Success Criteria:

Summary:

The model successfully meets its defined success criteria outlined in the model card and represents an improvement over the previous version, particularly in identifying instances of Resting Order (RO).

The earlier model exhibited subpar performance in production, primarily due to discrepancies between the training data and the data used during deployment. This issue was not visible to the model validators.

The performance criteria are established with precision and recall thresholds set at over 85%. The current model achieves a precision of 89.6% and a recall of 96.1% on the training data, while it performs with a precision of 92.3% and a recall of 92.3% on the test data.

The model is trained on a total of 878 samples, with 702 samples allocated for training and 176 samples for testing. The training dataset comprises 467 positive samples and 235 negative samples, while the test dataset includes 117 positive samples and 59 negative samples, resulting in a ratio of approximately 0.5, as shown in Table 1. The data contains no NaN values or duplicates. We also conducted tests for data leakage between the training and testing datasets and found no evidence of leakage. The distribution of positive and negative cases in both datasets is illustrated in Figure 1.

The model utilizes seven features. To assess the conformity of the training and testing data, we performed two sets of analyses. Given the limited number of features, we first compared the distribution of these features between the training and testing datasets. While most features exhibited similar distributions, two features—focus\_firm\_opp\_side\_avg\_non\_ro\_exec\_qty and feature\_xcnfirm\_segment\_median\_cum\_event\_qty\_gradient—displayed noticeable differences in their distributions across the datasets, as depicted in Figure 2. Previous experiences with varying serving data patterns suggest that these discrepancies could impact our conclusions regarding model performance.

Additionally, we clustered the data into six groups. The test set exhibited a gap in two of the clusters, indicating potential distribution differences, as shown in Figure 3. Figure 4 presents the distribution of positive cases across these clusters.

Next, we analyzed feature redundancy. Principal Component Analysis (PCA) indicated that most of the data variation could be captured using two components, suggesting potential redundancy among features. However, Pearson correlation analysis and Variance Inflation Factor (VIF) calculations did not flag any redundant features. Some features, as indicated in Table 2, had condition numbers greater than 30, signaling potential risk, which aligns with the PCA observations. The homogeneity scores, presented in Table 3, reveal that the features feature\_xcnfirm\_segment\_median\_cum\_event\_qty\_gradient, feature\_xcnfirm\_segment\_mean\_liquidity, and focus\_firm\_same\_side\_liquidity\_taking\_qty exhibited relatively high homogeneity scores with other features in the model.

The model developed for Resting Order (RO) is based on XGBoost. To evaluate the model's reproducibility, the training pipeline was executed ten times, and the outputs were compared. The analysis of the model's loss demonstrated reproducibility. Hyperparameters were optimized using HyperOpts. The confusion matrix for the model is presented in Figure 5, while the AUC is displayed in Figure 6.

Overall, the model meets its performance objectives as outlined in the model card. Table 4 provides a comprehensive list of performance metrics for both the training and testing sets. We also examined the model's performance across different clusters. Figure 7 indicates that most false negatives are concentrated within a single cluster, suggesting a potential pattern in these errors. To investigate this, we trained a Decision Tree Classifier, achieving 84% accuracy in identifying false negative cases. Figure 8 illustrates the distribution of precision and recall across the clusters, revealing a drop in recall in cluster 1 and a decline in precision in cluster 2. Additionally, K-fold cross-validation was employed, with results shown in Table 5.

We also assessed the model's robustness against adversarial samples. For each sample and feature, we altered the feature value by increasing and decreasing it once. We recorded how often the sample label changed, marking samples with label changes as edge cases. A total of 0.01% of experiments resulted in edge cases. The most vulnerable features identified were feature\_xcnfrm\_total\_fdid and focus\_firm\_same\_side\_exec\_cnt, indicating that minor adjustments in sample values led to changes in predicted labels.

To analyze feature importance, the model was processed with sklearn. Figure 9 displays the importance of the model features, highlighting feature\_xcnfrm\_total\_fdid and focus\_firm\_ro\_exec\_qty as significant, while feature\_xcnfirm\_segment\_mean\_liquidity and focus\_firm\_same\_side\_exec\_cnt were found to be the least important. We also conducted Partial Dependence Plots (PDP) in Figure 10, Individual Conditional Expectation (ICE) in Figure 11, and Accumulated Local Effects (ALE) in Figure 12. Additionally, SHAP analysis flagged feature\_xcnfirm\_segment\_mean\_liquidity and focus\_firm\_same\_side\_exec\_cnt as the two least important features, as illustrated in Figure 13. This finding was corroborated by LIME analysis.

The calibration curve of the model, shown in Figure 14, demonstrates reasonable performance, closely aligning with the expected line.

Finally, to evaluate the adequacy of alternative modeling approaches, we trained several other classifiers and compared their results with the XGBoost model utilized in this study. The findings revealed that the XGBoost model outperformed the other analyzed models.