Credit_card_ML.R

rocka

2024-03-04

```r
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.3.3

## -- Attaching packages -------------------------------------- tidymodels 1.1.1 --

## v broom        1.0.5     v recipes      1.0.8
## v dials        1.2.1     v rsample      1.2.0
## v dplyr        1.1.2     v tibble       3.2.1
## v ggplot2      3.4.4     v tidyr        1.3.0
## v infer        1.0.6     v tune         1.1.2
## v modeldata    1.3.0     v workflows    1.1.4
## v parsnip      1.2.0     v workflowsets 1.0.1
## v purrr        1.0.2     v yardstick    1.3.0

## Warning: package 'dials' was built under R version 4.3.2

## Warning: package 'scales' was built under R version 4.3.2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'infer' was built under R version 4.3.2

## Warning: package 'modeldata' was built under R version 4.3.2

## Warning: package 'parsnip' was built under R version 4.3.2

## Warning: package 'recipes' was built under R version 4.3.2

## Warning: package 'rsample' was built under R version 4.3.2

## Warning: package 'tune' was built under R version 4.3.2

## Warning: package 'workflows' was built under R version 4.3.2

## Warning: package 'workflowsets' was built under R version 4.3.2

## Warning: package 'yardstick' was built under R version 4.3.2
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```r
library(vip)
```

```
## Warning: package 'vip' was built under R version 4.3.3

##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

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

```
## Warning: package 'rpart.plot' was built under R version 4.3.2

## Loading required package: rpart

##
## Attaching package: 'rpart'

## The following object is masked from 'package:dials':
##
##     prune
```

```r
credit_card_df<-read.csv("customer.csv")

credit_card_df$customer_status <- factor(credit_card_df$customer_status,
                                          levels = c("closed_account", "active"))

#Splitting the data into training and testing sets

set.seed(7)
cdf_split<-initial_split(credit_card_df,prop=0.75,strata=customer_status)

cdf_training<-cdf_split %>% training()

cdf_testing<-cdf_split %>% testing()

cdf_split
```

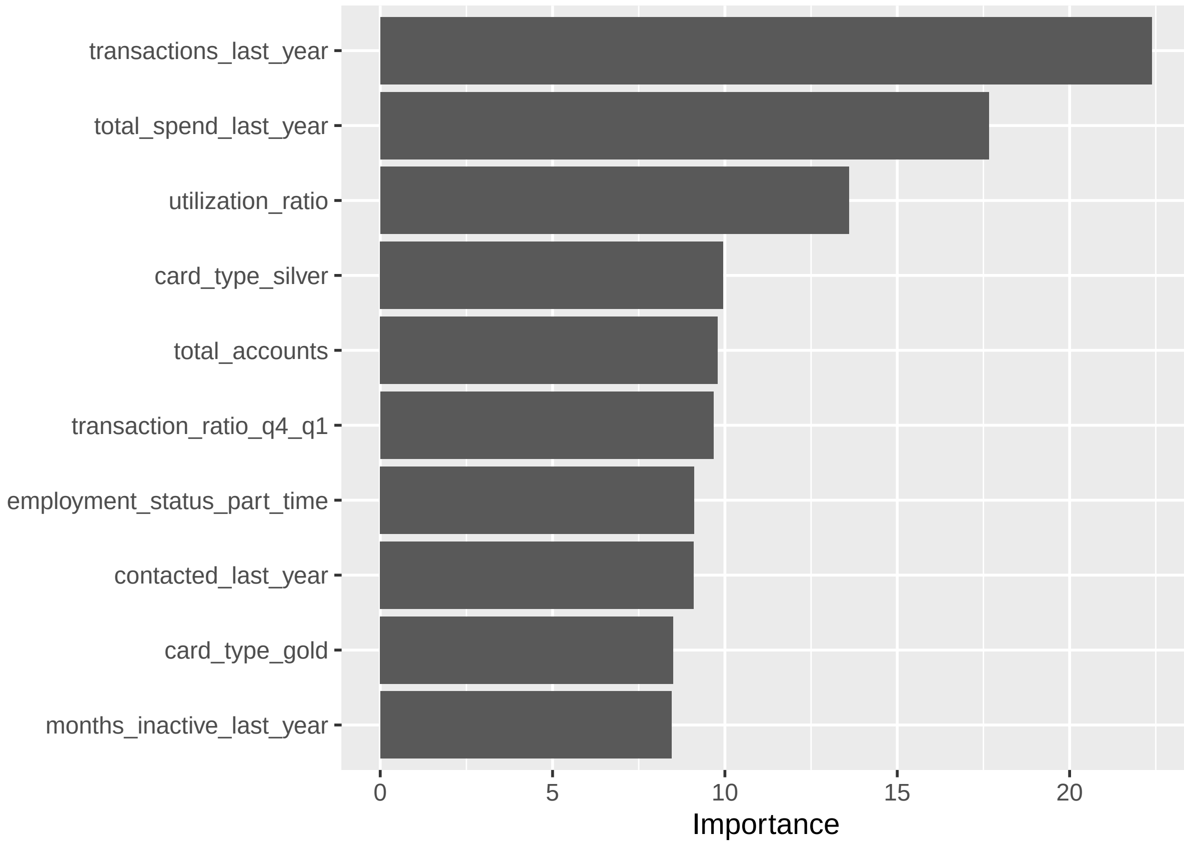
```
## <Training/Testing/Total>
## <3470/1157/4627>
```

```r
#Creating fold for cross validation
set.seed(7)
cdf_folds <- vfold_cv(cdf_training, v = 5)

cdf_recipe <-
  recipe(customer_status ~ ., data = cdf_training)  %>%
  step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_dummy(all_nominal(), -all_outcomes())

#Cross checking the feature engineering
(cdf_recipe %>%
    prep(training = cdf_training) %>%
    bake(new_data = NULL))
```

```
## # A tibble: 3,470 x 24
##        X     age dependents income months_since_first_account total_accounts
##    <dbl>   <dbl>      <dbl>  <dbl>                      <dbl>          <dbl>
##  1 -2.17 -0.441      -0.272 -0.976                     -0.172         -0.383
##  2 -2.16  0.959      -1.05  -0.928                      1.40           1.45
##  3 -2.16 -0.315       0.493 -0.800                     -0.793          1.45
##  4 -2.13  2.00       -0.272 -0.896                      1.94           1.45
##  5 -2.13  0.0642      0.493  0.675                     -0.0457        -1.04
##  6 -2.13  0.702       1.25  -0.885                      1.53           0.857
##  7 -2.12 -0.441       1.25  -0.789                     -0.0457         1.45
##  8 -2.12 -0.692       0.493 -0.996                     -0.671         -1.04
##  9 -2.10 -1.56       -0.272 -1.13                      -2.18           0.246
## 10 -2.10  1.48       -1.86  -0.925                     -0.0457         1.45
## # i 3,460 more rows
## # i 18 more variables: months_inactive_last_year <dbl>,
## #   contacted_last_year <dbl>, credit_limit <dbl>, utilization_ratio <dbl>,
## #   spend_ratio_q4_q1 <dbl>, total_spend_last_year <dbl>,
## #   transactions_last_year <dbl>, transaction_ratio_q4_q1 <dbl>,
## #   customer_status <fct>, education_bachelors <dbl>,
## #   education_doctorate <dbl>, education_masters <dbl>, ...
```

```r
##Model 1 - Logistic Regression Model

#Specifying the model
logistic_model <- logistic_reg() %>%
  set_engine('glm') %>%
  set_mode('classification')

#Creating a workflow
cdf_wf <- workflow() %>%
  add_model(logistic_model) %>%
  add_recipe(cdf_recipe)

#Fitting the model
cdf_logistic_fit <- cdf_wf %>%
  fit(data = cdf_training)

#Exploring the model to see the importance of the predictors and plotting the same
```

```r
cdf_trained_model <- cdf_logistic_fit %>%
  extract_fit_parsnip()

vip(cdf_trained_model)
```



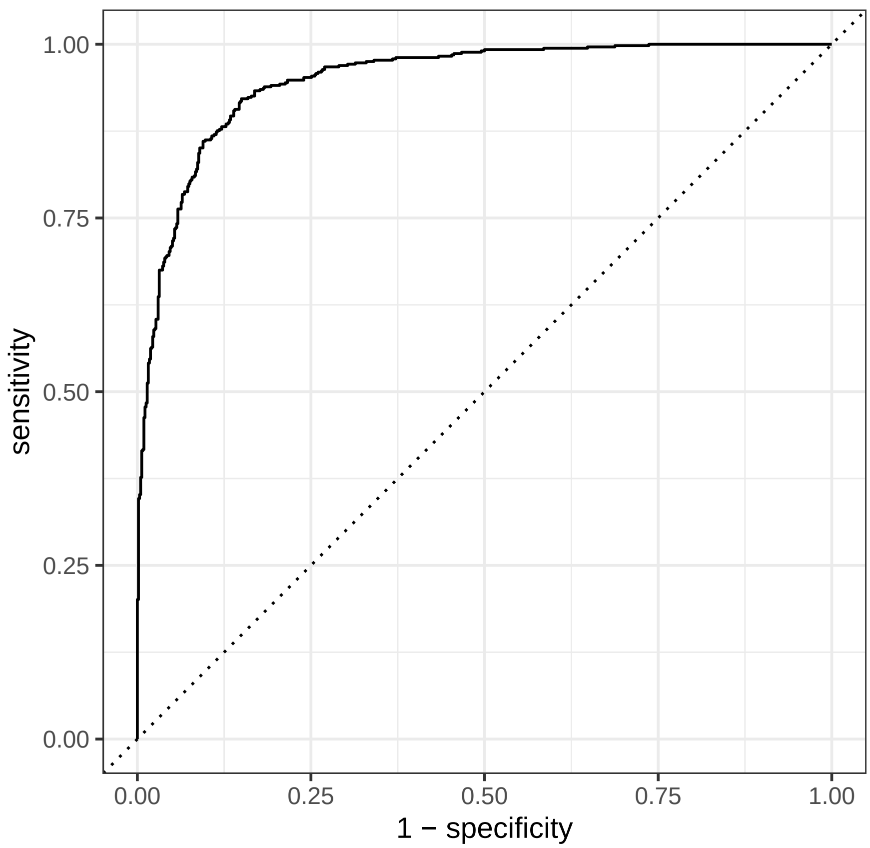

```r
#Predicting using the test data
predictions_categories <- predict(cdf_logistic_fit, new_data = cdf_testing)
predictions_probabilities <- predict(cdf_logistic_fit, new_data = cdf_testing, type = 'prob')


test_results <-
  cdf_testing %>%
  dplyr::select(customer_status) %>%
  bind_cols(predictions_categories) %>%
  bind_cols(predictions_probabilities)

head(test_results)
```

```
##   customer_status    .pred_class .pred_closed_account .pred_active
## 1  closed_account         active         0.3389566667   0.66104333
## 2  closed_account closed_account         0.5113060000   0.48869400
## 3          active         active         0.0003688341   0.99963117
## 4  closed_account closed_account         0.9797043905   0.02029561
## 5          active closed_account         0.6727272090   0.32727279
## 6          active closed_account         0.6263812123   0.37361879
```

```r
#Evaluating the model performance
#Confusion Matrix
conf_mat(test_results,
         truth = customer_status,
         estimate = .pred_class)
```

```
##                 Truth
## Prediction       closed_account active
##   closed_account            451     63
##   active                     72    571
```

```r
#ROC curve
roc_curve(test_results,
          truth = customer_status,
          .pred_closed_account) %>%
  autoplot()
```




```r
#Looking at various performance metrics
perf_metrics <- metric_set(accuracy, sens, spec, f_meas, roc_auc)

perf_metrics(test_results,
             truth = customer_status,
             estimate = .pred_class,
             .pred_closed_account)
```

```
## # A tibble: 5 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.883
## 2 sens     binary         0.862
## 3 spec     binary         0.901
## 4 f_meas   binary         0.870
## 5 roc_auc  binary         0.949
```

```
##Model 2 -Decision Tree Model

#Specfying the model
tree_model <-
  decision_tree(cost_complexity = tune(),
                tree_depth = tune(),
                min_n = tune()) %>%
  set_engine('rpart') %>%
  set_mode('classification')

#Creating a workflow
tree_workflow <-
  workflow() %>%
  add_model(tree_model) %>%
  add_recipe(cdf_recipe)

#Creating a grid of hyperparameter values to test
tree_grid <-
  grid_regular(cost_complexity(),
               tree_depth(),
               min_n(),
               levels = 2)

#Tuning decision tree workflow
set.seed(7)
tree_tuning <-
  tree_workflow %>%
  tune_grid(resamples = cdf_folds, grid = tree_grid)

#Showing top 5 best models (roc_auc metric)
tree_tuning %>% show_best('roc_auc')
```

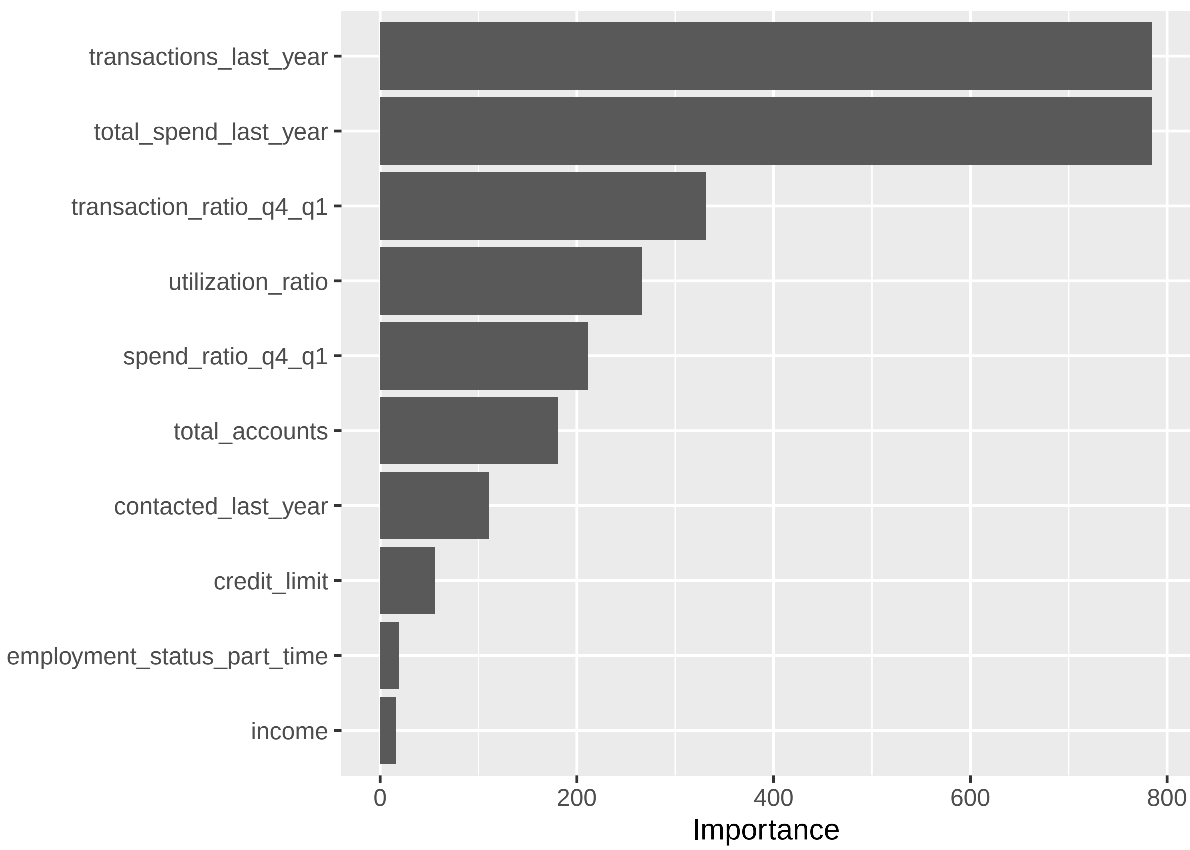
```
## # A tibble: 5 x 9
##   cost_complexity tree_depth min_n .metric .estimator  mean     n std_err
##             <dbl>      <int> <int> <chr>   <chr>      <dbl> <int>   <dbl>
## 1    0.0000000001         15    40 roc_auc binary     0.950     5 0.00401
## 2    0.0000000001         15     2 roc_auc binary     0.906     5 0.00248
## 3    0.0000000001          1     2 roc_auc binary     0.769     5 0.00793
## 4    0.1                   1     2 roc_auc binary     0.769     5 0.00793
## 5    0.1                  15     2 roc_auc binary     0.769     5 0.00793
## # i 1 more variable: .config <chr>
```

```
#Filtering out the best model based on roc_auc and checking the tree parameters for the same
(best_tree <-
```

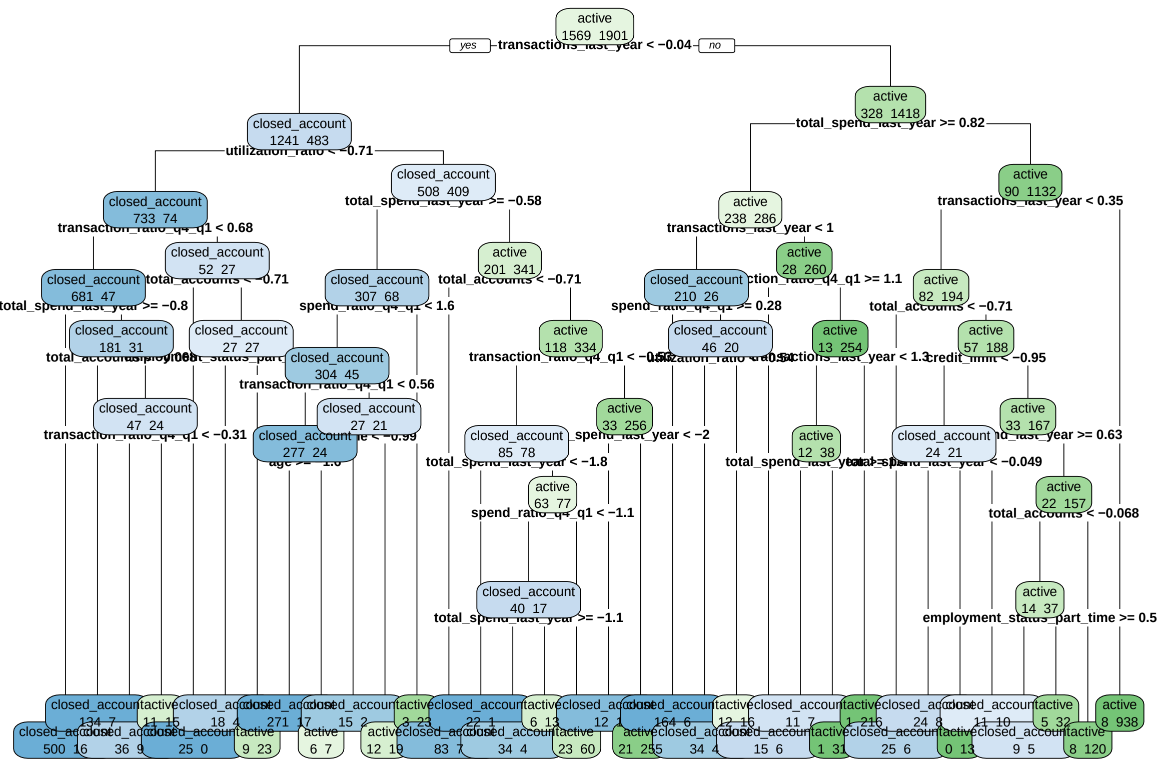
```r
    tree_tuning %>%
    select_best(metric = 'roc_auc'))
```

```
## # A tibble: 1 x 4
##   cost_complexity tree_depth min_n .config
##             <dbl>      <int> <int> <chr>
## 1    0.0000000001         15    40 Preprocessor1_Model7
```

```r
#Finalizing workflow
final_tree_workflow <-
  tree_workflow %>%
  finalize_workflow(best_tree)

#Fitting the model to the training data
tree_wf_fit <-
  final_tree_workflow %>%
  fit(data = cdf_training)

#Exploring the model to see the importance of the predictors and plotting the same
tree_fit <-
  tree_wf_fit %>%
  extract_fit_parsnip()

vip(tree_fit)
```



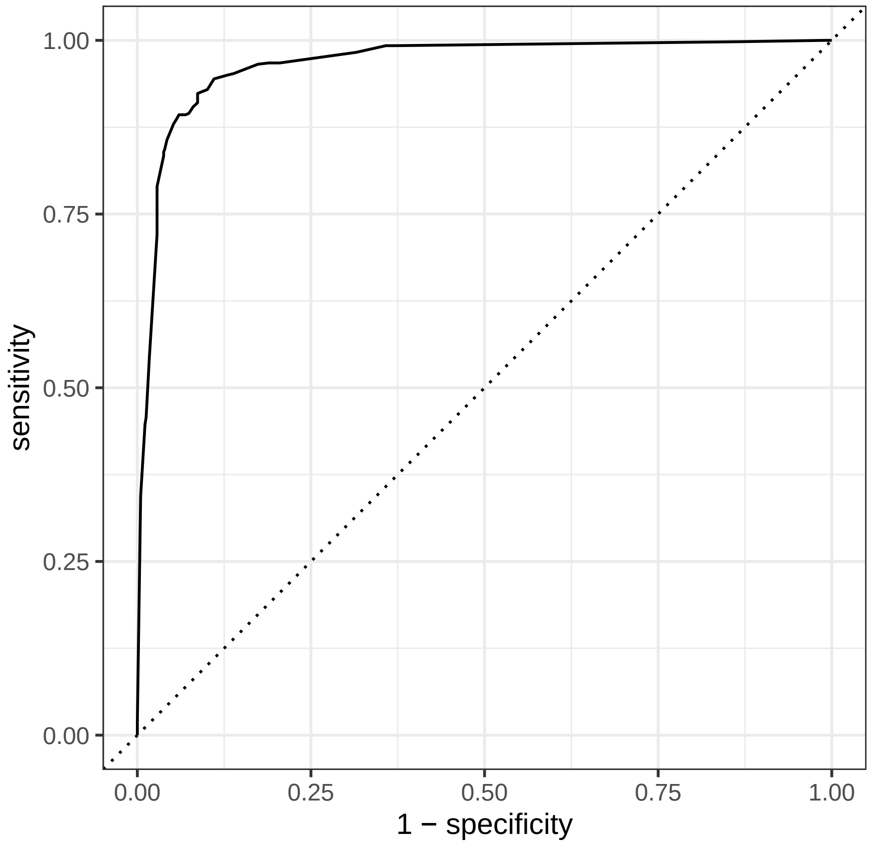
```r
#Training the model and generating predictions on the test data
tree_last_fit <-
  final_tree_workflow %>%
  last_fit(cdf_split)

#Plotting the decsion tree
rpart.plot(tree_fit$fit, roundint = FALSE, extra = 1,cex=0.45)
```




```r
#Evaluating the model performance
#Peformance metrics
tree_last_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.913 Preprocessor1_Model1
## 2 roc_auc  binary         0.966 Preprocessor1_Model1
```

```r
#roc_auc plot
tree_last_fit %>%
  collect_predictions() %>%
  roc_curve(truth = customer_status, .pred_closed_account) %>%
  autoplot()
```




```r
#Confusion Matrix
tree_predictions <- tree_last_fit %>% collect_predictions()

conf_mat(tree_predictions, truth = customer_status, estimate = .pred_class)
```

```
##                   Truth
## Prediction        closed_account active
##   closed_account             473     51
##   active                      50    583
```

```r
## Model 3 - Random Forest

#Specifying the model
rf_model <-
  rand_forest(mtry = tune(),
              trees = tune(),
              min_n = tune()) %>%
  set_engine('ranger', importance = "impurity") %>%
  set_mode('classification')

#Creating a workflow
rf_workflow <-
  workflow() %>%
  add_model(rf_model) %>%
  add_recipe(cdf_recipe)
```

```r
#Creating a grid of hyperparameter values to test
set.seed(7)
rf_grid <-
  grid_random(mtry() %>% range_set(c(2, 4)),
              trees(),
              min_n(),
              size = 10)

#Tuning the hyperparameters created
set.seed(7)
rf_tuning <-
  rf_workflow %>%
  tune_grid(resamples = cdf_folds, grid = rf_grid)
```

```
## Warning: package 'ranger' was built under R version 4.3.2
```

```r
rf_tuning %>% show_best('roc_auc')
```

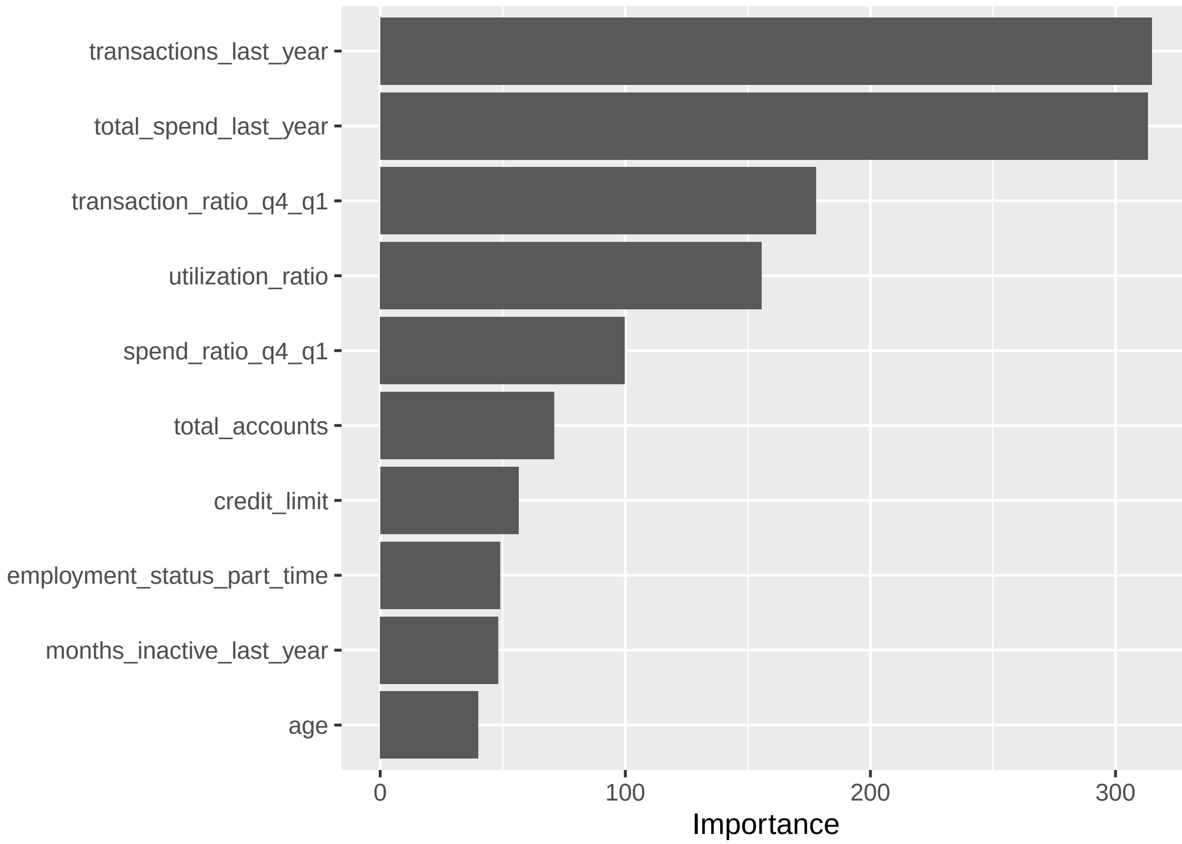
```
## # A tibble: 5 x 9
##    mtry trees min_n .metric .estimator  mean     n  std_err .config
##   <int> <int> <int> <chr>   <chr>      <dbl> <int>    <dbl> <chr>
## 1     4   947    12 roc_auc binary     0.987     5 0.000759 Preprocessor1_Model~
## 2     4  1928    17 roc_auc binary     0.986     5 0.000761 Preprocessor1_Model~
## 3     3   207     3 roc_auc binary     0.986     5 0.00111  Preprocessor1_Model~
## 4     3  1455     7 roc_auc binary     0.986     5 0.000997 Preprocessor1_Model~
## 5     4  1114    22 roc_auc binary     0.986     5 0.000816 Preprocessor1_Model~
```

```r
# Selecting and viewing the paramaters of the best model based on roc_auc
(best_rf <-
    rf_tuning %>%
    select_best(metric = 'roc_auc'))
```

```
## # A tibble: 1 x 4
##    mtry trees min_n .config
##   <int> <int> <int> <chr>
## 1     4   947    12 Preprocessor1_Model06
```

```r
#Finalizing the workflow
final_rf_workflow <-
  rf_workflow %>%
  finalize_workflow(best_rf)

#Fitting the model
rf_wf_fit <-
  final_rf_workflow %>%
  fit(data = cdf_training)

#Exploring the model to see the importance of the predictors and plotting the same
rf_fit <-
  rf_wf_fit %>%
  extract_fit_parsnip()
```

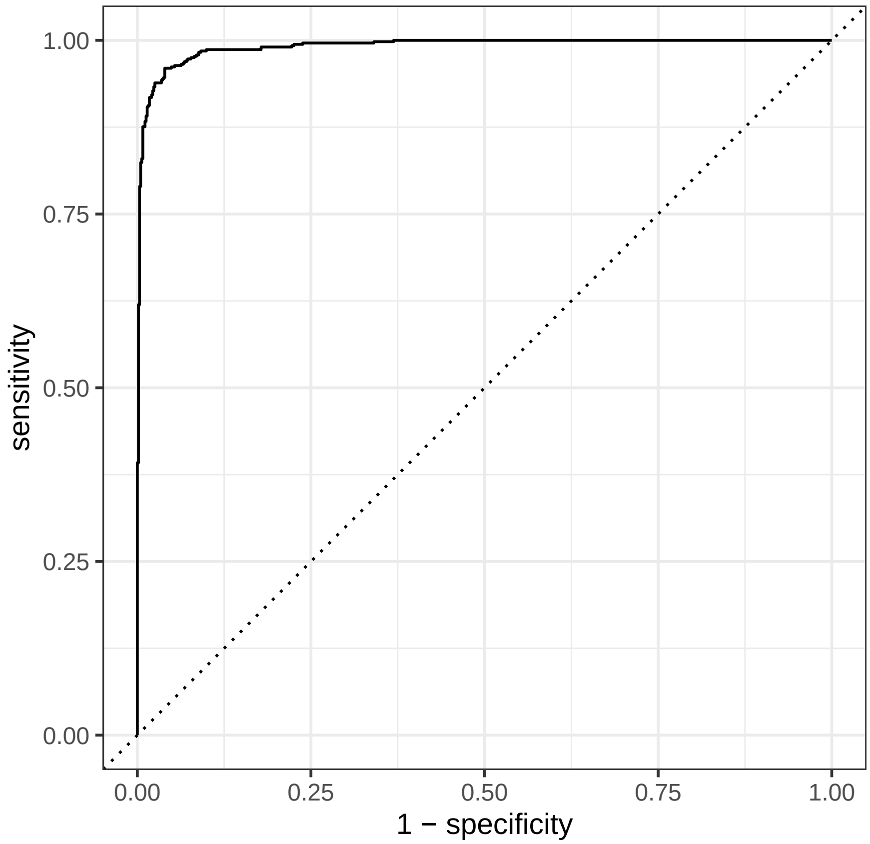
```r
vip(rf_fit)
```




```r
#Training the model and generating predictions on the test data
rf_last_fit <-
  final_rf_workflow %>%
  last_fit(cdf_split)

# Evaluating the model performance
# Performance metrics
rf_last_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.955 Preprocessor1_Model1
## 2 roc_auc  binary         0.991 Preprocessor1_Model1
```

```r
# roc_auc plot
rf_last_fit %>% collect_predictions() %>%
  roc_curve(truth = customer_status, .pred_closed_account) %>%
  autoplot()
```




```r
#Confusion matrix
rf_predictions <- rf_last_fit %>% collect_predictions()

conf_mat(rf_predictions, truth = customer_status, estimate = .pred_class)
```

```
##                   Truth
## Prediction        closed_account active
##   closed_account             502     31
##   active                      21    603
```