C15_Batch__original.docx

by

Submission date: 10-May-2022 12:13PM (UTC+0530)
Submission ID: 1832785254
File name: C15_Batch__original.docx (4.74M)
Word count: 5863
Character count: 42741

A Mini Project with Seminar On

Breast Cancer Prediction

Submitted in partial fulfillment of the requirements for the award of the

Bachelor of Technology

In

Department of Computer Science and Engineering

By

M Viswa Sowrabh Reddy	**19241A05E9**
Kudikyala Nikhil	**19241A05E7**
Nagendram Vinod	**19241A05F2**
Kummari Naveen	**19241A05E8**

Under the Esteemed guidance of

Dr S Govinda Rao

Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Autonomous under JNTUH, Hyderabad, Bachupally, Kukatpally, Hyderabad-500090

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Autonomous under JNTUH, Hyderabad, Bachupally, Kukatpally, Hyderabad-500090)

CERTIFICATE

This is to certify that the mini project entitled "**Breast Cancer Prediction**" is submitted by **M Viswa Sowrabh Reddy(19241A05E9), Kudikyala Nikhil(19241A05E7), Nagendram Vinod(19241A05F2), Kummari Naveen(19241A05E8)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2021-2022.

INTERNAL GUIDE	HEAD OF THE DEPARTMENT
Dr S Govinda Rao	**Dr. K. MADHAVI**
Assistant Professor	Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The help provided by many people has resulted in the completion of this project successfully. We would like to take this opportunity to express my thanks to all who has helped us to complete the project. First, we would like to express our deep gratitude towards our internal guide **Dr S Govinda Rao, Professor,** Department of CSE for her support in the completion of our dissertation. We extend our gratitude to our concerned lab faculty **Ms. Rubeena.** We wish to express our sincere thanks to **Dr. K. Madhavi, HOD, Department of CSE** and to our principal **Dr. J. Praveen** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

M Viswa Sowrabh Reddy(19241A05E9)

Kudikyala Nikhil(19241A05E7)

Nagendram Vinod(19241A05F2)

KummariNaveen(19241A05E9)

DECLARATION

We hereby declare that the industrial major project entitled **"Breast Cancer Prediction "** is the work done during the period from **9th February 2022 to 28th April 2022** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad).The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

M Viswa Sowrabh Reddy(19241A05E9)

Kudikyala Nikhil(19241A05E7)

Nagendram Vinod(19241A05F2)

KummariNaveen(19241A05E8)

ABSTRACT

Breast cancer is the most common disease among women, affecting 2.1 million women each year, and it also causes the most cancer-related deaths among women, according to the World Health Organization (WHO). Breast cancer claimed the lives of 627,000 women in 2018, accounting for nearly 15% of all cancer deaths among women. While breast cancer rates are greater among women in more developed countries, they are rising in almost every location around the world. Early detection is crucial for improving breast cancer outcomes and survival. Breast cancer has two early detection strategies: early diagnosis and screening.. Early detection initiatives based on knowledge of early signs and symptoms and timely referral to diagnosis and treatment should be prioritised in low-resource areas with weak health systems where the majority of women are identified late. Early diagnosis strategies aim to reduce barriers to care and/or improve access to effective diagnosis services in order to provide prompt cancer treatment. The goal is to raise the percentage of breast cancers that are detected early, allowing for more effective treatment and lowering the risk of death from breast cancer. We employ several machine learning algorithms to predict whether a tumour is benign or malignant depending on the information presented, because early detection of cancer is critical for effective treatment of breast cancer.

Breast cancer is one of the diseases that kills a lot of people every year all over the world, and early identification and diagnosis of the disease is tough to achieve in order to reduce the number of deaths. Various machine learning and data mining techniques are currently being used in medical diagnostics, and they have proven to be efficient in predicting chronic diseases such as cancer, which can save the lives of individuals who are suffering from such diseases. The fundamental purpose of this study is to determine the accuracy of classification algorithms like Support Vector Machine, J48, Nave Bayes, and Random Forest so that the optimal strategy can be recommended.

Chapter-1

INTRODUCTION

According to global statistics, breast cancer (BC) is one of the most frequent malignancies among women globally, accounting for the majority of new cancer cases and cancer-related deaths, making it a significant public health burden in today's society. Early detection of BC improves the prognosis and chances of survival by allowing patients to receive timely clinical treatment. More precise classification of benign tumors can help patients avoid needless therapies. As a result, accurate BC diagnosis and classification of individuals into malignant or benign groups are the focus of extensive research. Machine learning (ML) is widely regarded as the approach of choice in BC pattern classification and forecast modelling due to its unique benefits in detecting essential characteristics from complex BC datasets.

Some Breast Cancer Risk Factors

The following are some of the recognized breast cancer risk factors. Most incidences of breast cancer, however, cannot be attributed to a specific cause. Consult your doctor about your particular risk.

Age. As women become older, their chances of developing breast cancer increase. Women over the age of 50 account for about 80% of breast cancer cases.

Personal experience with breast cancer. A woman who has had breast cancer in one breast is more likely to have cancer in the other breast.

Breast cancer runs in the family. If a woman's mother, sister, or daughter had breast cancer when she was young, she has an increased risk of breast cancer (before 40).

Genetic influences. Women with particular genetic abnormalities, such as alterations in the BRCA1 and BRCA2 genes, have a greater lifetime chance of developing breast cancer. Other gene variations may also increase the risk of breast cancer.

Menstrual history and childbearing. The higher a woman's risk of breast cancer is when she has her first kid, the older she is.

1.Women who menstruate for the first time at a young age are also at a higher risk (before 12)
2. Women who have a late menopause (after age 55)
3. Women who have never given birth

Machine Learning's Role in Breast Cancer Detection

An x-ray image of the breast is called a mammogram. It can be used to detect breast cancer in women who have no symptoms or indicators of the disease. If you have a lump or other sign of breast cancer, it can also be used. A screening mammogram is one that analyses your breasts while you have no symptoms. It has the potential to minimize the incidence of breast cancer deaths among women aged 40 to 70. It can, however, have disadvantages.

Mammograms can occasionally detect something odd that isn't cancer. This leads to more testing, which can be stressful. Mammograms can sometimes overlook cancer when it is present. You are also exposed to radiation. You should discuss the advantages and disadvantages of mammography with your doctor. You and your doctor can determine when to begin and how often to get mammograms.

While it is difficult for physicians to determine whether a tumor is harmful or not based on x-ray pictures alone, creating a machine learning model based on tumor identification can be extremely beneficial.

1.1 Rationale :

Breast cancer prediction is necessary because one out of every eight people dies from it. Early identification of cancer lowers the risk of mortality and makes treatment less expensive. The stage of cancer for treatment will be determined by classifying whether sort of tumour (malignant or benign) will form in the breast. Every year, 2.7 million people are diagnosed with this illness; this classification can aid in determining when and at what age people are most likely to develop this disease.

1.2Goal :

The purpose of this little project is to use the highest accuracy technique (Logistic Regression, Decision Tree, Random Forest Classification) to predict the type of tumour that has occurred in a woman's breast (whether it is benign or malignant).

1.3Methodology :

1.3.1LOGISTIC REGRESSION :

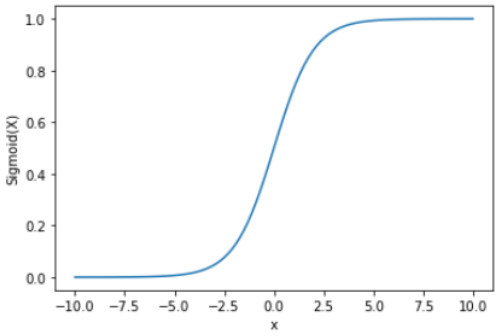
Logistic regression is another powerful supervised machine learning algorithm used for binary classification problems (when the goal is categorical). The best way to think about logistic regression is linear regression, but for classification problems. Logistic regression essentially models binary output variables using the logistic function defined below (Tolles & Meurer, 2016). The main difference between linear and logistic regression is that logistic regression is limited to the range from 0 to 1. Also, unlike linear regression, logistic regression does not require a linear relationship between the input and output variables. This is due to the application of a non-linear logarithmic transformation to the odds ratio (to be determined soon).

Below is the function for Logistic Regression :

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}}$$

>E is log base.

>X is the numerical value that needs to be transformed.



1.3.2DECISION TREE ALGORITHM :

DT is a predictive model expressed as a recursive partition of a feature space into subspaces that form the basis of prediction (Rokach, 2016). DT is a root-directed tree.
A node with an outgoing edge from a DT is an inner node. All other nodes are leaf nodes or leaves of the DT. DTs are classified using a set of hierarchical feature decisions. Decisions made by internal nodes are the basis for separation. In DT, each leaf belongs to one class or its probability.
Small changes in the training set result in different splits, resulting in different DTs.
Therefore, the contribution of error due to variance is large for DT. Ensemble learning, discussed in the next section, can help mitigate errors due to variance.

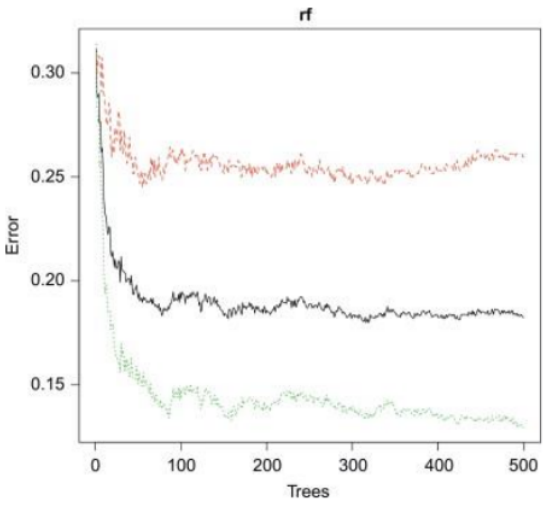
There are several things that enhance the overall benefits of using decision trees. If you have data sets of very different attributes, such as income recorded in millions and loan maturity recorded in years, many algorithms require scale normalization before building and applying the model.
These variable transformations are not required for decision trees because the tree structure remains the same with or without transformations.

1.3.3RANDOM FOREST CLASSIFICATION :

Random forest (RF), developed by Breiman (2001), is an ensemble classification scheme that utilizes a majority vote to predict classes based on the partition of data from multiple decision trees. RF grows multiple trees by randomly subsetting a predefined number of variables to split at each node of the decision trees and by bagging. Bagging generates training data for each tree by sampling with replacement a number of samples equal to the number of samples in the source dataset (Breiman, 1996). RF implements the Gini index to determine a best split threshold of input values for given classes. The Gini index returns a measure of class heterogeneity within child nodes as compared to the parent node (Breiman, 2017; Waske et al., 2009). RF requires the selection of an mtry parameter that sets the number of possible variables that can be randomly chosen to split at each tree node in the forest.

The randomForest package (Liaw and Wiener, 2002) is suitable for multiple trees (hence the name forest). To avoid overfitting, a random forest loads the data, selects covariates, and fits a decision tree. In this way, many trees are created and individual observations and individual covariates have less impact on the overall prediction. Each tree gets a "vote" which is a prediction. In binary it is either 0 or 1. You can average these predictions to extract P(Y = 1) or classify the results by the category with the most votes.

The figure below shows the training data misclassification rates as a whole (black) by units (red, gray for the printed version) and zeros (green, light gray for the printed version). It can be seen that the positive result has the worst classification and the overall misclassification rate is about 18%.



1.4ExistingSystems :

Many studies on breast cancer have been published in the literature of medical data analysis, and the majority of them show good classification accuracy.

1.4.1Polat et al introduced the LS-SVM classifier method for the diagnosis of breast cancer and used 10-fold cross validation to reach a classification accuracy of 88.53 percent.

1.4.2Akay proposed a new method for breast cancer diagnosis that uses a support vector classification algorithm on the best predictive features to achieve 89.02 percent classification accuracy without cross-validation.

1.4.3By combining statistical approaches with swarm optimization, Yeh et al. provide a unique technique for breast cancer diagnosis with an accuracy of 88.71 percent.

In their paper, Kaya and Uyar proposed a hybrid strategy for identifying hepatitis illness using a rough set and an extreme machine learning algorithm. The hepatitis illness dataset that was used came from the UCI library. Using rudimentary set theory, 20 reducts with three to seven qualities were created. The records with missing values are eliminated from each reduct after the reducts are selected. The back propagation neural network was used to classify selected reducts, and the accuracy was 98.6 percent.

All of the studies described above are only a small sample of the vast amount of research that has been done in applying machine learning and data mining techniques to a variety of healthcare areas for forecasting and pattern identification.

1.5Contribution of Project

1.5.1Innovativeness :

This project aims to create a machine learning model that has the highest accuracy in predicting breast cancer in its early stages, using parameters from mammography tests to determine the type of tumor the patient has (benign or malignant).

1.5.2Usefulness :

Early diagnosis boosts the chances of successful therapy and survival, but the process is time-consuming and frequently results in pathologists disagreeing. Computer-aided diagnostics methods have shown promise in terms of increasing diagnostic accuracy. However, early detection and prevention can greatly minimize the likelihood of death. Breast cancer should be detected as soon as feasible.

1.6ReportOrganization

Theremainingsectionofthereport is structuredas follows:

- **Chapter 2** provides detailed technical requirements, analysis and design of thisproject
- **Chapter3**providesConstruction,implementationdetailsofthisproject
- **Chapter4**providesConclusionandscopeofthisproject

Chapter-2

SYSTEMANALYSIS

2.1Objective :

The objective of this study is to assess the prediction accuracy of the classification algorithms in terms of efficiency and effectiveness.

2.2ProblemStatement

One of the leading causes of cancer death worldwide is breast cancer. Early diagnosis boosts the chances of successful therapy and survival, but the process is time-consuming and frequently results in pathologists disagreeing. Computer-aided diagnostics methods have shown promise in terms of increasing diagnostic accuracy. However, early detection and intervention can greatly minimize the likelihood of death. Breast cancer should be detected as soon as feasible.

2.3FunctionalRequirement

Wisconsin datasets were used to create the data set. The dataset was divided into training and testing sets in order to implement the machine learning methods. A comparison of all six methods will be performed. The website will be given a model of the algorithm that produces the best results. The website will be built using the flask Python framework. The database will be hosted on Xampp, Firebase, or the native Python and flask libraries. The UCI Machine Learning Repository has this data set available. It is made up of 32 multivariate real-world properties. The total number of cases in this data collection is 569, and there are no missing values. The proposed system's procedure is as follows:

1.Our website is used by the patient to schedule an appointment.

2.The patient will next meet with the doctor in person for the appointment.

3.The doctor will manually examine the patient before doing a breast mammography or ultrasound. This ultrasound will produce a picture of the breast, whether it has lumps or not.

4.A biopsy will be conducted if the lumps are discovered. The dataset's features are based on a digitized image of the Fine Needle Aspirate (FNA).

5.The doctor will input those statistics into the system, and the model will determine if it is a benign or malignant malignancy.

6.After that, the report will be sent to the patient on their account.

2.3.1InterfaceRequirement:

Here in this project, we developed an interface where the user will find a web page where theycangiveinput asfilepathof themusic filesandgets thepredicted outputof musictype.

- Screen1to acceptuserinput.
- Field1 acceptsparametersof thegiven fields or attributes.
- Submitbuttonto sendtheparametersand get thepredictedoutput.
- Screen2displayspredictedtumortype andtumorname i.e., Benign or Malignant

2.4NonFunctionalRequirement :

Non functional requirements are those requirements of the system which are not directlyconcerned with specific functionality delivered by the system. These are mainly concerned withthe functionalities that are not mentioned as the core functionalities of project. They may berelatedto emergent properties such asreliability,usabilityetc.

- Easeofuse.
- Availability
- Reliability
- Maintainability

2.5SoftwareDetails :

- AnacondaDistribution(v5.1)
- Python(3.6.5)
- JupyterNotebook
- VSCode
- Flask :Micro Web Application framework for Python

2.6HardwareDetails :

- Operatingsystem:Windows 7ornewer, 64-bitmacOS10.9+, orLinux.
- Systemarchitecture: 64-bitx86, 32-bitx86withWindowsorLinux.
- CPU: IntelCore2QuadCPUQ6600@2.40GHzorgreater.
- RAM:4GBor greater.

.

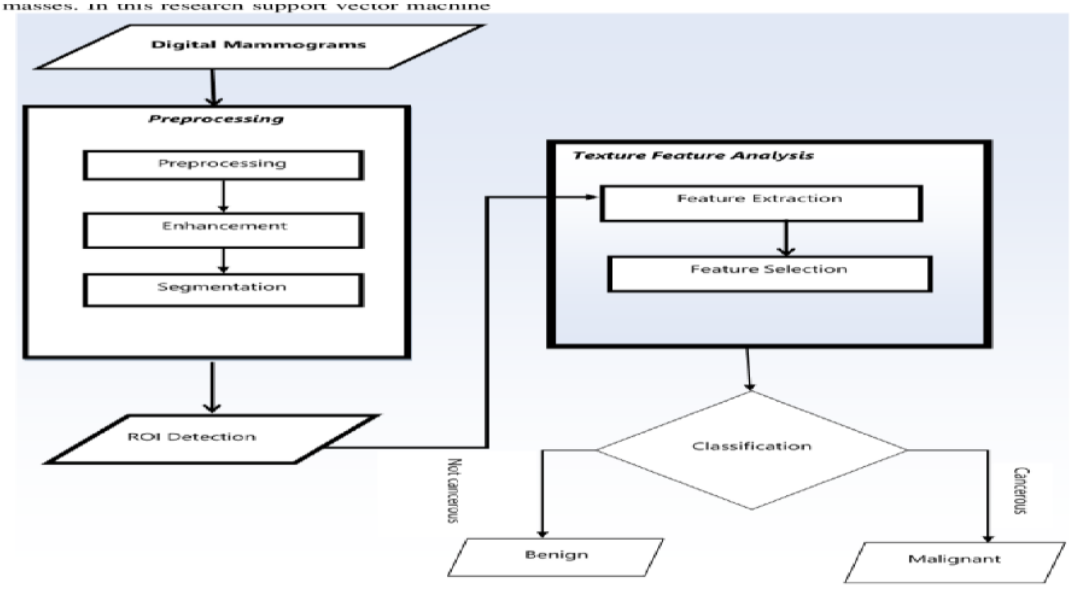
2.7Architecture :

29

An architecture of a system defines the working model and behavior of a system connected totask division. It provides an overview of a system in a diagrammatic and figurative structures.Simple,wecan understandthe modellingand descriptionof asystemfroman architectureofa

System.

it masses. In this research support vector machine

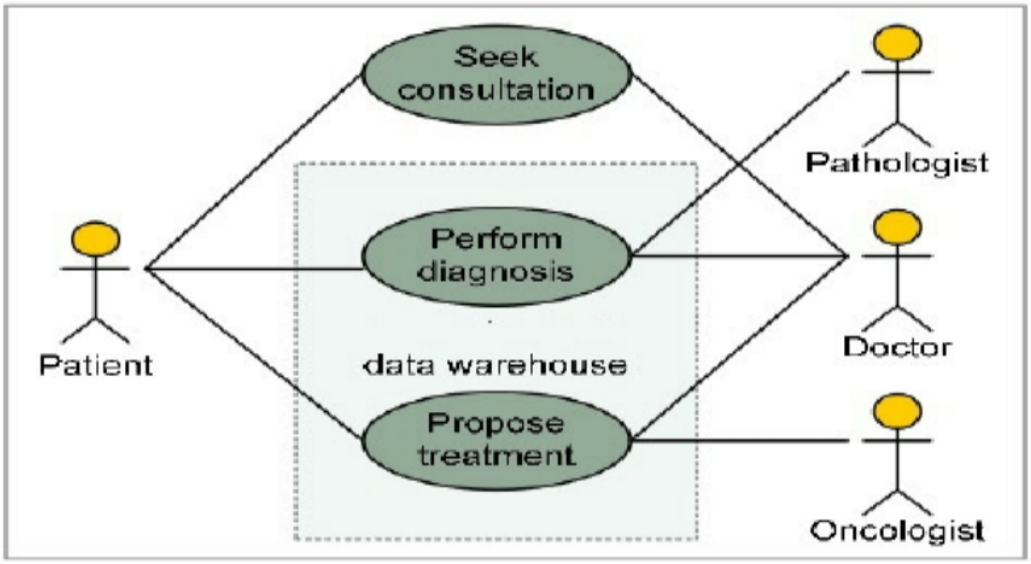


2.8 Process Design :

2.8.1 Usecase diagram :

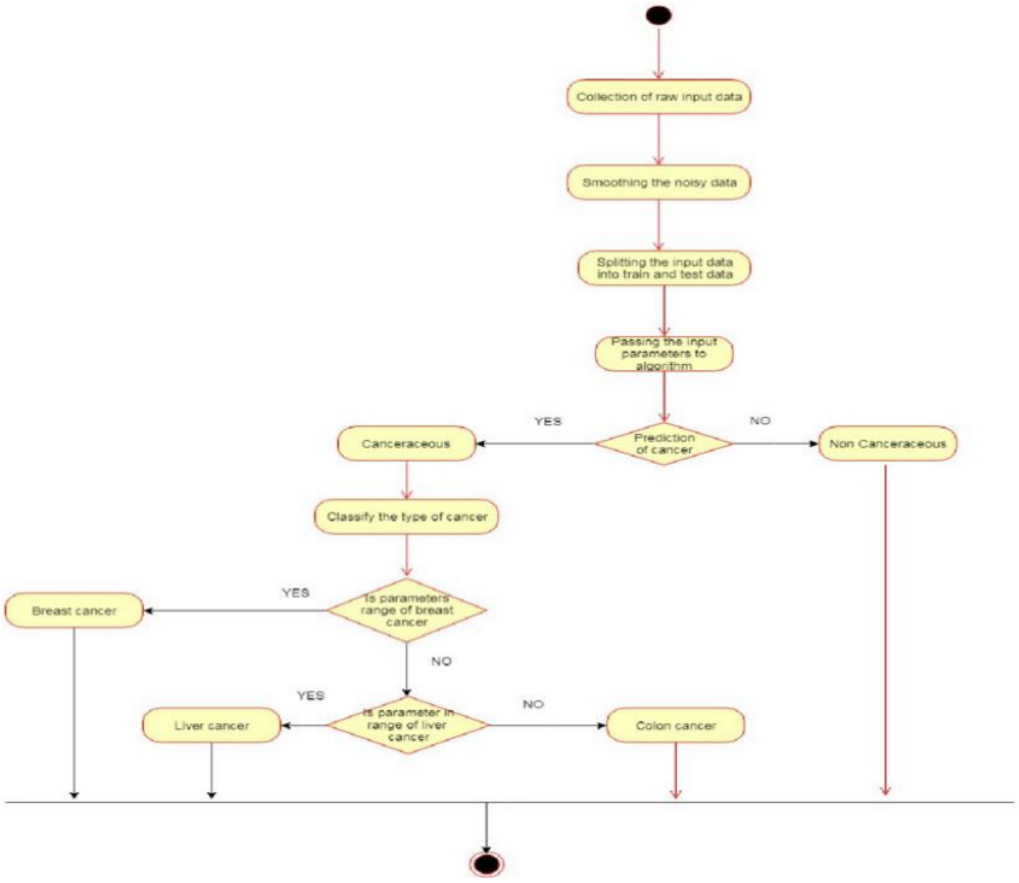
Use case diagrams represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between a user and a system.

Thus a use case is a set of scenarios tied together by some goal. The use case diagrams are drawn for exposing the functionalities of the system.



2.8.2Activitydiagram :

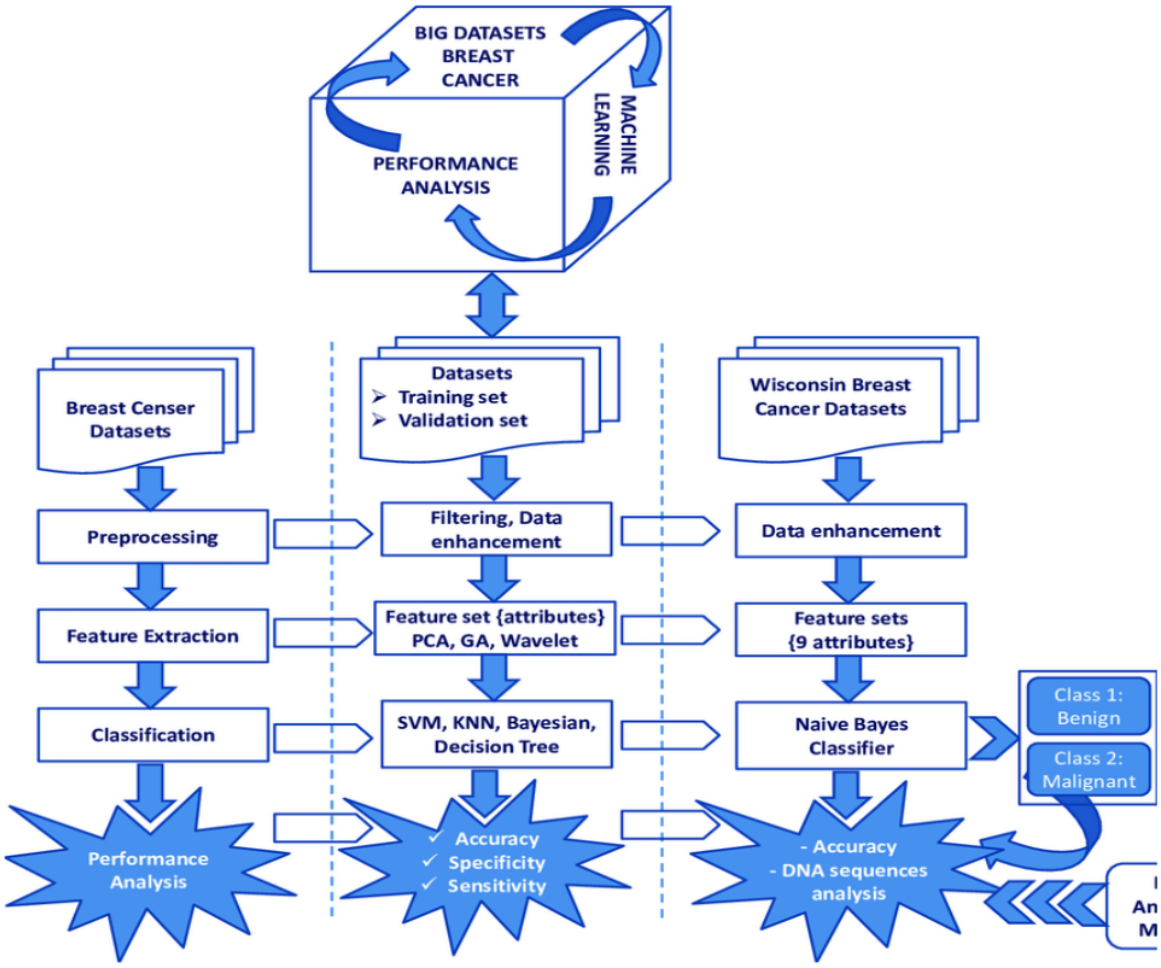
Theactivitydiagramisagraphicalrepresentationforrepresentingtheflowofinteraction within specific scenarios. It is similar to a flowchart in which variousactivitiesthat can beperformed in thesystemarerepresented.



2.8.3Sequencediagram :

Inthesequencediagramhowtheobjectinteractswiththeotherobjectisshown.Therearesequence of events that arerepresented.

It is a period situated perspective on the association between items to achieve a conductobjectiveof theframework



Chapter-3

IMPLEMENTATION

Theimplementationoftheprojectisdonewiththehelpofpythonlanguage.Tobeparticular,for thepurposeof machine learning Jupyter isbeingused.

3.1DataPreparation:

3.1.1LoadRequiredLibraries

```
In [2]: import pandas as pd
        import numpy as np
        import  matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

3.1.2LoadData:

Import the dataset from the local directories.

```
In [3]: #Load the data
        # https://www.kaggle.com/uciml/breast-cancer-wisconsin-data
        df = pd.read_csv('Breast-Cancer-Detection.csv')
        df.head()
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	tex
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

3.2Size and Names:

Finding the DataSet Size and Attribute Names,
The size and Names of the Attributes from the dataset.

```
In [4]: #count the number of rows and columns in the dataset
        df.shape

Out[4]: (569, 33)

In [5]: #count the number of empty(Nan,NaN, na) values in each column
        df.isna().sum()

Out[5]: id                         0
        diagnosis                  0
        radius_mean                0
        texture_mean               0
        perimeter_mean             0
        area_mean                  0
        smoothness_mean            0
        compactness_mean           0
        concavity_mean             0
        concave points_mean        0
        symmetry_mean              0
        fractal_dimension_mean     0
        radius_se                  0
        texture_se                 0
        perimeter_se               0
        area_se                    0
        smoothness_se              0
        compactness_se             0
        concavity_se               0
        concave points_se          0
        symmetry_se                0
        fractal_dimension_se       0
        radius_worst               0
        texture_worst              0
        perimeter_worst            0
        area_worst                 0
        smoothness_worst           0
        compactness_worst          0
        concavity_worst            0
        concave points_worst       0
        symmetry_worst             0
        fractal_dimension_worst    0
        Unnamed: 32              569
        dtype: int64
```

3.3Types of Tumor:

Classifying the rows which are Malignant or Benign**:**

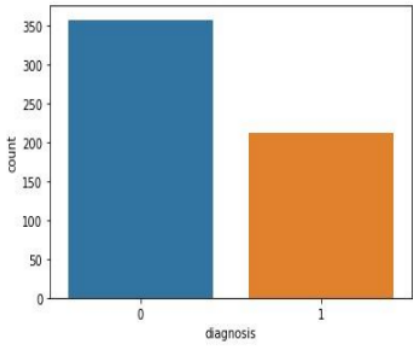
```
In [8]: # get a count of the number of Malignant (M) or benign(B) cells
        df['diagnosis'].value_counts()

Out[8]: B    357
        M    212
        Name: diagnosis, dtype: int64

In [25]: #Visualize the count
         sns.countplot(df['diagnosis'], label='count')

         C:\Users\NIKHIL KUDIKYALA\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as
         a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
         explicit keyword will result in an error or misinterpretation.
           warnings.warn(

Out[25]: <AxesSubplot:xlabel='diagnosis', ylabel='count'>
```



3.4DefineModel

3.4.1RANDOM FOREST:

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Random Forest : Classification Approach

The random forest is formed in two phases: the first is to combine N decision trees to build the random forest, and the second is to make predictions for each tree created in the first phase.

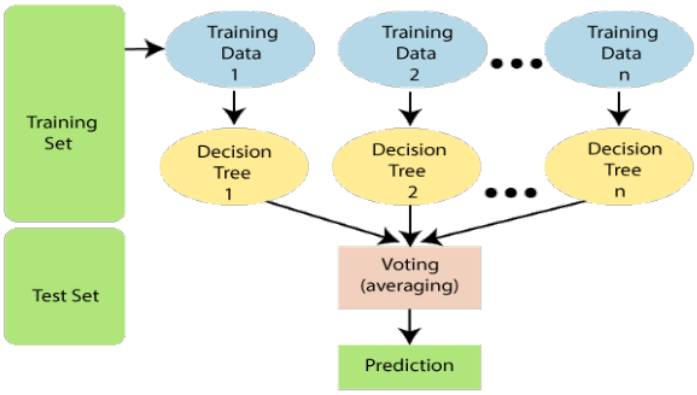
The following steps and diagram can be used to demonstrate the working process:

Step 1: Pick K data points at random from the training set.

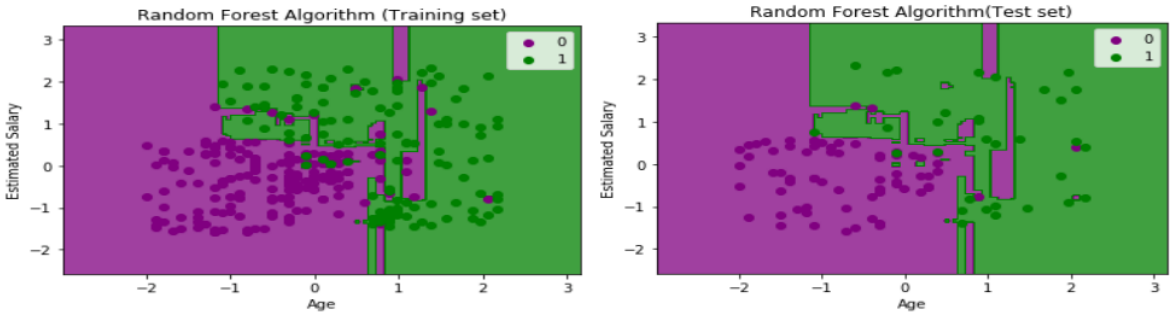
Step 2: Create decision trees for the data points you've chosen (Subsets).

Step 3: Decide on the number N for the decision trees you wish to create.

Steps 1–4 are repeated.



Principlediagramof RANDOM FOREST



3.4.2Assumptions for Random Forest

Because the random forest combines numerous trees to forecast the dataset's class, some decision trees may correctly predict the output while others may not. However, when all of the trees are combined, the proper result is predicted. As a result, two assumptions for a better Random forest classifier are as follows:

o The dataset's feature variable should have some actual values so that the classifier can predict accurate outcomes rather than a guess.

o Each tree's predictions must have very low correlations.

In [32]:
```
#split the datset into independent (X) and dependent (y) dataset
X = df.iloc[:,2:31].values
Y = df.iloc[:,1].values
```

In [33]:
```
#split the dataset into 75% training and 25% testing
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size= 0.25, random_state = 0)
```

In [34]:
```
#Scale the data(Feature Scaling)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [35]:
```
from pandas.core.common import random_state
#create a function for the model
def models(X_train, Y_train):

  #Logistic Regression
  from sklearn.linear_model import LogisticRegression
  log = LogisticRegression(random_state=0)
  log.fit(X_train, Y_train)

  #decision Tree
  from sklearn.tree import DecisionTreeClassifier
  tree = DecisionTreeClassifier(criterion = 'entropy', random_state=0)
  tree.fit(X_train, Y_train)

  #random forest classifier
  from sklearn.ensemble import RandomForestClassifier
  forest = RandomForestClassifier(n_estimators =10, criterion = 'entropy', random_state = 0)
  forest.fit(X_train, Y_train)

  #print the model accuracy on the training data
  print('[0]Logistic Regression Training Accuracy:', log.score(X_train,Y_train))
  print('[1]Decision Tree Classifier Training Accuracy:', tree.score(X_train,Y_train))
  print('[2] Random Forest Classifier Training Accuracy:', forest.score(X_train,Y_train))

  return log,tree,forest
```

3.5 Evalution

3.5.1 Metrics

- **Accuracy:** Refers to the percentage of correctly classified test samples. This metrics evaluates how accurate the model prediction is compared to the true data.

```
#Fitting Decision Tree classifier to the training set
from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
classifier.fit(x_train, y_train)
```

- This code will generate the accuracy of the metrics.

```
In [38]: #show another way to get metrics of the models
         from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score
         for i in range(len(model) ):
           print('Model', i)
           print( classification_report(Y_test, model[i].predict(X_test)))
           print(accuracy_score(Y_test,model[i].predict(X_test)))
           print()
```

3.5.2Accuracy

We classified 2 Types of tumors (malignant and benign)using random Forest classifier. We can get the confusion matrix of three best algorithms. It gave an accuracy of 96%. Random Forest is found to be performing better the another models such as Logistic Regression and Decision Tree . The below table shows accuracy of models.

In [37]:
```
# test model sccuracy on tests data on confusion matrix
from sklearn.metrics import confusion_matrix

for i in range(len(model) ):
  print('Model', i)
  cm = confusion_matrix(Y_test, model[i].predict(X_test))

  TP = cm[0][0]
  TN = cm[1][1]
  FN = cm[1][0]
  FP = cm[0][1]

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

```
Model 0
[[86  4]
 [ 3 50]]
Testing Accuracy =  0.951048951048951

Model 1
[[83  7]
 [ 2 51]]
Testing Accuracy =  0.9370629370629371

Model 2
[[87  3]
 [ 2 51]]
Testing Accuracy =  0.965034965034965
```

Models	**Accuracy**
Decision Tree	93%
Logistic Regression	95%
Random Forest	96%

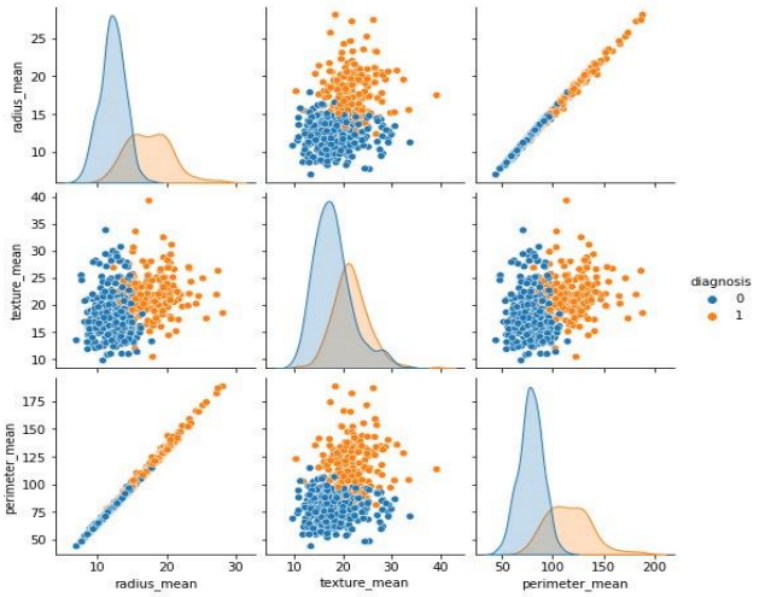
We have selected the three best Models out of Supervised and Unsupervised Machine Learning .

Out of these three Random Forest has the highest Accuracy with 96 percentage.

In [27]:
```
#encode the categorical data values
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
df.iloc[:,1] = labelencoder_Y.fit_transform(df.iloc[:,1].values)
```

In [28]:
```
#create a pair plot
sns.pairplot(df.iloc[:,1:5],hue ='diagnosis')
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x2254499f760>
```

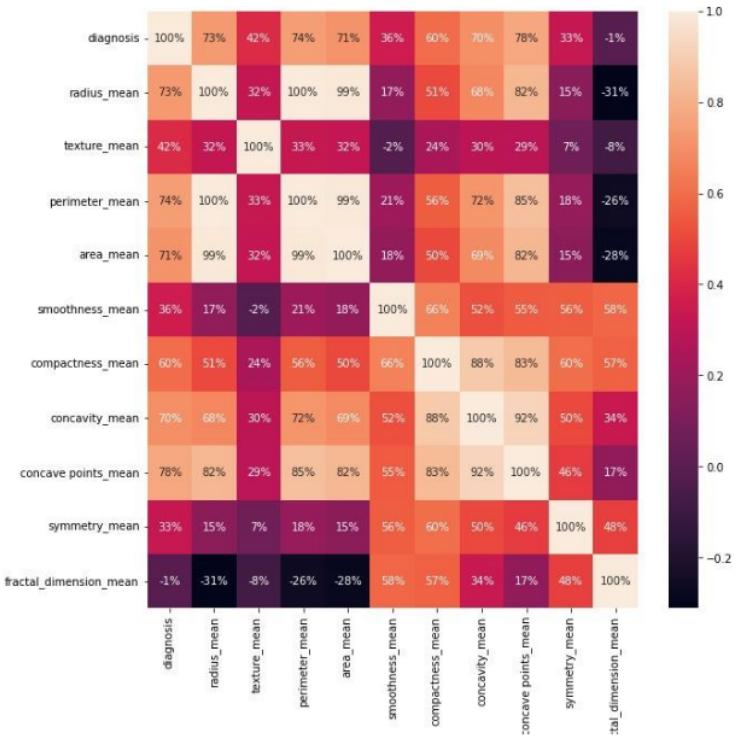


3.6Correlation between the Attributes:

This plot will show us the relation between the Attributes.

```
In [31]: #visualize the correlation
         plt.figure(figsize=(10,10))
         sns.heatmap(df.iloc[:,1:12].corr(), annot =True, fmt = '.0%')
```

```
Out[31]: <AxesSubplot:>
```



3.7Predictions

We predicted the type of tumor going to be form in the Breast using this Random Forest machine learning model. The predicted result is shown below:

```
In [39]: #print the prediction of random foest classifier model
         pred = model[2].predict(X_test)
         print(pred)
         print()
         print(Y_test)
```

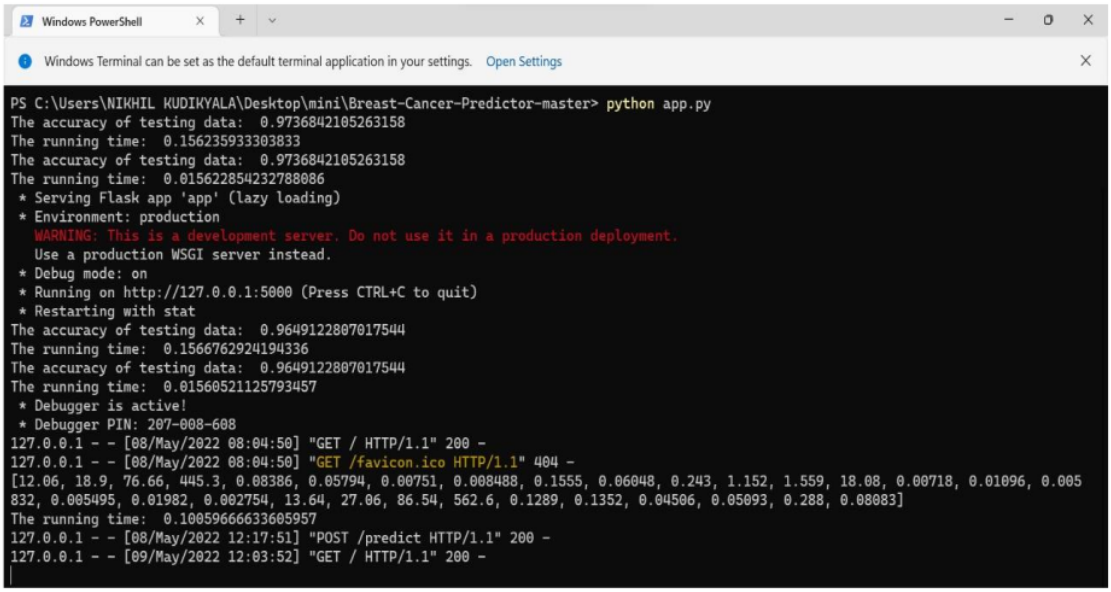
```
[1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 0
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1
 1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]
```

Using the Flask Server for Front End :

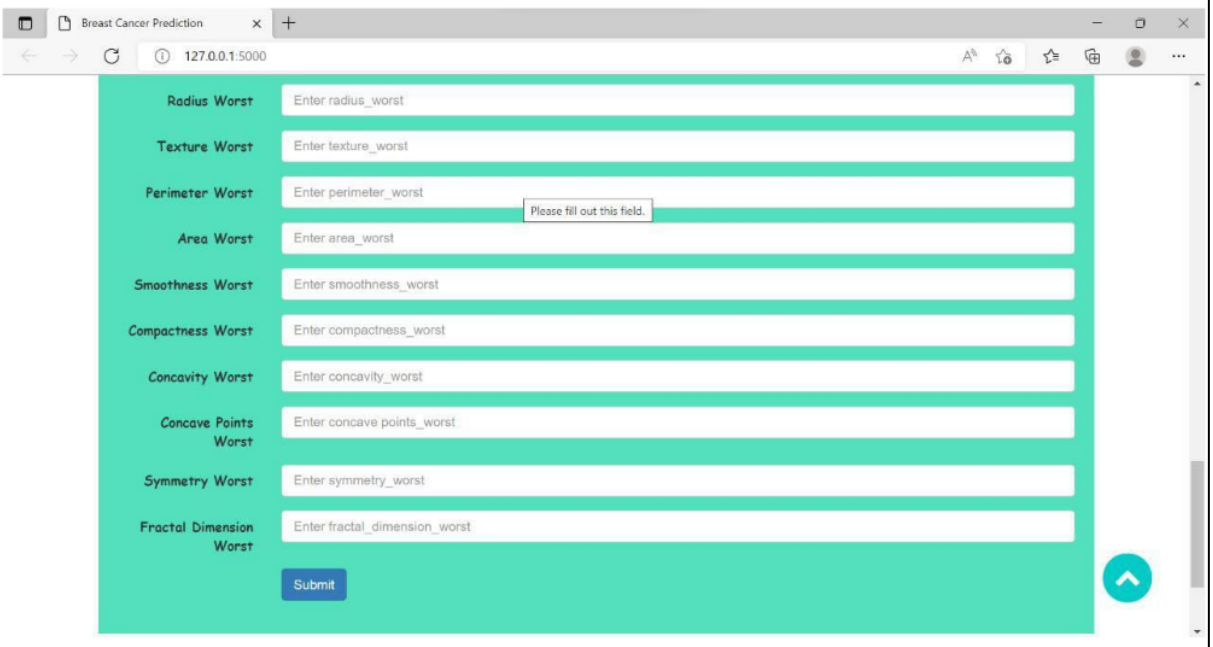
Use the Command "Python app.py" in Windows Terminal

We can get an local Host "http://127.0.0.1:5000"



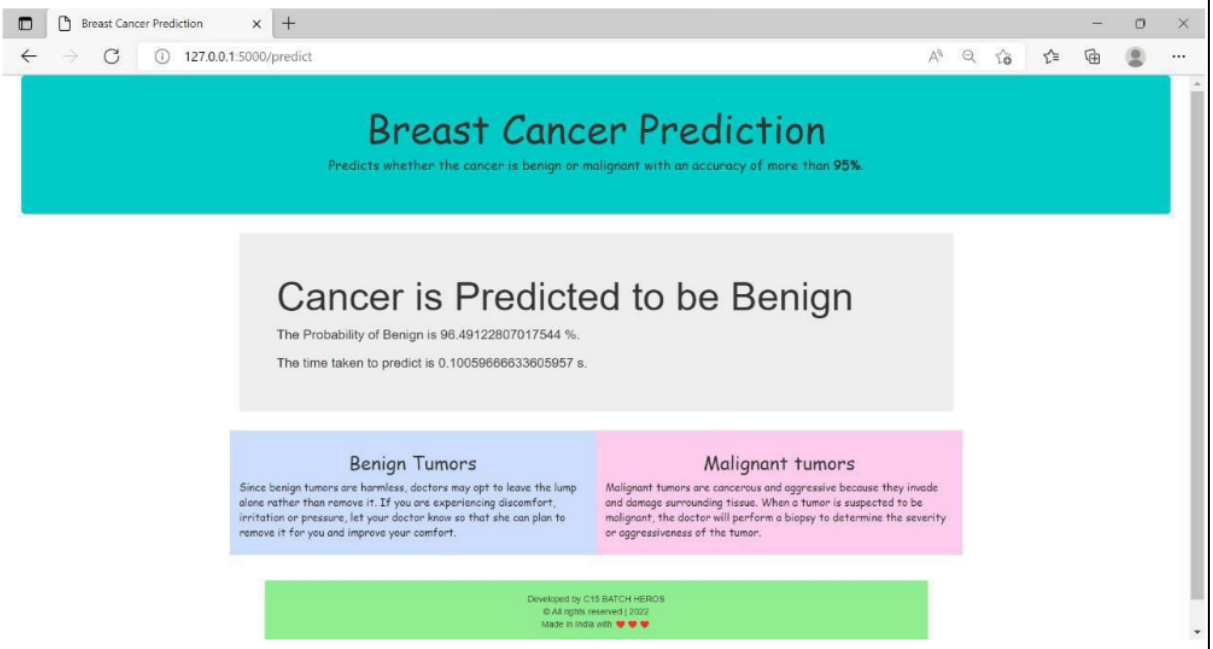
This is the outlook of how the website looks which takes parameters as input:


Breast Cancer Prediction
127.0.0.1:5000
Texture SE
Enter texture_se
Perimeter SE
Enter perimeter_se
Area SE
Enter area_se
Smoothness SE
Enter smoothness_se
Compactness SE
Enter compactness_se
Concavity SE
Enter concavity_se
Concave Points SE
Enter concave points_se
Symmetry SE
Enter symmetry_se
Fractal Dimension SE
Enter fractal_dimension_se
Radius Worst
Enter radius_worst
Texture Worst
Enter texture_worst
Perimeter Worst
Enter perimeter_worst


Breast Cancer Prediction
127.0.0.1:5000
Radius Worst
Enter radius_worst
Texture Worst
Enter texture_worst
Perimeter Worst
Enter perimeter_worst
Please fill out this field.
Area Worst
Enter area_worst
Smoothness Worst
Enter smoothness_worst
Compactness Worst
Enter compactness_worst
Concavity Worst
Enter concavity_worst
Concave Points Worst
Enter concave points_worst
Symmetry Worst
Enter symmetry_worst
Fractal Dimension Worst
Enter fractal_dimension_worst
Submit

Here we pass the values of the attributes to get the probability and Time Taken to find the type of tumor is going to be developed in the future.

after the user passing the parameter as input, then click enter the classification process begins. The user has to wait for some time to get the output which type of tumor is going to be form in the future. The output looks like this:



Chapter-4

CONCLUSIONANDSCOPE

4.1Conclusion

Machine learning is the process of automatically recognizing meaningful patterns in data. Over the previous few decades, it has become a popular technique in practically any endeavour that involves information extraction from massive data sets. Anti-spam software learns to screen our email messages, while software that learns to recognize frauds protects credit card transactions. Smartphones with intelligent personal assistant software learn to interpret voice commands, and digital cameras learn to recognize faces. Automobile accident avoidance systems are built using machine learning techniques.

Scope

This project's future potential is enormous. We can speculate on whether a gene is mutating, if so, which one is altering, and what the risks are of developing breast cancer. Furthermore, we can try to anticipate which sort of breast cancer the patient is likely to get, as well as what other types of tumors and cancers may emerge, so that additional diagnosis and tests can help avoid it, as prevention is always preferable than cure (reduction).

8

References

1.M. R. Al-Hadidi, A. Alarabeyyat and M. Alhanahnah, "Breast Cancer Detection Using K-Nearest Neighbor Machine Learning Algorithm," 2016 9th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, 2016, pp. 35-39.

2.C. Deng and M. Perkowski, "A Novel Weighted Hierarchical Adaptive Voting Ensemble Machine Learning Method for Breast Cancer Detection," 2015 IEEE International Symposium on Multiple-Valued Logic, Waterloo, ON, 2015, pp. 115-120.

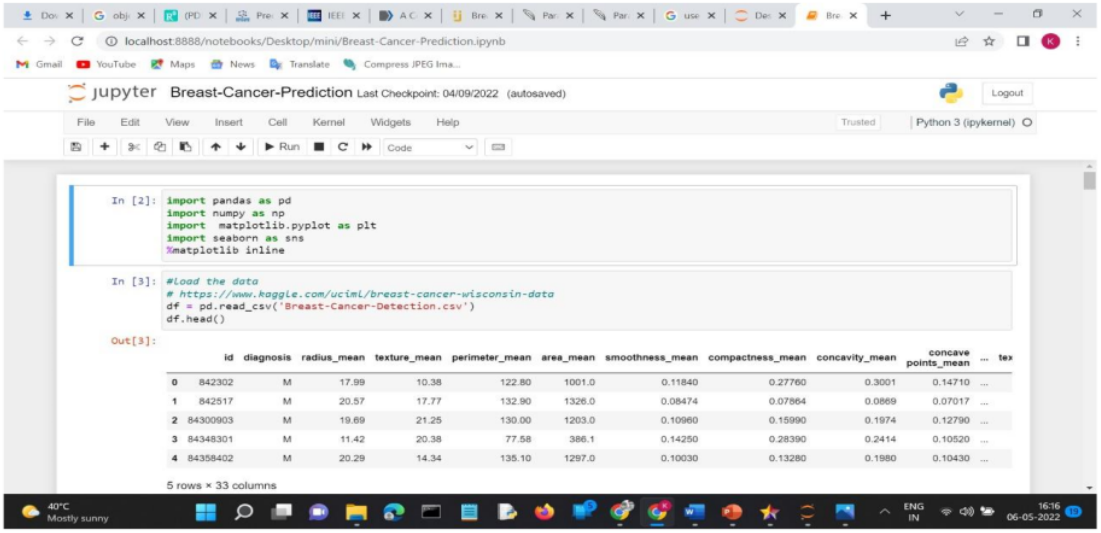
3.A. Qasem et al., "Breast cancer mass localization based on machine learning," 2014 IEEE 10th International Collo quium on Signal Processing and its Applications, Kuala Lumpur, 2014, pp. 31-36.

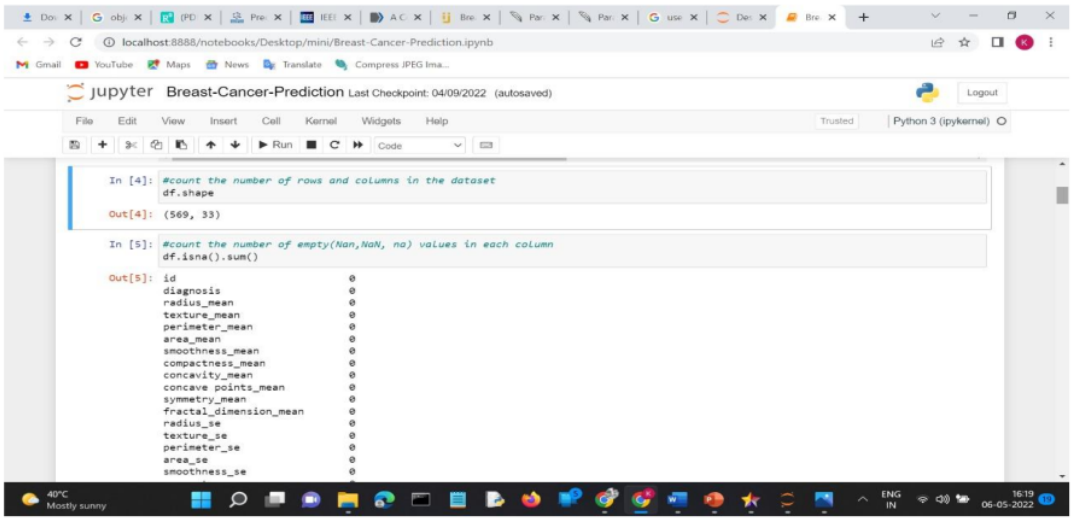
A. Osareh and B. Shadgar, "Machine learning techniques to diagnose breast cancer," 2010 5th International Symposium on Health Informatics and Bioinformatics, Antalya, 2010, pp. 114120.

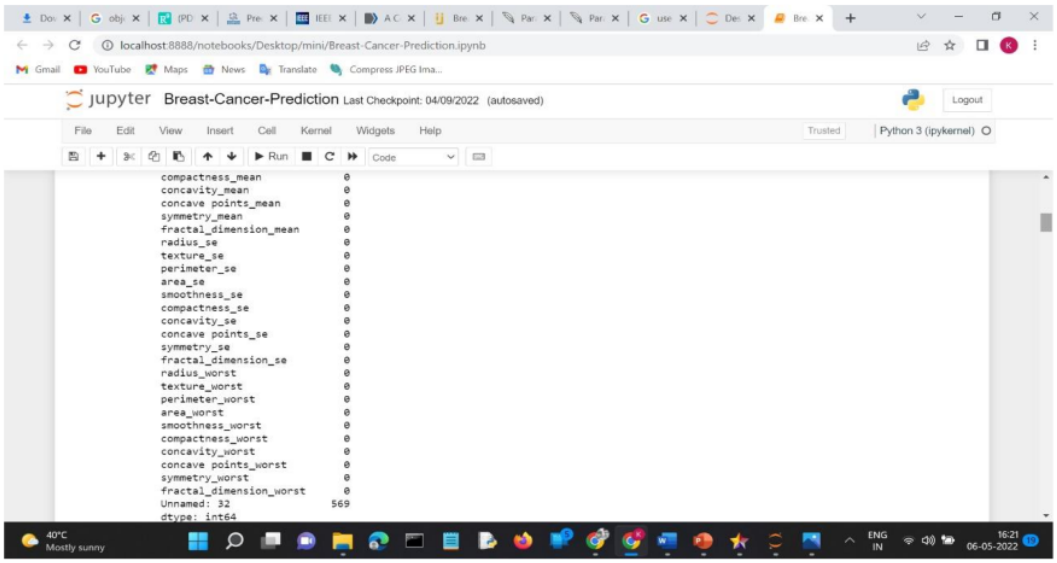
4.J. A. Bhat, V. George and B. Malik, "Cloud Computing with Machine Learning Could Help Us in the Early Diagnosis of Breast Cancer," 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, 2015, pp. 644- 648.

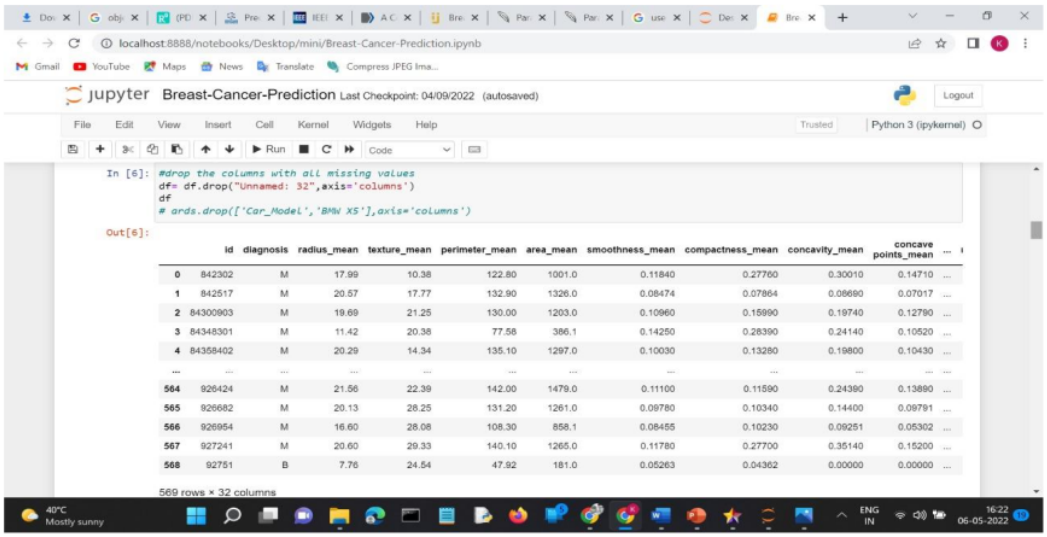
APPENDIXSCR

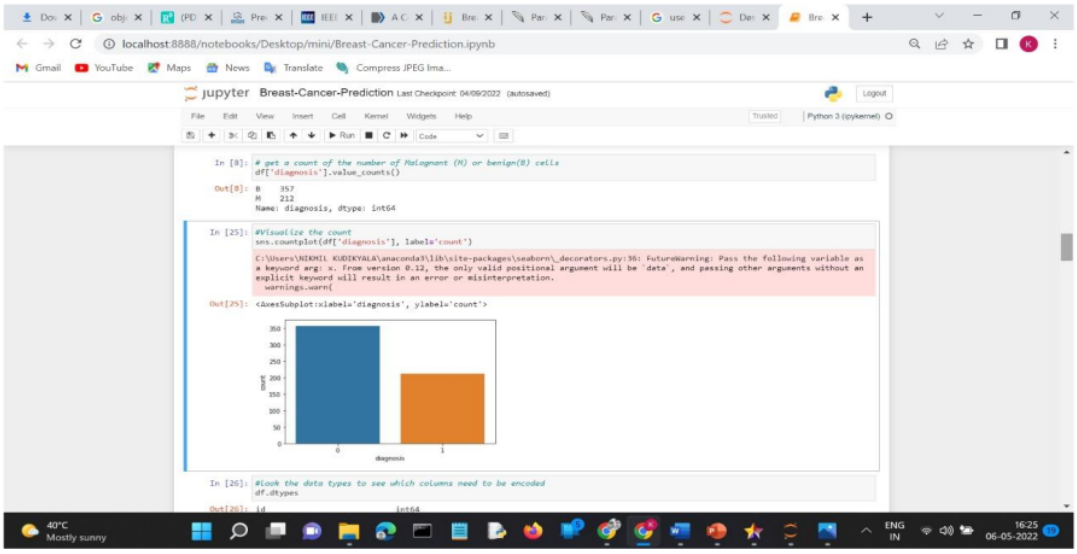
EENSHOTS

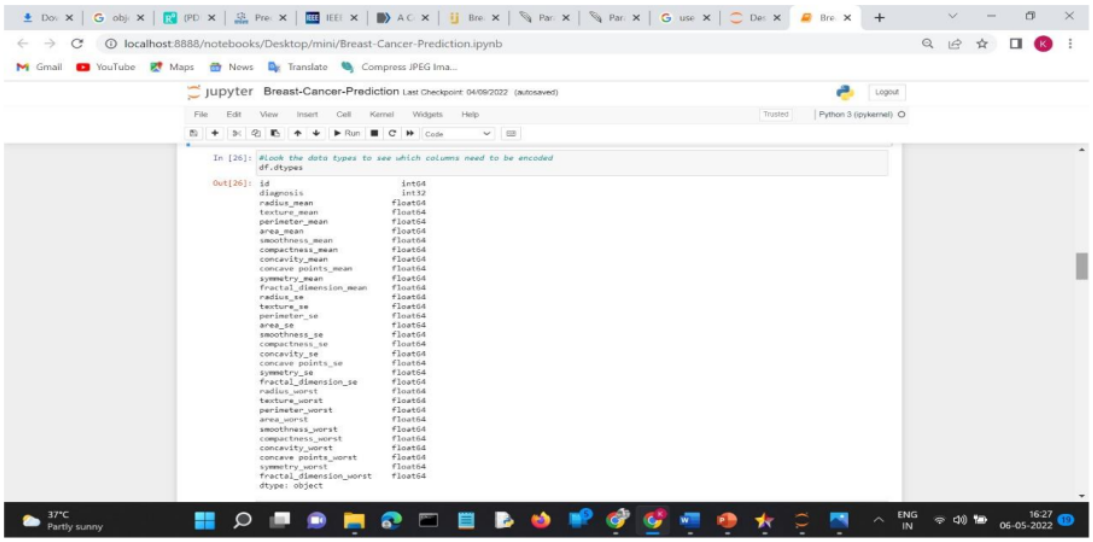


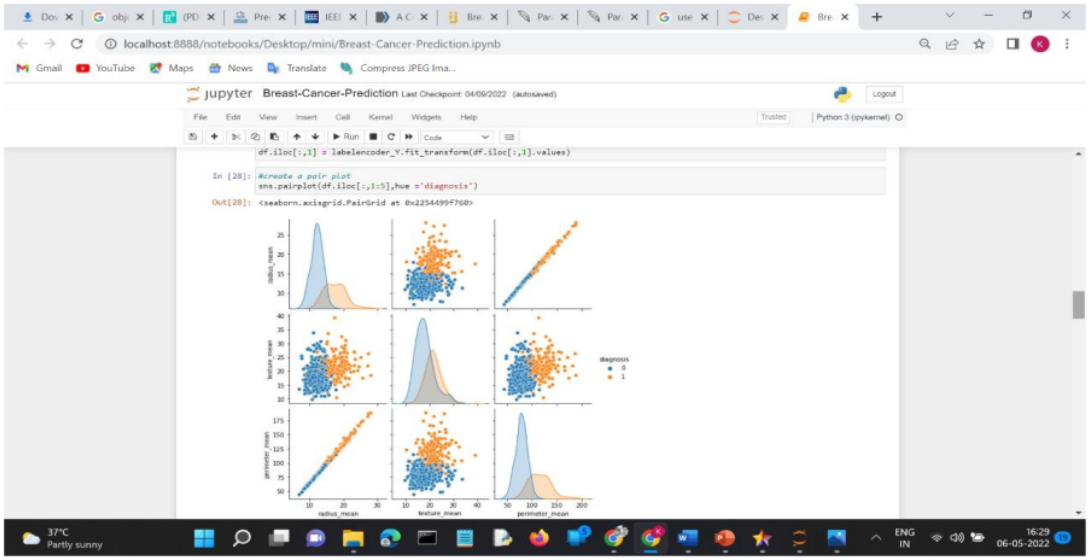


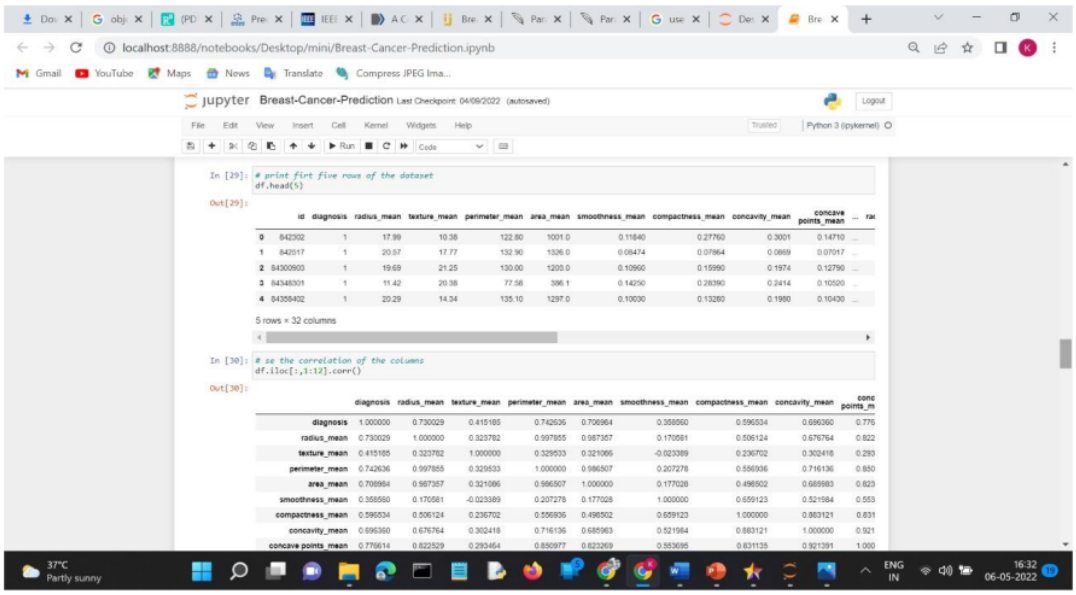

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
compactness_mean 0
concavity_mean 0
concave points_mean 0
symmetry_mean 0
fractal_dimension_mean 0
radius_se 0
texture_se 0
perimeter_se 0
area_se 0
smoothness_se 0
compactness_se 0
concavity_se 0
concave points_se 0
symmetry_se 0
fractal_dimension_se 0
radius_worst 0
texture_worst 0
perimeter_worst 0
area_worst 0
smoothness_worst 0
compactness_worst 0
concavity_worst 0
concave points_worst 0
symmetry_worst 0
fractal_dimension_worst 0
Unnamed: 32 569
dtype: int64
40°C
Mostly sunny
ENG
IN
16:21
06-05-2022

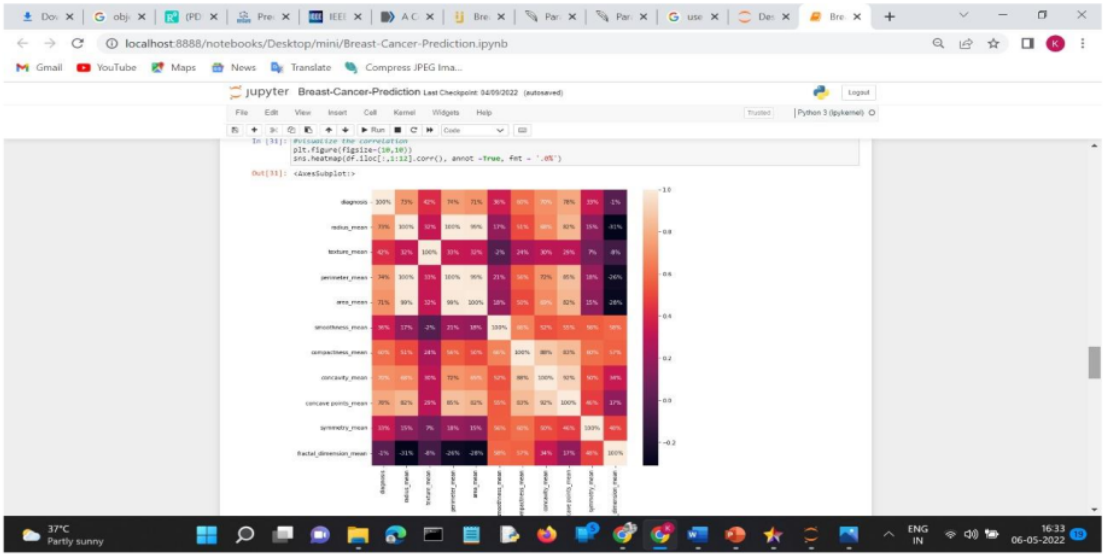

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [6]:
#drop the columns with all missing values
df= df.drop("Unnamed: 32",axis='columns')
df
ards.drop(['Car_Model','BMW X5'],axis='columns')
Out[6]:
id diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean concave points_mean
0 842302 M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.30010 0.14710
1 842517 M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.08690 0.07017
2 84300903 M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.19740 0.12790
3 84348301 M 11.42 20.38 77.58 386.1 0.14250 0.28390 0.24140 0.10520
4 84358402 M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.19800 0.10430
564 926424 M 21.56 22.39 142.00 1479.0 0.11100 0.11590 0.24390 0.13890
565 926682 M 20.13 28.25 131.20 1261.0 0.09780 0.10340 0.14400 0.09791
566 926954 M 16.60 28.08 108.30 858.1 0.08455 0.10230 0.09251 0.05302
567 927241 M 20.60 29.33 140.10 1265.0 0.11780 0.27700 0.35140 0.15200
568 92751 B 7.76 24.54 47.92 181.0 0.05263 0.04362 0.00000 0.00000
569 rows × 32 columns
40°C
Mostly sunny
ENG
IN
16:22
06-05-2022

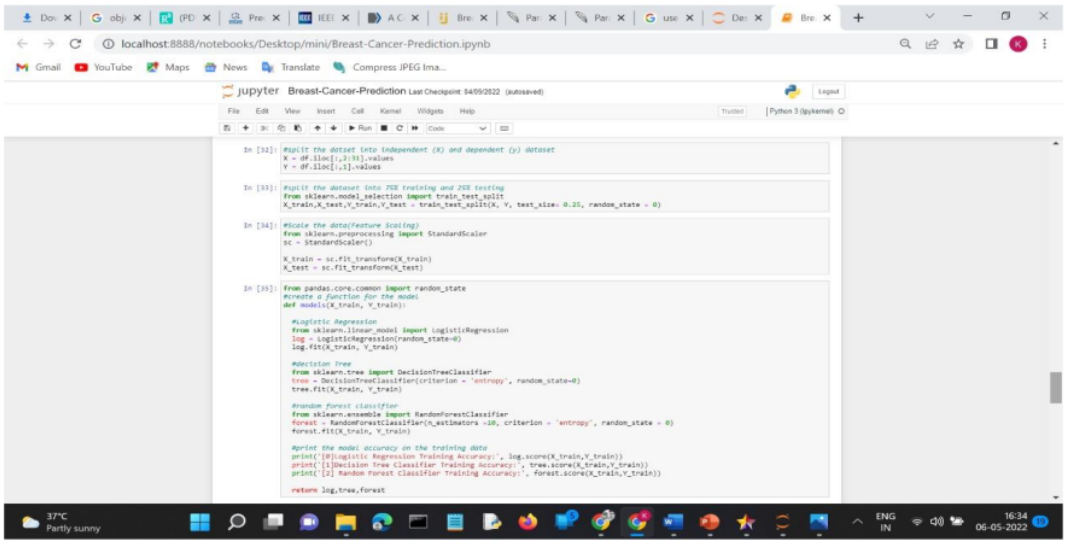

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [8]:
get a count of the number of Malognant (M) or benign(B) cells
df['diagnosis'].value_counts()
Out[8]:
B 357
M 212
Name: diagnosis, dtype: int64
In [25]:
#Visualize the count
sns.countplot(df['diagnosis'], label='count')
C:\Users\NIKHIL KUDIKYALA\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(
Out[25]:
<AxesSubplot:xlabel='diagnosis', ylabel='count'>
count
diagnosis
In [26]:
#look the data types to see which columns need to be encoded
df.dtypes
Out[26]:
id int64
40°C
Mostly sunny
ENG
IN
16:25
06-05-2022

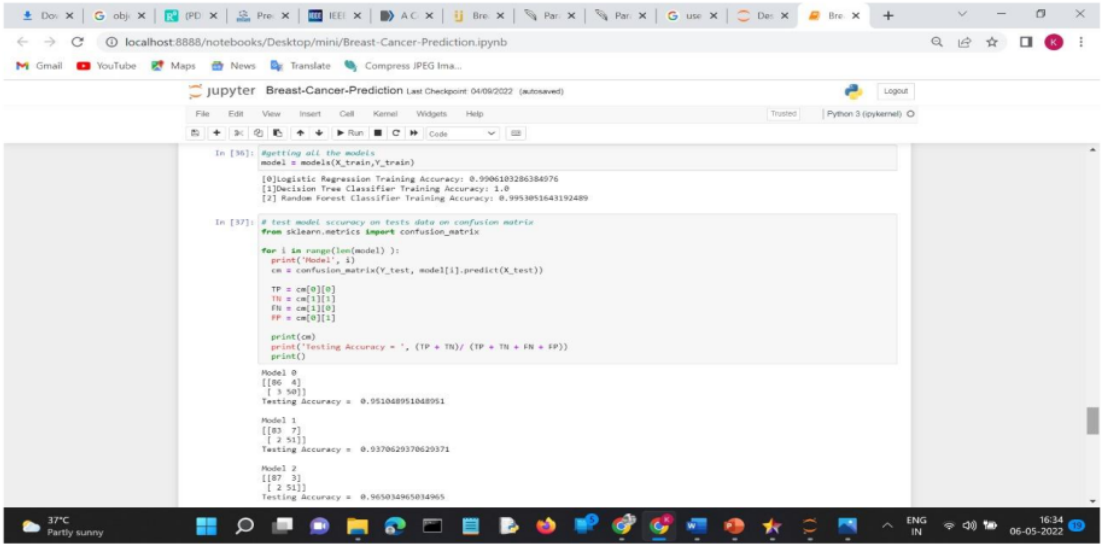

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [26]: #Look the data types to see which columns need to be encoded
df.dtypes
Out[26]:
id int64
diagnosis int32
radius_mean float64
texture_mean float64
perimeter_mean float64
area_mean float64
smoothness_mean float64
compactness_mean float64
concavity_mean float64
concave points_mean float64
symmetry_mean float64
fractal_dimension_mean float64
radius_se float64
texture_se float64
perimeter_se float64
area_se float64
smoothness_se float64
compactness_se float64
concavity_se float64
concave points_se float64
symmetry_se float64
fractal_dimension_se float64
radius_worst float64
texture_worst float64
perimeter_worst float64
area_worst float64
smoothness_worst float64
compactness_worst float64
concavity_worst float64
concave points_worst float64
symmetry_worst float64
fractal_dimension_worst float64
dtype: object

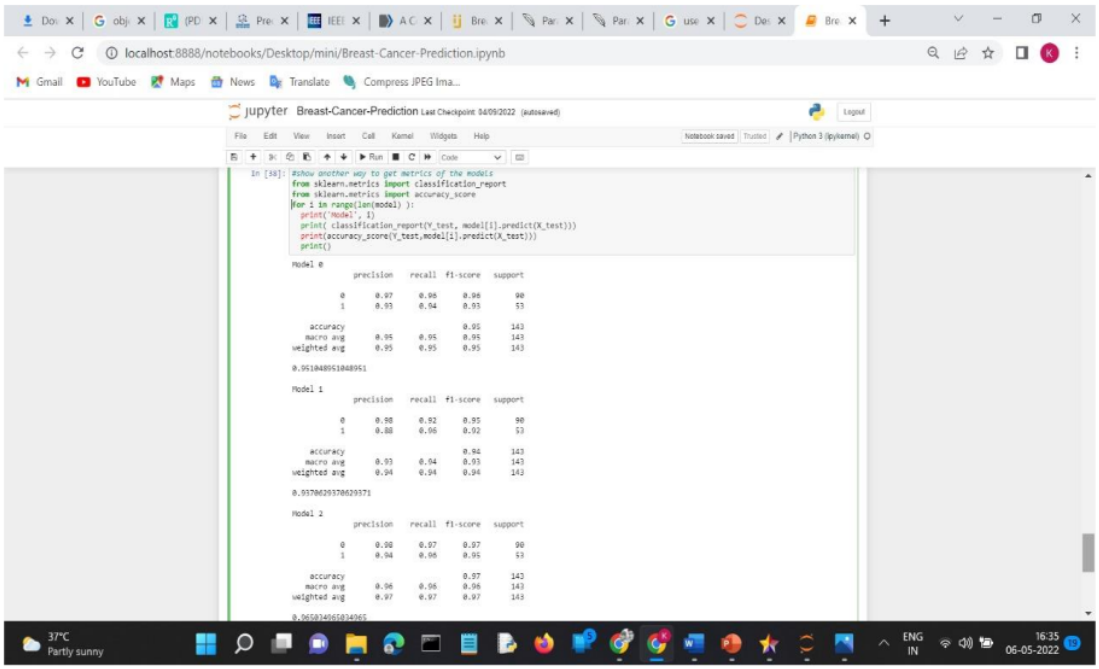

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
df.iloc[:,1] = labelencoder_Y.fit_transform(df.iloc[:,1].values)
In [28]: #create a pair plot
sns.pairplot(df.iloc[:,1:5],hue ='diagnosis')
Out[28]: <seaborn.axisgrid.PairGrid at 0x2254499f760>
radius_mean
texture_mean
perimeter_mean
diagnosis

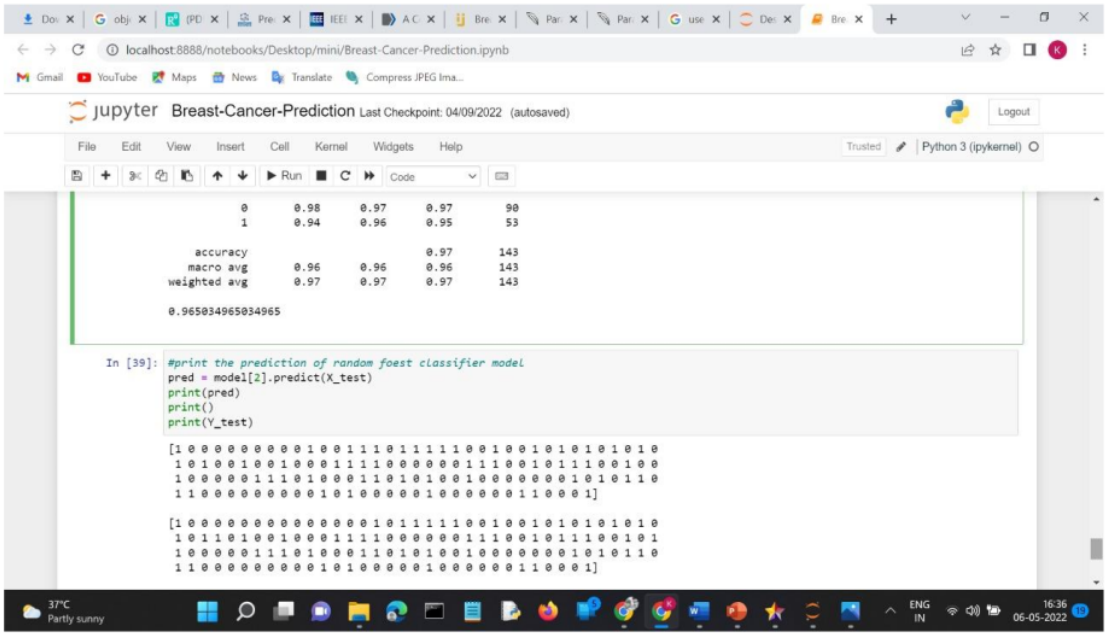

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [29]: # print first five rows of the dataset
df.head(5)
Out[29]:
5 rows × 32 columns
In [30]: # se the correlation of the columns
df.iloc[:,1:12].corr()
Out[30]:
37°C
Partly sunny
ENG
IN
16:32
06-05-2022

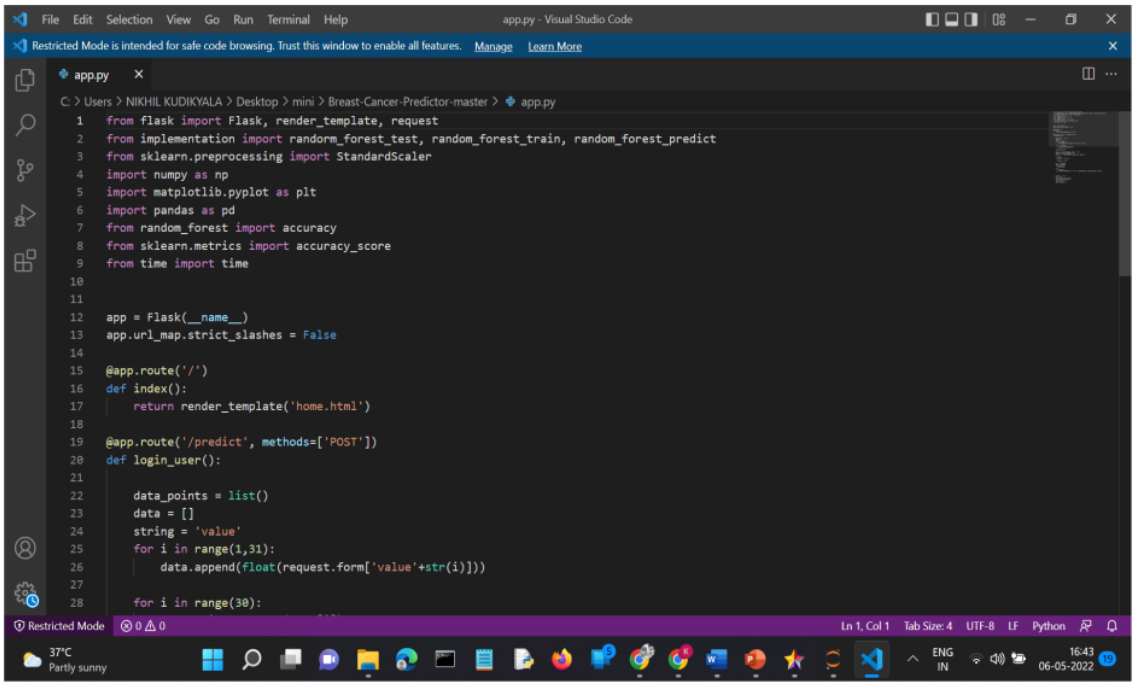

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [31]: #visualize the correlation
plt.figure(figsize=(10,10))
sns.heatmap(df.iloc[:,1:12].corr(), annot =True, fmt = '.0%')
Out[31]: <AxesSubplot:>
37°C
Partly sunny
ENG
IN
16:33
06-05-2022

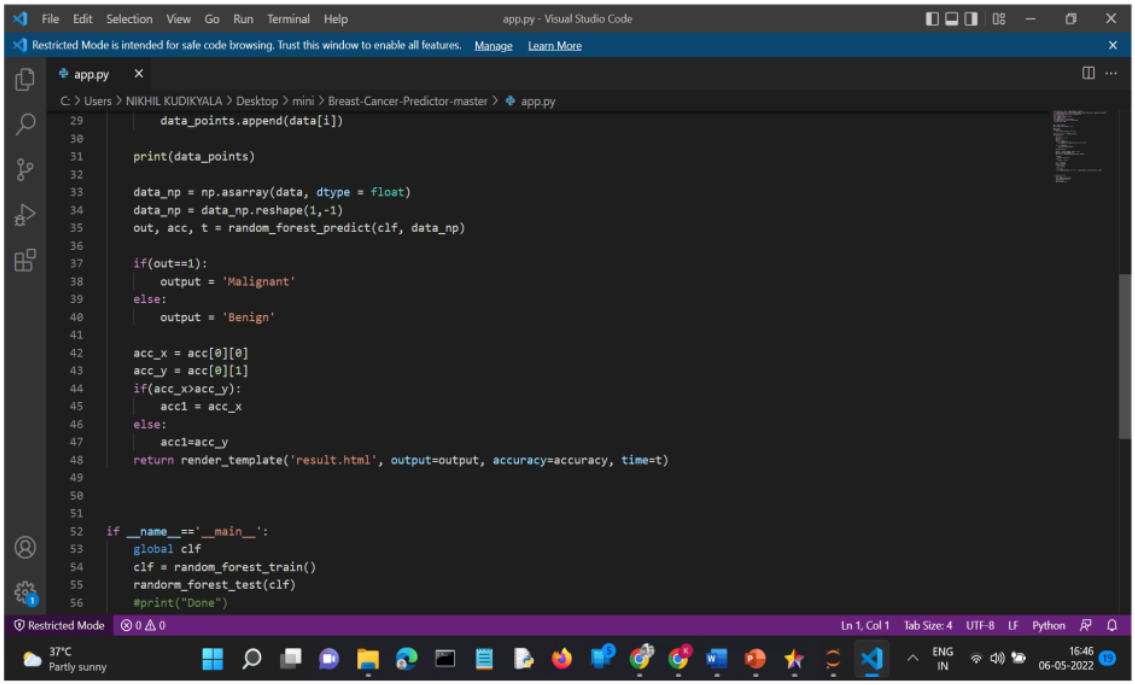

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [32]: #split the dataset into independent (X) and dependent (y) dataset
X = df.iloc[:,2:31].values
Y = df.iloc[:,1].values
In [33]: #split the dataset into 75% training and 25% testing
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size= 0.25, random_state = 0)
In [34]: #Scale the data(Feature Scaling)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
In [35]: from pandas.core.common import random_state
#create a function for the model
def models(X_train, Y_train):
#Logistic Regression
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state=0)
log.fit(X_train, Y_train)
#decision Tree
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(criterion = 'entropy', random_state=0)
tree.fit(X_train, Y_train)
#Random Forest classifier
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators =10, criterion = 'entropy', random_state = 0)
forest.fit(X_train, Y_train)
#print the model accuracy on the training data
print('[0]Logistic Regression Training Accuracy:', log.score(X_train,Y_train))
print('[1]Decision Tree Classifier Training Accuracy:', tree.score(X_train,Y_train))
print('[2] Random Forest Classifier Training Accuracy:', forest.score(X_train,Y_train))
return log,tree,forest

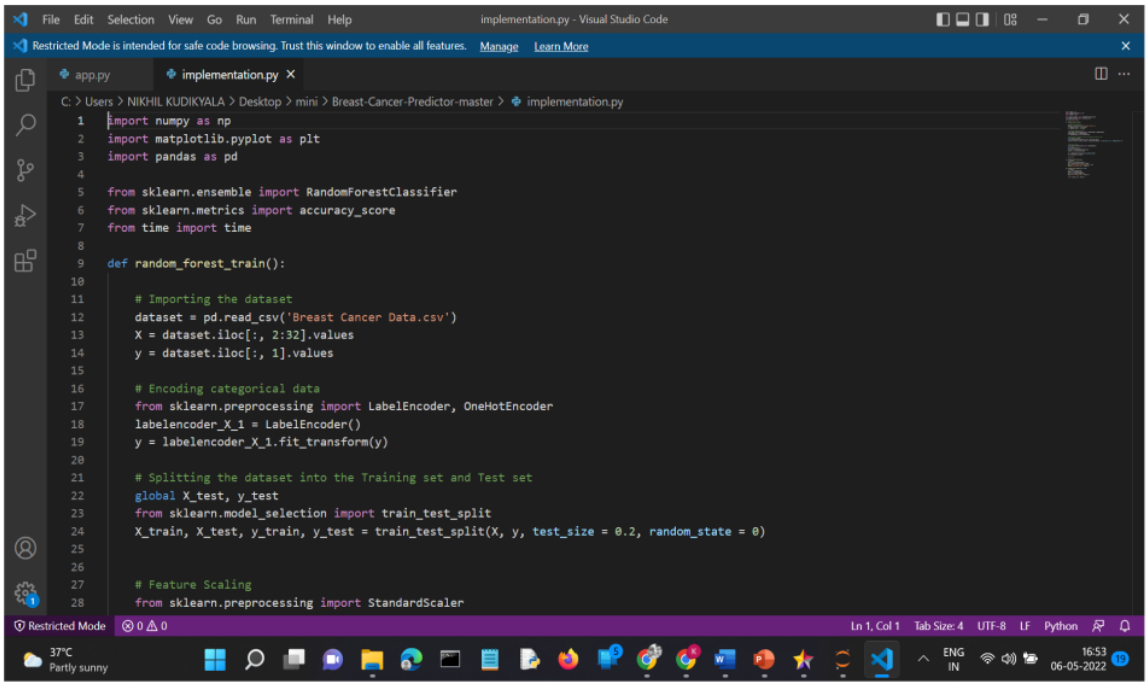

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [36]: #getting all the models
model = models(X_train,Y_train)
[0]Logistic Regression Training Accuracy: 0.9906103286384976
[1]Decision Tree Classifier Training Accuracy: 1.0
[2] Random Forest Classifier Training Accuracy: 0.9953051643192489
In [37]: # test model accuracy on tests data on confusion matrix
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
print('Model', i)
cm = confusion_matrix(Y_test, model[i].predict(X_test))
TP = cm[0][0]
TN = cm[1][1]
FN = cm[1][0]
FP = cm[0][1]
print(cm)
print('Testing Accuracy = ', (TP + TN)/ (TP + TN + FN + FP))
print()
Model 0
[[86 4]
[3 50]]
Testing Accuracy = 0.9510489510489511
Model 1
[[83 7]
[2 51]]
Testing Accuracy = 0.9370629370629371
Model 2
[[87 3]
[2 51]]
Testing Accuracy = 0.9650349650349651

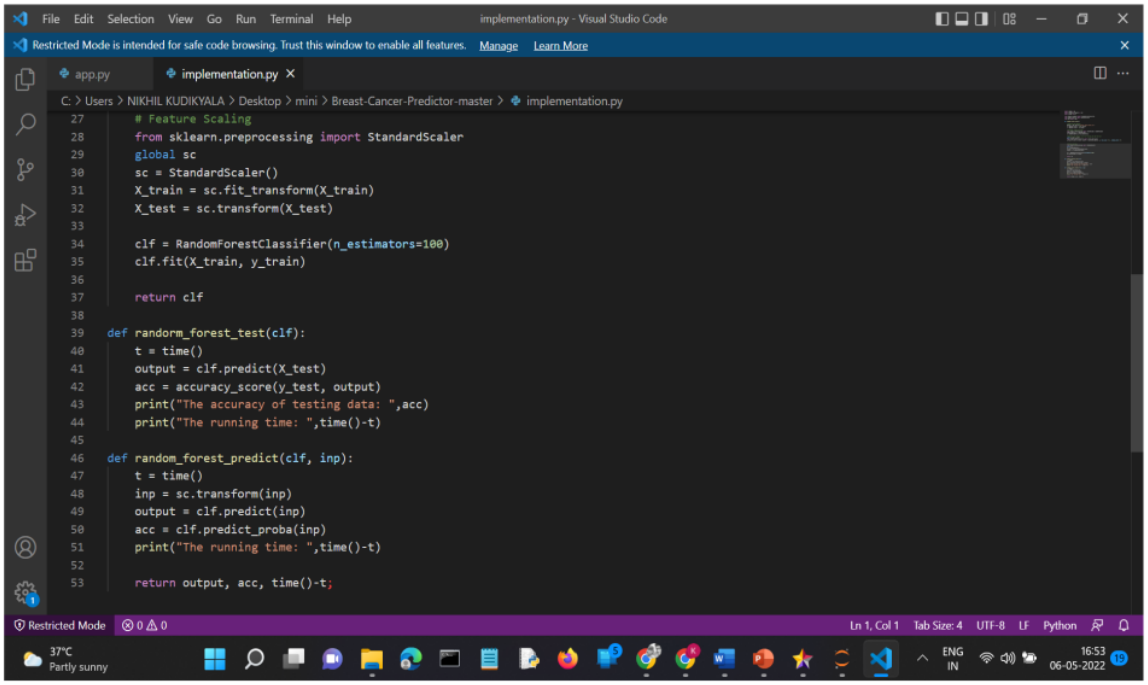

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
In [58]:
#show another way to get metrics of the models
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
for i in range(len(model)):
print('Model', i)
print(classification_report(Y_test, model[i].predict(X_test)))
print(accuracy_score(Y_test,model[i].predict(X_test)))
print()
Model 0
precision recall f1-score support
0 0.97 0.98 0.98 90
1 0.93 0.94 0.93 53
accuracy 0.95 143
macro avg 0.95 0.95 0.95 143
weighted avg 0.95 0.95 0.95 143
0.9510489510489511
Model 1
precision recall f1-score support
0 0.98 0.92 0.95 90
1 0.88 0.96 0.92 53
accuracy 0.94 143
macro avg 0.93 0.94 0.93 143
weighted avg 0.94 0.94 0.94 143
0.9370629370629371
Model 2
precision recall f1-score support
0 0.98 0.97 0.97 90
1 0.94 0.96 0.95 53
accuracy 0.97 143
macro avg 0.96 0.96 0.96 143
weighted avg 0.97 0.97 0.97 143
0.965034965034965
37°C
Partly sunny
ENG
IN
16:35
06-05-2022

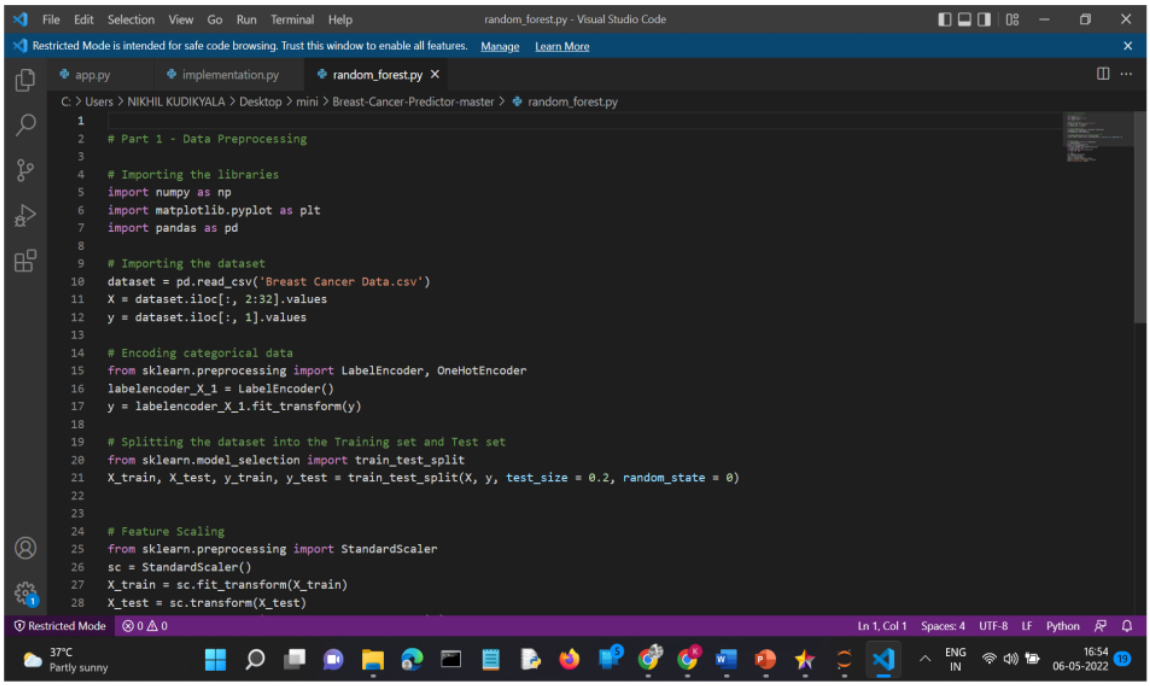

localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
0 0.98 0.97 0.97 90
1 0.94 0.96 0.95 53
accuracy 0.97 143
macro avg 0.96 0.96 0.96 143
weighted avg 0.97 0.97 0.97 143
0.965034965034965
In [39]:
#print the prediction of random foest classifier model
pred = model[2].predict(X_test)
print(pred)
print()
print(Y_test)
[1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 0
1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1]
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1
1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1]
37°C
Partly sunny
ENG
IN
16:36
06-05-2022

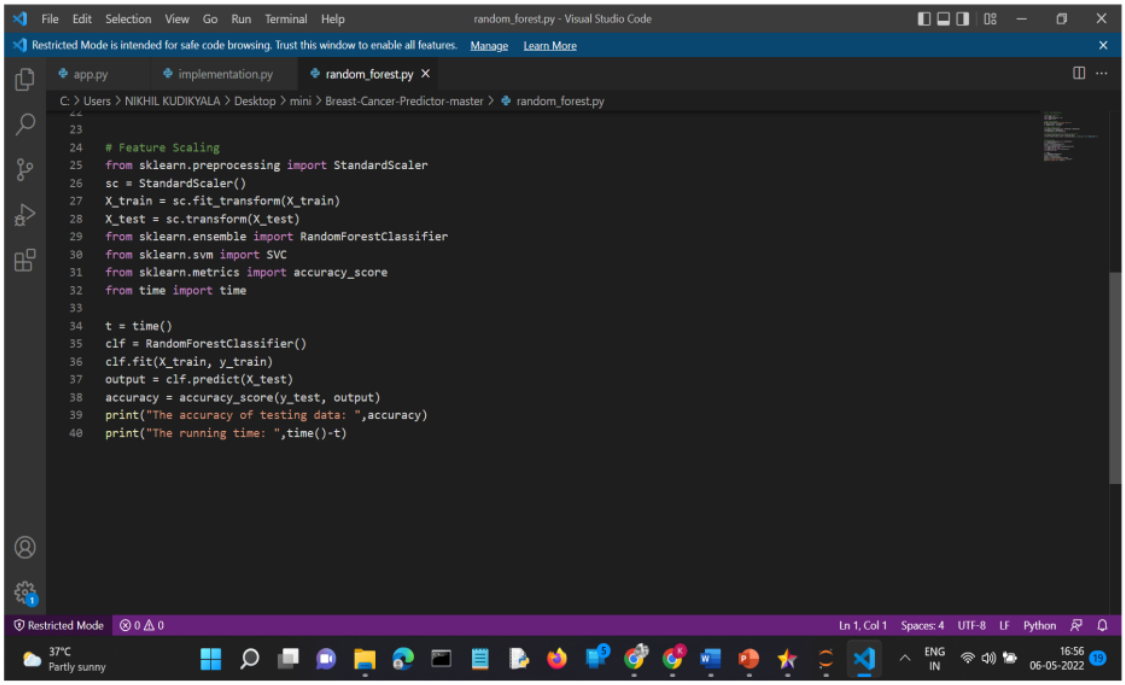

File Edit Selection View Go Run Terminal Help
app.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > app.py
from flask import Flask, render_template, request
from implementation import random_forest_test, random_forest_train, random_forest_predict
from sklearn.preprocessing import StandardScaler
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from random_forest import accuracy
from sklearn.metrics import accuracy_score
from time import time
app = Flask(__name__)
app.url_map.strict_slashes = False
@app.route('/')
def index():
return render_template('home.html')
@app.route('/predict', methods=['POST'])
def login_user():
data_points = list()
data = []
string = 'value'
for i in range(1,31):
data.append(float(request.form['value'+str(i)]))
for i in range(30):
Restricted Mode
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Python
37°C
Partly sunny
ENG
IN
16:43
06-05-2022

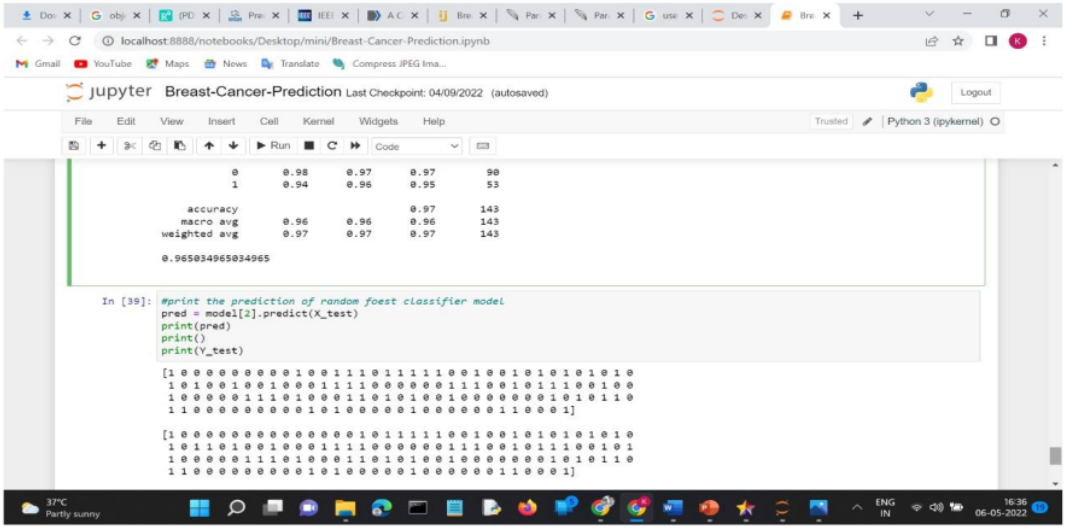

File Edit Selection View Go Run Terminal Help
app.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > app.py
data_points.append(data[i])
print(data_points)
data_np = np.asarray(data, dtype = float)
data_np = data_np.reshape(1,-1)
out, acc, t = random_forest_predict(clf, data_np)
if(out==1):
output = 'Malignant'
else:
output = 'Benign'
acc_x = acc[0][0]
acc_y = acc[0][1]
if(acc_x>acc_y):
acc1 = acc_x
else:
acc1=acc_y
return render_template('result.html', output=output, accuracy=accuracy, time=t)
if __name__=='__main__':
global clf
clf = random_forest_train()
random_forest_test(clf)
#print("Done")
Restricted Mode
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Python
37°C
Partly sunny
ENG
IN
16:46
06-05-2022


File Edit Selection View Go Run Terminal Help
implementation.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
implementation.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > implementation.py
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from time import time
def random_forest_train():
# Importing the dataset
dataset = pd.read_csv('Breast Cancer Data.csv')
X = dataset.iloc[:, 2:32].values
y = dataset.iloc[:, 1].values
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
y = labelencoder_X_1.fit_transform(y)
# Splitting the dataset into the Training set and Test set
global X_test, y_test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
Restricted Mode
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Python
37°C Partly sunny
ENG IN
16:53 06-05-2022


File Edit Selection View Go Run Terminal Help
implementation.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
implementation.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > implementation.py
# Feature Scaling
from sklearn.preprocessing import StandardScaler
global sc
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
clf = RandomForestClassifier(n_estimators=100)
clf.fit(X_train, y_train)
return clf
def randorm_forest_test(clf):
t = time()
output = clf.predict(X_test)
acc = accuracy_score(y_test, output)
print("The accuracy of testing data: ",acc)
print("The running time: ",time()-t)
def random_forest_predict(clf, inp):
t = time()
inp = sc.transform(inp)
output = clf.predict(inp)
acc = clf.predict_proba(inp)
print("The running time: ",time()-t)
return output, acc, time()-t;
Restricted Mode
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Python
37°C Partly sunny
ENG IN
16:53 06-05-2022


File Edit Selection View Go Run Terminal Help
random_forest.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
implementation.py
random_forest.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > random_forest.py
# Part 1 - Data Preprocessing
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('Breast Cancer Data.csv')
X = dataset.iloc[:, 2:32].values
y = dataset.iloc[:, 1].values
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
y = labelencoder_X_1.fit_transform(y)
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
Restricted Mode
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python
37°C
Partly sunny
ENG
IN
16:54
06-05-2022


File Edit Selection View Go Run Terminal Help
random_forest.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
app.py
implementation.py
random_forest.py
C: > Users > NIKHIL KUDIKYALA > Desktop > mini > Breast-Cancer-Predictor-master > random_forest.py
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from time import time
t = time()
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
output = clf.predict(X_test)
accuracy = accuracy_score(y_test, output)
print("The accuracy of testing data: ",accuracy)
print("The running time: ",time()-t)
Restricted Mode
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python
37°C
Partly sunny
ENG
IN
16:56
06-05-2022


localhost:8888/notebooks/Desktop/mini/Breast-Cancer-Prediction.ipynb
jupyter Breast-Cancer-Prediction Last Checkpoint: 04/09/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
Run Code
0 0.98 0.97 0.97 90
1 0.94 0.96 0.95 53
accuracy 0.97 143
macro avg 0.96 0.96 0.96 143
weighted avg 0.97 0.97 0.97 143
0.9650349650349651
In [39]:
#print the prediction of random foest classifier model
pred = model[2].predict(X_test)
print(pred)
print()
print(Y_test)
[1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 0
1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0
1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 0 1
1 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1]
37°C Partly sunny
ENG IN
16:36 06-05-2022

Code

HTMLCode

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Breast Cancer Prediction</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link href="https://fonts.googleapis.com/css2?family=Balsamiq+Sans&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
   /* css for back to top button */
 html {
  scroll-behavior: smooth;
 }
 a , a:hover
 {
  text-decoration: none;
 }
 #button {
  display: inline-block;
  background-color: #01CBC6;
  width: 52px;
  height: 52px;
  text-align: center;
  border-radius: 25px;
  position: fixed;
  bottom: 10px;
  right: 10px;
  transition: background-color .3s,
    opacity .5s, visibility .5s;
  opacity: 0;
  visibility: hidden;
  z-index: 1000;
 }
 #button::after {
  content: "\f077";
  font-family: FontAwesome;
  font-weight: normal;
  font-style: normal;
  font-size: 2em;
  line-height: 50px;
  color: #fff;
 }
 #button:hover {
  cursor: pointer;
  background-color: lightgreen;
```

```
}
#button:active {
  background-color: lightgreen;
}
#button.show {
  opacity: 1;
  visibility: visible;
}

/* Styles for the content section */


@media (min-width: 500px) {

  #button {
    margin: 30px;
  }
}
</style>
</head>
<body>
<!-- Back to top button -->
<a id="button"></a>


{% include "footer.html" %}

<div class="container">

  {% include "tumor.html" %}



<div style="margin: 40px;padding: 20px;font-size: 1.3em;background-color: #67E6DC">

<ul class="nav nav-tabs" style="background-color: #67E6DC">
<li class="active"><a data-toggle="tab" href="#home" style="font-family: 'Balsamiq Sans', cursive;">What is Breast Cancer</a></li>
<li><a data-toggle="tab" href="#menu1" style="font-family: 'Balsamiq Sans', cursive;">Who gets Breast Cancer</a></li>
<li><a data-toggle="tab" href="#menu2" style="font-family: 'Balsamiq Sans', cursive;">What Are the Symptoms of Breast Cancer</a></li>
</ul>

<div class="tab-content">
<div id="home" class="tab-pane fade in active">
<h3 style="font-family: 'Balsamiq Sans', cursive;">What is Breast Cancer</h3>
<p style="font-family: 'Balsamiq Sans', cursive;">Cancers are typically named after the part of the body from which they originate. Breast cancer originates in the breast tissue. Like other cancers, breast cancer can invade and grow into the tissue surrounding the breast. It can also travel to other parts of the body and form new tumors, a process called metastasis..</p>
</div>
```

```
<div id="menu1" class="tab-pane fade">
<h3 style="font-family: 'Balsamiq Sans', cursive;">Who gets Breast Cancer</h3>
<p style="font-family: 'Balsamiq Sans', cursive;">Breast cancer ranks second as a cause of cancer death in women (after lung cancer). Today, about 1 in 8 women (12%) will develop breast cancer in her lifetime. The American Cancer Society estimated that in 2017, about 252,710 women will be diagnosed with invasive breast cancer and about 40,610 will die from the disease.</p>
</div>
<div id="menu2" class="tab-pane fade">
<h3 style="font-family: 'Balsamiq Sans', cursive;">Some Symptoms of Breast Cancer include:</h3>
<p>
<ul style="font-family: 'Balsamiq Sans', cursive;">
<li>A mass or lump, which may feel as small as a pea.</li>
<li>A change in the size, shape, or contour of the breast.</li>
<li>A blood-stained or clear fluid discharge from the nipple.</li>
<li>Redness of the skin on the breast or nipple.</li>
<li>An area that is distinctly different from any other area on either breast.</li>
</ul>
</p>
</div>
</div>
</div>


<div style="background-color:#53E0BC; padding: 20px; margin: 40px">

<form class="form-horizontal" action="/predict" method="post">
<div class="form-group">
<label class="control-label col-sm-2" for="value1" style="font-family: 'Balsamiq Sans', cursive;">Radius Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value1" placeholder="Enter radius_mean" name="value1"  required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value2" style="font-family: 'Balsamiq Sans', cursive;">Texture Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value2" placeholder="Enter texture_mean" name="value2" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value3" style="font-family: 'Balsamiq Sans', cursive;">Perimeter Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value3" placeholder="Enter perimeter_mean" name="value3" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value4" style="font-family: 'Balsamiq Sans', cursive;">Area
```

```
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value4" placeholder="Enter area_mean" name="value4" required>
</div>
</div>


<div class="form-group">
<label class="control-label col-sm-2" for="value5" style="font-family: 'Balsamiq Sans', cursive;">Smoothness
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value5" placeholder="Enter smoothness_mean" name="value5"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value6" style="font-family: 'Balsamiq Sans', cursive;">Compactness
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value6" placeholder="Enter compactness_mean" name="value6"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value7" style="font-family: 'Balsamiq Sans', cursive;">Concavity
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value7" placeholder="Enter concavity_mean" name="value7"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value8" style="font-family: 'Balsamiq Sans', cursive;">Concave Points
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value8" placeholder="Enter concave points_mean" name="value8"
required>
</div>
</div>


<div class="form-group">
<label class="control-label col-sm-2" for="value9" style="font-family: 'Balsamiq Sans', cursive;">Symmetry
Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value9" placeholder="Enter symmetry_mean" name="value9"
required>
</div>
</div>
```

```
<div class="form-group">
<label class="control-label col-sm-2" for="value10" style="font-family: 'Balsamiq Sans', cursive;">Fractal
Dimension Mean</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value10" placeholder="Enter fractal_dimension_mean"
name="value10" required>
</div>
</div>



<div class="form-group">
<label class="control-label col-sm-2" for="value11" style="font-family: 'Balsamiq Sans', cursive;">Radius
SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value11" placeholder="Enter radius_se" name="value11" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value12" style="font-family: 'Balsamiq Sans', cursive;">Texture
SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value12" placeholder="Enter texture_se" name="value12" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value13" style="font-family: 'Balsamiq Sans', cursive;">Perimeter
SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value13" placeholder="Enter perimeter_se" name="value13"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value14" style="font-family: 'Balsamiq Sans', cursive;">Area
SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value14" placeholder="Enter area_se" name="value14" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value15" style="font-family: 'Balsamiq Sans', cursive;">Smoothness
SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value15" placeholder="Enter smoothness_se" name="value15"
required>
</div>
</div>
```

```
<div class="form-group">
<label class="control-label col-sm-2" for="value16" style="font-family: 'Balsamiq Sans', cursive;">Compactness SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value16" placeholder="Enter compactness_se" name="value16" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value17" style="font-family: 'Balsamiq Sans', cursive;">Concavity SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value17" placeholder="Enter concavity_se" name="value17" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value18" style="font-family: 'Balsamiq Sans', cursive;">Concave Points SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value18" placeholder="Enter concave points_se" name="value18" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value19" style="font-family: 'Balsamiq Sans', cursive;">Symmetry SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value19" placeholder="Enter symmetry_se" name="value19" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value20" style="font-family: 'Balsamiq Sans', cursive;">Fractal Dimension SE</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value20" placeholder="Enter fractal_dimension_se" name="value20" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value21" style="font-family: 'Balsamiq Sans', cursive;">Radius Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value21" placeholder="Enter radius_worst" name="value21" required>
</div>
```

```
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value22" style="font-family: 'Balsamiq Sans', cursive;">Texture
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value22" placeholder="Enter texture_worst" name="value22"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value23" style="font-family: 'Balsamiq Sans', cursive;">Perimeter
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value23" placeholder="Enter perimeter_worst" name="value23"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value24" style="font-family: 'Balsamiq Sans', cursive;">Area
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value24" placeholder="Enter area_worst" name="value24" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value25" style="font-family: 'Balsamiq Sans', cursive;">Smoothness
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value25" placeholder="Enter smoothness_worst" name="value25"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value26" style="font-family: 'Balsamiq Sans', cursive;">Compactness
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value26" placeholder="Enter compactness_worst" name="value26"
required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value27" style="font-family: 'Balsamiq Sans', cursive;">Concavity
Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value27" placeholder="Enter concavity_worst" name="value27"
required>
</div>
```

```
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value28" style="font-family: 'Balsamiq Sans', cursive;">Concave Points Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value28" placeholder="Enter concave points_worst" name="value28" required>
</div>
</div>


<div class="form-group">
<label class="control-label col-sm-2" for="value29" style="font-family: 'Balsamiq Sans', cursive;">Symmetry Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value29" placeholder="Enter symmetry_worst" name="value29" required>
</div>
</div>

<div class="form-group">
<label class="control-label col-sm-2" for="value30" style="font-family: 'Balsamiq Sans', cursive;">Fractal Dimension Worst</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="value30" placeholder="Enter fractal_dimension_worst" name="value30" required>
</div>
</div>
<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<input class="btn btn-primary" type="submit" name="Submit Request">
</div>
</div>
</form>

</div>

   {% include "copyrights.html" %}

</div>


</div>
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

<script>
     var btn = $('#button');

$(window).scroll(function() {
  if ($(window).scrollTop() > 300) {
    btn.addClass('show');
```

```
  } else {
    btn.removeClass('show');
  }
});

btn.on('click', function(e) {
  e.preventDefault();
  $('html, body').animate({scrollTop:0}, '300');
});


</script>
</body>
</html>
```

Tumor.html:

```
<div class="row">
<div class="col-sm-6 " style="background-color:#ccddff;padding: 15px">
<h2 style="text-align: center; font-family: 'Balsamiq Sans', cursive;">Benign Tumors</h2>
<p style="font-size: 1.2em; font-family: 'Balsamiq Sans', cursive;">
      Since benign tumors are harmless, doctors may opt to leave the lump alone rather than remove it. If you are experiencing discomfort, irritation or pressure, let your doctor know so that she can plan to remove it for you and improve your comfort.
</p>
</div>
<div class="col-sm-6" style="background-color:#ffccee;padding: 15px">
<h2 style="text-align: center; font-family: 'Balsamiq Sans', cursive;">Malignant tumors</h2>
<p style="font-size: 1.2em; font-family: 'Balsamiq Sans', cursive;">
      Malignant tumors are cancerous and aggressive because they invade and damage surrounding tissue. When a tumor is suspected to be malignant, the doctor will perform a biopsy to determine the severity or aggressiveness of the tumor.
</p>
</div>
</div>
```

Result.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Breast Cancer Prediction</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container-fluid" style="background-color: #fff">

 {% include "footer.html" %}

<div class="container">


<div class="jumbotron">
<h1>Cancer is Predicted to be {{ output }}</h1>
<p>The Probability of {{ output }} is {{ accuracy * 100 }} %.</p>
<p>The time taken to predict is {{ time }} s.</p>
</div>

   {% include "tumor.html" %}

   {% include "copyrights.html" %}

</div>


</div>

</body>
</html>
```

Footer.html

```
<div class="container-fluid" style="background-color: #fff">
<div class="jumbotron" style="background-color: #01CBC6">
<h1 align="center" style="margin-top:1px; font-family: 'Balsamiq Sans', cursive;">Breast Cancer Prediction</h1>
<p align="center" style="font-family: 'Balsamiq Sans', cursive;">Predicts whether the cancer is benign or malignant with an
accuracy of more than <b>95%</b>.</p>
</div>
```

Copyrights.html:

```
<div class="well" style="margin: 40px; background-color:lightgreen;text-align: center;">
   Developed by C15 BATCH HEROS
<br>
&copy All rights reserved | 2022
<br>
   Made in India with ❤❤❤
</div>
```

Pythoncode

app.py

```
from flask import Flask, render_template, request
from implementation import randorm_forest_test,
random_forest_train, random_forest_predict
from sklearn.preprocessing import StandardScaler
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from random_forest import accuracy
from sklearn.metrics import accuracy_score
from time import time



app = Flask(_name_)
app.url_map.strict_slashes = False


@app.route('/')
def index():
    return render_template('home.html')


@app.route('/predict', methods=['POST'])
def login_user():


    data_points = list()
    data = []
    string = 'value'
```

```
    for i in range(1,31):
        data.append(float(request.form['value'+str(i)]))

    for i in range(30):
        data_points.append(data[i])

    print(data_points)

    data_np = np.asarray(data, dtype = float)
    data_np = data_np.reshape(1,-1)
    out, acc, t = random_forest_predict(clf, data_np)

    if(out==1):
        output = 'Malignant'
    else:
        output = 'Benign'

    acc_x = acc[0][0]
    acc_y = acc[0][1]
    if(acc_x>acc_y):
        acc1 = acc_x
    else:
        acc1=acc_y
    return render_template('result.html', output=output,
accuracy=accuracy, time=t)
```

```
if _name=='main_':
    global clf
    clf = random_forest_train()
    randorm_forest_test(clf)
    #print("Done")
    app.run(debug=True)
```

Implementation.py:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from time import time


def random_forest_train():


    # Importing the dataset
    dataset = pd.read_csv('Breast Cancer Data.csv')
    X = dataset.iloc[:, 2:32].values
    y = dataset.iloc[:, 1].values


    # Encoding categorical data
```

```
    from sklearn.preprocessing import LabelEncoder, OneHotEncoder
    labelencoder_X_1 = LabelEncoder()
    y = labelencoder_X_1.fit_transform(y)

    # Splitting the dataset into the Training set and Test set
    global X_test, y_test
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)

    # Feature Scaling
    from sklearn.preprocessing import StandardScaler
    global sc
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)

    clf = RandomForestClassifier(n_estimators=100)
    clf.fit(X_train, y_train)

    return clf

def randorm_forest_test(clf):
    t = time()
    output = clf.predict(X_test)
```

```
    acc = accuracy_score(y_test, output)
    print("The accuracy of testing data: ",acc)
    print("The running time: ",time()-t)

def random_forest_predict(clf, inp):
    t = time()
    inp = sc.transform(inp)
    output = clf.predict(inp)
    acc = clf.predict_proba(inp)
    print("The running time: ",time()-t)

    return output, acc, time()-t;
```

RandomForest.py:

```
# Part 1 - Data Preprocessing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Breast Cancer Data.csv')
X = dataset.iloc[:, 2:32].values
y = dataset.iloc[:, 1].values
```

```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
y = labelencoder_X_1.fit_transform(y)


# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from time import time
t = time()
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
output = clf.predict(X_test)
accuracy = accuracy_score(y_test, output)
print("The accuracy of testing data: ",accuracy)
print("The running time: ",time()-t)
```

C15_Batch__original.docx

ORIGINALITY REPORT

42%	28%	20%	25%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	datainflow.com Internet Source	6%
2	www.webmd.com Internet Source	2%
3	Submitted to CSU, San Marcos Student Paper	2%
4	indiaai.gov.in Internet Source	2%
5	Khadija El Bouchefry, Rafael S. de Souza. "Learning in Big Data: Introduction to Machine Learning", Elsevier BV, 2020 Publication	2%
6	www.coursehero.com Internet Source	2%
7	Submitted to Gates-Chili High School Student Paper	2%
8	Naresh Khuriwal, Nidhi Mishra. "Breast cancer diagnosis using adaptive voting ensemble	2%

machine learning algorithm", 2018 IEEMA Engineer Infinite Conference (eTechNxT), 2018
Publication

9	stackoverflow.com Internet Source	1 %
10	innovate.mygov.in Internet Source	1 %
11	Vikrant A. Dev, Mario R. Eden. "Gradient Boosted Decision Trees for Lithology Classification", Elsevier BV, 2019 Publication	1 %
12	Submitted to SASTRA University Student Paper	1 %
13	www.teknowebapp.com Internet Source	1 %
14	lab.anahuac.mx Internet Source	1 %
15	Submitted to Gitam University Student Paper	1 %
16	John Muschelli, Joshua Betz, Ravi Varadhan. "Binomial Regression in R", Elsevier BV, 2014 Publication	1 %
17	Hoss Belyadi, Alireza Haghighat. "Supervised learning", Elsevier BV, 2021 Publication	1 %

18	Victor Chang, Vallabhanent Rupa Bhavani, Ariel Qianwen Xu, MA Hossain. "An artificial intelligence model for heart disease detection using machine learning algorithms", Healthcare Analytics, 2022 Publication	1 %
19	Submitted to Michigan Technological University Student Paper	1 %
20	breastcancer.about.com Internet Source	1 %
21	Submitted to The NorthCap University, Gurugram Student Paper	1 %
22	www.eurekaselect.com Internet Source	1 %
23	Submitted to University of Wales Institute, Cardiff Student Paper	1 %
24	umpir.ump.edu.my Internet Source	1 %
25	github.com Internet Source	<1 %
26	A K Dalai, A K Jena, B V Ramana, B Maneesha, Nibedan Panda. "Supervised Machine Learning Approaches for Medical Data	<1 %

Classification", 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), 2022
Publication

27	Vijay Kotu, Bala Deshpande. "Classification", Elsevier BV, 2019 Publication	<1 %
28	Submitted to Universita del Piemonte Orientale Student Paper	<1 %
29	Submitted to University of Wolverhampton Student Paper	<1 %
30	Submitted to University of Essex Student Paper	<1 %
31	onlinewebtutorblog.com Internet Source	<1 %
32	Submitted to Glasgow Caledonian University Student Paper	<1 %
33	github.coventry.ac.uk Internet Source	<1 %
34	Arun Solanki, Deepak Kumar Jain. "Emerging Trends and Applications in Cognitive Computing", Recent Advances in Computer Science and Communications, 2020 Publication	<1 %
35	learn.theprogrammingfoundation.org	

	Internet Source	<1%
36	Submitted to University of New Haven Student Paper	<1%
37	grietinfo.in Internet Source	<1%
38	Deepak Pareta, Indukuri Nishat Verma, Bhanu Prakash Lohani, Pradeep Kumar Kushwaha, Vimal Bibhu. "IoT Enabled Smart and Efficient Musical Water Fountain", 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022 Publication	<1%
39	Submitted to University of Sunderland Student Paper	<1%
40	nsing.tistory.com Internet Source	<1%
41	www.programcreek.com Internet Source	<1%
42	Submitted to University of Ulster Student Paper	<1%
43	Vijay Kotu, Bala Deshpande. "Classification", Elsevier BV, 2015 Publication	<1%

44	docshare.tips Internet Source	<1%
45	www.slideshare.net Internet Source	<1%
46	www.tutorialspoint.com Internet Source	<1%
47	gitlab.fbk.eu Internet Source	<1%
48	dalspace.library.dal.ca Internet Source	<1%
49	gist.github.com Internet Source	<1%
50	www.groundai.com Internet Source	<1%
51	"Predicting Breast Cancer using Modern Data Science Methodology", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	<1%
52	ebin.pub Internet Source	<1%
53	medium.com Internet Source	<1%
54	www.planetpython.org Internet Source	<1%

55	Ram MurtiRawat, Shivam Panchal, Vivek Kumar Singh, Yash Panchal. "Breast Cancer Detection Using K-Nearest Neighbors, Logistic Regression and Ensemble Learning", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020 Publication	<1 %
56	Yifu Han, Siddharth Misra, Yuteng Jin. "Multifrequency electromagnetic data acquisition and interpretation in the laboratory and in the subsurface", Elsevier BV, 2021 Publication	<1 %
57	advising.sunysb.edu Internet Source	<1 %
58	muhammadbilalyar.github.io Internet Source	<1 %
59	origin.geeksforgeeks.org Internet Source	<1 %
60	tiaonmmn.github.io Internet Source	<1 %
61	Ekaba Bisong. "Building Machine Learning and Deep Learning Models on Google Cloud Platform", Springer Science and Business Media LLC, 2019 Publication	<1 %

62	Jesper Jansson, Andrzej Lingas, Ramesh Rajaby, Wing-Kin Sung. "Determining the Consistency of Resolved Triplets and Fan Triplets", Journal of Computational Biology, 2018 Publication	<1 %
63	chouette.beta.pole-emploi.fr Internet Source	<1 %
64	arun-aiml.blogspot.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	Off		