

Credit Reliability Model

Data Mining Report



Submitted by Group 04-Section 002 :

Jyoti Shivhare, Mohsin Asif, Tausif Parkar, Rakitha Bansayake, William Willits

Contents

| | |
|------------------------------|----|
| Introduction..... | 3 |
| System Design | 4 |
| Business Understanding | 5 |
| Algorithms | 7 |
| Final Recommendation | 15 |
| Challenges | 15 |
| Next Steps..... | 16 |
| Conclusion..... | 16 |
| Appendix A | 18 |

Introduction

A large E-Commerce client is preparing to roll out a new consumer credit card, much like is common with competitors (Macy's, Amazon, Kroger, etc.). Besides encouraging customer loyalty, offering a consumer credit card also assists a company in offering promotions, collecting customer data, and making money from credit. As part of the preparation process, Group 4 Consultants has been tasked with developing predictive models to help this client evaluate credit card applications and quantify the risk associated with granting lines of credit to specific people. Determining risk is very important, because it will allow the company to determine whether or not to approve new lines of credit, and also the size of the credit limits. To this end, one of the highest priority models needs to accurately predict the likelihood that a credit applicant will make their next payment. This report will detail the work of Group 4 Consulting by introducing the sample data set, describing the modelling process, and recommending the best solutions.

System Design

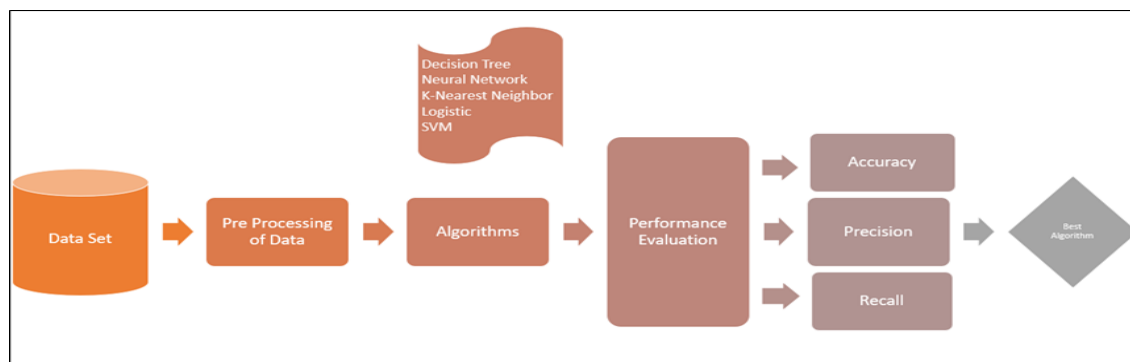


Figure 1: CRISP-DM Methodology

We have followed CRISP-DM Methodology in our Data mining process

Business Understanding

This initial phase focused on understanding the project objectives and requirements from a business perspective. Credit default has been a major challenge for our client as it endangers their system and pose chances of losses that might be hard to revert. Credit card defaulters are on the rise too. Our client wants to predict if a particular customer will default in his next month's payment. Based on the prediction our client will make necessary measures to tackle default customer as Credit risk will consumes capital and also eat into net profit margin (via loan loss provision expense)In order to create this predictive model, Group 4 Consulting used a data set that contained the credit card history and demographic data of 30,000 consumers in Taiwan (which is one of this E- Commerce company's primary markets). The set is called *Default of Credit Card Clients* and was found on the UCI Machine Learning Repository. This data initially contained 24 attributes, including demographic data (Gender, Education, Marital Status, Age) and credit history data (Credit Limit, Payment History, Bill Amount, Payment Amount). The fields of Payment History, Bill Amount, and Payment Amount all contained data for the past six months. For the purposes of this study, Bill Amount and Payment Amount were deemed redundant (much of the relevant information in these fields can be inferred by the Payment History field), so these fields were omitted. Below is a description of each field:

| Variable Name | Type | Description |
|-------------------|-------------|---|
| Limit Balance | Continuous | |
| Sex | Flag | Male Female |
| Education | Nominal | High School, University, Graduate School |
| Marriage | Categorical | Married, Single |
| Age | Nominal | |
| Pay (6 months) | Categorical | Revolving Credit Paid Duly Credit not used X Month Delay |
| Default | Categorical | Yes No |

Figure 2 Data Description

Limit Balance can be understood as the total credit limit available to the applicant. Sex (Male/Female), Education (High School, College, Graduate School), Marriage (Single, Married), and Age are fairly self-explanatory. The Pay variable breaks down into the previous six months, and describes the payment history of the applicant. 'Revolving Credit' takes place when the applicant completely pays off their credit card bill. This applicant carries the least risk, but will not generate interest. 'Paid Duly' means that the applicant has been dutifully paying their credit card bills, but is not totally paying the credit line off. This can be a lucrative type of applicant from the perspective of the creditor, because they will accrue significant interest, but also pay their bills on time. 'Credit not used' simply happens when an applicant has not used their credit card in a given month and is not carrying a balance. Finally, 'X Month Delay' describes the riskiest type of applicant, and the variable X is the number of consecutive months that the applicant has missed their payments. The 'Default' field is the target variable that Group 4 Consulting is trying to develop predictive models for and is a Boolean value of whether or not the person paid the next bill.

Algorithms

Decision Tree

One of the most common, successful and transparent ways to do the required binary classification to “Yes” and “No” is via a Decision Tree. A tree structure is established with known facts and classifications in order to generalize relevant judgment rules. The decision trees used in this paper are CART, CHAID and C5.0 which are explained below.

CART (classification and regression tree) is a binary decision-tree technique. The Gini index aims to separate the largest category (measured by the number of observations) from others in the node. CHAID (chi-square automatic interaction detector) is a branch of the decision tree algorithm. The CHAID algorithm mainly relies on chi-square tests in the process of constructing decision trees, and the optimal splitting branch is identified by repeating the process of combinations and divisions. C5.0 was developed by Quinlan as an improvement of ID3. The ID3 methodology refers to information gain as the criteria of constructing decision trees, and this typically results in over-learning due to an excessively large number of input variables. C5.0 uses the gains ratio to replace the previous criteria. However, the fundamental concept remains the same.

Table 1: Evaluation Summary

| Measures | CHAID | CART | C5 |
|-------------|--------|--------|--------|
| Accuracy | 79.24% | 81.34% | 81.66% |
| Sensitivity | 35.77% | 36.88% | 32.30% |
| AUC | 0.73 | 0.69 | 0.64 |
| Specificity | 92.02% | 94.41% | 96.17% |

Artificial Neural Networks

Artificial neural networks can be a good solution for solving binary questions with very large data sets. In this case, our team is working with a data set of approximately 30,000 entries, and our

question is binary (will the applicant pay their next credit card bill, or not?). So, we attempt to model the problem with the following ANN methods:

- Multi-layer perceptron
- Radial basis function

Table 2: Summary of Results

| Model | Overall Accuracy | | | Based on Validation Data | |
|----------------------------|------------------|---------|------------|--------------------------|-------------|
| | Training | Testing | Validation | Sensitivity | Specificity |
| Multi-Layer Perceptron MLP | 81.69% | 82.33% | 81.10% | 66.15% | 94.78% |
| Radial Basis Function RBF | 80.03% | 80.66% | 79.12% | 63.54% | 92.07% |

In the chart above, the MLP produced stronger results than RBF in most categories. It produced a slightly higher accuracy in each partition, and a significantly higher sensitivity and specificity. Since the training and validation data accuracy percentages are close in both models, overfitting was not an issue here. Neither of the methods had an issue with perpetual training either, and both models finished running in under 1 minute. Regardless of this, MLP is recommended as the reference model for Artificial Neural Network.

Predictor Importance

In the MLP, the strongest predictor of the target variable was the field PAY_REVOLVING_CREDIT, which represents the most recent month's payment history. The second strongest predictor was LIMIT_BAL, which is the credit limit of the applicant. Both fields totally make sense from a logical and contextual perspective as well, so they probably have value in a business context. After that, the predictor importance plateaus amongst several determinants, so any other inferences wouldn't be very valuable:

ANN Recommendation

This model seemed to excel at predicting certain outcomes and fail at others. The confusion matrix below will help to elaborate:

Table 3: ANN Classification Table

| Observed | Predicted | |
|----------|-----------|-------|
| | No | Yes |
| No | 95.7% | 4.3% |
| Yes | 67.3% | 32.7% |

This matrix is taken from all aggregated partitions of the MLP model. As can be seen, the model is extremely successful at accurately predicting 'No', which means that the applicant will not pay the next month's credit card bill. The model will almost never predict 'Yes' when the result is 'No' (hence the very high specificity). However, the model fails miserably at predicting 'Yes' outcomes. Approximately 2/3 of the time, the model will be wrong when the outcome is 'Yes'. From a contextual standpoint, this means that the model will be very selective in predicting who will pay their next bill. If the business wants to be conservative and only grant credit to applicants who will almost surely make their payments, the MLP ANN could be a good model to use to help make decisions. But, if the company wants to liberally grant credit to more applicants, even if they might not pay the bill, this would be a poor model choice.

Support Vector Machine:

Since we want to predict whether a potential customer will default on their next payment or not, it is a binary classification problem. Therefore, SVM can be a great algorithm to solve this problem. We will run following kernels for SVM algorithm:

- 1) Radial Basis Function (RBF)
- 2) Polynomial
- 3) Sigmoid
- 4) Linear**

Table 4: Summary of Results:

| Model | Overall Accuracy | | | Based on Validation Data | |
|----------------------------|------------------|---------|------------|--------------------------|-------------|
| | Training | Testing | Validation | Sensitivity | Specificity |
| RBF (c=1, RBF Gamma =0.25) | 84.65% | 82.55% | 81.32% | 34.52% | 95.09% |
| RBF (c=10, RBF Gamma =0.5) | 92.54% | 79.16% | 77.47% | 37.10% | 89.35% |
| RBF (c=3, RBF Gamma =0.3) | 90.76% | 80.58% | 78.93% | 38.95% | 90.70% |
| RBF (c=2, RBF Gamma =0.4) | 91.01% | 80.61% | 79.22% | 38.88% | 91.09% |
| Polynomial | - | - | - | - | - |
| Linear | 81.79% | 82.42% | 81.61% | 32.15% | 96.15% |
| Sigmoid | 77.80% | 77.78% | 77.27% | - | - |
| Sigmoid | 77.80% | 77.78% | 77.27% | - | - |

RBF:

RBF function has various settings with which we will play to tune the model

Settings:

Since we have 12 variables in our dataset, the recommended value for RBF gamma is 3/12 to 6/12 which is from 0.25 to 0.5. Another parameter is C also called as regularization parameter; the recommended value is between 1 and 10. We will start with the minimum value of C as higher values can lead to overfitting problem.

RBF (First Run)

We ran RBF model with minimum settings i.e. c=1, RBF gamma=0.25. After running we got the results that are shown in summary results. We can see that the overfitting is rather less as the difference between training and validation results is minimal. The specificity (detection of default) is outstanding. On the other hand, sensitivity, the detection of customers who will not default on the next payment is rather poor. Which warrants for tuning the model

RBF (Second Run)

This time we ran the model with maximum setting i.e. c=10 and RBF gamma=0.5. These settings give the worst result. Overfitting is very high as can be seen in the summary results. Moreover, specificity drops down to 89 from 95. May be we need to have midrange settings.

RBF (Third run)

In this iteration, we have $c=3$ and RBF gamma is 0.30. After running these settings, the results are not much different.

RBF (Fourth run)

We further tuned the model with $c=2$ and RBF gamma to 0.4. Performance improved only a tiny bit but the overfitting is still very high. It seems that RBF gamma value higher than 0.25 drastically reduces prediction power of our model.

Dropping variables:

We tried to drop some variables such as age to see how it impacts our model. The results showed that it did not make any significant difference. Therefore, due to limitations of time and scope we will not tune the model any further.

In conclusion, RBF with $C=1$ and RBF gamma =0.25 gives us the best result.

Polynomial and Sigmoid:

For our dataset, polynomial and sigmoid functions did not perform at all. The polynomial function ran for more than 12 hours and still did not produce any results so we concluded that it is not relevant for our dataset and the question we are trying to answer.

Similarly, sigmoid function did not give complete confusion matrix when we ran it. It did not indicate any TP values which indicates that it is also not the best model for our dataset.

Linear Function:

Surprisingly, despite being the simplest kernel, linear kernel gives us the best result. The overfitting is minimal, and the specificity is higher than RBF.

SVM Conclusion:

We prefer linear kernel as it is simpler than RBF and provides superior results.

K-Nearest Neighbor

KNN has its own advantage when dealing with large dataset since KNN uses lazy learning which leads less computational power and time. However, finding the optimal k value take some effort.

In order to calculate the k value for “Default Credit Card Client” dataset, 10% of sample data was ran against KNN algorithm and let the model to choose k value between 3 to 12. Selecting sex, age, education and Marital status from the feature selection gave the best k which is 8. (See Appendix A Figure 1,2 & 3)

After finding the optimal k the model was ran against the actual data which was partitioned into training, validation and testing, generating overall accuracy of 82.78%, 82.05% and 83.10% respectively. (See Appendix A Figure 4)

Table 5: Classification Table

| Observed | Predicted | |
|-------------|-------------|---------|
| | Not Default | Default |
| Not Default | 4471 | 179 |
| Default | 937 | 388 |

Table 6: KNN Evaluation Summary

| Model | Accuracy | Precision | Recall | Specificity | F1 |
|-------|----------|-----------|--------|-------------|--------|
| KNN | 81.32% | 82.67% | 96.15% | 29.28% | 88.90% |

KNN recommendation

- high recall means that an algorithm returned most of the relevant results
- Using a test with **poor specificity** will result in a customer appearing to be a default customer, when he is, in fact, not default which may result losing some good customers

Logistic Regression:

Logistic regression is an efficient and powerful way to analyze the effect of a group of independent variables on a binary outcome by quantifying each independent variable unique contribution. Using components of linear regression reflected in the logit scale, logistic regression iteratively identifies the strongest linear combination of variables with the greatest probability of detecting the observed outcome.

Assumptions:

- We have one or more independent variables, which can be either continuous (i.e., an interval or ratio variable) or categorical (i.e., an ordinal or nominal variable)
- There needs to be a linear relationship between any continuous independent variables and the logit transformation of the dependent variable
- We should have independence of observations and the dependent variable should have mutually exclusive and exhaustive categories

One of the most common, successful and transparent ways to do the required binary classification to “good” and “bad” is via a logistic function. This is a function that takes as input the client characteristics and outputs the probability of default.

$$p = \frac{\exp(\beta_0 + \beta_1 \cdot x_1 + \dots + \beta_n \cdot x_n)}{1 + \exp(\beta_0 + \beta_1 \cdot x_1 + \dots + \beta_n \cdot x_n)}$$

where in the above

- p is the probability of default
- x_i is the explanatory factor i
- β_i is the regression coefficient of the explanatory factor i
- n is the number of explanatory variables

For each of the existing data points it is known whether the client has gone into default or not (i.e. $p=1$ or $p=0$). The aim in the here is to find the coefficients β_0, \dots, β_n such that the model's probability of default equals to the observed probability of default. Typically, this is done through

maximum likelihood. In reality, default probability will depend on the client characteristics in a more complicated way.

Model Performance:

As shown in fig 5, the evaluation metrics of all training, testing and validation datasets are provided. We are choosing validation dataset results which is tuned.

Accuracy is 82.66% with Area Under the curve is 0.714

Goodness of Fit:

- A statistically significant result (i.e., $p < .05$) indicates that the model does not fit the data well. You can see from the table above that the p -value is 0.259 (i.e., $p = 0.259$) therefore, not statistically significant. "Pearson", presents the Pearson chi-square statistic.

| Goodness-of-Fit | | | |
|-----------------|------------|------|-------|
| | Chi-Square | df | Sig. |
| Pearson | 1604.896 | 1569 | .259 |
| Deviance | 1336.045 | 1569 | 1.000 |

Pseudo R-Square

| | |
|---------------|------|
| Cox and Snell | .244 |
| Nagelkerke | .370 |
| McFadden | .260 |

- Large chi-square values indicate a poor fit for the model.
- Based on this measure, the model fits the data well.
- In multinomial logistic regression you can also consider measures that are similar to R^2 in ordinary least-squares linear regression, which is the proportion of variance that can be explained by the model.

| | |
|--------------------|--------|
| Accuracy | 82.66% |
| Sensitivity | 36.00% |
| Specificity | 95.32% |
| AUC | 0.714 |

We found out that the above-mentioned measurements from Logistic Regression. We also observed that the revolving credit had the most importance (fig: 6)

Final Recommendation

After comparing the performance of all the models, we had quite similar results for SVM, and decision tree. However, we prefer and recommend C5-decision tree algorithm over SVM. The simple reason for doing so is that we can dig deep into the decision trees and see if the rules generated by algorithm make real business sense or not. In contrast, SVM does not provide any visibility on how it reached to the solution therefore we cannot analyze if the output really makes business sense or not.

While analyzing C5, we found that it had great specificity i.e. detecting default on next payment with over 96.01% success rate based on testing data and 35.09% detection rate of customers who will not default on their next payment. The most influential variable in detecting default was revolving credit and this makes sense because paying minimum amount causes deep to increase gradually, making it likely for customer to default on the next payment. We can, with certain confidence, recommend our model since it will perform well in detecting default.

Challenges

The dataset we obtained was quite large and complex and required a lot of cleaning. We did not have enough time and scope to tune our models to perfection or to go back and re-prepare data for finer results. We, however, provided our best estimate for detecting customer default. We

also had huge problems with some of the kernels of SVM such as polynomial that ran for hours and did not produce any results and sigmoid that did not give complete results even though we tried dropping some variables.

Next Steps

To take this forward, we would like to spend more time on understanding the business and ask more relevant and specific business questions. In addition, we would like to spend more time on cleaning our data for better results. This will require considerable amount of time on tuning and going back to the data for adjustment. We would also like to come up with a schedule to re-train our model to keep it up to date with the latest data so that it can continue to detect fraud accurately.

Conclusion

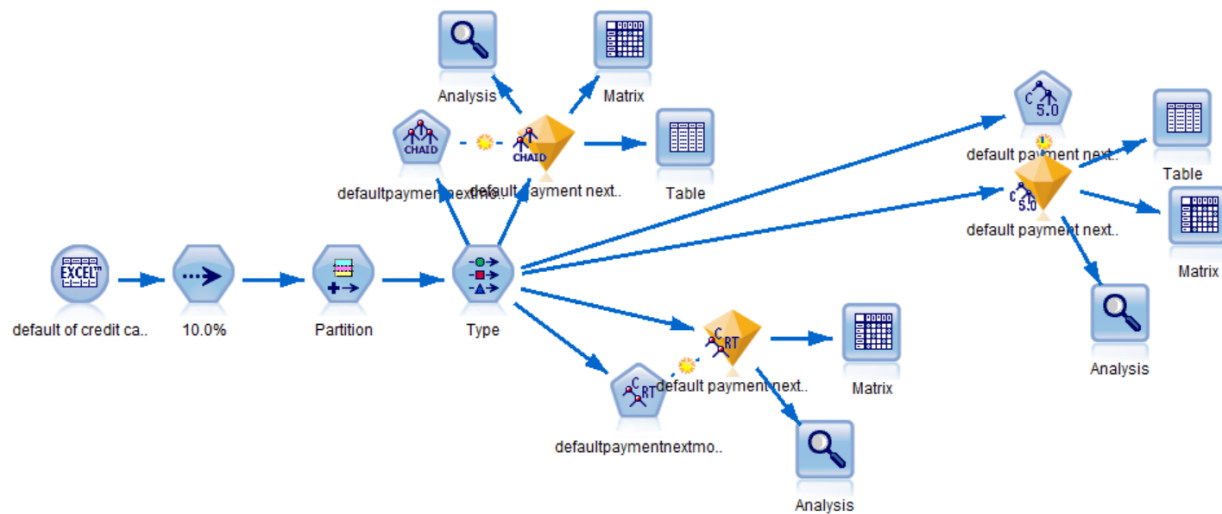
Our client, a large online retailer, wants to extend its line of credit to customers to increase its sales and convenience for customers. However, to mitigate the risk of default on payments, they want a reliable model to predict the potential customers who will default on their next payments. We, Group 4, accessed, cleaned and prepared the financial data for analysis. We analyzed 12 variables such as age, credit limit, payment history, marital status, etc and ran the data through five machine learning algorithms namely, decision trees, artificial neural networks, support

vector machine, k-nearest neighbors, and logistic regression to predict the default on next payment.

After careful analysis, we recommend C5 decision tree algorithm to detect potential default on next payment. C5 is better as it provides us visibility to the rules it generates, and we can analyze decision tree by digging deep into it if it makes any real-world sense. Moreover, C5's specificity and sensitivity were one of the highest among other ML algorithms we tested. According to our analysis, C5 can predict potentially defaulting customers with more than 96% success and the customers who will not default with the success of 35%. Since our model performs fairly well in detecting fraud, we can confidently recommend it. Finally, we will continue to work on perfecting our model and address the challenges and follow the next steps we have mentioned in our report.

Appendix A

Decision Tree- Tausif



C5:

Results for output field default payment next month

Individual Models

Comparing \$C-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 1,376 | 81.86% | 508 | 83.83% | 494 | 81.12% |
| Wrong | 305 | 18.14% | 98 | 16.17% | 115 | 18.88% |
| Total | 1,681 | | 606 | | 609 | |

Coincidence Matrix for \$C-default payment next month (rows show actuals)

| 'Partition' = 1_Training | No | Yes |
|--------------------------|-------|-----|
| No | 1,260 | 37 |
| Yes | 268 | 116 |

| 'Partition' = 2_Testing | No | Yes |
|-------------------------|-----|-----|
| No | 466 | 16 |
| Yes | 82 | 42 |

| 'Partition' = 3_Validation | No | Yes |
|----------------------------|-----|-----|
| No | 454 | 18 |
| Yes | 97 | 40 |

Performance Evaluation

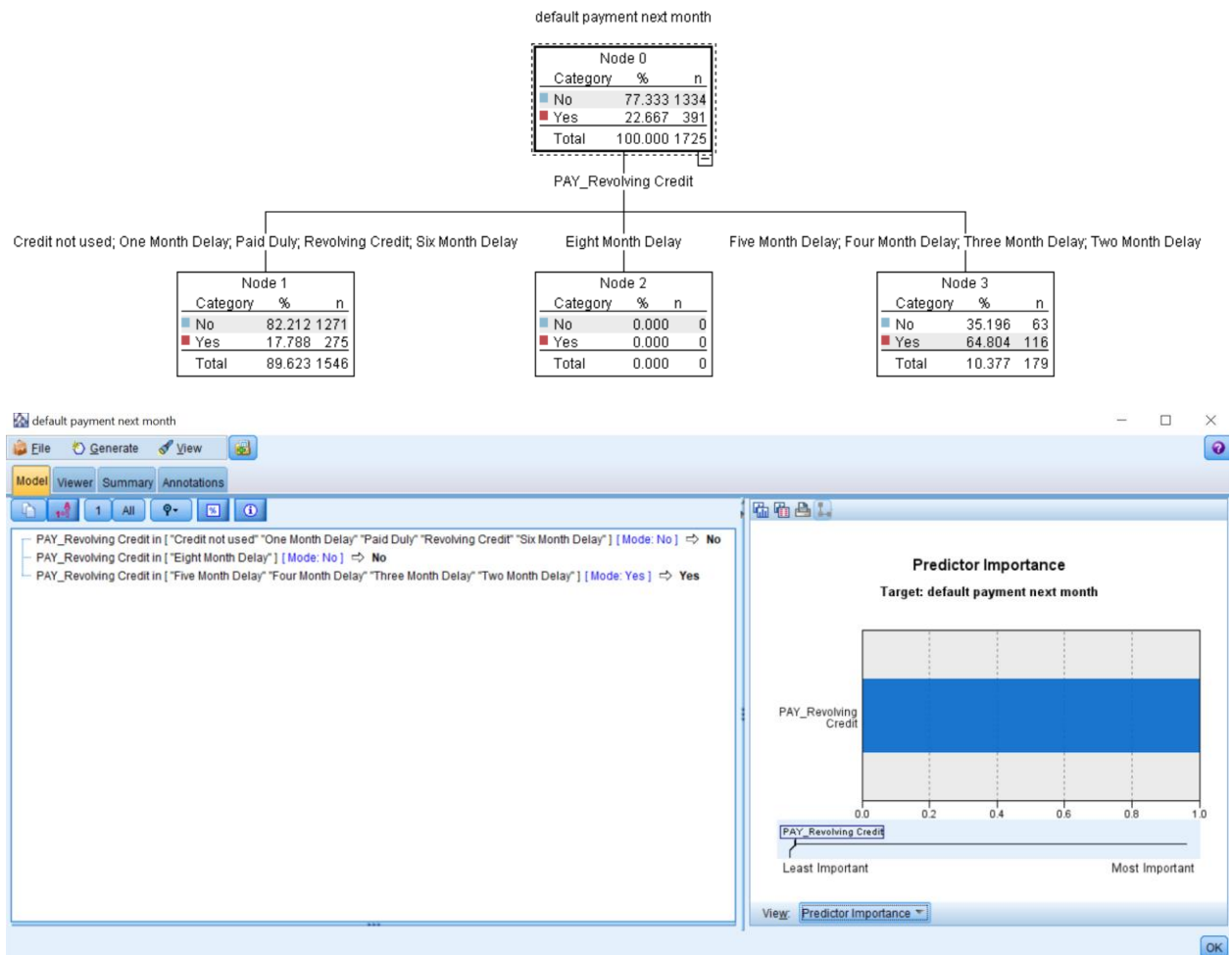
| 'Partition' = 1_Training | |
|--------------------------|-------|
| No | 0.066 |
| Yes | 1.2 |

| 'Partition' = 2_Testing | |
|-------------------------|-------|
| No | 0.067 |
| Yes | 1.264 |

| 'Partition' = 3_Validation | |
|----------------------------|-------|
| No | 0.061 |
| Yes | 1.12 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|--------------------------------|------------|-------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$C-default payment next month | 0.639 | 0.279 | 0.653 | 0.306 | 0.63 | 0.261 |



CART:

Results for output field default payment next month

Individual Models

Comparing SR-default payment next month with default payment next month

| 'Partition' | 1_Training | 2_Testing | 3_Validation |
|-------------|--------------|------------|--------------|
| Correct | 1,352 80.67% | 502 82.84% | 493 81.35% |
| Wrong | 324 19.33% | 104 17.16% | 113 18.65% |
| Total | 1,676 | 606 | 606 |

Coincidence Matrix for SR-default payment next month (rows show actuals)

'Partition' = 1_Training

| | No | Yes |
|-----|-------|-----|
| No | 1,204 | 82 |
| Yes | 242 | 148 |

'Partition' = 2_Testing

| | No | Yes |
|-----|-----|-----|
| No | 450 | 28 |
| Yes | 76 | 52 |

'Partition' = 3_Validation

| | No | Yes |
|-----|-----|-----|
| No | 447 | 24 |
| Yes | 89 | 46 |

Performance Evaluation

'Partition' = 1_Training

| | |
|-----|-------|
| No | 0.082 |
| Yes | 1.017 |

'Partition' = 2_Testing

| | |
|-----|-------|
| No | 0.081 |
| Yes | 1.124 |

'Partition' = 3_Validation

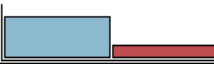
| | |
|-----|-------|
| No | 0.07 |
| Yes | 1.082 |

Evaluation Metrics

| 'Partition' | 1_Training | 2_Testing | 3_Validation |
|-------------------------------|------------|-------------|--------------|
| Model | AUC Gini | AUC Gini | AUC Gini |
| SR-default payment next month | 0.675 0.35 | 0.699 0.398 | 0.694 0.389 |

default payment next month

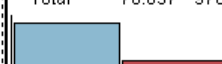
| Node 0 | | |
|----------|---------|------|
| Category | % | n |
| No | 77.393 | 962 |
| Yes | 22.607 | 281 |
| Total | 100.000 | 1243 |




PAY_Revolving Credit
Improvement=0.058

Credit not used; Paid Duly; Revolving Credit; Six Month Delay Four Month Delay; One Month Delay; Three Month Delay; Two Month Delay

| Node 1 | | |
|----------|--------|-----|
| Category | % | n |
| No | 86.392 | 838 |
| Yes | 13.608 | 132 |
| Total | 78.037 | 970 |



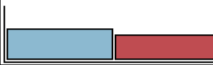
| Node 2 | | |
|----------|--------|-----|
| Category | % | n |
| No | 45.421 | 124 |
| Yes | 54.579 | 149 |
| Total | 21.963 | 273 |



PAY_Four Month Delay
Improvement=0.009

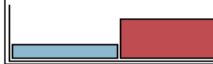
Credit not used; Paid Duly; Revolving Credit; Six Month Delay; Three Month Delay Seven Month Delay; Two Month Delay

| Node 5 | | |
|----------|--------|-----|
| Category | % | n |
| No | 55.618 | 99 |
| Yes | 44.382 | 79 |
| Total | 14.320 | 178 |



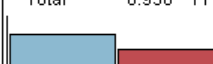
PAY_Revolving Credit
Improvement=0.004

| Node 6 | | |
|----------|--------|----|
| Category | % | n |
| No | 26.316 | 25 |
| Yes | 73.684 | 70 |
| Total | 7.643 | 95 |




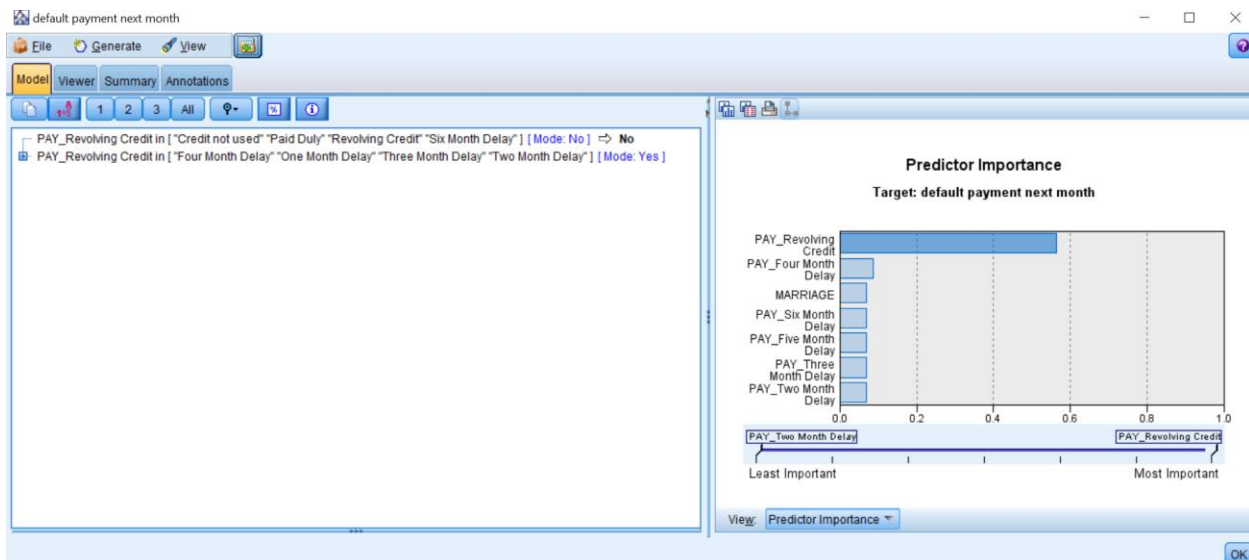
One Month Delay Three Month Delay; Two Month Delay

| Node 9 | | |
|----------|--------|-----|
| Category | % | n |
| No | 64.865 | 72 |
| Yes | 35.135 | 39 |
| Total | 8.930 | 111 |



| Node 10 | | |
|----------|--------|----|
| Category | % | n |
| No | 40.299 | 27 |
| Yes | 59.701 | 40 |
| Total | 5.390 | 67 |





CHAID:

Results for output field default payment next month

Individual Models

Comparing \$R-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 1,367 | 79.62% | 502 | 80.58% | 499 | 80.61% |
| Wrong | 350 | 20.38% | 121 | 19.42% | 120 | 19.39% |
| Total | 1,717 | | 623 | | 619 | |

Coincidence Matrix for \$R-default payment next month (rows show actuals)

| 'Partition' = 1_Training | | No | Yes |
|--------------------------|--|-------|-----|
| No | | 1,223 | 119 |
| Yes | | 231 | 144 |

| 'Partition' = 2_Testing | | No | Yes |
|-------------------------|--|-----|-----|
| No | | 438 | 36 |
| Yes | | 85 | 64 |

| 'Partition' = 3_Validation | | No | Yes |
|----------------------------|--|-----|-----|
| No | | 451 | 38 |
| Yes | | 82 | 48 |

Performance Evaluation

| 'Partition' = 1_Training | | |
|--------------------------|--|-------|
| No | | 0.073 |
| Yes | | 0.919 |

| 'Partition' = 2_Testing | | |
|-------------------------|--|-------|
| No | | 0.096 |
| Yes | | 0.984 |

| 'Partition' = 3_Validation | | |
|----------------------------|--|-------|
| No | | 0.069 |
| Yes | | 0.977 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|--------------------------------|------------|-------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$R-default payment next month | 0.726 | 0.452 | 0.794 | 0.588 | 0.705 | 0.411 |

Support Vector Machines

RBF 1

Analysis of [default payment next month]

File Edit

Analysis Annotations

Collapse All Expand All

Results for output field default payment next month

Individual Models

Comparing \$\$-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 14,949 | 84.65% | 4,944 | 82.55% | 4,841 | 81.32% |
| Wrong | 2,710 | 15.35% | 1,045 | 17.45% | 1,112 | 18.68% |
| Total | 17,659 | | 5,989 | | 5,953 | |

Coincidence Matrix for \$\$-default payment next month (rows show actuals)

'Partition' = 1_Training

| | No | Yes |
|-----|--------|-------|
| No | 13,256 | 482 |
| Yes | 2,228 | 1,693 |

'Partition' = 2_Testing

| | No | Yes |
|-----|-------|-----|
| No | 4,414 | 244 |
| Yes | 801 | 530 |

'Partition' = 3_Validation

| | No | Yes |
|-----|-------|-----|
| No | 4,374 | 226 |
| Yes | 886 | 467 |

Performance Evaluation

'Partition' = 1_Training

| | |
|-----|-------|
| No | 0.096 |
| Yes | 1.254 |

'Partition' = 2_Testing

| | |
|-----|-------|
| No | 0.085 |
| Yes | 1.125 |

'Partition' = 3_Validation

| | |
|-----|-------|
| No | 0.073 |
| Yes | 1.087 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---------------------------------|------------|------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$\$-default payment next month | 0.885 | 0.77 | 0.73 | 0.459 | 0.711 | 0.422 |

OK

RBF-2

Analysis

Analysis of default of credit card clients_cleaned.xls (Apr 17, 2018 11:44:26 PM)

Fields

Build Settings

- Use partitioned data: true
- Partition: Partition
- Calculate predictor importance: true
- Calculate raw propensity scores: false
- Calculate adjusted propensity scores: false
- Mode: Expert
- Append all probabilities (valid only for categorical targets): true
- Stopping criteria: 1.0E-3
- Kernel type: RBF
- Regularization parameter : 10
- Regression precision (epsilon): 0.1
- RBF gamma: 0.5
- Gamma: 1.0
- Bias: 0.0
- Degree: 3

Training Summary

Results for output field default payment next month

Individual Models

Comparing \$\$-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 16,342 | 92.54% | 4,741 | 79.16% | 4,612 | 77.47% |
| Wrong | 1,317 | 7.46% | 1,248 | 20.84% | 1,341 | 22.53% |
| Total | 17,659 | | 5,989 | | 5,953 | |

Coincidence Matrix for \$\$-default payment next month (rows show actuals)

| 'Partition' = 1_Training | | No | Yes |
|----------------------------|--|--------|-------|
| No | | 13,548 | 190 |
| Yes | | 1,127 | 2,794 |
| 'Partition' = 2_Testing | | No | Yes |
| No | | 4,170 | 488 |
| Yes | | 760 | 571 |
| 'Partition' = 3_Validation | | No | Yes |
| No | | 4,110 | 490 |
| Yes | | 851 | 502 |

Performance Evaluation

| 'Partition' = 1_Training | | |
|----------------------------|--|-------|
| No | | 0.171 |
| Yes | | 1.439 |
| 'Partition' = 2_Testing | | |
| No | | 0.084 |
| Yes | | 0.886 |
| 'Partition' = 3_Validation | | |
| No | | 0.07 |
| Yes | | 0.8 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---------------------------------|------------|-------|-----------|------|--------------|------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$\$-default payment next month | 0.924 | 0.847 | 0.71 | 0.42 | 0.695 | 0.39 |

RBF-3

Analysis

Analysis of default of credit card clients_cleaned.xls (Apr 18, 2018 12:08:17 AM)

Fields

Build Settings

- Use partitioned data: true
- Partition: Partition
- Calculate predictor importance: true
- Calculate raw propensity scores: false
- Calculate adjusted propensity scores: false
- Mode: Expert
- Append all probabilities (valid only for categorical targets): true
- Stopping criteria: 1.0E-3
- Kernel type: RBF
- Regularization parameter : 3
- Regression precision (epsilon): 0.1
- RBF gamma: 0.3
- Gamma: 1.0
- Bias: 0.0
- Degree: 3

Training Summary

Results for output field default payment next month

Individual Models

Comparing \$\$-default payment next month with default payment next month

| 'Partition' | 1_Training | 2_Testing | 3_Validation |
|-------------|---------------|--------------|--------------|
| Correct | 16,028 90.76% | 4,826 80.58% | 4,699 78.93% |
| Wrong | 1,631 9.24% | 1,163 19.42% | 1,254 21.07% |
| Total | 17,659 | 5,989 | 5,953 |

Coincidence Matrix for \$\$-default payment next month (rows show actuals)

'Partition' = 1_Training

| | No | Yes |
|-----|--------|-------|
| No | 13,444 | 294 |
| Yes | 1,337 | 2,584 |

'Partition' = 2_Testing

| | No | Yes |
|-----|-------|-----|
| No | 4,238 | 420 |
| Yes | 743 | 588 |

'Partition' = 3_Validation

| | No | Yes |
|-----|-------|-----|
| No | 4,172 | 428 |
| Yes | 826 | 527 |

Performance Evaluation

'Partition' = 1_Training

| No | 0.156 |
|-----|-------|
| Yes | 1.397 |

'Partition' = 2_Testing

| No | 0.09 |
|-----|-------|
| Yes | 0.965 |

'Partition' = 3_Validation

| No | 0.077 |
|-----|-------|
| Yes | 0.887 |

Evaluation Metrics

| 'Partition' | 1_Training | 2_Testing | 3_Validation | | | |
|---------------------------------|------------|-----------|--------------|------|-------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$\$-default payment next month | 0.909 | 0.818 | 0.72 | 0.44 | 0.702 | 0.404 |

RBF-4

- [-] Analysis
 - [-] Analysis of default of credit card clients_cleaned.xls (Apr 18, 2018 12:14:15 AM)
- [-] Fields
- [-] Build Settings
 - Use partitioned data: true
 - Partition: Partition
 - Calculate predictor importance: true
 - Calculate raw propensity scores: false
 - Calculate adjusted propensity scores: false
 - Mode: Expert
 - Append all probabilities (valid only for categorical targets): true
 - Stopping criteria: 1.0E-3
 - Kernel type: RBF
 - Regularization parameter : 2
 - Regression precision (epsilon): 0.1
 - RBF gamma: 0.4
 - Gamma: 1.0
 - Bias: 0.0
 - Degree: 3
- [-] Training Summary

[-] Results for output field default payment next month

[-] Individual Models

[-] Comparing \$\$-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 16,072 | 91.01% | 4,828 | 80.61% | 4,716 | 79.22% |
| Wrong | 1,587 | 8.99% | 1,161 | 19.39% | 1,237 | 20.78% |
| Total | 17,659 | | 5,989 | | 5,953 | |

[-] Coincidence Matrix for \$\$-default payment next month (rows show actuals)

| 'Partition' = 1_Training | | No | Yes |
|----------------------------|--|--------|-------|
| No | | 13,450 | 288 |
| Yes | | 1,299 | 2,622 |
| 'Partition' = 2_Testing | | No | Yes |
| No | | 4,240 | 418 |
| Yes | | 743 | 588 |
| 'Partition' = 3_Validation | | No | Yes |
| No | | 4,190 | 410 |
| Yes | | 827 | 526 |

[-] Performance Evaluation

| 'Partition' = 1_Training | |
|----------------------------|-------|
| No | 0.159 |
| Yes | 1.401 |
| 'Partition' = 2_Testing | |
| No | 0.09 |
| Yes | 0.967 |
| 'Partition' = 3_Validation | |
| No | 0.078 |
| Yes | 0.905 |

[-] Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---------------------------------|------------|-------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$\$-default payment next month | 0.909 | 0.817 | 0.727 | 0.454 | 0.707 | 0.415 |

- Analysis
 - Analysis of default of credit card clients_cleaned.xls (Apr 18, 2018 11:37:45 PM)
 - Analysis of default of credit card clients_cleaned.xls (Apr 18, 2018 11:39:16 PM)
- Fields
- Build Settings
 - Use partitioned data: true
 - Partition: Partition
 - Calculate predictor importance: false
 - Calculate raw propensity scores: false
 - Calculate adjusted propensity scores: false
 - Mode: Expert
 - Append all probabilities (valid only for categorical targets): false
 - Stopping criteria: 1.0E-3
 - Kernel type: Sigmoid
 - Regularization parameter : 10
 - Regression precision (epsilon): 0.1
 - RBF gamma: 0.1
 - Gamma: 2.0
 - Bias: 0.0
 - Degree: 3
- Training Summary

Results for output field default payment next month

Individual Models

Comparing \$\$-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|-------|-----------|--------|--------------|--------|
| Correct | 13,738 | 77.8% | 4,658 | 77.78% | 4,600 | 77.27% |
| Wrong | 3,921 | 22.2% | 1,331 | 22.22% | 1,353 | 22.73% |
| Total | 17,659 | | 5,989 | | 5,953 | |

Coincidence Matrix for \$\$-default payment next month (rows show actuals)

| | | |
|----------------------------|--|--------|
| 'Partition' = 1_Training | | No |
| No | | 13,738 |
| Yes | | 3,921 |
| 'Partition' = 2_Testing | | No |
| No | | 4,658 |
| Yes | | 1,331 |
| 'Partition' = 3_Validation | | No |
| No | | 4,600 |
| Yes | | 1,353 |

Performance Evaluation

| | | |
|----------------------------|--|-----|
| 'Partition' = 1_Training | | |
| No | | 0.0 |
| 'Partition' = 2_Testing | | |
| No | | 0.0 |
| 'Partition' = 3_Validation | | |
| No | | 0.0 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|---------------------------------|------------|-------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$\$-default payment next month | 0.52 | 0.041 | 0.517 | 0.034 | 0.52 | 0.041 |

Sigmoid-2

Analysis

Analysis of default of credit card clients_cleaned.xls (Apr 19, 2018 12:54:13 AM)

Fields

Build Settings

- Use partitioned data: true
- Partition: Partition
- Calculate predictor importance: false
- Calculate raw propensity scores: false
- Calculate adjusted propensity scores: false
- Mode: Expert
- Append all probabilities (valid only for categorical targets): false
- Stopping criteria: 1.0E-3
- Kernel type: Sigmoid
- Regularization parameter : 10
- Regression precision (epsilon): 0.1
- RBF gamma: 0.1
- Gamma: 3.0
- Bias: 0.0
- Degree: 3

Training Summary

Results for output field default payment next month

Individual Models

Comparing \$S-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|-------|-----------|--------|--------------|--------|
| Correct | 13,738 | 77.8% | 4,658 | 77.78% | 4,600 | 77.27% |
| Wrong | 3,921 | 22.2% | 1,331 | 22.22% | 1,353 | 22.73% |
| Total | 17,659 | | 5,989 | | 5,953 | |

Coincidence Matrix for \$S-default payment next month (rows show actuals)

| | | |
|----------------------------|--|--------|
| 'Partition' = 1_Training | | No |
| No | | 13,738 |
| Yes | | 3,921 |
| 'Partition' = 2_Testing | | No |
| No | | 4,658 |
| Yes | | 1,331 |
| 'Partition' = 3_Validation | | No |
| No | | 4,600 |
| Yes | | 1,353 |

Performance Evaluation

| | | |
|----------------------------|--|-----|
| 'Partition' = 1_Training | | |
| No | | 0.0 |
| 'Partition' = 2_Testing | | |
| No | | 0.0 |
| 'Partition' = 3_Validation | | |
| No | | 0.0 |

Evaluation Metrics

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|--------------------------------|------------|-------|-----------|-------|--------------|-------|
| Model | AUC | Gini | AUC | Gini | AUC | Gini |
| \$S-default payment next month | 0.528 | 0.057 | 0.535 | 0.071 | 0.531 | 0.061 |

Linear

default payment next month

File Generate View Preview

Model Settings **Summary** Annotations

Collapse All Expand All

Analysis

- Analysis of default of credit card clients_cleaned.xls (Apr 16, 2018 12:51:45 PM)
- Fields
- Build Settings
 - Use partitioned data: true
 - Partition: Partition
 - Calculate predictor importance: false
 - Calculate raw propensity scores: false
 - Calculate adjusted propensity scores: false
 - Mode: Expert
 - Append all probabilities (valid only for categorical targets): false
 - Stopping criteria: 1.0E-3
 - Kernel type: Linear
 - Regularization parameter : 5
 - Regression precision (epsilon): 0.1
 - RBF gamma: 0.1
 - Gamma: 1.0
 - Bias: 0.0
 - Degree: 3
- Training Summary

Analysis of [default payment next month]

File Edit

Analysis Annotations

Collapse All Expand All

Results for output field default payment next month

Comparing \$S-default payment next month with default payment next month

| 'Partition' | 1_Training | | 2_Testing | | 3_Validation | |
|-------------|------------|--------|-----------|--------|--------------|--------|
| Correct | 14,443 | 81.79% | 4,936 | 82.42% | 4,858 | 81.61% |
| Wrong | 3,216 | 18.21% | 1,053 | 17.58% | 1,095 | 18.39% |
| Total | 17,659 | | 5,989 | | 5,953 | |

Coincidence Matrix for \$S-default payment next month (rows show actuals)

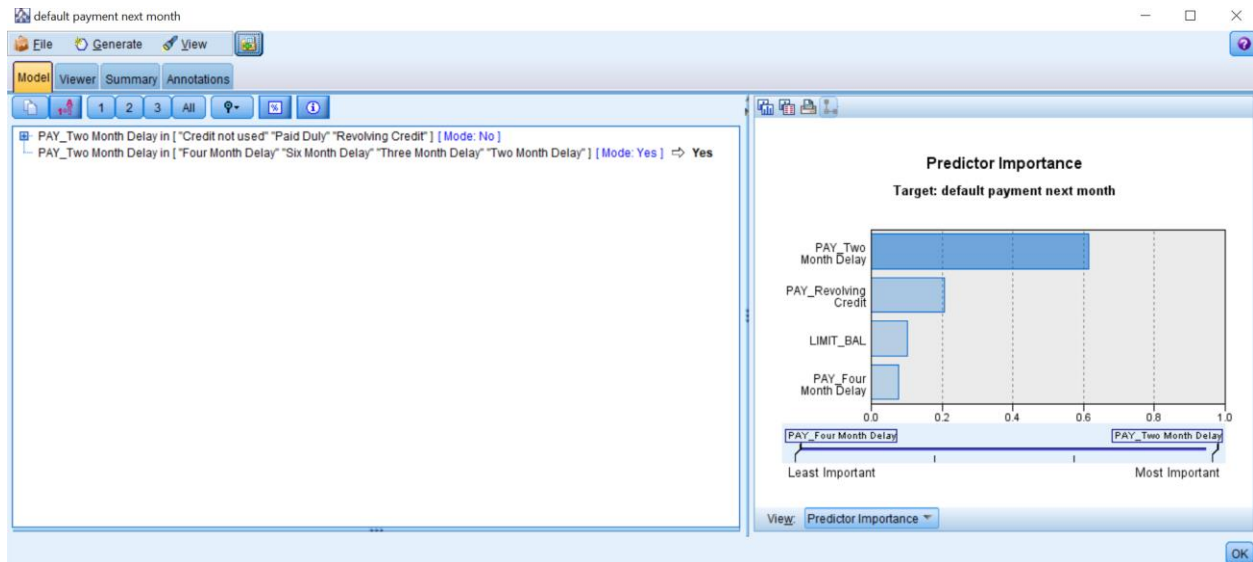
| 'Partition' = 1_Training | | No | Yes |
|--------------------------|--|--------|-------|
| No | | 13,194 | 544 |
| Yes | | 2,672 | 1,249 |

| 'Partition' = 2_Testing | | No | Yes |
|-------------------------|--|-------|-----|
| No | | 4,468 | 190 |
| Yes | | 863 | 468 |

| 'Partition' = 3_Validation | | No | Yes |
|----------------------------|--|-------|-----|
| No | | 4,423 | 177 |
| Yes | | 918 | 435 |

OK

K-NN



The screenshot shows the 'Number of Nearest Neighbors (k)' settings dialog. It has two radio buttons: 'Specify fixed K' and 'Automatically select k'. The 'Automatically select k' option is selected. Below it, there are two spin boxes: 'Minimum' set to 3 and 'Maximum' set to 12.

Figure 1: K Selection

The screenshot shows the 'Select Fields' dialog box. It has a 'Sort by' dropdown set to 'Natural'. The list of fields includes ID, LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, and PAY_Revolving Credit. The 'All' button is highlighted. At the bottom, there are 'OK', 'Apply', 'Cancel', and 'Help' buttons.

Figure 2: Perform feature selection

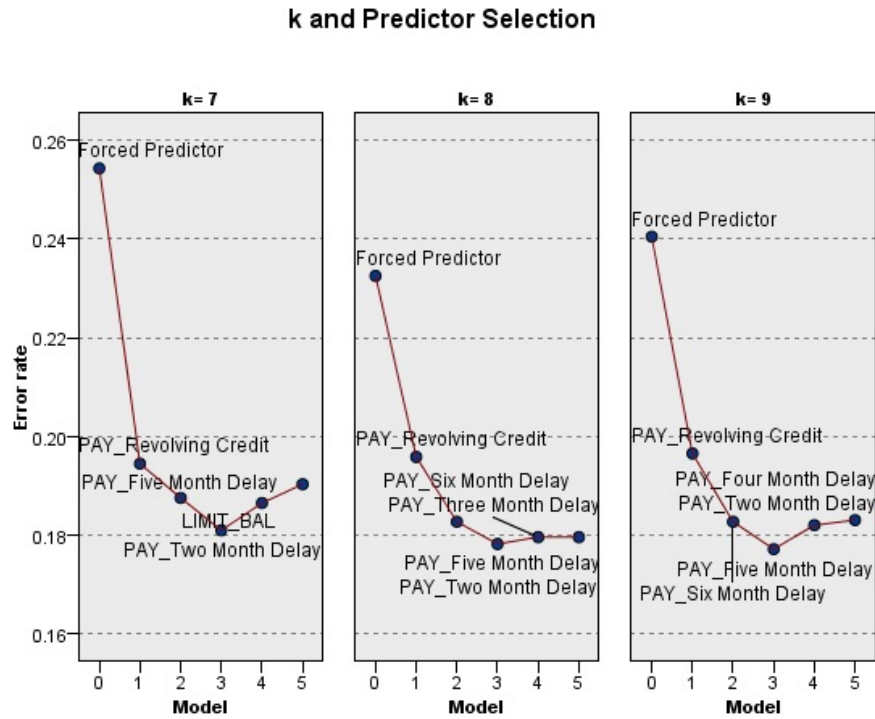


Figure 4: Finding optimal k value by lowering error rate

| Graph | Model | Partition | No. Records in Split | No. Fields Used | Overall Accuracy (%) |
|-------|-------|--------------|----------------------|-----------------|----------------------|
| | | 2_Testing | 5989 | 9 | 83.102 |
| | | 1_Training | 17659 | 9 | 82.779 |
| | | 3_Validation | 5953 | 9 | 82.059 |

Figure 5: Model performance for each partition

Logistic Regression

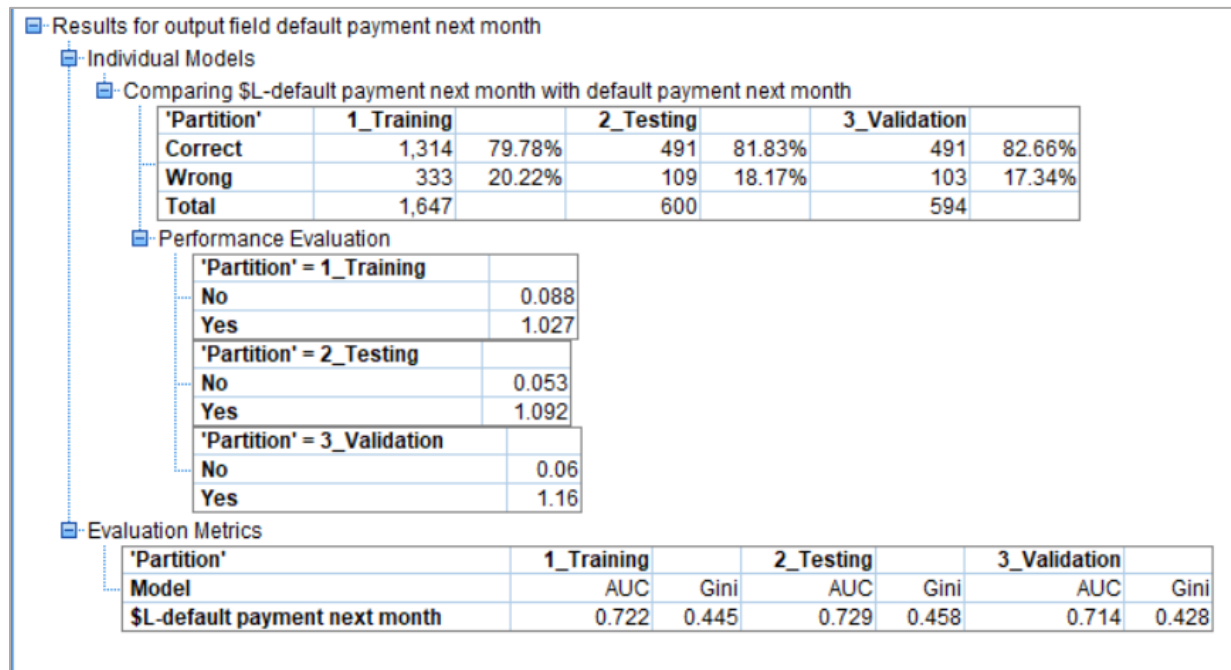


Figure 6: Logistic Regression Model Performance

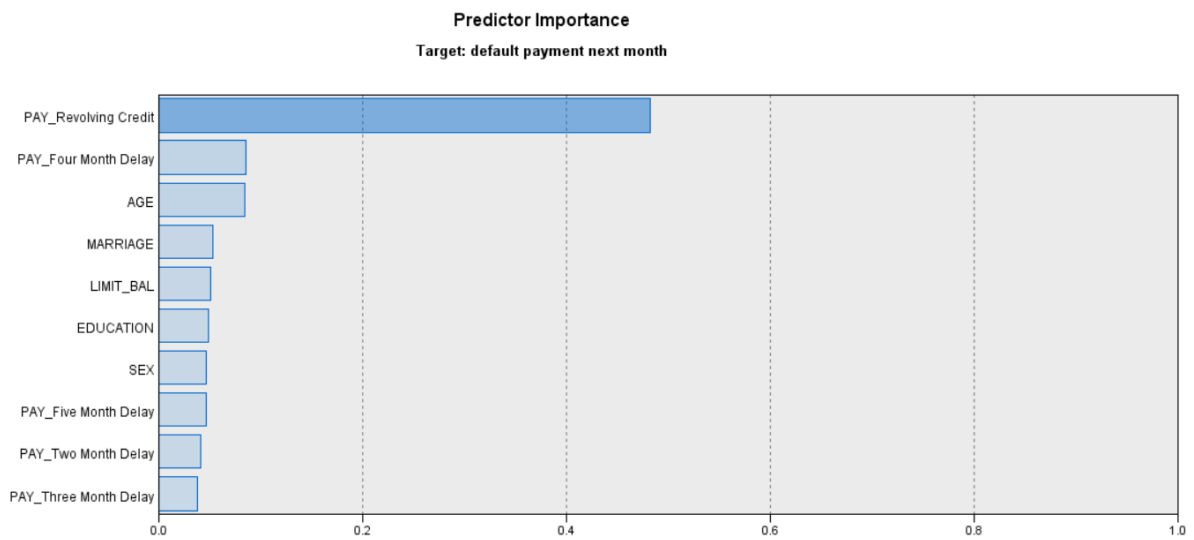


Figure 6: Predictor Importance

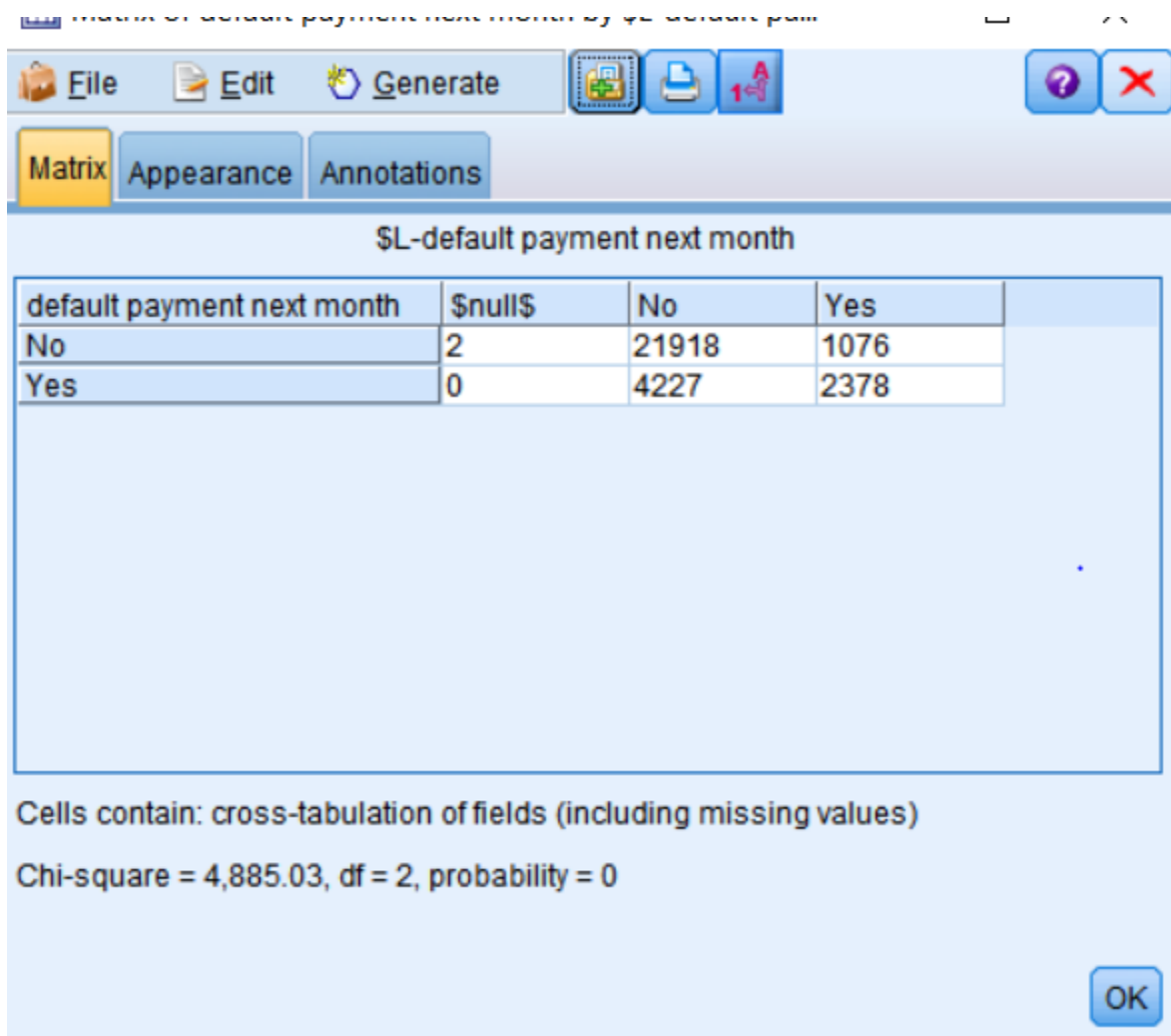


Figure 1 Logistic Regression Matrix