BREAST CANCER DETECTION MODEL

A MINI PROJECT REPORT

18CSC305J – ARTIFICIAL INTELLIGENCE

Submitted by

Mrunal Bhonde (RA2011003010142)

Vinay Sonkusale (RA2011003010167)

Rishit Shivesh(RA2011003010200)

Under the guidance of

M Rajalakshmi , Assistant Professor, Department of Computer Science and Engineering

In a partial fulfillment for the award of the degreeof

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that the mini project report titled "**Breast Cancer detection Model**" is the bonafide work of **Mrunal Bhonde (RA2011003010142), Vinay Sonkusale(RA2011003010167), Rishit Shivesh (RA2011003010200)** who carried out the mini project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion by any other candidate.

SIGNATURE

M Rajalakshmi

Assistant Professor

Department of Computing Technologies

SIGNATURE

Dr. M. Pushpalatha

HEAD OF THE DEPARTMENT

Professor & Head

Department of Computing Technologies

ABSTRACT

Breast cancer is the most common cancer among women worldwide, and early detection is crucial for improving survival rates. Machine learning techniques have emerged as effective tools for analyzing complex medical data and detectingbreast cancer. This paper presents an abstract on breast cancer detection using supervised machine learning models such as logistic regression, decision tree, and random forest.

The dataset used in this project contains various features such as age, tumor size, tumor grade, and histology. The first step in the analysis is data pre-processing, which involves handling missing values, scaling the data, and encoding categorical variables. After pre-processing, the dataset is split into training and testing sets to evaluate the performance of the models.

Logistic regression is a linear model that predicts the probability of an event occurring. It is a simple and fast algorithm that is commonly used for binary classification problems. Decision trees, on the other hand, are non-parametricmodels that can handle both categorical and continuous data. They are easy tointerpret and can capture complex relationships between features. Random forest is an ensemble method that combines multiple decision trees to improvethe predictive performance of the model.

The models are trained on the training set and evaluated on the testing set using various performance metrics such as accuracy, precision, recall, and F1-score. The results show that all the models achieved high accuracy, with random forest performing the best. The models also provide feature importance rankings, which can help identify the most important features in the dataset.

In conclusion, supervised machine learning models such as logistic regression, decision tree, and random forest can effectively detect breast cancer and provide valuable insights into the most important features. These models have the potential to improve early detection and contribute to better treatment outcomesfor breast cancer patients.

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
1.	Abstract	3
2.	Table of contents	4
3.	List of figures	5
4.	Abbreviations	6
5.	Introduction	7
6.	Literary survey	8
7.	System architecture and design ● Proposed architecture ● Models explanation	10
8.	Source code and testing	14
9.	Methodology/ Code explanation	15
10	Output/ Result Screenshots	21
11	Conclusion and Future enhancements	30
12	References	32

List of Figures

- System Diagram Fig 7.1
- SNS Plot Fig 21.1, Fig 21.2
- Heat Map Fig 21.3, Fig 21.4

Abbreviations

RF : Random Forest
B cell : Benign cell
M cell : Malignant cell
DT : Decision Tree
LR : Logistic Regression

INTRODUCTION

According to World health organization, Breast cancer is the most frequent cancer among women and it is the second dangerous cancer after lung cancer. In 2018, from the research it is estimated that total 627,000 women lost their life due to breast cancer that is 15% of all cancer deaths among women. In caseof any symptom, people visit to oncologist. Doctors can easily identify breast cancer by using Breast ultrasound, Diagnostic mammogram, Magnetic resonance imaging (MRI), Biopsy. Based on these test results, doctor may recommend further tests or therapy. Early detection is very crucial in
breast cancer. If chances of cancer are predicted at early stage then survivability chances of patient may increase. An alternate way to identify breast cancer is using machine learning algorithms for prediction of abnormaltumor. Thus, the research is carried out for the proper diagnosis and categorization of patients into malignant and benign groups.

Three types of tumors are as follows:-

a. **Benign tumors** are not cancerous they cannot spread or they can grow very slowly. And if doctors remove them, then they cannot cause any harm to the humanbody.
b. In **Premalignant tumor**s the cells are not cancerous but they havepotential to become malignant.
c. **Malignant cells** are cancerous and they can spread rapidly in body.

In machine learning, cancer classification can be done using benign or malignant cells could significantly improve our ability for prognosis in cancer patients, poor progress has been made for their application in the clinics. However, before gene expression profiling can be used in clinical practice, studies with larger data samples and more adequate validation are needed. And thus, our aim is to develop a prediction systemthat can predict chances of breast cancer on a huge data.

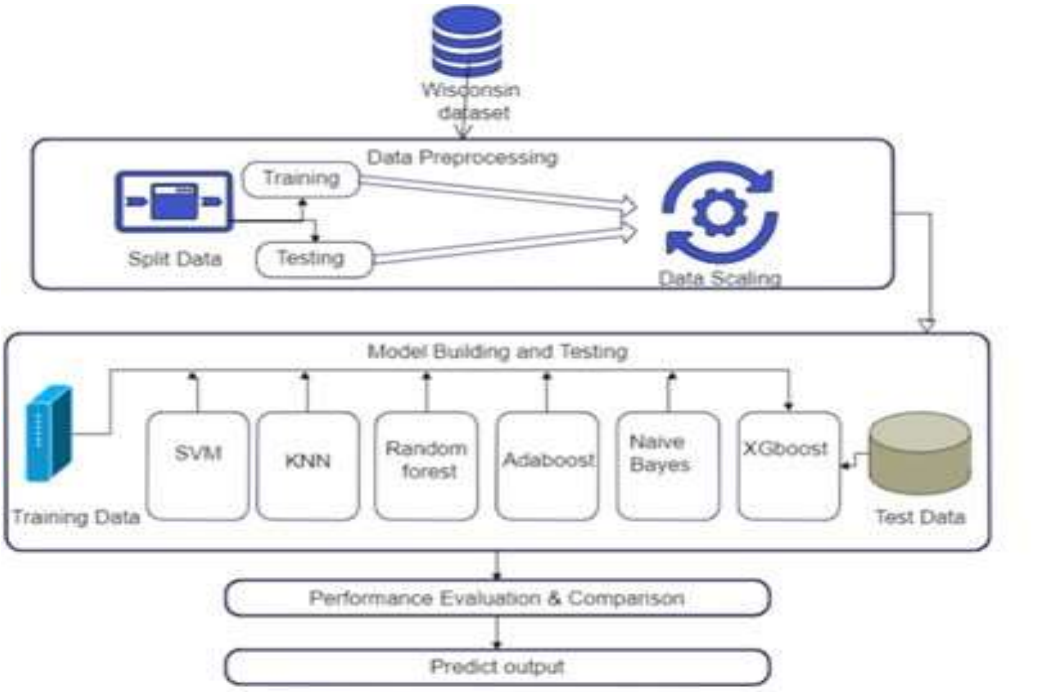
The dataset used in this project is from Breast Cancer Wisconsin (Diagnostic) Data Set (https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data)

LITERARY SURVEY

This project provides review of the present day studies being accomplished on breast cancer and the usage of various data mining techniques to predict and diagnoise the breast cancer. Presently, most of the physicians opt to make surgical biopsy in order to figure out different kinds of cancers for benign breast tumors from malignant). As biopsy could be very crucial challenge maximum of them believed that it should be stopped as much as possible. Thus, to recognize the kind of cancer and keep away from needless surgical biopsy, a smart system was presented which can be beneficial for both patients and physicians [3]. Subsequently, Vikas Chaurasia et al. carried out easy RBF, Logistic and REP Tree for prognosis of breast cancer [4]. To predict breast cancer comparative analysis was done by using neural network, decision tree, genetic algorithm and logistic regression by Wei-pin Chang et al. [5]. Their experimental outcomes found out that, amongst those applied techniques for predicting breast cancers lowest prediction accuracy was obtained by decision tree model and better accuracy rate was obtained by logistic regression model. In addition to this, genetic algorithm achieved maximum accuracy by generating standard classification rules inside the class of breast cancers. By making use of specific classification techniques for diagnosis of breast cancers. Shweta Kharya observed from a comprehensive survey and claimed that, decision tree yielded excessive accuracy rate and is the first-class predictor among the involved techniques and the Bayesian network, a well-known technique that's used in medical world Logistic regression, Decision Tree, Naïve Bayes, neural networks, multi-layer perceptron, and support Vector machine are some algorithms that were applied to diagnoise breast cancers by Senturk et al. Their experimental outcomes stated that,support Vector machine obtained high classification accuracy compared to other classifiers. Using C4.5 algorithm patients were Categorized into either "Carcinoma in situ" or "Malignant potential" group and confirmed that to diagnoise breast cancers, C4.5 obtained high accuracy by Rajesh et al. Observations were made from a survey by applying the numerous techniques which can also be utilized by several investigators to diagnoise breast cancers by Gupta and et al. Sooner or later they noted that, the optimal technique which can yield high rate of accuracy can be determined after developing numerous varieties of models using various algorithms on each instance, by trying distinctive techniques or algorithms. The

most prominent methods used in mining of data are Classification and Prediction. supervised learning is performed for classification tasks. Some portion of data called is taken as training set comprising of instances. Further each instance comprises collection of features and further these features describes one single entity known as as class. The foremost objective of classification technique is to generate a model with proficiency of forecasting the class label as accurate as feasible in earlier hidden records. Furthermore, to predict the correctness or accuracy of the created model, a test set is used [10]. Classifying tumor cells, studying the effectiveness of remedies are some applications of classification in medical diagnosis. Numerous algorithms for classification are proposed such as Decision Tree Induction, Rule-Based Methods, k-Nearest-Neighbor which are Memory-Based Methods, Genetic Programming [14,15]. Grounded on the other attributes values, Regression similar to classification tries to predict the value of an attribute [16]. Separating the data set into two continuous variables and associating them to discover the quality of attributes is the main aim of regression. To simply state regression predicts the quality of one specific attribute grounded on the value of other attribute thereby resulting in feature selection with best quality. Regression can be applied in various medical diagnosis and cancer diagnosis or prognosis

PROPOSED ARCHITECTURE



Role Of Machine Learning In Detection Of Breast Cancer

A mammogram is an x-ray picture of the breast. It can be used to check for breast cancer in women who have no signs or symptoms of the disease. It canalso be used if you have a lump or other sign of breast cancer.

Screening mammography is the type of mammogram that checks you when you have no symptoms. It can help reduce the number of deaths from breast cancer among women ages 40 to 70. But it can also have drawbacks. Mammograms cansometimes find something that looks abnormal but isn't cancer. This leads to further testing and can cause you anxiety. Sometimes mammograms can miss cancer when it is there. It also exposes you to radiation. You should talk to your

doctor about the benefits and drawbacks of mammograms. Together, you candecide when to start and how often to have a mammogram.

Now while its difficult to figure out for physicians by seeing only images of x-raythat weather the tumor is toxic or not training a machine learning model according to the identification of tumour can be of great help.

Breast cancer detection is a widely researched area in machine learning, andthere are several approaches that can be used to build a predictive model.

One possible architecture for a breast cancer detection project using supervisedlearning algorithms like logistic regression, decision tree, and random forest could include the following steps:

- Data Collection and Preprocessing: The first step in any machine learning project is to collect and preprocess the data. For breast cancerdetection, the data could include features such as age, family history ofcancer, mammogram results, and biopsy reports. The data should be cleaned, normalized, and transformed as needed to ensure that it is suitable for use with the chosen algorithms.

- Feature Selection: Once the data is preprocessed, the next step is to selectthe most relevant features to be used for building the model. This can be done using techniques like correlation analysis and feature importance ranking.

- Model Selection: With the features selected, the next step is to choose the appropriate algorithm(s) to use for building the predictive model. For breast cancer detection, common algorithms include logistic regression, decision trees, and random forests. Each algorithm has its strengths and weaknesses, so it's important to experiment with different algorithms andhyperparameters to find the best approach.

- Model Training: After selecting the algorithm(s), the next step is to train the model using the preprocessed data. This involves splitting the data intotraining and testing sets and using the training set to fit the model.

- Model Evaluation: Once the model is trained, it's important to evaluate its performance using metrics like accuracy, precision, recall, andF1 score. A confusion matrix can also be used to visualize the performance of the model.

- Model Tuning: Based on the evaluation results, the model mayneed to be fine-tuned to improve its performance. This could involve adjusting hyperparameters, using different feature selection techniques, or trying out different algorithms.

- Model Deployment: Once the final model has been developed and evaluated, it can be deployed for use in breast cancer detection. This could involve creating a web or mobile application or integrating the model into an existing healthcaresystem.

Overall, building a breast cancer detection model using supervised learning algorithms like logistic regression, decision tree, and randomforest can be a complex process, but by following these steps, it's possible to develop an accurate and reliable predictive model that canbe used to improve patient outcomes.

EXPLANATION OF VARIOUS MACHINE LEARNING MODELS USED IN BREAST CANCER DETECTION

SUPERVISED MACHINE LEARNING MODELS

Supervised Learning is a type of system in which both input and desired output data are provided. Input and output data are labelled forclassification to provide a learning basis for future data processing.
Supervised systems provide the learning algorithms with known quantitiesto support future judgments.

1. Decision Tree

Decision tree is the most powerful and popular tool for classification and prediction in machine learning.The Decision trees algorithm consists of two parts: nodes and rules (tests).A Decision tree is like tree structure, where node denotes a test on an attribute, Branch represents an outcome of the test, and each leaf node holds a class label. For detecting breast cancer, its leaf nodes are categorised as benign and malignant. And then certain rules are established to check tumor is benign or malignant.

2. Random Forest Classification

Random forest algorithm is a supervised classification algorithm. In this classifier, the higher the number of trees in the forest gives the highaccuracy results.

a. Random forest algorithm can use for both classification and the regression task and can handle the missing values too. For this dataset, we have already handled missing values of attributes. Random forest classifier doesn't over fits the model, if it includesmany trees.

b. It can also work on categorical values too. In this case, we had categorical data as B & M representing benign & malignant which is further converted to numeric data as 0 & 1 respectively.

3. Logistic Regression:

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independentvariables. It is generally used when the dependent
variable is binary and provides a constant output

Source Code

```
#Description: This program detects breast cancer, based offof data.


# importing libraries
import numpy
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns



# reading data from the file
df=pd.read_csv("data.csv")

df.head()
df.info()

# return all the columns with null values count
df.isna().sum()

# return the size of dataset
df.shape

# remove the column
df=df.dropna(axis=1)

# shape of dataset after removing the null column
df.shape

# describe the dataset
df.describe()

# Get the count of malignant<M> and Benign<B> cells
df['diagnosis'].value_counts()
# label encoding(convert the value of M and B into 1 and 0)
from sklearn.preprocessing import LabelEncoder
```

```
labelencoder_Y = LabelEncoder() df.iloc[:,1]=labelencoder_Y.fit_transform(df.iloc[:,1].values)

df.head()

sns.pairplot(df.iloc[:,1:5],hue="diagnosis")# get

the correlation
df.iloc[:,1:32].corr()

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

# split the dataset into dependent(X) and Independent(Y) datasets
X=df.iloc[:,2:31].values
Y=df.iloc[:,1].values
```

In [18]:

```
# spliting the data into trainning and test dateset
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.20,random_state=0)
```

In [19]:

```
# feature scaling
from sklearn.preprocessing import StandardScaler
X_train=StandardScaler().fit_transform(X_train)
X_test=StandardScaler().fit_transform(X_test)


# models/ Algorithms

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(random_state=0,criterion="entropy")tree.fit(X_train,Y_train)

    #Random Forest
    from sklearn.ensemble import RandomForestClassifier

  forest=RandomForestClassifier(random_state=0,criterion="entropy",n_estimators=10)
    forest.fit(X_train,Y_train)

    print('[0]logistic regression accuracy:',log.score(X_train,Y_train))
    print('[1]Decision tree accuracy:',tree.score(X_train,Y_train))
    print('[2]Random forest accuracy:',forest.score(X_train,Y_train))

    return log,tree,forest


model=models(X_train,Y_train)
# testing the models/result

from sklearn.metrics import accuracy_score from
sklearn.metrics import classification_report

for i in range(len(model)):
   print("Model",i)
   print(classification_report(Y_test,model[i].predict(X_test)))
   print('Accuracy : ',accuracy_score(Y_test,model[i].predict(X_test)))


# prediction of random-forest
pred=model[2].predict(X_test)
print('Predicted values:')
print(pred)
print('Actual values:')
print(Y_test)

from joblib import dump
dump(model[2],"Cancer_prediction.joblib")
```

EXPLANATION OF CODE

- START PROGRAMMING

The first thing before writing a single line of code is to put in a description in comments of what the code does. This way anyone can look back on our code and know exactly what it does.

- import the packages/libraries to make it easier to write the program. Next we will load the data, and print the first 7 rows of data.

NOTE: Each row of data represents a patient that may or may not have cancer.

- Explore the data and count the number of rows and columns in the data set. Their are 569 rows of data which means their are 569 patients in this data set, and 33 columns which mean their are 33 features or data points for each patient.

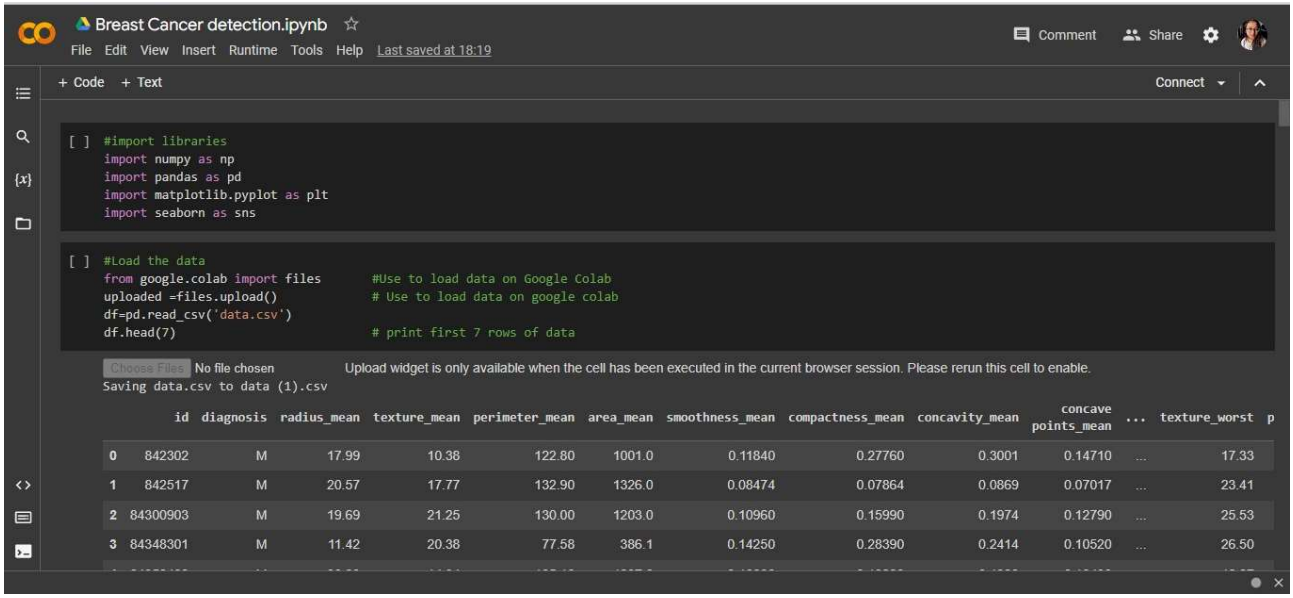
- Continue exploring the data and get a count of all of the columns that contain empty (NaN, NAN, na) values. Notice none of the columns contain any empty values except the column named ‘Unnamed: 32’ , which contains 569 empty values (the same number of rows in the data set, this tells me this column is completely useless)
- Remove the column ‘Unnamed: 32’ from the original data set sinceit adds no value.
- Get the new count of the number of rows and columns.
- Get a count of the number of patients with Malignant (M) cancerous and Benign (B) non-cancerous cells.
- Look at the data types to see which columns need to be transformed / encoded. We can see from the data types that all ofthe columns/features are numbers except for the column ‘diagnosis’, which is categorical data represented as an object in python.
- Encode the categorical data. Change the values in the column ‘diagnosis’ from M and B to 1 and 0 respectively, then print the

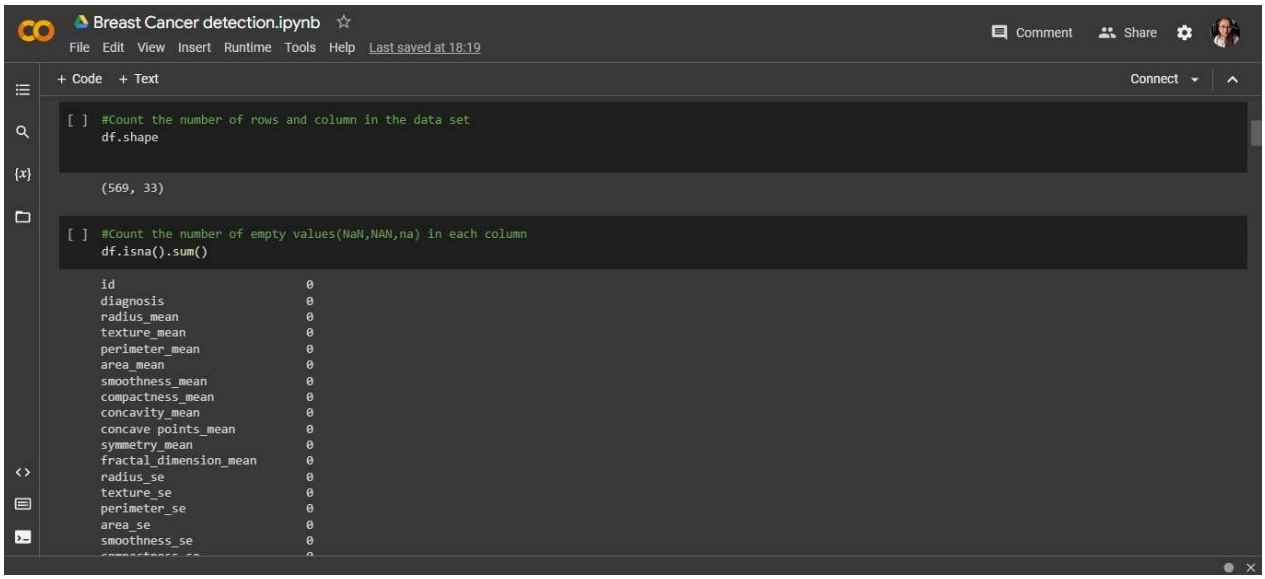
Results.

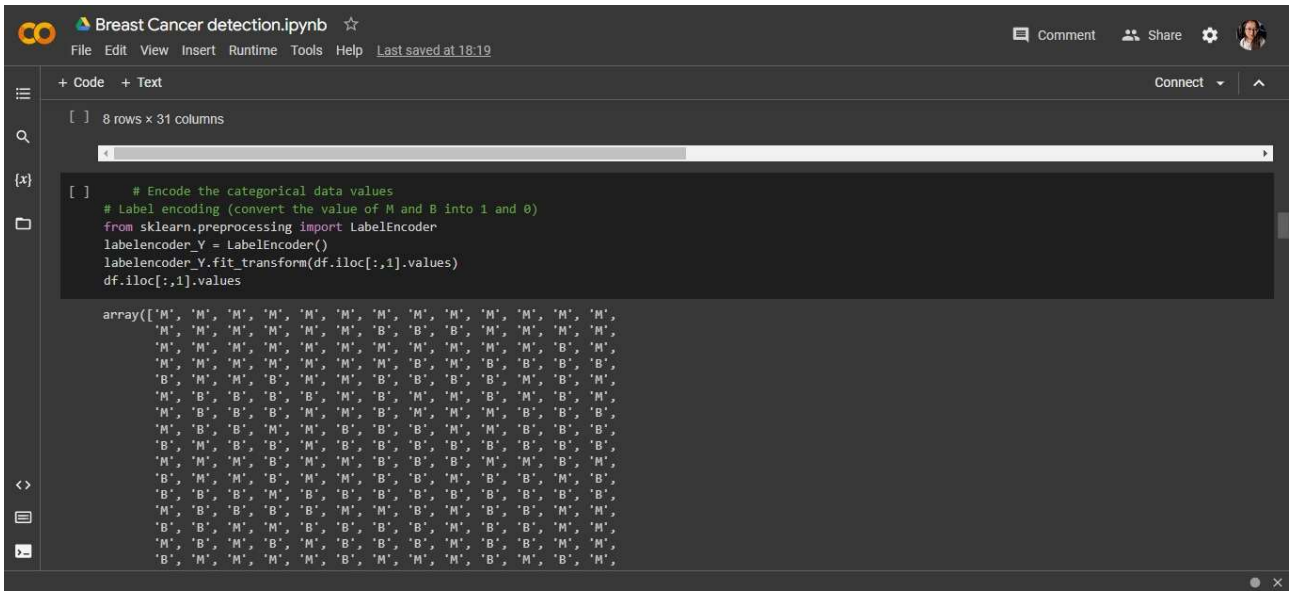
- Create a pair plot. A “pairs plot” is also known as a scatter plot, in which one variable in the same data row is matched with another variable’s value.
- Print the new data set which now has only 32 columns. Print only the first 5 rows.
- Get the correlation of the columns.
- Visualize the correlation by creating a heat map.
- Now We are done exploring and cleaning the data.
- Now We have to set up our data for the model by first splitting the data set into a feature data set also known as the independent data set (X), and a target data set also known as the dependent data set (Y).
- Split the data again, but this time into 75% training and 25% testing data sets.
- Scale the data to bring all features to the same level of magnitude, which means the feature / independent data will be within a specific range for example 0–100 or 0–1.

- Create a function to hold many different models (e.g. Logistic Regression, Decision Tree Classifier, Random Forest Classifier) to make the classification. These are the models that will detect if a patient has cancer or not. Within this function I will also printthe accuracy of each model on the training data.
- Create the model that contains all of the models, and look at the accuracy score on the training data for each model to classify if a patient has cancer or not.

OUTPUT SCREENSHOTS






Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] 8 rows × 31 columns
[ ] # Encode the categorical data values
# Label encoding (convert the value of M and B into 1 and 0)
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
labelencoder_Y.fit_transform(df.iloc[:,1].values)
df.iloc[:,1].values
array(['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'B', 'B', 'B', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'B', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'B', 'M', 'B', 'B', 'B', 'B',
'B', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'M',
'M', 'B', 'B', 'B', 'B', 'M', 'B', 'M', 'M', 'B', 'M', 'B', 'M',
'M', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'M', 'M', 'B', 'B', 'B',
'M', 'B', 'B', 'M', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'B', 'B',
'B', 'M', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'M', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'B', 'M', 'M', 'B', 'M',
'B', 'M', 'M', 'B', 'M', 'M', 'B', 'B', 'M', 'B', 'B', 'M', 'B',
'B', 'B', 'B', 'M', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B',
'M', 'B', 'B', 'B', 'B', 'M', 'M', 'B', 'M', 'B', 'B', 'M', 'M',
'B', 'B', 'M', 'M', 'B', 'B', 'B', 'B', 'M', 'B', 'B', 'M', 'M',
'M', 'B', 'M', 'B', 'M', 'B', 'B', 'B', 'M', 'B', 'B', 'M', 'M',
'B', 'M', 'M', 'M', 'M', 'B', 'M', 'M', 'M', 'B', 'M', 'B', 'M',


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] # Create a pair plot
sns.pairplot(df.iloc[:,1:5],hue="diagnosis")
<seaborn.axisgrid.PairGrid at 0x7ff9f0658520>
radius_mean
diagnosis

Fig 21.1

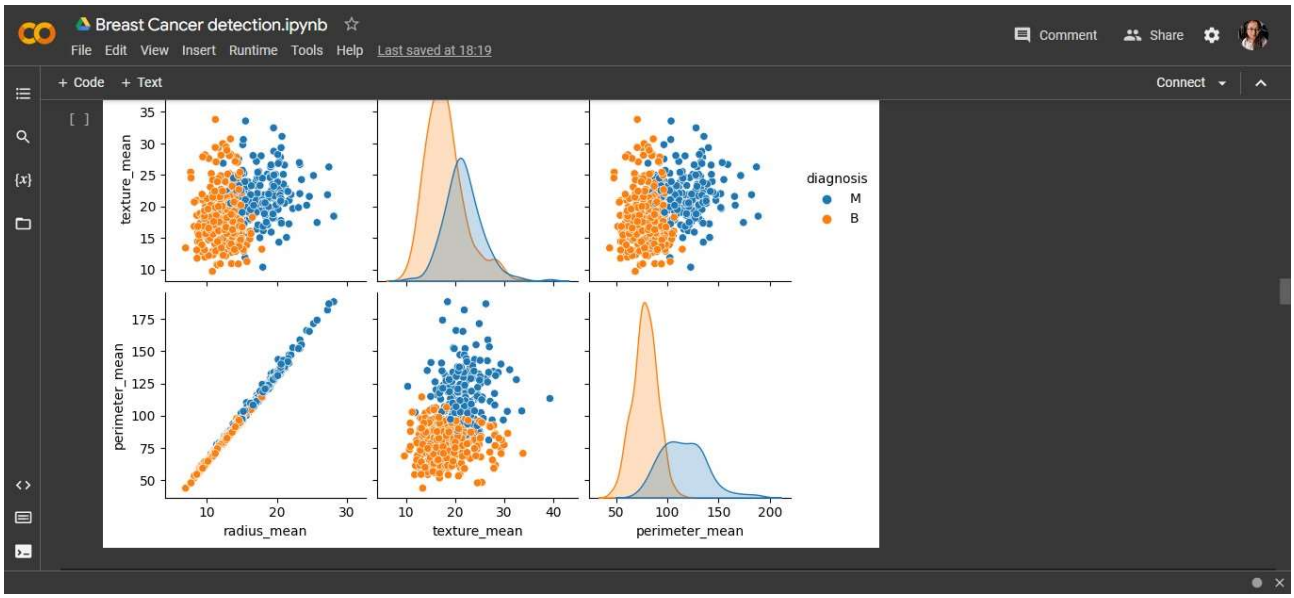
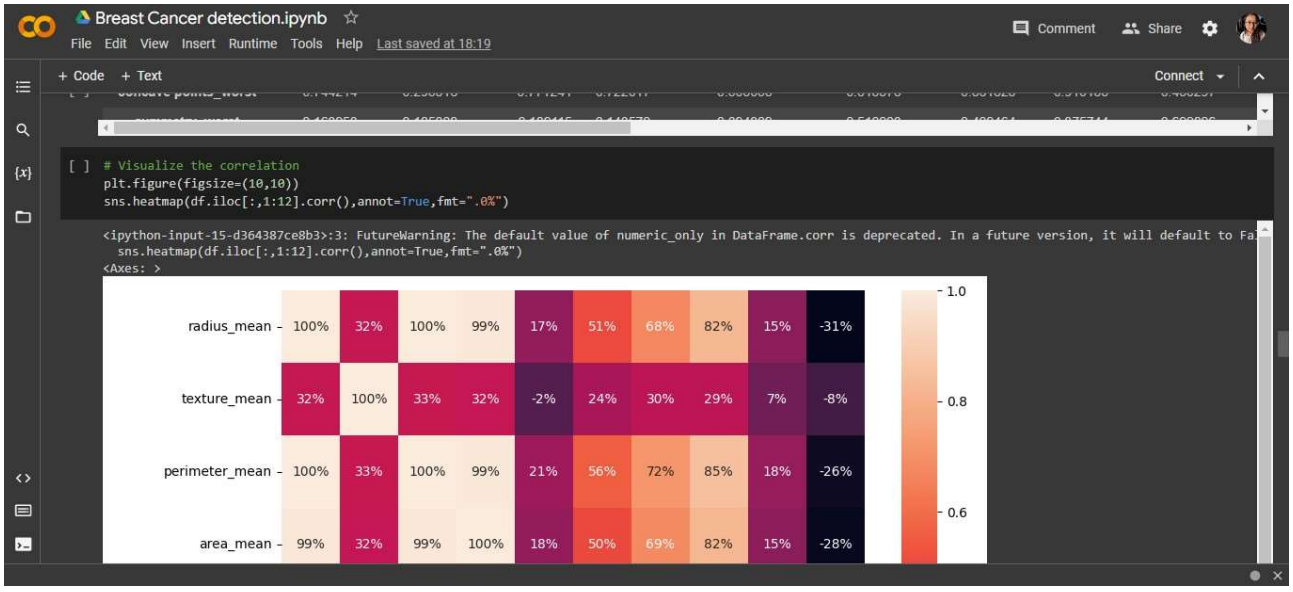


Fig 21.2

Fig 21.3



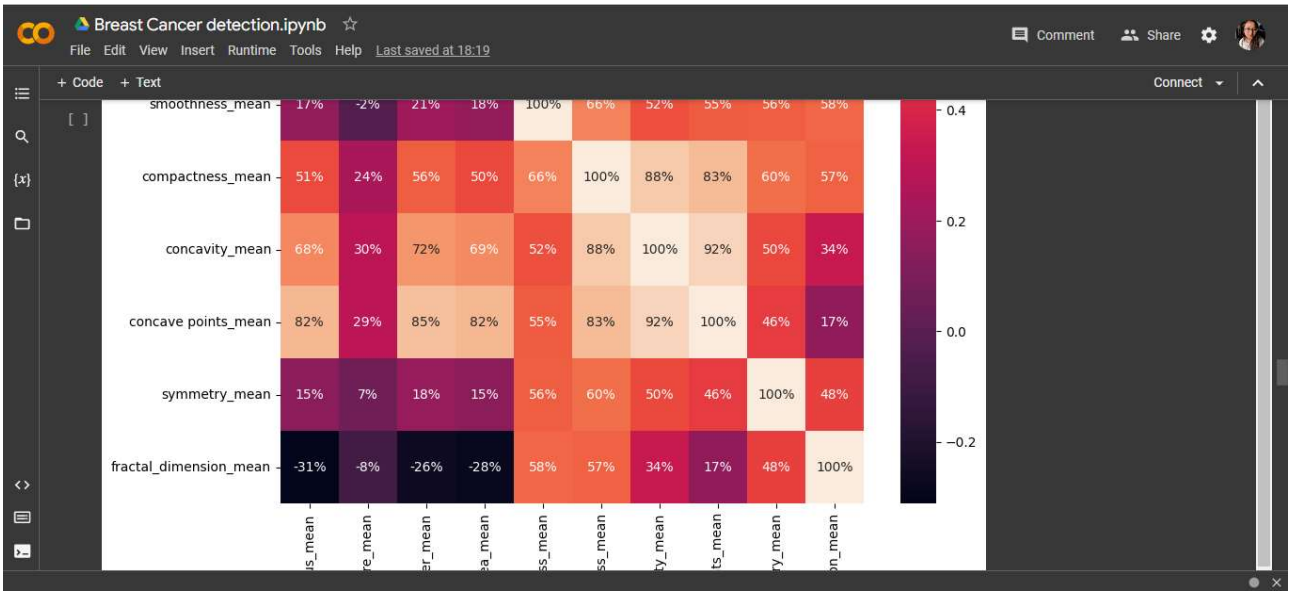
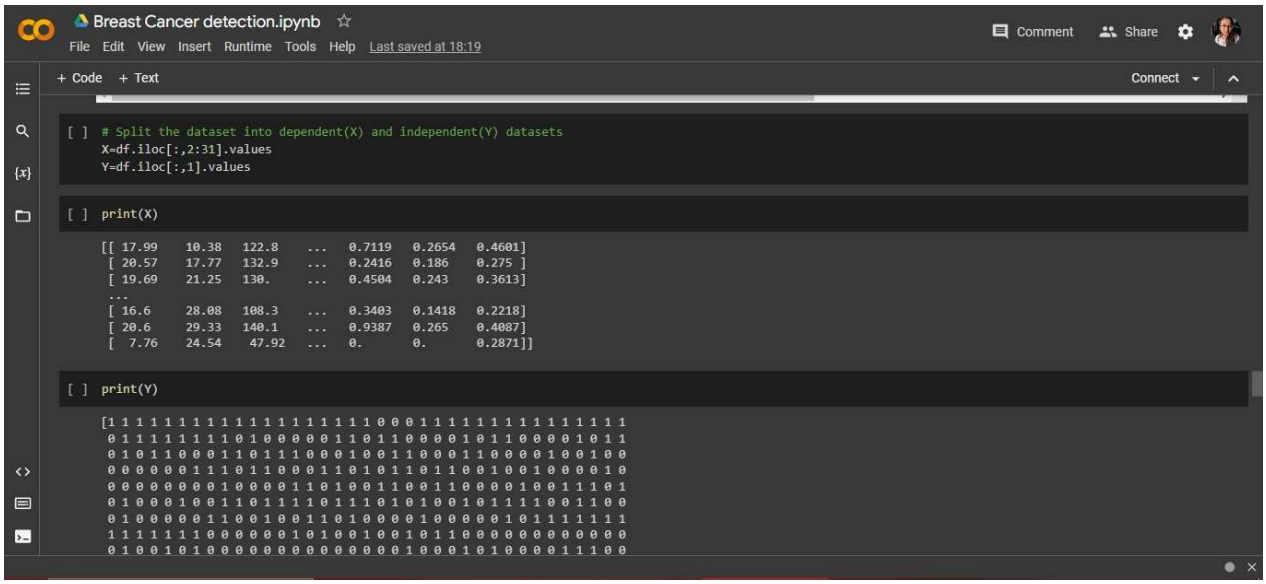
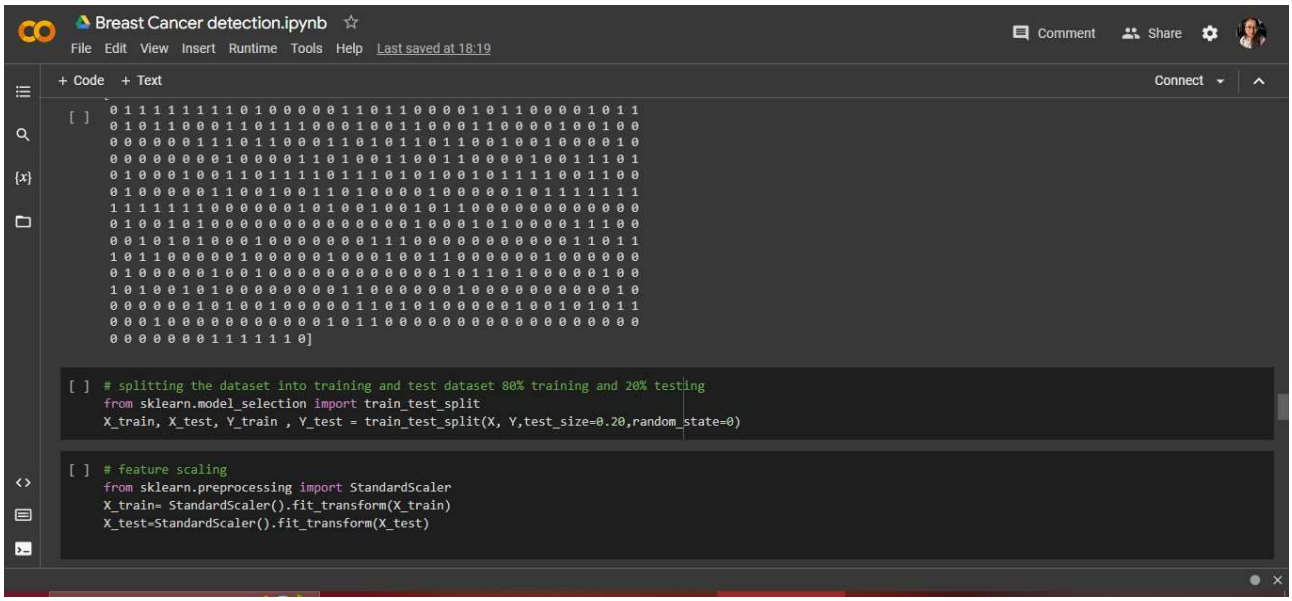
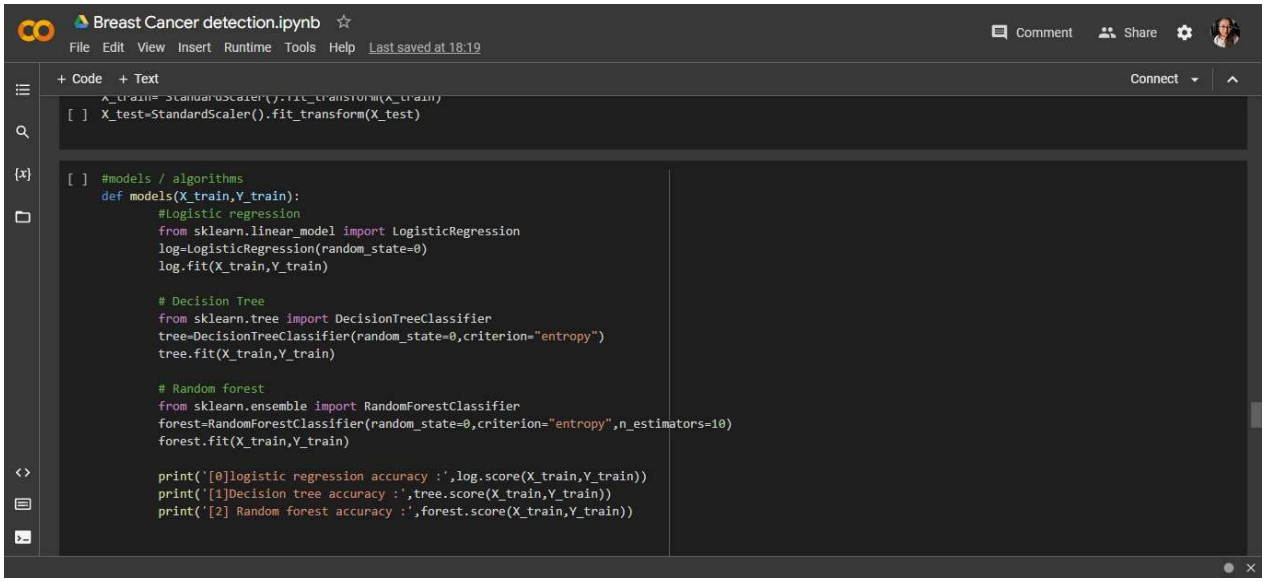
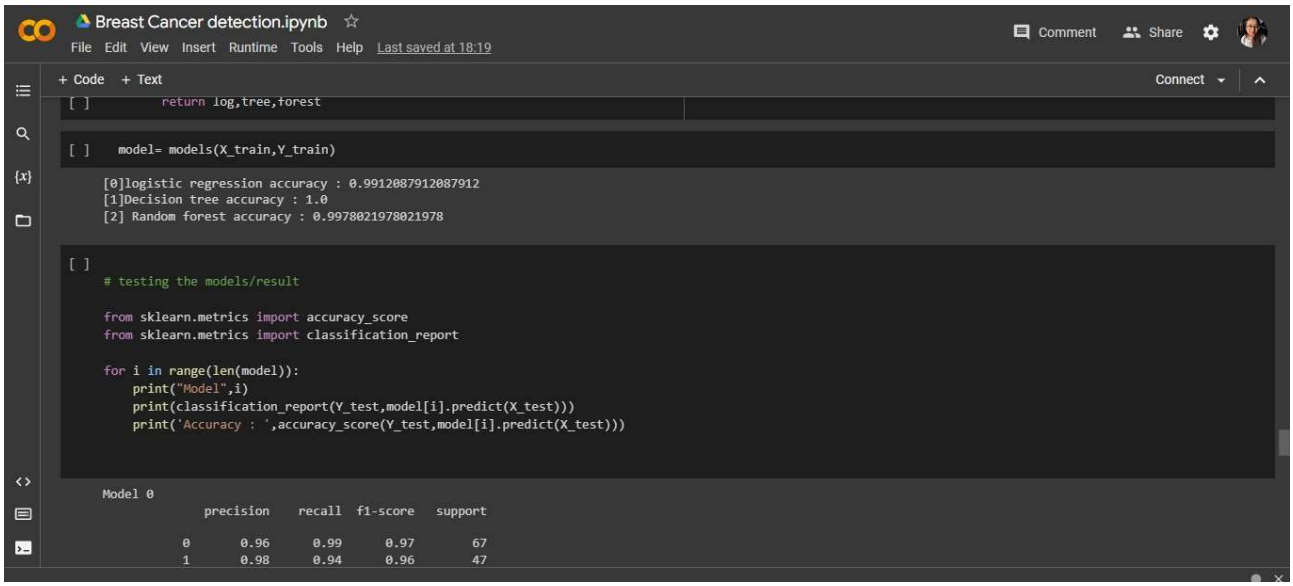

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code
+ Text
Connect
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
0.4
0.2
0.0
−0.2

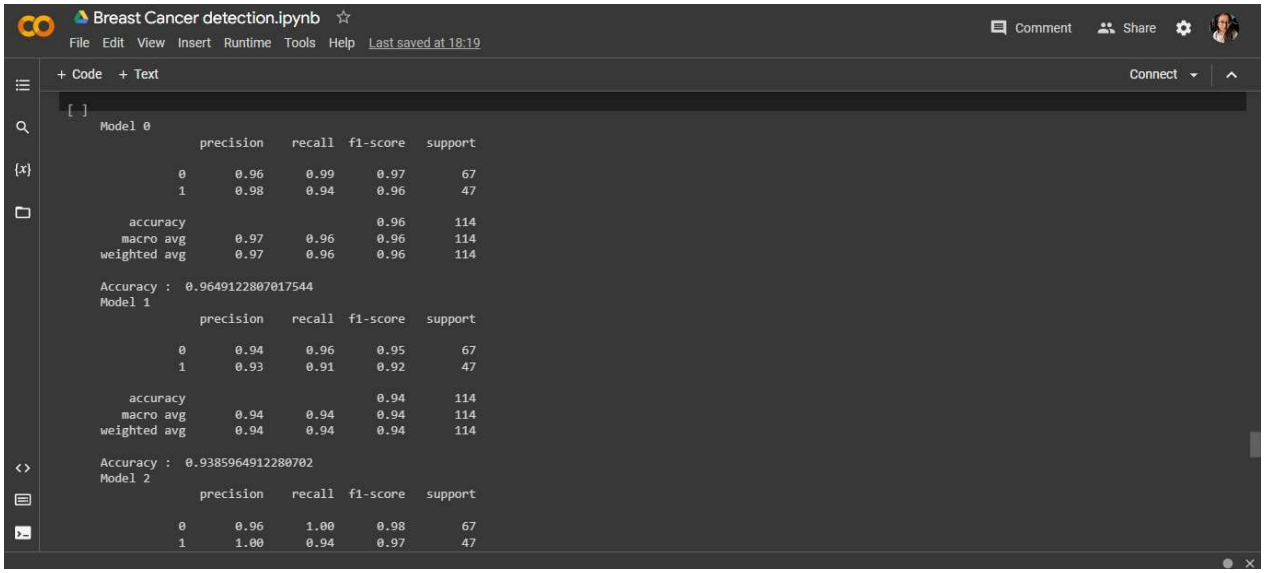
Fig 21.4

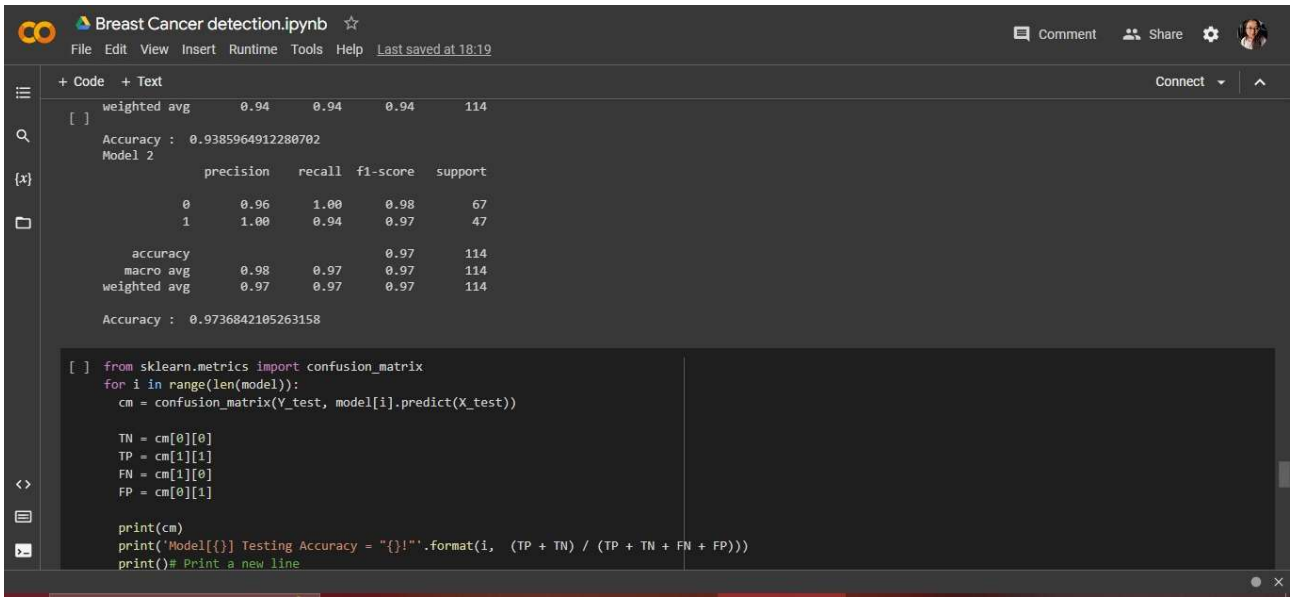

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code
+ Text
Connect
# Split the dataset into dependent(X) and independent(Y) datasets
X=df.iloc[:,2:31].values
Y=df.iloc[:,1].values
print(X)
[[17.99 10.38 122.8 ... 0.7119 0.2654 0.4601]
[20.57 17.77 132.9 ... 0.2416 0.186 0.275]
[19.69 21.25 130. ... 0.4504 0.243 0.3613]
...
[16.6 28.08 108.3 ... 0.3403 0.1418 0.2218]
[20.6 29.33 140.1 ... 0.9387 0.265 0.4087]
[7.76 24.54 47.92 ... 0. 0. 0.2871]]
print(Y)

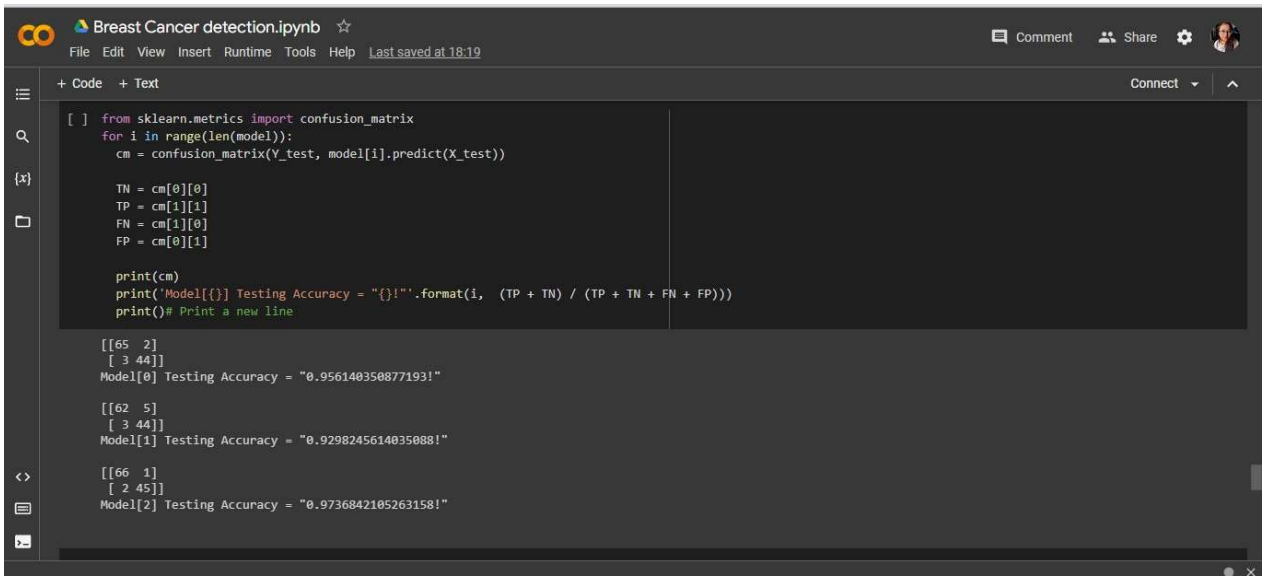

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
Connect
+ Code + Text
0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1
0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 1 1 1 0 1
0 1 0 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 1 0 0 1 1 0 0
0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0
0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0
1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 1
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 0]
# splitting the dataset into training and test dataset 80% training and 20% testing
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train , Y_test = train_test_split(X, Y,test_size=0.20,random_state=0)
# feature scaling
from sklearn.preprocessing import StandardScaler
X_train= StandardScaler().fit_transform(X_train)
X_test=StandardScaler().fit_transform(X_test)

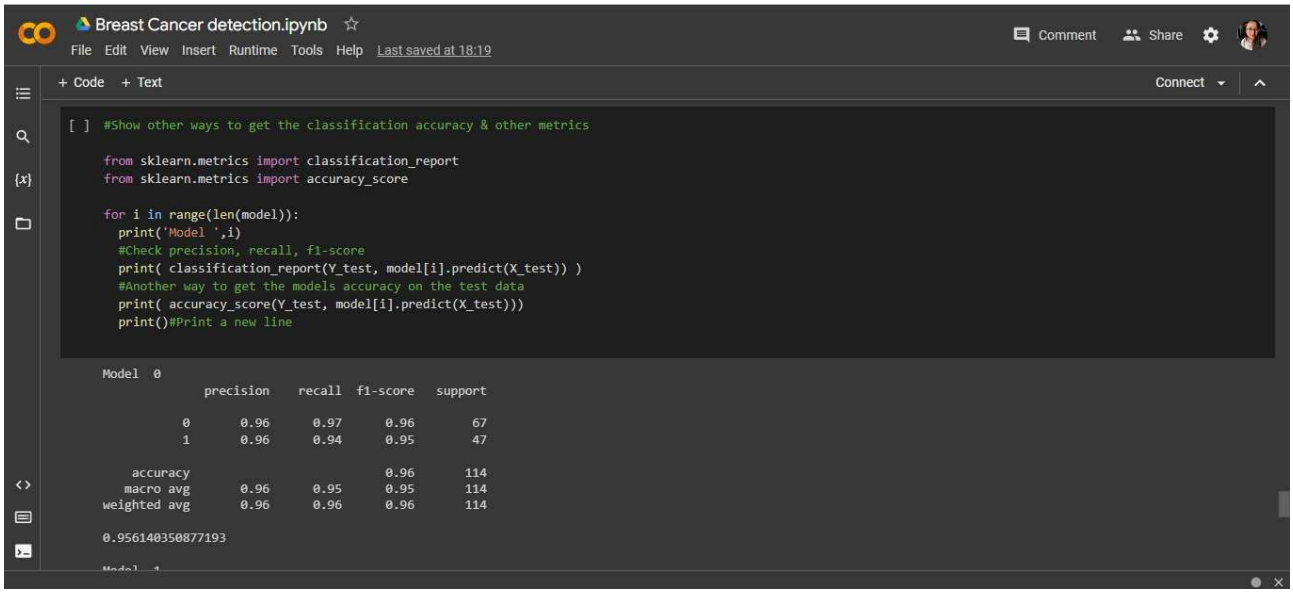

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
Connect
+ Code + Text
X_test=StandardScaler().fit_transform(X_test)
#models / algorithms
def models(X_train,Y_train):
#Logistic regression
from sklearn.linear_model import LogisticRegression
log=LogisticRegression(random_state=0)
log.fit(X_train,Y_train)
# Decision Tree
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier(random_state=0,criterion="entropy")
tree.fit(X_train,Y_train)
# Random forest
from sklearn.ensemble import RandomForestClassifier
forest=RandomForestClassifier(random_state=0,criterion="entropy",n_estimators=10)
forest.fit(X_train,Y_train)
print('[0]logistic regression accuracy :',log.score(X_train,Y_train))
print('[1]Decision tree accuracy :',tree.score(X_train,Y_train))
print('[2] Random forest accuracy :',forest.score(X_train,Y_train))

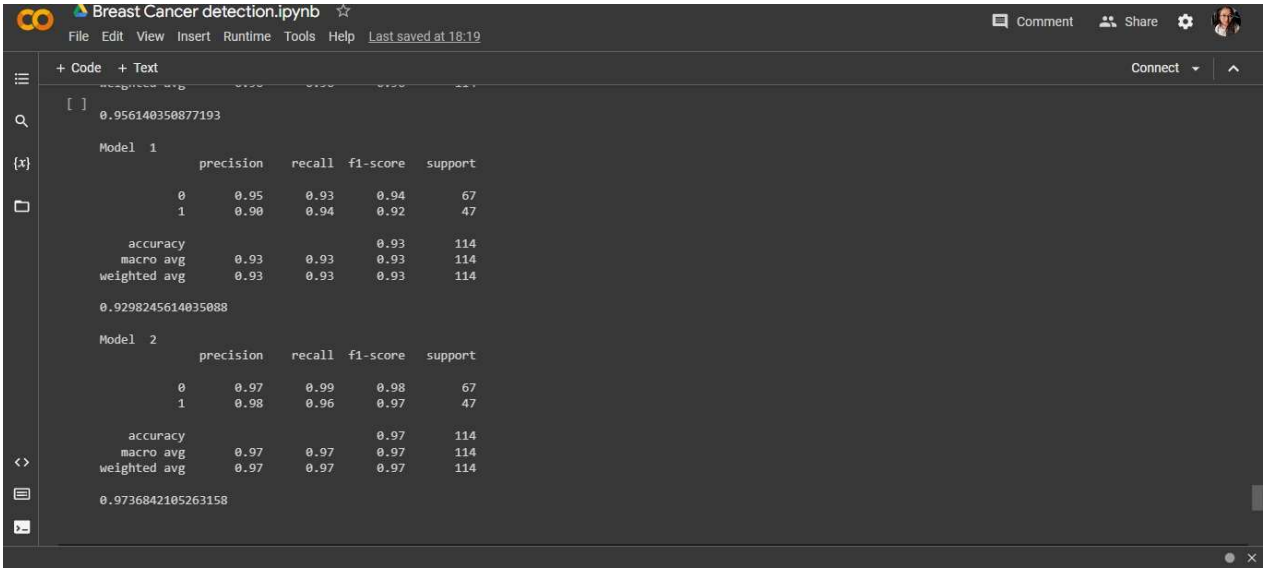

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] return log,tree,forest
[ ] model= models(X_train,Y_train)
[0]logistic regression accuracy : 0.9912087912087912
[1]Decision tree accuracy : 1.0
[2] Random forest accuracy : 0.9978021978021978
[ ]
# testing the models/result
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
for i in range(len(model)):
print("Model",i)
print(classification_report(Y_test,model[i].predict(X_test)))
print('Accuracy : ',accuracy_score(Y_test,model[i].predict(X_test)))
Model 0
precision recall f1-score support
0 0.96 0.99 0.97 67
1 0.98 0.94 0.96 47

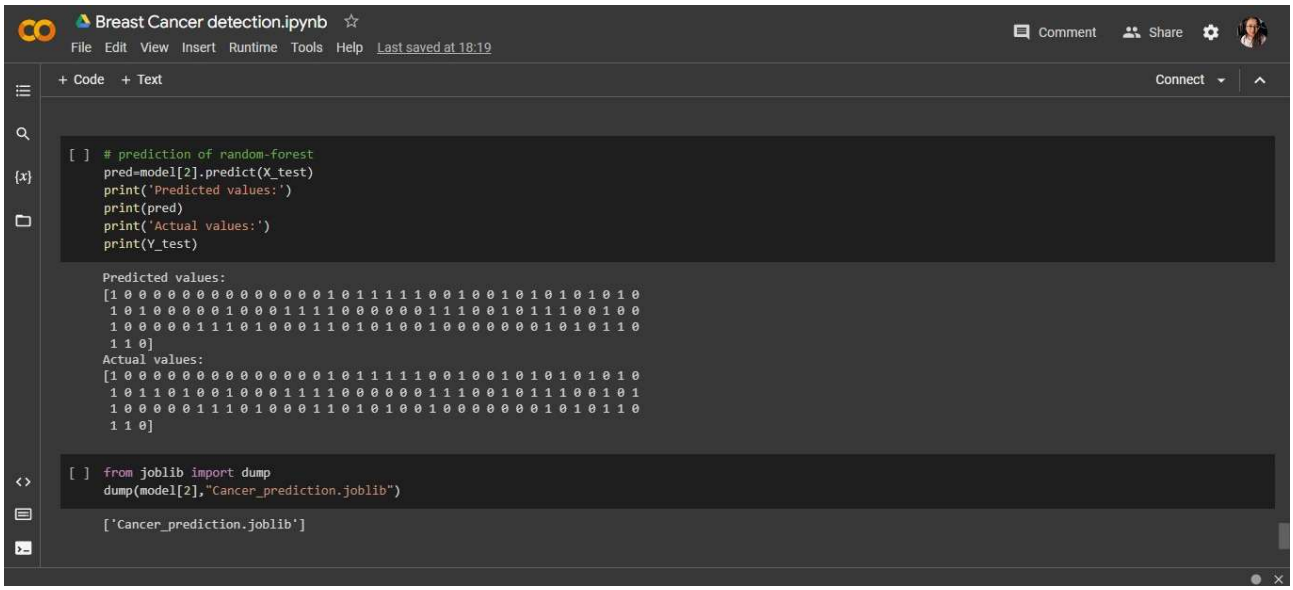

Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ]
Model 0
precision recall f1-score support
0 0.96 0.99 0.97 67
1 0.98 0.94 0.96 47
accuracy 0.96 114
macro avg 0.97 0.96 0.96 114
weighted avg 0.97 0.96 0.96 114
Accuracy : 0.9649122807017544
Model 1
precision recall f1-score support
0 0.94 0.96 0.95 67
1 0.93 0.91 0.92 47
accuracy 0.94 114
macro avg 0.94 0.94 0.94 114
weighted avg 0.94 0.94 0.94 114
Accuracy : 0.9385964912280702
Model 2
precision recall f1-score support
0 0.96 1.00 0.98 67
1 1.00 0.94 0.97 47


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
weighted avg 0.94 0.94 0.94 114
Accuracy : 0.9385964912280702
Model 2
precision recall f1-score support
0 0.96 1.00 0.98 67
1 1.00 0.94 0.97 47
accuracy 0.97 114
macro avg 0.98 0.97 0.97 114
weighted avg 0.97 0.97 0.97 114
Accuracy : 0.9736842105263158
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
cm = confusion_matrix(Y_test, model[i].predict(X_test))
TN = cm[0][0]
TP = cm[1][1]
FN = cm[1][0]
FP = cm[0][1]
print(cm)
print('Model[{}] Testing Accuracy = "{}!"'.format(i, (TP + TN) / (TP + TN + FN + FP)))
print()# Print a new line


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
from sklearn.metrics import confusion_matrix
for i in range(len(model)):
cm = confusion_matrix(Y_test, model[i].predict(X_test))
TN = cm[0][0]
TP = cm[1][1]
FN = cm[1][0]
FP = cm[0][1]
print(cm)
print('Model[{}] Testing Accuracy = "{}!"'.format(i, (TP + TN) / (TP + TN + FN + FP)))
print()# Print a new line
[[65 2]
[3 44]]
Model[0] Testing Accuracy = "0.956140350877193!"
[[62 5]
[3 44]]
Model[1] Testing Accuracy = "0.9298245614035088!"
[[66 1]
[2 45]]
Model[2] Testing Accuracy = "0.9736842105263158!"


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] #Show other ways to get the classification accuracy & other metrics
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
for i in range(len(model)):
print('Model ',i)
#Check precision, recall, f1-score
print(classification_report(Y_test, model[i].predict(X_test)))
#Another way to get the models accuracy on the test data
print(accuracy_score(Y_test, model[i].predict(X_test)))
print()#Print a new line
Model 0
precision recall f1-score support
0 0.96 0.97 0.96 67
1 0.96 0.94 0.95 47
accuracy 0.96 114
macro avg 0.96 0.95 0.95 114
weighted avg 0.96 0.96 0.96 114
0.956140350877193


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ]
0.956140350877193
Model 1
precision recall f1-score support
0 0.95 0.93 0.94 67
1 0.90 0.94 0.92 47
accuracy 0.93 114
macro avg 0.93 0.93 0.93 114
weighted avg 0.93 0.93 0.93 114
0.9298245614035088
Model 2
precision recall f1-score support
0 0.97 0.99 0.98 67
1 0.98 0.96 0.97 47
accuracy 0.97 114
macro avg 0.97 0.97 0.97 114
weighted avg 0.97 0.97 0.97 114
0.9736842105263158


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] # prediction of random-forest
pred=model[2].predict(X_test)
print('Predicted values:')
print(pred)
print('Actual values:')
print(Y_test)
Predicted values:
[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 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 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 1 0 1 0 1 1 0
1 1 0]
Actual values:
[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 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 1 0 1 0 1 1 0
1 1 0]
[ ] from joblib import dump
dump(model[2],"Cancer_prediction.joblib")
['Cancer_prediction.joblib']


Breast Cancer detection.ipynb
File Edit View Insert Runtime Tools Help Last saved at 18:19
Comment
Share
+ Code + Text
Connect
[ ] print(Y_test)
Predicted values:
[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 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 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 1 0 1 0 1 1 0
1 1 0]
Actual values:
[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 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 1 0 1 0 1 1 0
1 1 0]
[ ] from joblib import dump
dump(model[2],"Cancer_prediction.joblib")
['Cancer_prediction.joblib']

CONCLUSION

In this minor project, different types of models are reviewed andtheir accuracies are computed and compared with each other, so that the best cancer prediction model can be used by doctors in real life to identify breast cancer relatively faster thanprevious methods. Above examined study, proposed that the Random Forest Classification algorithm is proficiently utilized and efficient for detection of breast cancer as compared to Decision tree and Logistic Regression algorithms.

FUTURE ENHANCEMENTS

In the future, we plan to incorporate more data into our breast cancer detection project, either by gathering additional medical data on patients or leveraging publicly available datasets. We also intend to explore different feature engineering techniques, such as PCA or normalization, toidentify more effective ways of selecting and transforming the features in our dataset. Additionally, we plan to try other machine learning algorithms,such as support vector machines or gradient boosting, and incorporate ensemble methods like bagging or stacking. Finally, we intend to explore explainability techniques such as feature importance analysis and SHAP values to better understand how our machine learning models are makingpredictions about breast cancer.

REFERENCES

- **'WHO | Breast cancer', *WHO*. http://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/**
- **H. Asri, H. Mousannif, H.A. Moatassime, T. Noel**
  'Using Machine Learning Algorithms for Breast Cancer RiskPrediction and Diagnosis'
  Procedia Computer Science, 83 (2016), pp. 1064-1069
- **https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data**
- **Fabian Pedregosa, *et al.***
  "Scikit-learn: Machine Learning in Python"
  Journal of Machine Learning Research., 12 (2011), pp. 2825-2830