Artificial Intelligence (CSE422)

Project Report

Social Media Sentiment Analysis

Platform: Twitter

Submitted By:

Mazharul Islam Rakib (20101408)

Maisoon Tasnia (20301076)

Saiwara Mahmud Tuhee (20101465)

Summer’22 || Section - 13

Table of Contents

Introduction

Twitter is yet another social media platform that centralises around the concept of “microblogging”. What has set Twitter apart from other social media platforms is that the posts of users, or on the other hand, known as "Tweets,” are scan-friendly. In a world where people are both hyperactive and prefer everyday things to be minimal and microscale, this platform has garnered a salient space in everyday life. People send tweets regarding everything, starting from sharing their everyday mundane thoughts, opinions regarding any notion or event, amateur reporting, promoting products or businesses, entertainment, and so on and so forth. Regardless, like any other platform, Twitter has an impressive ratio of positive and negative tweets, ranging from genuinely expressed aspirations and wishes to downright hateful content that more than often borders on social qualms.

Our project aims to execute sentiment analysis of bulk Twitter data and examine whether a tweet is positive, neutral, or negative. To achieve that, we are using Natural Language Processing Toolkits (NLTK) to preprocess our dataset and implement various machine learning algorithms to determine the accuracy of our findings. We are using Python 3 and libraries like Panda, Numpy, Scikit learn, Seaborn, Wordcloud, nltk, etc. to implement, parse, cleanse, and visualise our data and findings. We have set a bunch of stopwords to cleanse our data, stemmed and lemmatized to gain the most efficiency. We are using Logistic Regression, SVC (Support Vector Classification), and RFC (Random Forest Classification) as our machine learning models.

Methodology

Dataset Description

The dataset that we used is a collection of 44,950 tweets extracted from the Twitter API. 41,152 of which is used as training data and the rest of 3798 is used as test data. The training dataset has 41,152 rows and two columns, “text” and “y”. The test column contains the tweets extracted and y contains the analysed value of the tweets. To differentiate the tweets, we have set the value “2” as a positive prediction, “1” as a neutral prediction, and "0" as a negative prediction.

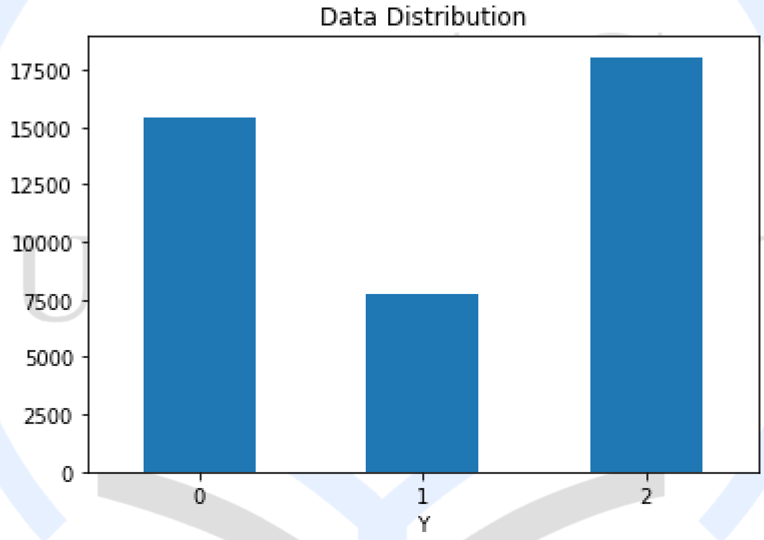


Fig: Visualisation of data based on the set variables

Preprocessing

Preprocessing data is a vital part of Machine Learning due to the fact that the standard and necessary information that needs to be extracted will directly affect the models’ ability to learn and comprehend. Thus, it is critical that we preprocess our data accordingly before catering it to our models’ (Kumar, 2021). The pre-processing techniques that we have used are on par with working with textual data in ML.

Firstly, we notice that there's an imbalance in our data and we must balance it out so that our ML model performance is not inhibited. We used the resample module from scikit learn to oversample our data to 17,500.

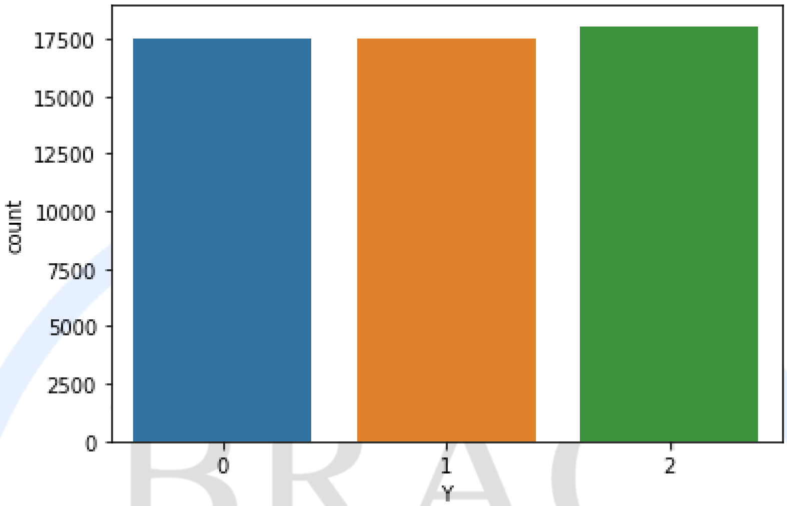


Fig: oversampled data

Secondly, we performed lowercasing on our data to avoid duplicate dimensions of the same text in the vector space. Then, we defined our list of stop words to remove words like 'a', 'and', 'but', 'through," etc. which are not necessary for our model to distinguish. We then cleansed our data based on the list of defined stopwords. Furthermore, we cleansed any repeating words, URLs, numbers, etc. We have used the 'string' and 're' modules to execute this. Then, using NLTK, we tokenized our data. We furthermore performed stemming, reducing our textual training data into their root forms. We then performed lemmatization to bring words that are synonyms under the same radar to further organise our data.

We then visualised our processed data using wordcloud. The visualisations are attached below:

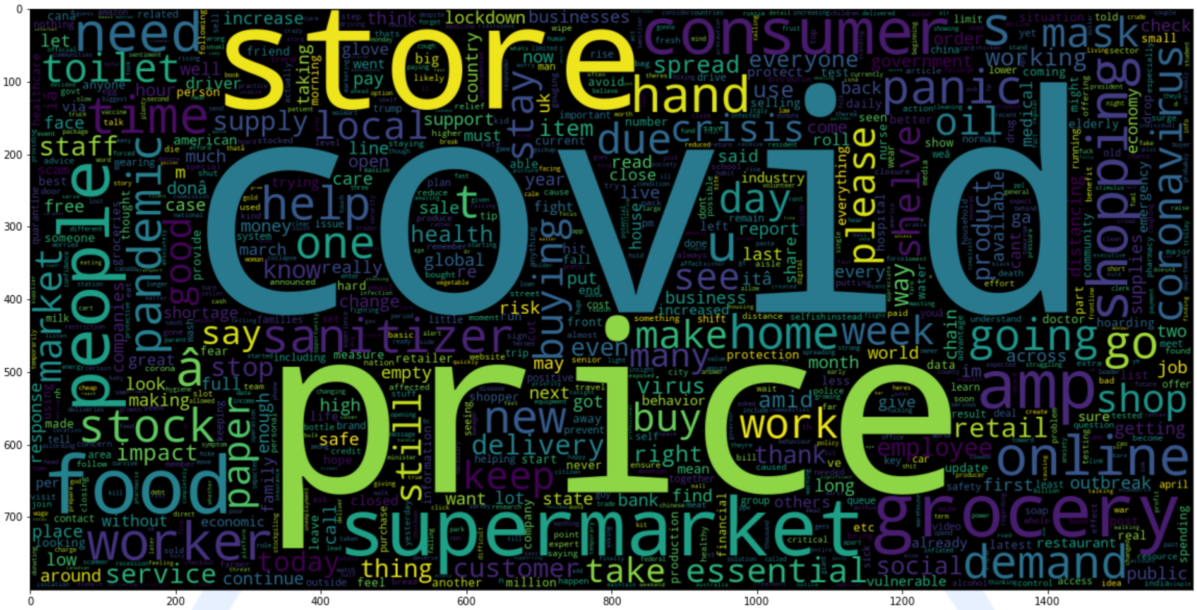


Fig: positive tweets

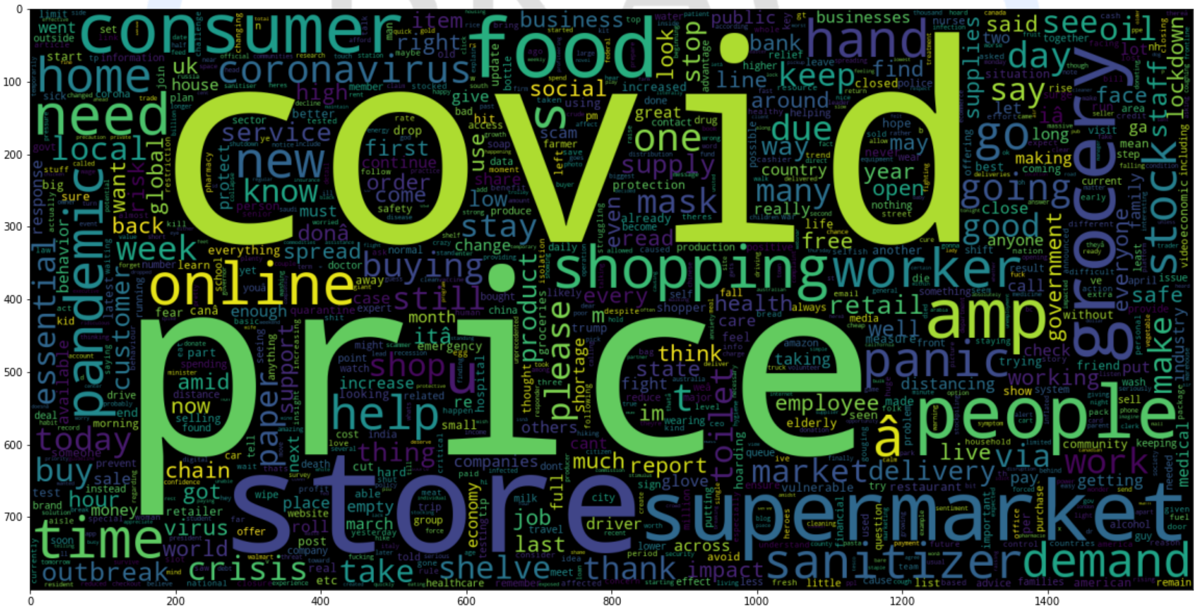
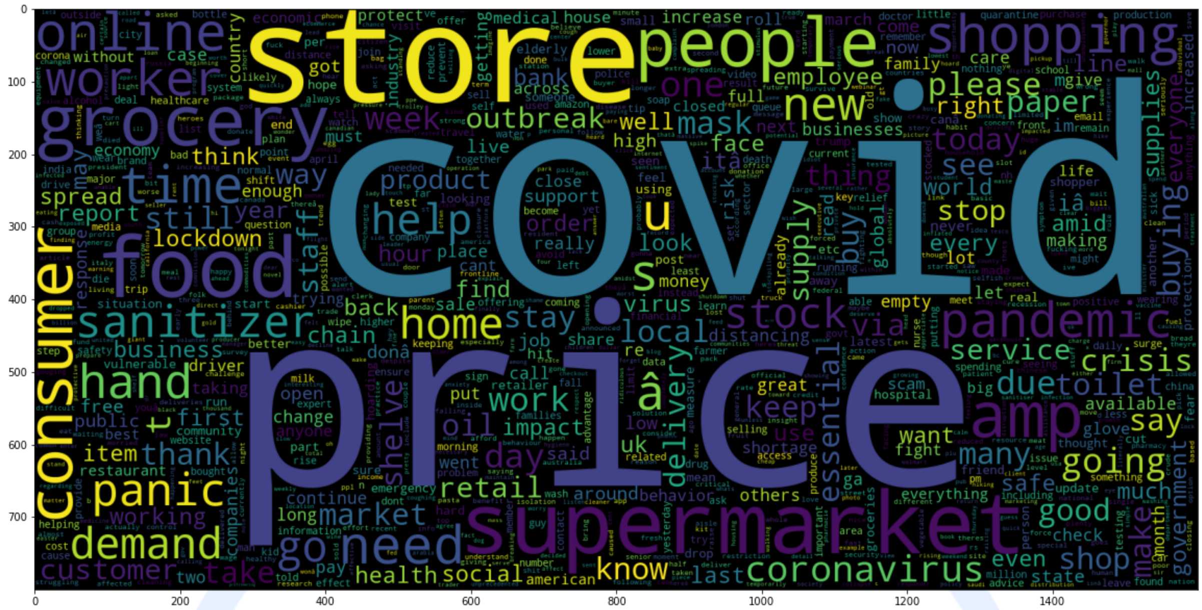


Fig: negative tweets



Fig; neutral tweets

Lastly, before moving on to our ML models, we vectorized our data using TfidfVectorizer with a max feature of 5000.

ML models used

One of the models that we used is the SVC or A Linear SVC (Support Vector Classifier)'s is to fit to the provided data, returning a "best fit" hyperplane that divides or categorizes the data. After obtaining the hyperplane, we can feed some features into our classifier to determine the "predicted" class. As a result, this algorithm is ideal for our purposes.(*Python Programming Tutorials*, n.d.)

Moreover, the supervised machine learning algorithm known as SVM can be applied to classification or regression problems. Regression predicts a continuous value, whereas classification predicts a label or group. By locating the hyper-plane that distinguishes the classes we plotted in n-dimensional space, SVM accomplishes classification.

A supervised machine learning approach based on ensemble learning is called random forest.

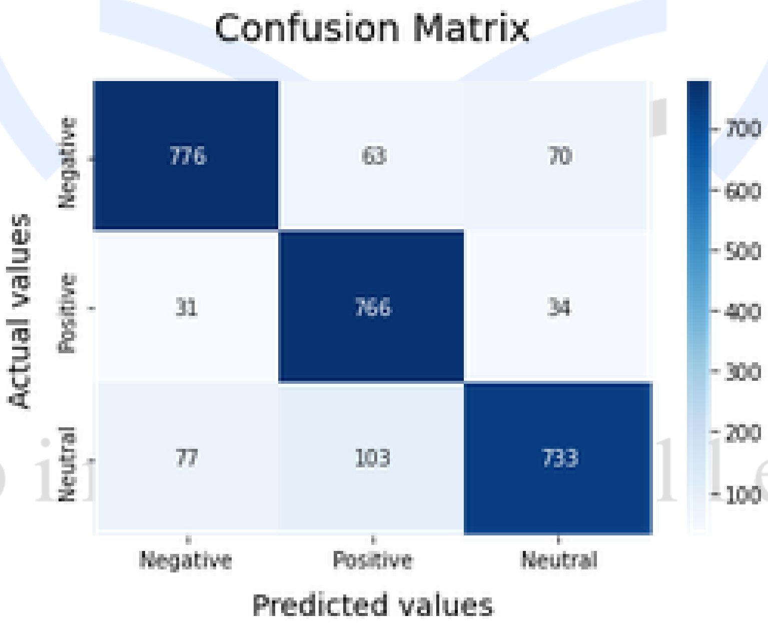
Lastly, the binary classification problem is resolved by the classification method known as logistic regression. In models that include a double circumstance, the outcome is typically defined as 0, 1 or 2 where, ‘0’ is stated as negative, ‘1’ as neutral and ‘2’ as positive. By combining binary classification with logistic regression on the data set's training and test data, estimation is accomplished.

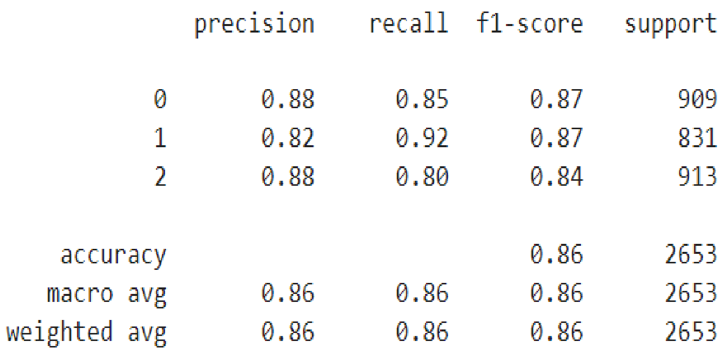
<u>Results :</u>

We will be depicting the results in terms of confusion matrices. A confusion matrix is a table that is used to define the performance of a classification algorithm.

The confusion matrices for the applied models are as follows:

<u>For SVC</u>

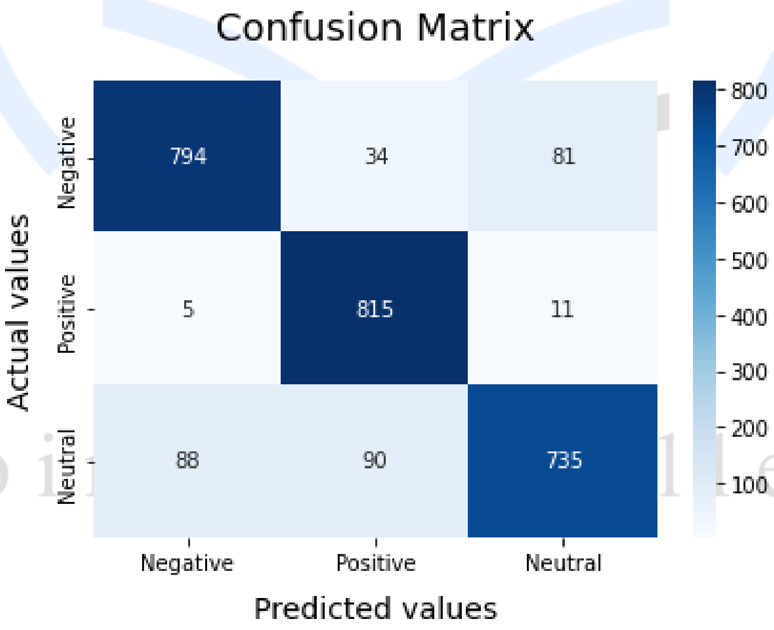


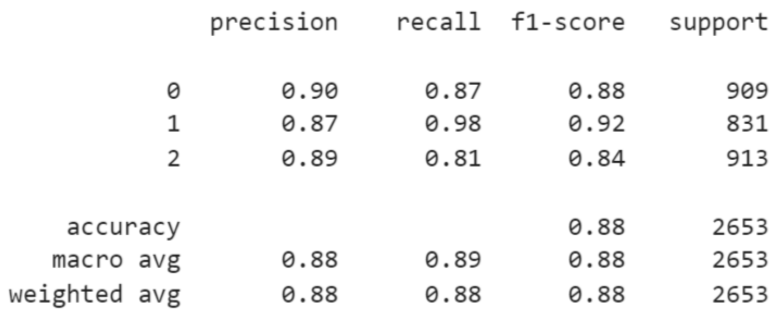


```
              precision    recall  f1-score   support

           0       0.88      0.85      0.87       909
           1       0.82      0.92      0.87       831
           2       0.88      0.80      0.84       913

    accuracy                           0.86      2653
   macro avg       0.86      0.86      0.86      2653
weighted avg       0.86      0.86      0.86      2653
```

Here we got an overall accuracy of 86% and the precision over negative data is 88%, neutral data is 82% and positive data is 88%. The test accuracy which is the f1 score for negative, neutral, and positive data is 87% ,87% and 84% percent respectively.

Random Forest Classification

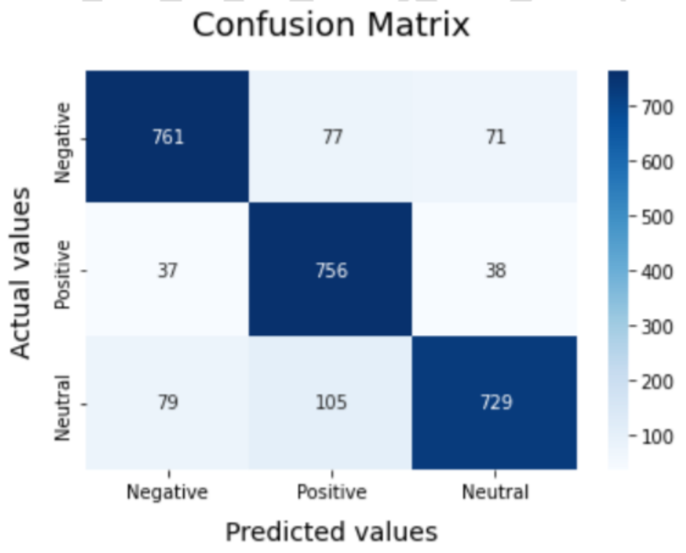


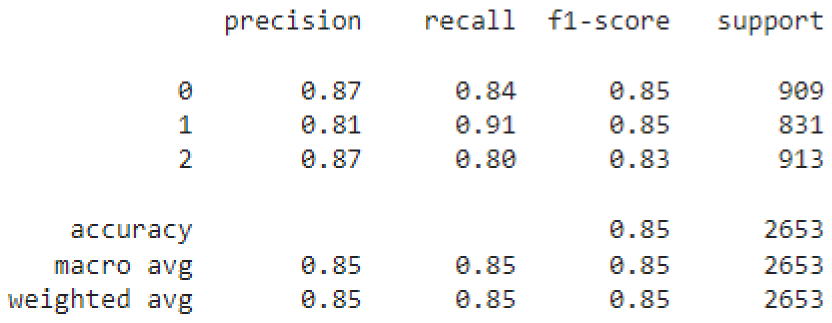


```
              precision    recall  f1-score   support

           0       0.90      0.87      0.88       909
           1       0.87      0.98      0.92       831
           2       0.89      0.81      0.84       913

    accuracy                           0.88      2653
   macro avg       0.88      0.89      0.88      2653
weighted avg       0.88      0.88      0.88      2653
```

Here we got an overall accuracy of 88% and the precision over negative data is 90%, neutral data is 87% and positive data is 89%. The test accuracy which is the f1 score for negative, neutral, and positive data is 88%, 92% and 84% percent respectively

Logistic Regression





```
              precision    recall  f1-score   support

           0       0.87      0.84      0.85       909
           1       0.81      0.91      0.85       831
           2       0.87      0.80      0.83       913

    accuracy                           0.85      2653
   macro avg       0.85      0.85      0.85      2653
weighted avg       0.85      0.85      0.85      2653
```

Lastly, here we got an overall accuracy of 85% and the precision over negative data is 87%, neutral data is 81% and positive data is 87%. The test accuracy which is the f1 score for negative, neutral, and positive data is 85% , 85% and 83% respectively.

References

Kumar, D. (2021, December 7). *Introduction to Data Preprocessing in Machine Learning*. Medium. https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d#:%7E:text=Data%20preprocessing%20is%20an%20integral,feeding%20it%20into%20our%20model.

Python Programming Tutorials. (n.d.). Python Programming. https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/

Goyal, G. (2022, August 25). *Twitter Sentiment Analysis- A NLP Use-Case for Beginners*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/twitter-sentiment-analysis-a-nlp-use-case-for-beginners/