Machine Learning Project on Natural Language Processing
Predicting the cooking time for recipes

Individual Project

(21 May 2021)

I. INTRODUCTION

Machine learning is a branch of Artificial Intelligence and computer science which focuses on the uses of data and algorithms to imitate the ways that human learns, to solve complex problems. Among the complex problems, this project will aim to predict the cooking time for recipes based on their steps, ingredients and other features. The cooking time of a recipe has been categorised into three classes, corresponding to quick, medium and slow.

II. DATA & FEATURE SELECTION

The cooking time prediction was performed on data from Food.com (Majumder et al. 2019). Each recipe was associated with a corresponding coking time duration label of 1, 2, or 3, as well as 5 attributes (*names*, *n_steps*, *n_ingredients*, *ingredients* and *steps*).

A) Pre-processing

Three out of five features were chosen to train our model as both *n_steps* and *n_ingredients* can be derived from *steps* and *ingredients* respectively. We had excluded punctuation and stop words from the data given. Removing punctuation marks is crucial in NLP processing because it will affect the result of any text pre-processing approach, especially what depends on the occurrence frequencies of words and phrases because in general punctuation marks are frequently used in the text (Etaiwi and Naymat 2017). Apart from that, we also opted to include stop words, as stop words gave no information to the meaning of the text (Singh 2019). Taking into account the nature of the input data, we decided to implement the Term Frequency Inverse Document Frequency algorithm (TF-IDF) for the raw string data given.

B) N-grams

Unigrams (Bag-of-Words) and Bigrams were considered in this project. In the unigrams approach, each of the words represents a unique meaning whereas bigrams generate pairs of words that might represent their meaning. Considering the limitations of unigrams that fail to interpret the meaning from pairs of words such as “small saucepan”, “frozen lemonade” and “sour cream”, bigrams are included to overcome this flaw.

C) Filtering method

Over 580,000 features were generated using TF-IDF. To reduce the computational cost as well as to improve the performance of the models, reducing the number of input features is desirable. We used one of the most popular strategy, the filtering method with χ^2 to reduce the number of columns. We decided to keep the 20% most informative columns (which has the lowest p-values). The resulting number of features is 116,196.

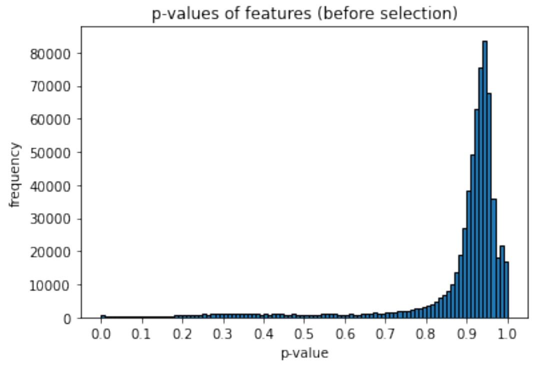

Figure 1.1: P-values of feature (before selection)

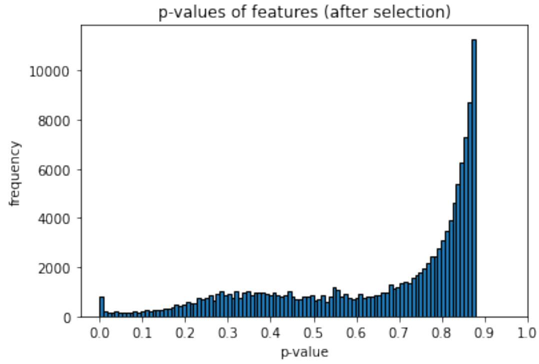

Figure 1.2: P-values of feature (after selection)

According to Figure 1.1 and Figure 1.2, after removing 80% of uninformative columns, the highest p-value of the features is around 0.90, which mean there are approximately 80% of the columns have p-values of above 0.90, which are recognised as ineffectual features

II. MODEL CONSTRUCTION

We implemented 3 different types of supervised machine learning models. We used 5-fold cross-validation to find the optimal hyperparameter for each of the models. 80% of the data were used to train the model while 20% of the data were used to validate the data. Before constructing the models, we established a baseline using zero-R, which achieved 50.615% accuracy.

A) Logistic Regression

We trained logistic regression model using *ridge (L2)* regularisation and *Limited memory Broyden Fletcher Goldfarb Shanno* (*lbfgs*) optimizer. An advantage of logistic regression is that it allows the evaluation of multiple explanatory variables by extension of the basic principles (Hoffman 2019).

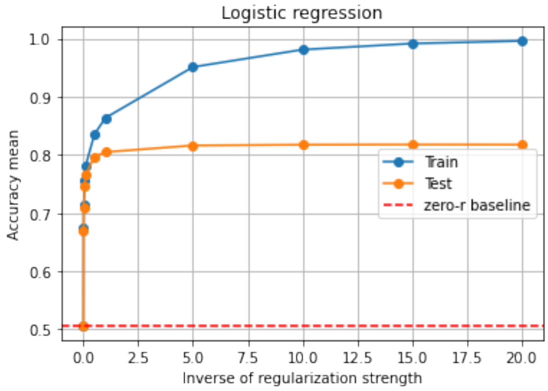

Figure 2.1: Accuracy mean (Logistic Regression)

C	Test accuracy	Variance
0.001	0.5062	2.500-09
0.005	0.6595	1.157e-05
0.010	0.7094	8.550e-06
0.050	0.7452	1.095e-05
0.100	0.7598	1.050e-05
0.500	0.7970	1.162e-05
1.000	0.8049	1.614e-05
5.000	0.8162	2.388e-05
10.000	0.8175	2.221e-05
15.000	0.8178	2.330e-05
20.000	0.8176	2.782e-05

Table 1.1: Logistic Regression accuracy

Figure 2.1 and Table 1.1 shows logistic regression reached the highest accuracy (81.78%) at C = 15.0 and its variance is approximately 0.0000233, which is low.

B) Decision Tree

Decision trees are one of the most popular classifiers in machine learning. It is used for non-linear prediction effectively as it breaks down complex data into more manageable parts. We constructed a decision tree model with *Gini* impurity and tested it with different max depth.

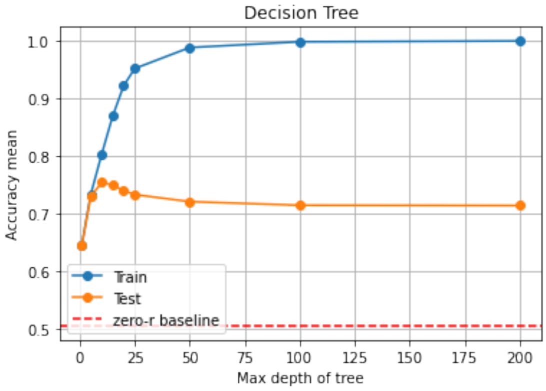

Figure 2.2: Accuracy mean (Decision Tree)

Max depth	Test accuracy	Variance
1	0.6468	3.240e-05
5	0.7300	3.172e-05
10	0.7546	1.858e-05
15	0.7507	1.636e-05
20	0.7401	1.607e-05
25	0.7335	3.069e-05
50	0.7224	8.865e-05
100	0.7169	9.696e-06
200	0.7179	5.877e-06

Table 1.2: Decision Tree accuracy

As shown in Figure 2.2, the decision tree model started to overfit at max depth = 15, and the best score (75.46%) was achieved at max depth = 10. Table 1.2 shows the model has very low variance.

C) Linear SVM

Linear SVM is used for linearly separable data. Ideally, we can use three lines or hyperplanes to separate data points to classify three duration labels. We built the model with *squared hinge* loss function and *One-vs-Rest* strategy and assessed it with different regularization parameters.

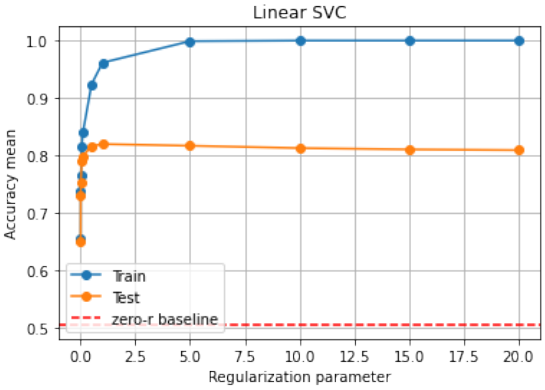

Figure 2.3: Accuracy mean (Linear SVC)

C	Test accuracy	Variance
0.001	0.6516	1.093e-05
0.005	0.7311	1.204e-05
0.010	0.7545	6.484e-06
0.050	0.7900	1.143e-05
0.100	0.7994	1.652e-05
0.500	0.8166	1.976e-05
1.000	0.8201	2.654e-05
5.000	0.8172	1.529e-05
10.000	0.8131	2.065e-05
15.000	0.8108	2.547e-05
20.000	0.8095	2.617e-05

Table 1.3: Linear SVC accuracy

Figure 2.3 shows the model obtained the highest accuracy (82.01%) at C = 1.00. and started to overfit at C = 5.00. The best model has a low variance of 0.00002654.

IV. EVALUATION

A) Comparison

Model	Best accuracy
0-R	50.62%
Logistic Regression	81.620%
Decision Tree	75.46%
Linear SVC	82.01%

Table 2.1: Model accuracy

According to Table 2.1, logistic regression and linear svc significantly outperformed than 0-R and decision tree.

B) Error Analysis

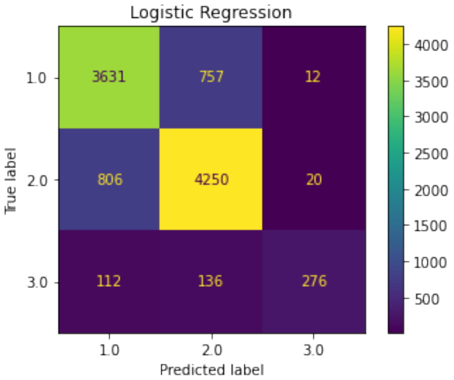

Figure 3.1: Logistic regression confusion matrix

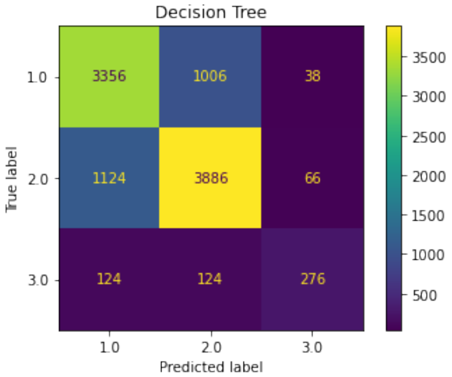

Figure 3.2: Decision tree confusion matrix

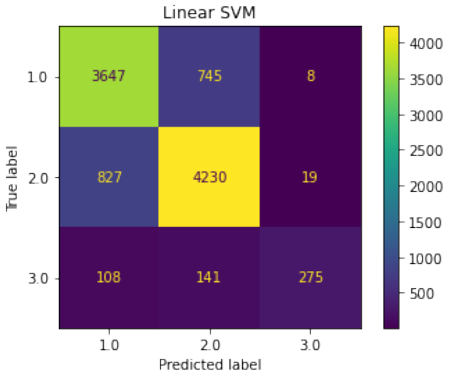

Figure 3.3: Linear SVM confusion matrix

The confusion matrix plot described the performance of all 3 models. As shown in Figure 3.1, Figure 3.2, and Figure 3.3, all 3 models could not predict well for 3.0 duration label. Furthermore, all of the 3 models were affected by bias as the models overpredicted 2.0 and 3.0 duration label. The potential reason is because the class label is unbalanced, where the data truly labelled as 3.0 is significantly less than 1.0 and 2.0. Thus, there are not enough data in 3.0 duration label to train the models.

V. LIMITATIONS AND CONCLUSION

In this research, logistic regression, decision tree and linear svc were constructed to predict the cooking time. Among the models, logistic regression and linear svc performed the best in this task. With more computing power, we might be able to further optimise the models by testing every possible hyperparameter for each model.

VI. REFERENCES

1) Majumder, BP, Li, S, Ni, J & McAuley, J 2019, in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).
2) Etaiwi, W and Naymat, G 2017, The Impact of applying Different Preprocessing Steps on Review Spam Detection, *Procedia Computer Science 113 (2017) 273–279*, p. 275.
3) Singh, S 2019, ‘stopwords might not add much value to the meaning of the document’, viewed 16th May 2021, https://www.analyticsvidhya.com/blog/2019/08/how-to-remove-stopwords-text-normalization-nltk-spacy-gensim-python/
4) Hoffman, JIE 2019, in Basic Biostatistics for Medical and

Biomedical Practitioners (Second Edition).