**Analysis of default of credit card client**

Group 6

Noah Olsen Jianing Wang Jia Chen

Professor Amir Jafari

DATS6202: Machine learning I

August,, 2018

Outline

Introduction

Description of the data set

Learning network and algorithm

Experimental setup

Result

Conclusion

Reference

Appendix

1. Introduction.

As the use of credit cards become more convenient and offer better rewards than debit cards, credit cards became a more common method of payment in everyday use. But an impact of this method is, the more people use credit cards, the more problem arise. One of the problems is defaulting payments. If there was a method to predict the possibility of a consumer defaulting on payments, using computationally guided discovery, then this would save the company a tremendous amount of loss through proactive reporting.

This study aims to predict the possibility of default payments of credit card clients. To solve this problem, we plan to use three different approaches (Support Vector Machine, Random Forest and Naïve Bayes Model) will be used to forecast the default and the predictive accuracy of the three methods will be compared.

2. Description of the data set.

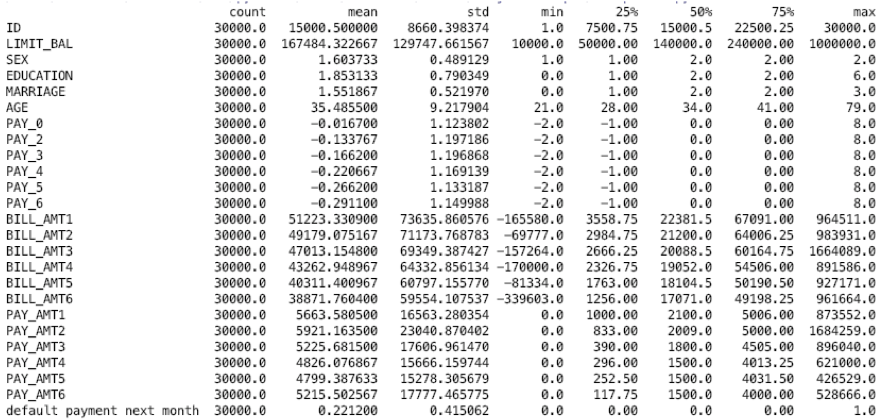
Our dataset is looking at defaults on credit card payments in Taiwan. Taken from the UCI Machine Learning Repository at this link. The dataset has 30,000 observations and includes 23 explanatory variables along with a binary target variable in the last column on whether the person defaulted on the next month’s payment (1 = Yes, 0 = No). Some brief descriptive info on the variables is presented in the Table 1 below using the pandas describe function.

Table 1

Column 1 is a unique person ID, column 2 is the person’s balance limit, column 3,4,5 and 6 are the person’s sex, education level, marriage status, and age respectively. Columns 7-12 contain a history of past payments on a monthly level for the past 6 months coded based on what their payment status is like. Columns 13-18 have their monthly bill totals for the past 6 months. Columns 19-24 contain the dollar amounts they have repaid off of their bill on a monthly level for the past 6 months. After a brief EDA our plan is to test several different types of models to help predict whether or not the person is going to default on their bill.   
We hope to test a Support Vector Machine, a Random Forest, Neuron Network and a Naïve Bayes Model. After looking at the misclassification rates for the 3 basic models we start we will choose 1 type of model to then optimize for performance and try to minimize the misclassification rate.

3. Learning network and algorithm (background information on the development of the algorithm )

3.1 random forest

3.2 NN

3.3 Naïve Bayes

Naïve Bayes is one of classifiers in machine learning area basing on the Bayes' theorem. The aim of Naïve Bayes is to get the posterior probability, which is what kind of outcome basing on a series of given conditions. Figure 1 shows the basic equation ofNaïve Bayes*.*

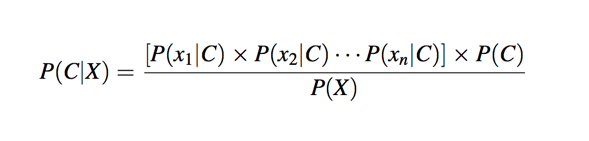
**

Figure 1

First it calculates the posterior probability for each class variable by prior probabilities, frequency tables and likelihood tables. Then output is the class with the highest probability with conditions. During the entire process of Naïve Bayes, necessary calculations are some simple probability multiplication. Because of that, we can apply this algorithm faster than the other.

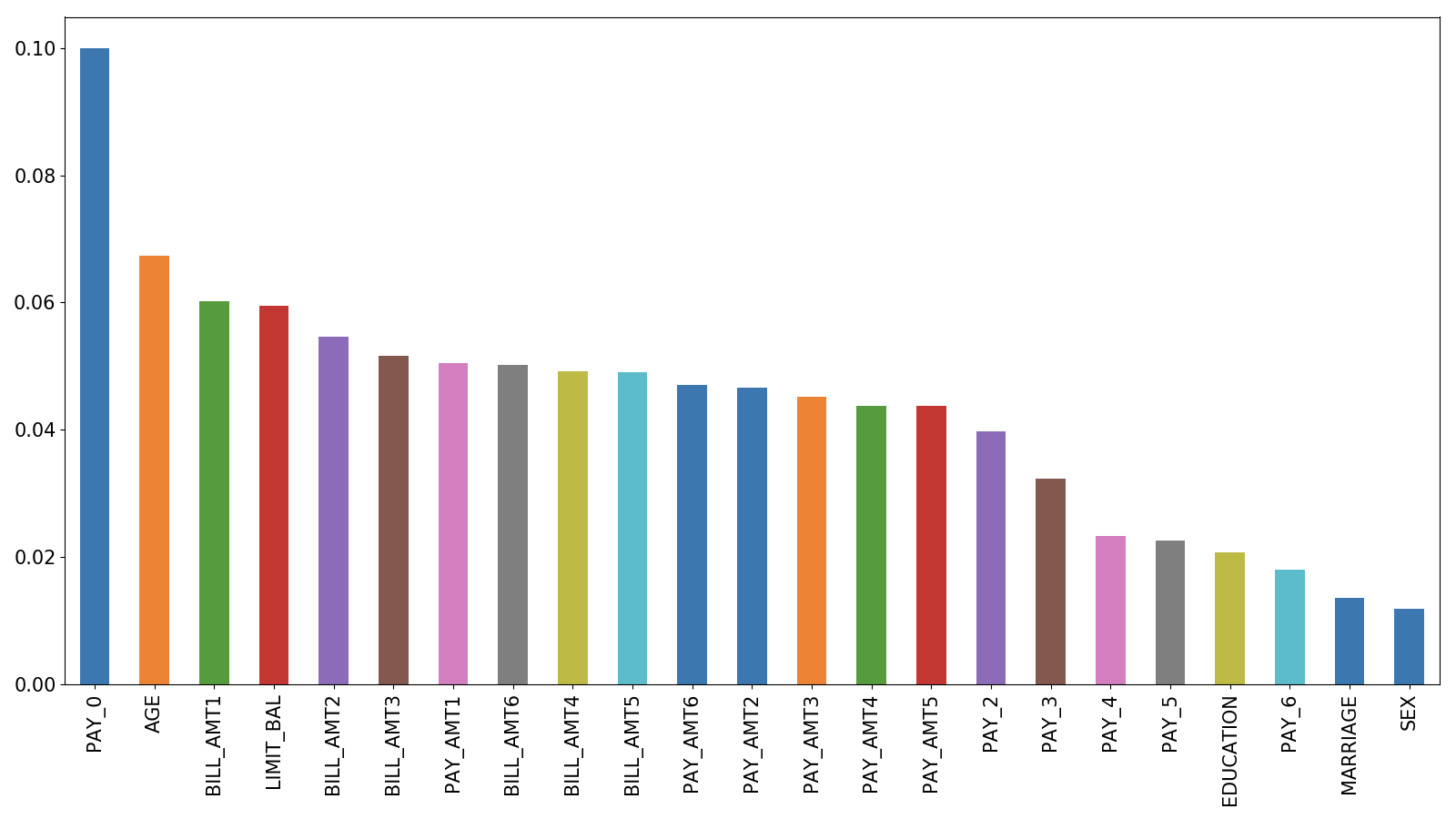
There are three different functions about Naïve Bayes we can find in Scikit-learn package: GaussianNB, BernoulliNB and MultinomialNB. The gaussianNB is often used for continuous variables, BernoulliNB is applicable to multiple binomial distributions, while MultinomialNB usually is for those discrete features.

3.4 SVM

4. Experimental setup

4.1

For the Random Forest model we started by using sklearn RandomForestClassifier function. I started by looking at the feature importance which yielded the plot below.



Given the fact that the dataset doesn’t have that many features to begin with combined with the fact that many of the features appear to have similar importance I decided to use all of the features in the final model. I decided to use 100 classifiers as it performed the best without being too computationally intensive. I also tried using both a gini and entropy models. Other than that there are not too many tunable parameters in random forest models.

4.2 NN

4.3 Naïve Bayes

After reading in the dataset, the first thing is checking the type of the feature and class. Half of the feature are discrete variables and the rest is continuous variables. Then we use function (Figure 2) to observe the distribution of the continuous variables. However, the last 12 columns distribute concentrated and almost have no normality trend. So, we decide to implement MultinomialNB function to train the data.

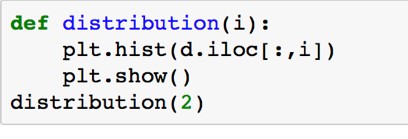


Figure 2

There are two prerequisites to apply MultinomialNB: discrete variables and non-negative variables. We transfer the continuous various to discrete ones and switch the negative data to non-negative. (Figure 3). After that we train the data, and the accuracy is only 58%. One of assumptions of Naïve Bayes is independence. If we can drop certain dependent variables, the accuracy of the model may go up. The next step is to check all the correlation (Figure 4) between each feature and delete some columns which are high correlated with each other. However, the result only increases by 0.1%.

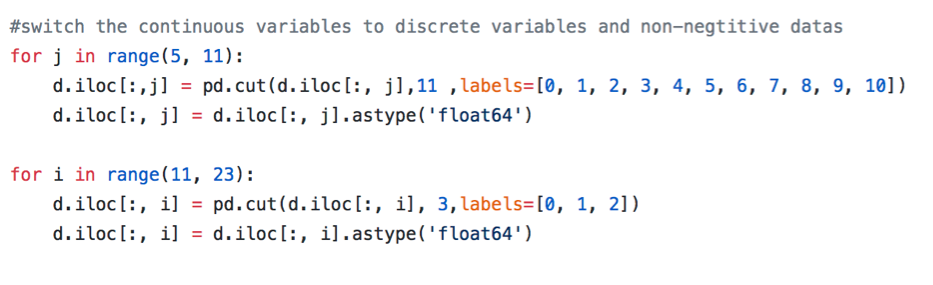


Figure 3

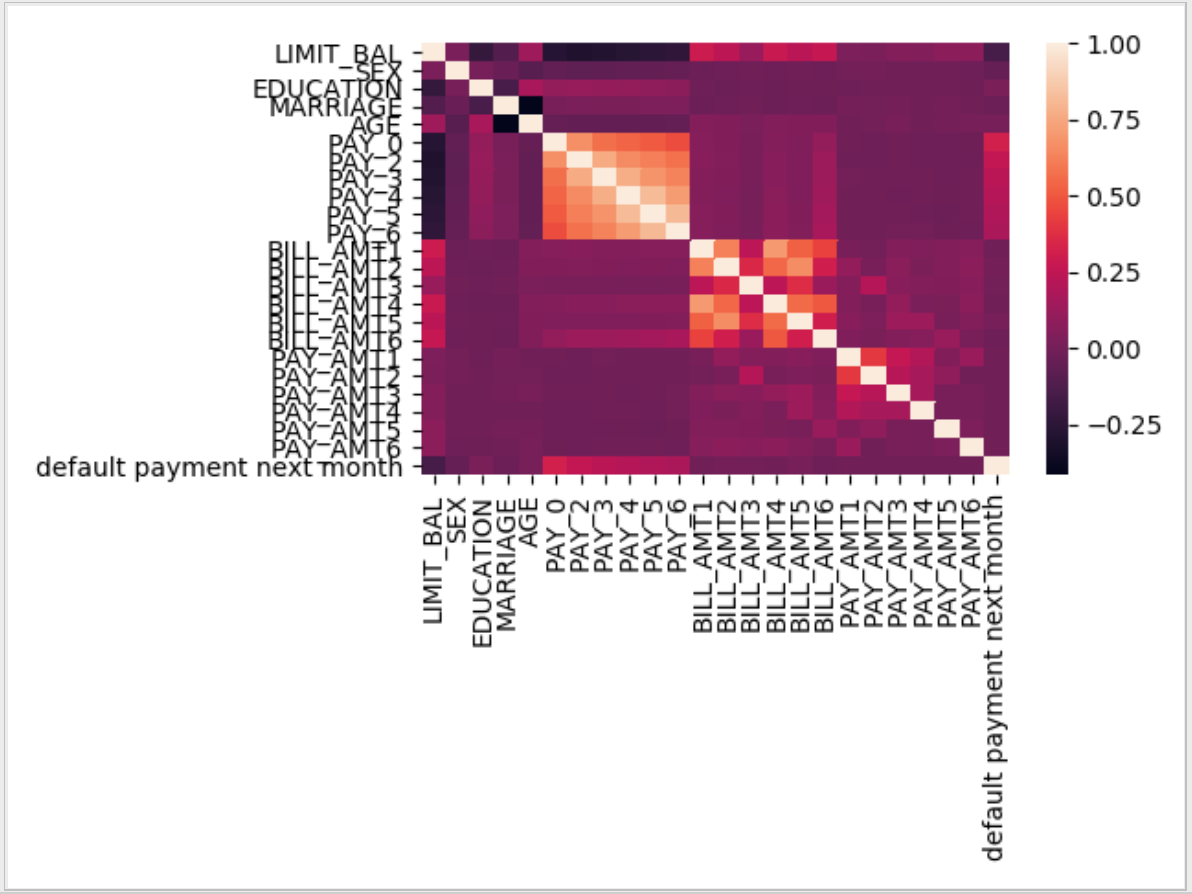


Figure 4

Because the accuracy is not good enough, so we want to try a different function. We can use BernoulliNB function only when every feature is binomial distribution. The last one we can experiment is GaussianNB. To apply GaussianNB function, the first thing to process data is normalizing (Figure 5). On one hand, it could decrease the probability of overfitting. On the other hand, after normalization, each feature has same scale to the outcome which is another assumption in Naïve Bayes and it can switch mean and variance to 0 and 1. The result rises up to 77.8% by using GaussianNB after it.

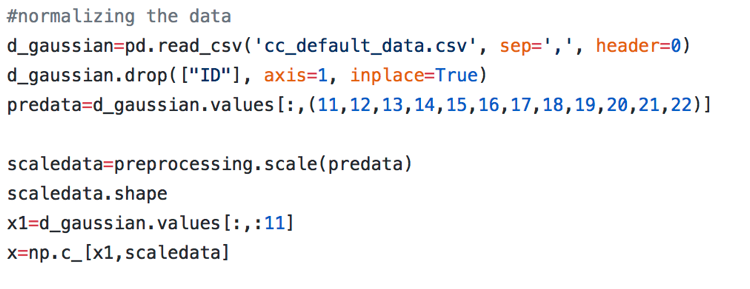


Figure 5

4.4 SVM

5. Result

5.1 random forest

5.2 NN

5.3 Naïve Bayes

Figure 6 provides the classification report and confusion matrix of multinomial function. As can be seen in the percentage of recall, more test data which belongs to class 1 (default payment) have been predicted to the right class comparing to those in class 0. However, this is not a really good model as the accuracy is only 58.1%.

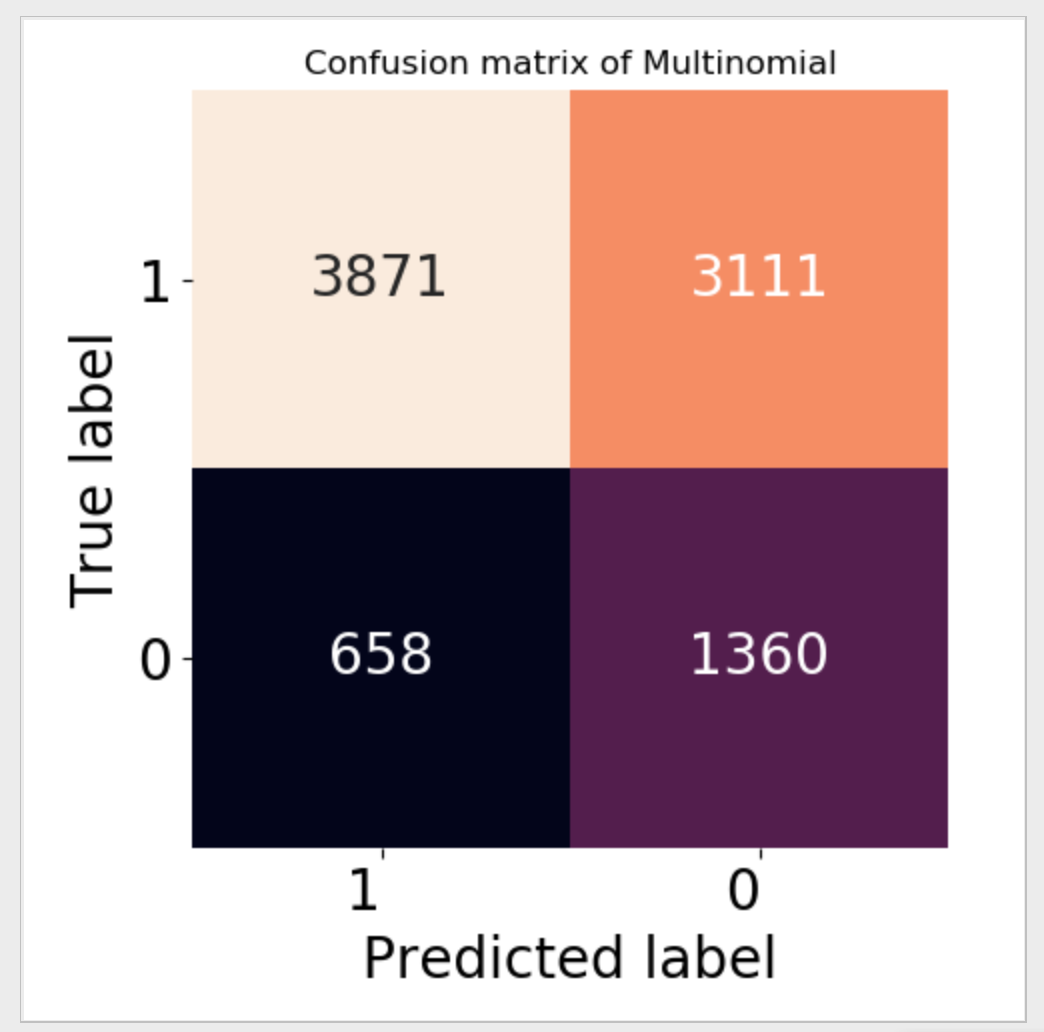
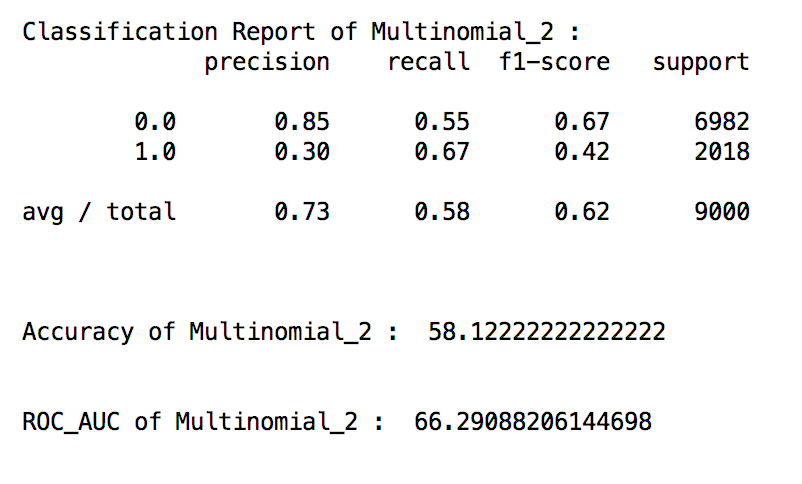


Figure 6

Confusion matrix and classification report of Gaussian is shown in Figure 7. According to Figure 7, the accuracy is about 20 percent higher (77.8%) than using multinomial function, even though there are only 6 class 0 inputs in test data that have been classified to the right class.

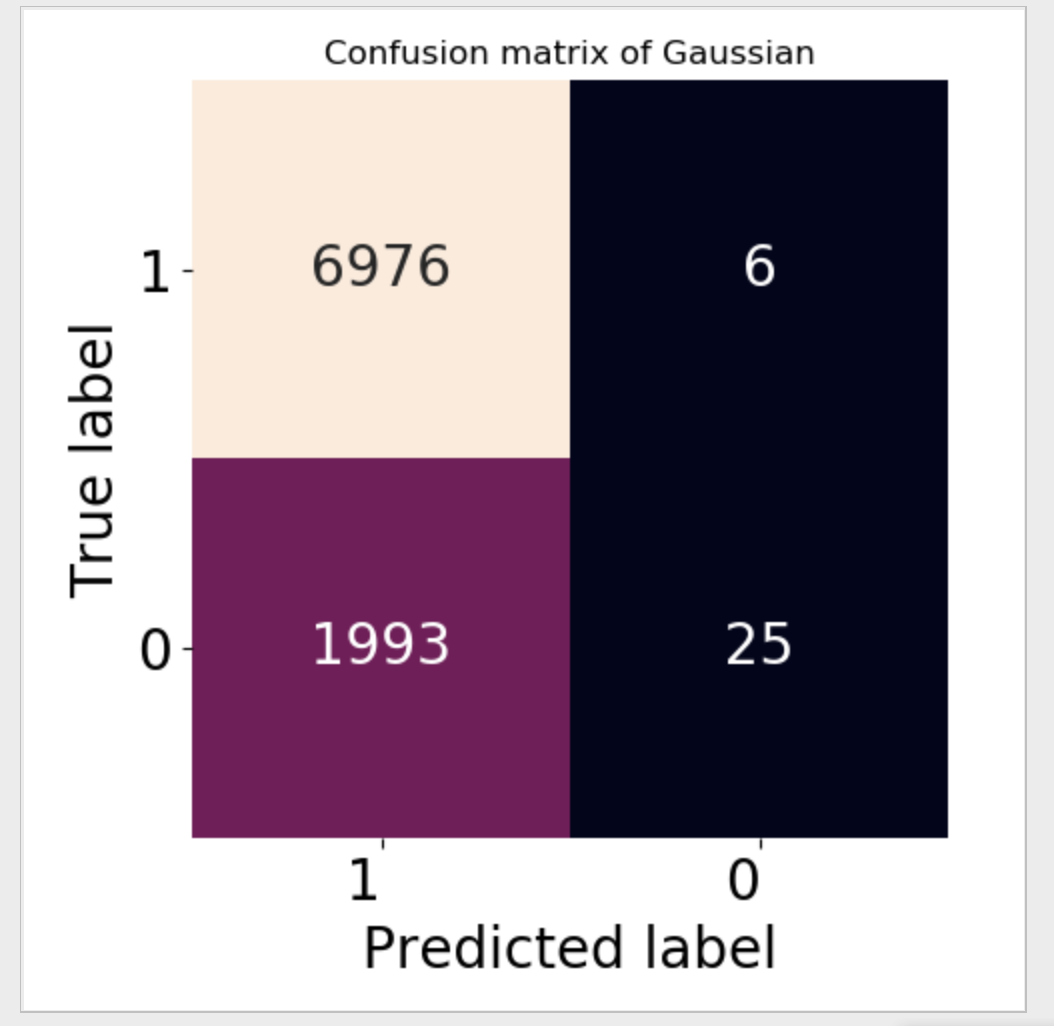
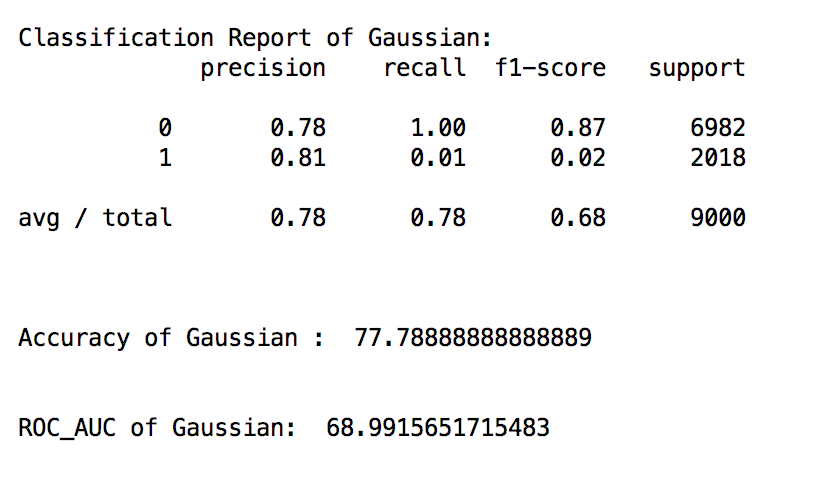


Figure 7

The objective of the project was to find a best model to predict whether the credit card user will have a default payment or not. As mentioned above, even though the accuracy in Gaussian function was higher than the other one, the accuracy for class I was only 1%, which means this model is not able to predict the default payment.

This result may be explained by the limitation in Naïve Bayes that accuracy can be reduced if applied to large amounts of data or if the correlations among features are strong. Also, the types of the feature are various, it is hard to train if we only use the algorithm of Naïve Bayes before a bunch of complex data preparation work done. (may move to summary part)   
 Therefore, Naïve Bayes is not an appropriate method to apply in this dataset.

5.4 SVM

6. Conclusion

7. Reference

8. Appendix

1. [Default of credit card clients Data Set](https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients)