**Credit Score Prediction Project**

**Objective:**

The primary goal of this project is to predict customers' credit scores using various financial and behavioral factors. The credit score is classified into three categories: **Poor**, **Standard**, and **Good**. A neural network (ANN) is employed to perform the classification, and the project involves thorough data preprocessing, model building, and evaluation.

**Dataset Overview:**

The dataset contains various features representing the customer's financial behavior and personal information. Key features include:

* **ID**: Unique identifier for each record.
* **Customer\_ID**: Unique identifier for each customer.
* **Age**: Age of the customer.
* **Annual\_Income**: Annual income in USD.
* **Num\_Bank\_Accounts**: Number of bank accounts held by the customer.
* **Num\_Credit\_Card**: Number of credit cards held by the customer.
* **Interest\_Rate**: Average interest rate on loans or credit cards.
* **Num\_of\_Loan**: Number of loans held by the customer.
* **Outstanding\_Debt**: Total outstanding debt.
* **Credit\_Score**: Target variable indicating customer’s creditworthiness (Poor, Standard, Good).

**Analysis Process:**

**1. Exploratory Data Analysis (EDA):**

* **Data Info and Descriptive Statistics**:
  + The dataset was inspected for missing values, data types, and summary statistics.
  + Anomalies and extreme outliers in features such as **age** and **interest rate** were identified and corrected.
* **Case Normalization and Duplicate Removal**:
  + The dataset was cleaned, including normalization of column names using the **skimpy** library.
  + Duplicate rows were identified and removed.
* **Unique Value Analysis**:
  + Each feature was inspected for unique values. For categorical variables like **Occupation** and **Credit Mix**, their distribution and categories were analyzed.

**2. Data Preprocessing:**

* **Dropping Unnecessary Columns**:
  + Irrelevant columns such as **ID**, **Customer\_ID**, **Name**, and **SSN** were dropped from the dataset.
* **Handling Missing Values**:
  + Several columns contained missing data. Different strategies were applied:
    - For numerical columns like **monthly\_inhand\_salary**, **interest\_rate**, and **num\_of\_loan**, missing values were replaced by median values.
    - For categorical columns such as **payment\_of\_min\_amount** and **credit\_mix**, the most frequent category was used for imputation.
* **Encoding Categorical Features**:
  + Categorical features were encoded using **Label Encoding** or **One-Hot Encoding**, especially for features like **occupation**, **credit\_mix**, and **type\_of\_loan**.
* **Feature Transformation**:
  + **Age**: Cleaned for any inconsistencies and transformed into a numeric format.
  + **Credit History Age**: Converted from a string format into total months using a custom function to standardize the data.
  + **Type of Loan**: Exploded into separate rows for each loan type held by the customer and one-hot encoded.

**3. Data Visualization:**

* Visualizations were created to understand the distribution and relationships in the data:
  + **Countplots**: For visualizing the distribution of categorical features like **Credit Score**, **Occupation**, and **Payment Behavior**.
  + **Histograms**: To analyze the skewness and distribution of continuous variables like **Annual Income** and **Outstanding Debt**.
  + **Correlation Heatmap**: To examine the relationships between numerical features and identify multicollinearity issues.

**Modeling:**

**Train-Test Split:**

* The data was split into training and testing sets with an 80/20 ratio, ensuring stratified sampling to maintain the class distribution of the target variable.
* **Scaling**: The numerical features were scaled using the **MinMaxScaler** to ensure all variables were on a similar scale for better performance of the neural network.

**Artificial Neural Network (ANN) Implementation:**

* **Model Architecture**:
  + The model was built using **TensorFlow** and **Keras**, with the following layers:
    - **Input Layer**: The number of neurons corresponds to the number of features.
    - **Hidden Layers**: Several dense layers with **ReLU** activation, **Batch Normalization**, and **Dropout** to prevent overfitting.
    - **Output Layer**: A dense layer with 3 neurons and **softmax** activation for multi-class classification.
* **Optimization**:
  + The model was compiled with **Adam optimizer** and a learning rate of 0.002.
  + The loss function used was **sparse\_categorical\_crossentropy**, appropriate for multi-class classification.
* **Early Stopping**:
  + An **EarlyStopping** callback was implemented to prevent overfitting by monitoring validation loss. The model stopped training if no improvement was seen for 45 epochs.

**Model Performance:**

**Metrics:**

* **Confusion Matrix**: Used to check the performance of the model across the three credit score categories (Poor, Standard, Good).
* **Classification Report**: Showed precision, recall, and F1-score for each class, providing insights into how well the model performs in predicting each credit score category.

**Evaluation Results:**

* The evaluation metrics on both the **training** and **testing** datasets indicated the model’s ability to generalize well on unseen data.

**Model Saving:**

* The trained model was saved using **TensorFlow's save\_model** functionality for future use.

**Conclusion:**

The project successfully predicted customers' credit scores using a robust artificial neural network model. The data preprocessing steps ensured that the input data was clean and ready for modeling. The performance evaluation, combined with appropriate hyperparameter tuning and early stopping, resulted in a well-trained model that generalizes well to new data.