COMP30027 Project 2 Report

Anonymous

1. Introduction

The goal of the task is to predict the cooking time of the recipes, where each instance contains two numeric features (n_steps, n_ingredients), three text features (name, steps, ingredients), and one categorical label (duration_label) with three possible levels (1: quick, 2: medium, 3: slow). This report would mainly be divided up into four parts. Firstly, a detailed description of pre-processing which includes data separation, feature engineering, and model selection. Secondly, evaluation of these models based on different metrics. Thirdly, a detailed discussion that involves the interpretability and critical analysis of model behaviours. Finally, some error analysis of models.

2. Pre-processing

2.1 Data Separation

There are 40,000 instances in the training set, and 10,000 instances without class labels in the test set. Therefore, a validation set would be required to test the model performance. Since the training set is considerably large, using cross-validation might be computationally expensive. As a result, the **hold-out strategy** is chosen instead to split the original training set into a new training set with 30,000 instances and a validation set with the same size as the test set.

2.2 Feature Engineering

The pre-processing in this task can be summarized as normalization, vectorization, and feature selection. For example, **MinMaxScaler** is used to normalize the numeric attributes to a range between 0 and 1, while **CountVectorizer** is selected to vectorize the text features in order to be fitted into machine learning models. However, one of the issues related to the CountVectorizer is that it may **generate redundant features that are unrepresentative in terms of the whole model** since each unique word/word combination will correspond to an attribute. As a result, it is necessary to filter features and obtain robust predictive ability using some techniques such as **Variance Thresholding** and **SelectKBest** which eliminate attributes with low variances (below 0.001) and select the remaining k best features according to a statistical test (chi-square in this case) respectively. Finally, **5,200 out of 370,840** features are selected for model constructions.

2.3 Model Selection

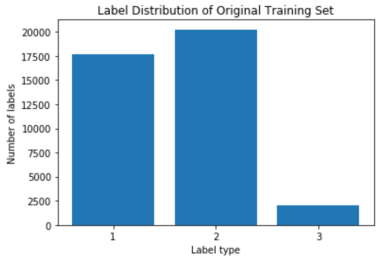
The first supervised classifier to construct is the **multinomial logistic regression** which might be suitable for multi-class classification with numeric inputs and categorical outputs. Support vector machine was supposed to be chosen as the second model because of its comprehensiveness. However, **the time complexity of the support vector machine reduced its priority**. As a result, in terms of achieving robust performance, three ensemble learning models (**random forest**, **light gradient boosting machine**, and **stacking**) are selected as the rest of the classifiers.

3. Model Evaluation

This section will mainly compare and analyse the performance of models through several aspects.

3.1 Label Distribution

It is shown by Figure 1 that Classes 1 and 2 consist of approximately 95% of the recipes which indicates **the imbalance of the dataset** that may potentially affect model performance.




```
2.0    20246
1.0    17705
3.0     2049
Name: duration_label, dtype: int64
```

Figure 1- Label distribution of the original training set

3.2 Similarity in Predicted Labels

Table 1 records the proportion of the predicted results that are the same between every two classifiers (exclude stacking) using the same features and testing data to **reflect the dependencies** between models to some extent.

	Multinomial Logistic Regression	Random Forest	Light GBM
Multinomial Logistic Regression	1	0.865	0.8678
Random Forest	0.865	1	0.9128
Light GBM	0.8678	0.9128	1

Table 1- Similarities in predicted labels of distinct models.

It can be seen from above that multinomial logistic regression and random forest have the lowest similarity which indicates that the correlation between these two classifiers may not be significant, whereas light gradient boosting machine and random forest have a similarity score of 0.9128 which might be attributed to **the homologous behaviors of tree-based classifiers**. As a result, multinomial logistic regression and random forest are chosen as the **base classifiers** of the stacking since **the independence of base classifiers is one of the most essential concerns**, and light gradient boosting machine is selected as the **meta classifier** instead.

3.3 Accuracy/Precision/Recall

In order to obtain general results, the testing accuracies are calculated using **5-fold cross-validation** to minimize the error. Moreover, macro precision/recall and weighted precision/recall are applied to gain the average measurements of precision/recall since it is a multi-class classification problem. Table 2 illustrates the corresponding results of different models where the light gradient boosting machine achieves the highest accuracy (0.8095), followed by stacking with an accuracy of 0.8039, and multinomial logistic regression has the lowest score (0.7886). The reasons that may result in such a situation will be discussed later.

Model Name	Testing Accuracy	Macro Precision	Macro Recall	Weighted Precision	Weighted Recall
Multinomial Logistic Regression	0.7886	0.7872	0.7261	0.7879	0.7880
Random Forest	0.7933	0.8504	0.7112	0.8076	0.8038
Light GBM	0.8095	0.8198	0.7441	0.8145	0.8131
Stacking	0.8039	0.8022	0.7461	0.8047	0.8039

Table 2- Some basic measurements of the models.

Another noteworthy point is that the difference between macro precision and recall is much larger than that of weighted precision and recall which indicates that one of them may be biased. Macro averaging tends to assign each prediction with similar weights, while **weighted averaging will assign weights based on the proportion of the data**. As mentioned in Section 3.1 that the dataset is imbalanced, it can be argued that **using macro-averaging may lead to biased results**.

3.4 ROC Curve

ROC curve is built based on confusion matrix, which has FP rate as the horizontal axis and TP rate as the vertical axis. In general, a steep ROC curve is demanded for an excellent classifier since a high TP rate and low FP rate are often desired. Moreover, AUC is the area under the ROC curve, it generally indicates the **measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve**. A larger AUC generally means better model performance.

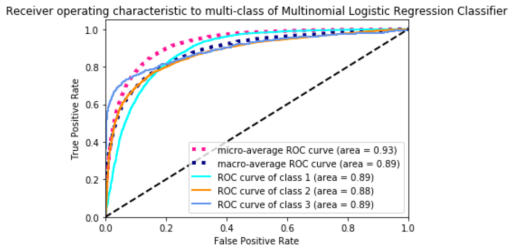


Figure 2- Receiver operating characteristic curve to multi-class of Multinomial Logistic Regression Classifier.

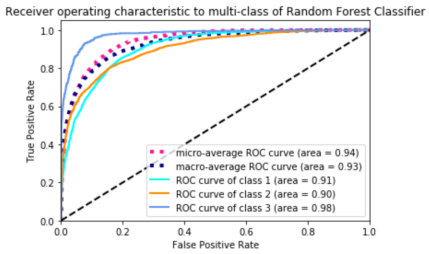


Figure 3- Receiver operating characteristic curve to multi-class of Random Forest Classifier.

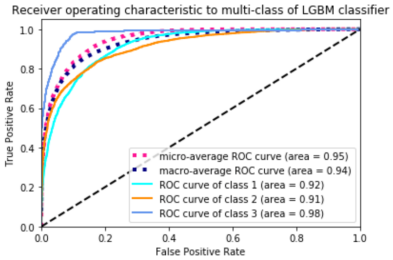


Figure 4- Receiver operating characteristic curve to multi-class of Light Gradient Boosting Machine.

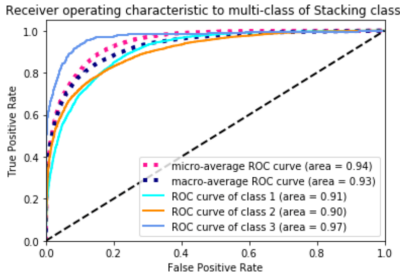


Figure 5- Receiver operating characteristic curve to multi-class of Stacking classifier.

As can be seen from the plots (Figure 2, 3, 4, 5), all of them display a similar pattern that AUC of Label 3 and Label 1 obtain the highest value. This might be because of **the over-optimistic characteristic of the ROC curve to imbalanced data (Figure 1)**. Apart from that, all of the classifiers show high predicting performance with light gradient boosting machine achieves a slightly higher score in both micro-averaging and macro-averaging performances.

3.5 Time Complexity

Model Name	Multinomial Logistic Regression	Random Forest	Light GBM	Stacking Classifier
Execution time (s)	7.93s	301.20s	103.2s	867.286s

Table 3- Execution time for different models (Note: execution time might vary when executing)

Table 3 shows execution time for different models to predict cooking time labels using 5-fold cross-validation respectively. It could be witnessed that the Stacking classifier is the slowest, with a value of 867.3s. It might be due to **the nested cross-validation strategy of stacking**. Also, the fastest is multinomial logistic regression, with only 7.93s execution time.

4. Interpretability & Critical Analysis

4.1 Multinomial Logistic Regression

Multinomial logistic regression uses one VS rest logistic regression classifier. For each logistic regression classifier, it uses a sigmoid function to transform features to acquire a probability value. There are some advantages of multinomial logistic regression. Firstly, the training speed of the logistic regression classifier is relatively fast. As indicated before (Table 3), the logistic regression classifier shows the most efficient training time, which might be **due to simple transformation from linear regression to logistic regression**. This result matches the theoretical properties of the multinomial logistic regression classifier that its time complexity is only $O(C^2D)$ where C is the number of different class labels and D is the number of features. Nevertheless, multinomial logistic regression could not solve complex problems due to its simple transformation, unlike neural networks or other gradient boosting machines. By Table 1, even though logistic regression shows efficient training time, its accuracy is slightly less than that of the Light Gradient boosting machine.

4.2 Random Forest

It can be argued that random forest is likely to obtain a strong performance in this task since the bootstrap aggregating approach generally performs better with more data and the combination of a significant amount of relatively **uncorrelated** base models would possibly outperform any of the individual models. It is proved by Table 2 that random forest gains a decent score for accuracy, weighted precision, and weighted recall. The reason behind this may be **due to the stagger distribution of errors** in each tree such that the voting strategy could **minimize the total error** of the model. Furthermore, because of this property, **the predicting result of random forest is more stable** than normal decision trees since small changes in the training set would not lead to a significant variation in predictions. However, **although random forest should not be prone to overfitting**, the training accuracy is still much higher than the testing accuracy (almost approaches to 1). The choice of hyperparameters may be the most likely reason that results in the overfitting of the model.

4.3 Light Gradient Boosting Machine

Light GBM is the abbreviation of light gradient boosting machine which is a type of ensemble model. To be specific, a boosting classifier. The main property of this type of classifier is to use iterative sampling and weighted voting to minimize instance bias and therefore obtain a high classification accuracy. Light GBM displays **not only accuracy but also efficiency** among all classifiers. From Table 2 and Table 3 above, it is obvious that light GBM achieves a high accuracy (roughly 0.81) and efficient training time (103.2s), especially for a huge dataset (40000 training samples and 5200 features selected). This matches the theoretical expectation that light GBM is fast and accurate particularly for huge datasets. Interestingly, light GBM tends to overfit,

due to its characteristics to minimize instance bias during training.

```
LightGBM Model accuracy score: 0.8095
Training accuracy:  0.9159333333333334
Mean Absolute Error of Training: 0.08803333333333334
Mean Squared Error of Training: 0.09596666666666667
Root Mean Squared Error of Training: 0.3097848715910231
Mean Absolute Error of Testing: 0.199
Mean Squared Error of Testing: 0.2232
Root Mean Squared Error of Testing: 0.4724404724407087
```

Figure 6- Evaluation metrics of Light Gradient Boosting Machine classifier.

However, in practice, there is no strong evidence to prove light GBM is overfitting. It is suggested that the training accuracy of this task is approximately 0.9181, which has a small difference between accuracy using cross-validation (Figure 6). The reason behind this might be due to the use of regularization term to penalize if the classifier tends to overfit.

4.4 Stacking

In general, **the results of stacking would be slightly better than any other base classifiers**. For instance, from Table2, it is clear that there is an enhancement of the testing accuracy in the stacking compared with the multinomial logistic regression and random forest. This is probably because that the meta-classifier could **enlarge the strength** and **reduce the variance** of each base model by a voting strategy. Nevertheless, time consumption might be one of the issues of the stacking, especially dealing with a large amount of data. It can be directly proved by Table 3 that the execution time of the stacking is longer than the sum of the other three classifiers. Another issue related to the current stacking model may be **the insufficient number of base models**. The accuracy is supposed to be further improved if there is an increase in the number of base classifiers that contain considerably strong predicting performance since the variance might be further decreased.

4.5 Hyper-parammeters Tunning

Multinomial logistic regression has the least hyper-parameters to tune. It is found that reducing the regularization term C can lead to better performance. As a result, C is reduced from the default value (1.0) to 0.006. Moreover, to analyze stable results, the random state has been changed to 0. Compared with multinomial logistic regression, the number of hyper-parameters of the random forest has been increased. For example, the number of estimators and the max depth of each estimator. In this assignment, the random forest has 400 estimators and an unlimited number of depths to completely fit the model. However, this might result in overfitting as discussed in Section 4.2. The model that occupies the majority of the time tunning hyper-parameters should be light GBM. In order to find the global optimum, the number of iterations is increased from 300 to 415 and the learning rate is dropped from 0.1 to 0.085. Furthermore, to reduce overfitting, the maximum number of bins is decreased from 415 to 50. As a result, the testing accuracy has been improved from 0.8056 to 0.8095.

5. Error Analysis

5.1 Identifying

Figures 7, 8, 9, 10 demonstrate the confusion matrices of multinomial logistic regression, random forest, light GBM, and stacking respectively. It can be seen from these figures that light GBM and stacking are the two best-performing models in terms of the TP, FP, and FN of each class. By contrast, some issues can be found in the multinomial logistic regression and random forest. For example, although the FP cases in Class 3 of the random forest have been minimized compared with others, **the overall distinguishability of Class 3 is still the worst**. Furthermore, **correctly identified cases from Class1 in the multinominal logistic regression confusion matrix are the least** among all four classifiers.

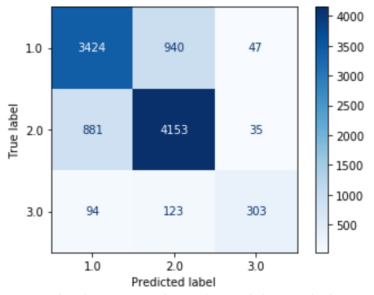


Figure 7- Confusion matrix of Multinomial Logistic Regression

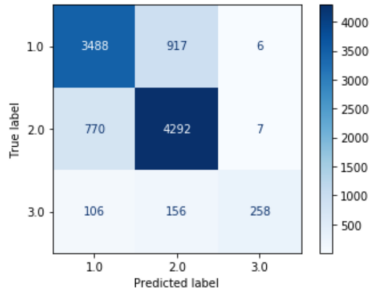


Figure 8- Confusion matrix of Random Forest

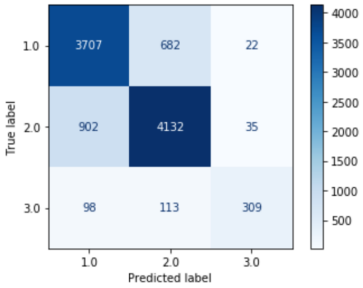


Figure 9- Confusion matrix of Light Gradient Boosting Machine

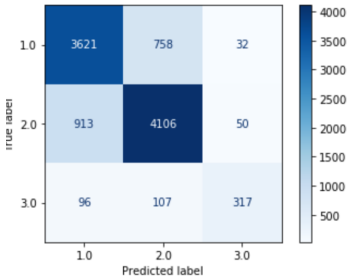


Figure 10- Confusion matrix of Stacking

5.2 Hypothesising & Quantifying

5.2.1 Multinominal Logistic Regression

Due to the weakness to classify Label 1, the hypothesis is raised that this is **mainly because of limited training size**. Therefore, the training is increased to 90% to observe if it would result in the enhancement of classification performance of Label 1 while maintaining all other hyperparameters. For this specific hypothesis, the precision of Label 1 would be used as an evaluation metric.

5.2.2 Random Forest

The errors related to random forest may be **because of the imbalanced data distribution** which tends to classify the result as Label 1 or Label 2 since they occupy the majority of the data. Therefore, modifying the hyperparameters to reduce the overfitting may be unhelpful to solve this issue. To test this hypothesis, **RandomizedSearchCV** is used in this case to randomly select different combinations of hyperparameters in a limited number of runs to decide which set of parameters can maximize the model performance and minimize overfitting within a reasonable time. In this case, min_sample_split is increased from 2 to 5 and the bootstrap option is changed from True to False.

5.3 Feeding

5.3.1 Multinominal Logistic Regression

The precision of the previous multinomial logistic regression model is $\frac{3424}{4411} \approx 0.776$. Nevertheless, the precision of the modified classification is using metrics is $\frac{1349}{1762} \approx 0.765$ (Figure 11). This figure is even worse than the previous model, which might indicate that the **non-ideal performance of Label 1 is irrelevant to the size of training data**. However, due to the limited size of given data, larger data might be required to validate this hypothesis in the future.

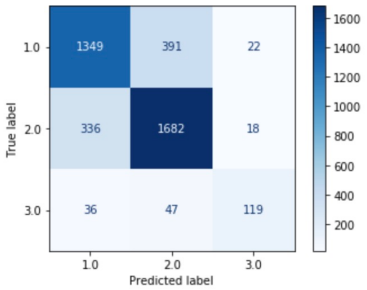


Figure 11- Confusion matrix generated by Multinomial Logistic Regression Classifier after adding more sample sizes

5.3.2 Random Forest

Compared with Table 3 and Figure 8, although a little increase in testing accuracy is witnessed (increased from 0.7933 to 0.7935), **the identified issue is scarcely solved (as seen in Figure 12)**. Therefore, it can be concluded that the hypothesis may be correct.

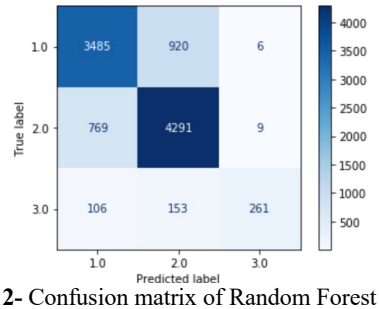


Figure 12- Confusion matrix of Random Forest after using the selected parameters

6. Conclusions

In conclusion, this report has mainly demonstrated the pre-processing steps of this task, assessing the performance, interpretability, and error analysis about relative merits and detriments of various models. It is suggested that even though a comprehensive analysis of the whole task is given in this report, it is still significant to indicate potential improvement. To illustrate an example, during the pre-processing

procedure, **it might be more appropriate to utilize different pre-processing strategies corresponding to different level-0 and level-1 models**. In the future, a more thorough analysis might be conducted to discover potential reasons why instances are classified poorly in some models.

7. References

Majumder, B. P., Li, S., Ni, J. & McAuley, J. Generating personalized recipes from historical user preferences. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.