MACHINE LEARNING IN FINANCE

FIN-407

Machine Learning Project

Salim AMEZIANE (330575)
Charaf Eddine DAHBI (330063)
Rami MJALLI (389790)
Victor NAHOUL (339407)
Jeffrey RACHED (341551)

Lausanne, Switzerland
June 10th 2025

Contents

1 Introduction and problem statement **2**

2 Description of Data Preprocessing and Feature Engineering **2**
- 2.1 Label Construction . . . 2
- 2.2 Feature Engineering . . . 2
- 2.3 Train-Test Split . . . 3

3 Predictive Models Used **3**
- 3.1 Baseline Models . . . 3
- 3.2 Gradient Boosting: XGBoost . . . 3
- 3.3 Neural Networks: MLP Architectures . . . 3
- 3.4 Thresholding and Abstention Mechanism . . . 4
- 3.5 Stacked Ensemble: XGBoost + MLP . . . 4

4 Evaluation Methodology and Performance Metrics **4**

5 Results and Discussion **5**
- 5.1 Model Comparison . . . 5
- 5.2 Detailed Analysis: Tuned XGBoost Model . . . 5
- 5.3 Prediction Confidence and Probability Distributions . . . 6
- 5.4 Precision-Recall Tradeoff . . . 6
- 5.5 Discriminative Ability: ROC and AUC . . . 6
- 5.6 Probability Calibration . . . 6
- 5.7 Explainability and Feature Relevance . . . 6
- 5.8 Robustness to Initialization . . . 6
- 5.9 Performance of PyTorch MLP Classifier . . . 7
- 5.10 MLP Classifier with XGBoost-inspired Thresholds . . . 7
- 5.11 Soft-Labeled Ensemble (XGBoost + MLP) . . . 7
- 5.12 Comparison of Predicted Probabilities . . . 7

6 Proposed Novel Model: Stacked MLP + XGBoost **7**

7 Conclusion **8**

8 Appendix **9**

1 Introduction and problem statement

Forecasting financial market regimes (periods characterized by persistent bullish or bearish behavior) is a central challenge in quantitative finance. Accurately identifying such regimes can significantly improve asset allocation, risk management, and trading strategies. In our project, we propose a machine learning pipeline to predict monthly U.S. equity market direction (bullish or bearish), using historical return patterns and interest rate data as predictors.

We frame the task as a binary classification problem, where the target variable reflects the aggregate S&P 500 excess return over the final 10 trading days of each month. A positive return signifies a bullish regime, while a negative return indicates a bearish one. To construct predictive features, we aggregate market behavior over the first 10 trading days of each month (computing summary statistics such as mean, standard deviation, and cumulative return) along with contemporaneous interest rate levels and changes.

We will compare a variety of models, including logistic regression, random forests, XGBoost, and deep learning MLPclassifiers. Beyond traditional metrics such as accuracy and F1-score, we emphasize confident predictions: we categorize outputs as bullish, bearish, or neutral depending on the predicted probability's distance from a central threshold. This design reflects the high cost of false signals in financial decision-making and introduces the possibility of abstaining from predictions under uncertainty.

2 Description of Data Preprocessing and Feature Engineering

To build a meaningful regime prediction model, we construct a monthly dataset that captures both return dynamics and macroeconomic signals. The raw inputs consist of:

- Daily excess returns of the S&P 500 index, extracted from `daily_crsp.csv` (column `sprtrn`).
- Daily U.S. interest rates from the Federal Reserve H.15 release, obtained from `FRB_H15.csv`.

2.1 Label Construction

Our target variable is a binary label indicating the market regime at the end of each calendar month:

- Bullish (1) if the cumulative excess return over the final 10 calendar days of the month is positive.
- Bearish (0) otherwise.

We first aggregate the daily excess returns by calendar date and convert them to a time series. For each month, we calculate the sum of the last 10 available daily excess returns. The resulting regime labels are timestamped to the end of each month. This choice reflects the idea that end-of-month returns signal the prevailing market regime.

2.2 Feature Engineering

To avoid look-ahead bias, all predictive features are computed using only information available at the start of each month.

Return-based features are calculated from the first 10 trading days of each calendar month:

- Mean daily return (`mean_ret_10d`)
- Standard deviation of returns (`std_ret_10d`)
- Minimum and maximum daily returns (`min_ret_10d`, `max_ret_10d`)
- Cumulative return (`sum_ret_10d`)

This 10-day window captures short-term momentum and volatility patterns that may carry predictive power about the end-of-month regime.

Macroeconomic features are derived from interest rate data:

- Interest rate level at month-start (`IR`)
- First difference: monthly change in interest rate during previous month (`IR_change`)

These features incorporate monetary policy signals and expectations, which can influence equity markets.

All features are merged into a single monthly panel indexed by calendar month-end. Observations with missing values (typically due to unavailable early-month or end-of-month data) are excluded from the final dataset.

Feature Scaling: Prior to model training, continuous features are standardized using z-score normalization. This step is especially important for models sensitive to feature scale, such as neural networks and regularized classifiers.

2.3 Train-Test Split

To simulate a realistic forecasting scenario, we partition the dataset chronologically. The first 80% of monthly observations are used as the training set. The remaining 20% form the test set. This time-based split ensures that future information is not leaked into the training process and mirrors a real-world backtest.

3 Predictive Models Used

This section outlines our modeling choices, architectures, and training procedures.

3.1 Baseline Models

We begin with standard classifiers as performance baselines:

- **Logistic Regression**: A linear classifier trained using L2 regularization and balanced class weights. It serves as a benchmark for linear separability of the feature space.
- **Random Forest**: A nonparametric ensemble of decision trees, trained with balanced class weights and 100 estimators. It captures nonlinear interactions without strong assumptions on feature distributions.

Both models are evaluated on the test set using accuracy, F1-score, and confusion matrices. However, they lack the ability to express uncertainty in their predictions or adapt well to complex decision boundaries.

3.2 Gradient Boosting: XGBoost

To improve performance and interpretability, we implement XGBoost, a tree-based gradient boosting algorithm. XGBoost is well-suited for tabular datasets and offers fine-grained control over model complexity.

Hyperparameters: The final XGBoost model uses the following configuration:

- `max_depth` = 3, `learning_rate` = 0.05
- `n_estimators` = 400
- `scale_pos_weight` = 0.42 (to address class imbalance)
- `reg_alpha` = 0.2, `reg_lambda` = 1

We also combine multiple XGBoost models trained with different seeds to quantify prediction variability. The standard deviation of predicted probabilities across seeds is plotted to illustrate uncertainty.

We use SHAP (SHapley Additive exPlanations) values to identify the most influential features and visualize global and local model explanations. In addition, XGBoost's built-in feature importance (by gain) is plotted for comparison.

3.3 Neural Networks: MLP Architectures

To capture complex nonlinear patterns and interactions, we implement several multilayer perceptron (MLP) architectures using PyTorch. The models vary in depth, activation functions, and loss design.

Standard MLP: Our base MLP consists of two hidden layers with ReLU activations and dropout for regularization:

- Layers: $64 \to 32 \to 1$
- Activation: ReLU, final layer with Sigmoid
- Optimizer: Adam with learning rate 0.001
- Loss: Binary Cross-Entropy (BCE)

Soft-Label MLPs: We implement a variant with label smoothing, replacing hard labels (0/1) with soft targets (e.g., 0.2/0.8) to regularize the model and reduce overconfidence. This MLP uses Mean Squared Error (MSE) as the loss function and produces smoother probability distributions.

Advanced MLPs: We experiment with additional refinements:

- Batch normalization and Leaky ReLU for improved convergence.
- Oversampling of the minority (bearish) class to balance training batches.
- Asymmetric Focal Loss to penalize misclassified bearish signals more heavily. This encourages the model to focus on underrepresented but important cases.

3.4 Thresholding and Abstention Mechanism

We frame our classification task using a three-class confidence-based regime system:

- **Bullish (1):** predicted probability $P(\text{Bullish}) > \theta_{\text{high}}$
- **Bearish (0):** predicted probability $P(\text{Bullish}) < \theta_{\text{low}}$
- **Neutral:** if $P(\text{Bullish}) \in [\theta_{\text{low}}, \theta_{\text{high}}]$

Typical threshold values are $\theta_{\text{low}} = 0.4$ and $\theta_{\text{high}} = 0.6$, unless otherwise stated. Predictions classified as Neutral are excluded from metric computation. This approach reduces the likelihood of overconfident errors in ambiguous market conditions and aligns with real-world risk management practices.

3.5 Stacked Ensemble: XGBoost + MLP

To combine the interpretability of XGBoost with the flexibility of neural networks, we implement a two-stage stacked ensemble. The pipeline proceeds as follows:

1. Train XGBoost on a sub-sample of the training set and obtain out-of-sample predicted probabilities.
2. Use these XGBoost probabilities as an additional feature when training a downstream MLP on the remaining data.
3. The final model is evaluated on the test set with XGBoost features included.

This architecture allows the MLP to adaptively correct XGBoost errors and model residual nonlinearities. The ensemble consistently achieves superior performance compared to either model alone.

4 Evaluation Methodology and Performance Metrics

Accurate evaluation of financial prediction models requires not only standard classification metrics, but also uncertainty-aware performance analysis. In this section, we detail the methodology used to evaluate our models, including metrics, validation protocols, and abstention handling.

Classification Metrics: The following standard metrics are computed over the set of confident predictions:

- **Accuracy:** Proportion of correct predictions among all confident predictions.
- **Precision:** Correct bullish predictions as a fraction of all predicted bullish outcomes.
- **Recall:** Correct bullish predictions as a fraction of actual bullish outcomes.
- **F1-score:** Harmonic mean of precision and recall.

We report separate metrics for the Bullish and Bearish classes to assess model symmetry and bias. Confusion matrices are also provided for visual clarity.

Abstention Rate: We compute the proportion of samples for which the model abstains from making a confident prediction:

$$\text{Abstention Rate} = \frac{\#\text{ Neutral predictions}}{\text{Total predictions}}$$

A higher abstention rate indicates greater uncertainty or conservatism, while a lower rate suggests higher model confidence. Comparisons account for both accuracy and abstention.

ROC Curve and AUC: To assess discrimination between bullish and bearish regimes across thresholds, we use the ROC curve and its Area Under the Curve (AUC). Higher AUC values (closer to 1) reflect stronger discriminatory power. This threshold-independent measure complements our confidence-aware metrics.

Probability Calibration: Calibration is evaluated using reliability diagrams, which compare predicted probabilities to observed outcomes. A well-calibrated model aligns closely with the diagonal, indicating reliable confidence estimates that are crucial in financial contexts.

Prediction Variability Across Seeds: To assess robustness, we train each model with multiple random seeds and compute the standard deviation of predicted probabilities. We visualize the distribution of these values to gauge prediction stability. High variability may suggest overfitting or sensitivity to initialization.

Feature Importance and Explainability: To interpret model behavior:

- **XGBoost Gain-Based Importance** ranks features by their average contribution to tree splits.
- **SHAP Values** offer both local and global insights by quantifying each feature's marginal impact on predictions.

These tools enhance transparency and ensure meaningful use of macro and return-based features.

5 Results and Discussion

This section presents the empirical performance of the models described earlier, including baseline classifiers, XGBoost variants, and confidence-aware thresholding strategies.

5.1 Model Comparison

Table 1 summarizes the performance of several models on the test set. All metrics are computed over confident predictions only, with abstentions excluded. The proportion of abstentions is also reported.

Model	Accuracy	Precision	Recall	F1-score	Abstained
Logistic Regression	0.62	0.71	0.69	0.70	0 / 56
Random Forest	0.57	0.63	0.81	0.71	0 / 56
XGBoost (basic)	0.65	0.73	0.75	0.74	8 / 56
XGBoost (tuned)	0.68	0.75	0.72	0.73	16 / 56

Table 1: Performance Comparison Across Models (Confident Predictions Only)

Among all models tested, the tuned XGBoost classifier with abstention (thresholds at 0.4 and 0.6) yields the highest overall accuracy (68%) on confident predictions. It also achieves a strong balance between precision and recall for both classes. Although it abstains from 16 predictions, this mechanism avoids uncertain classifications and improves reliability.

5.2 Detailed Analysis: Tuned XGBoost Model

We examine the performance of the tuned XGBoost model in greater depth. The confusion matrix and classification report are shown below:

- **Confusion matrix:**

$$\begin{bmatrix} 9 & 6 \\ 7 & 18 \end{bmatrix}$$

- **Accuracy (confident predictions):** 68%
- **Neutral predictions:** 16 out of 56 (29% abstained)
- **F1-score:** 0.73 (bullish), 0.58 (bearish)

While the model exhibits slightly better recall for bullish signals, its performance on bearish cases remains robust, especially when compared to earlier models that exhibited strong bias toward the majority class (bullish months comprise approximately 61% of the dataset).

5.3 Prediction Confidence and Probability Distributions

To visualize model certainty, we plot the histogram of predicted probabilities. As shown in Figure 2, the tuned XGBoost model produces a relatively bimodal distribution, with distinct clusters near 0 and 1. This allows confident classification and validates the usefulness of abstention thresholds.

In contrast, the less-regularized model with default hyperparameters (Figure 1) exhibits a sharper skew toward extreme probabilities, which can lead to overconfidence and misclassification under uncertainty.

5.4 Precision-Recall Tradeoff

Figure 3 shows the precision-recall curve for the bullish class. The curve indicates that the model maintains strong precision across a wide range of recall levels. This reinforces the decision to focus on high-confidence predictions rather than optimize only for global accuracy.

5.5 Discriminative Ability: ROC and AUC

The receiver operating characteristic (ROC) curve in Figure 4 evaluates the classifier's ability to discriminate between bullish and bearish regimes at all thresholds. The area under the curve (AUC) is 0.64, indicating modest discriminative performance beyond random guessing (AUC = 0.5). While not exceptionally high, this level of separation is consistent with expectations given the noisy and partially efficient nature of financial markets.

5.6 Probability Calibration

Figure 6 presents the calibration curve for the tuned XGBoost model. While probabilities in the mid-range are well aligned with true frequencies, the model tends to slightly overestimate bullish likelihoods in higher bins. Nevertheless, the overall calibration is acceptable, supporting the use of probability thresholds (rather than hard 0.5 cutoffs) in our abstention mechanism.

5.7 Explainability and Feature Relevance

We examine both global and local interpretability using model introspection tools:

- **Feature importance by gain** (Figure 7) shows that return volatility (`std_ret_10d`) is the most influential feature in XGBoost splits, followed by `min_ret_10d`, `max_ret_10d`, and interest rate level (`IR`).
- **SHAP values** (Figure 8) provide a more granular view of how individual features push predictions toward the bullish or bearish class. The directionality of SHAP contributions suggests, for example, that higher interest rates and high early-month volatility are associated with bearish outcomes.

These findings are consistent with financial intuition: unstable markets and tighter monetary conditions tend to coincide with lower expected returns.

5.8 Robustness to Initialization

To assess prediction stability, we retrained XGBoost five times with different random seeds and computed the standard deviation of predicted probabilities across seeds for each test sample. Figure 9 displays the histogram of these standard deviations.

Remarkably, prediction variability is negligible: over 85% of test samples have near-zero standard deviation (on the order of 10^{-8}). This suggests that the model is robust to training noise and that predictions are reproducible, an important consideration in high-stakes financial environments.

5.9 Performance of PyTorch MLP Classifier

We first evaluated a basic Multi-Layer Perceptron (MLP) trained with binary cross-entropy loss and standard thresholding at 0.4 and 0.6. The model demonstrated relatively strong performance on bullish regimes, with a precision of 0.64 and recall of 1.00, but failed to identify any bearish observations (precision and recall of 0.00). The confusion matrix shows.

- Overall accuracy on confident predictions: 64%
- Abstention rate: 3 out of 56 test samples

The associated probability distribution (Figure 10) shows highly concentrated predictions above 0.6, indicating model overconfidence toward the bullish class. This highlights the importance of addressing class imbalance and improving calibration.

5.10 MLP Classifier with XGBoost-inspired Thresholds

To mitigate the overconfidence, we retrained the MLP using class-weighted loss with `BCEWithLogitsLoss`, incorporating the same class imbalance ratio (`scale_pos_weight` = 0.42) used in the XGBoost configuration. We also adjusted the thresholds to 0.3 (bearish) and 0.57 (bullish) based on the XGBoost model's posterior distribution.

However, this configuration overcorrected in favor of the bearish class, leading to a complete failure in identifying bullish observations: the model predicted only bearish classes with no correct positive classifications. Accuracy dropped to 36%. The corresponding probability distribution (Figure 11) indicates extremely narrow output ranges, again suggesting overconfident and poorly calibrated behavior.

We also implemented a custom loss function called Asymmetric Focal Binary Cross-Entropy (AF-BCE) to address the class imbalance and asymmetry in financial regime classification. This loss extends the standard focal loss by introducing separate weighting terms for bearish and bullish classes. Specifically, the parameter α amplifies the penalty on misclassified bearish observations, while β controls the contribution of bullish cases. The focal component, modulated by γ, emphasizes harder-to-classify examples by down-weighting well-classified predictions. The combination of asymmetric weighting and focal scaling allows the model to focus on rare but critical bearish events while maintaining stable learning on the majority bullish class.

5.11 Soft-Labeled Ensemble (XGBoost + MLP)

To combine the strengths of both architectures, we implemented a soft-labeled ensemble model. The MLP was trained on smoothed target labels ($y = 0.2$ for bearish, $y = 0.8$ for bullish) using mean squared error loss, encouraging probabilistic predictions. The final ensemble output was constructed as a weighted average:

$$P_{\text{ensemble}} = 0.8 \cdot P_{\text{XGB}} + 0.2 \cdot P_{\text{MLP}}$$

This ensemble achieved the best balance between recall and precision, correctly identifying both bullish and bearish cases. Its confident prediction accuracy reached 69.4%, with an abstention rate of 20 out of 56. The ensemble probability distribution (Figure 12) demonstrated a broader and more stable distribution across thresholds, suggesting improved model calibration.

5.12 Comparison of Predicted Probabilities

Figure 13 compares the predicted probability distributions of the MLP and XGBoost models. The MLP outputs are sharply peaked within a narrow band, whereas XGBoost provides more dispersed probability mass. The ensemble distribution benefits from this heterogeneity, achieving more robust confidence calibration.

6 Proposed Novel Model: Stacked MLP + XGBoost

The final model integrates both XGBoost and an MLP via a stacking approach. The predicted class probabilities from XGBoost are used as an input feature to the MLP, enabling the model to learn from both raw features and initial tree-based predictions. This section evaluates the final stacked model using two different thresholding strategies for classification and provides additional performance insights via ROC and probability correlation plots.

Classification Performance: Using standard thresholds of $[0.4, 0.6]$, the stacked model achieves an accuracy of 72% on confident predictions (excluding neutral class). The precision and recall for the bullish class (label 1) are 0.76 and 0.81 respectively, indicating that the model is particularly effective in capturing bullish signals. The bearish class (label 0), however, shows weaker recall (0.56), suggesting some false negatives. This is consistent with the observed class imbalance and the model's optimization strategy that prioritizes bullish recall. A total of 6 predictions are labeled as neutral, and 15 predictions fall below the 0.3 bearish confidence threshold (Figure 17).

When applying stricter thresholds of $[0.05, 0.95]$, model accuracy increases to 76.92%, but only 26 out of 56 predictions are classified as confident. The bullish class recall further improves to 0.94, while bearish recall drops to 0.38. This trade-off illustrates the impact of thresholding: higher confidence leads to fewer but more accurate predictions. Thirty instances are labeled neutral under this setup, reinforcing the conservative nature of the decision rule (Figure 18).

Probability Calibration and ROC Analysis: The ROC curve for the MLP (Figure 19) yields an AUC of 0.64, indicating modest discriminative power when used in isolation. While this is not outstanding, the combined model benefits from leveraging XGBoost's structured feature learning in conjunction with the MLP's representation capacity.

The scatter plot comparing XGBoost and MLP predicted probabilities (Figure 20) reveals a strong monotonic relationship, suggesting that the MLP effectively builds on the XGBoost output without overfitting or contradicting it. This correlation supports the rationale for stacking and validates the hybrid design.

Conclusion: The stacked model demonstrates competent predictive performance, particularly for bullish signals. While bearish detection remains more challenging, adjusting thresholds improves precision and overall confidence. The ROC and calibration analysis affirm that the ensemble captures meaningful patterns, and the high-confidence neutral filtering strategy enhances interpretability for decision-making in financial applications.

Updated Training with Holdout for Stacking

In this version, the training pipeline was modified to introduce a two-stage stacking procedure: XGBoost was trained on the first 60% of the training data, and its predicted probabilities were then used as an additional feature for the remaining 20%, which in turn trained the MLP. This out-of-fold strategy ensures that the MLP does not overfit to XGBoost's training predictions, promoting better generalization. The classification performance on the test set with a confidence interval of $[0.4, 0.6]$ shows an overall accuracy of 73.91%. The model achieves a recall of 87% and a precision of 76% on the bullish class, indicating its continued strength in detecting positive market signals. The bearish class remains less accurately detected, with only 4 true negatives and a recall of 50%. Notably, the model filtered out 33 out of 56 instances as neutral, demonstrating a high degree of caution. The low number of predictions falling below the bearish confidence threshold (only 2) suggests that the final model is generally less confident in bearish signals under this configuration. The histogram of predicted probabilities shows a narrower and more symmetric distribution centered around 0.55, reflecting a more conservative confidence profile compared to earlier versions (Figure 22).

7 Conclusion

In this project, we developed and evaluated several machine learning models to forecast monthly U.S. equity market regimes. Among them, the stacked MLP + XGBoost ensemble demonstrated the best trade-off between predictive accuracy and confidence, particularly for bullish signals. While bearish detection remains more challenging due to class imbalance, the abstention mechanism effectively reduced overconfident misclassifications. Our findings highlight the value of combining interpretable models with flexible deep learning architectures, especially in high-stakes financial decision-making.

8 Appendix

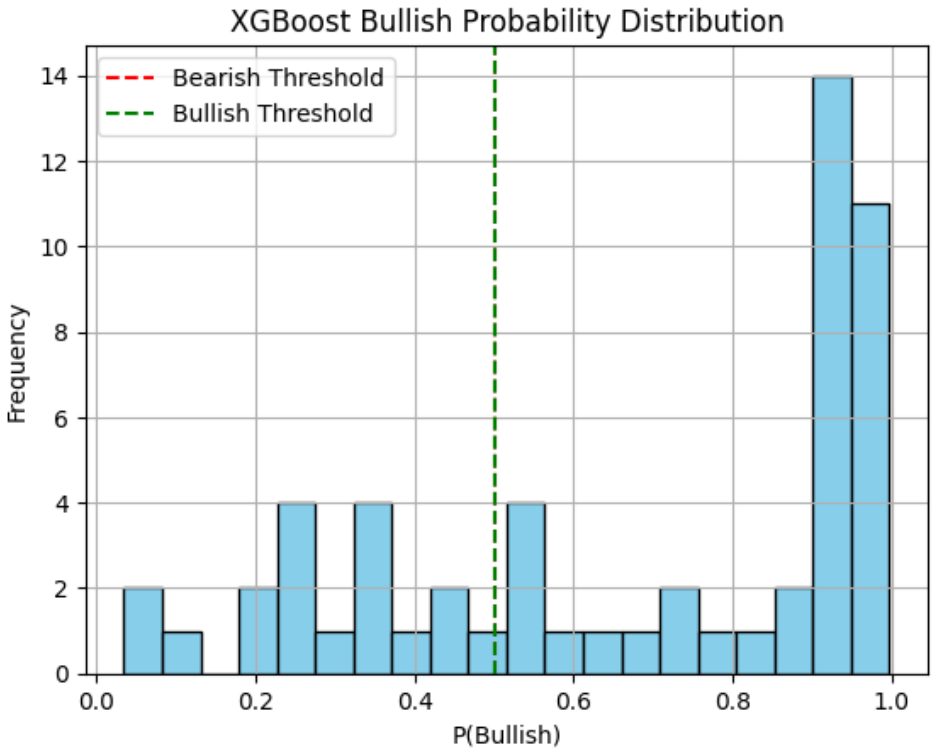

Figure 1: Basic XGBoost Bullish Probability Distribution

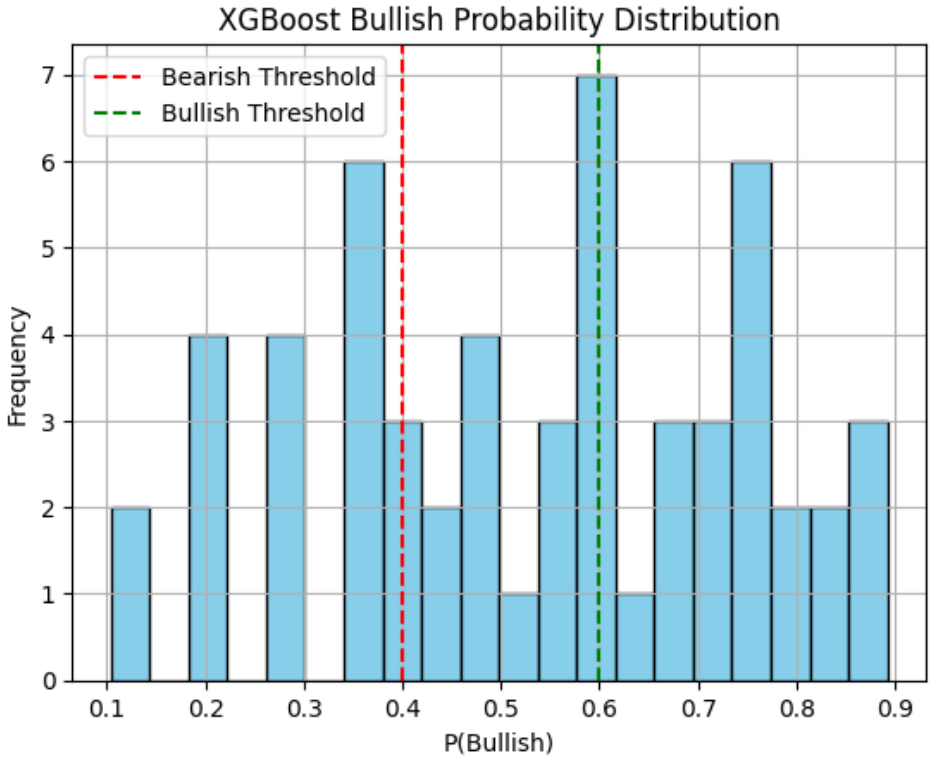

Figure 2: Tuned XGBoost Bullish Probability Distribution

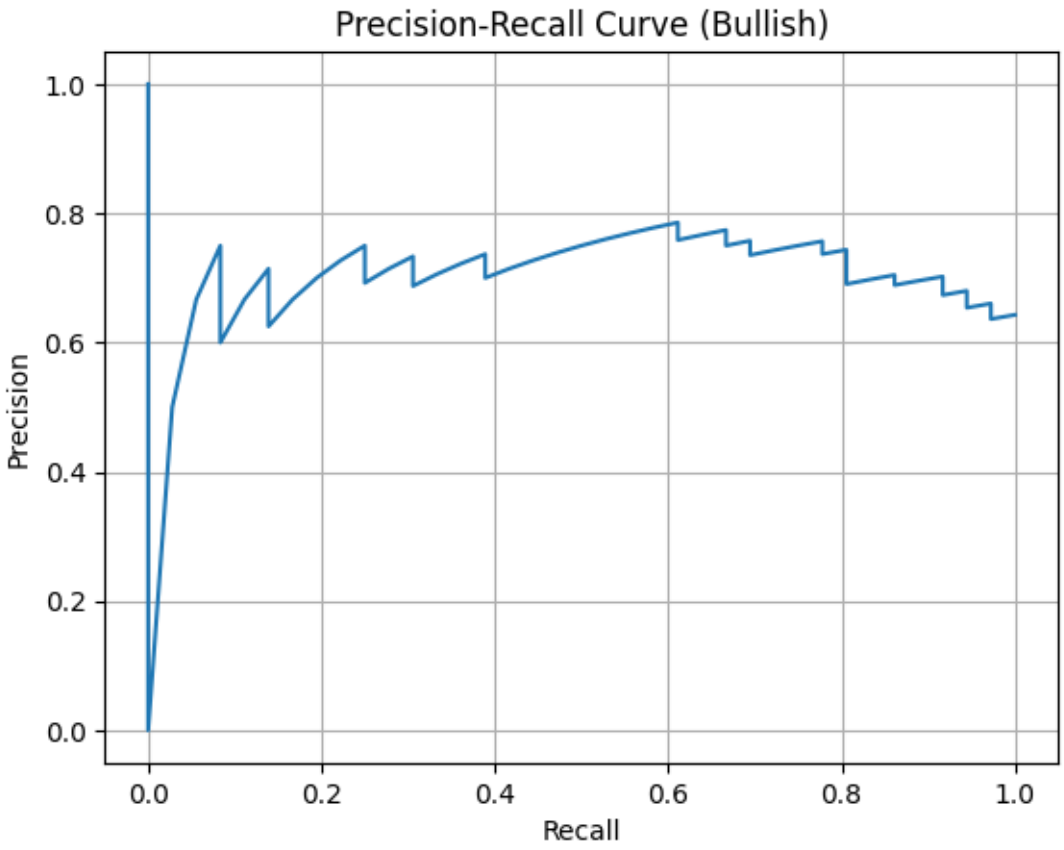

Figure 3: Precision-Recall Curve (Bullish)

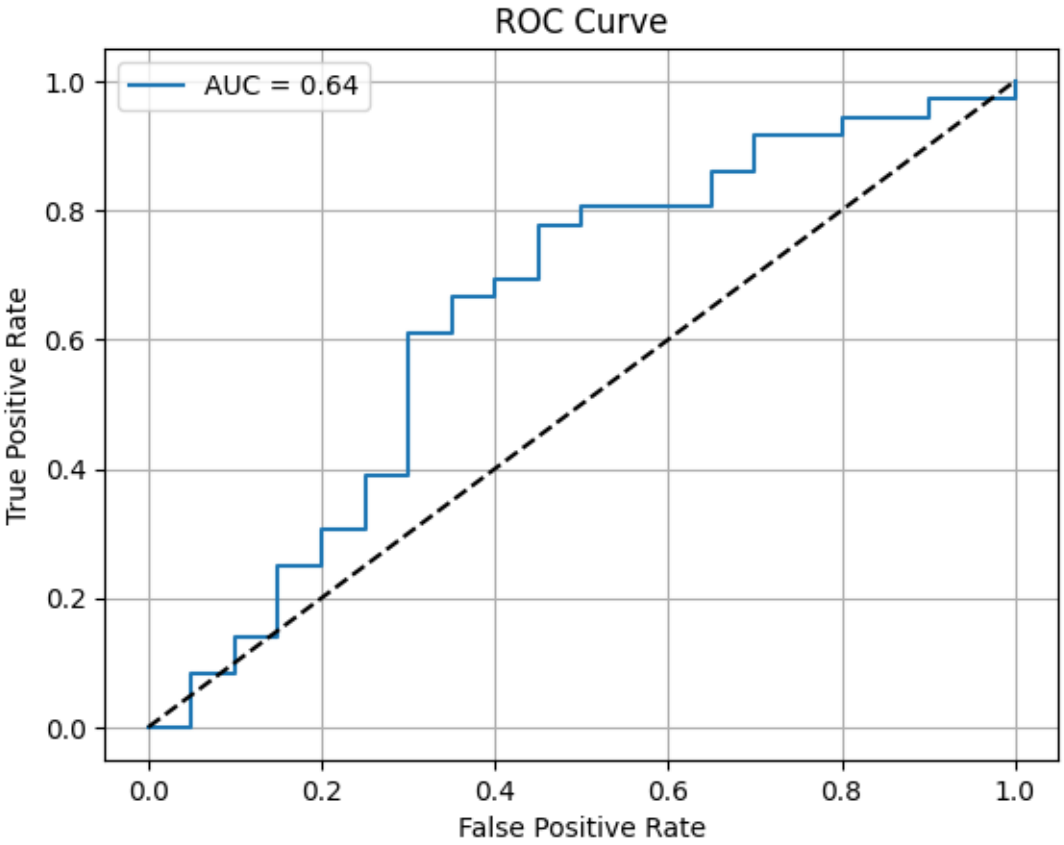

Figure 4: ROC Curve for Tuned XGBoost

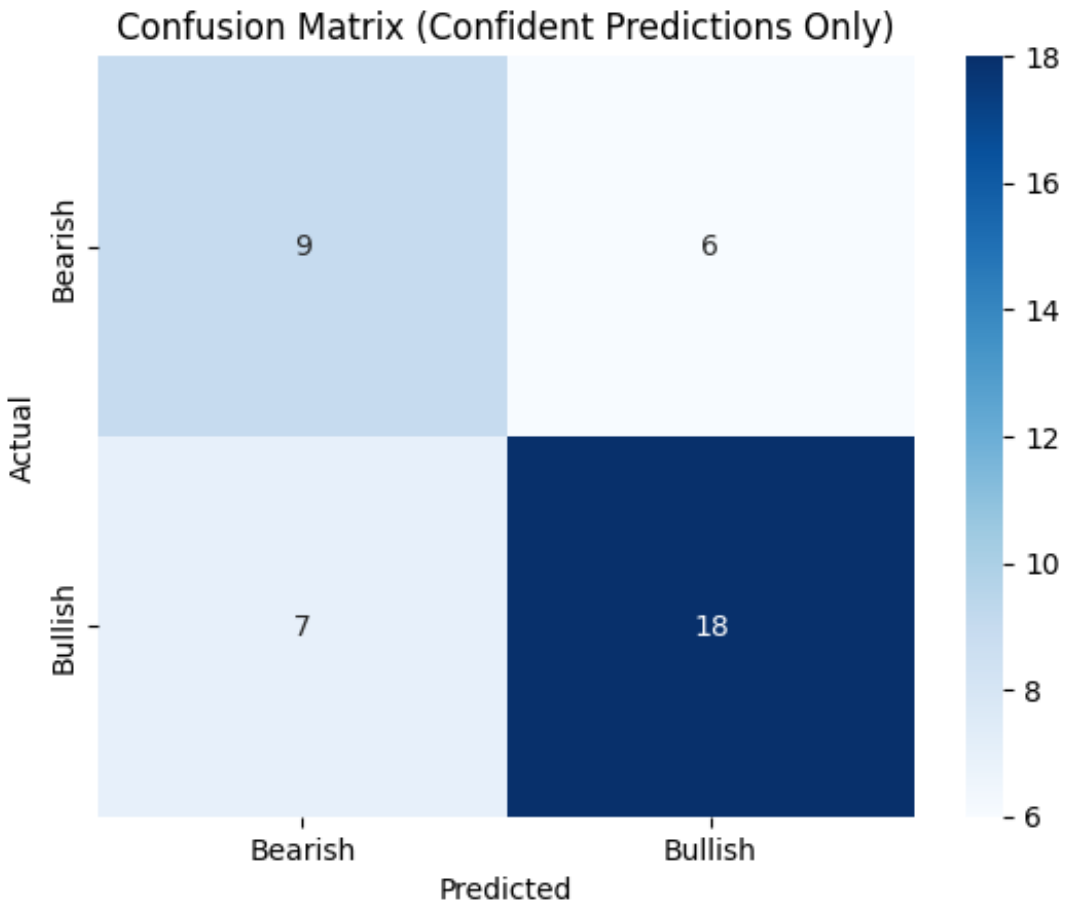

Figure 5: Confusion Matrix (Confident Predictions Only)

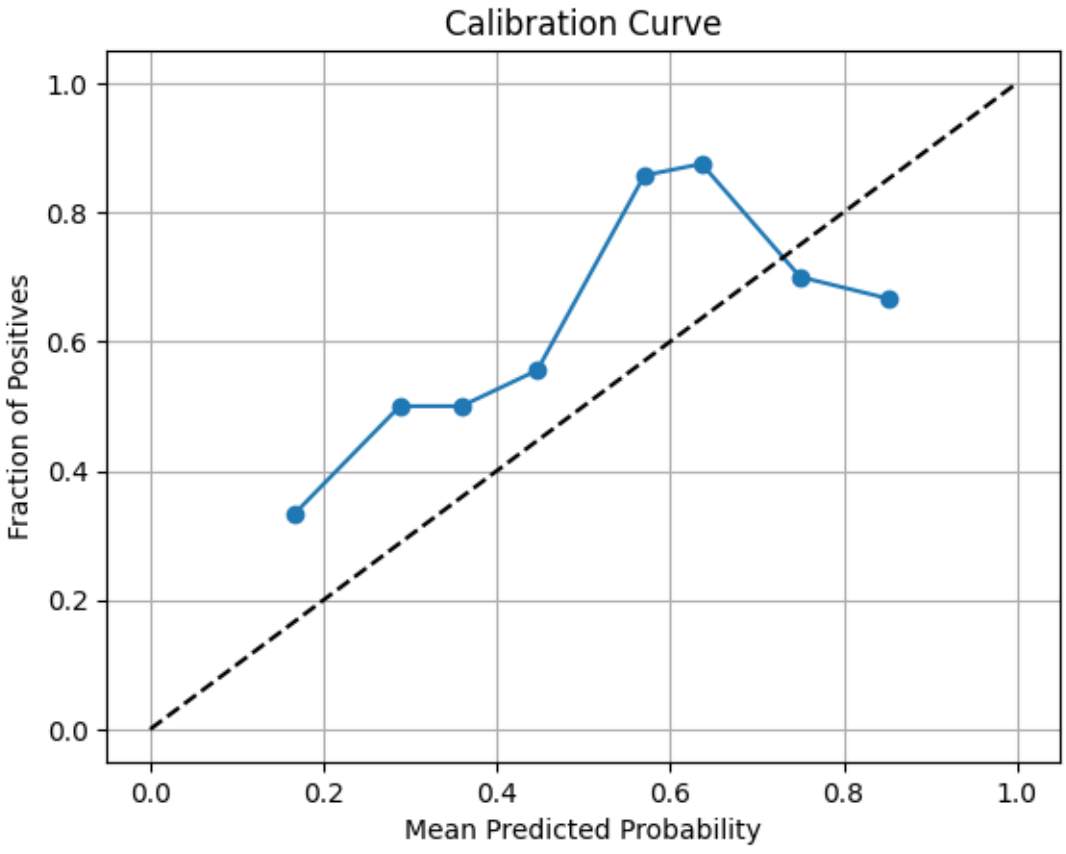

Figure 6: Calibration Curve

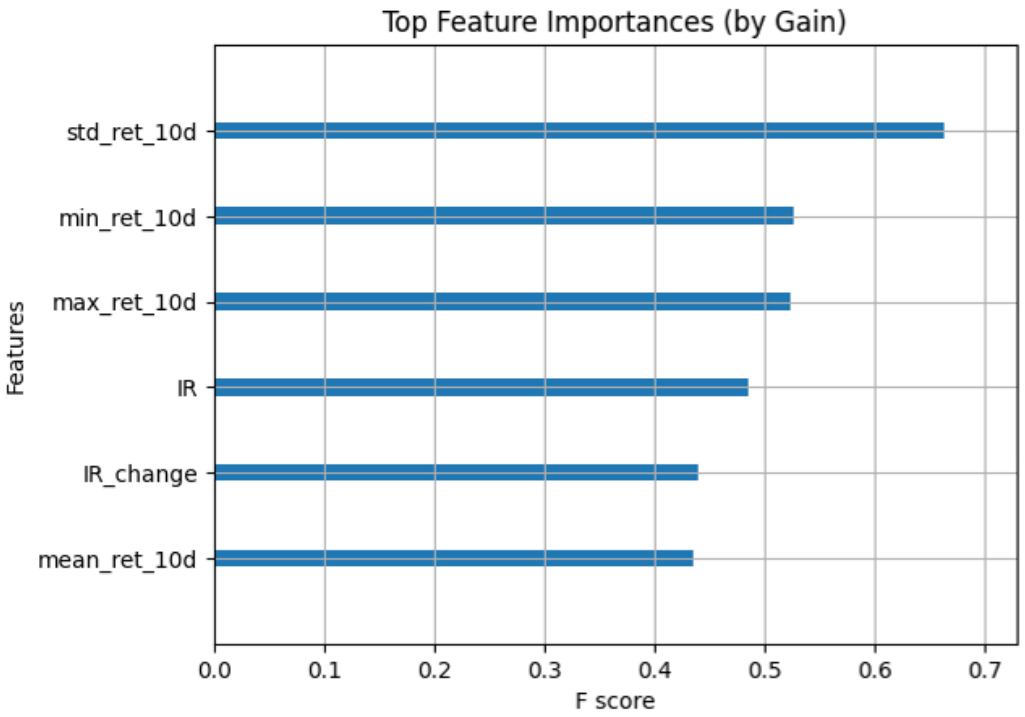

Figure 7: Top Feature Importances (by Gain)

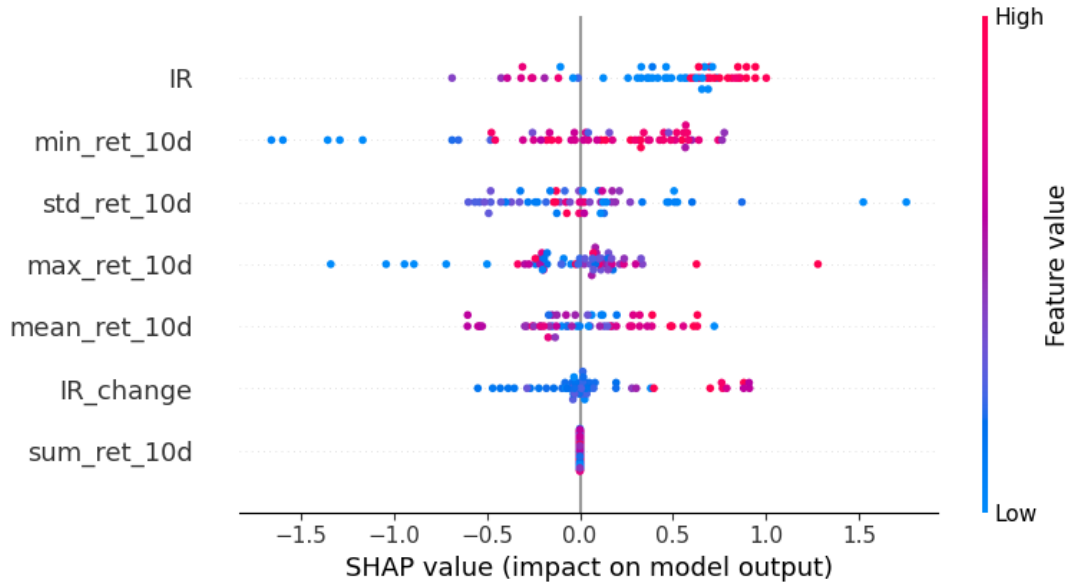

Figure 8: SHAP Value (impact on model output)

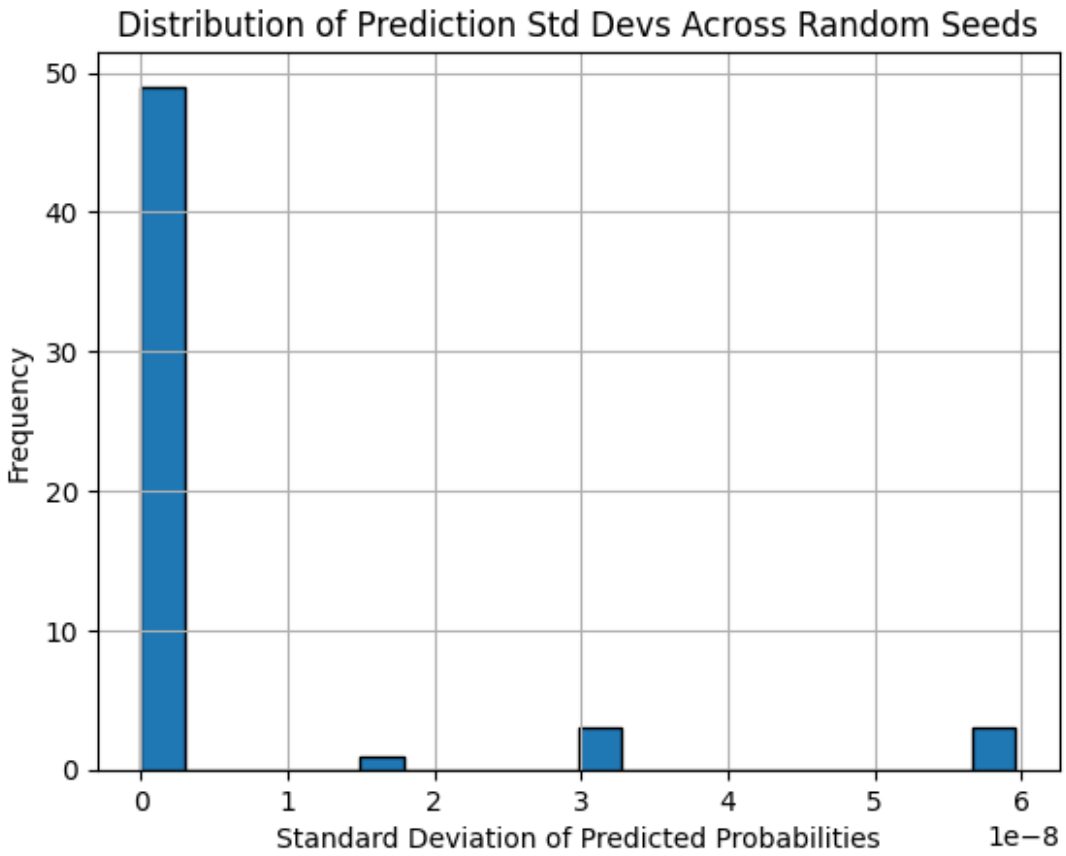

Figure 9: Distribution of Prediction Std Devs Across Random Seeds

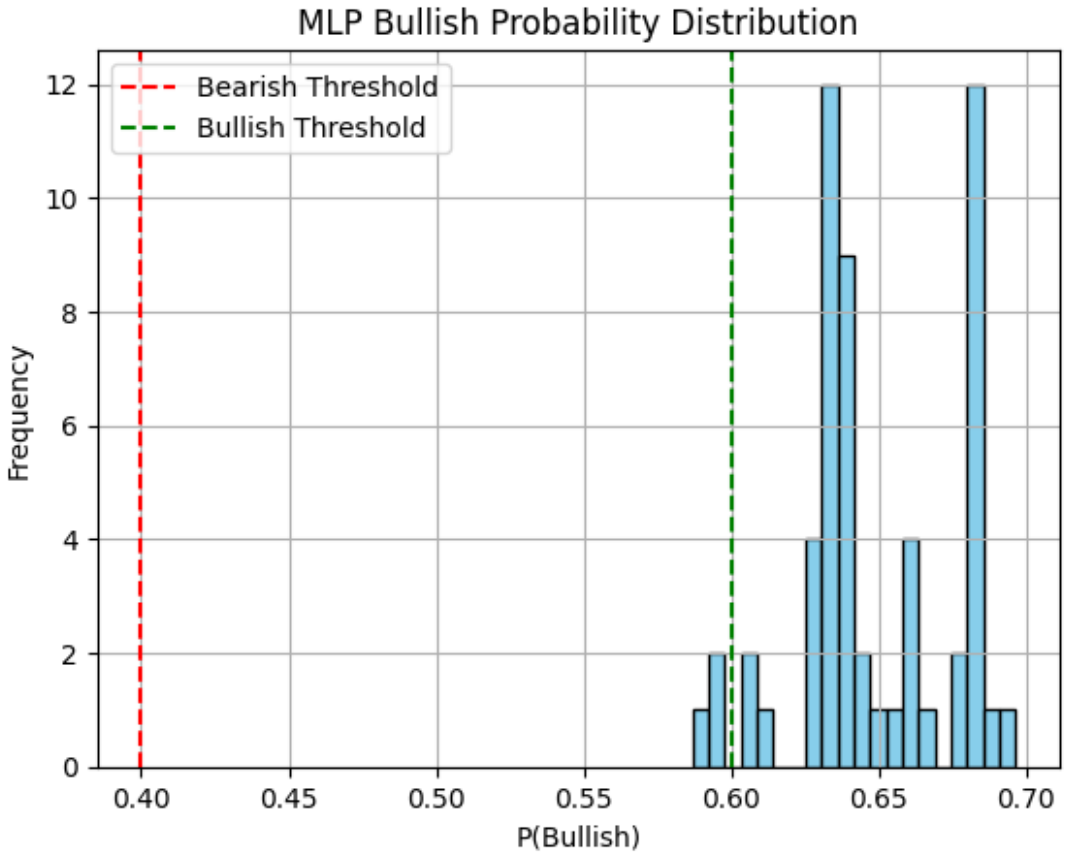

Figure 10: MLP Bullish Probability Distribution

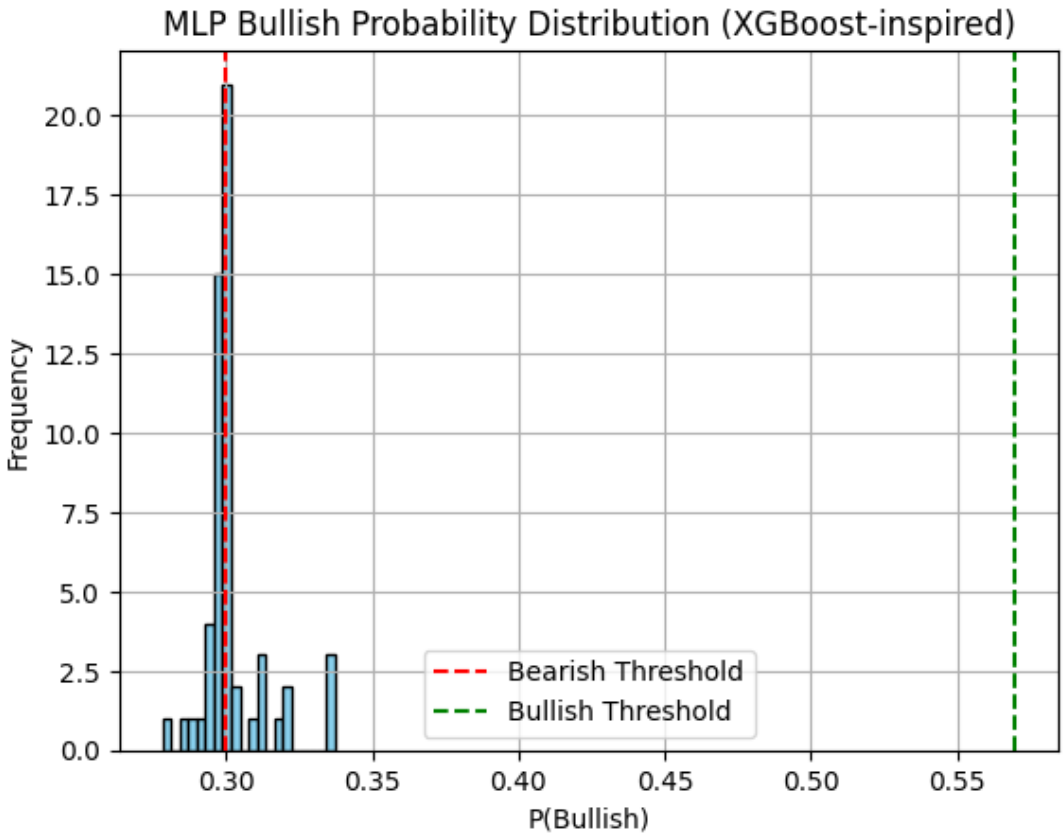

Figure 11: MLP Bullish Probability Distribution (XGBoost-inspired)

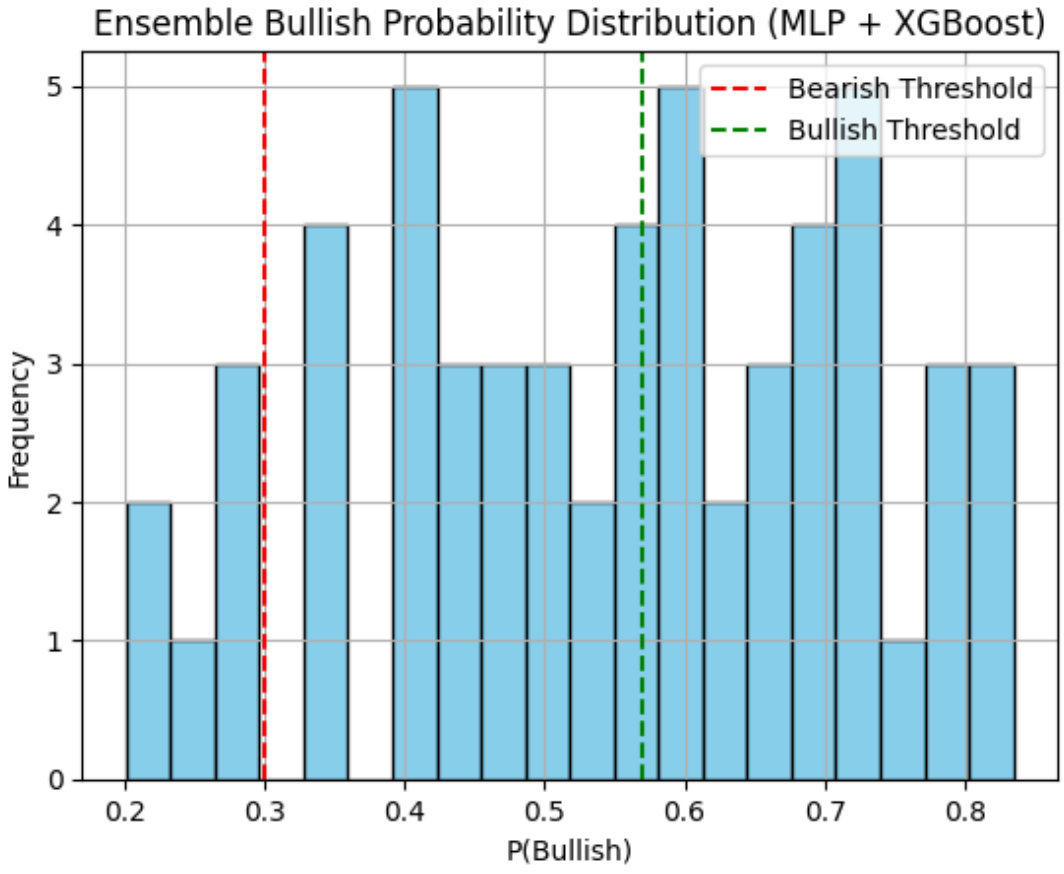

Figure 12: Ensemble Bullish Probability Distribution (MLP + XGBoost)

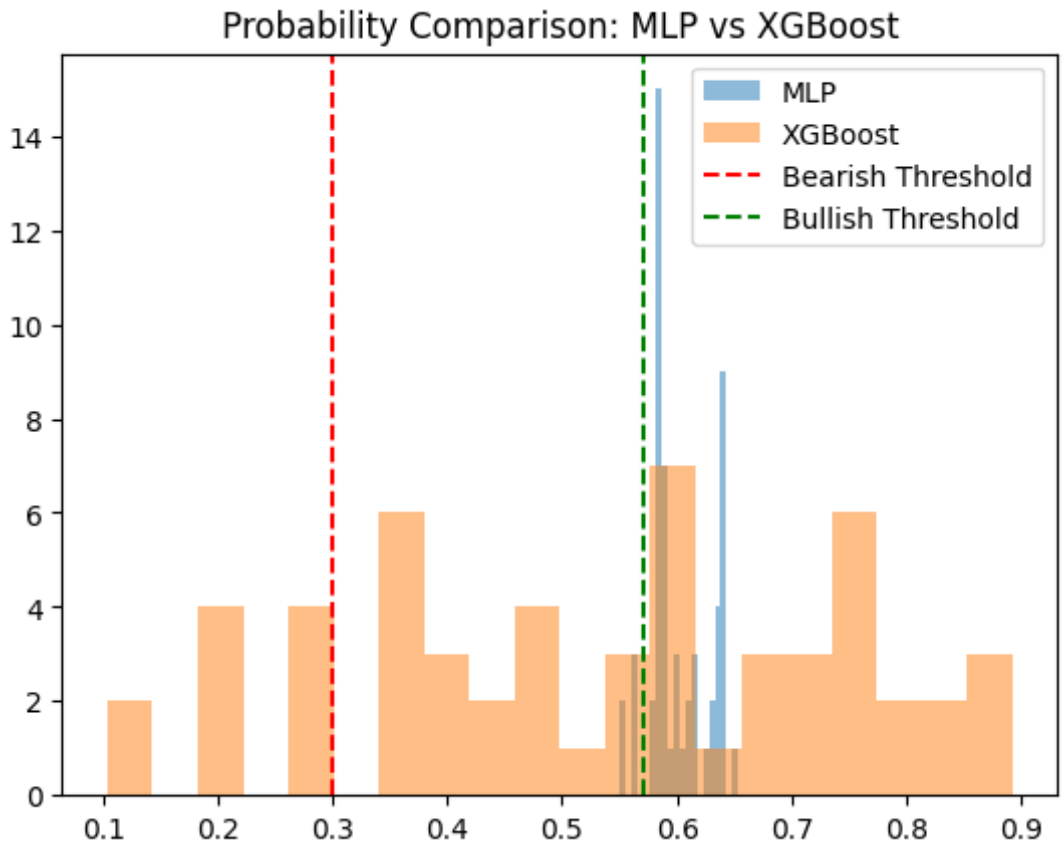

Figure 13: Probability Comparison: MLP vs XGBoost"

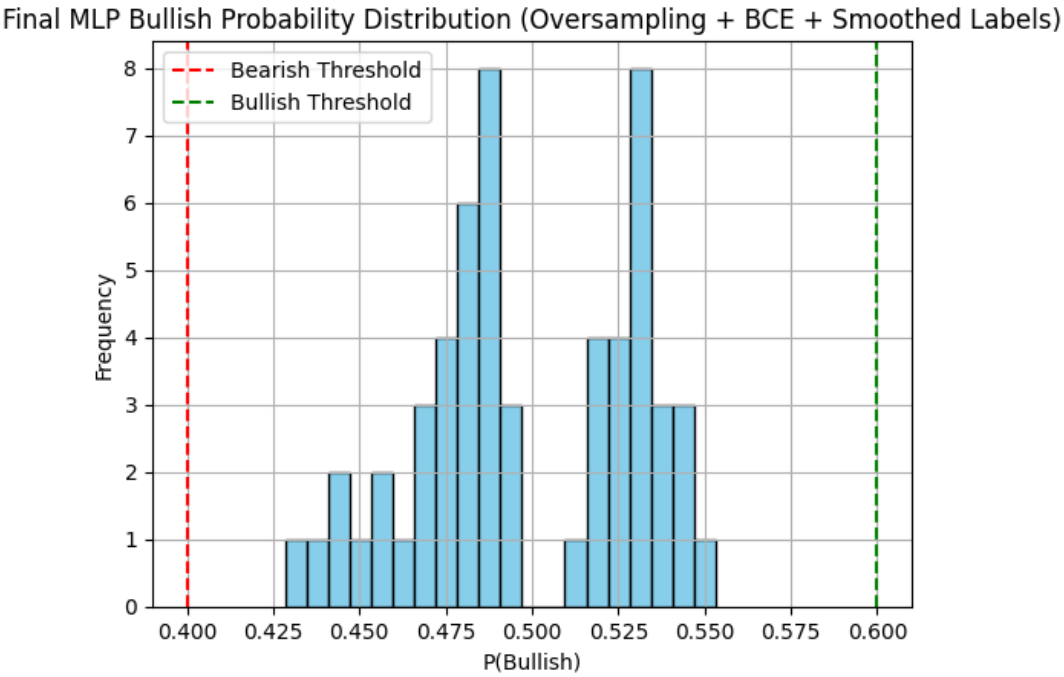

Figure 14: Final MLP Bullish Probability Distribution (Oversampling + BCE + Smoothed Labels)

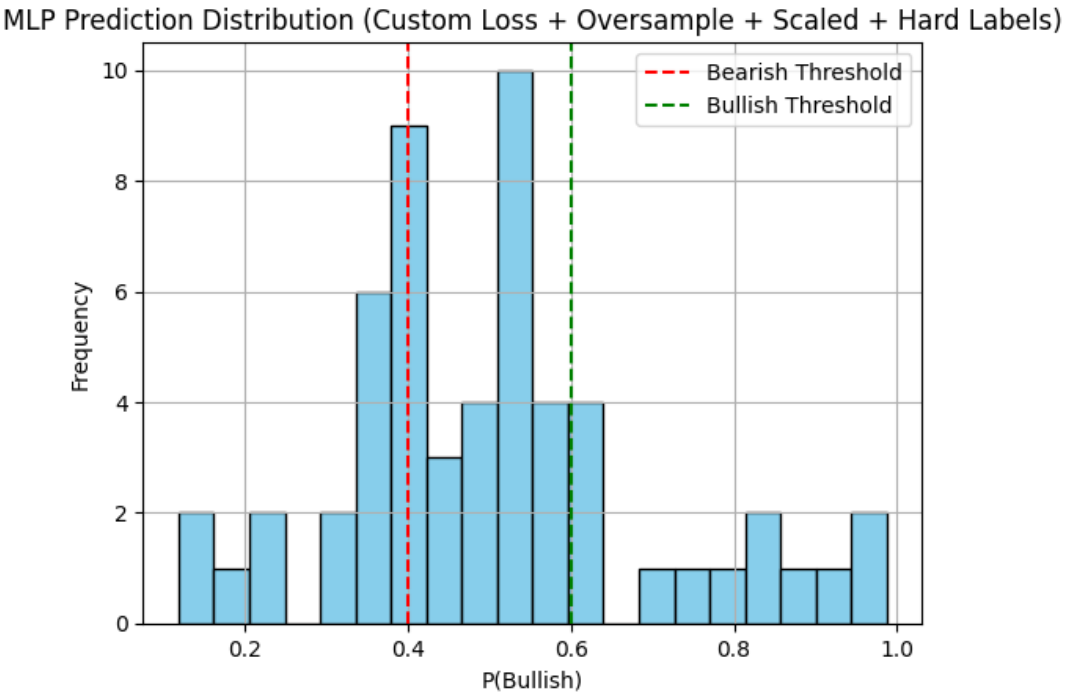

Figure 15: MLP Prediction Distribution (Custom Loss + Oversample + Scaled + Hard Labels)

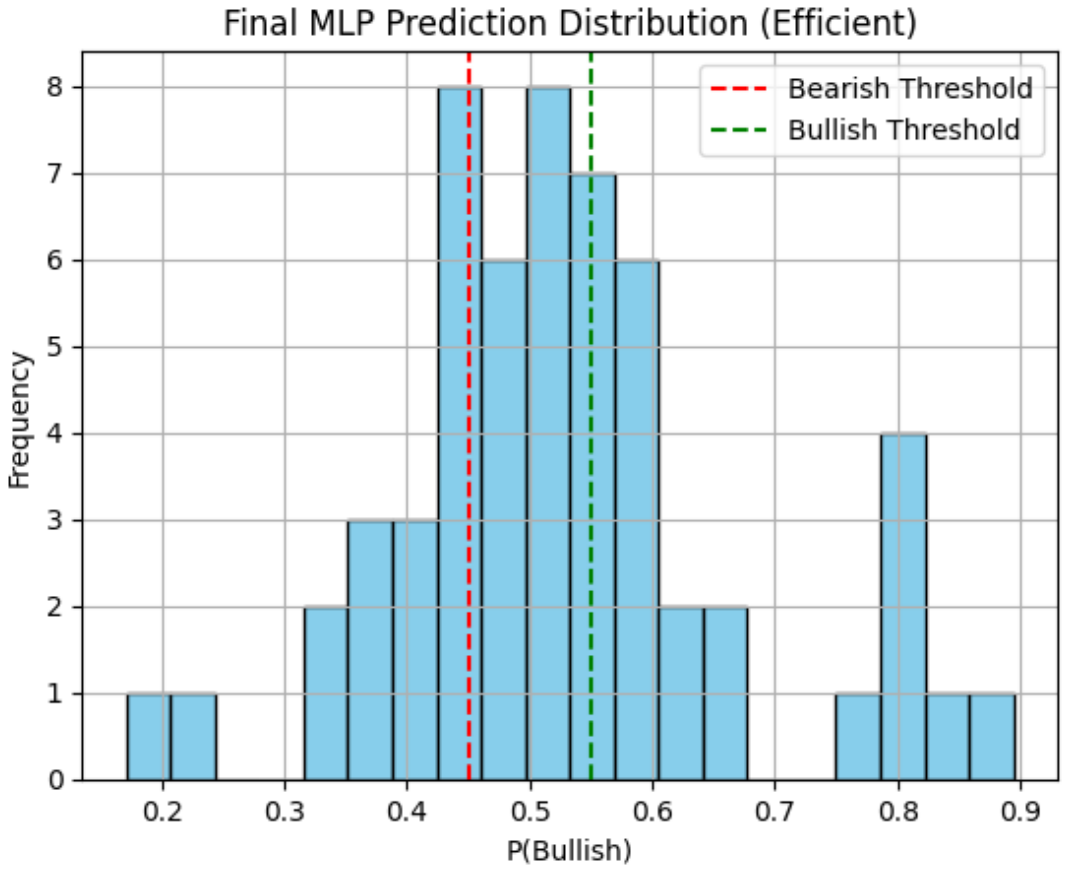

Figure 16: Final MLP Prediction Distribution (Efficient)

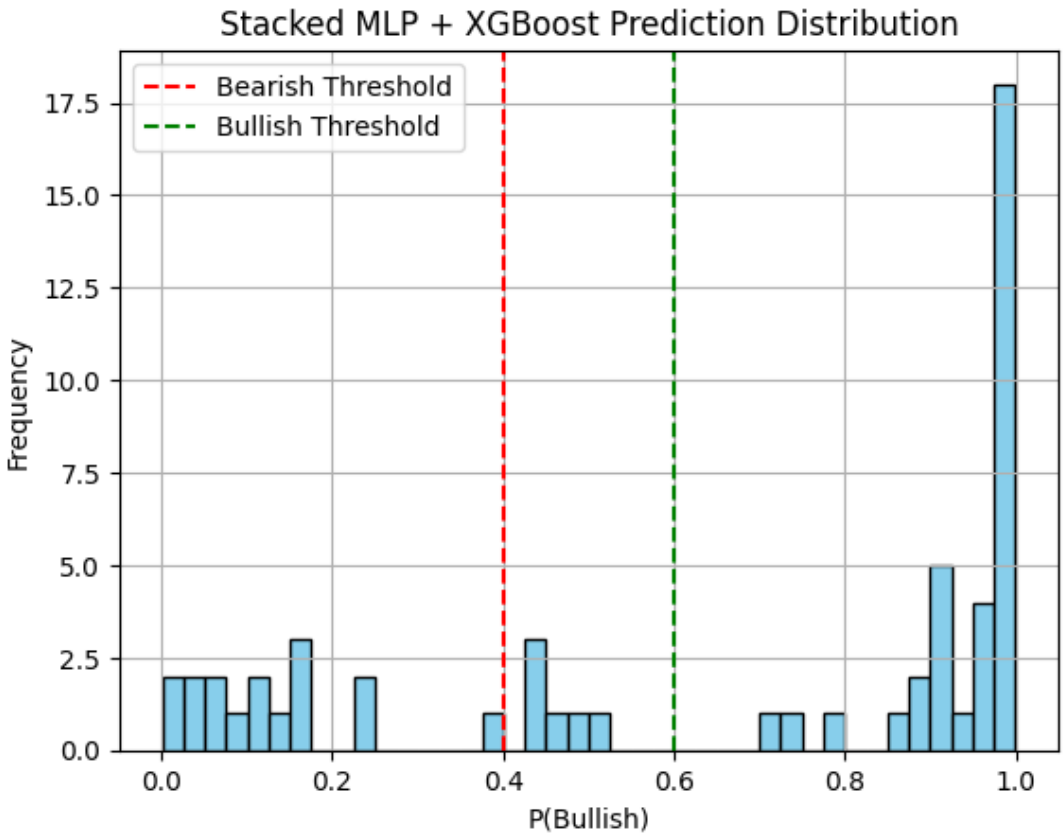

Figure 17: Stacked MLP + XGBoost Prediction Distribution (risk averse thresholds)

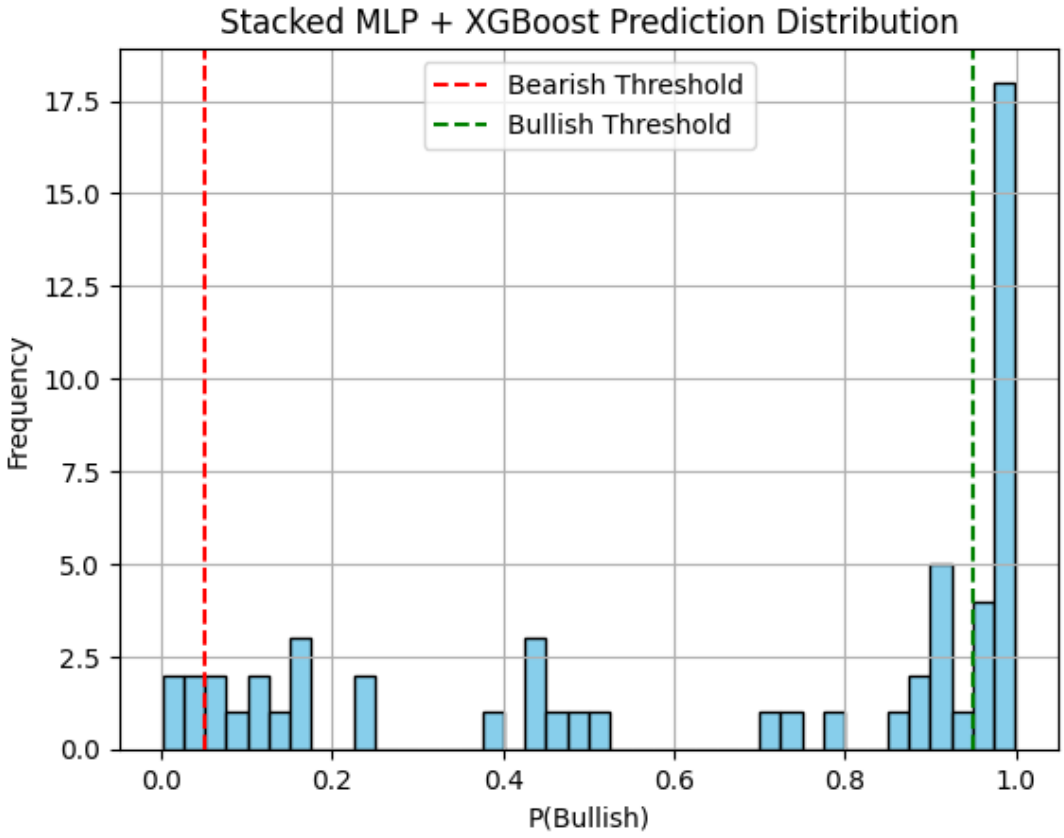

Figure 18: Stacked MLP + XGBoost Prediction Distribution (risk seeking thresholds)

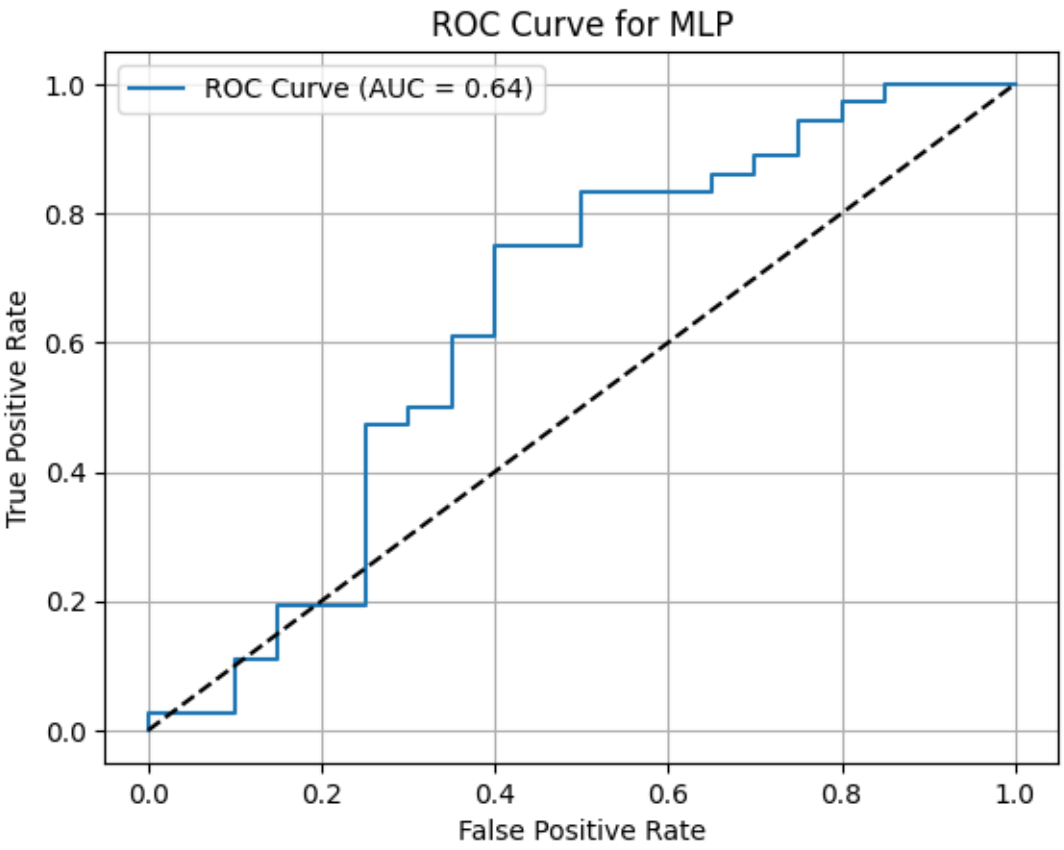

Figure 19: ROC Curve for MLP

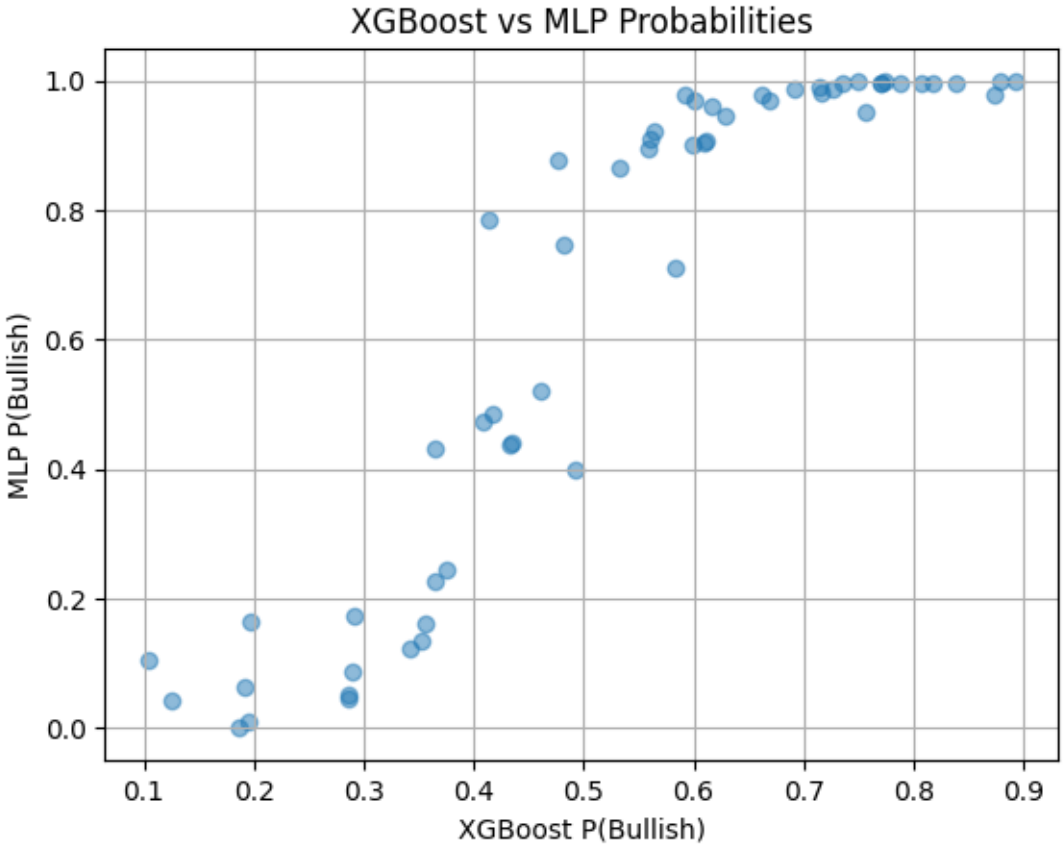

Figure 20: XGBoost vs MLP Probabilities

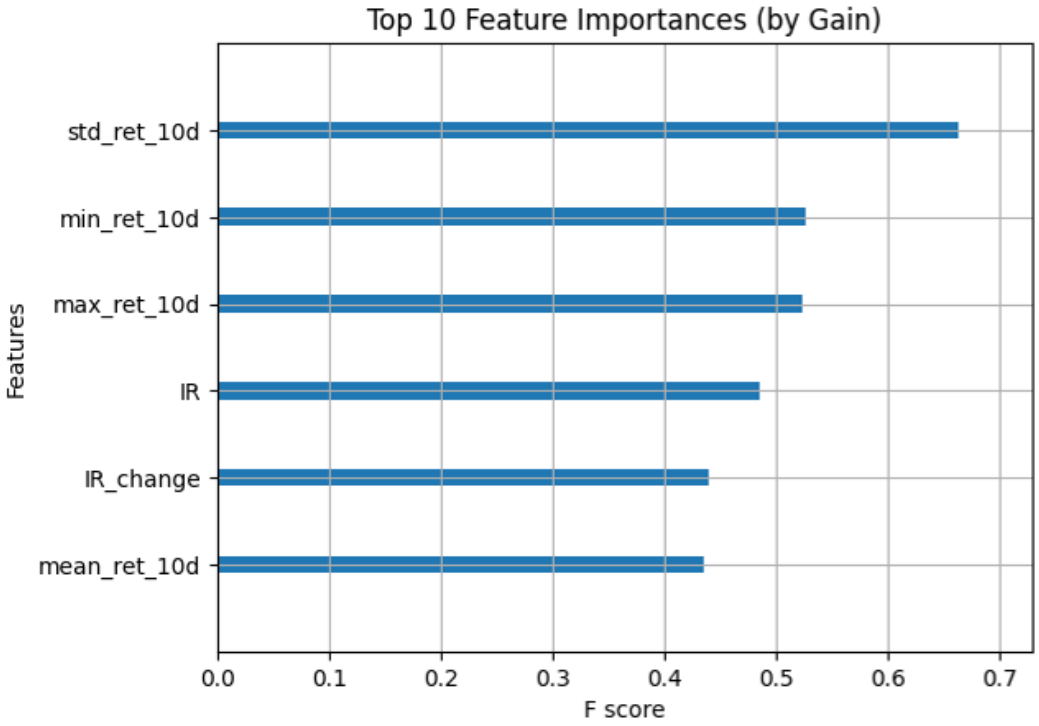

Figure 21: Top 10 Feature Importances (by Gain)

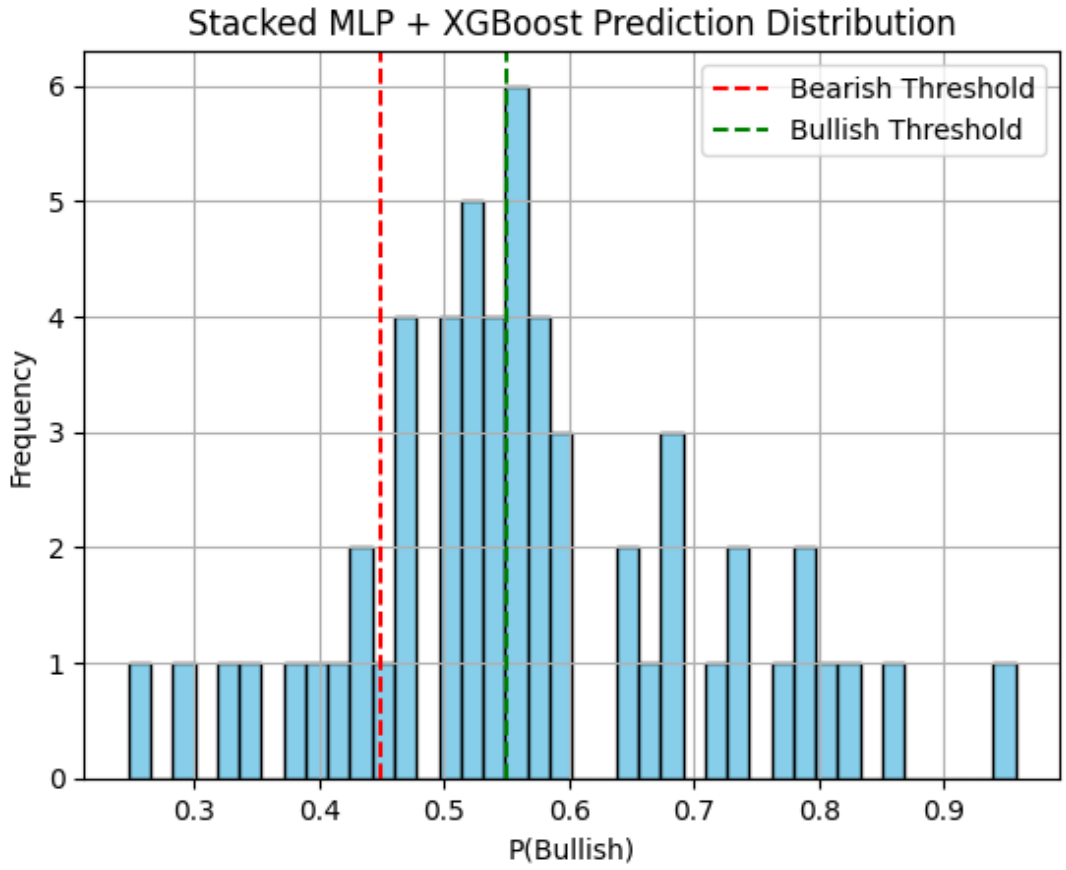

Figure 22: Stacked MLP + XGBoost Prediction Distribution

References

[1] Center for Research in Security Prices (CRSP).
Daily Stock File (daily_crsp.csv).

[2] Board of Governors of the Federal Reserve System.
Federal Reserve Statistical Release H.15: Selected Interest Rates (daily data).
Data Download Program (DDP), last updated June 10, 2025.
Available at:
`https://www.federalreserve.gov/econres/feds/open-source-cross-sectional-asset-pricing.htm`