NK Securities Hackathon

Solution Documentation

Author: Tejas Shrivastava
Date: June 2025

Problem Understanding and Initial Strategy

The challenge involved predicting implied volatilities (IV) across different strike prices for call and put options, where the training and test sets had mismatched strike columns. This mismatch immediately suggested that traditional supervised learning approaches would be insufficient — the problem required interpolation and extrapolation of volatility surfaces.

My approach evolved through three conceptual phases, each building upon insights from the previous:

Phase 1: Traditional curve fitting with machine learning **Phase 2:** Deep learning with contextual modeling **Phase 3:** Formulating the task as a missing data imputation problem, solved through iterative techniques.

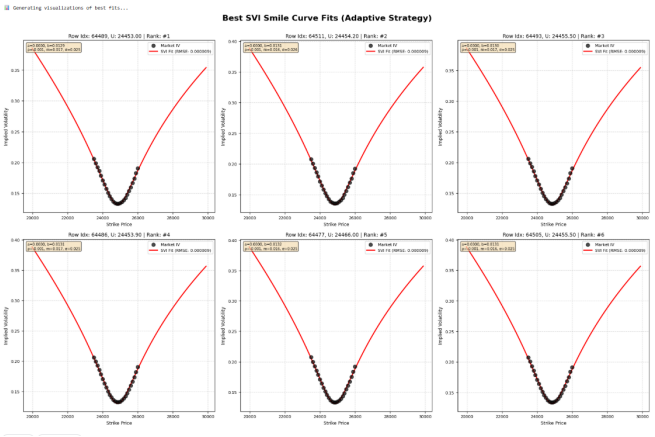
Phase 1: Parametric Curve Fitting Approaches

Methodology

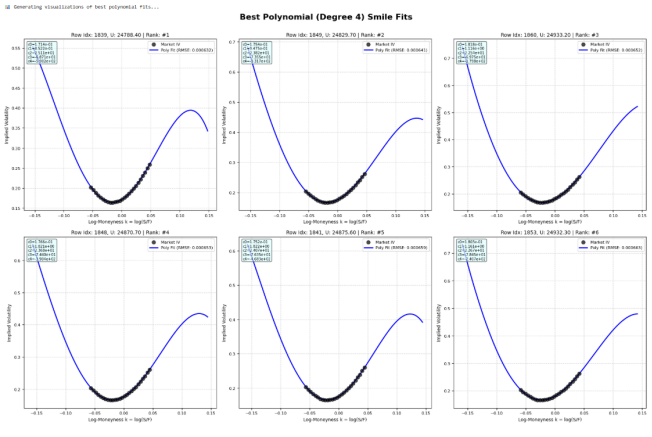
The initial strategy treated this as a curve fitting problem. Since implied volatility typically follows well-known parametric forms across strikes, I attempted to:

1. Fit standard volatility curves (quadratic, SVI) to each training sample's available IV points.
2. Use these fitted curves to interpolate IVs at the test set's required strike prices.
3. Now since the mismatch of strike prices between train and test dataset was resolved, the next step was to Train boosting models (XGBoost, LightGBM) on market features to predict the IVs.

As an alternative, I explored fitting SVI (Stochastic Volatility Inspired) curves numerically and training models to predict the five SVI parameters directly from market features.



SVI Curves



Polynomial Curves

Figure 1: Numerically fitted curves using differential evolution

Key Insights and Limitations

Public Score $\sim 10^{-3}$

Results: Both approaches yielded poor performance, leading to two critical realizations:

Insight 1: **Ignoring Available Test Information:** These methods completely ignored the partially available IVs in the test set — a significant waste of valuable information that could constrain predictions.

Insight 2: **Weak Predictive Power of Market Features:** Even when accounting for test set information, performance was still very poor, suggesting that features(underlying, X0, X1...X41) had limited predictive power for individual strike-level IVs.

These insights fundamentally redirected my approach toward methods that could leverage all available information in the test set.

Phase 2: Autoencoders-Based Approach

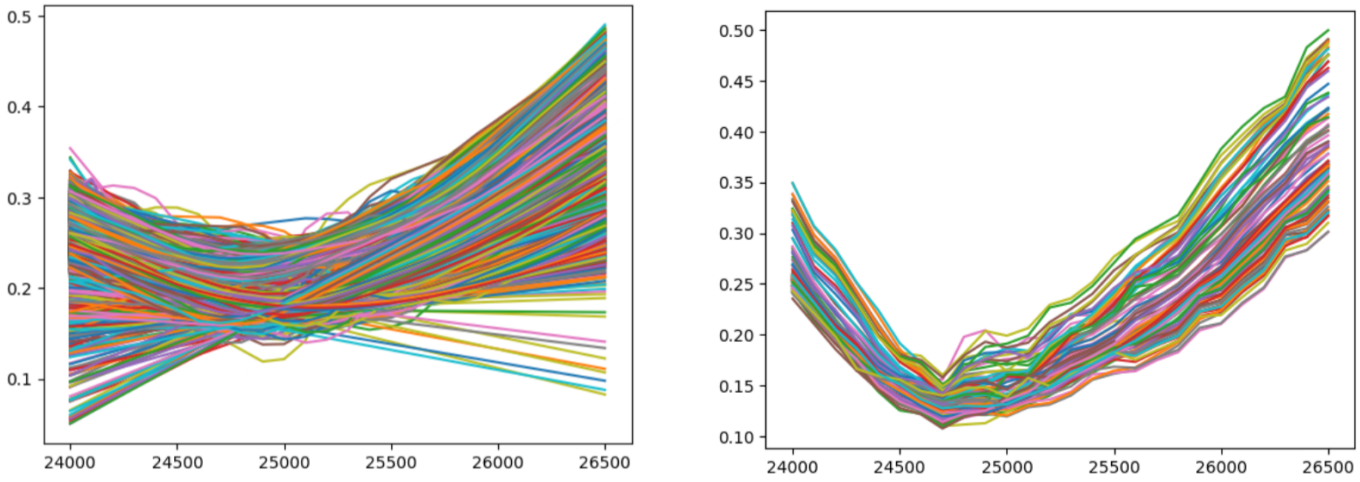
Methodology and Rationale

Building on the insight that known IV values in the test set should be utilized, I adopted an autoencoder-based approach. The primary objective was to replicate the test set's missing data patterns during training, enabling the model to learn realistic imputation strategies. The training process involved the following steps:

1. Randomly mask portions of complete IV vectors in the training set
2. Train an autoencoder to predict masked IVs using visible ones
3. Apply this trained model to impute missing IVs in the test set

Technical Innovations

Masking Strategy Evolution: Initial uniform random masking caused instability at extreme strikes. I developed an edge-focused masking scheme that more aggressively masked extreme strikes during training, better simulating real test conditions.



Uniform Masking

Aggressive Edge Masking

Figure 2: Comparison of Masking Techniques

Feature Relevance Analysis

This phase provided crucial validation of earlier suspicions about market features through multiple analytical approaches:

- **Feature Regularization:** L1/L2 regularization in the model consistently suppressed weights for market features
- **Mutual Information Analysis:** Yielded a low mean score of 0.12 between market features and strike-level IVs
- **Correlation Analysis:** Pearson correlation confirmed weak linear relationships

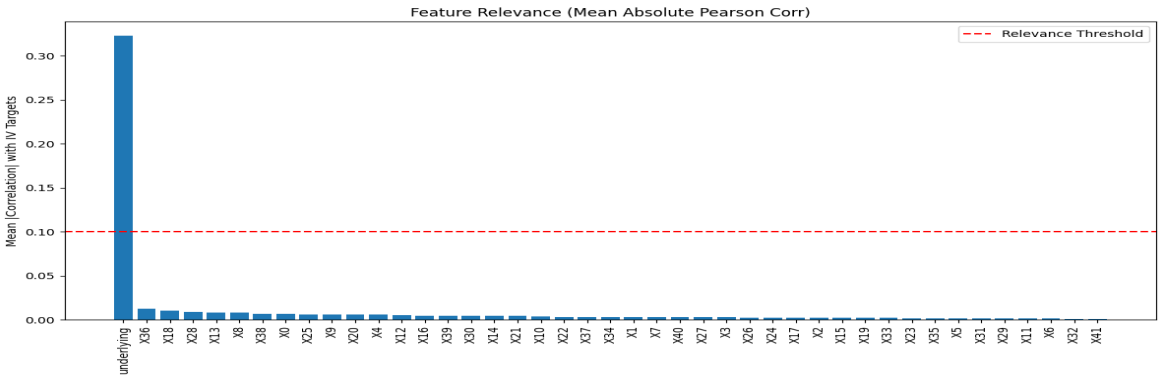


Figure 3: Pearson Correlation Between Market Features and Strike-Level IVs

Public Score $\sim 10^{-5}$ to 10^{-6}

Critical Breakthrough in Problem Framing

Key Insight: The autoencoder experiments led to a fundamental reframing: instead of treating this as a sequence modeling problem, treat each missing IV as a supervised learning target where other available IVs serve as features.

This realization suggested that the problem was fundamentally about **missing data imputation** rather than curve fitting or sequence modeling.

Phase 3: MICE-Based Iterative Imputation (Final Solution)

Conceptual Foundation

The insights from previous phases converged on a powerful realization: this problem is optimally solved as iterative missing data imputation. MICE (Multiple Imputation by Chained Equations) provides an elegant framework for this approach.

Understanding MICE: The Algorithm

MICE operates on a simple but powerful principle: use all available information to predict each missing value iteratively until convergence.

Detailed MICE Process:

1. **Initialization:** Fill all missing values with initial estimates (e.g., column medians)
2. **Iterative Chained Regression:** For each column containing missing values:
   - Designate this column as the target variable
   - Use all other columns (both originally observed and currently imputed) as features
   - Train a regression model on rows where the target is observed
   - Predict missing values in the target column using this model

- Update the dataset with these new predictions

3. **Convergence:** Repeat step 2 until the imputed values stabilize (typically 5-10 iterations)

Why MICE is Optimal for This Problem:

- **Leverages All Available Information:** Each prediction uses every other available IV value, maximizing information utilization
- **Captures Complex Inter-Column Dependencies:** The iterative process allows complex, non-linear relationships between strikes to emerge
- **Naturally Handles Varying Missing Patterns:** MICE adapts to different missing data patterns across samples
- **No Distributional Assumptions:** Unlike parametric curve fitting, MICE makes no assumptions about volatility surface shape

Key Experimental Discoveries

Training Data Integration Experiments: Initially, I attempted to use training data by processing it in batches with the test set, then averaging results. Despite extensive batch size tuning, this yielded only marginal improvements over autoencoder approaches.

The Breakthrough Discovery: Excluding training data entirely and applying MICE only to the test set produced a dramatic performance improvement.

Why Training Data Exclusion Worked:

- **Clustering Separation:** Final predictions and training set samples formed clearly distinct clusters, indicating fundamental differences in IV structure
- **Preserved Test Structure:** Focusing solely on the test set allowed the model to better capture its inherent patterns without being influenced by mismatched training behaviors
- **Avoided Feature Drift:** Including training data risked introducing patterns not present in the test set, reducing imputation accuracy

Model Selection: Why Tree-Based Regressors Excelled

Through systematic comparison of regression algorithms within the MICE framework, `ExtraTreesRegressor` emerged as the clear winner over gradient boosting methods.

Theoretical Advantages of Tree-Based Models in MICE:

1. **Parallel Learning Architecture:** Random forests build trees independently, avoiding the sequential error accumulation inherent in boosting algorithms. In iterative imputation, this prevents compounding of imputation errors across iterations.
2. **Robust Handling of Sparse Features:** When some IV values are missing, tree-based models naturally handle the resulting sparsity without requiring extensive hyperparameter tuning.
3. **Natural Multi-Target Compatibility:** Forest models excel at capturing complex, non-linear relationships between multiple correlated targets (different strike IVs) without overfitting.
4. **Stability Under Iterative Updates:** The ensemble nature of forests provides stability as imputed values change across MICE iterations, unlike boosting which can be sensitive to input perturbations.

Final Architecture and Performance

The winning solution combined:

- **MICE iterative imputation** applied exclusively to test data excluding all the market features. X0...x41
- **ExtraTreesRegressor** as the base estimator with optimized hyperparameters

- **Feature set limited to** IV values and underlying price (market features excluded based on Phase 2 analysis)
- **5-10 MICE iterations** for convergence
- To further boost accuracy, perform a **weighted ensemble** of multiple solutions derived from varying parameter configurations.

Public Score $\sim 10^{-7}$

Conclusion: The Evolution of Understanding

This solution represents a journey through different conceptual frameworks, where each phase provided crucial insights that informed the next:

From Phase 1: Understanding that available test information must be leveraged, and that market features have limited predictive power.

From Phase 2: Confirming the weakness of market features through rigorous analysis, and recognizing that the core challenge is missing data imputation rather than curve fitting.

From Phase 3: Discovering that distributional coherence within the test set is more valuable than additional training data, and that tree-based models provide optimal stability for iterative imputation.

The final MICE-based approach succeeded because it aligned perfectly with the problem's fundamental nature: leveraging the rich inter-dependencies among implied volatilities across strikes to recover missing values through iterative, stable regression.

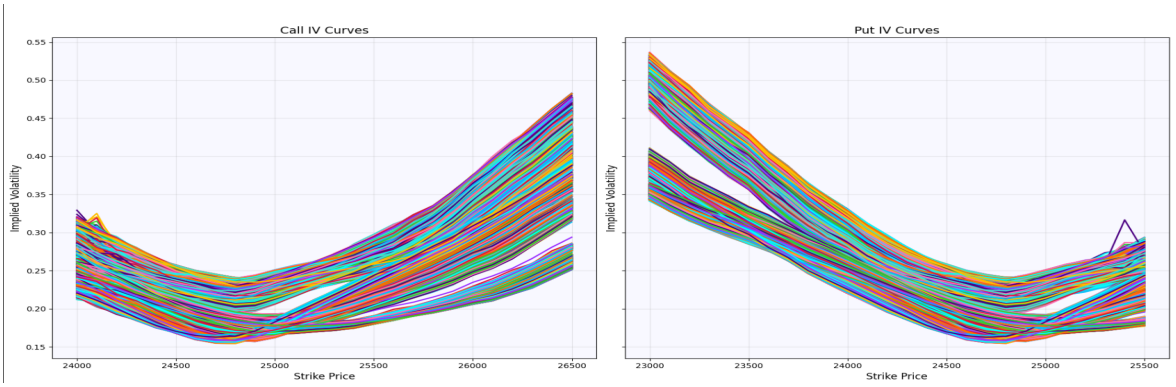


Figure 4: Final Submission Plots