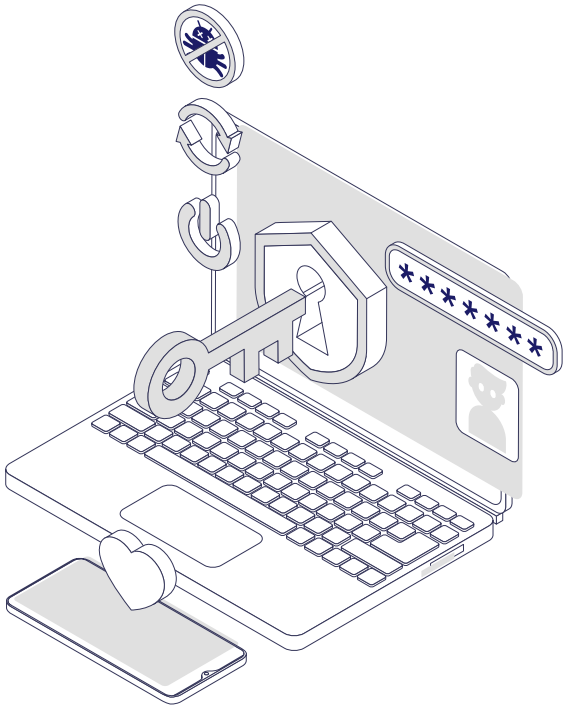
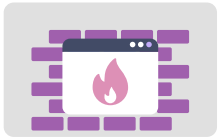


TWITTER/X SENTIMENT ANALYSIS

Team : ELITE

Introduction



People frequently use Twitter as a means of expressing their thoughts and showing their emotions on a variety of events.

Sentiment analysis is a method for analyzing data and locating the sentiment that it contains.

It is a natural language processing problem where sentiment analysis is carried out by separating positive tweets from negative tweets using machine learning models for classification, text mining, text analysis, data analysis, and data visualization.

Dataset : Twitter
Dataset Description: The dataset contains two columns. One is the id and next is the tweet. given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist

Objective

The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.

Requirement

This approach will be executed using Python programming, leveraging libraries such as NLTK, CSV, Time, Json, RE, Twitter, and word cloud. The data will be obtained from the Twitter dataset, with the Twitter API serving as the key source of data.

Potential challenges

A notable challenge revolves around effectively leveraging the Twitter API and Tweepy to access real-time tweets while also retrieving historical offline tweets from the past. This dual-purpose data acquisition process requires careful implementation and coordination.

Project Pipeline

01

Import Necessary Dependencies

02

Read and Load the Dataset

03

Exploratory Data Analysis

04

Data visualization of Target Variables

05

Data Preprocessing

06

Splitting our data into Train and Test sets

07

Transforming Dataset using TF-IDF Vectorizer

08

Function for Model Evaluation

09

Model Building

10

Model Evaluation

Models Used

Support Vector Machine

SVM aims to find a hyperplane that best separates different classes in the feature space.The goal of SVM is to maximize the margin between the support vectors of different classes while still correctly classifying as many data points as possible

Random Forest

Random forests construct decision trees using randomly chosen data samples. By aggregating predictions from each tree and selecting the most favorable outcome through a voting process, the algorithm produces accurate results. A forest consists of individual trees, and a greater number of trees contributes to enhanced forest robustness.

XGBoost

XGBoost (Extreme Gradient Boosting) is a high-performance machine learning algorithm that enhances predictive models through iterative improvement. It belongs to the gradient boosting family and is specifically designed to improve the accuracy of predictive model

Evaluation Criteria

Accuracy

Accuracy is a performance metric used to assess the correctness of a classification model. In other words, accuracy indicates how well the model's predictions align with the true labels.

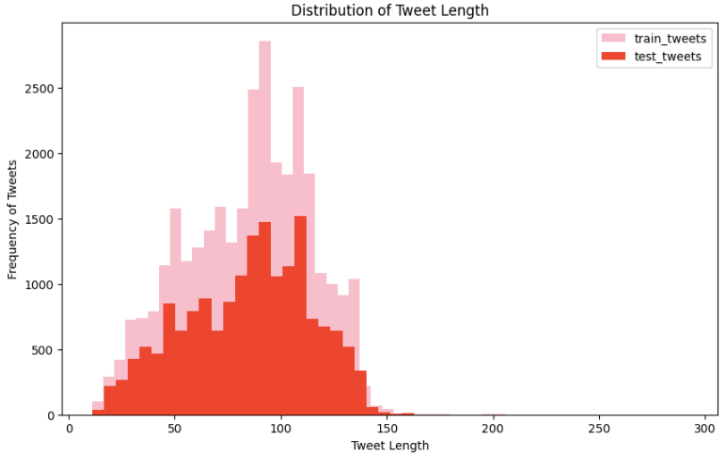
F1 score

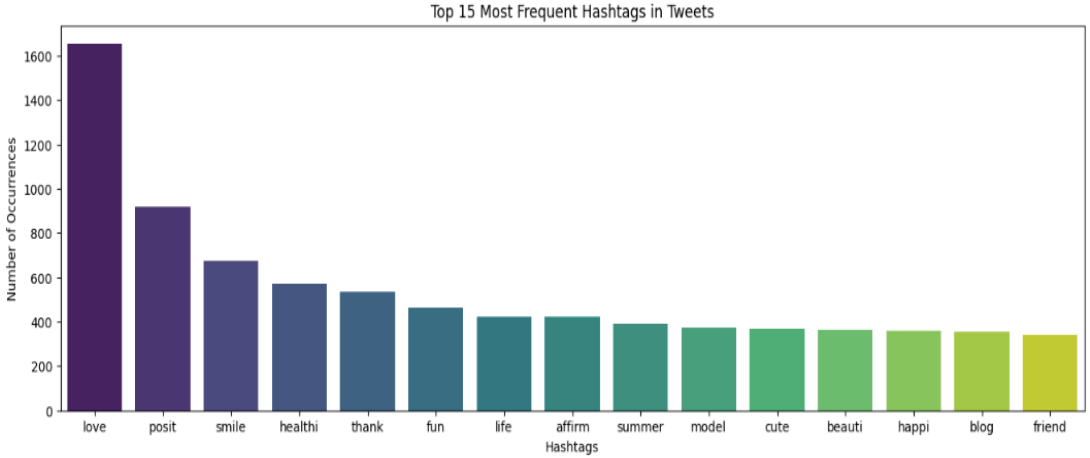
The F1 Score represents the Harmonic Mean between precision and recall, two vital metrics in classification assessment. The F1 Score's value is confined to the range of [0, 1]. When precision is high but recall is low, the classifier is accurate but may overlook a considerable number of challenging instances.

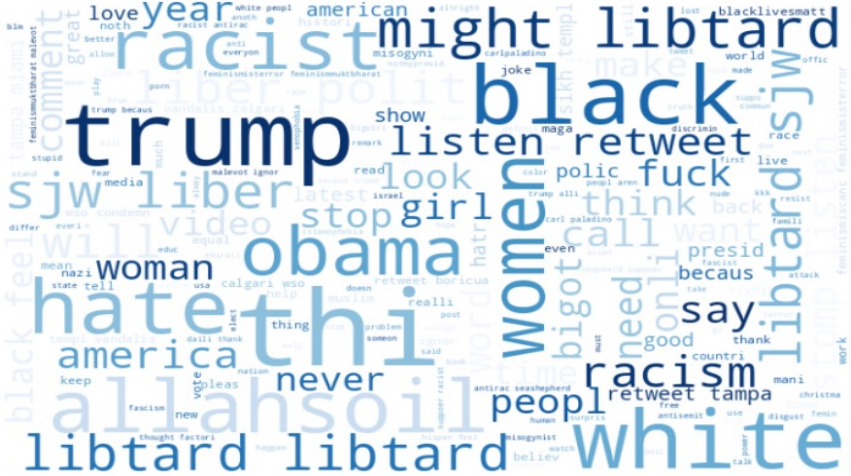
Hyperparamter Tuning

Parameters which define the model architecture are referred to as **hyperparameters** and thus this process of searching for the ideal model architecture is referred to as *Hyperparameter tuning*.

Data visualization







Conclusion

We employed multiple training models to make predictions and endeavored to identify the most suitable model for our dataset. By utilizing distinct vectorization methods such as TF-IDF and Count Vectorizer, we observed variations in the outcomes.

In evaluating the performance of diverse models like SVM, Random Forest, and XGBoost across a range of extracted features including Bag of Words, Word2Vec, Doc2Vec, and TF-IDF, we considered the F1 score as our evaluation metric.

Among these models, our top-performing one proved to be XGBoost, utilizing tuned parameters applied to Word2Vec features, achieving an F1 score of 0.66 and accuracy of 0.96.

	Model_ID	F1Score	Accuracy
0	XGBOOST	0.657168	0.959850
1	Support Vector Classifier	0.623839	0.949317
2	Random Forest Classifier	0.519535	0.952550