OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

IFM

LEHRSTUHL FÜR
INNOVATIONS- UND
FINANZMANAGEMENT

# Automated Anomaly Detection in Financial Transactions Applying Machine Learning Algorithms

**Master's Scientific Project in**

**'Project in Fintech and Blockchain Innovation'**

Chair Of Business Management,

Insb. Innovation and Financial Management

**Of Professor Dr. Elmar Lukas**

**Submitted By:**

**Md Mehedi Hasan**
ID: 230319
Email: md2.hasan@st.ovgu.de

**Md Najmul Sharkar**
ID: 229940
Email: md.sharkar@st.ovgu.de

**Supervised By:**

**Kaja Cordes**

Research Assistant
Chair of Business Administration Insb.
Innovation and Financial Management

**Magdeburg 10.09.2021**

# Table of Contents

# 1 Introduction

Financial Companies globally are losing billions of dollars from abnormal and fraudulent activities. Although Institutions and businesses have been trying to mitigate the damages by spending funds and investigating fraudulent actions, the risk of losing money from financial anomalies has not yet been stopped. According to the Concise Oxford Dictionary of Mathematics Clapham (2013), Anomaly defines if the financial activities are unusual or do not follow a normal pattern of a drawn population. Hawkins (1980) coined that an anomaly is an observation that significantly differs from the mean observations.

Various anomalies can occur in financial organizations, such as credit card fraud, loan default, false insurance claims that deviate from usual activities. The detection of such financial anomalies is inevitably necessary for the institutions like banks, non-bank, insurance companies, credit agencies, and peer-to-peer lending. However, detecting these anomalies using traditional or manual work does not always lead to productive results.

Nevertheless, classifying the anomalies applying Machine Learning (ML) can benefit the firms to counter these tremendous losses. According to Branka Stojanovi'c et al., identifying malicious activities is a great challenge for the financial sector. Therefore, approaches from machine learning domain knowledge are applied to detect suspicious fraudulent patterns is very crucial and beneficial for the FinTech industry.

This paper demonstrates one example of financial anomalies, which is a loan default prediction using machine learning algorithms. When customers failed to payback their installments according to the original arrangement known as the loan default. Research indicates that loan defaults have negative impacts on the financial sector’s growth. According to the LCD Survey, The loan default is expected to rise 4% in Europe in 2021. Thus preventing these losses is significantly essential for the Banks and other lenders. This paper aims to classify the loan default prediction applying numerous ML models such as Logistic Regression (LR), Random Forest (RF), and Light Gradient Boosting Machine (LightGBM) and then deploy optimal performed algorithms into a website platform where financial institutions can predict customer loan worthiness (paid/default) by given some information.

# 2 Scientific Background

## 2.1 Literature Review

Banks, non-bank financial institutions, and other credit agencies generate most of their revenues through lending credits. Therefore, deciding on the proper loan sanctions for suitable candidates is a key to financial corporations. For instance, Amoako, K. (2015) describes how risky it could be for financial institutions, including banks, for the non-performing or so-called bad loans. Moreover, Fredriksson, O. & Frykström, N.(2019) mentioned in a scientific paper that increasing bad loans could decline banks' profit drastically and make it difficult to issue new loans to other proper candidates.

To assess the applicant's affordability to repay the installment, detecting the loan default or non-performing loan is a key to any lending institution. Unfortunately, old-fashioned work is sometimes challenging, tiresome, and slow to find the appropriate candidates to lend the credit. On the other hand, The Machine Learning model can efficiently address this issue and detect the potential candidates who may default.

Some notable work has already been done about predicting loan default using Machine Learning and Deep Learning Models to help banks and other businesses to improve the loan process's efficiency. Lin Zhu et al. (2019) publish a paper illustrating the performance of different algorithms of ML to predict the credit default and compare the results of each of the algorithms. The report shows that Random Forest and Decision tree have outperformed the Support Vector Machine (SVM) and Logistic Regression. Random Forest and Decision tree have 98 and 95 percent accuracy, respectively, which performed way better than the SVM (75%) and Logistic Regression (73%). The Paper also mentioned that some algorithms' results might lead to state-of-the-art performances if the model is built by properly tuned their parameters. However, Dhruba et al. (2018) finding is different from the Lin Zhu et al. (2019). Dhruba et al. (2018) carried out a credit risk assessment project executing four supervised ML Models: SVM, Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGB). The authors conclude that SVM and XGB can outperform the other two classification models for the lending club dataset (2007-2011). This paper also recapitulates that SVM costs computational time as it takes most of the time to train the dataset. Therefore, the authors suggested using the neural network model since it tends to outperform the classification models.

Ghatasheh, N. (2014) demonstrates that the Random Forest algorithm is an optimal performance ML Model for predicting credit default. The Paper has also shown that Random forest algorithms have many advantages for imbalanced and skewed datasets, such as overfitting immunity. Also, the correlation makes a reasonable estimate for the ability of the prediction and performs better in parallel processing. Goyal, A. & Kaur, R. (2016) uses several supervised ML algorithms in the R-language platform to determine which algorithms have better outcome and accuracy in predicting the probability of loan default which would be fruitful for the lenders. The report employs the Decision Tree, Linear Regression, Random Forest, Neural Network, SVM, and tree based model for genetic algorithms. The tree based model for genetic algorithms has the highest accuracy (81.25%) compared to all models.

Mehul Madaan et al. (2021) created an intelligent ML Model using the Lending club dataset and proposed that the Random Forest classifier (80%) defeats Decision Tree algorithm (73%) in terms of accuracy. Furthermore, the authors identify that Random forest gives a better result for this type of large dataset. The Paper also explained that a customer has a high probability of defaulting on a loan if the customer does not own a home, applied for small business loans, or asked for wedding loans. Besides, the authors argued that assessing the right features is crucial for the lending club data analysis.

Jency, X.F. et al. (2018) publishes a paper about credit failure prediction based on the nature of the clients. The Paper represents several components of the loan's nature like annual income vs. the purpose of the loan, Loan Term vs. Years in Current Job, Loan Payment Chances vs. Home Ownership. The authors illustrate an admirable work describing the chance of loan default by qualitative and quantitative analysis.

Ma, X. Et al. (2018) performed a study based on an original P2P transaction dataset with xgboost and LightGBM algorithms. The authors of this paper stated that the LightGBM algorithm operates considerably well over xgboost with a 1.28% performance increment. K Huang (2020) also mentioned that LightGBM could perform excellently in fraud detection and exceeded all performance measure matrices compared to logistics regression and SVM.

# 3 Research Methodology

Financial industries are investing money over the years to minimize the customer default risk. The evolution of machine learning and artificial intelligence technology allows them to mitigate the risk through data-centric tools innovation. This project would like to create a tool that will help credit institutions to detect customer default probability in advance. In order to achieve this goal, this project will be carried out within three different sections, and each step will follow specific procedures.

The previous literature and report analyzed and interpreted the various datasets and tried to estimate the non-performing loan. However, this report will analyze the variables associated with the loan decision and developed a mechanism where lending corporations like lending clubs can predict their potential customers.

## 3.1 Data Collection

First of all, it is required to collect the data to build a data-centric model. The data source should be reliable and valuable, should have the ability to carry out the predictive analysis..

## 3.2 Model Build-Up

After collecting the data, it should be cleaned, processed to carry out the explorative analysis. After processing, the data will be fetched into the different machine learning models to analyze. Necessary hyperparameters will be tuned and optimized according to the model performance. All the analysis will be performed using python and its built-in libraries.

According to the above scientific background, a tree-based model like Random Forest, xgboost, and LightGBM performed very well with a large dataset with significant performance capacity. So, this project will apply only Logistics regression to see the general performance without tuning. Later, Random Forest and LightGBM will be applied with proper tuning of the parameters and comparing the results

## 3.3 Model Deployment

From The above analysis, the best-performing model will be deployed into the websites to detect the creditworthiness of a particular customer.

# 4 Exploratory Data Analysis

## 4.1 Data Source

To create the machine learning models and predict better accuracy, the relevance of the dataset is essential. For a data-driven scientific project, the Lending Club loan dataset is used. Lending Club is one of the largest US-based FinTech marketplaces that facilitate connecting borrowers to lenders. Consumers request the loans on Lending Club's website by filling out the necessary details of the candidates; the potential lenders can then skim the website and choose the suitable candidates out of all the potential applicants. The website stores every record of each loan applicant in its databases. The complete dataset is obtained from the Kaggle. Precisely, complete loan data for all loans accepted from 2007 to 2020 are applied. Therefore, the entire dataset contains 142 columns and 29, 25,493 rows *(Appendix-I, Table-1).*

## 4.2 Target Variable

The dependent variable is the *loan_status*, where there are numeric loan categories rows available. Among all the loan status classes, Fully paid, Charged off, and Default is employed. Fully paid loans are those categories that repay the loan successfully with interest. Charges off customers are those who could not continue to pay and hence charges off; Default is those who could not return the loan. However, For the binary classification model, Fully paid would be considered as 0, and Charges off and Default would be regarded as 1. After considering only these rows of the sub-classes, the new dataset reduces to 18, 60,764 rows and 142 columns.

## 4.3 Data Cleaning

The lending club dataset comprises 142 attributes and almost 3 million rows. Therefore, there would be a high chance that some data are irrelevant in this vast dataset, thus not producing fruitful results when predicting the model. Hence, Identifying and removing unnecessary features is vital before the data preprocessing and building the Machine Learning model.

### 4.3.1 Deal with Missing Values:

> “Missing values are common occurrences in data. Unfortunately, most predictive modeling techniques cannot handle any missing values.Notably, Naive Bayes and

k-Nearest Neighbors support missing values. Thus, removing missing values is necessary before applying in to the model. Therefore, this problem must be addressed prior to modeling."- Kuhn, M. & Johnson, K.(2019). Therefore, the columns consisting of more than 50% missing values are eliminated from the dataset. If the dataset becomes small, imputing missing values with mean and median will become useful. However, there are several columns in the lending club dataset where 90% of missing values are present. Hence, imputing would not be a wise decision and not lead to a productive modeling result. The elimination of missing and NaN values reduced the number of features from 151 to 107.

### 4.3.2 Removing Unique Features:

Features with a unique constant value will be excluded from the dataset. Those features will not be appropriately explained to the dependent variable since the variance of unique features value is zero. The code for removing unique features will go through each feature sequentially and detect any features that have only constant/unique rows in the entire column. Since the tree-based ML algorithm does not produce better accuracy for these unique features, elimination is necessary.

### 4.3.3 Removing Duplicate Rows and Columns:

Removing duplicate features and rows is a standard style for data preprocessing. First, the duplicate rows are deleted from the dataset to make each row unique. Furthermore, the features that have the same value and meaning have been eliminated. By doing so, the number of features currently present is 105.

### 4.3.4 Dropping Features that have no Explainability:

Now, every feature has been traversed individually to see the explainability of the Target variable loan_status. Some features are being dropped as they do not have any relational meaning with the explanatory variable. Independent features with no explanation power are removed, e.g., issue_month, emp_length. The month when the loan is issued, and the applicants' employment length does not influence whether the loan was fully paid or Default. The resulting features have been decreased from 105 to 40 independent variables where the remaining data analysis will take place.

### 4.3.5 Correlation with Target Value:

The highly collinear features should be removed to make each independent variable exogenous because multicollinearity affects the model's predictive accuracy. The model accuracy is still as good as when presenting the highly collinear features, but it estimates the bias accuracy and reduces the prediction quality. For example, the correlation between the funded amount and loan amounts is +1; hence, there is no point in keeping both the variables. Moreover, some unnecessary features are dropped from the reduced data frame, making the number of predicting variables 16.

# 5 Feature Engineering

Feature Engineering is essential before applying the dataset into the ML model, and it can make the difference between a weak and robust model.

## 5.1 Feature Scaling

Feature scaling is accomplished through normalization and standardization of the data. For example, when the feature is required to keep only two values, e.g. [0, 1], it is referred to as normalization. Notably, the feature "term" is made to a [0, 1] value instead of keeping the original value [36, 72] respectively. In addition, the loan amount, annual income, and FICO Score feature's value have been log-transformed so that the normal distribution of the data is maintained and the Machine can predict better as the data points are less skewed.

## 5.2 Handling Imbalanced Data

Some techniques can balance the skewed data, such as clustering the abundant class, downsampling, upsampling. Downsampling reduces the abundant classes' data points; on the contrary, upsampling is used when there are no sufficient data points available. In the case of imbalanced data analysis, both of the procedures are beneficial. The ratio of the dataset for the Fully paid and defaults are 80.3% and 19.7%, respectively. It indicates that the dataset is imbalanced. That is why, in the project, downsampling and undersampling are used to examine the model performance.

# 6 Model Description

## 6.1 Logistic Regression

Logistic regression (LR), also called a logit model, is a Machine Learning classification model to classify the dependent variable as a binary output based on multiple independent variables. Logistic regression usually builds the probability model of certain classes or groups like win or loss, up or down, spam or not spam. LR uses the sigmoid activation function to calculate each class's probability, where a predefined threshold determines the class label. The following equation illustrates the logistic regression model, where X is the Input value introduced as the Greek letter Beta to make the probability of the output Y. The primary difference from the linear regression is that LR uses binary values (0 or 1) instead of numeric values.

$$\boldsymbol{y = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1)x}}}$$

## 6.2 Random Forest

Random Forest is a supervised ensemble ML method applied for classification, regression analysis, or other tasks that operate by various individual decision trees at training time and predict the output by averaging individual decision trees. Every individual tree is unrelated to each other. There are some advantages of the Random forest over other classification algorithms. It predicts better accuracy when it comes to large imbalanced datasets. Furthermore, it can compute efficiently even with large missing values while maintaining accuracy.

## 6.3 Light Gradient Boosting Machine (LightGBM)

Gradient Boosting is another ensemble learning algorithm that is applied for classification and regression analysis. The ensemble learning is operated through the decision tree models. *Light Gradient Boosting Machine* is an open-source library that provides an efficient implementation of gradient boosting. Brownlee, J.(2020) describes that

"LightGBM extends the gradient boosting algorithm by adding a type of automatic feature selection as well as focusing on boosting examples with larger gradients. This algorithm can speed up the result of training and improved predictive performance."

# 7 Model Validation and Evaluation

## 7.1 Cross-Validation

Several statistical methods can be applied to measure the performance, and cross-Validation is one of them. It is employed to defend against overfitting in a predictive model, precisely when data may be limited. A predefined number of folds is used, run the data on each fold, and average the performance to compare the test dataset results.

### 7.1.1 Holdout method:

The holdout method is the simple and easy cross-validation method. It is the process of splitting the data into two categories- train and test data set. First, the Machine Learning algorithms are trained on using the training data, and then the Machine predicts the output based on unseen data, i.e., test data. Thus, the data set is split in such a way that 80% of the data are used as training data, and the remaining 20% are used as testing data.

### 7.1.2 K-fold cross-validation:

K-fold cross-validation is another improved method of cross-validation where the training and testing data splits randomly. Usually, the model is trained on and predicts the accuracy based on testing data. "K" is the number of cross-validation, where the validation is repeatedly performed depending on the number of K. For example- for the 5-fold cross-validation, the data is split into five different groups, and each data set would perform training and testing. For that reason, the Individual group of the dataset would get a chance to test alone and predict output accordingly. In this project, 5-fold cross-validation is used to check the sustainability of the model's performance.

## 7.2 Evaluation Metrics:

### 7.2.1 Confusion Matrix:

After the data cleaning, preprocessing, and feature engineering, the training dataset put into the Machine Learning model and test data would predict the model performance. To measure the performance of the ML Model, the Confusion Matrix is an essential tool to use. Confusion Matrix classifies into two classes, predicted class, and actual class.

**True Positive:** The loan default dataset would predict positive, and it is positive.

**True Negative:** The dataset of loan default predicts negative, and it is negative.

**False Positive:** The Model predicts positive, and it is false. It is also known as type-I error.

**False Negative:** The model predicts a negative, and it is not actually true. This is also referred to as Type-II error.

### 7.2.2 Classification Report

**Accuracy-** This is one of the best ways to estimate the performance of the Machine Learning Model. Accuracy illustrates the rate of correctly predicted data out of all the data points available.

**Recall-** Recall is calculated as True positive divided by True positive and False Negative. Thus, it measures the number of predictions that are correctly predicted over all of the datasets. It is also known as the sensitivity of the model.

**Precision-** Precision is calculated as True positive divided by True Positive and False Positive. It estimates how many times the ML model correctly predicts the data over the correct and incorrect predictions.

**F1 Score-** It also measures the efficiency of the ML model. F1 Score measures the balance of Precision and Recall score.

### 7.2.3 ROC-AUC Curve:

AUC-ROC curve is another performance metric for classification that can be adopted to evaluate the model efficiency. AUC measures the degree of separability, and ROC represents the probability curve. The higher the AUC curve is, the better the model's predictability, i.e., a high AUC curve illustrates distinguishing between the fully paid and default class. A good model can have the AUC curve near 1, which means a good measure of separability, and a wrong model can have a score of 0, which does not classify any data.

# 8 Results and Discussion

This section discusses the Result of the Machine Learning models that are applied in the project. Three different ML algorithms are used, as specified earlier. The data are selected based on the feature engineering are segregated as train and test data. Therefore, the model can predict the accuracy and provide the Result of each algorithm based on test data. As mentioned in the data analysis section, the upsampling and downsampling method is applied since the dataset is imbalanced.

## 8.1 Logistic Regression

Logistic regression (LR) is simplistic and straightforward to implement. Before moving on to the complex algorithms, it is decent to start with the logistic regression model. The model provides relatively good work for prediction with an overall accuracy of 67% on upsample data. Though the result is not that impressive, LR predicts for the 0 and 1 classes equally. According to *Appendix III, Table -5*, it is successfully predicted that 73% of the loan is fully paid off but degraded in predicting charged-off loans with only 61%. The weighted average F1 score provides 67% as well. In contrast, the overall accuracy and other evaluation matrix remain the same (67%) for the downsampling method except for the precision. The precision has slightly decreased to 65% for the fully paid loan class.

## 8.2 Random Forest

The Random Forest algorithm for the loan default prediction is a Random forest classifier. Since most of the variables of the data frame are categorical, the tree-based algorithm would be ideal in this situation of imbalanced data analysis. The random forest gives 75% accuracy, which is 8% better than the Logistic regression. According to Appendix III, Table -6, RF estimates the number of borrowers who fully paid the loans with an accuracy of 75% (recall -0). The defaulted loan prediction accuracy rate is also 75% (recall 1), which is very good compared to the LR Model. The F1 score or harmonic mean of precision and recall is 75%.

Additionally, while downsampling the total dataset, results seem not promising with 68% accuracy. In predicting paid-off loans, it provides 70% but is not good in case of charged-off a loan (66%), which worsens than oversampling performance. Random Forest performance was also validated with cross-validation, and mean results are pretty same with initial hold-out sample performance.

## 8.3 LightGBM

The Light Gradient boosting Machine (LightGBM) did an excellent job predicting the anomaly class of the data frame. After the hyperparameter tuning and upsampling of the minority class, LightGBM predicts 86% accuracy. This ensemble learning algorithm presents 88% prediction accuracy considering the loan which is being defaulted (recall). Furthermore, it predicts 84% where the loan is fully paid (*Appendix III, Table-7*). ROC-AUC score is also very high, precisely 0.93 (*Appendix III, Table-3)*. Therefore, out of the other two previous models, LightGBM provides greater accuracy and promising results. LightGBM with downsampling dataset did not perform well compared to the upsampling procedures. The recall score for class 1 (anomaly detection) is only 67% compare to the 88% in the upsampling. In addition, the overall accuracy, f1 score, and AUC-ROC score are falling behind. LightGBM cross-validation results also matched with the hold-out sample result (*Appendix III, Table-8*).

# 9 Website

## 9.1 Website Deployment Using Heroku

Heroku is an open-source platform as a service used to deploy the python code using a flask library. First, the machine learning model, data processing, a python code using flask library, website content and design using HTML, CSS, javascript, a profile are uploaded in the Github account. After that, a Heroku account was created and then started deploying the model using the Heroku platform to run the website globally.

## 9.2 Website Development

After performing the Machine Learning Model, It is seen that LightGBM is the optimal performed Machine Learning Model. Now, a website is created using the LightGBM algorithm with upsampling dataset, so that financial institutions can use it to see the predictability of their customer's chances of getting default. A preloaded template of a website that had all the HTML, CSS, and JavaScript files linked was downloaded. The template, code, pictures, and variables were edited as per the project's ML model requirement. The website has two important tabs- home and prediction. The home tab included the logo, a background picture of the website, and the developer's brief introduction with their LinkedIn and Github profile. The prediction of credit default has been made with the final selected features which are mentioned given in the *Appendix-III Table-2.*

If the model predicts as default (1) as per the given input, the result shown *" Loan Denied! Customer May Default"* else the result shown that *" Loan Approved!".* A screenshot of the prediction tab of the website is given in the *Appendix IV.* In addition to that, the website link is - *https://automated-anomaly-detection.herokuapp.com/.*

# 10 Scope for Further Research

The principal aim towards developing the product is to minimize the losses that financial institutions incur. By detecting the credit anomaly, lending corporations can save unexpected losses. However, at the same time, the model may detect a clean customer as default. The company may lose profit from that customer now and in subsequent years. The cost-benefit analysis can be done carried out in further scope.

Moreover, the period of the dataset that was collected is not that large. The larger the dataset is, the model can train them better. Ideally, a larger sample would be more beneficial so that the holdout set can cover a longer period.

Due to software skills and technical difficulties, the prediction model website kept it a straightforward design. So, there is an area where improvement needs to be addressed.

The analysis performed in this report may be a promising approach for the financial industry that provides credit to individuals and businesses. However, there will be a scope to analyze the peer-to-peer lending analysis as the data gathered are from the peer-to-peer lending platform. This study might improve the overall performance as some features will not be eliminated as the analysis concentrates entirely on the peer-to-peer lending industry.

# 11 Summary and Conclusion

Finding the potential borrowers who will likely to be default is the most crucial task for any financial lender. Many institutions have been using manual processes to detect the candidates for not lending the credits. That would be a tedious and time-consuming task to check the affordability of each loan candidate. FinTech can serve an essential purpose in the credit lending industry. Many finance organizations have been using the automation process to find the applicants whether they can repay the loan with interest or the chance that applicants can not repay the installments. Machine Learning tools is one of the best ways to instantiate these processes of checking the relevant candidates. In this project, the Lending Club, a peer-to-peer credit lending platform, data is used to build up the project. Appropriate data preprocessing, feature engineering are performed so that only relevant variables are chosen that can explain the target variables. After that, Various ML algorithms are implemented to see the accuracy of each Model. The LightGBM, a best-performed model, is selected that has a accuracy of 88% for the loan default class. That means the Model can predict 88 out of 100 times to accurately predict the candidates who will likely not repay the loan to the lenders. The LightGBM performed excellent work in terms of model accuracy, precision, recall score. After that, A python built-in library flask is used to deploy the Model into the Heroku platform to create the website for the credit lending user. For the web development, necessary tools such as HTML, CSS, JavaScript code are utilized that are uploaded in the GitHub account, including the ML model and flask code. The website is considered an excellent product for Banks, non-bank financial institutions, and other credit agencies.

## References:

Morozov, I.(2016).Anomaly Detection in Financial Data by Using Machine Learning Methods. *Haw Hamburg*. Retrieved from here.

Hawkins, D.M.(1980). Identification of outliers, volume 11.

Stojanovi´c, B et al.(2021). Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications. *MDPI*. Retrieved from here.

Cox, D, & Witt, I.(2020). European leveraged loan defaults expected to hit 4% in 2021 — LCD survey. *S&P Global Market Intelligence*. Retrieved from here.

Amoako, K.A.(2015). The Effect of Bad Loans on the Profitability and Lending Potential of Rural Banks. *Semantics Scholar*. Retrieved from here.

Fredriksson, O. & Frykström, N.(2019).” Bad loans” and their effects on banks and financial stability. *Economic Commentary*. Retrieved from here.

Zhu, L et al. (2019). A study on predicting loan default based on the random forest algorithms. ScienceDirect. Retrieved from here.

Ghatasheh, N.(2014). Business Analytics using Random Forest Trees for Credit Risk Prediction: A Comparison Study. *International Journal of Advanced Science and Technology Vol.72 (2014), pp.19-30*. Retrieved from here.

Goyal, A. & Kaur, R.(2016). Accuracy Prediction for Loan Risk Using Machine Learning Models. *International Journal of Computer Science Trends and Technology (IJCST) – Volume 4*. Retrieved from here.

Madaan, M et al.(2021). Loan default prediction using decision trees and random forest: A comparative study. IOPscience. Retrieved from **here**

Wikipedia. 2009. “LendingClub.” 9 July 2021. Retrieved from **here**

Dhruvo, M.I.M et al.(2018). Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking. *IEEE Xplore*. Retrieved from **https://s3-us-west-2.amazonaws.com/ieeeshutpages/xplore/xplore-ie-notice.html**?

Brownlee, J.(2020). How to Develop a Light Gradient Boosted Machine (LightGBM) Ensemble. Retrieved from here

Kuhn, M. & Johnson, K.(2019). Feature Engineering and Selection: A Practical Approach for Predictive Models. Page 203, Retrieved from here.

Allibhai, E. (2018). Hold-out vs. Cross-validation in Machine Learning. *Medium*. Retrieved from here.

Narkhede, S.(2018). Understanding Confusion Matrix. *Toward Data Science. Retrieved from* here.

Lending Club. (2021). *Wikipedia.* Retrieved from here.

Heroku. (2021). *Wikipedia.* Retrieved from here.

Ma, X. Et al. (2018).Study on A Prediction of P2P Network Loan Default Based on the Machine Learning LightGBM and XGboost Algorithms according to Different High Dimensional Data Cleaning. *ResearchGate.* Retrieved from here.

K Huang (2020).An Optimized LightGBM Model for Fraud Detection. *Journal of Physics: Conference Series*. Retrieved from here.

# Appendix-I: All Lending Club variables:

**Table-1: All Variables name and description.**

| Sl. No. | Index Name | Description |
|---|---|---|
| 1 | Id | A unique LC assigned ID for the loan listing. |
| 2 | fico_range_high | The upper boundary range the borrower’s FICO at loan origination belongs to. |
| 3 | revol_bal | Total credit revolving balance |
| 5 | out_prncp | Remaining outstanding principal for total amount funded |
| 6 | out_prncp_inv | Remaining outstanding principal for portion of total amount funded by investors |
| 7 | total_pymnt | Payments received to date for total amount funded |
| 8 | total_pymnt_inv | Payments received to date for portion of total amount funded by investors |
| 9 | total_rec_prncp | Principal received to date |
| 10 | total_rec_int | Interest received to date |
| 11 | total_rec_late_fee | Late fees received to date |
| 12 | Recoveries | post charge off gross recovery |
| 13 | collection_recovery_fee | post charge off collection fee |
| 14 | last_pymnt_amnt | Last total payment amount received |
| 15 | last_fico_range_high | The upper boundary range the borrower’s last FICO pulled belongs to. |
| 16 | last_fico_range_low | The lower boundary range the borrower’s last FICO pulled belongs to. |
| 17 | policy_code | publicly available policy_code=1<br>new products not publicly available policy_code=2 |

| | | |
|---|---|---|
| 19 | fico_range_low | The lower boundary range the borrower’s FICO at loan origination belongs to. |
| 23 | funded_amnt_inv | The total amount committed by investors for that loan at that point in time. |
| 26 | Installment | The monthly payment owed by the borrower if the loan originates. |
| 30 | loan_amnt | The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value. |
| 33 | funded_amnt | The total amount committed to that loan at that point in time. |
| 4 | initial_list_status | The initial listing status of the loan. Possible values are – W, F |
| 18 | application_type | Indicates whether the loan is an individual application or a joint application with two co-borrowers |
| 20 | addr_state | The state provided by the borrower in the loan application |
| 21 | debt_settlement_flag | Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company. |
| 22 | verification_status | Indicates if income was verified by LC, not verified, or if the income source was verified. |
| 24 | Term | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| 25 | int_rate | Interest Rate on the loan |
| 27 | Grade | LC assigned loan grade |
| 28 | sub_grade | LC assigned loan subgrade |
| 29 | home_ownership | The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER |
| 31 | issue_d | The month which the loan was funded |
| 32 | loan_status | Current status of the loan |
| 34 | url | URL for the LC page with listing data. |

| | | |
|---|---|---|
| 35 | pymnt_plan | Indicates if a payment plan has been put in place for the loan |
| 36 | Purpose | A category provided by the borrower for the loan request. |
| 37 | zip_code | The first 3 numbers of the zip code provided by the borrower in the loan application. |
| 38 | annual_inc | The self-reported annual income provided by the borrower during registration. |
| 39 | total_acc | The total number of credit lines currently in the borrower's credit file |
| 40 | pub_rec | Number of derogatory public records |
| 41 | open_acc | The number of open credit lines in the borrower's credit file. |
| 42 | delinq_amnt | The past-due amount owed for the accounts on which the borrower is now delinquent. |
| 43 | delinq_2yrs | The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years |
| 45 | acc_now_delinq | The number of accounts on which the borrower is now delinquent. |
| 44 | earliest_cr_line | The month the borrower's earliest reported credit line was opened |
| 46 | inq_last_6mths | The number of inquiries in past 6 months (excluding auto and mortgage inquiries) |
| 47 | last_credit_pull_d | The most recent month LC pulled credit for this loan |
| 48 | tax_liens | Number of tax liens |
| 49 | collections_12_mths_ex_med | Number of collections in 12 months excluding medical collections |
| 50 | chargeoff_within_12_mths | Number of charge-offs within 12 months |
| 51 | pub_rec_bankruptcies | Number of public record bankruptcies |
| 52 | revol_util | Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit. |

| | | |
|---|---|---|
| 53 | Dti | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income. |
| 54 | last_pymnt_d | Last month payment was received |
| 55 | Title | The loan title provided by the borrower |
| 56 | hardship_flag | Flags whether or not the borrower is on a hardship plan |
| 57 | total_bc_limit | Total bankcard high credit/credit limit |
| 58 | mort_acc | Number of mortgage accounts. |
| 59 | total_bal_ex_mort | Total credit balance excluding mortgage |
| 60 | acc_open_past_24mths | Number of trades opened in past 24 months. |
| 61 | num_bc_sats | Number of satisfactory bankcard accounts |
| 62 | num_sats | Number of satisfactory accounts |
| 63 | num_op_rev_tl | Number of open revolving accounts |
| 64 | num_actv_rev_tl | Number of currently active revolving trades |
| 65 | mo_sin_rcnt_tl | Months since most recent account opened |
| 66 | num_bc_tl | Number of bankcard accounts |
| 67 | num_actv_bc_tl | Number of currently active bankcard accounts |
| 68 | total_rev_hi_lim | Total revolving high credit/credit limit |
| 69 | num_rev_tl_bal_gt_0 | Number of revolving trades with balance >0 |
| 70 | num_il_tl | Number of installment accounts |
| 71 | num_tl_30dpd | Number of accounts currently 30 days past due (updated in past 2 months) |
| 72 | total_il_high_credit_limit | Total installment high credit/credit limit |
| 73 | tot_coll_amt | Total collection amounts ever owed |
| 74 | tot_cur_bal | Total current balance of all accounts |

| | | |
|---|---|---|
| 75 | tot_hi_cred_lim | Total high credit/credit limit |
| 76 | num_accts_ever_120_pd | Number of accounts ever 120 or more days past due |
| 77 | num_tl_op_past_12m | Number of accounts opened in past 12 months |
| 78 | num_tl_90g_dpd_24m | Number of accounts 90 or more days past due in last 24 months |
| 79 | mo_sin_old_rev_tl_op | Months since oldest revolving account opened |
| 80 | mo_sin_rcnt_rev_tl_op | Months since most recent revolving account opened |
| 81 | num_rev_accts | Number of revolving accounts |
| 82 | avg_cur_bal | Average current balance of all accounts |
| 83 | pct_tl_nvr_dlq | Percent of trades never delinquent |
| 84 | mths_since_recent_bc | Months since most recent bankcard account opened. |
| 85 | bc_open_to_buy | Total open to buy on revolving bankcards. |
| 86 | percent_bc_gt_75 | Percentage of all bankcard accounts > 75% of limit. |
| 87 | bc_util | Ratio of total current balance to high credit/credit limit for all bankcard accounts. |
| 88 | mo_sin_old_il_acct | Months since oldest bank installment account opened |
| 89 | num_tl_120dpd_2m | Number of accounts currently 120 days past due (updated in past 2 months) |
| 90 | emp_length | Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years. |
| 91 | emp_title | The job title supplied by the Borrower when applying for the loan.* |
| 92 | mths_since_recent_inq | Months since most recent inquiry. |
| 93 | open_rv_24m | Number of revolving trades opened in past 24 months |
| 94 | inq_fi | Number of personal finance inquiries |
| 95 | open_act_il | Number of currently active installment trades |

| | | |
|---|---|---|
| 96 | max_bal_bc | Maximum current balance owed on all revolving accounts |
| 97 | open_rv_12m | Number of revolving trades opened in past 12 months |
| 98 | total_bal_il | Total current balance of all installment accounts |
| 99 | open_il_24m | Number of installment accounts opened in past 24 months |
| 100 | open_il_12m | Number of installment accounts opened in past 12 months |
| 101 | inq_last_12m | Number of credit inquiries in past 12 months |
| 102 | total_cu_tl | Number of finance trades |
| 103 | open_acc_6m | Number of open trades in last 6 months |
| 104 | all_util | Balance to credit limit on all trades |
| 105 | mths_since_rcnt_il | Months since most recent installment accounts opened |
| 106 | il_util | Ratio of total current balance to high credit/credit limit on all install acct |
| 107 | mths_since_last_delinq | The number of months since the borrower's last delinquency. |
| 108 | next_pymnt_d | Next scheduled payment date |
| 109 | mths_since_recent_revol_delinq | Months since most recent revolving delinquency. |
| 110 | mths_since_last_major_derog | Months since most recent 90-day or worse rating |
| 111 | mths_since_recent_bc_dlq | Months since most recent bankcard delinquency |
| 112 | mths_since_last_record | The number of months since the last public record. |
| 113 | annual_inc_joint | The combined self-reported annual income provided by the co-borrowers during registration |
| 114 | dti_joint | A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported |

| | | |
|---|---|---|
| | | monthly income |
| 116 | sec_app_fico_range_low | FICO range (high) for the secondary applicant |
| 117 | sec_app_fico_range_high | FICO range (low) for the secondary applicant |
| 118 | sec_app_inq_last_6mths | Credit inquiries in the last 6 months at time of application for the secondary applicant |
| 119 | sec_app_mort_acc | Number of mortgage accounts at time of application for the secondary applicant |
| 120 | sec_app_open_acc | Number of open trades at time of application for the secondary applicant |
| 121 | sec_app_collections_12_mths_ex_med | Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant |
| 122 | sec_app_open_act_il | Number of currently active installment trades at time of application for the secondary applicant |
| 123 | sec_app_num_rev_accts | Number of revolving accounts at time of application for the secondary applicant |
| 124 | sec_app_chargeoff_within_12_mths | Number of charge-offs within last 12 months at time of application for the secondary applicant |
| 115 | sec_app_earliest_cr_line | Earliest credit line at time of application for the secondary applicant |
| 125 | revol_bal_joint | Sum of revolving credit balance of the co-borrowers, net of duplicate balances |
| 126 | verification_status_joint | Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified |
| 127 | sec_app_revol_util | Ratio of total current balance to high credit/credit limit for all revolving accounts |

| | | |
|---|---|---|
| 128 | hardship_last_payment_amount | The last payment amount as of the hardship plan start date |
| 129 | hardship_amount | The interest payment that the borrower has committed to make each month while they are on a hardship plan |
| 130 | hardship_payoff_balance_amount | The payoff balance amount as of the hardship plan start date |
| 131 | orig_projected_additional_accrued_interest | The original projected additional interest amount that will accrue for the given hardship payment plan as of the Hardship Start Date. This field will be null if the borrower has broken their hardship payment plan. |
| 132 | deferral_term | Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan |
| 136 | hardship_length | The number of months the borrower will make smaller payments than normally obligated due to a hardship plan |
| 133 | hardship_start_date | The start date of the hardship plan period |
| 134 | hardship_end_date | The end date of the hardship plan period |
| 135 | payment_plan_start_date | The day the first hardship plan payment is due. For example, if a borrower has a hardship plan period of 3 months, the start date is the start of the three-month period in which the borrower is allowed to make interest-only payments. |
| 137 | hardship_type | Describes the hardship plan offering |
| 138 | hardship_dpd | Account days past due as of the hardship plan start date |
| 139 | hardship_status | Describes if the hardship plan is active, pending, canceled, completed, or broken |
| 140 | hardship_reason | Describes the reason the hardship plan was offered |
| 141 | hardship_loan_status | Loan Status as of the hardship plan start date |

# Appendix II: Tools, Technology and Platform Used

**Python:** Python is a dynamic and well-structured, open-source programming language that is easy to implement. It is one of the popular programming languages for web applications. The Python programming language is used for data preprocessing, manipulation, and building the machine learning modeling in the scientific project. There are many libraries and packages used in the project such as-

**Pandas:** for data frame operations.

**Numpy :** for array and numerical operations.

**Matplotlib, Seaborn, and Plotly :** for visualization of the data.

**Sklearn:** for various machine learning classification models like the random forest, Logistic Regression, and metrics like roc_auc_score, etc.

**Dill:** for saving machine learning models for pickling the model.

**Flask:** For deployment the model into website.

**LighGBM**: a ML Model

**Google Colaboratory:** The entire programming is carried out in the Google Colaboratory for the efficiency of the runtime. The Google Colab supports the Python programming language, and it’s well standard and easy to manage.

**HTML**: The Hypertext Markup Language, shortly HTML, is the primary and standard markup language for creating web development. It defines the outline and structure of the website pages. In addition, HTML is used to describe the content of the Scientific Project’s website.

**CSS**: CSS is a language used to describe and design the presentation of the markup language written in HTML.

**JavaScript**: JavaScript is a robust programming language that is used for website creation alongside HTML and CSS.

**Github**: Github is a software development company that serves source code management functionality using Git. Github account was opened to upload the code to deploy the model into the website platform. Microsoft acquired Github in 2012.

**Heroku**: Heroku is a cloud that performs as a service (PaaS) that supports multiple programming languages. Heroku supports the Python programming language. The model is deployed applying flask libraries and displayed as a website in the Heroku platform. It's free and easy to implement.

# Appendix III: Figure and Tables

**Figure 1: Correlations of among the selected features.**

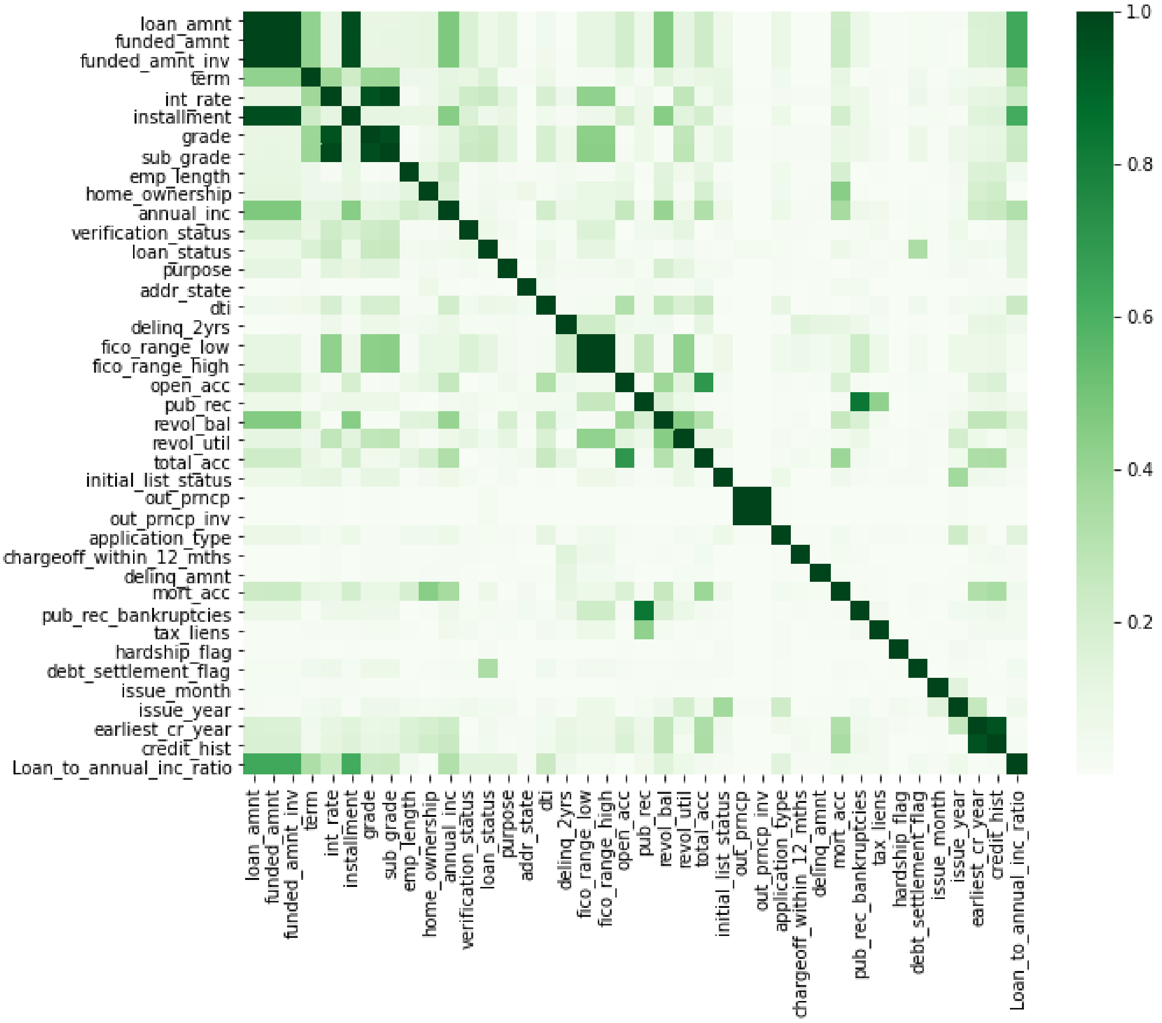


**Table 2: Final variable used for the prediction:**

| Sl | Variables | Description |
|---|---|---|
| 1 | loan_amount | The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value. |
| 2 | term | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| 3 | sub_grade | LC assigned loan subgrade. |
| 4 | emp_length | Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years. |
| 5 | home_ownership | The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER. |
| 6 | annual_inc | The self-reported annual income provided by the borrower during registration. |
| 7 | verification_status | Indicates if income was verified by LC, not verified, or if the income source was verified. |
| 8 | dti | A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income. |
| 9 | fico_range_high | The upper boundary range the borrower's FICO at loan origination belongs to. |
| 10 | revol_util | |
| 11 | application_type | Indicates whether the loan is an individual application or a joint application with two co-borrowers. |
| 12 | mort_act | Number of mortgage accounts. |
| 13 | pub_rec_bankruptcies | Number of public record bankruptcies. |
| 14 | debt_settlement_flag | Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company. |

| | | |
|---|---|---|
| 15 | credit_hist | This is created manually. |

**Table 3: Performance Comparison of ML Model with Upsampling**

| Algorithms | Accuracy | ROC-AUC Score | Class | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|
| Logistic Regression | 67% | 0.75 | 0 | 66% | 73% | 69% |
| | | | 1 | 70% | 61% | 65% |
| Random Forest | 75% | 0.83 | 0 | 75% | 75% | 75% |
| | | | 1 | 75% | 75% | 75% |
| LightGBM | 86% | 0.93 | 0 | 87% | 84% | 85% |
| | | | 1 | 84% | 88% | 86% |
| LightGBM with Threshold (0,40) | 83% | 93% | 0 | 93% | 71% | 80% |
| | | | 1 | 76% | 95% | 85% |

**Table 4: Performance Comparison of ML Model with Downsampling**

| Algorithms | Accuracy | ROC-AUC Score | Class | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|
| Logistic Regression | 67% | 0.74 | 0 | 65% | 73% | 69% |
| | | | 1 | 70% | 61% | 65% |
| Random Forest | 68% | 0.75 | 0 | 67% | 70% | 69% |
| | | | 1 | 69% | 66% | 68% |
| LightGBM | 69% | 0.77 | 0 | 69% | 72% | 70% |
| | | | 1 | 71% | 67% | 68% |
| LightGBM with Threshold (0,45) | 69% | 0.77 | 0 | 71% | 67% | 69% |
| | | | 1 | 68% | 72% | 70% |

**Table 5: Logistic Regression Classification Report**

| Dataset | Accuracy | Class | Precision | Recall | Weighted F1 |
|---|---|---|---|---|---|
| **Downample** | 67% | 0 | 65% | 73% | 67% |
| | | 1 | 70% | 61% | |
| **Upsample** | 67% | 0 | 67% | 73% | 67% |
| | | 1 | 70% | 61% | |

**Figure 2: Confusion Matrix of Logistic Regression-Upsampling**

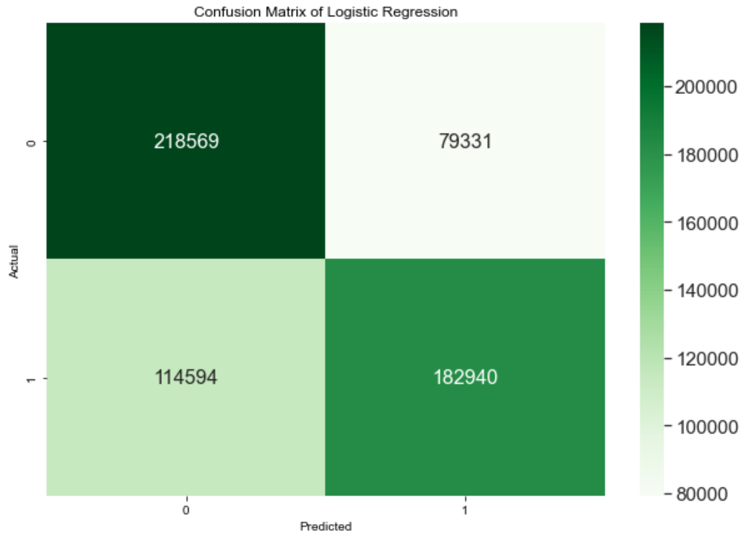


**Figure 3: ROC Curve for Logistic Regression-Upsampling**

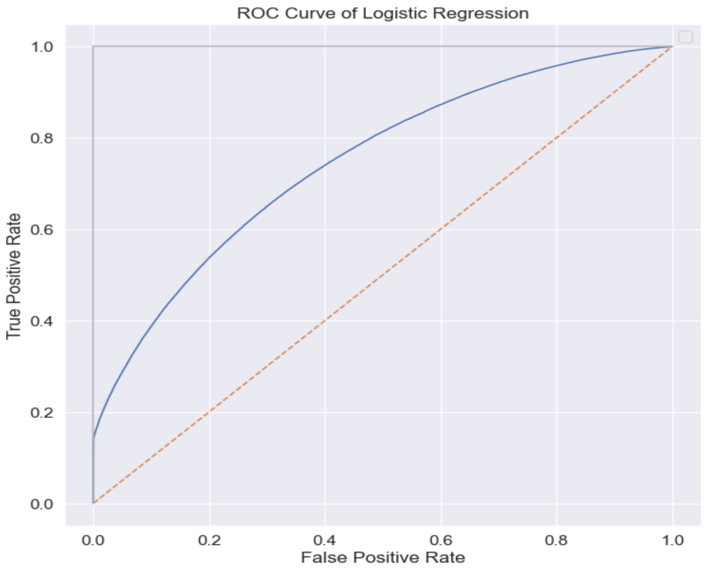


**Figure 4: Confusion Matrix of Logistic Regression-Downsampling**

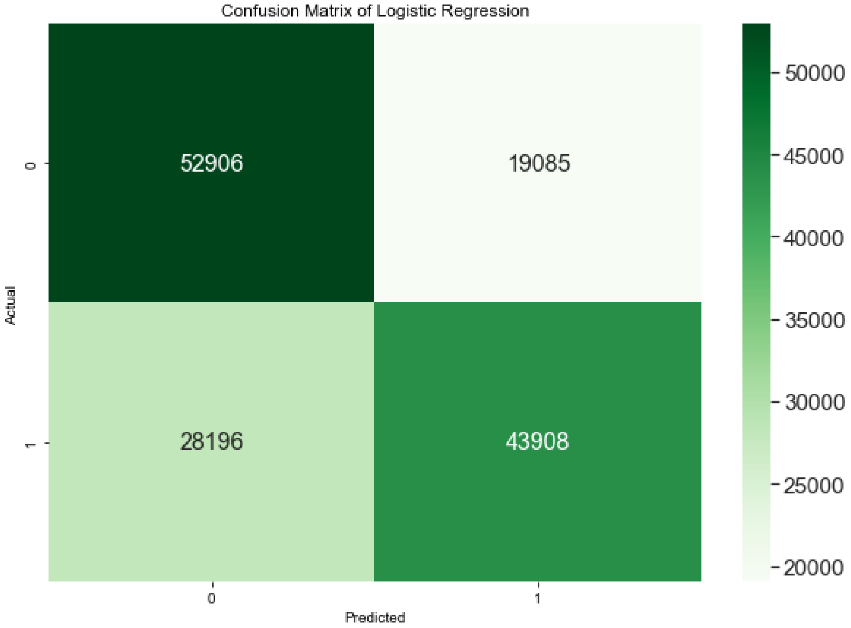


**Figure 5: ROC Curve for Logistic Regression- Downsampling**

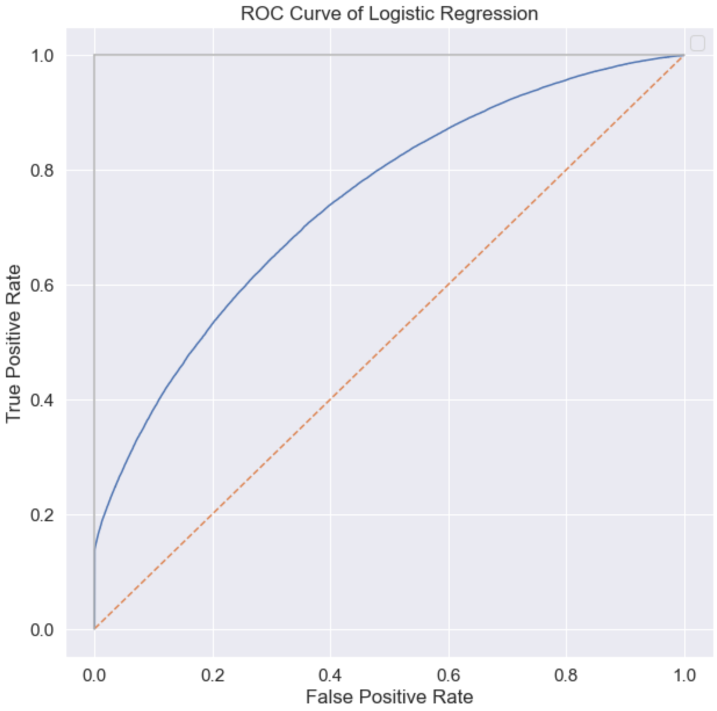


**Table 6: Classification Report of Random Forest**

| Dataset | Accuracy | Class | Precision | Recall | Weighted F1 |
|---|---|---|---|---|---|
| **Downample** | 68% | 0 | 67% | 70% | 68% |
| | | 1 | 69% | 66% | |
| **Upsample** | 75% | 0 | 75% | 75% | 75% |
| | | 1 | 75% | 75% | |

**Figure 6: Confusion Matrix of Random Forest-Upsampling**

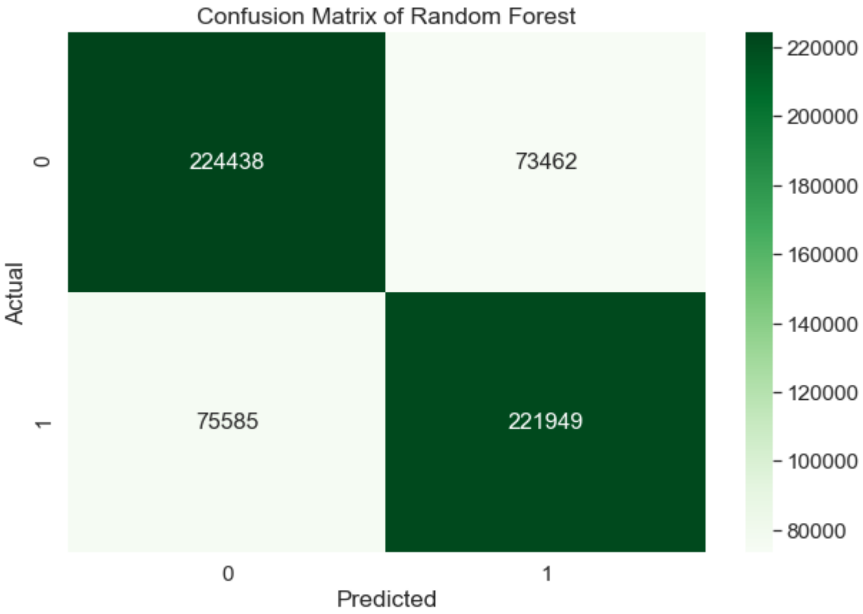


**Figure 7: ROC Curve for Random Forest- Upsampling**

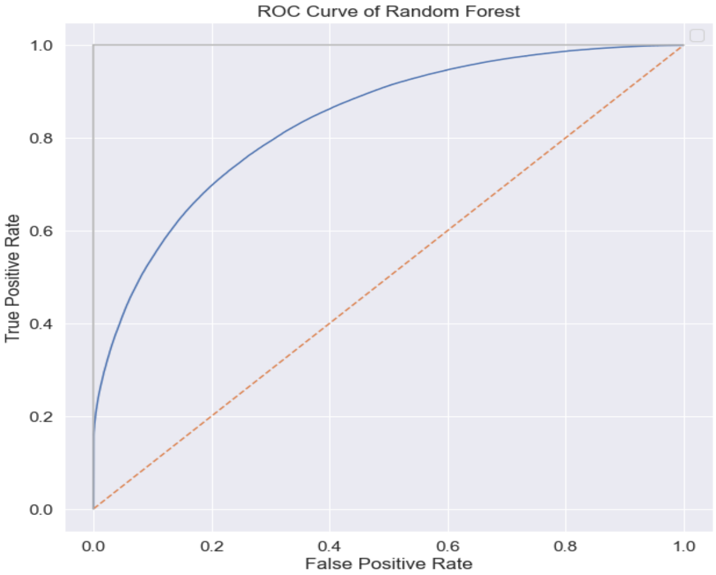


**Figure 8: Confusion Matrix of Random Forest-Downsampling**

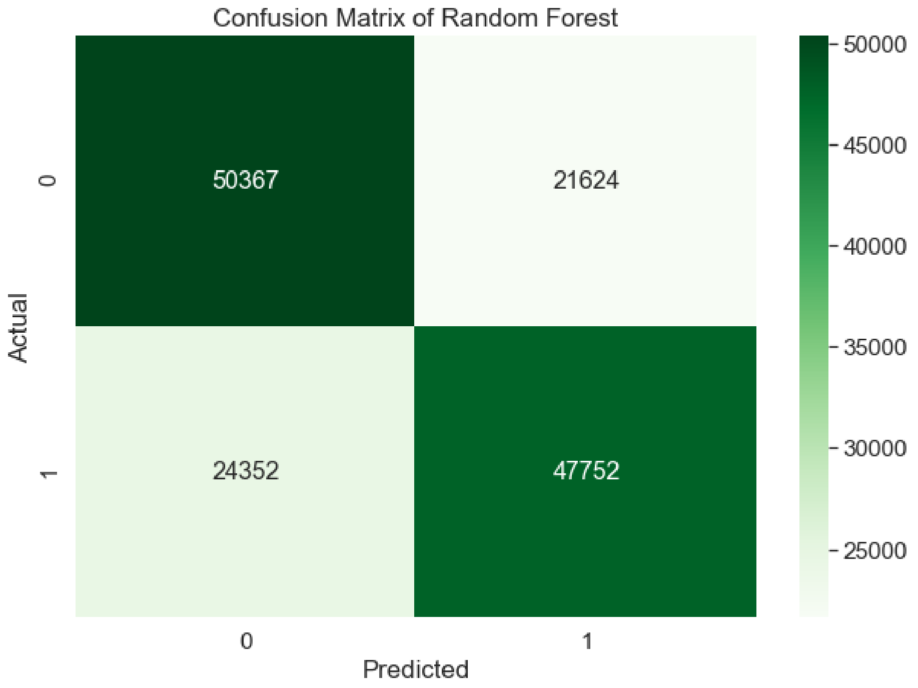


**Figure 9: ROC Curve for Random Forest- Downsampling**

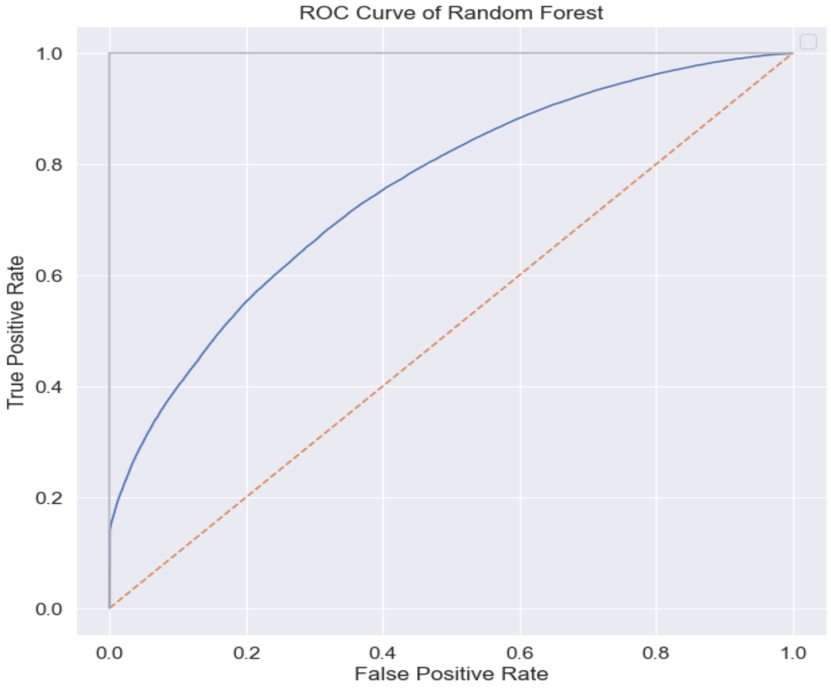


**Table 7: Classification report of LightGBM**

| Dataset | Accuracy | Class | Precision | Recall | Weighted F1 |
|---|---|---|---|---|---|
| **Downample** | 69% | 0 | 69% | 72% | 70% |
| | | 1 | 71% | 67% | |
| **Upsample** | 86% | 0 | 87% | 84% | 86% |
| | | 1 | 84% | 88% | |

**Figure 10: Confusion Matrix for LightGBM-Upsampling**

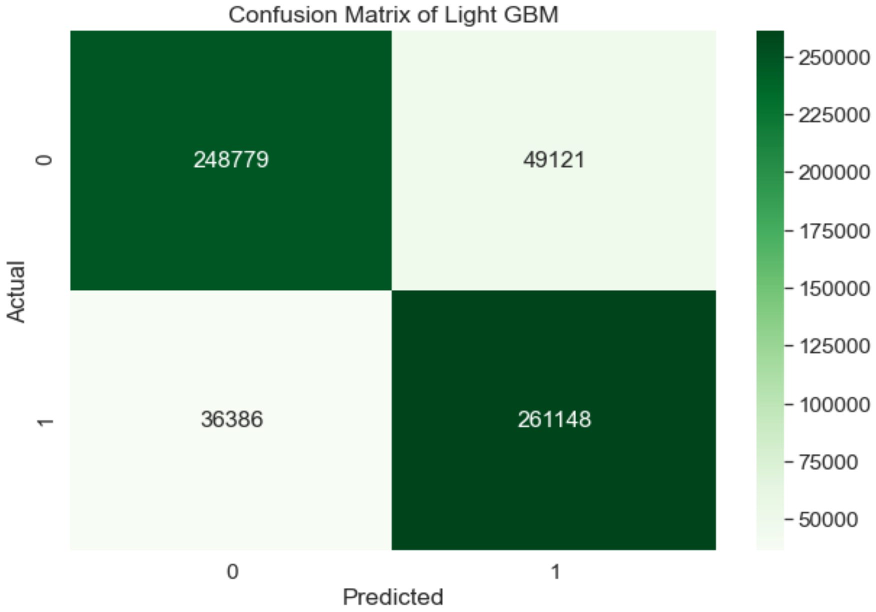


**Figure 11: ROC Curve for LightGBM- Upsampling**

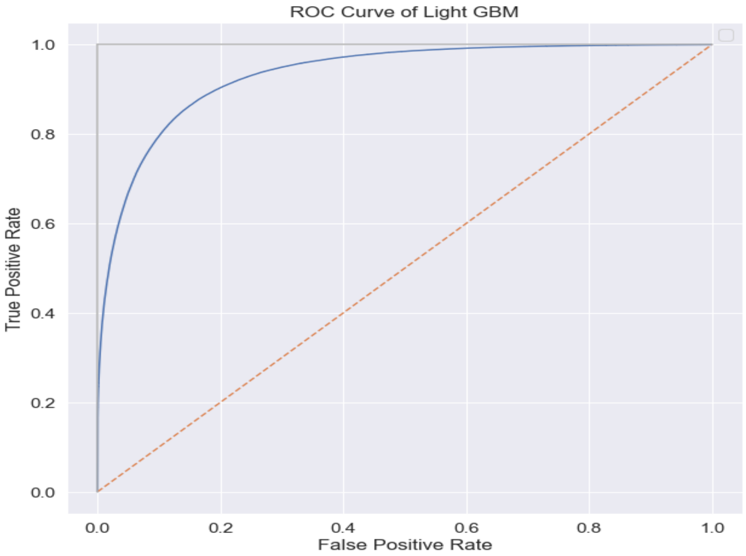


**Figure 12: Confusion Matrix for LightGBM-Downsampling**

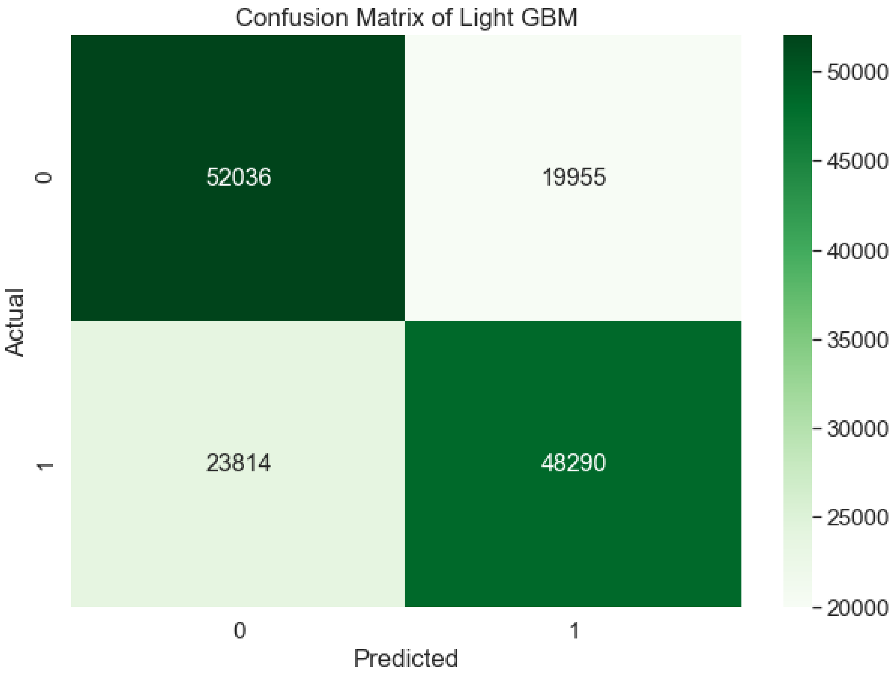


**Figure 13: ROC Curve for LightGBM-Downsampling**

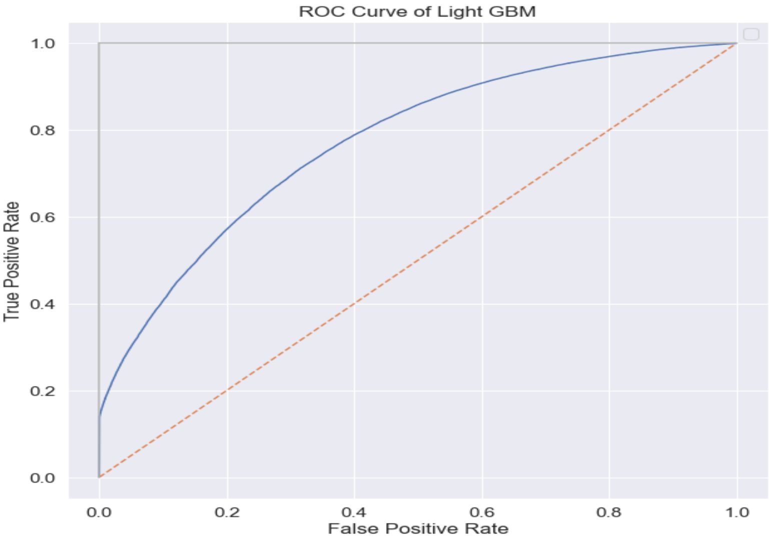
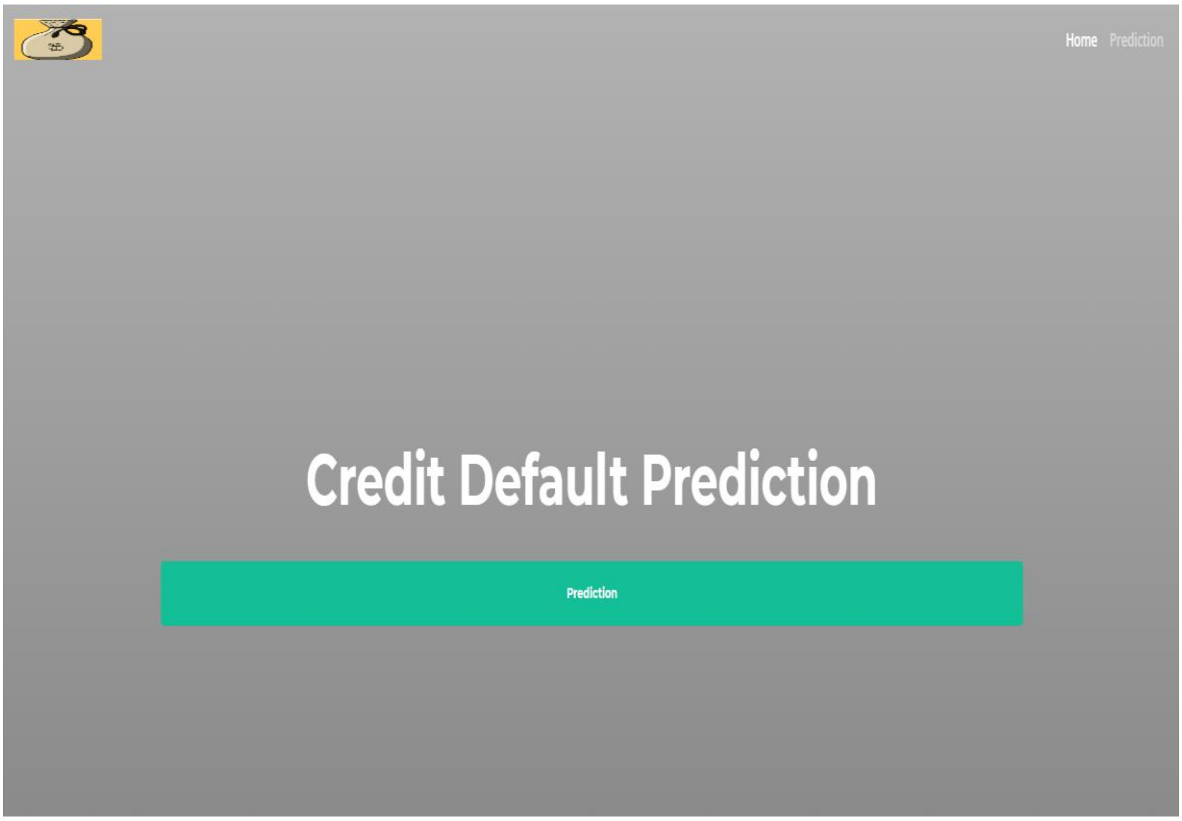


**Table 8: Cross –Validation Performence**

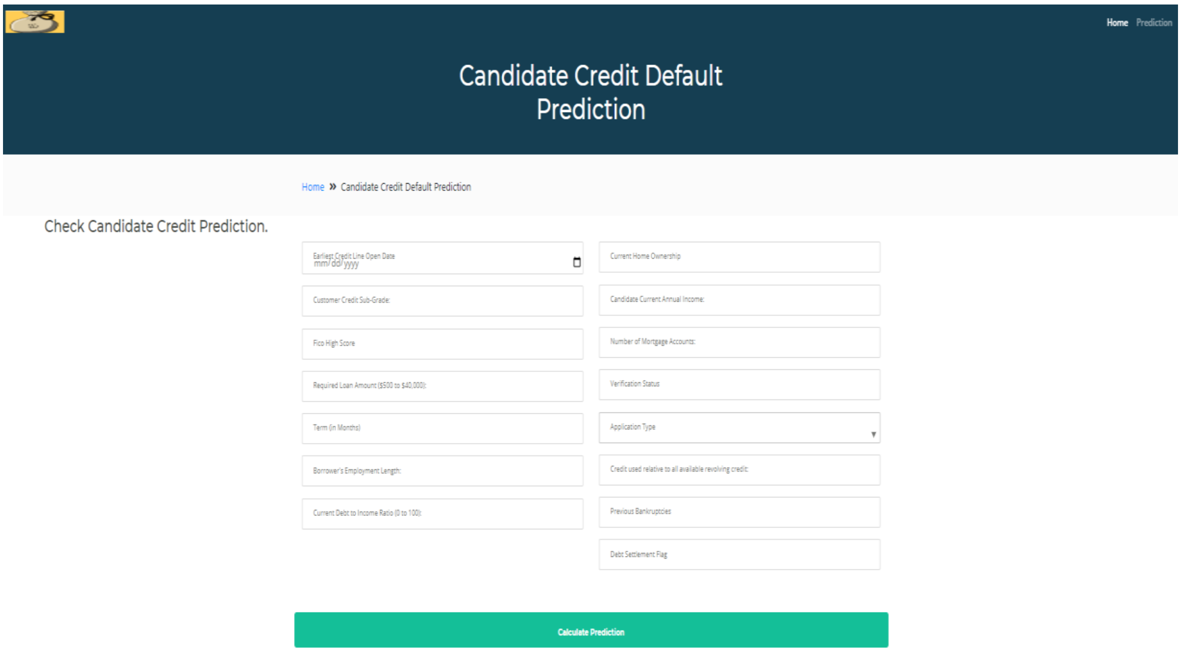
| k-fold Cross Validation | | | |
|---|---|---|---|
| Upsampling | **Evaluation** | **Random Forest** | **LightGBM** |
| | Folds | 10 | 5 |
| | Accuracy_mean | 0.75 | 0.85 |
| | Recall_mean | 0.74 | 0.85 |
| | F1 Score_mean | 0.75 | 0.85 |
| Downsampling | **Evaluation** | **Random Forest** | **LightGBM** |
| | Folds | 5 | 5 |
| | Accuracy_mean | 0.67 | 0.69 |
| | Recall_mean | 0.67 | 0.69 |
| | F1 Score_mean | 0.68 | 0.77 |

# Appendix IV: Screenshots of the website.

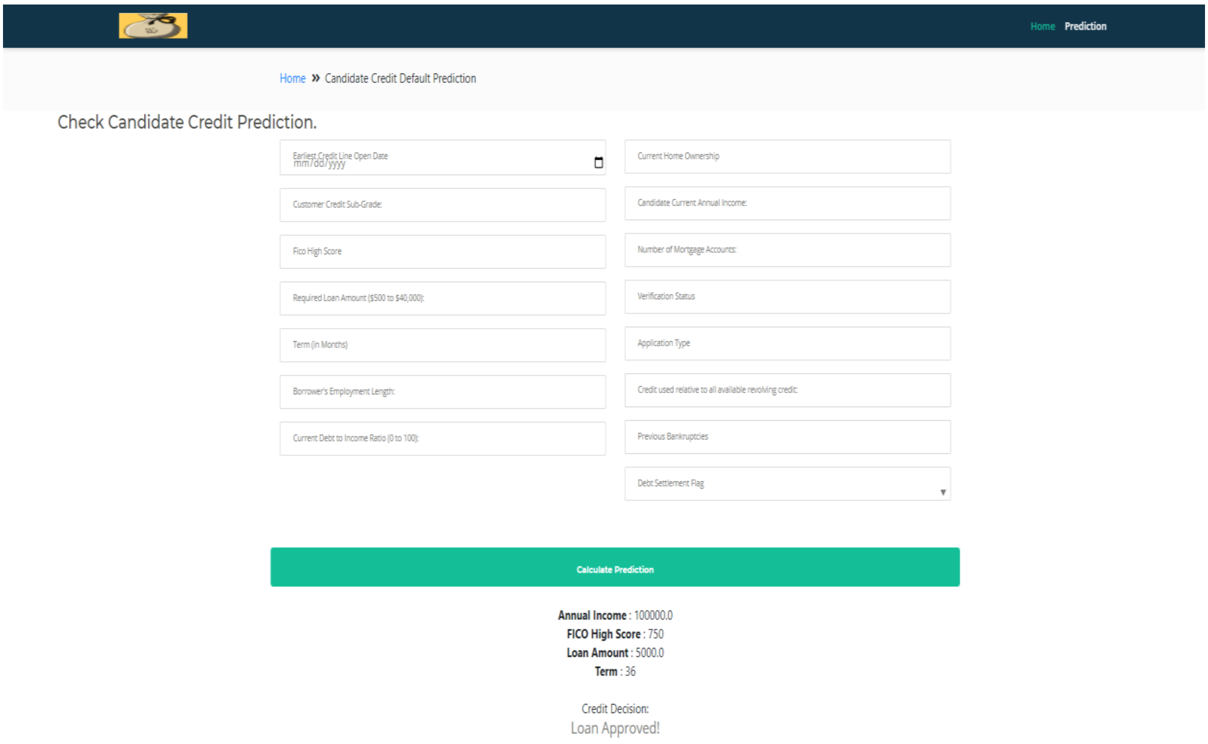
## Figure 14: Website Landing Page



## Figure 15: Website Prediction Page


Home
Prediction
Candidate Credit Default Prediction
Home » Candidate Credit Default Prediction
Check Candidate Credit Prediction.
Earliest Credit Line Open Date
mm/dd/yyyy
Current Home Ownership
Customer Credit Sub-Grade:
Candidate Current Annual Income:
Fico High Score
Number of Mortgage Accounts:
Required Loan Amount ($500 to $40,000):
Verification Status
Term (in Months)
Application Type
Borrower's Employment Length:
Credit used relative to all available revolving credit:
Current Debt to Income Ratio (0 to 100):
Previous Bankruptcies
Debt Settlement Flag
Calculate Prediction

## Figure 16: Website Prediction Page with Loan approved decision

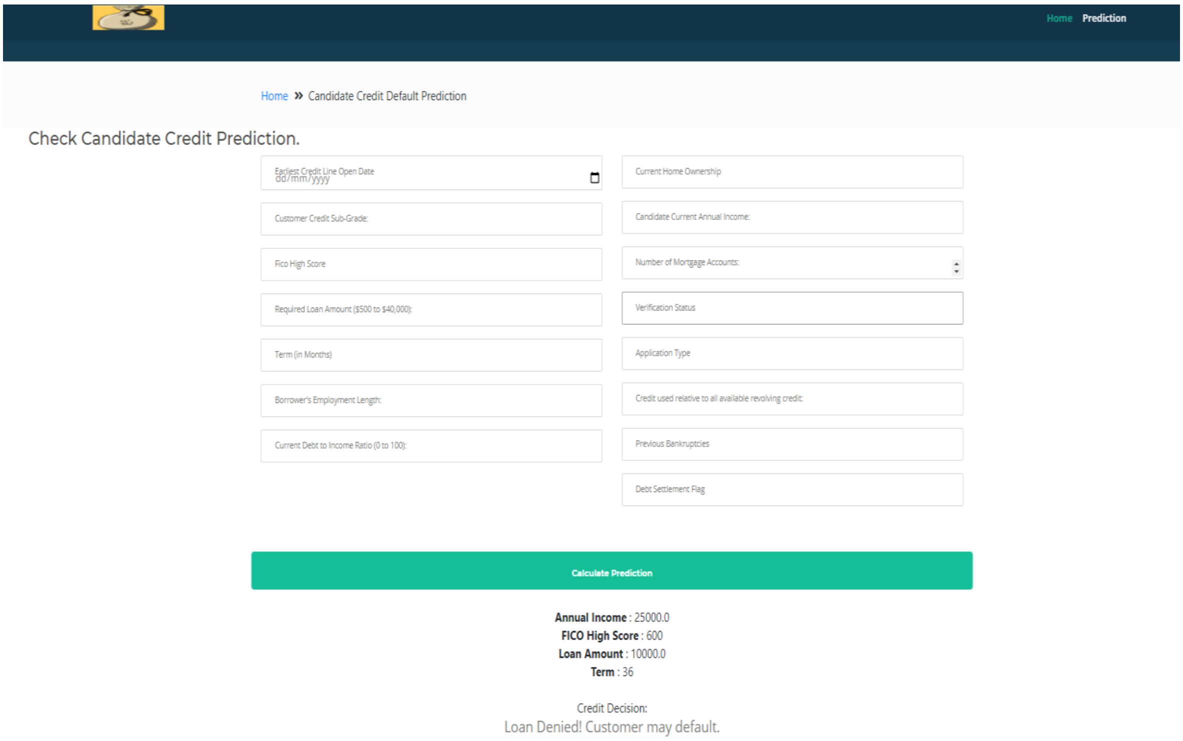

Home
Prediction
Home » Candidate Credit Default Prediction
Earliest Credit Line Open Date
mm/dd/yyyy
Current Home Ownership
Customer Credit Sub-Grade:
Candidate Current Annual Income:
Fico High Score
Number of Mortgage Accounts:
Required Loan Amount ($500 to $40,000):
Verification Status
Term (in Months)
Application Type
Borrower's Employment Length:
Credit used relative to all available revolving credit:
Current Debt to Income Ratio (0 to 100):
Previous Bankruptcies
Debt Settlement Flag
Calculate Prediction

Check Candidate Credit Prediction.

**Annual Income** : 100000.0
**FICO High Score** : 750
**Loan Amount** : 5000.0
**Term** : 36

Credit Decision:
Loan Approved!

## Figure 17: Website Prediction Page with Loan denied decision.


Home
Prediction
Home » Candidate Credit Default Prediction
Earliest Credit Line Open Date
dd/mm/yyyy
Current Home Ownership
Customer Credit Sub-Grade:
Candidate Current Annual Income:
Fico High Score
Number of Mortgage Accounts:
Required Loan Amount ($500 to $40,000):
Verification Status
Term (in Months)
Application Type
Borrower's Employment Length:
Credit used relative to all available revolving credit:
Current Debt to Income Ratio (0 to 100):
Previous Bankruptcies
Debt Settlement Flag
Calculate Prediction

Check Candidate Credit Prediction.

**Annual Income** : 25000.0
**FICO High Score** : 600
**Loan Amount** : 10000.0
**Term** : 36

Credit Decision:
Loan Denied! Customer may default.