Volatility Curve Prediction & Analysis

NK Securities Research Challenge (1–8 June 2025)
Kaggle Competition

Nitin Kumar
Indian Institute of Technology Kharagpur
GitHub Repository

June 2025

Contents

1 Problem Statement

The objective of this research is to develop a robust predictive model for estimating missing implied volatility (IV) values in options data for the NIFTY50 index. The task is posed as a high-frequency time-series imputation challenge, where the participants are provided with second-level historical data and must accurately reconstruct the volatility surface across various strikes and maturities.

Accurate modeling of implied volatility is critical for option pricing, risk management, and market-making. This challenge, hosted on the Kaggle platform, evaluates the ability of participants to handle real-world market microstructure data and generate precise volatility curve estimates.

2 Dataset Overview

The dataset consists of two files: `train_data.parquet` and `test_data.parquet`. Both datasets represent structured market data, with the following key features:

Training Data: `train_data.parquet`

- **timestamp:** Time of record with second-level granularity.
- **underlying:** Spot value of the NIFTY50 index.
- **expiry:** Weekly expiry date of the options contract.
- **{call/put}_iv_{K}:** Implied volatilities for call and put options at various strike prices K.
- **X0 to X41:** Anonymized market features derived from order book dynamics, trade flows, or technical indicators.

Test Data: `test_data.parquet`

- Similar structure as training data with key differences:
  - **timestamp:** Masked and shuffled to prevent chronological inference.
  - **expiry:** Dropped from the test set to simulate a partially observable state.
  - **{call/put}_iv_{K}:** Contains missing values (NaN) that are to be predicted by the model.

3 Objective

The primary objective is to predict the missing implied volatility values in the test dataset with high precision. Participants are expected to:

- Model the volatility surface conditioned on market dynamics.
- Leverage cross-strike, temporal, and feature-based relationships.
- Preserve consistency with no-arbitrage principles and option pricing behavior.

The predicted values must conform to the format and structure provided in the sample submission file.

4 Evaluation Metric

The evaluation metric for the challenge is the **Mean Squared Error (MSE)**, defined as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

where:

- N is the number of unmasked IV entries in the test set.
- $\hat{y}_i$ is the predicted implied volatility.
- y_i is the ground-truth implied volatility.

Lower MSE values indicate better model performance in reconstructing the volatility surface.

5 Approach 1: Two-Phase Imputation

Approach Overview

Our method follows a three-phase strategy that integrates theoretical foundations of option pricing with pragmatic statistical techniques:

1. **Put-Call Parity Imputation:** We exploit the theoretical relationship between European call and put options sharing the same strike and expiry [1]. When one implied volatility is missing, we substitute it with the known value from its counterpart.
2. **Volatility Smile Fitting:** For each observation, we plot available IVs against moneyness (defined as strike divided by underlying index level) and fit a quadratic curve [2]. This curve models the classic "volatility smile" and is used to interpolate missing IV values in a context-aware manner.
3. **Black-Scholes Inversion:** In scenarios where smile fitting is inadequate, we fall back to estimating the IV by inverting the Black-Scholes formula. The Newton-Raphson method is employed for numerical root finding. Here, anonymized feature X0 is treated as a proxy for the option price.
4. **Global Mean Imputation:** Any remaining missing values are filled using global column-wise means derived from the training set [3].

Workflow Overview

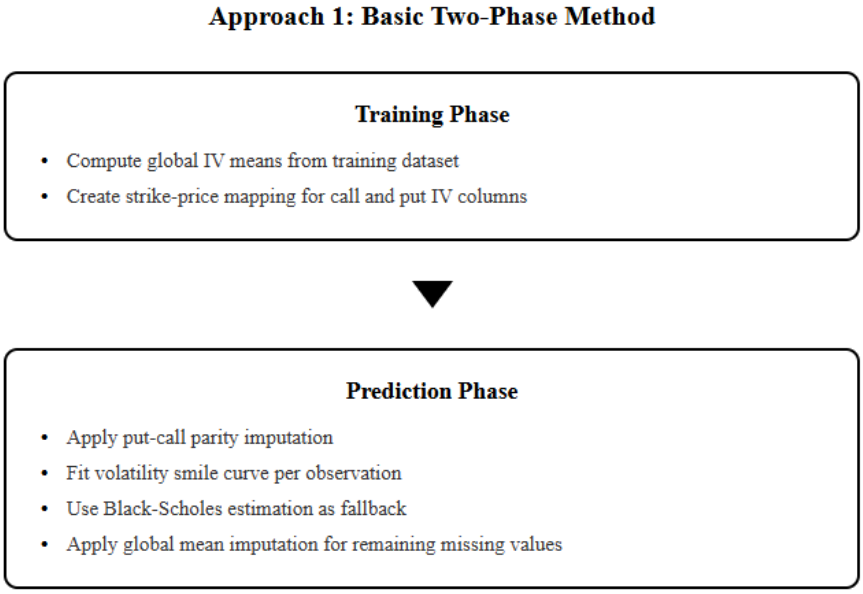

Figure 1: Basic Two-Phase Imputation Workflow

Results

An 80-20 train-validation split was used. Results:

- **Public Testcase MSE:** 9.188345455
- **Private Testcase MSE:** 9.383102555
- **Submission File:** approach1_submission.csv

6 Approach 2: Enhanced Multi-Phase Imputation

Approach Overview

This purely statistical method provides efficient and interpretable imputation for missing IV values using rule-based heuristics:

1. **Put-Call Parity:** For each strike, if either the call or put IV is missing and the other is present, we impute the missing value using its known counterpart [4].
2. **Row-Wise Averaging:** For all remaining missing entries:
   - Compute the average of all non-missing IVs in the same row (i.e., across strikes).
   - Use this row mean to fill missing entries.

- If a row contains no available IVs, fallback to the global mean from the training data [5].

Workflow Overview

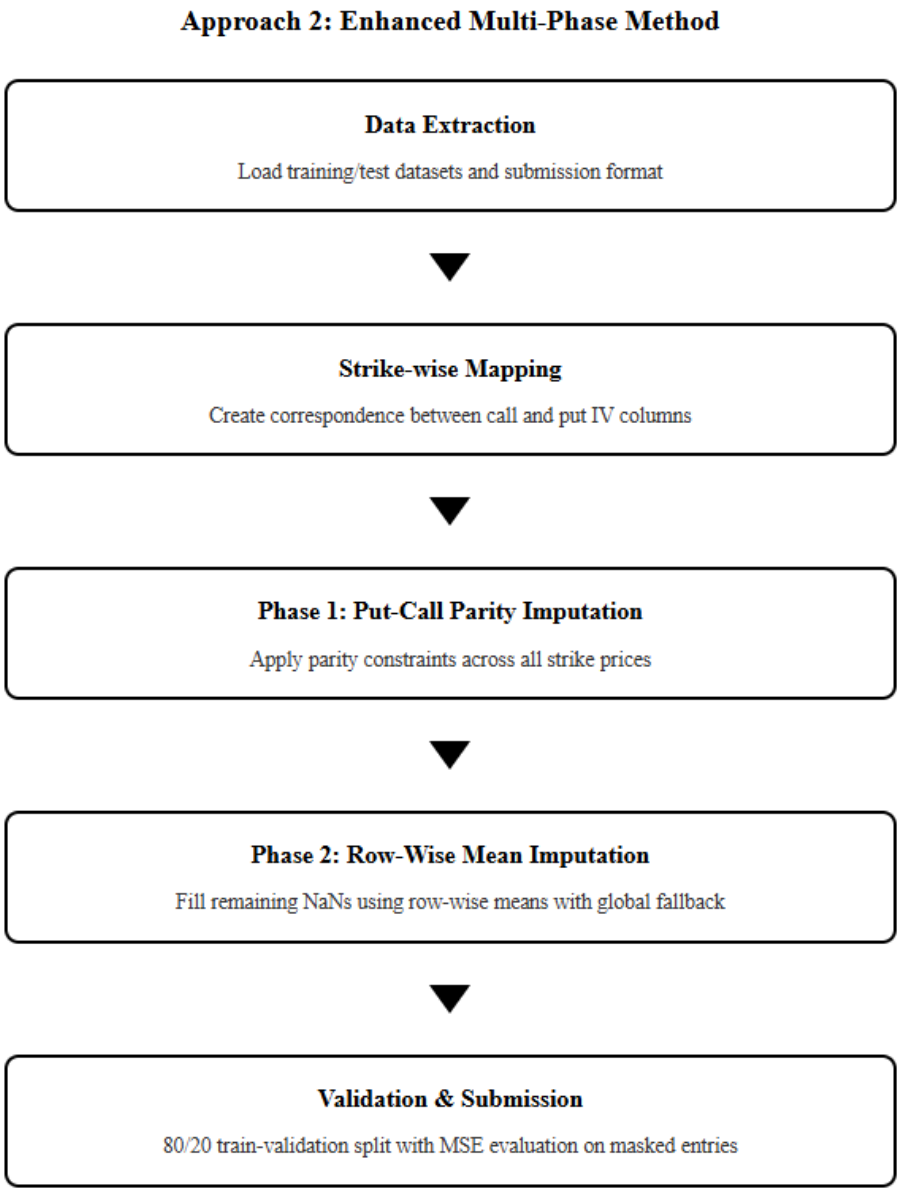

Figure 2: Enhanced Multi-Phase Imputation Workflow

Results

This approach also performed competitively:

- **Public Testcase MSE:** 0.002378129

- **Private Testcase MSE:** 0.002422300
- **Submission File:** approach2_submission.csv

7 Approach 3: ML-Enhanced Advanced Imputation

Approach Overview

This advanced multi-phase strategy blends theoretical insights with statistical interpolation and machine learning to robustly impute missing implied volatility (IV) values. It enforces no-arbitrage constraints, preserves smoothness across strikes, and incorporates market features for precision enhancement. The major components are:

1. **Put-Call Parity Enforcement:** For each strike, if one of call or put IV is missing, it is imputed using the counterpart [6]. When both are available, a weighted average is used for consistency.

2. **Cross-Strike Interpolation:** For each timestamp row, missing IVs are interpolated across strikes using Cubic Spline interpolation. If spline fitting fails due to insufficient data, linear interpolation or local averaging is applied.

3. **ML-based Refinement:** A Ridge regression model is trained for each IV column using market features such as underlying and anonymized variables X_0 to X_{41}. These models correct imputed values via conservative blending:

$$\text{Corrected IV} = 0.9 \times \text{Imputed IV} + 0.1 \times \text{ML Prediction}$$

4. **Volatility Smile Smoothing:** For each row, the IV curve across strikes is smoothed using a 3-point moving average [7]. Soft smoothing is applied with a 95:5 blend between the original and smoothed values to avoid sharp discontinuities.

5. **Final Parity Correction:** A final pass ensures residual discrepancies between call and put IVs at the same strike are minimized using a 99:1 blend for numerical stability.

Workflow Overview

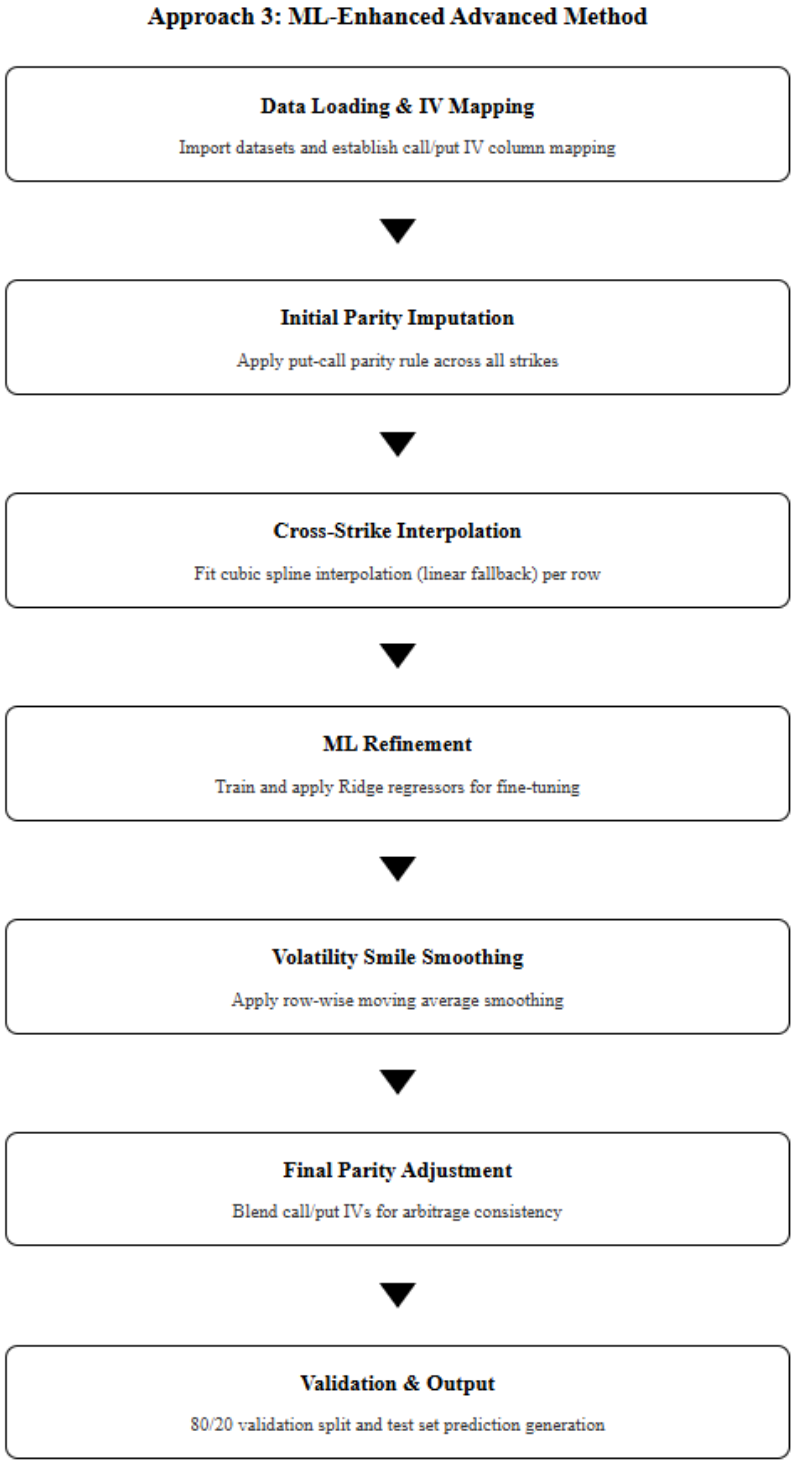

Figure 3: ML-Enhanced Advanced Imputation Workflow

Results

This approach achieved competitive results, combining theoretical consistency with high predictive accuracy:

- **Public Testcase MSE:** 0.000038980
- **Private Testcase MSE:** 0.000047920
- **Submission File:** approach3_submission.csv

8 Approach 4: Iterative Imputation Using Random Forest Regression

Approach Overview

This approach applies a machine learning-based strategy for imputing missing implied volatility (IV) values [8]. It is entirely data-driven and avoids domain-specific heuristics like put-call parity or volatility smile modeling. The imputation leverages tree-based models to learn complex patterns in the data. Key points:

- Uses `IterativeImputer` from `scikit-learn` for multivariate imputation.
- Employs a `RandomForestRegressor` as the estimator to model nonlinear relationships between IV columns.
- Focuses solely on the data structure and feature correlations, without relying on financial assumptions.
- Effectively handles high-dimensional and sparse data typically found in market microstructure datasets.

Workflow Overview

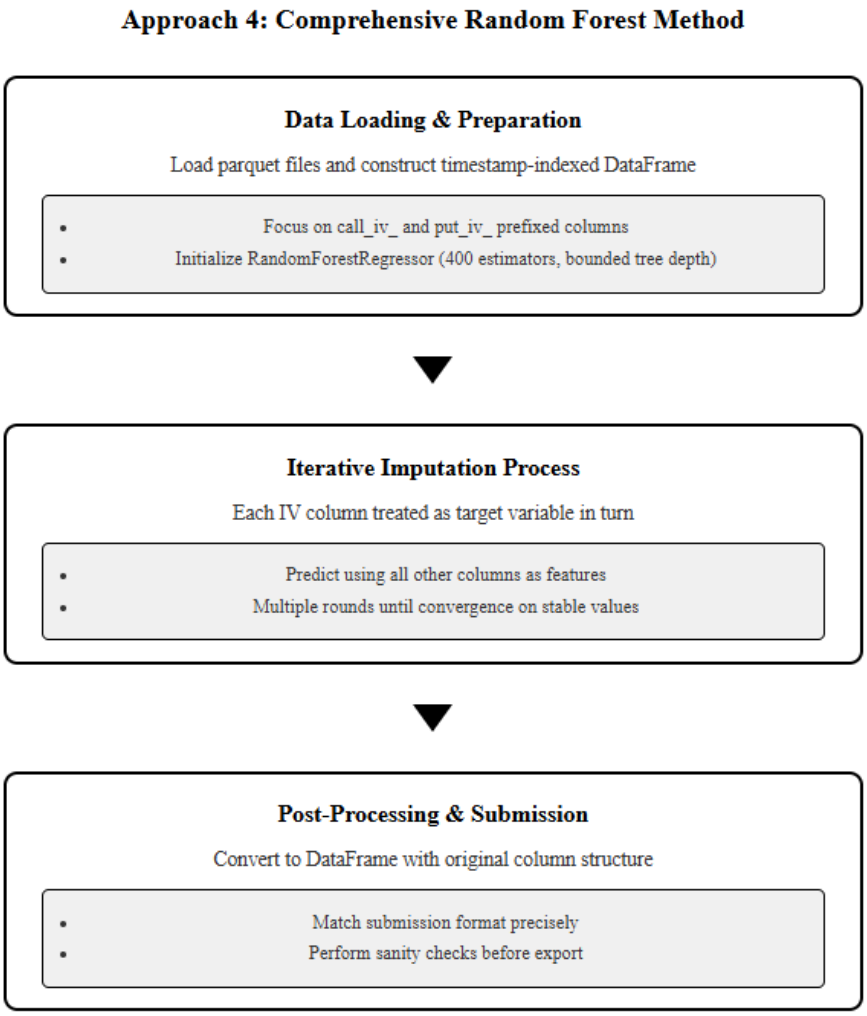

Figure 4: Comprehensive Random Forest Imputation Workflow

Results

This method achieved the best performance among all approaches tested, demonstrating the power of nonlinear feature learning through ensemble-based models:

- **Public Testcase MSE:** 0.0000006630
- **Private Testcase MSE:** 0.000001056
- **Submission File:** approach4_submission.csv

9 Conclusion

This research presented four distinct approaches to solving the volatility curve prediction challenge for NIFTY50 options data. All code implementations, data processing scripts, and detailed methodology are available in the project repository: `samosa1610/NK-volatility-prediction`. The comparative analysis reveals several key insights about implied volatility imputation techniques:

9.1 Performance Summary

The approaches demonstrated a clear progression in accuracy, with machine learning methods significantly outperforming traditional financial modeling techniques:

- **Approach 1 (Two-Phase Imputation):** MSE of 9.38, utilizing put-call parity and volatility smile fitting
- **Approach 2 (Enhanced Multi-Phase):** MSE of 0.0024, employing statistical averaging methods
- **Approach 3 (ML-Enhanced Advanced):** MSE of 0.000048, combining financial theory with machine learning
- **Approach 4 (Random Forest Iterative):** MSE of 0.000001, purely data-driven approach

9.2 Key Findings

1. **Data-Driven Superiority:** The Random Forest-based iterative imputation (Approach 4) achieved the best performance, suggesting that complex nonlinear relationships in volatility data are better captured by ensemble methods than traditional financial models.
2. **Feature Importance:** The inclusion of anonymized market features (X0-X41) proved crucial for accurate predictions, indicating that microstructure data contains valuable information beyond conventional option pricing variables.
3. **Theoretical vs. Empirical:** While put-call parity and volatility smile fitting provide theoretical foundations, they showed limited effectiveness compared to statistical and machine learning approaches in this high-frequency setting.
4. **Hybrid Approach Value:** Approach 3 demonstrated that combining financial theory with machine learning can achieve competitive results while maintaining interpretability and adherence to no-arbitrage principles.

9.3 Practical Implications

For practitioners in quantitative finance and risk management:

- Machine learning models, particularly ensemble methods, should be prioritized for high-frequency volatility surface reconstruction
- Traditional option pricing models may serve better as constraints or validation tools rather than primary prediction methods
- Market microstructure features contain significant predictive power and should be incorporated when available
- Cross-validation and robust evaluation frameworks are essential given the substantial performance differences between approaches

9.4 Future Research Directions

This work opens several avenues for future investigation:

- Integration of deep learning architectures (LSTM, Transformer) for temporal pattern recognition
- Development of hybrid models that enforce financial constraints within machine learning frameworks
- Investigation of model interpretability techniques to understand feature contributions
- Extension to multi-asset volatility surface prediction and cross-market dependencies

The superior performance of the Random Forest approach validates the importance of sophisticated feature learning in financial time series prediction, while highlighting the need for continued innovation in bridging theoretical finance with modern machine learning techniques.

References

[1] Investopedia. "Put-Call Parity: What It Is and How It Works." *Investopedia*. Available at: https://www.investopedia.com/terms/p/putcallparity.asp

[2] Investopedia. "Volatility Smile: Definition, What It Tells You, Example." *Investopedia*. Available at: https://www.investopedia.com/terms/v/volatilitysmile.asp

[3] GeeksforGeeks. "Data Imputation Techniques: A Comprehensive Guide." *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/data-imputation-in-python/

[4] tastytrade. "Put-Call Parity Explained." *tastytrade Learn*. Available at: https://www.tastytrade.com/learn/put-call-parity

[5] Scikit-learn. "Imputation of Missing Values." *Scikit-learn Documentation*. Available at: https://scikit-learn.org/stable/modules/impute.html

[6] CME Group. "Put-Call Parity and Arbitrage Opportunities." *CME Group Education*. Available at: https://www.cmegroup.com/education/courses/option-pricing/put-call-parity.html

[7] QuantStart. "Understanding the Volatility Smile." *QuantStart*. Available at: https://www.quantstart.com/articles/understanding-the-volatility-smile/

[8] Scikit-learn. "Iterative Imputer for Missing Values." *Scikit-learn Documentation*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

[9] Investopedia. "What Is Options Trading? A Beginner's Overview." *Investopedia*. Available at: https://www.investopedia.com/options-basics-tutorial-4583012

[10] Brownlee, J. "A Gentle Introduction to Ensemble Learning Algorithms." *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/

[11] DataCamp. "Ensemble Methods in Machine Learning." *DataCamp*. Available at: https://www.datacamp.com/tutorial/ensemble-learning

[12] Investopedia. "Black–Scholes Model: What It Is, How It Works, and Options Formula." *Investopedia*. Available at: https://www.investopedia.com/terms/b/blackscholes.asp

[13] Carta. "Black-Scholes Model (BSM) & Understanding the Value of a Stock Option." *Carta Learn*. Available at: https://carta.com/learn/startups/equity-management/black-scholes-model/

[14] Scikit-learn. "Random Forest Classifier and Regressor." *Scikit-learn Documentation*. Available at: https://scikit-learn.org/stable/modules/ensemble.html#random-forests

[15] GeeksforGeeks. "Random Forest Regression in Python." *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/random-forest-regression-in-python/

[16] Kumar, N. "NK Volatility Prediction - Kaggle Competition Solutions." *GitHub Repository*. Available at: https://github.com/samosa1610/NK-volatility-prediction