Finity - Grads Stats Exercise

2023-02-08

The attached data set contains information on policies purchased from a motor insurer. Each row in the data represents a policy and includes various predictor variables, along with the target variable "claim_flag", which indicates whether the policy had a claim during the coverage period of one year.

The goal of this report is to build a model that predicts whether a policy will make a claim based on the predictor variables and analyze the impact of these variables on the likelihood of making a claim.

Data Check

The data contain 7 predictors. While car_age, driver_age and density are continuous, power, brand, gas and region are discrete with 12, 7, 2 and 10 levels respectively as shown below in the data structure.

```
str(motor)
```

```
## 'data.frame':    413169 obs. of  8 variables:
##  $ claim_flag: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ power     : Factor w/ 12 levels "d","e","f","g",..: 4 4 3 3 4 4 1 1 1 6 ...
##  $ car_age   : int  0 0 2 2 0 0 1 0 9 0 ...
##  $ driver_age: int  46 46 38 38 41 41 27 27 23 44 ...
##  $ brand     : Factor w/ 7 levels "Fiat","Japanese (except Nissan) or Korean",..: 2 2 2 2 2 2 2 2 1 2
##  $ gas       : Factor w/ 2 levels "Diesel","Regular": 1 1 2 2 1 1 2 2 2 2 ...
##  $ region    : Factor w/ 10 levels "Aquitaine","Basse-Normandie",..: 1 1 8 8 9 9 1 1 8 6 ...
##  $ density   : int  76 76 3003 3003 60 60 695 695 7887 27000 ...
```

The data is well-built with no missing value.

```
colSums(is.na(motor))
```

```
## claim_flag      power    car_age driver_age      brand        gas     region
##          0          0          0          0          0          0          0
##    density
##          0
```

But the data set have a number of duplicated data which shall be removed. However, considering the nature of rental car market, same car can be ranted for a number of times so this problem is mild. From modelling point of view, the duplicated data will be placed with greater emphasis.

```
sum(duplicated(motor))
```

```
## [1] 108373
```

```
#motor<-unique(motor)
```

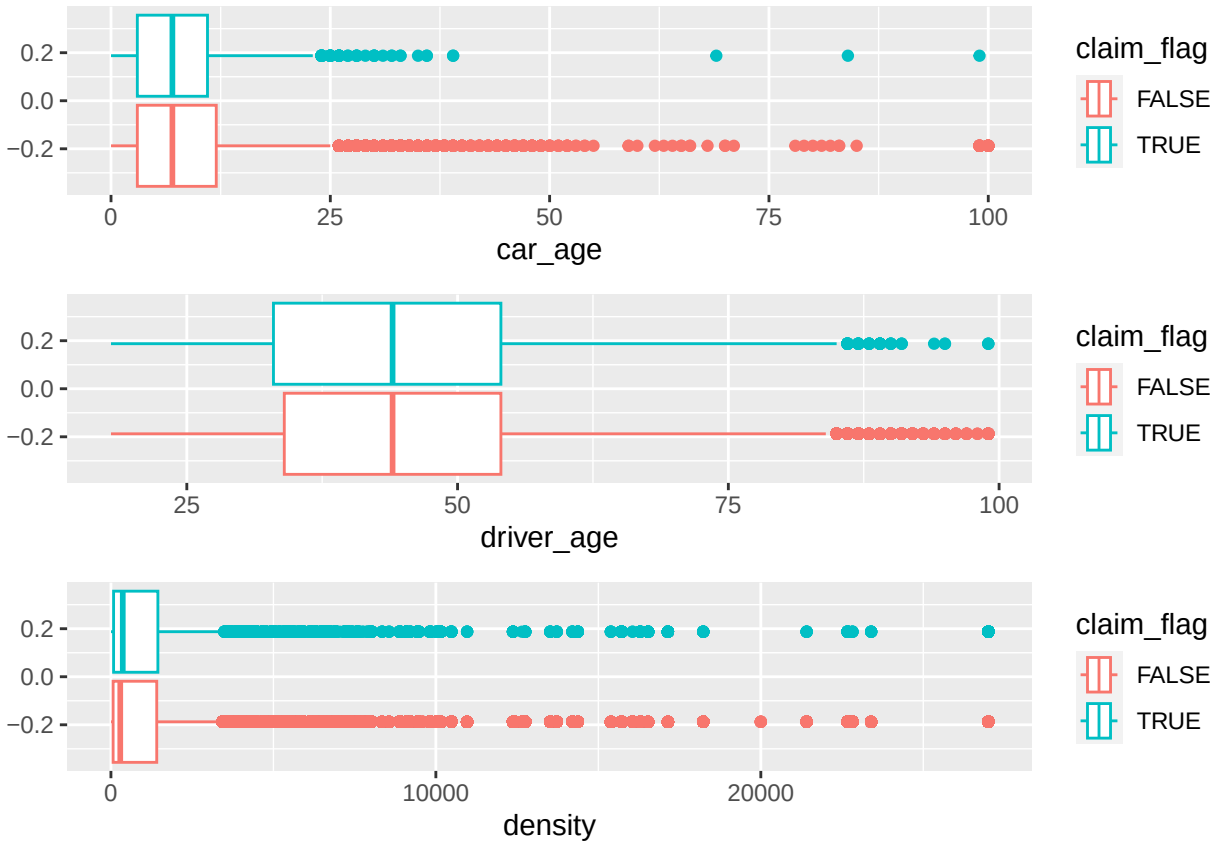
The behavior between claimers and non-claimers are similar, which can be seen from the mean. It seems the number of claims made decreases as the car gets aged, however, considering the fact that no claim is made in majority of our case, it is reasonable that this is only because of the magnitude.

```
summary(motor)
```

```
##  claim_flag          power          car_age          driver_age
##  Mode :logical   f      :95718   Min.   :  0.000   Min.   :18.00
##  FALSE:397779    g      :91198   1st Qu.:  3.000   1st Qu.:34.00
##  TRUE :15390     e      :77022   Median :  7.000   Median :44.00
##                  d      :68014   Mean   :  7.532   Mean   :45.32
##                  h      :26698   3rd Qu.: 12.000   3rd Qu.:54.00
##                  j      :18038   Max.   :100.000   Max.   :99.00
##                  (Other):36481
##                                  brand            gas
##  Fiat                               : 16723   Diesel :205945
##  Japanese (except Nissan) or Korean: 79060   Regular:207224
##  Mercedes, Chrysler or BMW          : 19280
##  Opel, General Motors or Ford       : 37402
##  other                              :  9866
##  Renault, Nissan or Citroen         :218200
##  Volkswagen, Audi, Skoda or Seat    : 32638
##                 region          density
##  Centre             :160601   Min.   :    2
##  Ile-de-France      : 69791   1st Qu.:   67
##  Bretagne           : 42122   Median :  287
##  Pays-de-la-Loire   : 38751   Mean   : 1985
##  Aquitaine          : 31329   3rd Qu.: 1410
##  Nord-Pas-de-Calais: 27285   Max.   :27000
##  (Other)            : 43290
```

The summary statistics suggest all the continuous variables contain outliers as many points lie above the Q3 + 1.5*IQR edge. Also, it shows that the target suffers from imbalanced class. Later, a sampling technique (oversampling) and probability calibration should be introduced for logistics regression.

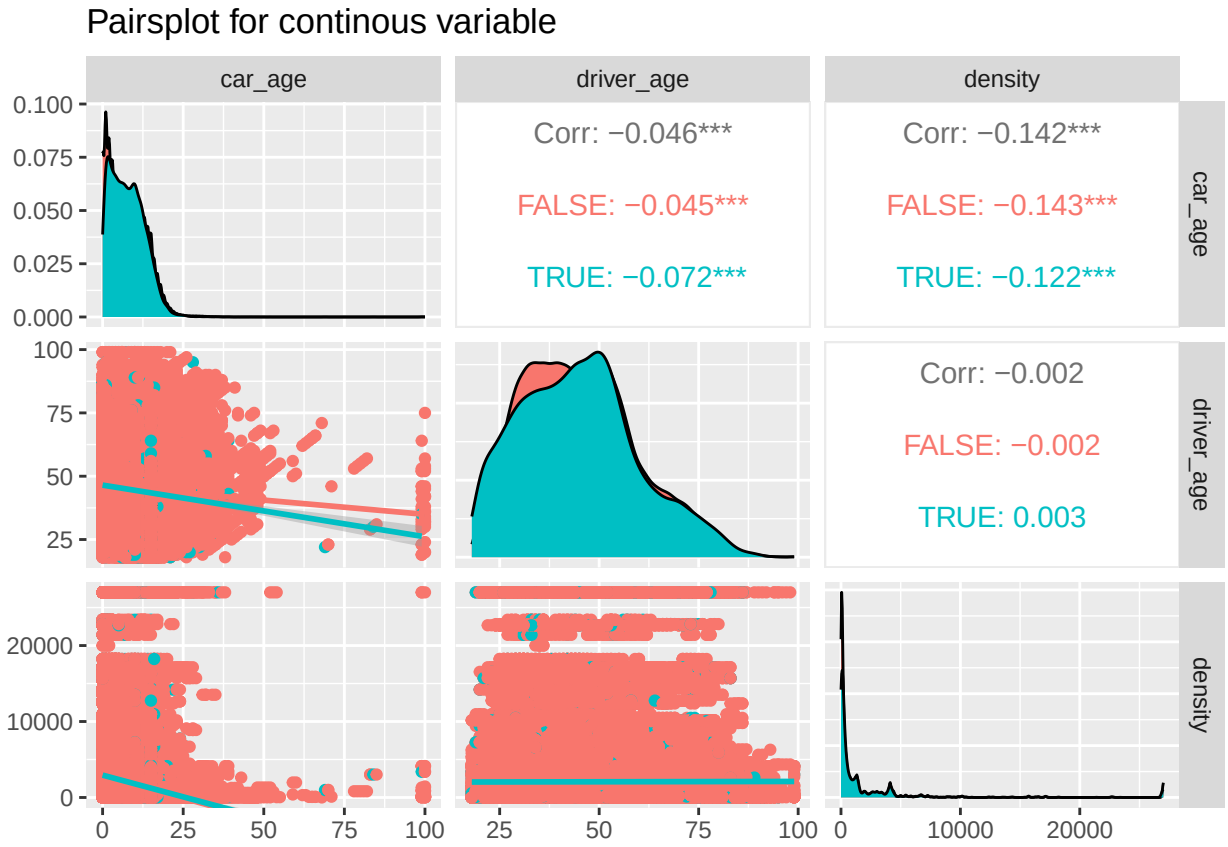
```
p1<-ggplot(motor, aes(x=car_age, color=claim_flag)) +geom_boxplot()
p2<-ggplot(motor, aes(x=driver_age, color=claim_flag)) +geom_boxplot()
p3<-ggplot(motor, aes(x=density, color=claim_flag)) +geom_boxplot()
ggarrange(p1, p2,p3,
          ncol = 1, nrow = 3)
```



Data Exploration

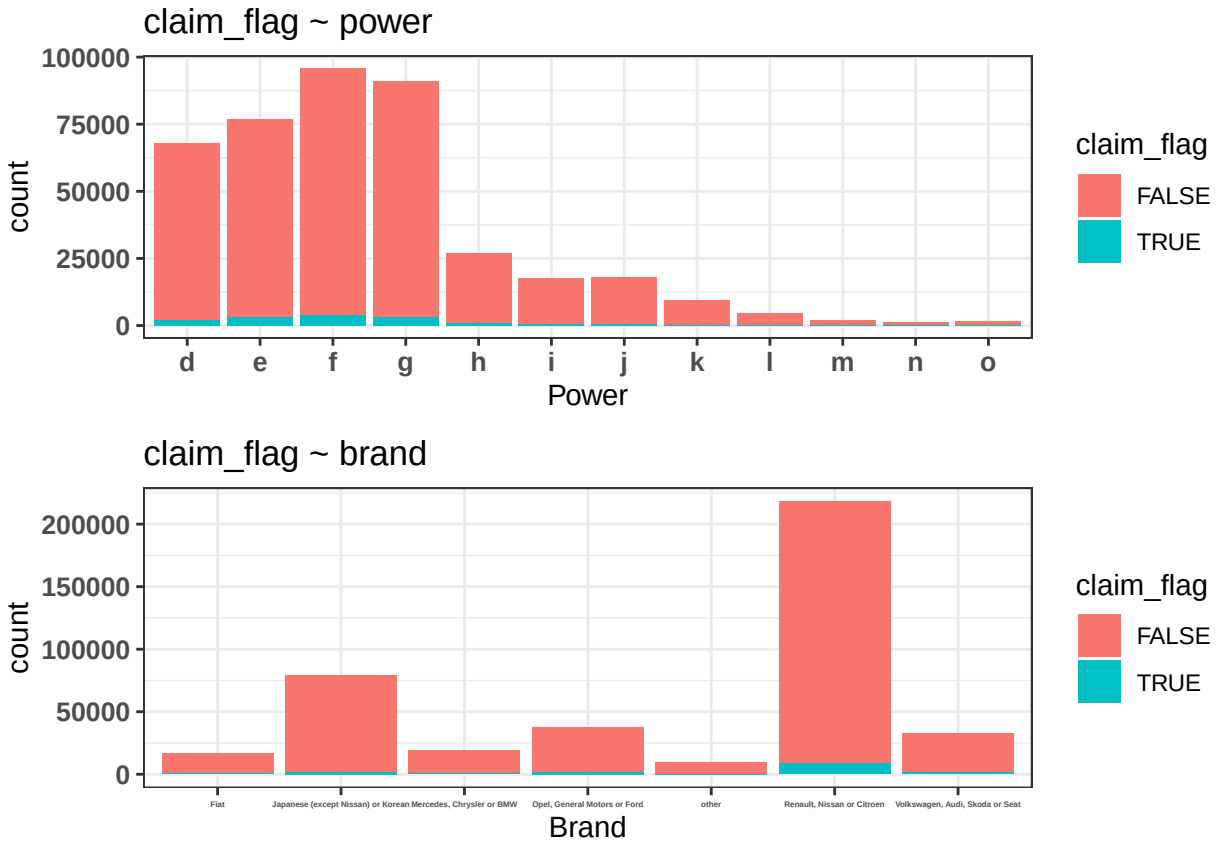
With regard to continuous variables, the pairplot agrees with previous analysis that the claimers and non-claimers groups are similar in distribution. The correlations between our continuous variables are very low, showing a creditable and safe signal to pass to our model.

```
ggpairs(motor,columns = c(3:4,8),
        ggplot2::aes(colour=claim_flag),
        upper = list(combo = "box_no_facet"),
        lower = list(continuous = "smooth"),title = "Pairsplot for continous variable"
        )
```



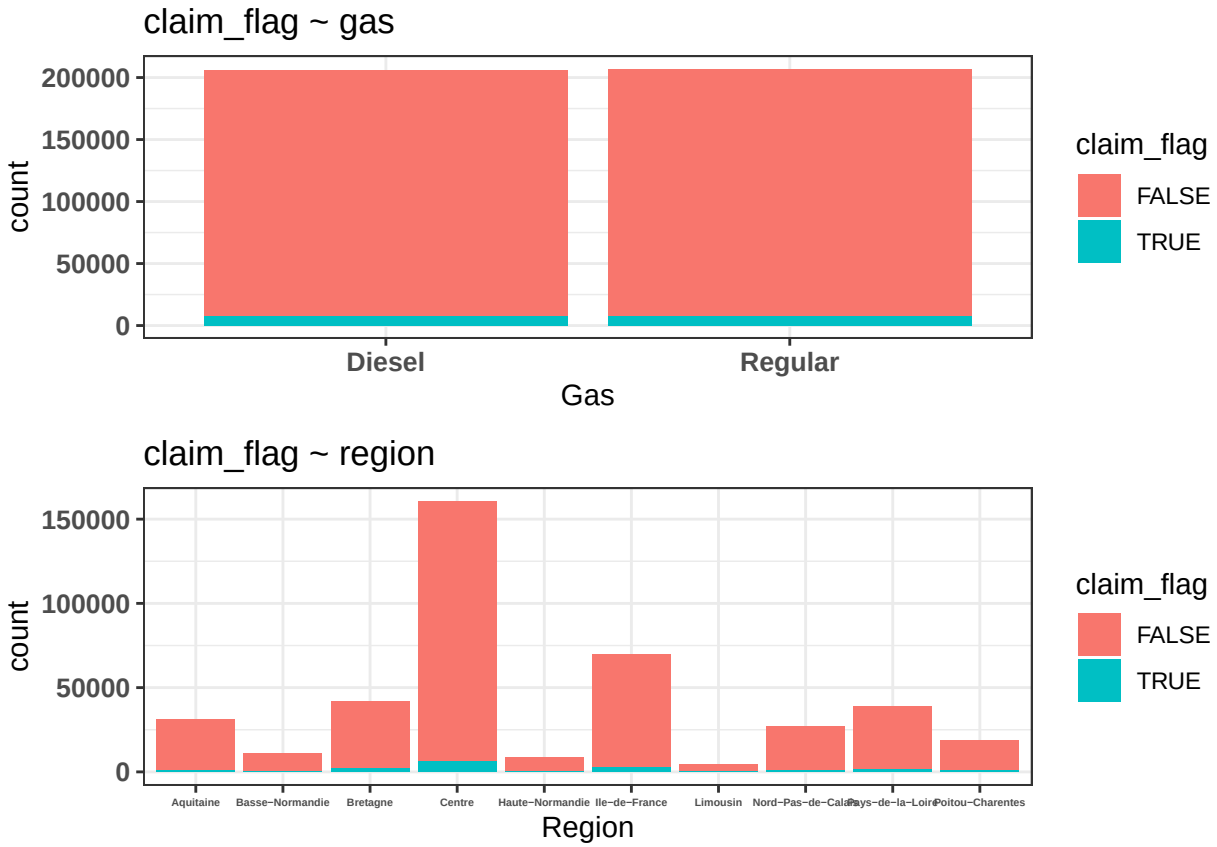
In respect of discrete variables, a number of power types have higher number of claim-making, such as type "g", "e", "f" and "g" as well as brands namely "Renault, Nissan or Citroen". However, those power types and brands are out-numbered others. It is hard to tell the effect clearly before modelling.

```
p4<-ggplot(motor) +
  geom_bar(aes(x = power, fill = claim_flag)) +
  labs(x = 'Power') +
  ggtitle("claim_flag ~ power ") +
  theme_bw() +
  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10)
  )

p5<-ggplot(motor) +
  geom_bar(aes(x = brand, fill = claim_flag)) +
  labs(x = 'Brand') +
  ggtitle("claim_flag ~ brand ") +
  theme_bw() +
  theme(axis.text.x = element_text(face = 'bold', size = 3),
        axis.text.y = element_text(face = 'bold', size = 10)
  )
ggarrange(p4, p5,
          ncol = 1, nrow = 2)
```



The region of the car has similar pattern while the gas type seems no such difference. A number of location have higher number of claim-making, such as "center" and "lle-de France".

```
p6<-ggplot(motor) +
  geom_bar(aes(x = gas, fill = claim_flag)) +
  labs(x = 'Gas') +
  ggtitle("claim_flag ~ gas ") +
  theme_bw() +
  theme(axis.text.x = element_text(face = 'bold', size = 10),
        axis.text.y = element_text(face = 'bold', size = 10)
  )

p7<-ggplot(motor) +
  geom_bar(aes(x = region, fill = claim_flag)) +
  labs(x = 'Region') +
  ggtitle("claim_flag ~ region ") +
  theme_bw() +
  theme(axis.text.x = element_text(face = 'bold', size = 4
                                   ),
        axis.text.y = element_text(face = 'bold', size = 10)
  )
ggarrange(p6, p7,
          ncol = 1, nrow = 2)
```



Model Building and Tuning

The data is pre-processed by test-train split.

```
index <- createDataPartition(motor$claim_flag, p = .9,
                                  list = FALSE,
                                  times = 1)
test<-motor[-index,]
train<-motor[index,]
```

Concerning interpretability and predictability, a lasso type generalized linear regression model is firstly proposed and trained on the available data. The predictor variables can be analyzed to determine their effect on the likelihood of making a claim by looking at the regression coefficients. The loss function is shown below.

$$argmin\frac{1}{2}|logit(E(y)) - X\beta|^2 + \lambda||\beta||_1$$

However, the imbalanced data make the model suffer from heavily biased prediction even after oversampling and tuning by probability calibration. Therefore, we will consider tree model and rely on partial dependence plot to find the relationship between each predictor and target.

Considering a large data set and limited training resource we have, we will use the ranger which is a fast C++ based random forest implementation. For tuning, a random search method of 5-fold parallel cross validation is applied to shorten the training time. As we have a imbalanced class classification problem,

the oversampling is performed in pre-processing the data. The ROC metrics is used to access the model goodness of fit. The approximated training time is 5 hours.

```
control <- trainControl(method='cv',
                        number=5,
                        summaryFunction = twoClassSummary,
                        allowParallel = TRUE,
                        classProbs = TRUE,
                        #verboseIter = TRUE,
                        search = "random",
                        sampling = "up")

#Metric compare model is ROC
set.seed(123)
rf.up <- train(make.names(claim_flag)~.,
                data=train,
                method='ranger',
                metric="ROC",
                importance = "impurity",
                #verbose  = TRUE,
                #tuneGrid=tunegrid,
                trControl=control)
```

The model with 18 mtry, extratrees splitting rule and 14 minimum node size (the default value is 1 in classifications) gives us the best prediction.

```
rf.up$bestTune
```

```
##   mtry splitrule min.node.size
## 2   18      gini            14
```

Model Evaluation and Diagnostics

The tuning process confirms our choice of parameters. The best set of tuning parameters yields a ROC of 0.5654564, which is the best under our random search regime.

```
rf.up$results
```

```
##   min.node.size mtry  splitrule       ROC      Sens       Spec       ROCSD
## 3            19   10       gini 0.5531255 0.9680505 0.04172957 0.008658761
## 2            14   18       gini 0.5651092 0.9782180 0.02714575 0.009218971
## 1             3   22 extratrees 0.5512500 0.9738242 0.03573752 0.006576389
##         SensSD      SpecSD
## 3 0.0006590236 0.004226510
## 2 0.0005973984 0.002988222
## 1 0.0007398282 0.001053628
```

The confusion matrix on the training set performs nicely, where the recall is 1. However, such a high recall may indicate overfitting in our random forest.

```
pred.rf.train <- predict(rf.up, newdata = train)
conMatrix.train<-table(pred = as.factor(pred.rf.train), true = as.factor(train$claim_flag))
conMatrix.train
```

```
##         true
## pred      FALSE   TRUE
##   FALSE. 351940      0
##   TRUE.    6062  13851
```

However, on test data, the ratio of true positive and false positive rates, also named as AUC, tends to underestimate the true positive cases, where can be seen in that only 50 cases of true claims are predicted in our test cases. The recall and precision for this model on test data are both very low, a signal to further improvement. This shall be considered in the decision-making process.

```
pred.rf.test <- predict(rf.up, newdata = test)
conMatrix.test<-table(pred = as.factor(pred.rf.test), true = as.factor(test$claim_flag))
conMatrix.test
```

```
##         true
## pred     FALSE  TRUE
##   FALSE. 38750  1491
##   TRUE.   1027    48
```

We can calculate the error rate of our model, which is 1.648232% on the train set and 6.079969% on our test set.

```
(conMatrix.train[1,2]+conMatrix.train[2,1])/nrow(train)
```

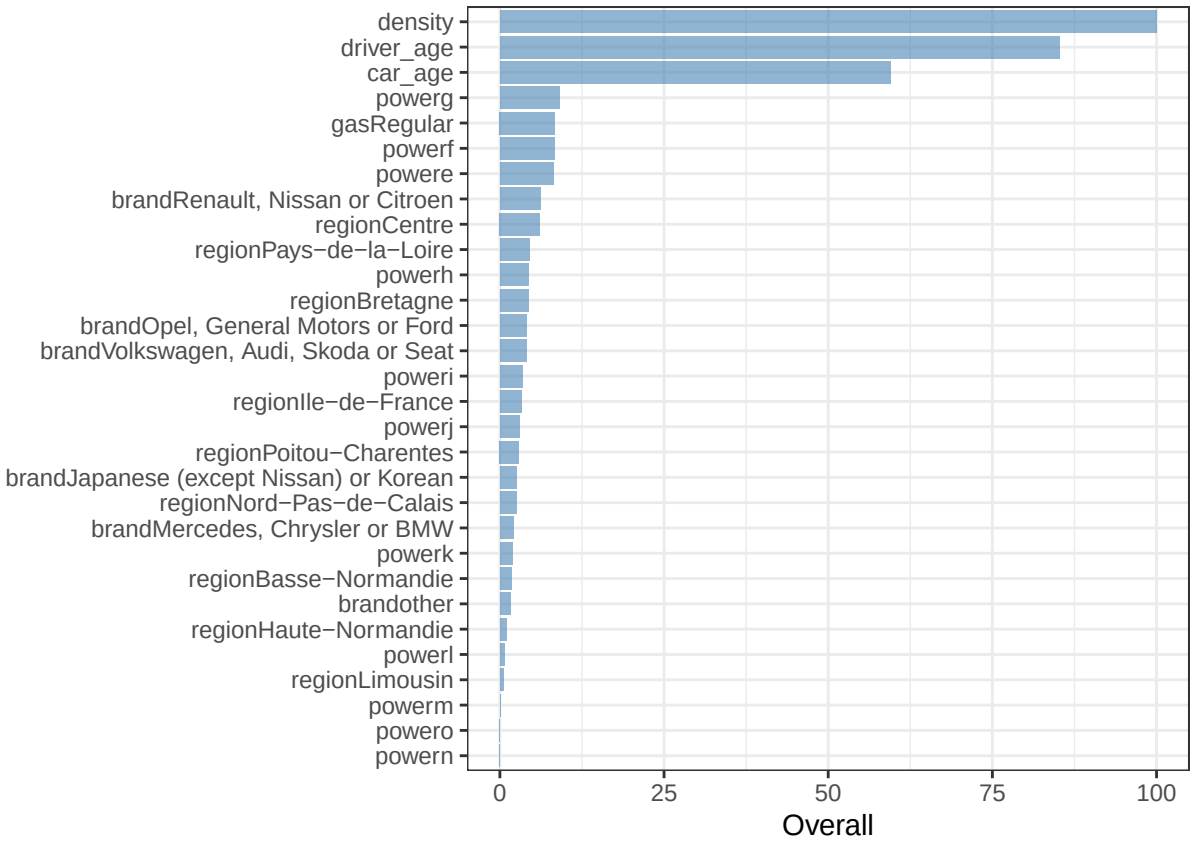
```
## [1] 0.01630214
```

```
(conMatrix.test[1,2]+conMatrix.test[2,1])/nrow(test)
```

```
## [1] 0.06094491
```

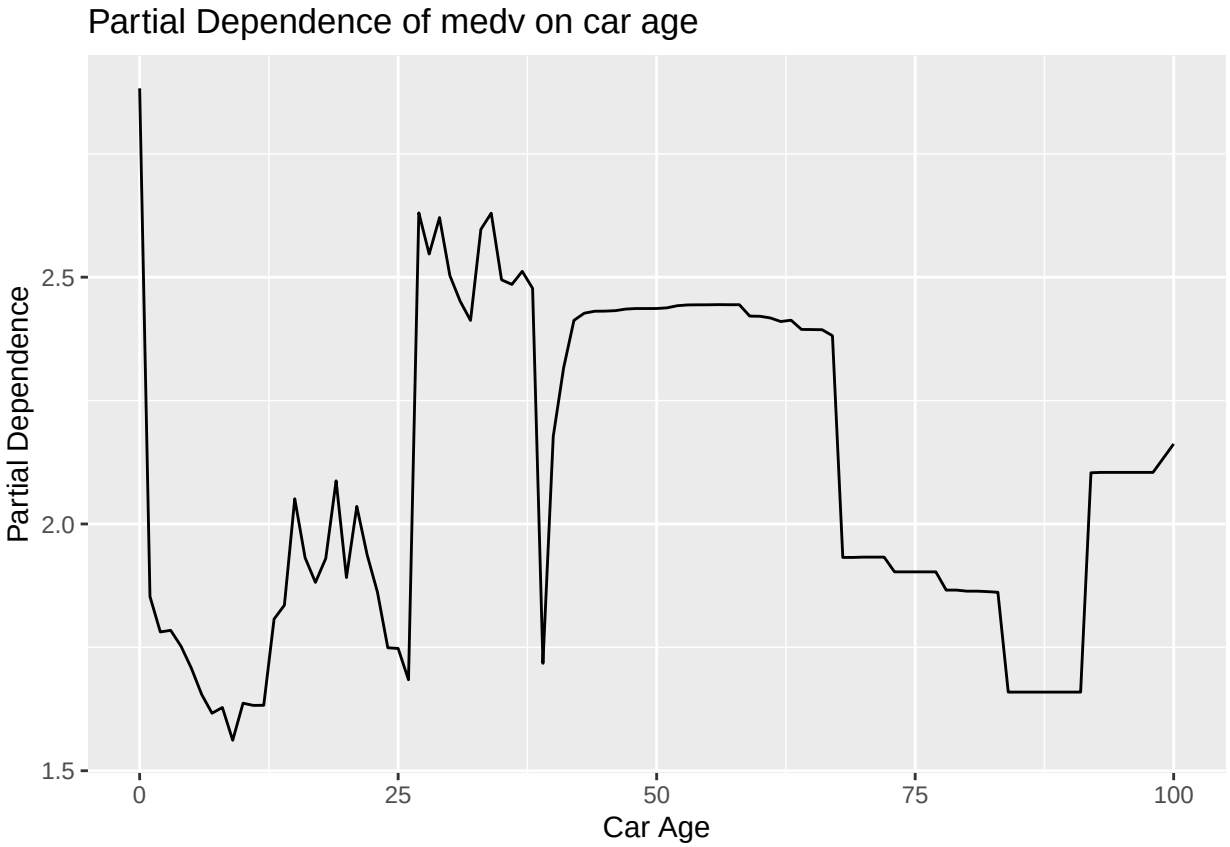
From the variable importance plot, the three continuous variables play a vital roles, ranking the top three important factors followed by gas, power, region and brand.

```
VI <- varImp(rf.up)
name<-rownames(varImp(rf.up)$importance)
df.VI<-cbind(VI$importance,name)
df.VI<-df.VI[
  with(df.VI, order(Overall,decreasing = TRUE)),
]
# visualize variable importance with horizontal bar plot
df.VI %>%
  mutate(name = fct_reorder(name, Overall)) %>%
  ggplot( aes(x=name, y=Overall)) +
    geom_bar(stat="identity", fill="steelblue", alpha=.6) +
    coord_flip() +
    xlab("") +
    theme_bw()
```



Let's look at the partial dependence plot next. The partial dependence plot for car's age indicates that there exists a complex curvature characteristics between the claim and car age, showing below in the chart.

```
rm.pdp <- partial(rf.up, pred.var = "car_age", grid.resolution = 100)
rm.df <- as.data.frame(rm.pdp)
ggplot(rm.df, aes(x = car_age, y = yhat)) +
  geom_line() +
  labs(title = "Partial Dependence of medv on car age", x = "Car Age", y = "Partial Dependence")
```

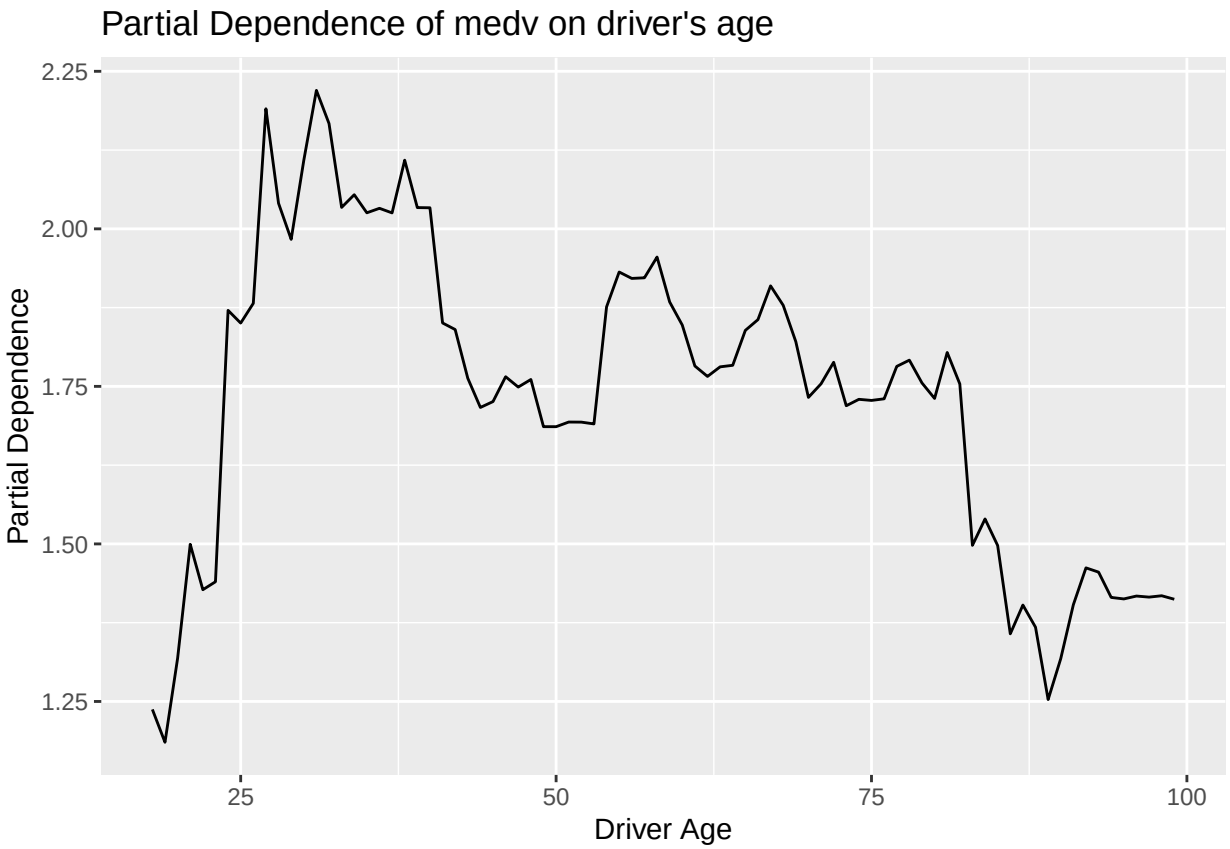


The driver's age is as expected to have a increase and then decrease curvature overall, confirming our previous guess.

```
rm.pdp1 <- partial(rf.up, pred.var = "driver_age", grid.resolution = 99)
```

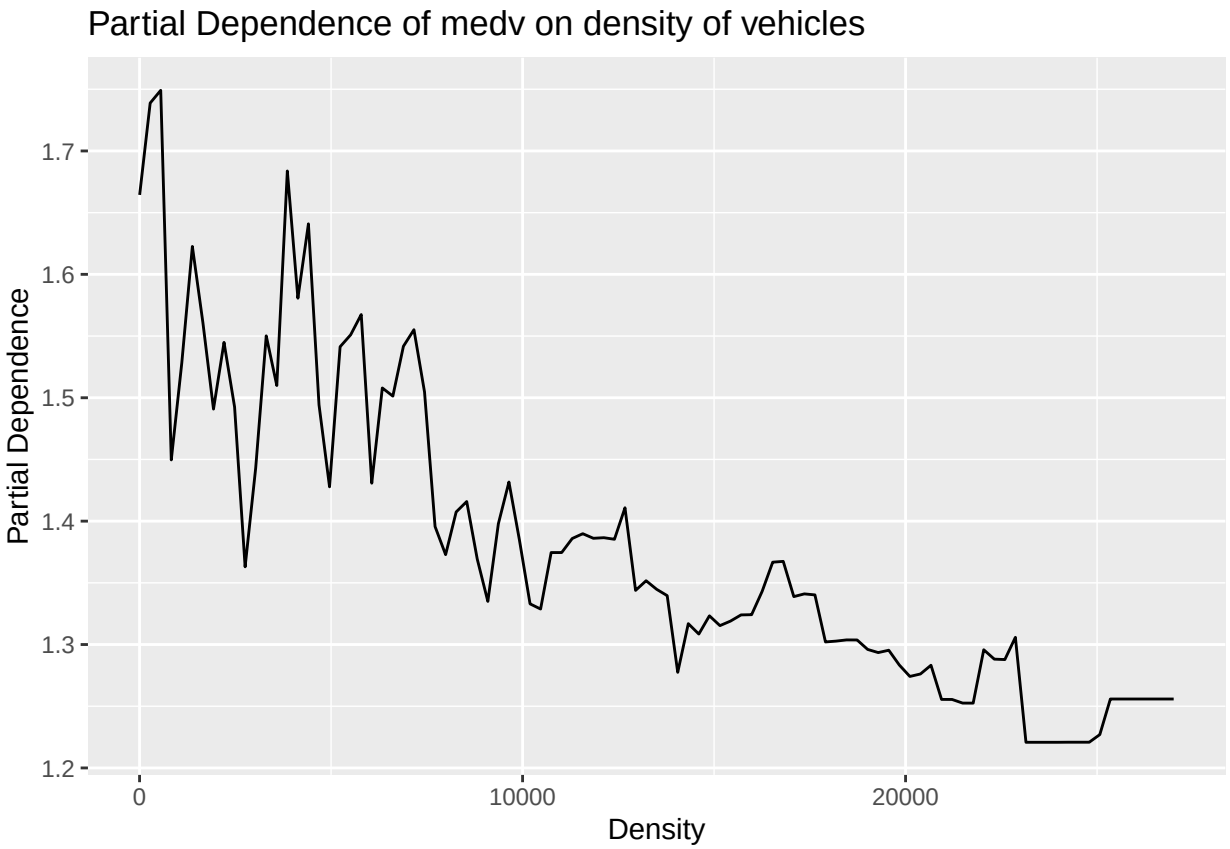
```
## Aggregating predictions.. Progress: 40%. Estimated remaining time: 25 minutes, 33 seconds.
## Predicting.. Progress: 19%. Estimated remaining time: 1 hour, 6 minutes, 57 seconds.
```

```
rm.df1 <- as.data.frame(rm.pdp1)
ggplot(rm.df1, aes(x = driver_age, y = yhat)) +
  geom_line() +
  labs(title = "Partial Dependence of medv on driver's age", x = "Driver Age", y = "Partial Dependence")
```



The density of vehicles where policy holders live have a negative effect on the claim, a factor that may need further investigation.

```
rm.pdp2 <- partial(rf.up, pred.var = "density", grid.resolution = 99)
rm.df2 <- as.data.frame(rm.pdp2)
ggplot(rm.df2, aes(x = density, y = yhat)) +
  geom_line() +
  labs(title = "Partial Dependence of medv on density of vehicles", x = "Density", y = "Partial Depender
```



The effect of discrete variables can be shown similarly by the example above.

Conclusion

All of our 7 predictor is vital in our model, however, the relationship between the response and predictors varies. From our model, it is clear that each power, region, brand and gas type is related to our claim-making, however, each level may vary in terms of the negative and positive effect.

The continuous variables driver age and car age have quadratic effects, while the effect of density is monotone. It is reasonable to suspect that older cars may be rented less as negative relation between car's age and claim-making is counter-intuitive.

In conclusion, when considers the likelihood of the claim-making, analysts should consider all the variable, but some level have little effect may be less important in the decision-making process.