Prediction for Credit Card Default

Introduction:

Financial institutions like banks usually use a person's credit to decide how much money can be loaned, how much interest should a person pay, and so on. Thus, credit is important to financial institutions, and one factor affecting credit is the ability paying back credit card.

Our task is to decide whether a person will pay back in October 2005 given some data from April to September 2005. In the dataset used for prediction, the prediction target is represented as default.payment.next.month with two values: 1—will not pay back next month, 0—will pay back next month.

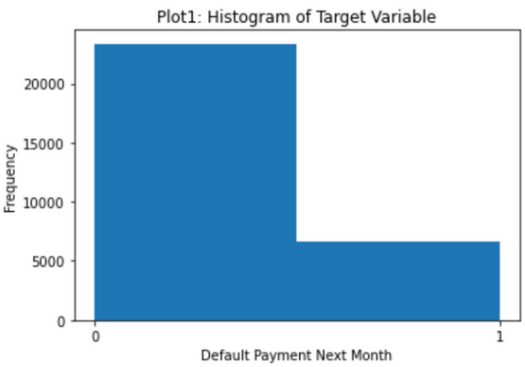
Many information are given for prediction, including gender, education level, age, marriage status, amount of given credit, repayment status, amount of bill statement, amount of previous payment for every month from April to September 2005. These features are recorded in dataset as SEX, EDUCATION, AGE, MARRIAGE, LIMIT_BAL, PAY_X, PAY_X, PAY_AMTX respectively (X=1-6 representing each month).

The work will be developed as the following: exploratory data analysis(EDA), feature engineering, data preprocessing, predictive modelling, feature importance analysis and final conclusion.

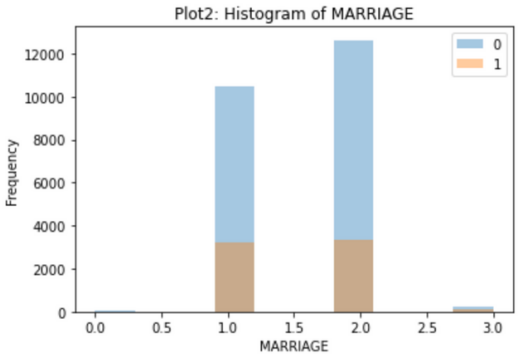
EDA:

First, in order to gain insightful information, EDA is performed:

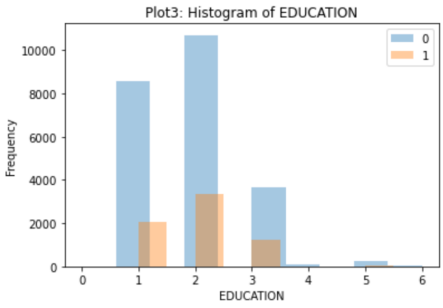
▶ Expand to see the code used to generate the plot



From plot 1, we can see the dataset is imbalanced. This reflects the reality as most people will pay back to maintain a good credit, having default.payment.next.month = 0, while some people will not pay back, having default.payment.next.month = 1.
Because of the data imbalance, the metrics used for model performance assessment will be `f1` score. This is because a decent f1 score indicates a balanced prediction: identifying as many customers who will not pay on time as we can so that banks can avoid giving high credit to people who cannot payback on time, without misidentifying customers who can pay on time as cannot payback as their credit may be negatively affected.




```
l]: 2    15964
    1    13659
    3      323
    0       54
    Name: MARRIAGE, dtype: int64
```




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

▶ Expand to see the code used to generate plot 2 and 3

From plot 2, 3 and the unique column values count, we can see the distribution of feature marriage and education. There are some miscellaneous values for both features because in the description provided for this dataset, there are only 3 possible values (1=married, 2=single, 3=others) for MARRIAGE and 6 possible values (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown) for EDUCATION. However, from what we seen, there are 0s in both features with unknown meanings. Therefore, for MARRIAGE, 0 and 3 will be combined as others before applying one hot encoding. For EDUCATION, 0, 4, 5, 6 will be combined as others before ordinal encoding with the order below: 1 = graduate school; 2 = university; 3 = high school; 0, 4, 5, 6 = others.

Feature Engineering:

Columns of average amount of bill statement **BILL_AMT_mean** using BILL_AMTX(X=1-6) and average amount of previous payment **PAY_AMT_mean** using PAY_AMTX(X=1-6) from April to September 2005 are generated.

These columns can be helpful in the prediction model because:

- **PAY_AMT_mean** shows their average pay back ability. Higher paying back amount may indicate less delayed payings.
- **BILL_AMT_mean** shows their normal spending level. Higher spending may indicate difficulty in paying back. Both factors can affect the ability to pay back.

▶ Expand to see the code used to generate columns BILL_AMT_mean and PAY_AMT_mean

Data Preprocessing:

OHE is used for categorical features ("SEX", "MARRIAGE") and ordinal encoding is used for EDUCATION (ranked as stated in exploratory data analysis).

▶ Expand to see the code for data preprocessing

Model Training:

The model used here is a tree-based ensemble model, lightGBM. It is chosen because a single decision tree or other single models is likely to overfit, while using a collection of diverse decision trees reduces overfitting by averaging the results. Also, compared to other tree-based models like random forest, LightGBM usually has better scores and a faster training time.

Cross validation is also carried out to see how the model performs.

▶ Expand to see the code

```
{'fit_time': array([0.61774492, 0.40324402, 0.20808196, 0.2579689 , 0.23592234]),
 'score_time': array([0.02771497, 0.02293682, 0.02678275, 0.02753091, 0.02140474]),
 'test_f1': array([0.46621622, 0.48233696, 0.48438584, 0.50408719, 0.46453408]),
 'train_f1': array([0.58486432, 0.58918646, 0.57549086, 0.58286493, 0.58889838]),
 'test_recall': array([0.36819637, 0.37886873, 0.37206823, 0.39445629, 0.35607676]),
 'train_recall': array([0.46254332, 0.46627566, 0.45333333, 0.45626667, 0.45973333]),
 'test_precision': array([0.63535912, 0.6635514 , 0.69383698, 0.69811321, 0.668     ]),
 'train_precision': array([0.79514207, 0.80009149, 0.78776645, 0.80669496, 0.81900238])}
```

From the cross validation result, the f1 score for validation set is between 0.470 to 0.499 and for training set is approximately 0.58. There is a relatively small difference between training and validation f1 scores, and f1 score is relatively decent.

Applying on test set:

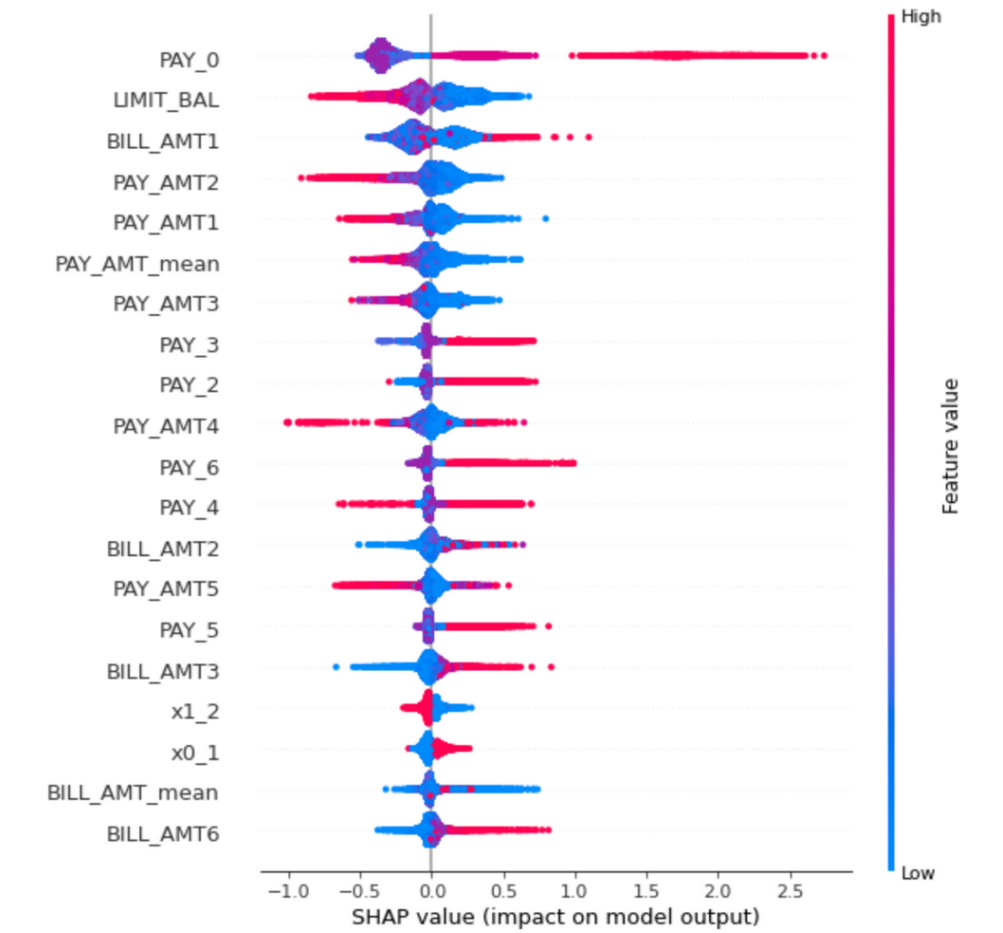
▶ Expand to see the code

The model is tested on test set, and the performance is measured by f1 as below. The f1 score is similar to the validation set score, meaning there is potentially no overfitting, and it is pretty decent.

```
f1 for test data is: 0.474
recall for test data is: 0.376
precision for test data is: 0.642
```

Feature Importance:

▶ Expand to see the code



Plot 4: SHAP summary plot

From plot 4, the most important features for predicting NOT paying back is shown with direction of how it's going to drive the prediction: **more positive SHAP values means more likely to predict not paying back, and vice versa.** The feature importance in line with the reality:

- PAY_X = *a* means delay for *a* months, and PAY_X=-1 means pay on time. Thus, larger number of PAY_X strongly pushes the prediction towards NOT paying back.
- LIMIT_BAL is the amount of given credit. People with good credit often have high LIMIT_BAL. This means they have the ability to pay back on time and they always do that to maintain the good credit; thus, drives the prediction to paying back.
- BILL_AMTX is the amount of bill statement. Higher bill amount means more difficult to payback, driving prediction to NOT paying back.

Final Conclusion:

The model has a decent f1 score in test set, and the score is similar to the scores generated during cross validation, meaning the model is likely to perform similarly in real world dataset without overfitting. Also, the SHAP analysis of feature importance for the model all inline with real world situations, making it interpretable and more reliable.

PAY_AMT_mean and BILL_AMT_mean is not as useful as thought. The reason of this may be higher paying amount and higher billing amount does not mean paying or not paying back. Some people with high income have the ability to spend more and pay more, while some people with lower income have lower bills and pay back amount. Instead, PAY_X which is the pay back status is a more crucial feature as it shows their pay back habits.

Notably, there may be some potential errors. First, the prediction could be misleading because there are so many real life variables to consider other than the features given in the dataset, like liquid assets, job, etc. We should be careful using this simple model to predict a customer's pay back status and determine their credit.

Also, there could be interacting effects between each feature, which makes the feature importance analysis inaccurate. The presence of one feature may affect another feature's importance. Plus, the miscellaneous values may affect the model. For example, 4-others in EDUCATION may be rank highest, while we rank it as the lowest because for 1-3, the rank is larger when number is larger.

Besides, the precision (roughly 0.8 and 0.65 respectively for train and test set) is much higher than recall(roughly 0.45 and 0.37 respectively for train and test set), which may be the reason why f1 score is decent. Thus, we should not be too confident on the model's ability to find all the people who cannot pay back next month, and should be careful on saying a person's cannot pay back even though predict so.