MALIGNANT AND BENIGN BREAST CANCER CLASSIFICATION USING MACHINE LEARNING ALGORITHMS

**Abstract:**

At the moment, the most prevalent form of cancer diagnosed in women across the globe is breast cancer. It develops in the breast tissue and is one of the most frequent causes of womens death. This cancer can be cured if it is diagnosed at preliminary stage. Malignant and benign are two types of tumor found in case of breast cancer. Malignant tumors are deadly as their rate of growth is much higher than benign tumors. So, early identification of tumor type is pivotal for the appropriate treatment of a patient having breast cancer. In this work, Wisconsin Breast Cancer Dataset has been used which was collected from UCI repository. Our goal is to analyze the dataset and evaluate the performance of various machine learning algorithms for predicting breast cancer. Here Decision Tree, and Random Forest classifiers have been implemented for classifying tumors into benign and malignant. The accuracy of each algorithm is calculated and compared to find the most suitable one.These classifiers can be used to build an automatic diagnostic system for preliminary diagnosis of breast cancer.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | 1 |
| **1.** | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   1.1.1 THE MACHINE LEARNING SYSTEM  1.1.2 FUNDAMENTAL  1.2 JUPYTER  1.3 MACHINE LEARNING  1.4 CLASSIFICATION TECHNIQUES  1.4.1 NEURAL NETWORK AND DEEP LEARNING   * 1. OBJECTIVE AND SCOPE OF THE PROJECT   1.6 EXISTING SYSTEM  1.6.1 DISADVANTAGES OF EXISTING SYSTEM  1.6.2 LITERATURE SURVEY  1.7 PROPOSED SYSTEM  1.7.1 PROPOSED SYSTEM ADVANTAGES | 4  4  6  9  12  12  13  14  17 |
| **2.** | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 INTRODUCTION  2.2 DETAILED DIAGRAM  2.2.1 FRONT END DESIGN  2.2.2 BACK END FLOW  2.3 SOFTWARE SPECIFICATION  2.3.1 HARDWARE SPECIFICATION  2.3.2 SOFTWARE SPECIFICATION  2.4 MODULE DESCRIPTION  2.4.1 DATA PRE PROCESSING  2.4.2 DATA CLEANING  2.4.3 DATA SPLITTING  2.4.4 TRAINING AND TESTING  2.4.5 PERFORMANCE METRICES  2.4.6 CONFUSION MATRIX  2.5 MODULE DIAGRAM  2.5.1 SYSEM ARCTHITURE  2.5.2 USECASE DIAGRAM  2.5.3 CLASS DIAGRAM  2.5.4 ACTIVITY DIAGRAM  2.5.5 SEQUENCE DIAGRAM | 28  29  29  30  30  32  33  33  34 |
| **3.** | **CHAPTER 3 : SOFTWARE SPECIFICATION**  3.1 general  3.2 ANACONDA  3.3 PYTHON  3.3.1 SCIENTIFIC AND NUMERIC COMPUTING  3.3.2 CREATING SOFTWARE PROTOTYPES  3.3.3 GOOD LANGUAGE TO TEACH PROGRAMMING | 35  37  40    40  40 |
| **4.** | **CHAPTER 4 : IMPLEMENTATION**  4.1 GENERAL  4.2 IMPLEMENTATION CODING  4.3 SNAPSHOTS | 41  41  46 |
| **5.** | **CHAPTER 5 : CONCLUSION & REFERENCES**  5.1 CONCLUSION  5.2 APPLICATION  5.2 REFERENCES | 52  52  53 |

**CHAPTER I**

**INTRODUCTION**

**1.2 Jupyter**

Jupyter, previously known as IPython Notebook, is a web-based, interactive development environment. Originally developed for Python, it has since expanded to support over 40 other programming languages including Julia and R.

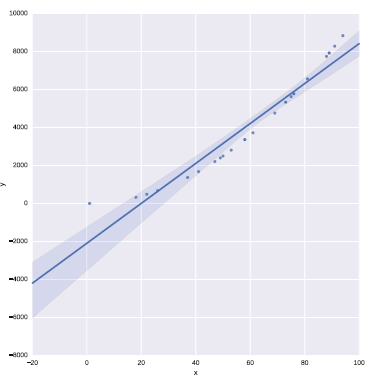
Jupyter allows for *notebooks*to be written that contain text, live code, images, and equations. These notebooks can be shared, and can even be hosted on GitHubfor free.

For each section of this tutorial, you can download a Juypter notebook that allows you to edit and experiment with the code and examples for each topic. Jupyter is part of the Anaconda distribution; it can be started from the command line using the jupyter command:



**1.3 Machine Learning**

We will now move on to the task of machine learning itself. In the following sections we will describe how to use some basic algorithms, and perform regression, classification, and clustering on some freely available medical datasets concerning breast cancer and diabetes, and we will also take a look at a DNA microarray dataset.

****

**SciKit-Learn**

SciKit-Learn provides a standardised interface to many of the most commonly used machine learning algorithms, and is the most popular and frequently used library for machine learning for Python. As well as providing many learning algorithms, SciKit-Learn has a large number of convenience functions for common preprocessing tasks (for example, normalisation or *k*-fold cross validation).

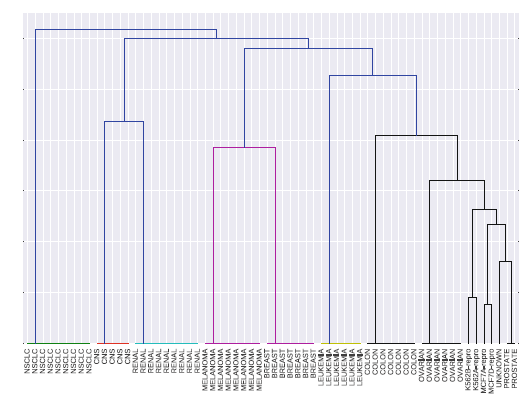
SciKit-Learn is a very large software library.

**Clustering**

Clustering algorithms focus on ordering data together into groups. In general clustering algorithms are unsupervised—they require no **y** response variable as input. That is to say, they attempt to find groups or clusters within data where you do not know the label for each sample. SciKit-Learn have many clusteringalgorithms, but in this section we will demonstrate hierarchical clustering on a DNA expression microarray dataset using an algorithm from the SciPy library.

We will plot a visualisation of the clustering using what is known as a dendrogram, also using the SciPy library.

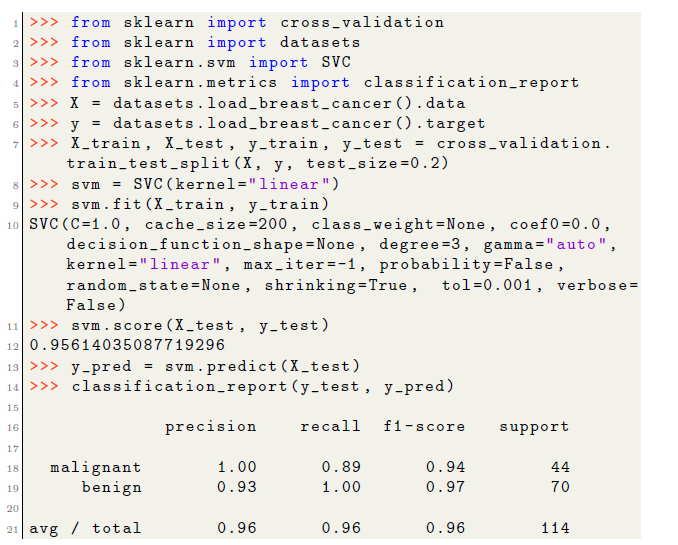
The goal is to cluster the data properly in logical groups, in this case into the cancer types represented by each sample’s expression data. We do this using agglomerative hierarchical clustering, using Ward’s linkage method:



**1.4 Classification**

weanalysed data that was **unlabelled**—we did not know to what class a sample belonged (known as unsupervised learning). In contrast to this, a supervised problem deals with **labelled** data where are aware of the discrete classes to which each sample belongs. When we wish to predict which class a sample belongs to, we call this a classification problem. SciKit-Learn has a number of algorithms for classification, in this section we will look at the Support Vector Machine.

We will work on the Wisconsin breast cancer dataset, split it into a training set and a test set, train a Support Vector Machine with a linear kernel, and test the trained model on an unseen dataset. The Support Vector Machine model should be able to predict if a new sample is malignant or benign based on the features of a new, unseen sample:



You will notice that the SVM model performed very well at predicting the malignancy of new, unseen samples from the test set—this can be quantified nicely by printing a number of metrics using the classification report function. Here, the precision, recall, and *F*1 score (*F*1 = 2*·* precision*·*recall*/*precision+recall) for each class is shown. The support column is a count of the number of samples for each class.

Support Vector Machines are a very powerful tool for classification. They work well in high dimensional spaces, even when the number of features is higher than the number of samples. However, their running time is quadratic to the number of samples so large datasets can become difficult to train. Quadratic means that if you increase a dataset in size by 10 times, it will take 100 times longer to train.

Last, you will notice that the breast cancer dataset consisted of 30 features. This makes it difficult to visualize or plot the data. To aid in visualization of highly dimensional data, we can apply a technique called dimensionality reduction.

**Dimensionality Reduction**

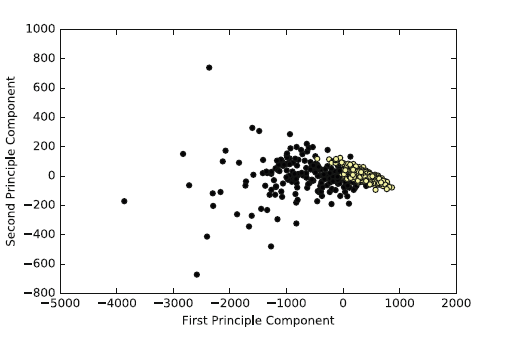
Another important method in machine learning, and data science in general, is dimensionality reduction. For this example, we will look at the Wisconsin breast cancer dataset once again. The dataset consists of over 500 samples, where each sample has 30 features. The features relate to images of a fine needle aspirate of breast tissue, and the features describe the characteristics of the cells present in the images. All features are real values. The target variable is a discrete value (either malignant or benign) and is therefore a classification dataset.

You will recall from the Iris example in Sect. 7.3 that we plotted a scatter matrix of the data, where each feature was plotted against every other feature in the dataset to look for potential correlations (Fig. 3). By examining this plot you could probably find features which would separate the dataset into groups. Because the dataset only had 4 features we were able to plot each feature against each other relatively easily. However, as the numbers of features grow, this becomes less and less feasible, especially if you consider the gene expression example in Sect. 9.4 which had over 6000 features.

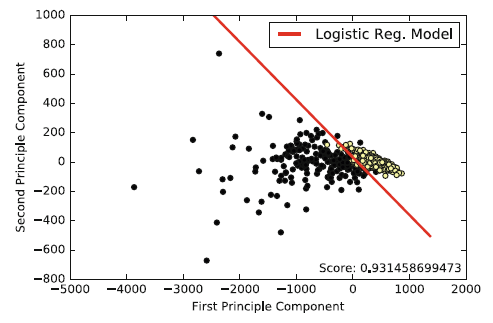
One method that is used to handle data that is highly dimensional is Principle Component Analysis, or PCA. PCA is an unsupervised algorithm for reducing the number of dimensions of a dataset. For example, for plotting purposes you might want to reduce your data down to 2 or 3 dimensions, and PCA allows

you to do this by generating components, which are combinations of the original features, that you can then use to plot your data.

PCA is an unsupervised algorithm. You supply it with your data, **X**, and you specify the number of components you wish to reduce its dimensionality to. This is known as transforming the data:



Again, you would not use this model for new data—in a real world scenario, you would, for example, perform a 10-fold cross validation on the dataset, choosing the model parameters that perform best on the cross validation. This model would be much more likely to perform well on new data. At the very least, you would randomly select a subset, say 30% of the data, as a test set and train the model on the remaining 70% of the dataset. You would evaluate the model based on the score on the test set and not on the training set

.

**1.4.2 NEURAL NETWORKS AND DEEP LEARNING**

While a proper description of neural networks and deep learning is far beyond the scope of this chapter, we will however discuss an example use case of one of the most popular frameworks for deep learning: Keras4.

In this section we will use Keras to build a simple neural network to classify theWisconsin breast cancer dataset that was described earlier. Often, deep learning algorithms and neural networks are used to classify images—convolutional neural networks are especially used for image related classification. However,

they can of course be used for text or tabular-based data as well. In this we will build a standard feed-forward, densely connected neural network and classify a text-based cancer dataset in order to demonstrate the framework’susage.

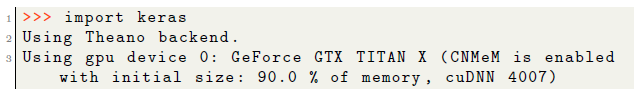
In this example we are once again using the Wisconsin breast cancer dataset, which consists of 30 features and 569 individual samples. To make it more challenging for the neural network, we will use a training set consisting of only 50% of the entire dataset, and test our neural network on the remaining 50% of the data.

Note,Keras is not installed as part of the Anaconda distribution, to install it use pip:



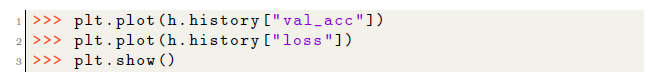
Keras additionally requires either Theano or TensorFlow to be installed. In the examples in this chapter we are using Theano as a backend, however the code will work identically for either backend. You can install Theano using pip, but it has a number of dependencies that must be installed first. Refer to the Theano and TensorFlow documentation for more information [12].

Keras is a modular API. It allows you to create neural networks by building a stack of modules, from the input of the neural network, to the output of the neural network, piece by piece until you have a complete network. Also, Keras can be configured to use your Graphics Processing Unit, or GPU. This makes training neural networks far faster than if we were to use a CPU. We begin by importing Keras:



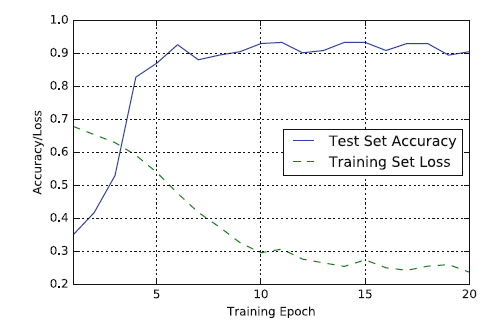
We may want to view the network’s accuracy on the test (or its loss on the training set) over time (measured at each epoch), to get a better idea how well it is learning. An epoch is one complete cycle through the training data.

Fortunately, this is quite easy to plot as Keras’ fit function returns a history object which we can use to do exactly this:



This will result in a plot similar to that shown. Often you will also want to plot the loss on the test set and training set, and the accuracy on the test set and training set.

Plotting the loss and accuracy can be used to see if you are over fitting (you experience tiny loss on the training set, but large loss on the test set) and to see when your training has plateaued.



**1.5 OBJECTIVE STATEMENT:**

Breast cancer is the most common type of cancer in women. When cancers are found early, they can often be cured. There are some devices that detect the breast cancer but many times they lead to false positives, which results is patients undergoing painful, expensive surgeries that were not even necessary. These type of cancers are called **benign** which do not require surgeries and we can reduce these unnecessary surgeries by using Machine Learning. We take a dataset of the previous breast cancer patients and train the model to predict whether the cancer is **benign** or **malignant**. These predictions will help doctors to do surgeries only when the cancer is malignant, thus reducing the unnecessary surgeries for woman.

**SCOPE OF THE PROJECT**

* To predict classification accurately.
* To reduce the missclassification.
* To improve the model performance

**1.6 EXISTING SYSTEM**

Researchers are currently trying to detect markers of such genetic damage. Newly developed methods for growing breast cells in culture and new automated systems for screening large numbers of genes or proteins in cells should aid this endeavor.

Currently, abnormalities detected with mammography are crudely classified as malignant or benign based on their structural appearance under the microscope, and not by what genetic changes they may have. If reliable markers of breast cancer progression can be identified, tests for such markers may play a role in determining whether doctors should treat the pre-malignant abnormalities and early-stage lesions that are now so commonly identified by screening mammography.

Many of these abnormalities may not progress to life-threatening disease. The discovery and development of molecular markers for breast cancer, consequently, might help reduce the “overtreatment” of harmless abnormalities. They might also be able to identify women who should undergo more frequent screening or consider prophylactic treatment (for example, mastectomy or tamoxifen), or those who might benefit from newer imaging technologies. If and when these molecular markers prove useful for diagnosing or predicting the aggressiveness of breast cancers, researchers could then also examine their usefulness in breast cancer screening.

There are many ways that gene or protein screens could be used for breast cancer detection. Researchers are trying to develop specialized imaging systems that can use “smart” contrast agents to reveal telltale genes that may be activated in cancerous breast tissue, or other biochemical markers of early breast cancer. Such systems might eventually be used in breast cancer screening, but their development is too preliminary at this point to assess their usefulness for this purpose.

**1.6.1 DISADVANTAGES OF EXISTING SYSTEM**

* Other researchers are trying to develop screening tests for tumor markers or tumor cells in breast fluid or blood serum.
* Breast fluid can be obtained from women who are not pregnant or breast-feeding is required.
* In addition, researchers have recently developed a device that is inserted into the nipple and uses salt water to flush breast duct cells out of some of the breast ducts (a process known as ductal lavage).
* The FDA recently approved this device for breast fluid removal. More studies need to be done to assess the usefulness of this device in breast cancer screening or diagnosis

**1.6.2 LITERATURE SURVEY**

**1.TITLE:** **Breast Cancer Classification and Prediction using Machine Learning**

**Author:** Nikita Rane, Jean Sunny, Rucha Kanade

**Year:2021**

**Abstract:**

Breast cancer is a dominant cancer in women worldwide and is increasing in developing countries where the majority of cases are diagnosed in late stages. The projects that have already been proposed show a comparison of machine learning algorithms with the help of different techniques like the ensemble methods, data mining algorithms or using blood analysis etc. This paper proposed now presents a comparison of six machine learning (ML) algorithms: Naive Bayes (NB), Random Forest (RT), Artificial Neural Networks (ANN), Nearest Neighbour (KNN), Support Vector Machine (SVM) and Decision Tree (DT) on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset which is extracted from a digitised image of an MRI. For the implementation of the ML algorithms, the dataset was partitioned into the training phase and the testing phase. The algorithm with the best results will be used as the backend to the website and the model will then classify the cancer as benign or malignant.

# 2.TITLE: Breast Cancer Prediction Using Machine Learning Algorithms

**Author**:Dr.B.Santhosh Kumar, T.Daniya, Dr. J.Ajayan

**Year:2020**

**Abstract:**

Breast cancer has turned into a typical danger to ladies' physical and emotional well-being. The point of this task is to build up a framework that takes a cancer medicinal dataset as an info and after that performs examination of the dataset for creating results that aides in causing the therapeutic researchers to comprehend about the status of illness. Persons can only be saved having cancer if it is detected in first two stages (stage1 and stage2) otherwise we can’t survive them. To give suitable treatment to the patients, side effects must be contemplated appropriately and a programmed forecast framework is required which will order the tumor into kindhearted or harmful. The manual elucidation of the disease CT pictures is tedious and basic, to conquer this trouble, we are utilizing arrangement calculation which would order the malignant growth pictures rapidly and viably. This paper will concentrate on the information pre-handling stage so as to clean the crude information and set up the last informational index and it is given as input to the K-Nearest Neighbour classification algorithm to predict the cancer in early stages.

# 3.TITLE: Developing A Web based System for Breast Cancer Prediction using XGboost Classifier

**Author:**  Nayan Kumar Sinha , Menuka Khulal , Manzil Gurung , Arvind Lal

**Year:2020**

**Abstract:**

 In todays world cancer is the most common diseases which lead to greatest number of death. Cancer is not one disease; it is a group of more than 100 different and distinctive diseases. Cancer can involve in any tissue of the body and have many different forms and in each body part. Breast Cancer is a grim disease and it is the only type of cancer that is widespread among women worldwide. As the diagnosis of this disease manually takes long hours and the lesser availability of systems, there is a need to develop the automatic diagnosis system for early detection of cancer. So in this project we are developing a web based diagnosis system for which we have done the comparative study of the supervised machine learning classifiers to get to know which classifier is giving the best accuracy. For that we have taken dataset from the Wisconsin breast cancer database (WBCD) which is the benchmark database for comparing the results through different algorithms. In which we will use following classification techniques of machine learning like Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), Adaboost Classifier and XGboost Classifier for the classification of benign and malignant tumor in which the machine is learned from the past data and can predict the category of new input.

# 4.TITLE: Breast Cancer Detection and Prediction using Machine Learning

# Author:  [Anoy Chowdhury](https://www.researchgate.net/profile/Anoy-Chowdhury-2)

# Year:2020

**Abstract:**

# According to the world health organization (WHO) Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year, and also causes the greatest number of cancer-related deaths among women. In 2018, it is estimated that 627,000 women died from breast cancer-that is approximately 15% of all cancer deaths among women. While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally. In order to improve breast cancer outcomes and survival, early detection is critical. There are two early detection strategies for breast cancer: early diagnosis and screening. Limited resource settings with weak health systems where the majority of women are diagnosed in late stages should prioritize early diagnosis programs based on awareness of early signs and symptoms and prompt referral to diagnosis and treatment. Early diagnosis strategies focus on providing timely access to cancer treatment by reducing barriers to care and/or improving access to effective diagnosis services. The goal is to increase the proportion of breast cancers identified at an early stage, allowing for more effective treatment to be used and reducing the risks of death from breast cancer. Since early detection of cancer is key to effective treatment of breast cancer we use various machine learning algorithms to predict if a tumor is benign or malignant, based on the features provided by the data.

# 5.TITLE: Breast Cancer Prediction System

# Author: Madhu Kumaria , Vijendra Singh

# Year:2019

# Abstract

# Breast cancer became the major source of mortality between women. The accessibility of healthcare datasets and data analysis promote the researchers to apply study in extracting unknown pattern from healthcare datasets. The intention of this study is to design a prediction system that can predict the incidence of the breast cancer at early stage by analyzing smallest set of attributes that has been selected from the clinical dataset. Wisconsin breast cancer dataset (WBCD) have been used to conduct the proposed experiment. The potential of the proposed method is obtained using classification accuracy which was obtained by comparing actual to predicted values. The outcome confirms that the maximum classification accuracy (99.28%) is achieved for this study.

* 1. **PROPOSED SYSTEM**

Many models have been proposed which use different feature sets and methods of machine learning to diagnose breast cancer. The scarcity of large datasets and inequality between negative and positive classes are the main challenges in the research area of breast cancer prediction. the prime goal of the analysis is to detect the algorithm that operates quicker, more reliably, and more effectively in breast cancer prediction. With a precision of 99.76%, Random Forest surpasses all other algorithms. In authors have performed comparative analysis of the precise breast cancer prediction offered by existing ML algorithms. The dataset used in these papers is WBCD. In have used Random Forest classifier for the identification and prediction of breast cancer to determine whether or not the person has breast cancer. This offers the highest identification accuracy since both classification and regression approaches are used for Random Forest algorithm. A study on breast cancer was provided by the authors to develop predictive models for breast cancer survival. In this paper, three breast cancer survivability prediction models were applied to two classes: benign and malignant cancer. In highlighted all previous research on ML algorithms used for breast cancer determination. They suggested that the problem of limited available dataset can be solved by data augmentation techniques. In authors presented a technique that can be used to detect and identify cell morphology in automated systems that carry out the classification using computer-aided mammogram image features. In authors have compared various classification and clustering algorithms in the survey. The result shows that the algorithms for classification are better predictors than the clustering algorithms.

**1.7.1 PROPOSED SYSTEM ADVANTAGES**

* Accuracy may be improved.
* Data extraction process is highly load.

.

**Algorithms:**

**DECISION TREE**

Let’s now start with Decision tree’s and I assure you this is probably the easiest algorithm in Machine Learning. There’s not much mathematics involved here. Since it is very easy to use and interpret it is one of the most widely used and practical methods used in Machine Learning.

## What is a Decision Tree?

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

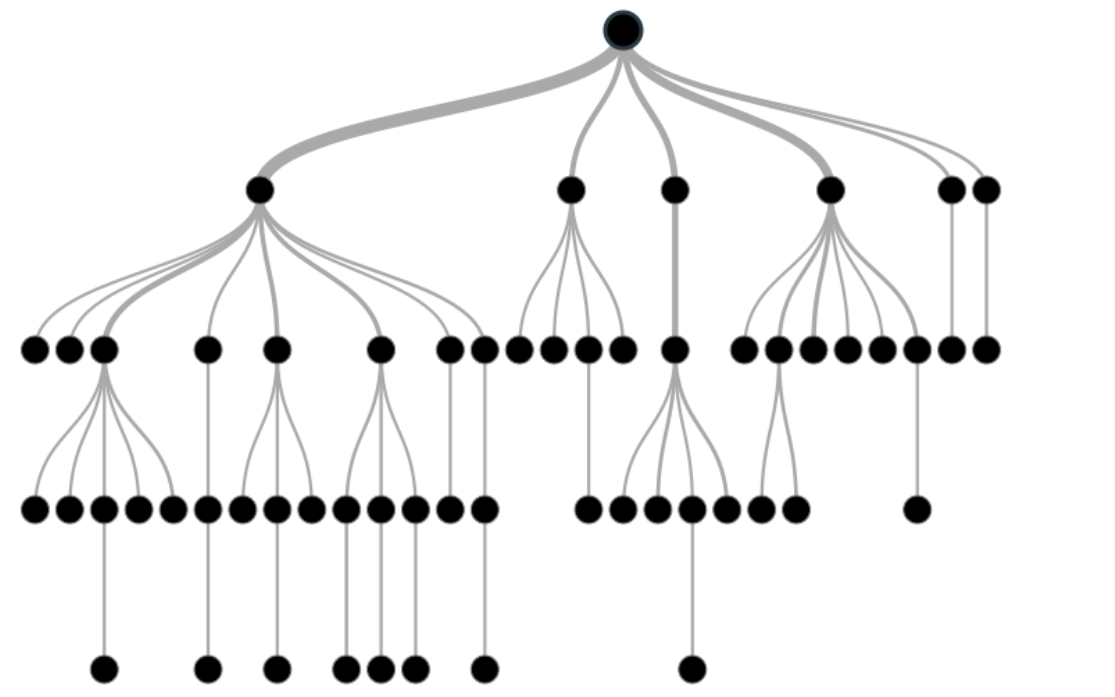


Image 1

Before learning more about decision trees let’s get familiar with some of the terminologies.

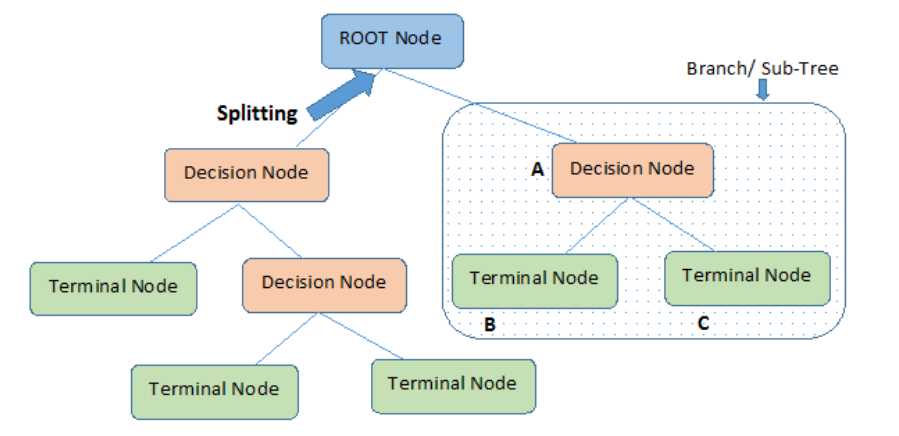
**Root Nodes**– It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

**Decision Nodes** – the nodes we get after splitting the root nodes are called Decision Node

**Leaf Nodes**– the nodes where further splitting is not possible are called leaf nodes or terminal nodes

**Sub-tree** – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.

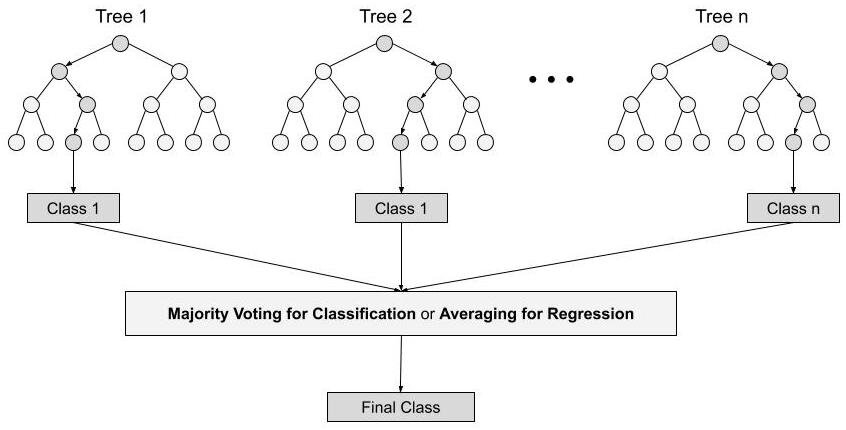
**Pruning** – is nothing but cutting down some nodes to stop overfitting.



# RANDOMFOREST

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.



**Steps involved in random forest algorithm:**

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging***for Classification and regression respectively.

**CHAPTER 2**

**2.1 INTRODUCTION**

According to the Centers for Disease Control and Prevention (CDC) Trusted Source, breast cancer is the most common cancer in women. Breast cancer survival rates vary widely supported by many factors. Two of the most important factors are the type of cancer women have and the stage of cancer at the time they receive a diagnosis. Breast cancer is cancer that develops in breast cells. Typically, the cancer forms in either the lobules or the ducts of the breast. Cancer also can occur within the adipose tissue or the fibrous connective tissue within your breast. The uncontrolled cancer cells often invade other healthy breast tissue and may visit the lymph nodes under the arms.

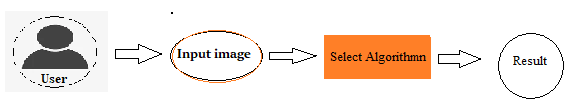
Doctors say that breast cancer happened due to abnormal growth of cells in the breast and these cells spread in size like Meta Size from breast to lymph nodes or the other parts of the body also. Hence it is necessary to detect and stop the growth of these unwanted cells as early as possible to avoid the next phase consequences. If a tumor is diagnosed then the first step taken by the doctor is, they check whether the tumor is Benign or Malignant. Because the treatment and prevention methods of both the tumors are different. Benign cells are neither cancerous and nor spread but Malignant cells are cancerous and can spread to other parts of bodies. The problem with this disease is, there is no such proper diagnostic machine is present to detect cancer in the early phase so the person can start the treatment as early as possible and try to stop the growth of unwanted cells or tumors.

Early diagnosis of any disease is often curable with a touch amount of human effort. Most people fail to detect their disease before it becomes chronic. It leads to an increase in the death rate around the world. Breast cancer is one of the diseases that could be cured when the disease is identified at earlier stages before it is spreading across all the parts of the body.

The lack of prognosis models results in difficulty for doctors to prepare a treatment plan that may prolong patient survival time. Hence, time requires developing the technique which gives minimum error to increase accuracy. The available tests to detect breast cancer such as mammogram, ultrasound, and biopsy were time- consuming, so there was a need for a computerized diagnostic system in which Machine Learning methodology was used. This methodology includes algorithms that help for the classification of the tumor and detect the cells more accurately and take less time a well

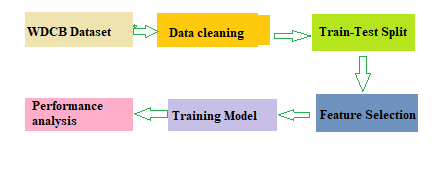
**2.2 DETAILED DIAGRAM**

**2.2.1Front End Module Diagrams:**



**2.2.2 Back End Module Diagrams:**

In back end we have the dataset in .csv format. First load the data using pandas library. There are rows and columns or features. There are some null values in the data. Detect the null values and replace with mean if the feature are numeric feature otherwise most frequently word replace. Here I have used Simple Imputer to replace the all null values. This is binary class classification and imbalanced dataset. I convert the text data into numeric data.



**2.3 SYSTEM SPECIFICATION:**

**2.3.1 HARDWARE REQUIREMENTS:**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented

PROCESSOR : Intel I5

RAM : 4GB

HARD DISK : 40 GB

**2.3.2 SOFTWARE REQUIREMENTS:**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

PYTHON IDE **:** Anaconda Jupyter Notebook

PROGRAMMING LANGUAGE : Python

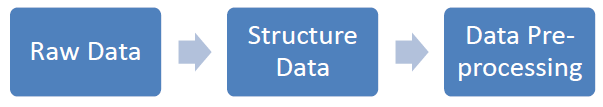
**2.4 MODULE DESCRIPTION**

**DATA COLLECTION:**

The first phase we do is to collect the data that we are interested in collecting for pre-processing and to apply classification and Regression methods. Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real world data is often incomplete, inconsistent, and lacking certain to contain many errors. Data pre-processing is a proven method of resolving such issues. Data pre-processing prepares raw data for further processing. For pre-processing we have used standardization method to pre-process the UCI dataset. This step is very important because the quality and quantity of data that you gather will directly determine how good your predictive model can be. In this case we collect the Breast Cancer samples which are Benign and Malignant. This will be our training data.

**PRE-PROCESSING OF DATA :**

Data preprocessing is a process in which that is actual use for converting the basic data into the clean data set. It is the step in which the data transform or an encode to the state that the machine can be easily parse. The major task of data preprocessing in learning process is to remove the unwanted data and filling the missed value. So that it help to machine can be trained easily.



**DATA CLEANING:**

In this module the data is cleaned. After cleaning of the data, the data is grouped as per requirement. This grouping of data is known as data clustering. Then check if there is any missing value in the data set or not. It there is some missing value then change it by any default value. After that if any data need to change its format, it is done. That total process before the prediction is known is data pre-processing. After that the data is used for the prediction and forecasting step.

**DATA PREPROCESSING**

In this step, the pre-processed data is taken for the prediction. This prediction can be done in any process which are mentioned above. For that, the pre-processed data is splitted for the train and test purpose. Then a predictive object is created to predict the test value which is trained by the trained value. Then the object is used to forecast data for next few years.

**DATA SPLITTING:**

For each experiment, we split the entire dataset into 70% training set and 30% test set. We

used the training set for resampling, hyper parameter tuning, and training the model and we used test set to test the performance of the trained model. While splitting the data, we specified a random seed (any random number), which ensured the same data split every time the program executed.

**TRAIN DATASET AND TEST DATASET**

The training data is a initial set of data which is used to understand the program. This is the one in which we have to train the model first because to set the feature and this data is available on system. This data is used to teach the machine for do different actions. It is the data in which model can learn with algorithm to teach the model and doing work automatic.

Testing data is the input given to a software. It shows the data affects when the execution of the module that specifying and this is basically used for testing.

**MODEL SELECTION:**

Supervised learning is the method in which the machine is trained on the data which the input and output are well labelled. The model can learn on the training data and can process the future data to predict outcome. They are grouped to Regression and Classification techniques. A regression problem is when the result is a real or continuous value, such as “salary” or “weight”. A classification problem is when the result is a category like filtering emails spam” or “not spam”. Unsupervised Learning: Unsupervised learning is giving away information to the machine that is neither classified nor labelled and allowing the algorithm to analyse the given information without providing any directions. In unsupervised learning algorithm the machine is trained from the data which is not labelled or classified making the algorithm to work without proper instructions. In our dataset we have the outcome variable or Dependent variable i.e. Y having only two set of values, either M (Malign) or B (Benign). So Classification algorithm of supervised learning is applied on it. We have chosen three different types of classification algorithms in Machine Learning. We can use a small linear model, which is a simple.

**RANDOM FOREST:**

Random forests also known as random decision forests creates a large number of trees that achieve their output through ensemble learning methods for classification, regression. Bagging and feature randomness are the features it uses to construct those trees. The random forest has an advantage over the decision tree which, is that it does not overfit the data.

**DECISION TREE:**

DTs apply a top-down approach to data so that given a knowledge set, they struggle to group and label observations that are similar between them, and appearance for the simplest rules that split the observations that are not the same between them until they reach a certain degree of similarity. They use a layered splitting process, where at each layer they struggle to separate the info into two or more groups, in order that data that fall under an equivalent group is most similar to every other (homogeneity), and groups are as die-rent as possible from one another.

**PERFORMANCE MATRICES:**

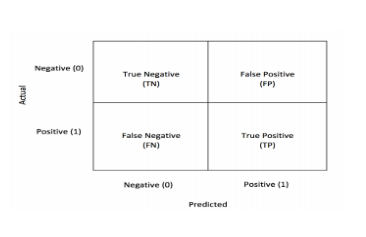
Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these six algorithms were applied on same dataset using Enthought Canaopy and results were obtained.



Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm.

**CONFUSION MATRIX:**

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.



All predicted true positive and true negative divided by all positive and negative. True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) predicted by all algorithms are presented in table.

True positive (TP) indicates that the positive class is predicted as a positive class, and the number of sample positive classes was actually predicted by the model.

False negative indicates (FN) that the positive class is predicted as a negative class, and the number of negative classes in the sample was actually predicted by the model.

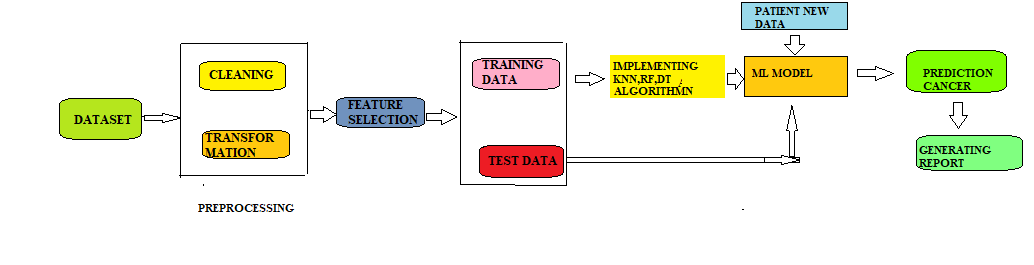
False positive (FP) indicates that the negative class is predicted as a positive class, and the number of positive classes of samples was actually predicted by the model.

True negative (TN) indicates that the negative class is predicted as a negative class, and the number of sample negative classes was actually predicted by the model.

**2.5 MODULE DIAGRAM**

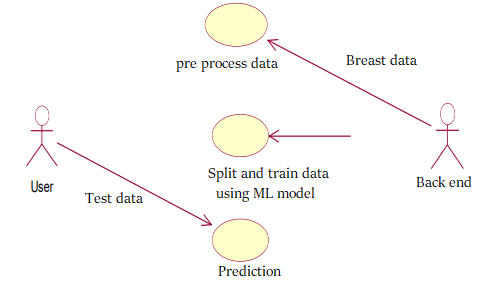
Designing of system is the process in which it is used to define the interface, modules and data for a system to specified the demand to satisfy. System design is seen as the application of the system theory. The main thing of the design a system is to develop the system architecture by giving the data and information that is necessary for the implementation of a system.

**2.5.1 SYSTEM ARCHITECTURE**

****

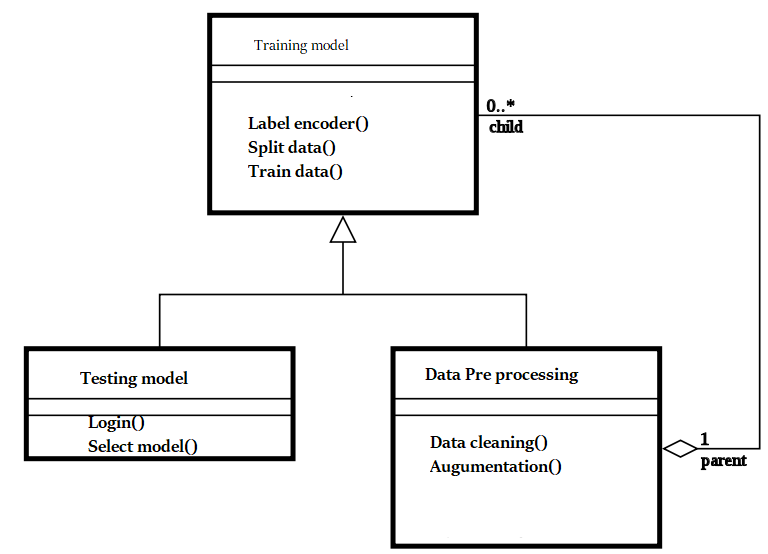
**2.5.2 Use case Diagram:**

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system

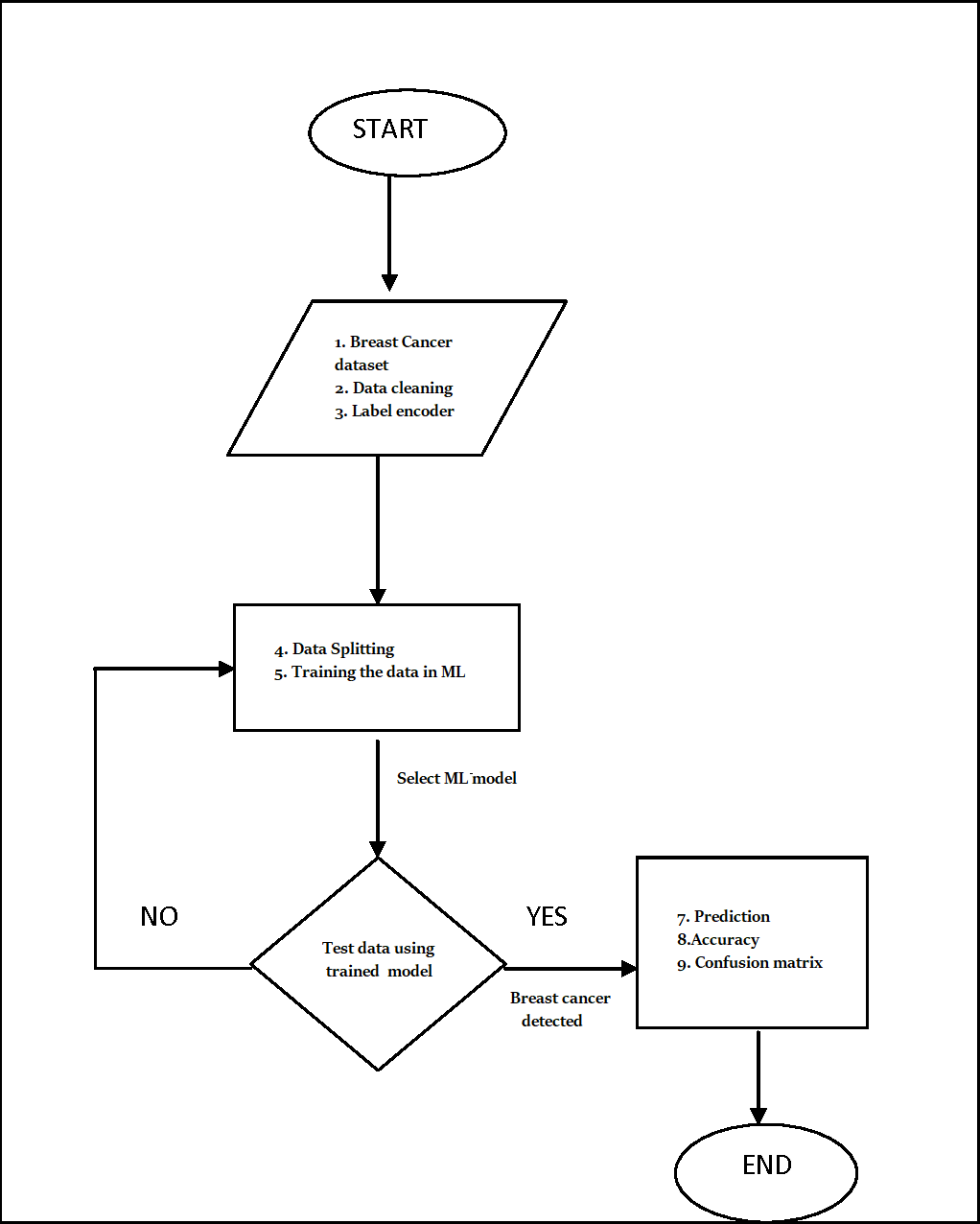
****

**2.5.3 CLASS DIAGRAM**

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application

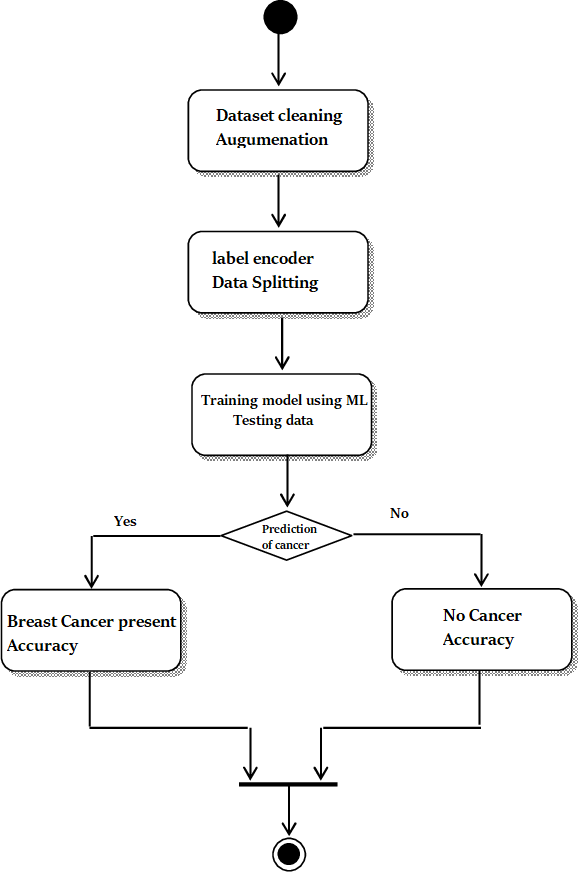
****

**DATA FLOW DIAGRAM:**



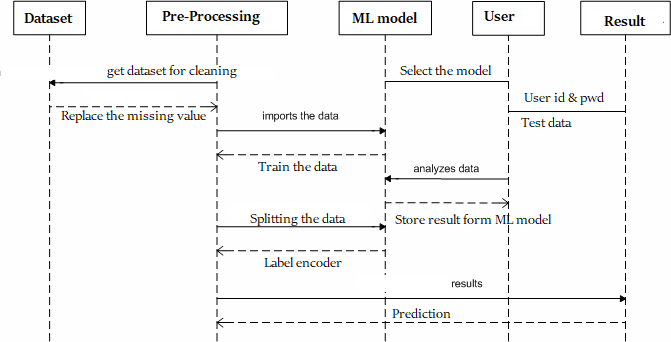
**2.5.4Activity of Diagram:**

The Activity Diagram forms effective while modeling the functionality of the system. Hence this diagram reflects the activities, the types of flows between these activities and finally the response of objects to these activities

****

**2.5.6 SEQUENCE Diagram:**

The sequence diagram of a system shows the entity interplay are ordered in the time order level. So, that it drafts the classes and object that are imply in the that plot and also the series of message exchange take place betwixt the body that need to be carried out by the purpose of that scenario.

****

**CHAPTER 3**

**SOFTWARE SPECIFICATION**

**3.1 general**

**3.2 ANACONDA**

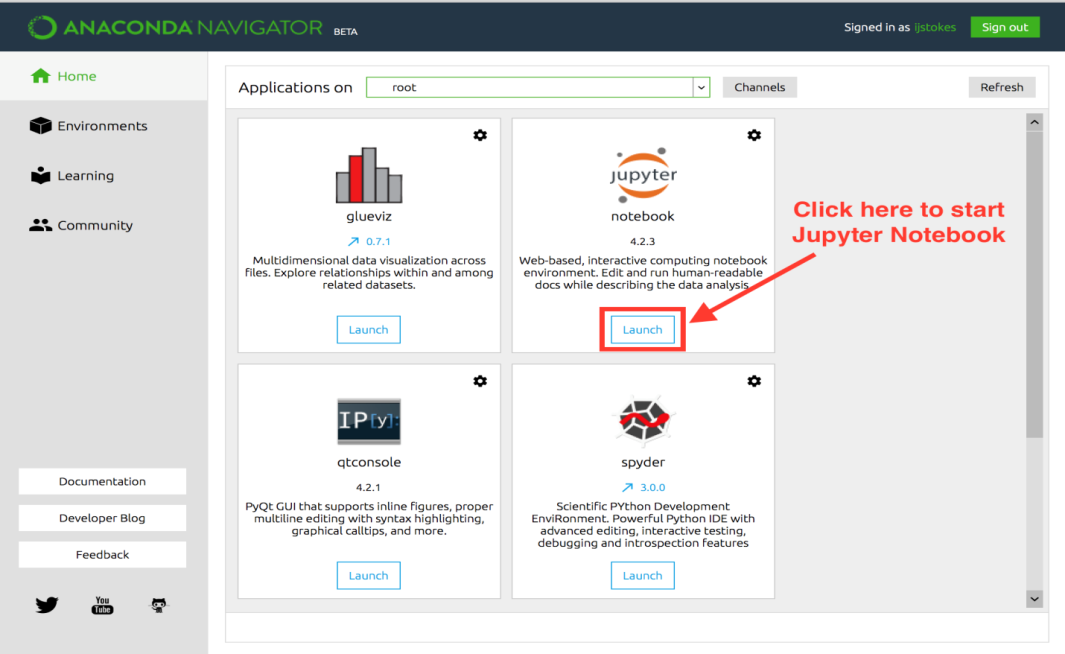
It is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Anaconda distribution comes with more than 1,500 packages as well as the [Conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the Command Line Interface (CLI).

The big difference between Conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason Conda exists. Pip installs all Python package dependencies required, whether or not those conflict with other packages you installed previously.

So your working installation of, for example, Google Tensorflow, can suddenly stop working when you pip install a different package that needs a different version of the Numpy library. More insidiously, everything might still appear to work but now you get different results from your data science, or you are unable to reproduce the same results elsewhere because you didn't pip install in the same order.

Conda analyzes your current environment, everything you have installed, any version limitations you specify (e.g. you only want tensorflow>= 2.0) and figures out how to install compatible dependencies. Or it will tell you that what you want can't be done. Pip, by contrast, will just install the thing you wanted and any dependencies, even if that breaks other things.Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the conda install command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. You can also install anything on PyPI into a Conda environment using pip, and Conda knows what it has installed and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories.The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.



Anaconda Navigator is a desktop [Graphical User Interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

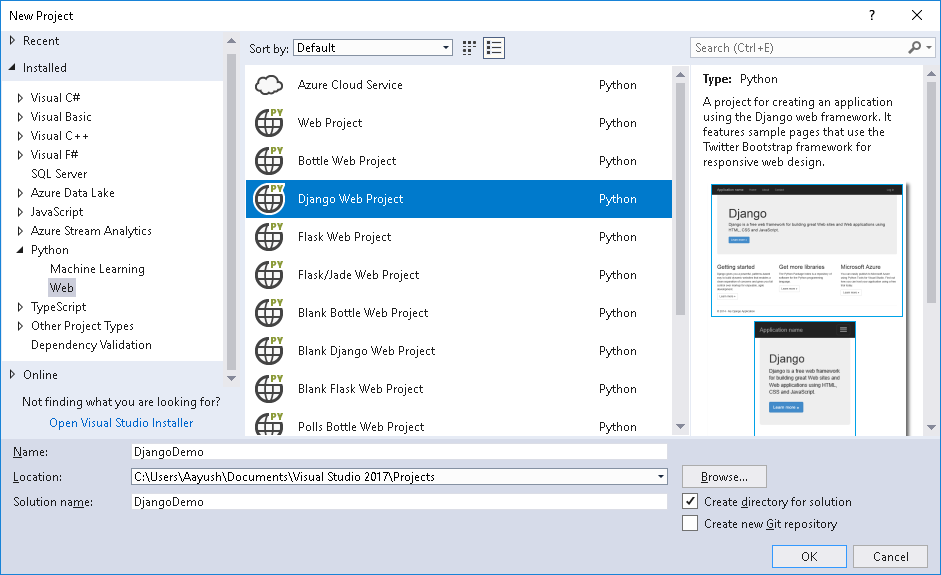
The following applications are available by default in Navigator:

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Lab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* [QtConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glueviz](http://glueviz.org/)
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [Rstudio](https://en.wikipedia.org/wiki/Rstudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

Microsoft VISUAL STUDIO  is an Integrated Development Environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.



Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Python features are:

* Easy to code
* Free and Open Source
* Object-Oriented Language
* GUI Programming Support
* High-Level Language
* Extensible feature
* Python is Portable language
* Python is Integrated language
* Interpreted
* Large Standard Library
* Dynamically Typed Language

**3.3 PYTHON:**

* Python is a powerful multi-purpose programming language created by Guido van Rossum.
* It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Features Of Python :

1.Easy to code:  
Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2. Free and Open Source:  
Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword.  
Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3.Object-Oriented Language:  
One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4. GUI Programming Support:  
Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.  
PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:  
Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6.Extensible feature:  
Python is a Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language:  
Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language:  
Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language:  
Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library  
Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language:  
Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance.because of this feature we don’t need to specify the type of variable.

APPLICATIONS OF PYTHON :

WEB APPLICATIONS

* You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are:Django, Flask, Pyramid, Plone, Django CMS.
* Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

3.3.1 **SCIENTIFIC AND NUMERIC COMPUTING**

* There are numerous libraries available in Python for scientific and numeric computing. There are libraries like:SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.
* Also, the language is heavily used in machine learning, data mining and deep learning.

**3.3.2 CREATING SOFTWARE PROTOTYPES**

* Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.
* However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

**3.3.3 GOOD LANGUAGE TO TEACH PROGRAMMING**

* Python is used by many companies to teach programming to kids
* It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use system.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 GENERAL**

Python is a program that was originally designed to simplify the implementation of numerical linear algebra routines. It has since grown into something much bigger, and it is used to implement numerical algorithms for a wide range of applications. The basic language used is very similar to standard linear algebra notation, but there are a few extensions that will likely cause you some problems at first.

**4.2 CODE IMPLEMENTATION**

**Back End Code:-**

**#list of useful imports that I will use**

%matplotlib inline

import os

import matplotlib.pyplot as plt

import pandas as pd

import cv2

import numpy as np

from glob import glob

import seaborn as sns

import random

import pickle

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import roc\_curve

data.describe()

data.shape

data.isnull().sum()

data['diagnosis'].value\_counts()

from sklearn.utils import resample

**# Separate majority and minority classes**

df\_majority = data[data['diagnosis']== 'B']

df\_minority = data[data['diagnosis']== 'M']

**# Downsample majority class and upsample the minority class**

df\_minority\_upsampled=resample(df\_minority,replace=True,n\_samples=1000,random\_state=123)

df\_majority\_downsampled=resample(df\_majority,replace=True,n\_samples=1000,random\_state=123)

**# Combine minority class with downsampled majority class**

df\_upsampled=pd.concat([df\_minority\_upsampled,df\_majority\_downsampled])

**# Display new class counts**

df\_upsampled['diagnosis'].value\_counts()

from sklearn.model\_selection import train\_test\_split

### Label encoder

from sklearn.preprocessing import LabelBinarizer

enc = LabelBinarizer()

y = enc.fit\_transform(data['Class\_label'])

y[:10]

### Spliting data into training and testing

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(images, y, test\_size=0.3,stratify = y,random\_state=42)

**DECISION TREE:**

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score,confusion\_matrix

from sklearn.model\_selection import GridSearchCV

dept = [1, 5, 10, 50, 100, 500,800, 1000]

min\_samples = [5, 10, 100, 500]

param\_grid={'min\_samples\_split':min\_samples , 'max\_depth':dept}

model = GridSearchCV(clf,param\_grid,scoring='accuracy',n\_jobs=-1,cv=3)

model.fit(x\_train,y\_train)

print("optimal min\_samples\_split",model.best\_estimator\_.min\_samples\_split)

print("optimal max\_depth",model.best\_estimator\_.max\_depth)

**#print the test accuracy**

score = model.evaluate(X\_test, y\_test, verbose=0)

print('Test accuracy:', score[1])

**RANDOM FOREST CLASSIFIER**

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score,confusion\_matrix

from sklearn.model\_selection import GridSearchCV

param\_grid={'n\_estimators':n\_estimators , 'max\_depth':dept}

clf = RandomForestClassifier()

model = GridSearchCV(clf,param\_grid,scoring='accuracy',n\_jobs=-1,cv=3)

model.fit(x\_train,y\_train)

print("optimal n\_estimators",model.best\_estimator\_.n\_estimators)

print("optimal max\_depth",model.best\_estimator\_.max\_depth)

optimal\_n\_estimators = model.best\_estimator\_.n\_estimators

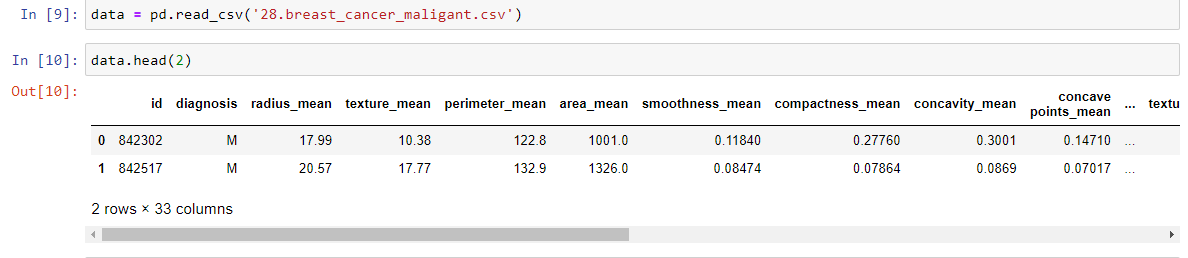
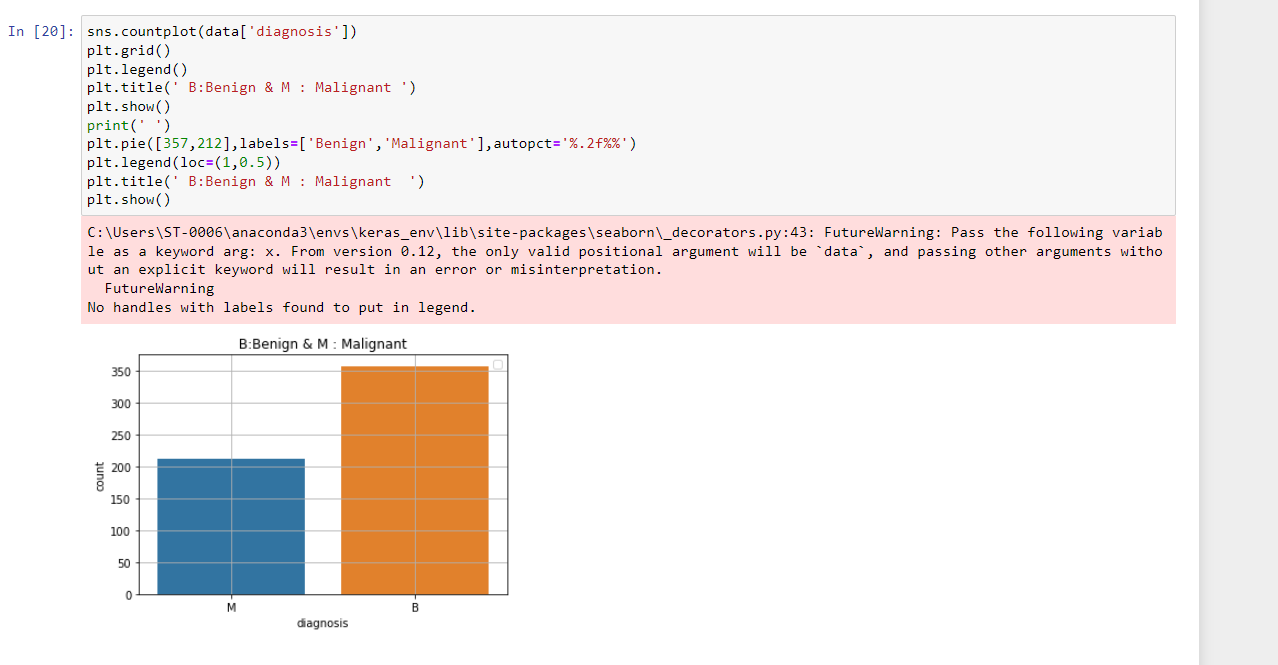
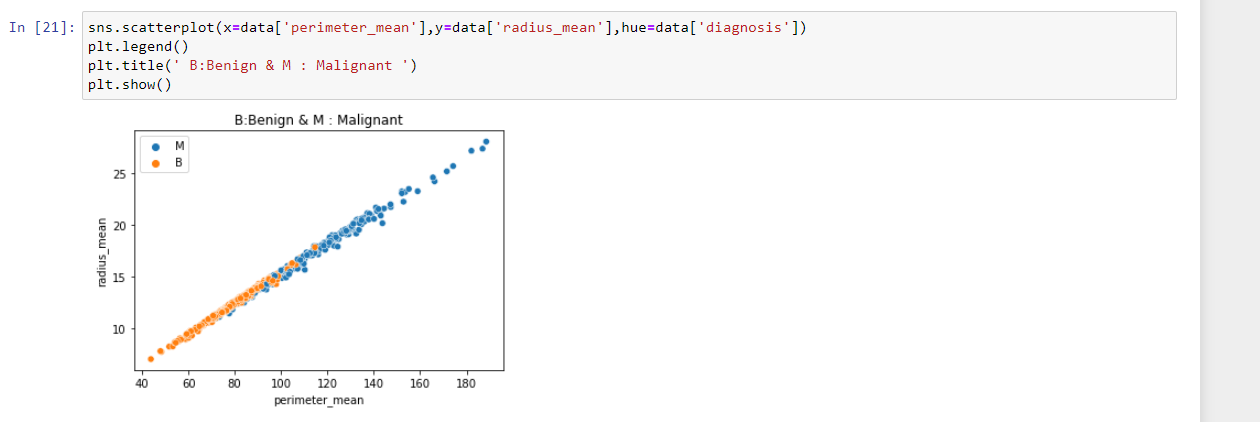
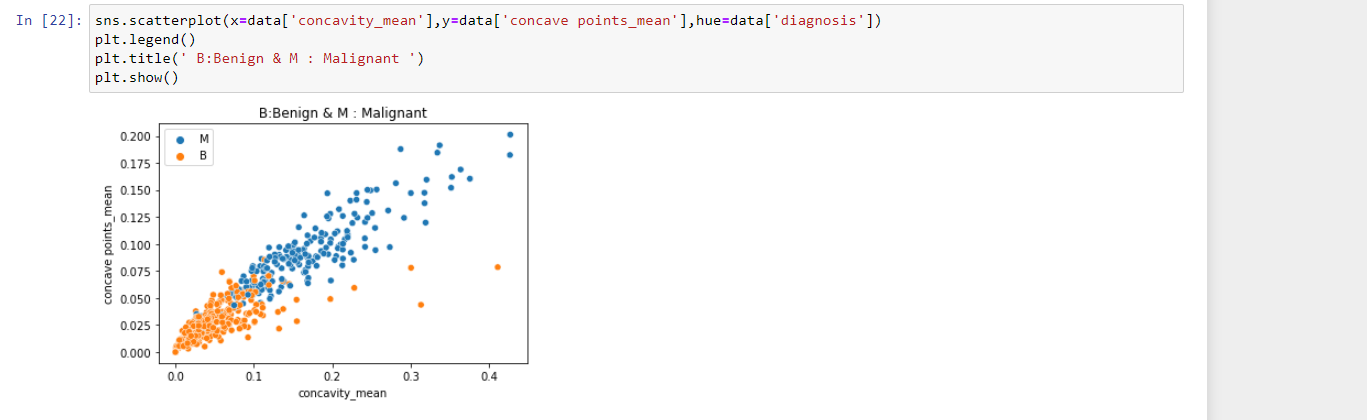
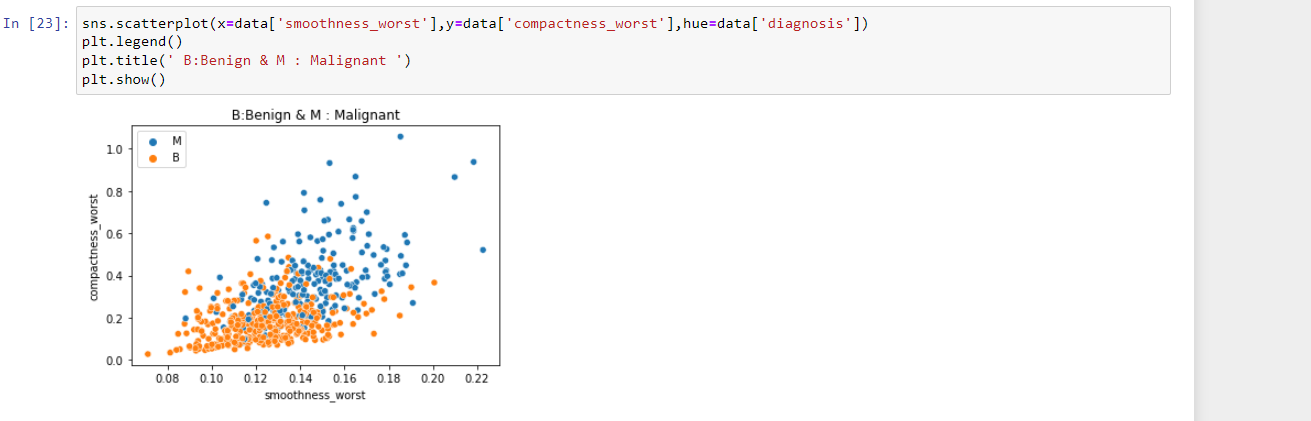
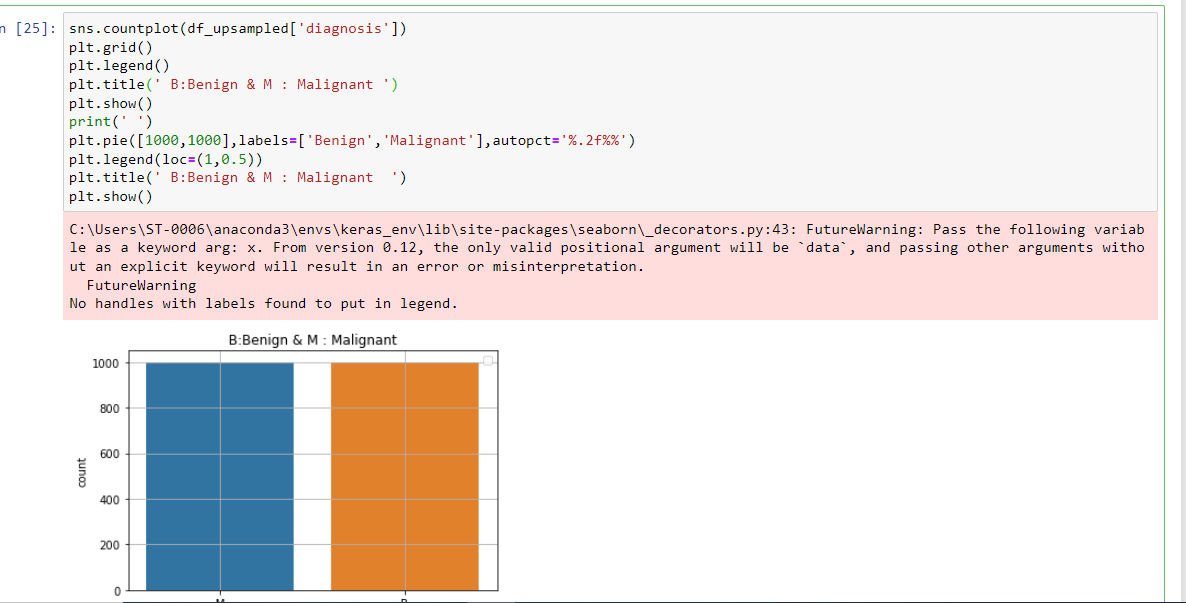
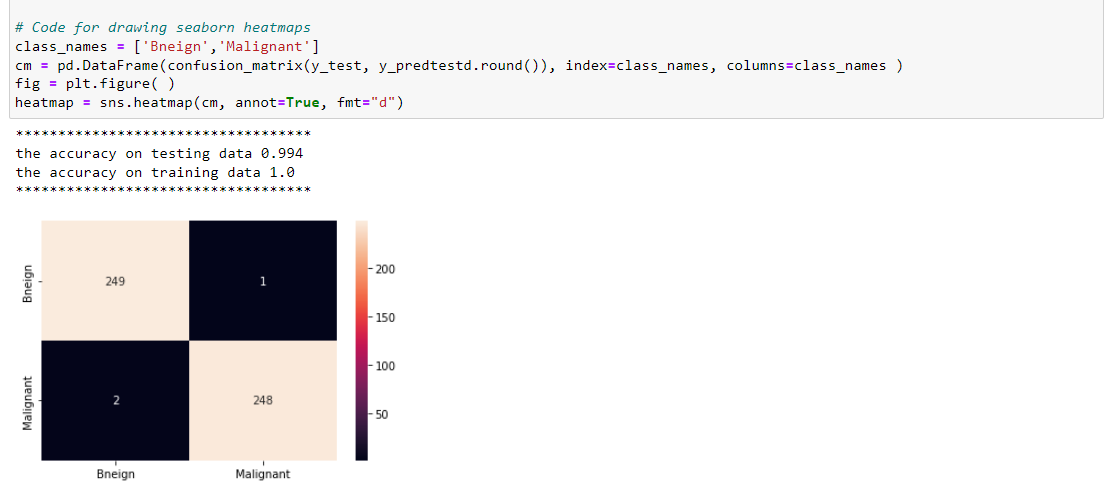
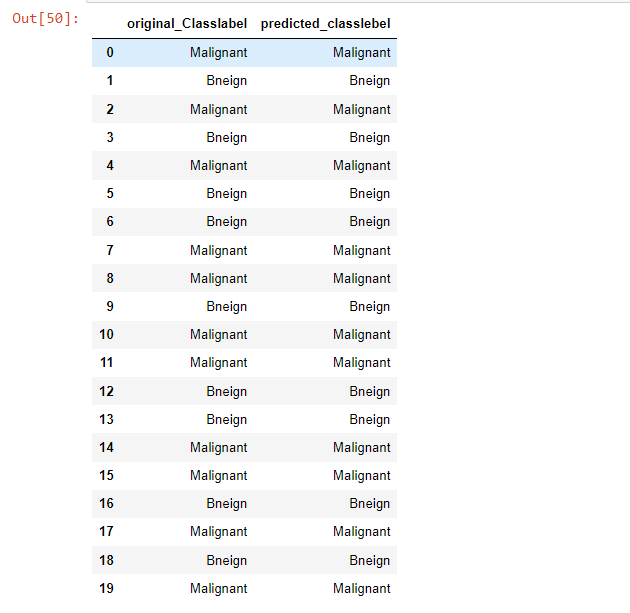
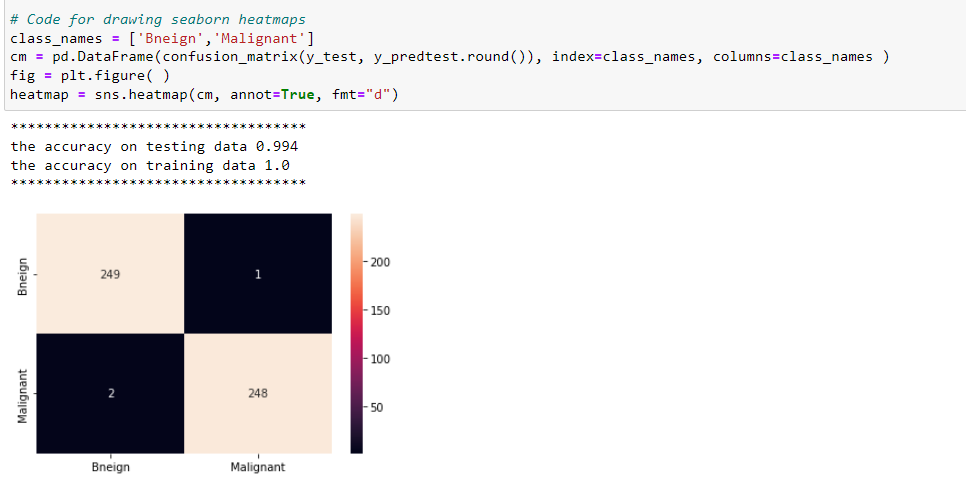
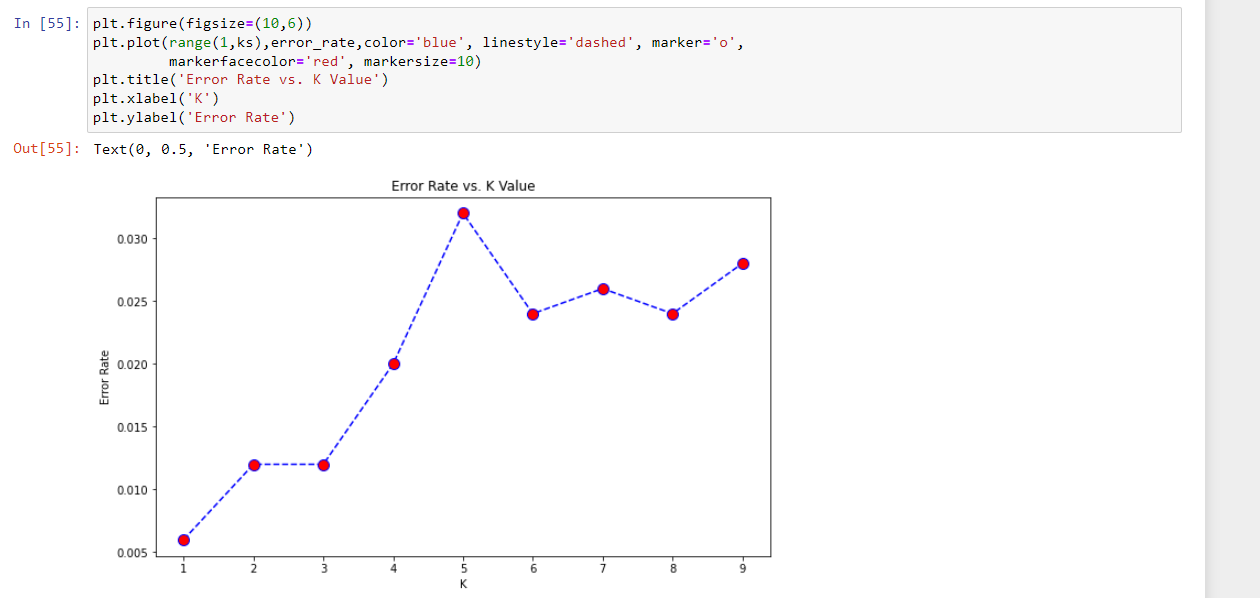
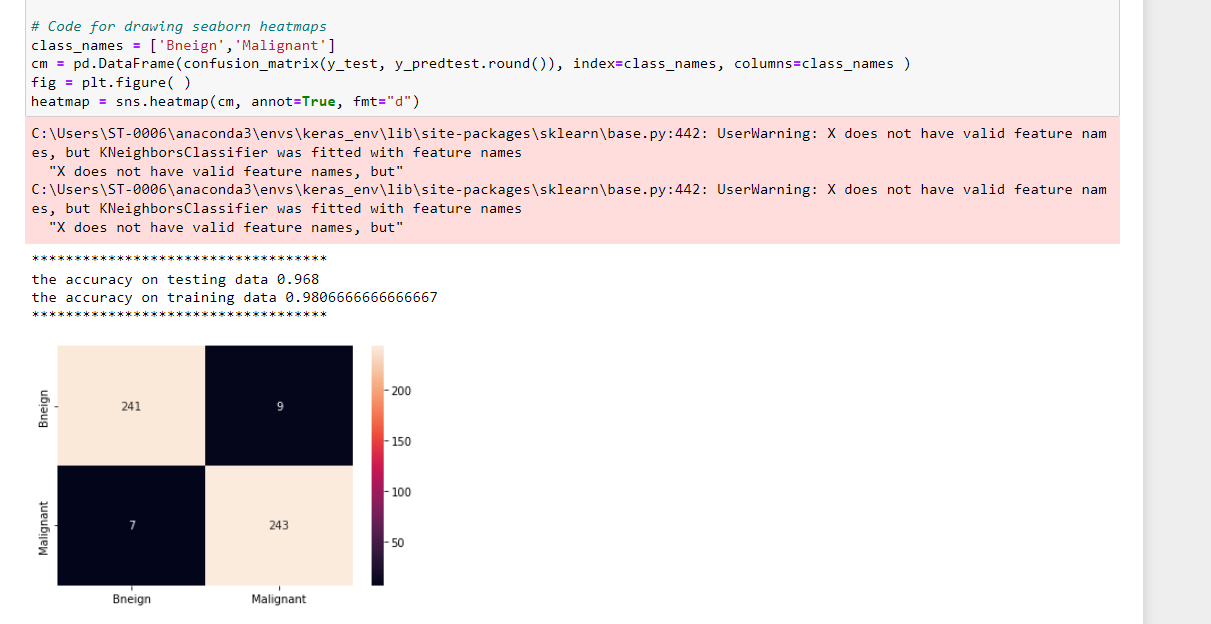
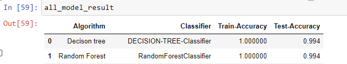
optimal\_max\_depth = model.best\_estimator\_.max\_depth

**#print the test accuracy**

score = model.evaluate(X\_test, y\_test, verbose=0)

print('Test accuracy:', score[1])

**4.3 SNAPSHOTS**

**           ** 

**CHAPTER 5**

**5.1 CONCLUSION AND REFERENCES**

Now-a-days, one of the deadly diseases affecting women is breast cancer. In our work, the Wisconsin Breast Cancer Dataset was utilized and several ML algorithms were applied to assimilate the efficacy and usefulness of these algorithms to find the highest accuracy of classifying malignant and benign breast cancer. The correlation between different features of the dataset has been analyzed for feature selection. The results will assist to pick the best ML algorithm for the construction of an automatic breast cancer diagnostic system. From our study, we can conclude that SVM and Random Forest give the maximum accuracy with an accuracy .

**5.2APPLICATION**

* It used hospital sectors
* It use identify early age detection either malignant and benign breast cancer

**5.3REFERENCES:**

[1] J. Sivapriya, A. Kumar, S. Siddarth Sai, and S. Sriram, “Breast cancer prediction using machine learning,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, 2019.

[2] Y. Khourdifi and M. Bahaj, “Applying best machine learning algorithms for breast cancer prediction and classification,” in 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS). IEEE, 2018, pp. 1-5.

[3] N. K. Sinha, M. Khulal, M. Gurung, and A. Lal, “Developing a web based system for breast cancer prediction using xgboost classifier,” International Journal of Engineering Research Technology (IJERT), vol. 9, 2020.

[4] R. Dhanya, I. R. Paul, S. S. Akula, M. Sivakumar, and J. J. Nair, “A comparative study for breast cancer prediction using machine learning and feature selection,” in 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, 2019, pp. 1049-1055.

[5] M. M. Islam, H. Iqbal, M. R. Haque, and M. K. Hasan, “Prediction of breast cancer using support vector machine and k-nearest neighbors,”in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC).* IEEE, 2017, pp. 226-229.

[6] M. S. Yarabarla, L. K. Ravi, and A. Sivasangari, “Breast cancer prediction via machine learning,” in *2019 3rd International Conference* *on Trends in Electronics and Informatics (ICOEI).* IEEE, 2019, pp. 121-124.

[7] V. Chaurasia, S. Pal, and B. Tiwari, “Prediction of benign and malignant breast cancer using data mining techniques,” *Journal of Algorithms &* *Computational Technology*, vol. 12, no. 2, pp. 119-126, 2018.

[8] N. Fatima, L. Liu, S. Hong, and H. Ahmed, “Prediction of breast cancer, comparative review of machine learning techniques, and their analysis,” *IEEE Access,* vol. 8, pp. 150360-150376, 2020.

[9] A. Toprak, “Extreme learning machine (elm)-based classification of benign and malignant cells in breast cancer,” *Medical science monitor:* *international medical journal of experimental and clinical research*, vol. 24, p. 6537, 2018.

[10] D. S. Jacob, R. Viswan, V. Manju, L. PadmaSuresh, and S. Raj, “A survey on breast cancer prediction using data miningtechniques,” in *2018* *Conference on Emerging Devices and Smart Systems (ICEDSS).* IEEE, 2018, pp. 256-258.

[11] K. L. Kashyap, M. K. Bajpai, and P. Khanna, “Breast cancer detection in digital mammograms,” in *2015 IEEE international conference on* *imaging systems and techniques (IST).* IEEE, 2015, pp. 1-6.