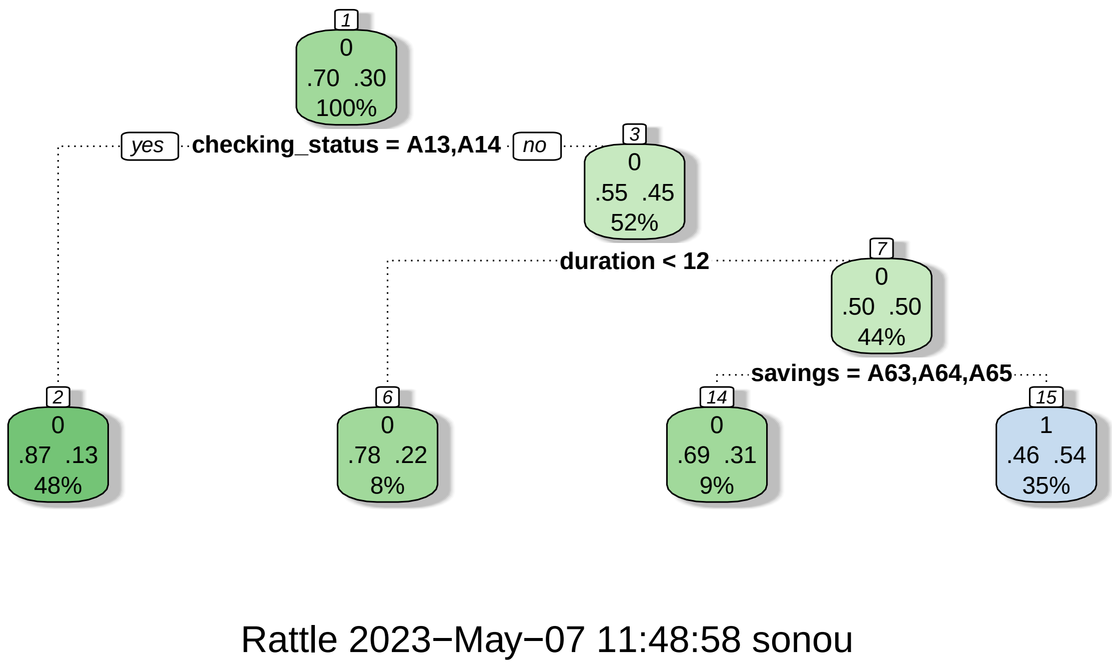
Week 6 - AYU - Pod

Contents

1. Classification Tree

```
library(CASdatasets)
library(tidyverse)
library(caret)
data(credit)
df <- credit
df <- df %>% rename(target=class)

df <- df %>%
  mutate(target = as.factor(target))


library(caret)
set.seed(2020)
splitIndex <- createDataPartition(df$target, p = .70,
                                  list = FALSE)
df_train <- df[ splitIndex,]
```

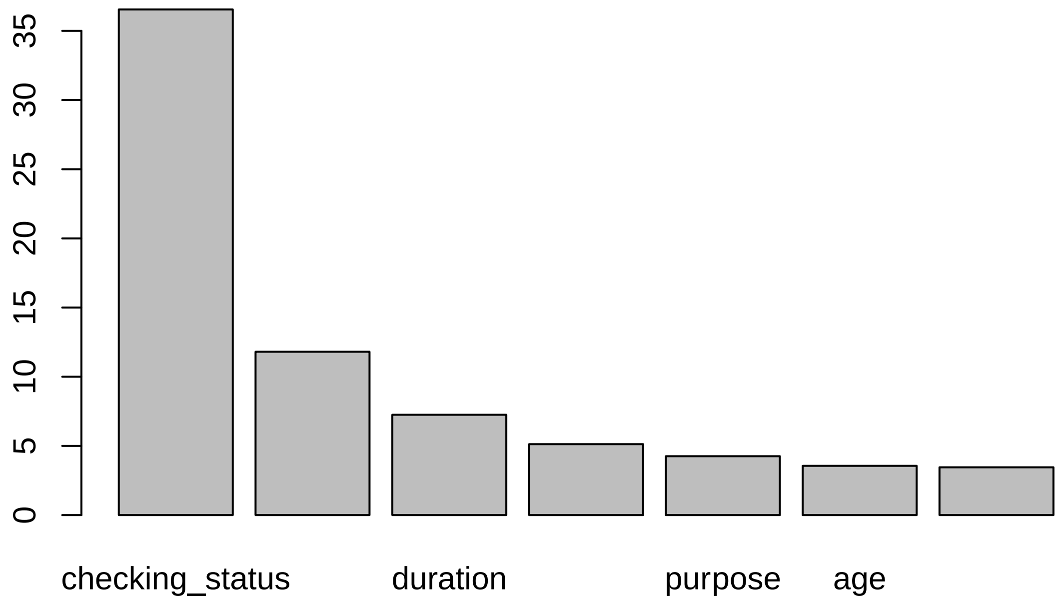
```r
df_test <- df[-splitIndex,]

library(rpart) #load the rpart package
# Create a tree
tree_model <- rpart(target ~ ., data = df_train,
                    control = rpart.control(maxdepth = 3))
library(rattle)
fancyRpartPlot(tree_model)
```




```r
tree_model$variable.importance
```

```
## checking_status         savings        duration  credit_history         purpose
##       36.549990       11.805346        7.251034        5.127909        4.255073
##             age   credit_amount
##        3.559540        3.454994
```

```r
barplot(tree_model$variable.importance)
```




```r
pred <- predict(tree_model, df_test, type = "class")
#Evaluate the predictions
cm <- confusionMatrix(data = pred, reference = df_test$target, positive = "1")
cm$overall[1]
```

```
## Accuracy
##     0.71
```

Question: We will work with the Actuarial Loss dataset. The data dictionary is as follows.

ClaimNumber: Unique policy identifier DateTimeOfAccident: Date and time of accident DateReported: Date that accident was reported Age: Age of worker Gender: Gender of worker MaritalStatus: Martial status of worker. (M)arried, (S)ingle, (U)nknown. DependentChildren: The number of dependent children DependentsOther: The number of dependants excluding children WeeklyWages: Total weekly wage PartTimeFullTime: Binary (P) or (F) HoursWorkedPerWeek: Total hours worked per week DaysWorkedPerWeek: Number of days worked per week ClaimDescription: Free text description of the claim InitialIncurredClaimCost: Initial estimate by the insurer of the claim cost UltimateIncurredClaimCost: Total claims payments by the insurance company. This is the field you are asked to predict in the test set. Claim_Cost_Category: 1 for claim cost higher than the median cost and 0 otherwise.

- Partition the data into 70% training and 30% testing.
- Create a decision tree with maximum depth of 5 on the training data to predict the claim cost category (i.e., `claim_cost_category` is your target variable).

- Plot the decision tree
- Calculate the accuracy of the decision tree on the test data.
- Plot the bar chart of the variable importance according to the tree.

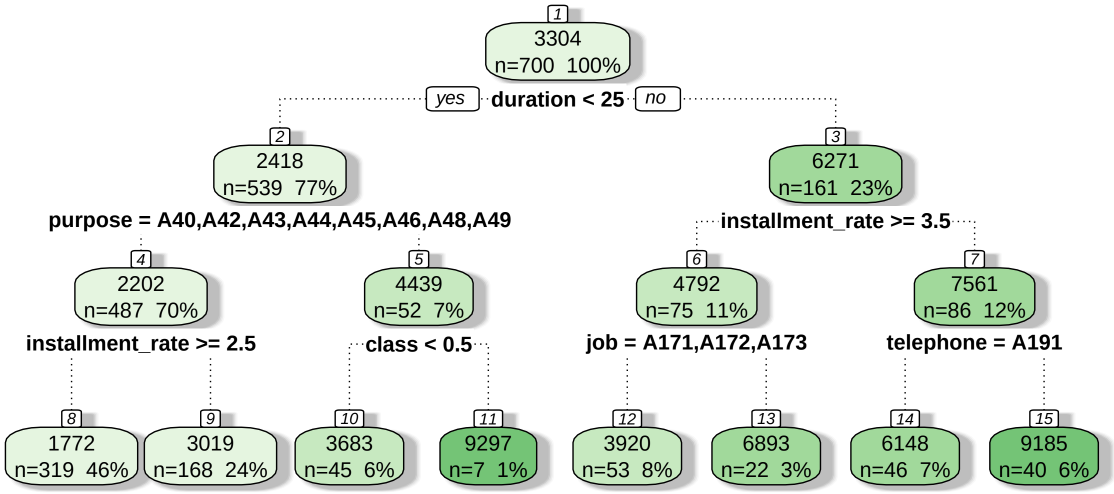
2. Random Forest for classification

Question: Continue work with the same Actuarial Loss dataset

- Train a random forest of 1000 trees and `mtry=5` to predict claim cost category on the training data.
- Calculate the accuracy of the forest on the testing data.

3. Regression Tree

```
library(tidyverse)
library(caret)
df <- read_csv('german_credit.csv')
df <- df %>% rename(target=credit_amount)

library(caret)
set.seed(2020)
splitIndex <- createDataPartition(df$target, p = .70,
                                  list = FALSE)
df_train <- df[ splitIndex,]
df_test <- df[-splitIndex,]

library(rpart) #load the rpart package
# Create a tree
tree_model <- rpart(target ~ ., data = df_train,
                 control = rpart.control(maxdepth = 3))
library(rattle)
fancyRpartPlot(tree_model)
```

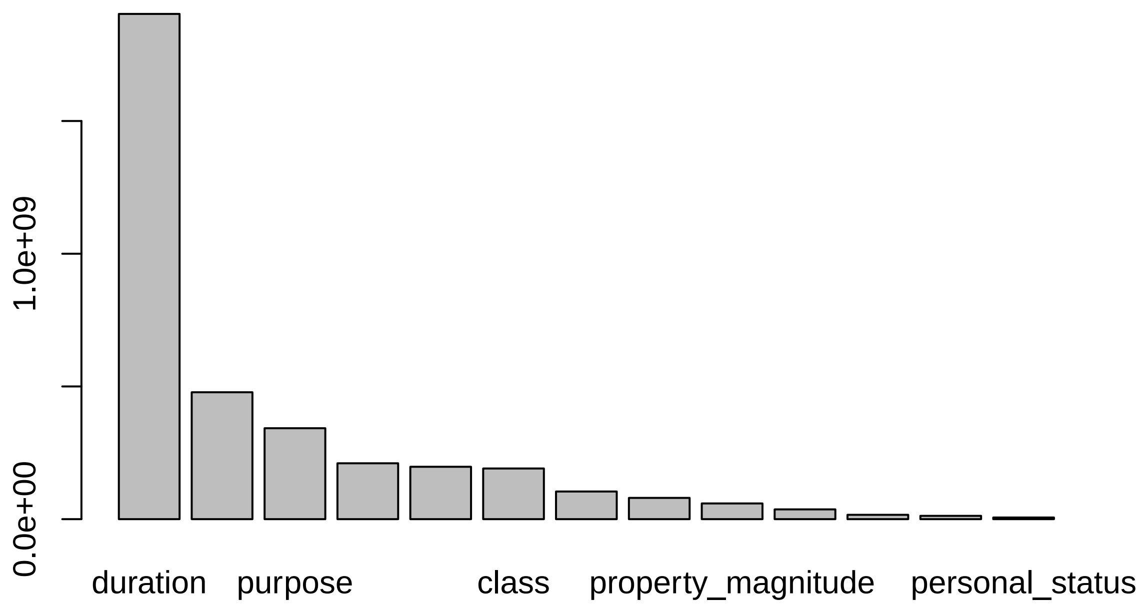


Rattle 2023−May−07 11:48:58 sonou

```
tree_model$variable.importance
```

```
##            duration    installment_rate             purpose                 job
##          1903362509           478063388           342539050           210511875
##           telephone               class                 age          employment
##           197319823           190914218           104201587            80132056
##  property_magnitude     residence_since      credit_history other_payment_plans
##            59195947            36839309            16373026            12488806
##     personal_status
##             6244403
```

```
barplot(tree_model$variable.importance)
```




```
pred1 <- predict(tree_model, df_test)
#Evaluate the predictions
postResample(pred = pred1, obs = df_test$target)
```

```
##         RMSE     Rsquared          MAE
## 2252.9604437    0.3400649 1477.7516971
```

- Create a decision tree with maximum depth of 3 on the training data to predict the ultimate claim cost(i.e., `UltimateIncurredClaimCost` is your target variable).

- Plot the decision tree
- Calculate the RMSE, Rsquared and MAE of the decision tree on the test data.
- Plot the bar chart of the variable importance according to the tree.

4. Random Forest for Regression

```
library(ranger)
forest_model <- ranger(target ~ ., data=df_train, importance='impurity', mtry=3, num.trees = 500,)
pred2 <- predict(forest_model, df_test)
#Evaluate the predictions
postResample(pred = pred2$predictions, obs = df_test$target)
```

```
##         RMSE     Rsquared          MAE
## 1933.4688282    0.5334729 1340.4880132
```

Question: Continue work with the same Actuarial Loss dataset

- Train a random forest of 1000 trees and `mtry=5` to predict the ultimate claim cost on the training data.
- Calculate the RMSE, Rsquared and MAE of the decision tree on the test data.