Breast Cancer Detection Model Selection

Rose Ellison

Table of Contents



# Purpose

The purpose of this program is to find the best model to detect breast cancer. The models will be evaluated using the cumulative accuracy profile (CAP) curve analysis.

# Setup

Libraries to be used

# Libraries to be used  
library(tidyverse)  
library(magrittr)  
library(caTools)  
library(rpart)  
library(randomForest)  
library(class)  
library(e1071)  
library(kableExtra)  
  
# Read the data  
data <- read.csv("./data/breast\_cancer.csv")

# Exploration

* **Sample code number** : id number
* **Clump Thickness** : 1-10
* **Uniformity of Cell Size** : 1-10
* **Uniformity of Cell Shape** : 1-10
* **Marginal Adhesion** : 1-10
* **Single Epithlial Cell Size** : 1-10
* **Bare Nuclei** : 1-10
* **Bland Chromatin** : 1-10
* **Normal Nucleoli** : 1-10
* **Mitoses** : 1-10
* **Class** (2 for benign, 4 for malignant)

# Cleaning the data

# Check for missing values  
sum(is.na(data))

## [1] 0

# Drop the first column for ID numbers  
data <- data[ , -1]  
  
#   
data$Class <- ifelse(data$Class == 4, 1,0)

# Splitting the data  
set.seed(1)  
split <- sample.split(data$Class, SplitRatio = 0.75)  
training <- subset(data, split == TRUE)  
test <- subset(data, split == FALSE)  
  
# Feature Scaling  
training[1:8] <- scale(training[1:8])  
test[1:8] <- scale(test[1:8])

# Classification Models

### Logistic Regression Model

# Fitting Logistic Regression to the training set  
classifier\_logistic\_regression <- glm(Class ~ . ,  
 family = binomial,  
 data = data)  
  
  
# Predicting the test set results  
prob\_pred <- predict(classifier\_logistic\_regression,   
 type = 'response',   
 newdata = test[-10])  
  
y\_pred\_lr <- ifelse(prob\_pred > 0.5, 1, 0)

### K-Nearest\_Neighbors Model

y\_pred\_knn = knn(train = training[, -10],  
 test = test[, -10],  
 cl = training[, 10],  
 k = 5,  
 prob = TRUE)

### Support Vector Machine Model(SVM)

classifier\_svm = svm(formula = Class ~ .,  
 data = training,  
 type = 'C-classification',  
 kernel = 'radial')  
  
# Predicting the Test set results  
y\_pred\_svm = predict(classifier\_svm, newdata = test[-10])

### Kernel Support Vector Machine Model

classifier\_kernel\_svm = svm(formula = Class~ .,  
 data = training,  
 type = 'C-classification',  
 kernel = 'radial')  
  
# Predicting the Test set results  
y\_pred\_kernel\_svm = predict(classifier\_kernel\_svm, newdata = test[-10])

### Naive Bayes Model

classifier\_nb = naiveBayes(x = training[-10],  
 y = training$Class)  
  
# Predicting the Test set results  
y\_pred\_nb = predict(classifier\_nb, newdata = test[-10])

### Random Forest Model

classifier\_rf = randomForest(x = training[-10],  
 y = training$Class,  
 ntree = 10)

## Warning in randomForest.default(x = training[-10], y = training$Class, ntree =  
## 10): The response has five or fewer unique values. Are you sure you want to do  
## regression?

# Predicting the Test set results  
y\_pred\_rf = predict(classifier\_rf, newdata = test[-10])

### Decision Tree Model

training$Class = factor(training$Class, levels = c(0, 1))  
classifier\_dt = rpart(formula = Class ~ .,  
 data = training)  
  
# Predicting the Test set results  
y\_pred\_dt = predict(classifier\_dt, newdata = test[-10], type = 'class')

# Plotting the CAP curve for the models

# Model Selection

# Conclusion