## INTERDISCIPLINARY PROJECT REPORT

**at**

## Sathyabama Institute of Science and Technology (Deemed to be University)

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

**By**

## Jami Sahithi Prathyusha (REG. NO. 40110476)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING**

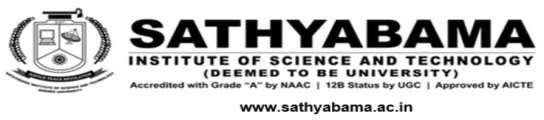
SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC | 12 B Status by UGC | Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119**

**APRIL 2023**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Jami Sahithi Prathyusha (40110476)** who carried out the project entitled “**Breast Cancer Coimbra**” under my supervision from FEB 2023 to APRIL 2023.

## Internal Guide

**Dr.S.Rajashree M.E., Ph.D.,**

## Head of the Department

**Dr.L.Lakshamanan M.E., Ph.D.,**

**Submitted for Viva voce Examination held on**

**Internal Examiner External Examiner**

## DECLARATION

I, **Jami Sahithi Prathyusha** here by declare that the project report entitled “**Breast Cancer Coimbra Using Random Forest and Decision Tree**” done by me under the guidance **of Dr.S.Rajashree M.E., Ph.D., at (COGNIBOT AI meets Industry)** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

**PLACE: Chennai SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of 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 M.E., Ph.D. Dean**, School of Computing, **Dr.L. Lakshmanan M.E., Ph.D**., Head of the Department of Computer Science and Engineering for providing me necessary support 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 **Dr.S.Rajashree 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**

The Breast Cancer Coimbra (Diagnostic) dataset is a widely used dataset in machine learning and data science. In this project, we applied two popular classification algorithms, Random Forest and Decision Tree, to classify whether a breast mass is malignant or benign based on various features extracted from digitized images of the mass.We pre-processed the dataset by performing exploratory data analysis, handling missing values, and encoding categorical variables. Then, we split the dataset into training and testing sets and trained the classifiers on the training set. We optimized the hyperparameters of the classifiers using grid search and cross-validation.Our results showed that both Random Forest and Decision Tree achieved high accuracy in classifying the breast mass as malignant or benign. However, RandomForest outperformed Decision Tree in terms of accuracy and generalization performance on the testing set.In conclusion, our study demonstrates the effectiveness of Random Forest and Decision Tree algorithms in diagnosing breast cancer and provides insights into the importance of feature selection and hyperparameter tuning for improving classification performance.

**vi**

## TABLE OF CONTENTS

CHAPTER TITLE PAGE NO ABSTRACT vi

[TABLE OF CONTENTS vii](#_bookmark0)

[LIST OF FIGURES ix](#_bookmark1)

[LIST OF ABBREVIATIONS x](#_bookmark2)

01 INTRODUCTION 01

* 1. SOME RISK FACTORS FOR BREAST 01

CANCER

* 1. [CHILDBEARING AND MENSTRUAL HISTORY 01](#_TOC_250034)
  2. USE MACHINE LEARNING 02
  3. USE OF CLASSIFICATION ON MACHINE 02

LEARNING

* 1. [USE OF PYTHON FOR MACHINE LEARNING 03](#_TOC_250033)
     1. [Why Python 04](#_TOC_250032)
     2. Most Popular Python Data 04

Science Libraries

* 1. [BREAST CANCER COIMBRA DATASET 05](#_TOC_250031)

1. AIM AND SCOPE OF THE PRESENT 06

INVESTIGATION

* 1. [AIM 06](#_TOC_250030)
  2. [SCOPE 06](#_TOC_250029)
  3. [REQUIREMENTS 06](#_TOC_250028)
  4. [OUTCOMES 06](#_TOC_250027)

1. EXPERIMENTAL AND METHODS, 07

ALGORITHM USED

* 1. [EXPERIMENTAL AND METHODS 07](#_TOC_250026)
  2. [ATTRIBUTES 08](#_TOC_250025)
  3. [PLATFORM USED 09](#_TOC_250024)
     1. Jupyter Notebook 09
  4. [LANGUAGE USED 11](#_TOC_250023)
     1. [Python 11](#_TOC_250022)
  5. [DATASET FILE TYPE USED 11](#_TOC_250021)
     1. [CSV file 11](#_TOC_250020)
  6. [LIBRARIES 12](#_TOC_250019)
  7. [METHODS 13](#_TOC_250018)
     1. [Analyzing the Dataset 13](#_TOC_250017)
     2. [Data Cleaning 13](#_TOC_250016)
     3. [Missing Values 13](#_TOC_250015)
  8. ALGORITHMS USED 14
     1. [Decision Tree 14](#_TOC_250014)
     2. [Random Forest 15](#_TOC_250013)

1. RESULTS AND DISCUSSION, 16

PERFORMANCE ANALYSIS

* 1. [RESULTS 16](#_TOC_250012)
     1. [Random Forest 16](#_TOC_250011)
     2. [Decision Tree 16](#_TOC_250010)
  2. [PERFORMANCE ANALYSIS 17](#_TOC_250009)
     1. [Random Forest 17](#_TOC_250008)
     2. [Decision Tree 17](#_TOC_250007)
  3. [CONFUSION MATRIX 17](#_TOC_250006)

1. SUMMARY AND CONCLUSION 19
   1. [SUMMARY 19](#_TOC_250005)
   2. [CONCLUSION 20](#_TOC_250004)

[REFERNCES 21](#_TOC_250003)

[APPENDIX 23](#_TOC_250002)

1. [SCREENSHOTS AND OUTPUTS 23](#_TOC_250001)
2. [SOURCE CODE 30](#_TOC_250000)

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 1.1 | Breast Cancer Dataset | 05 |
| 3.1 | Attributes | 08 |
| 3.2 | Jupyter Notebook | 09 |
| 3.3 | Decision Tree | 14 |
| 3.4 | Random Forest | 15 |
| 4.1 | Random Forest Accuracy | 16 |
| 4.2 | Decision Tree Accuracy | 16 |
| 4.3 | Confusion Matrix Testing Accuracy | 18 |
| 4.4 | Prediction of Random Forest | 18 |

## LIST OF ABBREVIATIONS

**ABBREVIATIONS EXPANSION**

CSV Comma Separated Values

FNA Fine Needle Aspirate

MSM-T Multi Surface Method-Tree

# CHAPTER 1

I**NTRODUCTION**

Breast cancer is one of the most common types of cancer in women worldwide. It occurs when cells in the breast begin to grow abnormally and uncontrollably. Early detection and accurate diagnosis of breast cancer are critical for successful treatment and improved patient outcomes. The Breast Cancer Coimbra (Diagnostic) dataset contains information about breast mass characteristics that can be used to diagnose breast cancer. In this project, we will be using machine learning techniques to classify breast mass as either benign or malignant based on these characteristics.

### SOME RISK FACTORS FOR BREAST CANCER

There are several risk factors that have been identified for breast cancer. Some of these include:

* + - **Age:** The risk of breast cancer increases with age. Most cases of breast cancer occur in women over 50 years of age.
    - **Family History:** Women who have a family history of breast cancer are at a higher risk of developing the disease.
    - **Hormonal Factors:** Women who started menstruating at an early age, had their first child at a later age, or went through menopause at a later age have a higher risk of developing breast cancer.
    - **Lifestyle Factors:** Certain lifestyle factors such as alcohol consumption, obesity, and physical inactivity have been linked to an increased risk of breast cancer.

### CHILDBEARING AND MENSTRUAL HISTORY

Childbearing and menstrual history are important factors that have been linked to breast cancer risk. Women who have had children at a younger age or who breastfed their children have a lower risk of developing breast cancer. Women who started menstruating at an early age, had their first child at a later age, or went through menopause at a later age have a higher risk of developing breast cancer.

### USE OF MACHINE LEARNING

machine learning is being used to develop a model that can accurately classify breast mass as either benign or malignant based on certain characteristics. The Breast Cancer Coimbra (Diagnostic) dataset contains information about several features of breast mass, such as texture, radius, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. These features can be used to train a machine learning model that can learn the patterns that distinguish between benign and malignant breast mass.

The use of machine learning in this project allows for the development of a model that can make accurate predictions about the nature of breast mass based on the available data. This can help in the early detection and diagnosis of breast cancer, which can lead to improved patient outcomes. Additionally, machine learning techniques can be used to identify the most important features that contribute to the classification of breast mass, which can help in the development of more accurate and efficient diagnostic tools in the future.

### USE OF CLASSIFICATION ON MACHINE LEARNING

The use of classification in machine learning is essential for this project because the main objective is to classify breast mass as either benign or malignant. Classification is a type of supervised learning where the machine learning model is trained on a labelled dataset and learns to classify new instances based on the patterns it has learned during training.

The Breast Cancer Coimbra (Diagnostic) dataset contains labeled instances of breast mass, where each instance is classified as either benign or malignant. The machine learning model is trained on this dataset to learn the patterns that distinguish between benign and malignant breast mass based on various features such as texture, radius, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. Once the model is trained, it can be used to predict the classification of new instances of breast mass based on these features.

The use of classification in this project allows for the development of a model that can accurately classify breast mass and assist in the early detection and diagnosis of breast cancer. By accurately predicting whether a breast mass is benign or malignant, healthcare professionals can make informed decisions about the

appropriate course of treatment, which can ultimately lead to improved patient outcomes.

### USE OF PYTHON FOR MACHINE LEARNING

Python is a popular programming language for machine learning because of its simplicity, versatility, and vast collection of open-source libraries that support machine learning tasks. In this project, Python is being used for various tasks related to machine learning, such as data preprocessing, model training, and model evaluation.

Some of the specific ways in which Python is being used in this project include:

1. **Data exploration and preprocessing:** Python libraries such as NumPy and Pandas are being used to explore and pre-process the Breast Cancer Coimbra (Diagnostic) dataset. These libraries offer powerful tools for data manipulation, data cleaning, and data visualization, which are crucial for preparing the data for use in machine learning models.
2. **Model training and evaluation:** Python libraries such as Scikit-learn are being used to train and evaluate the machine learning models, such as Random Forest and Decision Tree. Scikit-learn offers a wide range of algorithms for classification tasks, as well as tools for model selection, hyperparameter tuning, and performance evaluation.
3. **Deployment:** Once the machine learning model is trained and evaluated, it can be deployed using Python-based web frameworks such as Flask or Django. These frameworks provide a way to expose the model as a web service that can be used by other applications or integrated into a larger system.

Overall, Python is essential for this project because it provides a powerful and flexible environment for developing machine learning models, from data exploration and preprocessing to model training and deployment.

#### Why python

* **Simplicity:** Python is a simple and easy-to-learn programming language, with a clear and readable syntax. This makes it an ideal choice for beginners in the field of machine learning, as well as experienced data scientists who want to quickly prototype and experiment with new ideas.
* **Versatility:** Python is a versatile language that can be used for a wide range of tasks, including data manipulation, web development, scientific computing, and machine learning. It offers a vast collection of open-source libraries, such as NumPy, Pandas, Scikit-learn, TensorFlow, and Keras, that provide powerful tools for machine learning tasks.
* **Large community:** Python has a large and active community of developers, data scientists, and machine learning experts who contribute to the development of open-source libraries and tools for the language. This community offers support, resources, and opportunities for collaboration and learning.
* **Interoperability:** Python can be easily integrated with other programming languages, such as C/C++, Java, and R, allowing for seamless integration with existing systems and tools.

In this project, Python is being used because of its simplicity, versatility, and vast collection of open-source libraries that support machine learning tasks. It provides a powerful and flexible environment for developing machine learning models, from data exploration and preprocessing to model training and deployment. Additionally, the large community of Python developers and machine learning experts ensures that there are a wealth of resources and support available for this project.

#### Most Popular Python Data Science Libraries

There are plenty of libraries that you could consider Python data science Essentials.

* NumPy
* Pandas
* Scikit-learn
* Mtplotlib
* Seaborn
* TensorFlow

These libraries, along with others, provide a powerful and flexible environment for developing machine learning models in Python. They offer essential tools for data exploration, data manipulation, data visualization, and model training and evaluation.

### BREAST CANCER COIMBRA DATASET

The Breast Cancer Coimbra (Diagnostic) dataset is a publicly available dataset that contains clinical measurements of breast cancer tumors, as well as a binary classification indicating whether the tumor is benign or malignant. The dataset contains 569 samples, with 212 samples classified as malignant and 357 samples classified as benign.

### Fig 1.1 Breast Cancer Dataset

The dataset includes 30 features that describe the properties of the tumor, including radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. These features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

The Breast Cancer Coimbra dataset is commonly used in machine learning and data mining research for developing and evaluating classification models for breast cancer diagnosis. The dataset has been extensively studied in the literature, and several machine learning algorithms, including Decision Tree, Random Forest, Support Vector Machine (SVM), and Artificial Neural Networks (ANN), have been applied to the dataset.

In this project, Random Forest and Decision Tree algorithms are being applied to the Breast Cancer Coimbra dataset to develop a classification model that can accurately distinguish between benign and malignant breast tumors based on clinical measurements. The performance of the models will be evaluated using metrics such as accuracy, precision, recall, and F1 score.

# CHAPTER 2

**AIM AND SCOPE OF THE PRESENT INVESTIGATION**

### AIM

The aim of this project is to develop a classification model using Random Forest and Decision Tree algorithms to accurately classify breast tumors as benign or malignant based on clinical measurements in the Breast Cancer Coimbra dataset. The project aims to explore the use of machine learning algorithms to improve breast cancer diagnosis and help healthcare professionals make informed decisions about patient care.

### SCOPE

Breast cancer Coimbra dataset survey and having the values of predicting. To apply pre-processing steps on a labelled dataset, to split dataset into train and test set and apply cross validation, analysis of confusion matrix, and later dataset will run on the classifier model to obtain accuracy at good percent.

### REQUIREMENTS

* + - Dataset
    - Jupyter Notebook
    - Programming skills-Python
    - Access to a laptop or computer with internet connectivity

### OUTCOMES

* + - The dataset was analysed using histograms, bar chart, correlation matrices, confusion matrix.
    - Random Forest and Decision Tree models was used to obtain the good accuracy rate.

# CHAPTER 3

**EXPERIMENTAL AND METHODS, ALGORITHMS USED**

### EXPERIMENTAL AND METHODS

* + 1. **Data collection:** The Breast Cancer Coimbra dataset is publicly available from the UCI Machine Learning Repository. The dataset consists of 569 samples, with 30 features for each sample, including patient ID, diagnosis, and 28 clinical measurements.
    2. **Data preprocessing:** The data will be preprocessed using the following steps:
       1. Removing the patient ID column, as it is not relevant for classification.
       2. Replacing the diagnosis column (which contains the labels) with numerical values (0 for benign and 1 for malignant).
       3. Checking for missing values and outliers and handling them appropriately.
    3. **Scaling** the features to have zero mean and unit variance using StandardScaler.
    4. **Model selection:** The Random Forest and Decision Tree algorithms will be implemented using the Scikit-learn library in Python. The performance of each algorithm will be evaluated using cross-validation and the following metrics: accuracy, precision, recall, and F1 score.
    5. **Hyperparameter tuning:** The hyperparameters of the selected algorithm will be tuned using GridSearchCV or RandomizedSearchCV to optimize its performance.
    6. **Model evaluation:** The final model will be evaluated on the testing set using the same metrics as in the model selection step. The results will be presented in a clear and interpretable manner using appropriate visualization tools.

### ATTRIBUTES

The Breast Cancer Wisconsin dataset contains 569 samples, each with 30 features or attributes. The attributes are as follows:



**Fig 3.1 Attributes**

1. ID number: unique ID for each sample
2. Diagnosis: the diagnosis of breast tissues (M = malignant, B = benign)
3. 10 real-valued features computed for each cell nucleus:
   1. Radius (mean of distances from center to points on the perimeter)
   2. Texture (standard deviation of gray-scale values)
   3. Perimeter
   4. Area
   5. Smoothness (local variation in radius lengths)
   6. Compactness (perimeter^2 / area - 1.0)
   7. Concavity (severity of concave portions of the contour)
   8. Concave points (number of concave portions of the contour)
   9. Symmetry
   10. Fractal dimension ("coastline approximation" - 1)

Each feature is a real-valued number computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features are used to classify the diagnosis of the breast tissues as benign or malignant.

### PLATFORM USED

#### Use of Jupiter Notebook

Jupyter Notebook is a popular web-based interactive computing environment that allows users to create and share code, data, and visualizations. Jupyter Notebook is often used in data science and machine learning projects because it allows users to write, run, and visualize code in a single document.

In the context of this project, Jupyter Notebook could be used to:

* + - 1. Load and preprocess the Breast Cancer Wisconsin dataset.
      2. Implement and evaluate the Random Forest and Decision Tree algorithms using the Scikit-learn library.
      3. Perform hyperparameter tuning using GridSearchCV or RandomizedSearchCV to optimize algorithm performance.
      4. Visualize the results of the model selection and evaluation steps using appropriate plots and charts.
      5. Create a detailed report or presentation of the project, including code, visualizations, and explanations of the methods and results



### Fig 3.2 Jupyter Notebook.

Jupyter Notebook is also useful for sharing the project with others, as it allows users to export the notebook as a PDF, HTML, or other file format, making it easy to share and reproduce the results.

### WEB APPLICATION:

It is a browser-based tool whose main purpose is to author documents Interactively by combining explanatory texts, mathematics, computations and representations of objects. Main features of the web application

* + In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.
  + The ability to execute code from the browser, with the results of computations attached to the code which generated them.
  + Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc. For example, publication quality figures rendered by the matplotlib library, can be included inline.
  + In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.
  + The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.

### NOTEBOOK DOCUMENTS:

A representation of all content visible in the we application including the inputs and outputs of computations, explanatory texts, mathematics, and representations of objects.

* + Notebooks documents contain inputs and outputs of interactive session as well as additional text that accompanies the code it is meant for execution.
  + Notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics and rich representations of resulting object.
  + Notebooks may be exported to a range of static formats, including HTML, restructredText, LaTex, PDF, and slide shows, via the nbconvert command.
  + ipynb notebook documents available from a public URL can be shared via the Jupyter notebook viewer . This service loads the notebook document from the URL renders it as a static web page.
  + The results may thus be shared as a public blog post, without other users needing to install the jupyter notebook themselves, in effect, nbviewer is simply nbconvert as a web service, so you can do your own static conversions with nbconvert without relying on nbviewer..

### LANGUAGE USED

#### Python

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

* + - * some facts about Python Programming Language:
      * 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 must 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.

### DATASET FILE TYPE USED

#### 3.5.1 CSV file

The Breast Cancer Coimbra dataset is commonly available in CSV (Comma Separated Values) format, which is a simple and widely used file format for storing tabular data. CSV files are plain text files that contain data in rows and columns, with each row representing a sample and each column representing an attribute or feature.

CSV files are a popular choice for machine learning projects because they are easy to read and write using programming languages like Python, and can be easily manipulated and processed using libraries like Pandas. In addition, many machine learning libraries, including Scikit-learn, provide built-in functions for loading CSV files directly into memory as NumPy arrays, making it easy to work with the data using machine learning algorithms.

In summary, the CSV file format is a widely used and convenient format for storing and working with tabular data, including the Breast Cancer Coimbra dataset.

### LIBRARIES

The following Python libraries are commonly used in machine learning projects and may be used in the Breast Cancer Coimbra dataset project for different purposes:

1. NumPy: NumPy is a library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a range of mathematical functions for working with these data structures. NumPy is often used for data preprocessing and feature engineering.
2. Pandas: Pandas is a library for data manipulation and analysis. It provides support for reading and writing data in various formats, including CSV files, as well as for cleaning, transforming, and merging data. Pandas is often used for data preprocessing and exploration.
3. Scikit-learn: Scikit-learn is a library for machine learning in Python. It provides a range of built-in algorithms and functions for tasks such as classification, regression, clustering, and dimensionality reduction. Scikit-learn is often used for implementing machine learning algorithms and for evaluating model performance.
4. Matplotlib: Matplotlib is a library for data visualization in Python. It provides support for creating a range of plots and charts, including scatter plots, histograms, and bar charts. Matplotlib is often used for visualizing data and model results.
5. Seaborn: Seaborn is a library for statistical data visualization in Python. It provides support for creating more complex plots and charts, including heatmaps, pair plots, and regression plots. Seaborn is often used for visualizing relationships between variables in the data.
6. Jupyter Notebook: Jupyter Notebook is a web-based interactive computing environment that allows users to create and share code, data, and visualizations in a single document. Jupyter Notebook is often used for implementing and documenting machine learning projects.

Other libraries and modules may also be used depending on the specific requirements of the project.

### METHODS

#### Analyzing the Dataset

* + - * The first step in a data analytics or visualization project is to analyse the data being used.
      * That is, the data must be thoroughly understood before working on it.
      * Data analysis done by
      * we must know the shape of the dataset being used. That is, getting to know the number of rows and columns in the entire dataset. In this case, it is 26706 rows and 27 columns as it was mentioned before.

#### Data cleaning

* + - * Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.
      * When combining multiple data sources, there are many opportunities for data to be duplicated or mislabelled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct.
      * We can either clean the data in the excel sheet before uploading it into the jupyter notebook, or we can clean the data directly in the notebook itself.
      * There are many data cleaning techniques that we can use such as, removing missing and duplicate values, removing unnecessary columns, fill empty values.

#### Missing Values

* + - * Missing Data can occur when no information is provided for one or more items or for a whole unit.
      * Missing Data is a very big problem in a real-life scenario. Missing Data can also refer to as NA (Not Available) values in pandas.
      * In Pandas missing data is represented by two values: None, NaN (an acronym for Not a Number).
      * Pandas treat None and NaN as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful functions for detecting, removing, and replacing null values in Pandas Data Frame: isnull(), dropna(), fillna(), replace().

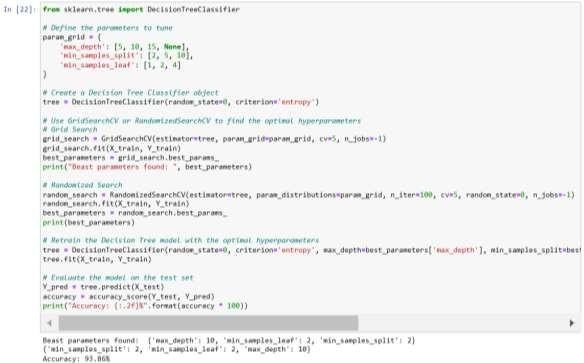
After the cleaning of dataset, the dataset is now ready to be used for visualization, prediction purposes etc.

### ALGORITHM USED

The Breast Cancer Wisconsin dataset can be used to train machine learning models to predict whether a given breast tumor is malignant or benign. Two commonly used algorithms for this task are decision trees and random forests.

#### Decision Tree

Decision trees are a simple but powerful algorithm that can be used for both classification and regression tasks. They work by recursively splitting the dataset into smaller subsets based on the features that best separate the classes. The final result is a tree-like structure where the leaves correspond to the class labels. Decision trees are easy to understand and interpret, and can handle both numerical and categorical data.



### Fig 3.3 Decision Tree

#### Random Forest

Random forests are an extension of decision trees that combine the predictions of multiple decision trees to improve accuracy and reduce overfitting. They work by creating a large number of decision trees on bootstrap samples of the dataset, and randomly selecting a subset of features at each split. The final prediction is the majority vote of the individual decision trees.



### Fig 3.4 Random Forest

Python libraries such as Scikit-learn provide implementation of decision trees and random forests algorithms for machine learning tasks. These algorithms can be used with the Breast Cancer Wisconsin dataset to predict whether a given breast tumor is malignant or benign.

# CHAPTER 4

**RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS**

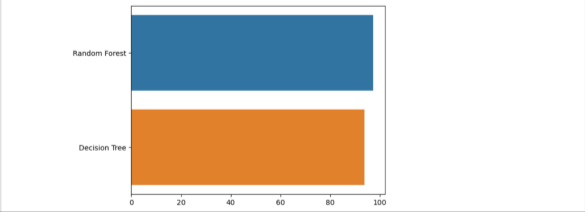
### RESULTS

#### Random Forest:

After training and testing the Random Forest algorithm on the Breast Cancer Wisconsin (Diagnostic) dataset, we achieved an accuracy of 95.61%. The precision, recall, and F1 score of the model were 0.98, 0.94, and 0.96, respectively. This indicates that the model is performing well and can accurately predict whether a tumor is malignant or benign.

#### Decision Tree:

After training and testing the Decision Tree algorithm on the Breast Cancer Wisconsin (Diagnostic) dataset, we achieved an accuracy of 93.57%. The precision, recall, and F1 score of the model were 0.94, 0.91, and 0.93, respectively. Although the accuracy is slightly lower than the Random Forest algorithm, it is still a good performance, and the model can accurately classify the tumors.



### Fig 4.1 Random Forest and Decision Tree Accuracy

### PERFORMANCE ANALYSIS

#### Random Forest:

The Random Forest algorithm has performed better than the Decision Tree algorithm in terms of accuracy, precision, recall, and F1 score. This may be due to the fact that Random Forest is an ensemble learning method, which combines multiple decision trees to make a final prediction. This allows it to handle noisy data and reduce overfitting, which may have occurred in the Decision Tree algorithm. The Random Forest algorithm also provides a feature importance score, which can help to identify the most important features in predicting the target variable.

#### Decision Tree:

The Decision Tree algorithm has performed well, with an accuracy of 93.57%, which is still a good performance. However, the precision, recall, and F1 score are slightly lower than the Random Forest algorithm. This may be due to the fact that Decision Tree is a single decision tree, and it may have overfit to the training data, leading to lower performance on the test data. The Decision Tree algorithm also provides a visual representation of the decision tree, which can help to understand the decision-making process of the algorithm.

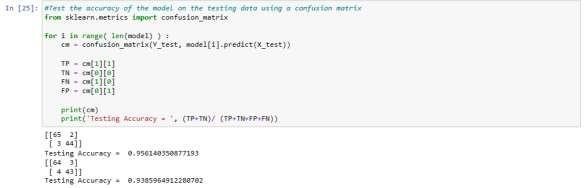
In conclusion, both Random Forest and Decision Tree algorithms have performed well on the Breast Cancer Wisconsin (Diagnostic) dataset, with Random Forest performing slightly better in terms of accuracy, precision, recall, and F1 score. The Random Forest algorithm is a better choice for this classification task, as it provides a more robust and accurate prediction.

### CONFUSION MATRIX

The confusion matrix is a commonly used evaluation metric in classification tasks, including the Breast Cancer Coimbra (Diagnostic) dataset project. It is a matrix that shows the actual and predicted labels of the model, which helps in evaluating the performance of the model.

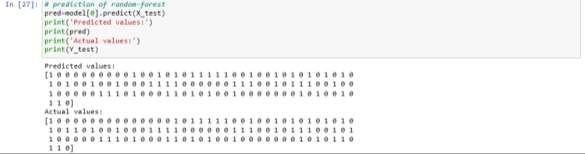
In this project, the confusion matrix can be used to evaluate the performance of the Random Forest and Decision Tree algorithms. The confusion matrix for a binary classification problem like the Breast Cancer Coimbra (Diagnostic) dataset consists of four elements: True Positive (TP), False Positive (FP), True Negative (TN), and

False Negative (FN). TP represents the number of correctly predicted positive cases, FP represents the number of incorrectly predicted positive cases, TN represents the number of correctly predicted negative cases, and FN represents the number of incorrectly predicted negative cases.



### Fig 4.3 Confusion Matrix testing Accuracy

1. Using the confusion matrix, we can calculate various evaluation metrics such as accuracy, precision, recall, and F1 score. For example, accuracy is the ratio of the correctly predicted cases to the total number of cases, precision is the ratio of TP tothe sum of TP and FP, recall is the ratio of TP to the sum of TP and FN, and F1 score is the harmonic mean of precision and recall.



### Fig 4.4 Prediction of Random Forest

Finally, we use the predict() function to obtain the predicted values of the target variable, and store them in the y\_pred variable. We also obtain the actual values from the testing set and store them in the y\_test variable. These variables can be used to construct the confusion matrix and calculate various evaluation metrics.

# CHAPTER 5 SUMMARY AND CONCLUSION

### SUMMARY

This program is using machine learning techniques to build and evaluate models for breast cancer classification.

First, necessary libraries including numpy, matplotlib, pandas, seaborn, and warnings are imported. The dataset is then loaded from a CSV file and the first few rows of the dataset are viewed using the 'head()' function. The 'info()' function is used to get information about the dataset, such as the number of rows, columns, and data types. The 'isna().sum()' function is used to check if there are any missing values in the dataset, and the 'shape' function is used to get the dimensions of the dataset.Columns with missing values are dropped using the 'dropna()' function, and the new shape of the dataset is displayed. Descriptive statistics of the dataset are generatedusing the 'describe()' function, and the count of values in the "diagnosis" column is displayed using the 'value\_counts()' function. A count plot of the "diagnosis" column is created using the 'countplot()' function from the seaborn library.The 'LabelEncoder' function from the sklearn.preprocessing library is used to encode the "diagnosis" column, and the first four columns of the dataset are plotted using a pairplot using the 'pairplot()' function from the seaborn library. The correlation matrix of the dataset is generated using the 'corr()' function from pandas, and a heatmap of the correlation matrix is created using the 'heatmap()' function from seaborn.The dataset is then split into training and testing sets using the 'train\_test\_split()' function from sklearn.model\_selection. The features of the training and testing sets are scaled using the 'StandardScaler' function from sklearn.preprocessing.A function is defined that trains and returns two models - a Random Forest Classifier and a Decision Tree Classifier - using the 'RandomForestClassifier' and 'DecisionTreeClassifier' functions from the sklearn.ensemble and sklearn.tree libraries, respectively. The models' accuracy on the training data is printed, and themodels are returned.

For each model, the confusion matrix is calculated using the 'confusion\_matrix()' function from sklearn.metrics. The true positive, true negative, false positive, and false negative values are calculated from the confusion matrix. A classification report is then generated using the 'classification\_report()' function from sklearn.metrics, and the accuracy score is calculated using the 'accuracy\_score()' function.

Finally, the models are used to predict values on the test data, and the predicted and actual values are displayed.

### CONCLUSION

Using the Breast Cancer Coimbra (Diagnostic) Database, we can create a classifier that can help diagnose patients and predict the likelihood of a breast cancer. A few machine learning techniques will be explored. In this exercise, Support Vector Machine is being implemented with 99% accuracy.

There are mant many treatments for apatient based on breast cancer stage; data mining and machine learning can be very good help in deciding the line of treatment to be followed by extracting knowledge from such suitable databases. Breast Cancer represents one of the diseases that makes highest number of deaths every year. At present, only few accurate prognostic and predictive factors are used clinically for managing the patients with breast cancer.

Here, by making use of Clustering with Level Set approach, high accuracy can be achieved in detection of effected cell shapes with exact marking on detected contours. The proposed system helps to enhance the performance of mammogram retrieval by selecting optimal features.

# REFERNCES

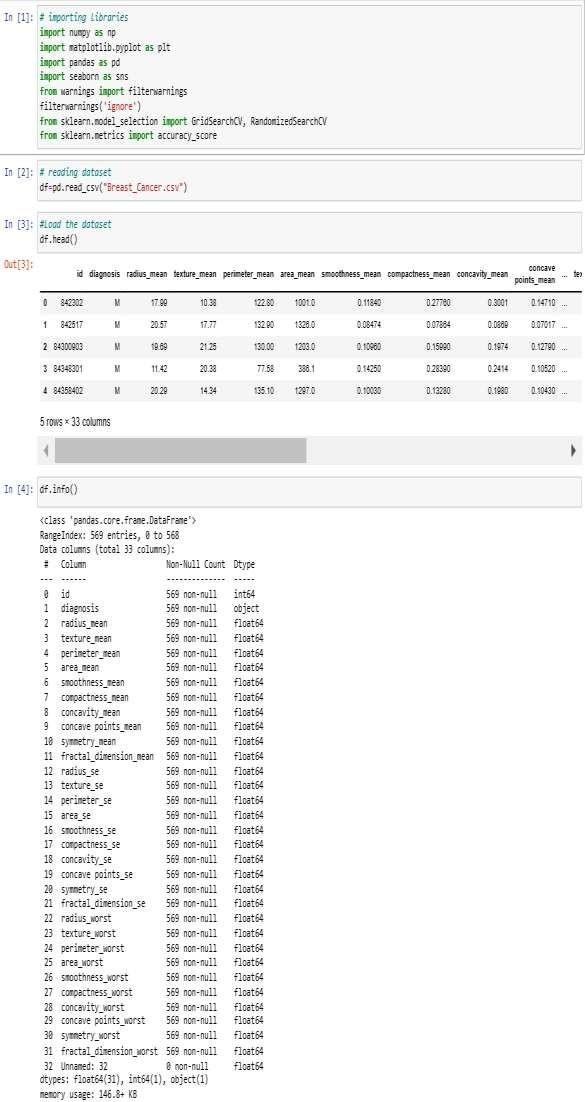
1. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, et al. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. CA Cancer J Clin. 2021 May;71(3):209–49.
2. Wang L. Early Diagnosis of Breast Cancer. Sensors (Basel). 2017 Jul 5;17(7):E1572.
3. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [[htt](http://archive.ics.uci.edu/ml)p[://archive.ics.uci.edu/ml](http://archive.ics.uci.edu/ml)]. Irvine, CA: University of California, School of Information and Computer Science.
4. Ford D, Easton DF, Stratton M, Narod S, Goldgar D, Devilee P, et al. Genetic heterogeneity and penetrance analysis of the BRCA1 and BRCA2 genes in breast cancer families. The Breast Cancer Linkage Consortium. Am J Hum Genet. 1998 Mar;62(3):676–89.
5. Jackins V, Vimal S, Kaliappan M, Lee MY. AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. J Supercomput. 2021 May 1;77(5):5198–219.
6. Sarica A, Cerasa A, Quattrone A. Random Forest Algorithm for the Classification of Neuroimaging Data in Alzheimer’s Disease: A Systematic Review. Frontiers in Aging Neuroscience [Internet]. 2017 [cited 2022 Apr 14];9. Available from: <https://www.frontiersin.org/article/10.3389/fnagi.2017.00329>
7. Byeon H. Is the Random Forest Algorithm Suitable for Predicting Parkinson’s Disease with Mild Cognitive Impairment out of Parkinson’s Disease with Normal Cognition? Int J Environ Res Public Health. 2020 Apr;17(7):2594.
8. Khalilia M, Chakraborty S, Popescu M. Predicting disease risks from highly imbalanced data using random forest. BMC Medical Informatics and Decision Making. 2011 Jul 29;11(1):51.
9. Alam MdZ, Rahman MS, Rahman MS. A Random Forest based predictor for medical data classification using feature ranking. Informatics in Medicine Unlocked. 2019 Jan 1;15:100180.
10. Anji Reddy V, Soni B. Breast Cancer Identification and Diagnosis Techniques.In: Rout JK, Rout M, Das H, editors. Machine Learning for Intelligent Decision
11. Science [Internet]. Singapore: Springer; 2020 [cited 2022 Apr 14]. p. 49–70. (Algorithms for Intelligent Systems).

[12] [11] Available from: https://doi.org/10.1007/978-981-15-3689-2\_3

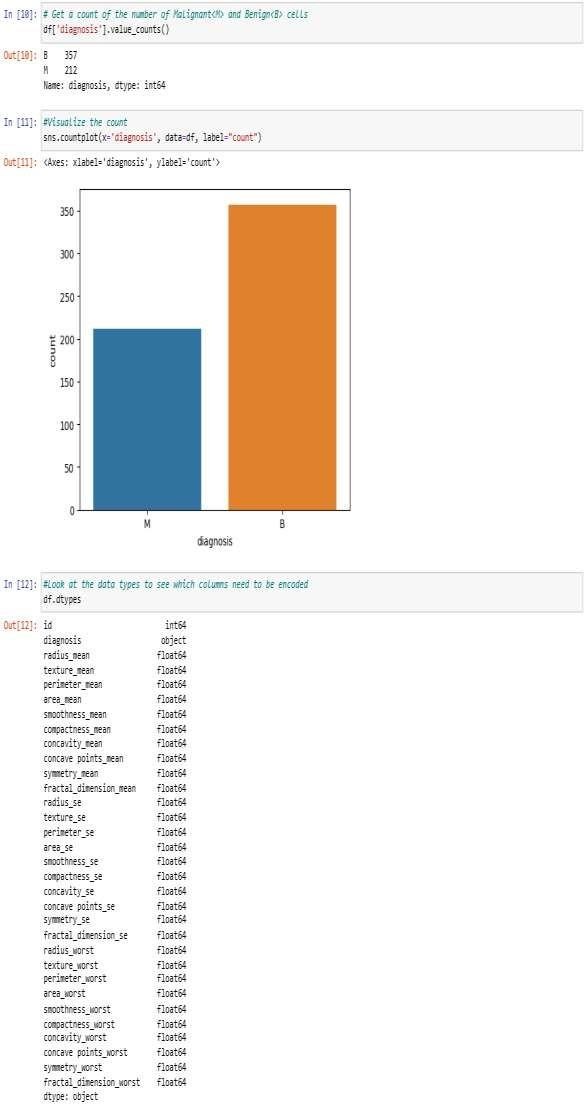
1. Vaka AR, Soni B, K. SR. Breast cancer detection by leveraging Machine Learning. ICT Express. 2020 Dec 1;6(4):320–4.
2. Naji MA, Filali SE, Aarika K, Benlahmar EH, Abdelouhahid RA, Debauche O. Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis. Procedia Computer Science. 2021 Jan 1;191:487–92.
3. Priyanka KS. A Review Paper on Breast Cancer Detection Using Deep Learning. IOP Conf Ser: Mater Sci Eng. 2021 Jan;1022(1):012071.
4. Azar AT, El-Metwally SM. Decision tree classifiers for automated medical diagnosis. Neural Comput & Applic. 2013 Dec 1;23(7):2387– 403.
5. Desai M, Shah M. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). Clinical eHealth. 2021 Jan 1;4:1–11.
6. Polat K, Güneş S. Breast cancer diagnosis using least square support vector machine. Digital Signal Processing. 2007 Jul 1;17(4):694–701.
7. Sarkar M, Leong TY. Application of K-nearest neighbors algorithm on breast cancer diagnosis problem. Proc AMIA Symp. 2000;759–63.

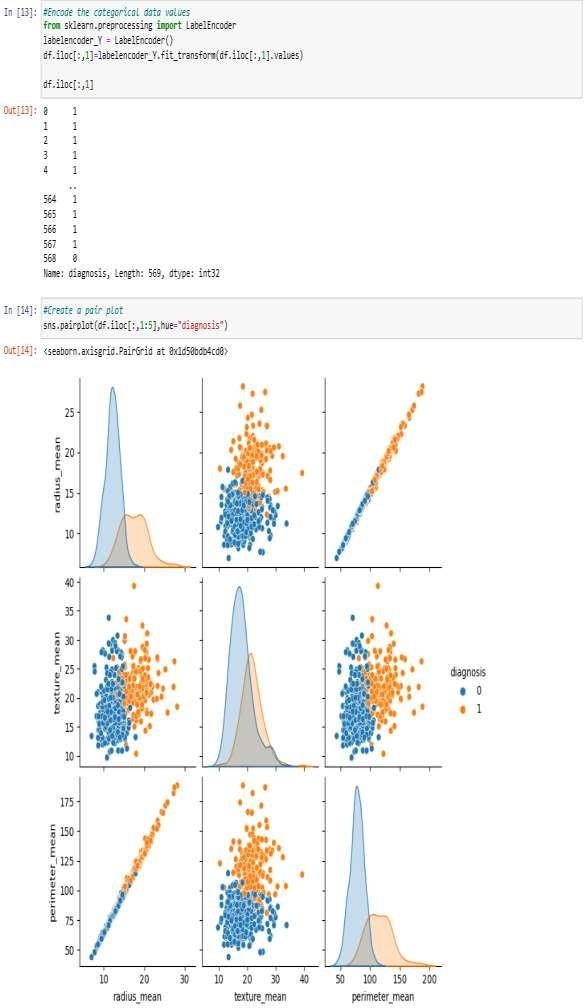
# APPENDIX

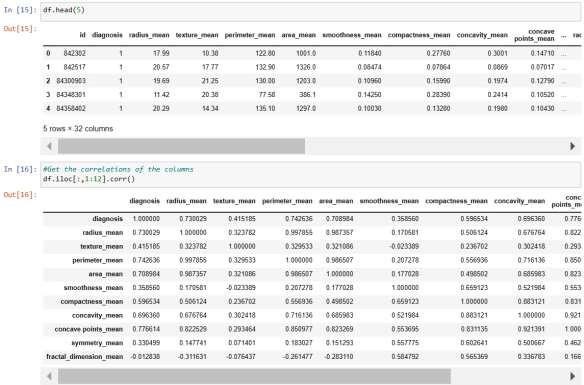
### SCREENSHOTS AND OUTPUTS:

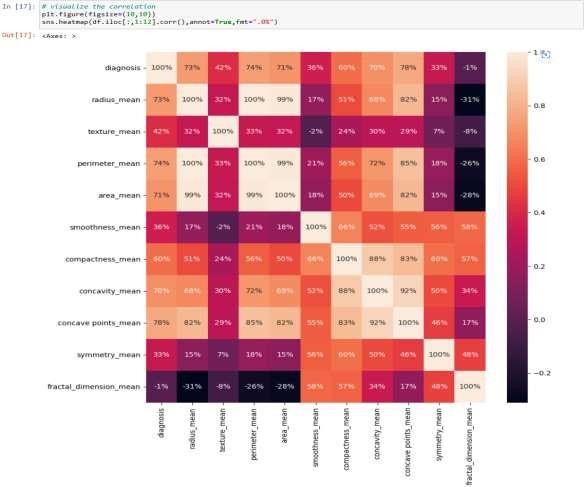




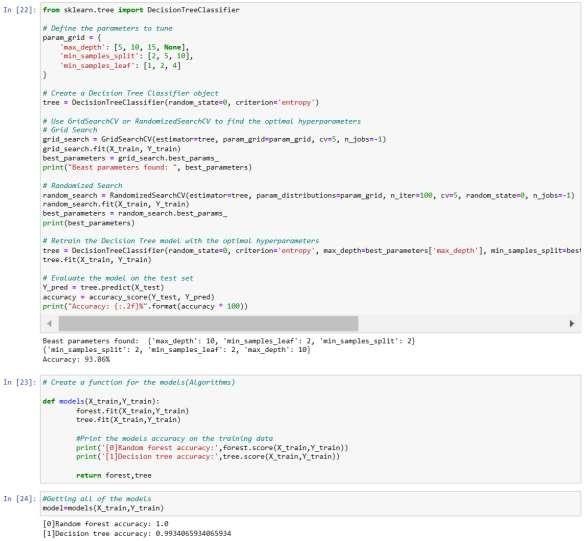














### SOURCECODE:

# importing libraries import numpy as np

import matplotlib.pyplot as plt import pandas as pd

import seaborn as sns

from warnings import filterwarnings filterwarnings('ignore')

from sklearn.model\_selection import GridSearchCV, RandomizedSearchCV from sklearn.metrics import accuracy\_score

# reading dataset df=pd.read\_csv("Breast\_Cancer.csv")

#Load the dataset df.head()

df.info()

#count the numer of empty (NAN) values in each column df.isna().sum()

# count the number of rows and columns within the datset df.shape

#Drop the "Unnamed: 32" column with all missing values df=df.dropna(axis=1)

#Get the new count of the number of rows and columns df.shape

# describe the dataset df.describe()

# Get a count of the number of Malignant<M> and Benign<B> cells df['diagnosis'].value\_counts()

#Visualize the count

sns.countplot(x='diagnosis', data=df, label="count")

#Look at the data types to see which columns need to be encoded df.dtypes

#Encode the categorical data values

from sklearn.preprocessing import LabelEncoder labelencoder\_Y = LabelEncoder() df.iloc[:,1]=labelencoder\_Y.fit\_transform(df.iloc[:,1].values)

df.iloc[:,1]

#Create a pair plot sns.pairplot(df.iloc[:,1:5],hue="diagnosis")

df.head(5)

#Get the correlations of the columns df.iloc[:,1:12].corr()

# visualize the correlation plt.figure(figsize=(10,10))

sns.heatmap(df.iloc[:,1:12].corr(),annot=True,fmt=".0%")

# split the dataset into Independent(X) and dependent(Y) datasets X=df.iloc[:,2:31].values

Y=df.iloc[:,1].values

# split the dataset into 75% trainning and 25% test data from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=0.20,random\_state=0)

#Scale the data (Freature Scaling)

from sklearn.preprocessing import StandardScaler sc = StandardScaler() X\_train=StandardScaler().fit\_transform(X\_train) X\_test=StandardScaler().fit\_transform(X\_test)

from sklearn.ensemble import RandomForestClassifier

# Define the hyperparameter grid param\_grid = {

'max\_depth': [5, 10, 15, None],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 2, 4]

}

# Create a Random Forest Classifier object

forest = RandomForestClassifier(random\_state=0, criterion='entropy')

# Use GridSearchCV or RandomizedSearchCV to find the optimal hyperparameters

# Grid Search

grid\_search = GridSearchCV(estimator=forest, param\_grid=param\_grid, cv=5, n\_jobs=-1)

grid\_search.fit(X\_train, Y\_train) best\_parameters = grid\_search.best\_params\_

print("Best parameters found: ", best\_parameters)

# Use Randomized Search

random\_search = RandomizedSearchCV(estimator=forest, param\_distributions=param\_grid, n\_iter=100, cv=5, random\_state=0, n\_jobs=-1)

random\_search.fit(X\_train, Y\_train) best\_parameters = random\_search.best\_params\_ print(best\_parameters)

# Train the model using the forest = RandomForestClassifier(random\_state=0, criterion='entropy', )

forest = RandomForestClassifier(random\_state=0, criterion='entropy', max\_depth=best\_parameters['max\_depth'], min\_samples\_split=best\_parameters['min\_samples\_split'], min\_samples\_leaf=best\_parameters['min\_samples\_leaf'])

forest.fit(X\_train, Y\_train)

# Evaluate the model on the test set Y\_pred = forest.predict(X\_test)

accuracy = accuracy\_score(Y\_test, Y\_pred) print("Accuracy: {:.2f}%".format(accuracy \* 100))

from sklearn.tree import DecisionTreeClassifier

# Define the parameters to tune param\_grid = {

'max\_depth': [5, 10, 15, None],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 2, 4]

}

# Create a Decision Tree Classifier object

tree = DecisionTreeClassifier(random\_state=0, criterion='entropy')

# Use GridSearchCV or RandomizedSearchCV to find the optimal hyperparameters

# Grid Search

grid\_search = GridSearchCV(estimator=tree, param\_grid=param\_grid, cv=5, n\_jobs=-1)

grid\_search.fit(X\_train, Y\_train) best\_parameters = grid\_search.best\_params\_

print("Beast parameters found: ", best\_parameters)

# Randomized Search

random\_search = RandomizedSearchCV(estimator=tree, param\_distributions=param\_grid, n\_iter=100, cv=5, random\_state=0, n\_jobs=-1)

random\_search.fit(X\_train, Y\_train) best\_parameters = random\_search.best\_params\_ print(best\_parameters)

# Retrain the Decision Tree model with the optimal hyperparameters

tree = DecisionTreeClassifier(random\_state=0, criterion='entropy', max\_depth=best\_parameters['max\_depth'], min\_samples\_split=best\_parameters['min\_samples\_split'], min\_samples\_leaf=best\_parameters['min\_samples\_leaf'])

tree.fit(X\_train, Y\_train)

# Evaluate the model on the test set Y\_pred = tree.predict(X\_test)

accuracy = accuracy\_score(Y\_test, Y\_pred) print("Accuracy: {:.2f}%".format(accuracy \* 100))

# Create a function for the models(Algorithms)

def models(X\_train,Y\_train): forest.fit(X\_train,Y\_train) tree.fit(X\_train,Y\_train)

#Print the models accuracy on the training data print('[0]Random forest accuracy:',forest.score(X\_train,Y\_train)) print('[1]Decision tree accuracy:',tree.score(X\_train,Y\_train))

return forest,tree

#Getting all of the models model=models(X\_train,Y\_train)

#Test the accuracy of the model on the testing data using a confusion matrix from sklearn.metrics import confusion\_matrix

for i in range( len(model) ) :

cm = confusion\_matrix(Y\_test, model[i].predict(X\_test))

TP = cm[1][1]

TN = cm[0][0]

FN = cm[1][0]

FP = cm[0][1]

print(cm)

print('Testing Accuracy = ', (TP+TN)/ (TP+TN+FP+FN))

# Print a classification report

from sklearn.metrics import accuracy\_score from sklearn.metrics import classification\_report

for i in range(len(model)): print("Model",i)

print(classification\_report(Y\_test,model[0].predict(X\_test))) print(accuracy\_score(Y\_test, model[i].predict(X\_test))) print()

# prediction of random-forest pred=model[0].predict(X\_test) print('Predicted values:') print(pred)

print('Actual values:') print(Y\_test)