COMP 551
Mini Project 2: Classification of Textual Data

Jerry Dong, Nithilasaravanan Kuppan, Thomas Hillyer *McGill University*
GROUP 86

***Abstract*—In this project, the performance of various Machine Learning (ML) algorithms, namely Logistic Regression (LR), Decision Trees (DT), Support Vector Machine (SVM), AdaBoost (AB), Random Forest (RF) and Multinomial Naïve Bayes (MNB) was investigated on two benchmark datasets. The text data was preprocessed where whitespace, punctuation and stop words were removed. The text was further processed through lemmatization. POS tagging was introduced before vectorizing the dataset. GridSearchCV was utilized to find the best hyperparameters and the winning algorithms were identified.**

I. INTRODUCTION

In this project, we explore various machine learning models on two datasets (20 Newsgroup dataset and IMBD dataset) as we analyze their accuracy as the metric for performance. The datasets were cleaned and lemmatized with part-of-speech tagging to optimally capture their sentiment. We also use grid search with cross validation for hyperparametric tuning. We found that MNB performed best on the 20 News group data set whereas SVM had the highest accuracy on the IMDB dataset.

1) Logistic Regression: Logistic Regression (LR) is a discriminative classification technique that maps the conditional distribution[1]. LR considers the functional form of P($x|y$) estimating directly from the dataset. Certain studies have shown that as you increase the number of training sets, LR will achieve a lower asymptomatic error rate[2].

2) Decision Trees: Decision Trees (DT) is a common method for establishing classification systems based on multiple covariates or developing predictions for specific target variables. This method employs branch-like segments that construct an inverted tree with a root node, internal nodes and leaf nodes[3].

3) Support Vector Machines: Support Vector Machines (SVM) perform pattern recognition between two-point classes through drawing a hyperplane determined by certain points in the dataset. The objective of the SVM is to find among the hyperplanes that correctly classifies data [4].

4) AdaBoost: Adaptive Boosting (AB) conventionally combines multiple weaker classification models that are independently trained and combines their predictions to produce a stronger classification[5]. Certain studies show that hybrid AB algorithms achieve higher detection probability than conventional Machine Learning based spectrum sensing algorithms.

5) Random Forest: Random Forest (RF) is a form of "ensemble learning" method that generates many classifiers and combines their results[6]. Contrary to DT, each node in RF is split using the best among a subset of predictors chosen randomly at that specific node.

6) Multinomial Naïve Bayes : The Multinomial Naïve Bayes (MNB) is a generative classifier considering the functional form of P($Fi|c$) [7]. MNB uses a multinomial distribution for each of the features.

A. The Task

This project investigates the performance of LR, DT, SVM, AB and RF as we compare the classification methods on textual data from two groups of datasets. These included data from a collection of newsgroup documents partitioned across 20 different news groups (Multi-class Classification) and data collected from IMDB movie reviews (Binary Classification).

B. Related Work

On the 20 Newsgroup dataset, Shuo Xu has demonstrated that the performance of Bayesian Naïve Bayes classifier with multinomial event model is similar to its classical counterpart. Whereas, Bayesian Naïve Bayes classifier with Gaussian event model performs better[8].

On the IMDB dataset, Maas et al. use a mixture of unsupervised and supervised techniques to learn word vectors to capture their sentiment[9]. They evaluated the sentiment polarity classification where they predicted whether a given review was positive or negative. Maas et al. also performed subjectivity classification at the sentence level to determine whether the sentence is subjectively expressing opinions. Other work includes using Vector Space Models (VSM) to model words directly[10].

II. DATASETS

A. 20 News Group Dataset

This being a text classification problem, utilizing this dataset to compare machine learning algorithms was essential. This dataset is a collection of about 20 000 newsgroup documents, each arranged into a different topic bucket. There are 20 such buckets and the primary objective of this project is to create features from these files, train the model and predict the categories for the files in the test folder. Both the train and test folders are split almost equally, which is illustrated by the pie charts given below.

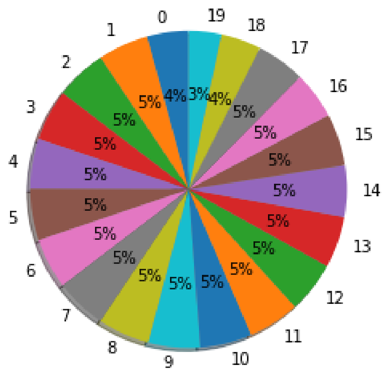

Figure 1: Class splits for train folder

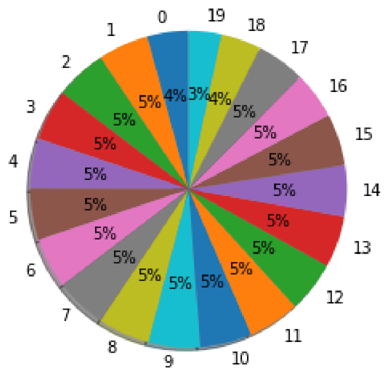

Figure 2: Class splits for test folder

B. IMDB Dataset

The main objective is to perform a binary sentiment classification on movie reviews to determine if they are positive or negative. The IMDB dataset used contains 50 000 text reviews that is split into test and train sets, each containing 25 000 reviews respectively. A negative rating is defined as a score of less than 4/10 whereas a positive rating is defined as a score of greater than 7/10 and the remaining scores are considered a neutral rating. There are no more than 30 reviews per given movie as the same movie will on average receive similar ratings.

C. Data Processing

Unlike regular classification problems where the data itself has a basic set of features, in textual data analytics the first step is to try to break the data down into features. The following section incisively explains the various processing steps the group undertook to come up with features for both the datasets.

1) Data Cleansing: The 20 newsgroups datasets were ingested into Python in a dataframe which essentially just had two columns - The raw text from the file and the target variable, i.e the category. Before trying to create features out of the text column, it was essential to remove digits, white-spaces and special characters (including punctuations).

On the IMDB-dataset, line breaks and white spaces were removed from each file as they were read in. Numbers, special characters (‘$,;&…etc) and punctuation was then removed and all the text were converted to lower case. Stop words such as “and, is, the…etc” were also dropped for optimal sentiment analysis.

2) Basic Metrics Generation: Some basic metrics - number of words, number of characters and average characters per word, were calculated on the 20 newsgroup dataset and were appended to the dataframe.

3) Lemmatization and Text Vectorization: On IMDB-dataset, each text review was then tokenized and lemmatization was performed on each word. For example, the word “movies” was converted to just “movie”. The text was vectorized and the term frequency – inverse document frequency (TF-IDF) scores were passed on for modelling, since classification algorithms need numerical features to work on. The choice to go with TF-IDF instead of count vectorized was fuelled by the fact that the TFIDFVectorizer’s value increases as the count increases but also offsets by a value that is equal to frequency of the word in the corpus.

Similarly, the texts were tokenized and lemmatized for further processing. However, these texts weren’t vectorized in this step.

4) Part of Speech Tagging: The lemmatized and tokenized texts from the 20 newsgroups dataset were then made to pass through a set of functions that appended the Part of Speech of every token to each of the tokens. This resulted in the addition of columns that indicated the number of Nouns, Pronouns, Verbs, Adverbs and Adjectives in the texts.

5) Normalization and Binarized Output Labels: On IMDB dataset, the rating was composed by the scalar number up to 10, which corresponded to the polarity of the review. To binarize the output, all ratings greater than 7 were considered as ’1’ (positive review) and all rating less than 4 were considered as ’0’ (negative review).

Since the 20 newsgroups classification is a multi-class classification, there was no need to binarize the output labels. But, following best practices of modelling, all the numerical columns were normalized.

D. Data Analysis

Prior to implementing classification algorithms, it is important to check if the methods being implemented make sense logically. The following visual proofs will assist in validating the vectorization approach.

Taking the 20 newsgroups datasets for this experiment, the following wordclouds showcase the major words that appear in texts for classes 1, 10 and 19 (post stop-word removal).

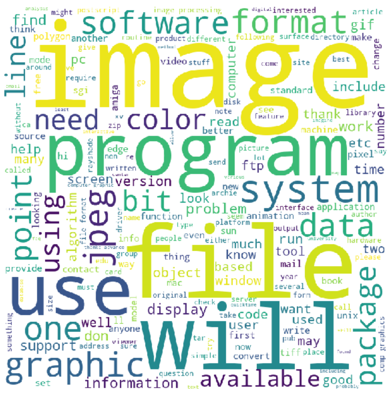
Figure 3: Wordcloud for Class 1 - Train

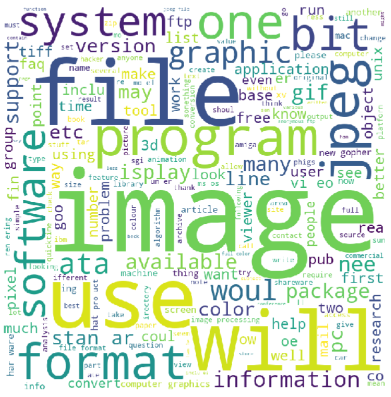
Figure 4: Wordcloud for Class 1 - Test

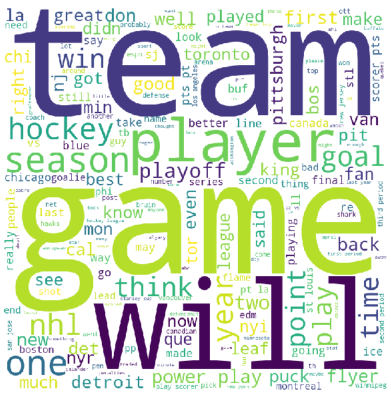
Figure 5: Wordcloud for Class 10 - Train

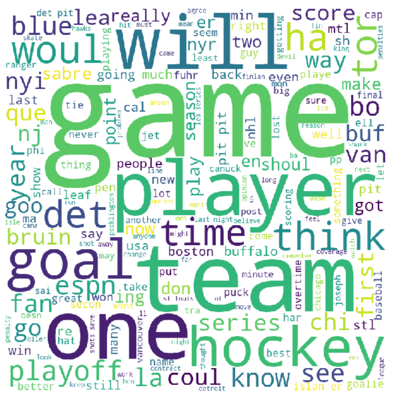
Figure 6: Wordcloud for Class 10 - Test

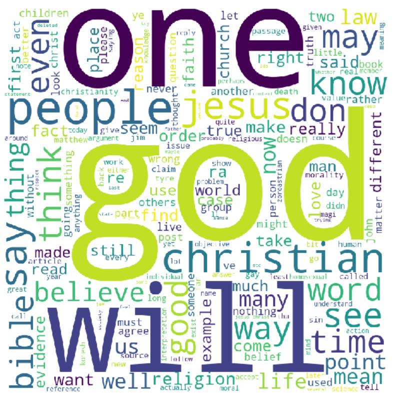
Figure 7: Wordcloud for Class 19 - Train

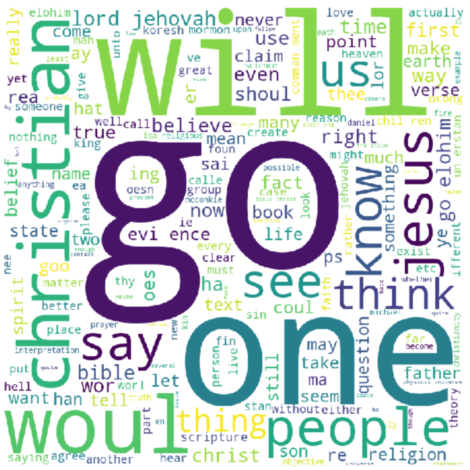
Figure 8: Wordcloud for Class 19 - Test

From the above figures, we can see that there is a strong relation between the train and the test sets of a particular class. Therefore, the logical way would be to count (specifically, count and adjust for the occurrence frequency) the words in the test and train - this will be the basis for all the machine learning models, discussed in the next section.

The following figure summarizes the 20 newsgroup datasets.

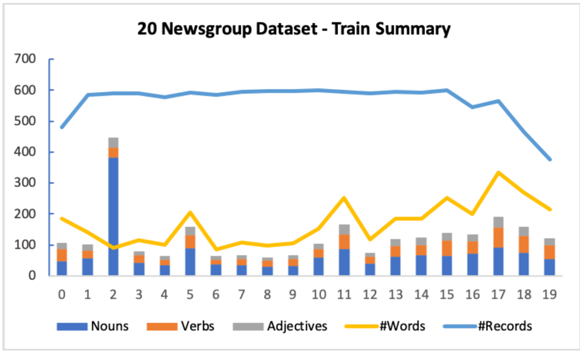

Figure 9: Summary - 20 newsgroups Train Dataset

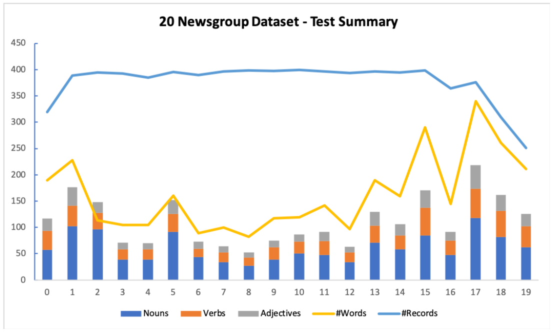

Figure 10: Summary - 20 newsgroups Test Dataset

III. Results

In this section we shall briefly discuss about the logic of the algorithms implemented, hyperparameter tuning and present a comparison.

A. Combining TFIDF with other features and ngrams

For the multiclass classification problem using 20 newsgroups dataset, DataFrameMapper was used to tie the TFIDF vectorizer with the other basic metrics that were already calculated from the dataframe. Additionally, on top of TFIDF vectorization, the function also created unigrams and bigrams stripped accents and limited the number of features to top 20 000.

B. Hyperparameter tuning

GridSearchCV was used to do a 5 fold (3 fold for 20 Newsgroup problem) cross validation on different hyperparameters for different classification algorithms. GridSearchCV was fed well researched ranges for all the major hyperparameters for cross validation and the best out of those paramters were used to train the model and score the test data. The tables below show the hyperparameters and the ranges provided to them for GridSearchCV.

Table I: Hyperparameter(s)

Algorithm	**Hyperparameters**
Logistic Regression	C, tolerance, penalty
Decision Trees	ccp_alpha, max_features, min_samples_split, min_samples_leaf
Support Vector Machines	C, tolerance, penalty
AdaBoost	n_estimators, learning_rate
Random Forest	ccp_alpha, max_depth
Multinomial NB	alpha

Table II: 20 Newsgroup Dataset - Results

Algorithm	**Best Hyperparameter(s)**	**% Accuracy**
Logistic Regression	C = 1	64.88%
Decision Trees	max_depth = 200	37.06%
Support Vector Machines	C =1	63.71%
AdaBoost	n_estimators = 150, lr = 0.5	41.47%
Random Forest	n_estimators = 300, max_depth = 90	55.96%
Multinomial NB	**alpha = 0.1, fit_prior = None**	**65.32%**

Table III: IMDb Dataset - Results

Algorithm	**Best Hyperparameter(s)**	**% Accuracy**
Logistic Regression	C: 10.0, penalty: ’l2’, tol: 1	86.04%
Decision Tree	ccp_alpha: 0.001, max_features: None	74.672%
Linear SVC	**C: 0.21544, penalty: ’l2’, tol: 1e-06**	**86.584%**
AdaBoost	learning_rate: 0.9, n_estimators: 200	82.92%
Random Forest	ccp_alpha: 0.0, max_depth: None	83.328%
Multinomial NB	alpha: 1.914482	83.088%

C. Final Results

The tables summarize the best hyperparameters and the corresponding algorithms’ accuracies.

The above tables clearly indicate the winning algorithms for each of the two datasets. In case of the 20 newsgroups dataset, it was difficult to achieve great accuracies despite the addition of extra non vectorized features - however, the addition of Multinomial Naive Bayes proved to be worthy because that gave the best test accuracy out of all the algorithms. Logistic Regression was a close second in terms of test accuracy.

In case of the IMDb dataset, Linear SVC performed the best out of all the algorithms followed by Logistic Regression.

On the other hand, it is clearly noticeable that Decision Trees did not really perform well for either of the tasks.

IV. Discussion and Conclusion

The primary motive of this project is to implement and compare five major classification algorithms, Logistic Regression, Decision Trees, Linear SVC, AdaBoost, and RandomForest.

Not every model is suited to every set of data and their relative strengths can be shown by comparing them on two different types of datasets. These five algorithms, plus Multinomial Naive Bayes, were employed on 2 different datasets. The first dataset, 20 News Groups is a multi class problem, whereas the second, IMDb reviews, is a binary classification problem.

- It is important to pre-process data in order to get accurate results. It may have been more effective for the IMDB database of reviews if words like actor, movie, film, were stripped out to only focus on sentiment holding words.
- Some classification models are more suited to certain types of problems. MultiNB was the most accurate for multi-class, yet lagged a little compared to Linear SVC for binary classification.
- Cross validation can substantially improve the training model. Simple guesses at ranges can be narrowed down and easily improve the overall accuracy.

For the IMDb dataset, only about half of the entries available in the set were used for training due to memory limitations. Future improvements could include better memory management or more interestingly how biased a trained model could be based on the subset of data used.

One major issue that troubled the whole process was the run times for some of the algorithms. Creation of final feature set using DataFrameMapper, running cross validations on complex AdaBoost and Random Forest algorithms took its toll on the system's resources and time. The group would love to look into more efficient ways of implementing these logics. However, using n_jobs = -1 in GridSearchCV and other sklearn algorithm syntaxes enabled those processes to work on all cores of the system, hence improving the run times.

Text Analytics is a huge topic and this project helped us understand the basics of it. With the addition of NLP and other Deep learning techniques, it is possible to massively increase accuracies by engineering new features.

V. Statement of Contributions

The collaborative work of this team involved everyone contributing to the report writing part. Jerry designed the base code and the models for the IMDb dataset and also contributed to the introduction, literature survey and references in the report.

Nithilasaravanan contributed by writing the base code for the 20 newsgroup dataset and then running the models, along with CV for the same. He assisted in writing the results and processing part of the report.

Thomas worked on implementing GridSearchCV for the IMDb dataset and worked on the results and conclusion section of the report. He also assisted with some of the figures in the report.

References

[1] L.M. Gladence, M.Karthi and V. Maria, A statistical comparison of logistic regression and different bayes classification methods for machine learning, *ARPN Journal of Engineering and Applied Sciences*, vol.10, pp.5947-5953, Jan. 2015.

[2] A.Y. Ng and M.I. Jordan,On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes, *Neural Information Processing Systems*, 2002.

[3] Y. Song and Y.Lu, Decision tree methods: applications for classification and prediction,*Shanghai Archives of Psychiatry*, 2015.

[4] S.Yue, P. Li and P. Hao. SVM classification:Its contents and challenges, *Applied Mathematics-A Journal of Chinese Universities*, 2003.

[5] . S. Chen, B. Shen, X. Wang and S. Yoo. A Strong Machine Learning Classifier and Decision Stumps Based Hybrid AdaBoost Classification Algorithm for Cognitive Radios,*sensors*2019.

[6] . A. Liaw and M. Weiner. Classification and Regression by RandomForest,2001.

[7] . G.Webb and X. Yu. Multinomial Naive Bayes for Text Categorization Revisited,*AI 2004: Advances in Artificial Intelligence*2004.

[8] . S. Xu. Bayesian Naïve Bayes classifiers to text classification Naive Bayes for Text Categorization Revisited,*Journal of Information Science*2016.

[9] . A. Maas, R. Daly, P. Pham and A. Ng. Learning Word Vectors for Sentiment Analysis ,2011.

[10] . P. Turney and P.Pantel. From Frequency to Meaning: Vector Space Models of Semantics ,2010.

[11] *20 News Groups Dataset* http://qwone.com/~jason/20Newsgroups/

[12] *IMDB Reviews Dataset* http://ai.stanford.edu/~amaas/data/sentiment/

[13] Logistic Regression SKLearn Model

[14] Decision Tree SKLearn Model

[15] Linear SVC SKLearn Model

[16] Adaboost SKLearn Model

[17] Random Forest SKLearn Model

[18] Multinomial Naive Bayes SKLearn Model