ANZ Task 2

Hayden Savage

05/08/2021

```r
library(tidyverse)
library(janitor)
library(ggplot2)
library(reshape2)
library(readxl)
library(rpart)
library(gridExtra)
library(rattle)
library(rpart.plot)
library(RColorBrewer)


# read in data
data = read_excel("ANZ synthesised transaction dataset.xlsx", sheet = "DSynth_Output_100c_3m_v3")
```

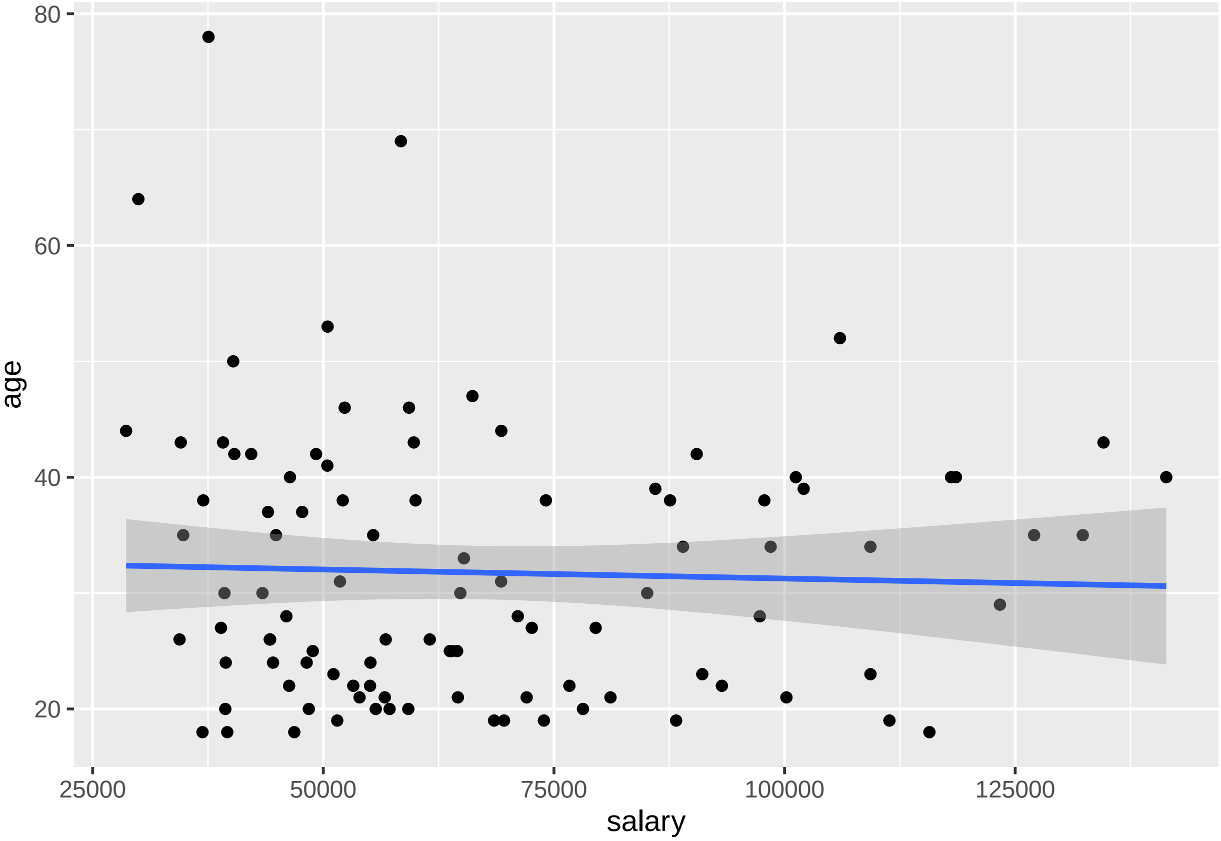
```r
qwe = data %>%
  filter(movement=='credit')

unique(qwe$txn_description) # all credits are salary related
```

```
## [1] "PAY/SALARY"
```

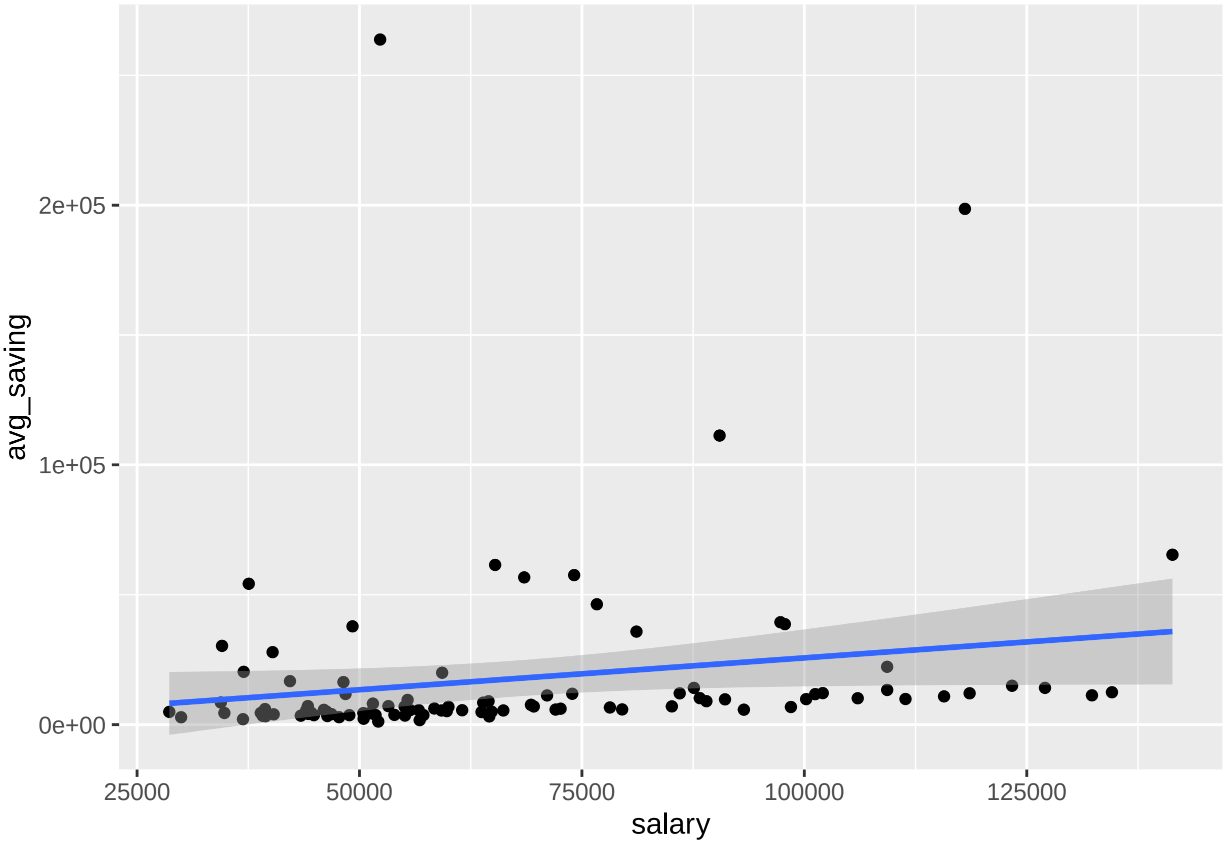
```r
# Salary, age and gender for each customer
salary_data = data %>%
  filter(movement=='credit') %>%
  group_by(customer_id, age, gender) %>%
  summarise(salary = sum(amount)*4)

# Average monthly spending for each customer
spending_data = data %>%
  filter(movement=='debit') %>%
  group_by(customer_id) %>%
  summarise(spending = sum(amount)/3)

# Average amount in savings for each customer
savings_data = data %>%
  group_by(customer_id) %>%
  summarise(avg_saving = mean(balance))

# Number of transactions with authorised status
auth_data = data %>%
  filter(status=='authorized') %>%
  group_by(customer_id) %>%
  count(status, name='auth')

# Number of transactions with posted status
post_data = data %>%
  filter(status=='posted') %>%
  group_by(customer_id) %>%
  count(status, name='post')

# Merge data frames
ml_data = merge(merge(merge(merge(salary_data, spending_data, by='customer_id'),
              savings_data, by='customer_id'), auth_data, by='customer_id'),
              post_data, by='customer_id')

print(paste("Mean salary = ", mean(ml_data$salary)))
```

```
## [1] "Mean salary =  67063.074"
```

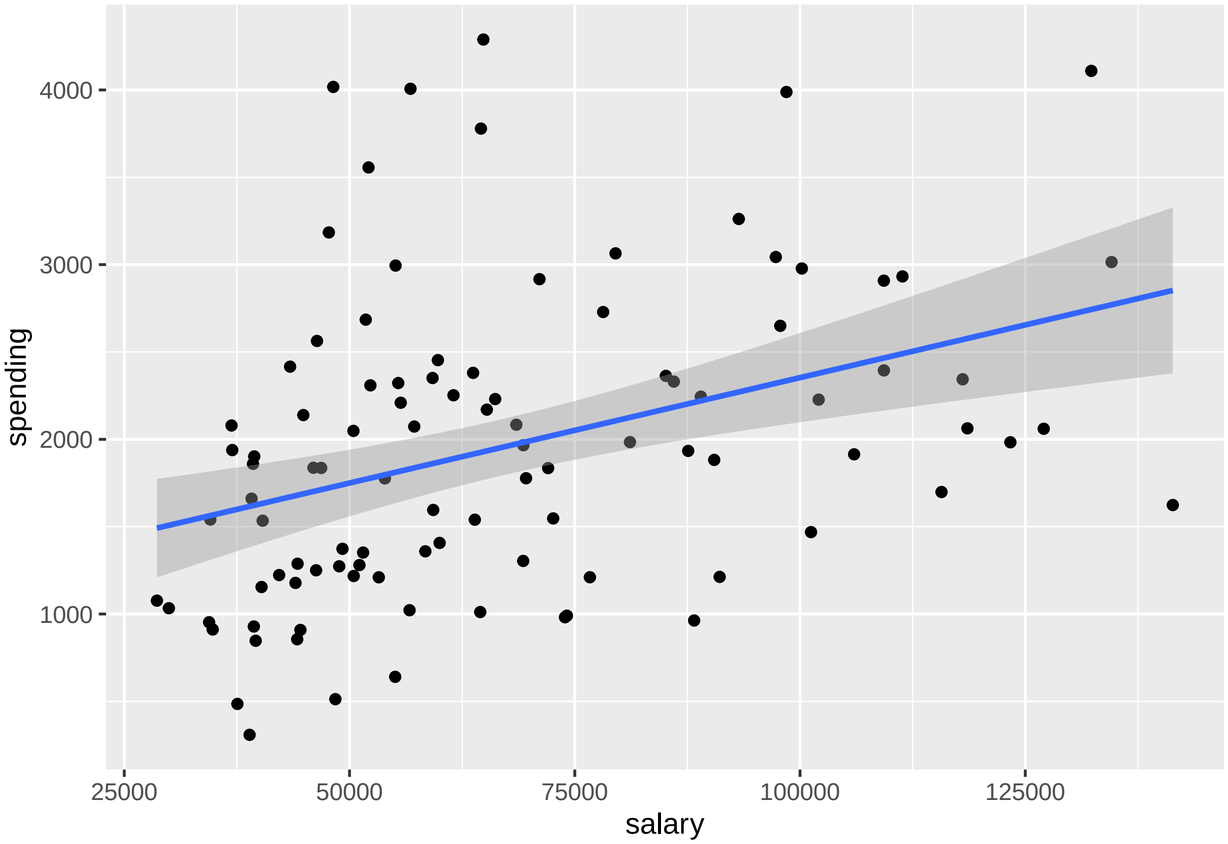
```
# Visualising correlations between salary and other customer attrubutes

ggplot(ml_data, aes(salary, age)) +
  geom_point() +
  geom_smooth(method=lm)
```



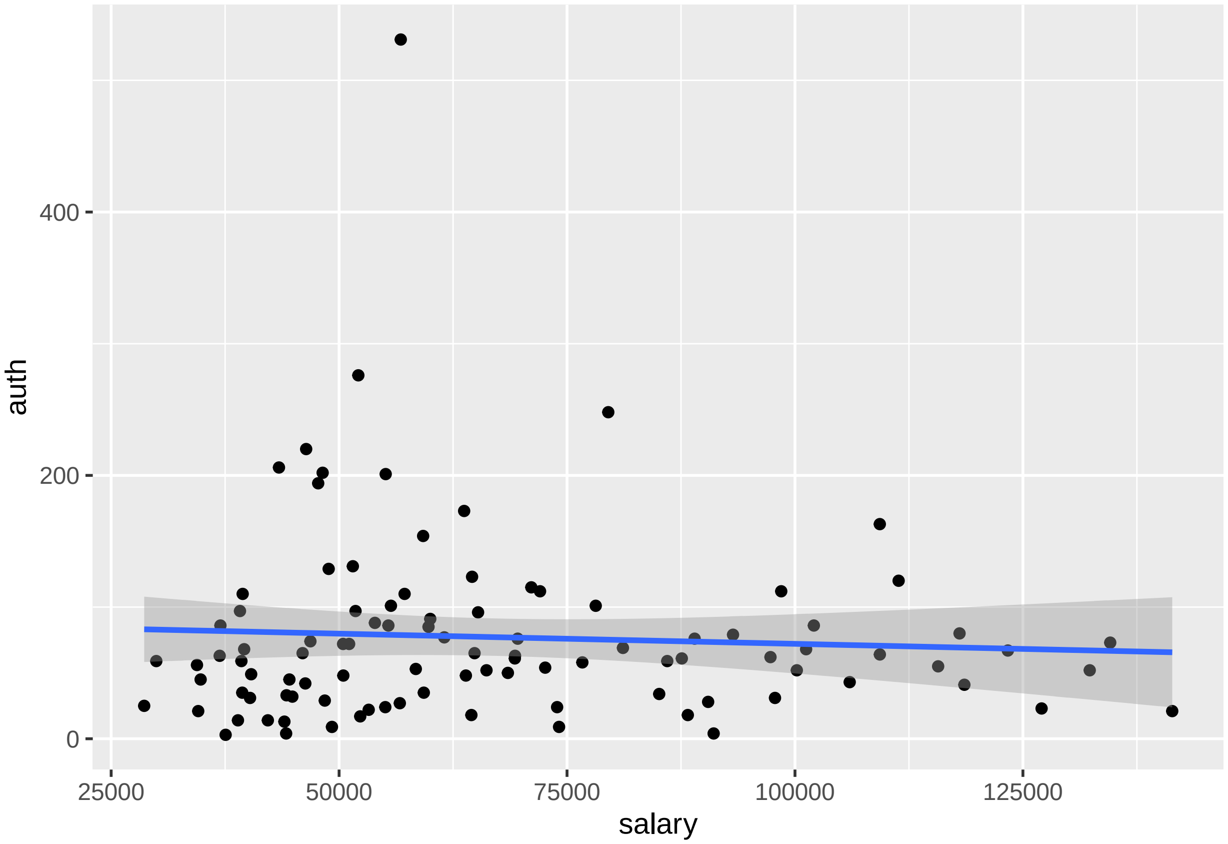
```
ggplot(ml_data, aes(salary, avg_saving)) +
  geom_point() +
  geom_smooth(method=lm)
```



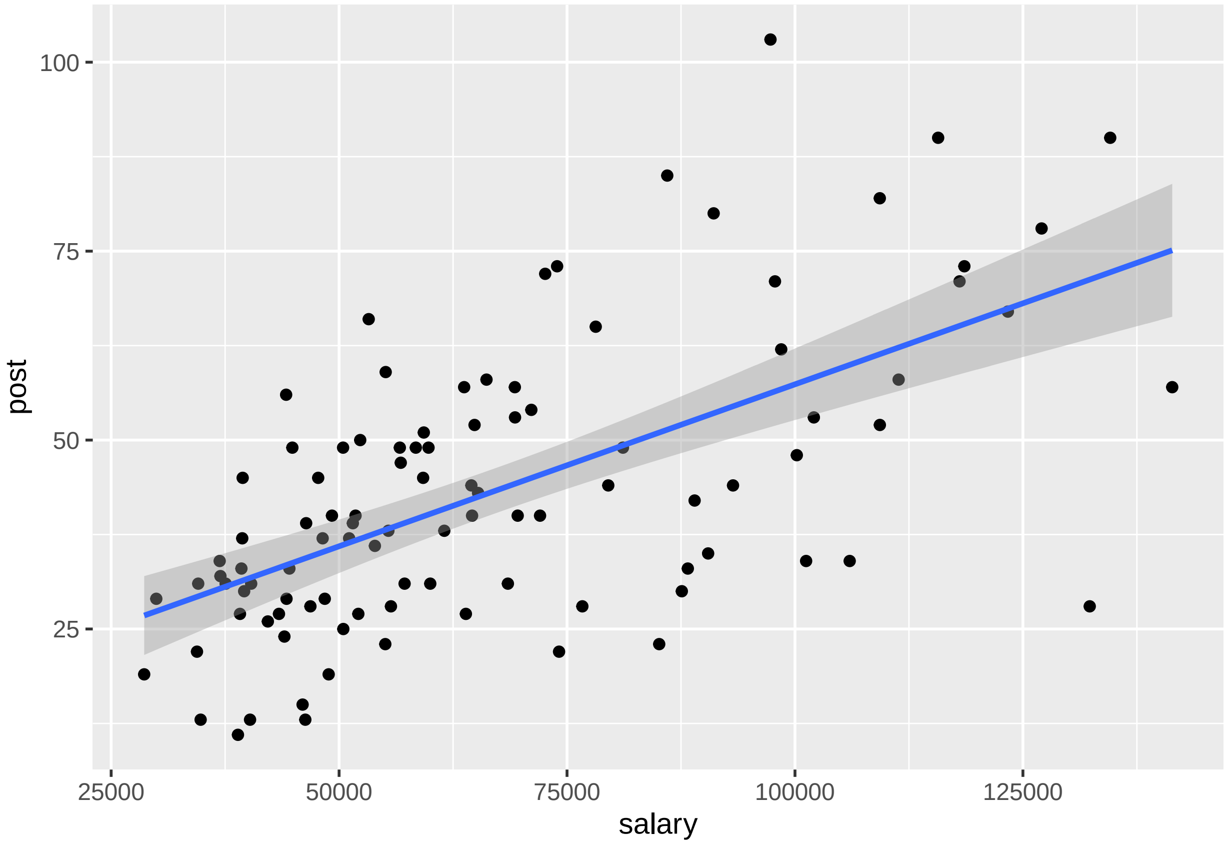
```
ggplot(ml_data, aes(salary, spending)) +
  geom_point() +
  geom_smooth(method=lm)
```



```
ggplot(ml_data, aes(salary, auth)) +
  geom_point() +
  geom_smooth(method=lm)
```



```
ggplot(ml_data, aes(salary, post)) +
  geom_point() +
  geom_smooth(method=lm)
```



```r
# Multiple regression using all variables
model = lm(salary ~ age + gender + spending + avg_saving + auth + post, data = ml_data)
summary(model)
```

```
##
## Call:
## lm(formula = salary ~ age + gender + spending + avg_saving +
##     auth + post, data = ml_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43515 -12381  -3696  10625  59191
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.287e+04  9.001e+03   2.541 0.012697 *
## age         -1.034e+02  1.827e+02  -0.566 0.572592
## genderM      4.144e+03  4.129e+03   1.003 0.318227
## spending     1.171e+01  3.160e+00   3.705 0.000359 ***
## avg_saving   5.352e-02  6.093e-02   0.878 0.381977
## auth        -1.079e+02  3.726e+01  -2.897 0.004702 **
## post         6.852e+02  1.176e+02   5.824 8.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20170 on 93 degrees of freedom
## Multiple R-squared:  0.4743, Adjusted R-squared:  0.4403
## F-statistic: 13.98 on 6 and 93 DF,  p-value: 2.817e-11
```

Variables used: age, gender, average monthly spending, avgerage savings, No. of authorised transactions, No. of posted transactions

Multiple R-squared: 0.4743

Adjusted R-squared: 0.4403

Residual standard error: 20170

As we can see, the p-value (tests the null hypothesis that the coefficient is equal to zero) is relatively high for the variables age, gender, and average savings. This indicates that these variables are insignificant and do not contribute much to the overall performance of the model.

```r
# Variable selection using AIC
new_model = step(model, direction = "backward", trace = FALSE)
summary(new_model)
```

```
##
## Call:
## lm(formula = salary ~ spending + auth + post, data = ml_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -37644 -14290  -2604  12778  62178
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22342.594   5934.856   3.765 0.000287 ***
## spending       11.967      3.131   3.822 0.000235 ***
## auth         -115.711     35.977  -3.216 0.001771 **
## post          699.157    116.349   6.009 3.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20090 on 96 degrees of freedom
## Multiple R-squared:  0.462,  Adjusted R-squared:  0.4452
## F-statistic: 27.48 on 3 and 96 DF,  p-value: 6.481e-13
```

Variables: average monthly spending, No. of authorised transactions, No. of posted transactions

Multiple R-squared: 0.462 Adjusted R-squared: 0.4452 Residual standard error: 20090

Backwards stepwise variable selection was performed using the Akaike information criterion (AIC). The AIC evaluates how well a model fits the data. This model has not used the variables age, gender and average savings as they weren't the best fit for the model. A benefit of this model is that is has less complexity and is easier to interpret. This model also has a lower residual standard error and higher adjusted r-squared.

```r
# Naive prediction (ZeroR): predict salary to be the mean of all salaries
error_ls = c()

for (i in 1:10) {
  d1 = ml_data[(i*10-9):(i*10),]
  d2 = ml_data %>%
    filter(!(customer_id %in% d1$customer_id))
  est = mean(d2$salary)
  test = d1 %>%
    mutate(naive_err = abs(est-salary))
  error_ls = append(error_ls, mean(test$naive_err))
}

mean(error_ls)
```

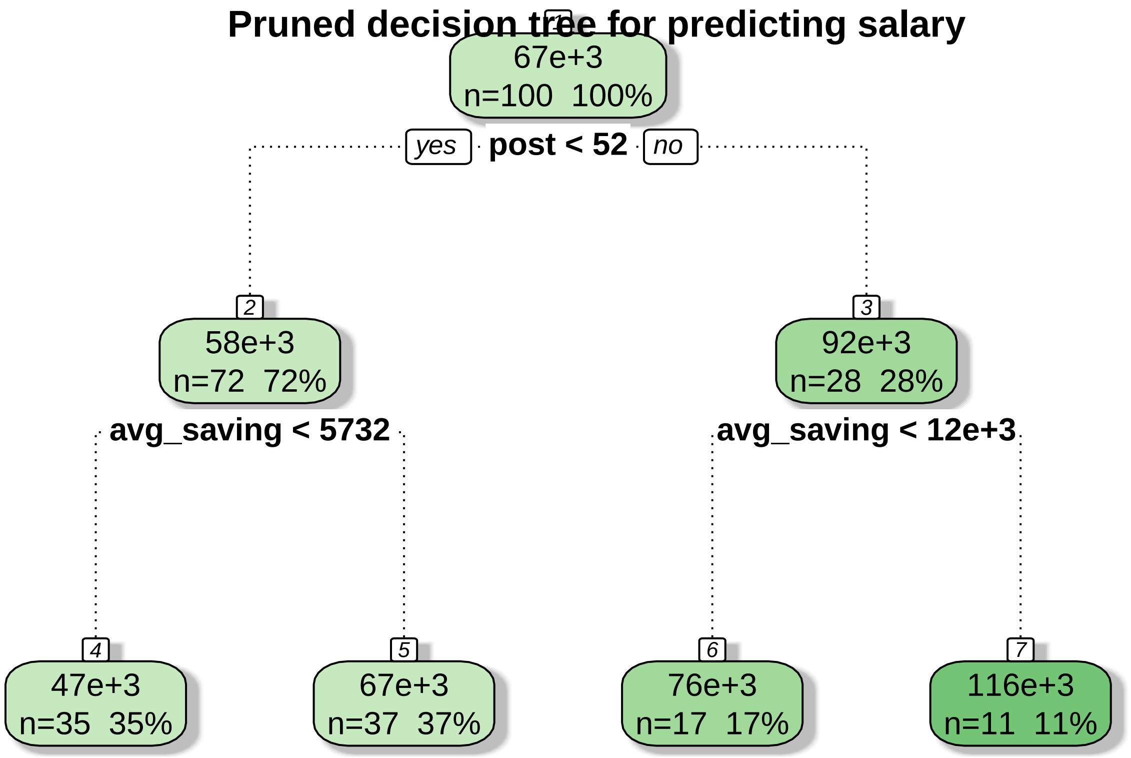
```
## [1] 21995.88
```

Above, a cross validation procedure was performed using a naive prediction. That is, the predicted salary in the test set was the mean salary in the training set. The average residual standard error for the folds was 21995.88. The residual standard error for the multiple regression model was 20090 which is a minimal improvement. Hence, ANZ should not use the regression model to segment customers as it does not accurately predict their salaries. This is because you would get a similar accuracy for just assuming a customers salary to be the mean of all the other known salaries.

```
# Decision-tree based model
set.seed(20)

tree = rpart(salary ~ spending + auth + post + age + gender + avg_saving,
             method = "anova", data = ml_data)

printcp(tree)
```

```
##
## Regression tree:
## rpart(formula = salary ~ spending + auth + post + age + gender +
##     avg_saving, data = ml_data, method = "anova")
##
## Variables actually used in tree construction:
## [1] auth       avg_saving post       spending
##
## Root node error: 7.199e+10/100 = 719902841
##
## n= 100
##
##         CP nsplit rel error  xerror    xstd
## 1 0.323570      0   1.00000 1.00960 0.13976
## 2 0.153739      1   0.67643 1.01035 0.14896
## 3 0.101556      2   0.52269 0.79922 0.13621
## 4 0.060490      3   0.42113 0.63643 0.12055
## 5 0.012791      4   0.36064 0.69762 0.12831
## 6 0.010000      5   0.34785 0.69550 0.12830
```

```
# Pruning the tree
pruned_tree = prune(tree,cp=tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"])
fancyRpartPlot(pruned_tree, caption = NULL)
title("Pruned decision tree for predicting salary", adj = .5)
```



```
residual_std = sd(residuals(pruned_tree))
print(paste("Residual standard error = ", residual_std))
```

```
## [1] "Residual standard error =  17499.670858256"
```

```
r_squared = 1 - tail(pruned_tree$cptable[,"rel error"], n=1)
print(paste("R-squared = ", as.numeric(r_squared)))
```

```
## [1] "R-squared =  0.578865260540084"
```

Cross validation is needed to test the performance of a decision tree, however, rpart already has built-in cross validation. The original tree is also pruned to avoid overfitting. This is done by selecting the tree size that minimises the cross validation error. The decision tree was given all the variables from the first regression model, however, the pruned decision tree only used the variables 'No. of posted transactions', and 'average savings' which was different from the regression model.

Pruned tree:

Residual standard error = 17499.67

R-squared = 0.5789

As we can see, the performance of this decision tree is better than the multiple regression model as it has a lower residual standard error and a higher r-squared.

All my code for the DATA@ANZ program can be found at https://github.com/savage64/ANZ