Predicting Alzheimer's Disease Risk Based on Demographic, Lifestyle, and Health Determinants

Dingshuo Li 1007934844 Tianjin Duan 1008062383
Qiduo He 1008374037

December 8, 2024

Course:	STA314
Group Number:	86
Dataset Chosen:	Classification of Alzheimer's Disease
Kaggle Team Name:	Group 86
Kaggle Ranking:	90
Kaggle Score:	0.91346

1 Introduction

Alzheimer's is a condition characterized by a progressive decline in memory, thinking, behaviour and the ability to perform daily activities Alzheimer's Association (2024). It accounts for 60-80% of dementia cases and mainly affecting individuals aged 65 and older Bird (2018). While it does not directly cause death, it greatly increases the likelihood of other complications which ultimately lead to the patient's death. As the disease progresses, it eventually leading to the death of brain cells and leaving the patient unable to care for themselves. Among the early symptoms, memory loss and amnesia often go unnoticed Alzheimer's Association (2024). Currently, there is no cure for Alzheimer's disease and early symptoms are often not apparent, leading to missed opportunities for timely intervention Alzheimer's Association (2024). As the global population ages, Alzheimer's disease has become a serious public health challenge that not only affects individuals and families, but also puts pressure on health care systems worldwide.

1.1 Problem Statement

Early diagnosis is crucial as it allows for interventions that can slow the progression of the disease and improve the quality of patient's life. This purpose of this project addresses the specific research question: What demographic, lifestyle, and health characteristics are most strongly related to the risk of leading Alzheimer's Disease? How can we develop a reliable predictive model to assist in its early diagnosis? By analyzing large amounts of Alzheimer's data and utilizing machine learning techniques, our research aims to identify key risk factors and build a predictive model to help healthcare professionals detect Alzheimer's disease early. Ultimately, the project aims to bridge the gap between early detection and effective management, highlighting the potential of approaches to address pressing public health challenges.

2 Data

The dataset we used was sourced from Kaggle (2024). The data was divided into two groups approximately 70% of which 1504 rows were used for training and the rest 30% were used for testing. This way we ensure sufficient data for our model development and evaluation. The numerical variable contains mixed variables such as Age, BMI, Alcohol Consumption, Physical Activity, Diet Quality, and Sleep Quality. Categorical variables presented as 1 for yes and 0 for no, it included Gender, Smoking, Memory Complaints Family History of Alzheimers and Diagnosis etc.

2.1 EDA

In order to better understand the dataset, we performed an exploratory data analysis at the very beginning. First, all the missing values were checked, then we checked the structure and distribution of the dataset. Summary statistics revealed differences between key numerical variables, such as lower MMSE and being strongly associated with an Alzheimer's diagnosis. Correlation analysis in Figure 1 further revealed relationships between variables. The detailed EDA is in Appendix- A.

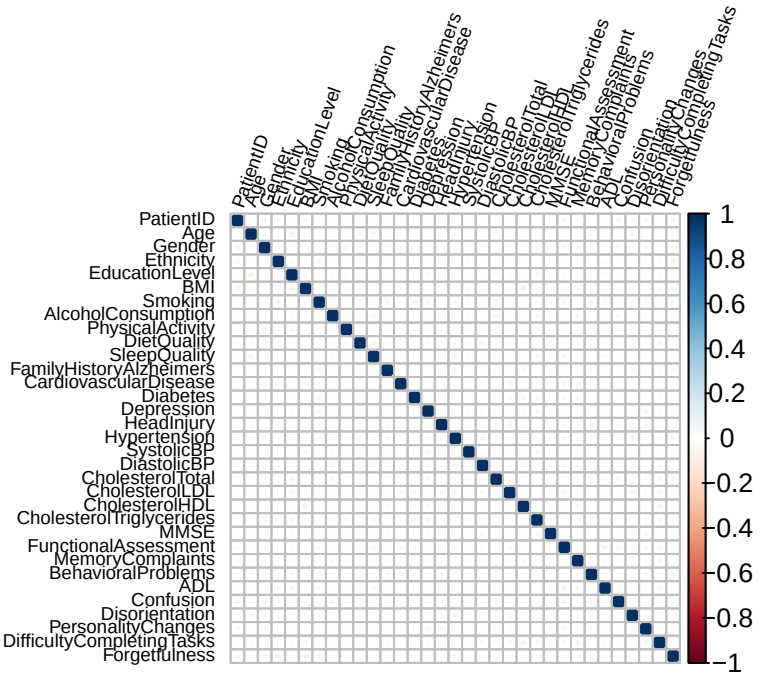


Figure 1: Correlation matrix showing pairwise relationships between variables, with blue indicating positive and red indicating negative correlations.

3 Parametric Modelling

Given the binary nature of the response variable Diagnosis no Alzheimer's disease = 0 and Alzheimer's disease = 1, we assumed logistic regression as the starting model. Logistic regression is a well-suited and interpretable method for binary classification problems. The initial model includes all potential predictors identified through exploratory data analysis, which helps to provide insight into the relationships and distributions of key variables.

To refine the set of predictors, we used a stepwise selection method. This method systematically adds or removes variables to optimize the model complexity. The resulting model 1 Appendix-

Table 1: Comparison of Models Based on AIC and BIC and Mean ROC-AUC

Model	AIC	BIC
Model1_Full	1623.017	1665.544
Model2_Reduced	1624.685	1651.264
Model3_LASSO	1138.041	1196.516
Model4_Ridge	1174.125	1349.549

Model	Mean_ROC
Model1_Full	0.7468952
Model2_Reduced	0.7620780
Model3_LASSO	0.8999466
Model4_Ridge	0.8950991

B retains key predictors that are considered to be significant predictors of Alzheimer's disease diagnosis. Based on Model 1, we selected significant hypothesis testing and p-values to further refine the model. Resulting in a reduced logistic regression model Appendix- F. Furthermore, we applied Lasso and Ridge regression regularization techniques widely used in machine learning. In addition, Cross-validation is used to determine the optimal regularization parameters.

Lasso model in Appendix- C penalizes the absolute size of coefficients, shrinking irrelevant ones to zero and producing a sparse model. This produces a sparse model that retains only the most influential predictors, improves interpretability, and focuses on variables that have a greater impact on the outcome. Moreover, Ridge regression modelling Appendix- D penalizes the squared magnitude of the coefficients, thereby reducing multicollinearity and stabilizing the coefficients without shrinking them to zero.

3.1 Compare Parametric Models

From Table 1, the Lasso model has the lowest AIC and BIC values and shows that the Lasso model has the highest average ROC-AUC of about 0.90. Thus, the Lasso logistic regression model identified key predictors for Alzheimer's diagnosis, they are Smoking, Sleep Quality, Cardiovascular Disease, Hypertension, Cholesterol Triglycerides, MMSE, Functional Assessment, Memory Complaints, Behavioral Problems, and ADL. In addition, LASSO is also a better choice here because it is consistent with the sparsity we observed in the data. With this model, we have identified the key variables, which marks the completion of our parametric approach to modeling.

3.2 Limitation of Parametric Modelling

After identifying the key variables, we tested the assumption of log-odds linearity required by logistic regression details in Appendix- I. By analyzing the residual plot Figure 2, we found strong evidence of nonlinearity in the data, which violates assumption of linear model. We attempted to improve the Lasso model by adding interaction and transformation terms but did not improve linearity as shown by Figure 3. This suggests that the logistic regression model is not suitable for our prediction.

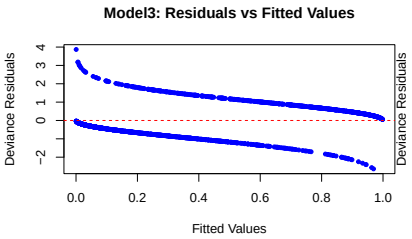


Figure 2: Check for Linearity

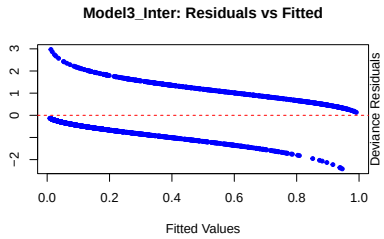


Figure 3: Check for Linearity

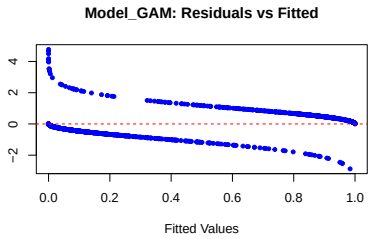


Figure 4: Check for Linearity

The Generalized Additive Model were also tested, details in Appendix- J. Although it achieved an adjusted R-square of 0.725 and deviance explained 59.7%, when we check the linearity shown in Figure 4, the residual plot shows the assumption of linearity does not hold.

4 Non-parametric Modelling

4.1 Decision Tree Modeling

A decision tree was constructed using the predictors identified by the Lasso model. The goal was to create a model that predicts an Alzheimer's diagnosis by decision rules. The Appendix- K shows the details of the modelling. We chose a low complexity parameter cp = 0.001 in order to allow the tree to go deeper and capture more subtle patterns in the data. As the result of cross-validation shows, the mean ROC-AUC for the best cp = 0.0902 was 0.844, with a mean sensitivity of 0.986 and a mean specificity of 0.611. The complexity parameter balances tree complexity and predictive accuracy, ensuring that the model is neither overfitted nor oversimplified.

4.2 Pruned Tree

Based on the cross-validation results, we selected the optimal complexity parameter cp = 0.10 to construct a pruned decision tree as Figure 5. This way we can improve the interpretability of the original decision tree.

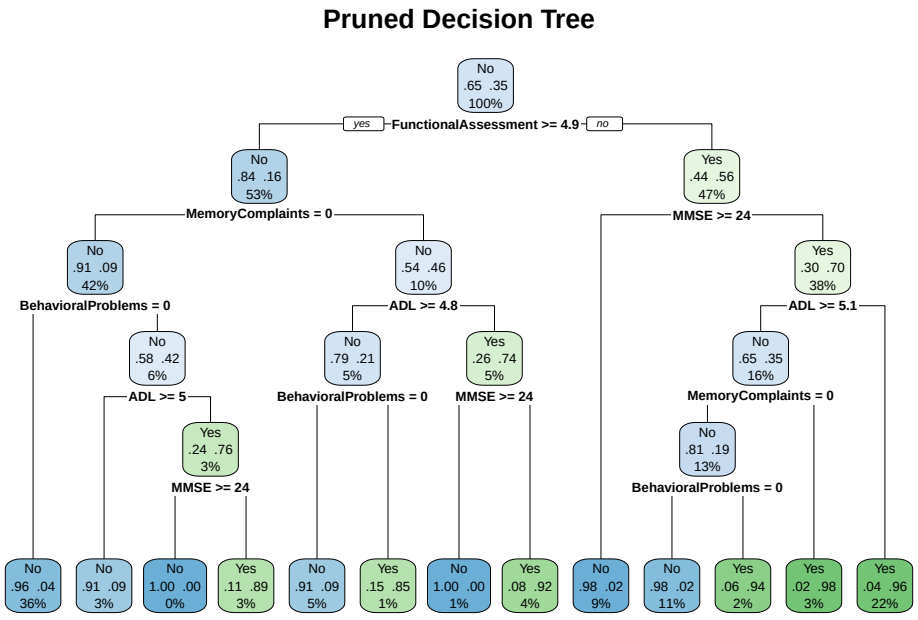


Figure 5: Pruned Decision Tree of Top Predictors in Alzheimer's Diagnosis

The Pruned Decision Tree shown FunctionalAssessment as the most important predictor, as well as other key variables such as MMSE, ADL, MemoryComplaints, and BehavioralProblems. The root differentiation was based on functional assessment >= 4.9, demonstrating its importance in identifying of Alzheimer's disease. To ensure robustness, 10 cross-validations were performed to optimize the cp. The details are in Appendix- M. As the result shows, pruned trees showed better accuracy with an average ROC-AUC of 0.958, achieved higher specificity mean: 0.92, and sensitivity remained high with mean: 0.96.

4.3 Random Forest

The predictors for the initial random forest model were selected based on the predictor in the Lasso logistic regression model which include Smoking, Sleep Quality, Cardiovascular Disease, Hypertension, Cholesterol Triglycerides, MMSE, Functional Assessment, Memory Complaints, Behavioral Problems, and ADL. After analyzing the variable importance scores, second models focused on refining the set of predictors to improve performance. The second random forest model reduced the predictors to the top five most important variables which are Functional Assessment, ADL, MMSE, Memory Complaints and Behavioral Problems. These steps were designed to improve interpretability with maintaining accuracy. The details of modelling are in Appendix- L. Moreover, we tuned the key parameters, including increasing the number of trees to 1,000 and reducing the minimum node size to 3. However, these adjustments did not result in significant improvements, and the OOB error rate remained same with insignificant changes in performance metrics.

4.4 Bagging

The bagged model was implemented using 500 bootstrap iterations and the result showed accuracy, sensitivity and specificity 100% on the training set with details in Appendix- O. While this indicates that the bagged model fits the training data very well, it could also indicate overfitting due to the absence of errors. So we implemented 10-fold cross validation to evaluate the bagging model. The average ROC for the cross validation was 0.9495, demonstrating its predictive power.

4.5 Tuned Random Forest

The tuned random forest model was used to optimize the performance of the original random forest by testing different hyperparameter values. Specifically, the number of variables considered in each split was adjusted using mtry = 2, 3, 4, 5 and evaluated using 10-fold cross-validation. The results showed that the mtry = 2 model had the highest average ROC-AUC value of 0.9565, indicating that this is the best model prediction for diagnosis of Alzheimer's disease.

4.5.1 Compare Non-Parametric Models

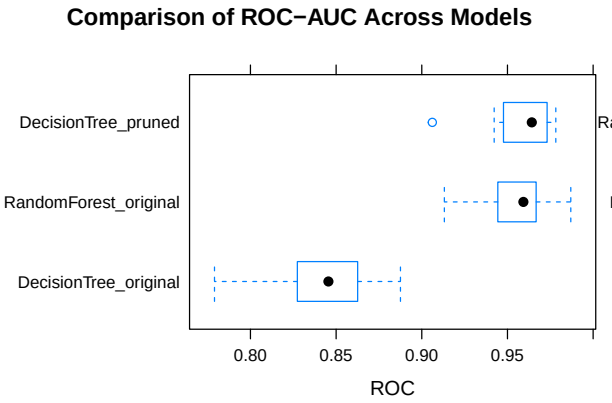


Figure 6: Comparison of Mean ROC-AUC Across Models

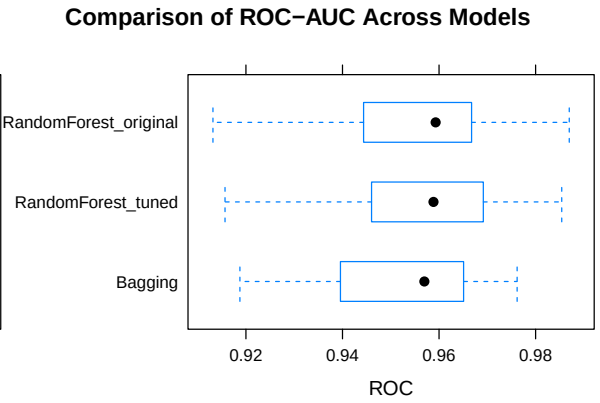


Figure 7: Comparison of Mean ROC-AUC Across Models

The box plot Figure 6 shows that the pruned decision tree achieves very high average ROC-AUC values, but similar to random forests. This makes it well-suited for situations where interpretability is critical. However, random forests perform better in terms of predictive stability, showing less variance in cross-validation folds, making it a better choice when maximizing predictive accuracy is a priority. In addition, using cross-validation on random forests

showed that a classification threshold of 0.5 yielded the best performance. With an average accuracy of 98.54%, an average precision of 99.61%, and an average recall of 96.3%, the random forest model achieved robust and accurate predictions for Alzheimer's disease cases.

Tuned random forests slightly improved compare to original random forests in terms of ROC-AUC, sensitivity, and specificity. As Figure 7 shows, the performance of the tuned random forest makes it an ideal model for this situation. Therefore, the final optimal model for this research question is the Tuned Random Forest model with key predictor variables: Functional Assessment, ADL, MMSE, Memory Complaints, and Behavioral Problem.

5 Results and Conclusion

This study investigated the demographic, lifestyle and health characteristics most closely associated with Alzheimer's disease and developed predictive models to aid early diagnosis. Through exploratory data analysis, parametric modelling, and non-parametric modelling, we have identified key variables including Functional Assessment, ADL, MMSE, Memory Complaints and Behavioral Problems, the most critical predictors of Alzheimer's disease. The analysis showed that Functional Assessment is the most important determinant of Alzheimer's disease, reflecting its strong correlation with patients' cognitive and functional abilities. These factors provide actionable insights for identifying people at risk for Alzheimer's, emphasizing their importance in clinical assessment and early intervention.

To develop a reliable predictive tool for early Alzheimer's diagnosis, we evaluated multiple models. The Tuned Random Forest is the most robust and accurate model with an average ROC-AUC of 0.9565. By adjusting hyperparameters such as the number of variables considered in each critical splits, the number of trees and other hyperparameters, the predictive stability and accuracy of the model were optimized. The random forest effectively captured the complex interactions and nonlinear relationships between predictors and response variables. These properties make the tuned random forest ideal for prioritizing predictive power and consistency.

On the other hand, the average ROC-AUC of the pruned decision tree was 0.958, which is comparable to that of a random forest, but more advantageous in terms of interpretability. The model has a hierarchical structure that emphasizes the relative importance of predictors, starting with functional assessment, followed by MMSE, ADL, memory complaints, and behavioural problems. The pruning technique reduces the complexity of the model by removing less critical splits, thereby increasing the generalization of the model and avoiding overfitting. Although slightly less stable than random forests, pruning trees provide a transparent framework for decision-making, which makes it particularly valuable in clinical applications where interpretation ability is critical.

This study shows that machine learning techniques, particularly ensemble models like Random Forest, can be effective in identifying key risk factors to support healthcare professionals in

the early diagnosis of Alzheimer's disease. Understanding the fact that functional assessment, ADL, MMSE, Memory Complaints, and Behavioural Problems are key factors in Alzheimer's disease can help develop more targeted early detection, intervention, and management strategies. Healthcare providers can prioritize these factors by incorporating them into daily screening, designing more predictive tools and improving diagnostic protocols. Customized intervention plans should focus on cognitive stimulation, behavioural management, and physical exercise, while educational activities and training can increase patient awareness and support. Integrating predictive algorithms into clinical workflows and emphasizing personalized care plans based on these factors can significantly improve early diagnosis, patient outcomes, and quality of life while reducing the burden on caregivers and the healthcare system.

6 Discussion

6.1 Limitation and Future Direction

This study has several limitations that affect its generalizability and applicability. The dataset used was limited in size and lacked diversity, particularly in terms of demographic and cultural background, which may limit the model to a wider population. Additionally, the lack of longitudinal temporality also limited our ability to analyze the progression of Alzheimer's disease over time, which may affect the temporal reliability of predictions. Finally, nonparametric models are less interpretable than parametric models, and this limitation in interpretability may prevent the model from achieving higher predictive accuracy.

Future research should focus on expanding the datasets, including different populations, and incorporating longitudinal data to improve the temporal accuracy of predictions. We need to actually validate our models in a real word setting as well as integrate these models into diagnostic workflows In addition, research into new predictive variables such as genetic markers or geographic data could further refine the model, which will provide a more reliable and comprehensive tool for early diagnosis of Alzheimer's disease and personalized patient care.

Appendix

A EDA

```
# Preview the data: First few and last few rows
head(data_train)
```

```
# A tibble: 6 x 35
  PatientID   Age Gender Ethnicity EducationLevel   BMI Smoking
      <dbl> <dbl>  <dbl>     <dbl>          <dbl> <dbl>   <dbl>
1         1    67      0         3              0  37.2       0
2         2    65      1         0              0  35.1       1
3         3    62      0         1              1  17.9       0
4         4    67      0         0              1  37.5       1
5         5    65      1         0              2  29.2       1
6         6    88      1         0              1  25.7       0
# i 28 more variables: AlcoholConsumption <dbl>, PhysicalActivity <dbl>,
#   DietQuality <dbl>, SleepQuality <dbl>, FamilyHistoryAlzheimers <dbl>,
#   CardiovascularDisease <dbl>, Diabetes <dbl>, Depression <dbl>,
#   HeadInjury <dbl>, Hypertension <dbl>, SystolicBP <dbl>, DiastolicBP <dbl>,
#   CholesterolTotal <dbl>, CholesterolLDL <dbl>, CholesterolHDL <dbl>,
#   CholesterolTriglycerides <dbl>, MMSE <dbl>, FunctionalAssessment <dbl>,
#   MemoryComplaints <dbl>, BehavioralProblems <dbl>, ADL <dbl>, ...
```

```
tail(data_train)
```

```
# A tibble: 6 x 35
  PatientID   Age Gender Ethnicity EducationLevel   BMI Smoking
      <dbl> <dbl>  <dbl>     <dbl>          <dbl> <dbl>   <dbl>
1      1499    75      1         2              1  31.7       0
2      1500    61      0         0              0  23.3       0
3      1501    78      1         3              2  34.3       0
4      1502    82      1         0              2  15.5       0
5      1503    87      1         2              0  22.7       0
6      1504    78      1         2              2  38.9       0
# i 28 more variables: AlcoholConsumption <dbl>, PhysicalActivity <dbl>,
#   DietQuality <dbl>, SleepQuality <dbl>, FamilyHistoryAlzheimers <dbl>,
#   CardiovascularDisease <dbl>, Diabetes <dbl>, Depression <dbl>,
#   HeadInjury <dbl>, Hypertension <dbl>, SystolicBP <dbl>, DiastolicBP <dbl>,
```

```
#   CholesterolTotal <dbl>, CholesterolLDL <dbl>, CholesterolHDL <dbl>,
#   CholesterolTriglycerides <dbl>, MMSE <dbl>, FunctionalAssessment <dbl>,
#   MemoryComplaints <dbl>, BehavioralProblems <dbl>, ADL <dbl>, ...
```

```
# Check the structure of the dataset
str(data_train)
```

```
spc_tbl_ [1,504 x 35] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ PatientID                : num [1:1504] 1 2 3 4 5 6 7 8 9 10 ...
 $ Age                      : num [1:1504] 67 65 62 67 65 88 83 72 90 61 ...
 $ Gender                   : num [1:1504] 0 1 0 0 1 1 1 1 1 1 ...
 $ Ethnicity                : num [1:1504] 3 0 1 0 0 0 1 1 3 0 ...
 $ EducationLevel           : num [1:1504] 0 0 1 1 2 1 3 1 2 0 ...
 $ BMI                      : num [1:1504] 37.2 35.1 17.9 37.5 29.2 ...
 $ Smoking                  : num [1:1504] 0 1 0 1 1 0 0 0 0 0 ...
 $ AlcoholConsumption       : num [1:1504] 12.216 17.111 13.526 19.952 0.533 ...
 $ PhysicalActivity         : num [1:1504] 7.78 6.65 9.59 1.95 8.76 ...
 $ DietQuality              : num [1:1504] 6.43 1.11 4.27 6.8 6.36 ...
 $ SleepQuality             : num [1:1504] 6.74 7.57 8.25 7.67 6.23 ...
 $ FamilyHistoryAlzheimers  : num [1:1504] 0 0 0 0 0 0 0 1 1 0 ...
 $ CardiovascularDisease    : num [1:1504] 0 0 0 1 1 0 0 0 0 0 ...
 $ Diabetes                 : num [1:1504] 0 0 1 1 0 0 0 0 0 0 ...
 $ Depression               : num [1:1504] 0 1 0 1 0 0 0 1 0 0 ...
 $ HeadInjury               : num [1:1504] 0 0 0 0 0 1 0 0 0 0 ...
 $ Hypertension             : num [1:1504] 0 0 0 0 0 0 0 0 0 0 ...
 $ SystolicBP               : num [1:1504] 137 111 131 121 158 126 165 117 115 126 ...
 $ DiastolicBP              : num [1:1504] 114 82 108 76 117 83 91 102 90 70 ...
 $ CholesterolTotal         : num [1:1504] 270 227 202 236 292 ...
 $ CholesterolLDL           : num [1:1504] 119 101 185 151 125 ...
 $ CholesterolHDL           : num [1:1504] 78 21.2 37 62.2 82.9 ...
 $ CholesterolTriglycerides : num [1:1504] 273 157 289 196 296 ...
 $ MMSE                     : num [1:1504] 0.695 23.79 6.592 25.343 6.628 ...
 $ FunctionalAssessment     : num [1:1504] 9.99 6.2 9.57 2.49 7.52 ...
 $ MemoryComplaints         : num [1:1504] 1 0 0 0 1 0 0 0 0 0 ...
 $ BehavioralProblems       : num [1:1504] 0 0 0 0 0 0 0 0 0 0 ...
 $ ADL                      : num [1:1504] 6.01 7.52 8.57 6.22 5.19 ...
 $ Confusion                : num [1:1504] 0 0 0 0 1 0 0 0 0 1 ...
 $ Disorientation           : num [1:1504] 0 0 0 0 0 1 0 0 1 0 ...
 $ PersonalityChanges       : num [1:1504] 0 0 0 0 0 1 0 0 0 0 ...
 $ DifficultyCompletingTasks: num [1:1504] 1 0 0 0 0 0 0 0 0 1 ...
 $ Forgetfulness            : num [1:1504] 1 1 0 1 1 0 0 0 0 1 ...
 $ Diagnosis                : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
$ DoctorInCharge           : chr [1:1504] "XXXConfid" "XXXConfid" "XXXConfid" "XXXConfid" .
- attr(*, "spec")=
 .. cols(
 ..   PatientID = col_double(),
 ..   Age = col_double(),
 ..   Gender = col_double(),
 ..   Ethnicity = col_double(),
 ..   EducationLevel = col_double(),
 ..   BMI = col_double(),
 ..   Smoking = col_double(),
 ..   AlcoholConsumption = col_double(),
 ..   PhysicalActivity = col_double(),
 ..   DietQuality = col_double(),
 ..   SleepQuality = col_double(),
 ..   FamilyHistoryAlzheimers = col_double(),
 ..   CardiovascularDisease = col_double(),
 ..   Diabetes = col_double(),
 ..   Depression = col_double(),
 ..   HeadInjury = col_double(),
 ..   Hypertension = col_double(),
 ..   SystolicBP = col_double(),
 ..   DiastolicBP = col_double(),
 ..   CholesterolTotal = col_double(),
 ..   CholesterolLDL = col_double(),
 ..   CholesterolHDL = col_double(),
 ..   CholesterolTriglycerides = col_double(),
 ..   MMSE = col_double(),
 ..   FunctionalAssessment = col_double(),
 ..   MemoryComplaints = col_double(),
 ..   BehavioralProblems = col_double(),
 ..   ADL = col_double(),
 ..   Confusion = col_double(),
 ..   Disorientation = col_double(),
 ..   PersonalityChanges = col_double(),
 ..   DifficultyCompletingTasks = col_double(),
 ..   Forgetfulness = col_double(),
 ..   Diagnosis = col_double(),
 ..   DoctorInCharge = col_character()
 .. )
- attr(*, "problems")=<externalptr>
```

```r
# Summary statistics of the dataset
summary(data_train)
```

```
   PatientID           Age            Gender         Ethnicity
 Min.   :   1.0   Min.   :60.00   Min.   :0.0000   Min.   :0.0000
 1st Qu.: 376.8   1st Qu.:67.00   1st Qu.:0.0000   1st Qu.:0.0000
 Median : 752.5   Median :75.00   Median :1.0000   Median :0.0000
 Mean   : 752.5   Mean   :74.91   Mean   :0.5086   Mean   :0.7114
 3rd Qu.:1128.2   3rd Qu.:83.00   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1504.0   Max.   :90.00   Max.   :1.0000   Max.   :3.0000
 EducationLevel       BMI            Smoking       AlcoholConsumption
 Min.   :0.000   Min.   :15.01   Min.   :0.0000   Min.   : 0.002003
 1st Qu.:1.000   1st Qu.:21.37   1st Qu.:0.0000   1st Qu.: 5.204286
 Median :1.000   Median :27.76   Median :0.0000   Median : 9.924320
 Mean   :1.296   Mean   :27.55   Mean   :0.2839   Mean   :10.030205
 3rd Qu.:2.000   3rd Qu.:33.78   3rd Qu.:1.0000   3rd Qu.:15.140505
 Max.   :3.000   Max.   :39.93   Max.   :1.0000   Max.   :19.988291
 PhysicalActivity    DietQuality        SleepQuality    FamilyHistoryAlzheimers
 Min.   :0.003616   Min.   :0.009385   Min.   : 4.003   Min.   :0.0000
 1st Qu.:2.538671   1st Qu.:2.302514   1st Qu.: 5.480   1st Qu.:0.0000
 Median :4.790574   Median :4.979274   Median : 7.100   Median :0.0000
 Mean   :4.914426   Mean   :4.937305   Mean   : 7.042   Mean   :0.2447
 3rd Qu.:7.452197   3rd Qu.:7.576618   3rd Qu.: 8.550   3rd Qu.:0.0000
 Max.   :9.987429   Max.   :9.998346   Max.   :10.000   Max.   :1.0000
 CardiovascularDisease    Diabetes        Depression       HeadInjury
 Min.   :0.0000        Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.0000        1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
 Median :0.0000        Median :0.0000   Median :0.0000   Median :0.00000
 Mean   :0.1343        Mean   :0.1596   Mean   :0.2081   Mean   :0.09508
 3rd Qu.:0.0000        3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
 Max.   :1.0000        Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
  Hypertension      SystolicBP     DiastolicBP     CholesterolTotal
 Min.   :0.0000   Min.   : 90.0   Min.   : 60.00   Min.   :150.1
 1st Qu.:0.0000   1st Qu.:112.0   1st Qu.: 74.00   1st Qu.:190.5
 Median :0.0000   Median :135.0   Median : 90.00   Median :224.4
 Mean   :0.1516   Mean   :134.7   Mean   : 89.71   Mean   :225.2
 3rd Qu.:0.0000   3rd Qu.:156.0   3rd Qu.:105.00   3rd Qu.:262.5
 Max.   :1.0000   Max.   :179.0   Max.   :119.00   Max.   :300.0
 CholesterolLDL   CholesterolHDL  CholesterolTriglycerides      MMSE
 Min.   : 50.40   Min.   :20.00   Min.   : 50.41           Min.   : 0.0353
 1st Qu.: 87.52   1st Qu.:39.15   1st Qu.:136.31           1st Qu.: 7.1155
 Median :124.52   Median :59.59   Median :229.55           Median :14.3225
```

```
 Mean   :124.88   Mean   :59.51   Mean   :226.90           Mean   :14.6491
 3rd Qu.:161.96   3rd Qu.:78.91   3rd Qu.:313.06           3rd Qu.:21.8386
 Max.   :199.97   Max.   :99.98   Max.   :399.94           Max.   :29.9914
 FunctionalAssessment MemoryComplaints BehavioralProblems      ADL
 Min.   :0.00046      Min.   :0.0000   Min.   :0.0000     Min.   :0.004354
 1st Qu.:2.65883      1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:2.358590
 Median :5.19113      Median :0.0000   Median :0.0000     Median :4.877862
 Mean   :5.13989      Mean   :0.2055   Mean   :0.1516     Mean   :4.903536
 3rd Qu.:7.61636      3rd Qu.:0.0000   3rd Qu.:0.0000     3rd Qu.:7.517219
 Max.   :9.99647      Max.   :1.0000   Max.   :1.0000     Max.   :9.972663
   Confusion      Disorientation   PersonalityChanges DifficultyCompletingTasks
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000     Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:0.0000
 Median :0.0000   Median :0.0000   Median :0.0000     Median :0.0000
 Mean   :0.2028   Mean   :0.1562   Mean   :0.1569     Mean   :0.1622
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000     3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000     Max.   :1.0000
 Forgetfulness    Diagnosis DoctorInCharge
 Min.   :0.0000   No :972   Length:1504
 1st Qu.:0.0000   Yes:532   Class :character
 Median :0.0000             Mode  :character
 Mean   :0.2999
 3rd Qu.:1.0000
 Max.   :1.0000
```

```r
# Dimensions of the dataset (rows and columns)
dim(data_train)
```

```
[1] 1504   35
```

A.1 Statistical Summary for Numeric Columns

```r
# Extract summary statistics for numeric columns
numeric_summary <- summary(data_train)

# Convert to a data frame for better presentation
numeric_summary_df <- as.data.frame(as.table(numeric_summary))

# View the result as a table
numeric_summary_df
```

	Var1	Var2	Freq
1		PatientID	Min. : 1.0
2		PatientID	1st Qu.: 376.8
3		PatientID	Median : 752.5
4		PatientID	Mean : 752.5
5		PatientID	3rd Qu.:1128.2
6		PatientID	Max. :1504.0
7		Age	Min. :60.00
8		Age	1st Qu.:67.00
9		Age	Median :75.00
10		Age	Mean :74.91
11		Age	3rd Qu.:83.00
12		Age	Max. :90.00
13		Gender	Min. :0.0000
14		Gender	1st Qu.:0.0000
15		Gender	Median :1.0000
16		Gender	Mean :0.5086
17		Gender	3rd Qu.:1.0000
18		Gender	Max. :1.0000
19		Ethnicity	Min. :0.0000
20		Ethnicity	1st Qu.:0.0000
21		Ethnicity	Median :0.0000
22		Ethnicity	Mean :0.7114
23		Ethnicity	3rd Qu.:1.0000
24		Ethnicity	Max. :3.0000
25		EducationLevel	Min. :0.000
26		EducationLevel	1st Qu.:1.000
27		EducationLevel	Median :1.000
28		EducationLevel	Mean :1.296
29		EducationLevel	3rd Qu.:2.000
30		EducationLevel	Max. :3.000
31		BMI	Min. :15.01
32		BMI	1st Qu.:21.37
33		BMI	Median :27.76
34		BMI	Mean :27.55
35		BMI	3rd Qu.:33.78
36		BMI	Max. :39.93
37		Smoking	Min. :0.0000
38		Smoking	1st Qu.:0.0000
39		Smoking	Median :0.0000
40		Smoking	Mean :0.2839
41		Smoking	3rd Qu.:1.0000
42		Smoking	Max. :1.0000

```
            AlcoholConsumption Min.   : 0.002003
            AlcoholConsumption 1st Qu.: 5.204286
            AlcoholConsumption Median : 9.924320
            AlcoholConsumption Mean   :10.030205
            AlcoholConsumption 3rd Qu.:15.140505
            AlcoholConsumption Max.   :19.988291
              PhysicalActivity  Min.   :0.003616
              PhysicalActivity  1st Qu.:2.538671
              PhysicalActivity  Median :4.790574
              PhysicalActivity  Mean   :4.914426
              PhysicalActivity  3rd Qu.:7.452197
              PhysicalActivity  Max.   :9.987429
                   DietQuality  Min.   :0.009385
                   DietQuality  1st Qu.:2.302514
                   DietQuality  Median :4.979274
                   DietQuality  Mean   :4.937305
                   DietQuality  3rd Qu.:7.576618
                   DietQuality  Max.   :9.998346
                  SleepQuality    Min.   : 4.003
                  SleepQuality    1st Qu.: 5.480
                  SleepQuality    Median : 7.100
                  SleepQuality    Mean   : 7.042
                  SleepQuality    3rd Qu.: 8.550
                  SleepQuality    Max.   :10.000
        FamilyHistoryAlzheimers    Min.   :0.0000
        FamilyHistoryAlzheimers    1st Qu.:0.0000
        FamilyHistoryAlzheimers    Median :0.0000
        FamilyHistoryAlzheimers    Mean   :0.2447
        FamilyHistoryAlzheimers    3rd Qu.:0.0000
        FamilyHistoryAlzheimers    Max.   :1.0000
          CardiovascularDisease    Min.   :0.0000
          CardiovascularDisease    1st Qu.:0.0000
          CardiovascularDisease    Median :0.0000
          CardiovascularDisease    Mean   :0.1343
          CardiovascularDisease    3rd Qu.:0.0000
          CardiovascularDisease    Max.   :1.0000
                       Diabetes    Min.   :0.0000
                       Diabetes    1st Qu.:0.0000
                       Diabetes    Median :0.0000
                       Diabetes    Mean   :0.1596
                       Diabetes    3rd Qu.:0.0000
                       Diabetes    Max.   :1.0000
                     Depression    Min.   :0.0000
```

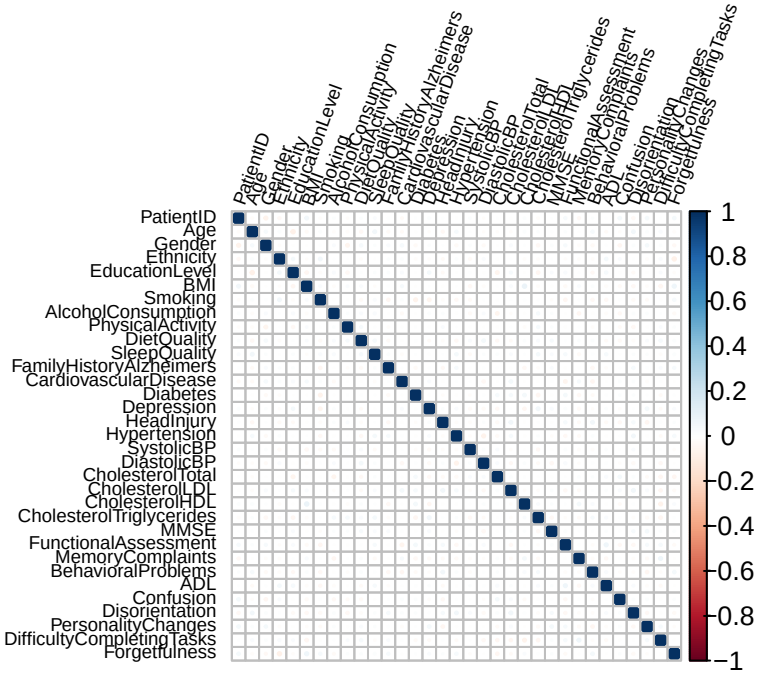
```
86                    Depression    1st Qu.:0.0000
87                    Depression    Median :0.0000
88                    Depression    Mean   :0.2081
89                    Depression    3rd Qu.:0.0000
90                    Depression    Max.   :1.0000
91                    HeadInjury   Min.   :0.00000
92                    HeadInjury   1st Qu.:0.00000
93                    HeadInjury   Median :0.00000
94                    HeadInjury   Mean   :0.09508
95                    HeadInjury   3rd Qu.:0.00000
96                    HeadInjury   Max.   :1.00000
97                  Hypertension    Min.   :0.0000
98                  Hypertension    1st Qu.:0.0000
99                  Hypertension    Median :0.0000
100                 Hypertension    Mean   :0.1516
101                 Hypertension    3rd Qu.:0.0000
102                 Hypertension    Max.   :1.0000
103                   SystolicBP     Min.   : 90.0
104                   SystolicBP     1st Qu.:112.0
105                   SystolicBP     Median :135.0
106                   SystolicBP     Mean   :134.7
107                   SystolicBP     3rd Qu.:156.0
108                   SystolicBP     Max.   :179.0
109                  DiastolicBP    Min.   : 60.00
110                  DiastolicBP    1st Qu.: 74.00
111                  DiastolicBP    Median : 90.00
112                  DiastolicBP    Mean   : 89.71
113                  DiastolicBP    3rd Qu.:105.00
114                  DiastolicBP    Max.   :119.00
115              CholesterolTotal     Min.   :150.1
116              CholesterolTotal     1st Qu.:190.5
117              CholesterolTotal     Median :224.4
118              CholesterolTotal     Mean   :225.2
119              CholesterolTotal     3rd Qu.:262.5
120              CholesterolTotal     Max.   :300.0
121                CholesterolLDL    Min.   : 50.40
122                CholesterolLDL    1st Qu.: 87.52
123                CholesterolLDL    Median :124.52
124                CholesterolLDL    Mean   :124.88
125                CholesterolLDL    3rd Qu.:161.96
126                CholesterolLDL    Max.   :199.97
127                CholesterolHDL     Min.   :20.00
128                CholesterolHDL     1st Qu.:39.15
```

```
129                CholesterolHDL     Median :59.59
130                CholesterolHDL     Mean   :59.51
131                CholesterolHDL     3rd Qu.:78.91
132                CholesterolHDL     Max.   :99.98
133       CholesterolTriglycerides    Min.   : 50.41
134       CholesterolTriglycerides    1st Qu.:136.31
135       CholesterolTriglycerides    Median :229.55
136       CholesterolTriglycerides    Mean   :226.90
137       CholesterolTriglycerides    3rd Qu.:313.06
138       CholesterolTriglycerides    Max.   :399.94
139                           MMSE   Min.   : 0.0353
140                           MMSE   1st Qu.: 7.1155
141                           MMSE   Median :14.3225
142                           MMSE   Mean   :14.6491
143                           MMSE   3rd Qu.:21.8386
144                           MMSE   Max.   :29.9914
145           FunctionalAssessment   Min.   :0.00046
146           FunctionalAssessment   1st Qu.:2.65883
147           FunctionalAssessment   Median :5.19113
148           FunctionalAssessment   Mean   :5.13989
149           FunctionalAssessment   3rd Qu.:7.61636
150           FunctionalAssessment   Max.   :9.99647
151               MemoryComplaints    Min.   :0.0000
152               MemoryComplaints    1st Qu.:0.0000
153               MemoryComplaints    Median :0.0000
154               MemoryComplaints    Mean   :0.2055
155               MemoryComplaints    3rd Qu.:0.0000
156               MemoryComplaints    Max.   :1.0000
157             BehavioralProblems    Min.   :0.0000
158             BehavioralProblems    1st Qu.:0.0000
159             BehavioralProblems    Median :0.0000
160             BehavioralProblems    Mean   :0.1516
161             BehavioralProblems    3rd Qu.:0.0000
162             BehavioralProblems    Max.   :1.0000
163                            ADL  Min.   :0.004354
164                            ADL  1st Qu.:2.358590
165                            ADL  Median :4.877862
166                            ADL  Mean   :4.903536
167                            ADL  3rd Qu.:7.517219
168                            ADL  Max.   :9.972663
169                      Confusion    Min.   :0.0000
170                      Confusion    1st Qu.:0.0000
171                      Confusion    Median :0.0000
```

```
                    Confusion    Mean   :0.2028
                    Confusion    3rd Qu.:0.0000
                    Confusion    Max.   :1.0000
                Disorientation   Min.   :0.0000
                Disorientation   1st Qu.:0.0000
                Disorientation   Median :0.0000
                Disorientation   Mean   :0.1562
                Disorientation   3rd Qu.:0.0000
                Disorientation   Max.   :1.0000
            PersonalityChanges   Min.   :0.0000
            PersonalityChanges   1st Qu.:0.0000
            PersonalityChanges   Median :0.0000
            PersonalityChanges   Mean   :0.1569
            PersonalityChanges   3rd Qu.:0.0000
            PersonalityChanges   Max.   :1.0000
     DifficultyCompletingTasks   Min.   :0.0000
     DifficultyCompletingTasks   1st Qu.:0.0000
     DifficultyCompletingTasks   Median :0.0000
     DifficultyCompletingTasks   Mean   :0.1622
     DifficultyCompletingTasks   3rd Qu.:0.0000
     DifficultyCompletingTasks   Max.   :1.0000
                Forgetfulness   Min.   :0.0000
                Forgetfulness   1st Qu.:0.0000
                Forgetfulness   Median :0.0000
                Forgetfulness   Mean   :0.2999
                Forgetfulness   3rd Qu.:1.0000
                Forgetfulness   Max.   :1.0000
                    Diagnosis          No :972
                    Diagnosis          Yes:532
                    Diagnosis               <NA>
                    Diagnosis               <NA>
                    Diagnosis               <NA>
                    Diagnosis               <NA>
                DoctorInCharge  Length:1504
                DoctorInCharge  Class :character
                DoctorInCharge  Mode  :character
                DoctorInCharge               <NA>
                DoctorInCharge               <NA>
                DoctorInCharge               <NA>
```

A.2 Correlation matrix for numeric columns

```
# install.packages("corrplot")
library(corrplot)
numeric_data <- data_train[, sapply(data_train, is.numeric)]
corr_matrix <- cor(numeric_data, use = "complete.obs")

corrplot(corr_matrix, method = "circle", tl.cex = 0.6, tl.srt = 70, tl.col = "black")
```




```
# considering about the univariable data visualization
```

B Stepwise Modelling

```
model <- glm(Diagnosis ~ Age + Gender + Ethnicity + EducationLevel + Smoking + AlcoholConsump
```

```
stepwise_model <- step(model,direction ="both",trace =0)
summary(stepwise_model)
```

```
Call:
glm(formula = Diagnosis ~ SleepQuality + Hypertension + Diabetes +
    CardiovascularDisease + MMSE + MemoryComplaints + BehavioralProblems,
    family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9855  -0.8436  -0.5184   0.8917   2.2947

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             0.282040   0.273893   1.030   0.3031
SleepQuality           -0.079899   0.034865  -2.292   0.0219 *
Hypertension            0.276982   0.167612   1.653   0.0984 .
Diabetes               -0.278396   0.170634  -1.632   0.1028
CardiovascularDisease   0.261178   0.177715   1.470   0.1417
MMSE                   -0.074324   0.007574  -9.813   <2e-16 ***
MemoryComplaints        1.760867   0.148717  11.840   <2e-16 ***
BehavioralProblems      1.649411   0.166288   9.919   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1607.0  on 1496  degrees of freedom
AIC: 1623

Number of Fisher Scoring iterations: 4
```

```r
# Stepwise model selection based on AIC
model1 <- glm(Diagnosis ~ SleepQuality + Hypertension + Diabetes +
    CardiovascularDisease + MMSE + MemoryComplaints + BehavioralProblems,
    family = binomial, data = data_train)

stepwise_model <- step(model, direction = "both", trace = 0)
summary(stepwise_model)
```

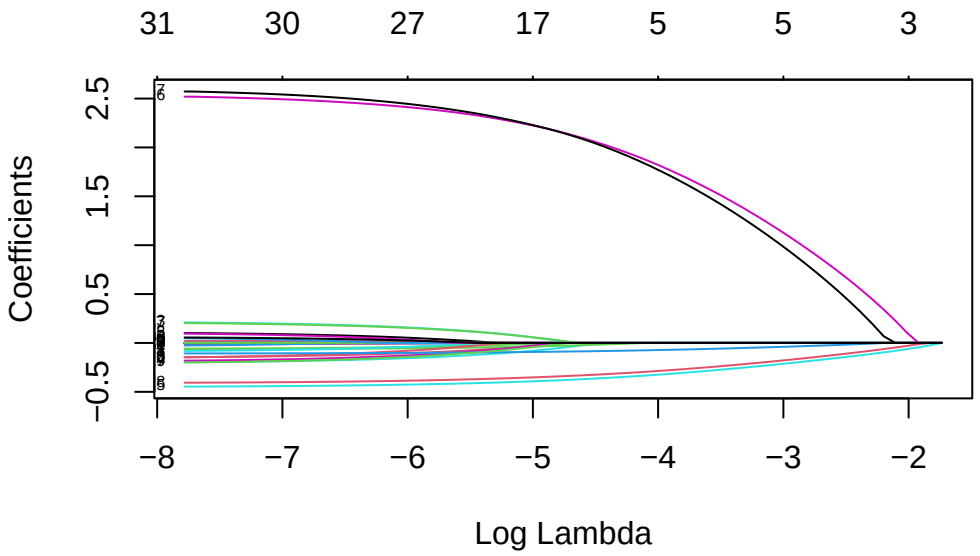
```
Call:
glm(formula = Diagnosis ~ SleepQuality + Hypertension + Diabetes +
```

```
    CardiovascularDisease + MMSE + MemoryComplaints + BehavioralProblems,
    family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9855  -0.8436  -0.5184   0.8917   2.2947

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)            0.282040   0.273893   1.030   0.3031
SleepQuality          -0.079899   0.034865  -2.292   0.0219 *
Hypertension           0.276982   0.167612   1.653   0.0984 .
Diabetes              -0.278396   0.170634  -1.632   0.1028
CardiovascularDisease  0.261178   0.177715   1.470   0.1417
MMSE                  -0.074324   0.007574  -9.813   <2e-16 ***
MemoryComplaints       1.760867   0.148717  11.840   <2e-16 ***
BehavioralProblems     1.649411   0.166288   9.919   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1607.0  on 1496  degrees of freedom
AIC: 1623

Number of Fisher Scoring iterations: 4
```

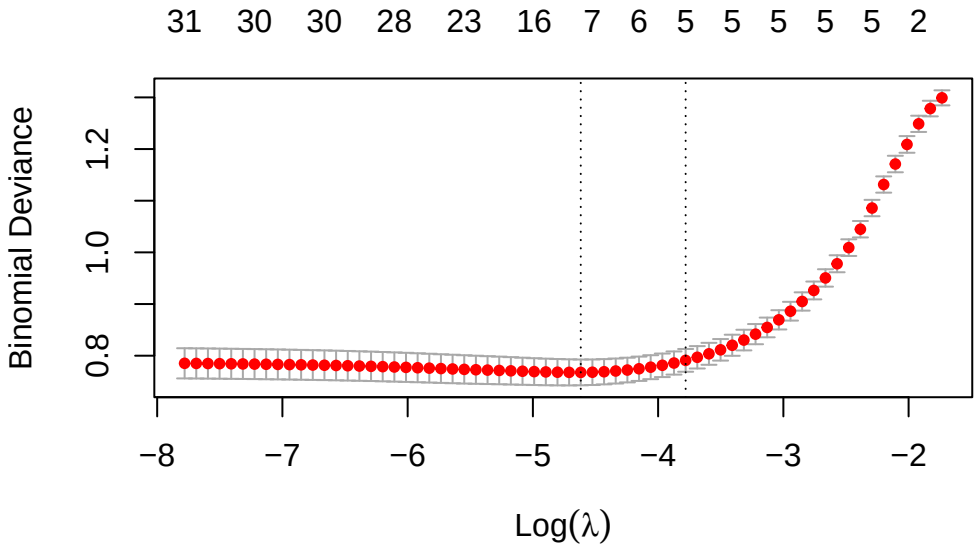
C Lasso

```r
# Lasso regression (L1 regularization)
library(glmnet)

# Prepare data for glmnet (predictors and response)
X <- data_train[, -which(names(data_train) == "Diagnosis")]
y <- data_train$Diagnosis

# Fit Lasso model (alpha = 1 for Lasso)
lasso_model <- glmnet(as.matrix(X), y, alpha = 1, family = "binomial")
```

```r
# Plot the Lasso path
plot(lasso_model, xvar = "lambda", label = TRUE)
```




```r
# Choose the best lambda via cross-validation
cv_lasso <- cv.glmnet(as.matrix(X), y, alpha = 1, family = "binomial")
plot(cv_lasso)
```



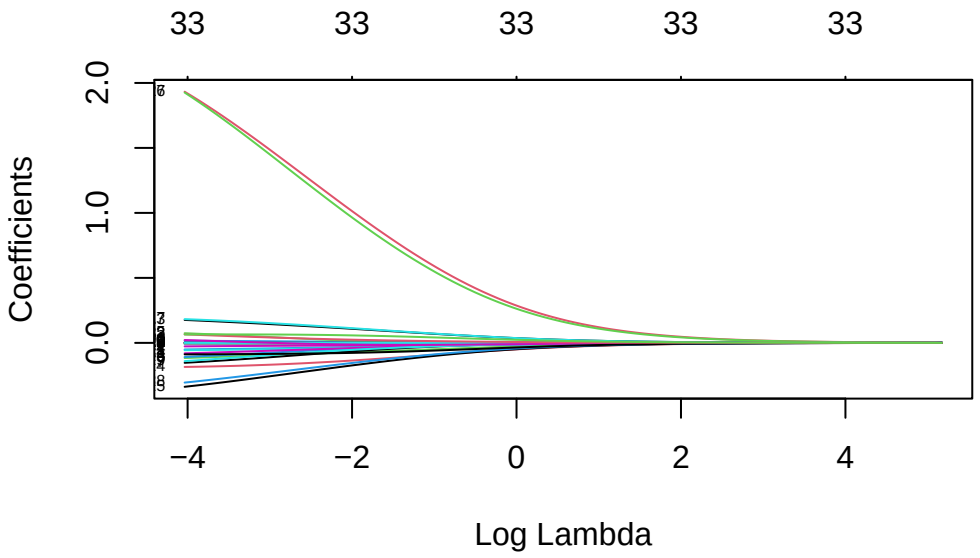

```r
# Fit the model with the best lambda
best_lambda <- cv_lasso$lambda.min
```

```r
lasso_best_model <- glmnet(as.matrix(X), y, alpha = 1, lambda = best_lambda, family = "binom

# Coefficients of the selected features
coef(lasso_best_model)
```

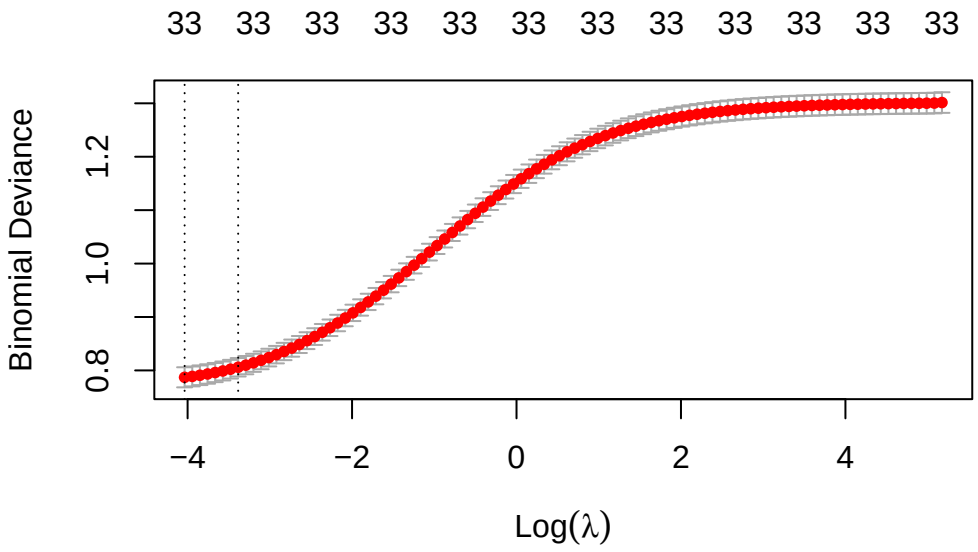
```
35 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)                3.22988880
PatientID                  .
Age                        .
Gender                     .
Ethnicity                  .
EducationLevel             .
BMI                        .
Smoking                   -0.02555237
AlcoholConsumption         .
PhysicalActivity           .
DietQuality                .
SleepQuality              -0.02312742
FamilyHistoryAlzheimers    .
CardiovascularDisease      .
Diabetes                   .
Depression                 .
HeadInjury                 .
Hypertension               .
SystolicBP                 .
DiastolicBP                .
CholesterolTotal           .
CholesterolLDL             .
CholesterolHDL             .
CholesterolTriglycerides   .
MMSE                      -0.08615812
FunctionalAssessment      -0.37182875
MemoryComplaints           2.10204326
BehavioralProblems         2.09257636
ADL                       -0.33205913
Confusion                  .
Disorientation             .
PersonalityChanges         .
DifficultyCompletingTasks  .
Forgetfulness              .
DoctorInCharge             .
```

D Ridge

```
# Ridge regression (L2 regularization)

# Fit Ridge model (alpha = 0 for Ridge)
ridge_model <- glmnet(as.matrix(X), y, alpha = 0, family = "binomial")

# Plot the Ridge path
plot(ridge_model, xvar = "lambda", label = TRUE)
```




```
# Choose the best lambda via cross-validation
cv_ridge <- cv.glmnet(as.matrix(X), y, alpha = 0, family = "binomial")
plot(cv_ridge)
```




```
# Fit the model with the best lambda
best_lambda_ridge <- cv_ridge$lambda.min
ridge_best_model <- glmnet(as.matrix(X), y, alpha = 0, lambda = best_lambda_ridge, family = '

# Coefficients of the selected features
coef(ridge_best_model)
```

```
35 x 1 sparse Matrix of class "dgCMatrix"
                                      s0
(Intercept)                3.040133e+00
PatientID                 -1.562046e-05
Age                       -4.558483e-03
Gender                    -1.164258e-01
Ethnicity                 -5.076628e-02
EducationLevel            -1.064207e-02
BMI                        3.291084e-03
Smoking                   -1.532306e-01
AlcoholConsumption        -5.267618e-03
PhysicalActivity          -1.628205e-03
DietQuality                1.637457e-02
SleepQuality              -5.666907e-02
FamilyHistoryAlzheimers   -2.709363e-02
CardiovascularDisease      1.764834e-01
Diabetes                  -1.851213e-01
Depression                 7.436718e-02
HeadInjury                -1.086466e-01
```

```
Hypertension                1.825768e-01
SystolicBP                  3.551449e-04
DiastolicBP                 1.947749e-03
CholesterolTotal            4.044755e-04
CholesterolLDL             -4.289550e-04
CholesterolHDL              2.129989e-03
CholesterolTriglycerides    5.537063e-04
MMSE                       -8.043204e-02
FunctionalAssessment       -3.380086e-01
MemoryComplaints            1.932831e+00
BehavioralProblems          1.926205e+00
ADL                        -3.055687e-01
Confusion                  -1.406385e-01
Disorientation              2.207733e-02
PersonalityChanges         -8.841200e-02
DifficultyCompletingTasks   6.517416e-02
Forgetfulness               6.505374e-02
DoctorInCharge              .
```

E Cross-Validation (k-Fold Cross-Validation)

E.1 VIF to detect multicollinearity

```
library(car)
vif(model)
```

```
                     Age                   Gender                Ethnicity
                1.030405                 1.025532                 1.023169
          EducationLevel                  Smoking        AlcoholConsumption
                1.031516                 1.028972                 1.014494
         PhysicalActivity              DietQuality             SleepQuality
                1.017358                 1.028633                 1.014870
                     BMI             Hypertension              SystolicBP
                1.025659                 1.024685                 1.009433
             DiastolicBP         CholesterolTotal            CholesterolLDL
                1.020109                 1.020453                 1.023274
           CholesterolHDL  CholesterolTriglycerides                 Diabetes
                1.022967                 1.027327                 1.022996
    CardiovascularDisease   FamilyHistoryAlzheimers              HeadInjury
                1.018353                 1.017582                 1.022446
```

```
          Depression                MMSE     MemoryComplaints
            1.022444            1.069412             1.071456
  BehavioralProblems           Confusion     Disorientation
            1.067370            1.016303             1.023509
  PersonalityChanges DifficultyCompletingTasks      Forgetfulness
            1.017601            1.029138             1.026895
```

Interpreting VIF Values: VIF < 5: Low multicollinearity, no concern. VIF between 5-10: Moderate multicollinearity, consider revising predictors. VIF > 10: High multicollinearity, action required.

F Model 2

F.1 Reduced Logistic Regression Model (Based on Hypothesis test, p-values)

```r
# Based on the significant p-values, pick the predictors to fit model2

# Logistic regression model with reduced set of features
model2 <- glm(Diagnosis ~ SleepQuality + MMSE + MemoryComplaints + BehavioralProblems,
              data = data_train, family = binomial)

# View the summary of the reduced model
summary(model2)
```

```
Call:
glm(formula = Diagnosis ~ SleepQuality + MMSE + MemoryComplaints +
    BehavioralProblems, family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8993  -0.8524  -0.5272   0.9000   2.2777

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.309373   0.269330   1.149   0.2507
SleepQuality       -0.079227   0.034721  -2.282   0.0225 *
MMSE               -0.074010   0.007541  -9.814   <2e-16 ***
MemoryComplaints    1.760806   0.147989  11.898   <2e-16 ***
```

```
BehavioralProblems  1.640637   0.165422   9.918   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1614.7  on 1499  degrees of freedom
AIC: 1624.7

Number of Fisher Scoring iterations: 4
```

G Model 3

G.1 Logistic Regression Model Based on Lasso

```r
# Load necessary libraries
library(tidyverse)
library(caret)
library(pROC)


# Fit Logistic Regression Model 3
model3 <- glm(Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +
                          Hypertension + CholesterolTriglycerides + MMSE +
                          FunctionalAssessment + MemoryComplaints +
                          BehavioralProblems + ADL, data = data_train, family = binomial)

# Summarize Model 3
summary(model3)
```

```
Call:
glm(formula = Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +
    Hypertension + CholesterolTriglycerides + MMSE + FunctionalAssessment +
    MemoryComplaints + BehavioralProblems + ADL, family = binomial,
    data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.6371  -0.5647  -0.2114   0.5068   3.8694

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                4.1189163  0.4419068   9.321   <2e-16 ***
Smoking                   -0.2173226  0.1685495  -1.289    0.197
SleepQuality              -0.0680511  0.0432353  -1.574    0.115
CardiovascularDisease      0.2084491  0.2119888   0.983    0.325
Hypertension               0.2067576  0.2095261   0.987    0.324
CholesterolTriglycerides   0.0007706  0.0007359   1.047    0.295
MMSE                      -0.1074868  0.0097429 -11.032   <2e-16 ***
FunctionalAssessment      -0.4469330  0.0313356 -14.263   <2e-16 ***
MemoryComplaints           2.5530765  0.1974110  12.933   <2e-16 ***
BehavioralProblems         2.6032162  0.2217427  11.740   <2e-16 ***
ADL                       -0.4071257  0.0308867 -13.181   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1116.0  on 1493  degrees of freedom
AIC: 1138

Number of Fisher Scoring iterations: 6
```

H Model 4

H.1 Logistic Regression Model Based on Ridge

```r
ridge_predictors <- c(
  "Age", "Gender", "Ethnicity", "EducationLevel", "BMI", "Smoking",
  "AlcoholConsumption", "PhysicalActivity", "DietQuality", "SleepQuality",
  "FamilyHistoryAlzheimers", "CardiovascularDisease", "Diabetes", "Depression",
  "HeadInjury", "Hypertension", "SystolicBP", "DiastolicBP", "CholesterolTotal",
  "CholesterolLDL", "CholesterolHDL", "CholesterolTriglycerides", "MMSE",
  "FunctionalAssessment", "MemoryComplaints", "BehavioralProblems", "ADL",
  "Confusion", "Disorientation", "PersonalityChanges",
  "DifficultyCompletingTasks", "Forgetfulness"
)
```

```r
# Build Model4: Logistic Regression with Ridge-selected predictors
model4 <- glm(Diagnosis ~ ., data = data_train[, c(ridge_predictors, "Diagnosis")], family =

summary(model4)
```

```
Call:
glm(formula = Diagnosis ~ ., family = binomial, data = data_train[,
    c(ridge_predictors, "Diagnosis")])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6554  -0.5624  -0.2050   0.5060   3.8331

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               4.1462625  1.1439390   3.625 0.000289 ***
Age                      -0.0067931  0.0085250  -0.797 0.425543
Gender                   -0.1547090  0.1530747  -1.011 0.312171
Ethnicity                -0.0619785  0.0777750  -0.797 0.425512
EducationLevel           -0.0023130  0.0845816  -0.027 0.978184
BMI                       0.0023121  0.0106890   0.216 0.828747
Smoking                  -0.2074445  0.1717844  -1.208 0.227206
AlcoholConsumption       -0.0058143  0.0131502  -0.442 0.658385
PhysicalActivity         -0.0017462  0.0259565  -0.067 0.946364
DietQuality               0.0220411  0.0264505   0.833 0.404678
SleepQuality             -0.0659227  0.0438223  -1.504 0.132499
FamilyHistoryAlzheimers  -0.0372964  0.1784721  -0.209 0.834467
CardiovascularDisease     0.2209397  0.2138862   1.033 0.301614
Diabetes                 -0.1938149  0.2134517  -0.908 0.363876
Depression                0.1121779  0.1844417   0.608 0.543053
HeadInjury               -0.1626499  0.2545229  -0.639 0.522798
Hypertension              0.2142018  0.2134160   1.004 0.315532
SystolicBP                0.0006931  0.0029546   0.235 0.814534
DiastolicBP               0.0031639  0.0042854   0.738 0.460333
CholesterolTotal          0.0004866  0.0017920   0.272 0.785955
CholesterolLDL           -0.0004726  0.0017914  -0.264 0.791942
CholesterolHDL            0.0028991  0.0033171   0.874 0.382129
CholesterolTriglycerides  0.0006480  0.0007493   0.865 0.387157
MMSE                     -0.1086078  0.0099055 -10.964  < 2e-16 ***
FunctionalAssessment     -0.4512320  0.0318726 -14.157  < 2e-16 ***
```

```
MemoryComplaints           2.5453201  0.1992550  12.774  < 2e-16 ***
BehavioralProblems         2.6022302  0.2248095  11.575  < 2e-16 ***
ADL                       -0.4116667  0.0313403 -13.135  < 2e-16 ***
Confusion                 -0.2121769  0.1900947  -1.116 0.264352
Disorientation             0.0606463  0.2087049   0.291 0.771370
PersonalityChanges        -0.0913363  0.2139548  -0.427 0.669456
DifficultyCompletingTasks  0.1092417  0.2087682   0.523 0.600788
Forgetfulness              0.0625052  0.1665791   0.375 0.707491
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1108.1  on 1471  degrees of freedom
AIC: 1174.1

Number of Fisher Scoring iterations: 6
```

#Compare models ## Compare AIC and BIC

```r
# AIC and BIC for Model 1 (Full Model)
aic_model1 <- AIC(model)
bic_model1 <- BIC(model)

# AIC and BIC for Model 2 (Reduced Model)
aic_model2 <- AIC(model2)
bic_model2 <- BIC(model2)

# AIC and BIC for Model 3 (LASSO-Selected Model)
aic_model3 <- AIC(model3)
bic_model3 <- BIC(model3)

# AIC and BIC for Model 4 (Ridge-Selected Model)
aic_model4 <- AIC(model4)
bic_model4 <- BIC(model4)

# Combine results into a data frame for comparison
comparison <- data.frame(
  Model = c("Model1_Full", "Model2_Reduced", "Model3_LASSO", "Model4_Ridge"),
  AIC = c(aic_model1, aic_model2, aic_model3, aic_model4),
  BIC = c(bic_model1, bic_model2, bic_model3, bic_model4)
```

```
)

# Display the comparison
print(comparison)
```

```
           Model      AIC      BIC
1    Model1_Full 1660.135 1824.927
2 Model2_Reduced 1624.685 1651.264
3   Model3_LASSO 1138.041 1196.516
4   Model4_Ridge 1174.125 1349.549
```

H.2 K-Fold Cross-Validation for each model

```
# Load necessary libraries
library(caret)
library(pROC)

# Define training control for K-fold Cross-Validation
train_control <- trainControl(
  method = "cv",
  number = 10,
  classProbs = TRUE,
  summaryFunction = twoClassSummary
)

# Ensure Diagnosis is a factor and rename levels
data_train$Diagnosis <- as.factor(data_train$Diagnosis)
levels(data_train$Diagnosis) <- c("No", "Yes")  # Rename levels to "No" and "Yes"


# Model 1: Full Set of Predictors
set.seed(123)
cv_model1 <- train(Diagnosis ~ Age + Gender + Ethnicity + EducationLevel + Smoking +
                AlcoholConsumption + PhysicalActivity + DietQuality + SleepQuality +
                BMI + Hypertension + SystolicBP + DiastolicBP + CholesterolTotal +
                CholesterolLDL + CholesterolHDL + CholesterolTriglycerides +
                Diabetes + CardiovascularDisease + FamilyHistoryAlzheimers +
                HeadInjury + Depression + MMSE + MemoryComplaints + BehavioralProblems +
                Confusion + Disorientation + PersonalityChanges + DifficultyCompletingTasks +
                Forgetfulness,  data = data_train,
```

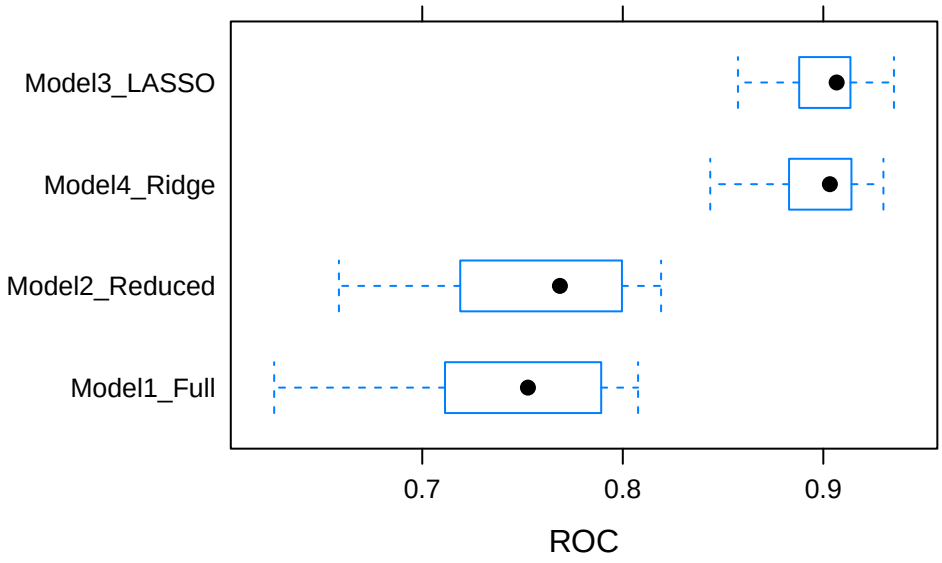
```
  method = "glm",
  family = "binomial",
  trControl = train_control,
  metric = "ROC"
)

# Model 2: Reduced Logistic Model (Hypothesis-based)
set.seed(123)
cv_model2 <- train(
  Diagnosis ~ SleepQuality + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "glm",
  family = "binomial",
  trControl = train_control,
  metric = "ROC"
)

# Model 3: LASSO-Selected Predictors
set.seed(123)
cv_model3 <- train(
  Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +
    Hypertension + CholesterolTriglycerides + MMSE +
    FunctionalAssessment + MemoryComplaints +
    BehavioralProblems + ADL,
  data = data_train,
  method = "glm",
  family = "binomial",
  trControl = train_control,
  metric = "ROC"
)

# Model 4: Ridge-Selected Predictors
set.seed(123)
cv_model4 <- train(
  Diagnosis ~ Age + Gender + Ethnicity + EducationLevel + BMI + Smoking +
    AlcoholConsumption + PhysicalActivity + DietQuality + SleepQuality +
    FamilyHistoryAlzheimers + CardiovascularDisease + Diabetes + Depression +
    HeadInjury + Hypertension + SystolicBP + DiastolicBP + CholesterolTotal +
    CholesterolLDL + CholesterolHDL + CholesterolTriglycerides + MMSE +
    FunctionalAssessment + MemoryComplaints + BehavioralProblems + ADL +
    Confusion + Disorientation + PersonalityChanges + DifficultyCompletingTasks +
    Forgetfulness,
```

```r
  data = data_train,
  method = "glm",
  family = "binomial",
  trControl = train_control,
  metric = "ROC"
)

# Compare Model Performance
results <- resamples(list(
  Model1_Full = cv_model1,
  Model2_Reduced = cv_model2,
  Model3_LASSO = cv_model3,
  Model4_Ridge = cv_model4
))

# Summarize and visualize results
summary(results)
```

```
Call:
summary.resamples(object = results)

Models: Model1_Full, Model2_Reduced, Model3_LASSO, Model4_Ridge
Number of resamples: 10

ROC
                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Model1_Full    0.6261428 0.7205310 0.7526944 0.7468952 0.7842136 0.8076250    0
Model2_Reduced 0.6584322 0.7290897 0.7686387 0.7620780 0.7993517 0.8191013    0
Model3_LASSO   0.8574207 0.8896237 0.9066508 0.8999466 0.9134977 0.9352266    0
Model4_Ridge   0.8436102 0.8833011 0.9032633 0.8950991 0.9128574 0.9299747    0

Sens
                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Model1_Full    0.8350515 0.8788660 0.8865979 0.8878708 0.9072165 0.9278351    0
Model2_Reduced 0.8775510 0.8865979 0.9020619 0.9022828 0.9151851 0.9278351    0
Model3_LASSO   0.8556701 0.8894382 0.9072165 0.9032927 0.9151851 0.9381443    0
Model4_Ridge   0.8350515 0.8695824 0.8917526 0.8899116 0.9149485 0.9278351    0

Spec
                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Model1_Full    0.3207547 0.4553634 0.4811321 0.4813417 0.5094340 0.6603774    0
```

```
Model2_Reduced 0.3773585 0.4444444 0.4622642 0.4888889 0.5377358 0.6226415    0
Model3_LASSO   0.6415094 0.7182914 0.7358491 0.7311670 0.7534067 0.7924528    0
Model4_Ridge   0.6792453 0.7041405 0.7382949 0.7293152 0.7547170 0.7735849    0
```

```r
bwplot(results, metric = "ROC") # Boxplot for ROC scores
```



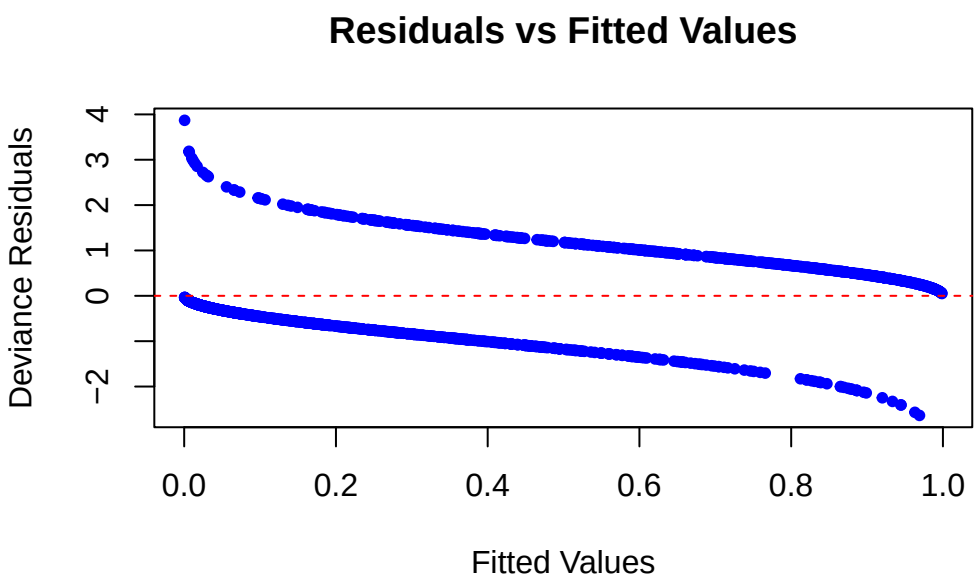

```r
summary_resamples <- summary(results)

# Access the mean ROC for each model
mean_roc <- summary_resamples$statistics$ROC[, "Mean"]
print(mean_roc)
```

```
   Model1_Full Model2_Reduced   Model3_LASSO   Model4_Ridge
     0.7468952      0.7620780      0.8999466      0.8950991
```

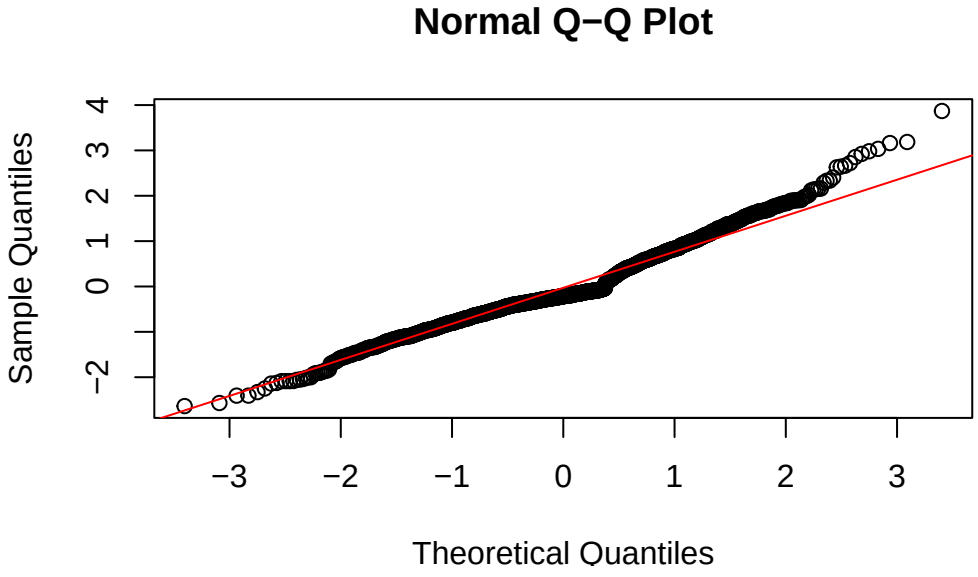
I Checking Linearity of Model3

```r
# Plot residuals vs fitted values for model3
plot(fitted(model3), residuals(model3, type = "deviance"),
     xlab = "Fitted Values",
     ylab = "Deviance Residuals",
     main = "Residuals vs Fitted Values",
```

```
     pch = 20, col = "blue")

# Add a horizontal line at 0 for reference
abline(h = 0, col = "red", lty = 2)
```




```
# Q-Q plot for residuals
qqnorm(residuals(model3, type = "deviance"))
qqline(residuals(model3, type = "deviance"), col = "red")
```



Residuals vs Fitted Plot: Indicates non-linearity in predictors. Q-Q Plot: Suggests approximate normality but highlights potential outliers or heavy-tailed residuals.

```
sapply(data_train[, c("Smoking", "SleepQuality", "CardiovascularDisease",
                      "Hypertension", "CholesterolTriglycerides", "MMSE",
                      "FunctionalAssessment", "MemoryComplaints",
                      "BehavioralProblems", "ADL")], function(x) length(unique(x)))
```

```
                Smoking             SleepQuality    CardiovascularDisease
                      2                     1504                        2
           Hypertension CholesterolTriglycerides                     MMSE
                      2                     1504                     1504
    FunctionalAssessment         MemoryComplaints       BehavioralProblems
                   1504                        2                        2
                    ADL
                   1504
```

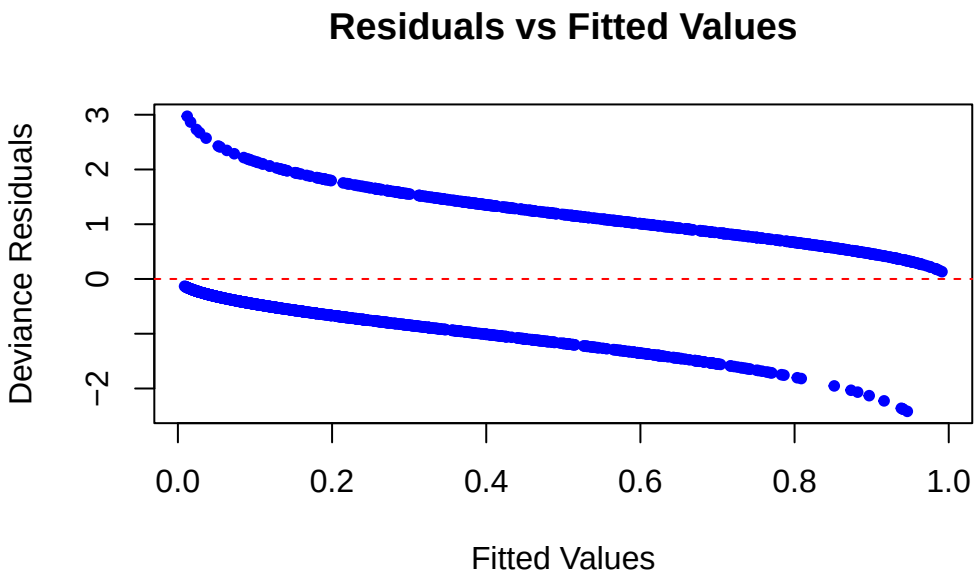
I.1 Try the interaction

```
# Add interaction terms
model3_inter <- glm(
  Diagnosis ~ Smoking * SleepQuality + CardiovascularDisease * Hypertension +
              CholesterolTriglycerides + MMSE + MemoryComplaints +
              BehavioralProblems + ADL,
  data = data_train, family = binomial
)

# View model summary
summary(model3_inter)
```

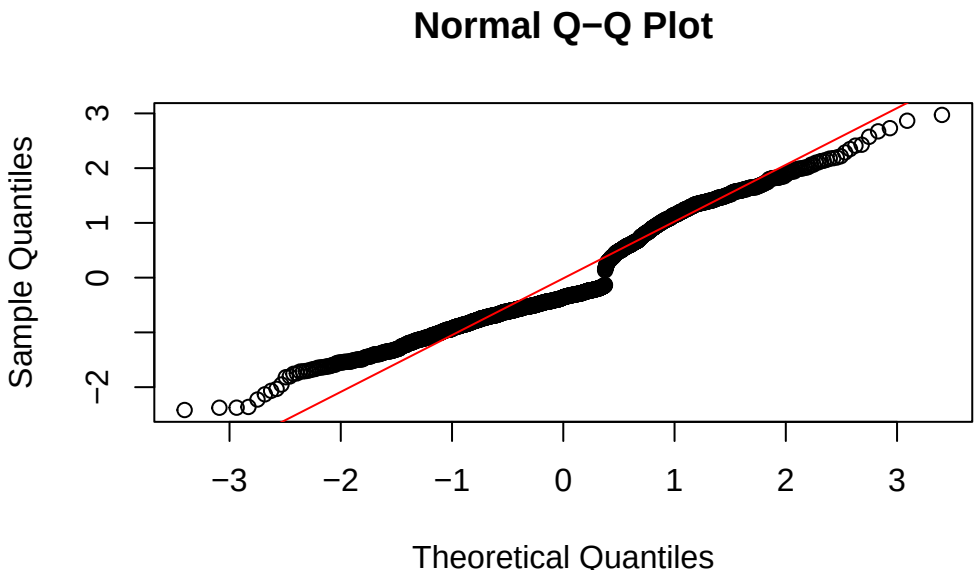
```
Call:
glm(formula = Diagnosis ~ Smoking * SleepQuality + CardiovascularDisease *
    Hypertension + CholesterolTriglycerides + MMSE + MemoryComplaints +
    BehavioralProblems + ADL, family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4178  -0.7124  -0.3581   0.6860   2.9724
```

```
Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       1.701383   0.371781   4.576 4.73e-06 ***
Smoking                          -0.604938   0.636529  -0.950   0.3419
SleepQuality                     -0.094183   0.043687  -2.156   0.0311 *
CardiovascularDisease             0.457824   0.212471   2.155   0.0312 *
Hypertension                      0.394952   0.198299   1.992   0.0464 *
CholesterolTriglycerides          0.001053   0.000651   1.617   0.1059
MMSE                             -0.087236   0.008425 -10.354  < 2e-16 ***
MemoryComplaints                  2.056578   0.168383  12.214  < 2e-16 ***
BehavioralProblems                2.059502   0.188504  10.925  < 2e-16 ***
ADL                              -0.343837   0.025937 -13.257  < 2e-16 ***
Smoking:SleepQuality              0.070147   0.087697   0.800   0.4238
CardiovascularDisease:Hypertension -0.742703   0.511245  -1.453   0.1463
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1954.4  on 1503  degrees of freedom
Residual deviance: 1391.4  on 1492  degrees of freedom
AIC: 1415.4

Number of Fisher Scoring iterations: 5
```

```r
# Plot residuals vs fitted values for model3
plot(fitted(model3_inter), residuals(model3_inter, type = "deviance"),
     xlab = "Fitted Values",
     ylab = "Deviance Residuals",
     main = "Residuals vs Fitted Values",
     pch = 20, col = "blue")

# Add a horizontal line at 0 for reference
abline(h = 0, col = "red", lty = 2)
```




```
# Q-Q plot for residuals
qqnorm(residuals(model3_inter, type = "deviance"))
qqline(residuals(model3_inter, type = "deviance"), col = "red")
```



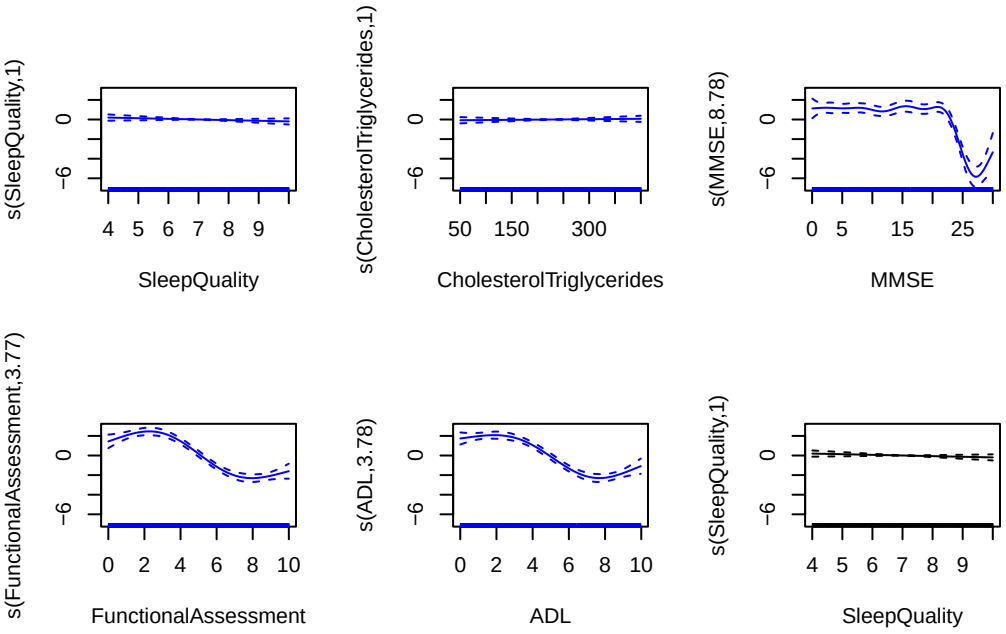
J Try the Generalized Additive Model

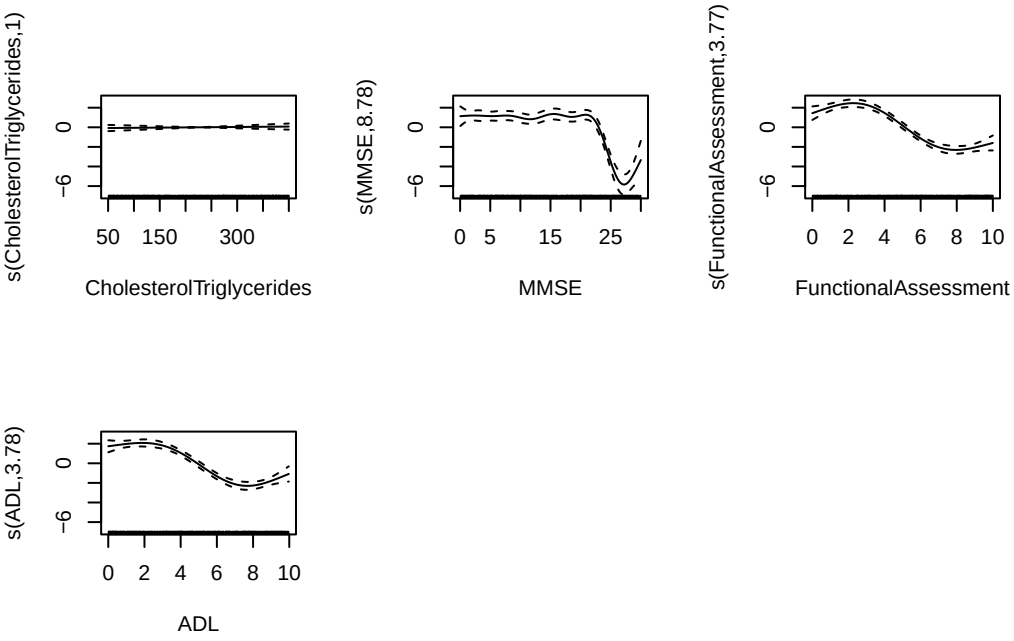
```r
library(mgcv)
# Fit the Generalized Additive Model
model_gam <- gam(Diagnosis ~ Smoking + CardiovascularDisease + Hypertension +
                     MemoryComplaints + BehavioralProblems +
                     s(SleepQuality, k = 5) + s(CholesterolTriglycerides, k = 5) +
                     s(MMSE, k = 10) + s(FunctionalAssessment, k = 5) + s(ADL, k = 5),
                 data = data_train, family = binomial)

# View the model summary
summary(model_gam)
```

```
Family: binomial
Link function: logit

Formula:
Diagnosis ~ Smoking + CardiovascularDisease + Hypertension +
    MemoryComplaints + BehavioralProblems + s(SleepQuality, k = 5) +
    s(CholesterolTriglycerides, k = 5) + s(MMSE, k = 10) + s(FunctionalAssessment,
    k = 5) + s(ADL, k = 5)

Parametric coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -3.06579    0.19685 -15.574   <2e-16 ***
Smoking              -0.16873    0.20542  -0.821    0.411
CardiovascularDisease  0.28411    0.25870   1.098    0.272
Hypertension           0.04929    0.25162   0.196    0.845
MemoryComplaints       3.60783    0.26766  13.479   <2e-16 ***
BehavioralProblems     3.73018    0.29651  12.580   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                             edf Ref.df  Chi.sq p-value
s(SleepQuality)            1.001  1.001   1.554   0.213
s(CholesterolTriglycerides) 1.000  1.001   0.219   0.640
s(MMSE)                    8.780  8.984 160.492  <2e-16 ***
s(FunctionalAssessment)    3.768  3.967 246.393  <2e-16 ***
s(ADL)                     3.781  3.971 221.304  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R-sq.(adj) =  0.725   Deviance explained = 59.7%
UBRE = -0.44333  Scale est. = 1         n = 1504
```

```r
# Plot the smooth terms
par(mfrow = c(2, 3))  # Arrange plots in a grid
plot(model_gam, se = TRUE, col = "blue")

# Plot the smooth terms to examine the relationships between predictors and diagnosis
plot(model_gam)
```



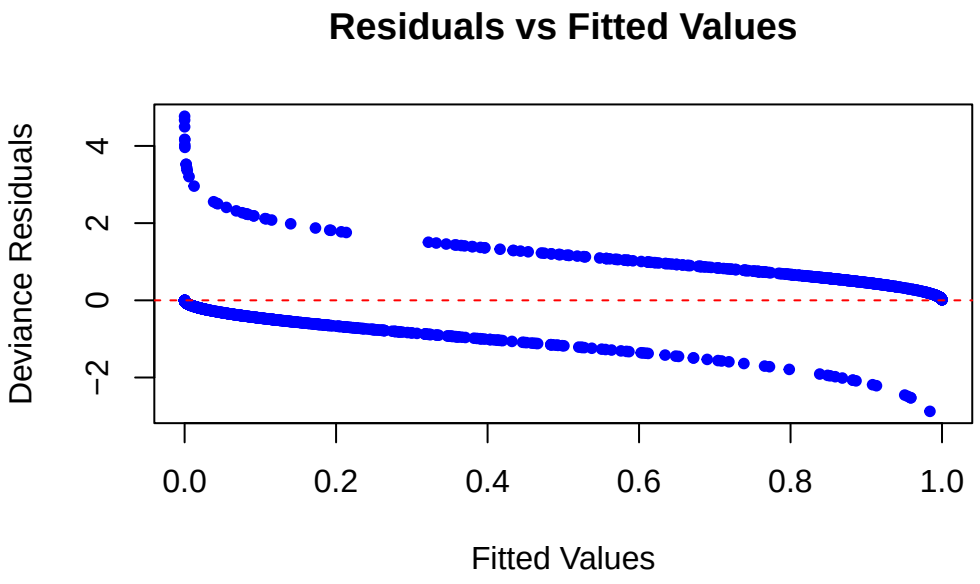



```
aic_model_gam <- AIC(model_gam)
aic_model_gam
```

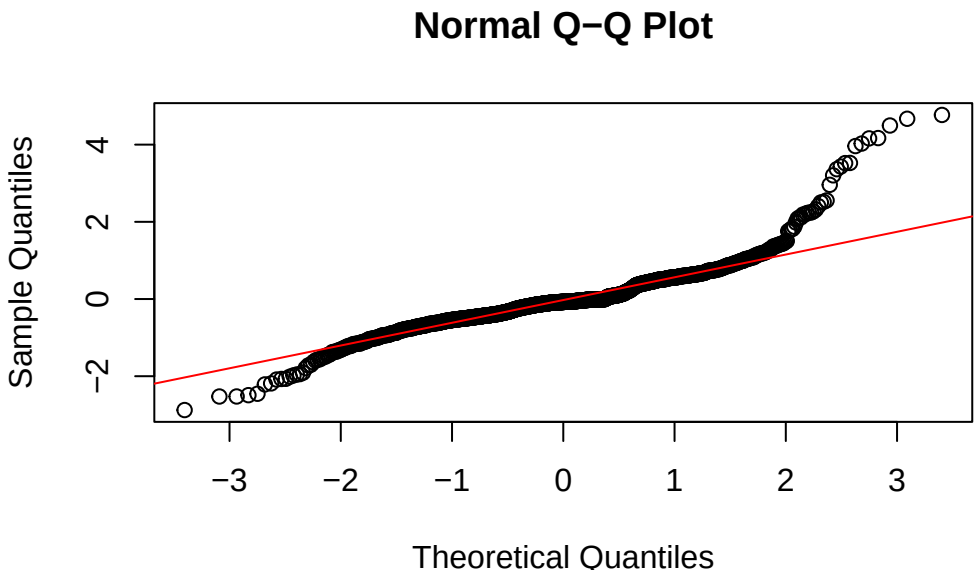
```
[1] 837.2389
```

J.1 Check the linearity of GAM

```
# Plot residuals vs fitted values for model3
plot(fitted(model_gam), residuals(model_gam, type = "deviance"),
     xlab = "Fitted Values",
     ylab = "Deviance Residuals",
     main = "Residuals vs Fitted Values",
     pch = 20, col = "blue")

# Add a horizontal line at 0 for reference
abline(h = 0, col = "red", lty = 2)
```




```
# Q-Q plot for residuals
qqnorm(residuals(model_gam, type = "deviance"))
qqline(residuals(model_gam, type = "deviance"), col = "red")
```




```
data_test <- read_csv(here::here("data/01-raw_data/test.csv"))
```

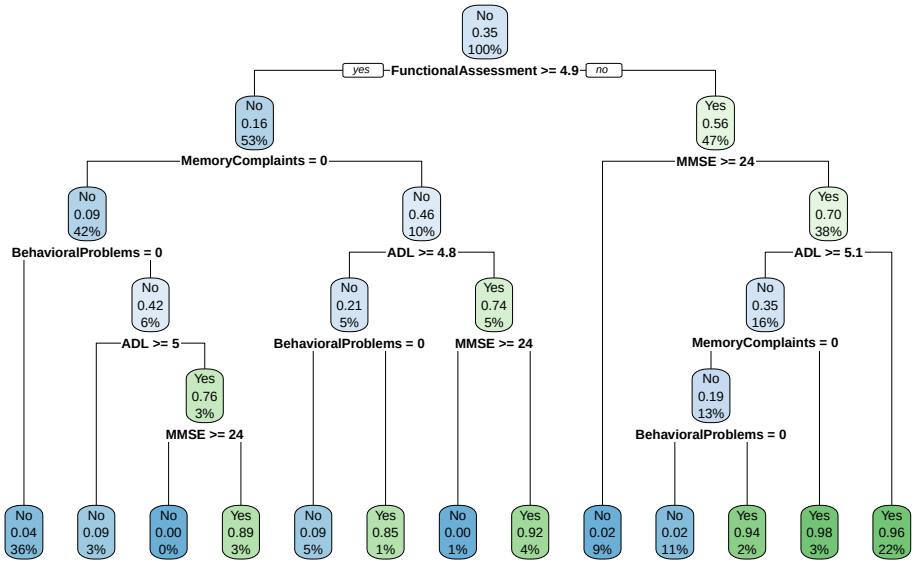
```
Rows: 645 Columns: 34
-- Column specification --------------------------------------------------------
```

```
Delimiter: ","
chr  (1): DoctorInCharge
dbl (33): PatientID, Age, Gender, Ethnicity, EducationLevel, BMI, Smoking, A...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

K Decision Tree

K.1 Based on our the predictors of Model 3 (Lasso) selected before

```
dt_model_lasso <- rpart(
  Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +
    Hypertension + CholesterolTriglycerides + MMSE +
    FunctionalAssessment + MemoryComplaints +
    BehavioralProblems + ADL,
  data = data_train,
  method = "class",
  control = rpart.control(cp = 0.001)  # Lower cp for deeper trees
)
rpart.plot(dt_model_lasso)
```

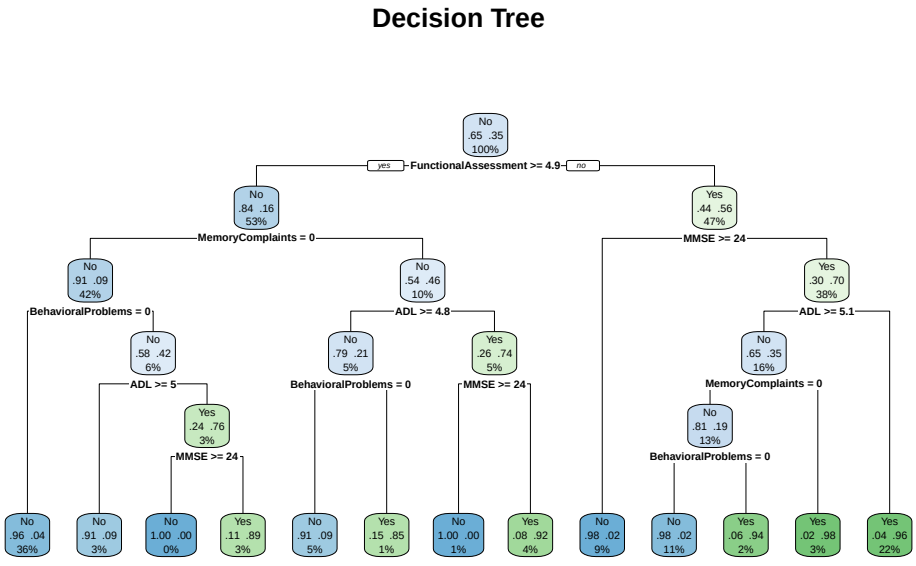


```r
# Select important variables from Lasso
importance <- dt_model_lasso$variable.importance
print(importance[1:5])
```

```
                 ADL                 MMSE FunctionalAssessment
           153.04812            139.84858            126.86176
MemoryComplaints   BehavioralProblems
           85.27250             83.70549
```

K.2 Fit a tree model based on importance selected from Lasso

```r
# Train Decision Tree
tree_model <- rpart(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "class",
  control = rpart.control(cp = 0.01)
)

# Visualize the Decision Tree
rpart.plot(tree_model, type = 2, extra = 104, main = "Decision Tree")
```



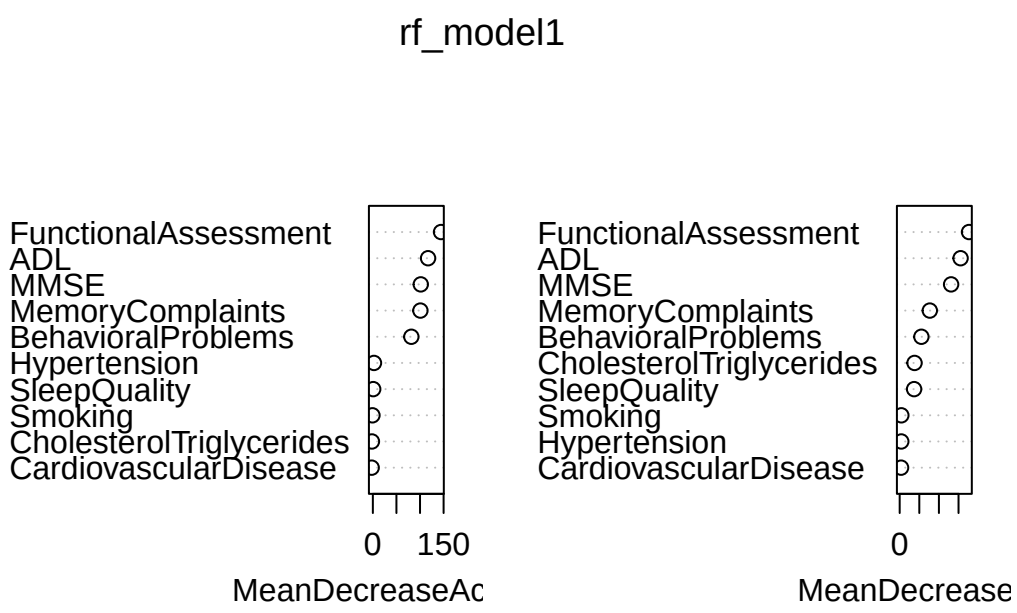
K.3 CV

```
# Set up cross-validation control for the Decision Tree


# Train the Decision Tree model using caret
set.seed(123)
dt_cv_model <- train(Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + Beha
  data = data_train,
  method = "rpart",
  trControl = train_control,
  metric = "ROC"  # Optimize for ROC-AUC
)

# Print CV results
print(dt_cv_model)
```

```
CART

1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results across tuning parameters:

  cp          ROC        Sens       Spec
  0.09022556  0.8442647  0.9866400  0.6109015
  0.14097744  0.8111510  0.8857774  0.6523410
  0.21052632  0.6647021  0.8816221  0.4269741

ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.09022556.
```

L Random Forest

```r
# Fit a Random Forest Model based on predictors selected by Lasso before

rf_model1 <- randomForest(Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +
    Hypertension + CholesterolTriglycerides + MMSE +
    FunctionalAssessment + MemoryComplaints +
    BehavioralProblems + ADL,
    data = data_train, importance = TRUE)

# Variable Importance Plot
varImpPlot(rf_model1)
```

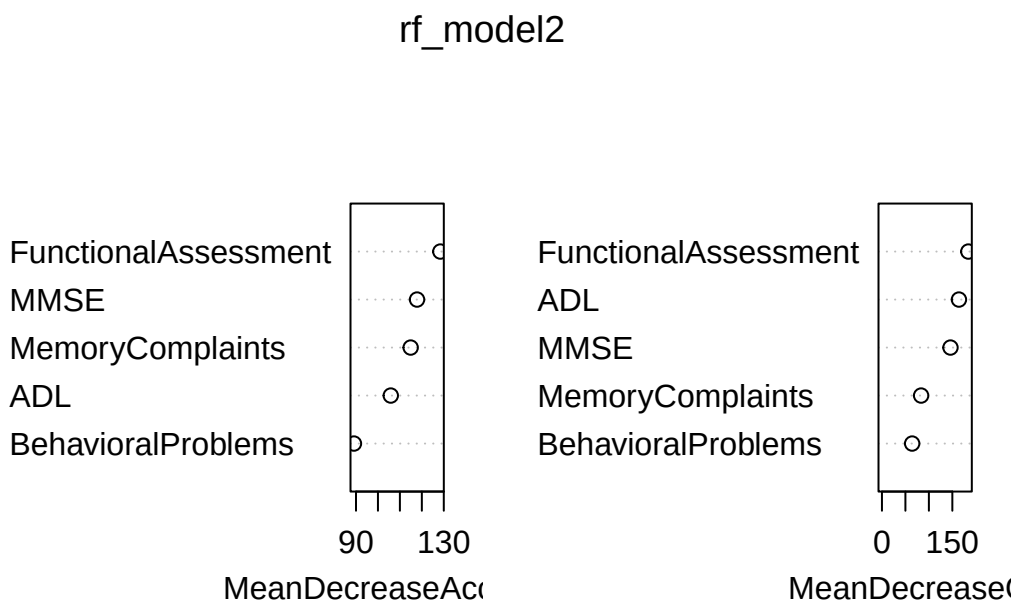



```r
# Predictions
rf_preds <- predict(rf_model1, newdata = data_test)
```

```r
# Take variable importance from Random Forest model 1
rf_importance <- importance(rf_model1)
rf_top_predictors <- rownames(rf_importance)[order(-rf_importance[, 1])] # Sort by importance

# Select the top predictors
rf_selected_predictors <- rf_top_predictors[1:5] # Top 5 predictors
print(rf_selected_predictors)
```

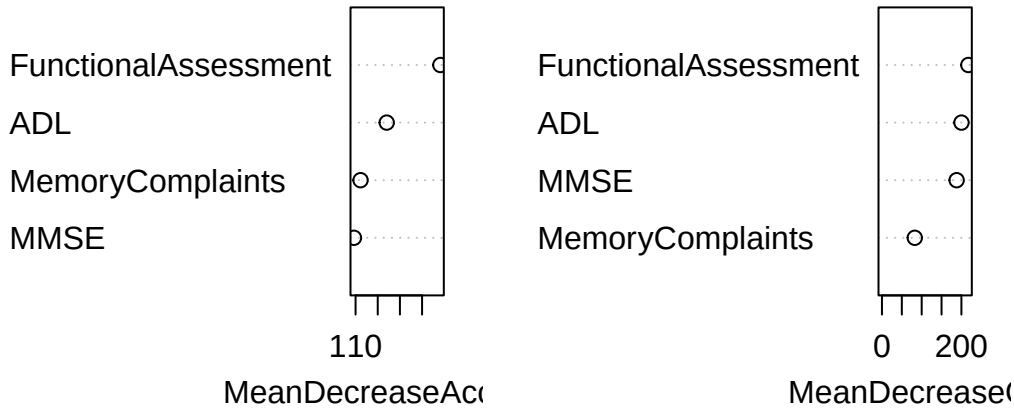
```
[1] "FunctionalAssessment" "ADL"                  "MMSE"
[4] "MemoryComplaints"     "BehavioralProblems"
```

```r
rf_model2 <- randomForest(Diagnosis ~ FunctionalAssessment + ADL + MMSE +
                          MemoryComplaints + BehavioralProblems,
                          data = data_train, importance = TRUE)

# Variable Importance Plot
varImpPlot(rf_model2)
```




```r
# Predictions
rf_preds <- predict(rf_model2, newdata = data_test)
```

```r
rf_model3 <- randomForest(Diagnosis ~ FunctionalAssessment + ADL + MMSE +
                          MemoryComplaints,
                          data = data_train, importance = TRUE)

# Variable Importance Plot
varImpPlot(rf_model3)
```




```r
# Predictions
rf_preds <- predict(rf_model3, newdata = data_test)
```

```r
print(rf_model1)
```

```
Call:
 randomForest(formula = Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +      Hy
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 4.39%
Confusion matrix:
     No Yes class.error
No  949  23  0.02366255
Yes  43 489  0.08082707
```

```r
print(rf_model2)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
```

```
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4.39%
Confusion matrix:
     No Yes class.error
No  946  26  0.02674897
Yes  40 492  0.07518797
```

```
print(rf_model3)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 9.57%
Confusion matrix:
     No Yes class.error
No  939  33  0.03395062
Yes 111 421  0.20864662
```

```
# Take variable importance from Random Forest
rf_importance <- importance(rf_model1)
rf_top_predictors <- rownames(rf_importance)[order(-rf_importance[, 1])] # Sort by importance

# Select the top predictors for building random forest
rf_selected_predictors <- rf_top_predictors[1:5] # Top 5 predictors
print(rf_selected_predictors)
```

```
[1] "FunctionalAssessment" "ADL"                  "MMSE"
[4] "MemoryComplaints"     "BehavioralProblems"
```

```
set.seed(314)
rf_model_selected <- randomForest(Diagnosis ~ FunctionalAssessment + ADL + MMSE +
                          MemoryComplaints + BehavioralProblems,
                          data = data_train, importance = TRUE)
print(rf_model_selected)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4.26%
Confusion matrix:
     No Yes class.error
No  948  24  0.02469136
Yes  40 492  0.07518797
```

L.1 Adjust some parameters

```r
set.seed(314)
rf_model_adjusted <- randomForest(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE +
    MemoryComplaints + BehavioralProblems,
  data = data_train,
  ntree = 1000,
  nodesize= 3,
  classwt = c(0.7, 0.3),
  importance = TRUE
)

print(rf_model_adjusted)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4.26%
Confusion matrix:
     No Yes class.error
No  948  24  0.02469136
Yes  40 492  0.07518797
```

By adjusting some parameters, there is no outstanding improvement.

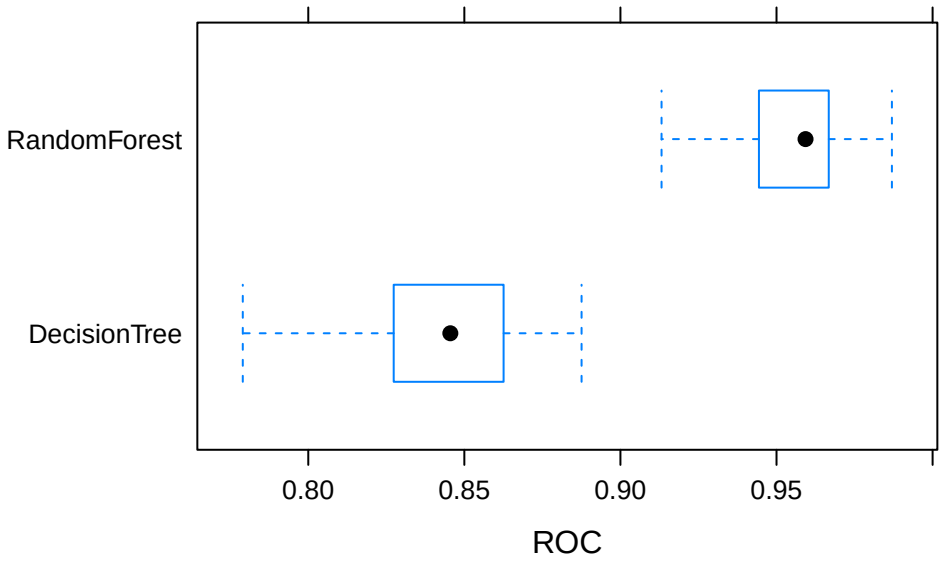
```
# Train Random Forest using caret
set.seed(123)
rf_cv_model <- train(Diagnosis ~ FunctionalAssessment + ADL + MMSE +
    MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "rf",
  trControl = train_control,
  metric = "ROC"
)

# Print CV results for Random Forest
print(rf_cv_model)
```

```
Random Forest

1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
  2     0.9559937  0.9732485  0.9228512
  3     0.9547147  0.9732485  0.9228512
  5     0.9545008  0.9691353  0.9172257

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```
# Combine the results for comparison
resamples <- resamples(list(DecisionTree = dt_cv_model, RandomForest = rf_cv_model))

# Summarize the results
summary(resamples)
```

```
Call:
summary.resamples(object = resamples)

Models: DecisionTree, RandomForest
Number of resamples: 10

ROC
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
DecisionTree 0.7790187 0.8283862 0.8455067 0.8442647 0.8621558 0.8875525    0
RandomForest 0.9131690 0.9457241 0.9592914 0.9559937 0.9661788 0.9869675    0

Sens
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
DecisionTree 0.9690722 0.9793814 0.9896907 0.9866400 0.9974490 1.0000000    0
RandomForest 0.9587629 0.9690722 0.9692300 0.9732485 0.9793814 0.9896907    0

Spec
                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
DecisionTree 0.4444444 0.5377358 0.6320755 0.6109015 0.6698113 0.7777778    0
RandomForest 0.8679245 0.9056604 0.9150943 0.9228512 0.9390287 0.9811321    0
```

```r
# Boxplot to compare performance
bwplot(resamples, metric = "ROC")
```



L.2 Prediction of Diagnosis based on Random Forest

L.2.1 Explore the Thresholds

```r
# Threshold
library(caret)  # For cross-validation
library(randomForest)  # For Random Forest if additional models are needed

# Number of folds for cross-validation
k <- 5
set.seed(42)  # For reproducibility

# Create cross-validation folds
folds <- createFolds(data_train$Diagnosis, k = k, list = TRUE, returnTrain = TRUE)

# Initialize performance metrics
accuracy_list <- c()
precision_list <- c()
recall_list <- c()

# Perform k-fold cross-validation
for (i in 1:k) {
  # Split train and validation sets
  train_fold <- data_train[folds[[i]], ]
  valid_fold <- data_train[-folds[[i]], ]

  # Predict probabilities on validation fold
  probs <- predict(rf_model_selected, newdata = valid_fold, type = "prob")[, 2]

  # Apply threshold p = 0.5
  preds <- ifelse(probs >= 0.5, 1, 0)
  actual <- valid_fold$Diagnosis

  # Confusion matrix
  cm <- table(Predicted = preds, Actual = actual)

  # Calculate metrics
  accuracy <- sum(diag(cm)) / sum(cm)
  precision <- ifelse(sum(preds == 1) > 0, sum(preds == 1 & actual == 1) / sum(preds == 1), N
  recall <- ifelse(sum(actual == 1) > 0, sum(preds == 1 & actual == 1) / sum(actual == 1), NA
```

```r
  # Store metrics
  accuracy_list <- c(accuracy_list, accuracy)
  precision_list <- c(precision_list, precision)
  recall_list <- c(recall_list, recall)
}

# Aggregate performance metrics
mean_accuracy <- mean(accuracy_list, na.rm = TRUE)
mean_precision <- mean(precision_list, na.rm = TRUE)
mean_recall <- mean(recall_list, na.rm = TRUE)

# Print results
cat("Cross-Validation Results (p = 0.5):\n")
```

```
Cross-Validation Results (p = 0.5):
```

```r
cat(paste("Mean Accuracy:", round(mean_accuracy, 4), "\n"))
```

```
Mean Accuracy: 0.9854
```

```r
cat(paste("Mean Precision:", round(mean_precision, 4), "\n"))
```

```
Mean Precision: 0
```

```r
cat(paste("Mean Recall:", round(mean_recall, 4), "\n"))
```

```
Mean Recall: NaN
```

```r
f1_score <- 2 * (mean_precision * mean_recall) / (mean_precision + mean_recall)
print(f1_score)
```

```
[1] NaN
```

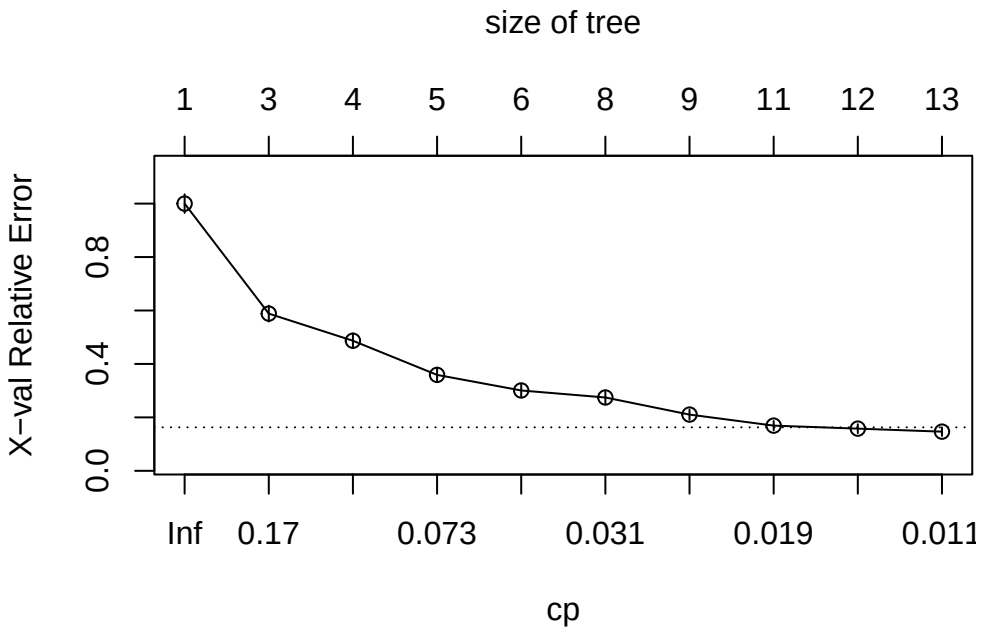
It shows that threshold = 0.5 is the best choice.

M Prune trees

```r
# Find the optimal cp using cross-validation
printcp(tree_model)
```

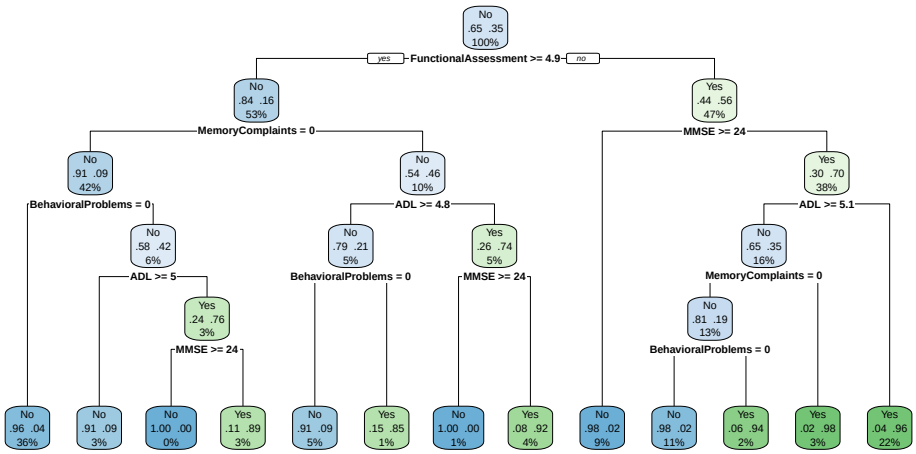
```
Classification tree:
rpart(formula = Diagnosis ~ FunctionalAssessment + ADL + MMSE +
    MemoryComplaints + BehavioralProblems, data = data_train,
    method = "class", control = rpart.control(cp = 0.01))

Variables actually used in tree construction:
[1] ADL                  BehavioralProblems   FunctionalAssessment
[4] MemoryComplaints     MMSE

Root node error: 532/1504 = 0.35372

n= 1504

         CP nsplit rel error  xerror     xstd
1  0.210526      0   1.00000 1.00000 0.034854
2  0.140977      2   0.57895 0.58835 0.029593
3  0.090226      3   0.43797 0.48684 0.027523
4  0.058271      4   0.34774 0.35902 0.024272
5  0.033835      5   0.28947 0.30075 0.022476
6  0.028195      7   0.22180 0.27444 0.021582
7  0.021617      8   0.19361 0.21053 0.019138
8  0.016917     10   0.15038 0.16917 0.017291
9  0.013158     11   0.13346 0.15789 0.016740
10 0.010000     12   0.12030 0.14662 0.016165
```

```r
# Plot the cross-validation error against cp values
plotcp(tree_model)
```




```r
# Prune the tree to the optimal cp
optimal_cp <- tree_model$cptable[which.min(tree_model$cptable[, "xerror"]), "CP"]
pruned_tree <- prune(tree_model, cp = optimal_cp)

# Visualize the pruned tree
rpart.plot(pruned_tree, type = 2, extra = 104, main = "Pruned Decision Tree")
```

Pruned Decision Tree



M.1 Validate the Pruned Tree

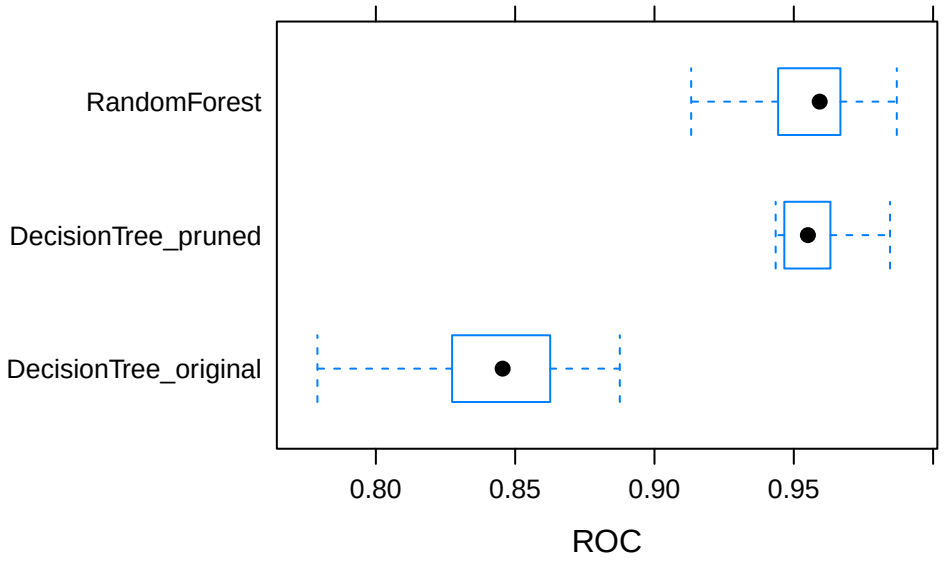
```
# Pruned Decision Tree
pruned_dt_cv <- train(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "rpart",
  trControl = train_control,
  metric = "ROC",
  tuneGrid = expand.grid(cp = optimal_cp)
)
```

```
# Collect Resampling Results
resamples <- resamples(list(
  DecisionTree_original = dt_cv_model,  # Original Decision Tree
  DecisionTree_pruned = pruned_dt_cv,  # Pruned Decision Tree
  RandomForest = rf_cv_model           # Random Forest
))

# Summary of Results
summary(resamples)
```

```
Call:
summary.resamples(object = resamples)

Models: DecisionTree_original, DecisionTree_pruned, RandomForest
Number of resamples: 10

ROC
                          Min.   1st Qu.    Median      Mean   3rd Qu.
DecisionTree_original 0.7790187 0.8283862 0.8455067 0.8442647 0.8621558
DecisionTree_pruned   0.9434925 0.9485950 0.9550671 0.9574057 0.9623857
RandomForest          0.9131690 0.9457241 0.9592914 0.9559937 0.9661788
                          Max. NA's
DecisionTree_original 0.8875525    0
DecisionTree_pruned   0.9845361    0
RandomForest          0.9869675    0

Sens
                          Min.   1st Qu.    Median      Mean   3rd Qu.
```

```
DecisionTree_original 0.9690722 0.9793814 0.9896907 0.9866400 0.9974490
DecisionTree_pruned   0.9285714 0.9511361 0.9742268 0.9650642 0.9793814
RandomForest          0.9587629 0.9690722 0.9692300 0.9732485 0.9793814
                           Max. NA's
DecisionTree_original 1.0000000    0
DecisionTree_pruned   0.9793814    0
RandomForest          0.9896907    0

Spec
                           Min.   1st Qu.    Median      Mean   3rd Qu.
DecisionTree_original 0.4444444 0.5377358 0.6320755 0.6109015 0.6698113
DecisionTree_pruned   0.8679245 0.9103774 0.9252271 0.9247030 0.9433962
RandomForest          0.8679245 0.9056604 0.9150943 0.9228512 0.9390287
                           Max. NA's
DecisionTree_original 0.7777778    0
DecisionTree_pruned   0.9814815    0
RandomForest          0.9811321    0
```

```r
# Create a Boxplot
bwplot(resamples, metric = "ROC")  # Boxplot for ROC-AUC
```




```r
summary_resamples <- summary(resamples)

# Show the mean ROC for each model
mean_roc <- summary_resamples$statistics$ROC[, "Mean"]
print(mean_roc)
```

```
DecisionTree_original   DecisionTree_pruned          RandomForest
            0.8442647             0.9574057             0.9559937
```

If interpretability is critical, choose the pruned Decision Tree. If predictive performance and stability are more important, choose Random Forest since it shows smaller variability (stability and consistency).

```
print(rf_model1)
```

```
Call:
 randomForest(formula = Diagnosis ~ Smoking + SleepQuality + CardiovascularDisease +      Hyp
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 4.39%
Confusion matrix:
     No Yes class.error
No  949  23  0.02366255
Yes  43 489  0.08082707
```

```
print(rf_model2)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 4.39%
Confusion matrix:
     No Yes class.error
No  946  26  0.02674897
Yes  40 492  0.07518797
```

```
print(rf_model3)
```

```
Call:
 randomForest(formula = Diagnosis ~ FunctionalAssessment + ADL +      MMSE + MemoryComplaints
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 9.57%
Confusion matrix:
     No Yes class.error
No  939  33  0.03395062
Yes 111 421  0.20864662
```

N Prediction for Kaggle Submission

```r
# Predicted Probability
test_probabilities <- predict(rf_model1, newdata = data_test, type = "prob")[, 2]

threshold <- 0.5
binary_predictions <- ifelse(test_probabilities >= threshold, 1, 0)

submission <- data.frame(PatientID = data_test$PatientID, Diagnosis = binary_predictions)
write_csv(submission, "/Users/dingshuo/Desktop/submission.csv")
```

O Bagging

```r
# Train a bagged model
set.seed(123)
bagged_model <- bagging(
  Diagnosis ~ .,
  data = data_train,
  nbagg = 500
)

# Predictions
bagged_preds <- predict(bagged_model, newdata = data_train, type = "class")
confusionMatrix(bagged_preds, data_train$Diagnosis)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  972   0
       Yes   0 532

               Accuracy : 1
                 95% CI : (0.9976, 1)
    No Information Rate : 0.6463
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.6463
         Detection Rate : 0.6463
   Detection Prevalence : 0.6463
      Balanced Accuracy : 1.0000

       'Positive' Class : No
```

```r
# Bagging Implementation for Your Data
set.seed(123)

# Bagging (mtry = Total Number of Predictors)
bagging_model <- randomForest(
  Diagnosis ~ .,
  data = data_train,
  mtry = ncol(data_train) - 1,  # Use all predictors
  ntree = 500,                  # Number of trees
  importance = TRUE
)

# Evaluate Bagging Performance
bagging_preds <- predict(bagging_model, newdata = data_test, type = "prob")[, 2]
```

```r
importance_values <- importance(bagging_model)
top_5_vars <- importance_values[order(-importance_values[, "MeanDecreaseGini"]), ][1:5, ]
print(top_5_vars)
```

```
                           No      Yes MeanDecreaseAccuracy MeanDecreaseGini
ADL                 139.61575 148.9391             185.6214        147.80418
MMSE                134.85589 129.6631             170.5507        145.78559
FunctionalAssessment 144.89437 165.9758             210.1697        138.59485
MemoryComplaints    103.48981 126.0208             144.7189         86.68303
BehavioralProblems   81.50375 115.4799             128.2553         77.46854
```

```r
# library(caret)

# Define a grid of hyperparameters to test
tune_grid <- expand.grid(mtry = c(2, 3, 4, 5))

# Perform 10-fold CV with ROC optimization
set.seed(123)
rf_tuned <- train(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "rf",
  trControl = trainControl(method = "cv", number = 10, classProbs = TRUE, summaryFunction = t
  tuneGrid = tune_grid,
  metric = "ROC",
  ntree = 1000  # Increase number of trees
)

# Print results
print(rf_tuned$bestTune)
```

```
  mtry
1    2
```

```r
print(rf_tuned)
```

```
Random Forest

1504 samples
```

```
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
  2     0.9561983  0.9732485  0.9228512
  3     0.9532619  0.9732485  0.9228512
  4     0.9530728  0.9711866  0.9209644
  5     0.9522524  0.9701557  0.9153389

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

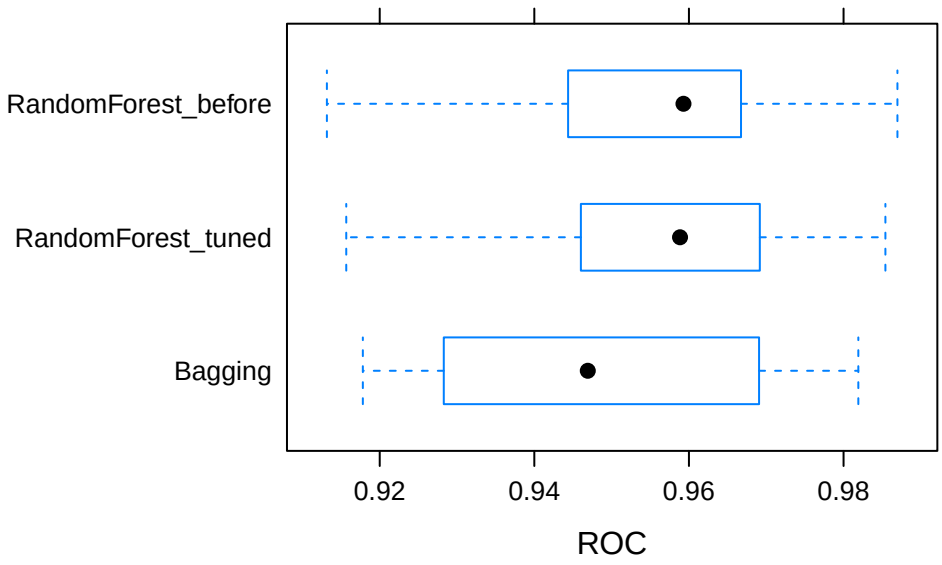
```r
bagged_cv <- train(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "treebag",
  trControl = trainControl(method = "cv", number = 10, classProbs = TRUE, summaryFunction = t
  metric = "ROC"
)
print(bagged_cv)
```

```
Bagged CART

1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1354, 1354, 1354, 1353, 1354, 1354, ...
Resampling results:

  ROC        Sens      Spec
  0.9495235  0.966053  0.9154088
```

```r
# Random Forest metrics
print(rf_tuned)
```

```
Random Forest

1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
  2     0.9561983  0.9732485  0.9228512
  3     0.9532619  0.9732485  0.9228512
  4     0.9530728  0.9711866  0.9209644
  5     0.9522524  0.9701557  0.9153389

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```r
# Combine resampling results
resamples <- resamples(list(
  Bagging = bagged_cv,
  RandomForest_tuned = rf_tuned,
  RandomForest_before = rf_cv_model
))

# Summary of results
summary_resamples <- summary(resamples)

# Boxplot for ROC comparison
bwplot(resamples, metric = "ROC")
```




```
# Show the mean ROC for each model
mean_roc <- summary_resamples$statistics$ROC[, "Mean"]
print(mean_roc)
```

```
          Bagging  RandomForest_tuned RandomForest_before
        0.9495235           0.9561983           0.9559937
```

P Bagging Model & Random Forest Model

```
# Train a Bagging model
set.seed(123)
bagging_model <- train(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "treebag",  # Method for Bagging
  trControl = trainControl(method = "cv", number = 10, classProbs = TRUE, summaryFunction = t
  metric = "ROC"
)

# Print Bagging results
print(bagging_model)
```

```
Bagged CART
```

```
1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results:

  ROC        Sens       Spec
  0.9531172  0.9691248  0.9134871
```

```r
# Train a Random Forest model with tuning
set.seed(123)
rf_tuned_model <- train(
  Diagnosis ~ FunctionalAssessment + ADL + MMSE + MemoryComplaints + BehavioralProblems,
  data = data_train,
  method = "rf",  # Method for Random Forest
  trControl = trainControl(method = "cv", number = 10, classProbs = TRUE, summaryFunction = t
  tuneGrid = expand.grid(mtry = c(2, 3, 4)),
  metric = "ROC"
)

# Print Random Forest (tuned) results
print(rf_tuned_model)
```

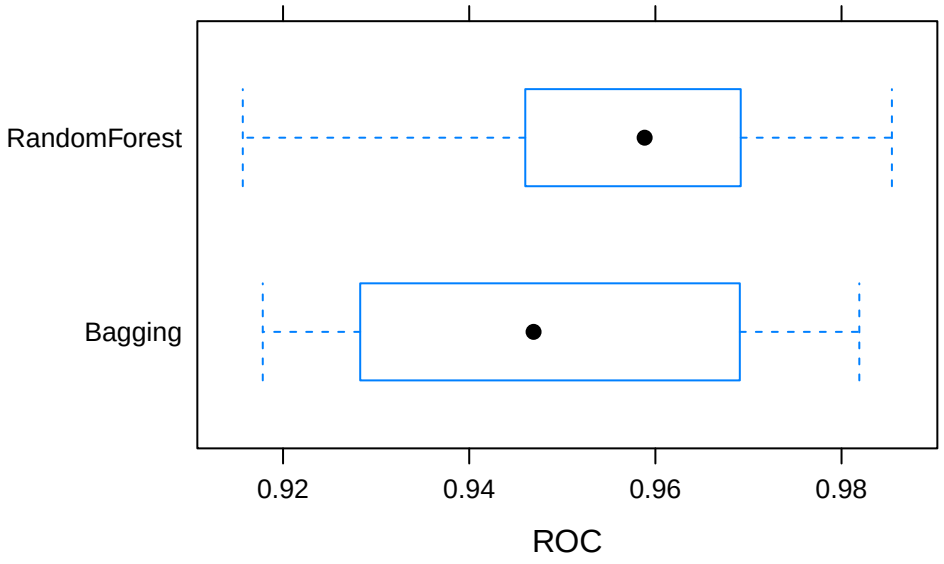
```
Random Forest

1504 samples
   5 predictor
   2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1353, 1354, 1353, 1354, 1354, 1354, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
  2     0.9559937  0.9732485  0.9228512
  3     0.9547147  0.9732485  0.9228512
  4     0.9536000  0.9722175  0.9209993
```

```
ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```r
# Collect cross-validation results for all models
results <- resamples(list(
  Bagging = bagged_cv,
  RandomForest = rf_tuned
))

summary_results <- summary(results)
```

Q Compare 3 Models

```r
# Box plot for comparing XGBoost, Bagging, RandomForest
bwplot(results, metric = "ROC")
```




```r
# Show the mean ROC for each model
mean_roc <- summary_results$statistics$ROC[, "Mean"]
print(mean_roc)
```

```
    Bagging RandomForest
  0.9495235    0.9561983
```

```r
# Predicted Probability
test_probabilities <- predict(rf_tuned_model, newdata = data_test, type = "prob")[, 2]

threshold <- 0.5
binary_predictions <- ifelse(test_probabilities >= threshold, 1, 0)

submission_1 <- data.frame(PatientID = data_test$PatientID, Diagnosis = binary_predictions)
write_csv(submission_1, "/Users/dingshuo/Desktop/submission_1.csv")
```

References

Alzheimer's Association. 2024. *What is Alzheimer's Disease?* https://www.alz.org/alzheimers-dementia/what-is-alzheimers.

Bird, Thomas D. 2018. *Alzheimer Disease*. https://www.ncbi.nlm.nih.gov/books/NBK499922/.

Kaggle. 2024. *Classification of the Alzheimer's Disease Dataset*. https://www.kaggle.com/competitions/classification-of-the-alzheimers-disease/data.