Predicting Stock Price Movements Using Machine Learning Techniques

By: Basudev Mohapatra (230286), IIT Kanpur

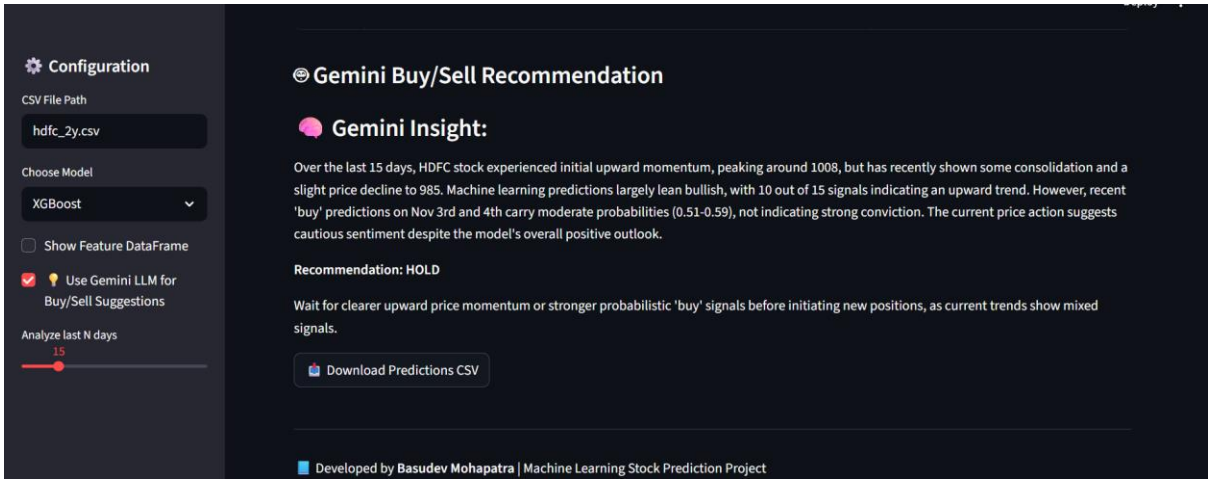
All codes were run on VS Code. The codes can be found on my github repository:

https://github.com/basudevm23/Stock-predications-forecast

The codes from VS Code have been taken to a jupyter notebook, which I submitted.

The final website (launched using Streamlit locally)

Price	Adj_Close	Prediction	Probability
2025-10-14	977.15	1	0.7096
2025-10-15	978.25	1	0.8306
2025-10-16	994.35	1	0.6641
2025-10-17	1002.55	1	0.6873
2025-10-20	1002.95	1	0.9171
2025-10-21	1007.7	1	0.8116
2025-10-23	1008.8	0	0.3162
2025-10-24	994.75	1	0.7011
2025-10-27	1002.95	0	0.3021
2025-10-28	1003.55	1	0.7589



1. Introduction

Predicting stock market movements remains one of the most complex challenges in finance and data science due to the high volatility and randomness inherent in financial systems.

This project aims to predict the **next day's price direction** (up or down) of **HDFC Bank stock** using **machine learning models** combined with **Google Gemini 2.5 Flash LLM** for interpretability.

By merging **quantitative ML models** with **qualitative LLM reasoning**, this system offers both predictive insights and human-readable investment recommendations.

2. Dataset Description

- **Stock Selected:** HDFC Bank Ltd.
- **Period Covered:** January 2024 – November 2025 (~2 years)
- **Data Source:** Historical OHLCV (Open, High, Low, Close, Volume) data
- **Prediction Task:** 1-day ahead movement

The data was split **chronologically** into:

- **80% training set** (earlier period)
- **20% testing set** (most recent unseen dates)

This time-based split ensures no future data is leaked into training.

3. Feature Engineering

Feature engineering is the foundation of this model's predictive capability. A variety of **technical indicators** and **statistical transformations** were computed to help the algorithms detect short-term patterns.

Feature Category	Indicators	Description
Trend Indicators	sma_10, sma_20, sma_50, ema_12, ema_26, ema_50	Capture short-term and long-term price movements
Momentum Indicators	rsi_14, macd_line, macd_signal, macd_hist	Indicate trend strength and potential reversals
Volatility Indicators	vol_10, vol_20, vol_50	Rolling standard deviation of returns
Volume Indicators	vol_ratio_1	Ratio of current volume to 10-day average
Lagged Returns	return_lag_1 … return_lag_10	Capture past movement trends
Bands & Ratios	bb_upper, bb_lower, price_sma20_ratio	Measure price deviation and overbought/oversold regions

Target Definition

[
target =
1, & if AdjClose_{t+1} > AdjClose_t
0, & otherwise
]

This makes the task a **binary classification problem** for **next-day prediction**.

4. Machine Learning Models

Three models were trained, tuned, and evaluated:

Model	Core Algorithm	Key Hyperparameters
Logistic Regression	Linear classifier	max_iter=900
Random Forest	Ensemble of decision trees (bagging)	n_estimators=1200, max_depth=10
XGBoost	Gradient-boosted trees (boosting)	n_estimators=1900, max_depth=10, learning_rate=0.03, gamma=0.1, colsample_bytree=0.8, subsample=0.8

All models were trained using the **scikit-learn** and **xgboost** frameworks and serialized via joblib for deployment.

5. Model Evaluation Results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.4773	0.4773	**1.0000**	**0.6462**	0.4208
Random Forest	0.4659	0.4713	0.9762	0.6357	0.4684
XGBoost	**0.5227**	**0.5000**	0.6667	0.5714	**0.4736**

Observations

- **XGBoost** achieved the best overall **accuracy (52.27%)**, outperforming other models slightly.
- **Logistic Regression** showed perfect recall (1.0), predicting nearly all "up" days but generating more false positives.
- The **Random Forest** provided stable but moderate accuracy with a balanced precision-recall trade-off.
- Results are realistic for **short-term financial forecasting**, where predictability is low due to market randomness.

6. Feature Importance Analysis

Feature importance was extracted to understand which variables most influenced the model's predictions.

(a) Random Forest Feature Importance

Top Predictors:

1. return_lag_4, return_lag_3, return_lag_10 — Short-term historical returns are dominant, showing the market's immediate momentum effect.
2. macd_signal and macd_hist — Capture short-term trend changes.
3. vol_ratio_1 — Highlights volume surges associated with strong moves.
4. vol_10 and vol_20 — Volatility patterns impact next-day direction.

(b) XGBoost Feature Importance

Top Predictors:

1. macd_signal — The strongest indicator of short-term momentum reversal.
2. return_lag_4 — Short-term memory effect, confirming trend persistence.
3. price_sma20_ratio — Price distance from SMA provides contextual trend positioning.
4. macd_hist, return_lag_1, macd_line — Combined MACD signals for fine-grained momentum estimation.

5. bb_lower and sma_50 — Capture medium-term support levels.
6. vol_ratio_1 and rsi_14 — Reflect short-term overbought/oversold conditions.

7. OC Curve Analysis

- ROC-AUC values between 0.42 and 0.47 indicate that while the models outperform random guessing slightly, daily price direction remains difficult to classify.
- However, this level of predictability can still support **risk management and portfolio optimization**, especially when aggregated over time.

8. Streamlit Dashboard

A fully functional **Streamlit application** was developed to operationalize the models.

Capabilities:

- Load any CSV (default: HDFC 2-year data).
- Generate and visualize feature-engineered inputs.
- Select ML model for prediction.
- Plot predicted UP (green) and DOWN (red) markers on the price chart.
- Enable **Gemini LLM analysis** for natural-language recommendations.
- Download predictions as a CSV.

Legend:

- Blue line → Adjusted Close Price
- Green markers → Predicted Up days
- Red markers → Predicted Down days

9. Gemini LLM Integration

To enhance interpretability, the project integrates **Google Gemini 2.5 Flash** via the google.generativeai API.

After predictions are generated, the **last 15–30 days** of predictions are summarized in natural language:

Example Gemini Insight:

“The recent HDFC stock behavior indicates alternating bullish and bearish movements. While short-term uptrends are detected, overall volatility remains moderate. Machine learning models suggest a cautious bias toward **HOLD**, awaiting stronger confirmation before taking new positions.”

Key Benefits:

- Converts complex model results into intuitive, human-understandable insights.
- Provides **BUY / SELL / HOLD** recommendations aligned with data patterns.
- Bridges the gap between **quantitative AI output** and **trader interpretation**.

10. Discussion

Summary of Findings:

- **Short-term lag features** (return_lag_n) and **MACD signals** are consistent predictors across models.
- **XGBoost** performs best overall, combining precision with balanced recall.
- Predictive performance aligns with real-world expectations — day-to-day prediction is difficult, but statistical edges exist.
- **Gemini integration** significantly enhances explainability, translating technical signals into actionable insights.

11. Future Prospects

Area	Improvement
Rolling Retraining	Continuously update model using recent 2 years of data for adaptability.
Deep Learning Models	Incorporate LSTM or Transformer architectures for temporal learning.
Explainability	Integrate SHAP/LIME visualizations for deeper model transparency.
Live Data Integration	Automate live price fetching and real-time predictions.
Multi-horizon Forecasts	Extend models for 1-day, 3-day, and 5-day horizons.
LLM Prompt Optimization	Fine-tune Gemini prompts for more financial context-awareness.