****

**IME672A: Data Mining and Knowledge Discovery**

**Twitter Sentiment Analysis**

**Group 17 - Term Report**

[**Data Link**](https://www.kaggle.com/kazanova/sentiment140)

**170117 ANIRUDH GARG**

**170402 MUDIT AGRAWAL**

**170818 YASH MITTAL**

**170478 PRAGYAN PANDEY**

**170625 SANIDHYA SHAH**

**------------------------------------------------------------------------------------------------**

**Contents:**

1. Problem Description
2. Data Understanding
3. Methodology
   1. Text Cleaning
   2. Visualizing Data
   3. Models Used
      1. Multinomial Naive Bayes
      2. Logistic Regression
      3. Long Short Term Memory (LSTM)
4. Results and their Interpretation

# **Problem Description**

With the quick increment in the quantity of web clients, the Internet has an enormous measure of data produced by the clients. Many people share their views regarding a topic on social media platforms such as Facebook and Twitter and give their feedback or review about a product on e-commerce web sites such as Amazon and Flipkart which leads to a huge amount of data. The identification of subjective statements from the data is known as subjectivity detection. To automate the analysis of such data, sentiment analysis is used. The aim is to find the opinionative data and classify it according to its polarity, i.e. positive, negative or neutral feedback, known as sentiment classification and then analysing it which is known as sentiment analysis. However, before performing sentiment examination, the information is exposed to different pre - processing procedures which finally give the desired optimized output. This allows us to get to know about the public’s mood or opinion about a particular topic. This helps the concerned organization or public to improve their product or service based on the feedback received.

**Data Understanding**

This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated (0 = negative sentiment, 4 = positive sentiment) and they can be used to detect sentiment.

It contains the following 6 fields:

1. **target**: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive
2. ids: The id of the tweet (*2087*)
3. date: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. flag: The query (*lyx*). If there is no query, then this value is NO\_QUERY.
5. user: the user that tweeted (*robotickilldozr*)
6. **text**: the text of the tweet (*Lyx is cool*)

Most of them are not useful for sentiment analysis. Only the **target and text** are useful. Also, there is **no missing data**.

**Methodology**

## **Text Cleaning**

#### **Cleaning retweets, mentions, and links**

Retweets are comments or previous tweets that are re-posted again or copied for your own followers to read on twitter. Such tweets are prefixed with *rt*. Similarly, tweets and retweets may contain mentions and URL links of other users and websites.

#### 

#### 

#### **Removing Punctuation and other special characters**

#### Punctuation and other special characters don’t add any value in sentiment analysis of a tweet. This is because data on social platforms is written in a very informal way hence even if we want to make use of punctuation, it is not guaranteed that they are written judiciously in tweets and thus using them can even deteriorate our Machine Learning models.

#### 

#### **Removing numbers and text specific characters**

By just having a peek at the data set, it was observed that the frequency of numeric digits is very less. For machine learning models to learn input data efficiently it should be present in an ample amount. So there are two choices, either we can convert numbers into their text form or remove them entirely.

#### 

#### **Converting Emojis to text and empty strings**

Emojis in themselves conveys a lot of information regarding the sentiment of a sentence. Though currently there is no effective way to make use of them in machine learning models. One way can be to convert emoji into text representation but then not every emoji is present syntactically in sentences and their converted textual forms may not provide any direct meaning with the sentence. For example - *"never use zomato food service deliver worst quality food help desk executive politeness ."*

This may convert into: *"never use zomato food service deliver worst quality food help desk executive politeness* ***sad****."*

Another example: *"I am really * *with Swiggy services"*

This may convert into: *"I am really* ***happy*** *with Swiggy services"*

#### **Removing stopwords and unit length characters**

A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

Stopwords use up storage and also consume our valuable processing time. Even then, we are not sure how stop words affect a deep learning model. So I have processed the text with and without stopwords and then choose the best resulting model out of them.

### 

## **Visualizing Data**

## 

## **Word Cloud**

Word clouds (also known as text clouds or tag clouds) work in a simple way: the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud. This graph provides a textual analysis and a general idea of which type of words are present more frequently in our tweets.

**Scatter Text**

Scattertext is a Python package that lets you interactively visualize how two categories of text are different from each other. This tool is able to display thousands of visible term-representing points and find space to legibly label hundreds of them. Scattertext also lends itself to a query-based visualization of how the use of terms with similar embeddings differs between document categories, as well as a visualization for comparing the importance scores of bag-of-words features to univariate metrics. It also shows the list of top words of both categories.

## **Models Used:**

**Multinomial Naive Bayes**

Multinomial Naive Bayes uses term frequency i.e. the number of times a given term appears in a document. Term frequency is often normalized by dividing the raw term frequency by the document length. After normalization, term frequency can be used to compute maximum likelihood estimates based on the training data to estimate the conditional probability.

There is one **hyper parameter** (alpha) which used to have non-zero probabilities for all the terms. Alpha is computed through the use of **Area under ROC curve (AUC)**. After finding the alpha we trained the model and **again used the AUC to find the best threshold** **for classifying the positive and negative sentiment**.

**Logistic Regression**

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) which in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex extensions exist. In regression analysis, logistic regression is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an [indicator variable](https://en.wikipedia.org/wiki/Indicator_variable), where the two values are labelled "0" and "1".

**Long Short-Term Memory (LSTM)**

A long short-Term Memory, or LSTM, is a type of machine learning neural networks. More specifically, it belongs to the family of Recurrent Neural Networks (RNN) in Deep Learning, which are specifically conceived in order to process *temporal data*. Temporal data is defined as data that is highly influenced by the order that it is presented in. This means that data coming before or after a given datum (singular for *data*) can greatly affect this datum. We use word embeddings which are basically a way for us to convert words to *representational vectors*. We do that using the GloVe functionality. The model can be improved (overfitting can be removed) via regularization and adding dropout layers.

**Results**

**Pre - Processing Result**

Basic Cleaning steps as mentioned above, results in following text changes.

**Original text:-** Never use @Zomato food service, **they** deliver worst quality food and **there** help desk executive had no politness.. .  https://t.co/0vM6OJadjP

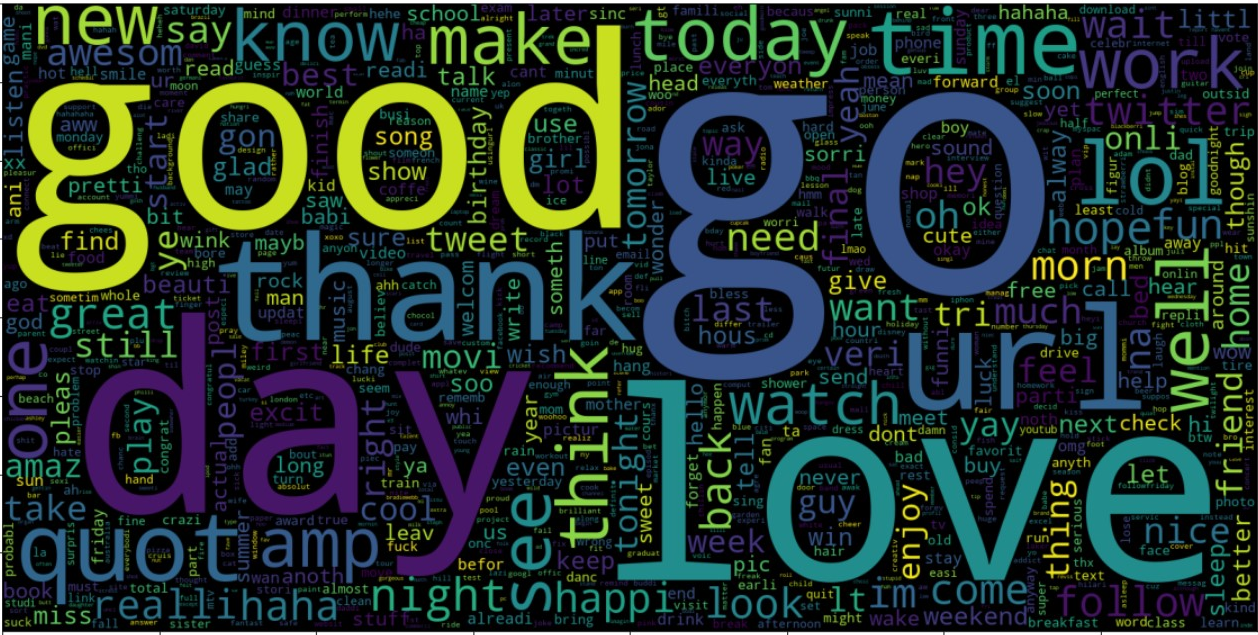
**Pre-processed:-**

**Without Stemmer - “**never use food service deliver worst quality food help desk executive politness **sad URL”**

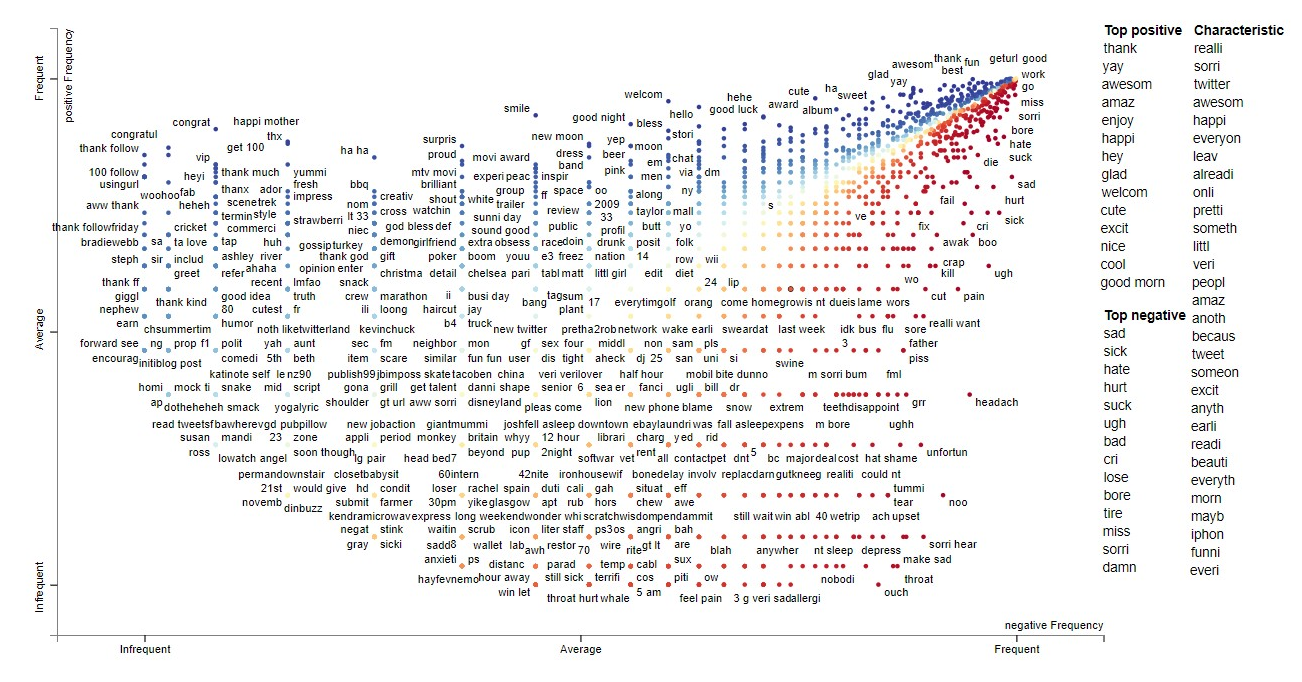
**With Stemmer (This gives more accuracy on test data) - “**never use food servic deliv worst qualiti food help desk execut polit **sad url”**

## **Visualizing Results**

## **Word Cloud (For Positive Sentiment)**



### **Scatter Text**



X-axis represents negative frequency while Y-axis represents positive frequency.

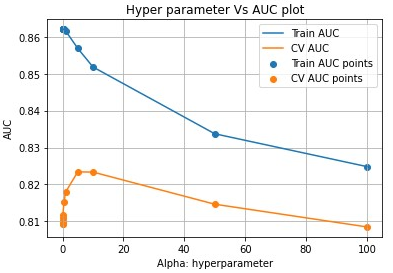
Blue Dots are those words which appear more in positive tweets

Red Dots are those words which appear more in negative tweets

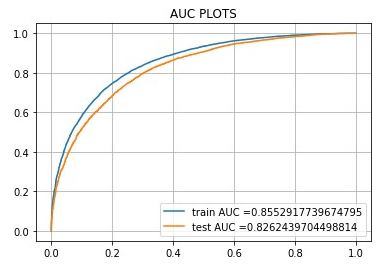
In the right we see the list of top positive and top negative words

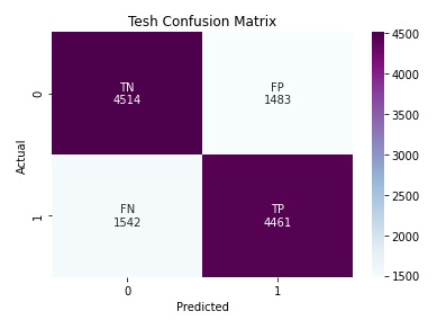
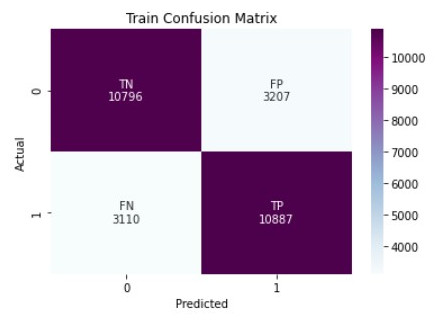
**Results**

**Multinomial Naive Bayes**



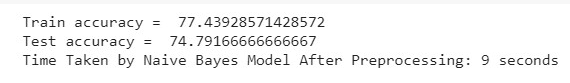
This shows AUC is maximum at alpha = 5 in Cross Validation.

The maximum value of tpr\*(1-fpr) 0.599 for threshold 0.5.



Training Confusion Matrix Test Confusion Matrix

**Accuracy:**



**Logistic Regression**

**Accuracy** - 74.27%

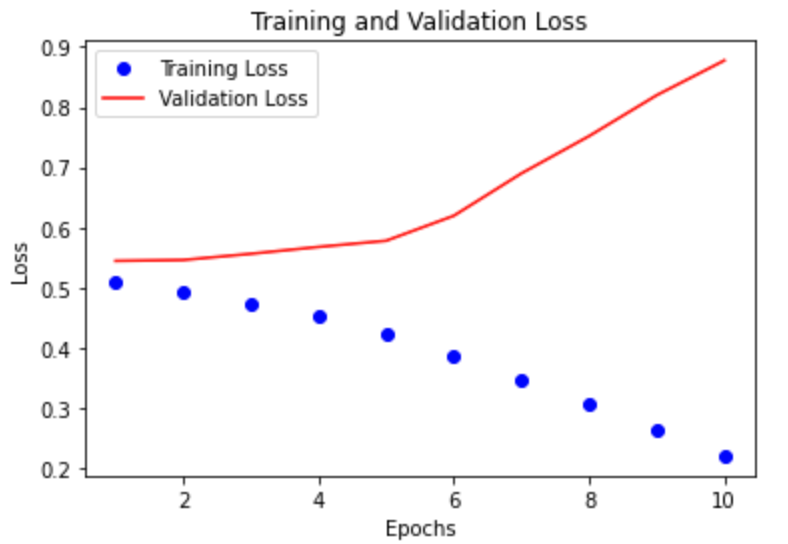
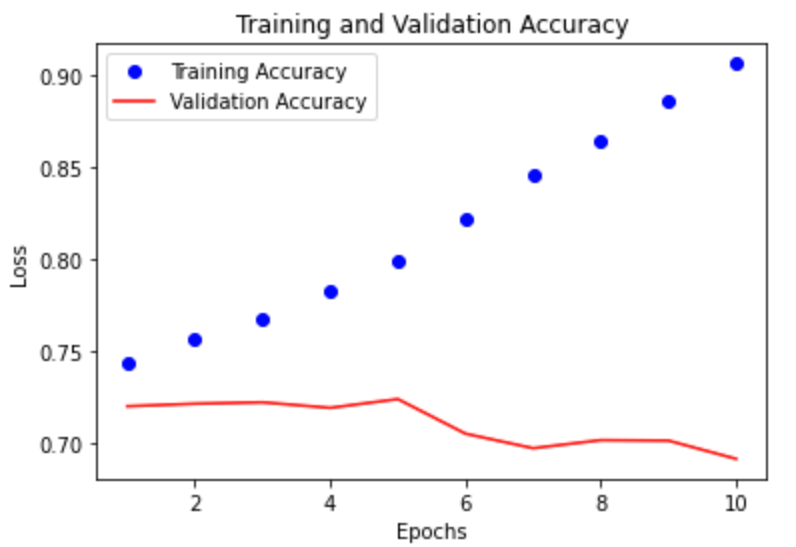
A picture containing chart

Description automatically generatedTable

Description automatically generated

**Long - Short Term Memory**

**Maximum Accuracy –** 72.47%

****

**Interpretations**

* We have also interpreted from the data that more users tweet positively in the non-working hours (early mornings and late nights) while the tweets during the working hours (9 am to 7 pm) are majorly negative
* The users tweet more positively during the weekends than the weekdays
* We get the best results at alpha = 5, in the multinomial naïve bayes model as shown in the figure. For values of alpha greater than 5, the model tends to underfit
* The multinomial naïve bayes model has a better accuracy than both the rest of the models with a decent training time.
* We can see that in LSTM, the validation accuracy first increases and then decreases which gives us an indication that over-fitting might be taking place in this model
* To prevent the above-mentioned overfitting, we have implemented dropout layers, but there was no appreciable change in accuracy.
* Thus, the best model which we can implement is the multinomial naïve bayes model