Financial Risk Analysis and prediction of Online P2P Lending Platform

Peer-To-Peer (P2P) lending platforms are online services provided by financial institutions as an intermediary to initiate loans for private individuals. Loans for borrowers are funded by multiple investors, bound with agreed-upon terms and conditions, with profits generated from the interest made on the loans as the borrowers are given a certain duration to pay back the loan and interest. P2P lending has gained popularity for personal, small business start-ups allowing individuals and businesses to loan money directly from investors or lenders without going through the strict requirements and criteria of traditional banks and financial institutions.This study presents a supervised machine learning model that predicts the probability of default by considering more information related to the clients rather than just evaluating their credit score. In this project , we have done the predictive analysis of the parameters like(** Loan Status ** , ** EMI**,**PROI**,**ELA**) in an Online P2P lending market related to determinants of performance predictability by conceptualizing its financial benefits to make further predictions whether one can successfully predict which loans will default. There is thus a significant financial incentive to accurately predicting which of the loans would eventually default or not.

Initial Data Overview

The Prosper Loan Dataset contains 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, and many others. Some columns are numeric and some are categorical variables.Categorical contains both (ordinal and nominal) and datetime variables.

Now we will work with features such as (Borrower Rate, Borrower APR, Prosper Score, Credit Score, Original Loan Amount, Monthly Payment, Borrower Occupation, Borrower State and others if needed).
There are some important features to look at including:

- BorrowerAPR: The Borrower's Annual Percentage Rate (APR) for the loan.
- ProsperScore: A custom risk score built using historical Prosper data. The score ranges from 1-10, with 10 being the best or lowest risk score. Applicable for loans originated after July 2009.
- LoanOriginationDate: The date the loan was originated.
- LenderYield: The Lender yields on the loan. Lender yield is equal to the interest rate on the loan less the servicing fee.

Other Features that will help us to support investigation in the Prosper Loan Data features are:

Loan Status and Employment Status will have a strong impact on loan and the features we are trying to explore further are also the Monthly Income will play a role here and the Term may have an effect. " Borrower APR is normally distributed with the peak between 15 and 20 percent in addition we have some increase in the 35 percent" "Prosper Scores are almost normally distributed and values 4, 6 and 8 are the most common." "Lender Yield is normally distributed with most of the values between 0.1 and 0.2 and we notice an increase at 0.3 "

Data Exploration

We began by loading the Prosper Loan dataset and performing an initial inspection. This included checking the shape of the data, listing the columns, and obtaining basic information about the dataset such as the number of non-null entries in each column and their respective data types. This initial exploration revealed that the dataset contains both numerical and categorical columns, some of which have missing values that need to be addressed.

Data Preprocessing

Handling Missing Values

The first step in data preprocessing was to identify columns with missing values. We calculated the number of null values in each column and filtered out those that contained missing data. This provided a clear understanding of which columns required attention.

Categorical Variables

We focused on categorical variables by isolating them and inspecting the extent of missing data within these columns. For categorical variables, missing values were filled with the mode (the most frequent value) of each respective column. This approach was chosen because it is a simple and effective way to handle missing categorical data without introducing bias.

We also created a new binary variable, LoanStatus, derived from the ClosedDate column. If ClosedDate was null, it indicated that the loan was still active (coded as 1). Additionally, if the LoanCurrentDaysDelinquent was greater than 180 days, it indicated a problematic loan, and LoanStatus was set to 1.

Dropping Unnecessary Columns

To reduce complexity and focus on the most relevant information, we dropped several columns deemed unnecessary for further analysis. These included ListingKey, LoanKey, GroupKey, MemberKey, LoanOriginationQuarter, and ListingNumber.

Filling Missing Values in Categorical Columns

For columns such as BorrowerState, Occupation, EmploymentStatus, and FirstRecordedCreditLine, missing values were filled with the mode of each column. This ensured that no categorical data was left missing.

Prosper Rating and Credit Grade

We filled missing values in the ProsperRating (Alpha) column with values from the CreditGrade column. Following this, the CreditGrade column was dropped from the dataset as it was no longer needed.

Continuous Variables

We also identified and addressed missing values in continuous (numerical) variables. Columns with missing values were filled with either the mean or median of the respective columns, depending on the nature of the data.

Exploratory Data Analysis

Univariate Analysis

Univariate analysis focuses on examining the distribution of individual variables in the dataset. This analysis helps us understand the range, central tendency, and variability of the data. Various graphs and visualizations are plotted to analyze the results effectively.

Distribution of Key Features

We plotted histograms and kernel density estimates (KDE) for several numerical features to visualize their distributions:

- **Loan Term**: The duration of the loan in months.
- **Borrower APR and Rate**: The annual percentage rate and interest rate charged to the borrower.
- **Lender Yield**: The yield earned by the lender.
- **Estimated Effective Yield, Loss, and Return**: Key financial metrics estimating yields and losses.
- **Prosper Rating (numeric)**: A numerical rating assigned to each loan.

- **Employment Status Duration**: The length of time the borrower has been employed.
- **Credit Scores and Credit Lines**: Various metrics related to the borrower's credit history.

These visualizations provided insights into the distribution and skewness of these features.

Categorical Variables

For categorical variables, bar and pie plots were used to analyze their frequency distributions:

- **Prosper Rating (Alpha)**: A proprietary rating system used to evaluate applicants.
- **Employment Status**: The employment status of the borrower, including categories like Employed, Full-time, Self-employed, etc.
- **Income Range**: The income range of the borrower.
- **Loan Status**: The current status of the loan, such as Canceled, Chargedoff, Completed, etc.

Target Variable: Loan Status

The target variable, LoanStatus, initially contained 12 different categorical values. To simplify analysis and perform binary classification, we converted this column into a binary variable using the ClosedDate and LoanCurrentDaysDelinquent columns:

- **ClosedDate**: If null, the loan is considered active.
- **LoanCurrentDaysDelinquent**: If greater than 180 days, the loan is considered problematic.

This conversion resulted in a binary column where 0 indicates incomplete loans, and 1 indicates completed loans.

Bivariate Analysis

Bivariate analysis examines the relationship between two variables. We used scatter plots to visualize these relationships:

- **Borrower Rate vs. Prosper Score**: Explores how the interest rate charged to the borrower relates to the Prosper Score.
- **Borrower Rate vs. Loan Term**: Examines the relationship between the loan term and the borrower's interest rate.

- **Estimated Effective Yield vs. Estimated Return**: Analyzes the relationship between estimated effective yield and estimated return, with the estimated loss represented by color.
- **Total Prosper Payments Billed vs. On Time Prosper Payments**: Investigates how the number of total payments relates to on-time payments.
- **Monthly Loan Payment vs. Debt to Income Ratio**: Studies the relationship between monthly loan payments and the borrower's debt-to-income ratio.

Correlation Analysis

To understand the linear relationships between numerical features, we computed the correlation matrix for a subset of numerical columns. This analysis helps identify which variables are strongly correlated, which can be crucial for feature selection and engineering in later stages of the analysis.

Visualizing Numerical vs. Categorical Features

Visualizing the relationships between numerical and categorical features helps to understand how different categories affect numerical outcomes and vice versa. Below is the explanation of the relationships derived from various plots.

Box Plots

Borrower Rate and Employment Status A box plot was used to show the distribution of Borrower Rate across different Employment Status categories. The plot reveals variations in interest rates charged to borrowers based on their employment status.

Borrower Rate and Income Range Another box plot illustrated the distribution of Borrower Rate across different Income Ranges. This helps to understand how interest rates vary among borrowers with different income levels.

Monthly Loan Payment and Employment Status The distribution of Monthly Loan Payments across different Employment Status categories was also visualized using a box plot. This plot highlights how employment status influences the amount borrowers need to pay monthly.

Borrower Rate and Prosper Rating (Alpha) The distribution of Borrower Rate across different Prosper Ratings (Alpha) was shown using a box plot. It helps to see how the proprietary rating impacts the interest rates assigned to borrowers.

From these visualizations, it was observed that employed and self-employed individuals tend to have higher monthly loan payments. Additionally, borrowers with a Prosper Rating of 'C' generally face higher interest rates.

Histograms

Borrower Rate and Loan Status A histogram was plotted to show the distribution of Borrower Rate among different Loan Status categories. This visualization indicates that loans with lower interest rates are more likely to be completed successfully compared to those with higher rates.

Borrower Rate and Prosper Rating (Alpha) Another histogram depicted the distribution of Borrower Rate across various Prosper Ratings (Alpha). The plot reveals that many loans with a Prosper Rating of 'C' tend to have higher interest rates.

Employment Status and Loan Status A histogram was used to visualize the distribution of Employment Status among different Loan Status categories. This helps in understanding the employment profile of borrowers with different loan statuses.

Multivariate Data Analysis

Multivariate analysis helps to understand the interactions between multiple variables.

Borrower Rate, Total Prosper Loans, and Loan Term A scatter plot matrix was used to illustrate the relationship between Borrower Rate, Total Prosper Loans, and Loan Term. The analysis showed that loans with a term of 3 years and 5 years have a higher number of loans compared to those with a 1-year term.

Borrower Rate, Monthly Loan Payments, and Loan Term Another scatter plot matrix depicted the relationship between Borrower Rate, Monthly Loan Payments, and Loan Term. It was observed that loans with a 1-year term generally have higher monthly payments compared to those with longer terms.

Heatmap

A heatmap was used to visualize the correlation matrix of numerical features, highlighting the strength and direction of relationships between variables. Key insights from the heatmap include:

1. **LP Gross Principal Loss and LP Net Principal Loss** have a strong effect on Loan Status, indicating that the principal loss significantly impacts whether a loan is successfully completed or not.
2. **Loan Number and Listing Number** are highly correlated, suggesting that these two variables are closely related and might be tracking similar information.

3. **Total Trades and Total Credit Lines Past 7 Years** are highly correlated, reflecting that borrowers with more credit activity over the past seven years also tend to have more trades.
4. **Loan Current Days Delinquent, Loan Original Amount, and Monthly Loan Payments** are influenced by Prosper Rating and Prosper Score, showing that better-rated loans tend to have more favorable payment terms and fewer delinquencies.
5. **Credit Score Upper and Lower Ranges** are correlated with Monthly Loan Payments, LP Customer Payments, LP Customer Principal Payments, and Loan Original Amount. This indicates that borrowers with better credit scores tend to secure larger loans and make higher monthly payments.

Feature Engineering

Feature Selection

Feature selection is crucial for optimizing model performance and reducing overfitting. Initially, we split the dataset into input features (X) and the target variable (Y, loan status). Features such as 'LoanOriginationDate' and 'ListingCreationDate' were dropped from X as they were deemed irrelevant for prediction.

Using ExtraTreesClassifier, we evaluated feature importances and identified the top influential features. Features contributing to 95% of the total importance were selected based on cumulative feature importance scores. Additionally, mutual information scores were computed to measure the dependency between features and loan status, aiding in further feature selection.

Feature Scaling

To ensure uniformity and optimal performance across different features, we applied feature scaling techniques. Standardization (using StandardScaler) and normalization (using MinMaxScaler) were implemented on the input features (X_train and X_test). Standardization transformed features to have a mean of zero and a standard deviation of one, while normalization scaled features to a range between zero and one.

Model Building: 1

Logistic Regression

Performance Evaluation:

- **Confusion Matrix Analysis:**
    - True Negatives (TN): 19653
    - False Positives (FP): 122
    - False Negatives (FN): 1
    - True Positives (TP): 3012
- **Classification Report Insights:**
    - **Precision:** The model correctly predicts 96% of loan approvals (class 1).
    - **Recall:** The model captures all actual loan approvals (100%).
    - **F1-score:** Achieves a balance between precision and recall, with an overall score of 98%.
- **Accuracy Score:** Achieves 99.5% accuracy in predicting loan status.
- **Cross-Validation Scores:** Consistently high with an average of 99.6%.

Naive Bayes

Performance Evaluation:

- **Confusion Matrix Analysis:**
    - TN: 18960
    - FP: 815
    - FN: 0
    - TP: 3013
- **Classification Report Insights:**
    - **Precision:** Successfully predicts 79% of loan approvals (class 1).
    - **Recall:** Identifies all actual loan approvals (100%).
    - **F1-score:** Balanced score at 88%.
- **Accuracy Score:** Achieves 96.4% accuracy in loan status prediction.
- **Cross-Validation Scores:** Stable, with an average of 98%.

Support Vector Machine (SVM)

Performance Evaluation:

- **Confusion Matrix Analysis:**
    - TN: 19718
    - FP: 57
    - FN: 1
    - TP: 3012
- **Classification Report Insights:**
    - **Precision:** Accurately predicts 98% of loan approvals (class 1).
    - **Recall:** Captures all actual loan approvals (100%).
    - **F1-score:** Balanced at 99%.

- **Accuracy Score:** Achieves 99.7% accuracy in loan status prediction.
- **Cross-Validation Scores:** Consistently high with an average of 99.8%.

Adaboost

Performance Evaluation:

- **Confusion Matrix Analysis:**
  - TN: 19775
  - FP: 0
  - FN: 0
  - TP: 3013
- **Classification Report Insights:**
  - **Precision:** Perfect precision in predicting loan approvals (class 1).
  - **Recall:** Identifies all actual loan approvals (100%).
  - **F1-score:** Achieves a perfect balance at 100%.
- **Accuracy Score:** Achieves 100% accuracy in loan status prediction.
- **Cross-Validation Scores:** Perfect scores with an average of 100%.

Creating New Target Variables

From the dataset, new target variables are created as continuous output variables.

LoanTenure = $(MaturityDat\_Originalyear - LoanDateyear)\ x\ 12 - (MaturityDate\_Originalmonth - LoanDatemonth)$

$MaturityDat\_Originalyear$ and $MaturityDate\_Originalmonth$ are taken from ClosedDate column.

$LoanDateyear$ and $LoanDatemonth$ are taken from LoanOriginationDate.

1.Equated Monthly Installments (EMI):

For each row in the dataset:

Calculate result_1 = P * r * 〖(1+r)〗^n
Calculate result_2 = 〖(1+r)〗^n – 1
Calculate EMI = result_1 / result_2

Tenure ---> Loan Tenure
Principle repayment ---> LP_CustomerPrinciplePayments
Interest ---> BorrowerRate

2.Eligible Loan Amount (ELA):

Calculation Procedure: For each row in the dataset:

Calculate: Total Payment Due = (A + (A*r)) * n
Calculate: Max allowable amount = I * 12 * 30%
If (Total Payment Due <= Max allowable amount) Then ELA = AppliedAmount
Else ELA = Max allowable amount

A: “AppliedAmount” --- LoanOriginalAmount

R: “Interest” --- BorrowerRate

N: “Loan Tenure” --- Loan Tenure

I: “IncomeTotal” ---StatedMonthlyIncome

3.PreferredROI:

ROI= Interest amount / Total Amount

Interest Amount= loanOriginalAmount * Borrower Rate

Total amount = Interest amount * LoanOriginalAmount

Model Building: 2

Integrate the classification and regression models into a single pipeline using a custom pipeline object or using a library like scikit-learn's Pipeline.

SPLITTING THE DATA:

Target for classification:”loanStatus”

Target for Regression:”ELA”,”EMI”,”PROI”

```
X_train, X_test, y_class_train, y_class_test, y_reg_train, y_reg_test
```

Use the predictions from the classification model to filter data for the regression model.Train a regression model on the filtered data.With Ridge as base estimator Ada boost regression model is developed. Evaluate the performance of the classification model using metrics like accuracy, precision, recall, and F1-score.Evaluate the performance of the regression model using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared.Model is trained and saved as a combined pickle file for deployment.

Once a machine learning model has been trained and tested, it can be used to make predictions on new data. In the context of P2P lending, this might involve predicting the

probability of default for a new loan application based on the borrower's characteristics and loan details.
To use the model, users typically need to provide input data in a format that is compatible with the model's requirements. This may involve preprocessing the data to ensure it is in the correct format or extracting relevant features from text data.
After the model has made a prediction, it is important to carefully interpret and evaluate the output. This may involve considering the probability of default in the context of the overall risk portfolio, or comparing the predicted probability to established risk thresholds.Finally, it is important to continuously monitor and evaluate the performance of the model over time. This may involve tracking model accuracy, evaluating the model on new data, or updating the model with new features or data sources as they become available.Overall, building and using machine learning models for P2P lending requires careful consideration of the relevant features, algorithms, and performance metrics, as well as ongoing monitoring and evaluation to ensure the model remains accurate and fair over time.

Deployment

The model is deployed using streamlit.Create a new Python file for your Streamlit app (e.g., app.py), and set up the app to load the pickle file and make predictions based on user inputs.Make sure you have Streamlit and other necessary libraries installed. You can install them using pip.Navigate to the directory where app.py is located and run the following command to start the Streamlit app.Open a web browser and go to http://localhost:8501 to access your Streamlit app.

For production deployment, consider using a cloud service (e.g., Heroku, AWS, GCP) to host your Streapp.