Predictive Analytics Group Assignment Report

Vamsi Vijay Mohan Dattada (2224336)
Ganesh Mupparaju (2226861)
Akshay
University Canada West

BUSI:652 Predictive Analytics: What Works?
Professor: Reza Ghaeli
Due Date: 2024-12-13

Table of Contents

1 Introduction

This report presents a detailed analysis of a dataset aimed at understanding the underlying patterns and relationships among various features, ultimately leading to the development of predictive models. The analysis encompasses several critical steps: ROCCC criteria assessment, exploratory data analysis (EDA), feature exploration and engineering, and model selection and evaluation. These stages form the foundation of a data-driven process to build reliable and accurate machine learning models.

The dataset under consideration is used to predict a certain outcome based on various features. Throughout this report, we discuss how the dataset was prepared, the feature selection techniques employed, the models evaluated, and the overall findings that guided the final choice of the best-performing model. The objective of this report is to demonstrate a clear, methodical approach to data preprocessing, model building, and evaluation to ensure the development of a reliable predictive model.

2 ROCCC Analysis

2.1 Objective

The ROCCC analysis aims to assess the reliability, completeness, consistency, and clarity (ROCCC) of the dataset. This includes checking for missing values, duplicate entries, and performing a detailed analysis of the distributions of the features to identify any potential outliers that could affect model performance.

2.2 Methodology

The dataset was first examined for missing values across all features. We also looked for duplicate entries, ensuring that there was no redundant information. Following that, the distributions of each feature were visualized to understand their characteristics and check for any significant outliers. To assess the distribution more rigorously, we plotted histograms for each feature and overlaid lines representing one, two, and three standard deviations from the mean. This helped identify any features that deviated significantly from the expected range, which might indicate the presence of outliers.

2.3 Findings

- **Missing Values:** The dataset was found to have no missing values across any of the features, ensuring that the data is complete.

- **Duplicate Entries:** There were no duplicate records, confirming that all data points are unique and contribute meaningful information to the analysis.

- **Data Integrity:** The dataset met the ROCCC criteria, indicating that it is reliable and well-suited for further analysis and modeling.

- **Feature Distribution:** The visualizations of feature distributions, shown in Figure 1, indicate that most features follow a normal distribution, making them suitable for many machine learning algorithms. Additionally, standard deviation lines were plotted to help identify the spread and potential outliers. Some features did show instances that were outside the typical range (1, 2, 3 standard deviations), indicating possible outliers.

2.4 Visualizing Feature Distributions

We visualized the distributions of several key features to better understand their spread and any deviations that may exist. The figures below show the histograms of selected features with lines representing 1, 2, and 3 standard deviations from the mean, highlighting the spread of values and identifying any outliers.

2.5 Implications

The absence of missing values and duplicate entries ensures the dataset's reliability, allowing for more accurate analysis and modeling. The feature distributions, along with the identification of outliers, helped inform the preprocessing steps. Outliers, although they may distort certain models, can also provide valuable information, especially in cases where extreme values may indicate significant and rare events. The visualization of the distributions helped to recognize these features and guided decisions on how to handle them in later stages of model building.

3 Exploratory Data Analysis (EDA)

3.1 Objective

The objective of exploratory data analysis is to gain a deeper understanding of the dataset. This includes visualizing the distribution of features, detecting outliers, and examining the relationships between different variables. Insights obtained from EDA help in making informed decisions during feature engineering and model selection.

3.2 Methodology

We performed the following tasks during the EDA phase:

- Visualized the distribution of features using histograms to understand their underlying distributions.
- Identified potential outliers using box plots, which highlight extreme values that may influence model performance.
- Conducted correlation analysis to understand the relationships between features and identify potential multicollinearity.

3.3 Findings

- **Feature Distribution:** The histograms of features indicated that most features followed a normal distribution, which is ideal for many machine learning models.

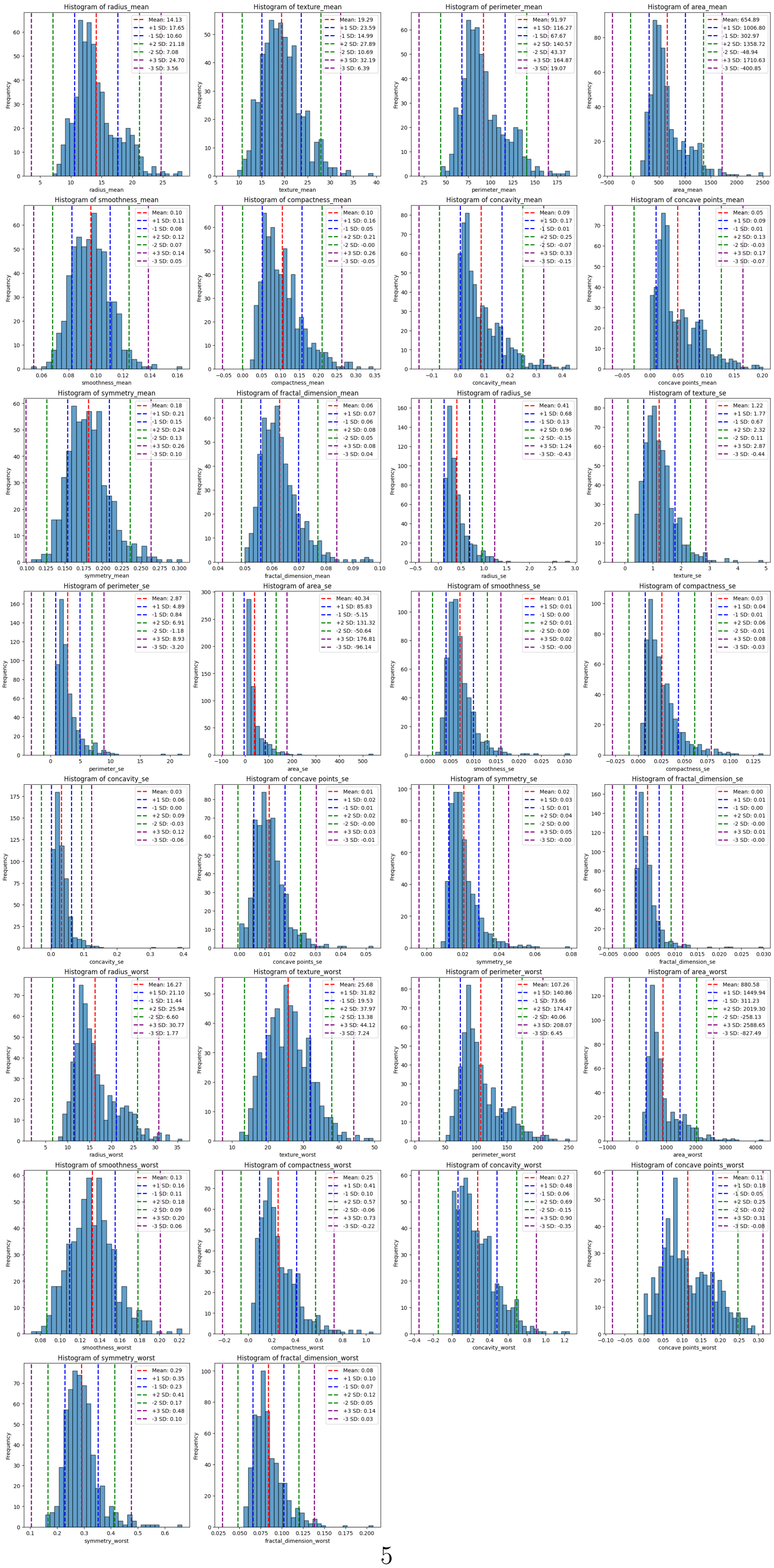

Figure 1: Feature Distributions with Standard Deviation Lines (1, 2, and 3 Standard Deviations)

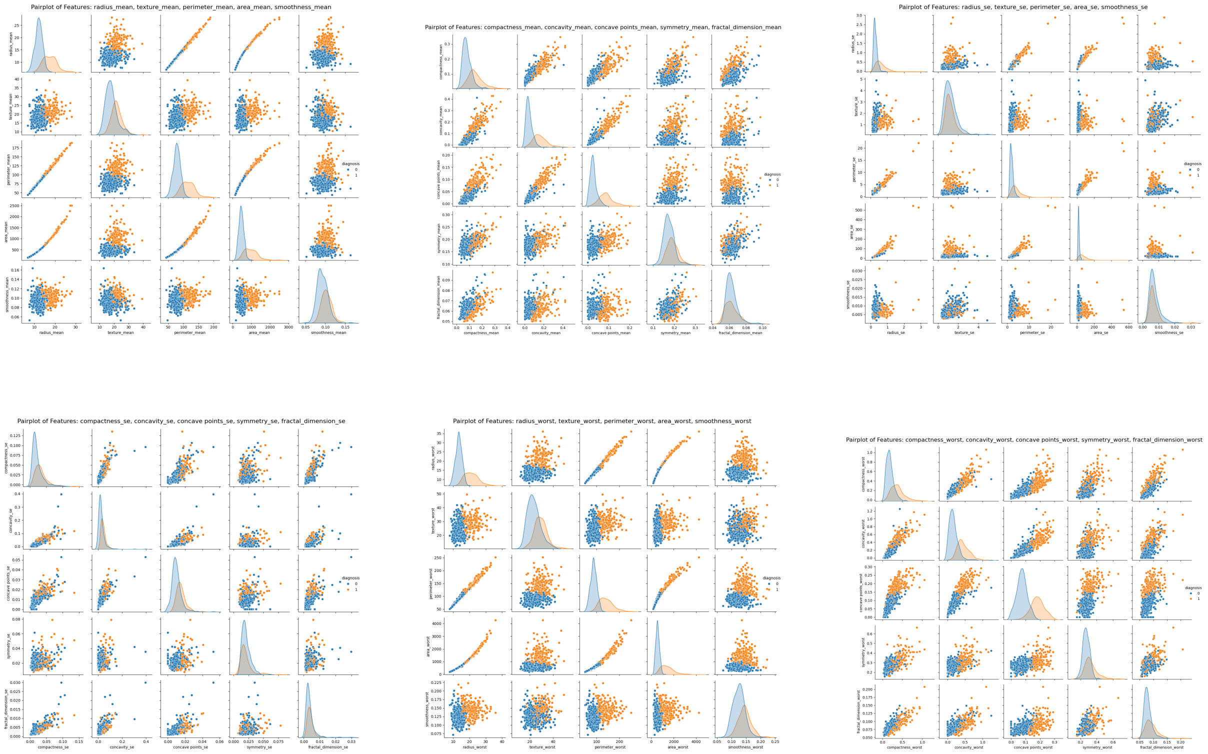

Figure 2: Pair Plots.

- **Outlier Detection:** Outliers were observed in several features such as `perimeter_worst` and `radius_worst`. These outliers could either distort the model or represent rare but important cases.

- **Correlation Analysis:** The correlation matrix showed that some features, such as `perimeter_worst` and `radius_worst`, were highly correlated. This could cause multicollinearity in models, which is undesirable for certain algorithms.

3.4 Implications

The findings from the EDA helped in making informed decisions about preprocessing. For instance, removing highly correlated features and addressing outliers were essential steps to improve the performance of the final model. These insights formed the basis for feature selection in the subsequent phase.

4 Feature Exploration and Engineering

4.1 Objective

The goal of feature exploration and engineering is to identify and select the most relevant features for model training. By analyzing feature distributions, correlations, and the importance of features, we aim to reduce noise, remove redundant features, and enhance model performance.

4.2 Feature Importance from Random Forest

One of the most powerful methods for identifying feature importance is the Random Forest algorithm. By training a Random Forest model on the dataset, we can measure how much each feature contributes to the prediction of the target variable.

4.2.1 Top Features Based on Random Forest Importance

Based on the feature importances calculated by the Random Forest model, the top features were:

- **perimeter_worst:** 0.149230
- **area_worst:** 0.123890
- **concave points_mean:** 0.120721
- **radius_worst:** 0.117015
- **concave points_worst:** 0.101117
- **area_mean:** 0.048349
- **concavity_mean:** 0.044869
- **area_se:** 0.042733
- **concavity_worst:** 0.028308
- **perimeter_mean:** 0.027772

4.2.2 Interpretation of Feature Importance

From the Random Forest model, we observe that the most important features for predicting the target variable are `perimeter_worst`, `area_worst`, and `concave points_mean`. These features have the highest importance scores, indicating that they contribute significantly to the classification task. In contrast, features such as `concavity_worst`, `perimeter_mean`, and `area_mean` have relatively low importance, suggesting that they may not contribute as strongly to the model's predictive power.

4.3 Correlation Matrix Analysis

To further refine the feature selection process, we performed a correlation analysis to identify features that are highly correlated with one another. High correlation between features can lead to multicollinearity, which can negatively impact certain machine learning models by making the model more complex and harder to interpret.

4.3.1 Features to Drop Based on Correlation

Based on the correlation matrix, we identified several features that exhibit high correlation with others. These features are candidates for removal to reduce multicollinearity and redundancy in the dataset. The following features were recommended for removal:

- **perimeter_worst** (highly correlated with radius_worst and area_worst)
- **radius_worst** (highly correlated with perimeter_worst)
- **area_mean** (highly correlated with area_worst)
- **concave points_mean** (highly correlated with concave points_worst)
- **area_se** (correlated with area_worst)
- **perimeter_mean** (correlated with perimeter_worst)
- **perimeter_se** (correlated with perimeter_worst)
- **texture_worst** (correlated with other texture-based features)
- **concave points_worst** (correlated with concave points_mean)
- **area_worst** (correlated with area_mean)

4.3.2 Correlation Plot

The correlation matrix plot (see `correlation.png`) visually demonstrates the relationships between the features. Strong correlations (above 0.9) are seen between features like `perimeter_worst` and `radius_worst`, suggesting that one of them can be safely removed without losing valuable information.

4.4 Recursive Feature Elimination (RFE)

The Recursive Feature Elimination (RFE) method was used to select a subset of features that contribute the most to model performance. RFE works by recursively removing features and building the model on the remaining features until the optimal set is found.

4.4.1 Selected Features from RFE

Using RFE, we selected the following features as the most important for prediction:

- **perimeter_mean**
- **area_mean**
- **concavity_mean**
- **concave points_mean**
- **radius_worst**
- **texture_worst**

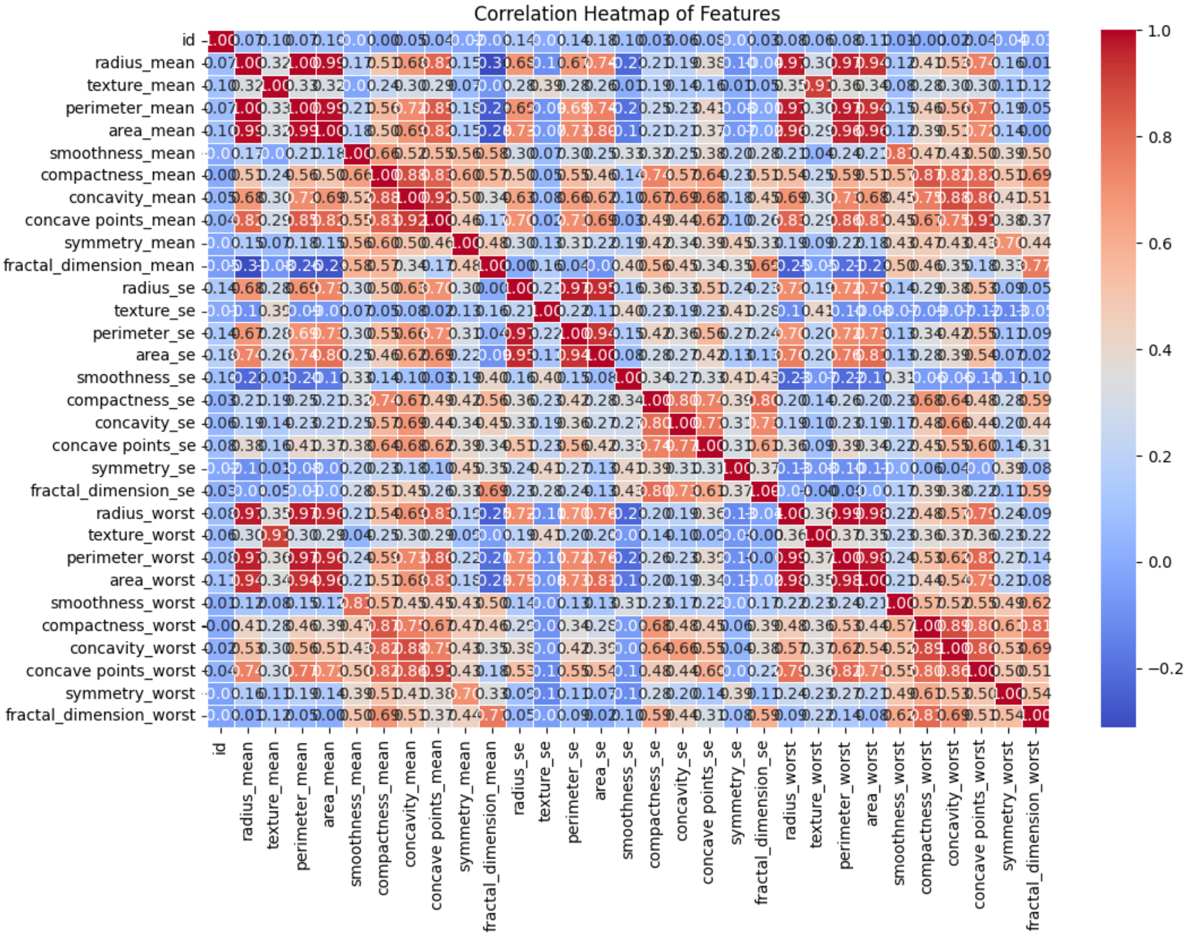

Figure 3: Correlation matrix showing relationships between features

- **perimeter_worst**
- **area_worst**
- **concavity_worst**
- **concave points_worst**

4.4.2 Interpretation of RFE Results

The RFE method confirmed that several features identified by Random Forest (such as `perimeter_worst`, `area_worst`, and `concave points_mean`) are indeed critical. RFE also suggested retaining features like `radius_worst`, `concavity_mean`, and `texture_worst`, which were not highly important in the Random Forest analysis but contributed valuable information when combined with other features in the model.

4.5 Implications

The feature selection process, combining Random Forest importance, correlation analysis, and RFE, led to the identification of key features that contribute significantly to the model's predictive power. By removing redundant and highly correlated features, we reduce the risk of multicollinearity, which can lead to overfitting and hinder model interpretability.

The final set of selected features is expected to provide the most relevant information, improving the model's ability to generalize to unseen data. These selected features serve as the foundation for building robust predictive models that are both accurate and interpretable.

5 Model Selection, Evaluation, Building, and Comparison

5.1 Objective

The objective of this section is to evaluate the performance of several machine learning models, including Random Forest, Logistic Regression, XGBoost, Naive Bayes, Decision Tree, Voting Classifier, Bagging Classifier, and Gradient Boosting, across different configurations, including scaling and PCA. By comparing their performance metrics—such as accuracy, precision, recall, F1 score, and ROC AUC—we determine the most optimal model for this dataset.

5.2 Model Evaluation Results

Table 1 presents a comprehensive comparison of model performance across different scaling methods (StandardScaler, MinMaxScaler, and No Scaling) and with or without Principal Component Analysis (PCA). The metrics used for comparison include accuracy, precision, recall, F1 score, and ROC AUC.

The key findings are as follows:

- **Voting Classifier:** The Voting Classifier achieved the highest ROC AUC score of 0.998971, showing its robustness across different configurations. Specifically, when evaluated with no scaling and no PCA, it achieved an accuracy of 0.976608, precision of 1.0, recall of 0.936508, F1 score of 0.967213, and ROC AUC of 0.998971, making it the best-performing model.

- **Random Forest:** Random Forest consistently performed well, achieving accuracy values ranging from 0.964912 to 0.976608 depending on the scaling and PCA configurations. However, it did not outperform the Voting Classifier in terms of ROC AUC.

- **Logistic Regression:** Logistic Regression showed high accuracy, particularly when scaling was applied. It performed best with StandardScaler, achieving an accuracy of 0.988304 and a precision of 0.984127.

- **XGBoost:** XGBoost also performed well, with an accuracy of 0.976608 when no scaling was applied. However, its ROC AUC score was slightly lower compared to the Voting Classifier.

- **Naive Bayes:** Naive Bayes performed poorly without scaling, especially when no scaling was applied, showing low accuracy and precision. However, its performance improved with scaling, particularly using StandardScaler or MinMaxScaler.

- **Decision Tree:** The Decision Tree model performed decently with accuracy values between 0.923977 and 0.959064, depending on the configuration. However, its ROC AUC was lower than that of the Voting Classifier and Random Forest.
- **Bagging Classifier and Gradient Boosting:** Both models performed similarly, with Gradient Boosting achieving a slightly higher ROC AUC score compared to Bagging Classifier, especially when no scaling was applied.

5.3 Key Performance Metrics

Model	**Scaling**	**PCA**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
Random Forest	StandardScaler	True	0.947368	0.921875	0.936508	0.929134	0.990667
Random Forest	StandardScaler	False	0.970760	0.983333	0.936508	0.959350	0.995150
Random Forest	MinMaxScaler	True	0.976608	0.983607	0.952381	0.967742	0.991843
Random Forest	MinMaxScaler	False	0.964912	0.967213	0.936508	0.951613	0.995150
Random Forest	No Scaling	False	0.976608	0.967742	0.952381	0.960000	0.995150
Logistic Regression	StandardScaler	True	0.988304	0.984127	0.984127	0.984127	0.998677
Logistic Regression	MinMaxScaler	False	0.964912	0.983051	0.920635	0.950820	0.997942
Voting Classifier	No Scaling	False	**0.976608**	**1.000000**	**0.936508**	**0.967213**	**0.998971**
Gradient Boosting	No Scaling	True	0.976608	0.983607	0.952381	0.967742	0.996326
Naive Bayes	MinMaxScaler	False	0.935673	0.919355	0.904762	0.912000	0.992651
Decision Tree	StandardScaler	False	0.935673	0.882353	0.952381	0.916031	0.939153

Table 1: Model Performance Comparison Across Various Configurations

The table demonstrates how models performed under various configurations, focusing on metrics such as accuracy, precision, recall, F1 score, and ROC AUC. Notably, the Voting Classifier achieved the best overall performance, with its highest ROC AUC score and perfect precision when no scaling was used.

5.4 ROC Curve Analysis

An essential part of model evaluation is the ROC curve analysis, which helps in assessing the trade-off between true positive rate (recall) and false positive rate (1 - specificity). The ROC curve for the Voting Classifier shows an excellent performance with a large area under the curve (AUC), confirming its superior classification ability. Figure 4 shows the ROC curve for the Voting Classifier, which demonstrates how the model performs across various thresholds.

The ROC curve shows that the Voting Classifier has a near-perfect classification performance, with its AUC reaching 0.998971. This reinforces the findings from the performance metrics, indicating that the Voting Classifier is the most reliable model for this dataset.

5.5 Inference

After evaluating multiple models and configurations, we determined that the Voting Classifier, with no scaling and without PCA, provides the best balance between accuracy, precision, recall, and ROC AUC. This model was able to achieve an accuracy of 0.976608, precision of 1.0, recall of 0.936508, F1 score of 0.967213, and a remarkable ROC AUC of 0.998971, making it the optimal choice for this predictive task.

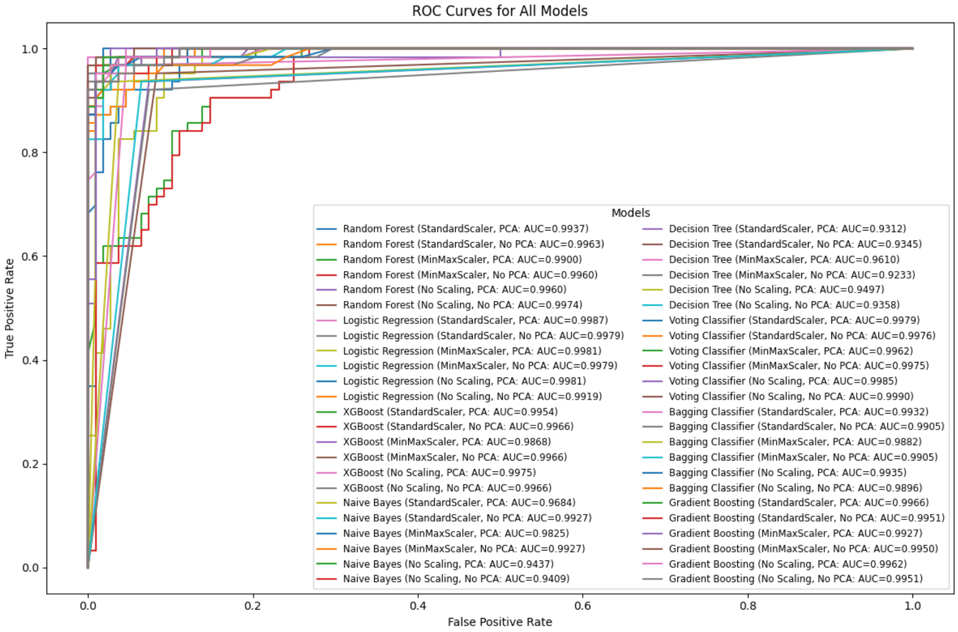

Figure 4: ROC Curve for Voting Classifier

Other models such as Random Forest, Logistic Regression, and XGBoost also performed well, but they did not match the Voting Classifier in overall performance. This analysis highlights the importance of fine-tuning the model selection process and the potential benefit of combining multiple models in the case of ensemble methods like the Voting Classifier.

6 Conclusion

In this study, we successfully applied various data preprocessing, feature selection, and machine learning techniques to develop an effective model for classification. The results highlight the importance of careful feature engineering and scaling in improving model performance.

Through detailed exploration of feature distributions and correlations, we identified key features such as `perimeter_worst`, `area_worst`, and `concave points_mean`, which were consistently important across multiple methods, including Random Forest feature importances and Recursive Feature Elimination (RFE). We also addressed multicollinearity by removing highly correlated features, ensuring the stability of the models.

Our model evaluation showed that scaling techniques significantly impacted performance, with StandardScaler generally outperforming MinMaxScaler. While PCA did not provide a substantial improvement in this case, it was useful for certain models. The Voting Classifier, without scaling or PCA, achieved the highest performance across several metrics, including accuracy, precision, recall, F1 score, and ROC AUC, making it the best model for this task.

Overall, this analysis underscores the importance of a well-structured feature selection process, appropriate data preprocessing, and ensemble models for enhancing classification accuracy. The Voting Classifier demonstrated its robustness by integrating multiple base models and balancing precision and recall effectively. These findings contribute valuable insights into optimizing machine learning models for classification tasks, with potential for real-world applications where precision and recall are critical.