

Loan Default Prediction



Project

Summary

Document

Table of Contents

- 1 Business Understanding
 - 1.1 Company profile
 - 1.2 Problem statement
 - 1.3 Stakeholder analysis
 - 1.4 Our proposal
- 2 Data understanding and preparation
 - 2.1 Baseline Model
 - 2.2 Dataset overview
 - 2.3 Key findings of the Dataset
 - 2.4 Data cleaning
 - 2.4.1 Rows reduction
 - 2.4.2 Columns reduction
 - 2.4.3 Missing Value Analysis – columns
 - 2.4.4 Missing values Analysis – Rows
 - 2.4.5 Outlier analysis
 - 2.4.6 Recoding of categorical variables
- 3 Evaluation Method
 - 3.1 Splitting Dataset
 - 3.2 Metric of success
 - 3.3 Data imputation
 - 3.4 Data Balancing
- 4 Modeling
 - 4.1 Feature selection
 - 4.2 Class weights (Charged off vs Fully Paid)
 - 4.3 Adding Number of Features
 - 4.4 Threshold of Probability for LR
 - 4.5 Results comparison on logistic regression model
 - 4.6 Results on Complex Algorithm1 – Elastic Net
 - 4.7 Results on Complex Algorithm2 – XgBoost
 - 4.8 Cost of Misclassification
- 5 Results
 - 5.1 Conclusion
 - 5.2 Recommendations to Business
- 6 Team

Business Understanding

1.1 Company Profile

Lending Club is a peer-to-peer lending platform, where investors deposit money with LC and borrowers apply for loan against the investment.



Current scenario



Outdated
Model that
predicts loan
default risk

Increased
loan
defaults



Decreased
investor
confidence

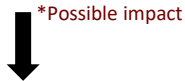


One of the main objective of LC is to maximize ROI of the investors and at the same time minimize the risk in order to provide a portfolio diversification opportunity for lenders and the potential to earn competitive returns compared to other investment instruments (stock market, CDs etc.) Lending Club enables borrowers to apply for personal loans and provides investors options to select loan listings to invest. Investors can select loans based on the information supplied about the borrower, amount of loan, loan grade, and loan purpose. Investors make money from interest. Lending Club makes money by charging borrowers an origination fee and investors a service fee. Lending Club enables borrowers to create loan listings on its website by supplying details about themselves and the loans that they would like to request. All loans are unsecured personal loans and can be between \$1,000 - \$40,000. On the basis of the borrower's credit score, credit history, desired loan amount and the borrower's debt-to-income ratio, Lending Club determines whether the borrower is credit worthy and assigns to its approved loans a credit grade that determines payable interest rate and fees.

1.2 Problem statement

The current model used by Lending club has a high misclassification rate, which is resulting in increased loan defaults than predicted. Due to this unpredictable increase in default, investors are losing trust and confidence in the organization. In order to revive their business and gain back the investors interests they need a better predicting model which can accurately as possible, classify the potential loan defaults in advance. By increasing the accuracy of prediction, number of default loans can be minimized. This reduction in number of defaults can increase the potential net returns of the investors.

Average interest rate for portfolio ⁶	14%
Estimated effect of charge-offs and prepayments ⁷	-8%
Effect of LendingClub fees ⁸	-1%
Annualized Net Return ⁹	= 5%



Average interest rate for portfolio ⁶	14%
Estimated effect of charge-offs and prepayments ⁷	-6%
Effect of LendingClub fees ⁸	-1%
Annualized Net Return ⁹	= 7%

⬆ Increase potential
Net return for the
lenders (investors)

⬇ Decrease
repayment and
collection cost

In the above example, let us assume that a customer who was predicted to pay the fully, has started defaulting on his/her payments. Assuming customer defaults about 8% of the total interest of 15%, each month, the net return for the investor decreases from 14% to 5%. Now eventually the customer stops payment and this customer will be 'charged off' by LC collection team.

On the other hand, if we approve loans to customers who will not default as easily as the current model then this 8% can be reduced to 6%(hypothetically) which will increase the net return of the investor from 5% to 7%(hypothetically). The goal is to not approve loans to customers who are most likely to default and eventually become 'charged off' accounts. This type of customers need to be predicted by our model more efficiently than the current model.

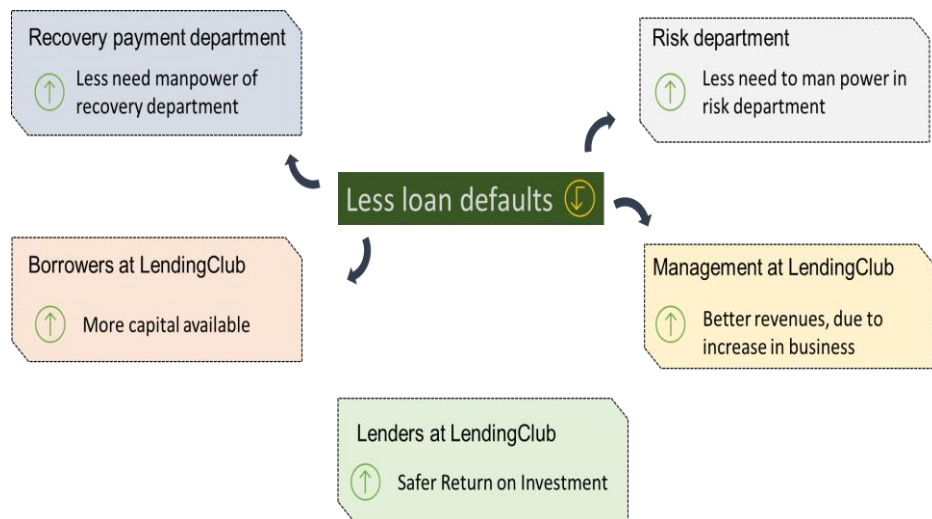
1.3 Stakeholder Analysis

A list of stakeholders along with their requirements, level of involvement, level of influence is arrived at during the initial phase. All stakeholders must understand the exact objective of this project and enable the fulfillment of the same with respective support. Our Data Science team will prepare a proper communication plan to keep all stakeholders updated of the progress and reviews of the milestones. Our Predictive Analytics Data model will impact and benefit following departments and stakeholders if there are less loan defaults:

Why ?

To understand the client perspective in a better way to deal with the business requirement.

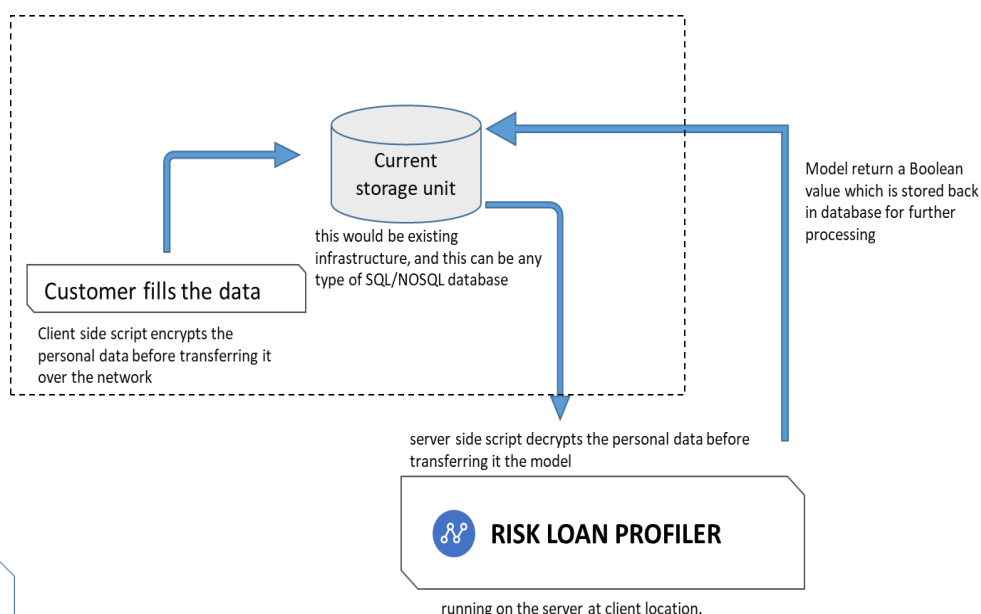
Based on sprints to complete each increment and evaluate the results at the end of each sprints further identification and to response to the client.



Our Data Science team will prepare a proper communication plan to keep all stakeholders updated of the progress and reviews of the milestones. Our Predictive Analytics Data model will impact and benefit following departments and stakeholders if there are less loan defaults: Our Data Science team will prepare a proper communication plan to keep all stakeholders updated of the progress and reviews of the milestones. Our Predictive Analytics Data model will impact and benefit following departments and stakeholders if there are less loan defaults:

1.4 Our proposal

A server based API that is fed with customer data and gives a prediction for default. This program is a machine learning based algorithm to check whether the person who is applying for a loan has the ability to pay back the loan or not. This is done by feeding the model legacy data of default and fully paid loans. The model learns from the data to classify loans.



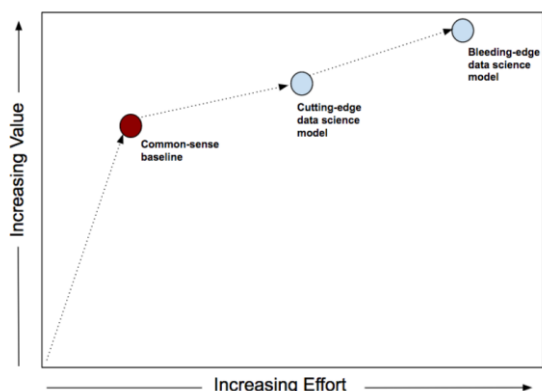
Value added features:

- Ability to give result in real time
- Ability to add another model which would change this model from black box, hence can give reasons for findings.
- Dashboard to show performance of model and other key features to access the quality

Data Understanding & Preparation

2.1 Baseline Model

A baseline is a method that uses heuristics, simple summary statistics, randomness, or machine learning to create predictions for a dataset. We can use these predictions to measure the baseline's performance (e.g., accuracy). This metric will be used to compare against other algorithms metric.



A common-sense baseline is how you would solve the problem if you didn't know any data science. Assume you don't know supervised learning, unsupervised learning, clustering, deep learning, whatever. Now ask yourself, how would I solve the problem?

Experienced practitioners do this routinely..

Rules of the Baseline Model



Salary is greater than 30.000 \$



The grade is that A,B,C

They first think about the data and the problem a bit, develop some intuition about what makes a solution good, and think about what to avoid. They talk to business end-users who may have been solving the problem manually.

Performance of this model:

- Accuracy – 69%
- Sensitivity – 74%
- Specificity – 50 %

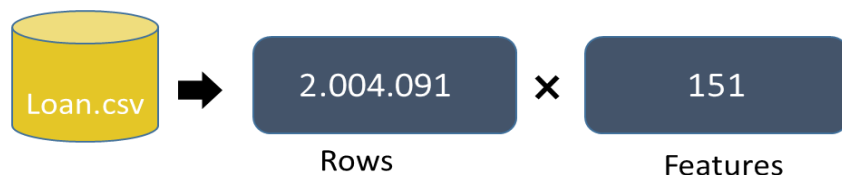
		Actual	
		1	0
Prediction	1	2097	4212
	0	2079	12180
Fully paid = 0			
Charged off = 1			

2.2 Dataset overview

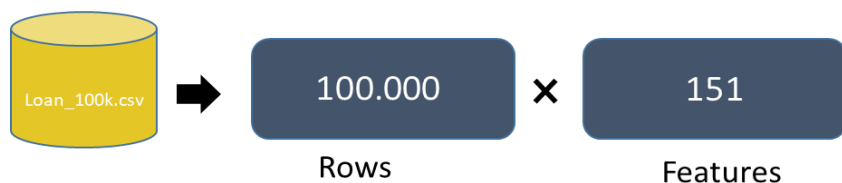
The original dataset provided by LendingClub consists of 20 million instances with little more than 150 features. Most of these features include information regarding customer's Credit history, Past, Current loans and its payment history along with personal information like number of active bank accounts, active credit lines, income, purpose of loan, interest rate of loan, any default history etc.. Based on the requirement of this project the target variable is identified as 'Loan Status'.

The original dataset is quite large. It was decided to select a representative sample from this original dataset consisting of about 100K instances. This number will be both representative of the sample and within the limits of computational power of the systems.

Original Dataset:



Sample Dataset:



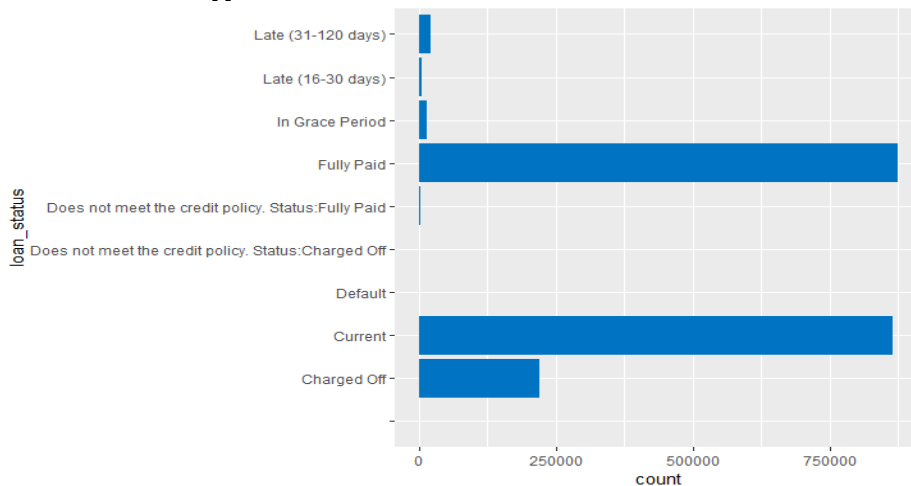
2.2 Dataset overview

We want to plot the frequency distribution of the target variable from the Original data to understand the distribution of data with respect to 'Loan Status'.

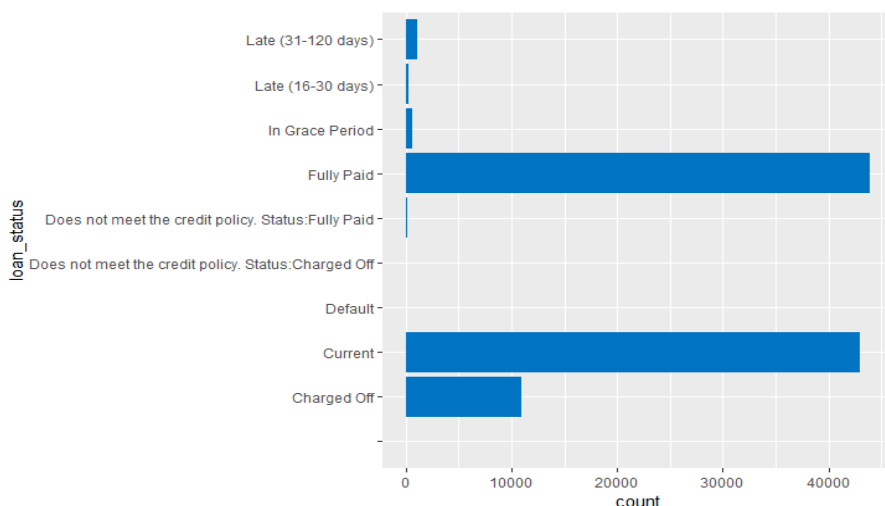
Below is the frequency distribution chart:

As we can see the sample is representative of the original dataset.

Original Dataset:



Sample Dataset:



2.3 Key Findings from the Dataset

From the frequency distribution we can infer following:

- About 43% of instances are current loans ('Loan Status = Current').
- About 43% of instances are Paid loans ('Loan Status = Fully Paid').
- About 12% of instances are default loans ('Loan Status = Charged Off').
- Target Variable ('Loan Status') has 9 different values.
- Data for some of the features are NOT available at the time of loan application (last installment paid, interest rate). These cannot be considered for analysis.
- Data for some of the features are available at the time of loan application (before approval or rejection). These features will be considered for the analysis.
- Sample data is representative of the original data.
- There seems to be different categories of 'Default' loans which can be grouped together.

Why ?

Statistical inference consists in the use of statistics to draw conclusions about some unknown aspect of a population based on a random sample from that population. exploratory data analysis (EDA) and summary statistics were also used to understand data.

2.4 Data Cleaning

2.4.1 Rows Reduction:

- Since the objective is to identify customers who will default a loan, this classification will not include loans already approved or in 'Current' status.
- Before Data Reduction

Loan Status	No. of Rows	Status
-	3	Removed
Charged off	10933	Considered
current	43033	Considered with outstanding balance less than 1\$
Default	1	Considered
Does not meet the credit policy. Status:Charged Off	37	Considered
Does not meet the credit policy. Status:Fully Paid	100	Considered
Fully paid	43863	Considered
In grace period	674	Removed
Late (16-30 days)	302	Removed
Late (31-120 days)	1054	Removed

- After Data Reduction

Loan Status	No. of Rows	New Loan Status
Charged off	10933	Charged off
Current	49	Fully paid
Default	1	Charged off
Does not meet the credit policy. Status:Charged Off	36	Charged off
Does not meet the credit policy. Status:Fully Paid	100	Fully paid
Fully paid	43863	Fully paid

2.4 Data Cleaning

2.4.2 Columns Reduction:

- Since the objective is to identify customers who will default a loan, this classification will not include features that are not available at the time of loan application.
- Out of 151 features available, 69 (46%) features were removed based on above conclusion.

Type of feature	Did we take it ?	Count of columns	Example
Credit History	Yes	68	No of open accounts, Total credit Balance
Current loan info	No	68	collection_recovery_fee, funded_amnt
Loan Information	Yes	7	loan_amnt, installment, int_rate
	No	1	Loan desc
Personal Informati	Yes	6	grade, emp_length
Target variable	Yes	1	Loan Status



Features we kept	82	54%
Features we removed	69	46%

- Also there were 3 categorical features with a lot of distinct values which is practically not feasible or not recommended to recode, so those 3 were removed, reducing dataset with 79 features.

emp_title	earliest_cr_line	addr_state
322476 Distinct Values	since this is not a time series problem, we would remove it	51 Distinct Values

- Features with 0 variance or same values were checked for, as these features doesn't contribute in the learning of the models. There were no such variable in the dataset.

2.4 Data Cleaning

2.4.3 Missing Values Analysis - Columns:

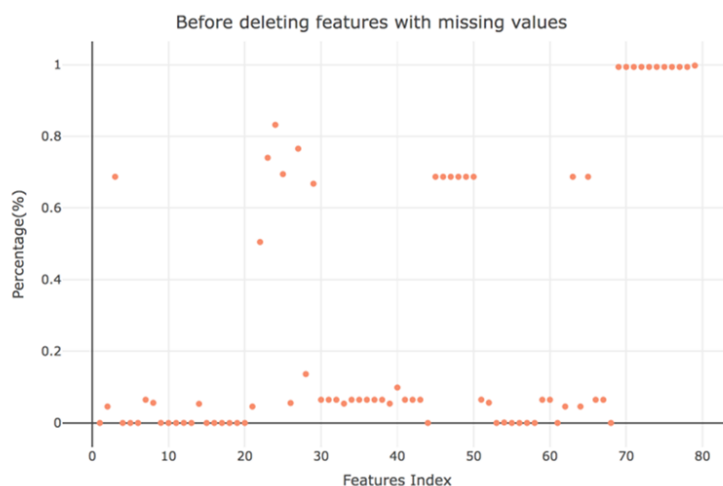
- While analyzing the features, some features were identified to be repeating based on if 'Application Type' was 'Individual' or 'Joint'. Features like Annual Income, Joint Annual Income, DTI, Joint DTI were decided to be merged. Merging features for Individual and Joint reduced missing values and added more meaning to the dataset.
- 2 more features were removed after merging these data.

Feature	% of missing values
dti_joint	67%
annual_inc_joint	69%

- Now features with more than 40% of total count with missing values were decided to be removed since they were considered not to provide much to the prediction of the model.
- After removing these features the dataset consist of 51 features, with only some of them having missing values less than 8%.

Why ?

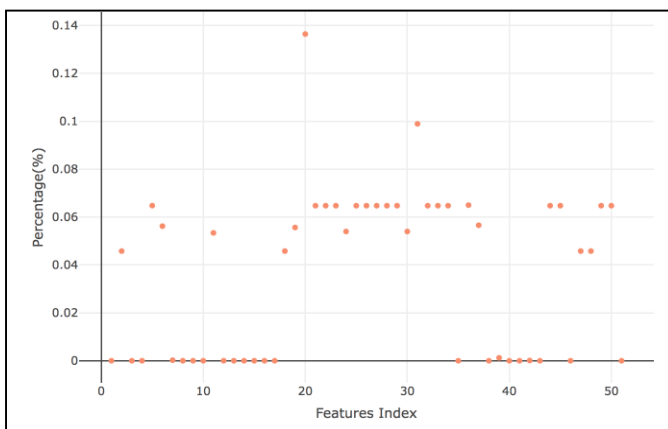
Normally most of the data will have some noise which is transferred from means of sources could be noise, incompleteness, missing fields. Identifying this enhances Accuracy, Integrity, Cleanliness, Correctness, Completeness, Consistency of data.



2.4 Data Cleaning

2.4.4 Missing Values Analysis - Rows:

- While analyzing the missing values row-wise a pattern was observed. Values from same features were missing for a set of rows.
- After confirming these rows were not inclined towards one specific target variable, but were similarly distributed among both class variables, they were removed.
- By doing this noise was reduced to much greater extent.



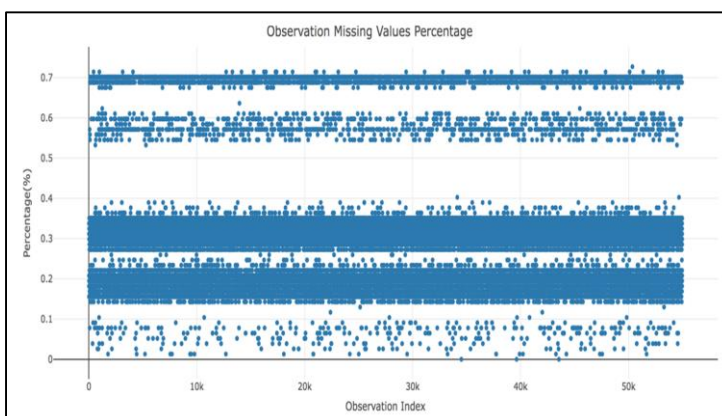
101,764

Missing Values

54,982

Rows

After removing rows
with missing values



11,241

Missing Values

51,422

Rows

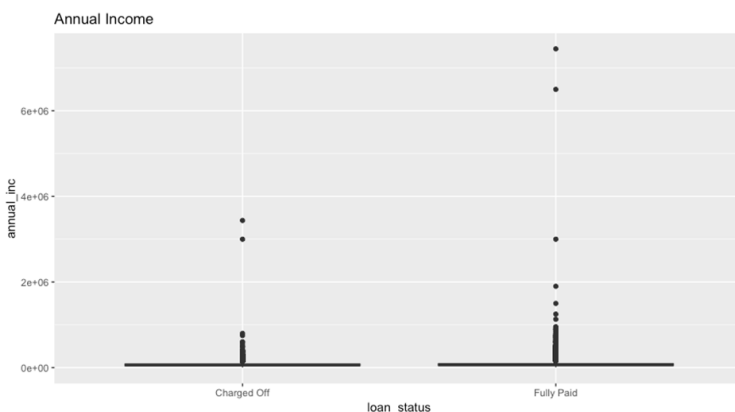
Reduced missing
values by 89%

Reduced rows
by 6%

2.4 Data Cleaning

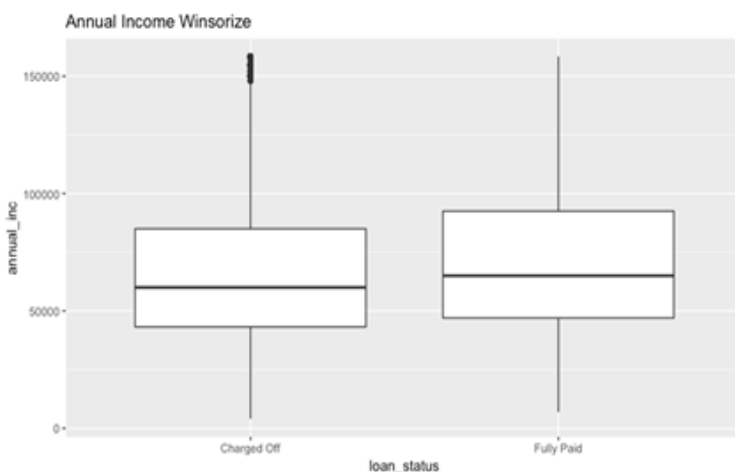
2.4.5 Outlier Analysis:

- A subset of dataset with only continuous features was created to check for outliers.
- Winsorizing technique was used to replace outliers.
- Winsorizing would replace data below the 5th percentile set to the 5th percentile, and data above the 95th percentile set to the 95th percentile.
- For e.g. feature Annual Income with following outliers was winsorized as follows:



```
> summary(df$annual_inc)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4080	46000	65000	76092	90000	7446395



```
summary(df.2$annual_inc)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4080	46000	65000	72384	90000	158404

2.4 Data Cleaning

2.4.6 One Hot Encoding:

- One hot encoding is a process by which categorical variables are converted into a different form readable by ML algorithms to do a better job in prediction.
- By performing “binarization” of each category the data is input to train the model. This removes unnecessary biasing due to misinterpretation of categorical features.
- New features are created due to this encoding, which will be included for further data analysis using complex algorithms.
- One hot encoding is better than integer coding since it eliminates the ordered relationship between integers. Ordered integers provide incorrect information to complex algorithms.

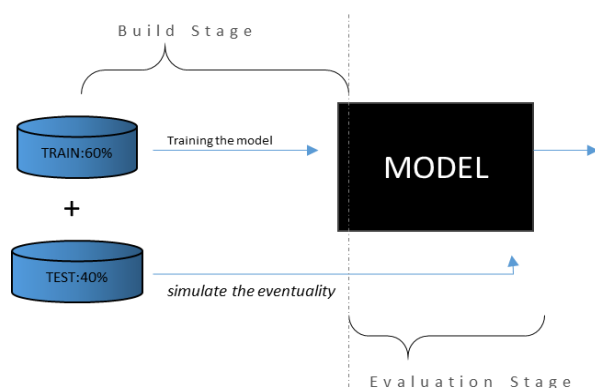
Why ?

In case of Supervised classifier one hot encoding is one of the basic approach to transform the categorical feature with out any change in the existing information, so the prediction is not influenced by their original values.

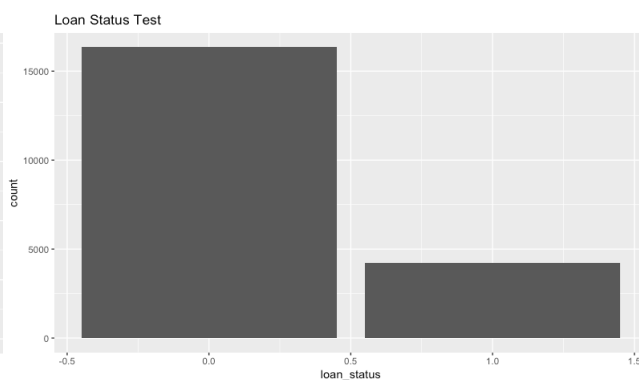
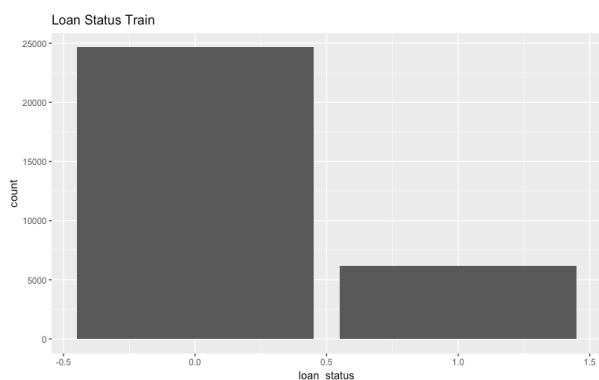
Evaluation Methods

3.1 Splitting Dataset

- As there is no thumb rule to define the ratio of splitting the dataset, it is always depended on the size of the data set. While making the split we have to make sure that there is enough data in the training set to train the model.
- In this case we have large data set. Hence a 60-40 split is possible.
- But to compare performance of models, it was decided to split the dataset into 60:40 and compare it with performance of 80:20 split.

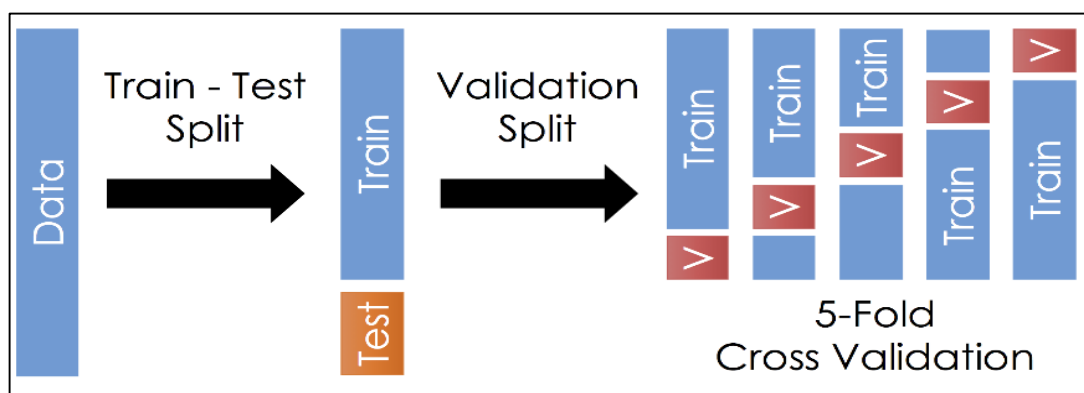


- Both 60:40 and 80:20 splits were done keeping in mind, proportion of target variable in both dataset is similar in order to avoid biased training of the model.



3.1 Splitting Dataset



- Cross validation in order to improve our results.
- K=10 folds were used.



3.2 Metric of Success

- Per the requirement of the project lowering the default rate i.e. no of people who default the loan is the objective.
- Hence intention was increase factor **Specificity**, which is a measure of true negatives per total

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP 
	Negative (0)	FN	TN 

- In order to get high specificity our model should be able to classify more true negative and less false positives.

3.3 Data Imputation

- To compare performance of models imputation of missing values was done using 2 different methods.
- Imputation with Mean of each feature:
Using mean did not drain the computational power of the systems. Also mean imputation can deal with data that are continuous, discrete, ordinal and categorical type of features.
- KNN imputation (KNN 4 & KNN 6)
KNN4, KNN6 imputation methods were used to compare against mean imputation method. This consumed more CPU time around 18 and 21 minutes respectively.
- Mean imputation method was much faster. Here although classification performance was almost same mean imputation took less CPU time and reduced variance of data.

Why ?

Missing data can introduce a substantial amount of bias and deletion of all instances with missing values may cause reduction in data. Imputation will help in filling the missing values in a sensible way which does not affect the real data observation.

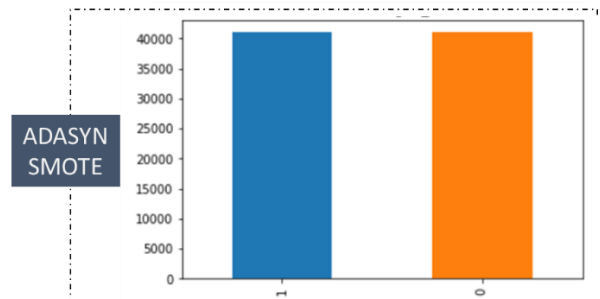
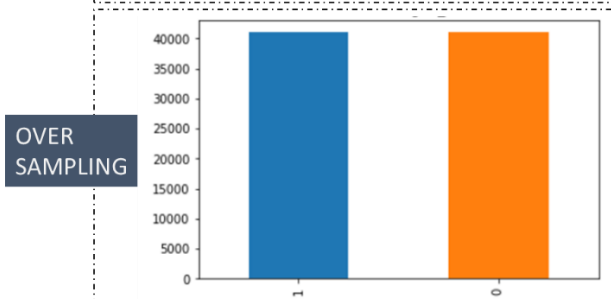
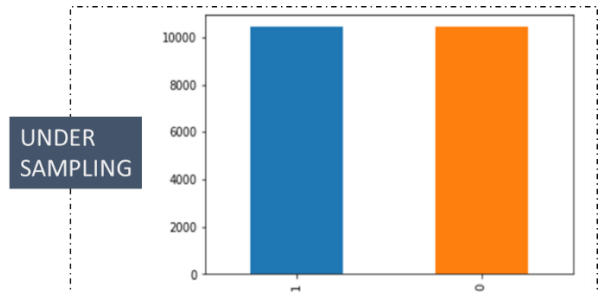
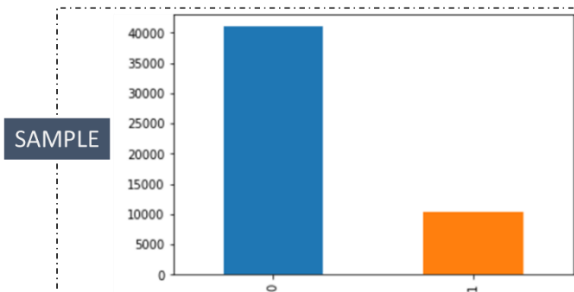
	Accuracy	Sensitivity	Specificity
df.knn4.mrmr_wt.6040	0,5995	0,5666	0,7259
df.knn6.mrmr_wt.6040	0,5944	0,5587	0,7347
df.train.mrmr_wt.8020	0,5917	0,5504	0,7519
df.knn4.mrmr_wt.6040 = not-scaled, KNN4 imputed, weighted class, 10 features, mRMR, 60:40 split df.knn6.mrmr_wt.6040 = not-scaled, KNN6 imputed, weighted class, 10 features, mRMR, 60:40 split df.train.mrmr_wt.8020 = not-scaled, mean imputed, weighted class, 10 features, mRMR, 80:20 split			

3.4 Data Balancing

Why ?

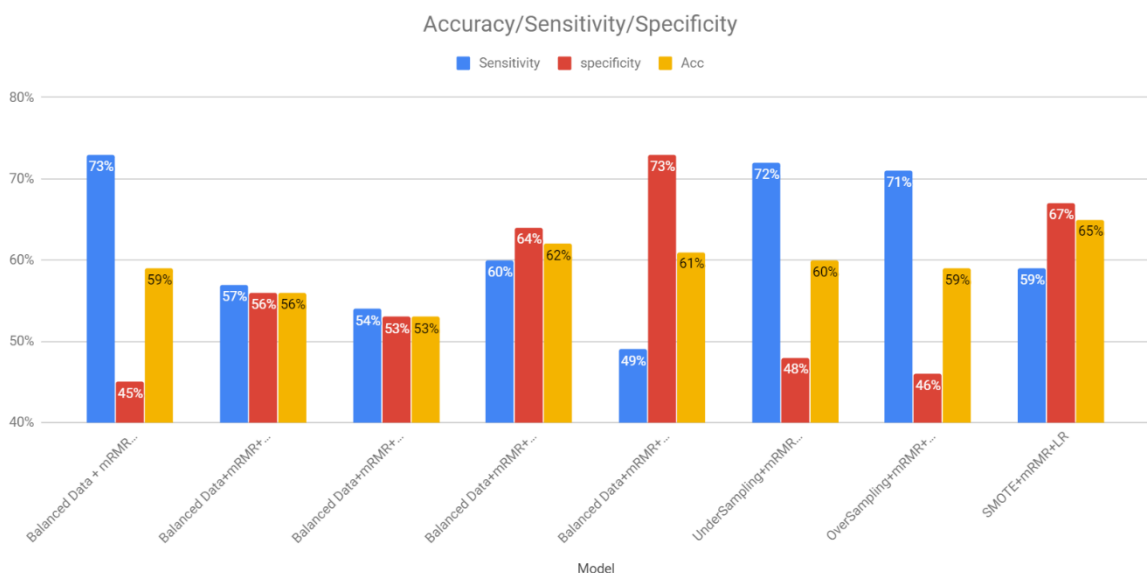
Most Machine learning classifiers fail to cope with imbalanced training datasets as they are sensitive to the proportions of the different classes. Hence, they tend to favor the class with the largest observations, which may lead to misleading accuracies.

- Machine learning algorithms tend to produce unsatisfactory or biased results due to imbalance in datasets.
- They tend to predict majority class data and will consider minority class data as noise. For this purpose it was decided to balance the data set.
- There are many ways and algorithms to balance the datasets. Since original dataset was huge, more minority class instances were extracted and added to sample dataset.
- Following are the comparison of different balancing techniques:



3.4 Data Balancing

- Following are the comparison of different performance outputs of different complex algorithms for testing data imbalance :



Modeling

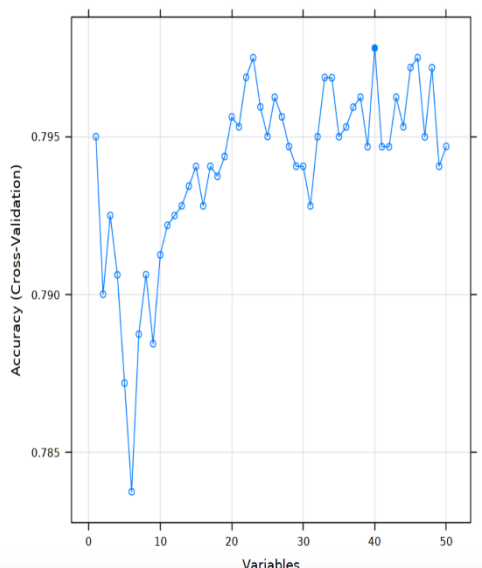
4.1 Feature Selection



Why ?

The filter use "relevance" of the features measured through univariate statistics. They are faster. The methods used are mathematically interpretable.

- For a classification model, feature selection can be done using 3 types of methods. Filter, Wrapper and Embedded.
- 1. Filter Method
- Since wrapper and embedded methods require a lot of computational power, filter methods were the first choice.
- In Filter method both univariate(Relief and Gain Ratio) and multi variate(mRMR) methods were used to compare the results between them.
- mRMR filter method gave best result as it deals with multicollinearity in itself.
- 2. Wrapper method – RFE (Recursive Feature Elimination)
- RFE algorithm recursively removes features, builds a model using remaining attributes.
- RFE is able to work out combination of attributes that contribute to the prediction on the target variable (or class).



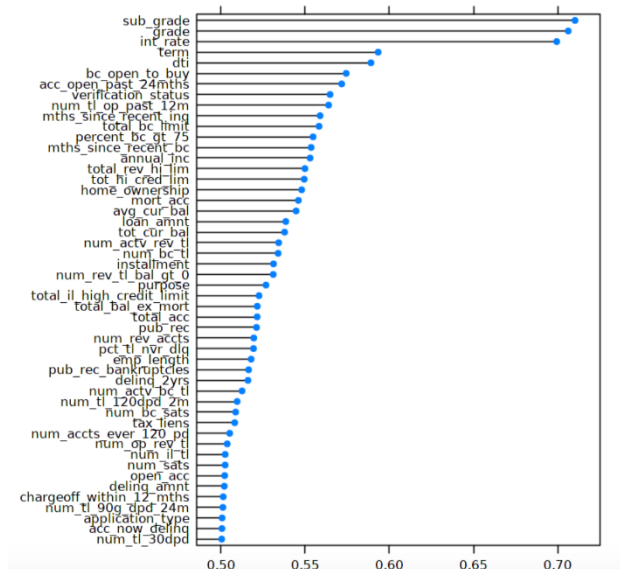
'sub_grade'	'int_rate'	'grade'	'term'	'dti'
'bc_open_to_buy'	'annual_inc'	'acc_open_past_24mths'		
'loan_amnt'	'tot_hi_cred_lim'	'total_bc_limit'	'num_op_rev_tl'	
'total_rev_hi_lim'	'num_actv_rev_tl'	'installment'	'num_rev_accts'	
'total_il_high_credit_limit'		'tot_cur_bal'	'avg_cur_bal'	
'num_bc_tl'	'total_acc'	'num_sats'		
'num_rev_tl_bal_gt_0'		'num_tl_op_past_12m'		
'open_acc'	'pub_rec'	'total_bal_ex_mort'		
'mort_acc'	'home_ownership'	'num_bc_sats'		
'mths_since_recent_bc'		'percent_bc_gt_75'		
'num_actv_bc_tl'	'num_il_tl'	'verification_status'		
'delinq_amnt'	'purpose'	'num_tl_120dpd_2m'		
'mths_since_recent_inq'		'tax_liens'		

4.1 Feature Selection

Why ?

Embedded and Wrapper methods choose the optimized subsets of features. both are better classifiers than filter methods. Learning capability of the model is the advantage.

- 2.Embedded method – LVQ (Learning Vector Quantization)
- LVQ is a for of neural network algorithm that uses a set of reference vectors to represent data points.
- For e.g. 10 data points/instances can be represented by 4 reference vectors, thereby classifying the instances.
- Since the reference vectors are consistently trained based on the data points (using Euclidean distance, quantization error), the resulting classification seems to be more accurate with respect to the input data.



ROC curve variable importance

only 20 most important variables shown (out of 50)

	Importance
sub_grade	0.7103
grade	0.7063
int_rate	0.6994
term	0.5934
dti	0.5891
bc_open_to_buy	0.5744
acc_open_past_24mths	0.5717
verification_status	0.5648
num_tl_op_past_12m	0.5639
mths_since_recent_inq	0.5589
total_bc_limit	0.5582
percent_bc_gt_75	0.5547
mths_since_recent_bc	0.5536
annual_inc	0.5529
total_rev_hi_lim	0.5499
tot_hi_cred_lim	0.5494
home_ownership	0.5479
mort_acc	0.5460
avg_cur_bal	0.5446
loan_amnt	0.5385

4.2 Class Weights

- The dataset being used for the classification contains 80% of 'Fully Paid' and 20% of 'Charged off' observations.
- Logistic Regression may bias the model towards 'Fully Paid' due to this.

4.3 Increasing Number of Features

- mRMR filter method selects features by removing multicollinearity and selecting features that contribute better to the target variables.
- Adding 10 more features to already 10 features selected by mRMR actually degraded specificity by 1%.
- This could be because mRMR seems to have selected best 10 features already and adding 10 more features involuntarily does not improve performance of mRMR algorithm.

	Accuracy	Sensitivity	Specifity
df.mean.mrmr_wt.6040.10	0,5861	0,5475	0,7352
df.mean.mrmr_wt.6040.20	0,5952	0,5621	0,7242
df.mean.mrmr_wt.6040.10-not-scaled, mean imputed, weighted class, 10 features, mRMR, 60:40 split			
df.mean.mrmr_wt.6040.20-not-scaled, mean imputed, weighted class, 20 features, mRMR, 60:40 split			

4.4 Threshold of Probability for LR

- Logistic Regression model by default classifies instances based on the probability of 0.5.
- If probability of an instance below 0.5 then its classified as Class1 and if its above its classified as Class2.

4.3 Increasing Number of Features

- This default threshold of 0.5 can be changed. By making above 0.4 as 'Charged off' and below 0.4 as 'Fully Paid' the results improved.
- The prediction of 'Charged off' customers increased from 75% to 90%.

LOGISTIC REGRESSION USING FEATURES FROM mRMR & SPLIT 80-20

mRMR

```
Confusion Matrix and Statistics
reference
data 0 1
0 7986 1889
1 190 219

Accuracy : 0.7978
95% CI : (0.7899, 0.8056)
No Information Rate : 0.795
P-Value [Acc > NIR] : 0.2435

Kappa : 0.1151
McNemar's Test P-Value : <2e-16

Sensitivity : 0.9768
Specificity : 0.1039
Pos Pred Value : 0.8087
Neg Pred Value : 0.5355
Prevalence : 0.7950
Detection Rate : 0.7765
Detection Prevalence : 0.9602
Balanced Accuracy : 0.5403

'Positive' Class : 0
```

mRMR & weight classes

```
Confusion Matrix and Statistics
reference
data 0 1
0 4500 523
1 3676 1585

Accuracy : 0.5917
95% CI : (0.5821, 0.6012)
No Information Rate : 0.795
P-Value [Acc > NIR] : 1

Kappa : 0.1944
McNemar's Test P-Value : <2e-16

Sensitivity : 0.5504
Specificity : 0.7519
Pos Pred Value : 0.8959
Neg Pred Value : 0.3013
Prevalence : 0.7950
Detection Rate : 0.4376
Detection Prevalence : 0.4884
Balanced Accuracy : 0.6511

'Positive' Class : 0
```

Fully paid = 0
Charged off = 1

CHANGE THE THRESHOLD TO IMPROVE THE PREDICTION

mRMR

```
Confusion Matrix and Statistics
reference
data 0 1
0 7698 1679
1 478 429

Accuracy : 0.7903
95% CI : (0.7823, 0.7981)
No Information Rate : 0.795
P-Value [Acc > NIR] : 0.8865

Kappa : 0.1839
McNemar's Test P-Value : <2e-16

Sensitivity : 0.9415
Specificity : 0.2035
Pos Pred Value : 0.8209
Neg Pred Value : 0.4730
Prevalence : 0.7950
Detection Rate : 0.7485
Detection Prevalence : 0.9118
Balanced Accuracy : 0.5725

'Positive' Class : 0
```

mRMR & weight classes

```
Confusion Matrix and Statistics
reference
data 0 1
0 2738 203
1 5438 1905

Accuracy : 0.4515
95% CI : (0.4418, 0.4612)
No Information Rate : 0.795
P-Value [Acc > NIR] : 1

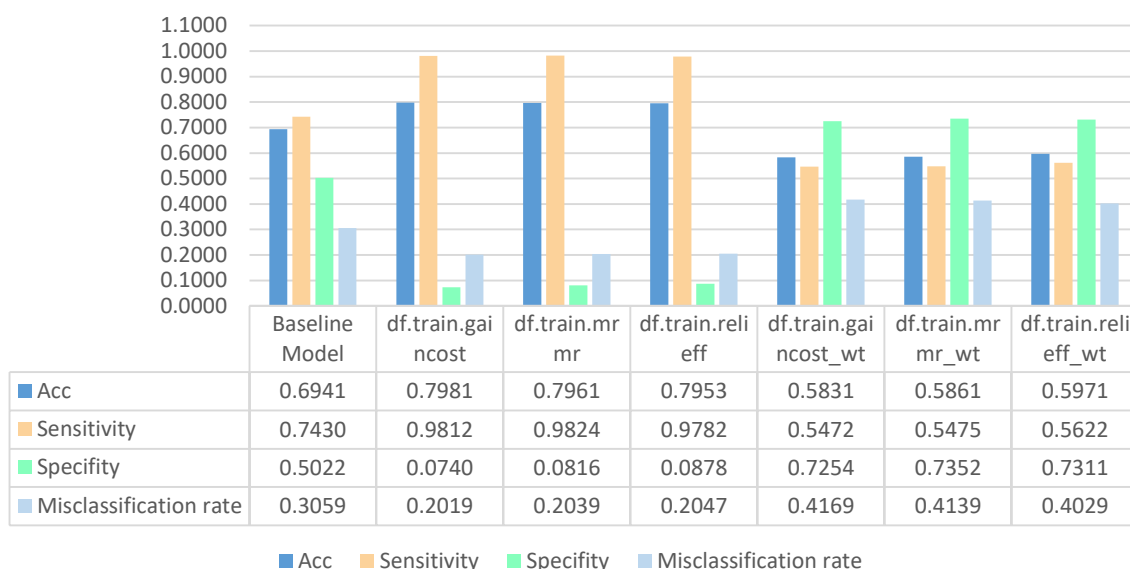
Kappa : 0.1242
McNemar's Test P-Value : <2e-16

Sensitivity : 0.3349
Specificity : 0.9037
Pos Pred Value : 0.3110
Neg Pred Value : 0.2594
Prevalence : 0.7950
Detection Rate : 0.2662
Detection Prevalence : 0.2860
Balanced Accuracy : 0.6193

'Positive' Class : 0
```

Fully paid = 0
Charged off = 1

4.5 Comparison Results of LR Model



Why ?

Hypotheses

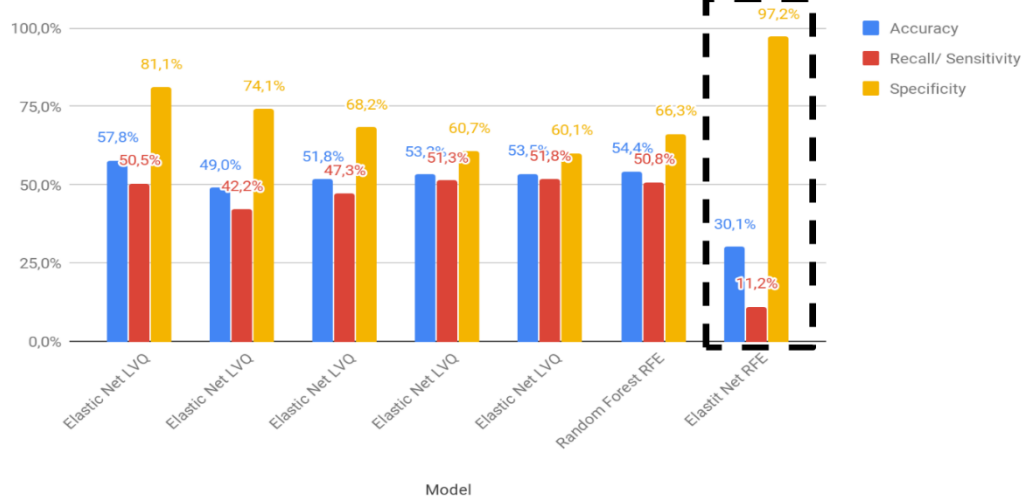
testing is an outcome of modern PDCA cycle (plan do act check). It is an idea made from limited evidence. It is a starting point for further investigation.

Multiple small experiments give an overall idea to validate the business ideas.

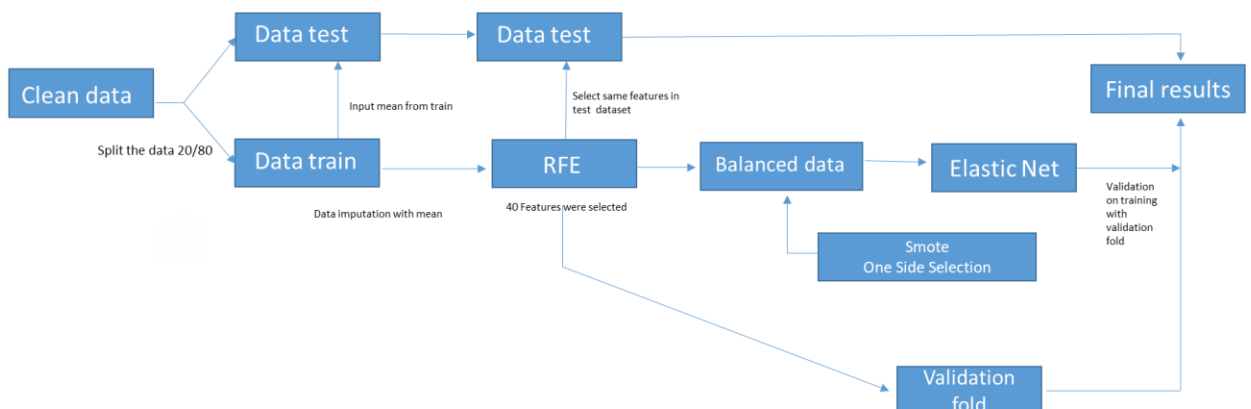
- A lot of hypothesis were verified using Logistic Regression model.
- Imputation methods were compared. (Mean vs KNN4 vs KNN6).
- Different split ratio was tested.(80:20 vs 60:40).
- Different feature selection methods were compared.(mRMR vs GainRatio vs ReliefF)
- Prediction outputs were compared with and without class weights.
- After comapring all the results and plotting a graph it was concluded, LR model performed better than baseline model.
- LR model using mRMR feature selection method, with class weights, 80:20 split ratio, classification probability threshold set at 0.4 performed best.

4.6 Results on – Elastic Net

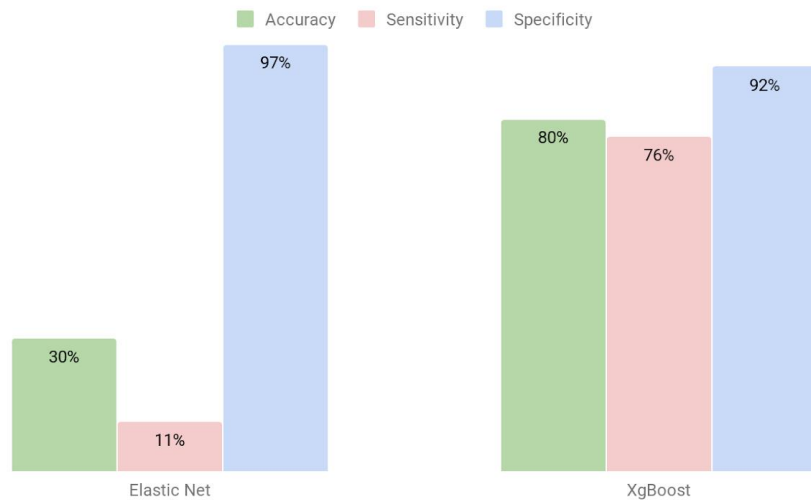
Accuracy, Recall/ Sensitivity y Specificity



- The elastic net performs simultaneous regularization and variable selection.
- It is able to select groups of correlated variables.
- LVQ and RFE feature selection methods were applied to Elastic Net along with different data balancing techniques.
- The results were compared and Elastic Net using RFE was concluded to perform with better outputs.



4.7 Results on – XGBoost



- Extreme Gradient Boost is a high performance tree model.
- Parallel computation is a reason for quicker response.
- Regularization and learning method allows dataset with missing values.
- Using XGBoost not only performed prediction of 'Charged off' customers but also performed better predicting 'Fully Paid' customers which was absent so far.

4.8 Cost of Misclassification



Why ?

In some contexts, certain errors are costlier than others. For example, it may be costlier to classify a high-risk credit applicant as low risk (False Positive) than to classify a low-risk applicant as high risk (False negative). Misclassification costs allows us to specify the relative importance of prediction errors.

- What is Misclassification ?
A data prediction model sometimes can make incorrect predictions. For e.g. A customer who would have 'Fully Paid' may be classified as 'Charged off' customer. Similarly misclassification can happen the other way.
- Cost of Misclassification:
How does misclassification impact Business. Rejecting a 'good' customer or approving a 'bad' customer can result in loss of revenue and profit for the Business. This is why all 3 outputs of a model should be looked at in general. (Specificity, Sensitivity, Accuracy).
- Calculate Cost of Misclassification
Cost of Misclassification was calculated for both complex models using an assumption of 5 units and 50 units loss for every misclassification of a 'good' and 'bad' customer. Calculation was done for both complex models, Elastic Net and xGBoost.

Unitary Cost Misclassification Paid

*Lost revenue as an
opportunity cost*

5\$

Unitary Cost Misclassification Default

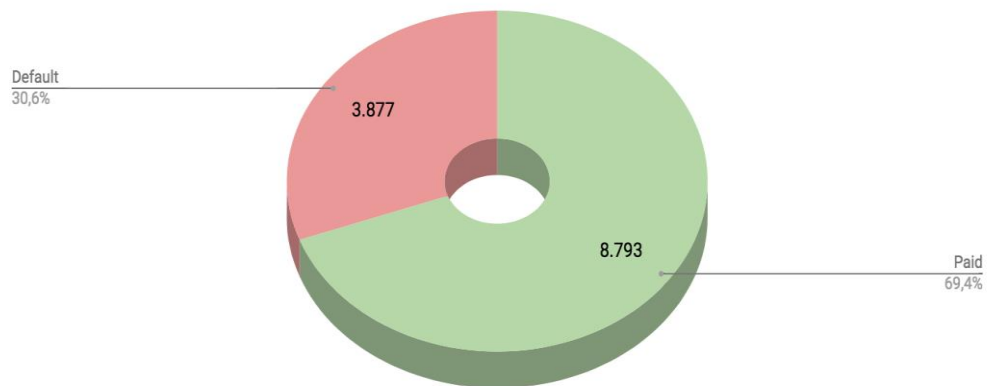
*Cost related to pre-collection (10\$)
Cost related to the recovery (10\$)
Cost related to customer's trust (30\$)*

50\$

4.8 Cost of Misclassification

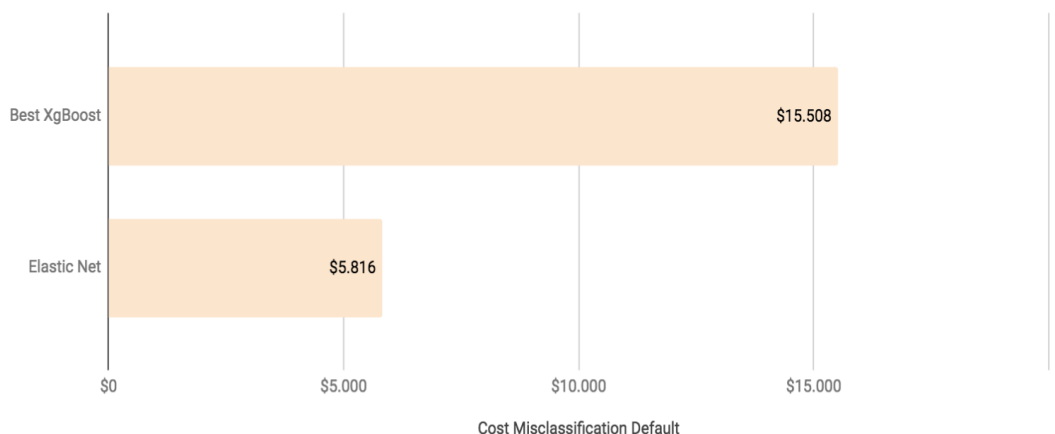
- For calculation purpose a forecast of January 2019 was assumed based on the history data.

January 2019 (estimation)



- Following were comparison of results from Elastic Net and xGBoost for the misclassification of 'Charged Off' customers. These are customers who defaulted but were predicted as 'good' during the approval of loans.
- Elastic Net misclassification = 310 loans $\rightarrow 310 \times 50 = \$15,500$.
- xGBoost misclassification = 116 loans $\rightarrow 116 \times 50 = \$5,800$.

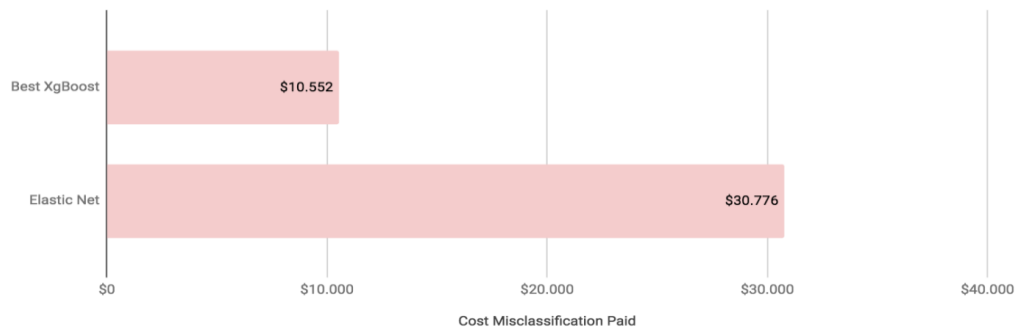
Cost Misclassification Default



4.8 Cost of Misclassification

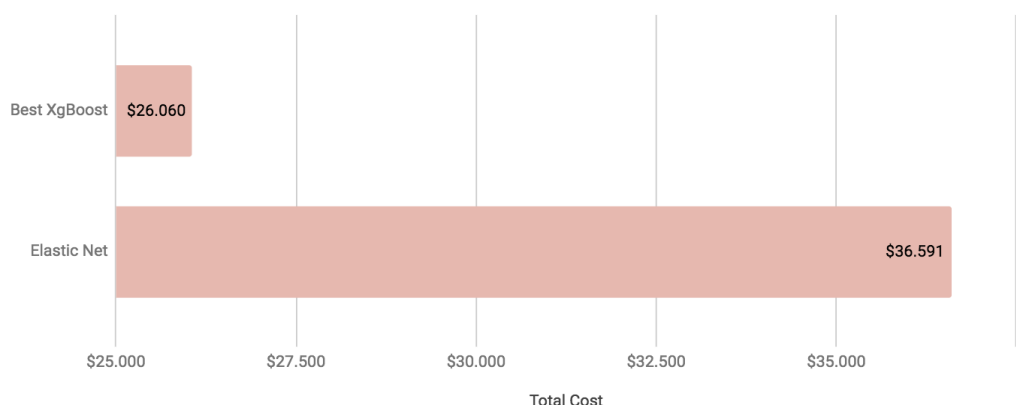
- Following were comparison of results from Elastic Net and xGBoost for the misclassification of 'Fully Paid' customers. These are customers who would have paid the loan but were rejected during application period.
- Elastic Net misclassification = 6155 loans
-> $6155 * 5 = \$30770$.
- xGboost misclassification = 2110 loans
-> $2110 * 5 = \$10550$.

Cost Misclassification Paid



- Adding both misclassification costs from above:
- Total Elastic Net misclassification =
(15500 + 30770) = \$46270
- Total xGboost misclassification =
(5800 + 10550 = \$16350.

Total Cost



Results

5.1 Conclusion

Model	Acc	Sensitivity	Specifity	Computation time	Additional Hardware
Baseline Model	0.69	0.74	0.50	< 5 min.	No
df.train.mmr_wt	0.58	0.54	0.73	Upto 10 min.	No
Elastic Net.RFE	0.30	0.11	0.97	Upto 20 min	No
XGBoost	0.80	0.70	0.92	Upto 20 min	No

- XGBoost produces better results in terms of accuracy, Specificity, Sensitivity.
- By predicting better sensitivity this model is able to reduce cost of misclassification also.
- Although there is a increase in computational time, the results are better and current hardware should support the new model.

5.2 Business Recommendations

- XGBoost model is able to predict 'Charged off' customers better than current model. (Objective)
- Also, the prediction of 'Fully Paid' customers has also improved. (Value addition)
- By increasing prediction of both 'Charged off' and 'Fully Paid' customers, the cost of operations (sending reminders, losing faith in investors, spending on collection agencies, making incorrect approvals) will be considerably brought down.