Project 2: Predicting LendingClub Interest Rates

OPAN 6604: Predictive Analytics

SAXA: Mike Johnson

1.0 Introduction

LendingClub is a peer-to-peer lending platform that serves individuals and small businesses and allows customers to borrow money directly from investors. The platform uses technology to assess an applicant's creditworthiness to determine an interest rate for their loan.

The objective of this project is to develop and compare machine learning models in Python that predict the interest rate assigned to a LendingClub loan at origination, using only borrower and loan attributes at the time of the application.

2.0 Methodology

The initial dataset used in the training and development of the two models included 100,000 loans originated from 2007 to 2020 and included 24 selected variables. The independent variables include information on the applicant's creditworthiness and details related to the loan.

2.1 Handling Missing Values

An initial exploration of the dataset revealed missing values in several columns. Each column was reviewed, and either imputation or feature removal were employed. Figure 1 in the appendix summarizes the handling of missing values in each column where they were present.

2.2 Feature Engineering and Outlier Handling

The distributions of all numerical features were reviewed. Outliers were removed if they were present in any given feature to help prevent overfitting. In addition, transformations were applied to features with skewed distributions when present. A summary of outlier handling and feature engineering can be found in figure 2 of the appendix. Finally, one-hot encoding was applied to categorical features. This creates binary columns for each category, preventing the model from assuming an ordinal relationship between categories.

2.2.1 Neural Network Feature Engineering

Neural Networks learn by updating weights through gradient descent. If the features have wildly different scales, the gradients can become unstable and take longer to converge. To address this, data for the neural network model was normalized by scaling to min and max of each feature.

2.3 Train-Test Split

To ensure a fair evaluation of our models, the dataset was split into a training set (60%), dev set (15%), and a testing set (25%). The models were trained exclusively on the training data, dev set used for hyperparameter tuning and model selection, and their final performance was evaluated on the unseen testing data. This separation is critical for assessing the models' ability to generalize new, unknown loan applications.

4.0 Data Analysis

4.1 Exploratory Analysis

4.1.1 Fico Score

Fico scores are a key variable to pay attention to since it is a metric that assesses an individual's creditworthiness. This score ranges from 300 to 850, where higher scores mean favorable creditworthiness. This is reflected in the dataset where higher scores translate to lower interest rates on average. However, the interest rate starts to increase on average for credit scores greater than 825. This is

largely due to limited observations of applicants greater than 825. To address this bias, these records with a Fico credit score greater than 825 were removed from training.

4.1.2 Fico Score and Balance to Credit Limit

A correlation analysis (Appendix: Figure 3) identified **Fico Score (fico_range_high)** and **Balance to Credit Limit (all_util)** to have moderate correlation with interest rates. Also, these two features are correlated with each other. This is expected since Balance to Credit Limit is used to determine Fico Scores. To address collinearity between the two, Balance to Credit Limit was removed.

4.1.3 Other Instances of Collinearity

Total credit balance excluding mortgage shows evidence of collinearity (Appendix: Figure 3) with multiple features and was removed. Finally, there are two variables related to inquiries that correlate each other. **Months Since Recent Inquiry (mths_since_recent_inq)** was dropped in favor of **Inquiries in The Last 12 Months (inq_last_12m)** as a result.

4.2 Model Development

The relationships between features and the interest rate are non-linear (Appendix: Figure 4). This means that linear regression would not be a good candidate for predicting interest rates. Instead, two powerful machine learning models from the scikit-learn library were selected for this predictive task: a Random Forest Regressor and a Multi-layer Perceptron (MLP) Regressor, which is a form of a neural network. Both excel at handling non-linear relationships.

For both models, randomized search cross-validation, a hyperparameter tuning technique that explores a random subset of hyper parameter combinations within a specified range, was used to determine the optimal hyperparameters. This technique was used over grid search cross validation due to its efficiency benefits.

5.0 Results

The performance of both models was evaluated on the test set using three standard regression metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and R-squared (R^2).

Model / Notes	RMSE	MAE	R-Squared
Random Forest	3.90	2.86	0.45
Neural Network	4.50	3.42	0.26
Notes	RMSE quantifies the differences between values predicted by a model and the values actually observed. It is helpful for comparing the performance of different models.	Mean Absolute Error (MAE) is a measure of the average magnitude of errors between predicted and actual values in a dataset.	Value is between 0 and 1 (i.e. standardized and easy to interpret). The closer to 1, the better the data points fit the model.

The Random Forest model demonstrated superior performance with a lower RMSE, MAE and a higher R^2 value. Based on these results, the Random Forest model was selected as the final, best-performing model.

Project 2: Predicting LendingClub Interest Rates
OPAN 6604: Predictive Analytics
Mike Johnson

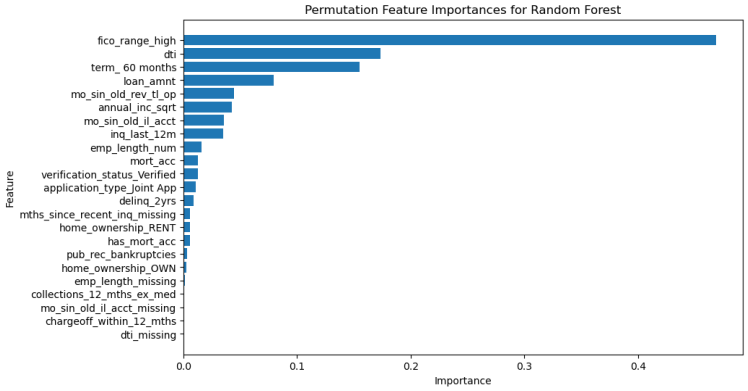
6.0 Discussion

6.1 Model Explainability

To gain trust in our model and understand its decision-making process, two model explainability techniques were applied: Permutation Importance and LIME.

6.1.1 Permutation Feature Importance

Permutation Feature Importance is a technique used to assess the global importance of features in a trained model. This method is model-agnostic and provides a clear, intuitive ranking of which features matter most. The plot below shows the most important features for our Random Forest model.



Observations:

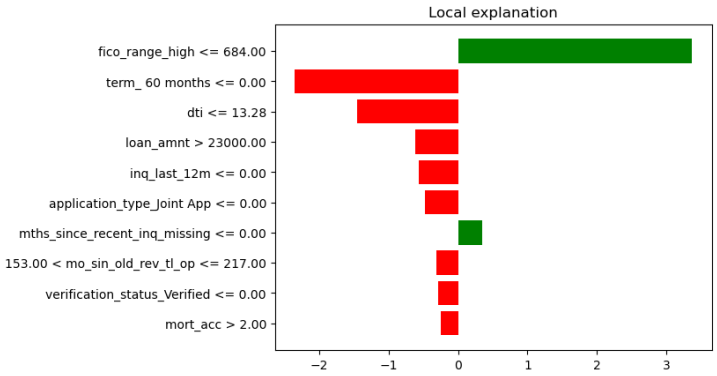
- **FICO is king:** fico_range_high accounts for 45% of the model's predictive power. Applicants with higher credit scores generally pay lower rates, and the model leans heavily on this signal.
- **DTI and Loan Terms:** dti (debt-to-income) and term (60-month term) each contribute 16% to the model's predictions. Higher DTI's push rates up, and 60-month loans push rates up.
- **Most binary features matter very little:** Features such as `dti_missing` have little importance in the model.

6.1.2 LIME (Local Interpretable Model-agnostic Explanations)

While Permutation Importance provides a global overview, LIME explains individual predictions. LIME works by creating a simple, interpretable local model around a specific prediction to approximate the behavior of the more complex global model. This allows us to see why a particular loan application received its predicted interest rate.

The plot below shows how one applicant's predicted interest rate was impacted. For example, this applicant had a lower Fico score, which resulted in their predicted interest rate increasing. However, they also had a 36-month term loan, which lowered the predicted interest rate.

Project 2: Predicting LendingClub Interest Rates
OPAN 6604: Predictive Analytics
Mike Johnson



6.2 Python vs R

When comparing the experiences working on with this dataset in Python vs R, there are some distinct differences between the two.

R offers a more approachable experience. The tidyverse package makes data manipulation and visualization coding consistent and readable. This differs from Python, where the equivalents of Pandas and Seaborn have a learning curve to them. The same is true when comparing the machine learning packages such R's Caret and Python's Scikit-learn. For example, the set-up of the train-test split felt cumbersome in Python, where it separates the x and y variables into separate objects that need to be merged for preprocessing and then re-split for model training.

The benefits of Python become apparent once performance and scalability are considered. For example, one challenge when training models in R for this dataset was the amount of time required for resource intensive models like random forests. There were ways to reduce the training time in R, but Python's Scikit-learn had more readily available options like n_jobs for parallel processing.

7.0 Conclusion

In this project, machine learning models were trained to predict loan interest rates from the LendingClub dataset. After rigorous preprocessing and comparison, the Random Forest model emerged as the most accurate model. By leveraging explainability techniques, transparent insights were gained to inform how the models predicted interest rates in addition to their predictive capabilities. The analysis confirmed that variables related to borrower risk, such as FICO scores, are the primary determinants of interest rates. Overall, this project highlights the effectiveness of the Random Forest model in predicting loan interest rates and underscores the importance of borrower risk variables like FICO scores.

Project 2: Predicting LendingClub Interest Rates
OPAN 6604: Predictive Analytics
Mike Johnson

Appendix

Figure 1: Summary of Handling of Missing Values

Feature	Handling	Notes
Debt-to-Income (dti)	Impute median and add a dummy variable that fags missing values in dti so that the model can learn whether the feature carries information.	An NA likely means that the applicant had no income and is reflected in the training set. Imputing 0 is implies an exceptional DTI, but no income suggests a credit risk. On the other hand, imputing 999 to represent these missing values might mislead the training of the model.
Employment Length (emp_lenth)	Converted to an integer, impute the median, and add a dummy variable that identifies a missing emp_length.	NA's likely represent cases where the applicant did not provide their data related to the feature.
Months Since the Oldest Bank Installment Account Opened (mo_sin_old_il_acct)	0 will be imputed and a dummy variable will be added to flag instances where there are no installment accounts.	NA's likely represent cases where the applicant has no installment account.
Months Since Last Public Record (mths_since_last_record)	Dropped since there's isn't enough data to work with.	There are a lot of NA's (90% in the training set).
Months Since Recent Inquiry (mths_since_recent_inq)	NA's will be imputed with 0 and add a dummy variable that flags the NA's.	NA's likely mean that there are no inquiries on file for the applicant.

Figure 2: Summary of Outlier Handling and Feature Engineering

Feature	Preprocessing	Feature Engineering
Balance to credit limit on all trades (all_util)	Instances greater than 100% are rare and create noise. Instances > 100% were removed.	NA
Number of charge-offs within 12 months (chargeoff_within_12_mths)	NA	Most applicants have no chargeoffs in the past 12 months. This was converted to a binary that measures if the applicant had a chargeoff in the last 12 months.
Number of collections in 12 months excluding medical collections (collections_12_mths_ex_med)	NA	Similar to above.
Past Due Incidences of Delinquency (delinq_2yrs)	NA	Similar to above.
Annual Income (annual_inc)	Annual income is skewed right some extreme values. Removed instances greater than the 99th Percentile	Square root transformation applied to address skewness.
Debt-to-Income (dti)	Instances > 40 removed.	NA
Fico Score Low (fico_range_low)	The difference between the low and high score is 4 for all records. Removed the low score since the high score alone is sufficient.	NA
Inquiries in the Last 12 months (inc_last_12m)	Data greater than the 99th percentile was removed.	NA
Months since oldest bank installment account opened (mo_sin_old_il_acct)	Data greater than the 99th percentile removed.	NA
Months since oldest revolving account opened (mo_sin_old_rev_tl_op)	Data greater than the 99th percentile removed.	NA
Number of mortgage accounts (mort_acc)	Removed instances greater than the 99th percentile.	Many applicants don't have a mortgage account. Adding a binary column to identify those who do and do not assist with training.
Number of public record bankruptcies (pub_rec_bankruptcies)		Similar to above
Total credit balance excluding mortgage (total_bal_ex_mort)	Removed instances greater than the 99th percentile.	Square root transformation applied to address skewness.

Figure 3: Correlation Matrix of Numerical Columns

Project 2: Predicting LendingClub Interest Rates
OPAN 6604: Predictive Analytics
Mike Johnson

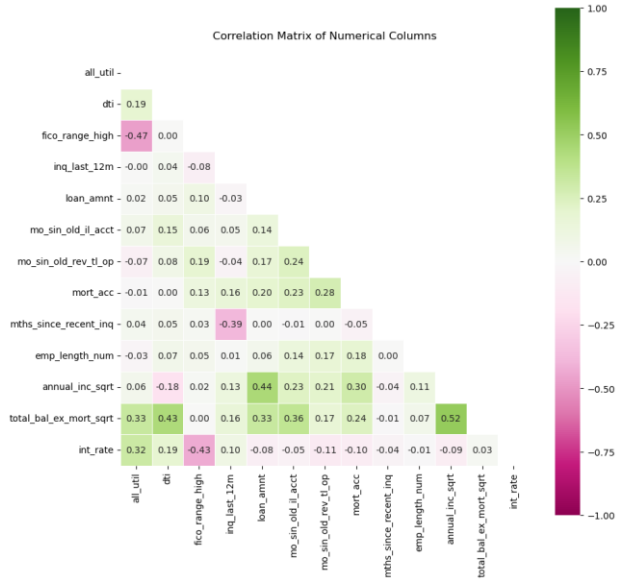


Figure 4: Non-Linear Relationship to Interest Rates

The plot below shows scatter plots of two variables vs interest rates. This is an example of the non-linear relationship between features and interest rates.