# Twitter Sentiments Analysis Using Machine Learning

## Objectives

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.

Formally, 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, your objective is to predict the labels on the given test dataset.

## Text preprocessing

Preprocessing text data is a crucial step since it prepares the unprocessed text for mining, making it simpler to extract information from the text and use machine learning algorithms on it. There is a greater likelihood that you will be working with chaotic and erratic data if we miss this stage.

The goal of this phase is to remove any irrelevant information from the tweets, such as punctuation, special characters, numerals, and terms that don't add much meaning when read in context.

## Text representation

In this project, we used the Bag of Words Technique. It is one of the most used text vectorization techniques. It is mostly used in text classification tasks. Bag of words is a little bit similar to one-hot encoding where we enter each word as a binary value and in a Bag of words we keep a single row and entry the count of words in a document. So we create a vocabulary and for a single document, we enter one entry of which words occur and how many times in a document. Let us get to IDE and implement the Bag-of-words model using the Count vectorized class of scikit-learn.

### Advantages

1. Simple and intuitive – Only a few lines of code are required to implement the technique.
2. It ignores the new words and takes only words which are vocabulary so creates a vector of fixed size.

### Disadvantages

1. Out of vocabulary situation – It keeps count of vocabulary words so if new words come in a sentence it simply ignores it and tracks the count of the words that are in the vocabulary. But what if the words it ignores are important in predicting the outcome this is a disadvantage, the only benefit is it does not throw an error.
2. Sparsity – when we have a large vocabulary, and the document contains a few repeated terms then it creates a sparse array.

## Model Building

The simplest and fastest classification approach for a sizable amount of data is Naive Bayes. Naive Bayes classifier is successfully utilized in numerous applications, including spam filtering, text classification, sentiment analysis, and recommendation systems. It uses the Bayes probability theorem for unknown class prediction. When used for textual data analysis, such as Natural Language Processing, the Naive Bayes classification yields good results.

This model applies Bayes theorem with a Naive assumption of no relationship between different features. According to Bayes theorem:

Posterior = likelihood \* proposition/evidence

P(A|B) = P(B|A) \* P(A)/P(B)

### Advantages

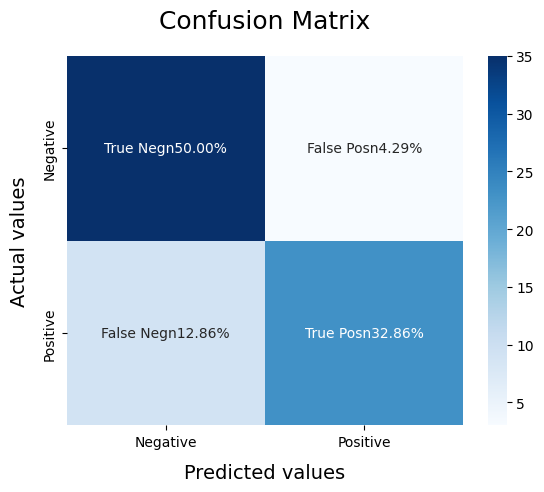
1. Requires a small amount of training data to learn the parameters
2. Can be trained relatively fast compared to sophisticated models

### Disadvantages

1. It’s a decent classifier but a bad estimator
2. It works well with discrete values but won’t work with continuous values (can’t be used in a regression)

## Evaluating the model

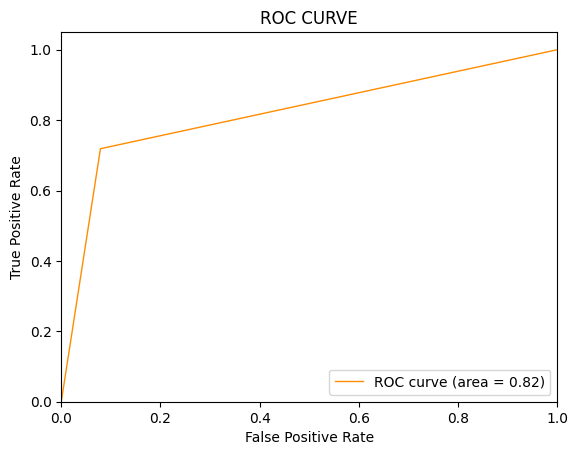
Here we quantify the quality of our model. We use the metrics module from the sklearn library to evaluate the predictions



We used the Receiver Operating Characteristic (R.O.C) curve. It is an evaluation metric for binary classification problems. It is a probability curve that plots the True Positive Rate against False Positive Rate at various threshold values.

The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve.

The ROC AUC score tells us how efficient the model is. The higher the AUC, the better the model’s performance at distinguishing between the positive and negative classes.



We had an AUC value of 0.82 which means the classifier could detect more numbers of true positives and true negatives than false negatives and false positives.