*#importing dataset*

data <- read.csv("heart.csv")

*#installing packages*

install.packages("GGally")

install.packages("reshape2")

install.packages("plotly")

install.packages("ggcorrplot")

#importing library

library(reshape2)

library(caret)

library(e1071)

library(randomForest)

library(naivebayes)

library(rpart)

library(class)

library(xgboost)

library(lda)

library(MASS)

library(ggplot2)

library(plotly)

library(ggcorrplot)

library(GGally)

library(dplyr)

library(tidyr)

*#data preparation*

dim(data)

str(data)

summary(data)

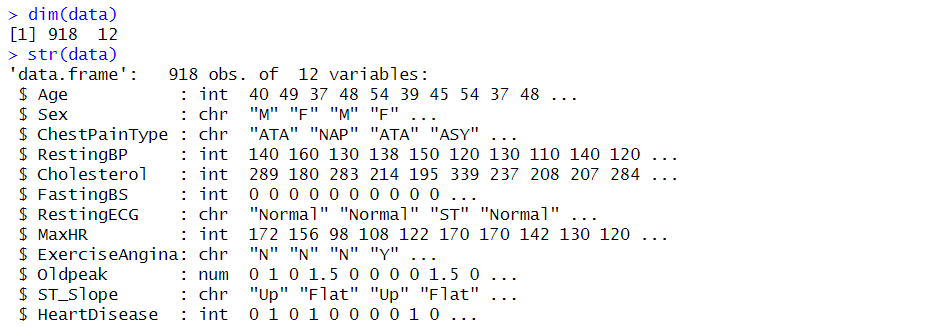
is.na(data)

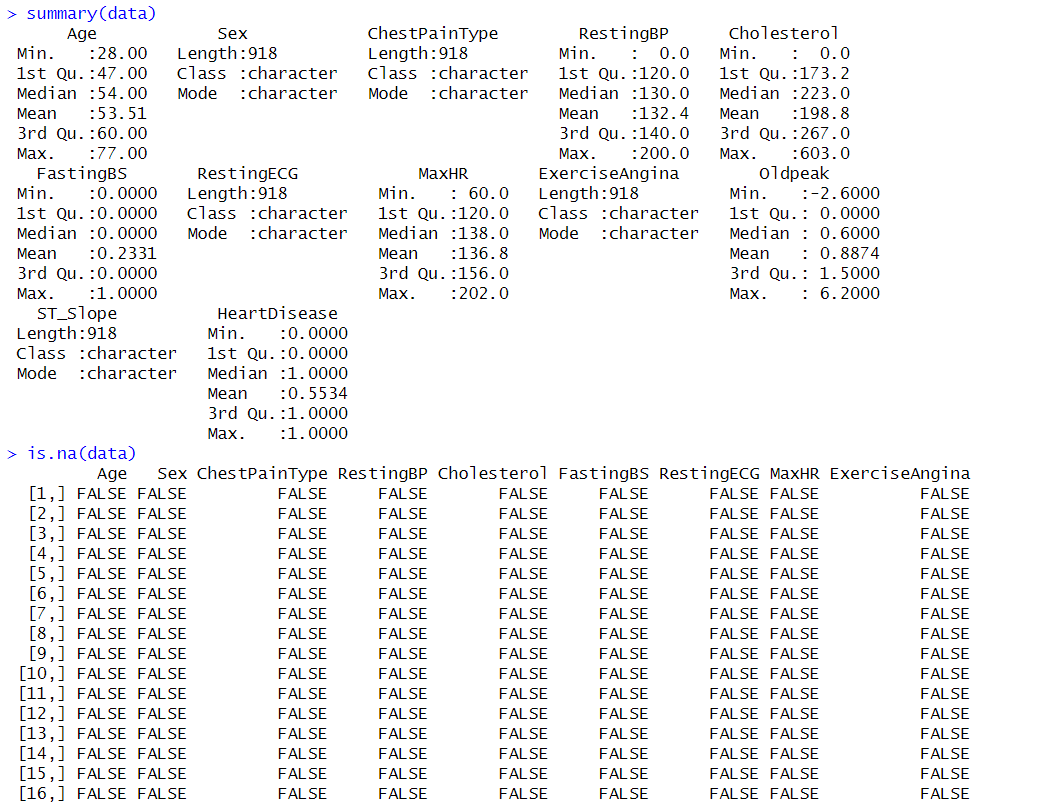
data %>% distinct()

infinite\_columns <- sapply(data, function(col) any(is.infinite(col)))

clean\_data <- data[!apply(data[infinite\_columns], 1, anyNA), ]

data <- clean\_data





A screenshot of a computer

Description automatically generated

#Check for null values in each column

null\_columns <- colSums(is.na(data))

print(null\_columns)

A close-up of a white background

Description automatically generated

*#data cleaning and transformation*

data <- na.omit(data)

data$Sex <- c(F=1 , M=0)[data$Sex]

data$ChestPainType <- c(ATA=0, NAP=1, ASY= 2)[data$ChestPainType]

data$RestingECG <- c(Normal=0, ST=1)[data$RestingECG]

data$ExerciseAngina <- c(N=0, Y=1)[data$ExerciseAngina]

data$ST\_Slope <- c(Up=0, Flat=1, Down=2)[data$ST\_Slope]

data$ChestPainType[is.na(data$ChestPainType)] <- median(data$ChestPainType, na.rm = TRUE)

data$Cholesterol[is.na(data$Cholesterol)] <- median(data$Cholesterol, na.rm = TRUE)

data$RestingECG[is.na(data$RestingECG)] <- median(data$RestingECG, na.rm = TRUE)

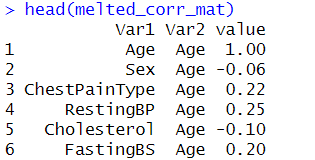
*#finding correlation*

corr\_mat <- round(cor(data),2)

# reduce the size of correlation matrix

melted\_corr\_mat <- melt(corr\_mat)

head(melted\_corr\_mat)



*# plotting the correlation heatmap*

ggplot(data = melted\_corr\_mat, aes(x=Var1, y=Var2,fill=value)) + geom\_tile()

A screenshot of a computer screen

Description automatically generated

*# Select the six features from the dataset*

selected\_features <- data %>%

select("Age", "Sex", "ChestPainType", "RestingBP", "Cholesterol", "MaxHR")

*# Calculate the correlation matrix for the selected features*

correlation\_matrix <- cor(selected\_features)

*# Create a ggplot heatmap*

heatmap\_plot <- ggplot(data = as.data.frame(as.table(correlation\_matrix)), aes(Var1, Var2)) +

geom\_tile(aes(fill = Freq), color = "white") +

scale\_fill\_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0, limits = c(-1, 1)) +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +

labs(title = "Correlation Heatmap of Selected Features",

x = "Features",

y = "Features")

# Print the heatmap plot

print(heatmap\_plot)

A screenshot of a diagram

Description automatically generated

*# Create a pair plot using GGally*

ggpairs(selected\_features , title = "Pair Plot of Heart Data")

A screenshot of a graph

Description automatically generated

*#Count plot*

ggplot(data, aes(x = HeartDisease)) + geom\_bar()

A graph of a graph showing a number of squares

Description automatically generated with medium confidence

*#the no of healthy and sick pepple in both gender*

result <- data %>%

group\_by(Sex,HeartDisease) %>%

summarize(Count = n())

result

A screenshot of a computer code

Description automatically generated

plot\_ly(data = data, x = ~Sex, color = ~HeartDisease, type = "histogram", barmode = "group", colors = c('#71AEC2', '#D58989')) %>%

layout(title = "Distribution")

A graph with a bar and a number of numbers

Description automatically generated with medium confidence

*# Pivot the data using dcast*

pivot\_data <- dcast(data, Sex ~ RestingECG, value.var = "HeartDisease", fun.aggregate = mean)

*# Melt the pivoted data*

melted\_data <- melt(pivot\_data, id.vars = "Sex", variable.name = "RestingECG")

*# Create a pie chart for each category*

pie\_chart\_list <- lapply(unique(melted\_data$RestingECG), function(category) {

subset\_data <- melted\_data[melted\_data$RestingECG == category, ]

pie\_chart <- ggplot(subset\_data, aes(x = "", y = value, fill = Sex)) +

geom\_bar(stat = "identity", width = 1) +

coord\_polar("y", start = 0) +

labs(title = paste("The effect of RestingECG of both sex relations to the disease"),

x = NULL, y = NULL) +

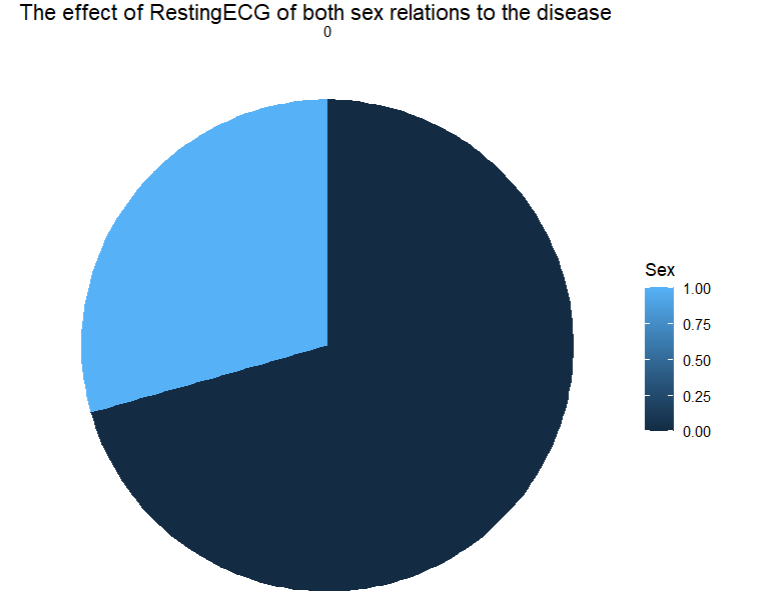
theme\_void() +

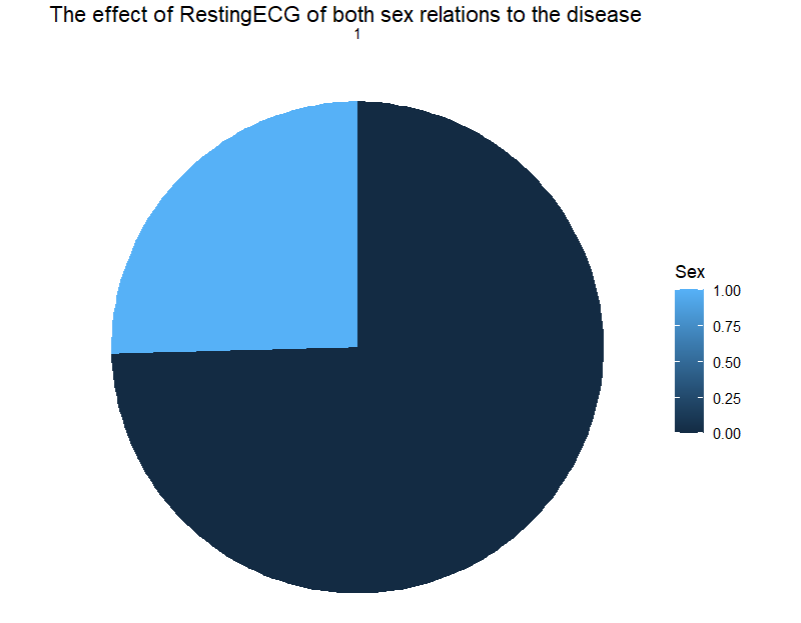
facet\_wrap(~ RestingECG)

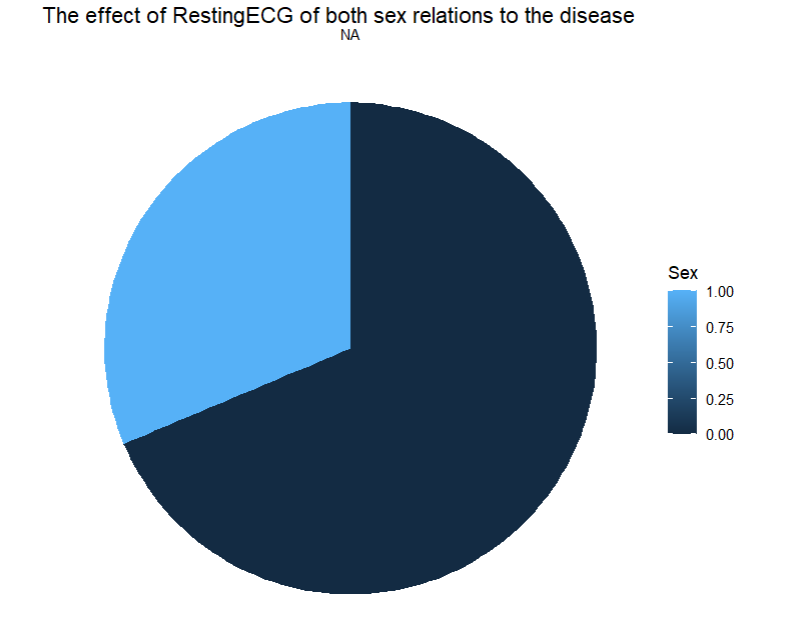
return(pie\_chart)

})

print(pie\_chart\_list)







# Reshape the data

reshaped\_data <- as.data.frame(pivot\_data)

reshaped\_data$Sex <- rownames(reshaped\_data)

# Convert the data to long format

library(reshape2)

long\_data <- melt(reshaped\_data, id.vars = "Sex")

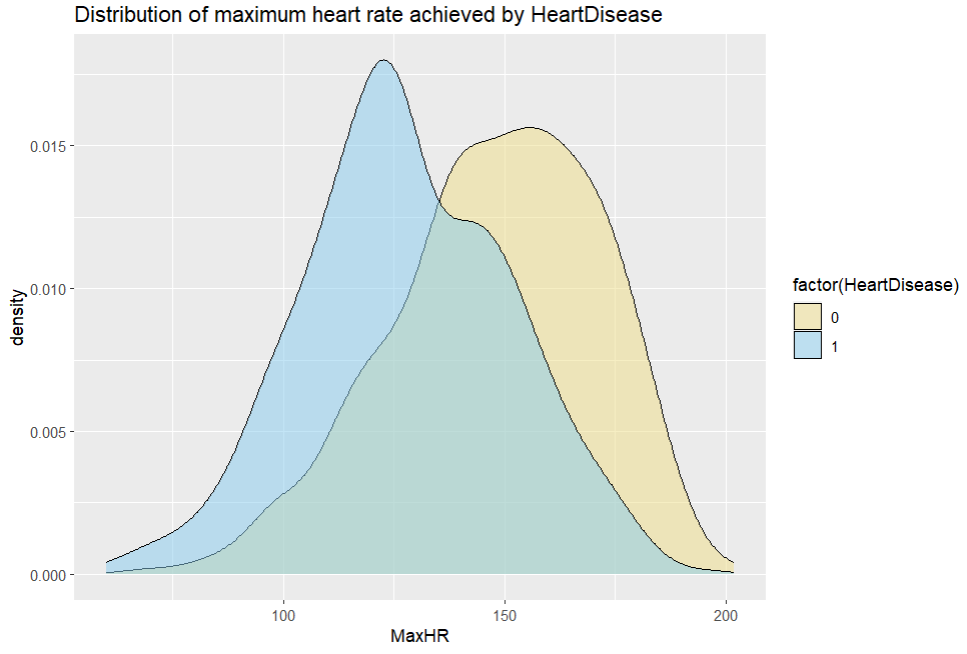
#checking the Distribution of maximum heart rate achieved by HeartDisease

ggplot(data, aes(x = MaxHR, fill = factor(HeartDisease))) +

geom\_density(alpha = 0.5) +

scale\_fill\_manual(values = c("#EEDD88", "#88CCEE")) + # Set custom colors

labs(title = "Distribution of maximum heart rate achieved by HeartDisease")



#checking Distribution of resting blood pressure by disease

ggplot(data, aes(x = as.factor(HeartDisease), y = RestingBP, color = as.factor(HeartDisease))) +

geom\_jitter() +

labs(title = "Distribution of resting blood pressure by disease") +

scale\_color\_manual(values = c("#E41A1C", "#377EB8"))

A graph showing different types of blood pressure

Description automatically generated

#effect of fasting blood sugar on disease

res <- data %>%

group\_by(FastingBS,HeartDisease) %>%

summarize(Count = n())

res

A screenshot of a computer code

Description automatically generated

plot\_ly(data = data, x = ~FastingBS, color = ~HeartDisease, type = "histogram", barmode = "group", colors = c('#71AEC2', '#D58989')) %>%

layout(title = "Fasting blood sugar on disease")

A graph of fasting blood sugar on disease

Description automatically generated

#the Average cholesterol for both sexes in relation to the disease

re <- aggregate(Cholesterol ~ HeartDisease + Sex, data = data, FUN = mean)

pivot\_result <- reshape(re, idvar = "HeartDisease", timevar = "Sex", direction = "wide")

re

A number of numbers in a row

Description automatically generated with medium confidence

#checking the average disease for each age group

data$age\_cat <- cut(data$Age, breaks = 4)

res1 <- data %>%

group\_by(age\_cat) %>%

summarise(mean\_HeartDisease = mean(HeartDisease)) %>%

arrange(desc(mean\_HeartDisease))

res1

A screenshot of a computer

Description automatically generated

data$HeartDisease <- factor(data$HeartDisease)

ggplot(data, aes(x = HeartDisease, fill = HeartDisease)) +

geom\_bar(stat = "count") +

facet\_wrap(~ age\_cat, ncol = 1) +

labs(title = "Distribution of age category by disease")

A screenshot of a graph

Description automatically generated

#checking the average disease for each Cholesterol group

data$Chol\_cat=cut(data$Cholesterol,breaks=3)

data$HeartDisease <- as.numeric(as.character(data$HeartDisease))

res2 <- data %>%

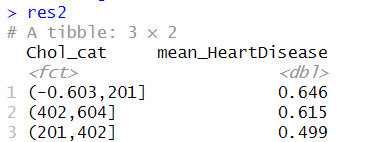
filter(is.numeric(HeartDisease)) %>%

group\_by(Chol\_cat) %>%

summarise(mean\_HeartDisease = mean(HeartDisease, na.rm = TRUE)) %>%

arrange(desc(mean\_HeartDisease))

res2



plot\_ly(data, x = ~HeartDisease, color = ~HeartDisease) %>%

add\_histogram(

facet\_col = ~Chol\_cat,

barmode = 'group',

marginal = 'rug'

) %>%

layout(

title = 'Distribution of Cholesterol categories by disease'

)

A yellow and purple squares

Description automatically generated

#checking the average disease for each maximum heart rate achieved group

data$MaxHR\_cat=cut(data$MaxHR,breaks = 3)

res3 <- data %>%

group\_by(MaxHR\_cat) %>%

summarise(mean\_HeartDisease = mean(HeartDisease)) %>%

arrange(desc(mean\_HeartDisease))

res3

A screenshot of a computer code

Description automatically generated

data$HeartDisease <- factor(data$HeartDisease)

ggplot(data, aes(x = HeartDisease, fill = HeartDisease)) +

geom\_bar(stat = "count") +

facet\_grid(. ~ MaxHR\_cat) +

labs(title = "Distribution of maximum heart rate achieved by disease")

A graph of different colored squares

Description automatically generated

#checking What is the average disease for each Oldpeak group

data$Oldpeak\_cat=cut(data$Oldpeak,breaks = 2)

data$HeartDisease <- as.numeric(as.character(data$HeartDisease))

res4 <- data %>%

group\_by(Oldpeak\_cat) %>%

summarise(mean\_HeartDisease = mean(HeartDisease)) %>%

arrange(desc(mean\_HeartDisease))

res4

A black text on a white background

Description automatically generated

plot\_ly(data, x = ~HeartDisease, color = ~HeartDisease, type = "histogram") %>%

layout(barmode = "group", title = "Distribution of Heart Disease")

data <- subset(data, select = -c(age\_cat, Chol\_cat, MaxHR\_cat, Oldpeak\_cat))

A yellow and purple squares

Description automatically generated

#feature selection

predictor <- data[,-c(12)]

correlation <- abs(cor(predictor,data$HeartDisease,use = "everything",

method = c("pearson", "kendall", "spearman")))

correlation <- na.omit(correlation)

correlation\_filter <- correlation[, 1] > 0.4

features <- names(predictor)[correlation\_filter]

features



#splitting training and testing data

set.seed(123)

trainIndex <- createDataPartition(data$HeartDisease, p = 0.7, list = FALSE)

train\_data <- data[trainIndex,]

test\_data <- data[-trainIndex,]

log\_model <- glm(HeartDisease ~ ., data=train\_data,family = binomial ,model = TRUE, method = "glm.fit")

log\_predictions <- predict(log\_model,test\_data,ype = "response")

log\_predictions <- ifelse(is.na(log\_predictions), 0, log\_predictions)

log\_predictions <- ifelse(log\_predictions > 0.5, 1, 0)

plot(test\_data$HeartDisease,type="l",col="red")

lines(log\_predictions)

log\_conf\_matrix <- table(test\_data$HeartDisease,log\_predictions)

log\_conf\_matrix

A bar code with numbers and a number of numbers

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

#evaluating logistic regression model

log\_accuracy <- sum(diag(log\_conf\_matrix))/sum(log\_conf\_matrix)

log\_accuracy

log\_precision <- log\_conf\_matrix[2,2]/(log\_conf\_matrix[2,1]+log\_conf\_matrix[2,2])

log\_precision

log\_recall <- log\_conf\_matrix[2,2]/(log\_conf\_matrix[2,2]+log\_conf\_matrix[1,2])

log\_recall

log\_specifi <- log\_conf\_matrix[1,1]/(log\_conf\_matrix[1,1]+log\_conf\_matrix[2,1])

log\_specifi

log\_f1\_score <- 2 \* ((log\_precision \* log\_recall) / (log\_precision + log\_recall))

log\_f1\_score

A computer screen shot of a code

Description automatically generated

#naive bayes model

nb\_model <- naiveBayes(HeartDisease ~ ., data = train\_data)

nb\_predictions <- predict(nb\_model, newdata = test\_data)

plot(test\_data$HeartDisease,type="l",col="red")

lines(nb\_predictions)

nb\_conf\_matrix <- table(test\_data$HeartDisease,nb\_predictions)

nb\_conf\_matrix

A bar code with numbers and a bar code

Description automatically generated with medium confidence

A white background with black text

Description automatically generated

#evaluating naive bayes model

nb\_accuracy <- sum(diag(nb\_conf\_matrix))/sum(nb\_conf\_matrix)

nb\_accuracy

nb\_precision <- nb\_conf\_matrix[2,2]/(nb\_conf\_matrix[2,1]+nb\_conf\_matrix[2,2])

nb\_precision

nb\_recall <- nb\_conf\_matrix[2,2]/(nb\_conf\_matrix[2,2]+nb\_conf\_matrix[1,2])

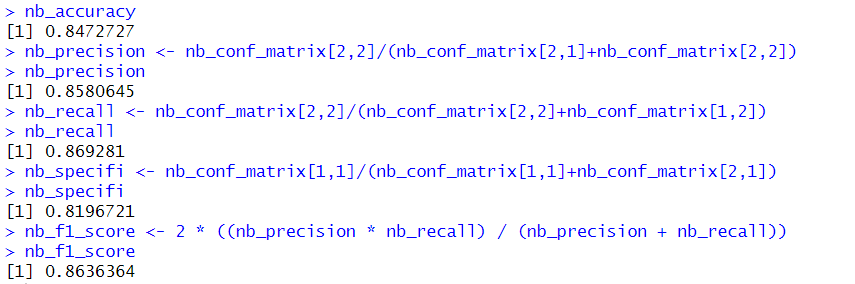
nb\_recall

nb\_specifi <- nb\_conf\_matrix[1,1]/(nb\_conf\_matrix[1,1]+nb\_conf\_matrix[2,1])

nb\_specifi

nb\_f1\_score <- 2 \* ((nb\_precision \* nb\_recall) / (nb\_precision + nb\_recall))

nb\_f1\_score



#support vector machine

svm\_models <- svm(HeartDisease ~ ., data = train\_data)

svm\_predictions <- predict(svm\_models, newdata = test\_data)

svm\_predictions <- ifelse(svm\_predictions>0.5,1,0)

plot(test\_data$HeartDisease,type="l",col="red")

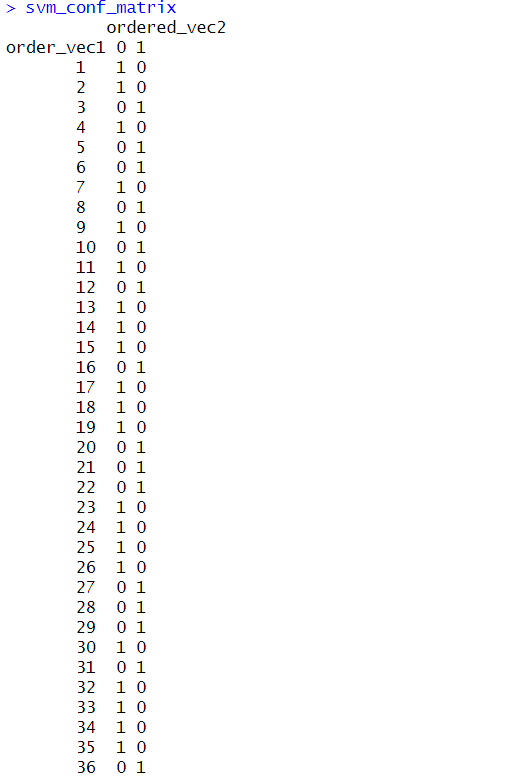
lines(svm\_predictions)

order\_vec1 <- order(test\_data$HeartDisease)

ordered\_vec2 <- svm\_predictions[order\_vec1]

svm\_conf\_matrix <- table(order\_vec1,ordered\_vec2)

svm\_conf\_matrix



A bar code with numbers

Description automatically generated

#evaluating svm model

svm\_accuracy <- sum(diag(svm\_conf\_matrix))/sum(svm\_conf\_matrix)

svm\_accuracy

svm\_precision <- svm\_conf\_matrix[2,2]/(svm\_conf\_matrix[2,1]+svm\_conf\_matrix[2,2])

svm\_precision

svm\_recall <- svm\_conf\_matrix[2,2]/(svm\_conf\_matrix[2,2]+svm\_conf\_matrix[1,2])

svm\_recall

svm\_specifi <- svm\_conf\_matrix[1,1]/(svm\_conf\_matrix[1,1]+svm\_conf\_matrix[2,1])

svm\_specifi

svm\_f1\_score <- 2 \* ((svm\_precision \* svm\_recall) / (svm\_precision + svm\_recall))

svm\_f1\_score

A white background with blue text

Description automatically generated

#decision tree

dt\_model <- rpart(HeartDisease ~ ., data = train\_data, method = "class")

dt\_predictions <- predict(dt\_model,newdata = test\_data, type="class")

plot(test\_data$HeartDisease,type="l",col="red")

lines(dt\_predictions)

dt\_conf\_matrix <- table(test\_data$HeartDisease,dt\_predictions)

dt\_conf\_matrix

A white background with black text

Description automatically generated

A bar code with red lines

Description automatically generated

#evaluating decision tree model

dt\_accuracy <- sum(diag(dt\_conf\_matrix))/sum(dt\_conf\_matrix)

dt\_accuracy

dt\_precision <- dt\_conf\_matrix[2,2]/(dt\_conf\_matrix[2,1]+dt\_conf\_matrix[2,2])

dt\_precision

dt\_recall <- dt\_conf\_matrix[2,2]/(dt\_conf\_matrix[2,2]+dt\_conf\_matrix[1,2])

dt\_recall

dt\_specifi <- dt\_conf\_matrix[1,1]/(dt\_conf\_matrix[1,1]+dt\_conf\_matrix[2,1])

dt\_specifi

dt\_f1\_score <- 2 \* ((dt\_precision \* dt\_recall) / (dt\_precision + dt\_recall))

dt\_f1\_score

A white background with blue text

Description automatically generated

#random forest

rf\_model <- randomForest(HeartDisease ~ .,data=train\_data, na.action = na.roughfix)

rf\_predictions <- predict(rf\_model, newdata = test\_data)

rf\_predictions <- ifelse(rf\_predictions>0.5,1,0)

rf\_predictions

plot(test\_data$HeartDisease,type="l",col="red")

lines(rf\_predictions)

rf\_conf\_matrix <- table(test\_data$HeartDisease,rf\_predictions)

rf\_conf\_matrixdata$ChestPainType[is.na(data$ChestPainType)] <- median(data$ChestPainType, na.rm = TRUE)

data$Cholesterol[is.na(data$Cholesterol)] <- median(data$Cholesterol, na.rm = TRUE)

data$RestingECG[is.na(data$RestingECG)] <- median(data$RestingECG, na.rm = TRUE)

rf\_model <- randomForest(HeartDisease ~ .,data=train\_data, na.action = na.roughfix)

rf\_predictions <- predict(rf\_model, newdata = test\_data)

rf\_predictions <- ifelse(rf\_predictions>0.5,1,0)

rf\_predictions

plot(test\_data$HeartDisease,type="l",col="red")

lines(rf\_predictions)

rf\_conf\_matrix <- table(test\_data$HeartDisease,rf\_predictions)

rf\_conf\_matrix

A bar code with red and black lines

Description automatically generated

A white background with black text

Description automatically generated

#evaluating random forest

rf\_accuracy <- sum(diag(rf\_conf\_matrix))/sum(rf\_conf\_matrix)

rf\_accuracy

rf\_precision <- rf\_conf\_matrix[2,2]/(rf\_conf\_matrix[2,1]+rf\_conf\_matrix[2,2])

rf\_precision

rf\_recall <- rf\_conf\_matrix[2,2]/(rf\_conf\_matrix[2,2]+rf\_conf\_matrix[1,2])

rf\_recall

rf\_specifi <- rf\_conf\_matrix[1,1]/(rf\_conf\_matrix[1,1]+rf\_conf\_matrix[2,1])

rf\_specifi

rf\_f1\_score <- 2 \* ((rf\_precision \* rf\_recall) / (rf\_precision + rf\_recall))

rf\_f1\_score

A white background with blue text

Description automatically generated

# Linear Discriminant Analysis (LDA)

lda\_model <- lda(HeartDisease ~ ., data = train\_data)

lda\_predictions <- predict(lda\_model, newdata = test\_data)$class

plot(test\_data$HeartDisease,type="l",col="red")

lines(lda\_predictions)

lda\_conf\_matrix <- table(test\_data$HeartDisease,lda\_predictions)

lda\_conf\_matrix

A bar code with red lines

Description automatically generated

A close-up of numbers

Description automatically generated

#evaluating lda model

lda\_accuracy <- sum(diag(lda\_conf\_matrix))/sum(lda\_conf\_matrix)

lda\_accuracy

lda\_precision <- lda\_conf\_matrix[2,2]/(lda\_conf\_matrix[2,1]+lda\_conf\_matrix[2,2])

lda\_precision

lda\_recall <- lda\_conf\_matrix[2,2]/(lda\_conf\_matrix[2,2]+lda\_conf\_matrix[1,2])

lda\_recall

lda\_specifi <- lda\_conf\_matrix[1,1]/(lda\_conf\_matrix[1,1]+lda\_conf\_matrix[2,1])

lda\_specifi

lda\_f1\_score <- 2 \* ((lda\_precision \* lda\_recall) / (lda\_precision + lda\_recall))

lda\_f1\_score

A computer code with blue text

Description automatically generated with medium confidence

# Quadratic Discriminant Analysis (QDA)

qda\_model <- qda(HeartDisease ~ ., data = train\_data)

qda\_predictions <- predict(qda\_model, newdata = test\_data)$class

plot(test\_data$HeartDisease,type="l",col="red")

lines(qda\_predictions)

qda\_conf\_matrix <- table(test\_data$HeartDisease,qda\_predictions)

qda\_conf\_matrix

A bar code with numbers and a red line

Description automatically generated

A white background with black text

Description automatically generated

#evaluating qda matrix

qda\_accuracy <- sum(diag(qda\_conf\_matrix))/sum(qda\_conf\_matrix)

qda\_accuracy

qda\_precision <- qda\_conf\_matrix[2,2]/(qda\_conf\_matrix[2,1]+qda\_conf\_matrix[2,2])

qda\_precision

qda\_recall <- qda\_conf\_matrix[2,2]/(qda\_conf\_matrix[2,2]+qda\_conf\_matrix[1,2])

qda\_recall

qda\_specifi <- qda\_conf\_matrix[1,1]/(qda\_conf\_matrix[1,1]+qda\_conf\_matrix[2,1])

qda\_specifi

qda\_f1\_score <- 2 \* ((qda\_precision \* qda\_recall) / (qda\_precision + qda\_recall))

qda\_f1\_score

A white background with blue text

Description automatically generated

#data visualization

#Accuracy

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

accuracies <- c(log\_accuracy,nb\_accuracy, svm\_accuracy, dt\_accuracy, rf\_accuracy, knn\_accuracy, lda\_accuracy, qda\_accuracy)

accuracy\_data <- data.frame(Model = model\_names, Accuracy = accuracies)

ggplot(accuracy\_data, aes(x = Model, y = Accuracy)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model Accuracies", x = "Model", y = "Accuracy")

A graph of blue rectangular bars

Description automatically generated

#Precision

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

precision <- c(log\_precision,nb\_precision, svm\_precision, dt\_precision, rf\_precision, knn\_precision, lda\_precision, qda\_precision)

precision\_data <- data.frame(Model = model\_names, Precision = precision)

ggplot(precision\_data, aes(x = Model, y = Precision)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model precision", x = "Model", y = "precision")

A graph of blue bars

Description automatically generated

#recall

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

recall <- c(log\_recall,nb\_recall, svm\_recall, dt\_recall, rf\_recall, knn\_recall, lda\_recall, qda\_recall)

recall\_data <- data.frame(Model = model\_names, Recall = recall)

ggplot(recall\_data, aes(x = Model, y = Recall)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model recall", x = "Model", y = " Recall")

A graph of blue rectangular bars

Description automatically generated

#specificity

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

specificity <- c(log\_specifi,nb\_specifi, svm\_specifi, dt\_specifi, rf\_specifi, knn\_specifi, lda\_specifi, qda\_specifi)

specificity\_data <- data.frame(Model = model\_names, Specificity = specificity )

ggplot(specificity\_data, aes(x = Model, y = Specificity )) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model specificity", x = "Model", y = "specificity")

A graph of blue bars

Description automatically generated

#f1 score

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

f1\_score <- c(log\_f1\_score,nb\_f1\_score, svm\_f1\_score, dt\_f1\_score, rf\_f1\_score, knn\_f1\_score, lda\_f1\_score, qda\_f1\_score)

f1\_score\_data <- data.frame(Model = model\_names, F1\_score = f1\_score )

ggplot(f1\_score\_data, aes(x = Model, y = f1\_score )) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model f1\_score", x = "Model", y = "f1\_score")

A graph of blue bars

Description automatically generated

#features

train\_data\_features <- subset(train\_data, select = c( "ChestPainType", "MaxHR", "ExerciseAngina", "Oldpeak", "ST\_Slope","HeartDisease"))

test\_data\_features <- subset(test\_data,select = c("ChestPainType", "MaxHR", "ExerciseAngina", "Oldpeak", "ST\_Slope","HeartDisease"))

#logistic regression model with selected features

log\_model\_f <- glm(HeartDisease ~ ., data = train\_data\_features,family = binomial ,model = TRUE, method = "glm.fit")

log\_predictions\_f <- predict(log\_model\_f,test\_data\_features,type = "response")

log\_predictions\_f <- ifelse(is.na(log\_predictions\_f), 0, log\_predictions\_f)

log\_predictions\_f <- ifelse(log\_predictions\_f > 0.5, 1, 0)

log\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,log\_predictions\_f)

log\_conf\_matrix\_f

A black text on a white background

Description automatically generated

#evaluating logistic regression model

log\_accuracy\_f <- sum(diag(log\_conf\_matrix\_f))/sum(log\_conf\_matrix\_f)

log\_accuracy\_f

log\_precision\_f <- log\_conf\_matrix\_f[2,2]/(log\_conf\_matrix\_f[2,1]+log\_conf\_matrix\_f[2,2])

log\_precision\_f

log\_recall\_f <- log\_conf\_matrix\_f[2,2]/(log\_conf\_matrix\_f[2,2]+log\_conf\_matrix\_f[1,2])

log\_recall\_f

log\_specifi\_f <- log\_conf\_matrix\_f[1,1]/(log\_conf\_matrix\_f[1,1]+log\_conf\_matrix\_f[2,1])

log\_specifi\_f

log\_f1\_score\_f <- 2 \* ((log\_precision\_f \* log\_recall\_f) / (log\_precision\_f + log\_recall\_f))

log\_f1\_score\_f

A computer code with blue text

Description automatically generated

#naive bayes model features

nb\_model\_f <- naiveBayes(HeartDisease ~ ., data = train\_data\_features)

nb\_predictions\_f <- predict(nb\_model\_f, newdata = test\_data\_features)

nb\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,nb\_predictions\_f)

nb\_conf\_matrix\_f

A black text on a white background

Description automatically generated

#evaluating naive bayes model

nb\_accuracy\_f <- sum(diag(nb\_conf\_matrix\_f))/sum(nb\_conf\_matrix\_f)

nb\_accuracy\_f

nb\_precision\_f <- nb\_conf\_matrix\_f[2,2]/(nb\_conf\_matrix\_f[2,1]+nb\_conf\_matrix\_f[2,2])

nb\_precision\_f

nb\_recall\_f <- nb\_conf\_matrix\_f[2,2]/(nb\_conf\_matrix\_f[2,2]+nb\_conf\_matrix\_f[1,2])

nb\_recall\_f

nb\_specifi\_f <- nb\_conf\_matrix\_f[1,1]/(nb\_conf\_matrix\_f[1,1]+nb\_conf\_matrix\_f[2,1])

nb\_specifi\_f

nb\_f1\_score\_f <- 2 \* ((nb\_precision\_f \* nb\_recall\_f) / (nb\_precision\_f + nb\_recall\_f))

nb\_f1\_score\_f

A close-up of a computer code

Description automatically generated

#support vector machine with features

svm\_models\_f <- svm(HeartDisease ~ ., data = train\_data\_features)

svm\_predictions\_f <- predict(svm\_models\_f, newdata = test\_data\_features)

svm\_predictions\_f <- ifelse(svm\_predictions\_f>0.5,1,0)

svm\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,svm\_predictions\_f)

svm\_conf\_matrix\_f

A black text on a white background

Description automatically generated

#evaluating svm model

svm\_accuracy\_f <- sum(diag(svm\_conf\_matrix\_f))/sum(svm\_conf\_matrix\_f)

svm\_accuracy\_f

svm\_precision\_f <- svm\_conf\_matrix\_f[2,2]/(svm\_conf\_matrix\_f[2,1]+svm\_conf\_matrix\_f[2,2])

svm\_precision\_f

svm\_recall\_f <- svm\_conf\_matrix\_f[2,2]/(svm\_conf\_matrix\_f[2,2]+svm\_conf\_matrix\_f[1,2])

svm\_recall\_f

svm\_specifi\_f <- svm\_conf\_matrix\_f[1,1]/(svm\_conf\_matrix\_f[1,1]+svm\_conf\_matrix\_f[2,1])

svm\_specifi\_f

svm\_f1\_score\_f <- 2 \* ((svm\_precision\_f \* svm\_recall\_f) / (svm\_precision\_f + svm\_recall\_f))

svm\_f1\_score\_f

A computer code with blue text

Description automatically generated

#decision tree

dt\_model\_f <- rpart(HeartDisease ~ ., data = train\_data\_features, method = "class")

dt\_predictions\_f <- predict(dt\_model\_f,newdata = test\_data\_features, type="class")

dt\_predictions\_f

dt\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,dt\_predictions\_f)

dt\_conf\_matrix\_f

A close-up of a number

Description automatically generated

#evaluating decision tree model

dt\_accuracy\_f <- sum(diag(dt\_conf\_matrix\_f))/sum(dt\_conf\_matrix\_f)

dt\_accuracy\_f

dt\_precision\_f <- dt\_conf\_matrix\_f[2,2]/(dt\_conf\_matrix\_f[2,1]+dt\_conf\_matrix\_f[2,2])

dt\_precision\_f

dt\_recall\_f <- dt\_conf\_matrix\_f[2,2]/(dt\_conf\_matrix\_f[2,2]+dt\_conf\_matrix\_f[1,2])

dt\_recall\_f

dt\_specifi\_f <- dt\_conf\_matrix\_f[1,1]/(dt\_conf\_matrix\_f[1,1]+dt\_conf\_matrix\_f[2,1])

dt\_specifi\_f

dt\_f1\_score\_f <- 2 \* ((dt\_precision\_f \* dt\_recall\_f) / (dt\_precision\_f + dt\_recall\_f))

dt\_f1\_score\_f

A white background with blue text

Description automatically generated

#random forest with features

rf\_model\_f <- randomForest(HeartDisease ~ .,data=train\_data\_features)

rf\_predictions\_f <- predict(rf\_model\_f, newdata = test\_data\_features)

rf\_predictions\_f <- ifelse(rf\_predictions\_f>0.5,1,0)

rf\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,rf\_predictions\_f)

rf\_conf\_matrix\_f

A close-up of a number

Description automatically generated

#evaluating random forest

rf\_accuracy\_f <- sum(diag(rf\_conf\_matrix\_f))/sum(rf\_conf\_matrix\_f)

rf\_accuracy\_f

rf\_precision\_f <- rf\_conf\_matrix\_f[2,2]/(rf\_conf\_matrix\_f[2,1]+rf\_conf\_matrix\_f[2,2])

rf\_precision\_f

rf\_recall\_f <- rf\_conf\_matrix\_f[2,2]/(rf\_conf\_matrix\_f[2,2]+rf\_conf\_matrix\_f[1,2])

rf\_recall\_f

rf\_specifi\_f <- rf\_conf\_matrix\_f[1,1]/(rf\_conf\_matrix\_f[1,1]+rf\_conf\_matrix\_f[2,1])

rf\_specifi\_f

rf\_f1\_score\_f <- 2 \* ((rf\_precision\_f \* rf\_recall\_f) / (rf\_precision\_f + rf\_recall\_f))

rf\_f1\_score\_f

A computer code with blue text

Description automatically generated with medium confidence

#knn

knn\_model\_f <- knn(train\_data\_features[, -1], test\_data\_features[, -1], train\_data\_features$HeartDisease, k = 2)

knn\_predictions\_f <- as.numeric(knn\_model\_f)

knn\_predictions\_f <- ifelse(knn\_predictions\_f==2,1,0)

knn\_predictions\_f

knn\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,knn\_predictions\_f)

knn\_conf\_matrix\_f

A black text on a white background

Description automatically generated

#evaluating knn model

knn\_accuracy\_f <- sum(diag(knn\_conf\_matrix\_f))/sum(knn\_conf\_matrix\_f)

knn\_accuracy\_f

knn\_precision\_f <- knn\_conf\_matrix\_f[2,2]/(knn\_conf\_matrix\_f[2,1]+knn\_conf\_matrix\_f[2,2])

knn\_precision\_f

knn\_recall\_f <- knn\_conf\_matrix\_f[2,2]/(knn\_conf\_matrix\_f[2,2]+knn\_conf\_matrix\_f[1,2])

knn\_recall\_f

knn\_specifi\_f <- knn\_conf\_matrix\_f[1,1]/(knn\_conf\_matrix\_f[1,1]+knn\_conf\_matrix\_f[2,1])

knn\_specifi\_f

knn\_f1\_score\_f <- 2 \* ((knn\_precision\_f \* knn\_recall\_f) / (knn\_precision\_f + knn\_recall\_f))

knn\_f1\_score\_f

A white background with blue text

Description automatically generated

# Linear Discriminant Analysis (LDA) with selected features

lda\_model\_f <- lda(HeartDisease ~ ., data = train\_data\_features)

lda\_predictions\_f <- predict(lda\_model\_f, newdata = test\_data\_features)$class

lda\_conf\_matrix\_f <- table(test\_data\_features$HeartDisease,lda\_predictions\_f)

lda\_conf\_matrix\_f

A screenshot of a computer code

Description automatically generated

#evaluating lda model

lda\_accuracy\_f <- sum(diag(lda\_conf\_matrix\_f))/sum(lda\_conf\_matrix\_f)

lda\_accuracy\_f

lda\_precision\_f <- lda\_conf\_matrix\_f[2,2]/(lda\_conf\_matrix\_f[2,1]+lda\_conf\_matrix\_f[2,2])

lda\_precision\_f

lda\_recall\_f <- lda\_conf\_matrix\_f[2,2]/(lda\_conf\_matrix\_f[2,2]+lda\_conf\_matrix\_f[1,2])

lda\_recall\_f

lda\_specifi\_f <- lda\_conf\_matrix\_f[1,1]/(lda\_conf\_matrix\_f[1,1]+lda\_conf\_matrix\_f[2,1])

lda\_specifi\_f

lda\_f1\_score\_f <- 2 \* ((lda\_precision\_f \* lda\_recall\_f) / (lda\_precision\_f + lda\_recall\_f))

lda\_f1\_score\_f

A close-up of a computer code

Description automatically generated

# Quadratic Discriminant Analysis (QDA)

qda\_model\_f <- qda(HeartDisease ~ ., data = train\_data\_features)

qda\_predictions\_f <- predict(qda\_model\_f, newdata = test\_data\_features)$class

qda\_conf\_matrixS\_f <- table(test\_data\_features$HeartDisease,qda\_predictions\_f)

qda\_conf\_matrix\_f

A white background with blue text

Description automatically generated

#data visualization

#Accuracy

model\_names\_f <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

accuracies\_f <- c(log\_accuracy\_f,nb\_accuracy\_f, svm\_accuracy\_f, dt\_accuracy\_f, rf\_accuracy\_f, knn\_accuracy\_f, lda\_accuracy\_f, qda\_accuracy\_f)

accuracy\_data\_f <- data.frame(Model\_f = model\_names\_f, Accuracy\_f = accuracies\_f)

ggplot(accuracy\_data\_f, aes(x = Model\_f, y = Accuracy\_f)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model Accuracies", x = "Model", y = "Accuracy")

A graph of blue rectangular objects

Description automatically generated with medium confidence

#Precision

precision\_f <- c(log\_precision\_f,nb\_precision\_f, svm\_precision\_f, dt\_precision\_f, rf\_precision\_f, knn\_precision\_f, lda\_precision\_f, qda\_precision\_f)

precision\_data\_f <- data.frame(Model = model\_names\_f, Precision = precision\_f)

ggplot(precision\_data\_f, aes(x = Model, y = Precision)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model precision", x = "Model", y = "precision")

A graph of blue rectangular bars

Description automatically generated

#recall

recall\_f <- c(log\_recall\_f,nb\_recall\_f, svm\_recall\_f, dt\_recall\_f, rf\_recall\_f, knn\_recall\_f, lda\_recall\_f, qda\_recall\_f)

recall\_data\_f <- data.frame(Model = model\_names\_f, Recall = recall\_f)

ggplot(recall\_data, aes(x = Model, y = Recall)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model recall", x = "Model", y = " Recall")

A graph of blue rectangular bars

Description automatically generated

#specificity

specificity\_f <- c(log\_specifi\_f,nb\_specifi\_f, svm\_specifi\_f, dt\_specifi\_f, rf\_specifi\_f, knn\_specifi\_f, lda\_specifi\_f, qda\_specifi\_f)

specificity\_data\_f <- data.frame(Model = model\_names\_f, Specificity = specificity\_f )

ggplot(specificity\_data\_f, aes(x = Model, y = specificity )) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model specificity", x = "Model", y = "specificity")

A graph of blue rectangular bars

Description automatically generated

#f1 score

f1\_score\_f <- c(log\_f1\_score\_f,nb\_f1\_score\_f, svm\_f1\_score\_f, dt\_f1\_score\_f, rf\_f1\_score\_f, knn\_f1\_score\_f, lda\_f1\_score\_f, qda\_f1\_score\_f)

f1\_score\_data\_f <- data.frame(Model = model\_names\_f, F1\_score = f1\_score\_f )

ggplot(f1\_score\_data\_f, aes(x = Model, y = F1\_score )) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Model f1\_score", x = "Model", y = "f1\_score")

A graph of blue rectangular objects

Description automatically generated

#evaluating models

log\_eval <- sum(log\_accuracy,log\_precision,log\_recall,log\_specifi,log\_f1\_score)

log\_eval\_f <- sum(log\_accuracy\_f,log\_precision\_f,log\_recall\_f,log\_specifi\_f,log\_f1\_score\_f)

nb\_eval <- sum(nb\_accuracy,nb\_precision,nb\_specifi,nb\_recall,nb\_f1\_score)

nb\_eval\_f <- sum(nb\_accuracy\_f,nb\_precision\_f,nb\_specifi\_f,nb\_recall\_f,nb\_f1\_score\_f)

svm\_eval <- sum(svm\_accuracy,svm\_precision,svm\_specifi,svm\_recall,svm\_f1\_score)

svm\_eval\_f <- sum(svm\_accuracy\_f,svm\_precision\_f,svm\_specifi\_f,svm\_recall\_f,svm\_f1\_score\_f)

dt\_eval <- sum(dt\_accuracy,dt\_precision,dt\_specifi,dt\_recall,dt\_f1\_score)

dt\_eval\_f <- sum(dt\_accuracy\_f,dt\_precision\_f,dt\_specifi\_f,dt\_recall\_f,dt\_f1\_score\_f)

rf\_eval <- sum(rf\_accuracy,rf\_precision,rf\_specifi,rf\_recall,rf\_f1\_score)

rf\_eval\_f <- sum(rf\_accuracy\_f,rf\_precision\_f,rf\_specifi\_f,rf\_recall\_f,rf\_f1\_score\_f)

knn\_eval <- sum(knn\_accuracy,knn\_precision,knn\_specifi,knn\_recall,knn\_f1\_score)

knn\_eval\_f <- sum(knn\_accuracy\_f,knn\_precision\_f,knn\_specifi\_f,knn\_recall\_f,knn\_f1\_score\_f)

lda\_eval <- sum(lda\_accuracy,lda\_precision,lda\_specifi,lda\_recall,lda\_f1\_score)

lda\_eval\_f <- sum(lda\_accuracy\_f,lda\_precision\_f,lda\_specifi\_f,lda\_recall\_f,lda\_f1\_score\_f)

qda\_eval <- sum(qda\_accuracy,qda\_precision,qda\_specifi,qda\_recall,qda\_f1\_score)

qda\_eval\_f <- sum(qda\_accuracy\_f,qda\_precision\_f,qda\_specifi\_f,qda\_recall\_f,qda\_f1\_score\_f)

#visualizing

model\_names <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

visualizing <- c(log\_eval,nb\_eval, svm\_eval, dt\_eval, rf\_eval, knn\_eval, lda\_eval, qda\_eval)

visual\_data <- data.frame(Model = model\_names, Eval = visualizing)

ggplot(visual\_data, aes(x = Model, y = Eval)) +

geom\_bar(stat = "identity", fill = "pink") +

labs(title = "Model evaluation", x = "Model", y = "Evaluation")

A graph of a graph

Description automatically generated with medium confidence

model\_names\_f <- c("logistic regression", "naive bayes", "svm", "decision tree", "random forest","knn", "lda", "qda")

visualizing\_f <- c(log\_eval\_f,nb\_eval\_f, svm\_eval\_f, dt\_eval\_f, rf\_eval\_f, knn\_eval\_f, lda\_eval\_f, qda\_eval\_f)

visual\_data\_f <- data.frame(Model = model\_names\_f, Eval = visualizing\_f)

ggplot(visual\_data\_f, aes(x = Model, y = Eval)) +

geom\_bar(stat = "identity", fill = "pink") +

labs(title = "Model evaluation with selected features", x = "Model", y = "Evaluation")

A graph of pink rectangular bars

Description automatically generated with medium confidence

CONCLUSION:

knn model works really well with selected features : FastingBS, RestingECG, MaxHR, ExerciseAngina

linear discriminant analysis works well with all the features