CREDIT RISK MODELLING SYSTEM

Contents

Abstract 2

Problem Statement 4

Objective 5

Understanding the Dataset: 6

Exploratory Data Analysis 8

Pandas Profiling 14

Feature Engineering 15

Feature Selection 16

Modeling 17

- **Logistic Regression** 17
- **XGBoost** 17
- **Random Forest** 18
- **Decision Tree** 18

Model Evaluation 20

Technology Stack 22

Conclusion 23

Future Scope 24

Abstract

Nowadays there are many risks related to bank loans, especially for the banks so as to reduce their capital loss. The analysis of risks and assessment of default becomes crucial thereafter. Banks hold huge volumes of customer behaviour related data from which they are unable to arrive at a judgement if an applicant can be defaulter or not. Data Mining is a promising area of data analysis which aims to extract useful knowledge from a tremendous amount of complex data sets. In this project we aim to design a model and prototype the same using a data set provided by the company. The model is a Random forest based prediction model that uses the features available after applying Extra Tree Classifier. Prior to building the model, the dataset is pre-processed, reduced and made ready to provide efficient predictions. The final model is used for prediction with the test dataset and the experimental results prove the efficiency of the built model

Introduction:

Loans have always been an important part of people's lives for quite some time now. Each individual has different reasons for borrowing a loan. It could be to buy a dream car or a home, to set up a business, or to buy some products. Even wealthy people prefer taking loans overspending their cash so as to get tax benefits and to keep the cash available for future unexpected and unconventional expenses.

Loans are also as important to Lenders as they are for Borrowers. Almost all Banking Organizations make most of their revenues from the interests generated through loans. However, the caveat here is that the lenders make a profit only if the loan gets repaid. The Lending Organizations are faced with the tough task of analyzing the risk associated with each client. Therefore, it is important to identify the risky behaviors of clients and make educated decisions.

We aim to build an end to end Machine Learning model for predicting the Defaulting Risk associated with a borrower.

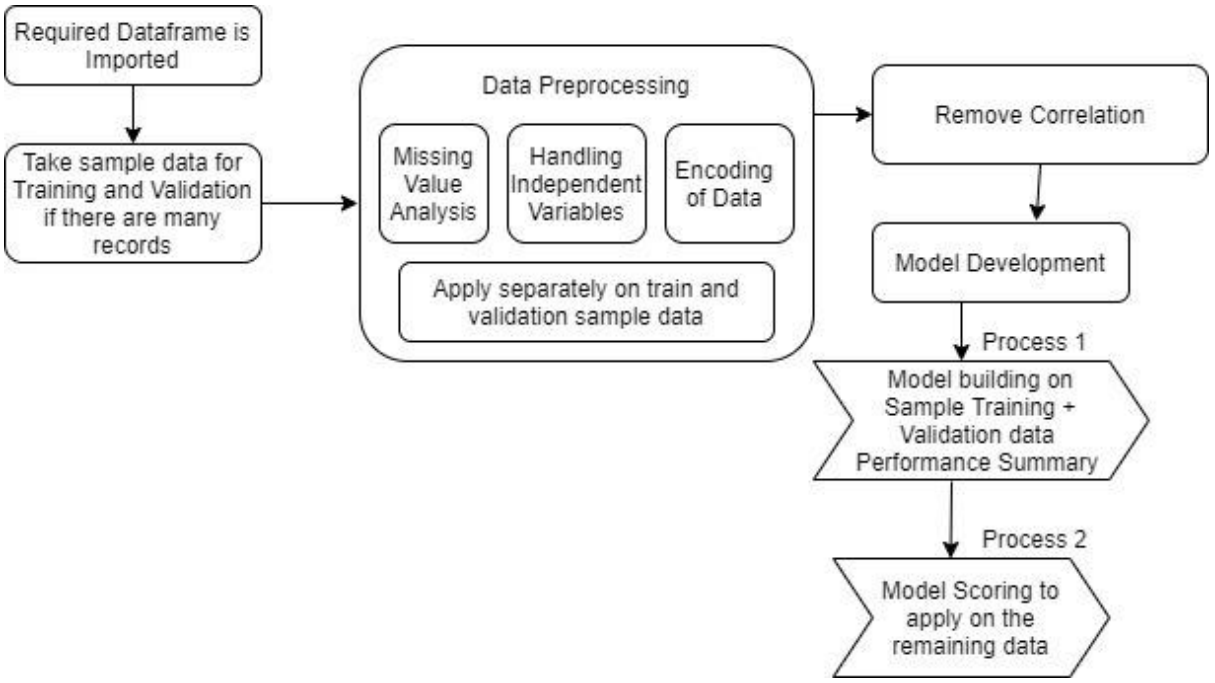
Problem Statement

Customers apply for various kinds of loans in a bank. The branch manager has the responsibility to determine if you should approve a loan application or not. The manager hires you as a data scientist, and asks you to understand and predict if the customer will be able to repay the loans or not. This dataset will help to determine such abilities of each customer. It contains a feature named “Target” where 1 represents customers with payment difficulties (like he/she had late payment more than X days on at least one of the first Y installments of the loan or any previous loan), and 0 for all other cases.

Objective

- To develop a solution focused on developing a clustering algorithm based on unsupervised learning.
- Try various clustering algorithms, and identify the best model for the business scenario and build and fine tune them based on characteristics.
- Deploy the machine learning model using Flask API and pickle files.
- Design a UI for entering model inputs and display results accordingly.

Architectural diagram of the project:



Procedure

Understanding the Dataset:

There are 8 tables of interest in total. Let's take a look at each of those tables below. These descriptions have been provided by the Home Credit Group.

- **application_{train|test}.csv**
    - This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET).
    - Static data for all applications. One row represents one loan in our data sample.

- **bureau.csv**
    - All client's previous credits provided by other financial institutions were reported to the Credit Bureau (for clients who have a loan in our sample).
    - For every loan in our sample, there are as many rows as the number of credits the client had in the Credit Bureau before the application date.

- **bureau_balance.csv**
    - Monthly balances of previous credits in the Credit Bureau.
    - This table has one row for each month of history of every previous credit reported to Credit Bureau — i.e. the table has (#loans in sample * # of relative previous credits * # of months where we have some history observable for the previous credits) rows.

- **POS_CASH_balance.csv**
    - Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample — i.e. the table has (#loans in sample * # of relative previous credits * # of months in which we have some history observable for the previous credits) rows.

- **credit_card_balance.csv**
    - Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample — i.e. the table has (#loans in sample * # of relative previous credit cards * # of months where we have some history observable for the previous credit card) rows.

- **previous_application.csv**
    - All previous applications for Home Credit loans of clients who have loans in our sample.
    - There is one row for each previous application related to loans in our data sample.

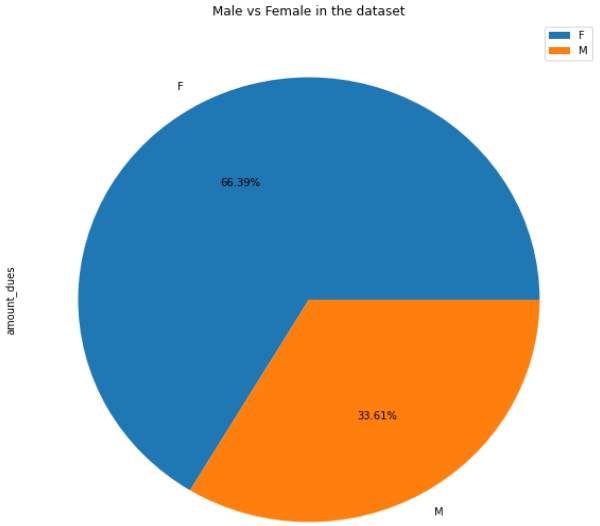
- **installments_payments.csv**
    - Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
    - There is a) one row for every payment that was made plus b) one row each for a missed payment.
    - One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.

Exploratory Data Analysis

- One of the most important and critical parts of Machine Learning is Data Analysis. Without understanding the data, there is no point in building the Machine Learning Models. We know that Feature Engineering is the core of every Machine Learning model, and if we cannot make sense of the data, we would not be able to build the explanatory features, which our models would ultimately use for classification purposes.
- Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often used for statistical graphics and other data visualization methods. Libraries used were matplotlib,seaborn,and plotly.

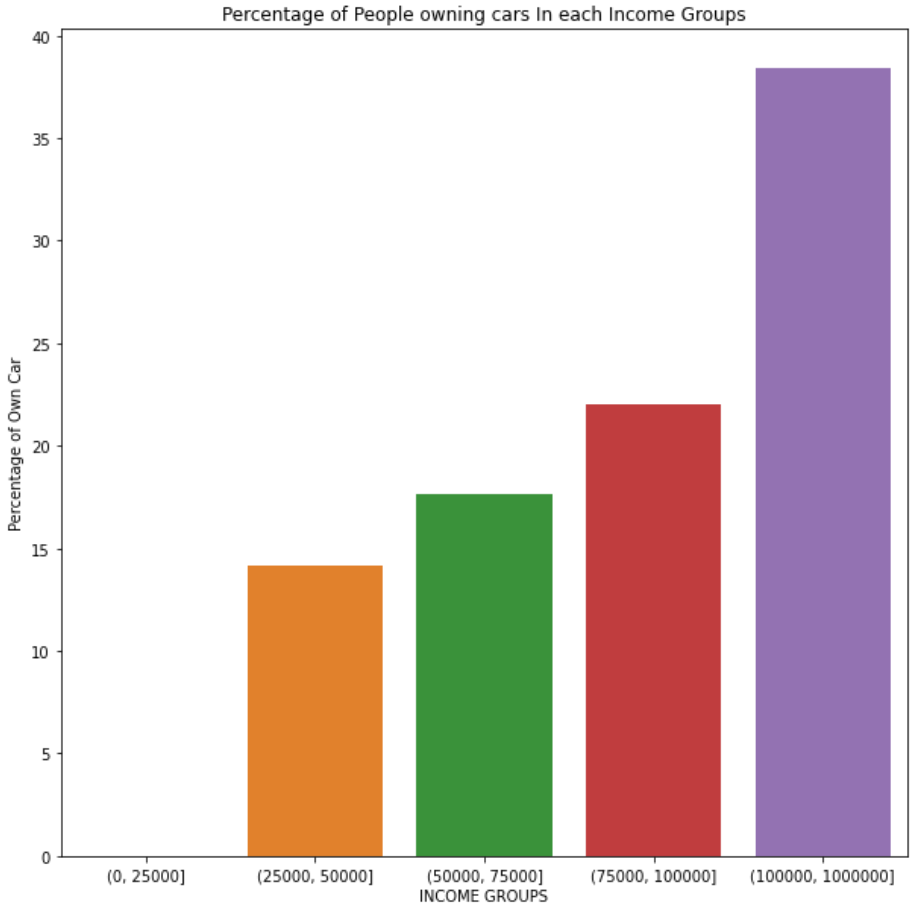
- **Male vs Female Graph**

- We can observe that there are more Female Clients who have applied for the loans and show more dues as compared to Men.
- One of the reasons for this is because the dataset contains more no of female customers.



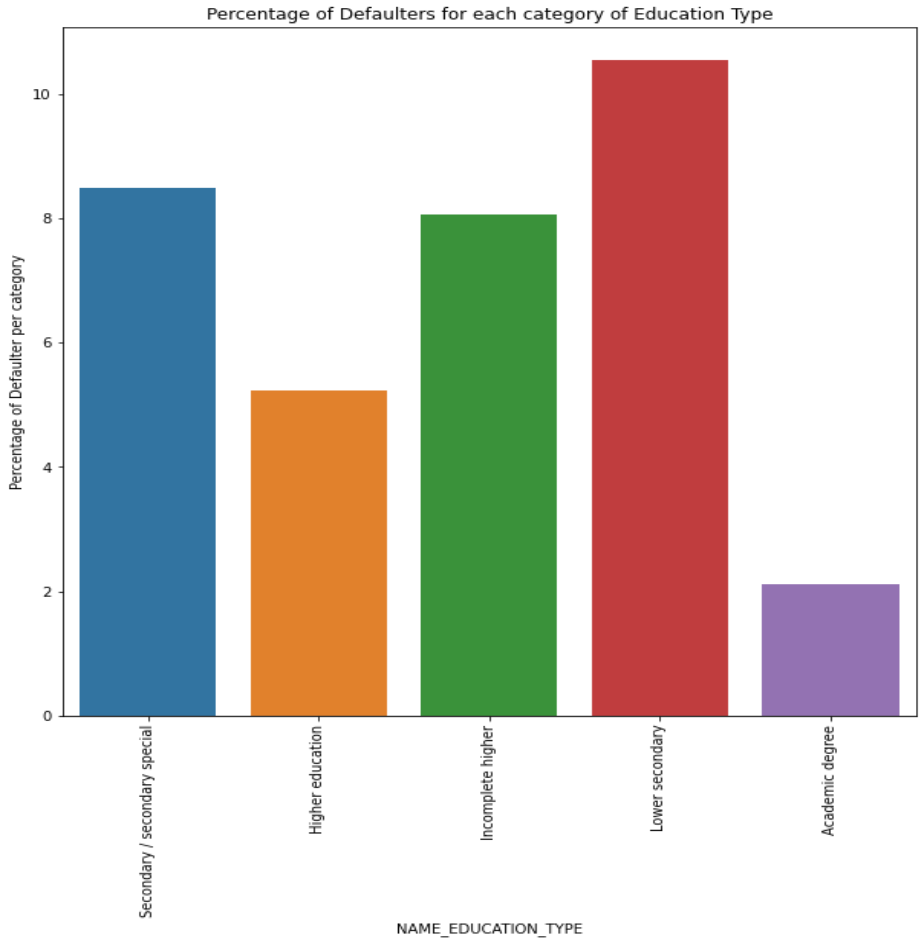
- **Percentage of people owning cars in each income group**

- The graph is between the income group and customers having their own car.
- This graph shows that customers falling in the highest income group have more no.of cars compared to other income groups.



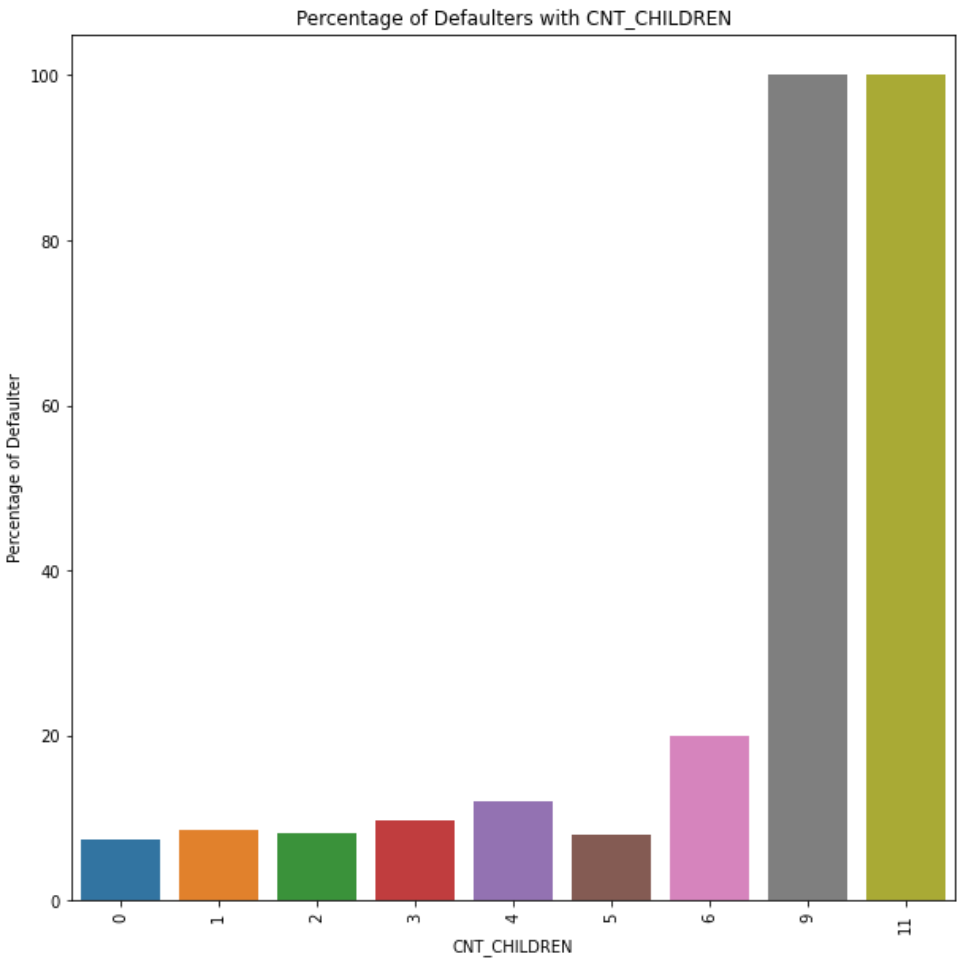
- **Percentage of Defaulters for each Category of Education Type**

- Our graph represent the percentage of defaulter for each category of education type
- So People having the education of Lower Secondary tend to show more defaulting characteristics.
- Whereas the people having an Academic Degree have the least defaulting characteristic



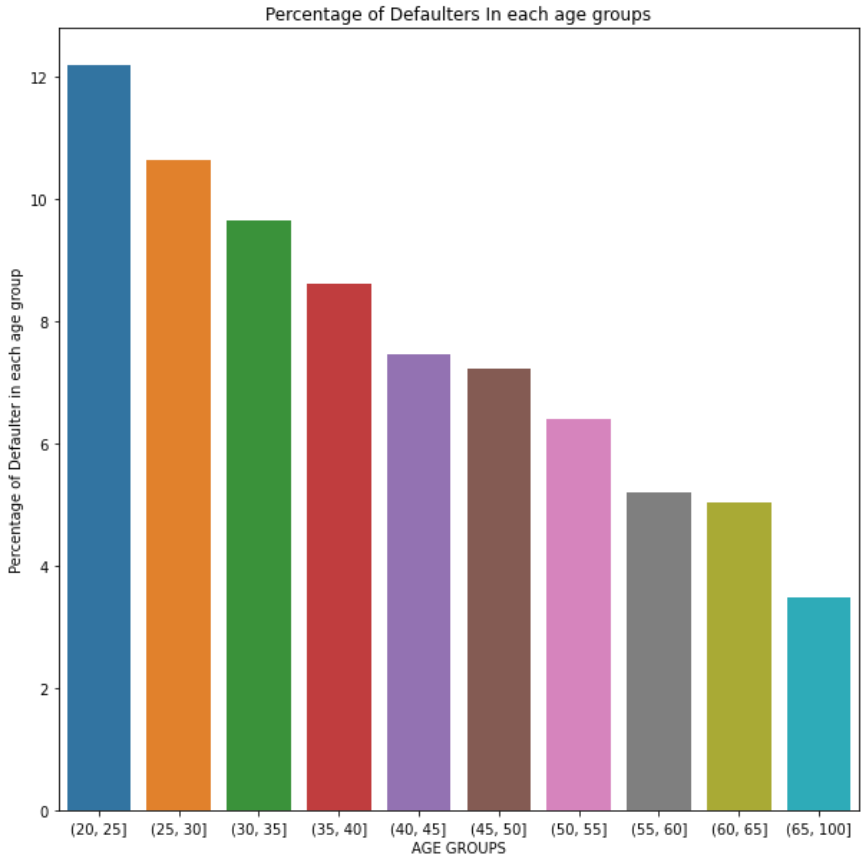
- **Percentage of defaulters With No.of Children**

- Our graph represents the percentage of defaulters with no.of children the customer has so from here we get to know that customers having more no of children have greater defaulting characteristics.
- It is very rare to see families having 9 or 11 childrenS, so we may have only a few outliers of the same.
- The more relevant thing to notice from the graph is that the families that have 0-4 childrens. Customers with 1 & 2 childrens have similar characteristics of defaulting whereas 3 and 4 show a relatively higher tendency of defaulting.



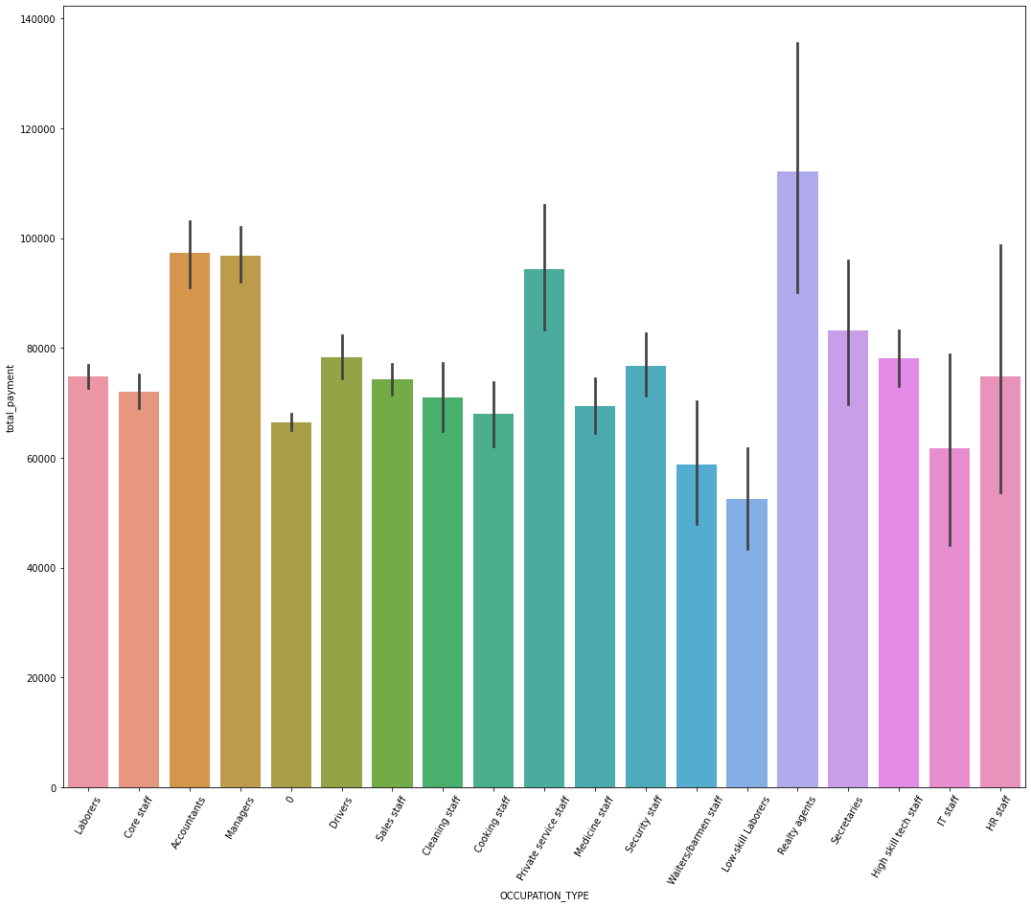
- **Percentage of Defaulters in each Age Groups**

- Our next graph represent the percentage of defaulter in each age group
- So we get to know that customers falling in the age group 20-25 tend to show higher defaulting characteristics. We can also observe that people in older age groups shows less defaulting characteristic



- **Occupation Types of Customers**

- Our last graph represents the occupation types of customers where we get to know that occupation type low skill labourers show least total payment and occupation type reality agent highest total payments.
- People having fixed income tend to pay rather than low income .



Pandas Profiling

- Pandas profiling is an open source Python module with which we can quickly do an exploratory data analysis with just a few lines of code. Besides, if this is not enough to convince us to use this tool, it also generates interactive reports in web format that can be presented to any person, even if they don't know programming.
- In short, what pandas profiling does is save us all the work of visualizing and understanding the distribution of each variable. It generates a report with all the information easily available.

Insights from Pandas Profiling

- FLAG_MOBIL has constant value "1.0"
- FLAG_DOCUMENT_2 has constant value "0.0"
- AMT_DOWN_PAYMENT has constant value "0.0"
- df_index has unique values
- Unnamed: 0 has unique values
- SK_ID_CURR has unique values

Feature Engineering

- Feature engineering is the process of using domain knowledge to extract features from raw data. A feature is a property shared by independent units on which analysis or prediction is to be done. Features are used by predictive models and influence results.
- While creating the LGD column, we have used several features such as total amount, total dues, recovery rate and features used to generate those features. These all columns along with LGD will lead to multicollinearity. Hence, to avoid multicollinearity, we dropped those columns.
- As per the insights from pandas profiling, we dropped 6 columns as they had constant or unique values only.
- In our dataset, we observe 20 columns for documents. Each column represents status whether the client has provided a document or not. So we summed them to create a new column DOCUMENT_SUM. Later with DOCUMENT_SUM, we generated another column FLAG_DOCUMENT_SUM which contains 0 if DOCUMENT_SUM has 0 else 1 for other cases.
- We also observed 6 columns which had a number of inquiries for the past 1 hour, 1 day(excluding 1 hour), 1 week(excluding 1 day),1 month(excluding 1 week), 1 quarter(excluding 1 month) & 1 year(excluding 1 quarter). Thus we calculated the total number of enquiries for the past 1 year by adding them all.
- We did not get relevant information from the data dictionary about floor area avg, basement area avg, etc.. so we decided to drop them.

Feature Selection

- Initially we created 2 dataframes, one with all independent variables and other with a dependent variable to apply to VIF and Extra Tree Classifier.
- Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables. Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable. This ratio is calculated for each independent variable. A high VIF indicates that the associated independent variable is highly collinear with the other variables in the model.
- Extra Trees Classifier is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output it's classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.
- Extra tree classifier implements a meta estimator(An estimator which takes another estimator as a parameter) that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- As VIF takes much time to execute, we decided to use Extra Tree Classifier and proceed.
- From the above result, we chose top 25 features for modelling and visualized them.
- Considering the top 25 features, we generated a new dataframe as the final dataframe.
- Later we created 2 dataframes, one with all independent variables and other with a dependent variable.

Modeling

Logistic Regression

Logistic regression is a linear algorithm whose aim is to find a plane which can classify the points into different classes.

Following are the advantages in using logistic regression for the construction of models:

- The generated model takes into account the correlation between variables, identifying relationships that would not be visible and eliminating redundant variables.
- It takes into account the variables individually and simultaneously.
- The user may check the sources of error and optimize the model. In the same text, the author further identifies some disadvantages of this technique.
- In many cases preparation of the variables takes a long time.
- In the case of many variables the analyst must perform a pre-selection of the more important, based upon separate analyses
- Some of the resulting models are difficult to implement.

XGBoost

XGBoost is an ensemble method which uses boosting intuition to create a strong classifier with low variance and low bias by additive combining the weak classifier having low var and high bias. After every iteration it adds one weak classifier which aims to correct the error made by the previous model.

Following are the advantages in using XGBoost for the construction of models:

- It is Highly Flexible
- It uses the power of parallel processing
- It supports regularization
- It is designed to handle missing data with its in-build features.
- The user can run a cross-validation after each iteration.

Random Forest

Random forest is an ensemble method which uses bagging intuition to create a strong classifier having low variance and low bias by parallelly training many weak classifiers having high variance and low bias on the bootstrapped samples and finally taking a majority vote to predict the output.

Following are the advantages in using Random Forest for the construction of models:

- It can perform both regression and classification tasks.
- A random forest produces good predictions that can be understood easily.
- It can handle large datasets efficiently.
- The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Following are the advantages in using Random Forest for the construction of models:

- Simple to understand and to interpret. Trees can be visualised.
- Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Able to handle both numerical and categorical data. However scikit-learn implementation does not support categorical variables for now. Other techniques are usually specialised in analysing datasets that have only one type of variable. See algorithms for more information.
- Able to handle multi-output problems.

- Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Model Evaluation

Performance measures in machine learning classification models are used to assess how well machine learning classification algorithms perform in a given context. These performance metrics include **accuracy, precision, recall and F1-score**. Because it helps us understand the strengths and limitations of these models when making predictions in new situations, model performance is essential for machine learning.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positives and false negatives are almost the same. Therefore, you have to look at other parameters to evaluate the performance of your model.
Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.
Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class.
Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Model	Precision	Recall	F1 score	Accuracy
Logistic Regression	0.67	0.71	0.7	0.68
XGBoost	0.97	0.89	0.93	0.93
Random Forest	0.97	0.91	0.94	0.94
Decision Tree	0.87	0.89	0.88	0.88

Technology Stack

- Colab: Open-source IDE used for development
- Python: Programming language
- Numpy - NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- Pandas - pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.
- Matplotlib - Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
- Seaborn - Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.
- Plotly - The Plotly Python library is an interactive open-source library. This can be a very helpful tool for data visualization and understanding the data simply and easily.
- Sklearn - Its is a software machine learning library for the Python programming language Sklearn have various features such as classification, regression and clustering algorithms including support vector machines,
- PandasProfiling - Pandas profiling is an open source Python module with which we can quickly do an exploratory data analysis with just a few lines of code.
- SMOTETOmek - SMOTETomek is a hybrid method which is a mixture of upsampling and downsampling, it uses an under-sampling method (Tomek) with an oversampling method (SMOTE).

Conclusion

- We were able to load the dataset into Colab and perform relevant analysis and data manipulation for deriving further insights.
- Initial Data Quality measures were performed for making the data useful for further analysis.
- Exploratory Data Analysis was performed in order to obtain visual information on how the data was impacting certain business decisions. Feature correlations were visualized and their distributions were plotted.
- Post EDA, further data quality improvement measures such as Standardization, and dimensionality reduction techniques were implemented.
- Feature selection techniques such as Variance Inflation Factor and Extra Tree Classifier were used to determine the top 25 best features from the dataset.
- Multiple prediction algorithms such as Logistic Regression, XGBoost, Random Forest, Decision Tree were implemented in order to finalize the best possible classification algorithm based on precision, recal, F1 score and accuracy.

Future Scope

- We can implement Sequential Forward Feature Selection for selecting the best set of features.
- We can achieve an even better score by performing Stacking of diverse base classifiers which would train on different sets of features and give us very strong results.
- We can design an interactive user interface where concerned authority can enter the customer ID to detect whether the customer will default or not.