# Credit Score Classification

## The problem statement: You are working as a data scientist in the global finance company. The company has collected lot of credit related information of many of their customers. The company wants to segregate customers based on this information. The company is currently doing the same process but manually, they want to minimize the manual effort and build an automated smart system.

## Credit score classification is a very important task for finance company because based on credit: Good, Standard and poor; finance company can take decision whether to provide loan to the customer or not. By adopting this strategy finance company can minimize their NPA (Non-performing asset), which is good for their financial growth.

## This is a supervised machine learning and a classification task where we need to classify customers into Good, Standard and poor credit rating.

## The dataset link: https://www.kaggle.com/datasets/parisrohan/credit-score-classification

## The dataset has 28 columns:

### ID, customer\_id, Month, Name, Age, SSN, Occupation, Annual income, Monthly in hand salary, Number of bank account, number of credit card, interest rate, number of loan, type of loan, delay from due date, changed credit limit, number of delayed payments, number of credit inquiries, credit mix, outstanding debt, credit utilization ratio, credit history age, payment of min. amount, Total emi per month, amount invested monthly, payment behavior, monthly balance, credit score

## The project is divided into following steps, the steps are explained in the details:

## Importing libraries

### In this part of the project required libraries such as NUMPY, PANDAS, SCIKITLEARN, MATPLOTLIB, SEABORN etc. are loaded into the python notebook.

## Importing the data

### Here, the csv data is loaded from the local computer path

## Data understanding

### After the data is loaded into the python notebook, the data needs to be understood before going to next steps.

### Dataset shape: 100000 rows and 28 columns

### Unique value count in each categorical columns

### Datatype of each column: There are more than one datatypes like: Object, Integer, Float

### Statistics of each column using describe function

### Checking for missing values: There are missing values in Name, Monthly in hand salary, type of loan, no. of delayed payments, no. of credit inquiries, Credit history age, Amount invested monthly columns.

### This step only involves understanding. There are no modifications made to the dataset during this step.

## Data cleaning and feature engineering

### This is very crucial stage of the project because quality of the data passed to next stage will affect the performance of machine learning model.

### Dropping duplicate records

### Dropping id column because all unique values, dropping Month column because test set has all the different months compared to training set and this is not a time-series problem. Also dropping Name and SSN columns

### There are many wrong/garbage entries in the dataset. For example, Age has -500, 8698 entries, no. of bank account has -1 entry, also negative value entries in delay from due date, number of delayed payments, Charged credit limit, number of loan.

### Before handling missing values, we need to handle outliers, otherwise mean will have high effect while imputing missing values.

### Outlier handling: When we try to remove outlier from the data- the remaining data has only 31% size of the original dataset. To resolve this issue, instead of removing the outliers they are declared as missing values in each column.

### Missing values handling: For handling missing values Median is used for numerical columns because there are few numerical columns which has only discrete values so using mean was impractical as it led to continuous values.

### There are many unique values in the type of loan categorical column, hence whole column is dropped.

### Credit history age is a numerical column but in string format. Ex. 22 years and 1 month. So using string method 22 and 1 are separated then 22 is multiplied with 12 and added to months and at last again whole result is divided by 12 to get credit history age in fraction format.

### Credit mix has some entries specified as “\_\_\_\_\_\_” so that’s replaced with “Other”

### Payment of min. amount has three unique values YES: 52326 entries, NO: 35667 entries, NM: 12007. Most of the columns in the dataset are balanced. Here it looks like “NM” is wrongly entered instead of “NO” because adding NO and NM we get 47674 which is near to 52326 (YES category entries). So replaced ALL “NM” with “NO”

### Customer\_id is dropped.

### In the occupation column there is entry as “\_\_\_\_\_” so replacing it with “Blank”

### Payment behavior column has one category as “!@9#%8” so replacing it with “Blanks”

## Data preprocessing

### Label encoding credit score target columns as 0 ,1 and 2.

### Although the target column is not balanced (Standard:53174, Good:17828, Poor:28998) it is not highly imbalanced like in range of 1:10. So target balancing is not carried out.

## EDA

### The average age is comparatively less for POOR credit score

### The average annual income is also comparatively less for POOR credit score

### Same goes for monthly in hand salary, amount invested monthly and monthly balance

### Interest rate, No. of bank account, No. of credit card, No. of loans, Interest rate, Delay from due date, Outstanding debt, Emi per month is comparatively higher for POOR credit score.

### Credit mixed standard has highest positive correlation with Credit score and Credit mix good has highest negative correlation with Credit score.

## Feature selection

### There are many features left after feature engineering. Few of them may be useful so feature selection is a must process before feeding the data to the machine learning model.

### Select k best method is used to identify feature importance of each feature.

### Top 14 features are selected for further use in training the model.

## Finding a good model using cross validation

### Due to big size of data cross validation takes more time hence, all time good performer i.e. Random forest classifier is chosen for training.

## Hyper parameter tuning of good model

### First the model is trained on default values of it and checked for performance

### In next stage randomized search CV is used to find good (not best) values for hyper parameters.

### Again due to big size of data the Randomized search CV took a bit more time. Very few combinations were tried hence the found hyper parameters gave result poor than default values for hyper parameters.

### Hence model is kept same as before which had default values for hyper parameters.

## Model evaluation

### The final testing results:

#### Accuracy: 78%

#### Precision macro avg.: 77%

#### Recall macro avg.: 78%

## 