```
#Modeling Part for IST 687 Final Project - sabdelra
```

```
#Library to call at the begining of the code
library (tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(RCurl)
```

```
##
## Attaching package: 'RCurl'
##
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#Library(ggplot)
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
##
## The following object is masked from 'package:purrr':
##
##     cross
##
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.2
```

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

```
library(rio)
```

```
## Warning: package 'rio' was built under R version 4.2.2
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.2
```

```
data <- data.frame(read_csv('HMO_data.csv'))
```

```
## Rows: 7582 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (8): smoker, location, location_type, education_level, yearly_physical, ...
## dbl (6): X, age, bmi, children, hypertension, cost
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data <- transform(
  data, expensive= ifelse(cost > 5000, TRUE, FALSE))
```

```
data <- data %>% mutate(across(bmi, ~replace_na(., mean(., na.rm=TRUE))))
```

```
HMO_data <- data[, c('bmi', 'age','smoker', 'exercise',  'cost')]
```

```
output <- lm(cost~., data= HMO_data)
```

```
summary(output)
```

```
##
## Call:
## lm(formula = cost ~ ., data = HMO_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12321  -1514   -376    989  41978
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -8778.260    224.018  -39.19   <2e-16 ***
## bmi                  181.523      6.261   28.99   <2e-16 ***
## age                  103.574      2.634   39.33   <2e-16 ***
## smokeryes           7690.012     93.830   81.96   <2e-16 ***
## exerciseNot-Active  2268.430     85.981   26.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3237 on 7577 degrees of freedom
## Multiple R-squared:  0.569,  Adjusted R-squared:  0.5687
## F-statistic:  2500 on 4 and 7577 DF,  p-value: < 2.2e-16
```

```
HMO_data <- data[, c('bmi', 'age','smoker', 'exercise', 'expensive')]
```

```
HMO_data$expensive <- as.factor(HMO_data$expensive)
```

```
set.seed(111)
trainList <- createDataPartition(y=HMO_data$expensive, p=.70,list=FALSE)
trainSet <- HMO_data[trainList,]
testSet <- HMO_data[-trainList,]
```

```
model <- ksvm(data= trainSet, expensive~., C=5, CV =3, prob.model =TRUE)
```

```
model
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.515084267225532
##
## Number of Support Vectors : 1453
##
## Objective Function Value : -6239.629
## Training error : 0.112472
## Probability model included.
```

```
svmPred <- predict(model,newdata = testSet)
```

```
confMatrix <- table(svmPred, testSet$expensive)
```

```
confMatrix[1, "FALSE"]
```

```
## [1] 1696
```

```
confMatrix
```

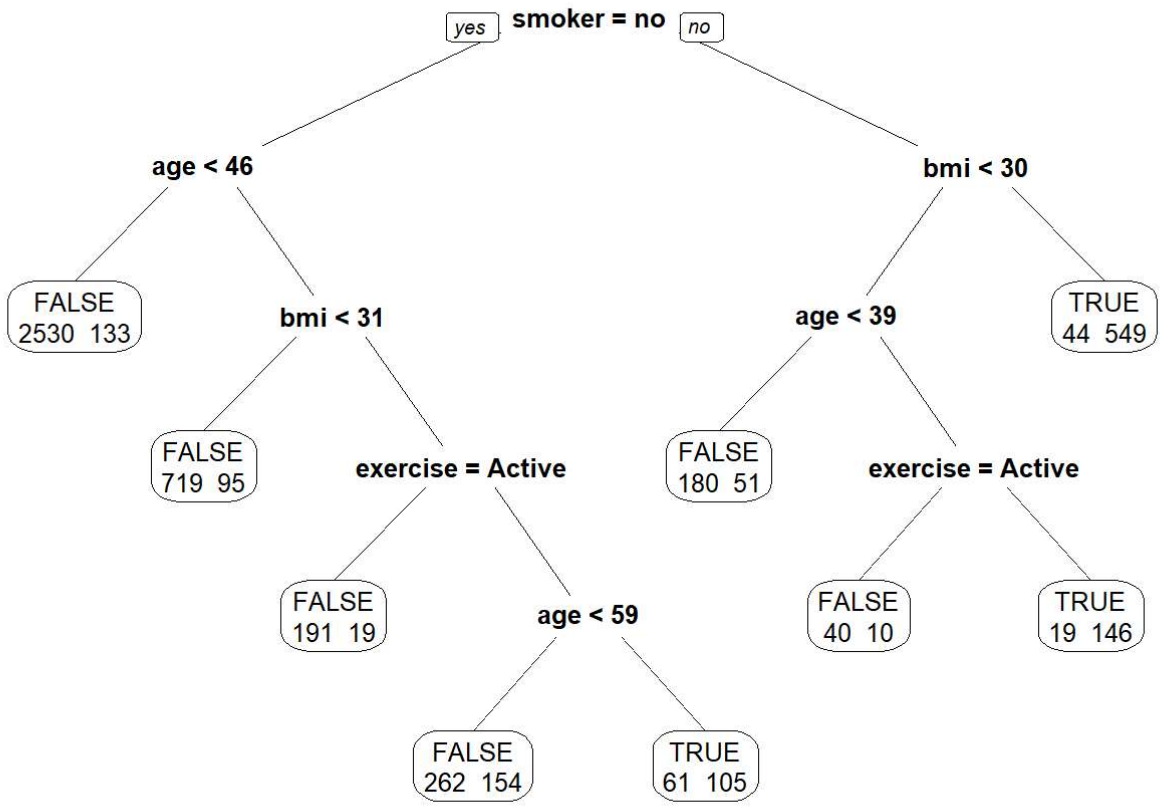
```
##
## svmPred FALSE TRUE
##   FALSE  1696  219
##   TRUE     38  321
```

```
confusionMatrix(svmPred,testSet$expensive )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1696  219
##      TRUE     38  321
##
##                Accuracy : 0.887
##                  95% CI : (0.8732, 0.8997)
##     No Information Rate : 0.7625
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6472
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9781
##             Specificity : 0.5944
##          Pos Pred Value : 0.8856
##          Neg Pred Value : 0.8942
##              Prevalence : 0.7625
##          Detection Rate : 0.7458
##    Detection Prevalence : 0.8421
##       Balanced Accuracy : 0.7863
##
##        'Positive' Class : FALSE
##
```

```
cartTree <- rpart(expensive~., data = trainSet, control = c(maxdepth = 5, cp=0.002))
```

```
prp(cartTree, faclen = 0, cex = 0.8, extra = 1)
```




```
predictValues <- predict(cartTree, newdata=testSet, type = "class")
```

```
confusionMatrix(predictValues,testSet$expensive )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  1686  209
##      TRUE     48  331
##
##                Accuracy : 0.887
##                  95% CI : (0.8732, 0.8997)
##     No Information Rate : 0.7625
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6522
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9723
##             Specificity : 0.6130
##          Pos Pred Value : 0.8897
##          Neg Pred Value : 0.8734
##              Prevalence : 0.7625
##          Detection Rate : 0.7414
##    Detection Prevalence : 0.8333
##       Balanced Accuracy : 0.7926
##
##        'Positive' Class : FALSE
##
```