ISYE7406 Project - Classification Methods for Predicting and Understanding Factors of Housing Foreclosures in India

Hannah Pavlovich, Stijn Jorissen (group 121)

April 2024

1 Abstract

This report explores different methods to predict foreclosure on bank loans. This is done by fitting the available data on this subject to 7 model types: decision trees, random forests, boosted trees, logistic regression, K - nearest neighbors, neural networks and support vector machines. The optimal variables used to predict foreclosure are investigated to have less complex model building, and also to achieve better interpretability for the overall model. The final models are compared using their average accuracies and sensitivities using the respective variances of these values in order to determine which one would perform the best on the dataset. However, no definitive answer was found because there were a lower amount of computational resources available than were needed for this (more iterations on the models would need to be done). Therefore, based on the t-test, four top models, rather than a full ranking of models, were decided on: decision trees, partial data logistic regression, partial data SVM and full data SVM models. The SVM model with all data is performing statistically better than the other, but is found undesirable due to high training times and lots of data needed. Based on this, the decision trees and partial data logistic regression models are recommended because of high interpretability and low amount of data needed. The 5 predictors used for the partial data models are: Product, Paid interest, EMI due amount, EMI received amount and Current Interest Rate Minimum. These where found to have the most predictive value and thus it is also recommended to use these predictors to simplify the models.

2 Introduction

The NBFC Loan Transaction dataset contains information regarding foreclosures in India. When a buyer cannot repay their mortgage, the mortgage goes into foreclosure, giving the bank license to sell the home and recoup their loss.

The foreclosure process is difficult for the buyer, who is losing their home, and the bank, who must use their resources to auction the house.

The purpose of this project is to use the 'Foreclosure Prediction of Financial Dataset' to minimize the foreclosure rate. Classification Models are built to accurately predict foreclosure based on the whole dataset and selected variables. The project is also meant to understand what the factors are, and how to explain to the buyer how high their risk of foreclosure is. A model which sacrifices accuracy for interpretability would meet this challenge.

The main research questions therefore are:

- What does the bank need to be aware of to detect risk of the client going into foreclosure?
- How can we use as few variables as possible while maintaining an accurate model? The model should use as few variables as possible, because high dimensionality adds complexity for the bank employee needing to interpret the results. The model also becomes more fragile because this data will be needed at each moment for evaluation
- How can we build the most sensitive model to capture more loans in risk of foreclosure at the expense of selecting loans that may not be at risk?

To answer these questions the 'Foreclosure Prediction of Financial Dataset' was used [1]. To increase interpretability variable selection is performed to determine which variables are the most impactful. A multitude of classification models where fitted to determine the right trade-off between complexity and interpretability, the models fitted are:

- Decision-trees

- Random-Forest
- Boosted Trees
- Logistic Regression
- K-nearest neighbours
- Neural networks
- Support vector machines

Furthermore, principal component analysis is performed to see if this would increase the accuracy or sensitivity of the model. Generally, the model's quality is determined by its sensitivity because the impact of going to foreclosure outweighs the risk of a false positive. It is less trouble to do some additional research based on the model compared to not perceiving a loan as risky and then going to foreclosure.

3 Exploratory Data Analysis

The foreclosure dataset supplied by Mukesh Tiwari on kaggle is high dimensional [1]. It expresses information about loans in India and whether or not the loan was foreclosed. There are 53 types of variables in the dataset and 20012 data points in the dataset. The variable "foreclosure" is described by a 0 or a 1; 1 indicating foreclosure proceedings were held, and 0 indicating no foreclosure was executed. In the dataset, 1795 data points indicate foreclosure was executed, which is only about 9% of all the data. Due to the uneven nature of the response between foreclosure or not, precaution is needed when looking at the classification accuracy. These methods will be described in Section 4. As stated before the sensitivity is taken as the primary parameter to determine model quality.

The data consists of the following primary categories of variables:

1. Financial balance data: describing the amount of money that needs to be paid or that was overpaid up until the point the data was gathered. Most of this data is kept, only data indicating the full loan was paid for was removed because this would go against the goal of predicting the validity of the goal by giving an overly strong indication for the answer.

2. Interest data: this data consists of the amount of interest paid and the interest rates for the loan, all of this data was kept.

3. Data/time information: this is information on time since the last payment was made, but also when the loan was given out and the next due date of the loan. We chose to remove these variables because our predictions are based on the financial data rather than the specific times.

4. Identification numbers (ID's): internal numbers the bank used to identify the customer and the place the loan was given out. These where all taken out as predictor variables because they generally have no predictive use.

5. Loan location: the location the loan was given, this is taken out because of the model is shooting to be applicable generally in India and not dependent on location.

6. Response variable: did the loan go to foreclosure? A 1 indicates that it did while a 0 indicates it did not.

All of the 53 variables with a short description on each are in Appendix **??**. A correlation matrix on the remaining 37 variables was made (Figure 1). Due to the size of the matrix, it was impossible to have good descriptions per variable and thus they are numerated. The legend between the variable and the number is in Appendix **??**. The distributions of the variables are shown in Figure BB.

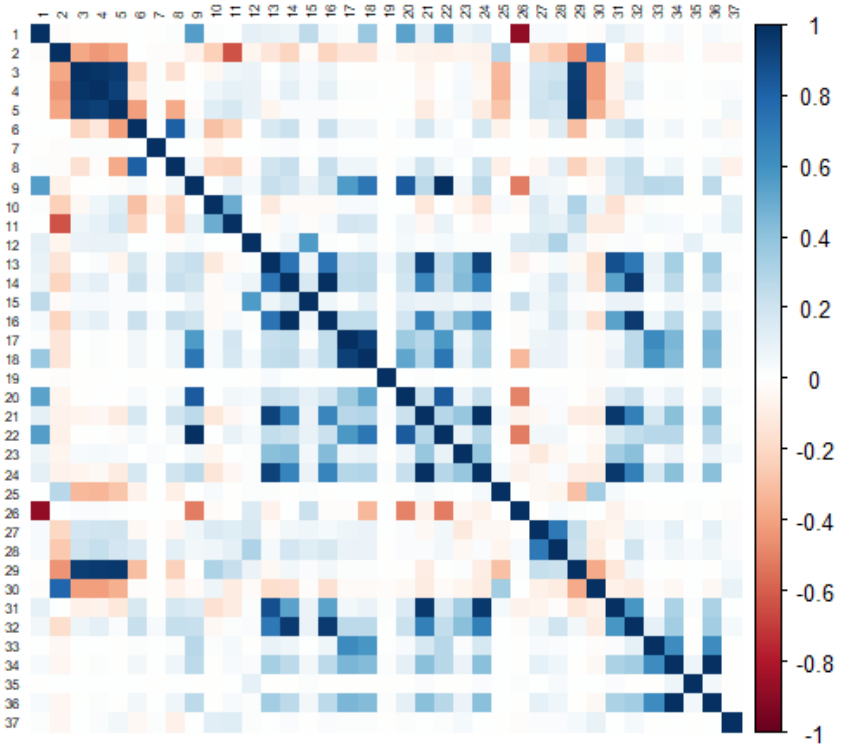


Figure 1: Correlation matrix for all investigated data

In Figure 1 you can see strong correlations between variables 3 to 5: these are interest rate data (maximum, minimum and changes). Also variables 13 to 24 have moderately to strong correlations. This is data describing the amount of equated monthly installment and the size of the loan. This makes sense: the amount of money due per month is usually positively correlated with the size of the loan. Also standing out in the matrix is the strong negative correlation between variables 1 and 26; these represent the balance excess of the loan and the receivable amount of the loan. The variables that indicates foreclosure is number 37, it does not have a strong relationship with any variable except for itself.

Principal component analysis was performed on the 36 predicting variables to see if many variables have variability in the same direction and thus the computational complexity of the model could be reduced. The results of this is shown in Figure 2. The first 5 principal components explain 60% of the variability, and 80% is explained by the first 10. The large drawback of this method is that it greatly reduces interpretability.

Histograms were made and studied for each variable, they are shown in Appendix B. The high amount of similar values is illustrated in these. A majority of the data is very one sided with only a few of outliers.

Based on iterations in the decision trees described in Section 5.1, five variables are chosen with which to build separate models. These variables are chosen because they where found to be the most impact full due to early branching on the decision tree model. These five variables are:

- Product
- Paid interest
- EMI due amount
- EMI received amount
- Current interest rate minimum

The data correlation with foreclosure is illustrated in boxplots 1.

In Figure 1 it is illustrated again that most of the data is distributed narrowly, for most of the boxplots the 50% is a single line. Only for the HL product this is not the case and a decent amount of the loan with this product went to foreclosure (28.4% of the loans with the HL product). The categories have the majority of the data as 0 for both loans going to foreclosure and loans not going to foreclosure except for the current interest rate min. Fore the current interest rate min variable the outer edges of the boxes range between 10 and 22 for loans that did not go to foreclosure and between 10 and 24 for loans that did.

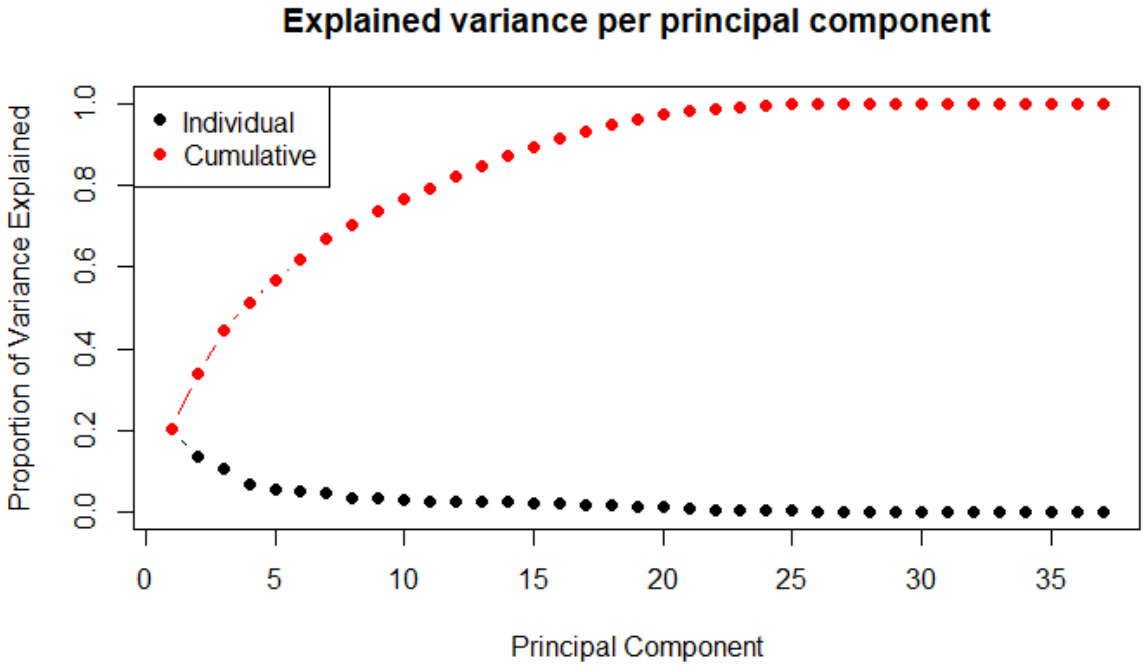


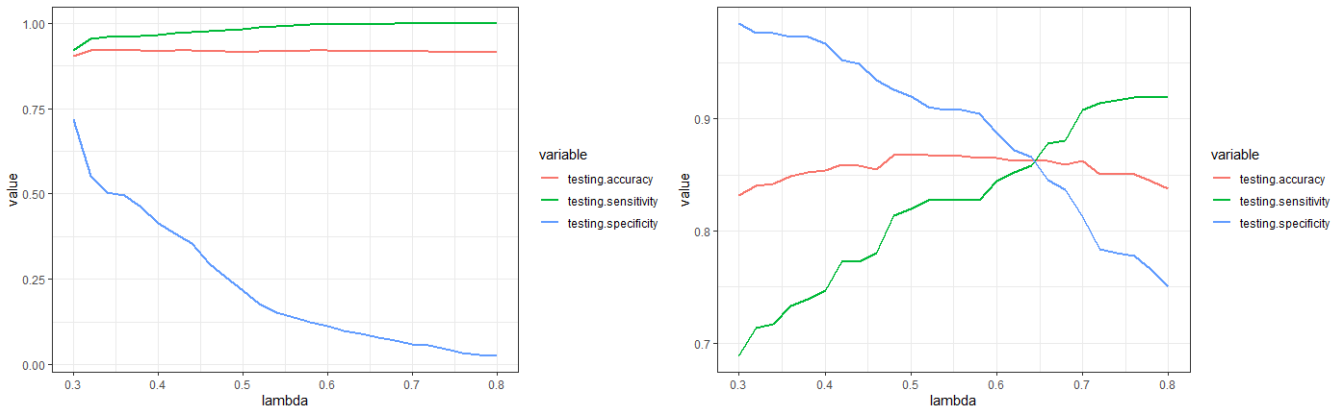
Figure 2: Principal component analysis summary

4 Balancing the Data

As described in EDA, of the 20012 data points in the model, only 1795 data points or 9% of all datapoints end in foreclosure. This unbalanced data results in models with near perfect accuracy. If all of the data was predicted as not going into foreclosure, the model would have an accuracy of 91% and sensitivity of 100%, which, by all other standards, would be a good result.

A simple logistic regression on the entire dataset with the intention of tuning for optimal lambda illustrates this point. The plot has three lines representing accuracy, sensitivity, and specificity in relation to lambda.

Using a full dataset, as lambda increases, the accuracy and sensitivity remain high, with sensitivity at near perfect with the high lambda. Conversely, the specificity drops, as the new model is predicting almost complete non-foreclosure, as seen in Figure 3a.



(a) Testing metrics vs λ using full dataset

(b) Testing metrics vs λ using balanced dataset

Figure 3: Testing Metrics of Full vs Balanced Datasets

To lessen the impact of the uneven data on our final model, we looked at balancing the data through the following procedure:

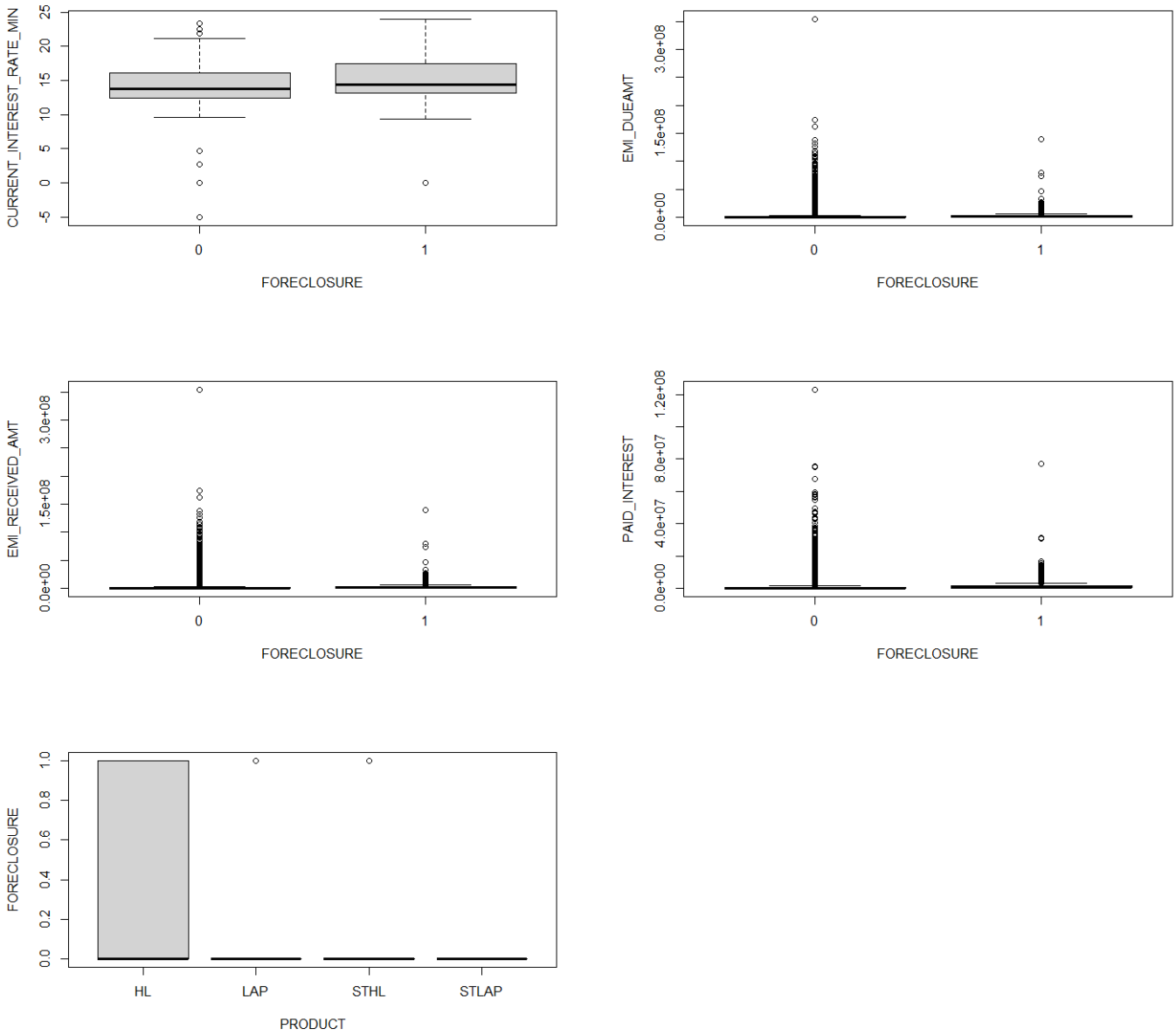


Table 1: Boxplots comparing the variance in foreclosure to five predictors

1. Split the full dataset into two sets
   (a) Set 0: all points that did not go to foreclosure (18217 points)
   (b) Set 1: all points that did go to foreclosure (1795 points)
2. Randomly choose 1795 data points in Set 0
3. Combine Set 0 and Set 1 to make (1795, x) dataframe DF, where x is the number of variables
4. Train and test model on new dataframe DF

Once the data is balanced, the problem remains that not all data points are being used. Thus, for each trained model, a Monte Carlo simulation is run with a different set of randomly selected variables from Set 0.

The result of balancing the data is shown in Figure 3b, showing accuracy, sensitivity, and specificity against λ. With the balanced dataset, sensitivity and specificity are inversely related, and they meet at the same point as accuracy. The nexus of all three line is the optimal lambda for a balanced model with specificity and sensitivity.

This method is applied to all models. As a result, the variance across all models is increased when compared with the unbalanced data.

5 Methods

In this section the various classification methods used to predict and describe the data are explored.

5.1 Decision trees

The Decision Tree model is a good choice for classification when interpretability is a priority. The full data set with all variables is used in building this model, as the tuning of the parameters reduces the complexity.

Tuning the Decision Tree is the first step of the model, as the decision tree is highly dependent on the node size and maximum depth. Tuning was done on both training and testing models to avoid problems with overfitting.

Node Size was tuned for values 2:10 by running a Monte Carlo Simulation with the balanced data. The results are shown in Figure 5 and Table 2.

node.size	training.accuracy	training.sensitivity	testing.accuracy	testing.sensitivity
2	0.9046	0.9428	0.8638	0.8988
3	0.9021	0.9399	0.8626	0.9007
4	0.9016	0.9396	0.8641	0.8999
5	0.9021	0.9357	0.8628	0.8955
6	0.8988	0.9354	0.8626	0.8994
7	0.8979	0.9369	0.8638	0.9029
8	0.8997	0.9365	0.8619	0.8988
9	0.8955	0.9356	0.8617	0.9006
10	0.8979	0.9335	0.8607	0.8966

Table 2: Accuracy and Sensitivity by Node Size

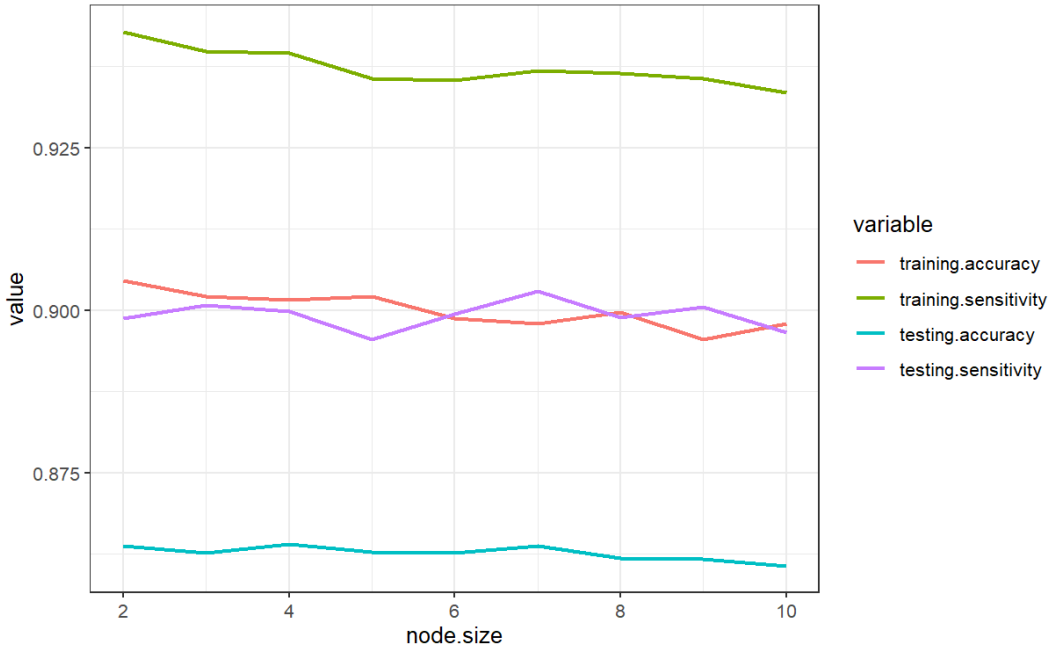


Figure 4: Tuning Metrics vs Node Size

The results show the optimal node size for training sensitivity is 7, followed by 3. The Decision Tree shows very high sensitivity, which may mean that this model is prone to overfitting. For this reason, nodesize = 7 is used in the final model, as the training sensitivity is lower than at nodesize=3.

Max Depth is also tuned, as a model with high maximum depth results in an overfit model, while a too low maximum depth may not be accurate enough. The max depth was tuned for values 4:30 by running a Monte Carlo Simulation with the balanced data. The results are shown with truncated results Table 5 and full results Table 3.

maxdepth	training.accuracy	training.sensitivity	testing.accuracy	testing.sensitivity
5	0.8923	0.9364	0.8627	0.9057
10	0.8982	0.9380	0.8641	0.9037
11	0.9004	0.9383	0.8618	0.9001
12	0.8982	0.9385	0.8621	0.8979
13	0.8989	0.9410	0.8636	0.9040
16	0.8990	0.9422	0.8642	0.9074
20	0.9017	0.9395	0.8645	0.9016
21	0.8998	0.9384	0.8642	0.9049
22	0.8986	0.9393	0.8630	0.9032
24	0.8990	0.9401	0.8619	0.9005
25	0.8974	0.9378	0.8621	0.9026
28	0.9002	0.9390	0.8637	0.9005
29	0.8978	0.9375	0.8628	0.9010

Table 3: Accuracy and Sensitivity by Maximum Depth

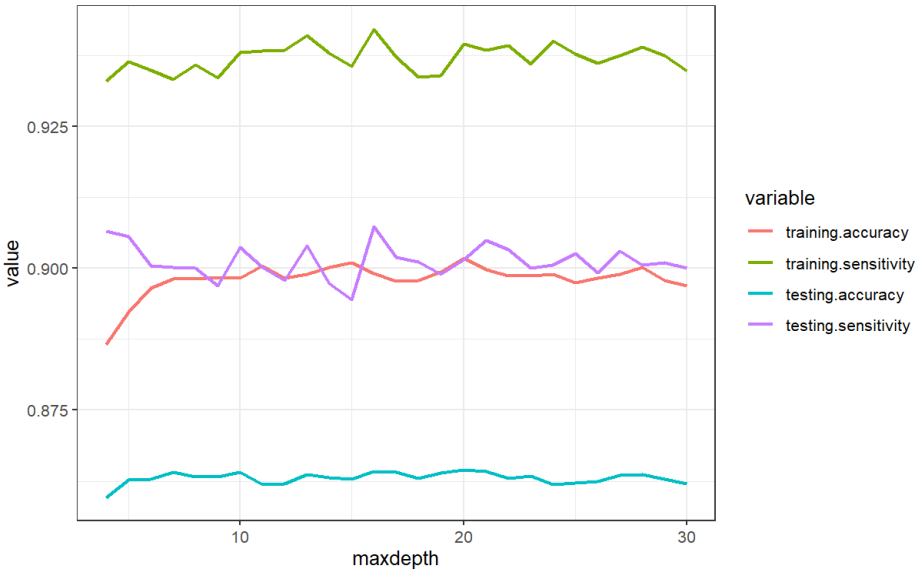


Figure 5: Accuracy and Sensitivity vs Maximum Depth

The results show the optimal maximum for training sensitivity is 16, followed by 4, 5, and 21. The Decision Tree shows very high sensitivity on the training data, which may mean that this model is prone to overfitting, so caution is taken in choosing the optimal maximum depth. For this reason and because of the focus on interpretability , we choose maxdepth = 4.

The final Decision Tree model is built with the aforementioned parameters and running them and run under a Monte Carlo simulation with balanced data. Figure 6 displays one tree from the 1000 produced under the simulation, and the results will be explored in Section 6.

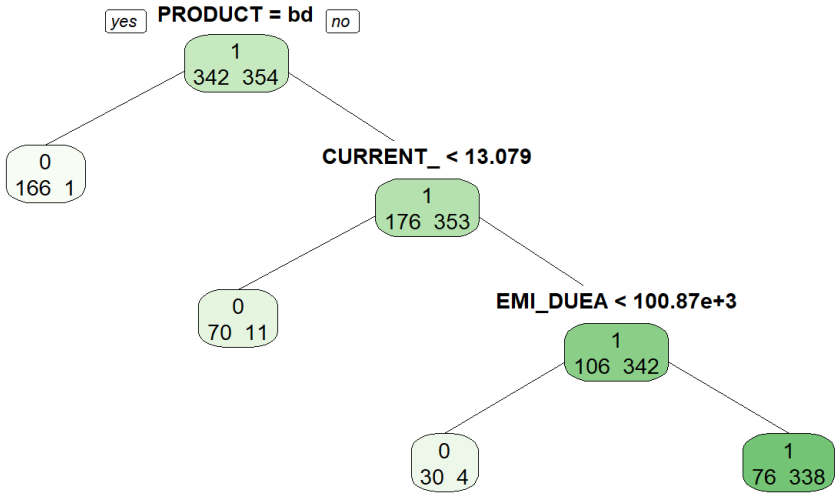


Figure 6: Decision Tree Example

The Most Important Variables are calculated after the Monte Carlo Simulation is run on the tuned Decision Trees. The extracted most frequently used variables include PRODUCT, PAID INTEREST, EMI DUEAMT, EMI RECEIVED AMT, and CURRENT INTEREST RATE.

Variable	Frequency
PRODUCT	999
PAID_INTEREST	997
EMI_DUEAMT	921
EMI_RECEIVED_AMT	810
CURRENT_INTEREST_RATE_MAX	618
ORIGNAL_INTEREST_RATE	506
PAID_PRINCIPAL	63
CURRENT_INTEREST_RATE_MIN	49
MIN_EMI_AMOUNT	17
DIFF_CURRENT_INTEREST_RATE_MAX_MIN	9
CURRENT_INTEREST_RATE	4
DIFF_ORIGINAL_CURRENT_INTEREST_RATE	4
DIFF_ORIGINAL_CURRENT_TENOR	3

Table 4: Most Frequently Used Variables in Decision Tree

5.2 Random Forest

Further model building is done with Classification Method Random Forests, which, while harder to interpret, provide more accurate results by using hundreds of decisions trees.

Tuning the RandomForest can lead to more accurate and sensitive results. The following tuning methods were done with both sensitivity and accurary as metrics.

Number of Trees is the amount of trees sampled for the random forest. Values tested for number of trees were 100:1000 by 10 and tested on balanced data. Figure 7 shows a peak at **ntrees=800**, with accuracy = 0.88 and sensitivity = 0.89

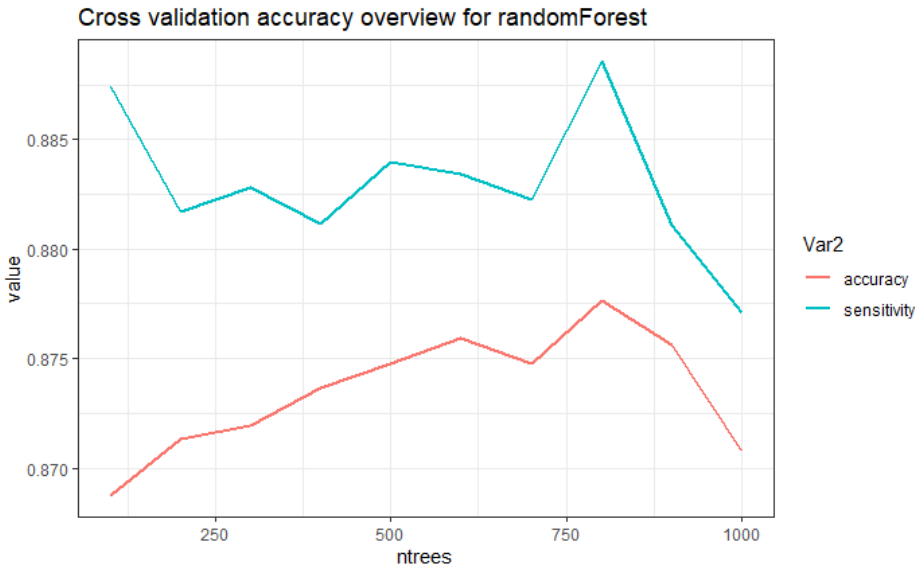


Figure 7: ntree tuning random forest

mtry is the amount of variables to randomly sample at each split also affects the model, and tuning was performed in the same way as the number of trees. Figure 8 shows an increasing accuracy and sensitivity as the value of mtry increases, with accuracy and sensitivity meeting at mtry=8. As sensitivity is taken into a higher consideration for this project

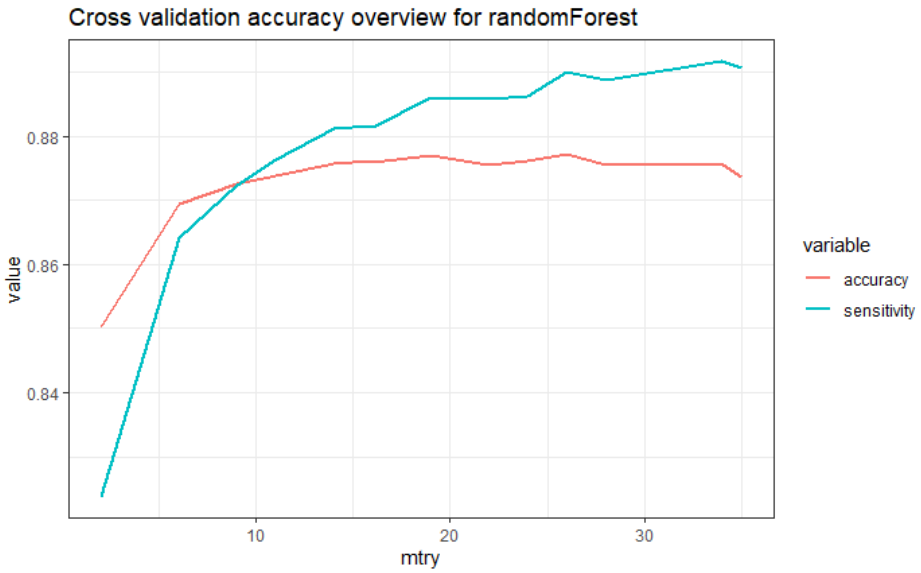


Figure 8: mtry tuning random forest

The Most Important Variables extracted from these models in the randomForest Package are taken into consideration during the variable selection process in different models.

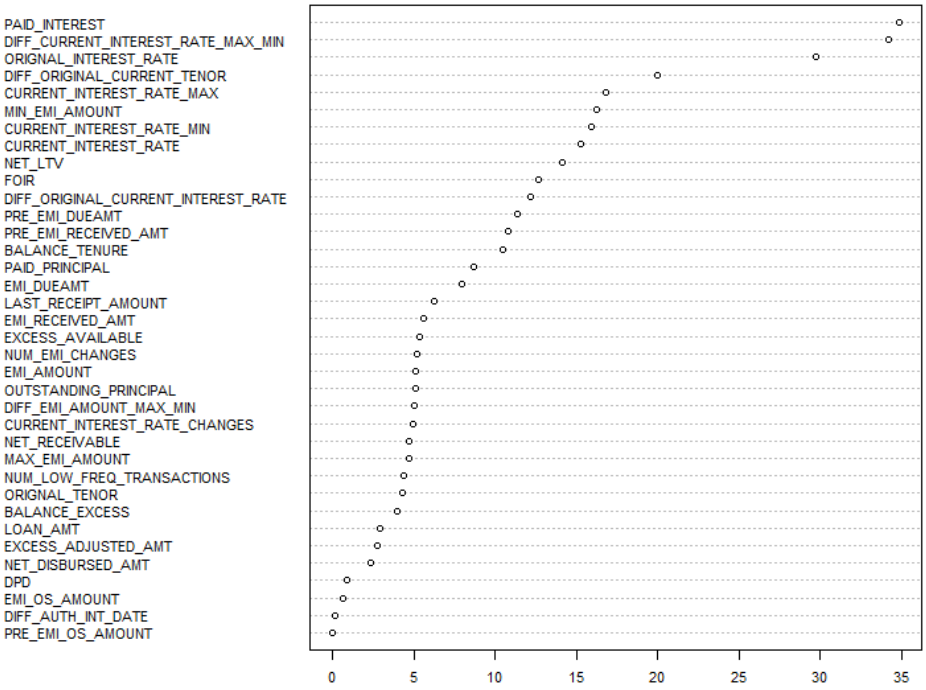


Figure 9: Most Important Variables by Gini Index

	freq
PAID_INTEREST	55.77
DIFF_CURRENT_INTEREST_RATE_MAX_MIN	49.54
PRODUCTLAP	47.73
MIN_EMI_AMOUNT	37.36
BALANCE_TENURE	34.45
EMI_DUEAMT	26.43
CURRENT_INTEREST_RATE_MIN	23.35
CURRENT_INTEREST_RATE	21.59
NET_LTV	21.47
LAST_RECEIPT_AMOUNT	21.04

Table 5: Most Important Variables, Random Forest

5.3 Boosting

The XGBoost package in R is used to preform building the model and tuning the parameters.

Max Tree Depth was found using a tuning grid in the Caret Package, and the results are in Figure 10. The plot displays the maximum depths over different boosting iterations and learning rates. The highest accuracy is at Max Tree depth = 6 and learning rate(eta) = 0.05

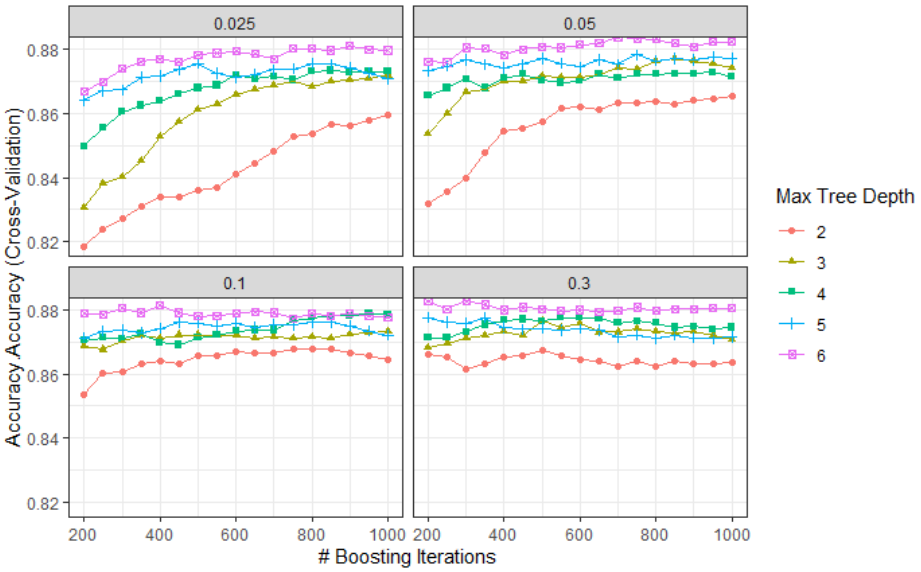


Figure 10: Max Tree Depth Tuning in Boosting

Gamma controls regularization, which is important when such a large dataset can be prone to overfitting. Using TuneGrid, values 0:1 with step sizes of 0.02 are fed through the model. All trees increase quickly and level off around 250 boosting iterations. In all models, Minimum Loss Reduction 0 performs well without reaching overfitting.

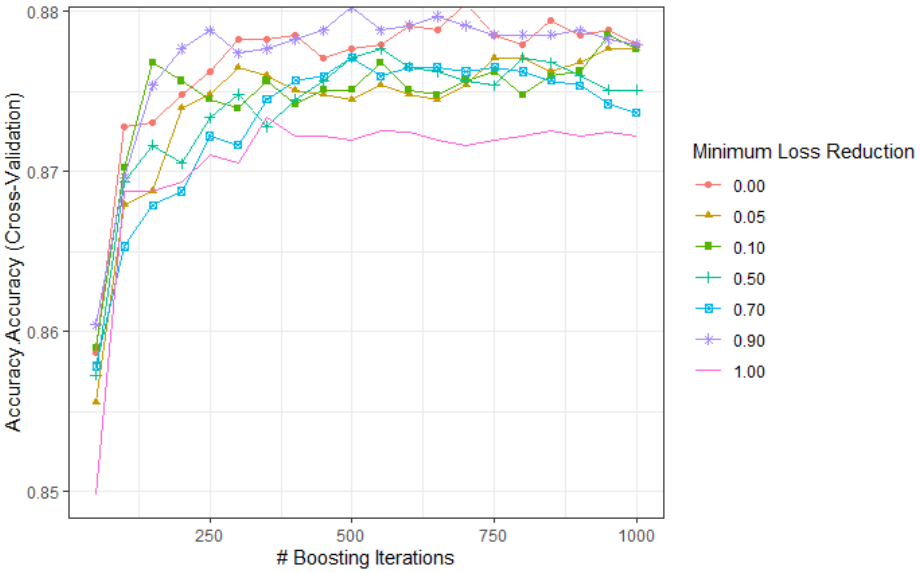


Figure 11: Gamma Tuning in Boosting

The Minimum Child Weight is tuned for values 1:3. The results in Figure 12 show that 3 is our optimal parameter, which reduces overfitting.

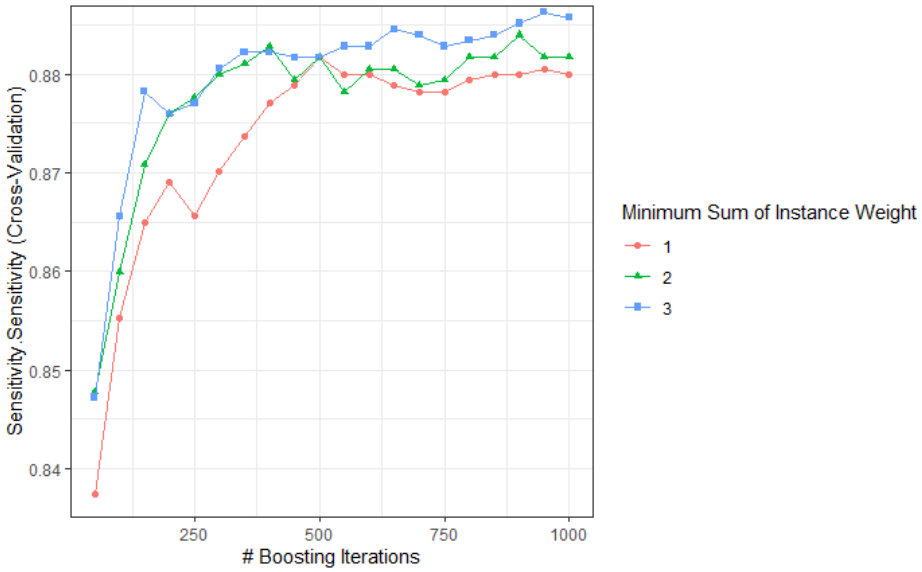


Figure 12: minimum child weight in Boosting

Chosen Parameters Our final chosen parameters are as follows:

nrounds	max_depth	eta	gamma	colsample_bytree	min_child_weight	subsample
700	6	0.05	0.00	1.00	1	1

Most Important Variables are found in this model through the Gain: the improvement in accuracy brought by a feature to the branches it is on; Cover: the number of times a feature is used to split data across trees weighted by training data points; and Frequency: how often the feature is used in the model. The top variables include ORIGNAL_INTEREST_RATE, PAID_INTEREST, DIFF_CURRENT_INTEREST_RATE_MAX_MIN, MIN_EMI_AMOUNT, and CURRENT_INTEREST_RATE, (Table 6, Fig. 13). The variables are similar to ones we have previously seen, and they will be discussed in another section.

	Variable	Gain	Cover	Frequency
1	ORIGNAL_INTEREST_RATE	0.09611	0.06822	0.04641
2	PAID_INTEREST	0.08936	0.06037	0.05016
3	DIFF_CURRENT_INTEREST_RATE_MAX_MIN	0.08911	0.03927	0.01945
4	MIN_EMI_AMOUNT	0.07039	0.07808	0.06368
5	CURRENT_INTEREST_RATE	0.05971	0.03704	0.03653
6	DIFF_ORIGINAL_CURRENT_TENOR	0.05375	0.03008	0.01809
7	CURRENT_INTEREST_RATE_MIN	0.04882	0.04309	0.03397
8	BALANCE_TENURE	0.04505	0.04007	0.05359
9	DIFF_ORIGINAL_CURRENT_INTEREST_RATE	0.04346	0.03168	0.01663
10	CURRENT_INTEREST_RATE_MAX	0.03256	0.02484	0.02336

Table 6: Most Important Variables in Boosting

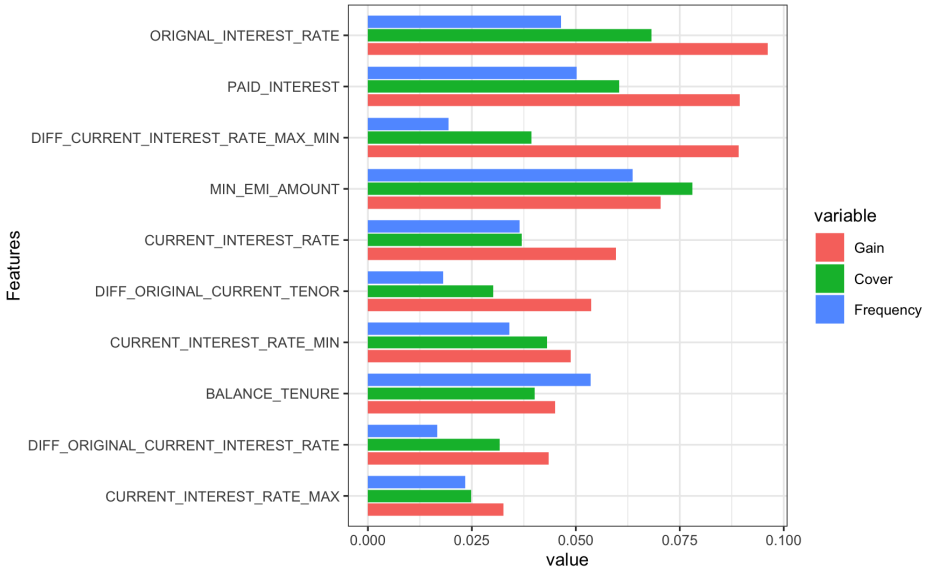


Figure 13: Most Important Variables in Boosting

5.4 Logistic Regression

classification method Logistic Regression was also modeled to predict Foreclosure. The large number of variables make the it difficult to build the model, so different sets of variables were chosen in each model:

1. Decision Tree Variables:
   PRODUCT + PAID_INTEREST + EMI_DUEAMT + EMI_RECEIVED_AMT +
   CURRENT_INTEREST_RATE_MAX

2. Correlation Matrix Variables:
   DIFF_ORIGINAL_CURRENT_TENOR +
   DIFF_ORIGINAL_CURRENT_INTEREST_RATE + CURRENT_INTEREST_RATE_MIN

3. Stepwise Selection

Logistic Regression requires tuning for λ, skewing the data more towards 1 or 0 based on the dataset. A Monte Carlo simulation with different values of lambda reported $\lambda = 0.64$ as the optimal value (Fig. 14). Sensitivity is more important in this project, so $\lambda = 0.66$ is used in the final model.

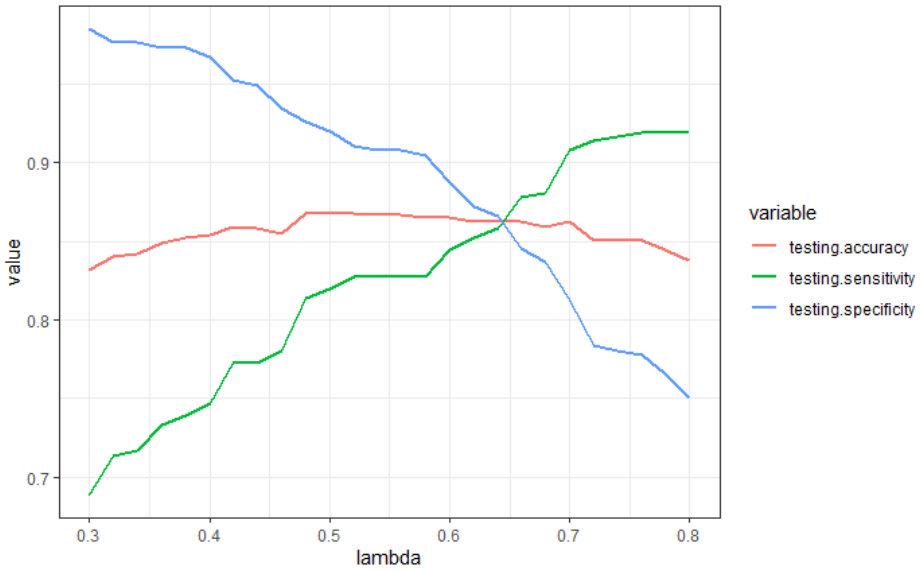


Figure 14: optimal λ

5.5 K-nearest neighbours

The k nearest neighbour algorithm is a classification method used to predict if a loan is going into foreclosure. The **amount of nearest neighbours** (K) that is best applicable for this dataset is searched for using **10 fold cross validation** on balanced datasets (where number of foreclosures and non-foreclosures are equal). There are 4 pre treatments of the data used:

1. The full dataset with all 37 variables
2. A dataset with the 5 first principal components of the full dataset
3. A dataset with the 10 first principal components of the full dataset
4. The dataset with the 5 predictors chosen based on the decision tree

The results of the cross validations is illustrated in figures 15, 21 and 22. These plots show that the (relatively low) accuracies, sensitivities and specificities flatten out at K = 8. Because of this, K = 8 is chosen to fit the full model with. All of the models where fitted using the knn method from the 'class' package in R. The sensitivity and specificity images are in appendix C.1.

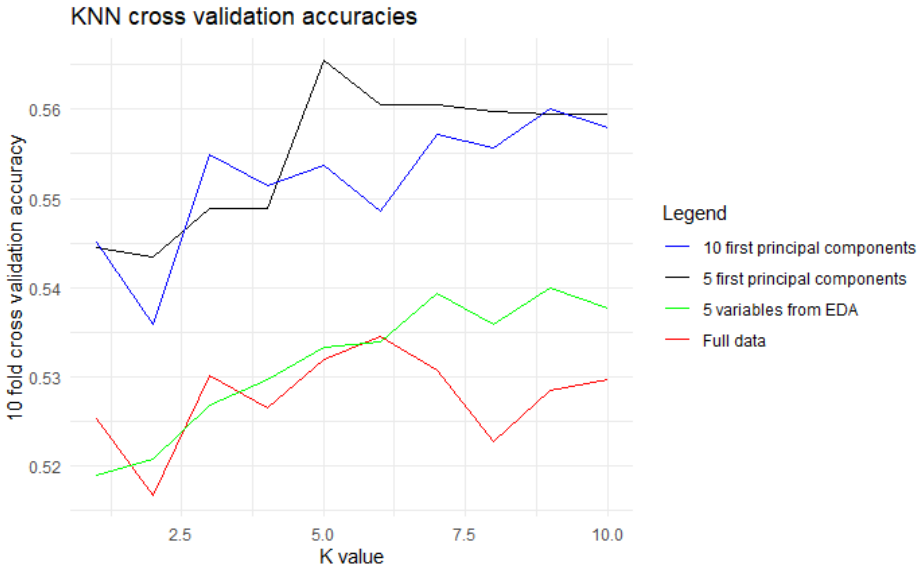
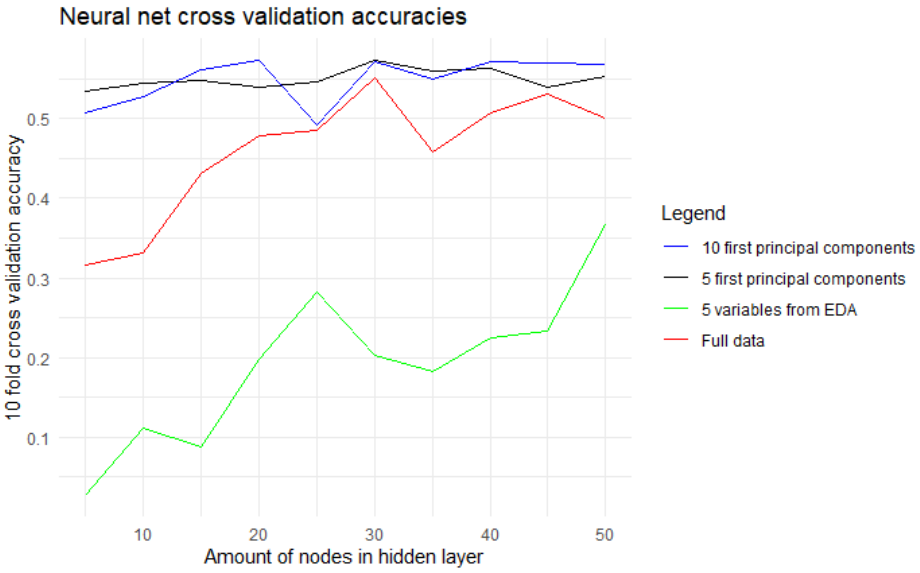


Figure 15: Cross validation plot for KNN accuracy

5.6 Neural network

Ten fold cross validation was performed on the balanced dataset to determine the most **effective amount of nodes** for a single layer neural network. The nnet function from the 'nnet' package in R was used to fit the data. The performance of the model on the dataset is relatively poor, barely reaching above 50% accuracy for all of the 4 dataset options (5 PC's, 10 PC's, partial based on EDA and full dataset). The results of the cross validation accuracies are shown in figure 16, the sensitivities and specificities are shown in figures 25 and 26 in appendix C.3. Because of the low changes in the accuracies based on the amount of nodes, the amount of nodes was chosen to be 20 to lower computational resources needed.

Figure 16: Cross validation plot for neural network accuracy



5.7 Linear support vector machine

Like the other models, SVM uses **10 fold cross validation** to determine the best hyper parameter for the balanced dataset. The **cost parameter** which influences the margin size around the separator is tuned. The same 4 pre treatments as for the KNN and neural network model where used. This is shown in figures 17,23 and 24.

Based on these, an optimal cost parameter of $10^1 = 10$ is chosen. The specificities (Figure 24) are higher around a cost parameter of $10^3 = 1000$, however the accuracies and sensitivities start declining after 10, these are found to be more important results because of the higher importance of finding someone going to foreclosure. The sensitivity and specificity images are in appendix C.2. All of the models where fitted using the svm function from the 'e1071' package in R. For this method a linear kernel was used.

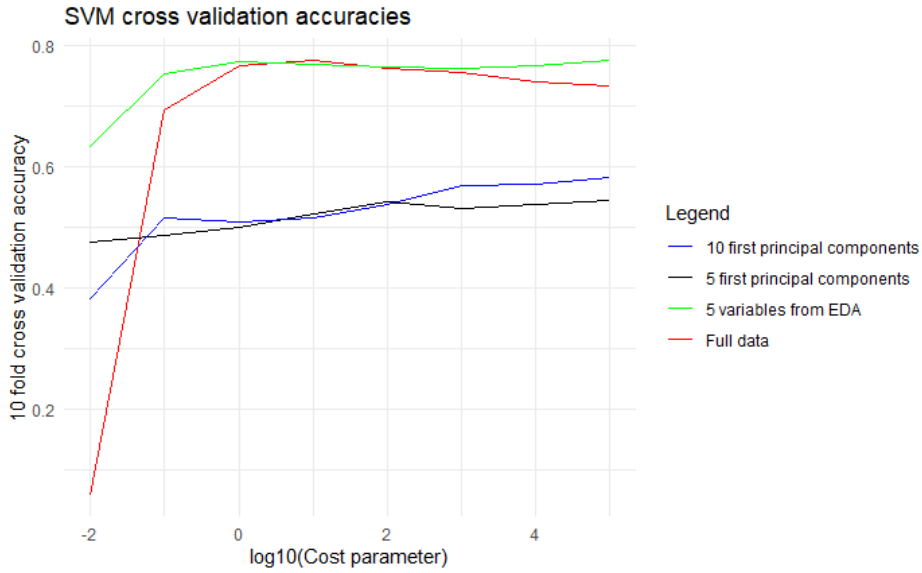


Figure 17: Cross validation plot for SVM accuracy

5.8 T-Test

T-tests are used to determine if there is a statistical difference between the best models based on their average performance and the variance found using Monte Carlo cross validation. This is a two sided test performed the following way:

$$S_p = \sqrt{\frac{(n_1 - 1 * s_1^2 + (n_2 - 1) * s_2^2}{n_1 + n_2 - 2}}$$

In this case s_p represents the pooled standard deviation between the two variables. Then the t-value is calculated the following way:

$$t = \frac{\bar{x_1} - \bar{x_2}}{s_p * \sqrt{1/n_1 + 1/n_2}}$$

Based on the degrees of freedom $(n_1 + N_2 + 2)$ and the t value the p value is looked up in a table. For all of the values in this report the amount of iterations n is 100 per model.

6 Analysis/results

The summary of the results for all the models is shown in Table 7.

Method	Data preprocessing	Accuracy	Accuracy variance	Sensitivity	Sensitivity variance	Interpretabilty
SVM	PCA(5)	0.69000	0.00025	0.70300	0.00043	low
SVM	PCA(10)	0.70000	0.00030	0.71000	0.00042	low
SVM	Full dataset	0.88700	0.00015	0.89800	0.00013	medium
SVM	Partial dataset	0.85970	0.00015	0.87800	0.00001	high
NN	PCA(5)	0.68520	0.00028	0.69013	0.00038	low
NN	PCA(10)	0.69180	0.00027	0.70400	0.00035	low
NN	Full dataset	0.63789	0.00118	0.63789	0.00523	low
NN	Partial dataset	0.56000	0.00309	0.65577	0.00549	low
RandomForest	Full dataset	0.82619	0.24000	0.81297	0.81000	medium
Boosting	Full dataset	0.83050	0.00010	0.93660	0.00020	medium
DecisionTree	Full dataset	0.86080	0.00010	0.91220	0.00090	high
LogR	tree variables	0.83915	0.00019	0.83302	0.00022	high
LogR	eda variables	0.55461	0.00036	0.43488	0.00020	high
LogR	step regression	0.68843	0.00032	0.61298	0.00043	high
KNN	PCA(5)	0.68300	0.00033	0.68700	0.00038	low
KNN	PCA(10)	0.69600	0.00029	0.69500	0.00038	low
KNN	Full dataset	0.44000	0.00040	0.32300	0.00032	medium
KNN	Partial dataset	0.65300	0.00036	0.65700	0.00039	medium

Table 7: Final Results Across All Methods

In the methods used, the Decision Tree, SVM, and Logistic Regression on 5 variables performed the best in accuracy and sensitivity.

The Decision tree is robust to outliers, interpretable, and it provides insight to feature importance in its output. The Decision Tree and Logistic Regression are some of the most interpretable models, so it is very easy for a bank employee or customer to understand. Because of the greedy algorithm, the model is also less computationally expensive.

The Decision Tree can be prone to overfitting, but the tuning parameters we used were conservative to avoid this problem. The decision tree can also face problems with unbalanced data, but the data was balanced before training the models. This is something that could be performed at a bank also. The tuning parameters on the Logistic Regression model also helped with overfitting, while also skewing the model towards higher sensitivity. Because it only used 5 variables, it is easily interpretable.

Generally the neural networks and KNN models performed quite poorly. For neural networks, this could be explained by the high dimensionality of the data and outliers in the data, for which the neurals networks are prone to overfitting. Another problem with using neural networks in this model is that after balancing the data, there were 3590 data points, which might have been too little with the high amount of predictors. The neural network models also take relatively long to train.

The KNN models performed worse than other model types, with the model fitted on the full dataset performing worst. This indicates that this dataset with high variable dimensionality that is highly mixed cannot be separated by euclidian distance alone. The data is better suited to be classified by planes within the dataspace. Benefits of the KNN models are that they are fast to train. The model is also not easy to interpret without looking at the data closely.

The point about the KNN model is strengthened by the high performance of the linear support vector machine models. The SVM model is not very susceptible to outliers and overfitting due to the margin method, and thus

manages to separate the data around the boundary quite well. However due to the long training time required for the model a method with similar results but with higher interpretibility and/or lower amount of computational resources needed might be preferable.

The Boosting model is one of the most accurate models, and is robust to overfitting, which can be a problem with Random Forest models. Another advantage is the output of most frequently used variables, which makes the model easy to explain. An employee can focus on certain metrics when deciding if a client should be contacted because their loan is in risk of foreclosure. However, this model was the most difficult to train and took a long time and computational energy to produce results. Because of the size of this dataset, it is not an appropriate choice.

The Random Forest model performed worse than the other CART methods. While the ensemble method is an advantage of Random Forest, it is also computational expensive, and as the results are not very accurate, it does not have an advantage over the other CART methods.

The linear support vector machine models perform well using the untransformed datasets (full and the 5 predictors from the EDA). They are also moderately interpretable because they form linear decision boundaries along the n dimensional space. The boundary is not overly dependent on outliers because of the margin method. In this case the margin is determined by the cost parameter (C=10).

Overall the decision tree, logistic regression on the 5 selected variables and the SVM models showed the highest accuracies and sensitivities. Out of all of these model results only the SVM model with full data is distinctively different from the other models based on accuracies and sensitivities ($p < 0.05$), this can be seen in table 8. Because of this more research (model iterations) are needed to determine the best model with a lower amount of predictors as described in the report goal.

	SVM full	SVM partial	Decision tree	Logistic regression partial
SVM full	x	0.01	0.03	0.00
SVM partial		x	0.91	0.10
Decision tree	0.03	0.03	x	0.08
Logistic regression partial	0.00	0.10	0.08	x

Table 8: t-test Results

Variables

In all CART methods we extracted the most important variables. The top five variables for each CART method are as in Table 9.

Decision Tree	PRODUCT PAID_INTEREST EMI_DUEAMT EMI_RECEIVED_AMT CURRENT_INTEREST_RATE_MAX
Random Forest	PAID_INTEREST DIFF_CURRENT_INTEREST_RATE_MAX_MIN PRODUCTLAP MIN_EMI_AMOUNT BALANCE_TENURE
Boosting (by gain)	ORIGINAL_INTEREST_RATE PAID_INEREST DIFF_CURRENT_INTEREST_RATE_MAX_MIN MIN_EMI_AMOUNT CURRENT_INTEREST_RATE

Table 9: Top Variables for CART Methods

In all models, paid interest, some form of EMI amount, and some form of Current Interest Rate are in the top 5 most important variables. This indicates that these variables should be especially scrutinized when assessing if a

loan is in danger of foreclosure. As stated in the project goals, building a model is not to preclude customers from receiving a loan, but to be able to better assess if they are in danger of foreclosure. The amount due and interest rate change throughout the loan's life cycle, and a banker in tune with these changes would be able to help her customer.

7 Conclusions & Findings

Based on the results in chapter 6 it can be stated the full data SVM model, partial data SVM model, decision tree and partial data logistic regression models are performing the best on the dataset. It can be stated the full SVM model does perform strictly the best, but not to a high degree. Because it is not to a high degree, no strong conclusion can be drawn on which model is the best out of the other models because of the higher variance compared to the amount of iterations per model (100). It can be concluded however that these models perform very similarly and because of this the decision tree model or the partial data logistic regression would be recommended to use to determine the probability of someone going to foreclosure. The primary reason for this is because the decision tree model can be explained relatively easily to a bank employee, and the logistic regression model only required 5 predictors while also being easy to interpret. The 5 predictors that are recommended are:

- Product
- Paid interest
- EMI due amount
- EMI received amount
- Current interest rate minimum

This project does not provide solutions for how to change the risk of foreclosure. This project identifies the risk, after which an expert can help the client. When a bank employee has information about these five variables above, she can reach out to her client to help him steer away from going into foreclosure. With these five variables that build accurate decision tree and logistic models, the foreclosure rate can be lessened by helping the client understand their risks and determine a new path towards repaying their loan.

8 Learning points

Read the Data Dictionary

This dataset has 53 variables, not all of which were relevant. As described in Section 3, not all variables were relevant to the project, and some of the data would have been detrimental to the results. Variables such as the date of last payment inferred that the loan was paid in full, those there was no foreclosure. Having this variable as part of the data would have made for an inaccurate model, as it would be a result of what the model is built to predict.

Balance the Data

In the first round of model building, all of the results had very high accuracy and sensitivity. The models appeared to be working perfectly. The nature of the data, with only 9% having an alternate response, lent to high accuracy in the models. The high numbers were a red flag that our data needed more investigation.

A simple test of accuracy and sensitivity on a Logistic Model illuminated the problems within the data. While tuning parameter λ went from 0 to 1, the accuracy and sensitivity of the model stayed about 90%, which is not realistic.

After balancing the data as described in Section 4, the accuracy and sensitivity of the model responded to the shifting tuning parameter, indicating that the model was a better fit.

A caveat of Balancing the Data is that for each iteration, only 1795 data points of the 18217 data points reflecting no foreclosure are used. For all of the data points to be considered, at least 10 simulations must be run for every dataset. For Monte Carlo simulations, at least 100 simulations are run to capture all datapoints.

Be Critical of Results

In line with the previous section, even after balancing data the results may be unrealistic. For example, the logistic regression models continuously gave a sensitivity equal 1, even though the accuracy was much lower and it was impossible to have a sensitivity of 1 with the tuning parameters. This required going back into the code and finding where the problems occured. Because the results were created in Monte Carlo Simulations, the results had to be run a few times to ensure the error was fixed.

Hyperparameter Tuning

Hyperparameter Tuning is an important part of any model building, and with a high dimensional dataset it is essential. For these models, they reduced the chances of over-fitting and the computational complexity of building the models. After trying to run the models and waiting several days, it was easy to see that tuning could cut that time of building the model.

Without tuning, the models had very high training accuracy and sensitivity, markers of an overfit models. The runing paramters brought these metrics to a reasonable level, leading to better models to predict the testing data.

A Summary of variables

	Number in correlation matrix	Description
AgreementID	Not included	Agreement ID of the loan account (a customer can have multiple loans)
Authorization_date	Not included	Authorization date of the loan
Balance_excess	1	Balance of excess amount
Balance_tenure	2	Remaining tenure
City	Not included	City of origination
Completed_tenure	Not included	Completed tenure
Current_interest_rate	3	Current rate of interest on the loan.
Current_interest_rate_max	4	Maximum value of the CURRENT ROI across transactions
Current_interest_rate_min	5	Minimum value of the CURRENT ROI across transactions
Current_interest rate_changes	6	Number of times the CURRENT ROI has changed
Current_tenor	Not included	Current tenor of the loan, length of time until loan is due
Customer_ID	Not included	Unique Customer ID given to each customer
Diff_auth_int_date	7	Difference between authorization and interest start date
Diff_current_current interest_rate_max_min	8	Difference between the maximum and minimum interest rate per agreement
Diff_EMI_emount_max_min	9	Difference between maximum and minimum EMI AMOUNT
Diff_original_current interest_rate	10	Difference in original ROI and current ROI (ORIGNAL_ROI CURRENT_ROI)
Diff_original_current_tenor	11	Difference in original and current tenor (ORIGNAL_TENOR CURRENT_TENOR)
DPD	12	Days past due
Due day	Not included	Next due date of the loan
EMI_amount	13	Mode of the receipt amount
EMI_DUEAMT	14	EMI due amountEMI outstanding amount
EMI_OS_Amount	15	EMI outstanding amount
EMI_received$_AMT$	16	EMI received amount
Excess_adjusted_AMT	17	Excess adjusted amount
Excess_available	18	Excess received
FOIR	19	Fixed obligation to income ratio (Value should range from 0-1 – Derived variable)
Interest_start_date	Not included	Interest start date on the loan
Last_receipt_amount	20	Last receipt amount
Last_receipt_date	Not included	Last receipt date
Latest_transaction_month	Not included	Month of last receipt date. In case account is Foreclosed, it will be month of Foreclosure
Loan_AMT	21	Loan amount which was sanctioned
Max_EMI_amount	22	Maximum receipt amount
Min_EMI_amount	23	Minimum receipt amount
Monthopening	Not included	Month of opening
Net_disbursed_AMT	24	Amount that was disbursed
NET_LTV	25	Net Loan to Value ratio (Value ranges from 0-100 (in %) Derived variable)
Net_receivable	26	Net receivable (EMI_DUEAMT EMI_RECEIVED_AMT = EMI_OS_AMOUNT) + (EXCESS_AVAILABLE EXCESS_ADJUSTED_AMT = BALANCE_EXCESS) = NET_RECEIVABLE)
Num_EMI_changes	27	Number of different values in the receipts amount
Num_low_freq_transactoins	28	Number of transactions done in less than 28 days
Original_interest_rate	29	Original rate of interest on the loan (when the loan was sanctioned). Renamed field
Original_tenor	30	Original tenor of the loan (when the loan was sanctioned)
Outstanding_principal	31	Outstanding principal
Paid_interest	32	Paid interst
Paid_principal	33	Paid principal
Pre_EMI_due_AMT	34	Pre EMI due amount for the loan
Pre_EMI_OS_amount	35	Pre EMI Outstanding amount
Pre_EMI_received_AMT	36	Pre EMI that was received
Product	Not included	Loan product
SchemeID	Not included	Scheme ID under which loan was given
NPA_in_last_month	Not included	Whether NPA in last month
NPA_in_current_month	Not included	Whether NPA in current month
MOB	Not included	Internal code
Foreclosure	37	ID to that indicates if the loan went to foreclosure

Table 10: Summary of variables

B Variable histograms

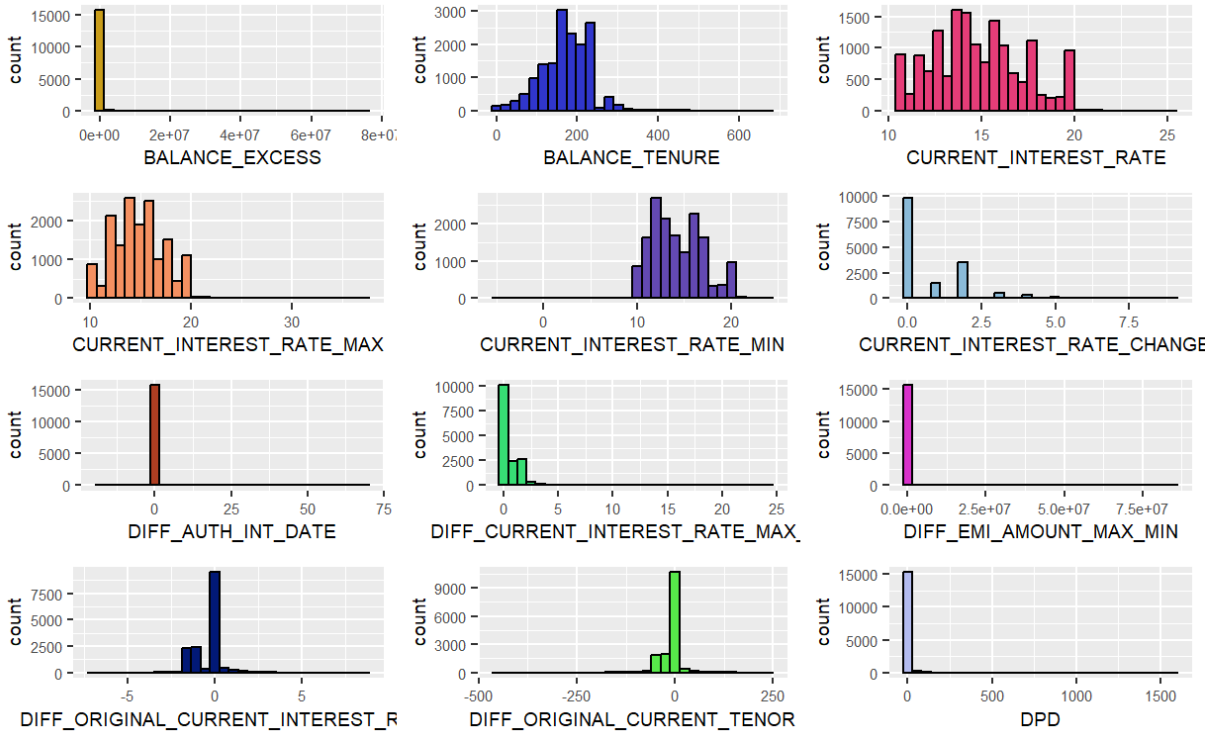


Figure 18: First set of histograms of data distribution

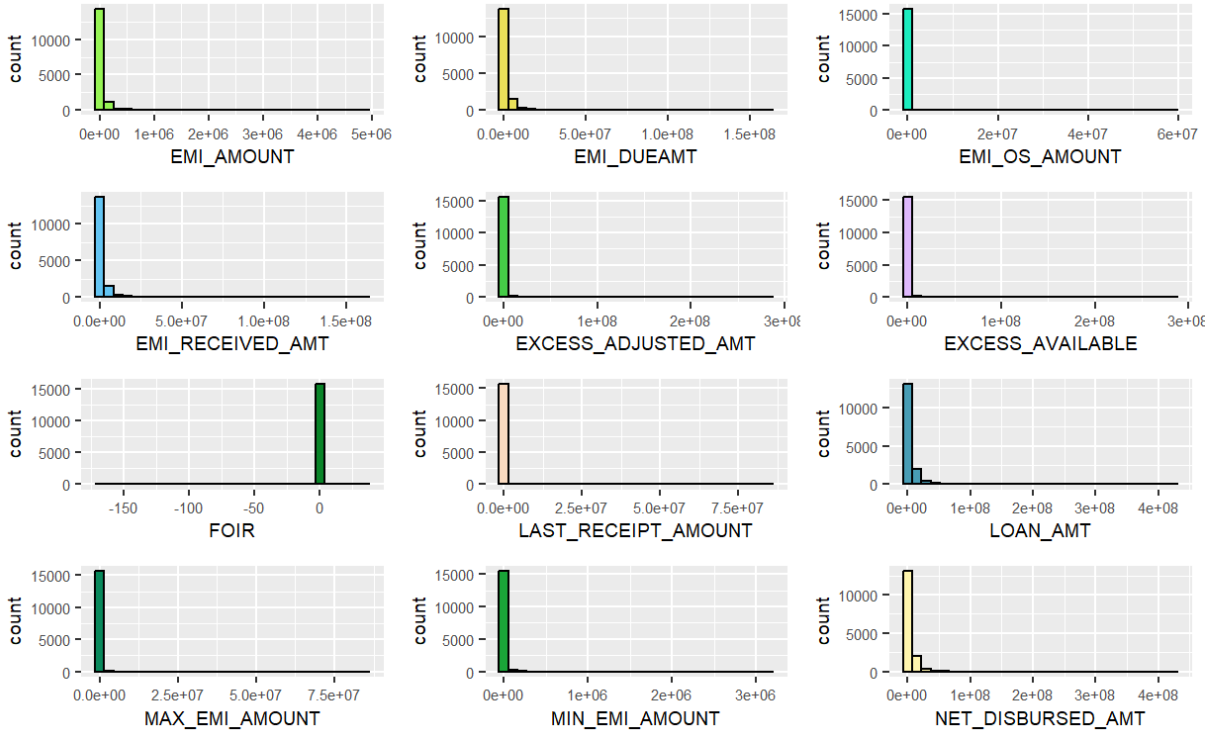


Figure 19: Second set of histograms of data distribution

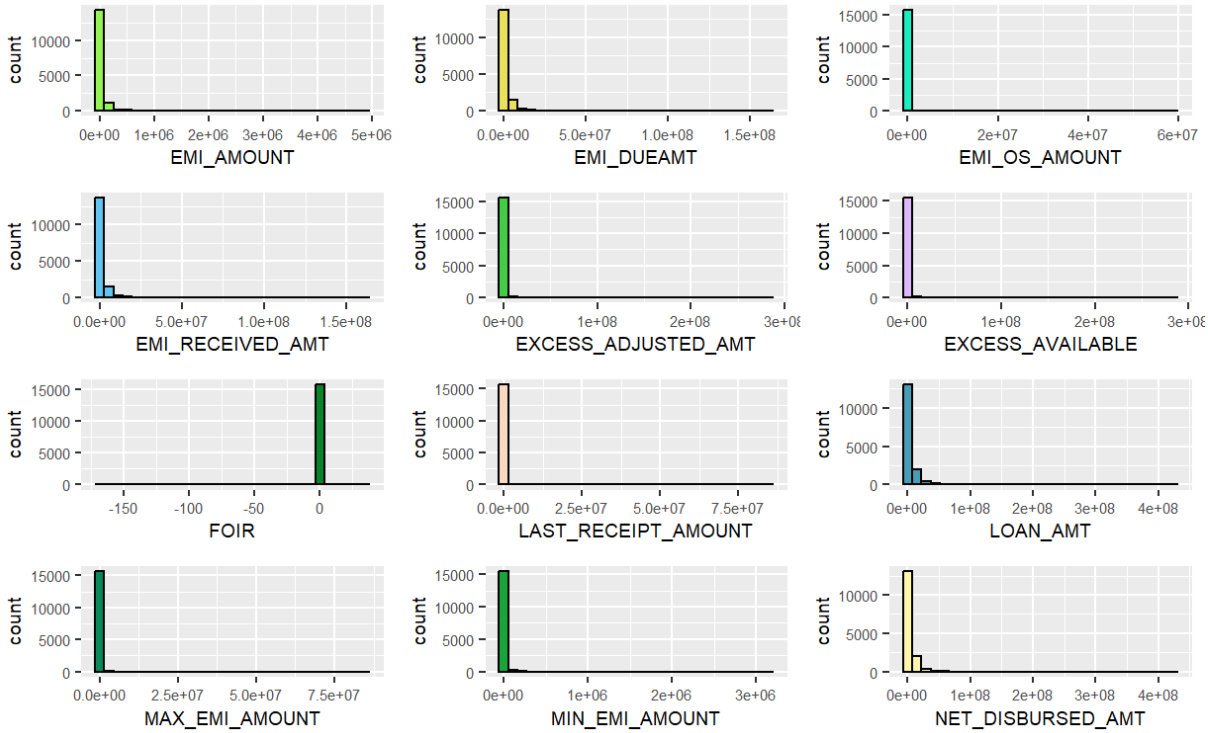


Figure 20: Third set of histograms of data distribution

C Cross validation images

In this chapter the sensitivity and specificity cross validation images are illustrated if applicable.

C.1 KNN cross validation images

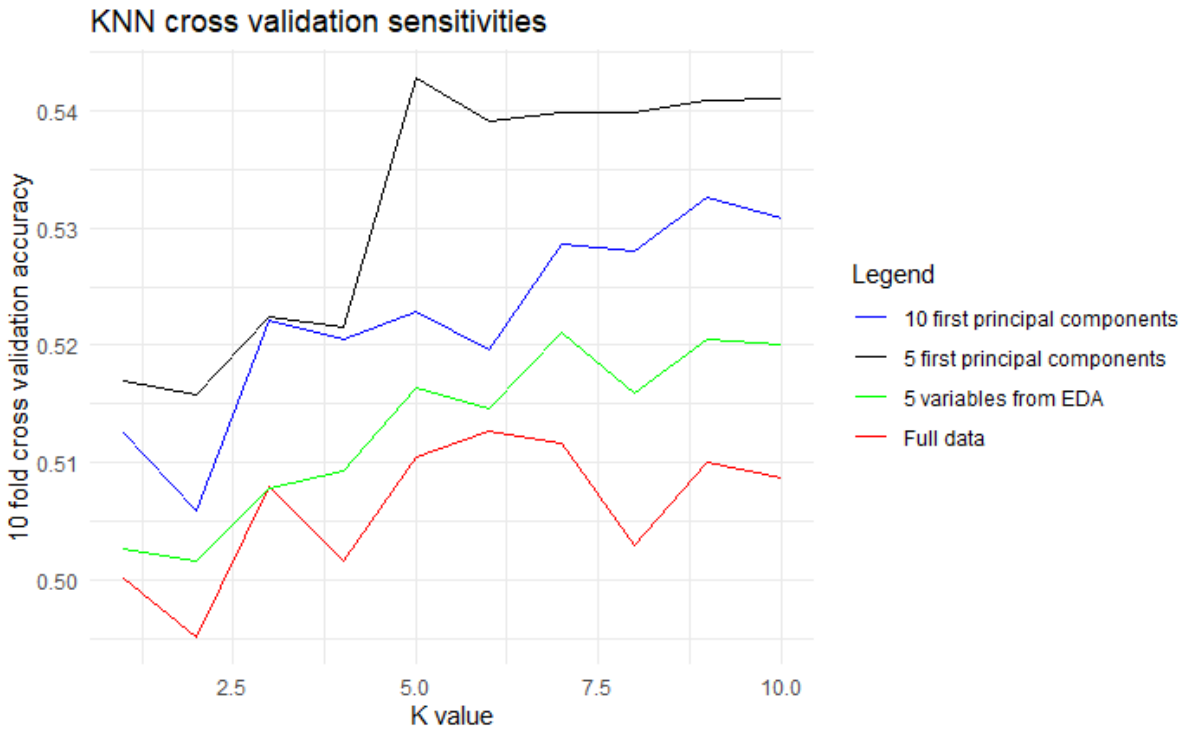


Figure 21: Cross validation plot for KNN specificity

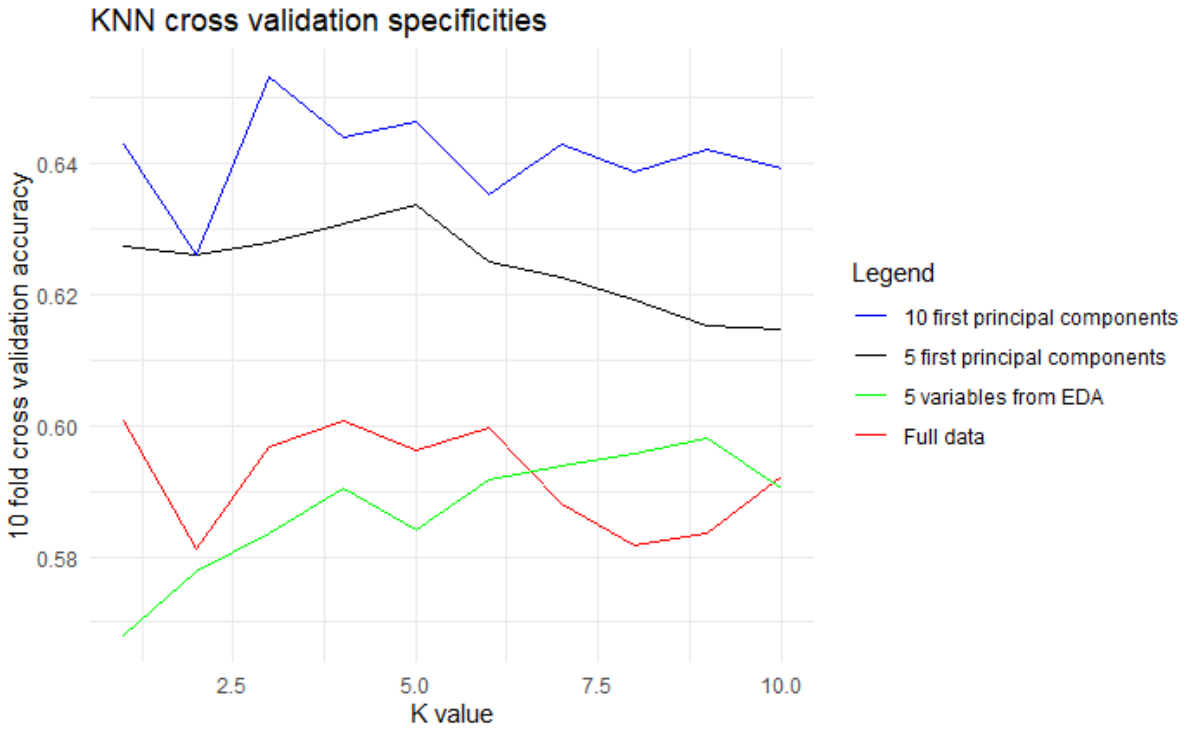


Figure 22: Cross validation plot for KNN accuracy

C.2 SVM cross validation images

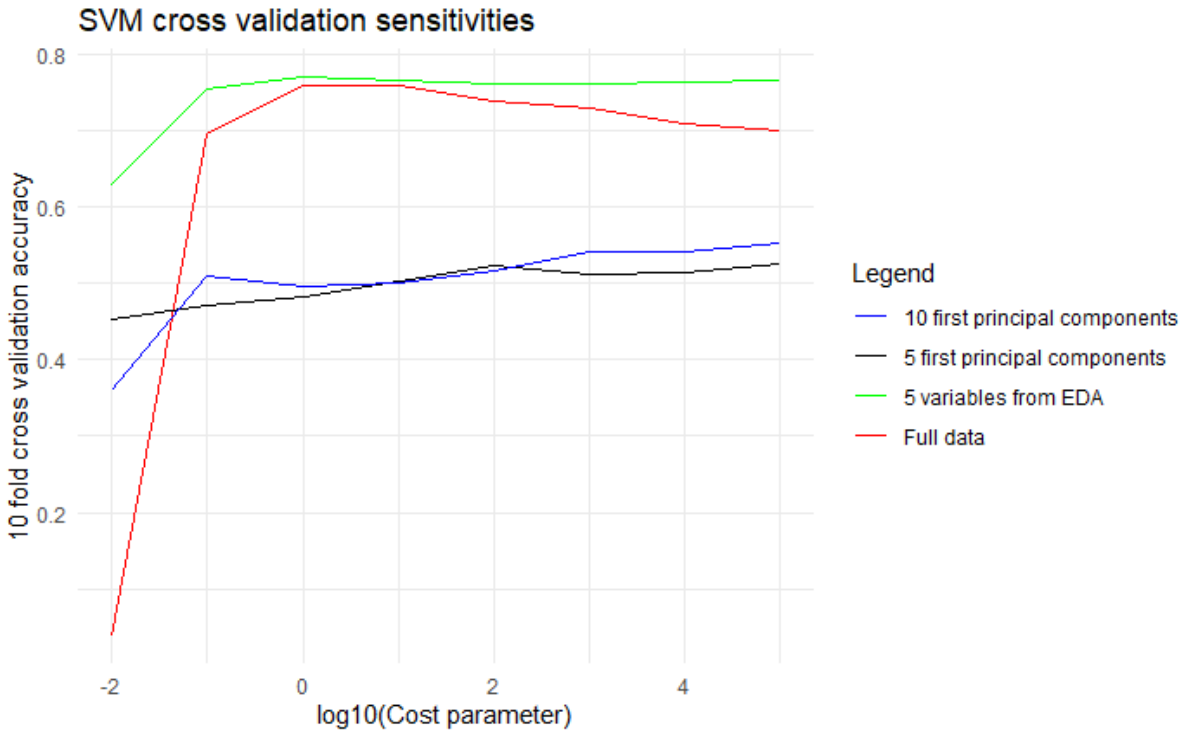


Figure 23: Cross validation plot for SVM sensitivity

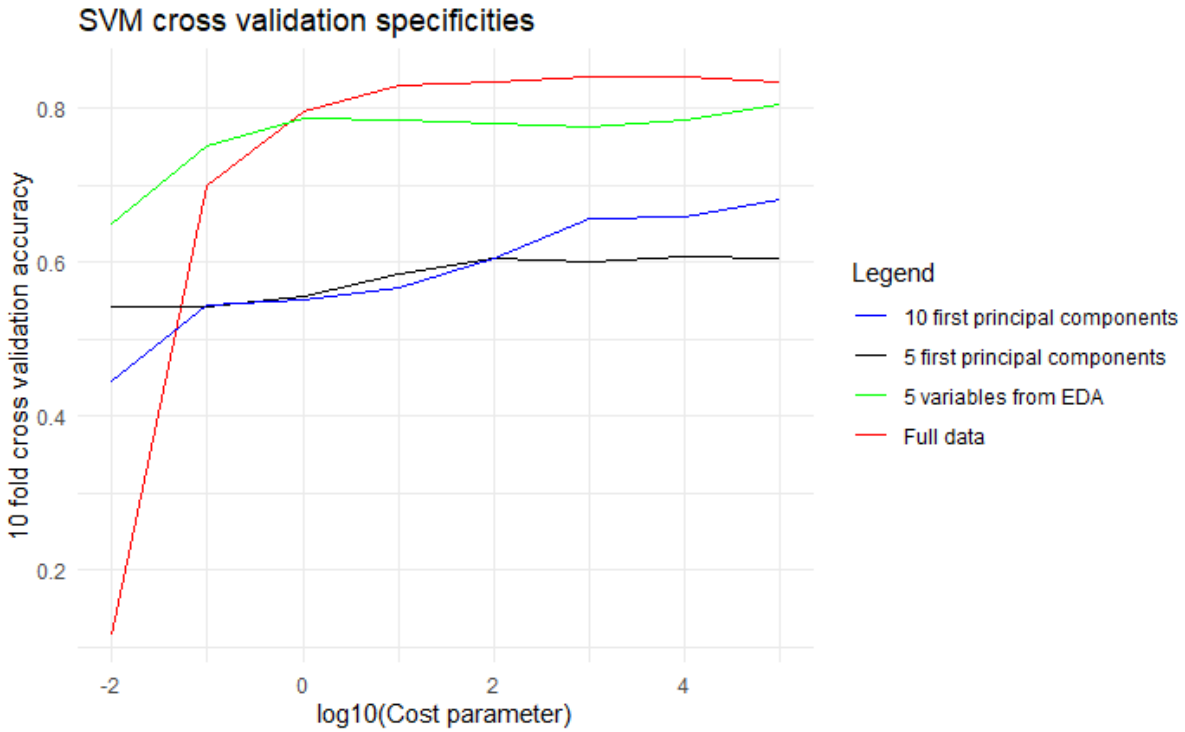


Figure 24: Cross validation plot for SVM specificity

C.3 Neural network cross validation images

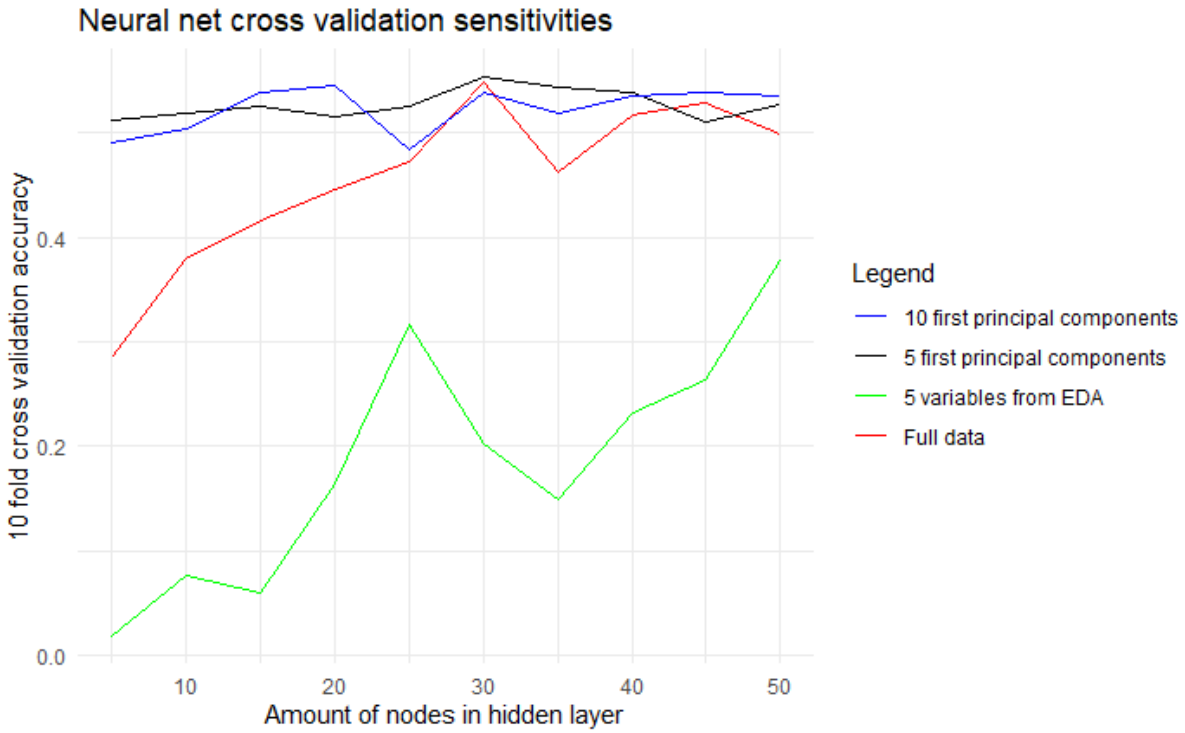


Figure 25: Cross validation plot for neural network sensitivity

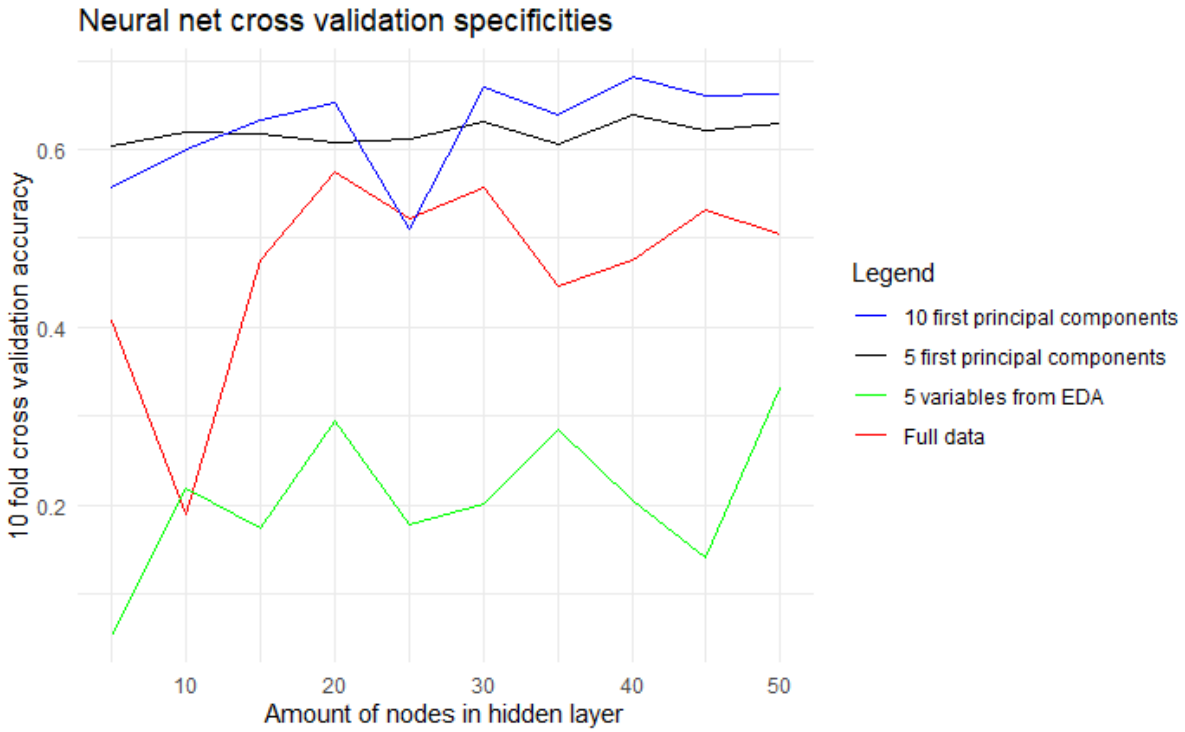


Figure 26: Cross validation plot for neural network specificity