Machine Learning Classifiers for Financial Trading Strategies

Rohan Niranjan Kalpavruksha, Paul Rohit Jangareddi, Poonam Jadhav, Sri Harshitha Gummadi

Pace University, NY

ABSTRACT

This project aims to implement and evaluate machine learning classifiers to predict stock trading signals based on historical financial data. Two trading strategies, "Next Day Price Comparison" and "Moving Average Crossover," are developed and tested using models such as K-Nearest Neighbors (KNN), Random Forest, and Gradient Boosting. The study emphasizes the performance of different classifiers and provides insights into the potential of machine learning in financial trading.

INTRODUCTION

- **Background**: Financial trading strategies have historically relied on technical and fundamental analysis. With advancements in machine learning, there is a growing opportunity to harness algorithms for predicting stock price movements.
- **Objective**: This project explores how ML classifiers can enhance trading strategies by analyzing historical data and generating buy/sell signals.

PROBLEM DEFINITION

The goal is to predict trading signals based on historical stock price data. We examine two strategies:

1. Predicting the next day's stock price movement.
2. Leveraging moving averages to identify trends.
   Challenges include handling noisy data, balancing model complexity, and mitigating overfitting risks.

METHODOLOGY

- **Data Collection**: Historical stock prices from 2015 to 2023 were downloaded using the Yahoo Finance (yfinance) Python library.
- **Data Preprocessing**:
  - Handling missing values and outliers.
  - Calculating features like 20-day and 50-day moving averages.
  - Defining target variables for trading signals.
- **Exploratory Data Analysis (EDA)**:
  - Visualized closing prices, trading volumes, and return distributions.
  - Analyzed moving averages for trend identification.

DATA ANALYSIS

- **Closing Price Over Time**: Showed high volatility with recoveries indicating market trends.
- **Volume Traded**: Spikes in trading volume were correlated with market events.
- **Moving Averages**: Crossovers between 20-day and 50-day moving averages indicated trend shifts.
- **Daily Returns**: Displayed a roughly normal distribution centered around zero.

TRADING STRATEGIES

- **Strategy 1: Next Day Price Comparison**
  - Logic: Generate a buy signal (+1) if tomorrow's close > today's close; otherwise, sell (-1).
  - Implementation: y = np.where(df['Stock_Price'].shift(-1) > df['Stock_Price'], 1, -1)
- **Strategy 2: Moving Average Crossover**
  - Logic: Use the golden cross (50-day MA > 200-day MA) as a buy signal and death cross (50-day MA < 200-day MA) as a sell signal.
  - Implementation: Calculated moving averages and detected crossovers.

MODEL DEVELOPMENT AND EVALUATION

- **Models Used**:
  - K-Nearest Neighbors (KNN): Simple and interpretable but sensitive to noise.
  - Random Forest (RF): An ensemble method providing robust predictions.
  - Gradient Boosting (GB): Captures complex patterns with sequential learning.
- **Process**:
  - Split data (80% training, 20% testing).
  - Train models using scikit-learn.
  - Evaluate performance using metrics such as accuracy, precision, recall, and F1-score.

RESULTS

- **K-Nearest Neighbors (KNN)**:
  - **Strategy 1 Accuracy**: 40%
  - **Strategy 2 Accuracy**: 66.67%
  - **Classification Report**:
    - **-1**:
      - Precision: 61%
      - Recall: 69%
      - F1-score: 65%
      - Support: 32
    - **1**:
      - Precision: 60%
      - Recall: 52%

                - F1-score: 56%
                - Support: 29
    - **Random Forest**:
        - **Strategy 1 Accuracy**: 73.34%
        - **Strategy 2 Accuracy**: 66.67%
        - **Classification Report**:
            - **-1**:
                - Precision: 52.63%
                - Recall: 38.46%
                - F1-score: 44.44%
                - Support: 26
            - **1**:
                - Precision: 54.29%
                - Recall: 67.86%
                - F1-score: 60.32%
                - Support: 28
    - **Gradient Boosting**:
        - **Strategy 1 Accuracy**: 53.73%
        - **Strategy 2 Accuracy**: 78.68%
        - **Classification Report**:
            - **Decrease**:
                - Precision: 79%
                - Recall: 81%
                - F1-score: 80%
                - Support: 32
            - **Increase**:
                - Precision: 79%
                - Recall: 76%
                - F1-score: 77%
                - Support: 29
- **Best Performing Models**:
    - Strategy 1: Random Forest provided the highest accuracy.
    - Strategy 2: Gradient Boosting was the most effective.
- **Insights**:
    - Ensemble models outperformed KNN due to their ability to manage noise and complexity.
    - Different strategies benefited from different models, highlighting the importance of model selection.

CHALLENGES

- Missing data and computational inefficiencies with large datasets.

- Potential overfitting with moving averages and limited features.
- Balancing sensitivity and specificity in trading signal predictions.

INSIGHTS AND FINDINGS

- **Insights**:
  - Gradient Boosting captures complex patterns in financial data.
  - Random Forest's robustness is suited for short-term price predictions.
- **Recommendations**:
  - Perform hyperparameter tuning for better model performance.
  - Explore time-based cross-validation to better mimic real-world scenarios.
  - Incorporate additional features like sector performance and macroeconomic indicators.

CONCLUSION AND FUTURE WORK

- **Summary**: The project demonstrated the viability of ML classifiers for trading strategies. Gradient Boosting emerged as the most effective for trend-based strategies, while Random Forest excelled in short-term predictions.
- **Future Work**:
  - Implement backtesting to simulate trading outcomes.
  - Explore deep learning models like LSTMs for sequential data.
  - Incorporate dynamic feature selection and real-time data streaming.

REFERENCES

- Python libraries: yfinance, scikit-learn, NumPy, Pandas.
- Machine Learning for Asset Managers by Marcos López de Prado.
- Advances in Financial Machine Learning by Marcos López de Prado.