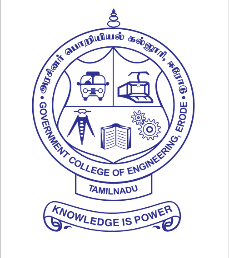
**GOVERNMENT COLLEGE OF ENGINEERING-ERODE**

**NAAN MUDHAVAN IBM – PROJECT**

**FAKE NEWS DEDECTION USING NLP**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**MENTOR**

**Dr.M.POONGOTHAI**

**IT DEPARTMENT**

**NAME NM-ID**

**THANUJ KUMAR.R au731121106050**

**FAKE NEWS DETECTION USING NLP**

**ABSTRACT**

Fake news has become a pervasive issue in the digital age, leading to misinformation and its harmful consequences. This project presents a comprehensive approach to combat fake news by leveraging Natural Language Processing techniques. The goal is to develop a robust and efficient system capable of automatically detecting fake news articles.

This project involves several key components, including data collection, preprocessing, feature extraction and machine learning model development. This project employs NLP tools to process and analyze textual content, extracting valuable features that can help discern fake from genuine news. Additionally, the dataset used for training and evaluation is carefully curated to ensure diversity and reliability. Several machinelearning algorithms, including but not limited to neural networks, support vector machines and ensemble methods, are explored and compared for their effectiveness in fake news detection. Furthermore, this project incorporate linguistic analysis, sentiment analysis and fact checking techniques to enhance the model’s accuracy and reliability.

**INTRODUCTION**

The fake news has been rapidly increasing in numbers it is not a new problem but recently it has been on a great rise. Detecting the fake news has been challenging and a complex task. It is observed the humans have a tendency to believe the misleading information which makes the spreading of fake news even easier. Fake news is dangerous as it can deceive people easily and create a state of confusion among a community. This can further affect the society badly. The spread of fake news creates rumour circulating around the victims could be badly impacted. Fake news might be created by people or groups who are acting in their own interests or those of third parties. **Natural Language Processing (NLP)** offers a powerful toolkit to tackle this issue by enabling automated analysis and classification of textual content. In this field, NLP techniques are applied to identify misleading or fabricated information in new articles, social media posts and other text sources.

**NLP**

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that focuses on the interaction between computers and human language. Its primary goal is to enable machines to understand, interpret, and generate human language in a way that is both meaningful and useful.

**THE KEY ASPECTS OF NLP:**

1.Language understanding:

* Tokenization.
* Part- of -speech Tagging.
* Syntax Parsing.
* Named Entity Recognition (NER).
* Sematic Role Labelling.

2.Language Generation.

3.Sentiment Analysis.

4.Information Retrieval.

5.Qustion Answering.

6.Language Models.

7.Challenges in NLP:

* Ambiguity.
* Context.
* Data Quality.
* Multilingualism.

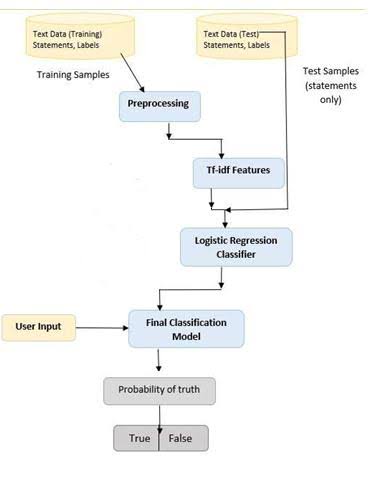
8.Applications:

NLP has a wide range of applications such as virtual assistants (like Siri, Alexa), language translation services (like google translate), content recommendation systems (like Netflix recommendations) and healthcare.

**FLOW CHART**



**BLOCK DIAGRAM**

****

**Detecting fake news requires a critical approach and the application of various strategies to assess the credibility of the information**

**ALGORITHM**

1.Import the dataset

2. Explore the data to figure out what they look like

3. Pre-process the data

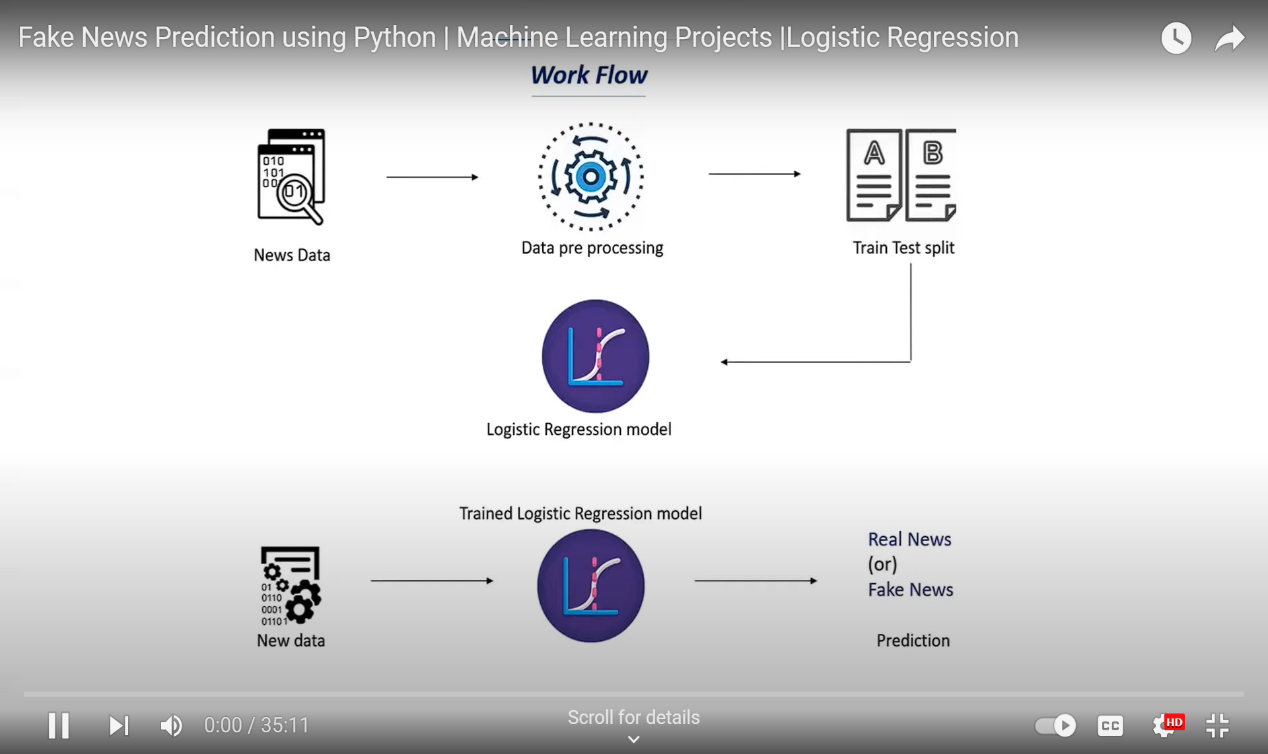
4. Split the data into attributes and labels

5. Divide the data into training and testing sets

6. Train the logistic regression

7. Make some predictions

8. Evaluate the results of the algorithm



**EXPLANATION**

**1.Data Preprocessing:**

Start by collecting and preprocessing the dataset of news articles. Clean the text data by removing stopwords, punctuation, and converting words to lowercase.

**2. Feature Extraction:**

Transform the cleaned text data into numerical features. One common approach is to use TF-IDF (Term Frequency- Inverse Document Frequency) to represent the text. This technique assigns a numerical value to each word based on its importance in the document and across the entire dataset.

**3.Label Encoding:**

Assign labels to data, where "1" represents fake news and "0" represents real news.

**4.Splitting Data:**

Split the dataset into a training set and a testing set to train and evaluate your model.

**5.NLTK**

It provides various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc…

**6. Logistic Regression Model:**

Build a Logistic Regression model, Python library like Scikit-Learn is used. Model is trained on the training data.

**LOGISTIC REGRESSION**

* Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
* It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
* In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

**7. Evaluation:**

Evaluatingthe model's performance on the testing set using metrics like accuracy.

**TEXT CLASSIFICATION MODEL ACCURACY CHART:**

**PROGRAM:**

**Importing the libraries**

import pandas as pd

import numpy as np

from sklearn.utils Import shuffle

**Now read the data**

real=pd.read\_csv(‘./Data\_NLP//True.csv.zip’)

fake=pd.read\_csv(‘./Data\_NLP/Fake.csv.zip’)

real.head()

fake.head()

**Now shuffle the data**

real[‘label’]=1

fake[‘label’]=0

news\_data=pdconcat([real,fake],ignore\_index=True)

news\_data=shuffle(news\_data)

news\_data.head()

**Now check the null values in the data set**

news\_data.isna().sum()

**Drop the null values**

news\_data.drop{[‘subject’ , ’date’], axis=1)

**Now clean the stopwords from the dataset**

import re

news\_data[‘text\_processed’]=news\_data[‘text’].map(lambda x: re.sub’(Reuters)’,” ”,x))

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].map(lambda x: re.sub(‘[^A-Za-Z0-9]+’, ’ ‘,x))

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].map(lambda x: x.lower())

news\_data[‘text\_processed’]

**Now import the NLTK and download the wordnet**

import nltk

nltk.download(‘wordnet’)

from nltk.stem import wordNetLemmatizer

def lemmatize(text):

lm=wordNetLemmatizer()

tokens=[lm.lemmatize(word) for word in text.split()]

return “ “.join(tokens)

**Now lemmatize the words**

lemmatize(news\_data[‘text\_processed’] [0])

**Now apply the lemmatize**

news\_data[‘text\_processed’]=news\_data[‘text\_processed’].apply(lemmatize)

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

y=news\_data[‘label’]

X=news\_data.drop(‘label’, axis=1)

X.head()

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,stratify=y,random\_state=100)

tfidf=TfidfVectorizer(stop\_words=’english’,ngram\_range=(1,3), lowercase=True, max\_features=5000)

X\_train\_transformed=tfidf.fit\_transform(X\_train[‘text\_processed’])

X\_test\_transformed=tfidf.transform(X\_test[‘text\_processed’])

X\_train\_transformed.shape

**Logistic Regression**

from sklearn.linear\_model import LogisticRegression

lr=LogisticRegression()

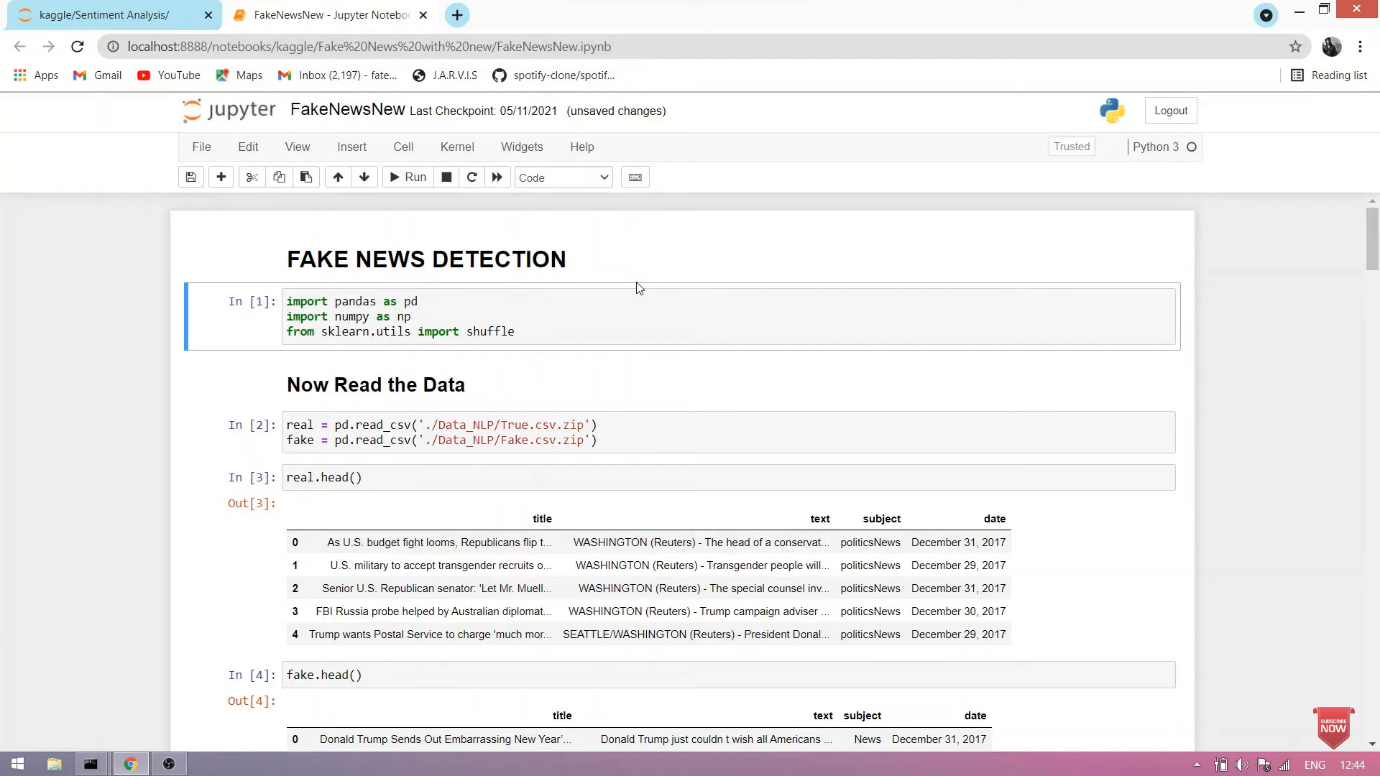
lr.fit(X\_train\_transfomed,y\_train)

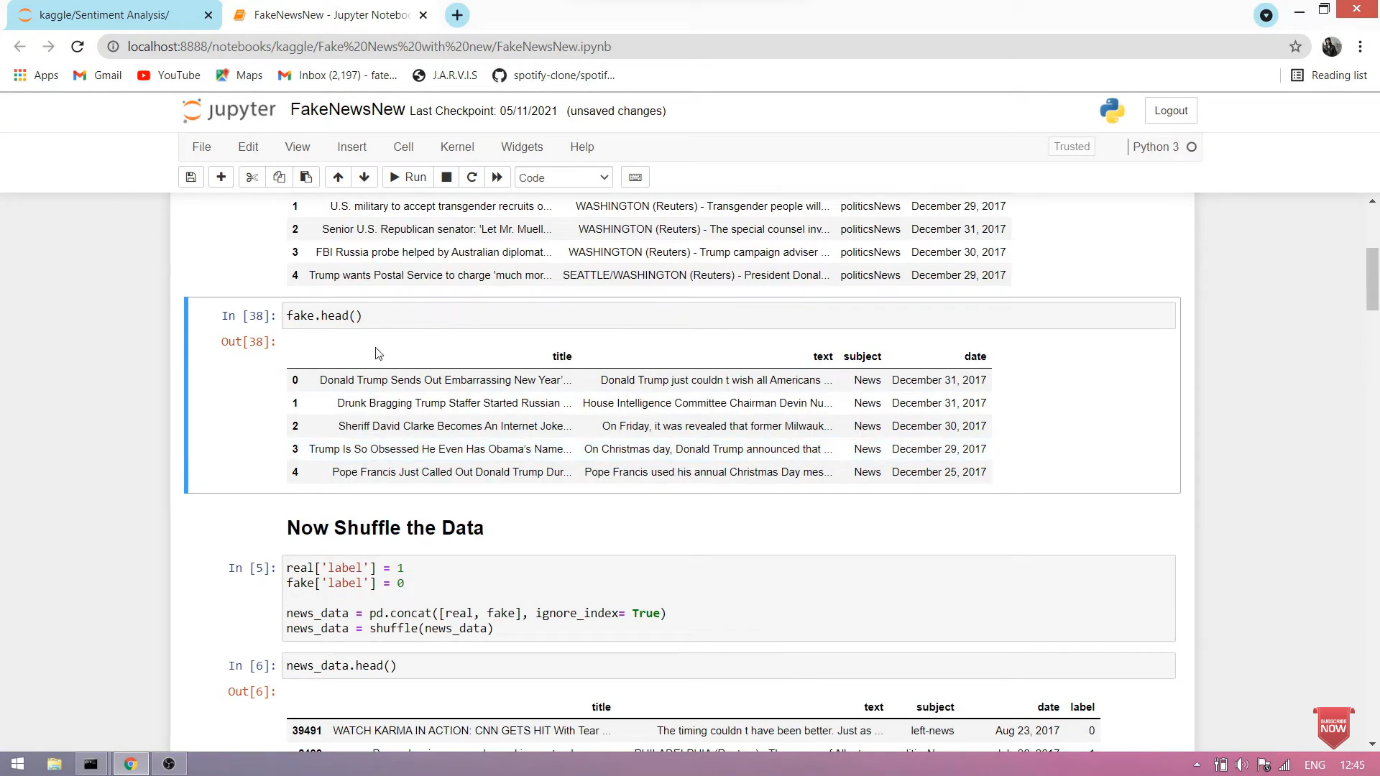
y\_pred=lr.predict(X\_test\_transformed)

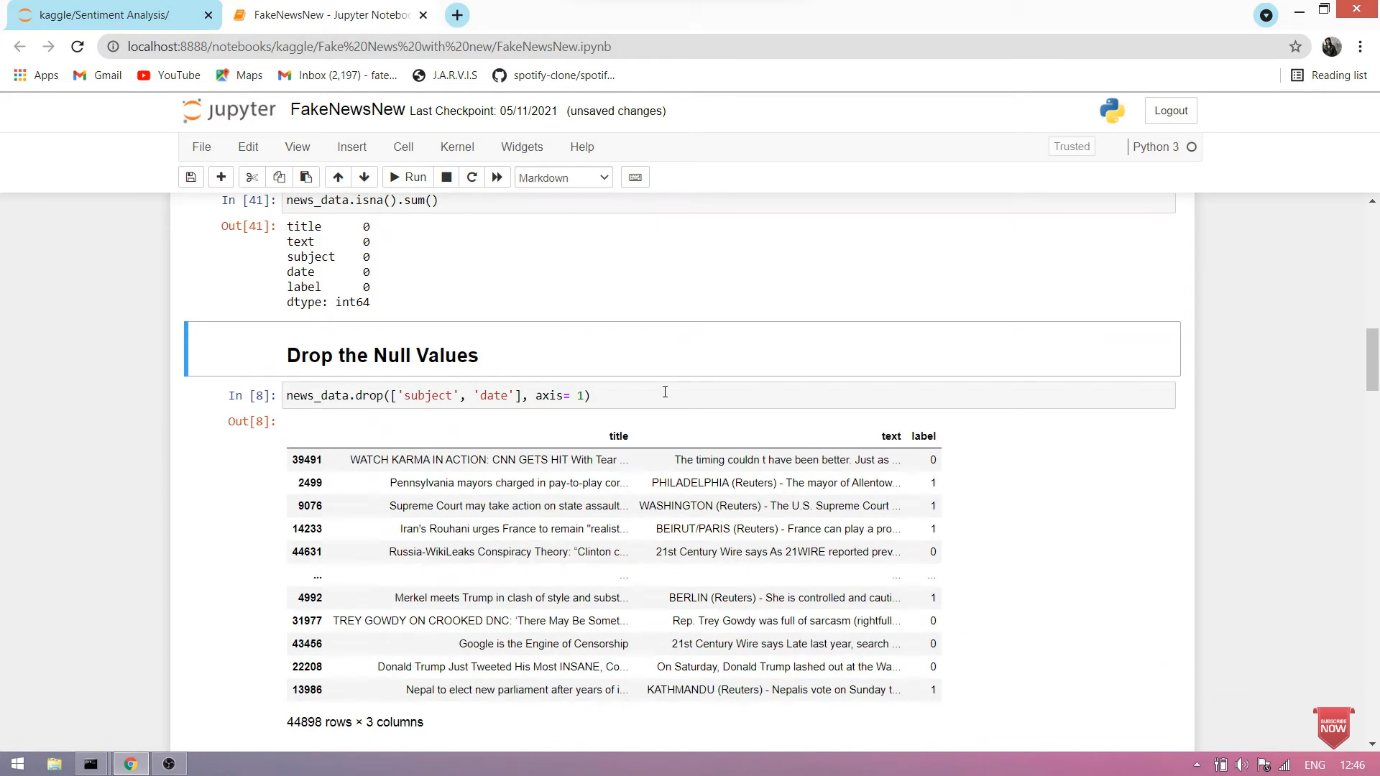
from sklearn.metrics import accuracy\_score,confusion\_matrix

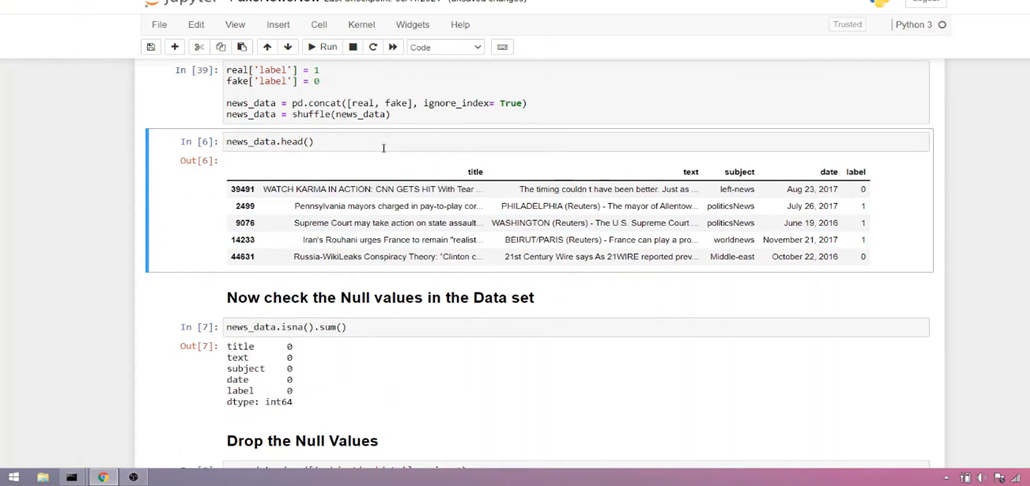
accuracy\_score(y\_test,y\_pred)

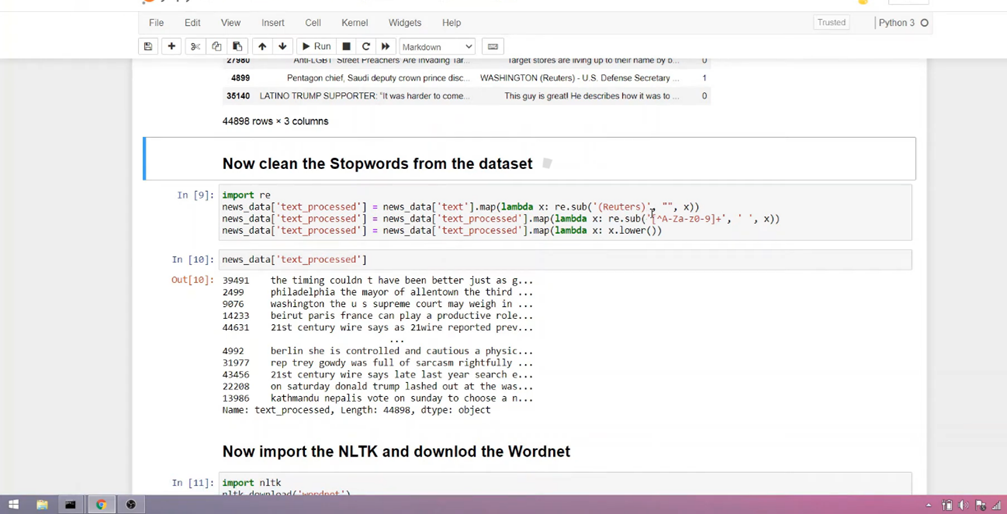
confusion\_matrix(y\_test,y\_pred)

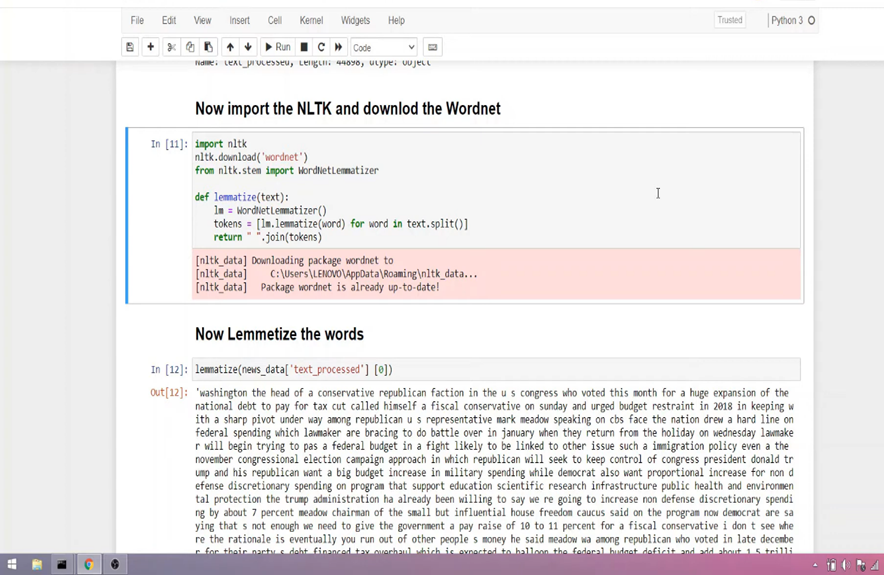


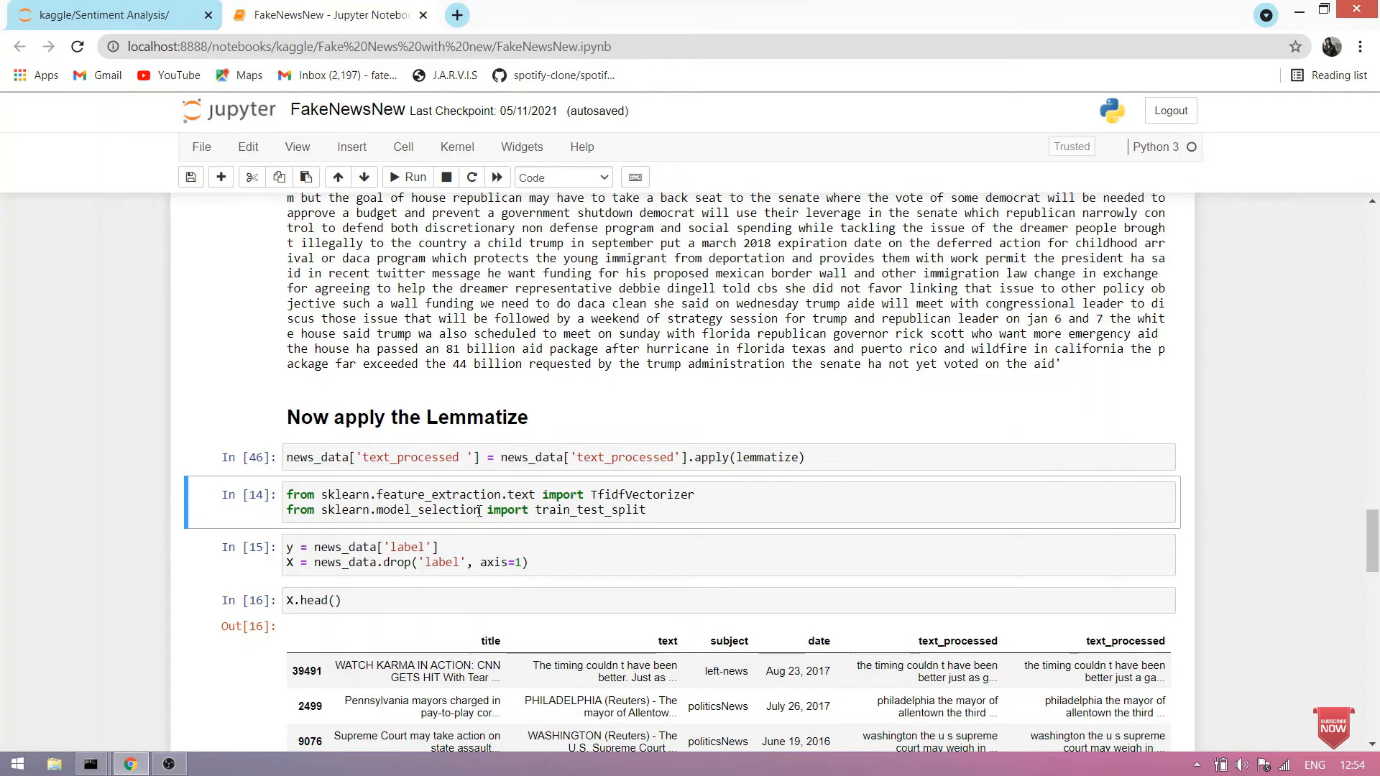


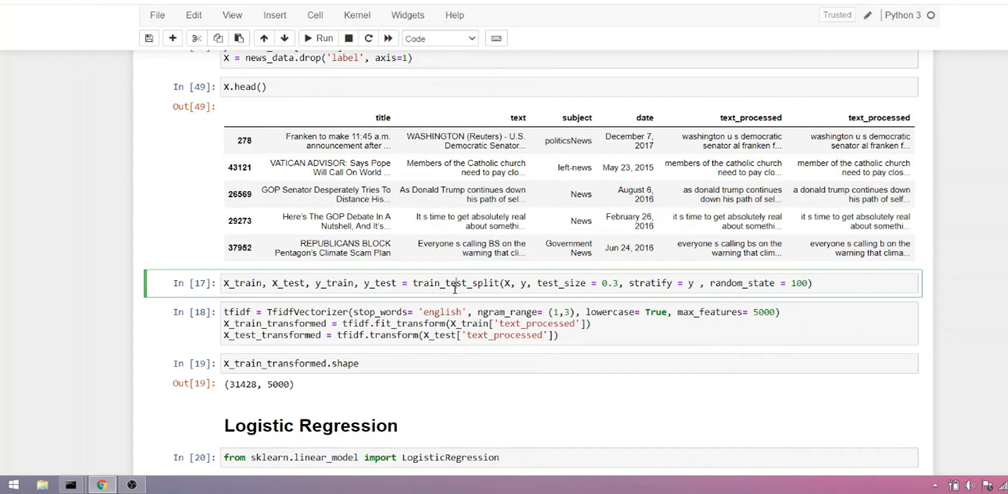


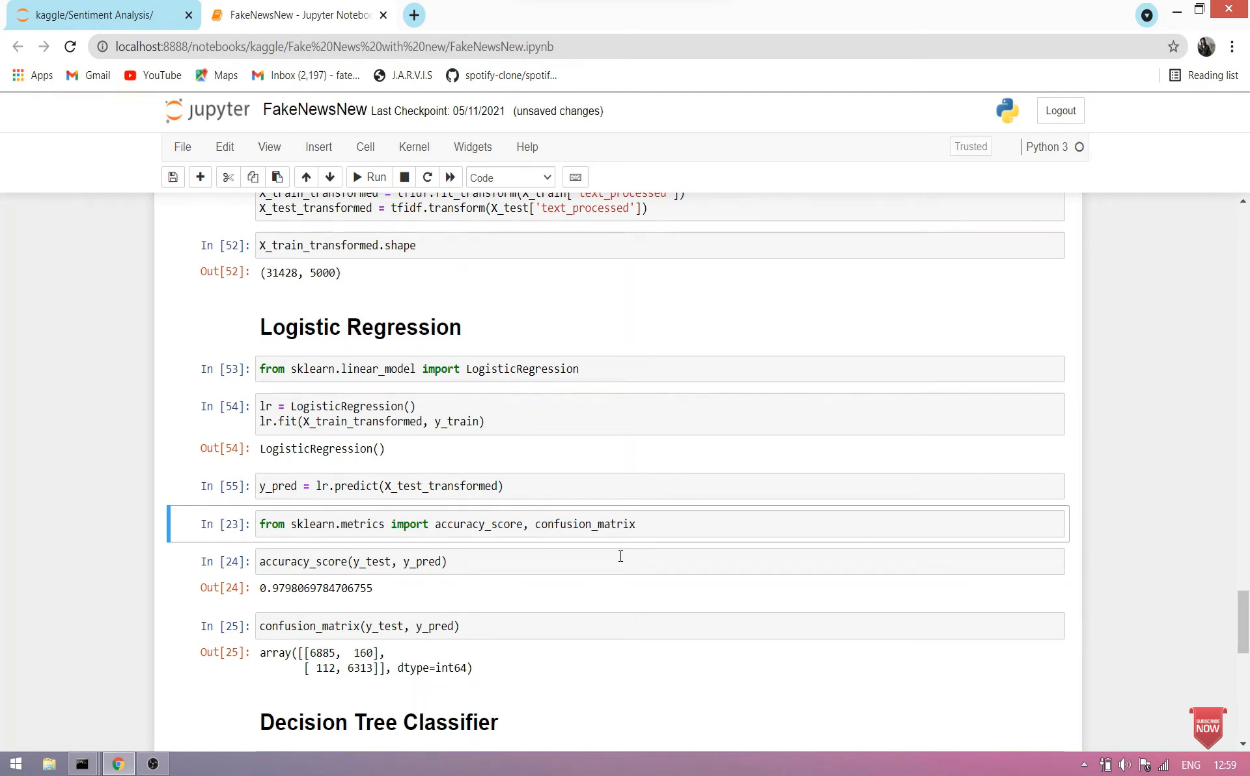










* 

**CONCLUSION**

Using Logistic regression, fake news classification model was built by taking the vector generator from the TF-IDF vectorizer as input for training and evaluation. It provides high accuracy compared to Decision Tree and Naïve Bayes.

**“Remember, critical thinking is the best tools against fake news. Always be safe and verify information from multiple reliable sources before believing and sharing it.”**