Machine Learning Project
Twitter Sentiment Analysis

Sandip Kumar Burnwal
B19CSE075

Nandini Verma
B19CSE057

Rohit Sunarthi
B19EE073

I. Introduction

Sentiment Analysis is a technique widely used in text mining. Twitter Sentiment Analysis, therefore means, using advanced text mining techniques to analyze the sentiment of the text (here, tweet) in the form of positive, negative and neutral.
It is also known as Opinion Mining, is primarily for analyzing conversations, opinions, and sharing of views (all in the form of tweets) for deciding business strategy, political analysis, and also for assessing public actions.
Our discussion will include, Twitter Sentiment Analysis in Python, and also throw light on its techniques and teach us how to generate the Twitter Sentiment Analysis project report, and the advantages of enrolling it.

II. Dependencies Required for the Project

- Pandas library
- re i.e. regular expression
- nltk i.e. Natural language toolkit
- matplotlib.pyplot as plt

III. Preprocessing of the Dataset

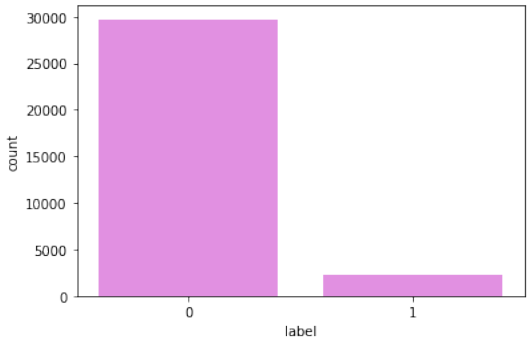
Data processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn therefore it is extremely important that we process our data before feeding it into our model.
It is also important to read through the data description file to thoroughly understand what each dataset is and what each column represents. A deep understanding of our dataset is a step towards success in building a good machine learning model.

A. Count for each target

We know that all the tweets present in the dataset have been classified in two groups 0 and 1 where 0 signifies that all the tweets belonging to this group are negative in nature whereas 1 signifies that all the tweets belonging to this group are positive in nature.
We checked the shape of each dataset and columns using train.shape and test.shape for train dataset and test dataset respectively.

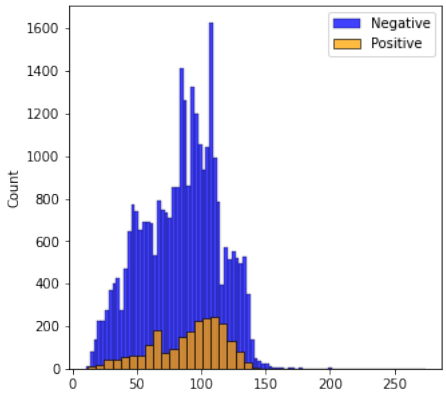
Identify applicable funding agency here. If none, delete this.

The size of train dataset is (31962,3) and that of test dataset is (17197,2).



B. Detecting Null values

- We classified the train dataset in positive and negative classes and also plotted bar graph for each class.



- We printed all the null value present in each column.There is not any null value present in the train dataset and hence we don't have to use dropna() function which drops all the null values present in the dataset.

C. Text Processing

We then prepared our data for the model. Here we have combined our datasets by appending test set on train set and the perform the following;

- Removing Double Spaces
- Removing HTML
- Removing URL
- Removing Stopwords
- Removing Emojis
- Removing URL
- Removing any other non English or Special Symbols

1.Removing Double spaces - In this we removed any extra space or unwanted space present in the dataset. Everytime we proceeded we added a new column of the dataset so that we can perform better analysis.

2.Removing HTML tags - In this we removed the unwanted HTML tags as they are not going to perform any role in data analysis.

3.Removing URLs - In this we removed all the URLs present in the dataset as they are not going to play any role in data analysis.

4.Removing Punctuation Marks - We will proceed with our data cleaning by removing punctuation marks, numbers and other special characters.

5.Removing emojis - An emoji is a small digital image or icon used to express an idea or emotion.We removed emojis as these are not going to be analysed by our model.

6.Removing emoticons - Emoticons are different from emojis.
An emoticon is a sequence of characters that represents a facial expression. Thus, an emoji can be basically anything, from facial expressions to food items to sporting goods to national flags.
An emoticon, meanwhile, is only a facial expression that conveys an emotion.

7.Replaced wrong spelling with the correct ones - Then we replaced all the wrong spellings with the correct ones. We can use spellchecker library for this purpose.

8.Removing Stopwords - Words such as ‘this’, ‘my’, ‘am’, ‘any’ may not provide valuable information in classification of the tweets. There words are generally referred to as stopword. These words increase the number of vectors that may slow down the training process and also ultimately make the model underperform. Therefore we removed the stopwords using nltk module.

9.We separated the data with label=0 and label=1 using groupby() function.

10.Plotted the word cloud of positive class and negative class -
We plotted the word cloud of target data using WordClouds and STOPWORDS library.

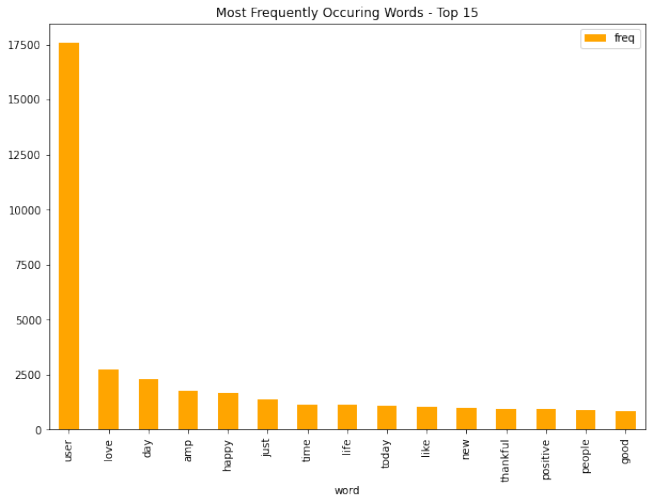
Word Cloud - A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is.

Word cloud of positive class

Word cloud of negative class

This type of visualization can assist evaluators with exploratory textual analysis by identifying words that frequently appear in a set of interviews, documents, or other text. It can also be used for communicating the most salient points or themes in the reporting stage.

11. Then we imported count vectorizer to see the frequency of each words and plotted bar graph for top 15 most frequently occuring words in the dataset.
CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.



12. Then we performed Stemming on the train dataset as well as test dataset.

Stemming is the process of extracting root word from words and phrases. For example, the root word for 'hopeful' and 'hopeless' is 'hope'. Stemming is usually applied in NLP to reduce the number of vectors used to create the model. This is because without stemming, we will have many words with the same meaning that would otherwise be reduced to one. It is therefore important to do stemming for your data. There are different word stemmers such as PorterStemmer and SnowballStemmer. In this case, we will use PorterStemmer.

The other alternative for Stemming is Lemmatization. Lemmatization is the process of grouping different forms of words so that they can be analyzed as a single term. It's is different from stemming since it does morphological analysis of the words. Lemmatization considers the context of the words.

13. After performing stemming we standardised both the dataset using fit transform which first fit the dataset and then reduce its dimensionality.

IV. Testing and Training

A. Split the dataset

We split the train dataset tha we got after standardization into train and valid part to train our model on different classifier. We have taken the test size = 0.25 and random state = 42.

B. Classifiers

Logistic Regression with non standardized data

We imported Logistic Regression from sklearn.linearmodel library to calculate the accuracy of our model on non standardized dataset so that we can compare the accuracy after standardizing the dataset. We have calculated training accuracy, validation accuracy and F1 score for the dataset.

F1 score for non-standardize data from logistic regression : 0.5597874224977856

Training Accuracy for non-standardize data from logistic regression model:
0.9808935797421885

Validation Accuracy for non-standardize data from logistic regression model:
0.937805030659492

We imported Random Forest Classifier to calculate the accuracy of model on non standardized dataset.The results are as follows: F1 score for non-standardize data from Random Forest : 0.6087844739530133

Training Accuracy for non-standardize data from Random Forest model: 0.9990822243544283

Validation Accuracy for non-standardize data from Random Forest model: 0.9520710799649605

We imported MLP classifier to calculate the accuracy and the results are as follws: F1 score for non-standardize data from MLP : 0.6087844739530133

Training Accuracy for non-standardize data from MLP model: 0.999123941429227

Validation Accuracy for non-standardize data from MLP model: 0.9510699536979101

We can observe that Validation accuracy in each cases is less than training accuracy because training data is something with which the model is already familiar with and validation data is a collection of new data points which is new to the model.
So obviously when the model is interacting with validation data, accuracy will be less than that of training data.

Standardization of Data
We imported PCA and then transformed the data using fit transform.

The principal components of a collection of points in a real coordinate space are a sequence of p unit vectors, where the i-th vector is the direction of a line that best fits the data while being orthogonal to the first i-1 vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis (PCA) is the process of computing the

principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest.

We calculated the training accuracy, validating accuracy and F1 score after applying PCA and the classifier used is Logisitic Regression. The results are as follows:

F1 score for standardize data from Logistic Regression : 0.5597874224977856

Training Accuracy for standardize data from Logistic Regression model: 0.9808935797421885

Validation Accuracy for standardize data from Logistic Regression model: 0.937805030659492
We also calculated the training accuracy on standardized data using Random forest classifier and obtained the following results:
F1 score for standardize data from Random Forest : 0.44566712517193946

Training Accuracy for standardize data from Random Forest model: 0.9990822243544283

Validation Accuracy for standardize data from Random Forest model: 0.9495682642973345

We have also calculated training accuracy, validation accuracy and f1 score using MLP classifier and got the following results:
F1 score for standardize data from MLP : 0.6666666666666666

Training Accuracy for standardize data from MLP model: 0.9980810145592591

Validation Accuracy for standardize data from MLP model: 0.9555750218996371

We have also calculated the accuracy value using cross validation. The classifier we used is MLP. the accuracy we get from cross validation is 0.9580751504451737.

C. Comparison of Accuracy Values

We can see although all the training accuracies values obtained in each classifier are almost comparable but the accuracy obtained using MLP is highest among all.The main reason behind this is MLPs are suitable for classification prediction problems where inputs are assigned a class or label. They are also suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs. We can see that after applying PCA (Standardisation of dataset) accuracy value has increased this is due to the functionality of PCA.

The main reason behind this is that distance based machine learning estimators calculate distances based on values for features. Ideally these values should be in a similar range of magnitude for correct predictions. However, in a dataset if the values of a feature are say 100000 times that of another feature then the calculated distances would be of a higher magnitude as well. This may cause the estimator to give more importance to the feature of higher magnitude. Thus, it could lead to incorrect predictions. Standardization comes to rescue! It helps to bring features of different magnitudes into a similar magnitude range.

Random Forest Classifier
Random forest is a good classifier, often used and also very efficient. It is an ensemble classifier made using many decision tree models. There are ensemble models that combine the different results. The random forest model can both run regression and classification models.Since this is a classification problem we get a very good accuracy using Random Forest Classifier.

Logistic Regression
Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.
We have used cross validation also which is one of the most important part of accuracy calculation because in machine learning, we could not fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the cross-validation technique and we can observe that accuracy value decreased after using cross validation. This proves that it is more difficult for model to predict unknown dataset.

D. Conclusion

Our model worked properly on every classifier and we got training accuracy and validation accuracy around 95percent. After standardizing data accuracy values increased in some of the classifiers as it depends on dataset. We can also observe that our dataset is very much biased but then also we get a very good accuracy as accuracy depends on test dataset also.Using cross validation accuracy value is decreased which was expected too as our model is tested against unknown dataset.It is always tough for any model to predict unknown dataset as compared to known dataset.

V. Contribution of each members

Sandip (B19CSE075)
Data Preprocessing, Data training,Data Visualisation, Report

Nandini (B19CSE057)
Data Preprocessing , Classifiers selection, Standardization of dataset,
Rohit (B19EE073)
Research the dataset and other required informations, Report analysis, Cross Validation, Readme file

All the members worked very hard and contributed as much as they can.

VI. Acknowledgment

We would like to express our special thanks to our Professors Dr. Richa Singh, Dr. Romi Banerjee and Dr. Yashashwi Verma for their efforts to teach such a knowledgeable course of Pattern Recognition and Machine Learning which helped us a lot in research fields. In this course we came to know about so many new things which we did not know earlier. We are really very grateful to them. We would also like to show our gratitude towards our TAs who helped us a lot during the entire course within the limited time period.

References

https://github.com/hvp004/twitter-sentiment-analysis
https://www.youtube.com/watch?v=ujId4ipkBio https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html