StockSense Pro – AI-Powered Stock Trend Prediction Dashboard

Project Overview

StockSense Pro is an end-to-end Machine Learning project that predicts short-term stock price direction (UP/DOWN) using historical stock data and technical indicators. The system is deployed as an interactive web application using Streamlit. The objective is to demonstrate applied machine learning in the financial domain as a decision-support tool.

Dataset

The dataset is fetched from Yahoo Finance using the yfinance Python library. It contains historical stock data including Open, High, Low, Close prices and trading Volume. The dataset spans multiple years, providing sufficient data for training and evaluation.

Data Preprocessing

Missing values created due to rolling indicators are removed. The data is cleaned and transformed into meaningful features. Tree-based models like Random Forest do not require feature scaling, which simplifies preprocessing.

Feature Engineering

Technical indicators are derived from raw price data. These include Moving Averages (MA10, MA20, MA50), Volatility (10-day, 20-day), RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), and momentum features based on past returns. Feature engineering helps the model capture trends, momentum, and market behavior.

Target Variable

The task is framed as a binary classification problem. The target variable indicates whether the next day’s closing price will increase or decrease. A small threshold is applied to reduce noise caused by very small price fluctuations.

Machine Learning Algorithm

The Random Forest Classifier is used for prediction. Random Forest is an ensemble learning method that combines multiple decision trees to improve robustness and generalization. It handles non-linear relationships well and performs effectively with engineered features.

Class Imbalance Handling

Stock market data often has slightly more upward movements than downward ones. To prevent bias toward the majority class, class weighting is applied during model training so that both classes are treated equally.

Model Evaluation

The model is evaluated using Accuracy, Balanced Accuracy, Precision, Recall, and F1-score. Balanced Accuracy is emphasized due to class imbalance. Typical results are slightly above random guessing, which is expected for short-term stock direction prediction due to market noise and external influences.

Web Application (Streamlit)

The trained model is deployed using Streamlit. The web app allows users to input stock tickers, visualize candlestick charts, view technical indicators, receive Buy/Sell/Hold signals, perform backtesting, and export predictions as CSV files.

Trading Signal Logic

Predictions are converted into Buy, Sell, or Hold signals based on confidence thresholds. This reduces overconfident actions when the model is uncertain and improves the practical usability of the system.

Backtesting

Backtesting simulates how the model’s predictions would have performed on historical data. This helps demonstrate the potential usefulness of the model in a trading-like scenario. The backtesting implemented is simplified and intended for educational purposes only.

Conclusion

This project demonstrates the complete lifecycle of a machine learning system: data collection, preprocessing, feature engineering, model training, evaluation, and deployment. While stock market prediction is inherently uncertain, the project effectively showcases the application of ML in real-world financial data analysis.

Future Scope

Future enhancements may include integrating news sentiment analysis, deep learning models for time series forecasting, additional macroeconomic indicators, and cloud deployment for scalability.