Predicting Bank Loan Approval Based on Risk
(Mid-term Progress Report)

Student names:

- Jinwen “Eddie” Zhao (jinwenz@usc.edu)
- Jackson Crawford (jc37570@usc.edu)
- Cameron Ela (ceela@usc.edu)
- Ravish Kamath (rdkamath@usc.edu)

Project Idea:

We want to determine whether a bank should give loans to a specific client. We can use a Python machine learning program to take in different client factors, such as annual income, default history, grade of the loans, and loan amounts, to make a classification prediction if this client will default or not. If they default, the bank should think twice about giving them loans.

Description of Dataset:

The dataset was obtained through a statistics professor at York University, who provided multiple real-life datasets ranging from bank data to real-estate datasets. We specifically chose a bank loan dataset. We are looking at 5000 loan requests issued by a bank, with each data point having 21 features. The table contains vital information to determine if a bank should loan to clients, such as whether they have defaulted before, how much annual income they had in previous years, how much they are borrowing, their jobs, and the rating of the loan. The dataset is already cleaned and easily interpreted by a human viewer. It has few ambiguous entries and empty values. After a bit of cleaning, the dataset will be ready for Python machine learning.

Project plan:

Our system to predict bank loan approvals can be achieved in 4 phases, whose methodology will be elaborated later:

1. Obtaining the data / Data mining, (Complete)
2. exploratory data analysis (EDA),   (In-progress)
3. the machine learning process, and
4. interpretation of the result.

For the first phase, we have already obtained a sizable dataset from a statistics professor at York University. By glancing at the data, our group members can have an initial idea as to how we are going to approach the problem.

For the second phase, the methods we will use to clean the data can be divided into seven steps: univariate analysis, bi-variate analysis, missing value treatment, outlier treatment, variable transformation, and variable creation. We will find measures of central tendency, display distributions for each variable, and conduct transformations on variables to estimate them closer to known distributions. Furthermore, between any two specific variables, we will create a matrix of graphs, and perform transformations for variables that do not appear to correlate with the target to find their relationships.

For the third phase, machine learning, we will use classification. We may use several different models, including logistic regression, KNN, random forest, and more to predict our outcomes. Seventy percent of the dataset will be split into the training set, whose 25% will be reserved for cross-validation. The remaining 30% will be used as testing data. The model with the best combination of error metrics will be used for the final interpretation.

For the final step, we will interpret the result according to real-world circumstances and expertise in finance and risk management. We will see if our result is useful in determining whether one should receive a loan, whether a bank should make an exception to a particular case, and even give advice to people who are at a disadvantage when it comes to getting a loan.

As for the software, we use Python to write the program. The project and the machine learning process will be done in Google Collab. We will use Python and Tableau for EDA and dashboarding.

Project Progress (upon March 8):

The main objective we have accomplished is the exploratory data analysis in Python. We followed the seven steps of the EDA: univariate analysis, bi-variate analysis, missing value treatment, outlier treatment, variable transformation, and variable creation.

For univariate analysis, we looked at each variable's data types, determining whether they are floats, integers, or strings. Then, we applied the .describe() function to check the means, range, and spread for each variable, to further our initial understanding of what kind of data we are dealing with.

For bivariate analysis, we aim to see how given two variables are related to one another and to what degree. An easy way to do this is to draw a correlation heat map using Seaborn to see how much correlation every two combinations of variables have. It turns out that the variables have weak correlations to each other, save for the correlation between "installment" and "loan_amnt." This is reasonable given a borrower needs to pay back more per month their debt back if the debt amount is huge.

For missing value treatment, we use the .info() function to see the “Non-null Count.” It turns out that our data has no missing values, so we skipped this step.

For outlier treatment, we wrote a function in Python to detect if any numeric data point is outside of the 1.5 IQR of its corresponding column. While we do have a function to detect outliers, we are still working on what outliers are we interested in and how should we deal with them.

For variable transformation, we used logarithmic transformation on to some skewed data such as the users’ annual income and balance.

For the variable creation step, for now, we don’t need to create more variables.

References:

1. https://www.mlq.ai/classification-based-machine-learning-for-finance/
2. https://www.sciencedirect.com/science/article/abs/pii/S0957417423001410
3. https://www.projectpro.io/article/loan-prediction-using-machine-learning-project-source-code/632

Teammates and work division:

- Eddie: EDA, conducting bivariate analysis and detecting outliers
- Ravish: Data mining, creating the report and PPT
- Jackson: Constructing machine learning models, making plots in Tableau
- Cameron: Composing the project report and plotting univariate data distribution