SITL Project

Pingng(Benny) Chong

April 18, 2019

```
library(glmnetUtils)
```

```
## Warning: package 'glmnetUtils' was built under R version 3.5.3
```

```
library(e1071)
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 3.5.3
```

```
## Loading required package: ggplot2
```

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.5.3
```

```
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:ggplot2':
##
##     margin
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(leaps)
library(rpart)


mms_testset1 <- read.csv(file = "C:/Users/hkcyf/Desktop/UNHSEM2/ML/Project/testset1.csv")


sitl <- read.csv(file = "C:/Users/hkcyf/Desktop/UNHSEM2/ML/Project/merged_201701-03.csv")
```

```
#mms= mms_testset1 %>% mutate(X1+1)

#lm.fit <- lm(DES.N~FGM.Bt, data = mms_testset1 )
#log.odds <- predict(glm.fit, mms.target)
#probabilities <- exp(log.odds) / (1 + exp(log.odds))
#probabilities <- predict(glm.fit, mms.target, type="response")


new_merge= subset(sitl, select = -c(1,2,19,21))
#write.csv(new_merge, 'new_merge.csv')
#merge_matrix <-as.matrix(sapply(new_merge, as.numeric))
#summary(lm.fit)

#summary(new_merge)

#df = subset(mms_testset1, select = -c(1,21))

#tree.test= tree(Selected~.-Priority, new_merge)

#pca.out = prcomp(df, scale=TRUE, center = TRUE)
#autoplot(pca.out, loadings = TRUE, loadings.label = TRUE)
#summary(pca.out)
new_bestsub <- regsubsets(Selected ~ ., data = new_merge, nvmax = 16)
coef(new_bestsub ,8)
```

```
##   (Intercept)         DES.N    DES.T_para    DES.T_perp        FGM.Bz
##  0.1033279376 -0.0016796741  0.0005259769 -0.0005544449  0.0017268776
##        FGM.Bt        DIS.Vz    DIS.T_para    DIS.T_perp
##  0.0021650997 -0.0003226277  0.0001375427 -0.0001524460
```

```
summary(new_bestsub)
```

```
## Subset selection object
## Call: regsubsets.formula(Selected ~ ., data = new_merge, nvmax = 16)
## 16 Variables  (and intercept)
##            Forced in Forced out
## DES.N          FALSE      FALSE
## DES.Vx         FALSE      FALSE
## DES.Vy         FALSE      FALSE
## DES.Vz         FALSE      FALSE
## DES.T_para     FALSE      FALSE
## DES.T_perp     FALSE      FALSE
## FGM.Bx         FALSE      FALSE
## FGM.By         FALSE      FALSE
## FGM.Bz         FALSE      FALSE
## FGM.Bt         FALSE      FALSE
## DIS.N          FALSE      FALSE
## DIS.Vx         FALSE      FALSE
## DIS.Vy         FALSE      FALSE
## DIS.Vz         FALSE      FALSE
## DIS.T_para     FALSE      FALSE
## DIS.T_perp     FALSE      FALSE
## 1 subsets of each size up to 16
```

```
## Selection Algorithm: exhaustive
##           DES.N DES.Vx DES.Vy DES.Vz DES.T_para DES.T_perp FGM.Bx FGM.By
## 1  ( 1 )  " "   " "    " "    " "    " "        " "        " "    " "
## 2  ( 1 )  " "   " "    " "    " "    " "        " "        " "    " "
## 3  ( 1 )  "*"   " "    " "    " "    " "        " "        " "    " "
## 4  ( 1 )  " "   " "    " "    " "    " "        " "        " "    " "
## 5  ( 1 )  "*"   " "    " "    " "    " "        " "        " "    " "
## 6  ( 1 )  "*"   " "    " "    " "    " "        "*"        " "    " "
## 7  ( 1 )  "*"   " "    " "    " "    " "        "*"        " "    " "
## 8  ( 1 )  "*"   " "    " "    " "    "*"        "*"        " "    " "
## 9  ( 1 )  "*"   " "    " "    " "    "*"        "*"        "*"    " "
## 10  ( 1 ) "*"   " "    "*"    " "    "*"        "*"        " "    " "
## 11  ( 1 ) "*"   " "    "*"    " "    "*"        "*"        "*"    " "
## 12  ( 1 ) "*"   "*"    "*"    " "    "*"        "*"        "*"    " "
## 13  ( 1 ) "*"   "*"    "*"    " "    "*"        "*"        "*"    " "
## 14  ( 1 ) "*"   "*"    "*"    " "    "*"        "*"        "*"    "*"
## 15  ( 1 ) "*"   "*"    "*"    "*"    "*"        "*"        "*"    "*"
## 16  ( 1 ) "*"   "*"    "*"    "*"    "*"        "*"        "*"    "*"
##           FGM.Bz FGM.Bt DIS.N DIS.Vx DIS.Vy DIS.Vz DIS.T_para DIS.T_perp
## 1  ( 1 )  "*"    " "    " "    " "    " "    " "    " "        " "
## 2  ( 1 )  "*"    " "    " "    " "    " "    " "    " "        "*"
## 3  ( 1 )  "*"    " "    " "    " "    " "    " "    " "        "*"
## 4  ( 1 )  "*"    "*"    "*"    " "    " "    " "    " "        "*"
## 5  ( 1 )  "*"    "*"    " "    " "    " "    " "    "*"        "*"
## 6  ( 1 )  "*"    "*"    " "    " "    " "    " "    "*"        "*"
## 7  ( 1 )  "*"    "*"    " "    " "    " "    "*"    "*"        "*"
## 8  ( 1 )  "*"    "*"    " "    " "    " "    "*"    "*"        "*"
## 9  ( 1 )  "*"    "*"    " "    " "    " "    "*"    "*"        "*"
## 10  ( 1 ) "*"    "*"    " "    " "    "*"    "*"    "*"        "*"
## 11  ( 1 ) "*"    "*"    " "    " "    "*"    "*"    "*"        "*"
## 12  ( 1 ) "*"    "*"    " "    " "    "*"    "*"    "*"        "*"
## 13  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    "*"        "*"
## 14  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    "*"        "*"
## 15  ( 1 ) "*"    "*"    " "    "*"    "*"    "*"    "*"        "*"
## 16  ( 1 ) "*"    "*"    "*"    "*"    "*"    "*"    "*"        "*"
```

```
attach(new_merge)
#Choose=ifelse (Selected >0, "Yes","No")
#Treeset = data.frame(new_merge, Choose)
#model_t<-tree(Choose ~ DES.N+DES.T_para+DES.T_perp+FGM.Bz+FGM.Bt+DIS.Vz+DIS.T_para+DIS.T_perp , Treese

#summary(model_t)

#plot(model_t)
#text(model_t ,pretty =0)

#model_t


#train.id= sample.int(nrow(Treeset),nrow(Treeset)*0.7)
#test.id= sample.int(nrow(Treeset),nrow(Treeset)*0.3)

#Tree.train= Treeset[train.id,]
```

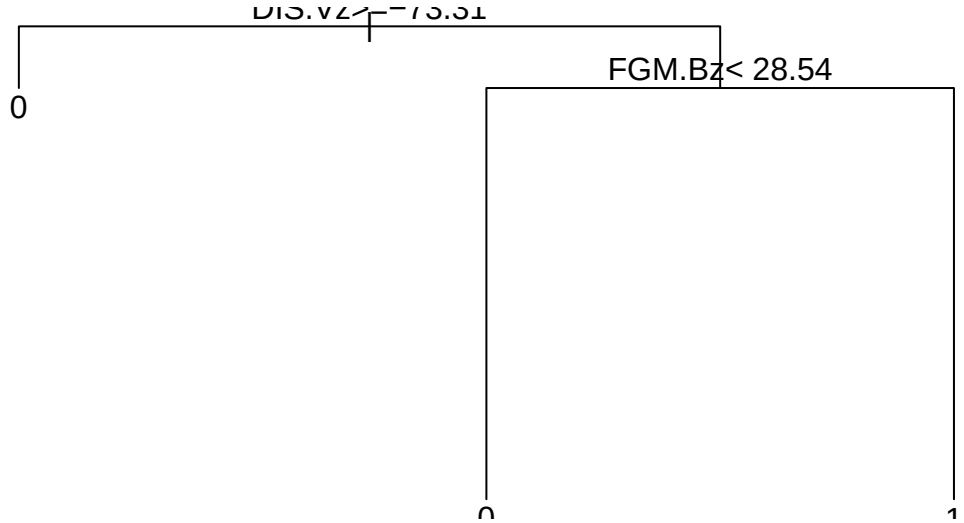
```
#model_t<-rpart(Selected ~ DES.N+DES.T_para+DES.T_perp+FGM.Bz+FGM.Bt+DIS.Vz+DIS.T_para+DIS.T_perp , met



#train=sample (1: nrow(Treeset ), 200)

#Choose.test=Choose[-train ]
#Tree.test= Treeset[-train]

#model_t<-tree(Choose ~ DES.N+DES.T_para+DES.T_perp+FGM.Bz+FGM.Bt+DIS.Vz+DIS.T_para+DIS.T_perp , Treese

#tree.pred<-predict(model_t, Tree.test, type ="class")

#table(tree.pred ,Choose.test)
```

First of all, we use try to find the importance of attributes to use such that we can avoid using all the features. FGM.Bz and DIS.T_prep are the most important, besides, we also need other features.

```
set.seed(123)
library(tree)
traintree=sample (1: nrow(new_merge), nrow(new_merge)/2)

#Tree.train= new_merge[train.id,]
Tree.test= new_merge[-traintree,]

model_t<-rpart(Selected ~DES.N+DES.T_para+DES.T_perp+FGM.Bz+FGM.Bt+DIS.Vz+DIS.T_para+DIS.T_perp, method



tree.pred<-predict(model_t, Tree.test, type ="class")
table(tree.pred ,Tree.test$Selected)
```

```
##
## tree.pred      0      1
##         0 177337  18700
##         1    584   1108
```

```
plot(model_t )
text(model_t ,pretty =0)
```



(177337+1108)/(177337+18700+584+1108)= 90% It looks good, but it does not give us enoguhe "selected" prediction. For 0(not selected), we quite accturately predict the true positive, however, our prediction has missed lots of 1(selected), we need to improve it.

Let's see what if we let the tree grow further using rpart.control

```
set.seed(123)
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```
library(RColorBrewer)
set.seed(123)
library(tree)

model_t2<-rpart(Selected ~DES.N+DES.T_para+DES.T_perp+FGM.Bz+FGM.Bt+DIS.Vz+DIS.T_para+DIS.T_perp, metho



tree.pred2<-predict(model_t2, Tree.test, type ="class")
table(tree.pred2 ,Tree.test$Selected)
```

```
##
## tree.pred2      0      1
##          0 177844   1641
##          1     77  18167
```

It seems that the result is much better, however, we still do not know whether there are overfitting problems. Nevertheless, it shows that using decision tree is a good way to go.