix are barely zero. and Separate a ratio of train set for the use of validation.

Implement the classifier

I have tried the three models , Decision Tree, Random Forest and LogisticRegression The reason behind this decision was I did some research and found these models are most common ones for this kind of the problem.

I chose Logistics regression as my model because of its efficiency and am going to check its accuracy with K-fold cross validation. it is for cross-validation, it divides the dataset into k different parts and uses each part as a test set once and the rest as a training set. At each iteration, the model fits the training set and makes predictions on the test set, then computes the average error (in this case, mean squared error) of the predictions. In this way, we can obtain the performance indicators of the model on multiple test sets, so as to evaluate the performance of the model more accurately. In this example, we are using a logistic regression model and take the training set and labels as input, then run the CVKFold function for cross-validation, and finally get the best logistic regression model with the highest accuracy.