Sentiment Analysis

Jitender Kumar

Introduction

**Sentiment analysis** (or **opinion mining**) is a natural language processing technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

Table of Content

1. Importing Dataset
2. Pre-processing and Cleaning Data
3. Analysing data
4. Splitting data
5. TF – IDF Vectoriser
6. Transforming Dataset
7. Training and Evaluating Model
8. Using the model for Predictions

Importing Dataset

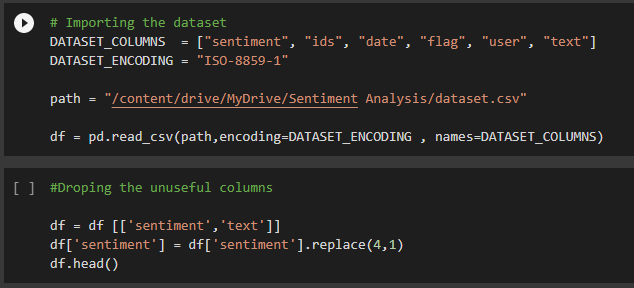
The dataset being used is the **sentiment140 dataset**. It contains 1,600,000 tweets extracted using the **Twitter API**. The tweets have been annotated **(0 = Negative, 4 = Positive)** and they can be used to detect sentiment.

**It contains the following 6 fields:**

1. **sentiment**: the polarity of the tweet *(0 = negative, 4 = positive)*
2. **ids**: The id of the tweet
3. **date**: the date of the tweet
4. **flag**: The query. If there is no query, then this value is NO\_QUERY.
5. **user**: the user that tweeted
6. **text**: the text of the tweet

We require only the **sentiment** and **text** fields, so we discard the rest.

Furthermore, we're changing the **sentiment** field so that it has new values to reflect the sentiment. **(0 = Negative, 1 = Positive)**

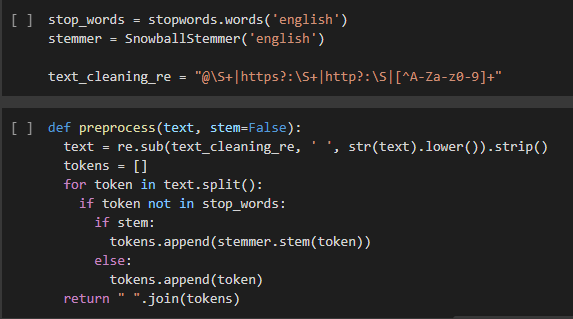


Pre Processing and Cleaning Data

Data pre processing is an essential step in building a Machine Learning model and depending on how well the data has been pre processed; the results are seen. In NLP, text pre processing is the first step in the process of building a model.

The various text pre processing steps are:

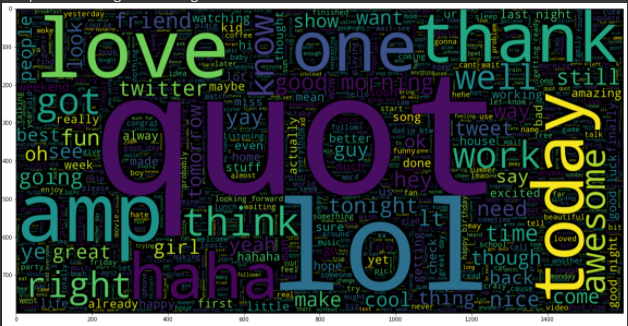
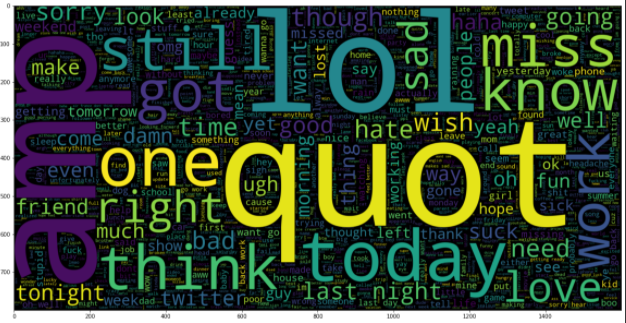
1. Tokenization : Splitting the sentence into words.
2. Lower casing :  Converting a word to lower case (NLP -> nlp).
3. Stop words removal : Stop words are very commonly used words (a, an, the, etc.) in the documents. These words do not really signify any importance as they do not help in distinguishing two documents.
4. Stemming :  It is a process of transforming a word to its root form.
5. Replacing URLs: Links starting with "http" or "https" or "www" are replaced by "URL".
6. Removing Non-Alphabets: Replacing characters except Digits and Alphabets with a space.

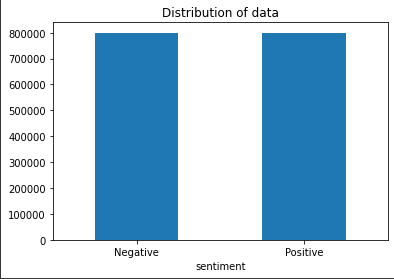


Analysing the Data

Now we're going to analyse the pre processed data to get an understanding of it. The data set is not skewed as the there are equally numbers of positive and Negative tweets.

We'll plot **Word Clouds** for **Positive and Negative** tweets from our dataset in order to visualize the text data such that the size of each word indicates its frequency or importance.

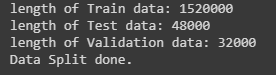
 



Splitting the Data

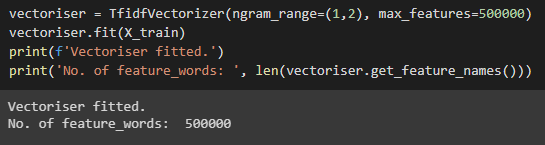
The Pre processed Data is divided into 2 sets of data:

* **Training Data:** The dataset upon which the model would be trained on. Contains 95% data.
* **Test Data:** The dataset upon which the model would be tested against. Contains 3% data.
* **Validation Data:** The dataset upon which the model would made make prediction. Contains 2 % of the data



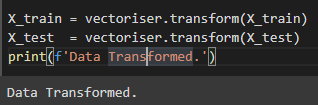
TF-IDF Vectoriser

**TF-IDF indicates what the importance of the word is in order to understand the document or dataset.** Let us understand with an example. Suppose you have a dataset where students write an essay on the topic, My House. In this dataset, the word a appears many times; it’s a high frequency word compared to other words in the dataset. The dataset contains other words like home, house, rooms and so on that appear less often, so their frequency are lower and they carry more information compared to the word. This is the intuition behind TF-IDF.



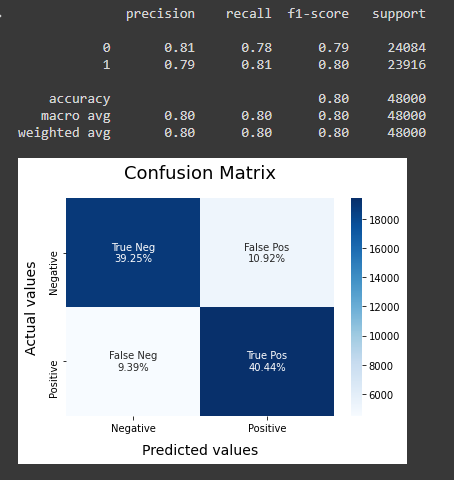
Tranforming the dataset

Transforming the **X\_train** and **X\_test** dataset into matrix of **TF-IDF Features** by using the **TF-IDF Vectoriser**. This datasets will be used to train the model and test against it.



Creating and Evaluating Models

We're using Logistic Regression for our sentiment analysis problem. Since our dataset is not **skewed**, i.e. it has equal number of **Positive and Negative** Predictions. We're choosing **Accuracy** as our evaluation metric. Furthermore, we're plotting the **Confusion Matrix** to get an understanding of how our model is performing on both classification types.

Since the model is not performing well and give out large number of False Positive and False Negative.

While making prediction we will be classifying tweets as neutral which are predicted with low confidence by the model.

Using the Model for making Predictions

Since the model is not performing well and give out large number of False Positive and False Negative. While making prediction we will be classifying tweets as neutral which are predicted with low confidence by the model.

To use the model for **Sentiment Prediction** we will need the **Vectoriser** and **LR Model**. The vectoriser can be used to transform data to matrix of TF-IDF Features. While the model can be used to predict the sentiment of the transformed Data. The text whose sentiment has to be predicted however must be pre processed.

