ANZ_Module_2_Predictive Analytics

Chanyanart KiattipornOpas

11/21/2021

About the Task

This task focus on **Predictive Analysis** aim to predict **The Annual Salary** of each customers.

Explore correlations between annual salary and various customer attributes (e.g. age). These attributes could be those that are readily available in the data (e.g. age) or those that you construct or derive yourself (e.g. those relating to purchasing behaviour). Visualise any interesting correlations using a scatter plot.

Build a simple regression model to predict the annual salary for each customer using the attributes you identified above

How accurate is your model?

For a challenge: build a decision-tree based model to predict salary. Does it perform better? How would you accurately test the performance of this model?

Preparation

1. Download Library

```
library(tidyverse)
library(readxl)
library(corrplot)
library(rpart)
```

2. Import Data into Rstudio

```
transac <- read_csv("ANZ_clean_data.csv")
```

3. Planning for Analysis

Predict "Annual Salary" (Y) by using these attributes:

```
- Age
- Pay Frequency
- Frequency of Sales Transaction
```

```
- Mean of Sales Transaction across 3 months
- Highest price of Sales Transaction
- Median Balance
```

```r
# Create the New List for each Customers by id

customers <- transac %>%
                select(customer_id, age) %>%
                group_by(customer_id) %>%
                distinct() %>%
                as.data.frame()

head(customers)
```

```
##      customer_id age
## 1 CUS-2487424745  26
## 2 CUS-2142601169  38
## 3 CUS-1614226872  40
## 4 CUS-2688605418  20
## 5 CUS-4123612273  43
## 6 CUS-3026014945  27
```

Derive the attributes from Data

1. Salary

```r
# Extract only PAY/SALARY transaction type.

pay_salary <- transac %>%
    select(customer_id,
           txn_description,
           date,
           amount) %>%
    filter(txn_description == "PAY/SALARY") %>%
    group_by(customer_id) %>%
    as.data.frame()

head(pay_salary)
```

```
##      customer_id txn_description                date  amount
## 1 CUS-1462656821      PAY/SALARY 2018-08-01 07:00:00 3903.95
## 2 CUS-2500783281      PAY/SALARY 2018-08-01 07:00:00 1626.48
## 3  CUS-326006476      PAY/SALARY 2018-08-01 07:00:00  983.36
## 4 CUS-1433879684      PAY/SALARY 2018-08-01 07:00:00 1408.08
## 5 CUS-4123612273      PAY/SALARY 2018-08-01 07:00:00 1068.04
## 6 CUS-2487424745      PAY/SALARY 2018-08-01 07:00:00 1013.67
```

2. Pay Frequency

```
pay_salary$date <- as.Date(pay_salary$date)

# How many time of each customers got paid in this 3 months?

pay_freq <- pay_salary %>%
    group_by(customer_id) %>%
    count(amount) %>%
    as.data.frame()

# rename columns
colnames(pay_freq)[3] <- "pay_frequent"

tail(pay_freq)
```

```
##       customer_id  amount pay_frequent
## 95  CUS-586638664 1952.29            6
## 96   CUS-72755508  725.32           12
## 97  CUS-809013380 1037.07           13
## 98  CUS-860700529 1808.62            6
## 99  CUS-880898248 1433.98            6
## 100 CUS-883482547 3977.46            7
```

3. Annual Salary (Y)

```
# Total Salary across 3 months

total_3m_salary <- pay_freq %>%
    mutate(threeM_salary = amount*pay_frequent)

head(total_3m_salary)
```

```
##      customer_id  amount pay_frequent threeM_salary
## 1 CUS-1005756958  970.47           13      12616.11
## 2 CUS-1117979751 3578.65            7      25050.55
## 3 CUS-1140341822 1916.51            6      11499.06
## 4 CUS-1147642491 1711.39           13      22248.07
## 5 CUS-1196156254 3903.73            7      27326.11
## 6 CUS-1220154422 2282.36            7      15976.52
```

Three months is one Quater, so 1 year = 4 quaters

```
# Calculate Annual Salary

annual_salary <- total_3m_salary %>%
    mutate(annual_salary = threeM_salary*4)

# rename columns
colnames(annual_salary)[2] <- "salary"
```

```r
tail(annual_salary)
```

```
##       customer_id  salary pay_frequent threeM_salary annual_salary
## 95  CUS-586638664 1952.29            6      11713.74      46854.96
## 96   CUS-72755508  725.32           12       8703.84      34815.36
## 97  CUS-809013380 1037.07           13      13481.91      53927.64
## 98  CUS-860700529 1808.62            6      10851.72      43406.88
## 99  CUS-880898248 1433.98            6       8603.88      34415.52
## 100 CUS-883482547 3977.46            7      27842.22     111368.88
```

```r
customers <- customers %>%
    inner_join(annual_salary, by = "customer_id")

tail(customers)
```

```
##        customer_id age  salary pay_frequent threeM_salary annual_salary
## 95   CUS-134833760  52 3785.78            7      26500.46     106001.84
## 96  CUS-2505971401  40 1946.57           13      25305.41     101221.64
## 97  CUS-2819545904  42 3231.26            7      22618.82      90475.28
## 98  CUS-3395687666  42 1757.81            6      10546.86      42187.44
## 99  CUS-1147642491  34 1711.39           13      22248.07      88992.28
## 100  CUS-261674136  29 4405.30            7      30837.10     123348.40
```

4. Frequency / Mean / Highest price of Sales Transaction across 3 months

```r
sale_transac <- transac %>% select(customer_id,
                            txn_description,
                            amount) %>%
            filter(txn_description == "POS" |
                     txn_description == "SALES-POS") %>%
            group_by(customer_id) %>%
            summarise(transac_freq = length(customer_id),
                      sum_amount = sum(amount),
                      mean_amount = mean(amount),
                      highest_amount= max(amount))

head(sale_transac)
```

```
## # A tibble: 6 x 5
##   customer_id    transac_freq sum_amount mean_amount highest_amount
##   <chr>                 <int>      <dbl>       <dbl>          <dbl>
## 1 CUS-1005756958           48      1811.        37.7           227.
## 2 CUS-1117979751           52      3976.        76.5          2886.
## 3 CUS-1140341822           65      4390.        67.5          1271.
## 4 CUS-1147642491           76      3886.        51.1           433.
## 5 CUS-1196156254          163      4941.        30.3           391.
## 6 CUS-1220154422           48      3165.        65.9           169.
```

```r
customers <- customers %>%
    inner_join(sale_transac, by = "customer_id")

head(customers)
```

```
##      customer_id age  salary pay_frequent threeM_salary annual_salary
## 1 CUS-2487424745  26 1013.67           14      14191.38      56765.52
## 2 CUS-2142601169  38 1002.13           13      13027.69      52110.76
## 3 CUS-1614226872  40  892.09           13      11597.17      46388.68
## 4 CUS-2688605418  20 2320.30            6      13921.80      55687.20
## 5 CUS-4123612273  43 1068.04           14      14952.56      59810.24
## 6 CUS-3026014945  27 2840.15            7      19881.05      79524.20
##   transac_freq sum_amount mean_amount highest_amount
## 1          531    9819.21    18.49192        1452.21
## 2          276    9685.76    35.09333        2349.55
## 3          220    6845.27    31.11486         235.36
## 4          101    4109.44    40.68752         444.28
## 5           85    4940.56    58.12424         760.27
## 6          248    6532.29    26.33988         385.87
```

5. Balance

```r
med_balance <- transac %>% select(customer_id,
                            balance) %>%
                  group_by(customer_id) %>%
            summarise(median_balance = median(balance))
```

```r
customers <- customers %>%
    inner_join(med_balance, by = "customer_id") %>%
    as_tibble()

head(customers)
```

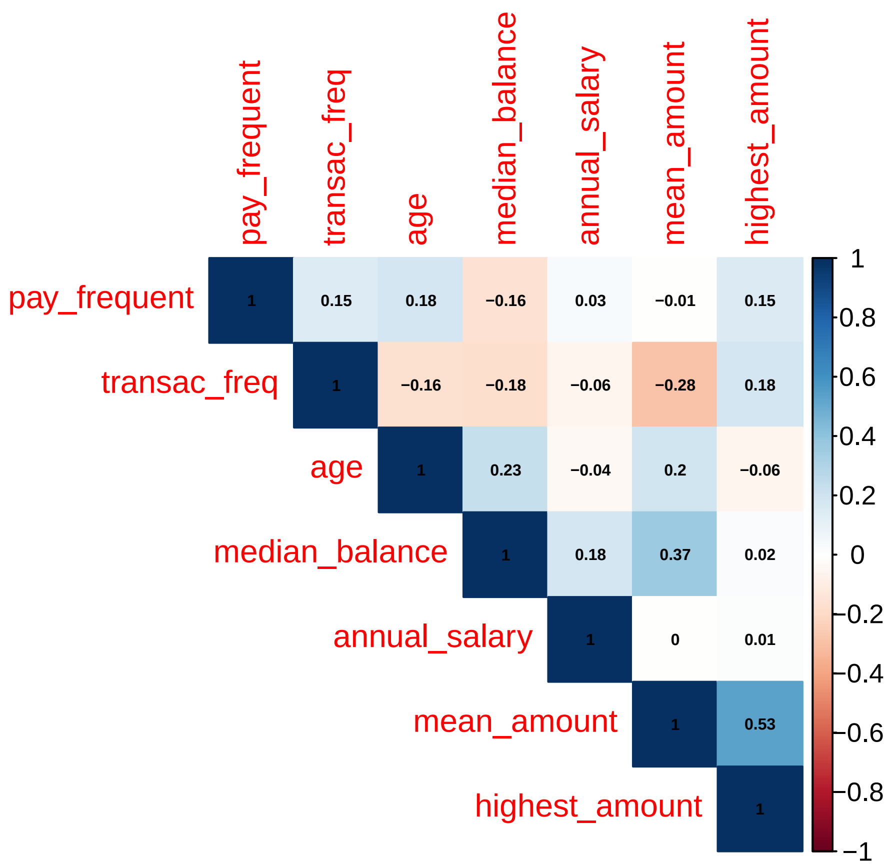
```
## # A tibble: 6 x 11
##   customer_id      age salary pay_frequent threeM_salary annual_salary transac_freq
##   <chr>          <dbl>  <dbl>        <int>         <dbl>         <dbl>        <int>
## 1 CUS-2487424745    26  1014.           14        14191.        56766.          531
## 2 CUS-2142601169    38  1002.           13        13028.        52111.          276
## 3 CUS-1614226872    40   892.           13        11597.        46389.          220
## 4 CUS-2688605418    20  2320.            6        13922.        55687.          101
## 5 CUS-4123612273    43  1068.           14        14953.        59810.           85
## 6 CUS-3026014945    27  2840.            7        19881.        79524.          248
## # ... with 4 more variables: sum_amount <dbl>, mean_amount <dbl>,
## #   highest_amount <dbl>, median_balance <dbl>
```

Correlation

```r
cor_matrix <- customers %>%
                select(annual_salary,
                       age,
                       pay_frequent,
                       transac_freq,
                       mean_amount,
                       highest_amount,
                       median_balance) %>%
                cor()
```

```r
# Visualize

corrplot(cor_matrix,
         method = "color",
         type="upper",
         addCoef.col = "black", number.cex = 0.5,
         order = "hclust")
```



Check Complete Observation

```r
mean(complete.cases(customers))
```

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

1 means all observations has no NULL value. All data are completed.

Prediction Model

```r
# Split data

set.seed(21)
n <- nrow(customers)
id <- sample(1:n, size = n*0.8)
train_data <- customers[id, ]
test_data <- customers[-id, ]
```

1. Linear Regression Model

1.1 Train Model

```r
lmModel <- lm(annual_salary ~ age +
                              pay_frequent +
                              transac_freq +
                              mean_amount +
                              highest_amount +
                              median_balance,
              data = train_data)

lmModel
```

```
##
## Call:
## lm(formula = annual_salary ~ age + pay_frequent + transac_freq +
##     mean_amount + highest_amount + median_balance, data = train_data)
##
## Coefficients:
##    (Intercept)             age    pay_frequent    transac_freq     mean_amount
##     69530.3716       -252.3831        807.8482        -38.3438        -72.2697
## highest_amount  median_balance
##         1.7474          0.3621
```

1.2 Score Model (Prediction)

```r
p1 <- predict(lmModel, newdata = test_data)
p1
```

```
##         1         2         3         4         5         6         7         8
##  65326.72 148352.24  70357.52  73520.43  71207.09  64422.20  61986.95  81697.52
```

```
##         9        10        11        12        13        14        15        16
## 61310.67  66636.36  69638.32  62921.58  68662.12  60777.08  65135.75  67562.00
##        17        18        19        20
## 69238.98  67918.85  64620.27  61941.98
```

1.3 Evaluate Model

```r
error1 <- p1 - test_data$annual_salary
error1
```

```
##          1          2          3          4          5          6          7
##   9639.520  96033.199  11140.441  -4627.006  -1396.874  24995.962  -7309.208
##          8          9         10         11         12         13         14
##   7570.884   2895.952  38012.524  21207.117  12457.143  11999.080  30825.077
##         15         16         17         18         19         20
##  -4140.730  27326.558   4730.576  21623.572  20384.915 -44059.860
```

```r
# RMSE

rmse_test1 <- sqrt(mean(error1**2))
rmse_test1
```

```
## [1] 29121.63
```

2. Decision Tree Model

2.1 Train Model

```r
rpartModel <- rpart(annual_salary ~ age +
                                pay_frequent +
                                transac_freq +
                                mean_amount +
                                highest_amount +
                                median_balance,
              data = train_data)

rpartModel
```

```
## n= 80
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 80 63262720000  69529.77
##    2) median_balance< 7265.298 37  5645786000  52092.25
##      4) pay_frequent< 6.5 13   318500900  43639.95 *
##      5) pay_frequent>=6.5 24  3895482000  56670.57
##       10) transac_freq< 74.5 9   940774300  49407.77 *
##       11) transac_freq>=74.5 15  2195132000  61028.25 *
```

```
##    3) median_balance>=7265.298 43 36685790000  84534.15  
##      6) pay_frequent< 6.5 15 11601310000  70901.20 *
##      7) pay_frequent>=6.5 28 20803130000  91837.52  
##       14) transac_freq< 29.5 7  6847314000  71977.91 *
##       15) transac_freq>=29.5 21 10274710000  98457.38  
##         30) median_balance< 8771.515 7  2600612000  81281.25 *
##         31) median_balance>=8771.515 14  4576391000 107045.50 *
```

2.2 Score Model (Prediction)

```r
p2 <- predict(rpartModel, newdata = test_data)
p2
```

```
##         1         2         3         4         5         6         7         8 
##  43639.95  70901.20  61028.25  81281.25  49407.77  61028.25  70901.20  71977.91 
##         9        10        11        12        13        14        15        16 
##  49407.77  49407.77  70901.20  49407.77  49407.77  49407.77  43639.95 107045.45 
##        17        18        19        20 
##  70901.20  43639.95  43639.95 107045.45
```

2.3 Evaluate Model

```r
error2 <- p2 - test_data$annual_salary
error2
```

```
##           1           2           3           4           5           6 
## -12047.2492  18582.1627   1811.1733   3133.8057 -23196.1911  21602.0133 
##           7           8           9          10          11          12 
##   1605.0427  -2148.7257  -9006.9511  20783.9289  22470.0027  -1056.6711 
##          13          14          15          16          17          18 
##  -7255.2711  19455.7689 -25636.5292  66810.0143   6392.8027  -2655.3292 
##          19          20 
##   -595.4092   1043.6143
```

```r
# RMSE

rmse_test2 <- sqrt(mean(error2**2))
rmse_test2
```

```
## [1] 20169.15
```

```r
## Compare Prediction Model with RMSE
cat("RMSE of Linear Regression Model: ", rmse_test1,
    "\nRMSE of Decision Tree Model: ", rmse_test2)
```

```
## RMSE of Linear Regression Model:  29121.63
## RMSE of Decision Tree Model:  20169.15
```

Conclusion

To predict **The Annual Salary** of each customers. We use 6 attributions to predict the annual salary, which are customer's age, the frequent of salary pay, the median balance of customer's bank account, the purchasing behaviors(the frequent of buying, the highest and the average amounts of transactions) in this three months.

With 2 Prediction Models that are "Linear Regression Model" and "Decision Tree Model". As the result, we found that Decision Tree Model is better than Linear Regression Model for this data. We clearly see that RMSE of Decision Tree Model (RMSE: 20169.15) less than Linear Regression Model (RMSE: 29121.63).

However RMSE is more than 20000, that indicates an inaccuracy of the model. The variable may not suit to predict the the Annual Salary, More data and More variable are required to develop the reliable model.