Analiza dhe Parashikimi i Diabetit

Sami Hoxha

2025-06-30

1) Instalimi i paketave të nevojshme

```
# Paketat
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
##
## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(class)
library(e1071)
```

2) Ngarkimi i datasetit dhe informacione të përgjithshme rreth tij

```r
# Leximi i datasetit
data <- read.csv("diabetes.csv")

# Informacione të përgjithshme rreth tij
head(data)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

```r
str(data)
```

```
## ’data.frame’:    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

```r
summary(data)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

3) Pastrimi dhe përgatitja e të dhënave (Data preprocessing)

```r
# Kontrollojmë për vlera 0 në kolonat që nuk duhet të ketë 0
sapply(data, function(x) sum(x == 0))
```

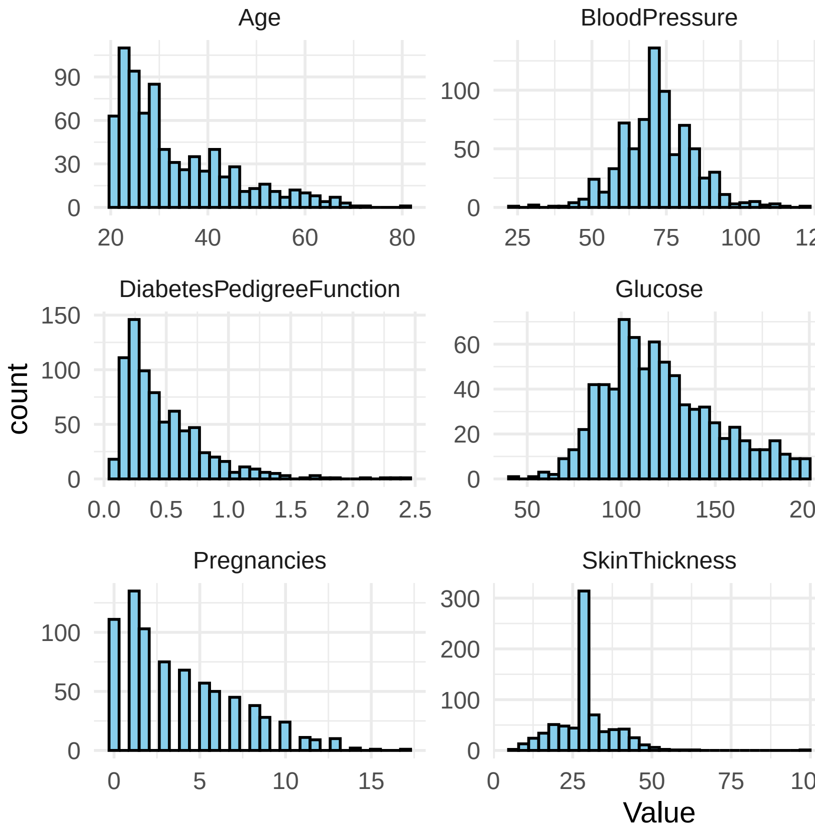
```
##              Pregnancies                  Glucose            BloodPressure
##                       111                       5                      35
##             SkinThickness                 Insulin                     BMI
##                       227                     374                      11
## DiabetesPedigreeFunction                     Age                 Outcome
##                         0                       0                     500
```

```r
# Zëvendësojmë 0 me NA ku nuk ka kuptim
na_cols <- c("Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI")
for (col in na_cols) {
  data[[col]][data[[col]] == 0] <- NA
}

# Plotëso NA me modën
for (col in na_cols) {
  data[[col]][is.na(data[[col]])] <- median(data[[col]], na.rm = TRUE)
}

# Kontrollojmë të dhënat
summary(data)
```

```
##   Pregnancies        Glucose       BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 44.00   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.75   1st Qu.: 64.00   1st Qu.:25.00
##  Median : 3.000   Median :117.00   Median : 72.00   Median :29.00
##  Mean   : 3.845   Mean   :121.66   Mean   : 72.39   Mean   :29.11
##  3rd Qu.: 6.000   3rd Qu.:140.25   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.00   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   : 14.0   Min.   :18.20   Min.   :0.0780           Min.   :21.00
##  1st Qu.:121.5   1st Qu.:27.50   1st Qu.:0.2437           1st Qu.:24.00
##  Median :125.0   Median :32.30   Median :0.3725           Median :29.00
##  Mean   :140.7   Mean   :32.46   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

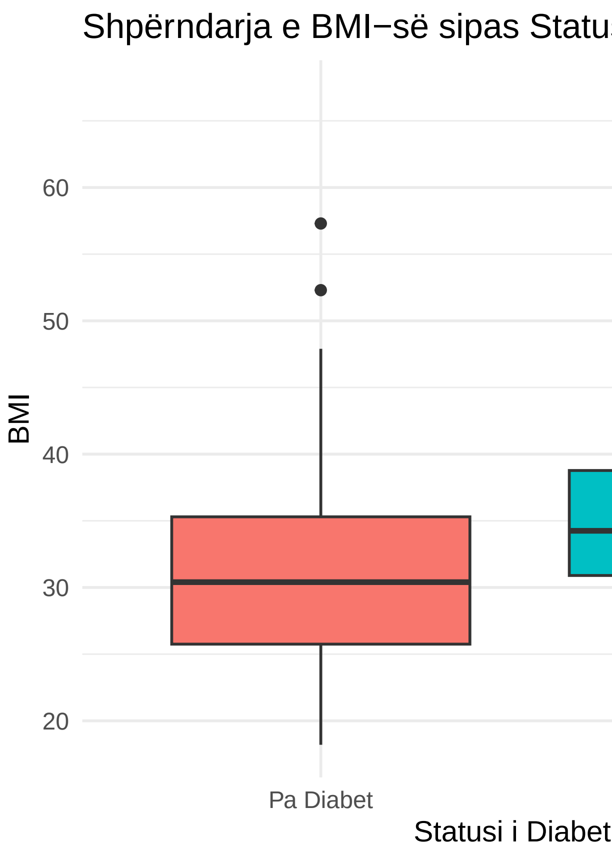
4) Analiza eksploruese e të dhënave (EDA)

```r
data %>%
  gather(key = "Variable", value = "Value", -Outcome) %>%
  ggplot(aes(x = Value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal()
```



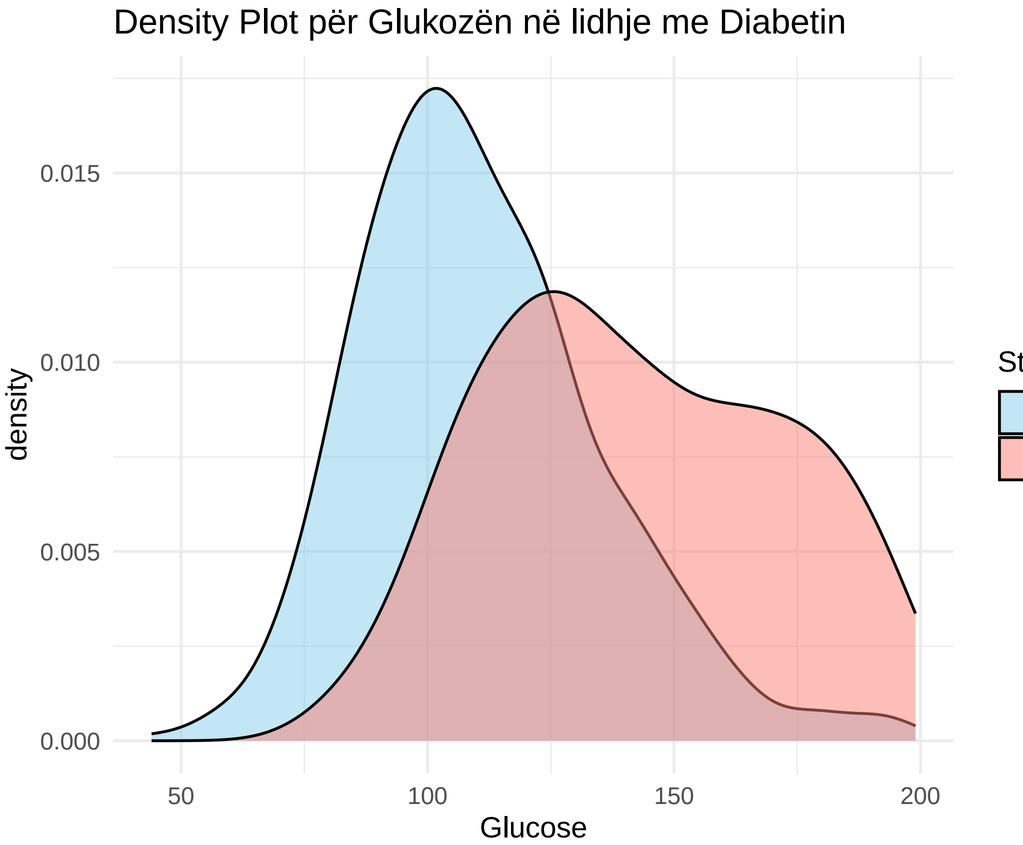
4.1 Histogramet e shpërndarjes së të dhënave

```r
ggplot(data, aes(x = factor(Outcome), y = BMI, fill = factor(Outcome))) +
  geom_boxplot() +
  scale_x_discrete(labels = c("Pa Diabet", "Me Diabet")) +
  labs(
    x = "Statusi i Diabetit",
    y = "BMI",
    title = "Shpërndarja e BMI-së sipas Statusit të Diabetit"
  ) +
  theme_minimal()
```



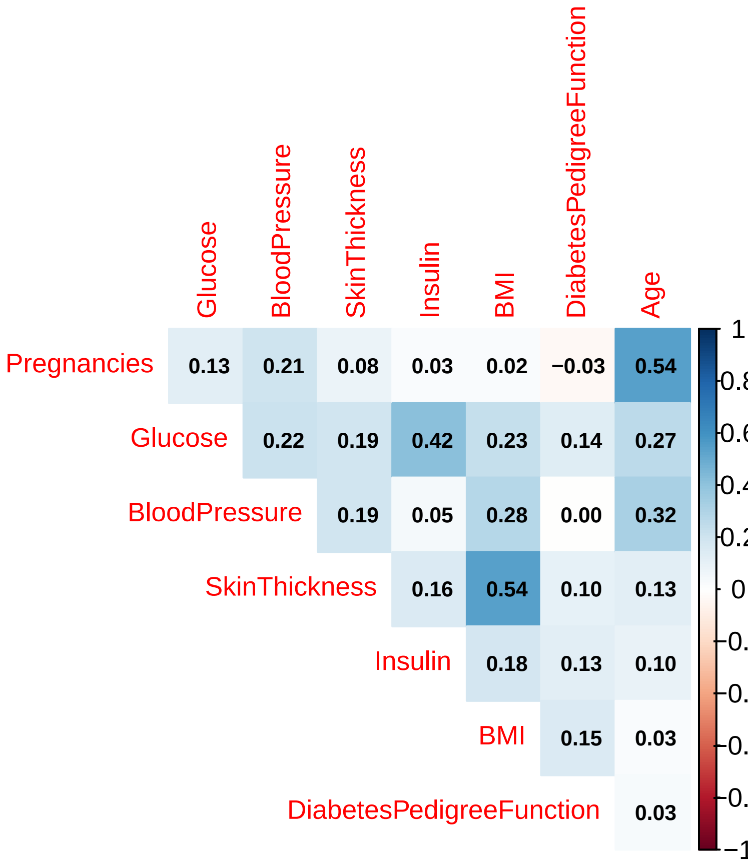
4.2 Boxplot: Shpërndarja e BMI-së sipas Statusit të Diabetit

```r
ggplot(data, aes(x = Glucose, fill = factor(Outcome))) +
  geom_density(alpha = 0.5) +
  labs(
    title = "Density Plot për Glukozën në lidhje me Diabetin",
    fill = "Statusi i Diabetit"
  ) +
  scale_fill_manual(values = c("0" = "skyblue", "1" = "salmon"),
                    labels = c("Pa Diabet", "Me Diabet")) +
  theme_minimal()
```



4.3 Density plot për glukozën

```
cor_mat <- cor(data[, -which(names(data) == "Outcome")])
corrplot(cor_mat, method = "color", type = "upper",
         tl.cex = 0.8, number.cex = 0.7,
         addCoef.col = "black", diag = FALSE)
```



4.4 Matrica e korrelacionit

5) Cluster Analysis

```r
data_features <- data[, setdiff(names(data), "Outcome")]
data_scaled <- scale(data_features)
```

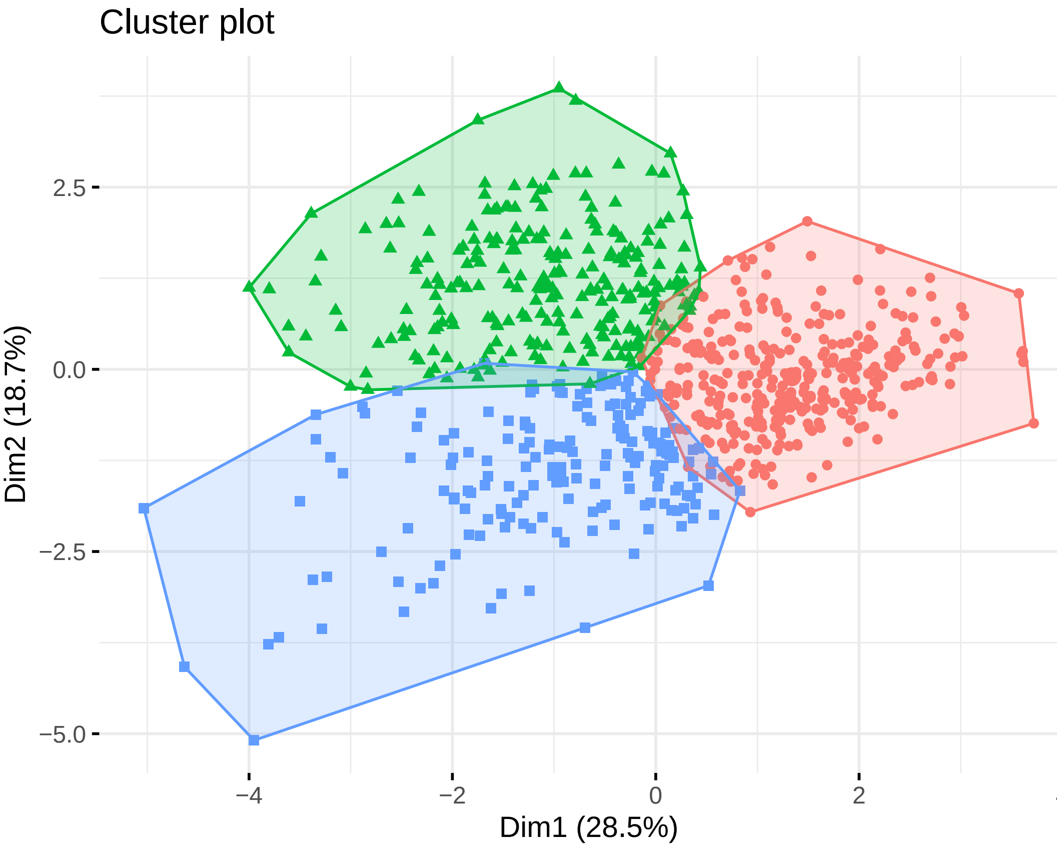
5.1 Përgatitja e veçorive për klaster-at

```r
set.seed(123)
kmeans_result <- kmeans(data_scaled, centers = 3, nstart = 25)
# Kontrollojmë sa raste nga Outcome 0/1 bien në secilin klaster
table(kmeans_result$cluster, data$Outcome)
```

5.2 Klaster K-means me 3 klasterë

```
##
##       0   1
##   1 293  44
##   2 120 128
##   3  87  96
```

```
fviz_cluster(kmeans_result, data = data_scaled,
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_minimal())
```



5.3 Vizualizimi i Klaster-it

5) Ndarja e të dhënave në trajnimi/testim dhe përgatitja e të dhënave

```
set.seed(42)

# Ndarja në train (80%) dhe test (20%)
split <- createDataPartition(data$Outcome, p = 0.8, list = FALSE)
train_data <- data[split, ]
test_data <- data[-split, ]

# Konvertimi i Outcome në faktor me nivelet 0 dhe 1
train_data$Outcome <- factor(train_data$Outcome, levels = c(0, 1))
test_data$Outcome <- factor(test_data$Outcome, levels = c(0, 1))

# Nxjerrja e labels si numerik (0 dhe 1)
train_label <- as.numeric(as.character(train_data$Outcome))
test_label <- as.numeric(as.character(test_data$Outcome))
```

6) Modeli i Regresit Logjistik

```r
# Modeli logjistik
model_log <- glm(Outcome ~ ., data = train_data, family = binomial)
summary(model_log)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -9.361342   0.913228 -10.251  < 2e-16 ***
## Pregnancies               0.151594   0.036912   4.107 4.01e-05 ***
## Glucose                   0.039287   0.004499   8.731  < 2e-16 ***
## BloodPressure            -0.006407   0.009403  -0.681   0.4956
## SkinThickness             0.013941   0.014921   0.934   0.3501
## Insulin                  -0.001585   0.001268  -1.251   0.2111
## BMI                       0.085774   0.020154   4.256 2.08e-05 ***
## DiabetesPedigreeFunction  0.699130   0.330471   2.116   0.0344 *
## Age                       0.009463   0.010788   0.877   0.3804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 794.80  on 614  degrees of freedom
## Residual deviance: 566.76  on 606  degrees of freedom
## AIC: 584.76
##
## Number of Fisher Scoring iterations: 5
```

```r
# Parashikimi me anë të modelit
probabilities <- predict(model_log, test_data, type = "response")
predictions <- ifelse(probabilities > 0.5, 1, 0)

# Vlerësimi i modelit
confusionMatrix(as.factor(predictions), as.factor(test_data$Outcome))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 87 23
##          1 12 31
##
##                Accuracy : 0.7712
##                  95% CI : (0.6965, 0.8352)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.0006263
##
##                   Kappa : 0.4748
```

```
##
##  Mcnemar's Test P-Value : 0.0909689
##
##             Sensitivity : 0.8788
##             Specificity : 0.5741
##          Pos Pred Value : 0.7909
##          Neg Pred Value : 0.7209
##              Prevalence : 0.6471
##          Detection Rate : 0.5686
##    Detection Prevalence : 0.7190
##       Balanced Accuracy : 0.7264
##
##        'Positive' Class : 0
##
```

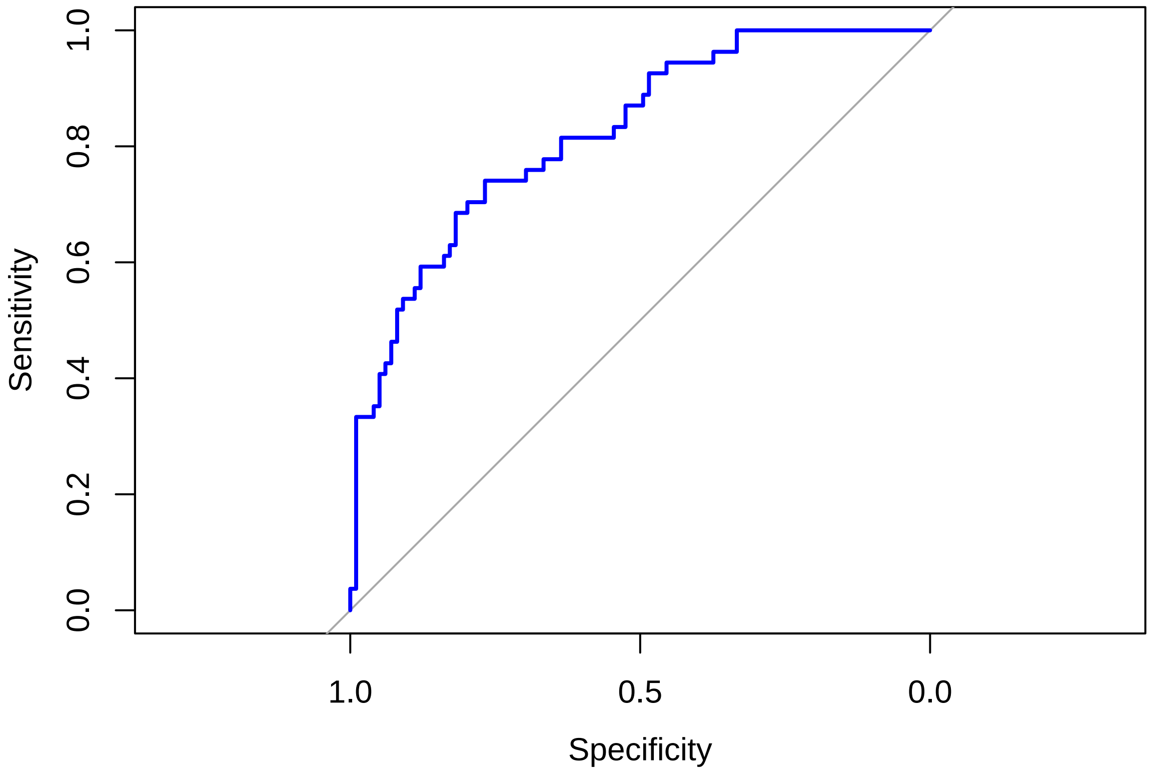
7) Vlerësimi i performancës (ROC, AUC) për modelin e regresit logjistik

```
roc_obj <- roc(test_data$Outcome, probabilities)
```

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

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

```
plot(roc_obj, col = "blue")
```



```r
auc(roc_obj)
```

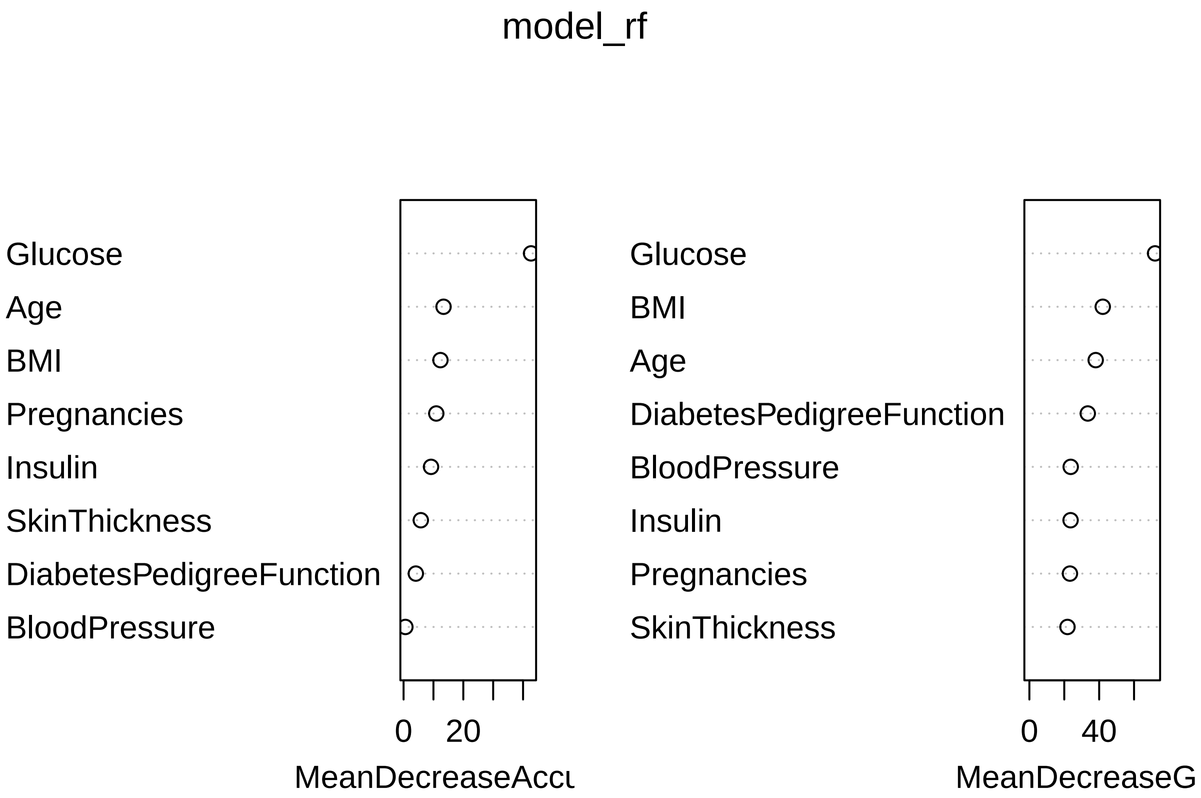
```
## Area under the curve: 0.8223
```

8) Modeli Random Forest

```r
set.seed(42)
model_rf <- randomForest(as.factor(Outcome) ~ ., data = train_data, importance = TRUE)

# Parashikimi
pred_rf <- predict(model_rf, test_data)

# Vlerësimi i modelit
confusionMatrix(pred_rf, as.factor(test_data$Outcome))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 87 21
##          1 12 33
##
##                Accuracy : 0.7843
##                  95% CI : (0.7106, 0.8466)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.0001631
##
##                   Kappa : 0.5092
##
##  Mcnemar’s Test P-Value : 0.1637344
##
##             Sensitivity : 0.8788
##             Specificity : 0.6111
##          Pos Pred Value : 0.8056
##          Neg Pred Value : 0.7333
##              Prevalence : 0.6471
##          Detection Rate : 0.5686
##    Detection Prevalence : 0.7059
##       Balanced Accuracy : 0.7449
##
##        ’Positive’ Class : 0
##
```

```r
# Rëndësia e veçorive
varImpPlot(model_rf)
```



9) Modeli KNN (me normalizim të dhënash)

```
# Normalizojmë veçoritë (pa Outcome)
normalize <- function(x) { (x - min(x)) / (max(x) - min(x)) }
train_norm <- as.data.frame(lapply(train_data[,-9], normalize))
test_norm  <- as.data.frame(lapply(test_data[,-9], normalize))

train_labels <- train_data$Outcome
test_labels  <- test_data$Outcome

# KNN me k=5
pred_knn <- knn(train = train_norm, test = test_norm, cl = train_labels, k = 5)

# Vlerësimi i modelit
confusionMatrix(pred_knn, as.factor(test_labels))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0  1
##          0 82 22
##          1 17 32
##
##                Accuracy : 0.7451
##                  95% CI : (0.6684, 0.812)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.006123
##
##                   Kappa : 0.4299
##
##  Mcnemar’s Test P-Value : 0.521839
##
##             Sensitivity : 0.8283
##             Specificity : 0.5926
##          Pos Pred Value : 0.7885
##          Neg Pred Value : 0.6531
##              Prevalence : 0.6471
##          Detection Rate : 0.5359
##    Detection Prevalence : 0.6797
##       Balanced Accuracy : 0.7104
##
##        ’Positive’ Class : 0
##
```

10) Modeli SVM

```
# Trajnimi i modelit SVM
model_svm <- svm(Outcome ~ ., data = train_data, kernel = "radial", probability = TRUE)

# Parashikimi mbi test set
pred_svm <- predict(model_svm, newdata = test_data)

# Vlerësimi i modelit me matricën e konfuzionit
conf_matrix <- confusionMatrix(pred_svm, test_labels)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 87 25
##          1 12 29
##
##                Accuracy : 0.7582
##                  95% CI : (0.6824, 0.8237)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.002092
##
##                   Kappa : 0.4399
##
##  Mcnemar’s Test P-Value : 0.048520
##
##             Sensitivity : 0.8788
##             Specificity : 0.5370
##          Pos Pred Value : 0.7768
##          Neg Pred Value : 0.7073
##               Prevalence : 0.6471
##           Detection Rate : 0.5686
##     Detection Prevalence : 0.7320
##        Balanced Accuracy : 0.7079
##
##         ’Positive’ Class : 0
##
```

11) Modeli XG-boost

```r
# Konvertimi i Outcome nga faktor/karakter në numeric 0/1 për XGBoost
train_label <- as.numeric(as.character(train_data$Outcome))
test_label  <- as.numeric(as.character(test_data$Outcome))

# Nxjerrja e veçorive (pa Outcome)
train_matrix <- as.matrix(train_data[, setdiff(names(train_data), "Outcome")])
test_matrix  <- as.matrix(test_data[, setdiff(names(test_data), "Outcome")])

# Krijimi i objekteve DMatrix për XGBoost
dtrain <- xgb.DMatrix(data = train_matrix, label = train_label)
dtest  <- xgb.DMatrix(data = test_matrix, label = test_label)

# Parametrat e modelit
params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  eval_metric = "auc",
  eta = 0.1,
  max_depth = 6,
  min_child_weight = 1,
  gamma = 0,
  subsample = 0.8,
  colsample_bytree = 0.8
)

# Watchlist për monitorim gjatë trajnimit
watchlist <- list(train = dtrain, eval = dtest)

# Trajnimi i modelit me early stopping
set.seed(42)
model_xgb <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 100,
  watchlist = watchlist,
  early_stopping_rounds = 10,
  print_every_n = 10,
  maximize = TRUE
)
```

```
## [1]  train-auc:0.866613  eval-auc:0.709035
## Multiple eval metrics are present. Will use eval_auc for early stopping.
## Will train until eval_auc hasn’t improved in 10 rounds.
##
## [11] train-auc:0.955969  eval-auc:0.826038
## [21] train-auc:0.969469  eval-auc:0.822858
## Stopping. Best iteration:
## [13] train-auc:0.957623  eval-auc:0.831650
```

```r
# Parashikimi i probabiliteteve në test
pred_prob <- predict(model_xgb, dtest)

# Konvertimi në klasë 0/1
pred_label <- ifelse(pred_prob > 0.5, 1, 0)
pred_label <- factor(pred_label, levels = c(0,1))
test_label_factor <- factor(test_label, levels = c(0,1))

# Matrica e konfuzionit dhe metrikat e performancës
conf_matrix <- confusionMatrix(pred_label, test_label_factor)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0  1
##          0 87 19
##          1 12 35
##
##                Accuracy : 0.7974
##                  95% CI : (0.7249, 0.858)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 3.671e-05
##
##                   Kappa : 0.5429
##
##  Mcnemar’s Test P-Value : 0.2812
##
##             Sensitivity : 0.8788
##             Specificity : 0.6481
##          Pos Pred Value : 0.8208
##          Neg Pred Value : 0.7447
##              Prevalence : 0.6471
##          Detection Rate : 0.5686
##    Detection Prevalence : 0.6928
##       Balanced Accuracy : 0.7635
##
##        ’Positive’ Class : 0
##
```