Course Project for STAT 8051

Charlie, Audrey, Lukas, Li

12/11/2021

Pull in data

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

Library the packages

```
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
```

```
library(glmm)
```

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

```
## Loading required package: trust
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: Matrix
```

```
## Loading required package: parallel
```

```
## Loading required package: doParallel
```

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

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

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

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

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

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

```
library(ROSE)
```

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

```
## Loaded ROSE 0.0-4
```

```
library(cvTools)
```

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

```
## Loading required package: robustbase
```

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

```
library(ROCR)
```

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

Diagnostics

```
str(train)
```

```
## 'data.frame':    17998 obs. of  25 variables:
##  $ claim_number           : int  1 3 4 5 6 7 8 10 12 15 ...
##  $ age_of_driver          : int  46 21 49 58 38 56 27 40 45 31 ...
##  $ gender                 : chr  "M" "F" "F" "F" ...
##  $ marital_status         : num  1 0 0 1 1 0 1 1 0 1 ...
##  $ safty_rating           : int  85 75 87 58 95 72 80 79 86 80 ...
##  $ annual_income          : int  38301 30445 38923 40605 36380 40240 32952 36891 38069 34324 ...
##  $ high_education_ind     : int  1 0 0 1 1 0 1 1 0 0 ...
##  $ address_change_ind     : int  1 1 1 0 0 0 0 1 1 1 ...
##  $ living_status          : chr  "Rent" "Rent" "Own" "Own" ...
##  $ zip_code               : int  80006 15021 20158 15024 50034 50047 15001 80016 15024 85032 ...
##  $ claim_date             : chr  "12/16/2016" "2/12/2015" "12/6/2016" "5/5/2016" ...
##  $ claim_day_of_week      : chr  "Friday" "Thursday" "Tuesday" "Thursday" ...
##  $ accident_site          : chr  "Local" "Highway" "Local" "Local" ...
##  $ past_num_of_claims     : int  1 1 0 3 0 0 0 5 0 0 ...
##  $ witness_present_ind    : num  0 1 0 0 1 0 0 0 0 0 ...
```

```
##  $ liab_prct              : int  74 79 0 99 7 64 50 95 3 9 ...
##  $ channel                : chr  "Broker" "Online" "Broker" "Broker" ...
##  $ policy_report_filed_ind: int  0 0 0 1 0 0 1 1 0 0 ...
##  $ claim_est_payout       : num  7531 2966 6284 6170 4541 ...
##  $ age_of_vehicle         : num  9 4 3 4 7 4 7 8 5 5 ...
##  $ vehicle_category       : chr  "Compact" "Large" "Compact" "Medium" ...
##  $ vehicle_price          : num  12885 29429 21701 13198 38060 ...
##  $ vehicle_color          : chr  "white" "white" "white" "other" ...
##  $ vehicle_weight         : num  16161 28692 22091 38330 25877 ...
##  $ fraud                  : int  0 0 1 1 0 1 0 0 0 0 ...
```

The data is havily skewed

```
table(train$fraud)
```

```
##
##     0     1
## 15182  2816
```

Data processing

Creat year and month var using claim_date

```
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)
```

Remain the first three digits of zip_code and treat it as factor variable

```
train$zip_code <- floor(train$zip_code/100)
train$zip_code <- as.factor(train$zip_code)
```

Factorization-characters and binaries to factors for modeling

```
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$claim_day_of_week <- as.factor(train$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)
str(train)
```

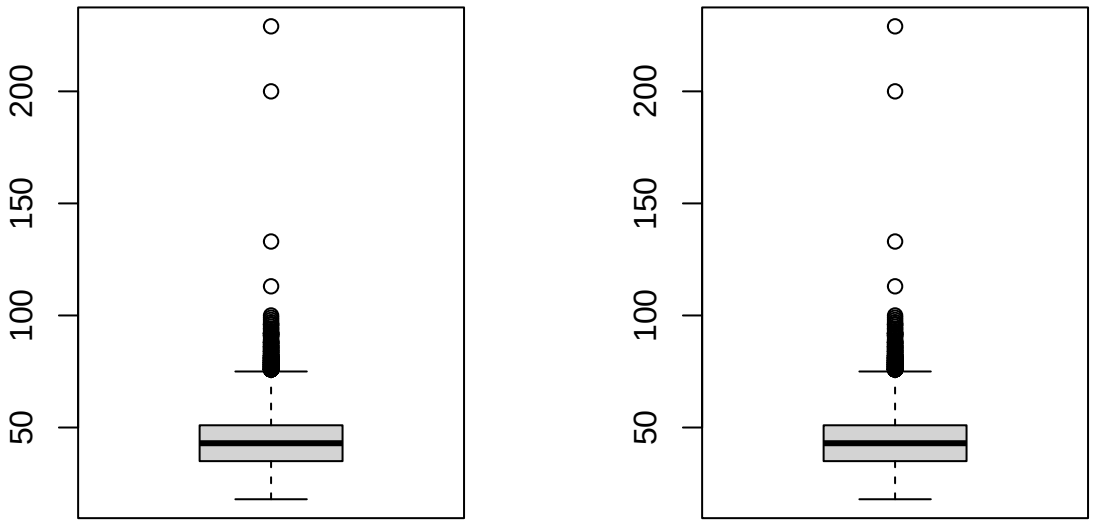
```
## 'data.frame':    17998 obs. of  27 variables:
##  $ claim_number           : int  1 3 4 5 6 7 8 10 12 15 ...
##  $ age_of_driver          : int  46 21 49 58 38 56 27 40 45 31 ...
##  $ gender                 : Factor w/ 2 levels "F","M": 2 1 1 1 2 1 2 2 1 2 ...
##  $ marital_status         : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 2 2 1 2 ...
##  $ safty_rating           : int  85 75 87 58 95 72 80 79 86 80 ...
##  $ annual_income          : int  38301 30445 38923 40605 36380 40240 32952 36891 38069 34324 ...
##  $ high_education_ind     : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 2 2 1 1 ...
##  $ address_change_ind     : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 2 2 2 ...
##  $ living_status          : Factor w/ 2 levels "Own","Rent": 2 2 1 1 2 1 2 1 1 2 ...
##  $ zip_code               : Factor w/ 7 levels "0","150","201",..: 5 2 3 2 4 4 2 5 2 7 ...
##  $ claim_date             : chr  "12/16/2016" "2/12/2015" "12/6/2016" "5/5/2016" ...
##  $ claim_day_of_week      : Factor w/ 7 levels "Friday","Monday",..: 1 5 6 5 6 7 3 5 6 5 ...
##  $ accident_site          : Factor w/ 3 levels "Highway","Local",..: 2 1 2 2 1 1 3 2 3 3 ...
##  $ past_num_of_claims     : int  1 1 0 3 0 0 0 5 0 0 ...
##  $ witness_present_ind    : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 1 1 1 ...
##  $ liab_prct              : int  74 79 0 99 7 64 50 95 3 9 ...
##  $ channel                : Factor w/ 3 levels "Broker","Online",..: 1 2 1 1 1 3 2 2 1 1 ...
##  $ policy_report_filed_ind: Factor w/ 2 levels "0","1": 1 1 1 2 1 1 2 2 1 1 ...
##  $ claim_est_payout       : num  7531 2966 6284 6170 4541 ...
##  $ age_of_vehicle         : num  9 4 3 4 7 4 7 8 5 5 ...
##  $ vehicle_category       : Factor w/ 3 levels "Compact","Large",..: 1 2 1 3 3 3 1 3 1 2 ...
##  $ vehicle_price          : num  12885 29429 21701 13198 38060 ...
##  $ vehicle_color          : Factor w/ 7 levels "black","blue",..: 7 7 7 4 3 1 5 4 1 7 ...
##  $ vehicle_weight         : num  16161 28692 22091 38330 25877 ...
##  $ fraud                  : int  0 0 1 1 0 1 0 0 0 0 ...
##  $ year                   : Factor w/ 2 levels "2015","2016": 2 1 2 2 1 2 1 2 1 1 ...
##  $ month                  : Factor w/ 12 levels "April","August",..: 3 4 3 9 11 10 8 5 8 12 ...
```

Dealing with NA in training data

```
train <- na.omit(train)
```

Dealing with outliers in trainning data

```
par(mfrow=c(1,2))
boxplot(train$age_of_driver)
boxplot(train$age_of_driver)
```



Age of the driver

```
age_threshold <- quantile(train$age_of_driver,0.99)
```

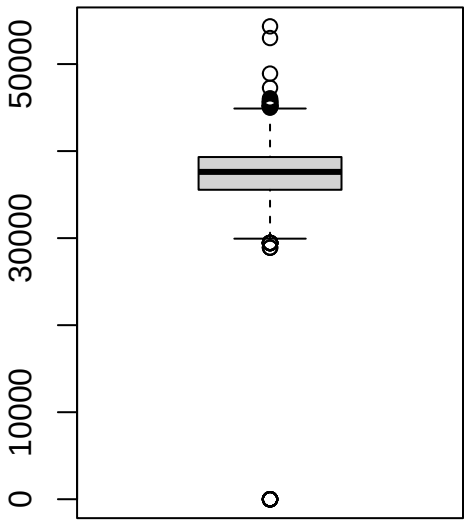
```
tab.age <- table(train$fraud,train$age_of_driver>age_threshold)
prop.table(tab.age)
```

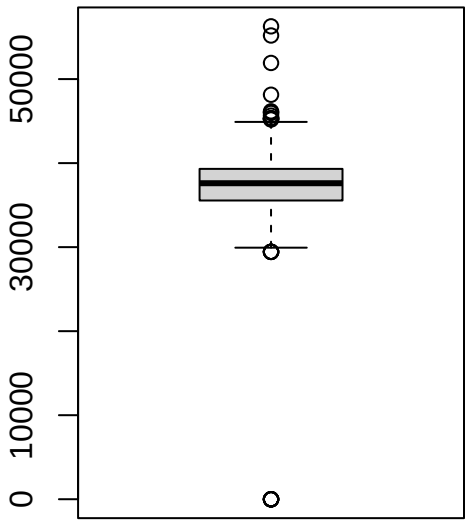
```
##
##           FALSE        TRUE
##   0 0.835389101 0.007849294
##   1 0.156032743 0.000728863
```

```
chisq.test(tab.age,simulate.p.value = FALSE)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab.age
## X-squared = 5.4822, df = 1, p-value = 0.01921
```

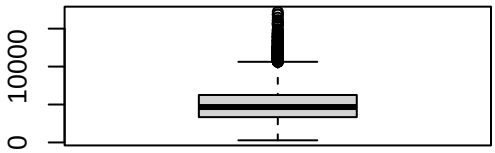
```
par(mfrow=c(1,2))
boxplot(train$annual_income)
boxplot(test$annual_income)
```

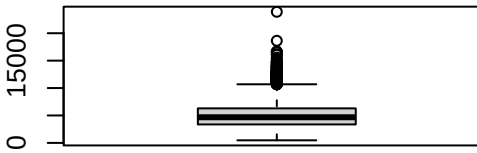




Annual income

```
par(mfrow=c(2,2))
boxplot(train$claim_est_payout)
boxplot(test$claim_est_payout)
```

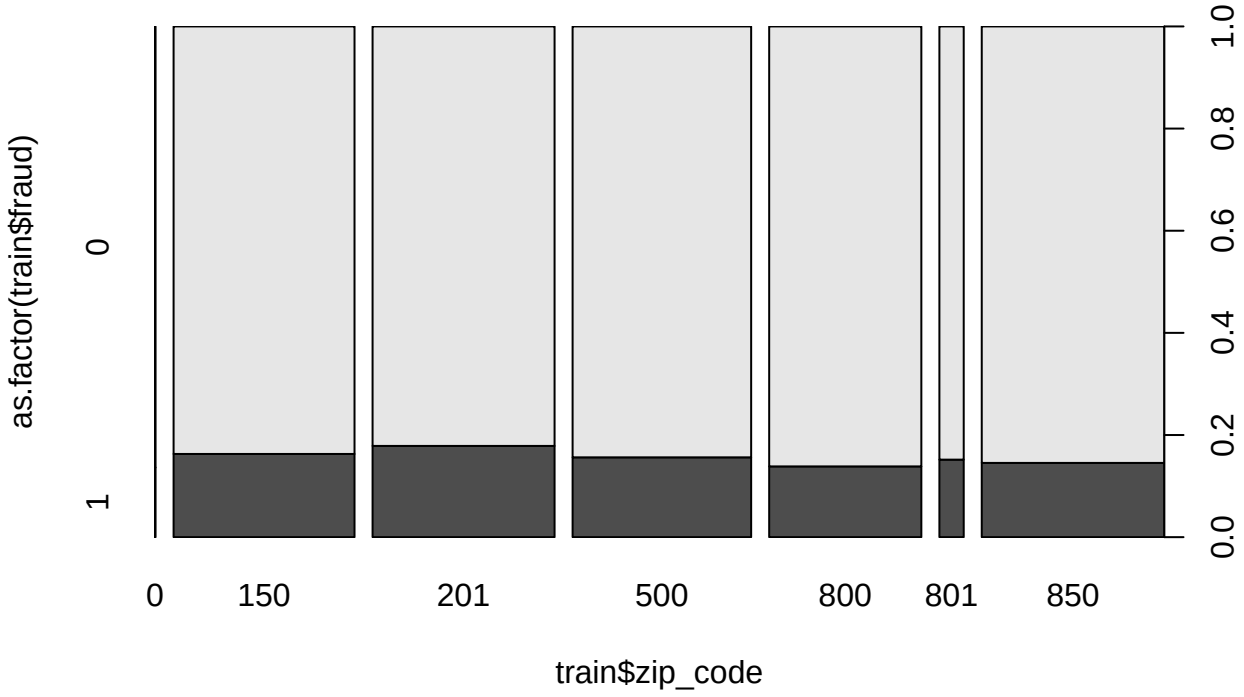




claim_est_payout

Data graphing

```
plot(as.factor(train$fraud)~train$zip_code)
```




```
tab.zip <- table(as.factor(train$fraud),train$zip_code)
chisq.test(tab.zip,simulate.p.value = TRUE)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  tab.zip
## X-squared = 25.307, df = NA, p-value = 0.0009995
```

Logistic regression

Stepwise AIC to select the best variable

```
BigMod <- glm(fraud~age_of_driver+gender+
    marital_status+safty_rating+annual_income+high_education_ind+address_change_ind+
    living_status+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+zip_code,family=binomial,data=train)

stepAIC(BigMod, scope = list(upper=~age_of_driver+gender+
    marital_status+safty_rating+annual_income+high_education_ind+address_change_ind+
    living_status+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+zip_code,
    lower=~1),direction="both",trace=0,k=2,data=train)
```

```
##
## Call:  glm(formula = fraud ~ age_of_driver + gender + marital_status +
##     safty_rating + annual_income + high_education_ind + address_change_ind +
##     living_status + accident_site + past_num_of_claims + witness_present_ind +
##     channel + claim_est_payout + age_of_vehicle + year + zip_code,
##     family = binomial, data = train)
##
## Coefficients:
##              (Intercept)             age_of_driver                    genderM
##                -4.617e+00                -5.772e-02                -2.831e-01
##          marital_status1              safty_rating             annual_income
##                -4.635e-01                -1.037e-02                 1.806e-04
##      high_education_ind1       address_change_ind1         living_statusRent
##                -6.147e-01                 4.399e-01                 1.453e-01
##       accident_siteLocal  accident_siteParking Lot        past_num_of_claims
##                -2.420e-01                -1.069e+00                 3.135e-01
##     witness_present_ind1             channelOnline               channelPhone
##                -7.244e-01                -1.413e-01                 2.631e-02
##         claim_est_payout            age_of_vehicle                  year2016
##                -2.109e-05                 6.996e-02                 2.467e-01
##              zip_code150               zip_code201               zip_code500
##                 1.115e-02                 1.511e-01                -5.386e-02
##              zip_code800               zip_code801               zip_code850
##                -2.162e-01                -5.520e-02                -1.627e-01
##
## Degrees of Freedom: 17835 Total (i.e. Null);  17812 Residual
## Null Deviance:       15490
## Residual Deviance: 14150     AIC: 14200
```

Cross-validation to choose cutoff point

```
dat <- train
n.fold <- 5
folds <- cvFolds(nrow(dat),n.fold)
result <- vector("list",n.fold)
for (k in 1:n.fold){
    dat.train <- dat[folds$subsets[folds$which!=k],]
    dat.test <- dat[folds$subsets[folds$which==k],]
    dat.train.over <- ovun.sample(fraud~.,data=dat.train,method = "over",p=0.5)$data
    mod <- glm(formula = fraud ~ age_of_driver+marital_status + gender +
      safty_rating + annual_income + high_education_ind + address_change_ind +
      living_status + accident_site + past_num_of_claims + witness_present_ind +
      channel + claim_est_payout + age_of_vehicle + year+ zip_code, family =
      binomial, data = dat.train.over)
    predict.mod <- predict(mod,newdata = dat.test,type = "response")
    pred <- prediction(predict.mod,dat.test$fraud)
    per <- performance(pred,"f")
```

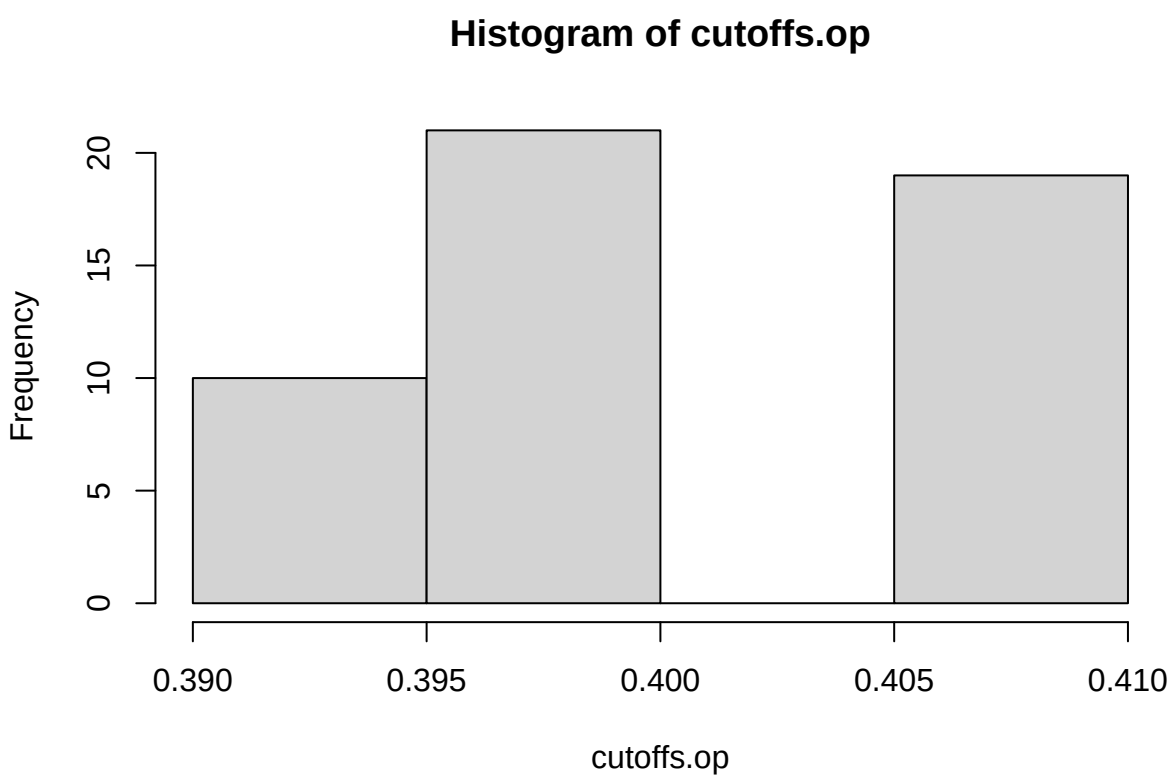
```
    result[[k]] <- per@x.values[[1]][which.max(per@y.values[[1]])]
}
```

```
cutoffs.pre <- result
```

```
cv.cutoff <- function(dat,n.fold,cutoff.list){
  folds <- cvFolds(nrow(dat),n.fold)
  p <- length(cutoff.list)
  result <- matrix(0,length(cutoff.list),n.fold)
  for (k in 1:n.fold){
    dat.train <- dat[folds$subsets[folds$which!=k],]
    dat.test <- dat[folds$subsets[folds$which==k],]
    dat.train.over <- ovun.sample(fraud~.,data=dat.train,method = "over",p=0.4)$data
    mod <- glm(formula = fraud ~ age_of_driver+marital_status + gender +
      safty_rating + annual_income + high_education_ind + address_change_ind +
      living_status + accident_site + past_num_of_claims + witness_present_ind +
      channel + claim_est_payout + age_of_vehicle + year+ zip_code, family =
      binomial, data = dat.train.over)
    predict.mod <- predict(mod,newdata = dat.test,type = "response")
    for (j in 1:p){
      pred <- ifelse(predict.mod > cutoff.list[j],1,0)
      expected_value <- factor(dat.test$fraud)
      predicted_value <- factor(pred)
      CM <- confusionMatrix(data=predicted_value, reference = expected_value,positive = "1")
      re = CM$byClass[1]
      prec = CM$byClass[5]
      result[j,k] = 2 * prec * re / (prec + re)
    }
  }
  Fs <- apply(result,1,mean)
  return(list(cutoff.list = cutoff.list,F = Fs))
}
```

```
cutoffs.op <- numeric(50)
best_values <- numeric(50)
for (k in 1: 50){
  cutoff.list <- seq(0.39,0.41,0.01)
  cutoffs <- cv.cutoff(train,5,cutoff.list)
  cutoffs.op[k] <- cutoffs$cutoff.list[which.max(cutoffs$F)]
  best_values[k] <- max(cutoffs$F)
}
```

```
hist(cutoffs.op)
```




```
cutoff.list <- seq(0.25,0.35,0.01)
cutoffs <- cv.cutoff(train,5,cutoff.list)
cutoff.op <- cutoffs$cutoff.list[which.max(cutoffs$F)]
```

Fitting to test data

```
#Organize test in the same way as train
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$zip_code <- floor(test$zip_code/100)
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)
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(na.omit(test$claim_est_payout))
test$age_of_vehicle[is.na(test$age_of_vehicle)] <- mean(na.omit(test$age_of_vehicle))
```

```
train.over <- ovun.sample(fraud~.,data=train,method = "over",p=0.4)$data
```

```
mod <- glm(formula = fraud ~ age_of_driver+marital_status + gender +
      safty_rating + annual_income + high_education_ind + address_change_ind +
      living_status + accident_site + past_num_of_claims + witness_present_ind +
      channel + claim_est_payout + age_of_vehicle + year+ zip_code, family =
      binomial, data = train.over)
newval <- predict(mod,newdata=test,type="response")
submit <- ifelse(newval>0.4,1,0)
mat <- cbind(claim_number=test$claim_number,pred=submit)
```

Writing into CSV format for submission

```
write.csv(mat,file="Prediction - Logistic Regression.csv",row.names = FALSE)
```