Predicting Credit Card Defaults with Machine Learning Algorithm Using Customer Database

Anushka[1], Nidhi Agarwal[1,2], Devendra K. Tayal[1], Vrinda Abrol[1](✉), Deepakshi[1], Yashica Garg[1], and Anjali Jha[1]

[1] Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, Kashmere Gate, India
{anushka017btcse21,vrinda047btcse21,deepakshi120btcse21, yashica048btcse21,anjali148btcse20}@igdtuw.ac.in
[2] Department of CSE, Delhi Technical Campus, Greater Noida, India
n.agarwal@delhitechnicalcampus.ac.in

Abstract. In the banking sector, credit risk is a significant factor. Banking's main activities include granting loans, credit cards, investments, mortgages, etc. Credit cards are one of the fastest growing financial services offered by banks in recent years. However, as the number of credit card users increases, banks are facing rising credit card failure rates. Therefore, data analytics can provide solutions to address current phenomena and manage credit risk. This document provides a performance evaluation of credit card default prediction. In this work, a prediction model for credit card defaulters was developed utilising a variety of unconnected decision trees. It helps speculate if someone might be a defaulter and helps the bank decide the credit limit for customers.

Keywords: Credit Card Defaulter · Taiwanese Bank · Decision Tree · Random Forest Regressor · Prediction Model · Credit Cards · Data Model

1 Introduction

The demand for financial services has grown significantly over the past few years, creating a vast amount of data in terms of bulk, accuracy, and uniqueness [1]. In order to enjoy certain goods in advance, people are increasingly likely to spend money up front and mortgage their "credit" to the bank. However, when engrossed, people frequently exhibit irrational behaviour and overestimate their capacity to make timely loan repayments to banks. On the one hand, it raises banks' loan risk. On the other hand, it worsens the credit crisis that customers are already experiencing.

When you consistently fail to pay your credit card's minimum required amount, it becomes a payment default. The card issuer typically sends the default notice following 6 consecutive missed payments. Your credit score is compromised, your credit card is blocked, and you could land yourself on a blacklist if you are flagged as a credit card defaulter. You may also face legal repercussions, and in some cases, this circumstance results in the acquisition of assets [2].

S. Nandan Mohanty et al. (Eds.): ICISML 2022, LNICST 470, pp. 262–277, 2023.
https://doi.org/10.1007/978-3-031-35078-8_22

This bank in Taiwan issued credit cards to many customers. They tracked credit card defaulters from April 2005 to September 2005 and stored it in a dataset that they kept up to date [3]. All banks would benefit greatly from being able to identify which clients are most likely to default [4]. This aids in determining the customer's authenticity and can enhance the consumer screening process [5]. Additionally, it aids the bank in lowering the credit card holder's spending limit or starting legal action to recover the debt.

2 Methodology and Experimentation

2.1 Importing Modules and Reading Data

Let's import the requisite Python modules, read the information from a csv file to make a Pandas DataFrame, and do a data-cleaning process (if required) (Figs. 1, 2).

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default.payment.next.month
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0.0	0.0	0.0	0.0	689.0	0.0	0.0	0.0	0.0	1
1	2	120000.0	2	2	2	26	-1	2	0	0	...	3272.0	3455.0	3261.0	0.0	1000.0	1000.0	1000.0	0.0	2000.0	1
2	3	90000.0	2	2	2	34	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	1500.0	1000.0	1000.0	1000.0	5000.0	0
3	4	50000.0	2	2	1	37	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	2019.0	1200.0	1100.0	1069.0	1000.0	0
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	20940.0	19146.0	19131.0	2000.0	36681.0	10000.0	9000.0	689.0	679.0	0

5 rows × 25 columns

Fig. 1. First five records in the database. Now let's identify the total number of rows and columns, the data types of the columns, and the missing values in the dataset.

The dataset [6] consists of 30,000 rows, 25 columns, and 0 missing (or null) values. We won't need to encode any non-numeric values into numeric values because all the columns have numeric values.

Here, we can observe that there are no white spaces in the column names. We shall rename PAY 0 to PAY 1 because we can also see that the column names in the payback condition are inconsistent. For simplicity, the target variable's column name, default.payment.next.month, can be modified to something shorter, such DEFAULT (Fig. 3).

We can observe that:

- The changed value for "PAY 0" is "PAY 1."
- Changed from "default.payment.next.month" to "DEFAULT"

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   30000 non-null  float64
 2   SEX                         30000 non-null  int64
 3   EDUCATION                   30000 non-null  int64
 4   MARRIAGE                    30000 non-null  int64
 5   AGE                         30000 non-null  int64
 6   PAY_0                       30000 non-null  int64
 7   PAY_2                       30000 non-null  int64
 8   PAY_3                       30000 non-null  int64
 9   PAY_4                       30000 non-null  int64
 10  PAY_5                       30000 non-null  int64
 11  PAY_6                       30000 non-null  int64
 12  BILL_AMT1                   30000 non-null  float64
 13  BILL_AMT2                   30000 non-null  float64
 14  BILL_AMT3                   30000 non-null  float64
 15  BILL_AMT4                   30000 non-null  float64
 16  BILL_AMT5                   30000 non-null  float64
 17  BILL_AMT6                   30000 non-null  float64
 18  PAY_AMT1                    30000 non-null  float64
 19  PAY_AMT2                    30000 non-null  float64
 20  PAY_AMT3                    30000 non-null  float64
 21  PAY_AMT4                    30000 non-null  float64
 22  PAY_AMT5                    30000 non-null  float64
 23  PAY_AMT6                    30000 non-null  float64
 24  default.payment.next.month  30000 non-null  int64
dtypes: float64(13), int64(12)
memory usage: 5.7 MB
```

Fig. 2. The total number of rows and columns, column data types, and missing values (if any) in the dataset are displayed.

2.2 Data Analysis

The primary components of this data set are as follows:

- The status of six prior bill cycles' payments.
- The amount owed, the repayment status, and the total amount paid.
- The clientele's data on demographics.

This dataset's target column, default.payment.next.month, divides the customers into two groups:

1 (yes) - indicates that the client intends to default by failing to pay the bill for the following month.

0 (no) - indicates that the customer will make the payment in full the following month and will not be considered in default.

Let's examine the data now to see if there are any patterns in the actions of defaulters.

To determine whether defaulters are more prevalent than non-defaulters, let's count both groups (Fig. 4).

```
At index 00, ID
At index 01, LIMIT_BAL
At index 02, SEX
At index 03, EDUCATION
At index 04, MARRIAGE
At index 05, AGE
At index 06, PAY_1
At index 07, PAY_2
At index 08, PAY_3
At index 09, PAY_4
At index 10, PAY_5
At index 11, PAY_6
At index 12, BILL_AMT1
At index 13, BILL_AMT2
At index 14, BILL_AMT3
At index 15, BILL_AMT4
At index 16, BILL_AMT5
At index 17, BILL_AMT6
At index 18, PAY_AMT1
At index 19, PAY_AMT2
At index 20, PAY_AMT3
At index 21, PAY_AMT4
At index 22, PAY_AMT5
At index 23, PAY_AMT6
At index 24, DEFAULT
```

Fig. 3. Table after Renaming column names

```
0    23364
1     6636
Name: DEFAULT, dtype: int64
```

Fig. 4. Screenshot showing counts of defaulters and non-defaulters

23,346 out of 30,000 credit card customers make on-time payments (non-defaulters) while 6,636 do not (defaulters). Let's look at the defaulter to non-defaulter ratio (Fig. 5).

Therefore, only approximately 22% of Taiwanese bank's credit card customers miss their due dates for payment.

Let's now determine the proportion of clients who are male and female among all the defaulters. For this,

Take a subset of the data frame that only has the "ID," "SEX," and "DEFAULT" columns.

```
0    0.7788
1    0.2212
Name: DEFAULT, dtype: float64
```

Fig. 5. Screenshot showing ratio of defaulters and non-defaulters

Sort the "DEFAULT" and "SEX" columns in the above sliced data set (Fig. 6).

		ID
DEFAULT	**SEX**	
0	**1**	9015
	2	14349
1	**1**	2873
	2	3763

Fig. 6. Screenshot showing number of female and male defaulters

2873 of the 6636 defaulters were men and 3763 were women.

Let's estimate the proportion of credit card users with various levels of education among all the defaulters (Fig. 7).

		ID
DEFAULT	**EDUCATION**	
0	**0**	14
	1	8549
	2	10700
	3	3680
	4	116
	5	262
	6	43
1	**1**	2036
	2	3330
	3	1237
	4	7
	5	18
	6	8

Fig. 7. Screenshot showing number of defaulters on the basis of education level

Majority of the 6636 defaulters are classified as:

3330 of them had a college degree.
2036 students received diplomas.
1237 of them had completed high school.

The fact that the majority of credit clients are university graduates may be the cause of the higher percentage of defaulters who hold a degree. By counting the various categorical values in the "EDUCATION" column, let's confirm this suspicion (Fig. 8).

```
2    14030
1    10585
3     4917
5      280
4      123
6       51
0       14
Name: EDUCATION, dtype: int64
```

Fig. 8. Screenshot showing number of different categorical values

As we can see, our suspicions were confirmed because the majority of credit clients are university graduates, followed by high school graduates and then graduates of graduating schools.

Here, we can see some redundant information in the EDUCATION column. According to the dataset description:

1. graduate school
2. university
3. high school
4. others
5. unknown
6. unknown

This explicitly shows that we may combine labels 5 and 6 in order to options for schools and swap out the undesirable values in the column. The label 4 indicates that a consumer has additional education, such as vocational training, a certificate, a degree, etc.

It will help us in simplifying the final decision tree and speeding up the computing process.

The syntax used is: pandas.DataFrame.loc [boolean_condition, column_name] = new_value (Fig. 9).

```
2    14030
1    10585
3     4917
5      345
4      123
Name: EDUCATION, dtype: int64
```

Fig. 9. Removing redundancy in EDUCATION column

Let's estimate the number of credit card users based on their marital status among all the defaulters (Fig. 10).

		ID
DEFAULT	**MARRIAGE**	
0	**0**	49
	1	10453
	2	12623
	3	239
1	**0**	5
	1	3206
	2	3341
	3	84

Fig. 10. Screenshot showing number of defaulters on the basis of marital status

Here, we can see that the education column contains some information that is redundant. According to the dataset description:

1. married
2. single
3. others

The marriage column in this dataset has an additional parameter of 0. That could happen as a result of a mistake or if the client is unwilling to dispense with the details. To fewer the classifications for marriage and replace the amount of undesirable entries in the column, we can merge the label 0 with 3 (Fig. 11).

```
2    15964
1    13659
3      377
Name: MARRIAGE, dtype: int64
```

Fig. 11. Removing redundancy in MARRIAGE column

Now, determine how many of the defaulters on credit cards are single or married (Fig. 12).

		ID
DEFAULT	**MARRIAGE**	
0	**1**	10453
	2	12623
	3	288
1	**1**	3206
	2	3341
	3	89

Fig. 12. Screenshot showing number of credit card clients having different marital status

Both single and married credit card users make up about the same percentage of defaulters.

Mean and median ages of defaulters and non-defaulters:- (Fig. 13)

	AGE		
	mean	**std**	**median**
DEFAULT			
0	35.417266	9.077355	34.0
1	35.725738	9.693438	34.0

Fig. 13. Screenshot showing computation of mean and median

The median and mean ages for defaulters and non-defaulters are roughly equal.

Let's compute the mean and median "LIMIT BAL" of defaulters and non-defaulters in a similar manner (Fig. 14).

	LIMIT_BAL		
	mean	std	median
DEFAULT			
0	178099.726074	131628.359660	150000.0
1	130109.656420	115378.540571	90000.0

Fig. 14. Screenshot showing computation of mean and median 'LIMIT_BAL' of defaulters and non-defaulters.

Comparing defaulters to non-defaulters, the mean and median credit card limits were lower for defaulters. This implies that defaulters were granted lower credit limits because they had less ability to repay than other borrowers.

Let's compute the mean, median, quartiles, and standard deviation values for the data-columns with continuously occurring numeric values (Fig. 15).

	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
count	30000.000000	30000.000000	30000.000000	30000.000000	3.000000e+04	30000.000000	30000.000000	30000.000000	30000.000000	3.000000e+04	30000.00000	30000.000000	30000.000000	30000.000000
mean	167484.322667	35.485500	51223.330900	49179.075167	4.701315e+04	43262.948967	40311.400967	38871.760400	5663.580500	5.921163e+03	5225.68150	4826.076867	4799.387633	5215.502567
std	129747.661567	9.217904	73635.860576	71173.768783	6.934939e+04	64332.856134	60797.155770	59554.107537	16563.280354	2.304087e+04	17606.96147	15666.159744	15278.305679	17777.465775
min	10000.000000	21.000000	-165580.000000	-69777.000000	-1.572640e+05	-170000.000000	-81334.000000	-339603.000000	0.000000	0.000000e+00	0.00000	0.000000	0.000000	0.000000
25%	50000.000000	28.000000	3558.750000	2984.750000	2.666250e+03	2326.750000	1763.000000	1256.000000	1000.000000	8.330000e+02	390.00000	296.000000	252.500000	117.750000
50%	140000.000000	34.000000	22381.500000	21200.000000	2.008850e+04	19052.000000	18104.500000	17071.000000	2100.000000	2.009000e+03	1800.00000	1500.000000	1500.000000	1500.000000
75%	240000.000000	41.000000	67091.000000	64006.250000	6.016475e+04	54506.000000	50190.500000	49198.250000	5006.000000	5.000000e+03	4505.00000	4013.250000	4031.500000	4000.000000
max	1000000.000000	79.000000	964511.000000	983931.000000	1.664089e+06	891586.000000	927171.000000	961664.000000	873552.000000	1.684259e+06	896040.00000	621000.000000	426529.000000	528666.000000

Fig. 15. Screenshot showing statistical description

The afore mentioned metrics of central tendency values can be divided into other classified columns. We must create pivot tables for this.

2.3 Pivot Table

It is a more potent variation of the grouping function seen in Pandas. Use the pivot table() method, which returns a two-dimensional table, from the Pandas module to create a pivot table (also a Pandas DataFrame object). The pivot table levels are stored in the MultiIndex objects (hierarchical index) of the indices and columns of the resulting DataFrame.

Its syntax is:

pandas.pivot_table(data, values = None, index = None, columns = None, aggfunc ='mean', sort = true).

where.

- Data is a Pandas DataFrame object containing the raw data.
- Values is an optional parameter that determines which column values are aggregated.
- The index determines which column the value should be split into. That is, by which column values you want to group other column values.
- Columns allow you to further group values by a specific column. Aggfunc is the aggregation function and mean is the default aggregator. Sort specifies if the result should be sorted. Its default value is True.
- Let's make a pivot table that groups the columns with continuous numeric values by DEFAULT and AGE columns and shows the median value of those columns (Fig. 16).

		AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	LIMIT_BAL	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
DEFAULT	SEX														
0	1	35	25742.0	23827.0	21103.0	19483.0	18440.0	17531.0	150000.0	2458.0	2200.0	1991.0	1600.0	1600.0	1501.0
	2	33	21368.0	20344.0	19570.0	18475.0	17318.0	15872.0	160000.0	2460.0	2300.0	2000.0	1838.0	1868.0	1853.0
1	1	36	20139.0	20114.0	19753.0	19032.0	18161.0	17959.0	80000.0	1600.0	1500.0	1158.0	1000.0	1000.0	960.0
	2	33	20226.0	20503.0	20023.0	19315.0	18843.0	18244.0	100000.0	1670.0	1600.0	1300.0	1000.0	1000.0	1001.0

Fig. 16. Screenshot showing median value of the columns consisting of continuous arithmetic values and sort the data into columns for 'DEFAULT' and 'SEX'.

We may deduce the distribution of defaulters and non-defaulters with regard to the customer's gender from the pivot table. We note that: (Fig. 17)

- When compared to male clients, female clients among defaulters have a lower median age.
- When compared to female clients, the LIMIT_BAL for defaulters is lower for male clients.

		AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	LIMIT_BAL	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
DEFAULT	MARRIAGE														
0	1	39	22573.0	21022.0	19905.0	18532.0	17390.0	15442.0	180000.0	2500.0	2386.0	2000.0	1877.0	1876.0	1800.0
	2	30	23469.0	22382.0	20401.0	19353.0	18336.0	17345.0	140000.0	2406.0	2200.0	2000.0	1680.0	1707.0	1681.0
	3	42	21953.5	20432.0	18411.0	17915.5	16784.0	14691.0	70000.0	1943.0	1678.5	1477.5	1062.5	1000.0	1000.0
1	1	40	21633.5	21190.5	20445.0	19588.0	19001.5	18320.5	100000.0	1616.5	1600.0	1238.0	1000.0	1000.0	1000.0
	2	29	19438.0	19844.0	19492.0	18859.0	18187.0	17976.0	80000.0	1640.0	1500.0	1214.0	1000.0	1000.0	1000.0
	3	44	18126.0	18459.0	18942.0	18300.0	17760.0	17327.0	50000.0	1700.0	1679.0	1000.0	1000.0	1000.0	800.0

Fig. 17. Screenshot showing the median value of the columns consisting of continuous arithmetic values and group them by 'DEFAULT' and 'MARRIAGE' columns.

The pivot table allows us to deduce the variation of defaulters and non-defaulters based on the customer's marital status.

We notice that

- If the bank's LIMIT_BAL is lower, married people around the age of 40 are more likely to default on payments.
- If the bank's LIMIT BAL is lower, singles under the age of 29 are more likely to default on payments.
- If the bank's LIMIT BAL is lower, customers who haven't indicated their marital status and are about 44 years old are more likely to miss a payment.

The pivot table shows the distribution of defaulters and non-defaulters based on the customer's EDUCATION status. Regardless of educational background, clients with lower approved LIMIT_BAL are more likely to default. Only the x tick labels for the bottom subplot are generated when two subplots share an abscissa along a column. Similar to this, when subplots share an ordinate along a row, only the y tick labels associated with the first column subplot are generated. Use tick params to enable the tick labels of other subplots later (Fig. 18).

		AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	LIMIT_BAL	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
DEFAULT	EDUCATION														
0	1	32	14712.0	14244.0	14394.0	13222.0	11599.0	10075.0	200000.0	2991.0	2800.0	2101.0	2000.0	2000.0	2000.0
	2	34	28084.5	27057.0	24908.5	21228.5	19683.0	19204.0	130000.0	2252.0	2100.0	2000.0	1700.0	1700.0	1600.0
	3	40	27220.5	25402.5	22889.0	19559.5	18610.0	17912.0	100000.0	2069.5	2000.0	1726.0	1396.5	1446.0	1304.0
	4	32	10734.5	8872.5	7183.0	10412.5	6342.0	5334.0	200000.0	3000.0	3000.0	2950.0	1908.5	1399.5	1786.0
	5	35	42407.0	36011.0	31500.0	27981.0	21617.0	12106.0	160000.0	3000.0	3078.0	2524.0	1842.0	1500.0	1610.0
1	1	33	10797.5	11730.5	11391.0	10819.5	10466.5	9667.0	150000.0	1500.0	1468.5	1025.5	929.5	1000.0	902.0
	2	33	25344.0	24653.0	23086.0	20423.0	19688.5	19414.5	80000.0	1700.0	1600.0	1300.0	1000.0	1009.5	1000.0
	3	41	21508.0	21410.0	20453.0	19584.0	18362.0	17636.0	50000.0	1600.0	1585.0	1200.0	1000.0	1000.0	1000.0
	4	29	5374.0	12360.0	20721.0	1000.0	326.0	390.0	120000.0	5000.0	2000.0	1000.0	326.0	390.0	780.0
	5	41	73699.5	67973.5	60048.5	45139.0	27405.5	25796.0	110000.0	3500.0	2690.5	1418.0	1868.5	1295.5	1167.5

Fig. 18. The median value of the columns containing ongoing numeric values and group them by 'DEFAULT' and 'EDUCATION' columns

2.4 Subplots

Let's make count plots that show the number of defaulters by gender, marital status, and education all in one figure. To plot all of the figures together, use the matplotlib.pyplot module's subplots() function. It produces a figure as well as a series of subplots. [8] In a single call, this utility wrapper simplifies the creation of common subplot layouts, including the enclosing figure object. Its syntax is:

subplots(nrows, ncols, figsize, sharex = false, sharey = False, squeeze = True)
where

- **nrows, ncols:** Defines the subplot grid's number of rows and columns.
- **sharex, sharey**: Controls the sharing of attributes between the x and y axes). **False** is its default value. It can accept inputs as well as the values **"none," "all," "row," and "col."**

- If set to **True or "all,"** the x or y axis will be shared by all subplots.
- If set to False or "none," the x and y axes of each subplot will be independent.
- If the value is set to "row," each subplot row will share an x or y axis.
- If the value is set to "col," each subplot column will share an x or y axis.

Only the bottom subplot's x tick labels are generated when subplots share an x-axis along a column. When subplots share a y-axis along a row, only the y tick labels for the first column subplot are created. To enable the ticklabels of other subplots later, use tick params.

When a shared axis across subplots has units, calling set units will update each axis with the new units.

- Squeezebool has a True default value.
- If True, additional dimensions are extracted from the returned axes array:
  - The object returned is a 1D numpy object array of Axes objects with N or 1M subplots.
  - Subplots with N > 1 and M > 1 are returned as a 2D array for N M
  - where N and M represent the number of rows and columns in the Axes grid, respectively.
- If False, no squeezing is done; the provided Axes object is always a 2D array of Axes instances, even if it ends up being 1x1.

To create such a plot:

- Use the **subplots()** function on the **matplotlib.pyplot** object to unpack the **figure** and **axis** objects into two distinct variables, fig and axis, and then call the function. Pass: inside the **subplots()** function.
- Create a figure with 1 row and 3 columns by entering the parameters **nrows = 1** and **ncols = 3**.
- using the **figsize = (14, 5)** argument, a figure with dimensions of (14, 5) is produced.
- To further magnify the image based on its pixel density, use the **dpi = 96** parameter.
- If **sharey** is set to **True**, all future subplots will share the figure's y-axis.
- To build count plots for SEX and MARRIAGE, call the **countplot()** function three times, one at a time, and pass the necessary input parameters. Together with the columns **EDUCATION**, ax = axis[0], and ax = axis[1] and ax = axis[2] as additional inputs.
- Call the **suptitle()** on the figure object to provide a title to the subplot.
- Call the **show()** function on the **matplotlib.pyplot** object (Fig. 19).

Similar to this, box plots can be made that show the age and credit limit of defaulters in a single figure, categorised by "SEX," "MARRIAGE," and "EDUCATION."

Note:

- Use axis[1, 2] to obtain an axis in the second row and third column of a grid of, let's say, three rows and five columns.
- Make that the sharey parameter in the subplots() function is set to False whenever there are multiple input values for the y parameter in any type of plot (Fig. 20).

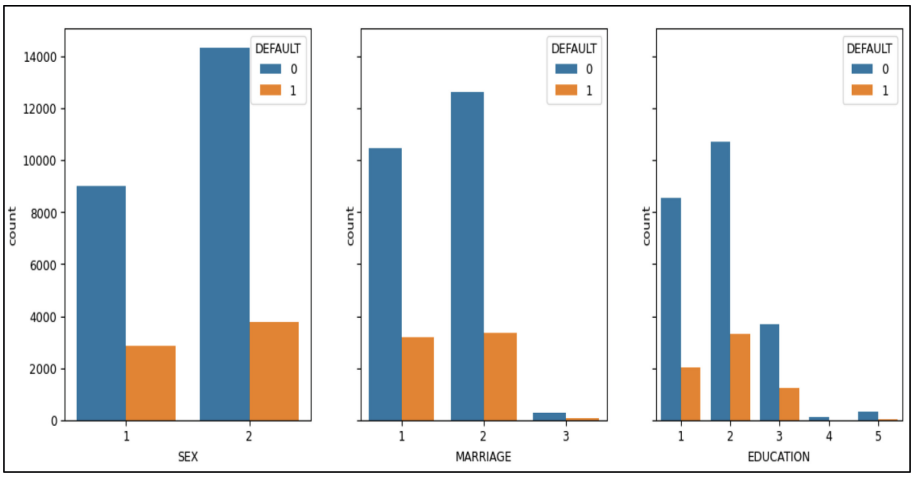

Fig. 19. Count Plots showing defaulters grouped by sex, marriage and education.

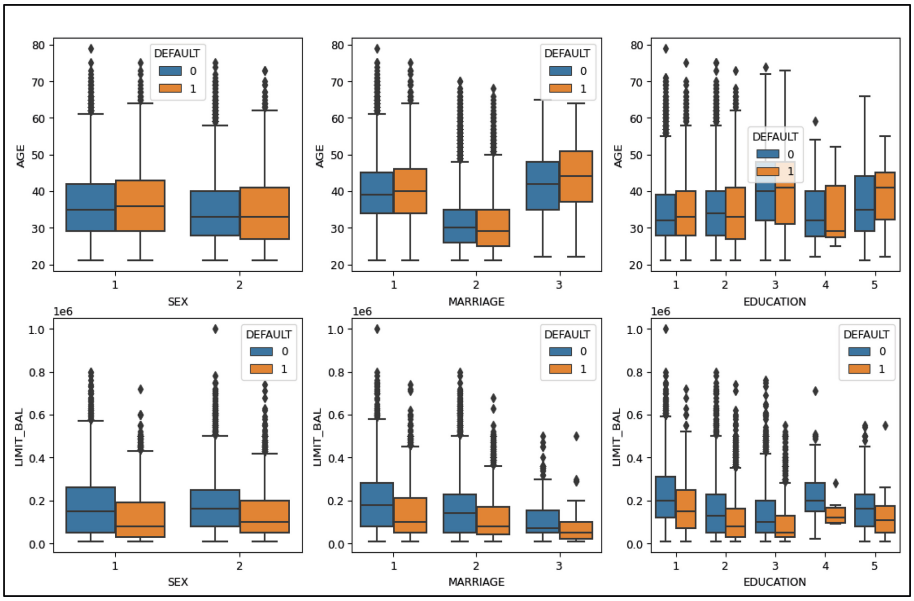

Fig. 20. Box Plots showing age and credit limit of defaulters grouped by sex, marriage and education.

3 Experiment Results

The distribution of defaulters can be deducted from various parameters such as Education Status, age group, employment status etc.

22% of Taiwanese bank's credit card customers miss their due dates for payment.

In comparison to the credit card limit of non-defaulters, the non-payers had lower median and mean credit card limits. This was done so since they had poorer repayment capacity.

When compared to male clients, female clients with defaults had a lower median age.

The male client default limit is significantly smaller than the female client default limit.

If the limit allowed by the bank is smaller, married persons in their 40s are more predisposed to miss payments.

If the limit allowed by the bank is lower, singles with an average age of roughly 29 are more likely to miss payments.

Customers who have not disclosed their marital status and are 44 or older are more prone to miss payments if the bank's maximum is lower (Fig. 21).

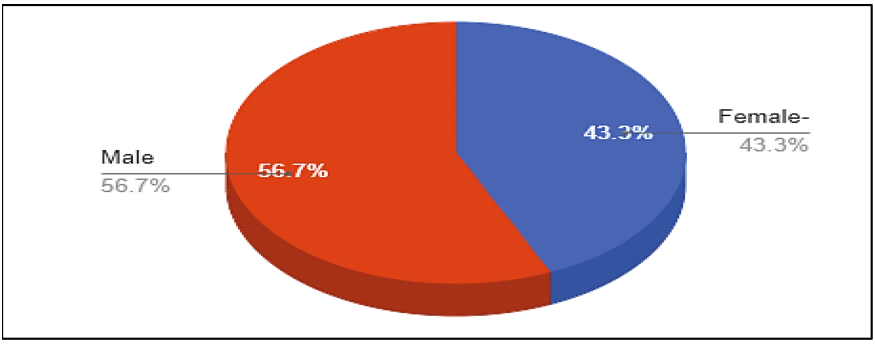

Fig. 21. XXX

According to our data, there is a non-linear relationship between several parameters and credit card default. With implications for strategic default, this model has made an effort to illuminate how demographic and individual factors can potentially affect repayments.

4 Conclusion

Credit card and cash card debt hit $268 billion USD in February 2006. More than 500,000 borrowers were unable to make their loan payments. They were "slaves" to their credit cards. The primary causes of the increase in the suicide rate from 2005 to 2015 are unemployment and credit card debt. This not only affected the people, but also had an adverse effect on the bank and the country's economy.

The fact that the majority of credit clients are university graduates may be the cause of the higher percentage of defaulters who hold a degree.

The majority of credit consumers are graduates from universities, which may account for the higher percentage of defaulters who hold degrees. The median and mean ages for defaulters and non-defaulters are roughly equal.

Comparing defaulters to non-defaulters, defaulters had lower mean and median credit card limits.

Credit card debt default is quite arbitrary and unpredictable because it entails moral dilemmas on a personal level.

A strong credit index system makes it easier to evaluate personal credit and create risk prediction models with improved classification capabilities.

Only the bottom subplot's x tick labels are generated when subplots share an x-axis along a column. When subplots share a y-axis along a row, only the y tick labels for the first column subplot are created. To enable the ticklabels of other subplots later, use tick params.

When a shared axis across subplots has units, calling set units will update each axis with the new units.

- Squeezebool has a True default value.
- If True, additional dimensions are extracted from the returned axes array:
    - The object returned is a 1D numpy object array of Axes objects with N or 1M subplots.
    - Subplots with N > 1 and M > 1 are returned as a 2D array for N M
    - where N and M represent the number of rows and columns in the Axes grid, respectively.
- If False, no squeezing is done; the provided Axes object is always a 2D array of Axes instances, even if it ends up being 1x1.

5 Future Scope

Although we believe that the goal of the research project has been accomplished to a great extent, there are plenty of improvements that could be done to achieve better results. In the following sections we present, some aspects of the research carried out in this project and some of the open issues that deserve further work and implementation.

Using a bigger test data set: The trustworthiness of any model's accuracy results is greatly enhanced when tested on huge datasets, and while our test data is considered large enough to be reliable; using an even bigger dataset would only add to its credibility and point out possible shortcomings if any.

References

1. Sayjadah, Y., Hashem, I.A.T., Alotaibi, F., Kasmiran, K.A.: Credit card default prediction using machine learning techniques. In: 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA) (2018)
2. Credit Card Defaulter, Credit Card Defaulter in India - Paisabazaar.com - 20 October 2022. Compare & Apply Loans & Credit Cards in India- Paisabazaar.com (2018). www.paisabazaar.com/credit-card/credit-card-defaulter
3. UCI Machine Learning Repository: Default of Credit Card Clients Data Set. UCI Machine Learning Repository: Default of Credit Card Clients Data Set. archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients. Accessed 20 Oct 2022

4. Yeh, I.-C., Lien, C.-H.: Redirecting. Redirecting (2009). https://doi.org/10.1016/j.eswa.2007.12.020
5. Pandas.Pivot_Table — Pandas 1.5.1 Documentation. Pandas.Pivot_Table — Pandas 1.5.1 Documentation (2000). pandas.pydata.org/pandas-docs/stable/reference/api/pandas.pivot_table.html
6. Yeh, H.-C., Yang, M.-L., Lee, L.-C.: An empirical study of credit scoring model for credit card. *Second* International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), p. 216 (2007). https://doi.org/10.1109/ICICIC.2007.138
7. Dua, D., Graff, C.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2019)
8. "matplotlib.pyplot.subplots — Matplotlib 3.6.0 Documentation." matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.subplots.html. Accessed 20 Oct 2022
9. Machine Learning Random Forest Algorithm - Javatpoint. www.javatpoint.com, www.javatpoint.com/machine-learning-random-forest-algorithm. Accessed 21 Oct 2022