PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

CHINTHALACHERUVU GOVARDHAN REDDY

(Reg.No.39110235)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119,TAMILNADU

NOVEMBER 2021

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119
www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **CHINTHALACHERUVU GOVARDHAN REDDY (Reg. No. 39110235)** who carried out the project entitled "**TEACHING ASSISTANCE EVALUATION**" under my supervision from September 2021 to November 2021.

Internal Guide

Ms.Sankari M,M.E.,(Ph.D),

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **CHINTHALACHERUVU GOVARDHAN REDDY** (Reg. No 39110235)hereby declare that the Professional Training Report on "**TEACHING ASSISTANCE EVALUATION**" done by me under the guidance Ms.**SANKARI M,M.E.,(Ph.D),** at **Sathyabama institute of science and technology** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE:**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA., ME., Ph.D**., Dean, School of Computing **Dr. L. LAKSHMANAN,ME., Ph.D**., **Dr.S.Vigneshwari M.E., Ph.D.,** Head of the Department, Dept. of Computer Science and Engineering for providing me necessarysupport and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Sankari M, M.E.,(Ph.D),**for her valuable guidance ,suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

Nowadays, most of the academic institutes facing a low-quality problem in the educational field. One of these factors is an educational student achievement and staff teaching quality. This study presents an efficient framework for assessment and prediction of teachers' performance in academic institutes using Random Forest Classifier Algorithm(RFCA). The proposed framework designed to predict the performance quality level of teachers in order to improve the learning outcomes. The prediction model was tested effectively using the TA dataset. The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. The scores were divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Nowadays, most of the academic institutes facing a low-quality problem in the educational field. One of these factors is an educational student achievement and staff teaching quality. Some studies had been done to engorge the students to improve their academic achievement, but still, the problem of the teaching quality need to be improved especially in the practical parts that normally performed by the Teaching Assistant staff and Lecturer. Teaching Assistants (TAs) are persons who help lecturer or professors with instructional tasks [1] and [2]. TAs can be in university level, secondary school level or elementary school level. Evaluation Teaching Assistant system introduced in agreement with the Rules as a recommendation from the Evaluation and Standards Uniform Performance Teacher as one of the improvement factor of learning quality [3]. Performance evaluation system assists the district's scholar attainment aims and the universities as well. This program offers us efficiency instructive environment in the education institutes. It is permitting for ingenuity and separate instructor initiative. The efficiency of the teachers may be effect in some courses because of teaching language, study time, the number of students in each classroom, etc. The aim is to assist the incessant growing and development progress of every instructor by analyzing, monitoring, and employing relevant information amassed within a model of meaningful reaction to enhance the progress of academic scholar and instructor efficiency. The significant objective ofthis study is to the proposal a solid evaluation system for TA using on Random Forest Classifier Algorithm (RFCA). The proposed model applied the RFCA as a possible solution to study and analyze the quality of teaching assistant teachers toimprove their skills to growth the educational achievement level.

1.2 MOTIVATION

Nowadays to survive in this busy and Machine oriented world studies plays an important role in the lives of Human Beings .However the people are spending a lot of money for the purpose of Education but the people are not reaching the expected level of Education due to lack of proper teachers .So in order to overcome the problem faced by the people we need to develop a model of machine learning using Random Forest Classifier Algorithm (RFCA) by using the data set of TA Evaluation for predicting the performance of a Teacher.

1.3 OBJECTIVE

In this paper, we have formulated the problem as a multi-class classification problem. Different techniques are available to solve this problem. After analyzing the research works on classification problems and prediction of the teacher's performances, it was found that Random Forest (RF) classifier was used in many works and its performance was better than other methods. So, our aim was to implement this method to predict teacher's' performance by using TA Dataset. The random forest consists of many decision trees. Each individual decision tree gives one class prediction and the class with the highest number of votes becomes the model's final prediction.

1.3.1 PROPOSED SYSTEM

As mentioned above, 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.

1.3.2 ADVANTAGES OF PROPOSED SYSTEM

a) It overcomes the problem of overfitting by averaging or combining the results of different decision trees.

b) Random forests work well for a large range of data items than a single decision tree does.

c) Random forest has less variance then single decision tree.

d) Random forests are very flexible and possess very high accuracy.

e) Scaling of data does not require in random forest algorithm. It maintains accuracy even after providing data without scaling.

f) Random Forest algorithms maintains good accuracy even a large proportion of the data is missing

CHAPTER 2
TECHNOLOGIES LEARNT

2.1 PYTHON

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following –

a) Machine Learning
b) GUI Applications (like Kivy, Tkinter, PyQt etc.)
c) Web frameworks like Django (used by YouTube, Instagram, Dropbox)
d) Image processing (like OpenCV, Pillow)
e) Web scraping (like Scrapy, BeautifulSoup, Selenium)
f) Test frameworks
g) Multimedia

2.1.1 ADVANTAGES OF PYTHON

a) Extensive Libraries
b) Extensible
c) Embeddable
d) Improved Productivity
e) IOT Opportunities
f) Readable
g) Object-Oriented
h) Free and Open Source

i) Portable

j) Interpreted

2.1.2 DISADVANTAGES OF PYTHON

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

a) Speed Limitations

b) Weak in Mobile Computing and Browsers

c) Design Restrictions

d) Underdeveloped Database Access layers

e) Simple

2.2 MACHINE LEARNING

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

2.2.1 CHALLENGES IN MACHINE LEARNING

While Machine Learning is rapidly evolving, making significant strides with cyber security and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are

(a) **Quality of data**

Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

(b) **Time-Consuming task**

Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

(c) **Lack of specialist persons**

As ML technology is still in its infancy stage,availability of expert resources is a tough job.

(d) **No clear objective for formulating business problems**

Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

(e) **Issue of over fitting & under fitting**

If the model is over fitting or under fitting, it cannot be represented well for the problem.

(f) **Curse of dimensionality**

Another challenge ML model faces is too many features of data points. This can be a real hindrance.

(g) **Difficulty in deployment**

Complexity of the ML model makes it quite difficult to be deployed in real life.

2.2.2 TYPES OF MACHINE LEARNING

Supervised Learning

This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

Unsupervised Learning

This involves using un labelled data and then finding the
underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

Semi-supervised Learning

This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

Reinforcement Learning

This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

2.2.3 ADVANTAGES OF MACHINE LEARNING

Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwaree, they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

Wide Applications

You could be an e tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

2.2.4 DISADVANTAGES OF MACHINE LEARNINGData Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

2.2.5 APPLICATIONS OF MACHINE LEARNING

These are the applications :

a) Emotion analysis
b) Sentiment analysis
c) Error detection and prevention
d) Weather forecasting and prediction
e) Stock market analysis and forecasting
f) Speech synthesis
g) Speech recognition
h) Customer segmentation
i) Object recognition
j) Fraud detection
k) Fraud prevention
l) Recommendation of products to customer in online shopping

2.3 MODULES USED IN PROJECT

NUMPY

Numpy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

a) A powerful N-dimensional array object
b) Sophisticated (broadcasting) functions
c) Tools for integrating C/C++ and Fortran code
d) Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

PANDAS

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging And preparation.It had very little contribution towards data analysis. Pandas solved this problem.Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of dataload, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc

MATPLOTLIB

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnailgallery.For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

SCIKIT – LEARN

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

2.4 INSTALL PYTHON STEP-BY-STEP IN WINDOWS

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3. Before you start with the installation process of Python. First, you need to know about your system Requirements. Based on your system type i.e. operating system and based

processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.Download the Correct version into the system.

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link**: https://www.python.org**

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system. Here we will install Windows x86-64 web-based installer. Here your first part regarding.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation.

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to path.

Step 3: Click on Install now, after the installation is successful. Click on Close.

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS

3.1 ALGORITHMS USED IN THIS PROJECT

3.1.1 RANDOM FOREST CLASSIFIER ALGORITHM

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm

Understanding decision trees

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.

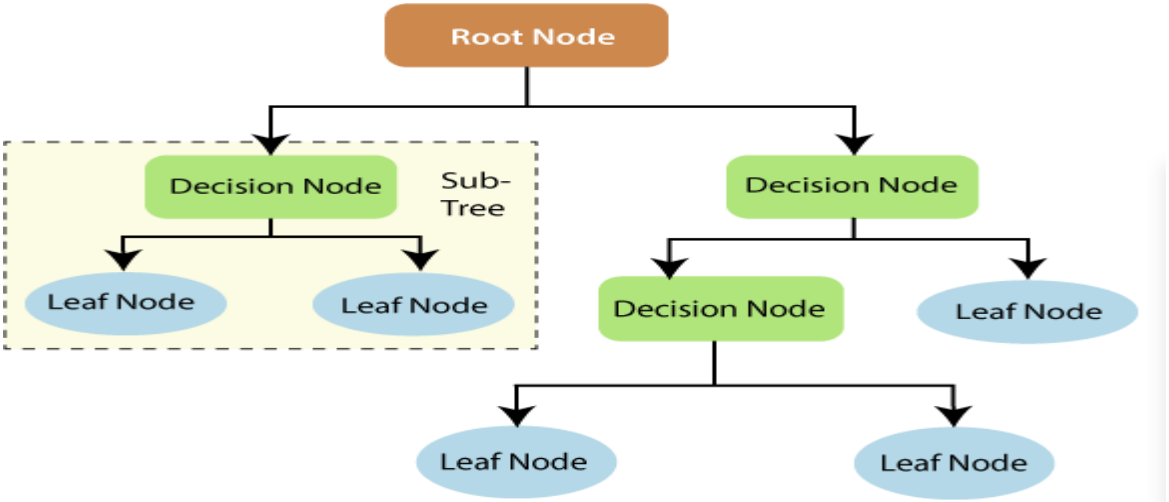

Fig 3.1 Decision tree

The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An overview of these fundamental concepts will improve our understanding of how decision trees are built.

Entropy is a metric for calculating uncertainty. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables.

The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees.

Let's take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram.

The root node and decision nodes of the decision represent the features of the

phone mentioned above. The leaf node represents the final output, either buying or not buying. The main features that determine the choice include the price, internal storage, and Random Access Memory (RAM). The decision tree will appear as follows.

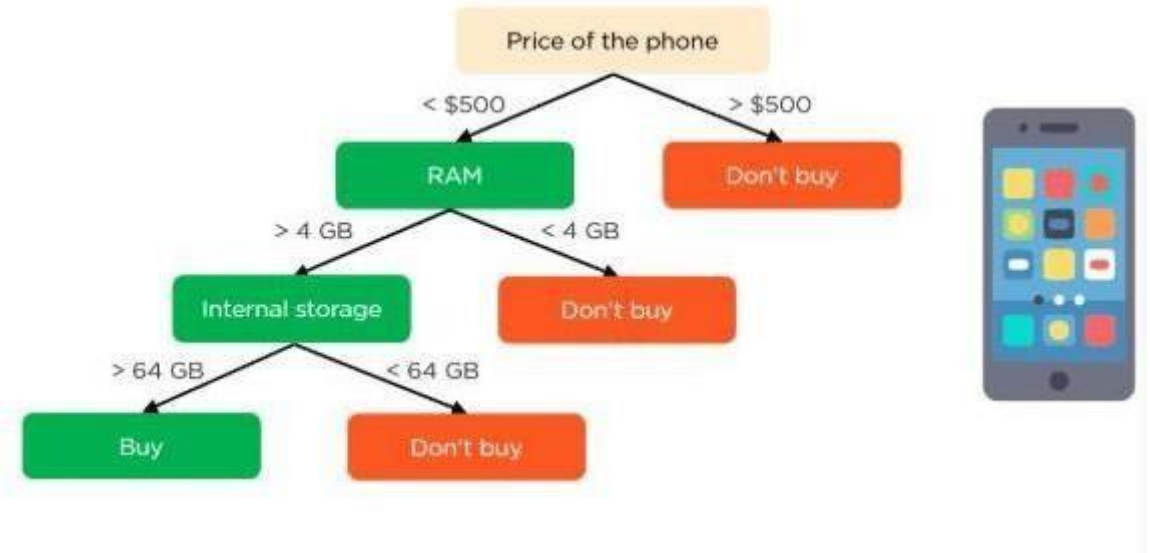

Fig 3.2 Decision Tree Example

Applying decision trees in random forest

The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction.

Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let's assume we have only four decision trees. In this case, the training data comprising the phone's observations and features will be divided into four root nodes.

The root nodes could represent four features that could influence the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees.

The outcome chosen by most decision trees will be the final choice. If three trees predict buying, and one tree predicts not buying, then the final prediction will be buying. In this case, it's predicted that the customer will buy the phone.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

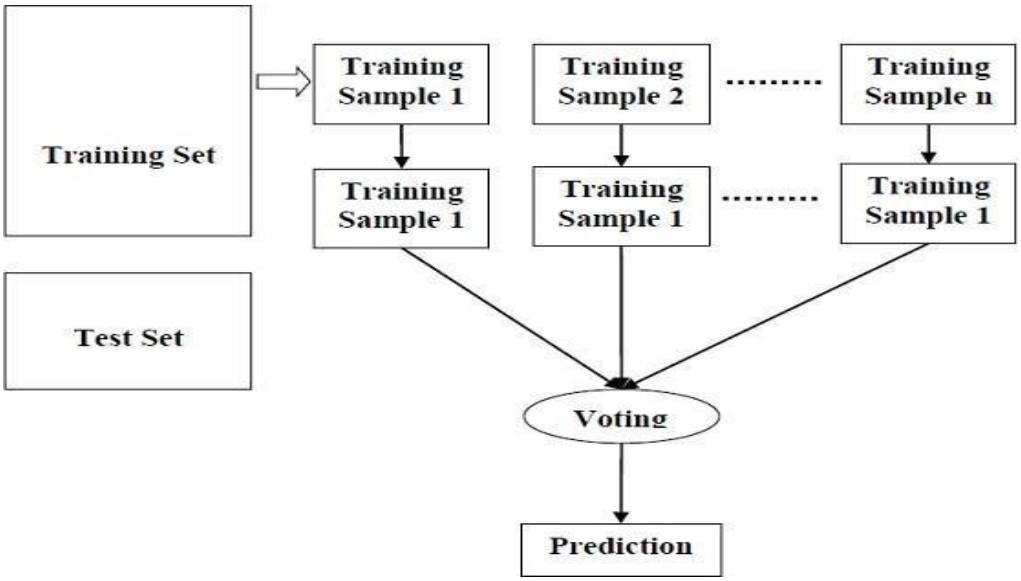

Fig 3.3 Random Forest Classifier

Let's take an example of a training dataset consisting of various fruits such as bananas, apples, pineapples, and mangoes. The random forest classifier divides this dataset into subsets. These subsets are given to every decision tree in the random forest system. Each decision tree produces its specific output. For example, the prediction for trees 1 and 2 is apple.

Another decision tree (n) has predicted banana as the outcome. The random forest classifier collects the majority voting to provide the final prediction. The majority of the decision trees have chosen apple as their prediction. This makes the classifier choose apple as the final prediction.

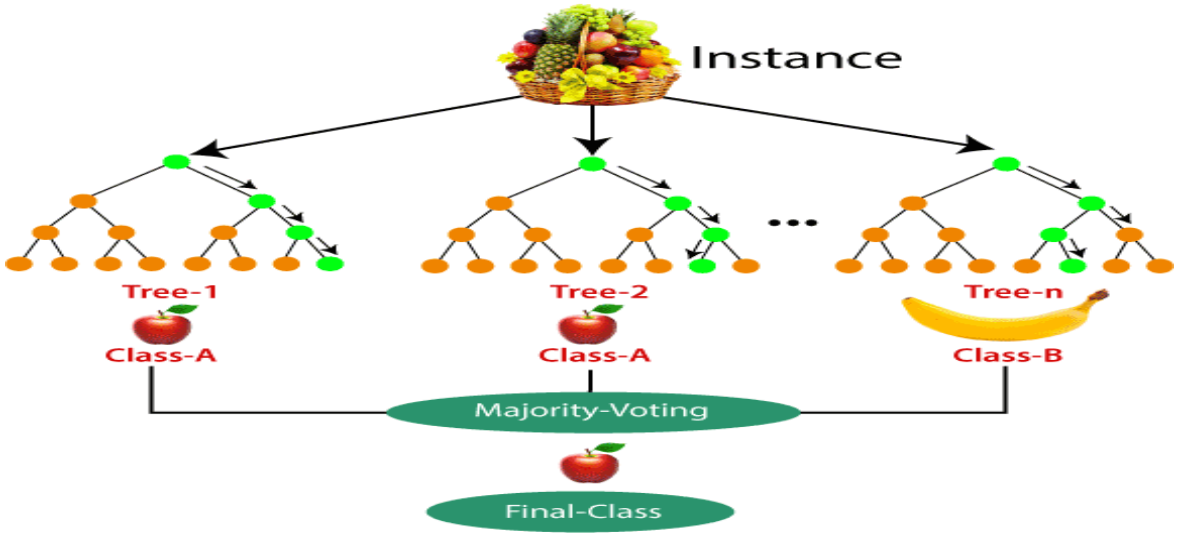

Fig 3.4 Random Forest Classifier Example

3.2 SYSTEM SPECIFICATION

3.2.1 SOFTWARE REQUIREMENTS

Functional requirements for a secure cloud storage service are straight forward:

a) The service should be able to store the user's data;

b) The data should be accessible through any devices connected to the Internet;

c) The service should be capable to synchronize the user's data between multiple

d) devices (notebooks, smart phones, etc.);

e) The service should preserve all historical changes (versioning);

f) Data should be shareable with other users;

g) The service should support SSO; and

h) The service should be interoperable with other cloud storage services, enabling

i) data migration from one CSP to another.

j) **Operating System:** Windows

k) **Coding Language:** Python 3.7

3.2.2 HARDWARE REQUIREMENTS

a) **Processor** – Intel core i5

b) **Speed –** 1.19 GHz

c) **RAM -** 512 MB (min)

d) **Hard Disk** - 20 GB

e) **Key Board -** Standard Keyboard

f) **Monitor –** 15 VGA Colour

CHAPTER 4

DESIGN

4.1 UNIFIED MODELING LANGUAGE

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates

The Primary goals in the design of the UML are as follows:

a) Provide users a ready-to-use, expressive visual modeling Language so that
b) they can develop and exchange meaningful models.
c) Provide extend ability and specialization mechanisms to extend the core concepts.
d) Be independent of particular programming languages and development process.
e) Provide a formal basis for understanding the modeling language.
f) Encourage the growth of OO tools market.

4.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

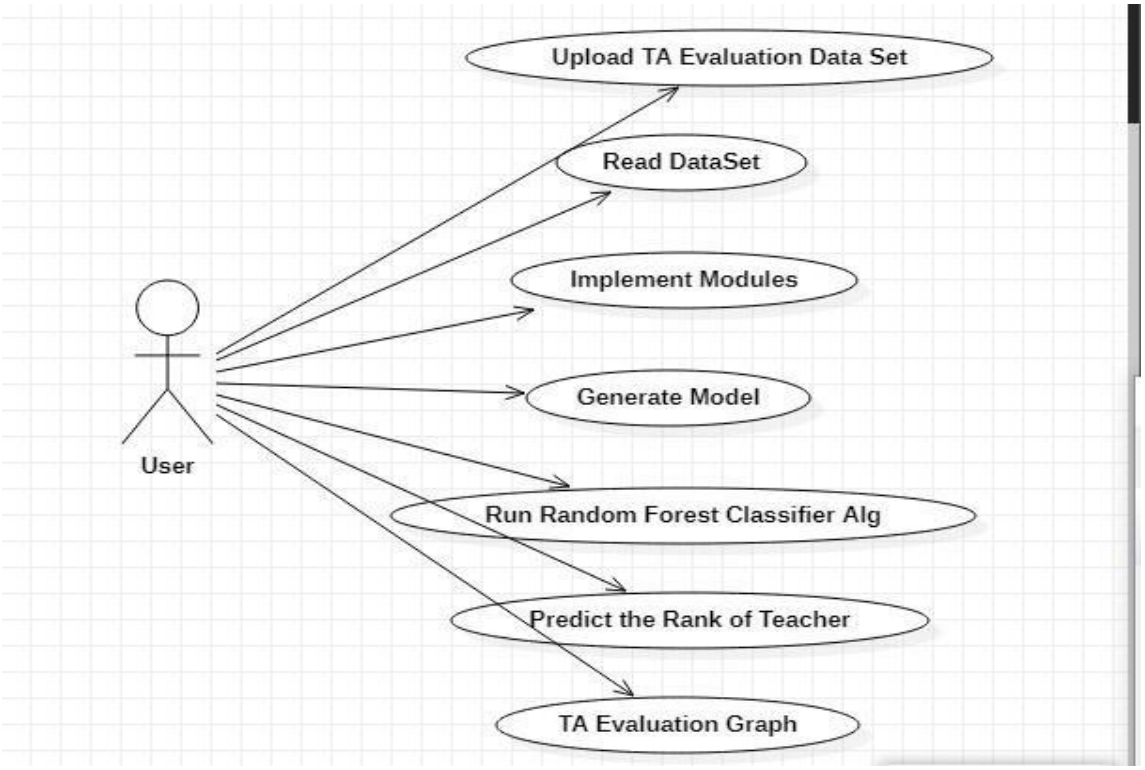

Fig 4.1 Use case Diagram

4.1.2 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

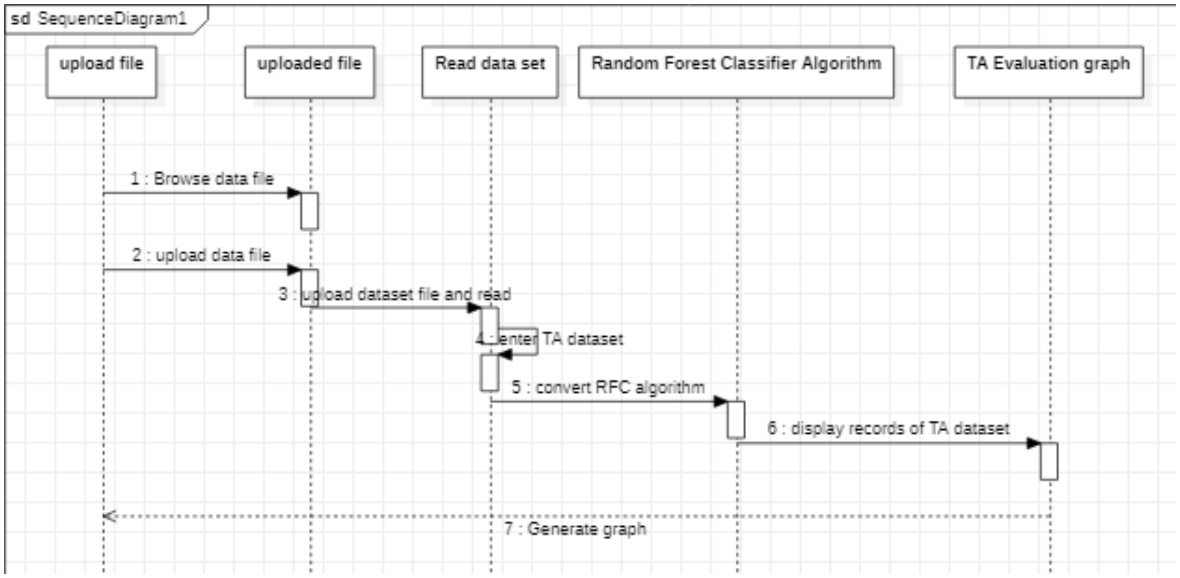

Fig 4.2 Sequence diagram

4.1.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

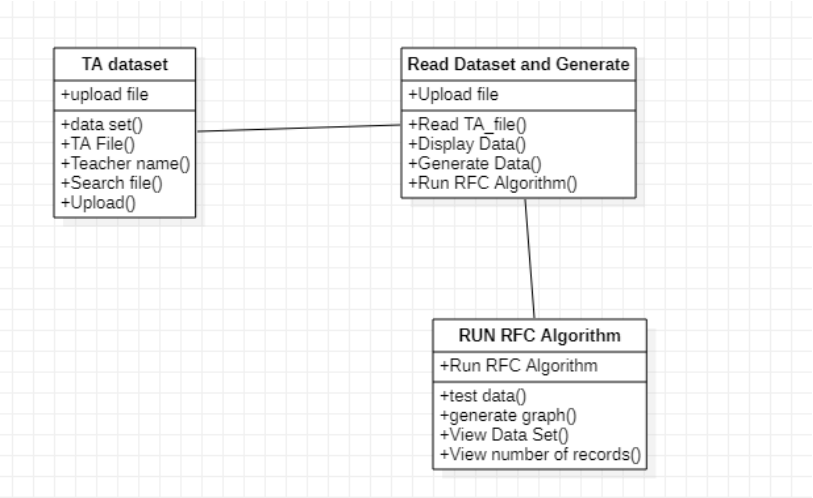

Fig 4.3 Class diagram

4.1.4 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus,from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

UML Component diagrams are used in modeling the physical aspects of object oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class

diagrams that focus on a system's components that often used to model the static implementation view of a system.

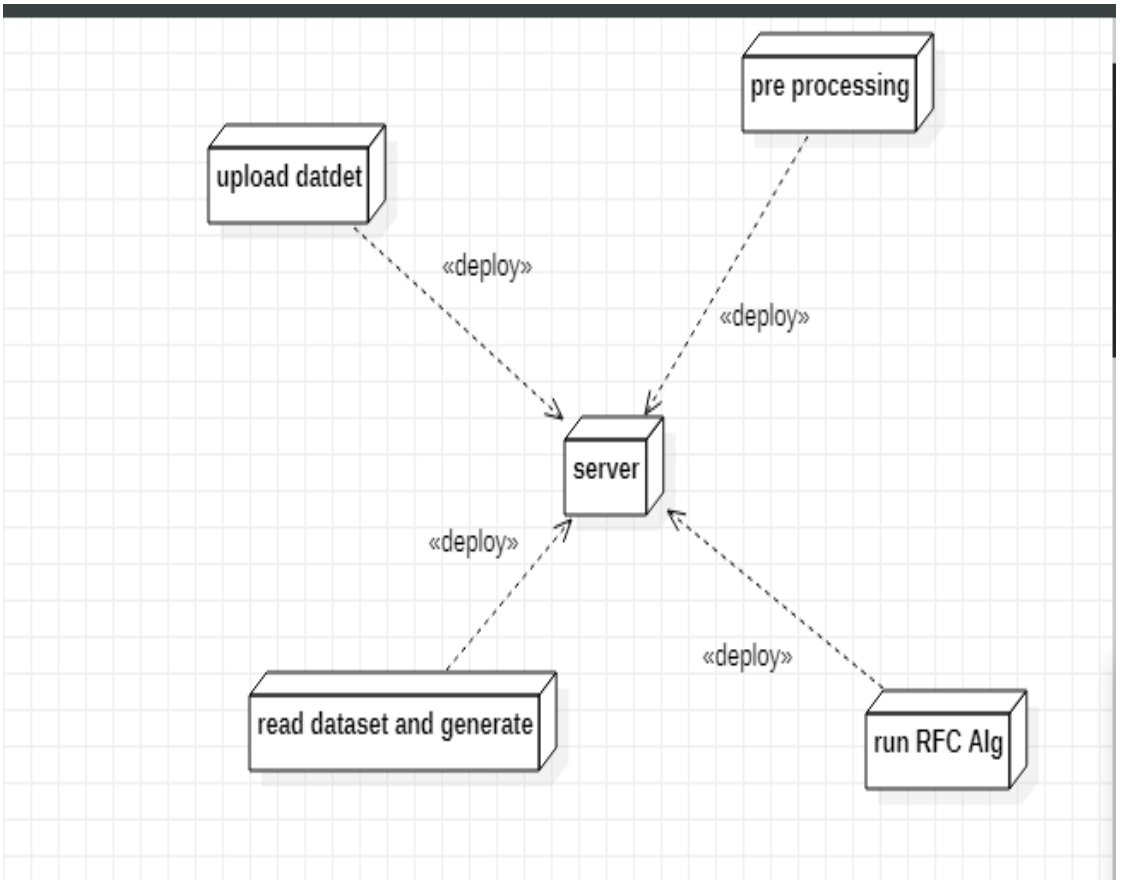

Fig 4.4 Component diagram

4.1.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

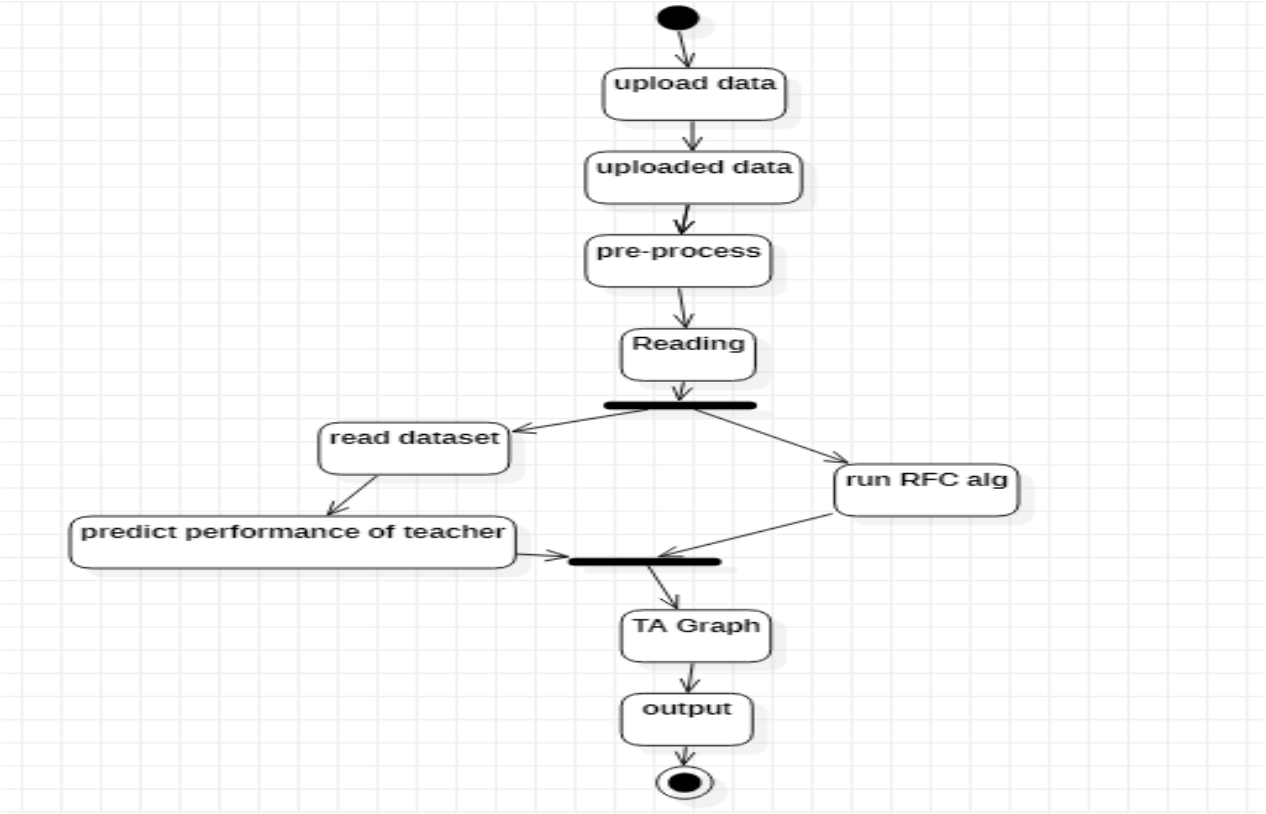

Fig 4.5 Activity diagram

4.1.6 FLOW CHART DIAGRAM

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

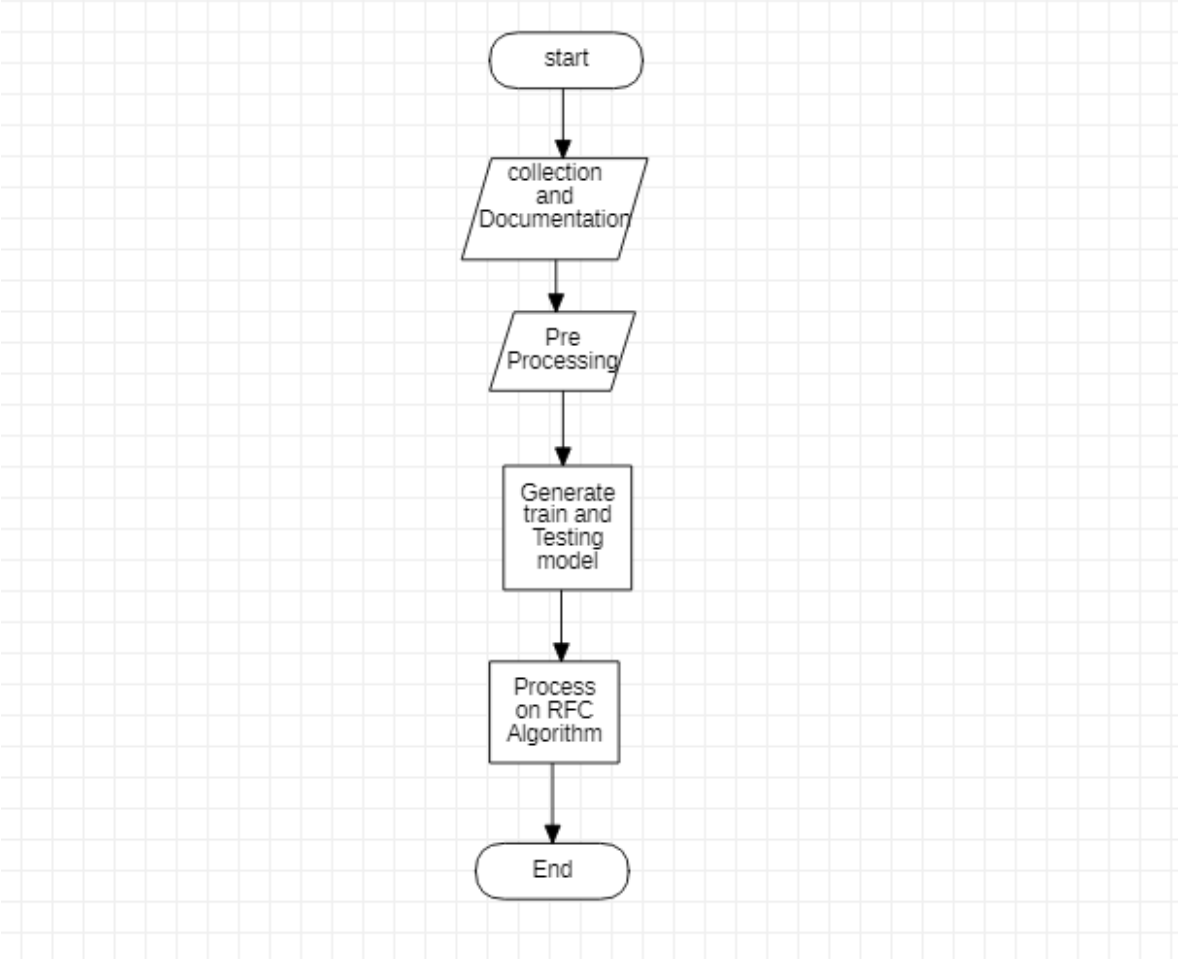

Fig 4.6 Flow chart diagram

CHAPTER 5
SUMMARY AND CONCLUSION

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.

This paper tried to investigate and evaluate the teaching assistant staff to adopt the prediction process based on the data size, time process, accuracy, estimated error factor. The quality of teaching assistant staff was emphasized using artificial neural network algorithms In this study, the experiments conducted based on UCI teaching assistant dataset. The results of the evaluation obtained using with different size in training and testing phases. The deep examinations highlighted thatthe group of 90% achieved better results in the prediction accuracy, estimated time,and error factor.

REFERENCES

1) Yang, J., Jiang, H., & Zhang, H. (2011). Teaching Assistant Evaluation Based on Support Vector Machines with Parameters Optimization. Information Technology Journal, 10(11), 2140-2146.

2) Badur, B., & MARDIKYAN, S. (2011). Analyzing teaching performance of instructors using data mining techniques. Informatics in Education-An International Journal, (Vol 10_2), 245-257.

3) Bao, Q., Newport, D., Chen, M., Stout, D. B., & Chatziioannou, A. F. (2009). Performance evaluation of the inveon dedicated PET preclinical tomograph based on the NEMA NU-4 standards. Journal of Nuclear Medicine. 50(3), 401-408.

4) Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students'

performance. arXiv preprint arXiv:1201.3417.

5) Yadav, S. K., & Pal, S. (2012). Data mining: A prediction for performance improvement of engineering students using classification. arXiv preprint arXiv:1203.3832.

6) Ball, D. L. and H. Bass. 2000. Interweaving content and pedagogy in teaching and learning to teach: Knowing and using mathematics. In Multiple perspectives on the teaching and learning of mathematics, ed. J. Bowler,83 – 104. Westport, CT: Ablex.

7) Brown J. 2002. The World Café: A resource guide for hosting conversations that matter. Mill Valley, CA: Whole Systems Associates.

8) Brown, J. and D. Isaacs. 2005. The World Café: Shaping our futures through conversations that matter. San.Francisco, CA: Berrett-Koehler Publishers.

9) Dillenbourg P. 1999. What do you mean by collaborative learning? In Collaborative-learning: cognitive and computational approaches, ed. P. Dillenbourg, 1 – 19. Oxford: Elsevier.

10) Dotger, S. 2011. Exploring and developing graduate teaching assistants' pedagogies via lesson study. Teaching in Higher Education 16 (2): 157 –169

APPENDIX

A. SCREENSHOTS

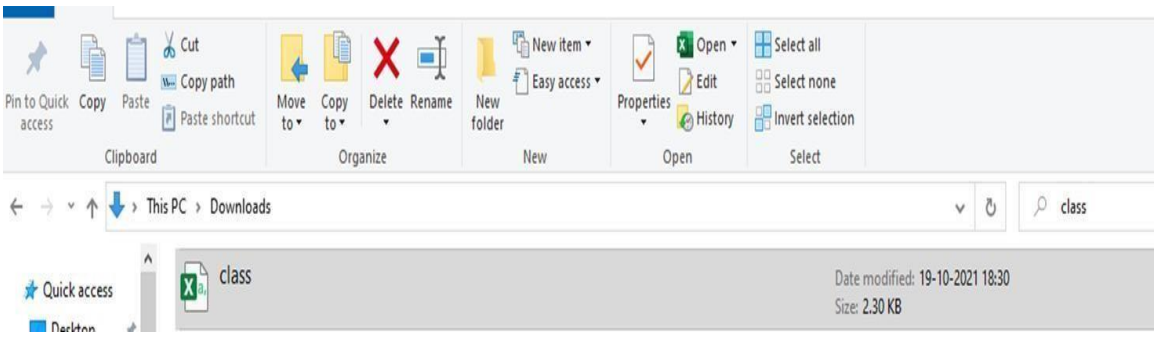

Fig 5.1 Uploading TA dataset

```
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 151 entries, 0 to 150
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Native English Speaker  151 non-null    int64
 1   Course Instructor       151 non-null    int64
 2   Course                  151 non-null    int64
 3   Summer or regular       151 non-null    int64
 4   Class size              151 non-null    int64
 5   Class attribute         151 non-null    int64
dtypes: int64(6)
memory usage: 7.2 KB
```

Fig 5.2 Dataset Information

```python
#Evaluating the model performance
from sklearn import metrics
cm = metrics.confusion_matrix(y_test, y_pred)
print("Confussion Matrix:\n",cm)
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Accuracy score:",accuracy)
```

```
Confussion Matrix:
 [[10  3  1]
 [ 3  7  1]
 [ 2  1 10]]
Accuracy score: 0.7105263157894737
```

Fig 5.3 Confusion Matrix and Accuracy

```
from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(classifier,x_test,y_test)
plt.show()
```

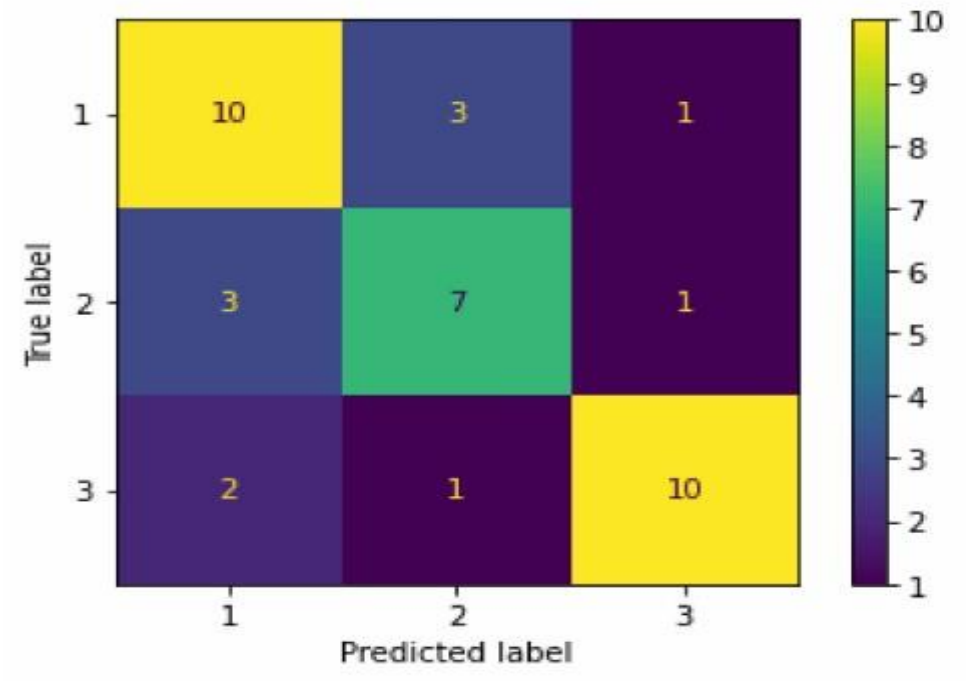

Fig 5.4 pictorial representation of Confusion matrix

```
plt.plot(y_train,y_train)
plt.xlabel('X_Trained Data')
plt.ylabel('y_trained Data')
plt.show()
```

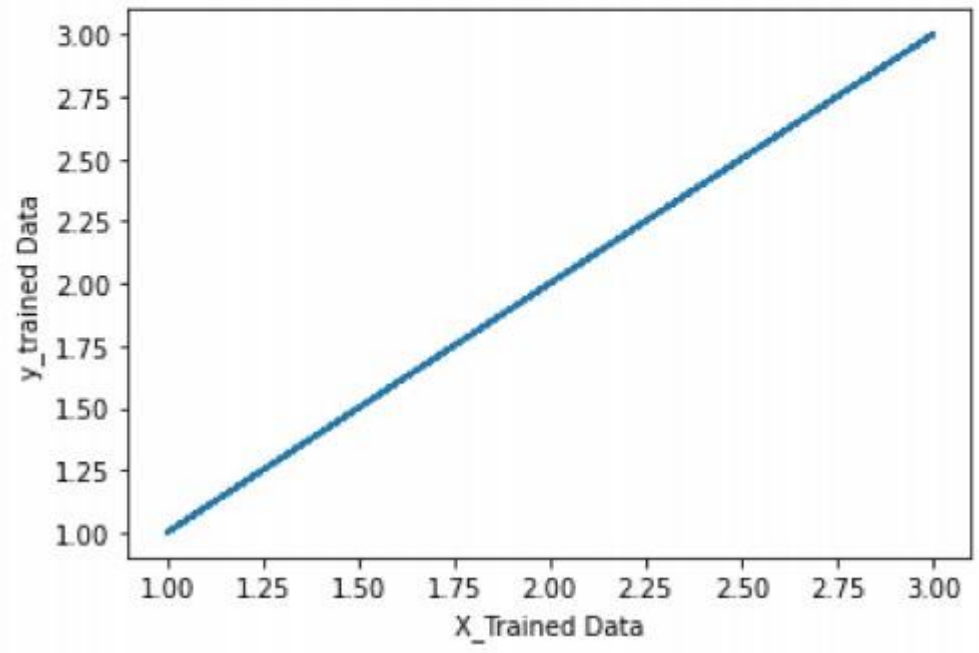

Fig 5.5 Plot of X_trained and Y_trained data

```
plt.plot(y_test,y_pred,color='red')
plt.xlabel('Tested Data')
plt.ylabel('Predicted Data')
plt.show()
```

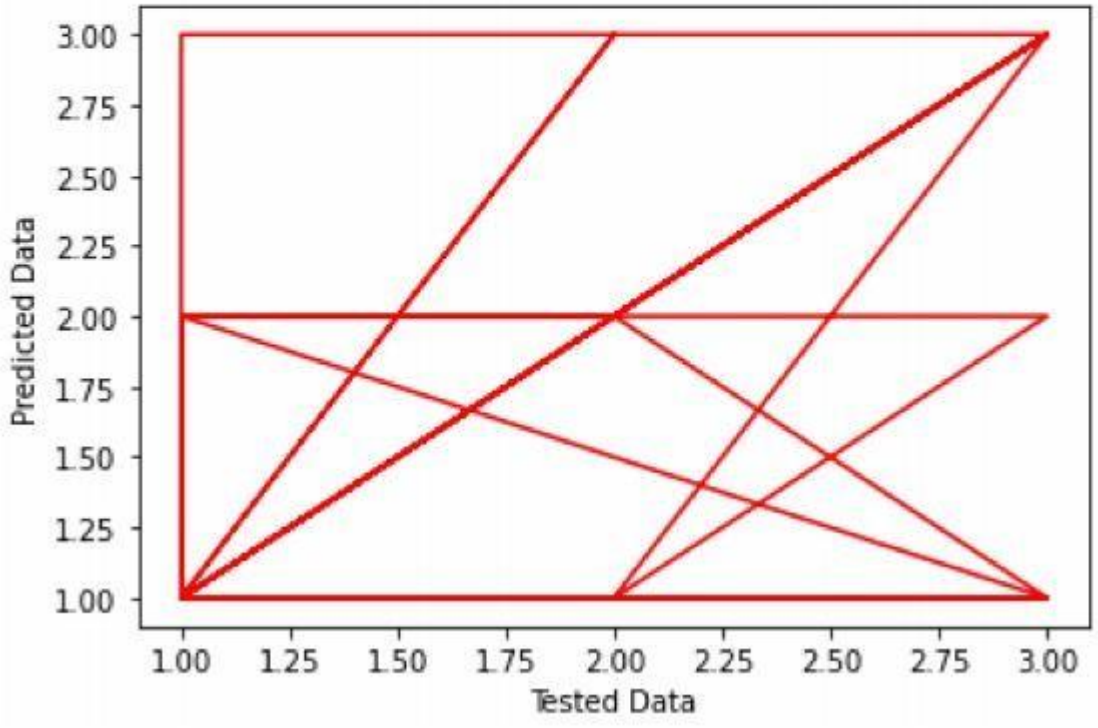

Fig 5.6 plot of Y_tested and Predicted data

```
plt.plot((accuracy,100))
plt.xlabel('No of trees')
plt.ylabel('Accuracy')
plt.show()
```

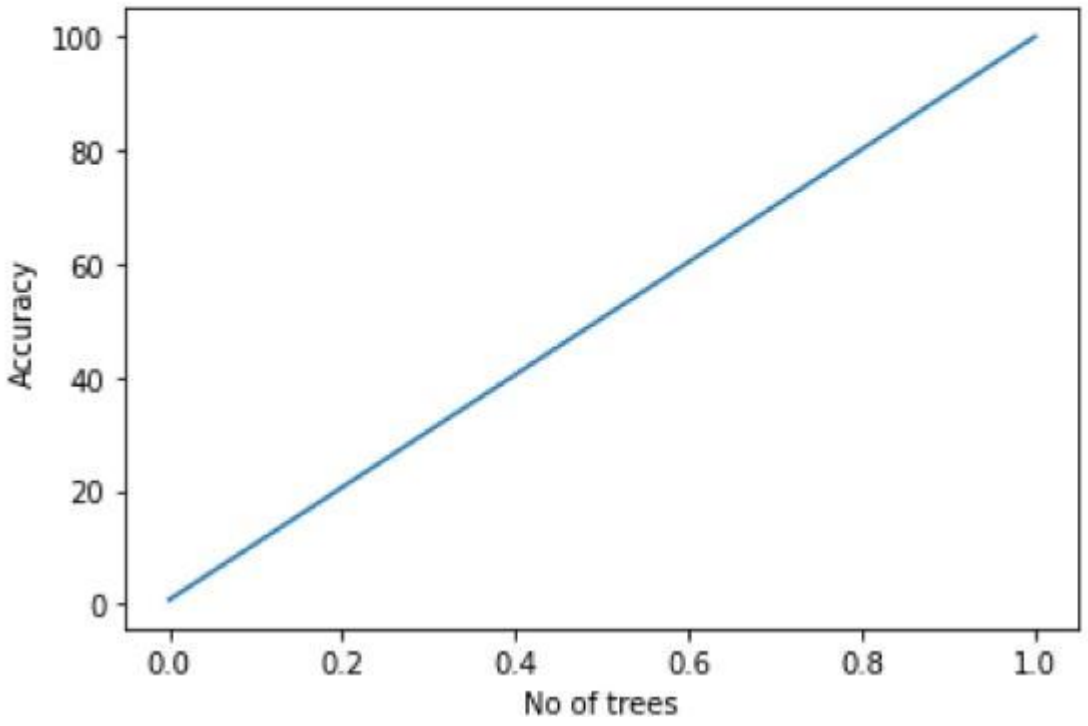

Fig 5.7 plot of Accuracy and N_estimators

B. SOURCE CODE

```
# importing libraries
import numpy as nm
import matplotlib.pyplot as plt
import pandas as pd

#importing datasets
data_set= pd.read_csv('class.csv')
data_set.shape
data_set.info()
data_set.head()

#Extracting Independent and dependent Variable
y = data_set['Class attribute']
x=data_set.drop('Class attribute', axis=1, inplace=True)
y.shape
y

#New Shape After Splitting
data_set.shape
data_set.head()

# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(data_set, y, test_size=0.25,
shuffle=True)
x_train.head()
x_test.head()
y_train.head()
y_test.head()
```

```
#Fitting Decision Tree classifier to the training set
from sklearn.ensemble import RandomForestClassifier
classifier=RandomForestClassifier(n_estimators=100,criterion='entropy',random_s
tate=42)
classifier=RandomForestClassifier(n_estimators=100,max_features=0.7,bootstrap
=True, max_depth=10, min_samples_leaf=2, random_state=42)

#fitting
classifier.fit(x_train, y_train)
# Prediction

y_pred = classifier.predict(x_test)#Metrics
from sklearn import metrics
cm = metrics.confusion_matrix(y_test, y_pred)
accuracy = metrics.accuracy_score(y_test, y_pred)

#Evaluating the model performance
from sklearn import metrics
cm = metrics.confusion_matrix(y_test, y_pred)
print("Confussion Matrix:\n",cm)
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Accuracy score:",accuracy)

from sklearn.metrics import plot_confusion_matrix
plot_confusion_matrix(classifier,x_test,y_test)
plt.show()

plt.plot(y_train,y_train)
plt.xlabel('X_Trained Data')
plt.ylabel('y_trained Data')
plt.show()
```

```
plt.plot(y_test,y_pred,color='red')
plt.xlabel('Tested Data')
plt.ylabel('Predicted Data')
plt.show()

plt.plot((accuracy,100))
plt.xlabel('No of trees')
plt.ylabel('Accuracy')
plt.show()
```