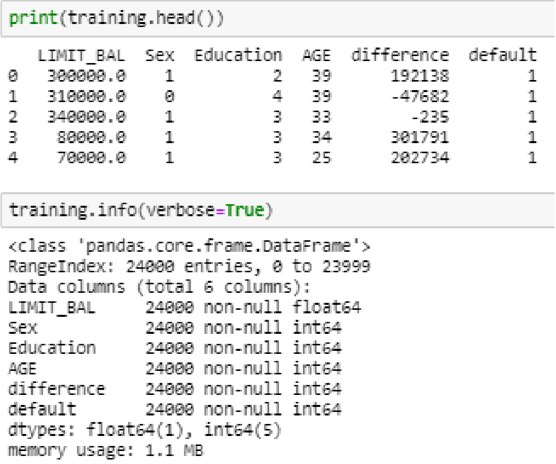
Team : Data Legend

Abstract

This an assignment on the foundation of credit scoring development. Credit scoring is the term used to describe formal statistical methods used for classifying applicants for credit into "good" and "bad" (skip payments) risk classes. Such methods have become increasingly important with the dramatic growth in consumer credit recently, in this case, is a consumer bank in Taiwan. This paper is aimed to provide an overview of statistical classification methods applicable to the dataset used in the case. We use Excel functions like Pivot Table to calculate the data and Python (Jupyter Notebook) to analyze and plot the data. For statistical classification we use Bayes(Generative Model) and k-Nearest Neighbor(Discriminative model), and we can calculate which accuracy is better.

Chapter I – determining the variables

First of all, what we need to do is cleaning the data, by observing if there's an empty cell, or error data input.



```
print(training.head())
```

```
   LIMIT_BAL  Sex  Education  AGE  difference  default
0   300000.0    1          2   39      192138        1
1   310000.0    0          4   39      -47682        1
2   340000.0    1          3   33        -235        1
3    80000.0    1          3   34      301791        1
4    70000.0    1          3   25      202734        1
```

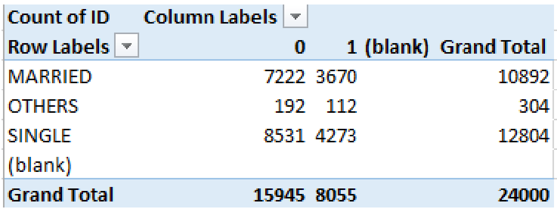
```
training.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24000 entries, 0 to 23999
Data columns (total 6 columns):
LIMIT_BAL     24000 non-null float64
Sex           24000 non-null int64
Education     24000 non-null int64
AGE           24000 non-null int64
difference    24000 non-null int64
default       24000 non-null int64
dtypes: float64(1), int64(5)
memory usage: 1.1 MB
```

Training.info gives us information about the data types, columns, null value counts, memory usage etc. then we can conclude that there's no missing or error data, so we can continue to the next step.

Next step, we need to determine the variables which are significant to the statistical classification. We will provide step by step explanation on this procedure. We use Chi Square to determine which categorical variables are significant.

1. We use PivotTable in Excel to map out the data for each variable. We provides an example below:



Count of ID	Column Labels			
Row Labels	0	1	(blank)	Grand Total
MARRIED	7222	3670		10892
OTHERS	192	112		304
SINGLE	8531	4273		12804
(blank)				
Grand Total	15945	8055		24000

2. In the Excel we apply Chi Square by using =CHITEST(observed_data;expected_data) for each variable. A Chi square test for independence compares 2 variables in a contingency table to see if they are related. It tests to see whether distributions of categorical variables differ from each another. The Formula used in chi square stats is :

$$\chi^2_c = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where the subscript "c" are the degrees of freedom. "O" is our observed value and E is our expected value.

- A large chi square test statistic means that most probably the relationship is by chance only.
- A small chi square test statistic means that your observed data fits your expected data extremely well. In short, there is a valid relationship.

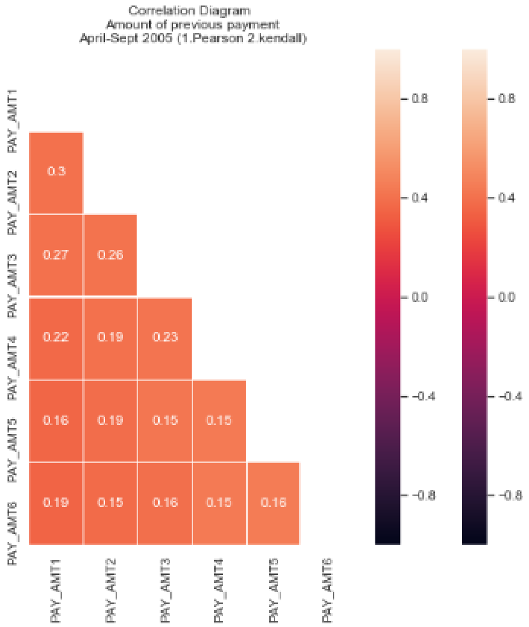
3. From the Chi Square test we found that some variables were significantly different as indicated by its low probability of chance (<0.5%).

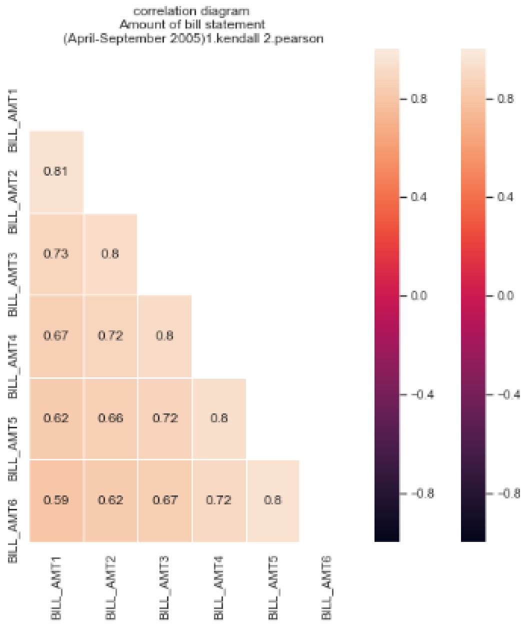
4. We found that these categorical variables are significant: SEX, EDUCATION and AGE. We found out that the variable MARRIAGE is not significant with chi-square stats reaches 41%.

VARIABLES	CHISTATS
gender	0.00011586
education	2.8938E-25
age	1.4758E-10
marriage	0.41499261

5. For the AGE variables we use the following range: <30, 30-40, 41-55, 55<. This is to adjust to the pension age (55)

6. While for the continuous variables, we develop a new variable, BILL_AMTn – PAY_AMT(N-1). This will test the balance between what a customer paid and what he/she owed. We named it amount difference, and we used the following range: <0, 0-278k, 278-1009k, 1009-1740k, 1740k<..





In Bill Amount Analysis, correlation is decreasing with distance between months. Lowest correlations are between September-April. But there's almost no correlation between amounts of previous payments for April-September 2005.

7. For the k-NN we use all the variables except MARRIAGE, as shown by its Chi Square. With additional status :

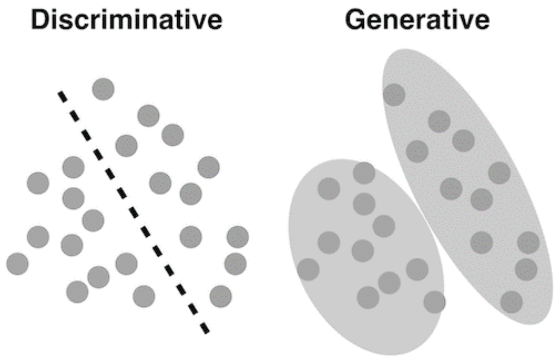
-education: Graduate school(4), University(3), High School(2), others(1)
-sex: Male(0), female(1)

Chapter 2 - determining the methodology

There are 2 main approaches in statistical classification, generative approach and discriminative approach. A generative model learns the joint probability distribution p(x,y). it predicts the conditional probability with the help of Bayes theorem. A discriminative model learns the conditional probability distribution p(y|x). both of these models were generally used in supervised learning problems.

Generative classifiers
.Assume some functional form for p(Y), P(X|Y)
.Estimate parameters of P(X|Y), P(Y) directly from training data
.Use Bayes Rule to calculate P(Y|X)
E.g. Naïve Bayes, Bayesian Network, Hidden Markov models, Markov random fields

Discriminative Classifiers
.Assume some functional form for P(X|Y)
.Estimate parameters of P(Y|X) directly from training data e.g. logistic regression, Nearest neighbor, traditional neural networks, support vector machine



In this assignment, we're using both methodology. We're using both methodology to better compare, aiming for more precise result. For the generative one, we use Naïve Bayes, and for the discriminative one, we're using the k-Nearest-Neighbor Method.

I. Naïve Bayes Classifiers

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

The **Naïve Bayes** algorithm gives us a way of combining the prior probability and conditional probabilities in a single formula, which we can use to calculate the probability of each of the possible classifications in turn. Having done this we choose the classification with the largest value.

The Naïve Bayes classification algorithm

Given a set of k mutually exclusive and exhaustive classifications c1, c2,..,c_k, which have prior probabilities P(c1), P(c2),…,P(c_k), respectively and n attributes a1, a2,…a_n Which for a given instance have values v1, v2,…v_n respectively, the posterior probability of class c_i occurring for the specified instance can be shown to be proportional to:

P(Ci) x P(a1=v1 and a2=v2.. and a_n=v_n | Ci)

Making the assumption that the attributes are independent, the value of this expression can be calculated using the product

P(Ci) x **P**(a1=v1 | Ci) x **P**(a2=v2 | Ci) x … x **P**(a_n=v_n | Ci)

We calculate this product for each value of *i* from 1 to *k* and choose the classification that has the largest value. The formula shown above combines the prior probability of Ci with the values of the n possible conditional probabilities involving a test on the value of a single attribute.

It is often written as

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

II. k-Nearest-Neighbor Classifiers

According to D.J. Hand, W.E. Henley, in the *Journal of the Royal Statistical Society*, (1997) Developed an adaptive metric nearest neighbor method (with a parameter D describing the shape of the metric) for credit scoring. The result shown in table below.

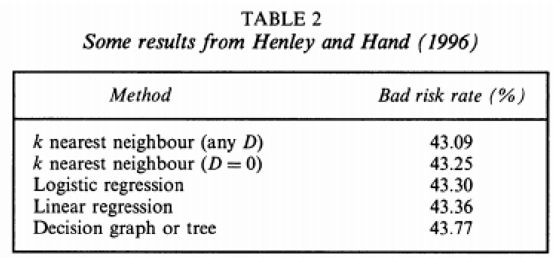


TABLE 2
Some results from Henley and Hand (1996)

Method	*Bad risk rate (%)*
k nearest neighbour (any *D*)	43.09
k nearest neighbour ($D=0$)	43.25
Logistic regression	43.30
Linear regression	43.36
Decision graph or tree	43.77

These figures are based on the test set samples of about 5000 with acceptance rates of 70%. The nearest neighbor are superior in this comparison. Thus, we choose the kNN for our credit scoring method.

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

kNN Algorithm

Basic kNN Classification Algorithm

- Find the k training instances that are closest to the unseen instance.
- Take the most commonly occurring classification for these k instances.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Distance functions in kNN:
(1)Euclidean

$$\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

(2)Manhattan

$$\sum_{i=1}^{k} |x_i - y_i|$$

(3) Minkowski

$$\left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q}$$

However, we must concern the fact that all three distance measures are only valid for continuous variables. The Hamming distance need to be used in the instance of categorical variables. Hamming Distance brings up the issue of standardization of the examples between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

Hamming Distance :

$$D_H = \sum_{i=1}^{k} \left| x_i - y_i \right|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

In kNN method, the parameter we used is k value. Higher value of k tend to reduce the noise effect in classification. However it will make the classification boundaries more unclear. Cross-validation is another way to retrospectively determine a K value by using an independent dataset to validate the K value.

From the sklearn.neighbors library in python, we use kNeighborsClassifier to determine the optimal value of K.

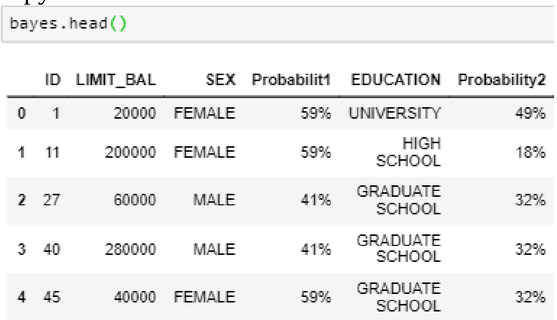
Chapter III – Data Exploration and Analysis

A. BAYES THEOREM

The default probability of every variable:

Category	Varian	Default Prob
Sex	Male	35%
Sex	Female	33%
Education	Graduate School	31%
Education	HIGH SCHOOL	38%
Education	UNIVERSITY	17%
Education	OTHERS	35%
Age	<30	36%
Age	30-40	31%
Age	41-55	34%
Age	55<	35%
Amount(B-P)	<0	17%
Amount(B-P)	0k-278k	38%
Amount(B-P)	278k-1009k	26%
Amount(B-P)	1009k-1740k	31%
Amount(B-P)	1740k<	30%

Jupyter notebook overview:



```
bayes.head()
```

	ID	LIMIT_BAL	SEX	Probabilit1	EDUCATION	Probability2
0	1	20000	FEMALE	59%	UNIVERSITY	49%
1	11	200000	FEMALE	59%	HIGH SCHOOL	18%
2	27	60000	MALE	41%	GRADUATE SCHOOL	32%
3	40	280000	MALE	41%	GRADUATE SCHOOL	32%
4	45	40000	FEMALE	59%	GRADUATE SCHOOL	32%

Using the Naïve Bayes Algorithm,
We can calculate the probability of default for a particular instance.

For Example :
Someone has a category of: Male (Sex), University (Education), 32 (Age), and 200k for Bill-Pay Amount.

Using the values in each of the columns of table above, we obtain the following posterior probabilities for each possible classification for the unseen instance:

Male - University - (30-40) - (0-278k)
Default = 1

35% x 17% x 31% x 38% = 0.0070091 (0.7%)

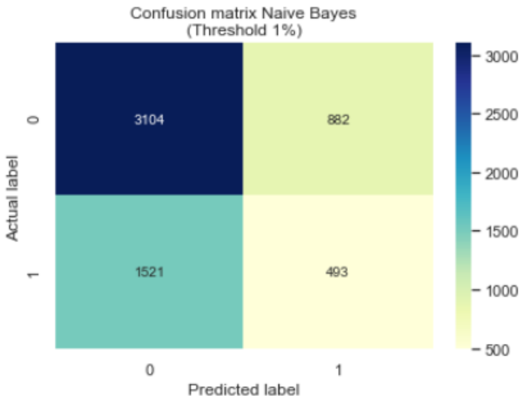
In this case, we use 1% probability as a threshold default rate. Most of the banks used 3% as a standard default rate, however in this case, we decided to use a more conservative figure (1%) after several trial, to see which one is the most accurate and plausible.

Because that person has only a 0.7% probability to become default, and is lower than threshold default rate (1%). Therefore we can assume that this person is not going to skip payment. (Default =1)
When we apply this to 6000 data we found out that it is sufficiently accurate.

Model Performance Analysis

1. Confusion Matrix
The confusion matrix is a technique used for summarizing the performance of a

classification algorithm i.e. it has binary outputs.



In this case:

Case in which the person is predicted default (1) and they do default will be termed as TRUE POSITIVE(TP).

Case in which the person is predicted not default (0) but they do default will be termed as TRUE NEGATIVE(TN).

Case in which the person is predicted default (1) but they do not default will be termed as FALSE POSITIVE(FP). Also known as “Type I error.”

Case in which the person is predicted not default (0) but they do default will be termed as FALSE NEGATIVE (FN). Also known as “Type II error.”

Classification Report

Report which includes Precision, Recall and F1-Score.

- *Precision Score*

TP – True Positives
FP – False Positives
Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Precision = TP/TP+FP

- *Recall Score*

FN – False Negatives

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? A recall greater than 0.5 is good.

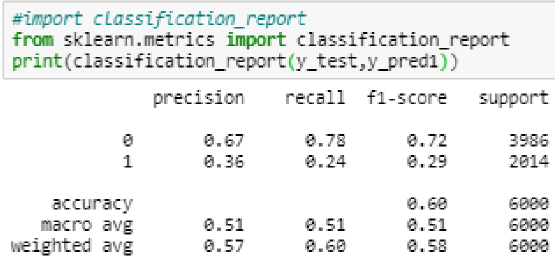
Recall = TP/TP+FN

- *F1-Score*

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F1 Score = 2(Recall Precision) / (Recall + Precision)

Below are the classification report for method I, Bayes Theorem. Using sklearn.metrics in python library.



```
#import classification_report
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred1))
```

```
              precision    recall  f1-score   support

           0       0.67      0.78      0.72      3986
           1       0.36      0.24      0.29      2014

    accuracy                           0.60      6000
   macro avg       0.51      0.51      0.51      6000
weighted avg       0.57      0.60      0.58      6000
```

B. K-Nearest-Neighbour

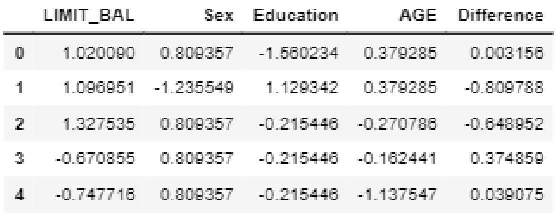
In this case were going to use the Euclidean distance (by far the most common and accurate) as explained before in chapter II, and the algorithm is:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

First, we include all the variable (Limit Balance, Age, Sex, education, and Bill-Pay amount difference. And scaling the data, through this formula:

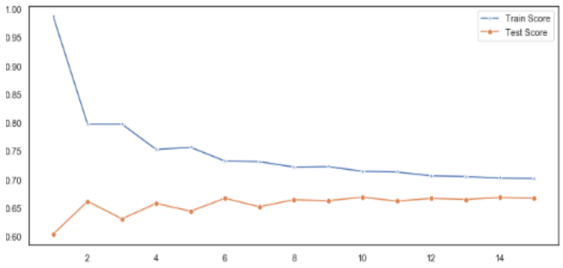
$$z = \frac{x_i - \mu}{\sigma}$$

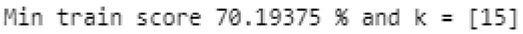
data Z is rescaled such that μ = 0 and **σ** = 1, and is done through the formula above. By using *StandardScaler*, imported from *sklearn.preprocessing* library in python. And the results are :



	LIMIT_BAL	Sex	Education	AGE	Difference
0	1.020090	0.809357	-1.560234	0.379285	0.003156
1	1.096951	-1.235549	1.129342	0.379285	-0.809788
2	1.327535	0.809357	-0.215446	-0.270786	-0.648952
3	-0.670855	0.809357	-0.215446	-0.162441	0.374859
4	-0.747716	0.809357	-0.215446	-1.137547	0.039075

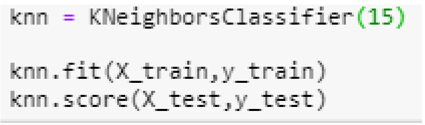
Then to find the value of k, we tried to analyze every value of k from range 1 to 16. We applied kNeighborsClassifier module imported from sklearn.neighbors in python library to see which one has the shortest difference between the training and testing data set.





```
Min train score 70.19375 % and k = [15]
```

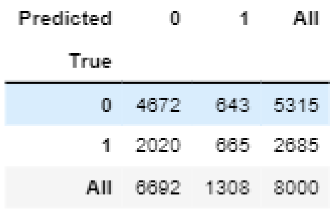
The best result is captured at k = 15, hence 15 is used for the final model. Then we setup a k-NN classifier with k neighbors:



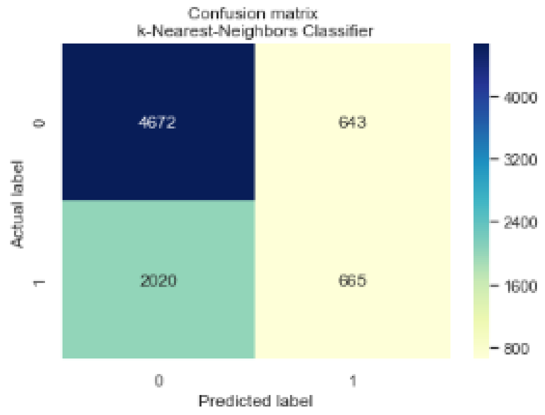
```
knn = KNeighborsClassifier(15)

knn.fit(X_train,y_train)
knn.score(X_test,y_test)
```

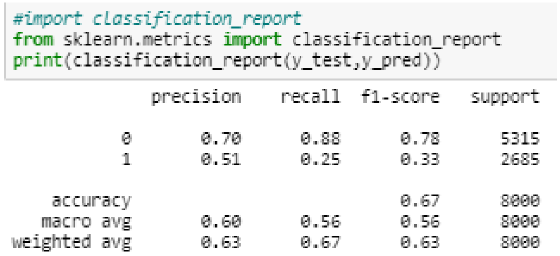
And then we made the confusion matrix of the prediction result in our k-NN method, using the same way in the Naïve Bayes method



Predicted	0	1	All
True			
0	4672	643	5315
1	2020	665	2685
All	6692	1308	8000



The above graph is the confusion matrix of k-NN method with the same axis as the Bayes method and the classification report is provided below:



```
#import classification_report
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.88      0.78      5315
           1       0.51      0.25      0.33      2685

    accuracy                           0.67      8000
   macro avg       0.60      0.56      0.56      8000
weighted avg       0.63      0.67      0.63      8000
```

Chapter IV – Conclusion and Suggestion

We compare the accuracy rate of both methods based on the classification report above.

For Bayes (weighted average):

	Bayes	kNN
Precision	57%	63%
Recall	60%	67%
F1-Score	58%	63%
Final Accuracy	60%	67%

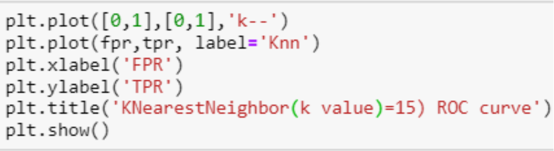
From there we can see that kNN method has a higher accuracy rate in which we can assume that in this case, kNN is slightly better at making credit scoring than Naïve Bayes.

Thus, we are going to make ROC – AUC of kNN method for better result seeing

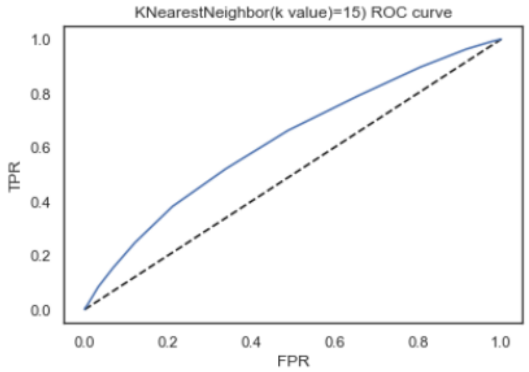
ROC – AUC

ROC (Receiver Operating Characteristic) Curve tells us about how good the model can distinguish between two things (e.g. If a client is going to skip payment or not). Better models can accurately distinguish between the two. Whereas, a poor model will have difficulties in distinguishing between the two

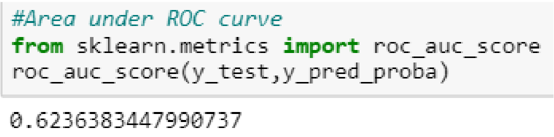
We can make the ROC by importing roc_curve from *sklearn.metrics* in python library. And plot it in the Jupyter Notebook



```
plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr, label='Knn')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('KNearestNeighbor(k value)=15) ROC curve')
plt.show()
```



And we can count the roc_auc_score by calculating the area under the curve. Lower score indicates poor classification model. The result (62%) is quite satisfactory.



```
#Area under ROC curve
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,y_pred_proba)
```

```
0.6236383447990737
```

Suggestion

We conclude that k Nearest Neighbor (k-NN) is a more suitable method for classification on the probability of default by using the given dataset, as compared to Bayes. This assignment is interesting as it requires a combination of analysis of categorical and continuous variables. Bayes is mostly used for categorical variables and kNN is for continuous one.

Further studies should be conducted with other algorithm like logistic regression or neural network to test and learn, however, k-NN is much easier to explain to the business community.

The weakness of the k-NN method is the huge consumption of computational resources. However, with the advancing technology and lower cost of hardware, this should be more and more manageable.

The analysis should also be more statistically meaningful and comprehensive if other personal variables are included, like: length of stay in current address, residence status (owned, credit or rent), number of dependents in the family, other debts owed, number of credit cards owned, length of stay in current job, type of occupation, monthly income, postal code, purpose of loan, length of time with the bank.

The relevant Python codes, computations, modules and graphs are attached in a separate file, available upon request.

REFERENCES

Bramer, Max, Principles of Data Mining, Springer, London, UK, 2007

Cios, et al., Data Mining – A Knowledge Discovery Approach, Springer, New York, USA, 2007

Hand and Henley, Statistical Classification Methods in Consumer Credit Scoring: A Review, Journal of the Royal Statistical Society, Series A, Volume 160, Issue 3, 1997

Hair, et al., Multivariate Data Analysis, Prentice Hall, New Jersey, USA, 4th edition, 1995

Mays, Elizabeth, Credit Scoring for Risk Managers: The Handbook for Lenders, Thomson Learning, Ohio, USA, 2004