Analysis of Cancer Data

Group_37

Introduction

Modern mass spectrometry collects thousands of molecular features, but analyzing such high-dimensional data is challenging. Identifying key patterns can aid early diagnosis and improve treatment.

To explore the question, the Arcene dataset will be divided into a training set and a test set. A unique training set containing 100 samples with 5,000 randomly selected features. A fixed test set (100 samples with all 10,000 features) will be used for model evaluation. The study will employ various classification techniques to assess their accuracy in distinguishing between cancerous and normal tissue samples. Through the model analysis, to explore the following research questions:

- **Primary research question**: Can biochemical features accurately distinguish between cancerous and normal tissue samples?
- **Secondary research question**: Compare the results of different classification models to find out a best classification model.

Data Processing

Initial rejection of potential probes

Based on the dataset description, we performed an initial feature selection process to reduce the dimensionality of the data and eliminate variables that may not provide meaningful insights for the model. We focused on removing features with extremely low variance, as these

are typically considered to be probes or noise sequences that do not contribute valuable information for predictive modeling. Such features often show minimal variability across observations and do not offer distinguishing power for classification or regression tasks. By removing these low-variance features, we reduced the data dimensionality, which facilitates more efficient analysis and model training, and helps to prevent overfitting by removing redundant or irrelevant information. This step is essential for improving the overall performance and interpretability of the model.

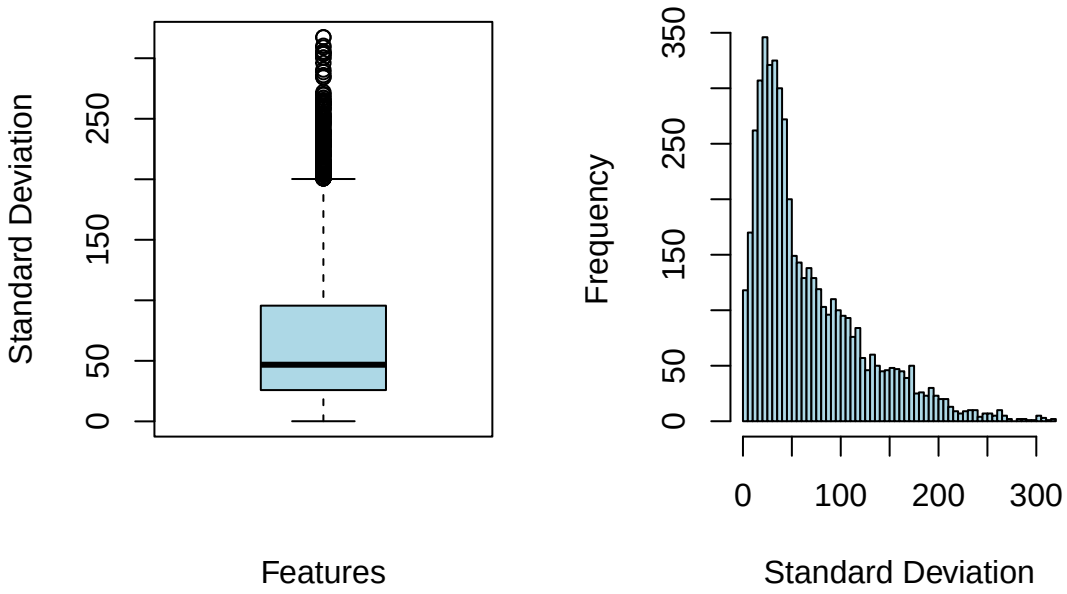

Figure 1: Variable Standard Deviation Distribution

Data Dimension Reduction Processing

PCA was further applied to reduce the dimensionality of the data. The Kaiser criterion and the cumulative variance contribution ratio were used to select the effective principal components. Based on these criteria, the original dataset was updated, leading to a significant reduction in its dimensionality.

Formal data analysis

Tree-based methods

Classification Tree

Classification Tree partition the feature space into a number of disjoint and non-overlapping regions. And predict the class of a given observation as the most commonly occurring class of training observations is the region to which it belongs.

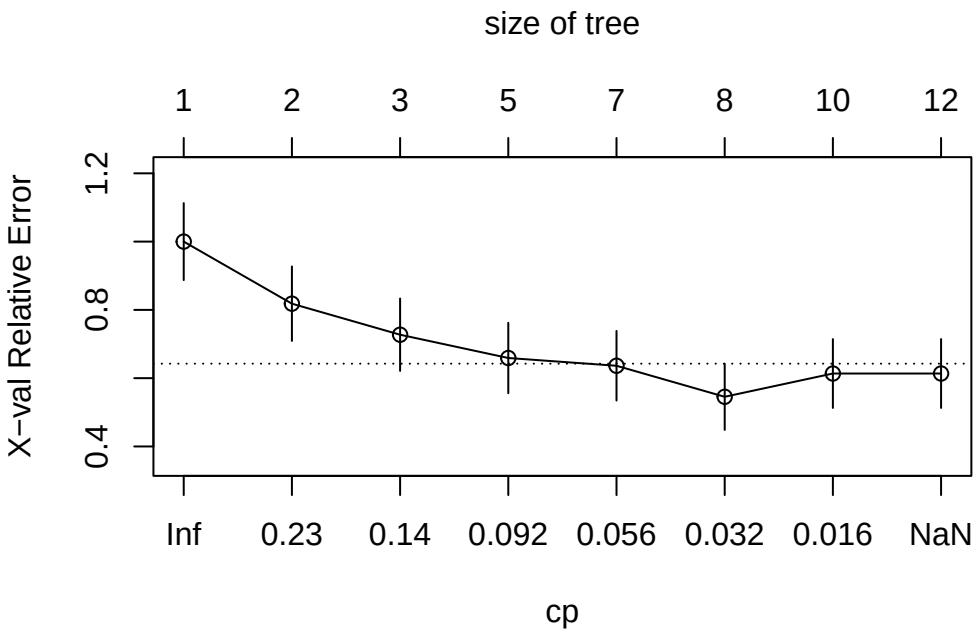

Figure 2: Xerror vs CP

The pruning process is based on the complexity parameter (cp) selection. The optimal cp is chosen as the largest value within the range where the cross-validation error remains within one standard deviation of the minimum error. Based on this criterion, a cp of 0.032 is selected to prune the new tree.

The classification tree also highlights the importance of variables, showing that PC1 is the most significant, followed by PC12 and PC2. Variables like PC18, PC13, PC21, and PC24 are considered less important.

The test accuracy is 0.68, indicating poor performance. While the classification tree has some predictive ability, its overall effectiveness is limited.

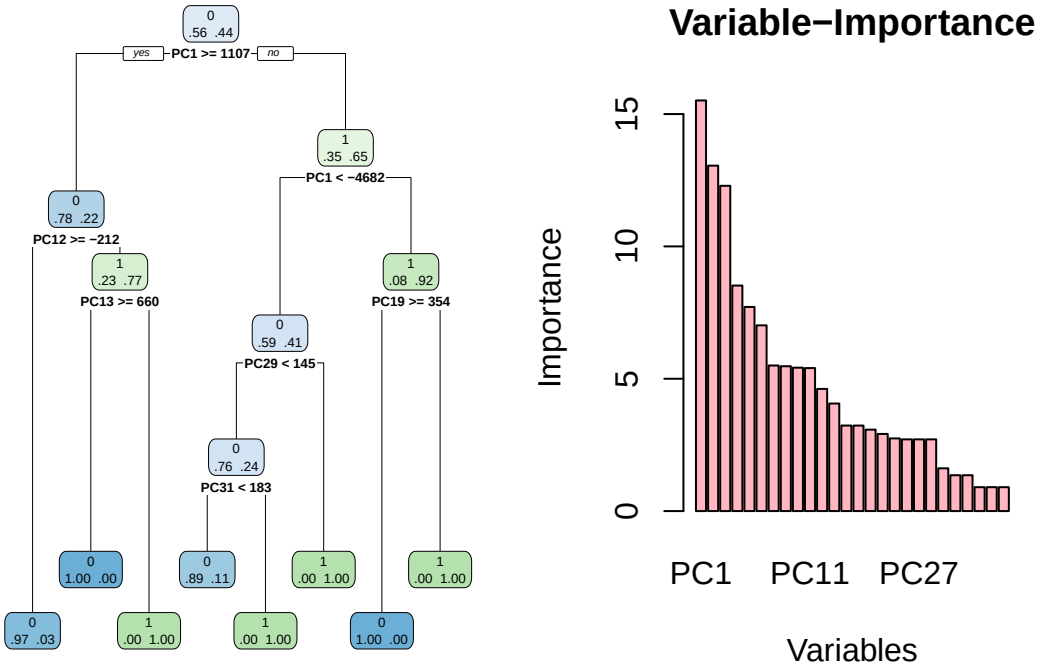

Figure 3: Pruned Classification Tree (cp=0.032)

Bagging Tree

The core idea of bagging is to repeatedly draw samples from the original dataset and build a classification tree on each bootstrapped sample. For each test observation, the class predicted by each tree is recorded. The final prediction is determined by a majority vote, where the class that appears most frequently across all predictions becomes the overall predicted class. To select the minimum number of trees that stabilize the OOB error, the model was trained with different numbers of trees, and the OOB error was monitored. Once the error stabilized, the smallest number of trees that achieved this stabilization was chosen to build the final Bagging Tree model.

According to the Bagging Tree, Variable.2640 is the most important factor, followed by Variable.4192 and Variable.6481, Variable.132 and Variable.1193, Variable.6442 and Variable.1235 are relatively unimportant.The Accuracy of test is 0.79, which is good. Showing the Bagging Tree has good classification ability.

[1] “Accuracy: 0.79” [1] “AUC: 0.895292207792208”

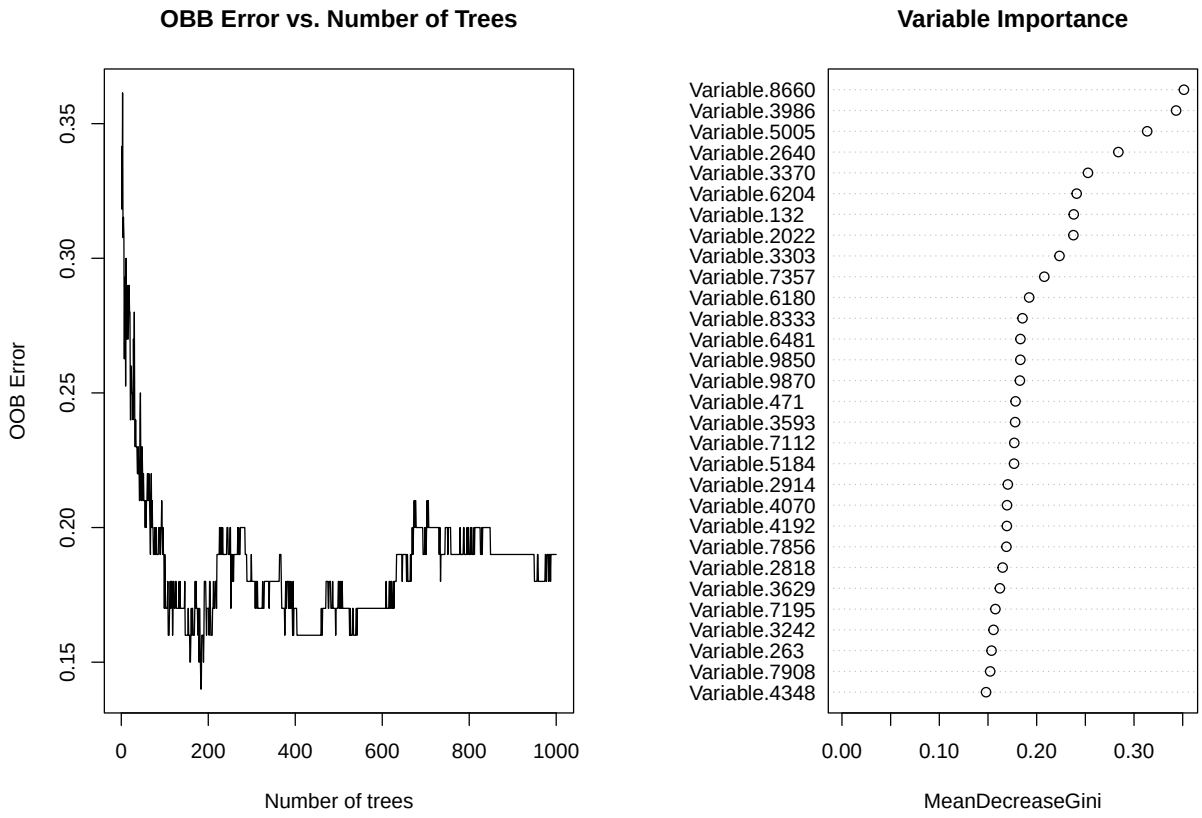

Figure 4: Model Performance

Random Forests

Random forests enhance Bagging Tree by reducing the correlation between individual trees. By randomly excluding a subset of variables at each split, the trees become more diverse, leading to more stable and reliable predictions.

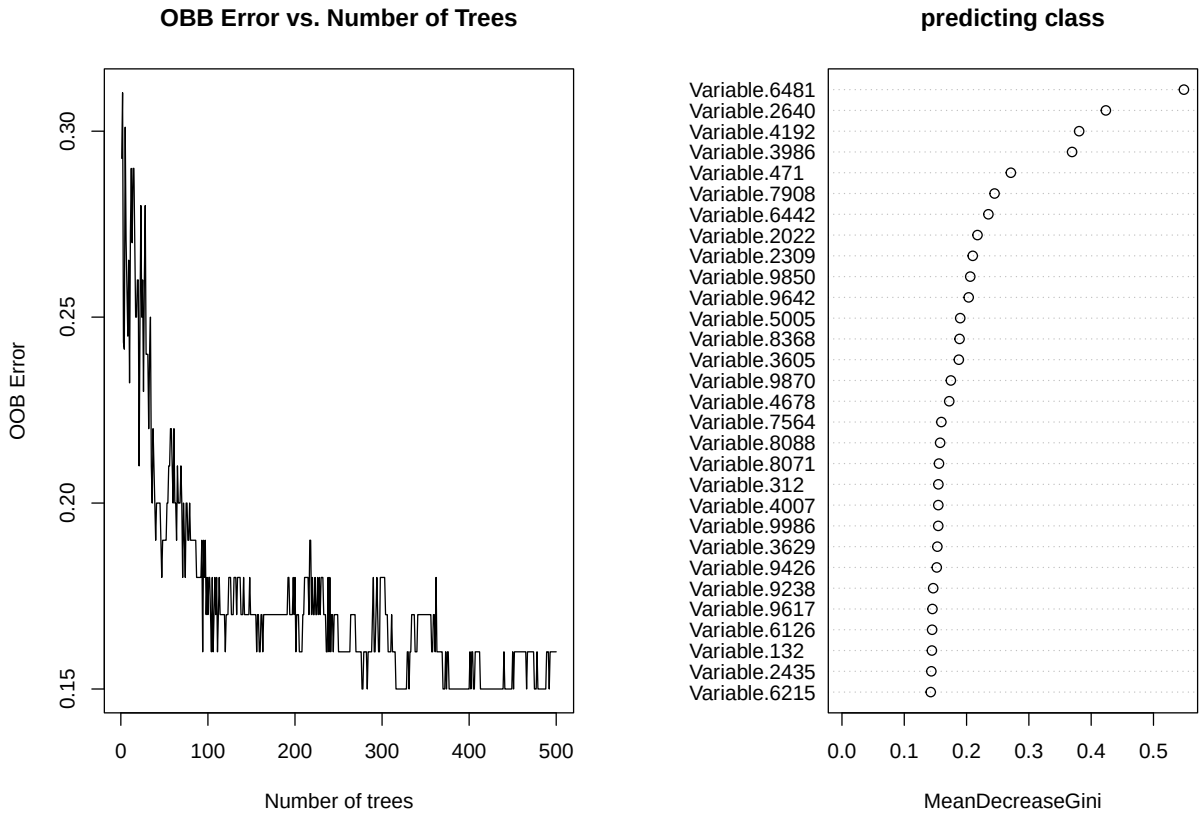

Figure 5: Model Performance

According to the random forests, Variable.2640 and Variable.4192, which were also important in the Bagging model, remain influential.

[1] "Accuracy: 0.81" [1] "AUC: 0.91112012987013"

The ROC comparison shows that the AUC values for the classification tree, bagging tree, and random forest are 0.6936, 0.8953, and 0.9111, respectively. Since the random forest achieves the highest AUC, it demonstrates the best predictive performance among the three models.

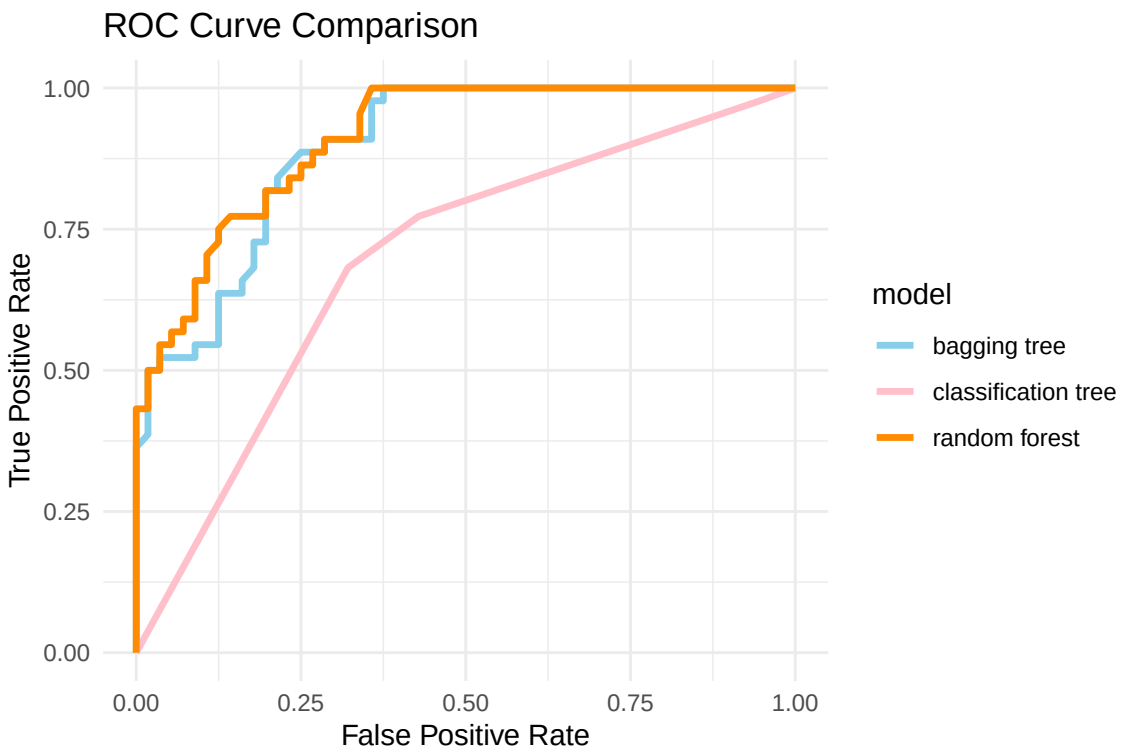

Figure 6: ROC Plot Compare

Discriminant Analysis

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are classification methods based on probability distributions. LDA assumes that different classes share the same covariance structure, meaning they have similar spread and orientation, which leads to a linear decision boundary. This makes LDA efficient when the assumption holds, but it may struggle when class distributions are complex. QDA, on the other hand, allows each class to have its own covariance structure, resulting in more flexible, curved decision boundaries. This makes QDA more powerful for non-linearly separable data but also more prone to overfitting when data is limited.

Test of boxm

The Box's M-test for homogeneity of covariance matrices was performed, with the results showing the p-value was extremely small ($< 2.2e\text{-}16$), indicating that the assumption of homogeneity of covariance matrices is violated. However, I still proceeded with LDA as a comparison method.

Next, the Shapiro-Wilk normality test was conducted for each feature, and the Bonferroni correction was applied to adjust for multiple comparisons. A total of 5 features failed the normality test after the correction, meaning they do not follow a normal distribution.

Two classification models, Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), were then applied to the data. For LDA, the accuracy was 73%, with a confusion matrix showing 45 true negatives, 28 true positives, 11 false positives, and 16 false negatives. For QDA, the accuracy improved to 79%, with a confusion matrix showing 48 true negatives, 31 true positives, 8 false positives, and 13 false negatives. Finally, ROC curves were plotted for both LDA and QDA. The AUC (Area Under the Curve) values were printed on the plots, providing a visual representation of the models' performance. The AUC for QDA was superior to that of LDA, indicating that QDA had better discriminatory power in distinguishing between the two classes.

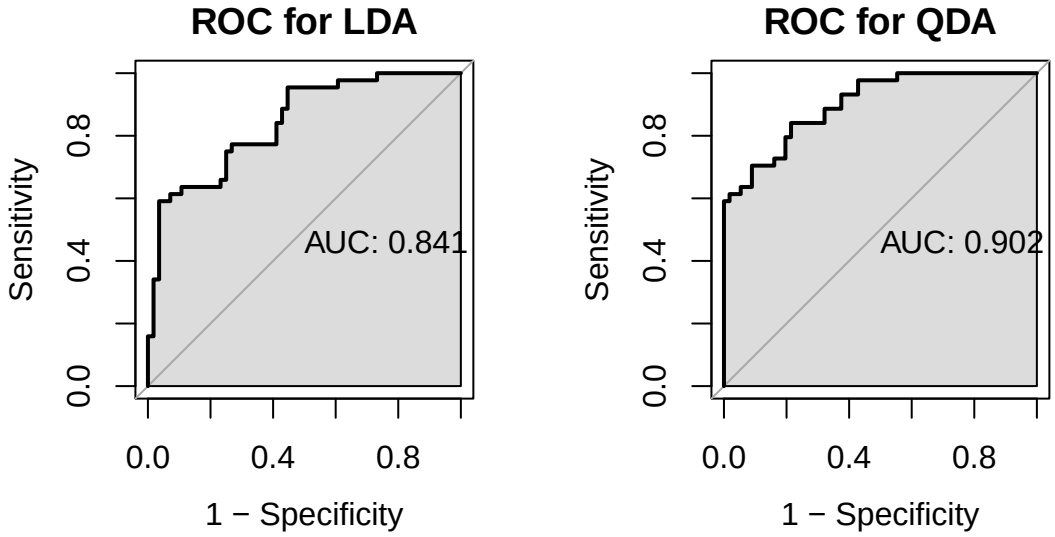

Figure 7: ROC

nn work

Neural networks improve predictive performance by capturing complex nonlinear relationships between features and the target variable. With multiple hidden layers and structured activation functions, they learn intricate patterns in the data. Additionally, their architecture reduces dependence on any single feature, enhancing generalization.

Based on the PCA results, we initially designed a neural network with two hidden layers: the first layer containing 47 neurons and the second layer containing 20 neurons. This structure was chosen to capture the primary features of the dataset while maintaining a balance between model complexity and computational efficiency. The aim was to leverage the dimensionality reduction insights from PCA to guide the architecture design and improve the model's ability to extract meaningful patterns from the data.

[1] "Accuracy: 0.61" [1] "AUC: 0.645900974025974"

The training performance was suboptimal, and the model did not meet expectations. One possible reason for this was the insufficient network capacity, which prevented the model from fully capturing the complex patterns within the data.

Adjustment Strategy

To improve the model's learning ability, we increased the number of neurons in the first hidden layer to 533, enhancing its capability to extract meaningful features. In the subsequent hidden layers, we gradually reduced the number of neurons to $256 \rightarrow 60 \rightarrow 20$, aiming to refine the model's representation while preventing overfitting. This hierarchical structure allows the network to capture high-dimensional features in the initial layers and progressively distill the most relevant information in the deeper layers.

The training performance improved significantly, with the AUC increasing from 0.64 to 0.95. This dramatic improvement suggests that the adjusted network architecture effectively enhanced the model's ability to learn complex patterns, leading to a much better classification performance. The increased network capacity in the initial layers allowed for better feature extraction, while the gradual reduction in neurons helped refine the representations, ultimately resulting in a more robust and well-generalized model.

[1] "Accuracy: 0.89" [1] "AUC: 0.950081168831169"

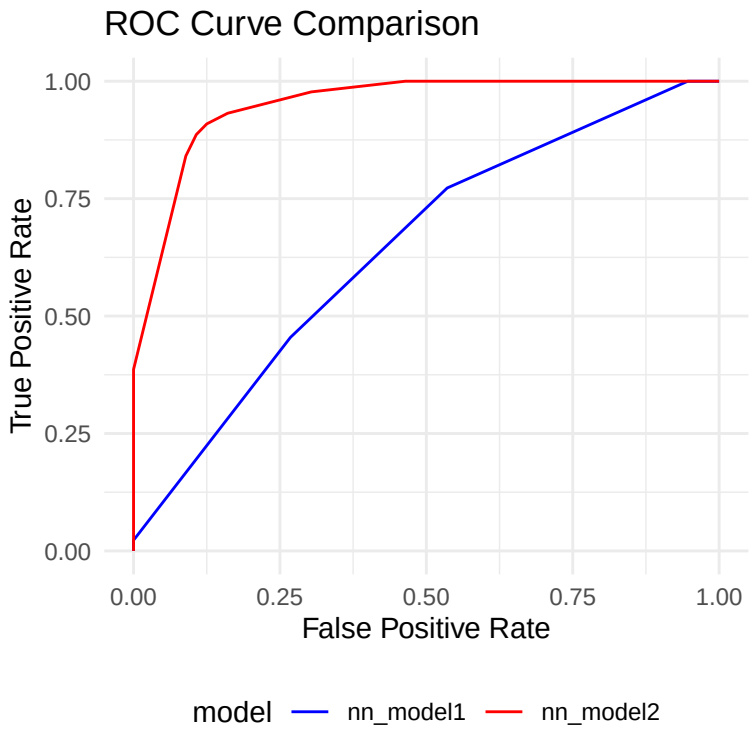

Figure 8: ROC of nn1 and nn2

SVM

Support Vector Machine (SVM) is a classification algorithm that finds the optimal boundary between different classes by maximizing the margin between data points. It identifies key data points, known as support vectors, that are closest to the decision boundary and uses them to define the classification rule. If the data is not linearly separable, SVM applies the kernel trick to transform it into a higher-dimensional space where separation is possible. One of the main reasons for using SVM is that it does not assume specific data distribution or variance requirements, unlike methods such as Discriminant Analysis (DA).

SVM is effective for high-dimensional datasets and works well with small to medium-sized data. However, it can be computationally expensive for very large datasets and requires careful tuning of hyperparameters like C, gamma, and kernel type to achieve optimal performance. The datasets contain features derived from different preprocessing methods:

- Dataset 1: Original dataset with 5000 features.
- Dataset 2: Feature selection reduced the number of features to 1824.

- Dataset 3: Principal Component Analysis (PCA) reduced the feature count to 47.

For each dataset, SVM models were trained, tested, and optimized using GridSearchCV to identify the best hyperparameters. The datasets were loaded and divided into features (X) and target labels (y). Standardization was applied using StandardScaler to ensure the SVM operates effectively. An initial SVM model with default parameters was trained on each dataset. Model performance was evaluated using accuracy and Area Under the Curve (AUC) scores. Hyperparameter tuning was performed using GridSearchCV to optimize SVM parameters (C, gamma, and kernel). The best model from the grid search was selected and evaluated on the test set. The following hyperparameters were explored: C: [0.01, 0.1, 0.5, 1, 2, 10, 100], gamma: ["scale", "auto", 0.01, 0.1, 1], kernel: ["linear", "rbf", "poly"]. The best hyperparameters were selected based on AUC scores.

Dataset	Features	Accuracy	AUC
Original	5000	0.84	0.930
Feature selection	1824	0.88	0.948
PCA Reduction	47	0.81	0.924

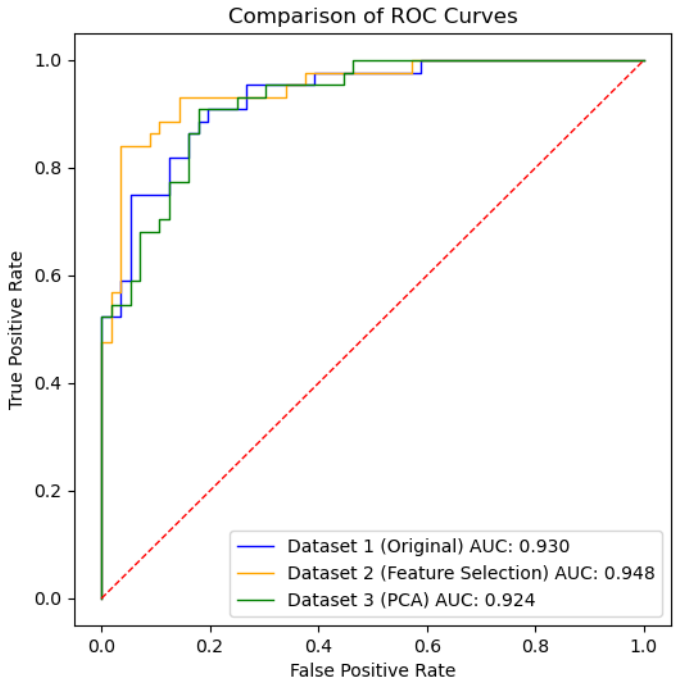

Figure 9: ROC of SVM

Reducing the number of features from 5000 to 1824 improved accuracy and AUC after tuning. PCA-reduced data (47 features) maintained good accuracy but performed slightly worse than feature selection. In most cases, GridSearchCV improved AUC over the default model. For Dataset 1, a linear kernel was chosen as optimal, while for Datasets 2 and 3, an RBF kernel was preferred.

Feature selection (1800 features) provided the best performance in terms of both accuracy and AUC. PCA (40 features) led to minor accuracy reduction but is still a viable option for dimensionality reduction.

Conclusion:

Classification Model	Accuracy	AUC
Classification tree	0.68	0.6936
Bagging tree	0.79	0.8953
Random forests	0.81	0.9111
LQA	0.73	0.908
QDA	0.79	0.902
NN works	0.89	0.95
SVM	0.88	0.948