7th SEMESTER PROJECT REPORT

"BREAST CANCER PREDICTION USING MACHINE LEARNING"

(Submitted in partial fulfilment for the award of degree of B. Tech. In Computer Science and Engineering under Assam Science and Technology University)

Submitted By:
Abhijit Saikia (203410007001)
Jyoti Raj Nath (203410007022)

Department of Computer Science and Engineering
Dhemaji Engineering College
Dhemaji, 787057

December, 2023

DECLARATION

We hereby declare that the presented work represents largely our own ideas and work in our own words. Where others ideas or words have been included, we have adequately cited and listed in the reference materials. We have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the work. We understand that any violation of the above will cause disciplinary action by the College.

Abhijit Saikia (203410007001)	Jyoti Raj Nath (203410007022)

Date: ___________

CERTIFICATE
(From Department of Computer Science and Engineering)

This is to certify that the 7th Semester Project-I report entitled “**Breast Cancer Using Machine Learning**” submitted by Abhijit Saikia(203410007001) and Jyoti Raj Nath(203410007022) is accepted in partial fulfilment for the award of degree of B. Tech. In Computer Science and Engineering under Assam Science & Technology University.

Signature of Faculty Coordinator

Pallabi Patowary
Computer Science and Engineering Department

Date: __________

ACKNOWLEDGEMENT

It is our great pleasure to acknowledge the assistance and contribution of the individuals who co-operated with us to complete the projects successfully. The completion of this undertaking was only possible with the participation and assistance of some people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged.

We would like to extend our gratitude to our project guide Pallabi Patowary (HOD, Department of Computer Science And Engineering) for her endless support, and kind and understanding spirit during our project.

Moreover, we would like to acknowledge the assistance and support we received from Bhrigu Kumar Katoky (Lab Technician, Department of Computer Science And Engineering) during our project time.

We also take the opportunity to express our sincere thanks to our Principal, Dr. Dilip Kumar Bora, and all the faculty members and lab assistants of the Department of Computer Science And Engineering for their help and encouragement.

We even record our gratitude to our parents for their ceaseless support, either morally, financially or physically.

ABSTRACT

Breast cancer is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society. The early diagnosis can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Thus, the correct diagnosis of Breast cancer and classification of patients is the subject of much research. Because of its unique advantages in critical feature detection from complex Breast Cancer Datasets, machine learning is widely recognized as the methodology of choice in Breast Cancer pattern classification and forecast modelling.

Here we use 3 machine learning models -Logistic Regression, SVM, and K Nearest Neighbors Algorithm

LIST OF CONTENT

Particulars **Pg. No**

Chapters

1. INTRODUCTION

Breast cancer is among the most serious illnesses/diseases in India, causing many deaths in the current situation. Due to changes in food and lifestyle, the number of cancer cases in women is increasing day by day. It is the second most common cause of death in women in the world. This uses concepts of Deep learning (DL) and Machine learning (ML) to predict breast cancer based on the data obtained. This cancer is produced by abnormal growth of fatty and fibrous tissues, and the different phases of cancer are caused by cancer cells spreading throughout the tissue. This is one of the most common cancers that affects women, but other types of cancer and those who are affected by them can be treated greatly, according to a government survey, when compared to breast cancer. The various phases of breast cancer are identified via proper treatment and detailing. If we do not provide proper therapy to our patients, it will result in their death. Some methods for establishing an accurate diagnosis of breast cancer have been presented. Because the dataset contains a variety of distinct report attributes, machine learning may be easily applied to the dataset for prediction. Even by using Technology that is not fully automatically designed to give the output. Hence here we propose the fully automatic classification and prediction of breast cancer based on dataset, using ensemble machine learning techniques. This learning technique is recognised as the method to predict and classify datasets.

Earlier methods for classifying data were used, despite their lower accuracy, because they could be used for proper categorization and prediction. Machine learning techniques are used to extract important and hidden features.

Typical cancer screening procedures are grounded on the "gold standard", which consists of three tests: clinical evaluation, radiological imaging, and pathology testing. This traditional technique, which is based on regression, detects cancer, whereas new ML techniques and algorithms are built on model creation. In its training and testing stages, the model is meant to forecast unknown data and offer a satisfactory predicted outcome. Preprocessing, feature selection or extraction, and classification are the three major methodologies used in machine learning. The feature extraction part of the machine learning method is crucial for cancer diagnosis and prediction. This process may differentiate between benign and malignant tumors.

The analysis of various breast cancer phases is contingent upon precise treatment and exhaustive detailing. The absence of tailored therapeutic interventions can precipitate fatal outcomes for patients. Presenting an array of methods for precise breast cancer diagnosis, this report advocates for the application of machine learning to the dataset due to its diverse and comprehensive attribute composition. Proposing a paradigm shift toward fully automated breast cancer classification and prediction leveraging ensemble machine learning techniques, this approach is heralded as an efficacious means for accurate dataset categorization.

Historically, conventional methods, albeit less accurate, were employed for their proficiency in facilitating appropriate categorization and prediction. Modern machine learning techniques, however, excel in uncovering crucial yet latent features. While conventional cancer screening methods rely on the triad of clinical evaluation, radiological imaging, and pathology testing, the advent of ML techniques has ushered in a transformative era of model-driven cancer detection. Distinguishing itself from traditional regression-based approaches, these novel ML algorithms aim to create models capable of forecasting unknown data, contributing to a more reliable and satisfactory predicted outcome. Emphasizing preprocessing, feature selection or extraction, and classification, machine learning methodologies critically underscore the significance of feature extraction in the cancer diagnosis and prediction process. This step becomes instrumental in discerning between benign and malignant tumors, marking a substantial leap forward in the pursuit of accurate and early cancer detection.

1.1 Objective:

The objective of this project is to develop a predictive model that can accurately assess the likelihood of an individual having breast cancer based on relevant features and data. The primary goals include :

Early Detection: Create a model that can identify potential cases of breast cancer at an early stage. Early detection is crucial for improving treatment outcomes and reducing mortality rates.

Accuracy and Reliability: Develop a machine learning model that is highly accurate and reliable in predicting breast cancer. The model should be able to minimize false positives and false negatives, ensuring trustworthy results.

Risk Stratification: Stratify individuals into different risk categories based on their likelihood of developing breast cancer. This can help in tailoring healthcare strategies and interventions according to the level of risk.

Feature Importance Analysis: Identify and analyze the most significant features or risk factors contributing to the prediction. This can provide valuable insights into the factors influencing breast cancer risk and guide future research and clinical practices.

User-Friendly Interface: Design an interface or application that allows healthcare professionals and individuals to easily input data and obtain predictions. The user interface should be intuitive and accessible for practical use.

Education and Awareness: Educate healthcare professionals and the public about the benefits and limitations of the predictive model. Raise awareness about the importance of regular screening and early detection of breast cancer.

1.2 Motivation:

This project can be highly motivating and impactful because of some reasons:

Health Impact: Breast cancer is a prevalent and potentially life-threatening disease. Predictive models can aid in early detection, allowing for timely intervention and improved chances of successful treatment. Our project has the potential to contribute significantly to healthcare outcomes and save lives.

By developing an accurate predictive model, we contribute to public health initiatives. Early detection and prevention strategies are crucial in reducing the overall burden of breast cancer on healthcare systems and society.

Empowering Patients: This project can empower individuals by providing them with information about their risks. This knowledge can lead to more informed decisions about screening, lifestyle choices, and proactive healthcare measures.

Advancement in Technology: Working on a breast cancer prediction project allows you to leverage cutting-edge technology and contribute to the growing field of healthcare technology. This can be intellectually stimulating and contribute to advancements in the intersection of healthcare and artificial intelligence.

Interdisciplinary Learning: This project provides an opportunity to work at the intersection of medicine, biology, and computer science. Collaborating with experts from different fields can expand our understanding and skill set, offering a unique learning experience.

Career Opportunities: Developing expertise in healthcare-related machine learning projects can open up career opportunities in the technology and healthcare sectors. It could be an initial step for further research, work in medical institutions, or contributing to health-tech startups.

1.3 Description of chapters:

The introduction chapter sets the stage for the report, providing an overview of the breast cancer detection model. It introduces the significance of early breast cancer diagnosis, the motivation behind developing the model, and the overall goal of the project. This section typically includes background information on breast cancer, the need for reliable detection methods, and an overview of the report structure.

In the Related Study /Background , we delve into existing literature and studies related to breast cancer detection and machine learning models. This section reviews relevant research, discusses the strengths and limitations of previous approaches, and highlights gaps in the existing knowledge. It provides a comprehensive understanding of the state-of-the-art in breast cancer detection, laying the foundation for the proposed methodology.

The Proposed Methodology outlines the approach taken to develop the breast cancer detection model. It details the machine learning algorithms used (ensemble of KNN, SVM, Logistic Regression), the preprocessing steps (such as feature scaling), and the rationale behind these choices. This chapter may also include information on data collection, dataset characteristics, and any modifications made to enhance model performance.

In the Results and Discussion, we present the outcomes of our breast cancer detection model. This includes the evaluation metrics, accuracy, confusion matrix, and any other relevant performance indicators. The section discusses the significance of the results, compares them with existing literature, and interprets findings. Visualizations, such as graphs and charts, may be included to enhance understanding.

Conclusion & Future Work summarizes the key findings and insights from the study. It discusses the implications of the results in the context of breast cancer detection. It serves as a reflection on the project's success and opens avenues for further exploration.

2. RELATED STUDY/BACKGROUND

2.1 Description ML model

The model is likely designed for cancer classification, particularly in the context of medical data. The goal is to assist medical professionals in identifying whether a tumor is malignant (cancerous) or benign (non-cancerous) based on specific characteristics or features.

Use Cases and Applications:

Early Detection of Cancer:

The model can aid in the early detection of cancerous tumors, which is crucial for timely medical intervention and improved treatment outcomes.

Decision Support for Medical Professionals:

Healthcare professionals can use the model's predictions as additional information when making decisions about patient diagnosis and treatment plans.

Reducing False Negatives:

By providing an additional layer of analysis, the model may help reduce instances of false negatives, where a cancerous tumor is incorrectly identified as non-cancerous.

2.2 Ensemble Method used:

In this project, the use of an ensemble method, specifically a Voting Classifier, signifies the adoption of a strategy where multiple machine learning models are combined to make more accurate and robust predictions. The rationale behind ensemble methods lies in the idea that aggregating the predictions of diverse models can compensate for individual model weaknesses and enhance overall performance.

The ensemble method used in this project, the Voting Classifier, operates on the principle of combining the predictions of multiple base models to make a final prediction. There are two primary types of voting in a Voting Classifier: hard voting and soft voting. In hard voting, each base model "votes" for a specific class, and the class with the majority of votes becomes the final prediction. On the other hand, soft voting involves taking the average probability of each class predicted by the base models, and the class with the highest average probability is chosen as the final prediction.

In the context of breast cancer prediction, different base models, each potentially capturing distinct patterns in the data, can be integrated into the ensemble. For instance, KNN, Support Vector Machines, and Logistic Regression models might be part of the ensemble. The Voting Classifier leverages the collective knowledge of these models, providing a more robust prediction than any individual model could achieve. This approach contributes to improved accuracy and generalization in predicting whether an individual is at risk of breast cancer, making it a valuable strategy in the context of healthcare applications.

2.3 Tools/ Packages used:

The **tools** and **libraries** used in the project and their functions are as follows :

1.**NumPy:** NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

2. **Pandas:** Pandas is a powerful data manipulation library that provides data structures like DataFrames. It's used for cleaning, transforming, and analyzing the dataset.

3. **Matplotlib and Seaborn:** Matplotlib and Seaborn are used for data visualization. Matplotlib is a basic plotting library, and Seaborn is built on top of Matplotlib, providing a high-level interface for statistical graphics.

4. **Scikit-learn:** Scikit-learn is a machine learning library that provides simple and efficient tools for data mining and data analysis. It includes various modules for tasks such as classification, regression, clustering, and model selection.

5. **Flask:** Flask is a web framework for building web applications. In your case, it's used to create a simple web interface for users to interact with your breast cancer detection model.

6. **Joblib:** Joblib is used for saving and loading Python objects, such as your trained machine learning model, efficiently.

7. **Scipy:** Scipy builds on NumPy and provides additional functionality for scientific and technical computing. It includes modules for optimization, integration, interpolation, and more.

8. **Flask-WTF(WTForms):** Flask-WTF is an extension for Flask that integrates with WTForms, a flexible forms validation and rendering library. It helps in creating and handling web forms.

9. **Bootstrap (Front-End Framework):** Bootstrap is a popular front-end framework for building responsive and visually appealing web pages. It can be used to enhance the design and layout of your web application.

10. **HTML and CSS:** HTML is the standard markup language for creating web pages, and CSS is used for styling and layout. Together, they define the structure and appearance of the web pages in your application

2.4 Software /Database used:

Jupyter Notebook : Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports various programming languages, with Python being one of the most widely used. Notebooks can be exported to various formats, such as HTML, PDF, or slideshows, making it easy to share your work with others

Anaconda : Anaconda is a distribution of Python and other scientific computing packages. It simplifies the process of managing and installing Python packages for data science and machine learning. Anaconda provides a user-friendly graphical interface called Anaconda Navigator, allowing you to manage environments, install packages, and launch applications like Jupyter

Notebook and Spyder.Environments in Anaconda help isolate different project dependencies, making it easier to manage package versions and avoid conflicts.

Python : Python is a computer programming language often used to build websites and software, automate tasks, and analyze data. Python is a general-purpose language, not specialized for any specific problems, and used to create various programmes. This versatility and its beginner-friendliness have made it one of the most used programming languages today.

Spyder Notebook : Spyder is an open-source integrated development environment (IDE) designed for scientific computing, data analysis, and machine learning. It comes pre-installed with the Anaconda distribution. It provides features like variable explorer, IPython console, and integrated help, making it a powerful IDE for data analysis and exploration. Spyder also supports integration with version control systems like Git, facilitating collaborative work on projects.

3. CODE OVERVIEW

The code consists of **three modules**:

3.1 Module 1: Machine Learning Model

Module Name: ensemble_model.joblib
Libraries used: numpy: For numerical operations.
Joblib: For loading and saving the trained model

Task & Function:

- Loading Pre-trained Model and Scaler:

ensemble_model = joblib.load('ensemble_model.joblib'): Load the pre-trained ensemble model.
scaler = joblib.load('scaler.joblib'): Load the pre-trained scaler.
Making Predictions:

Given user input features, scale them using scaler.transform.
Make predictions using ensemble_model.predict.
Return the predicted result (Malignant or Benign).

- The machine learning model is loaded using the joblib.load function in the Flask app.

The model is a pre-trained ensemble model created using K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression.
It takes user input, scales the features, and predicts whether the breast cancer is Malignant or Benign.
The prediction result is then rendered back to the user on the web page.
Overall, this setup allows users to interact with the machine learning model through a web interface, providing input features related to breast cancer and receiving predictions in real time.

3.2 Module 2: Flask App

Module Name: **app.py**
Libraries used: Flask: Web framework for building the application.
render_template: Function to render HTML templates.
request: Object to handle HTTP requests.
joblib: For loading the pre-trained model.
numpy: Used for numerical operations.

Task & Function:

This part is responsible for scheduling a Zoom meeting using the extracted meeting details. It uses the Selenium library to automate the web browser, interacts with the Zoom website, and schedules the meetings.

The Flask app has two routes: '/' for the home page and '/predict' for making predictions. It loads a pre-trained ensemble model (ensemble_model.joblib) and a scaler (scaler.joblib) at the beginning.

The home route ('/') renders the index.html template.
The /predict route is triggered when the user submits the form on the home page.
In the predict route, it retrieves user input, scales the features using the loaded scaler, makes predictions using the loaded ensemble model, and renders the result on the home page.

3.3 Module 3: Web page

Module Name: **index.html**

Task & Function:
Creating Input Form:

Design an HTML form with input fields for user input.
Use a loop to dynamically create input fields for each feature.
Result Display:

Display the result of the prediction in a result box (if available).
Use conditional rendering based on the availability of the prediction result.
Styling:

Apply styling using inline CSS to define the layout, colors, and formatting.

4. METHODOLOGY

4.1 System Specification:

1) <u>Technical Details</u>

- Type: Ensemble Model (Voting Classifier)
- Algorithms: K-Nearest Neighbors (KNN), Support Vector Machine , Logistic Regression.
- Library: scikit-learn
- File: ensemble_model.joblib

2) <u>Web Application Requirements</u>

- Framework: Flask
- File: app.py
- HTML Template: index.html

3) **<u>Software Requirements</u>**

- Python: Version 3.x
- Flask : Web framework for serving the machine learning model.
- Scikit-learn : Machine learning library for model development.
- Joblib : Library for model and scaler persistence.

4) **<u>Minimum Hardware Requirements</u>**

- Processor : Dual-core processor with clock speed 2.4 GHz or higher
- Memory (RAM): 4 GB RAM
- Storage: 128 GB SSD or HDD for storing datasets, model files, and related resources

5) **<u>Model Dependencies</u>**

- Ensemble Model: Developed using scikit-learn.
- Algorithms: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression.
- Scaler: StandardScaler for feature scaling.

4.2 Data Flow Diagram:

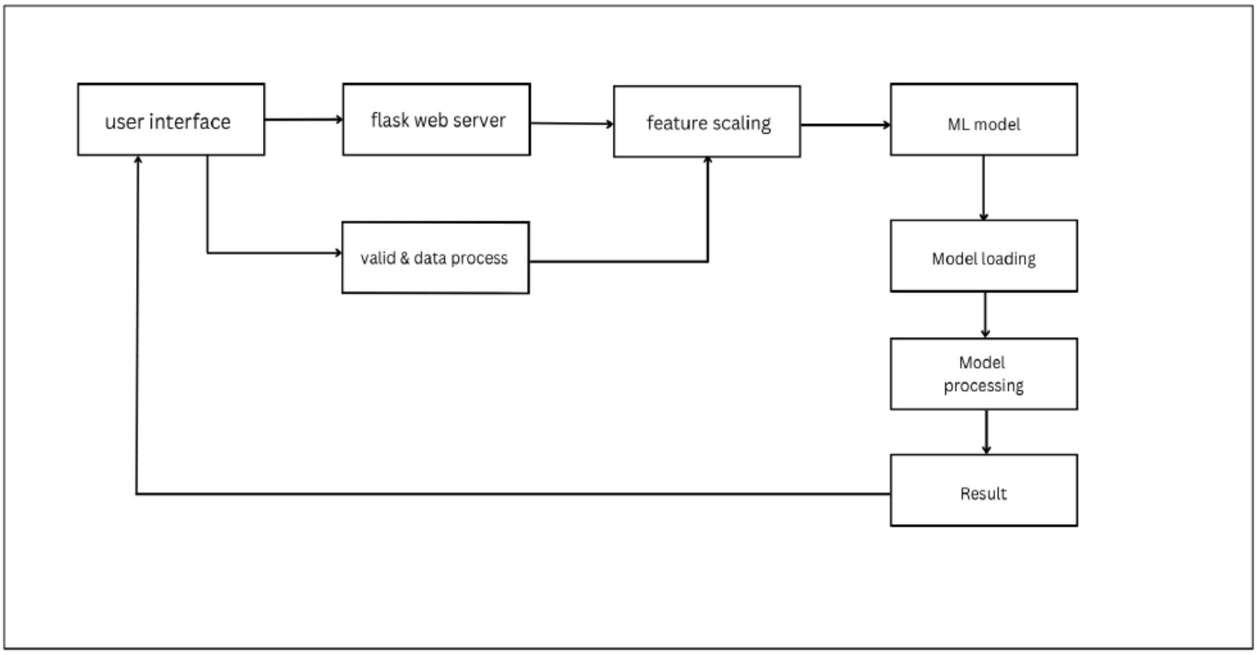


Figure 1: Data Flow Diagram

1. USER INTERFACE: Represent the web page where the user interacts with the model. The user enters information related to Breast Cancer Features.

2. FLASK WEB SERVER: A web server implemented using Flask, A web framework for Python. Handles requests from the user interface, performs data validation, and communicates with the Machine Learning Components.

3. Feature Scaling: Responsible for adjusting the scale of input features to ensure consistency with the scaling used during the training of the machine learning model.

4. Validation & Data Processing: Performs validation checks on user input, ensuring it meets the required criteria. It also processes the data before scaling, preparing it for input into the machine learning model.

5. Machine Learning Model: The core of the breast cancer detection application. Consists of two main sub-components:

a) Model Loading: Loads the pre-trained machine learning model (ensemble_model.joblib) and the scaler (scaler.joblib) into memory when the application starts.

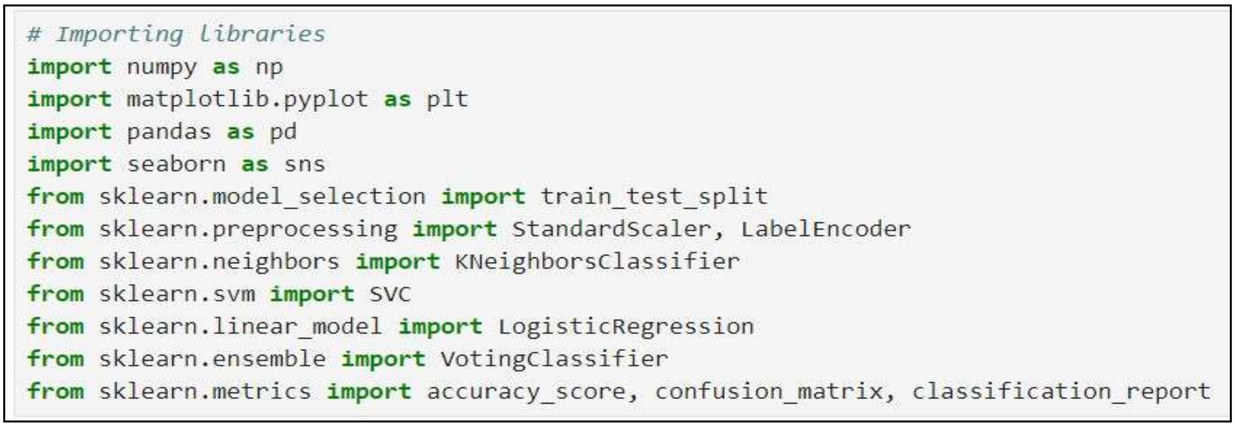
b) Model Processing: Internally processes the input data. This involves feature extraction, transformation, and ultimately, making a prediction regarding whether the breast cancer is malignant or benign.

6. Result: The web server updates the user interface with the prediction result, which is then displayed to the user.

4.3 Implementation

A) ML Model for The project

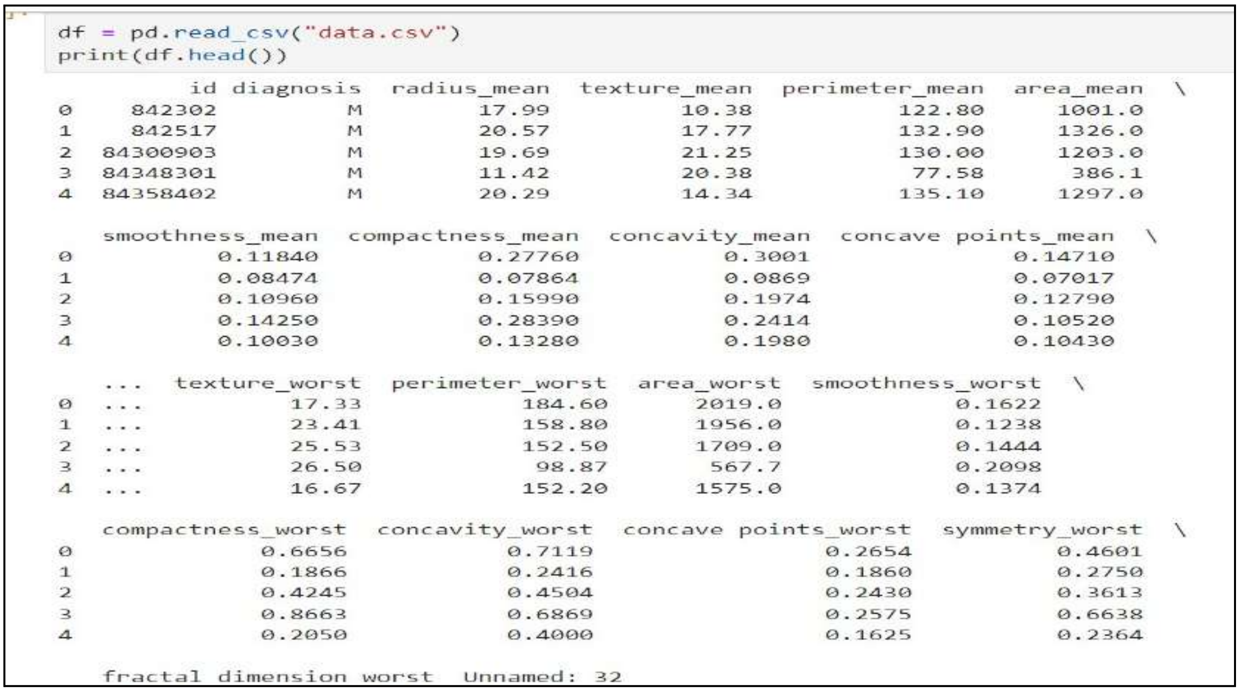
4.3.1 Imported Libraries:



```
# Importing Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

Figure 2. Imported Libraries

4.3.2 reading the data from the CSV file:

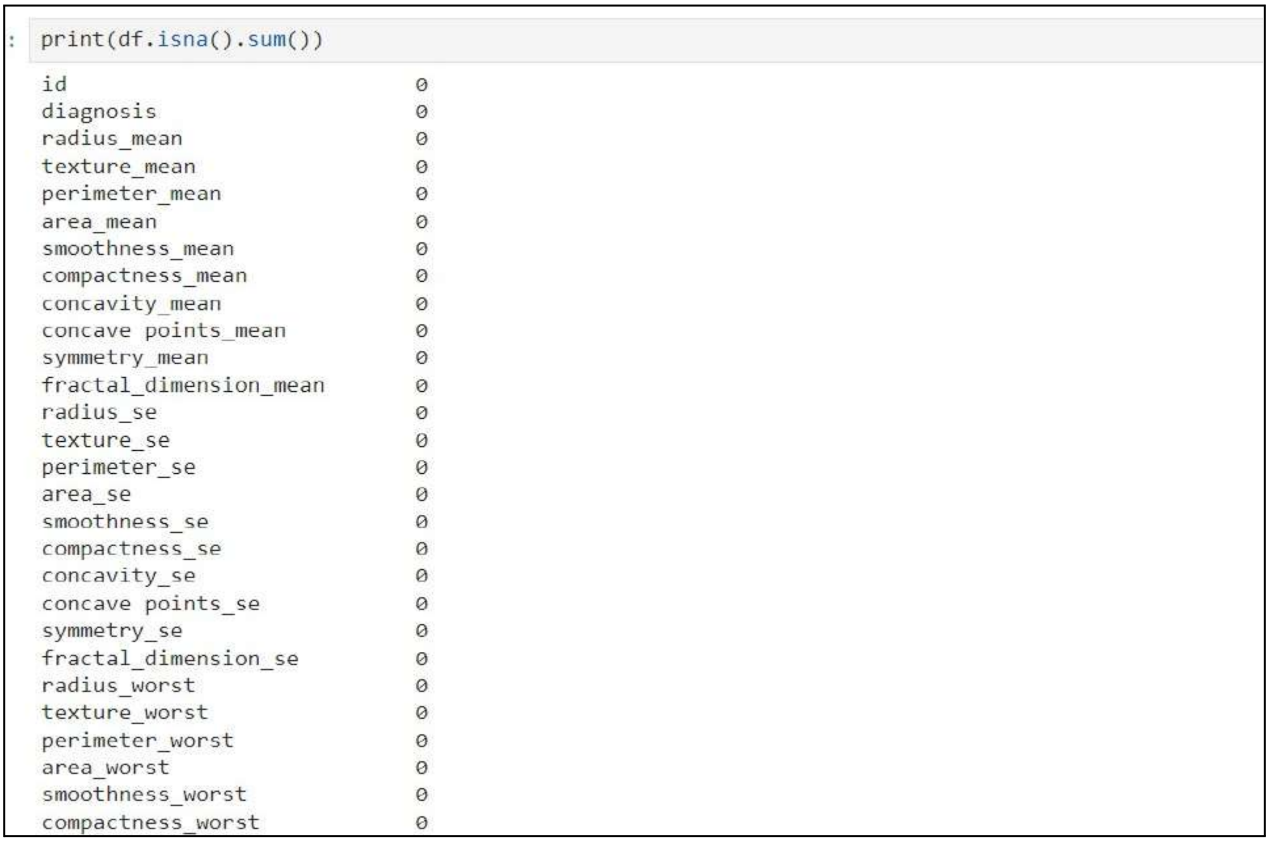


```
df = pd.read_csv("data.csv")
print(df.head())
```

```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  ...          17.33           184.60      2019.0            0.1622
1  ...          23.41           158.80      1956.0            0.1238
2  ...          25.53           152.50      1709.0            0.1444
3  ...          26.50            98.87       567.7            0.2098
4  ...          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
```

Figure 3. Reading the data from csv file

4.3.3 Display no. of non-null value:

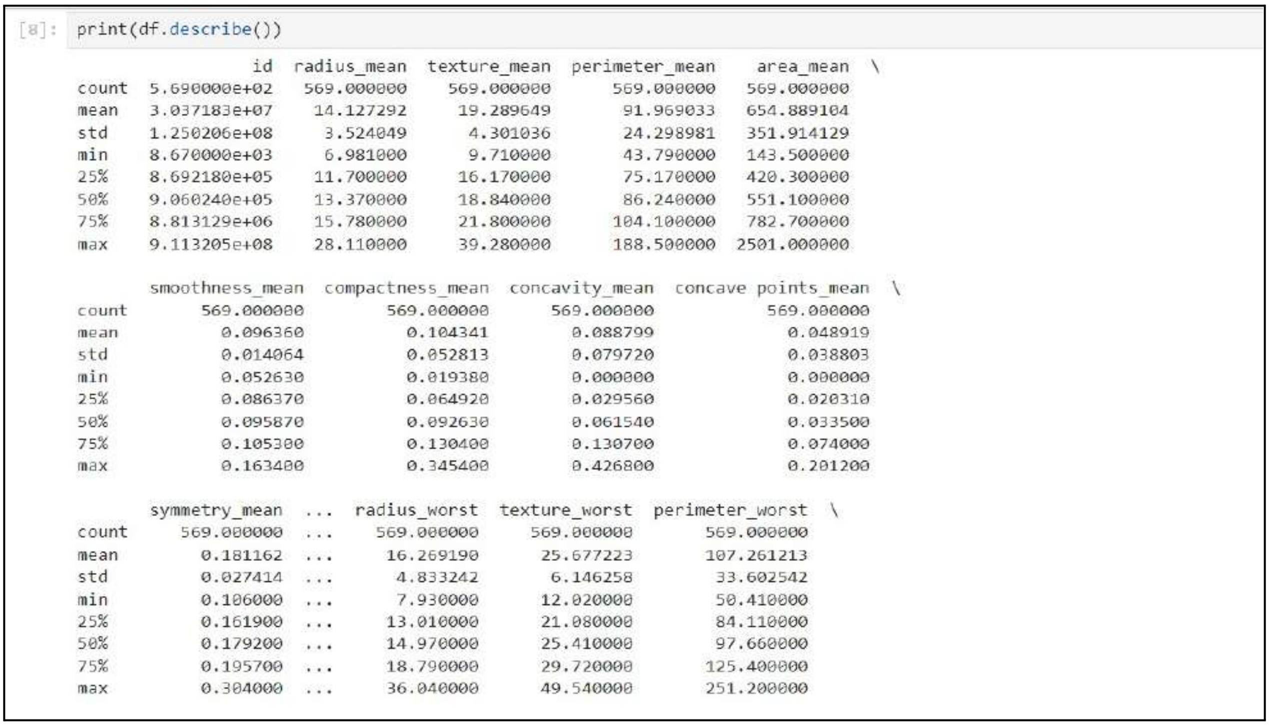


```
print(df.isna().sum())
```

```
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
```

Figure 4. non - null value

4.3.4 Description of data:

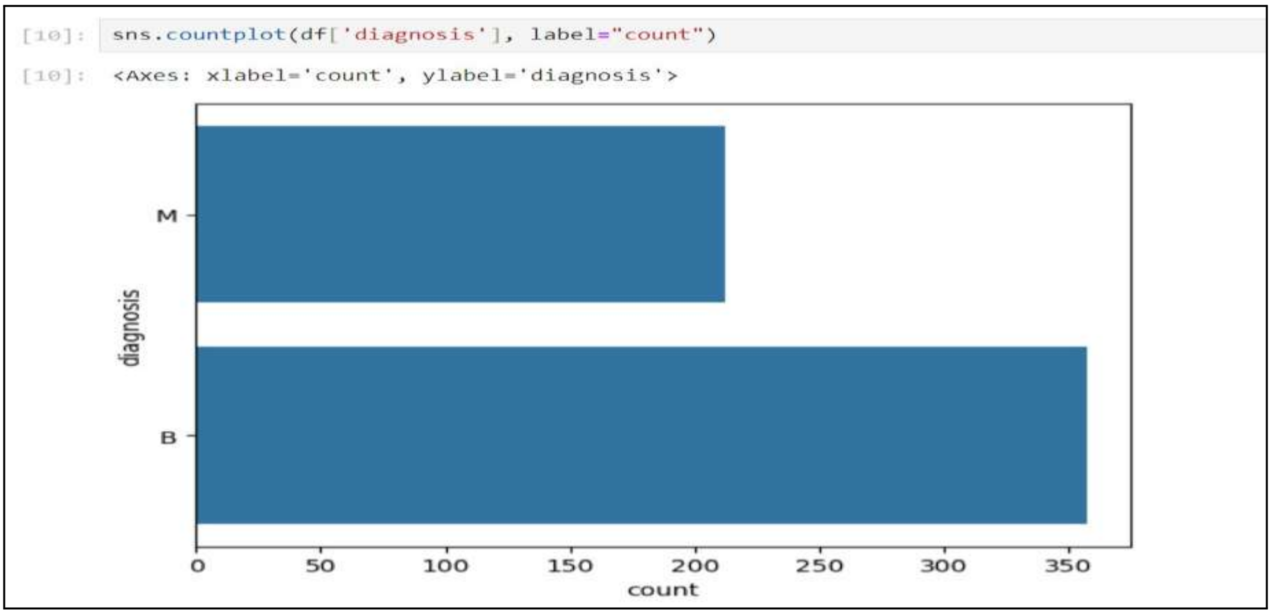


```
[8]: print(df.describe())
```

```
                 id  radius_mean  texture_mean  perimeter_mean    area_mean  \
count  5.690000e+02   569.000000    569.000000      569.000000   569.000000
mean   3.037183e+07    14.127292     19.289649       91.969033   654.889104
std    1.250206e+08     3.524049      4.301036       24.298981   351.914129
min    8.670000e+03     6.981000      9.710000       43.790000   143.500000
25%    8.692180e+05    11.700000     16.170000       75.170000   420.300000
50%    9.060240e+05    13.370000     18.840000       86.240000   551.100000
75%    8.813129e+06    15.780000     21.800000      104.100000   782.700000
max    9.113205e+08    28.110000     39.280000      188.500000  2501.000000

       smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       symmetry_mean  ...  radius_worst  texture_worst  perimeter_worst  \
count     569.000000  ...    569.000000     569.000000       569.000000
mean        0.181162  ...     16.269190      25.677223       107.261213
std         0.027414  ...      4.833242       6.146258        33.602542
min         0.106000  ...      7.930000      12.020000        50.410000
25%         0.161900  ...     13.010000      21.080000        84.110000
50%         0.179200  ...     14.970000      25.410000        97.660000
75%         0.195700  ...     18.790000      29.720000       125.400000
max         0.304000  ...     36.040000      49.540000       251.200000
```

Figure 5. Data output

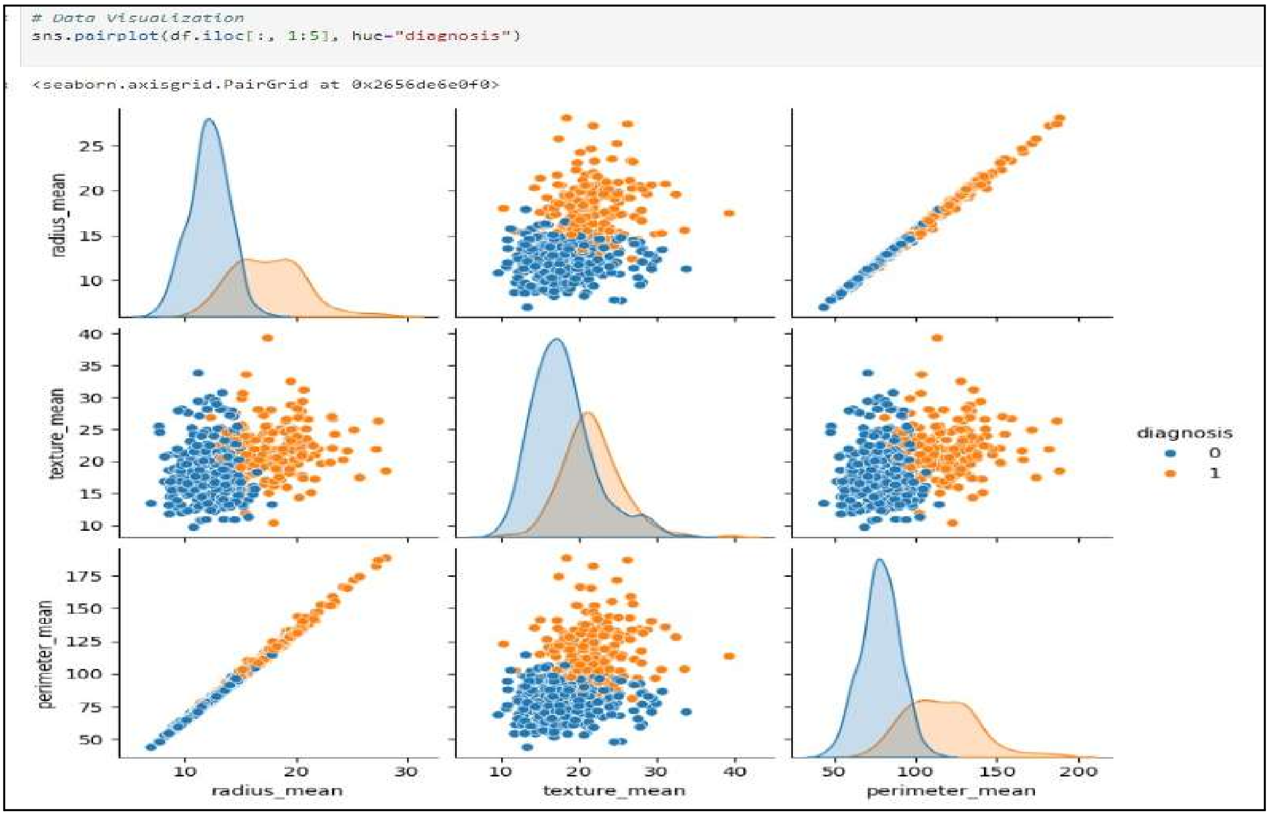
4.3.5 Count plot for the 'diagnosis' column in the DataFrame and visualization:



```
[10]: sns.countplot(df['diagnosis'], label="count")
```

```
[10]: <Axes: xlabel='count', ylabel='diagnosis'>
```

Figure 6. Count plot for the 'diagnosis' column



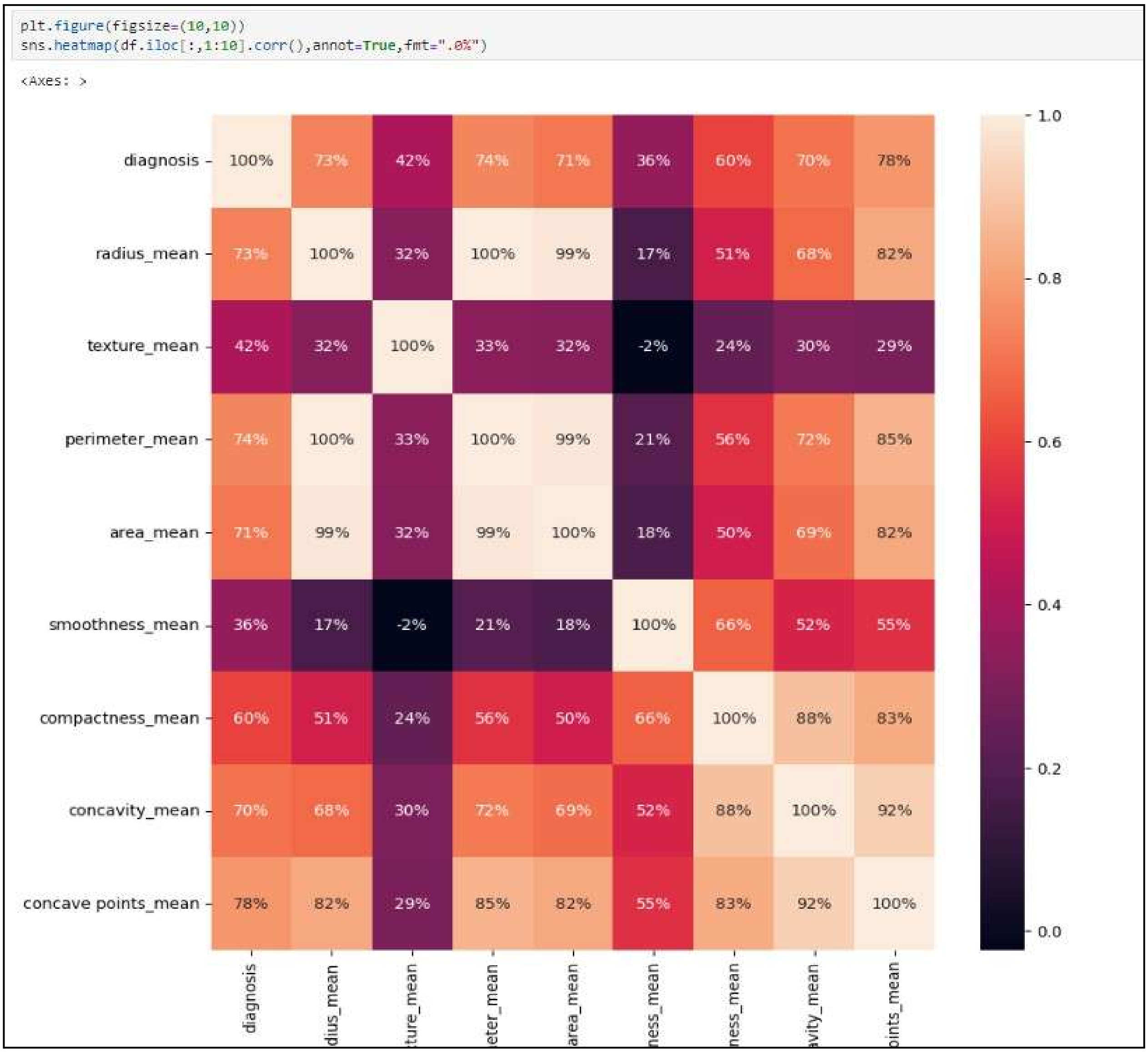
```
# Data Visualization
sns.pairplot(df.iloc[:, 1:5], hue="diagnosis")
```

```
<seaborn.axisgrid.PairGrid at 0x2656de6e0f0>
```

Figure 7: Data Visualization

The pair plot is used to visualize how different features in the dataset are distributed and whether there are any patterns or trends based on the diagnosis category. The colour differentiation provided by hue allows you to see if there's a visual separation between malignant (M) and benign (B) cases for different pairs of features.

4.3.6 Correlation Analysis:



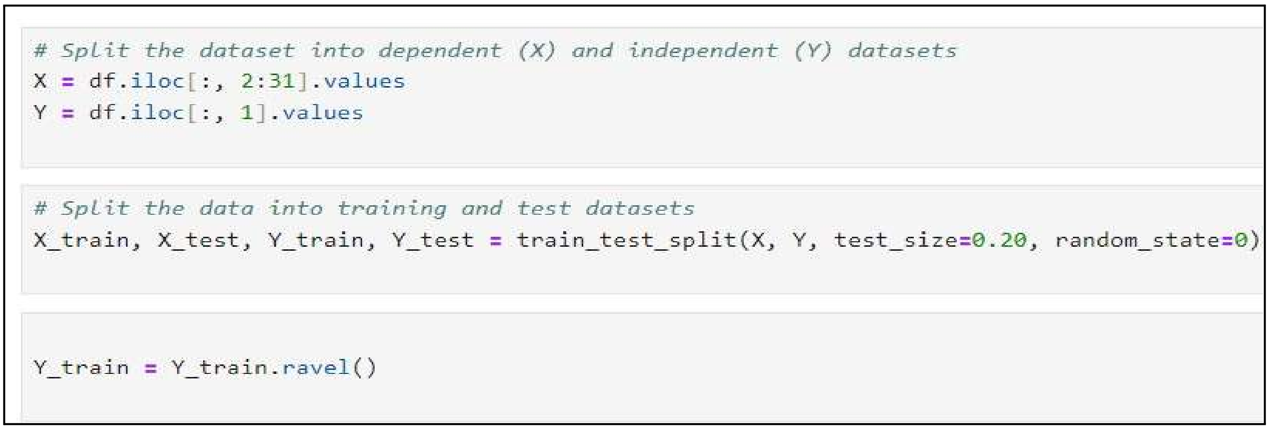
```
plt.figure(figsize=(10,10))
sns.heatmap(df.iloc[:,1:10].corr(),annot=True,fmt=".0%")
```

```
<Axes: >
```

	diagnosis	dius_mean	ture_mean	eter_mean	area_mean	ness_mean	ness_mean	avity_mean	oints_mean
diagnosis	100%	73%	42%	74%	71%	36%	60%	70%	78%
radius_mean	73%	100%	32%	100%	99%	17%	51%	68%	82%
texture_mean	42%	32%	100%	33%	32%	-2%	24%	30%	29%
perimeter_mean	74%	100%	33%	100%	99%	21%	56%	72%	85%
area_mean	71%	99%	32%	99%	100%	18%	50%	69%	82%
smoothness_mean	36%	17%	-2%	21%	18%	100%	66%	52%	55%
compactness_mean	60%	51%	24%	56%	50%	66%	100%	88%	83%
concavity_mean	70%	68%	30%	72%	69%	52%	88%	100%	92%
concave points_mean	78%	82%	29%	85%	82%	55%	83%	92%	100%

Figure 8 : Correlation Matrix

4.3.7 Splitting Data:



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

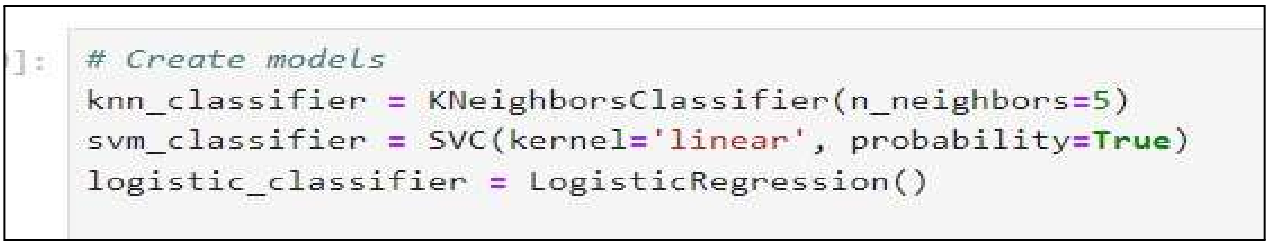
```
# Split the data into training and test datasets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=0)
```

```
Y_train = Y_train.ravel()
```

Figure 9: Splitting Data

X_train: The features for training the model.
X_test: The features for evaluating the model.
Y_train: The corresponding target values for training.
Y_test: The corresponding target values for evaluating the model.

4.3.8 Creating Model:



```
# Create models
knn_classifier = KNeighborsClassifier(n_neighbors=5)
svm_classifier = SVC(kernel='linear', probability=True)
logistic_classifier = LogisticRegression()
```

Figure 10: model creation

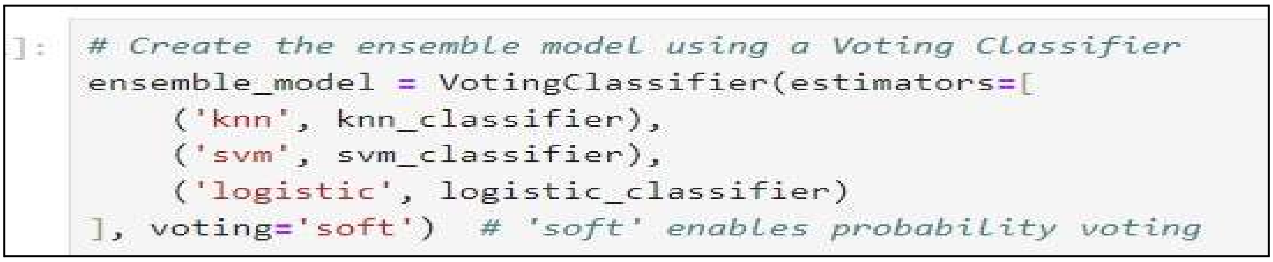
In this step, three different classifiers are instantiated:

K-Nearest Neighbors (k-NN):
knn_classifier is created using the KNeighborsClassifier class from scikit-learn.
The parameter n_neighbors=5 specifies that the model will consider the 5 nearest neighbors when making predictions.

Support Vector Machine (SVM):
svm_classifier is created using the SVC (Support Vector Classification) class from scikit-learn.
The kernel is set to linear (kernel='linear'), and probability estimates are enabled (probability=True).

Logistic Regression:
logistic_classifier is created using the LogisticRegression class from scikit-learn.

4.3.9 Creating Ensemble Model:

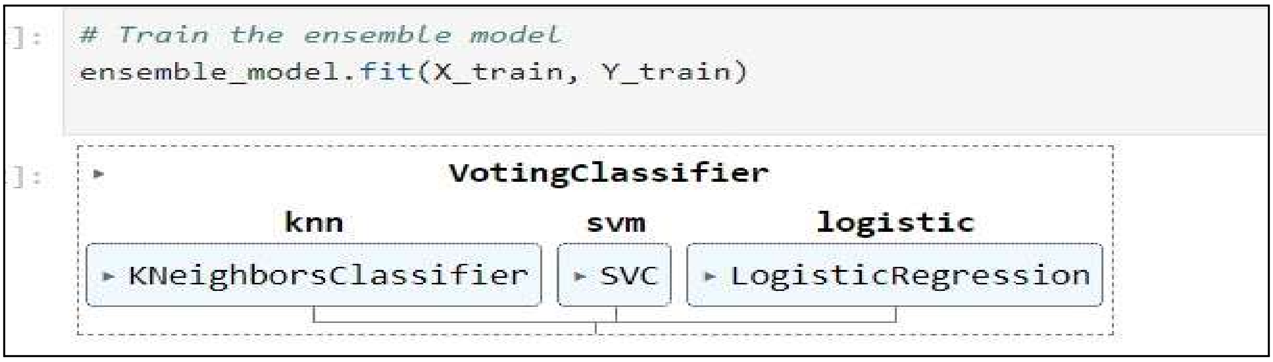


```
# Create the ensemble model using a Voting Classifier
ensemble_model = VotingClassifier(estimators=[
    ('knn', knn_classifier),
    ('svm', svm_classifier),
    ('logistic', logistic_classifier)
], voting='soft')  # 'soft' enables probability voting
```

Figure 11: Create Ensemble Model

Voting Classifier: VotingClassifier is an ensemble method in scikit-learn that combines the predictions from multiple machine learning algorithms. The estimators parameter takes a list of tuples where each tuple contains a name and a classifier.

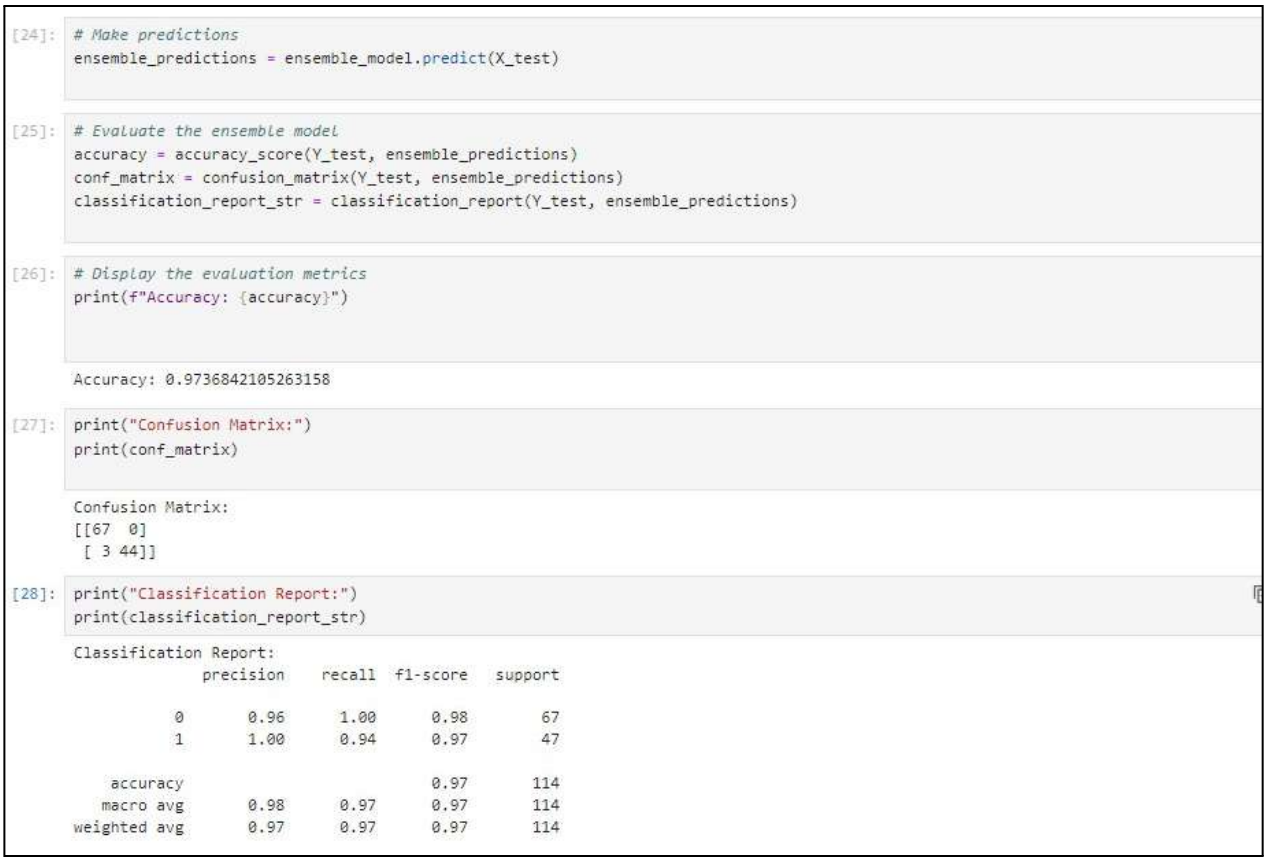
4.3.10 Train the Ensemble Model:



```
# Train the ensemble model
ensemble_model.fit(X_train, Y_train)
```

Figure 12: Train Ensemble Model

4.3.11 Evaluating the ensemble model:



```
# Make predictions
ensemble_predictions = ensemble_model.predict(X_test)
```

```
# Evaluate the ensemble model
accuracy = accuracy_score(Y_test, ensemble_predictions)
conf_matrix = confusion_matrix(Y_test, ensemble_predictions)
classification_report_str = classification_report(Y_test, ensemble_predictions)
```

```
# Display the evaluation metrics
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.9736842105263158
```

```
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[67  0]
 [ 3 44]]
```

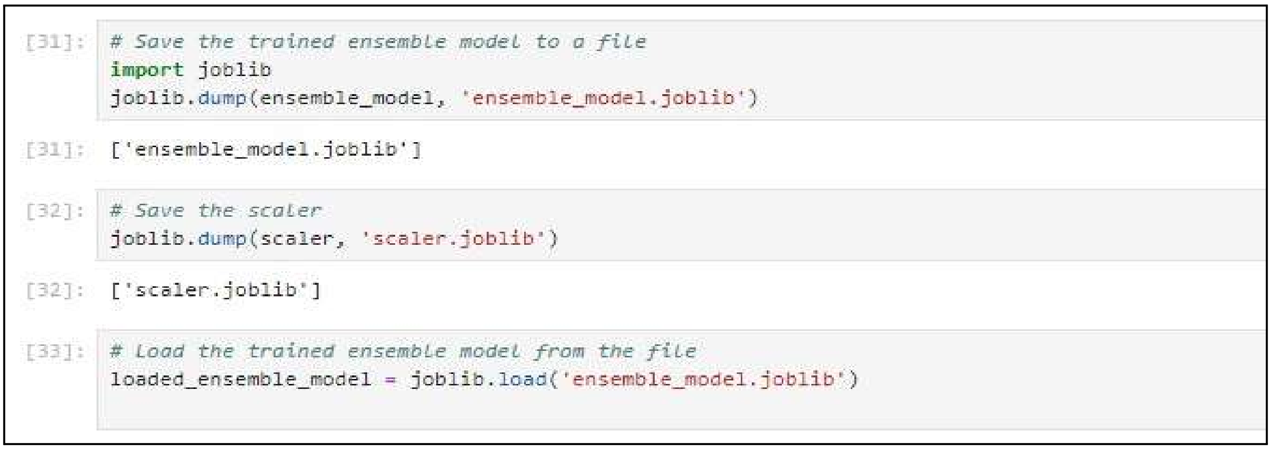
```
print("Classification Report:")
print(classification_report_str)
```

```
Classification Report:
              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
```

Figure 13: Evaluating ensemble model

The model's performance is evaluated using standard metrics like accuracy, confusion matrix, and classification report based on the predictions made on the test set (X_test).
Here, the accuracy of our model is about 97 %.

4.3.12 Save the Model and Scale:



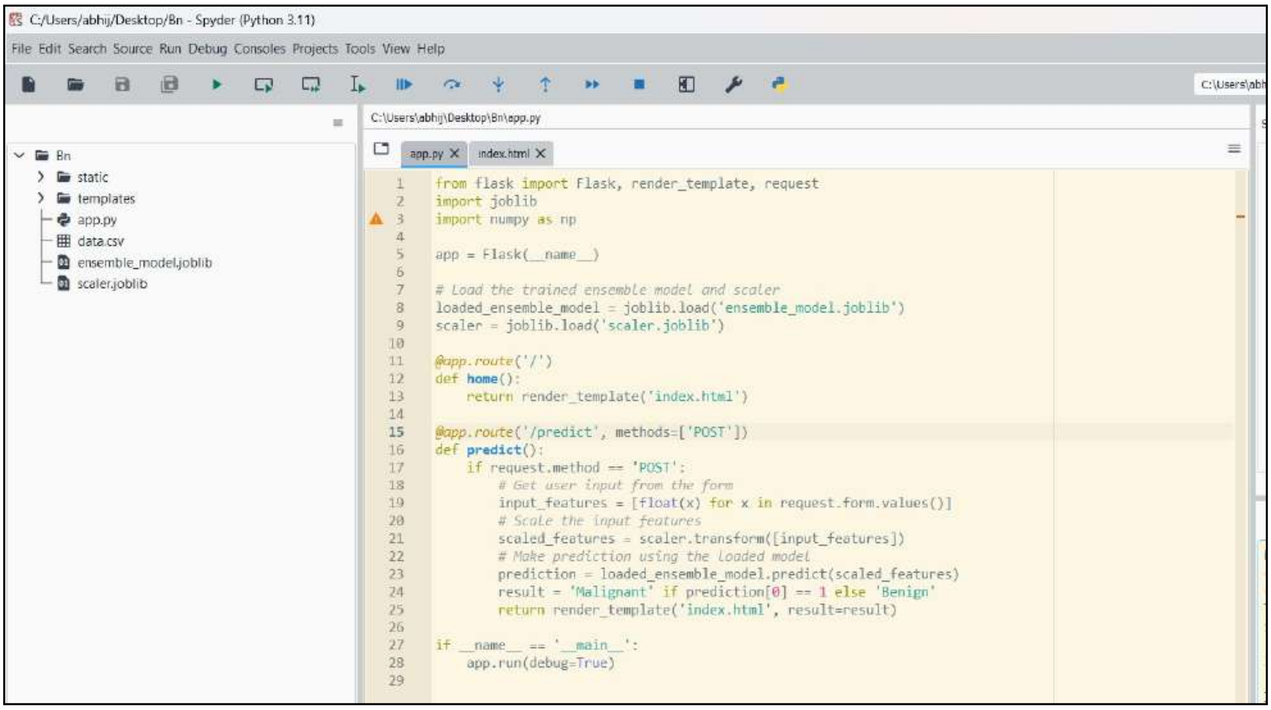
```
[31]: # Save the trained ensemble model to a file
      import joblib
      joblib.dump(ensemble_model, 'ensemble_model.joblib')

[31]: ['ensemble_model.joblib']

[32]: # Save the scaler
      joblib.dump(scaler, 'scaler.joblib')

[32]: ['scaler.joblib']

[33]: # Load the trained ensemble model from the file
      loaded_ensemble_model = joblib.load('ensemble_model.joblib')
```

Figure 14: model data save & scaling the data

joblib.dump(ensemble_model, 'ensemble_model.joblib'): Saves your trained ensemble model to a file named 'ensemble_model.joblib'.

joblib.dump(scaler, 'scaler.joblib'): Saves the scaler you used for data scaling to a file named 'scaler.joblib'.

4.4 Deployment machine learning model:


C:/Users/abhij/Desktop/Bn - Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\abh
C:\Users\abhij\Desktop\Bn\app.py
app.py
index.html
Bn
static
templates
app.py
data.csv
ensemble_model.joblib
scaler.joblib

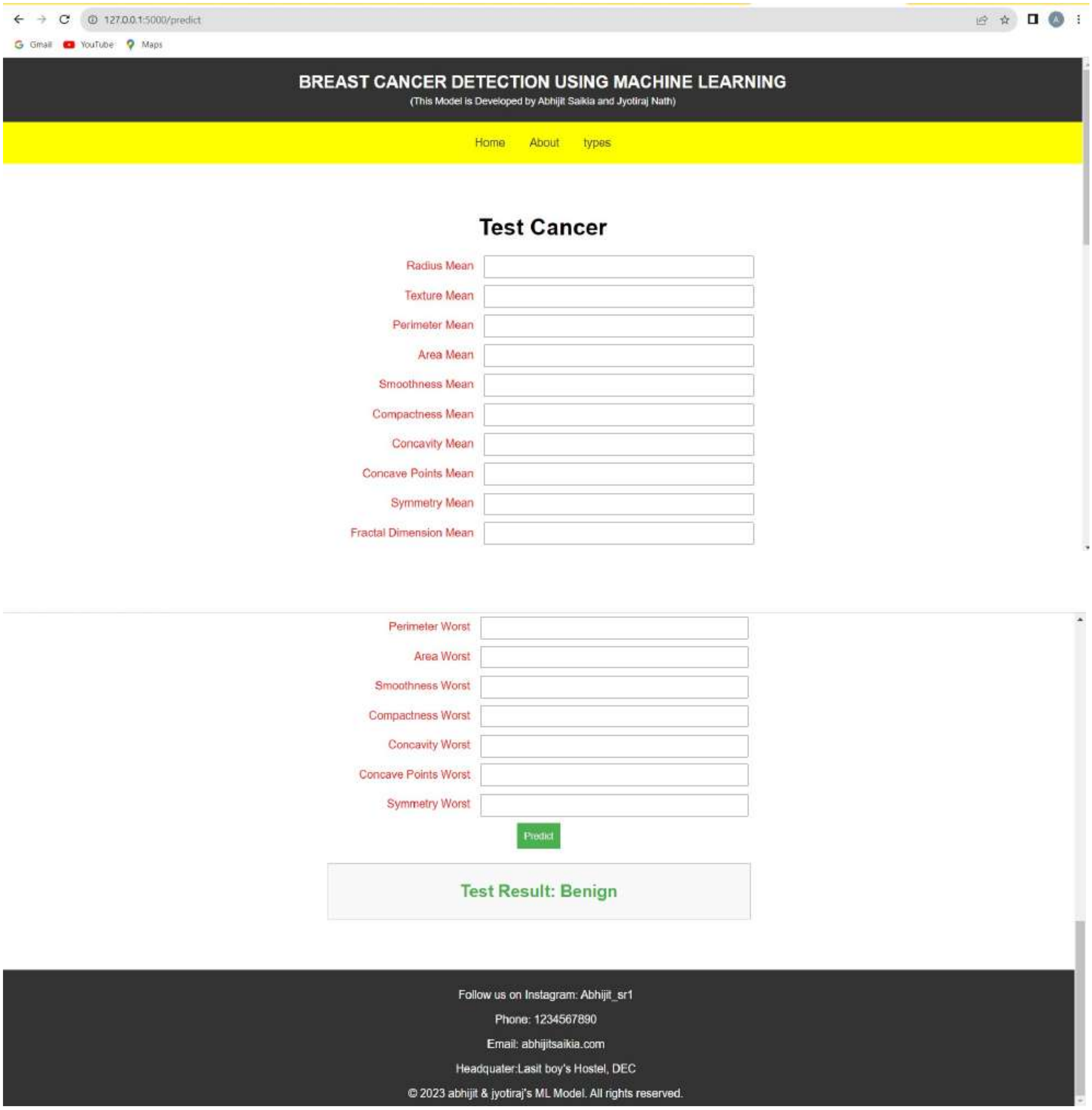
```
from flask import Flask, render_template, request
import joblib
import numpy as np

app = Flask(__name__)

# Load the trained ensemble model and scaler
loaded_ensemble_model = joblib.load('ensemble_model.joblib')
scaler = joblib.load('scaler.joblib')

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

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        # Get user input from the form
        input_features = [float(x) for x in request.form.values()]
        # Scale the input features
        scaled_features = scaler.transform([input_features])
        # Make prediction using the loaded model
        prediction = loaded_ensemble_model.predict(scaled_features)
        result = 'Malignant' if prediction[0] == 1 else 'Benign'
        return render_template('index.html', result=result)

if __name__ == '__main__':
    app.run(debug=True)
```

Figure 15: app.py file

@app.route('/predict', methods=['POST']): This decorator defines the '/predict' route, which handles form submissions (POST requests).

It gets user input from the HTML form.
Scales the input features using the loaded scaler.
Makes a prediction using the pre-trained ensemble model.
Renders the 'index.html' template with the prediction result.

4.5 Output of the Model:



BREAST CANCER DETECTION USING MACHINE LEARNING
(This Model is Developed by Abhijit Saikia and Jyotiraj Nath)

Test Cancer

Test Result: Benign

Follow us on Instagram: Abhijit_sr1
Phone: 1234567890
Email: abhijitsaikia.com
Headquater:Lasit boy's Hostel, DEC
© 2023 abhijit & jyotiraj's ML Model. All rights reserved.

5. RESULT & DISCUSSION

In the "Breast Cancer Prediction using Machine Learning" project, the model achieved an impressive accuracy of 97% when tested on the held-out 20% of the data. This high accuracy indicates the effectiveness of the machine learning model in correctly classifying instances into benign and malignant categories. Additional metrics such as precision, recall, and F1 score, as well as the area under the ROC curve, may have been calculated to provide a more nuanced understanding of the model's performance. These metrics help evaluate the trade-off between false positives and false negatives, providing a clearer picture of the model's precision and sensitivity.

The discussion surrounding the results would likely emphasize the significance of such a high accuracy in breast cancer prediction. Achieving a 97% accuracy suggests that the model can effectively distinguish between benign and malignant cases, potentially contributing to early detection and improved patient outcomes. However, it is essential to consider the context of these results and understand potential challenges and limitations.

The discussion may delve into the robustness of the model across different demographic groups, the potential biases in the training data, and the implications of false positives and false negatives. Ethical considerations, such as patient privacy and the responsible use of AI in healthcare, should be addressed. Furthermore, the feasibility of implementing the model in real-world clinical settings and collaboration with healthcare professionals for validation and integration into existing workflows would be important points of discussion.

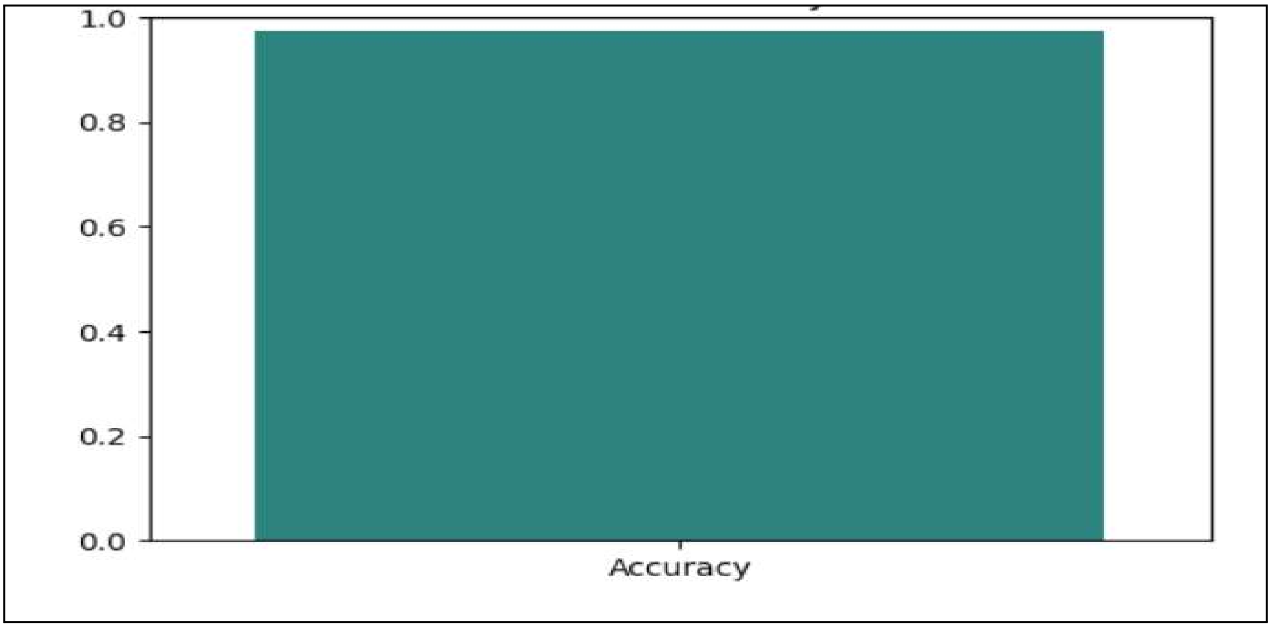


Figure 16 : Model Accuracy

6. CONCLUSION & FUTURE WORK

Breast Cancer represents one of the diseases that causes the highest number of deaths every year. Breast Cancer data is processed using the Standard Scaler module and feature selection is performed using Python’s scikit-learn package. The models were developed using multimodel sets of machine learning algorithms, K Nearest Neighbor, SVM and logistic regression. The study used a confusion matrix to compare anticipated outcomes with actual numbers and assessed performance metrics such as accuracy, area under the precision, recall, sensitivity, and f1-score. The results were summarized and compared using exploratory data analysis. The study found that maximum area worst and maximum area_mean values decreased after processing, potentially leading to false positives. The correlation between variables in breast cancer diagnosis is crucial for understanding the relationship between features and patient outlook. Logistic regression and SVC have similar performance in predicting target variables. Breast cancer is a prevalent disease affecting women worldwide, with machine-learning approaches potentially impacting early detection and prognosis. Early detection is crucial for successful treatment, and appropriate screening technologies are essential.

Future research on breast cancer diagnosis using ML might explore these and other possibilities. To make substantial strides forward in the detection and treatment of breast cancer, continued research and cooperation between data scientists, medical experts, and researchers is essential.

REFERENCES

1. https://www.atlantis-press.com/article/125960864.pdf
2. https://scholarworks.calstate.edu/downloads/tx31qq02n
3. https://www.mdpi.com/2075-4418/13/19/3113
4. https://www.kaggle.com/
5. https://www.javatpoint.com/
6. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8966510/