# Topic: Loan Default Prediction for Profit Maximization

## 1.Executive Summary

The data analysis industry is experiencing significant growth, and machine learning models are being utilized not only in new technology sectors, but also in traditional industries. In the following contents, we elaborated the current loan security problem and emphasized the importance of involving machine learning models to help maximize the accuracy of prediction in banking and loan industry. To showcase the model performance, we choose customer profile data from a loan product and applied machine learning models, such as logistic regression, K-nearest neighbor, classification and regression trees, random forest, and XGboost, to predict whether the customer will be a possible defaulter. Furthermore, since there is no perfect raw data, we also applied model calibration techniques, SMOTE sampling, to select the best model for future use by checking the evaluation metrics of confusion matrix, accuracy, recall, precision, F1-score, and AUC score. Based on the outcome of our final evaluation, we have decided to select the random forest model as our ultimate model.

## 2.Define Business Problem

Before applying data analysis and machine learning techniques in banking industries, employees who give the loan and approve the credit card application, they usually check on appliers' personal background and pervious banking history, and then make the decision by using manual understanding processes, which involve them to review and analyzing by themselves about the variety of factors related to applications' creditworthiness and history. Most of the banks have a set of predetermined criteria to help them make lending decisions. However, the criteria is lack of scientific provement, time-consuming, and cannot provide an accurate prediction while some extreme circumstances happened. To solve this problem, machine learning models play an important role in predicting the default possibility more accurately and also improve the efficiency of working flow by analyzing the customer's previous background.

## 3.Industrial Solution

The main business goal for banks and loan companies is to make sure that customers could pay back their check on time and avoid any unexpected payment defaults which can cause property loss. Machine learning models can analyze customer data to identify high-risk customers and develop predictive models to predict payment defaults. These models can process large amounts of data and variables, making it difficult for human employees to discover patterns and trends. By applying risk mitigation and providing suitable plans for high-risk customers, banks can reduce risks and maximize profits. In this project, our group developed several machine learning models (logistic regression, KNN, Random Forest, CART, XGboost) to do customer defaultor checking simulation with a bank customer's dataset.

## 4. Analysis

### 4.1 Data Preprocessing & EDA

**Dataset Summary:** The dataset consists of 252,000 rows and 13 variables, seven of which are categorical variables: 'Married/Single', 'House_Ownership', 'Car_Ownership', 'Profession', 'CITY', 'STATE', and 'Risk_Flag'. The remaining six variables are numeric variables: 'Income', 'Age', 'Experience', 'CURRENT_JOB_YRS', 'CURRENT_HOUSE_YRS', and 'Id'. For detailed variable explanation, please see**[Appendix 1].** We have dropped the 'Id' column as it does not have any predictive value for payment default. For modeling purposes, we use 'Risk_Flag' as the dependent variable and the others as independent variables. If 'Risk_Flag' equals 1 , it means the consumer is likely to default on a loan. Otherwises, consumers are likely to pay a loan on time. There are no missing values in the dataset.

**Variables' Distribution and Outliers:** We have checked the frequency distribution for all variables. Regarding categorical variables, 'Married/Single' and 'House_Ownership' have an imbalanced distribution, as well as the dependent variable 'Risk_Flag'. For the 'CITY' variable, most of the classes have fewer than 1,000 inputs. Converting it into a dummy variable would result in a sparse dataset. Therefore, we have dropped the 'CITY' variable and used 'STATE' instead. Furthermore, we have combined the classes in 'STATE' that have fewer than 1,400 inputs into one class named 'other' to avoid a sparse dataset.  Regarding the numeric variables, with the exception of 'CURRENT_JOB_YRS', all the other numeric variables' distributions are not right-skewed or left-skewed based on their summary statistics and distribution histogram. This suggests that they are likely evenly distributed. However, 'CURRENT_JOB_YRS' has a slightly left-skewed distribution. Moreover, we have checked for outliers in all numeric variables using the "z scores > 3" method and found none. Overall, the absence of significant skewness or outliers in the numeric variables indicates that they are suitable for use in machine learning models.

**Scaling for numeric variables and Creating Dummy**: When dealing with a dataset that includes numeric variables with different scales, such as 'Age' and 'Income', the variables with larger scales can have a greater impact on machine learning models. For instance, if we use these variables to predict a person's credit score, the 'Income' variable, which ranges from $10,310 to $9,999,938, would likely have a more significant impact on the model than the 'Age' variable, which ranges from 18 to 80.To remove the scaling effect, we can use the 'StandardScaler' method to scale the numeric variables. This method standardizes the variables to have zero mean and unit variance, making them more comparable and preventing variables with larger scales from dominating the model. After scaling the numeric variables, we can then create dummy variables for any categorical variables in the dataset. This involves converting categorical variables into multiple binary variables, with each variable representing a category. This step is necessary because most machine learning algorithms cannot work with categorical data directly, so we must convert them into a numeric format.

**Prepare dataset for model training and testing**: We split the dataset into two sets - training and testing sets. This is done to evaluate how well the model performs on unseen data. In this case, we use a ratio of 80:20, meaning that 80% of the data will be used for training the model, while the remaining 20% will be used for testing the model.

## 4.2 performance matrix

Before training models, we create a function called "metrics" that includes various performance metrics, such as a confusion matrix, accuracy, recall, precision, F1-score, and AUC score.

**Confusion matrix**: A confusion matrix is a summary of the model's predictive results in a classification problem. It shows the number of correct and incorrect predictions for each class and provides an overall picture of the model's performance.

**Accuracy:** Accuracy is a commonly used metric that measures the proportion of correct predictions made by the model.

**Precision:** Precision measures the ratio of correctly classified positive instances to the total number of instances predicted as positive. It assesses how many of the positive predictions made by the model were correct.

**Recall:** Recall measures the ratio of correctly classified positive instances to the total number of positive instances. In other words, it assesses how many of the positive instances the model correctly identified.

**F1-score**: To compare models with different precision and recall, we use the F1-score, which is the harmonic mean of precision and recall. The F1-score is a balanced metric that considers both precision and recall and is more suitable for evaluating the overall performance of a model.

**AUC:** The AUC (Area Under the Curve) score represents the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example. The AUC score is a measure of the model's ability to distinguish between positive and negative instances, regardless of the classification threshold chosen.

In an ideal scenario, we expect our classification model could achieve high scores across all performance metrics. However, in reality, there is a trade-off between true and false positives when setting the classification threshold.

In our application, we have two types of accounts: those with no default and those with default. To optimize our model's performance, we need to balance Type-I (false positives) and Type-II (false negatives) errors. In classical hypothesis testing, this involves selecting the performance metrics that are most valuable to us based on our objective of improving the accuracy of default predictions. We consider recall rate as the most important performance metric because we want our model to accurately identify default customers when they truly default(in other words, Type-II error is more significant in our case). Thus, we aim to achieve a high recall rate. In scenarios where recall are similar, we consider the F1 score, which is a balance between recall and precision. Finally, we also want to achieve an AUC score as close to 1 as possible.

### 4.3 Method 1- Logistic regression

In this case, our objective is to construct a predictive model to determine if a consumer will default on their payment (by predicting if the variable "Risk_Flag" equals 1 or 0). As this is a binary classification problem, we first employ a logistic regression model, a basic classification model, to address it. However, after conducting feature engineering on the dataset(i.e.creating dummy), we are left with 87 predictors. Therefore, it is essential to perform dimensionality reduction before fitting the logistic regression model. Since the relationship between the predictors and the response variable may be linear or nonlinear, we will apply both LASSO and PCA logistic regression to determine which model yields better performance.

For PCA logistic regression, we initially determine the number of PCA components required to explain 80% of the feature variance(use 12 out of 87 components). Next, we employ the selected PCA components to fit the logistic model. Subsequently, we generate the ROC curve and identify the optimal threshold value by maximizing Youden's J statistic (TPR - FPR). Lastly, we utilize this threshold value to classify loan default probabilities as either class 0 or 1 within the "Risk_Flag" column. Lasso logistic regression analysis has revealed that certain variables have a significant impact on the classification of consumers likely to default on their loan payments.

The original logistic regression model yielded a recall score of 0.55, an F-1 score of 0.22, and an AUC score of 0.55. After applying LASSO regularization, the logistic regression model improved its recall score to 0.65, its F-1 score to 0.23, and its AUC score to 0.56. On the other hand, applying PCA to the logistic regression model resulted in a recall score of 0.58, an F-1 score of 0.23, and an AUC score of 0.55.

## 4.4 Method 2- Tree-based models

**Decision Tree**: Decision tree models are a popular choice for their flexibility in handling both classification and regression problems. They are particularly effective for datasets that exhibit nonlinear relationships between features, as is the case with our dataset. Furthermore, decision trees can automatically detect and model interactions between features. However, decision tree models can be prone to overfitting and instability, which is why we opted to use a random forest model to improve the robustness of our predictions. The original decision tree model yielded a recall score of 0.56, an F-1 score of 0.54, and an AUC score of 0.74.

**Random Forest**: Random forest models consist of an ensemble of multiple decision trees,with each tree being independently trained on a random subset of the data. The final prediction is an average of the individual tree predictions, which helps to reduce the variance and improve the overall accuracy of the model. In our case, the random forest model provides a probability estimate for the likelihood of default, allowing us to make informed decisions about risk management. The original random forest model yielded a recall score of 0.53, an F-1 score of 0.56, and an AUC score of 0.74.

**XGBoost**: In addition to the random forest model, we also chose to use the xgboost algorithm, which is an extension of the decision tree and random forest models. Xgboost is known for its speed, scalability, and ability to handle large datasets with high-dimensional features. It has been shown to perform well in a wide range of machine learning tasks and is particularly useful in cases where accuracy and efficiency are both important considerations. The original XGBoost model yielded a recall score of 0.13, an F-1 score of 0.22, and an AUC score of 0.56.

## 4.5 Method 3- K-Nearest Neighbors(KNN)

KNN model aims to classify the customers with similar attributes and history by applying a classification algorithm model with the categorical response variable, which will simply present the likelihood if a customer defaults or not. We trained the KNN model with the number of neighbors equal to 5 which means that the model will consider the 5 closest neighbors to each data point when making predictions by assigning class labels to each point based on the majority class. The original KNN model yielded a recall score of 0.51, an F-1 score of 0.53, and an AUC score of 0.73.

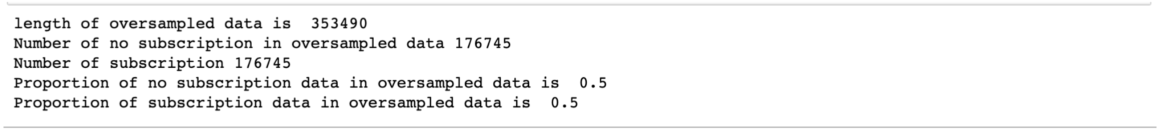
## 4.6 Models Evaluation and Comparison Summary

We have summarized the performance of models in **[Table 1]** and observed that the recall scores are consistently low. We suspect that this may be due to the imbalance in the data we used to train the models.

| Model | Tuning Method | Recall | F1-score | AUC | Accuracy | Precision |
|---|---|---|---|---|---|---|
| Logistics Regression | Logistics | 0.56 | 0.23 | 0.55 | 0.54 | 0.14 |
| | Lasso | 0.65 | 0.23 | 0.56 | 0.48 | 0.14 |
| | PCA | 0.59 | 0.23 | 0.55 | 0.51 | 0.14 |
| Decision Tree | Not Prune | 0.56 | 0.54 | 0.74 | 0.88 | 0.52 |
| | Pruned | 0.1 | 0.1 | 0.52 | 0.57 | 0.88 |
| Random Forest | - | 0.53 | 0.56 | 0.74 | 0.9 | 0.60 |
| XGBoost | - | 0.13 | 0.22 | 0.56 | 0.88 | 0.67 |
| KNN | - | 0.51 | 0.53 | 0.73 | 0.89 | 0.55 |

**Table 1: Original Result for all Models**

To address this issue, we applied the SMOTE sampling method. This technique involves generating synthetic samples from the minority class (in our case, the default class) rather than simply duplicating existing samples. It does this by randomly selecting one of the k-nearest-neighbors and using it to create a new observation that is similar but randomly tweaked. We retrained our models using the SMOTE-sampled data and will compare the results with those from our initial training. And the sampling result as follows**[Figure 1]**:



```
length of oversampled data is  353490
Number of no subscription in oversampled data 176745
Number of subscription 176745
Proportion of no subscription data in oversampled data is  0.5
Proportion of subscription data in oversampled data is  0.5
```

**Figure 1: SMOTE Sampling Result**

After applying SMOTE sampling to the data to address the imbalance problem, we retrained all the models and obtained the results shown in **[Table 2]**. We observed that the performance of all models improved after sampling, which suggests that SMOTE is an effective solution to dealing with imbalance problems. Among all the models, the Decision Tree and Random Forest models achieved the highest recall scores. However, the Decision Tree model's results were not stable, so we decided to use the Random Forest model as our final model. Additionally, the Random Forest model uses the average as the result, which is more persuasive.

| Model | Tuning Method | Recall | F1-score | AUC | Accuracy | Precision |
|---|---|---|---|---|---|---|
| Logistics Regression | Logistics | 0.61 | 0.23 | 0.55 | 0.51 | 0.14 |
| | Lasso | 0.63 | 0.23 | 0.56 | 0.50 | 0.5 |
| | PCA | 0.60 | 0.23 | 0.55 | 0.51 | 0.14 |
| Decision Tree | Not Prune | 0.82 | 0.61 | 0.85 | 0.87 | 0.48 |
| | Pruned | 0.69 | 0.25 | 0.56 | 0.45 | 0.14 |
| Random Forest | - | 0.78 | 0.63 | 0.84 | 0.89 | 0.53 |
| XGBoost | - | 0.76 | 0.54 | 0.82 | 0.84 | 0.42 |
| K-Nearest Neighborhoods | - | 0.68 | 0.49 | 0.76 | 0.82 | 0.38 |

**Table 2: Result for all Models after SMOTE sampling**

## 5.Recommendations

The Random Forest analysis has identified the top 10 features that significantly impact the classification of consumers likely to default on their loan payments. To improve the loan assessment process, we recommend that the bank prioritize these TOP 10 features: income, age, experience, current job years, current house years, and the applicants' state of residence (West Bengal, Uttar Pradesh, Andhra Pradesh, Bihar, and Maharashtra), refer to **[Figure 2].**

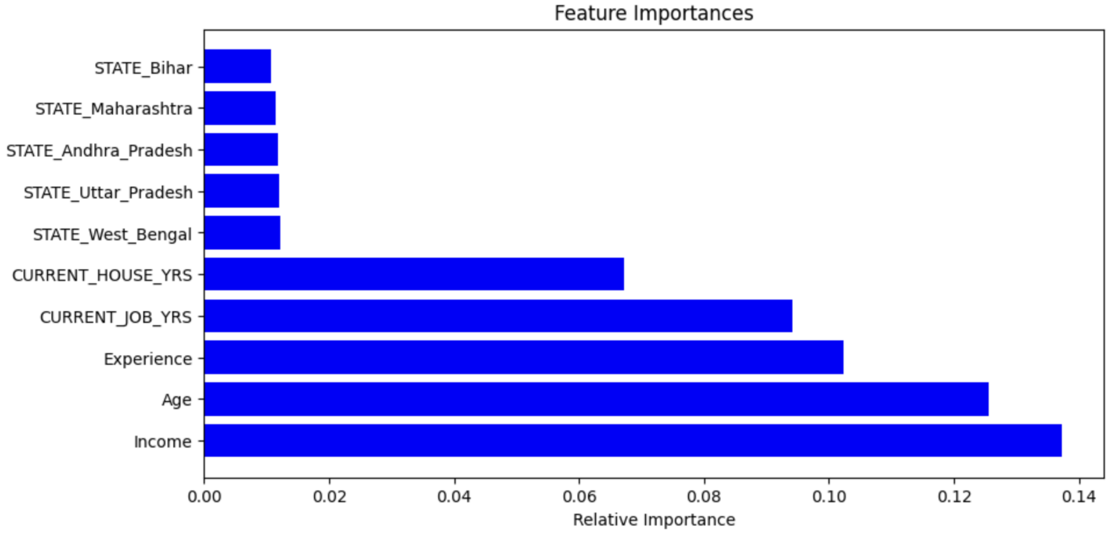


**Figure 2. Importance of components with Random Forest Model**

Key considerations for each feature include evaluating applicants' financial capacity and stability through their income, age, experience, job tenure, and housing situation. Additionally, pay attention to regional trends and economic conditions for applicants from the identified states.

To effectively incorporate these features into the loan assessment process, the bank should:

1) Establish a standardized procedure for evaluating each feature across applicants.
2) Implement a scoring system or risk assessment matrix that incorporates the top features identified by the Random Forest model to quantify applicants' risk.
3) Set thresholds for each feature to identify high-risk applicants requiring further scrutiny or risk mitigation. develop customized risk mitigation strategies for high-risk customers, such

as requiring higher collateral, offering lower credit limits, or implementing stricter repayment schedules. These measures will help reduce the likelihood of loan defaults and protect the bank's interests

4) Utilize insights from the Random Forest model to refine the evaluation criteria for each feature, ensuring accuracy and relevance.
5) Continuously monitor the loan assessment process's performance and adjust feature weights or thresholds as necessary to maintain accuracy and effectiveness.

## 6.Conclusion

In conclusion, we have addressed the issue of imbalance in our dataset by applying the SMOTE sampling method, and observed that it significantly improved the performance of all the models we trained. Among all the models, the Random Forest model was selected as our final model due to its stability and the use of averaging for the result.

Based on the Random Forest analysis, we recommend that the loan product prioritize the top 10 features that significantly impact the classification of consumers likely to default on their loan. To incorporate these features into the loan assessment process effectively, the peoduct should establish a standardized procedure for evaluating each feature, implement a scoring system or risk assessment matrix that incorporates the top features, set thresholds for each feature, and develop customized risk mitigation strategies for high-risk customers. Utilizing insights from the Random Forest model to refine the evaluation criteria for each feature and continuously monitoring the loan assessment process's performance are crucial for maintaining accuracy and effectiveness. These recommendations will help reduce the likelihood of loan defaults and protect the bank's interests.

## Appendix

1. **Dataset Link**
2. **Variable Explanation**

| **Column** | **Description** | **Type** |
|---|---|---|
| income | Income of the user | int |
| age | Age of the user | int |
| experience | Professional experience of the user in years | int |
| profession | Profession | string |
| married | Whether married or single | string |
| house_ownership | Owned or rented or neither | string |
| car_ownership | Does the person own a car | string |
| risk_flag | Defaulted on a loan | string |
| current_job_years | Years of experience in the current job | int |
| current_house_years | Number of years in the current residence | int |
| city | City of residence | string |
| state | State of residence | string |