# Python Project - Data Analytics and Machine Learning

## Topic:

Name: Chris Doyle
Employee Number: 92885

## Statement of Authentication:

By submitting this assignment, I consent that this work is my own, and where I have made use of the ideas/work of another individual, I have made appropriate acknowledgment to the original author of said work in the bibliography of this assignment.

All workings and tables can be found in the files attached with this submission.

Signed:

*Chris Doyle*

# 1. Abstract

## 2. Introduction

Credit card debit has become an increasingly more relevant topic in the banking industry, with the average cost of default estimated at around $7,000 USD per default in the US alone (Ma, 2020, pp. 231).

This sentiment was echoed by Alam *et al.* (2020, pp. 201173), who identifies how credit card debit reached an all time high over from 2015-2020, surpassing rates seen since the 2008 recession. They also emphasize monitoring and predictive analytics to track the tendency of clients to fall behind on loan payments, also known as the credit card delinquency rate.

In response to this discovery, the following project will attempt to implement machine learning and predictive modeling techniques associated with classification to try and predict the likelihood of a client defaulting on their credit card debt. This will be achieved using the CRISP-DM methodology (Chapman,2000) using a dataset provided by Yeh (2018) showing the default rates of a series of credit card holders within the Tiwanese market during 2005.

## 3. Data

The dataset employed in this project has been provided by Yeh (2018) derived from data provided by the Taiwanese banking industry, available here. For this reason, all monetary amounts in this project will be in New Taiwanese Dollars (NTD).

The target feature is “default payment next month”, which denotes that in the following month a client defaulted on the principal sum owed. The remaining features are a mix of numeric and categorical features that will be used as input features.

A comprehensive description of the dataset is available in Appendix 1.

## 4. Importing

The original dataset was originally stored as workbook 97-2003 or 'xls', which was not fully compatible with pandas. To ensure compatibility, please install xlrd-2.0.1-py2 using either Conda or pip (see figure 4.1). As a precaution, a standard excel workbook (xlsx) version of the dataset has been included in the submission.

Figure 4.1: Using PIP to install xlrd-2.0.1-py2

```
(base) C:\Users\resig>pip install xlrd
Defaulting to user installation because normal site-packages is not writeable
Collecting xlrd
  Downloading xlrd-2.0.1-py2.py3-none-any.whl.metadata (3.4 kB)
Downloading xlrd-2.0.1-py2.py3-none-any.whl (96 kB)
Installing collected packages: xlrd
Successfully installed xlrd-2.0.1
```

The data was imported as a Pandas Dataframe, *dfInital*, but the header had been imported as the first row of the dataset, with all the target features being labeled X1, x2, x3, etc, and the target feature labels as Y1 (see figure 4.2). To resolve this, the headers in the dataset were replaced with the values in row 1 while all other rows were moved to a new variable. (see figure 4.3).

Figure 4.2: Initial DQ Issues with respect to column headers

```
# Check file imported correctly
dfInital.head(5)
```

| | Unnamed: 0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | ID | LIMIT_BAL | SEX | EDUCATION | MARRIAGE | AGE | PAY_0 | PAY_2 | PAY_3 | PAY_4 | ... B |
| **1** | 1 | 20000 | 2 | 2 | 1 | 24 | 2 | 2 | -1 | -1 | ... |
| **2** | 2 | 120000 | 2 | 2 | 2 | 26 | -1 | 2 | 0 | 0 | ... |
| **3** | 3 | 90000 | 2 | 2 | 2 | 34 | 0 | 0 | 0 | 0 | ... |
| **4** | 4 | 50000 | 2 | 2 | 1 | 37 | 0 | 0 | 0 | 0 | ... |

5 rows × 25 columns

Figure 4.3: Dataset with corrected names.

```
# Remove top row from dataset
headers = dfInital.iloc[0]
df =  dfInital.iloc[1:].copy()
df.columns = headers
```

```
df
```

| | ID | LIMIT_BAL | SEX | EDUCATION | MARRIAGE | AGE | PAY_0 | PAY_2 | PAY_3 | PAY_4 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 20000 | 2 | 2 | 1 | 24 | 2 | 2 | -1 | -1 | ... |
| **2** | 2 | 120000 | 2 | 2 | 2 | 26 | -1 | 2 | 0 | 0 | ... |
| **3** | 3 | 90000 | 2 | 2 | 2 | 34 | 0 | 0 | 0 | 0 | ... |
| **4** | 4 | 50000 | 2 | 2 | 1 | 37 | 0 | 0 | 0 | 0 | ... |
| **5** | 5 | 50000 | 1 | 2 | 1 | 57 | -1 | 0 | -1 | 0 | ... |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **29996** | 29996 | 220000 | 1 | 3 | 1 | 39 | 0 | 0 | 0 | 0 | ... |
| **29997** | 29997 | 150000 | 1 | 3 | 2 | 43 | -1 | -1 | -1 | -1 | ... |
| **29998** | 29998 | 30000 | 1 | 2 | 2 | 37 | 4 | 3 | 2 | -1 | ... |
| **29999** | 29999 | 80000 | 1 | 3 | 1 | 41 | 1 | -1 | 0 | 0 | ... |
| **30000** | 30000 | 50000 | 1 | 2 | 1 | 46 | 0 | 0 | 0 | 0 | ... |

The dataset contained 30,000 rows, with 25 features (including the target features, see figure 4.4). The dataset was now ready to be analysed, cleaned, and subject to the first round of feature engineering.

Figure 4.4

```
# Check new shape is applied correctly
shapeNew = df.shape
shapeNew
```

```
(30000, 25)
```

# 5. Data Preparation

## Part 1: Data Inspection

The data was inspected for the correct data types (see figure 5.1.1). All features were imported as objects due to the issues with the headers in *3. Importing*. The numeric features identified in Appendix 1 were converted from object to numeric (see figure 5.1.2). Following this step, the dataset was temporarily separated into numeric and categorical (see figure 5.1.3) to better derive summary statistics.

Figure 5.1.1: Data types initial inspection

```
[85]: #Check Data types:
      df.dtypes
```

```
[85]: 0
      ID                          object
      LIMIT_BAL                   object
      SEX                         object
      EDUCATION                   object
      MARRIAGE                    object
      AGE                         object
      PAY_0                       object
      PAY_2                       object
      PAY_3                       object
      PAY_4                       object
      PAY_5                       object
      PAY_6                       object
      BILL_AMT1                   object
      BILL_AMT2                   object
      BILL_AMT3                   object
      BILL_AMT4                   object
      BILL_AMT5                   object
      BILL_AMT6                   object
      PAY_AMT1                    object
      PAY_AMT2                    object
      PAY_AMT3                    object
      PAY_AMT4                    object
      PAY_AMT5                    object
      PAY_AMT6                    object
      default payment next month  object
      dtype: object
```

Figure 5.1.2: Code used to convert objects to numeric features

```
# Convert the columns identified above to numeric

# Get list of numeric data types
payColumns = ["PAY_{}".format(x) for x in range(0,7)]
billColumns = ["BILL_AMT{}".format(x) for x in range(1,7)]
payAmtColumns = ["PAY_AMT{}".format(x) for x in range(1,7)]

numericColumns = ['LIMIT_BAL']+payColumns+billColumns+payAmtColumns
numericColumns.remove('PAY_1')

# Apply numeric data types
for i,col in enumerate(numericColumns):
    df[col] = pd.to_numeric(df[col])
```

Figure 5.1.3: Separation of Dataset into numeric and categorical

```
# Seperate into Numeric and Categorical features
dfNumeric = df.select_dtypes(include='number')
dfCategorical = df.select_dtypes(include='object')
```

The categorical summary statistics were generated, and value counts of each feature was taken (see figure

- Discuss the steps taken to prepare the data for analysis.
- Explain the creation of pandas DataFrames.
- Describe the sorting, indexing, filtering, and grouping operations performed on the data.
- Explain how duplicate entries and missing values were handled.
- Discuss the definition of custom functions for reusable code.
- Provide details on how multiple DataFrames were merged, if applicable.

## 6. Data Visualization

The

Generate at least four charts using the Matplotlib library.

- Generate at least four charts using the Seaborn library.
- Conduct univariate and bivariate analysis using appropriate charts and techniques.

## 7. Machine Learning

The

- Predict a target variable with a Supervised or Unsupervised algorithm.
- Implementation of Supervised or Unsupervised Model/s.
- Perform Model Evaluation using metrics suitable for the choice of ML model/s.
- Perform hyper parameter tuning or boosting, whichever is relevant to the model. If it is not relevant, justify that in your report and Python comments

## 8. Insights

The

- Derive at least eight valuable insights from your data analysis.
- Justify each insight with reference to the charts or analysis performed.

## 9. Results and Conclusion

The

- Summarize the key findings and insights obtained from the project.

# Appendix 1:

This provides a detailed description of the dataset used in the course of this project. It is adapted from the original script provided by the author Yeh (2018).

## Original Features:

| Name | Type | Description |
|---|---|---|
| ID | Numeric (Discrete) | An anonymous unique identifier for each client, running from 1 to 30,000 |
| LIMIT_BAL | Numeric (Discrete) | The overdraft limit (NTD) a client had access to at the time of the analysis. |
| SEX | Boolean | Denotes the gender of each client, (1 = male; 2 = female). |
| EDUCATION | Categorical (Nominal) | Denotes the highest level of education obtained by the client, with (1 = graduate school; 2 = university; 3 = high school; 4 = others).<br><br>It is unknown what encompasses others (e.g. DNF high school, trade school, professional accreditation). |
| MARRIAGE | Categorical (Nominal) | Denotes the marital status of the client, (1 = married; 2 = single; 3 = others). |
| AGE | Numeric (Discrete) | The age of the client, in years. |
| PAY_0 | Categical (Ordinal) | History of past payment within the last month.<br><br>For features PAY_0 to PAY_6, the features use the following nomenclature:<br><br>**-2:** No payment recorded/due<br>**-1**: Pay duly (no delay)<br>**1** : Payment delay for one month<br>**2** : Payment delay for two months<br>**3** : Payment delay for three months<br>**4** : Payment delay for four months<br>**5** : Payment delay for five months<br>**6** : Payment delay for six months<br>**7** : Payment delay for seven months<br>**8** : Payment delay for eight months<br>**9** : Payment delay for nine months or more |

| | | |
|---|---|---|
| PAY_2 | Categical (Ordinal) | History of past payment two months ago. |
| PAY_3 | Categical (Ordinal) | History of past payment three months ago. |
| PAY_4 | Categical (Ordinal) | History of past payment four months ago. |
| PAY_5 | Categical (Ordinal) | History of past payment five months ago. |
| PAY_6 | Categical (Ordinal) | History of past payment six months ago. |
| BILL_AMT1 | Numeric (Discrete) | Amount (NTD) of bill statement in 1st month. |
| BILL_AMT2 | Numeric (Discrete) | Amount (NTD) of bill statement in 2nd month. |
| BILL_AMT3 | Numeric (Discrete) | Amount (NTD) of bill statement in 3rd month. |
| BILL_AMT4 | Numeric (Discrete) | Amount (NTD) of bill statement in 4th month. |
| BILL_AMT5 | Numeric (Discrete) | Amount (NTD) of bill statement in 5th month. |
| BILL_AMT6 | Numeric (Discrete) | Amount (NTD) of bill statement in 6th month. |
| PAY_AMT1 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the previous month was outstanding). |
| PAY_AMT2 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the two months was resolved). |
| PAY_AMT3 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the three months was resolved). |
| PAY_AMT4 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the four months was resolved). |
| PAY_AMT5 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the five months was resolved). |

| | | |
|---|---|---|
| PAY_AMT6 | Numeric (Discrete) | Amount of previous payment made (i.e., how much of their outstanding balance from the six months was resolved). |
| default payment next month | Boolean | Denotes if the client defaulted on the outstanding balance or not (1 if they defaulted, 0 if they did not default). |

# Appendix 2: Data Cleaning and Feature Engineering

## Original Features:

| Name | Issue(s) | Solution |
|---|---|---|
| ID | - Redundant feature. Only purpose is to separate data points from one another. | - Delete |
| LIMIT_BAL | - 1. Stored as object rather than numeric | - 1. Convert to numeric |
| SEX | - Needs to be converted into a form that can be fed into an ML model. | - Apply one-hot encoding (see *Engineered Features*) |
| EDUCATION | - 1. Needs to be converted into a form that can be fed into an ML model.<br>- 2. Website lists only 4 potential varials, but 7 observed within the dataset. | - 1. Apply one-hot encoding (see *Engineered Features*)<br>- 2. [FIND A SOLUTION] |
| MARRIAGE | - 1. Needs to be converted into a form that can be fed into an ML model.<br>- 2. Website lists only 3 potential varials, but 4 observed within the dataset. | - 1. Apply one-hot encoding (see *Engineered Features*)<br>- 2. [FIND A SOLUTION] |
| AGE | - No issues | |
| PAY_0 - PAY_6 | - 1. Stored as object rather than numeric | - 1. Convert to numeric |
| BILL_AMT1 - BILL_AMT6 | - 1. Stored as object rather than numeric | - 1. Convert to numeric |
| PAY_AMT1 - PAY_AMT6 | - 1. Stored as object rather than numeric | - 1. Convert to numeric |
| default payment next month | - Name contains white spaces | - rename to default_payment_next_month |

## Engineered Features:

| Name | Type | Description |
|---|---|---|
| gender_male | Boolean | A one-hot encoded version of the Sex feature, with 1 denoting male and 0 denoting female. |
| | | |
| | | |
| | | |

## Bibliography:

Alam, T.M,. *et al.* (2020) “An Investigation of Credit Card Default Prediction in the Imbalanced Datasets”, *IEEE Xplore*, vol 8, November, pp.  201173 - 201198.

Chapman, P. (2000) *CRISP-DM 1.0: Step-by-step data mining guide*. Copenhagen: CRISP-DM Consortium.

Ma, Y. (2020) “Prediction of Default Probability of Credit-Card Bills”, *Open Journal of Business and Management*, vol 8, pp. 231-244.

Yeh, C. I. (2018) *Default of Credit Card Clients*. Available at: https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients (Accessed 14 April 2025).

NOTES:

XG BOOST:

WORKS WHEN NOTHING ELSE WORKS