Appendix –

R-codes

```r
install.packages("MASS")
library(MASS)
library(leaps)
library(glm)
library(boot)
library(caret)
data(Pima.tr)

dim(Pima.tr)
data(Pima.te)
head(Pima.te)
data(Pima.te)
dim(Pima.te)
head(Pima.te)

#Cleaning the data
comb_data <- rbind(Pima.tr, Pima.te)
dim(comb_data)

any(is.na(comb_data))
colSums(is.na(comb_data))
comb_data$type <- as.factor(comb_data$type)

comb_data$type <- ifelse(comb_data$type == "Yes", 1, 0)
head(comb_data)

#####EDA#####

summary(comb_data)

#Histograms

par(mfrow=c(3,3))
hist(comb_data$npreg, main="Distribution of Number of Pregnancies", xlab="Number of Pregnancies", col="lightblue", border="black
hist(comb_data$glu, main="Distribution of Glucose Levels", xlab="Glucose Levels", col="lightblue", border="black")
hist(comb_data$bp, main="Distribution of Blood Pressure", xlab="Blood Pressure", col="lightblue", border="black")
hist(comb_data$skin, main="Distribution of Skin Fold Thickness", xlab="Skin Fold Thickness", col="lightblue", border="black")
hist(comb_data$bmi, main="Distribution of BMI", xlab="BMI", col="lightblue", border="black")
hist(comb_data$ped, main="Distribution of Pedigree Function", xlab="Pedigree Function", col="lightblue", border="black") # Fixed
hist(comb_data$age, main="Distribution of Age", xlab="Age", col="lightblue", border="black")

#counting the the type variable
table(comb_data$type)
```

```r

#counting the the type variable
table(comb_data$type)

#plotting the scatterplot matirx

# Exclude the 'type' column as it's a binary categorical variable
pairs(comb_data[], pch = 19, col = ifelse(comb_data$type == 1, "red", "blue"))

#correlation chart

cor(comb_data)

#checking the mean values for the independent variables in every column

mean_npreg_by_type <- tapply(comb_data$npreg, comb_data$type, mean)
mean_glu_by_type <- tapply(comb_data$glu, comb_data$type, mean)
mean_bp_by_type <- tapply(comb_data$bp, comb_data$type, mean)
mean_skin_by_type <- tapply(comb_data$skin, comb_data$type, mean)
mean_bmi_by_type <- tapply(comb_data$bmi, comb_data$type, mean)
mean_ped_by_type <- tapply(comb_data$ped, comb_data$type, mean)
mean_age_by_type <- tapply(comb_data$age, comb_data$type, mean)

cat("Mean of Number of Pregnancies by Type:", "\n")
print(mean_npreg_by_type)

cat("\nMean of Glucose Levels by Type:", "\n")
print(mean_glu_by_type)

cat("\nMean of Blood Pressure by Type:", "\n")
print(mean_bp_by_type)

cat("\nMean of Skin Fold Thickness by Type:", "\n")
print(mean_skin_by_type)

cat("\nMean of BMI by Type:", "\n")
print(mean_bmi_by_type)

cat("\nMean of Diabetes Pedigree Function by Type:", "\n")
print(mean_ped_by_type)

cat("\nMean of Age by Type:", "\n")
print(mean_age_by_type)
```

```r
cat("\nMean of Age by Type:", "\n")
print(mean_age_by_type)

######Building our models######

# Constructing logistic regression model

lr_model <- glm(type ~ ., data=comb_data, family=binomial)
summary(lr_model)


#####Best Subset ######

full.model <- glm(type ~ ., data = comb_data, family = binomial)
subsets <- regsubsets(type ~ ., data = comb_data, method="exhaustive")

results <- summary(subsets)

bic = results$bic
which.min(bic)
aic = results$cp
which.min(aic)

coef(subsets, which.min(bic))

coef(subsets, which.min(aic))

aic_values = best_models$cp
bic_values = best_models$bic

#K-cross

k_values <- 1:length(aic_values)

optimal_k_bic <- 4
optimal_k_aic <- 5


cv.error <- function(data, indices, size) {
  train <- data[indices, ]
  test <- data[-indices, ]
  fit <- glm(type ~ ., data=train, family=binomial, subset = size)
  preds <- predict(fit, newdata=test, type="response")
  class_preds <- ifelse(preds > 0.5, 1, 0)
  return(mean(class_preds != test$type))
```

```r
set.seed(123)
k <- 10  #setting the number of folds
folds <- sample(1:k, nrow(comb_data), replace=TRUE)
cv.errors <- matrix(NA, k, ncol(comb_data)-1)

for(j in 1:k){
  for(i in 1:(ncol(comb_data)-1)){
    cv.errors[j, i] <- cv.error(comb_data, which(folds != j), i)
  }
}
mean.cv.errors <- apply(cv.errors, 2, mean)
optimal.number <- which.min(mean.cv.errors)
cat("Optimal number of predictors:", optimal.number, "\n")

best.subset <- regsubsets(type ~ ., data=comb_data, nvmax=ncol(comb_data)-1)
coef(best.subset, optimal.number)


mean.cv.errors <- apply(cv.errors, 2, mean)

# Plotting

par(mfrow=c(3,1))# 2 rows, 1 column

# Plot AIC
plot(k_values, aic_values, type="b", col="blue", ylab="AIC Value", xlab="k (Number of Variables)", main="AIC vs. k")
points(optimal_k_aic, aic_values[optimal_k_aic], col="red", pch=19, cex=2)
text(optimal_k_aic, aic_values[optimal_k_aic], labels="Optimal k", pos=3, col="blue")

# Plot BIC
plot(k_values, bic_values, type="b", col="green", ylab="BIC Value", xlab="k (Number of Variables)", main="BIC vs. k")
points(optimal_k_bic, bic_values[optimal_k_bic], col="red", pch=19, cex=2)
text(optimal_k_bic, bic_values[optimal_k_bic], labels="Optimal k", pos=4, col="green")
plot(1:length(mean.cv.errors), mean.cv.errors, type="b",
     xlab="Number of Predictors", ylab="Mean CV Error",
     main="Cross-validation Error vs. Subset Size", pch=19, col="red")

points(optimal.number, mean.cv.errors[optimal.number], col="blue", pch=4, lwd=3)

#####LDA model#####

predictors <- c("npreg", "glu", "bmi", "ped")

lda.model <- lda(type ~ npreg + glu + bmi + ped, data=comb_data)
print(summary(lda.model))
print(lda.model$means)
```

```
predictors <- c("npreg", "glu", "bmi", "ped")

lda.model <- lda(type ~ npreg + glu + bmi + ped, data=comb_data)
print(summary(lda.model))
print(lda.model$means)

############QDA MODEL########
qda.model <- qda(type ~ npreg + glu + bmi + ped, data = comb_data)

print(summary(qda.model))
print(qda.model$means)


#####Plotting confusion matirx#####

comb_data$type <- as.factor(comb_data$type)

#Train Models:
control <- trainControl(method="cv", number=10, savePredictions="final")



#Logistic Regression:
logistic_model <- train(type ~ npreg + glu + bmi + ped, data=comb_data, method="glm", family="binomial", trControl=control)

lda_model <- train(type ~ npreg + glu + bmi + ped, data=comb_data, method="lda", trControl=control)

qda_model <- train(type ~ npreg + glu + bmi + ped, data=comb_data, method="qda", trControl=control)

# plot confusion matrix

confusionMatrix(logistic_model$pred$pred, logistic_model$pred$obs)
confusionMatrix(lda_model$pred$pred, lda_model$pred$obs)
confusionMatrix(qda_model$pred$pred, qda_model$pred$obs)

###Training error###
# For Logistic Regression:
logistic_preds <- predict(logistic_model, newdata = comb_data)
logistic_error <- mean(logistic_preds != comb_data$type)

# For LDA:
lda_preds <- predict(lda_model, newdata = comb_data)
lda_error <- mean(lda_preds != comb_data$type)
```

```
# For LDA:
lda_preds <- predict(lda_model, newdata = comb_data)
lda_error <- mean(lda_preds != comb_data$type)

# For QDA:
qda_preds <- predict(qda_model, newdata = comb_data)
qda_error <- mean(qda_preds != comb_data$type)

# Print the training errors:
cat("Logistic Regression Training Error:", logistic_error, "\n")
cat("LDA Training Error:", lda_error, "\n")
cat("QDA Training Error:", qda_error, "\n")
```

Results –

Best subset Analysis AIC and BIC –

```
> bic = results$bic
> which.min(bic)
[1] 4
> aic = results$cp
> which.min(aic)
[1] 5
>
> coef(subsets, which.min(bic))
 (Intercept)         npreg           glu           bmi           ped
-0.018360784  0.028246117  0.006275351  0.012108102  0.186859358
>
> coef(subsets, which.min(aic))
 (Intercept)         npreg           glu           bmi           ped
-0.079262888  0.020206720  0.006007237  0.012009726  0.182745138
         age
 0.004014933
```

Best subset k- fold cross validation output

```
> cat("Optimal number of predictors:", optimal.number, "\n")
Optimal number of predictors: 1
> best.subset <- regsubsets(type ~ ., data=comb_data, nvmax=ncol(comb_d
ata)-1)
> coef(best.subset, optimal.number)
  (Intercept)           glu
-0.594630955  0.007662044
```

LDA and QDA models -

```
> print(lda.model$means)
     npreg      glu      bmi       ped
0 2.926761 110.0169 31.42958 0.4463155
1 4.700565 143.1186 35.81977 0.6165876
```

```
> print(qda.model$means)
     npreg      glu      bmi       ped
0 2.926761 110.0169 31.42958 0.4463155
1 4.700565 143.1186 35.81977 0.6165876
```

Confusion matrix for all models -

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 321  73
         1  34 104

               Accuracy : 0.7989
                 95% CI : (0.7623, 0.8321)
    No Information Rate : 0.6673
    P-Value [Acc > NIR] : 1.273e-11

                  Kappa : 0.5206
```

```
> confusionMatrix(qda_model$pred$pred, qda_model$pred$obs)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 307  71
         1  48 106

               Accuracy : 0.7763
                 95% CI : (0.7385, 0.811)
    No Information Rate : 0.6673
    P-Value [Acc > NIR] : 2.283e-08
```

```
> confusionMatrix(qda_model$pred$pred, qda_model$pred$obs)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 307  71
         1  48 106

               Accuracy : 0.7763
                 95% CI : (0.7385, 0.811)
    No Information Rate : 0.6673
    P-Value [Acc > NIR] : 2.283e-08

                  Kappa : 0.4793
```

Training errors -

```
> # Print the training errors:
> cat("Logistic Regression Training Error:", logistic_error, "\n")
Logistic Regression Training Error: 0.2011278
> cat("LDA Training Error:", lda_error, "\n")
LDA Training Error: 0.2105263
> cat("QDA Training Error:", qda_error, "\n")
QDA Training Error: 0.2218045
```