```r
#**Intro to Data Science**

### Final Project
### Group: EAST

#Group Members:

#Rose Gogliotti
#Pavel Mesa Neimane
#Doug Perez


#loading necessary libraries to perform analysis

library(tidyverse)
library(ggplot2)
library(rpart)
library(partykit)
library(randomForest)
library(class)

df <- read.csv("FNA_cancer.csv", header = T) #loading data set

df <- df[,-c(1, 33)]

df <- na.omit(df) #discarding rows with NA values

glimpse(df)

#reorder variables alphabetically with diagnosis moved to front

df <- df[ , order(names(df))]
df <- df %>%
  select(diagnosis, everything())

summary(df)

#number of each element in column diagnosis:

table(df$diagnosis)

#bar graph of diagnosis results

diag_bar <- ggplot(df, aes(df$diagnosis, fill = diagnosis)) +
  geom_bar() +
  labs(x = "Benign / Malignant diagnosis", title = "Histogram Diagnosis") +
  theme_bw() + scale_fill_manual(values = c("turquoise3", "indianred2"))+
  theme(legend.position = "none") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = 2)

diag_bar #displying result

#In the following section, we grouped variables by their nature in
#order to create histograms and geom point charts with the objective
#to explore possible relationships between them.The groups are: area,
#compactness, concave points, concavity, fractal dimension, perimeter,
#radius smoothness, symmetry and texture.

####area parameters

# select only area parameters
require(dplyr)

#new data frame with only area parameters
area <-  dplyr::select(df, area_mean, area_se, area_worst)
glimpse(area)

summary(area)

#histogram for area mean

area_hist <- ggplot(df, aes(area_mean)) +
  geom_histogram() +
  labs(x = "Mean area", title = "Mean Area") +
  theme_bw() +
  theme(legend.position = "none")

area_hist #displaying histogram above

#histogram for worst area

area_hist_max <- ggplot(df, aes(area_worst)) +
  geom_histogram() +
  labs(x = "Worst area", title = "Worst Area") +
  theme_bw() +
  theme(legend.position = "none")

area_hist_max #displaying results

#histogram for variable SE area

area_hist_se <- ggplot(df, aes(area_se)) +
  geom_histogram() +
  labs(x = "SE area", title = "SE Area") +
  theme_bw() +
  theme(legend.position = "none")

area_hist_se #displaying results

#geom point chart of mean area vs worst area

area_mean_vs_worst <- ggplot(df, aes(area_mean, area_worst)) +
  geom_point() +
  labs(x = "Mean Area", y= "Worst Area", title = "Mean Area vs Worst Area") +
  theme_bw() +
  theme(legend.position = "none")
```

```
area_mean_vs_worst #displaying results

#geom point chart of mean area vs se area

area_mean_vs_se <- ggplot(df, aes(area_mean, area_se)) +
  geom_point() +
  labs(x = "Mean Area", y= "SE Area", title = "Mean Area vs SE Area") +
  theme_bw() +
  theme(legend.position = "none")

area_mean_vs_se  #displaying results

#geom point of worst area vs se area

area_worst_vs_se <- ggplot(df, aes(area_worst, area_se)) +
  geom_point() +
  labs(x = "Worst Area", y= "SE Area", title = "Worst Area vs SE Area") +
  theme_bw() +
  theme(legend.position = "none")

area_worst_vs_se #displaying results

####compactness parameters

#select only compactness parameters
require(dplyr)

#data frame with only compactness parameters
compactness <-  dplyr::select(df, compactness_mean, compactness_se, compactness_worst)
glimpse(compactness)

summary(compactness)

#histogram of mean compactness

compactness_hist <- ggplot(df, aes(compactness_mean)) +
  geom_histogram() +
  labs(x = "Mean compactness", title = "Mean Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_hist #displaying results

#histogram of worst compactness

compactness_hist_max <- ggplot(df, aes(compactness_worst)) +
  geom_histogram() +
  labs(x = "Worst compactness", title = "Worst Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_hist_max #displaying results

#histogram of se compactness

compactness_hist_se <- ggplot(df, aes(compactness_se)) +
  geom_histogram() +
  labs(x = "SE compactness", title = "SE Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_hist_se #displaying results

#geom point of mean compactness vs worst compactness

compactness_mean_vs_worst <- ggplot(df, aes(compactness_mean, compactness_worst)) +
  geom_point() +
  labs(x = "Mean Compactness", y= "Worst Compactness", title = "Mean Compactness vs Worst Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_mean_vs_worst #displaying results

#geom point of mean compactness vs se compactness

compactness_mean_vs_se <- ggplot(df, aes(compactness_mean, compactness_se)) +
  geom_point() +
  labs(x = "Mean Compactness", y= "SE Compactness", title = "Mean Compactness vs SE Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_mean_vs_se #displaying results

#geom point of worst compactness vs se compactness

compactness_worst_vs_se <- ggplot(df, aes(compactness_worst, compactness_se)) +
  geom_point() +
  labs(x = "Worst Compactness", y= "SE Compactness", title = "Worst Compactness vs SE Compactness") +
  theme_bw() +
  theme(legend.position = "none")

compactness_worst_vs_se #displaying results

####concave.points parameters

# select only concave.points parameters

require(dplyr)
#data frame with only concave points variables
concave.points <-  dplyr::select(df, concave.points_mean, concave.points_se, concave.points_worst)

glimpse(concave.points)

summary(concave.points)
```

```r
#histogram mean concave points

concave.points_hist <- ggplot(df, aes(concave.points_mean)) +
  geom_histogram() +
  labs(x = "Mean concave.points", title = "Mean Concave.points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_hist #displaying results

#histogram worst concave points

concave.points_hist_max <- ggplot(df, aes(concave.points_worst)) +
  geom_histogram() +
  labs(x = "Worst Concave.points", title = "Worst Concave.points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_hist_max #displaying results

#histogram se concave points

concave.points_hist_se <- ggplot(df, aes(concave.points_se)) +
  geom_histogram() +
  labs(x = "SE concave points", title = "SE Concave points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_hist_se #displaying results

#geom point mean vs worst concave points

concave.points_mean_vs_worst <- ggplot(df, aes(concave.points_mean, concave.points_worst)) +
  geom_point() +
  labs(x = "Mean Concave points", y= "Worst Concave points", title = "Mean Concave points vs Worst Concave points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_mean_vs_worst #displaying results

#geom point mean vs se concave points

concave.points_mean_vs_se <- ggplot(df, aes(concave.points_mean, concave.points_se)) +
  geom_point() +
  labs(x = "Mean Concave points", y= "SE Concave points", title = "Mean Concave points vs SE Concave points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_mean_vs_se #displaying results

#geom point worst vs se concave points

concave.points_worst_vs_se <- ggplot(df, aes(concave.points_worst, concave.points_se)) +
  geom_point() +
  labs(x = "Worst Concave points", y= "SE Concave points", title = "Worst Concave points vs SE Concave points") +
  theme_bw() +
  theme(legend.position = "none")

concave.points_worst_vs_se #displaying results

####concavity parameters

# select only concavity parameters
require(dplyr)

#data frame with concavity variables
concavity <-  dplyr::select(df, concavity_mean, concavity_se, concavity_worst)
glimpse(concavity)

summary(concavity)

#histogram of mean concavity

concavity_hist <- ggplot(df, aes(concavity_mean)) +
  geom_histogram() +
  labs(x = "Mean concavity", title = "Mean Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_hist #displaying results

#histogram of worst concavity

concavity_hist_max <- ggplot(df, aes(concavity_worst)) +
  geom_histogram() +
  labs(x = "Worst concavity", title = "Worst Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_hist_max #displaying results

#histogram of se concavity

concavity_hist_se <- ggplot(df, aes(concavity_se)) +
  geom_histogram() +
  labs(x = "SE concavity", title = "SE Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_hist_se #displaying results

#geom point of mean concavity vs worst concavity

concavity_mean_vs_worst <- ggplot(df, aes(concavity_mean, concavity_worst)) +
  geom_point() +
  labs(x = "Mean Concavity", y= "Worst Concavity", title = "Mean Concavity vs Worst Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_mean_vs_worst #displaying results

#geom point of mean concavity vs se concavity

concavity_mean_vs_se <- ggplot(df, aes(concavity_mean, concavity_se)) +
  geom_point() +
  labs(x = "Mean Concavity", y= "SE Concavity", title = "Mean Concavity vs SE Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_mean_vs_se #displaying results

#geom point of worst concavity vs se concavity

concavity_worst_vs_se <- ggplot(df, aes(concavity_worst, concavity_se)) +
  geom_point() +
  labs(x = "Worst Concavity", y= "SE Concavity", title = "Worst Concavity vs SE Concavity") +
  theme_bw() +
  theme(legend.position = "none")

concavity_worst_vs_se #displaying results

####fractal_dimension parameters

# select only fractal_dimension parameters
require(dplyr)

#data set for fractal dimension variables
fractal_dimension <-  dplyr::select(df, fractal_dimension_mean, fractal_dimension_se, fractal_dimension_worst)
glimpse(fractal_dimension)

summary(fractal_dimension)

#histogram for mean fractal dimension

fractal_dimension_hist <- ggplot(df, aes(fractal_dimension_mean)) +
  geom_histogram() +
  labs(x = "Mean fractal_dimension", title = "Mean Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_hist #displaying results

#histogram of worst fractal dimension

fractal_dimension_hist_max <- ggplot(df, aes(fractal_dimension_worst)) +
  geom_histogram() +
  labs(x = "Worst fractal_dimension", title = "Worst Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_hist_max #displaying results

#histogram of se fractal dimension

fractal_dimension_hist_se <- ggplot(df, aes(fractal_dimension_se)) +
  geom_histogram() +
  labs(x = "SE fractal_dimension", title = "SE Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_hist_se #displaying results

#geom point of mean vs worst fractal dimension

fractal_dimension_mean_vs_worst <- ggplot(df, aes(fractal_dimension_mean, fractal_dimension_worst)) +
  geom_point() +
  labs(x = "Mean Fractal_dimension", y= "Worst Fractal_dimension", title = "Mean Fractal_dimension vs Worst Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_mean_vs_worst #displaying results

#geom point of mean vs se fractal dimension

fractal_dimension_mean_vs_se <- ggplot(df, aes(fractal_dimension_mean, fractal_dimension_se)) +
  geom_point() +
  labs(x = "Mean Fractal_dimension", y= "SE Fractal_dimension", title = "Mean Fractal_dimension vs SE Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_mean_vs_se #displaying results

#geom point of worst vs se fractal dimension

fractal_dimension_worst_vs_se <- ggplot(df, aes(fractal_dimension_worst, fractal_dimension_se)) +
  geom_point() +
  labs(x = "Worst Fractal_dimension", y= "SE Fractal_dimension", title = "Worst Fractal_dimension vs SE Fractal_dimension") +
  theme_bw() +
  theme(legend.position = "none")

fractal_dimension_worst_vs_se #displying results

####perimeter parameters

# select only perimeter parameters
require(dplyr)

#data frame for perimeter variables
perimeter <-  dplyr::select(df, perimeter_mean, perimeter_se, perimeter_worst)
glimpse(perimeter)

summary(perimeter)
```

```r
#histogram for mean perimeter

perimeter_hist <- ggplot(df, aes(perimeter_mean)) +
  geom_histogram() +
  labs(x = "Mean perimeter", title = "Mean Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_hist #displaying results

#histogram worst perimeter

perimeter_hist_max <- ggplot(df, aes(perimeter_worst)) +
  geom_histogram() +
  labs(x = "Worst perimeter", title = "Worst Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_hist_max #displaying results

#histogram se perimeter

perimeter_hist_se <- ggplot(df, aes(perimeter_se)) +
  geom_histogram() +
  labs(x = "SE perimeter", title = "SE Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_hist_se #displaying results

#geom point mean vs worst perimeter

perimeter_mean_vs_worst <- ggplot(df, aes(perimeter_mean, perimeter_worst)) +
  geom_point() +
  labs(x = "Mean Perimeter", y= "Worst Perimeter", title = "Mean Perimeter vs Worst Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_mean_vs_worst #displaying results

#geom point mean vs se perimeter

perimeter_mean_vs_se <- ggplot(df, aes(perimeter_mean, perimeter_se)) +
  geom_point() +
  labs(x = "Mean Perimeter", y= "SE Perimeter", title = "Mean Perimeter vs SE Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_mean_vs_se #displaying results

#geom point worst vs se perimeter

perimeter_worst_vs_se <- ggplot(df, aes(perimeter_worst, perimeter_se)) +
  geom_point() +
  labs(x = "Worst Perimeter", y= "SE Perimeter", title = "Worst Perimeter vs SE Perimeter") +
  theme_bw() +
  theme(legend.position = "none")

perimeter_worst_vs_se #displaying results

####radius parameters

# select only radius parameters
require(dplyr)

#data frame for radius variables
radius <-  dplyr::select(df, radius_mean, radius_se, radius_worst)
glimpse(radius)

summary(radius)

#histogram for mean radius

radius_hist <- ggplot(df, aes(radius_mean)) +
  geom_histogram() +
  labs(x = "Mean radius", title = "Mean Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_hist #displaying results

#histogram for worst radius

radius_hist_max <- ggplot(df, aes(radius_worst)) +
  geom_histogram() +
  labs(x = "Worst radius", title = "Worst Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_hist_max #displaying results

#histogram for se radius

radius_hist_se <- ggplot(df, aes(radius_se)) +
  geom_histogram() +
  labs(x = "SE radius", title = "SE Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_hist_se #displaying results

#geom point for mean vs worst radius

radius_mean_vs_worst <- ggplot(df, aes(radius_mean, radius_worst)) +
  geom_point() +
  labs(x = "Mean Radius", y= "Worst Radius", title = "Mean Radius vs Worst Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_mean_vs_worst #displaying results

#geom point of mean vs se radius

radius_mean_vs_se <- ggplot(df, aes(radius_mean, radius_se)) +
  geom_point() +
  labs(x = "Mean Radius", y= "SE Radius", title = "Mean Radius vs SE Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_mean_vs_se #displaying results

#geom point of worst vs se radius

radius_worst_vs_se <- ggplot(df, aes(radius_worst, radius_se)) +
  geom_point() +
  labs(x = "Worst Radius", y= "SE Radius", title = "Worst Radius vs SE Radius") +
  theme_bw() +
  theme(legend.position = "none")

radius_worst_vs_se #displaying results

####smoothness parameters

#select only smoothness parameters
require(dplyr)

#data frame of smoothness variables
smoothness <-  dplyr::select(df, smoothness_mean, smoothness_se, smoothness_worst)
glimpse(smoothness)

summary(smoothness)

#histogram of mean smoothness

smoothness_hist <- ggplot(df, aes(smoothness_mean)) +
  geom_histogram() +
  labs(x = "Mean smoothness", title = "Mean Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_hist #displaying results

#histogram of worst smoothness

smoothness_hist_max <- ggplot(df, aes(smoothness_worst)) +
  geom_histogram() +
  labs(x = "Worst smoothness", title = "Worst Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_hist_max #displaying results

#histogram of se smoothness

smoothness_hist_se <- ggplot(df, aes(smoothness_se)) +
  geom_histogram() +
  labs(x = "SE smoothness", title = "SE Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_hist_se #displaying results

#geom point of mean vs worst smoothness

smoothness_mean_vs_worst <- ggplot(df, aes(smoothness_mean, smoothness_worst)) +
  geom_point() +
  labs(x = "Mean Smoothness", y= "Worst Smoothness", title = "Mean Smoothness vs Worst Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_mean_vs_worst #displaying results

#geom point of mean vs se smoothness

smoothness_mean_vs_se <- ggplot(df, aes(smoothness_mean, smoothness_se)) +
  geom_point() +
  labs(x = "Mean Smoothness", y= "SE Smoothness", title = "Mean Smoothness vs SE Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_mean_vs_se #displaying results

#geom point of worst vs se smoothness

smoothness_worst_vs_se <- ggplot(df, aes(smoothness_worst, smoothness_se)) +
  geom_point() +
  labs(x = "Worst Smoothness", y= "SE Smoothness", title = "Worst Smoothness vs SE Smoothness") +
  theme_bw() +
  theme(legend.position = "none")

smoothness_worst_vs_se #displaying results

####symmetry parameters

# select only symmetry parameters
require(dplyr)

#data frame for symmetry variables
symmetry <-  dplyr::select(df, symmetry_mean, symmetry_se, symmetry_worst)
glimpse(symmetry)
```

```r
summary(symmetry)

#histogram of mean symmetry

symmetry_hist <- ggplot(df, aes(symmetry_mean)) +
  geom_histogram() +
  labs(x = "Mean symmetry", title = "Mean Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_hist #displaying results

#histogram of worst symmetry

symmetry_hist_max <- ggplot(df, aes(symmetry_worst)) +
  geom_histogram() +
  labs(x = "Worst symmetry", title = "Worst Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_hist_max #displaying results

#histogram of se symmetry

symmetry_hist_se <- ggplot(df, aes(symmetry_se)) +
  geom_histogram() +
  labs(x = "SE symmetry", title = "SE Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_hist_se #displaying results

#geom point of mean vs worst symmetry

symmetry_mean_vs_worst <- ggplot(df, aes(symmetry_mean, symmetry_worst)) +
  geom_point() +
  labs(x = "Mean Symmetry", y= "Worst Symmetry", title = "Mean Symmetry vs Worst Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_mean_vs_worst #displaying results

#geom point of mean vs se symmetry

symmetry_mean_vs_se <- ggplot(df, aes(symmetry_mean, symmetry_se)) +
  geom_point() +
  labs(x = "Mean Symmetry", y= "SE Symmetry", title = "Mean Symmetry vs SE Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_mean_vs_se #displaying results

#geom point of worst vs se symmetry

symmetry_worst_vs_se <- ggplot(df, aes(symmetry_worst, symmetry_se)) +
  geom_point() +
  labs(x = "Worst Symmetry", y= "SE Symmetry", title = "Worst Symmetry vs SE Symmetry") +
  theme_bw() +
  theme(legend.position = "none")

symmetry_worst_vs_se #displaying results

####texture parameters

# select only texture parameters
require(dplyr)

#data frame for texture variables
texture <-  dplyr::select(df, texture_mean, texture_se, texture_worst)
glimpse(texture)

summary(texture)

#histogram of mean texture

texture_hist <- ggplot(df, aes(texture_mean)) +
  geom_histogram() +
  labs(x = "Mean texture", title = "Mean Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_hist #displaying results

#histogram of worst texture

texture_hist_max <- ggplot(df, aes(texture_worst)) +
  geom_histogram() +
  labs(x = "Worst texture", title = "Worst Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_hist_max #displaying results

#histogram of se texture

texture_hist_se <- ggplot(df, aes(texture_se)) +
  geom_histogram() +
  labs(x = "SE texture", title = "SE Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_hist_se #displaying results

#geom point of mean vs worst texture

texture_mean_vs_worst <- ggplot(df, aes(texture_mean, texture_worst)) +
  geom_point() +
  labs(x = "Mean Texture", y= "Worst Texture", title = "Mean Texture vs Worst Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_mean_vs_worst #displaying results

#geom point of mean vs se texture

texture_mean_vs_se <- ggplot(df, aes(texture_mean, texture_se)) +
  geom_point() +
  labs(x = "Mean Texture", y= "SE Texture", title = "Mean Texture vs SE Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_mean_vs_se #displaying results

#geom point of worst vs se texture

texture_worst_vs_se <- ggplot(df, aes(texture_worst, texture_se)) +
  geom_point() +
  labs(x = "Worst Texture", y= "SE Texture", title = "Worst Texture vs SE Texture") +
  theme_bw() +
  theme(legend.position = "none")

texture_worst_vs_se #displaying results


####In this section we created the training and test data sets

set.seed(1874)

n <- nrow(df)
n
flux_df <- sample.int(n, size = round(0.2 * n))

train_df <- df[-flux_df, ] #this is the training data set
test_df <- df[flux_df, ] #this is the test data set

nrow(train_df)
nrow(test_df)

#Creating general formula for large model

attach(df)

#formula with giagnosis as response variable and all variables as predictors.
form <- as.formula(diagnosis~.)

#Creating decision tree

set.seed(1874)
#decision tree using train data set
tree_1 <- rpart(form, data = train_df)
tree_1

plot(as.party(tree_1)) #plotting decision tree with party kit.

printcp(tree_1)

plotcp(tree_1)

#when we plotted the tree we realized that critical x-val
#relative error is aproximately 0.021

#We proceeded with prunning process based in critical x-val
#relative error of 0.021

set.seed(1874)
tree_2 <- prune(tree_1, cp = 0.021) #prunning algorythm.


plot(as.party(tree_2)) #plotting with part kit.

printcp(tree_2)
plotcp(tree_2) #displaying new x-val rel error vs cp chart.

####In this section we proceeded with estimating bagged tree

set.seed(1874)
#parameters used were mtry = 30 and ntree =1000

bagged_tree <- randomForest(form, data = train_df,
                            mtry = 30, ntree = 1000,
                            na.action = na.roughfix)

bagged_tree

#Computing importance
#computing importance for bagged tree

imp_bagged_tree <- importance(bagged_tree)
imp_bagged_tree

#Predicting using bagged tree

test_df$diagnosis_pred <- predict(bagged_tree, test_df, type = "class")

#confusion matrix for predicted values

table(test_df$diagnosis, test_df$diagnosis_pred)

#In this section we constructed different random forests.
#We used mtry values of 1, 5, 6 and 10

##Random forest with mtry = 1
```

```
set.seed(1874)
rf_m_1 <- randomForest(form, data = train_df,
                            mtry = 1, ntree = 500,
                            na.action = na.roughfix)
rf_m_1

##Random forest with mtry = 5
set.seed(1874)
rf_m_5 <- randomForest(form, data = train_df,
                       mtry = 5, ntree = 500,
                       na.action = na.roughfix)
rf_m_5

##Random forest with mtry = 6
set.seed(1874)
rf_m_6 <- randomForest(form, data = train_df,
                       mtry = 6, ntree = 500,
                       na.action = na.roughfix)
rf_m_6

##Random forest with mtry = 10
set.seed(1874)
rf_m_10 <- randomForest(form, data = train_df,
                       mtry = 10, ntree = 500,
                       na.action = na.roughfix)
rf_m_10

#Random forests with mtry=6 and mtry=10 provided equal results with
#same OOB (3.3%) and same confusion matrix.However, we prefer
#simplicity therefore, we are selecting random forest with mtry=6

test_df$diagnosis_pred_forest <- predict(rf_m_6, test_df, type = "class")

#confusion matrix for predicted values
table(test_df$diagnosis, test_df$diagnosis_pred_forest)

#Pre-work for KNN algorithm
#Convert categorical variable into indicators

# duplicate data set in order not to lose important information

df1 <- df

#Convert Diagnosis to factor
df$diagnosis <- factor(df$diagnosis, levels = c("B", "M"), labels = c("benign", "malignant"))
df1$diagnosis <- factor(df1$diagnosis, levels = c("B", "M"), labels = c("benign", "malignant"))

#When we pruned our tree, the most important variables we found were:
#perimeter_worst, concave.points_mean and area_worst we are going to
#work with these variables and the first step is to re scale them.

#rescale function
rescale_x <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

df1[2:31]<- as.data.frame(lapply(df1[2:31], rescale_x))

#Creating new training and test data sets

set.seed(1874)
n1 <- nrow(df1)
n1
flux_df1 <- sample.int(n1, size = round(0.2 * n1))

train_df1 <- df1[-flux_df1, ]
test_df1 <- df1[flux_df1, ]

nrow(train_df1)
nrow(test_df1)

table(train_df1$diagnosis)
table(test_df1$diagnosis)

#chose an odd number near the square root of size of the training set.

sqrt(nrow(train_df1))

#Running the classifier
set.seed(1874)
diagnosis_knn_21 <- knn(train_df1[-1],
                       test = test_df1[-1],
                       cl=train_df1$diagnosis, k=21)
diagnosis_knn_21

table(test_df1$diagnosis, diagnosis_knn_21)
prop.table(table(test_df1$diagnosis, diagnosis_knn_21))

#Testing other k values
set.seed(1874)
diagnosis_knn_1 <- knn(train_df1[-1],
                        test = test_df1[-1],
                        cl=train_df1$diagnosis, k=1)

set.seed(1874)
diagnosis_knn_3 <- knn(train_df1[-1],
                       test = test_df1[-1],
                       cl=train_df1$diagnosis, k=3)
set.seed(1874)
diagnosis_knn_7 <- knn(train_df1[-1],
                       test = test_df1[-1],
                       cl=train_df1$diagnosis, k=7)
set.seed(1874)
diagnosis_knn_15 <- knn(train_df1[-1],
                       test = test_df1[-1],
                       cl=train_df1$diagnosis, k=15)

set.seed(1874)
diagnosis_knn_31 <- knn(train_df1[-1],
                       test = test_df1[-1],
                       cl=train_df1$diagnosis, k=31)


table(test_df1$diagnosis, diagnosis_knn_1)
table(test_df1$diagnosis, diagnosis_knn_3)
table(test_df1$diagnosis, diagnosis_knn_7)
table(test_df1$diagnosis, diagnosis_knn_15)
table(test_df1$diagnosis, diagnosis_knn_31)

#k=3, k=7 and k=15 appear to have the same lowest misclassification
#rate of 5.
```