AgroChill Forecasting System: Optimizing the "Freezer Gambit“

AgroChill's "Freezer Gambit" strategy relies entirely on accurate price forecasts to determine whether to sell fresh produce immediately or store it for future sales. This presentation outlines our comprehensive data science solution, system architecture, and business insights to optimize profitability for perishable goods in weekly markets.

Data Science Problem & Approach

- Understanding the problem
- Identifying Data
- Data pre-processing
- Exploratory Data Analysis
- Feature Engineering
- Model Selection

Understanding the problem

Predicting weekly prices 1 to 4 weeks into future

Predicting weekly fresh prices for various fruits and vegetables across multiple geographical regions (economic centres) 1 to 4 weeks into the future.

Making rolling forecasts

Predictions generated at any point in time must utilize all available historical data up to that point. As new weekly data arrives, the forecast horizon should roll forward accordingly.

Ingesting new data and automatic re-training with new data

Ingesting new data for weather and prices dynamically via a defined API and automatically retraining the predictive models periodically to adapt to new data patterns and maintain forecast accuracy over time.

Resource constraints

Operating within resource constraints and delivering predictions with low latency.

Identifying Data

Weather Data:

Contained historical records presumably associated with weekly market cycles, including:

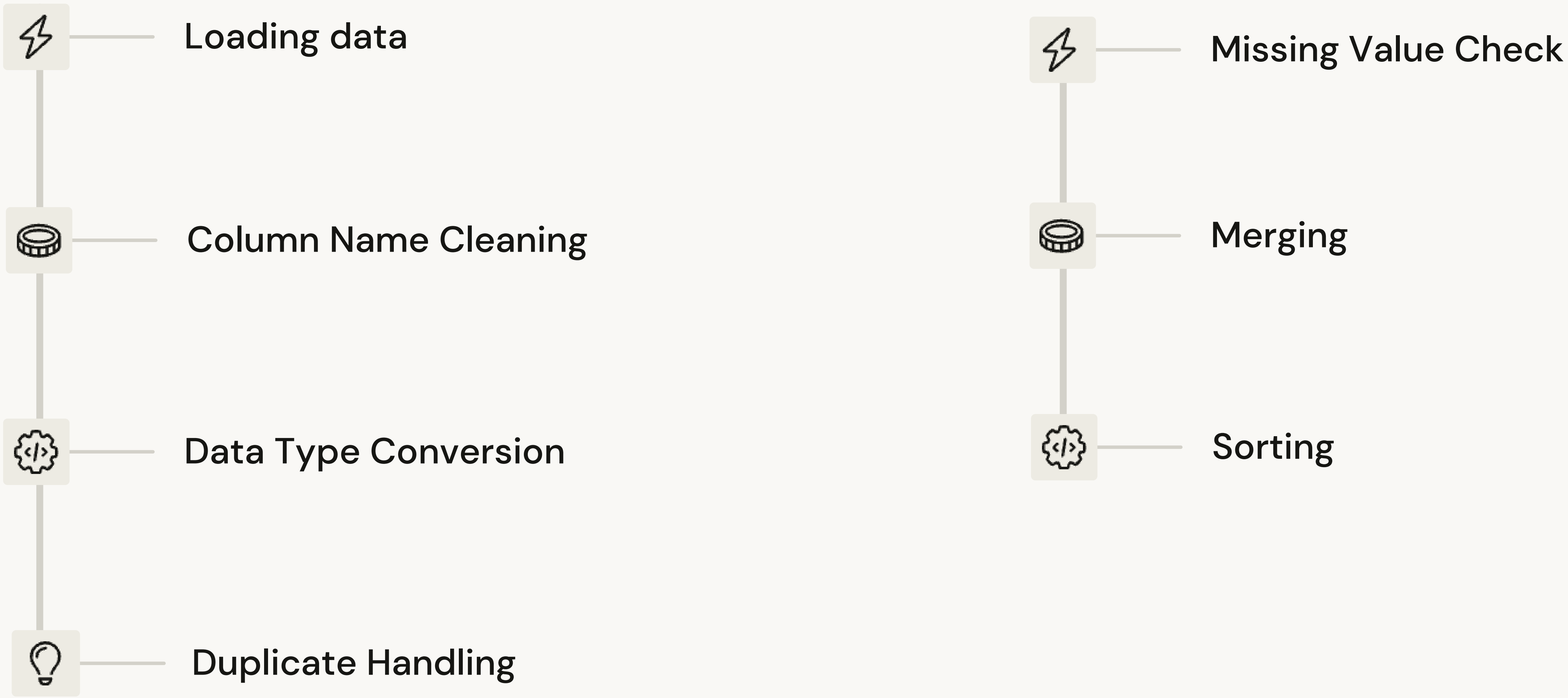
- **Date**: The date of the record.
- **Region**: The geographical economic centre.
- **Temperature** (K): Temperature in Kelvin.
- **Rainfall** (mm): Precipitation in millimetres.
- **Humidity** (%): Relative humidity percentage.
- **Crop Yield Impact Score**: A calculated metric indicating the favourability of environmental conditions for crop yield (higher score implies better conditions).

Price Data:

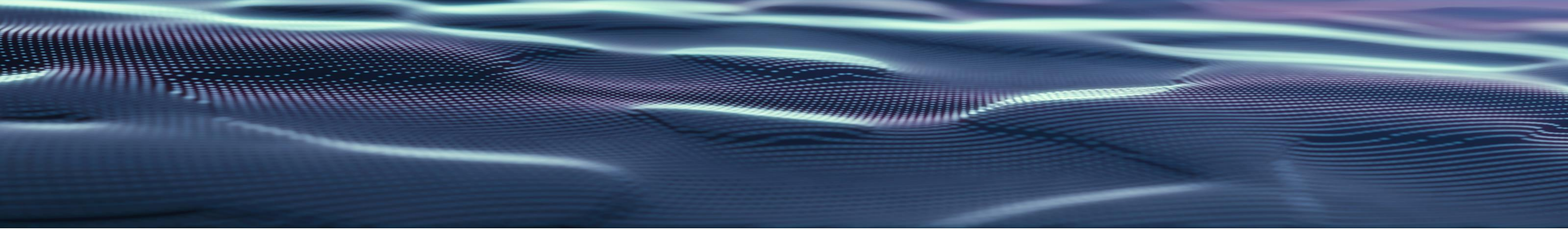
Contained historical price records, including:

- **Date**: The date corresponding to the weather record.
- **Region**: Matching the weather record region.
- **Commodity**: The specific fruit or vegetable (Mapped from 'Crop' if necessary).
- **Price per Unit** (Silver Drachma/kg): The target variable for prediction.
- **Type**: Classification as 'Fruit' or 'Vegetable'.

Data pre-processing and cleaning



- Loading data
- Column Name Cleaning
- Data Type Conversion
- Duplicate Handling
- Missing Value Check
- Merging
- Sorting

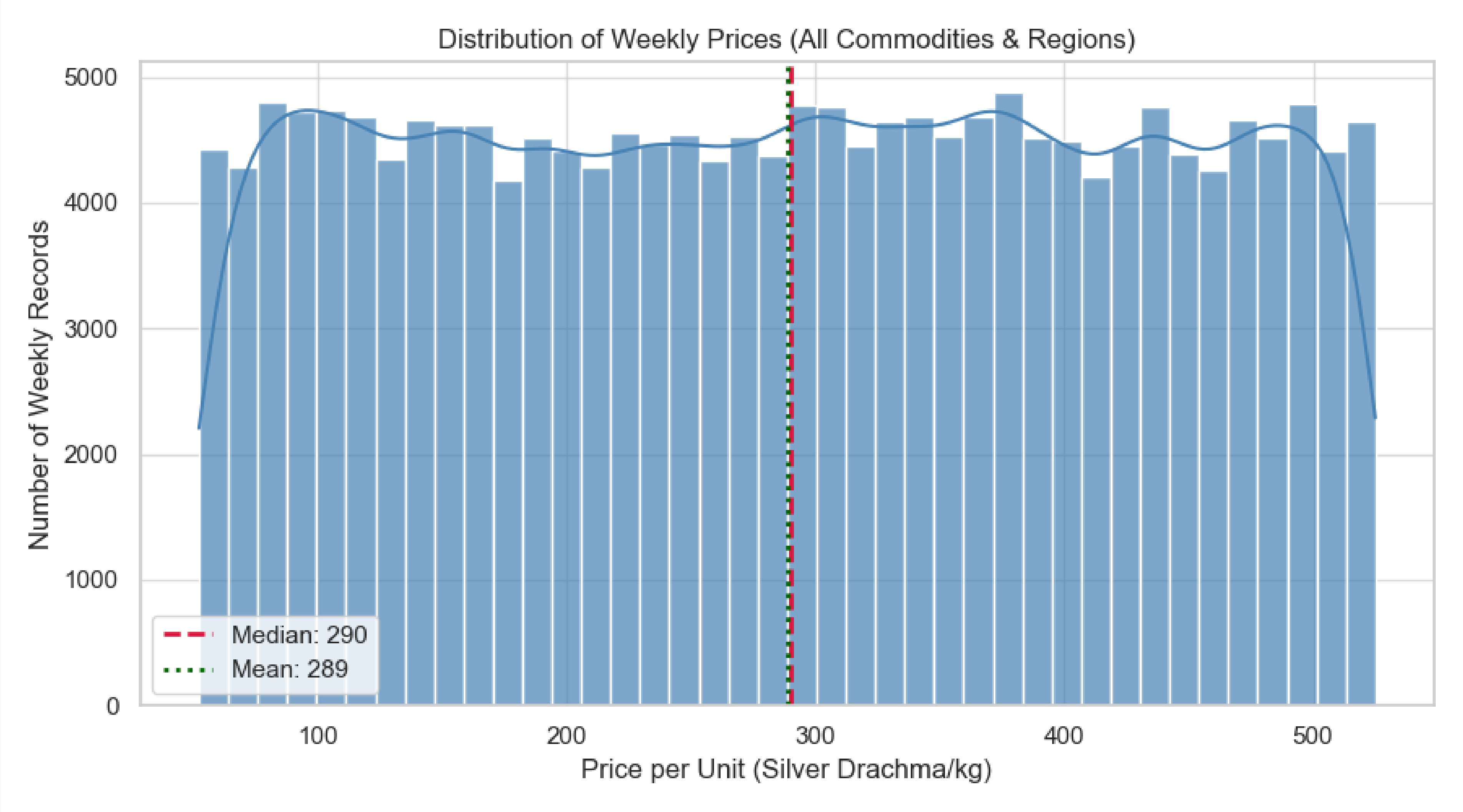


Exploratory Data Analysis(EDA)

EDA was performed on the pre-processed training dataset. Key findings include:

- **Price Characteristics**
  1. Overall Distribution
  2. Regional Variation
  3. Type Variation
- **Temporal Patterns**
  1. Individual price trends
  2. Seasonality
- **Weather Patterns and Correlation**
  1. Regional Weather Trends
  2. Correlation Analysis
- **Data Volume**
  1. Records per region

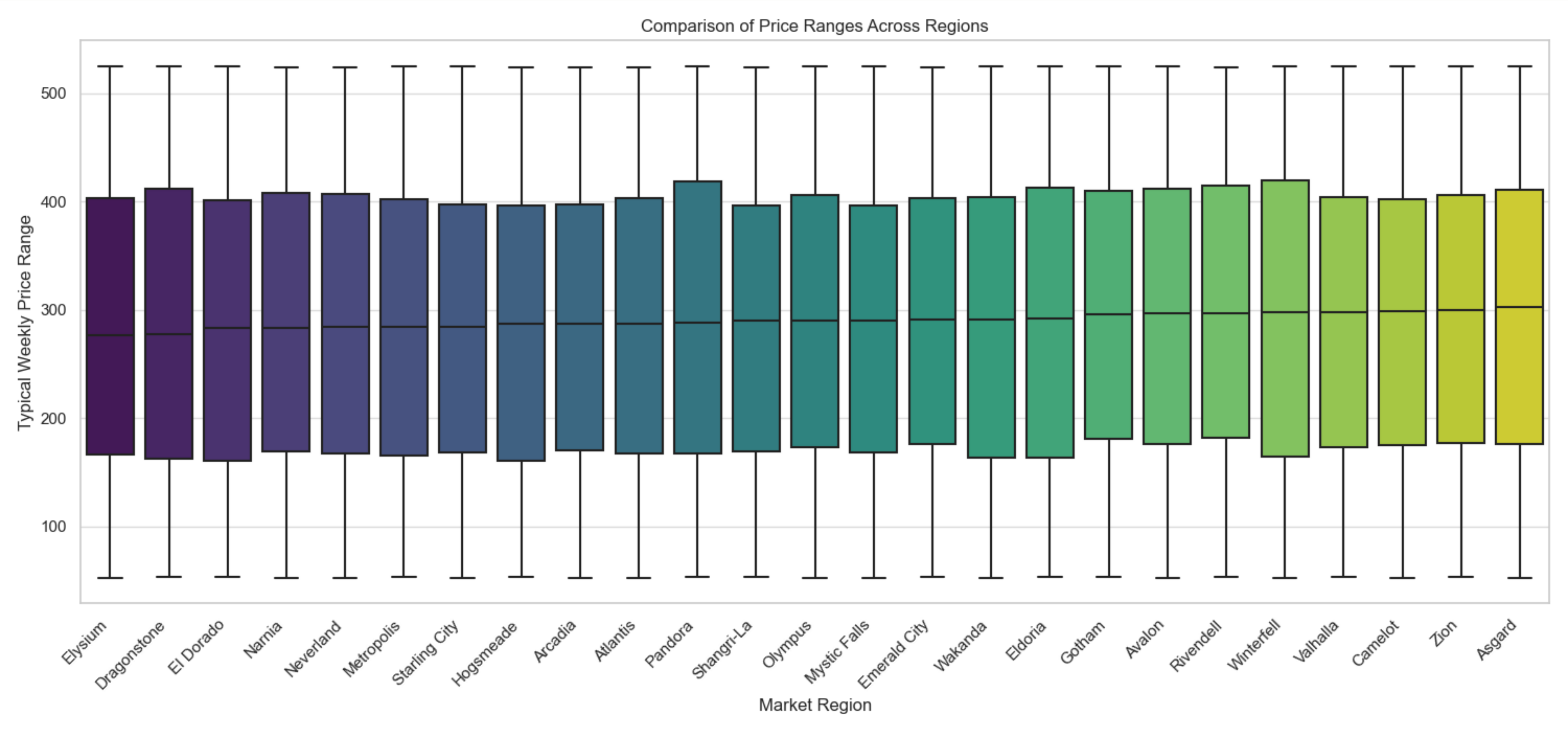
Exploratory Data Analysis(EDA) : Price Characteristics



Overall Distribution

Prices span a wide range (approx. 53 to 525 Silver Drachma/kg) with a distribution that is somewhat flat but slightly right skewed (Mean: ~289, Median: ~290). This wide spread and lack of a single strong peak suggests diverse pricing behaviors across commodities/regions and supports the use of non-linear models.

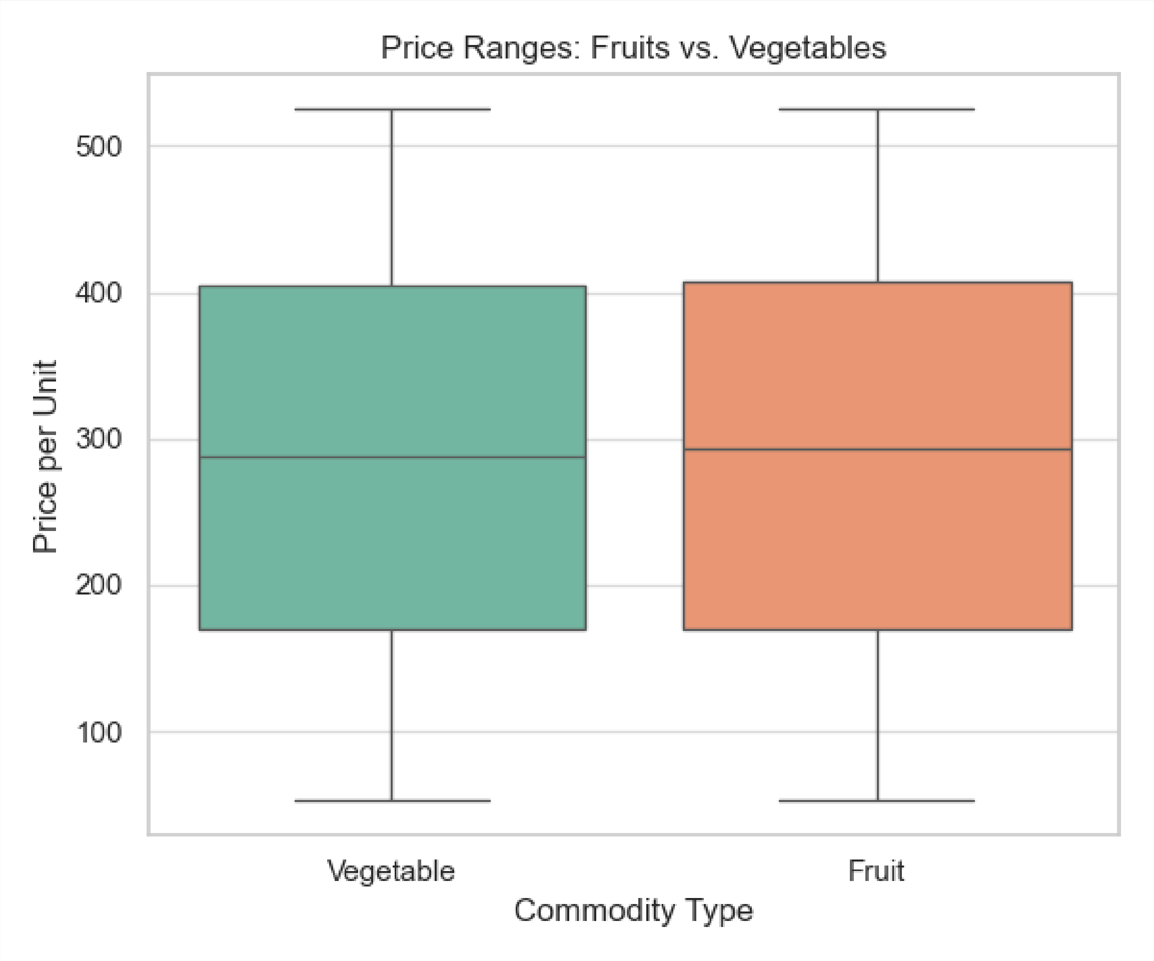
Exploratory Data Analysis(EDA) : Price Characteristics



Regional Variation

Box plots comparing regions show clear differences in median price levels (e.g., Elysium lowest at ~276, Asgard highest at ~303) and price variability (spread of the boxes). This confirms Region is a vital predictive feature.

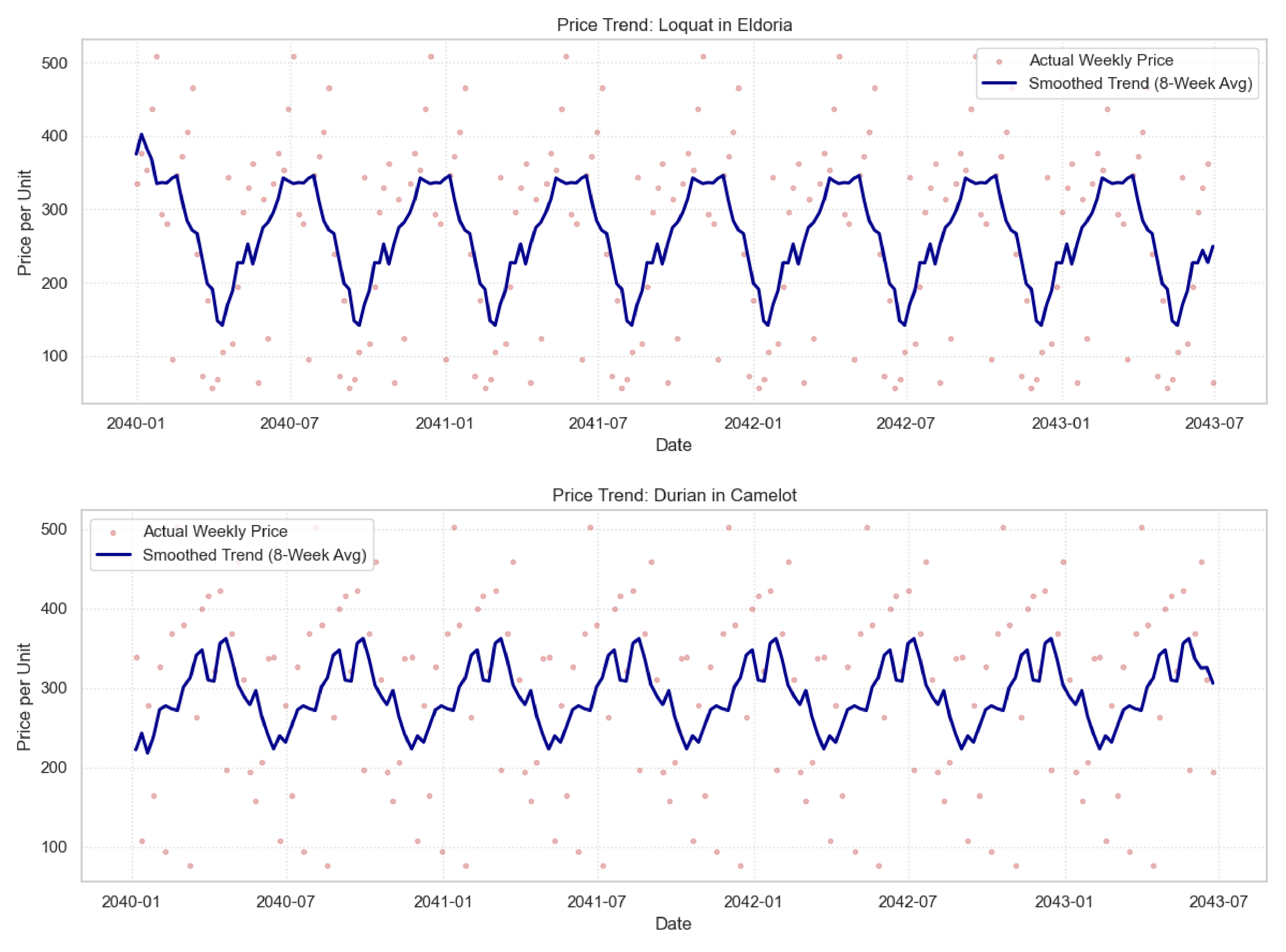
Exploratory Data Analysis(EDA) : Price Characteristics



Type Variation

The price distributions for Fruits (Median ~293) and Vegetables (Median ~288) are broadly similar in terms of central tendency and spread. This suggests that while Type might offer some information, the specific Commodity is likely a much stronger predictor.

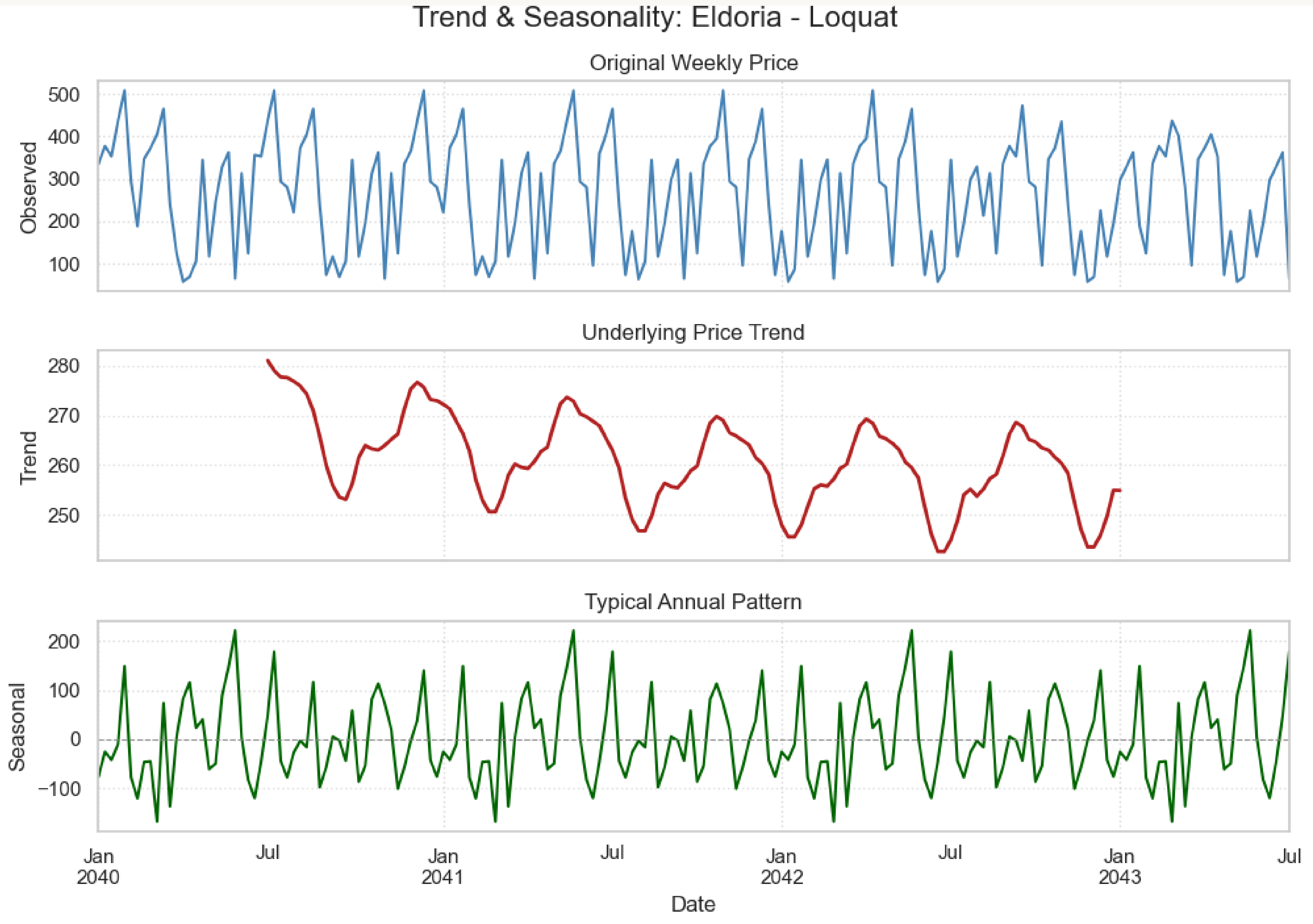
Exploratory Data Analysis(EDA) : Temporal Patterns



Individual Price Trends

Examining individual time series (e.g., Loquat in Eldoria, Durian in Camelot) reveals distinct patterns, volatility, and potential seasonality specific to each item/location. Smoothing these series with an 8 week rolling average highlights underlying trends and cyclical behavior more clearly than the noisy weekly data. This strongly justifies a time-series approach with appropriate features

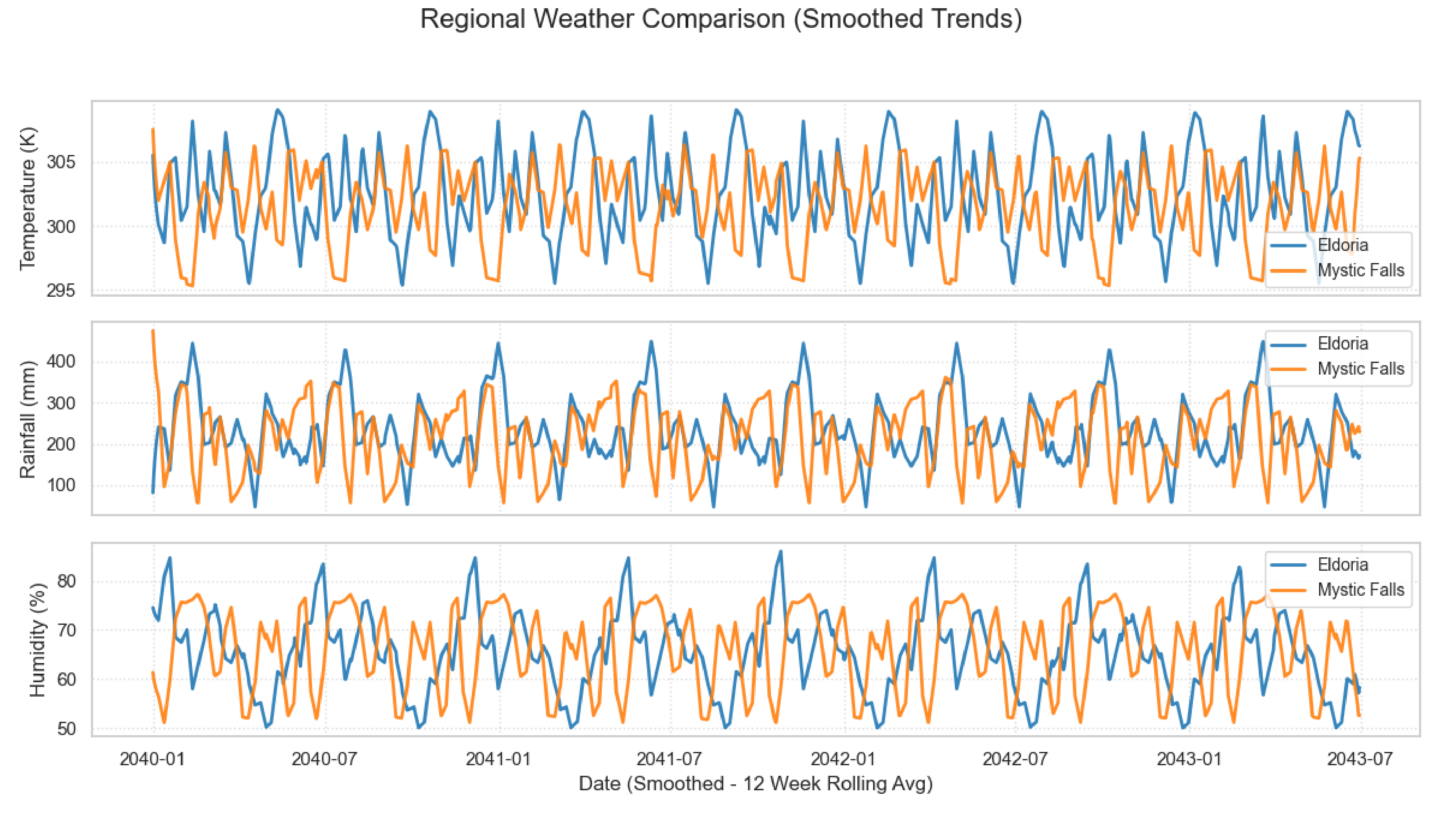
Exploratory Data Analysis(EDA) : Temporal Patterns



Seasonality

Seasonal decomposition of the 'Eldoria - Loquat' weekly price series clearly isolates a repeating annual pattern (Seasonal component) from the longer-term underlying trend. This confirms the presence of strong seasonality and necessitates features
like month or weekofyear to capture this predictable variation.

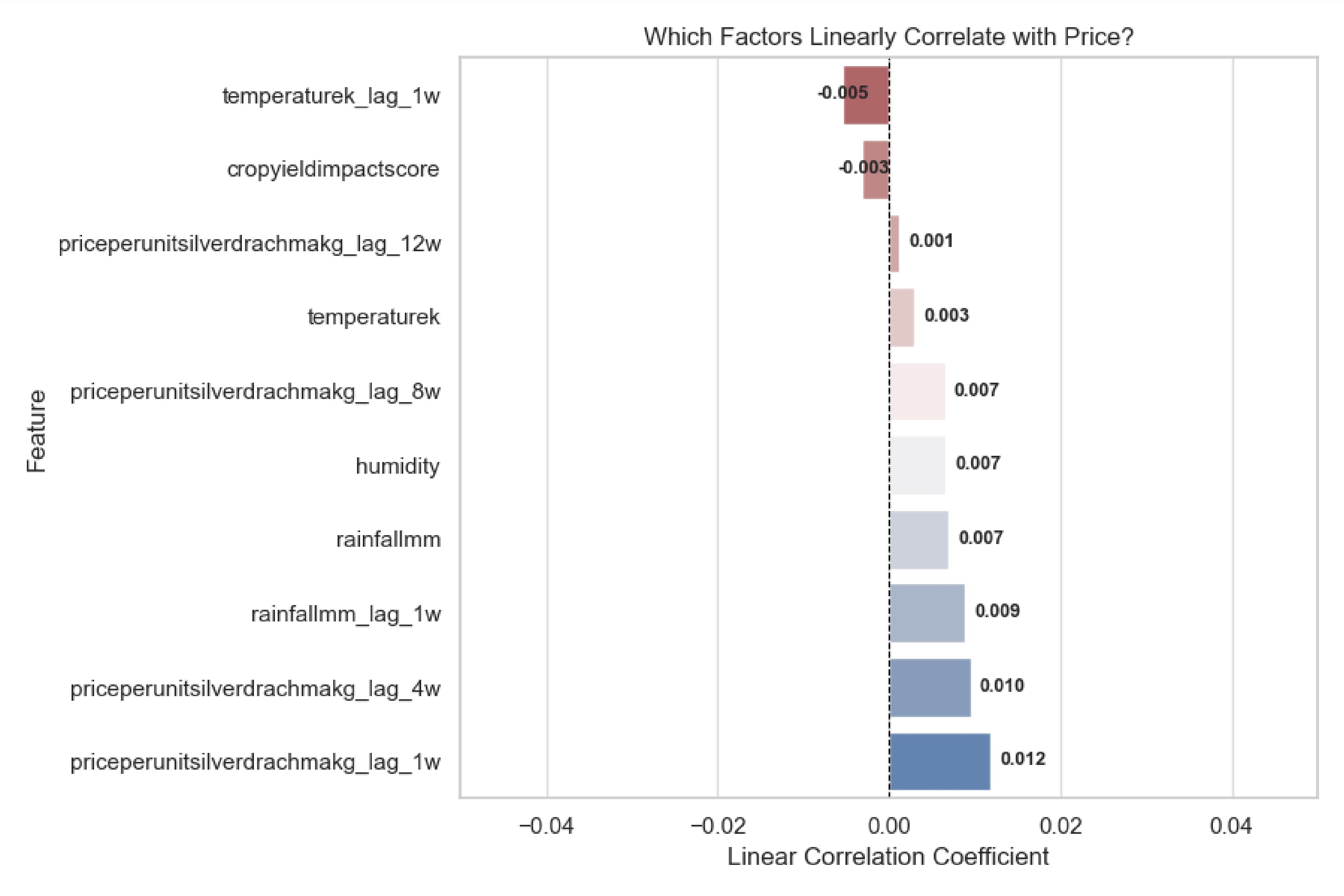
Exploratory Data Analysis(EDA) : **Weather Patterns & Correlation**



Regional Weather Trends

Comparing smoothed (12-week rolling average) weather patterns for representative regions like Eldoria and Mystic Falls shows distinct regional climates and clear seasonality, especially in temperature. Rainfall and humidity patterns also differ between regions. This provides context for why region is important and suggests weather features might interact with region.

Exploratory Data Analysis(EDA) : **Weather Patterns & Correlation**

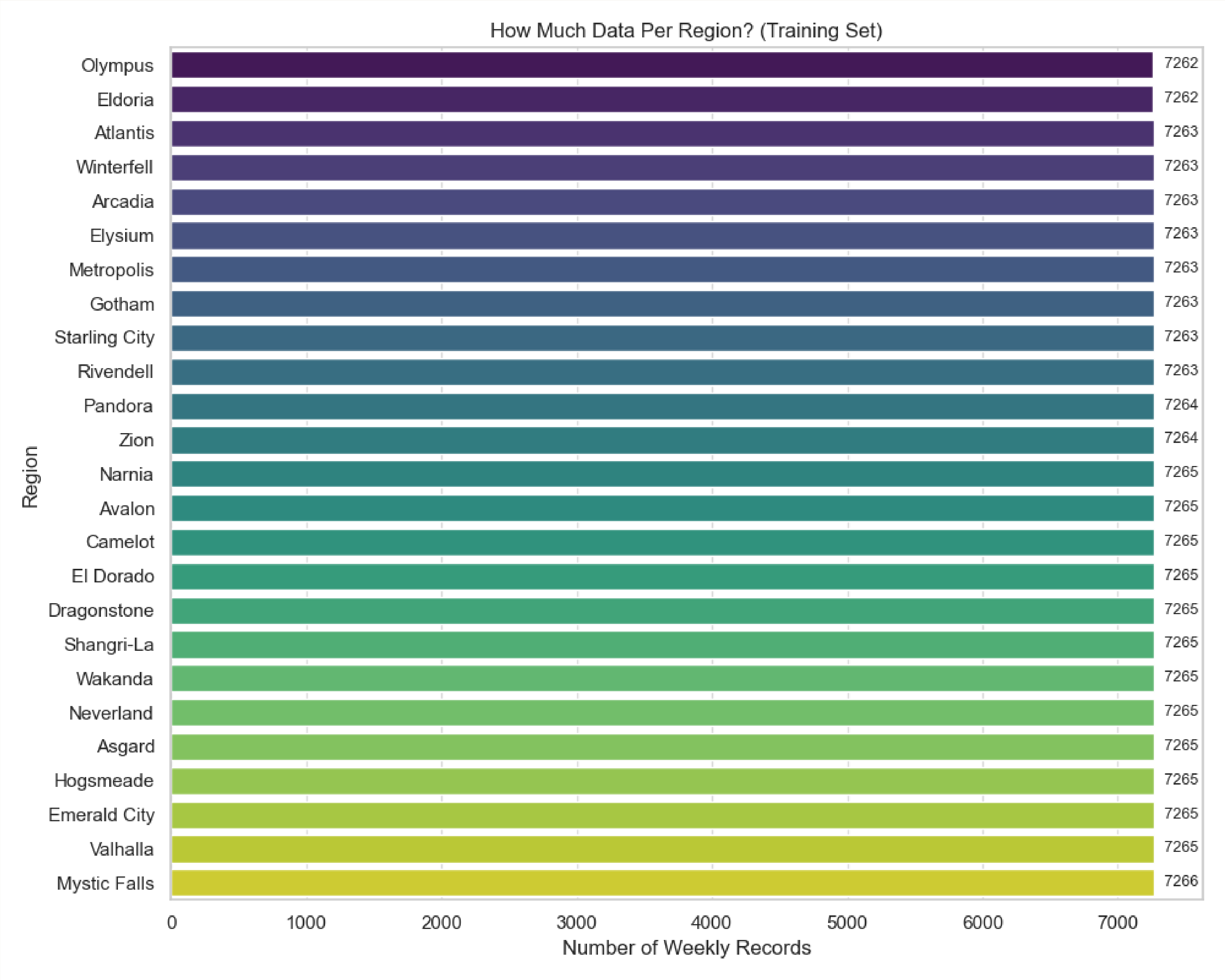


Correlation Analysis

A bar chart showing linear correlations with the current price reveals:

- The strongest positive correlations are with recent price lags (price_lag_1w, price_lag_4w), confirming strong autocorrelation.
- Correlations with weather variables (current and lagged temperaturek, rainfallmm, humidity) and cropyieldimpactscore are very weak (close to zero).

Exploratory Data Analysis(EDA) : Data Volume



Records per Region

The training data is remarkably well-balanced across regions, with each having a very similar number of records (approx. 72627266). This reduces concerns about model bias towards heavily represented regions.

Feature Engineering Strategy : Design Overview

Lag Features

Created for price and key weather indicators (1, 4, 8, 12 weeks prior) within each Region-Commodity group to capture autocorrelation and prevent data leakage.

Rolling Window Features

Calculated mean and standard deviation for price/weather variables over 4, 8, 12-week windows to capture trends and volatility.

Time-Based Features

Extracted year, month, week of year, day of week from the date to capture seasonality patterns essential for annual cycles.

Categorical Features

Region, Commodity, and Type were handled using One-Hot Encoding, with Region and Commodity proving to be important predictive features.

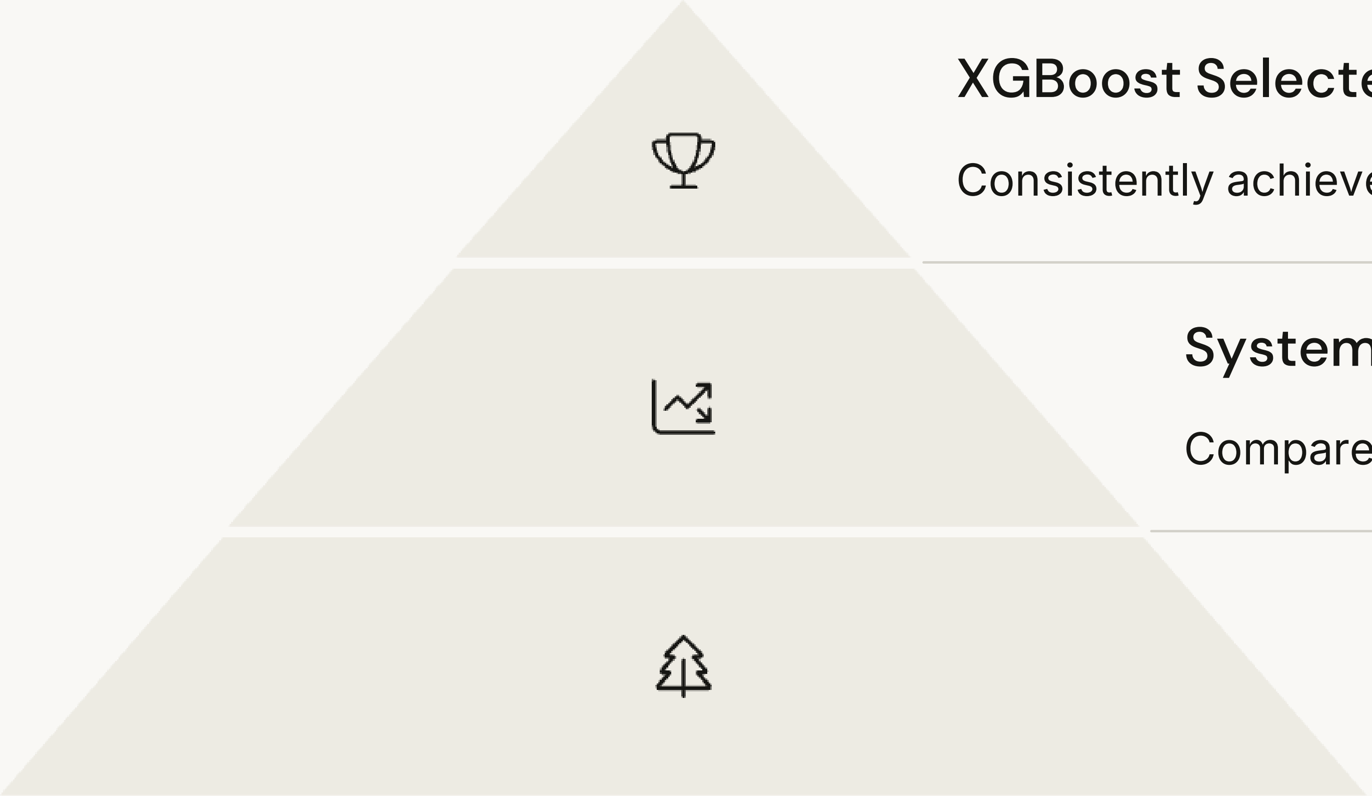
Model Selection & Performance

Given the time series nature, the need to capture non-linearities, and the strict resource constraints, Gradient Boosted Decision Trees (GBDTs) were identified as the most promising class of models. We systematically evaluated three leading GBDT Implementations. The performance on the evaluation set was the primary criterion for final model selection.

Horizon	LightGBM (Eval RMSE)	XGBoost (Eval RMSE)	CatBoost (Eval RMSE)	Selected Model (Eval RMSE)
1w	136.72	106.56	145.51	XGBoost
2w	126.87	120.65	151.22	XGBoost
3w	151.03	123.77	168.87	XGBoost
4w	161.95	139.79	162.44	XGBoost

XGBoost models have consistently achieved significantly lower RMSE across all four prediction horizons. XGBoost demonstrated superior generalization performance for this specific dataset and feature set.

Model Selection & Performance



XGBoost Selected

Consistently achieved lowest RMSE across all horizons

Systematic Evaluation

Compared LightGBM, XGBoost, and CatBoost

Gradient Boosted Decision Trees

Best for capturing non-linear relationships

Data Pipeline Strategy

Data Ingestion

API endpoints receive new data

Data Persistence

Received data is stored persistently.

Triggering

Retraining is initiated automatically on a schedule or manually via an API call.

Aggregation

The retraining job loads both original training data and all persisted incoming data

Processing & Feature Engineering

The aggregated dataset undergoes cleaning, merging, sorting, and the full feature engineering process

Tuning & Training

Hyperparameters are re-tuned (using Optuna), and new models(XGBoost) are trained on the latest aggregated, featured data for each forecast horizon

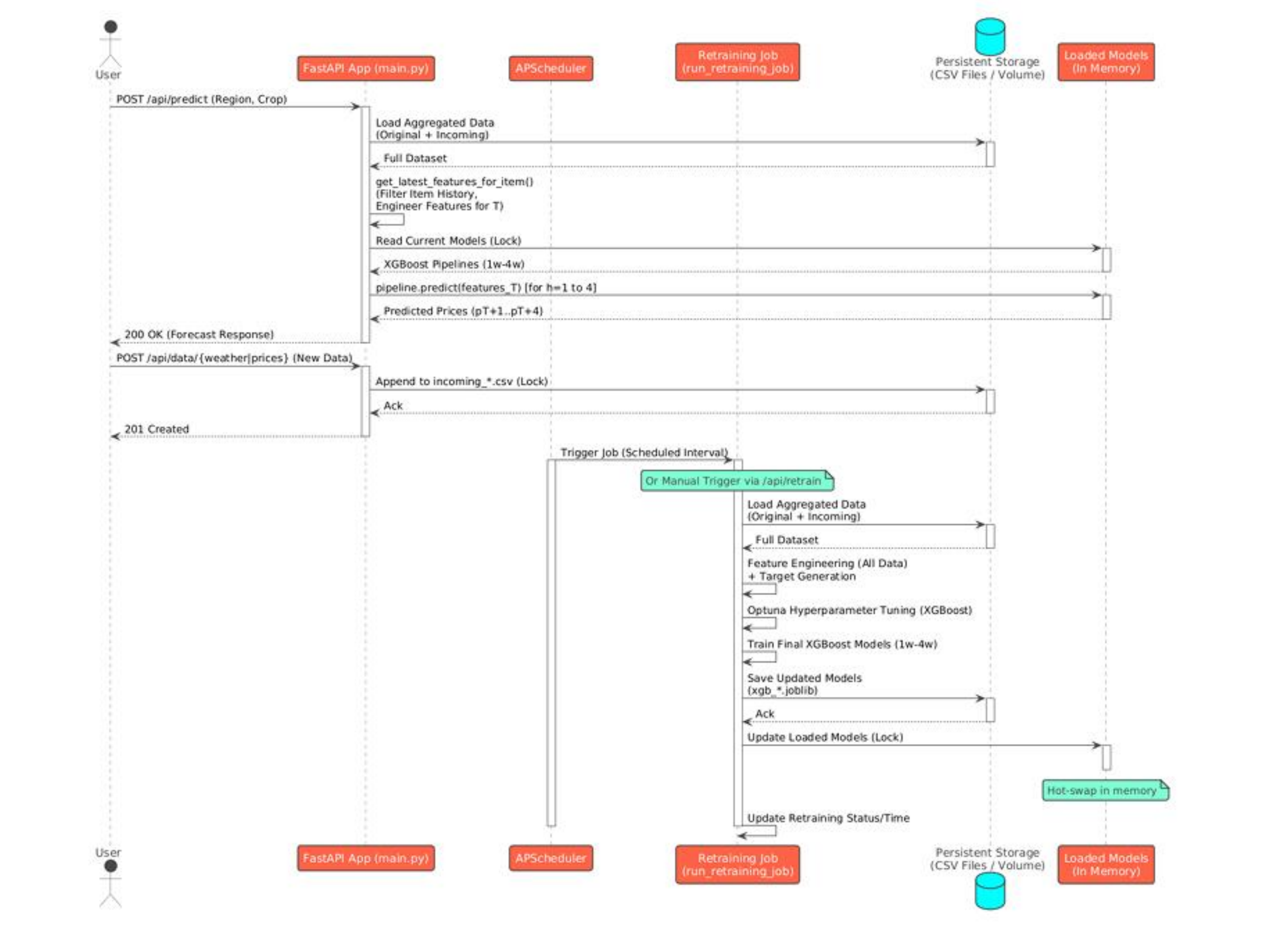
Model Deployment (Hot Swap)

The newly trained models replace the existing models currently loaded in the running API's memory

Prediction

The prediction endpoint dynamically uses the aggregated data to generate features for the requested item's latest time point and applies the currently loaded models

Data Pipeline Strategy



User

FastAPI App (main.py)

APScheduler

Retraining Job
(run_retraining_job)

Persistent Storage
(CSV Files / Volume)

Loaded Models
(In Memory)

POST /api/predict (Region, Crop)

Load Aggregated Data
(Original + Incoming)

Full Dataset

get_latest_features_for_item()
(Filter Item History,
Engineer Features for T)

Read Current Models (Lock)

XGBoost Pipelines (1w-4w)

pipeline.predict(features_T) [for h=1 to 4]

Predicted Prices (pT+1..pT+4)

200 OK (Forecast Response)

POST /api/data/{weather|prices} (New Data)

Append to incoming_*.csv (Lock)

Ack

201 Created

Trigger Job (Scheduled Interval)

Or Manual Trigger via /api/retrain

Load Aggregated Data
(Original + Incoming)

Full Dataset

Feature Engineering (All Data)
+ Target Generation

Optuna Hyperparameter Tuning (XGBoost)

Train Final XGBoost Models (1w-4w)

Save Updated Models
(xgb_*.joblib)

Ack

Update Loaded Models (Lock)

Hot-swap in memory

Update Retraining Status/Time

User

FastAPI App (main.py)

APScheduler

Retraining Job
(run_retraining_job)

Persistent Storage
(CSV Files / Volume)

Loaded Models
(In Memory)

System Architecture

FastAPI Web Service

RESTful API interface running on Uvicorn ASGI server

XGBoost Models Pipeline

Four models for different forecast horizons

Automated Retraining

APScheduler background job for model updates

Our containerized web service provides key endpoints for predictions (/api/predict), data ingestion (/api/data/weather, /api/data/prices), system status (/api/status), and manual retraining (/api/retrain). The system dynamically loads historical data, generates features, and applies the appropriate models to deliver accurate forecasts.

For data persistence, we use simple CSV files with concurrency control via python-filelock, while scikit-learn Pipelines ensure consistent transformations between training and prediction.

Business Insights from Data Analysis

Regional Price Variation

Prices and volatility differ noticeably across regions, meaning the "best market" is not fixed. This highlights the importance of targeting specific regions based on predicted price advantages.

Price Autocorrelation

Recent prices are strong indicators of future prices. Tracking recent trends is crucial, and the model heavily relies on these lagged features, though sudden market shocks remain harder to forecast.

Seasonal Patterns

Both prices and weather show clear annual cycles. Pricing strategies should anticipate seasonal highs and lows, which the model captures through time-based features.

Weather-Price Relationship

Current weather has a weak linear correlation with current price. Weather's influence is likely more complex (non-linear, lagged supply effects), meaning direct weather reports aren't immediate price predictors.

Thank You

from

Team Fouette Bytes

Wansajee Illukkumbura(leader)

Sithum Konthasinghe

Sesandi Samarakoon