RF Travelers

Charlie, Audrey, Lukas, Li

12/4/2021

Packages

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.2
```

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.1.2
```

```
library(MASS)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

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

```
## Loading required package: lattice
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## 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(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.1.2
```

```
## Loaded gbm 2.1.8
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
##
## Attaching package: 'lubridate'
```

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

Pull in data

```
test <- read.csv("test_2021.csv")
train <- read.csv("train_2021.csv")
```

Formatting Data

```
#train
train$fraud <- as.factor(train$fraud)
train$year <- format(parse_date_time(train$claim_date, orders = c("ymd", "mdy", "dmy")),format="%Y")
train$month <- months(as.Date(parse_date_time(train$claim_date, orders = c("ymd", "mdy", "dmy"))))
train$year <- as.factor(train$year)
train$month <- as.factor(train$month)
train$gender <- as.factor(train$gender)
train$marital_status <- as.factor(train$marital_status)
train$high_education_ind <- as.factor(train$high_education_ind)
train$address_change_ind <- as.factor(train$address_change_ind)
train$living_status <- as.factor(train$living_status)
train$zip_code <- as.factor(train$zip_code)
train$claim_day_of_week <- as.factor(train$claim_day_of_week)
train$accident_site <- as.factor(train$accident_site)
train$witness_present_ind <- as.factor(train$witness_present_ind)
train$channel <- as.factor(train$channel)
train$policy_report_filed_ind <- as.factor(train$policy_report_filed_ind)
train$vehicle_category <- as.factor(train$vehicle_category)
train$vehicle_color <- as.factor(train$vehicle_color)
train$marital_status[is.na(train$marital_status)] = 0
train$witness_present_ind[is.na(train$witness_present_ind)] = 0
train$claim_est_payout[is.na(train$claim_est_payout)] =
  mean(train$claim_est_payout,na.rm=TRUE)
train$age_of_vehicle[is.na(train$age_of_vehicle)] =
  mean(train$age_of_vehicle,na.rm=TRUE)
#test
```

```
test$gender <- as.factor(test$gender)
test$marital_status <- as.factor(test$marital_status)
test$high_education_ind <- as.factor(test$high_education_ind)
test$address_change_ind <- as.factor(test$address_change_ind)
test$living_status <- as.factor(test$living_status)
test$zip_code <- as.factor(test$zip_code)
test$claim_day_of_week <- as.factor(test$claim_day_of_week)
test$accident_site <- as.factor(test$accident_site)
test$witness_present_ind <- as.factor(test$witness_present_ind)
test$channel <- as.factor(test$channel)
test$policy_report_filed_ind <- as.factor(test$policy_report_filed_ind)
test$vehicle_category <- as.factor(test$vehicle_category)
test$vehicle_color <- as.factor(test$vehicle_color)
test$year <- format(parse_date_time(test$claim_date, orders = c("ymd", "mdy", "dmy")),format="%Y")
test$month <- months(as.Date(parse_date_time(test$claim_date, orders = c("ymd", "mdy", "dmy"))))
test$year <- as.factor(test$year)
test$month <- as.factor(test$month)
#NA Check
na_count <-sapply(test, function(y) sum(length(which(is.na(y)))))
data.frame(na_count)
```

```
##                         na_count
## claim_number                   0
## age_of_driver                  0
## gender                         0
## marital_status                 2
## safty_rating                   0
## annual_income                  0
## high_education_ind             0
## address_change_ind             0
## living_status                  0
## zip_code                       0
## claim_date                     0
## claim_day_of_week              0
## accident_site                  0
## past_num_of_claims             0
## witness_present_ind           88
## liab_prct                      0
## channel                        0
## policy_report_filed_ind        0
## claim_est_payout              14
## age_of_vehicle                 3
## vehicle_category               0
## vehicle_price                  0
## vehicle_color                  0
## vehicle_weight                 0
## year                           0
## month                          0
```

```
test$marital_status[is.na(test$marital_status)] = 0
test$witness_present_ind[is.na(test$witness_present_ind)] = 0
test$claim_est_payout[is.na(test$claim_est_payout)] =
  mean(train$claim_est_payout,na.rm=TRUE)
```

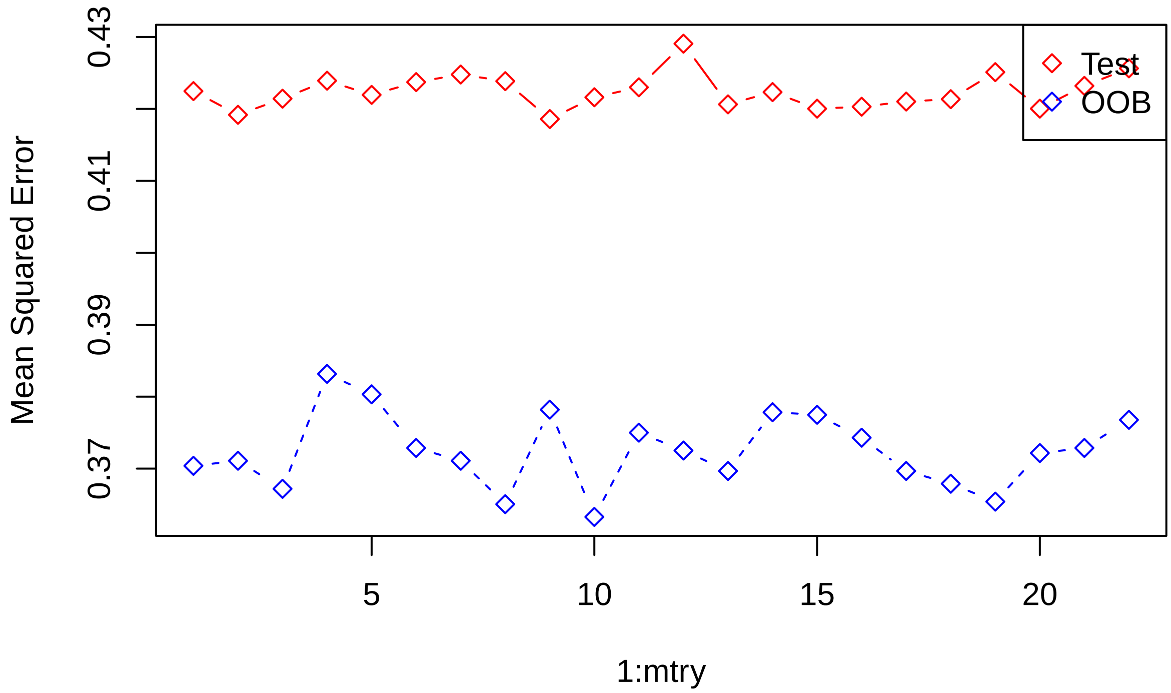
```
test$age_of_vehicle[is.na(test$age_of_vehicle)] =
  mean(train$age_of_vehicle,na.rm=TRUE)
```

Random Forest

```
set.seed(8051)
rF <- data.frame(Accuracy=0, Recall=0, Precision=0, Fmeasure=0,oob.err=0,test.err=0)
pos <- which(train$fraud==1)
neg <- which(train$fraud==0)
posdata <- train[pos,]
negdata <- train[neg,]
samp <- c(sample(pos,1408),sample(neg,1408))
traindata <- train[samp,]
sub <- as.numeric(train$fraud[-samp])-1
rf.fraud <- randomForest(fraud~age_of_driver+gender+
    marital_status+safty_rating+annual_income+high_education_ind+address_change_ind+
    living_status+claim_day_of_week+accident_site+past_num_of_claims+witness_present_ind+
    liab_prct+channel+policy_report_filed_ind+claim_est_payout+age_of_vehicle+
    vehicle_price+vehicle_weight+year+month,data=train,subset=samp,ntree=350)
oob.err = double(22)
test.err = double(22)
n.tree = 500
for(mtry in 1:22){
  mod = randomForest(fraud~age_of_driver+gender+
    marital_status+safty_rating+annual_income+high_education_ind+address_change_ind+
    living_status+claim_day_of_week+accident_site+past_num_of_claims+witness_present_ind+
    liab_prct+channel+policy_report_filed_ind+claim_est_payout+age_of_vehicle+
    vehicle_price+vehicle_weight+year+month,data=train,subset=samp,ntree=n.tree)
  oob.err[mtry] = mod$err.rate[n.tree,1]
  pred = as.numeric(predict(mod, train[-samp,]))-1
  test.err[mtry] = with(train[-samp,], mean( (sub-pred)^2 ))
  rf.pred <- predict(mod,newdata = train[-samp,])
  expected_value <- factor(train[-samp,]$fraud)
  predicted_value <- factor(rf.pred)
  CM <- confusionMatrix(data=predicted_value, reference = expected_value,positive = "1")
  acc <- CM$overall[1]
  re = CM$byClass[1]
  prec = CM$byClass[5]
  F1= 2*prec*re/(prec+re)
  #data.frame(FMeasure = 2 * prec * re / (prec + re),row.names = NULL)
  rF <- rbind(rF, c(Accuracy = acc, Recall = re, Precision = prec, Fmeasure = F1,
                    oob.err = oob.err[mtry],test.err=test.err[mtry]))
}
rF
```

```
##     Accuracy    Recall Precision  Fmeasure   oob.err  test.err
## 1  0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## 2  0.5780530 0.7031250 0.1418745 0.2361078 0.3703835 0.4224740
## 3  0.5812805 0.7009943 0.1425683 0.2369463 0.3710938 0.4191806
## 4  0.5790410 0.7038352 0.1422828 0.2367132 0.3671875 0.4214201
## 5  0.5758135 0.7073864 0.1417995 0.2362429 0.3831676 0.4239231
## 6  0.5784482 0.7038352 0.1420992 0.2364591 0.3803267 0.4219470
## 7  0.5759452 0.7009943 0.1409195 0.2346648 0.3728693 0.4237255
## 8  0.5748255 0.7116477 0.1421075 0.2369074 0.3710938 0.4247793
## 9  0.5769991 0.7095170 0.1424701 0.2372922 0.3650568 0.4238572
## 10 0.5814122 0.7002841 0.1425061 0.2368200 0.3781960 0.4185878
## 11 0.5787116 0.6953125 0.1409444 0.2343787 0.3632812 0.4216177
## 12 0.5771967 0.7045455 0.1418156 0.2361062 0.3750000 0.4230009
## 13 0.5707417 0.7052557 0.1399577 0.2335646 0.3725142 0.4290607
## 14 0.5790410 0.6953125 0.1410460 0.2345191 0.3696733 0.4206297
## 15 0.5777236 0.6981534 0.1410532 0.2346902 0.3778409 0.4223422
## 16 0.5802266 0.6931818 0.1411016 0.2344745 0.3774858 0.4200369
## 17 0.5793044 0.7095170 0.1431847 0.2382826 0.3742898 0.4203004
## 18 0.5793044 0.7045455 0.1424673 0.2370087 0.3696733 0.4210249
## 19 0.5786458 0.7038352 0.1421604 0.2365437 0.3678977 0.4213542
## 20 0.5752865 0.7080966 0.1417401 0.2362000 0.3654119 0.4251087
## 21 0.5796996 0.6974432 0.1415598 0.2353505 0.3721591 0.4200369
## 22 0.5769332 0.6924716 0.1399856 0.2328914 0.3728693 0.4231985
## 23 0.5742985 0.6981534 0.1400085 0.2332424 0.3767756 0.4256356
```

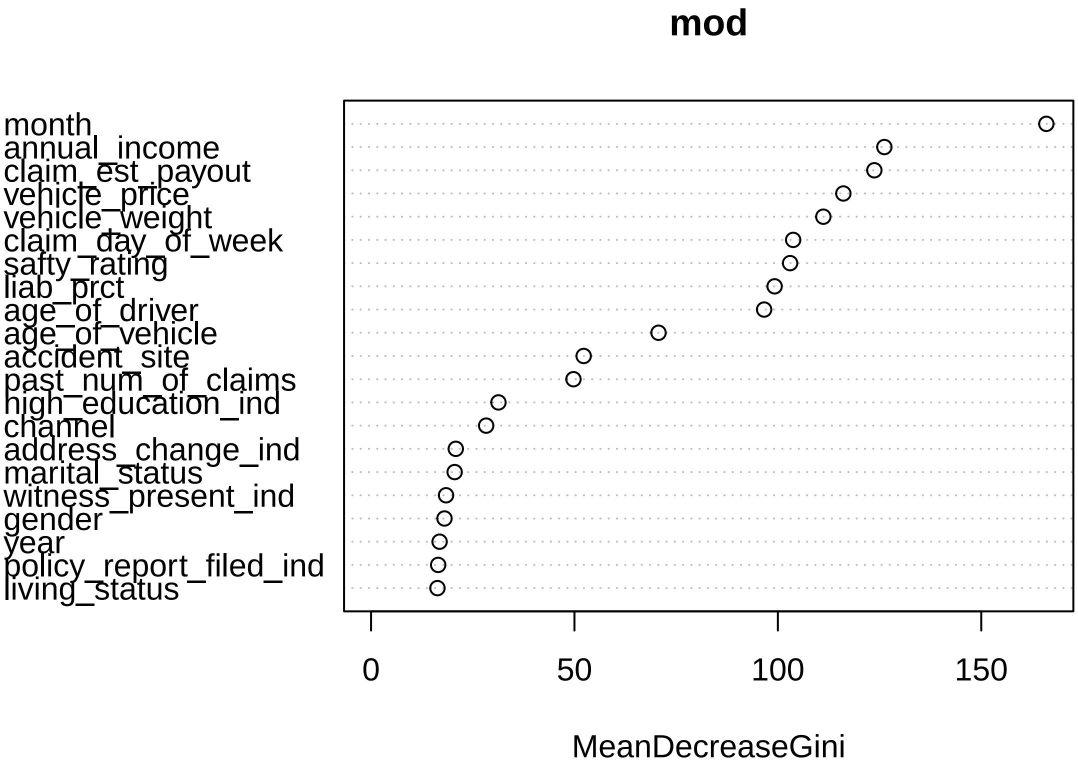
```
matplot(1:mtry, cbind(test.err, oob.err), pch = 23, col = c("red", "blue"), type = "b", ylab="Mean Squa
legend("topright", legend = c("Test", "OOB"), pch = 23, col = c("red", "blue"))
```



```
importance(mod)
```

```
##                            MeanDecreaseGini
## age_of_driver                      96.62860
## gender                             18.04599
## marital_status                    20.55112
## safty_rating                     103.01132
## annual_income                    126.18075
## high_education_ind                31.31896
## address_change_ind                20.82915
## living_status                     16.32201
## claim_day_of_week                103.76901
## accident_site                     52.26631
## past_num_of_claims                49.75835
## witness_present_ind               18.43456
## liab_prct                         99.20607
## channel                           28.29499
## policy_report_filed_ind           16.50014
## claim_est_payout                 123.71801
## age_of_vehicle                    70.66908
## vehicle_price                    116.10589
## vehicle_weight                   111.18121
## year                              16.85132
## month                            166.00243
```

```
varImpPlot(mod,scale=FALSE)
```



```
F1
```

```
## Precision
## 0.2332424
```