BREAST CANCER CLASSIFICATION

PROJECT CODE: OP03

TEAM NUMBER: 5

STUDENTS DETAILS

1. Students Names

Sneha Das

Debolina Chakraborty

Anandi Roy Chowdhury

Devanshi Bhattacharjee

2. Course and Batch

B.Sc. (Hons) Computer Science

Batch: 2023-27

3. Institute Name

St. Xavier's College (Autonomous), Kolkata

Project Guide/Mentor

Dr. Snehalika Lall

Report Submitted To

IDEAS - Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata.

1. ABSTRACT

Breast cancer is one of the most life-threatening diseases among women worldwide. Thus, the early detection and accurate diagnosis of breast cancer is very crucial. Late detection or wrong diagnosis of tumors have caused significant number of deaths each year. Fortunately, the introduction of artificial intelligence has made diagnosis more speedy, effective and reliable.

Thereby, with the help of **machine learning**, our study is mainly aimed at developing a more **accurate and speedy way of diagnosing and prognosing breast cancer**.

2.INTRODUCTION

Breast cancer is a major health concern, making early and accurate diagnosis and prognosis essential. This project presents two models among which one classifies breast tumors as **benign** or **malignant**, while the other predicts whether the disease is **Recurrent** or **Non-Recurrent**, helping to support timely clinical decisions. Using the **Wisconsin Breast Cancer Diagnostic Dataset** and the **Wisconsin Breast Cancer Prognostic Dataset** from UCI Machine Learning Repository, supervised machine learning algorithms like **Logistic Regression, Random Forest, Extreme Gradient Boosting (XGBoost)** and **Support Vector Machine (SVM)** were applied.

The process includes **data cleaning and preprocessing**, **visualizing trends of target variable against various features**, **model training**, and **evaluation using performance metrics**. A background review of similar studies guides model selection and methodology. The project highlights machine learning's relevance and effectiveness in enhancing medical decision-making.

This project delves into data analysis to uncover meaningful patterns in breast cancer diagnosis and prognosis. By exploring relationships among **clinical features**, we aim to enhance understanding of **tumor behavior**. The insights gained serve as a foundation for building predictive models, contributing to more accurate and timely breast cancer detection. Ultimately, the project aspires to support early detection, enhance diagnostic as well as prognostic precision and contribute to more informed clinical decision-making in the fight against breast cancer.

3.PROJECT OBJECTIVE

This project aims to classify breast cancer tumors and demonstrate machine learning's role in accurate and early diagnosis and prognosis. The objective of our project focuses on:

- Building a machine learning model that classifies breast tumors as **benign** or **malignant**.
- Building a machine learning model that determines whether the disease is **Recurrent** or **Non-Recurrent.**
- Comparing performance of classification models using **Accuracy, Precision**, **Recall, F1-score, and Confusion Matrix and determining the optimal algorithm**.
- Showing how machine learning aids **early and accurate cancer detection**.

4.TEAM ROLES AND RESPONSIBILITIES

NAME	*ROLE(S)*	*RESPONSIBILITIES*
Debolina Chakraborty	Team Lead, Model Developer, Report Writer	Coordinate team activities, organize meetings, track progress, develop the diagnostic model, prepare project report
Sneha Das	Data Preprocessor, Presentation Developer	Clean, preprocess and visualize data, prepare project presentation
Anandi Roy Chowdhury	Model Developer, Report Writer	Develop the prognostic model, prepare project report
Devanshi Bhattacharjee	Model Tester, Presentation Developer	Evaluate the diagnostic and prognostic models, prepare project presentation

WORK TIMELINE

WEEK / TASK	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
PROJECT PLANNING	■				
DATA COLLECTION		■			
DATA PREPROCESSING		■			
DATA ANALYSIS		■			
MODEL TRAINING AND TUNING			■	■	
MODEL EVALUATION			■	■	
REPORT WRITING			■	■	■
PRESENTATION PREPARATION				■	■
FINAL SUBMISSION					■

Legend: ■ **=** Task Performed

5.METHODOLOGY

This project performs **Exploratory Data Analysis** to understand the nature of features, identify outliers, and discover hidden patterns among different data points which help in **choosing optimal modelling techniques**.

The various steps carried out for the fulfilment of this project are as follows:

- **Importing** the appropriate breast cancer classification datasets from **UCI Machine Learning Repository**.
- **Cleaning the data** by handling null values and outliers.
- **Preprocessing data** so that it is in usable form.
- **Visualizing data** to uncover hidden patterns.
- **Dividing the data** into train set and test set.
- **Handling imbalance** in training data.
- **Normalizing** the training and testing sets.
- **Training binary classification models** on appropriate classification algorithms and hyper-tuning the parameters.
- **Evaluating the models** by observing performance metrics like accuracy, precision, recall, F1-score and confusion matrix.
- **Improving accuracy** as much as possible.

DIAGNOSTIC MODEL

5.1 Cleaning and Preprocessing the data

5.1.1 Handling Null Values

No null values were present in the dataset.

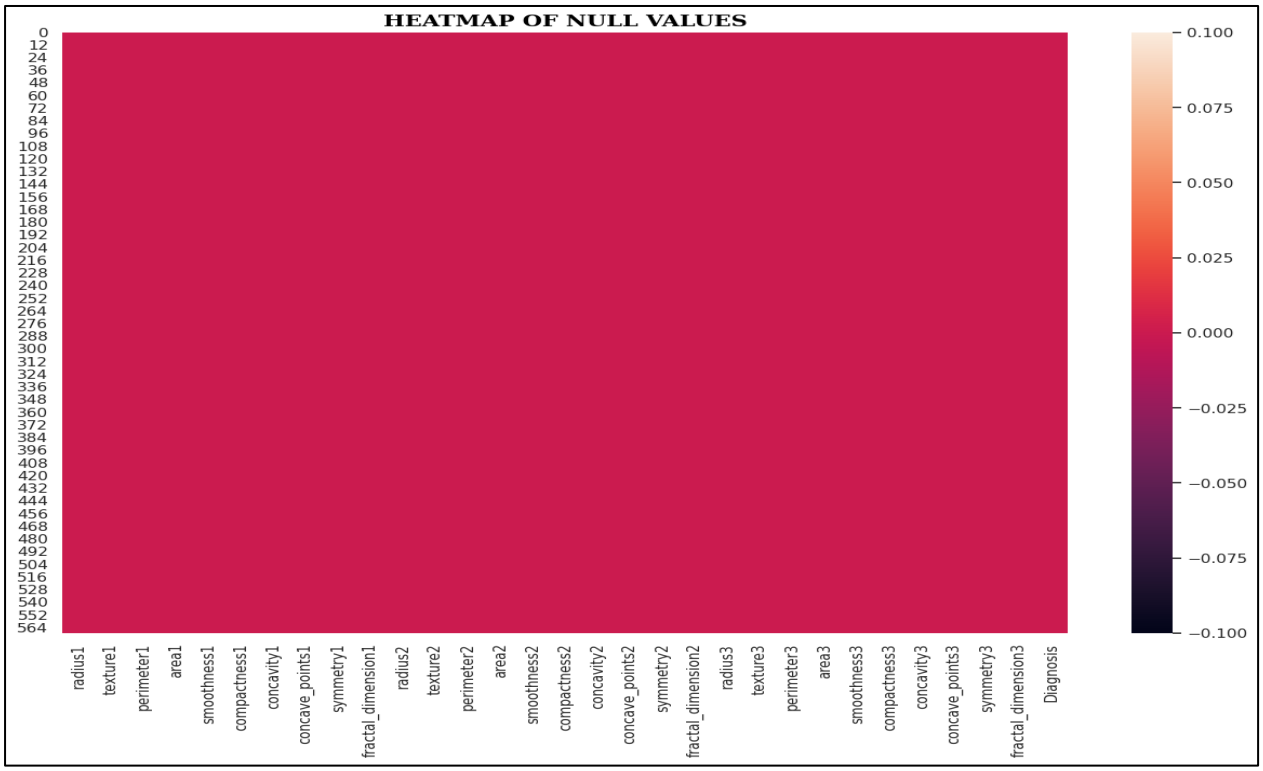

Fig 1.1

5.1.2 Visualizing Outliers

Outliers are present in most of the features. These outliers are medically valid and modifying them may cause fatal effects on accurate diagnosis. Therefore, the outliers are retained.

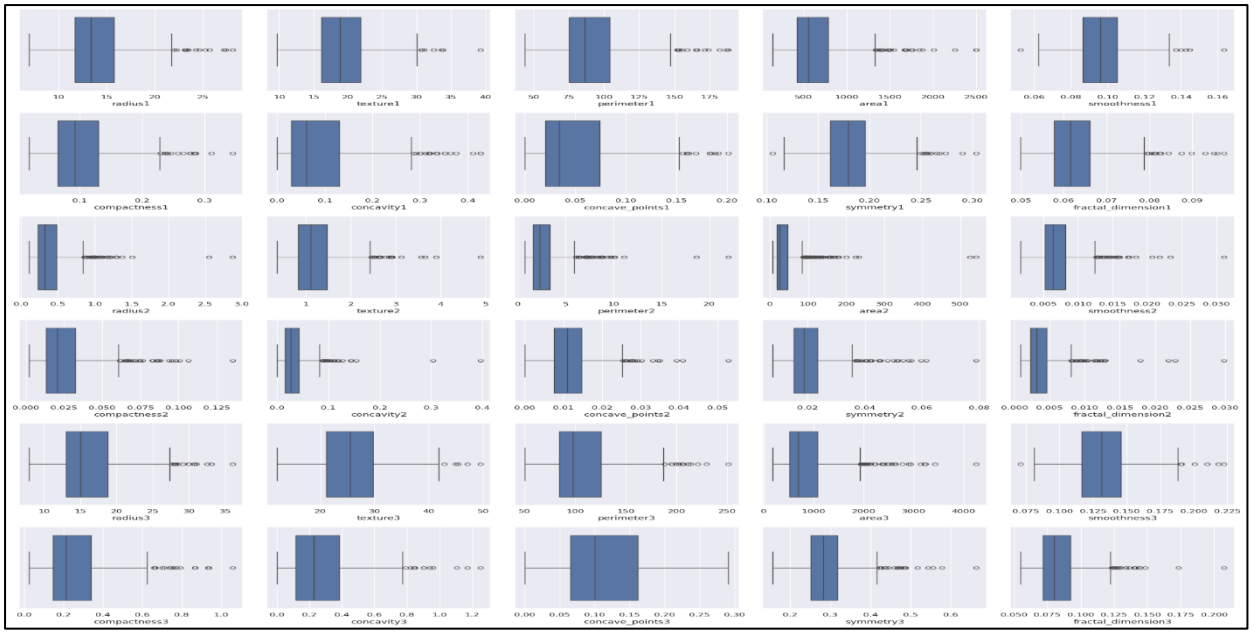

Fig 1.2

5.1.3 Visualizing Frequency Distribution of Different Features

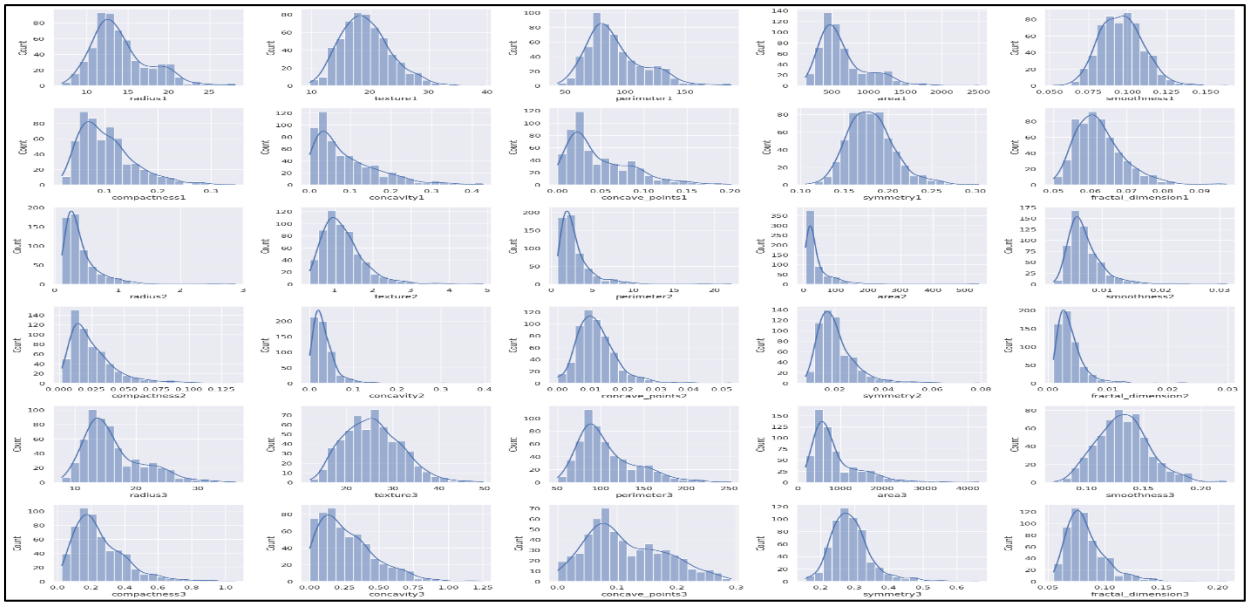

Fig 1.3

5.1.4 Label Encoding the Target Variable (Diagnosis)

Label encoded the target variable as:

1 = Malignant (M)

0 = Benign (B)

using LabelEncoder() from sklearn.

5.1.5 Visualizing Data

Plots like **Heatmap** for correlation matrix and **Pairplot** for discovering diagnosis patterns among all the features were used.

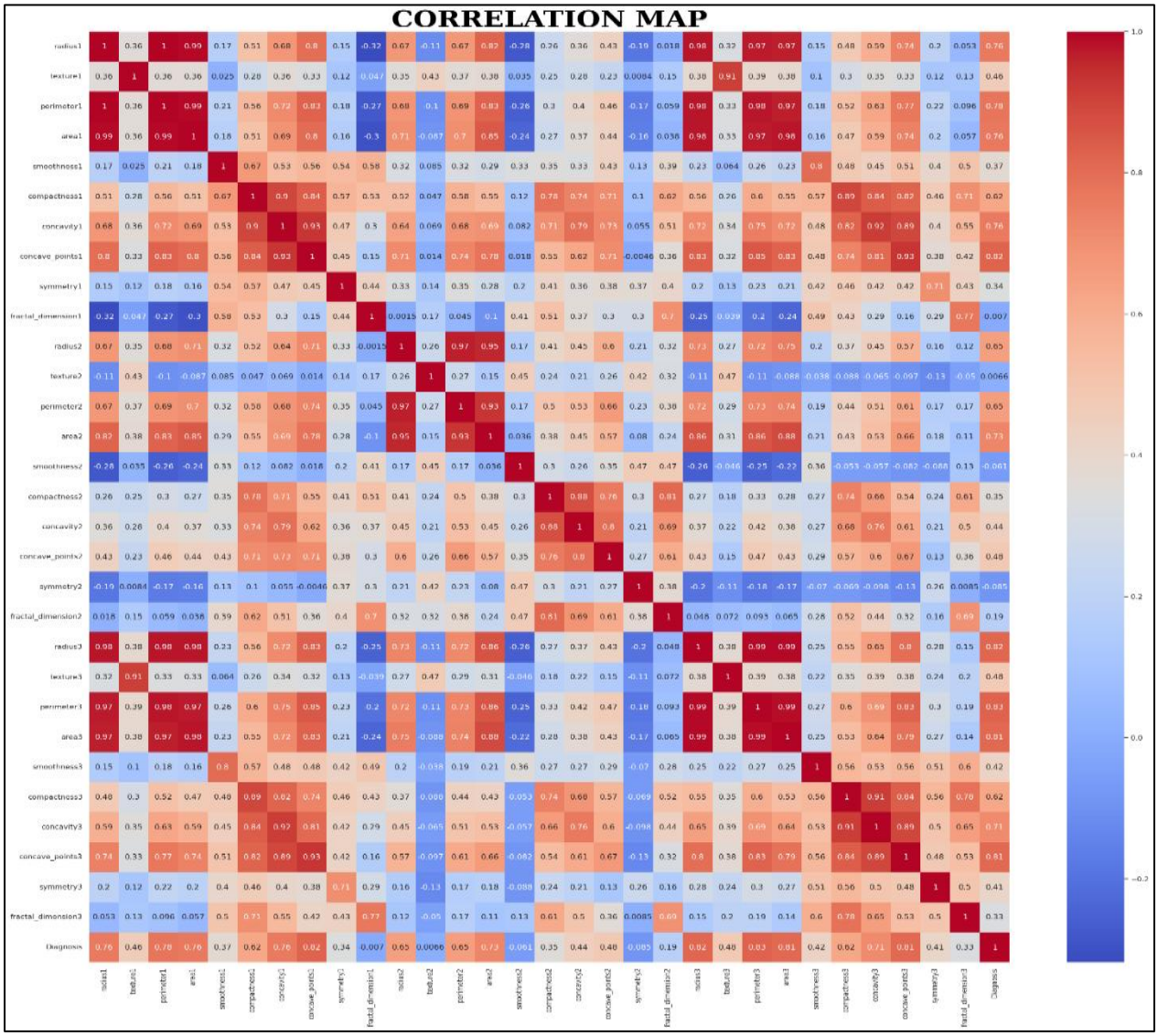

Fig 1.4

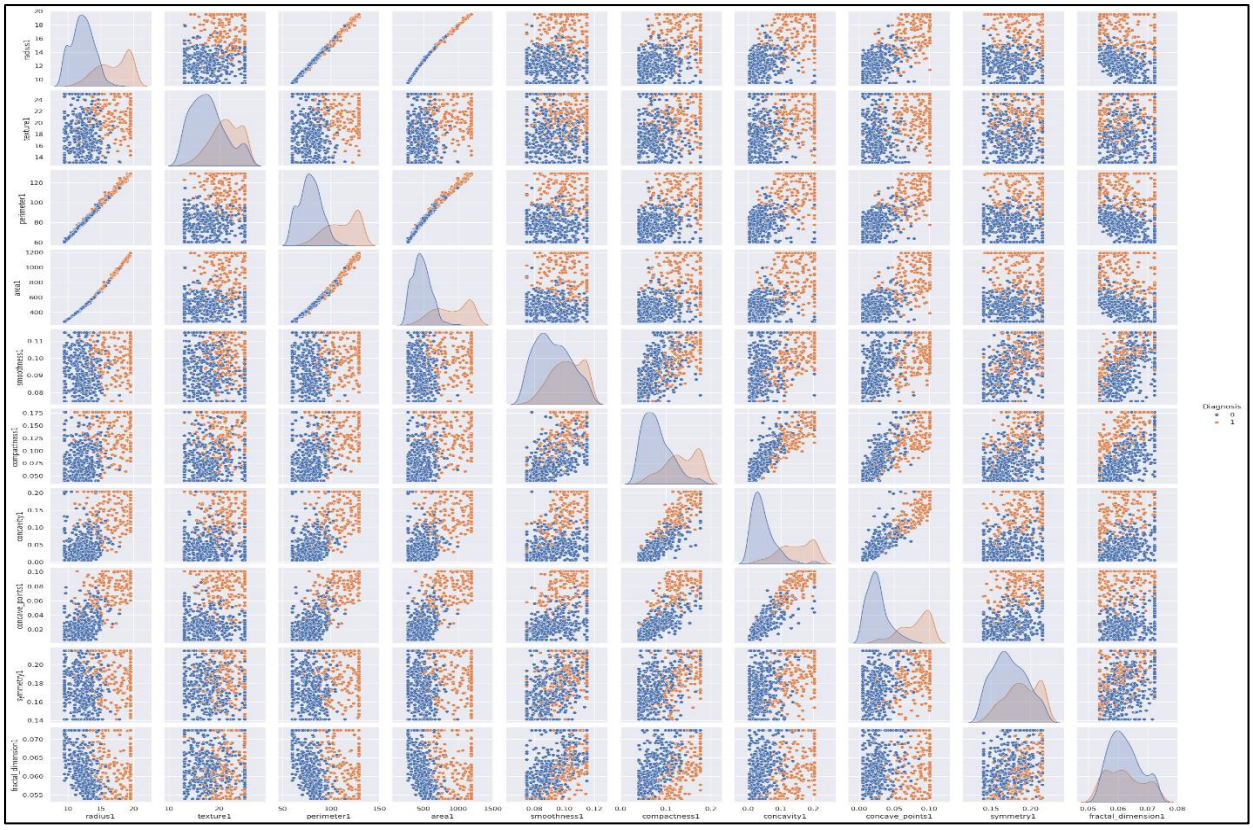
radius1
texture1
perimeter1
area1
smoothness1
compactness1
concavity1
concave_points1
symmetry1
fractal_dimension1
Diagnosis
0
1

Fig 1.5

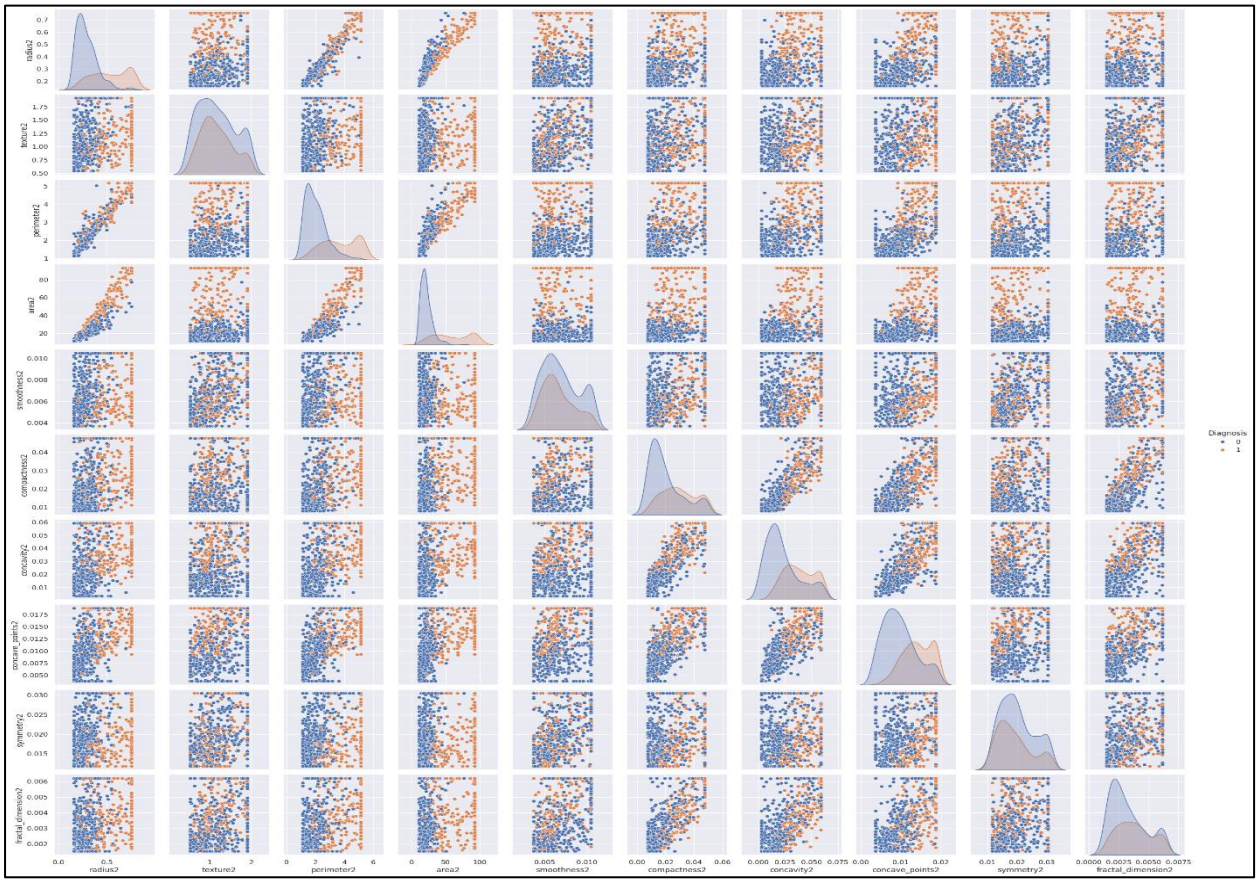
radius2
texture2
perimeter2
area2
smoothness2
compactness2
concavity2
concave_points2
symmetry2
fractal_dimension2
Diagnosis
0
1

Fig 1.6

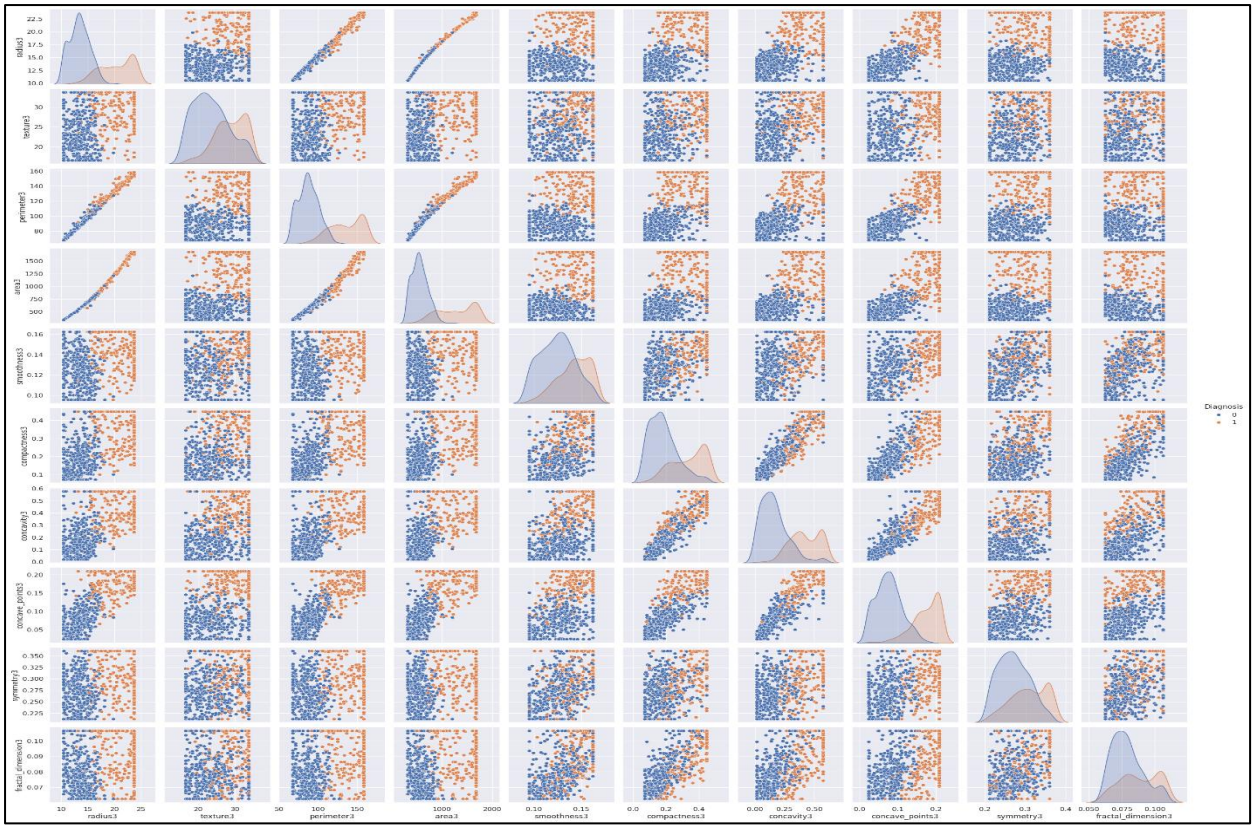

Fig 1.7

5.1.6 Dividing the Data into Training Set and Testing Set

The data was divided into **80 percent training data** and **20 percent testing data.**

5.1.7 Handling Imbalance in Training Data using Synthetic Minority Over-sampling Technique (SMOTE)

Before SMOTE

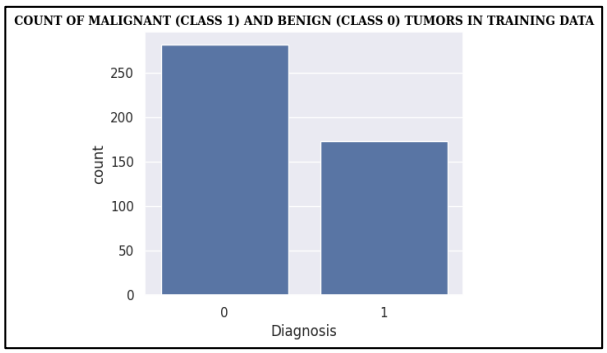

Fig 1.8

After SMOTE

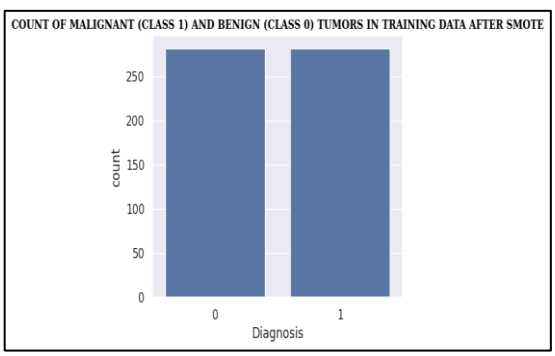

Fig 1.9

5.1.8 Normalizing the data using StandardScaler() of sklearn

5.2 Training and Evaluating the Model

The model was trained on a variety of classification algorithms and the parameters were hyper-tuned using **GridSearchCV**.

5.2.1 Logistic Regression

Parameters of the best fit model:

penalty='l2', dual=False, tol=1e-4, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Training:

Accuracy: 0.9858156028368794 ≈ 98.6 %

Confusion Matrix: [[281 1]
[7 275]]

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	282
1	1.00	0.98	0.99	282
accuracy			0.99	564
macro avg	0.99	0.99	0.99	564
weighted avg	0.99	0.99	0.99	564

Validation:

Accuracy: 0.9751738305941846 ≈ 97.5 %

Testing:

Accuracy: 0.9824561403508771 ≈ 98.2 %

Confusion Matrix: [[74 1]
[1 38]]

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	75
1	0.97	0.97	0.97	39
accuracy			0.98	114
macro avg	0.98	0.98	0.98	114
weighted avg	0.98	0.98	0.98	114

5.2.2 Random Forest

Parameters of the best fit model:

n_estimators=40, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=2, min_weight_fraction_leaf=0.0, max_features='log2', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None, monotonic_cst=None

Training:

Accuracy: 0.9875886524822695 ≈ 98.8 %

Confusion Matrix: [[281 1]
[6 276]]

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	282
1	1.00	0.98	0.99	282
accuracy			0.99	564
macro avg	0.99	0.99	0.99	564
weighted avg	0.99	0.99	0.99	564

Validation:

Accuracy: 0.9663242730720608 ≈ 96.6 %

Testing:

Accuracy: 0.9824561403508771 ≈ 98.2 %

Confusion Matrix: [[74 1]
[1 38]]

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	75
1	0.97	0.97	0.97	39
accuracy			0.98	114
macro avg	0.98	0.98	0.98	114
weighted avg	0.98	0.98	0.98	114

5.2.3 Extreme Gradient Boosting (XGBoost)

Parameters of the best fit model:

base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, device=None,
early_stopping_rounds=None, enable_categorical=False,

eval_metric=None, feature_types=None, gamma=0,
grow_policy=None, importance_type=None,
interaction_constraints=None, lambda=0.5,
learning_rate=0.3, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=4, max_leaves=None,
min_child_weight=0, missing=nan,
monotone_constraints=None, multi_strategy=None,
n_estimators=None, n_jobs=None,
num_parallel_tree=None, subsample=0.5

Training:
Accuracy: 1.0 = 100 %

Confusion Matrix: [[282 0]
[0 282]]

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	282
1	1.00	1.00	1.00	282
accuracy			1.00	564
macro avg	1.00	1.00	1.00	564
weighted avg	1.00	1.00	1.00	564

Validation:
Accuracy: 0.9769595448798988 ≈ 97.7 %

Testing:
Accuracy: 0.9912280701754386 ≈ 99.1 %

Confusion Matrix: [[75 0]
[1 38]]

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	75
1	1.00	0.97	0.99	39
accuracy			0.99	114
macro avg	0.99	0.99	0.99	114
weighted avg	0.99	0.99	0.99	114

5.2.4 Support Vector Machine (SVM)

The parameters of the best fit model are:
C=2, kernel='poly', degree=1, gamma='2', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size = 200, class_weight=None, verbose=False, max_iter = -1, decision_function_shape='ovr', break_ties=False, random_state=None

Training:
Accuracy: 0.9875886524822695 ≈ 98.8 %

Confusion Matrix: [[282 0]
[7 275]]

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	282
1	1.00	0.98	0.99	282
accuracy			0.99	564
macro avg	0.99	0.99	0.99	564
weighted avg	0.99	0.99	0.99	564

Validation:
Accuracy: 0.987594816687737 ≈ 98.8%

Testing:

Accuracy: 0.9736842105263158 ≈ 97.4 %

Confusion Matrix: [[73 2]
 [1 38]]

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	75
1	0.95	0.97	0.96	39
accuracy			0.97	114
macro avg	0.97	0.97	0.97	114
weighted avg	0.97	0.97	0.97	114

PROGNOSTIC MODEL

5.3 Cleaning and Preprocessing the data

5.3.1 Dropping the 'Time' feature as it has no relevance with the purpose.

5.3.2 Handling Null Values

There are **4 null values** in the feature **'lymph_node_status'**.

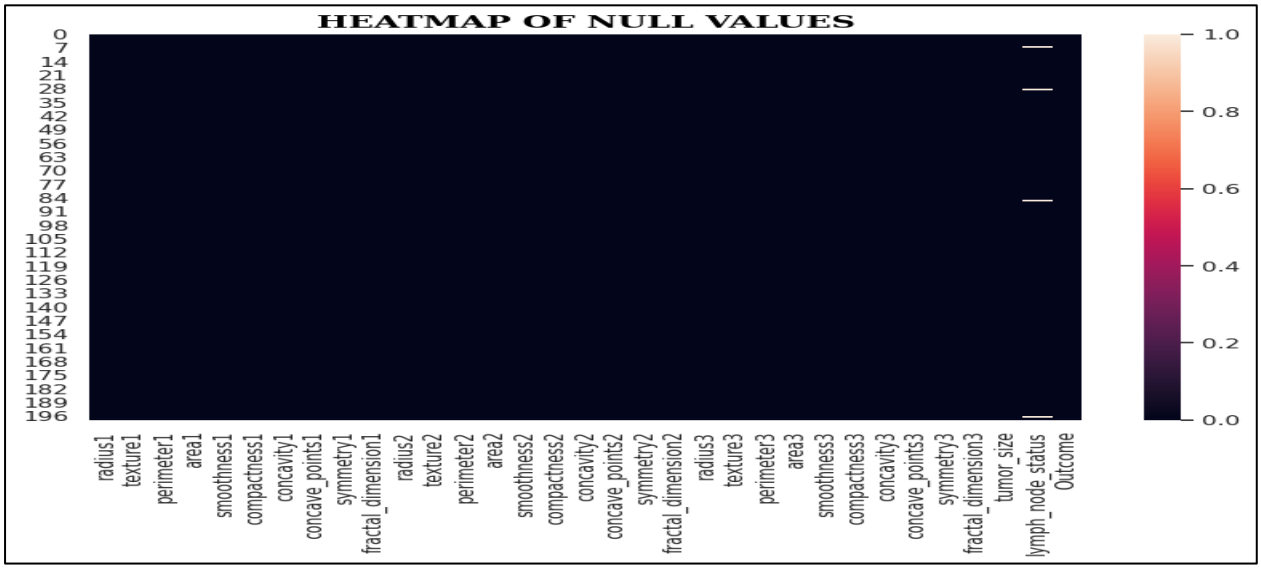

Fig 2.1

The null values were replaced with the **median** value of 'lymph_node_status'.

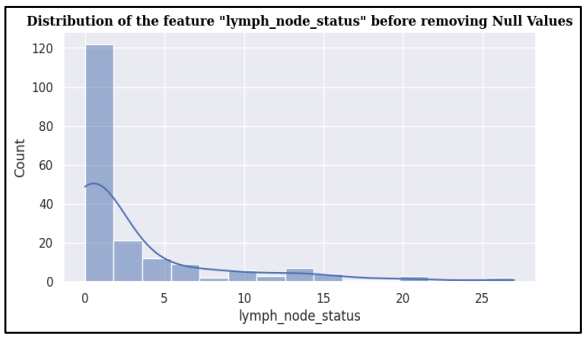

Fig 2.2

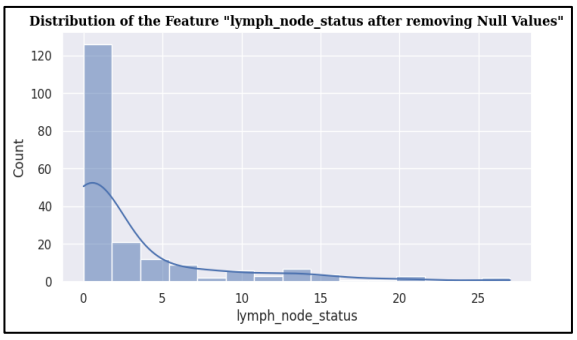

Fig 2.3

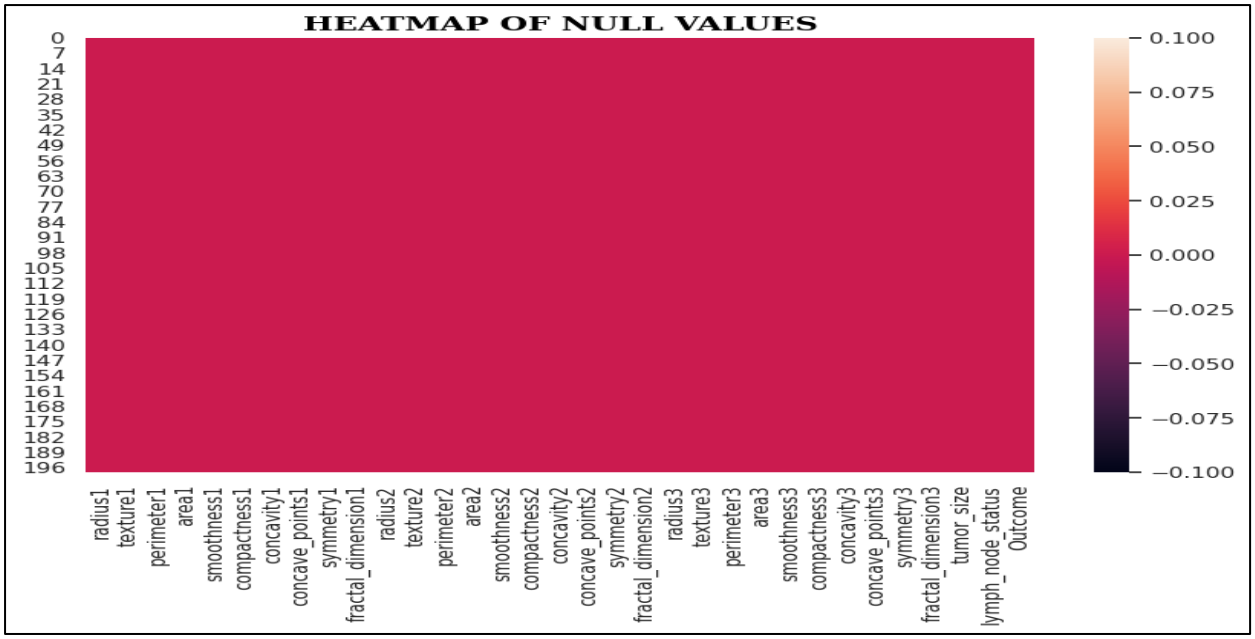

Fig 2.4

5.3.3 Visualizing Outliers

Outliers are present in all of the features. These outliers are medically valid and modifying them may cause fatal effects on accurate prognosis. Therefore, the outliers are retained.

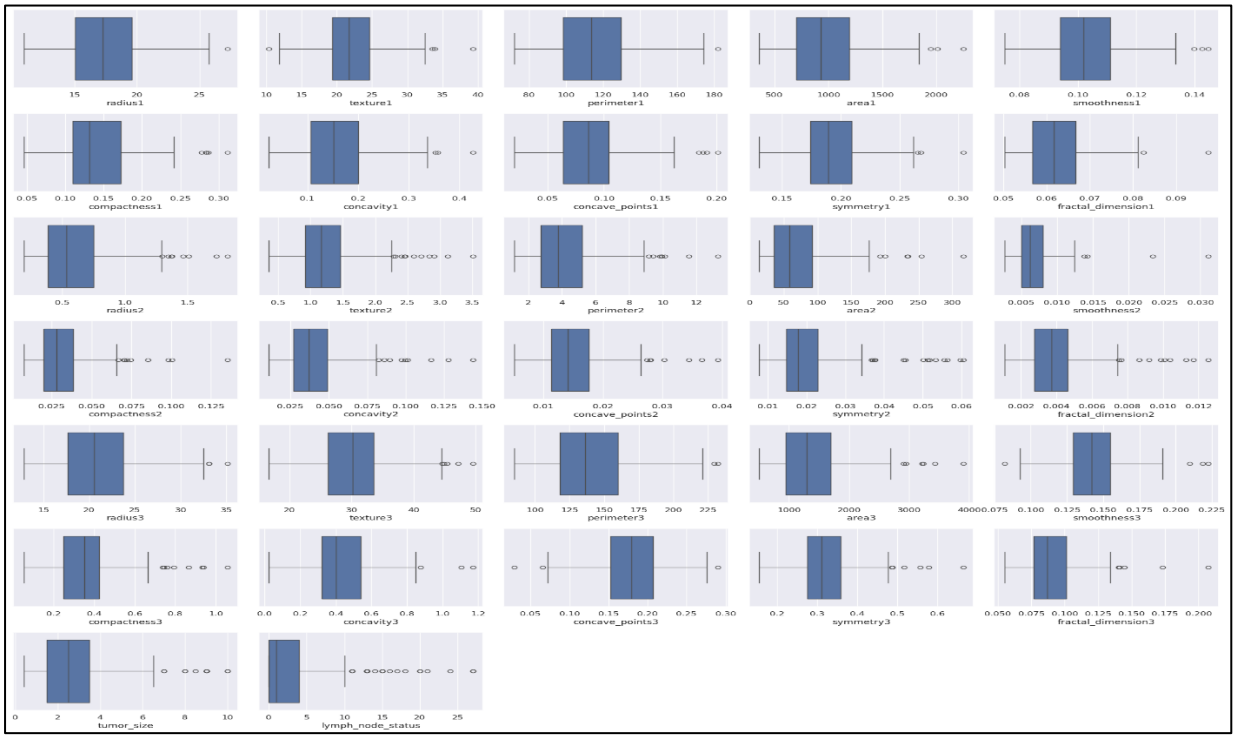

Fig 2.5

5.3.4 Visualizing the Frequency Distribution of Different Features

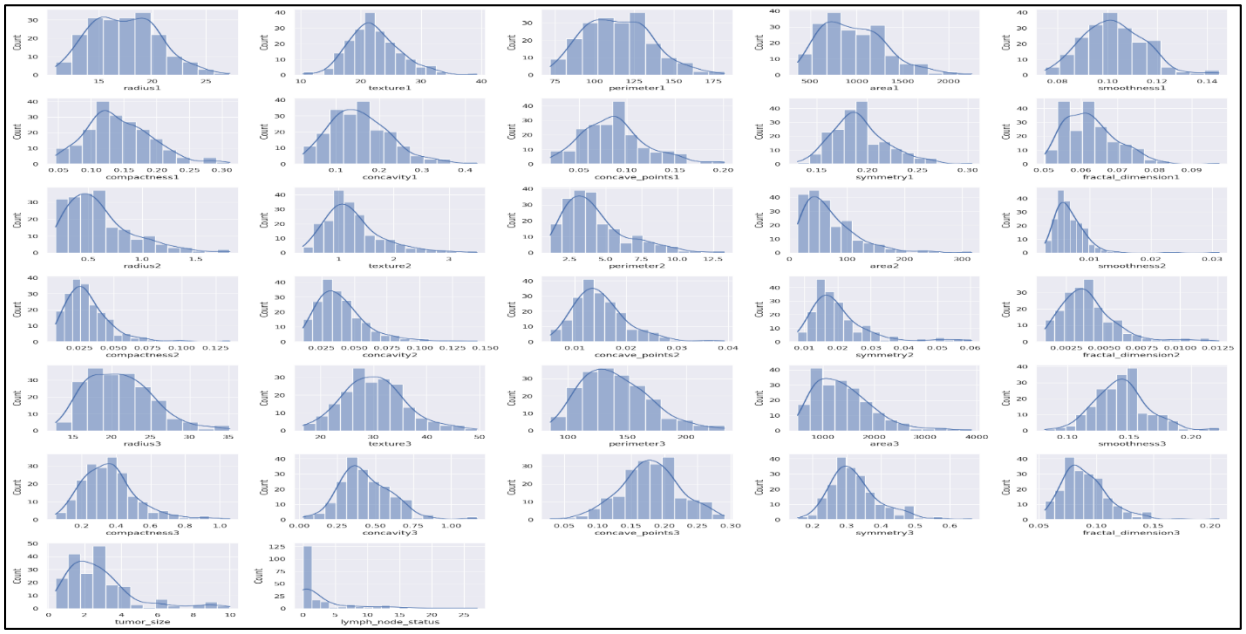

Fig 2.6

5.3.5 Visualizing Data

Pairplot for discovering prognosis patterns among all the features were used.

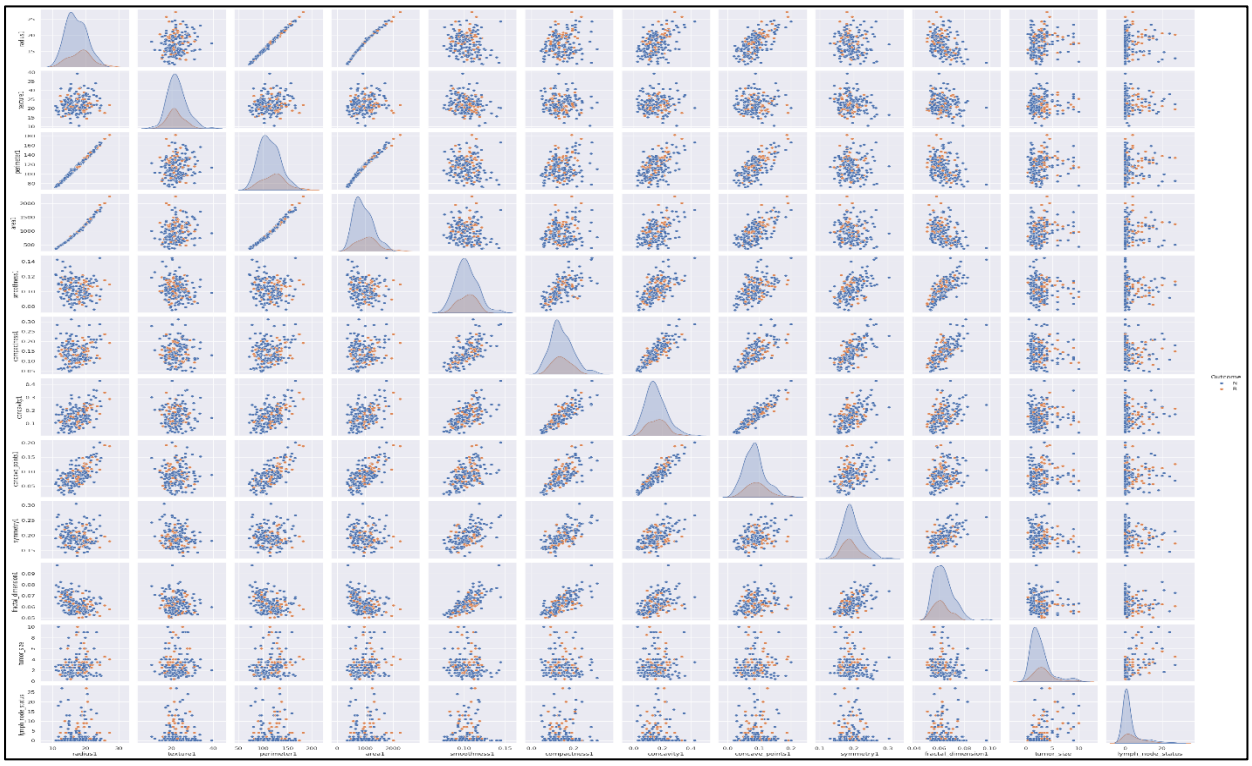

Fig 2.7

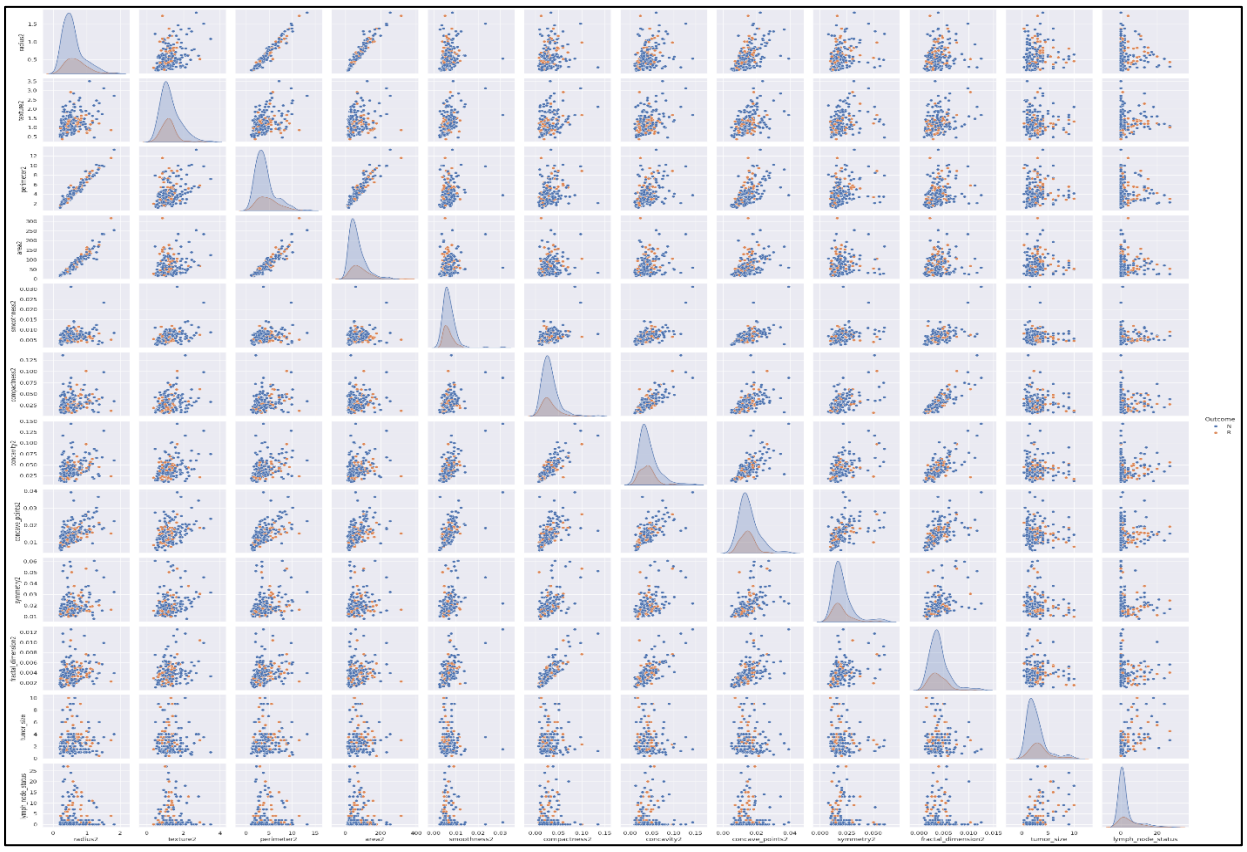

Fig 2.8

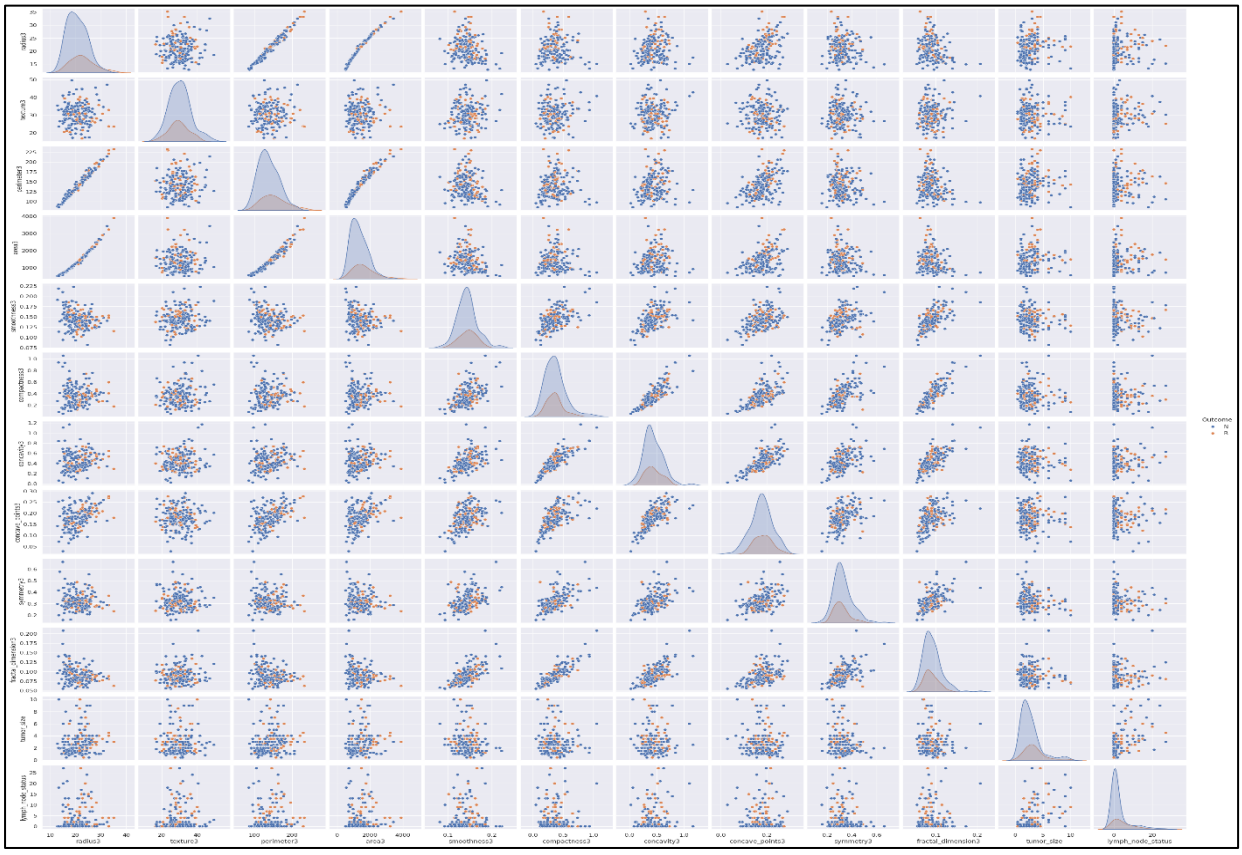

Fig 2.9

5.3.6 Label Encoding the Target Variable (Prognosis)

Label encoded the target variable as:

1 = Recurrent (R)

0 = Non-Recurrent (N)

using LabelEncoder() from sklearn.

5.3.7 Visualizing Correlation Matrix Using Heatmap

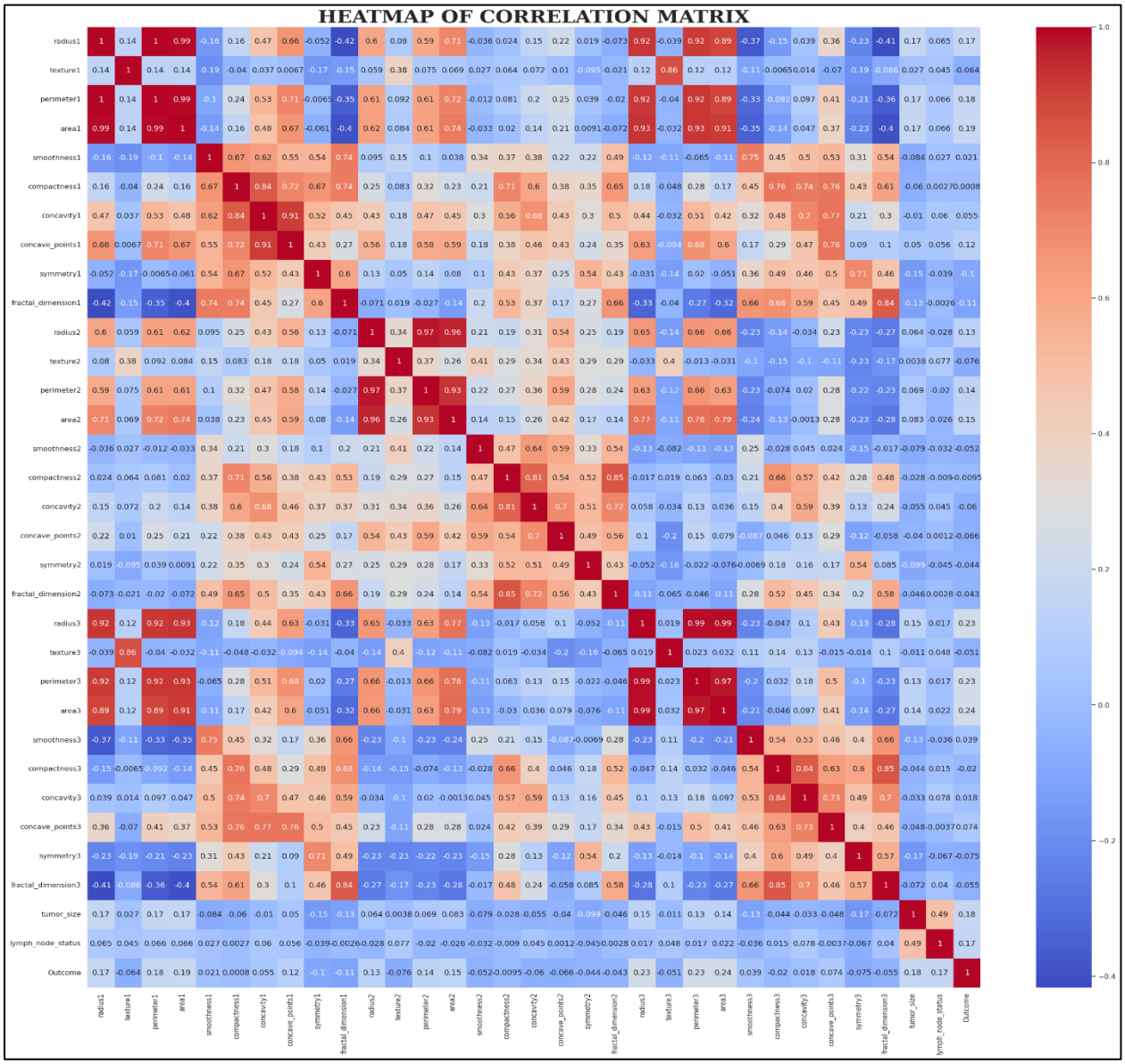

Fig 2.10

5.3.8 Dividing the Data into Training Set and Testing Set

The data was divided into **70 percent training data** and **30 percent testing data.**

5.3.9 Handling Imbalance in Training Data using Synthetic Minority Over-sampling Technique (SMOTE)

Before SMOTE

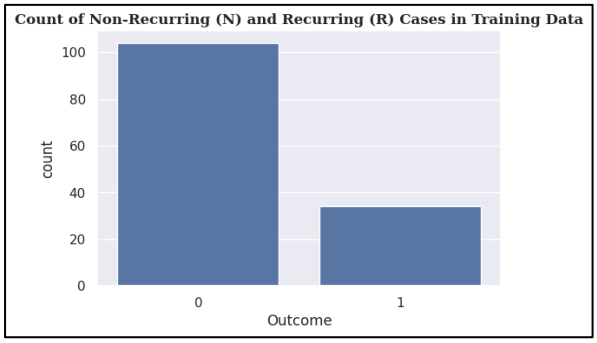

Fig 2.11

After SMOTE

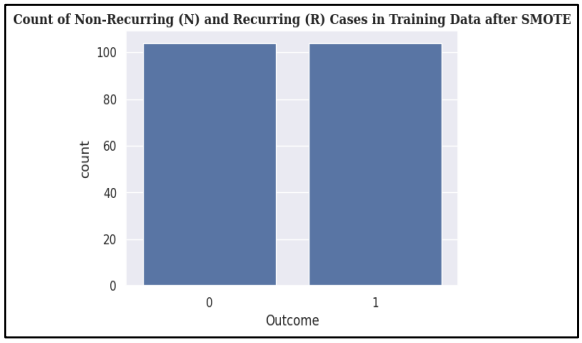

Fig 2.12

5.3.10 Normalizing the data using StandardScaler() of sklearn

5.4 Training and Evaluating the Model

The model was trained on a variety of classification algorithms and the parameters were hyper-tuned using **GridSearchCV**.

5.4.1 Logistic Regression

Parameters of the best fit model:

penalty='l1', dual=False, tol=1e-4, C=0.1, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Training:

Accuracy: 0.7067307692307693 ≈ 70.7%

Confusion Matrix: [[79 25]
[27 77]]

Classification Report:

	precision	recall	f1-score	support
0	0.66	0.86	0.74	104
1	0.79	0.56	0.66	104
accuracy			0.71	208
macro avg	0.73	0.71	0.70	208
weighted avg	0.73	0.71	0.70	208

Validation:
Accuracy: 0.6627177700348433 ≈ 66.3 %

Testing:
Accuracy: 0.6666666666666666 ≈ 66.7 %

Confusion Matrix: [[33 14]
[6 7]]

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.70	0.77	47
1	0.33	0.54	0.41	13
accuracy			0.67	60
macro avg	0.59	0.62	0.59	60
weighted avg	0.74	0.67	0.69	60

5.4.2 Random Forest

Parameters of the best fit model:

n_estimators=40, criterion='gini', max_depth=2, min_samples_split=2, min_samples_leaf=2, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None, monotonic_cst=None

Training:

Accuracy: 0.8221153846153846 ≈ 82.2%

Confusion Matrix: [[85 19]
[18 86]]

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.82	0.82	104
1	0.82	0.83	0.82	104
accuracy			0.82	208
macro avg	0.82	0.82	0.82	208
weighted avg	0.82	0.82	0.82	208

Validation:

Accuracy: 0.759349593495935 ≈ 75.9 %

Testing:

Accuracy: 0.5833333333333334 ≈ 58.3 %

Confusion Matrix: [[27 20]
[5 8]]

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.57	0.68	47
1	0.29	0.62	0.39	13
accuracy			0.58	60
macro avg	0.56	0.59	0.54	60
weighted avg	0.72	0.58	0.62	60

5.4.3 Extreme Gradient Boosting (XGBoost)

Parameters of the best fit model:
base_score=None, booster=None, callbacks=None, colsample_bylevel=None, colsample_bynode=None, colsample_bytree=None, device=None, early_stopping_rounds=None, enable_categorical=False, eval_metric=None, feature_types=None, gamma=1, grow_policy=None, importance_type=None, interaction_constraints=None, lambda=0.2, learning_rate=0.3, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=2, max_leaves=None, min_child_weight=2, missing=nan, monotone_constraints=None, multi_strategy=None, n_estimators=None, n_jobs=None, num_parallel_tree=None, subsample=0.5

Training:
Accuracy: 0.9903846153846155 ≈ 99.0 %

Confusion Matrix: [[103 1]
[1 103]]

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	104
1	0.99	0.99	0.99	104
accuracy			0.99	208
macro avg	0.99	0.99	0.99	208
weighted avg	0.99	0.99	0.99	208

Validation:

Accuracy: 0.8166085946573751 ≈ 81.7 %

Testing:

Accuracy: 0.6833333333333333 ≈ 68.3 %

Confusion Matrix: [[33 14]
[5 8]]

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.70	0.78	47
1	0.36	0.62	0.46	13
accuracy			0.68	60
macro avg	0.62	0.66	0.62	60
weighted avg	0.76	0.68	0.71	60

5.4.4 Support Vector Machine (SVM)

The parameters of the best fit model are:

C=2, kernel='poly', degree=1, gamma='2', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size = 200, class_weight=None, verbose=False, max_iter = -1, decision_function_shape='ovr', break_ties=False, random_state=None

Training:
Accuracy: 99.03846153846155 ≈ 99.0 %

Confusion Matrix: [[104 0]
[2 102]]

Classification Report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	104
1	1.00	0.98	0.99	104
accuracy			0.99	208
macro avg	0.99	0.99	0.99	208
weighted avg	0.99	0.99	0.99	208

Validation:
Accuracy: 0.8653890824622532 ≈ 86.5%

Testing:
Accuracy: 0.75 = 75.0 %

Confusion Matrix: [[43 4]
[11 2]]

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.91	0.85	47
1	0.33	0.15	0.21	13
accuracy			0.75	60
macro avg	0.56	0.53	0.53	60
weighted avg	0.70	0.75	0.71	60

6.ANALYSIS AND RESULTS

DIAGNOSTIC MODEL

Algorithm / Performance Metric	Logistic Regression	Random Forest	Extreme Gradient Boosting (XGBoost)	Support Vector Machine (SVM)
TEST CONFUSION MATRIX	[[74 1] [1 38]]	[[74 1] [1 38]]	[[75 0] [1 38]]	[[73 2] [1 38]]
TEST ACCURACY	98.2 %	98.2 %	99.1 %	97.4 %

The **Extreme Gradient Boosting (XGBoost)** algorithm gives the **optimal performance** for the **Diagnostic Model** having **high precision, high recall, high F1-score and high accuracy**.

PROGNOSTIC MODEL

Algorithm / Performance Metric	Logistic Regression	Random Forest	Extreme Gradient Boosting (XGBoost)	Support Vector Machine (SVM)
TEST CONFUSION MATRIX	[[33 14] [6 7]]	[[27 20] [5 8]]	[[33 14] [5 8]]	[[43 4] [11 2]]
TEST ACCURACY	66.7 %	58.3 %	68.3 %	75.0 %

Although the **Support Vector Machine (SVM)** algorithm gives the highest accuracy, **Extreme Gradient Boosting (XGBoost)** algorithm has **slightly better precision, strongly better recall** and a **much better F1-score** than SVM, indicating a good balance between precision and recall. Therefore, the **Extreme Gradient Boosting** algorithm gives the **optimal performance** for the **Prognosis Model**.

Performance of the Diagnostic Model is more efficient than the Prognostic Model because in the prognostic data, the target clusters are overlapping, the outcome of prognosis in the dataset does not show any considerable difference against the features. Additionally, the size of the dataset is very small (198 records). As a result, it becomes harder for machine learning algorithms to learn the underlying patterns for accurate prognosis. On the other hand, the diagnosis data, has well separated clusters for the target variable and the size of the dataset is considerably large (569 records). Therefore, the machine learning algorithms learn the underlying patterns for accurate diagnosis better.

7.CONCLUSION

Breast cancer classification using Artificial Intelligence and Machine Learning offers an accurate and efficient method for outcome prediction and early decision-making. By preprocessing the data and training both diagnostic and prognostic models using the Wisconsin Breast Cancer Datasets for Diagnosis and Prognosis respectively, the system helps in identifying tumor stage and predicting disease progression. This approach supports faster, more reliable, and personalized treatment decisions in healthcare.

The performance of the prognostic model is weaker than the diagnostic model due to unclear patterns of the target variable across features and lack of sufficient prognosis data. The prognostic model can be improved with more advanced technologies and a better dataset.

The project underscores the significance of data-driven approaches in modern medicine, paving the way for more efficient, and accessible diagnostic and prognostic solutions. Ultimately, it reflects how technology, when thoughtfully applied, can empower life-saving advancements and foster hope in the fight against cancer.

8.APPENDICES

For the completion of this project on **“Breast Cancer Classification”**, extensive **data analysis, model building, and validation** were undertaken to develop a reliable tool aimed at enhancing early detection and supporting informed clinical decisions. Some notables which were used:

- **Data:**
  i. W. Wolberg, O. Mangasarian, N. Street, and W. Street. "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1993. [Online]. Available: https://doi.org/10.24432/C5DW2B.

  ii. W. Wolberg, W. Street, and O. Mangasarian. "Breast Cancer Wisconsin (Prognostic)," UCI Machine Learning Repository, 1995. [Online]. Available: https://doi.org/10.24432/C5GK50.

- **GitHub Repository:** https://github.com/AI-fanatic24/Breast-Cancer-Classification/tree/main

- **Libraries Used:** Numpy, Pandas, Scikit-Learn, Matplotlib and Seaborn

- **Cloud Platform:** Google Colab Notebook

- **Code Editor:** Visual Studio Code

- **Programming Language:** Python