project375

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## Warning: package 'purrr' was built under R version 4.0.3
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
## Warning: package 'forcats' was built under R version 4.0.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.0.3
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(glmnet) # backward criterion
```

```
## Warning: package 'glmnet' was built under R version 4.0.3

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.0-2
```

```
library(jpeg)   # high-quality plotting
```

```
## Warning: package 'jpeg' was built under R version 4.0.3
```

```
credit <- read_csv("https://docs.google.com/spreadsheets/d/1jFkOKgD5NGeD8mDj_42oBNJfFVK42-1cMKk0JxVFxeA/
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   duration = col_double(),
##   credit_amount = col_double(),
##   installment_commitment = col_double(),
##   residence_since = col_double(),
##   age = col_double(),
##   existing_credits = col_double(),
##   num_dependents = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
glimpse(credit)
```

```
## Rows: 1,000
## Columns: 21
## $ checking_status        <chr> "<0'", "0<=X<200'", "no checking'", "<0'", "...
## $ duration               <dbl> 6, 48, 12, 42, 24, 36, 24, 36, 12, 30, 12, 4...
## $ credit_history         <chr> "critical/other existing credit'", "existing...
## $ purpose                <chr> "radio/tv", "radio/tv", "education", "furnit...
## $ credit_amount          <dbl> 1169, 5951, 2096, 7882, 4870, 9055, 2835, 69...
## $ savings_status         <chr> "no known savings'", "<100'", "<100'", "<100...
## $ employment             <chr> ">=7'", "1<=X<4'", "4<=X<7'", "4<=X<7'", "1<...
## $ installment_commitment <dbl> 4, 2, 2, 2, 3, 2, 3, 2, 2, 4, 3, 3, 1, 4, 2,...
## $ personal_status        <chr> "male single'", "female div/dep/mar'", "male...
## $ other_parties          <chr> "none", "none", "none", "guarantor", "none",...
```

```
## $ residence_since        <dbl> 4, 2, 3, 4, 4, 4, 4, 2, 4, 2, 1, 4, 1, 4, 4,...
## $ property_magnitude     <chr> "real estate'", "real estate'", "real estate...
## $ age                    <dbl> 67, 22, 49, 45, 53, 35, 53, 35, 61, 28, 25, ...
## $ other_payment_plans    <chr> "none", "none", "none", "none", "none", "non...
## $ housing                <chr> "own", "own", "own", "for free'", "for free'...
## $ existing_credits       <dbl> 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1,...
## $ job                    <chr> "skilled", "skilled", "unskilled resident'",...
## $ num_dependents         <dbl> 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ own_telephone          <chr> "yes", "none", "none", "none", "none", "yes"...
## $ foreign_worker         <chr> "yes", "yes", "yes", "yes", "yes", "yes", "y...
## $ class                  <chr> "good", "bad", "good", "good", "bad", "good"...
```

```
credit <-  credit %>% mutate_if(is.character, as.factor)
glimpse(credit)
```

```
## Rows: 1,000
## Columns: 21
## $ checking_status        <fct> <0', 0<=X<200', no checking', <0', <0', no c...
## $ duration               <dbl> 6, 48, 12, 42, 24, 36, 24, 36, 12, 30, 12, 4...
## $ credit_history         <fct> critical/other existing credit', existing pa...
## $ purpose                <fct> radio/tv, radio/tv, education, furniture/equ...
## $ credit_amount          <dbl> 1169, 5951, 2096, 7882, 4870, 9055, 2835, 69...
## $ savings_status         <fct> no known savings', <100', <100', <100', <100...
## $ employment             <fct> >=7', 1<=X<4', 4<=X<7', 4<=X<7', 1<=X<4', 1<...
## $ installment_commitment <dbl> 4, 2, 2, 2, 3, 2, 3, 2, 2, 4, 3, 3, 1, 4, 2,...
## $ personal_status        <fct> male single', female div/dep/mar', male sing...
## $ other_parties          <fct> none, none, none, guarantor, none, none, non...
## $ residence_since        <dbl> 4, 2, 3, 4, 4, 4, 4, 2, 4, 2, 1, 4, 1, 4, 4,...
## $ property_magnitude     <fct> real estate', real estate', real estate', li...
## $ age                    <dbl> 67, 22, 49, 45, 53, 35, 53, 35, 61, 28, 25, ...
## $ other_payment_plans    <fct> none, none, none, none, none, none, none, no...
## $ housing                <fct> own, own, own, for free', for free', for fre...
## $ existing_credits       <dbl> 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1,...
## $ job                    <fct> skilled, skilled, unskilled resident', skill...
## $ num_dependents         <dbl> 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ own_telephone          <fct> yes, none, none, none, none, yes, none, yes,...
## $ foreign_worker         <fct> yes, yes, yes, yes, yes, yes, yes, yes, yes,...
## $ class                  <fct> good, bad, good, good, bad, good, good, good...
```

```
doubcount = 0
for (i in credit) {
  if (typeof(i)=="double") {
    doubcount = doubcount+1
  }
}
doubcount
```

```
## [1] 7
```

1.Summary data set

```
summary(credit)
```

```
##      checking_status    duration                                credit_history
##  <0'         :274    Min.   : 4.0   all paid'                          : 49
##  >=200'      : 63    1st Qu.:12.0   critical/other existing credit':293
##  0<=X<200'   :269    Median :18.0   delayed previously'                : 88
```

```
##  no checking':394    Mean   :20.9   existing paid'                 :530
##                      3rd Qu.:24.0   no credits/all paid'           : 40
##                      Max.   :72.0
##
##                 purpose    credit_amount             savings_status
##  radio/tv           :280   Min.   :  250   <100'            :603
##  new car'           :234   1st Qu.: 1366   >=1000'          : 48
##  furniture/equipment:181   Median : 2320   100<=X<500'      :103
##  used car'          :103   Mean   : 3271   500<=X<1000'     : 63
##  business           : 97   3rd Qu.: 3972   no known savings':183
##  education          : 50   Max.   :18424
##  (Other)            : 55
##       employment  installment_commitment            personal_status
##  <1'       :172   Min.   :1.000          female div/dep/mar':310
##  >=7'      :253   1st Qu.:2.000          male div/sep'      : 50
##  1<=X<4'   :339   Median :3.000          male mar/wid'      : 92
##  4<=X<7'   :174   Mean   :2.973          male single'       :548
##  unemployed: 62   3rd Qu.:4.000
##                   Max.   :4.000
##
##        other_parties residence_since          property_magnitude
##  co applicant': 41   Min.   :1.000   car               :332
##  guarantor    : 52   1st Qu.:2.000   life insurance'   :232
##  none         :907   Median :3.000   no known property':154
##                      Mean   :2.845   real estate'      :282
##                      3rd Qu.:4.000
##                      Max.   :4.000
##
##       age         other_payment_plans      housing    existing_credits
##  Min.   :19.00   bank  :139          for free':108   Min.   :1.000
##  1st Qu.:27.00   none  :814          own      :713   1st Qu.:1.000
##  Median :33.00   stores: 47          rent     :179   Median :1.000
##  Mean   :35.55                                       Mean   :1.407
##  3rd Qu.:42.00                                       3rd Qu.:2.000
##  Max.   :75.00                                       Max.   :4.000
##
##                           job      num_dependents  own_telephone foreign_worker
##  high qualif/self emp/mgmt':148   Min.   :1.000   none:596      no : 37
##  skilled                   :630   1st Qu.:1.000   yes :404      yes:963
##  unemp/unskilled non res'  : 22   Median :1.000
##  unskilled resident'       :200   Mean   :1.155
##                                   3rd Qu.:1.000
##                                   Max.   :2.000
##
##   class
##  bad :300
##  good:700
##
##
##
##
##
```

```
# splitting into training and testing dataset
credit_split_70 = createDataPartition(credit$class, p = 0.7, list = FALSE)
credit_split_80 = createDataPartition(credit$class, p = 0.8, list = F)
training_70 = credit[credit_split_70,]
```

```
## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
training_80 = credit[credit_split_80,]
test_70_features = credit[-credit_split_70, !(colnames(credit) %in% c('class'))]
test_70_target = credit[-credit_split_70, 'class']
test_80_features = credit[-credit_split_80, !(colnames(credit) %in% c('class'))]
test_80_target = credit[-credit_split_80, 'class']
```

```
# create decision trees
credit_tree_70 = rpart(class ~ . , data = training_70)
credit_tree_80 = rpart(class ~ . , data = training_80)
```

70% training:

```
credit_tree_70$variable.importance
```

```
##        checking_status                purpose          credit_amount
##             32.3432812             18.0329051             17.1152300
##               duration         credit_history         savings_status
##             14.5566732             13.8601778             12.8332418
##             employment        personal_status                    age
##              9.2154767              6.1626457              4.5596203
##     property_magnitude                    job                housing
##              3.4392282              2.1825554              1.6328884
##          own_telephone          other_parties        residence_since
##              1.5143449              1.2252698              0.4010449
##    other_payment_plans installment_commitment
##              0.3266667              0.2506156
```

```
jpeg(filename="credit_70.jpeg", width=1920, height = 1080)
rpart.plot(credit_tree_70)
dev.off()
```

```
## pdf
##   2
```

Validate tree before pruning

```
credit_tree_70_pred = predict(credit_tree_70, newdata = test_70_features)
pred_output_70_good = (credit_tree_70_pred[, "bad"] < 0.5)
test_target_good = test_70_target == "good"
(credit_70_mse = mean(test_target_good != pred_output_70_good))
```

```
## [1] 0.27
```

```
# detach('package:MASS', unload = TRUE)
# do this many times
# rate error of decision tree ()
times = 10
p = 0.7
```

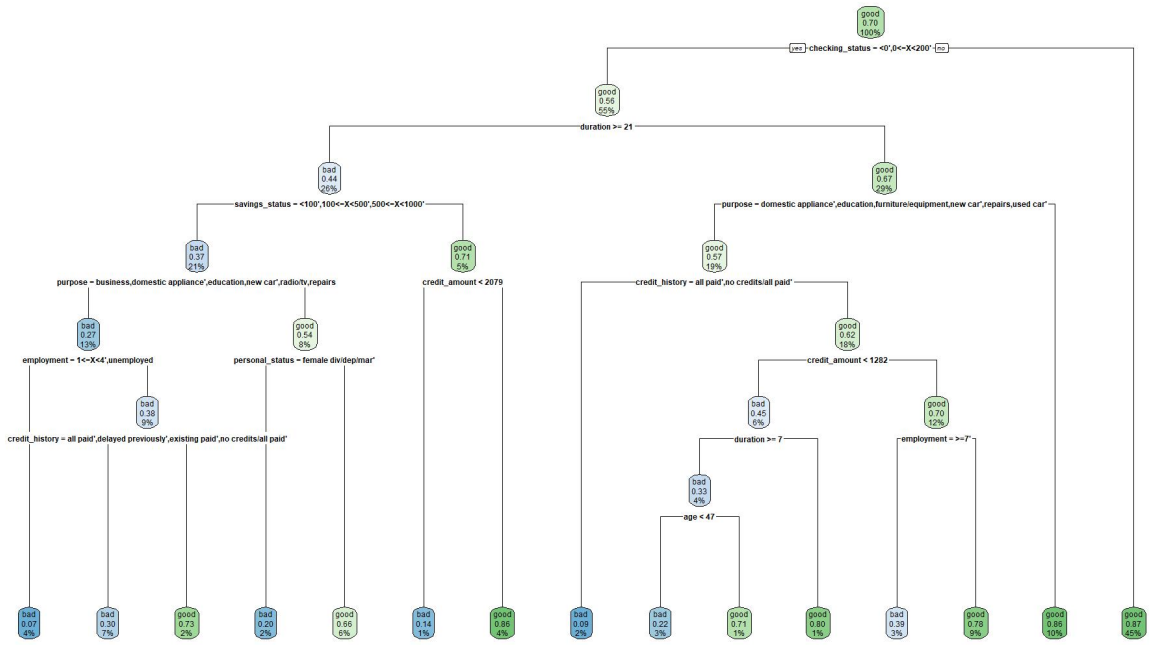


Figure 1: credit_70_tree

```
dec_tree_error_vect = as.numeric(times)
log_err_vect = as.numeric(times)
for(i in 1:times) {
  training_idx = createDataPartition(credit$class, p = p, list = F)
  training = credit[training_idx,]
  test_feats = credit[-training_idx, !(colnames(credit) %in% c('class'))]
  test_class = credit[-training_idx, 'class']
  dec_tree = rpart(class ~ . , data = training)
  fname = sprintf("dec_tree_%d.jpeg", i)
  jpeg(filename=fname, width=1920, height=1080)
  rpart.plot(dec_tree)
  dev.off()
  # test
  dec_tree_pred = predict(dec_tree, newdata = test_feats)
  predict_good = dec_tree_pred[, "bad"] < 0.5
  test_good = test_class == "good"
  dec_tree_error_vect[i] = mean(predict_good != test_good)

  # logistic regression
  log_model = glm(class ~ . , family = "binomial", data = training)
  log_pred = predict(log_model, newdata = test_feats, type = "response")
  log_pred = log_pred >= 0.5
  log_err_vect[i] = mean(log_pred != test_good)
}
```

```
# visualization
dec_tree_error_vect
```

```
##  [1] 0.2900000 0.2566667 0.2600000 0.2866667 0.2466667 0.2633333 0.2633333
##  [8] 0.2800000 0.3033333 0.3433333
```

```
log_err_vect
```

```
##  [1] 0.2966667 0.2600000 0.2700000 0.2233333 0.2433333 0.2433333 0.2733333
##  [8] 0.2700000 0.2500000 0.2866667
```

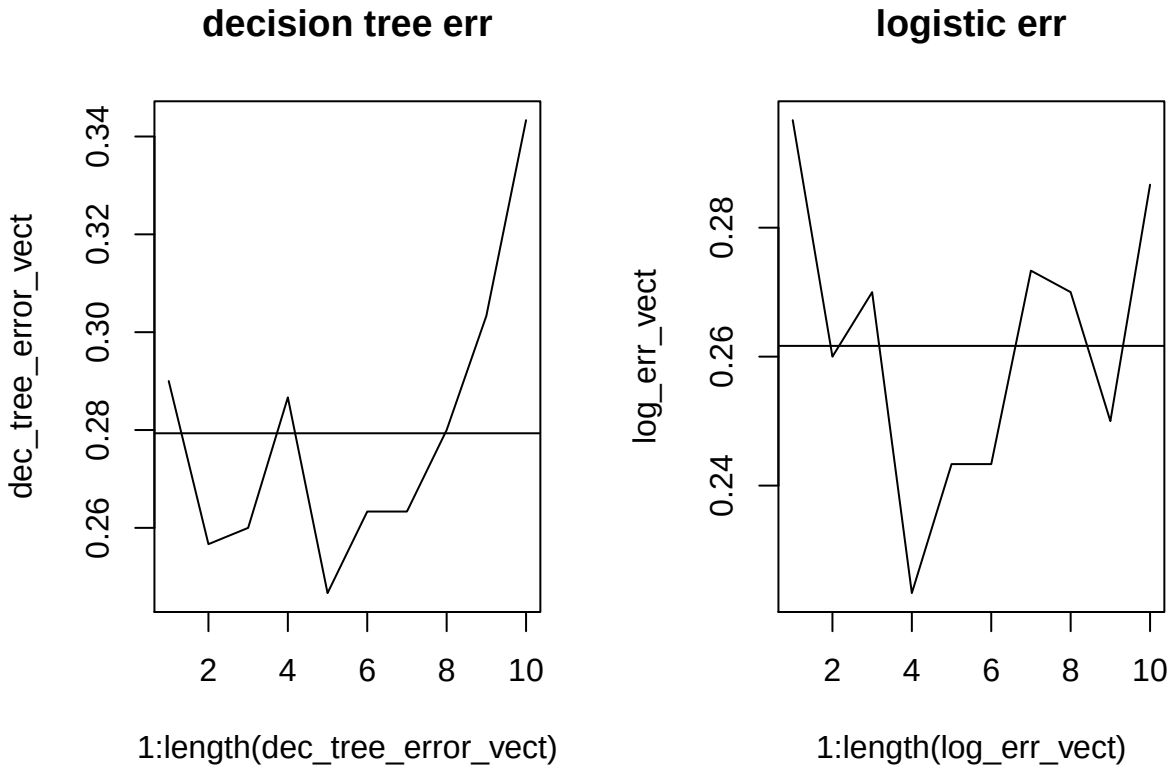
```
par(mfrow=c(1,2))
(dec_tree_err = mean(dec_tree_error_vect))
```

```
## [1] 0.2793333
```

```
(log_tree_err = mean(log_err_vect))
```

```
## [1] 0.2616667
```

```
plot(1:length(dec_tree_error_vect), dec_tree_error_vect, type = "l", main = "decision tree err")
abline(h = dec_tree_err)
plot(1:length(log_err_vect), log_err_vect, type = "l", main = "logistic err")
abline(h=log_tree_err)
```

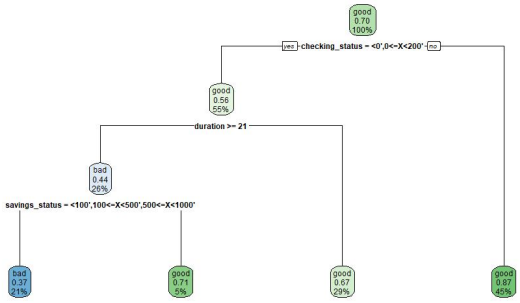



```
# attach("package:MASS")
```

Prune tree

```
credit_tree_70_pruned = prune(credit_tree_70, cp = 0.045)
jpeg(filename="credit_70_pruned.jpeg", width=1920, height = 1080)
rpart.plot(credit_tree_70_pruned)
dev.off()
```

```
## pdf
##   2
```



2. Backward criterion

```
full_mod <- glm(class~., family = "binomial", credit)
summary(full_mod)
```

```
##
## Call:
## glm(formula = class ~ ., family = "binomial", data = credit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6116  -0.7095   0.3752   0.6994   2.3410
##
## Coefficients:
##                                              Estimate Std. Error z value
## (Intercept)                                 1.505e+00  1.248e+00   1.206
## checking_status>=200'                       9.657e-01  3.692e-01   2.616
## checking_status0<=X<200'                    3.749e-01  2.179e-01   1.720
## checking_statusno checking'                 1.712e+00  2.322e-01   7.373
## duration                                   -2.786e-02  9.296e-03  -2.997
## credit_historycritical/other existing credit'  1.579e+00  4.381e-01   3.605
## credit_historydelayed previously'           9.965e-01  4.703e-01   2.119
## credit_historyexisting paid'                7.295e-01  3.852e-01   1.894
## credit_historyno credits/all paid'          1.434e-01  5.489e-01   0.261
## purposedomestic appliance'                 -2.173e-01  8.041e-01  -0.270
## purposeeducation                           -7.764e-01  4.660e-01  -1.666
## purposefurniture/equipment                  5.152e-02  3.543e-01   0.145
## purposenew car'                            -7.401e-01  3.339e-01  -2.216
## purposeother                                7.487e-01  7.998e-01   0.936
## purposeradio/tv                             1.515e-01  3.370e-01   0.450
```

```
## purposerepairs                                    -5.237e-01  5.933e-01  -0.883
## purposeretraining                                   1.319e+00  1.233e+00   1.070
## purposeused car'                                    9.264e-01  4.409e-01   2.101
## credit_amount                                      -1.283e-04  4.444e-05  -2.887
## savings_status>=1000'                               1.339e+00  5.249e-01   2.551
## savings_status100<=X<500'                           3.577e-01  2.861e-01   1.250
## savings_status500<=X<1000'                          3.761e-01  4.011e-01   0.938
## savings_statusno known savings'                     9.467e-01  2.625e-01   3.607
## employment>=7'                                      2.097e-01  2.947e-01   0.712
## employment1<=X<4'                                   1.159e-01  2.423e-01   0.478
## employment4<=X<7'                                   7.641e-01  3.051e-01   2.504
## employmentunemployed                               -6.691e-02  4.270e-01  -0.157
## installment_commitment                             -3.301e-01  8.828e-02  -3.739
## personal_statusmale div/sep'                       -2.755e-01  3.865e-01  -0.713
## personal_statusmale mar/wid'                        9.162e-02  3.118e-01   0.294
## personal_statusmale single'                         5.406e-01  2.102e-01   2.572
## other_partiesguarantor                              1.415e+00  5.685e-01   2.488
## other_partiesnone                                   4.360e-01  4.101e-01   1.063
## residence_since                                    -4.776e-03  8.641e-02  -0.055
## property_magnitudelife insurance'                  -8.690e-02  2.313e-01  -0.376
## property_magnitudeno known property'               -5.359e-01  4.017e-01  -1.334
## property_magnitudereal estate'                      1.945e-01  2.360e-01   0.824
## age                                                 1.454e-02  9.222e-03   1.576
## other_payment_plansnone                             6.463e-01  2.391e-01   2.703
## other_payment_plansstores                           1.232e-01  4.119e-01   0.299
## housingown                                         -2.402e-01  4.503e-01  -0.534
## housingrent                                        -6.839e-01  4.770e-01  -1.434
## existing_credits                                   -2.721e-01  1.895e-01  -1.436
## jobskilled                                         -7.524e-02  2.845e-01  -0.264
## jobunemp/unskilled non res'                         4.795e-01  6.623e-01   0.724
## jobunskilled resident'                             -5.666e-02  3.501e-01  -0.162
## num_dependents                                     -2.647e-01  2.492e-01  -1.062
## own_telephoneyes                                    3.000e-01  2.013e-01   1.491
## foreign_workeryes                                  -1.392e+00  6.258e-01  -2.225
##                                                   Pr(>|z|)
## (Intercept)                                       0.227801
## checking_status>=200'                             0.008905 **
## checking_status0<=X<200'                          0.085400 .
## checking_statusno checking'                       1.66e-13 ***
## duration                                          0.002724 **
## credit_historycritical/other existing credit'     0.000312 ***
## credit_historydelayed previously'                 0.034105 *
## credit_historyexisting paid'                      0.058238 .
## credit_historyno credits/all paid'                0.793921
## purposedomestic appliance'                        0.786976
## purposeeducation                                  0.095718 .
## purposefurniture/equipment                        0.884391
## purposenew car'                                   0.026668 *
## purposeother                                      0.349202
## purposeradio/tv                                   0.653002
## purposerepairs                                    0.377428
## purposeretraining                                 0.284625
## purposeused car'                                  0.035645 *
## credit_amount                                     0.003894 **
```

```
## savings_status>=1000'                          0.010729 *
## savings_status100<=X<500'                      0.211130
## savings_status500<=X<1000'                     0.348476
## savings_statusno known savings'                0.000310 ***
## employment>=7'                                 0.476718
## employment1<=X<4'                              0.632415
## employment4<=X<7'                              0.012271 *
## employmentunemployed                           0.875475
## installment_commitment                         0.000185 ***
## personal_statusmale div/sep'                   0.476040
## personal_statusmale mar/wid'                   0.768908
## personal_statusmale single'                    0.010113 *
## other_partiesguarantor                         0.012834 *
## other_partiesnone                              0.287700
## residence_since                                0.955920
## property_magnitudelife insurance'              0.707115
## property_magnitudeno known property'           0.182211
## property_magnitudereal estate'                 0.409743
## age                                            0.114982
## other_payment_plansnone                        0.006871 **
## other_payment_plansstores                      0.764878
## housingown                                     0.593687
## housingrent                                    0.151657
## existing_credits                               0.151109
## jobskilled                                     0.791419
## jobunemp/unskilled non res'                    0.469086
## jobunskilled resident'                         0.871450
## num_dependents                                 0.288249
## own_telephoneyes                               0.136060
## foreign_workeryes                              0.026095 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  895.82  on 951  degrees of freedom
## AIC: 993.82
##
## Number of Fisher Scoring iterations: 5
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
step.model <- full_mod %>% stepAIC(trace = FALSE)
step.model$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
```

```
## Initial Model:
## class ~ checking_status + duration + credit_history + purpose +
##     credit_amount + savings_status + employment + installment_commitment +
##     personal_status + other_parties + residence_since + property_magnitude +
##     age + other_payment_plans + housing + existing_credits +
##     job + num_dependents + own_telephone + foreign_worker
##
## Final Model:
## class ~ checking_status + duration + credit_history + purpose +
##     credit_amount + savings_status + installment_commitment +
##     personal_status + other_parties + age + other_payment_plans +
##     housing + own_telephone + foreign_worker
##
##
##                   Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                          951   895.8178 993.8178
## 2                - job  3 0.73855851       954   896.5563 988.5563
## 3 - property_magnitude  3 3.23574958       957   899.7921 985.7921
## 4    - residence_since  1 0.02024351       958   899.8123 983.8123
## 5     - num_dependents  1 0.99789651       959   900.8102 982.8102
## 6   - existing_credits  1 1.99124290       960   902.8015 982.8015
## 7         - employment  4 7.69651997       964   910.4980 982.4980
```

2. Random forest

```
split_index2 <- createDataPartition(credit$class, p = 0.8, list = F)
training2 <- credit[split_index2,]
features_test2 <- credit[-split_index2, !(colnames(credit) %in% c('class'))]
target_test2 <- credit[-split_index2, 'class']
sqrt(20)
```

```
## [1] 4.472136
```

```
rf_train <- randomForest(class~., data= training2, mtry = 5)
importance(rf_train)[order(importance(rf_train), decreasing = TRUE),]
```

```
##        checking_status          credit_amount                    age
##              41.995154              41.460842              31.186275
##               duration                purpose         credit_history
##              30.873031              29.444732              22.141373
##             employment         savings_status     property_magnitude
##              19.450953              16.871337              15.955793
##       residence_since installment_commitment        personal_status
##              12.924640              12.098596              11.784183
##                    job    other_payment_plans                housing
##               9.938835               8.530450               6.856880
##       existing_credits          other_parties          own_telephone
##               6.610255               6.558719               4.901846
##         num_dependents         foreign_worker
##               3.459159               1.171740
```

```
rf_preds <- predict(rf_train, newdata = features_test2)
rf_error_rate <- mean(rf_preds != target_test2$class)
rf_error_rate
```

```
## [1] 0.235
```