LendingClub Loan Default Prediction Project

# Problem Statement

You are provided with a subset of the LendingClub dataset, which contains historical data on loans issued by LendingClub, a major US peer-to-peer lending company. Your task is to build a machine learning model that predicts whether a borrower will pay back their loan or default (charge-off), based on the information available at the time of loan application.  
  
The main objective is to use the provided features to predict the 'loan\_status' of each loan. This is a classification problem. You should focus on understanding the data, performing appropriate preprocessing and feature engineering, and building a predictive model using the Keras API.

# Dataset Details

The dataset is a specially prepared version of LendingClub data (not the full Kaggle version), with some extra feature engineering left for you to perform. The data includes the following columns (features):

|  |  |
| --- | --- |
| Feature | Description |
| loan\_amnt | Amount of the loan applied for. |
| term | Number of payments (in months, 36 or 60). |
| int\_rate | Interest rate on the loan. |
| installment | Monthly payment owed if the loan originates. |
| grade | Loan grade assigned by LendingClub. |
| sub\_grade | Loan subgrade. |
| emp\_title | Borrower’s job title. |
| emp\_length | Employment length in years (0-10). |
| home\_ownership | Home ownership status (RENT, OWN, MORTGAGE, OTHER). |
| annual\_inc | Self-reported annual income. |
| verification\_status | Whether income was verified. |
| issue\_d | Month the loan was funded. |
| loan\_status | Target variable – current status of the loan. |
| purpose | Category for the loan request. |
| title | Loan title provided by the borrower. |
| zip\_code | First 3 digits of the borrower’s zip code. |
| addr\_state | State provided in the application. |
| dti | Debt-to-income ratio. |
| earliest\_cr\_line | Month of the earliest reported credit line. |
| open\_acc | Number of open credit lines. |
| pub\_rec | Number of derogatory public records. |
| revol\_bal | Total revolving balance. |
| revol\_util | Revolving line utilization rate. |
| total\_acc | Total number of credit lines. |
| initial\_list\_status | Initial listing status (W, F). |
| application\_type | Individual or joint application. |
| mort\_acc | Number of mortgage accounts. |
| pub\_rec\_bankruptcies | Number of public record bankruptcies. |

The dataset contains approximately 396,000 entries and 27 columns, with a mix of numerical and categorical features. Some columns have missing values.

# Project Tasks

1. Exploratory Data Analysis (EDA):  
 - Understand the distribution of the target variable (loan\_status).  
 - Explore the distributions and relationships of key features (e.g., loan amount, interest rate, employment length).  
 - Visualize correlations between continuous variables.  
 - Identify and handle missing data.  
  
2. Data Preprocessing:  
 - Handle missing values appropriately.  
 - Convert categorical variables to numerical representations (e.g., one-hot encoding).  
 - Feature engineering as needed (e.g., extracting information from dates, grouping rare categories).  
  
3. Model Building:  
 - Split the data into training and testing sets.  
 - Build a neural network classifier using the Keras API.  
 - Select appropriate architecture, activation functions, and loss metrics for classification.  
  
4. Model Evaluation:  
 - Evaluate the model using suitable classification metrics (accuracy, precision, recall, F1-score, ROC-AUC, etc.).  
 - Analyze confusion matrix and discuss model performance.  
  
5. Interpretation and Reporting:  
 - Discuss which features are most important for prediction.  
 - Reflect on the limitations of your model and possible improvements.

# Approach Discussion

- EDA: Start by visualizing the target variable to check for class imbalance. Use histograms, boxplots, and countplots to understand the distribution of numerical and categorical features. Correlation heatmaps can help identify relationships between continuous variables.

- Preprocessing: Address missing values using imputation or by removing rows/columns as appropriate. Encode categorical variables using techniques like one-hot encoding or label encoding. Consider normalizing or standardizing numerical features.

- Modeling: Use Keras to build a neural network suited for binary classification. Decide on the number of layers and neurons, activation functions (e.g., ReLU, sigmoid), and compile the model with an appropriate loss function

- Evaluation: After training, evaluate the model on the test set using classification metrics. Use confusion matrix and ROC curves to interpret results.

- Feature Importance: Use model weights, permutation importance, or SHAP values to discuss which features contribute most to the prediction.

# Additional Notes

- You are encouraged to be creative in your feature engineering and model design.  
- Document your process and findings clearly.