SMU
SINGAPORE MANAGEMENT UNIVERSITY

ACCT337 Statistical Programming
Group Report

Credit Risk Modelling

Prepared for: Professor Benjamin Lee

Prepared by:
Section G1 Group 8

Daniel Joel Stoffel
Janessa Soh Jia Yi
Lu Kaiqi
Neo Jun Hao Jeffrey
Keith Poon Jun Kang

Date: 17 November 2019

Overview of the R application

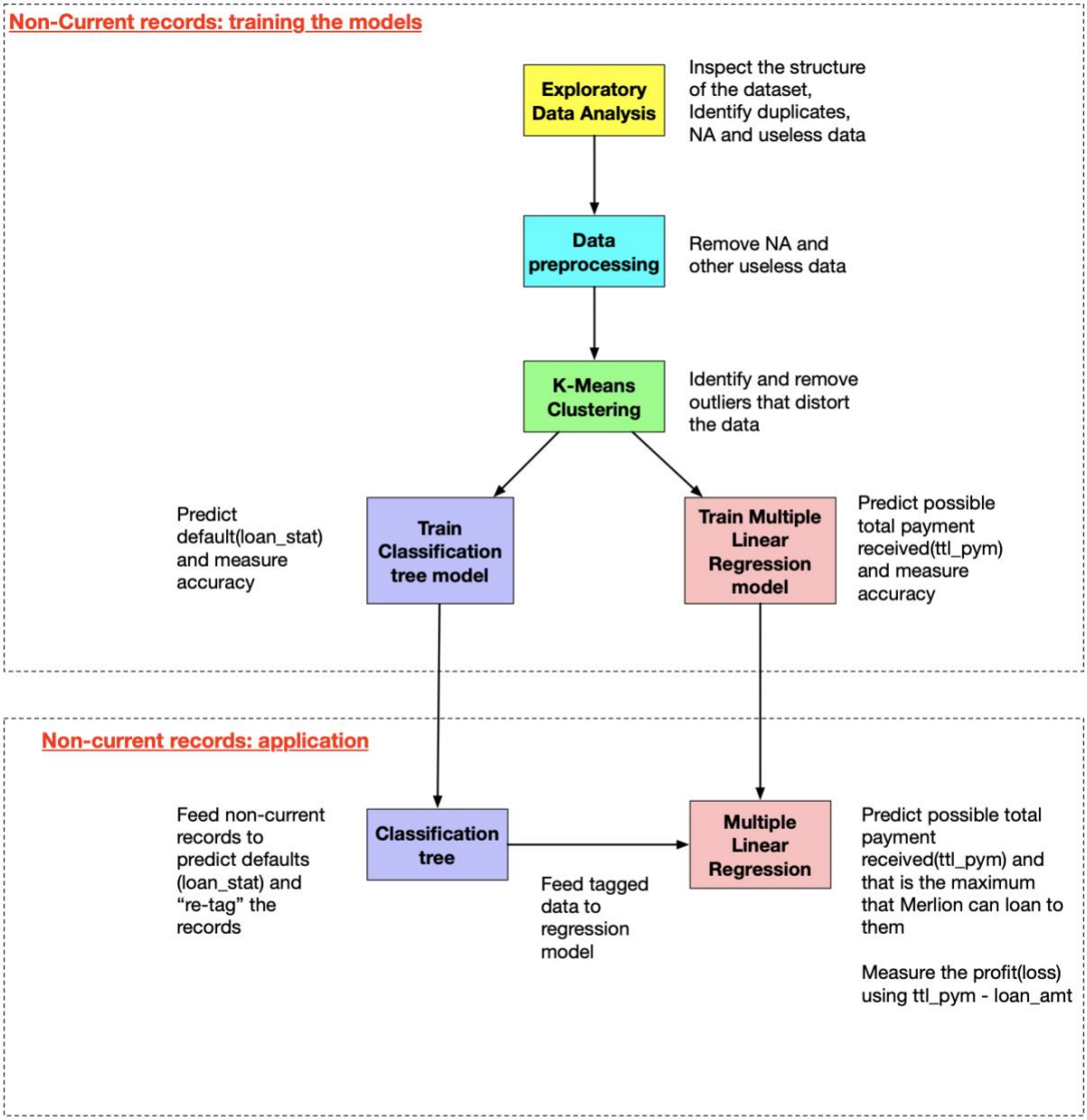

Figure 1: Project Flowchart

From Figure 1, We conducted an analysis on the data set through performing a series of tests. We aim to predict if a person would default, and therefore, the appropriate loan amount. We conducted unsupervised learning on the data set to find out if there were any anomalies that could possibly affect our study of how the various characteristics of the loan could impact the risk of a default or late payment. Later on, we performed supervised learning in the form of classification and regression.

1.0 Exploratory Data Analysis

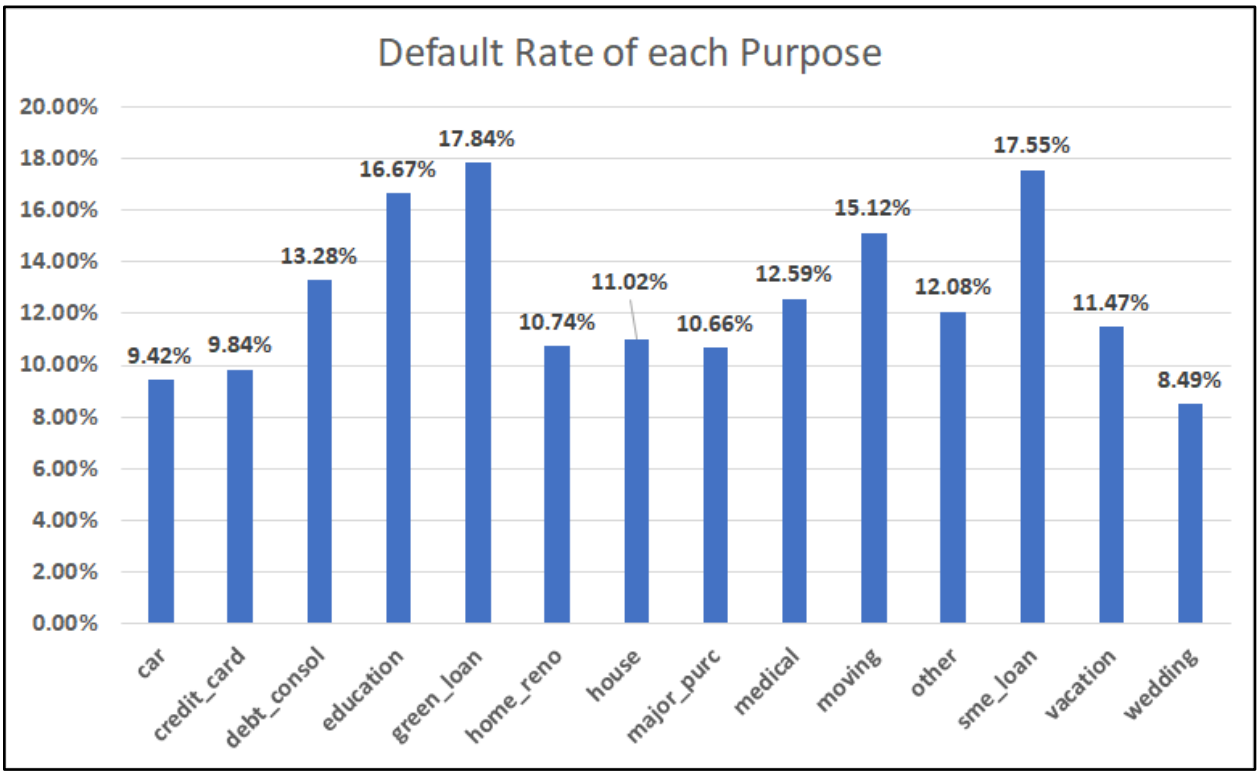

Figure 2: Default Rate of Purpose

We can see that some loan purposes are more prone to default than others.

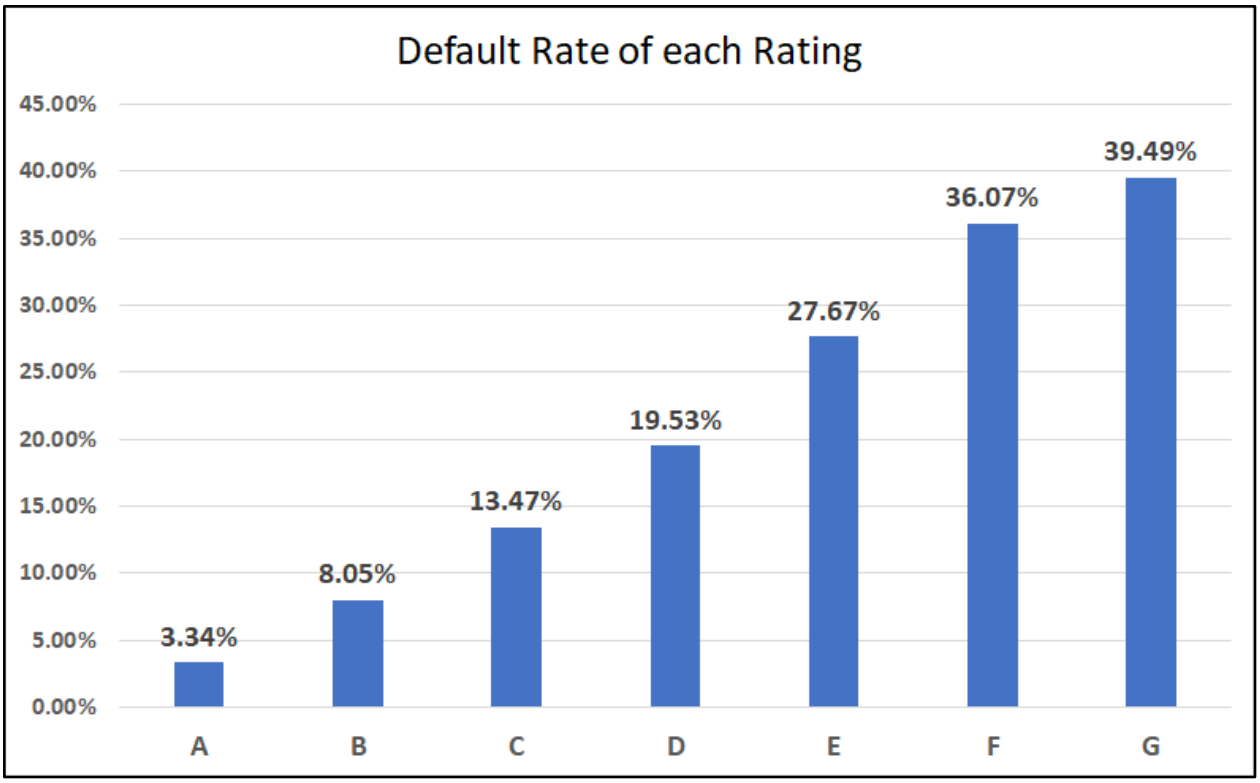

Figure 3: Default Rate of each Rating

We observe that applicants of poorer credit rating are more prone to default. This might be a potential indicator.

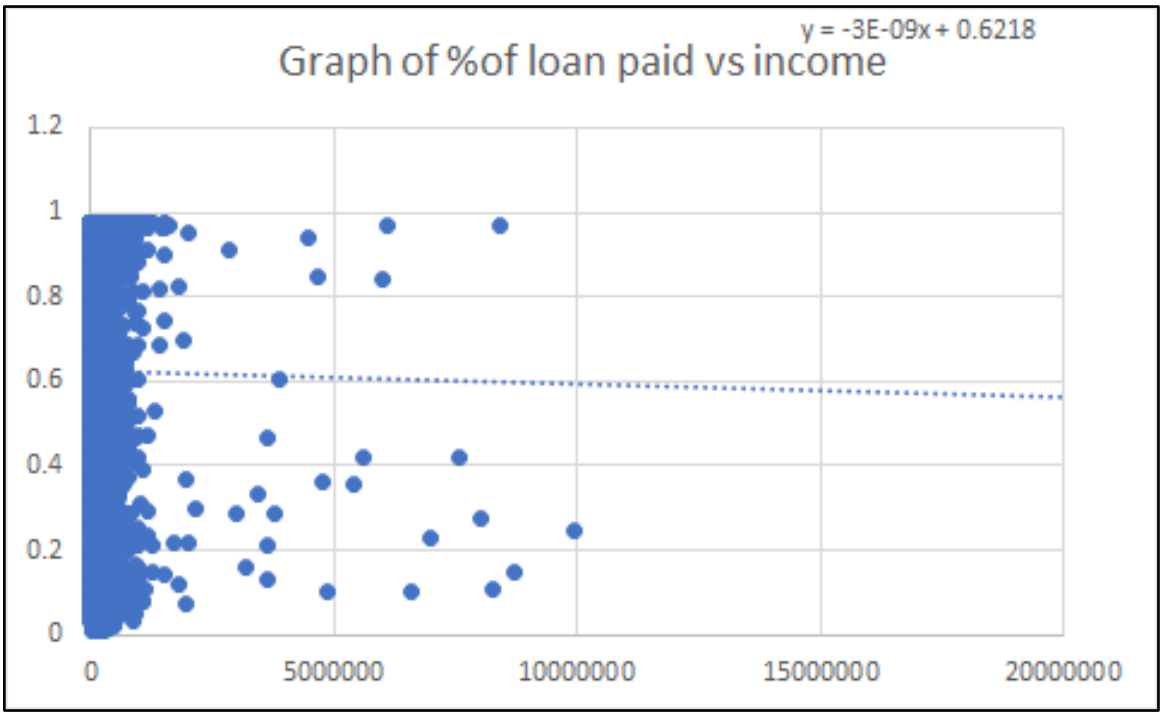

Figure 4: Graph of % of Loan Paid vs Income

We observe no clear trend between an applicant's income and how much of his loan will be repaid.

We can see that some variables function as potential indicators of default. We expected that a person with higher income would likely pay a larger proportion of his loan away. However, EDA has shown otherwise.

As we are not able to pinpoint strong indicators of potential default, we find it necessary to perform the following machine learning procedures.

2.0 Data Preprocessing

The dataset is loaded onto the R platform using the *read_csv* function in the *readr* library. Each of the variables are treated in the procedures described below.

2.1 Distinct

8 rows of data were assumed to be duplicates were dropped from the dataset and extracted as Duplicated_Entries.csv, as it is unlikely for a person with the same employment length, income, etc to occur at the together.

2.2 No change

Variables that are already in numeric form will not be changed, as numeric form is the ideal form for clustering, regression, and classification.

Such variables include *accts_pastd, bankr_rec, inc_ann, int_rate, loan_amt, mort_ac, ttl_int_rec, ttl_pr_rec and ttl_pym.*

2.3 Drop

Variables not useful in analysis would skew the results if left in the model. Examples include redundant or irrelevant variables.

We dropped *fund_mt*. This column shows the month and year of loan application. The year of loan is not useful for a predictive model as a new application will not carry the same year again. We have also dropped month in our models as we aim to create a credit assessment model. It is not intuitive for a person's credit worthiness to be affected by time. We might use this information when we expand our model to include seasonal analysis.

We dropped *inc_ann_jt* as the entire column is filled *NA* variables. This is because none of the loans are joint loans, and as such, joint income is irrelevant. It is also for this reason we have decided to drop *app_typ*.

2.4 Integer encode

Some of the variables are categorical. Where there is no need to give meaning to the hierarchy of each category item, we can integer encode the categories. Giving each category a numeric identity allows the clustering algorithm to find the Euclidean distance between observations.

We have carried out integer encoding on *home_own* and *purp,* by using the *mutate_if* function in the *dplyr* package.

We note that integer encoding is only carried out for clustering purposes. Our predictive models can accept factorial information, and will appropriately treat them.

2.5 Ranked encode

For categorical variables that exhibit ordinal qualities, it is not sufficient to integer encode them. Naively implementing integer encoding on ordinal variables might cause the hierarchical information within the data to be lost, as a category of larger numeric significance might be assigned a smaller group number.

Ranked encoding is an approach we have designed, where the numeric group name of each ordinal variable is stored in a dictionary. We then use the *mutate* function to append the said names to the dataset by querying the created dictionaries.

Such variables include *rating*, *plan,* and *emp_l.* Ratings contain ordinal information, where rating A is of a higher credit worthiness than rating E, for example. In the same way, a 3-year plan is of a shorter duration than a 5-year plan, and 10 years of employment is objectively longer than 3.

2.6 Overwrite

As charge-offs are essentially the same as defaults, we have standardized and renamed charge-offs as defaults.

3.0 K-means Clustering

After preprocessing, we further remove all the columns we would not have at the time of the application, namely: *loan_stat, ttl_pym, ttl_int_rec & ttl_pr_rec* for K-Means Clustering. Intentionally limiting the variables to information available at application would allow us to use this model to predict which group a new applicant belongs to.

We conducted clustering using 8 clusters as suggested by the Elbow Plot, which then generated the cluster plot below.

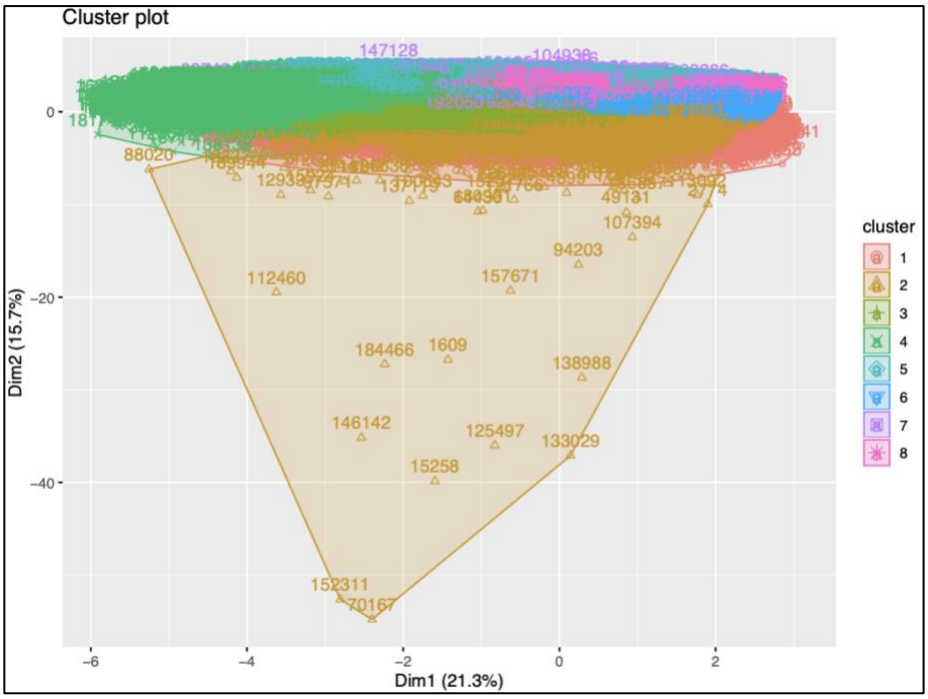

Figure 5: Cluster plot with 8 clusters

Minmax scores are calculated and by observation, most of the minmax scores are less than 0.2. Hence, We removed observations with a minmax score greater than 0.2 because. There are 13 of such records.

We have found that 42.3% of cluster 4 members have defaulted. This is more than twice as much as the average ratio of defaults.

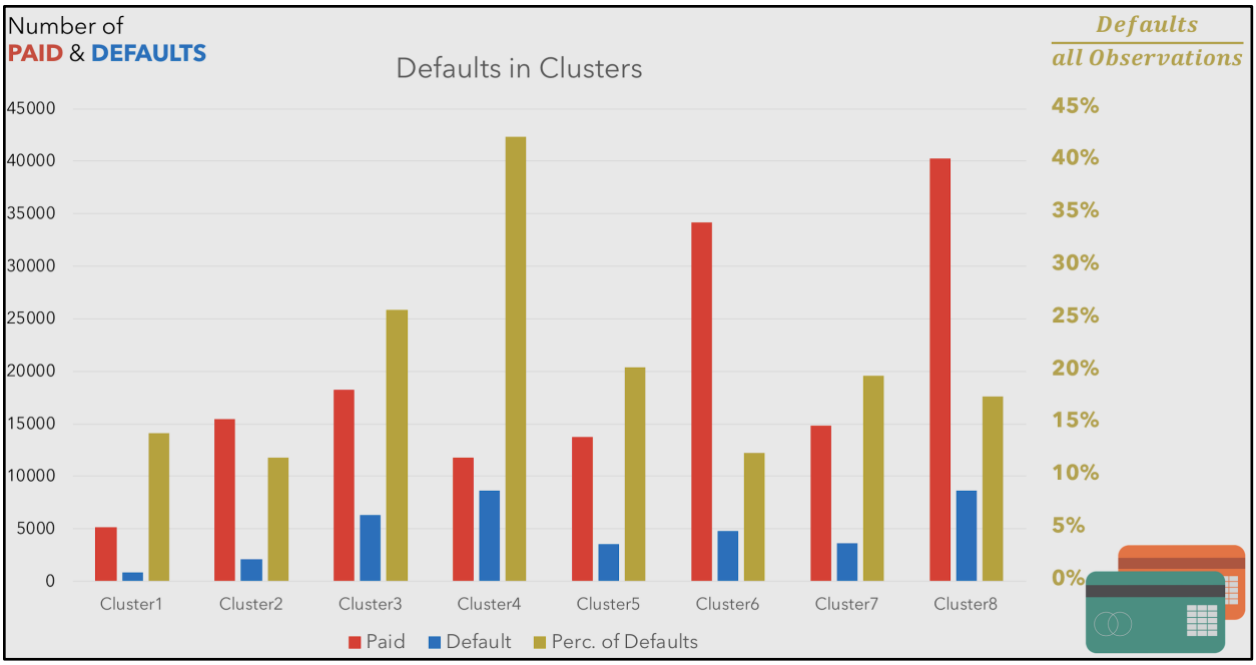

Figure 6: Defaults in Clusters

We also investigated if local outliers of each cluster are more likely to default.

We found that cluster 2, (1.78 times more defaults in the 20% with the highest min-max-score than in the 20% with the lowest min-max-score) had the highest influence of a high min-max-score, followed by cluster 7 (1.56 times more defaults in the top 20%) and 6 (1.38 times more defaults in the top 20%). This trend is weaker in other clusters (1.17 times more defaults in the top 20%).

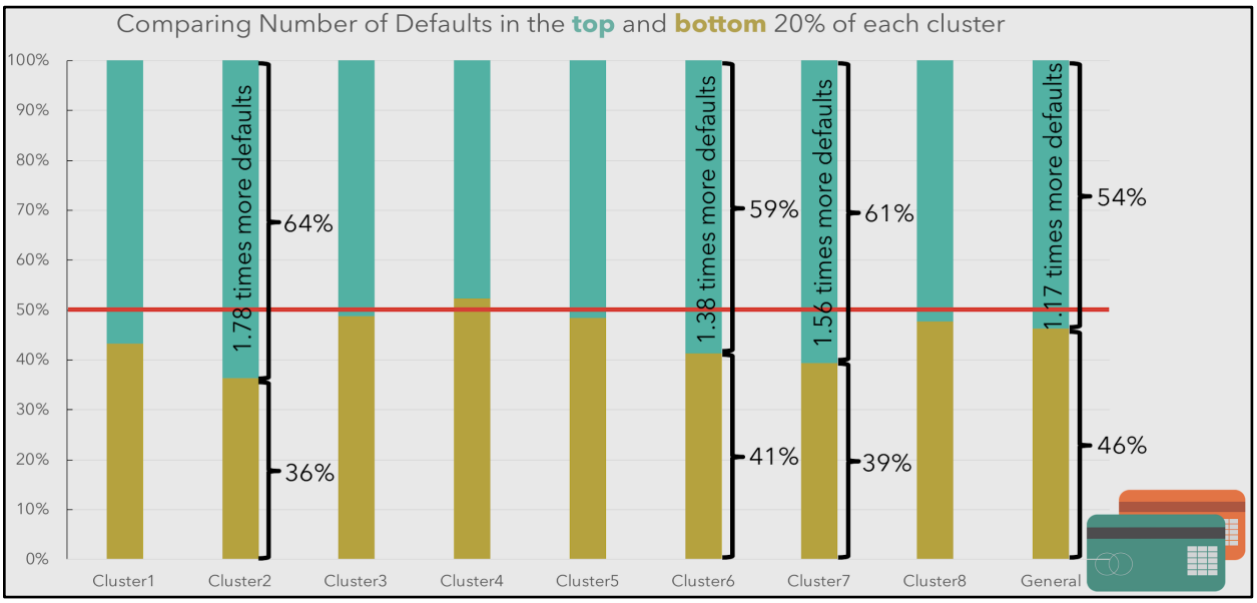

Figure 7: Defaults in top/bottom 20% of each cluster

4.0 Classification model

We assume that a person's background and current financial situation can be used to predict his credit-worthiness. As such, we fit a classification model on the data, to predict if the observations would default.

The model aims to determine if an applicant is likely to default, at the specified loan amount and interest rate.

We have implemented a recursive partitioning decision tree. We have attempted other classification models, and the reasons for not choosing those are outlined in Appendix A.

4.1 Recursive partitioning decision tree

This is the approach we have found to strike a balance between predictive power and ease of implementation. We recognize that the imbalanced nature of the data would cause biases.

To address this, overweighting was implemented. The *weight* parameter in the *rpart* function allows us to define the weight of each observation. By assigning the defaulters a weight equal to the ratio of number of non-defaulters to defaulters in the training set, we can artificially balance the data. Shown below is the confusion matrix.

Confusion matrix	Actual Default	Actual Paid
Predicted Default	8135	19120
Predicted Paid	3338	27027

Accuracy : 0.6437
95% CI : (0.6397, 0.6476)
No Information Rate : 0.8009
P-Value [Acc > NIR] : 1

Kappa : 0.207

Mcnemar's Test P-Value: <2e-16

Sensitivity : 0.6494
Specificity : 0.6422
Pos Pred Value : 0.3110
Neg Pred Value : 0.8805
Prevalence : 0.1991
Detection Rate : 0.1293
Detection Prevalence : 0.4158
Balanced Accuracy : 0.6458

'Positive' Class : Default

Figure 8: Confusion Matrix & Statistical Values

The model can identify about 2/3 of the defaulters. However, of all the predicted defaults, about 2/3 of them are wrongly classified. Attempting to increase the accuracy

of the model by making it less stringent would cause more defaulters to not be identified.

4.2 Management Section

Average loss per default is $7964.50 while average profit per settled loan is $1566.36. The following confusion matrix was generated from a less stringent version of our model.

Confusion matrix (low stringency)	Actual Default	Actual Paid
Predicted Default	7004	15890
Predicted Paid	4469	30257

Figure 9: Confusion Matrix

Profit generated based on the models:

$$final\ model: \ 3338(-7964.50) + 27027(1566.36) = 15,748,510.72$$

$$less\ stringent\ model: 4469(-7964.50) + 30257(1566.36) = 11,800,004.02$$

$$current\ procedure: \ (7004+4469)(-7964.50) + (15890+30257)(1566.36) = -19'093'893.58$$

Figure 10: Profit generated

We decided to increase the stringency of the model in order to identify more potential defaulters, as the loss on default is larger than gain per loan. It is seen that both the low & high stringency model lead to substantial improvements in the profitability of the fund.

5.0 Multiple Linear Regression

The purpose of multiple regression is to predict the amount of loan that is expected to be paid. We will not offer more than this amount on loan.

1. 14 outlying records identified from K-means clustering are removed because they can potentially distort the regression model, causing inaccurate prediction.

2. The *df_clean* data frame is further split into training set(70% of records) and test set(30% of record).
3. We regress *ttl_pym* as the Y variable against everything else. The coefficients are as shown.

```
> round(loan.reg$coefficients,2)
   (Intercept)     accts_pastd       bankr_rec    home_ownNONE   home_ownOTHER     home_ownOWN    home_ownRENT
     -10066.31           16.04         -166.43         2084.23        -1118.56         -172.39          -82.00
        inc_ann        int_rate        loan_amt         mort_ac     plan5 years purpcredit_card purpdebt_consol
           0.00          -24.61            1.03           10.44          335.02          529.78          455.07
 purpeducation  purpgreen_loan   purphome_reno       purphouse  purpmajor_purc     purpmedical      purpmoving
        -265.24          639.91           89.72         -399.96        -242.69           43.47          317.85
     purpother   purpsme_loan    purpvacation     purpwedding         ratingB         ratingC         ratingD
          56.37           19.55           26.83          179.02          777.07         1431.34         1916.45
        ratingE         ratingF         ratingG          emp_l  loan_statPaid
        2407.01         2485.70         2041.21           6.67        10003.09
```

Figure 11: Coefficients of regression model

4. We have left *loan_stat* in the model as we intend to merge this regression model with our predictions of the default classification model.
5. The model has an Adjusted R2 of 0.8741 which means the variables explain 87.41% of the variances in *ttl_pym*.
6. Accuracy of the model as shown

```
> accuracy(test.pred,test.mlm$ttl_pym)
                   ME      RMSE       MAE        MPE      MAPE
Test set -10.02402 3664.857 2191.749 -4.028867 29.21569
```

Figure 12: Statistical values of regression model

7. To improve the prediction, Forward Selection, Backward Elimination and Stepwise regression are implemented. However, the 3 methods generate the same output with the same accuracy as before.
8. The model gives a MAPE of 29.22% which means each of the prediction has 29.22% deviation on average.

5.1 Management Section

Given that the regression models' accuracies stagnate at a MAPE of 29.22%, it suggests that the current loan recording system needs more features.

Current loan profiling system may be insufficient: the fact that the Forward, Backward and Stepwise models do not remove any feature shows that all these features contribute to the prediction of *ttl_pym*. However, the accuracy of the results still did not improved. This suggests that the current application profiling system does not allow us or the management to build a reliable and accurate regression model. The

system should seek to record more attributes of the loan profiling system using the 5Cs of credit framework-Character, Capacity, Capital, Conditions and Collateral. In terms of capacity, the borrower's ability to pay the loan is already reflected in the credit rating. Some improvements include; in terms of character, an analysis of the borrowers' trustworthiness and credibility can be assessed, in terms of collateral, we can assess the income of the guarantors and in terms of capital, we can assess the amount of capital placed into the borrower's business venture. These additional variables will narrow down more relevant features that affect the loan amount and help us to improve the accuracy of our regression model.

We observe larger regression coefficients for increasing rating classes. This implies that poorly rated individuals are expected to return a larger amount of money. This is counter-intuitive to our understanding of credit scoring, and we recommend that the management look into this phenomenon.

6.0 Final Model

With the models we have developed, we can predict the amount of money to lend to each applicant. We run the applications through the classification model in order to first predict if this application is likely to default. We tag each application with its predicted loan stat, and run it through our regression model. The resultant output is the prediction of how much the applicant is expected to pay. This is the maximum we would be willing to offer on loan. The results of this model are explained in Appendix B. This combined model will be applied to the "Current" records to predict default rate, and the amount expected to receive as well.

7.0 Further Recommendations for the management

We have analyzed the dataset and found the systemic issues outlined below. Appendix D elaborates and provides additional insight.

1. Labeling of loan status: There are numerous clients that are wrongly classified, such as having paid more than loan amount but classified as "Chargeoff", while another client is considered "paid" when only 30% of the loan is repaid.
2. ttl_pym: About 10.59% of the dataset have a wrongly calculated total payment. This is elaborated upon in Appendix C.
3. Inc_ann: 239 were observed to have no income despite being employed. This could be a sign of data entry error or proper checks were not done.

4. Duplicates: the duplicated observations might be due to input error, where an administrative personnel entered the same application twice. Management is advised to review their operational processes.

Our team came to the conclusion that despite having double-verified system for loan status, such a high percentage of error, inconsistencies and dataset discrepancies suggests that there could be fraud within the company. Similarly, with so many data entry errors, this suggests weak internal control within the company to spot these errors.

Appendix A

Random forest classification

The random forest algorithm guarantees a fitted model that explains the training dataset. However, we have found that the precise nature of the algorithm makes it prone to overfitting. We have experienced a 30% drop in accuracy when validating the model on the test set. Furthermore, it takes a long time to train a random forest model as the algorithm generates multiple decision trees to assess fit. As such, we have decided not to use this algorithm.

Data balancing algorithm

The provided dataset is imbalanced, with the occurrence of non-default observations 4 times more likely than defaults. There is a need to balance this data. Without balancing the data, most algorithms will attempt to maximize accuracy, classifying every occurrence as non-default. Doing so will guarantee a high accuracy score, but the machine would be useless for prediction.

SMOTE and ROSE packages allow for synthesis of artificial data points to balance the data. Both algorithms involve interpolating observations to generate new points. However, we have concluded that this approach is also not suitable for this project. Given that the points are generated from interpolation and extrapolation of current data, it synthesizes floating point values, even in columns that contain integer values. For example, the interpolation may generate data points such as 2.3 mortgages, or 3.7 accounts past due date. This may cause biases in the training of the model.

Furthermore, we did not observe significant improvements in our accuracy matrix. As such, we have not elected to implement this approach.

Neural Network

We tried to build a neural network for the classificator (with the “neuralnet” package) but did not succeed to get some useful results. Additionally, we also assume that our dataset might be too small to successfully use a neural network with that many variables. Thus, we decided not to focus anymore on that approach.

Appendix B

Final prediction model in “Final machine.R”

As mentioned in the report, our final model is a combination model where it first predicts if the application is likely to default, then this prediction is fed into the regression model. The model then predicts the amount that the applicant is likely to repay. This should be the maximum amount of money we are willing to extend on loan.

We applied this model to the **non-current records** to measure the profit(loss) generated. Using this combine model, the firm would incur a loss of $540,345,588. The firm would incur a loss of $65,350,439 using their current model. As such, we can see that the model is grossly inaccurate. The accuracy stats of the model is outlined below.

	ME	RMSE	MAE	MPE	MAPE
Test set	2473.083	7159.033	4972.691	6.132801	50.99533

We tried another approach, where we rely purely on the classification model, and not combine it with the regression model. That is to say, we loan the full amount applied for to an applicant who is predicted to repay it, and nothing to the one who is predicted to default.

In doing so, we incur a loss of $3,888,243. This is a significant improvement from both the combined model and the current approach taken by the firm. Theoretically, as the combined approach uses more information in decision making, it should lead to a more insightful prediction machine. Its lackluster performance might be due to inadequate variables.

We recommend the firm not to make use of the combined model for credit analysis until more variables are made available. Until then, the firm would see a significant improvement in expected loss using the classification model in deciding to loan money to an applicant.

Appendix C

Dataset Discrepancies

Purpose

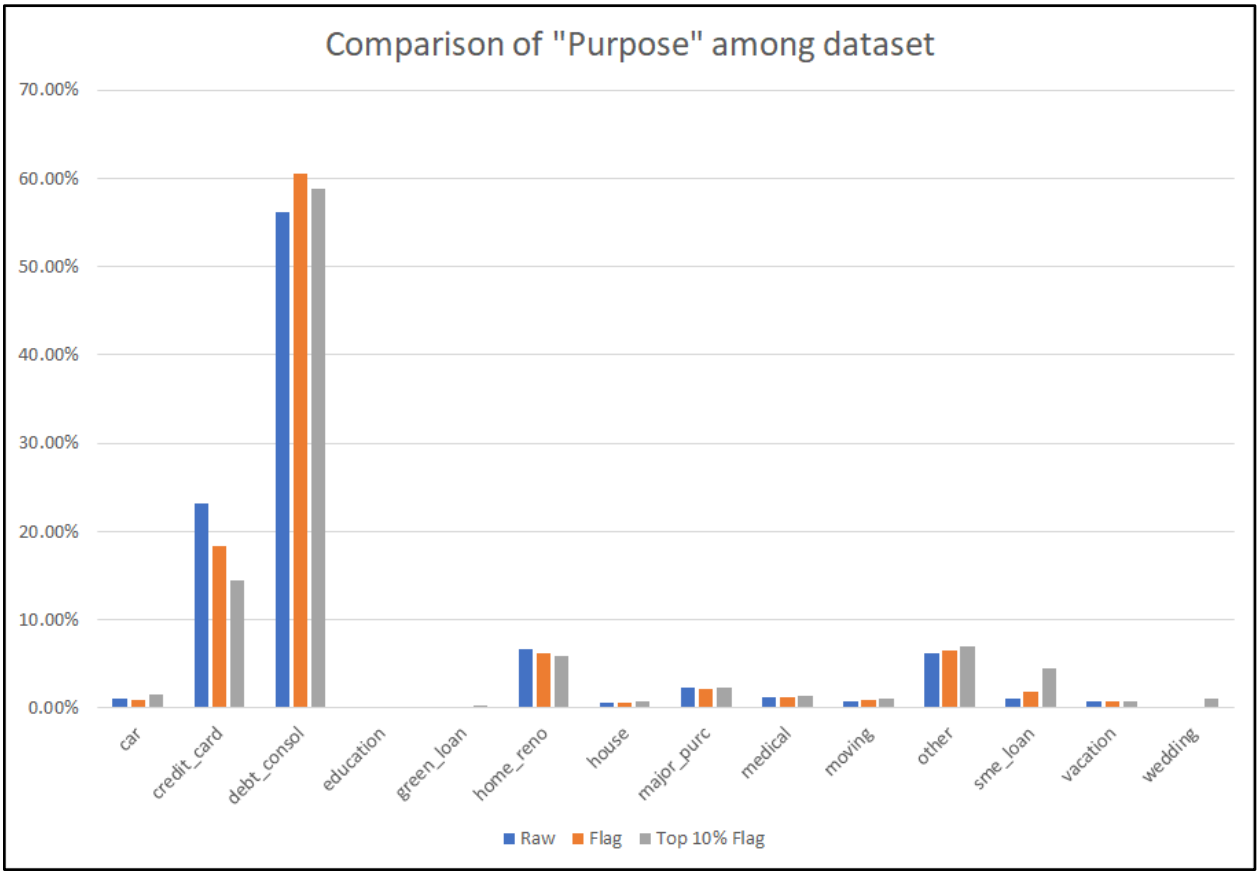

Figure 13: Comparison of Purpose among dataset

Similar distribution across all 3 datasets. Nothing out of the norm.

Loan Status

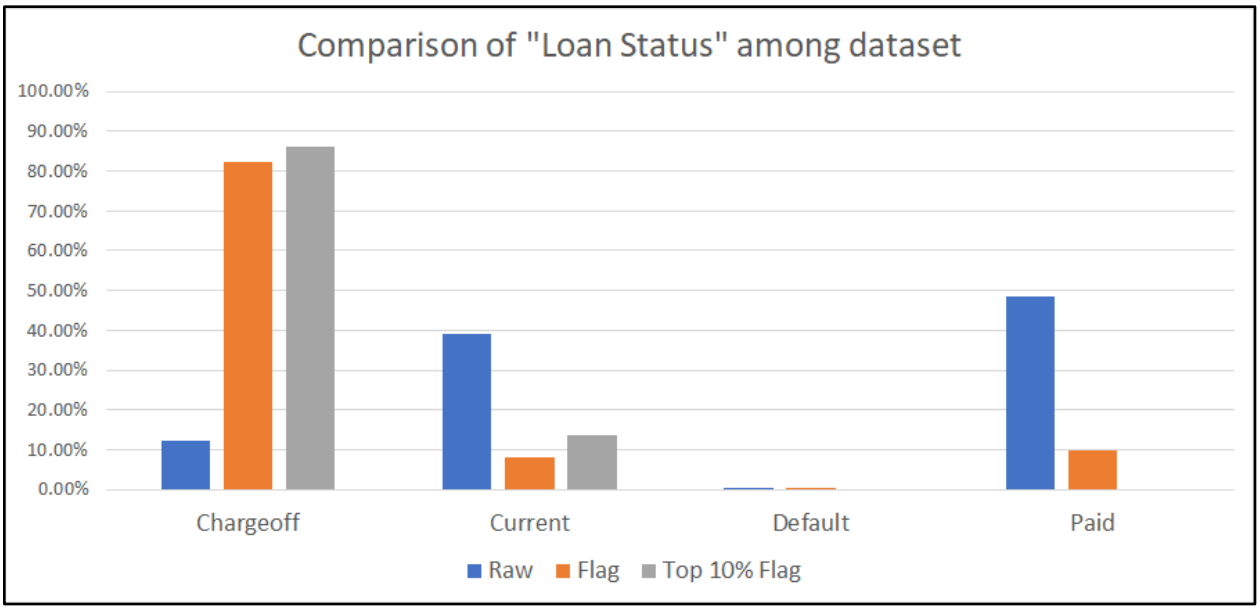

Figure 14: Comparison of Loan Status

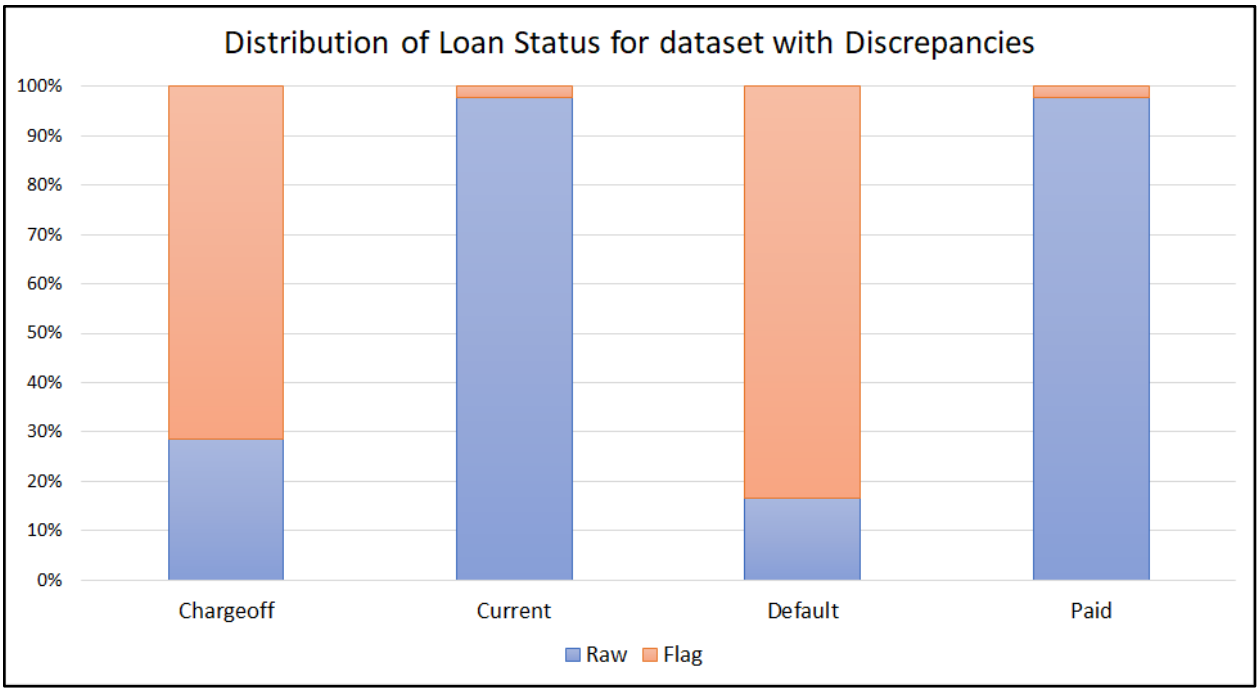

Figure 15: Distribution of Loan Status for Dataset with Discrepancies

Large percentage of chargeoff in discrepancies dataset compared to raw dataset. In addition, 71.5% of “chargeoff” and 83.3% of “Default” loan have discrepancies issues. Upon further analysis, we found that 4.7% (1,279) of customer with discrepancies and labeled as chargeoff have already paid back than the loan amount.

Loan Status Breakdown

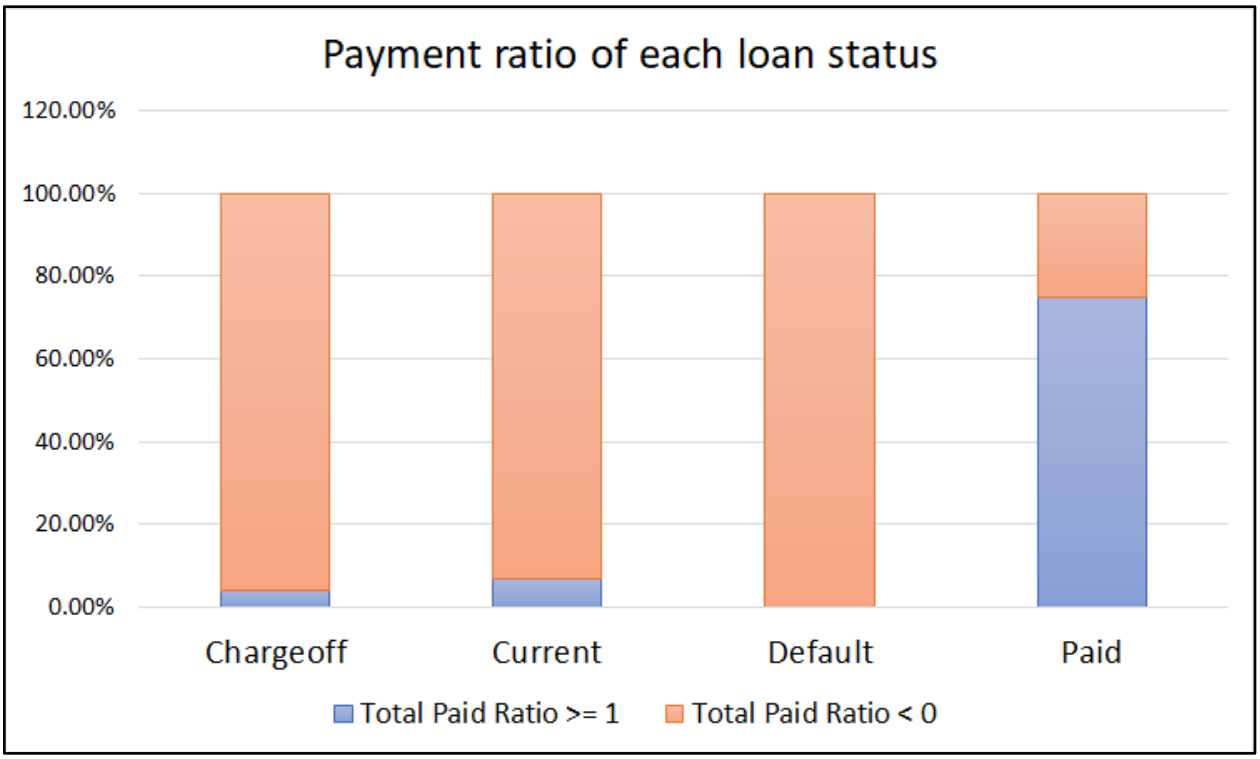

Figure 16: Payment ratio in respect to loan amount for each loan status

4.2% & 7.1% of chargeoff and current have already paid back more than the loan amount. 25.2% of paid loan are being classified as paid even when amount paid is less than the loan amount.

Appendix D

The codes for the generation of the files below can be found in the “Appendix D.R” file.

Documentation of each csv generated in “Appendix D.R”

CSV File Name	Purpose / Explanation / Documentation
Chargeoff_Discrepancy	List of dataset with loan status being “chargeoff” and the difference between “ttl_int_rec + ttl_pr_rec” and ttl_pym being more than 251.51.
Dataset_with_Loanstat_Anomalies	List of dataset with “Chargeoff”, “Current”, “Default” loan status of having paid more than 70% of the loan amount. In addition, the dataset also contains “Paid” loan status of having paid less than 30% of the loan amount. The cutoff “70%” and “30%” can be adjusted in the R file “Appendix D.R”, line 201, 202.
Discrepancy_Flag	List of flag out dataset with difference between “ttl_int_rec + ttl_pr_rec” and ttl_pym being more than 251.51.
Discrepancy_Loan_Stat	Distribution and amount in loan status of top 10% difference in “Discrepancy_Flag.csv”, “Discrepancy_flag.csv” and the original dataset without the duplicated entries. The values here are also used in figure 12 of the management report.
Discrepancy_Purp	Distribution and amount in Purpose of top 10% difference in “Discrepancy_Flag.csv”, “Discrepancy_flag.csv” and the original dataset without the duplicated entries. The values here are also used in figure 11 of the management report.
Duplicated_Entries	List of entries that are duplicated in the dataset.
Loan_Stat_Findings	Distribution of “Chargeoff”, “Current”, “Default”, “Paid” dataset. They are split into whether the customer had paid back more than or equal to their loan amount and vice versa. The values here are also used in figure 14 of the management report.