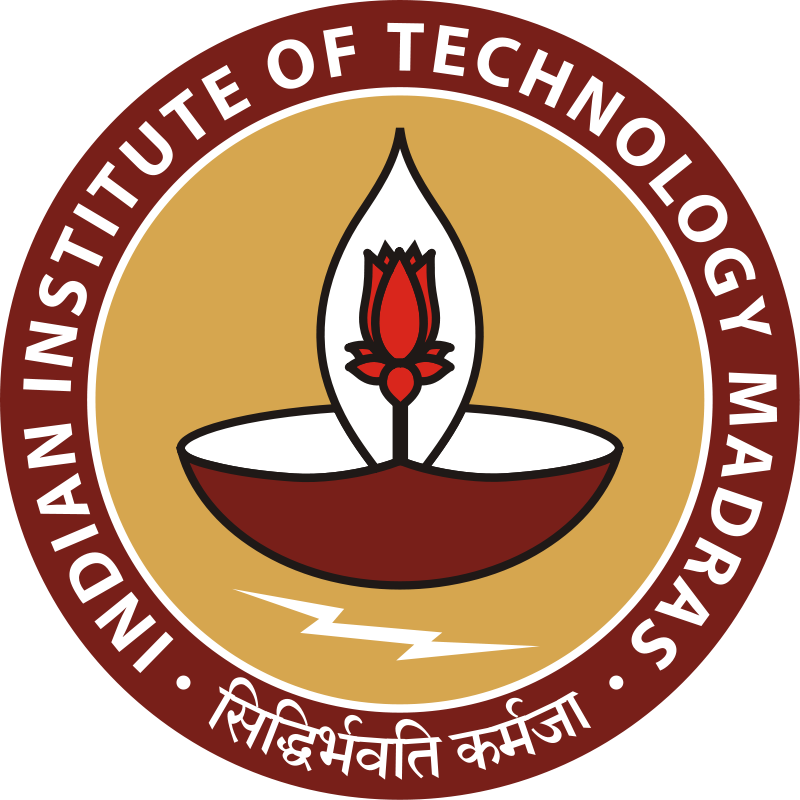


­­­­­­­­­



**MS6032: PREDICTIVE AND PRESCRIPTIVE DATA ANALYTICS**

**PROJECT REPORT**

Prepared By: Gerrit Mertz (MS19F004)

Akash Kumbhar (ED18S001)

Vedant Anil Diwan (MS18A007)

Santhosh Egappan (MS18A037)

Suruj S (MS18A048)

Date: December 4th, 2019

**­PROBLEM STATEMENT:** Prediction of the applicant going default in the next twelve months.

**OBJECTIVE**

To build a Machine Learning model for predicting whether the applicant will go a default or not. It can be viewed as a Binary Classification problem.

**ASSUMPTIONS**

1. A resident of the city can submit only a single application form.
2. None of the applications submitted are fraudulent.
3. Location ID of an applicant does not have an effect in this classification task.

**DATA PREPROCESSING**

The data had several missing entries which were labelled differently like “missing”, “na”, “N/A”. All the missing values were converted to the same format and imputed using three methods – simple mean, KNN and predictive mean matching. Out of the three techniques, simple mean imputation worked well during model building.

**METHODOLOGY**

**PART A: Significant Features using Principal Component Analysis (PCA)**

1. Column 48 was a location ID variable which is removed from the dataset because it has effect on the output variable.
2. Performing principal component analysis on the remaining 47 variables and selecting the features which have significant effect on the output.
3. PCA applied on input data (X) evaluates the Eigenvectors(47X47) and Eigenvalues (47X47) of the (X**T**X) matrix.
4. The graph below depicts the components of Eigenvector (47X1) corresponding to maximum Eigenvalue.
5. Components of Eigenvector will be giving us information about weightage associated with each variable.

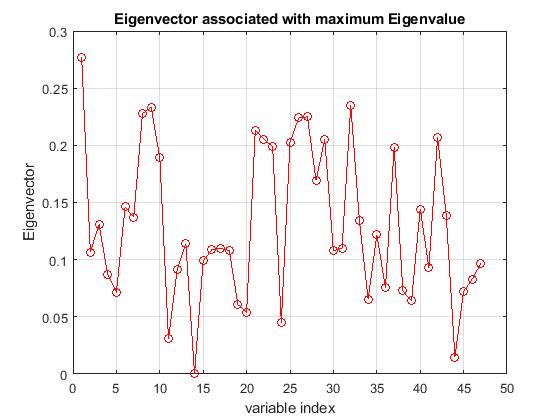


Figure 1: Distribution of coefficients associated with each variable (Generated in Matlab)

1. It can be inferred that the variable index associated with components of eigenvector less than 0.05 are variables having less effect on the output variable.
2. It was also cross checked to ensure that the accuracy of the model still remains hence we can safely neglect these variables.

**PART B: Model Selection**

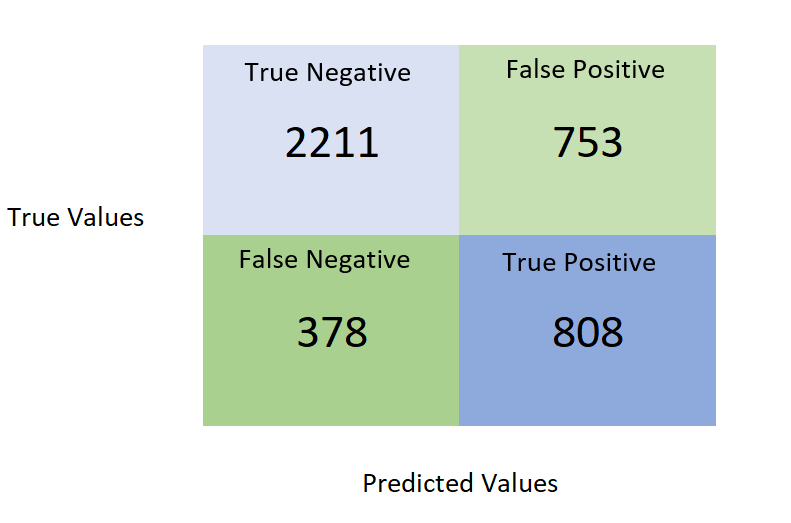
We built base version of the prediction model using several algorithms and compared them based on their training and text accuracies. The performance of the different models has been tabulated below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Data Splitting** | **Parameters** | **Training Accuracy** | **Test**  **Accuracy** |
| SVM | 80 : 20 | Kernel ‘ Rbf ’ | 71.35% | 70.86% |
| Ensemble of  Decision trees | 80 : 20 | Number of Iteration : 1000 | 76.31% | 75.91% |
| ANN | 80:20 | Hidden Layer : 4  Number of neurons in each layer : 10 | 72.06% | 71.71% |
| XGBoost | 80:20 | Learning rate : 0.15  Max. depth : 3  Min. child weight : 1 | 77.62% | 76.84% |
| Logistic Regression | 80:20 | Max iterations = 100 | 77.12% | 76.06% |
| Linear Discriminant Analysis | 80:20 | Max iterations=100 | 73.25% | 72.01% |
| LightGBM | 80:20 | Learning rate : 0.10  Max depth : 8  Min data in leaf : 5  No. of  leaves : 12  No. of iterations : 100 | 76.49% | 77.03% |

Considering the above test accuracy as the criterion for selecting the model, we selected the **LightGBM model** for further processing.

**PART C: Parameter Tuning and Optimization of Metrics**

1. The hyper parameters of the LightGBM model were tuned using gridsearch function in sklearn package of python. The optimal parameters were determined to be: learning rate - 0.25, max depth – 8, min dta in lead – 5, max bin – 100, num leaves – 12 and num iteration – 200.
2. To optimize the accuracy metrics namely overall accuracy, balanced accuracy and F1 score, we tuned the model as follows:
3. The test and train data split was set to 95:05 in comparison to 80:20 of the base model.
4. The threshold was optimized to 0.33 with a view of improving F1 score. (if the probability of the prediction was greater than 0.33, it was considered to be a default)



The above confusion matrix depicts the accuracies for the fine-tuned LightGBM model. In this case we obtained the ***best possible combination of all three accuracy metrics***:

Overall Accuracy = 72.75%

F1 Score = 58.83%

Balanced Accuracy = 71.36%