1. Data Loading and Target Setup

This section loads the raw data and defines the main prediction goal.

- Load Data: The large dataset (accepted_2007_to_2018Q4.csv) was loaded into a pandas DataFrame. The initial data had over 2.2 million rows and 151 columns.
- Define the Target: The loan_status column was chosen as the target. It originally had many categories (like 'Current', 'Late', 'Charged Off', etc.).
- Create Binary Target: To make this a simple "yes/no" problem, these categories were mapped to two new classes:
    - 0 = Paid: (Includes 'Fully Paid', 'Current', 'In Grace Period', etc.)
    - 1 = Default: (Includes 'Charged Off', 'Default', 'Late', etc.)
- Clean Target: A few rows (33) missing a loan_status were dropped. This step revealed the data is imbalanced: 86.9% of loans are "Paid" and 13.1% are "Default."

2. Data Cleaning & Feature Reduction

This section cleans up the dataset by removing unnecessary columns.

- Remove Columns with Missing Data: The notebook calculated the percentage of missing values for all 151 columns. Any column missing more than 40% of its data was automatically dropped. This removed 46 columns.
- Remove Irrelevant Columns: Several columns were manually dropped because they weren't useful for prediction:
    - Identifiers: id, url (all were unique values which may not be useful)
    - Redundant: zip_code, emp_title, title

After this, the dataset was reduced to 105 columns.

3. Feature Engineering

Categorical (Text) Features

- Binary Conversion: Columns with 'Y'/'N' values (like debt_settlement_flag) were converted to 1/0.
- Ordinal Conversion: Columns with a clear order were converted to numbers.
    - emp_length (e.g., '< 1 year' became 0, '10+ years' became 11).
    - term (e.g., '36 months' became 36).
- Redundancy: sub_grade was dropped because the grade column already captures its main information.
- Consolidation: In home_ownership, very rare categories ('ANY', 'NONE') were grouped into the 'OTHER' category.
- One-Hot Encoding: All other text columns (grade, purpose, addr_state, etc.) were converted into "dummy" columns. For example, home_ownership was split into new columns like home_ownership_RENT and home_ownership_MORTGAGE, which hold a 1 or 0.

Numerical Features

- Averaging: FICO scores, given as ranges (e.g., fico_range_low and fico_range_high), were averaged into a single new feature: fico_score.
- Imputation (Filling Missing Data):
    - Missing values in most numerical columns were filled with the median of that column.
    - Missing emp_length values were filled with the mode.
- Normalization (Handling Skew): The annual_inc (annual income) column was very skewed. A log transform was applied to it to create a more normal distribution, which helps the model learn.
- Outlier Removal: A small number of extreme outliers were found in the dti (debt-to-income) column. These rows (dti >= 60) were removed from the dataset.

Date Features

- All date columns (issue_d, earliest_cr_line, etc.) were converted into numerical features by extracting the Year and Month into new, separate columns.
- The original datetime columns were then dropped.

4. Model Preparation

- Train-Test Split: The dataset was split into two parts:
    - Training Set (70%): The data the model learns from.
    - Test Set (30%): The data held back to check the model's performance on unseen data.
- Standardization: All numerical features were scaled using StandardScaler. This rescales the data so they all have a similar range, which is essential for an ANN to train properly.

5. ANN Model Training & Evaluation

- Model Architecture: An Artificial Neural Network (ANN) was built using tensorflow.keras. It consists of:
    - An input layer.
    - A hidden layer with 64 neurons (ReLU activation).
    - A hidden layer with 32 neurons (ReLU activation).
    - Dropout layers were added to prevent the model from overfitting.
    - An output layer with 1 neuron (sigmoid activation), which outputs a probability between 0 and 1.
- Training: The model was trained on the training data. EarlyStopping was used to monitor the performance on a validation set and automatically stop training when the model was at its best.
- Evaluation: The trained model was tested on the unseen test set.
    - Test Accuracy: 98.67%
    - F1-Score: 0.946 (This is a strong score, especially for an imbalanced dataset, as it balances accuracy and precision).

- Confusion Matrix: The matrix showed that the model was very good at correctly identifying both Paid and Default loans, with a low number of errors.