CS Crystal Report

Milton Candela

3/6/2022

Packages used:

- caret
- randomForest

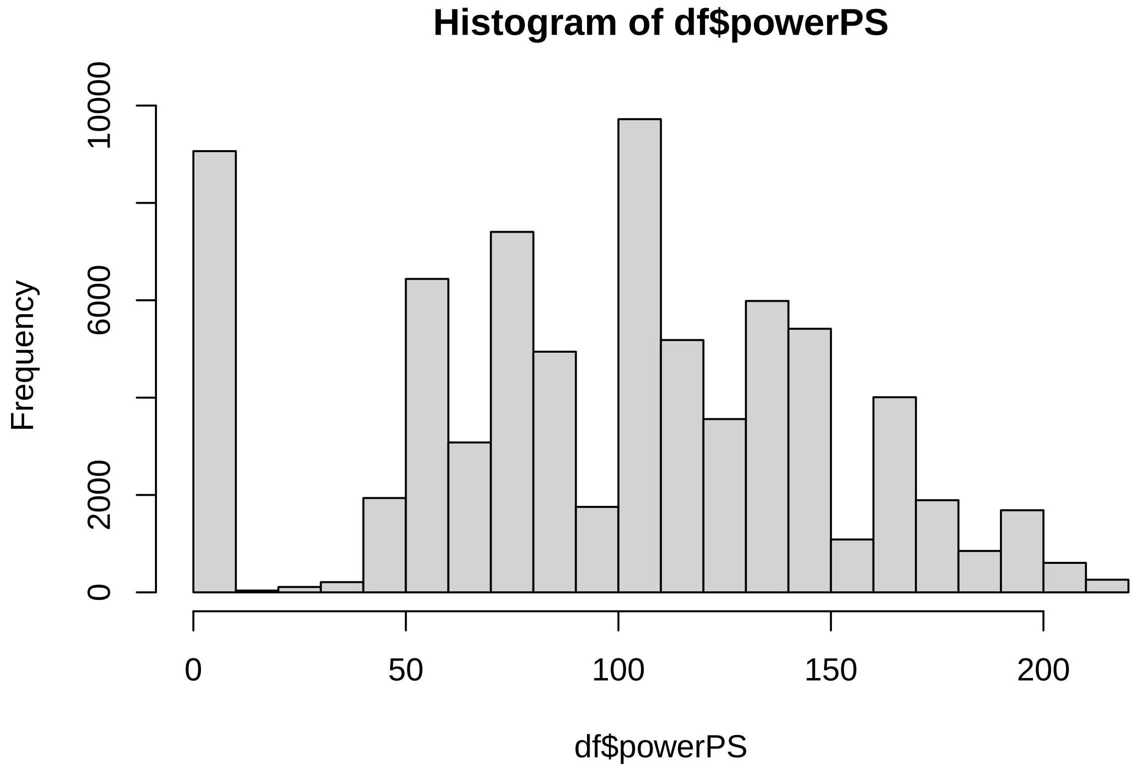
Data Preparation

```
df <- read.csv('Automotive_2.csv', colClasses = c('character', 'character', 'factor', 'factor', 'charac
                                                  'factor', 'factor', 'character', 'factor', 'character
                                                  'factor', 'numeric', 'character', 'factor', 'factor',
                                                  'factor', 'character', 'integer', 'character', 'chara
# df <- na.omit(df)

# (dateCrawled) Data is not relevant, should be removed.
df <- df[, -grep('dateCrawled', colnames(df))]

# (name) Information could be extracted, but would be removed.
df <- df[, -grep('name', colnames(df))]

# (seller) Factor variable, 2 levels but 99% of samples are "privat", so column should be removed.
df <- df[, -grep('seller', colnames(df))]

# (offerType) Remove numeric values, 99% of samples are "Angebot", so column should be removed.
df <- df[, -grep('offerType', colnames(df))]

# (price) Outliers are considered 95%, so they should be removed.
df[,'price'] <- as.integer(df$price)
```

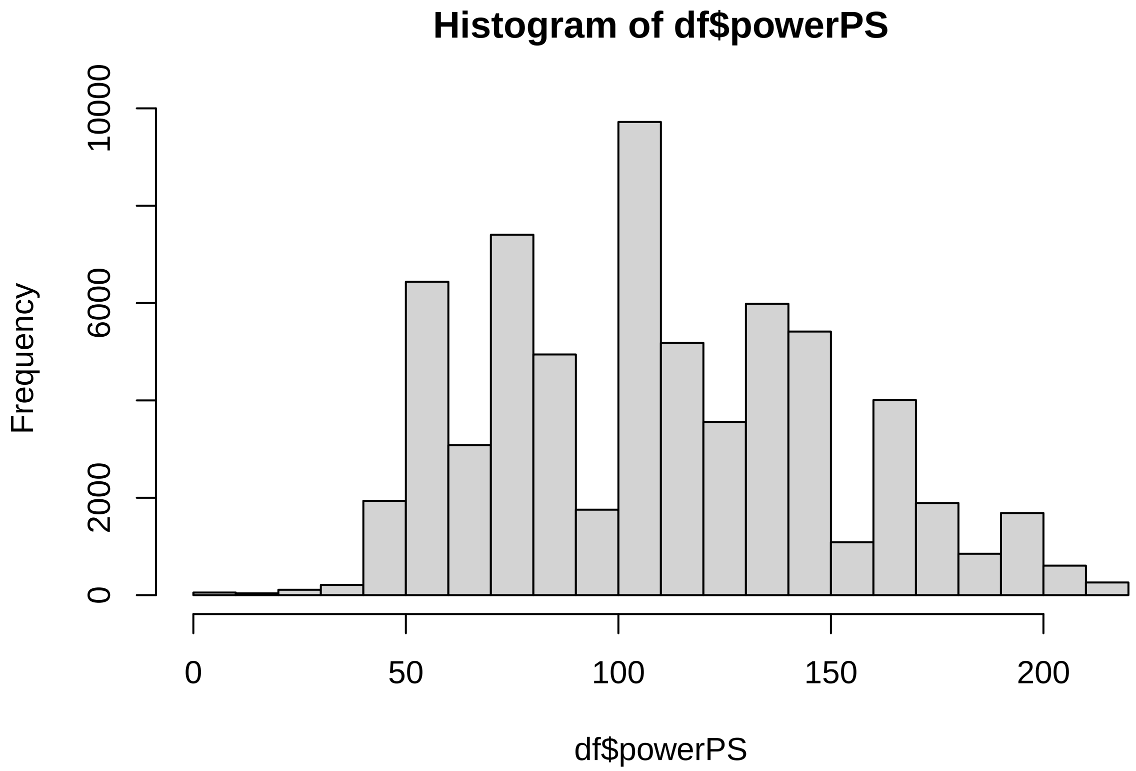
```
## Warning: NAs introduced by coercion
```

```
# A series of NAs are on every column
df <- df[!is.na(df$price), ]
df <- df[df$price < quantile(df$price, 0.95, na.rm = TRUE),]

# (abtest) Removing levels with 0 samples.
df[, 'abtest'] <- droplevels(df$abtest)

# (vehicleType) Removing levels with 0 samples.
# 8422 are blank samples (10.61 % of data).
```

```
df[, ’vehicleType’] <- droplevels(df$vehicleType)

# (yearOfRegistration) Remove characters, blank spaces and outliers.
df[,’yearOfRegistration’] <- as.integer(df$yearOfRegistration)
df <- df[(df$yearOfRegistration > 1900) & (df$yearOfRegistration < 2022),]

# (gearbox) Removing levels with 0 samples.
# 4519 are blank samples (5.69 % of data)
df[, ’gearbox’] <- droplevels(df$gearbox)

# (powerPs) Converting to numeric, maybe 0 was considered as a NA value as for the histogram.
df[, ’powerPS’] <- as.integer(df$powerPS)
df <- df[df$powerPS < quantile(df$powerPS, 0.95),]
hist(df$powerPS)
```



```
# Looking at the histogram, 0s might be NAs, so we then remove 0s
df <- df[df$powerPS != 0, ]
hist(df$powerPS)
```



```
# Now that histogram looks better! Now blank samples from "gearbox" passed from 4519 to 1454 and "vehic

# (model) Too many levels, may not be useful.
summary(df$model)
```

```
##         golf      andere         3er                    polo       corsa
##         5991        4360        3744        2763        2529        2461
##        astra      passat          a4    c_klasse          a3         5er
##         2168        2030        1902        1591        1237        1202
##        focus      fiesta    e_klasse     2_reihe transporter      fortwo
##         1179        1147        1111        1021         971         930
##           a6      twingo      vectra    a_klasse         1er      mondeo
##          865         865         848         792         720         709
##       touran     3_reihe        clio       punto      zafira      megane
##          706         691         657         652         607         536
##        ibiza        lupo          ka       fabia     octavia      cooper
##          517         510         498         440         418         393
##        micra       caddy          80      sharan      laguna     1_reihe
##          355         324         307         301         286         285
##          clk      scenic       omega     6_reihe     i_reihe       civic
##          285         284         283         272         270         256
##       galaxy        leon       yaris      meriva         slk    mx_reihe
##          239         235         226         219         215         202
##         bora      escort         one        colt         v40        vito
##          187         185         182         180         179         178
##     b_klasse      beetle      kangoo         500    sprinter          tt
##          177         174         169         166         157         157
##      x_reihe       arosa         fox       tigra     transit    berlingo
##          155         154         154         154         151         139
##       tiguan       panda         v70         147          a1     corolla
##          137         135         133         131         131         130
##        swift     4_reihe    seicento    scirocco       c_max     qashqai
##          130         125         124         121         119         116
##      avensis       stilo     z_reihe     primera      espace          c3
##          115         115         112         111         110         109
##          156       s_max      almera         eos    insignia          c1
##          106         106         105         104         103         102
##           c5      signum       grand     (Other)
##          101          96          94        5501
```

```
df <- df[, -grep('model', colnames(df))]

# (kilometer) Histogram looks skewed to the right, may not be useful.
# df <- df[, -grep('kilometer', colnames(df))]

# (monthOfRegistration) Converting to integer, as outliers have been removed.
df[, 'monthOfRegistration'] <- as.integer(df$monthOfRegistration)

# (fuelType) Main classes are benizin and diesel, others could be removed.
df[, 'fuelType'] <- droplevels(df$fuelType)

# (brand) Too many levels, may not be useful.
df[, 'brand'] <- droplevels(df$brand)
# <- df[, -grep('brand', colnames(df))]

# (notRepairedDamage) Removing levels with 0 samples (dates).
# 10610 blank samples (16.02 % of data)
df[, 'notRepairedDamage'] <- droplevels(df$notRepairedDamage)

# (dateCreated) Not useful, should be removed.
df <- df[, -grep('dateCreated', colnames(df))]

# (nrOfPictures) All values are 0, should be removed.
df <- df[, -grep('nrOfPictures', colnames(df))]

# (postalCode) Not useful, should be removed.
df <- df[, -grep('postalCode', colnames(df))]

# (lastSeen) Used for our classification model (if NAs, then car was not sold yet)
encoding_class <- function(x){
     if (is.na(x)){
          0
     } else {
          1
     }
}
df[, 'sold'] <- sapply(as.Date(df$lastSeen), FUN = encoding_class)
df <- df[, -grep('lastSeen', colnames(df))]

df2 <- df
```

```
y <- df2[, 'sold']
df2 <- df2[, -grep('sold', colnames(df2))]
```

Filling missing values

```
set.seed(1002)
missing_cat_cols <- c('gearbox', 'vehicleType', 'notRepairedDamage', 'fuelType')
full_cols <- colnames(df2)[!colnames(df2) %in% missing_cat_cols]

fill_missing <- function(df2, col_name){
     print(col_name)
     print(summary(df2[, col_name]))

     train_data <- df2[df2[,col_name]!='',c(full_cols, col_name)]
     train_data[, col_name] <- droplevels(train_data[, col_name])
     names(train_data)[names(train_data) == col_name] <- 'Class'
     #train_data <- downSample(x = train_data[, -grep(col_name, colnames(train_data))], y = train_data[

     print(summary(train_data[, 'Class']))

     predict_data <- df2[df2[,col_name]=='',full_cols]
     model <- train(Class ~ ., data = train_data, method = 'rpart')

     df2[df2[, col_name]=='', col_name] <- predict(model, predict_data, type = 'raw')
     df2[, col_name] <- droplevels(df2[, col_name])

     print(summary(df2[, col_name]))

     df2
}

for (missing_col in missing_cat_cols){
     df2 <- fill_missing(df2, missing_col)
}
```

```
## [1] "gearbox"
##           automatik    manual
##      1454     10707     54064
## automatik    manual
##     10707     54064
## automatik    manual
##     10891     55334
## [1] "vehicleType"
##                 andere        bus     cabrio      coupe kleinwagen      kombi
##       5017        565       5990       3807       2767      16053      12766
##  limousine        suv
##      17288       1972
##     andere        bus     cabrio      coupe kleinwagen      kombi  limousine
##        565       5990       3807       2767      16053      12766      17288
##        suv
##       1972
##     andere        bus     cabrio      coupe kleinwagen      kombi  limousine
##        565       5990       3807       2767      18094      14714      18316
##        suv
##       1972
## [1] "notRepairedDamage"
##           no   yes
## 10610 48815  6800
##    no   yes
## 48815  6800
##    no   yes
## 58414  7811
## [1] "fuelType"
##           andere  benzin     cng  diesel elektro  hybrid     lpg
##    4497      27   41786     101   18815      14      51     934
##  andere  benzin     cng  diesel elektro  hybrid     lpg
##      27   41786     101   18815      14      51     934
##  andere  benzin     cng  diesel elektro  hybrid     lpg
##      27   44919     101   20179      14      51     934
```

```
df3 <- df2
```

Classification model

```
set.seed(500)

df3 <- downSample(df3, as.factor(y), yname = 'sold')

trainIndex <- createDataPartition(df3$sold, p = 0.70, list = FALSE)
training <- df3[trainIndex,]
testing <- df3[-trainIndex,]

imp <- importance(randomForest(sold ~., data=training))
imp_cols <- names(imp[imp > 1000,])

training <- training[, c(imp_cols, 'sold')]
testing <- testing[, c(imp_cols, 'sold')]

write.csv(training, 'training.csv')
write.csv(testing, 'testing.csv')

set.seed(1000)

model <- train(sold ~ ., data = training, method = 'rpart')
pred <- predict(model, testing, type = 'raw')
print(confusionMatrix(pred, testing$sold))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction    0    1
##          0 5078 4383
##          1 2005 2700
##
##                Accuracy : 0.5491
##                  95% CI : (0.5408, 0.5573)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.0981
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7169
##             Specificity : 0.3812
##          Pos Pred Value : 0.5367
##          Neg Pred Value : 0.5739
##              Prevalence : 0.5000
##          Detection Rate : 0.3585
##    Detection Prevalence : 0.6679
##       Balanced Accuracy : 0.5491
##
##        'Positive' Class : 0
##
```