Real-time CPI Forecasting using Machine Learning

ECO723 Course Project

Pranav Krishna (230772)
Aradhana Papnai (240167)
Kumar Natrajan (231250072)
Pallab Mandal (220746)
Nitin Gautam (220733)

Indian Institute of Technology, Kanpur

July 7, 2025

Contents

Introduction

The Challenge: Forecasting Inflation

Forecasting inflation is crucial for central banks and policymakers to maintain economic stability. The **Consumer Price Index (CPI)** serves as a primary measure of inflation.

This study applies **machine learning models** to forecast CPI in India, using time series data from January 2014 to May 2025.

Dataset Overview

Dataset Characteristics:

- **Dimensions:** 136 observations $\times$ 86 features
- **Time Period:** Jan 2014 - May 2025
- **Target:** Consumer Food Price Index Combined
- **Features:** The dataset includes 29 core economic indicators, each broken down into **Rural**, **Urban**, and **Combined** measurements, resulting in a total of 86 features. These cover categories like food, fuel, housing, and transportation.

Key Limitations:

Small Sample Size

Only 136 monthly observations, which is very small for traditional machine learning models.

High Dimensionality

86 features for only 136 samples leads to the "curse of dimensionality", increasing the risk of overfitting.

Seasonality

Economic data often has yearly patterns that need to be accounted for.

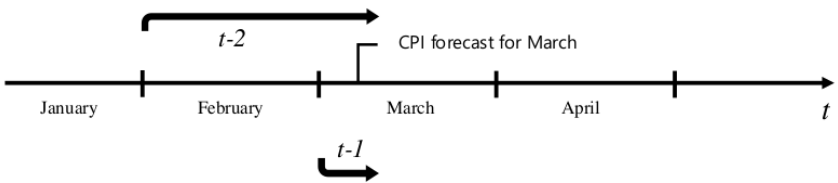
Methodology

Core Principle: Avoiding Data Leakage

Data leakage is a critical error in time series modeling where future information is accidentally used to train the model, leading to overly optimistic results that fail in practice.

Our Approach: Lag Features

- We only use **past data** to predict the **present**.
- This is achieved by creating "lagged" features. For example, to predict CPI for February, we use data from January.
- This simulates a real-world forecasting scenario.



Experiment 1: Baseline Model (t-1 Lag)

Objective: Establish a baseline performance using only the most recent past data (t-1).

Features: All 86 features from the previous month.

Results

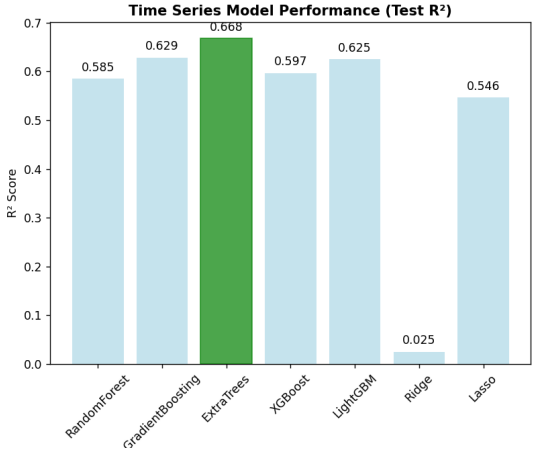
The baseline model shows some predictive power, but the test MAE is modest, indicating room for improvement.

Table: Performance with t-1 Lag

Model	Train MAE	Test MAE
RandomForest	1.1051	1.4354
GradientBoosting	1.5458	1.6132
ExtraTrees	1.2414	1.4928
XGBoost	1.6080	1.7368
LightGBM	1.6494	1.6822
Ridge	0.9828	2.5472
Lasso	1.5172	1.8306

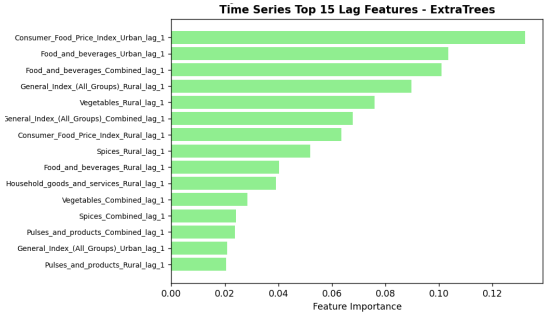
Experiment 1: Test R^2

- The **ExtraTrees** model achieved the highest Test R^2 score of **0.668**, indicating the best performance among the models tested.
- **GradientBoosting** and **LightGBM** also showed strong results, with R^2 scores of **0.629** and **0.625**, respectively.
- The linear models, **Ridge** and **Lasso**, performed poorly, suggesting that non-linear relationships are important in this dataset.



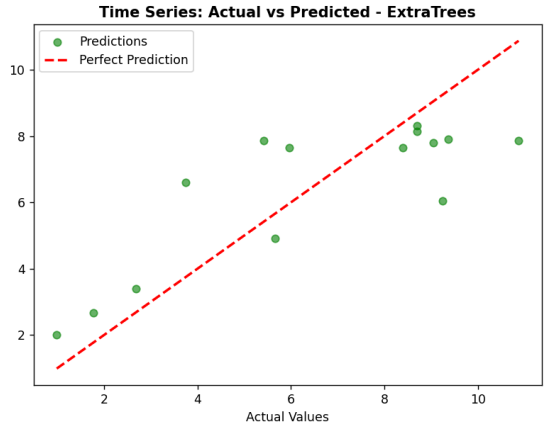
Experiment 1: Feature Importance

- The most influential feature is the **Consumer Food Price Index (Urban)** from the previous month (lag 1).
- **Food and beverages** indices (Urban, Combined, and Rural) are also highly significant predictors.
- This highlights the strong auto-regressive nature of the CPI data, where past values are strong predictors of future values.



Experiment 1: Actual vs. Predicted

- This plot shows the ExtraTrees model's predictions against the actual values.
- The points are clustered around the red dashed line, which represents a perfect prediction.
- While there is some variance, the model captures the general trend of the data.



Experiment 2: Adding More History (t-2 Lag)

Objective: Does including data from two months ago (t-2) improve the forecast?

Features: All 86 features from t-1 and t-2.

Results

Adding the t-2 lag provides a 4.6% improvement in Test MAE for the RandomForest model. This suggests that information from two months prior is valuable.

Table: Performance with t-1 & t-2 Lags

Model	Train MAE	Test MAE
RandomForest	1.0839	1.5024
GradientBoosting	1.5441	1.6446
ExtraTrees	1.1657	1.6485
XGBoost	1.5432	1.7459
LightGBM	1.6115	1.7604
Ridge	0.5728	1.5841
Lasso	1.5225	1.8311

Experiment 3: Diminishing Returns (t-3 Lag)

Objective: Is even more historical data (t-3) better?

Features: All 86 features from t-1, t-2, and t-3.

Results

Adding the t-3 lag provides **no significant improvement** over the t-1, t-2 model. The model complexity increases without a corresponding benefit in accuracy.

Table: Performance with t-1, t-2 & t-3 Lags

Model	Train MAE	Test MAE
RandomForest	1.0951	1.5555
GradientBoosting	1.5288	1.5747
ExtraTrees	1.1635	1.4344
XGBoost	1.5619	1.6866
LightGBM	1.6267	1.8403
Ridge	0.7332	1.4377
Lasso	1.5240	1.8261

Experiment 4: Taming the Curse of Dimensionality

Problem: With lags, our feature count explodes (86 features * 2 lags = 172 features), increasing the risk of overfitting.

Solution: Correlation-Based Feature Selection

1. Calculate the correlation of each lagged feature with the target CPI.
2. Rank features by their absolute correlation value.
3. Select the top ‘k‘ features. In our case, between 50 and 100.

Benefit

This focuses the model on the most relevant predictors, reducing noise and improving generalization.

Experiment 5: Capturing Seasonality (t-12 Lag)

Objective: Can we improve the model by accounting for yearly seasonal patterns?
Features: Add the t-12 lag (data from the same month in the previous year) to our best model.

Results

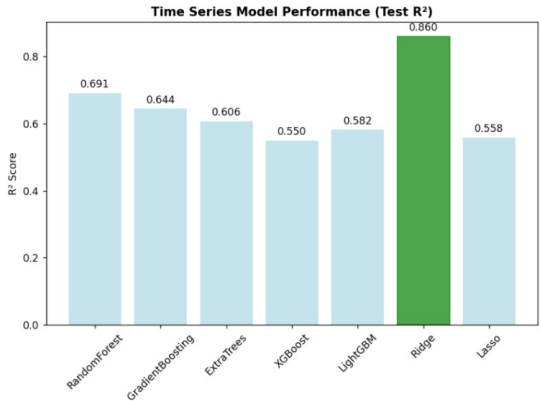
Including the t-12 lag provides a **noticeable boost** in performance, confirming the presence of seasonality in the data. The Ridge model's Test MAE dropped to **0.9279**, a **35.3%** improvement over the baseline RandomForest model.

Table: Performance with Seasonal Lag (t-1, t-12)

Model	Train MAE	Test MAE
RandomForest	1.0328	1.1653
GradientBoosting	1.5194	1.6533
ExtraTrees	1.1587	1.6982
XGBoost	1.5986	1.8039
LightGBM	1.5918	2.0131
Ridge	0.8856	0.9279
Lasso	1.4944	1.8146

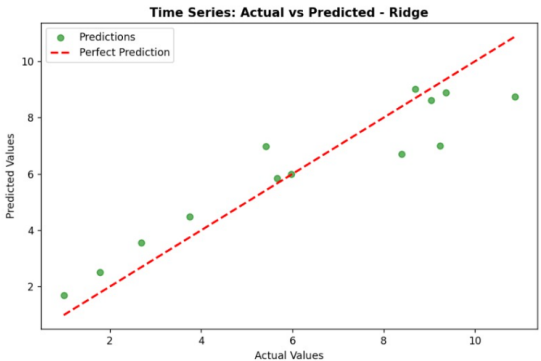
Experiment 5: Test R^2

- The **Ridge** model now achieves the highest Test R^2 score of **0.860**, a significant improvement.
- The performance of tree-based models like **RandomForest** and **GradientBoosting** also improved, but Ridge is the clear winner.
- This suggests that the linear model with regularization is better at capturing the seasonal pattern without overfitting.



Experiment 5: Actual vs. Predicted

- This plot shows the Ridge model's predictions against the actual values.
- The predictions are now much closer to the perfect prediction line, indicating a better fit.
- The model is now better at predicting the peaks and troughs in the data.



Results & Conclusion

Final Model Performance

Table: Final Model with Lags (t-1, t-12) and Feature Selection

Model	Train R^2	Test R^2	Train MAE	Test MAE
RandomForest	0.8304	0.6911	1.0328	1.1653
GradientBoosting	0.6925	0.6444	1.5194	1.6533
ExtraTrees	0.7998	0.6064	1.1587	1.6982
XGBoost	0.6558	0.5497	1.5986	1.8039
LightGBM	0.6456	0.5816	1.5918	2.0131
Ridge	**0.8807**	**0.8599**	**0.8856**	**0.9279**
Lasso	0.7122	0.5585	1.4944	1.8146

Key Finding

The regularized linear model, Ridge, performed the best, likely due to the small sample size and high dimensionality, where simpler models are less prone to overfitting.

Conclusion

- **Lag matters:** A combination of recent (t-1) and seasonal (t-12) lags provided the best results.
- **Less is more:** Feature selection was crucial to prevent overfitting in this high-dimensional, low-sample setting.
- **Simplicity wins:** A regularized linear model (Ridge) outperformed more complex tree-based models.
- **Future Work:** Explore more sophisticated feature engineering and selection techniques.

Thank You!

Questions?

- **Reference Paper:** Real-time CPI Forecasting using Machine Learning
- **GitHub Repo:** PranavKrishna6939/CPI-GA
- **Code Credits:** Pranav Krishna
- **Presentation Credits:** Aradhana Papnai, Pranav Krishna