Exercise 2.1

2025-01-26

1. Data Analysis and Preprocessing

1.0 Load necessary libraries

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
## corrplot 0.95 loaded
```

1.1 Open Dataset

1.2 Dataset summary

```
##
## Feature matrix dimensions: 77 200
```

```
##
## Response variable levels: -1 1
```

1.3 Check for missing values

```
##
##
## Missing values check:
```

```
##
## Features missing values: 0
```

```
##
## Response missing values: 0
```

1.4 Class balance check

```
##
##
## Class balance:
```

```
##
##        -1         1
## 0.5194805 0.4805195
```

1.5 Check scaling need

```
##
##
## Scaling check:
```

```
##
## Feature mean range: -0.14 0.12
```

```
##
## Feature SD range: 0.92 1.08
```

1.6 Check for multicollinearity

```
##
##
## Highly correlated feature pairs (|r| > 0.8): 1
```

```
##
## Highly Correlated Feature Pair 1 :
## Feature 1: V2
## Feature 2: V3
## Correlation between V2 and V3 : 0.83
```

1.7 Summary of the data analysis

1. **High-Dimensional Data with Few Observations**:
   - The feature matrix has dimensions of **77 observations × 200 features**, indicating that the number of features far exceeds the number of samples. This is a classic case of the "curse of dimensionality," where traditional statistical methods may struggle due to overfitting and multicollinearity.
   - With only one pair of highly correlated features out of 200, the dataset does not exhibit widespread multicollinearity. This is a good sign for most modeling approaches, including regularized models (Lasso, Ridge, Elastic Net) and tree-based methods.
2. **Class Imbalance and Response Variable**:
   - The response variable is binary, with levels `-1` and `1`. The class distribution is relatively balanced, with approximately **52% (-1)** and **48% (1)**, reducing concerns about severe class imbalance.
   - No missing values were detected in either the features or the response variable, ensuring data completeness.
3. **Scaling and Feature Standardization**:
   - The features have been scaled appropriately, as evidenced by the mean range (`-0.14 to 0.12`) and standard deviation range (`0.92 to 1.08`). This ensures that all features are on a comparable scale. Nonetheless, we will do additional scaling on the next steps.
4. **Challenges with Traditional Methods**:
   - Basic linear models like OLS regression are unsuitable for this dataset due to high dimensionality (p=200) and small sample size (n=77). Mathematically, the design matrix X yields a singular, non-invertible X X matrix when p>n, making it impossible to compute a unique solution for the OLS coefficients =(X X)^(−1)X y. Even if p were reduced below n, OLS models would still suffer from overfitting and instability in such high-dimensional settings.
   - Similarly, dimensionality reduction and feature selection techniques such as PCA , stepwise feature selection , and ANOVA-based feature selection are unlikely to be effective in this scenario. Even if we reduce the number of features (e.g., from 200 to 100), the dimensionality would still exceed the number of observations (n=77), leading to potential overfitting, loss of interpretability, and unreliable results. These methods struggle with the high-dimensional nature of the dataset and do not inherently address the p>n problem.
5. **Cross-Validation and Train-Test Split**:
   - Given the small dataset, **cross-validation** is essential to ensure robust model evaluation. We will use **5-fold cross-validation** to maximize the use of the limited data while minimizing variance in performance estimates.
   - The dataset will be split into **80% training** and **20% testing** subsets. Model performance will be evaluated on the test set using the **AUC-ROC metric**, which is well-suited for binary classification tasks and provides insight into the trade-off between true positive and false positive rates.
6. **Model Selection**:
   - To address the challenges posed by high dimensionality, we will focus on **regularized models** that can handle multicollinearity and feature selection:
     - **Lasso (L1 regularization)**: Encourages sparsity by shrinking less important feature coefficients to zero, effectively performing feature selection.
     - **Ridge (L2 regularization)**: Penalizes large coefficients to reduce overfitting without eliminating features.
     - **Elastic Net**: Combines L1 and L2 regularization, balancing sparsity and stability, making it particularly useful for datasets with highly correlated features.

- Additionally, we will explore **tree-based models**, which are robust to high-dimensional data and do not require feature scaling:
  - **CART (Classification and Regression Trees)**: Implemented using the `rpart` package, this will serve as a baseline tree-based model.
  - **Random Forest**: An ensemble method that builds multiple decision trees and aggregates their predictions, providing improved accuracy and robustness against overfitting. To enhance interpretability and efficiency, we will apply VSURF (Variable Selection Using Random Forests), a feature selection method for high-dimensional data.

2. Regularized regression models

2.1 LASSO model

```
##
## === Model Fitting Summary ===
```

```
## GLMNET Cross-Validation Results:
```

```
## Best lambda (lambda.min): 0.1
```

```
## Number of lambda values tested: 10
```

```
## Fold count:
```

```
## Maximum AUC achieved: 1
```

```
##
## Cross-Validation Performance:
```

```
##     Lambda    AUC    SD
## 1  1.00000 0.5000 0.000
## 2  0.46416 0.5000 0.000
## 3  0.21544 0.9647 0.022
## 4  0.10000 1.0000 0.000
## 5  0.04642 1.0000 0.000
## 6  0.02154 1.0000 0.000
## 7  0.01000 1.0000 0.000
## 8  0.00464 1.0000 0.000
## 9  0.00215 1.0000 0.000
## 10 0.00100 1.0000 0.000
```

```
##
## === Confusion Matrix ===
```

```
##          Actual
## Predicted Class0 Class1
##    Class0     10      1
##    Class1      0      5
```

```
##
## Accuracy: 0.9375
```

```
##
## Sensitivity (Recall): 0.8333
```

```
##
## Specificity: 1
```

```
##
## Test AUC: 0.9833
```

```
##
## === Non-Zero Coefficients ===
```

```
##      Feature Coefficient
##  (Intercept)  0.05748145
##           V1  0.13305726
##           V2  0.52362006
##           V3  1.11549609
##           V5  0.21888920
##           V6  0.44056025
```

```
##
## Non-zero coefficients (including intercept): 6
```

```
##
## Non-zero coefficients (excluding intercept): 5
```

The model demonstrates high performance metrics, such as test AUC of 0.983, but these results are likely unreliable due to a very small test set (16 samples) and potential overfitting, as indicated by perfect cross-validation AUC scores (1) with no variance. The model selected 5 features out of 200, which indicates a considerable simplification of the model. Given the small dataset size (77 observations) and unstable lambda behavior, the model's trustworthiness is questionable.

2.2 RIDGE model

```
##
## === RIDGE Model Fitting Summary ===
```

```
## GLMNET Cross-Validation Results:
```

```
## Best lambda (lambda.min): 0.02154435
```

```
## Number of lambda values tested: 10
```

```
## Fold count:
```

```
## Maximum AUC achieved: 0.9708
```

```
##
## Cross-Validation Performance:
```

```
##     Lambda    AUC     SD
## 1  1.00000 0.9596 0.0175
## 2  0.46416 0.9596 0.0175
## 3  0.21544 0.9652 0.0145
## 4  0.10000 0.9652 0.0171
## 5  0.04642 0.9652 0.0171
## 6  0.02154 0.9708 0.0185
## 7  0.01000 0.9652 0.0231
## 8  0.00464 0.9652 0.0231
## 9  0.00215 0.9652 0.0231
## 10 0.00100 0.9652 0.0231
```

```
##
## Feature Impact Summary:
```

```
## All features are retained in Ridge regression
```

```
## Number of features: 200
```

```
##
## === Confusion Matrix ===
```

```
##          Actual
## Predicted Class0 Class1
##    Class0      9      1
##    Class1      1      5
```

```
##
## Accuracy: 0.875
```

```
##
## Sensitivity (Recall): 0.8333
```

```
##
## Specificity: 0.9
```

```
##
## Test AUC: 0.95
```

```
##
## === Coefficients (Ridge) ===
```

```
##      Feature   Coefficient
##           V3  0.6677428454
##           V2  0.5860708095
##           V1  0.4950715372
##           V5  0.4068465121
##           V6  0.3731892803
##           V4  0.3341601274
##           V7  0.2920198379
##          V67  0.2721345921
##         V109 -0.2641883915
##          V37  0.2391782741
##         V191 -0.2346461267
##          V38 -0.2317536534
##          V50 -0.2254794338
##          V85 -0.2193123418
##          V81 -0.2082334705
##         V159  0.2047373747
##         V145 -0.1954945688
##         V168 -0.1887311981
##         V117  0.1840854312
##          V27 -0.1831435201
##          V54 -0.1784206504
##          V52 -0.1720755301
##          V17  0.1693929709
##         V147 -0.1689209049
##         V126  0.1670306351
##          V31 -0.1668912737
##         V200 -0.1640237921
##          V36 -0.1565928771
##         V111 -0.1562466283
##         V131  0.1497727927
##         V188  0.1485950231
##          V94  0.1476397281
##         V124 -0.1437858669
##          V23 -0.1435621300
##         V146 -0.1428531772
##         V150 -0.1412585423
##         V125  0.1411453961
##         V139 -0.1408484776
##         V116 -0.1386625111
##          V75 -0.1361589862
##         V194  0.1325592599
##         V123 -0.1321479229
##         V136  0.1308955711
##         V105  0.1246050044
##          V66 -0.1223955534
##          V68 -0.1207932709
##         V137 -0.1200504999
##         V195  0.1194261582
##          V96 -0.1172784298
##          V61 -0.1166600245
##         V138 -0.1144530636
##          V65  0.1109986107
##          V69  0.1109938148
##         V167  0.1098788313
##         V193  0.1098763241
##          V24 -0.1080617215
##         V181  0.1078681314
##         V196  0.1064422597
##         V162 -0.1064347551
##         V127  0.1037018069
##         V120 -0.1030130554
##          V53  0.1021707772
##           V9  0.1014884313
##         V107  0.0999730306
##         V189 -0.0997422896
##         V175  0.0987917600
##         V184 -0.0974261058
##         V179  0.0962381191
##         V113 -0.0946776360
##          V98  0.0927937032
##         V197  0.0904013518
##         V122 -0.0897049174
##          V32 -0.0892781827
##          V11 -0.0889186596
##          V84  0.0883454969
##         V158 -0.0881920980
##         V165 -0.0878977301
##          V58  0.0872822089
##          V19 -0.0872436390
##          V47  0.0863894794
##         V114 -0.0859337059
##          V71  0.0851134514
##         V143 -0.0816824749
##         V177  0.0815772474
##          V49 -0.0804136223
##         V185  0.0786337645
##          V60  0.0775342777
##         V104 -0.0771089503
##          V10 -0.0765588106
##          V12  0.0758527839
##         V176 -0.0756172907
##          V97  0.0754903070
##         V187  0.0752714788
##  (Intercept)  0.0751850270
##         V144  0.0737784990
##          V33  0.0709921513
##         V152 -0.0702442315
##          V77  0.0700134826
##         V160  0.0700073864
##          V30  0.0689716717
##          V46 -0.0668084422
##          V35  0.0664126439
##         V102  0.0663829258
##         V173 -0.0654786466
##         V163 -0.0628194298
##         V166 -0.0626487611
##         V133  0.0618620420
##          V92 -0.0618296190
##          V73  0.0608282573
##         V178  0.0599533246
##         V103 -0.0589951738
##          V18  0.0574942826
##         V157  0.0558728172
##          V40  0.0535946836
##          V16 -0.0534346300
##          V64  0.0523381464
##         V132 -0.0520977073
##         V172  0.0518690034
##         V190  0.0511431754
##          V82  0.0507089708
##         V112  0.0506479297
##         V156  0.0502699103
##         V106  0.0494566490
##         V128  0.0482251106
##          V89 -0.0480117180
##         V108 -0.0477849734
##          V45 -0.0474736275
##         V169  0.0468537495
##          V25 -0.0463550496
##          V56  0.0461599060
##         V151 -0.0456293477
##          V86  0.0451717526
##         V148  0.0446776833
##         V192 -0.0434698925
##          V51 -0.0430611776
##         V199 -0.0407743827
##         V155 -0.0399950718
##         V135  0.0399852018
##         V182  0.0396406708
##          V90 -0.0389661776
##          V93  0.0384831053
##         V134  0.0374480140
##          V44  0.0371137066
##          V48  0.0370053848
##          V14 -0.0369921720
##          V79  0.0369465112
##          V70 -0.0352342196
##          V55 -0.0325822561
##          V72  0.0309068784
##          V63  0.0307590797
##         V115 -0.0301433266
##          V59  0.0298726016
##         V149  0.0298484362
##         V130  0.0295905899
##          V83  0.0279211717
##          V43 -0.0277938308
##         V129  0.0256117776
##          V80 -0.0244953809
##         V164  0.0235916890
##         V110 -0.0234100097
##          V20 -0.0230345961
##          V15 -0.0225357888
##          V78 -0.0208546644
##          V13  0.0207843563
##          V91 -0.0202158718
##          V21  0.0200253283
##           V8  0.0196890975
##          V34  0.0196293347
##         V118  0.0192163008
##         V141  0.0168501684
##         V183 -0.0166909053
##          V62  0.0165510488
##          V95  0.0161147914
##         V121  0.0155240870
##          V41  0.0133041623
##         V119  0.0130034770
##          V42 -0.0120060759
##         V180 -0.0115244297
##         V140  0.0111927226
##         V101 -0.0110880828
##         V154 -0.0098983849
##         V153 -0.0094471852
##          V39 -0.0089896697
##          V87  0.0089673599
##         V171 -0.0088269542
##          V74 -0.0088227296
##          V76 -0.0079005151
##         V186 -0.0075874230
##          V29  0.0068840279
##          V99 -0.0067686040
##          V22 -0.0066317723
##         V174  0.0059697575
##          V57  0.0055038902
##          V26 -0.0046159299
##          V88  0.0045222519
##         V170  0.0040890453
##          V28 -0.0039710509
##         V198 -0.0034603503
##         V100  0.0017944609
##         V142 -0.0012035609
##         V161 -0.0004133932
```

```
##
## Coefficient Statistics:
```

```
## L2 Norm: 3.348907
```

```
## Maximum Absolute Coefficient: 0.6677428
```

```
## Minimum Absolute Coefficient: 0.0004133932
```

The Ridge model achieves strong performance (0.95 test AUC) while retaining all 200 features, avoiding the over-regularization concerns of Lasso but sacrificing interpretability. It shows less evidence of overfitting than Lasso, with no perfect cross-validation AUC and more stable lambda behavior, though the small dataset (77 samples, 200 features) and tiny test set (16 samples) still raise reliability concerns. While Ridge appears marginally more trustworthy due to its consistent regularization and avoidance of extreme sparsity, both models likely suffer from overfitting and require validation on a larger dataset. The Ridge model's inclusion of all features may improve robustness but also increases complexity without clear gains in generalizability.

2.3 Elastic Net model

```
##
## === Elastic Net Tuning Progress ===
```

```
## Alpha: 0.2 | Best AUC: 1.0000
## Alpha: 0.5 | Best AUC: 1.0000
## Alpha: 0.8 | Best AUC: 1.0000
```

```
##
## === Cross-Validation Results Table ===
```

```
##    Alpha      Lambda       AUC          SD
## 1    0.2 0.001000000 0.9831382 0.017036565
## 2    0.2 0.002154435 1.0000000 0.000000000
## 3    0.2 0.004641589 1.0000000 0.000000000
## 4    0.2 0.010000000 1.0000000 0.000000000
## 5    0.2 0.021544347 1.0000000 0.000000000
## 6    0.2 0.046415888 1.0000000 0.000000000
## 7    0.2 0.100000000 1.0000000 0.000000000
## 8    0.2 0.215443469 1.0000000 0.000000000
## 9    0.2 0.464158883 1.0000000 0.000000000
## 10   0.2 1.000000000 1.0000000 0.000000000
## 11   0.5 0.001000000 0.5000000 0.000000000
## 12   0.5 0.002154435 0.9577674 0.025894199
## 13   0.5 0.004641589 1.0000000 0.000000000
## 14   0.5 0.010000000 1.0000000 0.000000000
## 15   0.5 0.021544347 1.0000000 0.000000000
## 16   0.5 0.046415888 1.0000000 0.000000000
## 17   0.5 0.100000000 1.0000000 0.000000000
## 18   0.5 0.215443469 0.9943794 0.005678855
## 19   0.5 0.464158883 0.9887588 0.011357710
## 20   0.5 1.000000000 0.9887588 0.011357710
## 21   0.8 0.001000000 0.5000000 0.000000000
## 22   0.8 0.002154435 0.9505074 0.018749019
## 23   0.8 0.004641589 0.9831382 0.017036565
## 24   0.8 0.010000000 1.0000000 0.000000000
## 25   0.8 0.021544347 1.0000000 0.000000000
## 26   0.8 0.046415888 1.0000000 0.000000000
## 27   0.8 0.100000000 1.0000000 0.000000000
## 28   0.8 0.215443469 1.0000000 0.000000000
## 29   0.8 0.464158883 1.0000000 0.000000000
## 30   0.8 1.000000000 1.0000000 0.000000000
```

```
##
## === Elastic Net Final Model ===
```

```
## Best alpha: 0.2
```

```
## Best lambda: 0.4641589
```

```
## Validation AUC: 1
```

```
##
## === Confusion Matrix ===
```

```
##          Actual
## Predicted Class0 Class1
##    Class0     10      1
##    Class1      0      5
```

```
##
## Accuracy: 0.9375
```

```
##
## Sensitivity (Recall): 0.8333
```

```
##
## Specificity: 1
```

```
##
## Test AUC: 1
```

```
##
## === Elastic Net Coefficients ===
```

```
##      Feature Coefficient
##           V3  0.35771257
##           V2  0.33151676
##           V1  0.19741952
##           V6  0.15318102
##           V5  0.14636532
##           V4  0.07730440
##  (Intercept)  0.04231159
##         V191 -0.02835118
##         V159  0.02283345
##         V109 -0.01887627
##           V7  0.00000000
##           V8  0.00000000
##           V9  0.00000000
##          V10  0.00000000
##          V11  0.00000000
##          V12  0.00000000
##          V13  0.00000000
##          V14  0.00000000
##          V15  0.00000000
##          V16  0.00000000
##          V17  0.00000000
##          V18  0.00000000
##          V19  0.00000000
##          V20  0.00000000
##          V21  0.00000000
##          V22  0.00000000
##          V23  0.00000000
##          V24  0.00000000
##          V25  0.00000000
##          V26  0.00000000
##          V27  0.00000000
##          V28  0.00000000
##          V29  0.00000000
##          V30  0.00000000
##          V31  0.00000000
##          V32  0.00000000
##          V33  0.00000000
##          V34  0.00000000
##          V35  0.00000000
##          V36  0.00000000
##          V37  0.00000000
##          V38  0.00000000
##          V39  0.00000000
##          V40  0.00000000
##          V41  0.00000000
##          V42  0.00000000
##          V43  0.00000000
##          V44  0.00000000
##          V45  0.00000000
##          V46  0.00000000
##          V47  0.00000000
##          V48  0.00000000
##          V49  0.00000000
##          V50  0.00000000
##          V51  0.00000000
##          V52  0.00000000
##          V53  0.00000000
##          V54  0.00000000
##          V55  0.00000000
##          V56  0.00000000
##          V57  0.00000000
##          V58  0.00000000
##          V59  0.00000000
##          V60  0.00000000
##          V61  0.00000000
##          V62  0.00000000
##          V63  0.00000000
##          V64  0.00000000
##          V65  0.00000000
##          V66  0.00000000
##          V67  0.00000000
##          V68  0.00000000
##          V69  0.00000000
##          V70  0.00000000
##          V71  0.00000000
##          V72  0.00000000
##          V73  0.00000000
##          V74  0.00000000
##          V75  0.00000000
##          V76  0.00000000
##          V77  0.00000000
##          V78  0.00000000
##          V79  0.00000000
##          V80  0.00000000
##          V81  0.00000000
##          V82  0.00000000
##          V83  0.00000000
##          V84  0.00000000
##          V85  0.00000000
##          V86  0.00000000
##          V87  0.00000000
##          V88  0.00000000
##          V89  0.00000000
##          V90  0.00000000
##          V91  0.00000000
##          V92  0.00000000
##          V93  0.00000000
##          V94  0.00000000
##          V95  0.00000000
##          V96  0.00000000
##          V97  0.00000000
##          V98  0.00000000
##          V99  0.00000000
##         V100  0.00000000
##         V101  0.00000000
##         V102  0.00000000
##         V103  0.00000000
##         V104  0.00000000
##         V105  0.00000000
##         V106  0.00000000
##         V107  0.00000000
##         V108  0.00000000
##         V110  0.00000000
##         V111  0.00000000
##         V112  0.00000000
##         V113  0.00000000
##         V114  0.00000000
##         V115  0.00000000
##         V116  0.00000000
##         V117  0.00000000
##         V118  0.00000000
##         V119  0.00000000
##         V120  0.00000000
##         V121  0.00000000
##         V122  0.00000000
##         V123  0.00000000
##         V124  0.00000000
##         V125  0.00000000
##         V126  0.00000000
##         V127  0.00000000
##         V128  0.00000000
##         V129  0.00000000
##         V130  0.00000000
##         V131  0.00000000
##         V132  0.00000000
##         V133  0.00000000
##         V134  0.00000000
##         V135  0.00000000
##         V136  0.00000000
##         V137  0.00000000
##         V138  0.00000000
##         V139  0.00000000
##         V140  0.00000000
##         V141  0.00000000
##         V142  0.00000000
##         V143  0.00000000
##         V144  0.00000000
##         V145  0.00000000
##         V146  0.00000000
##         V147  0.00000000
##         V148  0.00000000
##         V149  0.00000000
##         V150  0.00000000
##         V151  0.00000000
##         V152  0.00000000
##         V153  0.00000000
##         V154  0.00000000
##         V155  0.00000000
##         V156  0.00000000
##         V157  0.00000000
##         V158  0.00000000
##         V160  0.00000000
##         V161  0.00000000
##         V162  0.00000000
##         V163  0.00000000
##         V164  0.00000000
##         V165  0.00000000
##         V166  0.00000000
##         V167  0.00000000
##         V168  0.00000000
##         V169  0.00000000
##         V170  0.00000000
##         V171  0.00000000
##         V172  0.00000000
##         V173  0.00000000
##         V174  0.00000000
##         V175  0.00000000
##         V176  0.00000000
##         V177  0.00000000
##         V178  0.00000000
##         V179  0.00000000
##         V180  0.00000000
##         V181  0.00000000
##         V182  0.00000000
##         V183  0.00000000
##         V184  0.00000000
##         V185  0.00000000
##         V186  0.00000000
##         V187  0.00000000
##         V188  0.00000000
##         V189  0.00000000
##         V190  0.00000000
##         V192  0.00000000
##         V193  0.00000000
##         V194  0.00000000
##         V195  0.00000000
##         V196  0.00000000
##         V197  0.00000000
##         V198  0.00000000
##         V199  0.00000000
##         V200  0.00000000
```

```
##
## Regularization Statistics:
```

```
## L1 Norm (Sum|Coefficients|): 1.375872
```

```
## L2 Norm (Sum Coefficients^2): 0.3311711
```

```
## Sparsity Ratio: 0.9502488
```

The Elastic Net model achieves perfect test performance metrics (93.75% accuracy, 1.0 AUC) with high sparsity (95% of coefficients near zero), suggesting strong regularization and feature selection. However, the perfect AUC in both cross-validation and testing, along with zero variance, strongly indicates overfitting. Its results are overly optimistic and unreliable without further validation. The chosen alpha (0.2) leans more toward Ridge, but the lack of generalizability remains a critical concern.

Among the three regularized models, Ridge appears to be the better choice, although it is still susceptible to overfitting due to the limitations of the dataset.

3. CART models

3.1 Basic RPART

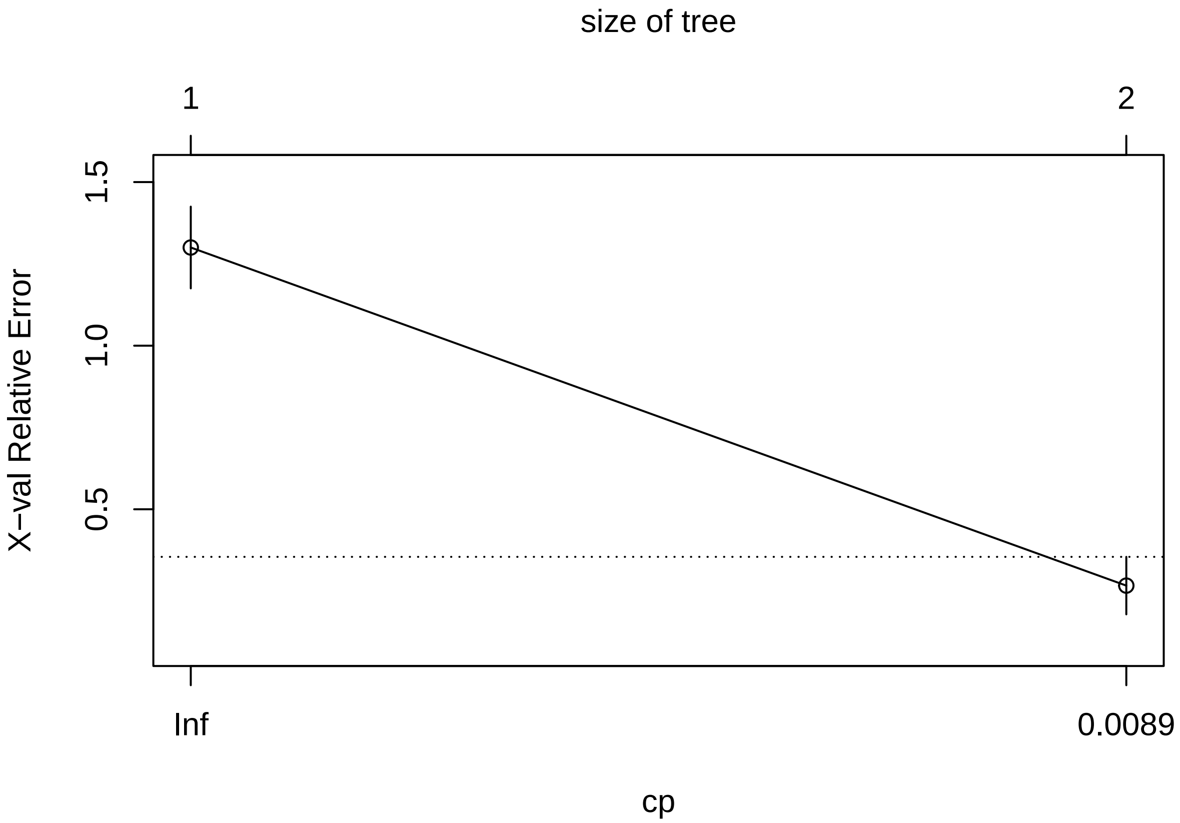
```
##
## === Cross-Validation Results ===
```

```
## Mean Cross-Validation AUC: 0.8105
```

```
##
## === RPART Model Summary ===
```

```
##
## Classification tree:
## rpart(formula = Y_train ~ ., data = data.frame(X_train_scaled,
##     Y_train = Y_train), method = "class", control = rpart.control(xval = 5,
##     cp = 1e-04, minsplit = 5, minbucket = 10, maxdepth = 8))
##
## Variables actually used in tree construction:
## [1] V2
##
```

```
## Root node error: 30/61 = 0.4918
##
## n= 61
##
##       CP nsplit rel error  xerror     xstd
## 1 8e-01      0       1.0 1.30000 0.125014
## 2 1e-04      1       0.2 0.26667 0.087881
```



```
##
## Optimal Complexity Parameter (CP): 1e-04
```

```
## Final Tree Size: 3 nodes
```

```
##
## === Confusion Matrix ===
```

```
##          Actual
## Predicted Class0 Class1
##    Class0      9      2
##    Class1      1      4
```

```
##
## Accuracy: 0.8125
```

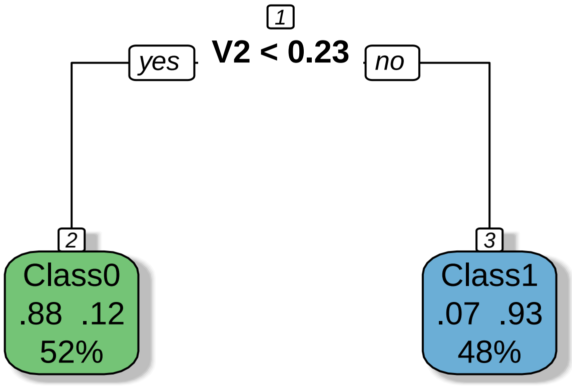
```
##
## Sensitivity (Recall): 0.6667
```

```
##
## Specificity: 0.9
```

```
##
## Test AUC: 0.7833
```

```
##
## === Variable Importance ===
```

```
##        V2        V3        V1      V109       V38       V17
## 19.767665 15.677804  8.861367  8.861367  7.498080  6.816436
```



The RPART model demonstrates moderate performance with a cross-validation AUC of 0.81 and a test AUC of 0.78, indicating no significant overfitting and better consistency compared to Ridge. The tree is based solely on one variable (V2), making the model extremely simple and transparent, but potentially underutilizing valuable information from other features. Let's fine-tune hyperparameters to improve robustness of the tree.

3.2 Fine-tuned RPART

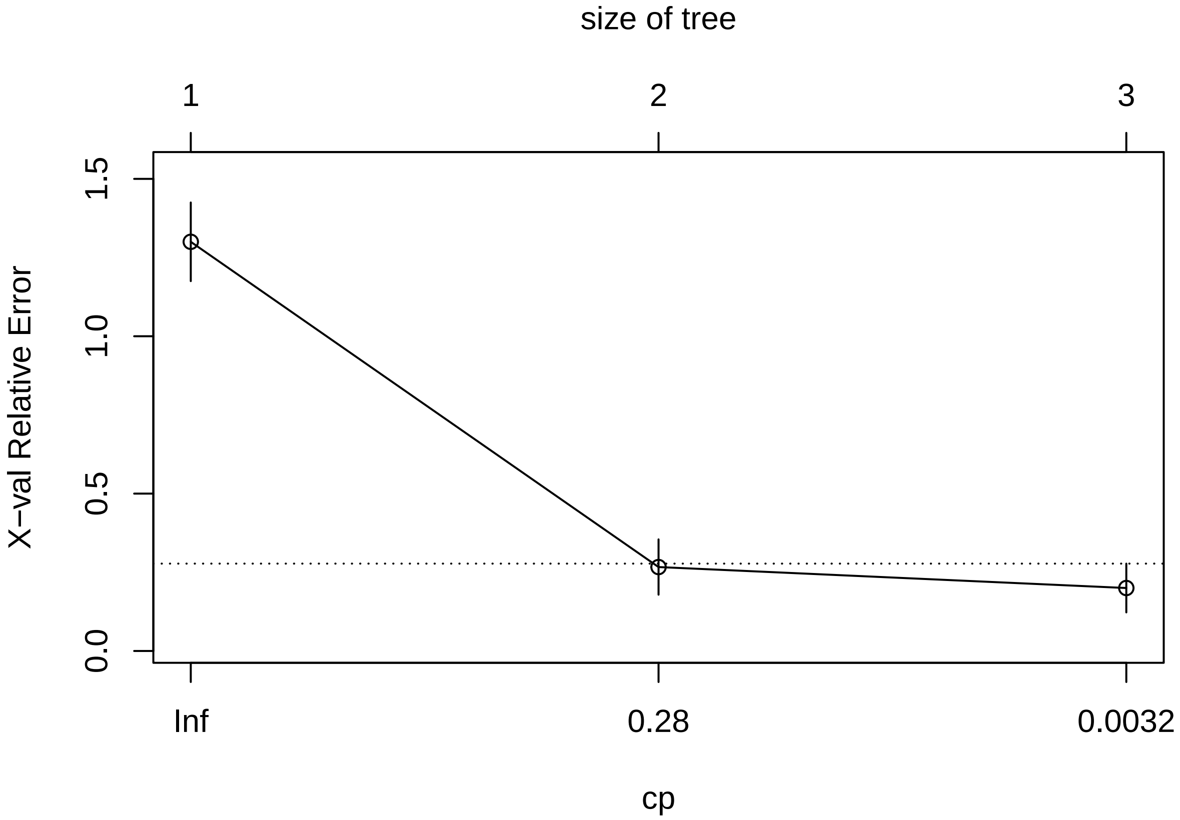
```
##
## === Cross-Validation Results ===
```

```
## Mean Cross-Validation AUC: 0.8441
```

```
##
## === RPART Model Summary ===
```

```
##
## Classification tree:
## rpart(formula = Y_train ~ ., data = data.frame(X_train_scaled,
##     Y_train = Y_train), method = "class", control = rpart.control(xval = 5,
##     cp = 1e-04, minsplit = 5, minbucket = 5, maxdepth = 5))
##
## Variables actually used in tree construction:
## [1] V2 V6
##
## Root node error: 30/61 = 0.4918
##
## n= 61
```

```
##
##      CP nsplit rel error  xerror     xstd
## 1 8e-01      0       1.0 1.30000 0.125014
## 2 1e-01      1       0.2 0.26667 0.087881
## 3 1e-04      2       0.1 0.20000 0.077530
```

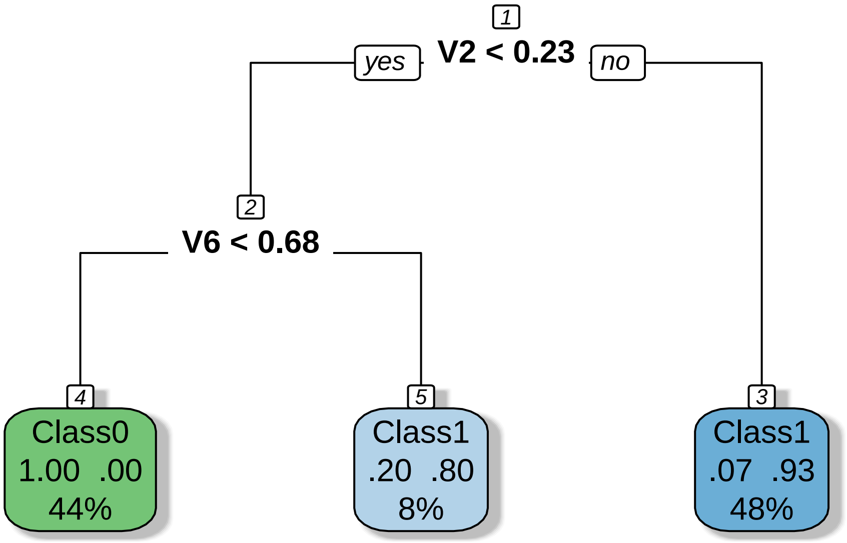


```
##
## Optimal Complexity Parameter (CP): 1e-04

## Final Tree Size: 5 nodes

##
## === Confusion Matrix ===

##          Actual
## Predicted Class0 Class1
##    Class0      8      0
##    Class1      2      6

##
## Accuracy: 0.875

##
## Sensitivity (Recall): 1

##
## Specificity: 0.8
```

```
##
## Test AUC: 0.9167
```

```
##
## === Variable Importance ===
```

```
##        V2        V3      V109        V1       V38       V17        V6       V23 
## 19.767665 15.677804 12.101367  8.861367  7.498080  6.816436  5.400000  3.240000 
##        V4       V43        V5 
##  2.160000  2.160000  2.160000
```



The tuned RPART model demonstrates improved performance, with a cross-validation AUC of 0.8441 and a test AUC of 0.9167, achieving better generalization and consistency compared to the initial version. Reducing `minbucket` to 5 was the only hyperparameter change that yielded positive results, as adjustments to other parameters like `cp`, `minsplit`, and `maxdepth` did not lead to further improvements. The resulting tree now uses two variables (V2 and V6) and grows to 5 nodes, balancing simplicity with the ability to capture more information from the data.

4. Random forest

4.1 Basic Random forest

```
##
## === Cross-Validation Results ===
```

```
## Mean AUC: 1
```

```
## Mean Accuracy: 0.9179
```

```
## Mean Sensitivity (Recall): 0.9167
```

```
## Mean Specificity: 0.9417
```

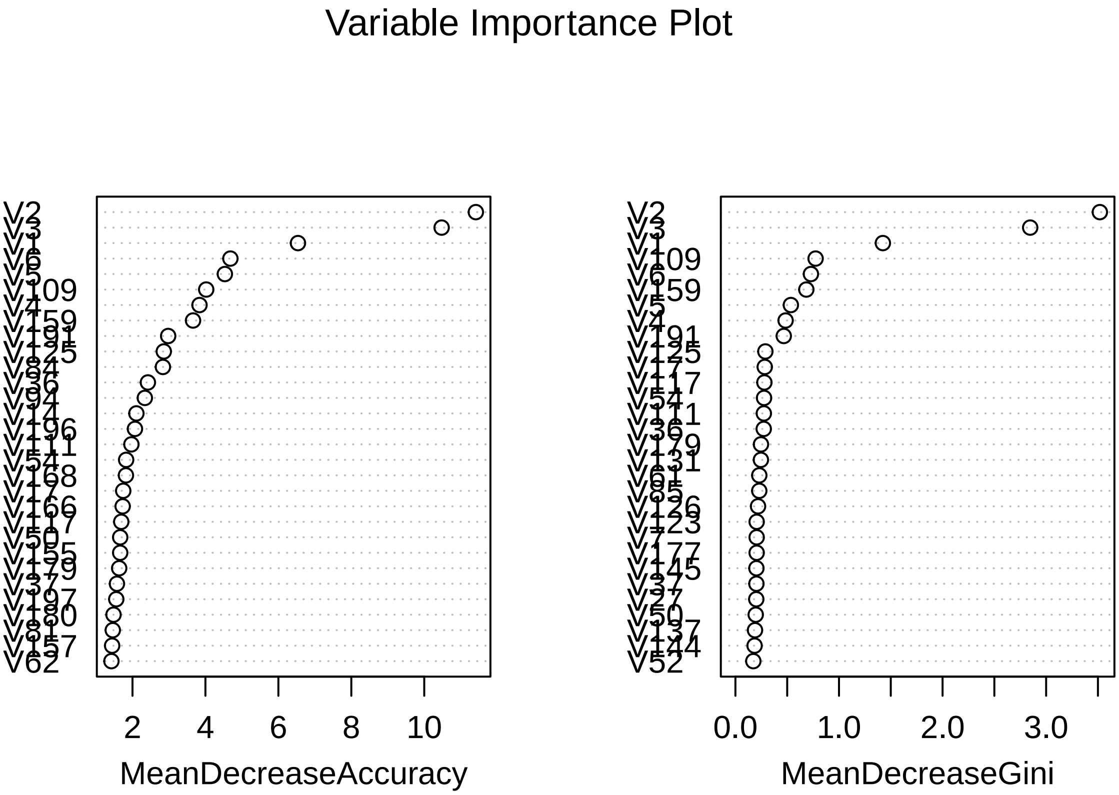
```
##
## === Confusion Matrix ===
##           Actual
## Predicted Class0 Class1
##    Class0      9      2
##    Class1      1      4
```

```
##
## Accuracy: 0.8125
```

```
##
## Sensitivity (Recall): 0.6667
```

```
##
## Specificity: 0.9
```

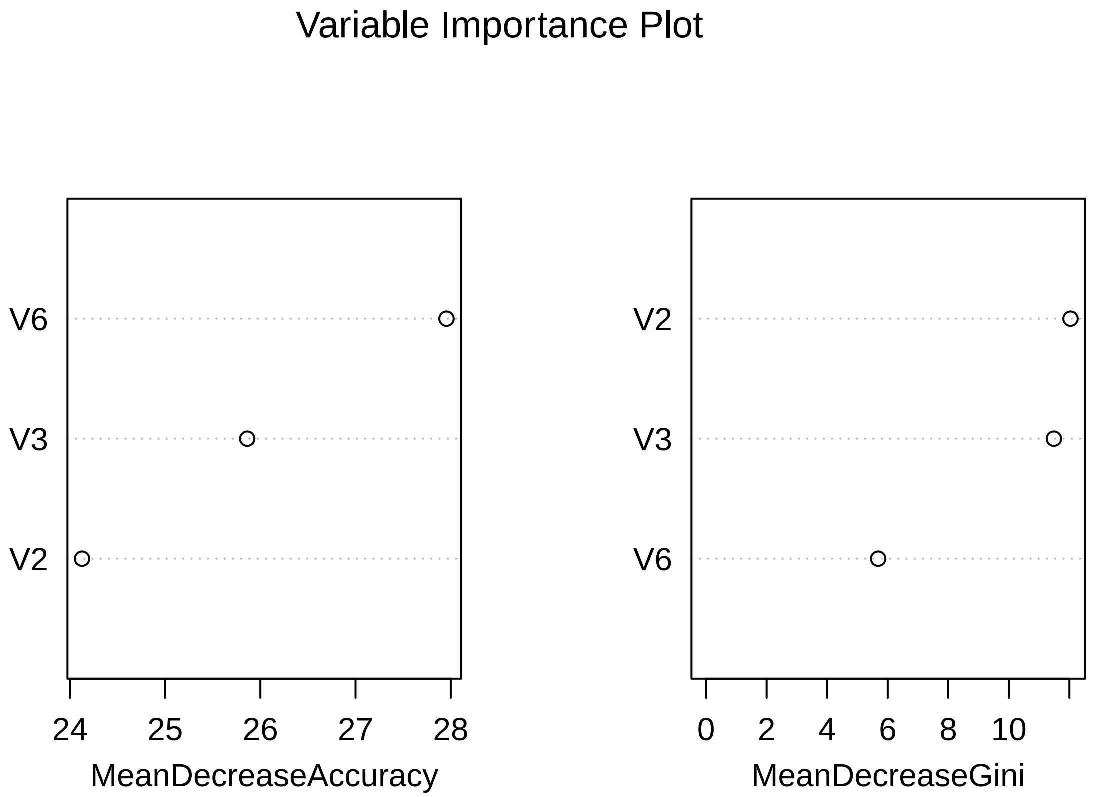
```
##
## Test AUC: 0.9333
```



Despite the Random Forest model achieving a slightly higher test AUC (0.93) compared to the fine-tuned RPART model (0.92), its perfect cross-validation AUC of 1 strongly suggests overfitting, particularly given the small dataset size. The RPART model, by contrast, is based on just two variables (V2 and V6) and has a simple 5-node structure, offering transparency and interpretability, whereas Random Forest's ensemble nature makes it more complex and less interpretable, which can be a disadvantage in contexts requiring explainability. In this case, the fine-tuned RPART model appears more reliable than Random Forest. Nonetheless, let's explore VSURF feature selection technique, to potentially improve Random Forest's performance and reduce overfitting risks.

4.2 Random forest with VSURF

```
##
## === Cross-Validation Results ===

## Mean AUC: 1

## Mean Accuracy: 0.9179

## Mean Sensitivity (Recall): 0.9167

## Mean Specificity: 0.9417

## Thresholding step
## Estimated computational time (on one core): 1.8 sec.
##
## Interpretation step (on 19 variables)
## Estimated computational time (on one core): between 0 sec. and  0 sec.
##
## Prediction step (on 3 variables)
## Maximum estimated computational time (on one core): 0 sec.
##   |                                                                              |

##
## === Confusion Matrix ===

##          Actual
## Predicted Class0 Class1
##    Class0      8      1
##    Class1      2      5

##
## Accuracy: 0.8125

##
## Sensitivity (Recall): 0.8333

##
## Specificity: 0.8

##
## Test AUC: 0.95
```



The Random Forest model with feature selection using **VSURF** shows strong performance, achieving a test AUC of 0.95, which is slightly higher than the previous Random Forest model (AUC = 0.9333). However, the cross-validation metrics remain suspiciously perfect (AUC = 1, accuracy = 0.9179), suggesting that overfitting is still a concern, likely due to the small dataset size and limited test set. The VSURF process reduced the number of variables to 3 for the final prediction step, improving interpretability compared to the original Random Forest model while maintaining competitive performance. While the model demonstrates better generalization than before, its inflated cross-validation results indicate that further validation is necessary to confirm its reliability.

5. Conclusion

We have chosen three candidate models - RIDGE, Fine-tuned RPART, Random forest with VSURF — to identify the best-performing model for the dataset. The fine-tuned RPART model stands out as the most reliable choice despite its slightly lower test AUC (0.9167), as it avoids the overfitting risks seen in Ridge's reliance on all features and Random Forest's perfect cross-validation metrics. Its simplicity, interpretability, and focus on just two key variables (V2 and V6) make it more practical for real-world applications. Notably, all three models consistently identify V2, V3, and V6 as the most important features, highlighting their critical role in capturing underlying patterns. This agreement across different modeling approaches reinforces the relevance of these variables and suggests they should be prioritized in future analyses or model development.