Report - I: Assignment VI

Name	Shivi Goyal
Reg. ID	2025SIP004

***NOTE:** Write the contents of the report inside this box. Caption the additional images and embed them at the end of the file. Use proper referencing.*

Abstract

This report presents a machine learning-based approach to classify tumors as malignant or benign using the Breast Cancer Wisconsin (Diagnostic) Dataset. The workflow includes data acquisition, visualization, preprocessing, dimensionality reduction using PCA, model training, evaluation of multiple algorithms, and deployment of the best-performing model. Among all tested models, Random Forest achieved the highest accuracy, while k-NN performed the worst due to sensitivity to outliers and high prediction cost.

Introduction

Breast cancer diagnosis is a critical field where machine learning can provide valuable support in early detection and classification. The goal of this project is to apply several machine learning classifiers to a real-world breast cancer dataset and determine the most suitable model for deployment. The project involves not only classification, but also visualization, dimensionality reduction, and model evaluation to ensure robustness.

Dataset Description and Preprocessing

The dataset used was the Breast Cancer Wisconsin (Diagnostic) dataset, obtained using the ucimlrepo Python library (Dataset ID: 17). It contains 569 samples and 30 numeric features extracted from digitized images of fine needle aspirates (FNA) of breast masses. The target variable is Diagnosis (M = Malignant, B = Benign).

Data Analysis

i. Outlier Analysis:
From the boxplots of selected features like radius1, area1, and perimeter1, we observed several high-value outliers. These outliers likely represent tumors with significantly larger sizes. While they are valid observations, they may affect models that are sensitive to feature scale or distance (e.g., k-NN). Therefore, standardization or robust scaling is important to reduce their impact during training.

ii. Data Analysis:
I used histogram to check skewness,correlation heatmap to check which features are strongly related, df.describe() to check stats (mean, std, min, max, etc).
The dataset consists of 569 samples and 30 numerical features. The target variable, Diagnosis, has two classes — benign (B) and malignant (M) — with benign cases being more frequent. Many

features are strongly correlated, especially the size-related ones such as radius1 and area1. The presence of skewed distributions and correlations suggests that preprocessing steps like normalization and dimensionality reduction could improve model performance and training stability.

iii. Principal Component Analysis (PCA):
PCA was applied to reduce dimensionality and visualize the structure of the data. After scaling the features, a 2D PCA plot showed a fairly clear separation between malignant and benign tumors. This suggests that the features contain meaningful patterns. The first two principal components explained a significant portion of the variance, confirming that PCA can be a useful tool to simplify the dataset while preserving most of the important information.

iv. Train-Test-Validation Split:
To ensure reliable model training and evaluation, the data was split into three parts: 70% for training, 15% for validation, and 15% for testing. Stratified sampling was used to maintain the original class distribution across all splits. This helps prevent bias during evaluation and supports fair hyperparameter tuning during model development.

Model Evaluation and Comparison

i. Performance

Each classifier was evaluated based on its predictive accuracy on the validation set. The key findings are summarized below:

1) Logistic Regression demonstrated strong performance on this dataset, particularly due to the linearly separable nature of the classes. It achieved high accuracy and provided interpretable coefficients.

2) Support Vector Machine (SVM) also showed excellent accuracy, benefiting from its ability to create a maximum-margin decision boundary. However, its performance was slightly more sensitive to parameter tuning.

3) Random Forest and Extra Trees classifiers both performed very well, with Random Forest slightly more stable and interpretable. These ensemble methods reduced overfitting by aggregating the results of multiple decision trees.

4) Naive Bayes, while based on the strong assumption of feature independence, yielded competitive results and served as an effective lightweight baseline.

5) k-Nearest Neighbors (k-NN) showed decent accuracy, but its performance degraded slightly in the presence of noisy or correlated features.

6) Decision Tree had moderate accuracy but showed signs of overfitting due to its tendency to model noise unless properly pruned.

ii. Efficiency

Efficiency was assessed in terms of training time and computational simplicity:

1) Naive Bayes and Logistic Regression were the most efficient, both in terms of model

training and resource usage. These models are suitable for rapid prototyping and deployment.

2) Decision Tree also trained quickly, though it required more careful tuning to avoid overfitting.
3) Random Forest and Extra Tree were relatively efficient considering they are ensemble models; however, they required more memory and time compared to simpler classifiers.
4) SVM was computationally heavier, especially with larger datasets, due to the complexity of solving the optimization problem.
5) k-NN, while quick to fit, was the slowest during prediction, as it requires computing distances to all training samples at inference time.

iii. Computational Complexity

Theoretical complexity and scalability were compared as follows:

1) Logistic Regression and Naïve Bayes are both linear in time complexity and scale well with dataset size.
2) Decision Tree has a complexity of $O(n \cdot \log n)$ and scales efficiently, but deep trees can become computationally expensive without pruning.
3) k-NN has low training cost but high prediction cost: $O(n \cdot d)$, where n is the number of training samples and d is the number of features.
4) SVM has a complexity between $O(n^2)$ and $O(n^3)$ in the worst case, making it computationally intensive for large datasets.
5) Random Forest and Extra Trees scale better than individual decision trees due to parallelism but require more resources due to the number of trees involved.

Overall, Random Forest and SVM offered the best balance of accuracy and generalization. Naive Bayes was the most efficient in terms of training time, while Logistic Regression stood out for its interpretability. Extra Trees delivered comparable accuracy with slightly better speed than Random Forest, though at the cost of reduced interpretability.

Best Performing Algorithm: Random Forest Classifier

The **Random Forest** model achieved the highest validation accuracy and demonstrated consistent, reliable performance across different metrics (precision, recall, F1-score). Its ensemble nature helped reduce overfitting, which was observed in the single decision tree model. Additionally, it handled correlated and non-linear features effectively due to its ability to average across multiple randomized trees.

Reasons for selection:

- Highest accuracy on validation data

- Low variance and good generalization
- Robust to outliers and noisy features
- Handles multicollinearity and feature importance naturally

Least Performing Algorithm: k-Nearest Neighbors (k-NN)

The **k-NN** model performed noticeably worse than other classifiers. Its accuracy was lower, and it was more sensitive to the choice of k, scaling of features, and presence of outliers. Furthermore, it was computationally inefficient during prediction, as it requires distance calculations against the entire training dataset.

Reasons for selection:

1) Lower validation accuracy compared to others
2) Sensitive to feature scaling and noise
3) Slow prediction time due to distance computation
4) Poor generalization in higher-dimensional space

Deployment and Prediction

After selecting Random Forest as the best model, it was saved using joblib and used to classify a new patient record provided. The output was 'B', indicating a benign tumor. This confirms the model's applicability in real-world scenarios.

Conclusion

This project demonstrates a complete machine learning workflow, including data analysis, model training, evaluation, and deployment. Among the seven models tested, Random Forest proved to be the most accurate, interpretable, and robust. This assignment highlighted the practical importance of preprocessing, model comparison, and validation in medical machine learning applications.

Figures (Embedded Below):

Figure 1: 2D PCA Visualization of Tumor Classes

Shows: Shows class separability after dimensionality reduction.

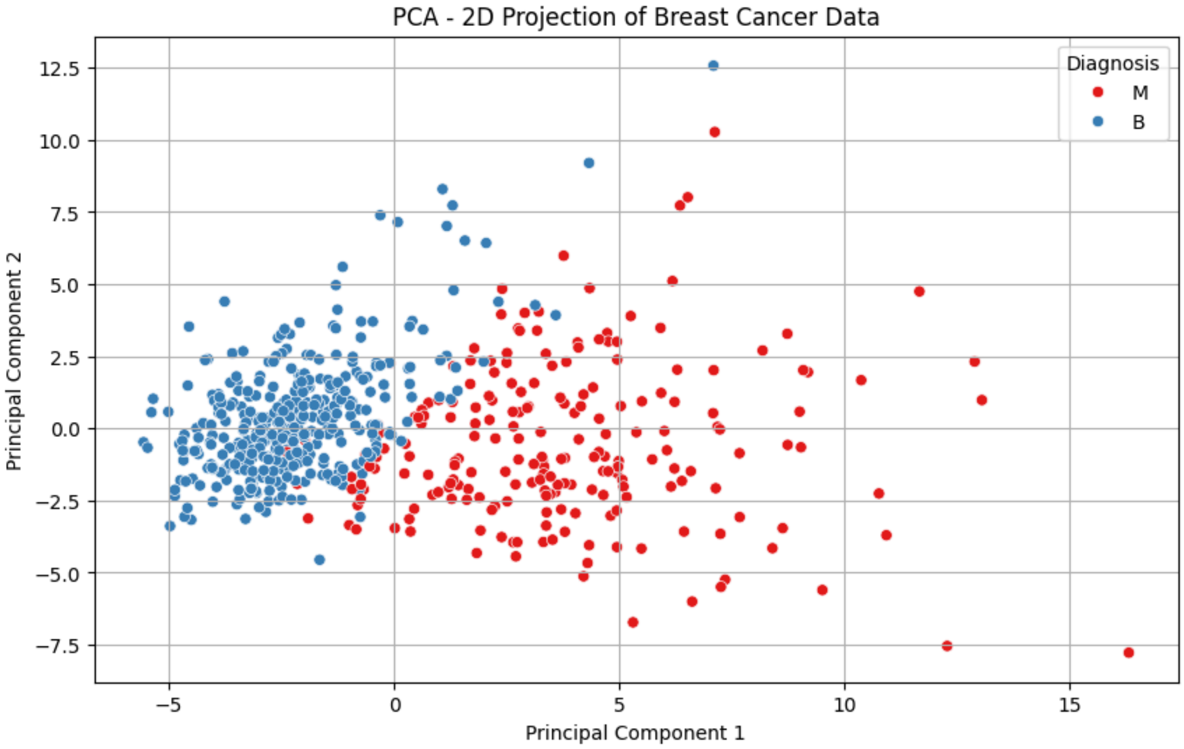


Figure 2: Correlation Matrix of Features

Visualizes which features are most related to each other.

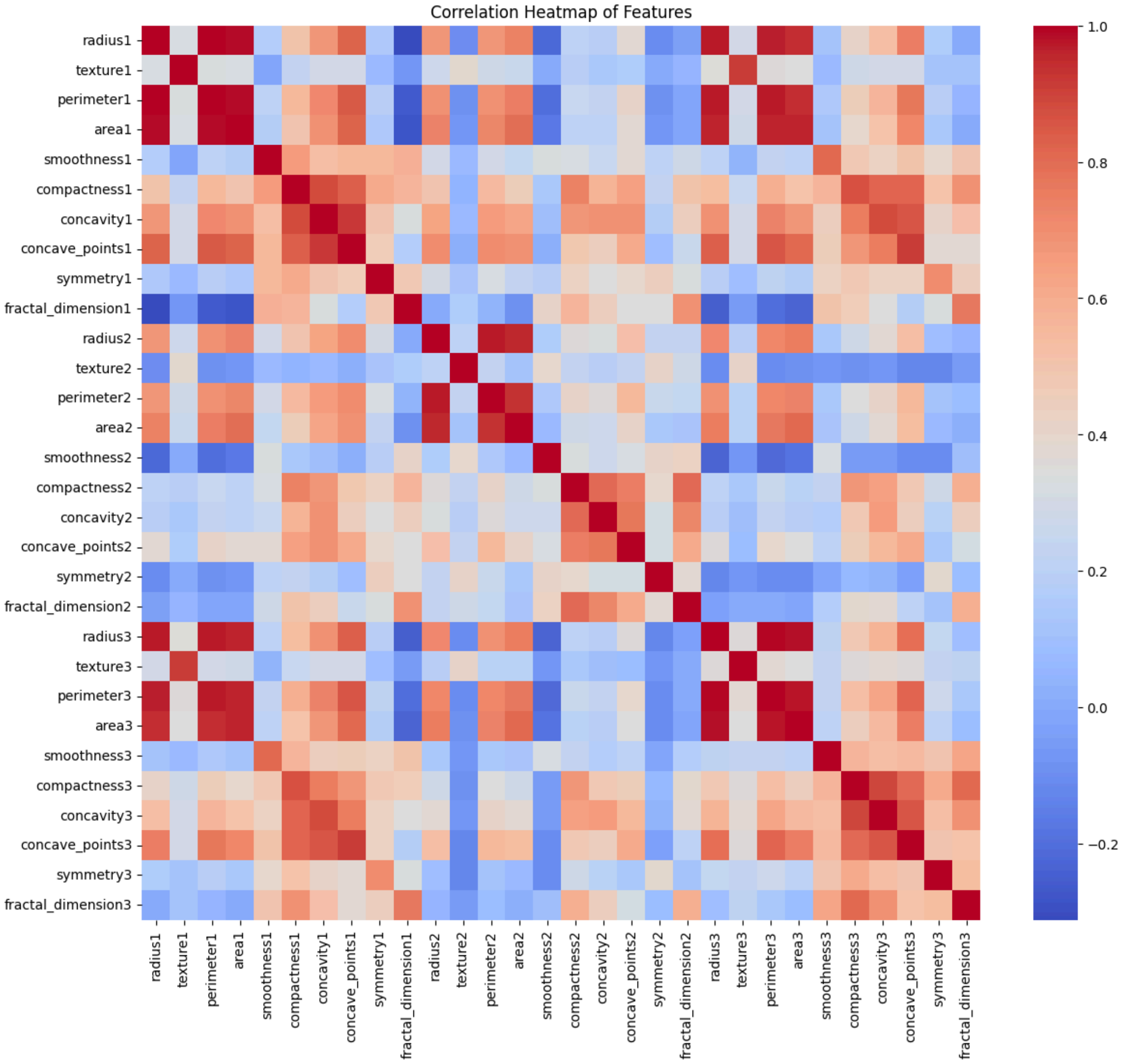


Figure 3: Distribution of Benign vs. Malignant Diagnoses

Shows class imbalance in the dataset important context for training.

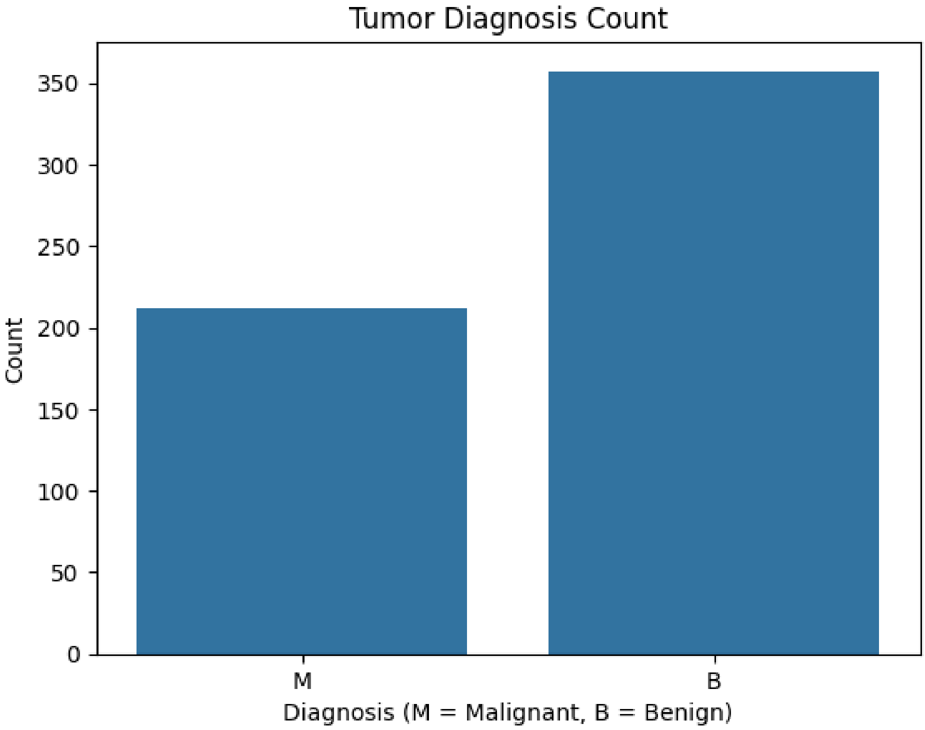


Figure 4: Boxplot Highlighting Outliers in Radius and Area

Supports outlier analysis.

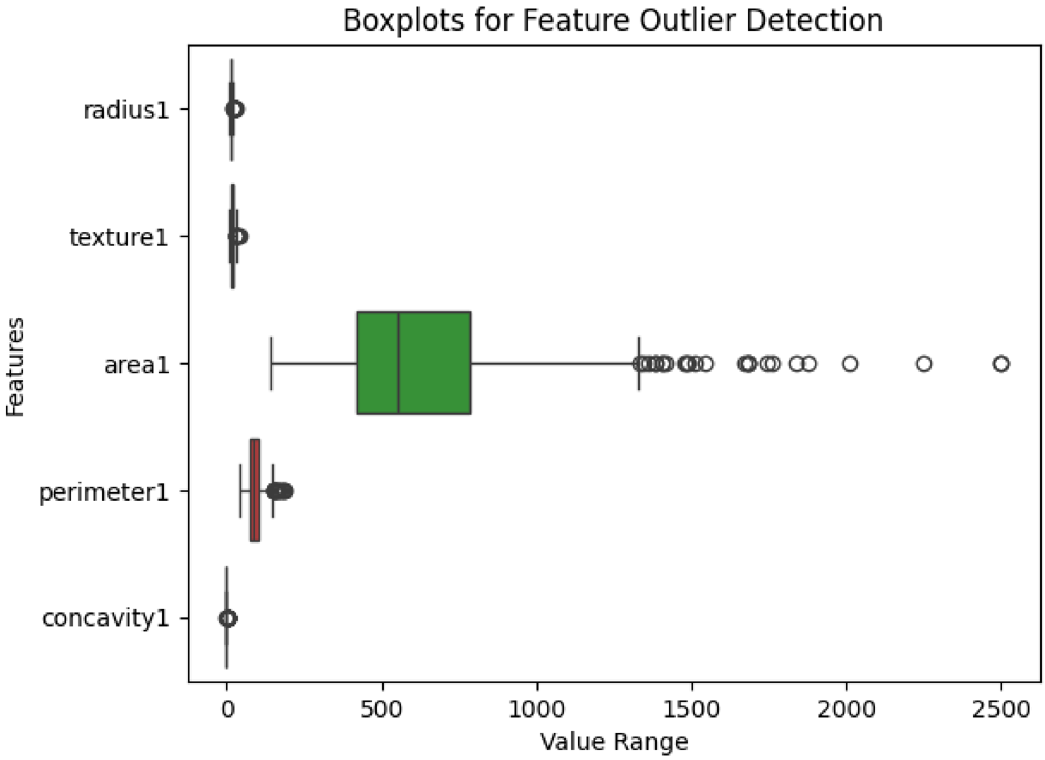
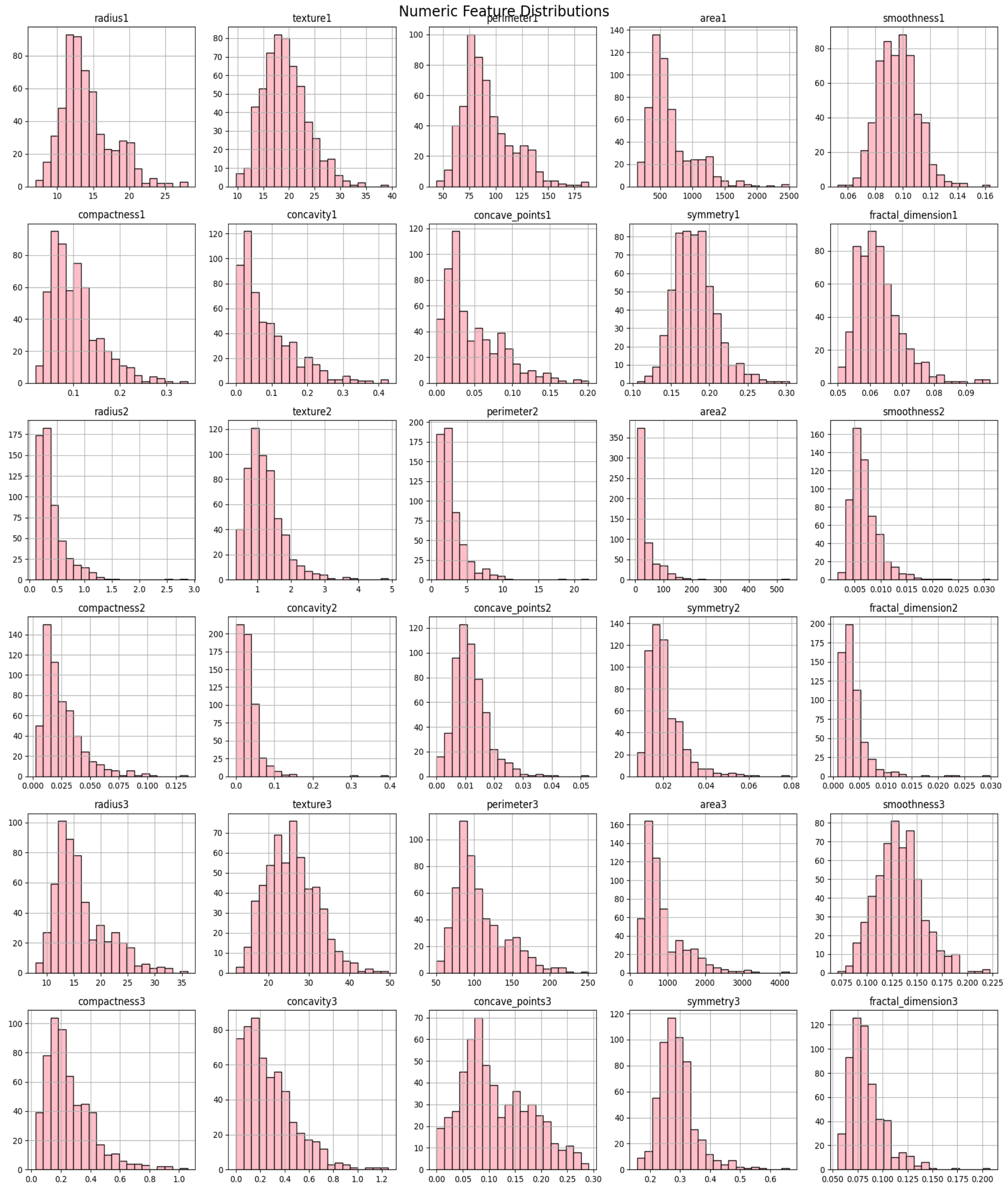


Figure 5: Histograms of Numeric Features

This figure displays the distribution of all 30 numeric features from the Breast Cancer Wisconsin (Diagnostic) dataset. Many features, such as `area1`*,* `radius1`*, and* `concavity3`*, exhibit right-skewed distributions, indicating the presence of extreme values. These patterns justify the need for feature scaling prior to model training.*



Numeric Feature Distributions

References

1. UCI Machine Learning Repository – Breast Cancer Wisconsin (Diagnostic) Data Set. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
2. scikit-learn: Machine Learning in Python. https://scikit-learn.org/
3. seaborn: Statistical Data Visualization. https://seaborn.pydata.org/
4. matplotlib: Python Plotting Library. https://matplotlib.org/
5. ucimlrepo Python Package. https://pypi.org/project/ucimlrepo/