[Fake News Detection]

Scientific Computing Department
Faculty Of Computer and Information Science
Ain Shams University

Team (69)
NLP Project

Under Supervision:
Dr Sally Saad

*From **OT3A** Team*

Name	ID	Section
Osama Anter Mohamed Afify	20191700091	1
Tarek Ashraf Mahmoud Hussein	20191700322	2
Ahmed Mohamed Ibrahim Mohamed	20191700059	1
Adham Mohamed Tawfik Mohamed	20191700086	1
Ahmed Mohamed Ali Abdelrahman	20191700068	1

Idea of Project:

A type of yellow journalism, fake news encapsulates pieces of news that may be hoaxes and is generally spread through social media and other online media. This is often done to further or impose certain ideas and is often achieved with political agendas. Such news items may contain false and/or exaggerated claims and may end up being virtualized by algorithms, and users may end up in a filter bubble.

Dataset:

We have a new CSV file contains 7796 rows and 4 columns:

1. The first column identifies the news.
2. title: represent title of news.
3. text: have news data.
4. label: representing new Belong to fake or real Class.

We Used Columns: Text & Label Only.

1-Preprocessing:

We Use This Techniques to Clean Data and Preprocessing Text:

- Remove Rows Have Nulls Cells
- Removing Duplicate Rows
- text cleaning:
  - Tokenization "Expand Contractions".
  - Remove punctuation.
  - Lower Case.
  - Remove words containing digits "Numbers".
  - Remove Stopwords.
  - Rephrase text → URL.
  - Stemming.
  - Lemmatization.
  - Remove Extra Spaces.
- Label Encoder.

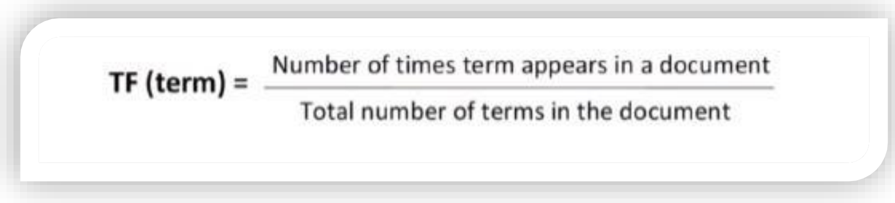
2- Techniques and Feature Extraction:

1- we will use **'TF-IDF Vectorizer'** in our "news" data.

2- We will initialize the classifier, transform, and fit the model and calculate the performance of the model using the appropriate performance matrix/matrices to see how well our model performs.

TF-IDF Vectorizer:

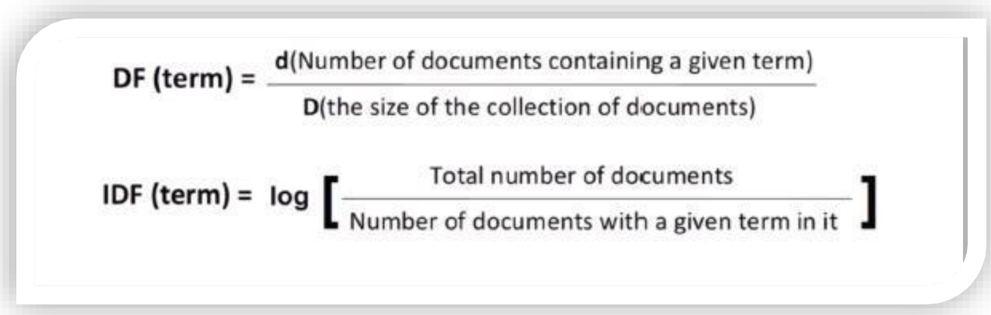
- **TF (Term Frequency):** In the document, words are present so many times that is called term frequency. If you get the largest values, it means that word is present so many times with respect to other words. When you get word is parts of a speech word that means the document is a very nice match.



$$\text{TF (term)} = \frac{\text{Number of times term appears in a document}}{\text{Total number of terms in the document}}$$

- **IDF (Inverse Document Frequency):** in a single document, words are present so many times, but also available so many times in another document also which is not relevant. IDF is a proportion of how critical a term is in the whole corpus.

  collection of word Documents will convert into the matrix which contains TF-IDF features using TF-IDF Vectorizer.



$$\text{DF (term)} = \frac{\textbf{d}\text{(Number of documents containing a given term)}}{\textbf{D}\text{(the size of the collection of documents)}}$$

$$\text{IDF (term)} = \log\left[\frac{\text{Total number of documents}}{\text{Number of documents with a given term in it}}\right]$$

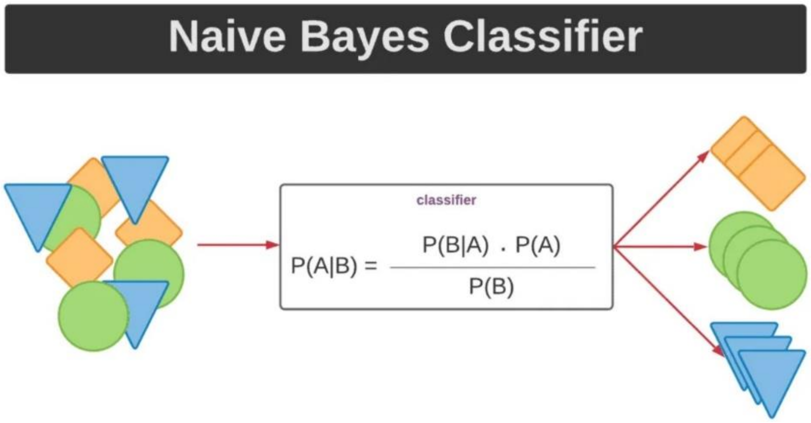
Count Vectorizer:

- Count vectorizer is a method to convert text to numerical data. It Convert a collection of text documents to a matrix of token counts.

3- Classification Models:

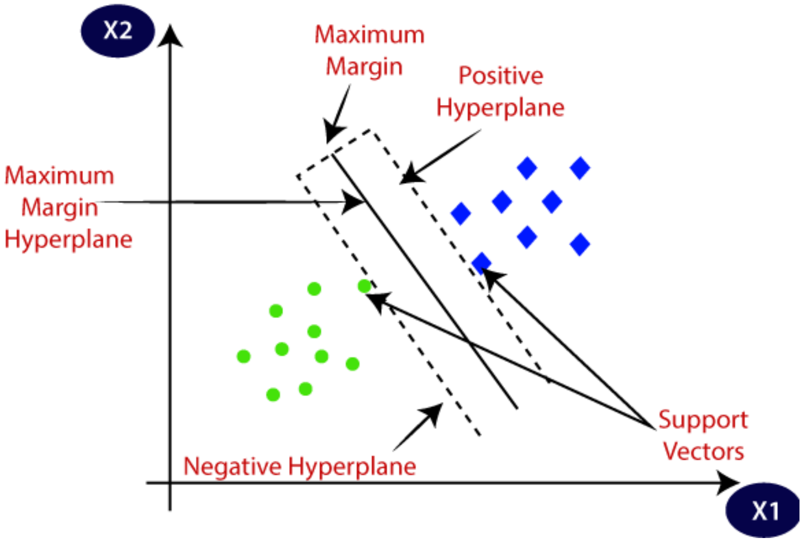
Multinomial Naive Bayes classifier:

- (MNB) is a probabilistic machine learning algorithm and learns the probability distribution of the features in the data for each class.
- Calculating Conditional Probability: For each class, calculate the conditional probability of each feature in the data given the class. This is done by counting the number of occurrences of each feature in the training data for each class and dividing it by the total number of features in that class.
- Calculating Prior Probability: Calculate the prior probability of each class. This is the probability of each class occurring in the training set.



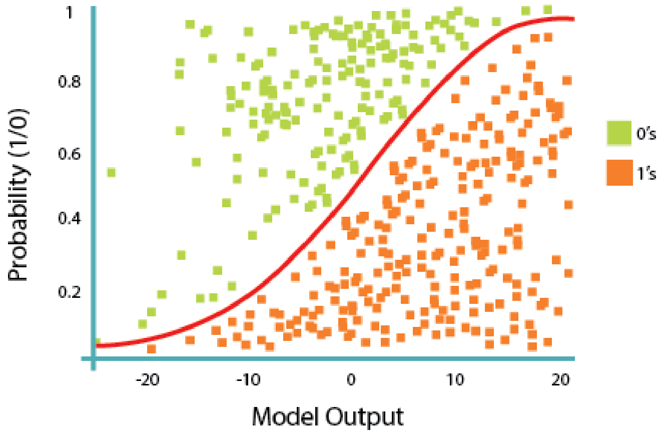
Support Vector Machine Classifier:

SVM is a supervised machine learning algorithm that works by finding the hyperplane that best separates the data points into their respective classes. In text classification, SVM represents each document as a vector of features, which could include word frequencies or the presence of certain keywords or patterns in the text. SVM then finds the hyperplane that maximizes the margin between the two classes and can use it to classify new text documents into one of the classes based on which side of the hyperplane they fall.



- **Logistic Regression Classifier:**

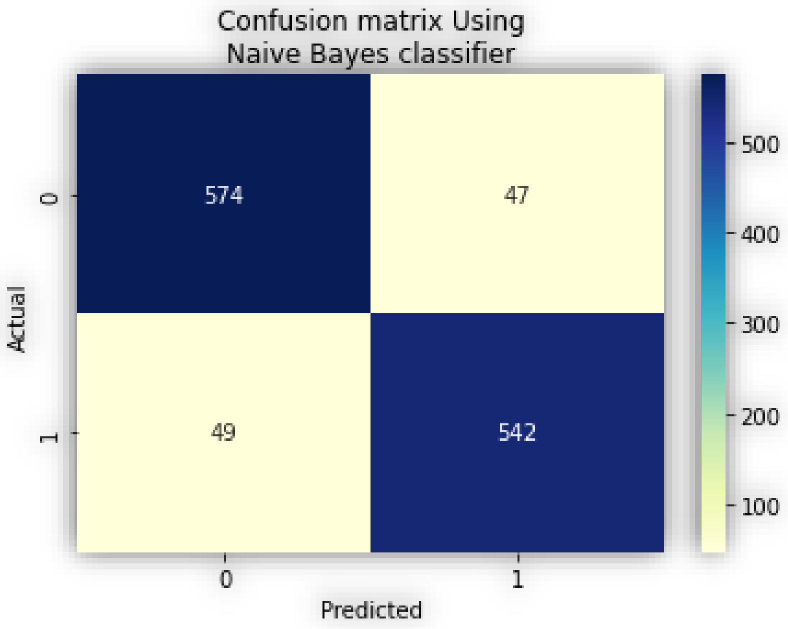
- Logistic Regression models the probability of the input text data belonging to a particular class.
- During training, the weights (coefficients) of the logistic regression model are learned using an optimization algorithm such as gradient descent. The goal is to find the optimal values of the weights that minimize the error between the predicted probabilities and the actual labels.
- The predicted probabilities can be thresholder to obtain discrete class labels. Generally, if the predicted probability of a text document belonging to a class is greater than a certain threshold (usually 0.5), then it is classified as belonging to that class.



4- Results visualization:

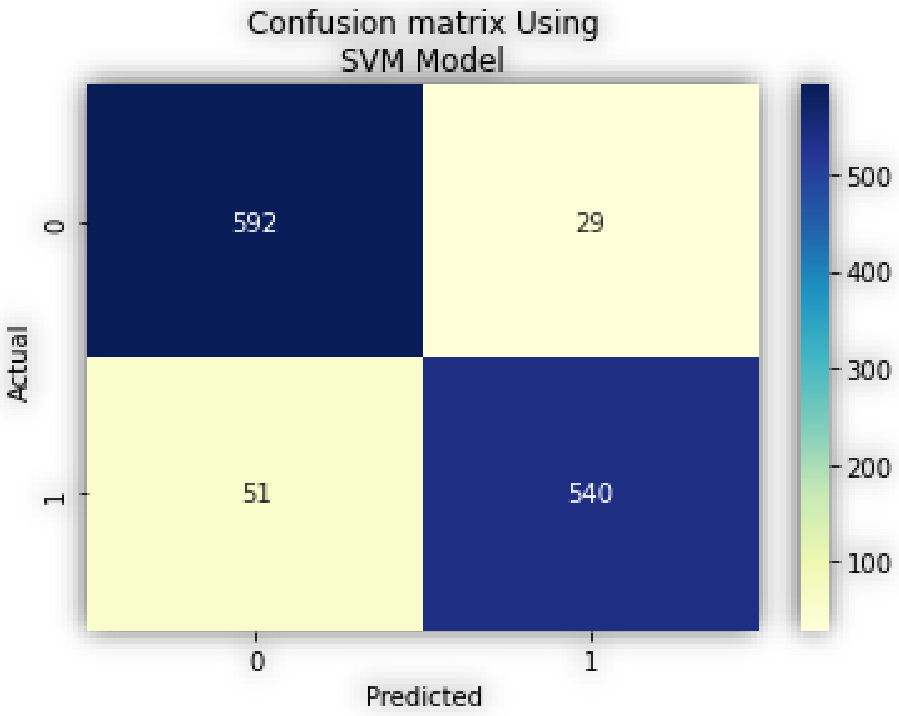
1) Multinomial Naive Bayes classifier:

- Accuracy: 92.07920792079209 %.



2) Support Vector Machine Classifier:

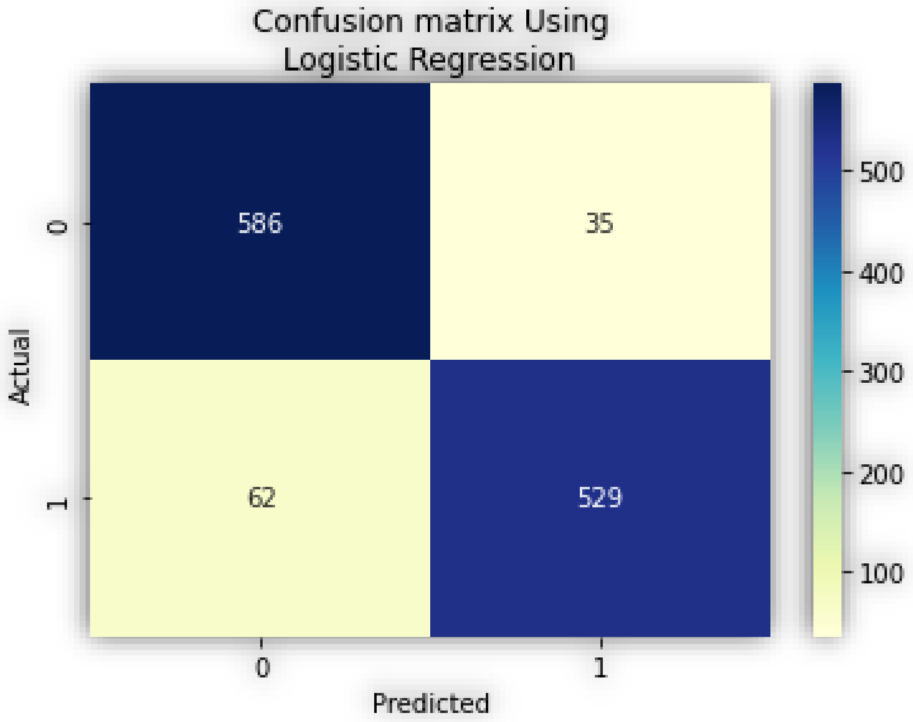
- Accuracy: 92.07920792079209 %



- :

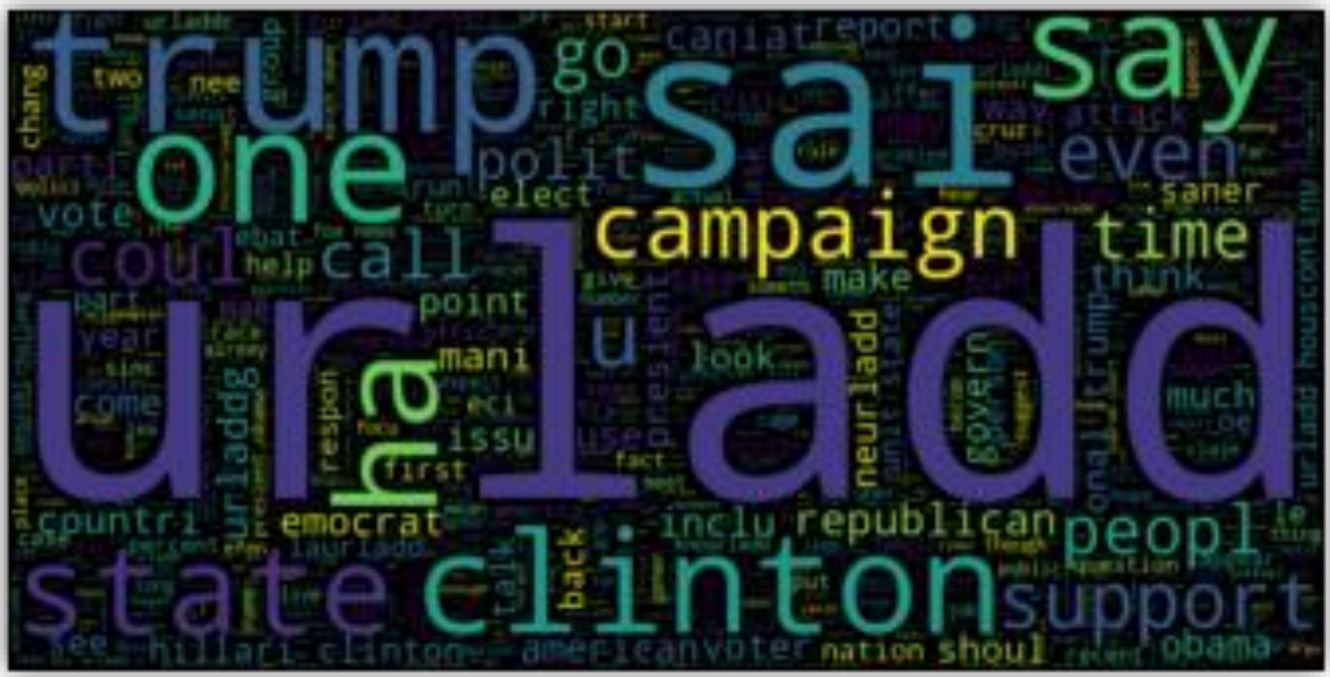
3) Logistic Regression Classifier:

- Accuracy: 91.996699669967 %

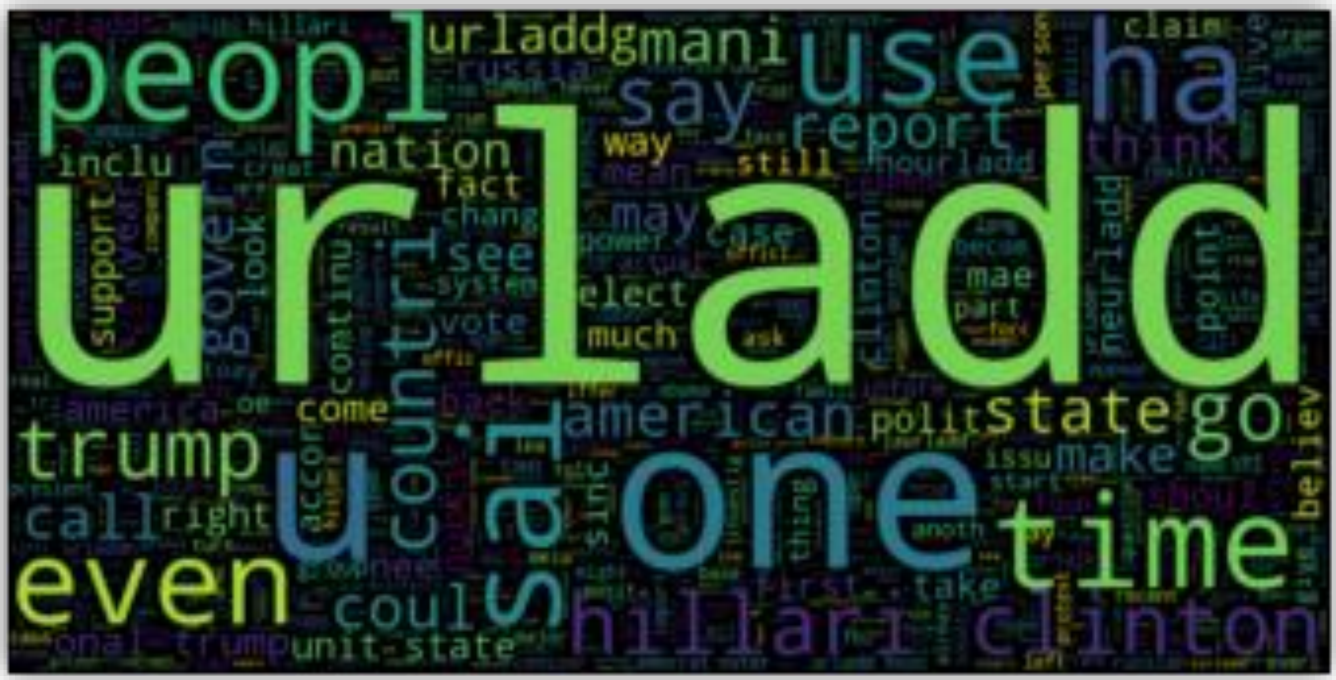


-

Real News Words:



Feck News Words:



Thank You :)