BUSA3020 - Assignment 1

Assignment Points: 100
Due Date: Friday Week 7 (6 September 2024) @ 11.59pm
Instructions: Provide answers within this Jupyter notebook using Python code, and submit via iLearn
Marking Criteria: See end of document below
Assignment Background: This assignment uses a dataset which is based on the Credit Card Defaults data discussed in Week 2 Tutorial

Put **all your work** into this file and name it `Task1_ID.ipynb` where ID is your Macquarie University student ID number

- E.g. if MQ_ID == 12345678 then you need to submit Task1_12345678.ipynb;

Problem 1 - Reading the dataset (Total Marks: 20)

Q1. Read the first 10,000 rows from the credit card dataset provided in the **assignment_data** folder

- Name your DataFrame `df`
- Rename the column 'PAY_0' to 'PAY_1' and the column 'default payment next month' to 'payment_default'
- Delete `ID` column

(5 marks)

```
# ---- provide your code here -----
```

Q2. List which *features* are *numeric*, *ordinal*, and *nominal* variables, and how many features of each kind there are in the dataset. To answer this question

- Find the definitions of the variables provided elsewhere in the course material (hint: make sure you do weekly tutorials)
- Find the definitions of numeric, ordinal and nominal variables
- Carefully consider the values of the data itself as well as the output of `df.info()`.

Your answer should be written up in Markdown and include: 1) Definitions of the three kinds of variables, 2) A table listing all the features present in the dataset and their type (fill out the table template provided below) and 3) A brief description of the contents of the table.

Variable Kind	Number of Features	Feature Names
Numeric	some number	some text
some text	some number	some text
some text	some number	some text

(10 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Q3. Missing Values.

- Print out the number of missing values for each variable in the dataset and comment on your findings.

(5 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Problem 2. Cleaning data and dealing with categorical features (Total Marks: 40)

Q1.

- Use an appropriate `pandas` function to impute missing values using one of the following two strategies: `mean` and `mode`. (10 marks)
  - Take into consideration the type of each variable and the best practices we discussed in class/lecture notes
- Explain what data imputation is, how you have done it here, and what decisions you had to make. (5 marks)

(Total: 15 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Q2.

- Print `value_counts()` of the 'SEX' column and add a dummy variable named 'SEX_FEMALE' to `df` using `get_dummies()` (3 marks)
- Carefully explain what the values of the new variable 'SEX_FEMALE' mean (2 mark)
- Make sure the variable 'SEX' is deleted from `df`

(Total: 5 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Q3. Print `value_counts()` of the 'MARRIAGE' column and *carefully* comment on what you notice in relation to the definition of this variable.

(Total: 5 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Q4.

- Apply `get_dummies()` to 'MARRIAGE' feature and add dummy variables 'MARRIAGE_MARRIED', 'MARRIAGE_SINGLE', 'MARRIAGE_OTHER' to `df`. (5 marks)

- Carefully consider how to allocate all the values of 'MARRIAGE' across these 3 newly created features (5 marks)
  - Explain what decisions you had to make
- Make sure that 'MARRIAGE' is deleted from `df`

(Total: 10 marks)

```
# ---- provide your code here -----
```

---- provide your text answer here ----

Q5. In the column 'EDUCATION', convert the values {0, 5, 6} to the value 4.

(Total: 5 marks)

```
# ---- provide your code here -----
```

Problem 3 Preparing X and y arrays (Total Marks: 10)

Q1.

- Create a numpy array `y` from the first 8,000 observations of 'payment_default' column from `df` (2.5 marks)

- Create a numpy array `X` from the first 8,000 observations of all the remaining variables in `df` (2.5 marks)

(Total: 5 Marks)

```
# ---- provide your code here -----
```

Q2.

- Use an appropriate `sklearn` library we used in class to create `y_train`, `y_test`, `X_train` and `X_test` by splitting the data into 75% train and 25% test datasets (2.5 marks)
  - Set random_state to 4 and stratify the subsamples so that train and test datasets have roughly equal proportions of the target's class labels
- Standardise the data to mean zero and variance one using an approapriate `sklearn` library (2.5 marks)

(Total: 5 marks)

```
# ---- provide your code here -----
```

Problem 4. Support Vector Classifier and Accuracies (Total Marks: 30)

Q1.

- Train a Support Vector Classifier on the standardised data (5 marks)
  - Use `rbf` kernel and set `random_state` to 3 (don't change any other parameters)
- Compute and print training and test dataset accuracies (5 marks)

(Total: 10 marks)

```
# ---- provide your code here -----
```

Q2.

- Extract 2 linear principal components from the standardised features using an appropriate `sklearn` library (5 marks)
- Train a Support Vector Classifier on the 2 principal components computed above (5 marks)
  - Use `rbf` kernel and set `random_state` to 3 (don't change any other parameters)
- Compute and print training and test dataset accuracies (5 marks)

(Total: 15 marks)

```
# ---- provide your code here -----
```

Q3.

- Comment on the suitability of the two classifiers to predict credit card defaults by commenting on (and comparing) the computed accuracies from the last two questions.
- Make comparisons both within and across the two questions

(Total: 5 marks)

---- provide your text answer here ----

Marking Criteria

To achieve a perfect score, your solutions must adhere to the criteria outlined below:

- Ensure that all numerical answers are accurate.
- Use the specific Python functions and libraries specified within the assignment instructions.
- For any written responses, provide accurate information, articulated in clear and complete sentences.
- Do not add extra cells beyond what is provided in the notebook.
- Do not print output with your code unless explicitly instructed to do so.
- Maintain a clean and organised notebook layout that is easy to follow.
