

## A Project of Data Science Lab



Jagannath University  
Department of Computer Science & Engineering  
CSEL 4236: Data Science Lab

## Project Title

# Breast Cancer InsightX: AI-Powered Diagnostic Analysis

By

Farhana Akter Suci | Rifah Sajida Deya

ID: B190305001

ID: B190305004

Submitted to

**Dr. Md. Manowarul Islam**

Associate Professor, CSE, Jagannath University

---

24th April, 2025

---

## Abstract

The early and accurate diagnosis of breast cancer is a crucial aspect of reducing mortality and improving treatment outcomes among women. The Breast Cancer Diagnostic dataset, based on features extracted from fine needle aspirate (FNA) of breast masses, serves as an effective source for identifying benign and malignant tumors. However, limitations in medical infrastructure and high testing costs still hinder widespread access to timely diagnosis. In this study, we propose a machine learning-based diagnostic framework that leverages audio-free, non-invasive data for the effective classification of breast cancer. We evaluate several ensemble classification algorithms, including AdaBoost, LightGBM, XGBoost, CatBoost, and Random Forest, by tuning their hyperparameters through both grid and randomized search with cross-validation. Our approach integrates various feature selection and dimensionality reduction techniques, such as Principal Component Analysis (PCA), Forward Selection (FS), Maximum Relevance Minimum Redundancy (mRMR), and Recursive Feature Elimination (RFE), to extract the most informative features. Experimental results indicate that XGBoost and CatBoost classifiers, when combined with Forward Selection, achieve the highest classification accuracy. To ensure interpretability of model predictions, we incorporated a SHAP-based explainable model with the best-performing classifiers. This interpretable approach allows for clear understanding of feature contributions, enabling clinicians to make informed, timely decisions and improving overall diagnostic reliability.

---

## Table of Contents

<b>Abstract</b> .....	1
<b>1. Introduction</b> .....	3
<b>2. Motivation</b> .....	4
<b>3. Problem Statement</b> .....	5
<b>4. Materials and Methods</b> .....	6
4.1. <b>Proposed Methodology (Flow Diagram)</b> .....	7
4.2. <b>Dataset Selection</b> .....	8
4.3. <b>Proposed Methodology (Algorithm)</b> .....	10
4.4. <b>EDA</b> .....	11
<b>5. Performance Evaluation</b> .....	18
5.1. <b>Results without feature selection</b> .....	18
5.2. <b>Results of mRMR feature selection</b> .....	19
5.3. <b>Results of RFE feature selection</b> .....	20
5.4. <b>Results of PCA feature selection</b> .....	22
5.5. <b>Results of FS feature selection</b> .....	23
<b>6. Performance Comparison</b> .....	25
6.1. <b>Accuracy Comparison</b> .....	25
6.2. <b>Confusion matrix</b> .....	27
6.3. <b>AUC-ROC</b> .....	30
6.4. <b>Table of hyperparameters</b> .....	30
<b>7. Explainability and Deployment</b> .....	34
7.1. <b>Feature Importance</b> .....	34
7.2. <b>SHAP</b> .....	35
7.3. <b>Web App View</b> .....	36
<b>8. Conclusion</b> .....	37
<b>9. References</b> .....	38

# **Chapter 1**

---

## **Introduction**

Breast cancer is one of the most prevalent and life-threatening diseases affecting women worldwide. Early and accurate diagnosis is critical for improving patient outcomes and reducing mortality rates. In this study, we developed a robust and interpretable machine learning pipeline for breast cancer classification, leveraging a combination of advanced preprocessing techniques, feature selection, ensemble modeling, and explainable AI.

Unlike traditional approaches that rely on baseline models, we employed a suite of powerful ensemble learning algorithms including Random Forest (RF), AdaBoost, XGBoost, Gradient Boosting, Bagging, and CatBoost. These models are known for their ability to enhance predictive performance by reducing variance and bias. Among them, CatBoost and XGBoost demonstrated the highest classification accuracy of 98.25%, particularly when integrated with Feature Selection (FS) methods such as MRMR, PCA, RFE, and Forward Selection.

To ensure model transparency and trustworthiness, we adopted SHapley Additive exPlanations (SHAP) for Explainable AI (XAI). SHAP values provided deep insights into feature contributions, enabling clinicians and researchers to better understand the decision-making process of the models.

Our proposed pipeline encompasses data preprocessing, feature engineering, k-fold cross-validation, hyperparameter tuning using grid and random search, and a comprehensive evaluation using metrics like Accuracy, Precision, Recall, F1-Score, and MCC. This holistic approach ensures both high performance and interpretability, paving the way for reliable and understandable AI-assisted breast cancer diagnosis.

## Chapter 2

---

### Motivation

Breast cancer remains a major global health concern, with millions of new cases diagnosed annually. Despite the advancements in medical imaging and diagnostic technologies, early detection continues to be a challenge due to the complex and heterogeneous nature of the disease. Traditional diagnostic approaches often rely on manual interpretation, which can be time-consuming, prone to human error, and dependent on the expertise of clinicians.

The rise of machine learning (ML) and artificial intelligence (AI) offers an opportunity to enhance diagnostic accuracy and support clinical decision-making. However, many ML models act as "black boxes," providing limited interpretability—a critical requirement in healthcare applications where understanding *why* a model makes a certain prediction is as important as the prediction itself.

This study is motivated by the need to develop a highly accurate, reliable, and explainable machine learning framework that can assist in the early detection of breast cancer. By using powerful ensemble learning techniques such as CatBoost, XGBoost, and others, we aim to leverage their ability to capture complex nonlinear patterns and interactions in the data. Additionally, integrating SHAP-based Explainable AI ensures that model decisions are transparent and justifiable, building trust among healthcare professionals.

The motivation also stems from the desire to explore and compare various feature selection techniques to enhance model performance and reduce computational complexity. Ultimately, this work aspires to bridge the gap between high-performance prediction and interpretability, contributing to the development of practical AI solutions in the medical domain.

## Chapter 3

---

### Problem Statement

1. **Diagnostic Challenge:** Breast cancer diagnosis is a complex task where early detection is vital for improving patient survival rates. Traditional diagnostic techniques are often time-consuming, prone to human error, and rely heavily on expert interpretation, which can lead to inconsistent results. Moreover, the increasing volume of medical data adds further pressure on healthcare professionals, highlighting the need for automated support systems.
2. **Need for High Performance and Interpretability:** While machine learning has shown promise in automating diagnosis, many models function as black boxes, lacking the transparency needed in critical domains like healthcare. There is a pressing need for a system that combines high predictive accuracy with explainability to build trust and support clinical decision-making. An interpretable model not only improves clinician confidence but also helps in identifying potential biomarkers and understanding disease mechanisms.
3. **Project Focus:** This study aims to develop a robust and interpretable breast cancer classification model using ensemble methods (RF, AdaBoost, XGBoost, GradientBoost, Bagging, CatBoost) alongside feature selection techniques. SHAP is utilized to explain the model's decisions, and CatBoost and XGBoost achieved 98.25% accuracy on FS-selected features—demonstrating both performance and interpretability.

# **Chapter 4**

---

## **Materials and Methods**

### **4.1 Proposed Methodology**

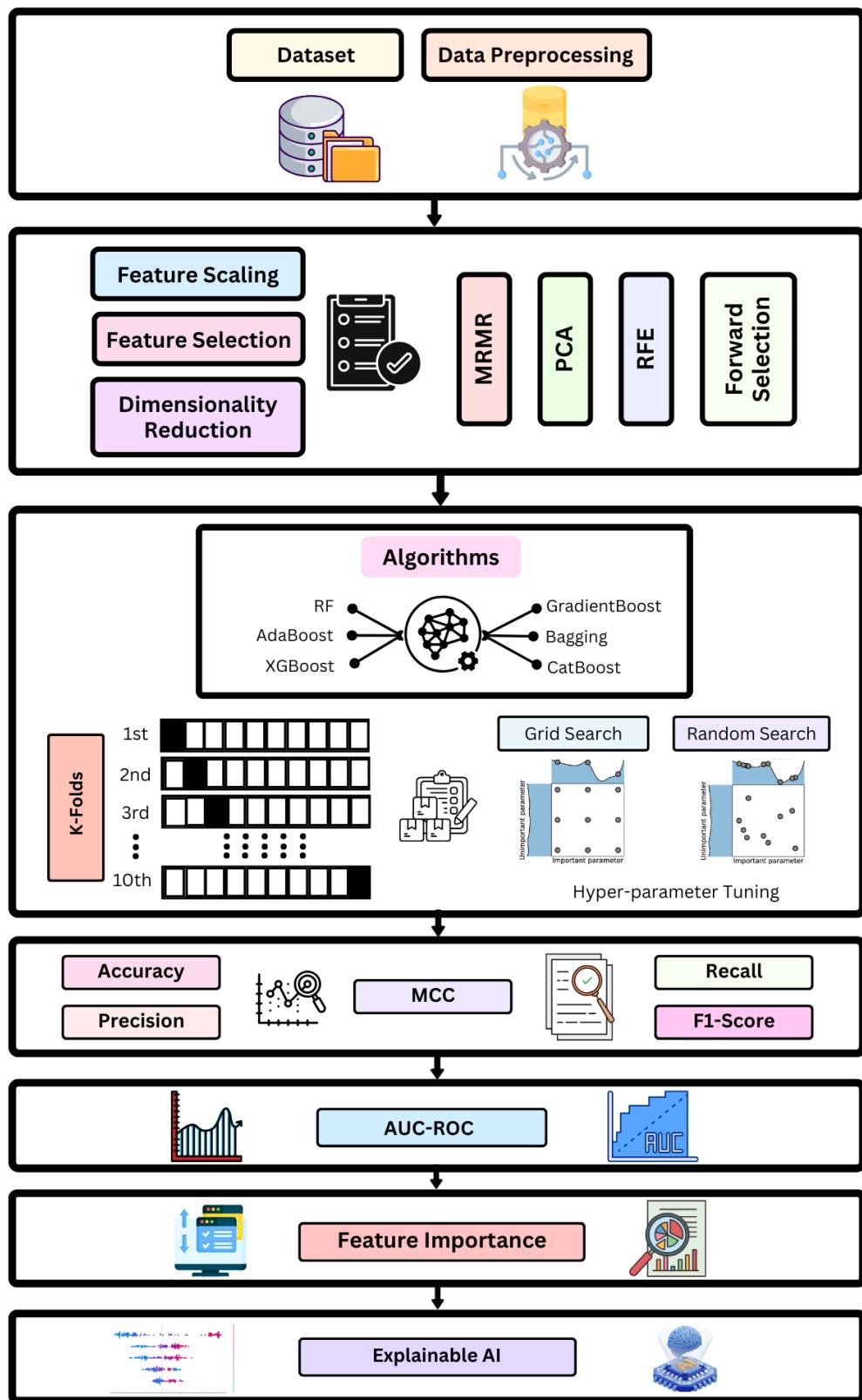
The process begins with collecting the dataset and preparing it for analysis through cleaning, handling missing values, and formatting. This step is followed by feature scaling to standardize the values, ensuring consistency across variables. Next, feature selection and dimensionality reduction methods like MRMR (Minimum Redundancy Maximum Relevance), PCA (Principal Component Analysis), RFE (Recursive Feature Elimination), and forward selection are applied to keep only the most relevant and informative features, reducing the complexity of the data.

After preprocessing, the data is fed into various machine learning algorithms, including Random Forest, XGBoost, AdaBoost, CatBoost, GradientBoost, and Bagging classifiers. These models are trained and tested using k-fold cross-validation, which ensures the model generalizes well to unseen data. To enhance performance, hyperparameter tuning is conducted using grid search and random search to identify the best model configurations.

Model performance is then evaluated using multiple metrics tailored for binary classification tasks. These include accuracy, precision, recall, F1-score, Matthews correlation coefficient (MCC), and AUC-ROC. These metrics provide a comprehensive view of the model's ability to correctly identify both classes and minimize errors.

Finally, the model's decision-making is examined using feature importance analysis and explainable AI (XAI) tools such as SHAP. These techniques help interpret which features influenced predictions most and make the model's behavior more transparent and trustworthy, which is crucial for clinical or sensitive decision-making applications.

The flow diagram is given in Figure 1.1 to visually represent the proposed methodology.



**Figure 4.1: Proposed Methodology**

## 4.2 Dataset Selection

The Wisconsin Breast Cancer Diagnostic (WBCD) dataset is collected from the **UCI Machine Learning Repository**. It's a widely used benchmark dataset in medical diagnostics research. It was originally compiled by Dr. William H. Wolberg at the University of Wisconsin Hospitals. The table 4.1 contains the dataset's details.

<b>Dataset Characteristic</b>	Multivariate
<b>No. of Instances</b>	568
<b>Attributes Characteristic</b>	Real
<b>No. of Attributes</b>	32
<b>Missing Values</b>	N/A
<b>Associated Tasks</b>	Classification
<b>Types of Classification</b>	Binary {0 for healthy and 1 for breast cancer patient}
<b>Creator</b>	Dr. William H. Wolberg, a physician at the University of Wisconsin-Madison

Tabel 4.1: Detail of Breast Cancer's Dataset

Table 4.2 provides details on the 568 records' attribute:

Attribute	Description
<b>ID</b>	Unique identification number for each patient (not used in model training).
<b>Diagnosis</b>	Tumor classification: M (malignant) or B (benign).
<b>Radius (mean, SE, worst)</b>	Average, standard error, and worst-case distance from center to perimeter of nuclei.
<b>Texture (mean, SE, worst)</b>	Standard deviation of gray-scale intensity values in the image.
<b>Perimeter (mean, SE, worst)</b>	Total length of the boundary of the nuclei.
<b>Area (mean, SE, worst)</b>	Total area covered by the nuclei.
<b>Smoothness (mean, SE, worst)</b>	Measure of variation in radius lengths (local irregularities in contour).
<b>Compactness (mean, SE, worst)</b>	Ratio of perimeter squared to area minus one
<b>Concavity (mean, SE, worst)</b>	Severity of concave portions of the contour (degree of indentations).
<b>Concave points (mean, SE, worst)</b>	Number of concave regions or points on the nucleus boundary.
<b>Symmetry (mean, SE, worst)</b>	Degree of symmetry of the nuclei.
<b>Fractal dimension (mean, SE, worst)</b>	Measure of complexity in the nucleus shape, calculated using a fractal method.

Table 4.2: Showing attribute information of dataset

### **4.3 Proposed Methodology**

---

**Algorithm 1** The performance of ensemble classification models using feature selection and dimension reduction techniques

---

```
1: procedure ModelPerformance(X, Y, List of Models, List of Feature Selection Methods, PCA  
Components Range, List of Hyperparameter Tuning Methods)  
2:   Data Preprocessing  
3:   Apply Min-Max scaling to normalize the features in X.  
4:   Model Evaluation (Initial):  
5:   for all model in List of Models do  
6:     Train model on the preprocessed dataset.  
7:     Evaluate model performance using metrics  
8:   end for  
9:   Feature Selection:  
10:  for all model in List of Feature Selection Methods do  
11:    Apply method to select relevant features.  
12:    for all model in List of Models do  
13:      Train model on the selected features  
14:      Evaluate model performance using metrics  
15:    end for  
16:  end for  
17:  Dimension Reduction (PCA):  
18:  for all n_components in PCA Components Range do  
19:    Apply PCA for dimension reduction with n_comonents  
20:    for all model in List of Models do  
21:      Train model on the reduced dataset  
22:      Evaluate model performance using metrics  
23:    end for  
24:  end for  
25:  Hyperparameter Tuning:  
26:  for all model in List of Models do  
27:    Apply GridSearchCV to find the best hyperparameters  
28:    Train model with the best hyperparameters.  
29:    Evaluate model performance using metrics.  
30:  end for  
31:  for all combination of preprocessing, feature selection,dimension reduction, tuning do  
32:    Record performance metrics  
33:  end for  
34:  Final Results: Compare and analyze the results, identify best performing models and  
configurations (Feature importance, SHAP)  
35: end procedure
```

## 4.4 Exploratory Data Analysis(EDA)

	count	mean	std	min	25%	50%	75%	max
Radius_mean	569.0	14.13	3.52	6.98	11.70	13.37	15.78	28.11
Texture_mean	569.0	19.29	4.30	9.71	16.17	18.84	21.80	39.28
Perimeter_mean	569.0	91.97	24.30	43.80	75.17	86.24	104.10	188.50
Area_mean	569.0	654.89	351.91	143.50	420.30	551.10	782.70	2501.00
Smoothness_mean	569.0	0.10	0.01	0.05	0.09	0.10	0.11	0.16
Compactness_mean	569.0	0.10	0.05	0.02	0.06	0.09	0.13	0.35
Concavity_mean	569.0	0.09	0.08	0.00	0.03	0.06	0.13	0.43
Concave_points_mean	569.0	0.05	0.04	0.00	0.02	0.03	0.07	0.20
Symmetry_mean	569.0	0.18	0.03	0.11	0.16	0.18	0.20	0.30
Fractal_dimension_mean	569.0	0.06	0.01	0.05	0.06	0.06	0.07	0.10
Radius_se	569.0	0.41	0.28	0.11	0.23	0.32	0.48	2.87
Texture_se	569.0	1.22	0.55	0.36	0.83	1.11	1.47	4.88
Perimeter_se	569.0	2.87	2.02	0.76	1.61	2.29	3.36	21.98
Area_se	569.0	40.34	45.49	6.80	17.85	24.53	45.19	542.20
Smoothness_se	569.0	0.01	0.00	0.00	0.01	0.01	0.01	0.03
Compactness_se	569.0	0.03	0.02	0.00	0.01	0.02	0.03	0.14
Concavity_se	569.0	0.03	0.03	0.00	0.02	0.03	0.04	0.40

Concave_points_se	569.0	0.01	0.01	0.00	0.01	0.01	0.01	0.05
Symmetry_se	569.0	0.02	0.01	0.01	0.02	0.02	0.02	0.08
Fractal_dimension_se	569.0	0.00	0.00	0.00	0.00	0.00	0.00	0.03
Radius_worst	569.0	16.27	4.83	7.93	13.01	14.97	18.79	36.04
Texture_worst	569.0	25.68	6.15	12.02	21.08	25.41	29.72	49.54
Perimeter_worst	569.0	107.26	33.60	50.41	84.11	97.66	125.40	251.20
Area_worst	569.0	880.58	569.36	185.20	515.30	686.50	1084.00	4254.00
Smoothness_worst	569.0	0.13	0.02	0.07	0.12	0.13	0.15	0.22
Compactness_worst	569.0	0.25	0.16	0.03	0.15	0.21	0.34	1.06
Concavity_worst	569.0	0.27	0.21	0.00	0.11	0.23	0.38	1.25
Concave_points_worst	569.0	0.11	0.07	0.00	0.06	0.10	0.16	0.29
Symmetry_worst	569.0	0.29	0.06	0.16	0.25	0.28	0.32	0.66
Fractal_dimension_worst	569.0	0.08	0.02	0.06	0.07	0.08	0.09	0.21

Table 4.3: Showing statistical measure of dataset

The main purpose of the dataset is to distinguish between persons who have been diagnosed with Breast Cancer's disease and those who are healthy, based on the values of the "Diagnosis" column, which have been allocated as 0 for healthy individuals and 1 for those with BC. According to figure 4.2, which uses a pie plot to illustrate the dataset, there are 62.7% of cases that are healthy and 37.3% of cases that have Breast Cancer disease.

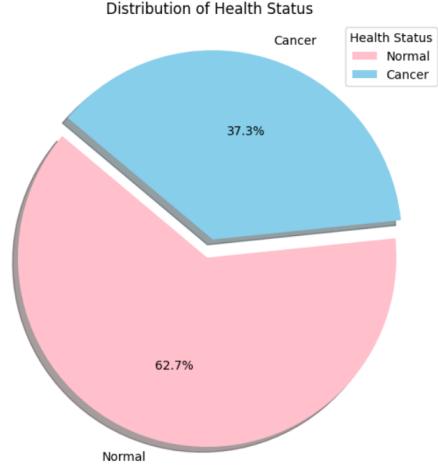


Figure 4.2: Distribution of Health Status

In the heatmap of this dataset, several notable patterns emerge. There are strong positive correlations among certain features—`radius_mean` is highly correlated with `perimeter_mean` and `area_mean`, while `concavity_mean` shows a strong correlation with `concave_points_mean`. Similarly, `area_worst`, `radius_worst`, and `perimeter_worst` are also highly intercorrelated. On the other hand, some features like `fractal_dimension_mean` and `smoothness_error` exhibit weak or no correlation with most others. Additionally, the heatmap reveals distinct groups of features: all "mean" features tend to cluster together, "worst" features form another group, and standard error (SE) features generally display weaker correlations with both the mean and worst feature sets.

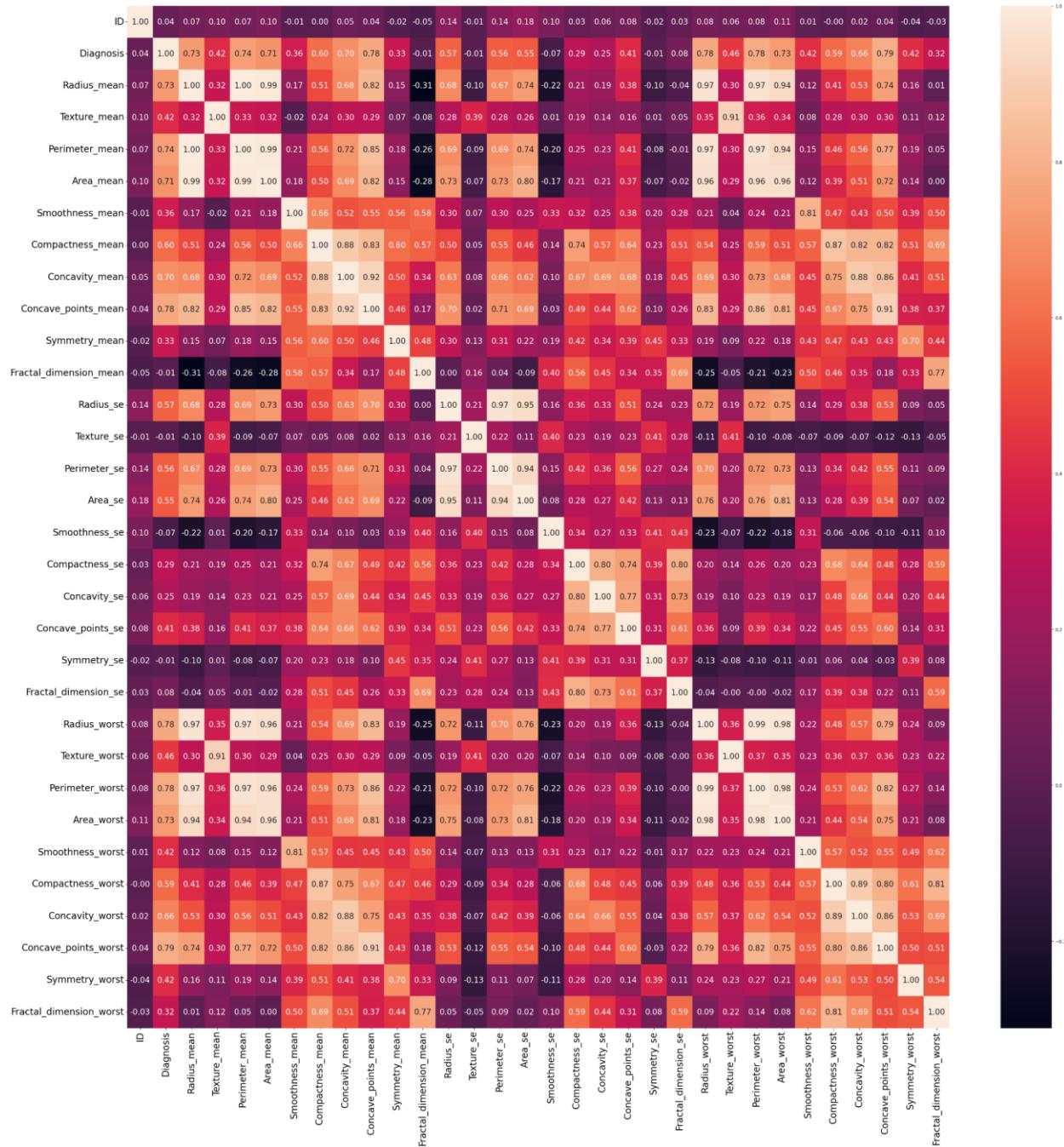


Figure 4.3: Correlation Heatmap of Breast Cancer Diagnostic Features

The distribution of features in the dataset reveals various patterns. Several attributes such as radius\_mean, area\_mean, concavity\_mean, and compactness\_mean are right-skewed, displaying a sharp peak on the left with a long tail on the right, indicating that most patients have low values while a few have significantly higher measurements—typically associated with tumor size or irregularity, which differ between benign and malignant cases. In contrast, features like texture\_mean, texture\_worst, and smoothness\_mean exhibit nearly symmetric distributions, suggesting a more even spread of values around the center. Additionally, some attributes such as fractal\_dimension\_se, smoothness\_se, concave\_points\_se, and symmetry\_se are also right-skewed but with low variance, forming tight unimodal peaks that reflect consistent measurements across the majority of patients.

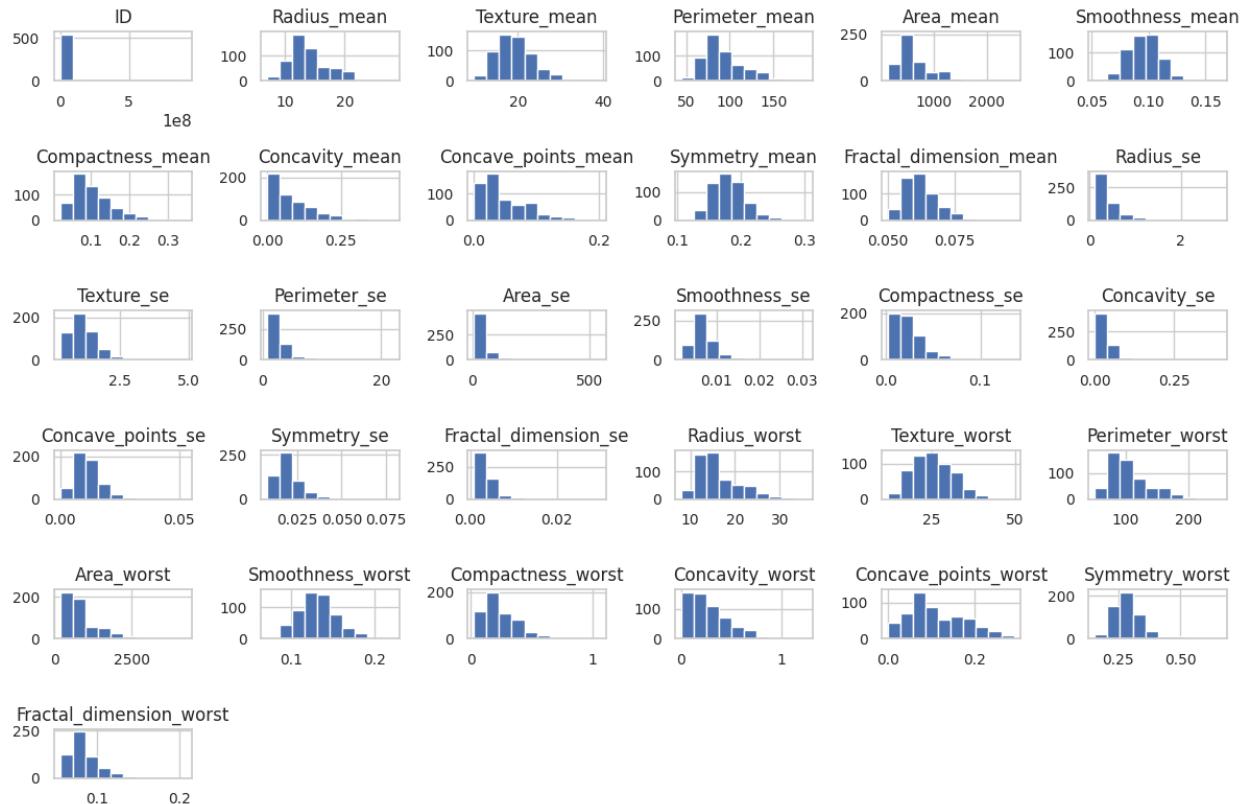


Figure 4.4: Histogram of Features Distribution

The boxplot offers a clear view of the distribution and spread of each feature while highlighting the presence of outliers. Most features exhibit a right-skewed distribution, with a longer upper whisker and numerous outlier points above the upper quartile, indicating that a subset of patients have unusually high values in features such as radius\_mean, area\_mean, concavity\_mean, and perimeter\_worst—typically associated with malignant tumors. In contrast, some features like fractal\_dimension\_se and smoothness\_se show tight box structures and minimal whisker lengths, suggesting low variance and consistency in those measurements across patients. The boxplots also help distinguish between features with broad variability and those with narrower, tightly packed values, offering insight into which attributes may be more discriminative in classifying tumor types. Overall, boxplots serve as an effective tool for visualizing outliers, understanding feature spread, and identifying the characteristics that could be important for diagnostic modeling.

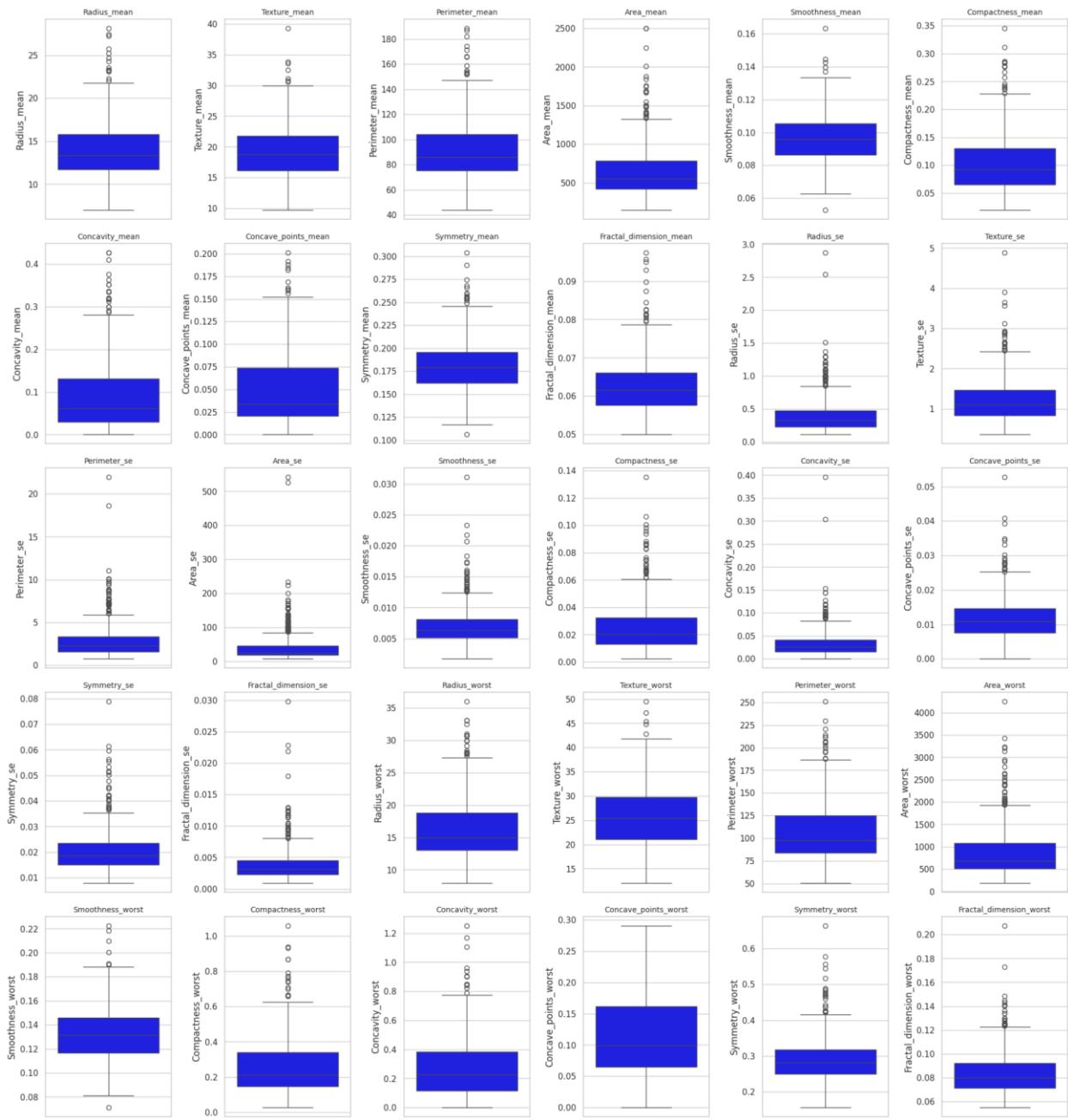


Figure 4.5: Exploratory Analysis of Outliers Using Boxplots

# Chapter 5

---

## Performance Evaluation

The Wisconsin Breast Cancer Diagnostic (WBCD) dataset of UCI Machine Learning Repository , all 32 features , is utilizes in this experiment. It predicts and partitions data into 80:20 for testing and training. we chose and retrieved features from the data after preprocessing it. First, we preprocessed the data and selected and extracted features. Next, we used GridSearchCV and RandomizedSearchCV to help us tune our hyperparameters, giving us the best values that have been used in all our models to the best accuracies on the Breast Cancer Diagnostic Dataset.

### 5.1 Results Without Feature Selection

Table 5.1: Results Without Feature Selection

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	96.49	97.56	93.02	95.24	92.53
AdaBoost	96.49	97.56	93.02	95.24	92.53
XGBoost	95.61	95.24	93.02	94.12	90.64
GradientBoost	95.61	95.24	93.02	94.12	90.64
Bagging	95.61	95.24	93.02	94.12	90.64
CatBoost	97.37	97.62	95.35	96.47	94.39

As depicted in Table [5.1], CatBoost achieved utmost accuracy of 97.37% in the scenario of results without feature selection with 31 features. The precision, recall, f1-score, MCC stand at 97.62%, 95.35%, 96.47% and 94.39% respectively.

## 5.2 Results of mRMR feature selection

Table 5.2: Results of mRMR feature selection without hyperparameter tuning

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	94.74	95.12	90.70	92.86	88.76
AdaBoost	94.74	95.12	90.70	92.86	88.76
XGBoost	93.86	95.00	88.37	91.57	86.89
GradientBoost	95.61	95.24	93.02	94.12	90.64
Bagging	94.74	95.12	90.70	92.86	88.76
CatBoost	97.37	95.45	97.67	96.55	94.44

As depicted in Table [5.2], CatBoost achieved utmost accuracy of 97.37% in mRMR feature selection scenarios without hyperparameter tuning. The precision, recall, f1-score, MCC stand at 95.45%, 97.67%, 96.55% and 94.44% respectively.

Table 5.3: Results of mRMR feature selection with grid search

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	95.61	95.24	93.02	94.12	90.64
AdaBoost	94.74	95.12	90.70	92.86	88.76
XGBoost	94.74	95.12	90.70	92.86	88.76
GradientBoost	94.74	95.12	90.70	92.86	88.76
Bagging	95.61	95.24	93.02	94.12	90.64
CatBoost	95.61	95.24	93.02	94.12	90.64

As depicted in Table [5.3], RF, Bagging and CatBoost achieved utmost accuracy of 95.61% in mRMR feature selection scenarios with grid search.

Table 5.4: Results of mRMR feature selection with randomized search

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	95.61	95.24	93.02	94.12	90.64
AdaBoost	94.74	95.12	90.70	92.86	88.76
XGBoost	93.86	95.00	88.37	91.57	86.89
GradientBoost	95.61	95.24	93.02	94.12	90.64
Bagging	95.61	95.24	93.02	94.12	90.64
CatBoost	97.37	95.45	97.67	96.55	94.44

As depicted in Table [5.4], CatBoost achieved utmost accuracy of 97.37% in mRMR feature selection scenarios without hyperparameter tuning. The precision, recall, f1-score, MCC stand at 95.45%, 97.67%, 96.55% and 94.44% respectively.

### 5.3 Results of RFE feature selection

Table 5.5: Results of RFE feature selection without hyperparameter tuning

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	96.49	95.35	95.35	95.35	92.53
AdaBoost	95.61	95.24	93.02	94.12	90.64
XGBoost	94.74	93.02	93.02	93.02	88.80
GradientBoost	94.74	95.12	90.70	92.86	88.76
Bagging	93.86	95.00	88.37	91.57	86.89
CatBoost	96.49	97.56	93.02	95.24	92.53

As depicted in Table [5.5], RF and CatBoost achieved utmost accuracy of 96.49 and 97.37% respectively, in RFE feature selection scenarios without hyperparameter tuning.

Table 5.6: Results of RFE feature selection with grid search

	In Percentage (%)				
Algorithms	Accuracy	Precision	Recall	F1-score	MCC
RF	96.49	95.35	95.35	95.35	92.53
AdaBoost	95.61	95.24	93.02	94.12	90.64
XGBoost	93.86	92.86	90.70	91.76	86.89
GradientBoost	94.74	95.12	90.70	92.86	88.76
Bagging	96.49	97.56	93.02	95.24	92.53
CatBoost	95.61	97.50	90.70	93.98	90.68

As depicted in Table [5.6], RF achieved utmost accuracy of 96.49% in RFE feature selection scenarios with grid search. The precision, recall, f1-score, MCC stand at 95.35%, 95.35%, 95.35% and 92.53% respectively.

Table 5.7: Results of RFE feature selection with randomized search

	In Percentage (%)				
Algorithms	Accuracy	Precision	Recall	F1-score	MCC
RF	96.49	97.56	93.02	95.24	92.53
AdaBoost	95.61	95.24	93.02	94.12	90.64
XGBoost	93.86	92.86	90.70	91.76	86.89
GradientBoost	94.74	95.12	90.70	92.86	88.76
Bagging	95.61	97.50	90.70	93.98	90.68
CatBoost	95.61	97.50	90.70	93.98	90.68

As depicted in Table [5.7], RF achieved utmost accuracy of 96.49% in RFE feature selection scenarios with random search. The precision, recall, f1-score, MCC stand at 97.56%, 93.02%, 95.24% and 92.53% respectively.

## 5.4 Results of PCA feature selection

Table 5.8: Results of PCA feature selection without hyperparameter tuning

	In Percentage (%)				
Algorithms	Accuracy	Precision	Recall	F1-score	MCC
RF	95.61	93.18	95.35	94.25	90.72
AdaBoost	97.37	95.45	97.67	96.55	94.44
XGBoost	98.25	97.67	97.67	97.67	96.27
GradientBoost	96.49	95.35	95.35	95.35	92.53
Bagging	95.61	93.18	95.35	94.25	90.72
CatBoost	96.49	95.35	95.35	95.35	92.53

As depicted in Table [5.8], XGBoost achieved utmost accuracy of 98.25% in PCA feature selection scenarios without hyperparameter tuning. The precision, recall, f1-score, MCC stand at 97.67%, 97.67%, 97.67% and 96.27% respectively.

Table 5.9: Results of PCA feature selection with grid search

	In Percentage (%)				
Algorithms	Accuracy	Precision	Recall	F1-score	MCC
RF	95.61	93.18	95.35	94.25	90.72
AdaBoost	97.37	95.45	97.67	96.55	94.44
XGBoost	96.49	95.35	95.35	95.35	92.53
GradientBoost	96.49	95.35	95.35	95.35	92.53
Bagging	96.49	93.33	97.67	95.45	92.66
CatBoost	96.49	95.35	95.35	95.35	92.53

As depicted in Table [5.9], AdaBoost achieved utmost accuracy of 97.37% in PCA feature selection scenarios with grid search. The precision, recall, f1-score, MCC stand at 95.45%, 97.67%, 96.55% and 94.44% respectively.

Table 5.10: Results of PCA feature selection with randomized search

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	95.61	93.18	95.35	94.25	90.72
AdaBoost	97.37	95.45	97.67	96.55	94.44
XGBoost	95.61	95.24	93.02	94.12	90.64
GradientBoost	96.49	95.35	95.35	95.35	92.53
Bagging	94.74	93.02	93.02	93.02	88.80
CatBoost	96.49	95.35	95.35	95.35	92.53

As depicted in Table [5.10], AdaBoost achieved utmost accuracy of 97.37% in PCA feature selection scenarios with randomized search. The precision, recall, f1-score, MCC stand at 95.45%, 97.67%, 96.55% and 94.44% respectively.

## 5.5 Results of FS feature selection

Table 5.11: Results of FS feature selection without hyperparameter tuning

Algorithms	In Percentage (%)				
	Accuracy	Precision	Recall	F1-score	MCC
RF	96.49	1.00	90.70	95.12	92.66
AdaBoost	96.49	95.35	95.35	95.35	92.53
XGBoost	98.25	1.00	95.35	97.62	96.30
GradientBoost	96.49	97.56	93.02	95.24	92.53
Bagging	95.61	97.50	90.70	93.98	90.68
CatBoost	98.25	1.00	95.35	97.62	96.30

As depicted in Table [5.11], XGBoost and CatBoost achieved utmost accuracy of 98.25% in FS feature selection scenarios without hyperparameter tuning. The precision, recall, f1-score, MCC for both algorithms stand at 1.00%, 95.35%, 97.62% and 96.30% respectively.

Table 5.12: Results of FS feature selection with grid search

<b>Algorithms</b>	<b>In Percentage (%)</b>				
	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>MCC</b>
RF	95.61	97.50	90.70	93.98	90.68
AdaBoost	96.49	95.35	95.35	95.35	92.53
XGBoost	98.25	1.00	95.35	97.62	96.30
GradientBoost	95.61	95.24	93.02	94.12	90.64
Bagging	98.25	1.00	95.35	97.62	96.30
CatBoost	98.25	1.00	95.35	97.62	96.30

As depicted in Table [5.12], XGBoost, Bagging and CatBoost achieved utmost accuracy of 98.25% in FS feature selection scenarios with grid search. The precision, recall, f1-score, MCC for three algorithms stand at 1.00%, 95.35%, 97.62% and 96.30% respectively.

Table 5.13: Results of FS feature selection with randomized search

<b>Algorithms</b>	<b>In Percentage (%)</b>				
	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>MCC</b>
RF	97.37	1.00	93.02	96.39	94.47
AdaBoost	96.49	95.35	95.35	95.35	92.53
XGBoost	98.25	1.00	95.35	97.62	96.30
GradientBoost	95.61	95.24	93.02	94.12	90.64
Bagging	96.49	1.00	90.70	95.12	92.66
CatBoost	98.25	1.00	95.35	97.62	96.30

As depicted in Table [5.13], and CatBoost achieved utmost accuracy of 98.25% in FS feature selection scenarios with grid search. The precision, recall, f1-score, MCC for both algorithms stand at 1.00%, 95.35%, 97.62% and 96.30% respectively.

# Chapter 6

## Performance Comparison

### 6.1 Accuracy Comparison

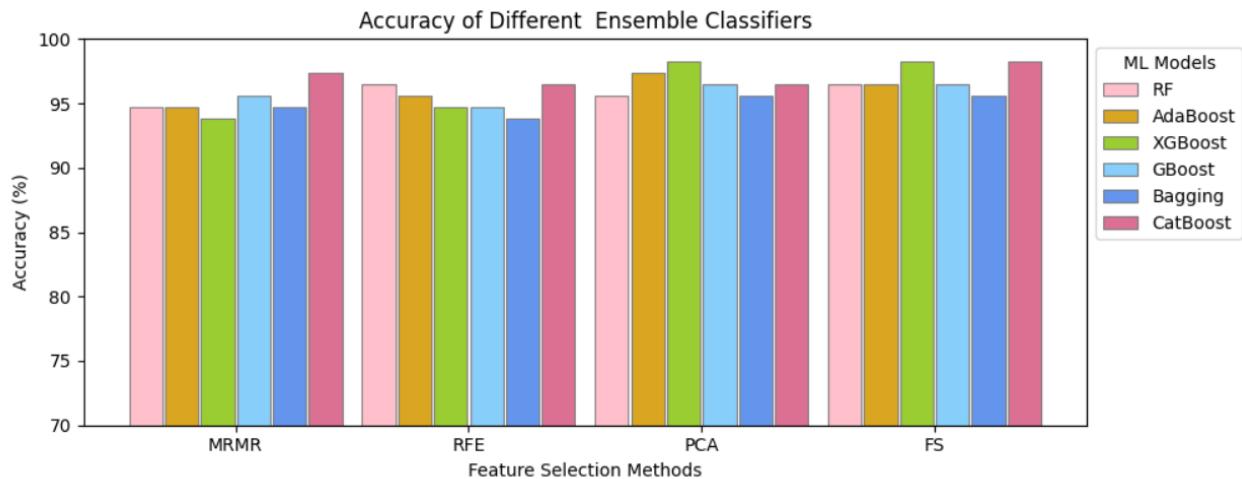


Figure 6.1: Accuracy without Hyperparameters

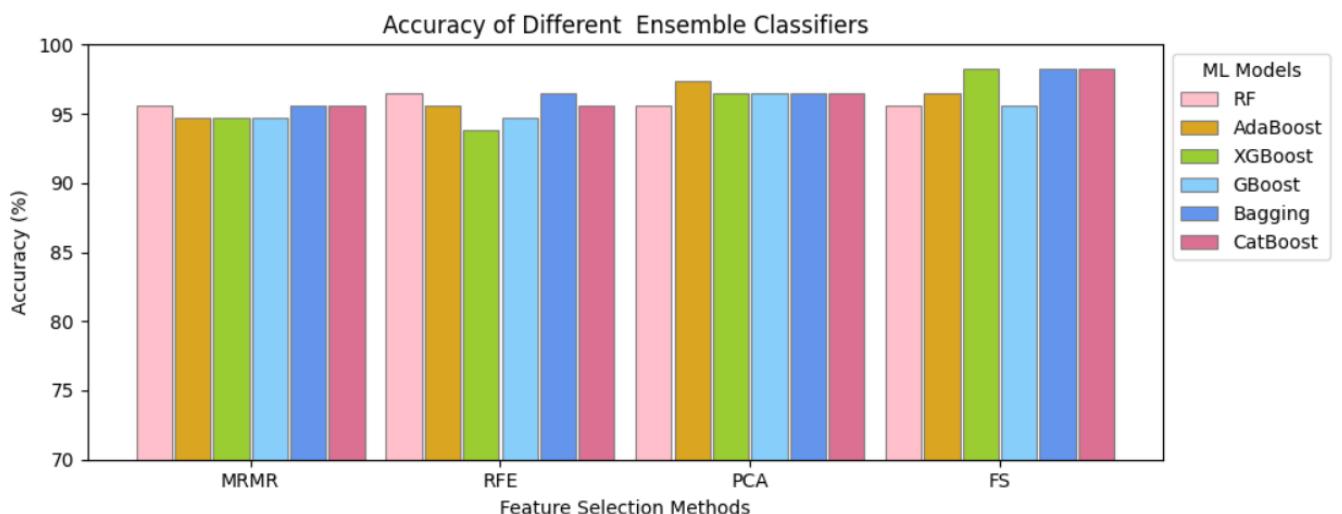


Figure 6.2: Accuracy using Grid Search

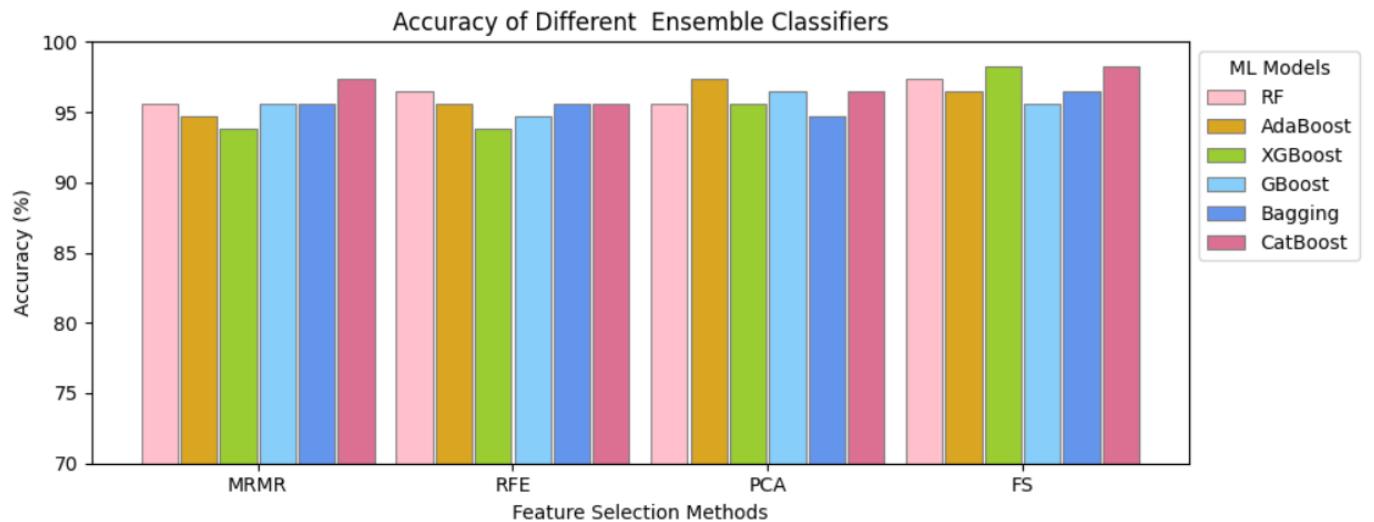


Figure 6.3: Accuracy using Randomized Search

The figures[6.1, 6.2, 6.3] illustrates the accuracy of each of the six algorithms without hyperparameter tuning, with Grid Search and randomized search, respectively.

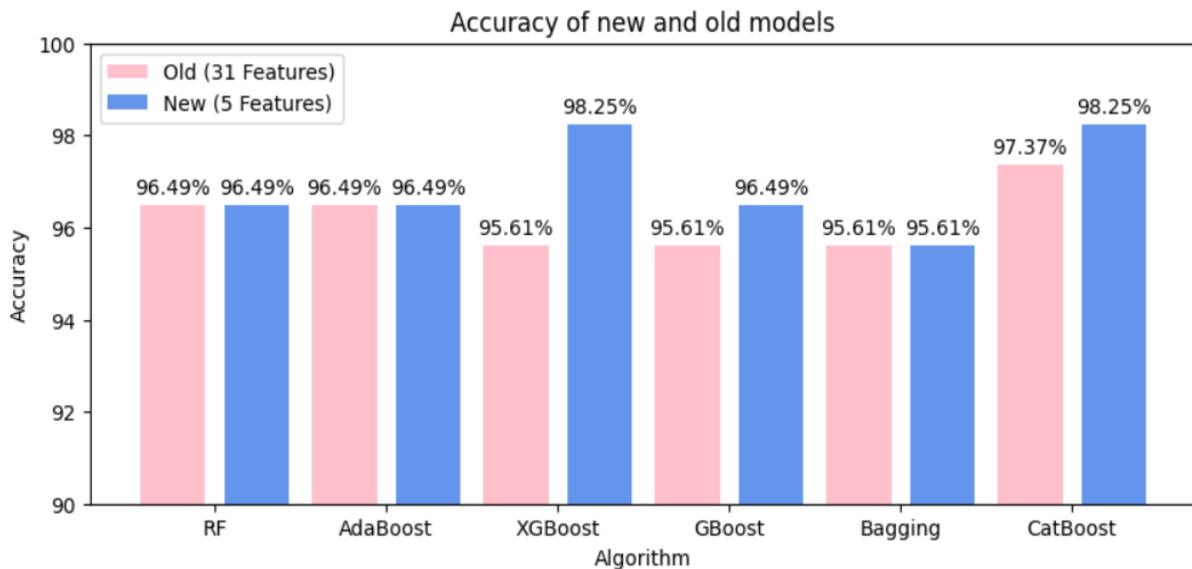


Figure 6.4: Comparison of accuracy with all features and selected features

Here, XGBoost and CatBoost both gave 98.25% accuracy and having same results for all performance metrics. We chose XGBoost for our project .With advanced regularization techniques like L1 and L2 regularization, it helps prevent overfitting, ensuring robustness in model predictions XGBoost is a highly efficient and flexible machine learning algorithm that excels in both accuracy and speed. Its wide adoption and strong community support ensure that you'll find plenty of resources for troubleshooting and improving your model, making it a reliable choice for both quick development and scalable solution.

## 6.2 Confusion Matrix

The confusion matrices for each of the six ensemble classifiers'(Random Forest,Gradient Boosting,AdaBoost , Bagging, XGBoost and CatBoost ) best outcomes are shown below:

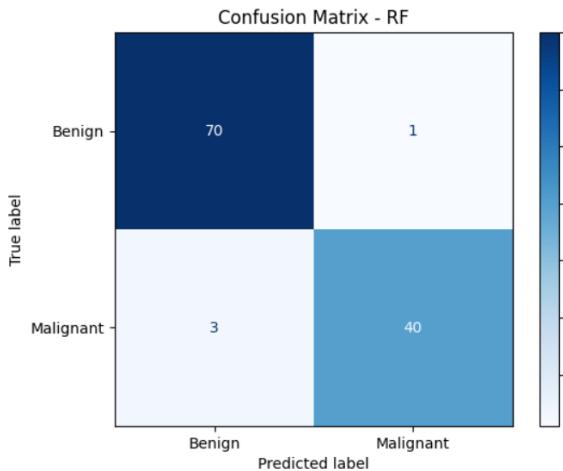


Figure 6.5: Random Forest's confusion matrix with FS using Randomized Search

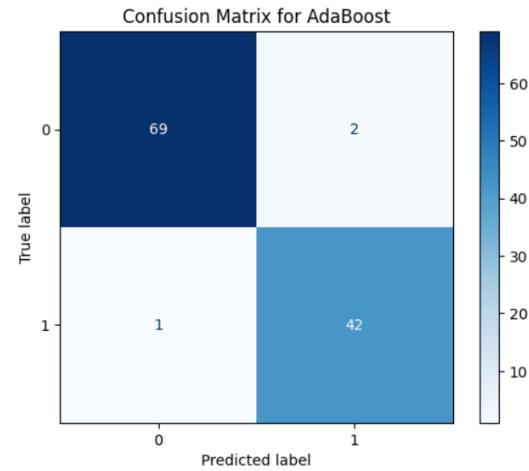


Figure 6.6: AdaBoost's confusion matrix with PCA using Grid Search

The figure[6.5] shows Random Forest classifier confusion matrix with FS after executing hyperparameter tuning using grid search. Following hyperparameter tuning with both grid search and randomized search, the RF achieved its best performance(Accuracy: 97.37%) with FS feature selection. Prior to hyperparameter tuning, the accuracy stood at 96.49%. This suggests that adjusting the hyperparameters improved the outcomes by 0.88%.

The figure[6.6] shows AdaBoost classifier confusion matrix with PCA after executing hyperparameter tuning using grid search. Following hyperparameter tuning with both grid search and randomized search, the PCA achieved its best performance(Accuracy: 97.37%) with FS feature selection. Prior to hyperparameter tuning, the accuracy stood at also 97.37%.

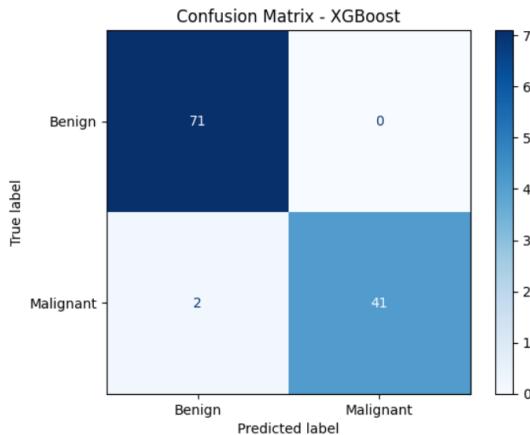


Figure 6.7: XGBoost's confusion matrix with FS using Grid Search

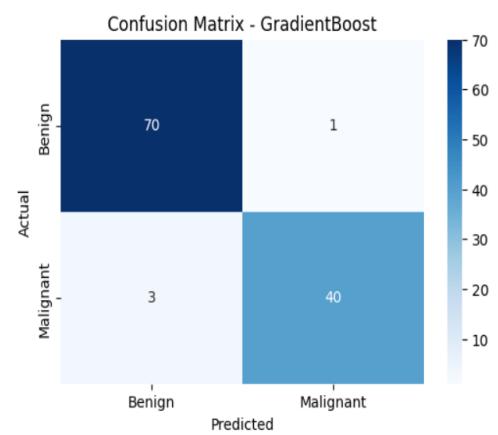


Figure 6.8: GradientBoost's confusion matrix with FS feature selection

The figure[6.7] shows XGBoost classifier confusion matrix with FS after executing hyperparameter tuning using grid search. Following hyperparameter tuning with both grid search and randomized search, the XGBoost achieved its best performance (Accuracy: 98.25%) with FS feature selection. Prior to hyperparameter tuning, the accuracy also stood at 98.25%.

The figure[6.8] shows Random GradientBoost classifier confusion matrix with FS a achieved its best performance(Accuracy: 96.49%) with FS feature selection.

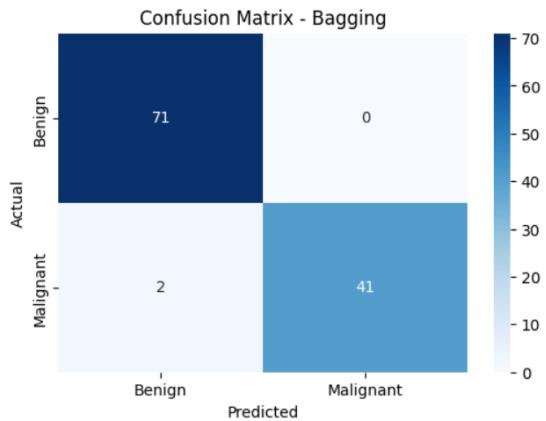


Figure 6.9: Bagging's confusion matrix with FS using Randomized Search

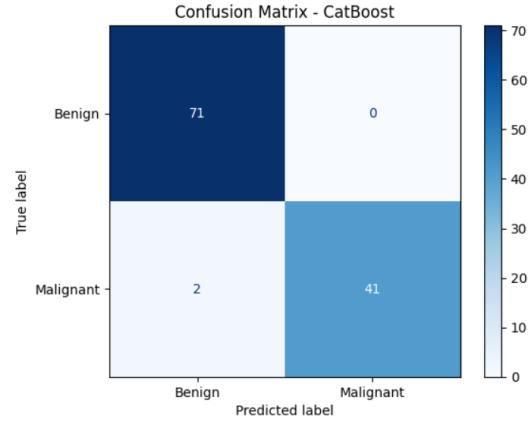


Figure 6.10: CatBoost's confusion matrix with FS using Grid Search

The figure[6.9] shows Bagging classifier confusion matrix with FS after executing hyperparameter tuning using randomized search. Following hyperparameter tuning with both grid search and randomized search, the Bagging achieved its best performance (Accuracy: 98.25%) with FS feature selection. Prior to hyperparameter tuning, the accuracy stood at 95.61%. This suggests that adjusting the hyperparameters improved the outcomes by 2.64%.

The figure[6.10] shows CatBoost classifier confusion matrix with FS after executing hyperparameter tuning using grid search. Following hyperparameter tuning with both grid search and randomized search, the XGBoost achieved its best performance (Accuracy: 98.25%) with FS feature selection. Prior to hyperparameter tuning, the accuracy also stood at 98.25%.

## 6.3 AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

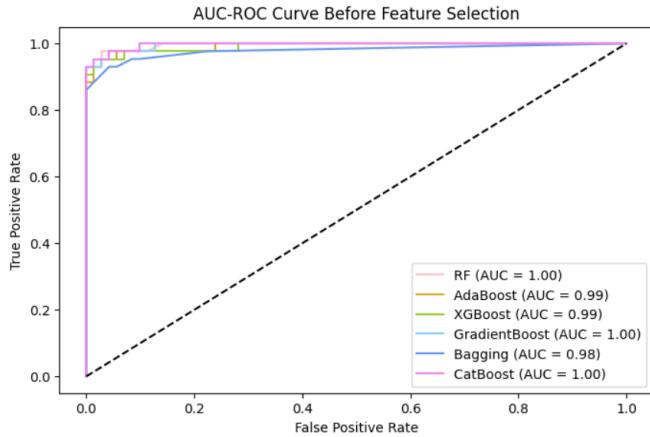


Figure 6.11: AUC-ROC Curve with all features

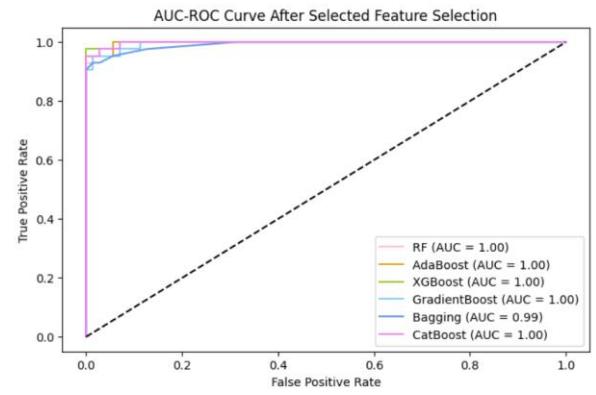


Figure 6.12: AUC-ROC Curve with 5 selected features

The increase in AUC-ROC score for AdaBoost and XGBoost from 0.99 to 1.00 after feature selection indicates a significant improvement in the model's ability to distinguish between classes. Before feature selection, the model was already performing exceptionally well, achieving a near-perfect ROC curve. After applying feature selection, the model likely benefited from the removal of irrelevant or redundant features, leading to a more focused and efficient model. This reduction in dimensionality can help the model generalize better, reducing overfitting and increasing its precision in classifying positive and negative cases. An AUC-ROC of 1.00 suggests that the model is now perfectly distinguishing between classes, achieving optimal performance.

## 6.4 Table of hyperparameters

**Table of Hyperparameter Tuning**

Machine Learning Algorithms	Hyperparameters	Feature Selection	Best Hyperparameter (Grid Search Space)	Best Hyperparameter (Random Search Space)
Random Forest	a) 'n_estimators': [50,100, 200], b) 'max_depth': [None, 10, 20], c) 'min_samples_split': [2, 5, 10],	mRMR	a) 100 b) 10 c) 5	a) 100 b) 10 c) 5
		RFE	a) 50 b) 10 c) 2	a) 1.0 b) 11 c) saga
		PCA	a) 100 b) None c) 2	a) 0.01 b) 11 c) saga
		FS	a) 100 b) 10 c) 5	a) 200 b) 20 c) 2
AdaBoost	a) 'n_estimators': [50,100,200], b) 'learning_rate': [0.01, 0.1, 1]	mRMR	a) 200 b) 1	a) 200 b) 1
		RFE	a) 50 b) 1	a) 50 b) 1
		PCA	a) 50 b) 1	a) 50 b) 1
		FS	a) 200 b) 1	a) 200 b) 1

Machine Learning Algorithms	Hyperparameters	Feature Selection	Best Hyperparameter (Grid Search Space)	Best Hyperparameter (Random Search Space)
XGBoost	a) 'n_estimators': [50,100, 200], b) 'learning_rate': [0.01, 0.1, 0.2], c) 'max_depth': [3,6,10,15],	mRMR	a) 50 b) 0.2 c) 6	a) 50 b) 0.2 c) 15
		RFE	a) 100 b) 0.1 c) 3	a) 100 b) 0.1 c) 3
		PCA	a) 50 b) 0.2 c) 3	a) 200 b) 0.1 c) 3
		FS	a) 50 b) 0.2 c) 6	a) 50 b) 0.2 c) 15
Gradient Boosting	a) 'n_estimators': [50,100, 200], b) 'learning_rate': [0.01, 0.1, 0.2,0.3], c) 'max_depth': [3, 5, 10],	mRMR	a) 100 b) 0.2 c) 3	a) 200 b) 0.3 c) 3
		RFE	a) 100 b) 0.1 c) 3	a) 200 b) 0.3 c) 3
		PCA	a) 100 b) 0.1 c) 3	a) 200 b) 0.1 c) 3
		FS	a) 100 b) 0.2 c) 3	a) – b) – c) –

Machine Learning Algorithms	Hyperparameters	Feature Selection	Best Hyperparameter (Grid Search Space)	Best Hyperparameter (Random Search Space)
Bagging	a) 'n_estimators': [10,50,100], b) 'max_samples':[0.5,0.7,1.0]	mRMR	a) 100 b) 0.5	a) 50 b) 0.5
		RFE	a) 100 b) 0.5	a) 10 b) 0.5
		PCA	a) 50 b) 0.7	a) 100 b) 1.0
		FS	a) 100 b) 0.5	a) 50 b) 0.5
CatBoost	a) 'iterations': [50,100, 200,500], b) 'learning_rate': [0.01, 0.1, 0.2], c) 'depth': [3,6,10])	mRMR	a) 100 b) 0.2 c) 10	a) 300 b) 0.01 c) 10
		RFE	a) 200 b) 0.1 c) 3	a) 200 b) 0.1 c) 3
		PCA	a) 500 b) 0.2 c) 10	a) 500 b) 0.01 c) 10
		FS	a) 100 b) 0.2 c) 10	a) 300 b) 0.01 c) 10

## Chapter 7

# Explainability and Deployment

## 7.1 Feature Importance

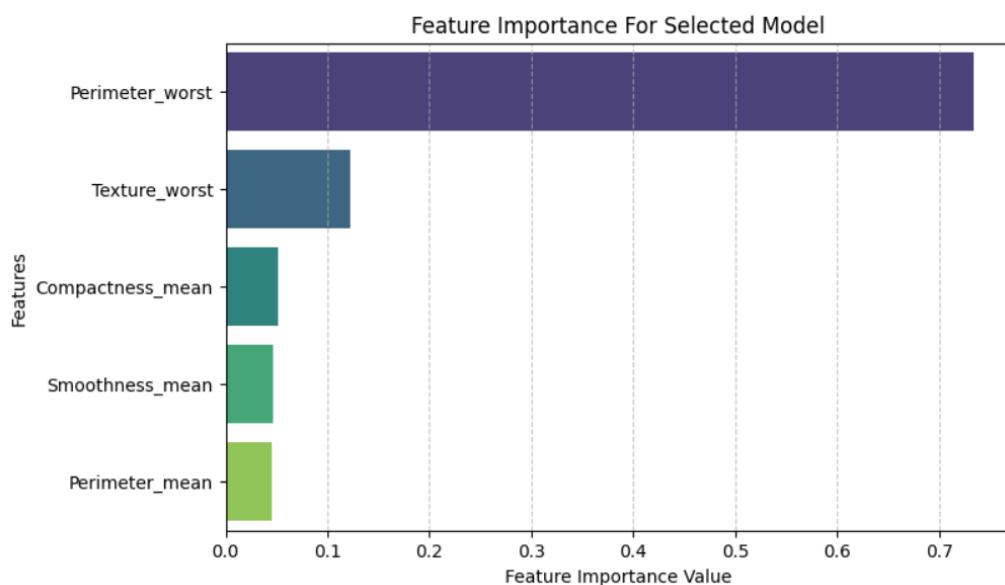


Figure 7.1: Feature Importance for selected model

The feature importance values provide insight into which attributes have the most influence on the model's predictions.

Among the top features, Perimeter\_worst stands out as the most significant, contributing the highest importance score of 0.7337, indicating that it plays a crucial role in distinguishing between the classes. Texture\_worst follows with an importance of 0.1229, suggesting it also has a notable impact on the model's decision-making. Features like Compactness\_mean, Smoothness\_mean, and Perimeter\_mean contribute less, with importance scores of 0.0515, 0.0460, and 0.0458, respectively.

## 7.2 SHAP

SHAP (SHapley Additive exPlanations) is used to explain the predictions of machine learning models by assigning importance values to each feature

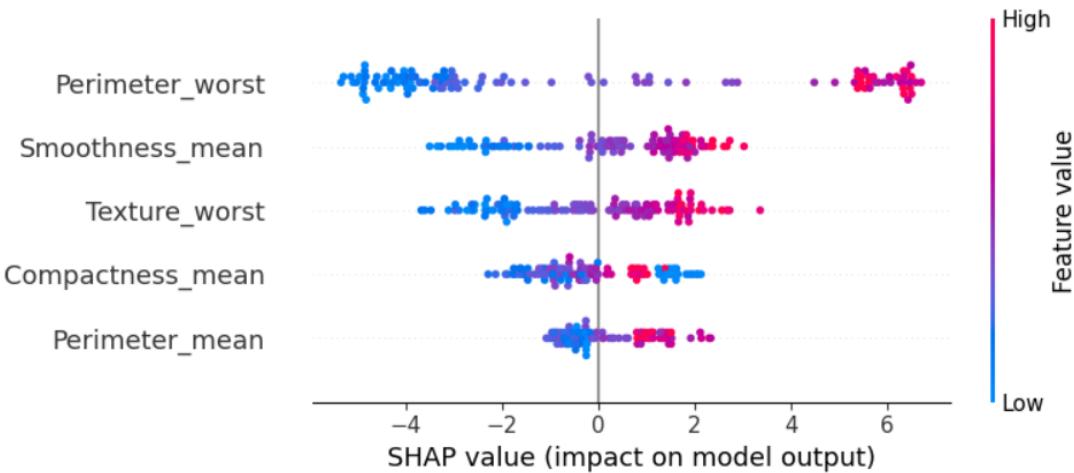


Figure 7.2: SHAP analysis for selected model

Features like Perimeter\_worst, Smoothness\_mean, Texture\_worst, Compactness\_mean, and Perimeter\_mean are among the most influential. Each dot represents a SHAP value for a single instance, colored by the feature value (blue for low, pink for high). For example, high values of Perimeter\_worst and Perimeter\_mean push the model toward a higher prediction output (likely indicating malignancy), while lower values of Smoothness\_mean or Compactness\_mean tend to have a negative SHAP impact. This plot offers a clear and interpretable view of how individual feature values drive the model's predictions in either direction.

## 7.3 Web App View

For Web app development here, we used streamlit. For coding development, we used to google colab and Microsoft Visual Studio.

The screenshots show the web application interface for 'Breast Cancer InsightX: AI-Powered Diagnostic Analysis'. The application has a sidebar on the left labeled 'Select Activity' with icons of butterflies and a dropdown menu. The main content area features a title 'Breast Cancer InsightX: AI-Powered Diagnostic Analysis' with a butterfly icon. Below the title is a list of features:

- AI-Powered Predictions : Classifies tumors as Benign or Malignant.
- Fast & Accurate : 98.25% accuracy with 5 diagnostic features.
- User-Friendly : Simple form-based design.
- Instant Results : Diagnosis in seconds.
- Secure : No personal data is stored.

Below the features is a box containing developer information:

Developers:  
Farhana Akter Suci (B190305001) & Rifah Sajida Deya (B190305004)  
CSE , JnU

The second screenshot shows the prediction form. It includes the same sidebar and title. The main content area contains a message: 'This application predicts whether a tumor is Benign or Malignant based on diagnostic features. Please enter the values below to get a prediction.' Below this are five input sliders for tumor features: Perimeter\_mean, Smoothness\_mean, Compactness\_mean, Texture\_worst, and Perimeter\_worst. A 'Predict' button is at the bottom.

Figure 7.3: The web app view

## **Chapter 8**

---

### **Conclusion**

This project demonstrates a comprehensive machine learning pipeline designed to enhance the diagnosis of breast cancer by integrating robust feature selection methods, powerful ensemble algorithms, and explainable AI techniques. The approach not only achieves high predictive accuracy but also ensures transparency in decision-making—an essential aspect in healthcare applications. By emphasizing interpretability alongside performance, the model supports clinical professionals in making informed decisions and paves the way for more trustworthy AI-driven diagnostics in the future.

---

## Reference

1. Breast Cancer Prediction Using Different Machine Learning Algorithms: A Comparative Study [<https://ieeexplore.ieee.org/document/10169729>]
2. Breast Cancer Detection Using Machine Learning Algorithms [<https://ieeexplore.ieee.org/document/10402232>]
3. Breast Cancer Prediction Using ML with Hyperparameter Tuning and Ensemble Learning and Explainability Analysis [<https://s24.ieeeecsbdc.org/papers/104-breast-cancer-prediction-using-ml-with-hyperparameter-tuning-and-ensemble-learning-and-explainability-analysis>]