DataDriven
PORTFOLIO SOLUTIONS

Stage E Report: Modeling and Evaluation

This report presents the outcomes of the modeling and evaluation stage, where we conducted extensive training, fine-tuning, and testing of our models on a dataset comprising 71 features and 326,403 instances. Our primary objective was to evaluate the models' performance from a statistical and scientific perspective, ensuring their efficacy and reliability. As detailed in the previous report, our approach to assessing the asset relevance of P2P loans involves two strategies: classifying loans with yields higher than the current company average (classification model) and predicting loan yields (regression model). Building upon the detailed investigation in the previous report, we implemented approaches to address the skewed distribution and utilized techniques such as oversampling, class weighting, and thresholding for classification, along with transformations. The dataset was divided into training and testing sets, with the optimal 70:30 split yielding superior results. Data normalization was tailored individually for each model and feature, with the robust scaler technique delivering the best outcomes. The modeling process involved evaluating various performance metrics, employing cross-validation, monitoring overfitting, and fine-tuning hyperparameters. For comprehensive details, please refer to the step D report.

Documenting Completed Work Assignments-

1. **To optimize and monitor overfitting of the model, the following steps were undertaken:**
   - **a.** Cross-validation: Cross-validation: In order to evaluate the performance of our model on various data subsets, we employed k-fold cross-validation. However, due to time limitations, we opted to conduct the tests using only the standard value of k, which is 5.
   - **b.** Monitor training set performance: Regularly check the model's performance on the training set as high performance could indicate overfitting.
   - **c.** Hyperparameter tuning: Fine-tune hyperparameters to control model complexity and mitigate overfitting such as max_depth, learning_rate, Tolerance for stopping criteria, degree etc..
   - **d.** Regularization: Apply L1 and L2 regularization to prevent overfitting and balance model complexity.

2. **Classification Models:**

   In the previous step, we evaluated three models: logistic regression (LG), AdaBoost with decision trees, and XGBoost. Following the procedures outlined in report D, we conducted extensive experiments with various hyperparameter combinations, normalization techniques, data imbalance handling strategies, and feature selection methods for each algorithm. After comparing the results using multiple evaluation metrics, taking into consideration the limitations of precision and recall due to data imbalance (as discussed in report D), we found that the logistic regression model consistently outperformed the others across all tested indices (model estimates can be found in Appendix A). Given our time constraints, our focus will be on further exploring the logistic regression model,we conducted the following steps:

   - **a. Logistic Regression Optimization:** We conducted a more in-depth analysis of the logistic regression model. We explored A wider variety of hyperparameters combinations, feature selection methods (including using all features, forward selection, and selection based on importance), and the number of features included in the model. We also examined the impact of applying data imbalance treatment techniques. For a detailed methodology and the results of the logistic regression model, please refer to Appendix B and Appendix C, respectively.

b. **AdaBoost with Logistic Regression:** We implemented an AdaBoost model using logistic regression as the weak classifier to enhance our classification model. AdaBoost, as an ensemble learning technique, has the potential to outperform standalone logistic regression by leveraging the strengths of multiple weak classifiers. In our investigation, we thoroughly explored the model by testing different feature selection methods, varying the number of features included, exploring different combinations of hyperparameters, and evaluating the impact of data imbalance treatment techniques. Through these extensive analyses, we aimed to identify the optimal configuration of the AdaBoost model for our dataset. For detailed information on the methodology and results of this model, please refer to Appendix D and Appendix E, respectively.
c. **Exploring KNN:** To conduct a comprehensive test, we incorporated the K-Nearest Neighbors (KNN) algorithm as an alternative to our existing classification techniques (logistic regression, XGBoost, and decision trees). The objective was to explore the effectiveness of distance-based methods in pattern recognition and gain valuable insights from the data. Throughout our exploration of KNN, we investigated various approaches for handling data imbalance, selecting features, and determining the optimal value of K. The goal was to determine the optimal KNN model configuration. For detailed methodology and results, please refer to Appendix F and Appendix G.
d. **Models Evaluation:** In Appendix H, the logistic regression model demonstrated consistently superior performance across a majority of metrics, reaffirming its effectiveness. Notably, the model achieved an improved AUC score of 0.65 on the test set, surpassing the previous report's results. While acknowledging that the model is not flawless, it still holds significant value in providing predictions regarding the probability of a loan yielding above 2%. In the subsequent report, which will concentrate on the business aspect, we will delve deeper into the analysis of this model in comparison to the existing grade model and we will explore the financial potential and implications further to gain comprehensive insights.

3. **Regression Models:**

In the preceding phase, we scrutinized two distinctive regression models: Linear Regression and Polynomial Regression. Abiding by the guidelines elucidated in appendix K and L in the previous report, comprehensive experiments were carried out deploying a myriad of hyperparameter combinations, normalization techniques, feature selection approaches for each regression model. Post extensive comparison of results with the application of our key evaluation metrics - Mean Squared Error (MSE) and R-squared (R^2), and understanding the constraints due to potential overfitting and model complexity (appendix K and L in the previous report), we discerned that the Polynomial Regression model consistently showcased superior performance over the Linear Regression model across all established parameters. Given our time constraints, our focus will be on further exploring the Polynomial Regression model,we conducted the following steps:

a. **Evaluation metrics:** We use Mean Squared Error (MSE) and R-squared (R^2) as evaluation metrics for our regression model. MSE helps measure the accuracy of our model in predicting realized returns, allowing us to make informed investment decisions. R^2 quantifies the proportion of variability in returns explained by the model, providing insights into the model's ability to capture factors influencing P2P lending profitability. In the next step, we will incorporate additional evaluation metrics for a more comprehensive assessment.
b. **Lasso and Ridge regression:** In this experiment, Ridge and Lasso regularization were applied to a polynomial regression model to mitigate multicollinearity and prevent overfitting. Ridge spreads the weights of coefficients evenly while Lasso can reduce less significant feature coefficients to zero.After transforming data into polynomial features and scaling, grid search with cross-validation was used to find the optimal regularization parameter (alpha). For Ridge regularization, the optimal alpha value was 10, yielding an MSE of 0.00706 and R^2 of 0.02778.

Lasso regularization, with an alpha value of 0.01, yielded better results with an MSE of 0.00704 and R^2 of 0.03043. This suggests that for a degree 2 polynomial regression model, Lasso regularization provided a better fit to the data, with no substantial overfitting observed. However, further tests and metrics are needed for final confirmation.

However, when using the best features, both Ridge and Lasso regularizations with an alpha value of 10 led to negative R^2 values, suggesting a poor fit to the data. In this case, it seems the model performed worse than a simple average model. This indicates the selected "best features" may not provide sufficient or correct information for predicting the outcome variable in the regression model.(see Appendix I)

c. **Models Evaluation:**The Polynomial Regression model with degree=2, Lasso regularization (alpha=0.01), and all_features performed the best among the evaluated models. It achieved a lower mean squared error (MSE) of 0.007041 and a higher R2 score of 0.030425. Lasso regularization helped in feature selection, reducing overfitting and improving generalization. The inclusion of polynomial terms allowed capturing non-linear relationships. Considering all available features contributed to leveraging the full information for better predictions. However, the final determination of the "best" model depends on specific data, problem domain, and project goals (refer to Appendix J).

d. **Financial Potential Analysis:** For our final model predicting the realized return for each loan, we analyzed the estimated volumes (amount of money) for the different predicted yields for all peer-to-peer lending loans. The analysis of the cumulative loan amount and the corresponding predicted and real weighted yields revealed the following (refer to the graphs in appendix K).

e. **Existing Model vs. Investigated Models-**In the next step, we will further Investigate the models and we will compare these results with the existing Grade model for a comprehensive analysis.

4. **Selected models:** After careful analysis, we have selected the logistic regression model for the classification task and the polynomial regression model for the regression task. This two-step modeling approach effectively addresses our objective. The initial classification model categorizes loans into two groups based on their yield (above or below 2%), enabling a focused analysis on the subset of loans with higher potential returns. The regression model is then applied exclusively to the loans classified as above 2% to predict their specific yield values. By combining these models, we leverage the strengths of both classification and regression techniques, resulting in deeper insights into loan performance and more accurate predictions for investment opportunities. In the next phase, when we examine the business aspect of these models, we will make the final decision on whether to utilize both models or just one of them.

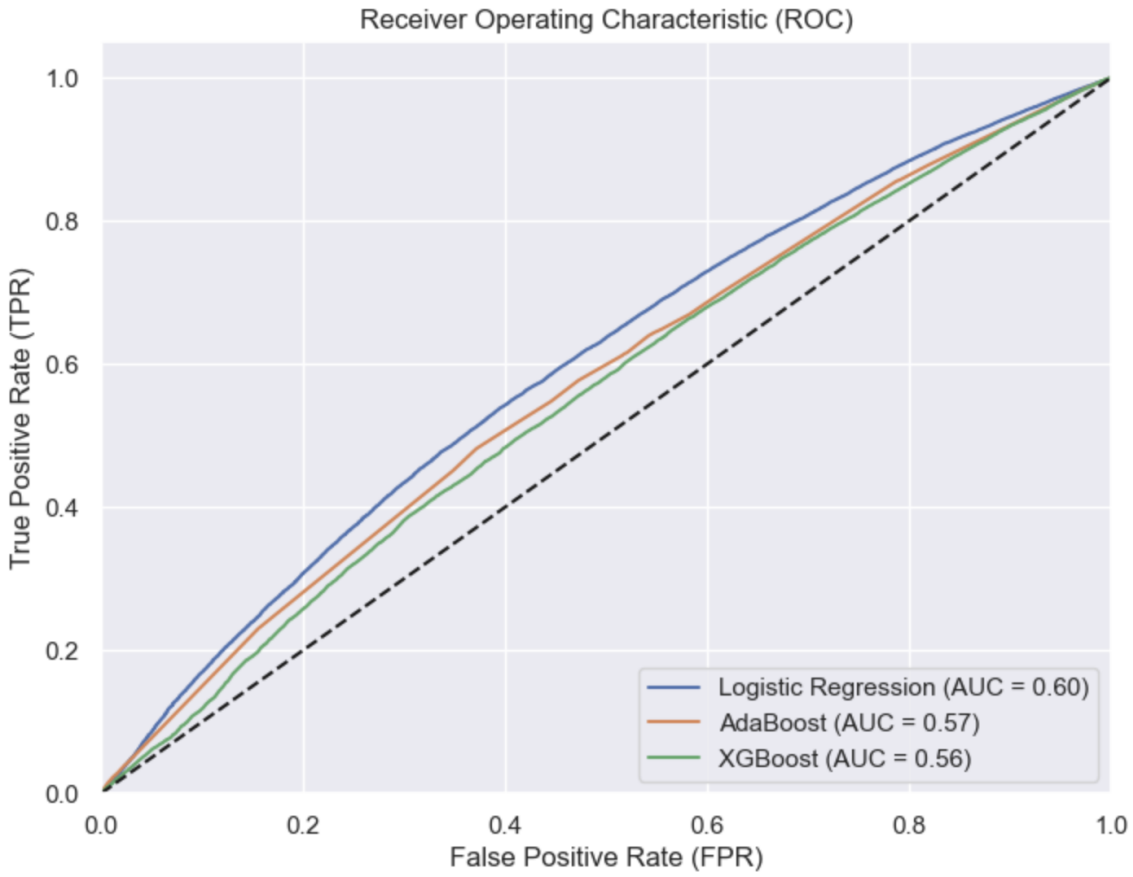
5. **Potential Pitfalls:** Potential Pitfalls: we have identified several potential pitfalls that require careful consideration. These pitfalls include issues related to model evaluation, multicollinearity, overfitting, skewness handling, and other important factors. We have provided detailed information about these pitfalls in appendix L to ensure transparency and guide future research efforts.

Our next steps involve evaluating the models from a business perspective, comparing them to the existing grade model, assessing their financial potential, and addressing the CFO's questions. Our ultimate objective is to determine the best model for our business and evaluate the feasibility of integrating P2P loans into our investment portfolio.

Appendices

Appendix A- Recap: Evaluation of Classification Models in the Previous Stage:

Recall: After performing various experiments with different hyperparameter combinations, addressing target variable imbalance, selecting appropriate algorithms, normalization techniques and features selection methods, the logistic regression model consistently demonstrated the best performance across all evaluation metrics.



	Logistic Regression	AdaBoost	XGBoost
F1	0.722	0.604	0.612
Recall	0.714	0.604	0.6124
Precision	0.710	0.6045	0.6128
Accuracy	0.741	0.604	0.612

Appendix B- Logistic Regression Model and Methodology:

To develop an effective logistic regression model, we implemented a series of steps and made specific choices to ensure its accuracy and reliability. The following is an overview of the model and methodology employed:

Feature Selection: Feature selection is a critical step in identifying the most relevant predictor variables for our logistic regression model.
To ensure a comprehensive analysis, we adopt a multi-faceted approach, encompassing various techniques:

1. **Use of all existing features:** As a starting point, we construct a base model that incorporates all available features. This allows us to establish a baseline for performance and serves as a reference for subsequent improvements.
2. **Forward selection:** Employing a sequential variable addition approach based on predictive power, we mitigate the challenges posed by the curse of dimensionality. By iteratively incorporating variables with the highest predictive strength, we strike a balance between model complexity and performance. To determine the optimal number of features, we systematically explore three values: 10, 15, and 20. This range ensures that our model avoids oversimplification while also preventing excessive complexity.
3. **Utilization of tree-based feature importance:** Leveraging the power of a random forest classifier, we assess the importance of each feature in predicting the target variable. This approach aids in identifying the most influential predictors for inclusion in our model. To determine the ideal number of features, we experiment with five values: 10, 15, 20, 25, and 30. By commencing with a minimum of 10 features, we ensure a sufficient level of complexity. Simultaneously, we cap the feature selection at 30 to maintain a balanced trade-off between performance and model intricacy.

Through the integration of these feature selection techniques, we effectively navigate the challenges of dimensionality and identify a subset of features that significantly contribute to the predictive power of our logistic regression model. This meticulous approach enhances the robustness and accuracy of our analysis, enabling us to make informed and reliable predictions.

Hyperparameter Tuning: Hyperparameter tuning plays a crucial role in optimizing the logistic regression model. We specifically focused on three key hyperparameters: the maximum number of iterations (n_iter), tolerance (eps), C and penalty types (penalty). Given computational constraints, we carefully selected a manageable number of combinations to explore:

1. **n_iter:** To strike a balance between convergence and computational efficiency, we considered three values: 1000, 2000, and 5000. By varying the number of iterations, we aim to enhance convergence and improve the model's overall accuracy.
2. **eps:** To control the convergence threshold, we investigated a range of epsilon values, including 0.01, 0.001, 0.0001, 0.00001, and 0.000001. A lower epsilon value signifies a higher level of precision in achieving convergence, thereby influencing the model's performance.
3. **penalty:** In order to evaluate the impact of regularization on the model's performance, we assessed two penalty types: 'l1' and 'l2'. The choice of penalty type helps regulate overfitting and influences the selection of relevant features, ultimately affecting the model's predictive capabilities.

4. **C:** We explored the effect of different regularization strengths by evaluating four values: 0.1, 0.5, 1, and 10. The regularization parameter (C) controls the inverse of the regularization strength, with smaller values indicating stronger regularization. By varying C, we aimed to find the optimal balance between model complexity and generalization performance.

Skewness Handling and Threshold Adjustment: To address the imbalanced class distribution, we employed techniques for skewness handling and threshold adjustment. In comparing different approaches to address the skewness in the data, we utilized the following techniques:

1. Utilizing the 'class_weight' argument to account for the imbalance in class distribution.
2. Exploring over-sampling as a means to address the skewness.
3. Evaluating the impact of not applying any specific technique on the data.

Additionally, the model's performance was assessed at different classification thresholds: 0.40, 0.50, and 0.60. By adjusting the threshold, we aimed to strike a balance between minimizing false positives and false negatives, taking into account the business impact and the trade-off between investment opportunities and risks.

Evaluation Metric: To assess the model's performance, we employed k-fold cross-validation with k=5. This technique ensures reliable and robust evaluation by randomly dividing the data into training and validation sets, maintaining a 70:30 ratio. The F1 score, which considers the weighted average across the classes, was chosen as the evaluation metric. It provides a comprehensive assessment of both precision and recall.

Best Model Selection: Based on the averaged F1 scores, we identified the best-performing logistic regression models. The results of the best logistic regression models, including the chosen hyperparameter combinations, selected features, evaluated thresholds, and their corresponding F1 scores, can be found in Appendix C. These results provide valuable insights into the model's performance and guide decision-making in achieving accurate predictions.

Appendix C- Logistic Regression Model Evaluation Results and Best Hyperparameters:

Initially, we conducted a thorough analysis by considering various metrics to address the data imbalance issue, while also exploring different combinations of hyperparameters. After evaluating multiple models, we identified the top-performing models based on the following results:

	LG on Imbalanced Target Variable	LG with class_weight	LG with Oversampling
F1	0.694836	0.723743	0.720238
Recall	0.787621	0.738097	0.728281
Precision	0.719665	0.713381	0.706489
Accuracy	0.787621	0.738097	0.728281

As explained in the previous analysis, when dealing with imbalanced data, relying solely on precision or recall can be misleading. In our case, both precision and recall are crucial from a business perspective: precision ensures that our investments are successful, while recall helps us avoid missing valuable opportunities. The F1 score, which combines precision and recall, allows us to assess the overall performance considering this trade-off. We observed that in the model without handling the imbalanced data, the F1 score was the lowest, confirming our concerns. Therefore, we decided to employ data balancing methods. Among the two methods we compared, the class_weight approach demonstrated superior performance across all metrics, leading us to choose this method for implementation.

Subsequently, we proceeded to explore various applications of pitch selection, while conducting experiments with different combinations of hyperparameters. Through this process, we aimed to identify the most effective approach for determining the optimal pitch selection strategy. We identified the top-performing models based on the following results:

	LG with all the features	LG with forward selection	LG with feature importance
F1	0.463846	0.723743	0.724021
Recall	0.441476	0.738097	0.739447
Precision	0.759545	0.713381	0.727746
Accuracy	0.441476	0.738097	0.791447

As predicted, the model that included all features yielded the lowest F1 score. Interestingly, the precision score was higher for this model, underscoring the limitation of relying solely on precision or recall in isolation. However, after implementing feature selection based on tree-based feature importance, we observed significant improvements across all evaluation metrics. Consequently, we chose to incorporate this specific stage of feature selection in our model.

We acknowledge that a more comprehensive approach, examining all combinations of methods and hyperparameters, would have provided a more precise analysis. However, due to time and computational constraints, we were unable to explore every possible combination in-depth. We made the best use of the available resources to conduct a thorough analysis within the given limitations.

Best model parameters:

n_iter	eps	penalty	selected_features_list	Skewness Handling	Features selection method	C	threshold
5,000	1e-06	l2	[57,58,52,1,68,19,48,64,66,49,14,54,67,35,65,18,15,45,34,44,11,40,36,10,5,51,50,43,42,13]	class_weight	feature importance	1	0.4

Best model evaluation:

Metric	Value
F1	0.723458
recall	0.73072
precision	0.717307
accuracy	0.73072

Appendix D: AdaBoost with Logistic Regression Model and Methodology-

Our approach involved utilizing AdaBoost with logistic regression as the weak classifier to enhance our classification model. We conducted an in-depth analysis to identify the optimal configuration, considering various aspects such as feature selection methods, hyperparameter combinations, and data imbalance treatment techniques. The following provides a detailed overview of the model and methodology implemented:

Feature Selection: Due to computational limitations, we chose to use the features that yielded the best results in the previous step of logistic regression. While acknowledging that a more thorough examination could have potentially improved the outcome, we made this decision to manage computational constraints.

Hyperparameter Tuning: Our focus was on tuning three key hyperparameters for optimizing the AdaBoost classifier: the learning rate, the number of estimators, and the base classifier parameters. Considering computational constraints, we selected a manageable number of combinations to explore:

1. Learning rate and number of estimators: We evaluated different combinations of learning rates (0.1, 0.5, and 0.6) and the number of estimators (30, 50, and 80). By varying these parameters, we aimed to enhance convergence and improve the overall accuracy of the model.
2. Base estimator: We experimented with two types of base estimators in our AdaBoost model:
   a. Type 1 Base Estimator: Basic and Weak Model- consisted of a basic and weak model without any modifications to the hyperparameters.
   b. Type 2 Base Estimator: Optimized Model from Logistic Regression- utilized a base estimator incorporating the hyperparameters that yielded the best results in the previous logistic regression step. This allowed us to compare the performance of the AdaBoost model using both the default and optimized base estimators.

Skewness Handling and Threshold Adjustment: Considering computational constraints, we chose to utilize the most effective method identified in the previous in-depth analysis for the logistic regression model. We employed the class_weight technique for addressing skewness in the data. Similarly, we used the same threshold adjustment method as determined for the logistic regression model.

Evaluation Metric: To evaluate the performance of the AdaBoost classifier model, we employed k-fold cross-validation with k=5. This approach ensured reliable and robust evaluation by randomly dividing the data into training and validation sets, maintaining a 70:30 ratio. The F1 score, which considers the weighted average across classes, was selected as the evaluation metric. It provided a comprehensive assessment of both precision and recall.

Best Model Selection: Based on the averaged F1 scores, we identified the best-performing AdaBoost classifier models. The results, including the selected hyperparameter combinations, evaluated thresholds, and corresponding F1 scores, can be found in Appendix E. These findings provide valuable insights into the performance of the model and serve as a guide for making informed decisions to achieve accurate predictions.

Appendix E- AdaBoost with Logistic Regression Model Evaluation Results and Best Hyperparameters:

During our investigation, we explored various hyperparameter combinations for the two types of base classifiers. Subsequently, we identified the models that yielded the best results for each base classifier. Now we will present the best model that all the code produced:

	AdaBoost with Basic and Weak Model	Optimized Model from Logistic Regression
F1	0.695193	0.694558
Recall	0.788454	0.787992
Precision	0.621663	0.620933
Accuracy	0.788454	0.787992

The AdaBoost model incorporating the Type 1 Base Estimator, a basic and apparently weaker model without any parameter changes, demonstrated superior performance compared to the alternative model. This outcome can be attributed to the ensemble nature of AdaBoost, which leverages the collective strength of multiple weak classifiers. Despite the simplicity of the Type 1 Base Estimator, its integration within the ensemble framework effectively captured relevant patterns and yielded accurate predictions. Consequently, the model with the basic and weak base estimator, without any parameter modifications, emerged as the better-performing model in our analysis.

Best model parameters:

n_iter	eps	penalty	selected_features_list	threshold	learning_rate	n_estimators
5,000	1e-06	l2	[0, 1, 2, 4, 6, 7, 11, 12, 13, 14, 16, 17, 18, 19]	0.4	0.6	50

Best model evaluation:

Metric	Value
F1	0.695193
recall	0.788454
precision	0.621663
accuracy	0.788454

Appendix F: Knn Model and Methodology

To comprehensively assess classification performance, we introduced the K-Nearest Neighbors (KNN) algorithm as an alternative to our existing methods (logistic regression, XGBoost, and decision trees). Our objective was to leverage distance-based techniques for pattern recognition and gain valuable insights from the data.
The following provides a detailed overview of the model and methodology implemented:

Feature Selection: Given computational limitations, we decided to focus on two feature selection methods:

1. **Forward selection:** Employing a sequential variable addition approach based on predictive power, we mitigate the challenges posed by the curse of dimensionality. By iteratively incorporating variables with the highest predictive strength, we strike a balance between model complexity and performance. To determine the optimal number of features, we systematically explore three values: 10, 15, and 20. This range ensures that our model avoids oversimplification while also preventing excessive complexity.
2. **Utilization of tree-based feature importance:** Leveraging the power of a random forest classifier, we assess the importance of each feature in predicting the target variable. This approach aids in identifying the most influential predictors for inclusion in our model. To determine the ideal number of features, we experiment with five values: 10, 15, 20, 25, and 30. By commencing with a minimum of 10 features, we ensure a sufficient level of complexity. Simultaneously, we cap the feature selection at 30 to maintain a balanced trade-off between performance and model intricacy.

By employing these techniques, we aimed to prioritize the most informative features for our model.

Feature Scaling: Feature scaling is essential for KNN models as they heavily rely on distance metrics between data points. Varying feature scales can lead to disproportionate influence, hindering accurate distance calculations. To address this, we applied three normalization techniques: Min-Max scaling, standard scaling, and robust scaling, ensuring consistent value ranges across features during training. Through rigorous evaluation, robust scaling demonstrated superior performance, making it the optimal choice for preprocessing features in our KNN model. This approach mitigates scaling variations, enhances distance measurement accuracy, and facilitates reliable predictions based on nearest neighbors.

Hyperparameter Tuning: Our focus was on optimizing the KNN classifier by tuning key hyperparameters, including the number of features, the scaling method, the number of neighbors (k), and the distance metric. To efficiently explore the hyperparameter space, we employed a pipeline approach and evaluated a range of combinations. Specifically, we considered:

1. **various values of k:** ranging from 3 to 15, In increments of 2 to ensure that there will be an unequivocal majority decision.
2. **4 distance metrics:** Euclidean, Manhattan, Chebyshev, Minkowski.

Skewness Handling: When applying the k-nearest neighbors (KNN) model, it is not essential to handle skewness in the target variable. KNN is a non-parametric algorithm that does not make assumptions about the underlying distribution of the target variable. Instead, it relies on the proximity of neighboring data points. Therefore, skewness in the target variable does not impact the performance or assumptions of the KNN model, making it unnecessary to address this aspect specifically.

Evaluation Metric: To evaluate the performance of the KNN classifier model, we employed k-fold cross-validation with k=5. This approach ensured reliable and robust evaluation by randomly dividing the data into training and validation sets, maintaining a 70:30 ratio. The F1 score, which considers the weighted average across classes, was selected as the evaluation metric. It provided a comprehensive assessment of both precision and recall.

Best Model Selection: After evaluating the averaged F1 scores, we determined the top-performing KNN classifier models. Appendix G contains detailed results, showcasing the selected hyperparameter combinations, feature selection methods, feature scaling techniques, and corresponding F1 scores. These findings offer valuable insights into the model's performance and serve as a valuable reference for informed decision-making to ensure accurate predictions.

DataDriven
PORTFOLIO SOLUTIONS

Appendix G- KNN Model Evaluation Results and Best Hyperparameters:

Through our experimentation with two different feature selection methods (forward selection and feature importance), we ran two separate codes to explore various combinations of normalization types, k numbers, and measurement methods in the KNN model. Now we will present the best model that all the code produced:

	Knn by forward selection	Knn by features importance
F1	0.724328	0.695011
Recall	0.784289	0.788321
Precision	0.725432	0.621453
Accuracy	0.784289	0.788321

The analysis of the results reveals that the model utilizing forward selection for features selection outperformed the other model. As a result, we have chosen this model as the preferred option due to its superior performance. This decision is based on the observed outcomes and highlights the effectiveness of the forward selection method in improving the model's predictive capabilities.

Best model parameters:

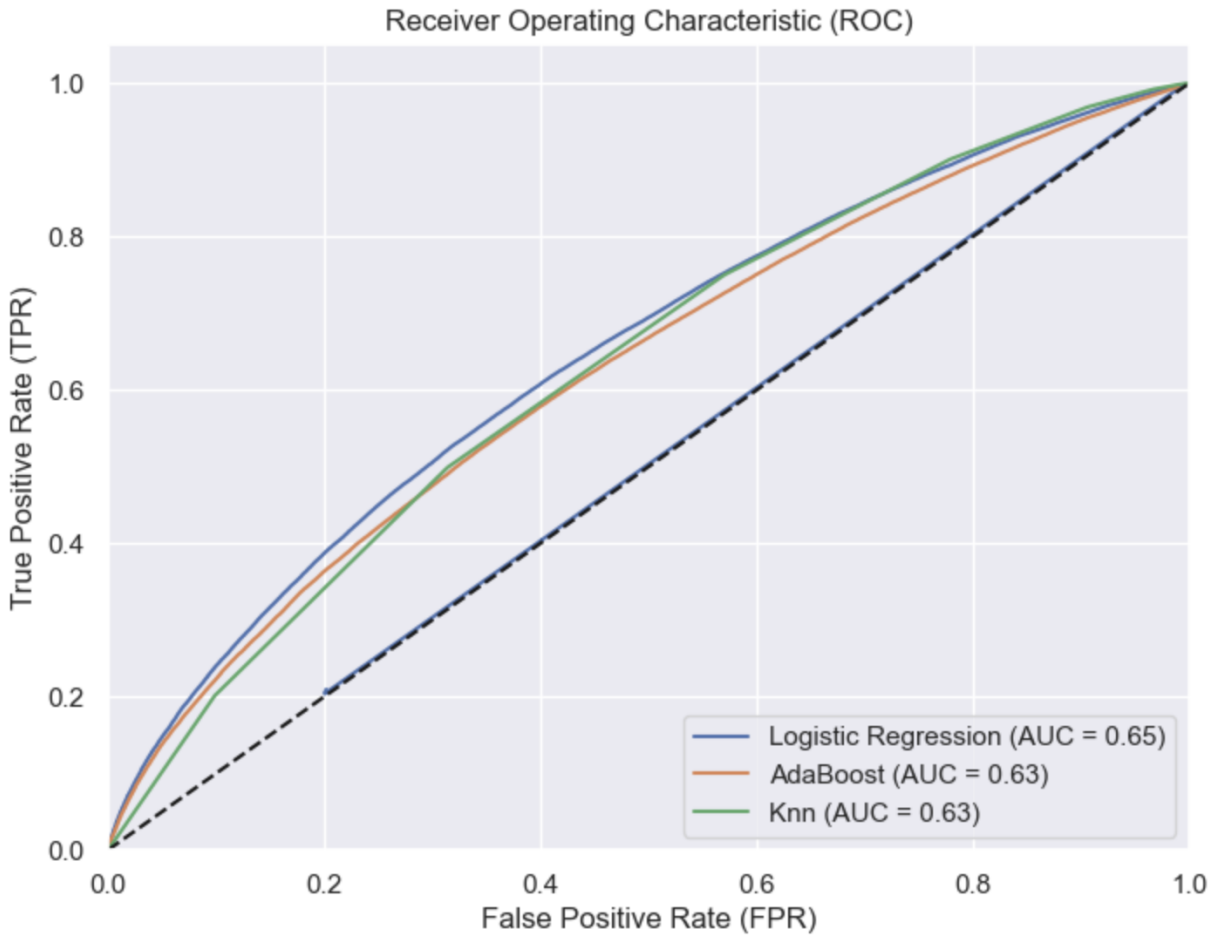
scaler	n_neighbors	metric	Features selection method	selected_features_list
RobustScaler()	9	chebyshev	forward selection	0, 1, 2, 4, 15, 24, 26, 27, 28, 29

Best model evaluation:

Metric	Value
F1	0.72334
recall	0.783509
precision	0.724735
accuracy	0.783509

Appendix H: Evaluation of the classification models:

The logistic regression model exhibited superior performance compared to other models, as evident from its highest AUC score, F1 score, precision, and accuracy scores. While the recall score, representing missed investment opportunities, was not the highest, the prioritization of other metrics aligns with our specific context. Considering the nature of our problem domain and the relative importance of different evaluation metrics, the overall effectiveness of the logistic regression model, as reflected in its comprehensive performance across multiple metrics, makes it the preferred choice.



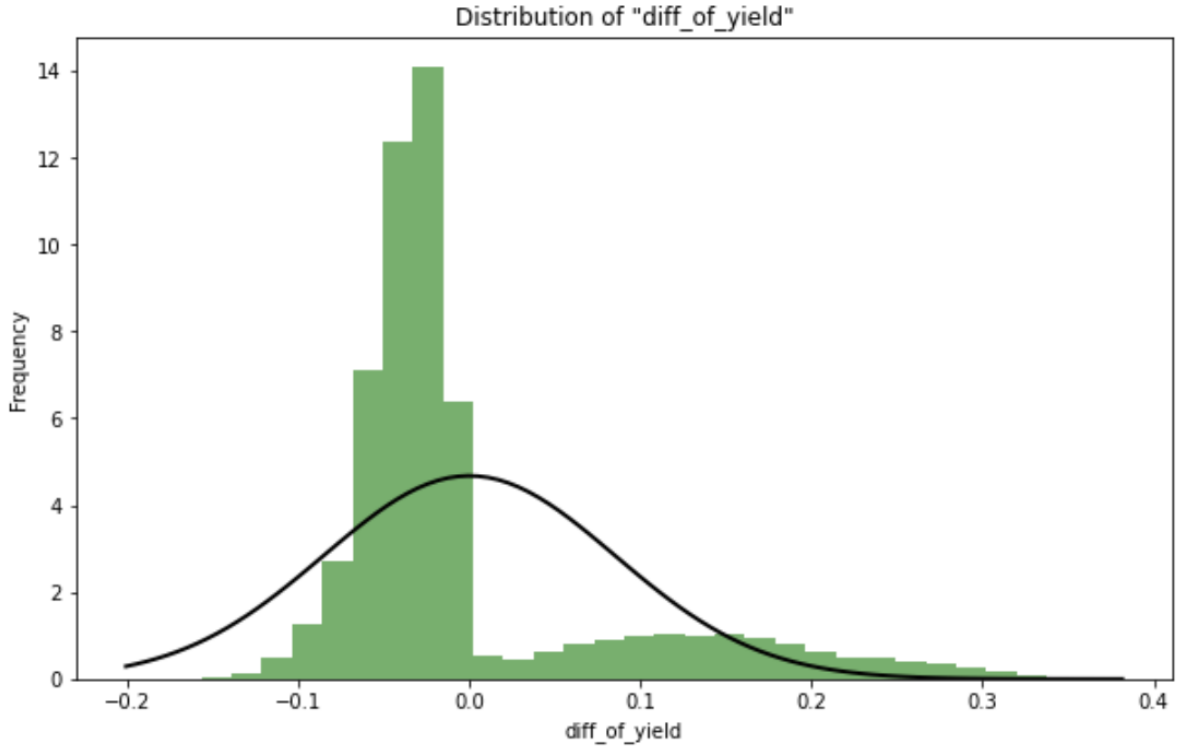
	Logistic Regression	Best AdaBoost with Logistic Regression	Best Knn
F1	0.725021	0.695193	0.724328
Recall	0.739447	0.788454	0.784289
Precision	0.727746	0.621663	0.725432
Accuracy	0.791447	0.788454	0.784289

Appendix I: Comparison of models:

Model_Name	MSE	R2_Score	Regularization	Alpha
Polynomial Regression (degree=2) - all_featuers	0.00706	0.027781	Ridge	10
Polynomial Regression (degree=2) - all_featuers	0.00702	0.030425	Lasso	0.01
Polynomial Regression (degree=2) - best_featuers	0.00733	-0.00882	Ridge	10
Polynomial Regression (degree=2)- best_featuers	0.00726	-0.00029	Lasso	10

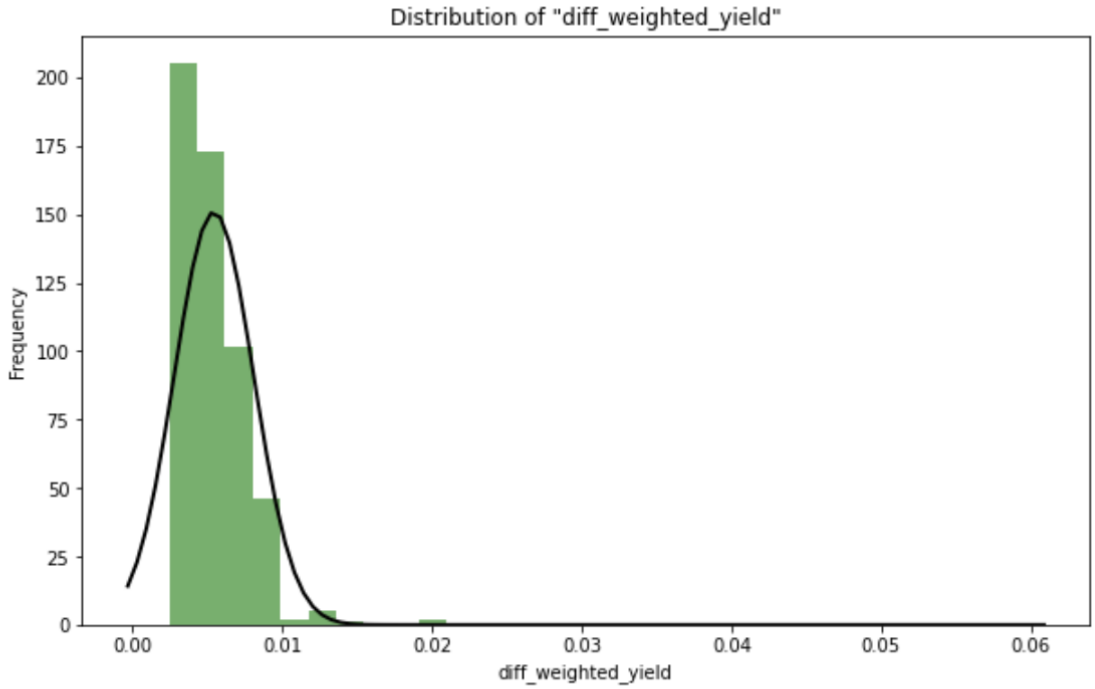
Appendix J: More statistical results

```
Mean of 'diff': -1.8535593212225408e-06
Standard Deviation of 'diff': 0.08528777623813089
```




```
Descriptive Statistics:
count   97170.00
mean       -0.00
std         0.09
min        -0.17
25%        -0.05
50%        -0.03
75%        -0.01
max         0.36
Name: diff, dtype: float64
```

```
Mean of 'diff_weighted': 0.0054587335295935
Standard Deviation of 'diff_weighted': 0.0026440640031357757
```




```
Descriptive Statistics:
count   97170.00
mean       -0.00
std         0.09
min        -0.17
25%        -0.05
50%        -0.03
75%        -0.01
max         0.36
Name: diff, dtype: float64
```

1. Mean: The near-zero mean value of 'diff' implies that the model's yield predictions, on average, are very close to the expected yields. The small negative sign indicates a slight tendency towards underestimation, but this is so minute that it's practically negligible. This suggests that your model has a good average predictive performance, but we can't make an overall assessment just from this.

2. Standard Deviation: The standard deviation of approximately 0.085 signifies the model's typical "error" or deviation from the expected yield. It represents how much, on average, the model's predictions deviate from the actual yields. A lower standard deviation signifies that the predictions are close to the actual yields, indicating the model is reasonably precise. However, the interpretation depends on the range of yield values. If yields vary significantly (for example, from -100% to 100%), then 0.085 might be quite good. But if yields range from -1% to 1%, this might not be satisfactory.

Remember, these are just statistical measures that provide a broad sense of the model's accuracy and precision. They don't offer specifics on how the model performs in different yield ranges, the shape of the errors, potential skewness, or the presence of outliers. The statistical analysis indicates that the model's yield predictions have a

near-zero mean, suggesting they are very close to the expected yields on average. The small negative sign implies a minor tendency towards underestimation, but it is negligible. The standard deviation of approximately 0.085 represents the model's typical deviation from the expected yield, indicating reasonably precise predictions. However, the interpretation depends on the range of yield values. Additional financial analysis considerations include evaluating the model's impact on profitability, risk management, portfolio optimization, conducting backtesting and sensitivity analysis, and validating the model using out-of-sample data. A comprehensive financial analysis is necessary to assess the model's performance in various market conditions and make informed investment
decisions based on its predictions.

Appendix K: Financial analysis:

cumulative_loan_amnt	weighted_yield_yearly_1	weighted_yield	diff
35000.00	0.08	0.14	0.05
295856050.00	0.02	0.03	-0.04
554453875.00	0.02	0.03	-0.04
811626650.00	0.02	0.02	-0.03
1081071725.00	0.02	0.02	-0.04

- Cumulative Loan Amount: The cumulative loan amounts range from 35,000.00 to 1,081,071,725.00 across different loans.
- Predicted Weighted Yield: The predicted weighted yields range from 0.02 to 0.14, indicating the estimated yield for each loan.
- Real Weighted Yield: The real weighted yields range from 0.02 to 0.08, representing the actual yield realized for each loan.

Further examination of the difference in weighted yields provides insights into the model's performance. The mean difference between predicted and real weighted yields is approximately 0.0055, indicating a slight positive bias in the predictions. The standard deviation of the difference is approximately 0.0026, suggesting a relatively low deviation from the expected yield predictions.The descriptive statistics for the difference in yields reveal that the majority of differences are close to zero, with a mean close to -0.00 and a standard deviation of approximately 0.09. The minimum difference is 0.00, while the maximum difference is 0.06, indicating some variations in the accuracy of yield predictions. The interquartile range (25th to 75th percentile) falls between -0.05 and -0.01, suggesting that a significant portion of the differences is negative, indicating an underestimation of yields in the model.Overall, these results indicate that the model's predictions are generally close to the actual yields, with a slight positive bias on average. However, there is some variability in the accuracy of the predictions, with some loans showing larger differences.

Appendix L - Potential Pitfalls:

- Overfitting: complex models that may overfit the training data and fail to generalize well to unseen data.

- Underfitting: overly simplistic models that fail to capture the complexity of the data, resulting in poor performance on both the training data and unseen data.Failure to account for changing market conditions and external factors: Ignoring the influence of evolving market conditions and external factors that can impact the model's performance and accuracy.
- Assumption of linearity: Logistic regression assumes a linear relationship between the independent variables and the log-odds of the dependent variable. If the relationship is non-linear, the model may not capture the true underlying pattern, leading to reduced predictive accuracy.
- Limited flexibility in capturing non-linear relationships: Logistic regression assumes a linear relationship between predictors and the log-odds of the outcome. If the relationship is non-linear or exhibits complex interactions, logistic regression may not adequately capture these patterns, leading to reduced model performance.
- Limited ability to handle categorical variables: While logistic regression can handle categorical predictors by using dummy coding or other encoding techniques, it may not capture the full complexity of categorical variables with multiple levels or interactions among categories.
- Feature Selection: The negative R^2 values indicate that the "best features" chosen may not have been the most suitable for our model, affecting its performance.
- Multicollinearity: This refers to predictors that are correlated with each other, potentially leading to unreliable and unstable estimates of regression coefficients, which can complicate model interpretation and prediction.
- Model Complexity: Polynomial regression models can quickly become complex and difficult to interpret, making it harder to extract actionable insights.
- Hyperparameter Tuning: Balancing regularization parameters is crucial. An incorrect alpha could either oversimplify the model causing underfitting or fail to manage overfitting effectively.
- Data Scaling: Incorrect or lack of data scaling could result in misleading outcomes, as the model may become sensitive to the range of the input features.
- Over-reliance on a 2% yield threshold, missing true potential and associated risks: Placing excessive emphasis on a specific yield threshold without considering the potential benefits and risks associated with different yield levels.
- Insufficient consideration of factors beyond yield (e.g., risk, default rates): Failing to take into account other important factors, such as risk and default rates, that can significantly impact the performance and accuracy of the model.
- Failure to address potential overfitting issues: Not taking necessary steps to prevent the model from fitting noise in the training data and lacking generalization capability for unseen data.
- Inadequate assessment of alternative skewness handling methods: Not sufficiently evaluating and comparing different techniques for handling skewed data distributions.
- Failure to balance skewed distribution while maintaining interpretability: Neglecting the need to balance skewed data distributions while still ensuring the interpretability of the model's predictions and insights.
- Neglecting potential drawbacks of oversampling techniques: Failing to consider and address the potential drawbacks and limitations associated with oversampling techniques used to handle imbalanced data.
- Limited evaluation of models using relevant metrics and benchmarks: Inadequate assessment of model performance using appropriate evaluation metrics and comparison with industry benchmarks.

- Accounting for non-linear relationships for improved accuracy: Considering non-linear relationships between variables to improve the accuracy and predictive power of the model.
- Lack of comparison with existing models or approaches: Not comparing the performance and effectiveness of the developed models with existing models or approaches in the field.
- Computational power limitations restricted model testing: Limitations in computational resources hindered thorough testing and exploration of the models' capabilities.