Default Payment Analysis of Credit Card Clients

Sunakshi Sharma[1], Vipul Mehra[2]

***Abstract*— Lending is one of the key business areas in the banking industry, credit cards as of late have seen huge success over the course of years. In pursuit to increase their market share,banks often issue credit cards to ineligible customers without adequate background checks. Also, many customers used their credit card beyond their repayment capabilities leading to high debt accumulation. Identifying the risky and non-risky customers is the biggest challenge for banks. So, the problem we are trying to analyze is how to identify the risky and non-risky customers, helping the bank to decide if a customer has the potential to repay the used credit of the bank [1].**

I. INTRODUCTION

A. Motivation

The motivation of the project was to provide any banking organizations to find a simple and an effective predictive model for the banks to determine if their customers could make the credit card payments on-time. The banks with the invent of credit card were more focused on the number of customers using their credit service but the drawback of them not being able to pay back the credit in time was an issue that soon followed, a system was in need to effectively decide the credit limit to be allowed to a person based on his previous credit history, concentrating on this issue we were motivated to consider various parameters such as sex, age , level of education along with customer details and the credit history of the past 6 months to give a reliable and effective prediction if a customer is able to pay the credit for the next(7th) month.

B. Report Organization

The report is organized to present the problems faced by the banks when a credit card is being issued, considering a simulated data, from a reliable data source we explain the characteristics of the data. The report follows to explain the various methods to adopt to deal with anomalies such as extreme values and outliers, removal of which increases the quality of the data. The data processing is followed by refining the data to improve the performance of the various models available, this is done by applying methods such as aggregation and discretization, which aims at the removal of irrelevant attributes which do not contribute towards the class values and discretizing the continuous values into discreet intervals known as bins. We then report the various test methods such as Training set, n fold Cross validation and Percentage split for every classification algorithms used to test the data. We show a statistical analysis for classification algorithms such as JRip, J48, Logistic Regression and Random Forest. By the statistical report we decide the best algorithm for the data set we have chosen. After the report about the data mining approach we move on to the logic of problems where we discuss about the merits and summarizes the problem in focus with various parameters. To conclude the report we specify the problems faced during data mining and discuss the future scope of this project.

[1]S. Sharma is a Graduate Student from the Department of Information Science and Technology, Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY 14623-5603, USA sxs7581 at rit.edu

[2]V. Mehra is a Graduate Student from the Department of Information Science and Technology, Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY 14623-5603, USA vm8176 at rit.edu

II. DATA EXPLORATION

A. Data Source

The data was extracted from the UCI Machine learning Repository, the file extracted was in the CSV format, the data was analyzed and concluded that only certain attributes contribute towards the class values which was processed in the later stages of the project. Data summary can be seen in Appendix A titled "Data Summary" [2].

III. METHODOLOGY

A. Data Pre-Processing

1) Missing Values: Since there were no missing values, so there was no need to remove or impute them.

2) Discretization: Using the Discretization function from attribute filters on the 24 attributes, we create a minimum of 10 bins for all the attributes limiting the number of possible states, where in the buckets are treated as ordered and discrete values, because of which the accuracy in all the algorithms increase which can be seen in Table 3 in appendix. As seen in the statistics report the improvement of accuracy is small as we have transformed continuous variables to discrete values. On the other hand,if we need sufficient increment in accuracy we need to do variable selection.

3) Feature/Attribute Selection: Using Weka we tried to find out the attributes that are least contributing towards the class value. To do that initially we loaded the data which is now free from extreme values and outliers (Training data) and by clicking the Select attribute tab in Weka we used InfoGainAttributeEval along with the Ranker. This gives us the features that are Ranked attributes by their individual evaluations.

After running the ranker filter and discretization we have the following attributes to be removed and the algorithm is tested for its accuracy to know the eect of these attributes on the models performance. Attributes: SEX, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_MT6. The statistical report of the model performance when we remove these

attributes on the discretized data is shown below in Table 2 & Table 4 in Appendix A (Data−Summary).

4) Aggregation and Removal of irrelevant features: There was marginal change in accuracy when the above least ranked attributes were removed from the data set but, when we consider the real-world banking scenario history of payments and demographics of clients is needed for default payment analysis of credit card clients. So, all these attributes were retained. We didnt perform aggregation on our data set.

5) Correlation Matrix: We also plotted the correlation matrix in python to analyze the relationship of attributes with the class value to determine if the attribute is highly correlated or not. After looking at the correlation matrix Figure 3 (Appendix A − Data Summary) we concluded that attributes like Pay_0, Pay_2, Pay_3, Pay_4, Pay_5, Pay_6 are highly correlated to the class value. These attributes represent the history of the past payment of a credit card holder wherein Pay_0 to Pay_6 shows the payment status of each month. The numeric value in these attributes showed the past history of a credit card holder, example -2 means: No consumption of credit card, -1 means that holder paid the full balance, 0 means the use of revolving credit; 1= payment delay of one month; 2= payment delay of two months and so on. Removing these attributes reduced our accuracy from 81% to 77% which certainly showed that these attributes are helpful in deciding the credit card defaulter.

B. Mining the data

1) J48: J48 is a re-implementation of C4.5 release 8. As C4.5 is also a decision tree algorithm, the difference is it allows multi-way splits. To select attributes C4.5 uses information gain. Gain ration measure is used to identify the splitting criteria. The splitting stops when number of splits reach a particular threshold. After the tree is grown, pruning is done which is mostly error-based. For our project the hyper parameters used are batchSize:100, binarySplits:False, unpruned:False, useLaplace:False, subTreeRaising:True,seed:1, numFolds:3.

2) JRIP: JRip classifier algorithm implements a proportional rule learner algorithm, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. JRip uses sequential covering algorithms for creating ordered rule lists. This algorithm has 4 stages:

1. Growing a rule: grow one rule by adding condition to the rule until the rule is perfect.
2. Pruning: Prune every rule incrementally and allow pruning of any final sequences of the antecedents.
3. Optimization: When the initial rules are generated, generate and prune two variants of each rule from randomized data using the above 2 procedures.
4. Delete the rules from the ruleset that would increase the descrition length of the whole ruleset if it were in it. Parameters that we used are batchSize:100, usePruning:True, folds:3, seed:1.

3) Naive Bayes: Naive Bayes algorithm is based on Bayes Theorem and takes into consideration strong assumption of independence of the predictor variables. Which simply means if we know value of one attribute it will not affect the other. This assumption of independence makes it naive. Despite being so simple, naive bayes has most of the time performed better than other algorithms [9]. Mathematically let us assume: evidence = x_1, x_2, x_n,. .., x_n They are all conditionally independent given outcome = y_i. Therefore, Naive Bayes can be written based on bayes theorem:

$$p(outcome \mid evidence) = \frac{\frac{p(likelihood\ of\ evidence) \times prior\ probability\ of\ outcome}{p(evidence)}}{= \frac{p(evidence \mid outcome) \times p(outcome)}{p(evidence)}}$$

Our parameters includes batchSize:100, numdecimalPlaces:2, useKernelEstimator:False, useSupervisedDiscretization:False.

4) Logistic Regression: Logistic regression is a machine learning technique that has been taken from the field of statistics. It is used for binary classification. Problems that are confined to two classes. Or we can simply say for a feature x_i a class $y_i \in$ [0, 1]. In logistic regression the hypothesis is that the conditional probability p of class belongs to "1" if probability is greater than threshold probability, generally 0.5, else it belongs to the class "0".

$$y_i = \begin{cases} 1 & p \geq 0.5 \\ 0 & p < 0.5 \end{cases}$$

Where

$$p(y = 1|\ \boldsymbol{x}) = h_\theta(\boldsymbol{x}) = \frac{1}{1 + exp\left(-\theta^T \boldsymbol{x}\right)} = \sigma\left(\theta^T.\boldsymbol{x}\right) \quad (1)$$

$$p(y = 0|\ \boldsymbol{x}) = 1 - p(y = 1|\ \boldsymbol{x}) = 1 - h_\theta(\boldsymbol{x}) \quad (2)$$

Here $\theta(z)$ is called sigmoid function or simply the logistic function. Which is a S-Shaped function and conforms the value of the θ^T.x in the range [0,1].And hence we can understand it in terms of probability distribution[4]. For our data the parameters used are batchSize:100, ridge:1.0E-8, useConjugateGradientDescent:false

C. Re-sampling Techniques

We have used 2 re-sampling techniques for this data-set. And then used different performance metrics on each one of them.

1) Percentage Split: We split the data-set into training and validation set. This helps us train the model on unseen data, and then use the rest of the data for validating our model. Though it is computationally not intensive, but this approach has problems to it. In machine learning or data-mining more the data (not noise), better is the model, but when we do the split the model is trained on the training set, and rest of the validation set is wasted. So in order to improve our model and best utilize the whole data-set, we use the approach of K-Cross Validation. Which is now-a-days gold standard in the industry. Here we have used the default 66%−33% train-test split.

2) K-Cross Validation: Because machine learning offers a high level of modeling freedom, it tends to overfit the data. A model overfits when it performs well on the training data but does not perform well on the validation data. k-cross validation helps minimizing overfitting. In k-cross validation approach the data is split into k equal folds. 1 fold is converted into validation set and (k-1) left folds are then used as training set. This goes through k iterations. And each time out of that k folds, 1 is hold out as validation set and rest as training. k-folds also helps in smoothing out noisy or random data. Also, this way we are able to utilize the whole data-set as training, and we are able to test on whole data-set as well[8].

This helps in making the model as well rounded, which is not just biased towards particular training or validation set. Here we have used 10-Fold Cross Validation, k = 10.

D. Performance Measures

1) Confusion Matrix: The confusion matrix for this model is:

	a=0	b=1
a=0	**TN**	**FP**
b=1	**FN**	**TP**

Naive Bayes		
	a=0	b=1
a=0	9836	1741
b=1	1903	1859

Logistic Regression		
	a=0	b=1
a=0	11064	513
b=1	2702	1060

JRIP		
	a=0	b=1
a=0	1085	692
b=1	2411	1351

J48		
	a=0	b=1
a=0	10436	1141
b=1	2295	1467

2) Performance Measure Table: The table below is the Performance Measure Table.

Performance	**Naive Bayes**	**Logistic Regression**	**JRIP**	**J48**
Percentage Split **Accuracy**	77.3154	79.7699	80.42	78.92
Percentage Split **ROC**	0.722	0.723	0.661	0.646
Percentage Split **F-Measure**	0.771	0.767	0.785	0.771
Cross Validation **Accuracy**	76.2436	79.04	79.7705	77.59
Cross Validation **ROC**	0.712	0.722	0.657	0.652
Cross Validation **F-Measure**	0.762	0.756	0.775	0.761

3) Accuracy: Classification accuracy is the number of correct predictions made as a ratio of all predictions made. One of the most common metric and only suitable when the data-set is balanced containing an equal number of observations in each class. Take for example a simple case of a balanced dataset where:

True Positives(TP) = 3, False Positives(FP) = 2, True Negatives(TN) = 2 and False Negatives(FN) = 3.

Accuracy =

$$\frac{TP+TN}{TP+FP+TN+FN}$$

=

$$\frac{3+2}{3+2+2+3}$$

= 0.5

The 0.5 value is acceptable as the dataset is balanced. The number of TP and FN are equal. But what if our classifier is unbalanced? In that case let us assume:

True Positives(TP) = 15, False Positives(FP) = 30, True Negatives(TN) = 150 and False Negatives(FN) = 20.

Accuracy =

$$\frac{TP+TN}{TP+FP+TN+FN}$$

=

$$\frac{15+150}{15+30+150+20}$$

= 0.76744

We can say that this nearly detected 76.74% of credit card defaults in our case.

Now what if we take into account a biased classifier which only detects "No credit default"? In that case our values will be:

True Positives(TP) = 0, False Positives(FP) = 0, True Negatives(TN) = 150 and False Negatives(FN) = 20.

Accuracy =

$$\frac{TP+TN}{TP+FP+TN+FN}$$

=

$$\frac{0+150}{0+0+150+20}$$

= 0.8823

Interestingly our accuracy is now 88.23% which has now increased, even with a model that is useless for predictions. Which proves that accuracy is not solely a good measure. Rather we can take accuracy altogether with other measures like precision, recall, F-measure and ROC curve.

Accuracy is known to increase when TP <FP for the negative rule. Also, similarly TN <FN for positive rule. This is known as accuracy paradox[2].

In our case we can see from the above table that JRIP and Logistic Regression perform the best, with 79.7705% and 79.04% respectively, on 10 cross validation. But since accuracy is not the best measure, when it comes to unbalanced data-set. We can also see from the confusion matrix above. We have to maximize our True Positives to catch the maximum fraud or default. For this, we will look at other measures, like sensitivity, precision, ROC curve and F-measure.

4) ROC Curve: ROC is a receiver operating characteristics curve which is used for the binary classification problems. ROC curve is a graphical plot between the true positive rate also called as recall and false positive rate. AUC stands for the Area under the ROC curve. If the models prediction is 100% correct then the AUC will be 1 and if the prediction is wrong then it will be 0. A binary classification problem is really a trade-off between sensitivity and specificity.
Sensitivity is the true positive rate also called the recall. It is the number of instances from the positive (first) class that actually predicted correctly. Specificity is also called the true negative rate.
Is the number of instances from the negative (second) class that were actually predicted correctly. More the Area under the curve (AUC), better is the prediction. The maximum can be 1 and worst is 0. AUC for Naive Bayes is 0.7122.

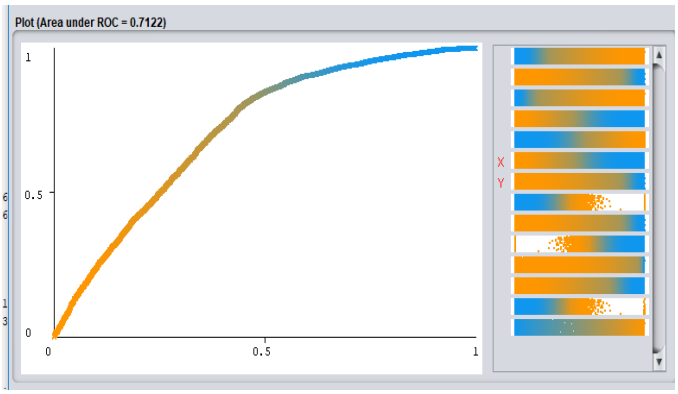

Figure 1: ROC curve for Naive Bayes

Similarly for Logistic Regression it can be seen around 0.722 which is the highest and JRIP and J48 could not perform well on this metric.

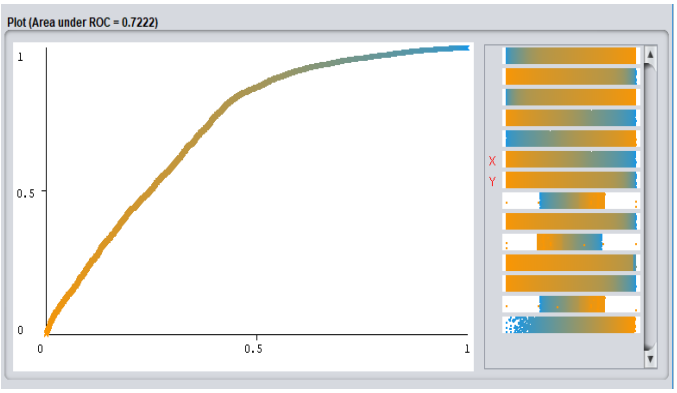

Figure 2: ROC curve for Logistic Regression

5) Precision and Recall: In the case of a system like this, we need high recall and reasonable precision and since our dataset is unbalanced, therefore our model selection is based on ROC curve, F Measure, Accuracy and Precision and Recall combined. For our purpose high recall means high sensitivity and will be able to maximize on the true positives we want to catch maximum defaulters. Where as a high precision it will be beneficial in terms of business perspective, because sometimes model may make mistake, and we might catch False Positives, hence customer can get annoyed. So a balance between both is required.
Precision is the ratio of correctly predicted positive values to the total predicted positive values. This metric highlights the correct positive predictions out of all the positive predictions. High precision indicates low false positive rate and is given by: **Precision** =

$$\frac{TP}{TP+FP}$$

The best overall weighted precision is of logistic regression which is 0.553, following logistic regression is JRIP with 0.498 then J48 with 0.485 and then lastly Naive Bayes with 0.419.
The recall is the ratio of correctly predicted positive values to the actual positive values. Recall highlights the sensitivity of the algorithm i.e. out of all the actual positives how many were caught by the program.
recall =

$$\frac{TP}{TP+FN}$$

Interestingly JRIP scored the best in this department with a recall of 0.780 followed by Logistic Regression at 0.772 then Naive Bayes at 0.759. J48 score the least at 0.757.

6) F-Measure: It is a balance between precision and recall. Put another way, the F1 score conveys the balance between the precision and the recall. And is given by:
$2 \times$

$$\frac{precision \times recall}{precision + recall}$$

The F1 score is a good way to summarize the evaluation in a single number, but its always a good practice to look at both precision and recall together to better understand how a classifier behaves.
After observing the F-Measure it can be seen that JRIP performed best here at 0.775 followed by Naive Bayes at 0.762, then J48 at 0.761 and the least by Logistic Regression at 0.756. All very close.

E. Pros and Cons

The best part of this tool is it will help in organizing our purpose, concepts, information, implications and inference before proceeding with analysis or building a project. It is very simple and easy to use. It functions in a systematic way, it helps the user in not deviating from the set goals and path. Also, it helps in creating a timeline like, what steps needs to be in what phase of analysis or project. Every section of the logic of problem comes with a set of questions, which must be answered before filling that section. This helps the user in answering all the sections with precision. The major disadvantage of this tool is that sometimes the user interface does not work as it is supposed to be, which might annoy the user.

F. Recommendations

Even though the tool is very helpful and serves its purpose, one major improvement that can be added to this tool would be fixing a time schedule to solve every problem. Because, the user might be having multiple activities running at the same time so, giving a time frame to solve every problem will help the user track and solve all the problems within stipulated time. Adding this function to the tool will make it more user-friendly.

IV. CONCLUSIONS

The whole exercise analyzes the performance of various algorithms to calculate the risk of person being defaulting the credit card payment. The performance of algorithms are measured using various metrics. Detecting credit card fraud (default) is a tricky problem in terms of business perspective. To catch maximum defaulters (True Positives) and minimize (False Positives) without hurting the sentiments of the customers. Model selection solely based on accuracy does not make much sense because the data-set is unbalanced and skewed. Therefore sensitivity or recall (True Positive Rate) along with precision of the model is of importance. High recall means more defaulters being caught. And reasonably high precision means less False Positives. A good model is a balance between high recall and reasonable high precision. So F-Measure and ROC curve are the metrics we are actually looking for along with Precision and Recall. Interestingly JRIP and Logistic Regression performed fairly well when F-Measure, Precision and Recall is considered for 10-cross validation. But JRIP and J48 failed miserably when it came to ROC-AUC where Logistic Regression topped and a close second spot was taken by Naive Bayes. Our analysis tells that Logistic Regression is better suited for this problem. Unlike other algorithms like JRIP, J48 and Naive Bayes, Logistic Regression has been consistent throughout all the performance measures and has been fairly close in the ones where it did not top. The future work for this project can include using neural network and support vector machine as an alternative data mining methodologies to actually see if the true positive rate for predicting the defaulters is increasing along with the accuracy of the model. We could also apply the bining of the ages to segment the customers according to their age range. This can certainly help us figure out that what can be the distinctive features of a person who is been classified as a defaulter. This can help the banks to make the informed decisions, for instance what all features need to be included in a person in order to issue a credit card or judge the eligibility or repaying capacity of a person.

APPENDIX

A. Data Summary

Table 4(the last table) provides the Summary of Analysis.

ACKNOWLEDGMENT

We would like to thank Dr.Jai Kang for his help and suggestions throughout this entire work.

Table 1: Data Summary

Total Number of Records: 30,000		Number of Attributes: 24
Data Set Characteristic: Multivariate		Characteristics of Attributes: Integer, Real
Associated Tasks: Classification		Missing Values: 0
Number of Records in Training Data Set: 20,000		Number of Records in Test Data Set: 10,000
Outliers	Number of records without outliers:**16972** Number of records with outliers:**3030**	
Extreme Values	Number of records with extreme values:**17373** Number of records without extreme values:**2629**	Number of Records in Test Data Set: 10,000
Number of records in Training Data Set after Processing:**15339**		
Data format: CSV (converted to ARFF)		

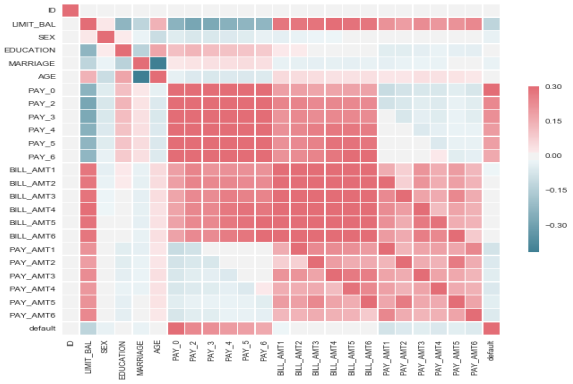

Figure 3: Correlation Plot

Table 2: Attributes removed - Algorithm Performance

Algorithm	BILL_AMT6	BILL_AMT6+4	BILL_AMT6+4+5	BILL_AMT6+4+5+3	BILL_AMT6+4+5+3+SEX
JRIP	79.97%	79.86%	79.86%	79.81%	80.07%
J48	79.12%	79.11%	79.15%	79.21%	79.83%
Naive Bayes	77.59%	77.69%	77.63%	77.75%	77.74%
Logistic	78.53%	78.62%	78.55%	78.70%	78.74%

Table 3: Data Statistics Analysis

Algorithms	Correctly Classified before discretization	Incorrectly classified after discretization	Correctly classified after discretization	Incorrectly classified before discretization
ZeroR	1157(75.4743%)	3762(24.5257%)	11577(75.4743%)	3762(24.5257%)
OneR	12280(80.0574%)	3059(19.9426%)	12280(80.0574%)	3059(19.9426%)
JRIP	12236(79.7705%)	3103(20.2295%)	12256(79.9009%)	3083(20.0991%)
J48	11903(77.5996%)	3436(22.4004%)	12271(79.9987%)	3068(20.0013%)
Naive Bayes	11695(76.2436%)	3644(23.7564%)	11893(77.5344%)	3446(22.4656%)
Logistic	12124(79.0404)	3215(20.9596)	12053(78.5775%)	3286(21.4225%)

Algorithm	Discretization + Attribute Removal
JRIP	80.7%
J48	79.83%
Naive Bayes	77.74%
Logistic	78.75%

REFERENCES

[1] I.-C.Yeh, default of credit card clients data set. https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients. Accessed on 4−23−2018.

[2] Wikipedia, Accuracy paradox. https://en.wikipedia.org/wiki/Accuracy_paradox. Accessed on 4−23−2018.

[3] R. Johri, Small classification. http://web.stanford.edu/ rjohari/teaching/notes/226_ lecture8_prediction.pdf. Accessed on 4-23-2018.

[4] C. Y. F. Y. M. C. S. A. C. A. M. A. H. B. H. T. W. S. T. Andrew Ng, Jiquan Ngiam, Logistic regression. http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/. Accessed on 4-23-2018.

[5] J. Brownlee, Machine learning mastery. https://machinelearningmastery.com/blog/. Accessed on 4-23-2018.

[6] D. L. Nikhil Subba, Rising credit card delinquencies to add to u.s. banks worries. https://www.reuters.com/article/us-usa-creditcards-delinquencies/ rising-credit-card-delinquencies-to-add-to-u-s-banks-worries-idUSKCN1BQ2E0. Accessed on 4-23-2018.

[7] M. Waskom, Plotting a diagonal correlation matrix. https://seaborn.pydata.org/examples/many_pairwise_correlations.html. Accessed on 4-23-2018.

[8] N. Krupa, Testing machine learning algorithms with kfold cross validation. https://www.talend.com/blog/2017/05/15/ machine-learning-algorithms-with-k-fold-cross-validation/. Accessed on 4-23-2018.

[9] S. Gollapudi, Practical Machine Learning. Packt Publishing, 2016.