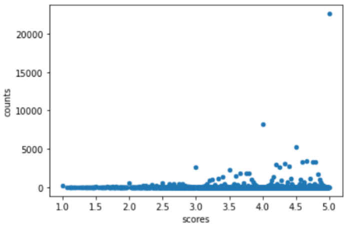
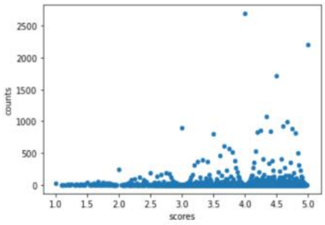
Amazon Movie Reviews Analysis Report

Preliminary Analysis

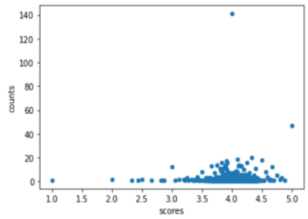
In this part, I analyze the different variables' relationship with the score. For each variable, I group by their values and calculated the average score. Then I count the number for each score and plot them. The diagrams we see can tell us how many unique (variable meaning) give the specific score. For example, in the first one, we can know users would like to give more high scores above 3, especially many users give 5.



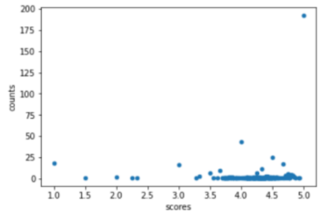
User-Scores



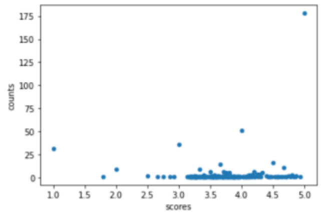
Product-Scores



Time-Scores



HelpfulnessNumerator-Scores



HelpfulnessDenominator-Scores

This would help us know the different distributions for each variable. From the analysis result, we can tell that different variables have different distributions on the score, especially time, helpfulnessNumerator and helpfulnessDenominator would be more helpful for the further analysis. Since their counts for different scores are distinct.

Feature Extraction

We should focus more on text data since they contain lots of information. Firstly, I find that different scores' review texts have different length distribution.

Score	1	2	3	4	5
Median Length	95	128	124	115	65

It's reasonable to add features as text_word_count (Count).

Secondly, since word counts play important roles in the classification. I believe words would perform better.

I pick the most common 50 words in each subset by scores. And then I put them together and remove duplicate ones to form a vocabulary. Each word in this vocabulary is a feature, by checking whether the review text contain this word.

So far, I add one Count and 67 word containing features to the data, plus original 3 items. There are total 71 features finally.

Used Techniques and Experiments

1. The primary feature extraction is based on the text column and process by the NLTK. I make use of this tool to preprocess the text as:
    1) Tokenize the text by space
    2) Converting letters to lower case
    3) Removing punctuations
    4) Removing stopwords;
2. The primary model technique is RandomForest implemented in sklearn. I also try other models, but it turns out that RandomForest would perform better:
    1. Logistic Regression: Relatively lower than KNN
    2. Decision Tree: Relatively lower than logistic Regreesion
    3. SVM(Linear SVC): takes too long to run and so that I can't get resuts.
    4. KNN: base implementation, a high RMSE score
3. Pandas and matplotlib are used for analyzing the data and visualizing.

Model Validation

According to the instructions, I use RMSE (mean square error) to measure my performance.

1. KNN: about 2.5
2. Logistic Regression: about 2.2
3. Decision Tree: about 1.9
4. SVM: still unknown. But it should perform well due to its idea of hyperplanes and kernel tricks.
5. Random Forest: 1.01

Creativity/Challenges/Effort

1. My first and important idea is to focus on the text column in the dataset. But how to extract features from this would be a big problem since we are not allowed to use deep learning to learn features automatically.
2. Hence, preprocess and extract features manually would require more idea about the text. Finally, I make use of the features of word count and word frequent
3. Data is quite large, it would take so long to train the model and preprocess the text. Maybe it's worth to try some other tricks to train or adjust features space.
4. I want to try the ensemble of different models, but due to the reason described in 3, it doesn't successfully give a result.
5. In addition, data is unbalanced in this case, which need to work further and may provide better results.