**Report**

**Intention**

Since this problem is trying to predict score (discrete values, 1,2,3,4,5) based on given features, all classification method we have learnt so far could be used except PCA, which is used for feature reduction.

**Preprocessing**

Since I used two different types of methodologies by using differing subset of features in the entire training dataset. I preprocessed all features in the dataset.

1. Clean up data by dropping all “scores” value with NaN

2. Converted “ID”, ”ProductID” into numbers start from 1 for each single unique “ID” and “Product ID”

3. For “Text”, “Summary”, I get rid of all punctuation, and make all letters lowercase.

**Methodology**

By the limited of time, I could only try 6 methods in total: Random Forest, Decision Tree, Logistic Regression, SVM, KNN, and SGD. I have tried two different models. First, I used all features given in metadata, except “Text” and “Summary”. Then used these directly to 6 methods I have. And I get training errors as: Random Forest is 100.0, Decision Tree is 100.0, SVM is 97.1 and the rest are under 60. Random Forest, Decision Tree and SVM seems are ideal ways to get a decent accuracy. Then I plugged them in into test set, separate from training set. The accuracy is very low.

Thus, I come up with a new method – transform data features. First, I combine “HelpfulnessNumerator” and “HelpefulnessDenominator” as a fraction called “helpfulness”. Then, I imported in two text files of positive words and negative words, counting attitude score by executing an “if” condition (if positive +1, if negative -1). The count score of “Summary” called “Summary\_score”, and the count score of “Text” called “Text\_score”. Then, I added a weight alpha on summary and plus text score and get the average and get final score I used in training model. Then I learned a basic function of classification – 2 features “helpfulness” and “Final\_score” and 1 outcome “score”.

N = 20000

Alpha = 2 Alpha = 5 Alpha = 8

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

It looks like alpha does not make a large difference. (I have tried other Ns) So, I just run Alpha = 5 for N = 50000

A screenshot of a cell phone

Description automatically generated

*Because of the runtime is large, the version in Kaggle is the version without text score which only increase 1%~2% of correctness while the runtime is significantly decreased.*

**Validation**

I set up a test set from a subset of train set. Then, I calculate the error by find the percentage of data that are successfully predicted. For example, there are 100 data, 50 data are correct, then the percentage would be 50%. Find the highest among all the modification version I have and upload to the Kaggle website.

**Improvement**

During the office hour, Professor informed me I could enhance my accuracy by changing the way I combined and transformed data. I modified features of training set. I Combined ID method’s features and score method’s features together as one training set. I also added or deleted features in training method using different way, but, overall, the correctness does not change significantly. I also tried to separate helpfulness score back to numerator and denominator, but things still stay the same. I will reach out to Professor and Ta for further enhancement.