**Report on Sentiment analysis on Facebook data**

**Data collection:**

Used data for the assignment is being collected through an application Facepager which use FB api to scrape data.

<https://github.com/Sudhakordas/Sentiment_analysis/blob/master/Amajonnew.csv>

**Data pre-processing**

This is the process of filtering the data from all types of noise. Here are the some steps taken while doing this

1.Removing hash tags and ats.

2. Removing url links.

3. Removing emojis and emoticons

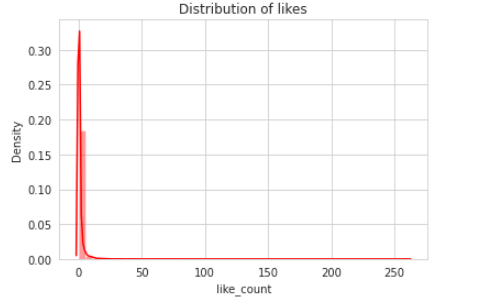
4. Removing stop words.

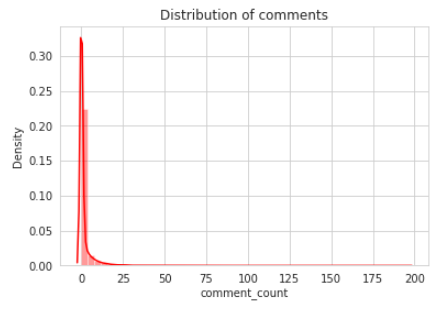
5. Last but not the least is lemmatization of the text to bring it into base form.

**Feature creation**

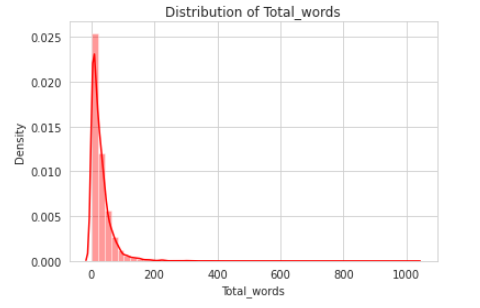
Sometimes considered it as a part of feature engineering which is the process of creating new feature out of those existing ones. Few created features are Now of words in each row. Separate features for text without stop words, hash tags, ats, emojis.

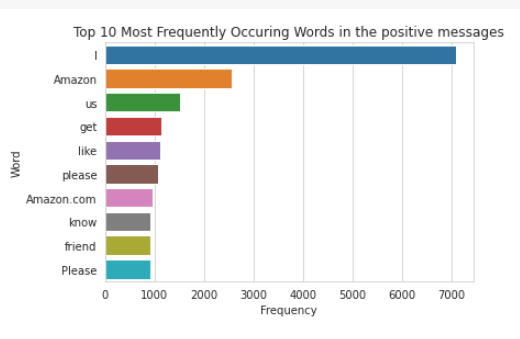
EDA

Meaningful insights, pattern extraction is the outcome of this steps.



EDA





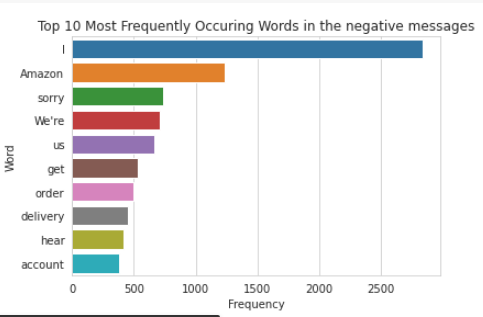


Fig: Most common Word frequencies on both Positive and Negative sentiments

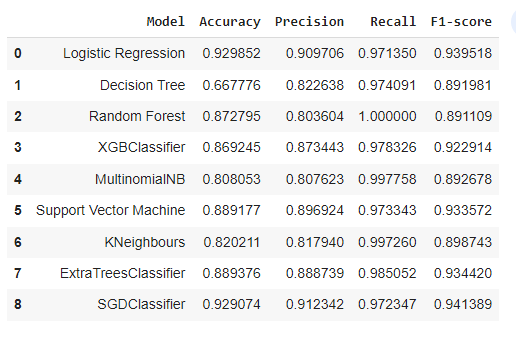
Scrapped data was unlabelled, but to train a ml model it requires labelled data, to counter this issue there are several techniques out there and polarity of each row of text is being calculated to derive the sentiment using text blob library.

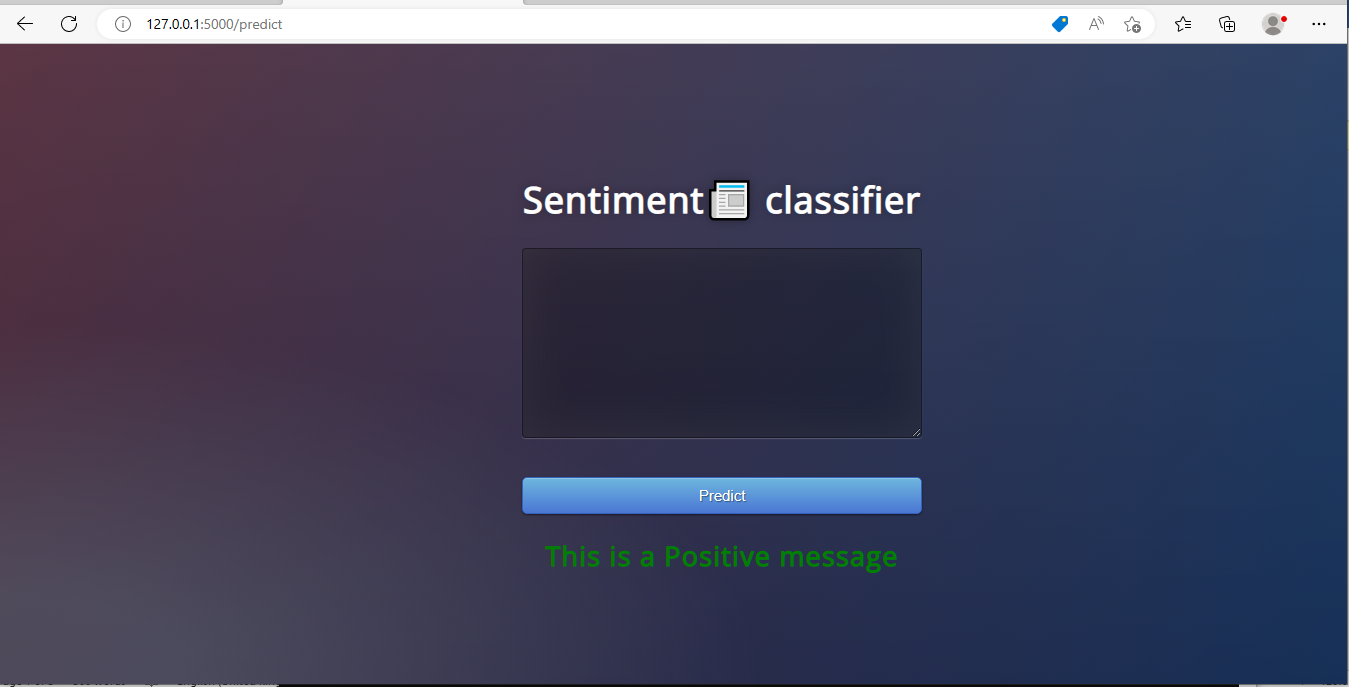
**Feature extraction**

ML models does not understand anything rather than the numerical values out of many feature extraction techniques tf-idf vectorizer is being implied here for converting the text into numeric value.

**Model training and evaluation:**

Several machine learning models is being trained and evaluated on the train and test data correspondingly on some classification matrix.





Finally convert this whole system into a web app using Flask and basic html and CSS.

Still there are lot’s of scope extend this work like using deep learning and Transformer based models with various types of word embedding techniques, for instance Word2Vec, Fast text etc.