COMP30027 Report
Sentiment Analysis

1. Introduction

Sentiment analysis is a technique to determine the opinions of people about products, companies, social topics etc. It is the process of studying any textual interaction and measure people's reactions. This information is then later used by entities to make informed decisions about the topic. It is a very useful tool because of how critical online presence is for companies and other entities. One of the applications of sentiment analysis is social media monitoring. For example, twitter one of the biggest social media platforms, has over 350 million users posting over 500 million tweets in 2021. The 2 main methodologies used for sentiment analysis are 1) Lexicon-based, 2) Machine learning. In this paper we are using the Machine learning method. We are accessing the effectiveness of different machine learning models on the problem of determining tweet's sentiment from twitter. We have a dataset of nearly 22000 tweets extracted from twitter and labelled with 3 different sentiments (negative, neutral, positive).

2. Method

In this paper we access the significant of logistic regression, naïve bayes classifier and neural network.

2.1 Data Description

Our primary data consist of the following:

- tweet id
- the tweet itself
- sentiment

this data has been stored in system using a comma separated file format for sentiment analysis.

2.2 Data Pre-processing

The tweets from this data initially need to be to clean. We achieved this by removing the following:

- stop words
- filler words
- usernames i.e., @something
- hashtags i.e., #something
- hyper links
- html tags
- symbols and characters

Then we proceeded to tokenize each tweet and lemmatize each token within each tweet. Lemmatization is the process of bringing each to its own stem i.e., running will be stemmed to run. As you can see in Figure 1 and 2 the length of our tweets has almost been halved after cleaning.

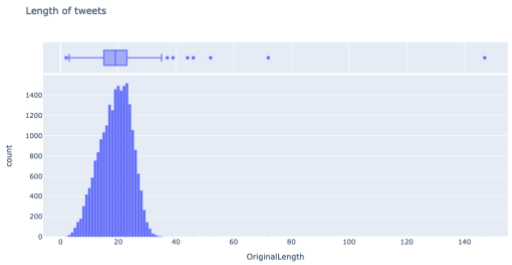

Figure 1- Length of tweets before cleaning

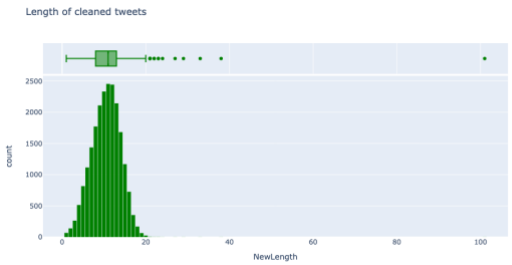

Figure 2- Length of tweets after cleaning

Then we proceeded to visualise each sentiment category using word cloud to get a better understanding of the data. Word clouds are a very fast and engaging tool to reveal the most used words in each category. One of the words most used in both positive and neutral category was **Tomorrow**. It is very useful to outline this as we're using word frequency as features in our machine learning models because it can cause overlap. In this data set we are dealing with

text and to use that effectively we need to transform the text to numbers. Text vectorization is essential for machine learning. I used tf-idf in this paper. tf-idf in simple words is the frequency of a word used in a text within multiple texts. tf-idf has some advantages and disadvantages. The reason I have chosen tf-idf is very easy to compute and you can easily extract keywords and associate them with their sentiment category. I have used logistic regression classifier as a base model and naïve bayes classifier and finally a neural network. I've used a recurrent neural network (RNN) for this experiment. RNN is type of neural network which keeps an internal memory of what has been used already while processing a new word each time. This way even the sequence of each word in our tweets will affect our model's predictions.

I wanted to compare these different classifiers and determine which one is the most efficient given our data set. For evaluation method I have included the model accuracy score, f-score, recall, precision, and a confusion matrix for logistic regression and naïve bayes classifier to get a better understanding of the model and how we can improve them and I have used a learning curve graph for the neural network to identify overfitting or underfitting.

3. Results

I have used the logistic regression as the base model. Instead of splitting the training data and I decided to allow the machine learning method train on the whole data and then proceeded to allow the model to predict the training model again and compare the results and evaluate using the metric mentioned above and received the following results as shown in figure 3.

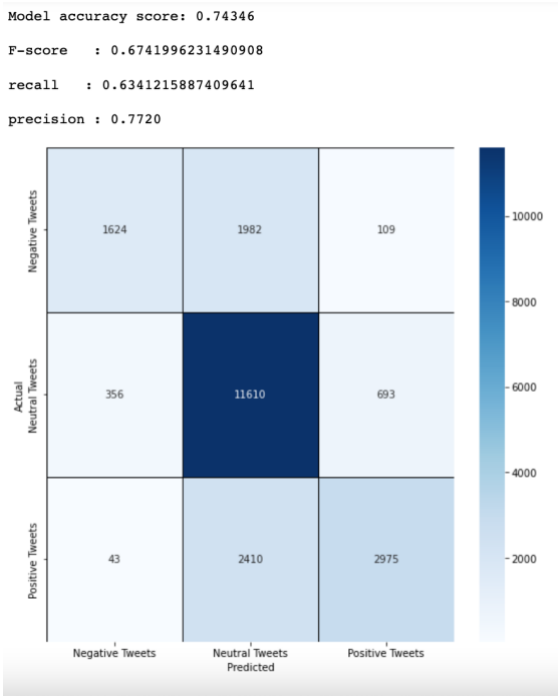

Figure 3- logistic regression evaluation

The overall accuracy of the model was average at 0.74 and receiving an f-score of 0.67 was quite poor.

Our second machine learning model was naïve bayes classifier. I used multinominal naïve bayes classifier. I received the following results shown in figure 4.

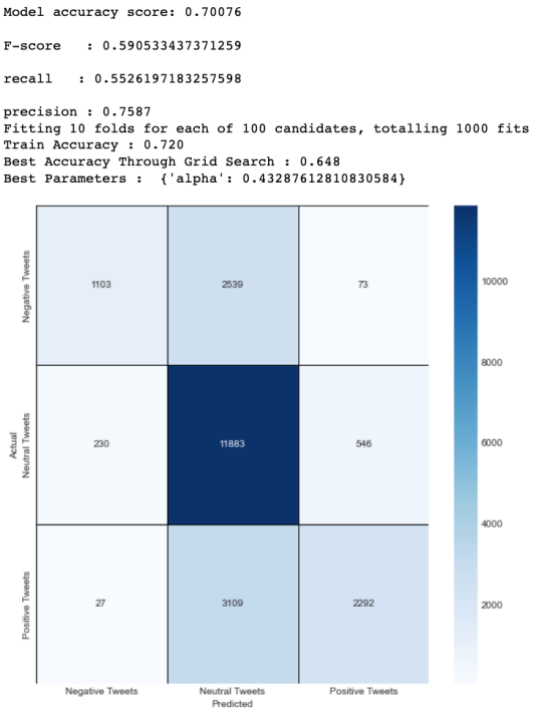

Figure 4- naïve bayes classifier evaluation

As you can see in the evaluation above after receiving an accuracy of 0.70 which is quite average and a very low f-score of 0.59 I decided to apply some hyperparameter tuning to improve these results and reached 0.72 for model accuracy which was still disappointing.

My final model for this data set was the recurrent neural network. I have used tensorflow in my implementation for our model. Tensorflow is quite easy to compute and understand. Initially we had to tokenize the tweets in a format so that tensorflow can understand them. After training the model and making predictions I received the following results shown in figure 5.

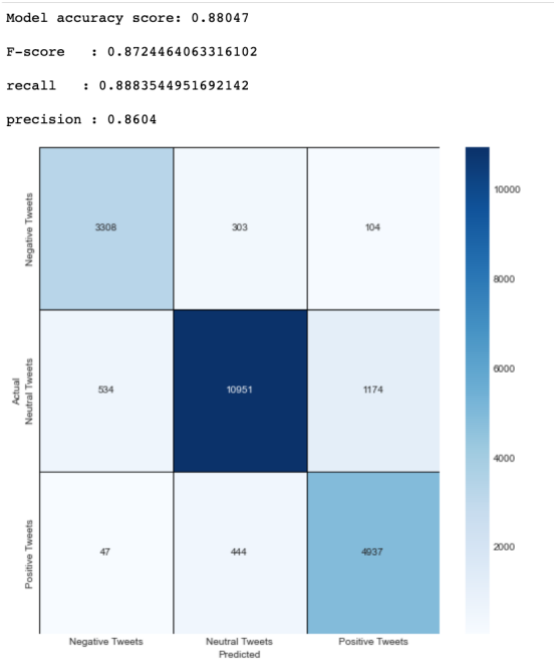

Figure 5- RNN classifier evaluation

As you can see above, we receive an accuracy of 0.88 for the RNN model which is quite high following by a very high f-score of 0.87.

4. Discussion

Initially I would like to start discussing our base model which was the logistic regression. Logistic regression is one of the simplest machine learning models which was a good starting point and a good benchmark to measure performance and compare it with other models. Logistic regression aims to predict probabilistic outcomes depending on independent features which makes it useful model for this data set. As we have used tf-idf for our feature engineering the frequency of each word is independent from other words. One slight problem with my approach for the logistic regression classifier is that there is a high chance of overfitting, that is overselling the accuracy of our prediction specially because I did not split the training data. This may have resulted in the model accuracy being inaccurately high for the training data and will result in different accuracy when applied to the testing data set.

Secondly, I like to discuss my naïve bayes classifier model for this dataset. Multinominal naïve bayes classifier is also a very fast machine learning model and works quite well with multi class labels which applies directly to our dataset. One of the downfalls of using naïve bayes classifier for our dataset is that given our feature engineering our testing and training data won't be made of the exact same features which will result in our model not predicting all the data therefore a smoothing technique is necessary. Naïve bayes classifier also makes very powerful assumption about our data distribution so it is important to mention to achieve the best result using naïve bayes we need to transform our training and testing data set, so they follow the same distribution.

Lastly, I like to discuss my recurrent neural network model for this dataset. RNN model takes on a tree-based representation. This allows the model to calculate the overall sentiment of a tweet by not only using each word but also phrases. In other words, this model picks up on negation and n-grams that carry novel sentiment analysis. It works very well with nonlinear data and large data inputs. One of the downfalls of RNN is that RNN heavily depends on the training data and is very much prone to overfitting. Using tensorflow I split the training data into a portion 20 percent for validation and graphed out the accuracy and loss of model as shown in figure 6.

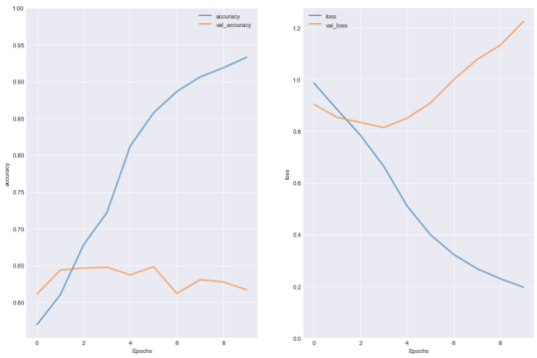

Figure 6- RNN train/validation graph comparison

As you can see in figure 6 the accuracy and

loss values in the training and validation data is quite different. This is a result of overfitting. After handling the overfitting, I received the following results shown in figure 7.

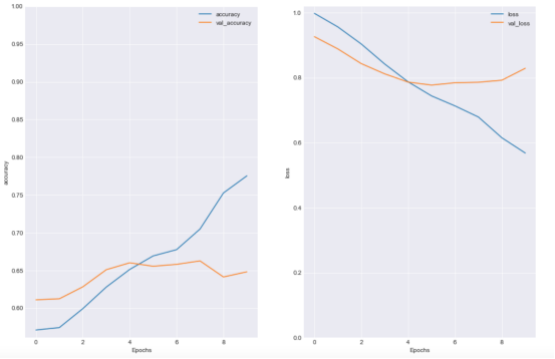

Figure 7- RNN train/validation graph comparison

As shown in figure 7 the overfitting problem has been solved to an extent. Changing my RNN model to handle overfitting resulted in changing the accuracy of the whole model you can view the new evaluation metrics below in figure 8.

```
Model accuracy score: 0.85717
F-score   : 0.8410792327009854
recall    : 0.8341037284541905
precision : 0.8487
```

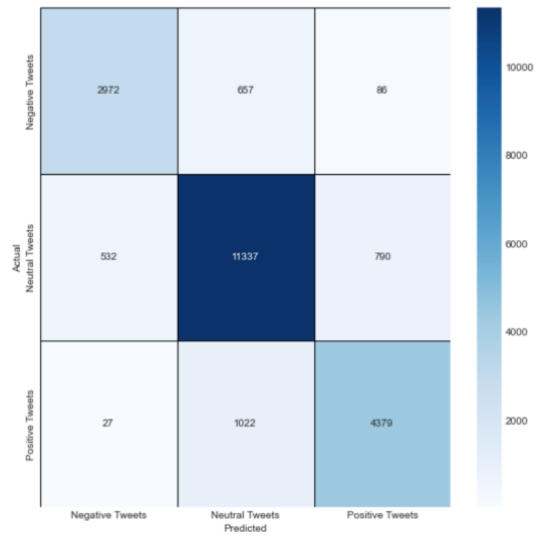

Figure 5- RNN classifier evaluation

As you can see in figure 8 our accuracy and f-score has dropped by 3 percent. This may had resulted in a worse performance, but it is a more realistic performance.

5. Conclusions

In conclusion when working with different machine learning models the most important step is understanding the data you are working with because different models favour different data sets. As clearly discussed above working with sentiment analysis deep learning machine learning models such as recurrent neural network, long short-term memory neural network etc will give you a better performance than your traditional machine learning method.

6. References

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.

Kaggle.com. 2022. Coronavirus tweets NLP - Text Classification. [online] Available at: <https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification> [Accessed 13 May 2022].

Daulatkar, S. and Deore, A., 2022. Post Covid-19 Sentiment Analysis of Success of Online Learning: A Case Study of India. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/9763272> [Accessed 13 May 2022].

Devi, C. and Devi, R., 2022. Big Data Analytics Based Sentiment Analysis Using Superior Expectation-Maximization Vector Neural Network in Tourism.

[online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/9753738> [Accessed 13 May 2022].

DataRobot AI Cloud. 2022. Using Machine Learning for Sentiment Analysis: a Deep Dive. [online] Available at: <https://www.datarobot.com/blog/using-machine-learning-for-sentiment-analysis-a-deep-dive/> [Accessed 13 May 2022].

HongDa, Y. and Takano, K., 2022. A Recommendation Method for Social Media Users based on a Sentiment Analysis Model. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/9754863?arnumber=9754863> [Accessed 13 May 2022].

Edgari, E., Thiojaya, J. and Qomariyah, N., 2022. The Impact of Twitter Sentiment Analysis on Bitcoin Price during COVID-19 with XGBoost. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/9756123?arnumber=9756123> [Accessed 13 May 2022].