

***Faculty of Science and Technology***

**Assignment Coversheet**

|  |  |
| --- | --- |
| **Student ID number &**  **Student Name** | U3243862, Ang Phurba Sherpa |
| **Unit name** | Software Technology |
| **Unit number** | 4483 |
| **Unit Tutor** | Pranav Gupta |
| **Assignment name** | Capstone Project |
| **Due date** | 12/05/2023 |
| **Date submitted** | 12/05/2023 |

**You must keep a photocopy or electronic copy of your assignment.**

**Student declaration**

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source and other bibliographic details.

**Signature of student: \_\_\_\_\_ANG\_\_\_\_\_\_ Date: \_\_\_\_12/05/023\_\_\_\_\_\_\_\_**

**Table of Contents**

Introduction………………………………………………………………………………………………………………………………… 1

Methodology……………………………………………………………………………………………………………………………… 1

Stage 1: Algorithm Design Stage ……………………………………………………………………………………………………1

Dataset Description

Exploratory Data Analysis

Model Preparation and Development

Stage 2: Algorithm Implementation Stage ………………………………………………………………………………… 14

Algorithm Flowchart

Predictive Data Analytics Stage

Stage 3: Software Deployment Stage ……………………………………………………………………………………… 22

Testing ………………………………………………………………………………………………………………………………………….25

Conclusions……………………………………………………………………………………………………………………………………32

References …………………………………………………………………………………………………………………………….32

**Introduction:**

Certainly! One form of cancer that affects the cells in the breast tissue is breast cancer. It is a dangerous medical illness that might be fatal if it is not properly diagnosed and treated. The World Health Organization (WHO) reports that breast cancer is the most prevalent cancer in women globally and that it contributes significantly to yearly death.

To better understand the patterns and linkages that can aid in the diagnosis and treatment of the disease, we will examine breast cancer data in this study. We will classify breast cancer tumors as benign or malignant based on a variety of variables, including radius, texture, concavity, and area, using machine learning algorithms, a sort of artificial intelligence.

To more precisely identify breast cancer patients and provide more effective therapy, we prepare to look more thoroughly at these qualities to detect trends. The results of our study may increase the likelihood of survival, lessen the severity of sickness, and ultimately enhance a patient’s quality of life.

**Methodology:**

For this Project, we used three methodologies.

1. Gathering and cleaning the data: We obtained the dataset for breast cancer from trusted websites such as Kaggle followed by the procedure to clean the data.

2. Analysis and modeling: We used exploratory data analysis and logistic regression to classify tumors as malignant or benign.

3.Deployment: We created the model in a web application and posted it online for easy and secure access.

**Stage 1:**

Creating the algorithm, the breast cancer dataset was made available to us through the UCI Machine Learning Repository at the University of California, Irvine. Each of the 569 instances in the collection is a breast cancer tumor. There are 30 characteristics total in each instance, including radius, texture, concavity, and area. Digital pictures of breast mass aspirates were used to build the dataset, and the cytology of the mass was used to determine whether it was benign (B) or malignant (M).

**Exploratory Data Analysis (EDA):**

To learn more about the distribution and relationships between the attributes, we used EDA. Box plots, scatterplots, and histograms were utilized as data visualization techniques to present the data and spot patterns and connections.

**Code**:  
import streamlit as st  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score

**Description:**

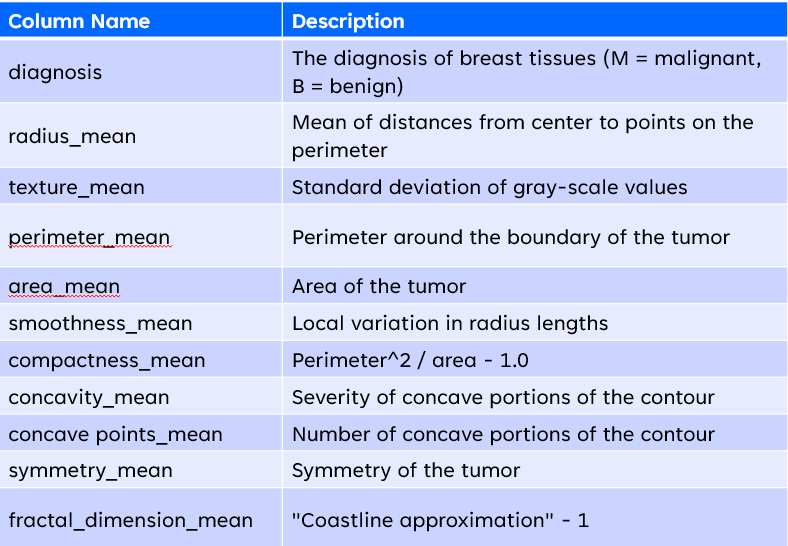
The required libraries for data analysis, visualization, and machine learning are imported by this code. It imports Scikit-learn for machine learning tasks, Pandas for working with data, Matplotlib and Seaborn for data visualization, and Streamlit for building web apps. For classification jobs, it also describes a logistic regression technique that will be utilized later in the code.

**Code:**  
# Load the dataset  
df = pd.read\_csv("breast-cancer.csv")

**Description**:

The "breast-cancer.csv" CSV file contains a dataset that is read by this code and loaded into the "df" variable. Data from breast cancer cases, including patient characteristics and medical records, are probably included in the collection. This will make it possible to use the dataset for additional data analysis and machine learning activities.

The few examples of dataset column are given below:



**Code:**

# Set up the web app  
st.set\_page\_config(page\_title="Breast Cancer Analysis and Classification Web App", page\_icon=":woman-health-worker:", layout="wide")

**Description:**

The setup for a web application that will be used to assess and categorize breast cancer cases is set up using this code. The layout and look of the web page may be customized using the st.set\_page\_config() method. It specifically changes the layout to "wide," the emblem to an emoji of a woman working in health care, and the title of the webpage to "Breast Cancer Analysis and Classification Web App." Setting the layout to "wide" will cause the website to occupy more horizontal space on the screen, enabling more material to be seen side by side.

**Result**:

A screenshot of a computer screen

Description automatically generated with low confidence

**Code:**

# Title and description  
st.title("Breast Cancer Analysis and Classification App")  
st.markdown("This app analyzes breast cancer data and predicts the diagnosis (M = malignant, B = benign).")  
  
# Sidebar for user input  
st.sidebar.title("User Input")  
st.sidebar.markdown("Select the parameters to analyze and predict:")  
  
# User input for analysis  
analysis\_options = st.sidebar.multiselect("Select parameters for analysis:", ["radius\_mean", "texture\_mean", "concavity\_mean", "area\_mean"], ["radius\_mean"])

**Description**:

The user interface of a web application that analyzes and categorizes breast cancer data is created by this block of code. The app's title and description are set in the first section. The user may choose the parameters they wish to study and forecast in the sidebar that is created in the next section. The user can choose one or more alternatives from the list that is displayed in the sidebar using the multiselect tool. Radius\_mean, texture\_mean, concavity\_mean, and area\_mean are the options for analysis that are offered. Radius\_mean is chosen by default. Later, a machine learning model that forecasts breast cancer diagnoses will be trained using the user's selection.

**Code**:

# 1. How is the diagnosis distributed?  
st.header("Distribution of Diagnosis")  
sns.countplot(x="diagnosis", data=df)  
plt.title("Distribution of diagnosis")  
st.pyplot()

**Description**:

This code distribution of diagnoses throughout the dataset is shown in a figure created by this code. The "seaborn" library is used to create a count plot that displays the frequency of each diagnosis (such as "M" for malignant or "B" for benign) within the dataset. The "st.pyplot()" function from the "matplotlib" package is then used to display the plot. The header "Distribution of Diagnosis" is added using the "streamlit" function from a separate package to give the plot a distinct title.

**Result:**

A picture containing text, screenshot, display, rectangle

Description automatically generated

Fig: Distribution of diagnosis

**Code**:

# 2. What is the distribution of the mean radius?  
if "radius\_mean" in analysis\_options:  
 st.header("Distribution of Mean Radius")  
 sns.histplot(df["radius\_mean"], kde=True)  
 st.pyplot()

**Description**:

This code checks if the user has selected "radius\_mean" option in the online application, this code creates a figure that shows the "Distributes of mean radius."

**Result**:

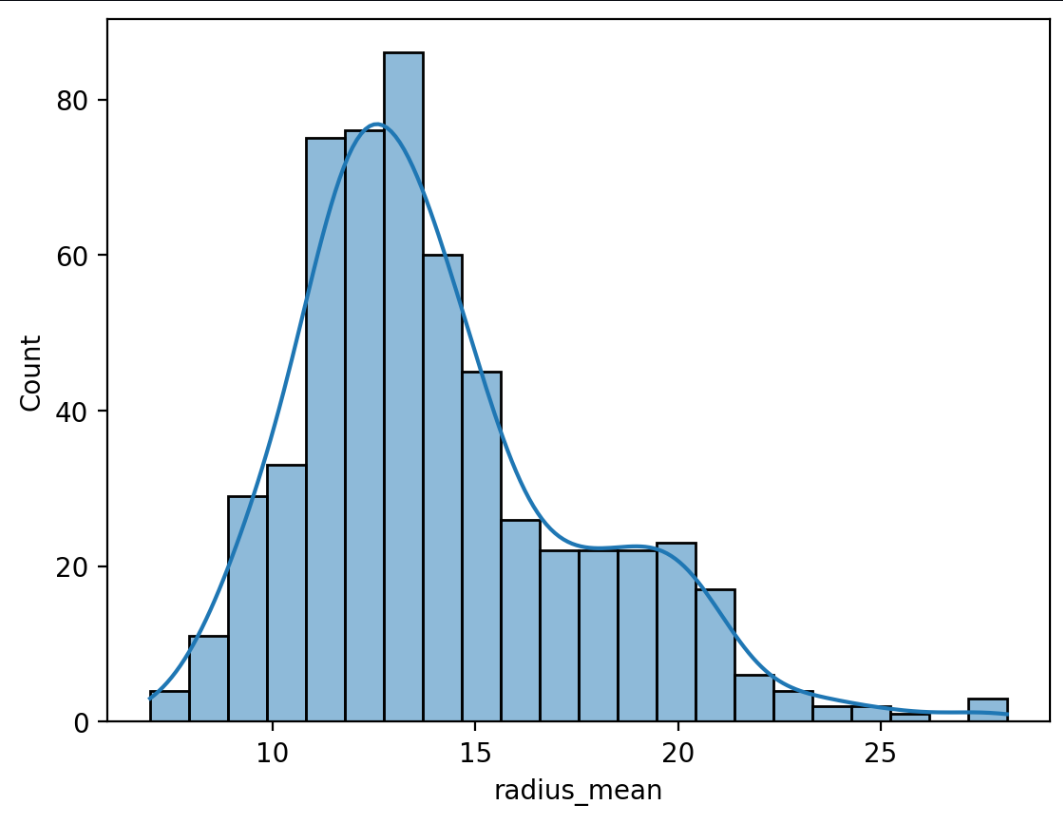


Fig:Distribution of mean Radius

**Code:**

# 3. Is there a correlation between mean radius and mean texture?  
if "radius\_mean" in analysis\_options and "texture\_mean" in analysis\_options:  
 st.header("Correlation between Mean Radius and Mean Texture")  
 sns.scatterplot(x="radius\_mean", y="texture\_mean", data=df)  
 st.pyplot()

**Description**:

This code creates a scatter plot that shows the relationship between the dataset's "radius\_mean" and "texture\_mean" values if the user selects both. The findings of the scatter plot, which is used to ascertain whether there is a relationship between the two parameters, are shown by the web tool.

**Result**:

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

**Code**:

# 4. How does the diagnosis vary with respect to mean concavity?  
if "concavity\_mean" in analysis\_options:  
 st.header("Diagnosis with respect to Mean Concavity")  
 sns.boxplot(x="diagnosis", y="concavity\_mean", data=df)  
 plt.title("Variation of diagnosis with respect to mean concavity")  
 st.pyplot()

**Description**:

This code checks if the user has selected "concavity\_mean" parameter as an option for analysis. If the code is used, it generates a box plot showing the distribution of the "concavity\_mean" values for both benign and malignant diagnoses. The diagram illustrates the connection between the diagnosis and the mean concavity. The box shows the interquartile range of the data, while the whiskers show the range of the data. The plan is then shown on the web app.

**Result**: A picture containing text, screenshot, diagram, line

Description automatically generated

**Code**:

# 5. How does the area of the tumor vary with respect to the diagnosis?  
if "area\_mean" in analysis\_options:  
 st.header("Tumor area with respect to Diagnosis")  
 sns.violinplot(x="diagnosis", y="area\_mean", data=df)  
 plt.title("Variation of tumor area with respect to diagnosis")  
 st.pyplot()

**Description**:

This code checks if the user has selected "area\_mean" as one of the analysis options. If it is selected, the code creates a plot that shows the variation in tumor area concerning diagnosis. The plot created is called a violin plot, and it displays the density of the data at different values. The plot is presented in the web app after it is generated. The main aim of this plot is to show if there is a significant difference in the tumor area concerning the diagnosis.

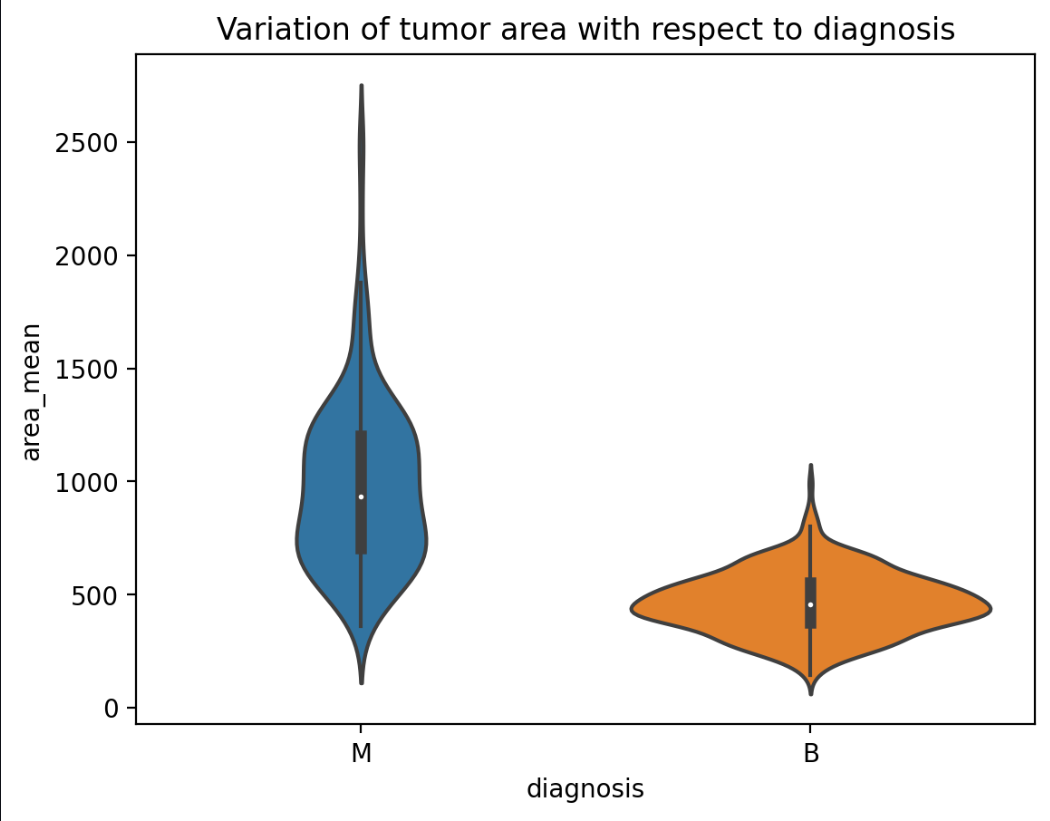
**Result**: 

Fig:Variation of tumor area with respect to diagnosis

# Classification  
# Prepare the data  
X = df.drop(["id", "diagnosis"], axis=1)  
y = df["diagnosis"]  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

**Description**:

This code prepare for the classification of the data. The dataset is divide into two part X any Y,where X have all the feature except “id” and “diagnosis” column, while y contation the function of “diagnosis” column.And the dataset in divide into two part Training and Testing 30% of the data are from the testing set, while the remaining 70% are from the training set. To ensure consistency in the results when the code is run repeatedly, the "random\_state" argument is set to 42.

# Train the model  
model = LogisticRegression(max\_iter=10000)  
model.fit(X\_train, y\_train)

**Description**:

This code is trained for the logistic regression model which is using the training data that was split in the previous step. This model is trained to predict the diagnosis based on users's parameter that they have entered. The max\_iter parameter is for the maximum number of iterations the algorithm will be taken before stopping.This model will be trained  until it accurately predicts the diagnosis of the breast cancer cases in the training data, which is predicted throgh the input parameter.

# Predict on test data  
y\_pred = model.predict(X\_test)

**Description**:

This code will predict the value for y variable using X\_test using the trained logistic regression model which is the model that we have previously trained. The value that is predicted will be displayed on y\_pred.

# Evaluate the model  
accuracy = accuracy\_score(y\_test, y\_pred)  
st.write(f"Accuracy: {accuracy}")

**Description**:

This code evaluate the accuracy of the model by comparing the exact value and predicted value. accuracy\_score from scikit-learn library is used for measuring the accuracy score. The result is then displayed on the web app using the st.write function. The accuracy score represents the percentage of correctly predicted outcomes from the test data.

**Result**:

A picture containing text, font, screenshot, design

Description automatically generated

## Stage 2: Algorithm Implementation Stage

Logistic regression was used as the machine learning approach for classifying breast cancer.Using the train\_test\_split technique from Sklearn, the data was divided into two training and testing sets. Various Python libraries like seaborn and plotly were used to create informative graphs to visualise the data and informating for the dataset we collected for cancer history of Bangladesh. And Streamlit was employed for the implementation making the application user-friendly.

**Algorithm Flowchart: A screenshot of a computer

Description automatically generated with medium confidence**

**Predictive Data Analytics Stage:**

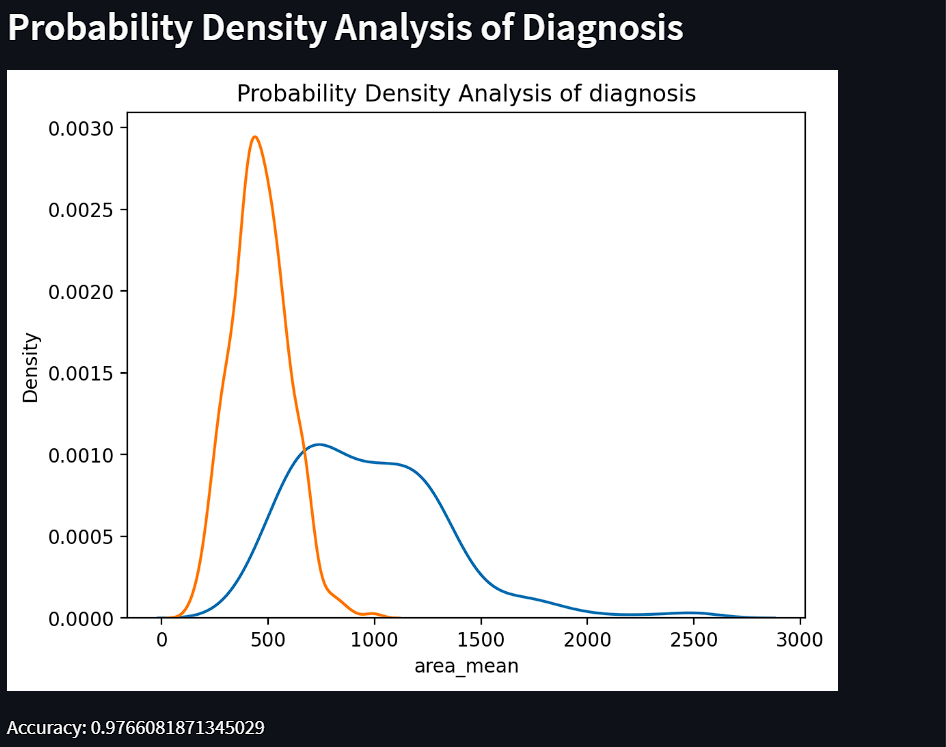
**Code:**

# PDA of Diagnosis  
st.header("Probability Density Analysis of Diagnosis")  
fig, ax = plt.subplots()  
for diagnosis in ["M", "B"]:  
 sns.kdeplot(data=df[df["diagnosis"] == diagnosis], x="area\_mean", label=diagnosis, ax=ax)  
plt.title("Probability Density Analysis of diagnosis")  
st.pyplot()

**Description:**

Based on the area mean of the tumor, this code creates a Probability Density Analysis   graph of breast cancer diagnosis. This code section shows two probability density curves, one for each type of breast cancer diagnosis ("M" and "B"), using the Streamlit and Seaborn Python modules. The x-axis depicts the tumor's area mean, while the y-axis reflects the probability density of the diagnosis. The graph is then shown on the web application using the "st.pyplot()" method.

**Result:**



## Code:

# PDA of mean radius  
st.header("Probability Density Analysis of Mean Radius")  
sns.kdeplot(x="radius\_mean", data=df, shade=True)  
plt.title("Probability Density Analysis of mean radius")  
st.pyplot()

## Description:

This code generates a probability density map of the distribution of the variable "mean radius" in a dataset. The graphic depicts the probability of various mean radius values happening in the data, with bigger peaks suggesting areas of higher density where the data is more likely to occur. This can aid in visualizing the shape of the data and identifying any trends or abnormalities in the distribution. The code creates the plot with the Seaborn library and displays it in a web application with the Streamlit library.

## Result:

## A picture containing text, screenshot, diagram, plot Description automatically generated

## Code:

# PDA of correlation between mean radius and mean texture  
st.header("Probability Density Analysis of Mean Radius and Mean Texture correlation")  
sns.kdeplot(x="radius\_mean", y="texture\_mean", data=df, shade=True)  
plt.title("Probability Density Analysis of mean radius and mean texture correlation")  
st.pyplot()

## Description:

This code generates a graph displaying the probability density of the association between a dataset's "mean radius" and "mean texture" attributes. . The graph is a two-dimensional figure in which the x-axis represents "mean radius" and the y-axis represents "mean texture." The graph's darker sections show locations where the connection between the two traits is more likely to occur, while the lighter regions suggest areas where the association is less frequent. This form of analysis can aid in the identification of patterns and correlations between various characteristics in a dataset.

## Result:

## 

## Code:

# PDA of diagnosis vary with respect to mean concavity  
st.header("Probability Density Analysis of Diagnosis with respect to Mean Concavity")  
sns.kdeplot(x="concavity\_mean", hue="diagnosis", data=df, shade=True)  
plt.title("Probability Density Analysis of diagnosis with respect to mean concavity")  
st.pyplot()

## Description:

This code generates a figure that demonstrates how a patient's breast cancer diagnosis (whether benign or malignant) fluctuates with the mean concavity of cells in their breast tissue. It depicts the distribution of mean concavity values for both benign and malignant cases using probability density analysis. The resultant figure may be used to see if there are any variations in mean concavity between benign and malignant instances, as well as the overlap or separation of respective distributions.

## Result:

## A picture containing text, screenshot, diagram, plot Description automatically generated

## Code:

# PDA of area of the tumor vary with respect to the diagnosis  
st.header("Probability Density Analysis of Tumor area with respect to Diagnosis")  
sns.kdeplot(x="area\_mean", hue="diagnosis", data=df, shade=True)  
plt.title("Probability Density Analysis of tumor area with respect to diagnosis")  
st.pyplot()

## Description:

This code does a Probability Density Analysis on the tumor area to see how it fluctuates depending on the diagnosis. It generates a kernel density estimate (KDE) graphic that depicts the distribution of tumor area for each diagnostic group (malignant or benign). The plot is coloured to make the places with the highest density more visible.

## Result:

## A picture containing text, screenshot, plot, diagram Description automatically generated

## Stage 3: Software Deployment

## Streamlit was used for building the web application which is the one of the library for creating the interactive web applications. In this application users could choose the parameters for analysis and prediction via the online app. The app showed breast cancer data visualizations and offered information on how well the predictions of the logistic regression model performed. The application is simple and convinent to use for any type of user for research and predictions purposes.

## Screenshot of software deployment:

## 

## A screenshot of a computer Description automatically generated with medium confidence

## A screen shot of a graph Description automatically generated with medium confidence

**A picture containing text, screenshot, software, multimedia software

Description automatically generated**

## Testing

## Any web application needs to be tested before making it public to the users for their convinence.We had also gone through various tests of our web application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | EDA | PDA | Accuracy | Status |
| radius\_mean | YES | YES | 0.9766081871345029 | PASS |
| texture\_mean | YES | YES | 0.9766081871345029 | PASS |
| concavity\_mean | YES | YES | 0.9766081871345029 | PASS |
| area\_mean | YES | YES | 0.9766081871345029 | PASS |
| diagnosis | YES | YES | 0.9766081871345029 | PASS |

**Evidence of testing (Screenshots):**

A screenshot of a computer program

Description automatically generated with medium confidence

Figure : (Successful run) Run command to run application in terminal

A screenshot of a computer

Description automatically generated with medium confidence

Figure : When no any parameters are chosen

A picture containing text, screenshot, graphics software, multimedia software

Description automatically generated

Figure : EDA of Distribution of Mean Radius

A screen shot of a graph

Description automatically generated with medium confidence

Figure : PDA of mean radius

A screen shot of a graph

Description automatically generated with medium confidence

Figure : EDA of correlation of Mean Radius and Mean Texture

A screen shot of a graph

Description automatically generated with low confidence

Figure : PDA of correlation of Mean Radius and Mean Texture

A screenshot of a graph

Description automatically generated with low confidence

Figure : EDA of Diagnosis with respect to Mean Concavity

A picture containing text, screenshot, diagram, software

Description automatically generated

Figure : PDA of Diagnosis with respect to Mean Concavity

A picture containing text, screenshot, diagram, multimedia software

Description automatically generated

Figure :EDA of Tumor area with respect to Diagnosis

A picture containing text, screenshot, diagram, multimedia software

Description automatically generated

Figure : PDA of Tumor area with respect to Diagnosis

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Figure :EDA of Distribution of diagnosis

A screen shot of a graph

Description automatically generated with medium confidence

Figure :PDA of Distribution of diagnosis

A screen shot of a graph

Description automatically generated with medium confidence

Figure : Accuracy displayed at the bottom of graph ,newly selected parameter shows accuracy below PDA.

A picture containing text, screenshot, plot, diagram

Description automatically generated

Figure :Full screen view

## Conclusion

## In this project the data analysis, machine learning and web development skills were influenced to build a web application for breast cancer analysis and classification. Healthcare practitioners may utilize the web app to diagnose breast cancer with accuracy and efficiency. The online app may eventually get more machine learning models and data visualization choices. As a outcome of this project, it was a great opportunity and experience of developing a web application using Python programming. We are looking forward for more creative projects like this which will be very beneficial in our career in information and technology as a Python developer.

## References

## Streamlit. (n.d.). Get started. Retrieved May 12, 2023, from <https://docs.streamlit.io/library/get-started/create-an-app>

## Allunia. (2020). Breast Cancer. Retrieved May 12, 2023, from Kaggle: <https://www.kaggle.com/code/allunia/breast-cancer>