Analysis of Gen Cancer Data

Group_37

Data sets created

Introduction

Research Question

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.

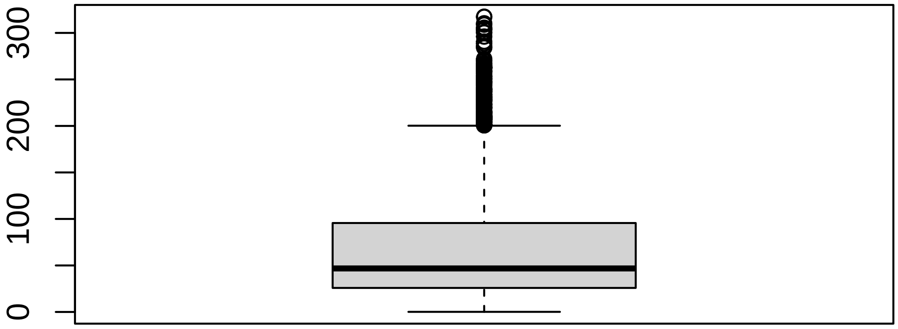


Figure 1: Variable Importance

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.

Tree-based methods

Classification Tree

Description: 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.

Reason: it's easy to understand and handle the categorical features without the need to create a long series of dummy variables.

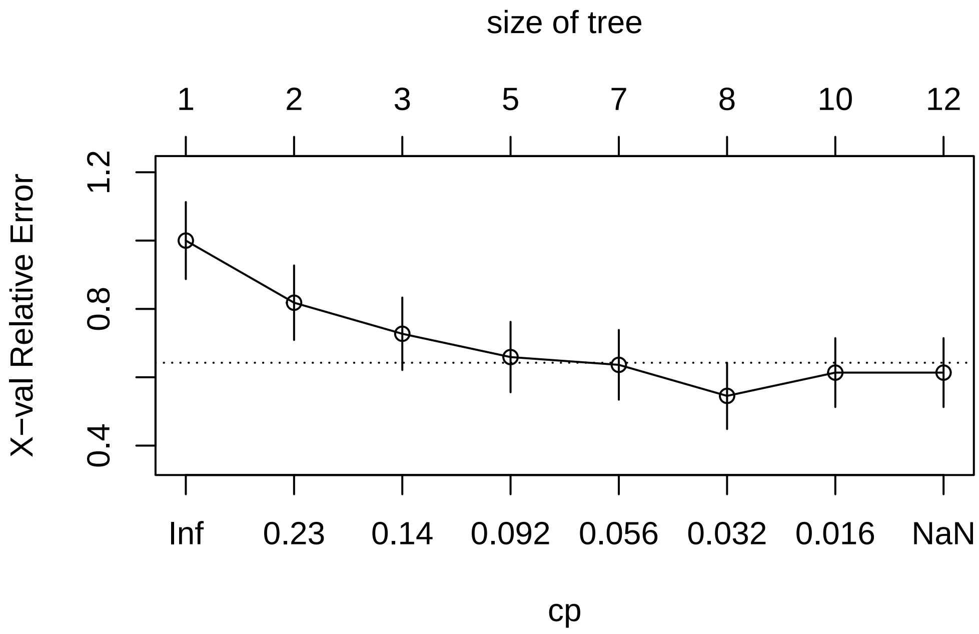


Figure 2: Xerror vs CP

```
Classification tree:
rpart(formula = Class ~ ., data = traindata, method = "class",
    cp = -1, minsplit = 2, minbucket = 1)

Variables actually used in tree construction:
[1] PC1  PC10 PC12 PC13 PC19 PC29 PC31 PC32

Root node error: 44/100 = 0.44

n= 100

         CP nsplit rel error  xerror     xstd
1  0.340909      0  1.000000 1.00000 0.112815
2  0.159091      1  0.659091 0.81818 0.109091
3  0.125000      2  0.500000 0.72727 0.106017
4  0.068182      4  0.250000 0.65909 0.103128
5  0.045455      6  0.113636 0.63636 0.102045
6  0.022727      7  0.068182 0.54545 0.097064
7  0.011364      9  0.022727 0.61364 0.100900
```

```
8 -1.000000     11  0.000000 0.61364 0.100900
```

Start with a tree is fully grown, to see the cross validation results use the printcp() function and use plotcp() function to check the complexity parameter value. The smallest tree strategy refers to the largest cp value which is under the dashed line; the intercept of this line equals to the minimum xerror plus its standard deviation 0.642514(0.54545+ 0.097064). Check the table, the value in the range of (0.045455, 0.068182), from the cp plot, the table should larger than 0.642514. So use the cp is 0.032 to prune the tree.

Pruning a tree

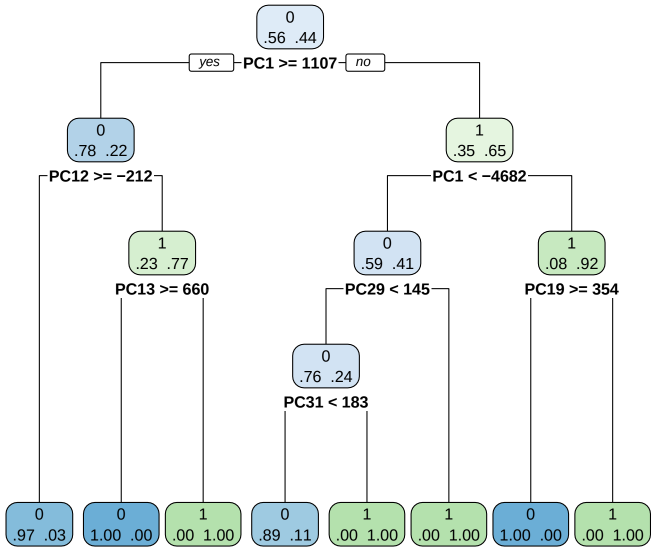


Figure 3: Pruned Classification Tree (cp=0.032)

After pruning trees with cp =0.032. Check the plot of tree, the variable where the split happens are PC1, PC12, PC13,PC29,PC19,PC31. The terminal nodes of the tree with the 1 as the predicted class have a high probability and the terminal nodes of the tree with 0(-1) as the predicted class have a high probability.

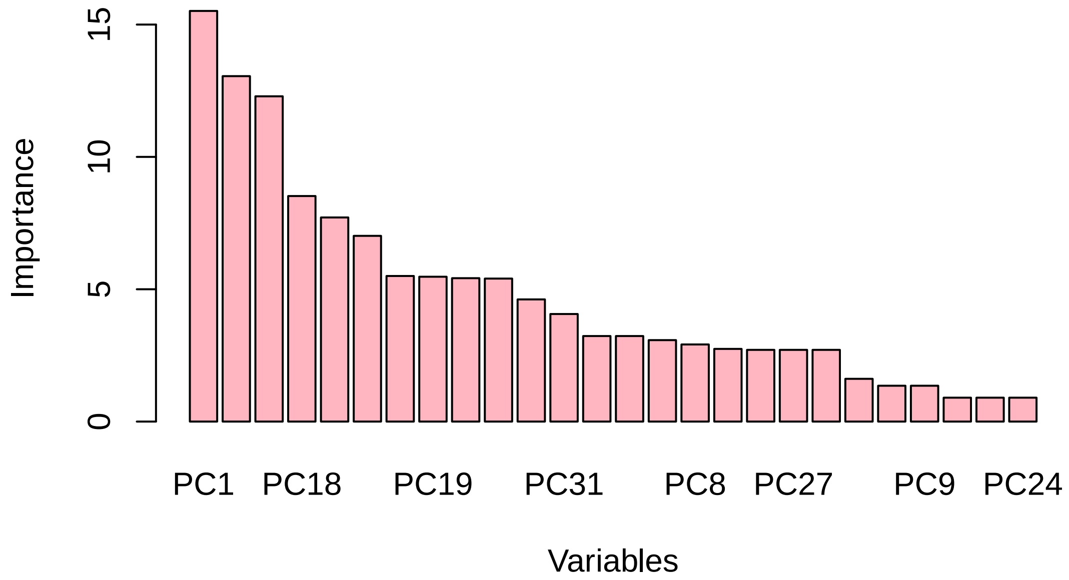


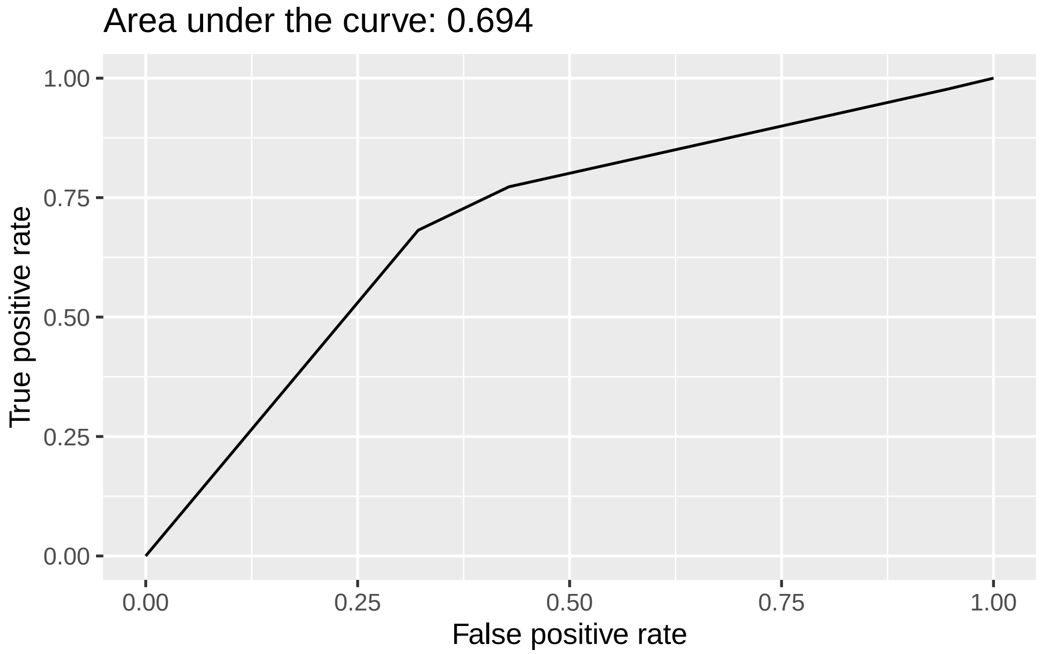
Figure 4: Variable Importance

Variable Importance

According to the classification tree, PC1 is the most important factor, followed by PC12 and PC2, then the PC18 and PC13, PC21 and PC24 are relatively unimportant.

ROC and AUC:

```
[1] “Accuracy: 0.68” Actual Predicted 0 1 0 38 14 1 18 30 [1] “AUC: 0.693587662337662”
```



The ROC curve show the classification performance. A higher AUC(close to 1) indicates a better-performing model. The AUC of classification tree is 0.694, which is greater than 0.5, show the classification effect is fair. After the predict of the test set, obtained the Confusion Matrix and the Accuracy. The Accuracy of test set is 0.68, which is poor. The classification tree model has some classification ability, but it is not particularly good.

Bagging Tree

Description: Repeatedly draw samples from the original dataset and build a classification tree on each bootstrapped sample. For a given test observation, record the class predicted from each tree and take a majority vote: the overall prediction is the most commonly occurring class across all the predictions.

Reason: Reduce the overfitting risk of a single tree, improve the stability and performance of the model. Can automatically adjust feature importance and easy to explain.

Set the dataset after the feature selection:

Start with randomForest() function and specify that the number of variables tried at each split,mtry,should be equal to the number of variables in the model. Set the ntree= 500, check the out-of-bag estimate of error rate. #### OOB:

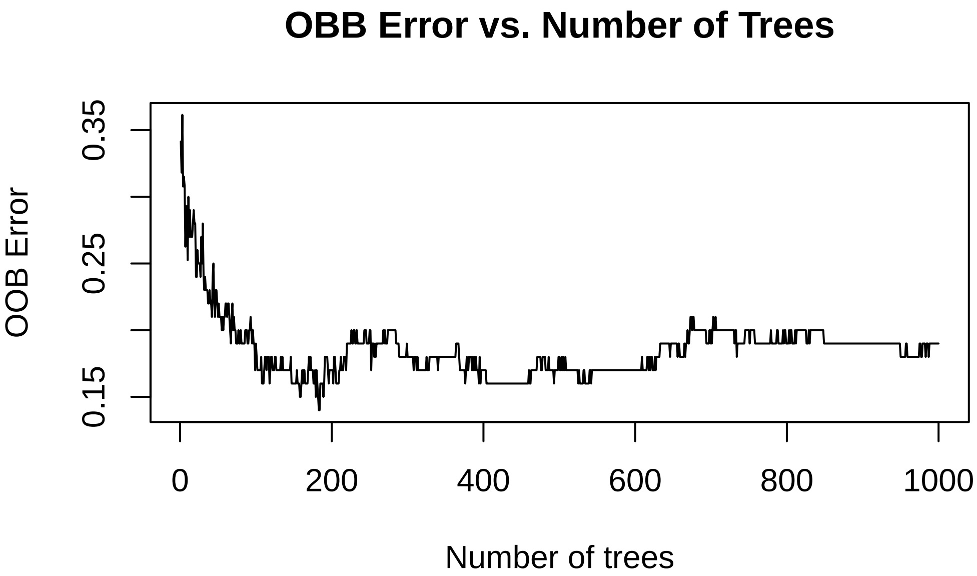


Figure 5: Model performance changes with the number of trees

According to the graph, in the range of (200,500), the OOB error starts to stabilize after decreasing. Select the minimum number of trees that can stabilize the OBB error, there choose ntree = 200.

new model:

Building a new bagging forest model with ntree =200.

Variable Importance:

According to the bagging forest, 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.

ROC and AUC:

```
[1] "Accuracy: 0.79"
```

```
         Actual
Predicted  0  1
        0 45 10
```

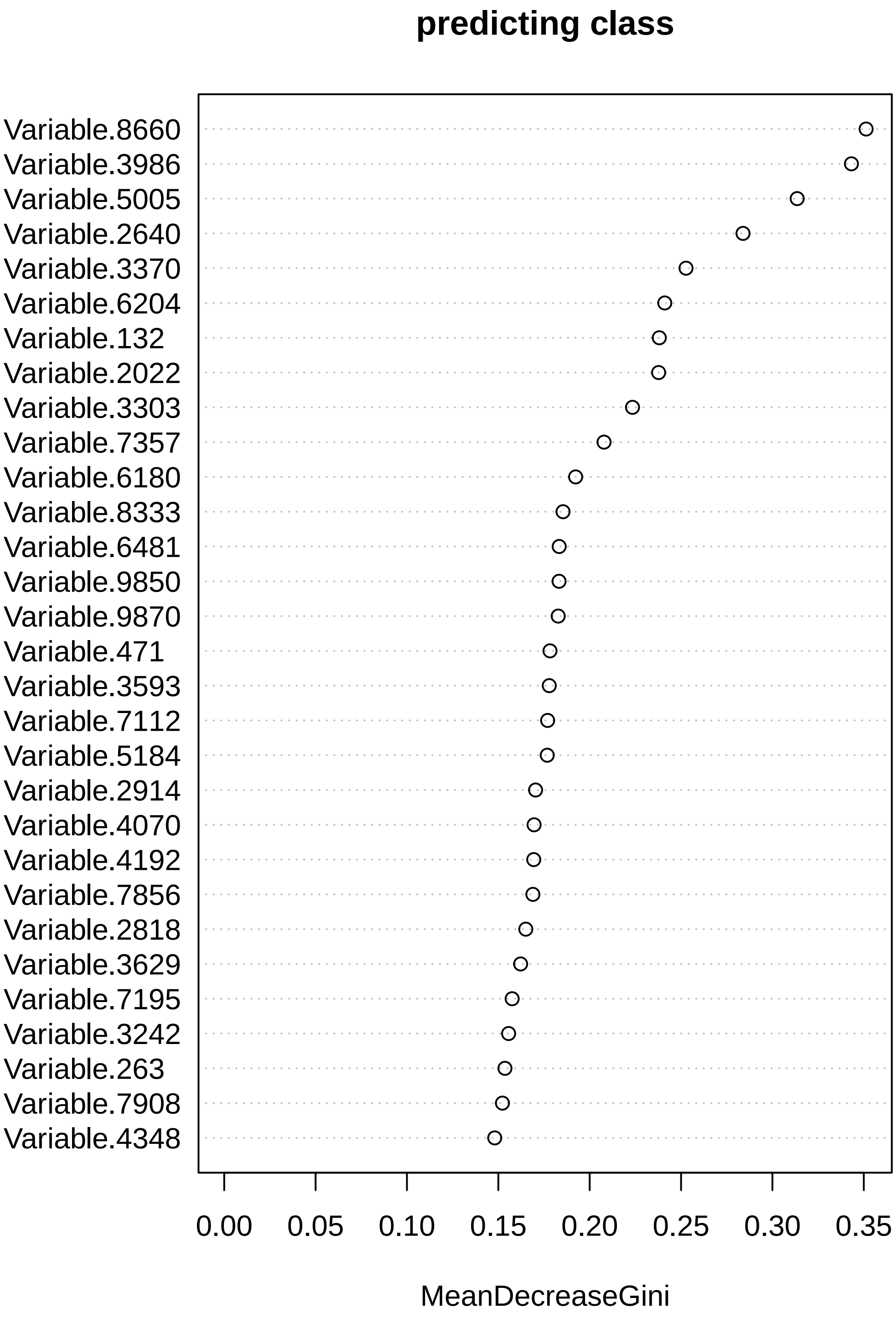
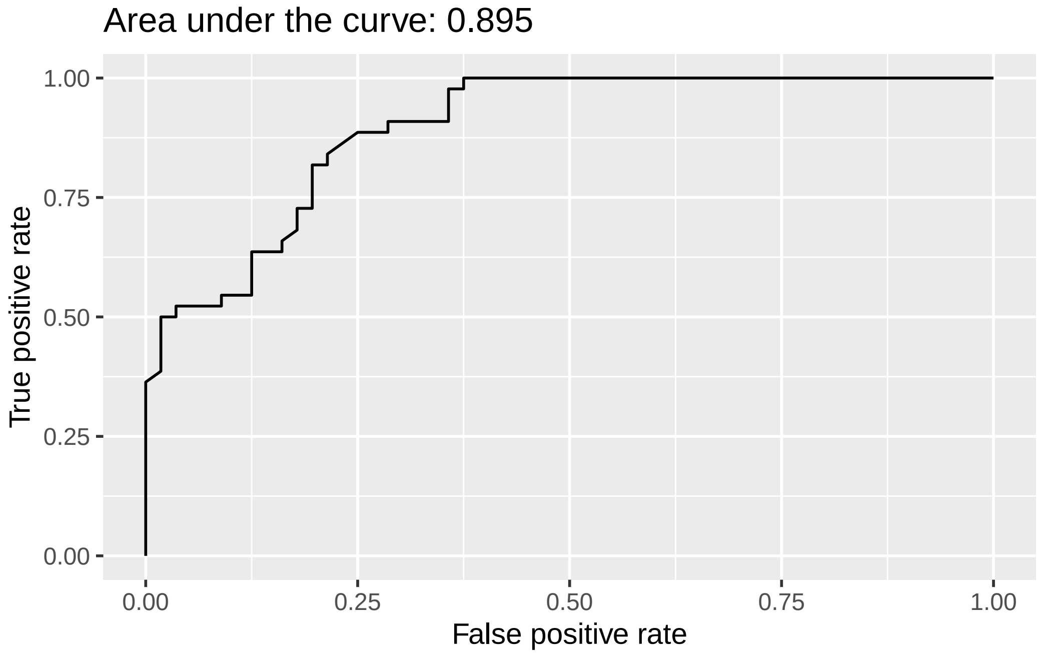


Figure 6: Variable Importance in Bagging Tree

```
      1 11 34
```

```
[1] "AUC: 0.895292207792208"
```



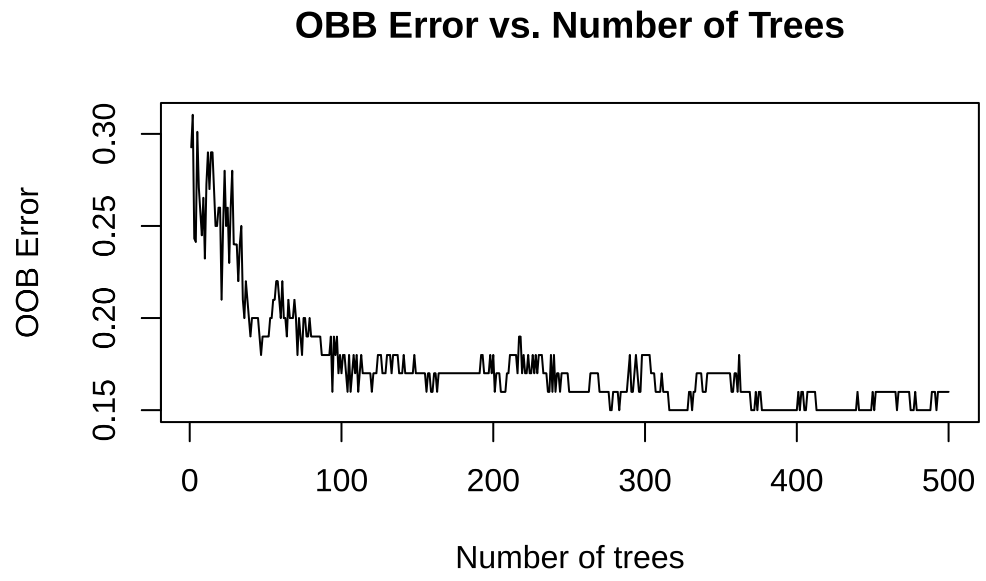
Prediction: After the predict of the test set, obtained the Confusion Matrix and the Accuracy. The Accuracy of test set is 0.79, which is good. The bagging forest model has good classification ability.

Random Forests

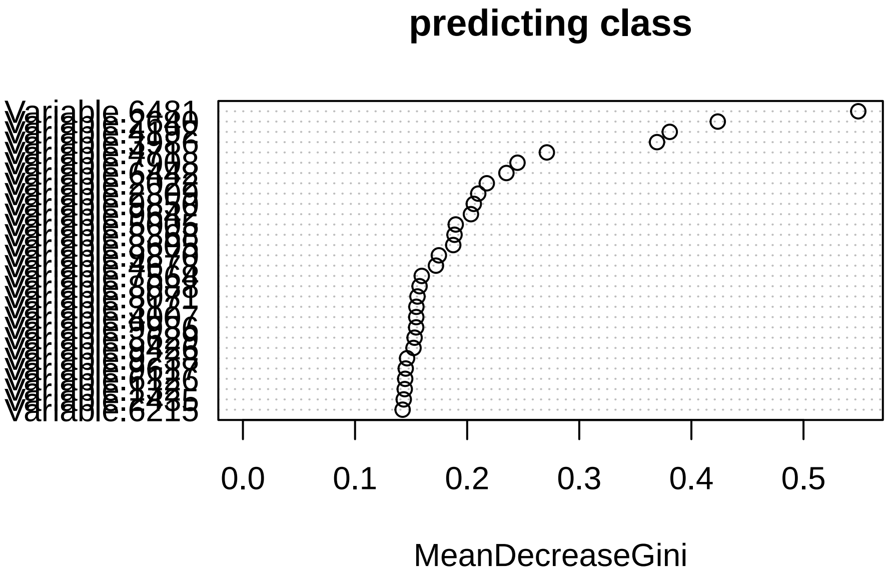
Description: Random forests improve upon bagging by decorrelating the individual trees. By forcibly excluding a random subset of variables, the correlation between any pair of trees is reduced. Therefore the average predictions will be more reliable.

Reason: Reduce the overfitting risk of a single tree, avoid strong feature influence ,improve the stability and performance of the model, make the average predictions will be more reliable.

OBB(out of bag)



Variable Importance:

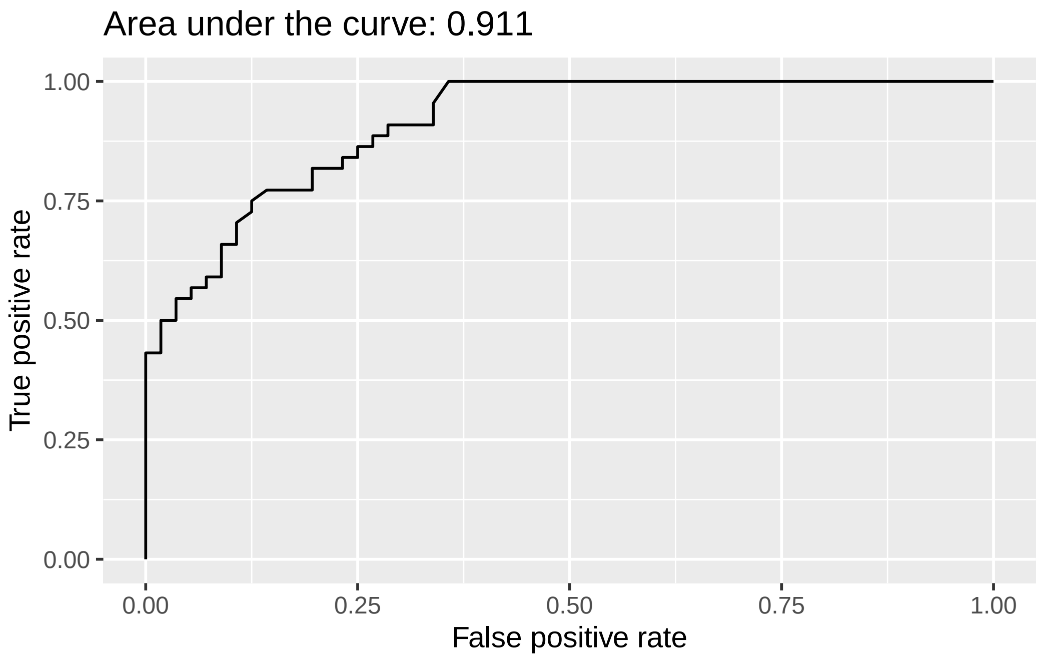


According to the random forests, Variable.6481 is the most important factor, followed by Variable.2640 and Variable.4192, Variable.3986and Variable.471, Variable.2435and Variable.6215 are relatively unimportant.

```
[1] "Accuracy: 0.81"
```

```
          Actual
Predicted  0  1
        0 45  8
        1 11 36
```

```
[1] "AUC: 0.91112012987013"
```



The ROC curve show the classification performance. A higher AUC(close to 1) indicates a better-performing model. The AUC of bagging forest is 0.911, which is greater than 0.5 and close to 1, show the classification effect is excellent.

After the predict of the test set, obtained the Confusion Matrix and the Accuracy. The Accuracy of test set is 0.81, which is good. The bagging forest model has good classification ability.

Discriminant Analysis

Test of boxm

```
    Box's M-test for Homogeneity of Covariance Matrices
```

```
data:  traindata[, -ncol(traindata)]
Chi-Sq (approx.) = NaN, df = 1128, p-value = NA


    Shapiro-Wilk normality test

data:  traindata[, 2]
W = 0.67816, p-value = 1.719e-13


    Shapiro-Wilk normality test

data:  traindata[, 3]
W = 0.88264, p-value = 2.351e-07
```

Lniear Discrimant Analysis

```
acc: 0.8
```

```
     0  1
  0 47 11
  1  9 33
```

QDA

```
acc: 0.79
```

```
     0  1
  0 48 13
  1  8 31
```

Roc

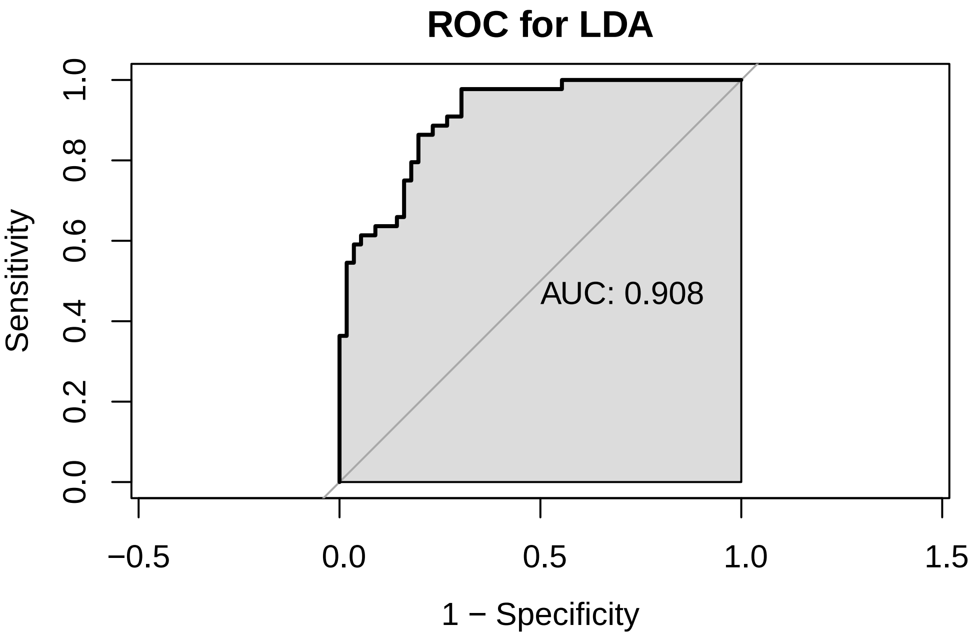


Figure 7: ROC

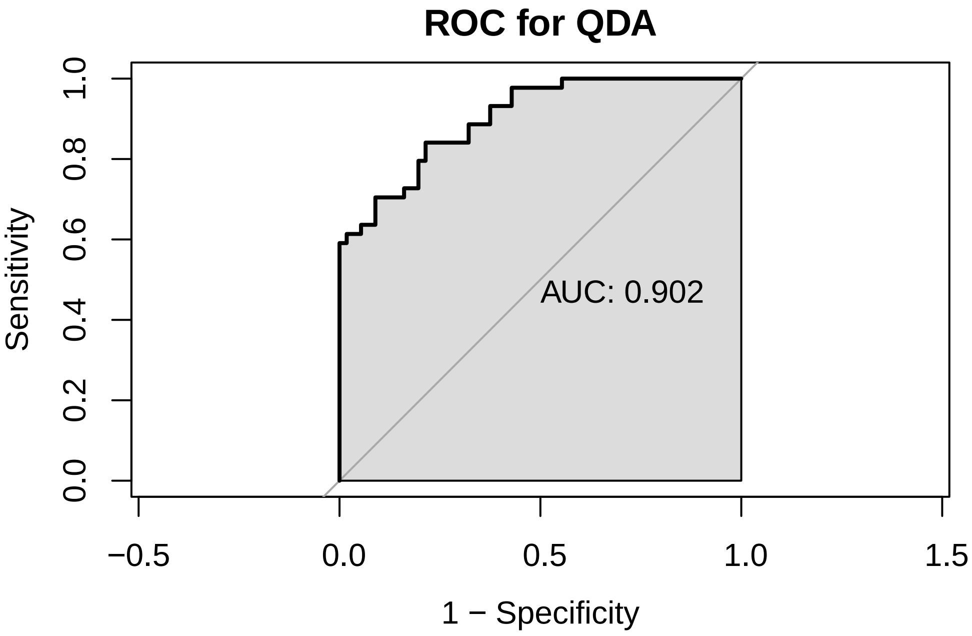


Figure 8: ROC

Distribution of Posterior Probabilities

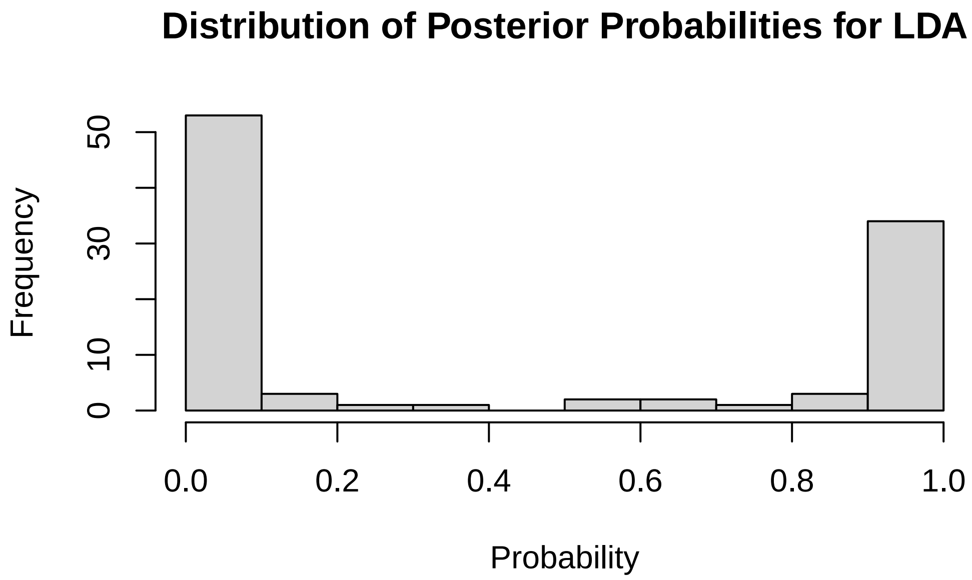


Figure 9: Distribution of Posterior Probabilities

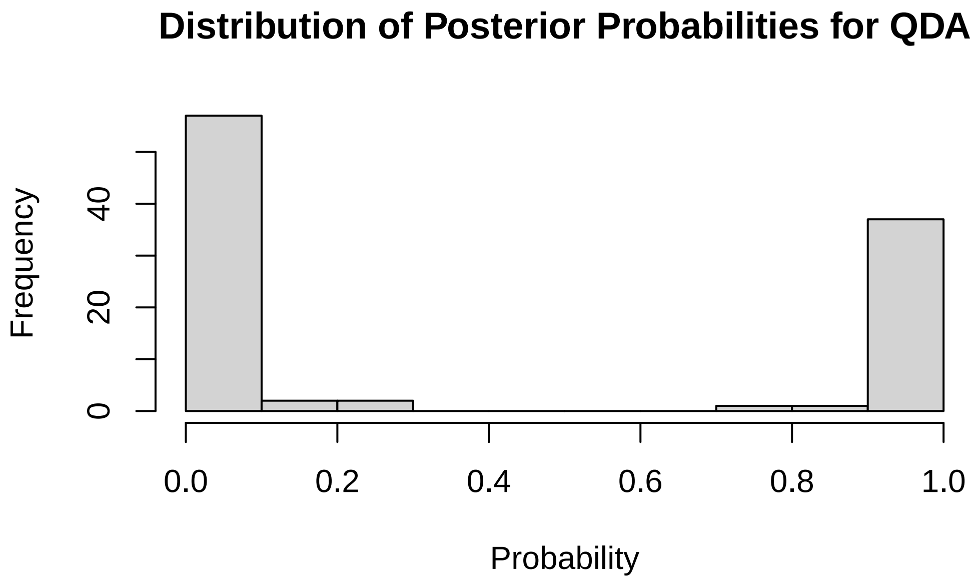


Figure 10: Distribution of Posterior Probabilities

nn work

Description: Neural networks enhance predictive performance by learning complex nonlinear relationships between features and the target variable. By incorporating multiple hidden layers and a structured activation function, the model captures intricate patterns in the data. Additionally, the architecture prevents over-reliance on any single feature, ensuring a more generalizable representation.

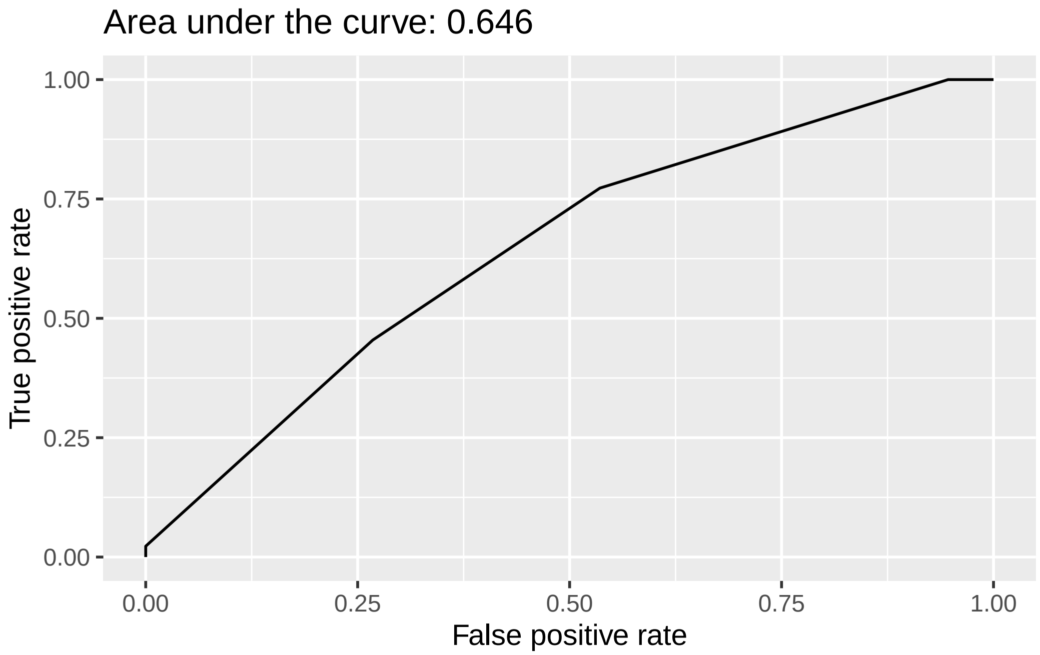
Reason: Reduce overfitting risk: By using multiple hidden layers and nonlinear activation functions, the model generalizes better to unseen data, mitigating overfitting. Avoid strong feature influence: The neural network distributes learning across multiple neurons and layers, preventing any single feature from dominating the predictions. Improve stability and performance: The network architecture enables robust learning from complex data distributions, enhancing overall model reliability. Ensure more reliable average predictions: Through iterative optimization and weight adjustments, the network minimizes bias and variance, leading to stable and consistent predictions.

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"
```

```
         Actual
Predicted  0  1
        0 41 24
        1 15 20
```

```
[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 → 60 → 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.

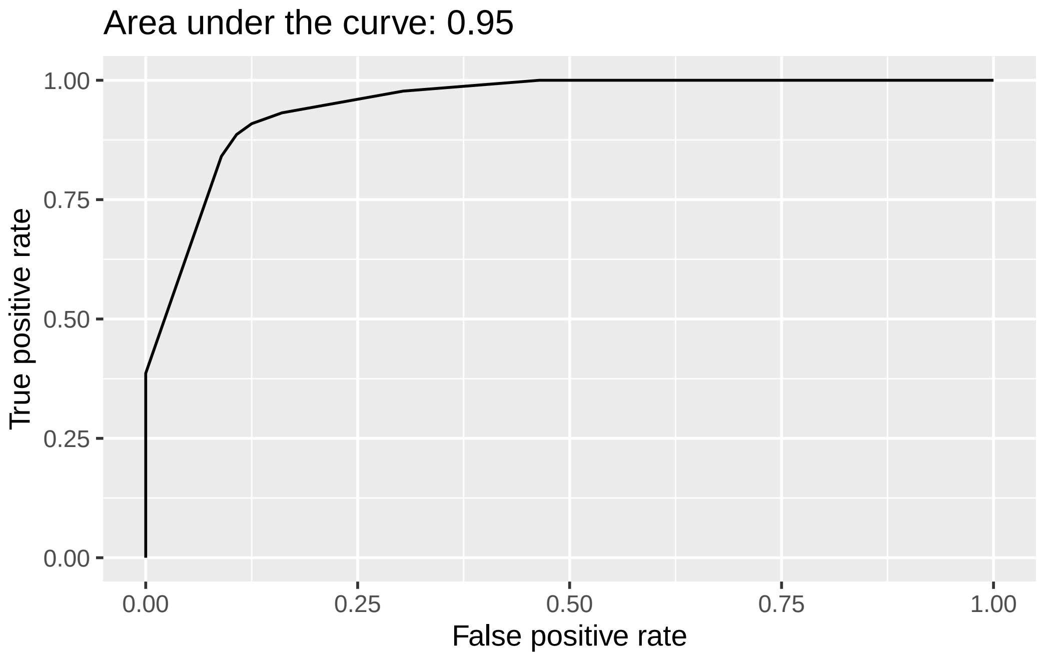
The training performance improved significantly, with the AUC increasing from 0.64 to 0.95. This dramatic improvement suggests that the adjusted network architecture effectively enhanced the model's ability to learn complex patterns, leading to a much better classification performance. The increased 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.

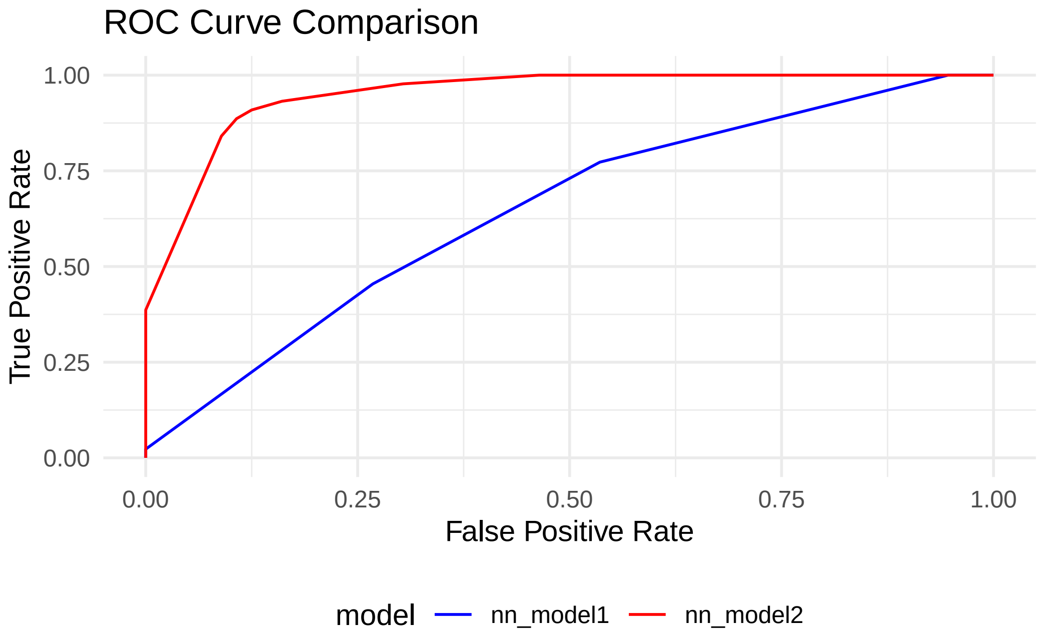
```
[1] "Accuracy: 0.89"
```

```
         Actual
Predicted  0  1
```

```
      0 50  5
      1  6 39
```

```
[1] "AUC: 0.950081168831169"
```


Area under the curve: 0.95
True positive rate
False positive rate


ROC Curve Comparison
True Positive Rate
False Positive Rate
model
nn_model1
nn_model2