Working Paper Step 4
Group B

Objectives and Scope

This report summarizes the progress made during the Proof of Concept stage. We started with a dataset of 62 features and 322,980 instances, with data preparation details provided in reports 2 and 3. The main focus is on developing a classification model to predict loan returns above 2% and a regression model for yield prediction, as detailed in the previous report (step 3).

We prioritized the precision index due to its crucial importance in achieving returns over 2%. Our goal was to minimize false positives while maintaining high precision and to identify a sufficient number of potential loans for investment, excluding those classified as negative. For a more technical understanding of the tasks completed in this step, please refer to the attached code (Step4.ipynb).

Documentation of completed tasks

In this step, we began by isolating the classification and regression models. Following thorough evaluation, we determined our next course of action on our coming step: to integrate them using a cascading method. Initially, we will employ the classification model to narrow down the pool of loans. Subsequently, we apply the regression model with the aim of achieving more accurate predictions of expected returns.

Classification

Before implementing our classification models, we observed significant skewness in the dataset. Specifically, 78.8% of loans yield above 2% return, while only 21.2% yield below 2%. This notable imbalance underscores the need for specialized techniques and models to ensure accurate and fair classification results.

Baseline: We initiated this step by establishing a baseline with a naive majority likelihood classifier. Using a 50% threshold, loans were classified based on their historical group probability. However, this approach produced a model inferior to random(See appendix A2). Consequently, in the next phase, our attention will shift to A1, denoting the lowest risk grade(See appendix A1). These loans are highly recommended for investment and act as predictions from the original platform model.

Feature selection: We used forward feature selection, tailored for each model, to pick the optimal 15 features (see Appendix B for a detailed explanation).

Model selection: (see appendix c for more details on the hyperparameters & GridSearch)

Logistic Regression: Chosen for its simplicity, interpretability, and robustness in capturing the relationship between input variables and class probabilities. We utilized forward selection to identify the most relevant features and adjusted for the imbalanced distribution using the class_weight argument. Hyperparameter optimization involved testing various combinations using grid search, prioritizing high F1- scores.

AdaBoost Classifier: Chosen for its sequential boosting approach, which enhances weak classifiers, especially in scenarios with class imbalance. After reviewing initial results, we adjusted sample weights to improve the model's

ability to address imbalance. Moving forward, we plan to explore a broader range of predictors to potentially enhance its performance further.

XGBoost Classifier: Employed for its excellent performance in handling complex datasets and its capability to fine-tune hyperparameters. Similar to Logistic Regression and AdaBoost, we tuned the hyperparameters using GridSearchCV. Despite its robustness, we implemented sample weights based on class frequencies to ensure equal attention to minority class samples during training, akin to the approach in AdaBoost. We plan to experiment with a higher number of predictors for this model as well.

Evaluation Metrics: We utilized various performance metrics to comprehensively assess our models. Recognizing the data imbalance, we understood that precision and recall alone were insufficient, leading us to incorporate accuracy despite the business emphasis on precision. We particularly focused on the false positive rate (FPR) due to its significant business implications, as investing in unprofitable loans can be detrimental. To address this, we employed the ROC curve to optimize the area under the curve, considering both FPR and the true positive rate.

Model evaluation result : Logistic Regression demonstrated the highest average F1 score (0.70) with of precision (0.74), while AdaBoost Classifier attained the highest average precision (0.7655**9**), and XGBoost Classifier closely followed with a competitive precision of (0.7655**2**) and the highest AUC (0.70) (see appendix D for ROC Curves and full evaluation details).

Regression

We opted to assess five regression models, four of which are variants of linear regression and Random forest regression. The main differences between these models lie in their loss functions, regularization techniques, and their abilities to handle outliers and multicollinearity.

Feature selection: Used Sequential Feature Selection, which involves adding or removing features sequentially according to a performance metric for each model by defining a threshold for improvement we ended up with 16 features .

Model selection: (see appendix E for more details on the hyperparameters & GridSearch)

Linear Regression: We chose Linear Regression to estimate the relationship between dependent and independent variables. It serves as a baseline starting model due to its simplicity and ease of interpretation. However, it has no regularization, making it sensitive to overfitting.

Lasso Regression: Lasso Regression was selected for its ability to incorporate L1 regularization, which encourages sparsity and aids in feature selection by shrinking some coefficients to zero. This regularization helps in reducing overfitting. The regularization parameter alpha was tuned using grid search over a range of values to optimize performance.

Ridge Regression: Ridge Regression was chosen for its L2 regularization, which penalizes large coefficients and helps reduce model complexity and multicollinearity. This regularization ensures that the model remains stable and generalizes better to new data. Similar to Lasso, the alpha parameter was tuned using grid search to find the optimal value.

Huber Regression: Huber Regression was selected for its robustness to outliers. It utilizes a combination of L1 and L2 penalties, making it effective at handling data disturbances and anomalies. This robustness is crucial for datasets with irregularities, ensuring the model remains accurate even in the presence of outliers.
Random Forest Regression: This model was chosen for its ability to handle non-linear relationships and feature interactions. By constructing multiple decision trees and averaging their predictions, it enhances accuracy and robustness. Using bootstrap aggregating (bagging), where each tree is trained on a random sample of the data, helps reduce variance. Additionally, considering a random subset of features at each split improves generalization. The hyperparameters, including the number of estimators, the maximum depth of the trees, and the maximum number of features considered at each split, were tuned using grid search to optimize performance.

Evaluation Metrics: To evaluate the performance of our model, we employed a 5-fold cross-validation technique. We use Mean Squared Error (MSE) and R-squared (R^2) as the primary evaluation metrics for our regression models. MSE assesses the accuracy of our model's predictions of realized returns, aiding in making informed investment decisions. The MSE should be interpreted in the context of our target variable range (expected return) which after normalization ranges from 0.0 to 1.0. R^2 indicates the proportion of variance in returns that the model can explain, providing insight into its effectiveness in capturing factors that influence P2P lending profitability. In future steps, we plan to include additional evaluation metrics to ensure a more comprehensive assessment of model performance.

Model evaluation result : Linear Regression, had an MSE of 0.1103 and an R^2 of 0.2510. Lasso Regression showed similar results with an MSE of 0.1103 and an R^2 of 0.2510. Ridge Regression also had an MSE of 0.1103 and an R^2 of 0.2510. Huber Regression had a higher MSE of 0.1308 and a lower R^2 of 0.1121. Random Forest Regression stood out with the lowest MSE of 0.1099 and the highest R^2 of 0.2538, demonstrating its superior ability to capture complex non-linear relationships and interactions among features. It's important to note that at this stage we are not satisfied with our results (See appendix F for a more detailed analysis)

Financial Potential Analysis To evaluate the financial potential given a specified budget, two line plots were created. The first plot illustrated the cumulative investment amount versus the model's predicted yield, offering insights into potential returns at various budget levels. The second plot juxtaposed the cumulative investment amount with the actual yield, enabling an assessment of the model's accuracy in predicting real returns. These visual representations provided a clear understanding of the financial outcomes based on budget allocations and the models' performance in forecasting yields. Notably, the Random Forest Regression model emerged as the best-performing regression model, which further underscores the reliability of our predictions. The line plots are detailed in Appendix G.

Potential Pitfalls: During our analysis, we identified several potential pitfalls that require meticulous consideration. These include aspects such as model evaluation, multicollinearity, overfitting, skewness handling, and other critical factors. To ensure a thorough understanding of these issues and their implications, we have elaborated on them extensively in Appendix H. This comprehensive detailing aims to promote transparency and guide future research efforts.

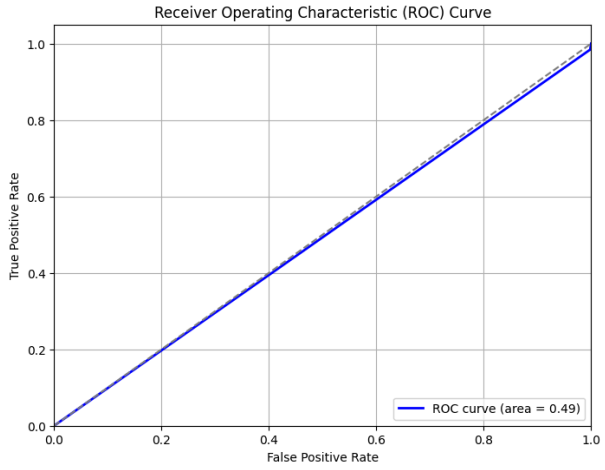
Overall, while the regression models focused on accurately predicting realized returns, the classification models prioritized optimizing precision and recall to mitigate the risks associated with investing in unprofitable loans. Both approaches provided valuable insights, with Random Forest Regression and Logistic Regression standing out in their respective categories.
However, it is important to note that our regression models are currently performing poorly. Moving forward, we will focus on their learning curves and evaluate their performance specifically on loans classified as yielding above 2%. We will also concentrate on addressing underfitting and overfitting to improve the models' accuracy and robustness.

Appendices

Appendix - A1 Naive majority classifier by grade group

	grade	percentage of loans under 2% yield	majority_grade_pred
0	A	5.610515	over
1	B	12.965733	over
2	C	21.497393	over
3	D	29.969165	over
4	E	37.827821	over
5	F	47.199632	over
6	G	68.759571	under

Appendix - A2 Roc curve of the baseline (lower than 0.5)



Appendix - B Forward Feature Selection

Forward feature selection is a step-by-step process used to select the most relevant features for a model. It starts with an empty set of features and iteratively adds one feature at a time. In each iteration, it evaluates the performance of the model with each possible new feature added to the existing set, selects the feature that improves the model performance the most, and includes it in the set. We started with a naive approach and set the number of features to 15 to get a first assessment of the performance.

Appendix C - Best hyper-parameters for the classification models using grid-search

GridSearch- GridSearch works by systematically searching through a predefined set of hyperparameters to determine the best combination for a model. It evaluates each possible combination using cross-validation to identify the configuration that yields the highest performance.

Hyperparameters Tuned and Best Results:

Logistic Regression:

- C: Regularization parameter controlling the inverse of regularization strength. Experimented with values in the range [1e-4, 1e4].
  - Best hyperparameters: {'C': 0.013257113655901081}

AdaBoost:

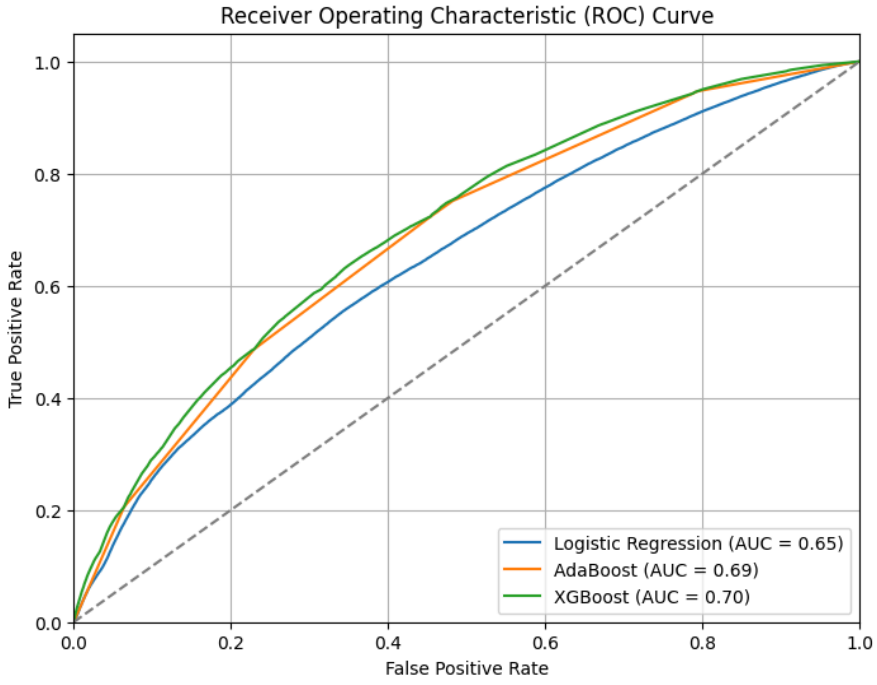
- n_estimators were kept to 100 and will be tested on the next step.
- learning_rate: The contribution of each weak learner to the final prediction. Experimented with values [0.01, 0.1, 1].
  - Best hyperparameters: {'learning_rate': 0.01}

XGBoost:

- n_estimators were kept to 100 and will be tested on the next step.
- learning_rate: Shrinkage rate to prevent overfitting. Experimented with values [0.01, 0.1, 0.3].
- max_depth: Maximum depth of each tree. Experimented with values [3, 6, 9].
  - Best hyperparameters: {'learning_rate': 0.01, 'max_depth': 3}

Appendix D - Classification Models evaluation

	Metric	Logistic Regression	AdaBoost	XGBoost	Grades Baseline
0	Average F1 score	0.703458	0.622546	0.632185	0.072327
1	Average Recall	0.682923	0.581923	0.592207	0.201400
2	Average Precision	0.735141	0.765590	0.765521	0.406711
3	Average Accuracy	0.682923	0.581923	0.592207	0.201400



Appendix E - Best hyper-parameters for the classification models using grid-search

To experiment this time the GridSearch method as explained in appendix c , however added a threshold on improvement in performance to avoid the rigid decision of number of features. We'll test interchange the two methods next step to see what yields better results.

Here's a summary of the hyperparameter tuning performed for each model, including the best performing parameters:

1. Lasso Regression:
  - Tuned the regularization parameter `alpha` over a range of values.
  - Best alpha value: 0.0001

2. Ridge Regression:
  - Tuned the regularization parameter `alpha` over a range of values.
  - Best alpha value: 1.0

3. Random Forest Regression:

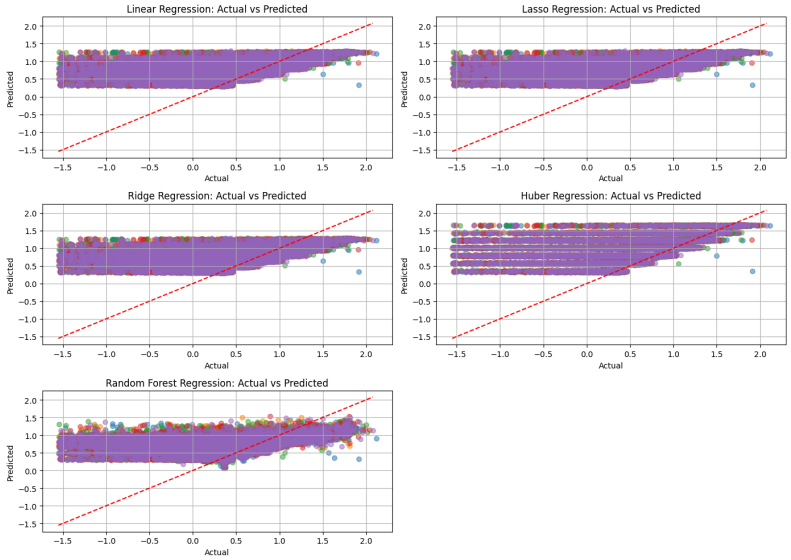
- Tuned the number of estimators, maximum number of features, and maximum depth of the tree.

Best parameters: n_estimators: 100, max_features: sqrt, max_depth: 10

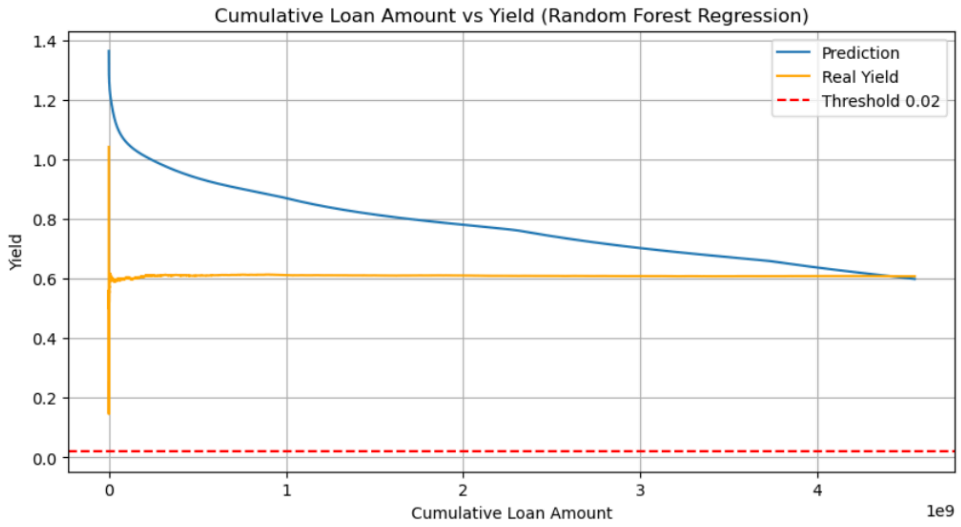
Appendix F - Regression Models Evaluation

Model	Mean MSE	Mean R^2
Linear Regression	0.110304	0.251005
Lasso Regression	0.110304	0.251001
Ridge Regression	0.110303	0.251011
Huber Regression	0.130764	0.112057
Random Forest Regression	0.109888	0.253832

The graphs show that the majority of data points are positioned above the diagonal line, implying a consistent tendency for the model to overestimate actual values. This trend indicates potential underlying complexities or nuances within the data that current models struggle to capture effectively. In the next step, we will experiment with further tuning and cascading methods on already classified loans to potentially yield more accurate results.



Appendix G-Financial Potential Analysis:



Our goal is to illustrate the relationship between the cumulative loan amount and both the predicted and actual yields. This graph aims to show the expected return for each amount of investment based on our model. Unfortunately, our regression models performed poorly in predicting loan returns and were overly optimistic regarding the expected returns. This is evident from the cumulative graph, where a significant gap exists between the predicted and actual yields. Additionally, the predicted returns appear unrealistically high.

While the trend is quite reasonable—the cumulative loan amount increases, and the predicted yield decreases due to higher-rated loans being prioritized for investment—the results depict that our model is not accurate enough. Further investigation regarding model complexity, data preprocessing, feature selection, training and testing, imbalanced data, or other factors needs to be conducted in the next step of the project. Addressing these issues will be crucial for improving the accuracy and reliability of our predictions and thus in displaying the financial potential.

Appendix H - Potential Pitfalls and Steps for Improvement:

Throughout our analysis, we identified several potential pitfalls that warrant careful attention and areas for further investigation to enhance our model's accuracy and robustness:

Insufficient Consideration of Factors Beyond Yield: In our current approach, we prioritized loans solely based on their yield without considering the associated risks and probabilities of not yielding more than 2%. Moving forward, we need to integrate the classification model (identifying loans yielding more than 2%) with the regression model. This integration will involve basing our regression predictions on the classification results and incorporating the probability and risk of a loan not yielding over 2% in our expected return predictions. This will help us balance high returns with associated risks to optimize our investment decisions.

Market Conditions and External Factors: Our models currently do not account for evolving market conditions and external factors that can influence their performance and accuracy. Incorporating these elements will be crucial for creating more robust predictions.

Benchmark Models: Our current benchmark models might not be sufficient or accurate enough for effective comparison. Improving our benchmarks will provide a better reference point for evaluating model performance.

Underfitting and Overfitting:

Underfitting: Our current models might be overly simplistic, failing to capture the complexity of the data. We should consider using more complex models with higher dimensionality, such as Polynomial Regression, to find relationships in the data.

Overfitting: On the other hand, overly complex models might perform well on training data but fail to generalize to unseen data. Balancing model complexity is crucial.

Feature Selection Approach: We should explore other methods and criteria for selecting the optimal number of features to ensure the best model performance.

Data Splitting Ratios: Testing different data splits for training and testing, such as 80:20 and 70:30 ratios, as these can significantly impact model performance.

Normalization Methods: Evaluating other normalization methods beyond Min-Max, such as robust scaling, to ensure data consistency.

Handling Imbalanced Data: Improving our approach to handling imbalanced data to ensure that the models perform well across all classes.

Addressing these pitfalls and areas for improvement will be crucial for developing a more reliable and effective investment model.