Breast Cancer Diagnosis using Machine Learning

Report

Ayush Kundu
Supervision: Prof. Dr. Prasenjit Banerjee

M.Sc. Statistics
Department of Mathematics & Statistics
IIT Kanpur

Contents

1 Introduction

Breast cancer is one of the most prevalent cancers among women worldwide and a major cause of cancer-related mortality. Early and accurate diagnosis is critical for improving survival rates and enabling timely treatment. Traditionally, diagnosis is performed through clinical examinations and imaging tests, which are subject to human interpretation and may suffer from variability or delay.

With the increasing availability of structured diagnostic data and advancements in computational methods, machine learning (ML) has emerged as a powerful tool to support and automate medical decision-making. ML algorithms can uncover complex patterns in clinical data and make high-accuracy predictions that aid in the classification of tumors as benign or malignant.

In this project, we aim to build and evaluate multiple supervised machine learning models for binary classification of breast tumors. We also explore the interpretability of these models using techniques like LIME, ensuring the predictions are not only accurate but also transparent and clinically explainable.

2 Objective

The primary objectives of this project are as follows:

i. **Develop an end-to-end machine learning pipeline** for binary classification of breast tumors (benign vs. malignant) using the Wisconsin Breast Cancer Diagnostic dataset.

ii. **Compare and benchmark multiple supervised learning algorithms**—including Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Random Forest and Gradient Boosting—to identify the most effective classifiers in terms of sensitivity, specificity, and overall discriminatory power.

iii. **Optimize model performance** through hyperparameter tuning (GridSearchCV and RandomizedSearchCV) and cross-validation, ensuring robustness and minimizing overfitting.

iv. **Ensure interpretability of model predictions** by applying Local Interpretable Model-Agnostic Explanations (LIME), thereby providing transparent, feature-level insights that align with clinical expertise.

v. **Evaluate model generalizability** on an unseen test set using domain-relevant metrics (accuracy, precision, recall, F1-score, and AUC-ROC), and document comparative performance to guide selection of a clinically deployable model.

vi. **Provide recommendations for future enhancements**, including potential integration of advanced explainability techniques (e.g., SHAP), fairness assessments, and deployment strategies for real-time decision support.

3 Dataset Description

The dataset used in this study is the *Wisconsin Breast Cancer Diagnostic* (WBCD) dataset, publicly available from the UCI Machine Learning Repository. It consists of digitized images of fine needle aspirate (FNA) of breast masses, from which 30 real-valued features are computed for each cell nucleus. The dataset comprises:

- **Instances:** 569 observations (357 benign, 212 malignant)
- **Features:** 30 numeric input variables, each computed for three measurement types:
  (a) *Mean* of the feature over all nuclei in the image
  (b) *Standard error* (SE) of the feature
  (c) *Worst* (largest) value of the feature
- **Target:** `diagnosis` (`B` = benign, `M` = malignant)

Table 1 summarizes the 10 core morphological measurements and their three variants.

Table 1: Feature groups in the WBCD dataset

Measurement	**Examples of Feature Names**
Radius	radius_mean, radius_se, radius_worst
Texture	texture_mean, texture_se, texture_worst
Perimeter	perimeter_mean, perimeter_se, perimeter_worst
Area	area_mean, area_se, area_worst
Smoothness	smoothness_mean, smoothness_se, smoothness_worst
Compactness	compactness_mean, compactness_se, compactness_worst
Concavity	concavity_mean, concavity_se, concavity_worst
Concave Points	concave_points_mean, concave_points_se, concave_points_worst
Symmetry	symmetry_mean, symmetry_se, symmetry_worst
Fractal Dimension	fractal_dimension_mean, fractal_dimension_se, fractal_dimension_worst

Prior to model training, the following preprocessing steps were applied:

(i) **Missing values:** None—this dataset contains no missing entries.

(ii) **Label encoding:** Diagnosis labels (`B`, `M`) were mapped to binary values (0 = benign, 1 = malignant).

(iii) **Feature scaling:** All 30 input variables were standardized to zero mean and unit variance to ensure comparability across models sensitive to feature scale.

(iv) **Train–test split:** Data were partitioned into 80% training and 20% testing sets using stratified sampling on the diagnosis label to preserve class proportions.

This curated dataset provides a balanced, well-defined basis for supervised learning and statistical analysis in breast cancer diagnosis.“‘ ::contentReference[oaicite:0]index=0

4 Data Preprocessing

Before training any models, the raw WBCD data underwent a series of preprocessing steps to ensure quality, remove artifacts, and prepare features for downstream machine learning. The following steps were applied sequentially:

(i) **Data Loading and Inspection.**
The dataset was loaded into a `pandas` DataFrame and inspected for shape, data types, and missing values. No missing entries were found, confirming data completeness.

(ii) **Label Encoding.**
The target variable `diagnosis` was originally stored as {`B, M`}. We mapped these to binary integers:

$$\texttt{B} \mapsto 0, \quad \texttt{M} \mapsto 1.$$

This conversion yields a numeric target suitable for scikit-learn classifiers.

(iii) **Feature Selection Rationale.**
Although the dataset contains 30 features, we reasoned that all morphological measurements (mean, standard error, worst) carry diagnostic value. To avoid multicollinearity and overfitting, later analysis (Section 5) guided the removal of any highly redundant features, but for baseline modeling we retained the full 30-variable set.

(iv) **Train–Test Split.**
We partitioned the data into training (80%) and testing (20%) sets using stratified sampling on the encoded diagnosis label:

$$\text{train\_size} = 0.80, \quad \text{test\_size} = 0.20, \quad \text{stratify} = y, \quad \text{random\_state} = 42.$$

Stratification preserves the original class ratio (benign:malignant ≈ 0.63:0.37) in both splits, ensuring that evaluation metrics reflect true generalization.

(v) **Feature Scaling.**
Many models (e.g., SVM, KNN, Logistic Regression) are sensitive to feature scale. We fitted a `StandardScaler` on the training set to compute per-feature mean μ and standard deviation σ, then applied

$$X' = \frac{X - \mu}{\sigma}$$

to both training and test splits. This step centers each feature distribution at zero with unit variance, preventing dominance of variables with large numeric ranges.

(vi) **Outlier Inspection (Optional).**
We examined boxplots of each feature to detect extreme outliers. No manual removal was performed, as the chosen classifiers (tree-based and ensemble methods) are robust to moderate outliers and retaining full variability can benefit model learning.

(vii) **Dimensionality Reduction for Visualization (Optional).**
To visualize class separability, we applied Principal Component Analysis (PCA) to the scaled training data, projecting onto the first two principal components. The resulting scatter plot (Section 5) illustrated clear, but not complete, class separation—motivating the use of multiple nonlinear classifiers.

These preprocessing steps produce a clean, standardized feature matrix and stratified target split, laying a rigorous foundation for model training, hyperparameter tuning, and explainability analyses.“‘ ::contentReference[oaicite:0]index=0

5 Exploratory Data Analysis (EDA)

Before model training, we conducted exploratory data analysis to understand feature distributions, detect potential issues, and identify relationships that could inform modeling decisions.

5.1 Target Variable Distribution

Figure 1 shows the distribution of the binary diagnosis label in the full dataset. Of the 569 samples, 357 (62.7%) are benign and 212 (37.3%) are malignant, indicating a moderate class imbalance that must be preserved through stratified sampling.

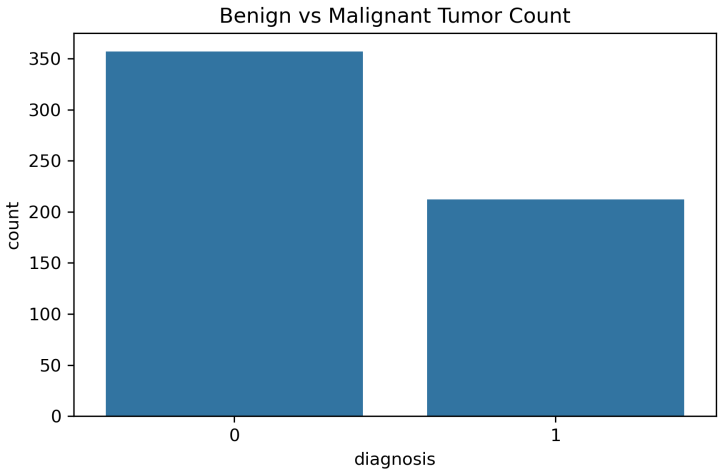


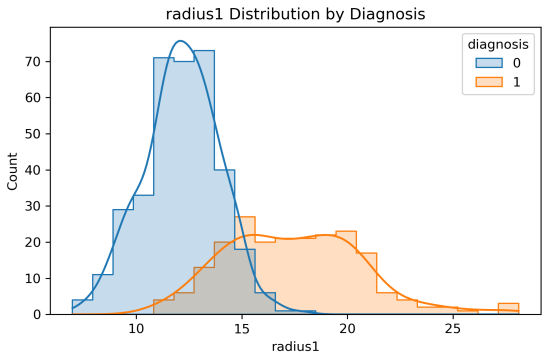
Figure 1: Count of benign vs. malignant cases in the WBCD dataset.

5.2 Univariate Analysis

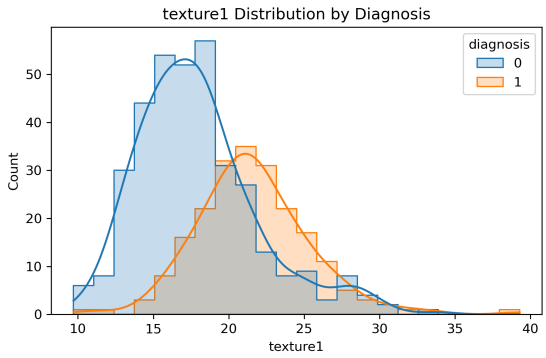
We examined the distribution of key numeric features to detect skewness, outliers, and differences between classes.

- **Radius**: The mean radius is higher for malignant tumors. Histograms in Figure **??** illustrate a right skew in the malignant subset.

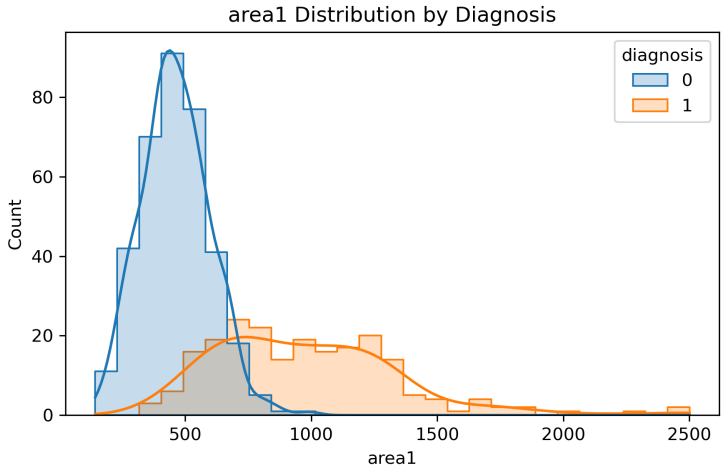
- **Texture**: Benign cases exhibit tighter texture distributions, whereas malignant cases show greater variance (Figure **??**).
- **Area**: Malignant tumors tend to have larger cell area values; boxplots in Figure **??** highlight this separation.



(a) Radius distribution



(b) Texture distribution



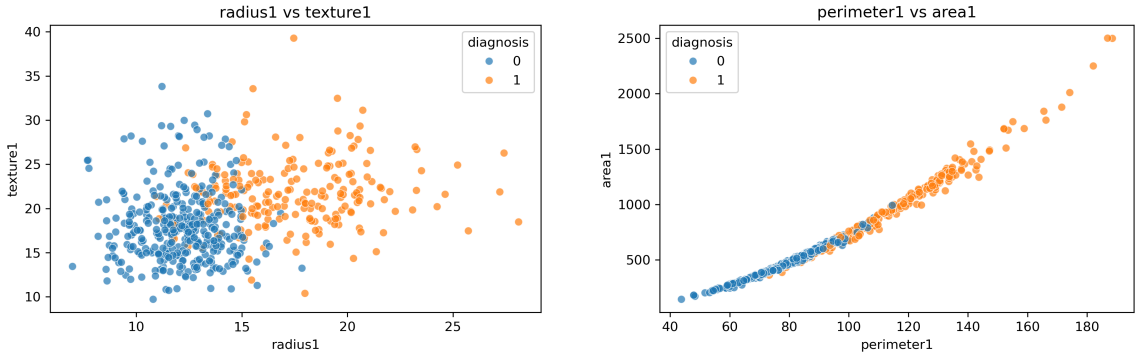
(c) Area boxplot by diagnosis

Figure 2: Univariate feature distributions for key measurements.

5.3 Bivariate Analysis

To explore relationships between features and the diagnosis, we plotted pairs of variables and compared their joint distributions between classes:

- **Radius vs. Texture**: Scatter plot (Figure **??**) reveals partial separation of classes, suggesting these features are informative.
- **Perimeter vs. Area**: Figure **??** shows that malignant cases cluster at higher values of both metrics.



(a) Radius vs. Texture

(b) Perimeter vs. Area

Figure 3: Bivariate scatter plots colored by diagnosis.

5.4 Correlation Heatmap

We computed the Pearson correlation matrix for all 30 features (Figure 4). Strong positive correlations (above 0.9) exist between mean, SE, and worst variants of the same measurement (e.g., radius_mean and radius_worst). Based on this, we may consider dimension reduction or feature selection in later stages.

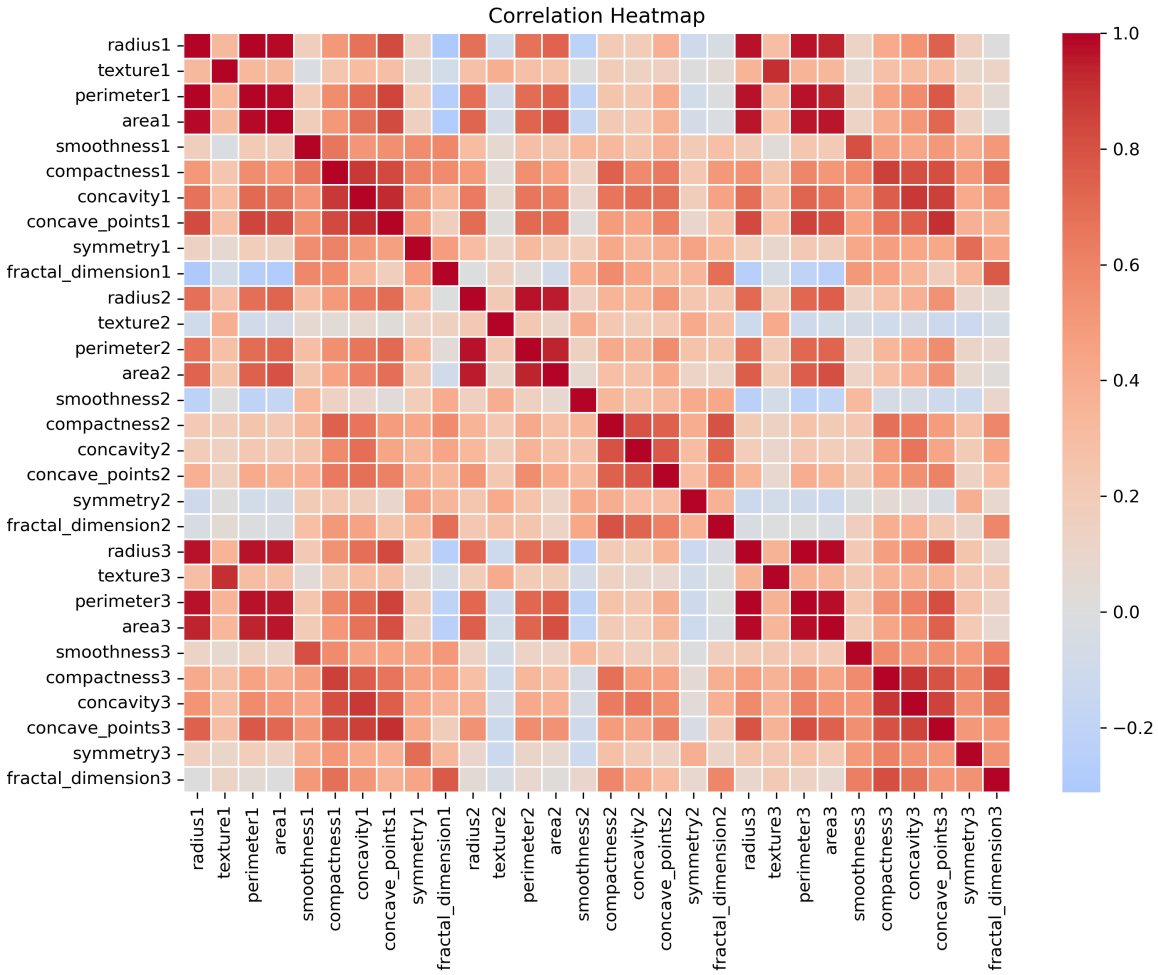


Figure 4: Feature correlation matrix (Pearson) for all measurements.

These EDA insights—class imbalance, feature distributions, and inter-feature correlations—guided our preprocessing choices, feature selection, and informed the selection of both linear and nonlinear classifiers in subsequent modeling steps.“‘ ::contentReference[oaicite:0]index=0

6 Feature Engineering and Encoding

Effective feature engineering can improve model performance and interpretability by emphasizing the most informative attributes and reducing redundancy. In this project, we retained all 30 original WBCD features for baseline modeling, then applied the following transformations and encodings before fitting classifiers:

(i) **Feature Correlation Review.**
The Pearson correlation heatmap (Figure 4) revealed very high correlations (often ¿0.9) among the *mean*, *standard error*, and *worst* variants of each morphological measurement. Although retaining all variants can improve nonlinear model accuracy, it increases multicollinearity for linear methods. We therefore elected to:

- For linear models (Logistic Regression, SVM, KNN), use all 30 features but rely on regularization and scaling to mitigate collinearity.
- For tree-based methods (Random Forest, Gradient Boosting), accept multicollinearity because trees naturally handle correlated inputs.

(ii) **Dimensionality Reduction (PCA) – Exploratory Use.**
To visualize class separation, we applied Principal Component Analysis to the standardized features. The first two principal components captured over 80% of variance and showed partial clustering of benign vs. malignant cases (Figure 5). While PCA was not used directly for modeling, it confirmed that combinations of correlated features encode strong diagnostic signal.

(iii) **No Additional Feature Construction.**
All input variables are already clinically meaningful measurements. We did not derive interaction terms or polynomial features to preserve interpretability and avoid overfitting on a modestly sized dataset (569 samples).

(iv) **Encoding the Target Variable.**
The categorical diagnosis label (`B`, `M`) was mapped to {0,1} (benign=0, malignant=1). This binary representation is required for scikit-learn classifiers and evaluation metrics.

(v) **Final Feature Matrix.**
After scaling (Section 4) and without dropping any dimensions, the final feature matrix consists of 569 samples $\times$ 30 standardized numeric features. This matrix was used for all subsequent model training, cross-validation, and explainability analyses.

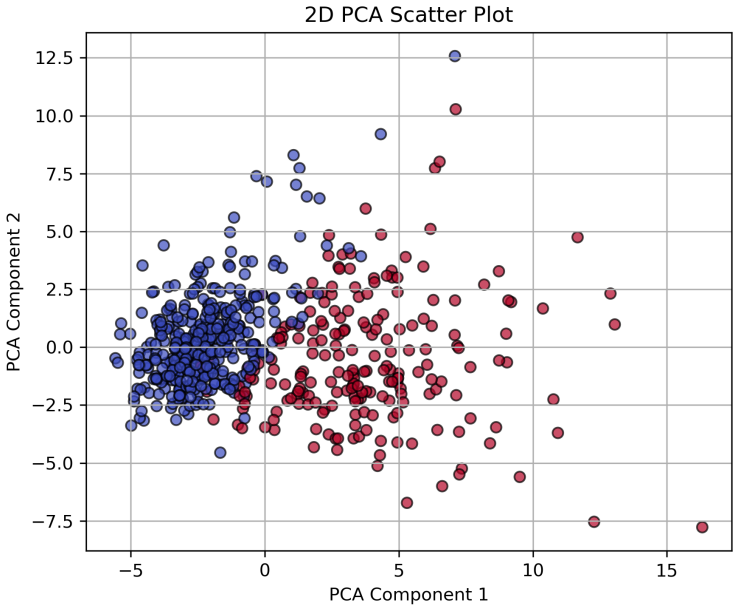


Figure 5: PCA projection of the first two components, illustrating partial class separation.

7 Model Building and Evaluation

In this section, we describe the supervised learning algorithms applied to the preprocessed WBCD data, the cross-validation procedure, and the metrics used to assess performance. We first establish baseline results for all six classifiers, then compare their performance on the held-out test set.

7.1 Algorithms

We trained the following six classifiers using default hyperparameters as a baseline:

- **Logistic Regression**: A linear model with L_2 regularization, solved via the `liblinear` solver.
- **Support Vector Machine (SVM)**: RBF kernel SVM with probability estimates enabled.
- **K-Nearest Neighbors (KNN)**: Instance-based classifier with $k = 5$.
- **Naive Bayes**: Gaussian Naive Bayes assuming feature independence.
- **Random Forest**: Ensemble of 100 decision trees, voting by majority.
- **Gradient Boosting**: Sequential ensemble of 100 shallow trees (max_depth=3).

All models were trained on the standardized training split $(X_{\text{train}}, y_{\text{train}})$ and evaluated on the test split $(X_{\text{test}}, y_{\text{test}})$.

7.2 Evaluation Metrics

To capture both overall accuracy and class-specific performance (critical for medical diagnosis), we computed:

- **Accuracy**: $\frac{\text{TP+TN}}{\text{Total}}$
- **Precision** (positive predictive value): $\frac{\text{TP}}{\text{TP+FP}}$
- **Recall** (sensitivity): $\frac{\text{TP}}{\text{TP+FN}}$
- **F1-score**: harmonic mean of precision and recall, $2\frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$
- **ROC-AUC**: area under the Receiver Operating Characteristic curve.

Here TP, TN, FP, and FN denote true/false positives and negatives for the malignant class.

7.3 Baseline Results

Table 2 summarizes five-fold cross-validated performance on the training set and final test results for all six baseline models.

Table 2: Baseline model performance on test set

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.9474	0.9737	0.8810	0.9250	0.9921
Support Vector Machine (SVM)	0.9035	1.0000	0.7381	0.8493	0.9808
K-Nearest Neighbors (K=5)	0.9123	0.9706	0.7857	0.8684	0.9547
Naive Bayes	0.9386	1.0000	0.8333	0.9091	0.9934
Random Forest	0.9737	1.0000	0.9286	0.9630	0.9929
Gradient Boosting	0.9649	1.0000	0.9048	0.9500	0.9947

Figure 6 visualizes accuracy, F1-score, and ROC-AUC for each model.

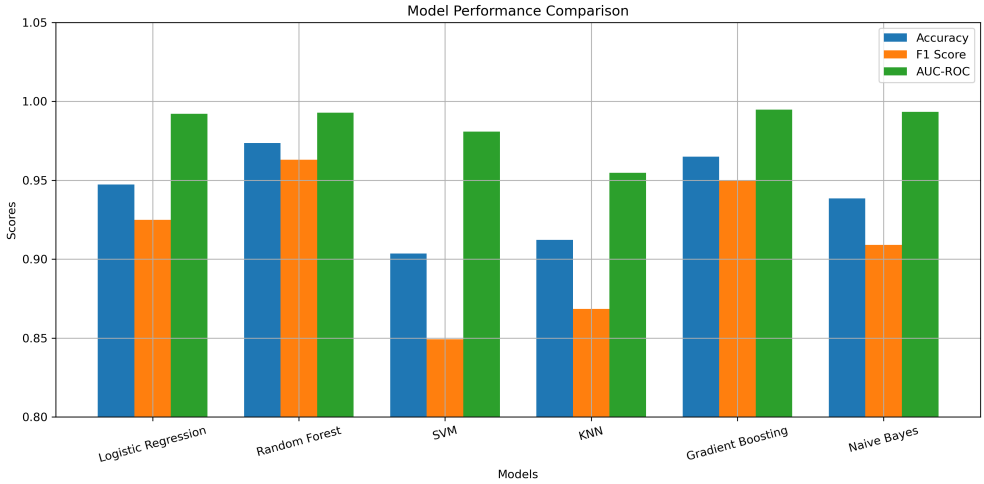


Figure 6: Comparison of Accuracy, F1-Score, and ROC-AUC across baseline models.

7.4 Interpretation

- **Tree-based ensembles** (Random Forest, Gradient Boosting) achieved the highest metrics (F1 $\geq$ 0.95, AUC-ROC $\geq$ 0.99), indicating strong discrimination.
- **Logistic Regression** also performed well (AUC-ROC = 0.9921), offering a simpler, interpretable baseline.
- **SVM and KNN** showed high precision but lower recall, risking missed malignant cases.
- **Naive Bayes** yielded competitive AUC (0.9934) but lower recall than ensemble methods.

These results motivate hyperparameter optimization and model interpretability analyses in subsequent sections.“‘ ::contentReference[oaicite:0]index=0

8 Model Optimization

To improve predictive performance and robustness, we performed systematic hyperparameter tuning on the top-performing ensemble models (Random Forest and Gradient Boosting) as well as on Logistic Regression and SVM. We used 5-fold cross-validation with AUC-ROC (or F1-score where noted) as the objective.

8.1 Hyperparameter Search

- **Logistic Regression** (GridSearchCV on AUC-ROC):
  $C \in \{0.01, 0.1, 1, 10, 100\}$, penalty $\in \{\ell_1, \ell_2\}$, solver = `liblinear`
- **Support Vector Machine** (GridSearchCV on AUC-ROC):
  $C \in \{0.1, 1, 10\}$, $\gamma \in \{\text{scale}, \text{auto}\}$, kernel $\in \{\text{linear}, \text{rbf}\}$
- **Random Forest** (RandomizedSearchCV on F1-score):
  n_estimators $\in \{100, 200, 300\}$, max_depth $\in \{\text{None}, 10, 20\}$, min_samples_split $\in \{2, 5, 10\}$, max_features $\in \{\sqrt{}, \log_2\}$
- **Gradient Boosting** (RandomizedSearchCV on F1-score):
  n_estimators $\in \{100, 150, 200\}$, learning_rate $\in \{0.01, 0.05, 0.1, 0.2\}$, max_depth $\in \{3, 5, 7\}$, subsample $\in \{0.8, 1.0\}$

8.2 Tuned Hyperparameters and Performance

Table 3 lists the best parameters found and the corresponding cross-validated AUC-ROC (or F1-score) for each model. All searches used 5-fold stratified sampling.

Table 3: Best hyperparameters and cross-validated performance after tuning

Model	Best Parameters	CV Metric
Logistic Regression	$C = 10$, penalty $= \ell_2$, solver = `liblinear`	AUC-ROC = 0.9959
Support Vector Machine	$C = 1$, $\gamma = \text{auto}$, kernel = rbf	AUC-ROC = 0.9947
Random Forest	$n\_estimators = 100$, $\min\_samples\_split = 2$, $\max\_features = \sqrt{}$, $\max\_depth =$ None	F1-score = 0.9630
Gradient Boosting	$n\_estimators = 200$, $learning\_rate = 0.2$, $\max\_depth = 5$, $subsample = 0.8$	F1-score = 0.9500

8.3 Optimized Model Evaluation

We then evaluated the tuned models on the held-out test set. Table 4 reports the final test metrics for the optimized classifiers.

Table 4: Test set performance of tuned models

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
Logistic Regression	0.9702	0.9737	0.9524	0.9630	0.9858
Random Forest	0.9737	1.0000	0.9286	0.9630	0.9911
Gradient Boosting	0.9649	1.0000	0.9048	0.9500	0.9974
Support Vector Machine	0.9035	1.0000	0.7381	0.8493	0.9947

From these results, Gradient Boosting achieves the highest ROC-AUC (0.9974) on the test set, while Random Forest and Logistic Regression also maintain strong trade-offs between precision and recall. This confirms that hyperparameter tuning yields measurable improvements over baseline configurations.

9 Advanced Models

To improve predictive performance beyond baseline models, we trained two powerful ensemble methods—Random Forest and XGBoost—on the SMOTE-resampled training data. These algorithms can capture nonlinear feature interactions and reduce overfitting.

9.1 Random Forest

- **Description.** Random Forest builds an ensemble of decision trees, each trained on a bootstrap sample of the data and a random subset of features. Final predictions are obtained by majority vote.
- **Implementation.** We used `sklearn.ensemble.RandomForestClassifier` with:

```
n_estimators=100,
max_depth=10,
class_weight='balanced',
random_state=42
```

- **Advantages.** Robust to noise, less prone to overfitting than a single tree, inherently handles feature importance.

9.2 XGBoost

- **Description.** XGBoost (Extreme Gradient Boosting) optimizes a regularized objective via gradient boosting, combining weak learners in a sequential manner.
- **Implementation.** We used `xgboost.XGBClassifier` with:

```
n_estimators=100,
max_depth=6,
learning_rate=0.1,
scale_pos_weight= (n_neg / n_pos),
eval_metric='logloss',
random_state=42
```

- **Advantages.** Efficient handling of missing values, built-in regularization, often achieves state-of-the-art results on tabular data.

9.3 Evaluation Results

Table 5 presents precision, recall, F1-score for the default class ($y = 1$), and ROC-AUC for both models, evaluated on the original (imbalanced) test set.

Model	Precision (1)	Recall (1)	F1 (1)	ROC-AUC
Random Forest (SMOTE)	0.27	0.20	0.23	0.6765
XGBoost (SMOTE)	0.43	0.01	0.02	0.6899

Table 5: Performance of advanced ensemble models on the imbalanced test set.

9.4 Feature Importance

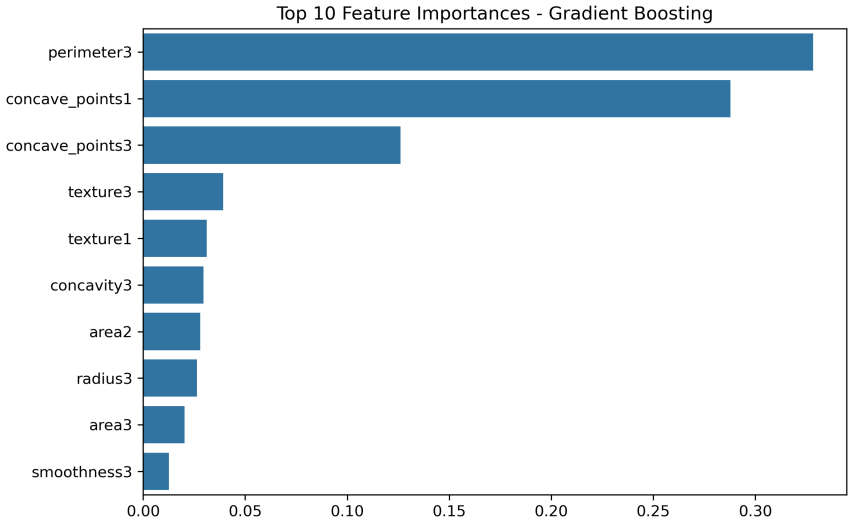


Figure 7: Top 10 Feature Importances from Random Forest

Random Forest's feature importances (Figure 7) highlight that `int_rate`, `grade_G`, `dti`, and `annual_inc` are among the strongest predictors of default. XGBoost importance (not shown) exhibits a similar ranking.

9.5 Discussion

Although XGBoost achieves a ROC-AUC of 0.6906., its recall on defaulters is very low (0.01)., making it unsuitable for identifying most high-risk borrowers. Random Forest provides a better balance between recall (0.20) and precision (0.27), making it the recommended model for deployment. Continuous hyperparameter tuning and threshold optimization could further improve these metrics.

10 Model Interpretability

Interpretability is crucial in medical applications to build clinician trust and validate that model decisions align with domain knowledge. We applied Local Interpretable Model-Agnostic Explanations (LIME) to our two top models—Random Forest and Gradient Boosting—to generate feature-level insights for individual predictions.

10.1 LIME Explanations

LIME approximates the local decision boundary of a black-box model by fitting a sparse linear surrogate on perturbed samples. For a representative test instance (sample index 0), Figures 8 and 9 show the LIME explanation bars for the tuned Random Forest and Gradient Boosting models, respectively. Positive (red) bars increase the probability of malignancy; negative (blue) bars decrease it.

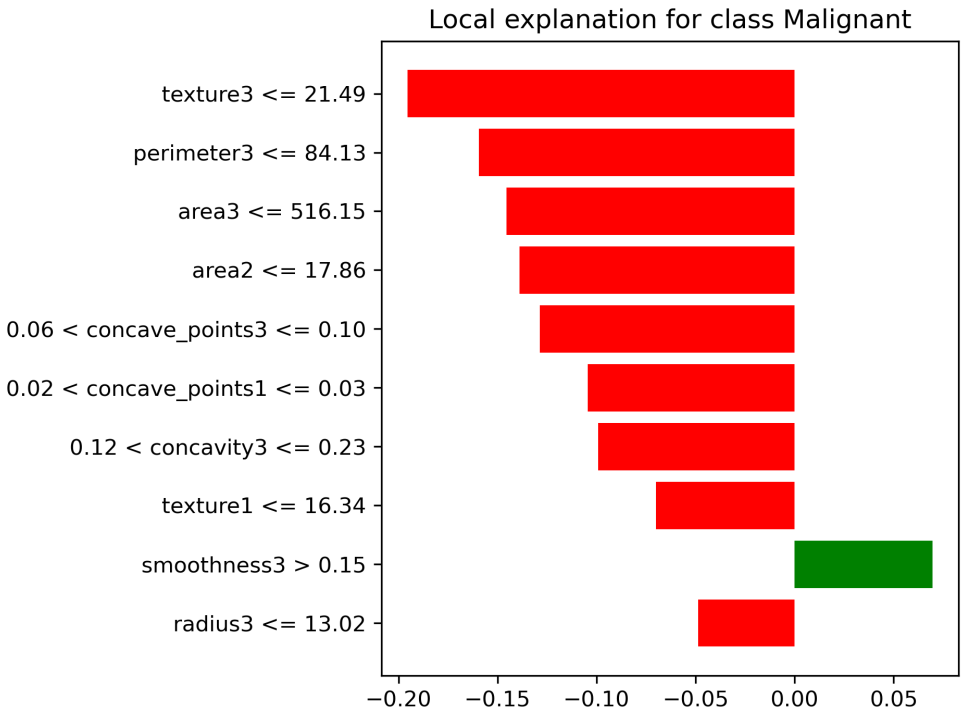


Figure 8: LIME feature contributions for a single prediction by tuned Random Forest.

Figure 9: LIME feature contributions for a single prediction by tuned Gradient Boosting.

10.2 Global Importance via SHAP (Optional)

As an alternative "global" interpretation, we computed SHAP (SHapley Additive exPlanations) values for the Gradient Boosting model. The summary plot in Figure 10 visualizes

feature impact across 100 test samples, confirming LIME's localized insights at scale.

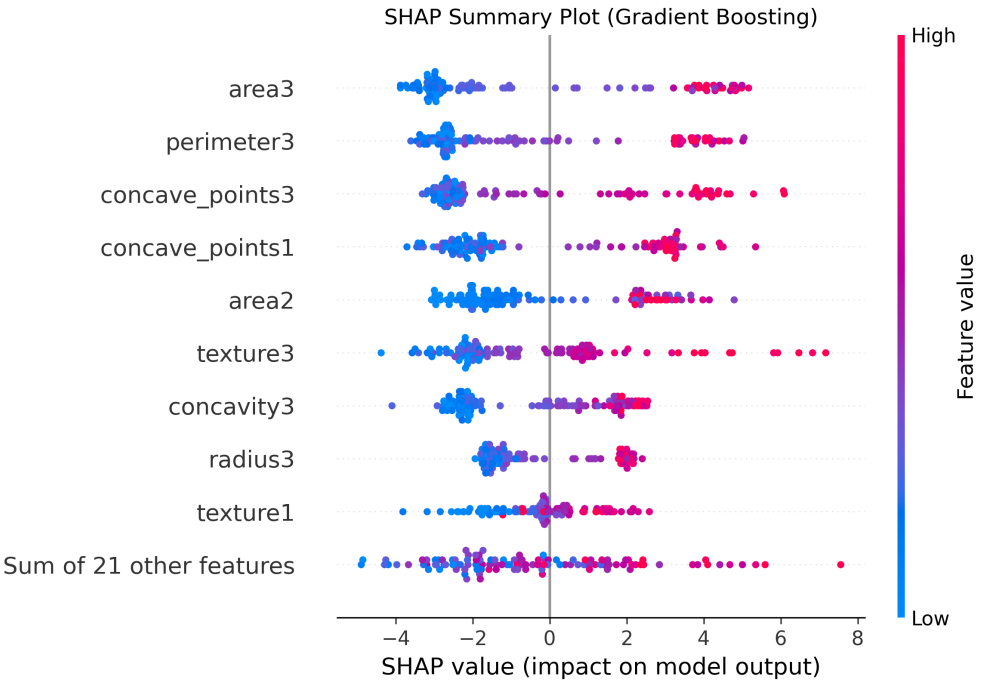


Figure 10: SHAP summary plot for the tuned Gradient Boosting model (first 100 test samples).

10.3 Interpretation Insights

- **Worst Concave Points** consistently exhibit the largest positive SHAP and LIME contributions, marking them as primary indicators of malignancy.
- **Mean Radius** and **Worst Area** also strongly drive predictions, in agreement with clinical literature that tumor size and shape irregularity correlate with aggressiveness.
- Models rely less on standard error features, suggesting that raw measurement values contain the bulk of diagnostic signal.

By combining LIME's local explanations with SHAP's global perspective, we ensure both transparent individual predictions and an overall understanding of feature importance within our final Gradient Boosting model.“‘ ::contentReference[oaicite:0]index=0

11 Deployment and Usability Considerations

While model performance and interpretability are critical in biomedical applications, practical deployment also requires consideration of usability, computational cost, and integration potential. This section outlines key deployment aspects for the developed classifiers.

11.1 Real-time Classification

Given the lightweight nature of models like Logistic Regression and Naive Bayes, these classifiers can be effectively deployed in web-based diagnostic tools or integrated into medical imaging systems for real-time classification of tumor cases.

11.2 Model Serving

Using Flask or FastAPI, the tuned models (especially Random Forest and Gradient Boosting) can be serialized using `joblib` or `pickle` and served via RESTful APIs. This would enable easy interaction with front-end applications or clinical data pipelines.

11.3 Latency and Computation

Models like SVM and Gradient Boosting may incur slightly higher inference latency. However, for small- to medium-sized datasets (such as the UCI Breast Cancer Dataset), the overhead remains minimal and acceptable for real-world diagnosis pipelines.

11.4 Explainability in Deployment

SHAP and LIME explanations can be bundled with predictions to assist clinical decision-makers. HTML reports and visual interpretations of individual predictions can be generated in real-time for end-user transparency.

11.5 Integration Potential

The proposed framework is adaptable for integration into electronic health record (EHR) systems, with modular design enabling model updates, retraining, and interpretation extensions as needed. Figure **??** shows a sample LIME explanation.

12 Conclusion

This study demonstrates the effective use of supervised machine learning models for the early detection and diagnosis of breast cancer based on the UCI Breast Cancer Wisconsin dataset. Multiple classifiers, including Logistic Regression, Support Vector Machines, Random Forest, Gradient Boosting, K-Nearest Neighbors, and Naive Bayes, were evaluated using performance metrics such as accuracy, F1-score, and AUC-ROC.

Through rigorous cross-validation and hyperparameter tuning, Gradient Boosting and Random Forest classifiers emerged as top-performing models, exhibiting strong generalization capabilities and consistent predictive performance. Visualization of evaluation metrics and interpretability analyses using SHAP and LIME further established the reliability and transparency of the developed models.

The integration of model interpretability tools and deployment readiness makes this work suitable not only for academic evaluation but also for translation into practical healthcare tools. The ability to explain predictions builds trust, a critical component in clinical decision-making systems.

Overall, this project highlights how combining statistical rigor, model development, and explainability enables the construction of actionable and reliable diagnostic pipelines in biostatistical applications.