CREDIT CARD DEFAULT PREDICTION

:- Machine learning using Python

SAQUIB NAZEER , CSE 3rd YEAR , KALYANI GOVERNMENT ENGINEERING COLLEGE

AKASH SHARMA , IT 3RD YEAR , KALYANI GOVERNMENT ENGINEERING COLLEGE

PUSPITA UTHYASANI ,CSE 3rd YEAR , KALYANI GOVERNMENT ENGINEERING COLLEGE

ANIKET MONDAL , IT 3RD YEAR ,KALYANI GOVERNMENT ENGINEERING COLLEGE

Table of Contents

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

I owe my deep gratitude to our project guide MR Titas Roy Chowdhury, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

SAQUIB NAZEER

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

I owe my deep gratitude to our project guide MR Titas Roy Chowdhury, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

AKASH SHARMA

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

I owe my deep gratitude to our project guide MR Titas Roy Chowdhury, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

PUSPITA UTHYASANI

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Titas Roy Chowdhury for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessings, help and guidance given time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

I owe my deep gratitude to our project guide MR Titas Roy Chowdhury, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

ANIKET MONDAL

PROJECT SCOPE

Credit default will cause huge loss for the banks, so they pay much attention on this issue and apply various method to detect and predict default behaviours of their customers. In this project, we are going to develop the basic process of loan default prediction with machine learning algorithms.

Machine learning algorithms have a pretty good performance on this purpose, which are widely-used by the banking. Various businesses use machine learning to manage and improve operations. While ML projects vary in scale and complexity requiring different data science teams This project will serve as a benchmark on how good popular methods are on large imbalanced datasets.

Due to the significance of credit card lending, it is a widely researched subject. Many statistical methods have been applied to developing credit risk prediction, such as discriminant analysis, logistic regression, and probabilistic classifiers such as Bayes classifiers. Advanced machine learning methods including decision trees and artificial neural networks have also been applied . The large extent of studies in this field will aid the project team in determining an appropriate methodology to achieve good results. The word default in credit card sector basically means make a payment on a debt by the due date. If this happens with a credit card, creditors might raise interest rates to the default (or penalty rate) or decrease the line of credit. In case of serious delinquency, the card issuer can even take legal action to enforce payment or to garnish wages.

With constant need for computer usage in banking sector, the increasing number of bank accounts created and credit outflows, computer can play the role of assistive banker in ensuring default less credits and safe banking. Among this credit default has been a major challenge for bankers as it endangers their system and pose chances of losses that might be hard to revert. Credit card defaulters are on the rise too. Predicting the nature of a customer of whether he might be a defaulter or not is a complex function. Though the law has stringent measures against credit card defaulting, it is still prevalent in most parts. In India, credit card defaulters are charged under both civil and criminal cases. In this research work, the performance measures of classification algorithm are compared on the Credit card default dataset published in UCI machine learning repository [3]. The effect of feature selection algorithms in analysed

The data set is originally from a Taiwanese bank, collected from October 2005. However, two additional data features (location, employer) were added by McKinsey to add further possibilities and depth into the analysis. At least one published study by I-Cheng Yeh and Che-hui Lien uses the original data to compare the predictive accuracy of probability of default of six different data 3 mining methods. However, the data used in this project is not a one-to-one match. In addition to academic research, the data set has been analysed in community-based platforms, such as Kaggle.

PROJECT OBJECTIVE

The fundamental objective of the project is implementing a proactive default prevention program and identifying customers with high probability of defaulting to improve the client's bottom line. The challenge is to help the bank to improve its credit card services for the mutual benefit of customers and the business itself. An emphasis on creating a human-interpretable solution must be put into consideration in each stage of the project. Even though plenty of solutions to the default prediction using the full data set have been previously done, even in published papers, the scope of our project extends beyond that, as our ultimate goal is to provide an easy-to-interpret default mitigation program to the client bank. In addition to default prevention, the case study includes a set of learning goals. The team must understand key considerations in selecting analytics methods and how these analytics methods can be used efficiently to create direct business value. McKinsey also sets the objective of learning how to communicate complex topics to people with different backgrounds.

The default prediction algorithm is, like its name says, a model used to predict credit card defaults based on our dataset. We think our best bet is to implement a machine learning algorithm since this has been previously done with a similar dataset [1]. There are many approaches to this problem and we also have to take into account the situation of the bank. For the bank, it would be the most beneficial to prevent defaults by filtering out risky customers by not giving them credit cards at all. However, this means that we cannot use the credit card payment history for the prediction since that would not be available at time of issuing the credit card. This would naturally make predicting much harder, as we would be using just a fraction of the data but it would be more beneficial for the bank financially. The financial benefit of the bank must be kept close to the algorithm development as predictive accuracy is not the only metric with which the algorithm performance is evaluated.

>>>>Data Description

Data source: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset We acknowledge the following: Lichman, M. (2013). UCI Machine Learning Repository http://archive.ics.uci.edu/ml Irvine, CA, University of California, School of Information and Computer Science.

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005

There are 25 variables:

- ID (id):
  Identity No. of each client.

  Data Type: Integer

  Value type: Continuous

  Null value percentage: 0

- LIMIT_BAL (lb): Amount of given credit in NT dollars (includes individual and family/supplementary credit.

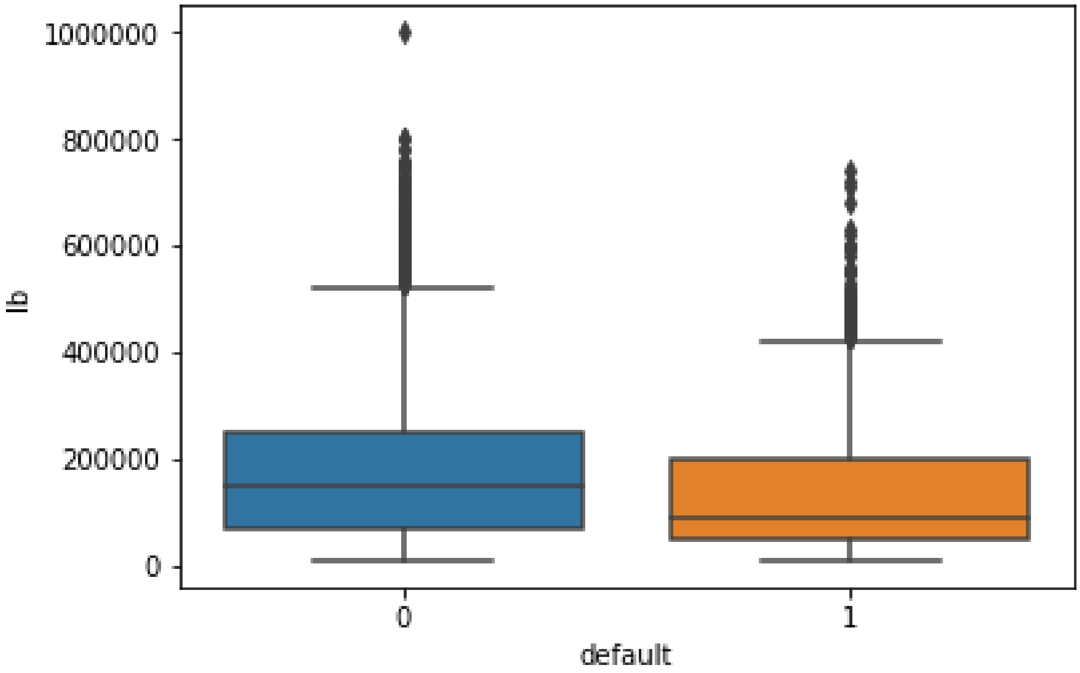
  Data Type: Integer

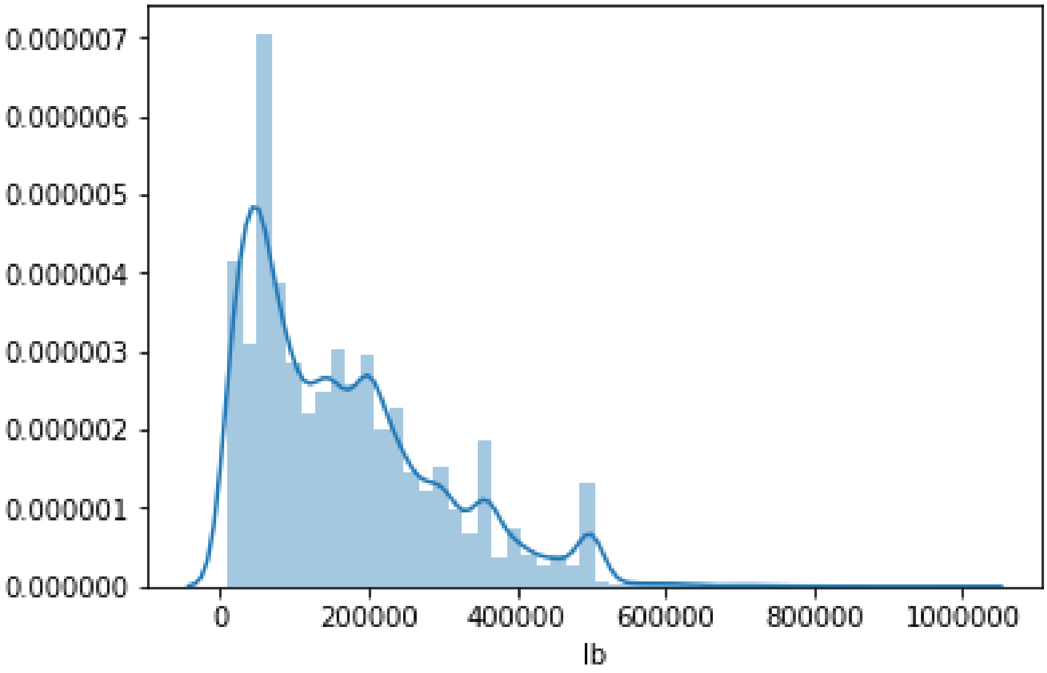
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean     167484.322667
std      129747.661567
min       10000.000000
25%       50000.000000
50%      140000.000000
75%      240000.000000
max     1000000.000000
```



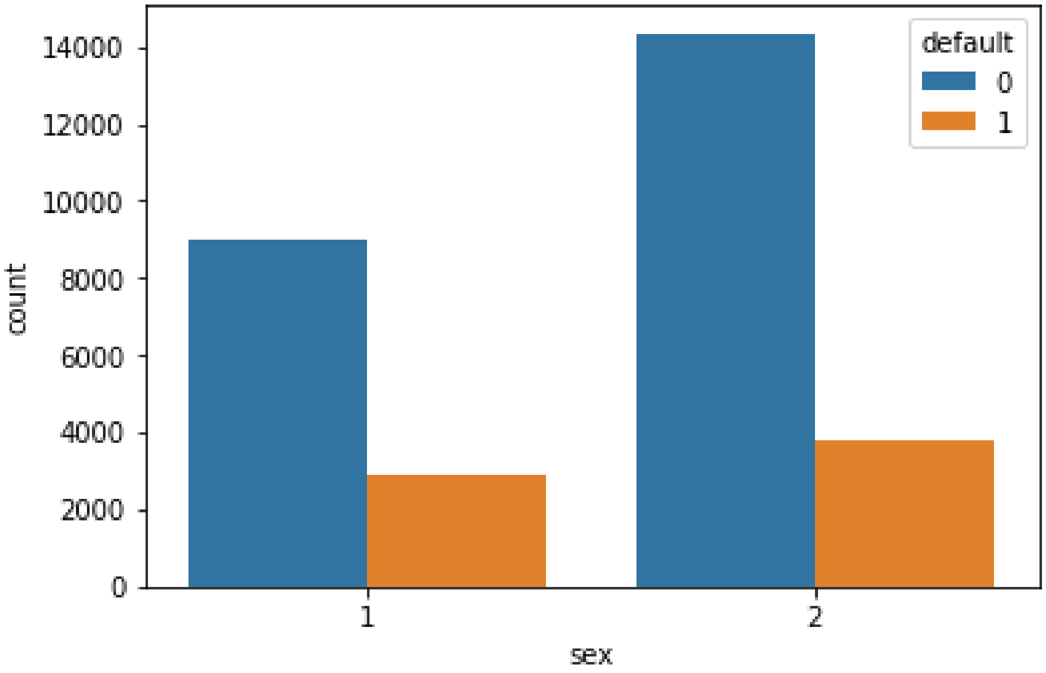


- SEX (sex): Gender
  (1=male, 2=female)

  Data Type: Integer

  Value type: Categorical

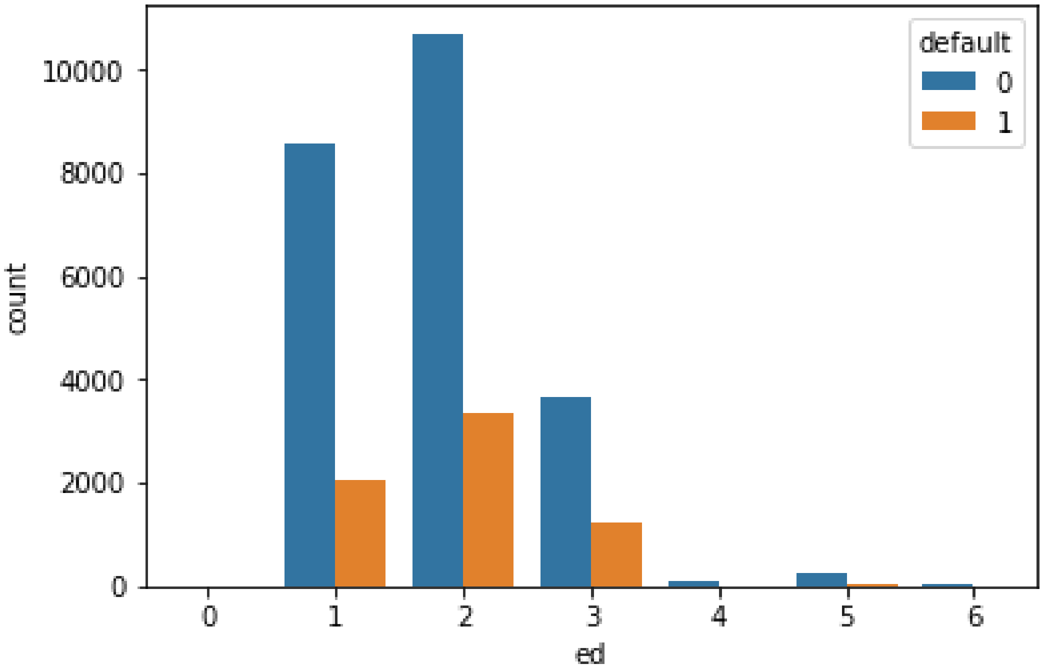
  Null value percentage: 0



- EDUCATION (ed):
  (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0

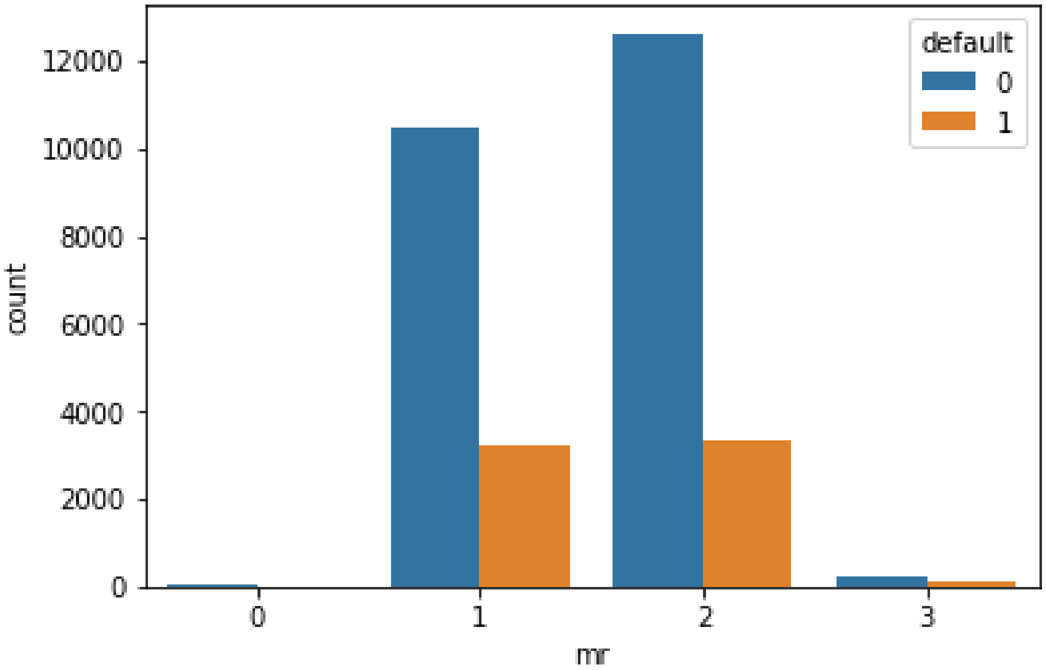


- MARRIAGE (mr): Marital status
  (1=married, 2=single, 3=others)

Data Type: Integer

Value type: Categorical

Null value percentage: 0



- AGE (age): Age in years

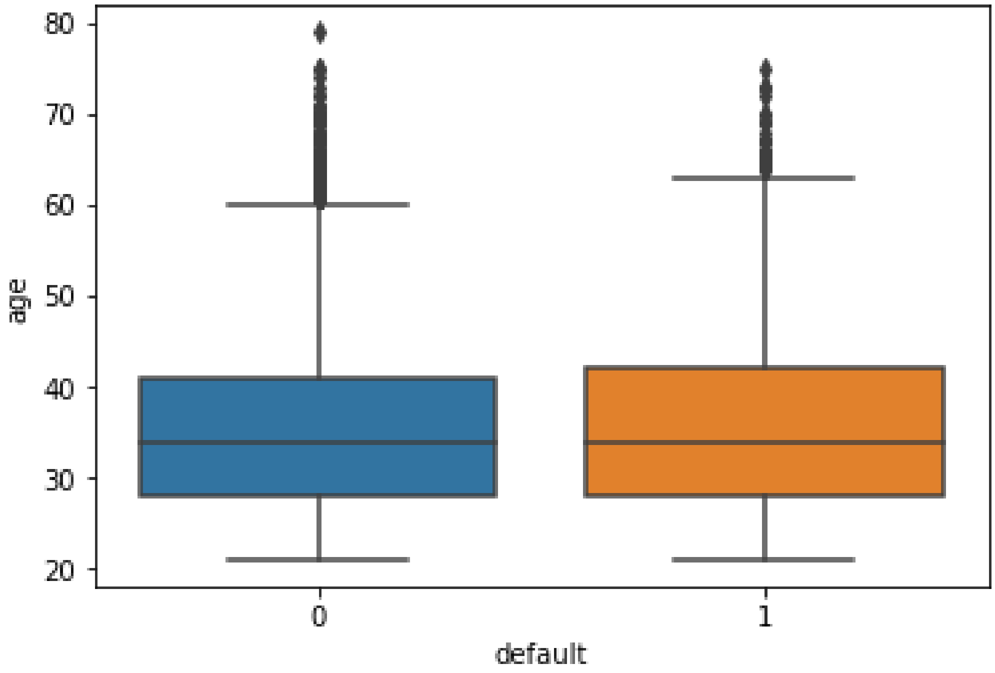
Data Type: Integer

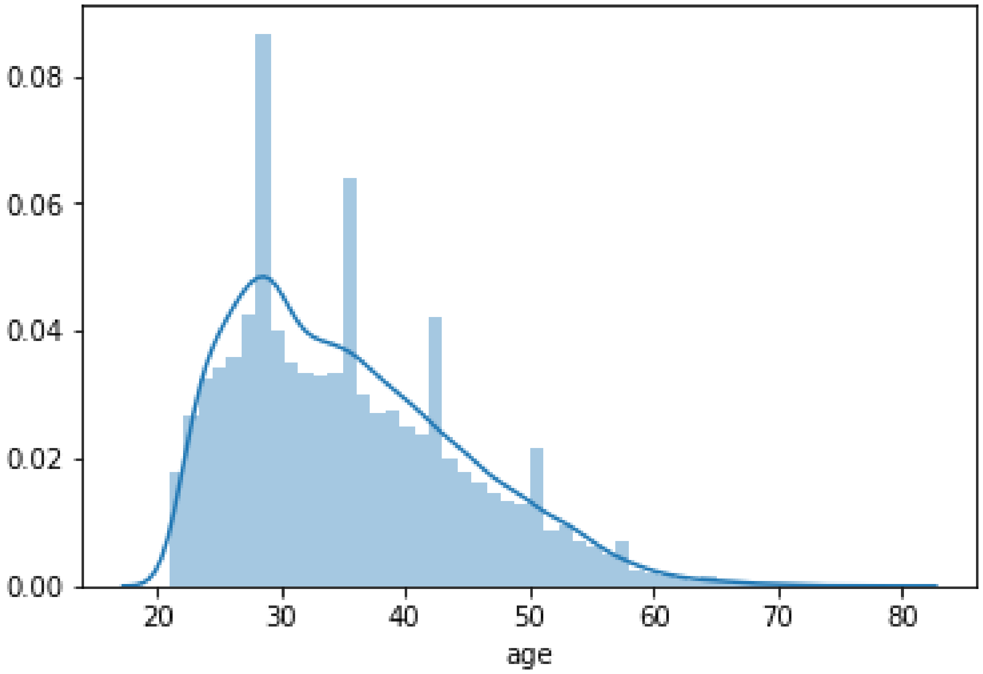
Value Type: Continuous

Null value percentage: 0

Statistics:

```
count    30000.000000
mean        35.485500
std          9.217904
min         21.000000
25%         28.000000
50%         34.000000
75%         41.000000
max         79.000000
```



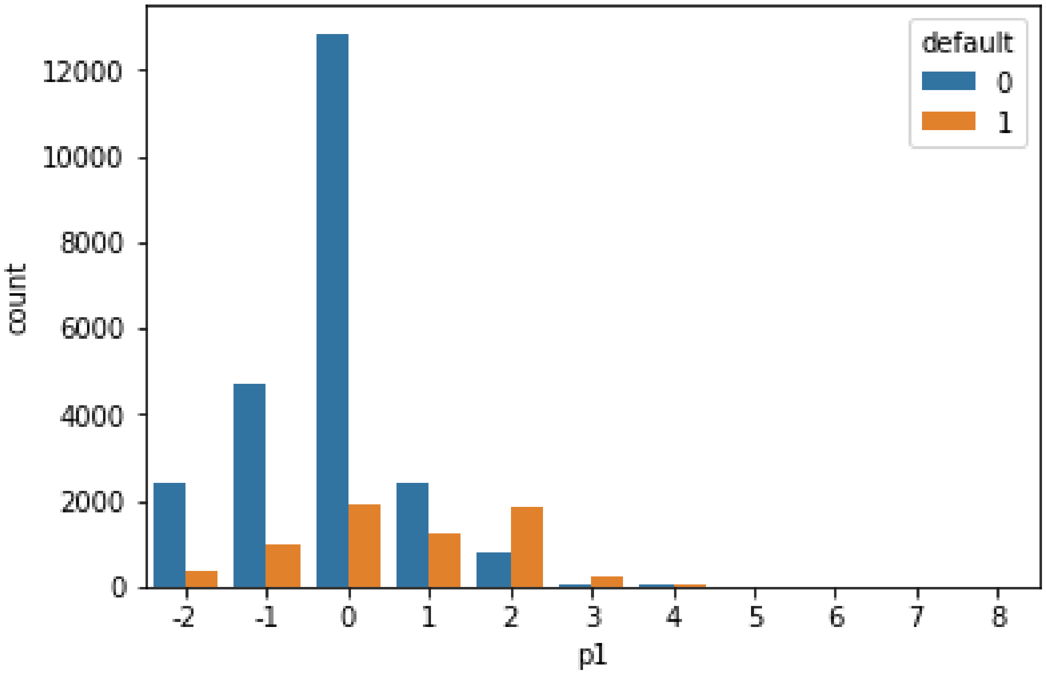


- PAY_0 (p1):
  Repayment status in September, 2005 (-1=>pay duly, 1=>payment delay for one month, 2=>payment delay for two months, ... 8=>payment delay for eight months, 9=>payment delay for nine months and above)

  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0

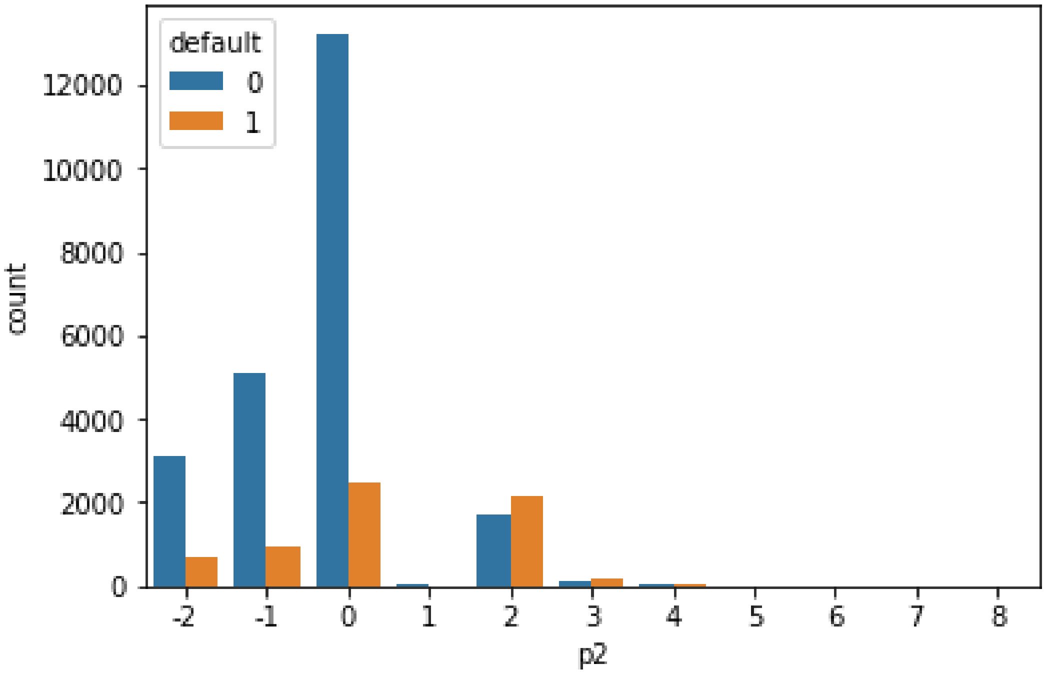


- PAY_2 (p2):
  Repayment status in August, 2005 (scale same as PAY_0)

  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0

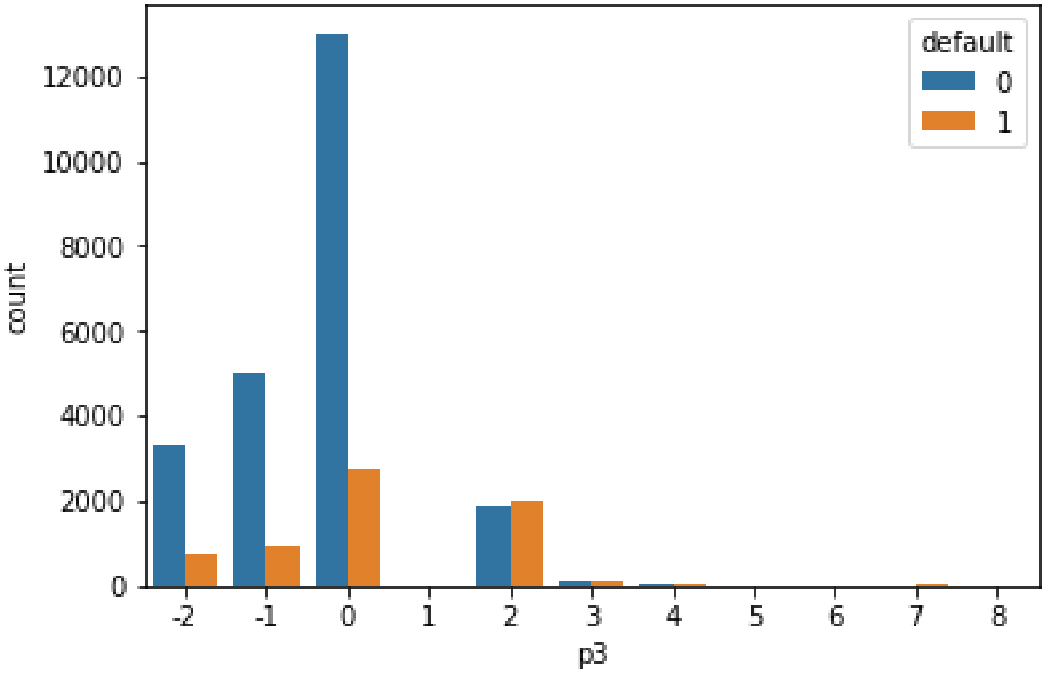


- PAY_3 (p3):
  Repayment status in July, 2005 (scale same as PAY_0)

  Data Type: Integer

Value type: Categorical

Null value percentage: 0

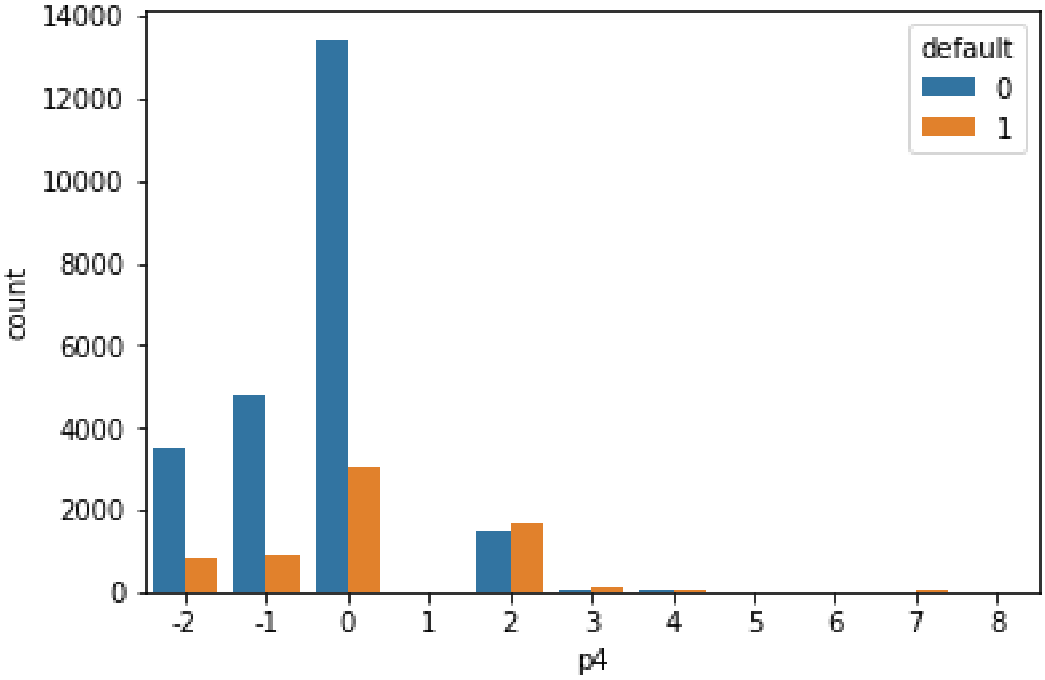


- PAY_4 (p4):
  Repayment status in June, 2005 (scale same as PAY_0)

  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0

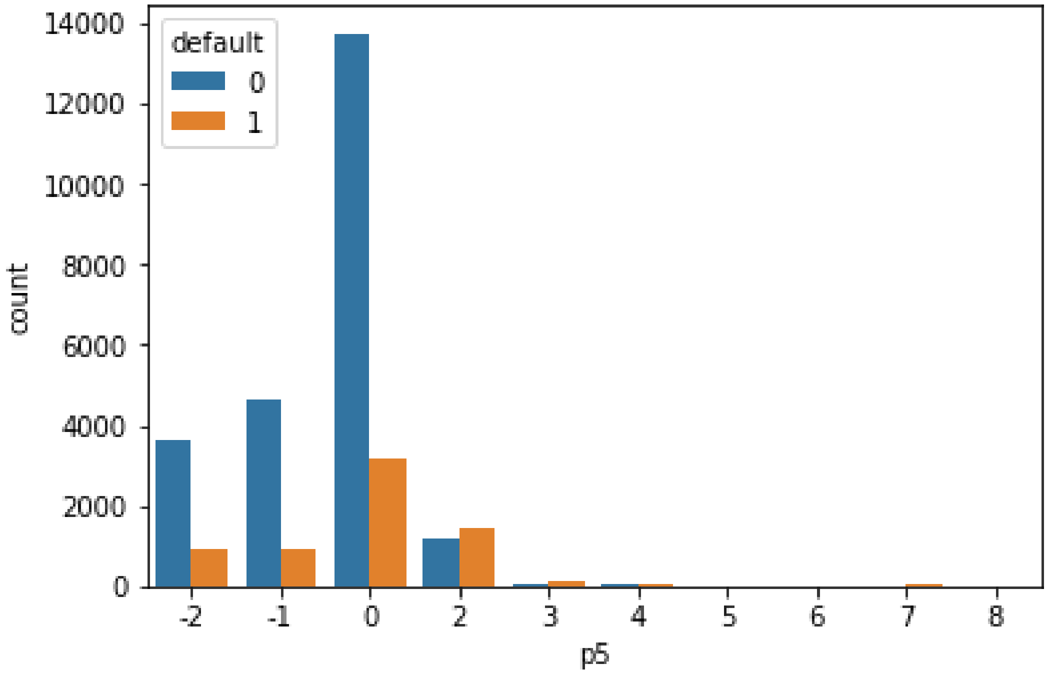


- PAY_5 (p5):
  Repayment status in May, 2005 (scale same as PAY_0)

  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0

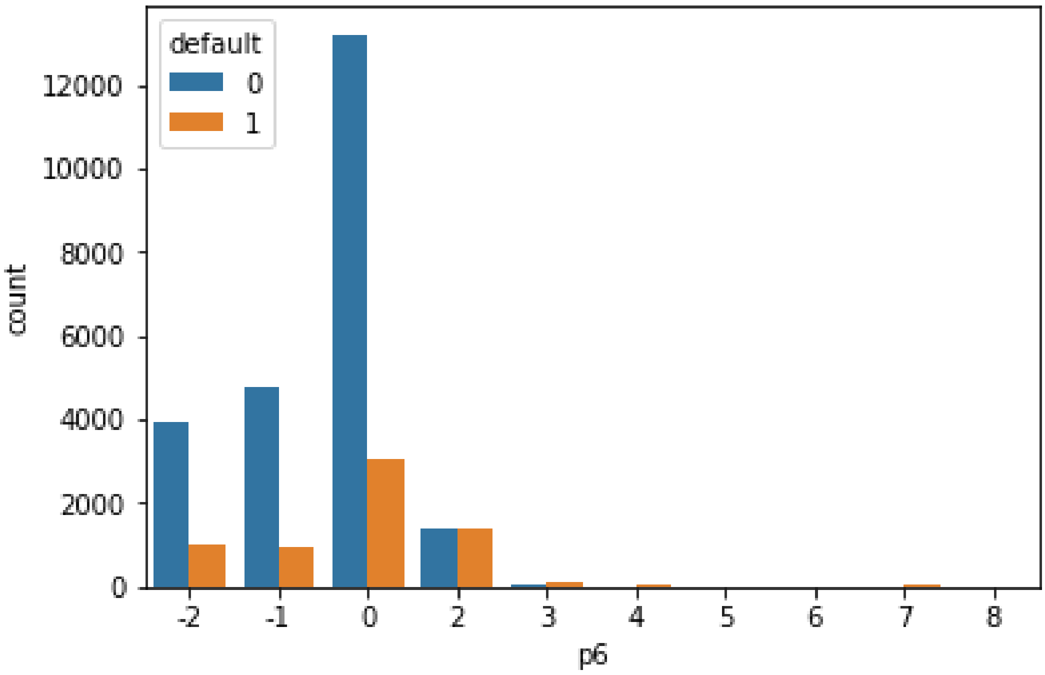


- PAY_6 (p6):
  Repayment status in April, 2005 (scale same as PAY_0)

  Data Type: Integer

  Value type: Categorical

  Null value percentage: 0



- BILL_AMT1 (ba1):
  Amount of bill statement in September, 2005 (NT dollar)

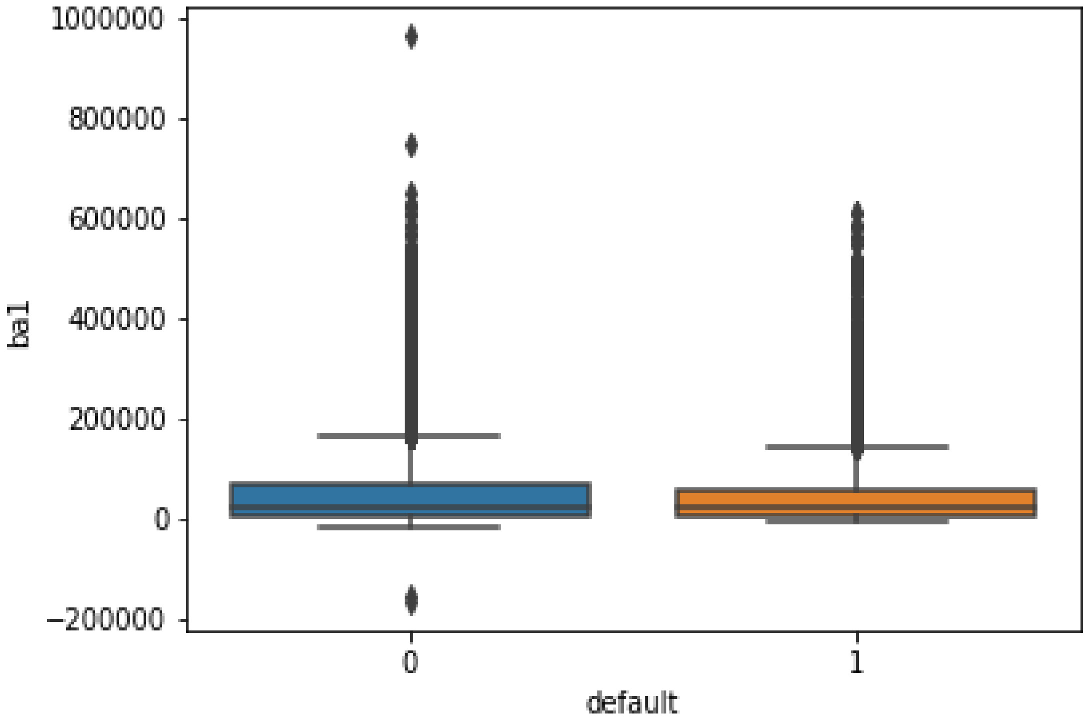
  Data Type: Integer

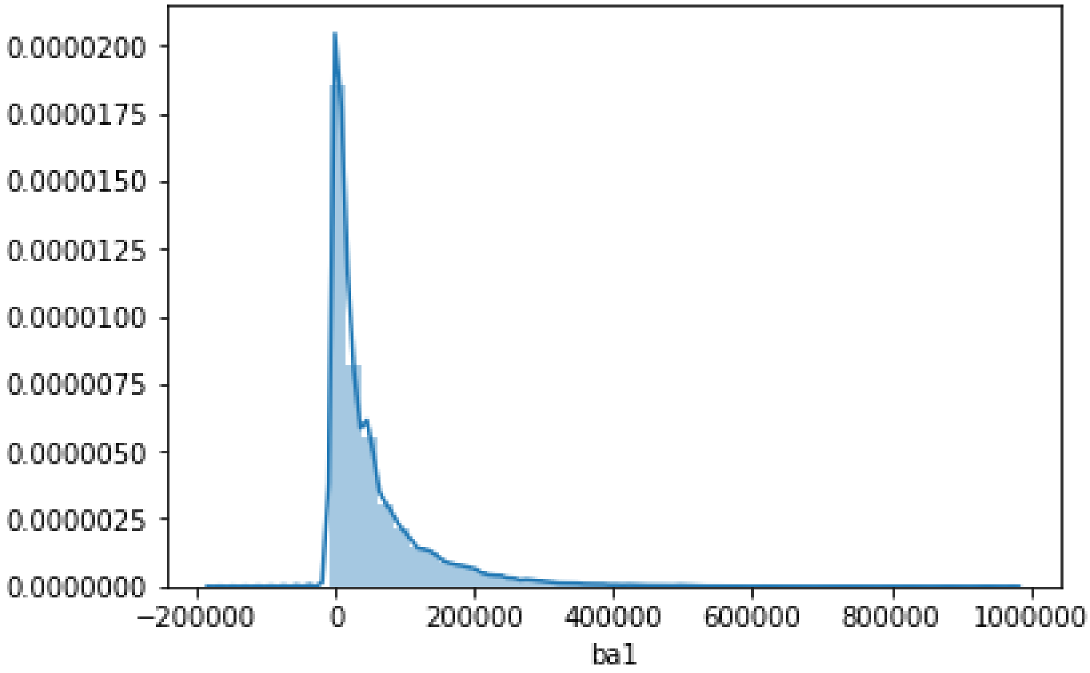
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean      51223.330900
std       73635.860576
min     -165580.000000
25%        3558.750000
50%       22381.500000
75%       67091.000000
max      964511.000000
```





- BILL_AMT2 (ba2):
  Amount of bill statement in August, 2005 (NT dollar)

  Data Type: Integer

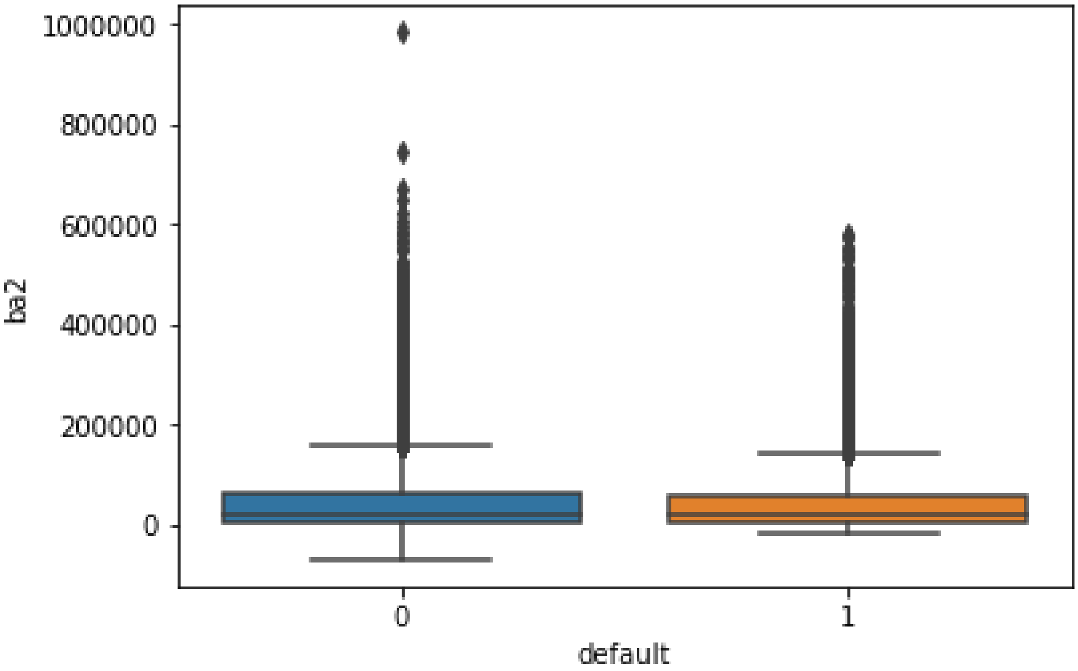
  Value Type: Continuous

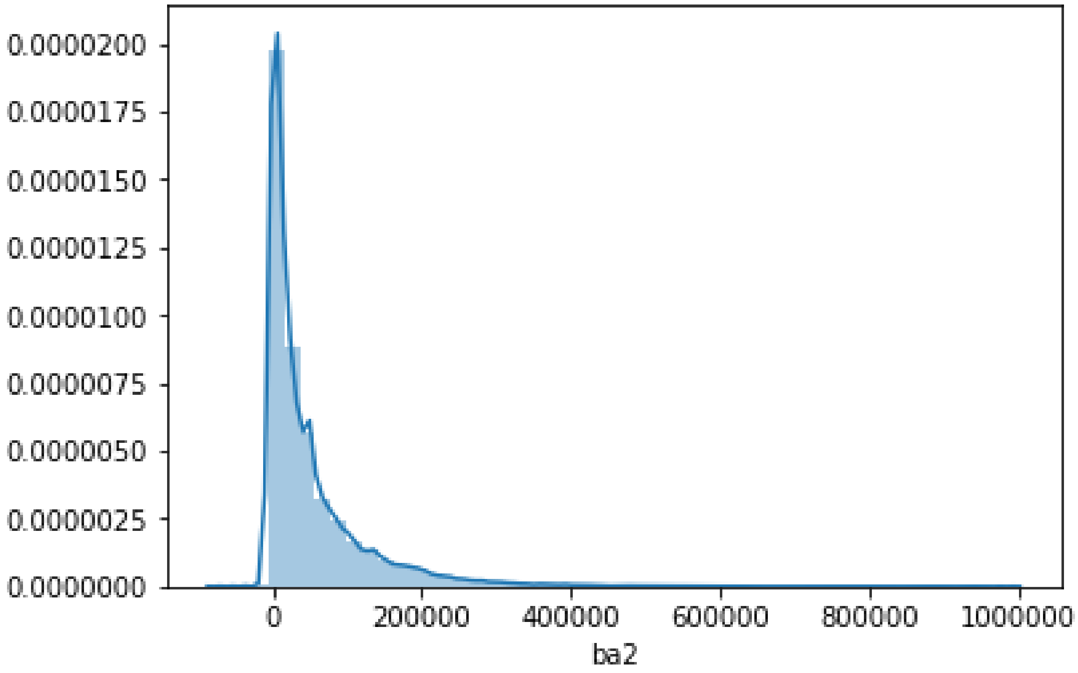
  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean      49179.075167
std       71173.768783
min      -69777.000000
25%        2984.750000
50%       21200.000000
```

```
75%       64006.250000
max      983931.000000
```





- BILL_AMT3 (ba3):
  Amount of bill statement in July, 2005 (NT dollar)

  Data Type: Integer

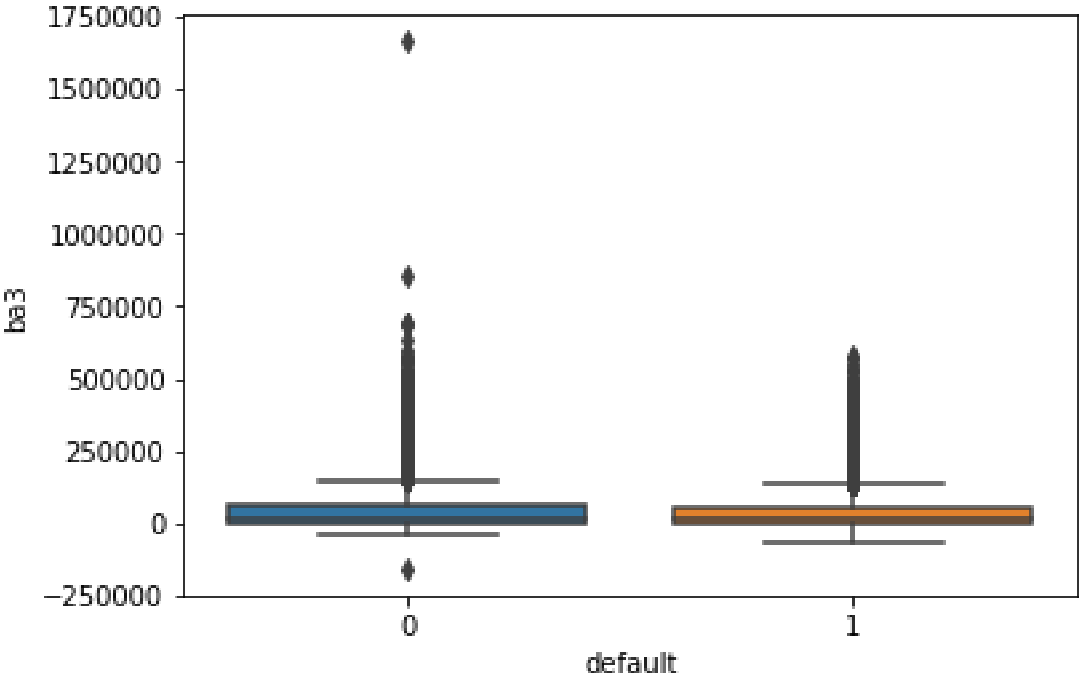
  Value Type: Continuous

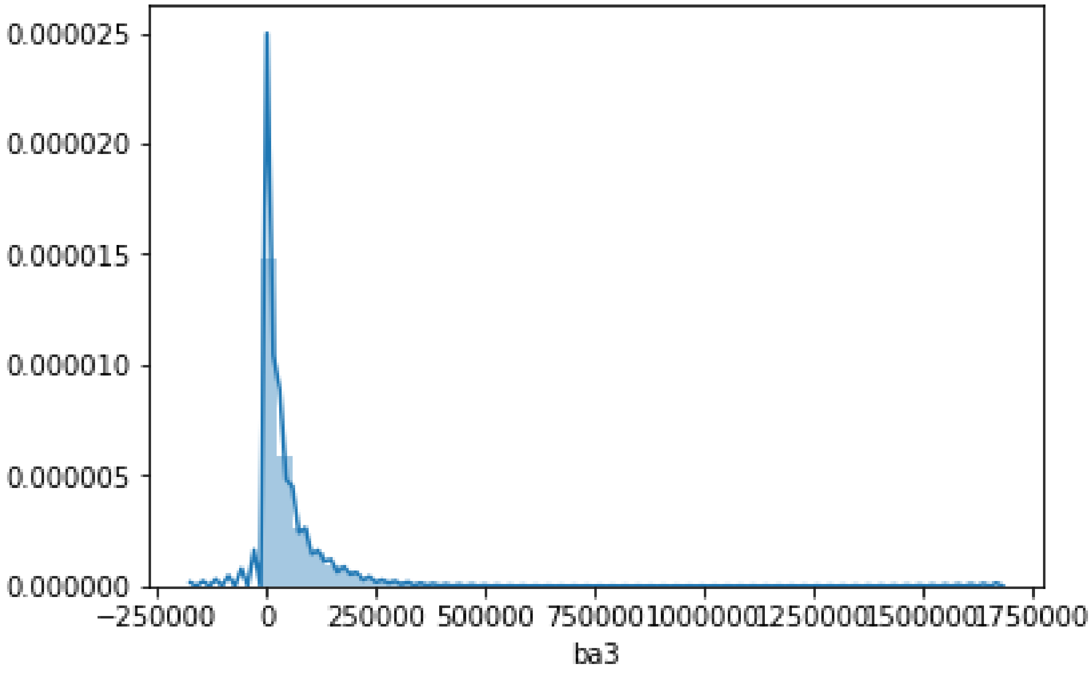
  Null value percentage: 0

  Statistics:

```
count    3.000000e+04
mean     4.701315e+04
```

```
std      6.934939e+04
min     -1.572640e+05
25%      2.666250e+03
50%      2.008850e+04
75%      6.016475e+04
max      1.664089e+06
```





- BILL_AMT4 (ba4):
  Amount of bill statement in June, 2005 (NT dollar)

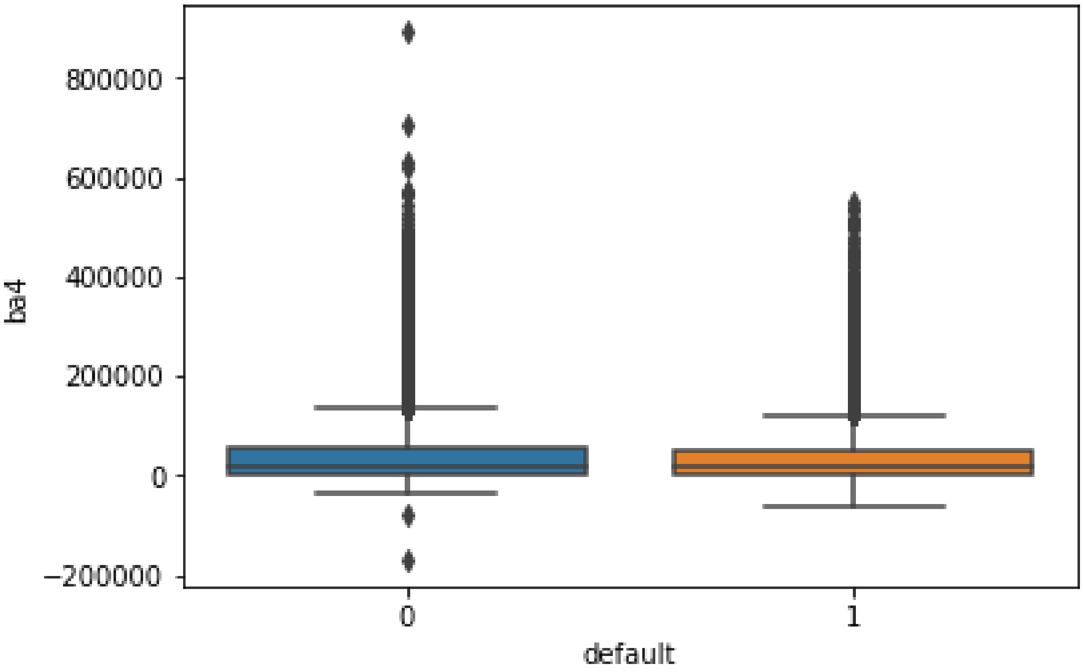
  Data Type: Integer

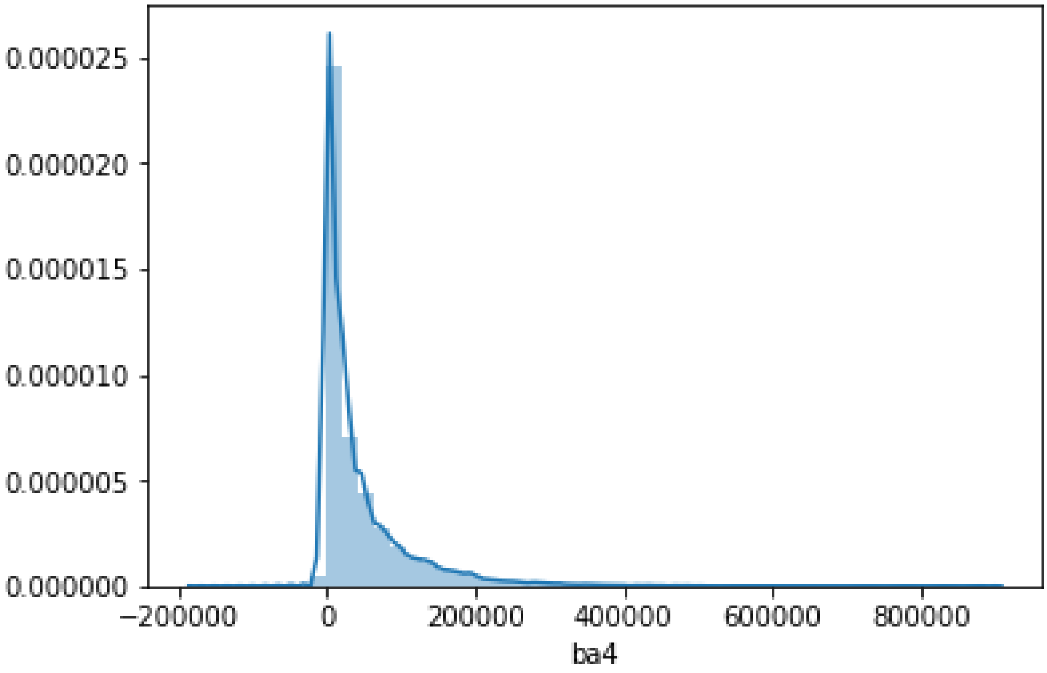
  Value Type: Continuous

Null value percentage: 0

Statistics:

```
count     30000.000000
mean      43262.948967
std       64332.856134
min     -170000.000000
25%        2326.750000
50%       19052.000000
75%       54506.000000
max      891586.000000
```





- BILL_AMT5 (ba5):
  Amount of bill statement in May, 2005 (NT dollar)

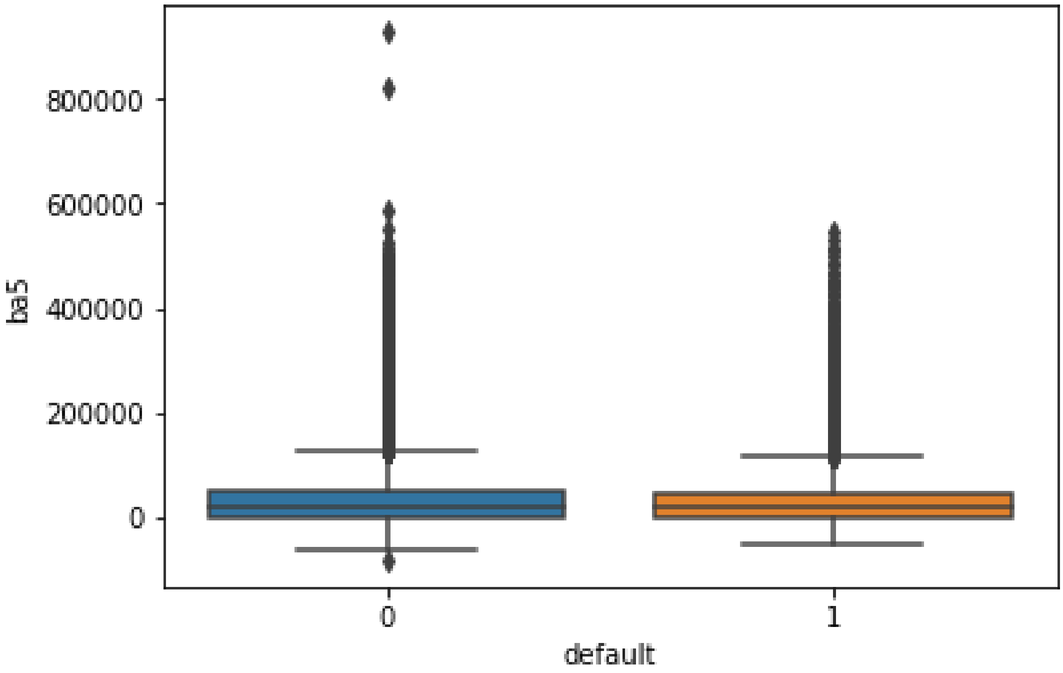
  Data Type: Integer

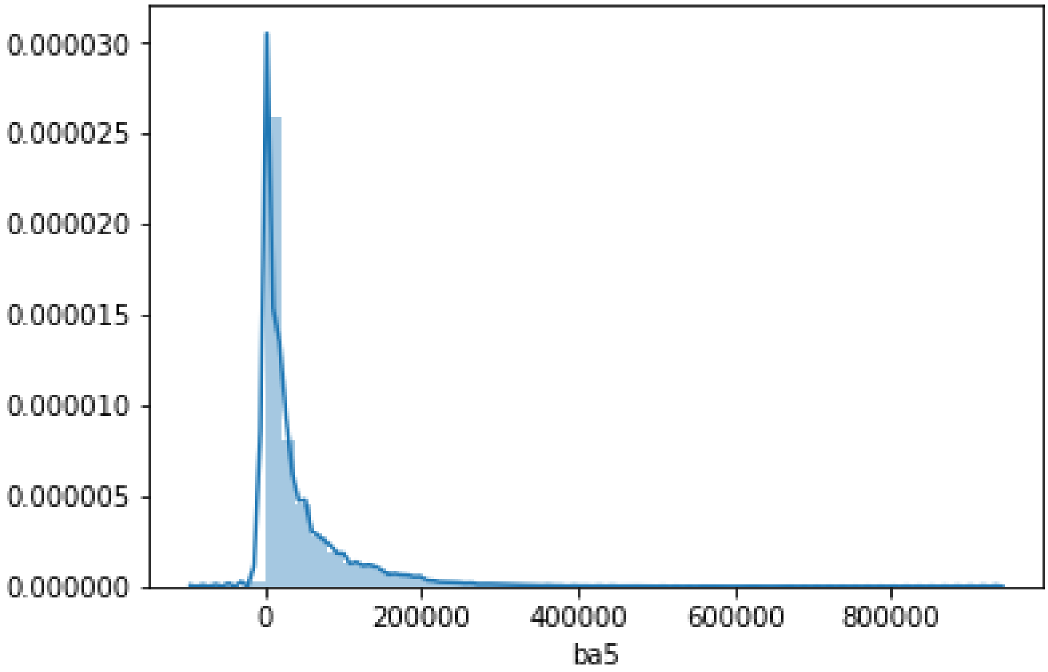
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

  ```
  count     30000.000000
  mean      40311.400967
  std       60797.155770
  min      -81334.000000
  25%        1763.000000
  50%       18104.500000
  75%       50190.500000
  max      927171.000000
  ```





- BILL_AMT6 (ba6):
  Amount of bill statement in April, 2005 (NT dollar)

  Data Type: Integer

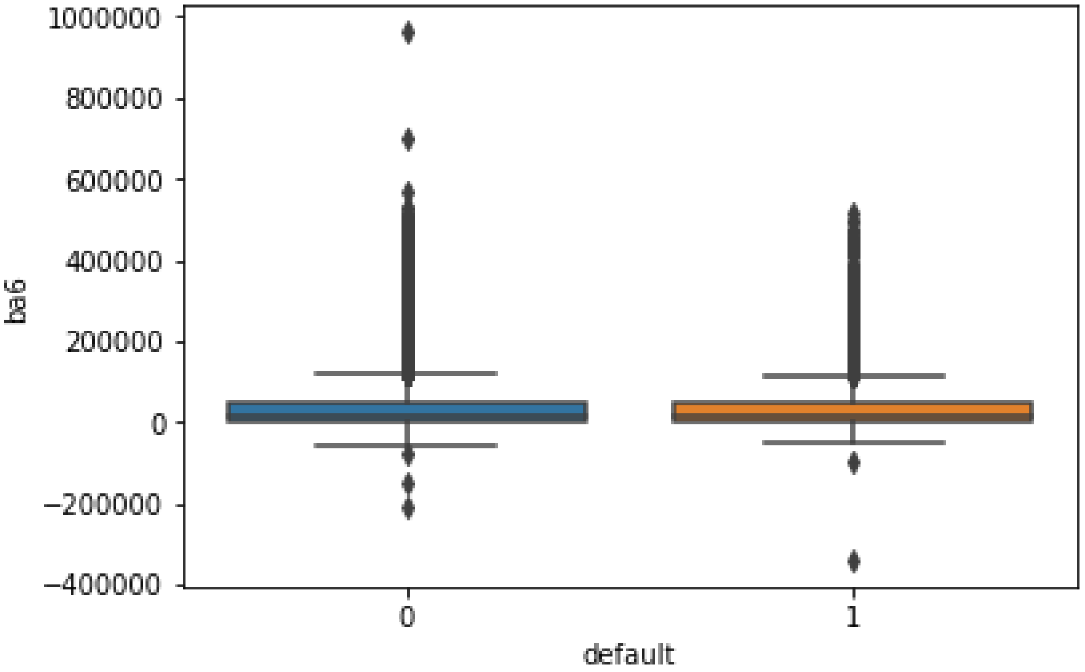
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean       38871.760400
std       59554.107537
min     -339603.000000
```

```
25%        1256.000000
50%       17071.000000
75%       49198.250000
max      961664.000000
```



- PAY_AMT1 (pa1):
  Amount of previous payment in September, 2005 (NT dollar)

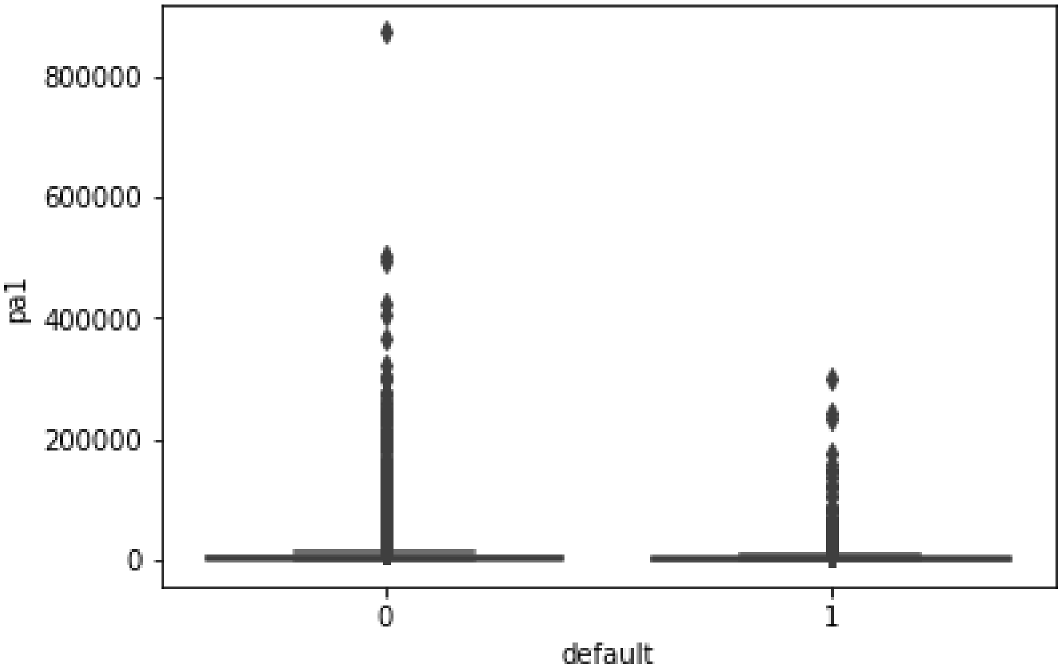
  Data Type: Integer

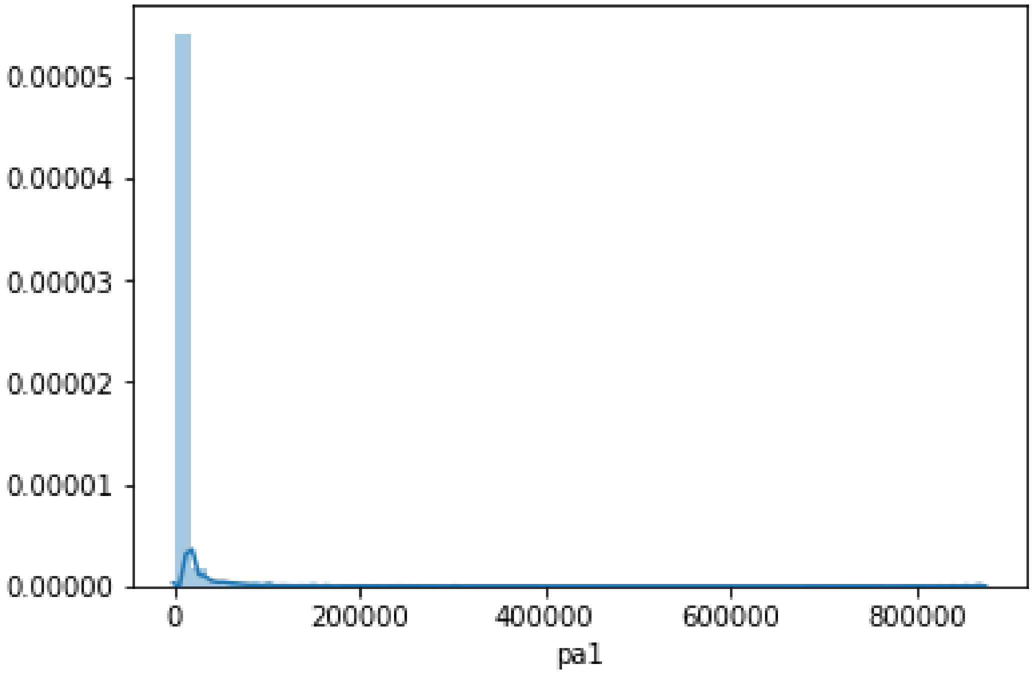
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean       5663.580500
std       16563.280354
min           0.000000
25%        1000.000000
50%        2100.000000
75%        5006.000000
max      873552.000000
```





- PAY_AMT2 (pa2):
  Amount of previous payment in August, 2005 (NT dollar)

  Data Type: Integer

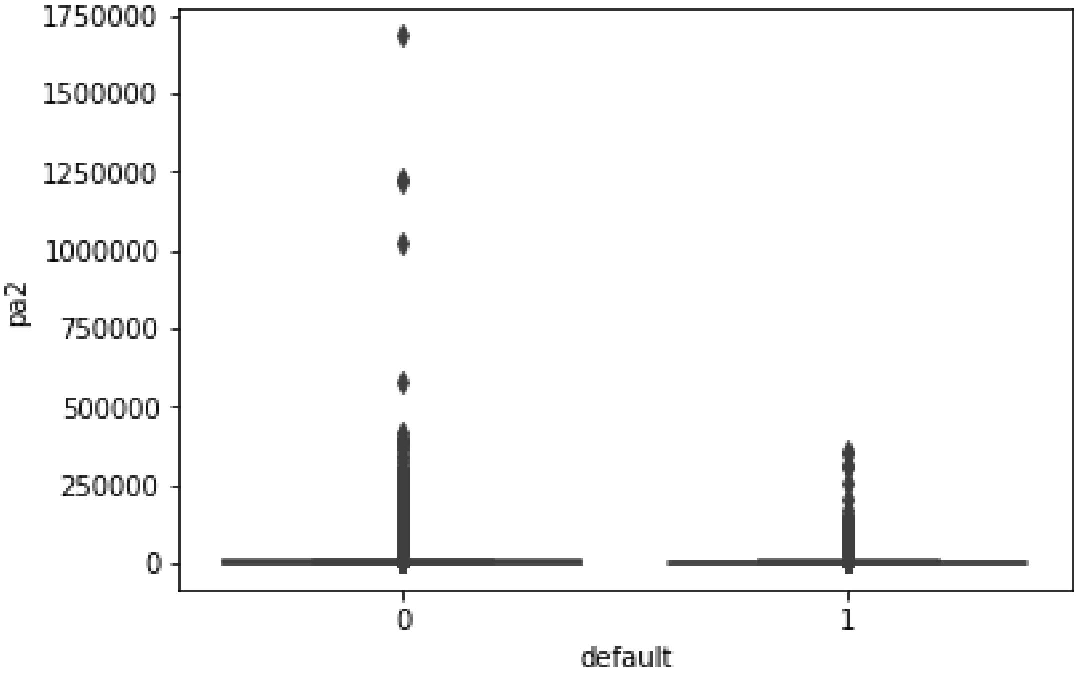
  Value Type: Continuous

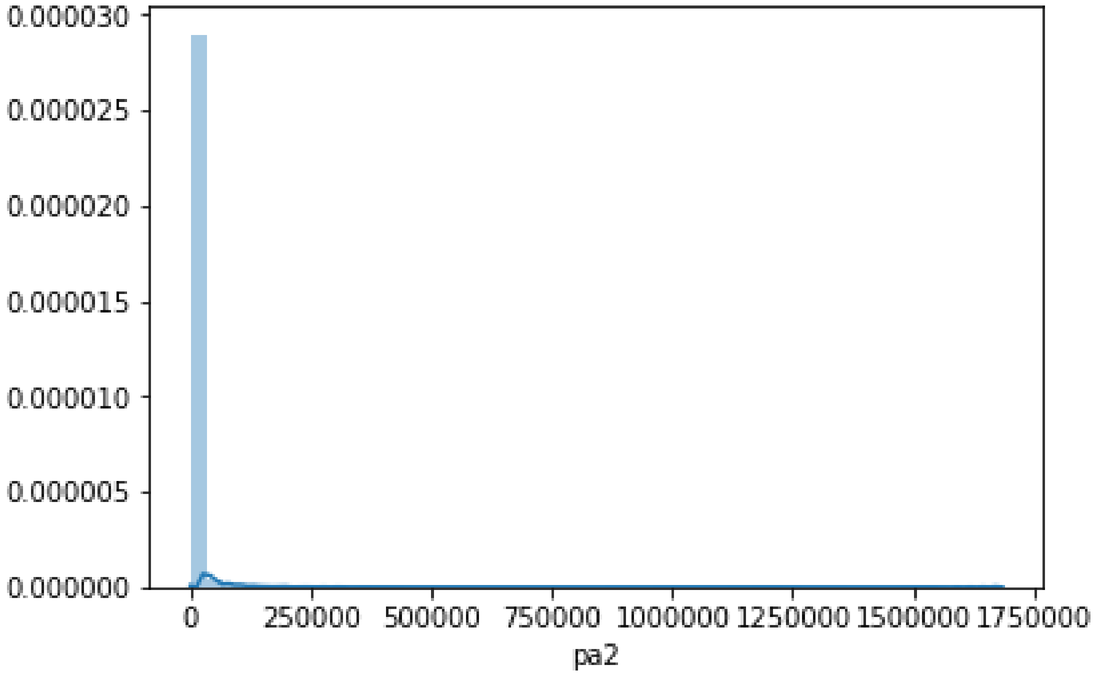
  Null value percentage: 0

  Statistics:

```
count    3.000000e+04
mean     5.921163e+03
std      2.304087e+04
min      0.000000e+00
```

```
25%     8.330000e+02
50%     2.009000e+03
75%     5.000000e+03
max     1.684259e+06
```





- PAY_AMT3 (pa3):

  Amount of previous payment in July, 2005 (NT dollar)

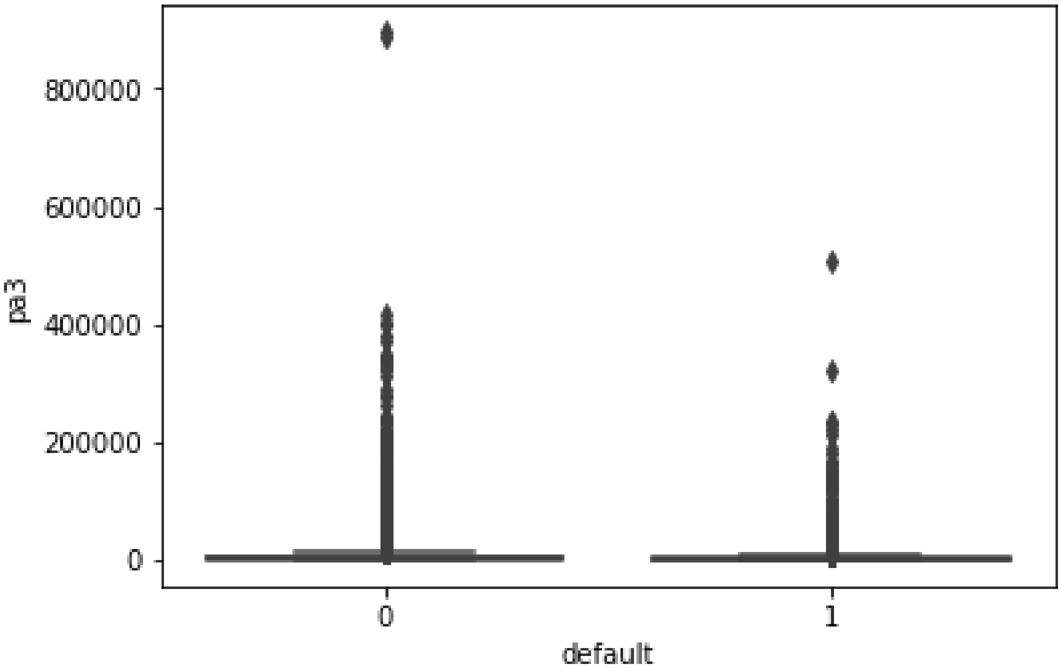
  Data Type: Integer

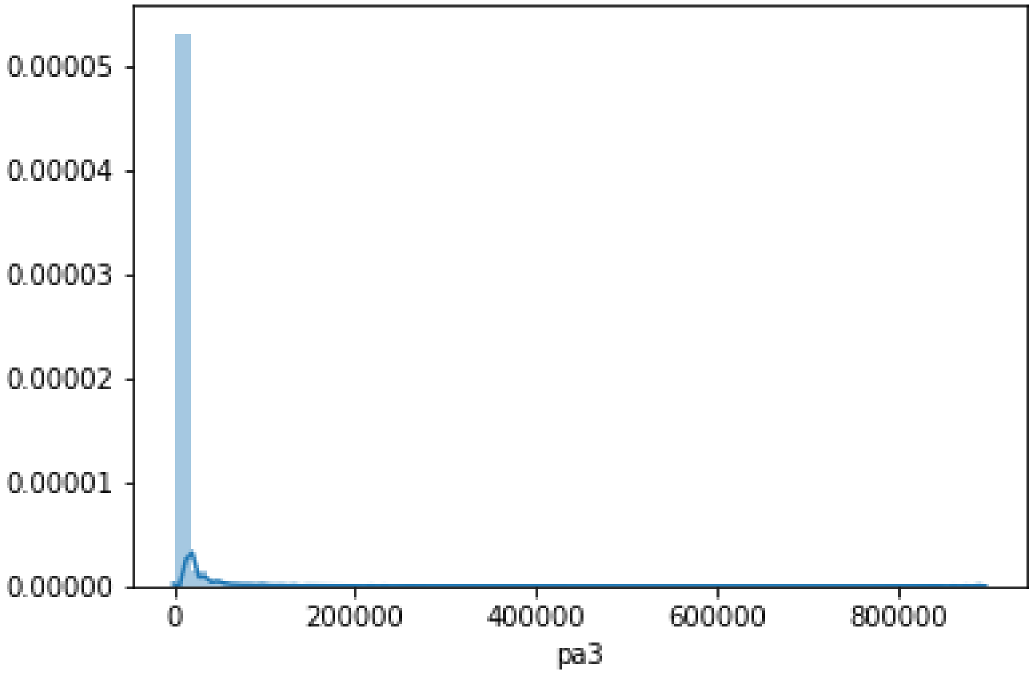
  Value Type: Continuous

Null value percentage: 0

Statistics:

```
count     30000.00000
mean       5225.68150
std       17606.96147
min           0.00000
25%         390.00000
50%        1800.00000
75%        4505.00000
max      896040.00000
```





- PAY_AMT4 (pa4):
  Amount of previous payment in June, 2005 (NT dollar)

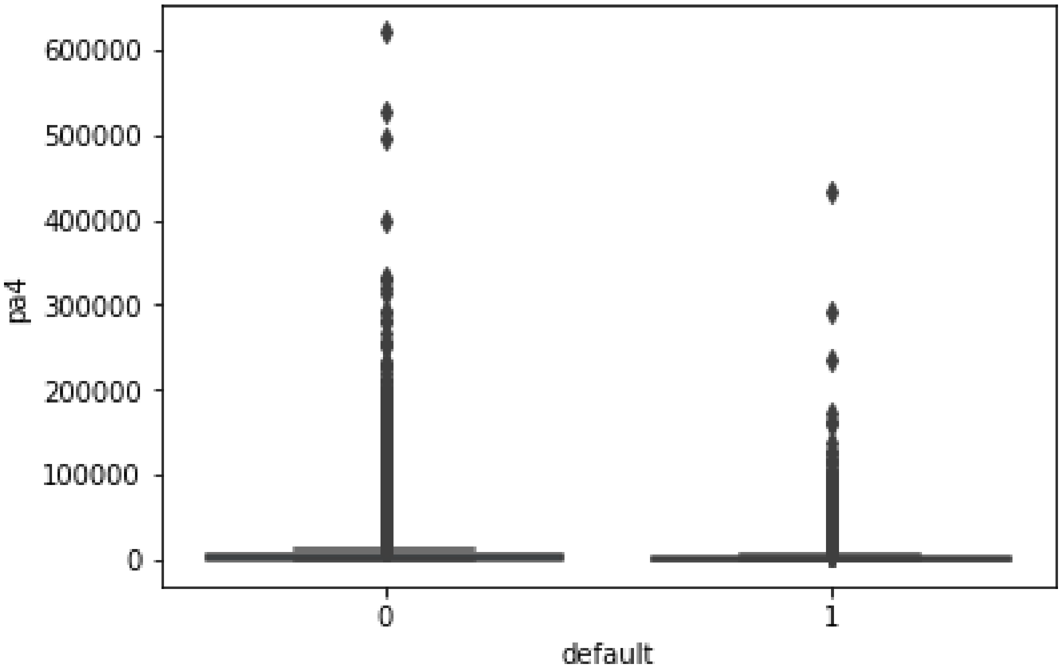
  Data Type: Integer

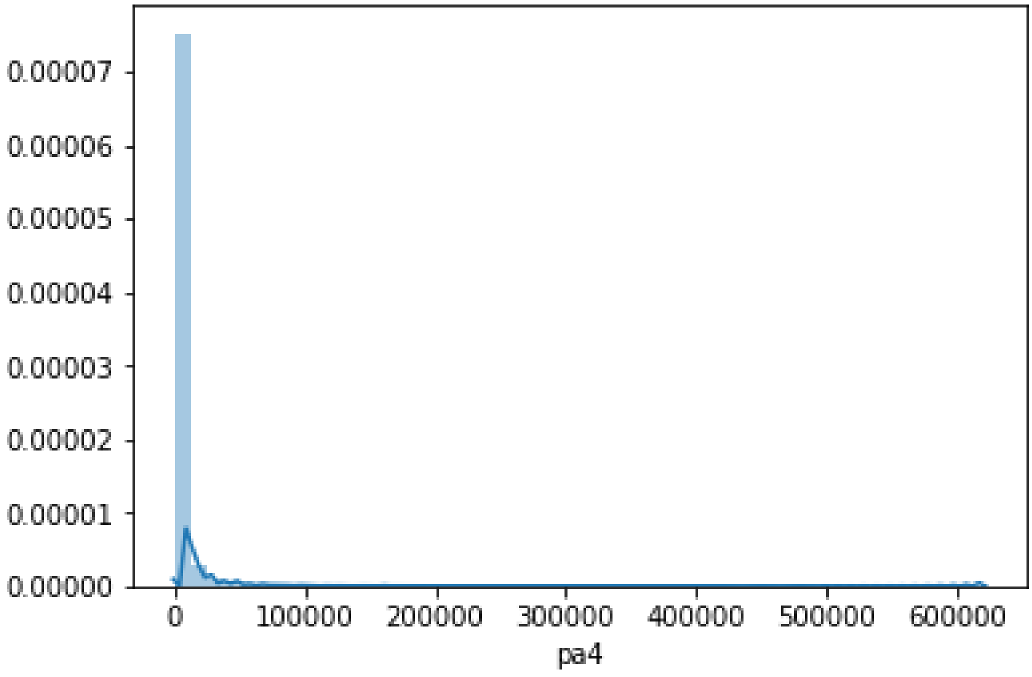
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean       4826.076867
std       15666.159744
min           0.000000
25%         296.000000
50%        1500.000000
75%        4013.250000
max      621000.000000
```





- PAY_AMT5 (pa5):

  Amount of previous payment in May, 2005 (NT dollar)

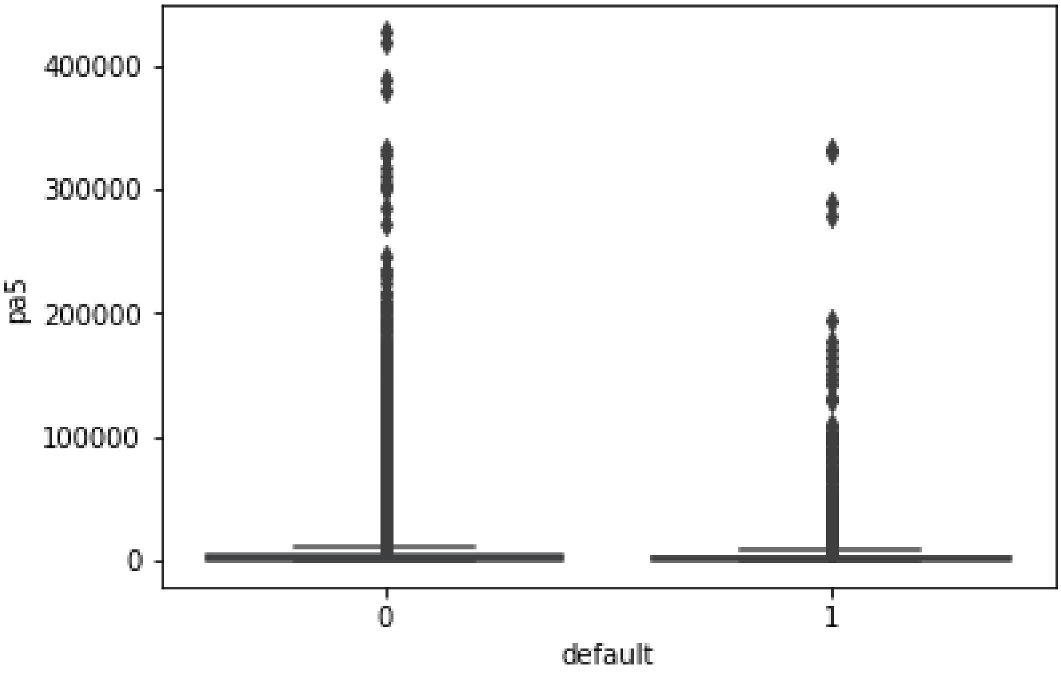
  Data Type: Integer

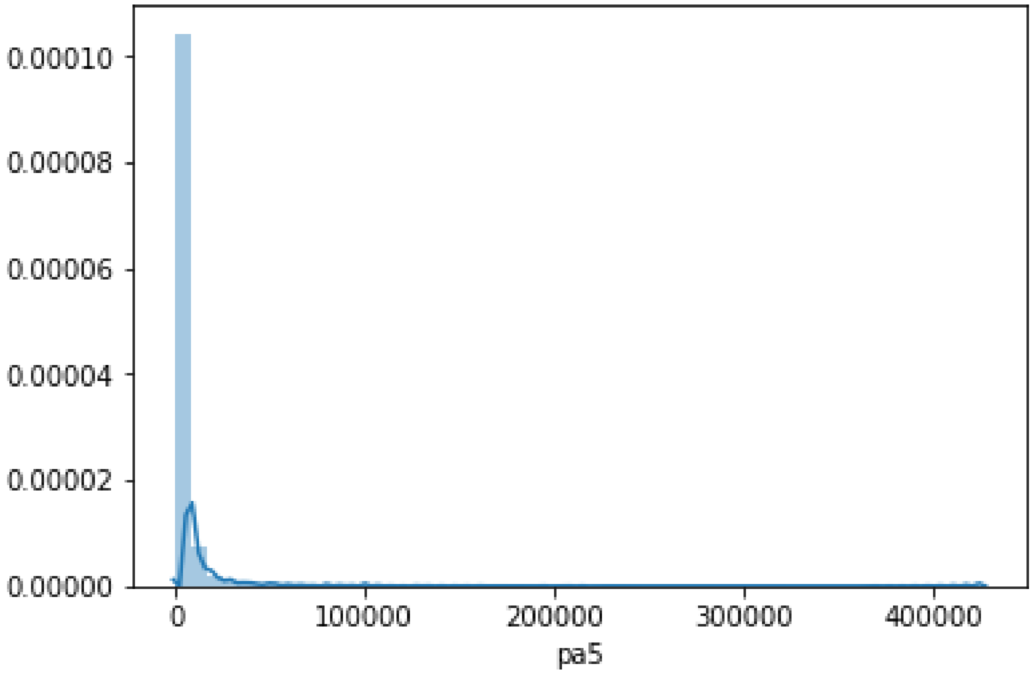
  Value Type: Continuous

  Null value percentage: 0

  Statistics:

```
count     30000.000000
mean       4799.387633
std       15278.305679
min           0.000000
25%         252.500000
50%        1500.000000
75%        4031.500000
max      426529.000000
```





- PAY_AMT6 (pa6):
  Amount of previous payment in April, 2005 (NT dollar)

  Data Type: Integer

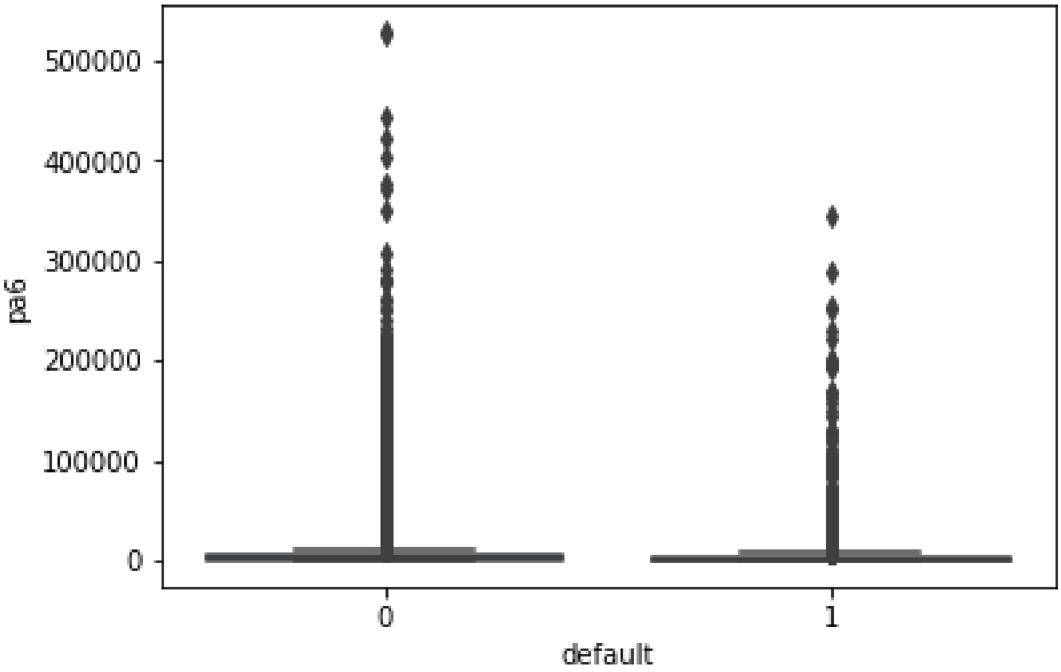
  Value Type: Continuous

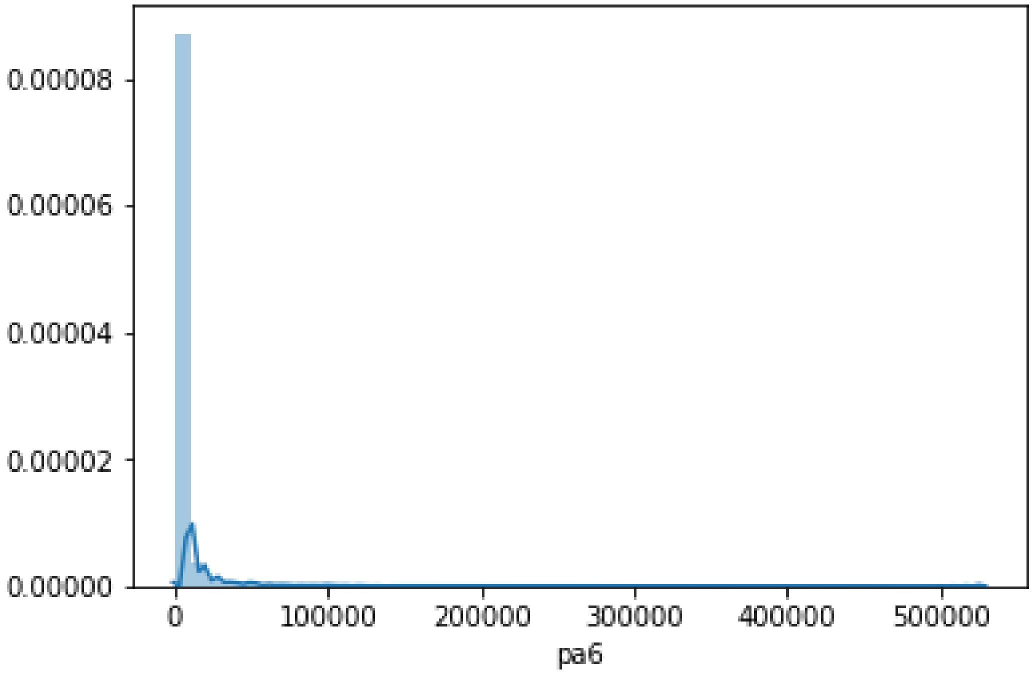
  Null value percentage: 0

  Statistics:

```
count    30000.000000
mean      5215.502567
std      17777.465775
min          0.000000
```

```
25%         117.750000
50%        1500.000000
75%        4000.000000
max      528666.000000
```



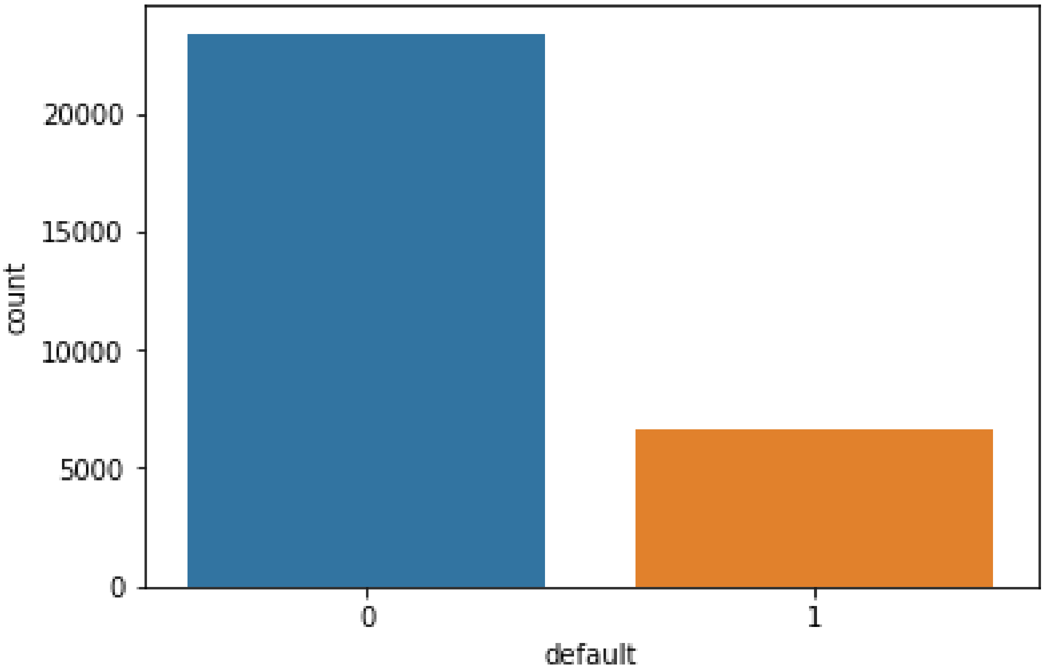


- default. payment. next. Month (default):
  Default payment (1=yes, 0=no)

  Data Type: Integer

  Value Type: Continuous

Null value percentage: 0

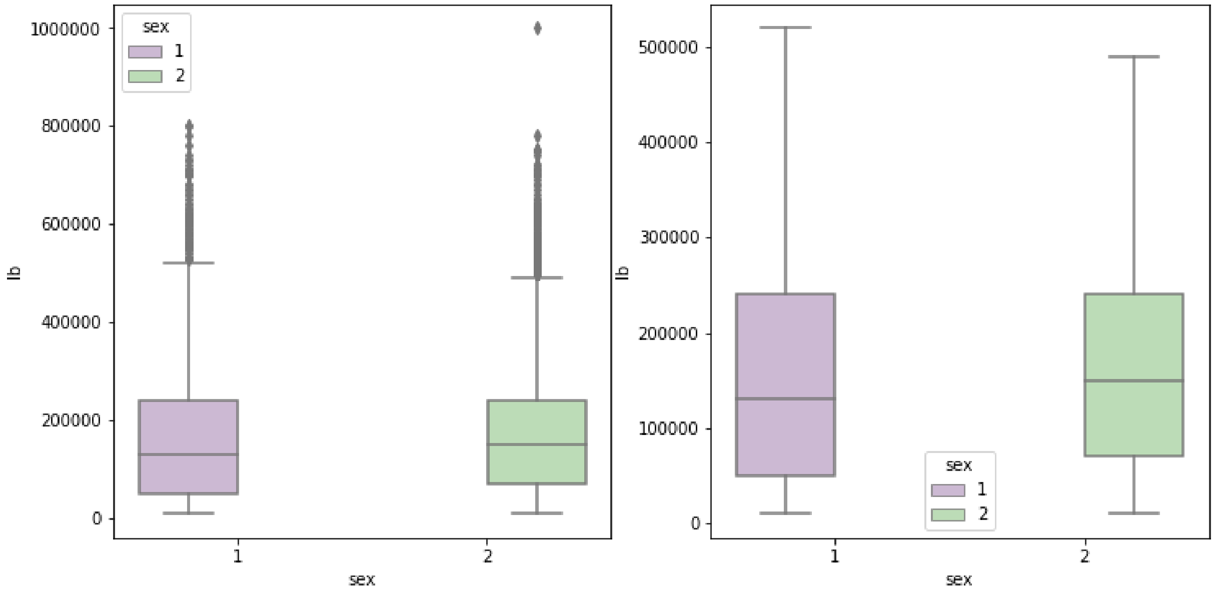


DEFAULT	count	Percentage
0	23364	0.78
1	6636	0.22

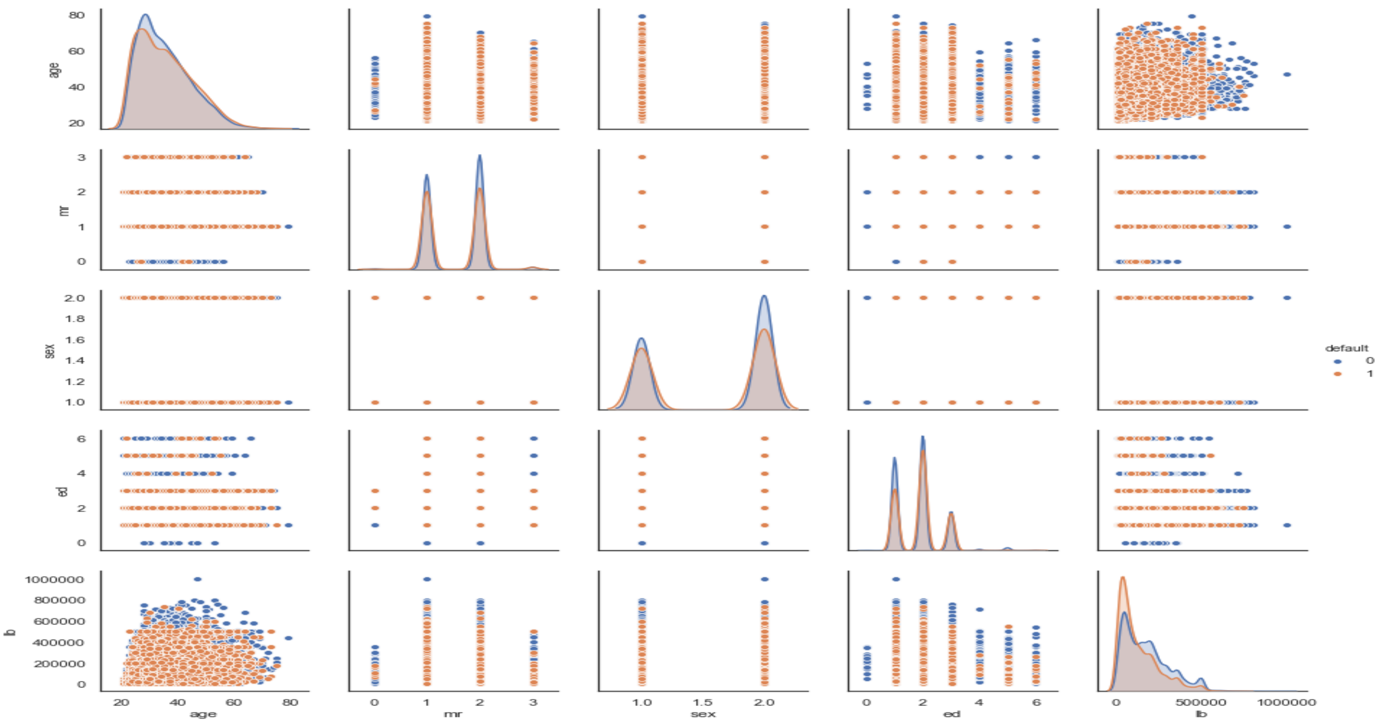
>>>>DATA VISUALISATION

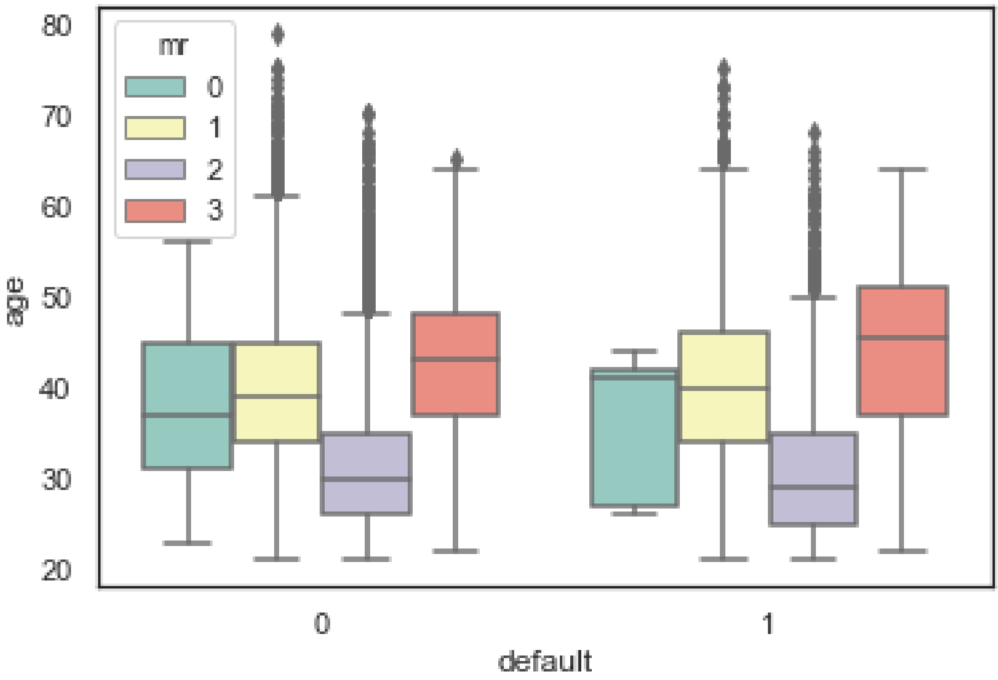
Credit limit vs. sex

Let's check the credit limit distribution vs. sex. For the sex, 1 stands for male and 2 for female.



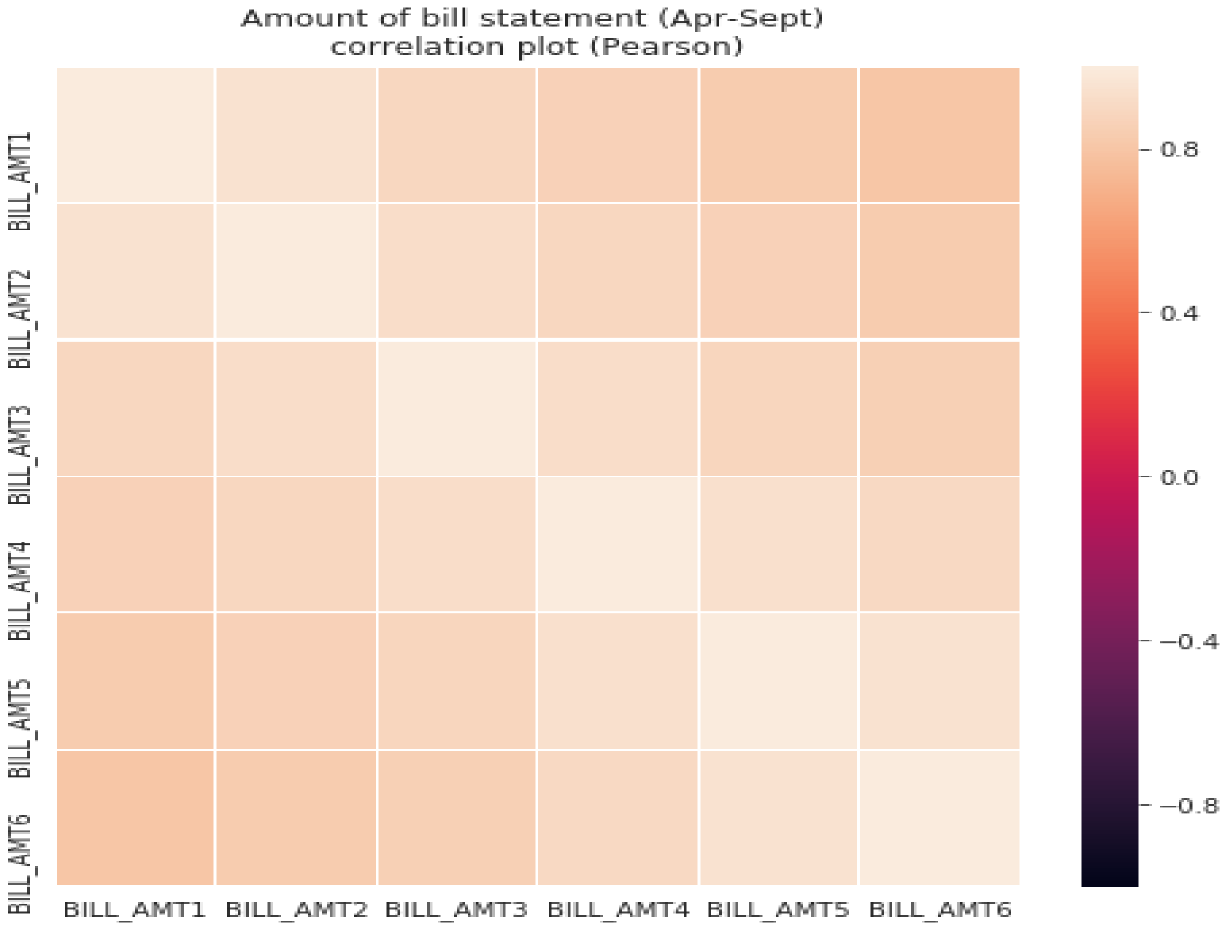
```
>>>sns.pairplot(dataset, hue = 'default', vars = ['age', 'mr’, ’age', 'sex', 'ed', 'lb'] )
```



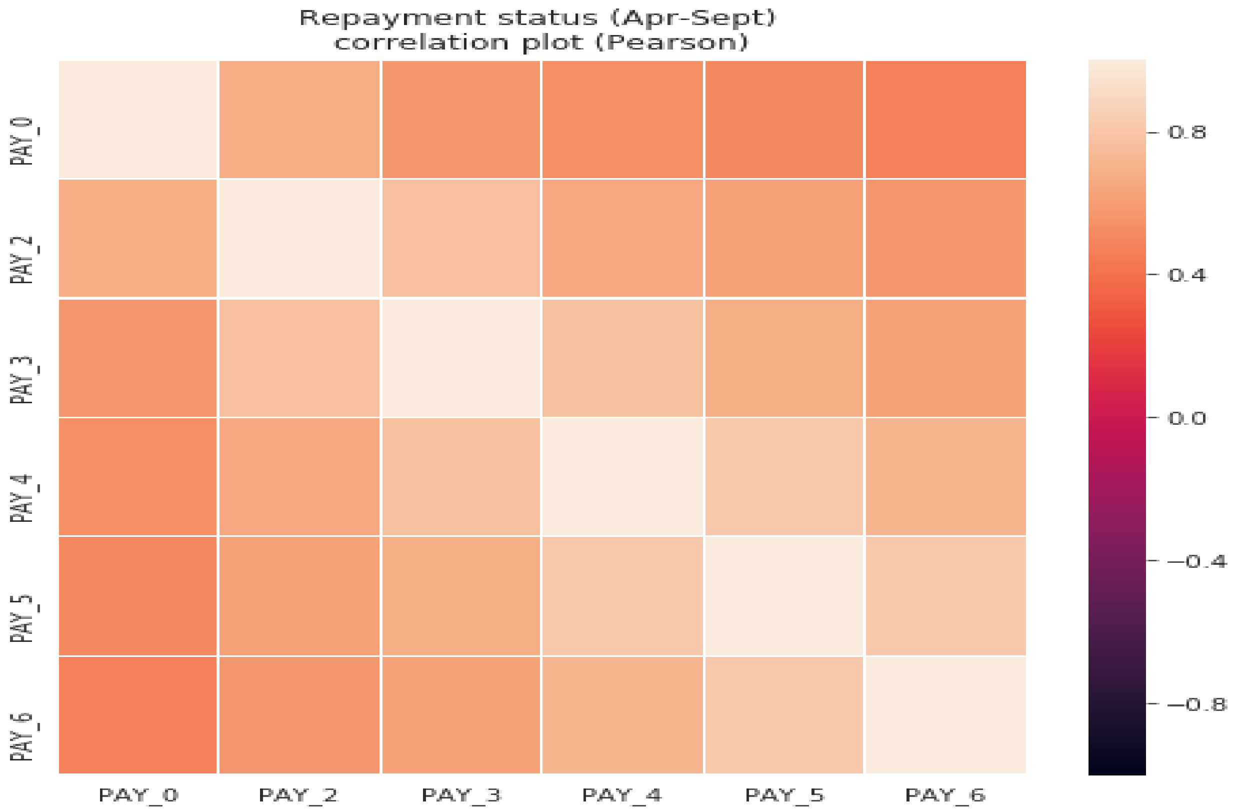


>>>> sns.boxplot(x='default', hue='mr', y='age', data=df, palette="Set3")

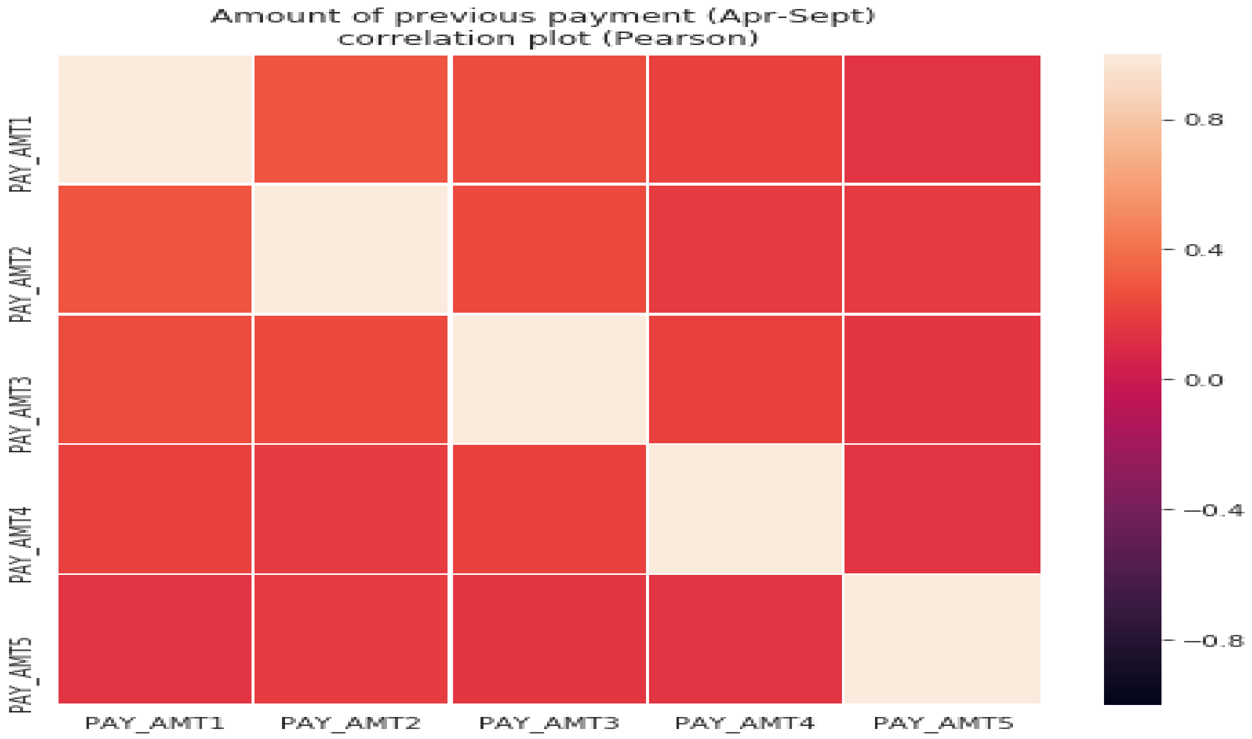
>>> co relation between bill statement



>>>> co relation between payment month



>>>>>> co relation between payment amount



MODEL BUIL DING

>>>>How we are planning to approach??

Since we want to find out each and every defaulter, we will be looking out for **Recall Score based model**.

Analyse the dataframe.

Remove irregularities in the data.

In MARRIAGE column 0s were replaced by 3 as all of them meant the same.
In EDUCATION column 0s and 6s were replaced by 5 as all of them meant the same.
In PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6 columns -1, -2 were replaced by 0 which means “duly paid”.

Drop the columns which have high multi collinearity and low contribution towards ‘label’.

Apply different classification models and note down the scores for each models.

Select the models of the best scores.

LOGISTIC REGRESSION

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

To predict credit card, default logistic regression analysis model is another popular technique that is based on supervised learning model. In this model a data set of past observation is used to see future possibilities of defaults.

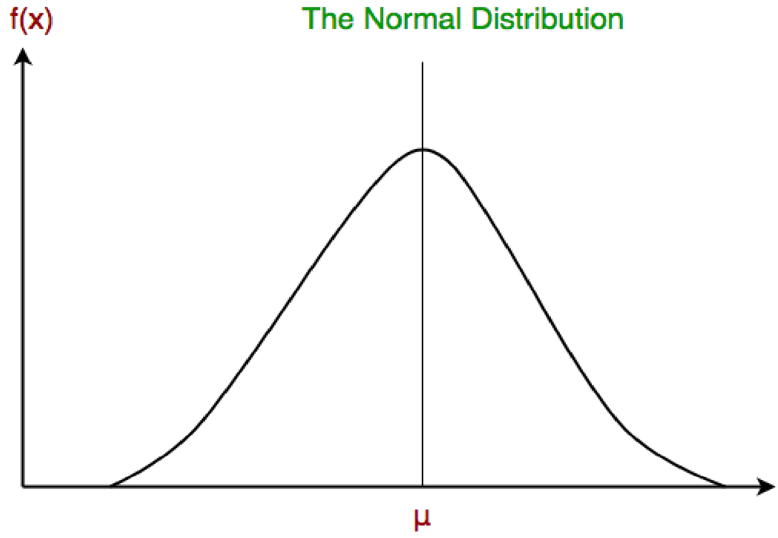
NAIVE BAYES MODEL

Naive Bayes classifier is a probabilistic classifier based on the Bayes theorem that assumes that the attributes are independent of each other. Algorithm Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

There are three types of Naïve Bayes Models used:

Gaussian Naive Bayes classifier

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:



Multinomial Naive Bayes

Feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution. This is the event model typically used for document classification.

Bernoulli Naive Bayes

In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence (i.e. a word occurs in a document or not) features are used rather than term frequencies (i.e. frequency of a word in the document).

DECISION TREE

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Construction of Decision Tree

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

Decision Tree Representation

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.
The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf. (in this case Yes or No).

Strengths and Weakness of Decision Tree approach

The strengths of decision tree methods are:

- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision tree methods:

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

RANDOM FOREST

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

K-NEAREST NEIGHBOURS

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

Intuition

If we plot these points on a graph, we may be able to locate some clusters, or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbours belong to. This means, a point close to a cluster of points classified as ‘Red’ has a higher probability of getting classified as ‘Red’.

Intuitively, we can see that the first point (2.5, 7) should be classified as ‘Green’ and the second point (5.5, 4.5) should be classified as ‘Red’.

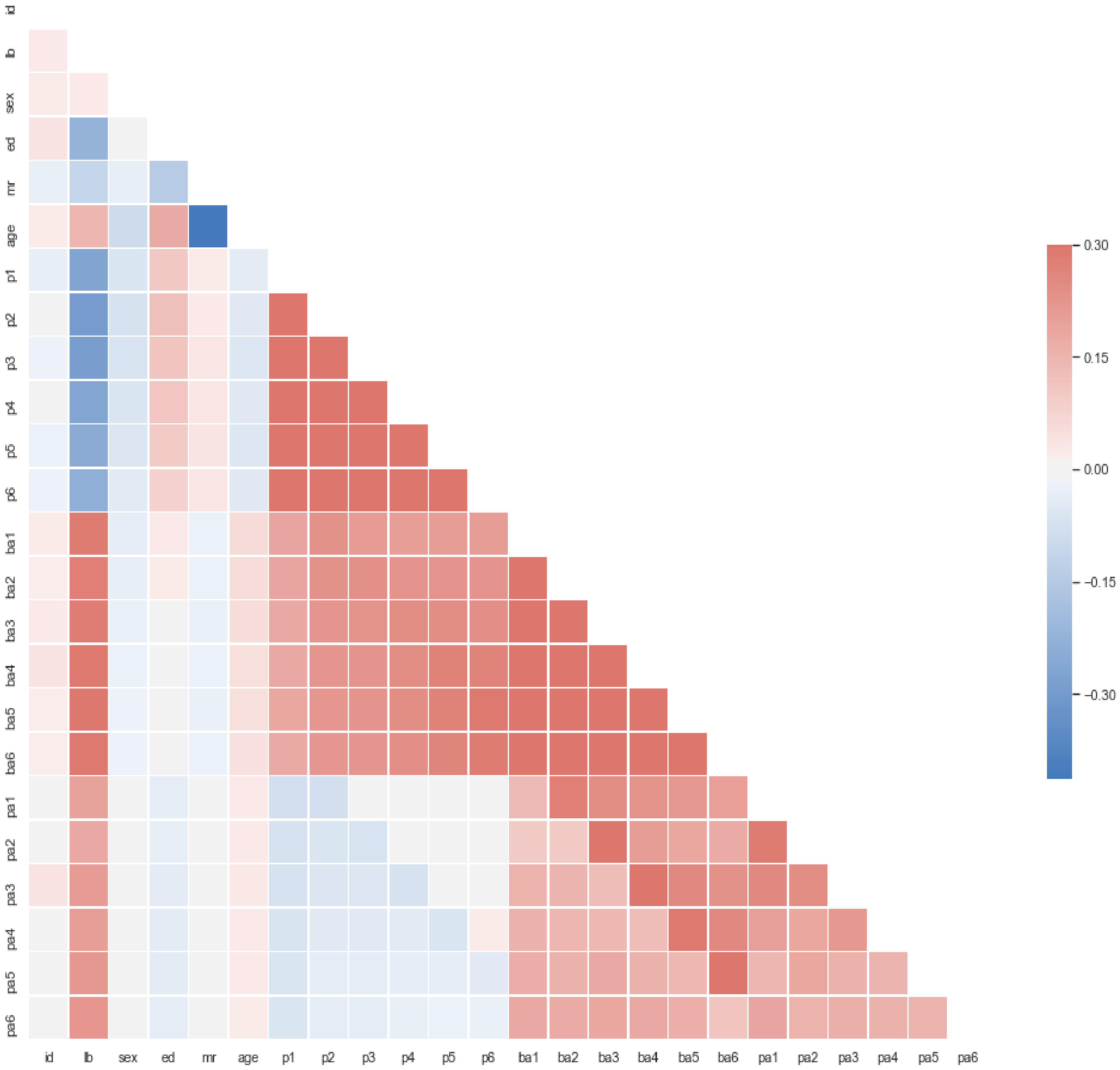
Algorithm

Let m be the number of training data samples. Let p be an unknown point.

1. Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
2. for i=0 to m:
3.   Calculate Euclidean distance d(arr[i], p).
4. Make set S of K smallest distances obtained. Each of these distances correspond to an already classified data point.
5. Return the majority label among S.

K can be kept as an odd number so that we can calculate a clear majority in the case where only two groups are possible (e.g. Red/Blue). With increasing K, we get smoother, more defined boundaries across different classifications. Also, the accuracy of the above classifier increases as we increase the number of data points in the training set.

HEATMAP


id
lb
sex
ed
mr
age
p1
p2
p3
p4
p5
p6
ba1
ba2
ba3
ba4
ba5
ba6
pa1
pa2
pa3
pa4
pa5
pa6
0.30
0.15
0.00
−0.15
−0.30

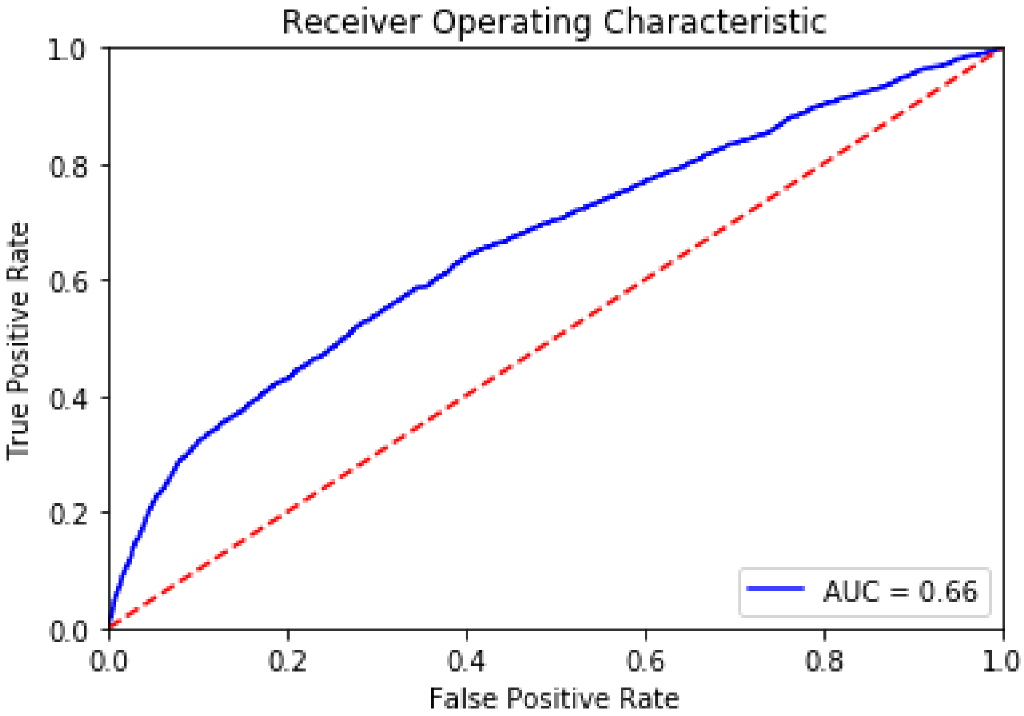
MODEL 1

x=df.drop("default",axis=1)

y=df["default"]

MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.778056	0.000000	0.000000	0.499929
LR-test	0.779750	0.000000	0.000000	0.500000
Decision Tree-train	1.000000	1.000000	1.000000	1.000000
Decision Tree-test	0.720250	0.378571	0.421112	0.612929
KNN (5)-train	0.813500	0.660768	0.327323	0.639709
KNN (5)-test	0.748917	0.359848	0.179720	0.544707
NB-train	0.365722	0.244106	0.886802	0.551990
NB-test	0.367750	0.243622	0.888763	0.554673
Random Forest train	0.8346	0.7646	0.3854	0.6740
Random Forest test	0.8192	0.6748	0.3454	0.6492

Since NB has the best Recall we plot ROC of NB model.



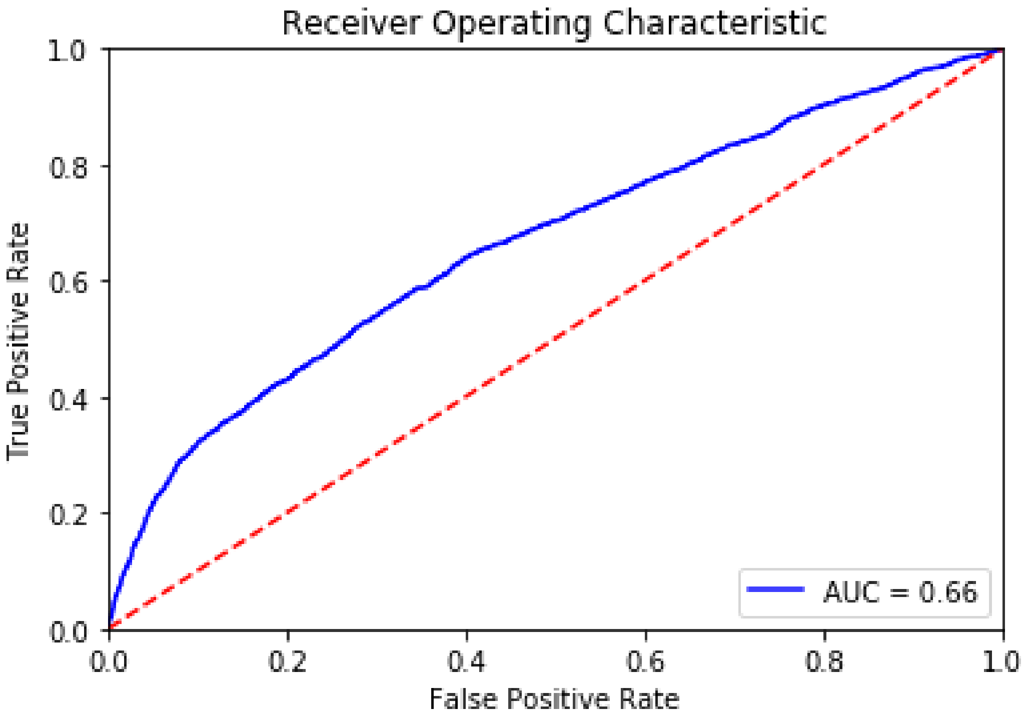
MODEL 2

We did some feature engineering and added some more columns.

```
df["bal"]=(df.ba2+df.ba3+df.ba4+df.ba5+df.ba6)-(df.pa1+df.pa2+df.pa3+df.pa4+df.pa5+df.pa6)
df["sump"]=df.p1+df.p2+df.p3+df.p4+df.p5+df.p6
df["lba1"]=df.lb-df.ba1
df["lba2"]=df.lb-df.ba2
df["lba3"]=df.lb-df.ba3
df["lba4"]=df.lb-df.ba4
df["lba5"]=df.lb-df.ba5
df["lba6"]=df.lb-df.ba6
x=df.drop("default",axis=1)
y=df["default"]
```

MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.778056	0.000000	0.000000	0.499929
LR-test	0.779750	0.000000	0.000000	0.500000
Decision Tree-train	1.000000	1.000000	1.000000	1.000000
Decision Tree-test	0.723750	0.384298	0.422247	0.615580
KNN (5)-train	0.813500	0.660768	0.327323	0.639709
KNN (5)-test	0.748917	0.359848	0.179720	0.544707
NB-train	0.365722	0.244106	0.886802	0.551990
NB-test	0.367750	0.243622	0.888763	0.554673
Random Forest train	0.8339	0.7448	0.3822	0.6724
Random Forest test	0.8207	0.6801	0.3507	0.6521

Since NB has the best Recall we plot ROC of NB model.



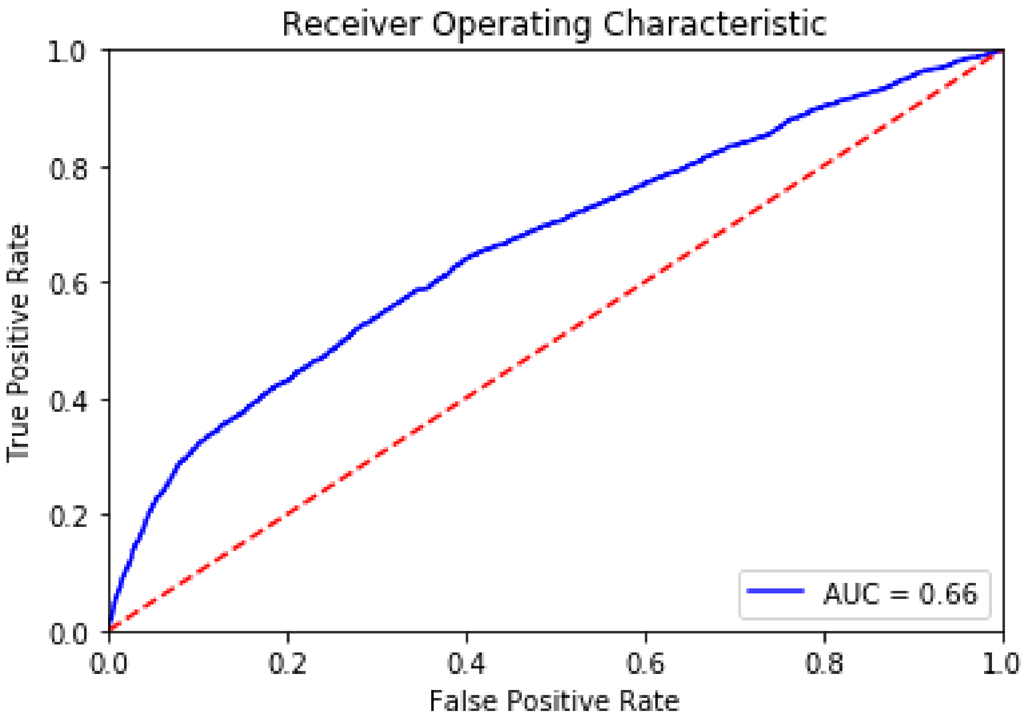
MODEL 3

x=df[["ba1","pa1","ba2","pa2","ba3","p3","lb","lba1","sump"]]

y=df["default"]

MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.778056	0.000000	0.000000	0.499929
LR-test	0.779750	0.000000	0.000000	0.500000
Decision Tree-train	1.000000	1.000000	1.000000	1.000000
Decision Tree-test	0.722417	0.380804	0.415815	0.612418
KNN (5)-train	0.813500	0.660768	0.327323	0.639709
KNN (5)-test	0.748917	0.359848	0.179720	0.544707
NB-train	0.365722	0.244106	0.886802	0.551990
NB-test	0.367750	0.243622	0.888763	0.554673
Random Forest train	0.8357	0.7527	0.3864	0.6751
Random Forest test	0.8187	0.6696	0.3488	0.6501

Since NB has the best Recall we plot ROC of NB model



MODEL 4

We did some feature engineering and added some more columns.

```
df["bal_sump"]=df.bal*df.sump
df["ba6_pa5"]=df.ba6-df.pa5
df["ba5_pa4"]=df.ba6-df.pa5
df["ba4_pa3"]=df.ba6-df.pa5
df["ba3_pa2"]=df.ba6-df.pa5
df["ba2_pa1"]=df.ba6-df.pa5
df["max_ba"]=df[["ba1","ba2","ba3","ba4","ba5","ba6"]].apply(np.max,axis=1)
df["max_pa"]=df[["pa1","pa2","pa3","pa4","pa5","pa6"]].apply(np.max,axis=1)
```

We modified the values of lba1, lba2, lba3, lba4, lba5, lba6.

```
df["lba1"][df["ba1"]==0]=0
df["lba2"][df["ba2"]==0]=0
df["lba3"][df["ba3"]==0]=0
df["lba4"][df["ba4"]==0]=0
```

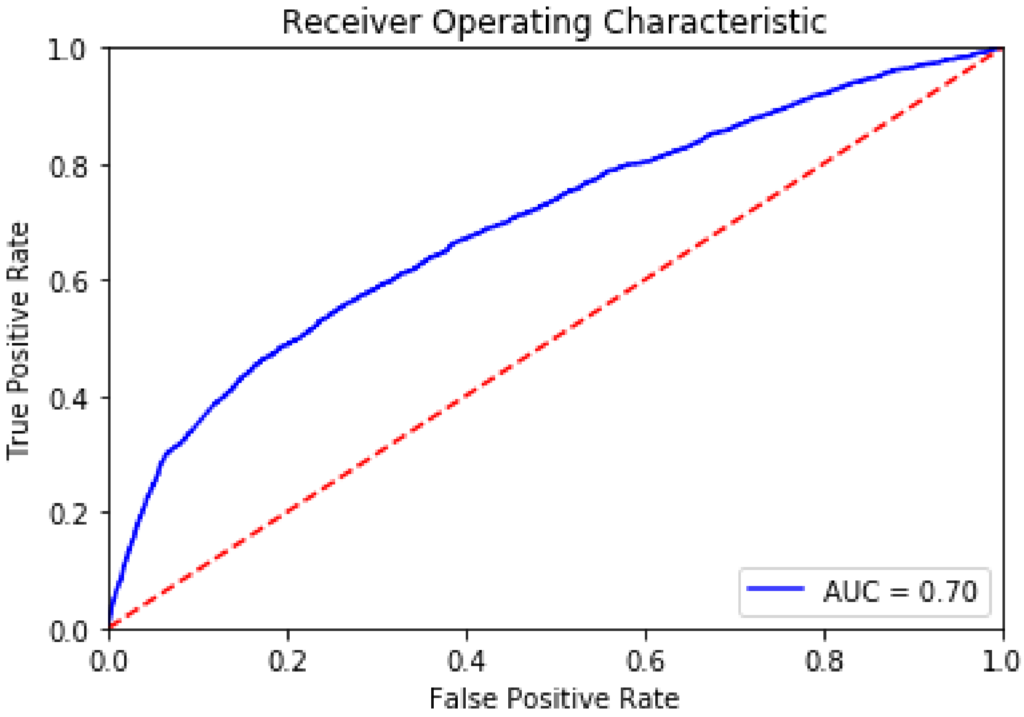
```
df["lba5"][df["ba5"]==0]=0
df["lba6"][df["ba6"]==0]=0
```

We Rescaled the continuous columns and again applied the above algorithms.

```
x=df[["pa1","ba2_pa1","ba1","ba3","lba1","lba2","lba3","lba4","lba5","lba6","bal_sump",”max_ba”]]
y=df["default"]
```

MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.795500	0.656627	0.163787	0.569685
LR-test	0.794667	0.638760	0.155883	0.565491
Decision Tree-train	0.988056	0.999471	0.946657	0.973257
Decision Tree-test	0.709833	0.347065	0.360197	0.584395
KNN (5)-train	0.818167	0.630624	0.435262	0.681292
KNN (5)-test	0.777417	0.492662	0.355656	0.626102
NB-train	0.554611	0.295819	0.730028	0.617316
NB-test	0.553167	0.294420	0.736663	0.618999
Random Forest train	0.8190	0.7330	0.2895	0.6297
Random Forest test	0.8002	0.6142	0.2493	0.6025

Since NB has the best Recall we plot ROC of NB model.

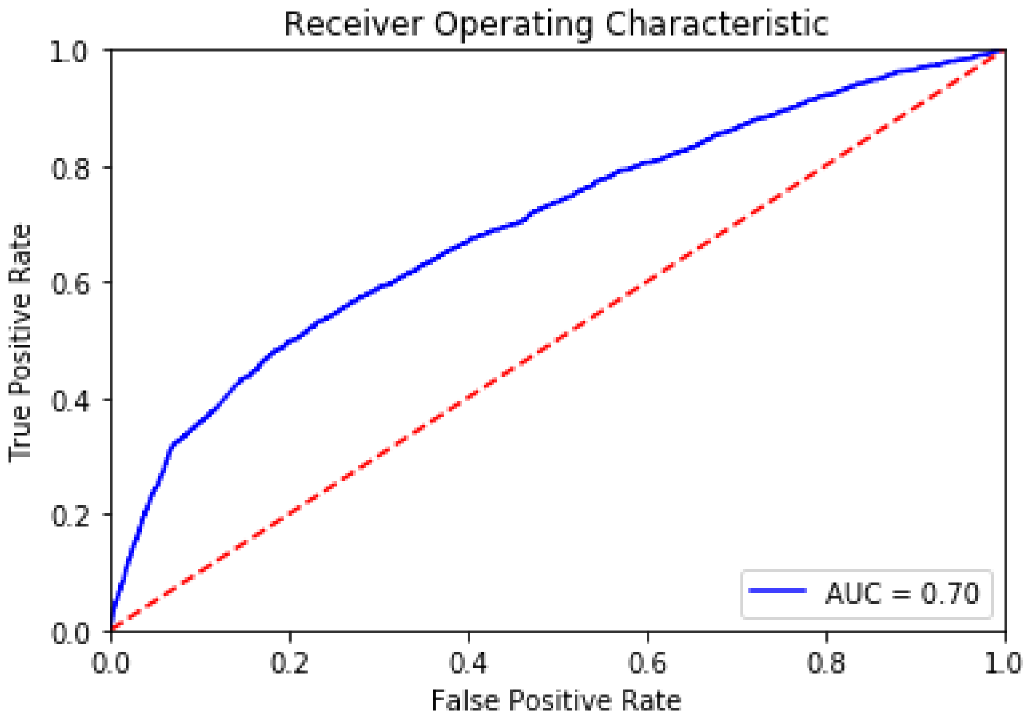


MODEL 5

```
x=df[["pa1","ba2_pa1","ba1","ba3","lba1","lba2","lba3","lba4","lba5","lba6","bal_sump"]]
y=df["default"]
```

MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.795056	0.648438	0.166291	0.570295
LR-test	0.794750	0.636778	0.158532	0.566495
Decision Tree-train	0.988056	0.999471	0.946657	0.973257
Decision Tree-test	0.712083	0.353111	0.369277	0.589095
KNN (5)-train	0.818944	0.631636	0.441022	0.683851
KNN (5)-test	0.777167	0.491777	0.350738	0.624177
NB-train	0.552167	0.294836	0.732031	0.61646
NB-test	0.550333	0.293225	0.738555	0.617861
Random Forest train	0.8177	0.7192	0.2925	0.6300
Random Forest test	0.7977	0.5961	0.2524	0.6020

Since NB has the best Recall we plot ROC of NB model.



FINALLY ACCEPTED MODEL

This model has the best recall, so this is finally accepted.

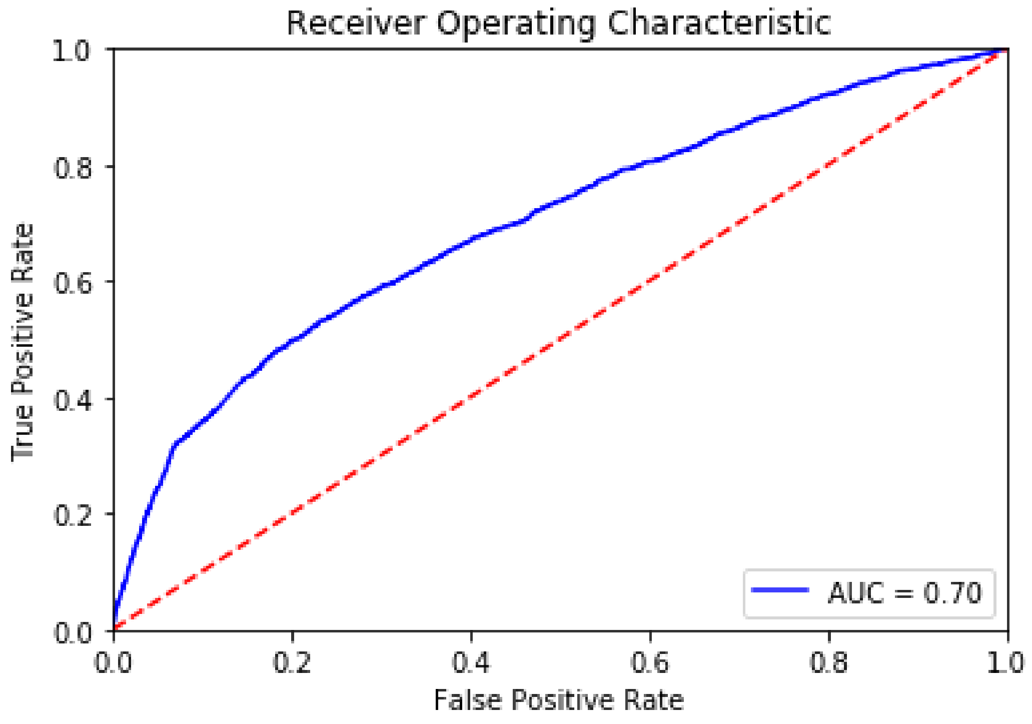
x=df[["pa1","ba2_pa1","ba1","ba3","lba1","lba2","lba3","lba4","lba5","lba6","bal_sump"]]

y=df["default"]

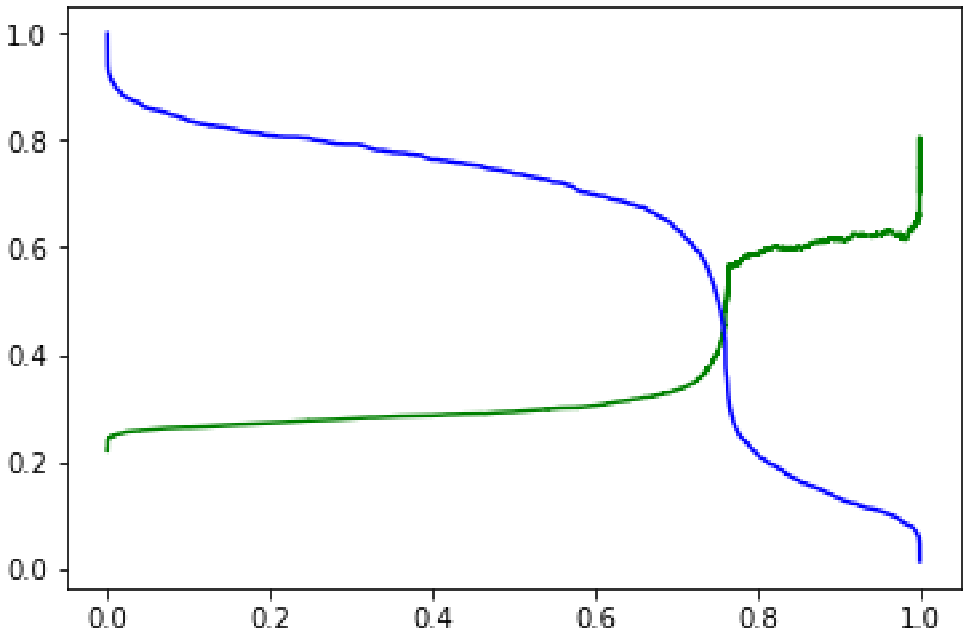
MODELNAME	ACCURACY	PRECISION	RECALL	AUC
LR-train	0.795056	0.648438	0.166291	0.570295
LR-test	0.794750	0.636778	0.158532	0.566495
Decision Tree-train	0.988056	0.999471	0.946657	0.973257
Decision Tree-test	0.712083	0.353111	0.369277	0.589095
KNN (5)-train	0.818944	0.631636	0.441022	0.683851
KNN (5)-test	0.777167	0.491777	0.350738	0.624177
NB-train	0.552167	0.294836	0.732031	0.61646
NB-test	0.550333	0.293225	0.738555	0.617861
Random Forest train	0.8177	0.7192	0.2925	0.6300
Random Forest test	0.7977	0.5961	0.2524	0.6020

Since NB has the best Recall we took the NB model.

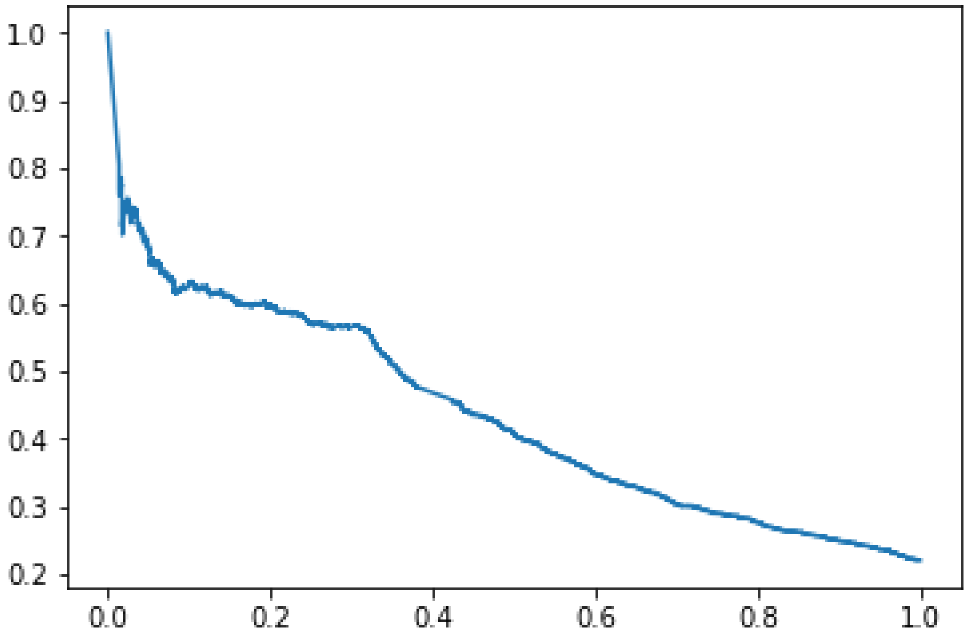
ROC



RECALL-PRECISION-THRESHOLD CURVE



RECALL-PRECISION CURVE



CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn import model_selection
from sklearn import metrics
from sklearn import preprocessing
from sklearn import feature_selection
from sklearn import utils
from sklearn import tree
from sklearn import ensemble
from sklearn import naive_bayes
from sklearn import svm
from sklearn import neighbors
def printresult(actual,predicted):
    confmatrix=metrics.confusion_matrix(actual,predicted)
    accscore=metrics.accuracy_score(actual,predicted)
    precscore=metrics.precision_score(actual,predicted)
    recscore=metrics.recall_score(actual,predicted)
    print(confmatrix)
    print("accuracy : {:.4f}".format(accscore))
    print("precision : {:.4f}".format(precscore))
    print("recall : {:.4f}".format(recscore))
    print("f1-score : {:.4f}".format(metrics.f1_score(actual,predicted)))
    print("AUC : {:.4f}".format(metrics.roc_auc_score(actual,predicted)))

def modelstats1(Xtrain,Xtest,ytrain,ytest):
    stats=[]
```

```
modelnames=["LR","DecisionTree","KNN","NB"]
models=list()
models.append(linear_model.LogisticRegression())
models.append(tree.DecisionTreeClassifier())
models.append(neighbors.KNeighborsClassifier())
models.append(naive_bayes.GaussianNB())
for name,model in zip(modelnames,models):
    if name=="KNN":
        k=[l for l in range(5,17,2)]
        grid={"n_neighbors":k}
        grid_obj = model_selection.GridSearchCV(estimator=model,param_grid=grid,scoring="f1")
        grid_fit =grid_obj.fit(Xtrain,ytrain)
        model = grid_fit.best_estimator_
        model.fit(Xtrain,ytrain)
        name=name+"("+str(grid_fit.best_params_["n_neighbors"])+")"
        print(grid_fit.best_params_)
    else:
        model.fit(Xtrain,ytrain)
    trainprediction=model.predict(Xtrain)
    testprediction=model.predict(Xtest)
    scores=list()
    scores.append(name+"-train")
    scores.append(metrics.accuracy_score(ytrain,trainprediction))
    scores.append(metrics.precision_score(ytrain,trainprediction))
    scores.append(metrics.recall_score(ytrain,trainprediction))
    scores.append(metrics.roc_auc_score(ytrain,trainprediction))
    stats.append(scores)
    scores=list()
    scores.append(name+"-test")
```

```
        scores.append(metrics.accuracy_score(ytest,testprediction))
        scores.append(metrics.precision_score(ytest,testprediction))
        scores.append(metrics.recall_score(ytest,testprediction))
        scores.append(metrics.roc_auc_score(ytest,testprediction))
        stats.append(scores)

    colnames=["MODELNAME","ACCURACY","PRECISION","RECALL","AUC"]
    return pd.DataFrame(stats,columns=colnames)

df=pd.read_csv("C:/Users/MY_PC/Desktop/DataSets/ credit_default.csv")
df.info()

coldict={"ID":"id","LIMIT_BAL":"lb","SEX":"sex","EDUCATION":"ed","MARRIAGE":"mr","AGE":"age","PAY_0":"p1","PAY_2":"p2","PAY_3":"p3","PAY_4":"p4","PAY_5":"p5","PAY_6":"p6","BILL_AMT1":"ba1","BILL_AMT2":"ba2","BILL_AMT3":"ba3","BILL_AMT4":"ba4","BILL_AMT5":"ba5","BILL_AMT6":"ba6","PAY_AMT1":"pa1","PAY_AMT2":"pa2","PAY_AMT3":"pa3","PAY_AMT4":"pa4","PAY_AMT5":"pa5","PAY_AMT6":"pa6","default.payment.next.month":"default"}
df.rename(columns=coldict,inplace=True)
df.mr.replace({0:3},inplace=True)
df.ed.replace({0:5,6:5},inplace=True)
df.p1.replace({-1:0,-2:0},inplace=True)
df.p2.replace({-1:0,-2:0},inplace=True)
df.p3.replace({-1:0,-2:0},inplace=True)
df.p4.replace({-1:0,-2:0},inplace=True)
df.p5.replace({-1:0,-2:0},inplace=True)
df.p6.replace({-1:0,-2:0},inplace=True)
for i in df.columns:
    print(i,df[i].unique().shape[0])
(df.corr()["default"]).sort_values(ascending=False)
df["bal"]=(df.ba2+df.ba3+df.ba4+df.ba5+df.ba6) - (df.pa1+df.pa2+df.pa3+df.pa4+df.pa5+df.pa6)
```

```
df["sump"]=df.p1+df.p2+df.p3+df.p4+df.p5+df.p6
df["lba1"]=df.lb-df.ba1
df["lba2"]=df.lb-df.ba2
df["lba3"]=df.lb-df.ba3
df["lba4"]=df.lb-df.ba4
df["lba5"]=df.lb-df.ba5
df["lba6"]=df.lb-df.ba6
df["bal_sump"]=df.bal*df.sump
df["ba6_pa5"]=df.ba6-df.pa5
df["ba5_pa4"]=df.ba6-df.pa5
df["ba4_pa3"]=df.ba6-df.pa5
df["ba3_pa2"]=df.ba6-df.pa5
df["ba2_pa1"]=df.ba6-df.pa5
df["max_ba"]=df[["ba1","ba2","ba3","ba4","ba5","ba6"]].apply(np.max,axis=1)
df["max_pa"]=df[["pa1","pa2","pa3","pa4","pa5","pa6"]].apply(np.max,axis=1)
df["lba1"][df["ba1"]==0]=0
df["lba2"][df["ba2"]==0]=0
df["lba3"][df["ba3"]==0]=0
df["lba4"][df["ba4"]==0]=0
df["lba5"][df["ba5"]==0]=0
df["lba6"][df["ba6"]==0]=0
df.info()
df.drop("id",axis=1,inplace=True)
df1=df[["ba1","ba2","ba3","ba4","ba6","pa1","pa2","pa3","pa4","pa5","pa6","bal","bal_sump","lba1","lba2","lba3","lba4","lba5","lba6","max_ba","max_pa","ba2_pa1","ba3_pa2","ba4_pa3","ba5_pa4","ba6_pa5"]]
colnames=df1.columns.values
rb_scaler=preprocessing.RobustScaler()
x_scaled=rb_scaler.fit_transform(df1)
x_sc=pd.DataFrame(x_scaled,columns=colnames)
df2=df1[["pa1","ba2_pa1","ba1","ba3","lba1","lba2","lba3","lba4","lba5","lba6","bal_sump"]]
```

```
xtrain,xtest,ytrain,ytest=model_selection.train_test_split(df7,y,test_size=0.4,random_state=42)
modelstats1(xtrain,xtest,ytrain,ytest)
rf=ensemble.RandomForestClassifier(n_estimators=100,criterion="entropy",max_depth=7)
rf.fit(xtrain,ytrain)
prediction1=rf.predict(xtrain)
printresult(ytrain,prediction1)
prediction2=rf.predict(xtest)
printresult(ytest,prediction2)

distances=model1.predict_proba(xtest)[:,1]
precision,recall,thresh=metrics.precision_recall_curve(ytest,distances)
plt.plot(thresh,precision[:-1],color="g")
plt.plot(thresh,recall[:-1],color="b")
plt.plot(recall,precision)

probs=model1.predict_proba(xtest)
preds=probs[:,1]
fpr,tpr,threshold=metrics.roc_curve(ytest,preds)
roc_auc=metrics.auc(fpr,tpr)
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

CONCLUSION AND FUTURE SCOPE OF IMPROVEMENTS

We have met our objectives in this problem, being able to apply various models, algorithms, and strategies to achieve relatively good predictions. We have obtained the final recall score of 0.738555.

We investigated the data, checking for data unbalancing, visualizing the features and understanding the relationship between different features.

We conclude that:

i) People who have already paid more.
ii) The people with less difference between total bill amount and total payment.
iii) People who have delayed the most.

All are basically the defaulters.

In terms of what we would have liked to do more of, a few things come to mind. These tasks can be addressed as a part of future scope of enhancements.

- Tune the parameters for Deep Learning using Theano/Keras and compare the predictive accuracy and performance against Stacking/Voting models
- Explore the possibility of adding new polynomial and transformed features, and evaluate the predictive accuracy.

With much better algorithms and Deep Learning much better results can be obtained.

CERTIFICATE

This is to certify that Mr. SAQUIB NAZEER, KALYANI GOVERNMENT ENGINEERING COLLEGE, Registration Number: 161020110035, has successfully completed a project on ‘Credit default prediction’ using “Machine learning using Python under the guidance of Mr. Titas Roy Chowdhury.

____

(Mr. Titas Roy Chowdhury)

Globsyn Finishing School

CERTIFICATE

This is to certify that Mr. AKASH SHARMA, KALYANI GOVERNMENT ENGINEERING COLLEGE, Registration Number: 161020110052, has successfully completed a project on ‘Credit default prediction’ using “Machine learning using Python under the guidance of Mr. Titas Roy Chowdhury.

____

(Mr. Titas Roy Chowdhury)

Globsyn Finishing School

CERTIFICATE

This is to certify that Ms. PUSPITA UTHYASANI, KALYANI GOVERNMENT ENGINEERING COLLEGE, Registration Number: 161020110035, has successfully completed a project on ‘Credit default prediction’ using “Machine learning using Python under the guidance of Mr. Titas Roy Chowdhury.

____

(Mr. Titas Roy Chowdhury)

Globsyn Finishing School

CERTIFICATE

This is to certify that Mr. ANIKET MONDAL, KALYANI GOVERNMENT ENGINEERING COLLEGE, Registration Number: 161020110054, has successfully completed a project on ‘Credit default prediction’ using “Machine learning using Python under the guidance of Mr. Titas Roy Chowdhury.

____

(Mr. Titas Roy Chowdhury)

Globsyn Finishing School