

# Bonus - SA 2

---

Cuerdo, Naomi Hannah A., Percia, Kyte Daiter M. 2025-05-19

```
library(ISLR2)
library(keras)
library(tensorflow)
library(magrittr)
library(reticulate)
library(nnet)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:tensorflow':
##
##      train

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine
```

## Summary

This analysis uses the **Default** dataset from the ISLR2 package, which contains information on 10,000 individuals, whether they defaulted on a credit card payment. Key features include:

balance: the average credit card balance

income: annual income

student: whether the individual is a student

default: the target variable (Yes/No).

The objective of this study is to build predictive models to estimate the probability that an individual will default on their credit card. We compare two models:

1. Logistic Regression – a simple, interpretable linear classifier
2. Neural Network – a more flexible model capable of capturing nonlinear patterns

By evaluating model performance on a validation set, we aim to determine which approach offers better predictive accuracy while considering the trade-offs between complexity and interpretability.

## Exploratory Data Analysis

### Structure and Summary

```
data("Default")
summary(Default)
```

```
## default      student      balance      income
## No :9667      No :7056      Min.   :  0.0      Min.   : 772
```

```
## Yes: 333    Yes:2944    1st Qu.: 481.7    1st Qu.:21340
##                               Median : 823.6    Median :34553
##                               Mean   : 835.4    Mean   :33517
##                               3rd Qu.:1166.3    3rd Qu.:43808
##                               Max.    :2654.3    Max.    :73554
```

This provides an overview of the data types and summary statistics. From the table, there are a few observations:

default and student are categorical

balance and income are numeric

Most balances range from \$0–\$2,500

Income varies widely up to \$73,000+

## Class Distribution of Default

```
table(Default$default)
```

```
##
##    No    Yes
## 9667   333
```

```
prop.table(table(Default$default))
```

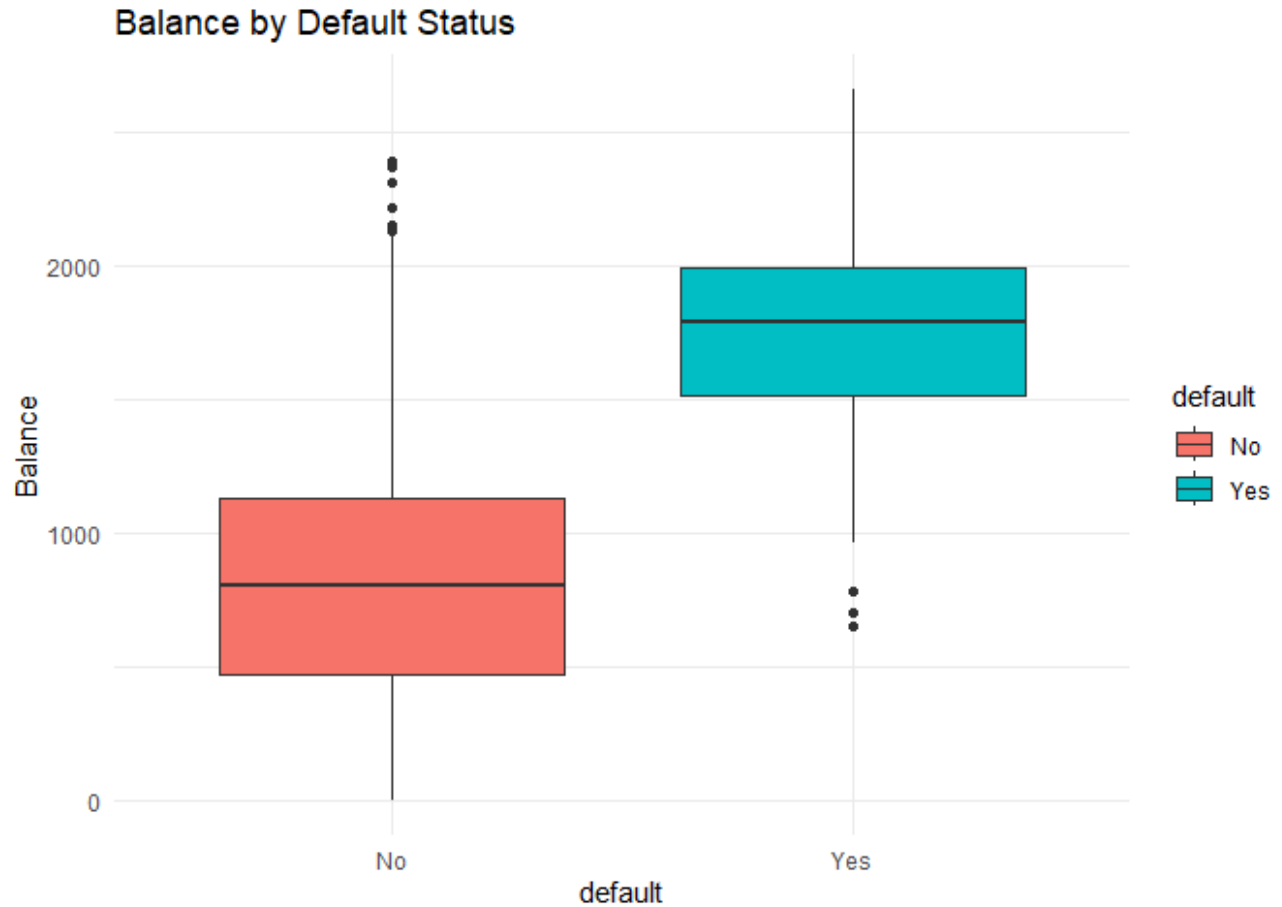
```
##
##      No      Yes
## 0.9667 0.0333
```

The dataset is imbalanced — only about 3.3% of customers default, while the rest do not. This imbalance should be kept in mind when evaluating model performance.

## Default vs Balance

```
ggplot(Default, aes(x = default, y = balance, fill = default)) +
  geom_boxplot() +
```

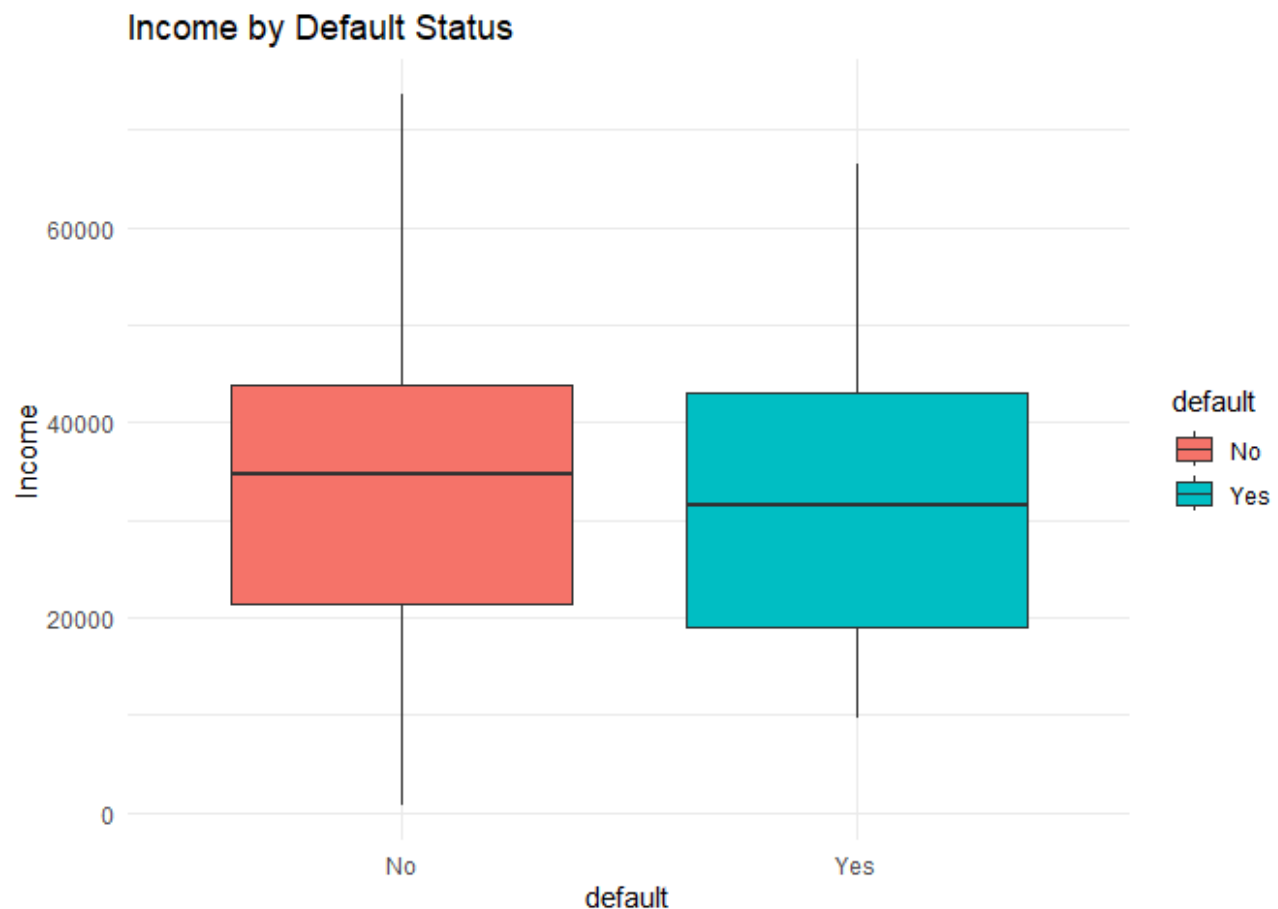
```
theme_minimal() +  
labs(title = "Balance by Default Status", y = "Balance")
```



From the plot above, it is evident that people who default tend to have much higher balances on average. This is a strong signal for predicting default.

## Default vs Income

```
ggplot(Default, aes(x = default, y = income, fill = default)) +  
  geom_boxplot() +  
  theme_minimal() +  
  labs(title = "Income by Default Status", y = "Income")
```



From the plot, it shows that there is no clear separation in income between defaulters and non-defaulters, suggesting income is a weaker predictor than balance.

## Correlation of Numeric Variables

```
cor(Default %>% select_if(is.numeric))
```

```
##           balance    income
## balance  1.0000000 -0.1522434
## income  -0.1522434  1.0000000
```

From the correlation matrix above, it shows that there is a **moderate positive correlation between balance and default**. There is also a **weak correlation** between income and default.

Thus, balance is the most informative numeric predictor.

## Comparing Linear Logistic Regression Model and Neural Network

## Preparing the Data

splitting the data into training and validation sets:

```
set.seed(1)
n <- nrow(Default)
ntest <- floor(n / 5)
testid <- sample(1:n, ntest)
y_test <- Default$default[testid] == 'Yes'

Default_train <- Default[-testid, ]
Default_test <- Default[testid, ]

Default_train$default <- as.factor(Default_train$default)
Default_test$default <- as.factor(Default_test$default)
```

## Training the Linear Logistic Regression Model


Let us now train a linear logistic regression model to predict the probability of default given the features student, balance, and income. The model estimates the conditional probability:

$$P(\text{Default}=\text{Yes}|X)$$

If the probability is greater than 0.5, the model predicts “Yes” (the person will default). Otherwise, it predicts “No” (the person will not default).

We then evaluate the model on the validation set by comparing the predictions to the actual default outcomes.

```
ll.reg <- glm(default~student+balance+income,family="binomial",data=Default[-testid,])
ll.pred <- predict(ll.reg, data=Default[testid,], type='response') > 0.5
ll.accuracy = mean(ll.pred == y_test)
ll.accuracy
```



```
## [1] 0.95225
```

From the result, the logistic regression model performs well on this validation set with an accuracy of 95.23%.

## Fitting a Neural Network

Now, we proceed with fitting a neural network

```
Default_train$student <- as.numeric(Default_train$student == "Yes")
Default_test$student  <- as.numeric(Default_test$student == "Yes")

preproc <- preProcess(Default_train[, c("balance", "income")], method = c("center", "
Default_train[, c("balance", "income")] <- predict(preproc, Default_train[, c("balanc
Default_test[, c("balance", "income")] <- predict(preproc, Default_test[, c("balance"
```

```
x = model.matrix(default ~. -1, data=Default)

x_train <- x[-testid,]
g_train <- Default$default[-testid]=='Yes'

x_test <- x[testid,]
g_test <- Default$default[testid] == 'Yes'

modnn <- keras_model_sequential()
modnn$add(layer_dense(units=10, activation='relu', input_shape = c(ncol(x_train))))
modnn$add(layer_dropout(rate=0.4))
modnn$add(layer_dense(units=1, activation='sigmoid'))
```

Now, we compile our model.

```
nn_model <- nnet(default ~ student + balance + income,
  data = Default_train,
  size = 5,
  decay = 0.1,
  maxit = 200,
  trace = FALSE)

nn_probs <- predict(nn_model, newdata = Default_test, type = "raw")
nn_pred  <- ifelse(nn_probs > 0.5, "Yes", "No")
nn_accuracy <- mean(nn_pred == Default_test$default)
nn_accuracy

## [1] 0.9705
```

The neural network achieves about 97.1% accuracy). ##### Final Comparison

```
accuracy_table <- data.frame(  
  Model = c("Logistic Regression", "Neural Network"),  
  Accuracy = c(ll_accuracy, nn_accuracy)  
)  
accuracy_table
```

```
##              Model Accuracy  
## 1 Logistic Regression  0.95225  
## 2      Neural Network  0.97050
```

From the table, it is clear that the **Neural Network** model achieved a higher accuracy (97.1%) compared to the **Logistic Regression model** (95.23%) on the validation set.

This suggests that the Neural Network was better able to capture complex patterns in the data, likely due to its ability to model nonlinear relationships. However, logistic regression still performed very well and is more interpretable.