IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

NIFTY 50 Weekly Intelligence Platform: A MLOps Research Study for Local Financial Forecasting, Monitoring, and Reporting

Jayesh Patil[1], Shubham Pote[2], Safiya Khan[3], Mit Shah[4], Prof. Amrapali Chavhan[5]
[1234]Students, [5] Project Guide, Department of Artificial Intelligence and Data Science
AISSMS Institute of Information Technology, Pune, India

Abstract: *This paper presents the NIFTY 50 Weekly Intelligence Platform, a local MLOps framework designed to demonstrate how a financial prediction system can be engineered as a complete lifecycle workflow rather than as a model-only prototype. The platform integrates market data ingestion, validation, feature engineering, drift-aware training decisions, model artifact management, API serving, dashboard visualization, and weekly reporting for stocks from the NIFTY 50 universe. Daily OHLCV data is acquired using yfinance, stored as parquet snapshots, transformed into technical-indicator features, and used to train a LightGBM classifier for 5-trading-day direction prediction. Outputs are served through FastAPI and consumed by a React dashboard, while model health and drift are exposed through Prometheus-compatible metrics and alert rules. The latest repository artifacts show that the processed dataset contains 35,289 rows, 42 columns, and 49 successfully covered symbols across the period 16 May 2023 to 17 April 2026. The latest saved classifier achieved an accuracy of 42.79%, precision of 40.14%, recall of 70.46%, F1-score of 51.15%, and ROC-AUC of 0.4528 on the held-out test set. Although these predictive results are modest and not yet investment-grade, the project succeeds as a reproducible end-to-end MLOps study that demonstrates data engineering discipline, traceable artifacts, drift monitoring, report generation, and frontend-backed explainability of system state. The work therefore contributes a realistic academic example of financial machine learning infrastructure, while also honestly documenting the limitations of baseline forecasting performance in noisy, non-stationary market environments.*

Keywords: MLOps, NIFTY 50, financial forecasting, LightGBM, drift detection, FastAPI, React dashboard, technical indicators, model monitoring.

I. INTRODUCTION

Forecasting stock-market movement remains one of the most difficult problems in applied machine learning. The difficulty comes not only from noisy price series, changing market regimes, and behavioral reactions, but also from the gap between research prototypes and deployable systems. Many student projects stop at a trained model and a few numerical metrics; however, a professional machine learning solution must also define where data comes from, how it is validated, how features remain stable across runs, how artifacts are versioned, when retraining is needed, and how model behavior is communicated to users. In practice, these engineering concerns often determine whether a machine learning solution is genuinely useful.

The present work addresses that gap by designing and implementing a local MLOps platform for the NIFTY 50 universe. The project goal is not to claim state-of-the-art market predictability. Instead, it is to build a coherent workflow that demonstrates the complete lifecycle of a machine learning system operating on Indian equity data. The platform includes five core capabilities: reliable market data ingestion, repeatable feature engineering, baseline model training, operational health visibility, and user-facing reporting.

DOI: 10.48175/IJARSCT-34489

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 19, April 2026 Impact Factor: 8.2

These components are intentionally integrated into one local workflow that can run on a laptop-class machine without cloud infrastructure. This design decision is important in academic contexts. First, it allows reproducibility during project evaluation and demonstration. Second, it surfaces the real trade-off between model complexity and operational simplicity. Third, it reflects how many practical ML systems begin: with one interpretable baseline, stable artifact storage, and disciplined monitoring, before any sophisticated ensemble or deep sequence model is introduced. For finance-related applications, this is especially relevant because strong historical fit can easily mask weak live performance.

The major contributions of this work are summarized below:

- A local end-to-end MLOps pipeline tailored to the NIFTY 50 stock universe.
- A practical feature-engineering layer built from technical indicators and temporal transformations.
- A LightGBM-based 5-day direction classifier with local model registry, metric storage, and prediction artifact generation.
- A drift-aware workflow using PSI-based monitoring and retraining triggers.
- A FastAPI and React stack that presents predictions, report history, market summaries, and model-health diagnostics in a professional interface.

II. RELATED WORK

Financial forecasting has historically been studied from several perspectives: market-efficiency theory, statistical time-series analysis, technical trading systems, classical machine learning, deep learning, and more recently MLOps-oriented deployment discipline. Fama's efficient-market formulation remains a foundational reference because it explains why persistent excess predictability is difficult to establish in public markets [1]. From an applied viewpoint, technical analysis remains influential in feature construction because moving averages, RSI, MACD, volatility bands, and momentum statistics transform raw prices into structured signals [2].

In structured tabular learning, tree-based ensembles have proven highly effective because they capture nonlinear interactions without requiring aggressive normalization assumptions. Random Forests introduced strong ensemble baselines [4], while XGBoost and LightGBM further advanced scalable gradient boosting for high-dimensional tabular tasks [5, 6]. In parallel, deep learning approaches such as LSTM networks, stacked autoencoders, and hybrid time-series architectures have been studied for directional financial prediction [7, 8, 9]. These models often improve representational capacity, but they also increase training complexity, reduce interpretability, and may overfit limited historical regimes.

Another relevant body of work concerns machine learning system reliability rather than forecasting architecture alone. Sculley et al. highlighted technical debt in ML systems and argued that hidden data dependencies, configuration fragility, and poor monitoring can make apparently successful models difficult to sustain [11]. Breck et al. later formalized production-readiness testing through a practical evaluation rubric [12]. Since market data is inherently non-stationary, concept-drift and feature-drift research is also directly relevant; Gama et al. provide one of the standard surveys on adaptation under distribution change [13]. Finally, interpretable ML has become increasingly important in high-stakes domains, and SHAP provides a principled foundation for feature-attribution analysis [14].

These studies collectively suggest that a credible financial ML system must be judged on two axes simultaneously: predictive behavior and engineering rigor. The present platform follows this principle by prioritizing pipeline reproducibility, artifact traceability, and model-health transparency in addition to baseline forecasting.

III. PROPOSED ARCHITECTURE

The architecture of the proposed system is layered so that each stage in the pipeline has a well-defined responsibility. The objective is not only to output a BUY, HOLD, or SELL-style signal, but also to preserve a traceable path from raw data to served artifacts.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 19, April 2026 Impact Factor: 8.2

A. Data Acquisition Layer

The first layer uses Yahoo Finance through the yfinance library to fetch daily OHLCV data for the NIFTY 50 symbol universe. The current repository artifacts show that 50 symbols were requested and 49 were successfully fetched in the stored raw snapshot, with one symbol failing in that particular cycle. This confirms that even in an academic setting, data ingestion must be treated as an operational component rather than a trivial preprocessing step.

B. Validation and Raw Storage Layer

Immediately after collection, the dataset is checked for required columns and elementary financial consistency. These checks include null detection, non-positive price detection, and high-low range consistency. Raw data is then preserved as versioned parquet snapshots, allowing re-use without repeated remote fetches and enabling reproducible weekly demonstrations.

C. Feature Engineering Layer

The feature-engineering component derives indicator families for each symbol separately. These include returns, price-to-SMA ratios, price-to-EMA ratios, RSI, MACD ratios, Bollinger Band width and position, ATR ratio, momentum variables, stochastic oscillator values, lagged returns, trend crossovers, intraday range percentages, and rolling volatility. The resulting processed dataset contains both predictor and target fields.

D. Training and Drift Control Layer

The core predictive model is a LightGBM classifier trained for 5-trading-day direction prediction. The platform also performs feature-drift checks using PSI on selected engineered features. If drift exceeds the threshold or the training is manually forced, a new model can be trained; otherwise the existing model artifact is reused. This architecture reduces unnecessary retraining while still responding to detectable distribution change.

E. Serving and Presentation Layer

The backend is implemented with FastAPI and exposes endpoints for health checks, market summary, model health, predictions, and reports. The frontend is implemented with React and presents four user-oriented workspaces: market overview, signal explorer, report history, and MLOps health. In addition, JSON and HTML report generation provides a static reporting channel for faculty review or project demonstration.

F. Monitoring Layer

Prometheus-compatible counters and gauges track drift score, model metrics, prediction counts, feature count, and last-training timestamp. Alert rules are defined for high drift, low F1-score, low accuracy, and lack of recent retraining. This layer is useful because it reframes the project as a system that can be supervised, not merely as a script that outputs predictions once.

IV. METHODOLOGY

The methodology mirrors the actual repository workflow. The following subsections describe each stage in detail.

A. Dataset Construction

The processed dataset used by the latest saved artifacts contains 35,289 rows and 42 columns, spanning 16 May 2023 to 17 April 2026. It covers 49 distinct NIFTY symbols in the most recent available feature snapshot. Each row corresponds to a symbol-date observation with both engineered indicators and forward-looking target fields.

B. Validation Logic

Let the raw market table be represented as $D = \{r1, r2, ..., rn\}$ where each record contains Date, Open, High, Low, Close, Volume, and Symbol. The validation step checks: schema completeness, null presence, strictly positive closing values, and logical consistency such that High >= Low. These checks prevent silent corruption from propagating into the training stage.

C. Feature Engineering Formulation

Daily return is computed as $Rt = (Ct - Ct\text{-}1) / Ct\text{-}1$, where Ct is the closing price at day t. The binary target for 5-day direction prediction is defined as $yt = 1$ if $(Ct+5 - Ct) / Ct > 0$, otherwise 0. Beyond raw return, the system constructs thirty predictive features. The principal feature groups are listed in Table I.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

TABLE I: Feature Groups Used by the Baseline Classifier

Group	Examples
Returns and lags	Returns, Lag_Return_1, Lag_Return_3, Lag_Return_5
Trend ratios	Price_SMA5_Ratio, Price_SMA20_Ratio, Price_EMA10_Ratio
Momentum	Momentum_5, Momentum_10, ROC_20
Oscillators	RSI, Stochastic_K, Stochastic_D
MACD family	MACD_Ratio, MACD_Signal_Ratio, MACD_Hist_Ratio
Volatility	BB_Width, ATR_Ratio, Volatility_20
Volume	Vol_Ratio, Volume_Change
Intraday structure	High_Low_Range_Pct, Close_Open_Pct
Calendar and crossover	Day_of_Week, SMA5_SMA20_Cross, MACD_Crossover

D. Model Training Strategy

The classifier is trained with a chronological split instead of a random split to preserve causality. The latest pipeline run uses: training rows: 28,185; validation rows: 3,528; test rows: 3,576. The time spans are: train: 16 May 2023 to 12 September 2025; validation: 15 September 2025 to 29 December 2025; test: 30 December 2025 to 17 April 2026. The choice of LightGBM is practical: it is fast on CPU, handles nonlinear interactions well, and fits the project goal of a locally executable MLOps demo. The implementation also optimizes the classification threshold on validation data before evaluating the final test predictions, which is preferable to always assuming a fixed threshold of 0.5.

E. Drift Detection

Feature drift is measured using the Population Stability Index (PSI): PSI = Sum_i (Ai - Ei) * ln(Ai / Ei), where Ei and Ai are the expected and actual proportions in the i-th histogram bin. In the implemented logic, drift is considered material when average PSI reaches at least 0.10 or a maximum feature PSI reaches at least 0.25. This offers a lightweight yet meaningful control for weekly snapshot comparison.

F. Serving Architecture

The API-level routing connects dashboard consumers, direct API clients, model-health endpoints, report endpoints, and the local artifact store. The FastAPI backend exposes endpoints including /health, /model/health, /predict/{sym}, /report/*, and /market/summary, all consumed by the React dashboard running on port 5173 and optionally by direct API clients via Swagger documentation.

V. RESULTS AND DISCUSSION

This section reports repository-backed results from the latest available saved artifacts. Importantly, the discussion distinguishes between system maturity and predictive strength. The project performs well as an MLOps demonstration, but the latest model does not yet achieve strong predictive discrimination.

A. Dataset and Artifact Summary

Table II summarizes the current data and artifact state extracted from the repository.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

TABLE II: Current Repository-Backed Data and Model Summary

Attribute	Value
Processed rows	35,289
Total columns	42
Engineered predictor count	30
Covered symbols	49
Date range	16 May 2023 to 17 April 2026
Latest report week	2026-W16
Latest market snapshot size	49 rows
Requested universe	50 symbols
Successful raw fetch count	49
Failed symbol in stored fetch summary	TATAMOTORS.NS

B. Performance Metrics

The latest saved model artifact (20260424_013208) reports the metrics shown in Table III. The classification threshold stored in the model metadata is approximately 0.48.

The confusion matrix recorded in the saved metrics is: [[459, 1597], [449, 1071]]. This reveals a prediction pattern with relatively high recall but many false positives. In other words, the model tends to flag upward movement often, yet its separation between positive and negative cases is weak. The low specificity and sub-0.5 ROC-AUC confirm that the current model is best interpreted as an operational baseline rather than a reliable financial signal engine. The repository also contains exploratory walk-forward evaluation results with an average accuracy of 50.58% and average ROC-AUC of 0.5129. Although these values are still modest, they are more realistic than a single fixed split and suggest that evaluation strategy materially affects perceived performance.

TABLE III: Performance of the Latest Saved LightGBM Classifier

Metric	Value
Accuracy	42.79%
Precision	40.14%
Recall	70.46%
F1-score	51.15%
ROC-AUC	0.4528
Log loss	0.7878
Specificity	22.32%
Acceptable flag	False

C. Prediction Signal Distribution

The generated report further helps interpret the current market context. In the latest report week (2026-W16), the platform produced 10 BUY, 31 HOLD, and 8 SELL signals across 49 symbols. The latest report also indicates that the consumer segment delivered the strongest grouped sector return, while IT and pharma were negative on average.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

D. Feature Importance Discussion

The latest saved feature-importance file ranks Volatility_20, ATR_Ratio, BB_Width, MACD_Signal_Ratio, and MACD_Hist_Ratio among the most influential variables. This indicates that the model is relying more on volatility structure and momentum dispersion than on any single directional cue. That behavior is plausible for short-horizon market classification, but it does not yet translate into strong out-of-sample decision quality. Table IV presents the top ten feature importances from the latest model artifact.

TABLE IV: Top Ten Feature Importances from the Latest Model Artifact

Feature	Importance
Volatility_20	916
ATR_Ratio	774
BB_Width	655
MACD_Signal_Ratio	641
MACD_Hist_Ratio	561
Vol_Ratio	557
ROC_20	555
Stochastic_D	549
Volume_Change	536
Lag_Return_3	524

E. Operational Success Versus Predictive Success

The strongest result of the project is operational coherence. From one local entrypoint, the system can fetch market data, validate it, engineer features, decide whether retraining is necessary, load or create model artifacts, generate predictions, build reports, and expose everything through a frontend and REST API. The current drift artifact also reports no material feature drift between the latest two compared processed snapshots, with a drift score of 0.0. By contrast, predictive success remains limited.

The saved model metadata explicitly marks the model as not acceptable according to the repository thresholds. This is actually a valuable academic result: it demonstrates that a well-built MLOps system can still host a weak baseline model, and that responsible reporting must separate infrastructure quality from forecasting quality.

VI. CHALLENGES AND LIMITATIONS

Several limitations remain in the current version of the platform. First, the model uses only price- and volume-derived technical indicators. It does not incorporate macroeconomic variables, sector-relative benchmarks, derivatives information, sentiment streams, or event-driven features. As a result, it captures only a narrow subset of market structure. Second, the baseline model is a tabular classifier trained on engineered snapshots rather than a richer temporal architecture. This simplifies deployment, but it may miss long-range sequential dependencies. Although deep sequence models such as LSTM variants are available in the literature [7, 8, 9], they were intentionally excluded from this first local version to keep the system reproducible and resource-efficient.

Third, data-source robustness remains a challenge. The pipeline depends on Yahoo Finance through yfinance, which is suitable for academic work but not a guaranteed production-grade market feed. The repository fetch summary already shows one failed symbol in the stored raw cycle, emphasizing that ingestion health should be monitored continuously. Fourth, the evaluation focus is directional classification rather than actionable trading strategy assessment. The present study does not include transaction costs, slippage, execution constraints, turnover control, or portfolio-level

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 19, April 2026 Impact Factor: 8.2

backtesting. Consequently, even if the metrics improve in future versions, they should not be interpreted as direct evidence of trading profitability. Finally, the monitoring stack is present but locally scoped. Prometheus metrics and Grafana configuration are included, yet the runtime path remains a local educational setup rather than a cloud-hosted production system.

VII. CONCLUSION AND FUTURE WORK

This paper presented a professional two-column research write-up of the NIFTY 50 Weekly Intelligence Platform, a local MLOps system for market data engineering, technical-feature generation, direction prediction, drift monitoring, API serving, dashboard analytics, and weekly reporting. The platform demonstrates that a financial ML project can be organized as a full lifecycle workflow rather than as an isolated training notebook.

The empirical findings show a clear distinction between engineering maturity and predictive maturity. The platform is structurally sound: it uses versioned raw and processed snapshots, model registry concepts, prediction artifacts, drift scoring, alert thresholds, report generation, and a multi-page frontend. However, the baseline LightGBM classifier still shows weak discriminative power on held-out data. This makes the project a realistic and academically honest study rather than an overstated forecasting claim.

Future work can extend the project in four directions:

- The feature layer can be enriched with sentiment, macroeconomic, and cross-sectional market context.
- The model layer can compare CatBoost, XGBoost, LSTM, temporal convolution, and transformer-based variants against the current LightGBM baseline.
- Interpretability can be deepened with SHAP-based local and global explanations.
- Evaluation can evolve toward walk-forward backtesting with transaction-cost-aware portfolio simulation.

REFERENCES

[1] E. F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," The Journal of Finance, vol. 25, no. 2, pp. 383–417, 1970.
[2] J. J. Murphy, Technical Analysis of the Financial Markets, New York, NY, USA: New York Institute of Finance, 1999.
[3] R. S. Tsay, Analysis of Financial Time Series, 3rd ed., Hoboken, NJ, USA: Wiley, 2010.
[4] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 785–794.
[6] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 3146–3154.
[7] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock Market's Price Movement Prediction With LSTM Neural Networks," in Proc. Int. Joint Conf. Neural Networks (IJCNN), 2017, pp. 1419–1426.
[8] W. Bao, J. Yue, and Y. Rao, "A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory," PLOS ONE, vol. 12, no. 7, e0180944, 2017.
[9] T. Fischer and C. Krauss, "Deep Learning With Long Short-Term Memory Networks for Financial Market Predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.
[10] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial Time Series Forecasting With Deep Learning: A Systematic Literature Review: 2005–2019," Applied Soft Computing, vol. 90, 106181, 2020.
[11] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in Advances in Neural Information Processing Systems, vol. 28, 2015, pp. 2503–2511.
[12] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction," in Proc. IEEE Int. Conf. Big Data, 2017, pp. 1123–1132.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

[13] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 1–37, 2014.
[14] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.
[15] J. Bollen, H. Mao, and X. Zeng, "Twitter Mood Predicts the Stock Market," Journal of Computational Science, vol. 2, no. 1, pp. 1–8, 2011.