# Default of Credit Card Clients in Taiwan

## Data Mining Project - First semester

**Elyani Tia**, 584584, Data Science and Business Informatics
**El-Shaer Faris**, 303728, Data Science and Business Informatics
**Matteoli Anna**, 491919, Data Science and Business Informatics
**Merendi Federica**, 584572, Data Science and Business Informatics

# Index

# 1. Introduction

The goal of the project is to predict credit card default among the credit card holders in Taiwan. Six data mining methods are applied, and corresponding default probabilities are investigated.

Python was used to complete the project.

# 2. Data understanding

## 2.1 Data Semantics

- Dataset is provided in tabular form composed of 10000 records (rows), each belonging to different client. Records are described by 24 attributes. Four attributes (sex, education, status and age) concern client's personal information, twelve attributes (six pa's and six ba's) describe client's monthly financial transaction amounts, six 'ps' attributes describe status of these transactions, limit and credit_default are client's account features.

- Monthly transactional data refer to a 6-month period of client's account life-cycle (from April to September of 2005), but for credit_default the time of client's default is not specified (it is sometime in the future). So, in this project we are trying to use data from specified time window to predict a possible future event in unspecified time instance.

- Semantic of ps attributes: domain of ps attributes are integers from an interval [-2,9]. Values from -1 to 9 are referring to delay in payment: -1 payment on time (no delay), 0 partial payment (due to allowed revolving credit), 1 delay of 1 month etc. Value -2 has completely different meaning, it means that credit has not been used for the specified month. This indicates that ps's conceptually could be divided in two attributes: one that refers to delay in repayment and another referring to monthly activity/inactivity of credit card account. We will come back to this problem later in this report.

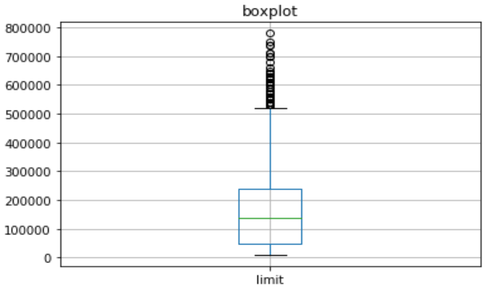
- The goal of the project is to predict probability of credit card default, so the necessary attributes for this task are those describing monthly financial transactions (pa and ba attributes). However, there are 259 records with all pa and ba values equal to zero. This is not unexpected due to a fact that transactional data belongs to a narrow time window of account's life-cycle. So, for some records this time window accidentally corresponded to account's inactivity period. However, these records should be eliminated because they are irrelevant for the analysis. This reduced the initial dataset to 9741 records.

## 2.2 Distribution of the variables and Statistics

### 2.2.1 Numerical attributes

- limit: the domain is composed of positive integer values. IQR is almost symmetric around the mean, the maximum value differs 10-fold from the median respect to the minimum value. Therefore, limit will be densely populated for the small values and sparsely for the high values. This is easily seen on the boxplot.

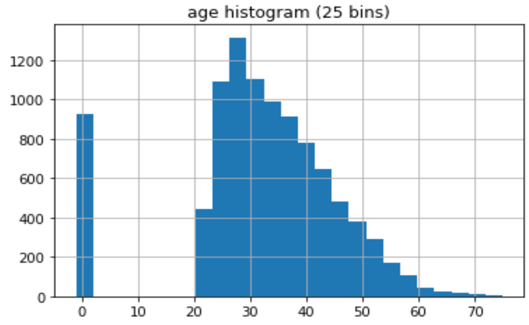
| | **Limit** |
|---|---|
| **count** | 9741.00 |
| **mean** | 166038.39 |
| **std** | 129127.95 |
| **min** | 10000.00 |
| **25%** | 50000.00 |
| **50%** | 140000.00 |
| **75%** | 240000.00 |
| **max** | 780000 |



*1 Table and boxplot for the attribute limit*

- age: the minimum age for cardholder according to Taiwan's law is 20 years. So, the domain for age should be composed of integers greater or equal to 20. But there are also 926 values -1. They can easily be seen as a separate pick on the histogram below. These are anomalous values and will be treated as missing values in the next paragraph. The values different from -1 have truncated bell shape distribution, as expected from the limit for cardholder's age. Statistical parameters for this part of distribution are showed in 'Age without -1' column of the next table. Similarly, as for limit there is many records with small values and much less with high.

| | Age | Age (without -1) |
|---|---|---|
| count | 9741.00 | 8815.00 |
| mean | 31.976594 | 35.440726 |
| std | 13.823026 | 9.214158 |
| min | -1.000000 | 21.000000 |
| 25% | 26.000000 | 28.000000 |
| 50% | 32.000000 | 34.000000 |
| 75% | 40.000000 | 41.000000 |
| max | 75.000000 | 75.000000 |



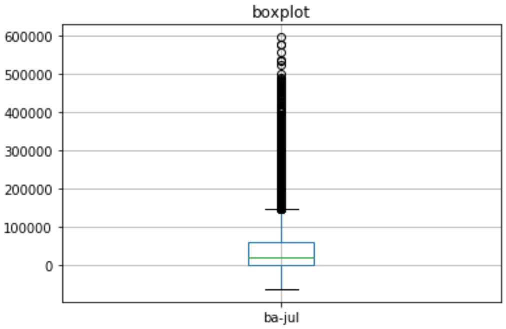
2 *Table and histogram for the attribute 'age'*

- ba attributes: the domains of all the ba attributes contain negative and positive integers. From the attribute describing the bill amount of client's account we primarily expect positive values. However, bill amount statement can be negative in some credit cards (depending on client's contract details) indicating excess of past payments. Due to this possibility we decided to consider all negative values as valid. In the table below number of records with negative ba values are indicated. All ba attributes are very sparse for high values and very dense for small values. We are showing statistical properties and boxplot for 'ba-jul', but similar situation holds for the rest of ba attributes.

| ba-apr | ba-may | ba-jun | ba-jul | ba-aug | ba-sep |
|---|---|---|---|---|---|
| 225 | 207 | 218 | 222 | 220 | 187 |

*Table 1 Records with negative value for each month*

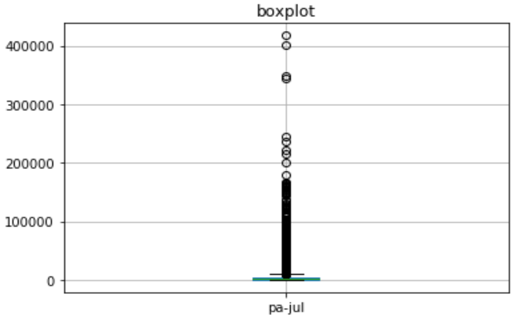
| | ba-jul |
|---|---|
| count | 9741.00000 |
| Mean | 48206.00225 |
| Std | 69427.17889 |
| Min | -61506.00000 |
| 25% | 3273.00000 |
| 50% | 21000.00000 |
| 75% | 61879.00000 |
| Max | 597415.00000 |



3 *Table and boxplot for the attribute 'ba-jul'*

- pa attributes: the domains of all pa attributes are non-negative integers, as expected according to the attributes description. All pa's are extremely sparse for high values and extremely dense for small values (these properties are even more pronounced respect to ba attributes). We report statistical data and boxplot for 'pa-jul'. The IRQ and the 1.5 IRQ range that can be seen on the boxplot are so narrow that are almost indistinguishable from a line. Similar is true for all others pa values.

| | pa-jul |
|---|---|
| count | 9741.00000 |
| Mean | 5268.348835 |
| Std | 15596.999422 |
| Min | 0.000000 |
| 25% | 514.000000 |
| 50% | 1980.000000 |
| 75% | 4685.000000 |
| Max | 417588.000000 |



4 *Table and boxplot for the attribute 'pa-jul'*

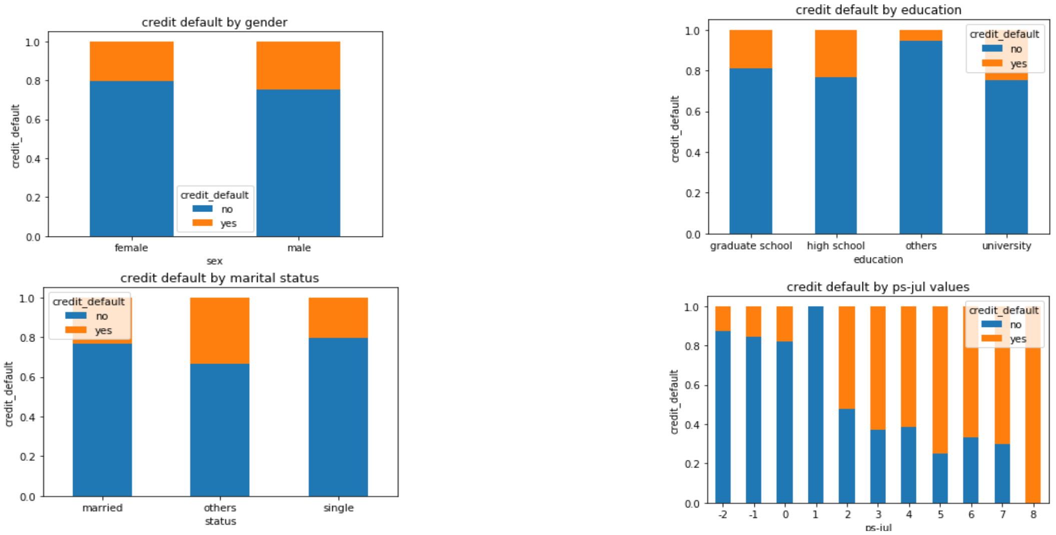
### 2.2.2 Categorical attributes

- sex: Binary attribute with 96 (9.9%) missing values. There are 61% of Females and 39% of Males among non-missing values.
- education: 4-value attribute with 121 (1.24%) missing vales. The distribution among non-missing values is the following: 47.8% University, 34.8% Graduate School, 17% High School and 0.4% Others.
- status: 3-value attribute with 1778 (18.19%) missing values. The distribution among non-missing values is following: Single 53.3%, Married 45.8% and Others 0.9%
- ps attributes: Six 10-value attributes without missing values. The table below summarizes their distributions in percentages. It is clear that revolving credit represents the most frequent way of repayment (more than 50% of transaction for each month). As a delay in repayment increases the number of transactions diminish as is expected from data describing economy out of greater turmoils (war, huge economic crisis ecc.). The row with value -2 is marked in different color in order to conceptually distinguish it from other values. Notice that there is a consistent amount of inactive accounts for each month (from 8-14%).

| | ps-apr(%) | ps-may(%) | ps-jun(%) | ps-jul(%) | ps-aug(%) | ps-sep(%) |
|---|---|---|---|---|---|---|
| **0** | 55.949081 | 57.766143 | 56.452110 | 54.234678 | 53.8035 | 50.076994 |
| **-1** | 19.659173 | 18.858433 | 19.597577 | 20.090340 | 20.3675 | 19.238271 |
| **-2** | 14.136126 | 13.068473 | 12.442254 | 11.579920 | 10.3685 | 8.376963 |
| **2** | 9.239298 | 9.116107 | 10.255621 | 12.822092 | 13.705 | 9.434350 |
| **3** | 0.585156 | 0.595421 | 0.667283 | 0.749410 | 1.10872 | 1.139513 |
| **7** | 0.195052 | 0.236115 | 0.236115 | 0.102659 | 0.0821271 | 0.020532 |
| **4** | 0.143722 | 0.277179 | 0.205318 | 0.266913 | 0.287445 | 0.236115 |
| **6** | 0.051329 | 0.020532 | 0.010266 | 0.092393 | 0.0307977 | 0.030798 |
| **5** | 0.030798 | 0.051329 | 0.123191 | 0.041064 | 0.102659 | 0.112925 |
| **8** | 0.010266 | 0.010266 | 0.010266 | 0.010266 | 0.0000 | 0.082127 |

*Table 2 Distribution of the ps values in percentages.*

- credit_default: Binary attribute without missing values with following distribution: No 78% and Yes 22%. This distribution is very unbalanced in favor of records with non-defaulted clients. Credit_default is a class attribute in this project, so we give a cross attribute plots between credit_default and every other categorical attribute. More precisely we took one attribute, than divided initial set to subsets corresponding to different attribute values and calculated credit_default distribution on each subset. For Gender, Education and Status all obtained subsets have credit_default distribution with mode No represented by 75-94%, so very similar to the initial dataset distribution. However, behavior of ps's differ in this aspect. Namely higher delay in repayment higher probability of default. For repayment delay of 2 months or more mode becomes Yes indicating that credit default is most probable outcome, eg. for delay of 8 months there is an average probability of 93% of credit default. Here we report only plot for 'ps-jul' but the same trend holds for all other ps's.



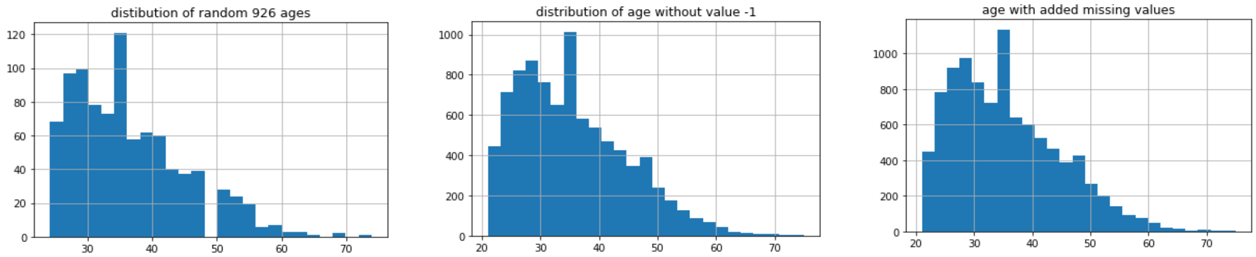
*5 credit_default distribution among the categorical values*

## 2.3 Assessing Data Quality (Missing Values, Outliers) and correlations

To decide which missing value strategy to apply for Sex and Education we investigated cross attribute plots for each of them, similarly we did for Credit_default. The mode of these attributes on the whole dataset remains the same on each subset. This together with the fact that small number of records for Sex and Education are missing implies that mode can be used for missing value substitution, without adding artificial peaks in the distribution. Mode for Sex is Female and for Education University.

Status attribute has many missing values. To disturb initial distribution at least possible, we generated random values according to Status distribution without missing values. Then we assigned these values randomly to records with missing values. The distribution remains almost the same: Single 53.1%, Married 45.9% and Others 0.9%.

In the case of Age there are also many missing values. Like in Status here we generated random values with respect to the distribution without missing values. First step was to discretize Age. We have selected 25 equally spaced bins among different binnings choosing one with the best qualitative shape of the histogram. Everything said is summarized in following plots:



*6 Distribution of age*

For local outliers of numerical attributes, we considered points outside the [Q1-1.5*IQR, Q3+1.5*IQR] interval.

| attribute | limit | Age | ba-sep | ba-aug | ba-jul | ba-jun | ba-may | ba-apr | pa-sep | pa-aug | pa-jul | pa-jun | pa-may | pa-apr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of local outliers | 50 | 82 | 780 | 757 | 795 | 843 | 903 | 886 | 889 | 917 | 846 | 959 | 951 | 1005 |

*Table 3 Number of local outliers for numerical attributes.*

Individual records have from zero to 14 local outliers. There is one record which has local outlier in all 14 dimensions. Therefore, this point can be considered a global outlier with certainty. But what with records that have fewer local outliers? We tried to find a threshold in a number of dimensions which would distinguish local outliers from global ones. All records with number of local outliers equal or higher to a threshold would be considered global outliers and therefore eliminated from dataset. Number of potential global outliers for different thresholds is given in the table:

| threshold | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of global outliers | 1 | 11 | 77 | 108 | 151 | 195 | 270 | 411 | 719 | 897 | 1128 | 1479 | 2100 |

*Table 4 Number of global outliers for different thresholds*

Eg. if threshold is 8 we have 270 records with 8 or more local outliers.

Next, for every threshold we eliminated potential global outliers and calculated correlation matrix on the resulting sets. It can be observed that for highly correlated attributes (correlation >=0.9) decreasing the threshold also decrease correlations. In the same time this increases the number of potential global outliers. We tried to choose the optimal threshold removing the smallest possible number of records while conserving maximum number of attribute correlations over 0.9. The optimal value of threshold obtained in this way is 8. So, we eliminated additional 270 records now considered to be global outliers.

There are 6 pairs of attributes with correlations over 0.9. The table below summarizes these results:

| attribute pair | ba-sep ba-aug | ba-aug ba jul | ba-jul ba-jun | ba-jun ba-may | ba-may ba-apr |
|---|---|---|---|---|---|
| correlation coefficient | 0.927 | 0.913 | 0.903 | 0.909 | 0.920 |

*Table 5 Pairs of attributes with a correlation coefficent > 0.9*

From every pair of correlated attributes only one can be taken for further analisys (another is considered redundant). In this way different sets of possible attribute combinations can be formed depending which attributes is taken from each pair. We selected one of the sets with minimal cardinality (which is 3) in order to eliminate maximal number of redundant attributes. We were left with: ba-may, ba-jul and ba-aug.

## 2.4 Variable Transformations

Transformation of ps attributes: as already mentioned values of ps attributes have two logically distinct meanings. Values -1 to 8 referring to delay in monthly repayments, and value -2 to activity/inactivity of account. These attributes can be transformed in measure of client's repayment during the 6-months time period. It is convenient to remap the values in more indicative way: -1 to 0, 0 to 0.5 and leaving other values intact. In this way 0 signifies no delay in repayment, 1...8 repayment with 1...8 months of delay and 0.5 partial repayment due to credit use. Then average over values different from -2 should be made. Rounding applied is slightly different from usual algebraic rules:

- [0,0.34) --> 0 meaning: average less than one third of a month is considered as no delay
- [0.34,0.67) -->0.5 meaning: average repayment delay between 1/3 and 2/3 is considered as partial repayment.

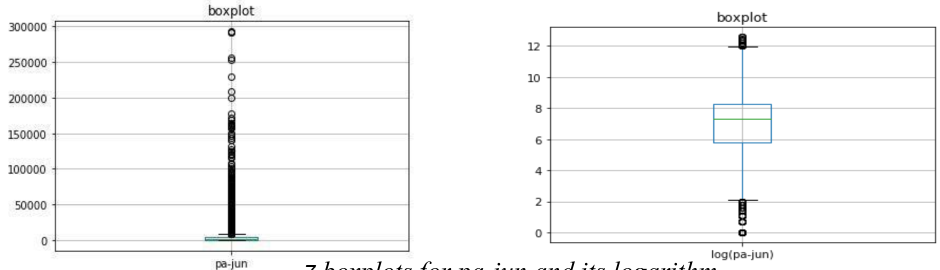
For other values the rounding is usual.
Example: consider a record with ps values 0, -2, 0, -1, -1, 2 --> first: eliminate -2: 0, 0, -1, -1, 2 --> second: remap values --> 0.5, 0.5, 0, 0, 2 -> third: calculate average --> 0.6 -->fifth: round --> 0.5.
This new attribute is called ps-avg.
In the initial dataset there are 703 records with all ps values -2. These are inactive accounts for 6 moths time period and therefore should have all ba and pa values zero (all transactional amounts zero). At the very beginning we eliminated 259 records with zero transactional amounts. But only 101 records among them had all ps values -2. So, we are left with 703 - 102 = 602 records with all ps values -2, and ba and pa values different from zero. This is inconsistent and there must be some error in process of data generation. As financial transactions are among the highly secured and checked human activities on Earth, it is almost certain that errors are among ps values.
To calculate average repayment delay (ps-avg) we should have at least one month of account's activity. These rows are inactive for whole 6-moths period and ps-avg cannot be calculated. So, these 602 values have been considered missing values in ps-avg column.
Statistical distribution of this ps-avg attribute is following: 0.5 47.3%, 0 25.8%, 1 18.3%, 2 7.5%, 3 0.6%, 5 0.2% 4 0.2% 6 0.1%
As in case of Status, we randomly generated and assigned 602 values according to distribution of 'ps-avg' without missing values.
Performance of data mining algorithms improves as number of numerical attributes diminish. As we already have 14 numerical attributes, we decided to transform age in categorical one by discretization. Binning intervals we used are: [20, 29) --> 2, [30, 39) --> 3, [40, 49) --> 4, [50, 59) --> 5, [60, 69) --> 6, [70, 79) --> 7
The new attribute is called age-discrete and has the following statistical distribution: 3 37.4%, 2 32.5%, 4 21.3%, 5 7.8%, 6 0.9%, 7 0.1%.

### 2.4.1 Logarithmic transformation

As already mentioned, ba and pa attributes have highly sparse distributions. After application of the logarithmic transformation on sparse distribution it becomes less sparse. So, we consider also logarithms instead of just plain ba and pa values. Logarithmic function is not defined for negative values, but all ba attributes have negative values. So, before the ln is applied translation of ba's should be performed. For values in the interval (0,1) log is negative. For the sake of convenience, we translated the values of ba's and pa's to non-negative numbers. So, we applied the following transformation to all ba's and pa's:

$f(x) = \ln(x + |min(attribute)| + 1)$ , where x is the element of attributes domain.

In this way the minimum of each attribute is mapped to 0.
Two following boxplots clearly show the effect of the transformation on the distribution of attributes.

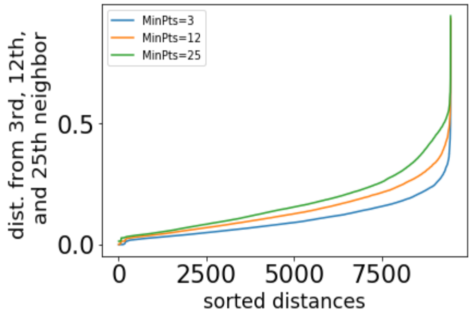


7 boxplots for pa-jun and its logarithm

# 3. Clustering

## 3.1 DBSCAN algorithm

DBSCAN algorithm is based on notion of n-dimensional density which requires two parameters, called Epsilon and MinPts. Changing their values, the quality of clustering result changes. For values of MinPts from 3 to 25 we plotted the 'MinPts-th nearest neighbor distances'. Investigating area around the knee of the function Epsilon for the best clustering should be found. Plot for each successive value of MinPts was lifted more respect to value before, as shown on the next plot:



*8 Sorted distances for differents neighbor*

We fitted every plot to polynomial of grade 15, and then analytically calculated the knee of each function, as the value of the highest change in derivative. For the measure of the best clustering initially we considered only the Silhouette coefficient. Silhouette coefficient was calculated after eliminating noise points determined by DBSCAN. Investigating interval around the knee we obtained the following maximal values of Sillhoutte coefficients:

| Silhouette | 0.192 | 0.377 | 0.341 | 0.397 | 0.536 | 0.537 | 0.523 | 0.532 | 0.527 | 0.525 | 0.521 | 0.519 | 0.516 | 0.513 | 0.501 | 0.507 | 0.503 | 0.490 | 0.488 | 0.451 | 0.448 | 0.447 | 0.418 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MinPts--Eps | 3--0.218 | 4--0.175 | 5--0.130 | 6--0.135 | 7--0.140 | 8--0.143 | 9--0.147 | 10--0.150 | 11--0.154 | 12--0.158 | 13--0.161 | 14--0.166 | 15--0.170 | 16--0.175 | 17--0.181 | 18--0.186 | 19--0.192 | 20--0.197 | 21--0.200 | 22--0.204 | 23--0.201 | 24--0.211 | 25--0.213 |

From this table the highest Silhouette coefficient is 0.537, for MinPts = 8 and Epsilon = 0.143. The number of noise points in every cluster is following:
[2355, 4147, 292, 224, 221, 214, 190, 172, 167, 140, 133, 121, 109, 77, 62, 52, 51, 51, 50, 49, 42, 40, 40, 39, 38, 38, 35, 32, 31, 30, 27, 25, 22, 22, 19, 18, 17, 17, 14, 12, 10, 9, 9, 8]
The number of noise points is represented by the first number. There are 25% of records labeled as noise, which is pretty high percentage. Other unfortunate feature of this clustering is high number of clusters: 43 after elimination of noise points. The third problem is that clustering contains one large cluster with 44% of points, and others small or very small (statistically insignificant).

Looking at the table there is not much variation in maximal Silhouette coefficient for MinPts from 7 to 21. What we tried next was to vary Epsilon in order to diminish number of clusters and number of noise points, simultaneously leaving Silhouette coefficient over 0.5. Number of noise points was inversely proportional to Epsilon as expected. Number of clusters was high around Epsilon with maximum Silhouette. Diminishing Epsilon toward both ends decrease the number of clusters, but also worsening the Silhouette.
The best result we obtained from these efforts is:
MinPts = 19, Epsilon = 0.194, No. Of noise points: 1646 (17.3%), No. Of clusters: 34
The clustering composition without noise points is:

| cluster No. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| number of points in cluster | 4511 | 350 | 269 | 264 | 234 | 224 | 214 | 211 | 151 | 143 | 128 | 123 | 95 | 70 | 61 | 60 | 59 | 58 | 54 | 51 | 48 | 46 | 45 | 44 | 43 | 40 | 39 | 33 | 31 | 30 | 29 | 24 | 22 | 21 |

Unfortunately, we were not able to much improve the relative distribution of points across the clusters.

- We run DBSCAN with two other metrics: L1 called 'cityblock' and L∞ called 'chebyshev'. Both metrics were worse respect to L2 clustering by increasing the number of clusters or number of noise points. So, we decided that for this task Euclidean metric is the most appropriate for DBSCAN.

- Our choice of the best clustering was the relative one because it was selected among different clusterings with high Silhouette. Now we want to verify that selected clustering has interesting structure also absolutely speaking. For this purpose, we made comparison between Silhouette coefficient of two datasets: one on which we have run DBSCAN and the other randomly generated. Two datasets have the same attributes and equal number of rows. Random values were generatrd between minimum and maximum of each attribute in the initial dataset. With these random values corresponding random set attributes were populated. Following table makes comparison between Silhouettes of initial and random datasets for different values of MinPts.

| MinPts | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial Silh. | 0.192 | 0.377 | 0.341 | 0.397 | 0.536 | 0.537 | 0.523 | 0.532 | 0.527 | 0.525 | 0.521 | 0.519 | 0.516 | 0.513 | 0.501 | 0.507 | 0.503 | 0.490 | 0.488 | 0.451 | 0.448 | 0.447 | 0.418 |
| Random Silh. | 0.697 | 0.603 | 0.410 | 0.374 | 0.404 | 0.456 | 0.335 | 0.150 | 0.120 | 0.027 | 0.038 | 0.071 | 0.057 | 0.072 | 0.085 | 0.117 | 0.123 | 0.154 | 0.171 | 0.205 | 0.227 | 0.231 | 0.256 |

For MinPts 3,4 and 5 random Silhouette is higher than non-random indicating very poor clustering. After MinPts 6 the non-random clustering starts to improve, and it becomes significantly better after MinPts 10. After MinPts 20 clustering starts to deteriorate again.
In the case of selected optimal clustering, MinPts = 19, Silhouette is 4 times higher than random one indicating that the data have clustering structure which is not due to chance.

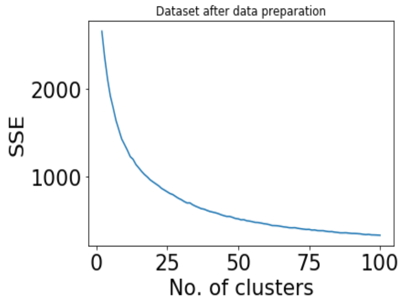
### 3.1.1 DBSCAN clustering characterization

As DBSCAN is density-based algorithm, clusters of non-globular shapes can be found. Therefore, the notion of average for numerical attribute loose meaning because it can be located outside the cluster. So, for DBSCAN we did not considered numerical attributes for cluster characterization.

- In cluster No. 0 (the most dominant one) 72% of records have avg-ps value 0.5 i.e. some delay in repayment, and credit default Yes dominates with 87%. Both these numbers are significantly higher respect to those found in initial dataset, 48% and 78% respectively. Distribution of other categorical attributes is very similar to that of the initial dataset.
- The clustering contains 8 clusters with credit default Yes majority all of which are very small (under 0.9% of records) making them alone not statistically significant. These are clusters number: 13,18,21,23,25,28,31 and 33. As credit default Yes is of interest for this project, we tried to merge these small clusters into bigger one, containing 4.2% of records. After the merge we are remained with 27 clusters. Recalculated Silhouette for this new clustering was 0.459 making it still much bigger respect to random value. Merged cluster has 55% of records with credit default Yes, and 81% of records with ps-avg 1 or 2. Distribution of all other categorical attributes is similar to that of the initial dataset.

## 3.2 K-means algorithm

For K-means algorithm the number of clusters must be specified as an input parameter. Changing this parameter changes the clustering. To determine the optimal number of clusters first the plot of SSE against number of clusters k should be made, and then values of k around the knee of the function should be investigated. Plotted function of our dataset does not have pronounced knee, which could suggest that K-means is not the best clustering algorithm for this dataset. The interval that could correspond to knee is [13-20].



To measure the quality of the clustering we considered SSE and Silhouette coefficient, trying to balance between small SSE, high Silhouette. We also prefered small number of clusters. With increasing k, SSE decreases, and Silhouette increases simultaneously. For each k we run K-means for 10000 different choices of initial centroids. The results with smallest SSE and corresponding Silhouette coefficients in each run are reported in the table below:

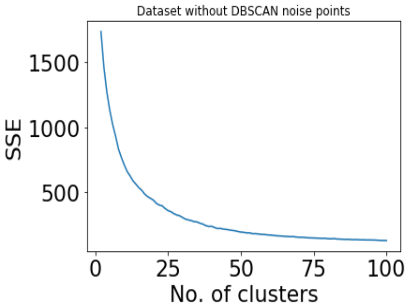
| No. of clusters | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|
| SSE | 1174 | 1130 | 1089 | 1050 | 1015 | 985 | 956 | 929 |
| Silhouette | 0.281 | 0.287 | 0.246 | 0.250 | 0.256 | 0.261 | 0.264 | 0.271 |

*Table 6 Number of clusters and relative SSE and Silhouette*

For higher number of clusters SSE continue to decrease as expected, and Silhouette fluctuates between 0.27 and 0.31.

The highest Silhouette is for k = 14. For k = 15 SSE is smaller, but there is one cluster more and also sharp drop in Silhouette. Therefor our choice for optimal clustering was k = 14. Its composition in term of number of points is following: [2786, 1911, 498, 492, 479, 439, 439, 425, 419, 385, 366, 283, 278, 271]

The DBSCAN algorithm determines the noise points in dataset. The noise points from the best DBSCAN clustering were removed, and K-means were run again on this new dataset. The SSE(k) function in this case is:



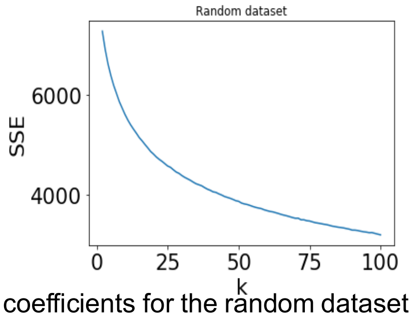
SSE(k) has a bit more pronounced knee in this case respect to previous one, indicating that K-means is more appropriate for this dataset. We investigated SSE and Silhouette around the knee interval [9-16]:

| No. of clusters | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|
| SSE | 763 | 706 | 658 | 622 | 587 | 552 | 524 | 504 |
| Silhouette | 0.304 | 0.318 | 0.328 | 0.281 | 0.292 | 0.302 | 0.309 | 0.313 |

Drop in SSE stabilizes after k = 11. The Silhouette is also the best for k = 11, therefore we selected this clustering as the optimal one. The clustering composition is:

| cluster No. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| number of points in cluster | 2616 | 1878 | 432 | 417 | 411 | 402 | 384 | 370 | 320 | 293 | 255 |

Comparison with random dataset – We plot SSE(k) for random dataset: this function does not have clear knee structure, and  SSE values are order of magnitude higher respect to two previous datasets.



Next table reports SSE and Silhouette coefficients for the random dataset for the same k values as for dataset without the noise.

| No. of clusters k | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|
| SSE for random data | 5721 | 5593 | 5483 | 5387 | 5300 | 5216 | 5140 | 5069 |
| Random Silh. | 0.070 | 0.071 | 0.072 | 0.072 | 0.073 | 0.073 | 0.074 | 0.075 |

For every clustering SSE of random dataset is order of magnitude higher respect to SSE of dataset without noise. Silhouette of random dataset is 5 times smaller respect to Silhouette of dataset without noise. All this indicate that K-means has found clustering structure in the dataset not due to chance.

### 3.2.1 K means clustering characterization

K-means clustering characterization was performed, and clusters with the distribution of at least one categorical attribute significantly different respect to initial dataset are reported. For these clusters categorical attributes with the distributions very similar to initial dataset are not indicated.

- Cluster No. 0:
  The biggest cluster with 34% of records. From the centroid analysis follows that typical customer in this cluster has average limit is 96194$, average monthly debt of 26163$ and average monthly repayment amount of 1432$, where latter two are uniformly distributed over the months.
  From the distribution of categorical attributes typical record in this cluster has 74% probability of using revolving credit, 83% of not going into default.
- Cluster No 1:
  Cluster with 5% of records. According to centroid the typical customer in this cluster has limit 121741$ and average ba of 8829$ evenly distributed over the months. The average pa is 348$ small amount respect to a mean of 2342$. Also, pa value for April, July and September are small (302$, 0$, 443$) respect to pa of May, June and August (1681$, 1695$, 3018$) indicating delays in repayment. From the distribution of ps-avg follows that 69% of records have repayment delay of 1 month (47%) or 2 months (22%). Credit default No has dropped respect to the initial dataset distribution to 62%.
- Next five clusters, No. 2, 3, 6, 7 and 10 have almost identical qualitative structure. From the centroids follow: Values of limits are somewhat above $100000. Average monthly ba values are uniformly distributed over 6-moths period. For clusters 2, 6, 7 and 10 they slightly increase over time. Pa values are very unevenly distributed fluctuating between 0 and some maximal value, suggesting delays in repayment. Average pa values are between $1000-$1600. Credit default majority is No in all cases and has value around 60% which is much less respect to initial dataset value of 78.3%.

The exact values mentioned above are summarized in the next table:

| Cluster No. | limit | ba-avg | ba-may | ba-jul | ba-aug | pa-avg | pa-min | pa-max | ps-avg | credit_default No (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 102106 | 24153 | 22428 | 24237 | 25795 | 1162 | 0 | 2372 | 1-35% 2-26% | 62 |
| 3 | 105645 | 28694 | 26109 | 25959 | 26541 | 1037 | 0 | 2238 | 1-42% 2-26% | 61 |
| 6 | 105710 | 26919 | 28860 | 31125 | 30180 | 1180 | 0 | 3018 | 1-47% 2-22% | 62 |
| 7 | 103446 | 30806 | 29888 | 32711 | 30027 | 1170 | 0 | 2512 | 1-62% 2-0.1% | 59 |
| 10 | 100743 | 36595 | 35569 | 36677 | 40721 | 1593 | 0 | 3399 | 1-63% 2-17% | 62 |

We merged these clusters into the bigger one containing 24% of records with following centroid and categorical attribute values:

| Merged cluster No. | limit | ba-avg | ba-may | ba-jul | ba-aug | pa-avg | pa-min | pa-max | ps-avg | credit_default No (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| 2,3,6,7,10 | 103684 | 28880 | 27911 | 29459 | 29807 | 1200 | 0 | 3399 | 1-48% 2-21% | 61 |

The new clustering contains 7 clusters, with new SSE = 1250 and new Silhouette = 0.235. The Silhouette is smaller than the clustering with k = 11, but is still 3.5 times higher respect to random dataset (0.068).

- Cluster No 10:
  According to centroid the typical customer in this cluster has limit $275464. Average monthly ba is $67157 slightly increasing over 6 months period (ba-may: $62546, ba-jul: $67624, ba-aug: $71316). Pa values have the same behavior increasing from April $5267 to September $6292, with the average pa of $5642. The ps-avg has 69% of records with the use of revolving credit. Credit default majority of No is high at 92%.

## 3.3 Hierarchical clustering algorithm

### 3.3.1 Complete link

For the dendogram cuts that have from 2 to 41 clusters the best Silhouette = 0.378 was found for 34 clusters. Three of these clusters was singletons (noise points), so after their elimination we have 31 clusters with following composition: [4600 147 471 237 122 84 364 100 77 70 419 290 404 69 223 274 267 62 84 143 172 126 63 59 62 50 217 25 49 97 41]

This clustering has one big cluster whose composition in terms of categorical attributes is identical to the big cluster in DBSCAN (No. 0). Similarly, to DBSCAN clustering the rest of the clusters are all small or very small. Differently to DBSCAN all these clusters have the credit default No as a majority. We tried to find some clusterings different from optimal that have some clusters with credit default No majority. Clusterings for 5,8,12 and 17 have been investigated, but credit default Yes majority has never been observed. Distribution of all other categorical attributes was similar to one in the initial dataset.

For random dataset maximal Silhouette is 0.018 (for 2 clusters), indicating that the structure of dataset according to hierarchical clustering (complete link, Euclidean metric) is very different respect to random data.

### 3.3.2 Single Link

For single link the best Silhouette = 0.339 is for 2 clusters. The dendogram cut for any number of trees from 2 to 41 produces clustering composed of one big clusters and all other singletons or very small clusters. Together with the fact that single link is very sensitive to noise points implies we have very noisy dataset. Single link Silhouette for random dataset is around 0.22 indicating that the best-found clustering is not very good in this case.

### 3.3.3 Average Link

For average link the best Silhouette = 0.405 for 2 clusters but this value is, like the previous one, much less significant respect to complete link because random data Silhouette = 0.2. Dendogram cut for any number of trees from 2 to 41 produces clustering composed of one big cluster, approximately one half of clusters are singletons and other half are small clusters. As average link is good compromise between single and complete link this result was expected.

### 3.3.4 Other metrics

We also investigated L1 (cityblock) and L∞ (chebyshev) metrics and all clusterings were poorer respect to L2. For the lack of space, we do not show the details.

## 3.4 Final evaluation of the best clustering approach

Credit_default Yes records are of interest for this project. There are only 22% of these records in initial dataset making it highly unbalanced respect to credit_default attribute. So, it is improbable to find evenly distributed clustering structure with clusters containing credit_default Yes majority.

DBSCAN and hierarchical clustering revealed one big cluster (around 47% of records), and K-means revealed two big clusters that are together of the similar size and structure. This big core of records has credit_default No majority with probability higher than that of initial dataset. So, credit_default Yes improved its distribution among the remaining records increasing the probability of finding clusters with credit_default Yes. However, among the three clustering algorithms only DBSCAN was able to find these clusters.

We also tried to run hierarchical clustering on dataset without the noise points found by DBSCAN. The best clustering was half a way between K-means and DBSCAN: 22 clusters with Silhouette 0.36 where two clusters had 47% of records with the same structure as before. But this time hierarchical clustering found 3 small clusters with credit_default Yes. After their merge we obtained cluster with credit_default Yes majority of equal size and structure as in the case of DBSCAN. This was the ulterior proof that initial dataset is very noisy and that hierarchical clustering was not able to reveal interesting structure due to so many noise points.

It seems that dataset has structure where one big part of the data is very similar respect to all clustering algorithms (ps-avg 0 or 0.5, with high probability of credit_default No), and rest of data with many smaller clusters. But as the data are very noisy only DBSCAN was able to reveal clustering of our interest. So, for this problem DBSCAN is the most appropriate algorithm.

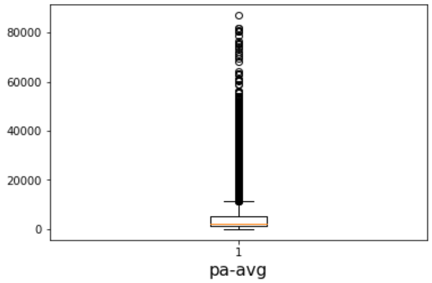
# 4. Classification

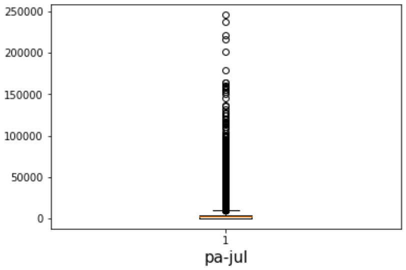
Our project up to now was concerned with unsupervised approach to predict credit default of cardholders in Taiwan. Now we move to classification, the supervised approach where the goal is to learn the predictive model (called classifier) using known records that will be able to predict credit default of new instances,

## 4.1 Data preparation for classification

We applied classification on three datasets. The first was the initial dataset with discretized 'age' attribute.

After data preparation we were left with the set of numerical attributes at which logarithmic transformation was applied. These were 3 ba and 6 pa attributes. This was the second dataset on which we built decision tree classifier. One of the properties of good classifier is clear interpretability of results. As each of ba and pa attributes alone is not easily comprehensible, we decided to introduce two new attributes ba average (ba-avg) and pa average (pa-avg) instead of pure ba's and pa's. But this transformation also improved their sparseness, so we did not use their logarithm. This improvement can easily be seen from the following boxplots:





This was the third dataset for decision tree construction. To summarize the datasets have the following attributes:

1. Dataset 1: a dataset composed by: ‘ba-aug’, ‘ba-jul’, ‘ba-may’, ‘pa-sep’, ‘pa-aug’, ‘pa-jul’, ‘pa-jun’, ‘pa-may’, ‘pa-apr’;
2. Dataset 2: the same values as Dataset 1 but with the logarithms of the attributes’ values;
3. Dataset 3: a dataset that contains the averages of ‘pa’ and ‘ba’: ‘ba-avg’, ‘pa-avg’

All the previous datasets contained also: ‘age-discrete’, which is the age discretized with 9 years binning, ‘sex’, ‘education’, ‘status’ and ‘ps-avg’ which is the average of all the ‘ps’ values. The datasets 1 and 2 do not include ‘ba-apr’, ‘ba-jun’ and ‘ba-sep’ because highly correlated to the other attributes.

Data needed ulterior processing because the *sklearn* library supports only numerical values in Decision Tree algorithm. Therefore, all categorical values have been mapped into numerical one. Education is an ordinal attribute, so we have chosen numerical values in such a way to preserve ordering. These transformations are the following:

- ‘sex’: we changed ‘male’ with 0 and ‘female’ with 1;
- ‘education’: we assigned 0 to ‘others’, 1 to ‘high school’, 2 to ‘university’ and 3 to ‘graduate school’;
- ‘status’: unlike ‘education’, ‘status’ is a nominal attribute, so we used dummies to process the attribute and create three different attributes, one for every status value: ‘single’, ‘married’, ‘others’.

Before the execution of the Decision Tree algorithm, we have split the datasets into training set and test set in proportion 80% - 20%. We further have split training set to new training set and validation set in the same proportion as before. So, we divided every original datasets in training, validation and test set with the respective proportions of 64%, 16% and 20%.
The distribution of the ‘credit_default’ attribute is unbalanced, 78% of values are ‘no’ and (the) 22% are ‘yes’. We also made the splitting with stratification respect to the class attribute.

| | Training set | Validation set | Test set |
|---|---|---|---|
| Total records | 6061 | 1516 | 1894 |
| Credit_default = ‘yes’ | 1310 (21.7%) | 329 (21.7%) | 411(21.7%) |
| Credi_default = ‘no’ | 4741(78.3%) | 1187(78.3%) | 1483(78.3%) |

*Table 7: Distribution of records within the sets.*

## 4.2 Model selection

The *sklearn* DecisionTreeClassifier method doesn't provide a postpruning function, so we decided to find the best classifier using the *sklearn* RandomizedSearchCV method. This method was applied with different combinations of parameters:

- 'criterion': it quantifies the split quality, with two possible impurity measures 'gini' for the Gini index and 'entropy' for information gain;
- 'max_depth': the maximum depth of the decision tree;
- 'min_samples_split': the minimum number of samples required to split an internal node;
- 'min_samples_leaf': the minimum number of samples required that node becomes a leaf;
- 'min_impurity_decrease': the node will be split if decrease in selected impurity measure is higher or equal to this parameter.

Different combinations of these parameters correspond to different preprunings of the decision tree in construction. For each dataset we performed the following:

The above method was run for different parameter combinations, and for two impurity measures: Gini index and Entropy. Among these we considered fifty best results with regard to the accuracy on the training set. Than the classifier with the smallest validation error has been chosen.

The following table summarizes these results for each dataset:

| Dataset | Criterion | Max depth | Min samples split | Min samples leaf | Min impurity decrease | Accuracy on Training set | Validation Error | Training error |
|---|---|---|---|---|---|---|---|---|
| 1 | Gini | 5 | 200 | 30 | 0.0002 | 0.8158 | 0.2031 | 0.1841 |
| 2 | Gini | 5 | 15 | 20 | 0 | 0.8173 | 0.2044 | 0.1826 |
| 3 | Gini | 7 | 10 | 20 | 0.001 | 0.8092 | 0.1965 | 0.1907 |

Gini index gives the best classifier for every dataset.

## 4.3 Model evaluation

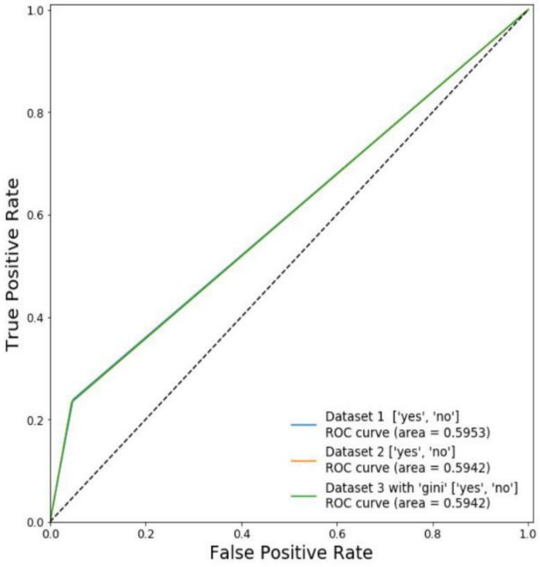
For model evaluation we used 10-fold cross validation. For each classifier average of test set errors has been calculated. This together with mean Accuracy, Recall and F1 is reported in the following table:

| Dataset | Accuracy (test set) | Recall | F1 | Mean test-set error |
|---|---|---|---|---|
| Dataset 1 | 0.7972 | 0.80 | 0.76 | 0.2029 |
| Dataset 2 | 0.7983 | 0.80 | 0.76 | 0.2031 |
| Dataset 3 | 0.7983 | 0.80 | 0.76 | 0.1991 |

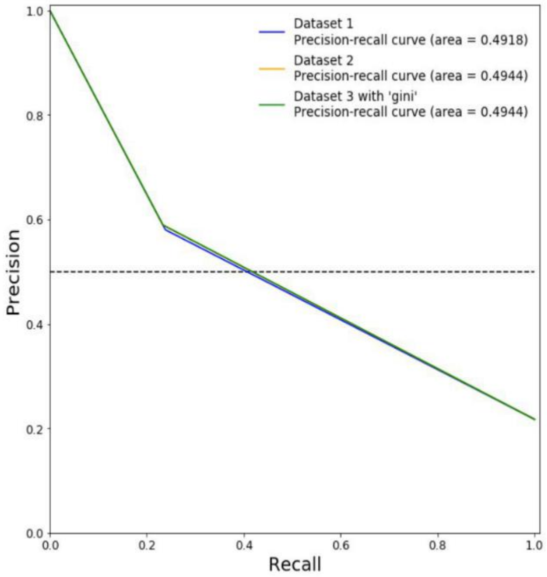
Comparing classifier errors on training and test sets, potential model overfitting can be verified. The test error is under 1.3% higher respect to training one in all cases, indicating neither model is in overfitting.

## 4.4 Model comparison

We plotted the ROC curves and AUC values for each classifier in order to choose the best one. We deal with highly unbalanced dataset respect to credit default attribute. As the ROC Curve is not good in this case we plotted also the Precision-Recall curve. The Precision-Recall curve summarizes the trade-off between the true positive rate (precision) and the positive predictive value (recall) for a model using different probability thresholds.



9 *ROC Curve for the models.*



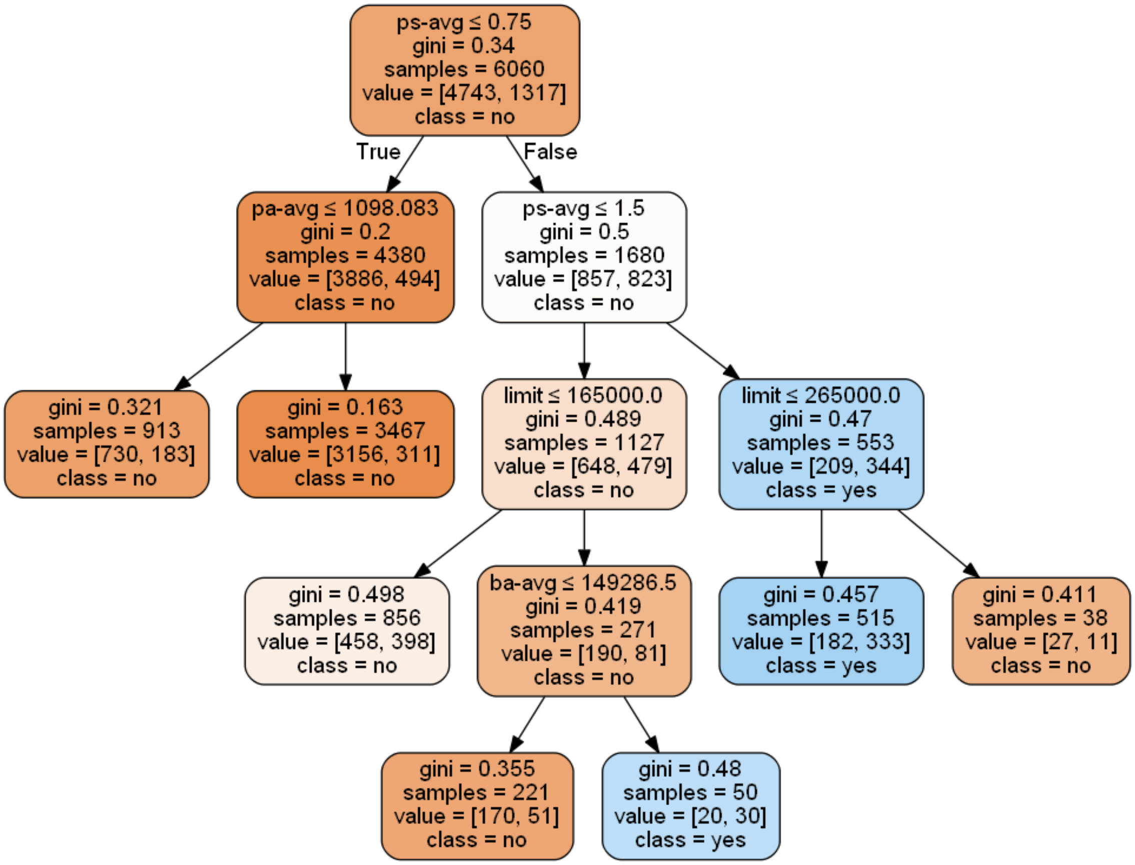
10 *Precision-Recall curves for the models.*

From the figure 9 and 10 it is very difficult to choose the best model, as the area under curve is very similar in all cases. So, we decided to use statistical test of significance for the best model selection. We compared classifiers in pairs, calculated interval of confidence for test error differences with confidence of 95%. Following table contains confidence itervalls for pairs of classififiers:

| pair classifiers | interval of confidence |
|---|---|
| 1,2 | 0.027 - 0.025 |
| 2,3 | 0.055 - 0.006 |
| 1,3 | 0.056 - 0.007 |

So, the third model has smaller test error respect to other two models. For first and second we cannot decide which has smaller error with 95% confidence.

## 4.5 Decision tree interpretation



*11 Decision Tree*

Figure 11 represents the Decision Tree of the best classifier (Dataset 3, gini). From the tree we can see that:

- The root shows that the subjects that usually repaid their debit on time (‘ps-avg’ <= 0.75) default ‘NO’. We can also assume that if the average amount of repayment the subject did is less than 1098$ it’s higher the risk of insolvency. In fact, there are 183 misclassified values.
- In the right branch, we can see that the distribution of defaulted and not defaulted is balanced. If we consider the child nodes:
    - if the delay of the repayment is greater than 1.5, there is a high risk of insolvency.
    - if the delay is less than 1.5 there is a balance in the distribution, but if they spent more than 149286.5$ there is a higher risk of insolvency also because they could easily spend more than their limit which is greater than 165000$.

# 5. Association Rules Mining

The association rules goal is to find patterns within the dataset that have a significance to our analysis.

## 5.1 Dataset Preparation

Similarly, to classification, attributes used in Pattern Mining and Association Rules have to be comprehensible and interpretable. Therefore, we used average values of ba and pa attributes, that give us mean monthly spending and repayment amount during considered time interval. But unlike the classification where discretization of numerical attributes is done by algorithms, here we have to select the optimal binning. As numerical values for ba are sparse and for pa very sparse we have chosen equal frequency binning instead of equal width binning. To select the optimal number of bins we had to compensate between two opposite trends: data are very densely populated near zero, so the bins should divide this region in number of intervals that are enough large to be interpreted as different spending amounts. But on the other hand, they should be not too large to contain very high percentage of records, making binning meaningless. As a result of these considerations we have chosen 10 bins of frequency 947. As number of records is not divisible by 10 (9741 records), the last bin has 1 record more, 948.

Next, we report edges of binning intervals and corresponding bin numbering for limit, ba-avg and pa-avg:

| limit:<br>bin edges | 10000 | 30000 | 50000 | 70000 | 100000 | 130000 | 170000 | 200000 | 260000 | 350000 | 740000 |
|---|---|---|---|---|---|---|---|---|---|---|---|

| bin No. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|

| ba-avg:<br>bin edges | -23259 | 650 | 2601 | 7045 | 13914 | 20683 | 31351 | 45234 | 68609 | 110698 | 386190 |
|---|---|---|---|---|---|---|---|---|---|---|---|

| bin No. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|

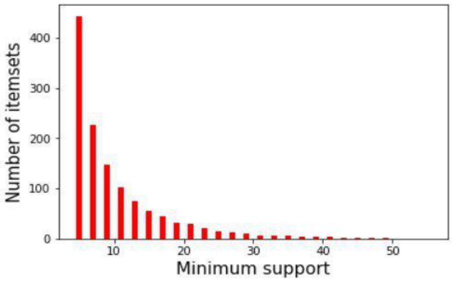
| pa-avg:<br>bin edges | 0 | 528 | 969 | 1338 | 1754 | 2342 | 3249 | 4517 | 6333 | 11160 | 316053 |
|---|---|---|---|---|---|---|---|---|---|---|---|

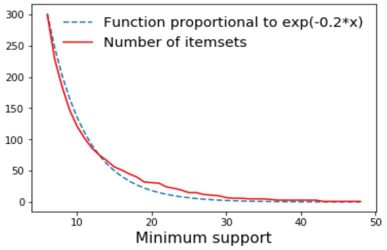
| bin No. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|

Above every bin number there are two values that indicating the edges of the binning interval.
All other categorical attributes from initial data preparation are used also here. These are status, sex, education, credit default, ps-avg and already discretized age.

## 5.2 Frequent patterns extraction and discussion of the most interesting frequent patterns

On the following left figure, we plotted number of itemsets in dependence of minimum support. The most frequent itemset has minimum support of 49%. Decreasing the value of minimum support, number of frequent itemsets increase exponentially, as can be seen from the plot on the right.





For some attribute values, respect to the others, their relative frequencies are particularly high. These are credit default: no (78%), sex: females (61%), status: single (53%), ps-avg: 0.5 (47%), education: university (47%). So, we expect that the most frequent itemsets should contain some of these items. We report the 6 most frequent itemsets in the following table:

| No. | Support | Itemset |
|---|---|---|
| 1 | 49 | ('female', 'no_default') |
| 2 | 42.5 | ('0.5_ps', 'no_default') |
| 3 | 42 | ('single', 'no_default') |
| 4 | 36 | ('university', 'no_default') |
| 5 | 35 | ('married', 'no_default') |
| 6 | 32 | ('single', 'female') |

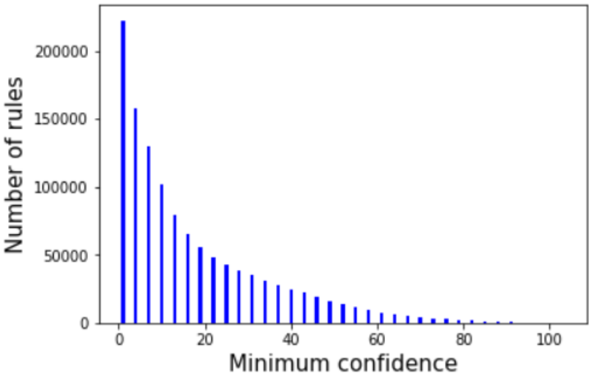
For frequent itemsets and later for association rules we used notation where bin number is followed by corresponding attribute abbreviation: '_age' for age, '_ps' for ps-avg, '_limit' for limit. Similar notation we used for credit default: 'yes_default' and 'no_default'. Other attribute values are identical to those in initial dataset.

- First itemset does not have any meaning that follows from attributes semantic but is consequence of the fact that female and credit default no are two most frequent attribute values in the dataset. Similar observation is valid for the sixth itemset.
- The third and the fifth itemsets contains status values with credit default no. This suggests that credit default no is independent of status. This is in accordance with results of clustering where every clustering algorithm and every significant cluster contained distribution of status very similar to initial dataset. Also, in classification the decision tree with leafs of significant number of records (after pruning) was not influenced by status attribute.
- The second one is somewhat expected-clients that use revolving credits will not go to credit default. Otherwise this type of account feature would not be the most used one.
- Forth itemset is also somehow expected, people with university degree are generally more paid which probably influence their repayment ability.

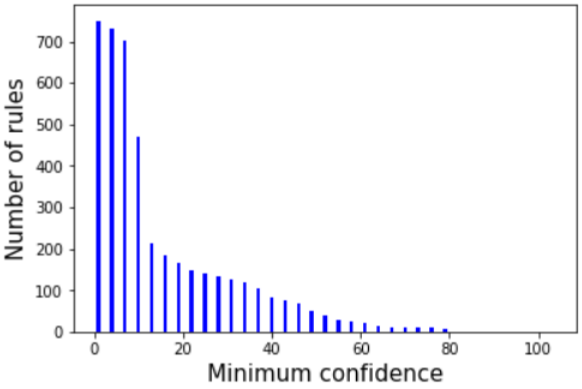
We investigated frequent itemsets down to support = 20 and we did not find any unexpected or interesting results. First itemset that contains 3 items is: ('0.5_ps', 'female', 'no_default'), with support = 26%.

## 5.3 Association rules extraction and discussion of the most interesting

The plot in figures 11 and 12 show the distribution of the number of rules as function of the minimum confidence with two different minimum support values. The plot on the left represents the distribution with *min_sup* = 1, otherwise the one on the right has *mis_sup* = 2. As we can see in both plots the number of rules extracted decrease as the minimum confidence increase. The different supports only change the numbers of rules extracted, our goal is to find a good balance between the minimum support value and the number of rules. In fact, we need to find meaningful rules maintaining a proper support level.

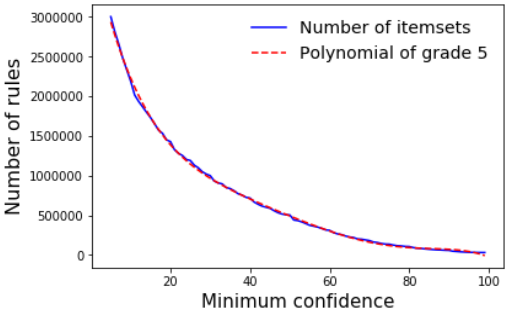


*11 Distribution of the number of rules as function of the minimum confidence with min_sup = 1*



*12 Distribution of the number of rules as function of the minimum confidence with min_sup = 30*

The number of all possible association rules growing exponentially with the cardinality of itemsets. For all practical purposes we are interested in subset of all association rules determined by some value of confidence. The following plot shows the number of association rules as a function of confidence for minimum support = 1%.

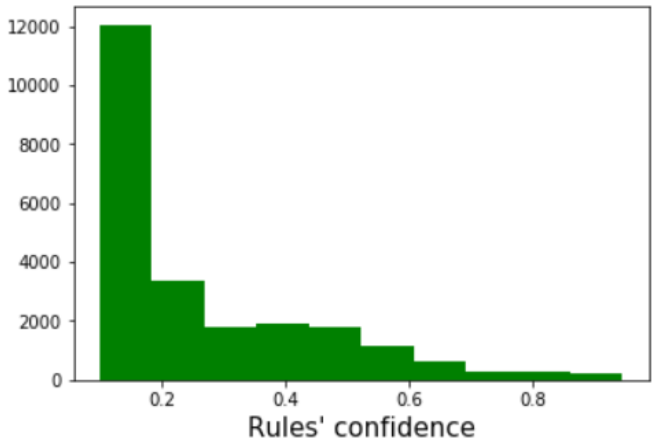


As a number of rules is reduced by the minimum confidence value this function is not exponential. The fit reveals polynomial behaviour.

Obtained rules will be applied on the initial dataset to treat missing values. We wanted to see what are the best rules for each missing attribute value to be treated. Lift and confidence were the primary selection criterion. Sometimes many rules had similar lift and accuracy but very different support. In that case we have chosen one with the highest support.  We can observe that regardless of the fact that itemset can have high support e.g. ('female', 'no_default') of 49%, corresponding rules 'female' → 'no_default' and , 'no_default' → 'female' have small lift of 1.02 indicating that items are not correlated.

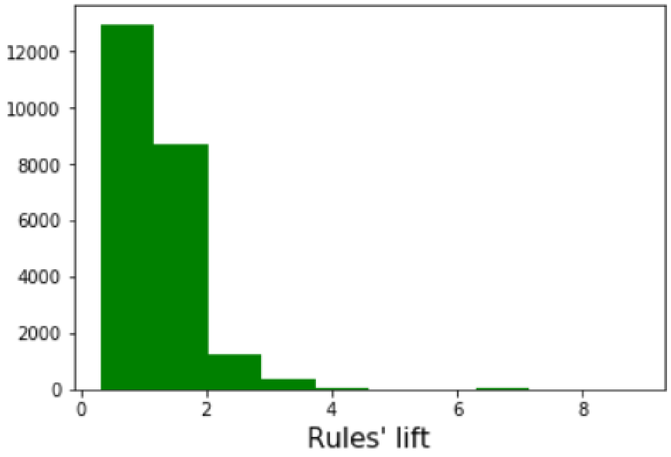
| No. | Rule | Support | Confidence | Lift |
|---|---|---|---|---|
| 1 | ('2.0ps',) à ('yes') | 4% | 60% | 2.8073 |
| 2 | ('8_ba', '0.5ps', 'female') à (‘no’) | 4% | 94% | 1.207 |
| 3 | ('2_age', 'graduate school') à ('single') | 9% | 85% | 1.5861 |
| 4 | ('4_age', '0.0ps', 'no') à ('married') | 3% | 70% | 1.5243 |
| 5 | ('0_limit', 'yes', 'single') à  ('male') | 1% | 66% | 1.701 |
| 6 | ('2_age', 'married', 'university', 'no') à  ('female') | 3% | 77% | 1.2513 |
| 7 | ('yes', '2_age', 'married', 'female') à ('university') | 1% | 77% | 1.5882 |
| 8 | ('0.0ps', '3_age', 'single') à ('graduate school') | 3% | 61% | 1.8024 |
| 9 | ('5_age', '3_limit', 'married', '0.5ps', 'no') à  ('high school') | 0.2% | 69% | 4.0565 |

The histogram in figure 13 represents the frequency of the rules’ confidence i.e. how many times a certain confidence appears in among the rules, starting from *min_conf* = *10*%.
The plot shows that there are a few rules with a high confidence and a lot of rules with a small one.



13 *Frequency of the rules' confidence*

The histogram in figure 14 represent the frequency of the rules' lift starting from *min_conf = 10*%.
The plot shows that there are a lot of rules with a low lift. There are no rules with a lift from 4.5 to 6 and there are just a few with a lift higher than 6.



14 *Frequency of the rules' lift*

## 5.4 Replacing the missing values and evaluation of the accuracy

In order to replace the missing values, we generated a dataset without outliers and maintained the missing values.
To fill the missing values with the predicted ones at first, we generated the rules without support and then filtered from the possible options the ones within a certain support and lift values to find to best most meaningful rules to apply.
We chose to set the *min_conf* = 60%, in order to maintain a high reliability of the rules generated.
Then, we extract the rows with the missing values in relation to a certain target variable. To be sure to apply the best rules to predict the missing values we sorted the rules following a descendent order with respect to the Lift value.
Finally, for every record, we searched for the best value to assign.

We replaced the values following the number of the missing values, to be able to assign the best predictions to the attributes with more missing values. So, we replaced in order: sex, education, age and status.

### Replacing sex's missing values

The attribute sex has 93 missing values.
We have selected 770 rules from the 252278 possible options, using a *min_sup* = 2 and Lift > 1. A higher lift or support would not generate enough rules to predict the values. And a lower would not produce as good rules.
The prediction assigned 87 values to 'female' and 6 to 'male'.

### Replacing education’s missing values

The attribute education has 118 missing values.
We have extracted 1230 rules from the 187569 possible options, using a *min_sup* = 0.5 and Lift > 1.2.
The prediction assigned 24 values to ‘graduate school’ and 94 to ‘university’.

### Replacing age’s missing values

The attribute age has 900 missing values.
We have extracted 5910 rules from the 225073 possible options, using a *min_sup* = 0.2 and Lift > 1.4. Because of the high number of possible values to obtain a good set of rules we had to decrease the confidence value to 50%.
The prediction assigned 526 values to ‘2_age’, 274 to ‘3_age’ and 100 to ‘a4_age’.

### Replacing status’s missing values

The attribute status has 1735 missing values.
We have extracted 1184 rules from the 241021 possible options, using a *min_sup* = 1 and Lift > 1.4.
The prediction assigned 992 values to ‘single’ and 743 to ‘married’.
To evaluate the accuracy of the missing values’ predictions we used the rules generated with the previous dataset to evaluate the accuracy on the dataset with the predicted missing values using as target the ‘credit_default’ value. The accuracy is 0.7989

## 5.5 Prediction of the target variable and evaluation of the accuracy

To predict the target variable ‘credit_default’, the process was the same as the one used in the missing values replacing.
We extracted the rules using a *min_conf = 60*, and we filtered the rules with a *min_sup* = 3 and a lift > 1. The number of meaningful rules extracted is 412 from the 288640 possible options. The resultant accuracy is 0.7994.

In addition, we use the same rules to predict the values for the test dataset and uploaded it on Kaggle. The resultant accuracy is 0.80166, so the rules we used are enough meaningful.

# 6. Conclusions

The class attribute distribution was very unbalanced. Together with the fact that data were very noisy, as revealed by the clustering diminished the possibility of finding relevant information for credit default attribute.

Unsupervised approach found intrinsic clustering structure in the data, but as expected in these circumstances DBSCAN was the only algorithm able to reveal some information about credit default yes.

Supervised approach gave more satisfying result, especially the Association Rules with the accuracy of 0.8. This is in accordance with the fact that Classification is more susceptible to the noise.