Fake News Detection Techniques for Diversified Datasets Using Machine Learning

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of,

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

P SAI KIRAN	**321910304008**
K MANJUNATH REDDY	**321910304031**
VIPUL MUNESWAR K	**321910304032**
K DEVINATH	**321910304036**
J D TEJA SAI	**321910304045**

Under the esteemed guidance of

Mrs. Asha G

Assistant Professor

Department of Computer Science & Engineering,

GITAM SCHOOL OF TECHNOLOGY

(Deemed to be University)

Bengaluru Campus.

November 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled **"Fake News Detection Techniques for Diversified Datasets Using Machine Learning"** is a Bonafide work carried out by **P SAI KIRAN (321910304008), K MANJUNATH REDDY (321910304031), VIPUL MUNESWAR K (321910304032), K DEVINATH (321910304036), J D TEJA SAI (321910304045)** submitted in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering.

Head of the Department
Computer Science & Engineering
GITAM School of Technology,
GITAM (Deemed to be University)
(Bengaluru Campus)
Nagadenahalli-561 203
Bengaluru Rural Dist. Karnataka

Project Guide.

SIGNATURE OF THE GUIDE

Mrs. Asha G

Assistant Professor.
Dept of CSE, GST

Head of the Department.

SIGNATURE OF THE HOD

Dr. Vamsidhar Yendapalli

Professor.
Dept of CSE, GST

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled **"Fake News Detection Techniques for Diversified Datasets Using Machine Learning"** is an original work done in the **GITAM School of Technology, GITAM (Deemed to be University)** in partial fulfillment of the requirements for the award of the degree of **B.Tech.** in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree.

Date:

Registration No(s).	Name(s)	Signature(s)
321910304008	P SAI KIRAN	
321910304031	K MANJUNATHA REDDY	
321910304032	VIPUL MUNESWAR K	
321910304036	K DEVINATH	
321910304045	J D TEJA SAI	

ACKNOWLEDGEMENT

We had been able to complete our project successfully. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to GITAM (Deemed to be University), Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project and for their support in completing the project.

We Would like to express our gratitude towards Prof. Vamsidhar Y (Professor, HOD of CSE, GST), Mrs. Asha G (Assistant Professor, CSE, GST) and to all the other supporting faculty and staff for their kind co-operation and encouragement which helped us in the completion of this project.

Student Name's	**Registration No.**
P SAI KIRAN	321910304008
K MANJUNATHA REDDY	321910304031
VIPUL MUNESWAR K	321910304032
K DEVINATH	321910304036
J D TEJA SAI	321910304045

ABSTRACT

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. The many lives of individuals now hang in the balance as a result of social media. Much has already been accomplished in these three fields, including contact, advertising, news, and agenda advancement Misinformation is sometimes used on Twitter, particularly by some malicious accounts. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain must explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we propose to use a machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. Social networking is one of the most critical subjects in the business world today. For that reason, it is critical to pinpoint a malicious account. Machine learning methods were applied in this study to try to identify accounts that could be manipulated to look like the real ones. The data has been analyzed for these purposes, and learning algorithms have been used to identify fake news. By using these properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Fake news detection (FND) has recently picked the attention of a large number of academics, with many sociological studies demonstrating the effect of fake news and how people respond to it. To describe fake news as any material capable of making readers believe in information that is not real, one must first define what false news is. Spreading false news widely harms society and the person. Initially, this kind of false news has the potential to change or destroy the authenticity balance in the news ecosystem.

Because of the features of fake news, people are coerced into accepting incorrect or skewed ideas they would otherwise reject. Political messages or influence is often communicated via the use of false news and propagandists.

Fake news has a lasting impact on how people interact with and react to genuine news. To reduce the harmful impacts of false news, it is critical to develop a system that can automatically detect it when it appears on social media. However, there are several difficult research issues with fake news detection on different social platforms.

A variety of research objectives observed in this regard includes the identification of the source of origin or uploading of the news or data on the social network, to understand the actual intention or meaning of the data uploaded and to determine the extent of authenticity and validate it to make decision to consider it as genuine or fake.

The peculiarities of news make automated fake news detection a difficult task. Existing knowledge bases fail to validate false news effectively when it is linked to time-critical events because there are not enough supporting claims or facts to back them up. The data (i.e., unstructured, noisy, unfinished, and large data) generated by false news is also on social media. Researchers have attempted in recent years to uncover problems with false news, their trustworthiness on social media, especially Twitter, YouTube, Facebook, and television.

Due to these network interactions, it is possible to extract valuable post features while also taking advantage of the network's interactions. Fake news has characteristics, kinds, and detection methods, and all of which are discussed in this study. Further research on fake news detection apps will be guided by appropriate explanations regarding false news.

The benefits and limits of conventional fake news detection are addressed, as well as the difficulties posed by false news on social media. However, there are several problems with the fake news detection social media presence that need additional study. Now-a-days Facebook and Instagram are the most leading media platforms.

It has existed at the center of much analysis following media attention. But these media platforms also ask the users to beware of fake news and help users with the help of reporting particular pages but most of the users are not much shown interest. A given algorithm must be politically unbiased.

The scope of the project is to find the effectiveness and limitations of language-based techniques for detection of fake news using machine learning algorithms.

The widespread problem of fake news is very difficult to tackle in today's digital world where there are thousands of information sharing platforms through which fake news or misinformation may propagate. It has become a greater issue because of the advancements in AI which brings along artificial bots that may be used to create and spread fake news.

The fake news is classified into two types Misinformation and disinformation

- **Misinformation**
    - Misinformation is "false information that is spread, regardless of intent to mislead."
    - Misinformation does not care about intent, and so is simply a term for any kind of wrong or false information.
    - Today, misinformation spreads very easily thanks to technology. On social media, users have as just one tiny instance shared stories about dolphins and swans swimming in the canals of Venice without checking if those stories are true (they weren't).
    - It is a part because of such frequent incidents, it is a hot topic of debate if big tech companies like Facebook and Google should be responsible for stopping the spread of misinformation or even if they even can without violating the First Amendment freedom of speech rights of their users.
    - Misinformation is, of course, related to the verb misinform, which means "to give wrong or misleading information to" and is first recorded around 1350–1400.
    - You will notice that misinform, like misinformation, also makes no mention of why this wrong information is being spread around, only that it is.

- **Disinformation**
    - Disinformation means "false information, as about a country's military strength or plans, disseminated by a government or intelligence agency in a hostile act of tactical political subversion."
    - It is also used more generally to mean "deliberately misleading or biased information; manipulated narrative or facts; propaganda."
    - So, disinformation is misinformation that is knowingly (intentionally) spread.
    - Our first definition of this word gives one major reason why a person or group might want to spread wrong information, but there are many other nefarious motivations lurking behind the creation of disinformation.
    - Disinformation is very powerful, destructive, and divisive, and is a common tool of espionage.
    - Countries often have an interest in intentionally spreading fake information to their rival nations, as the Soviet Union and United States did during the Cold War, for instance.

The overall report revolves around the objective of a fake news detection system using Machine Learning.

- First chapter deals with the introduction of a fake news detection system using Machine Learning. In that we have included overview, motivation, objectives, and scope.
- Second chapter deals with literature review. In that we include details of every literature survey we collected.
- Third chapter deals with system Analysis and requirements. In this we can see the existing system with disadvantages, proposed system with advantages and system requirements.
- Fourth chapter deals with the Problem statement and its objectives.
- Fifth chapter deals with Designing. This includes system architecture and methodology.
- Sixth chapter deals with implementation. In this chapter we will see how to implement the system.
- Seventh chapter deals with Testing of the Implementation.
- Eighth chapter deals with results and analysis.
- Ninth chapter deals with the conclusion part.
- Tenth chapter deals with Future work.

CHAPTER 2

LITERATURE SURVEY

1. Twitter Spam Account Detection Based on Clustering and Classification Methods.

In 2020, **K. S. Adewole, T. Han, W. Wu, H. Song, and A. K. Sangaiah** in their research paper concluded their approach for fake news detection using ML and DL algorithms to detect fake news that is consciously spread on online social networks. It will be helpful in adopting appropriate means and method for dealing with fake data resulting in the betterment of the society. [1]

2. Opinion spam detection framework using hybrid classification scheme.

In 2020, **M. Z. Asghar, A. Ullah, S. Ahmad, and A. KhanM. Z. Asghar, A. Ullah, S. Ahmad, and A. Khan** in their research paper concluded their approach for fake news detection using logistic regression model to detect Opinion spam. They are detected by labeling duplicate spam reviews as a positive training sample and other as negative training sample. The proposed technique assists in classifying the text as spam and non-spam by using a hybrid set of features, prioritize the spamicity features using revised feature weighting scheme, and finally, the text is classified as spam and non-spam. The experimental results in the form of accuracy, precision, recall, and F-measure show that the proposed system outperformed the comparing methods. [2]

3. Detection and visualization of misleading content on Twitter.

In 2018, **C. Boididou, S. Papadopoulos, M.Zampoglou, L.Apostolidis, O.Papadopoulou, and Y.Kompatsiaris** in their research paper concluded their approach for fake news detection using NLP technique and URL analysis to detect fake news in different parts of solution, which has become a very important problem in online networks. As a result, journalists and editors need new tools that can help them speed up the verification process for content that is sourced from social media. [3]

4. **Spam analysis of big reviews dataset using Fuzzy Ranking Evaluation Algorithm and Hadoop.**

In 2019, **K. Dhingra and S. K. Yadav** In their research paper concluded their approach for fake news detection using FREA (Fuzzy Ranking Evaluation Algorithm) and Hadoop to detect the spam of big reviews dataset. In further demonstration their proposed algorithm using a sample reviews dataset and Amazon reviews dataset achieving an accuracy of 80.77% which unlike other approaches remains steady for many groups and deals well with uncertainty involved in opinion spam detection. so, it is advantageous to separate fake news spread from online social networks on Twitter as spam or non-spam. [4]

5. **IJERT-Fake News Detection using Machine Learning Algorithms.**

In 2020, **Uma Sharma, Sidarth Saran, Shankar M. Patil** In their research paper concluded their approach for fake news detection using various types of classifier and they are:

- PA (Passive Aggressive) Classifier.
- Naive Bayes Classifier.
- Random Forest Classifier.

To implement this, various NLP and Machine Learning Techniques must be used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. [5]

CHAPTER 3
SOFTWARE AND HARDWARE SPECIFICATIONS

3.1. Specific Requirements

3.1.1 FUNCTIONAL REQUIREMENTS

- The functions of software systems are defined in functional requirements and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.
    - Our system should be able to read the data and preprocess data.
    - It should be able to analyze the fake data.
    - It should be able to group data based on hidden patterns.
    - It should be able to assign a label based on its data groups.
    - It should be able to split data into train sets and test sets.
    - It should be able to train models using a train set.
    - It must validate the trained model using a test set.
    - It should be able to classify the fake and real data.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

- Nonfunctional requirements illustrate how a system must behave and create constraints of its functionality. This type of constraints is also known as the system's quality features.
- Some Non-Functional Requirements are as follows:
    - Reliability
    - Maintainability
    - Performance
    - Portability
    - Scalability
    - Flexibility

3.2. Hardware and Software Requirement

3.2.1. Hardware:

- Minimum RAM:-2GB
- Hard Disk:-360 GB
- Processor:-Intel Pentium i3

3.2.2. Software:

- Language: - Python
- Backend: - Python
- Operating System: - Windows 11

3.3 NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called np array, it provides a lot of supporting functions that make working with np array very easy. Arrays are very frequently used in data science, where speed and resources are very important. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This is the main reason why NumPy is faster than lists. Also, it is optimized to work with latest CPU architectures. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted,[6] and they both allow the user to write fast programs if most operations work on arrays or matrices instead of scalars. Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing, or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as a universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

3.4 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. Advantages:

- easy to learn and use
- python is broadly adopted and supported
- python is not a toy language
- open-source with a vibrant community
- extensive support libraries Python offers concise and readable code. [9]

3.5 Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science. Pandas are also able to delete rows that are not relevant, or contain wrong values, like empty or NULL values. This is called cleaning the data.

Pandas is mainly used for data analysis and associated manipulation of tabular data in DataFrames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data Frames that were established in the R programming language. The pandas library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data Frames. [7]

3.6 Matplot

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Matplotlib was originally written by John D. Hunter. Since then it has had an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a NumFOCUS fiscally sponsored project. Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement. [8]

PROBLEM STATEMENT

The problem of knowing the news available from the websites, magazines, Daily newspaper etc…
We do not even know whether the available news in social media are Real or Fake. So, we came up with the solution of this problem with the help of some machine learning algorithms. We have designed a model which will help to analyze the data that is available in the social media is Real or Fake using the datasets.

Objectives

- The main objective for this project is to build a model that can differentiate the misleading information in social media and to differentiate between "Real" and "Fake" news that we see in our daily life.
- It is imperative that any attempts to manipulate or troll the Internet through fake news are encountered with absolute effectiveness so to avoid this news we are using this model.
- We get accuracy from this model with the help of datasets which will use to get the accuracy by using the machine learning algorithms like Passive Aggressive Classifier, Random forest, Logistic Regression and other algorithms etc..
- With this accuracy we calculated a plot diagram to get a detailed understanding of the accuracy of algorithms.

CHAPTER 5
DESIGNING

5.1 System Architecture

The below figure shows the architecture diagram for the fake news detection system using machine learning classifiers.

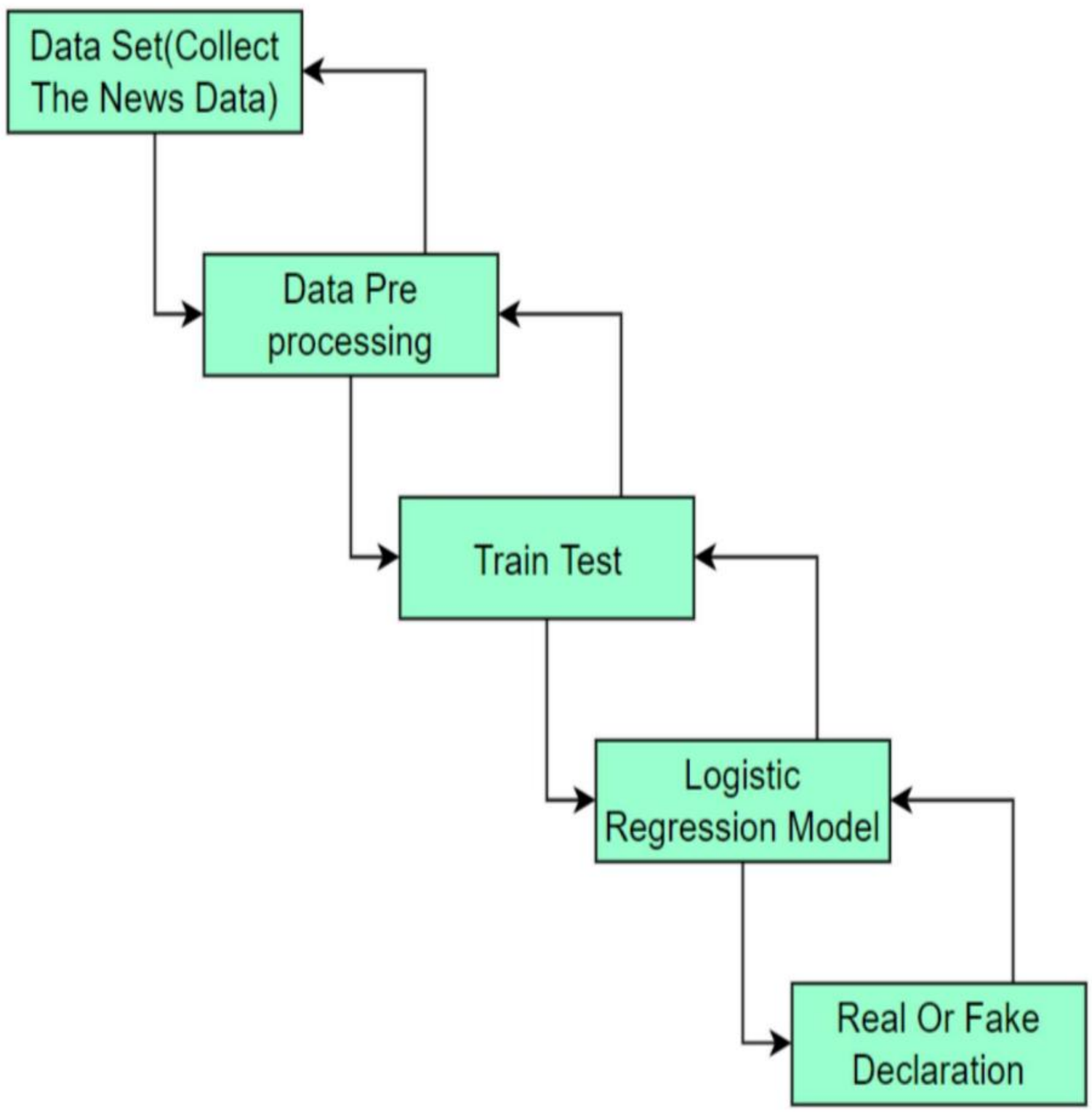

Fig 5.1

5.2 Methodology

5.2.1 Use case Diagram

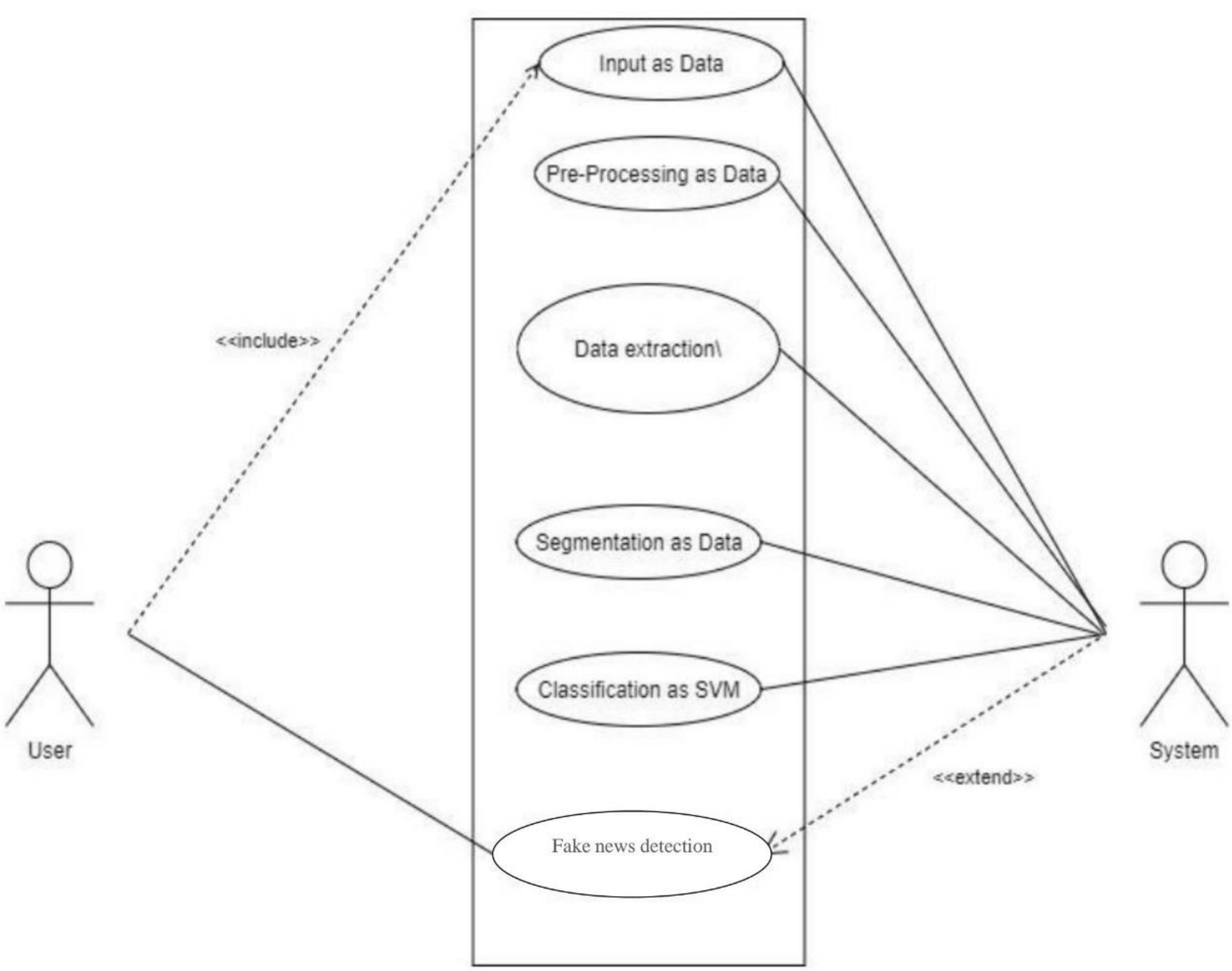

Fig 5.2.1

5.2.2 Sequence diagram

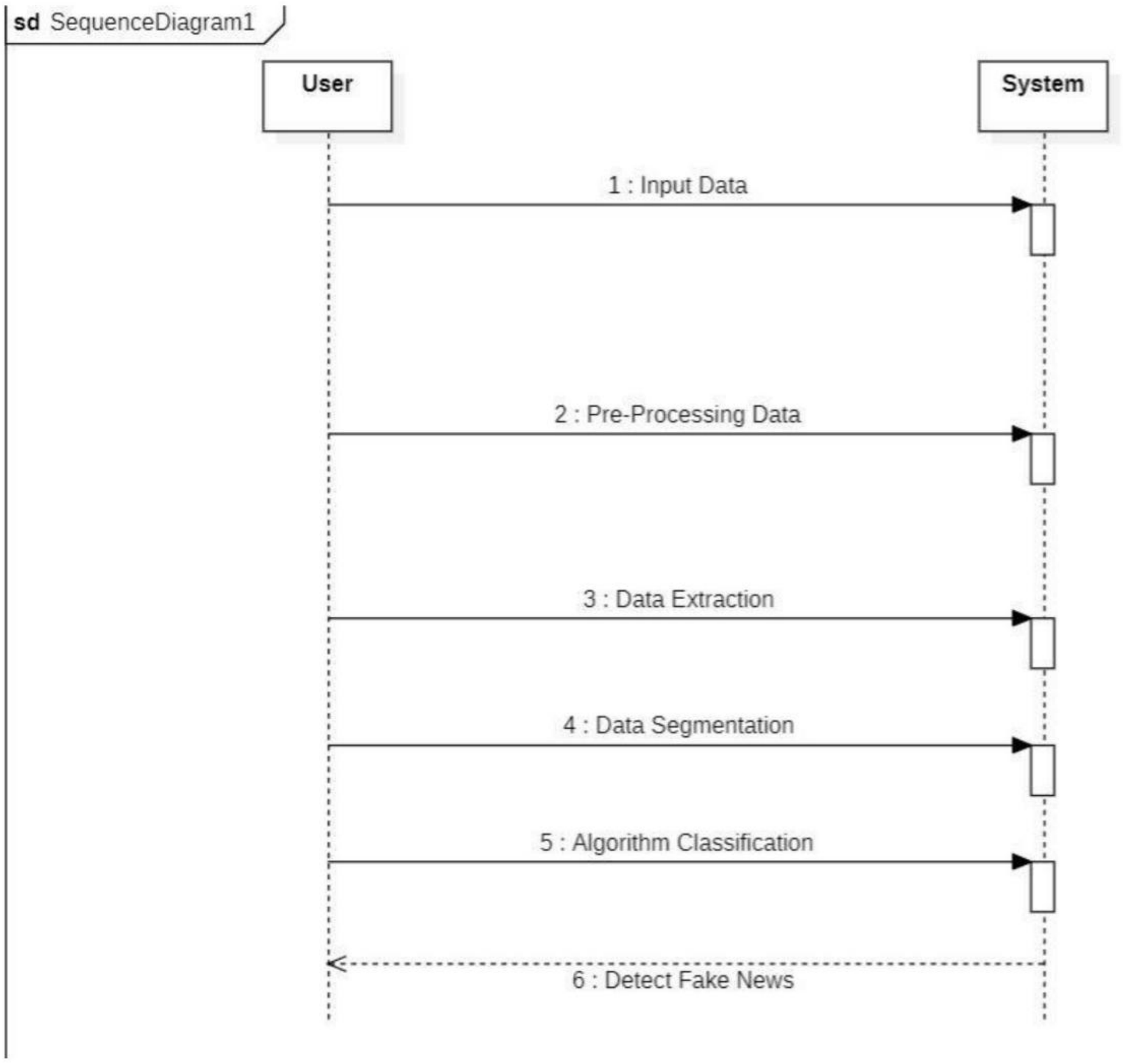

Fig 5.2.2

5.2.3 TFIDF Vectorizer

TFIDF, short for term frequency–inverse document frequency, is a mathematical measurement that is conscious of how significant a word is to a record in an assortment or corpus. It is regularly utilized as a weighting factor in searches of data recovery, text mining, and client displaying. The tfidf esteem augments proportionately to the occasions a word shows up in the record and is balanced by the quantity of archives in the corpus that contain the word, which helps to change for the way that a few words appear more regularly all in all.

Term frequency

Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:

- Number of times the word appears in a document (raw count).
- Term frequency adjusted for the length of the document (raw count of occurrences divided by number of words in the document).
- Logarithmically scaled frequency (e.g. log (1 + raw count)).
- Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).

Inverse document frequency

Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows where ***t*** is the term (word) we are looking to measure the commonness of and ***N*** is the number of documents (d) in the corpus (D). The denominator is simply the number of documents in which the term, ***t***, appears in.

$$idf\,(t, D) = log\,(\ \frac{N}{count\,(d \in D{:}t \in d)}\ )$$

TFIDF

To summarize the key intuition motivating TF-IDF is the importance of a term is inversely related to its frequency across documents.TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. By multiplying these values together, we can get our final TF-IDF value.

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

5.2.4 Logistic regression Classifier

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc. Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types

Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

Multinomial

In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent "Type A" or "Type B" or "Type C".

Ordinal

In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent "poor" or "good", "very good", "Excellent" and each category can have the scores like 0,1,2,3.

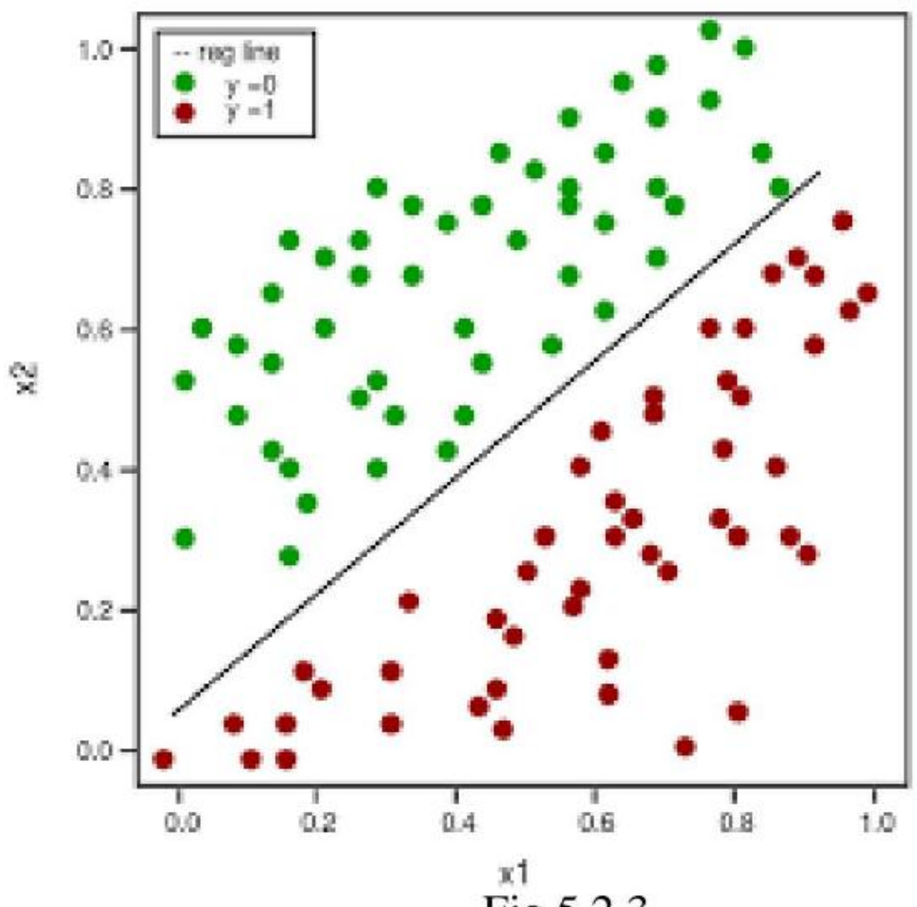

Fig 5.2.3

5.2.5 Decision Tree Classifier

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed based on features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

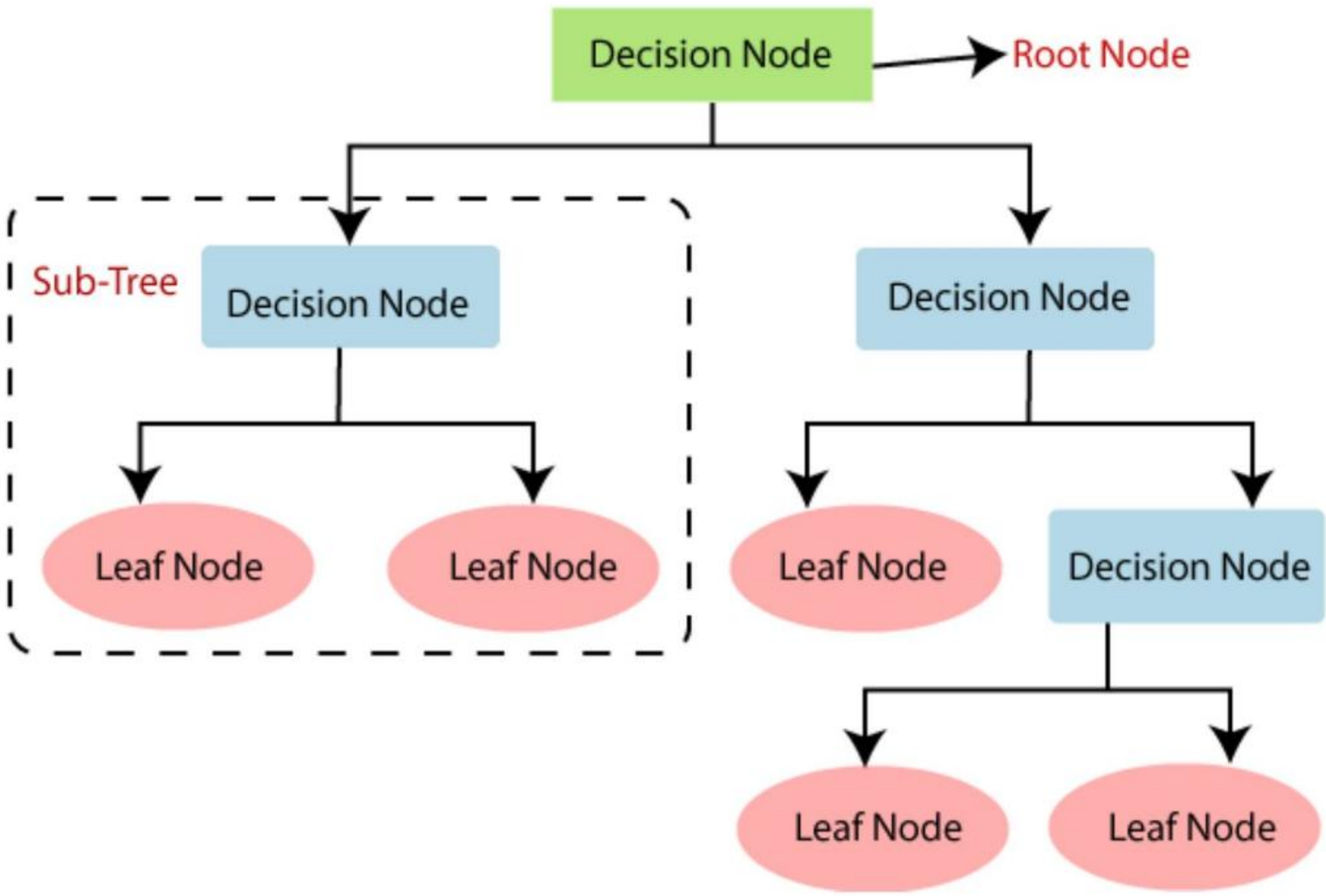

Fig 5.2.4

5.2.6 Random forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

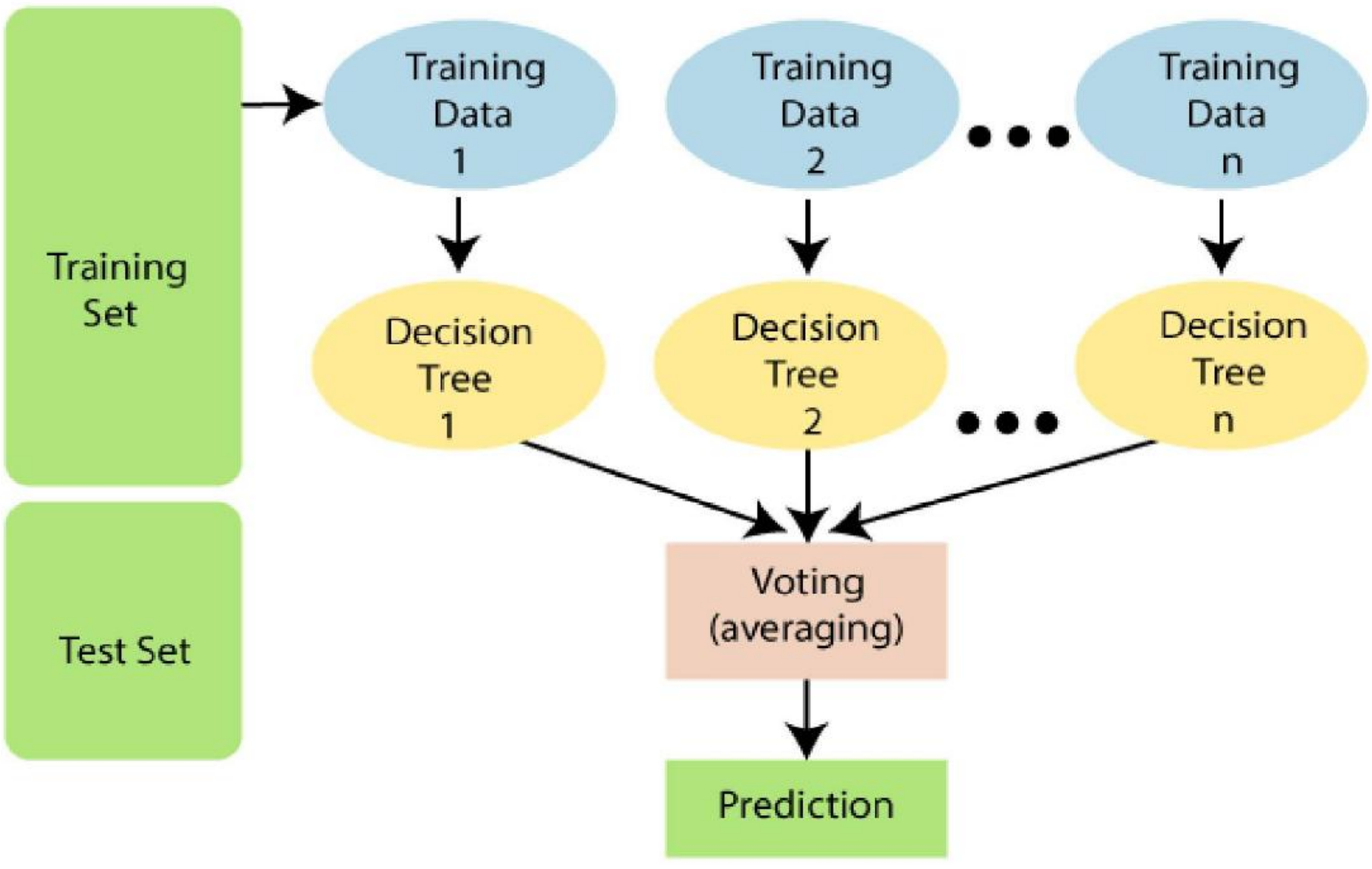

Fig 5.2.5

5.2.7 SVM (Support Vector Machine) classifier

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

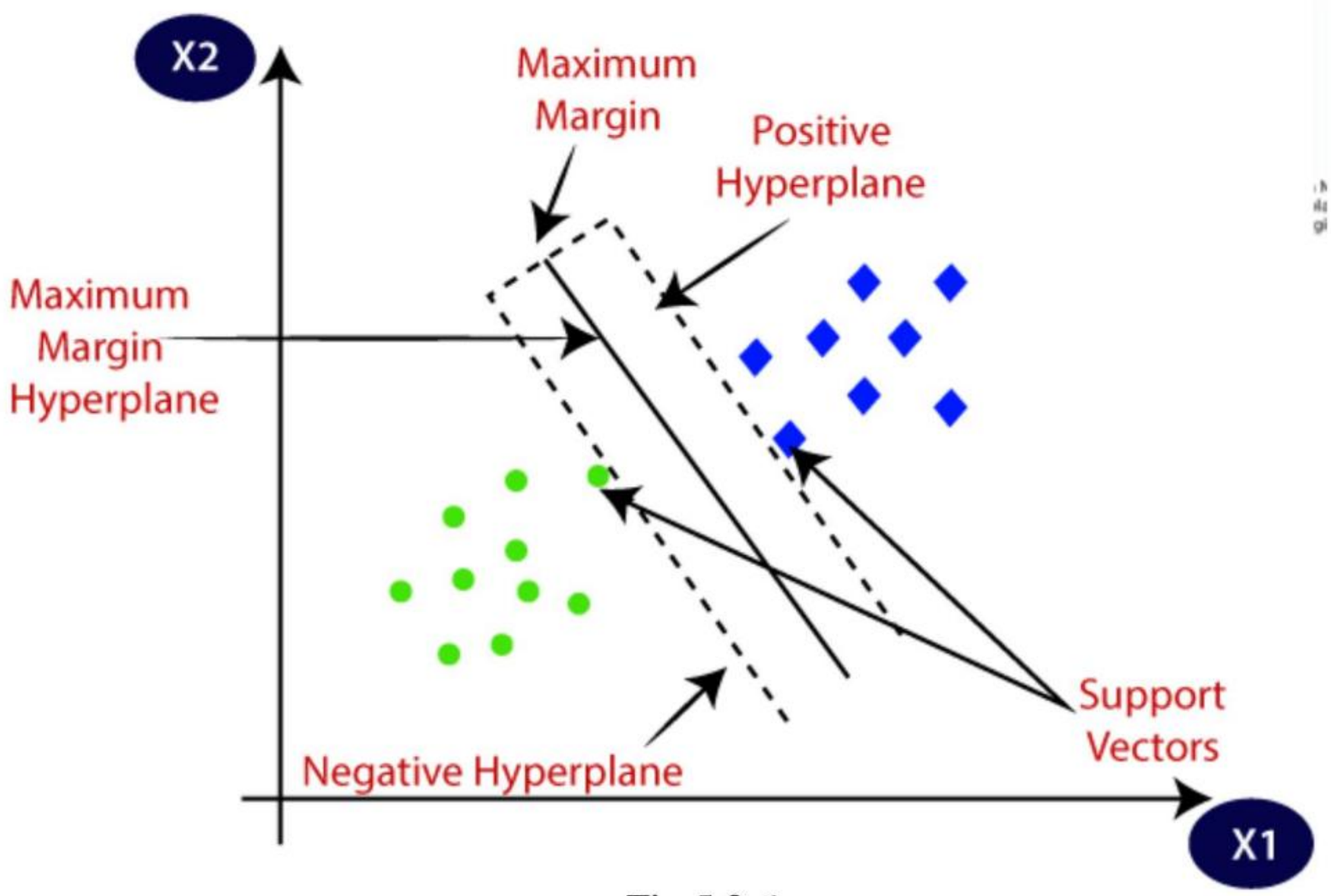

Fig 5.2.6

5.2.8 Naive Bayes

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naive Bayes Equation

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c\ /X) = P(x1\ /c) \times P(x2\ /c\ ) \times \ldots \times P(x2\ /\ c) \times P(c)$$

This algorithm works on Bayes theorem under the assuming that its free from predictors and is used in multiple machine learning problems. Simply put, Naive Bayes assumes that one function in the category has nothing to do with another. For example, the fruit will be classified as an apple when its of red color, swirls, and the diameter is close to 3 inches. Regardless of whether these functions depend on each other or on different functions, and even if these functions depend on each other or on other functions, Naive Bayes assumes that all these functions share a separate proof of the apples

Where:

P(c | X) is the posterior Probability.

P(x | c) is the Likelihood.

P(c) is the Class Prior Probability.

P(x) is the Predictor Prior Probability.

Naive Bayes Pseudo-code

Training dataset T,

F= (f1, f2, f3,..., fn) // value of the predictor variable in testing dataset.

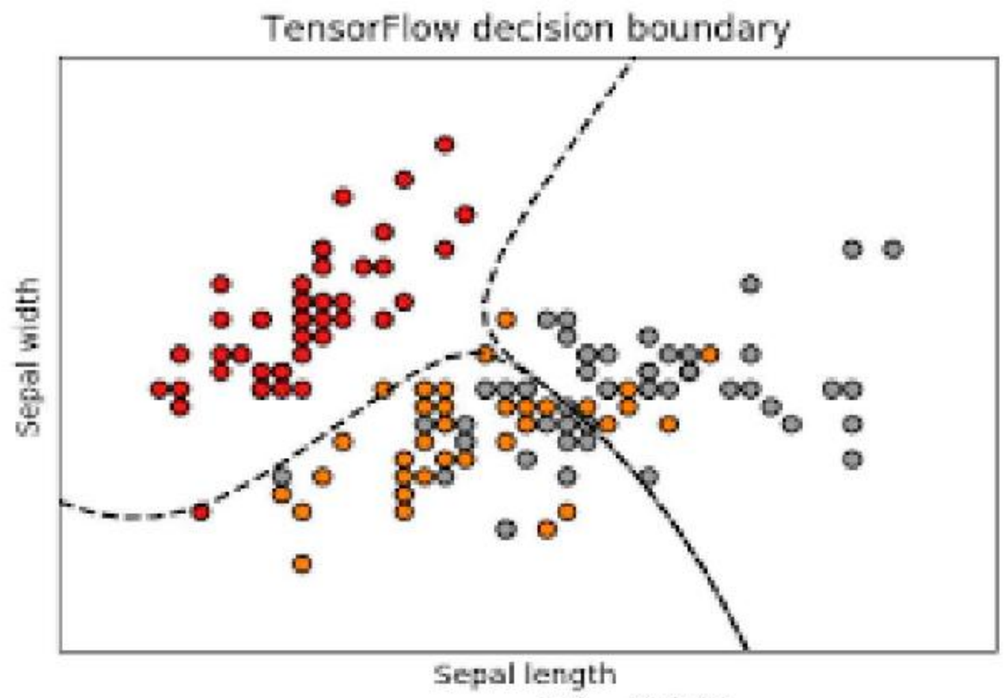

Fig 5.2.7

5.2.9 Passive Aggressive Classifier

The Passive-Aggressive algorithms are a family of Machine learning algorithms that are not very well known by beginners and even intermediate Machine Learning enthusiasts. However, they can be very useful and efficient for certain applications. This is a high-level overview of the algorithm explaining how it works and when to use it. It does not go deep into the mathematics of how it works. Passive-Aggressive algorithms are generally used for large-scale learning. It is one of the few 'online-learning algorithms. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once.

This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data. We can simply say that an online-learning algorithm will get a training example, update the classifier. Passive-Aggressive algorithms are called so because:

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

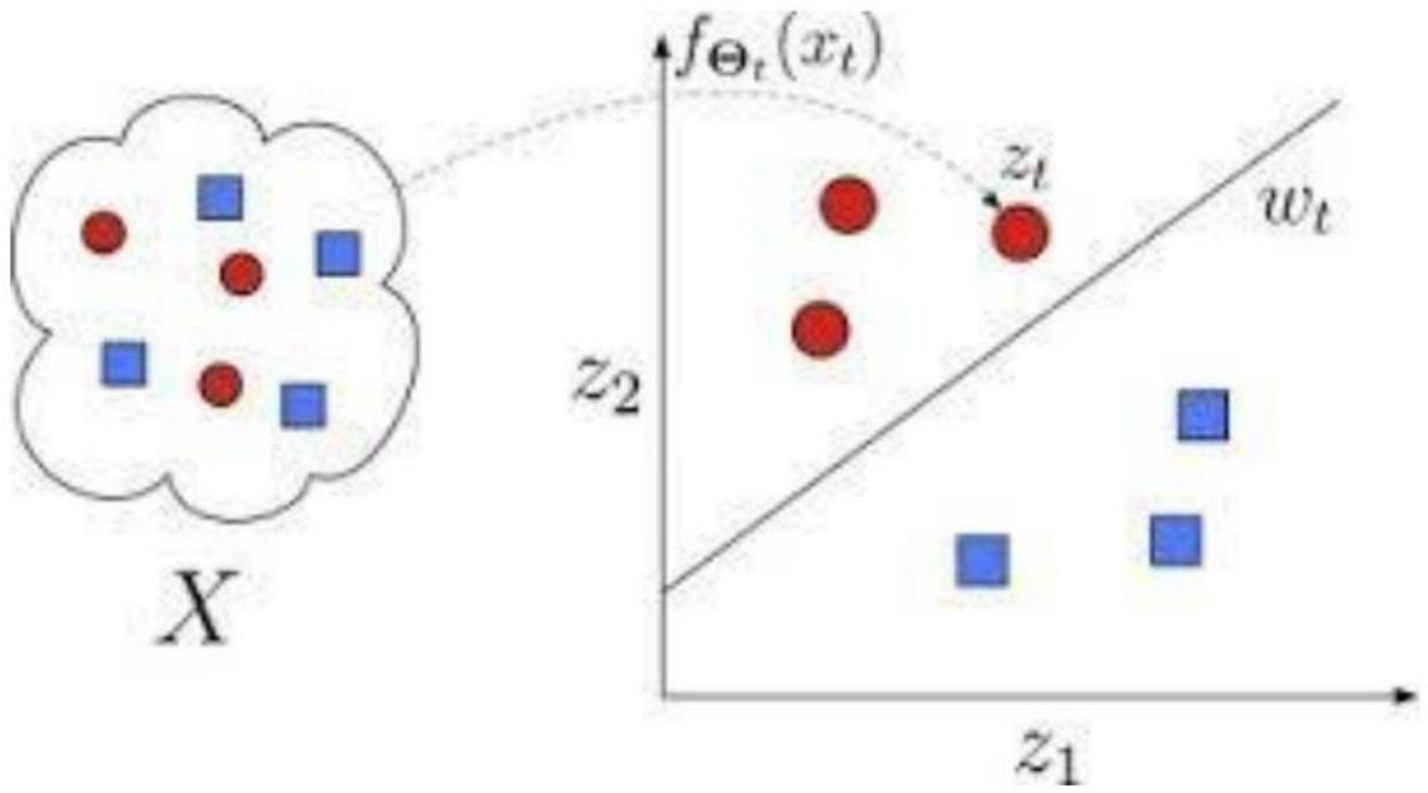

Fig 5.2.8

CHAPTER 6
IMPLEMENTATION

6.1 User Interface

We use Google Colaboratory as the model in this project. It allows you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. We can use Google Colabs like Jupyter notebooks. They are convenient because Google Colab hosts them, so we do not use any of our own computer resources to run the notebook. We can also share these notebooks so other people can easily run our code, all with a standard environment since it is not dependent on our own local machines. However, we might need to install some libraries in our environment during initialization. Now Go to the official website of Google Colaboratory and sign in to your colaboraty account

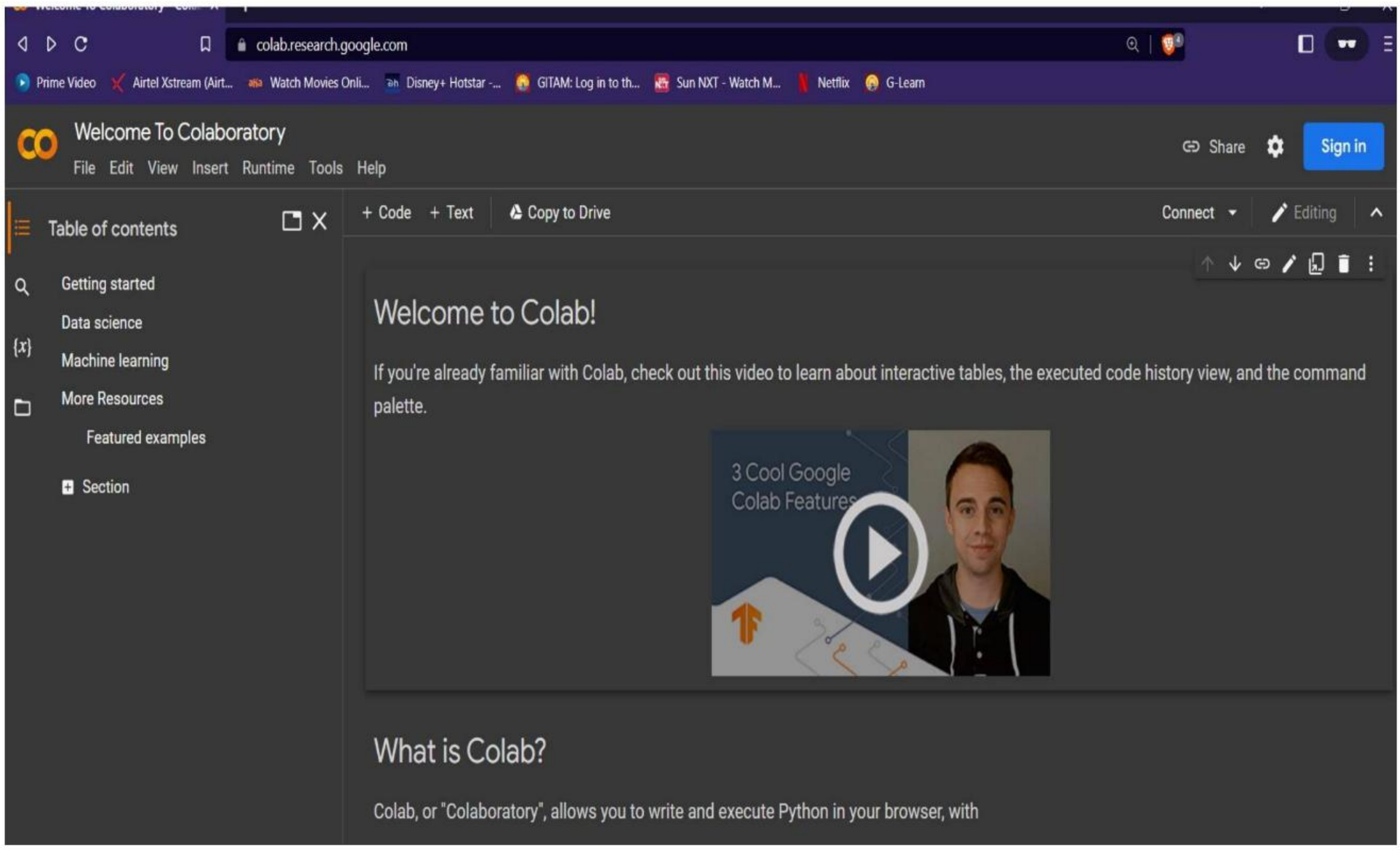

Fig 6.1

6.2 Execution

Create a File with name Mini project with the extension of ipynb

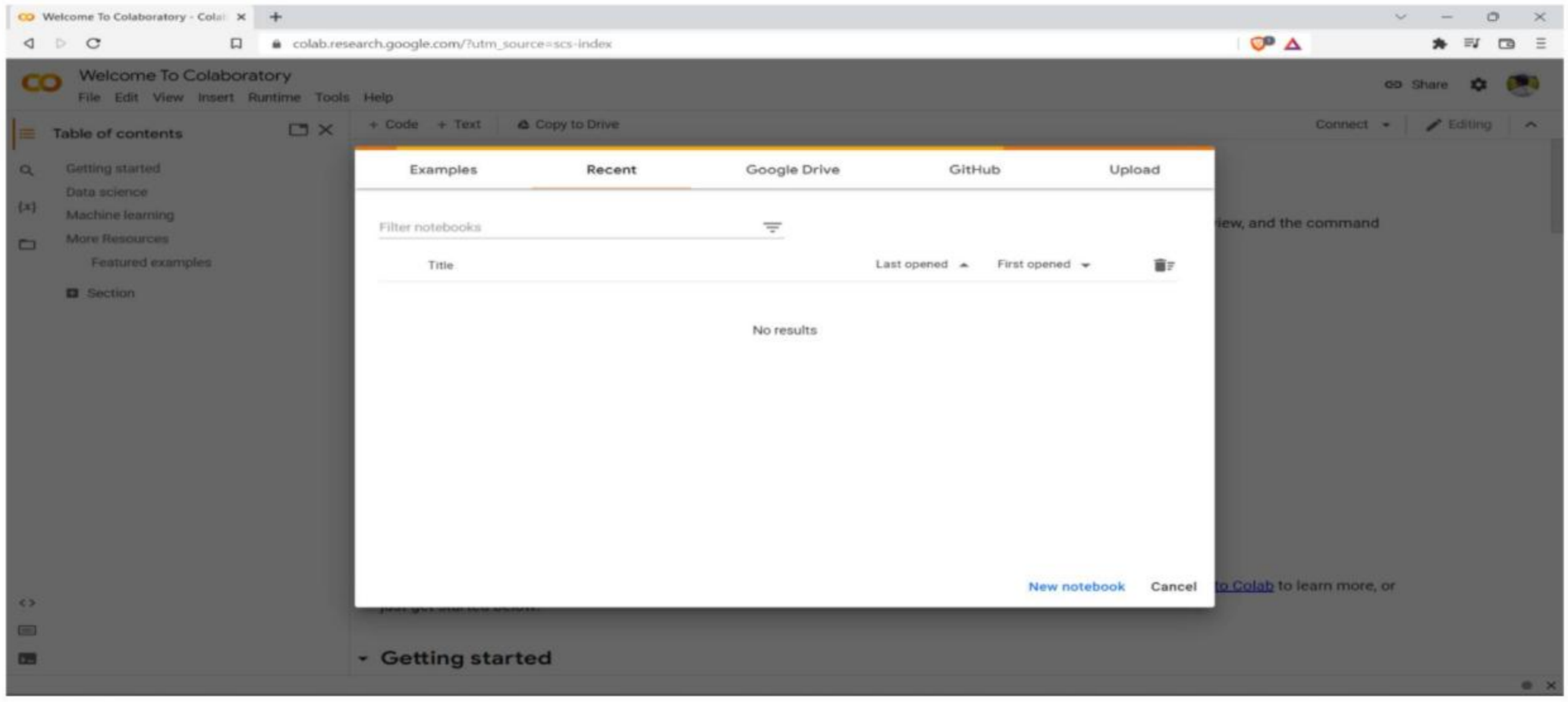

Fig 6.2.1

Here we implement the code for our model

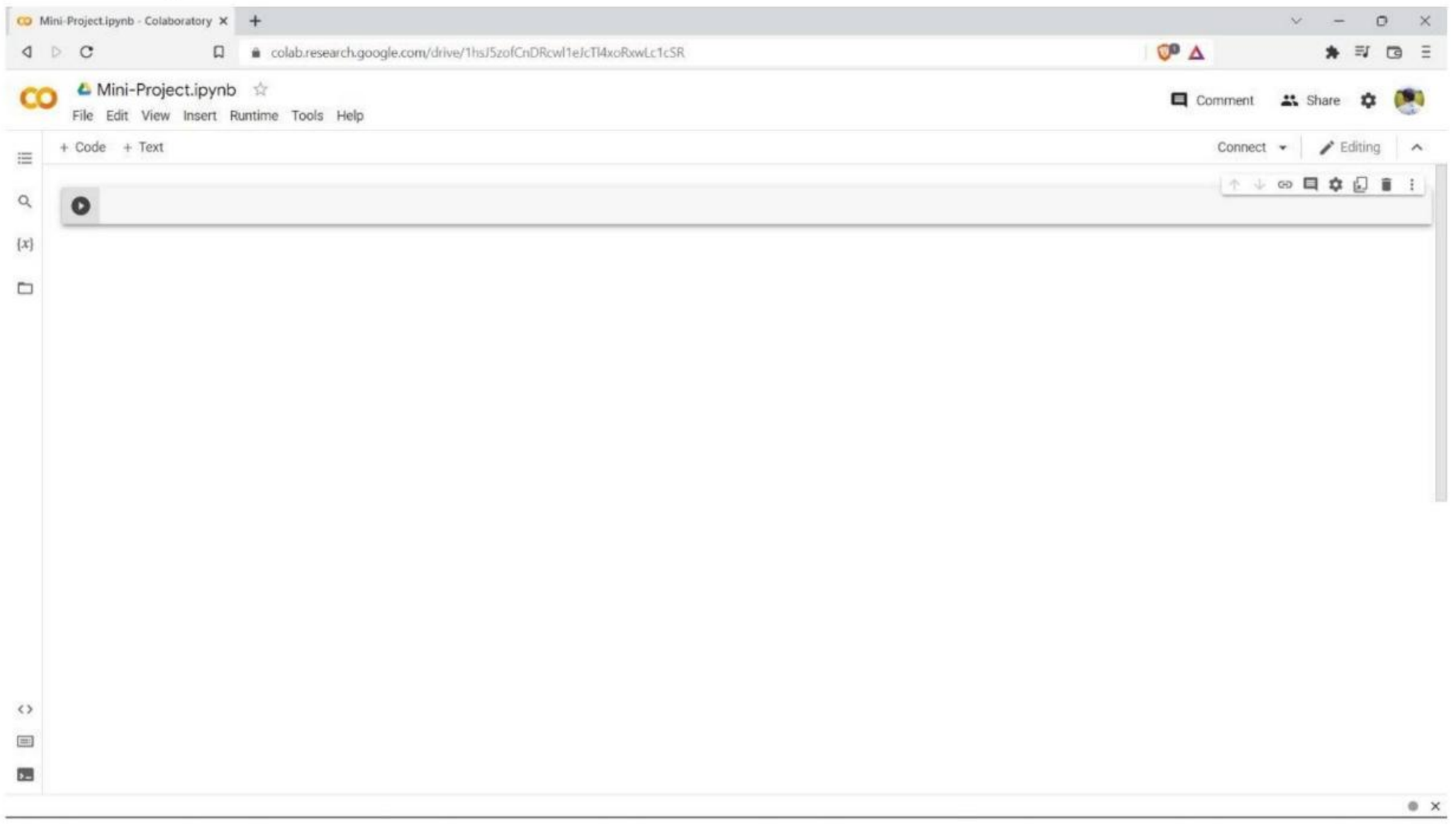

Fig 6.2.2

6.2.1Downloading the Datasets from Google

- Go to the google and search for Kaggle there we can get a link named Kaggle.
- Open the link Kaggle.com in your browser.
- The page will be shown as below in fig 6.2.3
- Now go to the left menu bar there we can find the datasets menu bar click on it.
- After opening the datasets, we can find lot of datasets in the website with several names and types like Spotify, Financial, sports, Indian headlines etc…
- For this project we considered news dataset which consists of two csv files
- The first file is news fake.csv and the second file is True.csv

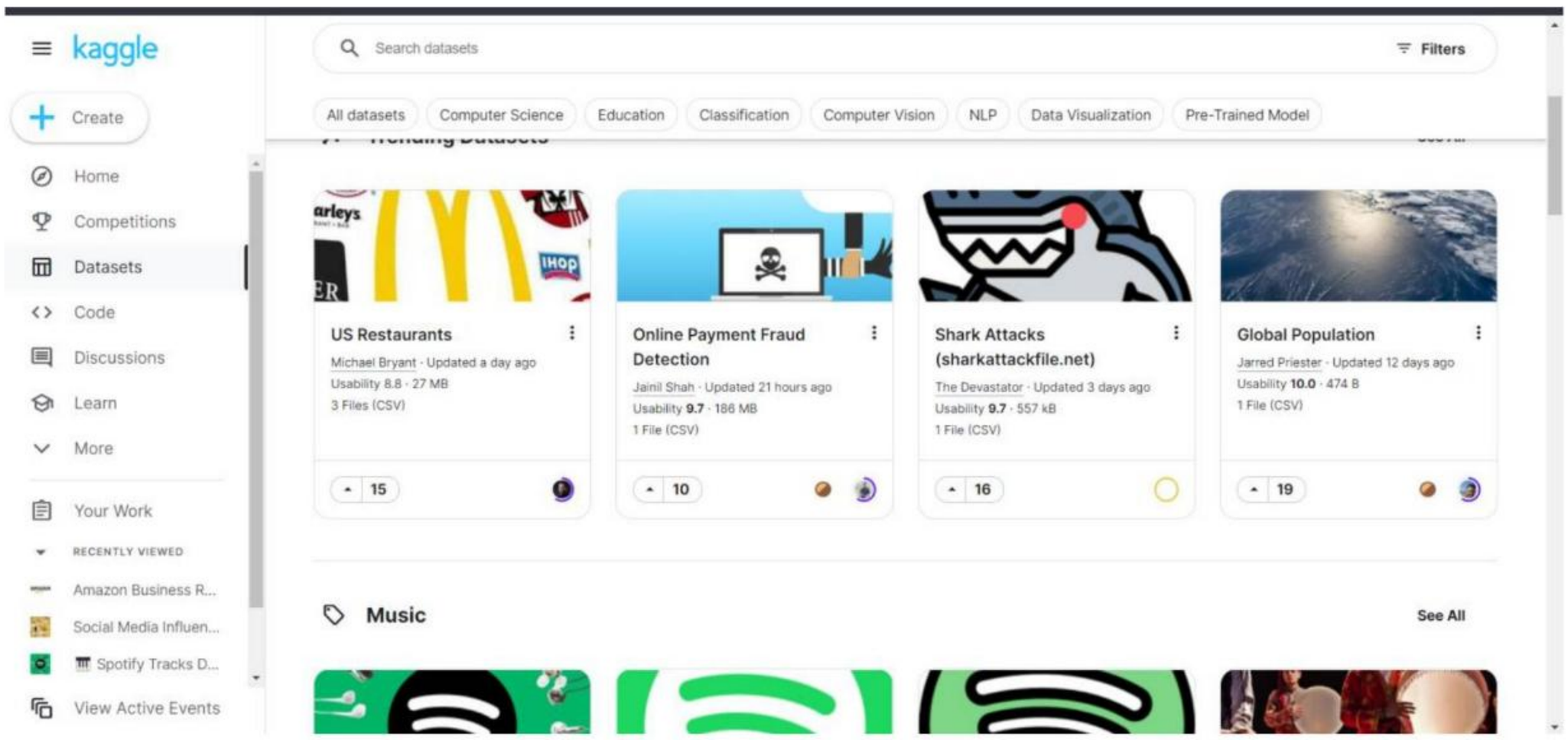

Fig 6.2.3

Fake.csv

- The left side dataset in the image is fake.csv which contains of 23503 rows and 4 columns .
- Its consists of the following headings as title, text, subject, date .
- The title consists of the fake news headlines.
- The text consists of the fake news article or news inside the headlines i.e main news.
- The subject consists of what type of news it consists .
- The date consists of when it was published.

True.csv

- The right side dataset in the image is fake.csv which contains of 21418rows and 4 columns .
- Its consists of the following headings as title, text, subject, date .
- The title consists of the True news headlines.
- The text consists of the True news article or news inside the headlines i.e main news.
- The subject consists of what type of news it consists .
- The date consists of when it was published.

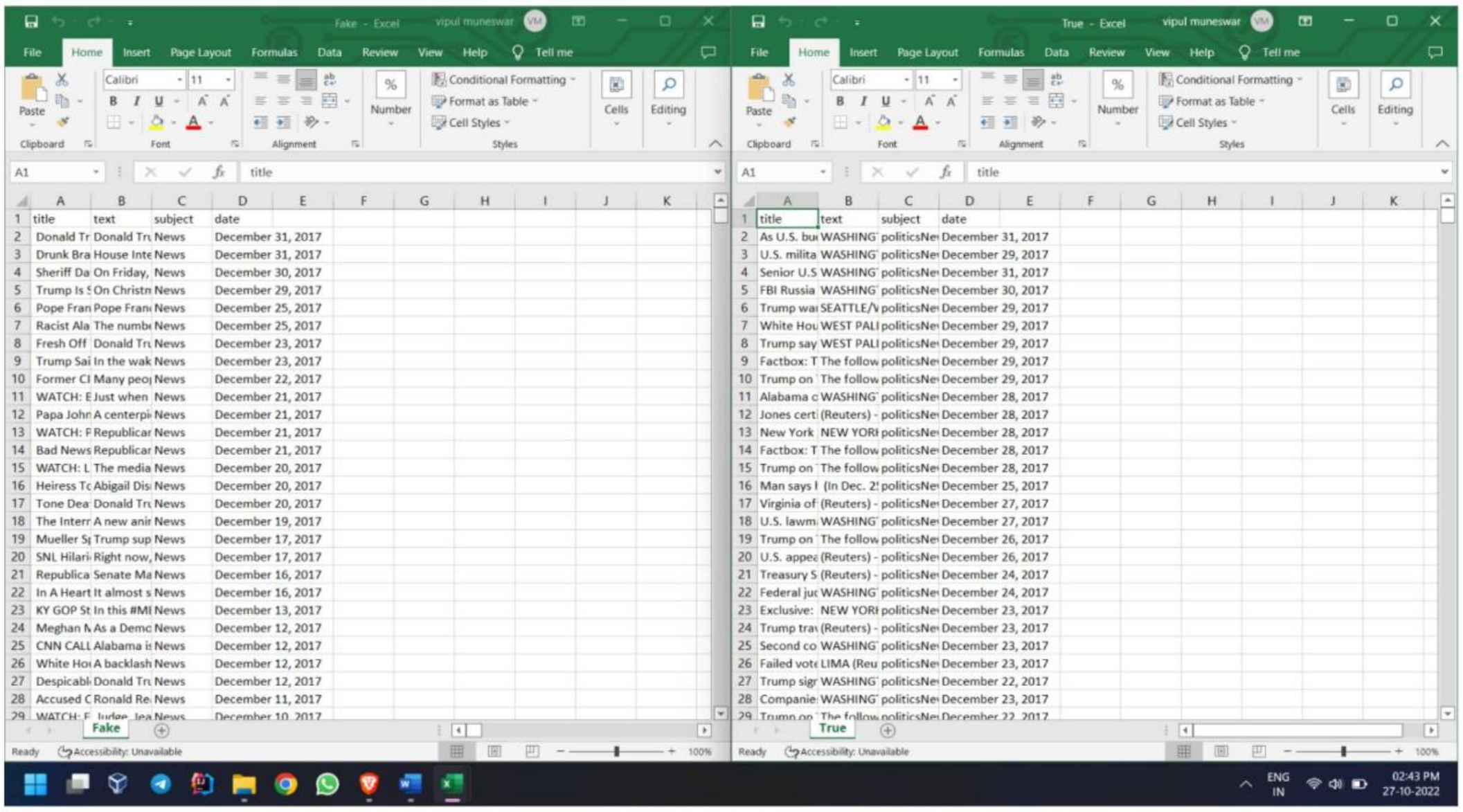

Fig 6.2.4

6.2.2 Importing Libraries and datasets

- First import the libraries of pandas as pd and numpy as np.
- Then import the sklearn libraries from feature extraction.text, import Tfidf vectorizer.
- And import feature_extraction, linear model, model_selection, preprocessing for preprocess the data.
- Then import accuracy score to calculate the accuracy value of the model.
- Later import train test split to train the model and test it.
- And finally import the pipeline for cross validating the data.
- Now import the dataset from the desktop or else at the left side of the colab we can get the upload option we can upload the news.csv and true.csv files to the colab and assign the dct parameter.
- Now give an extra column in the datasets as label at the fake.csv and true.csv to calculate the accuracy and this label is the bool values of datasets.
- Now merge the both datasets with pd.concat function, and assign it to data and get the data shape with the function data.shape now the final dataset size is “44898 X 5”

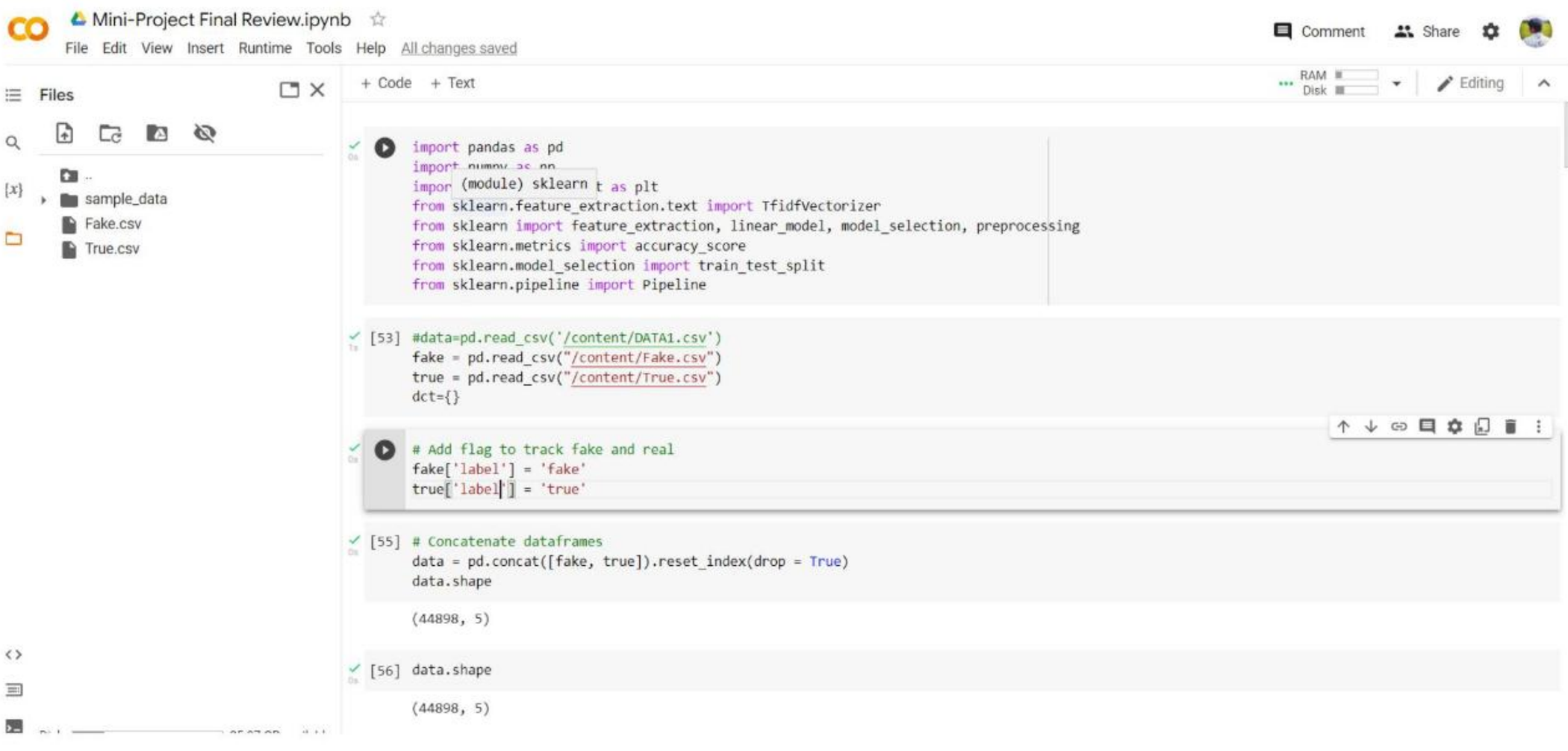

Fig 6.2.5

6.2.3 Dropping of data

- Now using the data.info function we get the information of the dataset
- It consist of the 44898 rows and 5 columns as shown.
- The column names are title, text, subject, date, and label which is an object.
- Now import the function shuffle to shuffle the data.
- we shuffle all the data inside the dataset using the function shuffle().
- To make the index as in its own position we use data.reset_index(drop=True) for keeping the index at the top.
- Now using this information we remove the unwanted columns from the dataset using datadrop function we remove title, subject, and date from the dataset as shown in the below fig.
- The data.head function will display the first five rows of the dataset.

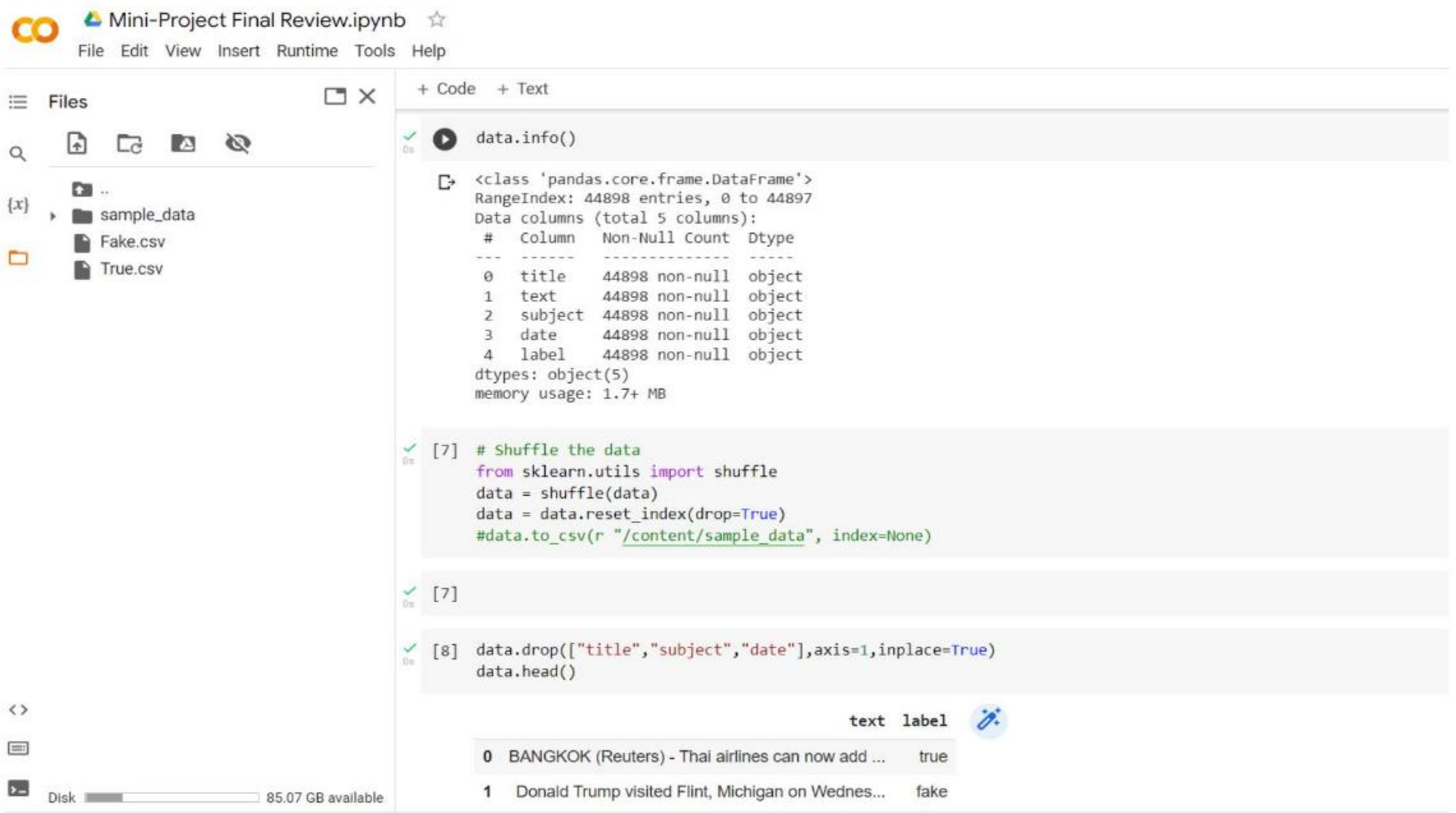

Fig 6.2.6

6.2.4 Preprocessing the dataset

- Now we preprocess the data by removing the punctuation marks from the dataset by using punctuation remover.
- From the dataset it will order the text in a list format and then characters in the list will check if there is a punctuation marks in the list they will be removed.
- After removal of punctuation marks join the list back to the text.
- This will be applied to each row of the dataset.
- Using the natural language toolkit we remove the stop words from the dataset.
- This preprocessing makes the dataset more easier for the model to give the best accuracy .

```
[9] import string

    def punctuation_remover(txt):
        lst = [char for char in txt if char not in string.punctuation]
        after_removing = ''.join(lst)
        return after_removing

    data['text'] = data['text'].apply(punctuation_remover)

    data.head()
```

	text	label
0	BANGKOK Reuters Thai airlines can now add fli...	true
1	Donald Trump visited Flint Michigan on Wednesd...	fake
2	The majority of Americans would likely find th...	fake
3	Julissa Arce who is now a Vice President at Go...	fake
4	The victim of the angry Bernie Sanders support...	fake

```
[10] import nltk
     nltk.download('stopwords')
     from nltk.corpus import stopwords as sw
     stop = sw.words('english')

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

     data.head()
```

Fig 6.2.7

6.2.5 Plotting confusion matrix

- Plotting of confusion matrix will helpful to see how well a classifier is doing by model. This function produces both 'regular' and normalized confusion matrices.
- The confusion matrix plots the model which the news in the dataset is varied in the range the matrix plot is build to give the understanding how many news is been able to predict by our model.
- The model is been separated the news which is predicted with blue color and the news which is not predicted in white color as shown in the results.
- The dataset will be send to the train test split in this 80% of data will be trained and 20% of data will be tested.
- The dataset gets divided into X_train, X_test , y_train and y_test. X_train and y_train sets are used for training and fitting the model.
- The X_test and y_test sets are used for testing the model.

```
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')
```

```
[12] X_train,X_test,y_train,y_test = train_test_split(data['text'], data.label, test_size=0.2, random_state=42)
```

Fig 6.2.8

CHAPTER 7

EXPERIMENTAL RESULTS

7.1 OUTPUT RESULTS FOR THE ALGORITHMS

7.1.1 The confusion matrix for the Decision Tree

- The accuracy for the decision tree is 99.68% which is the best accurate result for our model.
- Decision tree will change the its accuracy with the size of dataset we cannot say that this will give us the best accuracy at all time.
- In this confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

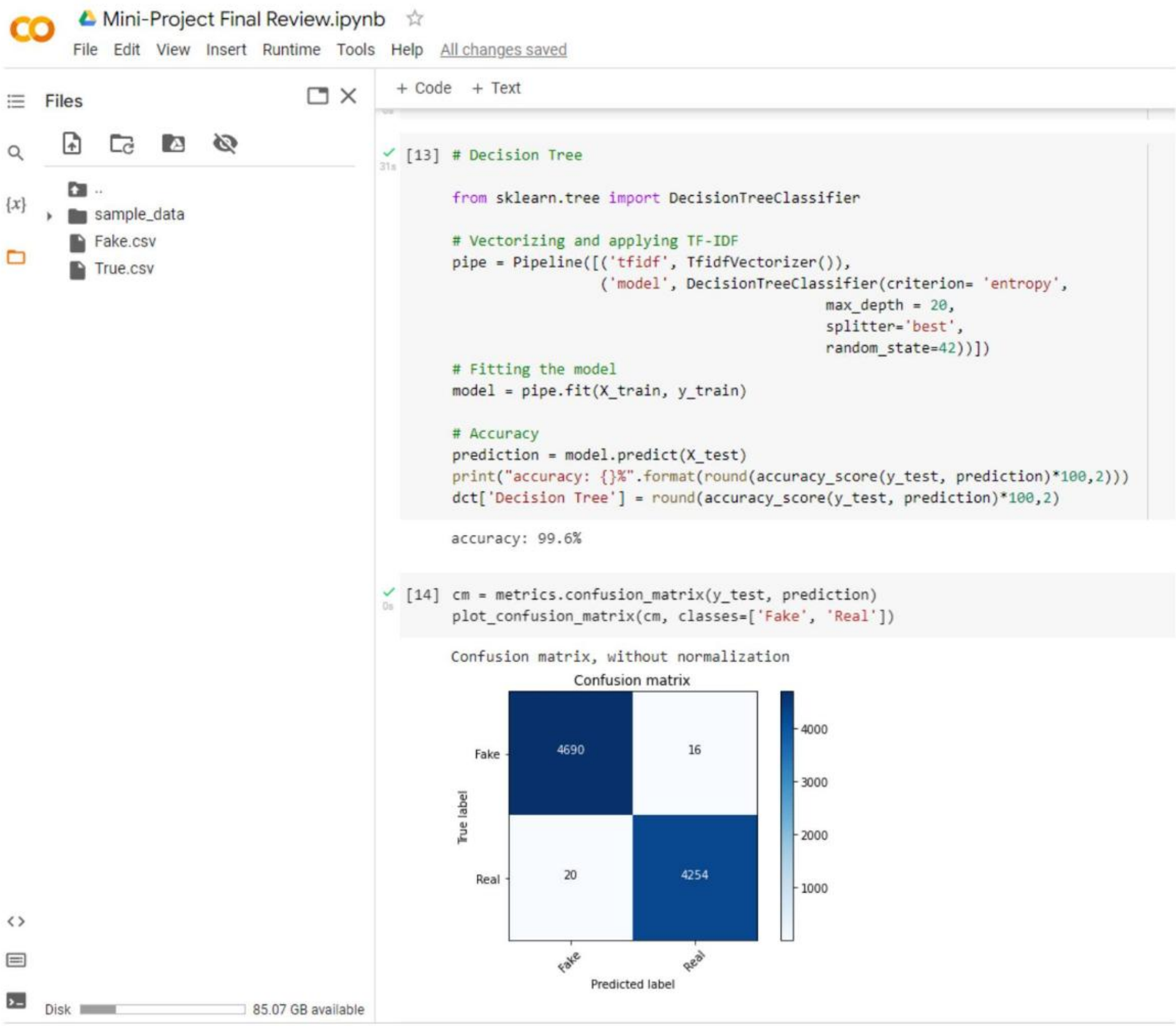

Fig 7.1.1

7.1.2 The confusion matrix for Logistic Regression Classifier

- The accuracy for the Logistic Regression Classifier is 99.57% which is the best accurate result for our model.
- In logistic regression in order to map the predicted values to probabilities, sigmoid function is used.
- This function maps any real value into another value between 0 to 1.
- This function has a non-negative derivative at each point and exactly one inflection point.
- In this confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

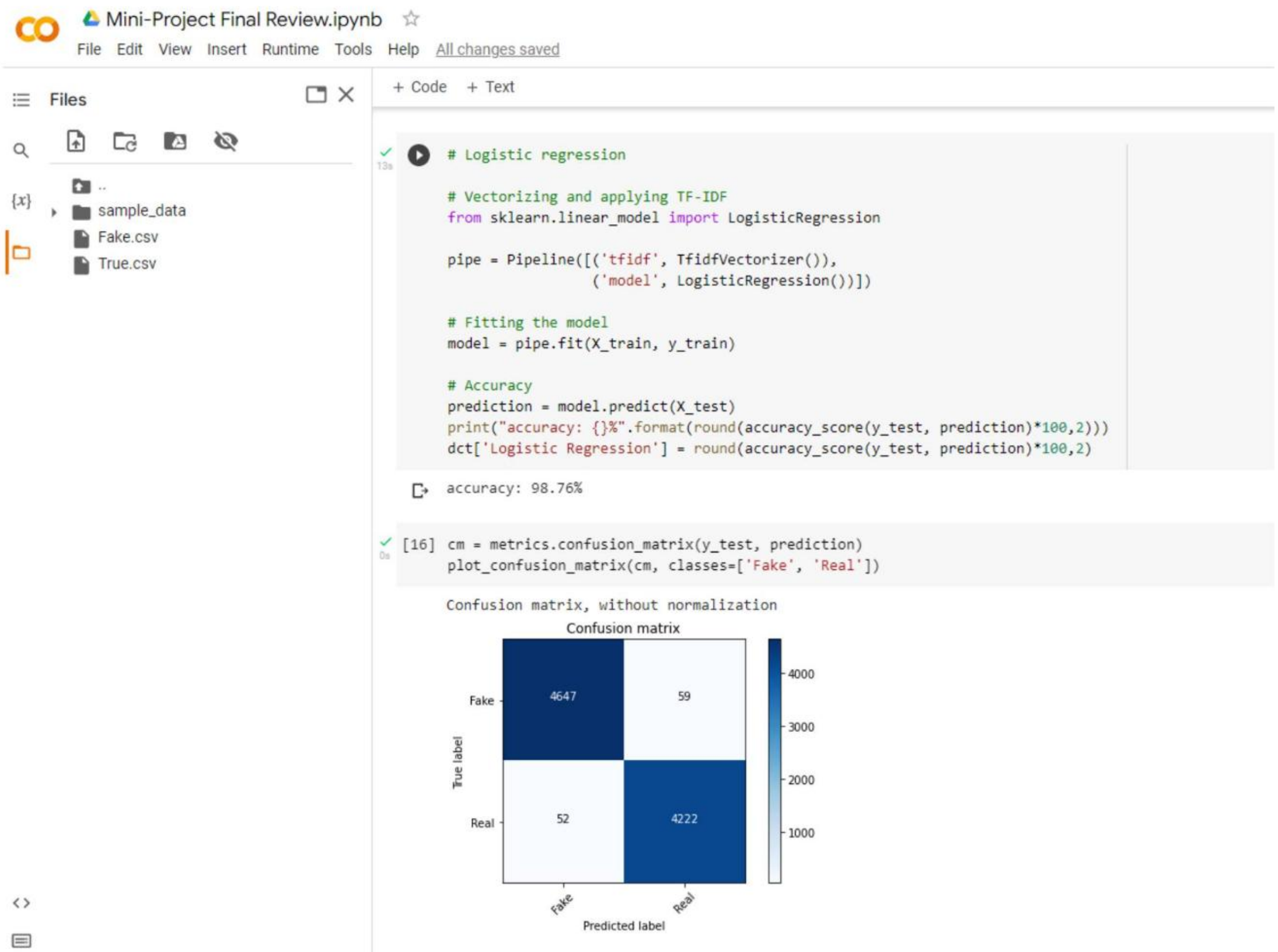

Fig 7.1.2

7.1.3 The confusion matrix for Passive aggressive classifier

- The accuracy for the Passive aggressive Classifier is 98.53% which is the best accurate result for our model.
- The Passive-Aggressive Classifier provides the good result by the speed of model and accuracy of model.
- The process results in a prediction object that can be used to calculate accuracy scores.
- In this plot confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

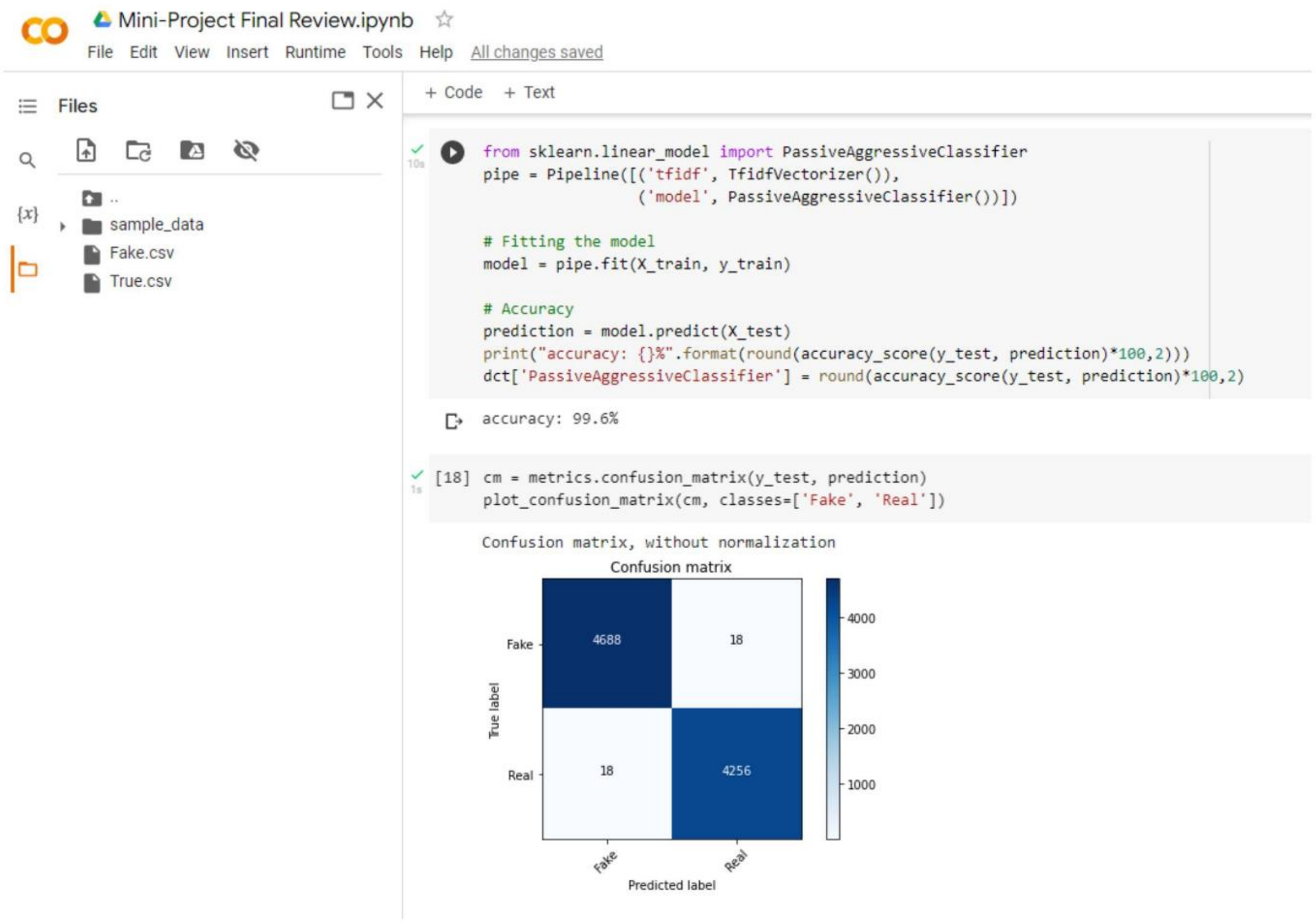

Fig 7.1.3

7.1.4 The confusion matrix for Random Forest Classifier

- The accuracy for the Random Forest Classifier is 98.79% which is the best accurate result for our model.
- The Random Forest Classifier is used to create a model that predicts the entropy of a dataset.
- The Random Forest Classifier provides the good result compared to other models like the speed of model and accuracy of model.
- In this confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

Fig 7.1.4

7.1.5 The confusion matrix for Naive Bayes Classifier

- The accuracy for the Naïve Bayes Classifier is 94.35% which is the best accurate result for our model.
- The arrays contain the training data for the Naïve Bayes Classifier.
- The prediction Function is then called on the NB_Classifier object to produce a prediction for each element in testing.
- The Naïve Bayes Classifier provides good results but compared to the other models its accuracy is low.
- In this confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

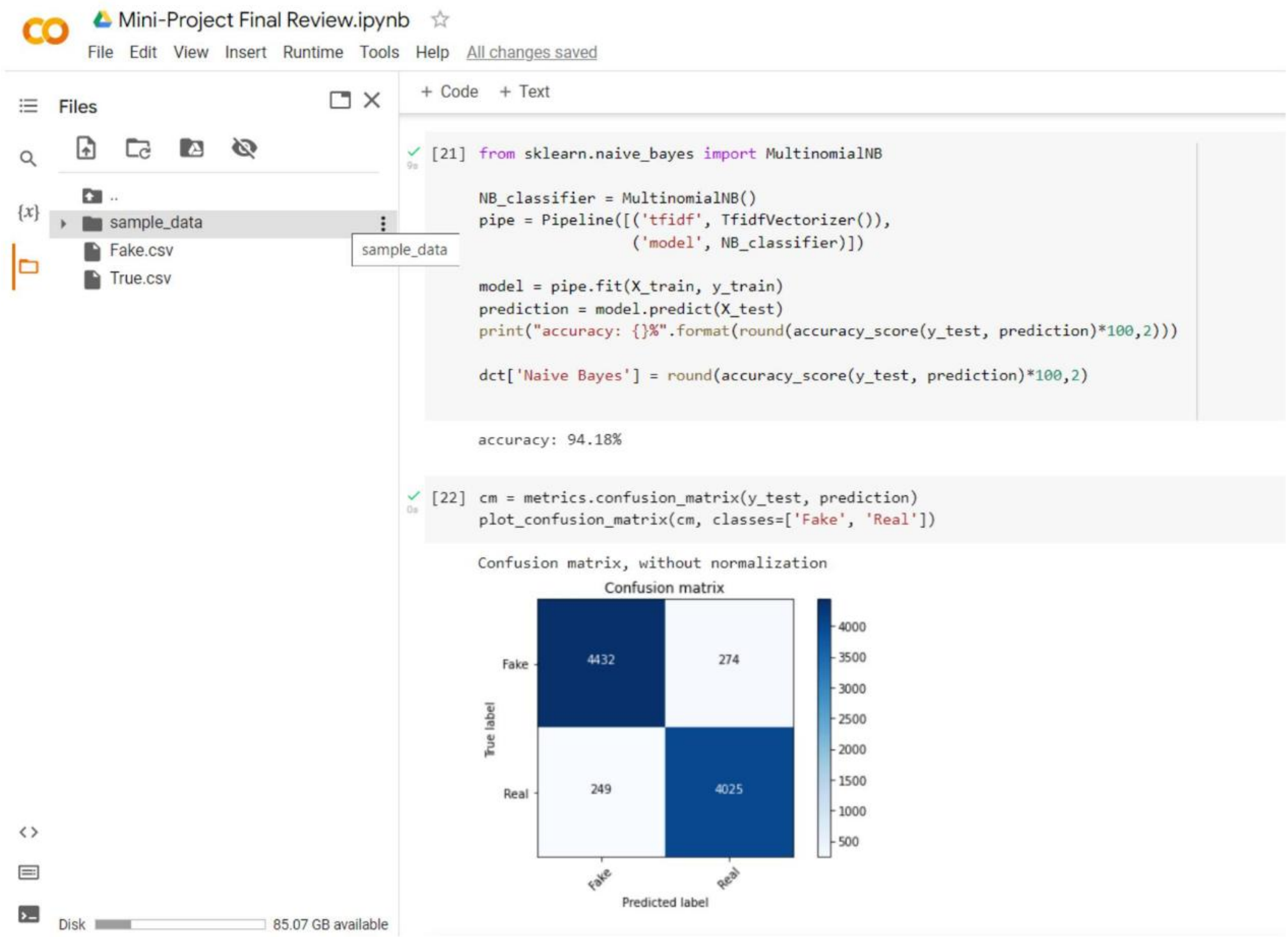

Fig 7.1.5

7.1.6 The confusion matrix for Support vector Machine classifier

- The accuracy for the Support Vector Machine Classifier is 99.51% which is the best accurate result for our model.
- The code calculates accuracy as follows: Accuracy = (Number of correct predictions) / (Total number of samples).
- The code above uses an SVM Classifier, which takes in two inputs: X_train and y_train.
- The Support Vector Machine Classifier provides the best result compared to other models the speed of model too slow.
- It will check each line of the dataset so that it took more time to give the accuracy result.
- In this confusion matrix we calculated the accuracy for this by using the accuracy formula with True Positive, True Negative, False Positive, False Negative.
- The accuracy is then calculated and displayed as a percentage as shown below.

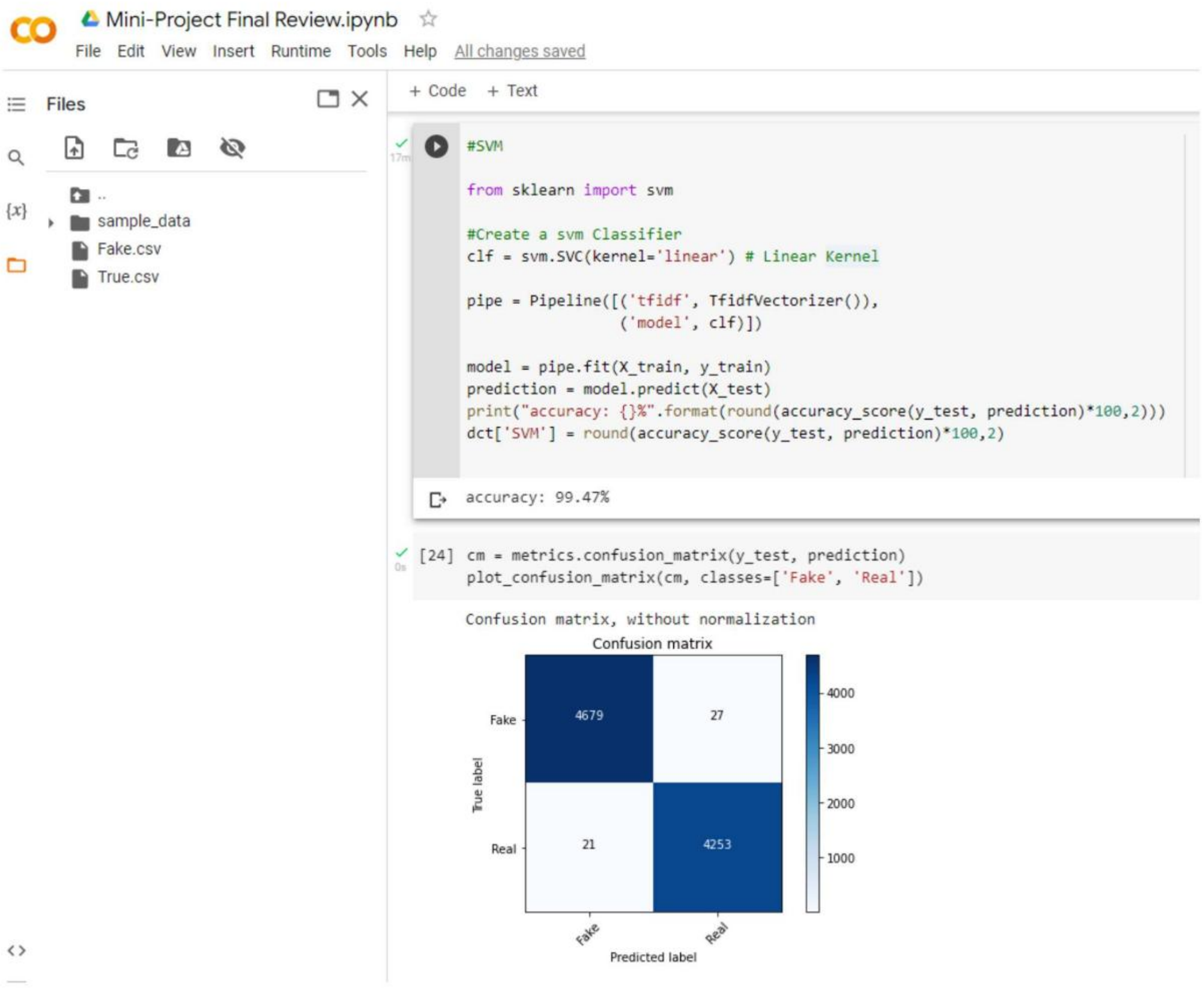

Fig 7.1.6

7.2 PLOT GRAPH

- It is plot graph drawn by using "matplotlib" Library which is imported from Python language.
- so from this graph we can see the best accuracy out of various algorithms used. but the accuracy for the algorithms will be not same for every time. The accuracy will gets changed for different datasets.
- By observing the plot graph, we can see that:
    - Decision Tree have 99.6% accuracy
    - Logistic regression has 98.76% accuracy
    - Passive Aggressive have 99.6% accuracy
    - Random Forest have 98.69% accuracy
    - Naive Bayes have 94.18% accuracy
    - SVM have 99.47% accuracy

```
plt.figure(figsize=(14,8))
plt.bar(list(dct.keys()),list(dct.values()))
plt.ylim(90,100)
plt.yticks((90,91,92,93,94,95,96,97,98,99,100))
```

```
([<matplotlib.axis.YTick at 0x7fa0a0ee6650>,
  <matplotlib.axis.YTick at 0x7fa0a0975b50>,
  <matplotlib.axis.YTick at 0x7fa0a0984c90>,
  <matplotlib.axis.YTick at 0x7fa0a0d1e3d0>,
  <matplotlib.axis.YTick at 0x7fa0a0cf6350>,
  <matplotlib.axis.YTick at 0x7fa0a0cf6890>,
  <matplotlib.axis.YTick at 0x7fa0a0cf6e10>,
  <matplotlib.axis.YTick at 0x7fa0a1719390>,
  <matplotlib.axis.YTick at 0x7fa0a17198d0>,
  <matplotlib.axis.YTick at 0x7fa0a1719e10>,
  <matplotlib.axis.YTick at 0x7fa0a171d350>],
<a list of 11 Text major ticklabel objects>)
```

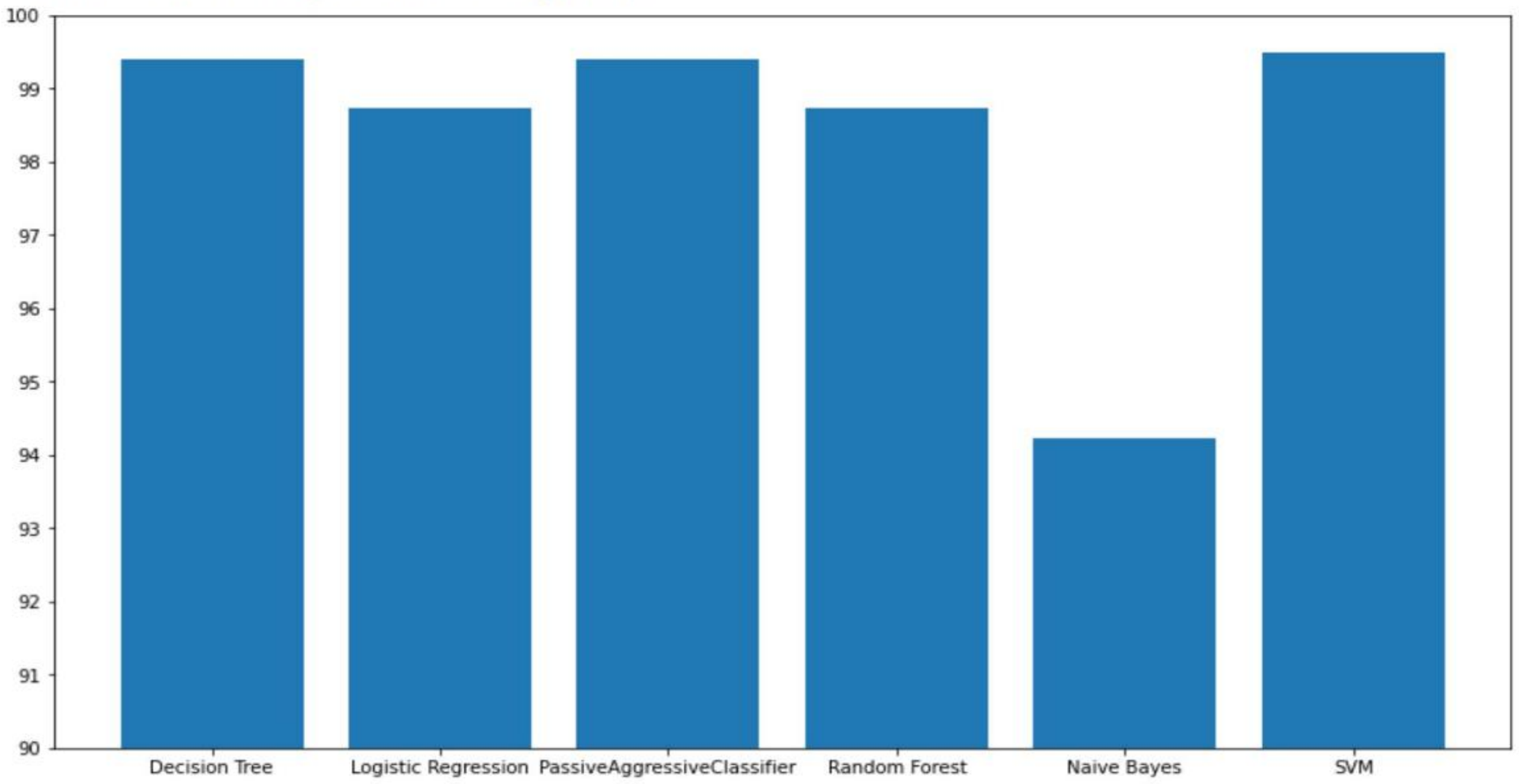

Fig 7.2

CHAPTER 8
CONCLUSION

With the help of these various machine learning algorithms such as Logistic Regression, Tfidf Vectorizer, Decision Tree, Random Forest, SVM classifier, Naive bayes classifiers, Passive aggressive Classifiers etc.. We have developed a model to predict the news we took is a "True news" or "Fake news". Moreover, each classifier's results are successful. Some of them give the best results which have more accuracy, some of them have low accuracy. We are choosing the best of these models so that the results of the model will have more accuracy and give the more accurate results for the models. As we can conclude that if we use the lower size data we get the accuracy results low as we use the larger size data set we get the results with more accuracy with these data we can have the Decision tree with higher accuracy as per the present dataset.it will change the accuracy according to the dataset, the second highest accuracy shown in the plot is SVM classifier but it takes more time than the other algorithms so we consider the third highest accuracy which gives the good results for our model as show in the plot diagram we consider Passive aggressive classifier as the best algorithm for our model

CHAPTER 9
FUTURE WORK

- In the future this Project may be implemented with some more algorithms for checking the best accuracy score better than current algorithms.
- We will be testing the model with different types of data sets.
- To gain a better understanding, we will use the data visualization process to compare the various models.

CHAPTER 10
REFERENCES

- [1] The Journal of Supercomputing, vol. 76, no.7, pp.4802–4837, 2020. K.S.Adewole, T.Han, W.Wu, H.Song, and A.K.Sangaiah. Twitter spam account detection based on clustering and classification methods.
  https://link.springer.com/article/10.1007/s11227-018-2641-x

- [2] Soft Computing, vol. 24, no. 5, pp. 3475–3498, 2020. M.Z.Asghar, A.Ullah, S.Ahmad, and A.Khan. Opinion spam detection framework using hybrid classification scheme.
  https://link.springer.com/article/10.1007/s00500-019-04107-y

- [3] International Journal of Multimedia Information Retrieval, vol. 7, no. 1, pp.71–86, 2020. C.Boididou, S.Papadopoulos, M.Zampoglou, L.Apostolidis, O.Papadopoulou, and Y.Kompatsiaris. Detection and visualization of misleading content on Twitter.
  https://link.springer.com/article/10.1007/s13735-017-0143-x

- [4] International Journal of Machine Learning and Cybernetics, vol. 10, no. 8, pp. 2143–2162, 2021. K.Dhingra and S.K.Yadav. Spam analysis of big reviews dataset using Fuzzy Ranking Evaluation Algorithm and Hadoop.
  https://link.springer.com/article/10.1007/s13042-017-0768-3

- [5] IJERT-Fake News Detection using Machine Learning Algorithms. Uma Sharma, Sidarth Saran, Shankar M. Patil.
  https://www.academia.edu/download/66254531/fake_news_detection_using_machine_IJERTCONV9IS03104.pdf

- [6] The SciPy Community. "NumPy for Matlab users". Retrieved 2 February 2017.

☞ [7]Brooks, Gabriel. "Introduction to Python Pandas for Beginners". Almabetter.com. Retrieved 24 October 2020.

☞ [8]"NumFOCUS Sponsored Projects". NumFOCUS. Retrieved 2021-10-25.
https://numfocus.org/sponsored-projects

☞ [9] "General Python FAQ — Python 3.9.2 documentation". docs.python.org. Archived from the original on 24 October 2012. Retrieved 28 March 2021.
https://pypi.python.org/pypi/PyAIML

☞ [10] Datasets used in this project
https://www.kaggle.com/datasets