COMP30027 Machine Learning - Project 2

Self-Reflection

Tuan Khoi Nguyen - 1025294
Word Count: 591

May 25, 2021

Recently, our Project 2 of classifying recipe's cooking duration met its deadline. Large amount of work has been put in to help it reaches completion. This self-reflection will describe the process of building our optimal model, consisting of how we planned our strategy, the challenges we faced, as well as accomplishments we have made, along with their potential improvements.

For this project, we worked as a group of 2. My teammate started with evaluation and visualisation of the provided features. This has later helped us to realise the lack of representation of Label 3.0 recipes, as well as CountVectorizer being the most efficient text processor for all text attributes. Then, I did research for the suitable models, trying to find sets of classifiers that has varied level of bias and variance. Research results included discovery of XGBoost, a powerful boosting algorithm that minimizes variance, as well as a diverse combination of base models for stacking method, which had helped us reaching the top on Kaggle leaderboard.

The process then moves on to improvement, where further steps of modifying the data or tuning inputs are carried out. My teammate continued the search on text processors, changing interest from best individual to optimal combination. Meanwhile, I observed special keywords from steps feature, then attempted to retrieve and incorporate them as a newly engineered feature – the time to finish the recipe. On the process, I have come up with Keyword Assign – a method that assigns specific words a value, which is successfully implemented in our model.

With optimal combination and engineered feature, model is efficiently improved. Noticing that feature space is large, feature selection is carried out. On the same models and dataset, we each tried different percentiles to set as the range of features to take. With each percentile taking approximately 15 minutes to run, 9 percentiles are tested on, ranging from 10 to 90% of the data. We found out 50% is optimal for our combination, which helped us further improved our accuracy by a small percentage. We also tried hyperparameter tuning using Randomized Search to see if it can help improving the model. However, best model from the search showed a performance decrease. Therefore, we finalized using our Top 50% Feature Selection as selected model for submission on Kaggle.

In general, for basic tasks, I am satisfied with our data evaluation and error analysis, with data characteristics being fully exploited for analysing. I am also happy with my feature engineering, believing that the method is most suitable for project dataset. However, there were few potentials that we could not implement, or incorporate into report, due to the timing and word constraints. So far, we have our ensembles of 3 base classifiers. With more research and word allowance, this could be extended to a larger number, giving the model more cases to consider. With feature selection, we can only try 9 percentiles on 1 combination so far. There is potential that different percentiles on different combinations can yield better results, but being able to test this requires further look in data characteristics, as well as computational resources to run the tests on. For hyperparameter tuning, where countless parameter combinations are possible, more searches will be needed to find a better parameter combination. All these processes will require more time to develop, as well as words to explain the concepts comprehensibly.

Overall, I believe our model implementation have satisfactory performance, with observations fully exploited, and optimal feature engineering. Further improvement will be possible when further resources of time, computation and research are given.