**Applicant’s Credit Risk Scoring**

**Jigar Mehta, Preethi Bojja**

**Introduction:**

Risk analysis is familiar and most talked about process in any industry. In credit modules, loans pose higher risk to the lending institutions. Even though in the preliminary stage for granting loan a borrower might turn out as good customer but eventually because of various factors they might default. Predicting likelihood of a customer to default after granting loan is as important as predicting a risky customer prior to granting loan. This model helps business in taking prior steps to curb the risk level.

**Objective:**

To predict the likelihood of customers to Default or Active after the loan grant.

**Proposed Method:**

To classify the current customers into defaulting /non-defaulting customers based on the various machine

learning algorithms like probabilistic model, boosting algorithm, classification tree methods.

**Data:**

Categorical, continuous features of 42,535 rows and 111 columns with details after granting loan, in the time period of 2007 – 2011 from lending club <https://www.lendingclub.com/info/download-data.action>.

Tools**: R, Python**

**Hypothesis:**

* Loan amount, instalment amount, DTI (debt to income ratio), credit line may be highly influencing parameters.
* Over 70% of the features might be redundant.

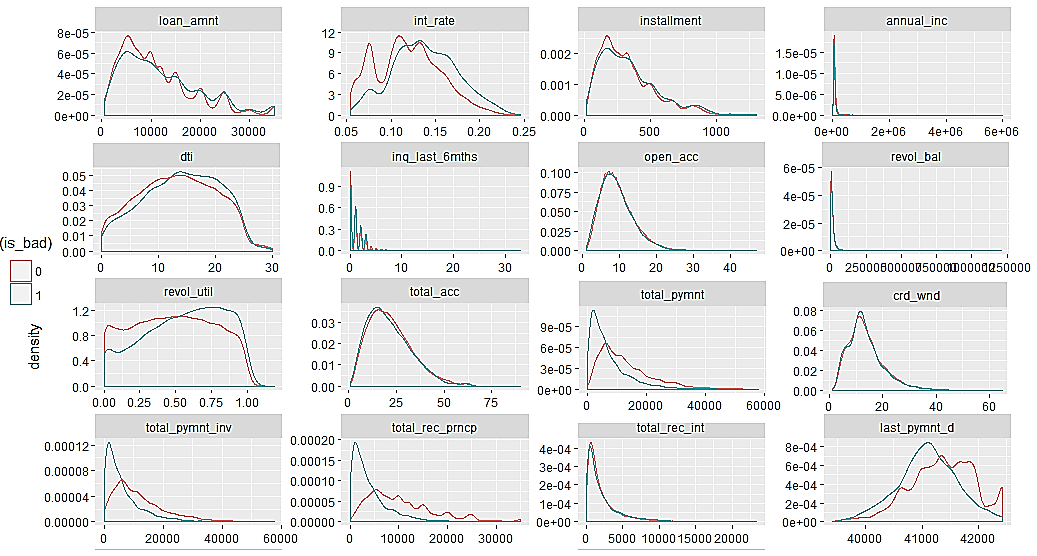
**Methodology:**

The below CRISP DM process flow was implemented in the project’s approach. Detailed explanation of each module is described below with illustrations.

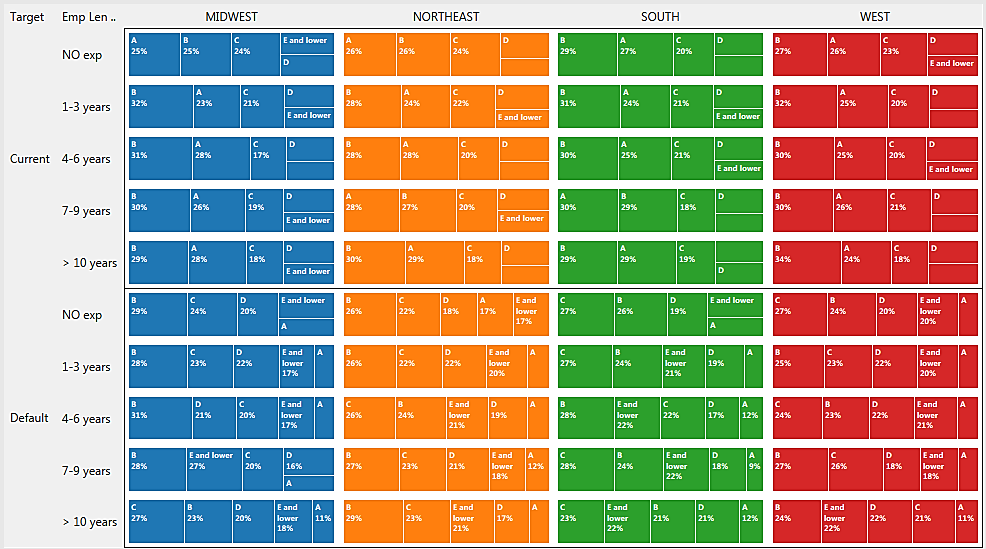
**Cross Industry Standard process for Data Mining (CRISP-DM)**

**Descriptive Analysis:**

This part of the process is required to understand the impact of the attributes on the target variable. Univariate and bivariate analysis was performed to understand the data and to formulate the ground for the hypothesis. Few results as below:



**Univariate Analysis of continuous variables to observe the impact on the target variable - Default (1), Current (0)**



**Behavior of customers with experience level vs access to grade of loan**

**Data Scrubing:**

The following steps were followed for cleaning the data as shown in the diagram:

1. Calculated the missing values in each of the 111 coulmns

* Removed the features with 60% & above missing data

1. Seperated continuous and categorical columns to impute the missing values

* Categorical columns were binned based on frequency value to fewer categories. For example,   
  employment length: No exp, 1-3 years, 4-6 years, 7-9 years, 10 years & above
* MICE package was used with 3 iterations to fill the values by Predictive Mean Matching (PMM) and Ployreg methods

1. Checked the presence of outliers in all the continuous features

* Replaced the outliers with median value of the feature in continuous data and mode/ high frequency value in categorical data

**Data Preparation:**

Data preparation / data modulation was the important step in the whole process. Each feature was important until and unless its statistical importance was not calculated. Feature selection helps predominantly here. The below methods were implemented in the feature selection:

**Feature Selection:**

**Variance inflation Factor Analysis (VIF):** This method finds the dependencies between more than two

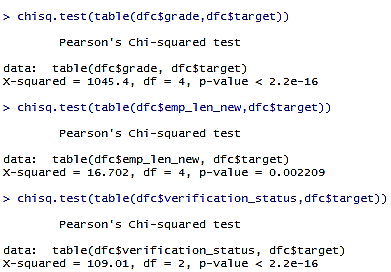
variables by making one feature dependent and rest independent. It calculates the value based on the correlation among these variables. Higher the VIF score is highly redundant the feature is. In this way, with VIF value of 5 and above and threshold redundant attributes are removed.

**VIFk1 =1/1- R k12 square value k1,k2,k3…kn = predictors**

|  |  |
| --- | --- |
| **Variables** | **VIF score** |
| int\_rate | 1.737808 |
| installment | 1.605507 |
| annual\_inc | 1.427994 |
| dti | 1.383475 |
| delinq\_2yrs | 1.087132 |
| inq\_last\_6mths | 1.158557 |
| open\_acc | 2.088079 |
| revol\_bal | 1.424421 |
| revol\_util | 1.598862 |
| total\_acc | 2.243173 |
| out\_prncp\_inv | 1.148736 |
| last\_pymnt\_d | 1.417981 |
| last\_pymnt\_amnt | 1.466801 |
| pub\_rec\_bankruptcies | 1.031818 |
| is\_bad | 1.162485 |
| crd\_wnd | 1.280837 |
| diff\_amt | 1.189071 |

**Variance Factor Analysis results with threshold of 5**

**Chisq Test:** To determine the correlation between the multiclass category variables, chisq test was used. Example results as shown below:



**Chisq Test Results on the categorical variables**

**Weight of Evidence (WOE):** This method calculates the information value of the attributes. This works better when there is high differene in the target class distribution as per the features.

**WOE = (f2-f1) x log (f2/f1)**

**Feature Transformation:**

Since all the models cannot be run on categorical data and to use uniform data in all of the models, categorical data was converted into numerical values based on the encoding techniques like **One hot encoding** and the **label encoder**.

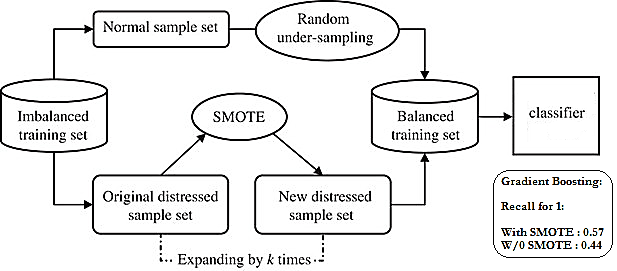
* One hot encoding technique assiigns values based on the various classes of categorical variables thus increasing the feature space. But, it is efficient in information retrieval. Label encoder worked fine inthis scenario after experimentation.

**Range Normalization:** After converting categorical data into numerical values, the dataset have numerical data in varying ranges. Values in all the attributes were normalized in the range of (0,1) for statistically correct computations.

All the above methods formulates the Analytics Base Table (ABT) i.e the data ready to be used for modeling and analysis with **21 features from initial 111 features**.

**Synthetic Minority Oversampling Technique (SMOTE):**

Unbalanced classification problems cause problems to many learning algorithms. These problems are characterized by the uneven proportion of cases that are available for each class of the problem. SMOTE is a well-known algorithm to fight this problem. The general idea of this method is to artificially generate new examples of the minority class using the nearest neighbors of these cases. Furthermore, the majority class examples are also under-sampled, leading to a more balanced dataset.



**SMOTE Process Flowchart with results**

The parameters percentage\_over and percentage\_under control the amount of over-sampling of the minority class and under-sampling of the majority classes, respectively. The parameter k controls the way the new examples are created. For each currently existing minority class example X new examples will be created (this is controlled by the parameter perc.over as mentioned above). These examples will be generated by using the information from the k nearest neighbors of each example of the minority class. The parameter k controls how many of these neighbors are used.

**Model Development:**

To observe the performance of various predicting methods following algorithms were implemented:

* **Logistic Regression** - binary logistic model is used to estimate the probability of a binary response based on one or more predictors. Its not a classification method but can be called as a[qualitative response/discrete choice model](https://en.wikipedia.org/wiki/Discrete_choice).
* **Stochastic Gradient Descent Algorithm** – is a [stochastic approximation](https://en.wikipedia.org/wiki/Stochastic_approximation) of the [gradient descent optimization](https://en.wikipedia.org/wiki/Gradient_descent_optimization) [method](https://en.wikipedia.org/wiki/Iterative_method) for minimizing an [objective function](https://en.wikipedia.org/wiki/Objective_function) that is written as a sum of differentiable functions.
* **Ada Boost Algorithm** – is short for "Adaptive [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm)) is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and [outliers](https://en.wikipedia.org/wiki/Outlier)
* **Random Forest** (Classification Tree) - [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 operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the classification or mean prediction of the individual trees.

The dataset is split into Training, Validation, Test on **60-20-20 ratio**. Above classifiers are run on the training data with various tuning parameters and validated their performance on the validation data set. The results proven to be satisfactory with considerable variance in the classification results of training and test datasets.

Important factors from the model to look out:

* **Relative parameter importance**

The below graph shows the relative predicting power or importance of each of the 21 predictors on the target variable with Last\_payment\_amount having high importance and Delinquency\_in\_2years being least significant.

**Relative Parameter Importance**

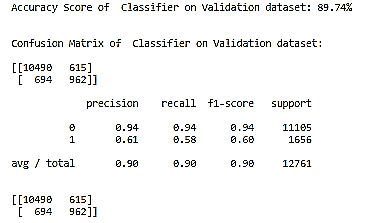
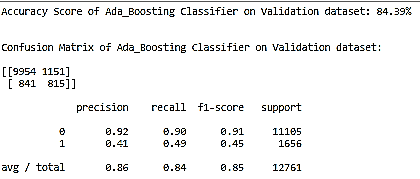
* **Optimum no. of estimators/ iterations based on the deviance analysis**

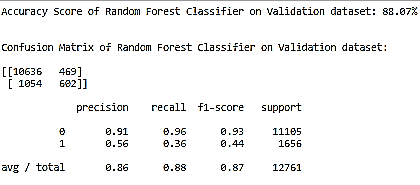
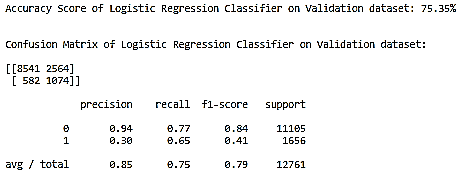
The below graph shows that 100 estimators/ iterations are sufficient enough for the better performance of the model with the error remaining constant after 100 iterations.

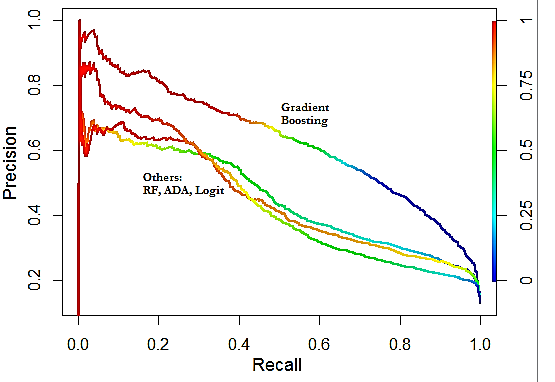
**Deviance plot showing the error variance of training and test dataset**

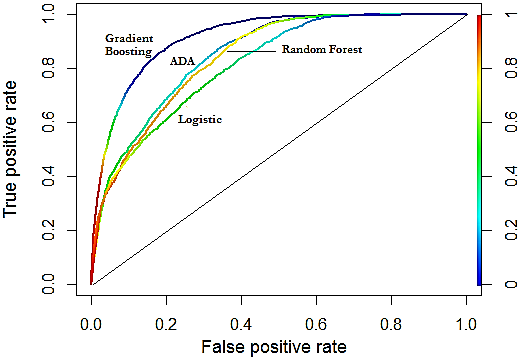
* **Precision, Recall values**

Confusion matrix of all the models are given below

 **Gradient Boosting Algorithm** **Ada Boost Algorithm**

 **Random Forest Classifier Logistic Regression**

* **Performance comparison of the models**
* The ROC curves of the algorithms shows relatively equivalent performance with Gradient Boosting leading over all other methods.
* The Precision vs Recall plot shows the accurate prediction by Gradient boosting algorithm



**ROC Curve Precision vs Recall plot**

* The below graph shows the performance comparison of the algorithms in predicting the default customers in terms of misclassification rate, Precision, recall, Area under curve (AUC) value.

**Performance comparison of Models**

**Conclusion:**

Boosting algorithm especially Stochastic Gradient Boosting Algorithm gave better performing results in

terms of predicting the 1’s i.e. Default customers followed by Ada Boost Algorithm, Logistic Regression, and Random Forest. The SGBA has higher propensity prediction in terms of less computation time, accuracy, and high precision and recall values.

**Code Structure:**

dff\_python.csv

**Data File and code:**

* LoadStats31.xlsx – original data file
* R\_code\_ML\_final project. R
* dff\_python.csv
* Machine Learning - Final Project. ipynb
* smo\_1.csv