# Machine Learning Project

**Sai Krishna Kaushik Pinnelli G01227643**

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

library(tidyverse)
library(tidymodels)
library(dplyr)
library(ggplot2)

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

view(credit_card_df)
```

## Data Analysis

## Question 1

**Question**:

What are the factors that are associated with customers closing their credit card accounts?

**Answer**: The factors associated with the customers closing their credit card account are total_spend_last_year, transactions_last_year, transaction_ratio, utilization_ratio, spend_ratio, total_accounts, credit_limit, employment_status_part_time, months_inactive_last_year and age.

## Question 2

**Question**:

How is the total spend trend of all the closed and active accounts?

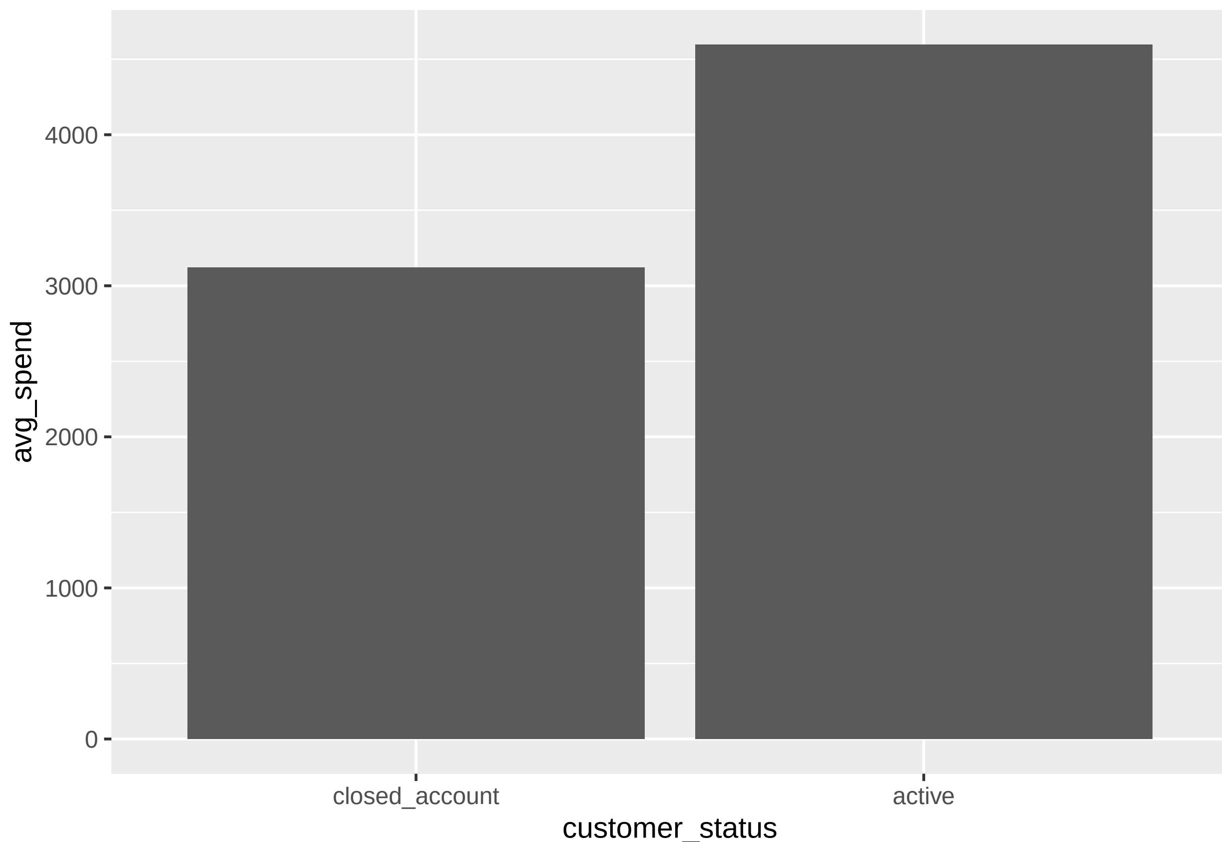
**Answer**: In total, there are 2092 closed accounts with an average spend of 3120 and 2535 active accounts with average spend of 4597 last year

```
spend_c_status <- credit_card_df %>% group_by(customer_status) %>%
    summarise(n_customers = n(),
              min_spend = min(total_spend_last_year),
              avg_spend = mean(total_spend_last_year),
              max_spend = max(total_spend_last_year),
              sd_spend = sd(total_spend_last_year))

spend_c_status
```

```
## # A tibble: 2 x 6
##   customer_status n_customers min_spend avg_spend max_spend sd_spend
##   <fct>                 <int>     <dbl>     <dbl>     <dbl>    <dbl>
## 1 closed_account         2092       510     3121.     10583    2345.
## 2 active                 2535       893     4597.     17498    3465.
```

```
ggplot(spend_c_status, aes(x= customer_status, y = avg_spend))+
         geom_col()
```




## Question 3

**Question**:

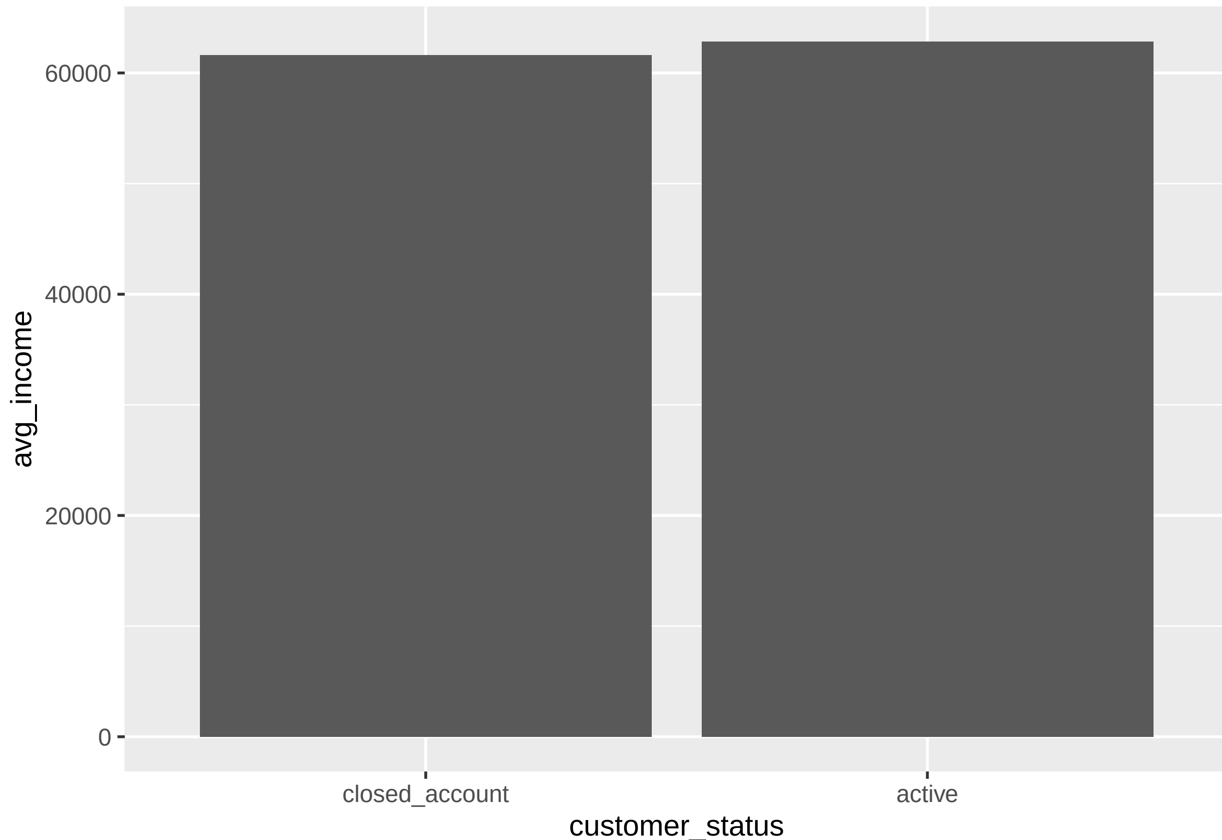
What is the average income of the customers with closed and active accounts?

**Answer**: The average income of the customers who closed their accounts was close to 61602 and that of active customers was 62843.

```
income <- credit_card_df %>% group_by(customer_status) %>%
    summarise(n_customers = n(),
              min_income = min(income),
              avg_income = mean(income),
              max_income = max(income),
              sd_income = sd(income))
income
```

```
## # A tibble: 2 x 6
##   customer_status n_customers min_income avg_income max_income sd_income
##   <fct>                 <int>      <dbl>      <dbl>      <dbl>     <dbl>
## 1 closed_account         2092      30198     61602.     168522    33659.
## 2 active                 2535      30094     62843.     166225    33306.
```

```
ggplot(income, aes(x= customer_status, y = avg_income))+
         geom_col()
```




## Question 4

**Question**:

What is the average age of the customers who have decided to close thier accounts?

**Answer**: From the data frame, we can find that most of the customers who chose to close their accounts are at an average age of 46 years.

```
age <- credit_card_df %>% group_by(customer_status) %>%
    summarise(n_customers = n(),
              min_age = min(age),
              avg_age = mean(age),
              max_age = max(age),
              sd_age = sd(age))
age
```

```
## # A tibble: 2 x 6
##   customer_status n_customers min_age avg_age max_age sd_age
##   <fct>                 <int>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 closed_account         2092      26    46.6      68   7.66
## 2 active                 2535      26    46.2      70   8.00
```

# Question 5

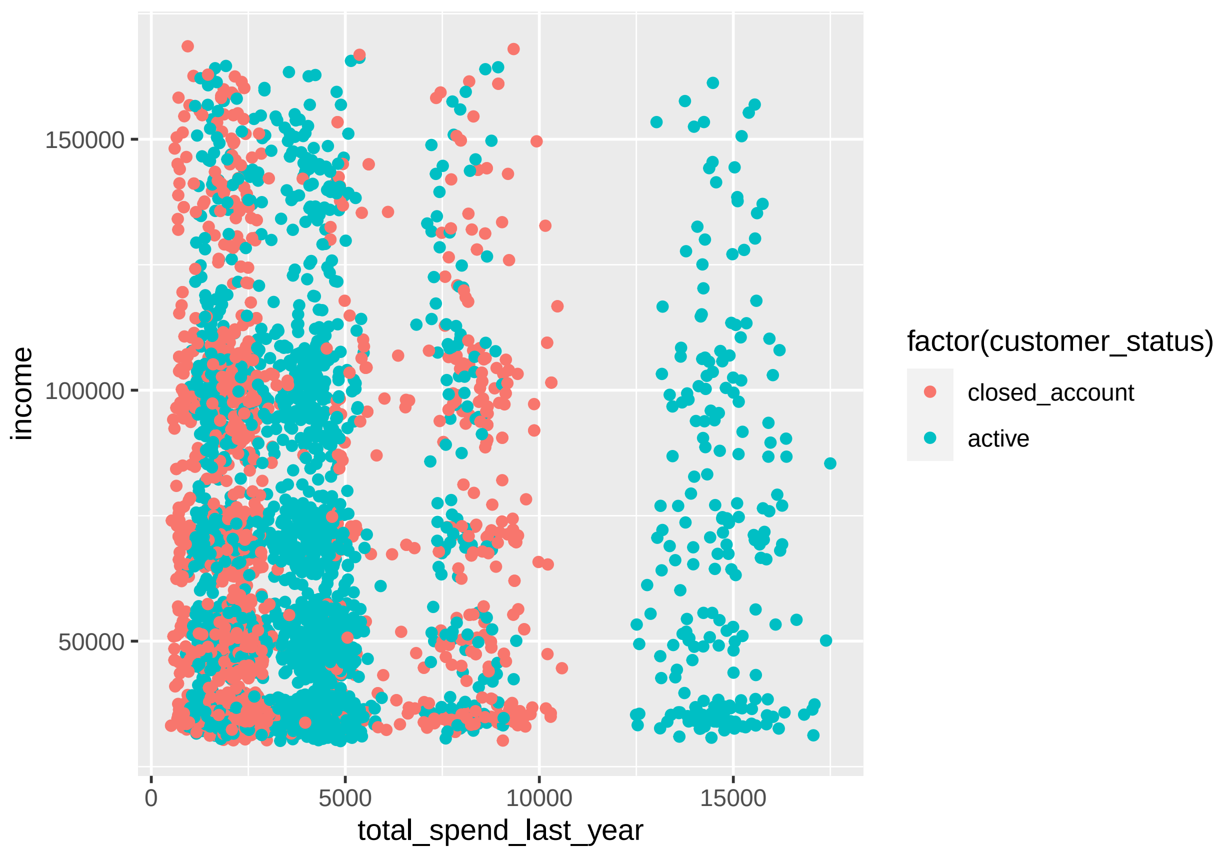
**Question**:

Is there any correlation between the total_spend_last_year and the Income of the customers with respect to the customer status?

**Answer**:

We can get the relation for the total_spend_last_year and income of the customer using the scatter plot and correlation plot. From the scatter plot, we can see that there is random relation between total_spend_last_year and income. By calculating the correlation, we get the value to be 0.084.

```
ggplot(credit_card_df,aes(x=total_spend_last_year,y=income))+
    geom_point(aes(color=factor(customer_status)))
```




```
cor(credit_card_df$income,credit_card_df$total_spend_last_year)
```

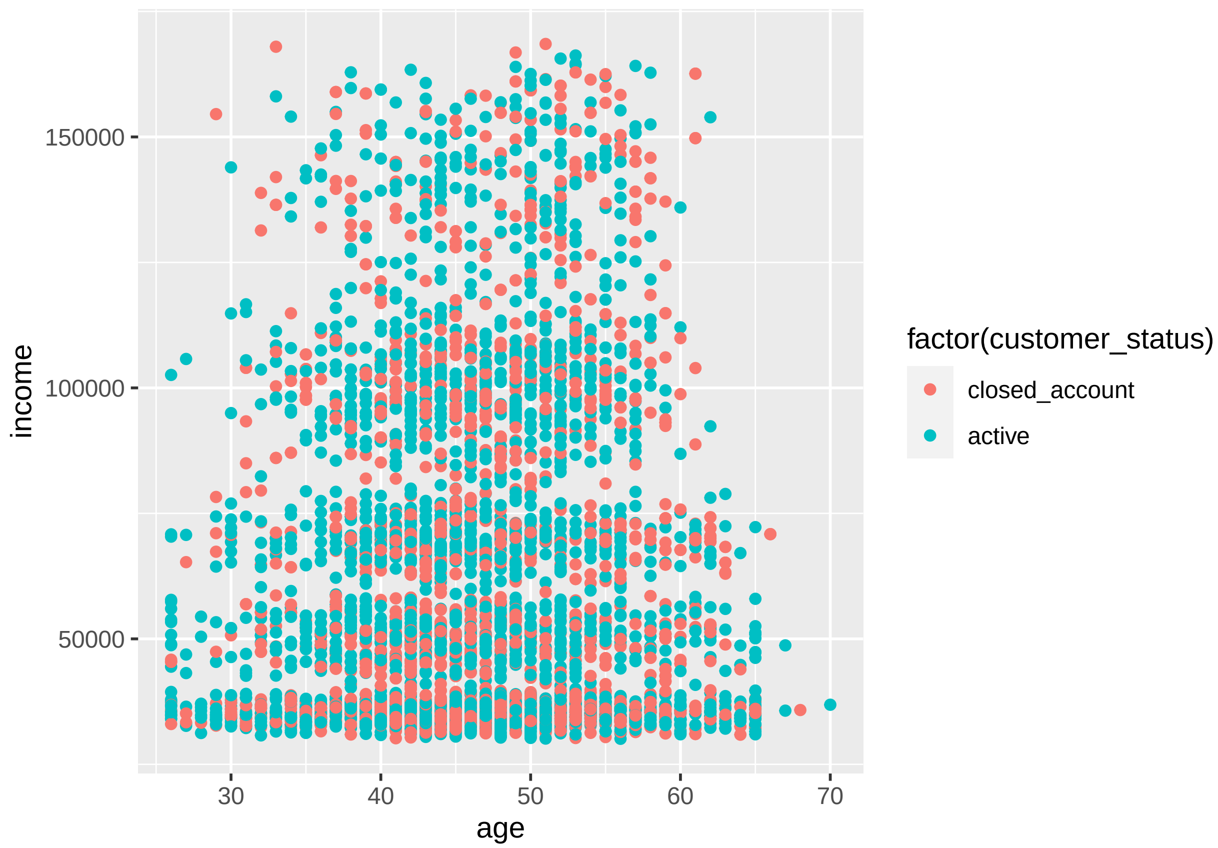
```
## [1] 0.08413525
```

# Question 6

**Question**: Is there any correlation between the age and the Income of the customers with respect to the customer status?

**Answer**: Again the answer is No as the correlation value is 0.032.

```
ggplot(credit_card_df,aes(x=age,y=income))+
    geom_point(aes(color=factor(customer_status)))
```




```
cor(credit_card_df$age,credit_card_df$income)
```

```
## [1] 0.03268417
```

# Question 7

**Question**: Is there any kind of relation between the customers expenditure in the previous year and the customers closing accounts?

**Answer**: Yes, the data indicates that customers who close the account tend to have lower average expenditure when compared to customers who do not. Among the 2092 employees that closed the account, the average expenditure was $3121. Also, Minimum expenditure was only 510 for people who have closed their accounts which is ~300 less than the active customers.

Among the customers who did not close the account, only ~32% have a total spend that is less than or equal to the average expenditure of closed account customers. When looking at customers who did close the account, this increases to ~80%.

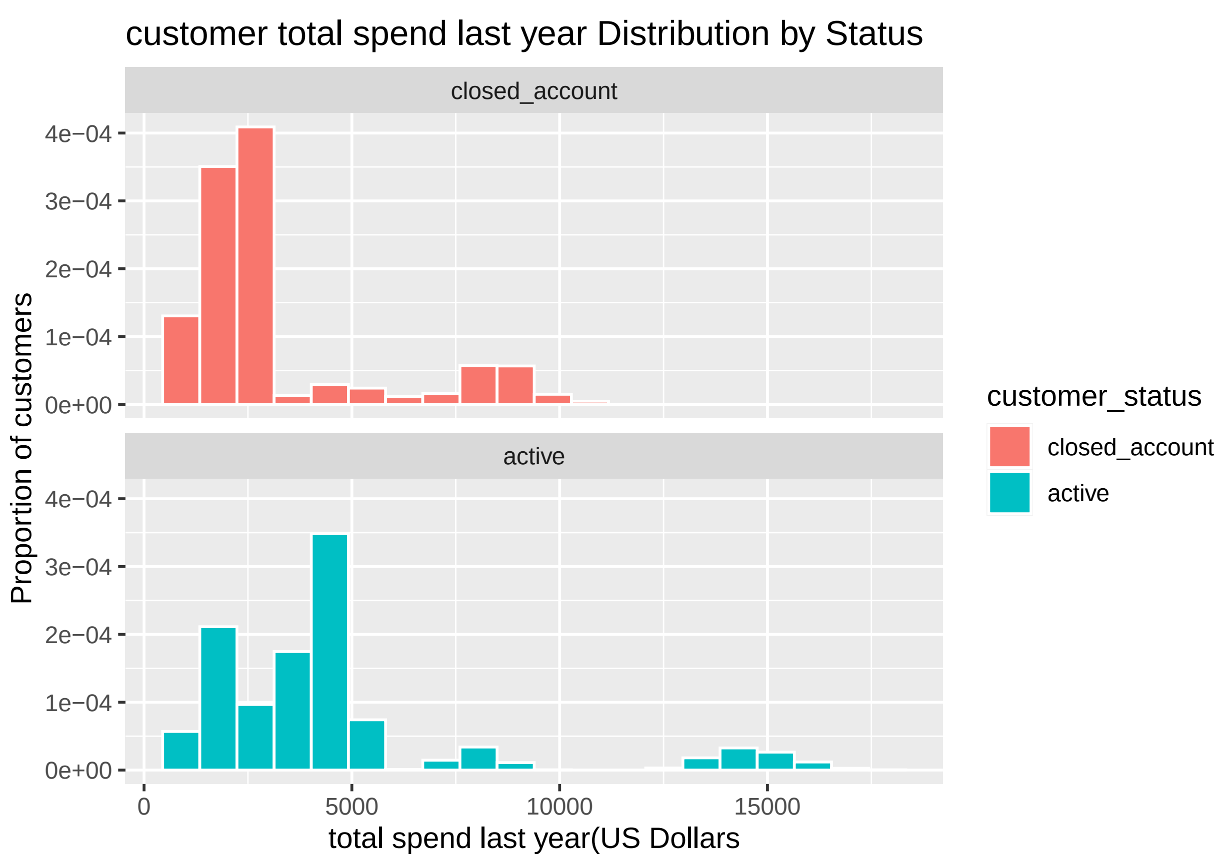
~~~Age: As seen above the age distribution is similar in both closed and active accounts. The mean age is about ~46.

~~~Utilization ratio: We can see that, the utilization ratio for most of the people who closed their accounts is ~ 0. Hence identifying these customers with 0 utilization ratio and asking business team of the company to take precautions will help the company not lose the customer.

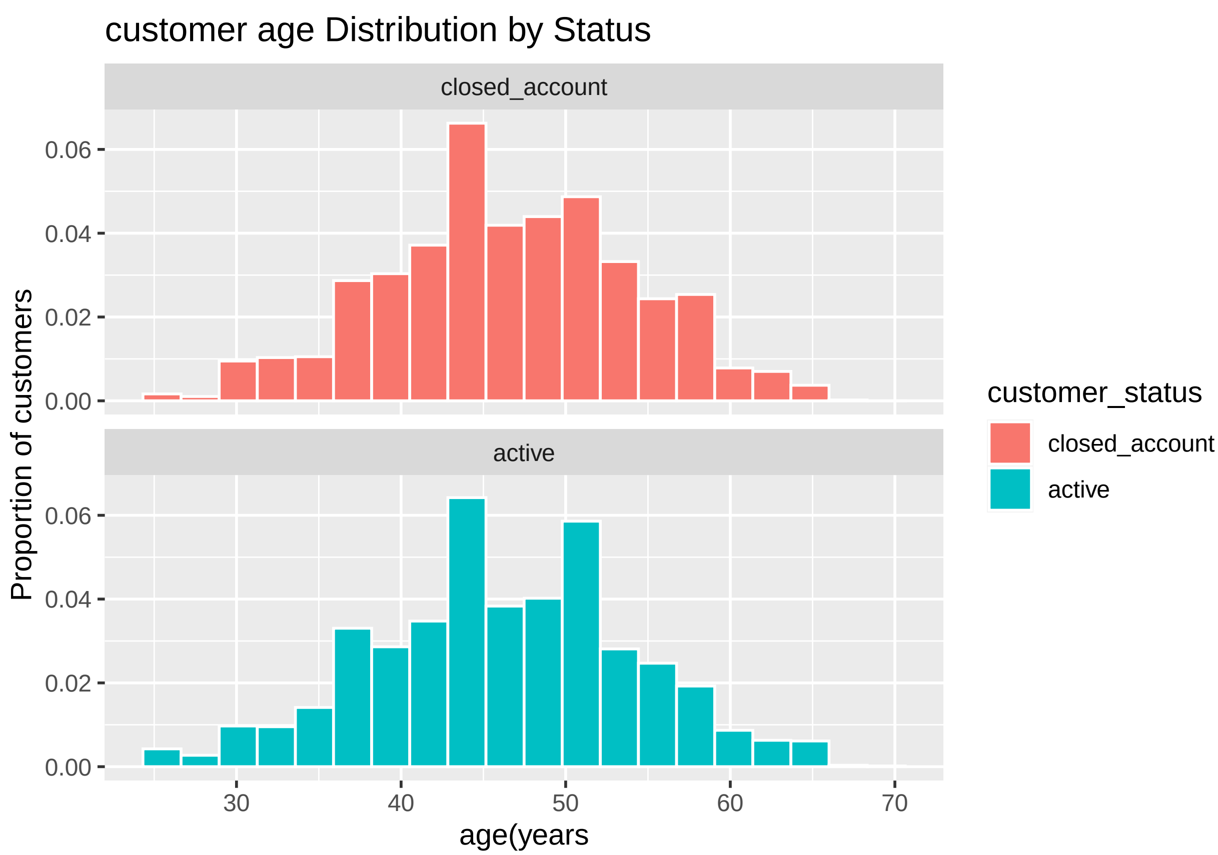
~~~Transaction last year: We can see that, the transactions last year for most of the people who closed their accounts is in range of 0 to 55 approximately. Hence identifying these customers will help the business team to take necessary steps to prevent these customers from closing the accounts and help the company profit.

~~~Transaction ratio last quarter: We can see that, the transaction ratio for most of the people who closed their accounts is ~ 0. Hence identifying these customers with 0 transaction ratio and asking business team of the company to take precautions will help the company not lose the customer.

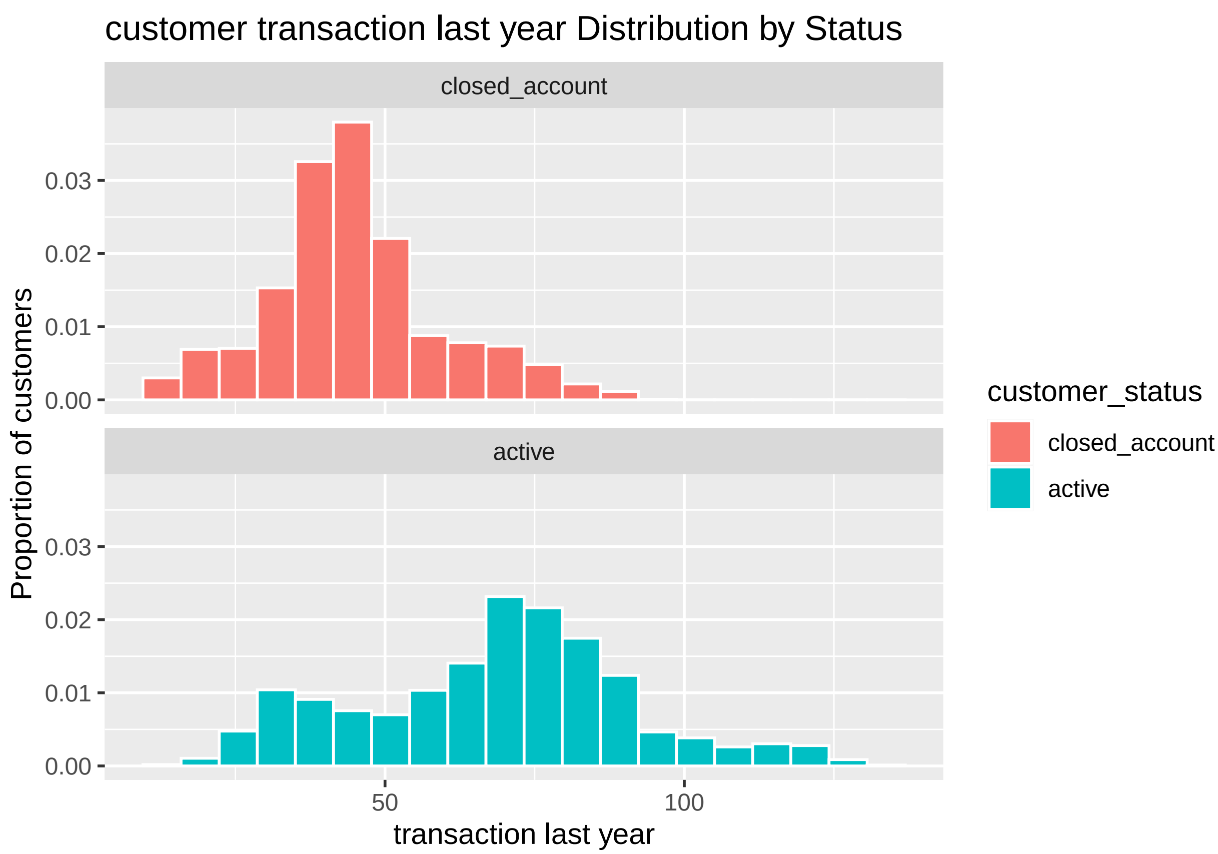
```
total_spend <- ggplot(data = credit_card_df, aes(x = total_spend_last_year, fill = customer_status)) +
    geom_histogram(aes(y = ..density..), color = "white", bins = 20) +
    facet_wrap(~customer_status, nrow = 2) +
    labs(title = "customer total spend last year Distribution by Status",
         x = "total spend last year(US Dollars", y = "Proportion of customers")
total_spend
```



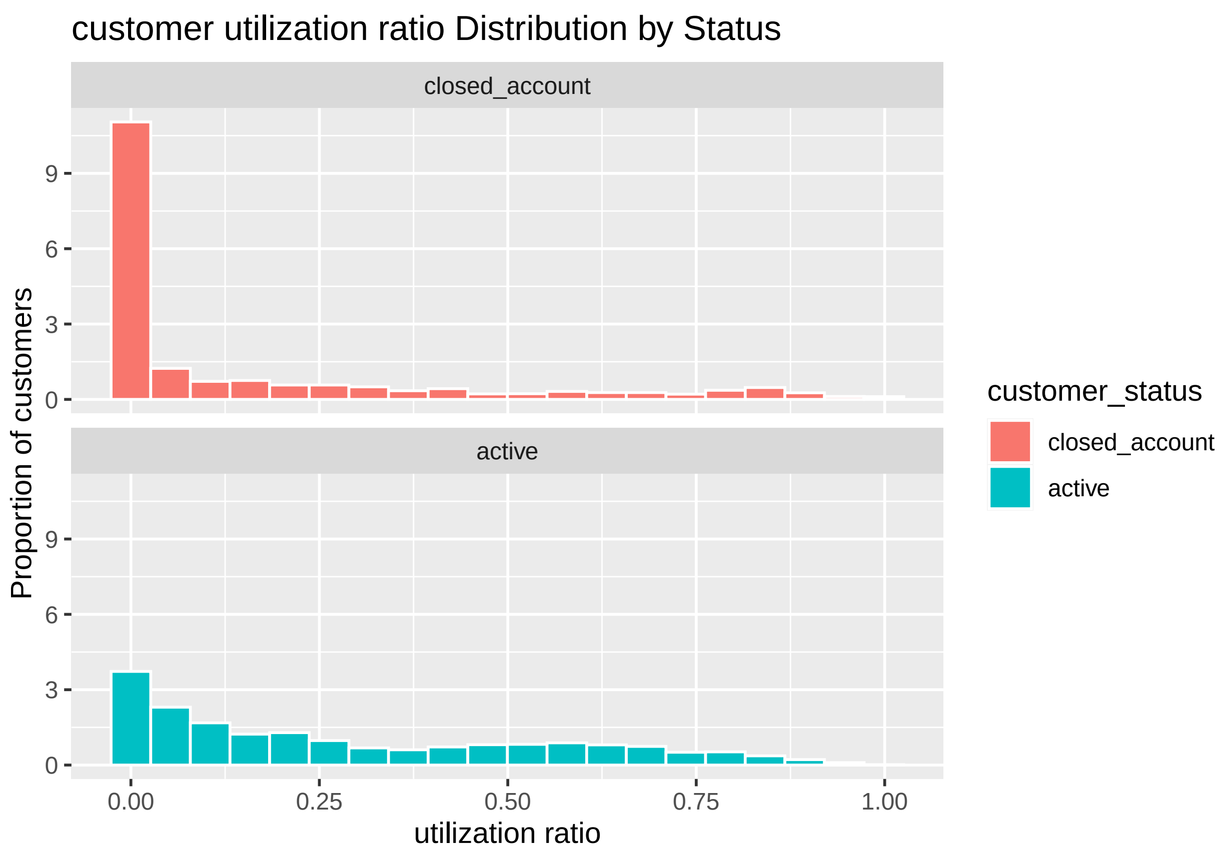
```r
age_s <- ggplot(data = credit_card_df, aes(x = age, fill = customer_status)) +
    geom_histogram(aes(y = ..density..), color = "white", bins = 20) +
    facet_wrap(~customer_status, nrow = 2) +
    labs(title = "customer age Distribution by Status",
         x = "age(years", y = "Proportion of customers")
age_s
```



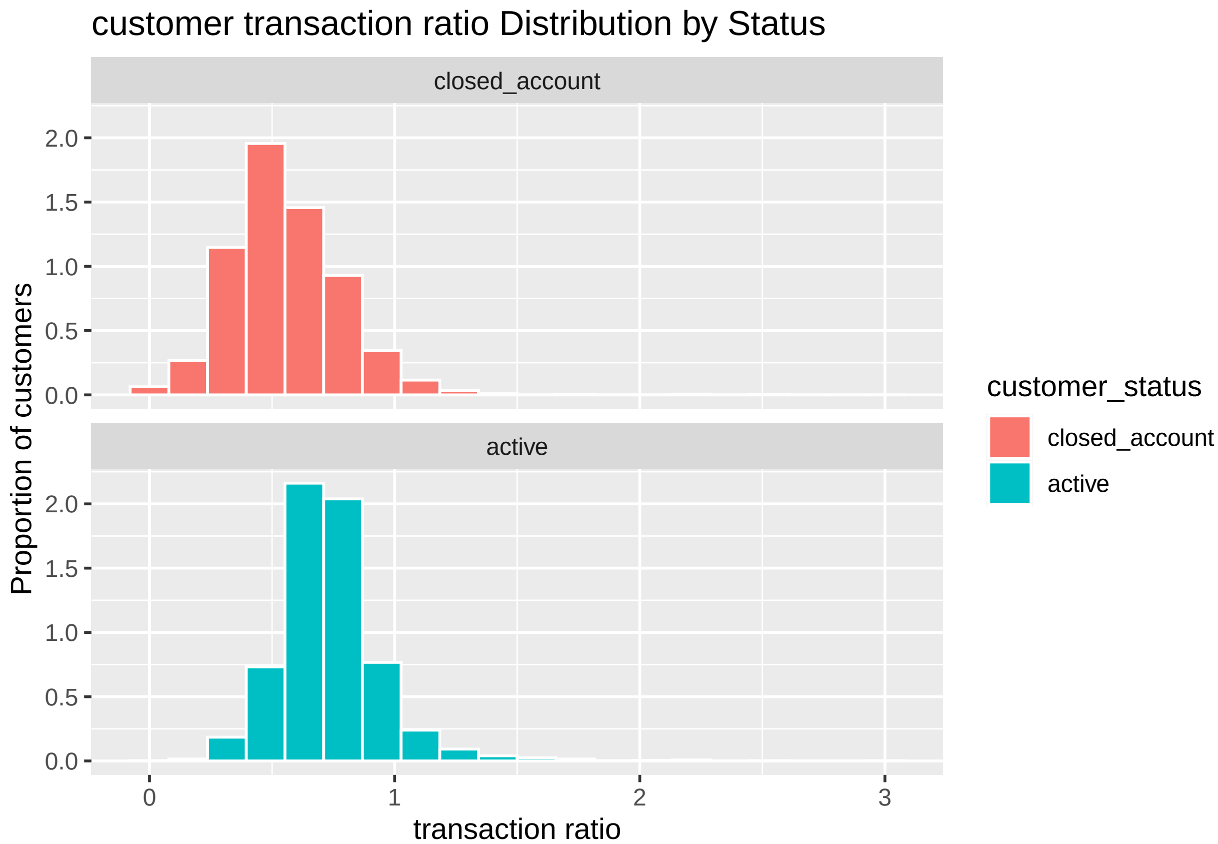

```r
tran_l_year <- ggplot(data = credit_card_df, aes(x = transactions_last_year, fill = customer_status)) +
    geom_histogram(aes(y = ..density..), color = "white", bins = 20) +
    facet_wrap(~customer_status, nrow = 2) +
    labs(title = "customer transaction last year Distribution by Status",
         x = "transaction last year", y = "Proportion of customers")
tran_l_year
```




```
util_ratio <- ggplot(data = credit_card_df, aes(x = utilization_ratio, fill = customer_status)) +
    geom_histogram(aes(y = ..density..), color = "white", bins = 20) +
    facet_wrap(~customer_status, nrow = 2) +
    labs(title = "customer utilization ratio Distribution by Status",
         x = "utilization ratio", y = "Proportion of customers")
util_ratio
```




```r
tran_ratio <- ggplot(data = credit_card_df, aes(x = transaction_ratio_q4_q1, fill = customer_status)) +
    geom_histogram(aes(y = ..density..), color = "white", bins = 20) +
    facet_wrap(~customer_status, nrow = 2) +
    labs(title = "customer transaction ratio Distribution by Status",
         x = "transaction ratio", y = "Proportion of customers")
tran_ratio
```




# Question 8

**Question**:

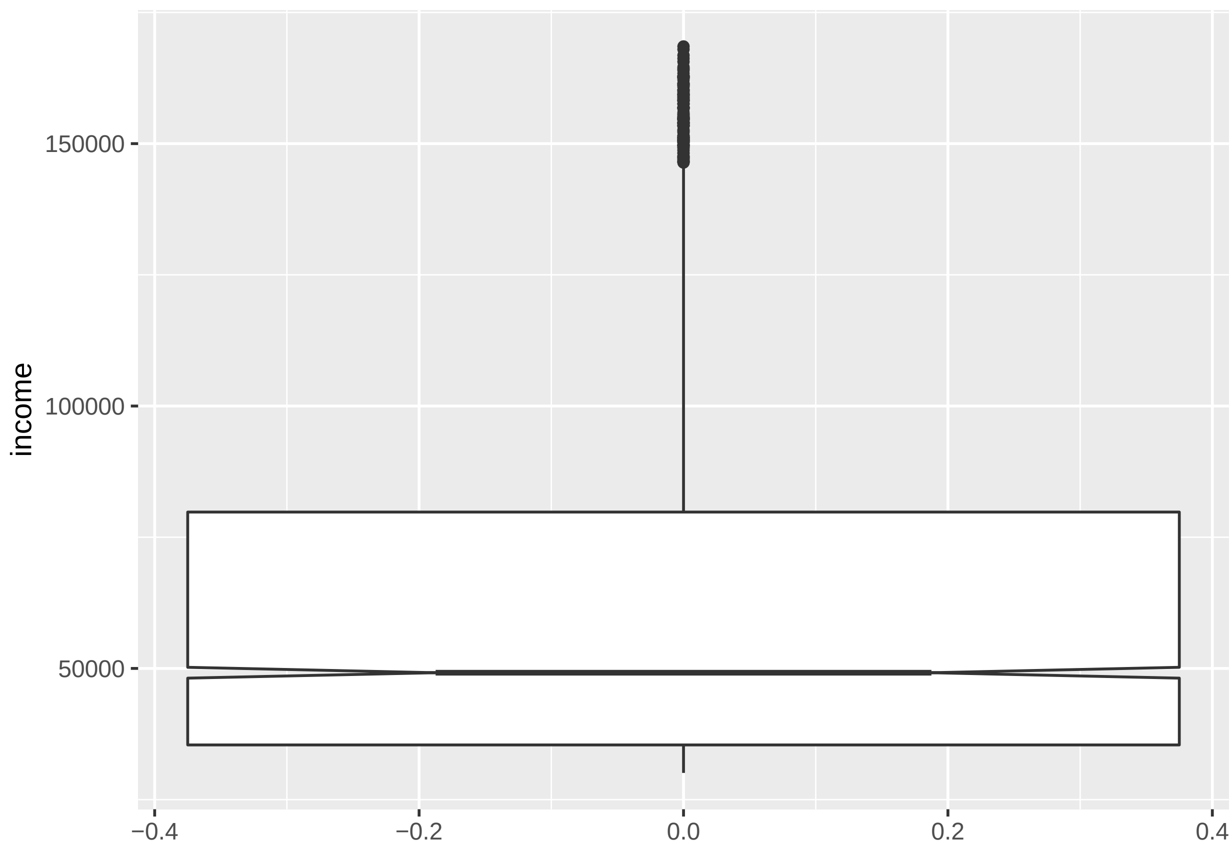
Are there any outliers in the credit card data when total spend last year and income is considered?

**Answer**:

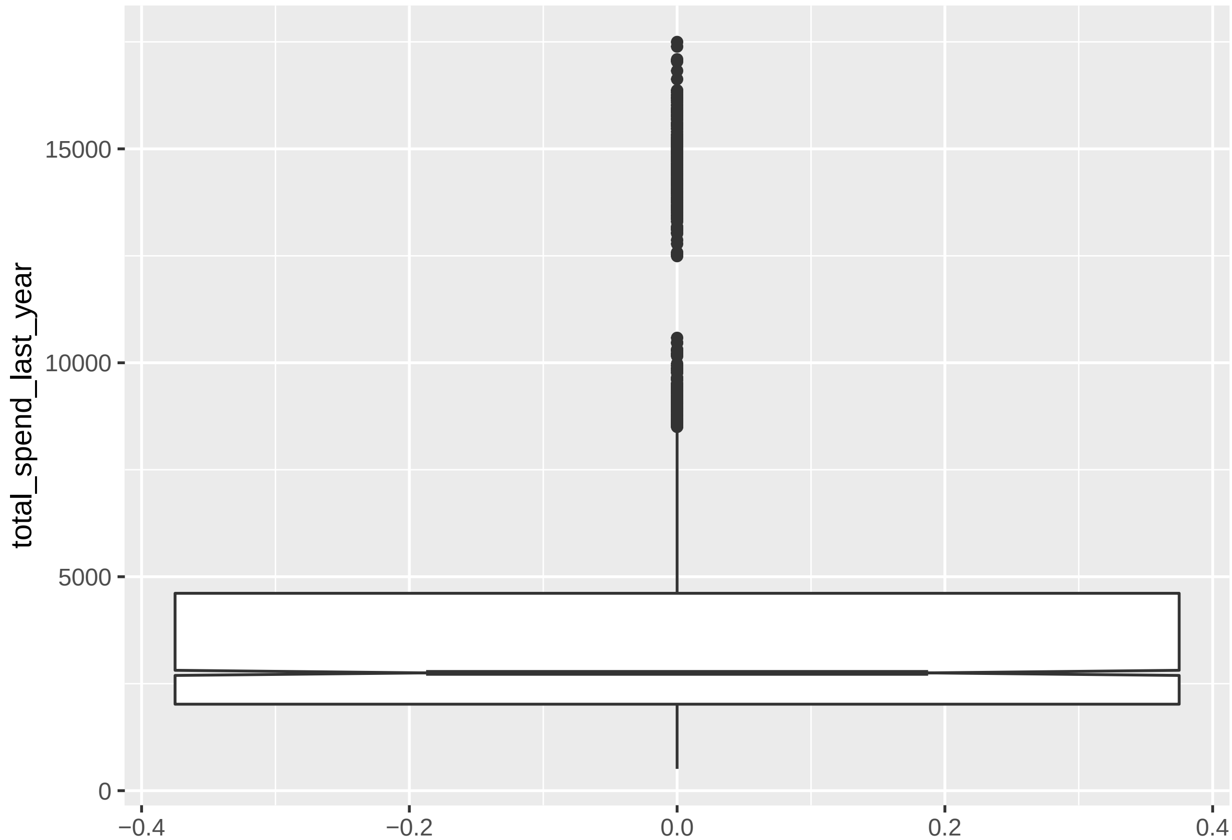
~~~Income: Yes, the boxplot generated from the income data, shows that there are many outliers in the data when considered income.

~~~total_spend_last_year: Yes, the boxplot generated from the total spend last year data, shows that there are many outliers in the data when considered total spend last year.

```
box1 <- ggplot(credit_card_df, aes( y=income)) +
    geom_boxplot(notch=TRUE)
box1
```




```
box2 <- ggplot(credit_card_df, aes( y=total_spend_last_year)) +
    geom_boxplot(notch=TRUE)
box2
```




# Question 9

**Question**:

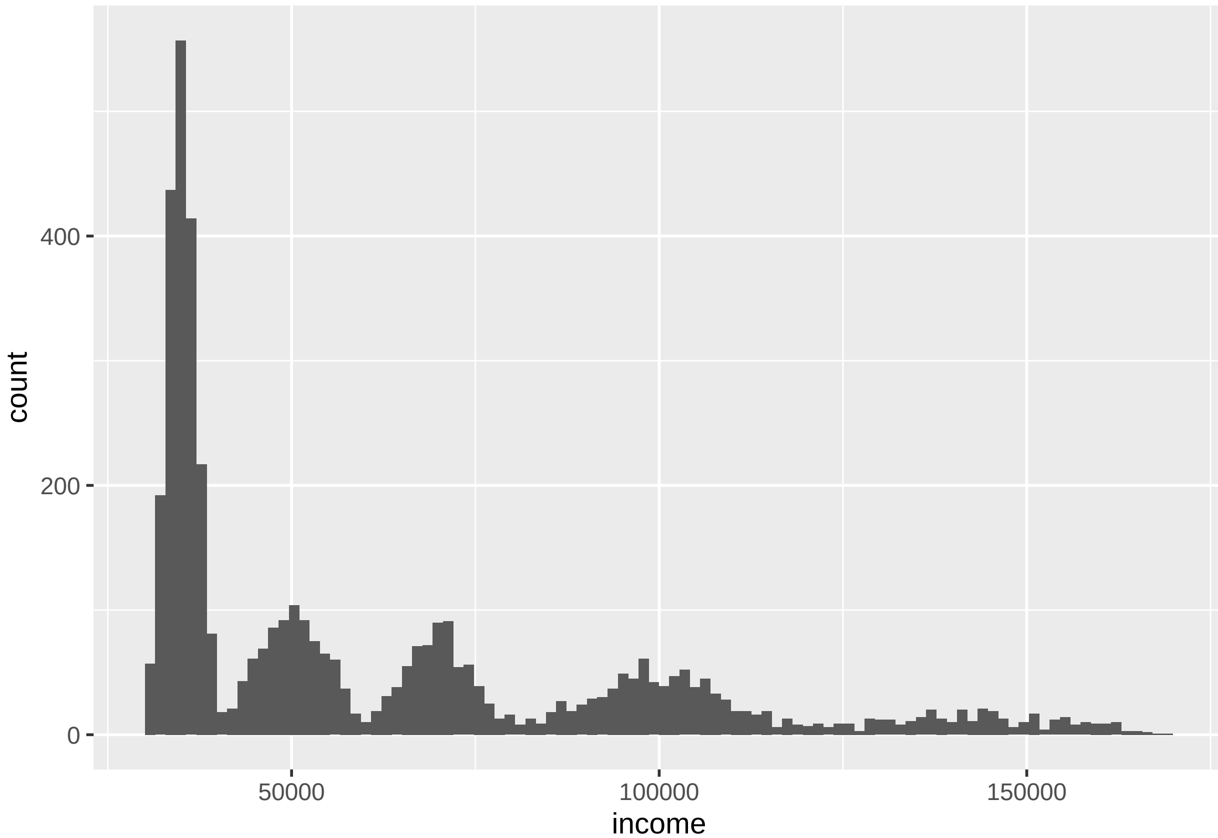
What kind of distribution does age and income belong to? Are they in normal distribution or not?

**Answer**:

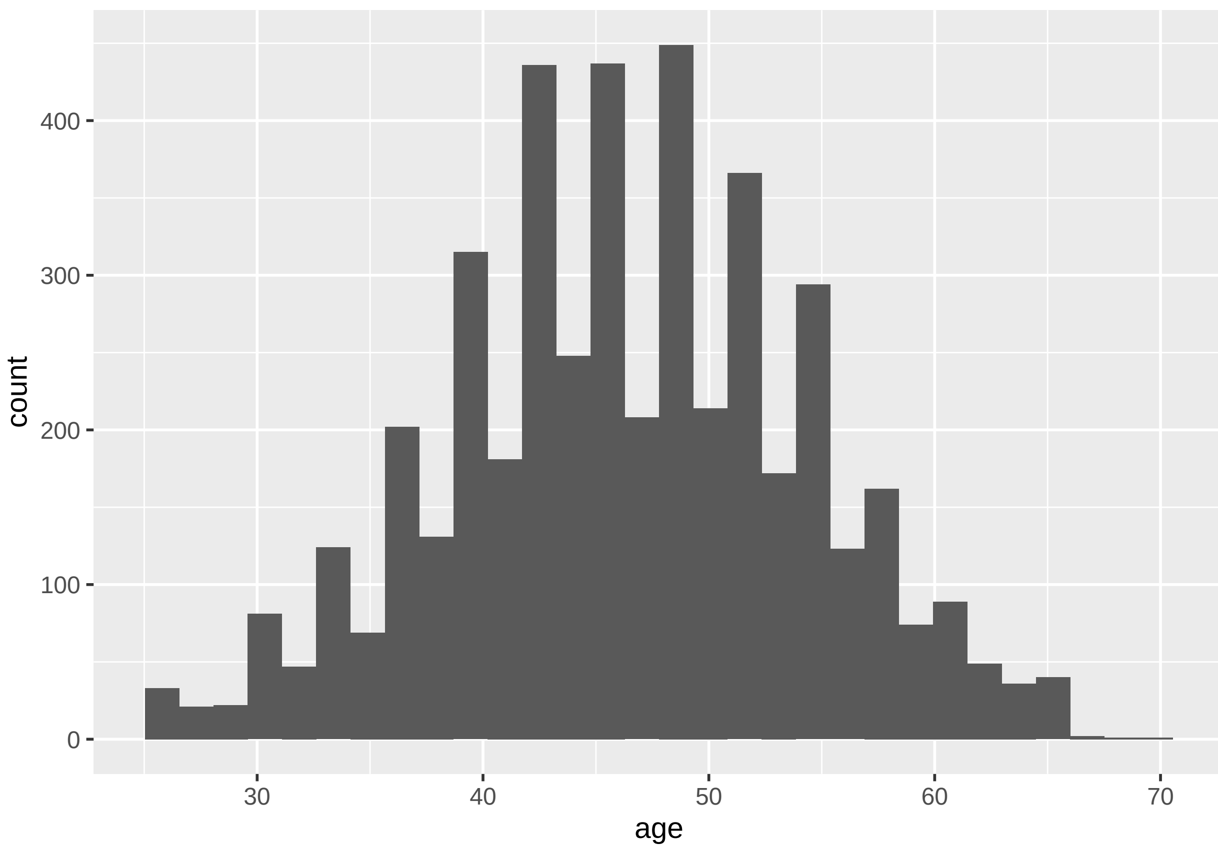
~~~Income: According to skewness and kurtosis which are 1.17 and 3.44, we can say that the income is skewed and not normal distribution.

~~~Age: According to skewness and kurtosis which are -0.057 and 2.744, we can say that the age is not skewed and belongs to normal distribution.

```
hist1 <- ggplot(credit_card_df, aes(x=income)) +
    geom_histogram( bins=100)
hist1
```




```
hist2 <- ggplot(credit_card_df, aes(x=age)) +
    geom_histogram( bins=30)
hist2
```




## Machine Learning

#Pre-process

```
library(tidymodels)
library(vip)
```

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

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

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

```
## Loading required package: rpart

##
## Attaching package: 'rpart'

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

```r
library(rsample)
library(recipes)
library(ranger)


set.seed(300)

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

customer_status_training <- customer_status_split %>% training()

customer_status_test <- customer_status_split %>% testing()

# Create folds for cross validation on the training data set
## These will be used to tune model hyperparameters
customer_status_split
```

```
## <Analysis/Assess/Total>
## <3470/1157/4627>
```

```r
customer_status_folds <- vfold_cv(customer_status_training, v = 5)

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

customer_status_recipe %>%
    prep(training = customer_status_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 -0.423      -0.291 -0.993             -0.152         -0.390             -1.59
##  2 -0.0388      1.23   0.718              0.629          0.241             -0.415
##  3 -0.295       0.473 -0.818             -0.788          1.45               0.609
##  4 -0.935       0.473  1.25              -0.409          1.45               3.18
##  5  2.01       -0.291 -0.913              1.97           1.45              -0.415
##  6  0.0891      0.473  0.657             -0.0235        -1.05               0.609
##  7  0.729       1.23  -0.903              1.56           0.854              2.39
##  8 -0.423       1.23  -0.807             -0.0235         1.45               2.39
##  9 -0.679       0.473 -1.01              -0.662         -1.05              -0.415
## 10 -1.06       -1.07   1.28              -1.41           0.241             -0.415
## # ... 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>, ...
```

## Model 1

```
# logistic regression

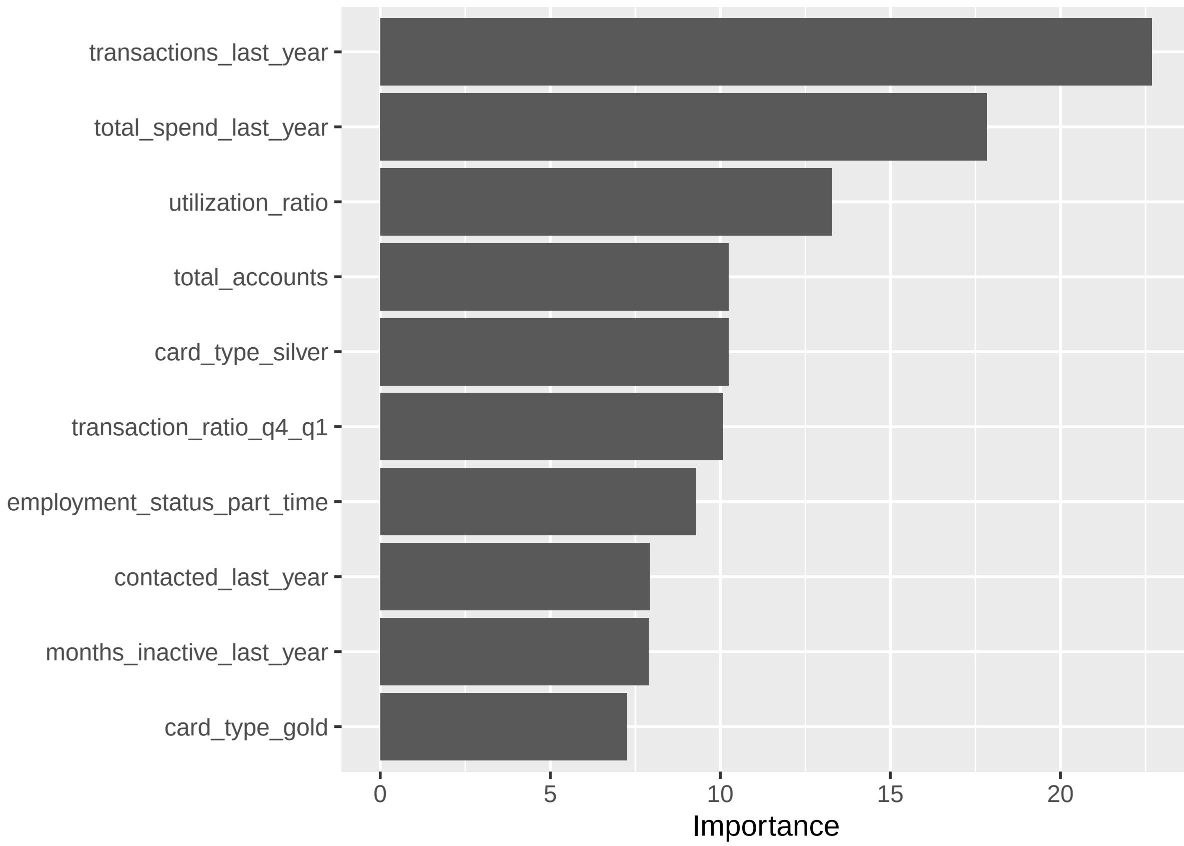
logistic_model <- logistic_reg() %>%
    set_engine('glm') %>%
    set_mode('classification')

lr_wf <- workflow() %>%
    add_model(logistic_model) %>%
    add_recipe(customer_status_recipe)

customer_status_lr_fit <- lr_wf %>%
    fit(data = customer_status_training)

lr_trained_model <- customer_status_lr_fit %>%
    extract_fit_parsnip()

vip(lr_trained_model)
```




```
predictions_categories <- predict(customer_status_lr_fit,
                                  new_data = customer_status_test)

predictions_categories
```

```
## # A tibble: 1,157 x 1
##    .pred_class
##    <fct>
##  1 active
##  2 active
##  3 closed_account
##  4 active
##  5 closed_account
##  6 closed_account
##  7 active
##  8 closed_account
##  9 active
## 10 active
## # ... with 1,147 more rows
```

```r
predictions_probabilities <- predict(customer_status_lr_fit,
                                     new_data = customer_status_test,
                                     type = 'prob')

predictions_probabilities
```

```
## # A tibble: 1,157 x 2
##    .pred_closed_account .pred_active
##                   <dbl>        <dbl>
##  1               0.0198       0.980
##  2               0.362        0.638
##  3               0.561        0.439
##  4               0.256        0.744
##  5               0.840        0.160
##  6               0.977        0.0229
##  7               0.372        0.628
##  8               0.955        0.0453
##  9               0.183        0.817
## 10               0.253        0.747
## # ... with 1,147 more rows
```

```r
test_results <- customer_status_test %>% dplyr::select(customer_status) %>%
    bind_cols(predictions_categories) %>%
    bind_cols(predictions_probabilities)

test_results
```

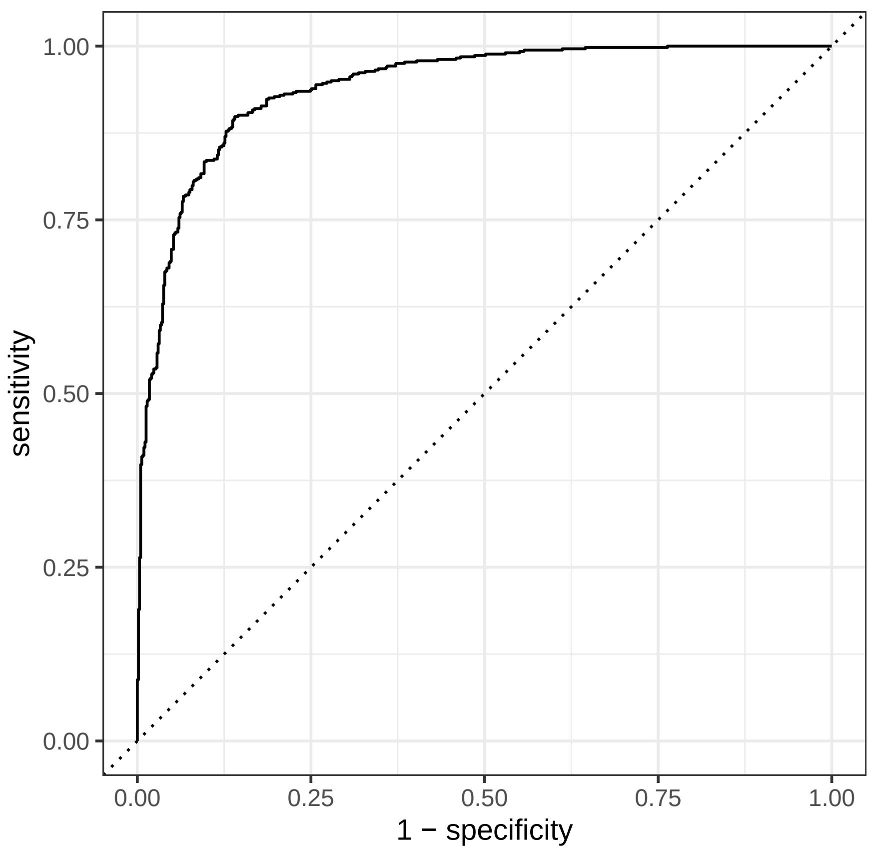
```
## # A tibble: 1,157 x 4
##    customer_status .pred_class    .pred_closed_account .pred_active
##    <fct>           <fct>                         <dbl>        <dbl>
##  1 active          active                       0.0198       0.980
##  2 closed_account  active                       0.362        0.638
##  3 closed_account  closed_account               0.561        0.439
##  4 active          active                       0.256        0.744
##  5 closed_account  closed_account               0.840        0.160
##  6 closed_account  closed_account               0.977        0.0229
##  7 active          active                       0.372        0.628
##  8 closed_account  closed_account               0.955        0.0453
```

```
##  9 active          active                          0.183         0.817
## 10 active          active                          0.253         0.747
## # ... with 1,147 more rows
```

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

```
##                 Truth
## Prediction       closed_account active
##   closed_account            444     74
##   active                     79    560
```

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




```
roc_auc(test_results,
        truth = customer_status,
        estimate = .pred_closed_account)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
```

```
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.942
```

```
my_metrics <- metric_set(accuracy, sens, spec, f_meas, roc_auc)

my_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         0.868
## 2 sens     binary         0.849
## 3 spec     binary         0.883
## 4 f_meas   binary         0.853
## 5 roc_auc  binary         0.942
```

```
last_fit_model <- lr_wf %>%
    last_fit(split = customer_status_split,
             metrics = my_metrics)

last_fit_model %>%
    collect_metrics()
```

```
## # A tibble: 5 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.868 Preprocessor1_Model1
## 2 sens     binary         0.849 Preprocessor1_Model1
## 3 spec     binary         0.883 Preprocessor1_Model1
## 4 f_meas   binary         0.853 Preprocessor1_Model1
## 5 roc_auc  binary         0.942 Preprocessor1_Model1
```

# Model 2

```
#decision tree model

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


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

```
tree_grid <- grid_regular(cost_complexity(),
                          tree_depth(),
                          min_n(),
                          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
```

```
set.seed(300)

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

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     0.948     5 0.00288
## 2    0.0000000001         15     2 roc_auc binary     0.914     5 0.00370
## 3    0.0000000001          1     2 roc_auc binary     0.765     5 0.00666
## 4    0.1                   1     2 roc_auc binary     0.765     5 0.00666
## 5    0.1                  15     2 roc_auc binary     0.765     5 0.00666
## # ... with 1 more variable: .config <chr>
```

```
best_tree <- tree_tuning %>%
    select_best(metric = 'roc_auc')

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
```

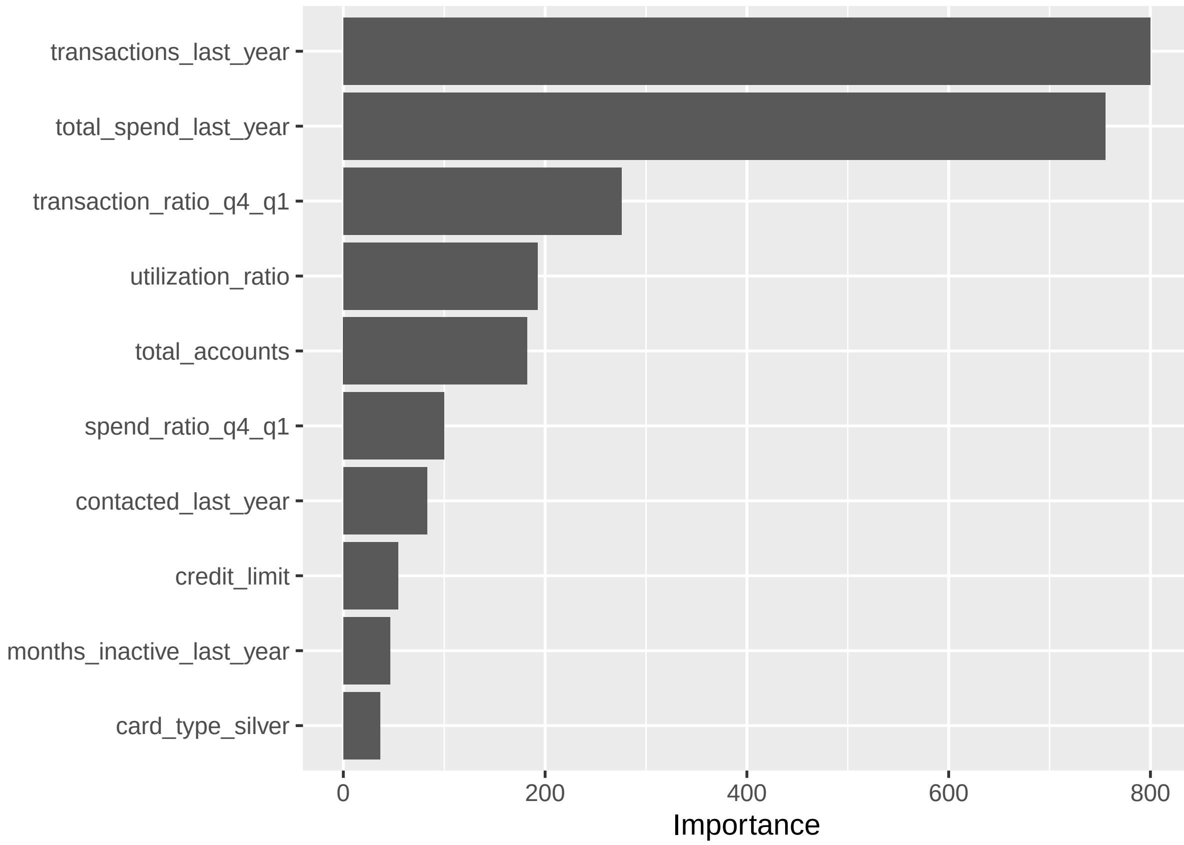
```
final_tree_workflow <- tree_workflow %>%
    finalize_workflow(best_tree)

tree_wf_fit <- final_tree_workflow %>%
```

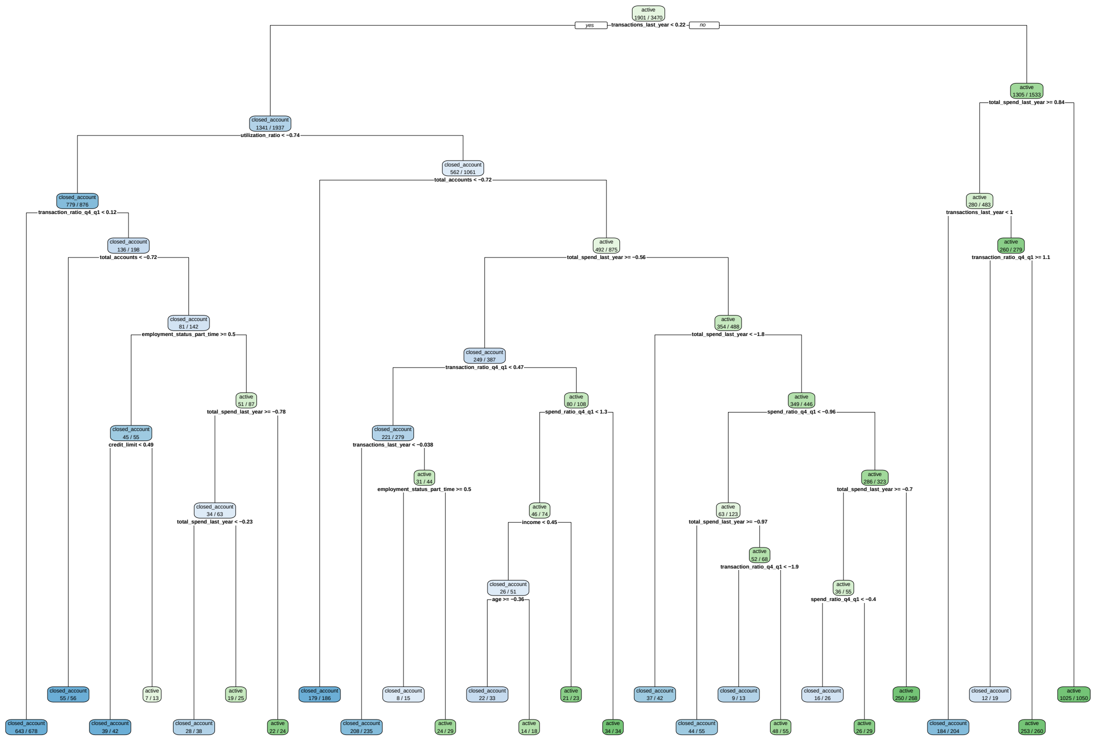
```
    fit(data = customer_status_training)

tree_fit <- tree_wf_fit %>%
    extract_fit_parsnip()

vip(tree_fit)
```




```
rpart.plot(tree_fit$fit, roundint = FALSE, extra = 2)
```

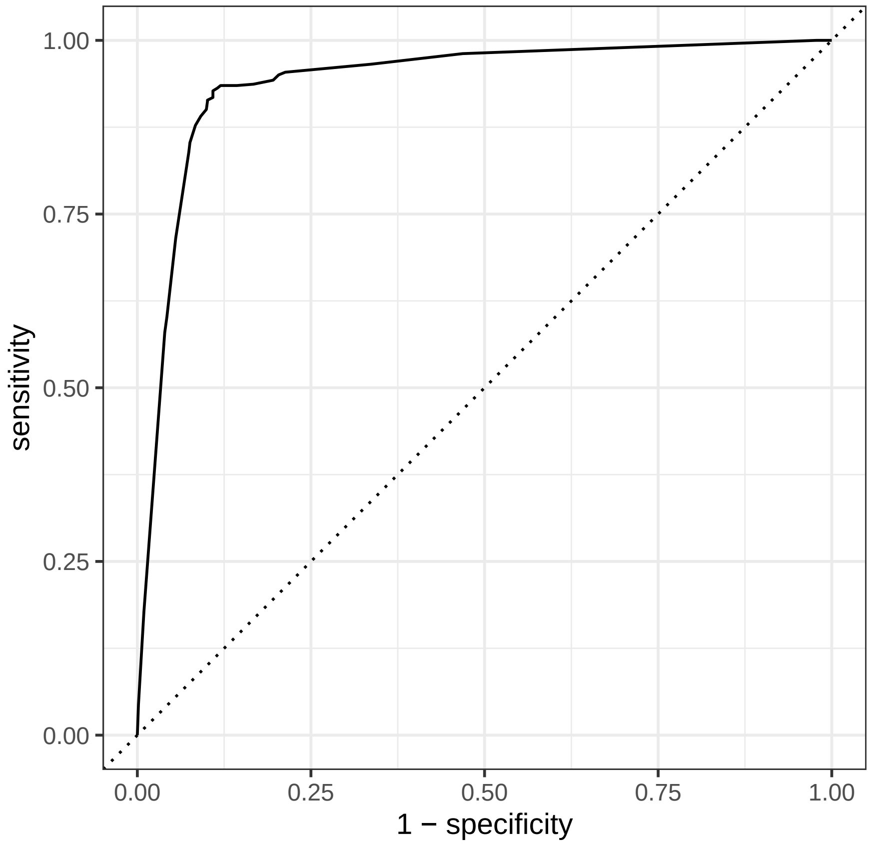



```
tree_last_fit <- final_tree_workflow %>%
    last_fit(customer_status_split)


tree_last_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.903 Preprocessor1_Model1
## 2 roc_auc  binary         0.939 Preprocessor1_Model1
```

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




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

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

```
##                    Truth
## Prediction         closed_account active
##    closed_account             480     69
##    active                      43    565
```

## Model 3

```
# Random forest model

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


rf_workflow <- workflow() %>%
    add_model(rf_model) %>%
    add_recipe(customer_status_recipe)
```

```
set.seed(300)

rf_grid <- grid_random(mtry() %>% range_set(c(2, 4)),
                       trees(),
                       min_n(),
                       size = 10)

rf_grid
```

```
## # A tibble: 10 x 3
##     mtry trees min_n
##    <int> <int> <int>
##  1     3  1868    18
##  2     3  1107    33
##  3     3   812    12
##  4     2   460    29
##  5     2   901     8
##  6     2   129    29
##  7     4  1174    12
##  8     2  1639    29
##  9     2  1733    28
## 10     4  1958    18
```

```
set.seed(300)

rf_tuning <- rf_workflow %>%
    tune_grid(resamples = customer_status_folds,
              grid = rf_grid)

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  1174    12 roc_auc binary     0.988     5 0.00114 Preprocessor1_Model07
## 2     4  1958    18 roc_auc binary     0.988     5 0.00126 Preprocessor1_Model10
## 3     3   812    12 roc_auc binary     0.987     5 0.00132 Preprocessor1_Model03
## 4     3  1868    18 roc_auc binary     0.985     5 0.00151 Preprocessor1_Model01
## 5     3  1107    33 roc_auc binary     0.983     5 0.00186 Preprocessor1_Model02
```

```
best_rf <- rf_tuning %>%
    select_best(metric = 'roc_auc')

best_rf
```

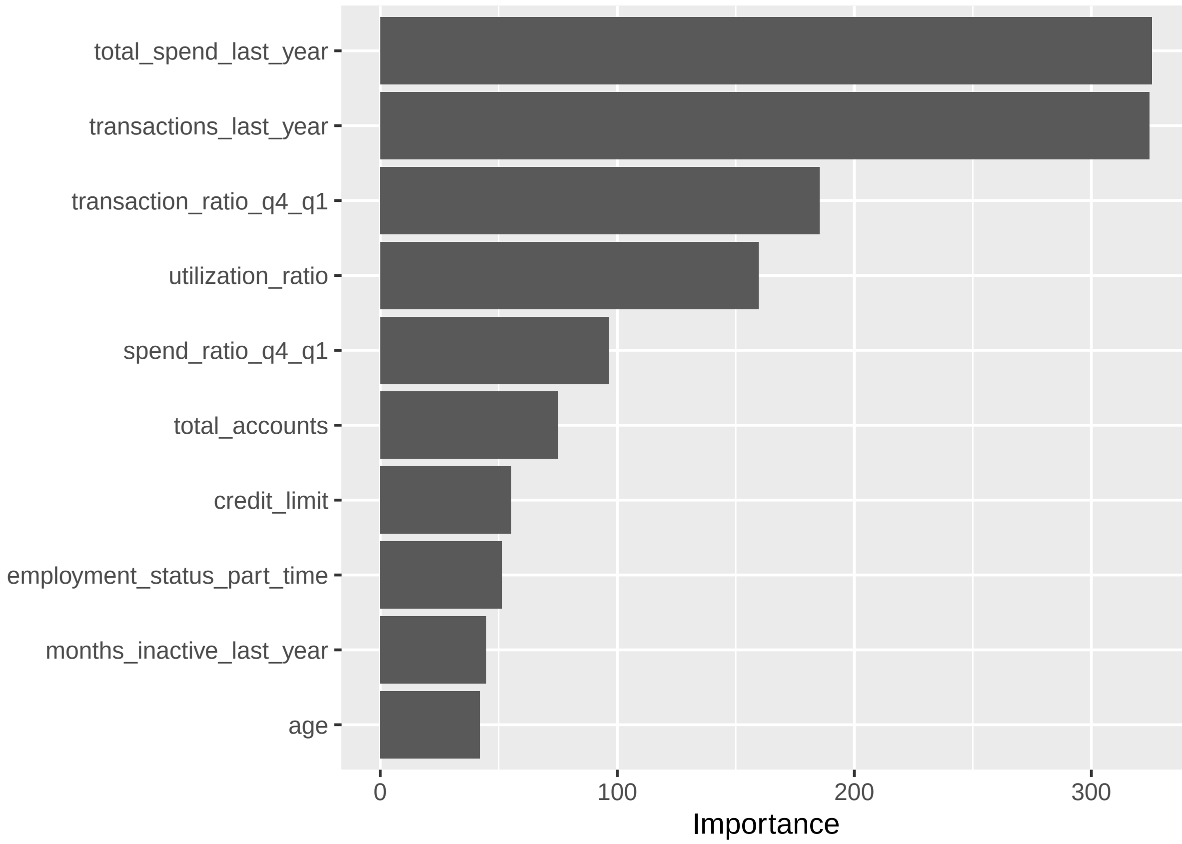
```
## # A tibble: 1 x 4
##    mtry trees min_n .config
##   <int> <int> <int> <chr>
## 1     4  1174    12 Preprocessor1_Model07
```

```
final_rf_workflow <- rf_workflow %>%
    finalize_workflow(best_rf)

rf_wf_fit <- final_rf_workflow %>%
    fit(data = customer_status_training)

rf_fit <- rf_wf_fit %>%
    extract_fit_parsnip()

vip(rf_fit)
```

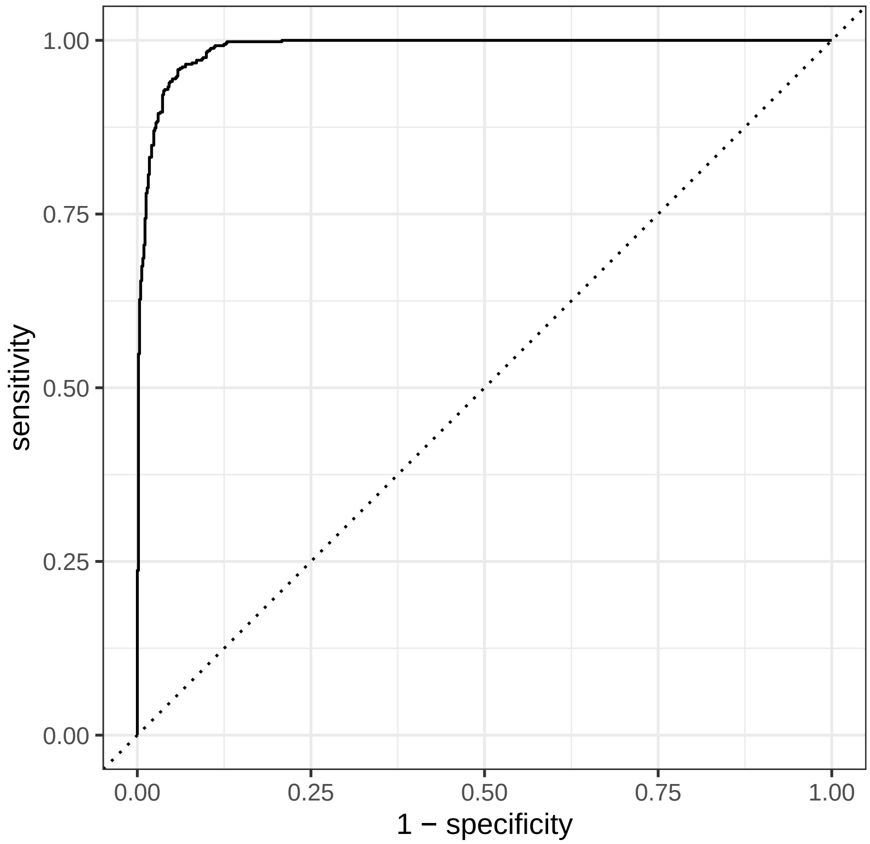



```
rf_last_fit <- final_rf_workflow %>%
    last_fit(customer_status_split)

rf_last_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.945 Preprocessor1_Model1
## 2 roc_auc  binary         0.988 Preprocessor1_Model1
```

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




```
rf_predictions <- rf_last_fit %>% collect_predictions()

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

```
##                  Truth
## Prediction       closed_account active
##   closed_account            496     37
##   active                     27    597
```

## Summary of Results

## Introduction:

The objective of this project is to explore the factors that lead to customers canceling their credit card accounts and develop machine learning algorithms that will predict the likelihood of a customer canceling their account in the future.

## Highlights and key findings from your Exploratory Data Analysis:

The business problem, we are trying to solve is to predict if a customer will close the account or not. Also, what factors are contributing to a customer who has closed the account. Solving this problem will help the company not lose the customers if they know who is going to close their account in the future. If a customer closes the account, the company starts to lose money. Hence predicting the customer who is going to close the account is important.

The goal of our analysis is to predict the customer who is going to close the account and to know which features or factors are contributing more to the decision of a customer closing the account.

We have some interesting results which were found in the Exploratory Data Analysis. We have observed that the total expenditure for the people who closed their account is less as compared to the active account customers. Also, almost as ~80 percent customers who have closed the account have an expenditure less than 3121 when compared to active customers who were only ~34 percent in the base of average expenditure less than 3121. Also, there are many outliers in the total spend last year, which shows that there are several customers who are spending huge amounts. Also, the income distribution is skewed which shows that most of the people are in the range of 0 to 40,000. Also, from the histogram of plot, we can see that most of the customers in the company are in the range 46-47 years old. The utilization ratio and transaction ratio for most of the people who closed their accounts is ~ 0. Hence identifying these customers with 0 transaction and utilization ratio and asking business team of the company to take precautions will help the company not lose the customer.

## Your "best" classification model and an analysis of its performance:

We have trained the data and tested it for performance using decision tree, random forest, and logistic regression models. The best model performance was obtained from the random forest model which had an AUC of 0.988. The accuracy of the model on the test data is 94.6 percent which will give us wrong predictions only about 5.4 percent of the time. The error rate for closed account is 28 out of 523 customers. The AUC for test data is 0.989 demonstrates that the capability of the model to separate the closed account customers and active customers is about 98.9 percent.

## Summary and Recommendations to the Bank:

As we can see, total_spend_last_year, transactions_last_year, transaction_ratio and utilization_ratio have a very strong impact on the model we have built. So, the company can see if someone has less utilization ratio or less total expenditure and call the customers and give them some incentives to spend more. It can also organize some marketing campaigns and use rewards to attract customers to make more transactions.

The total spend last year is suggested as a very important feature by our random forest model, hence we can see if the total spend is becoming less than the average spend for an active customer and the company can ask its business team to make some strategies like campaigns or incentives for the customers so they will not close their account. By this the company can retain its customers and profit.

The number of transactions also is an important feature as suggested by our random forest model, if the transaction number is decreasing, then the customer can be identified, and necessary precautions can be taken by the company business team to not lose the customer.

When it comes to utilization ratio and transaction ratio, most of the customers who have closed their accounts have utilization ratio and transaction ratio of zero, hence identifying such customers and taking precautions would help the company not lose the customer, which will in turn make the company profit.