Project

1. Introduction

This project uses a dataset of 299 patients with heart failure to predict survival status (alive or deceased). The data was obtained from the UCI Machine Learning Repository, which contains clinical and laboratory records of patients from the Faisalabad Institute of Cardiology and Allied Hospital (Pakistan), collected in 2015.

The dataset includes demographic, clinical, and laboratory measurements that are commonly used to assess heart failure severity and outcomes.

The goal is to model patient survival (binary classification) using multiple supervised learning models and evaluate their performance using accuracy and AUC metrics.

2. Load and Explore Data

```
library(tidyverse)
library(caret)
library(e1071)
library(randomForest)
library(class)
library(glmnet)
library(mgcv)
library(pROC)
library(rpart)
```

```
data <- read.csv("C:/Users/KojiTakagi/Dropbox/Koji/Master of  Statistics/ST 563/Project/heart_failure_cl
data$DEATH_EVENT <- as.factor(data$DEATH_EVENT)
summary(data)
```

```
##       age           anaemia       creatinine_phosphokinase    diabetes
##  Min.   :40.00   Min.   :0.0000   Min.   :  23.0           Min.   :0.0000
##  1st Qu.:51.00   1st Qu.:0.0000   1st Qu.: 116.5           1st Qu.:0.0000
##  Median :60.00   Median :0.0000   Median : 250.0           Median :0.0000
##  Mean   :60.83   Mean   :0.4314   Mean   : 581.8           Mean   :0.4181
##  3rd Qu.:70.00   3rd Qu.:1.0000   3rd Qu.: 582.0           3rd Qu.:1.0000
##  Max.   :95.00   Max.   :1.0000   Max.   :7861.0           Max.   :1.0000
##  ejection_fraction high_blood_pressure   platelets      serum_creatinine
##  Min.   :14.00     Min.   :0.0000      Min.   : 25100   Min.   :0.500
##  1st Qu.:30.00     1st Qu.:0.0000      1st Qu.:212500   1st Qu.:0.900
##  Median :38.00     Median :0.0000      Median :262000   Median :1.100
##  Mean   :38.08     Mean   :0.3512      Mean   :263358   Mean   :1.394
##  3rd Qu.:45.00     3rd Qu.:1.0000      3rd Qu.:303500   3rd Qu.:1.400
```

```
##  Max.   :80.00     Max.   :1.0000      Max.   :850000   Max.   :9.400
##   serum_sodium        sex            smoking            time       DEATH_EVENT
##  Min.   :113.0   Min.   :0.0000   Min.   :0.0000   Min.   :  4.0   0:203
##  1st Qu.:134.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 73.0   1: 96
##  Median :137.0   Median :1.0000   Median :0.0000   Median :115.0
##  Mean   :136.6   Mean   :0.6488   Mean   :0.3211   Mean   :130.3
##  3rd Qu.:140.0   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:203.0
##  Max.   :148.0   Max.   :1.0000   Max.   :1.0000   Max.   :285.0
```

```
str(data)
```

```
## 'data.frame':    299 obs. of  13 variables:
##  $ age                     : num  75 55 65 50 65 90 75 60 65 80 ...
##  $ anaemia                 : int  0 0 0 1 1 1 1 1 0 1 ...
##  $ creatinine_phosphokinase: int  582 7861 146 111 160 47 246 315 157 123 ...
##  $ diabetes                : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ejection_fraction       : int  20 38 20 20 20 40 15 60 65 35 ...
##  $ high_blood_pressure     : int  1 0 0 0 0 1 0 0 0 1 ...
##  $ platelets               : num  265000 263358 162000 210000 327000 ...
##  $ serum_creatinine        : num  1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ serum_sodium            : int  130 136 129 137 116 132 137 131 138 133 ...
##  $ sex                     : int  1 1 1 1 0 1 1 1 0 1 ...
##  $ smoking                 : int  0 0 1 0 0 1 0 1 0 1 ...
##  $ time                    : int  4 6 7 7 8 8 10 10 10 10 ...
##  $ DEATH_EVENT             : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
sapply(data, function(x) sum(is.na(x)))
```

```
##                      age                  anaemia creatinine_phosphokinase
##                        0                        0                        0
##                 diabetes        ejection_fraction      high_blood_pressure
##                        0                        0                        0
##                platelets         serum_creatinine             serum_sodium
##                        0                        0                        0
##                      sex                  smoking                     time
##                        0                        0                        0
##              DEATH_EVENT
##                        0
```

3. Data Partitioning

```
data_nontime <- data %>% select(-time)
set.seed(123)
splitIndex <- createDataPartition(data_nontime$DEATH_EVENT, p = 0.7, list = FALSE)
train <- data_nontime[splitIndex, ]
test <- data_nontime[-splitIndex, ]
```

4. Model Fitting and Evaluation

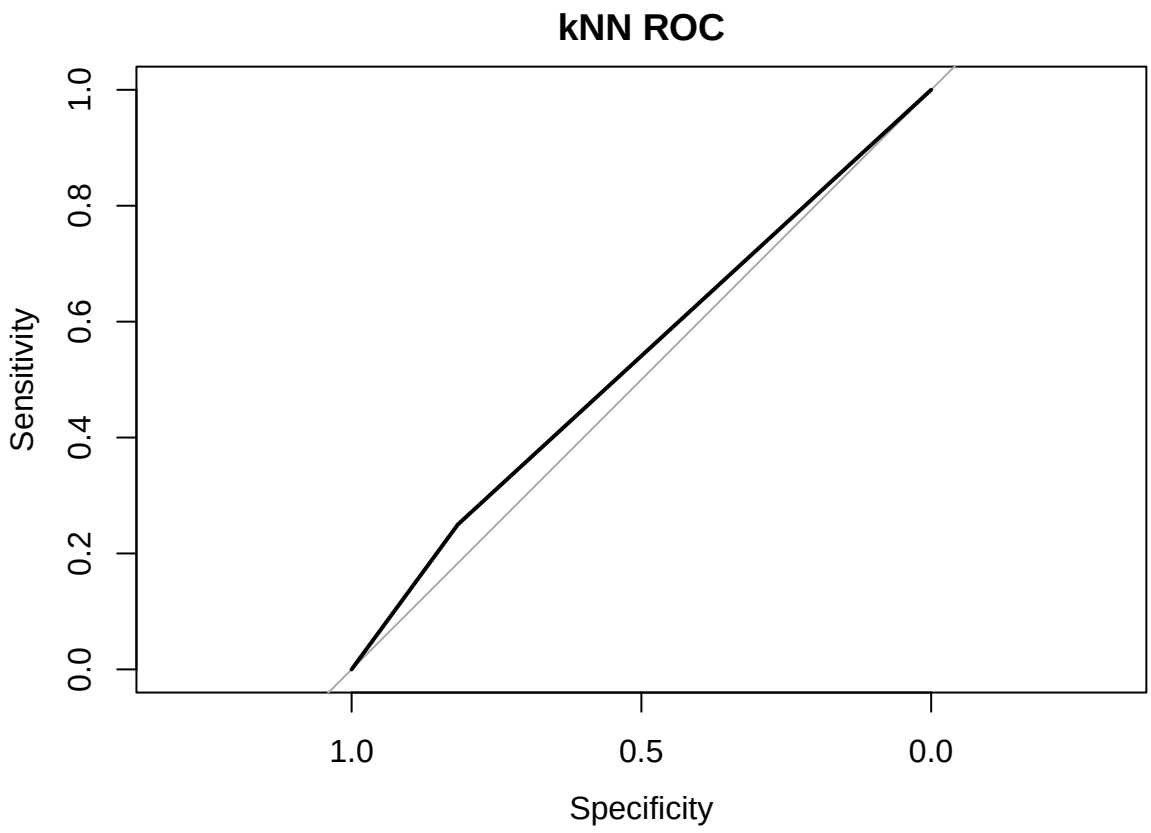
4.1 k-Nearest Neighbors (kNN)

```
train_scaled <- scale(train[, -which(names(train) == "DEATH_EVENT")])
test_scaled <- scale(test[, -which(names(test) == "DEATH_EVENT")],
                     center = attr(train_scaled, "scaled:center"),
                     scale = attr(train_scaled, "scaled:scale"))

knn_pred <- knn(train_scaled, test_scaled, train$DEATH_EVENT, k = 5)
confusionMatrix(knn_pred, test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 49 21
##          1 11  7
##
##                Accuracy : 0.6364
##                  95% CI : (0.5269, 0.7363)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.8484
##
##                   Kappa : 0.0737
##
##  Mcnemar's Test P-Value : 0.1116
##
##             Sensitivity : 0.8167
##             Specificity : 0.2500
##          Pos Pred Value : 0.7000
##          Neg Pred Value : 0.3889
##              Prevalence : 0.6818
##          Detection Rate : 0.5568
##    Detection Prevalence : 0.7955
##       Balanced Accuracy : 0.5333
##
##        'Positive' Class : 0
##
```

```
knn_prob <- as.numeric(knn_pred) - 1
roc_knn <- roc(as.numeric(test$DEATH_EVENT), knn_prob)
plot(roc_knn, main = "kNN ROC")
```




```
auc(roc_knn)
```

```
## Area under the curve: 0.5333
```

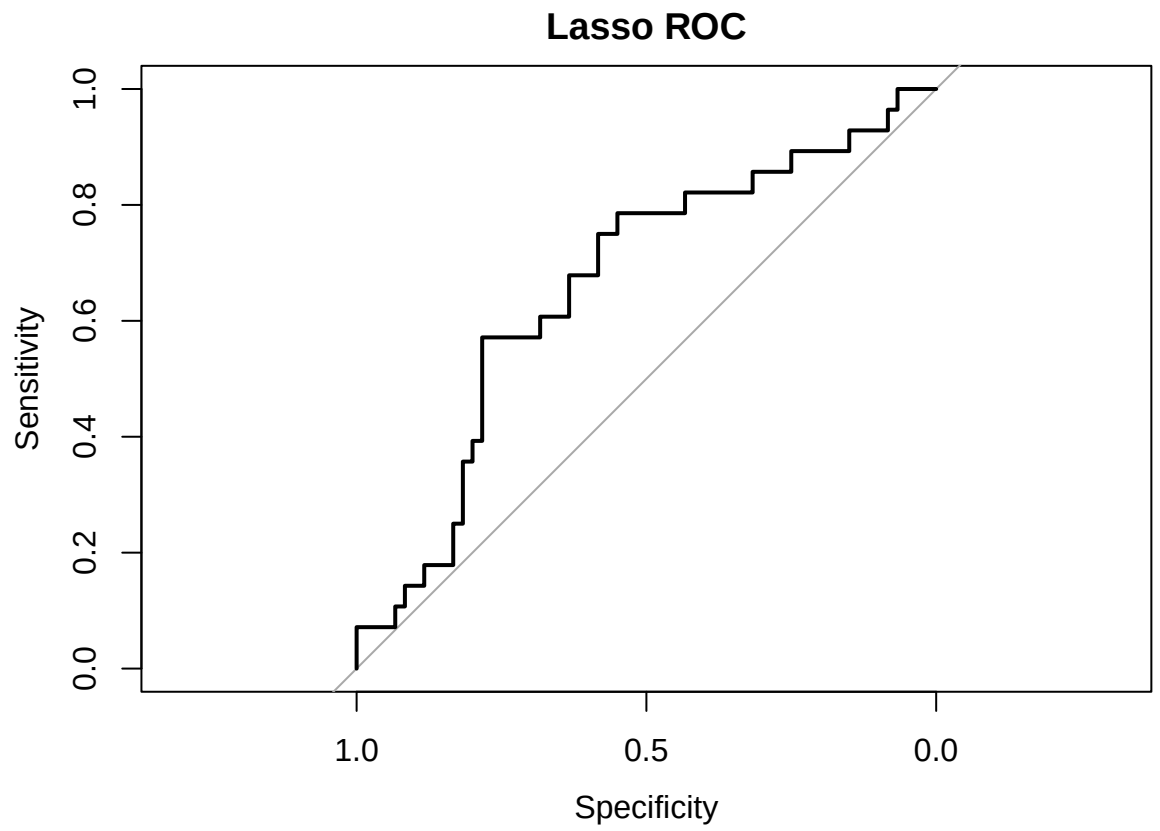
4.2 Logistic Regression with Lasso

```
x_train <- model.matrix(DEATH_EVENT ~ ., train)[,-1]
y_train <- train$DEATH_EVENT
cvfit <- cv.glmnet(x_train, y_train, family = "binomial", alpha = 1)
best_lambda <- cvfit$lambda.min
x_test <- model.matrix(DEATH_EVENT ~ ., test)[,-1]
pred_prob <- predict(cvfit, newx = x_test, s = best_lambda, type = "response")
pred_label <- ifelse(pred_prob > 0.5, 1, 0)
confusionMatrix(as.factor(pred_label), test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 50 21
##          1 10  7
##
##                Accuracy : 0.6477
```

```
##                  95% CI : (0.5386, 0.7466)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.78985
##
##                   Kappa : 0.0931
##
##  Mcnemar's Test P-Value : 0.07249
##
##             Sensitivity : 0.8333
##             Specificity : 0.2500
##          Pos Pred Value : 0.7042
##          Neg Pred Value : 0.4118
##              Prevalence : 0.6818
##          Detection Rate : 0.5682
##    Detection Prevalence : 0.8068
##       Balanced Accuracy : 0.5417
##
##        'Positive' Class : 0
##
```

```r
roc_lasso <- roc(as.numeric(test$DEATH_EVENT), as.vector(pred_prob))
plot(roc_lasso, main = "Lasso ROC")
```




```r
auc(roc_lasso)
```

```
## Area under the curve: 0.6619
```

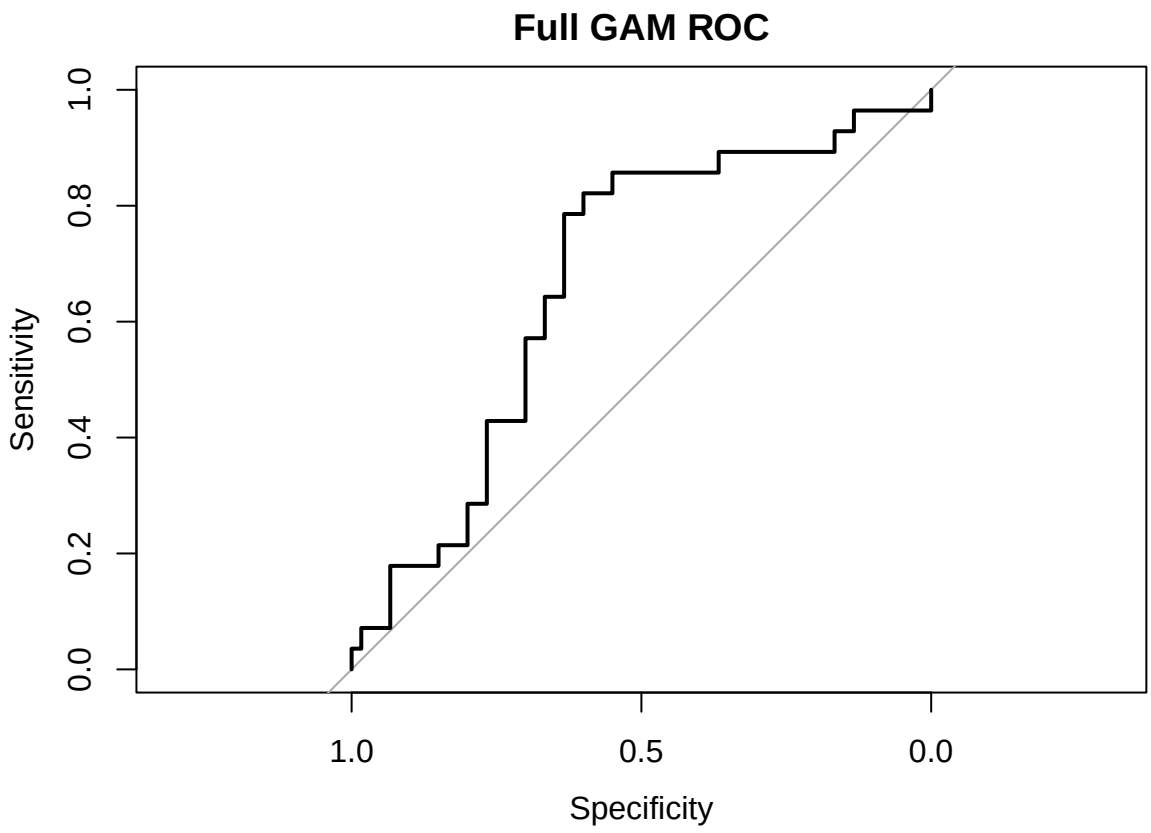
4.3 Generalized Additive Model (GAM)

```
gam_full <- gam(DEATH_EVENT ~
                  s(age) +
                  anaemia +
                  s(creatinine_phosphokinase) +
                  diabetes +
                  s(ejection_fraction) +
                  high_blood_pressure +
                  s(platelets) +
                  s(serum_creatinine) +
                  s(serum_sodium) +
                  sex +
                  smoking,
                family = binomial,
                data = train)

gam_full_pred <- predict(gam_full, newdata = test, type = "response")
gam_full_label <- ifelse(gam_full_pred > 0.5, 1, 0)
confusionMatrix(as.factor(gam_full_label), test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46 20
##          1 14  8
##
##                Accuracy : 0.6136
##                  95% CI : (0.5038, 0.7156)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.9297
##
##                   Kappa : 0.0556
##
##  Mcnemar's Test P-Value : 0.3912
##
##             Sensitivity : 0.7667
##             Specificity : 0.2857
##          Pos Pred Value : 0.6970
##          Neg Pred Value : 0.3636
##              Prevalence : 0.6818
##          Detection Rate : 0.5227
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.5262
##
##        'Positive' Class : 0
##
```

```
roc_gam_full <- roc(as.numeric(test$DEATH_EVENT), gam_full_pred)
plot(roc_gam_full, main = "Full GAM ROC")
```




```
auc(roc_gam_full)
```

```
## Area under the curve: 0.6708
```

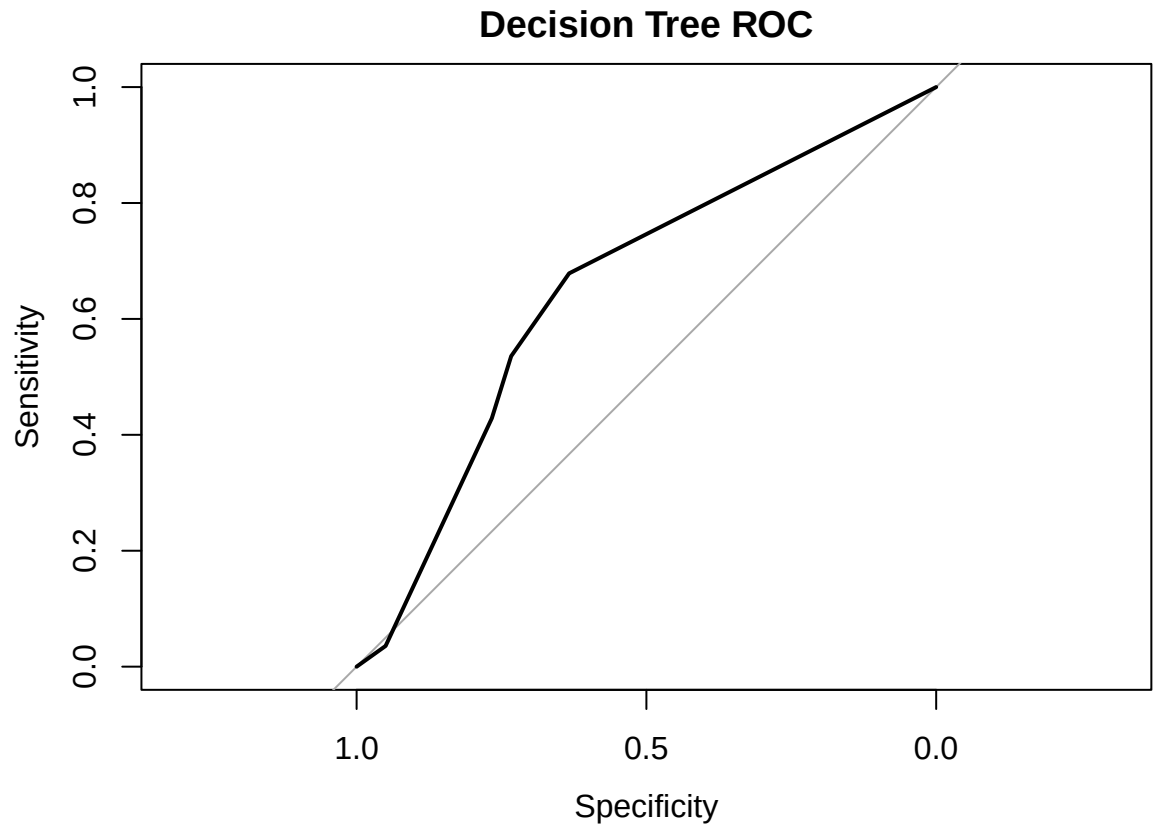
4.4 Decision Tree

```
tree_model <- rpart(DEATH_EVENT ~ ., data = train, method = "class")
tree_pred <- predict(tree_model, newdata = test, type = "class")
confusionMatrix(tree_pred, test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46 16
##          1 14 12
##
##                Accuracy : 0.6591
##                  95% CI : (0.5503, 0.7568)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.7194
##
##                   Kappa : 0.199
```

```
##
##  Mcnemar's Test P-Value : 0.8551
##
##             Sensitivity : 0.7667
##             Specificity : 0.4286
##          Pos Pred Value : 0.7419
##          Neg Pred Value : 0.4615
##              Prevalence : 0.6818
##          Detection Rate : 0.5227
##    Detection Prevalence : 0.7045
##       Balanced Accuracy : 0.5976
##
##        'Positive' Class : 0
##
```

```
tree_prob <- predict(tree_model, newdata = test)[,2]
roc_tree <- roc(as.numeric(test$DEATH_EVENT), tree_prob)
plot(roc_tree, main = "Decision Tree ROC")
```




```
auc(roc_tree)
```

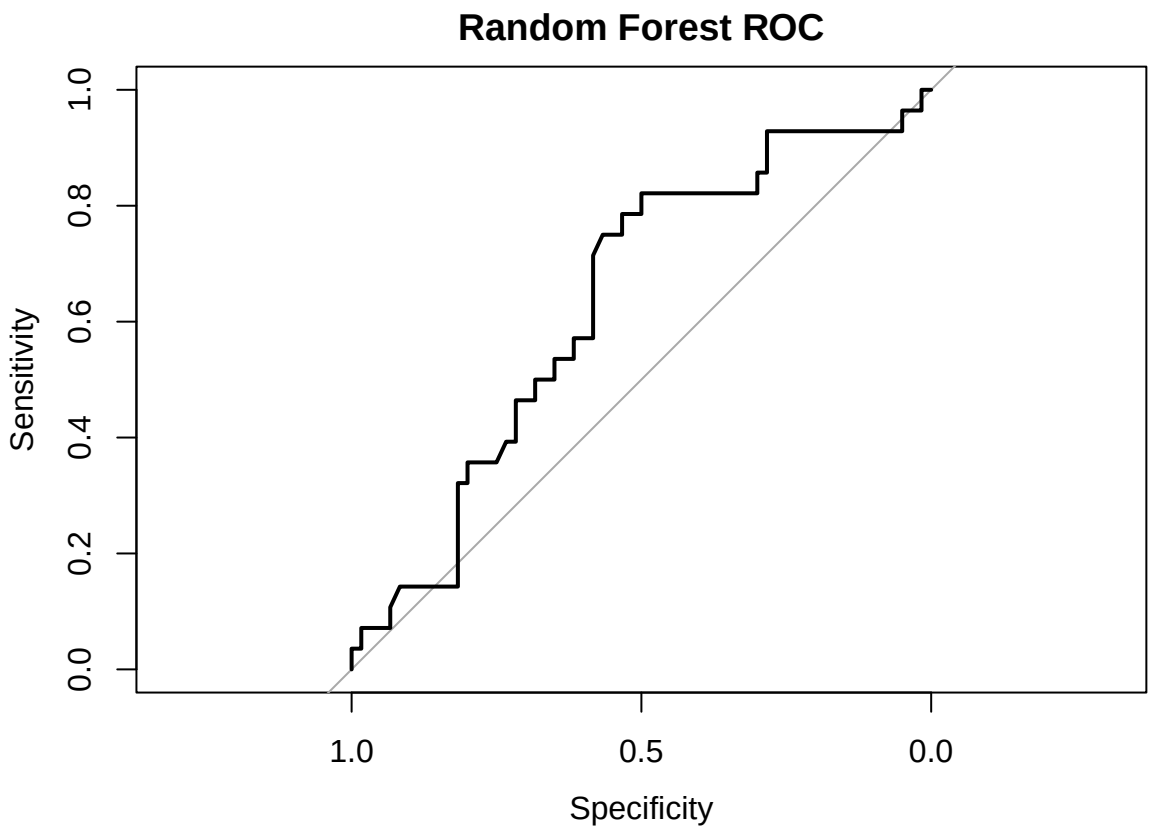
```
## Area under the curve: 0.6518
```

4.5 Random Forest

```
rf_model <- randomForest(DEATH_EVENT ~ ., data = train, importance = TRUE)
rf_pred <- predict(rf_model, newdata = test)
confusionMatrix(rf_pred, test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45 18
##          1 15 10
##
##                Accuracy : 0.625
##                  95% CI : (0.5153, 0.726)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.8947
##
##                   Kappa : 0.1103
##
##  Mcnemar's Test P-Value : 0.7277
##
##             Sensitivity : 0.7500
##             Specificity : 0.3571
##          Pos Pred Value : 0.7143
##          Neg Pred Value : 0.4000
##              Prevalence : 0.6818
##          Detection Rate : 0.5114
##    Detection Prevalence : 0.7159
##       Balanced Accuracy : 0.5536
##
##        'Positive' Class : 0
##
```

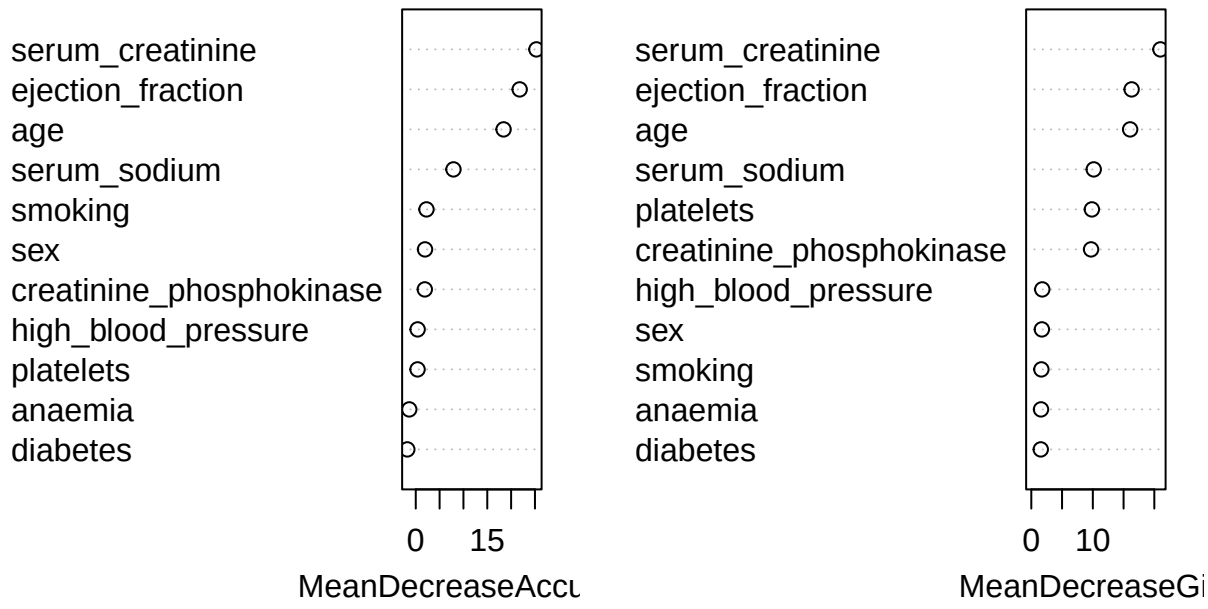
```
rf_prob <- predict(rf_model, newdata = test, type = "prob")[,2]
roc_rf <- roc(as.numeric(test$DEATH_EVENT), rf_prob)
plot(roc_rf, main = "Random Forest ROC")
```




```
auc(roc_rf)
```

```
## Area under the curve: 0.633
```

```
varImpPlot(rf_model)
```



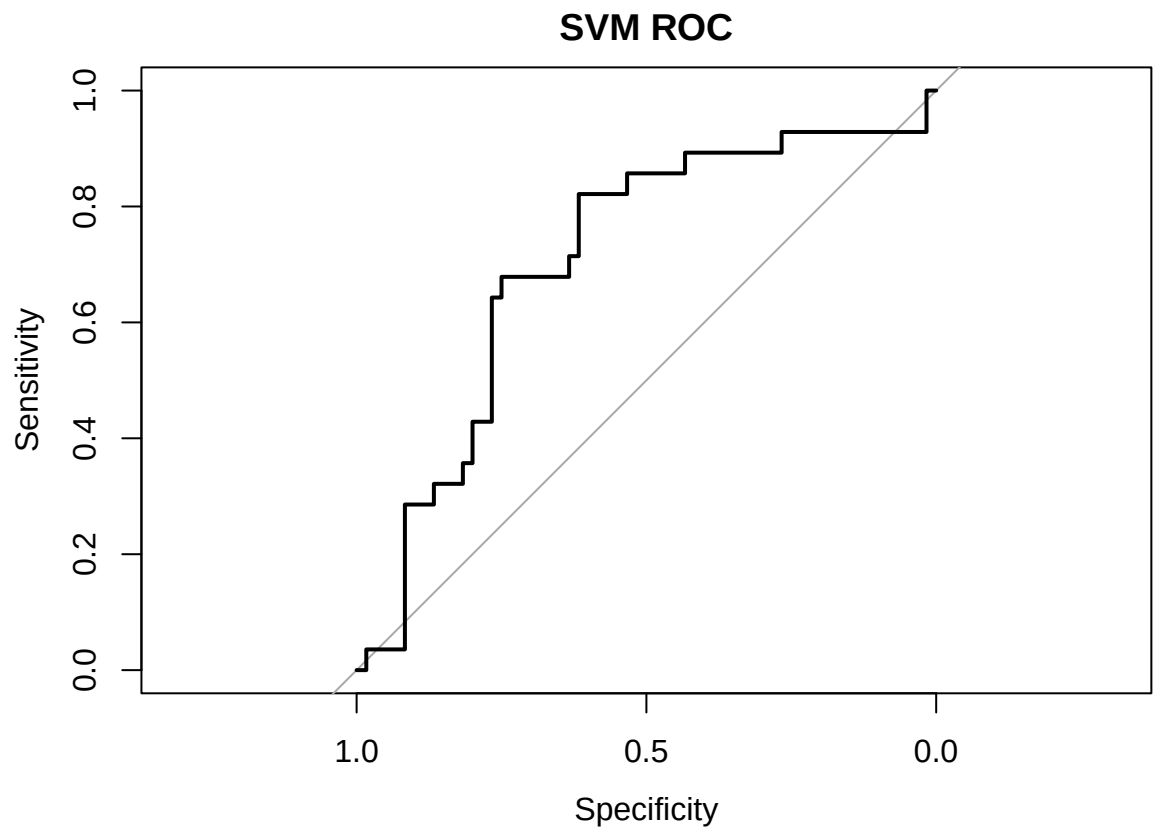
4.6 Support Vector Machine (SVM)

```
svm_model <- svm(DEATH_EVENT ~ ., data = train, kernel = "radial", probability = TRUE)
svm_pred <- predict(svm_model, newdata = test)
confusionMatrix(svm_pred, test$DEATH_EVENT)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 47 16
##          1 13 12
##
##                Accuracy : 0.6705
##                  95% CI : (0.5621, 0.767)
##     No Information Rate : 0.6818
##     P-Value [Acc > NIR] : 0.6388
##
##                   Kappa : 0.2181
##
##  Mcnemar's Test P-Value : 0.7103
##
##             Sensitivity : 0.7833
```

```
##             Specificity : 0.4286
##          Pos Pred Value : 0.7460
##          Neg Pred Value : 0.4800
##              Prevalence : 0.6818
##          Detection Rate : 0.5341
##    Detection Prevalence : 0.7159
##       Balanced Accuracy : 0.6060
##
##        'Positive' Class : 0
##
```

```r
svm_prob <- attr(predict(svm_model, newdata = test, probability = TRUE), "probabilities")[,2]
roc_svm <- roc(as.numeric(test$DEATH_EVENT), svm_prob)
plot(roc_svm, main = "SVM ROC")
```




```r
auc(roc_svm)
```

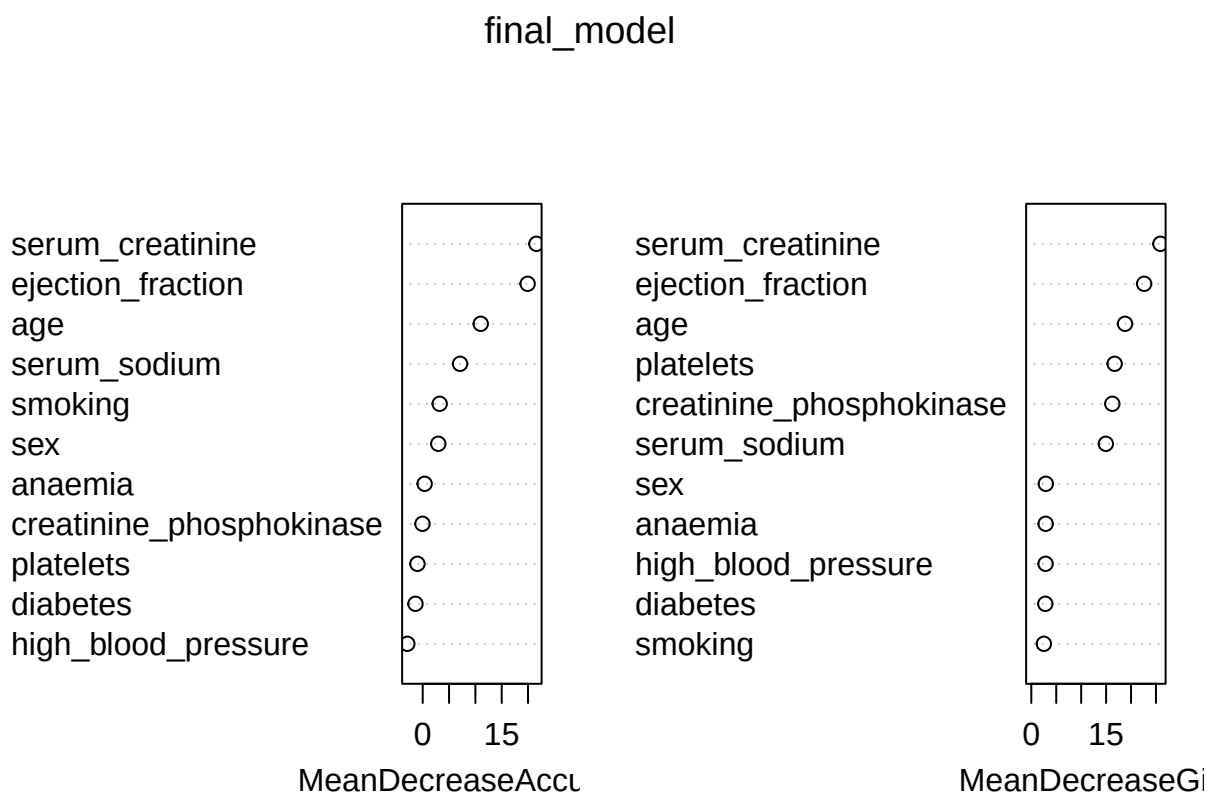
```
## Area under the curve: 0.7065
```

5. Final Model and Interpretation

While Lasso Logistic Regression showed slightly better predictive performance in terms of accuracy (64.8%) and AUC (0.6619), Random Forest was selected as the final model due to its flexibility in capturing nonlinear

patterns and its ability to provide intuitive variable importance measures. The clinical interpretability of top-ranked variables (e.g., serum creatinine, ejection fraction, age) was consistent across both models, but Random Forest offered a more robust and explanatory framework for real-world clinical applications.

```
final_model <- randomForest(DEATH_EVENT ~ ., data = data_nontime, importance = TRUE)
varImpPlot(final_model)
```



6. Conclusion

This project applied and compared multiple supervised learning models to predict survival in patients with heart failure.

Among the models, **Support Vector Machine (AUC = 0.7065)** achieved the highest AUC, followed by **Generalized Additive Model (GAM, AUC = 0.6708)** and **Lasso Logistic Regression (AUC = 0.6619)**.

However, **Random Forest** provided the best balance between **predictive performance**, **interpretability**, and **clinical relevance**. It consistently identified **serum creatinine**, **ejection fraction**, and **age** as key predictors—variables well-established in heart failure prognosis.

The full-variable GAM model, despite achieving a relatively high AUC, suffered from lower accuracy (61.4%), suggesting potential overfitting or limited added value from nonlinear terms in this setting.

While SVM achieved the best AUC, it lacks interpretability, which is a critical requirement in clinical decision-making.

In conclusion, we selected **Random Forest** as the final model due to its consistent performance, transparent variable importance ranking, and strong clinical interpretability. This model may serve as a useful tool for identifying high-risk patients and supporting medical decision-making in real-world practice.