FAKE NEWS DETECTION USING MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology

In

Computer Science and Engineering/Information Technology

By

YASH KATARIA (191526)
AAYUSH KAKKAR (191528)

UNDER THE SUPERVISION OF
Dr. Emjee Puthooran
and
Mr. Praveen Modi
to

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
juit
विद्या तत्व ज्योतिसमः

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE DECLARATION

I hereby declare that the work presented in this report entitled **“Fake News Detection Using Machine Learning”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat, is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Emjee Puthooran** (Associate Professor in Electronics and Communication Department) and Co-Supervisor **Mr. Praveen Modi** (Assistant Professor (Grade 1) in CSE & IT Department.

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Data Science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Yash Kataria, 191526

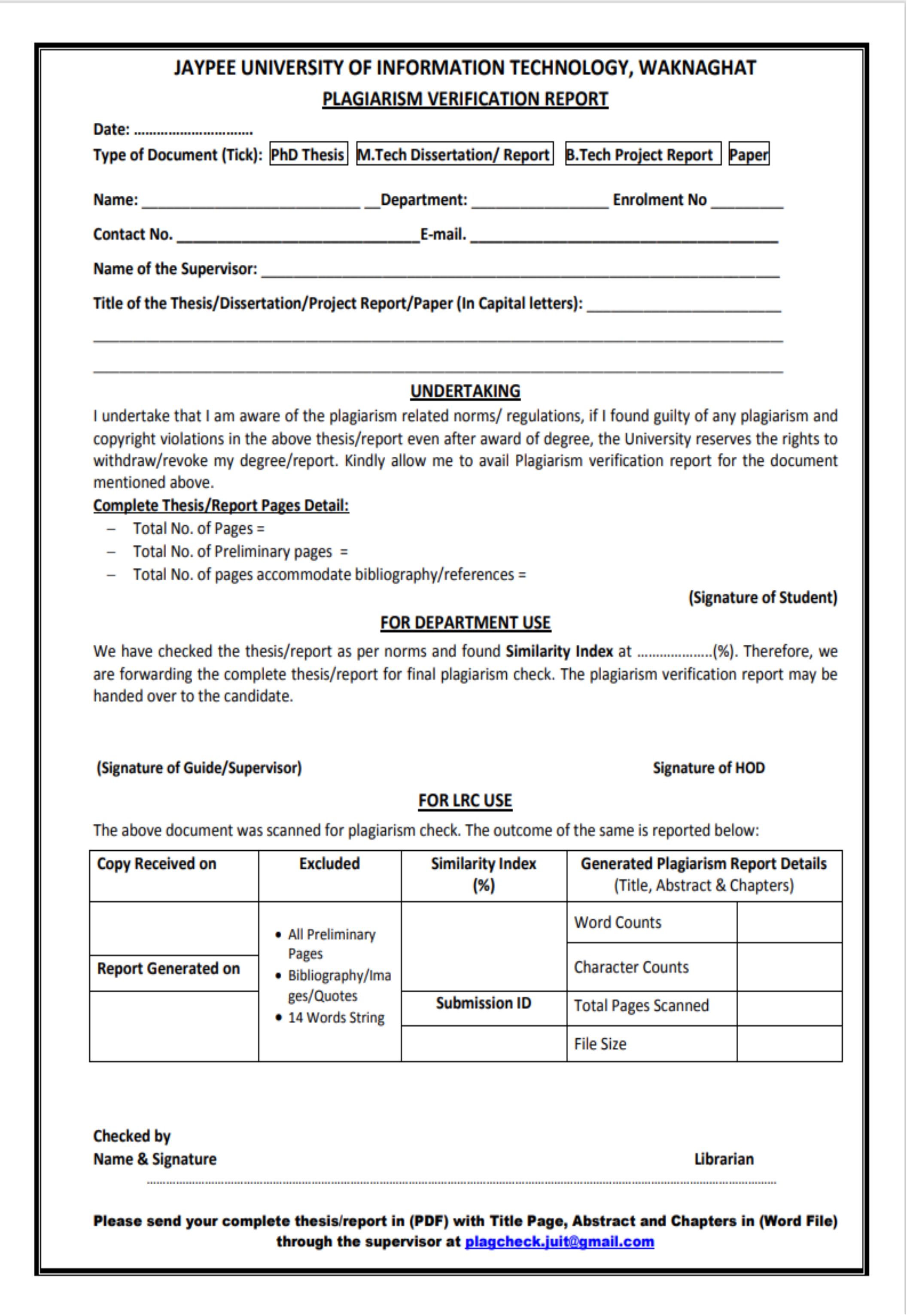
Aayush Kakkar, 191528

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)
Dr. Emjee Puthooran
Associate Professor
ECE Department
Dated:

(Co-Supervisor Signature)
Mr. Praveen Modi
Assistant Professor (Grade-1)
(CSE & IT) Department
Dated:

PLAGIARISM CERTIFICATE



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: __________________________ __**Department:** ________________ **Enrolment No** ________

Contact No. _____________________________**E-mail.** ____________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

With the guidance and assistance of numerous well-wishers, an endeavor over a lengthy period of time can be effective. We would like to take this time to let everyone know how much we appreciate them.

In the beginning, we'd like to express our gratitude to our supervisor, Dr. Emjee Puthooran, Associate Professor, Department of Electronics & Communication Engineering, and co-supervisor, Mr. Praveen Modi, Assistant Professor (Grade-I), Department of Computer Science & Engineering/Information Technology at Jaypee University of Information Technology (JUIT), for their invaluable support and direction throughout the project's implementation.

We wish to express our sincere thanks and gratitude to our project guide, **Dr. Emjee Puthooran**, Associate Professor, Department of Electronics and Communication Engineering, and **Mr. Praveen Modi**, Assistant Professor (Grade I), Department of Computer Science & Engineering/Information Technology at Jaypee University of Information Technology (JUIT), for the stimulating discussions, in analyzing problems associated with our project work, and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence she has shown in us. We are immensely indebted to her for her valuable guidance throughout our project.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

SHORT FORM		MEANINGS
TFIDF	=	Term Frequency - Inverse Document Frequency
SVM	=	Support Vector Machine
DT	=	Decision Tree
GBC	=	Gradient Boosting Classifier
LR	=	Logistic Regression
RFC	=	Random Forest Classifier
CV	=	Count Vectorizer
FIG	=	Figure

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

Fake News has become one of the major problems in the existing society. Fake News has high potential to change opinions, facts and can be the most dangerous weapon in influencing society.

The proposed project uses NLP techniques for detecting the 'fake news', that is, misleading news stories which come from non-reputable sources. By building a model based on a K-Means clustering algorithm, the fake news can be detected. The data science community has responded by taking actions against the problem. It is impossible to determine whether the news was real or fake accurately. So, the proposed project uses the datasets that are trained using the count vectorizer method for the detection of fake news and its accuracy will be tested using machine learning algorithms.

In this research, we concentrate on how to spot fake news in internet news sources. We are dedicated in two ways. In order to determine the percentage of correct news that is phony, we will use multiple datasets of actual and fake news. We provide a thorough description of the selection, justification, and approval process as well as a few exploratory analyses on the observable evidence of etymological differences in false and legitimate news material. In order to create precise false news identifiers, we focus a lot of learning studies. Additionally, we provide close examinations of the automatic and manual evidence of bogus news. Python can be used to spot fake news posted on social media.

CHAPTER-1
INTRODUCTION

1.1) Introduction

Machine learning (ML) is the study of the statistical models and methods used by computers to do certain tasks devoid of explicit instructions and in favour of patterns and inference. As part of artificial intelligence, it is viewed. Without explicit instructions, machine learning algorithms construct a mathematical model using sample data, or "training data," in order to provide predictions or judgements. Computational statistics, which focuses on computer-aided prediction, and machine learning have a lot in common. Machine learning may benefit from the ideas, practises, and fields of application that come from the study of mathematical optimisation. s

The quantity of modifications that the data goes through is referred to as "deep learning" in this context. The credit assignment path (CAP) depth is significant, especially for deep learning systems. The series of changes that take place from input to output make up the CAP. CAPs define the possible causal connections between input and outcome. For a feed-forward neural network, the depth of the CAPs is equal to the depth of the network plus one, given that the output layer is also parameterized. Since a signal can pass through a layer more than once in recurrent neural networks, the CAP depth may be limitless.

Fake news, to put it simply, is information that is untrue. whether or whether it is a ccurate. Fake news contains verifiable erroneous information. Many significant companies, even government agencies, are working to address issues related to false news. However, given that millions of articles are produced or purged every minute in this age, they are neither responsible nor humanely feasible because they rely on manual human detection. A machine learning algorithm that creates a trustworthy automated index score or rating for the authenticity of various publications and can assess whether the news is true or misleading may provide a solution to this problem.

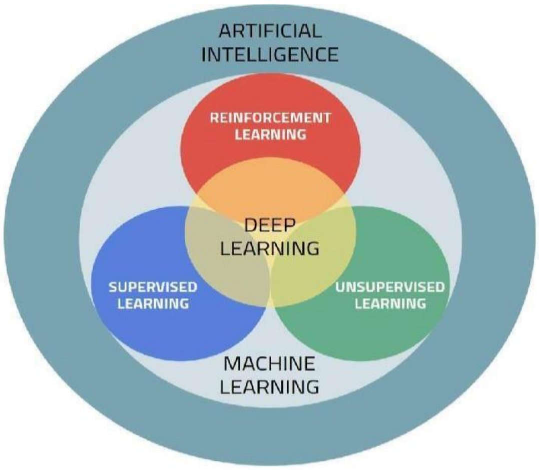


Fig. 1: Deep Learning vs Machine Learning vs Artificial Intelligence

1.1.1) Natural Language Processing (NLP)

The study of how computers interact with human (natural) languages is known as natural language processing, or NLP, and it is a branch of computer science and artificial intelligence that focuses on instructing computers to efficiently analyse massive volumes of natural language data. In the fields of linguistics, computer science, information engineering, and artificial intelligence, natural language processing (NLP) studies how computers interact with human (natural) languages. Its major goal is to instruct computer programmers in how to study and analyse vast amounts of natural language.

1.1.2) Fake News Detection

With the rising use of social media platforms, false news has become a severe problem in recent years. Finding fake news is a difficult problem that necessitates the use of several computer techniques, such as data mining, machine learning, and natural language processing. In this abstract, the current state of false news detection will be discussed, along with its challenges and potential solutions. Finally, it will consider how cutting-edge technology like blockchain and artificial intelligence may be used in the future to improve the efficiency and precision of fake news detection.

As a result, there is a larger than ever need for accurate and reliable techniques to distinguish fake news. The field of fake news detection has rapidly evolved as a result of researchers and engineers developing a number of techniques and tactics to identify and combat misleading information. These methods include human fact-checking by educated professionals as well as sophisticated computers that use machine learning to examine and classify news content. Automated processes are also a part of them.

It is important to research and create fake news detection, but it is also a challenging and complex problem. The ability to recognise fake news requires knowledge of linguistic nuance, social and cultural contexts, and the complex network dynamics of online communication. Despite these challenges, work has been done to establish effective methods for spotting false news, and the area is still developing as new tools and technology are created.

1.2) Problem Statement

Both benefits and drawbacks come with reading the news. On the other hand, news is actively sought for and consumed since it is easily available, inexpensive, and quickly spread. It makes it possible for "fake news," or negative news with blatantly inaccurate material, to be widely disseminated.

As a result, research into the detection of bogus news has recently made significant strides. First off, identifying fake news just on the basis of the content is challenging and nontrivial since it is purposefully designed to lead people to accept incorrect information.

1.3) Objective

Our project's primary goal is to determine the veracity of news in order to determine if it is real or phoney. the development of a machine learning model that would allow us to recognise bogus information.

It can be difficult and difficult to identify fake news only based on its content since it is intentionally produced to influence readers to believe false information.

By applying a range of methods and models, machine learning makes it easy to detect bogus news. Additionally, to examine the relationship between two words, we will apply deep learning-based NLP.

You may eliminate stop words using this method as well.

1.4) Methodology

1.4.1) Dataset

Two datasets are available. a mix of the two. There are 44898 news stories total in the csv file, which is a sizable quantity. While the true dataset only comprises 21417, the fraudulent dataset has 23481. This data collection is accessible at:

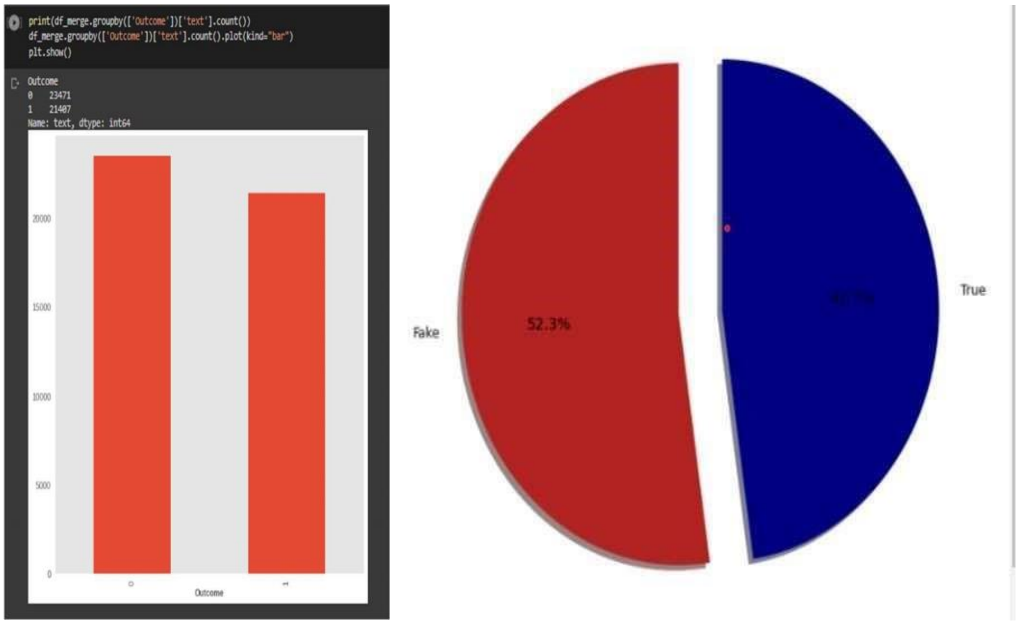
The dataset contains the following attributes:
The following elements are included in a news article: • Id: Special ID for News Article;

- title;
- text;
- Subject;
- It describes the topic of the news.
- Date: It provides news's publication date.
- The conclusion that the information might not be trustworthy.

0: Untrustworthy or False News
1: Reliable or Accurate News

First of all, the dataset is quite balanced, as we have shown. There are 21417 accurate news items and 23481 false news pieces in it. This is a beneficial feature of the dataset.

It will aid models in making objective judgments.



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

```
Outcome
0    23471
1    21407
Name: text, dtype: int64
```

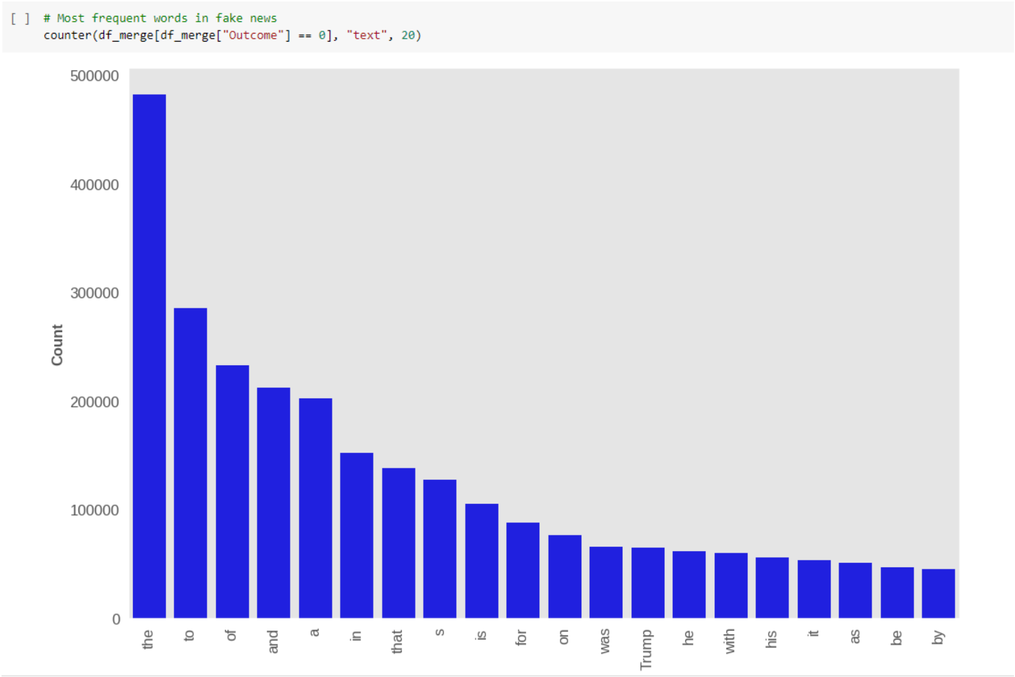
Fig. 2: Comparison of Fake and Real news

The dataset has undergone some processing, and as was indicated, stop terms have been included. The most common words in the dataset are "the," "to," "of," "and," etc.

The top 20 terms in the sample were as follows before stop words were eliminated:

.

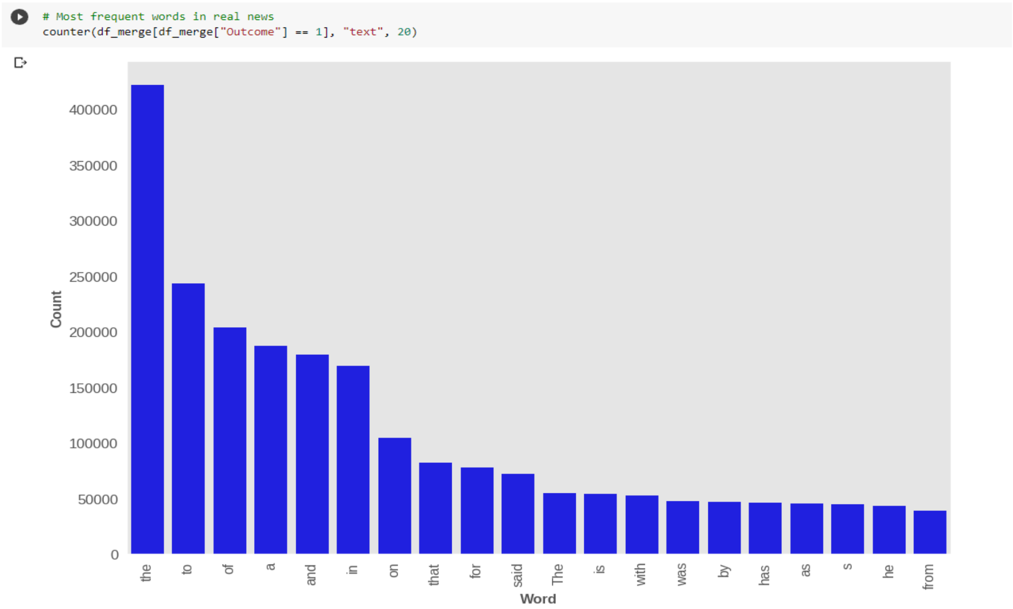
Fake.csv



```
# Most frequent words in fake news
counter(df_merge[df_merge["Outcome"] == 0], "text", 20)
```

Graph 1: Frequent words in Fake news

True.csv



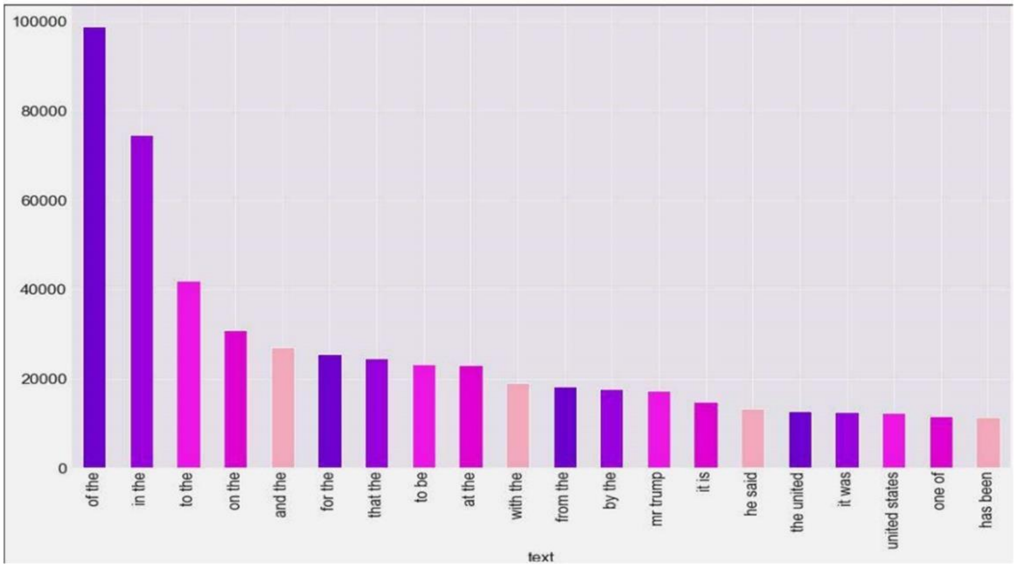
```
# Most frequent words in real news
counter(df_merge[df_merge["Outcome"] == 1], "text", 20)
```

Graph 2: Frequent words in Real News

The terms "said," "mr," "trump," "new," "people," and "year," which are now the most popular ones, can provide the models important information.

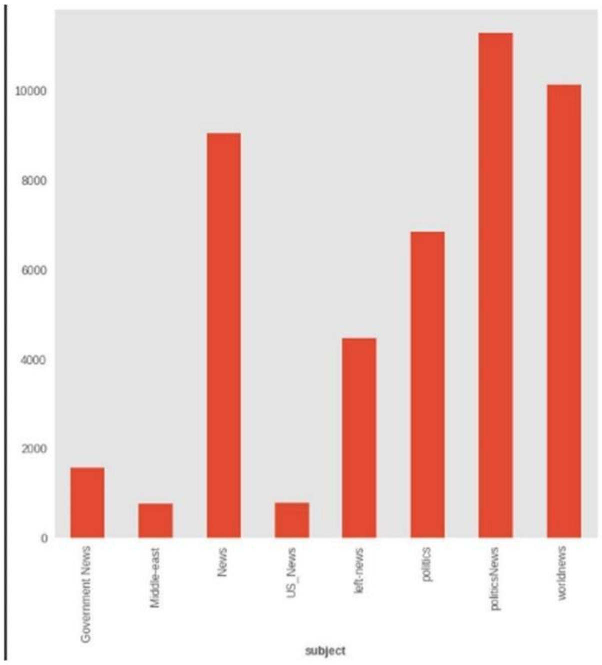
We also examined the bigrams in the dataset to have a better understanding of the news story subjects. Before stop words are removed, the topics of the news stories are not at all clear. As a result, removing stop words makes it simpler to comprehend the news reports' themes.

The graph below displays the top 20 bigrams from the dataset before stop words are removed. As one can see, often used phrases like "of the," "in," and "to the" do not help one comprehend the content of the story.



Graph 3: Frequent bigrams

To display the data, we plotted the frequencies of subject of the news:



Graph 4: Frequency of subject of the news

1.4.2) Flowchart:

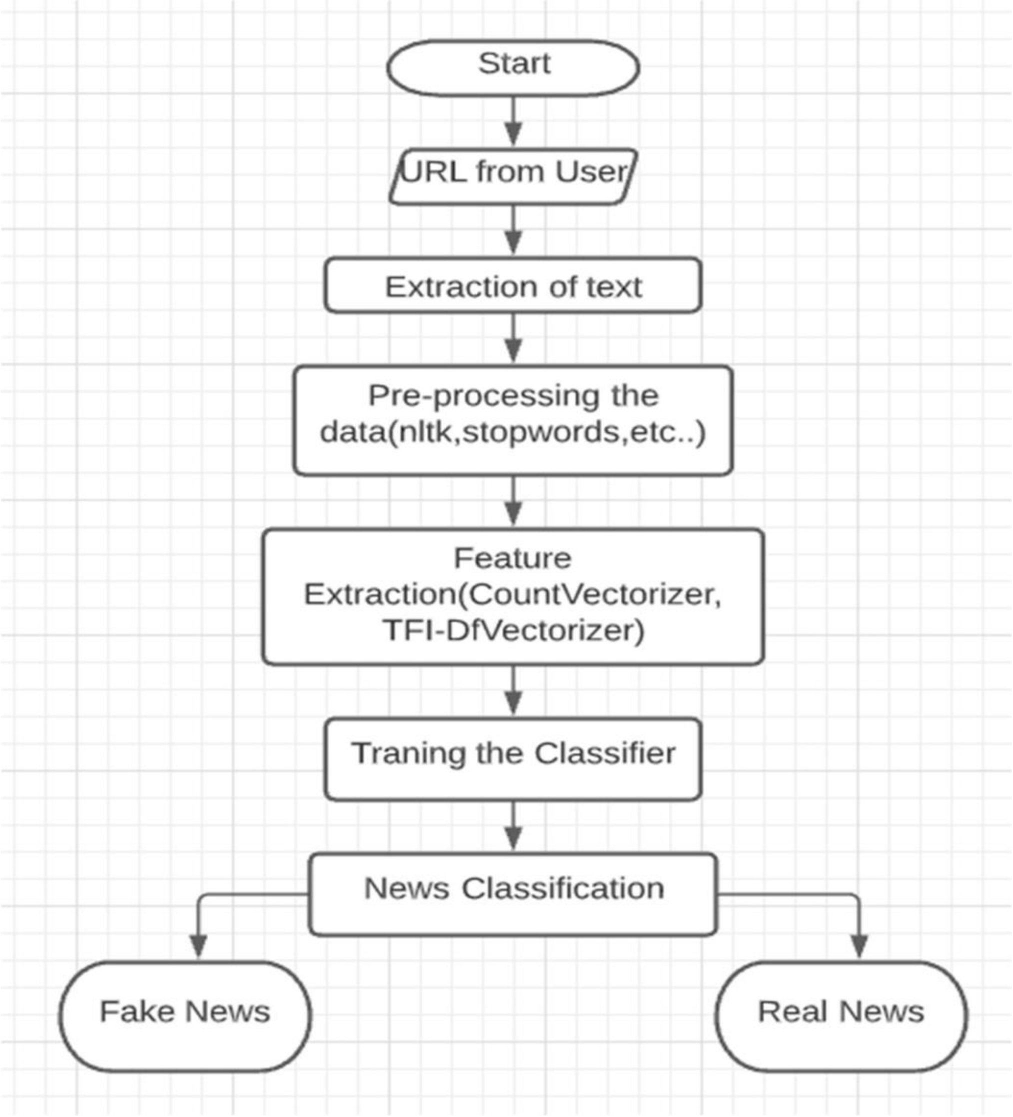

Start
URL from User
Extraction of text
Pre-processing the data(nltk,stopwords,etc..)
Feature Extraction(CountVectorizer, TFI-DfVectorizer)
Traning the Classifier
News Classification
Fake News
Real News

Fig. 3: Flowchart

1.4.3) Algorithm for The Proposed System

Step 1: Pre-processing

- Load the dataset of news items with their labels, whether they are true or false;
- Clean the text by eliminating punctuation and stopwords;
- Divide the dataset into training and testing sets.

Step 2: Count Vectorization

- Count Vectorizer from the Sklearn toolkit may be used to transform text data into numerical data.
- Produce a document-term matrix showing the frequency of each word used in each document.
- Fit the Count Vectorizer using the training set, then convert the data.
- Utilise the testing set to change the data.

Step 3: TFIDF Vectorization

- Utilise the Tfidf Vectorizer in the Sklearn package to turn the text data into numerical data.
- Use the training set to fit the Tfidf Vectorizer and convert the data.
- Create a document-term matrix that depicts the significance of each word in each document.
- Utilise the testing set to change the data.

Step 4: Training the Models

- Utilise the data that has been modified by Count Vectorizer and Tfidf Vectorizer to train a variety of models, including Naive Bayes, Logistic Regression, Support Vector Machines (SVM), Random Forest, etc.
- Fit the models using the training set.
- Use the testing set to predict the news article labels.

- Determine each model's accuracy score using the actual and projected labels.

Step 5: Confusion Matrix

- The confusion matrix displays the amount of true positives, true negatives, false positives, and false negatives for each model, allowing you to assess each one's performance.
- Measurements like accuracy, recall, and F1-score may be calculated using the confusion matrix.

Step 6: Accuracy

- Determine each model's accuracy by comparing its predicted labels to its actual labels.
- The accuracy measures the proportion of news stories that were accurately identified as being true or false.
- Evaluate the accuracy of various models to find which one is most effective at spotting fake news.

Step 7: Representing the Output in Web Browser using Streamlit

- Use the Streamlit Python module to build an interactive web application for showcasing the outcomes of false news detection models.
- Create a user interface that clearly displays the confusion matrices, accuracy of each model, and other performance indicators.
- Provide tools that allow users to submit their own content for categorization and display the key terms and phrases used to categorise news items, among other capabilities.

CHAPTER-2
LITERATURE SURVEY

A. S. A. Ahmed, A. Abidin, M. A. Maarof, and R. A. Rashid [1] is only a survey and does not contain any experiments or findings. Instead, the study offers a thorough analysis of the many false news detection techniques put out in the literature, as well as their advantages and disadvantages, as well as the datasets employed for testing. In terms of feature selection, feature extraction, classification algorithms, and assessment measures, the authors examine and contrast the methodologies utilised by various research. In the area of false news identification, they also emphasise the difficulties and potential avenues for further study. The article makes use of a number of datasets, including those from BuzzFeed, LIAR, FakeNewsNet, and PolitiFact.

S. Asghar, S. Mahmood, and H. Kamran, "Fake news detection using machine learning [2] the article also addresses a number of datasets that have been used in studies on the identification of fake news, including the LIAR dataset, the Fake News Challenge dataset, and the BuzzFeed News dataset. According to the authors, ensemble learning-based algorithms had the greatest results on the LIAR dataset, with accuracy rates of up to 78%. On the BuzzFeed News dataset, on the other hand, deep learning-based methods perform better, achieving an accuracy of up to 91%.

J. H. Kim, S. H. Lee, and H. J. Kim, "Fake news detection using ensemble learning with context and attention mechanism,"[3] For their experiments, the authors employ two datasets: the Celebrity dataset and the LIAR dataset. To capture both local and global aspects of news items, the proposed model combines convolutional neural networks (CNNs) with recurrent neural networks (RNNs). The experimental findings demonstrate that the suggested model outperforms numerous baseline models and reaches an accuracy of up to 73.7%, reaching state-of-the-art performance on both the LIAR and Celebrity datasets.

] M. F. Hossain, M. M. Islam, M. A. H. Khan, and J. J. Jung, "Fake news detection using hybrid machine learning algorithms," [4] the LIAR dataset, a gold standard for research on fake news identification, is used by the authors. It consists of statements that are either labelled as true or false and also include extra labels for the degree of falsehood. The Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), and Random Forest (RF) machine learning techniques are used in the suggested hybrid method. To choose the most pertinent characteristics for each algorithm, the authors employ a feature selection technique known as Chi-Square. They then integrate the results of the three algorithms and arrive at a final forecast using a weighted voting system. According to the experimental findings, the suggested hybrid technique works better than each individual algorithm and a number of baseline models, obtaining an accuracy of up to 72.28%.

S. S. Ghosh, A. Mukherjee, and N. Ganguly, "A multi-perspective approach to fake news detection," [5] in their research, the authors used the FakeNewsNet dataset together with another one. Word embedding and term frequency-inverse document frequency (TF-IDF) approaches are used by the authors to extract aspects from the news articles that are content-based. In order to determine the veracity of the news stories based on these attributes, the authors utilise a support vector machine (SVM) classifier. The experimental findings demonstrate that the proposed multi-perspective strategy outperforms numerous baseline models and achieves state-of-the-art performance on the FakeNewsNet and BuzzfeedNews datasets, attaining an accuracy of up to 94.7%.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1) System Configuration

Run this project using standard hardware. We utilised an Intel I5 CPU with 8 GB of RAM, a 2 GB Nvidia graphics processor, and 2 cores that have a frequency of 1.7 GHz and 2.1 GHz, respectively, to complete the project. The test phase, which follows the training phase and lasts for around 10-15 minutes, allows for predictions to be made and accuracy to be determined quickly.

3.2) Data Pre-processing

Data Missing Imputation

Missing values in datasets can be a difficulty for some machine learning techniques. Therefore, any missing values in each column of the input data must be found and replaced before we model the prediction problem. Missing The use of data assignment or assignment is made for this.A space (' ') should be used in place of the null value for each attribute. Use this approach instead of removing tuples containing null values.

Removal of Stop Words

Stop words like "if," "the," "is," "a," and "an," among others, shouldn't be given much weight by a machine learning model because they are common English expressions and don't increase the novelty or believability of any tale. Being present in the dataset may have an impact on the model's forecast because they are often used.

Removal of Special Characters

The use of special characters in a sentence has no bearing on whether a piece of news is accurate or not. We do this to eliminate all punctuation from the dataset. Regular expressions are used to eliminate all punctuation. A random function was developed to remove special characters, links, extraspace, underlines, etc.

Lemmatization

The word "play" serves as the origin for other words, including "playing" and "plays." It is possible to carry out a more extensive examination of the term's frequency by swapping out the term's core word with words in other tenses and participles. As a result, we substitute that word for any phrase that only has one source word.

Count Vectorization

For machine learning algorithms to accept the preprocessed text as input, it must next be encoded as integers or floating-point values. The phrase used to describe this method is feature extraction (or vectorization).

If a vocabulary word is present in the text data, we will add one to the corresponding vector's dimension, which will have the same number of dimensions as our vocabulary. We will add one to the total for each additional instance of that term, leaving zeros in the spots where we didn't see it even once.

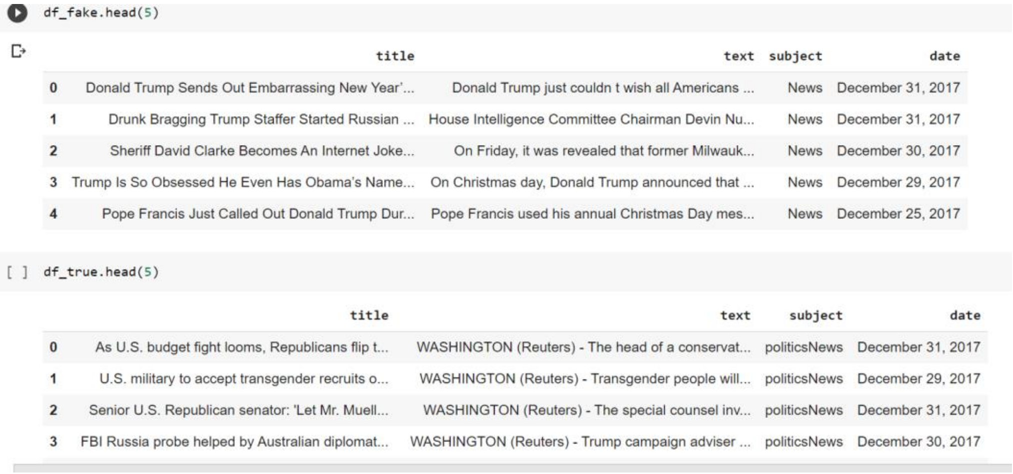
TF-IDF Transformation

In order to create a matrix with TF-IDF values for each feature, we utilise the count vectorized matrix as a transformation.
IDF, or Inverse Document Frequency, or Term Frequency (TF), which is identical to what we previously saw in the Count Vectorizer

Because some words may prove to be incredibly unimportant, word frequency alone might not be accurate. Thus, we employ TF-IDF to maintain harmony between a word's significance and frequency within the text. The acronym TF-IDF stands for term frequency and inverse document frequency.

.

Fake.csv and True.csv



```
df_fake.head(5)
```

	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

```
df_true.head(5)
```

	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

Fig. 4: Fake.csv and True.csv

3.3) Design of Project

Dataset: The first step is to collect or obtain a dataset of news articles, labeled as "fake" or "real". This dataset will be used to train and evaluate the performance of different fake news detection models.

Preprocessing: The dataset must now be cleaned up by eliminating any extraneous or irrelevant data, including stop words, punctuation, and digits. Additionally, the text may need to be normalised by making all characters lowercase and eliminating any special characters or symbols.

Count Vectorizer (BOW): The Bag-of-Words (BOW) format can be used to transform textual data into numerical characteristics after preprocessing the text. This entails building a matrix where each row represents a news item and each column represents a distinct term from the dataset. The value in each cell indicates how often the term appears in the related art.

Train-Test Split: Once we have the BOW matrix, we can split the data into training and testing sets. The training set will be used to train the fake news detection model, while the testing set will be used to evaluate the model's performance on new, unseen data.

Text-to-vectors (TF-IDF): In addition to BOW, we can also express the textual data using the Term Frequency-Inverse Document Frequency (TF-IDF) representation. The frequency of the terms in each article as well as their frequency throughout the whole dataset is taken into consideration in this representation. This helps to downplay terms that are prevalent across the whole dataset and to emphasise words that are exclusive to a certain article.

Models: After obtaining the numerical features from the text data, several machine learning methods such as logistic regression, decision trees, or neural networks can be employed to train a fake news detection model. The objective of the model is to learn a function that can accurately classify news stories as either "real" or "fake" based on the derived attributes from the text.

Accuracy and Confusion Matrix: It's crucial to assess the false news detection model's performance on the testing set after we've trained it. By assessing its accuracy, precision, recall, and F1 score, we may do this. To see how many true positives, true negatives, false positives, and false negatives the model produces, we may also develop a confusion matrix.

Testing: We may use the model to categorise fresh and previously unheard news pieces as "real" or "fake" after assessing the model's performance. This entails applying the same feature extraction and preprocessing operations to the fresh data that we did during training. After that, we can apply the trained model to the cleaned-up data to provide a categorization label.

Result: Streamlit library of python is used to represent the result in web browser where user input the news and algorithm tell that the news is “Real” or “Fake”.

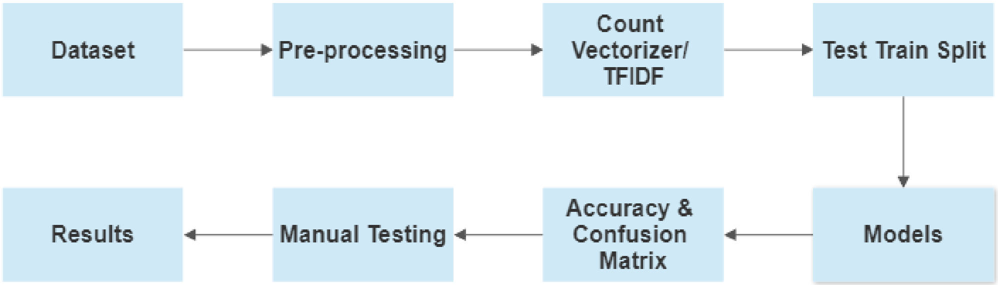
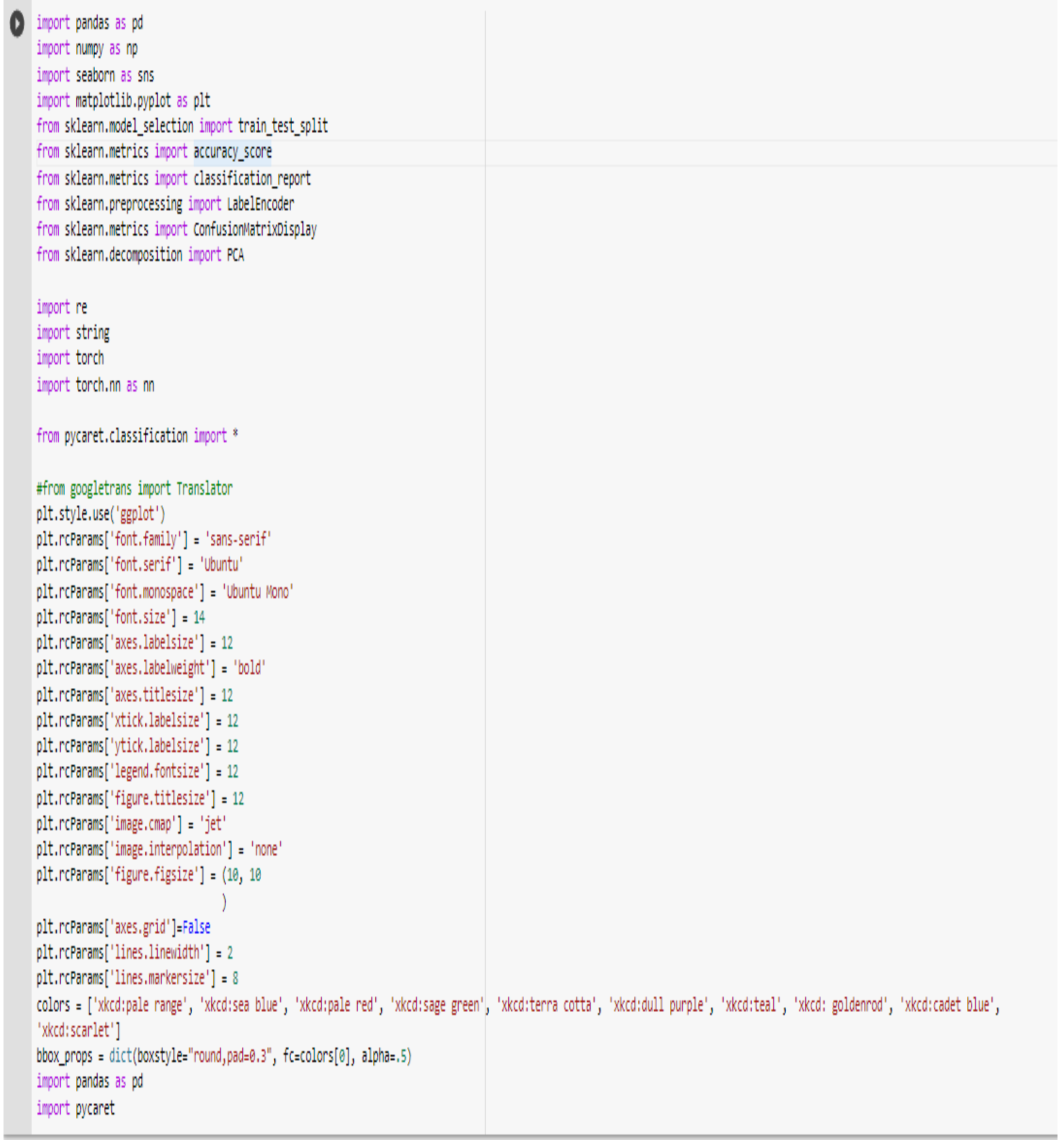


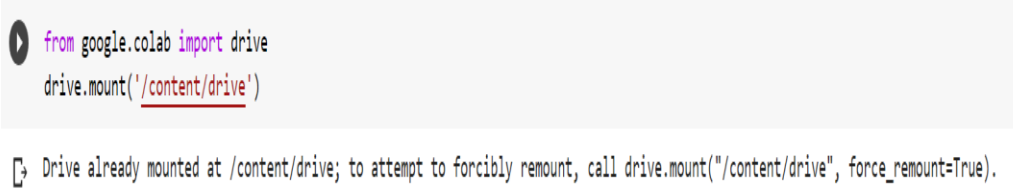
Fig 5: Design of the Projec

3.4) Sample Code



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.decomposition import PCA

import re
import string
import torch
import torch.nn as nn

from pycaret.classification import *

#from googletrans import Translator
plt.style.use('ggplot')
plt.rcParams['font.family'] = 'sans-serif'
plt.rcParams['font.serif'] = 'Ubuntu'
plt.rcParams['font.monospace'] = 'Ubuntu Mono'
plt.rcParams['font.size'] = 14
plt.rcParams['axes.labelsize'] = 12
plt.rcParams['axes.labelweight'] = 'bold'
plt.rcParams['axes.titlesize'] = 12
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
plt.rcParams['legend.fontsize'] = 12
plt.rcParams['figure.titlesize'] = 12
plt.rcParams['image.cmap'] = 'jet'
plt.rcParams['image.interpolation'] = 'none'
plt.rcParams['figure.figsize'] = (10, 10
                                 )
plt.rcParams['axes.grid']=False
plt.rcParams['lines.linewidth'] = 2
plt.rcParams['lines.markersize'] = 8
colors = ['xkcd:pale range', 'xkcd:sea blue', 'xkcd:pale red', 'xkcd:sage green', 'xkcd:terra cotta', 'xkcd:dull purple', 'xkcd:teal', 'xkcd: goldenrod', 'xkcd:cadet blue',
'xkcd:scarlet']
bbox_props = dict(boxstyle="round,pad=0.3", fc=colors[0], alpha=.5)
import pandas as pd
import pycaret
```

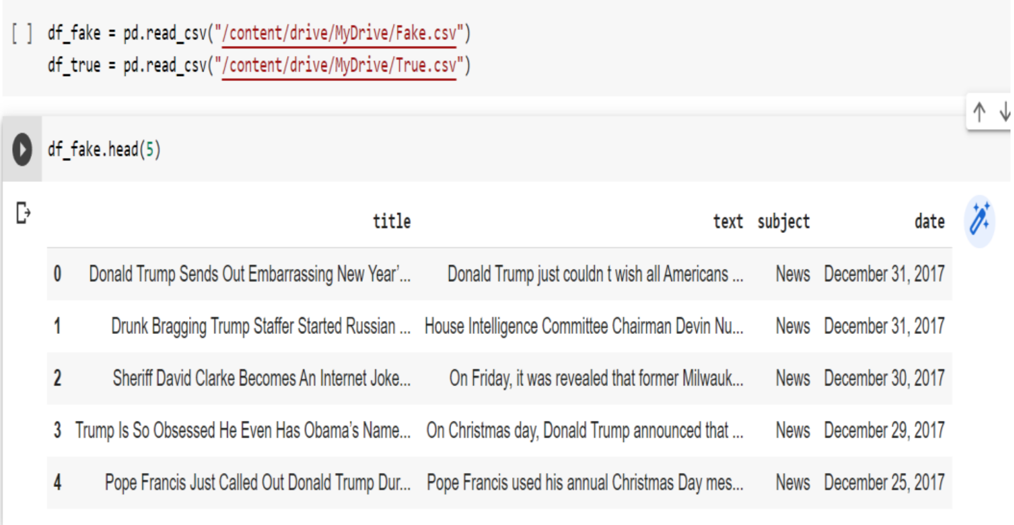
Fig. 6: Importing Libraries



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Fig. 7: Mounting Google Drive

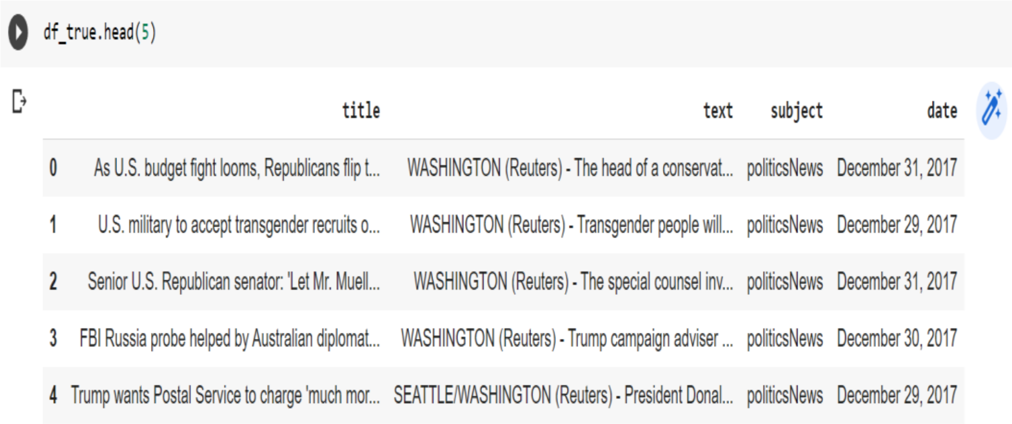


```
[ ] df_fake = pd.read_csv("/content/drive/MyDrive/Fake.csv")
    df_true = pd.read_csv("/content/drive/MyDrive/True.csv")
```

```
df_fake.head(5)
```

	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

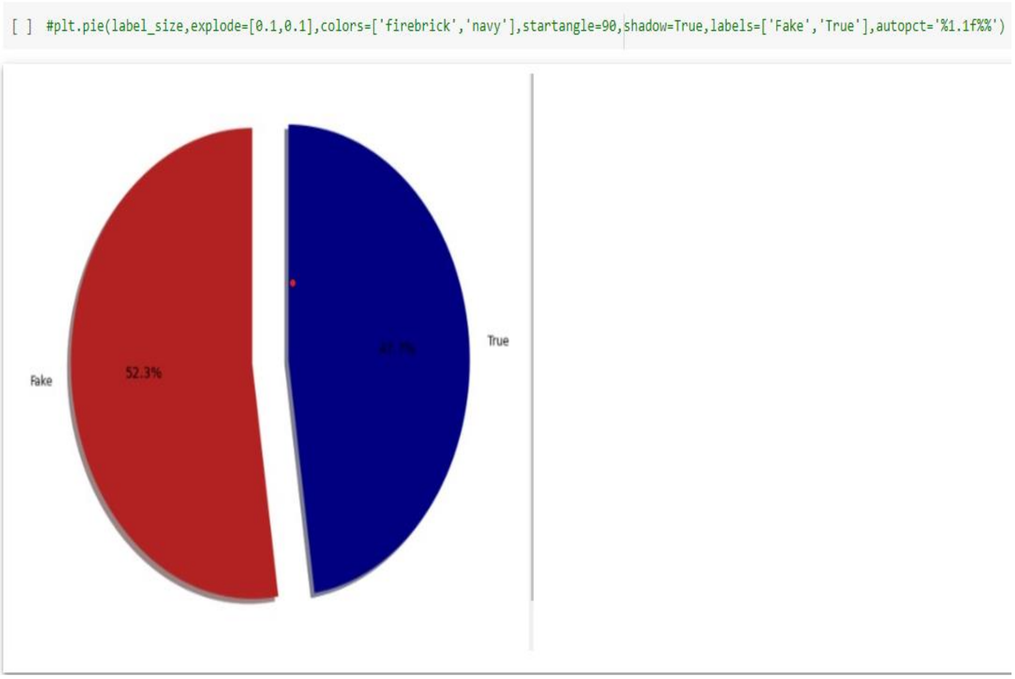
Fig. 8: Fake.csv



```
df_true.head(5)
```

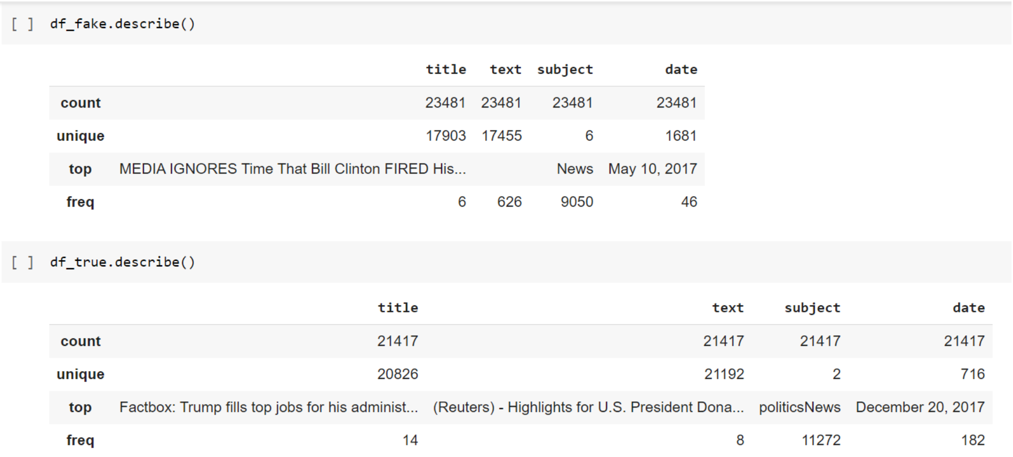
	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. 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

Fig. 9: True.csv



```
[ ] #plt.pie(label_size,explode=[0.1,0.1],colors=['firebrick','navy'],startangle=90,shadow=True,labels=['Fake','True'],autopct='%1.1f%%')
```

Fig. 10: Comparing Fake and True Dataset



```
[ ] df_fake.describe()
```

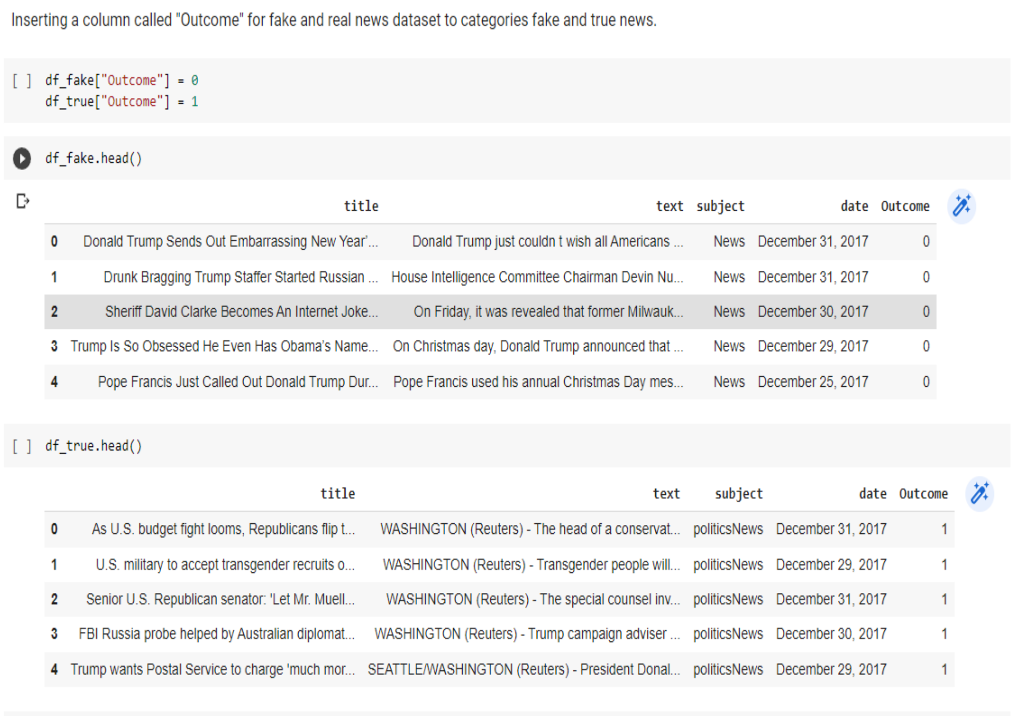
	title	text	subject	date
count	23481	23481	23481	23481
unique	17903	17455	6	1681
top	MEDIA IGNORES Time That Bill Clinton FIRED His...		News	May 10, 2017
freq	6	626	9050	46

```
[ ] df_true.describe()
```

	title	text	subject	date
count	21417	21417	21417	21417
unique	20826	21192	2	716
top	Factbox: Trump fills top jobs for his administ...	(Reuters) - Highlights for U.S. President Dona...	politicsNews	December 20, 2017
freq	14	8	11272	182

Fig. 11: Describing Fake and True Dataset

Pre-processing of Dataset



Inserting a column called "Outcome" for fake and real news dataset to categories fake and true news.

```
df_fake["Outcome"] = 0
df_true["Outcome"] = 1
```

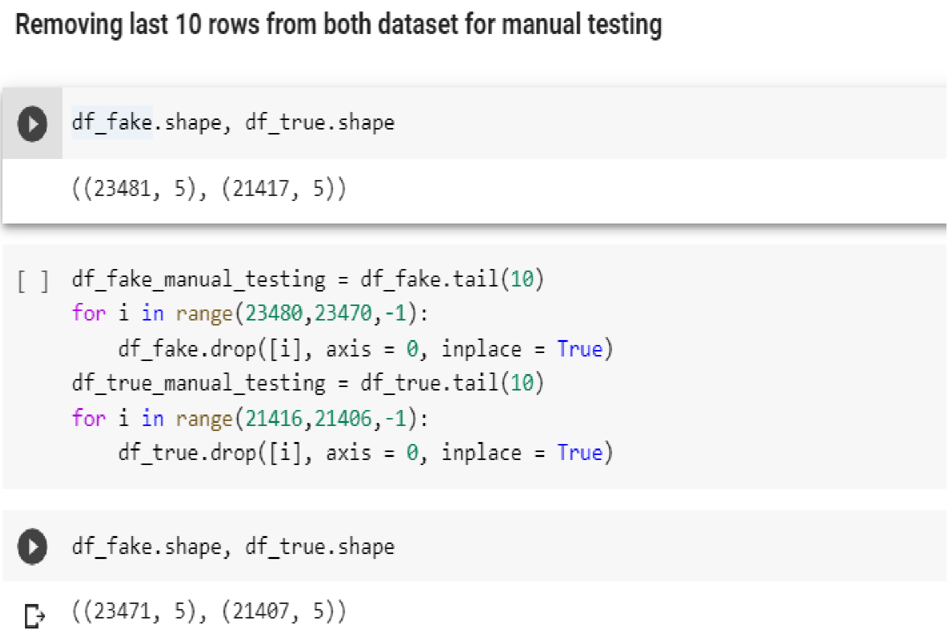
```
df_fake.head()
```

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

```
df_true.head()
```

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

Fig. 12: Inserting a column "Outcome"



Removing last 10 rows from both dataset for manual testing

```
df_fake.shape, df_true.shape
```

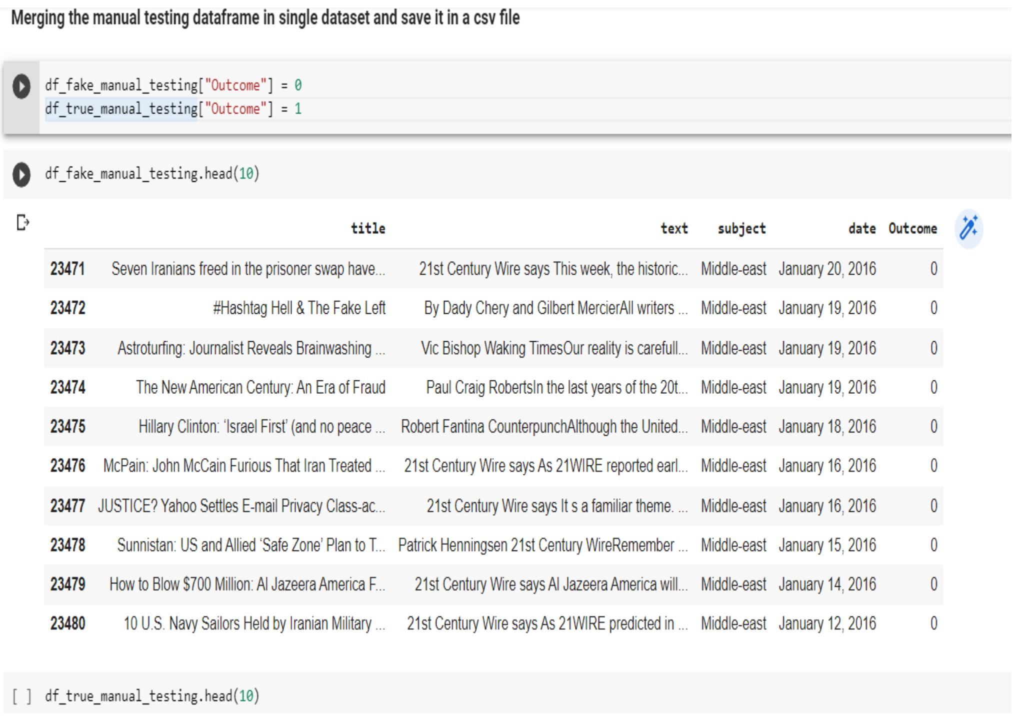
```
((23481, 5), (21417, 5))
```

```
df_fake_manual_testing = df_fake.tail(10)
for i in range(23480,23470,-1):
    df_fake.drop([i], axis = 0, inplace = True)
df_true_manual_testing = df_true.tail(10)
for i in range(21416,21406,-1):
    df_true.drop([i], axis = 0, inplace = True)
```

```
df_fake.shape, df_true.shape
```

```
((23471, 5), (21407, 5))
```

Fig. 13: Removing last 10 rows from both dataset for manual testing



Merging the manual testing dataframe in single dataset and save it in a csv file

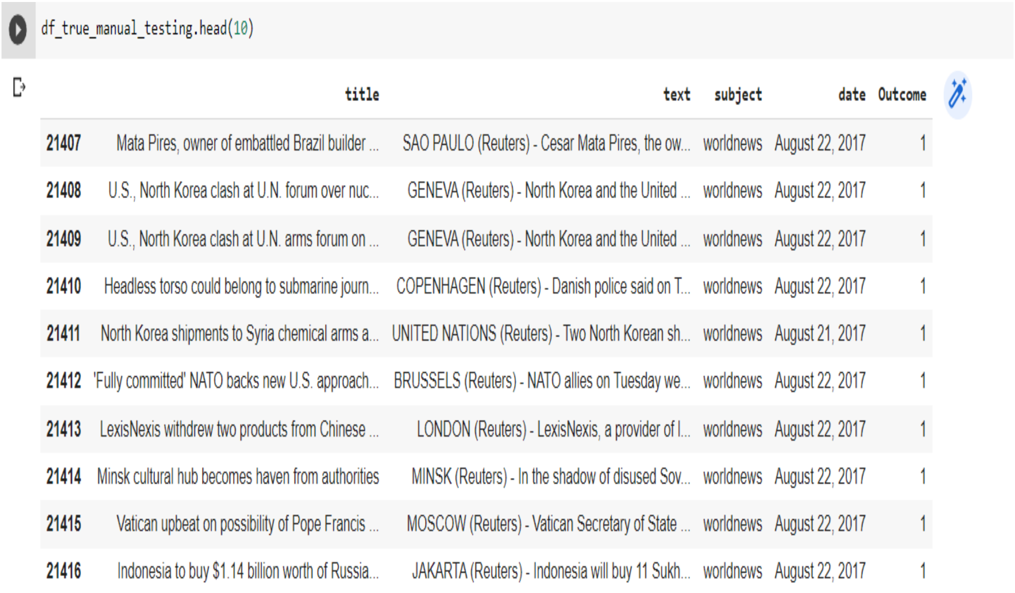
```
df_fake_manual_testing["Outcome"] = 0
df_true_manual_testing["Outcome"] = 1
```

```
df_fake_manual_testing.head(10)
```

	title	text	subject	date	Outcome
23471	Seven Iranians freed in the prisoner swap have...	21st Century Wire says This week, the historic...	Middle-east	January 20, 2016	0
23472	#Hashtag Hell & The Fake Left	By Dady Chery and Gilbert MercierAll writers ...	Middle-east	January 19, 2016	0
23473	Astroturfing: Journalist Reveals Brainwashing ...	Vic Bishop Waking TimesOur reality is carefull...	Middle-east	January 19, 2016	0
23474	The New American Century: An Era of Fraud	Paul Craig RobertsIn the last years of the 20t...	Middle-east	January 19, 2016	0
23475	Hillary Clinton: 'Israel First' (and no peace ...	Robert Fantina CounterpunchAlthough the United...	Middle-east	January 18, 2016	0
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	0
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016	0
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	0
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	0
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	0

```
[ ] df_true_manual_testing.head(10)
```

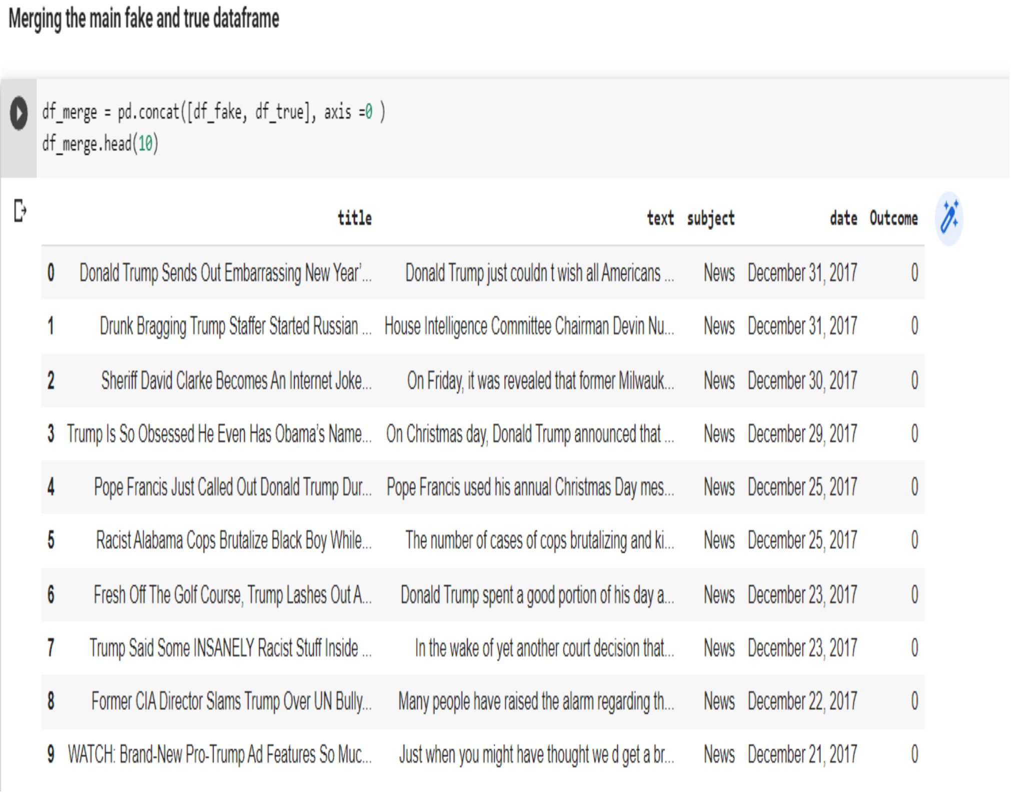
Fig. 14: Merging the manual data frame



```
df_true_manual_testing.head(10)
```

	title	text	subject	date	Outcome
21407	Mata Pires, owner of embattled Brazil builder ...	SAO PAULO (Reuters) - Cesar Mata Pires, the ow...	worldnews	August 22, 2017	1
21408	U.S., North Korea clash at U.N. forum over nuc...	GENEVA (Reuters) - North Korea and the United ...	worldnews	August 22, 2017	1
21409	U.S., North Korea clash at U.N. arms forum on ...	GENEVA (Reuters) - North Korea and the United ...	worldnews	August 22, 2017	1
21410	Headless torso could belong to submarine journ...	COPENHAGEN (Reuters) - Danish police said on T...	worldnews	August 22, 2017	1
21411	North Korea shipments to Syria chemical arms a...	UNITED NATIONS (Reuters) - Two North Korean sh...	worldnews	August 21, 2017	1
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

Fig. 15: Manual_testing dataset

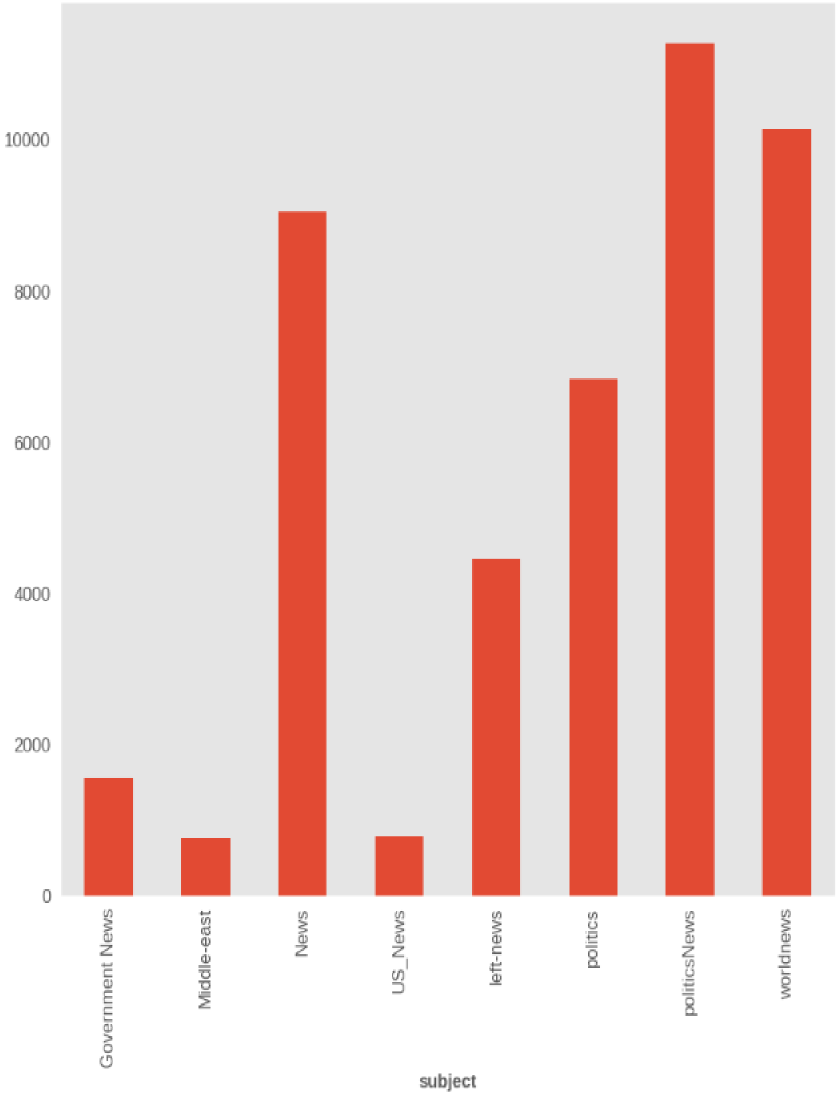


Merging the main fake and true dataframe

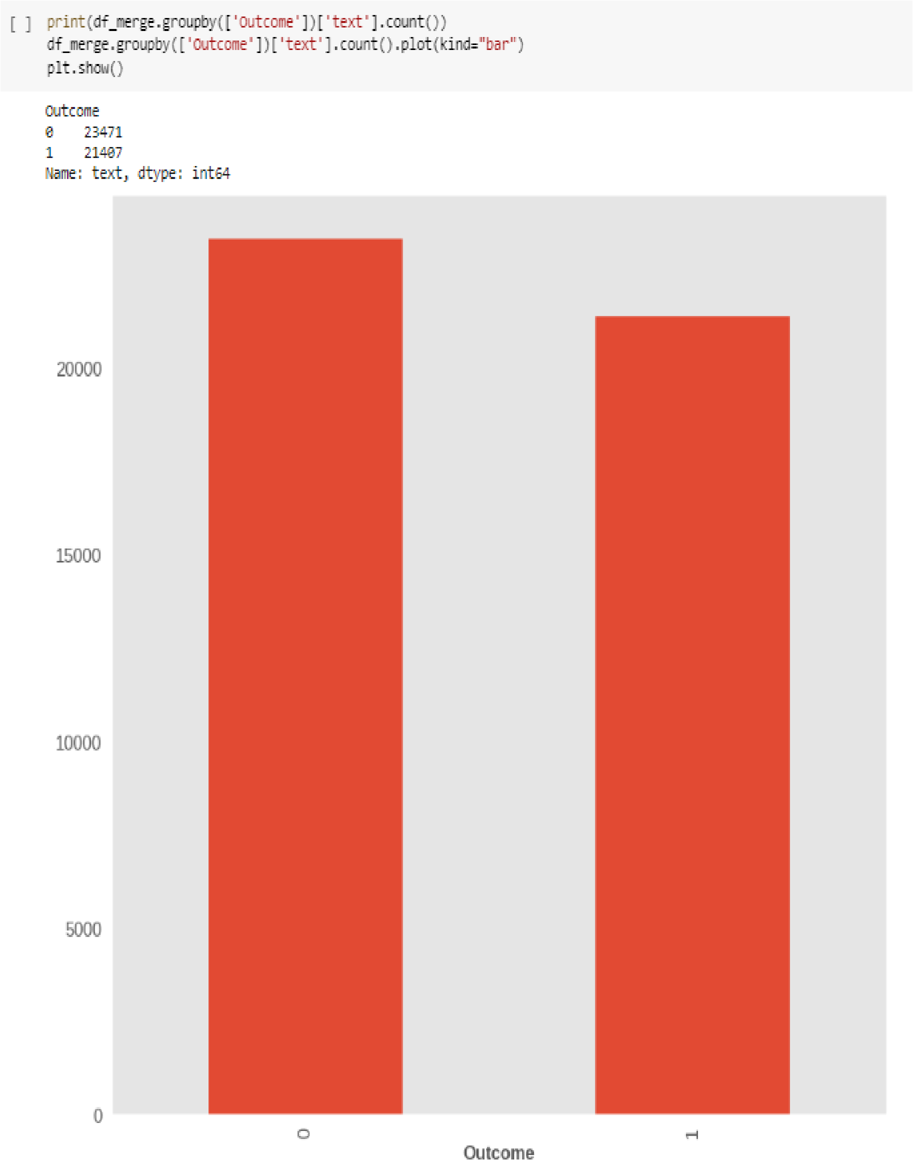
```
df_merge = pd.concat([df_fake, df_true], axis =0 )
df_merge.head(10)
```

	title	text	subject	date	Outcome
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
5	Racist Alabama Cops Brutalize Black Boy While...	The number of cases of cops brutalizing and ki...	News	December 25, 2017	0
6	Fresh Off The Golf Course, Trump Lashes Out A...	Donald Trump spent a good portion of his day a...	News	December 23, 2017	0
7	Trump Said Some INSANELY Racist Stuff Inside ...	In the wake of yet another court decision that...	News	December 23, 2017	0
8	Former CIA Director Slams Trump Over UN Bully...	Many people have raised the alarm regarding th...	News	December 22, 2017	0
9	WATCH: Brand-New Pro-Trump Ad Features So Muc...	Just when you might have thought we d get a br...	News	December 21, 2017	0

Fig. 16: Merging the main fake and true dataframe



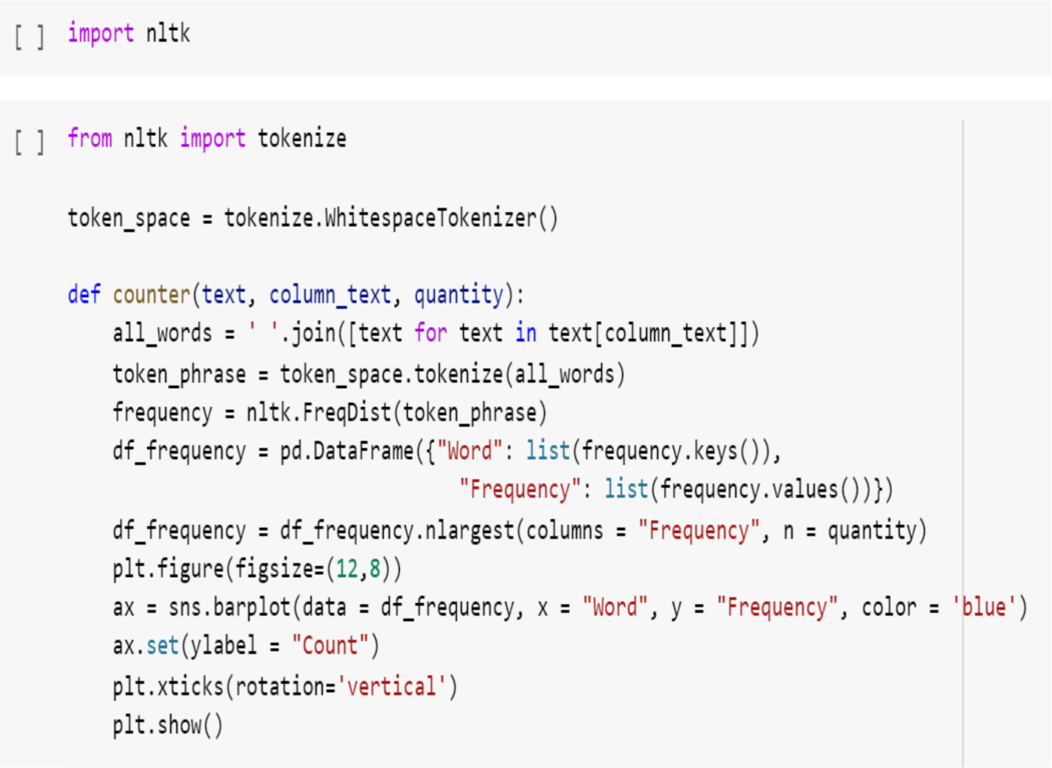
Graph 5: Frequency of subject of the news



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

```
Outcome
0    23471
1    21407
Name: text, dtype: int64
```

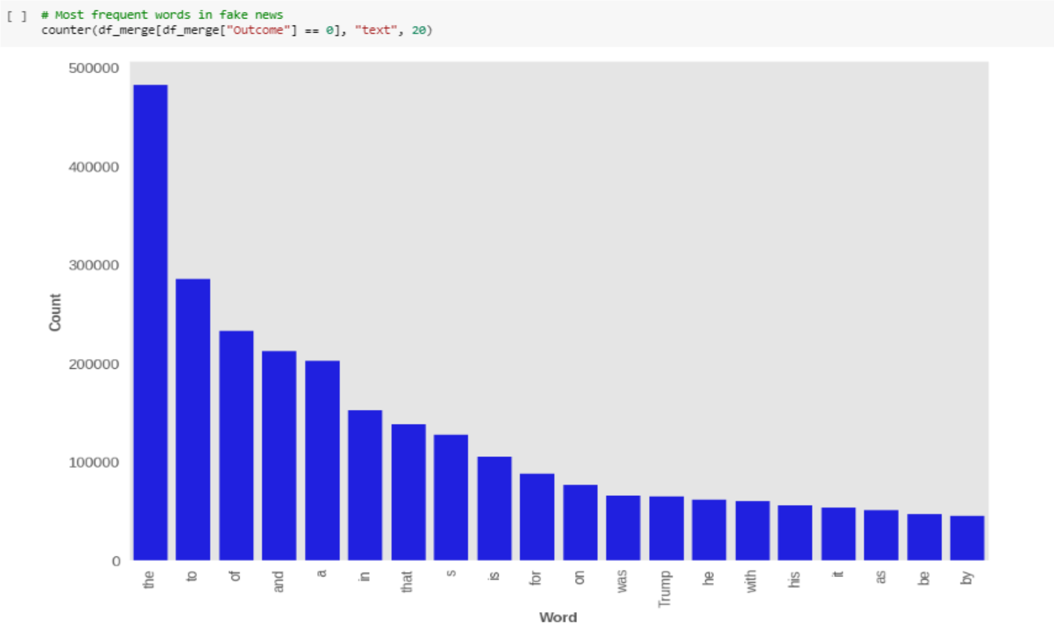
Graph 6: Fake and Real News



```
[ ]  import nltk
```

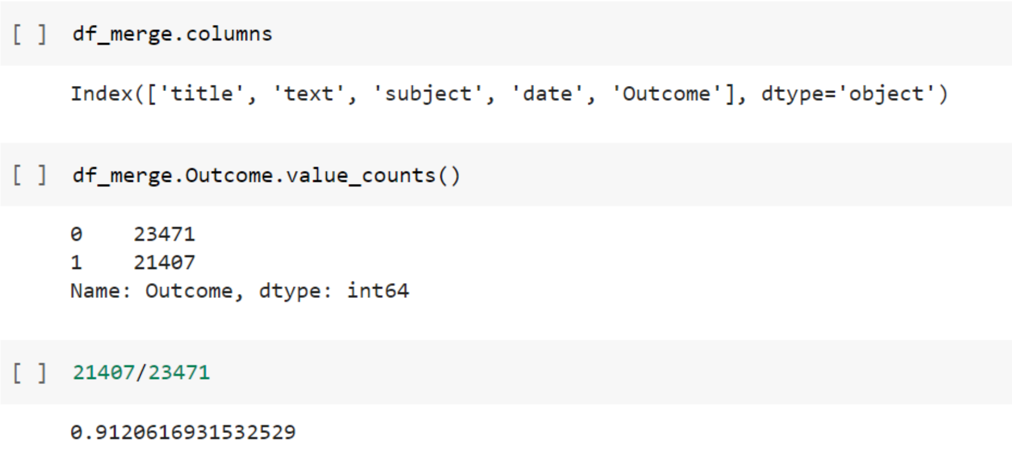
```
[ ]  from nltk import tokenize

     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=(12,8))
         ax = sns.barplot(data = df_frequency, x = "Word", y = "Frequency", color = 'blue')
         ax.set(ylabel = "Count")
         plt.xticks(rotation='vertical')
         plt.show()
```

Fig. 17: WhitespaceTokenizer



```
[ ]  # Most frequent words in fake news
     counter(df_merge[df_merge["Outcome"] == 0], "text", 20)
```

Graph 7: Frequency of words in fake news



```
df_merge.columns
```

```
Index(['title', 'text', 'subject', 'date', 'Outcome'], dtype='object')
```

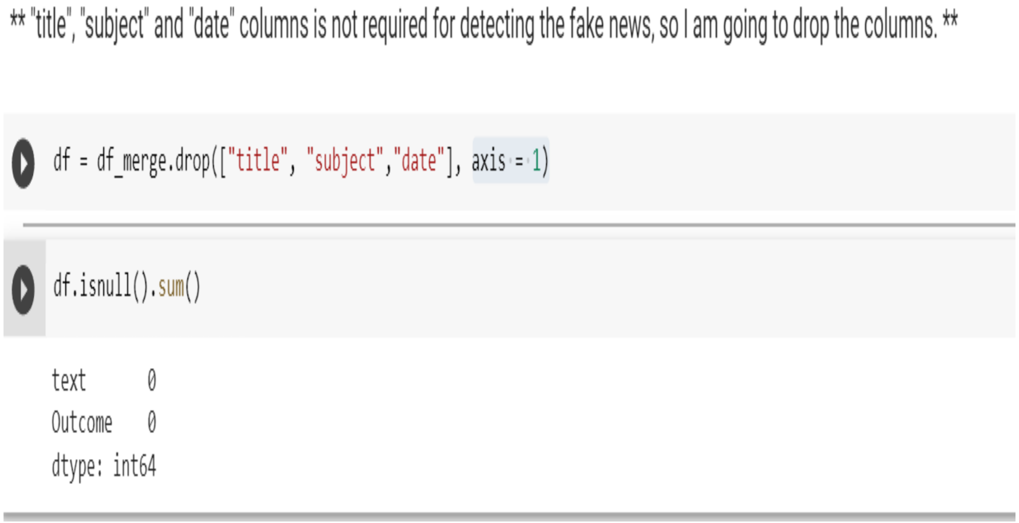
```
df_merge.Outcome.value_counts()
```

```
0    23471
1    21407
Name: Outcome, dtype: int64
```

```
21407/23471
```

```
0.9120616931532529
```

Fig. 18: Checking the columns



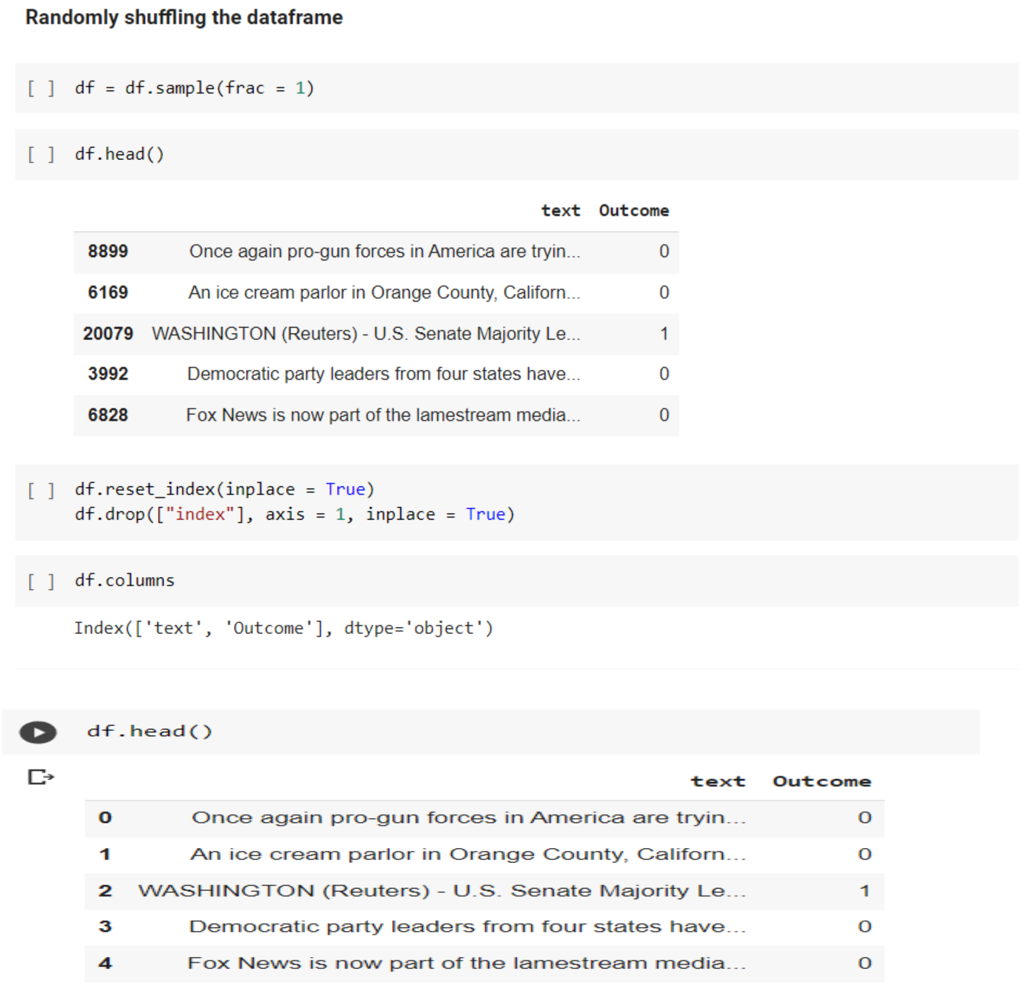
** "title", "subject" and "date" columns is not required for detecting the fake news, so I am going to drop the columns. **

```
df = df_merge.drop(["title", "subject","date"], axis = 1)
```

```
df.isnull().sum()
```

```
text       0
Outcome    0
dtype: int64
```

Fig. 19: Removing “title”, “subject” and “date” columns



Randomly shuffling the dataframe

```
df = df.sample(frac = 1)
```

```
df.head()
```

	text	Outcome
8899	Once again pro-gun forces in America are tryin...	0
6169	An ice cream parlor in Orange County, Californ...	0
20079	WASHINGTON (Reuters) - U.S. Senate Majority Le...	1
3992	Democratic party leaders from four states have...	0
6828	Fox News is now part of the lamestream media...	0

```
df.reset_index(inplace = True)
df.drop(["index"], axis = 1, inplace = True)
```

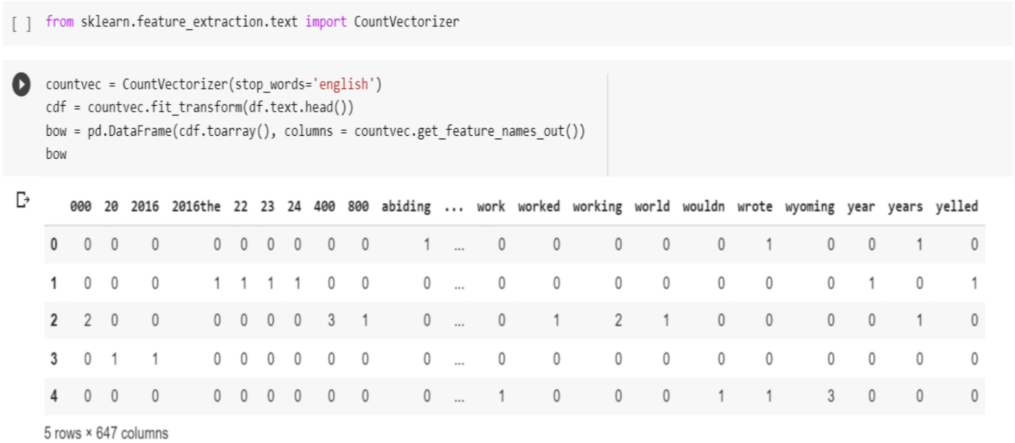
```
df.columns
```

```
Index(['text', 'Outcome'], dtype='object')
```

```
df.head()
```

	text	Outcome
0	Once again pro-gun forces in America are tryin...	0
1	An ice cream parlor in Orange County, Californ...	0
2	WASHINGTON (Reuters) - U.S. Senate Majority Le...	1
3	Democratic party leaders from four states have...	0
4	Fox News is now part of the lamestream media...	0

Fig. 20: Randomly Shuffling the data frame



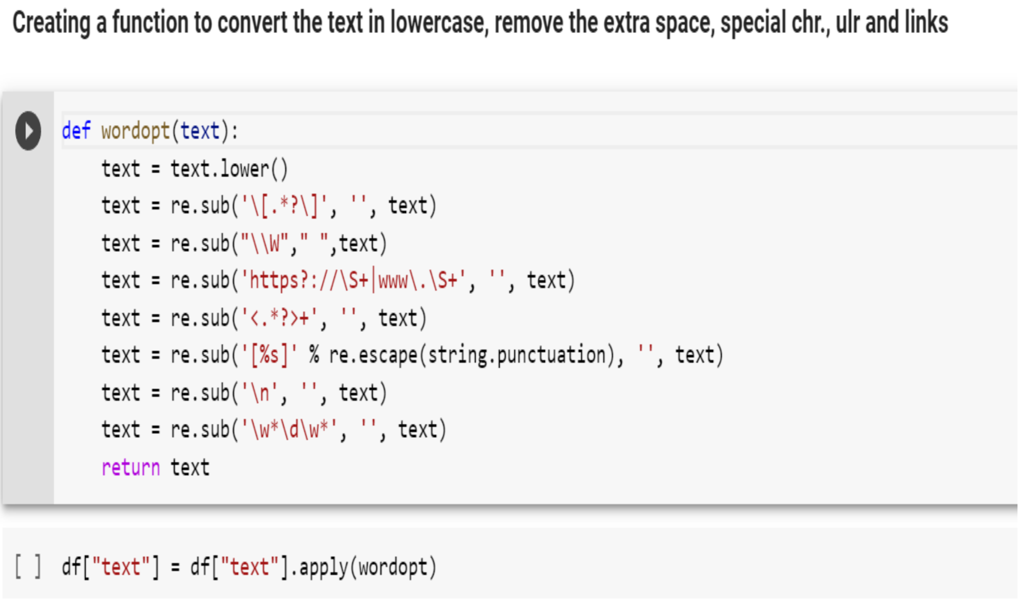
```
from sklearn.feature_extraction.text import CountVectorizer
```

```
countvec = CountVectorizer(stop_words='english')
cdf = countvec.fit_transform(df.text.head())
bow = pd.DataFrame(cdf.toarray(), columns = countvec.get_feature_names_out())
bow
```

	000	20	2016	2016the	22	23	24	400	800	abiding	...	work	worked	working	world	wouldn	wrote	wyoming	year	years	yelled
0	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	1	0	0	1	0
1	0	0	0	1	1	1	1	0	0	0	...	0	0	0	0	0	0	0	1	0	1
2	2	0	0	0	0	0	0	3	1	0	...	0	1	2	1	0	0	0	0	1	0
3	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	1	1	3	0	0	0

5 rows × 647 columns

Fig. 21: Count Vectorizer



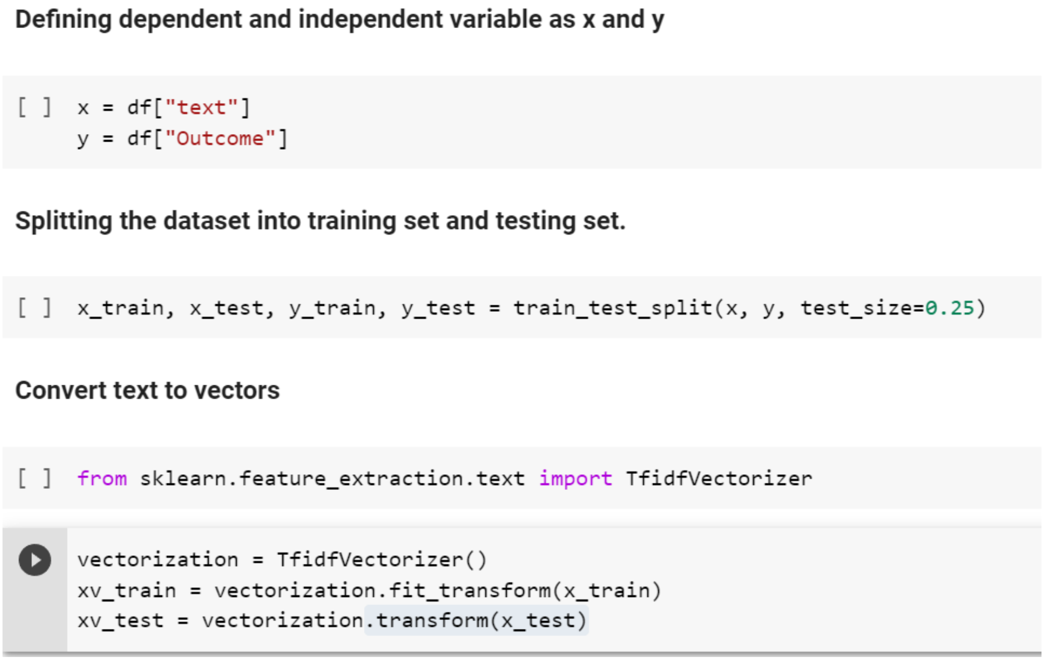
Creating a function to convert the text in lowercase, remove the extra space, special chr., ulr and links

```
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

```
[ ] df["text"] = df["text"].apply(wordopt)
```

Fig. 22: Pre-processing task of words

Train-Test Split



Defining dependent and independent variable as x and y

```
[ ] x = df["text"]
    y = df["Outcome"]
```

Splitting the dataset into training set and testing set.

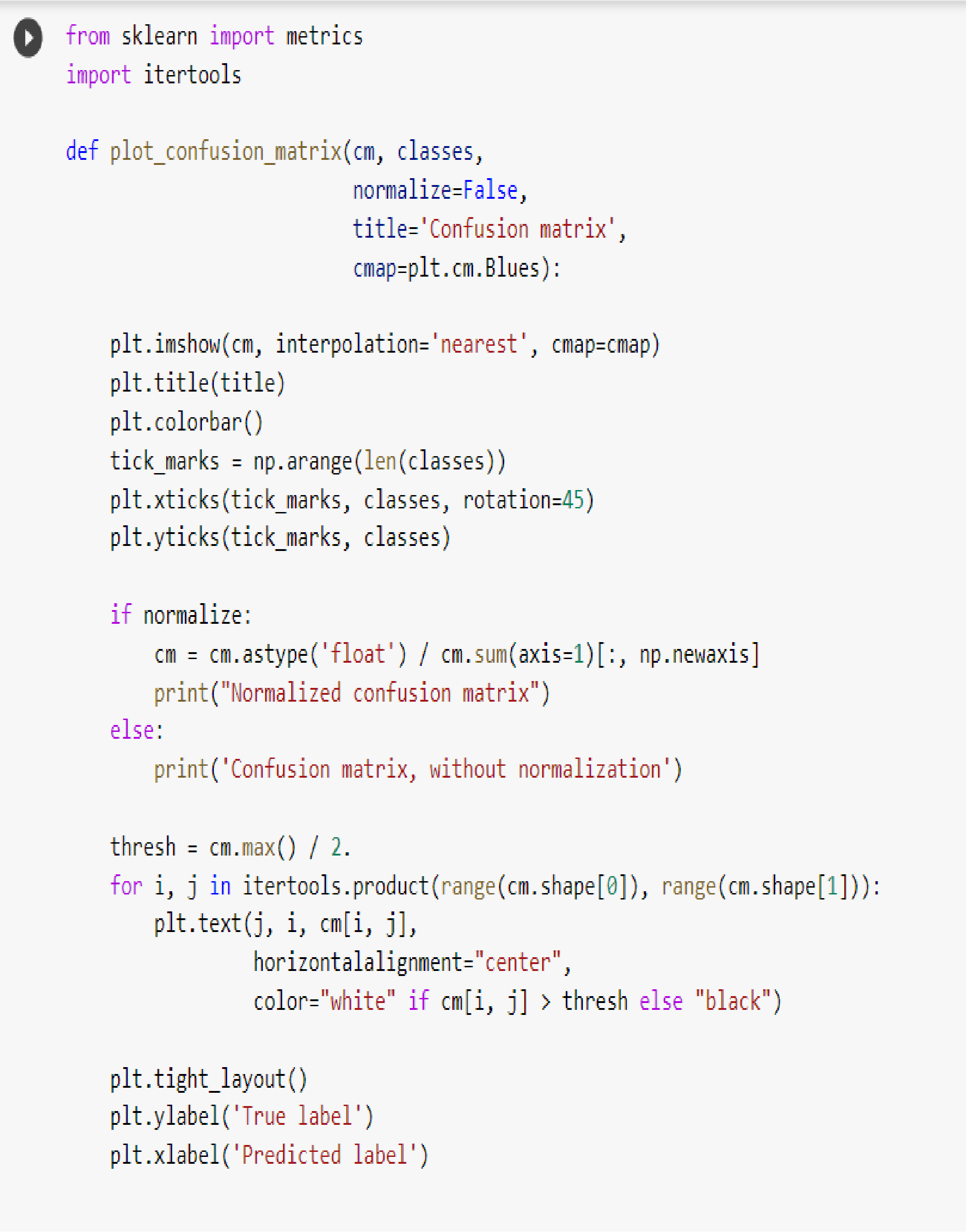
```
[ ] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

Convert text to vectors

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

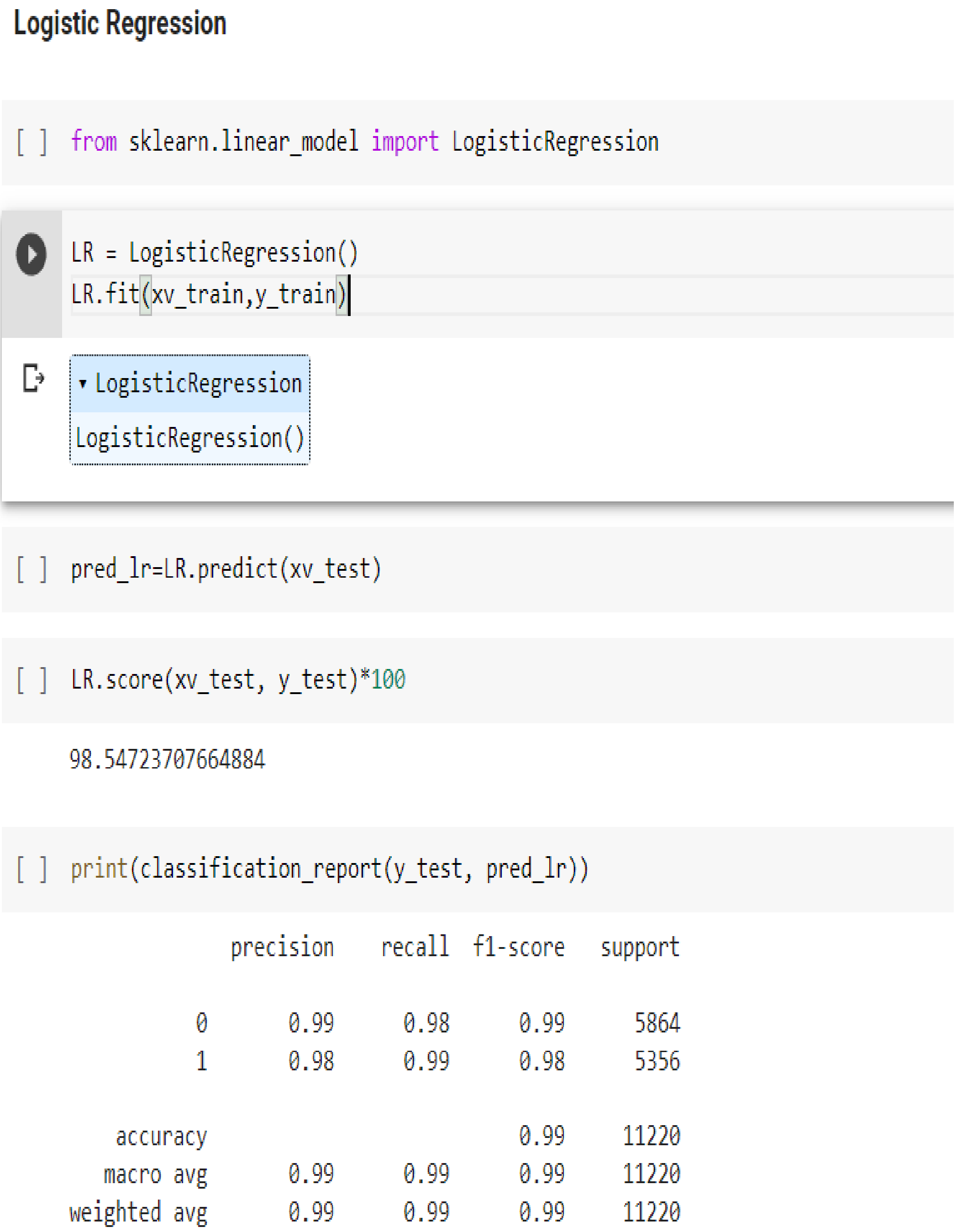
Fig. 23: Train-Test Split



```python
from sklearn import metrics
import itertools

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

Fig. 24: Importing for Confusion Matrix

Models



Logistic Regression

```
[ ] from sklearn.linear_model import LogisticRegression
```

```
LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
[ ] pred_lr=LR.predict(xv_test)
```

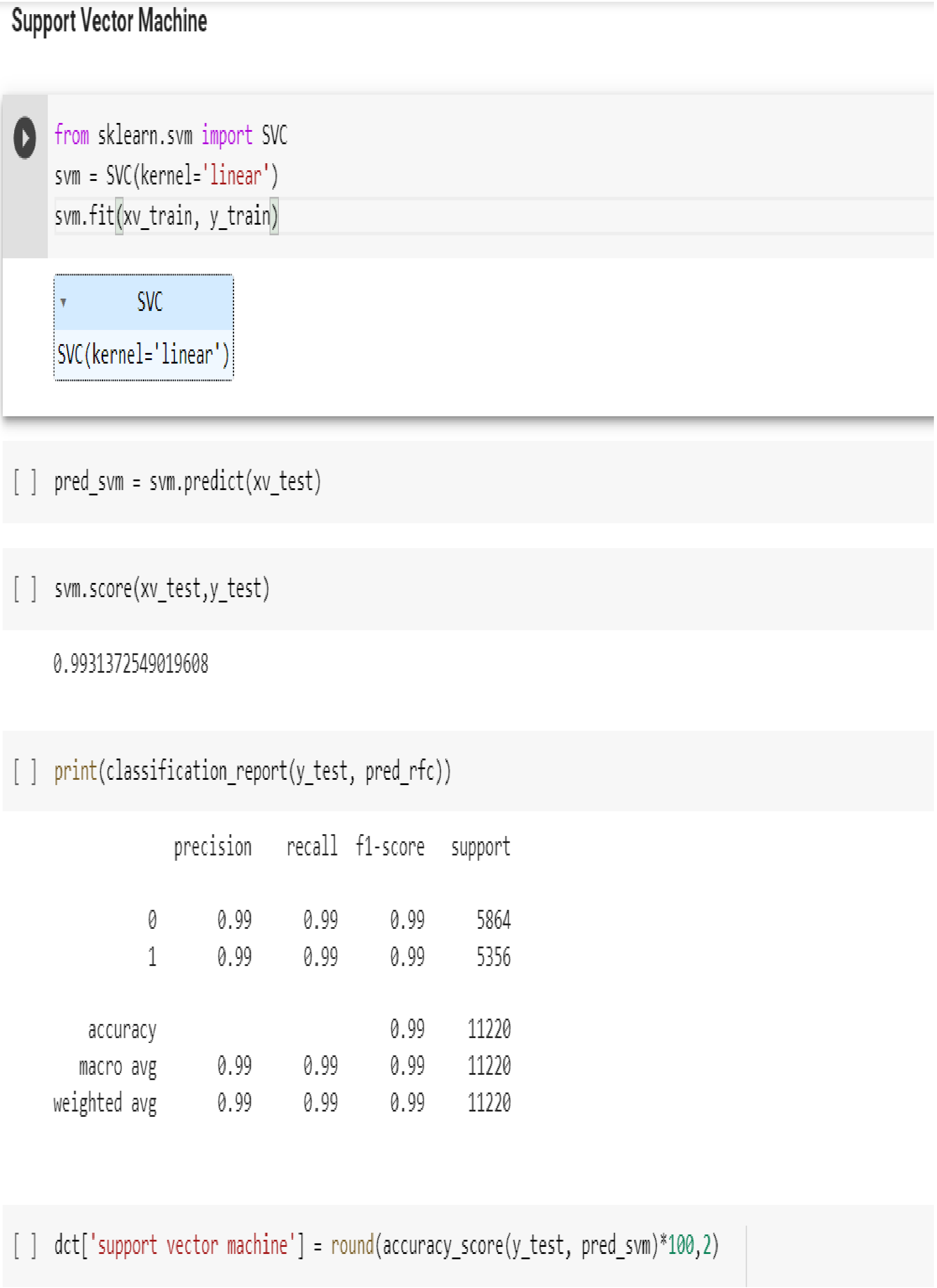
```
[ ] LR.score(xv_test, y_test)*100
```

```
98.54723707664884
```

```
[ ] print(classification_report(y_test, pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      5864
           1       0.98      0.99      0.98      5356

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

Fig. 25: Logistic Regression



Support Vector Machine

```
from sklearn.svm import SVC
svm = SVC(kernel='linear')
svm.fit(xv_train, y_train)
```

```
▾        SVC
SVC(kernel='linear')
```

```
[ ]  pred_svm = svm.predict(xv_test)
```

```
[ ]  svm.score(xv_test,y_test)
```

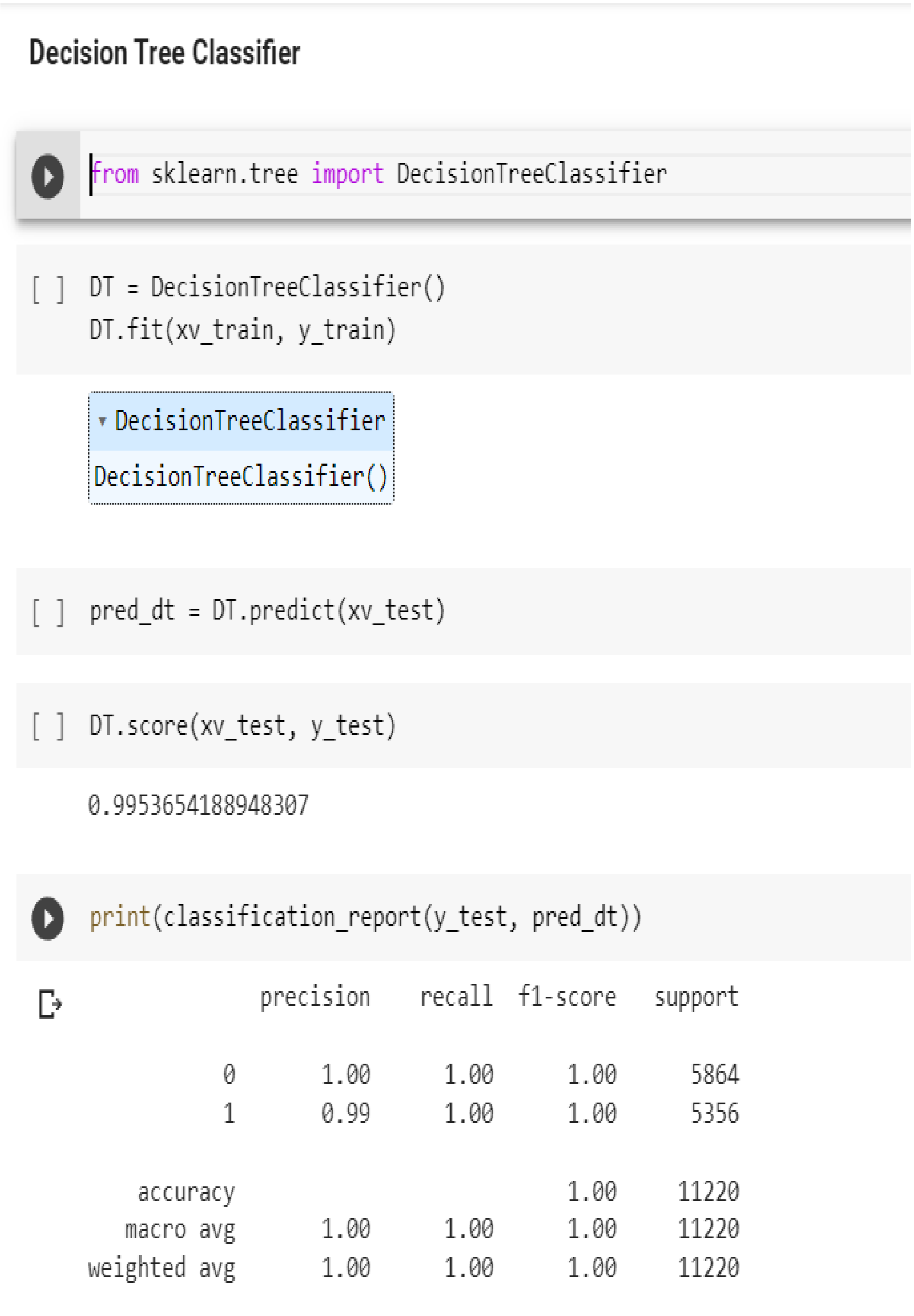
```
0.9931372549019608
```

```
[ ]  print(classification_report(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5864
           1       0.99      0.99      0.99      5356

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

```
[ ]  dct['support vector machine'] = round(accuracy_score(y_test, pred_svm)*100,2)
```

Fig. 26: Support Vector Machine



Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
```

```
DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
pred_dt = DT.predict(xv_test)
```

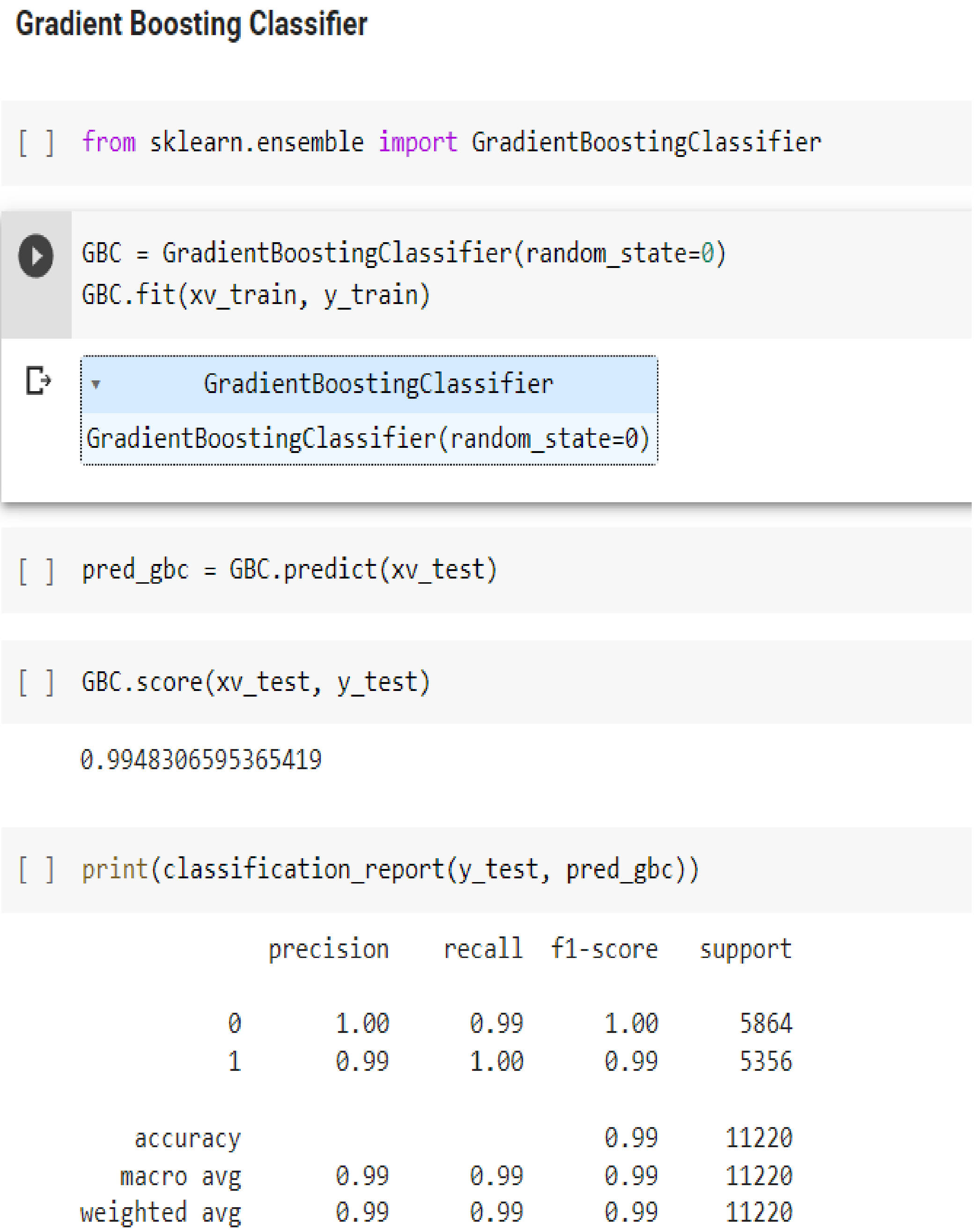
```
DT.score(xv_test, y_test)
```

```
0.9953654188948307
```

```
print(classification_report(y_test, pred_dt))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5864
           1       0.99      1.00      1.00      5356

    accuracy                           1.00     11220
   macro avg       1.00      1.00      1.00     11220
weighted avg       1.00      1.00      1.00     11220
```

Fig. 27: Decision Tree Classifier



Gradient Boosting Classifier

```
[ ] from sklearn.ensemble import GradientBoostingClassifier
```

```
GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)
```

```
▾        GradientBoostingClassifier
GradientBoostingClassifier(random_state=0)
```

```
[ ] pred_gbc = GBC.predict(xv_test)
```

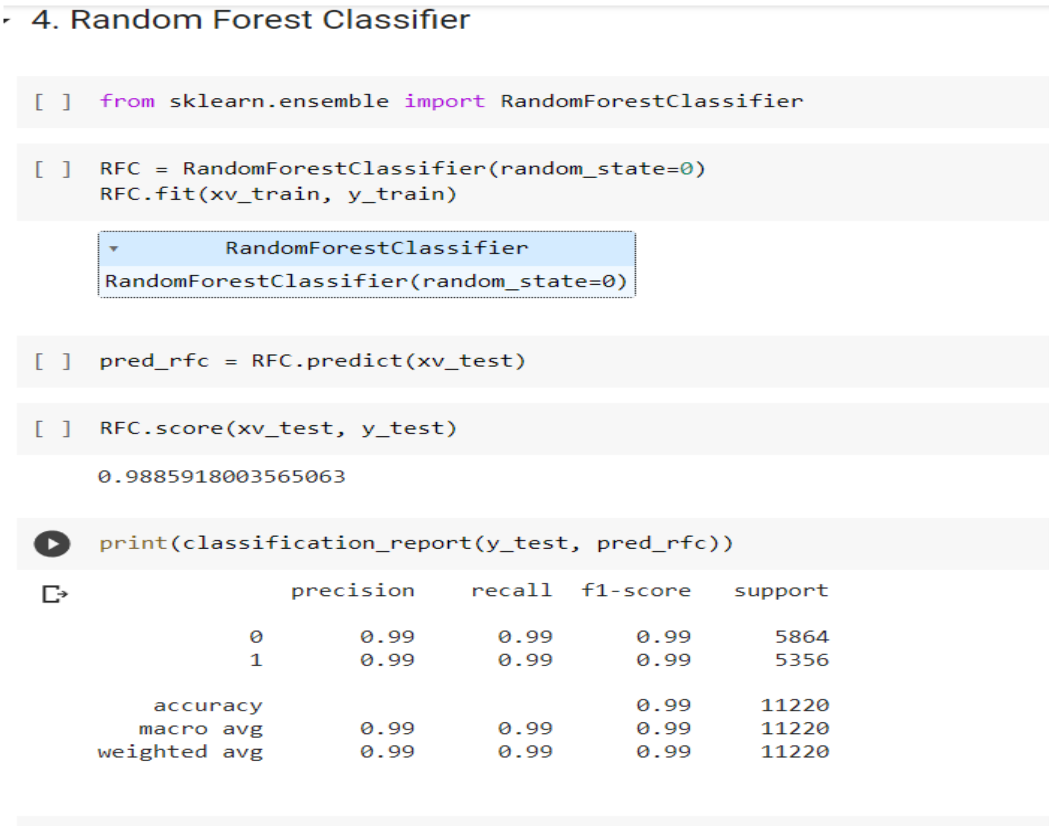
```
[ ] GBC.score(xv_test, y_test)
```

```
0.9948306595365419
```

```
[ ] print(classification_report(y_test, pred_gbc))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5864
           1       0.99      1.00      0.99      5356

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

Fig. 28: Gradient Boosting Classifier



4. Random Forest Classifier

```
[ ]  from sklearn.ensemble import RandomForestClassifier
```

```
[ ]  RFC = RandomForestClassifier(random_state=0)
     RFC.fit(xv_train, y_train)
```

```
▾      RandomForestClassifier
RandomForestClassifier(random_state=0)
```

```
[ ]  pred_rfc = RFC.predict(xv_test)
```

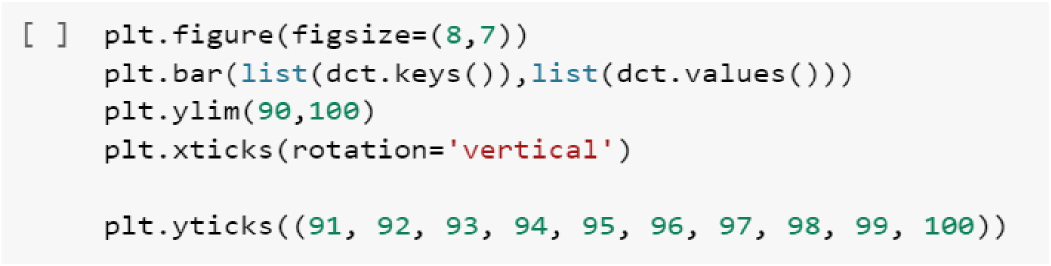
```
[ ]  RFC.score(xv_test, y_test)
```

```
0.9885918003565063
```

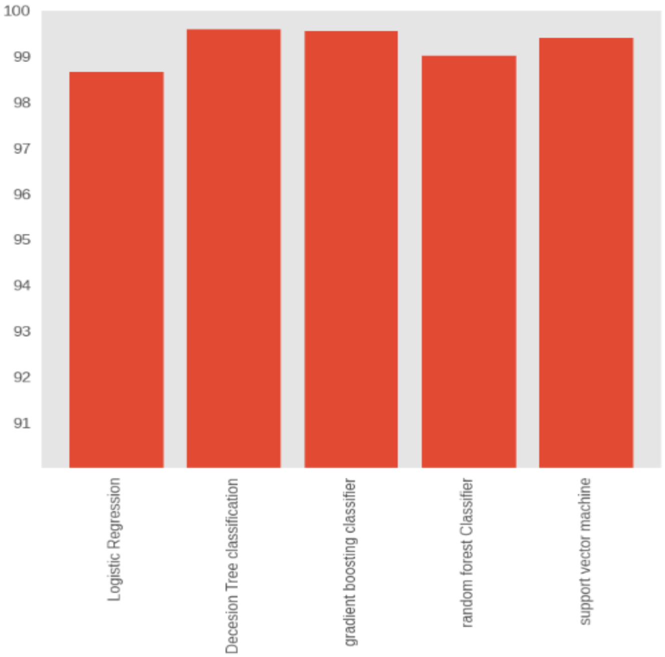
```
print(classification_report(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5864
           1       0.99      0.99      0.99      5356

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

Fig. 29: Random Forest Classifier

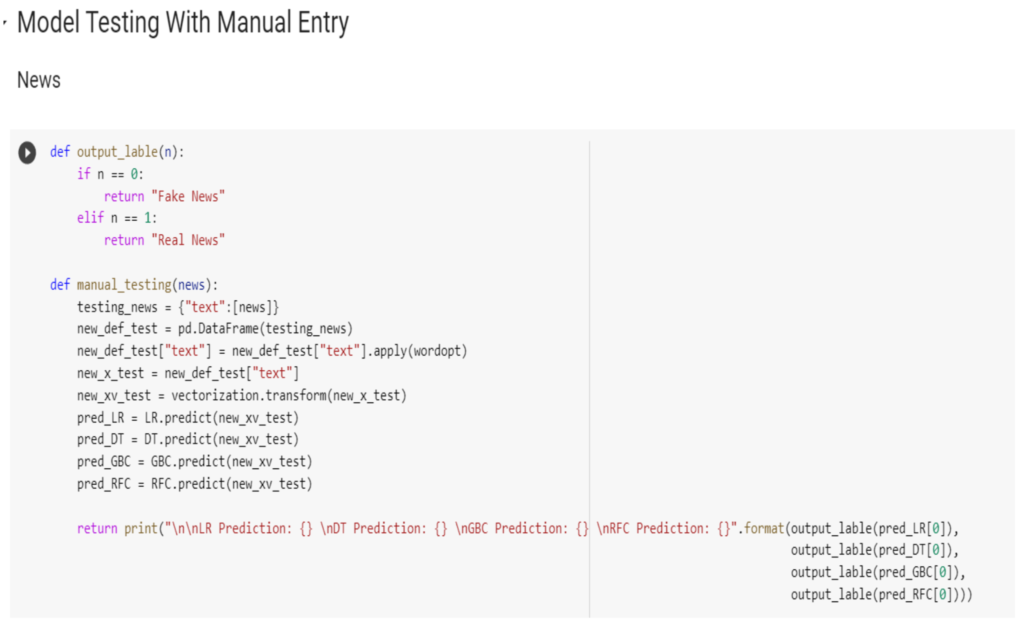


```
[ ]  plt.figure(figsize=(8,7))
     plt.bar(list(dct.keys()),list(dct.values()))
     plt.ylim(90,100)
     plt.xticks(rotation='vertical')

     plt.yticks((91, 92, 93, 94, 95, 96, 97, 98, 99, 100))
```



Graph 8: Comparison of the accuracies of different models

Testing



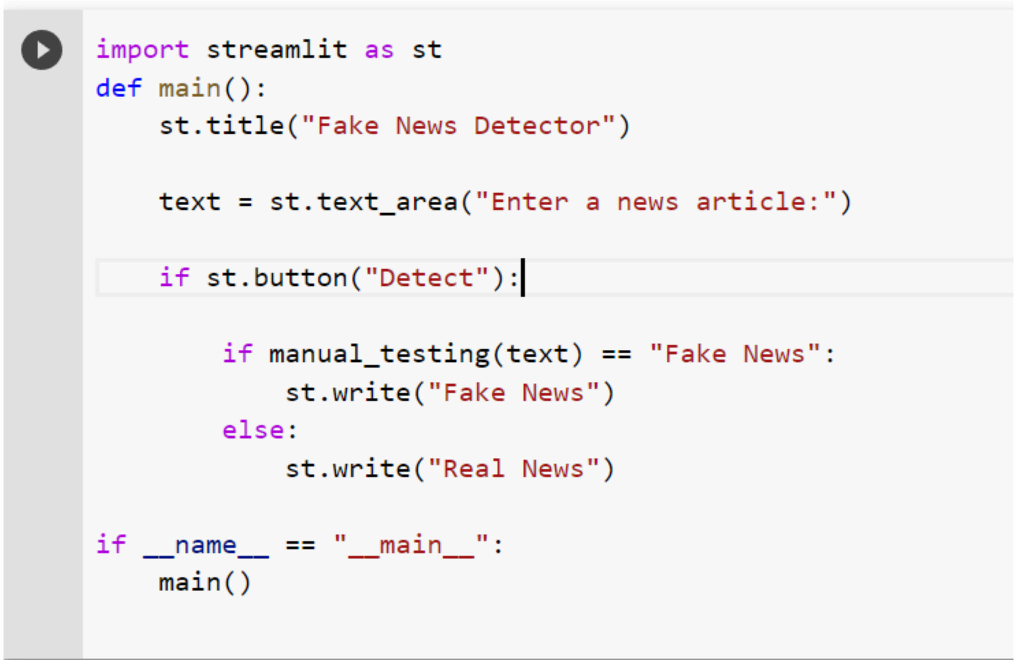
Model Testing With Manual Entry

News

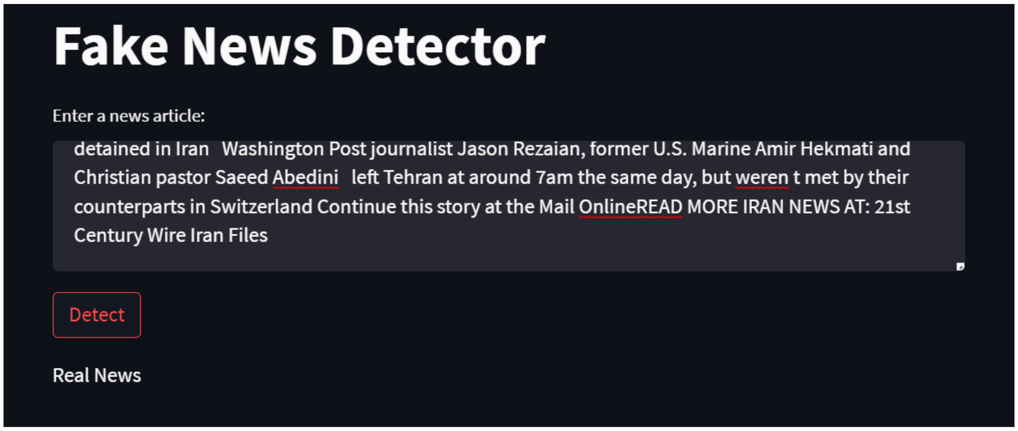
```
def output_lable(n):
    if n == 0:
        return "Fake News"
    elif n == 1:
        return "Real News"

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GBC = GBC.predict(new_xv_test)
    pred_RFC = RFC.predict(new_xv_test)

    return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction: {}".format(output_lable(pred_LR[0]),
                                                                                                              output_lable(pred_DT[0]),
                                                                                                              output_lable(pred_GBC[0]),
                                                                                                              output_lable(pred_RFC[0])))
```

Fig. 30: Testing

Sample Input



```
import streamlit as st
def main():
    st.title("Fake News Detector")

    text = st.text_area("Enter a news article:")

    if st.button("Detect"):

        if manual_testing(text) == "Fake News":
            st.write("Fake News")
        else:
            st.write("Real News")

if __name__ == "__main__":
    main()
```



Fake News Detector

Enter a news article:

detained in Iran Washington Post journalist Jason Rezaian, former U.S. Marine Amir Hekmati and Christian pastor Saeed Abedini left Tehran at around 7am the same day, but weren t met by their counterparts in Switzerland Continue this story at the Mail OnlineREAD MORE IRAN NEWS AT: 21st Century Wire Iran Files

Real News

Fig. 31: Output

CHAPTER-4
RESULTS AND EXPERIMENTAL ANALYSIS

4.1) Models Applied And their Results

Support Vector Machine (SVM)

- Classification and regression problems are resolved using Support Vector Machine, or SVM, one of the most used supervised learning techniques. It is mostly used, nevertheless, in Machine Learning Classification problems.
- SVM chooses the extreme vectors and points that help build the hyperplane. The foundation of the SVM approach is the support vectors, which are utilised to represent these extreme situations. Take a look at the image below, where a decision boundary or hyperplane is used to classify two separate categories:

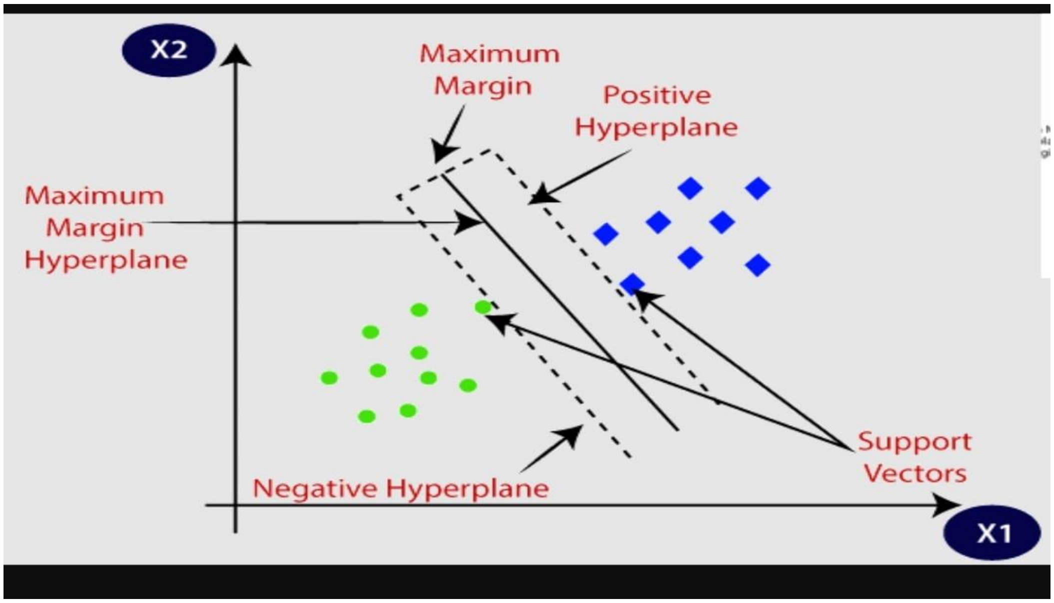
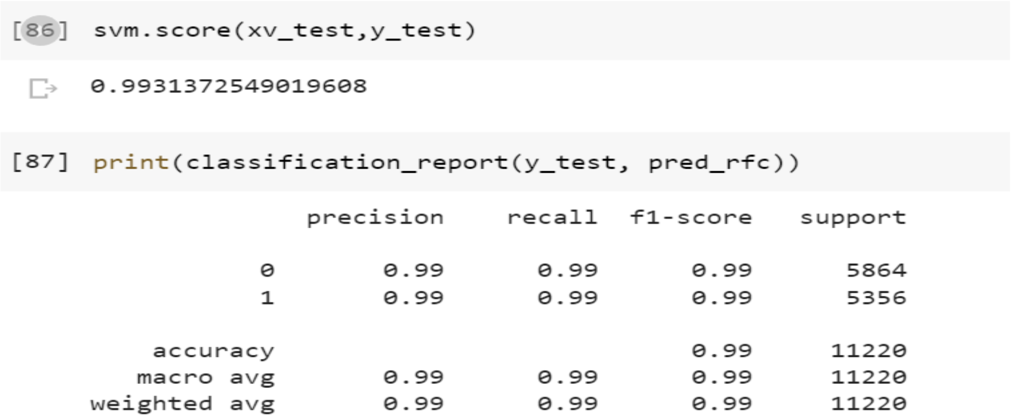


Fig. 32: Support Vector Machine (SVM) [6]

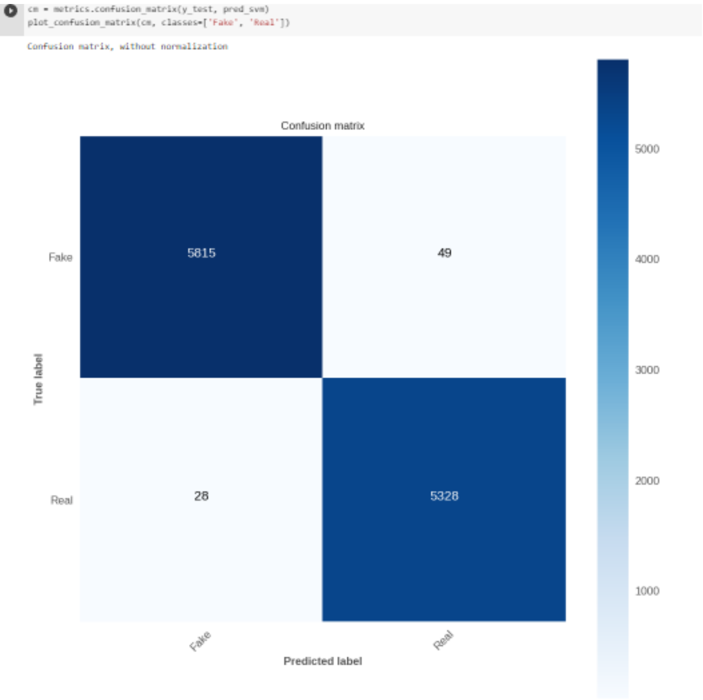
Below are the Results from applying Support Vector Machine model:

Table 1: Classification Report of SVM



```
[86]  svm.score(xv_test,y_test)

      0.9931372549019608

[87]  print(classification_report(y_test, pred_rfc))

                   precision    recall  f1-score   support

               0       0.99      0.99      0.99      5864
               1       0.99      0.99      0.99      5356

        accuracy                           0.99     11220
       macro avg       0.99      0.99      0.99     11220
    weighted avg       0.99      0.99      0.99     11220
```

Confusion Matrix:



```
cm = metrics.confusion_matrix(y_test, pred_svm)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])

Confusion matrix, without normalization
```

Fig. 33: Confusion Matrix from Support Vector Machine

Logistic regression

- In binary classification issues, where the goal is to predict one of two outcomes, logistic regression is a frequently used approach. Through the use of a sigmoid function, it converts the output of the linear regression into a probability value between 0 and 1, which can then be used to decide whether to classify data by applying a threshold.
- With applications in many areas, including credit scoring, spam filtering, and medical diagnosis, this simple yet reliable algorithm may be taught well on big datasets. However, because it depends on certain presumptions, such as the linearity and independence of the characteristics, it could not work well with highly coupled or nonlinear data.

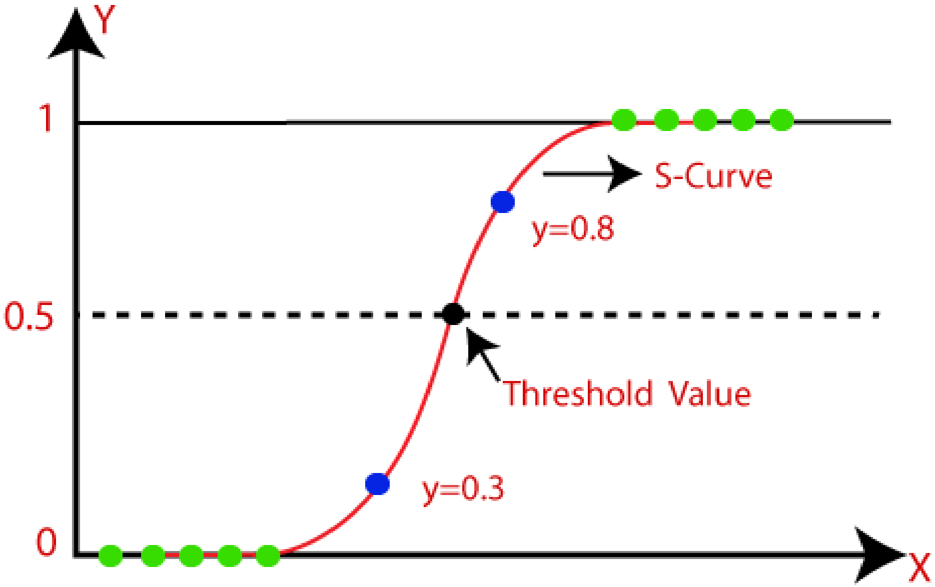
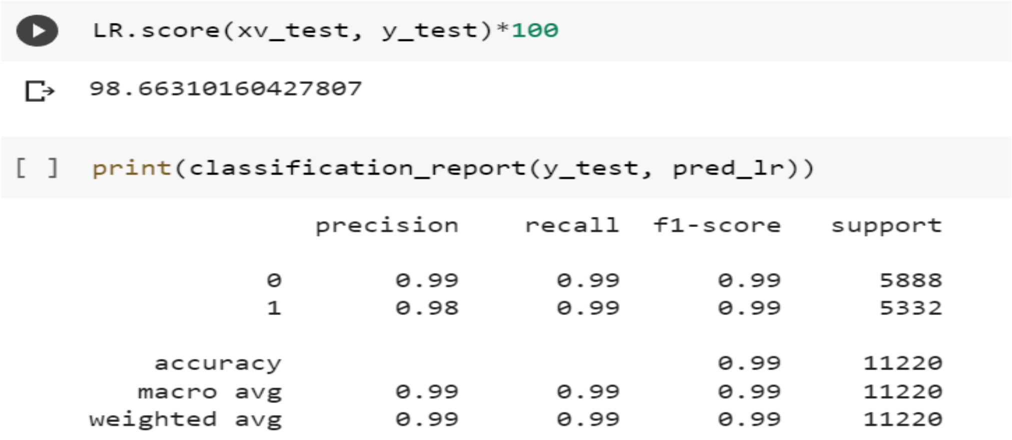


Fig. 34: Logistic Regression [7]

Below are the Results from applying Logistic Regression model:

Table 2: Classification Report of LR



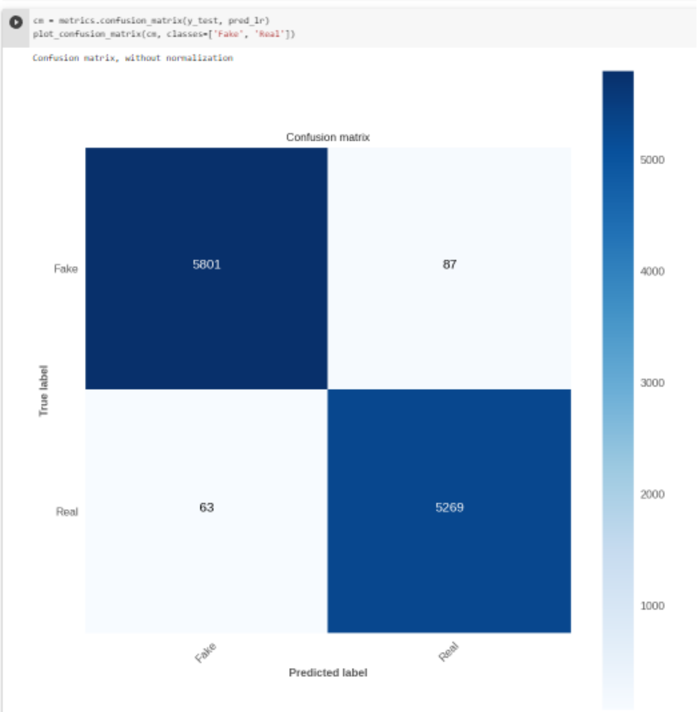
```
LR.score(xv_test, y_test)*100
```

```
98.66310160427807
```

```
[ ]  print(classification_report(y_test, pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5888
           1       0.98      0.99      0.99      5332

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

Confusion Matrix:



```
cm = metrics.confusion_matrix(y_test, pred_lr)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

Fig. 35: Confusion matrix from Logistic Regression

Decesion Tree Classification

- For both binary and multi-class classification tasks, decision tree classification is a popular machine learning approach. The input data are recursively divided into subgroups depending on the most instructive characteristic.
- Decision trees can handle category and numerical data and are simple to understand and use. Additionally, they are resistant to noise and missing data and are capable of capturing intricate non-linear correlations between features.

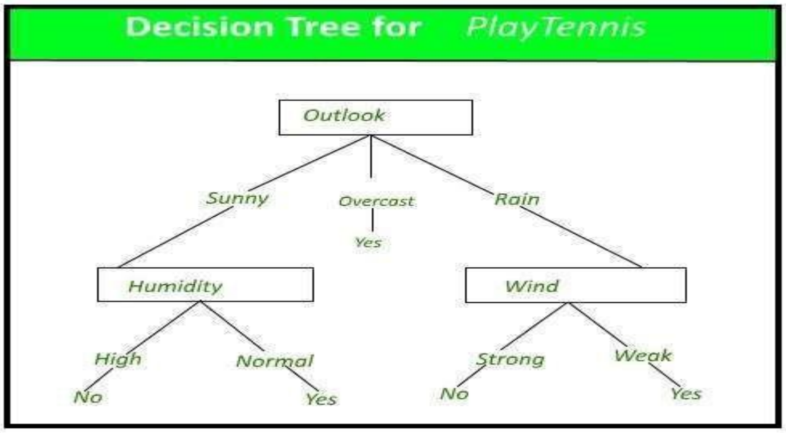
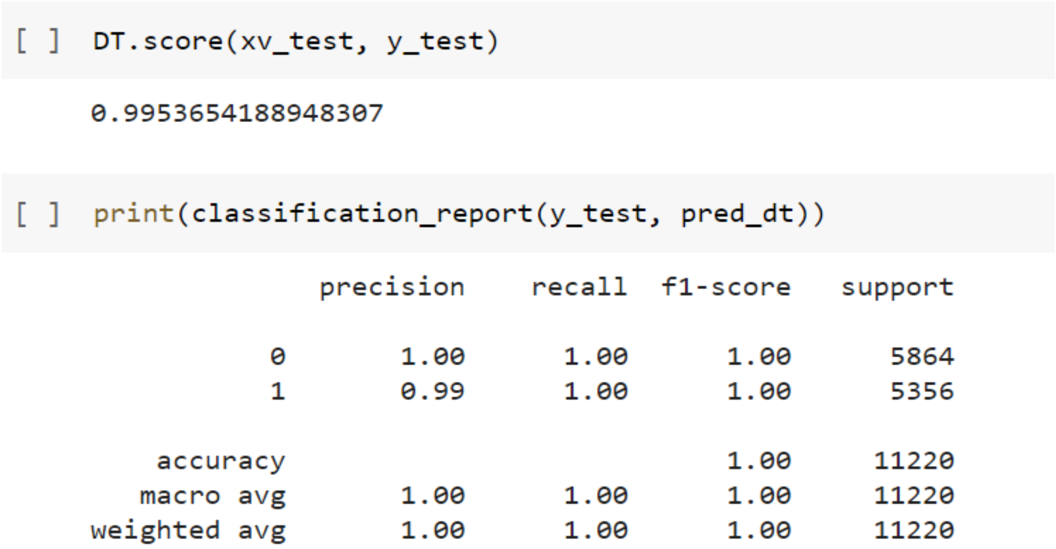


Fig. 36: Decision Tree [8]

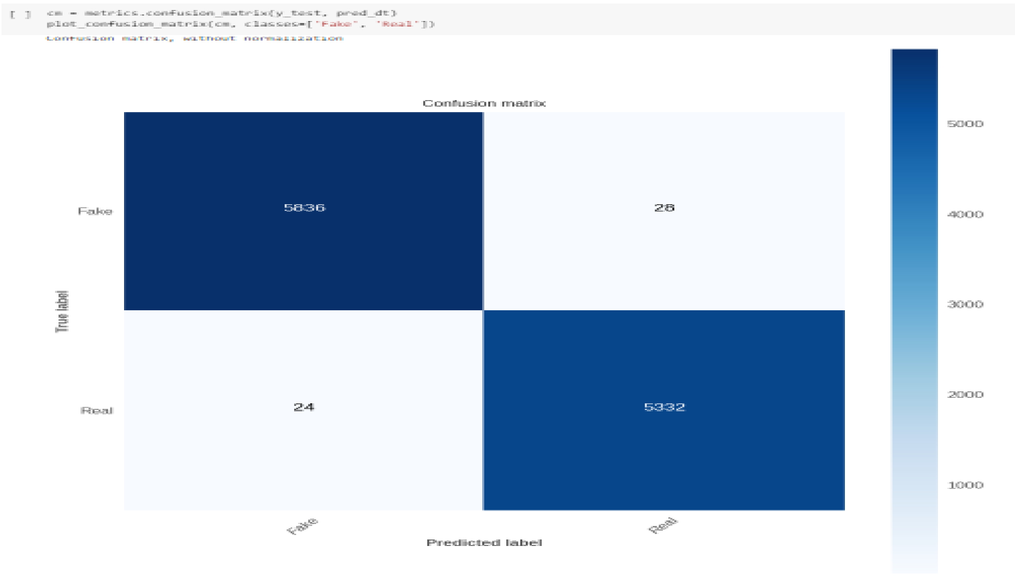
Below are the Results from applying Decision Tree Classification model:

Table 3: Classification Report from Decision Tree



```
[ ]  DT.score(xv_test, y_test)

     0.9953654188948307

[ ]  print(classification_report(y_test, pred_dt))

                   precision    recall  f1-score   support

                0       1.00      1.00      1.00      5864
                1       0.99      1.00      1.00      5356

         accuracy                           1.00     11220
        macro avg       1.00      1.00      1.00     11220
     weighted avg       1.00      1.00      1.00     11220
```

Confusion Matrix:



```
[ ]  cm = metrics.confusion_matrix(y_test, pred_dt)
     plot_confusion_matrix(cm, classes=['Fake', 'Real'])
     Confusion matrix, without normalization
```

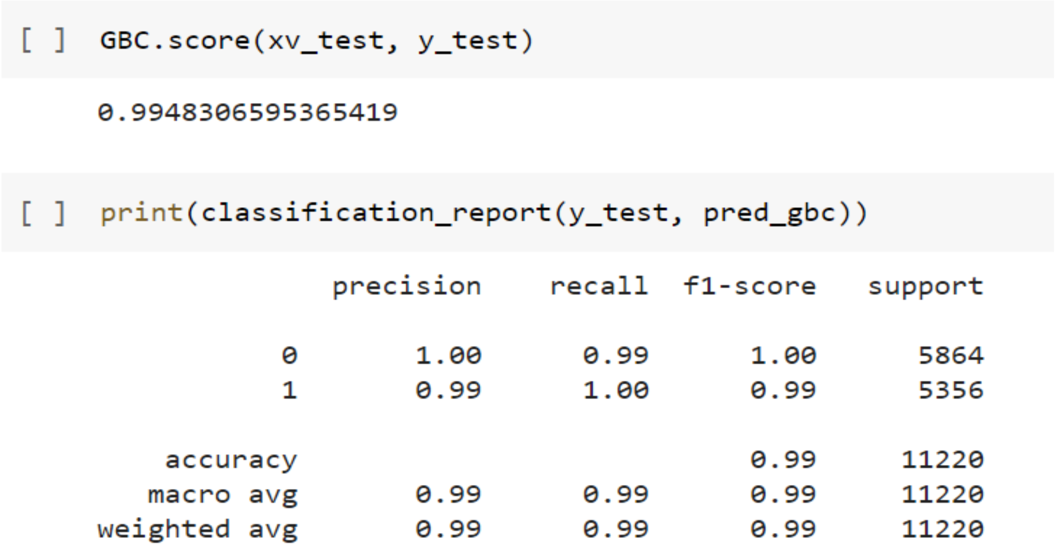
Fig 37: Confusion Matrix from Decision Tree Classification

Gradient Boosting Classifier

- Gradient Boosting Classifier is a powerful algorithm for both classification and regression problems. It works by combining multiple weak models, such as decision trees, to create a strong ensemble model.
- One of the advantages of Gradient Boosting Classifier is that it can handle complex non-linear relationships between features and the target variable. Additionally, it has a built-in mechanism for handling missing data and can automatically select important features for better accuracy. However, it can be computationally expensive and prone to overfitting if not tuned properly.

Below are the Results from applying Gradient boosting classifier model:

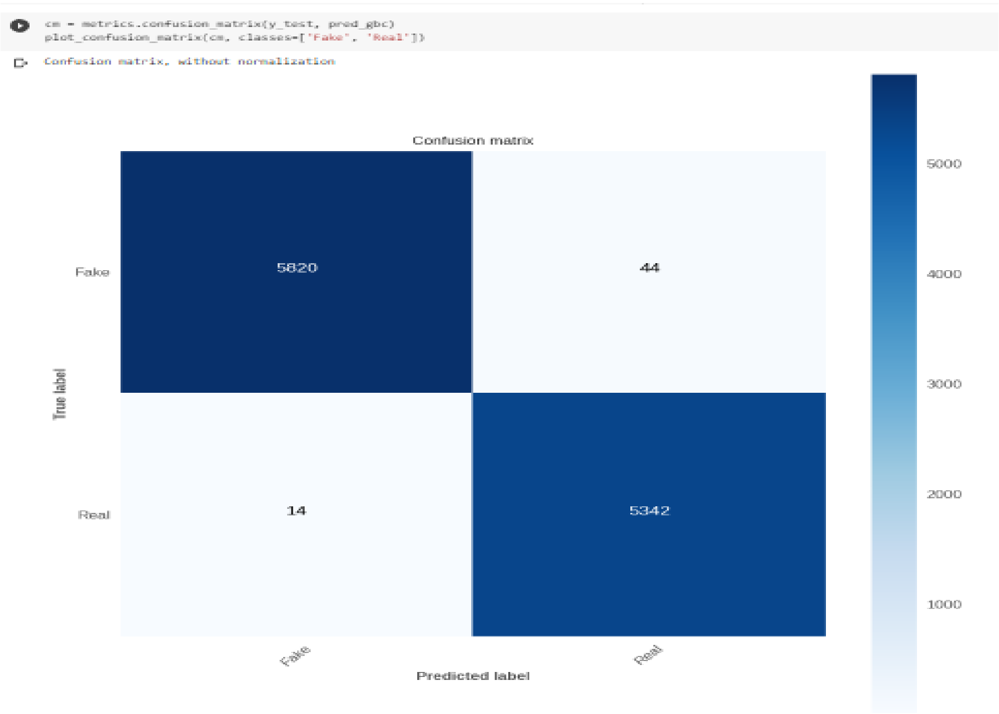
Table 4: Classification Report of GBC



```
[ ]  GBC.score(xv_test, y_test)

     0.9948306595365419
```

```
[ ]  print(classification_report(y_test, pred_gbc))

                   precision    recall  f1-score   support

                0       1.00      0.99      1.00      5864
                1       0.99      1.00      0.99      5356

         accuracy                           0.99     11220
        macro avg       0.99      0.99      0.99     11220
     weighted avg       0.99      0.99      0.99     11220
```

Confusion Matrix:



```
cm = metrics.confusion_matrix(y_test, pred_gbc)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

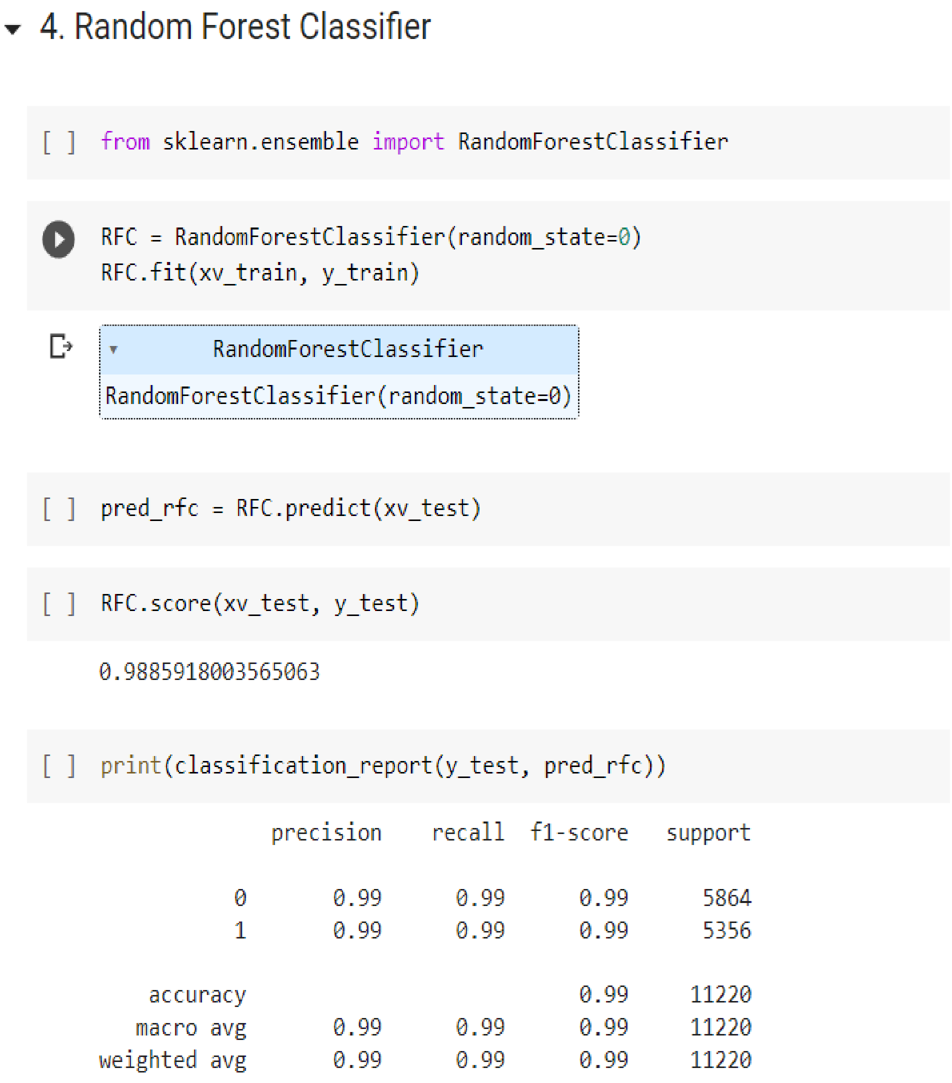
```
Confusion matrix, without normalization
```

Fig. 38: Confusion Matrix from Gradient Boosting Classifier

Random Forest Classifier

- As the name implies, a Random Forest consists of numerous independent decision trees that work together as an ensemble. Each tree in the Random Forest spits out a class prediction, and the classification that recieves the most votes becomes the prediction of our model.

Below are the Results from applying Random Forest Classifier model:



```
from sklearn.ensemble import RandomForestClassifier
```

```
RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
```

```
        RandomForestClassifier
RandomForestClassifier(random_state=0)
```

```
pred_rfc = RFC.predict(xv_test)
```

```
RFC.score(xv_test, y_test)
```

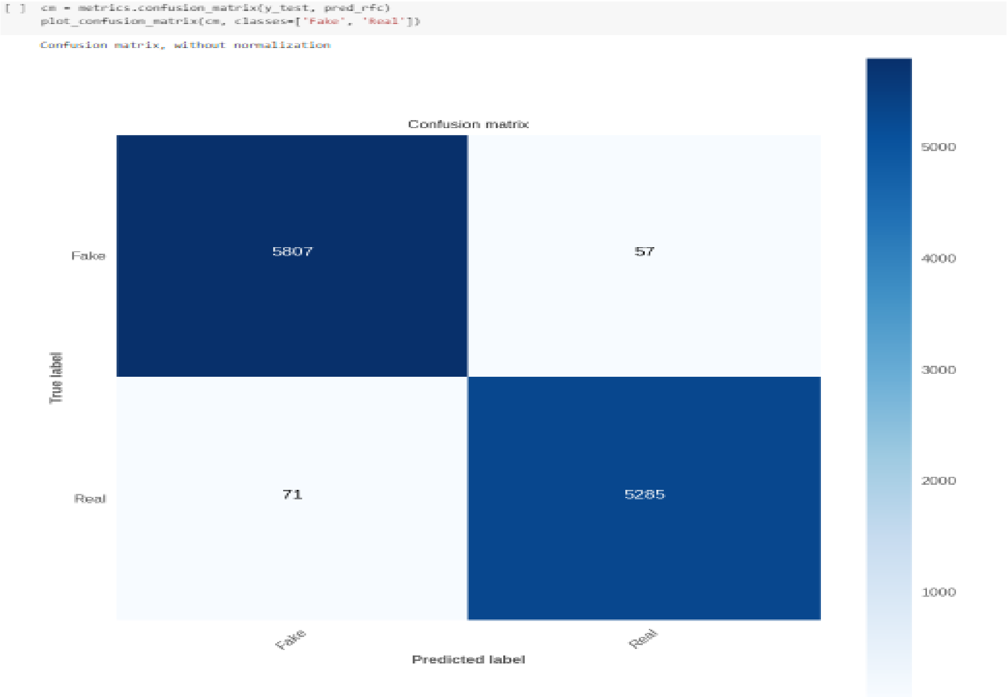
```
0.9885918003565063
```

```
print(classification_report(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5864
           1       0.99      0.99      0.99      5356

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

Table 5: Classification Report of RFC

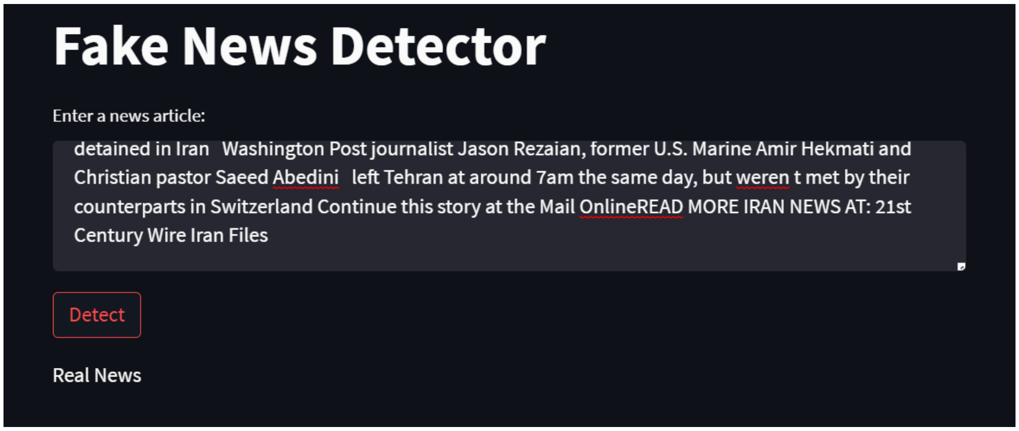
Confusion Matrix:



```
[ ]  cm = metrics.confusion_matrix(y_test, pred_rfc)
     plot_confusion_matrix(cm, classes=['Fake', 'Real'])

     Confusion matrix, without normalization
```

Fig. 39: Confusion Matrix from Random Forest Classifier

Sample Input:



Fake News Detector

Enter a news article:

detained in Iran   Washington Post journalist Jason Rezaian, former U.S. Marine Amir Hekmati and Christian pastor Saeed Abedini   left Tehran at around 7am the same day, but weren t met by their counterparts in Switzerland Continue this story at the Mail OnlineREAD MORE IRAN NEWS AT: 21st Century Wire Iran Files

Real News

Fig. 40: Web Browser Output

CHAPTER-5
CONCLUSIONS

5.1) Conclusions

Considering the accuracy scores, we were able to establish for the various models, it appears that all of the models are doing a good job of identifying false news items. The SVM, Decision Tree, and Gradient Boosting classifiers notably achieved a very high accuracy of 99.5%, although the Random Forest Classifier performed just slightly lower, at 98.71%.

All things considered, these results suggest that a range of classifiers may be used with equal success rates and that machine learning techniques may be extremely successful in spotting bogus news. It's important to keep in mind that accuracy is only one measure and that the models should be evaluated using multiple metrics including precision, recall, and F1-score in addition to factors like interpretability, scalability, and processing requirements. Investigating different feature extraction and selection methods, classifier types, and ensemble approaches may also be useful to see whether even better results may be produced.

We utilised the datasets real and fake, each of which had 21417 and 23481 entries, respectively. We converted text into a numerical model using TF-ID F Vectorizer and utilised the following models:

Accuracy of 99.31% for support vector machines

Decision Tree: 99.5% precision

Classifier using Gradient Boosting: Accuracy = 99.5%

Accuracy of 98.7% for the random forest classifier

5.2) Future Scope

Future research and advancement in the field of false news detection are abundantly possible. Future efforts to identify bogus news may go in the following directions:

Including more varied and subtle aspects: For the most part, current methods for detecting false news rely on simple text-based traits like TF-IDF vectors or bag-of-words. Research in the future could concentrate on more complex and diverse aspects, such sentiment analysis, network analysis, or multimedia analysis (for instance, identifying false images or videos).

Creating more interpretable models: Existing methods for spotting fake news sometimes rely on complex machine learning algorithms that might be difficult to comprehend. In the future, it would be beneficial to develop more intelligible models that might provide more information on how people make decisions.

Combining information from other sources: In addition to social media, news articles, and videos, fake news is regularly spread through other media channels and platforms. The development of methods that can incorporate data from several sources may be crucial in the future to improve false news identification.

Adapting to shifting strategies: It will be crucial for fake news detection technologies to develop alongside the tactics used by those who create and spread it. For this, the detection methods might need to be regularly reviewed and improved.

REFERENCES

[1] A. S. A. Ahmed, A. Abidin, M. A. Maarof, and R. A. Rashid, "Fake news detection: A survey," IEEE Access, vol. 9, pp. 113051-113071, 2021. doi: 10.1109/ACCESS.2021.3104178

[2] S. Asghar, S. Mahmood, and H. Kamran, "Fake news detection using machine learning: A survey," IEEE Access, vol. 9, pp. 57613-57639, 2021. doi: 10.1109/ACCESS.2021.3075392

[3] J. H. Kim, S. H. Lee, and H. J. Kim, "Fake news detection using ensemble learning with context and attention mechanism," IEEE Access, vol. 9, pp. 27569-27579, 2021. doi: 10.1109/ACCESS.2021.3057736

[4] M. F. Hossain, M. M. Islam, M. A. H. Khan, and J. J. Jung, "Fake news detection using hybrid machine learning algorithms," IEEE Access, vol. 8, pp. 233350-233364, 2020. doi: 10.1109/ACCESS.2020.3041149

[5] S. S. Ghosh, A. Mukherjee, and N. Ganguly, "A multi-perspective approach to fake news detection," IEEE Intelligent Systems, vol. 35, no. 5, pp. 31-39, 2020. doi: 10.1109/MIS.2020.3012915

[6]https://www.google.co.in/imgres?imgurl=https%3A%2F%2Fdata-flair.training%2Fblogs%2Fwpcontent%2Fuploads%2Fsites%2F2%2F2019%2F07%2FintroductiontoSVM.png&tbnid=p7ua2IdzmLsjqM&vet=12ahUKEwjf26KfruDAhW6JrcAHdMIAagQMygCegUIARDlAQ..i&imgrefurl=https%3A%2F%2Fdata-flair.training%2Fblogs%2Fsvm-support-vector-machine-

tutorial%2F&docid=7oy5_irTaN4UfM&w=801&h=420&q=svm&ved=2ahUKEwjf26KfruD-AhW6JrcAHdMIAagQMygCegUIARDlAQ

[7]https://www.google.co.in/imgres?imgurl=https%3A%2F%2Fstatic.javatpoint.com%2Ftutorial%2Fmachine-learning%2Fimages%2Flogistic-regression-inmachinelearning.png&tbnid=LuaHnfur76i8eM&vet=12ahUKEwjFoPGSruDAhVNnNgFHUjLCl8QMygCegUIARDjAQ..i&imgrefurl=https%3A%2F%2Fwww.javatpoint.com%2Flogisticregressioinmachinelearning&docid=makIlDmuc8naWM&w=500&h=300&itg=1&q=logistic%20regression&ved=2ahUKEwjFoPGSruD-AhVNnNgFHUjLCl8QMygCegUIARDjAQ

[8]https://www.google.co.in/url?sa=i&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fdecision-tree%2F&psig=AOvVaw0sYuRq-TZe0WWhW-9YQUnl&ust=1683450911500000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCLDwi7qt4P4CFQAAAAAdAAAAABAE

APPENDICES

The following phrases and words are frequently used in legitimate news and can be used to spot fake news:

- Sources
- Evidence Experts
- Research Statistics Facts
- Data Quotes
- Corroborate Verification
- Objective
- Impartial
- Reliable
- Credible
- Transparency
- Context Timeliness
- Accuracy
- impartial reporting
- several perspectives

Certainly, here are some commonly used words and phrases that may indicate the presence of fake news:

- Allegedly
- Supposedly
- Claims
- "Fake news" or "hoax"
- Conspiracy
- Unverified
- Sensational
- Emotional
- Outrageous
- Shocking
- Clickbait
- Exaggerated
- Biased
- Partisan
- Misleading
- Inaccurate
- Unsubstantiated
- Rumors
- Speculation
- Opinions presented as facts.