Kaggle Competition Report

Prompt:

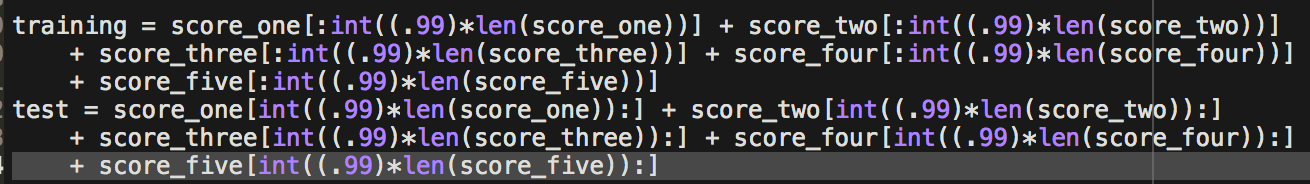
In addition to submitting your solution online, you need to provide us with a 2-page write-up that describes the algorithm you have implemented and the special tricks you used in order to make it work (or improve). It is important that you show us your thought procedure. Also, describe your strategy for selecting that particular algorithm and how you did your offline evaluation. Note that some sort of offline evaluation is required. Your write-up should not exceed 2 pages under any circumstance, else it will not be graded.

General Ideas:

The primary focus of this competition is to build a system that predicts the star rating associated with reviews in Fine Foods on Amazon using available features. The algorithm that I implemented is heavily based on NLTK package in Python that performs natural language processing analysis. My intuition tells me that I can use NLTK build-in multiclass classifiers to classify star rating into a 1-5 after applying NLP analysis on the “Summary” of each review. In a nutshell, each summary of the review will generate a rating score 1-5 based its sentiment towards Fine Foods on Amazon. With a list of predicted rating scores and ground truth, I can then proceed to use RMSE metric to measure my accuracy.

Offline Evaluation:

I conducted a few offline evaluations each time before I submitted my test file on Kaggle because I believe algorithm performance should be consistent across different testing sets if these sets are large and sampled from the same source. Since we have a fairly large dataset with 460804 rows of observation after dropping rows with NA rating scores, I divide such dataset into two parts: 99 % training and 1 % testing. Each part consists of samples with known rating scores from 1-5 so that my algorithm is robust. Ground truth is basically a list of known rating score in the testing set, which is a benchmark used to compare with my prediction. I choose RMSE as my error evaluation metric and the smaller RMSE is, the more accurate my prediction becomes.



../../../../Screen%20Shot%202017-12-04%20at%203.13.16%20PM

../../../../Screen%20Shot%202017-12-04%20at%202.40.22%20PM

Detailed Procedures:

I think the most interesting part of this project is data preprocessing. NLTK has a build-in word tokenized method that can split a string of summary into a list of words and punctuations. NLTK will then use different classifiers to train these sentimental words and output a rating of 1-5 based on the summary of each review. I have tired different classifiers such as DecisionTree Classifier, MaxentClassfier, and NaiveBayesClassfier. DecisionTree Classifier took more than 30 hours and never gave me any output. NaiveBayesClassfier worked naively and only able to achieve a medical RMSE score of 1.05. Among these three, MaxentClassfier worked out the best as it hit a RMSE score of 0.98747 with 60 iterations. I can imagine that with more iterations, MaxentClassfier can outperform my current result.

Future Improvements:

I believe there a variety of methods that can improve my accuracy score or minimize my RMSE score. For instance, in the data preprocessing part, I could have taken out stop words in each summary so that my NLTK classifier will focus on extracting information from more useful words than from less useful words. It will not only make predict more accurate, but also lessen processing time in training. Another potential improvement could be increase number of iterations in MaxentClassfier in NLTK package. I am convinced that the prediction accuracy increases as max\_iterations increases.

