Project Report on

FAKE NEWS DETECTION

FLIP ROBO

NAME OF THE PROJECT

FAKE NEWS DETECTION

Submitted by:

Mr. Vikas Kumar Mishra

FLIPROBO SME:

Ms Khushboo Garg

Project Report on

FAKE NEWS DETECTION

ACKNOWLEDGMENT

I would like to express my special gratitude to the "Flip Robo" team, who has allowed me to deal with a beautiful dataset and helped me improve my analysis skills. And I want to express my huge gratitude to Ms Khushboo Garg (SME Flip Robo).

Thanks to "Data trained" who are the reason behind my Internship at FlipRobo Technologies.

SOURCE USED IN THIS PROJECT:

1. Learn Library Documentation
2. Help from YouTube Channels, Blogs from Educational Websites
3. Notes on Machine Learning (YouTube Channel)
4. SCIKIT Learn Library Documentation
5. Help from Kaggle websites, analytical vidya, GeeksforGeeks, etc.

Project Report on

FAKE NEWS DETECTION

INTRODUCTION

What is a Fake News?

Fake news simply means incorporating information that leads people to the wrong path. Nowadays fake news spreads like water and people share this information without verifying it. This is often done to further or impose certain ideas and is often achieved with political agendas.

For media outlets, the ability to attract viewers to their websites is necessary to generate online advertising revenue. So it is necessary to detect fake news.

Context

Fake news has become one of the biggest problems of our age. It has a serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

Project Report on

FAKE NEWS DETECTION

Workflow

In this project, we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression), Scikit Learn

Natural Language Processing

Machine learning data only works with numerical features so we have to convert text data into numerical columns. So, we have to pre-process the text, called natural language processing

In-text pre-processing, we clean our text by steaming, lemmatization, removing stop words, removing special symbols and numbers, etc. After cleaning the data, we have to feed this text data into a vectorizer which will convert this text data into numerical features.

Dataset

Project Report on

FAKE NEWS DETECTION

I can find many datasets for fake news detection on Kaggle or many other sites. I download these datasets from Kaggle. There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. We combined both datasets using pandas' built-in function.

Analytical Problem Framing

Mathematical / Analytical Modelling of the Problem

Our objective is to detect Fake News which can be resolved by the use of the classification-based algorithm. In this project, we are going to use different types of algorithms which use their mathematical equation in the background. This project comes with two separate data set for Fake.csv & True.csv file. Initially, data cleaning & pre-processing perform over data. Feature engineering is performed to remove unnecessary features & for dimensionality reduction.
we are using some machine learning and Natural language processing libraries like NLTK, re (Regular Expression), Scikit Learn
In model building, the Final model is selected based on evaluation benchmarks among different models with different algorithms.

Project Report on

FAKE NEWS DETECTION

Data Sources and their formats

The data set provided by Flip Robo was in the format of CSV (Comma Separated Values). There are 2 data sets that are given. One is the ‘**Fake.csv**’ data and ‘**True.csv**’ data.

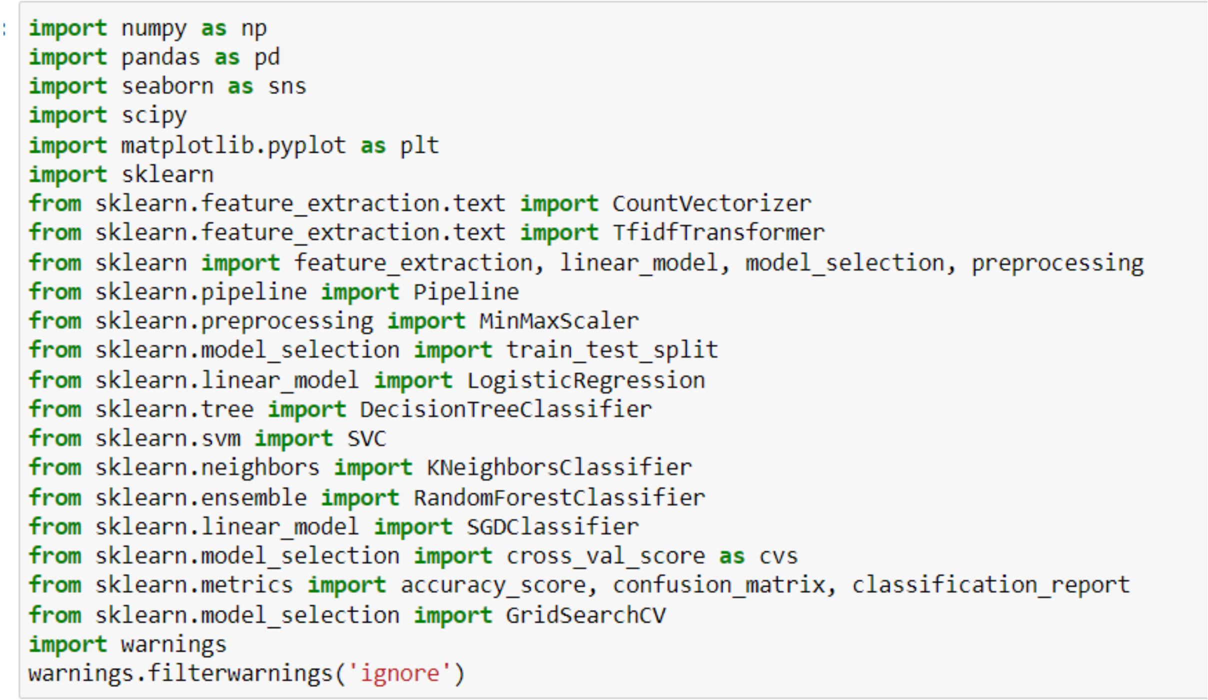
- There are two datasets one for fake news and one for true news.
- In true news, there is 21417 news, and in fake news, there is 23481 news.
- We add 1 label column ‘fake’ for fake news and ‘true’ for real news.
- We combined both datasets using pandas’ built-in function.

First Import Libraries

Project Report on

FAKE NEWS DETECTION

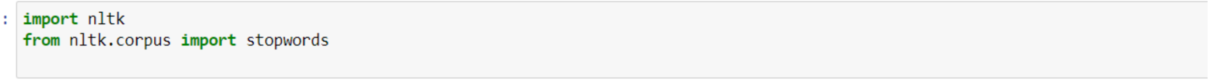
Importing All the necessary libraries.



```
import numpy as np
import pandas as pd
import seaborn as sns
import scipy
import matplotlib.pyplot as plt
import sklearn
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import feature_extraction, linear_model, model_selection, preprocessing
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import cross_val_score as cvs
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV
import warnings
warnings.filterwarnings('ignore')
```

Importing NLTK Libraries

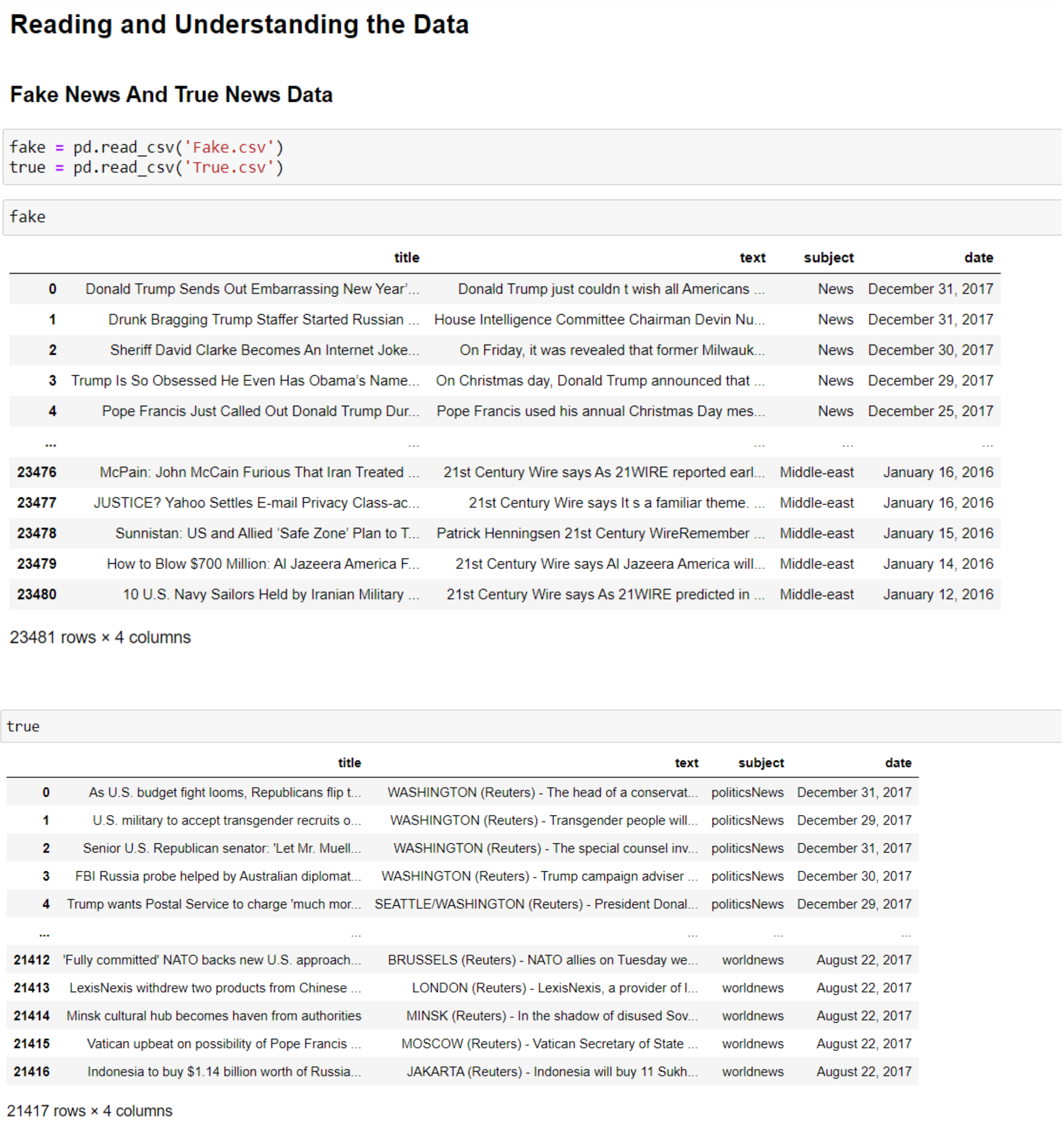
The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.



```
import nltk
from nltk.corpus import stopwords
```

Project Report on

FAKE NEWS DETECTION



Reading and Understanding the Data

Fake News And True News Data

```
fake = pd.read_csv('Fake.csv')
true = pd.read_csv('True.csv')
```

```
fake
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017
...	...	...	...	...
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016

23481 rows × 4 columns

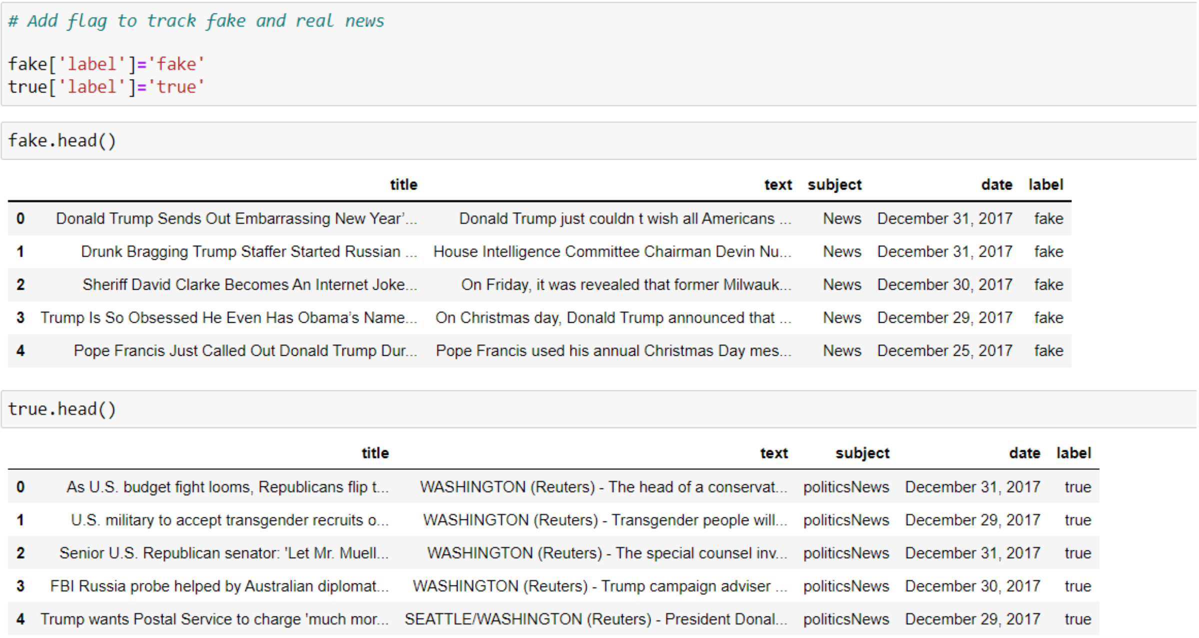
```
true
```

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017
...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017

21417 rows × 4 columns

- The Fake News dataset contains 4 columns with 23481 rows.
- The True News dataset contains 4 columns with 21417 rows.

Data Cleaning and Preparation



```
# Add flag to track fake and real news

fake['label']='fake'
true['label']='true'
```

```
fake.head()
```

	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	fake
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	fake
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	fake
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	fake
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	fake

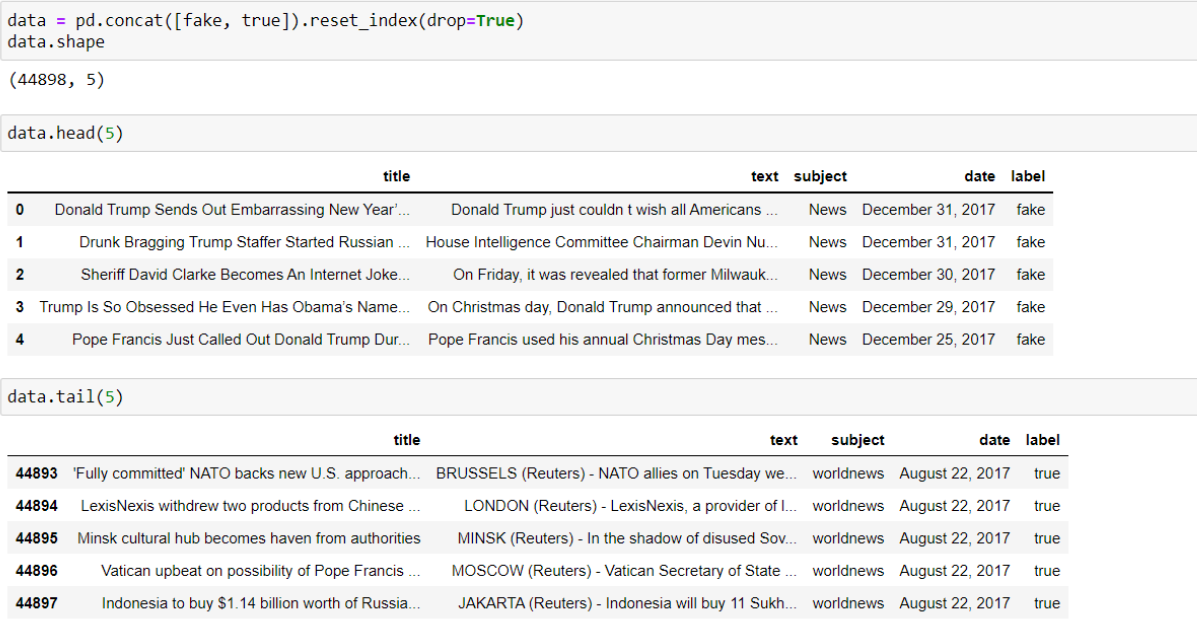
```
true.head()
```

	title	text	subject	date	label
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	true
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	true
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	true
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	true
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	true

- Add a label column 'fake' and 'true' for both data frames.
- 'fake' for fake news and 'true' for true news.

Concatenate Data Frames

With the help of the concat method, we merge the two CSV file data into a single file data.



```
data = pd.concat([fake, true]).reset_index(drop=True)
data.shape
```

```
(44898, 5)
```

```
data.head(5)
```

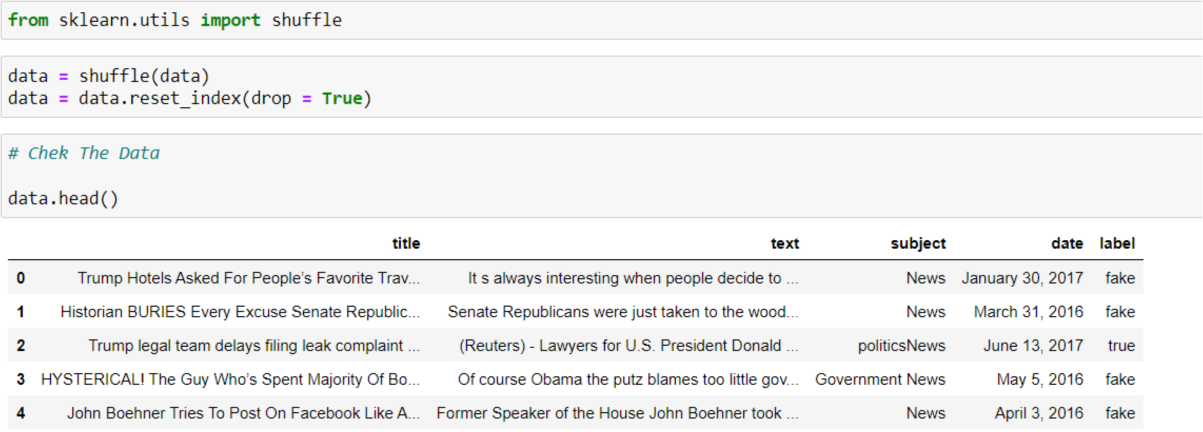
	title	text	subject	date	label
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	fake
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	fake
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	fake
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	fake
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	fake

```
data.tail(5)
```

	title	text	subject	date	label
44893	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	true
44894	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	true
44895	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	true
44896	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	true
44897	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	true

Shuffle The Data

shuffling techniques aim to mix up data and can optionally retain logical relationships between columns. It randomly shuffles data from a dataset within an attribute or a set of attributes.



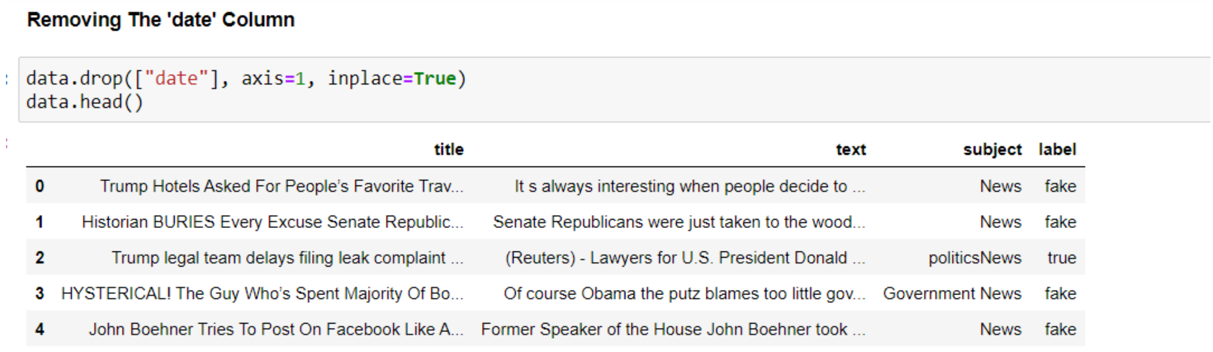
```
from sklearn.utils import shuffle
```

```
data = shuffle(data)
data = data.reset_index(drop = True)
```

```
# Chek The Data

data.head()
```

	title	text	subject	date	label
0	Trump Hotels Asked For People’s Favorite Trav...	It s always interesting when people decide to ...	News	January 30, 2017	fake
1	Historian BURIES Every Excuse Senate Republic...	Senate Republicans were just taken to the wood...	News	March 31, 2016	fake
2	Trump legal team delays filing leak complaint ...	(Reuters) - Lawyers for U.S. President Donald ...	politicsNews	June 13, 2017	true
3	HYSTERICAL! The Guy Who’s Spent Majority Of Bo...	Of course Obama the putz blames too little gov...	Government News	May 5, 2016	fake
4	John Boehner Tries To Post On Facebook Like A...	Former Speaker of the House John Boehner took ...	News	April 3, 2016	fake

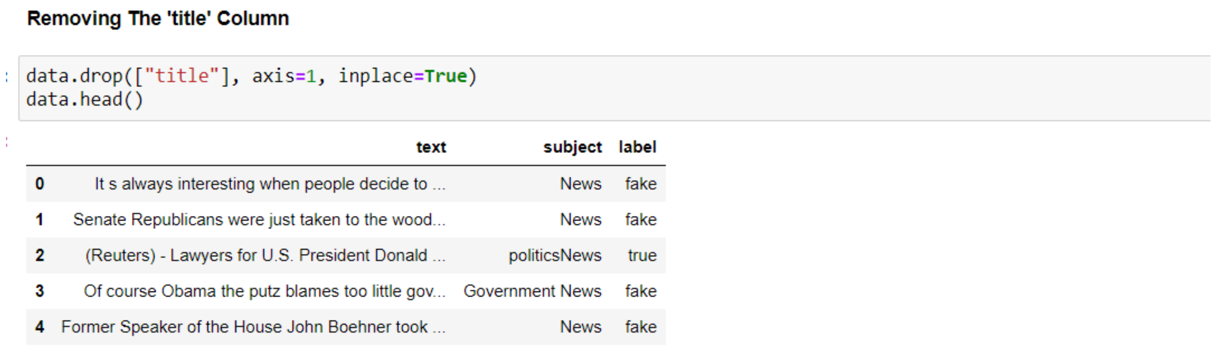
Removing Unusual Columns



Removing The 'date' Column

```
data.drop(["date"], axis=1, inplace=True)
data.head()
```

	title	text	subject	label
0	Trump Hotels Asked For People's Favorite Trav...	It s always interesting when people decide to ...	News	fake
1	Historian BURIES Every Excuse Senate Republic...	Senate Republicans were just taken to the wood...	News	fake
2	Trump legal team delays filing leak complaint ...	(Reuters) - Lawyers for U.S. President Donald ...	politicsNews	true
3	HYSTERICAL! The Guy Who's Spent Majority Of Bo...	Of course Obama the putz blames too little gov...	Government News	fake
4	John Boehner Tries To Post On Facebook Like A...	Former Speaker of the House John Boehner took ...	News	fake



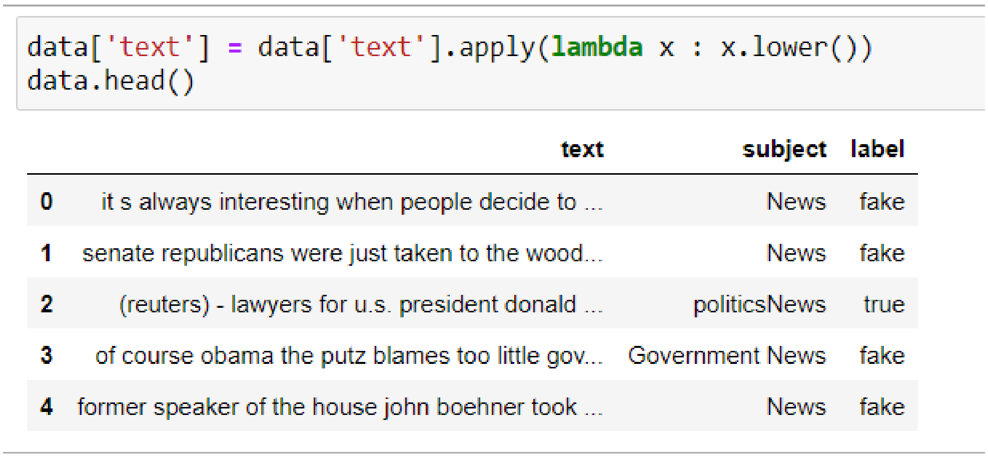
Removing The 'title' Column

```
data.drop(["title"], axis=1, inplace=True)
data.head()
```

	text	subject	label
0	It s always interesting when people decide to ...	News	fake
1	Senate Republicans were just taken to the wood...	News	fake
2	(Reuters) - Lawyers for U.S. President Donald ...	politicsNews	true
3	Of course Obama the putz blames too little gov...	Government News	fake
4	Former Speaker of the House John Boehner took ...	News	fake

Convert Data into Lower Case

Lowercase letters are used for common nouns and for every letter after the initial letter of the first word of a sentence, so we convert text into lowercase.

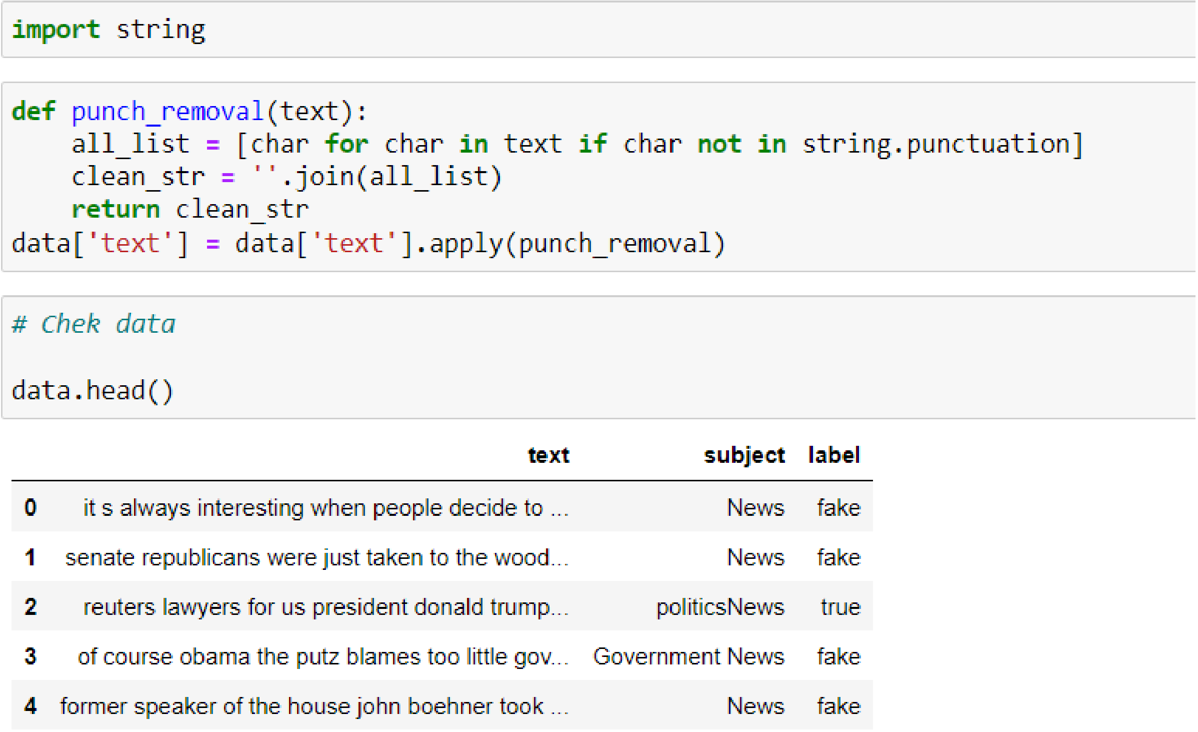


```
data['text'] = data['text'].apply(lambda x : x.lower())
data.head()
```

	text	subject	label
0	it s always interesting when people decide to ...	News	fake
1	senate republicans were just taken to the wood...	News	fake
2	(reuters) - lawyers for u.s. president donald ...	politicsNews	true
3	of course obama the putz blames too little gov...	Government News	fake
4	former speaker of the house john boehner took ...	News	fake

Removing Punctuations

Punctuations are special symbols that add grammatical structure to natural English. Natural English strings are not easily processed; hence we need to remove punctuation from strings before we can use them for further processing.



```
import string
```

```
def punch_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
    clean_str = ''.join(all_list)
    return clean_str
data['text'] = data['text'].apply(punch_removal)
```

```
# Chek data

data.head()
```

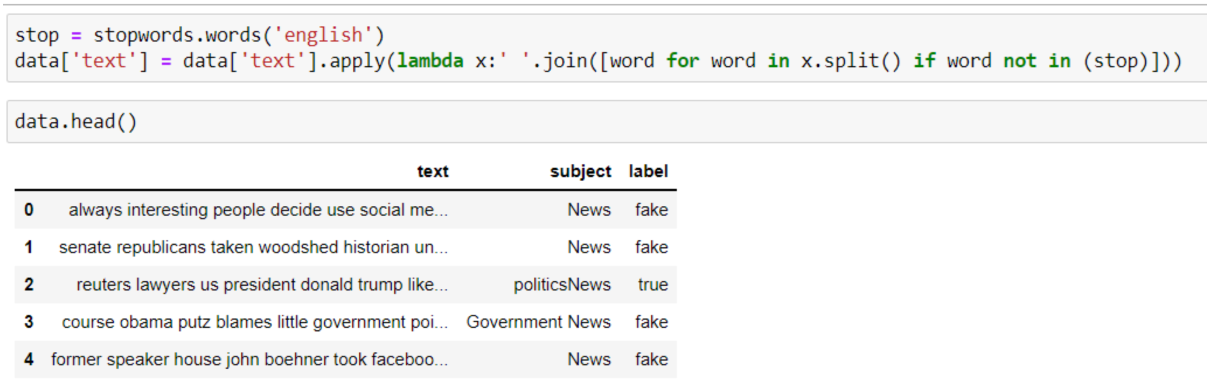
	text	subject	label
0	it s always interesting when people decide to ...	News	fake
1	senate republicans were just taken to the wood...	News	fake
2	reuters lawyers for us president donald trump...	politicsNews	true
3	of course obama the putz blames too little gov...	Government News	fake
4	former speaker of the house john boehner took ...	News	fake

Removing Stop Words

Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text in order to give more focus to the important information.

Project Report on

FAKE NEWS DETECTION



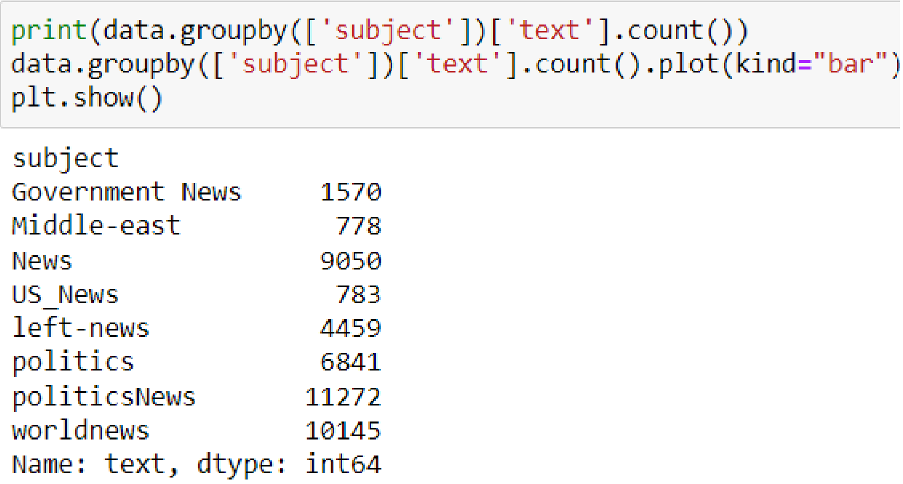
```
stop = stopwords.words('english')
data['text'] = data['text'].apply(lambda x:' '.join([word for word in x.split() if word not in (stop)]))
```

```
data.head()
```

	text	subject	label
0	always interesting people decide use social me...	News	fake
1	senate republicans taken woodshed historian un...	News	fake
2	reuters lawyers us president donald trump like...	politicsNews	true
3	course obama putz blames little government poi...	Government News	fake
4	former speaker house john boehner took faceboo...	News	fake

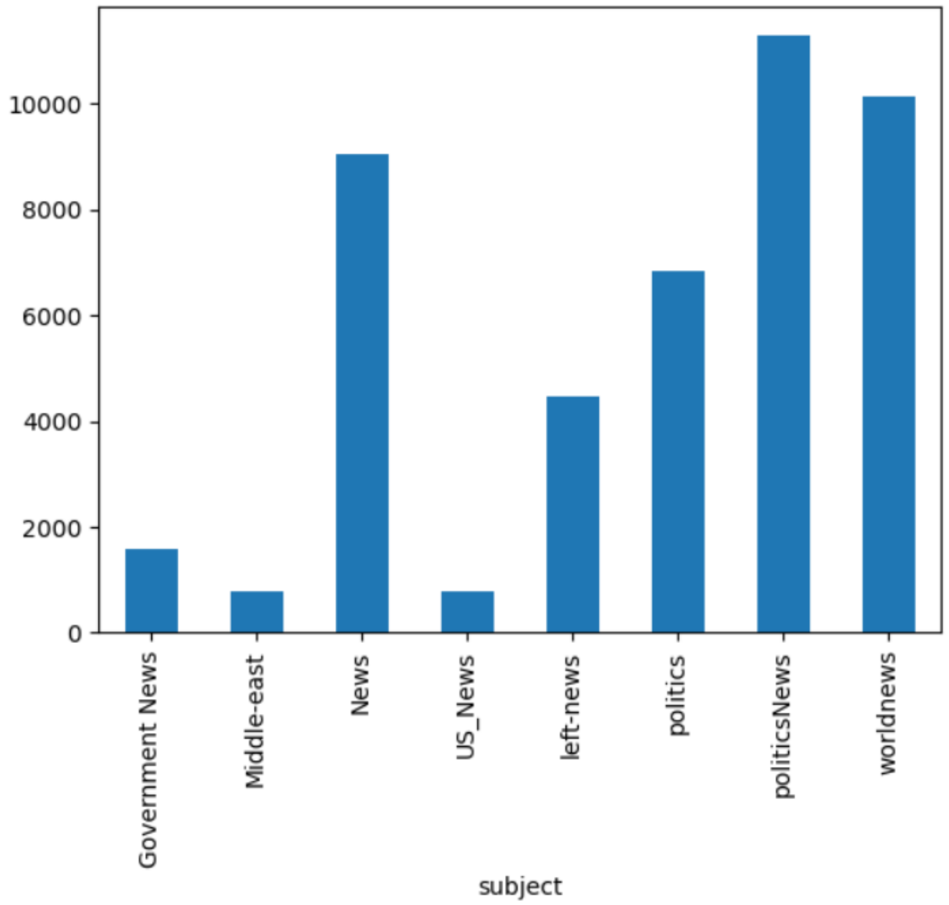
Data Visualization

News Articles Per Subject



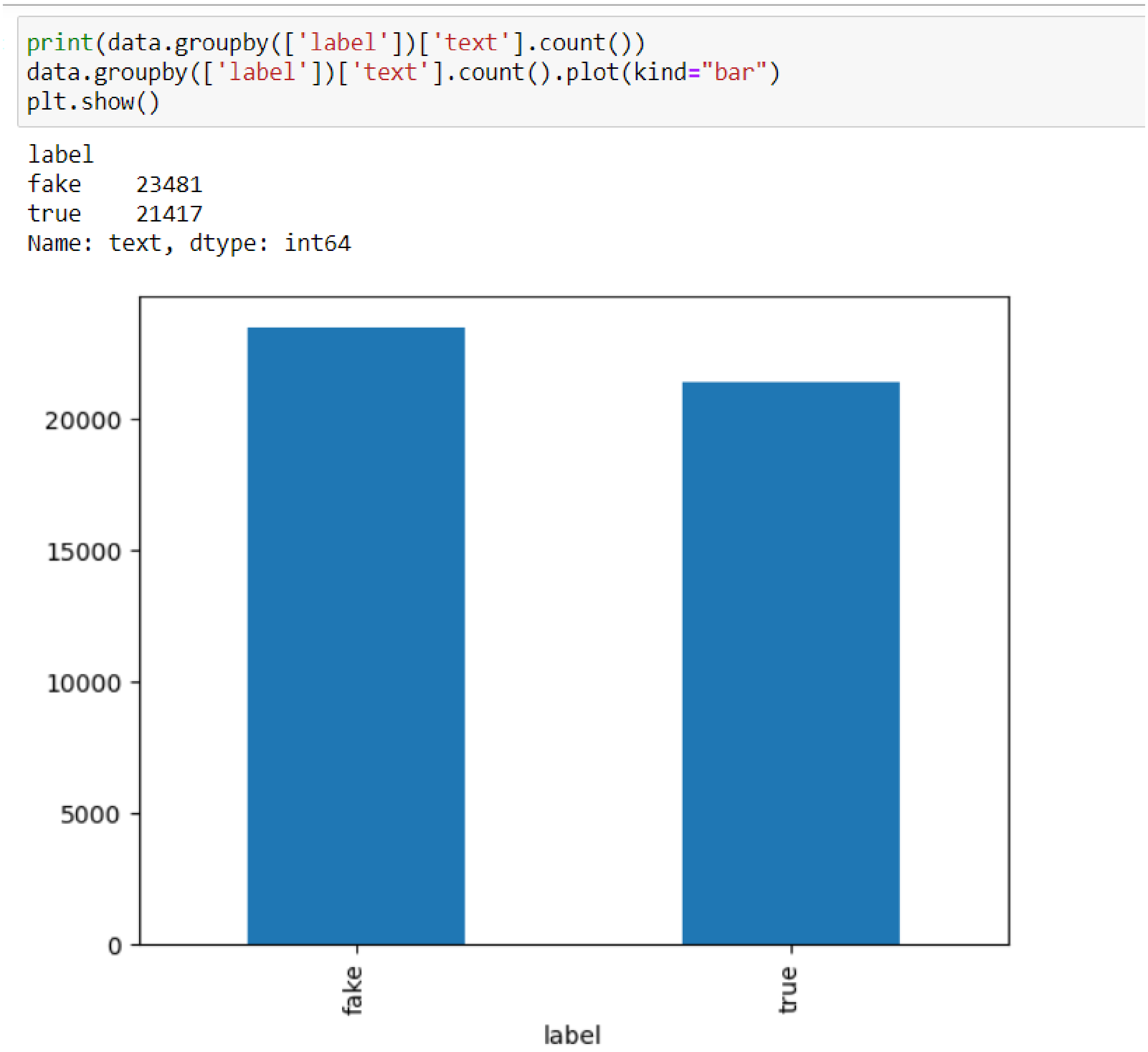
```
print(data.groupby(['subject'])['text'].count())
data.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()
```

```
subject
Government News     1570
Middle-east          778
News                9050
US_News              783
left-news           4459
politics            6841
politicsNews       11272
worldnews          10145
Name: text, dtype: int64
```



- In News Articles we can see the bar plot, political news are higher than other news.

Fake And Real News Articles



```
print(data.groupby(['label'])['text'].count())
data.groupby(['label'])['text'].count().plot(kind="bar")
plt.show()
```

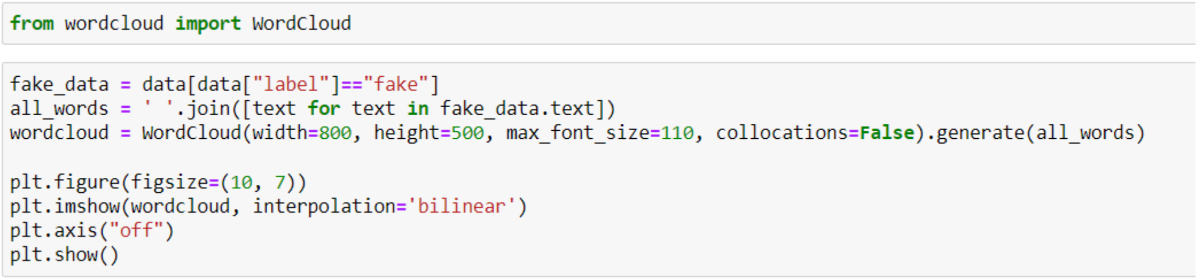
```
label
fake     23481
true     21417
Name: text, dtype: int64
```

We can see in the bar plot, the fake news is some high, but our data is balanced.

Install Word Cloud for Frequency of Images and Word

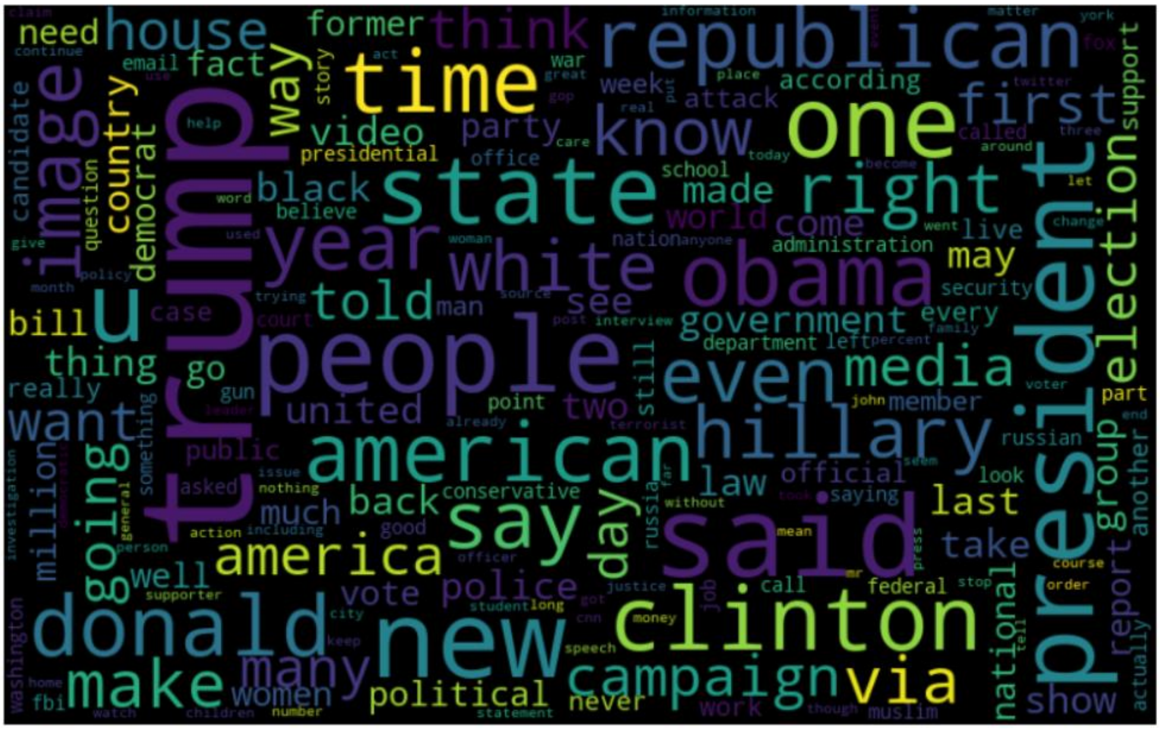
Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud.

Word Cloud Images for Fake News

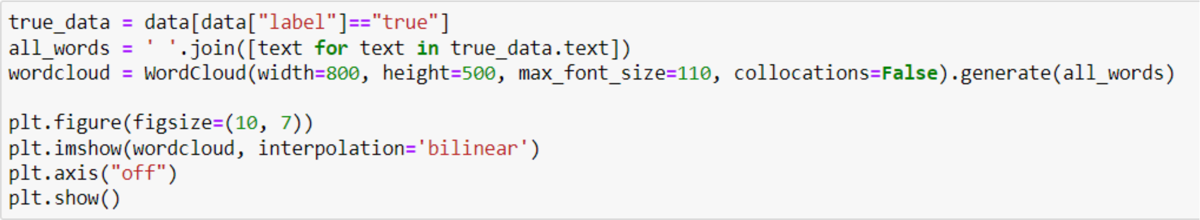


```
from wordcloud import WordCloud
```

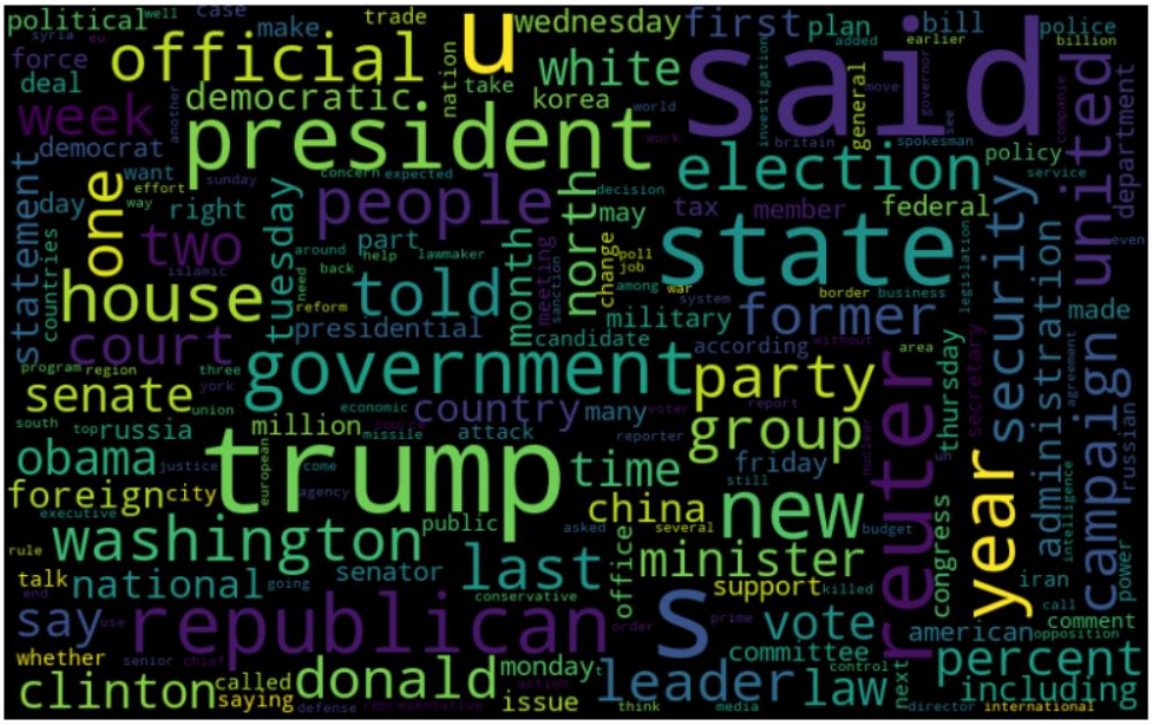
```
fake_data = data[data["label"]=="fake"]
all_words = ' '.join([text for text in fake_data.text])
wordcloud = WordCloud(width=800, height=500, max_font_size=110, collocations=False).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



Word Cloud Images for True News

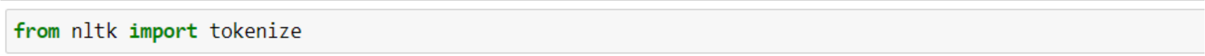


```
true_data = data[data["label"]=="true"]
all_words = ' '.join([text for text in true_data.text])
wordcloud = WordCloud(width=800, height=500, max_font_size=110, collocations=False).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

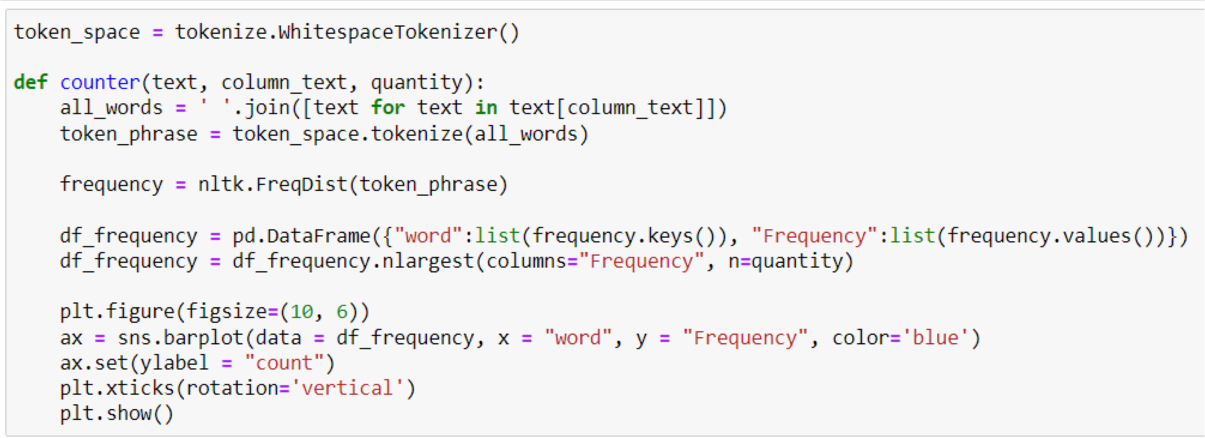


Most Frequent Words Counter (Tokenization)

Tokenization is used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned meaning. The first step of the NLP process is gathering the data (a sentence) and breaking it into understandable parts (words).

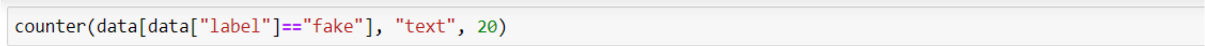


```
from nltk import tokenize
```

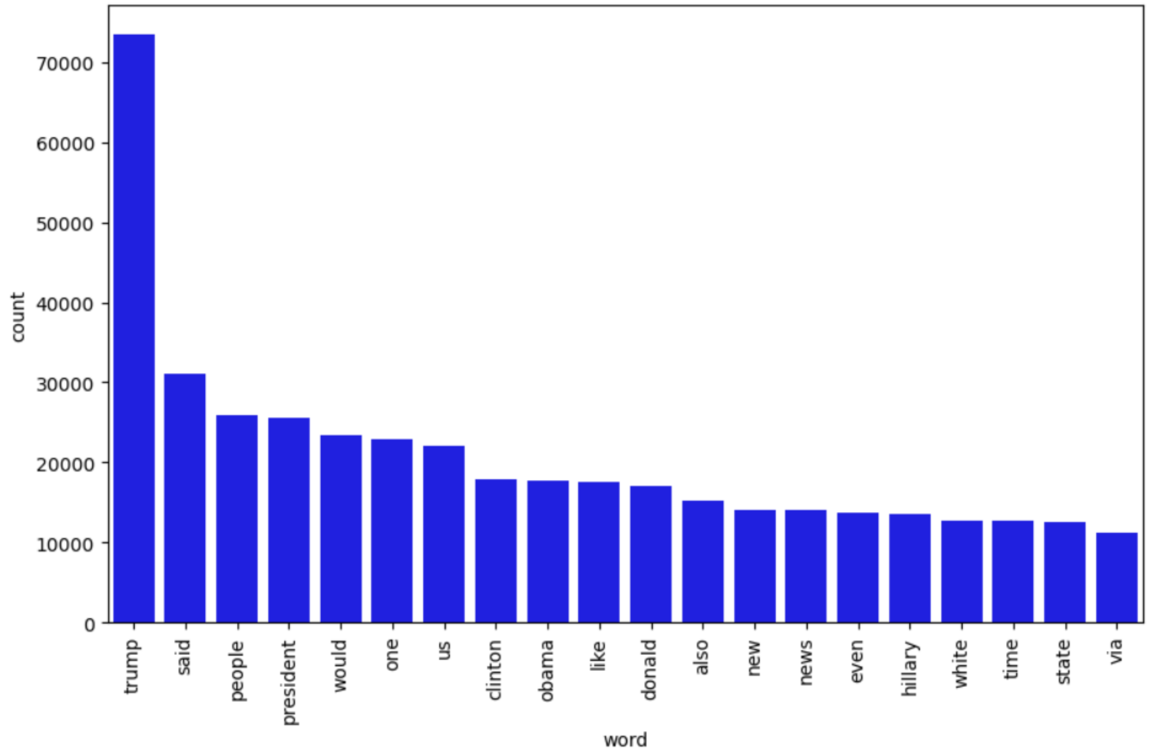


```
token_space = tokenize.WhitespaceTokenizer()

def counter(text, column_text, quantity):
    all_words = ' '.join([text for text in text[column_text]])
    token_phrase = token_space.tokenize(all_words)

    frequency = nltk.FreqDist(token_phrase)

    df_frequency = pd.DataFrame({"word":list(frequency.keys()), "Frequency":list(frequency.values())})
    df_frequency = df_frequency.nlargest(columns="Frequency", n=quantity)

    plt.figure(figsize=(10, 6))
    ax = sns.barplot(data = df_frequency, x = "word", y = "Frequency", color='blue')
    ax.set(ylabel = "count")
    plt.xticks(rotation='vertical')
    plt.show()
```

Most Frequent Words in Fake News

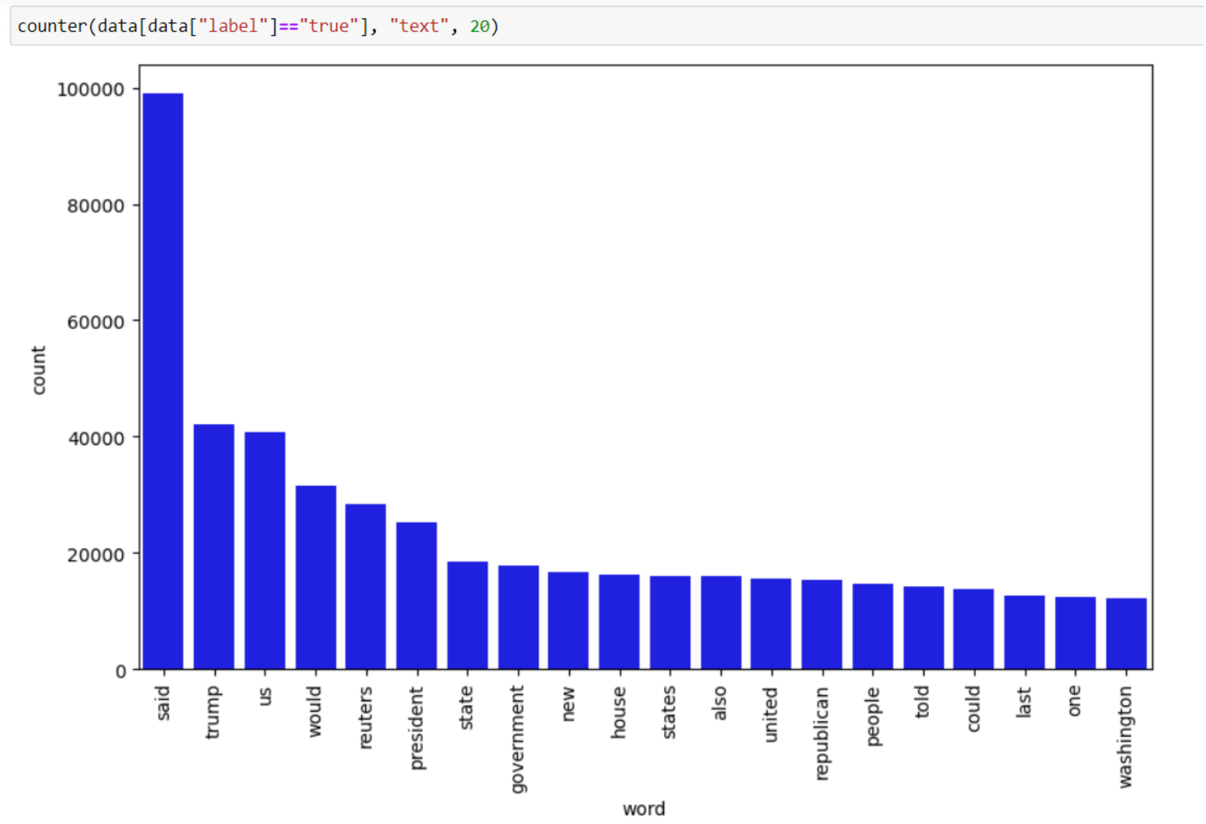


```
counter(data[data["label"]=="fake"], "text", 20)
```



We can see in fake news the most frequent word is 'trump', this word repeats mostly the time in fake news.

Most Frequent Words in Real News



```
counter(data[data["label"]=="true"], "text", 20)
```

We can see in true news the most frequent word is 'said', this word repeats mostly the time in true news.

Hardware & Software Requirements Tool Used

Hardware Used:

Processor — AMD Ryzen 5
RAM - 8 GB
ROM – 512 GB SSD
4GB Nvidia GEFORCE GTX Graphics card

Project Report on

FAKE NEWS DETECTION

Software utilized:

Anaconda – Jupyter Notebook

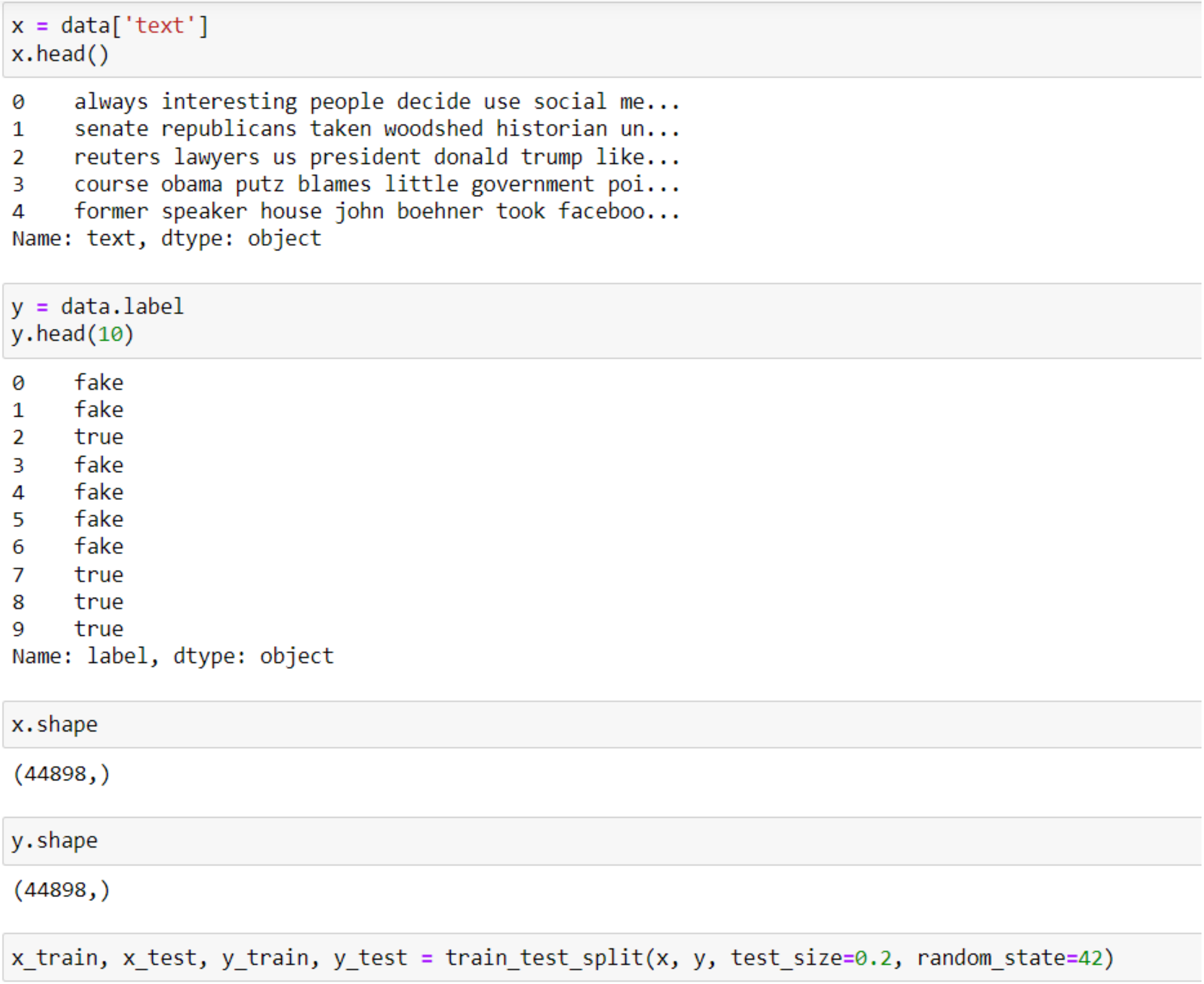
Models Development & Evaluation

IDENTIFICATION OF POSSIBLE PROBLEM-SOLVING APPROACHES:

- Our objective is to detect **fake news** and analyse whether the news is true or fake. This problem can be solved using Classification-based machine learning algorithms like Decision Tree Classifiers. For that purpose, the first task is to convert text data into numerical features with the help of the Vectorization Method.
- The final model is built over this scaled data. For building the ML model before implementing the classification algorithm, data is split into training & test data using train_test_split from the model selection module of the sklearn library.
- Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. After that model is trained with various classification algorithms and 5-fold cross-validation is performed.

Model Building

Separate The Data



```
x = data['text']
x.head()
```

```
0    always interesting people decide use social me...
1    senate republicans taken woodshed historian un...
2    reuters lawyers us president donald trump like...
3    course obama putz blames little government poi...
4    former speaker house john boehner took faceboo...
Name: text, dtype: object
```

```
y = data.label
y.head(10)
```

```
0    fake
1    fake
2    true
3    fake
4    fake
5    fake
6    fake
7    true
8    true
9    true
Name: label, dtype: object
```

```
x.shape
```

```
(44898,)
```

```
y.shape
```

```
(44898,)
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Testing of Identified Approaches (Algorithms)

The different classification algorithms used in this project to build the ML model are as below:

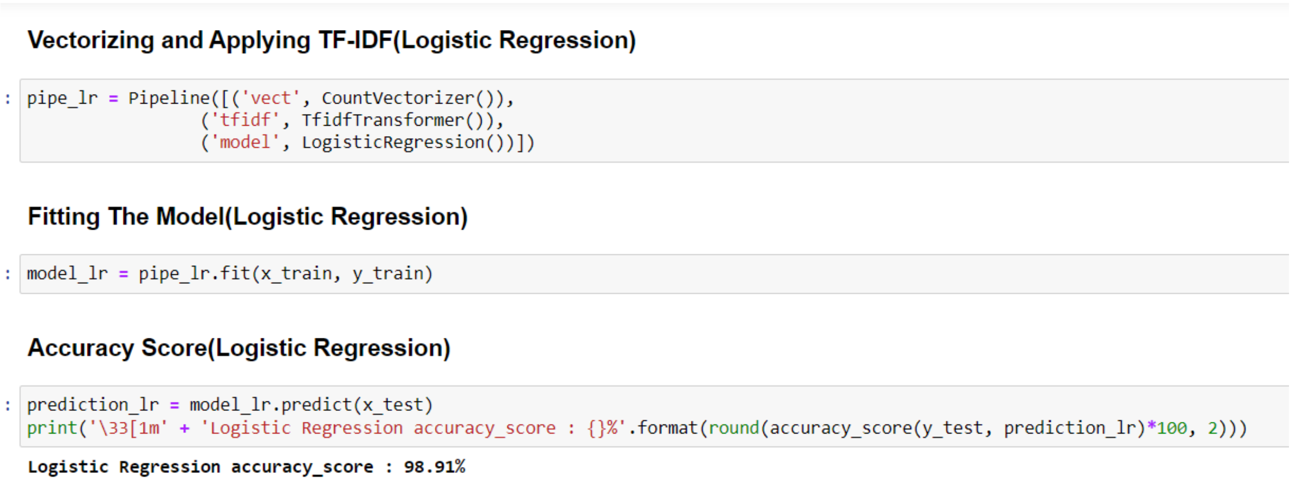
Project Report on

FAKE NEWS DETECTION

- Logistic Regression
- Random Forest Classifier
- Decision Tree Classifier
- KNeighbors Classifier

RUN AND EVALUATE SELECTED MODELS

Logistic Regression:



Vectorizing and Applying TF-IDF(Logistic Regression)

```
pipe_lr = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])
```

Fitting The Model(Logistic Regression)

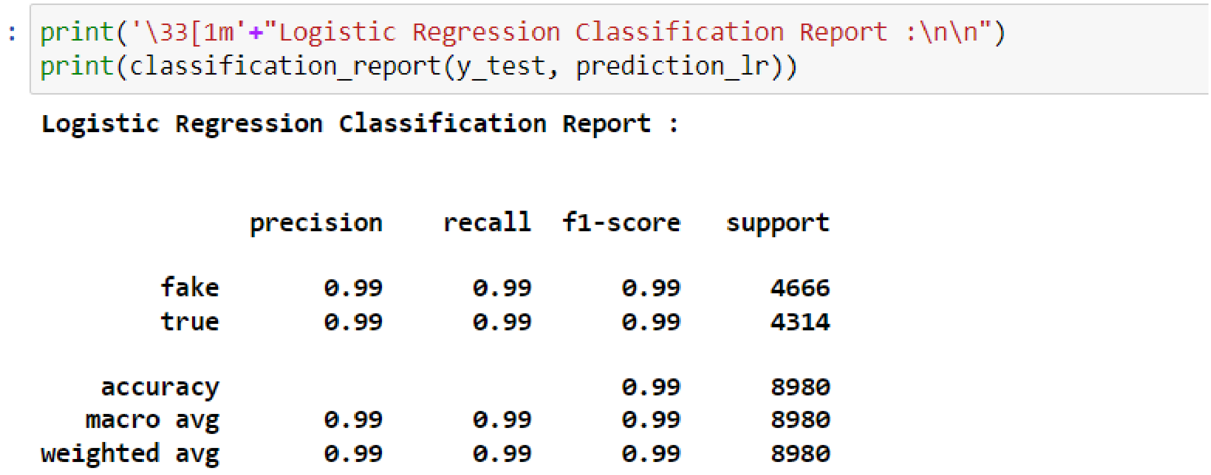
```
model_lr = pipe_lr.fit(x_train, y_train)
```

Accuracy Score(Logistic Regression)

```
prediction_lr = model_lr.predict(x_test)
print('\33[1m' + 'Logistic Regression accuracy_score : {}%'.format(round(accuracy_score(y_test, prediction_lr)*100, 2)))
```

```
Logistic Regression accuracy_score : 98.91%
```

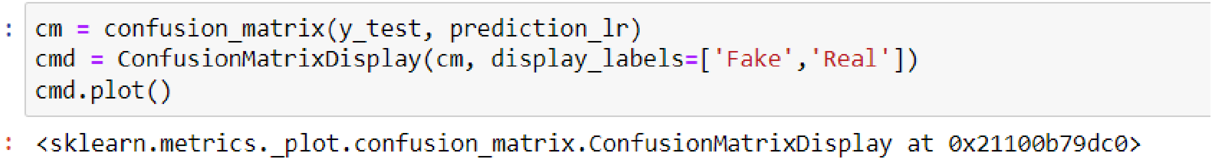
Classification Report(Logistic Regression)



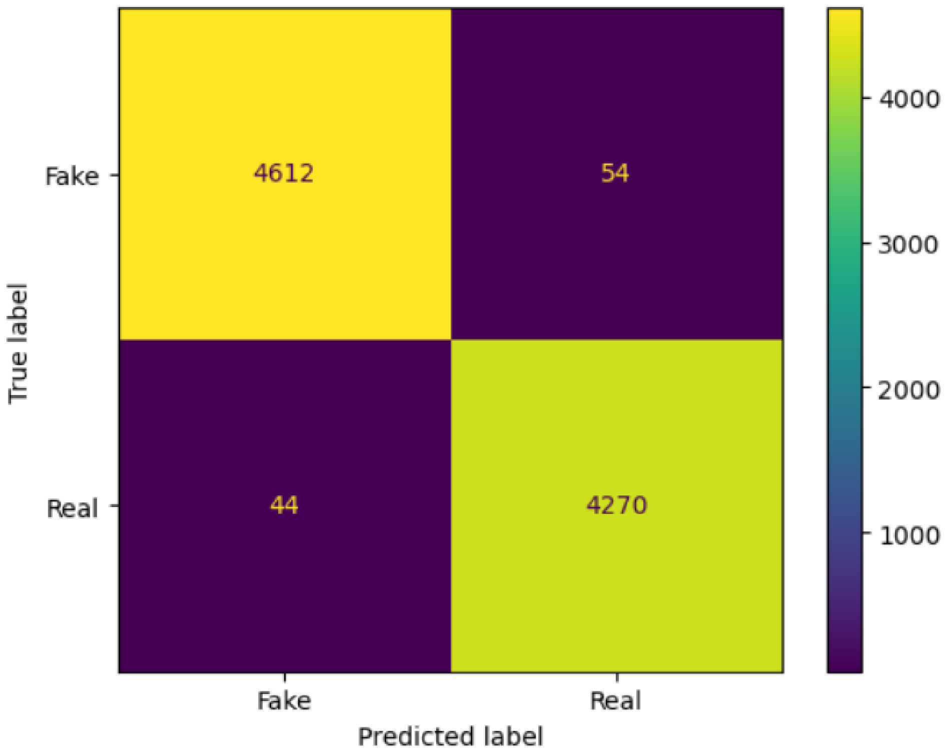
```
: print('\33[1m'+"Logistic Regression Classification Report :\n\n")
  print(classification_report(y_test, prediction_lr))
```

```
Logistic Regression Classification Report :


              precision    recall  f1-score   support

        fake       0.99      0.99      0.99      4666
        true       0.99      0.99      0.99      4314

    accuracy                           0.99      8980
   macro avg       0.99      0.99      0.99      8980
weighted avg       0.99      0.99      0.99      8980
```

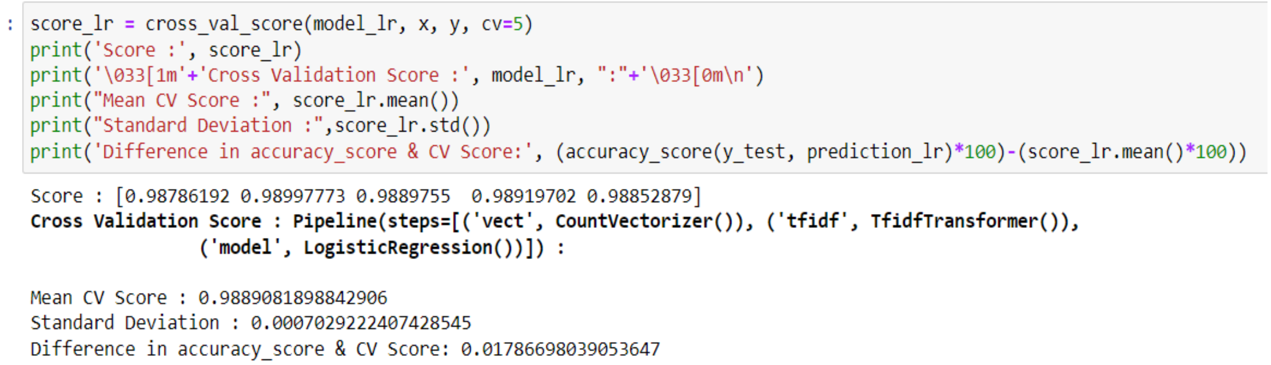
Confusion Matrix(Logistic Regression)



```
: cm = confusion_matrix(y_test, prediction_lr)
  cmd = ConfusionMatrixDisplay(cm, display_labels=['Fake','Real'])
  cmd.plot()

: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x21100b79dc0>
```

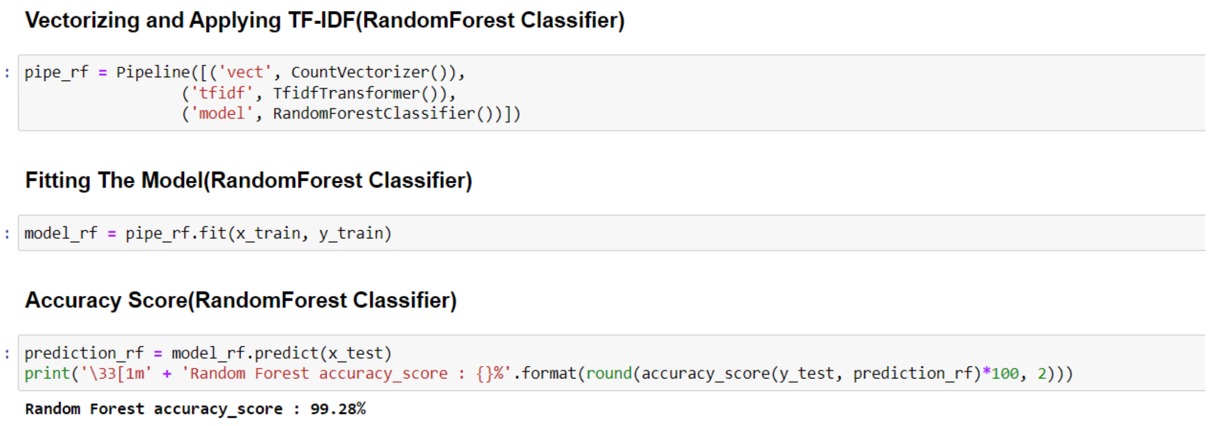


Cross Validation Score(Logistic Regression)



```
: score_lr = cross_val_score(model_lr, x, y, cv=5)
  print('Score :', score_lr)
  print('\033[1m'+'Cross Validation Score :', model_lr, ":"+'\033[0m\n')
  print("Mean CV Score :", score_lr.mean())
  print("Standard Deviation :",score_lr.std())
  print('Difference in accuracy_score & CV Score:', (accuracy_score(y_test, prediction_lr)*100)-(score_lr.mean()*100))

  Score : [0.98786192 0.98997773 0.9889755  0.98919702 0.98852879]
  Cross Validation Score : Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                  ('model', LogisticRegression())]) :

  Mean CV Score : 0.9889081898842906
  Standard Deviation : 0.0007029222407428545
  Difference in accuracy_score & CV Score: 0.01786698039053647
```

Random Forest Classifier



Vectorizing and Applying TF-IDF(RandomForest Classifier)

```
: pipe_rf = Pipeline([('vect', CountVectorizer()),
                  ('tfidf', TfidfTransformer()),
                  ('model', RandomForestClassifier())])
```

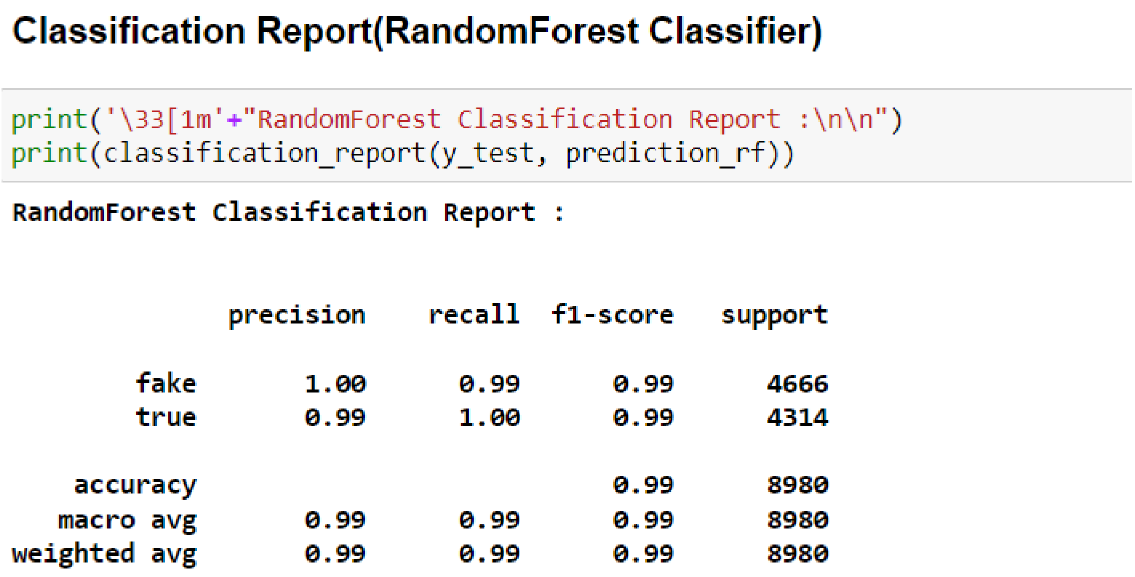
Fitting The Model(RandomForest Classifier)

```
: model_rf = pipe_rf.fit(x_train, y_train)
```

Accuracy Score(RandomForest Classifier)

```
: prediction_rf = model_rf.predict(x_test)
  print('\33[1m' + 'Random Forest accuracy_score : {}%'.format(round(accuracy_score(y_test, prediction_rf)*100, 2)))
```

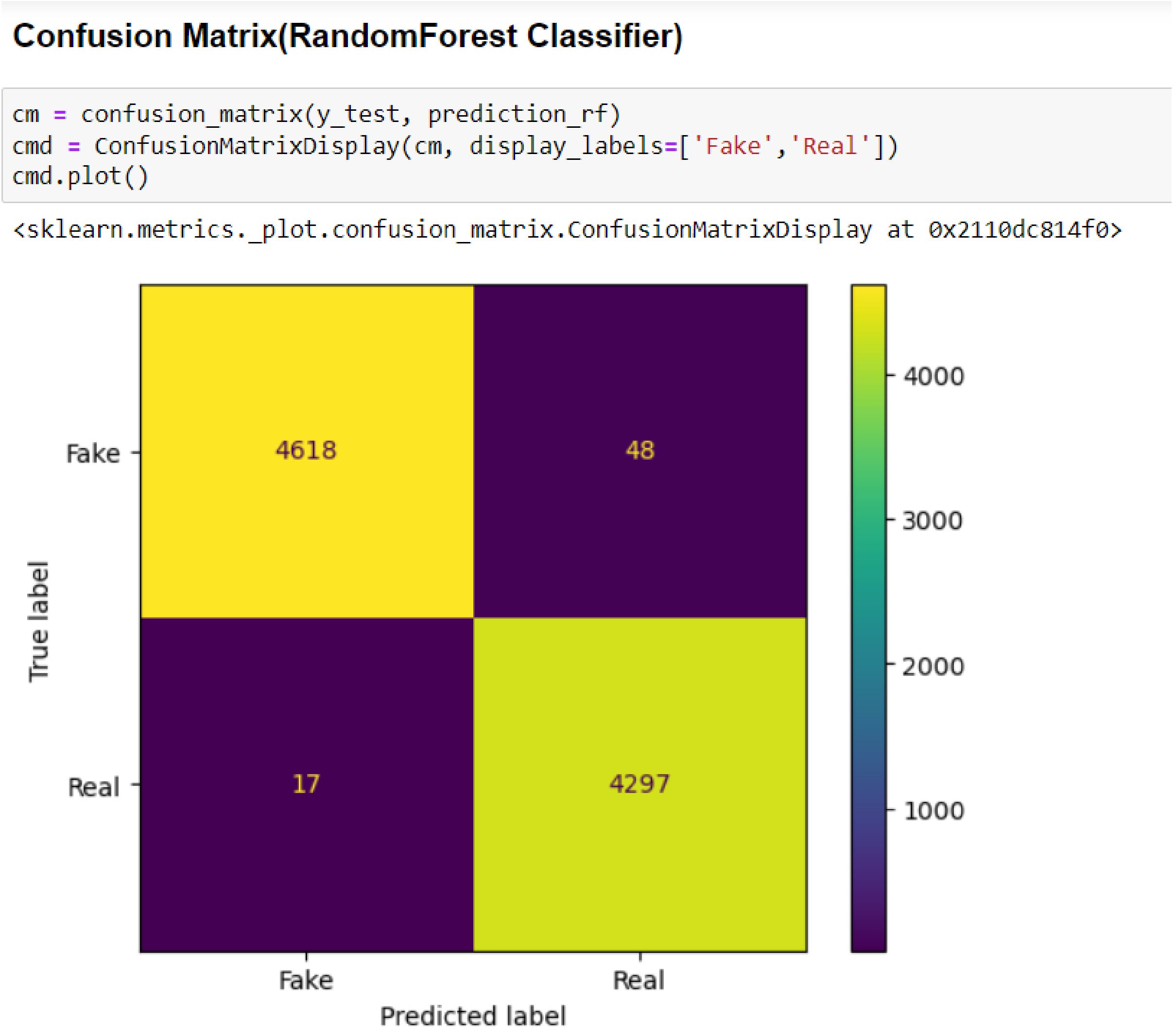
```
Random Forest accuracy_score : 99.28%
```



Classification Report(RandomForest Classifier)

```
print('\33[1m'+"RandomForest Classification Report :\n\n")
print(classification_report(y_test, prediction_rf))
```

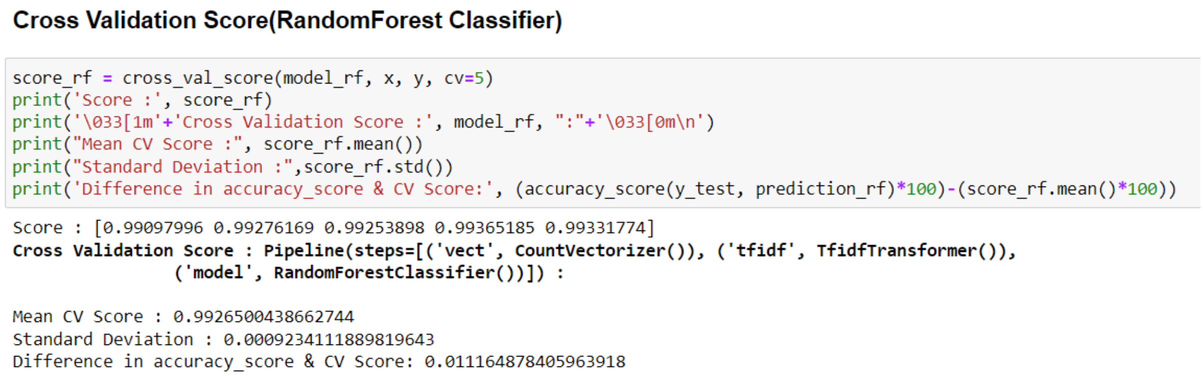
```
RandomForest Classification Report :


              precision    recall  f1-score   support

        fake       1.00      0.99      0.99      4666
        true       0.99      1.00      0.99      4314

    accuracy                           0.99      8980
   macro avg       0.99      0.99      0.99      8980
weighted avg       0.99      0.99      0.99      8980
```



Confusion Matrix(RandomForest Classifier)

```
cm = confusion_matrix(y_test, prediction_rf)
cmd = ConfusionMatrixDisplay(cm, display_labels=['Fake','Real'])
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2110dc814f0>
```

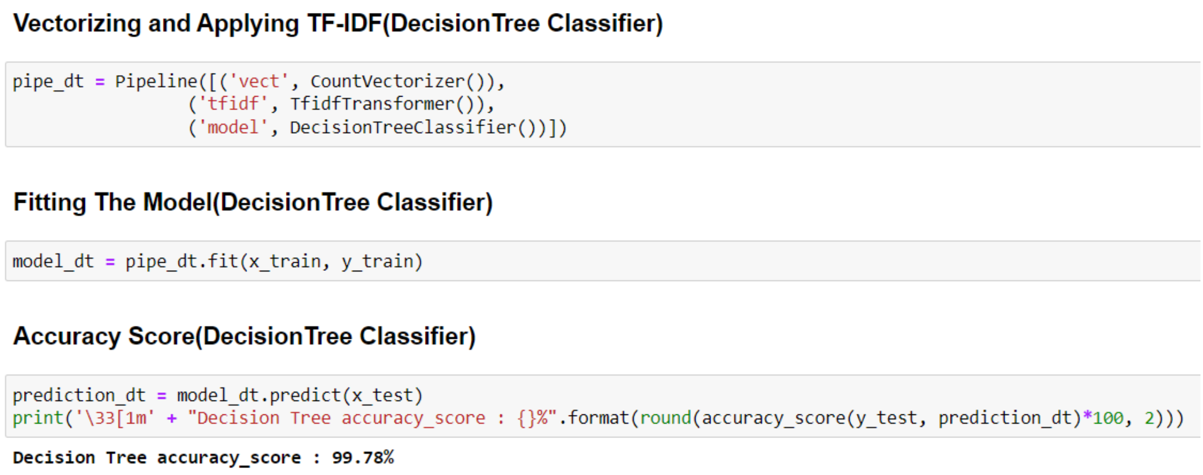


Cross Validation Score(RandomForest Classifier)

```
score_rf = cross_val_score(model_rf, x, y, cv=5)
print('Score :', score_rf)
print('\033[1m'+'Cross Validation Score :', model_rf, ":"+'\033[0m\n')
print("Mean CV Score :", score_rf.mean())
print("Standard Deviation :",score_rf.std())
print('Difference in accuracy_score & CV Score:', (accuracy_score(y_test, prediction_rf)*100)-(score_rf.mean()*100))
```

```
Score : [0.99097996 0.99276169 0.99253898 0.99365185 0.99331774]
Cross Validation Score : Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('model', RandomForestClassifier())]) :

Mean CV Score : 0.9926500438662744
Standard Deviation : 0.0009234111889819643
Difference in accuracy_score & CV Score: 0.011164878405963918
```

Decision Tree Classifier



Vectorizing and Applying TF-IDF(DecisionTree Classifier)

```
pipe_dt = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier())])
```

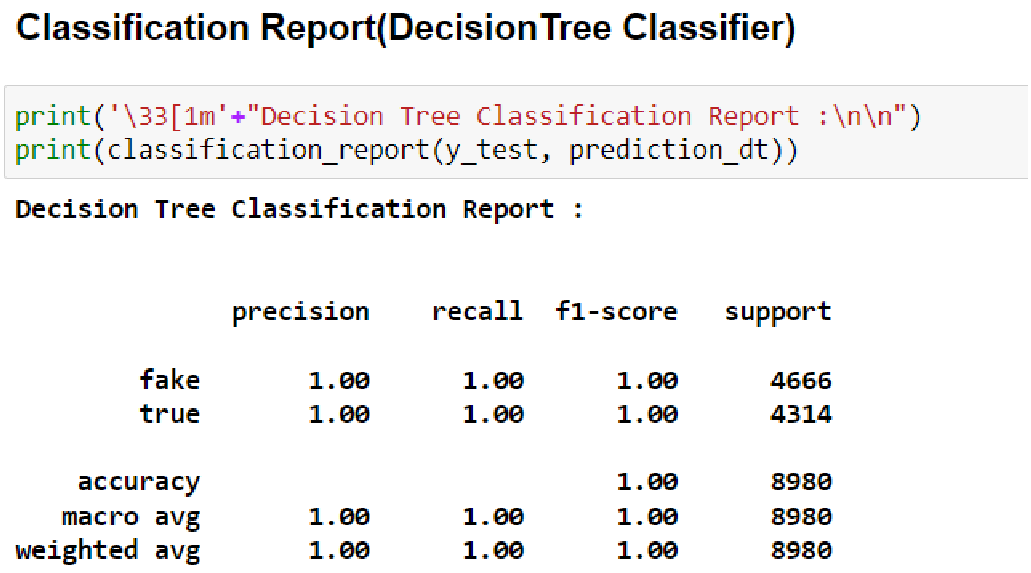
Fitting The Model(DecisionTree Classifier)

```
model_dt = pipe_dt.fit(x_train, y_train)
```

Accuracy Score(DecisionTree Classifier)

```
prediction_dt = model_dt.predict(x_test)
print('\33[1m' + "Decision Tree accuracy_score : {}%".format(round(accuracy_score(y_test, prediction_dt)*100, 2)))
```

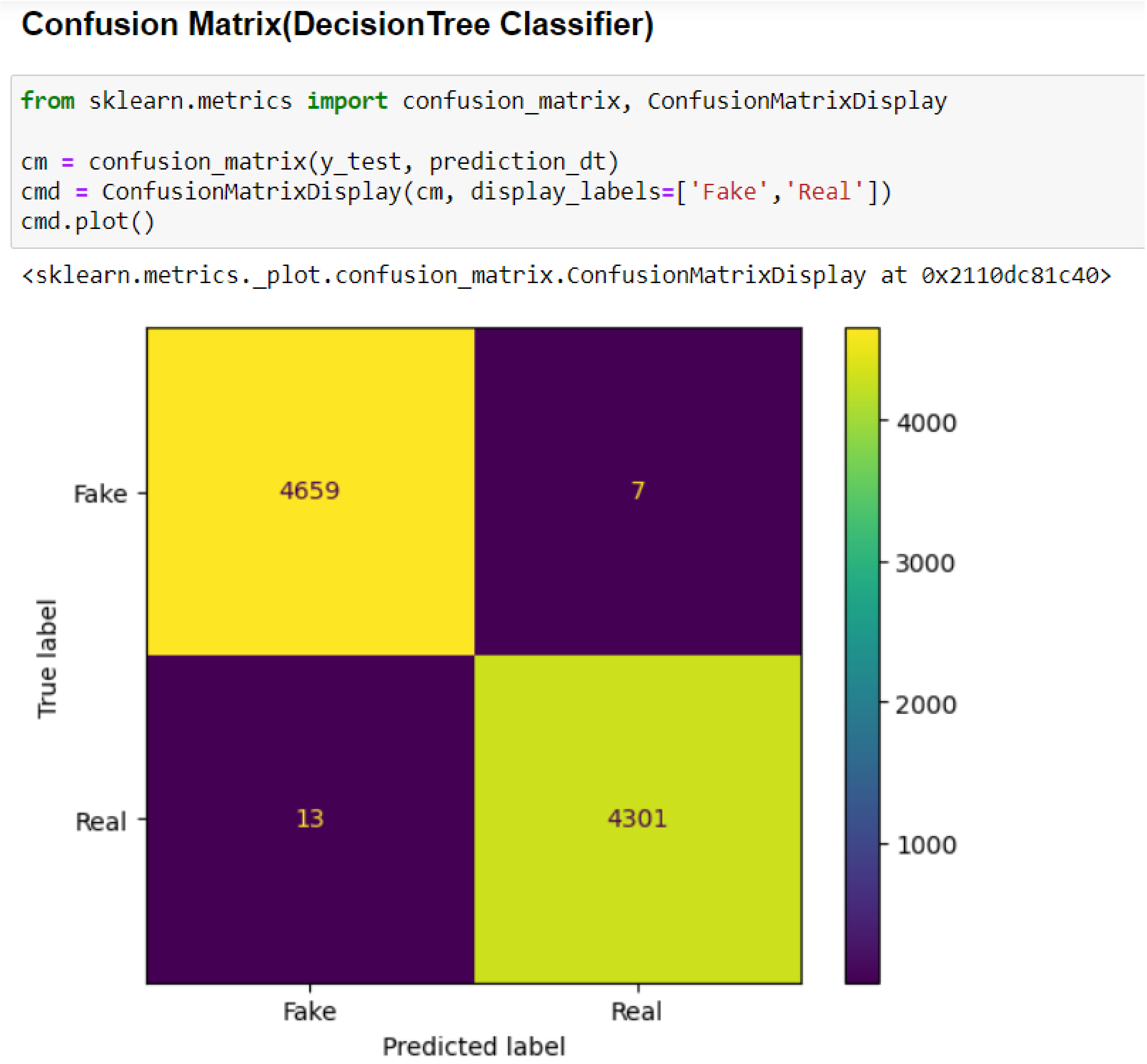
```
Decision Tree accuracy_score : 99.78%
```



Classification Report(DecisionTree Classifier)

```
print('\33[1m'+"Decision Tree Classification Report :\n\n")
print(classification_report(y_test, prediction_dt))
```

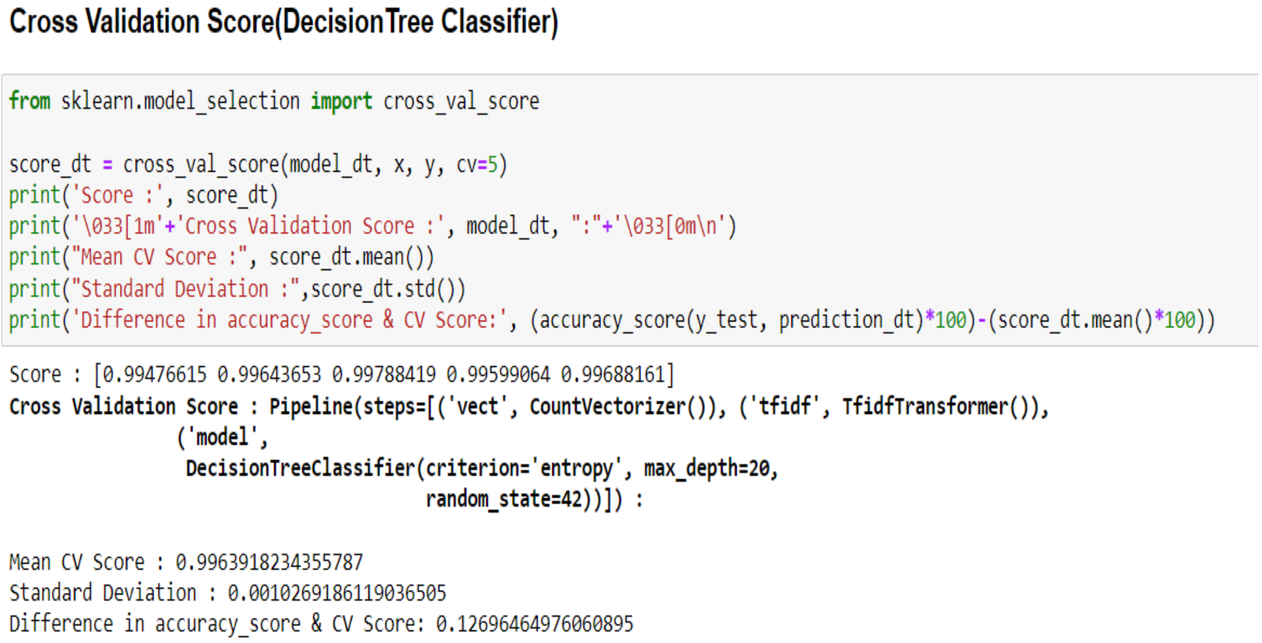
```
Decision Tree Classification Report :


              precision    recall  f1-score   support

        fake       1.00      1.00      1.00      4666
        true       1.00      1.00      1.00      4314

    accuracy                           1.00      8980
   macro avg       1.00      1.00      1.00      8980
weighted avg       1.00      1.00      1.00      8980
```



Confusion Matrix(DecisionTree Classifier)

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, prediction_dt)
cmd = ConfusionMatrixDisplay(cm, display_labels=['Fake','Real'])
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2110dc81c40>
```

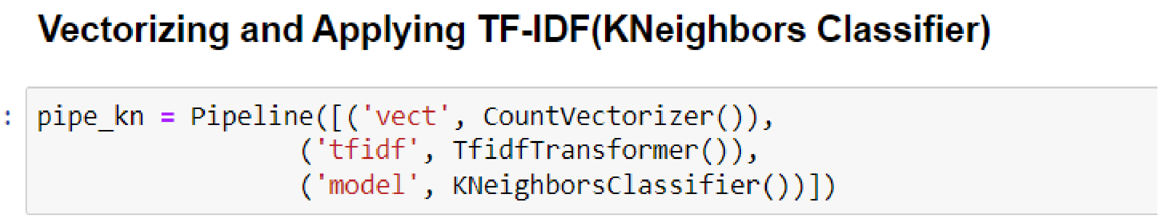


Cross Validation Score(DecisionTree Classifier)

```
from sklearn.model_selection import cross_val_score

score_dt = cross_val_score(model_dt, x, y, cv=5)
print('Score :', score_dt)
print('\033[1m'+'Cross Validation Score :', model_dt, ":"+'\033[0m\n')
print("Mean CV Score :", score_dt.mean())
print("Standard Deviation :",score_dt.std())
print('Difference in accuracy_score & CV Score:', (accuracy_score(y_test, prediction_dt)*100)-(score_dt.mean()*100))
```

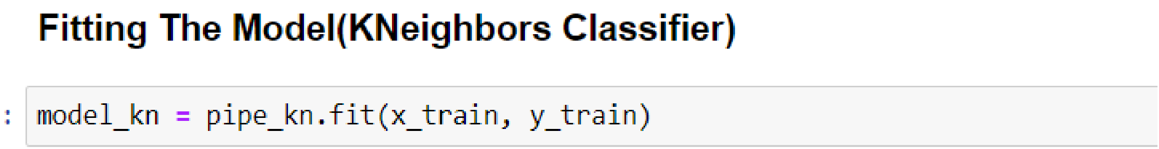
```
Score : [0.99476615 0.99643653 0.99788419 0.99599064 0.99688161]
Cross Validation Score : Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('model',
                 DecisionTreeClassifier(criterion='entropy', max_depth=20,
                                        random_state=42))]) :

Mean CV Score : 0.9963918234355787
Standard Deviation : 0.0010269186119036505
Difference in accuracy_score & CV Score: 0.12696464976060895
```

KNeighbors Classifier



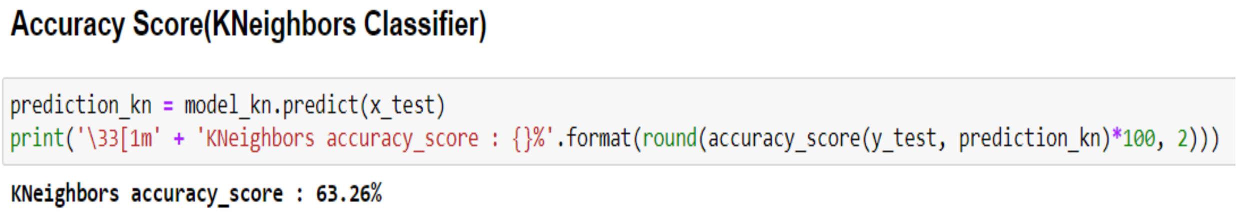
Vectorizing and Applying TF-IDF(KNeighbors Classifier)

```
: pipe_kn = Pipeline([('vect', CountVectorizer()),
                      ('tfidf', TfidfTransformer()),
                      ('model', KNeighborsClassifier())])
```



Fitting The Model(KNeighbors Classifier)

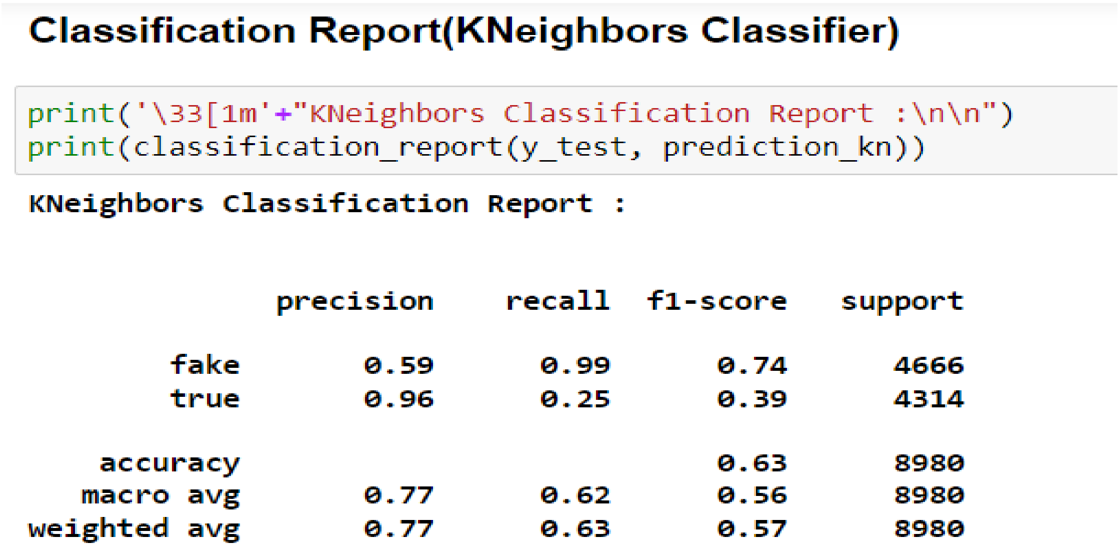
```
: model_kn = pipe_kn.fit(x_train, y_train)
```



Accuracy Score(KNeighbors Classifier)

```
prediction_kn = model_kn.predict(x_test)
print('\33[1m' + 'KNeighbors accuracy_score : {}%'.format(round(accuracy_score(y_test, prediction_kn)*100, 2)))
```

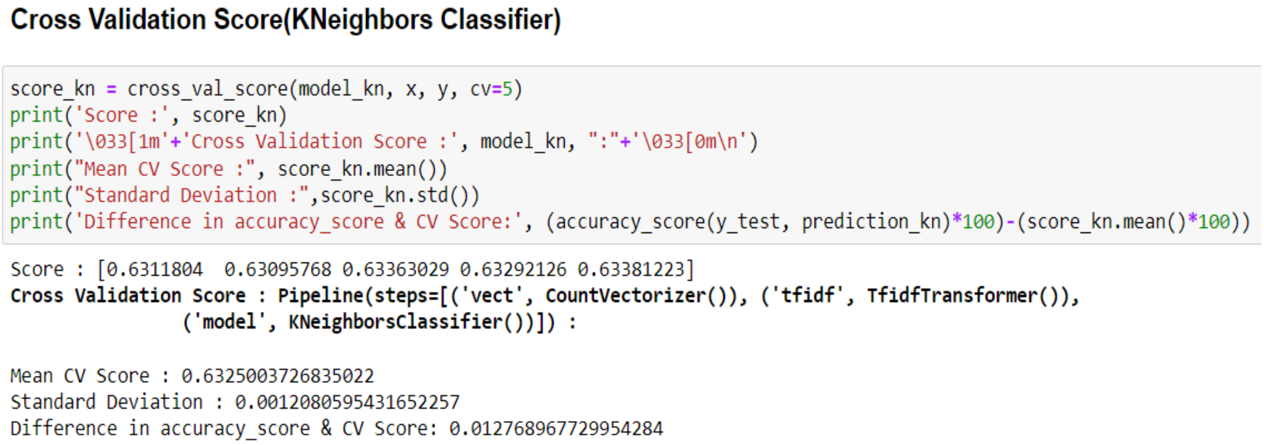
```
KNeighbors accuracy_score : 63.26%
```



Classification Report(KNeighbors Classifier)

```
print('\33[1m'+"KNeighbors Classification Report :\n\n")
print(classification_report(y_test, prediction_kn))
```

```
KNeighbors Classification Report :


              precision    recall  f1-score   support

        fake       0.59      0.99      0.74      4666
        true       0.96      0.25      0.39      4314

    accuracy                           0.63      8980
   macro avg       0.77      0.62      0.56      8980
weighted avg       0.77      0.63      0.57      8980
```

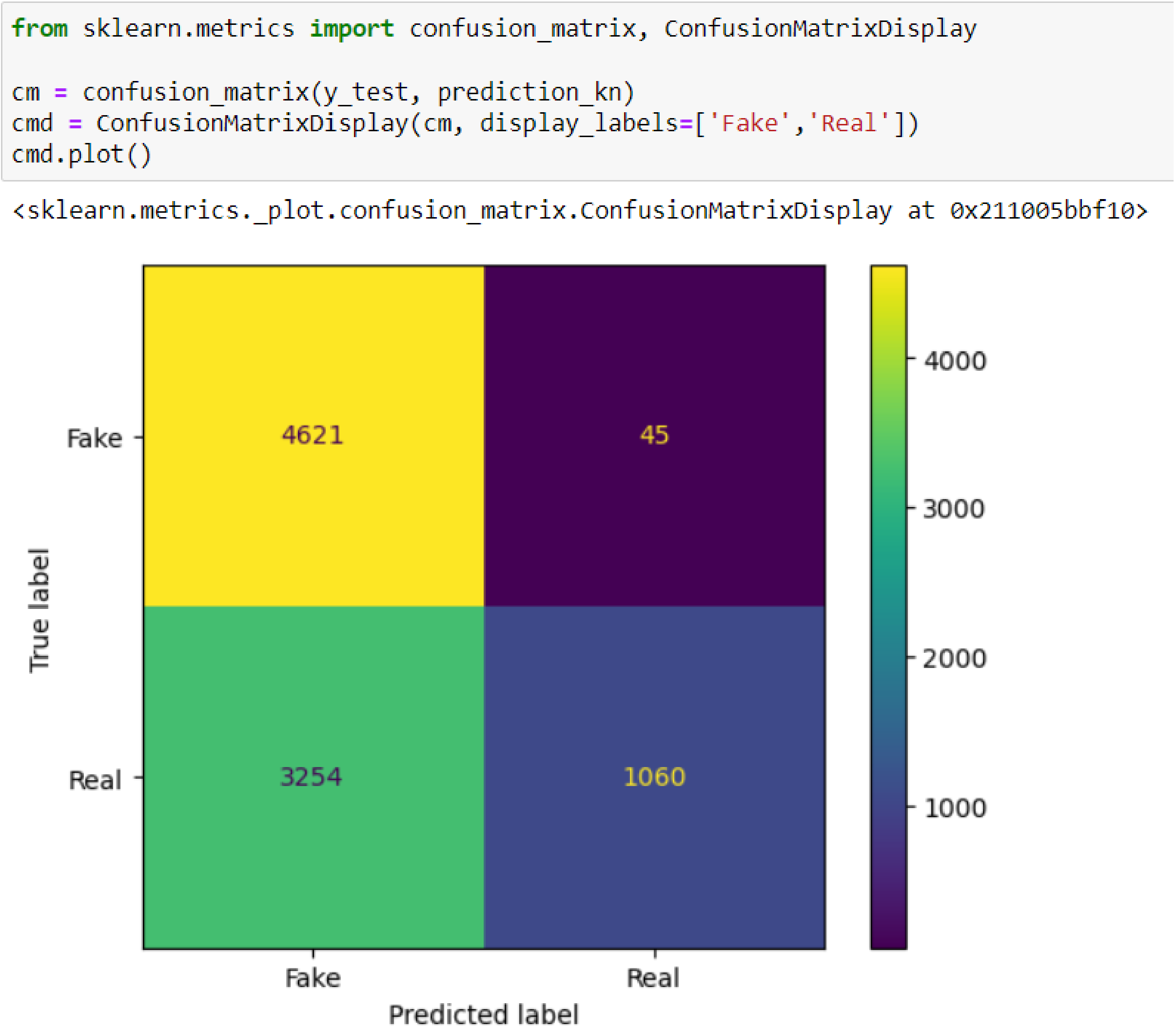


Cross Validation Score(KNeighbors Classifier)

```
score_kn = cross_val_score(model_kn, x, y, cv=5)
print('Score :', score_kn)
print('\033[1m'+'Cross Validation Score :', model_kn, ":"+'\033[0m\n')
print("Mean CV Score :", score_kn.mean())
print("Standard Deviation :",score_kn.std())
print('Difference in accuracy_score & CV Score:', (accuracy_score(y_test, prediction_kn)*100)-(score_kn.mean()*100))
```

```
Score : [0.6311804  0.63095768 0.63363029 0.63292126 0.63381223]
Cross Validation Score : Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('model', KNeighborsClassifier())]) :

Mean CV Score : 0.6325003726835022
Standard Deviation : 0.0012080595431652257
Difference in accuracy_score & CV Score: 0.012768967729954284
```

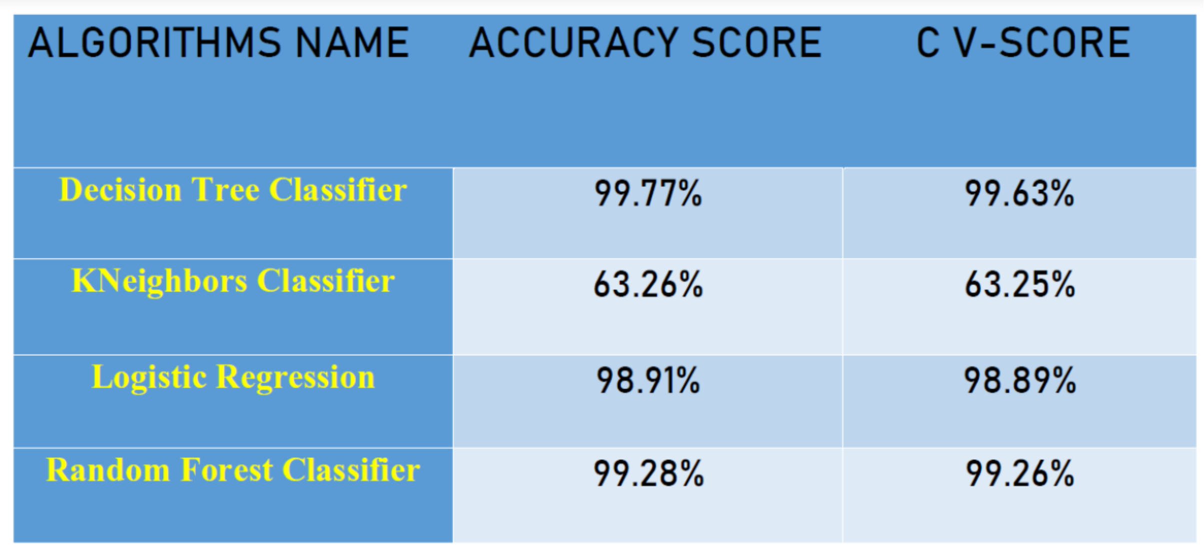
Confusion Matrix(KNeighbors Classifier)



```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, prediction_kn)
cmd = ConfusionMatrixDisplay(cm, display_labels=['Fake','Real'])
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x211005bbf10>
```

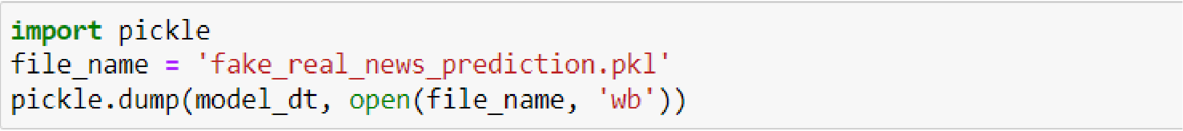
Algorithms Use in This Model:



ALGORITHMS NAME	ACCURACY SCORE	C V-SCORE
Decision Tree Classifier	99.77%	99.63%
KNeighbors Classifier	63.26%	63.25%
Logistic Regression	98.91%	98.89%
Random Forest Classifier	99.28%	99.26%

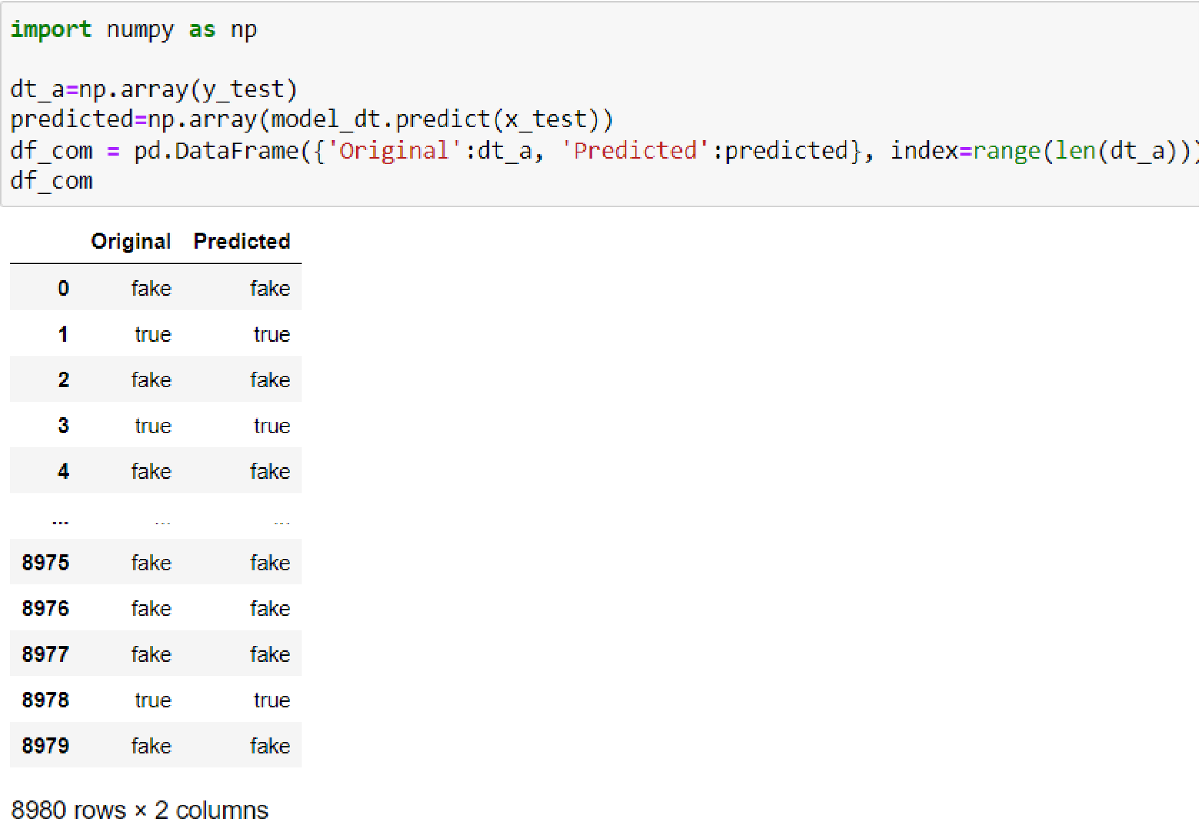
- We can see that the Decision Tree classifier and Random Forest Classifier both are the best accuracy score but the Decision Tree Classifier is higher by one or two points.
- Decision Tree Classifier gives maximum accuracy score of 99.77% and with cross validation score of 99.63%
- Decision Tree Classifier is a Final Model

Saving Final Model(DecisionTree Classifier)



```
import pickle
file_name = 'fake_real_news_prediction.pkl'
pickle.dump(model_dt, open(file_name, 'wb'))
```

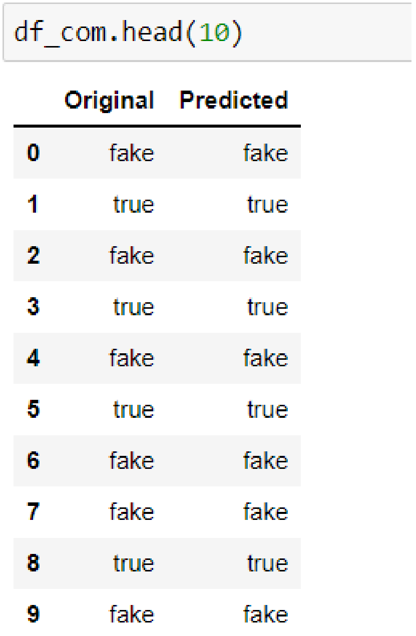
Predictions of Test Dataset Using Final Model



```
import numpy as np

dt_a=np.array(y_test)
predicted=np.array(model_dt.predict(x_test))
df_com = pd.DataFrame({'Original':dt_a, 'Predicted':predicted}, index=range(len(dt_a))
df_com
```

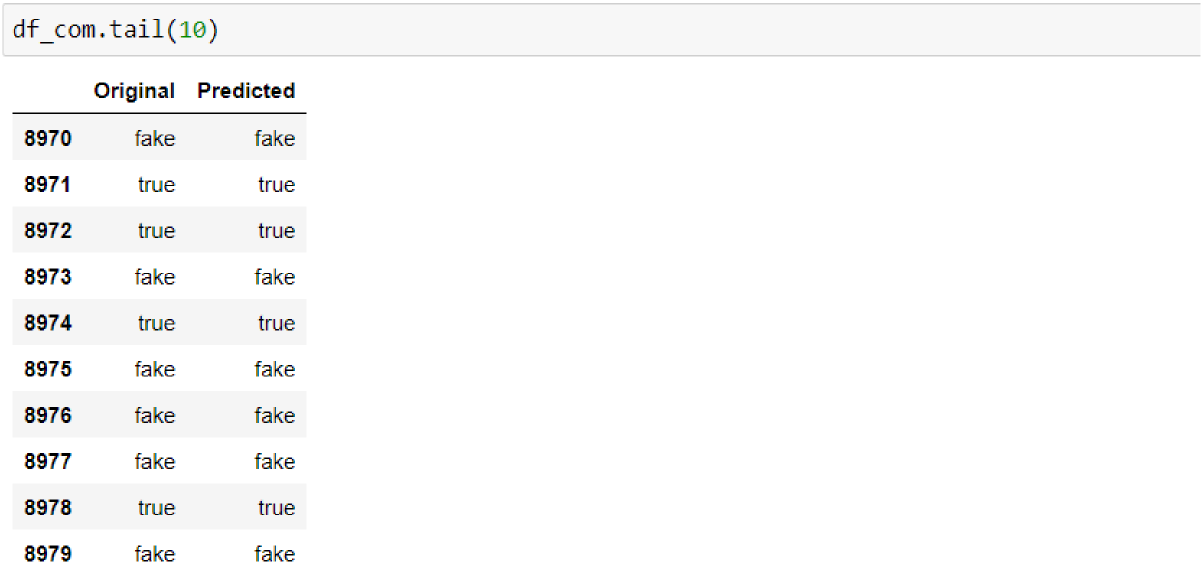
	Original	Predicted
0	fake	fake
1	true	true
2	fake	fake
3	true	true
4	fake	fake
...	...	...
8975	fake	fake
8976	fake	fake
8977	fake	fake
8978	true	true
8979	fake	fake

8980 rows × 2 columns



```
df_com.head(10)
```

	Original	Predicted
0	fake	fake
1	true	true
2	fake	fake
3	true	true
4	fake	fake
5	true	true
6	fake	fake
7	fake	fake
8	true	true
9	fake	fake



```
df_com.tail(10)
```

	Original	Predicted
8970	fake	fake
8971	true	true
8972	true	true
8973	fake	fake
8974	true	true
8975	fake	fake
8976	fake	fake
8977	fake	fake
8978	true	true
8979	fake	fake

We Can Visualize original and Predicted Value are 99.80% Correct Value

Conclusion

Key Findings and Conclusions of the Study

ALGORITHMS NAME	ACCURACY SCORE	C V-SCORE
Random Forest Classifier	99.28%	99.26%
KNeighbors Classifier	63.26%	63.25%
Logistic Regression	98.91%	98.89%
Decision Tree Classifier	99.78%	99.63%

Decision Tree Classifier gives us a maximum Accuracy Score of 99.78%, So Decision Tree Classifier is selected as the best model.