# MAT 4800 PROJECT_1

Le’ Sean Roberts

## Project: PREDICTING CREDIT CARD BALANCE

### Summary

In this project, we will work with a simulated data set that contains information that we can use to create a model to predict customer credit card balance. A bank might use such information to predict which customers might be the most profitable to lend to (customers who carry a balance, but do not default, for example).

Specifically, we wish to build a model to predict credit card balance (Balance column) based on income (Income column) and credit rating (Rating column).

We access this data set by accessing it from an R data package called Credit that is available inside the R package ISLR2. Loading that package gives access to a variety of data sets, including the Credit data set that we will be working with.

```r
# First, loading required packages for exploratory data analysis and machine learning.
library(tidyverse)
library(tidymodels)
# Then having a small peak into the Credit data set.
library(ISLR2)

head(Credit)
```

```
   Income Limit Rating Cards Age Education Own Student Married Region Balance
1  14.891  3606    283     2  34        11  No      No     Yes  South     333
2 106.025  6645    483     3  82        15 Yes     Yes     Yes   West     903
3 104.593  7075    514     4  71        11  No      No      No   West     580
4 148.924  9504    681     3  36        11 Yes      No      No   West     964
5  55.882  4897    357     2  68        16  No      No     Yes  South     331
6  80.180  8047    569     4  77        10  No      No      No  South    1151
```

```r
dim(Credit)
```

```
[1] 400  11
```

```r
# Specifically, to build a model to predict credit card balance (Balance column) based on

prediction_data <- select(Credit, Balance, Income, Rating)
dim(prediction_data)
```

```
[1] 400   3
```

```r
credit <- as_tibble(prediction_data)
head(credit)
```

```
# A tibble: 6 x 3
  Balance Income Rating
    <dbl>  <dbl>  <dbl>
1     333   14.9    283
2     903  106.     483
3     580  105.     514
4     964  149.     681
5     331   55.9    357
6    1151   80.2    569
```

```r
dim(credit)
```

```
[1] 400   3
```

### Before performing exploratory data analysis, will create the training and testing data sets. First, splitting the credit data set with 60% of the data for training.

```r
# Splitting method for training and testing, and setting the Balance variable we want to p
credit_split <- initial_split(credit, prop = 0.6, strata = Balance)

# Now constructing training data and testing data.
credit_training <- training(credit_split)
```

```r
credit_testing <- testing(credit_split)
head(credit_training)
```

```
# A tibble: 6 x 3
  Balance Income Rating
    <dbl>  <dbl>  <dbl>
1       0   15.0    138
2       0   20.1    200
3       0   20.1    213
4       0   20.2    199
5      50   35.0    253
6       0   15.3    138
```

```r
dim(credit_training)
```

```
[1] 239   3
```

```r
head(credit_testing)
```

```
# A tibble: 6 x 3
  Balance Income Rating
    <dbl>  <dbl>  <dbl>
1     580  105.     514
2     964  149.     681
3    1350   71.1    491
4    1081   43.7    511
5       0   53.6    286
6     368   36.5    339
```

```r
dim(credit_testing)
```

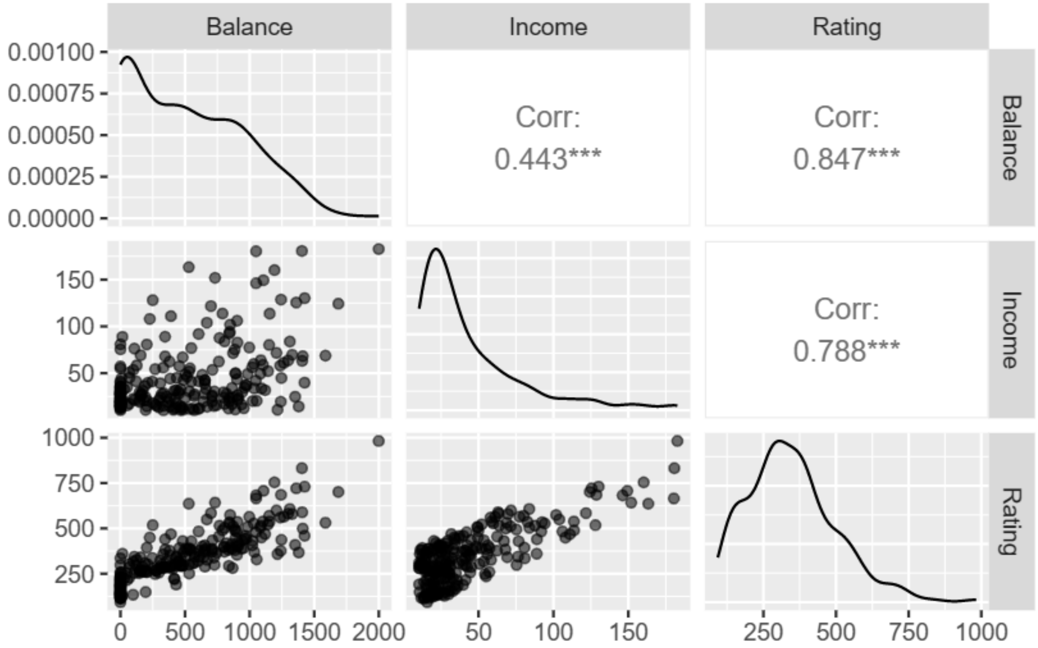
```
[1] 161   3
```

### For the observations in the training data set, creating a scatterplot of the variables we are interested in, including in our model.

```r
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```r
# Generating scatterplot of training data
credit_eda <- ggpairs(credit_training, aes(alpha = 0.4))
credit_eda
```



## STATEMENTS TO DETERMINE IF INCORRECT OR NOT:

### A. There is a strong positive relationship between the response variable (Balance) and the Rating predictor.

Answer: based on the observed correlation ## Corr(Balance, Rating) = 0.876 The value is positive, but value is under 0.9 but greater than 0.75, hence, there is a strong positive relationship between the response variable (Balance) and the Rating predictor. However, relationship should not be viewed as both dynamics being parallel. ## B. There is a strong relationship between the two predictors (Income and Rating) Answer: based on the observed correlation ## Corr(Income, Rating) = 0.814 The value is positive, but value is under 0.9 but greater than 0.75, hence, there is a strong positive relationship between the predictor variable (Income) and the the predictor variable (Rating). However, relationship should not be viewed as a both dynamics being parallel. ## C. There is a strong positive relationship between the response variable (Balance) and the Income predictor variable. Answer: based on the observed correlation ## Corr(Balance, Income) = 0.511 There is neither strong nor "moderately strong" relationship between the response variable (Balance) and the Income predictor variable.

### Fitting a Linear Regression Model based on Training Data

```
# Make the linear model specification
lm_spec<- linear_reg() |>
  set_engine("lm") |>
  set_mode("regression")

# Make a recipe. Acquire ingredients (variables of interest).
credit_recipe <- recipe(Balance ~ Income + Rating, data = credit_training)
```

### Model Specification and Recipe in a workflow, and Fit our Regression Model

```
# Put it all together in a workflow, then fit
credit_fit <- workflow() |>
  add_recipe(credit_recipe) |>
  add_model(lm_spec) |>
  fit(data = credit_training)
credit_fit
```

```
== Workflow [trained] ==========================================================
Preprocessor: Recipe
Model: linear_reg()

-- Preprocessor ----------------------------------------------------------------
0 Recipe Steps

-- Model -----------------------------------------------------------------------

Call:
stats::lm(formula = ..y ~ ., data = data)
```

```
Coefficients:
(Intercept)       Income       Rating
   -503.831       -7.440        3.864
```

## Based on above acquired coefficients from each of the predictors, the mathematical equation observed:

$ Credit Card Balance = -552.333 - 7.707 * Income + 3.992 * Credit Card Rating

## The Root Mean Squared Error (RMSE)

One of the two main performance indicators for a regression model. It measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy).

The lower the value of the Root Mean Squared Error, the better the model is. A perfect model (a hypothetic model that would always predict the exact expected value) would have a Root Mean Squared Error value of 0.

The Root Mean Squared Error has the advantage of representing the amount of error in the same unit as the predicted column making it easy to interpret. If you are trying to predict an amount in dollars, then the Root Mean Squared Error can be interpreted as the amount of error in dollars.

## Calculate the RMSE to assess goodness of fit

```r
# Pipeline mechanism "workflow" for RMSE
library(dplyr)
library(tidyverse)
library(tidymodels)
set.seed(2022)

lm_rmse <- credit_fit |>
  predict(credit_training) |>
  bind_cols(credit_training) |>
  metrics(truth = Balance, estimate = .pred) |>
  filter(.metric == 'rmse') |>
  select(.estimate) |>
  pull()
lm_rmse
```

```
[1] 173.3642
```

### Calculate the RMSPE to the test data

```
# Pipeline mechanism "workflow" for RMSPE
library (tidyverse)
library(tidymodels)
set.seed(2022)

lm_rmspe <- credit_fit |>
  predict(credit_testing) |>
  bind_cols(credit_testing) |>
  metrics(truth = Balance, estimate = .pred) |>
  filter(.metric == 'rmspe') |>
  select(.estimate) |>
  pull()
lm_rmspe
```

```
numeric(0)
```

## FINAL REMARKS

In this project a simulated data set was applied to create a model to predict customer credit card balance. Measures such as correlation exhibited relationships among variables, while a regression was built towards prediction. Such exploratory data analysis and machine learning serve as a good introduction.