Lending Club Loan Data Analysis by Thato Tladi
Course-end Project 3
Description

Create a model that predicts whether or not a loan will be default using the historical data.

Problem Statement:

For companies like Lending Club correctly predicting whether or not a loan will be a default is very important. In this project, using the historical data from 2007 to 2015, you have to build a deep learning model to predict the chance of default for future loans. As you will see later this dataset is highly imbalanced and includes a lot of features that make this problem more challenging.

Domain: Finance

Analysis to be done: Perform data pre-processing and build a deep learning prediction model.

Content:

Dataset columns and definition:
credit.policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
purpose: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").
int.rate: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be more risky are assigned higher interest rates.
installment: The monthly installments owed by the borrower if the loan is funded.
log.annual.inc: The natural log of the self-reported annual income of the borrower.
dti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
fico: The FICO credit score of the borrower.
days.with.cr.line: The number of days the borrower has had a credit line.
revol.bal: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
revol.util: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
inq.last.6mths: The borrower's number of inquiries by creditors in the last 6 months.
delinq.2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
pub.rec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).
Steps to perform:

Perform exploratory data analysis and feature engineering and then apply feature engineering. Follow up with a deep learning model to predict whether or not the loan will be default using the historical data.

Tasks:

1. Feature Transformation

Transform categorical values into numerical values (discrete)

2. Exploratory data analysis of different factors of the dataset.
3. Additional Feature Engineering

You will check the correlation between features and will drop those features which have a strong correlation

This will help reduce the number of features and will leave you with the most relevant features

4. Modeling

After applying EDA and feature engineering, you are now ready to build the predictive models

In this part, you will create a deep learning model using Keras with Tensorflow backend

Task as completed:

- ✓ Loaded necessary libraries and dataset
- ✓ Explored the structure of the data, including head, tail, statistical summary, shape, columns and verified that there are no missing values.
- ✓ Verified that the dataset is balanced, and used oversampling approach to balance the data
- ✓ Exploratory data analysis
  - Plotted charts for distribution of revolving balance
  - Distribution of FICO scores by Credit policy
  - Distribution of FICO scores by Loan status
  - Count of loans by purpose and loan status
  - Explored trend between FICO score and interest rate
  - Explored the trend between not.fully.paid and credit.policy
  - Performed correlaition analysis
- ✓ Deep Learning model
  - Build a deep learning model
  - Model evaluation provided the output of:
  - The model achieved an overall accuracy of approximately 72%, with a precision of 72% and recall of 73% for class 0 (not fully paid), and a precision of 72% and recall of 71% for class 1 (fully paid). Despite a balanced performance between the two classes, there were 683 false negatives, indicating room for improvement in correctly identifying fully paid loans."
  - Model Refinement , by refining the model , Overfitting issue have been reduced by adding in dropout layers
  - By changing the cut-off line to 0.2 from 0.5, we have dramatically brought down the Type 2 error.
- ✓ Saved the model and Scaler