Assessing Default Risk in Credit Card Payments: A Statistical Approach

Binitha Chandrasena

04 December, 2024

Packages and loading dataset

```
library(caret)
library(glmnet)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(viridis)
library(class)
library(gridExtra)

rm(list=ls())
data <- read.csv("default of credit card clients .csv", skip = 1)
```

```
#str(data)
```

Exploring the data

```
#setting the variables types accordingly
data$default <- factor(data$default)
data$AGE <- as.numeric(data$AGE)
data$SEX <- factor(data$SEX)
```

Box plot for sex and default

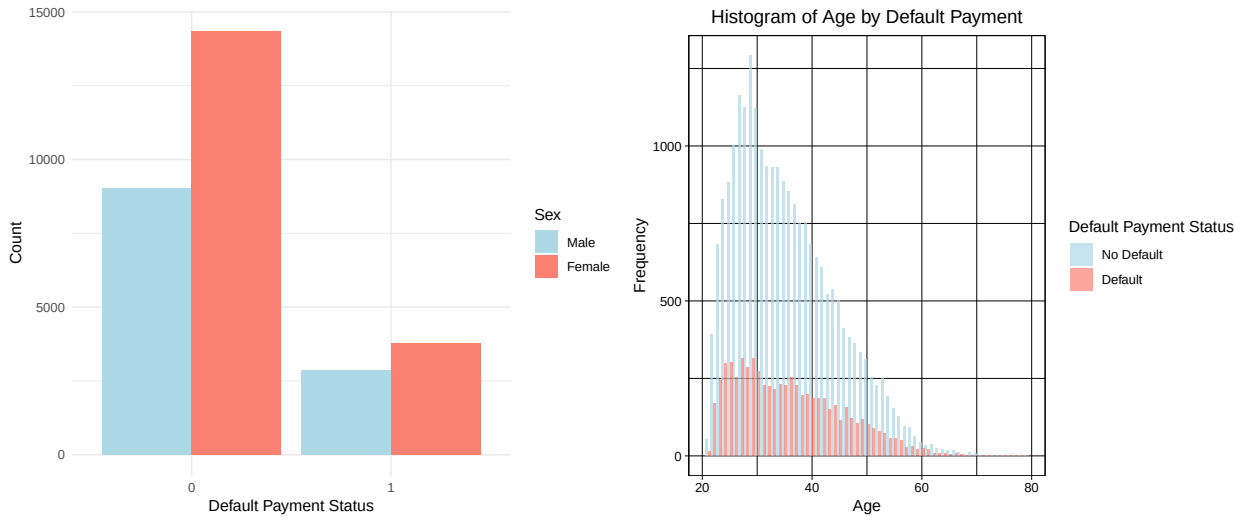
```
plot_1 <- ggplot(data, aes(x = default, fill = factor(SEX))) +
  geom_bar(position = "dodge") +
  labs(x = "Default Payment Status",
       y = "Count",
       fill = "Sex") +
  scale_fill_manual(values = c("1" = "lightblue", "2" = "salmon"),
                    labels = c("Male", "Female")) +
  theme_minimal()
```

```
plot_1
```

Histogram of Age by Default Status

```
plot_2 <- ggplot(data, aes(x = AGE, fill = default)) +
  geom_histogram(binwidth = 1, position = "dodge", alpha = 0.7) +
  labs(x = "Age", y = "Frequency", fill = "Default Payment Status", title = "Histogram of Age by Default
  scale_fill_manual(values = c("0" = "lightblue", "1" = "salmon"),
                    labels = c("No Default", "Default")) +
  theme_linedraw() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
grid.arrange(plot_1, plot_2, ncol = 2)
```



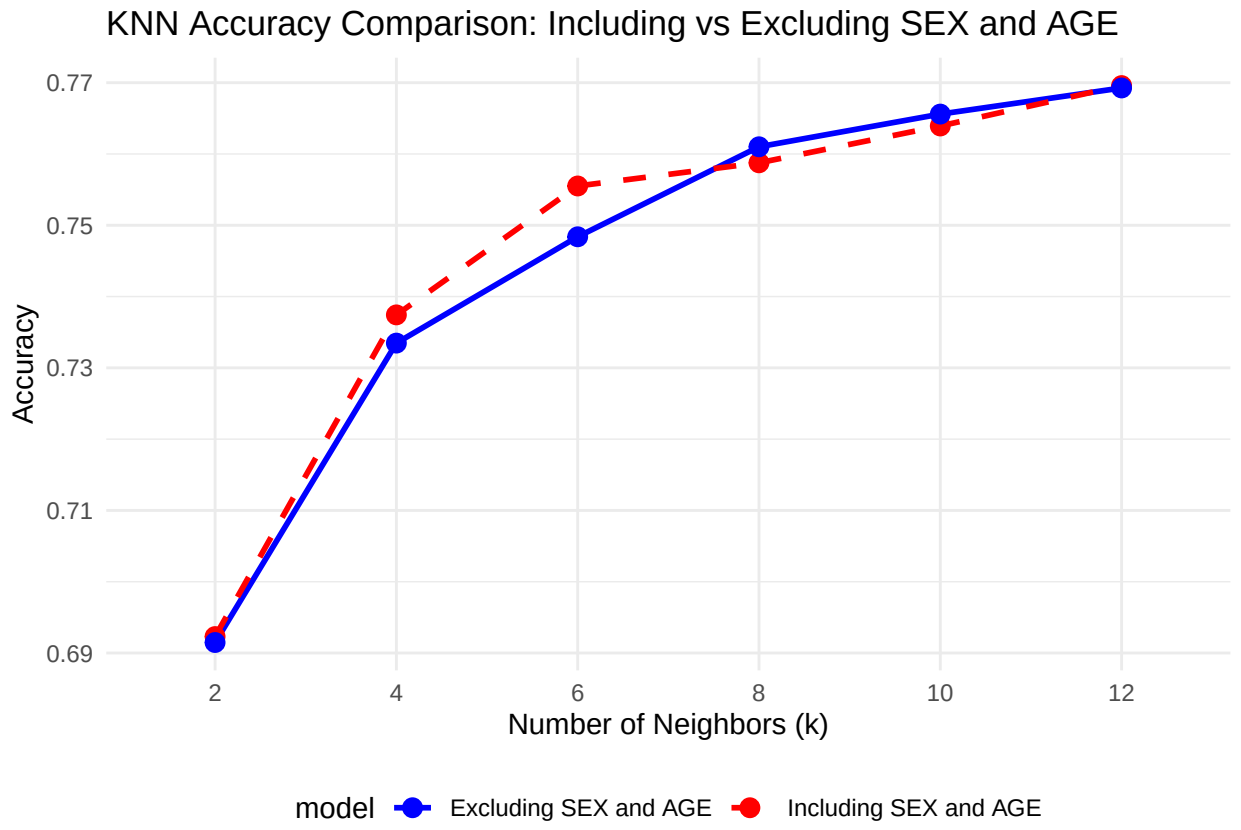
Cleaning and splitting the data for Training and Testing

```
data <- na.omit(data)
data <- data %>% select(-ID)
names(data)[names(data) == 'default.payment.next.month'] <- 'default'
columns_to_factor <- c("SEX", "EDUCATION", "MARRIAGE", "PAY_0", "PAY_2", "PAY_3",
                       "PAY_4", "PAY_5", "PAY_6", "default")
data[columns_to_factor] <- lapply(data[columns_to_factor], as.factor)

#str(data)

X <- data %>% select(-default)
y <- data$default

set.seed(12340)

train_index <- createDataPartition(y, p = 0.8, list = FALSE)
X_train <- X[train_index, ]
X_test <- X[-train_index, ]
y_train <- y[train_index]
y_test <- y[-train_index]
```

Model Prediction 1 (KNN)

```
# KNN model with SEX and AGE
knn_cv_all <- train(
  x = X_train,
  y = y_train,
  method = "knn",
  tuneGrid = data.frame(k = c(2, 4, 6, 8, 10, 12)),
  trControl = trainControl(method = "cv", number = 5)
```

```
)

# KNN model excluding SEX and AGE
X_train_knn_excl <- X_train %>% select(-SEX, -AGE)
X_test_knn_excl <- X_test %>% select(-SEX, -AGE)

knn_cv_excl <- train(
  x = X_train_knn_excl,
  y = y_train,
  method = "knn",
  tuneGrid = data.frame(k = c(2, 4, 6, 8, 10, 12)),
  trControl = trainControl(method = "cv", number = 5)
)

accuracy_all <- knn_cv_all$results$Accuracy
k_values_all <- knn_cv_all$results$k

accuracy_excl <- knn_cv_excl$results$Accuracy
k_values_excl <- knn_cv_excl$results$k

accuracy_df <- data.frame(
  k = rep(c(2, 4, 6, 8, 10, 12), 2),
  accuracy = c(accuracy_all, accuracy_excl),
  model = rep(c("Including SEX and AGE", "Excluding SEX and AGE"), each = 6)
)

ggplot(accuracy_df, aes(x = factor(k), y = accuracy, color = model, group = model)) +
  geom_line(aes(linetype = model), size = 1) +
  geom_point(size = 3) +
  labs(x = "Number of Neighbors (k)",
       y = "Accuracy",
       title = "KNN Accuracy Comparison: Including vs Excluding SEX and AGE") +
  scale_color_manual(values = c("blue", "red")) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

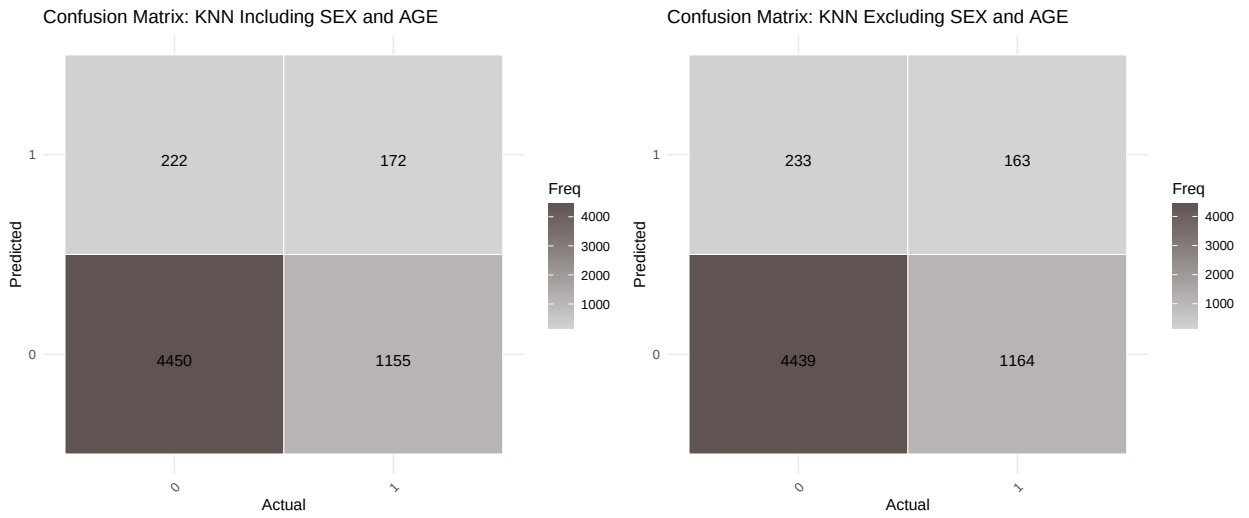



```r
set.seed(11111)

pred_all <- predict(knn_cv_all, newdata = X_test)
pred_excl <- predict(knn_cv_excl, newdata = X_test_knn_excl)

accuracy_all_test <- sum(pred_all == y_test) / length(y_test) * 100
accuracy_excl_test <- sum(pred_excl == y_test) / length(y_test) * 100

cat("Accuracy of KNN model including SEX and AGE on test set:", accuracy_all_test, "%\n")
## Accuracy of KNN model including SEX and AGE on test set: 77.04617 %
cat("Accuracy of KNN model excluding SEX and AGE on test set:", accuracy_excl_test, "%\n")
## Accuracy of KNN model excluding SEX and AGE on test set: 76.71279 %

cm_all <- confusionMatrix(pred_all, y_test)
cat("\nConfusion Matrix for KNN model including SEX and AGE:\n")
##
## Confusion Matrix for KNN model including SEX and AGE:
print(cm_all)
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4450 1155
##          1  222  172
##
##                Accuracy : 0.7705
```

```
##                  95% CI : (0.7596, 0.7811)
##     No Information Rate : 0.7788
##     P-Value [Acc > NIR] : 0.9414
##
##                   Kappa : 0.1097
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9525
##             Specificity : 0.1296
##          Pos Pred Value : 0.7939
##          Neg Pred Value : 0.4365
##              Prevalence : 0.7788
##          Detection Rate : 0.7418
##    Detection Prevalence : 0.9343
##       Balanced Accuracy : 0.5410
##
##        'Positive' Class : 0
##
cm_excl <- confusionMatrix(pred_excl, y_test)
cat("\nConfusion Matrix for KNN model excluding SEX and AGE:\n")
##
## Confusion Matrix for KNN model excluding SEX and AGE:
print(cm_excl)
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4439 1164
##          1  233  163
##
##                Accuracy : 0.7671
##                  95% CI : (0.7562, 0.7778)
##     No Information Rate : 0.7788
##     P-Value [Acc > NIR] : 0.9855
##
##                   Kappa : 0.0974
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9501
##             Specificity : 0.1228
##          Pos Pred Value : 0.7923
##          Neg Pred Value : 0.4116
##              Prevalence : 0.7788
##          Detection Rate : 0.7400
##    Detection Prevalence : 0.9340
##       Balanced Accuracy : 0.5365
##
##        'Positive' Class : 0
##

cm_all_df <- as.data.frame(as.table(cm_all))
```

```
cm_excl_df <- as.data.frame(as.table(cm_excl))

plot_3 <- ggplot(cm_all_df, aes(x = Reference, y = Prediction)) +
  geom_tile(aes(fill = Freq), color = "white") +
  scale_fill_gradient(low = "lightgrey", high = "#5F5451") +
  geom_text(aes(label = Freq), vjust = 1) +
  labs(title = "Confusion Matrix: KNN Including SEX and AGE",
       x = "Actual", y = "Predicted") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_4 <- ggplot(cm_excl_df, aes(x = Reference, y = Prediction)) +
  geom_tile(aes(fill = Freq), color = "white") +
  scale_fill_gradient(low = "lightgrey", high = "#5F5451") +
  geom_text(aes(label = Freq), vjust = 1) +
  labs(title = "Confusion Matrix: KNN Excluding SEX and AGE",
       x = "Actual", y = "Predicted") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

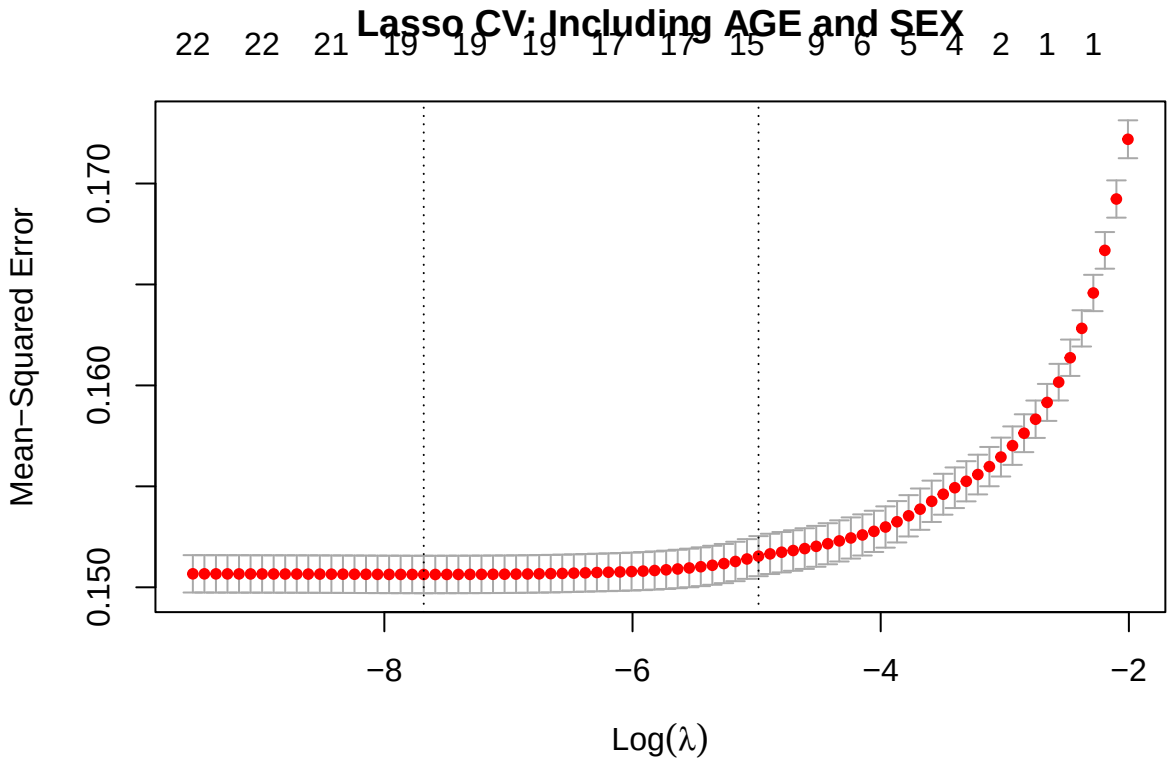
```
set.seed(111)
grid.arrange(plot_3, plot_4, ncol = 2)
```



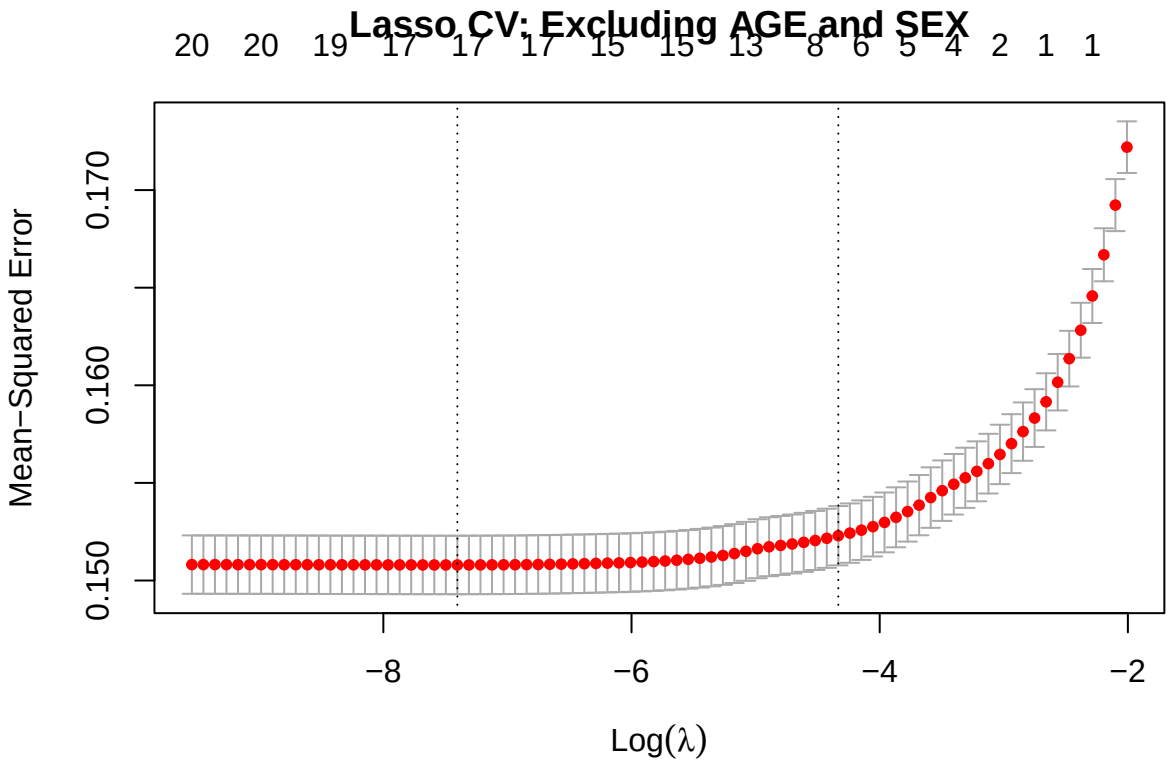
Lasso Regression with Cross Validation

```
columns_to_factor <- c("SEX", "EDUCATION", "MARRIAGE", "PAY_0", "PAY_2", "PAY_3",
                       "PAY_4", "PAY_5", "PAY_6")
X_train[columns_to_factor] <- lapply(X_train[columns_to_factor], function(x) as.numeric(as.factor(x)))
X_test[columns_to_factor] <- lapply(X_test[columns_to_factor], function(x) as.numeric(as.factor(x)))

y_train_numeric <- as.numeric(y_train)
y_test_numeric <- as.numeric(y_test)
```

```r
X_train_scaled_incl <- scale(X_train)
X_test_scaled_incl <- scale(X_test)
cv_incl <- cv.glmnet(x = as.matrix(X_train_scaled_incl),
                     y = y_train_numeric,
                     alpha = 1,
                     nfolds = 10)
plot_5 <- plot(cv_incl, main = "Lasso CV: Including AGE and SEX")
```




```r
X_train_excl <- X_train %>% select(-SEX, -AGE)
X_test_excl <- X_test %>% select(-SEX, -AGE)
X_train_scaled_excl <- scale(X_train_excl)
X_test_scaled_excl <- scale(X_test_excl)
cv_excl <- cv.glmnet(x = as.matrix(X_train_scaled_excl),
                     y = y_train_numeric,
                     alpha = 1,
                     nfolds = 10)

plot_6 <- plot(cv_excl, main = "Lasso CV: Excluding AGE and SEX")
```




```
lasso_test_incl= predict(cv_incl, newx = as.matrix(X_test_scaled_incl), s=cv_incl$lambda.min)
lasso_test_excl= predict(cv_excl, newx = as.matrix(X_test_scaled_excl), s=cv_excl$lambda.min)

mse_incl <- mean((y_test_numeric - lasso_test_incl)^2)
mse_excl <- mean((y_test_numeric - lasso_test_excl)^2)

lasso_mpse_incl = mean((y_test_numeric-lasso_test_incl)^2)
lasso_mpse_excl = mean((y_test_numeric-lasso_test_excl)^2)

cv_summary_stats_min_incl = c("Model (Including SEX and AGE)", mse_incl, lasso_mpse_incl)
cv_summary_stats_min_excl = c("Model (Excluding SEX and AGE)", mse_excl, lasso_mpse_excl)

comparison_table = c("model type", "MSE", "Test MSPE")
print(data.frame(cbind(comparison_table, cv_summary_stats_min_incl, cv_summary_stats_min_excl)))
##   comparison_table     cv_summary_stats_min_incl     cv_summary_stats_min_excl
## 1       model type Model (Including SEX and AGE) Model (Excluding SEX and AGE)
## 2              MSE               0.1504124768989             0.150557430296126
## 3        Test MSPE               0.1504124768989             0.150557430296126
```