CKME 136 Data Analytics: Capstone Course
Ryerson University
Fall 2018

Literature Review, Data Description and Approach

Richard Harman

Default Risk Analysis in the Low-Income Segment

1. Introduction

Major financial institutions do not typically service well the needs of lower income families. Lending to low-income or first-time borrowers is challenging when there is little or no credit history. This creates a segment of 'under-banked' individuals struggling to access formal, safe and fairly priced credit. For established banks, this represents a profit growth opportunity, while for new entrants offering financial services, an attractive underserved market. However, to be successful in this segment, a financial institution must find non-traditional and creative ways to identify and mitigate default risk.

The work that follows will explore a dataset from a lender operating in this under-banked segment. The dataset contains details of loan applicants at the time of application plus a target variable indicating whether the applicant ultimately defaulted (target = 1) or not (target = 0). I will look for relationships between the characteristics of a loan applicant and their risk of default. For example, are people employed in a certain industry more likely to default, does length of employment, family status or size and location of residence impact default risk? Lastly, through the use of machine learning algorithms, I will attempt to predict the likelihood of default for a new loan applicant.

2. Literature Review

A Machine Learning Approach for Predicting Bank Credit Worthiness [1]

15 models were investigated to predict customer credit card default. The models included Logistic Regression, K-Nearest Neighbor, Gaussian Naïve Bayes, Neural Networks, Support Vector Machines and multiple ensemble methods such as Random Forests, Ada Boost and Gradient Boosting. The subject dataset contained details of credit card customers from Taiwan with a binary target variable indicating whether a customer had defaulted or not. The dataset was unbalanced with 22% defaulting and 78% not defaulting. Accuracy, precision, recall, specificity and F-score were used to evaluate the models. The authors claim all models performed well, achieving accuracies from 76% to 98%, with the exception of Nearest Centroid and Gaussian Naïve Bayes. Of the 23 attributes, five with the most predictive power could be used to produce very similar results when compared to the full feature set.

A Comparison of Machine Learning Algorithms for the Prediction of Past Due Service in Commercial Credit [2]

The authors aim to predict commercial entities that will fall behind on their non-financial payment obligations (i.e. payments to suppliers, service providers, etc.). A large real-world dataset from Equifax was used containing 36 datasets each with over 11 million observations and 305 attributes. A binary target variable was created indicating whether the commercial entity was behind on payments or not. Significant data cleaning, missing value imputation and dimensionality reduction were performed to reduce the dataset to 16 independent variables. Three algorithms were tested: Logistic Regression, Decision Trees and Neural Networks. Accuracy, KS-statistic and ROC AUC were used to evaluate the models. Neural Networks produced the highest accuracy (at 93%) and Decision Trees had the highest KS-statistic (at 0.70). Both Neural Networks and Decision Trees outperformed Logistic Regression when using the ROC AUC measure. However, the actual numerical results were not shown in the paper.

Machine Learning Application in Online Lending Risk Prediction [3]

The author aims to predict defaulting loan applicants based on a combination of three datasets from an online Chinese lender, online phone records and third-party credit rating agencies. After amalgamation and removing observations with missing data, the dataset had over 8990 attributes and over 301,000 observations. A Random Forest model and an XGBoost model were trained and tested. The parameters of the Random Forest model were optimized using a precision-recall curve methodology, while the parameters of the XGBoost model were optimized using the ROC AUC measure. The prediction results of each model were compared using the KS-statistic. XGBoost outperformed Random Forest with a KS statistic of 0.72 versus 0.65.

Credit Default Mining Using Combined Machine Learning and Heuristic Approach [4]

The authors propose a two-step approach for predicting credit card default that combines machine learning with a custom rules-based algorithm (termed the “Heuristic Approach”). Machine learning is used to calculate a probability of default using static historical data, while the custom algorithm is used to calculate a probability of default based on “live” transaction data. The final probability of default is a linear combination of the two previously calculated probabilities. The dataset used appears to be the same Taiwan credit card dataset from [1]. Several algorithms were investigated for the machine learning step; the authors chose “Extremely Random Trees” (ER Trees) based on accuracy, precision, recall and F-score measures. Results of the final Heuristic Approach were compared to pure machine learning using ER Trees and the best results achieved in previous research on the Taiwan dataset. The authors claim the Heuristic Approach achieved accuracy of 93.1% while using ER Trees only had accuracy of 95.8%. Recall was 92.1% for the Heuristic Approach and 85.9% for ER Trees. The authors further claim that both approaches outperformed existing research on the Taiwan dataset for both accuracy and recall measures.

Personal Bankruptcy Prediction by Mining Credit Card Data [5]

A bankruptcy prediction system was built using credit card data from a major Canadian bank. This is the only paper reviewed that attempted to use the patterns in sequence/time-series data rather than the

more common approach of simple aggregation (which can lead to loss of information). For each attribute, the sequence data was encoded to categorical or ordinal values (e.g. if a data point in the sequence represents 'good' behavior, it was encoded as 0, if it represents 'bad' behavior it was encoded as 1, if 'worse' behavior then 2, and so on). A K-Means algorithm was applied using a custom distance measure (based on conditional probability that a sequence belongs to a particular group) to cluster the encoded sequences. Based on the resulting clusters, *'bankruptcy sequences'* were manually identified and then used to classify each sequence as either 'bankrupt' or 'not bankrupt'. Once all the attributes for each client were classified in this way, a Support Vector Machine model was built as the final bankruptcy predictor. The results of the final predictor were compared, using ROC curves, to both an SVM model built using simple aggregation of the original data and credit bureau credit-scoring. The predictor compared well to credit bureau credit-scoring with a very similar ROC curve and materially outperformed the SVM model built on simple aggregated data.

Implications of Literature Review

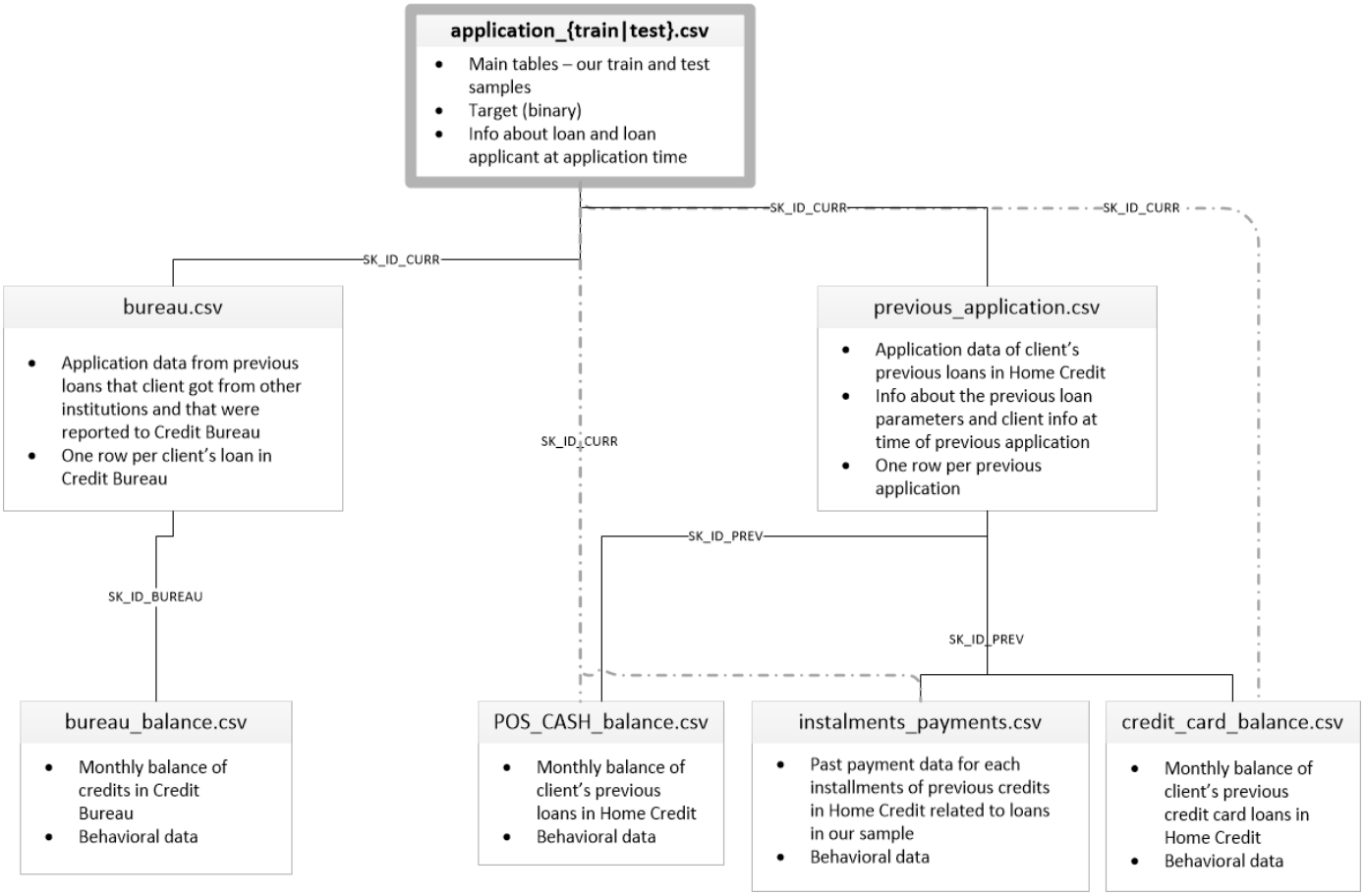
[1] showed there is a wide range of models available for default classification that are all capable of producing good results. [2] demonstrated that good prediction results can be achieved after significant dimensionality reduction of a very large dataset to a small number of attributes. While Neural Networks produced the best overall result, it was only by a small margin compared to Decisions Trees thus making Decision Trees potentially the better choice given their ease of interpretation. [3] showed the better performance of the XGBoost model versus Random Forests. [4] proposed a novel two-step approach using Extremely Random Trees that achieved good results. [5] introduced a methodology to make use of the patterns in sequential/time-series data that improved the results of a Support Vector Machine model.

Based on the review, Decision Trees and their ensemble derivatives such as Random Forests and Boosted Trees (e.g. XGBoost) appear to be the best solution for default prediction. These models show good results and are easier to interpret than Neural Networks. Lastly, the use of patterns in sequential/time-series data, as opposed to simple aggregation, can further enhance prediction results.

3. Data Description

The dataset was provided by Home Credit B.V., a lender operating in the 'under-banked' segment, for a Kaggle competition[1]. There are eight CSV files containing details of loan applicants at the time of application plus information (where available) on their historical repayment performance. Figure 1 shows a brief description of each file plus the relationships between the files.

Figure 1: Brief Descriptions and Relationships between Files



Source: https://www.kaggle.com/c/home-credit-default-risk/data

[1]Data Source: https://www.kaggle.com/c/home-credit-default-risk/data

Summary Details

application_train.csv (app)

This is the main data file that contains information about each applicant at the time of loan application. It contains attributes such as gender, type of loan, if the applicant owns a home, size and monthly payment of loan, type of employment, details of where the applicant lives, documentation provided and so on. This file also contains the target variable labeled as 1 = default and 0 = no default.

application_test.csv

This is the data used to test models when submitting to the Kaggle competition. It will not be used in this work because the data is unlabeled. That is, the target variable is missing.

bureau.csv (bureau)

Information about previous loans with other financial institutions for applicants in *application_train.csv* that have been reported to a credit bureau. For each applicant in *application_train.csv*, there may be zero, one or more entries in *bureau.csv* for each of their previous loans.

bureau_balance.csv (bureau_balance)

Basic historical information (where available) about the monthly balance of loans contained in *bureau.csv*. The information is a simple flag for each historical month (C = closed, X = unknown, 0 = nothing overdue, 1 = a payment is 0-30 days overdue, 2 = a payment is 31-60 days overdue, and so on).

previous_application.csv(prev_app)

Details on previous loan and credit card applications to Home Credit B.V. with attributes such as monthly payment, amount, purpose and interest rate of previous loan or credit card. Each applicant in *application_train.csv* may have zero, one or more entries in *previous_application.csv* for each of their previous loan or credit card applications to Home Credit B.V.

pos_cash_balance.csv (pos_cash_bal)

Status of the historical monthly balance of the previous loans in *previous_application.csv*. Information such as number of instalments left to pay, whether the loan is active or closed and whether there are payments past due is included.

instalments_payments.csv (install_pay)

Details on the repayment history of previous loans and credit cards given by Home Credit B.V. for previous applications in *previous_application.csv*. Details such as instalment amount required and instalment amount actually paid are included.

credit_card_balance.csv (credit_card_bal)

Historical monthly information on previous credit cards given by Home Credit B.V. for previous applications in *previous_application.csv*. Contains details such as outstanding monthly balance, credit card limit, monthly payment required, monthly payment actually made and so on.

A data dictionary containing descriptions of all attributes for each file was provided by Home Credit B.V. and is available at: https://www.kaggle.com/c/home-credit-default-risk/data

Table 1 shows the count of attributes and observations for each file plus a breakdown of categorical and numerical attributes before preprocessing. Table 2 shows the frequency of missing values across attributes and observations. Missing values will be addressed during preprocessing. Table 3 and Figure 2 show the highly imbalanced distribution of the target variable. This will be addressed during experiment design.

Table 1: Attribute and Observation Counts

	app	bureau	bureau_balance	prev_app	pos_cash_bal	instal_pay	credit_card_bal
Attributes	122	17	3	37	8	8	23
Observations	307511	1716428	27299925	1670214	10001358	13605401	3840312
Categorical	16	3	1	16	1	0	1
Numeric	106	14	2	21	7	8	22

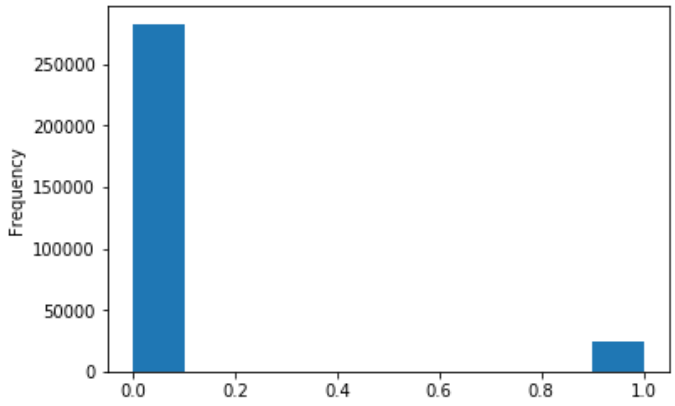
Table 2: Missing Value Counts

	app	bureau	bureau_balance	prev_app	pos_cash_bal	instal_pay	credit_card_bal
Attributes	122	17	3	37	8	8	23
Attributes with Missing Values	67	7	0	16	2	2	9
Attributes with Missing Values > 60%	17	2	0	2	0	0	0
Observations	307511	1716428	27299925	1670214	10001358	13605401	3840312
Observations with Missing Values	298909	1676762	0	1670143	26184	2905	826036
Observations with Missing Values > 50%	0	0	0	0	0	0	0

Table 3: Balance of Target Variable

	Count	Percent (%)
0	282686	91.9
1	24825	8.1

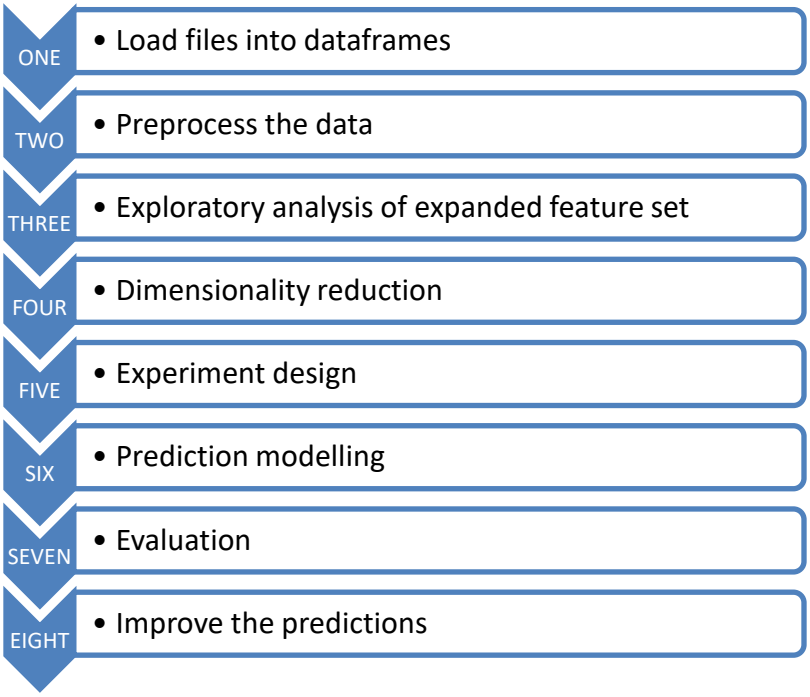
Figure 2: Distribution of Target Variable



The code to produce the Tables 1, 2 and 3 and Figure 2 is available here:
https://github.com/netvigate888/default/tree/master/Lit_Review_Data_Desc

Given the large size of the dataset and the complex mix of static and historical time series data, initial attention will focus on the main data file *application_train.csv* and the two files *bureau.csv* and *bureau_balance.csv*.

4. Approach



Step 1: Load Files into Dataframes

Each of the three files will be loaded into a Python pandas dataframe.

File Name	Dataframe Name
application_train.csv	app
bureau.csv	bureau
bureau_balance.csv	bureau_balance

Step 2: Preprocess the Data

- Examine the attributes of each dataframe:
  - Ensure attributes have correct data type
  - Check five number summary of numeric attributes
  - Check levels and frequencies of categorical attributes
- Look for inconsistencies in the data
- Examine time series data in *bureau_balance*
  - Define methodology to classify each time series as 'good credit', 'neutral credit', or 'bad credit'
  - Group the now labelled time series by applicant ID ('SK_ID_CURR')
  - Aggregate the grouped and labelled time series data at the applicant ID level. Aggregation methodology to be determined
  - Merge with *app* on applicant ID
- Examine data in *bureau*
  - Group data by applicant ID
  - Aggregate the data at the applicant ID level. Aggregation methodology to be determined
  - Merge with *app* on applicant ID
- Examine missing values by attribute and observation
  - Remove attributes and observations with too many missing values
    - Potentially remove attributes with greater than 70% missing values
    - Observations may not be removed as there are no observations with greater than 50% missing values
  - Impute remaining missing values
- Check for outliers
  - Where possible, understand the source/reason for outlying data

Step 3: Exploratory Analysis of the Expanded Feature Set

- Check balance of target variable
- Apply a low variance filter
- Univariant/bivariant/multivariant analysis
- Check distributions of attributes
- Subset data – look for insights by sub-groups

Step 4: Dimensionality Reduction

- Investigate possible dimensionality reduction through PCA

Step 5: Experiment Design

- Normalize data where required
- Encode categorical attributes
- Investigate over & under sampling techniques to create a new dataframe named *app_bal* from *app* with a more evenly balanced target variable
- Randomly split the dataframe *app_bal* into *app_train* with 70% of the data and *app_test* with the remaining 30% of data

Step 6: Prediction Modelling

- The task of predicating the target variable (0 = no default, 1 = default) is a binary classification problem.
- Investigate well established algorithms for binary classification such as:
    - Logistic Regression (LR)
        - LR will be used on the full feature set before dimensionality reduction (if used) to establish baseline performance metrics.
    - Decision Trees
        - Simple C4.5
        - Ensemble trees such as Random Forests
    - Support Vector Machines

Step 7: Evaluation

- Initial evaluation will use 10-fold cross validation on the training set only. Final evaluation will use the untouched test set.
- Evaluation metrics will include:
    - Confusion matrices
    - Sensitivity/Recall (True Positive Rate), Specificity (True Negative Rate), Accuracy
    - Area under the Receiver Operating Characteristic curve (ROC AUC)

Step 8: Improve the Predictions

- Investigate possible ways to improvement the prediction results
    - Model parameter optimization
    - New attribute engineering from the original attributes

References

[1] Turkson, R.E., Baagyere, E.Y. and Wenya, G.E., 2016, September. A machine learning approach for predicting bank credit worthiness. In *Artificial Intelligence and Pattern Recognition (AIPR), International Conference on* (pp. 1-7). IEEE.

[2] Liu, M.A., 2018. A Comparison of Machine Learning Algorithms for Prediction of Past Due Service in Commercial Credit. Grey Literature for Ph.D. Candidates, Kennesaw State University.

[3] Yu, X., 2017. Machine learning application in online lending risk prediction. *arXiv preprint arXiv:1707.04831*.

[4] Islam, S.R., Eberle, W. and Ghafoor, S.K., 2018. Credit Default Mining Using Combined Machine Learning and Heuristic Approach. *arXiv preprint arXiv:1807.01176*.

[5] Xiong, T., Wang, S., Mayers, A. and Monga, E., 2013. Personal bankruptcy prediction by mining credit card data. *Expert systems with applications*, *40*(2), pp.665-676..