Twitter Sensitive Analysis

1. Problem Understanding and Definition

Definition of the Problem Statement

Twitter Sentiment Analysis involves analyzing text data from Twitter to determine the sentiment expressed in each tweet. The core objective is to categorize tweets into three sentiment classes: negative (-1), neutral (0), and positive (1). This classification reflects the emotional tone or attitude conveyed in the tweet regarding a topic, person, or entity.

Significance of the Problem

Companies can use sentiment analysis to gauge public opinion about their brand, products, or services for business insights, market analysis, political campaigns and social research.

Identification of the Target Variable and Relevant Features

The target variable in this context is 'label', representing the sentiment of the tweet. It is a categorical variable with three categories: -1 (negative), 0 (neutral), and 1 (positive), and the primary feature for this analysis is the 'Tweet' text.

Framing the Problem in the Context of Machine Learning

This problem is best framed as a Supervised Learning task as the the dataset contains labeled examples to classify each tweet into one of the predefined categories (negative, neutral, positive).

2. Data Collection and Preprocessing

Data Sources

The primary data source for Twitter sentiment analysis is a dataset containing tweets.

Data Collection Methods

Pre-existing Datasets: Leveraging datasets available on platforms like Kaggle, which may have pre-collected and labeled tweets.

Data preprocessing steps and its justification

- Handling Missing Values:

According to the dataset's size and the missing data's nature, we imputed missing values to avoid skewing the analysis of missing values.

- Data Normalization:

We convert all text to lower case, removing special characters, URLs, and numbers, and possibly replacing or removing emojis and hashtags.

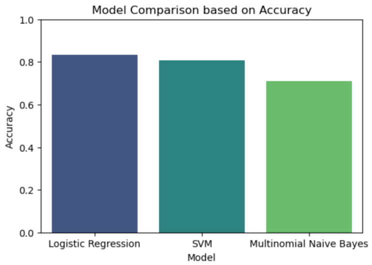
- Feature Engineering:

We convert text into numerical form using technique: TF-IDF to represent each sentence as a vector with each word represented as 1 for present and 0 for absent from the vocabulary.

3. Model Selection , Evaluation and Interpretation

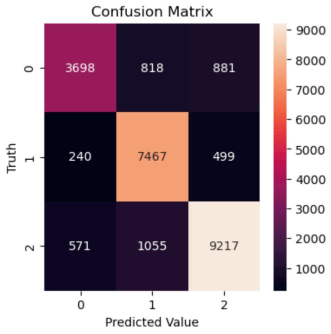
First, we built three different machine learning models: *Multinomial Naive Bayes*, *Logistic Regression*, and *Support Vector Machines*.

Then, we tuned the hyperparameters for each model individually. The tuning process aims to find the optimal set of hyperparameters to maximize accuracy and confusion matrices. We found that the Logistic Regression model exhibited the best overall performance.



4. Conclusion

Finally, we use the test set to further validate the performance of the best model. We get an accuracy score of 83.38% on the test set. From the confusion matrix of the best model, we can observed that the diagonal elements are much high relative to the off-diagonal elements, it means that this model is performing well. Therefore, this performance gives us confidence in the model's ability to generalize to real-world scenarios. You can find more details and explanations for this project from the Jupyter notebook.



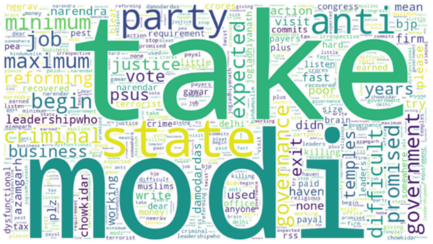
Here is the wordCloud picture of positive, neutral and negative words.

will
modi
now
prop
given
attack
promised
vote
say
bjp
soldiers
welcome
india
rahul
told
main
rss

Positive

india
modi
talk
nonsense
drama
refresh
karo
project
continue
comment
women
will
kiya
surat
vote

Neutral



Negative

5. Scope for future work

For future work, we can do data preprocessing differently by using Bag of Words (BOW) or Count Vectorizer. We can make a list of unique words in the text corpus called vocabulary. Then we can represent each sentence or document as a vector with each word represented as 1 for present and 0 for absent from the vocabulary.