CREDIT SCORE CLASSIFICATION

(Applied Machine Learning- Final Report- Group-3)

<u>Team Members GROUP -3 :</u>

Yamini Nathani – YXN230000
Sampath Mylavarapu-SXM220416
Naga Venkata Sai Sunil Parepalli- NXP230024
Varun Kumar Kujala - VXK230013
Saketh Kumar Dachepally- SXD230077
Venkata Sai Lakshmi Dedeepya Nekkalapudi - VXN230012

TABLE OF CONTENTS-

OBJECTIVE

The company has accumulated a wealth of credit-related information over the years and now aims to streamline its efforts by building an intelligent system to classify individuals into credit score brackets. Machine learning techniques, the system will continuously learn and adapt to evolving credit trends and customer behaviours, ensuring its long-term effectiveness and relevance. This initiative underscores the company's commitment to innovation and staying at the forefront of technological advancements in the financial industry. This project will leverage machine learning techniques to achieve this goal, ultimately reducing manual efforts and improving efficiency.

METHODOLOGY

DATA DESCRIPTION-

The dataset contains 1,00,000 instances along with 28 attributes related to individuals' credit profiles, including customer ID, age, occupation, annual income, monthly in-hand salary, number of bank accounts, number of credit cards, interest rate, number of loans, type of loan, delay from due date, number of delayed payments, changed credit limit, number of credit inquiries, credit mix, outstanding debt, credit utilization ratio, credit history age, payment of minimum amount, total EMI per month, amount invested monthly, payment behaviour, and monthly balance.

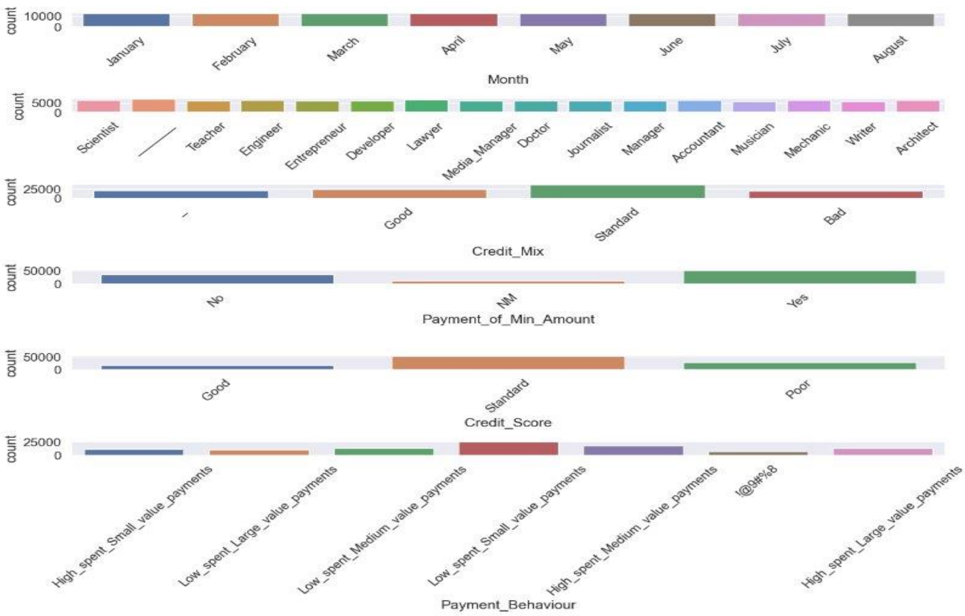
Started the pre-processing by conducting preliminary analysis. Upon initial analysis of the dataset, it's evident that there are numerous missing values and inconsistencies, such as the 'age' and 'annual income' columns being labelled as objects instead of numerical values, necessitating thorough data preprocessing.

```
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 28 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   ID                        100000 non-null  object
 1   Customer_ID               100000 non-null  object
 2   Month                     100000 non-null  object
 3   Name                      90015 non-null   object
 4   Age                       100000 non-null  object
 5   SSN                       100000 non-null  object
 6   Occupation                100000 non-null  object
 7   Annual_Income             100000 non-null  object
 8   Monthly_Inhand_Salary     84998 non-null   float64
 9   Num_Bank_Accounts         100000 non-null  int64
 10  Num_Credit_Card           100000 non-null  int64
 11  Interest_Rate             100000 non-null  int64
 12  Num_of_Loan               100000 non-null  object
 13  Type_of_Loan              88592 non-null   object
 14  Delay_from_due_date       100000 non-null  int64
 15  Num_of_Delayed_Payment    92998 non-null   object
 16  Changed_Credit_Limit      100000 non-null  object
 17  Num_Credit_Inquiries      98035 non-null   float64
 18  Credit_Mix                100000 non-null  object
 19  Outstanding_Debt          100000 non-null  object
 20  Credit_Utilization_Ratio  100000 non-null  float64
 21  Credit_History_Age        90970 non-null   object
 22  Payment_of_Min_Amount     100000 non-null  object
 23  Total_EMI_per_month       100000 non-null  float64
 24  Amount_invested_monthly   95521 non-null   object
 25  Payment_Behaviour         100000 non-null  object
 26  Monthly_Balance           98800 non-null   object
 27  Credit_Score              100000 non-null  object
dtypes: float64(4), int64(4), object(20)
```

DATA CLEANING AND ANALYSIS-

Upon running the 'describe' command, we have the arrangement of numerical data, uncovering anomalous values. Notably, the 'name of the bank account' features instances labelled as '-1', while 'delay from the due date' contains values of '-5'. Furthermore, we encountered extreme values, such as a 5000% interest rate and a staggering 1499 credits. Fortunately, no duplicates were identified in this dataset. However, the dataset does exhibit numerous missing values across multiple variables, highlighting the necessity for thorough data imputation or removal strategies during preprocessing.

In the categorical Variables -After plotting the count plot in the categorical columns, several anomalous values were identified. Notably, the presence of underscore in the 'occupation' column and 'credit mix,' as well as 'NM' values in the 'payment of minimum account,' were observed. To rectify this, these values could be standardized, for example, changing them to 'no' in the case of 'NM' in the payment column.



DATA PREPROCESSING –

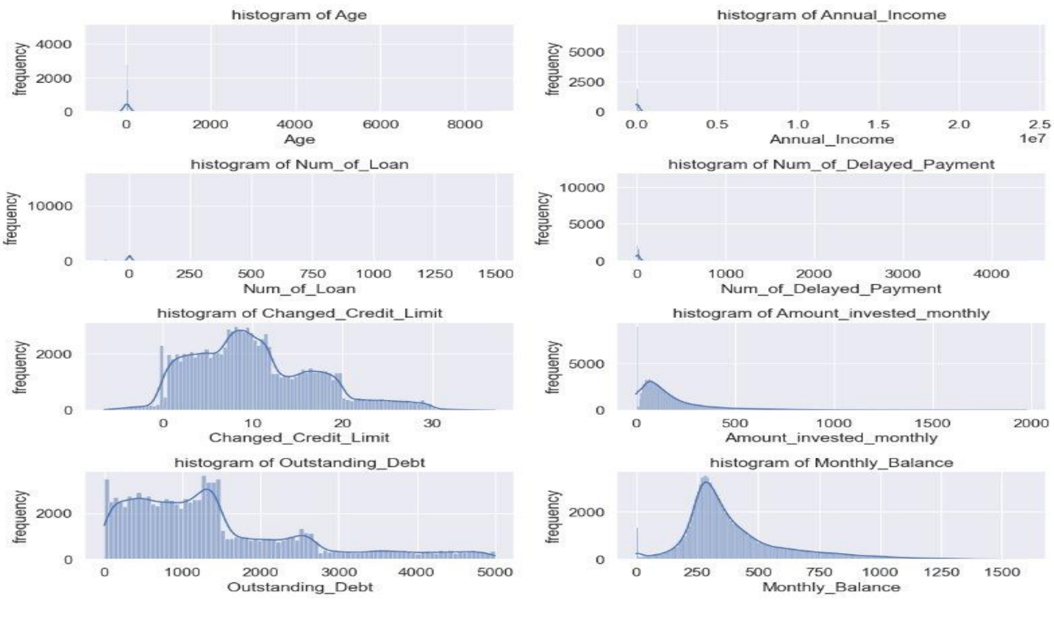
The data preprocessing phase reveals several key issues that need to be addressed for accurate credit score categorization. Firstly, columns like ID, Cust_ID, SSN, and Name are deemed irrelevant for categorization purposes and can be excluded from the analysis. Next, certain numerical attributes such as Age, Annual_Income, Num_of_Loan, Num_of_Delayed_Payment,changed_Credit_Limit,Amount_invested_monthly,Outstanding_ Debt, Credit_Mix, and Monthly_Balance are mislabelled as categorical, requiring proper numerical encoding.

Missing data is a prevalent issue, with numerous entries containing null values. Additionally, columns like Occupation and Credit_ Mix exhibit "__" values that need to be handled appropriately. Anomalies are also present, such as negative values in Num_Bank_Accounts, which must be rectified. Feature engineering is suggested to enhance model performance, including the creation of new features like Credit_History_Age, Payment_of_Min_Amount, Payment_ Behaviour, Credit_Mix, Type_of_Loan, and a more refined Num_Bank_Accounts attribute. These preprocessing steps are vital to ensure data quality and reliability for the subsequent machine learning model development.

Preprocessing categorical columns involves converting non-numeric data into a format suitable for machine learning algorithms, such as one-hot encoding or getting dummies. For instance, in the dataset, columns like Occupation and Type_of_Loan can be transformed using these techniques. One-hot encoding creates binary vectors for each category, while label encoding assigns unique numerical values to categories. This transformation is vital as most machine learning algorithms require numeric input, enabling accurate learning and prediction based on categorical features. Preprocessing numerical columns involved removed invalid values like "_" and correcting the data type labels where necessary.

Later the missing values were worked out by doing KNN imputation on numerical columns and Simple imputation of most frequent values in categorical variables.
Now that the data is cleaned, Power transformation can help make the data more normally distributed. This can be beneficial because many machine learning algorithms assume that the data is normally distributed. It can stabilize the variance across the features.

In the next step of test-train data preparation, the dataset is divided into two subsets: training data and testing data. The training data is used to train the machine learning model, while the testing data is used to evaluate its performance. Typically, the data is split into a training set and a testing set using technique random sampling. This ensures that the model is trained on a diverse set of examples and tested on unseen data, helping to assess its generalization ability. Additionally, data normalization or standardization may be applied to ensure consistency and improve model performance.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 56 columns):
 #   Column                                              Non-Null Count   Dtype
---  ------                                              --------------   -----
 0   Customer_ID                                         100000 non-null  int64
 1   Age                                                 100000 non-null  float64
 2   Annual_Income                                       100000 non-null  float64
 3   Monthly_Inhand_Salary                               100000 non-null  float64
 4   Num_Bank_Accounts                                   100000 non-null  float64
 5   Num_Credit_Card                                     100000 non-null  float64
 6   Interest_Rate                                       100000 non-null  float64
 7   Num_of_Loan                                         100000 non-null  float64
 8   Delay_from_due_date                                 100000 non-null  float64
 9   Num_of_Delayed_Payment                              100000 non-null  float64
 10  Changed_Credit_Limit                                100000 non-null  float64
 11  Num_Credit_Inquiries                                100000 non-null  float64
 12  Credit_Mix                                          100000 non-null  float64
 13  Outstanding_Debt                                    100000 non-null  float64
 14  Credit_Utilization_Ratio                            100000 non-null  float64
 15  Credit_History_Age                                  100000 non-null  float64
 16  Total_EMI_per_month                                 100000 non-null  float64
 17  Amount_invested_monthly                             100000 non-null  float64
 18  Monthly_Balance                                     100000 non-null  float64
 19  Credit-Builder Loan                                 100000 non-null  float64
 20  Personal Loan                                       100000 non-null  float64
 21  Debt Consolidation Loan                             100000 non-null  float64
 22  Student Loan                                        100000 non-null  float64
 23  Payday Loan                                         100000 non-null  float64
 24  Mortgage Loan                                       100000 non-null  float64
 25  Auto Loan                                           100000 non-null  float64
 26  Home Equity Loan                                    100000 non-null  float64
 27  Lawyer                                              100000 non-null  bool
 28  Architect                                           100000 non-null  bool
 29  Engineer                                            100000 non-null  bool
 30  Scientist                                           100000 non-null  bool
 31  Mechanic                                            100000 non-null  bool
 32  Accountant                                          100000 non-null  bool
 33  Developer                                           100000 non-null  bool
 34  Media_Manager                                       100000 non-null  bool
 35  Teacher                                             100000 non-null  bool
 36  Entrepreneur                                        100000 non-null  bool
 37  Doctor                                              100000 non-null  bool
 38  Journalist                                          100000 non-null  bool
 39  Manager                                             100000 non-null  bool
 40  Musician                                            100000 non-null  bool
 41  Writer                                              100000 non-null  bool
 42  Month_August                                        100000 non-null  bool
 43  Month_February                                      100000 non-null  bool
localhost:8889/notebooks/Downloads/credit-score-classification-eda-AML-Project.ipynb

5/3/24, 11:19 AM                                                credit-score-classificat
 44  Month_January                                       100000 non-null  bool
 45  Month_July                                          100000 non-null  bool
 46  Month_June                                          100000 non-null  bool
 47  Month_March                                         100000 non-null  bool
 48  Month_May                                           100000 non-null  bool
 49  Payment_of_Min_Amount_No                            100000 non-null  bool
 50  Payment_of_Min_Amount_Yes                           100000 non-null  bool
 51  Payment_Behaviour_High_spent_Medium_value_payments  100000 non-null  bool
 52  Payment_Behaviour_High_spent_Small_value_payments   100000 non-null  bool
 53  Payment_Behaviour_Low_spent_Large_value_payments    100000 non-null  bool
 54  Payment_Behaviour_Low_spent_Medium_value_payments   100000 non-null  bool
 55  Payment_Behaviour_Low_spent_Small_value_payments    100000 non-null  bool
```

MACHINE LEARNING MODELS

In the credit score classification, the choice between interpretability and accuracy plays a crucial role in model selection. Decision Tree Classifier is favoured for its interpretability, providing insights into how each feature impacts the credit score categorization.

For datasets with a balanced mix of both categorical and numerical features, Cat boost is an excellent choice due to its efficient handling of categorical data and ability to handle numerical features seamlessly, resulting in accurate predictions.

When dealing with non-linear relationships between variables, XGBoost (Extreme Gradient Boosting) shines with its capability to capture complex interactions and patterns in the data. So does SVM (Support Vector Machines) in handling non-linear relationships, providing flexibility and scalability for modelling intricate relationships within the dataset.

Ensemble techniques, such as Bagging, Extra Trees, Random Forest, and Hist Gradient Boosting, are chosen to harness the strengths of multiple models. Bagging reduces variance and overfitting, while Extra Trees further decorrelates base estimators. Random Forest and His Gradient Boosting offer robustness, scalability, and efficient optimization of differentiable loss functions. This ensemble approach aims to strike a balance between interpretability and accuracy, ensuring a reliable and effective credit score classification model. These ensemble models are chosen for their ability to address various challenges in credit score classification, such as handling non-linear relationships, reducing overfitting, and optimizing model performance with a balanced mix of categorical and numerical data. By leveraging the strengths of each model and combining them, we aim to create a robust and accurate predictive model capable of effectively categorizing individuals into credit score brackets.

MODEL – STACKING CLASSIFIER-

The Stacking Classifier is a powerful technique that combines predictions from multiple models to enhance accuracy. Its versatility lies in its ability to adapt to various data types and problem scenarios by selecting different models for different situations. Each model brings its own strengths to the table, and stacking leverages these strengths collectively to generate more precise predictions. Notably, the default Final Estimator for the Stacking Classifier is Logistic Regression, adding a layer of interpretability and reliability to the ensemble predictions.

```
bagging = BaggingClassifier(n_jobs=-1)
extraTrees = ExtraTreesClassifier(max_depth=10, n_jobs=-1)
randomForest = RandomForestClassifier(n_jobs=-1)
histGradientBoosting = HistGradientBoostingClassifier()
XGB = XGBClassifier(n_jobs=-1)

model = StackingClassifier([
    ('bagging', bagging),
    ('extraTress', extraTrees),
    ('randomforest', randomForest),
    ('histGradientBoosting', histGradientBoosting),
    ('XGB', XGB)
], n_jobs=-1)
```

FINDINGS:

COMPARISION OF ACCURACY

Among the classifiers evaluated, the Stacking Classifier emerged as the most accurate with an accuracy score of 0.7996. Given this superior performance, it's prudent to proceed with the Stacking Classifier for generating the classification report. This ensemble method, combining predictions from multiple models, has demonstrated its effectiveness in improving accuracy by leveraging the strengths of individual classifiers. Consequently, using the Stacking Classifier as the final model ensures that our classification report will be based on a robust and reliable framework, leading to more precise and informative results.

CLASSIFICATION REPORT:

The Stacking Classifier, boasting an accuracy of 0.7996, outshone other classifiers, making it the preferred choice for our classification report. This ensemble method combines predictions from diverse models, leveraging their strengths to enhance accuracy. The report showcases the Stacking Classifier's performance across different metrics, revealing a balanced precision, recall, and F1-score across multiple classes. Notably, Class 2 exhibits particularly high precision and recall, highlighting the classifier's reliability for this category. With a micro-average F1-score of 0.81, the Stacking Classifier demonstrates robustness and consistency in generating accurate predictions. Its overall accuracy of 0.7996 underscores its strong predictive capabilities.

```
In [54]:    print(classification_report(y_pred,y_test))

                  precision    recall  f1-score   support

               0       0.80      0.79      0.80      8936
               1       0.82      0.81      0.81     16045
               2       0.74      0.78      0.76      5019

        accuracy                           0.80     30000
       macro avg       0.79      0.79      0.79     30000
    weighted avg       0.80      0.80      0.80     30000
```

CONCLUSION AND INSIGHTS:

The machine learning techniques to create a robust tool for categorizing individuals into credit score brackets. The resulting model offers the company a powerful means to enhance decision-making and streamline credit assessment procedures. By leveraging this model, the company can expect improved accuracy and efficiency in evaluating credit worthiness.

Enhanced Accuracy: Machine learning techniques improve the accuracy of categorizing individuals into credit score brackets.
Efficiency Gains: Streamlined credit assessment procedures lead to faster decision-making and resource savings.
Data-Driven Insights: Advanced analytics enable data-driven decision-making, ensuring fairness and consistency.
Customer Experience: Faster assessments enhance customer experience by reducing waiting times.

CONTRIBUTIONS OF TEAM MEMBERS:

Yamini Nathani	-	Preprocessing
Sampath Mylavarapu	-	Cat boost, Documentation
Naga Venkata Sai Sunil Parepalli	-	Stacking classifier, Documentation.
Varun Kumar Kujala	-	Decision tree classifier, XGB
Saketh Kumar Dachepally	-	PPT, Proposal, Insights
Venkata Sai Lakshmi Dedeepya Nekkalapudi	-	Dataset, EDA