PROJECT REPORT

"Probability of Default" prediction using Machine Learning

Submitted towards the partial fulfillment of the criteria for award of Genpact Data Science Prodegree by Imarticus

Submitted By:

Sethumadhavan Aravindakshan

Course and Batch: DSP-23 - July 2020

Abstract

Evaluating the credit worthiness of an individual/counterparty before issuing a loan plays a crucial step in the loan underwriting process of any bank. This acts as an instrument to calculate the "Probability of Default" for any loan application and thus the loan issuer decides to issue or reject a particular request based on the acceptable level of Credit risk to avoid the firm from becoming lossy. Here, the proposed work ensures to build a model that accurately predicts the "Probability of default" using supervised machine learning algorithms.

Keywords

Machine learning, Feature Engineering, Modelling, Hyper-parameter Tuning, Train dataset, Test dataset, Model evaluation.

*Disclaimer: *Data shared by the customer is confidential and sensitive, it should not be used for any purposes apart from capstone project submission for DSP. The Name and demographic details of the enterprise is kept confidential as per their owners' request and binding.*

Acknowledgement

I would like to use this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I was fortunate to have **Mr.Vijay Kumar** as my mentor. He had readily shared his immense knowledge in data analytics and had guided me in a manner that the outcome resulted in enhancing my data skills.

I wish to thank, all the faculties(especially **Mr. Arun Kumar** and **Mr. Jagadeesan**), as this project utilized the knowledge gained from every course that formed the DSP program.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: July 20, 2020 Sethumadhavan Aravindakshan

Place: Chennai

Certificate of Completion

I hereby certify that the project titled **"Probability of Default" prediction using Machine Learning** was undertaken and completed under my supervision by Sethumadhavan Aravindakshan from the batch of DSP-23 (Jul 2020)

Mentor: Mr.Vijay Kumar

Date: July 20, 2020

Place – Chennai

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Title & Objective of the study

Title: **"Probability of Default" prediction using (ML) Machine Learning**

Objective: To build a data model for predicting the probability of default using machine learning classification algorithms.

1.2 Need of the Study

"Probability of Default" prediction plays a significant role in the loan underwriting process of the credit risk department of any financial institutions. It acts as a decision-making aid that assesses the credit history of the borrower/counterparty, thereby, enabling the financial institutions in advance, to either approve or disapprove any particular borrower's loan request. The issuers have a set of model/s and rule/s in place which take information regarding their current financial standing, previous credit history and some other variables as input and output a metric which gives a measure of the risk that the issuer will potentially take on issuing the loan.

1.3 Business or Enterprise under study

XYZ Corp. is a credit lending company and the main area of focus for this study is loan underwriting process. In general, whenever an individual/corporation applies for a loan from a bank (or any other loan issuer), their credit history undergoes rigorous checks and screenings to ensure if they are capable enough to repay the loan (in this industry it is referred to as credit-worthiness).

1.4 Data Sources

The dataset pulled in the study is a complete package containing details of all the loans issued by **XYZ Corp**. through 2007-2015, including the status of all the current loans, employment details, last payment information, credit scores, address details and many more, spanning over a large dataset characterized by **855,000 records and 73 Features**.

1.5 Tools & Techniques

Tools:

- Programming Language : **Python**
- IDE's : **Jupyter notebook, Kaggle notebook**
- Feature Engineering : **Pandas, Numpy**
- Data Visualization : **Matplotlib, Seaborn, Sweetviz**
- Machine Learning : **Sklearn, Xgboost**

Techniques:

Supervised Machine Learning algorithms and modelling for classification related problems.

1.6 Infrastructure Challenges

With the available system configurations:

- Increased time consumption while performing feature engineering on the data source under study was noticed.
- A fairly high model training and Hyper-parameter tuning time period was observed and thus a Kaggle - GPU kernel was leveraged wherever and whenever required to fasten up the process and to avoid any further resource contentions in the personal system.

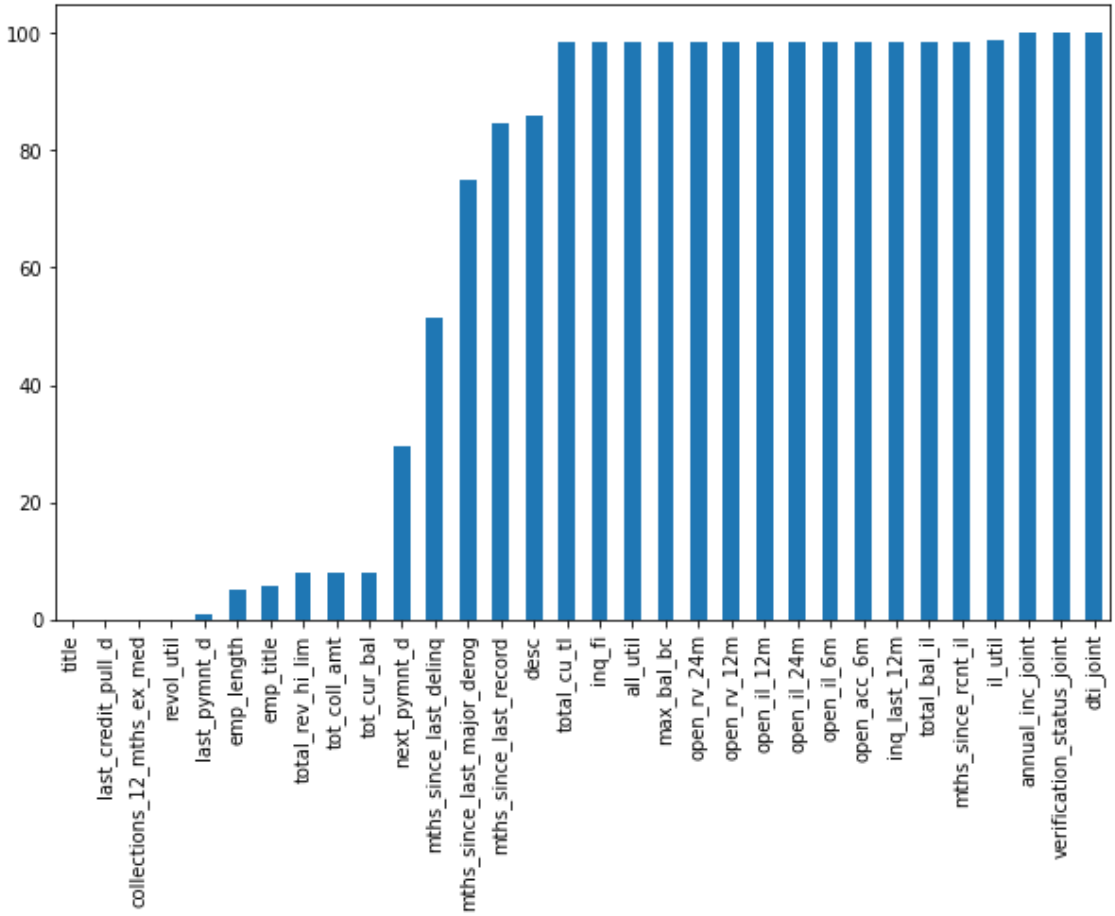
CHAPTER 2

DATA PREPARATION AND UNDERSTANDING

The efficiency of any modelling technique highly depends on the precise execution of the steps involved in it. This primarily requires outlining the phases / modules involved and developing a sound understanding of the same. Elaborated below are the descriptions of the afore-mentioned modules and the actions involved:

2.1 Data Extraction and Cleaning

2.1.1 Missing Value Analysis and Treatment – A module to estimate the missing data in percentage (%) is created, which returns the missing values in % along with a visualization as shown below:



- ✓ Features with the percentage of **voided data amounting to 84%** and more are dropped in the first iteration.
- ✓ In the second iteration, correlation between the remaining features having missing values and the target variable is determined and the feature which shows lesser correlation is also discarded to enable focusing on only the most prominent features.
- ✓ The next round of iteration that follows, involves creating a new category of data named **'UNKNOWN'** with features possessing a void data percentage in the range of **5-30 %** and this named category is further used in modelling.
- ✓ Lastly, complete rows are removed from the datasets characterized with features that had missing percentages **less than 1%** as it's **negligible and insignificant.**

2.1.2 Handling Outliers - The underlying objective of the project under study has been to analyze classification-based scenarios and hence the procedure of handling all the possible outliers is omitted.

2.2 Exploratory Data Analysis

'Sweetviz' is a powerful open source Python library that generates insightful, beautiful and high-density visualizations to kickstart EDA (Exploratory Data Analysis) with a single line of code. The corresponding output is a fully self-contained HTML application.
An analysis performed on the customer loan dataset using the library 'Sweetviz' usually provides all the correlations and data distributions of every individual feature consequently making it very convenient for analyzing and finalizing on further proceedings of the data preparation phase.

2.3 Feature Engineering

2.3.1 Feature Selection

After treating the Missing data, the data is divided into two categories and the feature selection varies for each category.

- **Numerical Features**

  BorutaPy method along with the **Random Forest Classifier** and **XgB Classifier algorithms** is used to determine the significant features and all the **inconsiderable features** are then **discarded** from the numerical features data frame.

- **Categorical Features**

  Chi-squared (χ^2) test is used to determine the significant features from the categorical features and again the inconsiderable features are expelled from the categorical features data frame to proceed to the encoding steps.

2.3.2 Feature Encoding

Sklearn models take numerical features either in **'int'** or **'float'** formats, and hence, encoding the acquired **categorical features** to convert it into numerical value notations is mandatory. Primarily, the nominal categorical features and ordinal categorical features are differentiated based on the data understanding. Secondarily, the below mentioned encoding approaches are followed on the differentiated branches:

- ✓ **For Nominal features**: One-Hot Encoding technique is performed
- ✓ **For Ordinal features**: Features are encoded manually based on the feature's order. Some features having higher number of categories are grouped together wherever possible, to limit the number of categories in a feature and still contribute to the target variable.

2.3.3 Data consolidation

In this phase, both the numerical features and categorical features (encoded version) are merged to form one final dataset and also to facilitate the model to consume for data-mining activities in the consecutive phases as well. Finally, the data is branched into **Train and Test datasets** based on the criteria mentioned in the problem statement.

CHAPTER 3

FITTING MODELS TO DATA

3.1 Model Comparison

The newly categorized **Train dataset** is fed into the below quoted classification algorithms followed by an extensive cross validation.

- **Logistic Regression**
- **Linear Discriminant Analysis**
- **Decision Trees**
- **Random Forest**
- **XGBoost**

Finally yet importantly, mean values of the following metrics are evaluated and the model that provides the best values for all these metrics is chosen.

- **Accuracy**
- **Precision**
- **Recall**

3.2 Fitting XGBoost classifier model to data

- Amongst the classification algorithms put under comparison, **XGBoost classifier** algorithm was found to provide better results under all the metrics that were considered of interest.
- Further, the train data was fitted into a basic XGBoost classifier model with default parameters and accuracy of the chosen model in predicting the target values for both train and test data was evaluated and assessed.
- The assessment of the XGBoost model with test data and training data showed a **fairly high efficiency** in target value prediction.
- Additionally, the possibility of an efficiency improvisation was checked with **Hyper-parameter** although the model built, portrayed a very optimal efficiency.

3.3 Performing Hyper-parameter Tuning to XGBoost model

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyper-parameter is a parameter whose value is used to control the learning process.

The optimal values for the below hyper-parameters using **Grid Search CV** on the XGBoost classifier model were determined and are given as follows:

- Max_depth : **5**
- Min_child_weight : **5**
- Gamma : **0.2**
- Subsample : **0.9**
- Colsample_bytree : **0.9**
- Reg_alpha : **0.01**
- Learning_rate : **0.1**

Next, a **new XGBoost** model was created using the afore-mentioned optimal values and was fitted to the train data. Further, a repetition of prediction efficiency evaluation was carried out in the way identical to the **XGBoost assessment**.

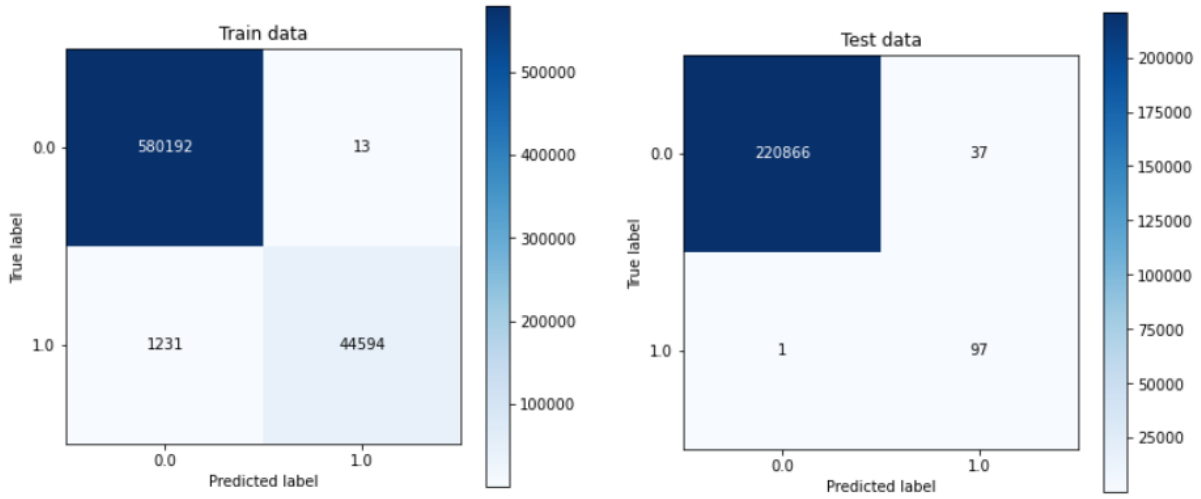
The comparison of metrics of the predictions **before and after hyper-parameter tuning**, showed that hyper-parameter tuning **did not introduce** any significant improvement to the performance of the basic XGBoost model.

3.4 Model Evaluation

Model evaluation is an integral part of the model development process. It helps to find the best model that represents data and also shows how accurately and effectively, the chosen model will work in the future as well. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate over-optimistic and over-fitted models and therefore, a test set (not seen by the model) is used as well to evaluate model performance.

Key evaluation methods incorporated to check for Train and Test data

Confusion Matrix



Classification Report

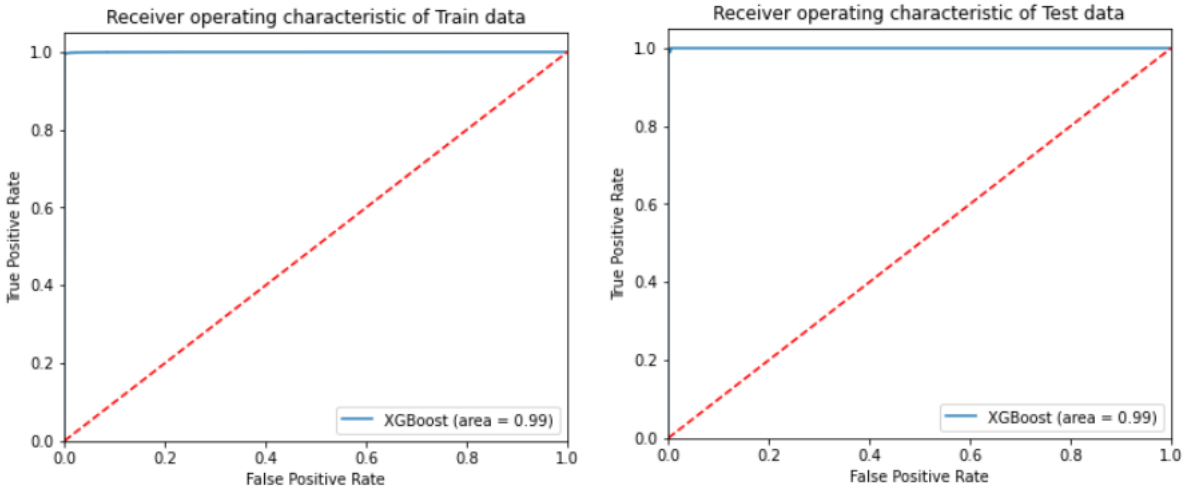
Classification report of Train data is :

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	580205
1.0	1.00	0.97	0.99	45825
accuracy			1.00	626030
macro avg	1.00	0.99	0.99	626030
weighted avg	1.00	1.00	1.00	626030

Classification report of Test data is :

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	220903
1.0	0.72	0.99	0.84	98
accuracy			1.00	221001
macro avg	0.86	0.99	0.92	221001
weighted avg	1.00	1.00	1.00	221001

ROC Curve



Summary of Accuracy, ROC Value, F1-Score and Log_loss

Models	Accuracy	Roc_values	F1-score	Log_Loss
XG_Boost_Train	99.801	0.99	0.986	0.069
XG_Boost_Test	99.983	0.99	0.836	0.006

CHAPTER 4

RECOMMENDATIONS AND CONCLUSION

The model was trained and tested for its prediction accuracy and a supremely higher efficiency of **99.801% and 99.983%** was achieved for both the training datum and testing datum prediction respectively. It is to be importantly noted that, not only does this analysis and study signify the remarkable prediction accuracy of the automated model built, it also does indicate the enormous monetary amount of **6.5 Billion USD** that can be saved by the XYZ Corp through an advance identification of the loan defaulter during the model implementation and execution. **Additionally, applications received from Joint accounts were identified to not fall under the defaulter category and this key finding is believed to be of significant used in decision making.**

CHAPTER 5

REFERENCES

1. https://towardsdatascience.com/running-xgboost-on-google-colab-free-gpu-a-case-study-841c90fef101
2. https://towardsdatascience.com/powerful-eda-exploratory-data-analysis-in-just-two-lines-of-code-using-sweetviz-6c943d32f34
3. https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a