Task 2 Report: Predictive Modeling and Performance Evaluation

Objective:

To build binary classification models predicting whether Apple stock's closing price will increase the next day, using engineered technical indicators as features.

Methodology:

- Created a **binary target variable**:
    - `1` if next day's closing price is higher than current day's, else `0`.
- Selected **10 features** from Task 1 technical indicators and returns:
    - SMA_20, EMA_20, RSI, MACD variants, ATR, OBV, Daily_Return, Log_Return.
- Split data into **train (80%) and test (20%)** sets, preserving time order (no shuffling).
- Trained two tree-based classifiers:
    - **Random Forest** (100 trees, max depth 5)
    - **XGBoost** (100 trees, learning rate 0.05, max depth 5)
- Evaluated using metrics:
  Accuracy, Precision, Recall, F1 Score, ROC AUC, Confusion Matrix, ROC Curve.

Results Summary:

Metric	Random Forest	XGBoost
Accuracy	48.97%	48.28%
Precision	72.73%	58.33%
Recall	10.13%	17.72%
F1 Score	17.78%	27.18%
ROC AUC	0.6020	0.4665
CV Accuracy	51.24%	50.40%

- **Random Forest** achieves higher precision but very low recall, indicating it is conservative in predicting upward movement, missing many positive cases.
- **XGBoost** shows slightly better recall and F1 score but lower precision and ROC AUC.
- Both models have **accuracy close to 50%**, indicating near-random performance.

Interpretation:

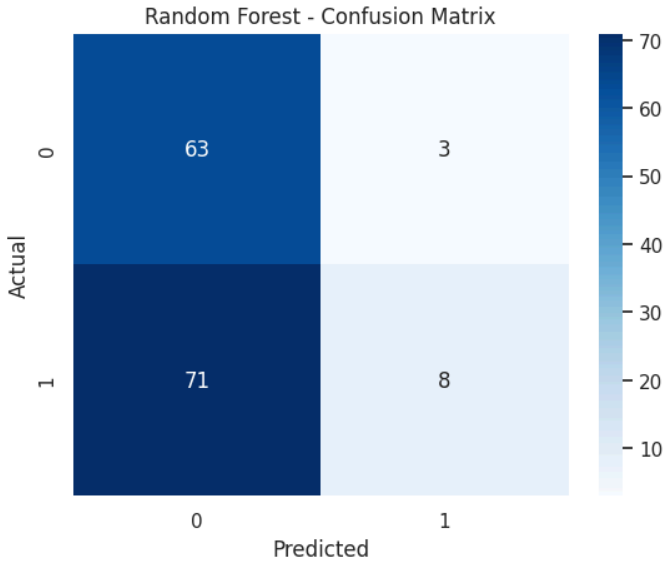
- The predictive performance is weak, suggesting the features and model settings are insufficient for reliable next-day price movement prediction.
- Imbalanced precision and recall highlight a trade-off:

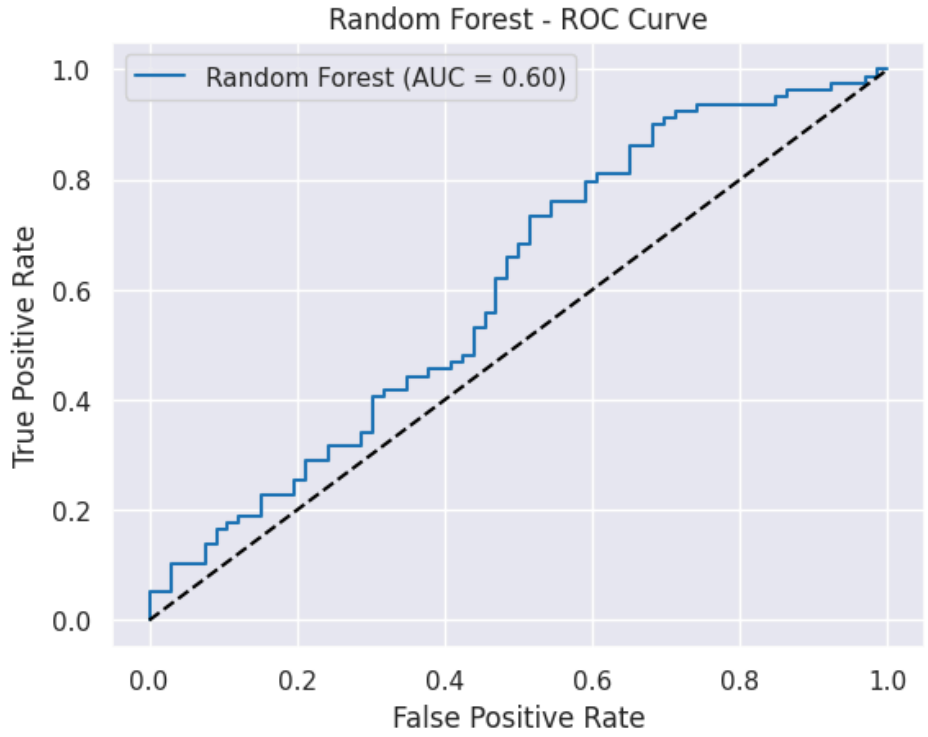
- RF favors fewer false positives (high precision) but misses many true positives (low recall).
- XGB captures more positives but at the cost of false positives.

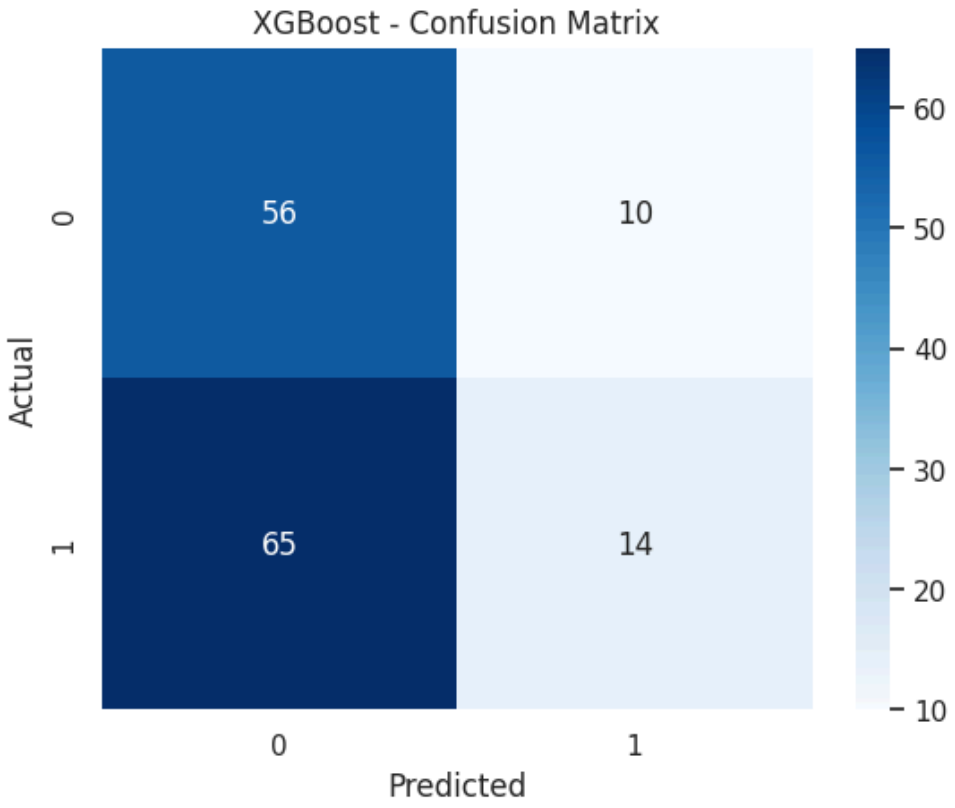
Recommendations for Improvement:

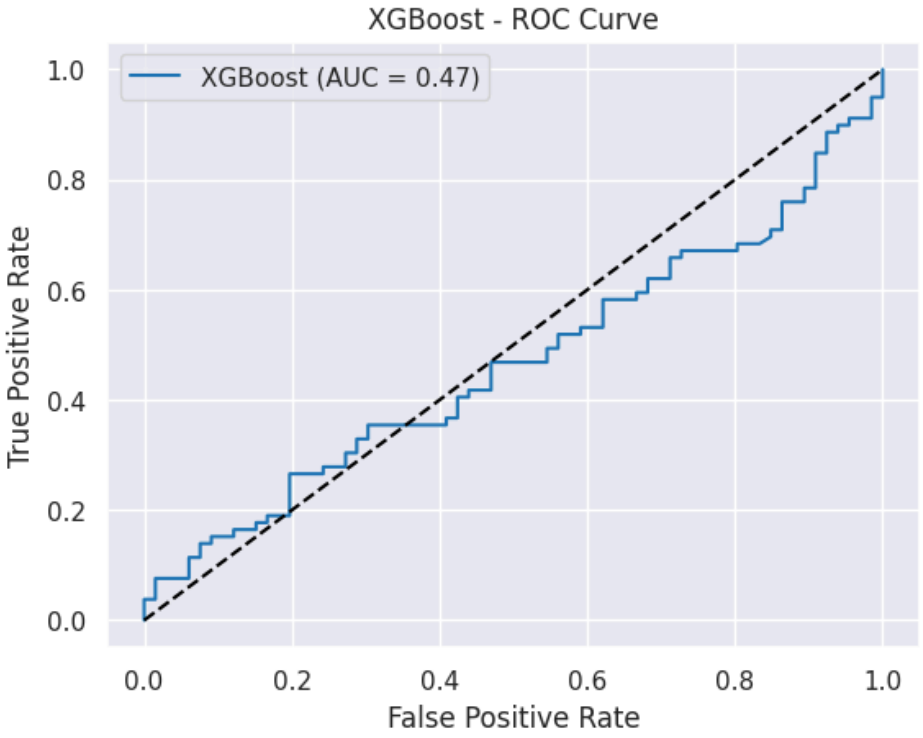
1. **Feature Engineering:**
   - Incorporate additional features, such as more lagged indicators, volume spikes, or fundamental data.
   - Use feature importance analysis to refine feature set.
2. **Model Complexity and Hyperparameter Tuning:**
   - Perform extensive hyperparameter tuning (e.g., grid search or Bayesian optimization).
   - Experiment with deeper trees, more estimators, or alternative boosting parameters.
3. **Address Class Imbalance:**
   - Use oversampling (SMOTE) or class-weight adjustments to improve recall.
4. **Temporal Validation:**
   - Implement walk-forward cross-validation to better simulate real trading environment.
5. **Alternative Models:**
   - Explore advanced models like Recurrent Neural Networks (LSTM) to capture temporal dependencies.
6. **Predicting Returns Instead of Direction:**
   - Consider regression approaches predicting return magnitude, not just direction.

Visual Results:






XGBoost - Confusion Matrix
56
10
65
14
Actual
0
1
Predicted
0
1
60
50
40
30
20
10


XGBoost - ROC Curve
XGBoost (AUC = 0.47)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0