Anly 530 Project group 2

Group 2

12/3/2020

library(readxl)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(naivebayes)

## Warning: package 'naivebayes' was built under R version 4.0.3

## naivebayes 0.9.7 loaded

library(gmodels)

## Warning: package 'gmodels' was built under R version 4.0.3

library(corrplot)

## corrplot 0.84 loaded

library(kernlab)

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(NbClust)

## Warning: package 'NbClust' was built under R version 4.0.3

library(cluster)  
library(rpart)  
library(rpart.plot)  
library(RColorBrewer)  
library(rattle)

## Warning: package 'rattle' was built under R version 4.0.3

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

trainData <- read.csv("C:/Users/nagar/Documents/ANLY 530 Machine Learning/Project 1 - Abstenteeism/Absenteeism\_at\_work\_train.csv")  
trainData <- na.omit(trainData)  
trainData <- trainData[-c(1,10,11,18,19)]  
#trainData$Education <- as.factor(trainData$Education)  
trainData$Age <- as.integer(trainData$Age)

## Warning: NAs introduced by coercion

#trainData$Reason.for.absence = as.factor(trainData$Reason.for.absence)  
#trainData$Month.of.absence=as.factor(trainData$Month.of.absence)  
#trainData$Day.of.the.week = as.factor(trainData$Day.of.the.week)  
#trainData$Seasons = as.factor(trainData$Seasons)  
#trainData$Disciplinary.failure = as.factor(trainData$Disciplinary.failure)  
#trainData$Social.drinker = as.factor(trainData$Social.drinker)  
#trainData$Social.smoker = as.factor(trainData$Social.smoker)  
  
 for(i in 1:663) {  
 if(trainData$Absenteeism.time.in.hours[i] == 0)   
   
 { trainData$Absenteeism.time.in.hours[i] = "Group 0"}  
 else if (trainData$Absenteeism.time.in.hours[i] >= 1 & trainData$Absenteeism.time.in.hours[i] <= 6)  
 {trainData$Absenteeism.time.in.hours[i] = "Group 1"}  
 else if (trainData$Absenteeism.time.in.hours[i] > 6)  
 {trainData$Absenteeism.time.in.hours[i] = "Group 2"}  
 }  
  
  
trainData$Absenteeism.time.in.hours <- as.factor(trainData$Absenteeism.time.in.hours)

testData <- read.csv("C:/Users/nagar/Documents/ANLY 530 Machine Learning/Project 1 - Abstenteeism/Absenteeism\_at\_work\_test.csv")  
  
testData <- na.omit(testData)  
testData <- testData[-c(1,10,11,18,19)]  
#testData$Education <- as.factor(testData$Education)  
testData$Age <- as.integer(testData$Age)  
#testData$Reason.for.absence = as.factor(testData$Reason.for.absence)  
# testData$Month.of.absence=as.factor(testData$Month.of.absence)  
# testData$Day.of.the.week = as.factor(testData$Day.of.the.week)  
# testData$Seasons = as.factor(testData$Seasons)  
# testData$Disciplinary.failure = as.factor(testData$Disciplinary.failure)  
# testData$Social.drinker = as.factor(testData$Social.drinker)  
# testData$Social.smoker = as.factor(testData$Social.smoker)  
#  
 for(i in 1:74) {  
 if(testData$Absenteeism.time.in.hours[i] == 0)   
   
 { testData$Absenteeism.time.in.hours[i] = "Group 0"}  
 else if (testData$Absenteeism.time.in.hours[i] >= 1 & testData$Absenteeism.time.in.hours[i] <= 6)  
 {testData$Absenteeism.time.in.hours[i] = "Group 1"}  
 else if (testData$Absenteeism.time.in.hours[i] > 6)  
 {testData$Absenteeism.time.in.hours[i] = "Group 2"}  
 }  
  
  
testData$Absenteeism.time.in.hours <- as.factor(testData$Absenteeism.time.in.hours)

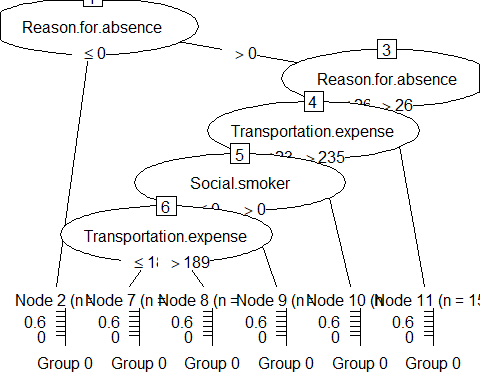
library(C50)  
  
 decision\_tree\_model <- C5.0(x = testData[-c(16)], y = testData$Absenteeism.time.in.hours)  
   
   
 decision\_tree\_model

##   
## Call:  
## C5.0.default(x = testData[-c(16)], y = testData$Absenteeism.time.in.hours)  
##   
## Classification Tree  
## Number of samples: 74   
## Number of predictors: 15   
##   
## Tree size: 6   
##   
## Non-standard options: attempt to group attributes

summary(decision\_tree\_model)

##   
## Call:  
## C5.0.default(x = testData[-c(16)], y = testData$Absenteeism.time.in.hours)  
##   
##   
## C5.0 [Release 2.07 GPL Edition] Sat Dec 05 13:37:51 2020  
## -------------------------------  
##   
## Class specified by attribute `outcome'  
##   
## Read 74 cases (16 attributes) from undefined.data  
##   
## Decision tree:  
##   
## Reason.for.absence <= 0: Group 0 (7)  
## Reason.for.absence > 0:  
## :...Reason.for.absence > 26: Group 1 (15)  
## Reason.for.absence <= 26:  
## :...Transportation.expense > 235: Group 2 (8)  
## Transportation.expense <= 235:  
## :...Social.smoker > 0: Group 2 (4/1)  
## Social.smoker <= 0:  
## :...Transportation.expense <= 189: Group 1 (19/3)  
## Transportation.expense > 189: Group 2 (21/10)  
##   
##   
## Evaluation on training data (74 cases):  
##   
## Decision Tree   
## ----------------   
## Size Errors   
##   
## 6 14(18.9%) <<  
##   
##   
## (a) (b) (c) <-classified as  
## ---- ---- ----  
## 7 (a): class Group 0  
## 31 11 (b): class Group 1  
## 3 22 (c): class Group 2  
##   
##   
## Attribute usage:  
##   
## 100.00% Reason.for.absence  
## 70.27% Transportation.expense  
## 59.46% Social.smoker  
##   
##   
## Time: 0.0 secs

plot(decision\_tree\_model)



library(gmodels)  
 library(descr)

## Warning: package 'descr' was built under R version 4.0.3

##   
## Attaching package: 'descr'

## The following object is masked from 'package:gmodels':  
##   
## CrossTable

#Evaluation:  
   
 decision\_tree\_pred <- predict(decision\_tree\_model, testData)  
 CrossTable(testData$Absenteeism.time.in.hours, decision\_tree\_pred, prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE, dnn = c('Actual Group', 'Predicted Group'))

## Cell Contents   
## |-------------------------|  
## | N |   
## | N / Table Total |   
## |-------------------------|  
##   
## ===================================================  
## Predicted Group  
## Actual Group Group 0 Group 1 Group 2 Total  
## ---------------------------------------------------  
## Group 0 7 0 0 7  
## 0.095 0.000 0.000   
## ---------------------------------------------------  
## Group 1 0 31 11 42  
## 0.000 0.419 0.149   
## ---------------------------------------------------  
## Group 2 0 3 22 25  
## 0.000 0.041 0.297   
## ---------------------------------------------------  
## Total 7 34 33 74  
## ===================================================

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':  
##   
## importance

## The following object is masked from 'package:ggplot2':  
##   
## margin

random\_forest\_model <- randomForest(trainData$Absenteeism.time.in.hours~., data = trainData, na.action=na.exclude)  
  
random\_forest\_model

##   
## Call:  
## randomForest(formula = trainData$Absenteeism.time.in.hours ~ ., data = trainData, na.action = na.exclude)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 22.21%  
## Confusion matrix:  
## Group 0 Group 1 Group 2 class.error  
## Group 0 36 1 0 0.02702703  
## Group 1 0 366 71 0.16247140  
## Group 2 0 75 113 0.39893617

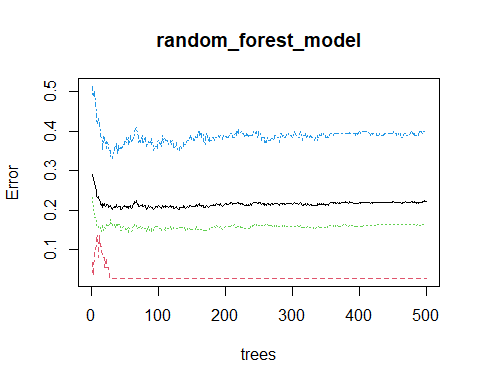
#Evaluation:  
   
 random\_forest\_pred <- predict(random\_forest\_model, testData[-16] )  
  
 random\_forest\_pred

## 1 2 3 4 5 6 7 8 9 10   
## Group 1 Group 2 Group 2 Group 1 Group 1 Group 1 Group 1 Group 1 Group 2 Group 2   
## 11 12 13 14 15 16 17 18 19 20   
## Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 2 Group 2 Group 1 Group 1   
## 21 22 23 24 25 26 27 28 29 30   
## Group 1 Group 1 Group 0 Group 1 Group 0 Group 1 Group 1 Group 2 Group 2 Group 1   
## 31 32 33 34 35 36 37 38 39 40   
## Group 1 Group 2 Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 2   
## 41 42 43 44 45 46 47 48 49 50   
## Group 2 Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 0 Group 0   
## 51 52 53 54 55 56 57 58 59 60   
## Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 2 Group 2 Group 1 Group 2   
## 61 62 63 64 65 66 67 68 69 70   
## Group 1 Group 2 Group 1 Group 2 Group 2 Group 1 Group 2 Group 1 Group 2 Group 2   
## 71 72 73 74   
## Group 2 Group 2 Group 2 Group 1   
## Levels: Group 0 Group 1 Group 2

CrossTable(testData$Absenteeism.time.in.hours, random\_forest\_pred, prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE, dnn = c('Actual Group', 'Predicted Group'))

## Cell Contents   
## |-------------------------|  
## | N |   
## | N / Table Total |   
## |-------------------------|  
##   
## ===================================================  
## Predicted Group  
## Actual Group Group 0 Group 1 Group 2 Total  
## ---------------------------------------------------  
## Group 0 4 1 2 7  
## 0.054 0.014 0.027   
## ---------------------------------------------------  
## Group 1 0 33 9 42  
## 0.000 0.446 0.122   
## ---------------------------------------------------  
## Group 2 0 10 15 25  
## 0.000 0.135 0.203   
## ---------------------------------------------------  
## Total 4 44 26 74  
## ===================================================

plot(random\_forest\_model)



set.seed(123)  
  
library(e1071)  
library(kernlab)  
svm\_vanilla = ksvm(trainData$Absenteeism.time.in.hours ~ ., data = trainData, kernel = "vanilladot")

## Setting default kernel parameters

## Setting default kernel parameters  
svm\_vanilla

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 1   
##   
## Linear (vanilla) kernel function.   
##   
## Number of Support Vectors : 367   
##   
## Objective Function Value : -2.103 -1.9241 -341.2453   
## Training error : 0.228097

svm\_vanilla\_pred = predict(svm\_vanilla, testData)  
svm\_vanilla\_pred

## [1] Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1  
## [10] Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 2 Group 1  
## [19] Group 1 Group 1 Group 1 Group 1 Group 0 Group 1 Group 0 Group 1 Group 1  
## [28] Group 2 Group 1 Group 1 Group 2 Group 2 Group 1 Group 1 Group 1 Group 2  
## [37] Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1  
## [46] Group 1 Group 1 Group 1 Group 0 Group 0 Group 1 Group 1 Group 1 Group 1  
## [55] Group 1 Group 1 Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1  
## [64] Group 1 Group 1 Group 1 Group 2 Group 1 Group 2 Group 2 Group 2 Group 2  
## [73] Group 2 Group 1  
## Levels: Group 0 Group 1 Group 2

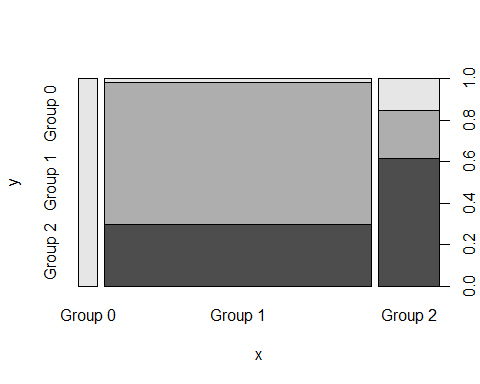
table(svm\_vanilla\_pred, testData$Absenteeism.time.in.hours)

##   
## svm\_vanilla\_pred Group 0 Group 1 Group 2  
## Group 0 4 0 0  
## Group 1 1 39 17  
## Group 2 2 3 8

##   
## absent\_predictions\_vanilla 0 1 2  
## 0 7 0 0  
## 1 1 77 6  
## 2 0 21 51  
agreement\_svm\_vanilla = (svm\_vanilla\_pred == testData$Absenteeism.time.in.hours)  
table(agreement\_svm\_vanilla)

## agreement\_svm\_vanilla  
## FALSE TRUE   
## 23 51

plot(svm\_vanilla\_pred, testData$Absenteeism.time.in.hours)



svm\_rbf = ksvm(trainData$Absenteeism.time.in.hours ~ ., data = trainData, kernel = "rbfdot")  
## Setting default kernel parameters  
svm\_rbf

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 1   
##   
## Gaussian Radial Basis kernel function.   
## Hyperparameter : sigma = 0.0481274745511382   
##   
## Number of Support Vectors : 412   
##   
## Objective Function Value : -9.1498 -9.2504 -309.8067   
## Training error : 0.203927

svm\_rbf\_pred = predict(svm\_rbf, testData)  
svm\_rbf\_pred

## [1] Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1  
## [10] Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 1 Group 1  
## [19] Group 1 Group 1 Group 1 Group 1 Group 0 Group 1 Group 0 Group 1 Group 1  
## [28] Group 1 Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1 Group 2  
## [37] Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 1 Group 2  
## [46] Group 1 Group 1 Group 1 Group 0 Group 0 Group 1 Group 1 Group 1 Group 1  
## [55] Group 1 Group 1 Group 1 Group 1 Group 1 Group 2 Group 1 Group 1 Group 1  
## [64] Group 1 Group 1 Group 1 Group 2 Group 1 Group 2 Group 2 Group 1 Group 2  
## [73] Group 2 Group 1  
## Levels: Group 0 Group 1 Group 2

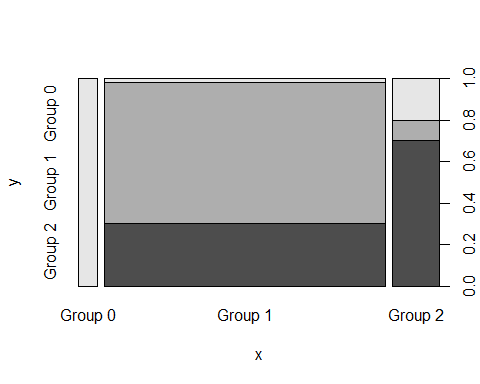
table(svm\_rbf\_pred, testData$Absenteeism.time.in.hours)

##   
## svm\_rbf\_pred Group 0 Group 1 Group 2  
## Group 0 4 0 0  
## Group 1 1 41 18  
## Group 2 2 1 7

##   
## absent\_predictions\_vanilla 0 1 2  
## 0 7 0 0  
## 1 1 77 6  
## 2 0 21 51  
agreement\_svm\_rbf= (svm\_rbf\_pred == testData$Absenteeism.time.in.hours)  
table(agreement\_svm\_rbf)

## agreement\_svm\_rbf  
## FALSE TRUE   
## 22 52

plot(svm\_rbf\_pred, testData$Absenteeism.time.in.hours)



library(naivebayes)  
 naive\_model <- naive\_bayes(trainData$Absenteeism.time.in.hours ~ .,   
 data = trainData)  
 naive\_model

##   
## ================================== Naive Bayes ==================================   
##   
## Call:   
## naive\_bayes.formula(formula = trainData$Absenteeism.time.in.hours ~   
## ., data = trainData)  
##   
## ---------------------------------------------------------------------------------   
##   
## Laplace smoothing: 0  
##   
## ---------------------------------------------------------------------------------   
##   
## A priori probabilities:   
##   
## Group 0 Group 1 Group 2   
## 0.05580694 0.66063348 0.28355958   
##   
## ---------------------------------------------------------------------------------   
##   
## Tables:   
##   
## ---------------------------------------------------------------------------------   
## ::: Reason.for.absence (Gaussian)   
## ---------------------------------------------------------------------------------   
##   
## Reason.for.absence Group 0 Group 1 Group 2  
## mean 0.7297297 22.2465753 16.7553191  
## sd 4.4387727 6.2655518 7.5099736  
##   
## ---------------------------------------------------------------------------------   
## ::: Month.of.absence (Gaussian)   
## ---------------------------------------------------------------------------------   
##   
## Month.of.absence Group 0 Group 1 Group 2  
## mean 7.945946 6.219178 6.659574  
## sd 2.914961 3.668456 3.383044  
##   
## ---------------------------------------------------------------------------------   
## ::: Day.of.the.week (Gaussian)   
## ---------------------------------------------------------------------------------   
##   
## Day.of.the.week Group 0 Group 1 Group 2  
## mean 3.891892 3.981735 3.675532  
## sd 1.264555 1.436572 1.405123  
##   
## ---------------------------------------------------------------------------------   
## ::: Seasons (Gaussian)   
## ---------------------------------------------------------------------------------   
##   
## Seasons Group 0 Group 1 Group 2  
## mean 3.243243 2.529680 2.473404  
## sd 1.064722 1.112766 1.158246  
##   
## ---------------------------------------------------------------------------------   
## ::: Transportation.expense (Gaussian)   
## ---------------------------------------------------------------------------------   
##   
## Transportation.expense Group 0 Group 1 Group 2  
## mean 248.51351 209.84703 247.85106  
## sd 78.94429 58.42178 74.35994  
##   
## ---------------------------------------------------------------------------------  
##   
## # ... and 10 more tables  
##   
## ---------------------------------------------------------------------------------

naive\_model\_pred = predict(naive\_model, testData)

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

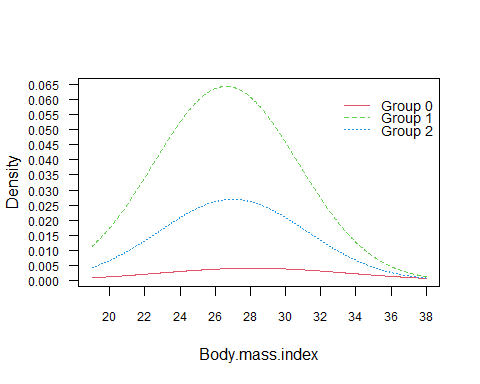
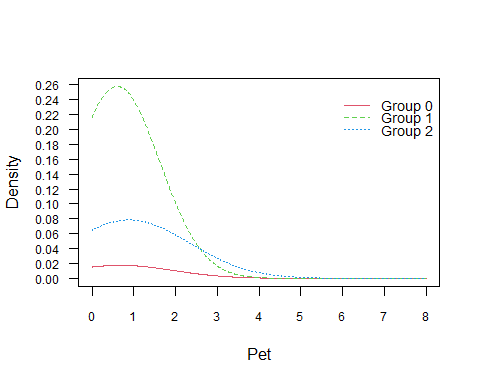
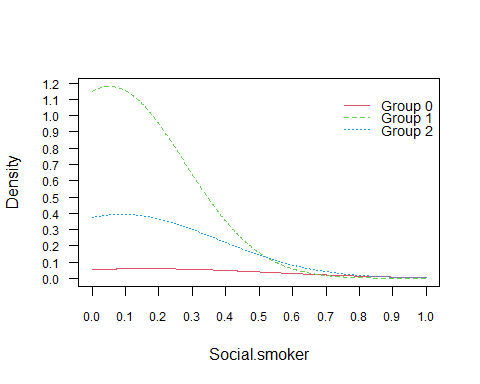
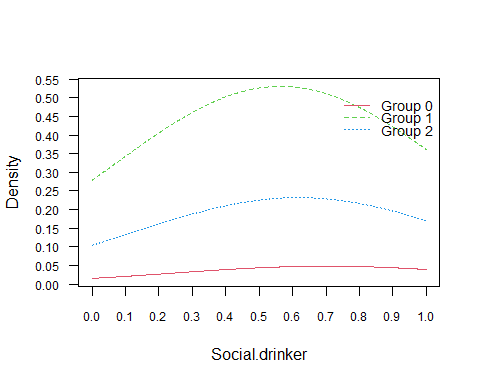
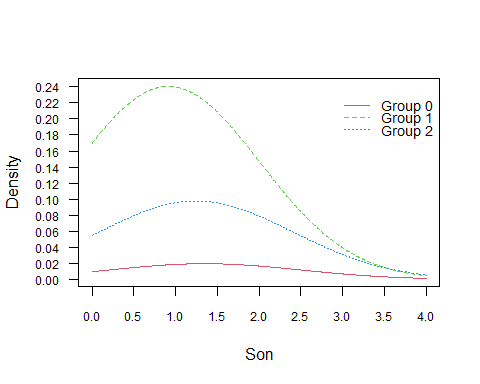
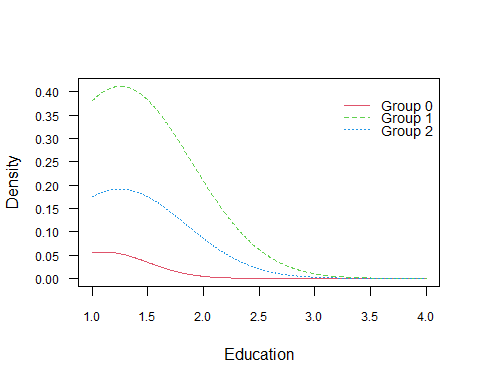
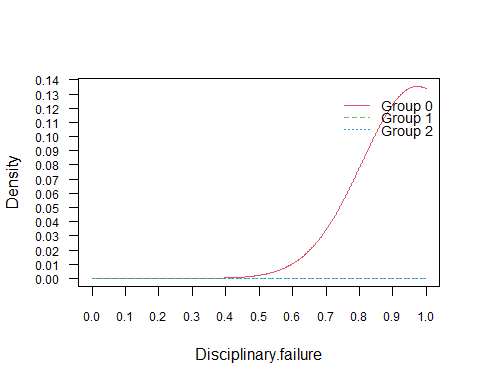
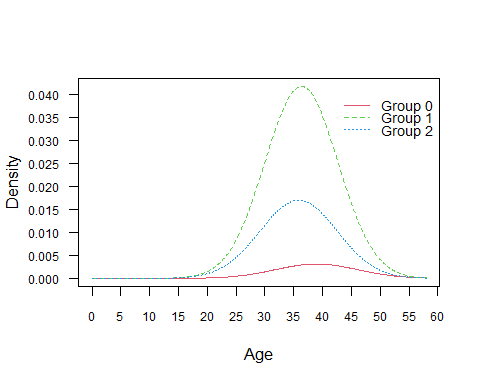
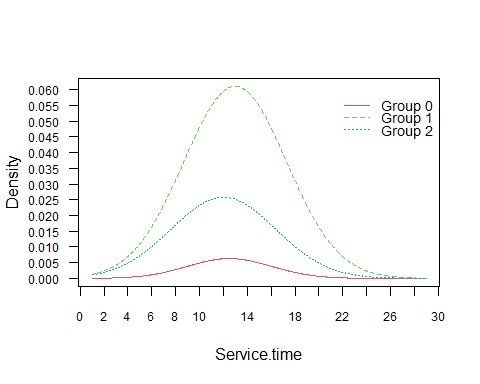
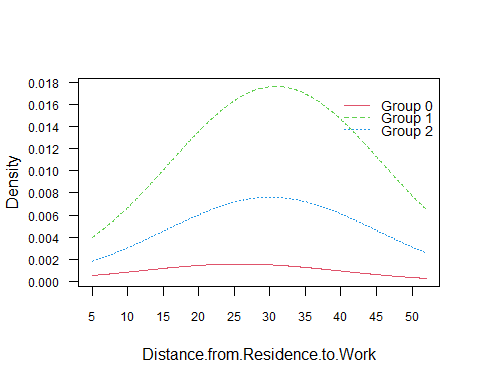
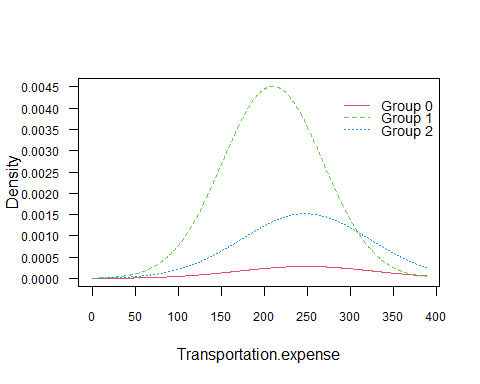
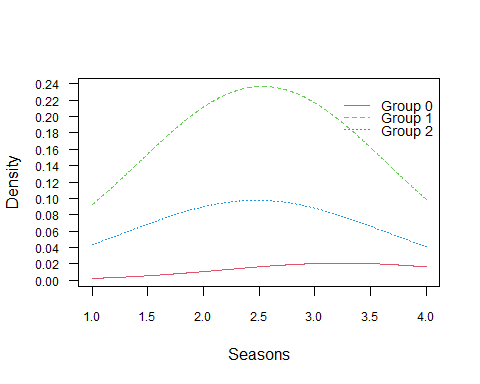
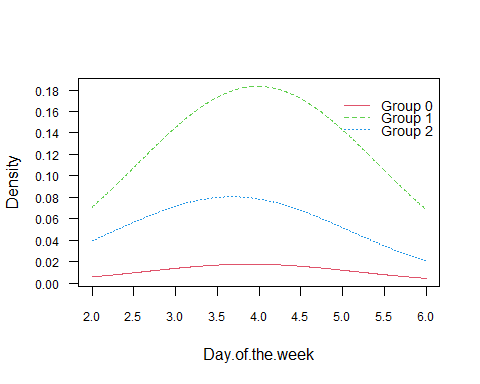
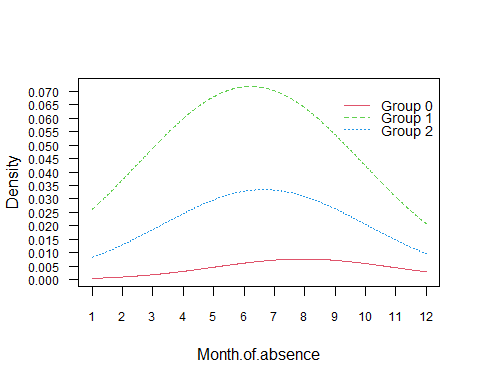
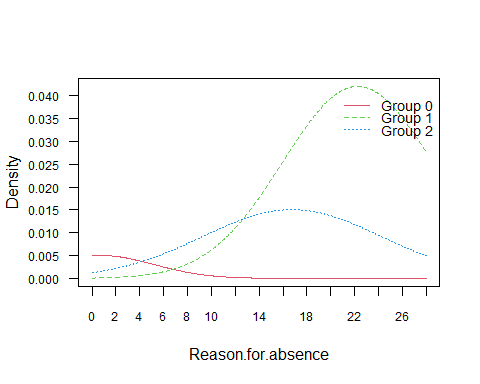
table(naive\_model\_pred, testData$Absenteeism.time.in.hours)

##   
## naive\_model\_pred Group 0 Group 1 Group 2  
## Group 0 4 0 0  
## Group 1 0 35 12  
## Group 2 3 7 13

##   
## absent\_predictions\_vanilla 0 1 2  
## 0 7 0 0  
## 1 1 77 6  
## 2 0 21 51  
agreement\_naive\_model\_pred= (naive\_model\_pred == testData$Absenteeism.time.in.hours)  
table(agreement\_naive\_model\_pred)

## agreement\_naive\_model\_pred  
## FALSE TRUE   
## 22 52

plot(naive\_model)



trainData\_ec <- read.csv("C:/Users/nagar/Documents/ANLY 530 Machine Learning/Project 1 - Abstenteeism/Absenteeism\_at\_work\_train.csv")  
trainData\_ec <- na.omit(trainData\_ec)  
trainData\_ec <- trainData\_ec[-c(1,10,11,18,19)]  
  
testData\_ec <- read.csv("C:/Users/nagar/Documents/ANLY 530 Machine Learning/Project 1 - Abstenteeism/Absenteeism\_at\_work\_test.csv")  
  
testData\_ec <- na.omit(testData\_ec)  
testData\_ec <- testData\_ec[-c(1,10,11,18,19)]  
  
  
mlr\_model <- lm(trainData\_ec$Absenteeism.time.in.hours ~., data = trainData\_ec)  
  
mlr\_model

##   
## Call:  
## lm(formula = trainData\_ec$Absenteeism.time.in.hours ~ ., data = trainData\_ec)  
##   
## Coefficients:  
## (Intercept) Reason.for.absence   
## 14.014316 -0.538449   
## Month.of.absence Day.of.the.week   
## -0.019848 -0.767817   
## Seasons Transportation.expense   
## 0.420462 -0.001785   
## Distance.from.Residence.to.Work Service.time   
## -0.056781 0.041334   
## Age27 Age28   
## -3.716328 -3.470583   
## Age29 Age30   
## -7.621527 0.355850   
## Age31 Age32   
## -6.406164 -6.553237   
## Age33 Age34   
## -3.085133 1.232106   
## Age36 Age37   
## -1.087801 -5.290325   
## Age38 Age39   
## -9.154062 -10.017114   
## Age40 Age41   
## -4.040645 -4.588580   
## Age43 Age46   
## -11.256420 -7.235828   
## Age47 Age48   
## -9.385046 -6.400292   
## Age49 Age50   
## -13.419140 -6.227916   
## Age58 AgeR   
## 11.034681 -8.259558   
## Disciplinary.failure Education   
## -18.899685 -0.973335   
## Son Social.drinker   
## -0.045155 3.634442   
## Social.smoker Pet   
## 1.794443 -0.253623   
## Body.mass.index   
## 0.447214