Comparative analysis of Malignancy detection of Breast Cancer using various machine learning Models

Abdulwahid

4/20/2022

Load libraries

The libraries contain functions specific to the requirements of the project and can vary depending on the project needs.

```
library(ellipse)
library(caret)
library(e1071)
library(rattle)
```

Import Dataset

Assign path to filepath variable and load the CSV file from the local path

```
filepath <- "C:/Users/DELL/Desktop/Spring 2021/T and W analytics/breast-cancer-wisconsin.csv"
dataset <- read.csv(filepath, header=TRUE,fileEncoding="UTF-8-BOM")
```

Data Preprocessing

Change attribute Group from character to factor

```
dataset[, 'Group'] <- as.factor(dataset[, 'Group'])
```

Check dimensions of dataset

```
dim(dataset)
```

```
## [1] 683  10
```

List the types for each attribute

```
sapply(dataset, class)
```

```
##        Thickness        Cell.size       Cell.shape         Adhesion
##        "integer"        "integer"        "integer"        "integer"
## Single.cell.size           Nuclei        Chromatin         Nucleoli
##        "integer"        "integer"        "integer"        "integer"
##          Mitoses            Group
##        "integer"         "factor"
```

Summarize attribute distributions

```
summary(dataset)
```

```
##    Thickness        Cell.size        Cell.shape        Adhesion
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.00
##  Median : 4.000   Median : 1.000   Median : 1.000   Median : 1.00
##  Mean   : 4.442   Mean   : 3.151   Mean   : 3.215   Mean   : 2.83
##  3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 4.00
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00
##  Single.cell.size     Nuclei         Chromatin         Nucleoli
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00
##  1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.00
##  Median : 2.000   Median : 1.000   Median : 3.000   Median : 1.00
##  Mean   : 3.234   Mean   : 3.545   Mean   : 3.445   Mean   : 2.87
##  3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.00
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00
##     Mitoses              Group
##  Min.   : 1.000   benign    :444
##  1st Qu.: 1.000   malignancy:239
##  Median : 1.000
##  Mean   : 1.603
##  3rd Qu.: 1.000
##  Max.   :10.000
```

Take a peek at the first 5 rows of the data

```
head(dataset)
```

```
##   Thickness Cell.size Cell.shape Adhesion Single.cell.size Nuclei Chromatin
## 1         5         1          1        1                2      1         3
## 2         5         4          4        5                7     10         3
## 3         3         1          1        1                2      2         3
## 4         6         8          8        1                3      4         3
## 5         4         1          1        3                2      1         3
## 6         8        10         10        8                7     10         9
##   Nucleoli Mitoses      Group
## 1        1       1     benign
## 2        2       1     benign
## 3        1       1     benign
## 4        7       1     benign
## 5        1       1     benign
## 6        7       1 malignancy
```

List the levels for the class

```
levels(dataset$Group)
```

```
## [1] "benign"     "malignancy"
```

Check the percentage distribution of classes

```
percentage <- prop.table(table(dataset$Group)) * 100
cbind(freq=table(dataset$Group), percentage=percentage)
```

```
##            freq percentage
## benign      444   65.00732
## malignancy  239   34.99268
```

Move all all 9 inputs into a dataframe

```
df<- data.frame(dataset$Thickness,dataset$Cell.size,dataset$Cell.shape,dataset$Adhesion,dataset$Single.
```

Change the datatype to numeric for all inputs

```
df<-lapply(df,as.numeric)
```

Recheck the list types for each attribute

```
sapply(df, class)
```

```
##        dataset.Thickness        dataset.Cell.size       dataset.Cell.shape
##                "numeric"                "numeric"                "numeric"
##         dataset.Adhesion dataset.Single.cell.size           dataset.Nuclei
##                "numeric"                "numeric"                "numeric"
##        dataset.Chromatin         dataset.Nucleoli          dataset.Mitoses
##                "numeric"                "numeric"                "numeric"
```

Split input and output

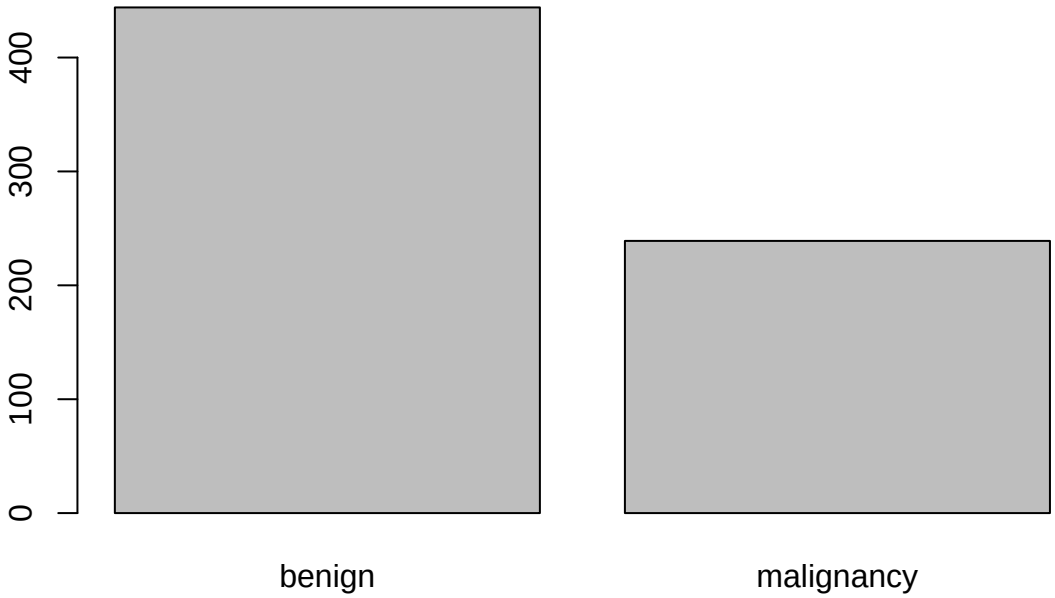
```
x <- df[1:9]
y <- dataset[,10]
```

Exploratory Data Analysis

The process of using data visualizations to get a deeper understanding of the dataset and all features included.

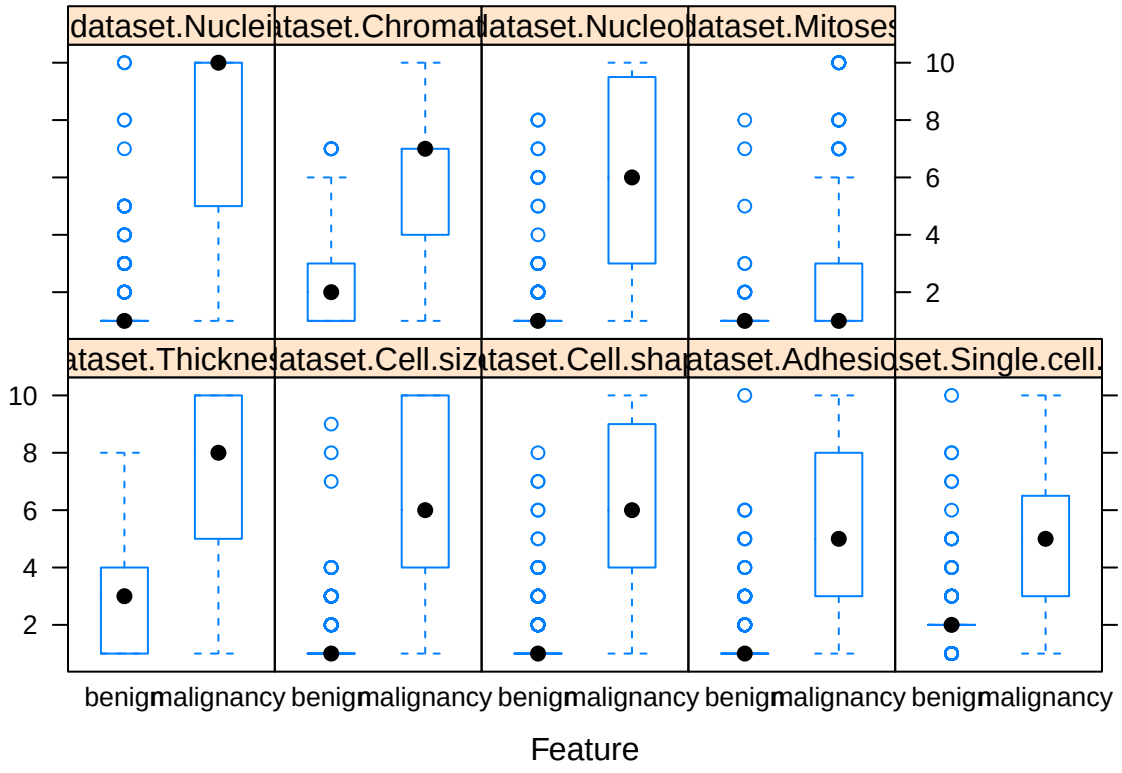
Construct a bar plot for classes (X and Y)

```
plot(y)
```



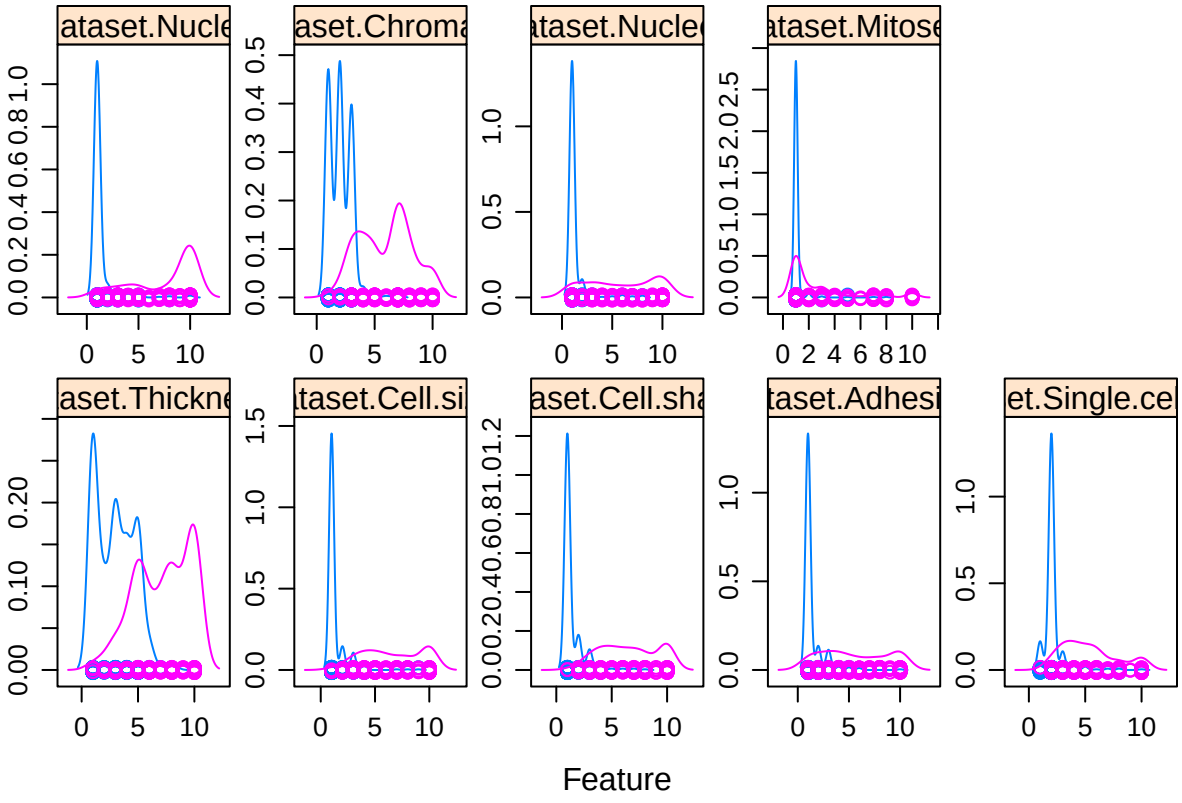
Box plot for for Features & Target variable

```
featurePlot(x=x, y=y, plot="box")
```



Density plot for for Features & Target variable

```
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```



Split Data

The data is divided into training and testing set. Training set will be used to train the model and testing set will be used to assess the model.

Create a list of 80% of the rows in the original dataset we can use for training

```
validation_index <- createDataPartition(dataset$Group, p=0.80, list=FALSE)
```

Select 20% of the data for validation

```
test <- dataset[-validation_index,]
```

Use the remaining 80% of data to training and testing the models

```
train <- dataset[validation_index,]
```

Test the Harness- Run algorithms using 10-fold cross validation

```
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

CART Model

Build Model

```
set.seed(7)
fit.cart <- train(Group~., data=train, method="rpart", metric=metric, trControl=control)
```

Summarize Model

```
print(fit.cart)
```

```
## CART
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00000000  0.9360245  0.8620692
##   0.05729167  0.9250806  0.8365800
##   0.79166667  0.8365753  0.5675150
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.
```

Estimate effectiveness of CART on the test dataset

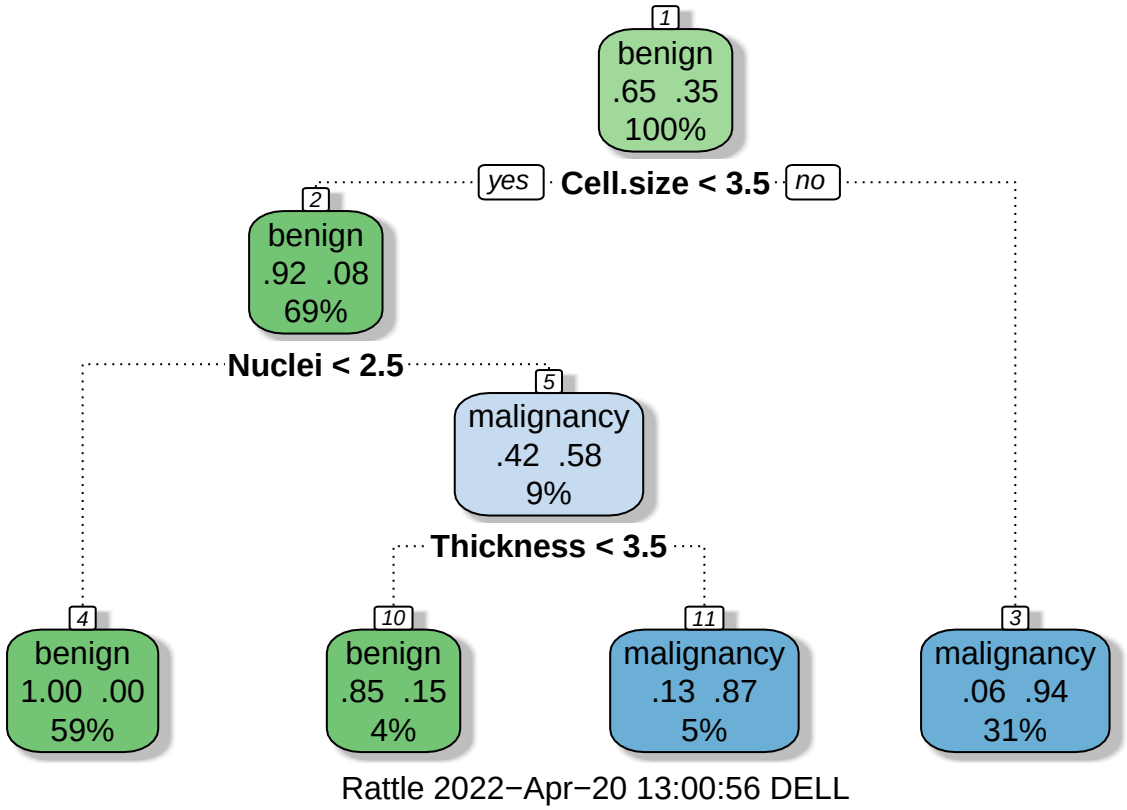
```
predictions <- predict(fit.cart, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         85          2
##   malignancy      3         45
##
##                Accuracy : 0.963
##                  95% CI : (0.9157, 0.9879)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9188
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9659
```

```
##             Specificity : 0.9574
##          Pos Pred Value : 0.9770
##          Neg Pred Value : 0.9375
##              Prevalence : 0.6519
##          Detection Rate : 0.6296
##    Detection Prevalence : 0.6444
##       Balanced Accuracy : 0.9617
##
##        'Positive' Class : benign
##
```

Plot graph for CART

```
fancyRpartPlot(fit.cart$finalModel)
```



LDA Model

Build Model

```
set.seed(7)
fit.lda <- train(Group~., data=train, method="lda", metric=metric, trControl=control)
```

Summarize Model

```
print(fit.lda)
```

```
## Linear Discriminant Analysis
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9561592  0.9024376
```

Estimate effectiveness of LDA on the test dataset

```
predictions <- predict(fit.lda, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         88          3
##   malignancy      0         44
##
##                Accuracy : 0.9778
##                  95% CI : (0.9364, 0.9954)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9503
##
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 1.0000
##             Specificity : 0.9362
##          Pos Pred Value : 0.9670
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6519
##          Detection Rate : 0.6519
##    Detection Prevalence : 0.6741
##       Balanced Accuracy : 0.9681
##
##        'Positive' Class : benign
##
```

KNN Model

Build Model

```
set.seed(7)
fit.knn <- train(Group~., data=train, method="knn", metric=metric, trControl=control)
```

Summarize Model

```
print(fit.knn)
```

```
## k-Nearest Neighbors
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9708057  0.9365565
##   7  0.9670683  0.9284107
##   9  0.9652501  0.9243396
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

Estimate effectiveness of KNN on the test dataset

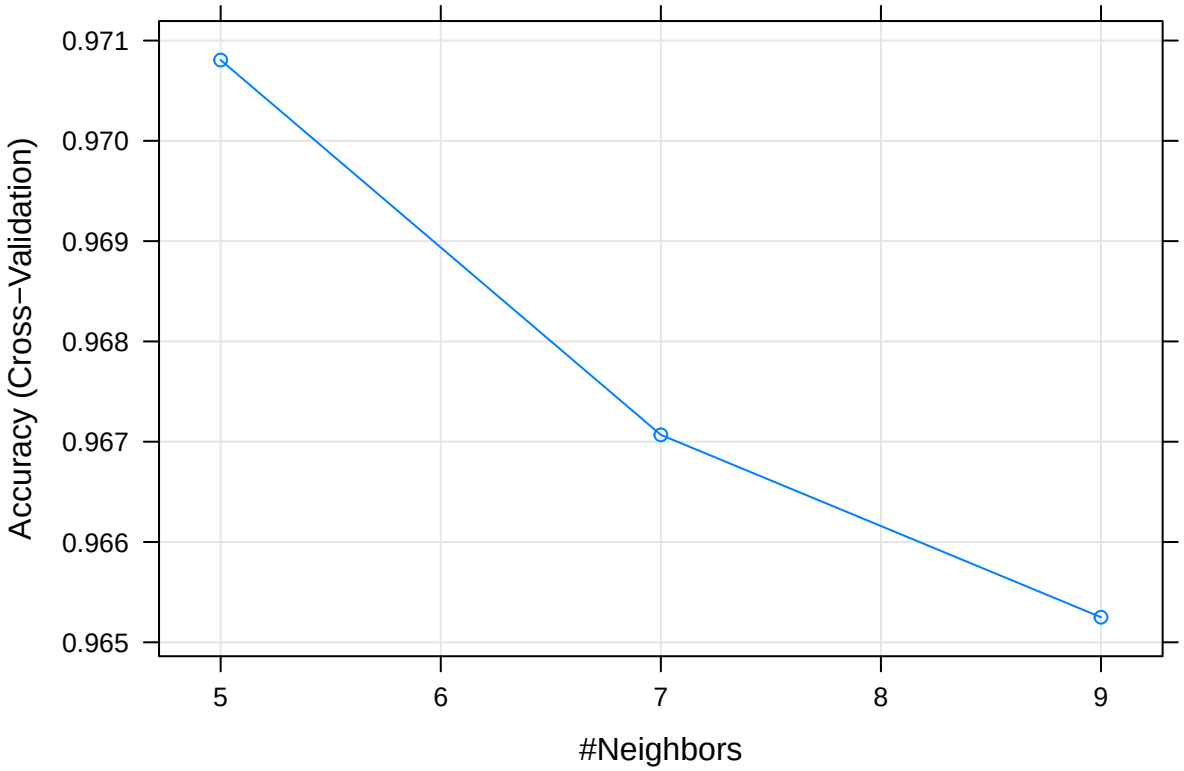
```
predictions <- predict(fit.knn, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         88          1
##   malignancy      0         46
##
##                Accuracy : 0.9926
##                  95% CI : (0.9594, 0.9998)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9836
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9787
##          Pos Pred Value : 0.9888
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6519
```

```
##          Detection Rate : 0.6519
##    Detection Prevalence : 0.6593
##       Balanced Accuracy : 0.9894
##
##        'Positive' Class : benign
##
```

Plot graph for KNN

```
plot(fit.knn)
```



SVM Model

Build Model

```
set.seed(7)
fit.svm <- train(Group~., data=train, method="svmRadial", metric=metric, trControl=control)
```

Summarize Model

```
print(fit.svm)
```

```
## Support Vector Machines with Radial Basis Function Kernel
```

```
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.9378427  0.8698142
##   0.50  0.9378427  0.8698142
##   1.00  0.9396934  0.8729487
##
## Tuning parameter 'sigma' was held constant at a value of 1.015289
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 1.015289 and C = 1.
```

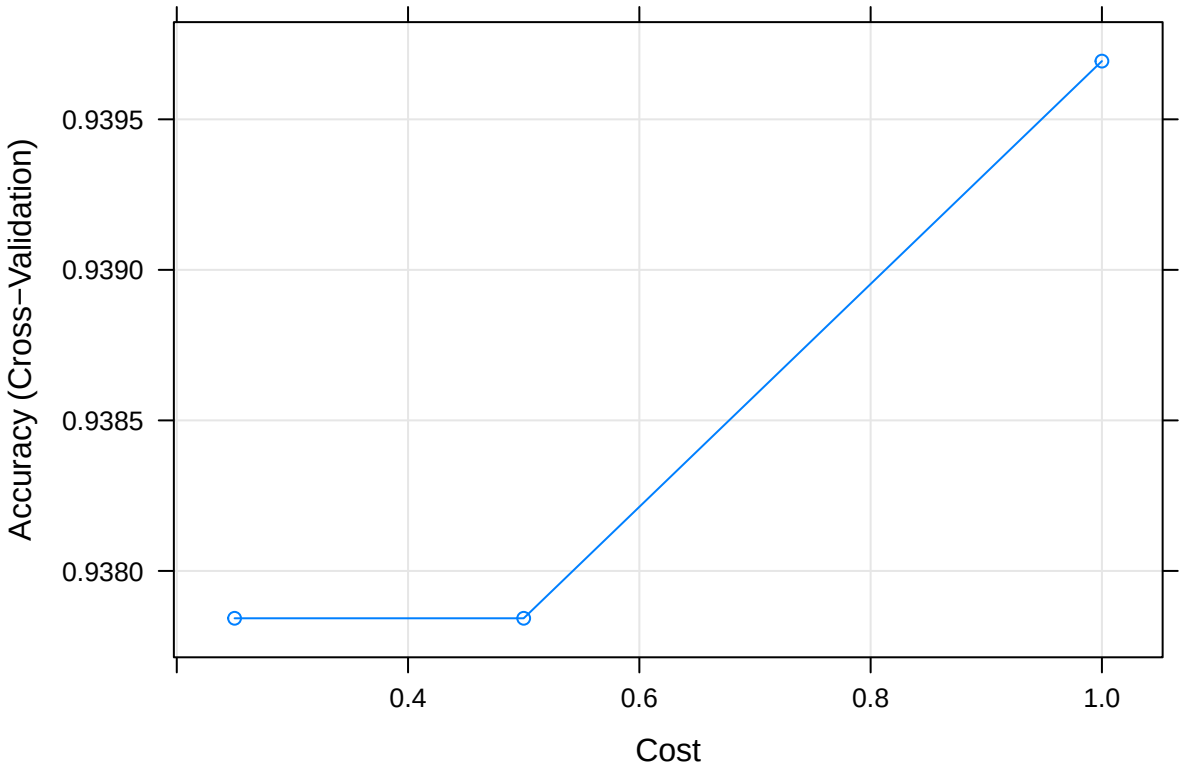
Estimate effectiveness of SVM on the test dataset

```
predictions <- predict(fit.svm, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         84          0
##   malignancy      4         47
##
##                Accuracy : 0.9704
##                  95% CI : (0.9259, 0.9919)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.936
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 0.9545
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9216
##              Prevalence : 0.6519
##          Detection Rate : 0.6222
##    Detection Prevalence : 0.6222
##       Balanced Accuracy : 0.9773
##
##        'Positive' Class : benign
##
```

Plot graph for SVM

```
plot(fit.svm)
```



RF Model

Build Model

```
set.seed(7)
fit.rf <- train(Group~., data=train, method="rf", metric=metric, trControl=control)
```

Summarize Model

```
print(fit.rf)
```

```
## Random Forest
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results across tuning parameters:
```

```
##
##   mtry  Accuracy   Kappa
##   2     0.9690200  0.9325096
##   5     0.9653836  0.9241560
##   9     0.9653836  0.9244422
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

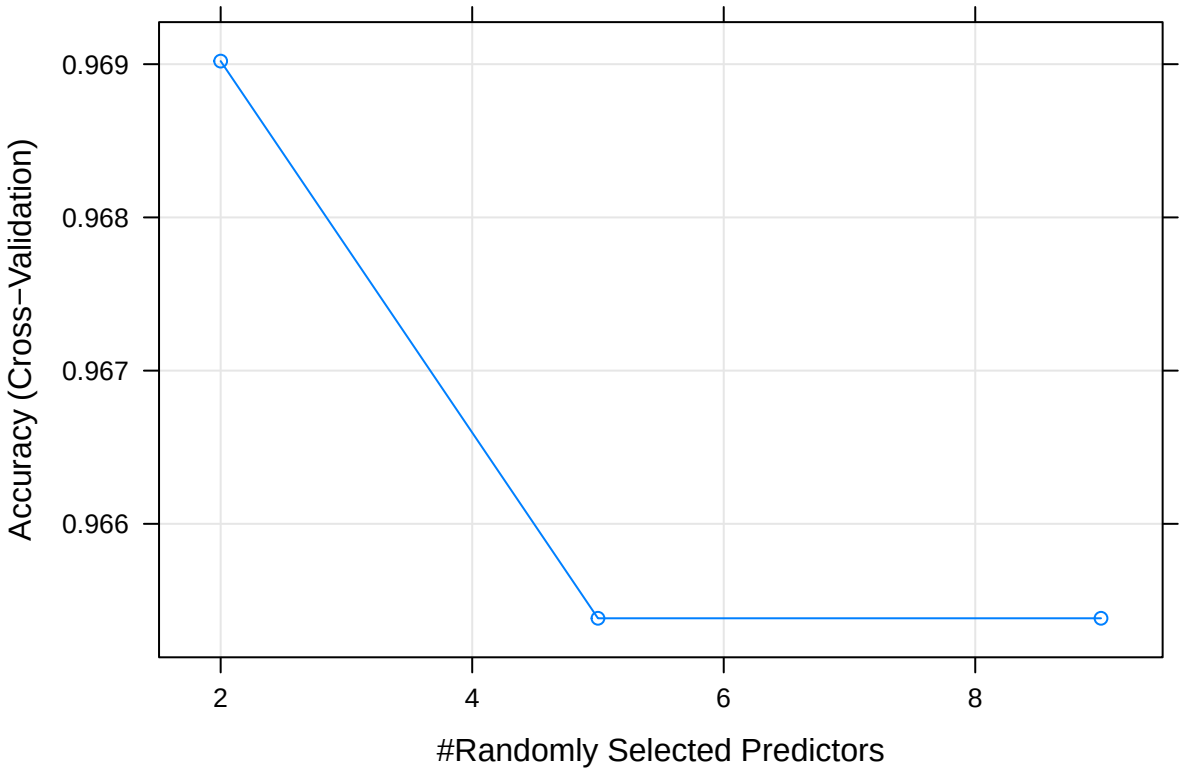
Estimate effectiveness of RF on the test dataset

```
predictions <- predict(fit.rf, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         87          1
##   malignancy      1         46
##
##                Accuracy : 0.9852
##                  95% CI : (0.9475, 0.9982)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9674
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9886
##             Specificity : 0.9787
##          Pos Pred Value : 0.9886
##          Neg Pred Value : 0.9787
##              Prevalence : 0.6519
##          Detection Rate : 0.6444
##    Detection Prevalence : 0.6519
##       Balanced Accuracy : 0.9837
##
##        'Positive' Class : benign
##
```

Plot graph for RF

```
plot(fit.rf)
```



Comparitive Analysis

Comparing various machine learning models to determine the best model fit for the dataset.

Summarize accuracy of models

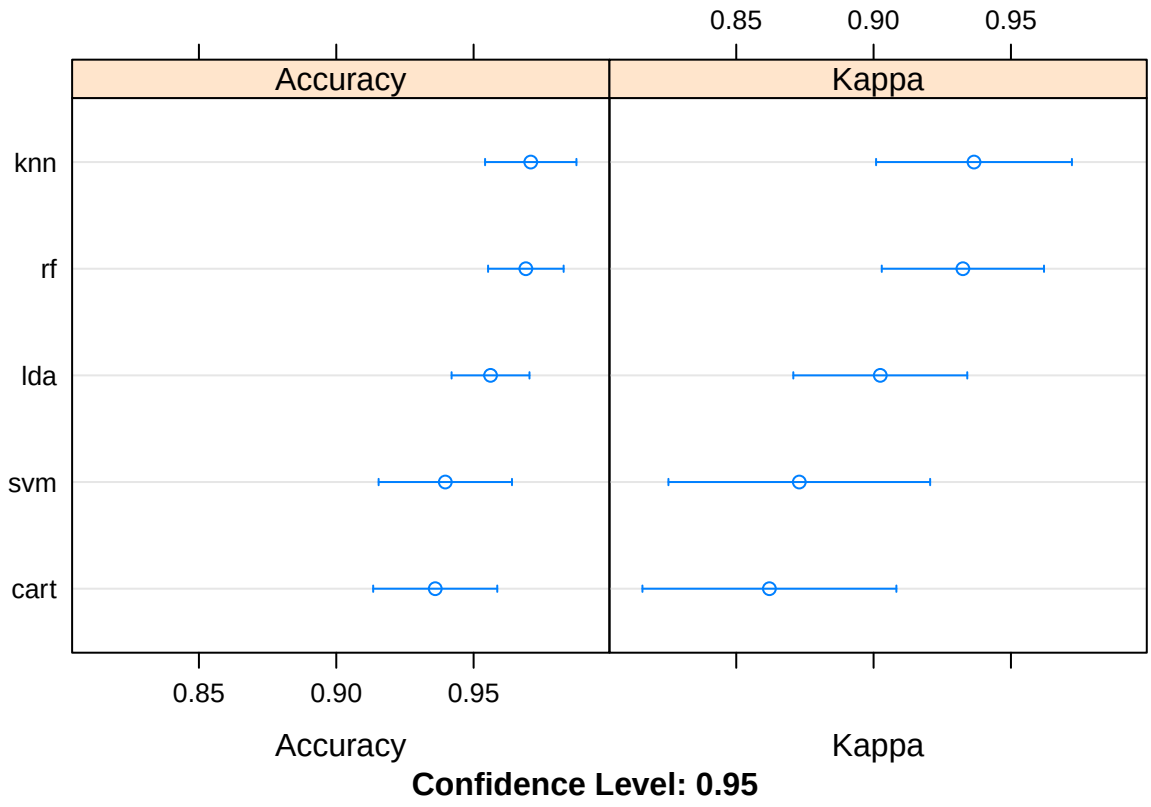
```
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart, knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda  0.9259259 0.9446970 0.9636364 0.9561592 0.9641234 0.9818182    0
## cart 0.8888889 0.9132997 0.9454545 0.9360245 0.9585859 0.9818182    0
## knn  0.9259259 0.9636364 0.9728836 0.9708057 0.9818182 1.0000000    0
## svm  0.8518519 0.9318182 0.9459416 0.9396934 0.9629630 0.9636364    0
```

```
## rf   0.9444444 0.9507305 0.9725589 0.9690200 0.9817340 1.0000000    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda  0.8375940 0.8739050 0.9175412 0.9024376 0.9220518 0.9602888    0
## cart 0.7675753 0.8136388 0.8762918 0.8620692 0.9099957 0.9592894    0
## knn  0.8414097 0.9185411 0.9407558 0.9365565 0.9601869 1.0000000    0
## svm  0.7037037 0.8555456 0.8841682 0.8729487 0.9207048 0.9215407    0
## rf   0.8796434 0.8928468 0.9402224 0.9325096 0.9597332 1.0000000    0
```

Compare accuracy of models

```
dotplot(results)
```



Summarize Best Model

```
print(fit.rf)
```

```
## Random Forest
##
## 548 samples
##   9 predictor
##   2 classes: 'benign', 'malignancy'
##
## No pre-processing
```

```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 493, 492, 493, 493, 493, 493, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9690200  0.9325096
##   5     0.9653836  0.9241560
##   9     0.9653836  0.9244422
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

Estimate effectiveness of RF on the test dataset

```
predictions <- predict(fit.rf, test)
confusionMatrix(predictions, as.factor(test$Group))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   benign malignancy
##   benign         87          1
##   malignancy      1         46
##
##                Accuracy : 0.9852
##                  95% CI : (0.9475, 0.9982)
##     No Information Rate : 0.6519
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9674
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9886
##             Specificity : 0.9787
##          Pos Pred Value : 0.9886
##          Neg Pred Value : 0.9787
##              Prevalence : 0.6519
##          Detection Rate : 0.6444
##    Detection Prevalence : 0.6519
##       Balanced Accuracy : 0.9837
##
##        'Positive' Class : benign
##
```