Account Closure Analysis

Your Name: Yueying Wang **Your G Number**: G01062022

Add R libraries here

```
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(tidymodels)
```

```
## Registered S3 method overwritten by 'tune':
##   method                   from
##   required_pkgs.model_spec parsnip

## -- Attaching packages --------------------------------------- tidymodels 0.1.3 --

## v broom        0.7.9      v rsample      0.1.0
## v dials        0.0.9      v tune         0.1.6
## v infer        1.0.0      v workflows    0.2.3
## v modeldata    0.1.1      v workflowsets 0.1.0
## v parsnip      0.1.7      v yardstick    0.0.8
## v recipes      0.1.16

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

```r
library(discrim)
```

```
##
## Attaching package: 'discrim'

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

```r
library(klaR)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

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

```r
library(kknn)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(dplyr)
library(ggplot2)
library(vip)
```

```
##
## Attaching package: 'vip'

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

```r
# Suppress dplyr summarise grouping warning messages
options(dplyr.summarise.inform = FALSE)


credit_card_df <- readRDS(url('https://gmubusinessanalytics.netlify.app/data/credit_card_df.rds'))
credit_card_df
```

```
## # A tibble: 4,627 x 18
##    customer_status   age dependents education  marital_status employment_status
##    <fct>           <dbl>      <dbl> <fct>      <fct>          <fct>
##  1 closed_account     46          3 masters    married        self_employed
##  2 closed_account     46          3 associates married        self_employed
##  3 closed_account     44          4 masters    single         part_time
##  4 closed_account     62          1 masters    single         part_time
##  5 closed_account     38          1 masters    married        full_time
##  6 closed_account     43          3 associates single         part_time
##  7 active             43          2 masters    married        full_time
##  8 closed_account     39          3 associates married        part_time
##  9 active             54          1 masters    single         full_time
## 10 active             46          4 masters    divorced       full_time
## # ... with 4,617 more rows, and 12 more variables: income <dbl>,
## #   card_type <fct>, months_since_first_account <dbl>, total_accounts <dbl>,
## #   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>
```

Data Analysis

In this section, you must think of at least 5 relevant questions that explore the relationship between `customer_status` and the other variables in the `credit_card_df` data set. The goal of your analysis should be discovering which variables drive the differences between customers who do and do not close their account.

You must answer each question and provide supporting data summaries with either a summary data frame (using `dplyr/tidyr`) or a plot (using `ggplot`) or both.

In total, you must have a minimum of 3 plots (created with `ggplot`) and 3 summary data frames (created with `dplyr`) for the exploratory data analysis section. Among the plots you produce, you must have at least 3 different types (ex. box plot, bar chart, histogram, scatter plot, etc...)

See the Data Analysis Project for an example of a question answered with a summary table and plot.

Note: To add an R code chunk to any section of your project, you can use the keyboard shortcut `Ctrl` + `Alt` + `i` or the `insert` button at the top of your R project template notebook file.

Question 1

Question:

Is there a relationship between spend_ratio and customer status?

Answer: Yes, from the data below we can see that people who did not close the account have higher spend_ratio with the card (around 0.77) than people who closed the account which is around 0.69. Additionally, when we look at the box plot we can clear see even though the average spending ratios are very close, the maximum and minimum have very a big difference. People who did not close the account minimum spend ratio is 0.256 compared to ratio 0 that people who did close the account. And the maximum spend ratio on people who did not close the account is 2.282 compared to ratio 1.492 of people who did close the account.

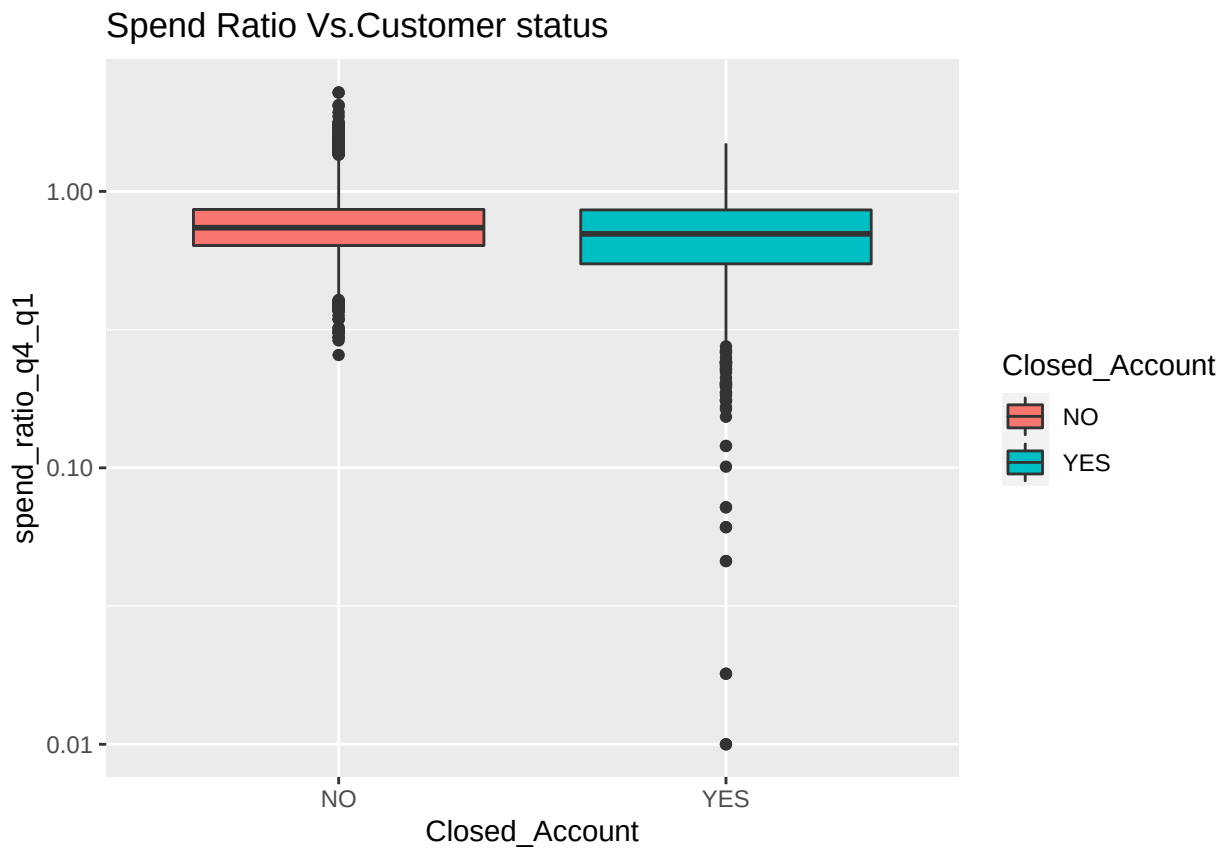
```
credit_card_data <- credit_card_df %>%
  add_column(Closed_Account = if_else(.$customer_status == "closed_account", "YES", "NO"))

credit_card_data %>% group_by(Closed_Account) %>%
                   summarise(n_customer = n(),
                             min_spend_ratio = min(spend_ratio_q4_q1),
                             max_spend_ratio = max(spend_ratio_q4_q1),
                             avg_spend_ratio = mean(spend_ratio_q4_q1))
```

```
## # A tibble: 2 x 5
##   Closed_Account n_customer min_spend_ratio max_spend_ratio avg_spend_ratio
##   <chr>               <int>           <dbl>           <dbl>           <dbl>
## 1 NO                   2535           0.256            2.28           0.771
## 2 YES                  2092           0                1.49           0.695
```

Data Visulatization

```
ggplot(credit_card_data, aes(x = Closed_Account , y = spend_ratio_q4_q1,fill = Closed_Account)) +
  geom_boxplot() +
  scale_y_log10() +ggtitle("Spend Ratio Vs.Customer status")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```



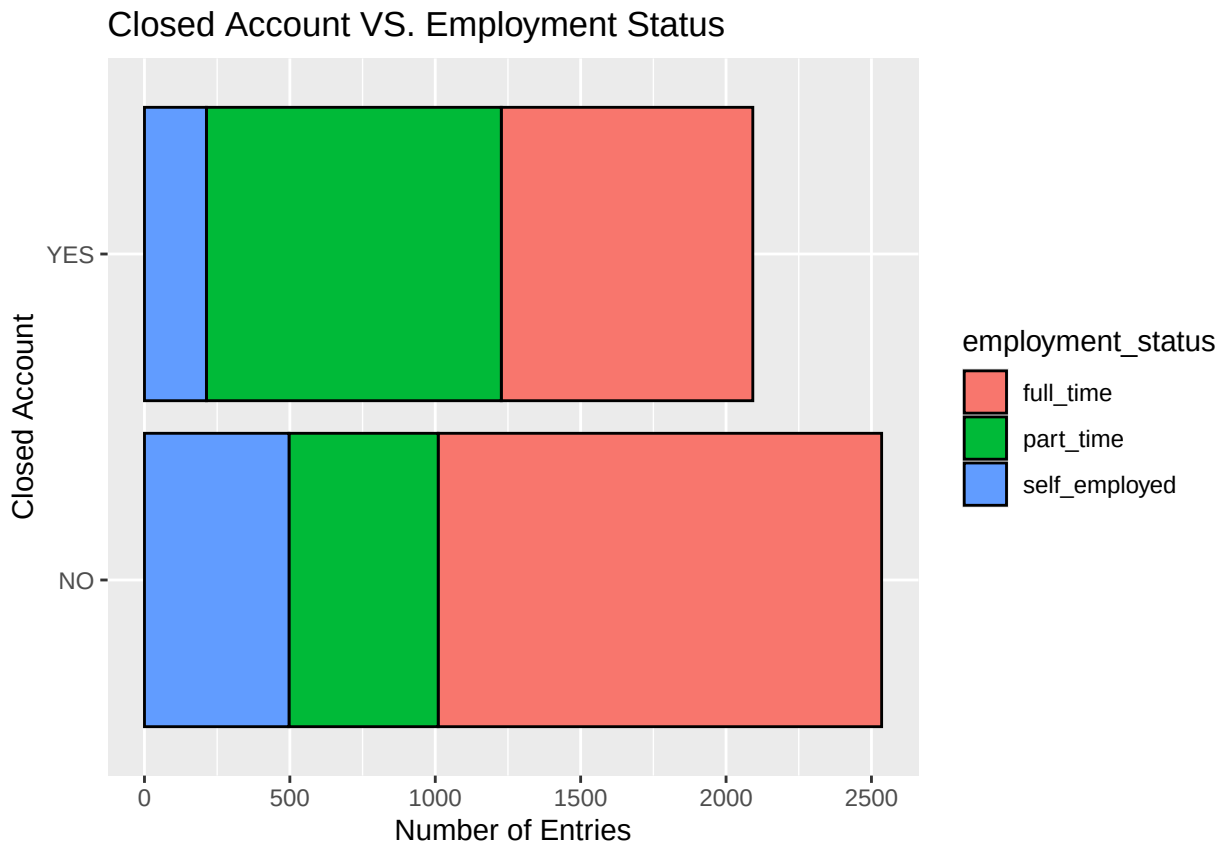
Question 2

Question:

Is there a relationship between employment status and customer status?

Answer: Yes, from the plot below we can see that there is a strong relationship between employment status and customer status.The customer who has employment status part-time is more likely to close the account. People who have employment status of full-time and self-employed are more willing to keep the account.

```
ggplot(credit_card_data, aes(x = Closed_Account, fill = employment_status)) +
  geom_bar(color="Black") +
  labs(x = "Closed Account",
       y = "Number of Entries",
       title = "Closed Account VS. Employment Status", fill = 'employment_status') +
  coord_flip()
```



Question 3

Question:

Are there relationships between months_since_first_account and months_inactive_last_year with customer status?

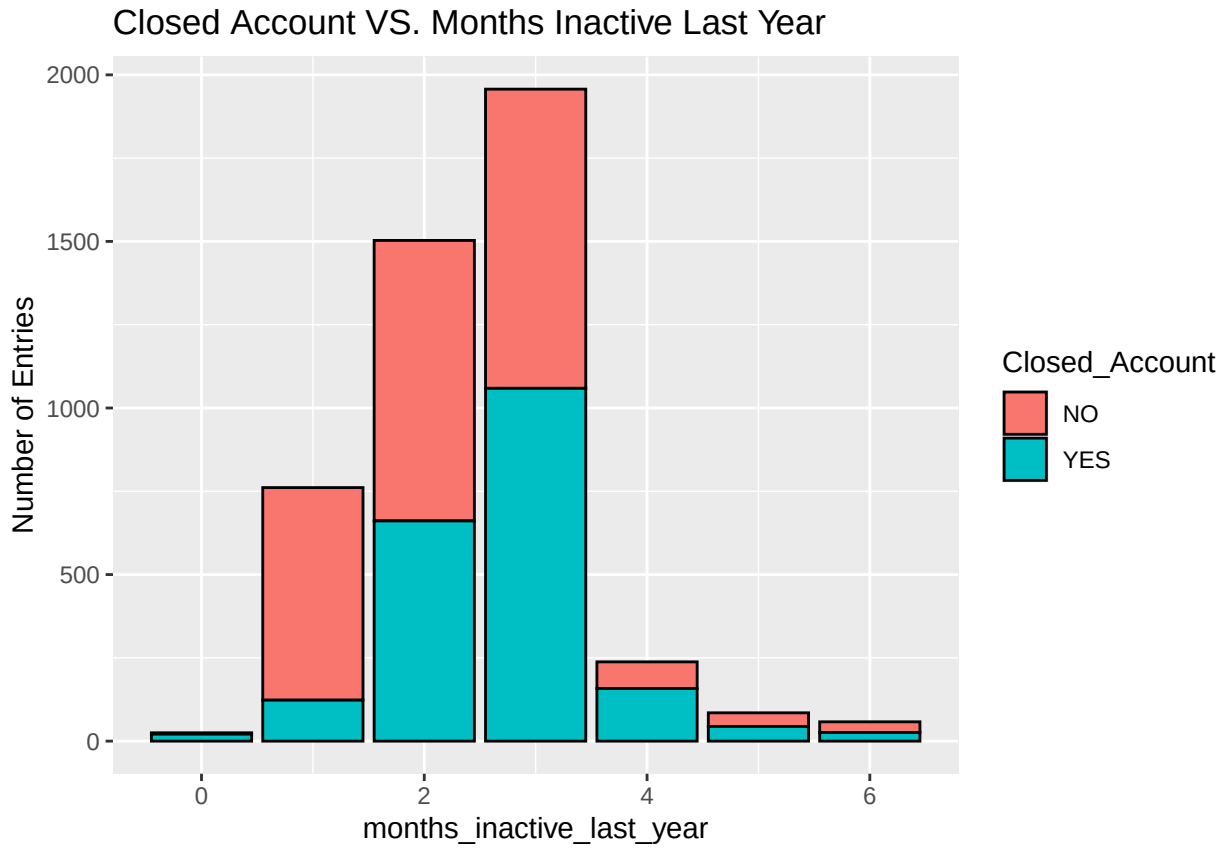
Answer:

Yes, there is a relationship between months_inactive_last_year with customer status. From the data below, we can see that customers are willing to close the account if the average number of months inactive last year is higher. But there is a very small impact on the relationship between months_since_first_account with customer status. From the data below, we can see that the average months since the first account between closed accounts and active accounts are almost the same.

```
credit_card_data %>% group_by(Closed_Account)%>%
                            summarise(n_Customer = n(),
                            avg_months_since_first_account = mean(months_since_first_account),
                            avg_months_inactive_last_year = mean(months_inactive_last_year))
```

```
## # A tibble: 2 x 4
##   Closed_Account n_Customer avg_months_since_first_ac~ avg_months_inactive_last~
##   <chr>               <int>                      <dbl>                      <dbl>
## 1 NO                   2535                       35.9                       2.26
## 2 YES                  2092                       36.2                       2.69
```

```
ggplot(credit_card_data, aes(x = months_inactive_last_year,fill = Closed_Account)) +
  geom_bar(color="Black")+
  labs(x = "months_inactive_last_year",
       y = "Number of Entries",
       title = "Closed Account VS. Months Inactive Last Year", fill = 'Closed_Account')
```



Question 4

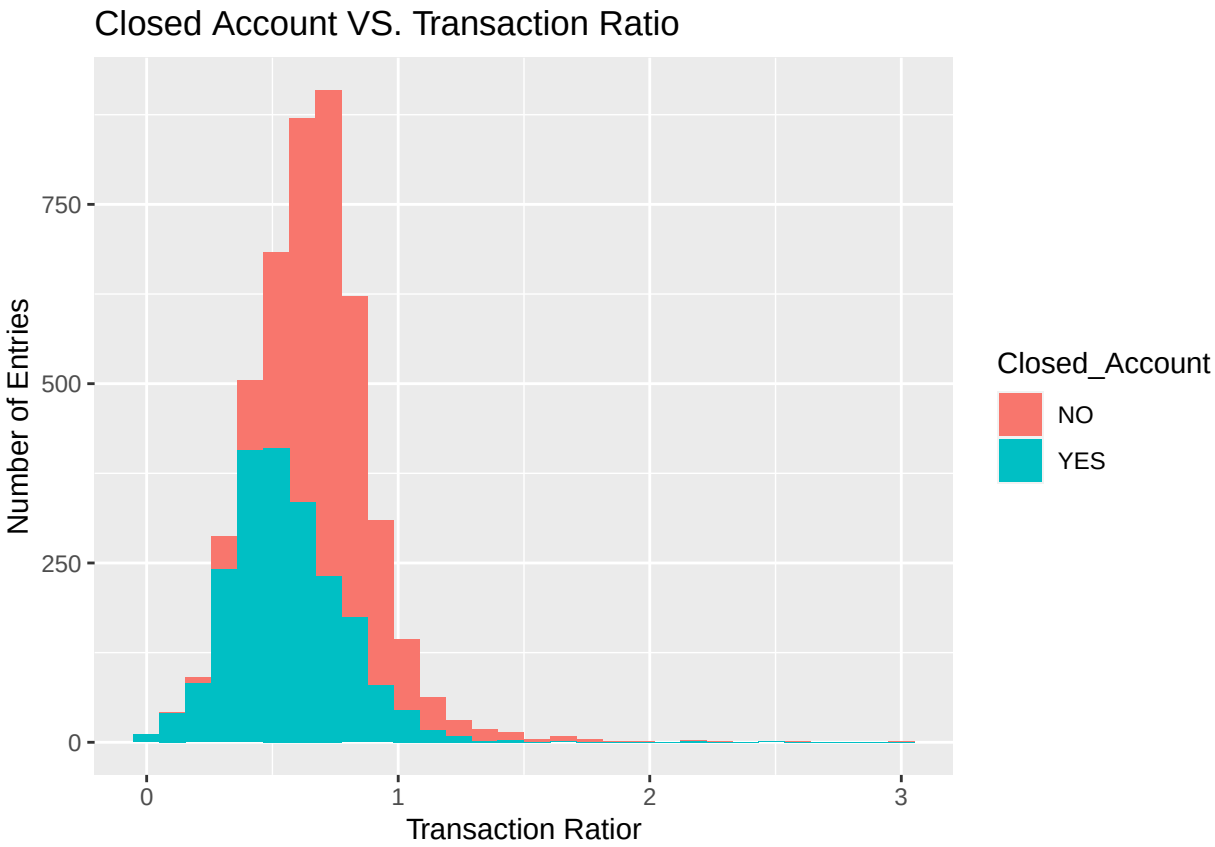
Question: Is there a relationship between transaction ratio and customer status?

Answer: Yes, from the data below we can see that the customer who has a lower transaction ratio will be more likely to close the account. The average transaction ratio for customers who stays active is 0.74, and the average transaction ratio for customers who closed accounts is 0.56. Therefore, the company needs to improve service of the customer that has a transaction ratio of less than 0.74.

```
credit_card_data %>% group_by(Closed_Account) %>%
                  summarise(n_customer = n(),
                            min_transaction_ratio = min(transaction_ratio_q4_q1),
                            max_transaction_ratio = max(transaction_ratio_q4_q1),
                            avg_transaction_ratio = mean(transaction_ratio_q4_q1),
                            avg_total_spend_last_year = mean(total_spend_last_year))
```

```
## # A tibble: 2 x 6
##   Closed_Account n_customer min_transaction_~ max_transaction_~ avg_transaction~
##   <chr>               <int>             <dbl>             <dbl>            <dbl>
## 1 NO                   2535             0.028               3              0.737
## 2 YES                  2092             0                   2.5            0.555
## # ... with 1 more variable: avg_total_spend_last_year <dbl>
```

```
ggplot(credit_card_data, aes(x = transaction_ratio_q4_q1,fill = Closed_Account)) +
  geom_histogram(bins = 30)+
  labs(x = "Transaction Ratior",
       y = "Number of Entries",
       title = "Closed Account VS. Transaction Ratio", fill = 'Closed_Account')
```



Question 5

Question:

Is there a relationship between card type and customer status?

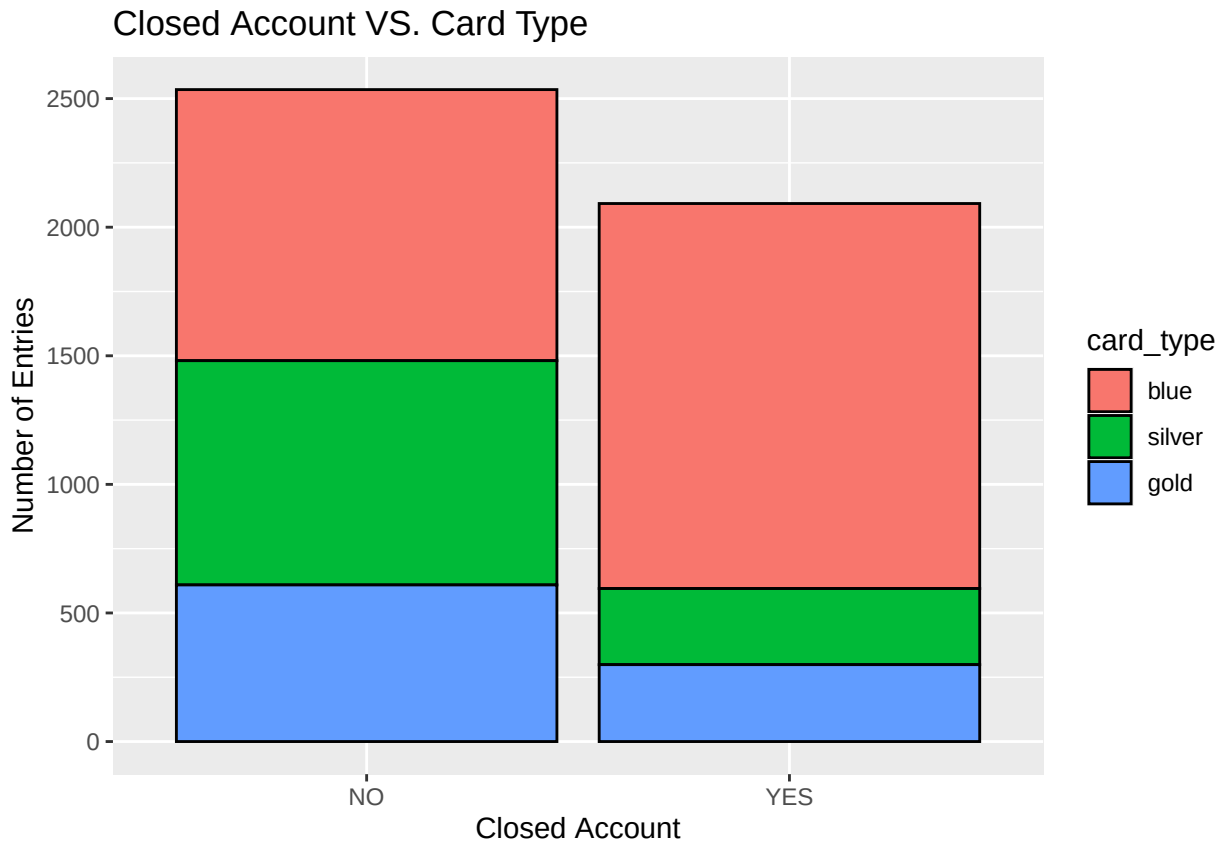
Answer: Yes, from the data below we can see that the customers who have a blue card are more likely to close the account (more than 400 blue card customers chose to close account compared to a customer who stays active based on customer number on 2500 scale), and over half of customers who hold the silver card and gold card are more willing to stay.

```
credit_card_data_type <-credit_card_data %>%
group_by(card_type,Closed_Account) %>%
summarize(n_customer = n(),
          mean_utilization_ratio = mean(utilization_ratio),
          mean_spend_ratio = mean(spend_ratio_q4_q1),
          mean_transaction_ratio = mean(transaction_ratio_q4_q1))
credit_card_data_type
```

```
## # A tibble: 6 x 6
## # Groups:   card_type [3]
##   card_type Closed_Account n_customer mean_utilization_ratio mean_spend_ratio
##   <fct>     <chr>               <int>                 <dbl>            <dbl>
## 1 blue      NO                   1054                 0.278            0.772
## 2 blue      YES                  1497                 0.162            0.696
```

```
## 3 silver    NO                  872                 0.292           0.772
## 4 silver    YES                 296                 0.159           0.687
## 5 gold      NO                  609                 0.306           0.768
## 6 gold      YES                 299                 0.159           0.700
## # ... with 1 more variable: mean_transaction_ratio <dbl>
```

```
ggplot(credit_card_data, aes(x = Closed_Account, fill =     card_type)) +
  geom_bar(color="Black") +
  labs(x = "Closed Account",
       y = "Number of Entries",
       title = "Closed Account VS. Card Type", fill = 'card_type')
```



Machine Learning

In this section of the project, you will fit **three classification algorithms** to predict the outcome variable,`customer_status`.

You must follow the machine learning steps below.

The data splitting and feature engineering steps should only be done once so that your models are using the same data and feature engineering steps for training.

- Split the `credit_card_df` data into a training and test set (remember to set your seed)
- Specify a feature engineering pipeline with the `recipes` package

  - You can include steps such as skewness transformation, correlation filters, dummy variable encoding or any other steps you find appropriate
- Specify a `parsnip` model object
  - You may choose from the following classification algorithms:
    * Logistic Regression
    * LDA
    * QDA
    * KNN
    * Decision Tree
    * Random Forest
- Package your recipe and model into a workflow
- Fit your workflow to the training data
  - If your model has hyperparameters:
    * Split the training data into 5 folds for 5-fold cross validation using `vfold_cv` (remember to set your seed)
    * Perform hyperparamter tuning with a random grid search using the `grid_random()` function
    * Refer to the following tutorial for an example - Random Grid Search
    * Hyperparameter tuning can take a significant amount of computing time. Be careful not to set the `size` argument of `grid_random()` too large. I recommend `size` = 10 or smaller.
    * Select the best model with `select_best()` and finalize your workflow
- Evaluate model performance on the test set by plotting an ROC curve using `autoplot()` and calculating the area under the ROC curve on your test data

Data Splitting

We will split the data into a training and test set. The training data will be further divided into 5 folds for hyperparameter tuning.

```
set.seed(314)

cc_split <- initial_split(credit_card_df  , prop = 0.75,
                              strata = customer_status)

cc_training <- cc_split %>% training()

cc_test <- cc_split %>% testing()

# Create folds for cross validation on the training data set
## These will be used to tune model hyperparameters
set.seed(314)

cc_folds <- vfold_cv(cc_training, v = 5)
```

Feature Engineering

```
cc_recipe <- recipe(customer_status ~ ., data = cc_training) %>%
                        step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
                        step_normalize(all_numeric(), -all_outcomes()) %>%
                        step_dummy(all_nominal(), -all_outcomes())
```

Let's check to see if the feature engineering steps have been carried out correctly.

```
cc_recipe %>%
  prep(training = cc_training) %>%
  bake(new_data = NULL)
```

```
## # A tibble: 3,470 x 23
##        age dependents income months_since_firs~ total_accounts months_inactive_~
##      <dbl>      <dbl>  <dbl>              <dbl>          <dbl>              <dbl>
##  1 -0.437      -0.283 -0.984            -0.163          -0.379            -1.59
##  2 -0.0541      1.23   0.739            0.614           0.250            -0.413
##  3 -0.310       0.480 -0.807            -0.790           1.45              0.587
##  4 -0.945       0.480  1.26             -0.417           1.45              3.02
##  5  2.02       -0.283 -0.903            1.96            1.45             -0.413
##  6 -0.437       1.23  -0.796            -0.0357          1.45              2.27
##  7 -0.692       0.480 -1.00             -0.667          -1.04             -0.413
##  8 -1.07       -1.06   1.29             -1.40            0.250            -0.413
##  9  1.50       -1.87  -0.932            -0.0357          1.45              0.587
## 10  0.589      -1.87  -1.33             -0.163           0.859             0.587
## # ... with 3,460 more rows, and 17 more variables: 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_masters <dbl>,
## #   education_doctorate <dbl>, marital_status_married <dbl>,
## #   marital_status_divorced <dbl>, employment_status_part_time <dbl>,
## #   employment_status_self_employed <dbl>, card_type_silver <dbl>,
## #   card_type_gold <dbl>
```

Model 1 Decision Tree

```
tree_model <- decision_tree(cost_complexity = tune(),
                            tree_depth = tune(),
                            min_n = tune()) %>%
              set_engine('rpart') %>%
              set_mode('classification')
```

Workflow

```
tree_workflow <- workflow() %>%
                 add_model(tree_model) %>%
                 add_recipe(cc_recipe)
```

Hyperparameter Tuning

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

```
# View grid
tree_grid
```

```
## # A tibble: 8 x 3
##   cost_complexity tree_depth min_n
##             <dbl>      <int> <int>
## 1    0.0000000001          1     2
## 2    0.1                   1     2
## 3    0.0000000001         15     2
## 4    0.1                  15     2
## 5    0.0000000001          1    40
## 6    0.1                   1    40
## 7    0.0000000001         15    40
## 8    0.1                  15    40
```

```
tree_grid <- grid_regular(parameters(tree_model),
                          levels = 2)
tree_grid
```

```
## # A tibble: 8 x 3
##   cost_complexity tree_depth min_n
##             <dbl>      <int> <int>
## 1    0.0000000001          1     2
## 2    0.1                   1     2
## 3    0.0000000001         15     2
## 4    0.1                  15     2
## 5    0.0000000001          1    40
## 6    0.1                   1    40
## 7    0.0000000001         15    40
## 8    0.1                  15    40
```

Tuning Hyperparameters with `tune_grid()`

```
## Tune decision tree workflow
set.seed(314)

tree_tuning <- tree_workflow %>%
               tune_grid(resamples = cc_folds,
                         grid = tree_grid)
```

```
## Show the top 5 best models based on 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     0.953     5 0.00339
## 2    0.0000000001         15     2 roc_auc binary     0.908     5 0.00307
## 3    0.0000000001          1     2 roc_auc binary     0.765     5 0.00620
## 4    0.1                   1     2 roc_auc binary     0.765     5 0.00620
## 5    0.1                  15     2 roc_auc binary     0.765     5 0.00620
## # ... with 1 more variable: .config <chr>
```

```
## Select best model based on roc_auc
best_tree <- tree_tuning %>%
             select_best(metric = 'roc_auc')

# View the best tree parameters
best_tree
```

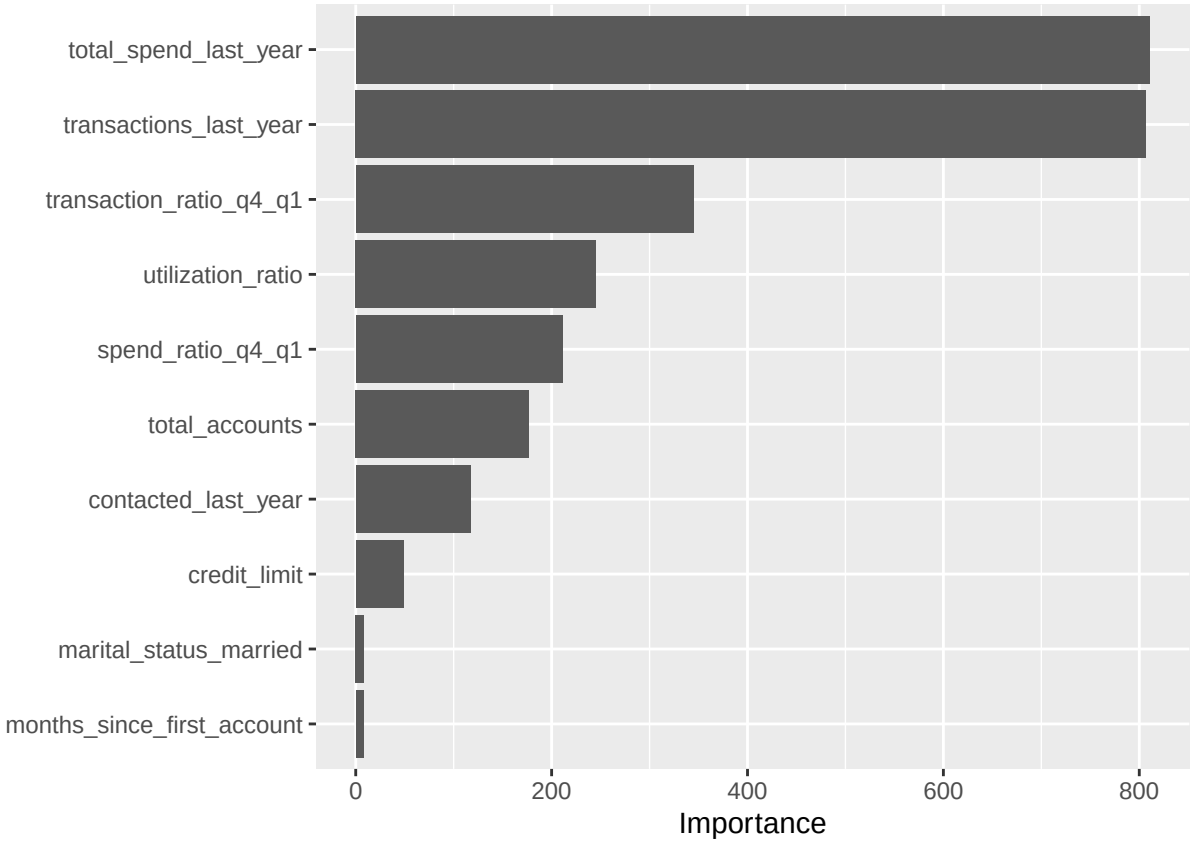
```
## # A tibble: 1 x 4
##   cost_complexity tree_depth min_n .config
##             <dbl>      <int> <int> <chr>
## 1    0.0000000001         15    40 Preprocessor1_Model7
```

Finalize Workflow

```
final_tree_workflow <- tree_workflow %>%
                       finalize_workflow(best_tree)
```

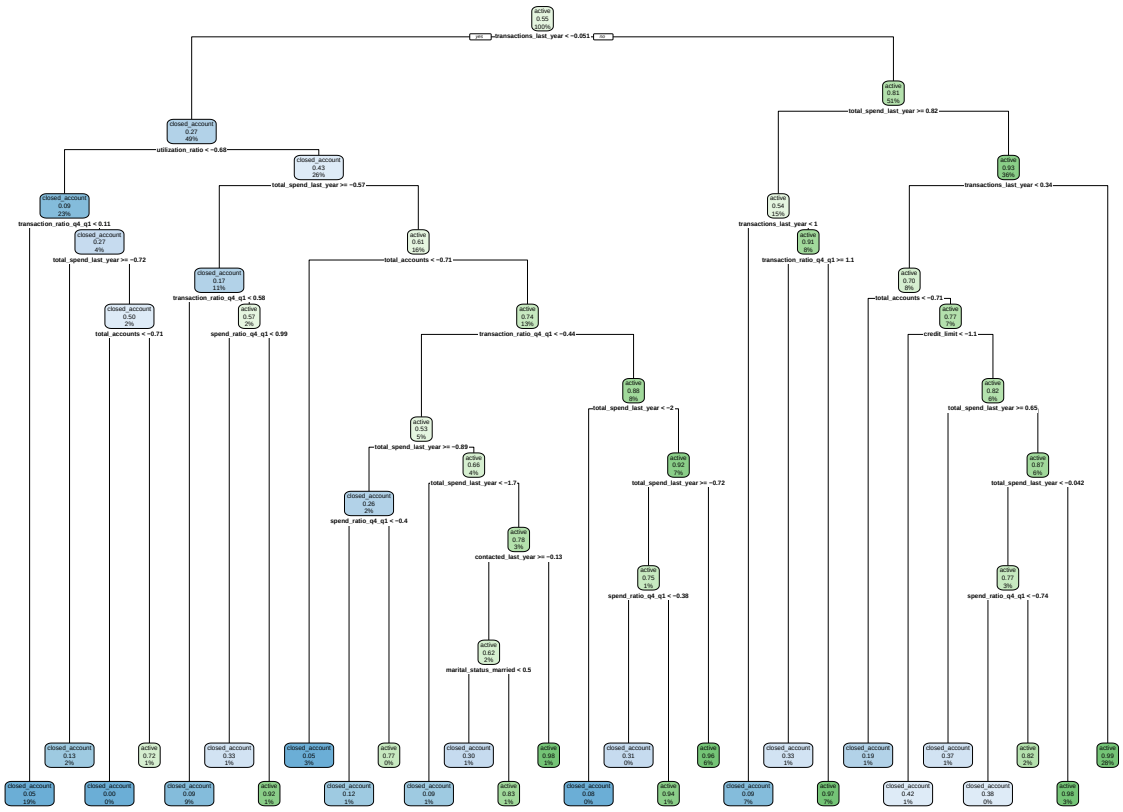
Visualize Results

```
tree_wf_fit <- final_tree_workflow %>%
               fit(data = cc_training)

tree_fit <- tree_wf_fit %>%
            extract_fit_parsnip()

vip(tree_fit)
```



Decision Tree Plot

```
library(rpart.plot)
rpart.plot(tree_fit$fit, roundint = FALSE)
```

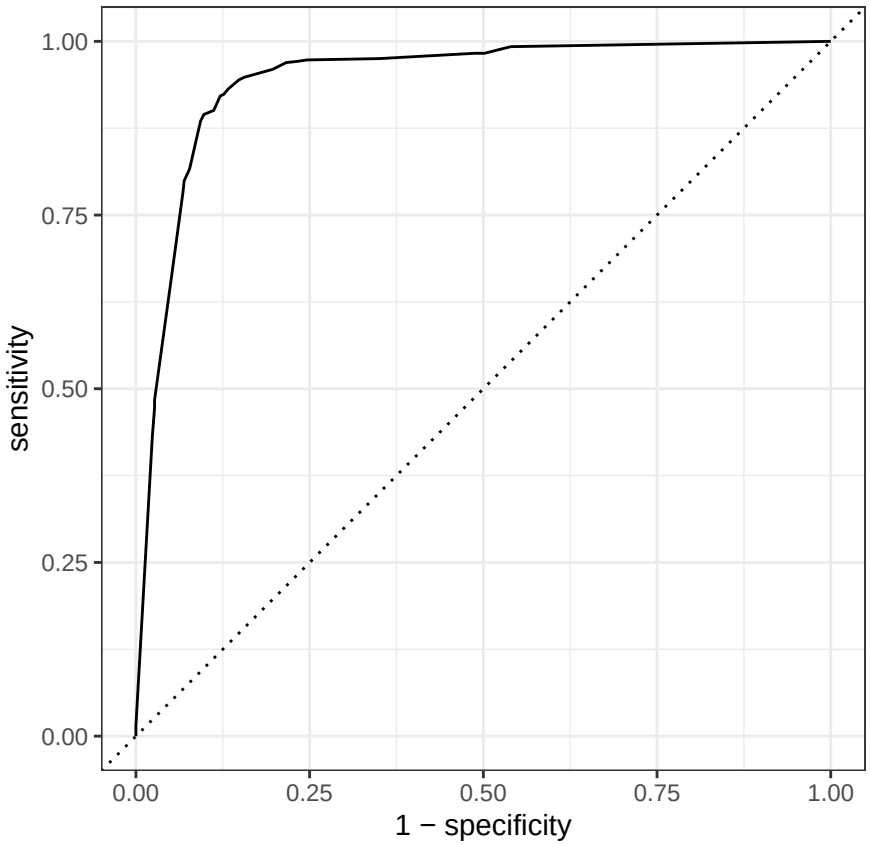



```
tree_last_fit <- final_tree_workflow %>%
                 last_fit(cc_split)

tree_last_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.896 Preprocessor1_Model1
## 2 roc_auc  binary         0.944 Preprocessor1_Model1
```

ROC Curve

```
tree_last_fit %>% collect_predictions() %>%
                 roc_curve(truth  = customer_status, estimate = .pred_closed_account) %>%
                 autoplot()
```



Confusion Matrix

We see that our model made 40 false negatives and 34 false positives on our test data set.

```
tree_predictions <- tree_last_fit %>% collect_predictions()

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

```
##                 Truth
## Prediction       closed_account active
##   closed_account            487     84
##   active                     36    550
```

Model 2 Regression Model

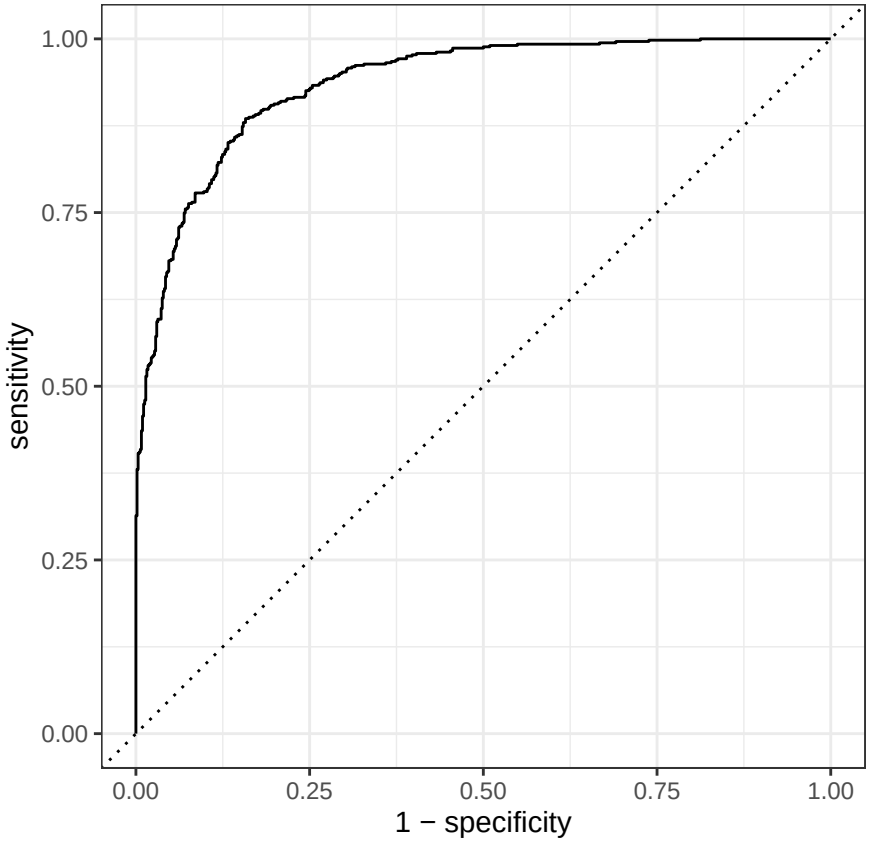
```
# Create cross validation folds for hyperparameter tuning
set.seed(314)

cc_folds <- vfold_cv(cc_training, v = 10)

# Feature Engineering
```

```
cc_recipe <- recipe(customer_status ~ ., data = cc_training) %>%
                step_YeoJohnson(all_numeric(), -all_outcomes()) %>%
                step_normalize(all_numeric(), -all_outcomes()) %>%
                step_dummy(all_nominal(), -all_outcomes())

# Feature Engineering
cc_recipe %>%
  prep(training = cc_training) %>%
  bake(new_data = NULL)
```

```
## # A tibble: 3,470 x 23
##        age dependents income months_since_firs~ total_accounts months_inactive_~
##      <dbl>      <dbl>  <dbl>              <dbl>          <dbl>             <dbl>
##  1 -0.437      -0.283 -0.984            -0.163          -0.379            -1.59
##  2 -0.0541      1.23   0.739            0.614           0.250            -0.413
##  3 -0.310       0.480 -0.807            -0.790          1.45              0.587
##  4 -0.945       0.480  1.26             -0.417          1.45              3.02
##  5  2.02       -0.283 -0.903            1.96            1.45             -0.413
##  6 -0.437       1.23  -0.796            -0.0357         1.45              2.27
##  7 -0.692       0.480 -1.00             -0.667          -1.04             -0.413
##  8 -1.07       -1.06   1.29             -1.40           0.250            -0.413
##  9  1.50       -1.87  -0.932            -0.0357         1.45              0.587
## 10  0.589      -1.87  -1.33             -0.163          0.859             0.587
## # ... with 3,460 more rows, and 17 more variables: 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_masters <dbl>,
## #   education_doctorate <dbl>, marital_status_married <dbl>,
## #   marital_status_divorced <dbl>, employment_status_part_time <dbl>,
## #   employment_status_self_employed <dbl>, card_type_silver <dbl>,
## #   card_type_gold <dbl>
```

```
# Specify Logistic Regression Model
logistic_model <- logistic_reg() %>%
                  set_engine('glm') %>%
                  set_mode('classification')

# Create a Workflow
logistic_wf <- workflow() %>%
               add_model(logistic_model) %>%
               add_recipe(cc_recipe)
# Fit Model
logistic_fit <- logistic_wf %>%
                last_fit(split = cc_split)

# Collect Predictions

logistic_results <-  logistic_fit %>%
                     collect_predictions()

roc_curve(logistic_results,
          truth = customer_status,
```

```
          estimate = .pred_closed_account) %>%
  autoplot()
```




```
# ROC AUC
roc_auc(logistic_results,
        truth = customer_status,
        .pred_closed_account)
```

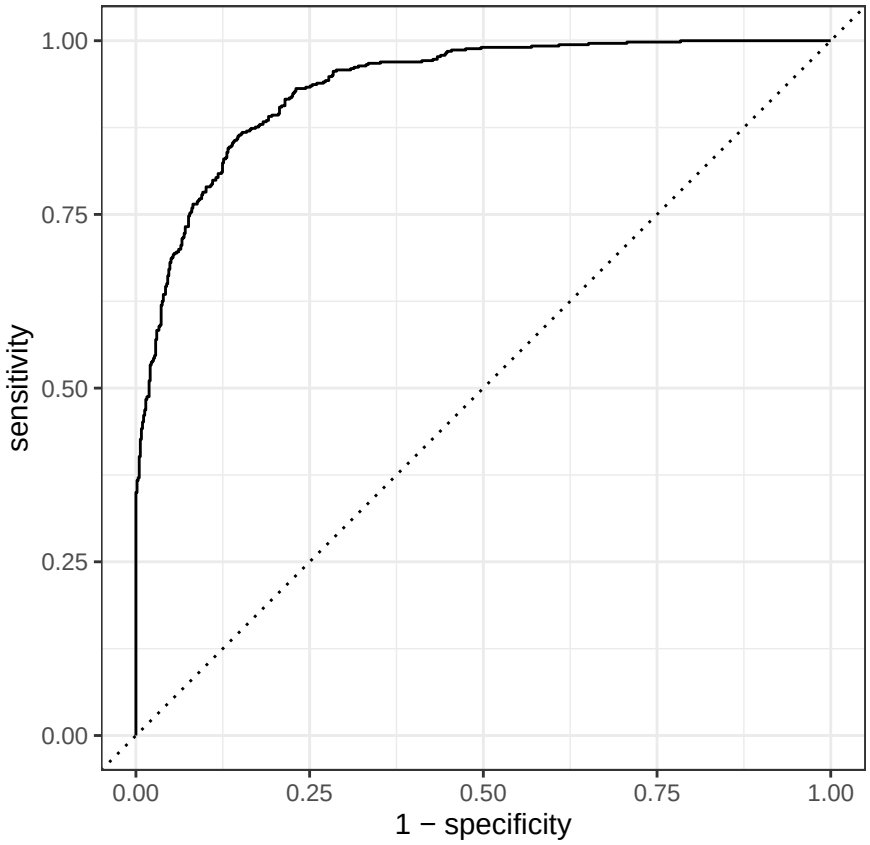
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.936
```

```
# Confusion Matrix
conf_mat(logistic_results,
         truth = customer_status,
         estimate = .pred_class)
```

```
##                 Truth
## Prediction       closed_account active
##   closed_account            444     84
##   active                     79    550
```

Model 3 Specify LDA model

```
lda_model <- discrim_regularized(frac_common_cov = 1) %>%
             set_engine('klaR') %>%
             set_mode('classification')

lda_wf <- workflow() %>%
          add_model(lda_model) %>%
          add_recipe(cc_recipe)

lda_fit <- lda_wf %>%
           last_fit(split = cc_split)

lda_results <-  lda_fit %>%
                collect_predictions()

roc_curve(lda_results,
          truth = customer_status,
          estimate = .pred_closed_account) %>%
  autoplot()
```




```
roc_auc(lda_results,
        truth = customer_status,
        .pred_closed_account)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.935
```

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

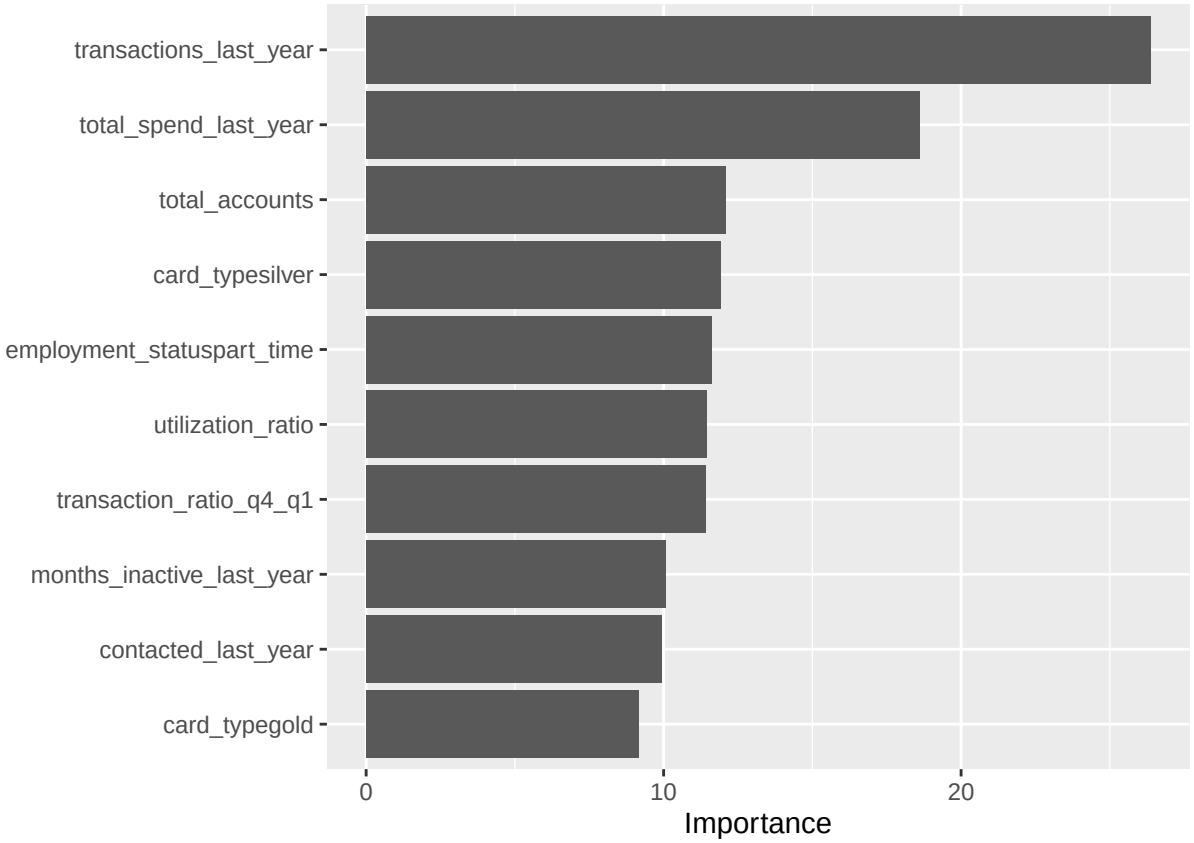
```
##                 Truth
## Prediction       closed_account active
##   closed_account            434     82
##   active                     89    552
```

```
model <- glm( customer_status ~., data = credit_card_df, family = binomial)
tidymodel <- tidy(model)
summary(model)
```

```
##
## Call:
## glm(formula = customer_status ~ ., family = binomial, data = credit_card_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.98101  -0.43299   0.09365   0.42078   3.14316
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -7.876e+00  4.892e-01 -16.099  < 2e-16 ***
## age                             6.563e-03  9.558e-03   0.687 0.492277
## dependents                     -1.354e-01  3.646e-02  -3.714 0.000204 ***
## educationbachelors              8.518e-02  1.664e-01   0.512 0.608792
## educationmasters                7.951e-02  1.015e-01   0.783 0.433591
## educationdoctorate             -1.164e-01  2.181e-01  -0.534 0.593430
## marital_statusmarried           6.035e-01  9.955e-02   6.062 1.34e-09 ***
## marital_statusdivorced          2.341e-01  1.897e-01   1.234 0.217089
## employment_statuspart_time     -1.216e+00  1.046e-01 -11.623  < 2e-16 ***
## employment_statusself_employed  4.253e-01  1.366e-01   3.113 0.001852 **
## income                          3.337e-06  1.726e-06   1.933 0.053190 .
## card_typesilver                 1.413e+00  1.186e-01  11.916  < 2e-16 ***
## card_typegold                   1.138e+00  1.242e-01   9.165  < 2e-16 ***
## months_since_first_account      5.858e-03  9.530e-03   0.615 0.538766
## total_accounts                  3.904e-01  3.228e-02  12.096  < 2e-16 ***
## months_inactive_last_year      -5.043e-01  5.010e-02 -10.064  < 2e-16 ***
## contacted_last_year            -4.412e-01  4.448e-02  -9.920  < 2e-16 ***
## credit_limit                    3.259e-05  6.628e-06   4.917 8.77e-07 ***
## utilization_ratio               2.174e+00  1.899e-01  11.449  < 2e-16 ***
## spend_ratio_q4_q1               8.649e-01  2.297e-01   3.765 0.000166 ***
## total_spend_last_year          -5.463e-04  2.935e-05 -18.610  < 2e-16 ***
## transactions_last_year          1.281e-01  4.859e-03  26.357  < 2e-16 ***
## transaction_ratio_q4_q1         2.571e+00  2.254e-01  11.406  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6371.9  on 4626  degrees of freedom
## Residual deviance: 2968.3  on 4604  degrees of freedom
## AIC: 3014.3
##
## Number of Fisher Scoring iterations: 6
```

```
vip(model)
```



Summary of Results

Write a summary of your overall findings and recommendations to the executives at the bank. Think of this section as your closing remarks of a presentation, where you summarize your key findings, model performance, and make recommendations to improve customer retention and service at the bank.

Your executive summary must be written in a professional tone, with minimal grammatical errors, and should include the following sections:

1. An introduction where you explain the business problem and goals of your data analysis
    - What problem(s) is this company trying to solve? Why are they important to their future success?

- What was the goal of your analysis? What questions were you trying to answer and why do they matter?

2. Highlights and key findings from your Exploratory Data Analysis section

   - What were the interesting findings from your analysis and **why are they important for the business**?
   - This section is meant to **establish the need for your recommendations** in the following section

3. Your "best" classification model and an analysis of its performance

   - In this section you should talk about the expected error of your model on future data
     - To estimate future performance, you can use your model performance results on the **test data**
   - You should discuss at least one performance metric, such as an F1, sensitivity, specificity, or ROC AUC for your model. However, you must explain the results in an **intuitive, non-technical manner**. Your audience in this case are executives at a telecommunications company with limited knowledge of machine learning.

4. Your recommendations to the company on how to reduce customer attrition rates

   - Each recommendation must be supported by your data analysis results
   - You must clearly explain why you are making each recommendation and which results from your data analysis support this recommendation
   - You must also describe the potential business impact of your recommendation:
     - Why is this a good recommendation?
     - What benefits will the business achieve?

Summary Add your summary here. Please do not place your text within R code chunks.

1. An introduction where you explain the business problem and goals of your data analysis

This company is trying to find out the reason why people closed their credit card account. It is very important because the business can improve their weakness/business strategies to keep the clients in business, so the bank can make more profit. My goal for this project is to help this company find the key reason and provide useful suggestions to help to reduce the rate of customer lose.

2, Highlights and key findings from your Exploratory Data Analysis section.

Here are some highlights and key findings from my analysis. First, based on the average spend ratio for people who closed the account is 0.69 lower than people who kept the account 0.77 . The average total spent last year for people who closed the account is around $3121 and the average total spend last year for people who did not close the account is around $4597. Second, avg_transaction_ratio is a key factor for customer status. As we can see that the avg_transaction_ratio for people who closed the account is 0.56 it is much less than the number of people who kept the account 0.74. Third, the relationship between customer status and card_type, according to the number I have, more than half of people that hold a blue card closed their account, and they all have low utilization ratio, spend ratio, and transaction ratio. More than half of the people that hold silver and gold cards chose to stay active.

3, Your "best" classification model and an analysis of its performance

I have built a Decision Tree, Logistic Regression, and LDA model to find the most significant factors by using ROC AUC to measure the accuracy of the model. The most accurate model has a higher ROC AUC to find the factors that have the most serious impact on credit card closure. The ROC AUC of the Decision Tree

model is 94.41677%, The ROC AUC of the Logistic Regression model is 93.60128%. The ROC AUC of the LDA model is 93.4915%. Therefore, the Decision Tree model is the best model. There are also some significant factors that have arisen from this model. As we can tell from the VIP function, there are 4 factors that affect users in closing their account. The 4 factors include total_spend_last_year, transactions_last_year, transaction_ratio_q4_q1, and utilization_ratio. Among the 4 factors, total_spend_last_year and transactions_last_year are the most important factors.

4,Your recommendations to the company on how to reduce customer attrition rates Based on the Decision Tree model and the analysis I did on the dataset, here are some recommendations to the bank on how to reduce customer account closure rate.

(1). total_spend_last_year and transactions_last_year are the most important factors based on the Decision Tree model. Combining the decision tree model and the analysis we did, the average total amount for customers who did not close the account is around $4600, and the average total amount for customers who did close the account is around $3100. So I suggest that the company should offer a promotion that applies to the level of spend amount during the year. for example, the customer who spends more than 5k a year in travel and restaurants will reward a personal gift and free upgraded card.

(2). transaction_ratio is another important factor that is a strong relationship between customer status and transaction_ratio. From the analysis above we know that people who are more frequently using the card are less likely to close the account. The average transaction ratio is around 0.74 for customers who did not close account, so the company needs to find the customers that have lower transaction ratio than 0.74 to encourage them to use the card more often. The bank can offer more reward points, shopping discounts, etc to those customers.

(3). Employee status is a very important factor as well. From the analysis above we see that customers whose employment status is part-time is more likely to close the account. So the bank should focus on these customers, by offering a credit card that has a bigger reward point system on essential need transactions such as grocery shopping, restaurants, and gas. On the other hand, the bank also needs to improve the service of the customer who has employment status on full-time and self-employed to keep them to stable, by offering a credit card that has exclusive service such as personal concierge program, access to airport VIP lounges, complimentary memberships on cooperative merchants, etc.

(4). Card type is the key factor to customer status that I found in the analysis. As I mentioned before, the customer who has a blue card is more likely to close the account and the customer who holds the silver card and gold card is more willing to stay. The data shows in a 2500 scale blue card type customer, almost 1500 customers closed the account which is around 58%, and compared to silver and gold card type, the account closure rate is 25% and 32%. That's a huge difference! The company needs to inform the customer by sending an email to encourage blue cardholders to upgrade their card to silver levels.