Big Data and Credit Defaults – An Analysis

Ciaran Egan
National College of Ireland
Dublin IE
x19163568@Student.ncirl.ie

***Abstract*—The following paper investigates how Big Data processing can be used in the analytical field of credit scoring. Two datasets are taken, one with default information on Vehicle Loans and one with default information on Mortgages. The datasets were cleaned, combined and analyzed using Hadoop map-reduce. A logistic regression was then trained on the data. The whole process was set up using a Linux shell script. This script was then deployed on an Openstack cloud virtual machine. It was proved that big data technologies such as cloud computing and map-reduce can be used to analyze and model credit defaults. Some valuable insights from the data were discovered such as how mortgages and vehicle loans have contrasting default rate distributions and how the age of the applicant is a good predictor of credit quality among others. A Logistic regression-based credit scorecard model was successfully built on the derived data. This paper will serve as a good benchmark if banks or lending institutions ever decide to incorporate Big-Data technologies to their usual credit scoring processes.**

I. Introduction

Banks and Financial Institutions are heavily exposed to default risk. Default risk is the inability of obligors to meet their contractual repayments on their debt obligation. Default risk is the largest operational expense that banks face. Banks become insolvent when many of their obligors cannot meet their contractual repayments. The barriers to entry in most banking markets are significant and as a result, many banks operate in markets with a small number of competitors. These barriers to entry and the lack of fluidity in banking markets mean that many banks meet the "too big to fail" criteria and the failure of a large bank can result in wide-ranging adverse systemic economic effects. It because of this that many banks invest heavily in data-driven models to understand and forecast default risk. Because of the adverse economic effects that can result from banks facing default risk, there exists a lot of regulatory pressure on banks to build effective and explainable models.

The data that banks use to build these models is growing. As time goes on, more observations are becoming available to banks and banks are seeing a much wider range of consumer behaviour. Banks are also exploring the option of sharing data. There is also the option of using unconventional (by banking standards) data sources such as mobile phone and social media data. All of this will increase computational cost and as a result, traditional data processing methods will not be suitable or efficient enough to analyse the data and build the credit risk models that banks require.

The aim of this project is to take datasets of loans given to obligors and then analyse model them using map-reduce parallel processing. It will be investigated if the customers can be profiled using map-reduce such that riskier obligors can be identified. The Weights of Evidence (WoE) of the groups will be calculated and a logistic regression will be trained on the resulting dataset. This type of research will be beneficial to banks if they ever do decide to embrace big data whether it comes from regulatory pressure or otherwise.

II. Data

The datasets had to be picked such that they are related and complement each other in some way such that they can eventually be combined. The decision was taken to acquire two datasets that describe different loan types. Two datasets were picked:

- **Home Credit Default Risk** - This is data from an institution known as "Home Credit". This is mortgage data from an institution that specializes in loans to consumers with sparse or non-existent credit histories [1]
- **Vehicle Loan Default Prediction** – This is data about vehicle loans. It is not indicated what type of institution this data is from. [2]

These datasets are sourced from Kaggle. They will be extracted and cleaned.

The Home Credit Default Risk dataset was set up as a Kaggle competition with a significant competition prize. The 1st place solution [3] was an ensemble of methods. These methods included Neural Networks, Extra Tree and a Hill Climber linear model. Although this would provide a very effective forecast, it would not be accepted by most regulatory authorities because of a lack of explain ability. Big data processing such as map-reduce were not used anywhere in this competition which means that these methods might not be very efficient to use on a much larger dataset.

The Vehicle Loan Default Prediction dataset was not offered as a Kaggle competition but rather was sourced from a Hackathon competition. The dataset was from an Indian bank. Work done on this data was difficult to find. It was not possible to find any analysis done on this data from a reputable source.

Overall, the literature is very sparse for big data in credit scoring. It was investigated in "The Value of Big Data for Credit Scoring: Enhancing Financial Inclusion using Mobile Phone Data and Social Network Analytics" [4] . Here the author explores the possibility of using mobile phone data as a method of assessing creditworthiness or if the obligor is likely to default. Such data would be large and traditional processing methods would not be appropriate for analysing data of this magnitude.

III. Methodology

A. Data Quality

The data used was assessed for standards as defined by SO/IEC 25012 standard [5]. Both datasets are open-sourced and publicly available. The data quality was assessed under the following headings:

- Accuracy

- Completeness
- Consistency

It can be confirmed that both datasets are specifically intended for purpose and meet the above criteria. The Data cleaning and conforming process was done such that the resulting dataset met these standards.

B. Data Cleaning

The datasets were extracted and required cleaning. Before any cleaning commenced, both datasets were checked for a unique ID variable. The cleaning process for each data was as follows:

1) Home Credit Default Risk – Cleaning

The first step for the cleaning process of this dataset was to identify columns with missing values. For character variables It was assumed that missing values can be left as they are, however when grouping these variables, Missing values must be treated as a separate group. For the numeric variables, the missing values had to be filled. For highly correlated variables such as the annual repayment amount and the obligor's income, a linear regression was used to infill the missing values. Given that there was a relatively small number of observations in these variables that were blank, it was safe to assume that this was appropriate method of imputation. There were three variables in this dataset that represented a normalized external credit score. Each one of these has a significant number of missing values. To remedy this, the three external score variables were combined. Where there was a missing value in one of the external score columns, the average of the other two external score columns was used. After a single external score column was created, there remained a small number of missing values. To remedy these missing values, the average credit score split by the default indicator (the target variable) was imputed. It is probably not wise to use the target variable as a method to impute the predictor variables but given that a small number of variables remained to be imputed, the fact that external score is an indicator of default risk and the aim of the project this was deemed to be appropriate. This dataset contained a few features that described the same thing. An example of this would be the number of doors in the house or the size of the living area. There are three columns that describe variables such as this and these are highly correlated and as a result only one of them will be necessary. These variables have a lot of missing values, so much so that they were unable to be imputed. For binary variables such as "Observations in social circle where default occurred", missing values were assumed as 0 indicating that no on in the social circle could be identified as going into default. For other variables where there was a small number of observations missing, the median was taken.

2) Vehicle Loan Default Prediction – Cleaning

There were not many missing values observed in this dataset. The cleaning process involved reformatting some of the variables. All date columns were reformatted to be readable and consistent and some such as age were changed from character to numeric. The variable that represent the external credit score of the obligor was normalized.

C. Combining the two Datasets

When the datasets were cleaned, they were then combined to create a larger dataset with more rows. To do this, some of the features needed to be transformed so that they can match each other in the combination phase of processing. Each dataset was assigned a portfolio name to identify which portfolio the observation related to when the datasets were combined. For the Mortgage dataset the annual repayment amount had to be transformed to monthly, while in the vehicle loans dataset the instalment amount had to be converted from Rupees to USD. The date of birth in both data sets was transformed to get the obligors age at application. To Identify the IDs by portfolio and to ensure that no duplication of ID occurred, the IDs in the Vehicle loans dataset were multiplied by -1. For the Mortgages Dataset, a Loan-to-Value LTV feature was created by dividing the Loan amount by the underlying property value. The External score was normalized for the Mortgages dataset. The two datasets were then stacked on top of each other with only the columns common to both datasets remaining. The columns remaining are as follows:

1. ID – The unique identifier
2. DEFAULT_IND – A binary indicator that indicates whether the loan is in default or not
3. PORTFOLIO – An indicator whether the loan is a vehicle loan or a mortgage
4. LTV – the ratio of the loan amount to the value of the underlying asset
5. INSTLMNT_AMT – The monthly instalment paid by the obligor
6. AGE – the Age in years of the customer
7. EXT_SCORE – A normalised value indicating the External credit score of the obligor

The resulting dataset will serve as the Analytical Base Table (ABT) for the Map-Reduce procedure and will be exclusively used for any further analysis.

IV. Implementation and Architecture

The following describes how the analysis was built and the structure surrounding it.

A. Application Workflow and Tools Used

The data cleaning procedure was done using R data-frames. R was chosen because it offers a simple and intuitive way to clean the data however R data-frames might not be effective if the dataset becomes very large. This is because R loads the entire dataset into memory. Ideally an alternative method of cleaning the dataset must be used. For the purposes of this research, R data frames were deemed suitable for cleaning (and eventually combining) the data.

The main tool used to perform the analysis was Hadoop. Hadoop provides a software framework for distributed storage and processing of big data using the map-reduce programming model. Map-reduce allows for efficient parallel processing. This type of parallel processing allows for quick and efficient processing of large datasets. The project was developed using a localized Linux virtual machine. This was a user-friendly development environment where various IDEs were utilized. To deploy the project, a Linux script was created with the intention that is could be run on a variety of different machines. This means that the analysis can be easily executed on a wide range of platforms including cloud computers assuming the right applications are pre-installed. To run the analysis, cloud platforms such as Openstack or AWS were chosen. Both cloud platforms offer various Ubuntu command

line interfaces which meant that the script could be easily deployed. The overall application workflow is as follows:

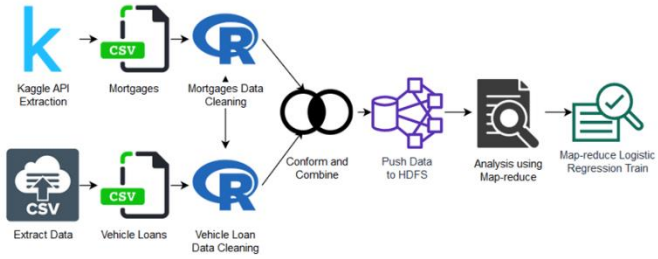

Figure 1: Overall Project Process Flow

The Mortgages data was extracted using the Kaggle API [6]. Although the Vehicle loans data was discovered in Kaggle, it was extracted using a Github link [7]. The resulting CSV's were then cleaned and combined using R. The resulting data was saved as a CSV and then pushed to the Hadoop Distributed File System (HDFS). From there the map-reduce operations to group the data and train a logistic regression.

B. Analysis

The aim was to find out if a standard credit scorecard model could be built using the map reduce programming model. This involves two steps:

1. Binning each variable and calculating its default rate and its Weight of Evidence (WoE).
2. Training a logistic regression on the calculated weights of evidence

The weight of evidence is a measure of the separation of good and bad customers. It is as follows:

$$WoE = \ln\left(\frac{Distribution\ of\ Good\ Customers}{Distribution\ of\ Bad\ Customers}\right)$$

The weight of evidence is calculated for each bin. The right binning chosen is subjective and no single method exists that is widely used across all banks. However, the following rules of thumb apply when applying a binning method:

1. Bins must be monotonic.
2. Missing values are grouped together.

It was investigated if the "Monotone Optimal Binning Method" discussed in the paper "Monotone optimal binning algorithm for credit risk modelling" [6] could be applied using map-reduce, however this would prove to be difficult given that this algorithm would require consecutive line-by-line processing to implement. For the purposes of this research, the decision was taken to use an iterative process for binning. This meant that arbitrary bins were tried until suitable bins that explain the data sufficiently were found.

These groups were summarized using a map-reduce programming structure. The structure of which is as follows:

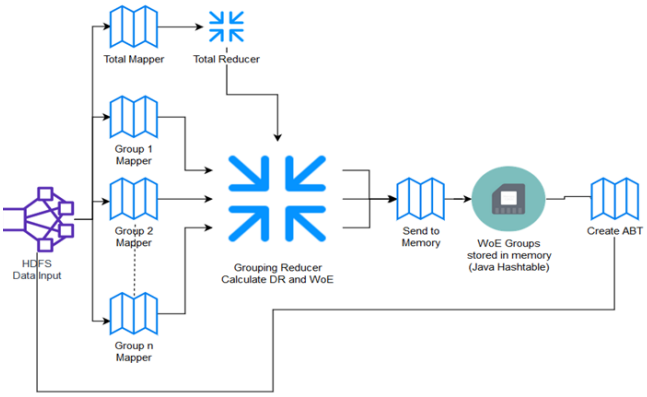

Figure 2: Overall Map-Reduce Procedure

The input data is a single table from the HDFS. The total values are calculated such as; total number of defaults, total number of accounts, maximum and minimum of certain variables are derived. The total values will be required to calculate the WoEs in the grouping stage. The bins are assigned at the mapper stage and the derived bins are the keys which feed into the reducer stage. In these stages the total number of defaults is counted. The same reducer is used repeatedly for each grouped WoE calculation. This reducer has two mapper inputs. The total input and the output from the respective grouping mapper. The calculated Weights of evidence are then stored in memory using a java Hashtable. Given that a small number of bins are present, it is not considered inefficient to store the derived WoEs in memory. The calculated WoEs were then applied to the input data and the ABT for the Logistic regression trainer was created.

The resulting dataset was then trained using a logistic regression also built using Hadoop map reduce. The logistic regression code used for this was leveraged heavily off the logistic regression mapper and reducer in [8]. The code uses the gradient descent algorithm to optimize the coefficient values. The code such that the training set is taken from the HDFS, and a logistic regression is trained given a learning rate parameter α and the number of times to iterate over the training set as inputs.

V. Results

A. Deployment

Once the shell script and all R and Java sub-components were prepared, it was time to run this process on the cloud. The original plan was to use an AWS virtual machine to deploy the project. This did not work because the free tier that was available did not have enough Random-Access Memory to complete the end-to-end task. The issue was driven by the data preparation component of the project which was done using R. R data-frames are loaded into entirely memory and given that the input data was large, the AWS session crashed. Openstack was used instead because there was access available to cloud virtual machines with more RAM. The shell script ran successfully on the Openstack virtual machine.

The results of this show that map-reduce can be used to analyse bank default data and hence build a credit scorecard model off it.

B. Insights

After the analysis was done the derived bins are as follows:

Feature	Bin	Default Rate	Weight of Evidence
LTV and Portfolio	Mortgage: LTV less than 113%	6.56%	0.8362
	Mortgage: LTV between 113% and 120%	8.64%	0.5394
	Mortgage: LTV greater than 120%	11.71%	0.2005
	Vehicle Loan: LTV less than 66%	15.01%	0.0854
	Vehicle Loan: LTV between 66% and 78%	20.47%	-0.4620
	Vehicle Loan: LTV greater than 78%	25.65%	-0.7550
Monthly Instalment	Greater than 750	8.33%	0.5797
	Less than 750	21.02%	-0.4953
Age	Less than 35	18.22%	-0.3175
	Between 35 and 55	12.43%	0.1328
	Greater than 55	6.5%	0.8419
Normalized External Score	Less than 33% (Bad or no score)	23.10%	-0.6164
	Greater than 33% (Good score)	10.22%	0.3540

Figure 3: Results of binning analysis

Valuable insights were derived from the data using the map-reduce programming framework. These are as follows:

1. **Vehicle loans have a much higher default rate that Mortgages** – This was to be expected however the difference in default rates was surprising. For Vehicle loans a default rate of ~20% was observed while for mortgages the default rate was around 8%. This information would be very useful for banks as it would mean that it would be beneficial to apply varying credit policies between mortgages and vehicle loans.

2. **The older the applicant is the less likely they are to go into default** - This is an interesting insight and using this information, banks can use this to target older customers. Banks can also use this to charge a higher default premium on credit given to younger customers.

3. **The loan to value (LTV) distribution between mortgages and vehicle loans was so different that the model had to be adjusted to accommodate this** – The average LTV for vehicle loans was much lower than that for mortgages. This could be attributed by the fact that Vehicles tend to depreciate a lot faster. This gave problems for the model as LTV is associated with having a positive relationship with default risk. Given that Vehicle loan default rates are much higher than Mortgage default rates, binning by LTV alone would not result in monotonic behaviour. To remedy this, the decision was taken to group the LTV and Portfolio variables together.

4. **External credit scores do play a role in measuring credit quality** – The best way to bin external credit score was to split between normalized credit scores of above 33% and below 33%. A more granular split between external scores was expected from this analysis however it could be argued that this still is useful in determining credit quality.

The resulting regression all had coefficients with negative signs. This was as expected given that WoE has a negative relationship with credit defaults. The chosen learning rate parameter was 0.01 and the training set was iterated over 200 times.

VI. Conclusions And Future Work

The overall results of this project are successful. It was proved that big-data applications such as cloud-computing and Hadoop map-reduce can be used to analyse and model bank defaults. A logistic regression model was successfully trained. This research will become useful if banks ever do decide to use big-data technologies. Some valuable insights were derived from this analysis and they could be used to drive important banking business decisions.

If such a project was undertaken again, different decisions would have been made. R data-frames would not have been used to clean and combine the data. Although they worked for this analysis, memory issues would have arisen if the input datasets were much larger. A processing method that does not load the whole dataset into memory would have been considered.

Future work hopes to build on this analysis. Examples of could be investigating if a standardized binning algorithm could be implemented using map-reduce. It would be desirable to test and validate the derived logistic regression model and see if a better model can be derived.

Acknowledgments

I would like to acknowledge my lecturer Luis Gustavo Nardin, who has taught me everything related to cloud-computing, Linux and Hadoop mentioned in this paper. He has helped me with every technical issue I have presented, and this project could not have been completed without him. I would also like to thank my colleague Stephen McMullan who has helped me with various Java-related issues and has been very responsive to my class forum posts.

References

[1] H. C. Group, "Home Credit Default Risk," [Online]. Available: https://www.kaggle.com/c/home-credit-default-risk.

[2] A. Paul, "Vehicle Loan Default Prediction," [Online]. Available: https://www.kaggle.com/avikpaul4u/vehicle-loan-default-prediction.

[3] Kaggle, "Kaggle Home Credit Default Risk: 1st Place Solution," [Online]. Available: https://www.kaggle.com/c/home-credit-default-risk/discussion/64821?rvi=1#782970.

[4] M. e. a. Óskarsdóttir, "The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics," *Applied Soft Computing*, 2018.

[5] I. 25000, "ISO/IEC 2012," 2019. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25012.

[6] Kaggle, "Official Kaggle API," Github, [Online]. Available: https://github.com/Kaggle/kaggle-api.

[7] N. Bhavsar, "Vehicle Loans Data csv Github link," Github, [Online]. Available: https://raw.githubusercontent.com/NishantBhavsar/ltfs-vehicle-loan-default-prediction/master/input/train.csv.

[8] A. A. a. J. Bravo, "A Non-Parametric-Based ComputationallyEfficient Approach for Credit Scoring," 2019.

[9] V. T. Pavel Mironchyk, "Monotone optimal binningalgorithm for credit risk modeling," 2017.

[10] "Machine Learning Using Hadoop," Program Creek, January 2004. [Online]. Available: https://www.programcreek.com/java-api-examples/index.php?project_name=punit-naik%2FMLHadoop#.