Models

Xue Qin

2024-10-23

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.3

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

##
## Attaching package: 'pROC'

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

```
data <- read_csv("../data/raw-data/aids_clinical_trials_combined.csv")
```

```
## Rows: 2139 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (24): time, trt, age, wtkg, hemo, homo, drugs, karnof, oprior, z30, zpri...
##
## 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.
```

Logistic Regression

```
binary_cols <- c("hemo", "homo", "drugs", "karnof", "oprior", "z30", "zprior",
                "preanti", "race", "gender", "str2", "strat", "symptom",
                "treat", "offtrt")

# Survival variables (don't scale these)
survival_cols <- c("time", "cid")

# Get all column names
all_cols <- names(data)

# Identify continuous columns to scale
# (those not in binary_cols and not in survival_cols)
continuous_cols <- setdiff(all_cols, c(binary_cols, survival_cols))

# Create new dataframe
data_scaled <- data

# Scale only continuous variables
data_scaled[continuous_cols] <- scale(data[continuous_cols])

# Binary and survival variables remain unchanged
data_scaled[binary_cols] <- data[binary_cols]
data_scaled[survival_cols] <- data[survival_cols]
data_scaled_950<- data_scaled%>%
  filter(time<=950)

write.csv(data_scaled_950,"../data/derived-data/data_scaled_950.csv",row.names = FALSE)

library(caret)

set.seed(123)

trainIndex <- createDataPartition(data_scaled_950$cid, p = 0.7, list = FALSE)

train_data <- data_scaled_950[trainIndex, ]
test_data <- data_scaled_950[-trainIndex, ]
train_data$cid <- as.factor(train_data$cid)
test_data$cid <- as.factor(test_data$cid)
```

```
logistic_model <- glm(cid ~ age + trt + wtkg + hemo + homo + drugs + karnof+oprior+z30+zprior+preanti+ra
                   data = train_data, family = binomial)
model_summary<- summary(logistic_model)
```

```
test_predictions <- predict(logistic_model, newdata = test_data, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
test_pred_class <- ifelse(test_predictions > 0.5, 1, 0)
```

```r
confusionMatrix(factor(test_pred_class), factor(test_data$cid))
```

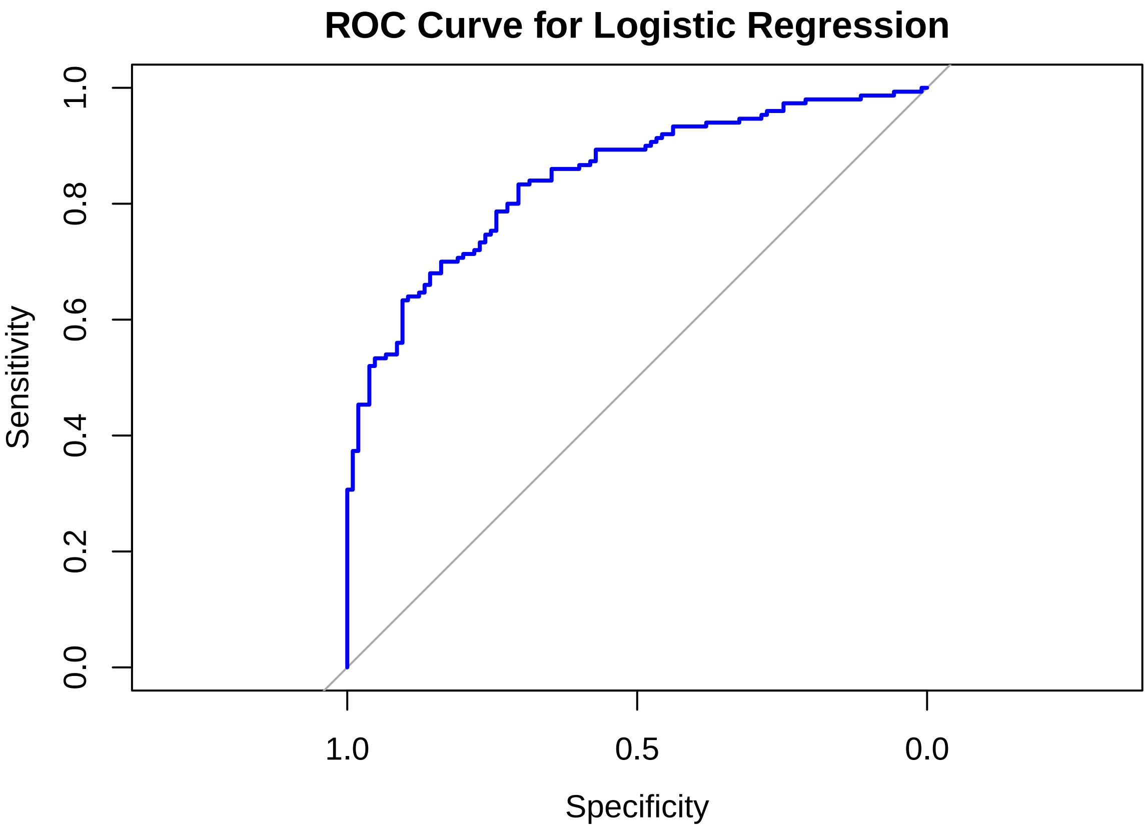
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  78  36
##          1  27 114
##
##                Accuracy : 0.7529
##                  95% CI : (0.6953, 0.8046)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 2.634e-08
##
##                   Kappa : 0.4965
##
##  Mcnemar's Test P-Value : 0.3135
##
##             Sensitivity : 0.7429
##             Specificity : 0.7600
##          Pos Pred Value : 0.6842
##          Neg Pred Value : 0.8085
##              Prevalence : 0.4118
##          Detection Rate : 0.3059
##    Detection Prevalence : 0.4471
##       Balanced Accuracy : 0.7514
##
##        'Positive' Class : 0
##
```

```r
# Roc Curve

roc_curve <- roc(test_data$cid, test_predictions)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(roc_curve, col = "blue", main = "ROC Curve for Logistic Regression")
```



```
auc(roc_curve)
```

```
## Area under the curve: 0.845
```

```
p_values <- model_summary$coefficients[, 4]

significant_vars <- names(p_values[p_values < 0.05])

print(significant_vars)
```

```
## [1] "age"     "hemo"    "drugs"   "preanti" "race"    "offtrt"  "cd420"
## [8] "cd820"
```

fit with selected features with P<0.05

```
logistic_model2 <- glm(cid ~ karnof+preanti+symptom+treat+cd40+cd420+cd820,
                      data = train_data, family = binomial)

# Summary of the model
model_summary2<- summary(logistic_model2)
```

```
test_predictions2 <- predict(logistic_model2, newdata = test_data, type = "response")

test_pred_class2 <- ifelse(test_predictions2 > 0.5, 1, 0)

confusionMatrix(factor(test_pred_class2), factor(test_data$cid))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0  68  28
##          1  37 122
##
##                Accuracy : 0.7451
##                  95% CI : (0.687, 0.7974)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 1.179e-07
##
##                   Kappa : 0.467
##
##  Mcnemar's Test P-Value : 0.3211
##
##             Sensitivity : 0.6476
##             Specificity : 0.8133
##          Pos Pred Value : 0.7083
##          Neg Pred Value : 0.7673
##              Prevalence : 0.4118
##          Detection Rate : 0.2667
##    Detection Prevalence : 0.3765
##       Balanced Accuracy : 0.7305
##
##        'Positive' Class : 0
##
```

Lasso

```
library(glmnet)
```

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

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

```
X <- model.matrix(cid ~ age + trt + wtkg + hemo + homo + drugs + karnof + oprior + z30 + zprior + prean
y <- data_scaled_950$cid

lasso_model <- glmnet(X, y, family = "binomial", alpha = 1)


cv_lasso <- cv.glmnet(X, y, family = "binomial", alpha = 1)


best_lambda <- cv_lasso$lambda.min

coef(cv_lasso, s = "lambda.min")
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)  1.3664899627
## age          0.2412310076
## trt          .
## wtkg          .           
## hemo        -0.5386997533
## homo          .           
## drugs       -0.4377875085
## karnof      -0.0064280283
## oprior       0.0834176549
## z30          0.0790940635
## zprior        .           
## preanti      0.0006121674
## race        -0.7437455386
## gender       0.3723106804
## str2          .           
## strat        0.1255874934
## symptom      0.5319893666
## treat       -0.1991194205
## offtrt      -1.9331766778
## cd40         0.0782609922
## cd420       -1.1160426646
## cd80         0.0962111514
## cd820        0.2581837543
```

fit model with selected feature

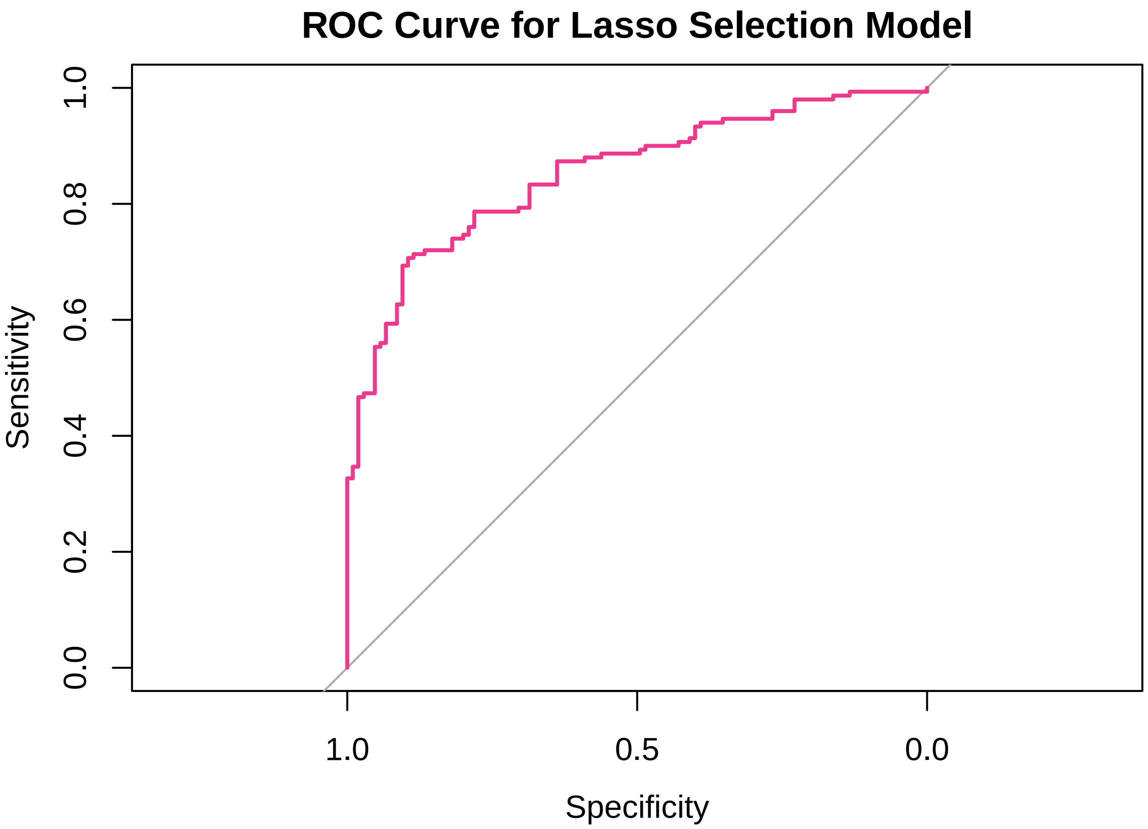
```
X_train <- model.matrix(cid ~ offtrt + cd420 + race + hemo + symptom + drugs + gender + cd820 + age + s
X_test <- model.matrix(cid ~ offtrt + cd420 + race + hemo + symptom + drugs + gender + cd820 + age + st


y_train <- train_data$cid
y_test <- test_data$cid
cv_lasso <- cv.glmnet(X_train, y_train, family = "binomial", alpha = 1)

best_lambda <- cv_lasso$lambda.min

lasso_model <- glmnet(X_train, y_train, family = "binomial", alpha = 1, lambda = best_lambda)

test_predictions_lasso <- predict(lasso_model, newx = X_test, type = "response")

roc_curve_lasso <- roc(test_data$cid, test_predictions_lasso)
```

```
## Setting levels: control = 0, case = 1
```

```
## Warning in roc.default(test_data$cid, test_predictions_lasso): Deprecated use a
## matrix as predictor. Unexpected results may be produced, please pass a numeric
## vector.
```

```
## Setting direction: controls < cases
```

```
plot(roc_curve_lasso, main = "ROC Curve for Lasso Selection Model", col="violetred2")
```



ROC Curve for Lasso Selection Model

```
auc(roc_curve_lasso)
```

```
## Area under the curve: 0.8523
```

```
test_pred_class_lasso <- ifelse(test_predictions_lasso > 0.5, 1, 0)
# Confusion Matrix
confusionMatrix(factor(test_pred_class_lasso), factor(y_test))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  82  35
##          1  23 115
##
##                Accuracy : 0.7725
##                  95% CI : (0.7161, 0.8225)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 4.321e-10
##
##                   Kappa : 0.5384
##
##  Mcnemar's Test P-Value : 0.1486
##
##             Sensitivity : 0.7810
##             Specificity : 0.7667
##          Pos Pred Value : 0.7009
##          Neg Pred Value : 0.8333
##              Prevalence : 0.4118
##          Detection Rate : 0.3216
##    Detection Prevalence : 0.4588
##       Balanced Accuracy : 0.7738
## 
##        'Positive' Class : 0
## 
```

```
print(coef(lasso_model))
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept)  0.4814170
## offtrt      -2.1018493
## cd420       -1.1764704
## race        -0.7538408
## hemo        -0.7099865
## symptom      0.4233906
## drugs       -0.5224033
## gender       0.3986055
## cd820        0.3687365
## age          0.3094244
## strat        0.3432900
```

Backward Selection

```
# Perform backward selection
backward_model <- step(logistic_model, direction = "backward", trace = FALSE)

backward_model_summary <- summary(backward_model)

test_predictions_backward <- predict(backward_model, newdata = test_data, type = "response")
test_pred_class_backward <- ifelse(test_predictions_backward > 0.5, 1, 0)

conf_matrix <- confusionMatrix(factor(test_pred_class_backward), factor(test_data$cid))
print("Confusion Matrix and Performance Metrics for Backward Selection Model:")
```

```
## [1] "Confusion Matrix and Performance Metrics for Backward Selection Model:"
```

```
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  79  35
##          1  26 115
## 
##                Accuracy : 0.7608
##                  95% CI : (0.7036, 0.8118)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 5.421e-09
## 
##                   Kappa : 0.5125
## 
```

```
##  Mcnemar's Test P-Value : 0.3057
##
##             Sensitivity : 0.7524
##             Specificity : 0.7667
##          Pos Pred Value : 0.6930
##          Neg Pred Value : 0.8156
##              Prevalence : 0.4118
##          Detection Rate : 0.3098
##    Detection Prevalence : 0.4471
##       Balanced Accuracy : 0.7595
##
##        'Positive' Class : 0
##
```

```
# Get the selected features
selected_features <- names(coef(backward_model))
print("Features Selected by Backward Selection:")
```

```
## [1] "Features Selected by Backward Selection:"
```

```
print(selected_features)
```

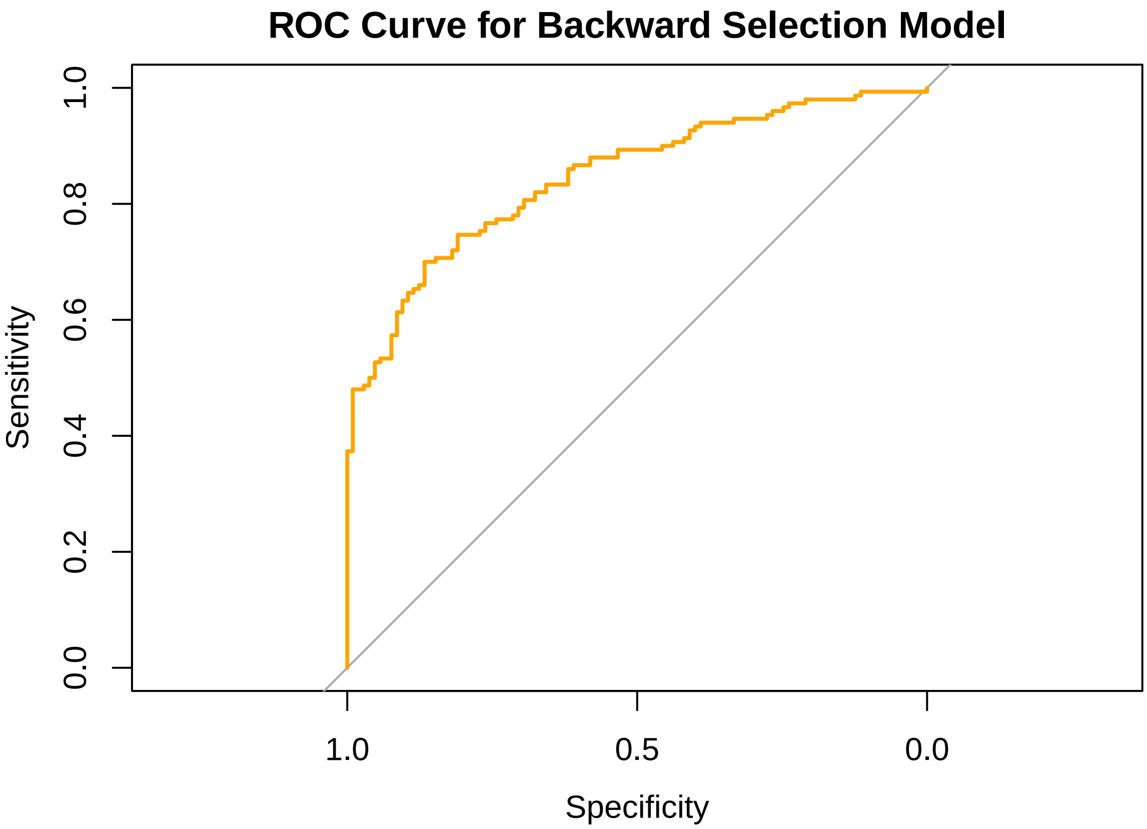
```
##  [1] "(Intercept)" "age"         "hemo"        "drugs"       "preanti"
##  [6] "race"        "gender"      "symptom"     "offtrt"      "cd420"
## [11] "cd820"
```

```
# Plot ROC curve for backward selection model
roc_curve_backward <- roc(test_data$cid, test_predictions_backward)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roc_curve_backward, main = "ROC Curve for Backward Selection Model", col="orange")
```



```
auc(roc_curve_backward)
```

```
## Area under the curve: 0.8465
```

```
# Print coefficients of the final model
print("Coefficients of the Final Model:")
```

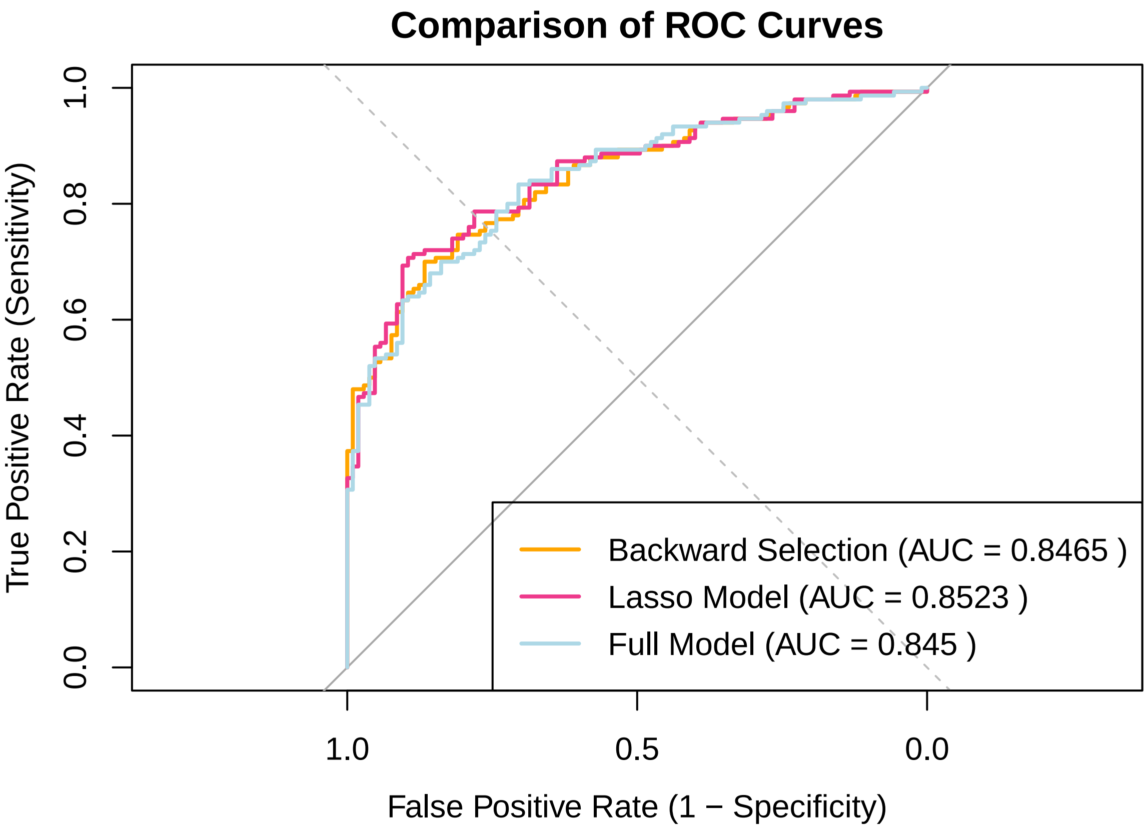
```
## [1] "Coefficients of the Final Model:"
```

```
print(coef(backward_model))
```

```
##   (Intercept)           age          hemo         drugs       preanti
##  0.8632302848  0.3129571806 -0.8787791179 -0.5721991108  0.0009780816
##          race        gender       symptom        offtrt         cd420
## -0.7719568158  0.4265073018  0.4871033437 -2.2108410758 -1.2404869416
##         cd820
##  0.4021572669
```

```
plot(roc_curve_backward, col = "orange", main = "Comparison of ROC Curves",
     xlab = "False Positive Rate (1 - Specificity)",
     ylab = "True Positive Rate (Sensitivity)")

lines(roc_curve_lasso,col="violetred2")
# Add the second ROC curve
lines(roc_curve, col = "lightblue")

# Add diagonal reference line
abline(a = 0, b = 1, lty = 2, col = "gray")
```

```
# Add legend
legend("bottomright",
       legend = c(paste("Backward Selection (AUC =", round(auc(roc_curve_backward), 4), ")"),
                  paste("Lasso Model (AUC =", round(auc(roc_curve_lasso), 4), ")"),
                  paste("Full Model (AUC =", round(auc(roc_curve), 4), ")")),
       col = c("orange", "violetred2","lightblue"),
       lwd = 2)
```


Comparison of ROC Curves
True Positive Rate (Sensitivity)
False Positive Rate (1 − Specificity)
Backward Selection (AUC = 0.8465)
Lasso Model (AUC = 0.8523)
Full Model (AUC = 0.845)