Analysis of Breast Cancer Data

Viji Avali

2025-12-10

Analysis of Breast Cancer Wisconsin (Diagnostic) Data from "The UCI Machine Learning Repository"

Capstone Project for "Choose your own project"

```
## Load the needed packages
library(tidyverse)
library(caret)
library(randomForest)
library(ggplot2)
library(GGally)
library(pROC)
library(reshape2)
library(ggcorrplot)
library(gt)
set.seed(123)
```

Load the data

```
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data"
bc_data <- read.csv(url, header = FALSE)
```

##Check the data structure

```
head(bc_data)
```

```
##         V1 V2    V3    V4     V5     V6      V7      V8     V9     V10    V11
## 1   842302  M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.3001 0.14710 0.2419
## 2   842517  M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.0869 0.07017 0.1812
## 3 84300903  M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.1974 0.12790 0.2069
## 4 84348301  M 11.42 20.38  77.58  386.1 0.14250 0.28390 0.2414 0.10520 0.2597
## 5 84358402  M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430 0.1809
## 6   843786  M 12.45 15.70  82.57  477.1 0.12780 0.17000 0.1578 0.08089 0.2087
##       V12    V13    V14   V15    V16      V17     V18     V19     V20     V21
## 1 0.07871 1.0950 0.9053 8.589 153.40 0.006399 0.04904 0.05373 0.01587 0.03003
## 2 0.05667 0.5435 0.7339 3.398  74.08 0.005225 0.01308 0.01860 0.01340 0.01389
```

```
## 3 0.05999 0.7456 0.7869 4.585  94.03 0.006150 0.04006 0.03832 0.02058 0.02250
## 4 0.09744 0.4956 1.1560 3.445  27.23 0.009110 0.07458 0.05661 0.01867 0.05963
## 5 0.05883 0.7572 0.7813 5.438  94.44 0.011490 0.02461 0.05688 0.01885 0.01756
## 6 0.07613 0.3345 0.8902 2.217  27.19 0.007510 0.03345 0.03672 0.01137 0.02165
##        V22   V23   V24    V25    V26    V27    V28    V29    V30    V31     V32
## 1 0.006193 25.38 17.33 184.60 2019.0 0.1622 0.6656 0.7119 0.2654 0.4601 0.11890
## 2 0.003532 24.99 23.41 158.80 1956.0 0.1238 0.1866 0.2416 0.1860 0.2750 0.08902
## 3 0.004571 23.57 25.53 152.50 1709.0 0.1444 0.4245 0.4504 0.2430 0.3613 0.08758
## 4 0.009208 14.91 26.50  98.87  567.7 0.2098 0.8663 0.6869 0.2575 0.6638 0.17300
## 5 0.005115 22.54 16.67 152.20 1575.0 0.1374 0.2050 0.4000 0.1625 0.2364 0.07678
## 6 0.005082 15.47 23.75 103.40  741.6 0.1791 0.5249 0.5355 0.1741 0.3985 0.12440
```

Variable Information from the UCI Dataset

- **ID number**
- **Diagnosis**: M = malignant, B = benign
- **Features (3–32)**: Ten real-valued features computed for each cell nucleus:
  - radius
  - texture
  - perimeter
  - area
  - smoothness
  - compactness
  - concavity
  - concave points
  - symmetry
  - fractal dimension

Use the above info from the dataset to name the variables for better understanding

```
names(bc_data) <- c(
  "id", "diagnosis",
  paste0(rep(c("radius", "texture", "perimeter", "area", "smoothness",
               "compactness", "concavity", "concave_points",
               "symmetry", "fractal_dimension"), each = 3),
         rep(c("_mean", "_se", "_worst"), 10))
)

str(bc_data)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844
##  $ diagnosis              : chr  "M" "M" "M" "M" ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ radius_se              : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ radius_worst           : num  122.8 132.9 130 77.6 135.1 ...
##  $ texture_mean           : num  1001 1326 1203 386 1297 ...
```

```
##  $ texture_se             : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ texture_worst          : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ perimeter_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ perimeter_se           : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ perimeter_worst        : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ area_mean              : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ area_se                : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ area_worst             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ smoothness_mean        : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ smoothness_se          : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_worst       : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_mean       : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ compactness_se         : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ compactness_worst      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ concavity_mean         : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ concavity_se           : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ concavity_worst        : num  25.4 25 23.6 14.9 22.5 ...
##  $ concave_points_mean    : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ concave_points_se      : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ concave_points_worst   : num  2019 1956 1709 568 1575 ...
##  $ symmetry_mean          : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ symmetry_se            : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ symmetry_worst         : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ fractal_dimension_mean : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ fractal_dimension_se   : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```r
table(bc_data$diagnosis)
```

```
##
##   B   M
## 357 212
```

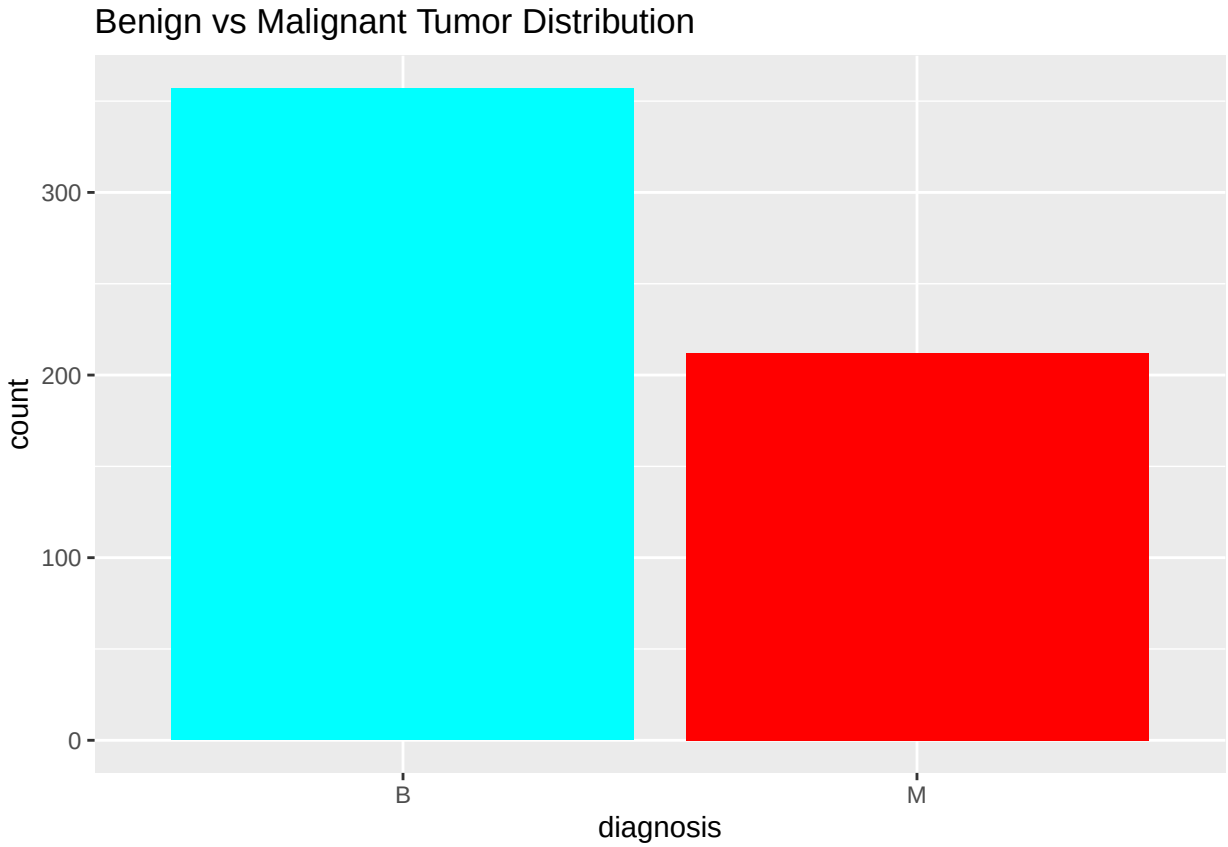
Prepare the data for Analysis

```r
#Convert diagnosis to factor
bc_data$diagnosis <- factor(bc_data$diagnosis, levels = c("B", "M"))

# Remove 'id' column since we do not need that
bc_data <- bc_data %>% select(-id)
```

Exploratory Analysis of data

```r
ggplot(bc_data, aes(diagnosis)) +
  geom_bar(fill = c("cyan", "red")) +
  ggtitle("Benign vs Malignant Tumor Distribution")
```

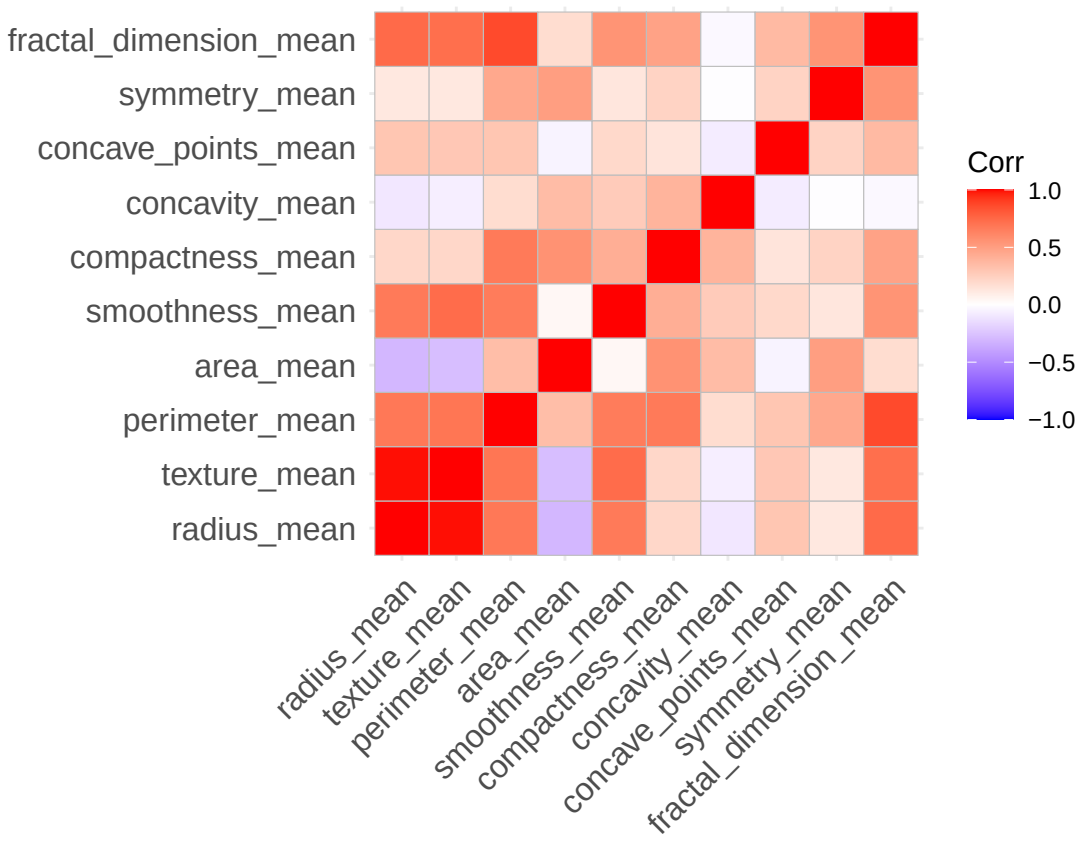



```
summary(bc_data)
```

```
##  diagnosis  radius_mean       radius_se      radius_worst     texture_mean
##  B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##            Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##            Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##            3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##            Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##    texture_se      texture_worst     perimeter_mean     perimeter_se
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##  perimeter_worst    area_mean          area_se         area_worst
##  Min.   :0.1060   Min.   :0.04996   Min.   :0.1115   Min.   :0.3602
##  1st Qu.:0.1619   1st Qu.:0.05770   1st Qu.:0.2324   1st Qu.:0.8339
##  Median :0.1792   Median :0.06154   Median :0.3242   Median :1.1080
##  Mean   :0.1812   Mean   :0.06280   Mean   :0.4052   Mean   :1.2169
##  3rd Qu.:0.1957   3rd Qu.:0.06612   3rd Qu.:0.4789   3rd Qu.:1.4740
##  Max.   :0.3040   Max.   :0.09744   Max.   :2.8730   Max.   :4.8850
##  smoothness_mean  smoothness_se     smoothness_worst   compactness_mean
##  Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
```

```
##  1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
##  Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
##  Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
##  3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
##  Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
##  compactness_se    compactness_worst  concavity_mean      concavity_se
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948
##  1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480
##  Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870
##  Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949
##  3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580
##  Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400
##  concavity_worst concave_points_mean concave_points_se concave_points_worst
##  Min.   : 7.93   Min.   :12.02       Min.   : 50.41    Min.   : 185.2
##  1st Qu.:13.01   1st Qu.:21.08       1st Qu.: 84.11    1st Qu.: 515.3
##  Median :14.97   Median :25.41       Median : 97.66    Median : 686.5
##  Mean   :16.27   Mean   :25.68       Mean   :107.26    Mean   : 880.6
##  3rd Qu.:18.79   3rd Qu.:29.72       3rd Qu.:125.40    3rd Qu.:1084.0
##  Max.   :36.04   Max.   :49.54       Max.   :251.20    Max.   :4254.0
##  symmetry_mean      symmetry_se      symmetry_worst   fractal_dimension_mean
##  Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
##  Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
##  Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
##  3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
##  Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100
##  fractal_dimension_se fractal_dimension_worst
##  Min.   :0.1565       Min.   :0.05504
##  1st Qu.:0.2504       1st Qu.:0.07146
##  Median :0.2822       Median :0.08004
##  Mean   :0.2901       Mean   :0.08395
##  3rd Qu.:0.3179       3rd Qu.:0.09208
##  Max.   :0.6638       Max.   :0.20750
```

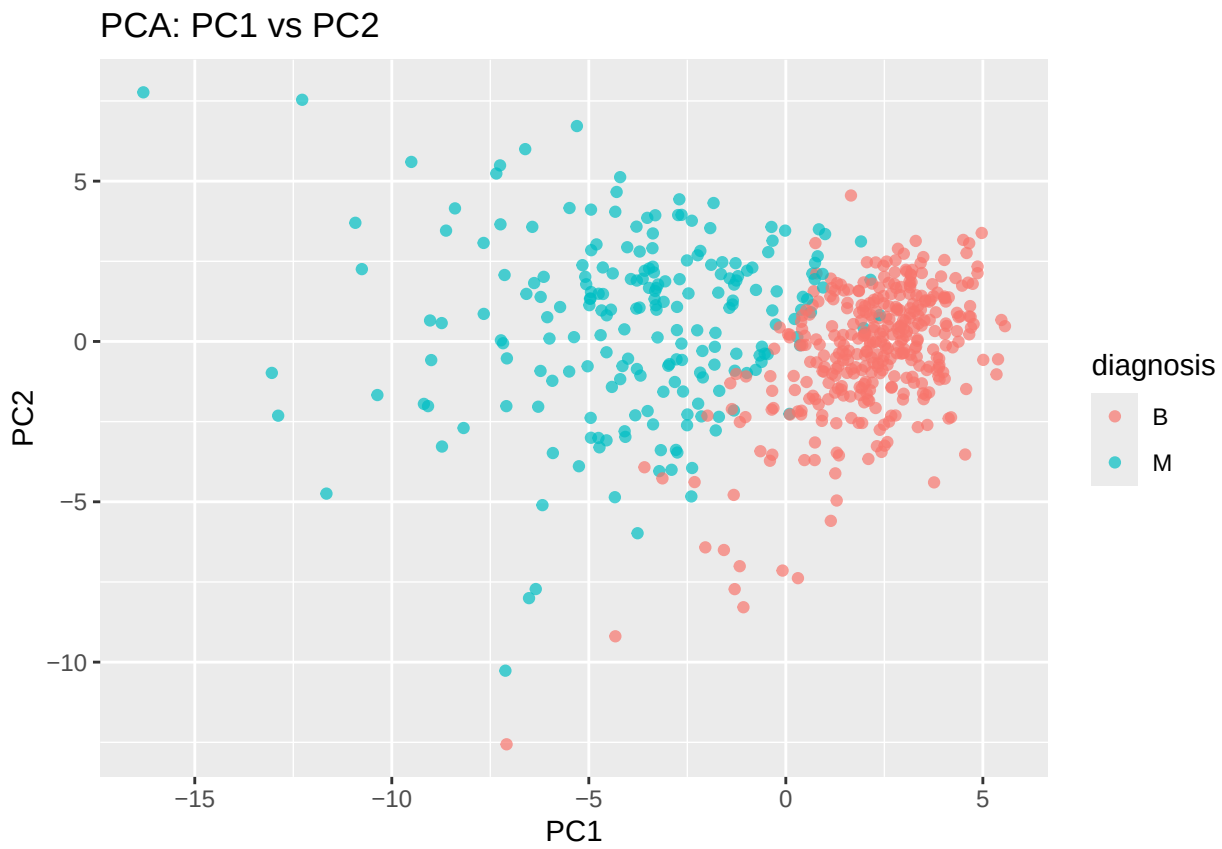
```r
## correlation heatmap
bc_data_mean <- bc_data %>% select(contains("mean"), diagnosis)
corr <- cor(bc_data_mean %>% select(-diagnosis))
ggcorrplot::ggcorrplot(corr)
```




```
##Principal Component Analysis (PCA)

pca <- prcomp(bc_data %>% select(-diagnosis), scale. = TRUE)
pca_df <- data.frame(pca$x[,1:2], diagnosis = bc_data$diagnosis)

ggplot(pca_df, aes(PC1, PC2, color = diagnosis)) +
  geom_point(alpha = .7) +
  ggtitle("PCA: PC1 vs PC2")
```



Split the data into train and test

```
index <- createDataPartition(bc_data$diagnosis, p = 0.8, list = FALSE)
train <- bc_data[index, ]
test <- bc_data[-index, ]
```

Model 1 - Logistic regression

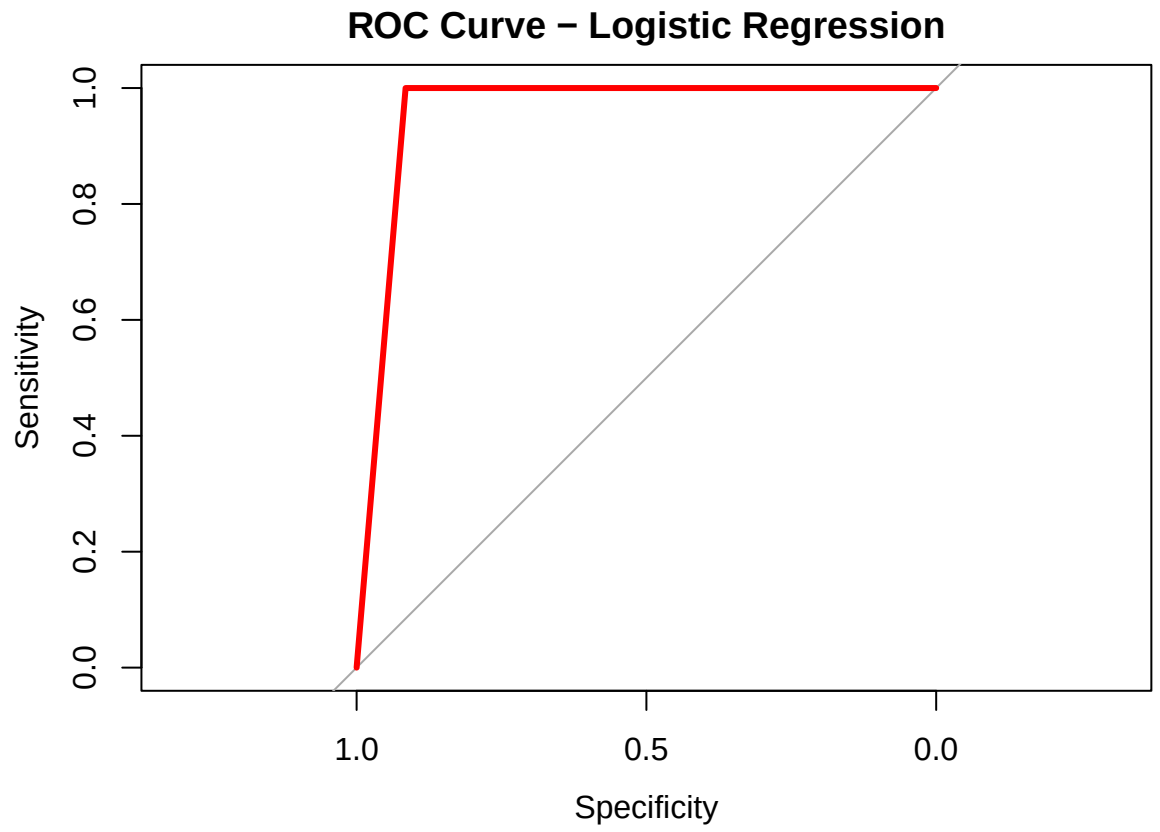
```
log_fit <- train(
  diagnosis ~ .,
  data = train,
  method = "glm"
)

# Classification predictions
log_pred <- predict(log_fit, test)
log_cm <- confusionMatrix(log_pred, test$diagnosis)

# Probabilities for ROC/AUC
log_prob <- predict(log_fit, test, type = "prob")[, "M"]

# ROC curve and AUC
```

```r
roc_obj <- roc(test$diagnosis, log_prob)
plot(roc_obj, col = "red",lwd = 3, main = "ROC Curve - Logistic Regression")
```




```r
log_auc <- auc(roc_obj)

log_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 64  0
##          M  7 42
##
##                Accuracy : 0.9381
##                  95% CI : (0.8765, 0.9747)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : 1.718e-14
##
##                   Kappa : 0.8717
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.9014
##             Specificity : 1.0000
```

```
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8571
##              Prevalence : 0.6283
##          Detection Rate : 0.5664
##    Detection Prevalence : 0.5664
##       Balanced Accuracy : 0.9507
##
##        'Positive' Class : B
##
```

```
log_auc
```

```
## Area under the curve: 0.9577
```

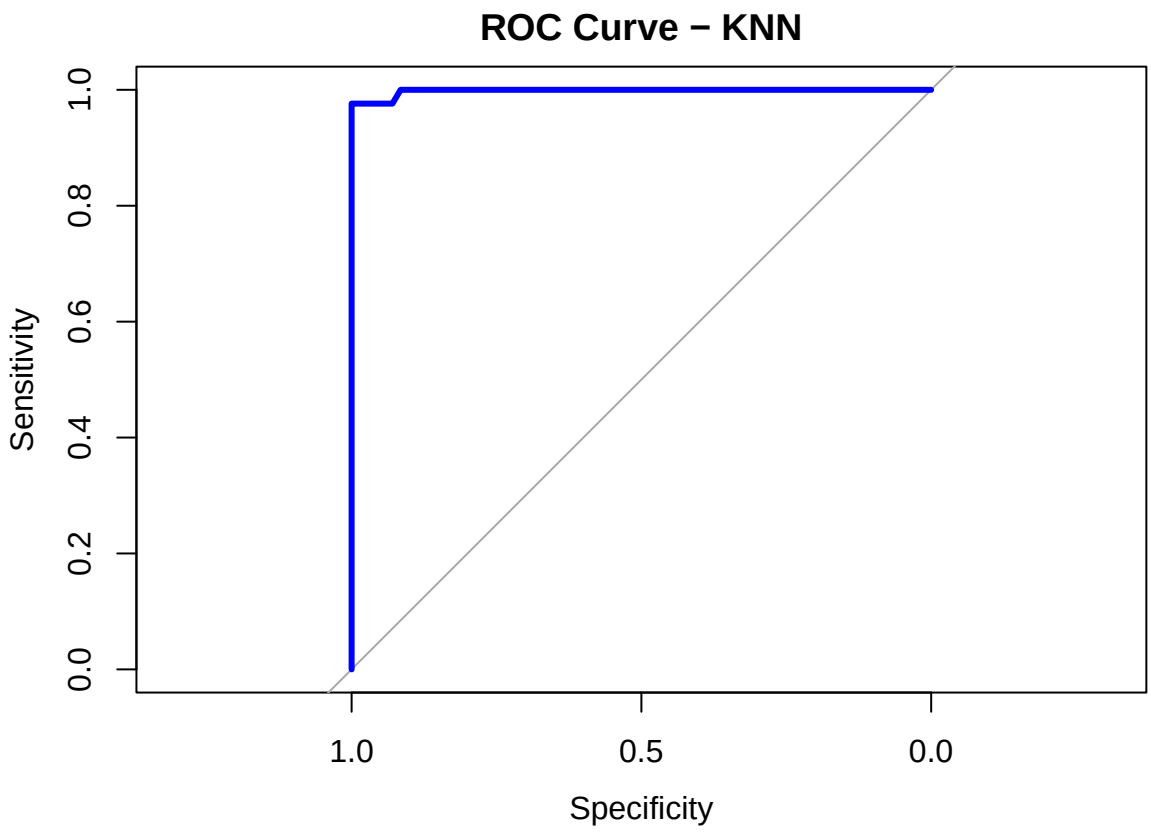
##Model 2 K- Nearest Neighbors

```
# Train KNN
knn_fit <- train(
  diagnosis ~ ., data = train,
  method = "knn",
  tuneGrid = data.frame(k = seq(3, 21, 2)),
  trControl = trainControl(method = "cv", number = 10),
  preProcess = c("center", "scale")   # VERY IMPORTANT FOR KNN
)

# Predictions (class)
knn_pred <- predict(knn_fit, test)
knn_cm <- confusionMatrix(knn_pred, test$diagnosis)

# Probabilities for ROC
knn_prob <- predict(knn_fit, test, type = "prob")[, "M"]

roc_obj <- roc(test$diagnosis, knn_prob)
plot(roc_obj, col = "blue",lwd = 3, main = "ROC Curve - KNN")
```




```
knn_auc <- auc(roc_obj)

knn_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 71  1
##          M  0 41
##
##                Accuracy : 0.9912
##                  95% CI : (0.9517, 0.9998)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.981
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9762
##          Pos Pred Value : 0.9861
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6283
```

```
##          Detection Rate : 0.6283
##    Detection Prevalence : 0.6372
##       Balanced Accuracy : 0.9881
##
##        'Positive' Class : B
##
```

```
knn_auc
```

```
## Area under the curve: 0.9982
```

##Model 3 - Random Forest

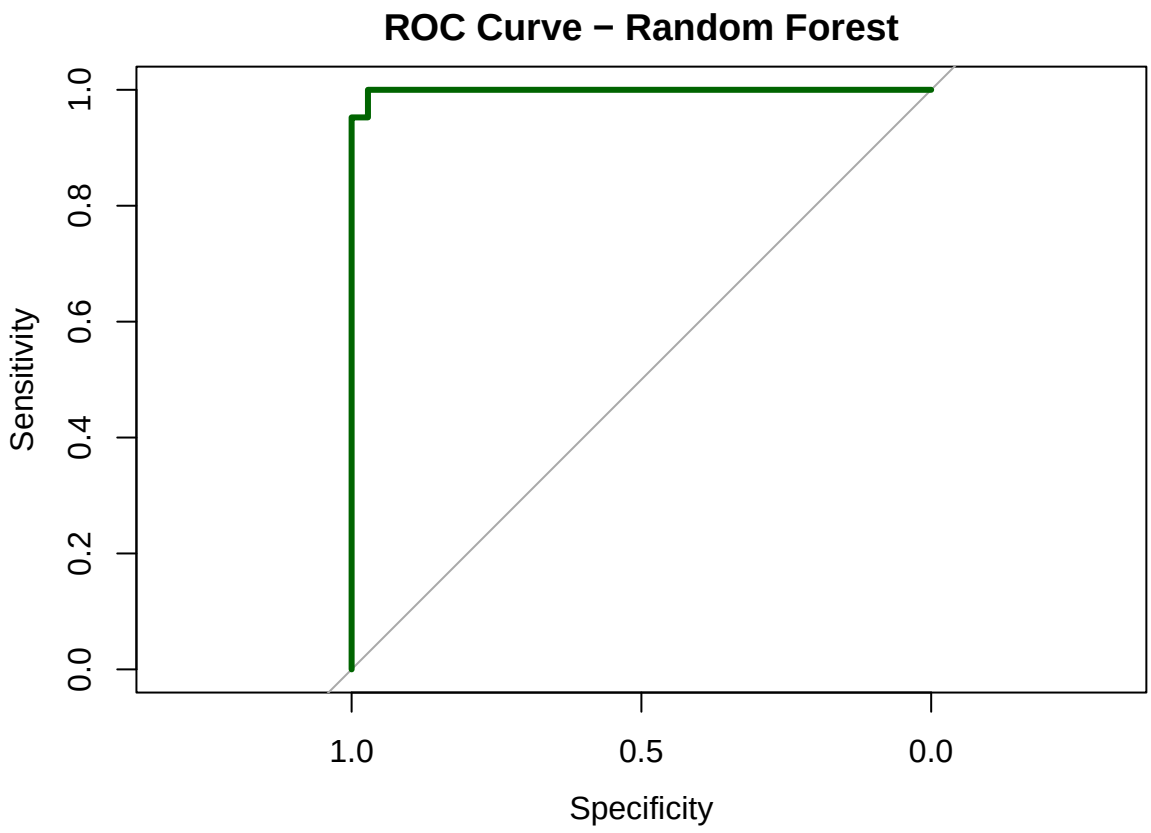
```
rf_fit <- train(
  diagnosis ~ ., data = train,
  method = "rf",
  trControl = trainControl(method = "cv", number = 10),
  importance = TRUE
)

# Train Random Forest
rf_fit <- train(
  diagnosis ~ .,
  data = train,
  method = "rf",
  trControl = trainControl(method = "cv", number = 5),
  importance = TRUE
)

# Predictions (class labels)
rf_pred <- predict(rf_fit, test)
rf_cm <- confusionMatrix(rf_pred, test$diagnosis)

# Probabilities for ROC
rf_prob <- predict(rf_fit, test, type = "prob")[, "M"]

# ROC Curve and AUC
roc_obj <- roc(test$diagnosis, rf_prob)
plot(roc_obj, col = "darkgreen", lwd = 3, main = "ROC Curve - Random Forest")
```




```
rf_auc <- auc(roc_obj)

rf_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 69  1
##          M  2 41
##
##                Accuracy : 0.9735
##                  95% CI : (0.9244, 0.9945)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9434
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9718
##             Specificity : 0.9762
##          Pos Pred Value : 0.9857
##          Neg Pred Value : 0.9535
##              Prevalence : 0.6283
```

```
##          Detection Rate : 0.6106
##    Detection Prevalence : 0.6195
##       Balanced Accuracy : 0.9740
##
##        'Positive' Class : B
##
```

```
rf_auc
```

```
## Area under the curve: 0.9987
```

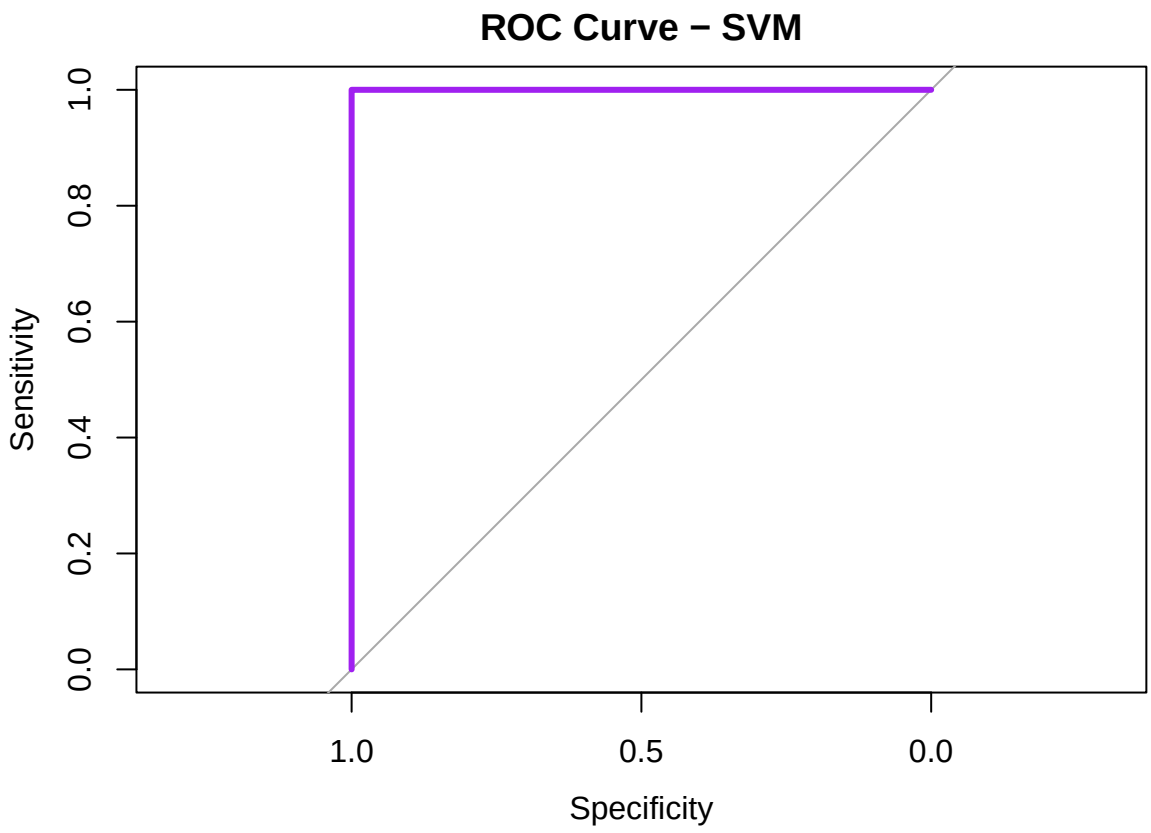
Model 4 - SVM

```
# Train SVM (Radial Kernel)
svm_fit <- train(
  diagnosis ~ .,
  data = train,
  method = "svmRadial",
  trControl = trainControl(method = "cv", number = 10,classProbs = TRUE),
  tuneLength = 10
)

# Predictions
svm_pred <- predict(svm_fit, test)
svm_cm <- confusionMatrix(svm_pred, test$diagnosis)

# Probabilities + ROC
svm_prob <- predict(svm_fit, test, type = "prob")[,"M"]
roc_obj <- roc(test$diagnosis, svm_prob)

plot(roc_obj, col = "purple", lwd = 3, main = "ROC Curve - SVM")
```




```
svm_auc <- auc(roc_obj)

svm_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 70  0
##          M  1 42
##
##                Accuracy : 0.9912
##                  95% CI : (0.9517, 0.9998)
##     No Information Rate : 0.6283
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9811
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9859
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9767
##              Prevalence : 0.6283
```

```
##          Detection Rate : 0.6195
##    Detection Prevalence : 0.6195
##       Balanced Accuracy : 0.9930
##
##        'Positive' Class : B
##
```

```
svm_auc
```

```
## Area under the curve: 1
```

Model comparison

```
results <- data.frame(
  Model = c("Logistic Regression","KNN","Random Forest","SVM"),
  Accuracy = c(log_cm$overall["Accuracy"],
               knn_cm$overall["Accuracy"],
               rf_cm$overall["Accuracy"],
               svm_cm$overall["Accuracy"]),
  Sensitivity = c(log_cm$byClass["Sensitivity"],
                  knn_cm$byClass["Sensitivity"],
                  rf_cm$byClass["Sensitivity"],
                  svm_cm$byClass["Sensitivity"]),
  Specificity = c(log_cm$byClass["Specificity"],
                  knn_cm$byClass["Specificity"],
                  rf_cm$byClass["Specificity"],
                  svm_cm$byClass["Specificity"]),
  AUC = c(log_auc, knn_auc, rf_auc, svm_auc)
)
```

##gt table

```
results %>%
  gt() %>%
  tab_header(
    title = "Model Performance Comparison",
    subtitle = "Accuracy, Sensitivity, Specificity, and AUC "
  ) %>%
  fmt_number(
    columns = c(Accuracy, Sensitivity, Specificity, AUC),
    decimals = 3
    )
```

Model Performance Comparison
Accuracy, Sensitivity, Specificity, and AUC

Model	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression	0.938	0.901	1.000	0.958
KNN	0.991	1.000	0.976	0.998
Random Forest	0.973	0.972	0.976	0.999
SVM	0.991	0.986	1.000	1.000