ANZ Task 2 - Predictive Analytics

Rinaldo Gagiano

20/11/2020

Data Import and Subsetting

```
anz <- read_xlsx("ANZ.xlsx") #Import
anz <- anz[,c(-3:-6,-8,-9,-19:-23)] #Irrelevant column Removal
```

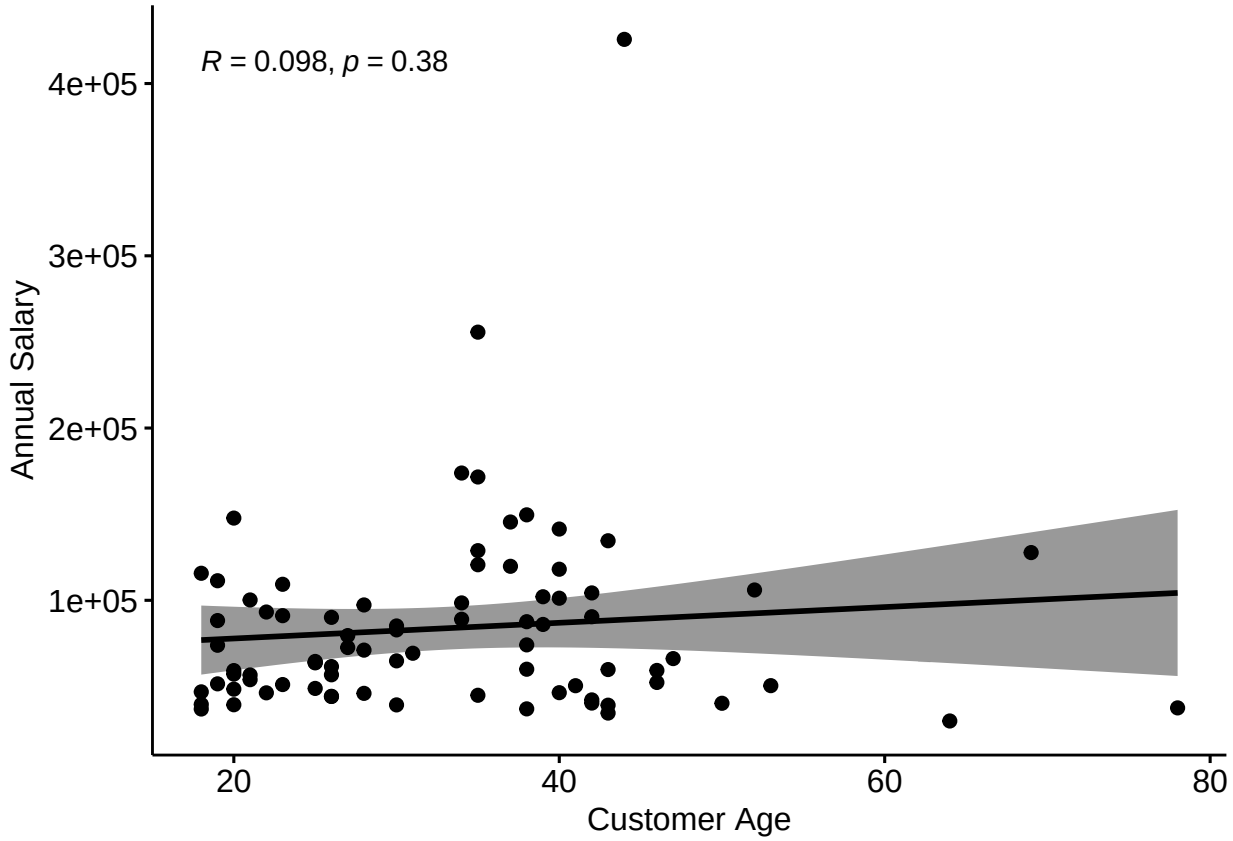
```
#Age, Gender and Annual Salary of each customer
salary <- anz %>%
  filter(txn_description=="PAY/SALARY") %>%
  group_by(first_name) %>%
  summarise(age = max(age),
            gender = max(gender),
            "salary" = (sum(amount)*4))

#Most assigned merchant suberb, merchant state, and txn description per customer
salary2 <- anz %>%
  filter(!txn_description=="PAY/SALARY") %>%
  group_by(first_name) %>%
  summarise(most_suberb = head(names(sort(table(merchant_suburb),decreasing=TRUE)),1),
            most_state = head(names(sort(table(merchant_state),decreasing=TRUE)),1),
            most_txn = head(names(sort(table(txn_description),decreasing=TRUE)),1))

#Balance info for each customer
salary3 <- anz %>%
  group_by(first_name)%>%
  arrange(extraction) %>%
  summarise("initial balance" = head(balance,1),
            "final balance" = tail(balance,1))
salary3 <- salary3 %>% mutate("overall balance change" = `final balance`-`initial balance`)

#Creation of new dataframe
ann_sal <- inner_join(salary3,salary2, by = 'first_name')
ann_sal <- inner_join(ann_sal,salary, by = 'first_name')
ann_sal <- ann_sal[,-1] #Removal of first names
```

Model matrix Creation

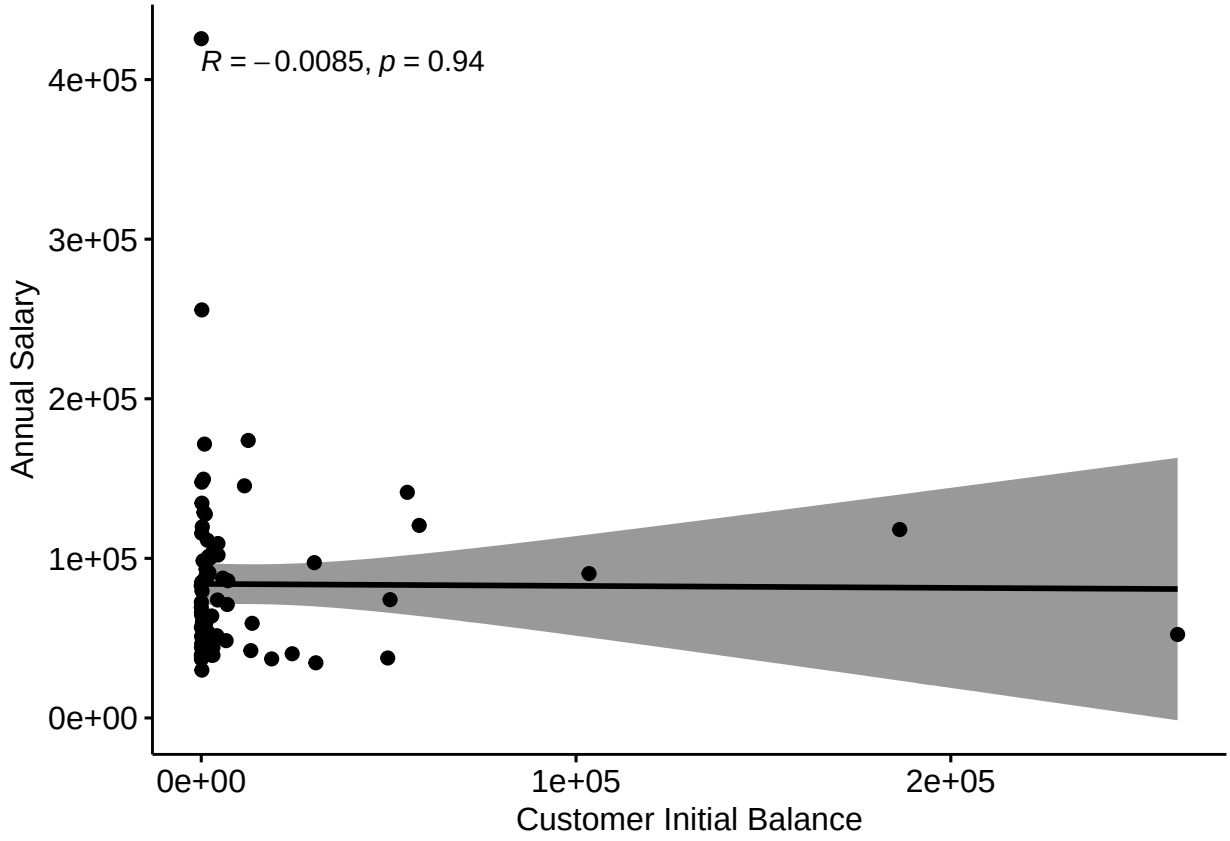
```
suberb <- data.frame(model.matrix(~most_suberb+0, data = ann_sal))
state <- data.frame(model.matrix(~most_state+0, data = ann_sal))
txn <- data.frame(model.matrix(~most_txn+0, data = ann_sal))

total_sal_anz <- ann_sal[,c(-4:-6)]
total_sal_anz$gender <- ifelse(total_sal_anz$gender=="M",1,0)
total_sal_anz <- cbind(total_sal_anz,suberb,state,txn)
```

Scatter Plots

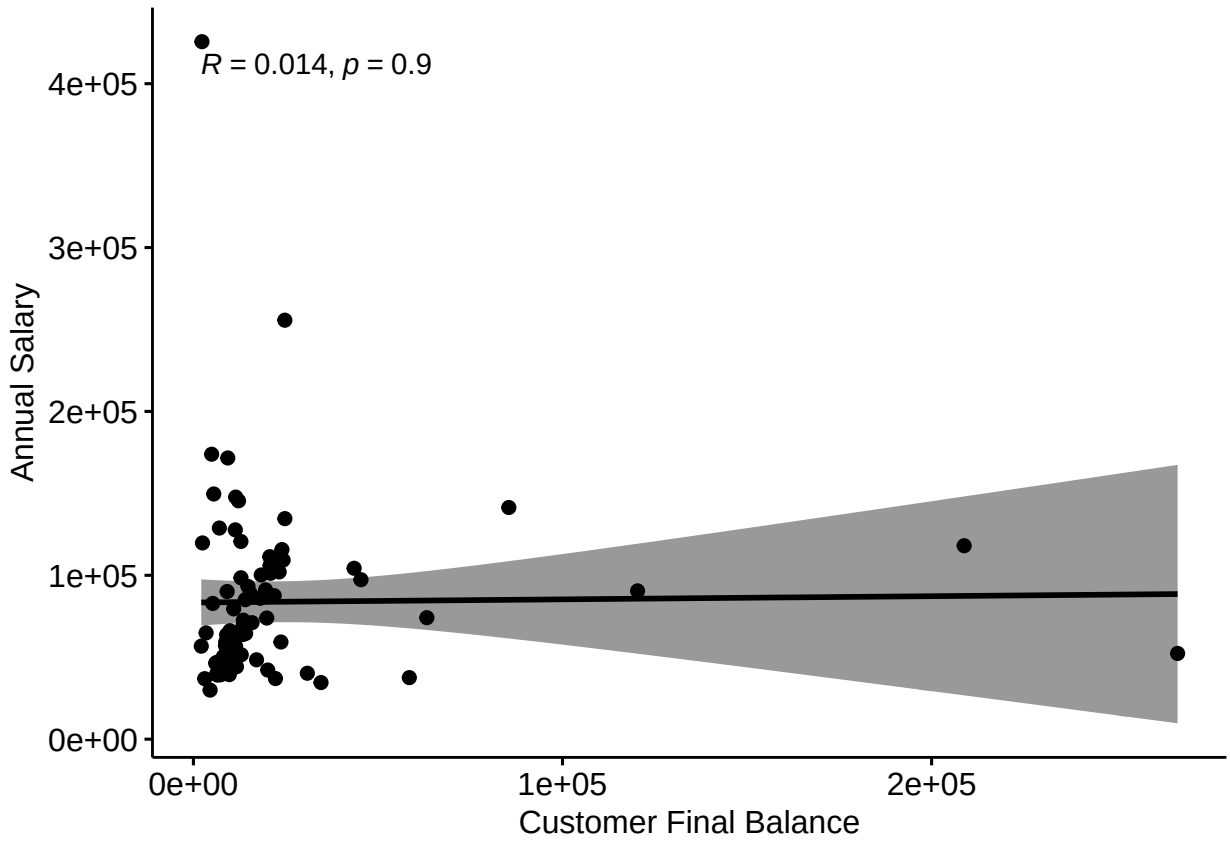
```
## `geom_smooth()` using formula 'y ~ x'
```



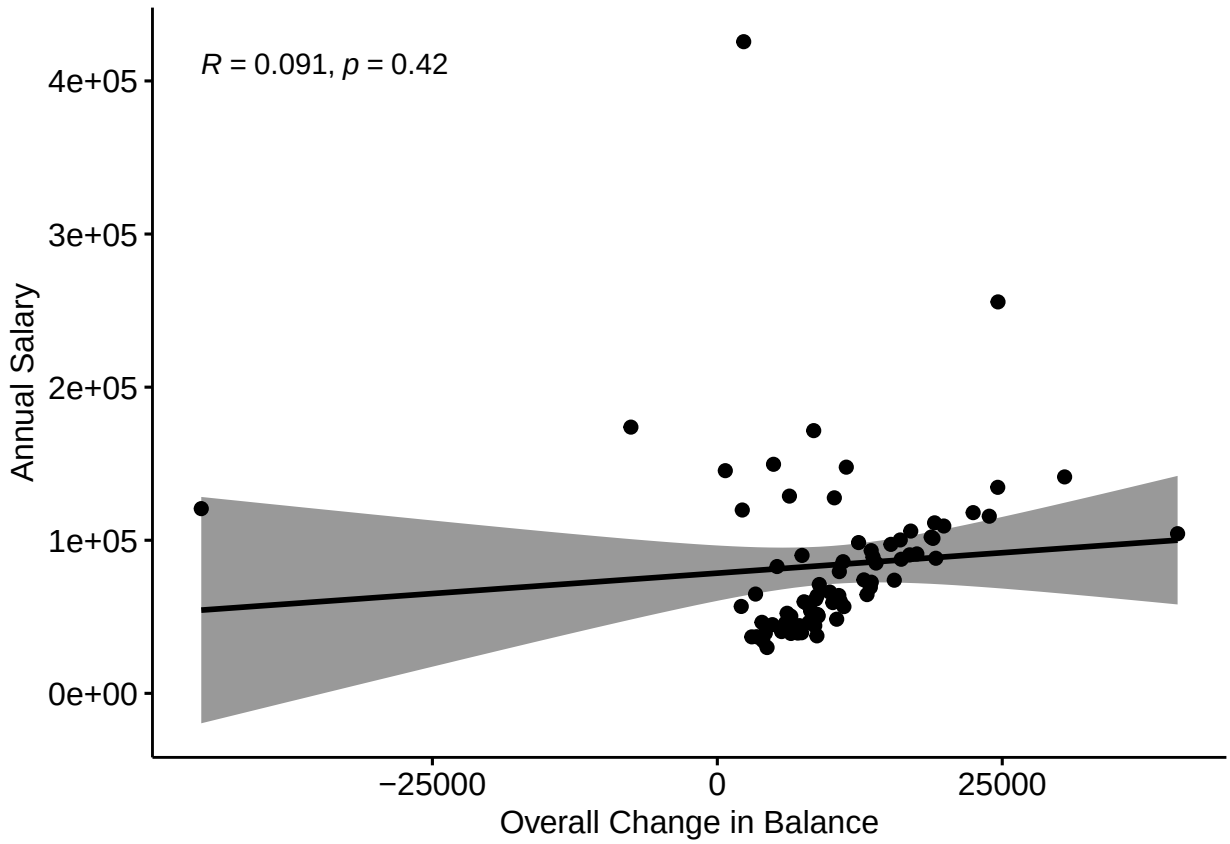

```
## `geom_smooth()` using formula 'y ~ x'
```




```
## `geom_smooth()` using formula 'y ~ x'
```




```
## `geom_smooth()` using formula 'y ~ x'
```


R = 0.091, p = 0.42
4e+05
3e+05
2e+05
1e+05
0e+00
Annual Salary
−25000
0
25000
Overall Change in Balance

Linear Regression Model on Train and Test data

```
# Create Training and Test data -
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(total_sal_anz), 0.8*nrow(total_sal_anz))  # row indices for training
train_x <- total_sal_anz[trainingRowIndex, ]  # model training data
test_x  <- total_sal_anz[-trainingRowIndex, ]   # test data

lrmodel <- lm(salary ~. , data = train_x)# build the model
salPred <- predict(lrmodel, test_x)  # predict salary

#prediction accuracy and error rates
actuals_preds <- data.frame(cbind(actuals=test_x$salary, predicteds=salPred))  # make actuals_predicted
correlation_accuracy <- cor(actuals_preds)  # 18.25%
paste0("Prediction Accuracy ",round(correlation_accuracy[1,2]*100,2),"%")
```

```
## [1] "Prediction Accuracy 6.5%"
```

```
## [1] "Not very good at all but still happy for my first attempt :)"
```

```
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
paste0("Min Max Accuracy ",round(min_max_accuracy*100,2),"%")
```

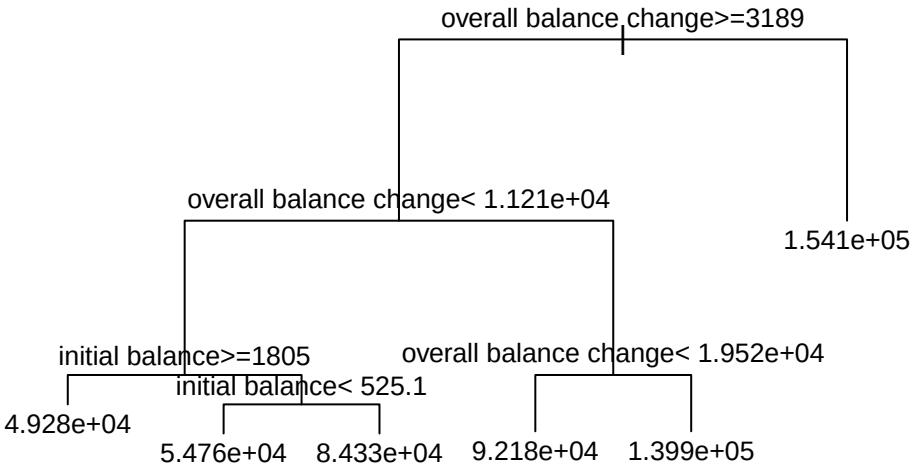
```
## [1] "Min Max Accuracy 10.19%"
```

```
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
paste0("Mean Absolute Percentage Deviation ",round(mape*100,2),"%")
```

```
## [1] "Mean Absolute Percentage Deviation 96.42%"
```

Decision TREE

```
anztree <- rpart(salary ~. ,data=total_sal_anz,method="anova")

plot(anztree,margin=0.1)
text(anztree,cex=.8)
```




```
plotcp(anztree)
```