COMP30027 Assignment 2 Report

1. Introduction

Twitter is an open online social media platform where users can access various types of information and share their emotional feelings with others. However, emotions can be easily expressed and accepted by another person, but difficult to monitor and understand by software.

The aim of this project is to build and critically analyse supervised Machine Learning methods to predict the sentiment of Tweets, classified as Positive, Neutral or Negative. The usefulness of these predictions can range from the public's general attitude towards a topic to monitoring an individual's interests and mental health.

Data used in this NLP model is from Rosenthal (2017), containing 21802 training samples and 6099 testing samples. Training samples contain twitter IDs, tweets, and sentiment, and testing data does not include sentiment. As seen in Figure 1, the distribution of training data is highly uneven, as it contains 12659 neutral instances, 5428 positive and 3715 negative instances. This suggests that models produced based on this data could be biased towards the majority class (neutral).

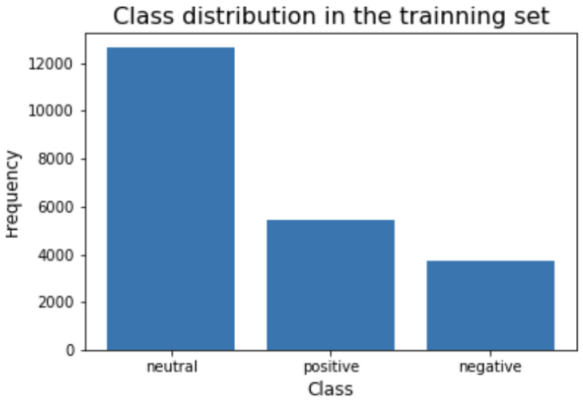

Figure 1 *- Class distribution in the training set.*

2. Method

2.1 Data cleaning

The raw data is a tweet in the form of plain text, which contains various pieces of information, including noises that are irrelevant to sentiments. Thus, it is necessary to clean the data before using them as features. Firstly, it is important to keep the words with identical meaning consistent in format to reduce the word variant by applying the following steps.

1. Convert all numeric in words to numbers (e.g. “twenty” to “20)
2. Convert all text to lower cases
3. Replace all elongated words (e.g. “gooood” to “good”)
4. Expand contractions (e.g. “they’re” to “they are”)
5. Lemmatize words to its base form according to its part of speech (Jonathan, 2016)

Secondly, removing all the noises that are irrelevant to sentiment, including URL links, usernames, numbers, punctuations, stopwords, meaningless single letter and non-ascii letters. This will reduce the feature size and only focus on the important words (69285 words to 24892 - a substantial 64% decrease of nuisance features).

2.2 Vectorisation

The cleaned data is still in the form of plain text which is unrecognisable by machine. Therefore, vectorisation is introduced to convert human readable data (text) to machine readable data (features).
Two different methods of vectorisation were approached, including Bag of Words(BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). The difference is that Bag of Words computes a set of vectors based on its occurrence whereas TF-IDF selects vectors based on its importance.
Each model was tested with both methods and the one which performs better was selected and will be specified at **2.4 Models** section.

2.3 Feature selection

In order to further eliminate the irrelevant words and reduce the feature size, k-best selection is applied to select the top k number of features (SelectKBest) based on its Chi-square scores. Instead of making a random choice of k, we introduced a parameter named "k_ratio" so that

$$k = ratio_k * Number\ of\ total\ features$$

The benefit of this parameter is that it gives a more intuitive idea of the percentage of features being used.

Other common feature selection methods were also considered, including SelectKBest-F-test, SelectKBest-Mutual Information and variance threshold. However, calculating mutual information is time-consuming, the variance threshold is hard to choose properly and thus does not perform well, and the F-test performs basically the same but slightly worse than the Chi-square. Consequently, SelectKBest and Chi-square scores feature selection were used for all of the models.

2.4 Models

There were six different kinds of classifiers which were generated and would be evaluated in this study, including a 0-R baseline model, multinomial naive bayes, one vs rest support vector machine, logistic regression and two ensemble learning models, random forest and stacking. The description of the methods and the choice of hyperparameters are detailed below.

Zero-R Baseline Model: It uses majority voting and predicts every instance in testing set to be the most frequent label which is "Neutral".

Multinomial Naïve Bayes (MNB): This is a simple model based on the estimation of joint probability distribution of observing a feature (word), which assumes that each feature is conditional independent from other features given the class. Laplace smoothing was applied with $\alpha = 1$. After comparing and adjusting parameters in feature selection, MNB model performs best with Bag of Words vectorization with unigrams, with k_ratio = 0.4 in feature selection.

One vs Rest Support Vector Machine (SVM): Ideally, this classifier aims to find a hyperplane which classifies classes correctly based on the assumption that classes are linearly separable. However, the assumption is almost impossible to achieve for this large dataset(Chen, 2019). Therefore, a soft margin of 1 and a linear kernel is applied to handle the nonlinearity. The choice of C is low (1) because for a complex dataset like this, we prefer a large margin hyperplane even if it has more errors. The choice of kernel is decided to be linear since that linear and RBF are the top 2 kernel choices for text classification (the best kernel has been controversial) (Martin, 2012) and linear kernel is also the fastest amongst all available kernels and preferably used on ultra large datasets (Lakshmana Phaneendra Maguluri & R Ragupathy, 2019). Considering the fact that SVM itself has a high time complexity, it is more cost-effective to use a linear kernel method.
The final SVM classifier uses TF-IDF vectorisation, chi-square score for k-best feature selection with k_ratio of 0.6, a soft margin C of 1 and a linear kernel.

Multi-Class Logistic Regression (LGR): This classifier estimates the probabilities of instances associating with binary variables (predicting class vs rest of the class). There are two methods which are suitable for large multi-class datasets, - SAGA and SAG solver, of which "SAGE" is a variant of SAG which supports the non-smooth *penalty L1* option. Since we decided to use the default penalty *L2* option, SAG is used as a solver. Finally, the LR model shows the best performance with TFIDF Vectorization with unigrams, k-best-chi feature selection with k_ratio of 0.6, margin of 1 and multi_class set to multinomial.

Ensemble model - Random Forest (RF): Instead of using a single decision tree, we decided to use random forest to reduce the probability of the model being overfit. This model constructs 200 (n_estimators) decision trees at training and predicts the label as the class selected by most trees. Each tree splits nodes based on information gain (entropy), randomly selects square-root of the original features size(max_features="sqrt"), and has a maximum depth, minimum samples split and minimum samples leaf.
The choice of splitting criterion and maximum features did not influence the model performance significantly. Increasing n_estimators might increase the model performance but is also time consuming. For each tree, an overly large maximum depth or an overly small minimum sample leaf of a single tree will cause the model to overfit the training data whereas a tiny maximum depth and a large minimum samples leaf will cause the model to underfit the training and thus perform badly on the testing set.
After adjusting hyperparameters and testing the model performance, it was decided that the final random forest model uses "TFIDF" vectorisation with ngram=(1,1), and "k-best-chi" as feature selection method with k_ratio of 0.6, n_estimators=200, criterion="entropy", max_depth=150, min_samples_split=10, min_samples_leaf=1, max_features="sqrt".

Ensemble model - Stacking: This classifier trains each of the previous models (except baseline model) as base classifiers and uses their predictions to construct a new attribute, and then uses logistic regression (LGR) as a meta-classifier to make the final prediction. The base classifiers include MNB, SVM, LGR, RF with their best performing hyperparameters. TFIDF vectorisation with k_ratio of 0.6 was used for stacking since 3 of the base classifiers used it, whereas only MNB uses BoW vectorization with k_ratio of 0.4. This might decrease the performance of the MNB classifier and consequently decrease the performance of the stacking model (which will be discussed later in **4. Discussion** section) . However, it is still expected that the stacking classifier would improve modelling performance, although it is not guaranteed to result in an improvement in all cases (Brownlee, 2020).

2.5 Evaluation Methods and Metrics

As mentioned before, the sentiment labels are not given in the testing set. Thus, it is necessary to generate a **validation set** which allows us to evaluate the performance of each model. The original training set was randomly splitted into 2 parts, while keeping the size of testing set consistent with the original testing set (6099 instances). Since there are 21802 instances in the original training set, the test_size ratio is $6099/21802 \simeq 0.28$. Consequently, 72% of the original training set was used to train the model and the remaining 28% was used as a validation testing set to check the model performance. The class distribution of validation set is shown in Figure 2.

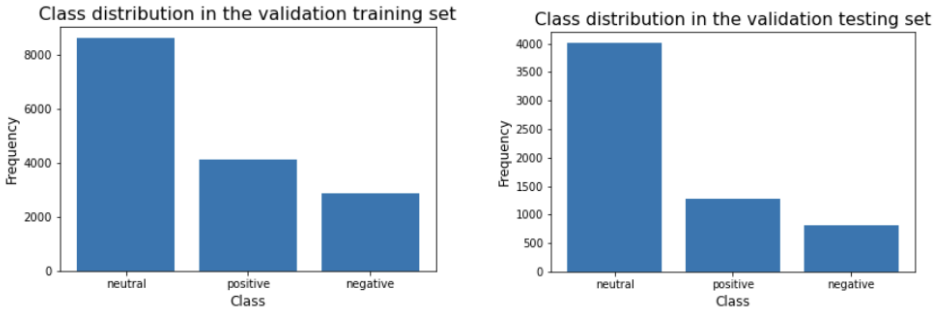

Figure 2 *- Class distribution in the validation set.*

Since the classes in the both training dataset and sub_training set distribute unevenly towards the “neutral” class, it is more suitable to evaluate the model performance through weighted averaging metrics based on each class.

Therefore, accuracy, weighted precision, weighted recall and weighted F1-score were being used as evaluation metrics for model performance. Besides the performance of the model, the time complexity is also important for such a dataset with a large number of instances and dimensions. Therefore, the time taken for each model to complete the procedure of vectorisation, feature selection, training, prediction and saving the prediction to a csv file are also recorded and taken in account to model evaluation.

3. Results

Model	Accuracy	weighted precision	weighted recall	weighted f1 score
0-R	0.6571	0.4318	0.6571	0.5212
MNB(BoW)	0.6737	0.6507	0.6737	0.59
SVM(TFIDF)	0.6765	0.6628	0.6765	0.6671
LGR(TFIDF)	0.6727	0.6575	0.6727	0.6621
RF(TFIDF)	0.6809	0.6508	0.6809	0.6395
Stacking(TFIDF)	0.6603	0.6612	0.6603	0.6607
Stacking(balanced dataset)(TFIDF)	0.6822	0.6788	0.6822	0.679

table 1 - *table showing 4 raw evaluation metrics results for all models*

Note: *The results from RF and Stacking model will vary every time due to randomness of random forest model.*

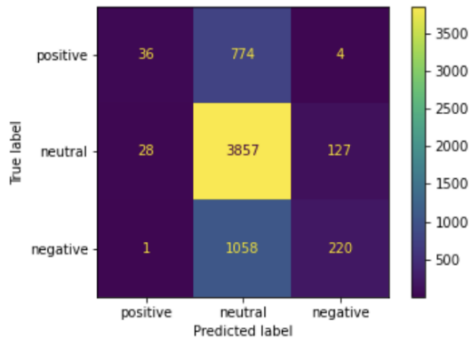

Figure 3 **-** *Confusion matrix for MNB prediction*

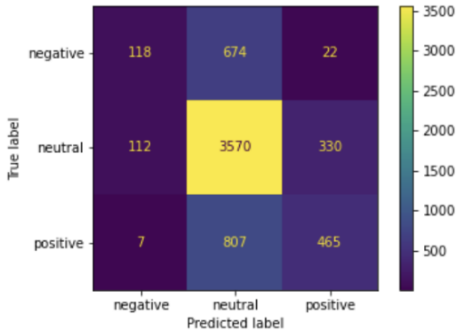

Figure 6 **-** *Confusion matrix for RF prediction*

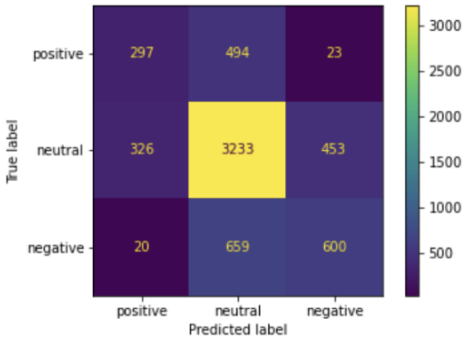

Figure 4 **-** *Confusion matrix for SVM prediction*

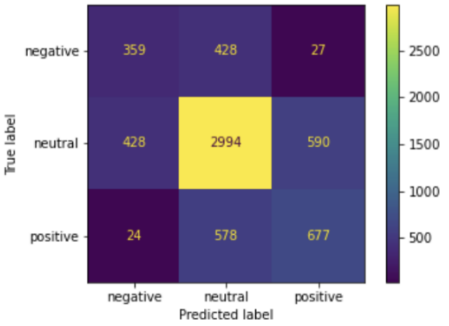

Figure 7 **-** *Confusion matrix for Ensemble Model-stacking prediction*

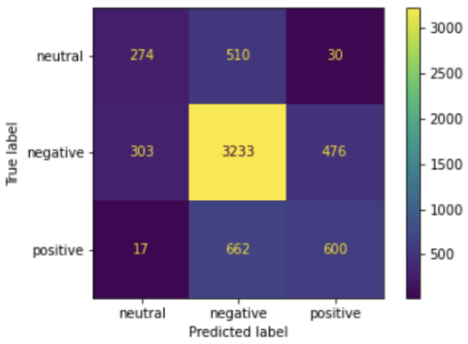

Figure 5 **-** *Confusion matrix for LGR prediction*

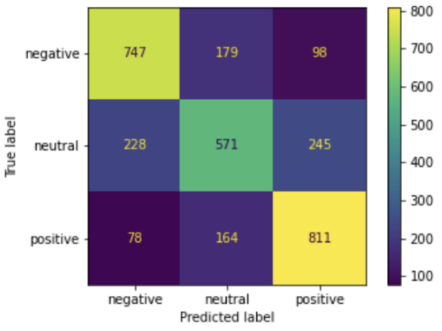

Figure 8 **-** *Confusion matrix for Ensemble Model - Balanced Stacking prediction*

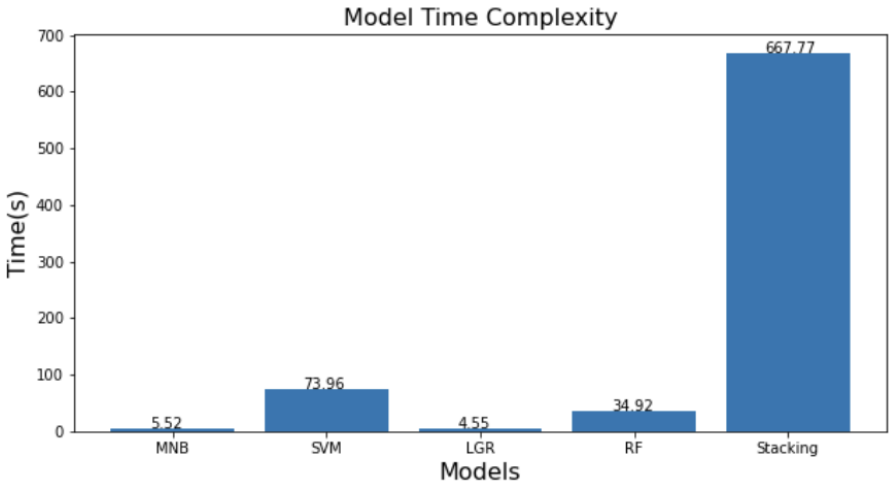

Figure 9 *- A bar chart compares the time taken for each model to complete the prediction process.*

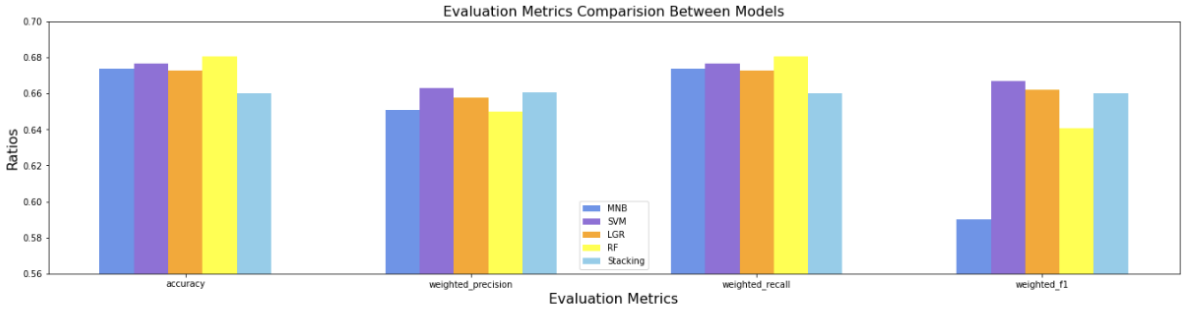

Figure 10 *- A bar chart which compares 4 evaluation metrics from 5 models, including accuracy, precision, recall and f1- score based on the validation set.*

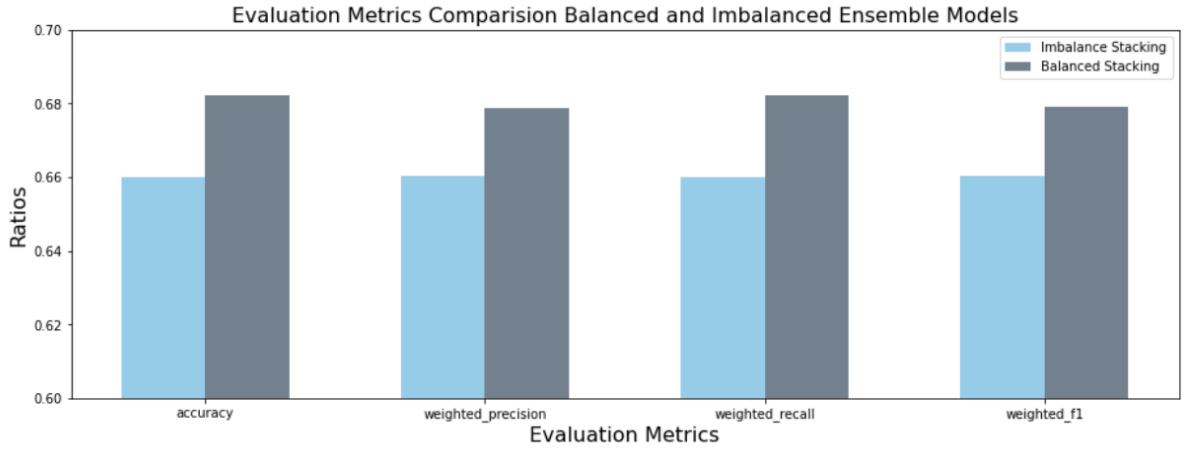

Figure 11 *- A bar chart compares evaluation metrics of stacking with imbalance and balanced dataset*

4. Discussion / Critical Analysis

MNB

As seen in Table 1, despite obtaining an accuracy of 67.37%, precision and recall over 65% on training data, MNB's problem is not neglectable as it has a low 59% F1 score. The imbalanced data has a great influence with MNB's performance due to its algorithm. Naive Bayes predict sentiment class based on the product of calculated probabilities of features in a tweet, which is very small. Prior for "neutral" under this condition has a strong effect on probability since it's significantly larger than the other two (Figure 2). This is shown as the majority of the accurate prediction is the more weighted "neutral" sentiment in the dataset (Figure 3). MNB also assumes feature independence, which is also hard to achieve for a large dataset, weakening the overall model performance. Hence, the MNB model is inadequate to fit the imbalanced dataset, as it's likely to overfit testing data.

SVM & LGR

SVM works by finding a hyperplane to distinguish training data, and LGR can handle imbalance datasets through cross-entropy loss function, therefore the two models are less sensitive to data distribution compared to MNB. Both SVM and LGR obtained similar evaluation metrics in prediction as seen in Table 1, with all evaluation metrics over 65%, and showed a similar prediction trend (figure 4 and 5).

RF

Theoretically, RF is suitable for dealing with the high dimensional noisy data in text classification (Islam, 2019) and improves the limitation of overfitting in a single decision tree since it selects the features at random. However, the actual result shows that RF holds the second lowest f1-score which suggests the imbalance between its recall and precision. As seen in Figure 10, the RF model has the lowest recall but the highest precision amongst all the models. This is likely caused by two reasons. Firstly, the unevenly distributed classes in the training model cause the model to overfit the training data and favour the "neutral" class. This is supported by Figure 6 as the high accuracy is mostly due to more "neutral" prediction. Secondly, the basis of the RF is decision trees which are naturally prone to overfitting (Samadi & Luo, 2022). This implies that RF is likely to overfit test data more than SVM and LGR, making it less practical for predicting sentiments.

Stacking

Stacking classifiers is expected to perform as good or better than individual models, yet, disappointingly, it has the lowest accuracy in training data. Hypothetically, the stacked ensemble model would reduce model bias and obtain a higher accuracy as it learns base model's output through nested cross validation. The reason for this result is believed to be caused by the inconsistent vectorization method and the uneven distributed-class training set.

As mentioned before in **Method**, TF-IDF with k_ratio of 0.6 was used for stacking model whereas our MNB model performs the best with BoW vectorization with k_ratio of 0.4. MNB with TF-IDF vectorization does not perform as well as the individual MNB, which consequently diminishes the performance of our Stacking model.

All the base classifiers significantly favours the Neutral class due to unevenly distributed dataset. They are not considered as reliable since the high accuracy was a consequence of overfitting to an unevenly distributed dataset. Therefore, it is suggested that the stacking model is the most reliable model in our study.

Further Improvement

To testify our best model and further improve its performance, we hypothesised that if we resolve the issue of imbalance dataset, the accuracy of stacking classifiers will improve significantly.

Instead of using the whole training set to generate a validation set, a balanced training set was extracted from the full training data with the same number of instances in each class. This balanced training set was then used to train the stacking model with the exactly same parameters and vectorisation methods TF-IDF with k_ratio of 0.6. The accuracy increased to 0.67446 but it was not yet the highest. Since the new balanced training dataset has a much smaller size than the original dataset, we believed that it was necessary to increase the k_ratio and the number of features in the new model. After several trials, the stacking model achieved its best performance with an accuracy of 0.67991 when adjusting the k_ratio to 0.8. The results of this model are promising, with all

evaluation metrics higher than the previous models, including the previous stacking model as seen in Figure 11 and Table 1.

However, it does have a limitation. As demonstrated in Figure 9, the stacking model took several times longer than other models. This is reasonable since the nature of the stacking model is to use a combination of all other models. For our study, I suggested that the running time of 11 minutes is acceptable since it does bring better performance. Nevertheless, in real life application of text classification, it is indeed the company's choice of whether the time complexity is worth it based on their demand.

5. Conclusion

In general, all the models trained by the unbalanced class-distributed training set were biased towards the Neutral class to some extent. Stacking classifier was considered to be the least biased model. Its performance can be further improved significantly if it is trained by a reduced training set of which classes are distributed evenly. As a result, the balanced stacking model gives an accuracy of 68.2%, which makes it the most reliable model to use for tweet sentiment prediction.

6. Reference

Brownlee, J. (2020, April 10). *Stacking Ensemble Machine Learning With Python*. Machine Learning Mastery. Retrieved May 11, 2022, from https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/

Chen, L. (2019, January 7). *Support Vector Machine — Simply Explained | by Lilly Chen*. Towards Data Science. Retrieved May 9, 2022, from https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496

Islam, M.Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019). A Semantics Aware Random Forest for Text Classification. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Lakshmana Phaneendra Maguluri, & R Ragupathy. (2019, NOVEMBER 11). *Automatic Classification Of Stock Twitter Data By Using Different Svm Kernel Functions*. International Journal of Scientific & Technology Research.

Martin, C. H. (2012, February 6). *Kernels Part 1: What is an RBF Kernel? Really?* calculated | content. Retrieved May 9, 2022, from https://calculatedcontent.com/2012/02/06/kernels_part_1/

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media, 2013.

Rayana, S. & Akoglu, L. Collective opinion spam detection: *Bridging review networks and metadata.* Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 985-994

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). *SemEval-2017 Task 4: sentiment analysis in Twitter.* In Proceedings of the 11th International Workshop on semantic evaluation (SemEval '17). Vancouver, Canada.

Samadi, H., & Luo, L. (2022). Decision Trees. *Machine Learning Lecture Slides*, *Week 4*(Lecture 1), 63. https://canvas.lms.unimelb.edu.au/courses/124130/pages/week-4-decision-tree-and-instance-based-learning?module_item_id=3196367