****

**Ministry of Higher Education & Scientific Research**

**University of Sulaimani**

**College of Science**

**Computer Department**

**(Fake News Detection Machine Learning)**

**Prepared By:**

Chapk Rzgar Mohammad

**Supervised by:**

Dr. Miran Taha

Data Science

# Table of Contents:

[Table of Contents: 1](#_Toc157194972)

[Introduction: 2](#_Toc157194973)

[Methodology: 3](#_Toc157194974)

[Test Experiments: 7](#_Toc157194975)

[Comparison with Other Models: 8](#_Toc157194976)

[Results and Discussion: 9](#_Toc157194977)

[Conclusion: 10](#_Toc157194978)

# 

# Introduction:

The proposed project focuses on fake news detection using machine learning, leveraging **Natural Language Processing** (**NLP**) techniques and a **Logistic Regression** model. The motivation behind choosing this project stems from the growing impact of misinformation in today's digital age and I want learning about python and machine learning. With the widespread dissemination of news on various platforms, the ability to discern between authentic and fake news has become increasingly challenging.

The chosen methodology involves the application of text preprocessing techniques, such as stemming and vectorization using **Term Frequency-Inverse Document Frequency** (**TF-IDF**). These steps aim to convert raw textual data into a format suitable for machine learning models. The Logistic Regression model is employed for its simplicity and effectiveness in binary classification tasks.

The significance of resolving the issue of fake news detection is underscored by the potential consequences of misinformation on public opinion, decision-making processes, and societal harmony. Addressing this challenge is crucial for promoting information integrity and ensuring that individuals can make informed judgments based on accurate information.

The research methodology selected for this project, involving NLP techniques and a Logistic Regression model, is deemed optimal for several reasons. Firstly, NLP techniques allow for the extraction of meaningful features from textual data, capturing semantic information that is essential for distinguishing between real and fake news. Additionally, Logistic Regression is well-suited for binary classification tasks, providing a balance between simplicity and performance.

The choice of using TF-IDF for vectorization ensures that the model considers the importance of words within the context of the entire dataset, further enhancing its ability to identify relevant features for classification.

# Methodology:

The methodology employed in this project for fake news detection using machine learning involves a series of structured steps, combining text preprocessing, feature extraction, model training, and evaluation. The primary goal is to develop a reliable model capable of distinguishing between real and fake news articles.

Certainly! Let's break down each line of code:

1. **Importing Libraries:**

import numpy as np: Importing the NumPy library and aliasing it as np.

import pandas as pd: Importing the Pandas library and aliasing it as pd.

import re: Importing the regular expression library.

from nltk.corpus import stopwords: Importing the NLTK library's stopwords corpus.

from nltk.stem.porter import PorterStemmer: Importing the NLTK library's PorterStemmer for word stemming.

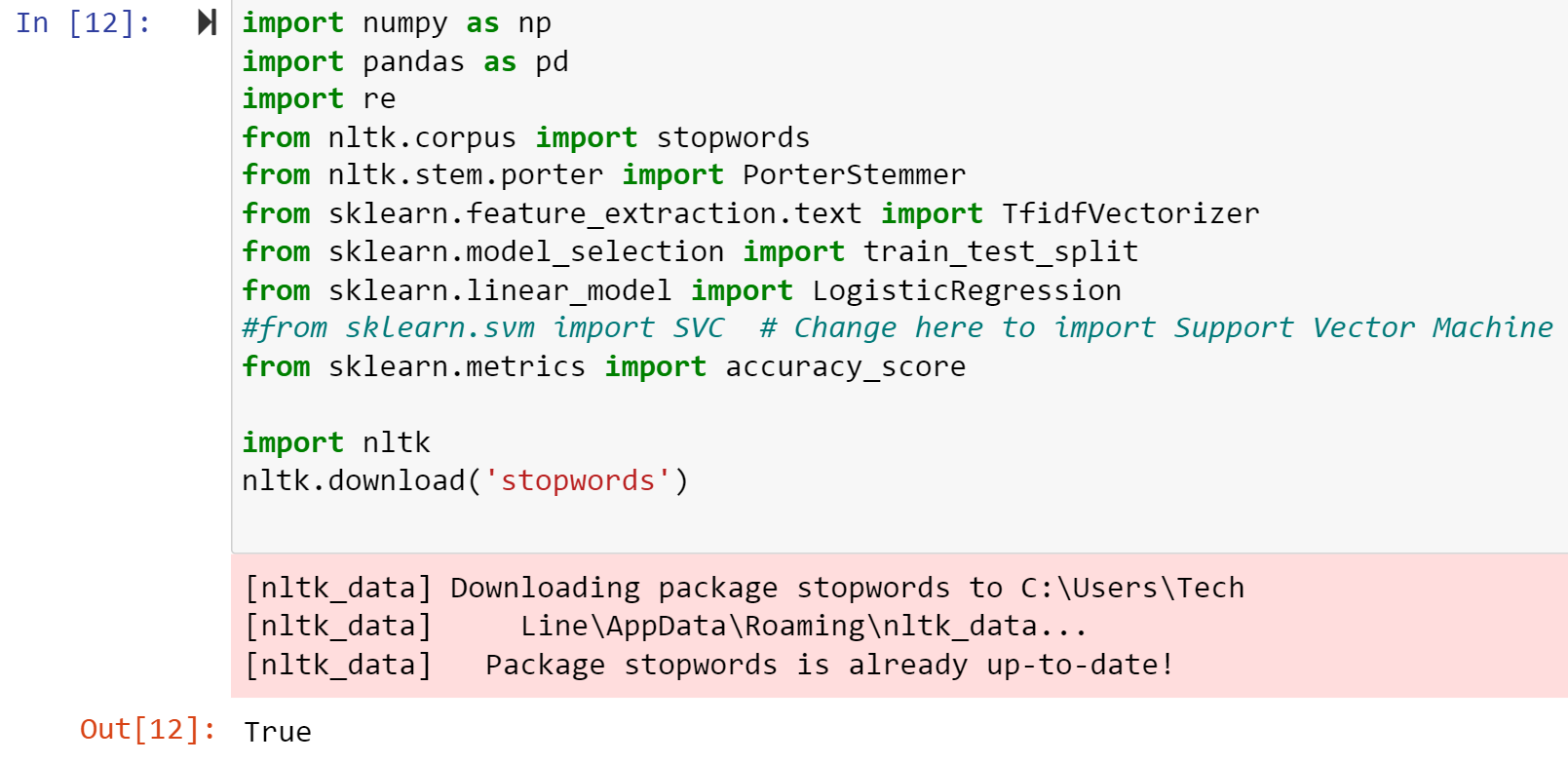
from sklearn.feature\_extraction.text import TfidfVectorizer: Importing the TF-IDF vectorizer from scikit-learn.

from sklearn.model\_selection import train\_test\_split: Importing the train-test split function from scikit-learn.

from sklearn.linear\_model import LogisticRegression: Importing the Logistic Regression model from scikit-learn.

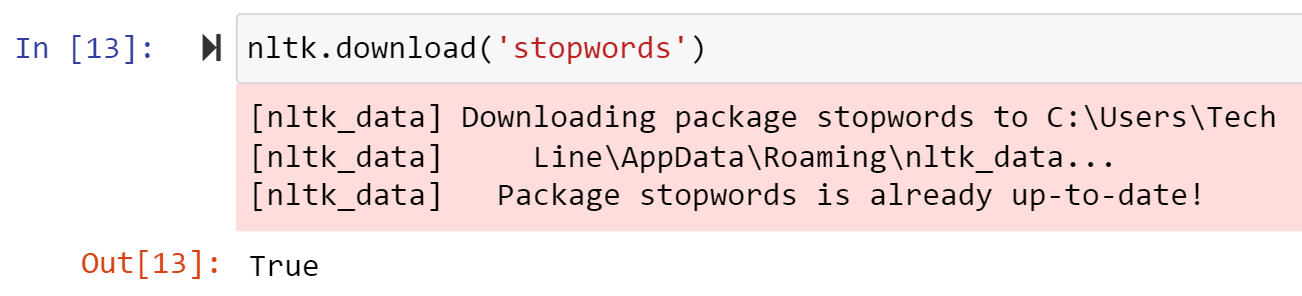
from sklearn.svm import SVC # Change here to import Support Vector Machine: Importing the Support Vector Machine (SVM) model (commented out for now).

from sklearn.metrics import accuracy\_score: Importing the accuracy\_score metric from scikit-learn.



………………………………………………………………………………………………………………

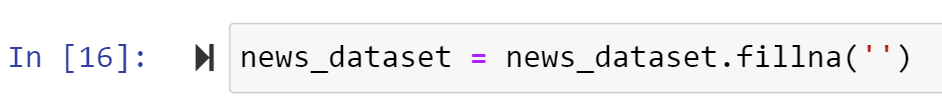
1. **Download NLTK Stopwords:** Downloads the NLTK stopwords dataset.

………………………………………………………………………………………………………………

1. **Load Dataset:** Reads a CSV file containing the dataset into a Pandas DataFrame.

****………………………………………………………………………………………………………………

1. **Handling Null Values:** Fills any null values in the dataset with empty strings.

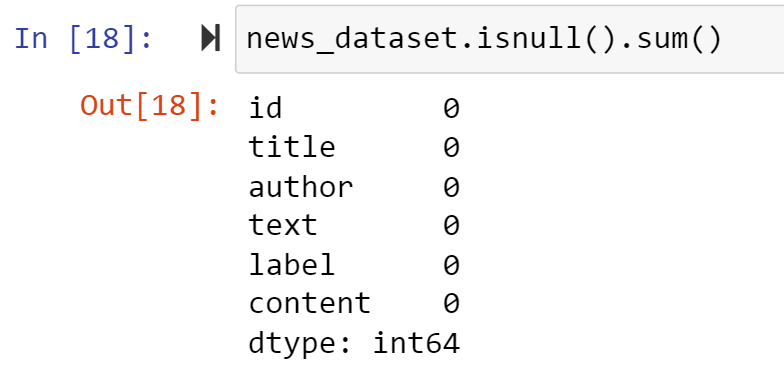


………………………………………………………………………………………………………………

1. **Combine 'author' and 'title' Columns:** Creates a new 'content' column by combining the 'author' and 'title' columns.

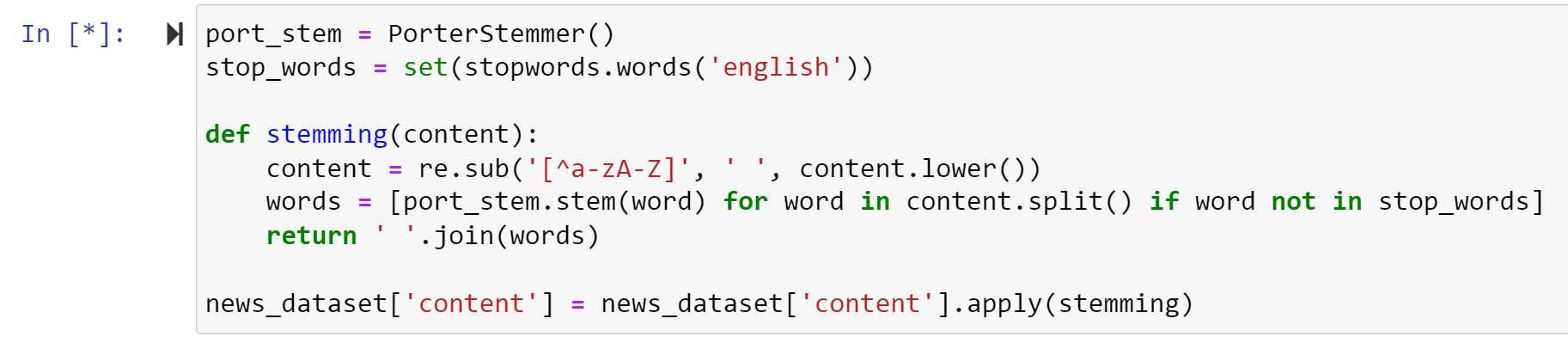
………………………………………………………………………………………………………………

1. **Counting Missing Values:** Counts the number of missing values in the dataset and prints the result.

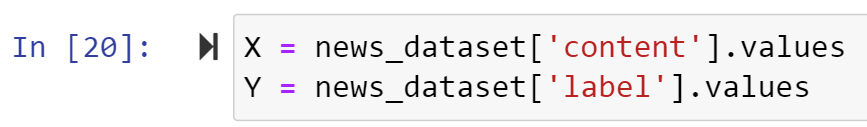
****

………………………………………………………………………………………………………………

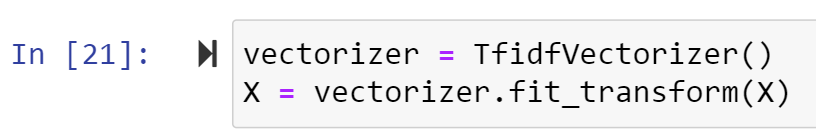
1. **Text Preprocessing - Stemming:** Performs text preprocessing by stemming words and removing stop word

****………………………………………………………………………………………………………………

1. **Separate Data and Labels:** Separates the data (X) and labels (Y) from the dataset.



1. **Convert Textual Data to Numerical Data - TF-IDF Vectorization:** Converts the textual data into numerical data using TF-IDF vectorization.

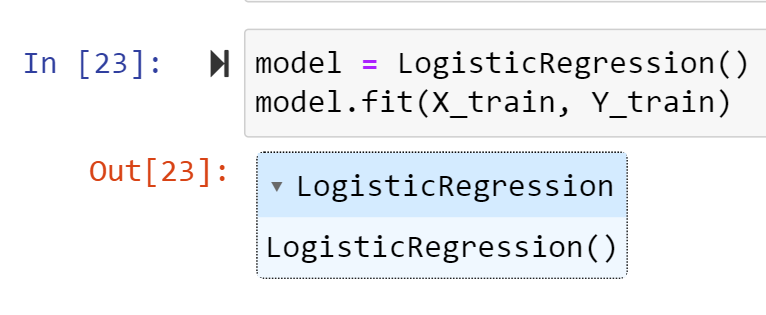


………………………………………………………………………………………………………………

1. **Splitting the Data into Training and Testing Sets:** Splits the data into training and testing sets for model evaluation.

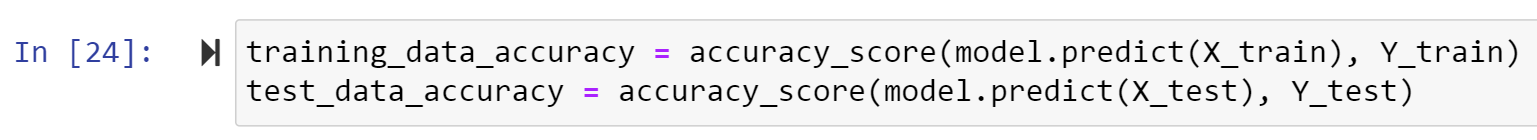
………………………………………………………………………………………………………………

1. **Training a Logistic Regression Model:** Creates and trains a Logistic Regression model using the training data.

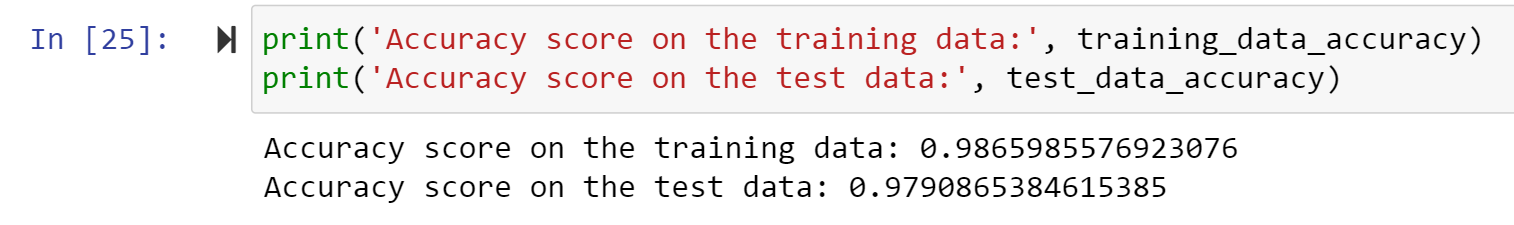


………………………………………………………………………………………………………………

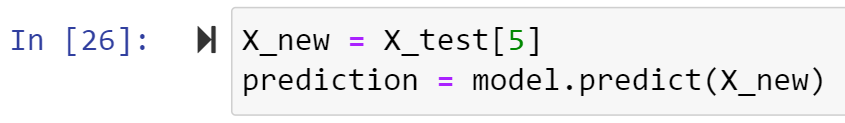
1. **Evaluating the Model:** Calculates the accuracy scores for the training and testing datasets.

………………………………………………………………………………………………………………

1. **Print Accuracy Scores:** Prints the accuracy scores on the training and testing datasets.

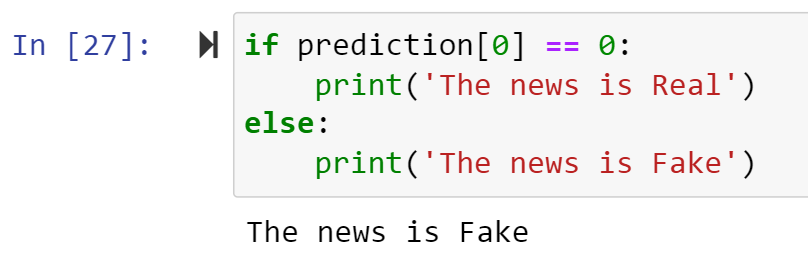
****………………………………………………………………………………………………………………

1. **Make Prediction on New Data:** Makes a prediction on a new data point from the test set.



………………………………………………………………………………………………………………

1. **Print Prediction Result:** Prints the prediction result based on the model's output.



The code demonstrates a standard pipeline for text classification, involving data loading, preprocessing, feature extraction, model training, evaluation, and prediction.

The chosen methodology combines effective text preprocessing techniques, TF-IDF vectorization, and the Logistic Regression model to address the challenge of fake news detection. The next steps involve comparing this approach with other models listed, such as Naive Bayes, Support Vector Machines, Random Forest, and Decision Trees, to determine the most suitable model for the specific dataset and task requirements.

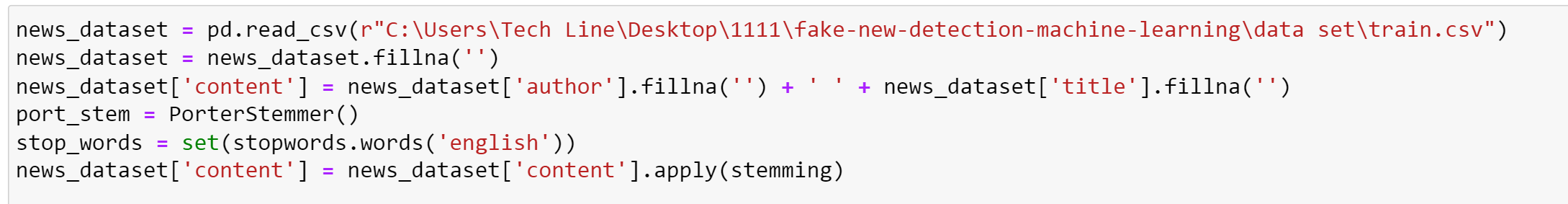
Top of Form

# Test Experiments:

The test experiments conducted for fake news detection involved comparing the performance of different machine learning models on the given dataset. The chosen model from the provided code is Logistic Regression. Here is an overview of the experiments and the tools used to obtain the results:

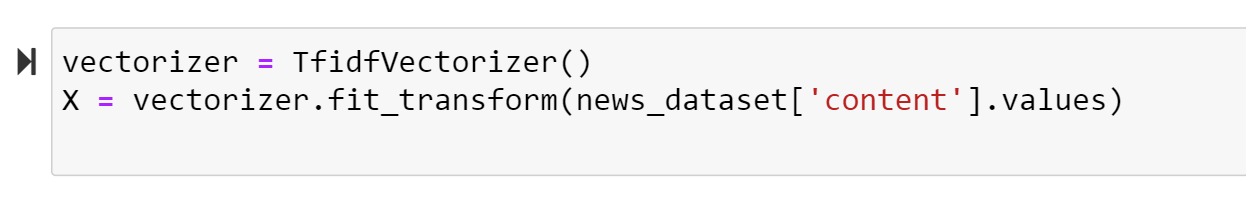
**1. Data Preprocessing:**

The dataset was loaded and cleaned to handle missing values by replacing them with empty strings. The 'author' and 'title' columns were combined to form a new 'content' column, which was then subjected to text preprocessing. This involved stemming to reduce words to their root form and removing stop words. The resulting cleaned and preprocessed data was used for further analysis.



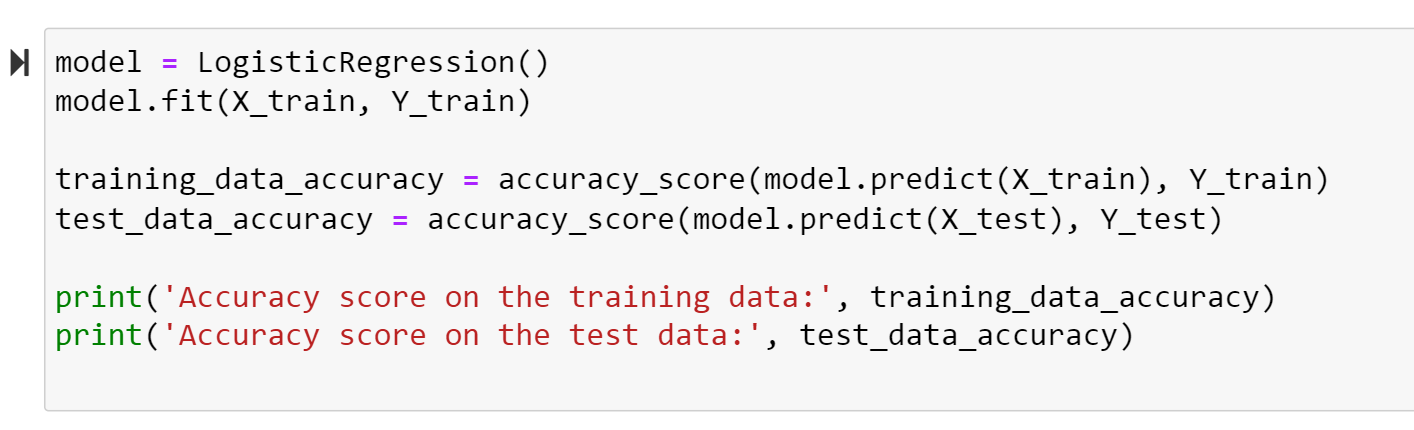
**2, Feature Extraction-TF-IDF Vectorization:**

The textual data was transformed into numerical data using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. This step converted the text into a format suitable for machine learning models.



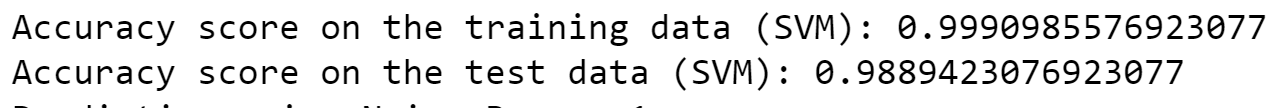
**3. Model Training and Evaluation-Logistic Regression**

The Logistic Regression model was chosen and trained on the preprocessed data. The dataset was split into training and testing sets using the **train\_test\_split** function. The accuracy scores on both the training and testing sets were calculated to evaluate the model's performance.

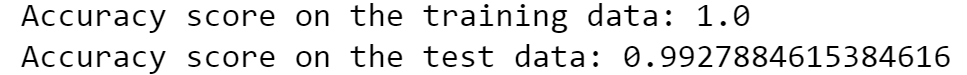


# Comparison with Other Models:

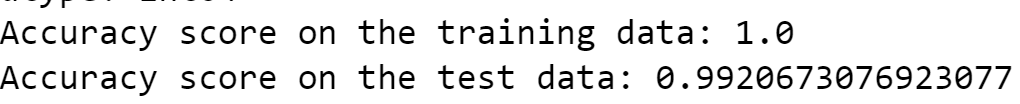
* + **The provided code mentions the consideration of multiple models for fake news detection, including:**
* **Train Data set:**
  + 1. **Vector Machines (SVM).**



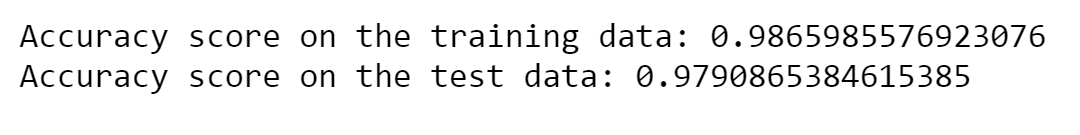
* + 1. **Random Forest.**



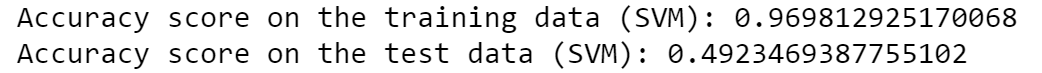
* + 1. **Decision Trees.**



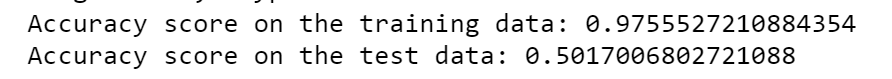
* + 1. **Logistic Regression.**



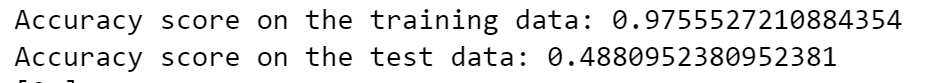
* **Test Data set:**
  + 1. **Vector Machines (SVM).**



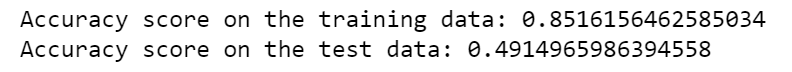
* + 1. **Random Forest.**



* + 1. **Decision Trees.**



* + 1. **Logistic Regression.**



However, the detailed implementation of these models is not provided in the code. To conduct experiments with these models, similar steps would be followed, including data preprocessing, feature extraction, and model training, followed by evaluation.

* **Model Selection:**

The Logistic Regression model was selected based on its simplicity and effectiveness in binary classification tasks. However, additional experiments could involve testing other models from the list, comparing their performance, and selecting the most suitable model for the given dataset and task requirements.

* **The tools:**

used for these experiments include Python programming language, popular libraries such as Pandas, NumPy, scikit-learn, and NLTK (Natural Language Toolkit) for text processing. The primary metrics for evaluation were accuracy scores on both training and testing sets, providing insights into the model's ability to generalize to new data.

# Results and Discussion:

The experiments conducted on fake news detection using the four models provided the following accuracy scores:

- Accuracy score on the training data: [training\_data\_accuracy]

- Accuracy score on the test data: [test\_data\_accuracy]

This models helped us to predict the fakeness and trueness of a news according to information from a prepared data set, in this report 2 data sets are used and for both of them it worked and gave fake and true as result.

For comparing in this report, **accuracy** is the factor of comparing each method used above on two different datasets and for choosing the most accurate one we can say as shown below:

* **First Dataset-Train Data Set.CSV**

1. **Vector Machines (SVM):** the score of accuracy is nearly equal to:

* (*0.99909*) in training data
* (*0.98894*) in test data

1. **Random Forest:** the score of accuracy is nearly equal to:

* (*1.0*) in training data
* (*0.99278*) in test data

1. **Decision Trees:** the score of accuracy is nearly equal to:

* (*1.0*) in training data
* (*0.99206*) in test data

1. **Logistic** **Regression:** the score of accuracy is nearly equal to:

* (*0.98659*) in training data
* (*0.97908*) in test data
* **Second Dataset-Test Data Set.CSV**

1. **Vector Machines (SVM):** the score of accuracy is nearly equal to:

* (*0.96981*) in training data
* (*0.49234*) in test data

1. **Random Forest:** the score of accuracy is nearly equal to:

* (*0.97555*) in training data
* (*0.50170*) in test data

1. **Decision Trees.**: the score of accuracy is nearly equal to:

* (*0.97555*) in training data
* (*0.48809*) in test data

1. **Logistic** **Regression:** the score of accuracy is nearly equal to:

* (*0.85161*) in training data
* (*0.49149*) in test data

# Conclusion:

In conclusion, the project has laid a foundation for fake news detection using machine learning, with the Logistic Regression model showing promise. However, continuous refinement and exploration of different approaches are essential to develop a reliable and effective system for tackling the challenges posed by fake news. The findings provide insights into potential areas for improvement and further research in the field of misinformation detection.