

AI for Market Trend Analysis

Presented by Manish Kumar Singh (iitrprai_24102019)

Project Type: Individual Open Project (Module E)

Introduction

Financial markets show complex and non-linear patterns. Traders and analysts often use technical indicators to understand whether the market is in a bullish or bearish regime. This project builds an AI/ML pipeline that learns these patterns from historical data and predicts the market trend regime 5 trading days ahead.

Problem Statement

Manual market trend analysis is time consuming and highly subjective. Traditional rule-based indicators may fail in noisy market periods. The goal is to build a machine learning classifier that predicts whether the market will be Bullish (1) or Bearish (0) 5 days in the future using engineered features from OHLCV (Open, High, Low, Close, Volume) time-series data.

Methodology

- Load daily OHLCV dataset (S&P 500 sample data).
- Perform feature engineering using technical indicators.
- Create target trend label (Bullish/Bearish) shifted by 5 days.
- Train Random Forest classifier using time-series split (avoids leakage).
- Evaluate with Accuracy, F1-score, ROC-AUC and confusion matrix.
- Save trained model as `rf_trend_model.joblib` for deployment.
- Deploy predictions via Streamlit UI (upload CSV, predict, download output).

Tools & Technologies

- Language: Python
- Libraries: NumPy, Pandas, Scikit-learn, Joblib, Matplotlib
- Model: Random Forest Classifier
- Validation: TimeSeriesSplit Cross-Validation
- UI: Streamlit Web App (`app.py`)
- Development: Jupyter Notebook / VS Code

Results & Conclusion

The trained model achieved strong performance on an unseen test period (chronological split). High performance is partly due to persistence of moving-average crossover regimes which form the target label.

Metric	Value
Accuracy (Test)	0.9666
ROC-AUC (Test)	0.9957
F1-score (Test)	0.9778
Precision (Test)	0.9700
Recall (Test)	0.9857

In conclusion, this project demonstrates an end-to-end AI application for market trend classification: data preprocessing → feature engineering → model training → evaluation → deployment. The Streamlit UI enables easy interaction and prediction export, making the solution suitable for academic demonstration.