GW Columbian College of Arts & Sciences

Data Science Capstone Presentation

Author: Deen Huang

Introduction

- This project is created to classify and predict the credible credit card clients based on the classic and advanced machine learning methods

- The idea is to use the default of credit card clients data set from *UCI Machine Learning* applied in different machine learning models and compared the performance of two models

Dataset

Default of Credit Card Clients Data Set:

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Dataset

- ID: ID of each client
- LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default.payment.next.month: Default payment (1=yes, 0=no)

Problem Statements

- How does the probability of default payment vary by categories of different demographic variables?
- Which variables are the strongest predictors of default payment?
- How do we predict the credit card clients may lead to default of credit card?

Packages

- pandas
- numpy
- seaborn
- matplotlib
- sklearn
- tensorflow

Exploratory Data Analysis

	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_A
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0.0	0.0	0.0	0.0	689.0	
1	2	120000.0	2	2	2	26	-1	2	0	0	...	3272.0	3455.0	3261.0	0.0	1000.0	1(
2	3	90000.0	2	2	2	34	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	1500.0	1(
3	4	50000.0	2	2	1	37	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	2019.0	1:
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	20940.0	19146.0	19131.0	2000.0	36681.0	10(

- Load the data and observe the 24 variables

```
data.shape
```

```
(30000, 24)
```

Exploratory Data Analysis

```
data.rename(columns={'default.payment.next.month':'default'}, inplace=True)
data.rename(columns={'PAY_0':'PAY_1'}, inplace=True)
data = data.drop('ID', axis=1)
data.head()
```

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PA
0	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	
1	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	
2	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	
3	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	
4	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	

- Remove “ID” column and rename two of other confusing columns

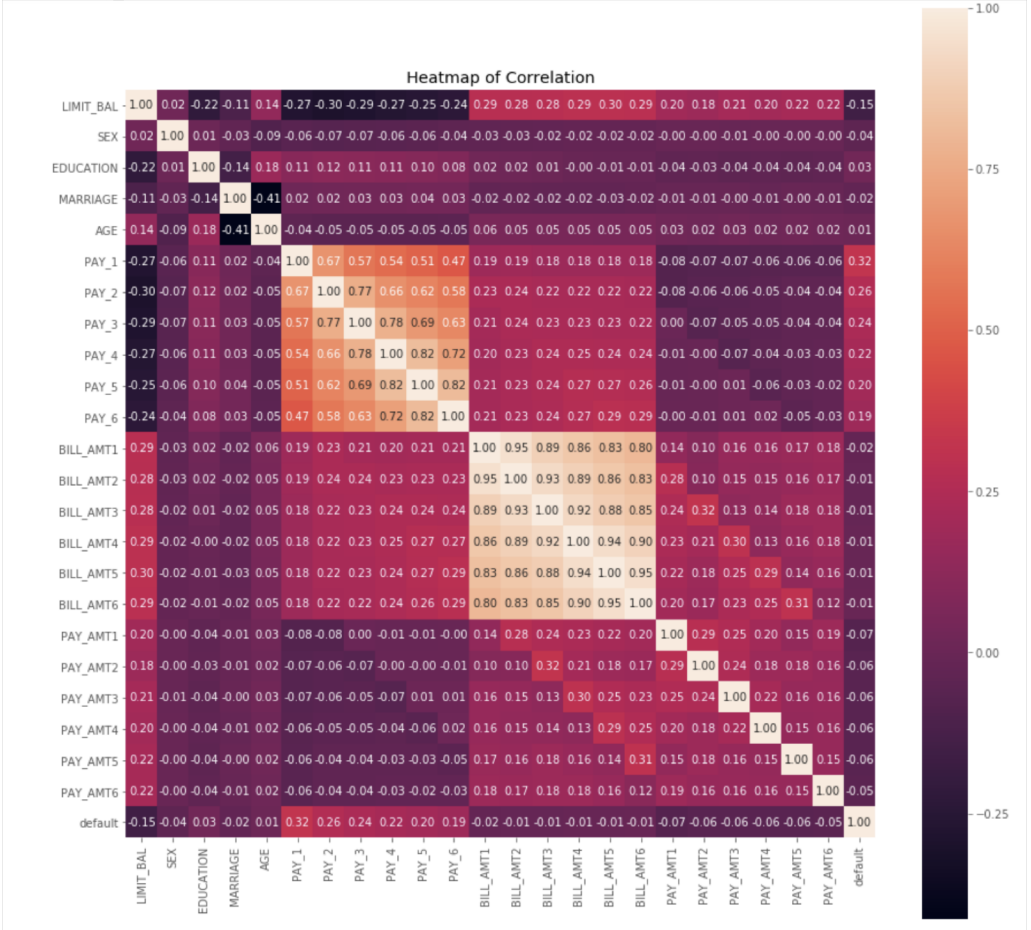
Exploratory Data Analysis

```
LIMIT_BAL      0
SEX            0
EDUCATION      0
MARRIAGE       0
AGE            0
PAY_1          0
PAY_2          0
PAY_3          0
PAY_4          0
PAY_5          0
PAY_6          0
BILL_AMT1      0
BILL_AMT2      0
BILL_AMT3      0
BILL_AMT4      0
BILL_AMT5      0
BILL_AMT6      0
PAY_AMT1       0
PAY_AMT2       0
PAY_AMT3       0
PAY_AMT4       0
PAY_AMT5       0
PAY_AMT6       0
default        0
dtype: int64
```

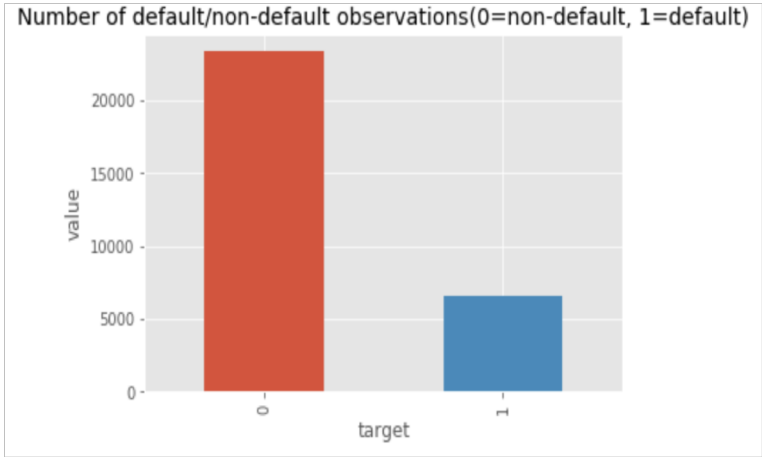
- Check the missing values: NO missing values!

Exploratory Data Analysis

- Plot heatmap for correlation among each variable to find out the strongly and weakly correlation between features and default



Exploratory Data Analysis

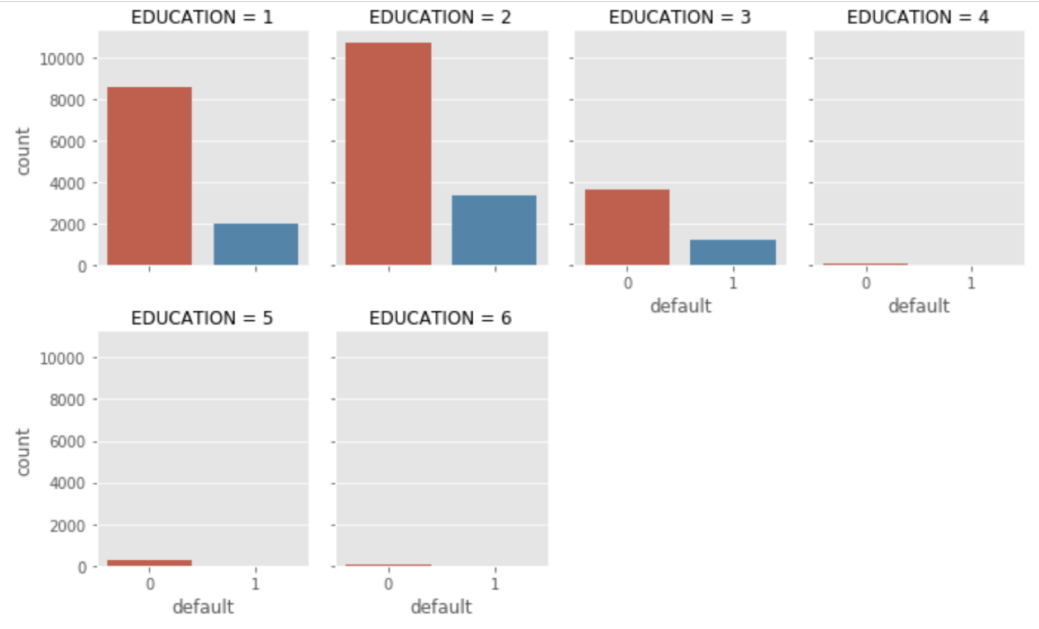



```
0    23364
1     6636
Name: default, dtype: int64

default Ratio : 0.2212
```

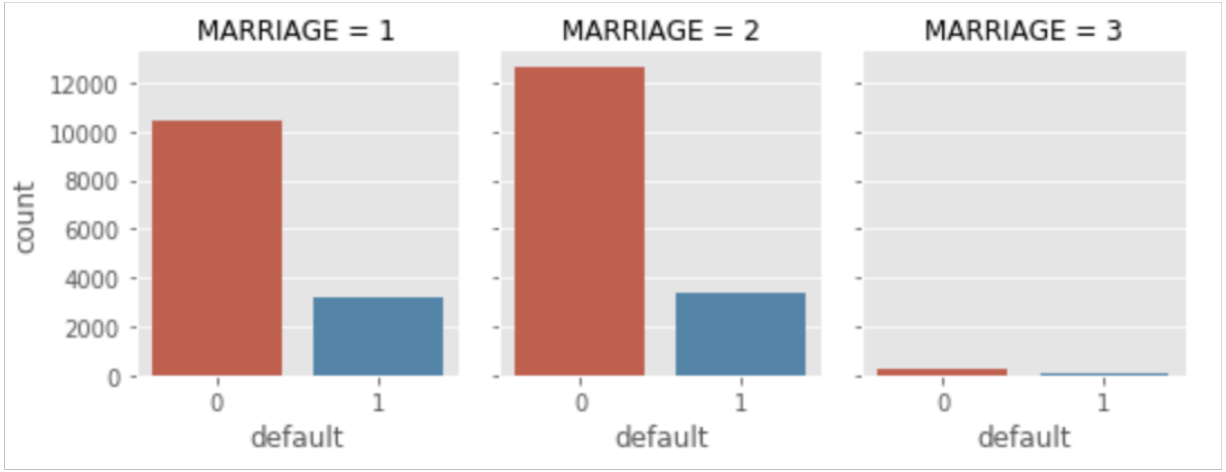
- Check balance: Imbalanced!
- Default ratio is 22.12%
- Split dataset
- Resampling strategies

Exploratory Data Analysis



- Compare the default to education, however, “education” does not suggest a strong impact to default payment
- People who received education from graduate school and university seems have more non-default payments than high school , but not a strong indicator

Exploratory Data Analysis



- Although people who are single have more non-default payment, it seems not a strong indicator
- Hypothesis: people get marriage may affect the default. However, the plot shows it does not really affect

Methodology

- Classic Machine Learning: Gaussian Naïve Bayes

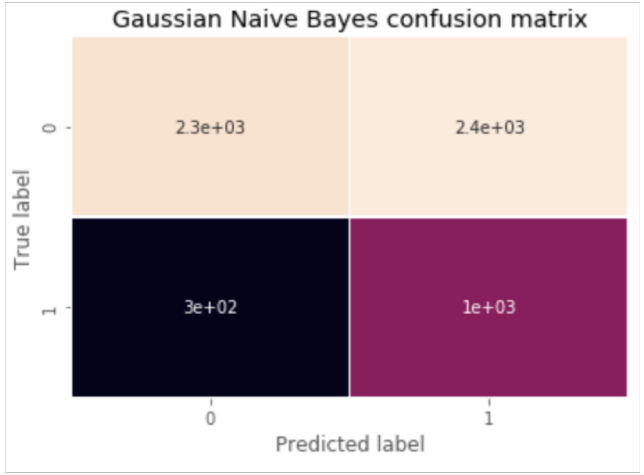
- Advanced Machine Learning: Multilayers Neural Network via Tensorflow

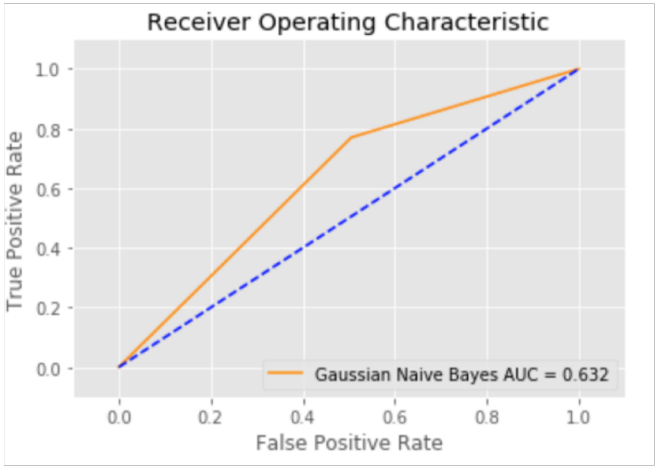
Machine Learning Model

- Gaussian Naïve Bayes from package Sklearn
  - Normalize the data
  - Scaling the features to a standard normal distribution with mean of 0 and standard deviation of 1
  - Fit the model for a good prediction
  - Process with Gaussian Naïve Bayes

Gaussian Naïve Bayes

	Model	Accuracy	Precision	Recall	F1 Score	ROC
0	GNB	0.554	0.295825	0.770239	0.427471	0.632302





- Created a table
- Accuracy is low as 55.4%
- Confusion matrix shows accuracy of predicting non-default is low

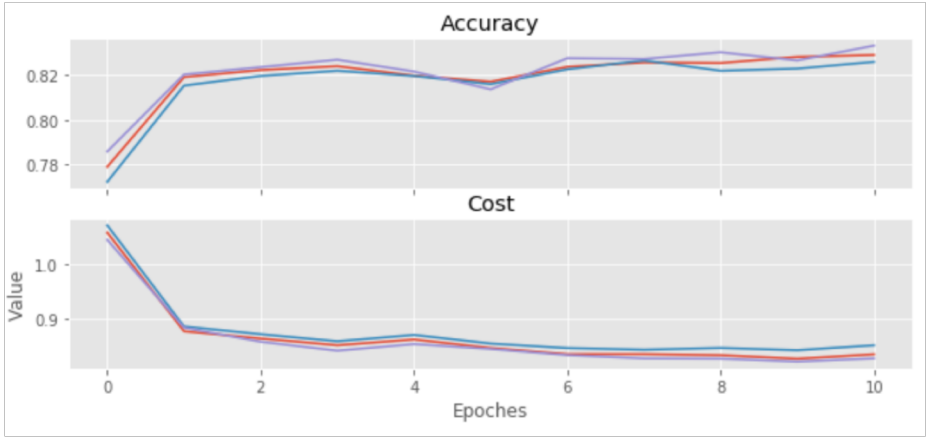
Deep Learning Model

- Neural Network with 5 hidden layers via Tensorflow
  - Scaling the features to a standard normal distribution with mean of 0 and standard deviation of 1 as well
  - Split the test data to test and validation
  - Declare the basic structure of the data
  - Define cost function and optimizer
  - Start training circle and display logs every 10 epochs

```
y_clipped = tf.clip_by_value(y, 1e-10, 0.9999999)
cost = -tf.reduce_mean(tf.reduce_sum(y_ * tf.log(y_clipped)
                        + (1 - y_) * tf.log(1 - y_clipped), axis=1))
```

```
optimizer = tf.train.AdamOptimizer(learning_rate).minimize(cost)
```

Neural Network




```
Epoch: 0 Acc = 0.779 Cost = 1.058 Valid_Acc = 0.772 Valid_Cost =  1.071 test_Acc = 0.786 test_Cost =  1.045
Epoch: 10 Acc = 0.819 Cost = 0.877 Valid_Acc = 0.815 Valid_Cost =  0.886 test_Acc = 0.820 test_Cost =  0.884
Epoch: 20 Acc = 0.822 Cost = 0.864 Valid_Acc = 0.820 Valid_Cost =  0.872 test_Acc = 0.824 test_Cost =  0.858
Epoch: 30 Acc = 0.824 Cost = 0.852 Valid_Acc = 0.822 Valid_Cost =  0.859 test_Acc = 0.827 test_Cost =  0.842
Epoch: 40 Acc = 0.820 Cost = 0.862 Valid_Acc = 0.820 Valid_Cost =  0.870 test_Acc = 0.822 test_Cost =  0.854
Epoch: 50 Acc = 0.817 Cost = 0.847 Valid_Acc = 0.816 Valid_Cost =  0.855 test_Acc = 0.814 test_Cost =  0.845
Epoch: 60 Acc = 0.824 Cost = 0.835 Valid_Acc = 0.823 Valid_Cost =  0.846 test_Acc = 0.828 test_Cost =  0.833
Epoch: 70 Acc = 0.826 Cost = 0.835 Valid_Acc = 0.827 Valid_Cost =  0.843 test_Acc = 0.827 test_Cost =  0.827
Epoch: 80 Acc = 0.826 Cost = 0.833 Valid_Acc = 0.822 Valid_Cost =  0.847 test_Acc = 0.830 test_Cost =  0.827
Epoch: 90 Acc = 0.828 Cost = 0.827 Valid_Acc = 0.823 Valid_Cost =  0.842 test_Acc = 0.827 test_Cost =  0.822
Epoch: 100 Acc = 0.829 Cost = 0.835 Valid_Acc = 0.826 Valid_Cost =  0.851 test_Acc = 0.833 test_Cost =  0.827

Optimization Finished!
```

- Accuracy is high, the model performs good
- With epochs increase, accuracy becomes higher, and cost goes down

Conclusion

- Deep learning model – Neural Networks provides a higher accuracy and better performance
- Cost becomes lower and accuracy becomes higher when the epochs increased
- There are not any extreme demographic variables that indicated the default payment
- Further process to make high accuracy in machine learning model : Adaboost, Gradient Boosting

Thank you!