HOMEWORK 4 – CLASSIFICATION

(Total points 10)

Classification

Business Context: Banks/Financial Companies

Assume you work as a data scientist for a bank or a financial institution (for example, Bank of America, Chase, etc.) and you are asked to analyze the dataset **credit.csv** containing information about a sample of customers that hold credit card accounts. You are asked to build a classification model that can help your institution predicting whether a customer will default or not, based on the available attributes.

Data Description

Each observation captures one individual; the following attributes are available:

- LIMIT_BAL: a continuous variable, capturing the credit limit on the customer's credit card
- GENDER: 1 = male, 2 = female
- MARRIAGE: 1 = married, 2 = single, 3 = other
- AGE: the age in years
- BILL_AMT1 to BILL_AMT6 (6 attributes in total): numerical variables containing the credit card balances for each of the previous 6 months. BILL_AMT1 is the most recent credit balance (for the last month) while BILL_AMT6 is the credit balance of 6 months ago.
- PAY_AMT1 to PAY_AMT6 (6 attributes in total): numerical variables containing the amount paid by the individual for each of the past 6 months. PAY_AMT1 is the most recent payment (so, payment made last month) while PAY_AMT6 is the amount paid 6 months ago.
- DEFAULT: the outcome variable or class variable. Default = NO, if the individual did not default this month; Default = YES if the individual did default.

Analysis

Perform the following analysis in R.

Data Preparation/Visualization

a) (0.3) Load the data. Use the command table() to get how many observations have Default = No and Default = Yes. What is the probability of default = YES?

b) (0.5) Use ggplot2 to create a histogram of Age, by Default. You histogram should proper title, axis labels and have different colors for Default = NO and Default = YES. Comment on the histogram. Is there any interesting relationship between Age and Default?

c) (0.5) Use ggplot2 to create a box-plot of LIMIT_BAL, by Default. You plot should have proper title and axis labels. When creating the graph, you can set your y = LIMIT_BAL/1000, so to make the numbers on the Y axis more interpretable. Also, after creating the box-plot, run the following code to get a sense of the mean and median values of LIMIT_BAL, by Default:

```
library(dplyr)
data %>%
```

```
group_by(DEFAULT) %>%
summarize(mean = mean(LIMIT_BAL/1000),
          median = median(LIMIT_BAL/1000))
```

Use the box-plot and the table produced by the code above to interpret the distribution of LIMIT_BAL by Default.

d) (0.2) Split the data into 80% training data and 20% test data. Simply report the code.

k-NN

e) (0.4) Next, you decide to train a k-NN model. First, assess whether you need to standardize the data. If yes, explain why and assess which attributes you should standardize and apply the standardization. If no, explain why. Report the code and your explanation.

f) (0.4) Train a k-NN model; use 5 different values of k (use 5 as lowest value and 40 as largest value. Pick any three values in between). Simply report the code. (Note: It may take a minute or so for the code to run. This is normal, since k-NN tends to be a slow method).

g) (0.5) Plot how the accuracy of the model changes with the value of k. Explain what accuracy captures; comment on which value of k is being picked by the algorithm, how the accuracy changes with the number of k.

h) (0.4) Get the (class) predictions from the k-NN model and produce the confusion matrix using only the option positive = "YES". Report the code and a screenshot of the Confusion Matrix and Statistics produced by the code.

i) (1) Consider the Results obtained in point h). First, comment on the overall accuracy of the k-NN model. Next, identify recall, precision and F1-Score for each class (Default No and Yes). Evaluate the performance of the classification model with respect to the individual classes and compare with the accuracy of the overall model.

Decision Trees

j) (0.4) Next, you decide to run a Decision Tree, using the same data partitions as for k-NN. First, decide whether to use the standardized data or not. Explain your choice. Next, train the decision tree. (It may take a minute or so for the code to run).

k) (0.2 for the plot only) Plot the decision tree and attach the plot. NOTE: if the tree obtained does not seem meaningful (for example, it only shows one node and no branches), try running again the code to train the decision tree. If that does not work, try creating a new data-partition and run the code again. It is ok if your tree is small or very large. On the base of the tree you obtain, answer the following questions:

- (0.3) Does your decision tree use all the attributes available in your data? Describe which attributes your tree is using and explain why it may be excluding some.
- (0.2) How many leaf nodes does your tree have?

- (0.8) Pick 2 leaf nodes, one with outcome NO and one with outcome YES and interpret each piece of information reported inside the leaf node (If you do not have leaf nodes with YES, pick any two).

l) (0.3) Get the class level predictions and produce the confusion matrix using the option positive = "YES". Report the code and a screenshot of the Confusion Matrix and Statistics produced by the code.

m) (1) Consider the Results obtained in point l). First, comment on the overall accuracy of the decision tree model. Next, identify recall, precision and F1-Score for each class (Default No and Yes). Evaluate the performance of the classification model with respect to the individual classes and compare with the accuracy of the overall model.

Evaluate Results

n) (0.8) Compare the results obtained for k-NN in points h) and i) with the results obtained for the Decision Tree in point l) and m). Based on those results, which model would you suggest your institution to use for customers' classification and why?

o) (1.8) A colleague in your team suggests to slightly revise the Decision Tree model, by getting the predicted probabilities (rather than just the class predictions) and setting a cut-off of 0.75 for the NO class. Run the code to achieve what your colleagues is suggesting. Then, produce and report the confusion matrix using the predictions obtained with the new cut-off and set again positive = "YES". Assess the performance of this modified version of the decision tree, commenting on the accuracy for the overall model and on recall, precision and F1-score for the individual classes. Interpret the results, and compare them to the results obtained for the original decision tree. Do you think that the suggestion made by your colleague make sense? Why or why not? Choose which Decision tree (between original version and this new version) you would prefer. And compare your chosen decision tree with the k-NN model. Does your answer from point n) changes?