Reflection

The goal of this project was to successfully predict the star ratings of restaurants as 1, 3 or 5 stars based on its reviews. The first step I took was to make sure any classifiers that were going to be implemented have an extension to multi-dimensional counterparts since this is a multi-class classification problem. A multinomial Naïve Bayes classifier was used as a benchmark as it is simple to use, and it grows linearly with the number of instances. Hence, Naïve Bayes was used as an indicator to test time and space complexity. Since the training dataset had over twenty-eight thousand features a multinomial variant Logistic Regression classifier was used as theoretically it should perform well under large N. Finally, a Stacking classifier was used combining Naïve Bayes, Logistic Regression and a Random Forest classifier. This was to maximise performance and yield the highest accuracy possible. I used Random Forest as it is robust to over fitting and it generally performs well and is very efficient.

The models that were built performed well as the Naïve Bayes, Logistic Regression and the Stacking classifier yielded accuracies of 68.41%, 87.22% and 87.39% respectively on a 75-25 train test split of the data. I am satisfied with the results of the models. Pre-processing of the data improved the performances of the classifiers. Pre-processing included stripping the features of all punctuations and non-word characters, the data was converted to lower case and lastly lemmatization. I am satisfied with how I cleaned the data; however, an improvement could be made by tuning the hyperparameters of the models whilst keeping the models generalisable and to score the highest score on the validation set. Two methods for encoding text were considered (Frequency Counts and Term Frequency Inverse Document Frequency), after testing classification accuracy for both vectorization methods, I used Frequency Counts for Naïve Bayes and Logistic Regression and Term Frequency Inverse Document Frequency for the Stacking classifier. I am happy with this choice as it served to improve the performances of the models. Comparing the performances of the Stacking classifier across the 75-25 train test split, 30% Kaggle dataset and 70% Kaggle dataset of 87.39%, 86.70% and 87.97%, it is solid throughout all the tests. I am extremely pleased by the results of the Stacking classifier as I was 26th in the Public Leaderboard and 18th in the Private Leaderboard.

In conclusion, I enjoyed building the models to predict star ratings of restaurants based on sentiment analysis. I would have liked to include some feature selection and to tune the hyperparameters, but I am satisfied with how the models (particularly the Stacking classifier) performed.