Genet Sebehat

MSDS696 - Practicum II

Default Credit Card Project Final Summary

**Problem statement**

This project examines the data concerning credit card payment defaults, and the goal of the project is to predict the probability of default based on personal and previous payment information and providing solutions using data science methods. The big picture of the project is to study the data and understand it more using explanatory data analysis (EDA), and ultimately to build the best model and make good predictions in order to provide the agency with sound advice. To accomplish this, data analysis and statistics techniques will be used, including machine learning methods.

**Project Overview/Background:**

This dataset contains information on payment defaults, demographic factors, credit data, payment history, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The goal of the project is to identify the key drivers that determine the likelihood or probability of credit card default. I then predict the likelihood or probability of credit card default among the bank’s customers. I use data science techniques throughout the project to identify, define, and analyze business problems. Moreover, my aim with this project is to gain analytics experience and to reconcile mathematical theory with business practices.

**Type of Data Science Learning**

This project will incorporate several different types of classification models using supervised and unsupervised learning. This project utilizes various machine learning algorithms, including “logistic regression,” “K-Nearest Neighbor (KNN),” “Support Vector Machine (SVM),” “Gaussian Naive Bayes,” “Decision Tree Classification,” and “Random Forest Classification” to find the best model that will be useful for the business and that banks might use to build good models and predict the probability of credit card defaults among their customers.

**Objective**

* To analyze the data to help the bank to identify the key factors determining the probability of credit card default.
* To the possibility of credit card default for customers of the bank.

**Key analysis of the project**

* Shows an understand of the data through descriptive statistics and visualization
* The model I built here will use all possible factors to predict data on customers to find which will be defaulters in the following month.
* Compares the performance of these methods
* The goal is to find whether the clients are able to pay their next month’s balance due.
* Identifies potential customers for the bank who are able to settle their credit balance.
* Determines whether customers are able to make credit card payments on-time
* Default is the failure to pay interest or principal on a loan or credit card payment.

**Analysis and discussion is laid out in the following sections:**

* About the Data Set
* Data Wrangling/Cleaning
* Pre-processing of the Data Set to Implement Machine Learning Algorithm
* Exploratory Descriptive Analysis (EDA)
* Create Different Models and Compare Their Accuracies

**Tools Used**

* I analyzed the data set using the Python programming language in pandas with Jupyter notebook.

**About the Data Set**

Credit card issuers in Taiwan faced a cash and credit card debt crisis, and law-breaking was expected to peak in the third quarter of 2006 (Chou, 2006). In order to increase market share, card-issuing banks in Taiwan over-issued cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, overused credit cards for consumption and accumulated heavy credit and cash card debts. The crisis dealt a blow to consumer financial confidence and represented a significant challenge for both banks and cardholders.

**Acknowledgements**

Any publications based on this dataset should acknowledge the following:

Lichman, M. (2013). UCI Machine Learning Repository [[http://archive.ics.uci.edu/ml]](http://archive.ics.uci.edu/ml%5d). Irvine, CA: University of California, School of Information and Computer Science.

***Source:***

The original dataset can be found [here](https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients) at the UCI Machine Learning Repository

* **Key data features**

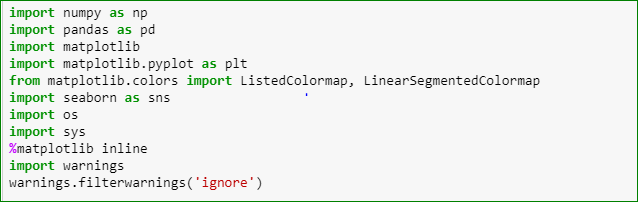
|  |  |
| --- | --- |
| **Default of Credit Card Client Dataset** | Dataset Descriptions |
| Data source | <https://www.kaggle.com> |
| Dataset | 30,000 rows and 25 columns (attributes) |
| Data representation | This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. |

**Dataset Information:**

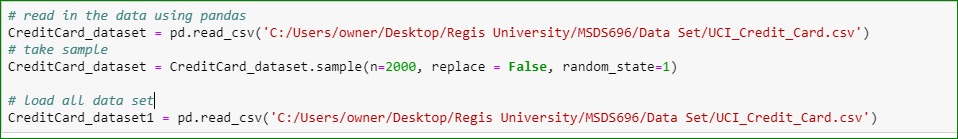
|  |  |
| --- | --- |
| UCI\_Credit\_Card | Dataset Descriptions |
| Data source | <https://www.kaggle.com>  <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients> |
| ID | ID of each client |
| LIMIT\_BAL | * Amount of given credit in NT dollars (includes individual and family/supplementary 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=paid on time, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months, etc.) |
| PAY\_2 | * Repayment status in August 2005 (same scale as above) |
| PAY\_3 | * Repayment status in July 2005 (same scale as above) |
| PAY\_4 | * Repayment status in June 2005 (as above) |
| PAY\_5 | * Repayment status in May 2005 (same scale as above) |
| PAY\_6 | * Repayment status in April 2005 (same scale 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) |
| BILL\_AMT1 | * Amount of bill statement 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) |

***Summary of the dataset types:***

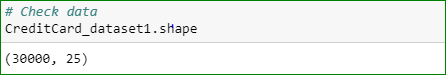
* Integer: 20
* Decimal: 4
* Key Id: 1
* ***Load packages***



* ***Load Data***



* ***Check Data***

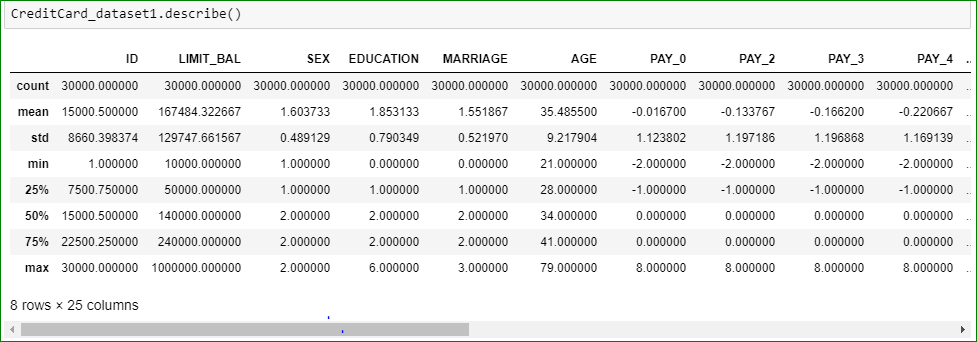


* ***Examine Data***

Started by looking at the data features (first 5 rows).



I then examined the data in more detail

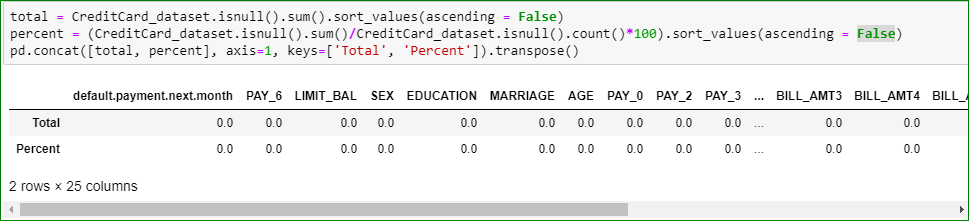


Summary of the above description:

* There are 30,000 distinct credit card clients.
* The average credit limit was 167,484. The standard deviation is unusually large, with the max value being 1M.
* Education level is mostly graduate school or university.
* Most of the clients are either married or single.
* Average age is 35.5 years, with a standard deviation of 9.2.
* As the value 0 for default payment means 'not default' and value 1 means 'default', the mean of 0.221 means that 22.1% of credit card contracts will default next month (this will be verified in subsequent sections of this analysis).
* **Data wrangling**

Data wrangling is the process of cleaning data by either removing rows with missing values or inputting missing values.

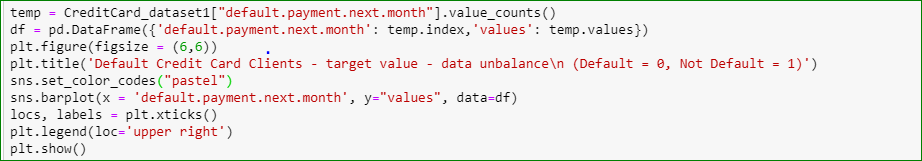
Data cleaning part I

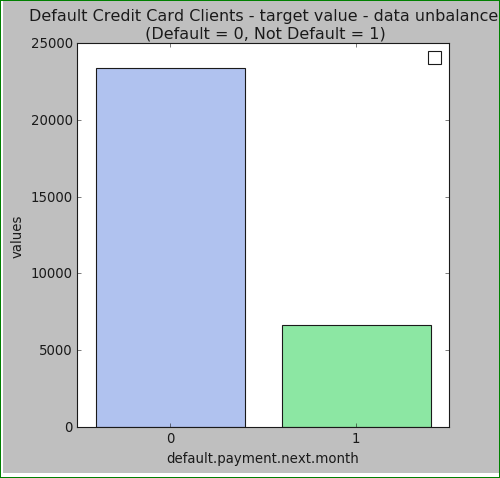


According to the above results, there is no missing data in the entire dataset.

* **Data unbalance**

I checked the data unbalance with respect to the target value, i.e., “default.payment.next.month”.

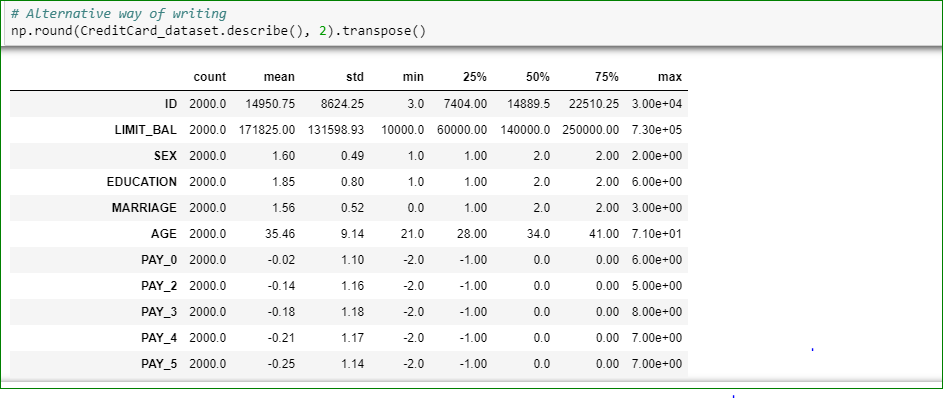


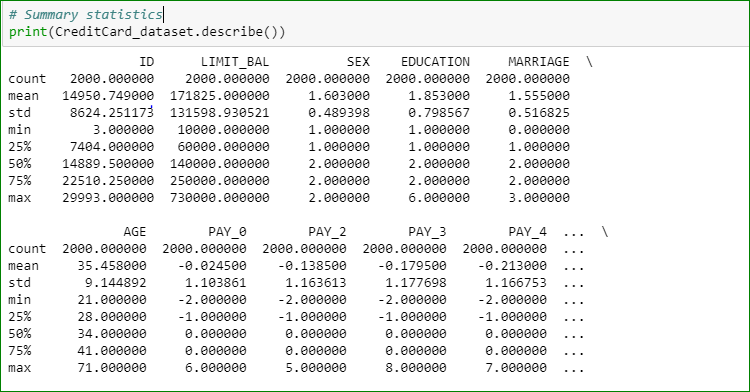


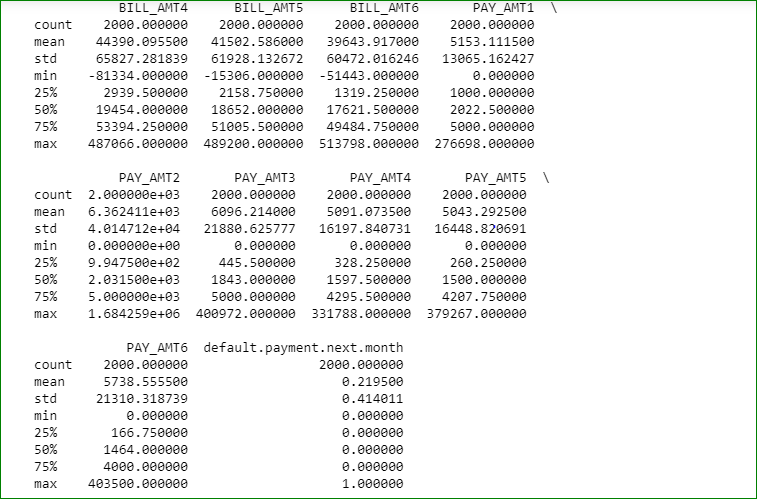
A total of **6,636** out of **30,000** (or **22%**) of clients will default next month. The data does not exhibit a large unbalance with respect to the target value (default.payment.next.month).

* **Descriptive Statistics**

A descriptive statistic is a summary statistic that quantitatively describes or summarizes features of a collection of information, while descriptive statistics is the process of using and analyzing those statistics. In other words, descriptive statistics is the use of data analysis to describe, show, or summarize data in a meaningful way ( ).





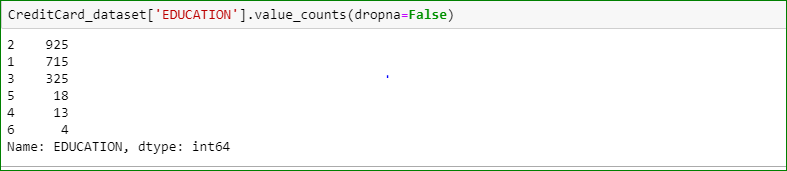


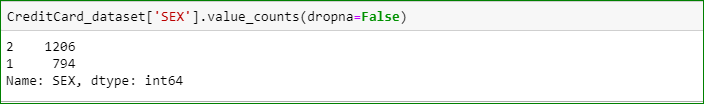
### **Data preprocessing**

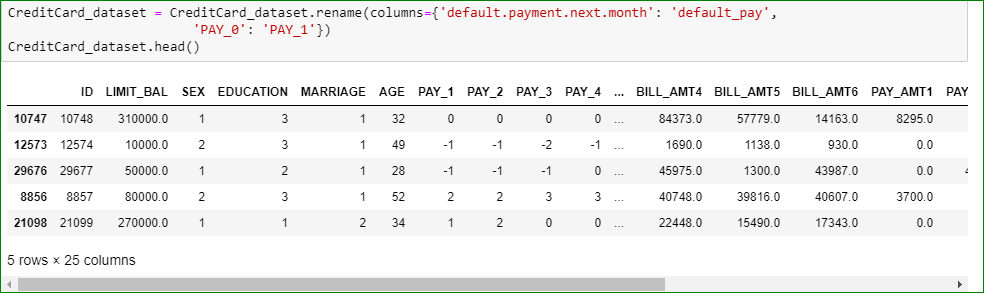
Data preprocessing refers to the steps applied to make data more suitable for data mining. The steps used for data preprocessing usually fall into two categories:

1. Selecting data objects and attributes for the analysis.
2. Creating/changing the attributes

**Data preprocessing part I**

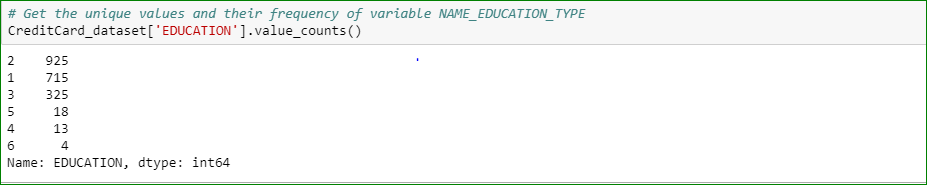






**Data preprocessing part II**



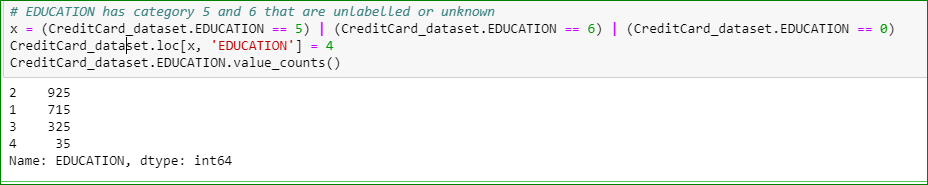


**Data Transformation I**

The 0 (undocumented), 5 and 6 (label unknown) in EDUCATION can also be put into the “Other” category (thus 4).

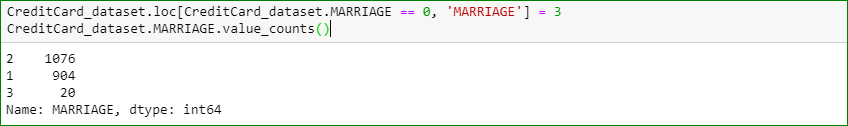
The loc () function is used to access a group of rows and columns by label(s) or by boolean array.

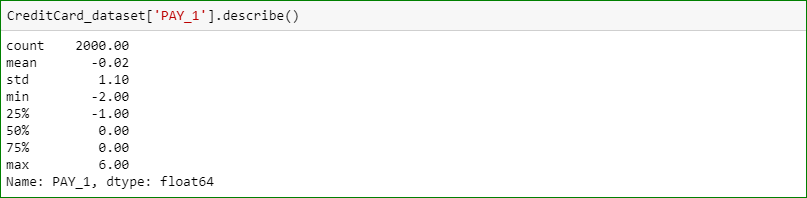
. .loc[] is primarily label based, but may also be used with a boolean array.



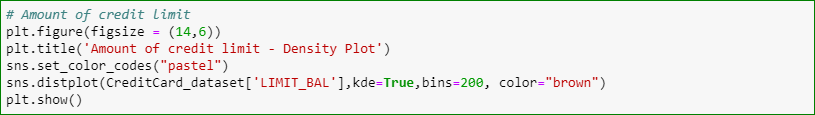
**Data Transformation II**

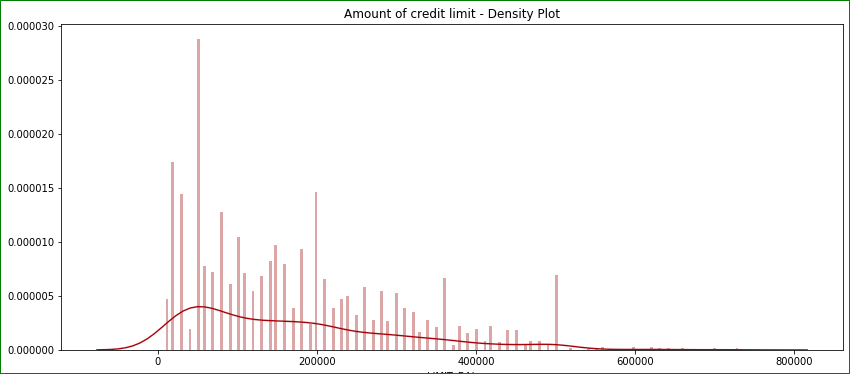
The 0 in MARRIAGE can be safely categorized as 'Other' (thus 3).



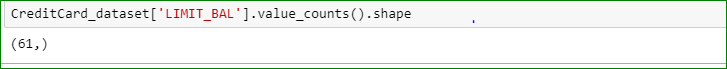


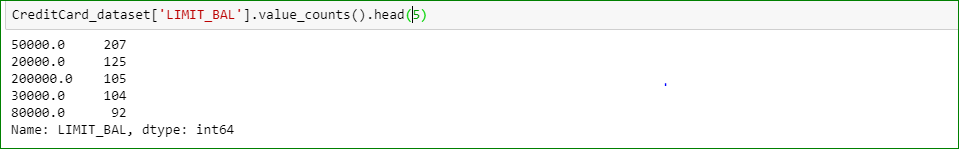
* **Explanatory data analysis (EDA)**
* Exploring and visualizing data helps to validate the business’s assumptions via thorough investigation
* It avoids any potential anomalies in data to avoid feeding incorrect data to a machine learning model
* It clarifies the model’s output and tests its assumptions





The most frequent value for credit limit is apparently 50K. Let's verify this.

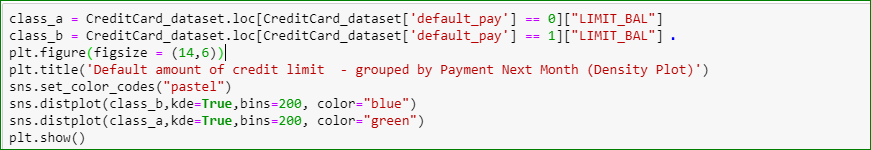


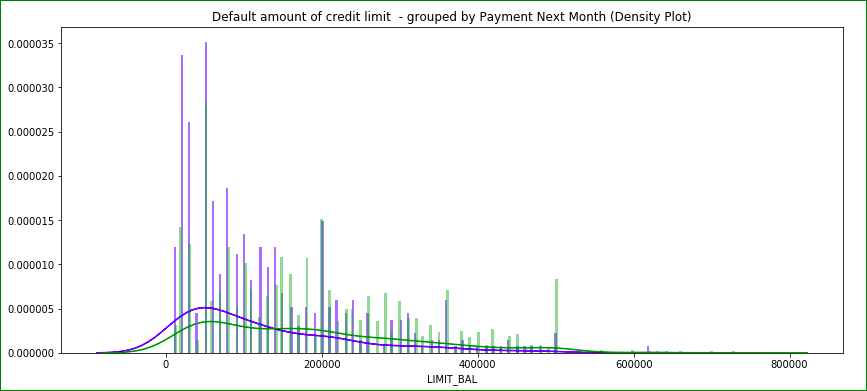


The largest number of credit cards have credit limits of 50,000 (207), followed by 20,000 (125), followed by 200,000 (105) and, finally, 30,000 (104).

**Credit limit grouped by payment default in the following month**

The figure below displays the density plot for credit limit (LIMIT\_BAL) grouped by default payment next month.

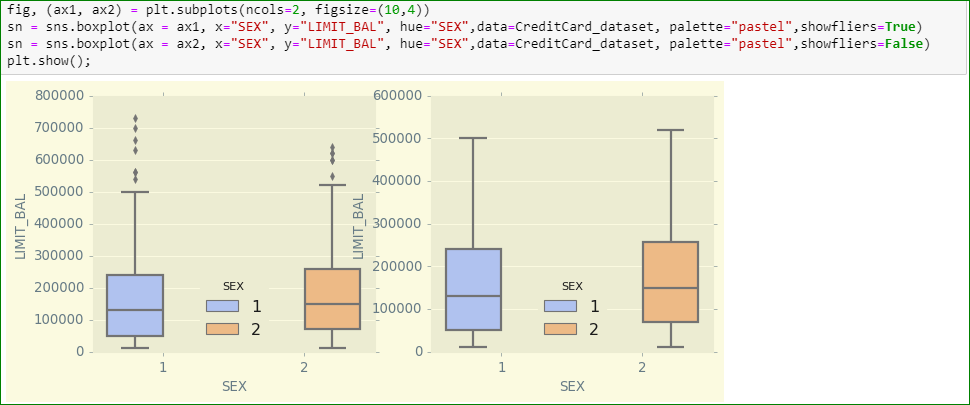




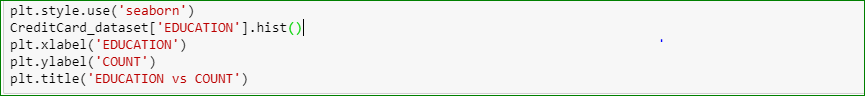
Most defaults occurred for credit limits between 0 - 100,000 (and density for this interval is larger for defaults than for non-defaults). The greatest number of defaults occurred among lcients with credit limits of 50,000, 20,000, and 30,000, in that order.

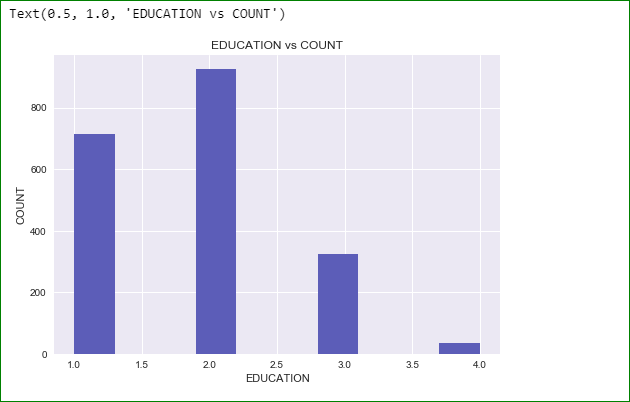
Credit limit vs. sex

Checking the credit limit distribution vs. sex. For sex, 1 stands for male and 2 for female.

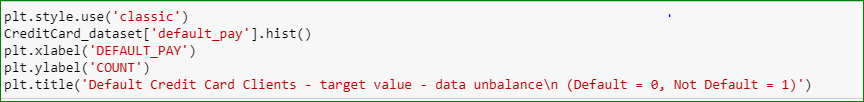


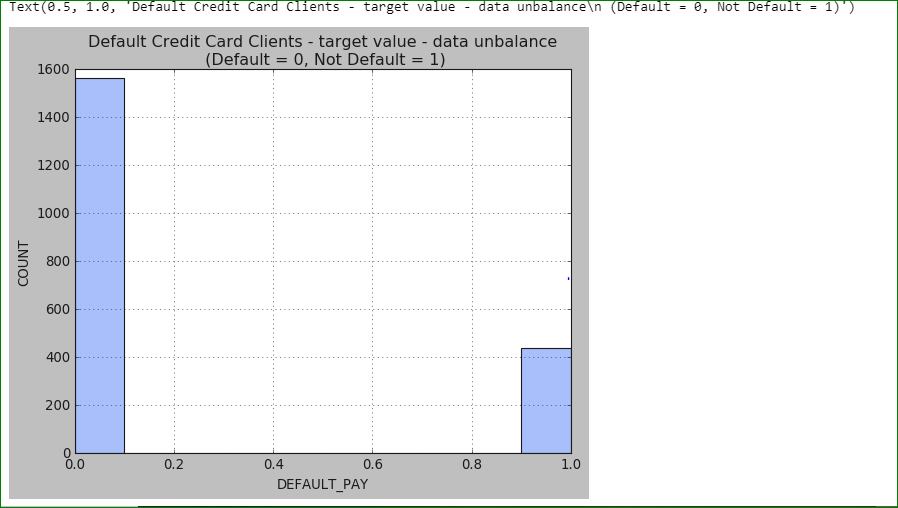
Credit limit amount is quite balanced between sexes. Men have a slightly smaller Q2 and larger Q3 and Q4 and a lower mean, while women have a larger outlier max value (above 700,000 NT dollars).





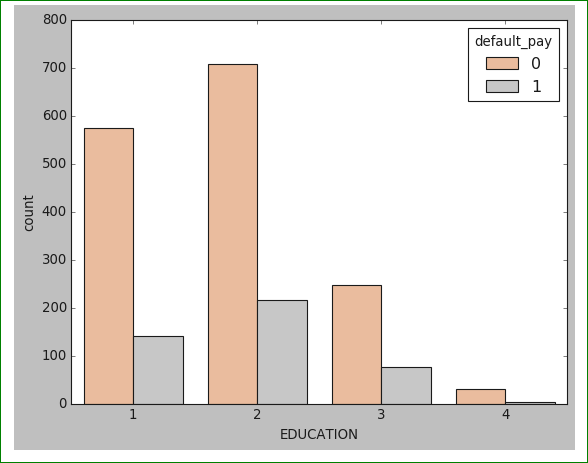
The highest credit limits were college students, graduate students, and high school graduates.





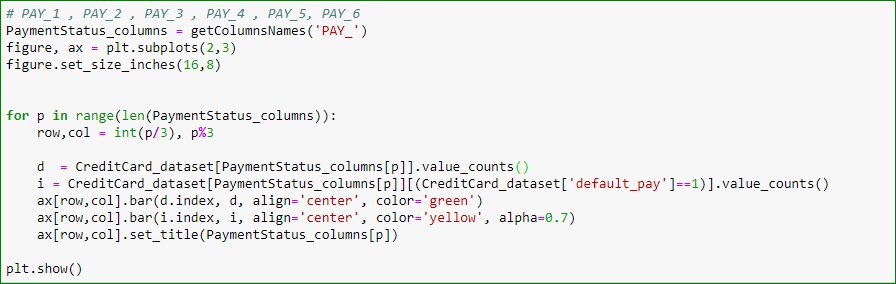
Most credit card holders were married.

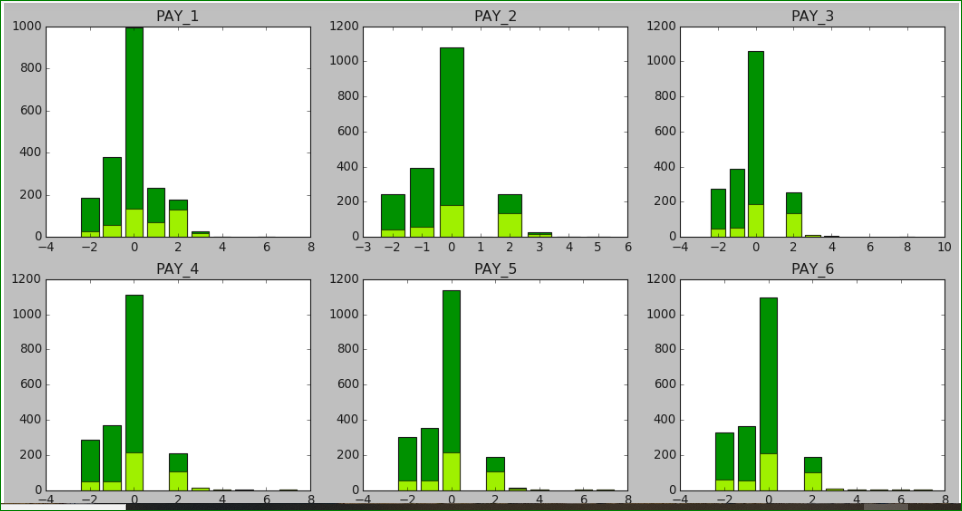




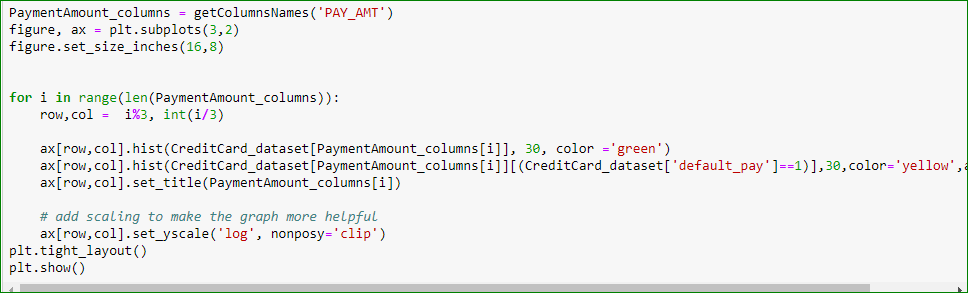
As the above plot shows those people who are university students have less payment defaults than graduates and high school people

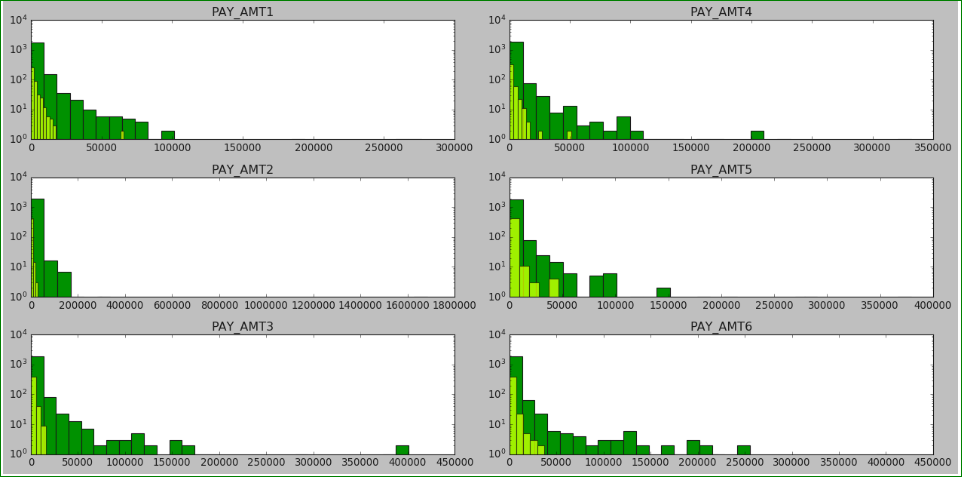




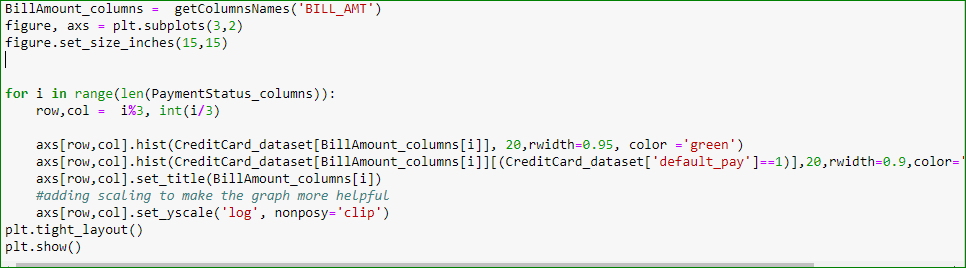


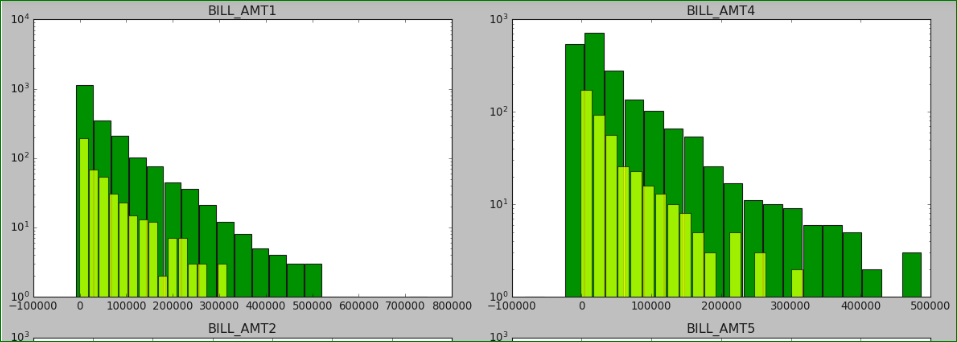
The above figure show bar plots for each month’s payment status, showing the count of defaulters and non-defaulters.

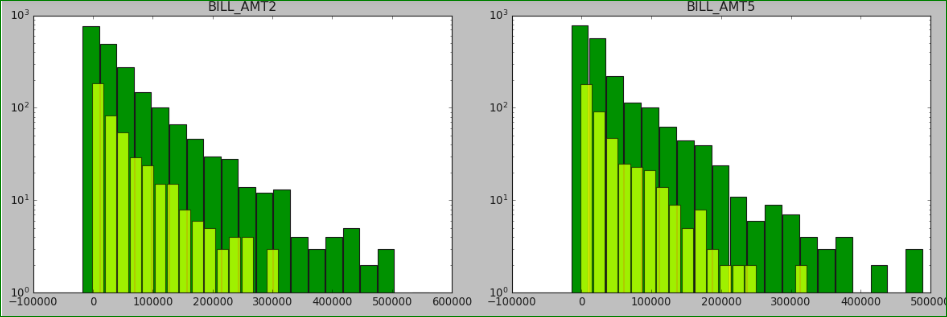


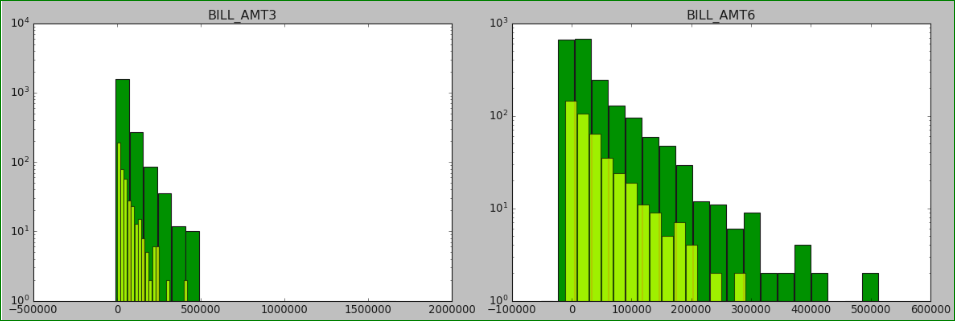


The above histogram shows the distribution of payment amount for each month explicitly for defaulters and non-defaulters.

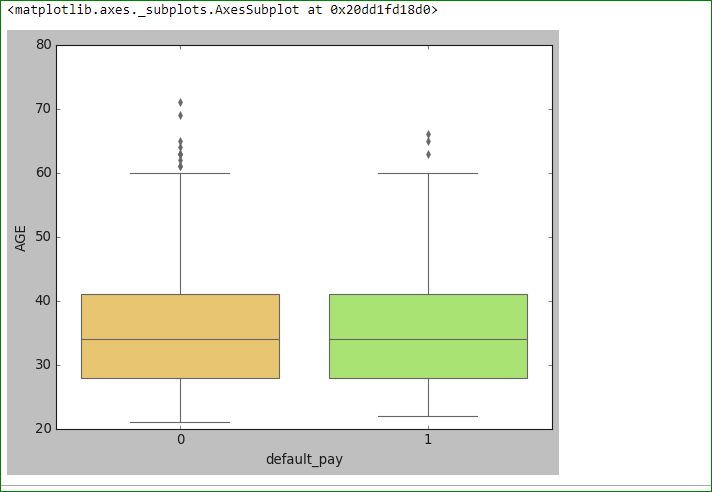




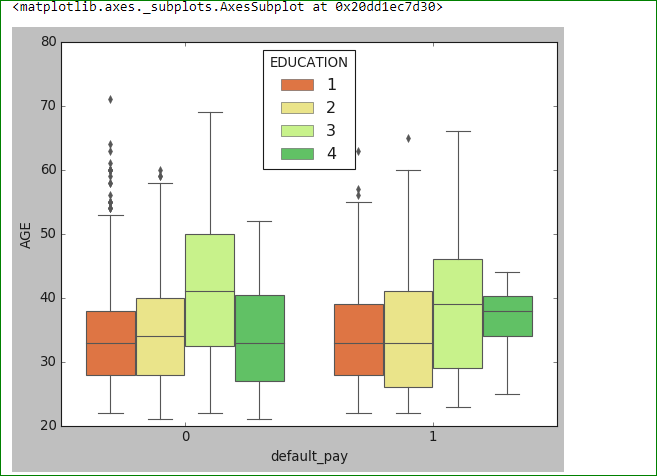




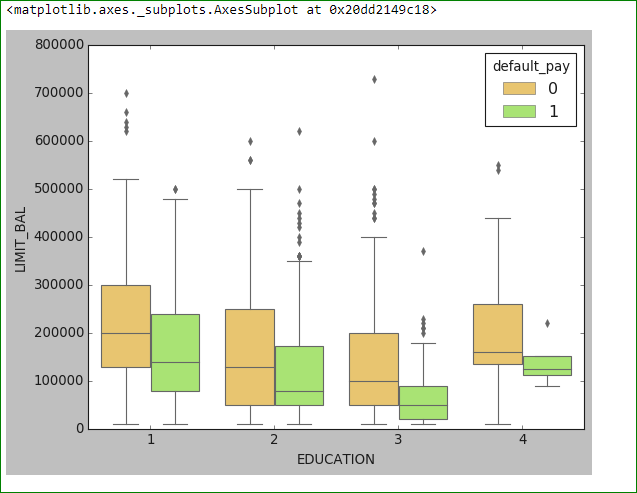




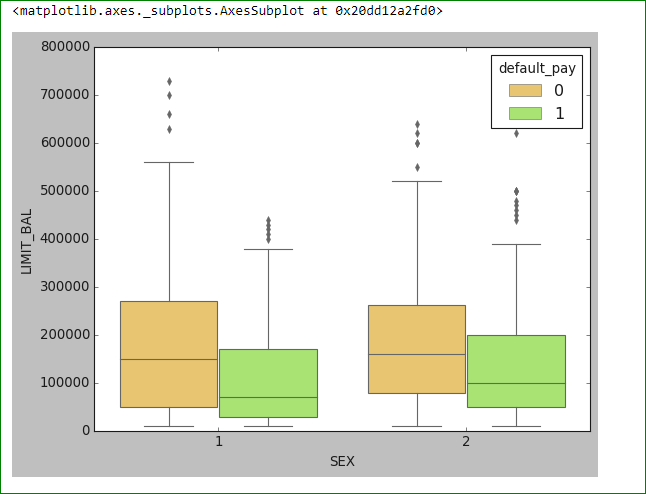












* **Correlation**

Variables within a dataset can be related for many reasons.

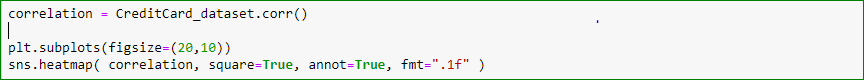
It can be useful in data analysis and modeling to better understand the relationships between variables. The statistical relationship between two variables is referred to as their correlation.

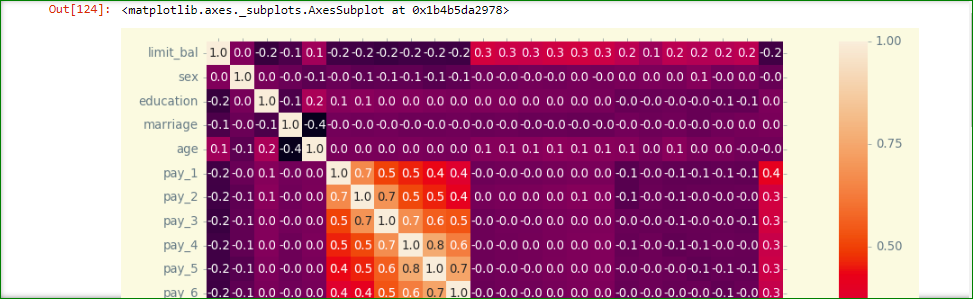
- Positive Correlation: Both variables change in the same direction.

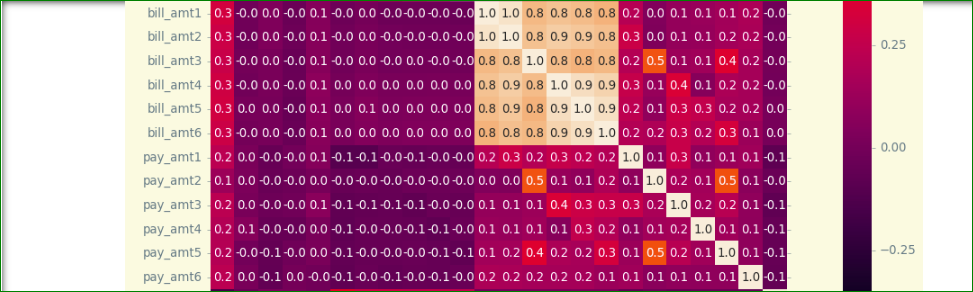
- Neutral Correlation: No relationship in the change of the variables.

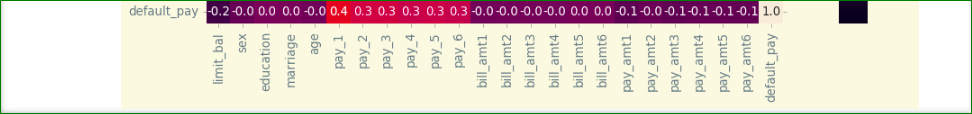
- Negative Correlation: Variables change in opposite directions.

Observing Correlations between Features of the Dataset





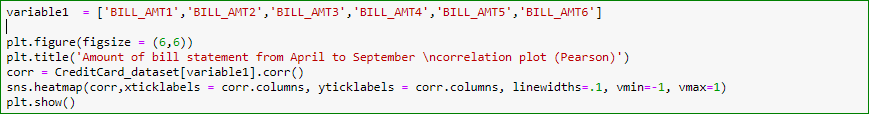


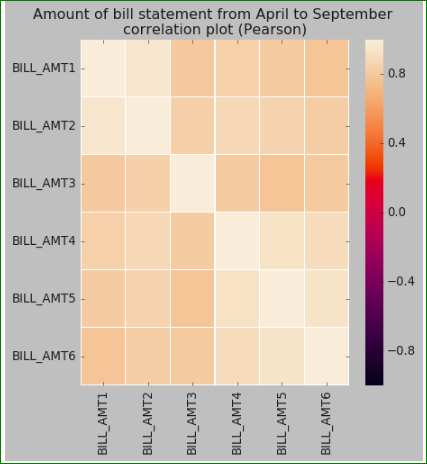


As the above heatmap shows, certain features are correlated with each other (collinearity), such us PAY\_1,2,3,4,5,6 and BILL\_AMT1,2,3,4,5,6. In those cases, the correlation is positive.

It appears that the PAY\_1 and PAY\_X variables are the strongest predictors of default, followed by the LIMIT\_BAL and PAY\_AMT\_X variables.

Checking the correlation of bill amount (BILL\_AMT1, BILL\_AMT2, …BILL\_AMT6) of statement in the April 2005 - September 2005

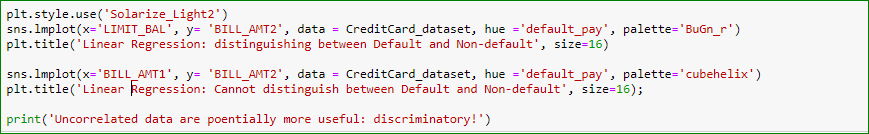


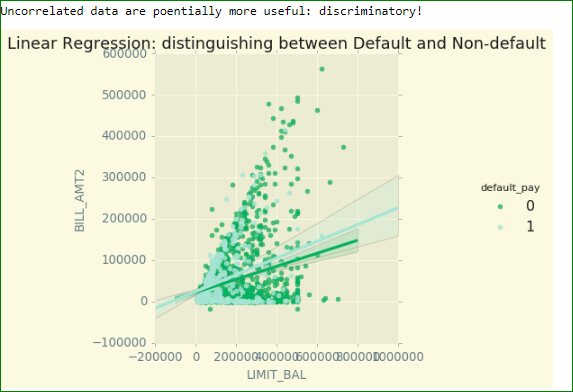


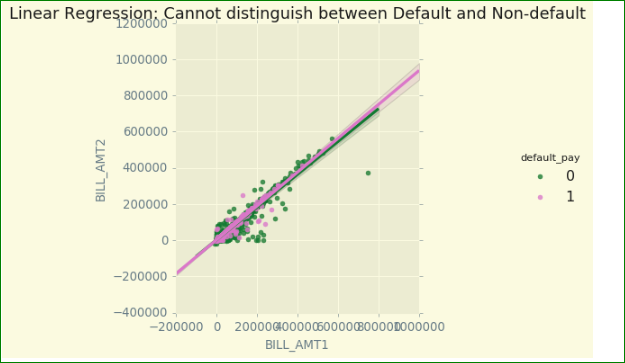
Correlation decreases with distance between months. The weakest correlation is

between September -April.

Separating fit\_correlated and uncorrelated data via linear regression:

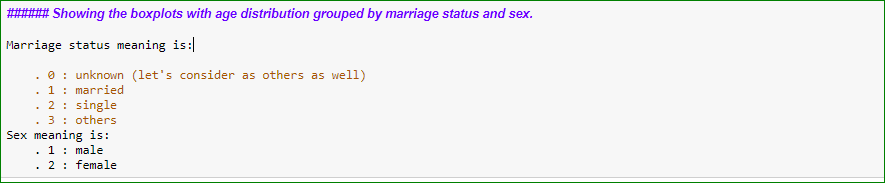


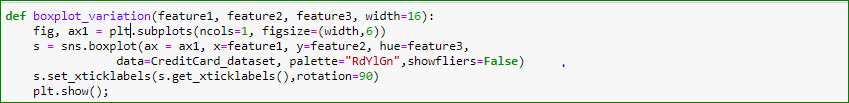




Correlation decreases with distance between months. Lowest correlations are between September - April.

Showing sex, education, age and marriage distributions.

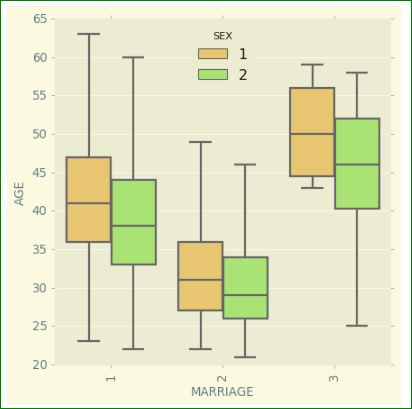


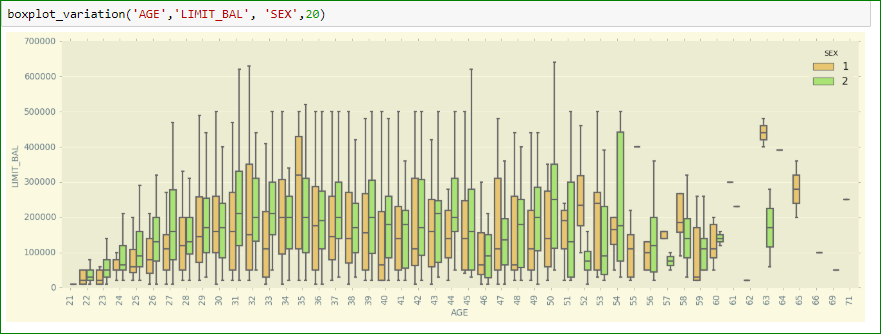


Age, sex, credit limit

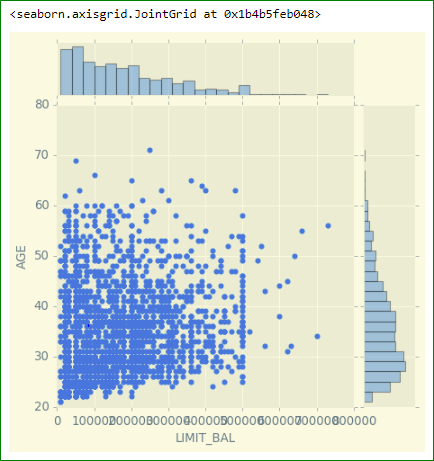
Below is a boxplot showing credit amount limit distribution grouped by age and sex.



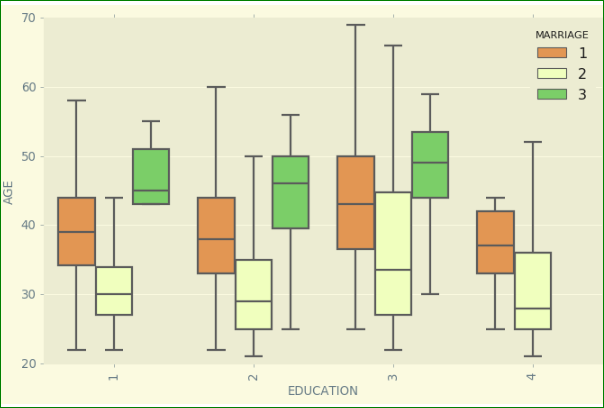


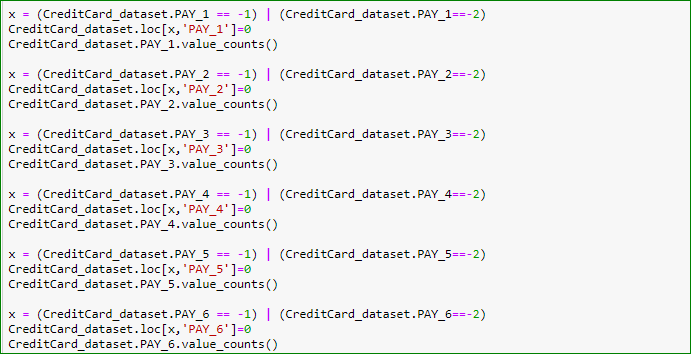


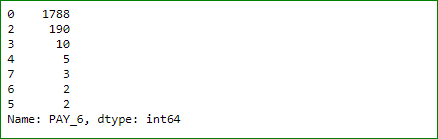




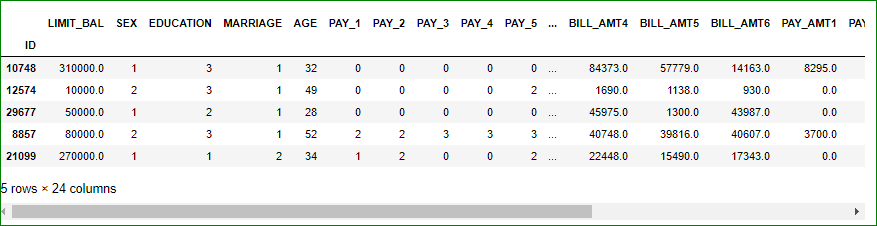




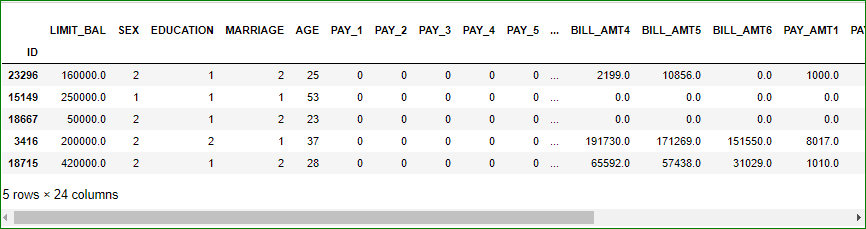




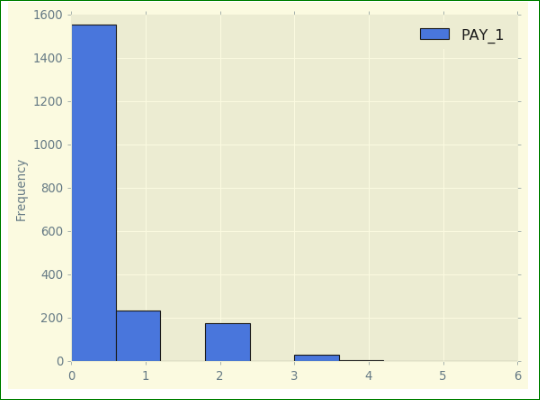




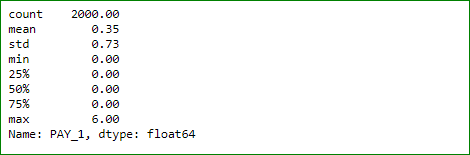


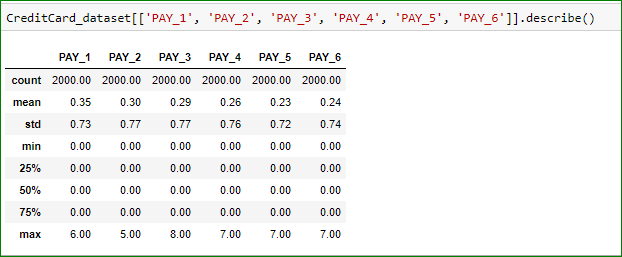


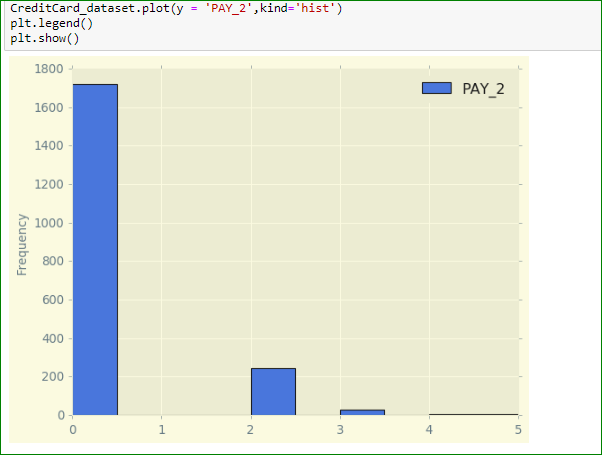


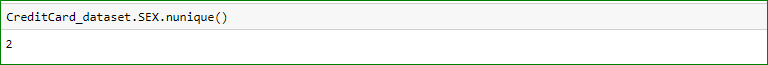


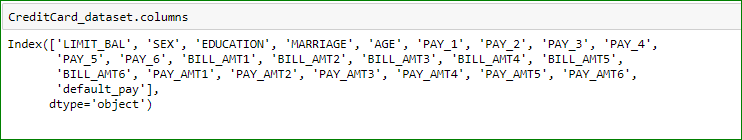












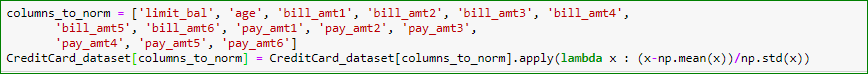
**One-hot encoding for categorical variable**

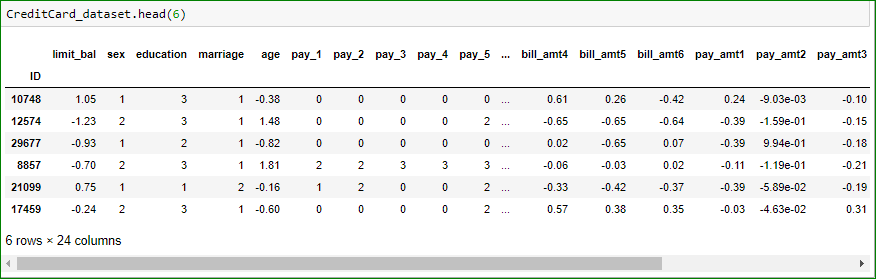
One-hot encoding is a representation of categorical variables as binary vectors. This first requires that the categorical values be mapped to integer values. Then, each integer value is represented as a binary vector that is all zero values except for the index of the integer, which is marked with a 1.

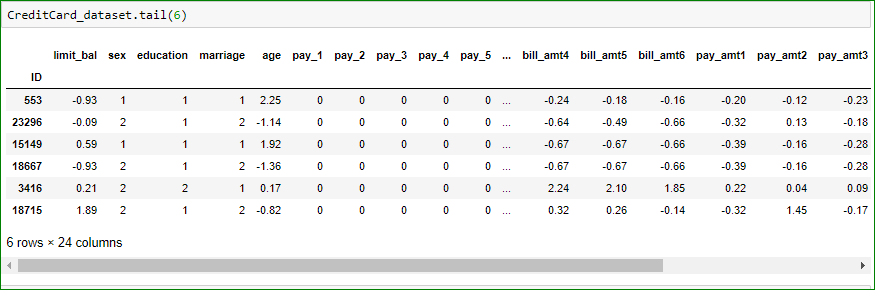




Scaling of Numerical Attributes



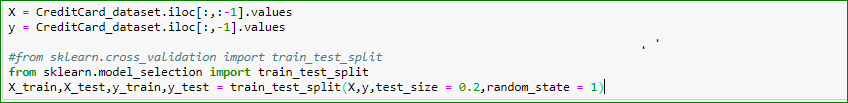




* **Predictive models**

Split dataset into training (80%) and test set (20%)

I will set “test\_size” to 0.2. This means that 20% of all the sample data will be used for testing, which leaves 80% of the data as training data for the model to learn from. Setting “random\_state” to 1 ensures that we get the same split each time so that we can reproduce our results.



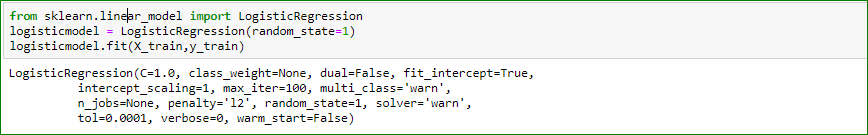




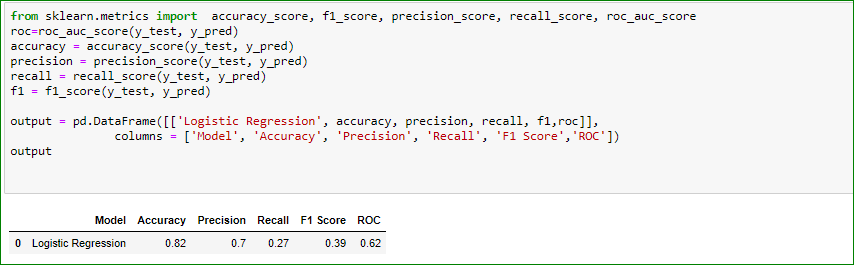
**Machine Learning Algorithm**

**Logistic Regression**

In Logistic Regression, we wish to model a dependent variable(Y) in terms of one or more independent variables(X). This is a method for classification. This algorithm is used for the dependent variable, which is, Categorical. Y is modeled using a function that gives an output between 0 and 1 for all values of X. In logistic regression, the sigmoid (aka logistic) function is used







**K-Nearest Neighbor (KNN)**

The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithm. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks.

We can implement a KNN model by following the below steps:

. Load the data

. Initialize the value of k

. To get the predicted class, iterate from 1 to the total number of training data points

. Calculate the distance between test data and each row of training data. Here we will use

Euclidean distance as our distance metric since it’s the most popular method. The other

metrics that can be used are Chebyshev, cosine, etc.

. Sort the calculated distances in ascending order based on distance values

. Get the top k rows from the sorted array

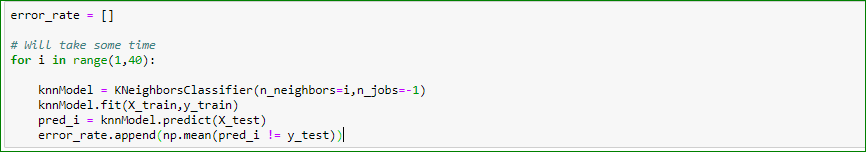
. Get the most frequent class of these rows

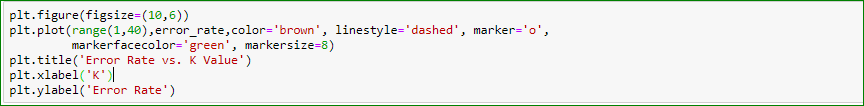
. Return the predicted class

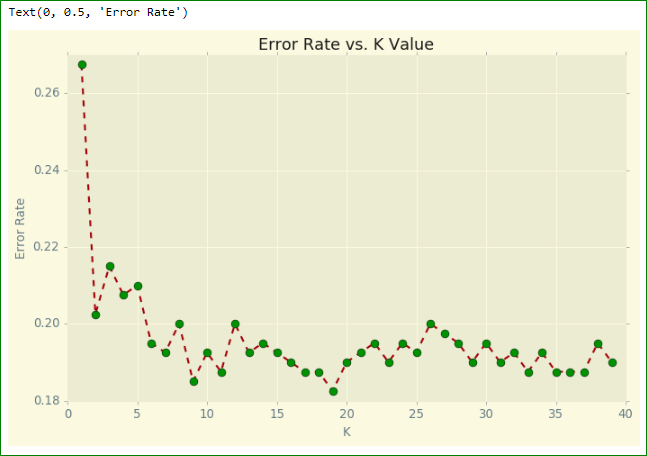
Using the elbow method to pick a good K Value!

Create a for loop that trains various KNN models with different k values, then keep track of the error\_rate for each of these models with a list. Refer to the lecture if you are confused on this step.

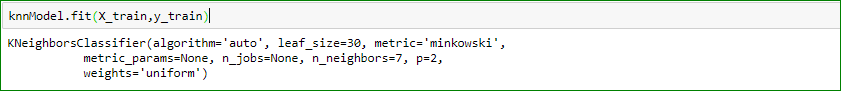




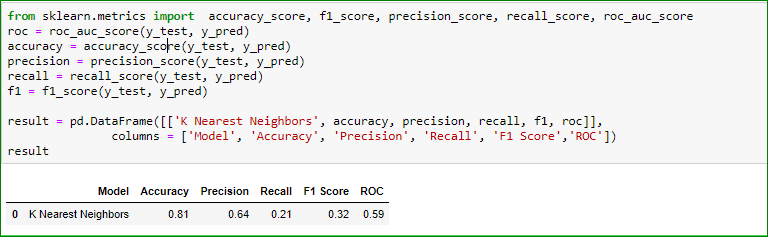








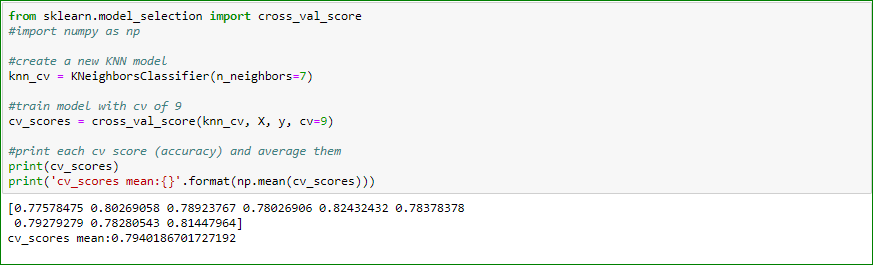




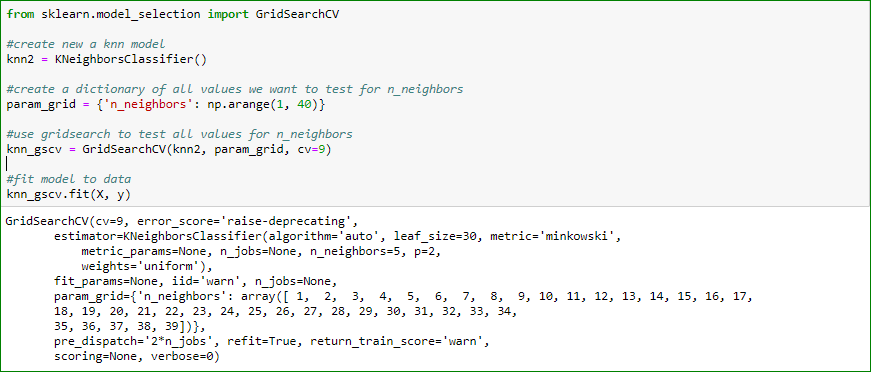
**k-Fold Cross-Validation**

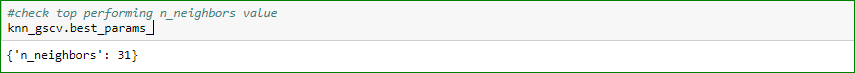
Cross-validation is when the dataset is randomly split up into ‘k’ groups. One of the groups is used as the test set and the rest are used as the training set. The model is trained on the training set and scored on the test set. Then the process is repeated until each unique group as been used as the test set.

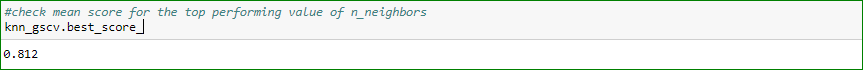




Using cross-validation, our mean score is about 79.40%. This is a more accurate representation of how our model will perform on unseen data than our earlier testing using the holdout method.







By using grid search to find the optimal parameters for our model, we have improved our model accuracy by ≈ 2%.

**Support Vector Machine**

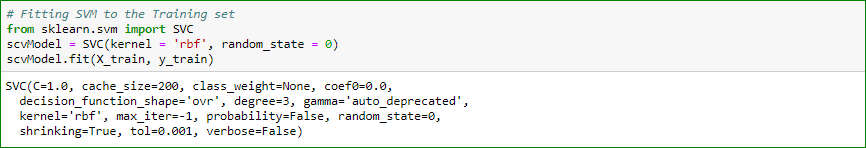
Generally, the use of Support Vector Machines (SVM) is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplanes in an iterative manner, which is used to minimize error.

**Advantages**

SVM Classifiers offer good accuracy and perform faster predictions compared to Naïve Bayes algorithms. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space.

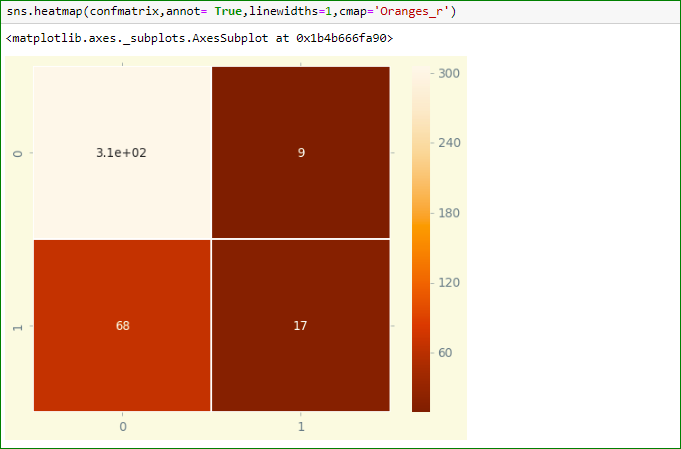
## **Disadvantages**

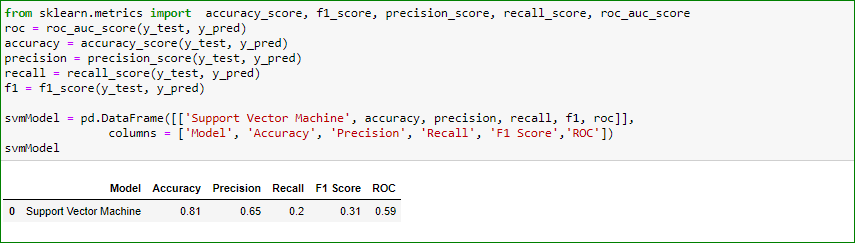
SVM is not suitable for large datasets because of its high training time. It also requires more time in training compared with Naïve Bayes. It works poorly with overlapping classes and is also sensitive to the type of kernel used.





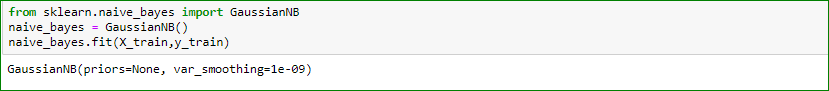






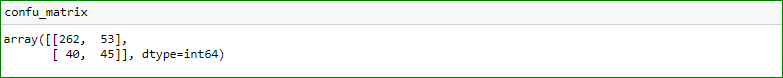
**Gaussian Naive Bayes**

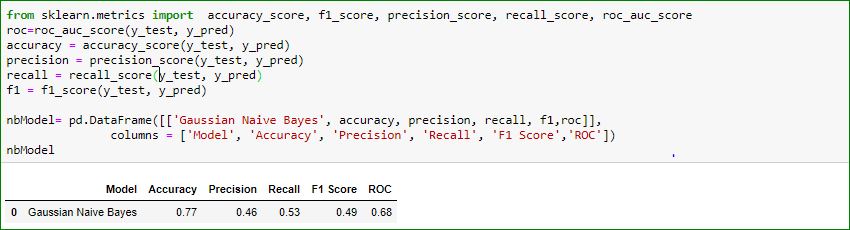
Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data.











**Decision Tree Classification**

A Decision Tree is a simple representation for classifying examples. It is a form of Supervised Machine Learning where the data is continuously split according to a certain parameter.

**Advantages of Decision Trees**

1. The algorithm is simple to understand, interpret, and visualize, as the idea is mostly used in our daily lives.

2. Decision Trees can be used for both classification and regression problems.

3. Decision Trees can handle both continuous and categorical variables.

4. No feature scaling (standardization and normalization) is required for Decision Trees because they use as rule-based approach instead of distance calculation.

5. Decision Trees handle non-linear parameters efficiently.

6. Decision Trees can automatically handle missing values.

7. Decision Trees are usually robust to outliers and can handle them automatically.

8. The training period is less as compared to Random Forest because it generates only one tree, unlike forest of trees in the Random Forest.

**Disadvantages of Decision Trees**

1. Overfitting: This is the main problem with Decision Trees. It generally leads to overfitting of the data, which ultimately leads to wrong predictions.

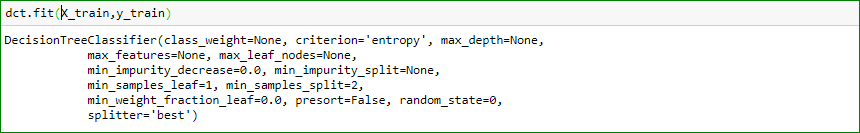
2. High variance: Decision Trees generally lead to the overfitting of data. Due to this overfitting, there is a very high chances of high variance in the output, which leads to many errors in the final estimation and shows high inaccuracy in the results.

3. Unstable: Adding a new data point can lead to re-generation of the overall tree, and all nodes then need to be recalculated and recreated.

4. Affected by noise: A little bit of noise can create instability, leading to incorrect predictions.

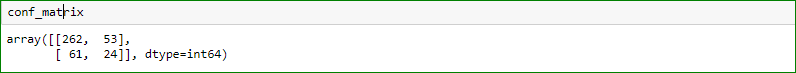
5. Not suitable for large datasets: If data size is large, then one single tree may grow complex and lead to overfitting. Thus, in this case, we should use Random Forest instead of a single Decision Tree.

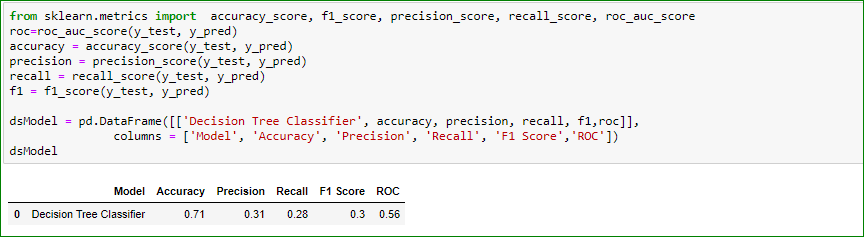












**Random Forest Classification**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. However, it is mainly used for classification problems. As we know, a forest is made up of trees, and more trees means a more robust forest.

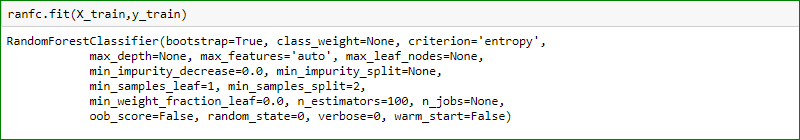
1. Pick at random K data points from the training set

2. Build the Decision tree associated to these K data points

3. Choose the number of trees(n) you want to build and repeat number 1 and number 2

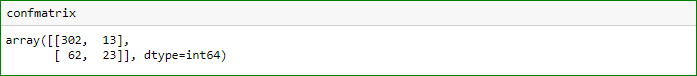
For new data points, make each one of your 'n' trees predict the category to which the data point belongs and assign the new data point to the category that wins the majority vote

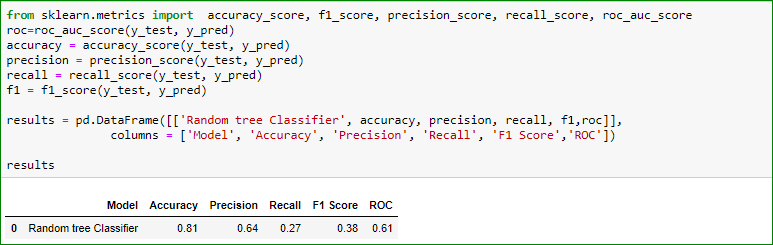












**Summary of predictive models**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **Roc** |
| **1** | **Logistic Regression** | **0.82** | **0.70** | **0.27** | **0.39** | **0.62** |
| **2** | **K-Nearest Neighbor** | **0.81** | **0.64** | **0.21** | **0.32** | **0.59** |
| **3** | **Support Vector Machine** | **0.81** | **0.65** | **0.20** | **0.31** | **0.59** |
| **4** | **Gaussian Naive Bayes** | **0.77** | **0.46** | **0.53** | **0.49** | **0.68** |
| **5** | **Decision Tree Classification** | **0.71** | **0.31** | **0.28** | **0.30** | **0.56** |
| **6** | **Random Forest Classification** | **0.81** | **0.64** | **0.27** | **0.38** | **0.61** |

As displayed in the above table, which contains a summary of the predictive model the logistic regression model is the best model. Therefore, I would like to suggest implementing a logistic regression model for this kind of business.

**Conclusion**

Overall, this project analyzes data regarding credit card payment defaults. The purpose of the project is to predict the likelihood of default based on individual and prior payment information in order to deliver solutions to the bank credit card customer agency using data science approaches. I investigated the data, checked data unbalancing, visualized the data’s features, and worked to understand the relationships between the different features. I then investigated six predicative models, finding “Logistic Regression” to be the best model for this specific business. In general, I was able to practice and review my previous data sciences classes such as “Machine Learning” and “Data Collection and Preparation.” Besides that, I was able to learn new techniques such as how to check unbalance data and how to work on cross validation using Python in pandas and so on. Moreover, I was able to practice and utilize the data statistics that I have learned about so far in the Data Science course programs. Overall, this project was very interesting; it gave me the opportunity, to practice data visualization and also to work widely with machine learning algorithms using Python 3 in panda.

References

Allibhai, E. (2018). Towards Data Science. Building a k-Nearest-Neighbors (k-NN) Model with Scikit-learn. Retrieved from <https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a>

Brownlee, J. (August 2019). Machine Learning Mastery. Making Developers Awesome at Machine Learning.

Retrieved from <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/#:~:text=A%20one%20hot%20encoding%20is,is%20marked%20with%20a%201>

Harrison, O. (2018). Towards Data Science. Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved from <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

Hills, K., Madhapur, Hyderabad, Telangana. (2020). Tutorials Point. Classification Algorithms - Random Forest. Retrieved from <https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_random_forest.htm>

Navlani, A. (2020). DataCamp. Decision Tree Classification in Python. Retrieved from <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>

Navlani, A. (2020). *Support Vector Machines with Scikit-learn***.** Data Camp. Retrieved from <https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python#svm>

Kumar, N. (2012). The professional Point. Advantages and Disadvantages of Decision Trees in Machine Learning. Retrieved from <http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of.html>

L. Breiman, J. Friedman, R. Olshen, and C. Stone. scikit-learn developers (BSD License). (2007 - 20200). [Decision Tree Classifier](Decision%20Tree%20Classifier). Retrieved from: <https://scikit-learn.org/stable/modules/tree.html#classification>

Patel, H., Mantri, N. (2020). Gaussian Naive Bayes.Machine learning (LM).Retrieved from<https://iq.opengenus.org/gaussian-naive-bayes/>