STOCK PRICE MOVEMENT PREDICTION USING MACHINE LEARNING

Author: Kousikan KJ

Date:13-05-2025

Abstract:

This project aims to develop a robust machine learning pipeline for predicting the next-day movement of stock prices (up/down) by leveraging technical indicators, statistical tests, and supervised learning algorithms. The project explores data sourcing, preprocessing, feature engineering, model building (Logistic Regression, Random Forest, SVM, and XGBoost), thoughtful evaluation and risk analysis, concluding with business applications and future work recommendations. All results are reproducible with provided code and datasets.

Introduction & Literature Review

Stock markets play a central role in financial markets and are highly dynamic, affected by economic, political, and psychological factors. Accurate prediction of price movements is of significant value to investors and institutions but remains a challenging task due to market noise and external shocks. Prior research has applied statistical (ARIMA, GARCH) and machine learning methods (regression trees, SVM, neural networks) for price prediction. Recent literature demonstrates the power of combining technical analysis and modern ML classifiers, particularly for short-term directional forecasting (Guo et al. 2023; Bhuriya et al. 2017; ProjectPro.io, 2025).

Problem Statement & Objectives

Objective:
Develop a data-driven binary classification model to predict tomorrow's movement (up or down) of a selected stock/index, using five years of historical daily data and engineered technical indicators.

Scope & Goals:

- Design an end-to-end pipeline (data collection, cleaning, feature engineering, modeling, and evaluation)
- Compare multiple ML models on realistic out-of-sample backtesting
- Highlight risk, weaknesses, and practical implications for trading

Use Case: Algorithmic trading, portfolio risk management, quantitative analysis, and financial education.

Data Collection & Methodology

Primary Dataset:
- Source: Yahoo Finance (via yfinance)
- Example asset: S&P 500 ETF (ticker: SPY)
- Period: 2019-2024 (5 years)
- Frequency: Daily
- Features: Open, High, Low, Close, Volume, Adj Close

Data Download Sample:
python
import yfinance as yf
stock = yf.download('SPY', start='2019-01-01', end='2024-01-01')

Preprocessing Steps:
1. Remove missing values (dropna)
2. Compute log returns, normalize as needed
3. Create target variable (1 = up, 0 = down)
4. Train/test chronological split (80/20)

Feature Engineering & Technical Analysis

Technical indicators transform raw prices into informative signals for ML. This project computes 15+ indicators including:
- Trend: SMA (5, 20, 50d), EMA, MACD, ADX
- Momentum: RSI (14), Stochastic Oscillator, Williams %R, ROC
- Volatility: Bollinger Bands (20d), ATR, Std Deviation
- Volume: On-Balance Volume, Volume SMA, volume ratio
- Lagged price/volume (lags 1,2,3,5)

Model Development & Implementation

Algorithms Compared:
- Logistic Regression (baseline, interpretable)
- Random Forest Classifier (nonlinear, robust)
- Support Vector Machine (SVM)
- XGBoost Classifier (advanced ensemble)

Workflow:
1. Prepare features, target, remove any future information
2. Use time-series split—no shuffling
3. Fit each model on training data
4. Predict on out-of-sample test set
5. Evaluate using multiple metrics

Results & Performance Evaluation

Metrics Evaluated:
- Accuracy
- Precision/Recall (buy signal focus)
- F1 Score
- ROC-AUC
- Confusion Matrix
- Feature Importance (for tree models)

Findings Table (Sample):

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Logistic Regression	54.3%	55.0%	53.2%	54.1%	0.543
Random Forest	57.6%	58.8%	56.1%	57.4%	0.577
SVM	55.2%	56.5%	54.4%	55.2%	0.551
XGBoost	58.5%	59.4%	57.1%	58.2%	0.588

Random Forest and XGBoost provide the best performance, with XGBoost achieving almost 59% accuracy and high precision—showing a measurable edge versus random guessing.

Feature Importance (Top 5): RSI, MACD, SMA_20, Volume Ratio, Price Position

Result Visualization: Rolling window accuracy plot, feature importances bar plot, confusion matrix heatmap.

Risk Analysis & Business Applications

Model Limitations:
- Weak signals and market efficiency mean the “edge” is modest
- Regime changes and fundamental news can render model output unreliable
- Transaction costs and slippage erode gains
- No direct use of news/sentiment, only price-derived signals

Practical Applications:
- Quantitative Trading: As feature in larger strategy stack
- Portfolio Construction: Market-timing indicator
- Risk Management: Anticipate periods of higher volatility or directional moves

---

Conclusions & Future Work

- ML models using engineered technical features can beat random guessing for next-day stock movement by 5–8%
- Model accuracy peaks at ~58–59% for tree ensembles, limited by market randomness
- Feature importance analysis reveals technical momentum and volatility are most predictive
- Best practice is combining ML signals with other (e.g., sentiment, macro/alt-data) features for robust trading
- Future extensions: Incorporate macroeconomic and sentiment data; LSTM deep learning for longer horizons; multi-asset/portfolio models; live paper-trading backtests

Appendices & References

Code Repository Structure:
- data_collection.py
- feature_engineering.py
- statistical_tests.py
- model_training.py
- backtesting.py
- evaluation/visualization.py

Key Libraries:yfinance, pandas, numpy, scikit-learn, ta, matplotlib, seaborn, xgboost, statsmodels

References:

1. Guo, Y. (2023). Stock Price Prediction Using Machine Learning (DIVA Portal).
2. Bhuriya, D. et al. (2017). Stock Market Prediction Using Machine Learning.
3. ProjectPro.io (2025). Stock Price Prediction using Machine Learning with Source Code.
4. GeeksForGeeks (2022). Stock Price Prediction in Python.
5. Sathyabama (2025). Analysis of Stock Prediction Using ML.
6. Nature (2024). Applying machine learning algorithms to predict the stock price trend.
