**Predictive model for credit default via Logistic Regression and Random Forest**

**Sean Stanislaw**

**Submitted in partial fulfilment of the requirements for**

**Professional Certificate in Data Science**



**Abstract**

The goal of credit scoring models is to predict the creditworthiness of a customer and determine whether they will be able to meet a given financial obligation or default on it. Such models allow a financial institution to minimize the risk of loss by setting decision rules regarding which customers receive loan and credit card approvals.

In this capstone project we use Logistic regression and Random Forest to predict the creditworthiness of bank customers using predictors related to their personal status and financial history. Model adequacy and robustness checks are performed to ensure that the model is being properly fitted and interpreted

Dataset - The source of dataset is Kaggle Credit risk modelling – Case Study

The dataset consists of the following fields:

* Loan Status: A categorical variable indicating if the loan was paid back or defaulted.
* Loan Amount: This is the loan amount borrowed
* Home Ownership: Categorical variable indicating home ownership. Values are "Rent", "Home Mortgage", and "Own". If the value is OWN, then the customer is a home owner with no mortgage.
* Person Income: Annual income of the applicant.
* Person employment length – For how long an individual has been at current Job.
* Loan Grade: Credit Score assigned to the application based on Internal credit scoring model.
* Loan Interest rate: Interest rate given by the system to that application.
* Loan percent Income: ratio of Loan applied to Income of the individual.
* Default on credit file: Weather the individual has any default listed on credit file.
* Length of credit file: The years since the first entry in the customer’s credit history
* Age of the Applicant.

Research Methodology.

Proposed Statistical tools: Logistic regression and Random Forest.

Data cleaning, data exploration and visualization:

1. All missing Values to be dropped from the data set row wise
2. Cross tabulations will be done on Loan status to identify the number of samples that defaulted to the Number of samples that did not default.
3. Cross tabulations will be done on Grades assigned through Internal credit scoring to ascertain proportion of default for every grade.
4. Cross tabulations will be done on home status to ascertain how their living situation i.e. renting, owner or mortgagee had a higher rate of default
5. The data set will be split in 75% Training and 25% testing.

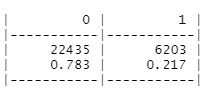
Logistic Regression.

* 3 Cut off levels will be used to ascertain best performing models 20%, 35% and 50%.
* We will use best AUC (area under curve) to ascertain the best cut-off, along with confusion matrix which basically tabulates each combination of prediction and actual value
* We all test the model for Sensitivity and specificity
* Finally, we will derive the overfall best fit score and compare all 3 to arrive at the best possible cut-off for out model

**Data exploration**

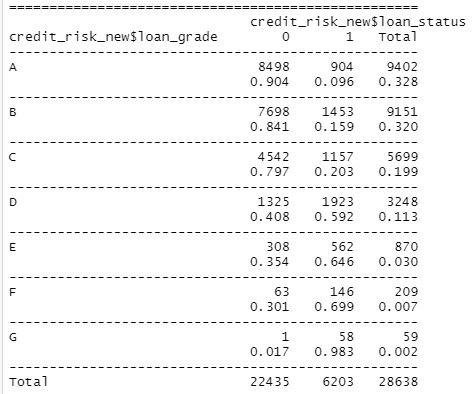
Our Dataset has 28638 observations of 11 variables after dropping NA variables

1. Loan Status
2. Did not default
3. Defaulted



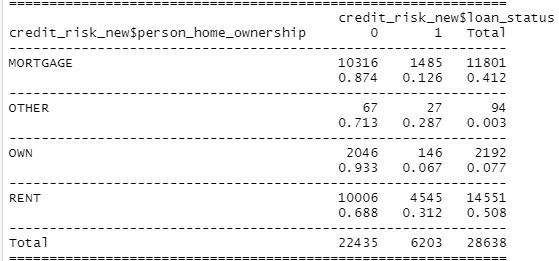
From the above table we can infer 78.3% of the applicants did not default on there loan while 21.7% applicant defaulted

1. Risk Grading



From the above table we can infer that garde A,B.C had the lowest default rate and there is a significant jump from 20.3% default rate for grade C to 59.2% for Grade D

1. Home Ownership status



From the above table we can infer Mortgage and Home owner were lower risk category with 87.4% and 93.3% not defaulting on loan.

Introduction - Logistic Regression

Logistic regression is one of the most important models for categorical response data. It is an example of a generalized linear model whose main use is to estimate the probability that a binary response occurs based on a number of predictor variables. Logistic regression is used in a wide variety of applications including biomedical studies, social science research, marketing as well as financial applications. One example of the latter is the use of binary logistic regression models for credit-scoring, that is: modelling the probability that a customer is creditworthy (i.e. able to meet a financial obligation in a timely manner) using a number of predictors.

We fit a binary logistic regression to the data, using the GLM function on the training data

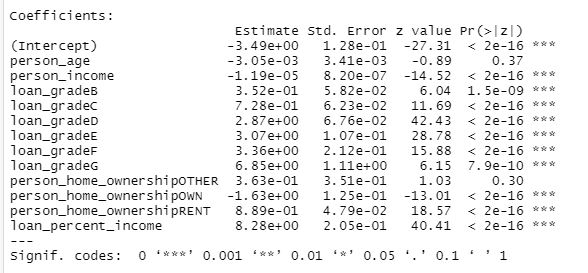
The linear predictor of the model we are fitting is given by:

Glm (formula = loan\_status ~ person\_age + person\_income + loan\_grade +

person\_home\_ownership + loan\_percent\_income, family = "binomial",

data = train)

Next, we assess which variables explain credit worthiness



Variables with \*\*\* is significant at .001 level of significance and explains credit worthiness of an applicant

To Predict outcome for test data we use the predict function in R

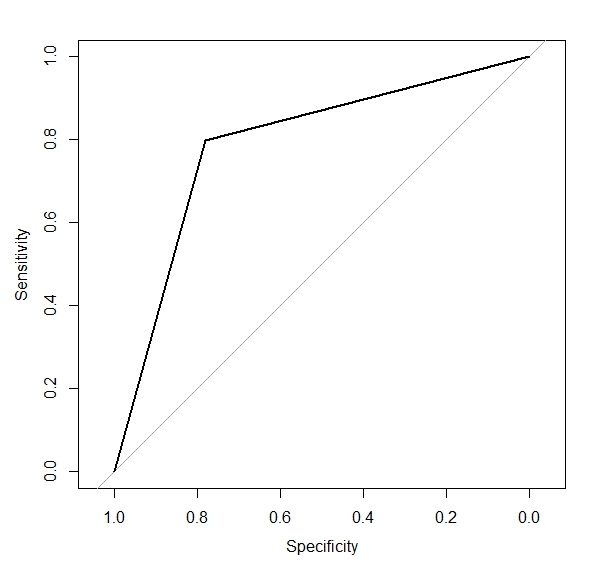
pred\_log\_regression\_model <- predict(logit\_model\_multi, newdata = test, type = "response")

Further we check the model for different cut off level 20%, 35% and 50% and plotting their receptive ROC curve

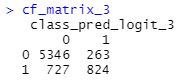
Model 1 Cut off 50%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cut off** | **AUC** | **sensitivity** | **Specificity** | **Mean Model accuracy** |
| **50%** | 0.742 | 0.88 | 0.758 | 0.862 |

**ROC curve** (**receiver operating characteristic curve**) for 50% Cut off



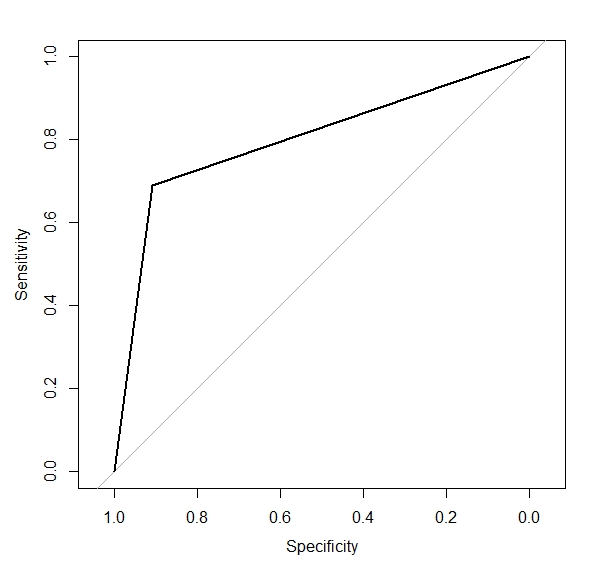
**Confusion Matrix plot for 50% Cut off**



Model 2- Cut off 35%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cut off** | **AUC** | **sensitivity** | **Specificity** | **Mean Model accuracy** |
| **35%** | 0.8 | 0.914 | 0.68 | 0.862 |

**ROC curve** (**receiver operating characteristic curve**) for 35% Cut off



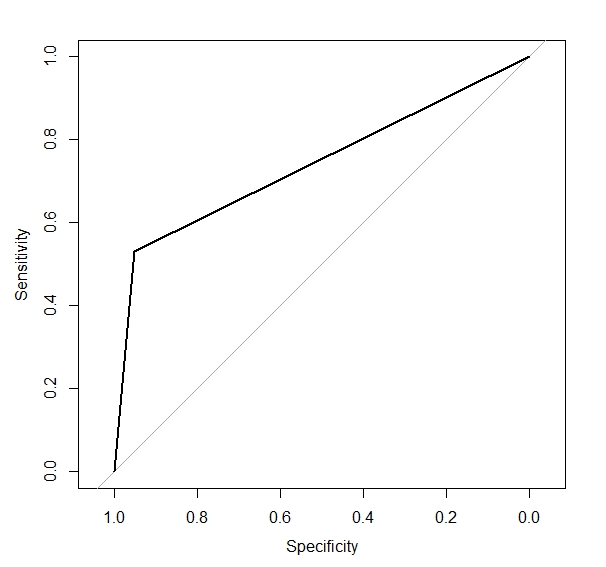
**Confusion Matrix plot for 35% Cut off**



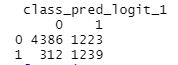
Model 3 -20% Cut off

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cut off** | **AUC** | **sensitivity** | **Specificity** | **Mean Model accuracy** |
| **20%** | 0.79 | 0.934 | 0.503 | 0.786 |

**ROC curve** (**receiver operating characteristic curve**) for 20% Cut off



**Confusion Matrix plot for 20% Cut off**

****

**Analysis Summary**

**Based on the 3 cut off points and various parameters we can conclude 35% performs the best**

**Has the highest AUC** 0.8 and gives Overall model performance score of 86.25 equivalent to 50% cut-off.

Confusion Matrix

For 50% cut off we can see it mis classifies 4.6% Credit worthy profile as defaulters

Whereas it misclassifies 46% Defaulters as credit worthy

For 35% % cut off we can see it mis classifies 8.96 % Credit worthy profile as defaulters

Whereas it misclassifies 31% Defaulters as credit worthy

**Thus, we accept 35% cut off as a better fitting model based on its AUC and Overall performance score along with significant reduction in mis classification of defaulters as credit worthy while only a small increase in Credit worthy being classified as defaulter**

**Random Forest – Introduction**

**Random forests** or **random decision forests** are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1)[[2]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1998-2) Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):587–588 Random forests generally outperform [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning), but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.[[4]](https://en.wikipedia.org/wiki/Random_forest#cite_note-:02-4)[[5]](https://en.wikipedia.org/wiki/Random_forest#cite_note-:0-5)

The first algorithm for random decision forests was created in 1995 by [Tin Kam Ho](https://en.wikipedia.org/wiki/Tin_Kam_Ho)[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1) using the [random subspace method](https://en.wikipedia.org/wiki/Random_subspace_method),[[2]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1998-2) which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.[[6]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg1990-6)[[7]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg1996-7)[[8]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg2000-8)

An extension of the algorithm was developed by [Leo Breiman](https://en.wikipedia.org/wiki/Leo_Breiman)[[9]](https://en.wikipedia.org/wiki/Random_forest#cite_note-breiman2001-9) and [Adele Cutler](https://en.wikipedia.org/wiki/Adele_Cutler),[[10]](https://en.wikipedia.org/wiki/Random_forest#cite_note-rpackage-10) who registered[[11]](https://en.wikipedia.org/wiki/Random_forest#cite_note-11) "Random Forests" as a [trademark](https://en.wikipedia.org/wiki/Trademark) in 2006 (as of 2019, owned by [Minitab, Inc.](https://en.wikipedia.org/wiki/Minitab)).[[12]](https://en.wikipedia.org/wiki/Random_forest#cite_note-12) The extension combines Breiman's "[bagging](https://en.wikipedia.org/wiki/Bootstrap_aggregating)" idea and random selection of features, introduced first by Ho[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1) and later independently by Amit and [Geman](https://en.wikipedia.org/wiki/Donald_Geman)[[13]](https://en.wikipedia.org/wiki/Random_forest#cite_note-amitgeman1997-13) in order to construct a collection of decision trees with controlled variance.

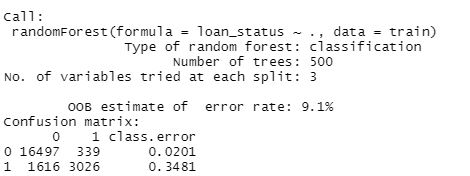
Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

Analysis of our Dataset

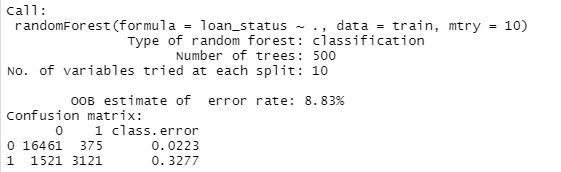
1. The data set will be split in 75% Training and 25% testing.
2. Multiple models be tested
3. Problem of Imbalanced dataset to be addressed using ROSE library all methods will be tested Over sampling, Under sampling, and both

The random forest package was initialised with correcting for any issues arising due to imbalance dataset

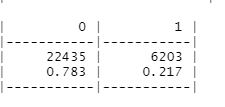
Model 1 – Number od variables tried for each split was 3



Model 2 – All 10 variables were tried at each split



**By analysing OOB estimate and confusion matrix we can ascertain there is probably an** An imbalanced classification problem Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class. The distribution of dataset for creditworthy and defaulters is given below which confirm the imbalance in data set with only 21.7% of defaulters included in data set.

****

**We will again estimate the Random Forest model and correcting the imbalance problem using ROSE package under all 3 methods Over sampling, Under sampling and Both,**

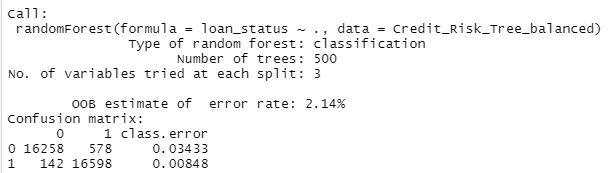
**Model 1- Oversampling**

**The following lines of code will be executed**

**# Solve imbalance problem in dataset using over sampling**

**Credit\_Risk\_Tree\_balanced <- ovun.sample(loan\_status~ ., data = train, p=0.5, seed=1,method="over")$data**

**Output:**

****

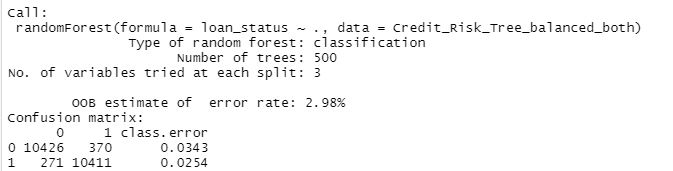
**Model -2 Method – Both (Over and under sampling)**

**The following lines of code will be executed**

**# Solve imbalance problem in dataset using both , over and under sampling**

**Credit\_Risk\_Tree\_balanced\_both <- ovun.sample(loan\_status~ ., data = train, p=0.5, seed=1,method="both")$data**

**Output:**

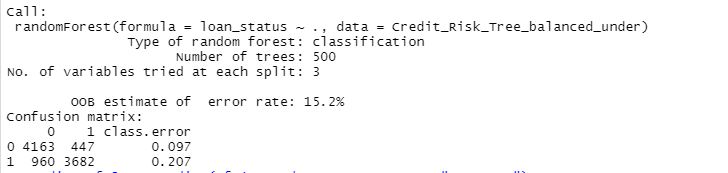
****

**Model 3- Method Under sampling**

**# Solve imbalance problem in dataset using under sampling**

**The following lines of code will be executed**

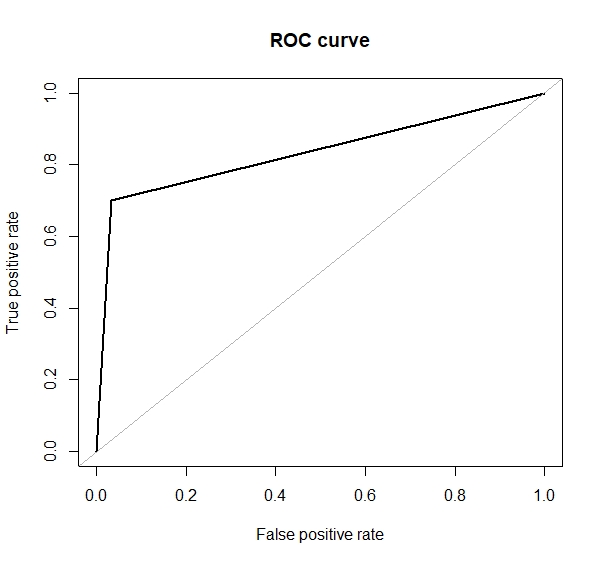
**Credit\_Risk\_Tree\_balanced\_under <- ovun.sample(loan\_status~ ., data = train, ,method="under",p=0.5, seed=1)$data**

****

**On observing the output, we can conclude that once we have corrected the dataset and transformed it to be balanced it OOB estimate error rate is significantly improved as well as we can observe from the confusion matrix lower mis classification error**

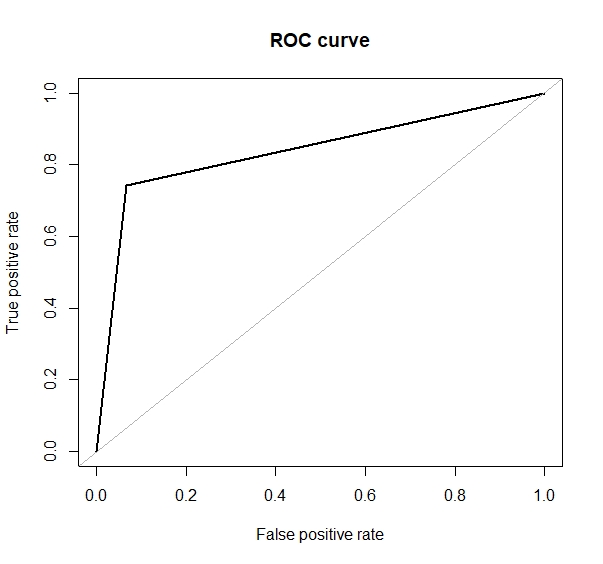
**Now we need to test model accuracy**

**Model 1- Oversampling - ROC curve plot**

****

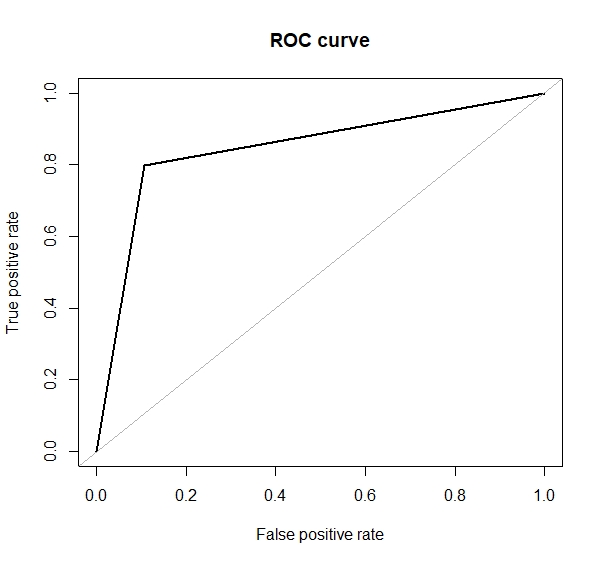
**Area under the curve (AUC): 0.834**

**Model -2 Method – Both (Over and under sampling) ROC curve plot**

****

**Area under the curve (AUC): 0.838**

**Model 3- Method Under sampling**

****

**Area under the curve (AUC): 0.845**

**Result Summary**

**As we can observe that OOB estimate of error is lowest with Oversampling Method and more or less all 3 models captures similar percentage of Area of under curve (AUC)- Wie will conclude Method 1 is the best fir model**

**References**

[**https://www.datacamp.com/**](https://www.datacamp.com/)

[**https://online.stanford.edu/courses/sohs-ystatslearning-statistical-learning**](https://online.stanford.edu/courses/sohs-ystatslearning-statistical-learning)

[**https://stackoverflow.com/**](https://stackoverflow.com/)

[**https://towardsdatascience.com/**](https://towardsdatascience.com/)

[**http://utstat.toronto.edu/~ali/papers/creditworthinessProject.pdf**](http://utstat.toronto.edu/~ali/papers/creditworthinessProject.pdf)

[**http://www.endmemo.com/r/ovun.sample.php**](http://www.endmemo.com/r/ovun.sample.php)

[**https://www.analyticsvidhya.com/**](https://www.analyticsvidhya.com/)

[**https://en.wikipedia.org/wiki/Random\_forest**](https://en.wikipedia.org/wiki/Random_forest)