Our client CreditOne realized an increase in customer default rates, which could risk losing business. CreditOne wants to know future possible probability of defaults on loan. In order to know the future possible probability of defaults I did the following steps during the analysis:

1. Overview the dataset and data dictionary
2. Preprocessed the dataset
3. Perform exploratory data analysis
4. Build the different models
5. Making the prediction and evaluating the result

Let’s take a look what I did in each of these steps.

**Overview**

Dataset was consisting of 30000 observation and 25 attributes. The description is provided below:

* X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
* X2: Gender (1 = male; 2 = female)
* X3: Education (1 = graduate school; 2 = university; 3 = high school; 0, 4, 5, 6 = others)
* X4: Marital status (1 = married; 2 = single; 3 = divorce; 0=others)
* X5: Age (year)
* X6 - X11: History of past payment (- 2: No consumption; -1: Paid in full; 0: The use of revolving credit; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for 8 months; 9 = payment delay for nine months and above)
* X12-X17: Amount of bill statement (X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005)
* X18-X23: Amount of previous payment (X18 = amount paid in September 2005; X19 = amount paid in August, 2005;. .;X23 = amount paid in April, 2005)
* Y: client's behavior (Y=0 then not default, Y=1 then default)

**Preprocessing**

In previous projects I used R for analysis, but in this project our client asked us to use Python as a data analytic tool. Preprocessing in Python was very similar to R, in this step I used Pandas library for manipulating data. I used the df.describe(), df.info(), df.isnull().values.any() functions from pandas library in order to see statistical information (mean, min, max, etc.), data types of each variables and detect any missing value in the dataframe.

I noted that the maximum amount of given credit was 1000000.000000, minimum was 10000.000000, most of the customers were women, most of them were around 35 and most majority of customers did not make defaults as the mean was close to 0 (0.22). All the variables were integers and there was not any missing value.

I used the df. drop() function to delete the ID column.

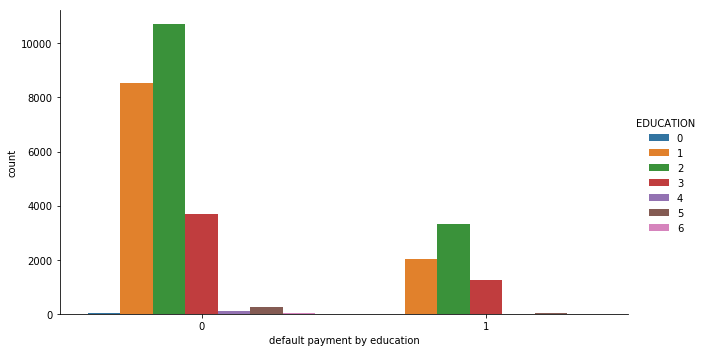
I also binned the ‘age’ variable. Binning help me to discover patterns, carefully develop algorithms and explain the result to an audience easily. pd.cut() function was used for binning.

**Exploratory data analysis**

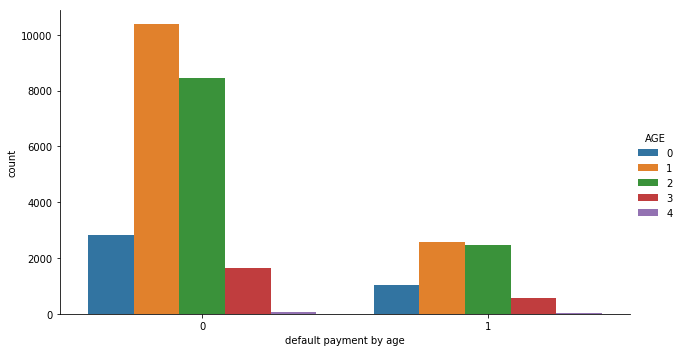
In this step I used different visualization methods in order to understand the relationship between the variables and checked the collinearity. I used the seaborn, matplotlib and pandas libraries.

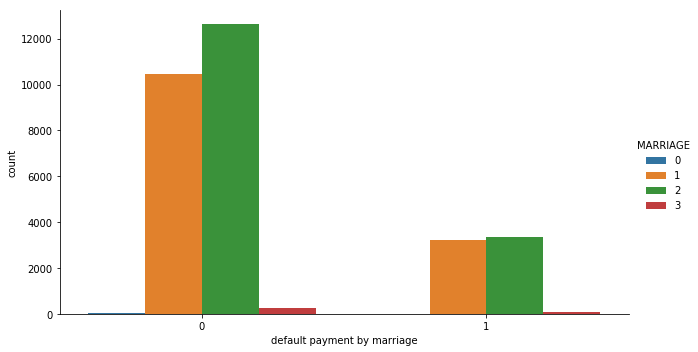
I mostly pay attention to visualize all the demographic variables and their relationship with default payment. These visualizations helped me to understand which customers are good to work with or which customer are riskier to work with. After visualized each demographic data I got the following information:

* most of the customers were women (18112)
* most of them were single (15964) and married (13659)
* most of them have university degree (14030) or they graduated school (10585)
* most of them were between the age group of 25-35 (12935) and 35-55 (10922).

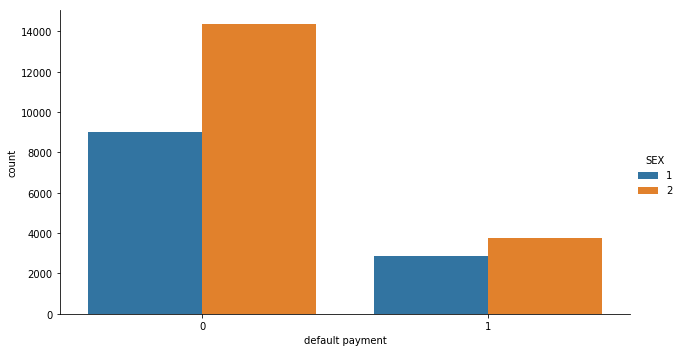


The customers who have the university degree made the most of default on loans.

.

 The customers who have the age between 35-55 made most of defaults on loans.

Single and married customers made most and almost the same number of defaults on loans.



 Men made the most of the defaults on loans.

I used [covariance to measure how two variables move together](https://www.investopedia.com/ask/answers/041315/how-covariance-used-portfolio-theory.asp). It measures whether two variables move in the same direction (a positive covariance) or in opposite directions (a negative covariance. Default payment (dependent variable) has positive covariance between the ‘Education’, ‘Age’ and ‘history of last payment’.

I used correlation coefficient to see these relationships in a standardized form and understand how strong these relationships are. Correlation coefficient between these variables were less than 0.80 which means they don’t really have strong linear relationship.

**Build the models**

On this step I used Sci-kit learn module for model building and evaluation. This package is similar to caret package in R. Before building the models, I change the categorical data into dummy variable. I used pd.get\_dummies() to change all the categorical variables into numeric except the dependent variable. pd.get\_dummies create a new dataframe which consists of zeros and ones.

Then I moved ‘default payment next month’ variable to the first position in the dataframe.

In Sci-kit learn, training and testing data sets have 2 areas which is different from caret.

X\_train and X\_test is the independent variables (features), y\_train and y\_test (ground trust) is the dependent variables.

I defined the independent and dependent variables in my dataframe and divided dataframe into four part as shown above.

To achieve higher accuracy, I tested out three different classifies (Random Forest Classifier, Support Vector Classifier and KNeighbors Classifier) and tried different parameters within each classifier as well. n\_estimotors and max\_features was tuned for Random Forest Classifier, n-neighbors was tuned for KNeighbors Classifier, kernel and c was tuned for Support Vector Classifier.

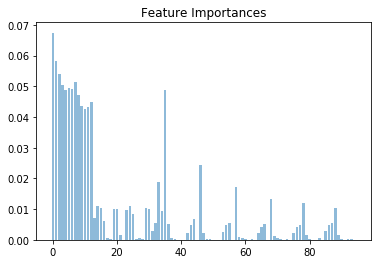
I also implemented Cross Validation by simply running the train\_test\_split() function.

After ran the models, I got the following accuracy score:

|  |  |
| --- | --- |
| **Models** | **Accuracy Score** |
| Random Forest Classifier | 0.99 |
| KNeighbors Classifier | 0.79 |
| Support Vector Classifier | 0.79 |

Random Forest Classifier was best fitted model; thus, I used this model to make the prediction.

Before making the prediction, I tried different feature elimination methods. As we know that feature elimination may help us to reduce overfitting, improves accuracy and reduces training time. First, I used Random Forest to see importance score for each variable.



As we can see from the chart, first 15 variables were most important.

After found the important features I used SelectFromModel, RFECV and RFE method to eliminates the features. During first feature elimination method (SelectFromModel) I created a selector object that will use the random forest model to keep the features that have an importance of more than median number. Then I transformed the data to create a new dataset containing only the most important features and created a new random forest classifier with the most important features. Original model and ‘limited’ model which contained only important features gave me almost same number of accuracies. When I used the second method (RFECV) I got the same accuracy level with the original model as well.

The test set is used to estimate the predictive accuracy of a classifier. I used the original model to predict the defaults on X-test dataset. Accuracy score on testing dataset was 0.813, which make me confident about this analysis.