**Final Report on Credit One Default Analysis**

### Overall Impression

Continued with the Exploratory Data Analysis, I am going to apply the Python Sci-Kit Learn module to further analyze in this task. I first used Jupyter Notebook to import numpy, pandas, matplotlib and scipy modules. Then, I uploaded the dataset ‘new\_credit.csv’ generated from the last step of previous EDA section. Further, I used some basic Python functions to check for the data types, locate/select the independent variables and dependent variable, split the dataset into four data frames (X\_train, X\_test, y\_train, and y\_test), this is so called train\_test\_split method, which helped to prepare for next step on model building. In this report, I will include five different classification algorithms. The Gradient Boosting Model was the best metric which yielded the highest accuracy and model scores.

### Data Cleaning and Pre-processing

#### Step 1 – Pre-process the original csv file

At the beginning at EDA step, I loaded preprocessed dataset of ‘DefaultofCreditCardClients.csv’, in which I updated to the actual month info based on the PDF explanation in the plan of attack. In this task, I uploaded the ‘new\_credit.csv’ into Jupyter Notebook which was generated from the last step of EDA.

#### Step 2 – Remove ID column

After checking for the general data info-credit.info(), I removed the ID column not only because it is redundant (Python provides the serial number automatically), but also it is not a relevant independent variable with regard to the dependent variable. Therefore, removing the ID column is an essential step in data cleaning.

#### Step 3 – Categorize the Age Column

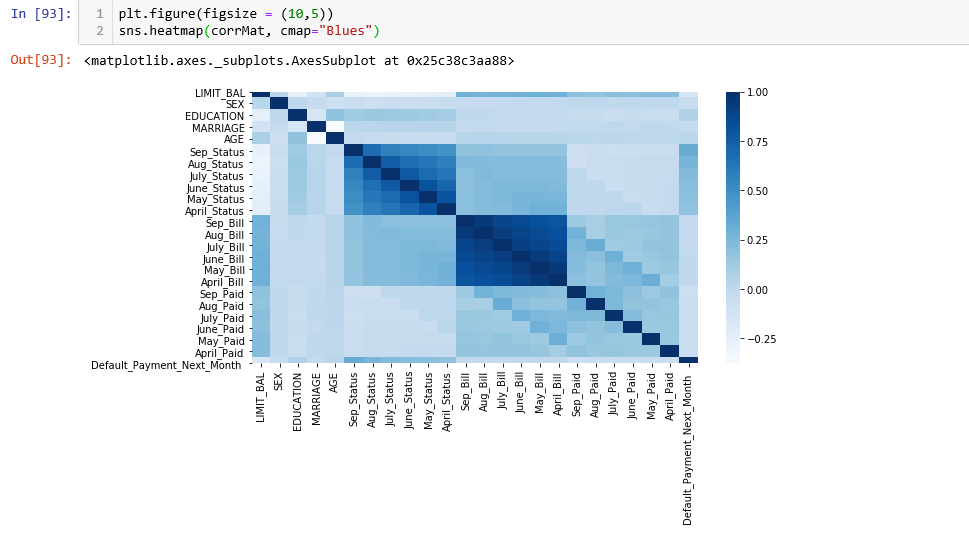
When exploring the entire dataset, I noticed that the original AGE column has a range from 21 to 79 years old. It is better to bin them into several categories at 15 years apart. As the result, I created 4 bins (20-35, 36-50, 51-65, 66-80) to break them up. Here I used pd.cut () function and relabel to [1,2,3,4].

#### Step 4 – Reorganize the Education Column

Initially we have 7 EDUCATION classes (from 0 to 6), in which 1 stands for graduate school, 2 stands for university, 3 stands for high school, and the rest of the values 0,4,5,6 all stand for others. In this case, I believe, it is better to rename 4,5,6 to 0 as well, so it is straightforward and easy to manage. Here I applied .replace() function to consolidate all others to 0. Doing so makes it easy for future data analysis.

### Correlation Estimation

Here I generated the heatmap to show the correlation between each variable. Darker blue colors show positive relationships and light blue colors indicate more negative relationships.



### EDA

In task two, I used seaborn to generate various histogram, line chart, heatmap, distribution chart and bar chart, to get a general idea of how the data look before and after the data preprocessing. Most of the clients are age between 25-40, with the peak around 28 years old. The default rate is 22%. The in-depth EDA can better provide us with two major factors which affect the defaults.

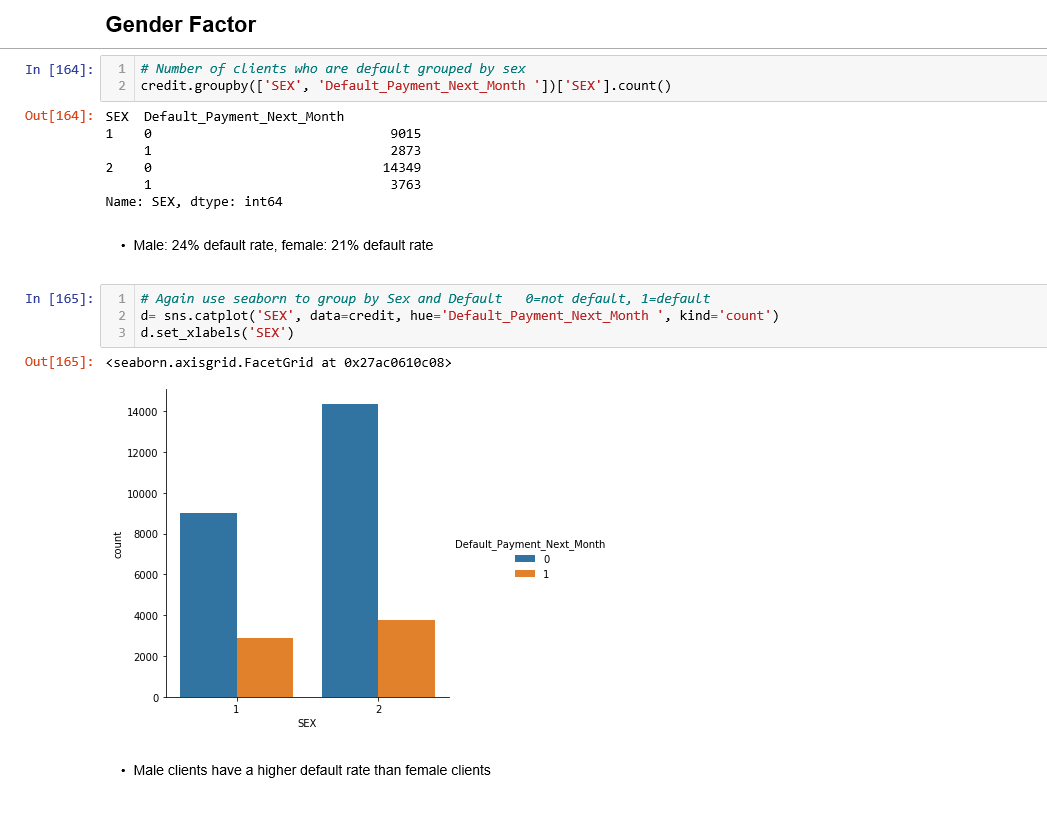
### Factors Affecting Default

#### Demographic Factors

* This is not an evenly distributed dataset. There are about 20% more females than males in the dataset. Therefore, the final result will be a little bias towards females.
* Based on my analysis, males have 24% of default rate Vs. females have 21% default rate. Marital factor is close to even split on default among married and single. There is a significant trend of less likely to default among singles.
* Those who have graduate education level tend to have lower default rate than university and high school graduates.
* High school graduates have a flat line which means no obvious relationship between age and default rate, but its default probability is the greatest among all four education groups.

#### Credit Limit Factors

Clients with lower credit limit are more likely to default.



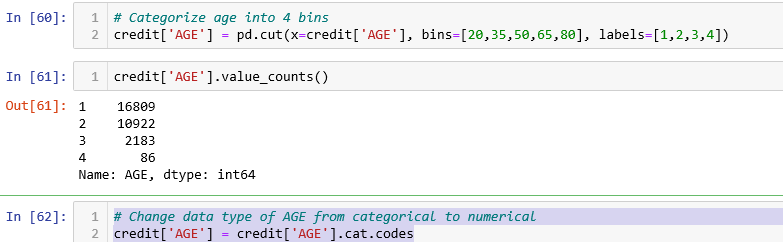
### Dimensionality Reduction

Here I did not perform dimensionality reduction due to highly related features do not bother in the classification problem.

### Label Encoder

When performing data analysis, Python is not able to work with any types of object value, everything has to be a number, there is no nominal data concept exist in Python. Therefore, I have to convert the object value from non-numerical data to numerical data. Luckily, Sklearn offers a very efficient tool for encoding the levels of categorical features into numeric values. LabelEncoder encode labels with a value between 0 and 1. I need to first import the LabelEncoder module from Sci-Kit Learn by syntax: from sklearn.processing import LabelEncoder, then I can apply this function by specifying the column that I want to change to numeric. The syntax is like: labelencoder = LabelEncoder()  
x[:, 0] = labelencoder.fit\_transform(x[:, 0])

What I used in this task was actually .cat.codes() function, after categorizing the age into 4 bins (initially age column contains numerical data) .



### Five Classification Models

Here I imported five different classification classifiers for each model, and built each model by (modelname.fit(X\_train,y\_train), then calculated the model score for each model. I have tried Random Forest (RF), Support Vector Machines (SVM), Gradient Boosting (GB), Logistic Regression (LR), and Decision Trees (DT). Among these models, RF, SVM, and DT all had an incredible high model scores (between 0.98-0.99) which are the strong sign of overfitting! Therefore, I have thrown them away from further analysis. I finally chose Gradient Boosting model as its model score was 0.826 which outperformed the Logistic Regression model (model score of 0.777)



### Model Evaluation

Here I used confusion matrix, accuracy score, kappa score and classification report to evaluate the prediction. The accuracy score for Gradient Boosting is quite closed to its model score mentioned earlier, and above 80% of the accuracy indicates it is a good model.



### Credit One Underlying Problem

Credit One has seen an increase in the number of customers who have defaulted on loans they have secured from various partners. Credit One, as their credit scoring service, could risk losing business if the problem is not solved right away. They have enlisted the help of our Data Science team to design and implement a creative, sound solution.

### How to Solve the Problem?

#### From Business Point of View

People who apply for a loan at Credit One should be associated with a different risk score based on the important factors discussed earlier (age, credit limit, etc.). And different risk score can be associated with different loan cost, i.e., interest rate. This will:

* Compensate high risk applicant with corresponding higher interest rate to balance the return on the loan
* Attract more people with low risk profiles with low interest rate, while naturally discourage people with high risk profiles to obtain a loan with Credit One, which automatically lead to lower default rate over time.

#### From Data Science Point of View

We need to deploy the optimal model: Gradient Boosting thoroughly to see if the client is more likely to default in the future.

* Auto-approval loans under certain amount for those with lower default rate customers using this model to improve the efficiency and productivity.
* Maintain the model regularly to ensure its best performance.