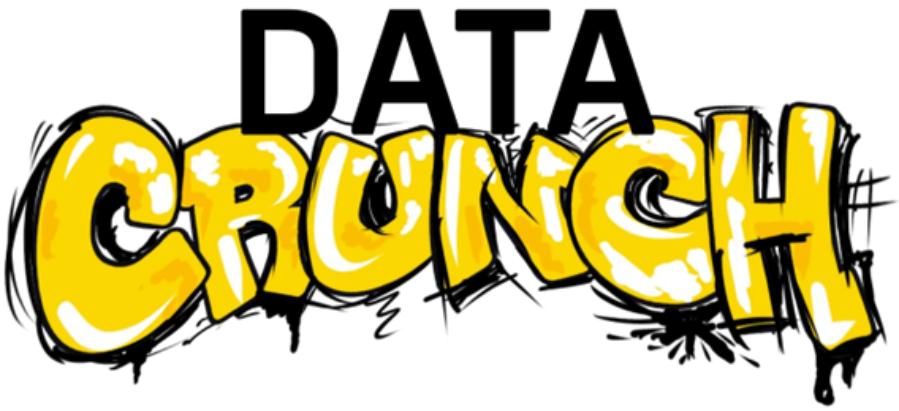


Legacy of the Market King: The Freezer Gambit

By Helloworld 2.0 - Data_Crunch_045
University of Moratuwa

1. Introduction

For the **AgroChill** forecasting system, we have designed a robust pipeline that forecasts weekly fresh produce prices one month ahead for various regions and commodities in Agrovia. Our solution integrates historical weather and price data, scales and preprocesses the data, applies time-series modeling techniques, and continuously updates the predictions in real-time. This allows AgroChill to decide optimally when to sell fresh or freeze produce, ultimately maximizing profitability while minimizing waste.

2. System Architecture

The **AgroChill** system is built on a modular architecture, consisting of the following key components:

- **Data Ingestion**: Data from multiple sources, including price and weather datasets, is ingested into the system. We use **Pandas** to load the CSV data and process it into a structured format.

- **Data Processing Pipeline**: The data undergoes cleaning, merging, feature engineering, and reindexing to ensure consistency and completeness. The cleaned dataset is used as input for our forecasting models.

- **Forecasting Engine**: The core of the system, where we use a **Linear Regression** model implemented with the **Darts** library for time-series forecasting. We've integrated **Prophet**-like functionality with **Darts' RegressionModel** for predicting future prices.

- **API Layer**: A set of REST APIs is exposed to interact with the model, allowing users to retrieve predictions, submit updated data, and access forecasting results.

- **Containerization**: The entire system is packaged in a **Docker** container, ensuring portability and scalability. This guarantees easy deployment across environments.

3. Data Pipeline

Our data pipeline ensures that the incoming data is clean, consistent, and ready for use in our forecasting model.

1. **Data Loading & Parsing**:
    - We load the **weather** and **price** datasets using `pandas.read_csv()`. The date columns in both datasets are converted into datetime format to ensure proper alignment.
2. **Data Aggregation**:
    - We aggregate the weather data by calculating the mean values for each `Region` and `Date`, ensuring that we don't have redundant entries.
    - The **price data** is merged with the aggregated **weather data** based on the `Region` and `Date` columns.
3. **Data Cleaning & Deduplication**:
    - Missing values are handled by dropping rows with `NaN` values using `.dropna()`.
    - Duplicate entries (based on `Region`, `Date`, and `Commodity`) are removed to avoid data redundancy using `.drop_duplicates()`.
4. **Weekly Grouping**:
    - We create a new column `RoundedDate` which rounds each date to the start of the week. This ensures that the time series is continuous on a weekly basis.
    - Data is aggregated by `Region`, `Commodity`, and `RoundedDate` to compute the weekly mean of each feature (e.g., Price, Temperature, Rainfall).
5. **Ensuring Weekly Continuity**:
    - Using the `ensure_weekly_continuity()` function, we reindex the dataset to ensure that all weeks are represented. If any week is missing, it's added with `NaN` values.

4. Forecasting Methodology

We leverage **Darts' RegressionModel** with **Linear Regression** to predict the price of commodities. Below is our step-by-step approach:

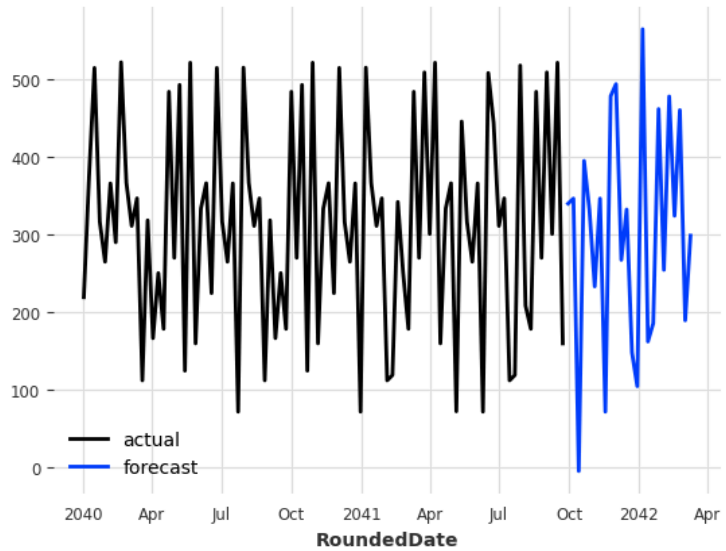
1. **Handling Missing Weeks**:
   - We ensure that all weeks in the dataset are represented by reindexing each commodity's data using `pd.date_range()` to fill any gaps in the weekly data.

2. **Feature Scaling**:
   - The data, including the price, temperature, and rainfall, is scaled using **MinMaxScaler** to standardize the values between 0 and 1. This improves the stability and performance of the regression model.

3. **TimeSeries Conversion**:
   - We convert the cleaned and scaled data into a **Darts TimeSeries** object, which is used as the input for the regression model. The target variable for our model is the `Price`.

4. **Model Training**:
   - A **Linear Regression** model is fitted to the TimeSeries data using Darts' `RegressionModel`. This model learns the relationship between historical price data and weather features, predicting the price for the next week based on past values.

5. **Rolling Forecast**:
   - We predict the price for the next 4 weeks by providing the model with the current data up until the last known week. The model generates forecasts for the next four weeks, ensuring that we have rolling predictions.
   - The predictions are plotted alongside the actual data to visually compare the model's accuracy.



5. Model Evaluation

We evaluate the model's performance using the **Root Mean Squared Error (RMSE)**, which measures the difference between predicted and actual prices. Here's how we evaluate:

1. **Prediction Evaluation**:
    - The initial forecast is compared with the actual data, and RMSE is computed to measure the model's accuracy.
    - We perform an **inverse transformation** on the scaled predictions to revert them to their original scale (Silver Drachma/kg).
2. **Extended Predictions**:
    - We use the forecasted values from the model to extend the predictions for the next few weeks, taking into account new incoming data.
    - The **RMSE** is re-calculated for the extended predictions to ensure that the forecasts maintain accuracy.

6. Business Insights & Recommendations

Based on the forecasting results, we derive actionable insights that help AgroChill optimize its operations:

1. **Market Trends & Price Prediction**:
    - We identify periods when prices are expected to rise or fall, enabling AgroChill to decide whether to store or sell produce at optimal times.
2. **Storage & Freezing Strategy**:
    - When prices are predicted to fall in the coming weeks, the system recommends freezing produce to avoid losses, helping AgroChill manage its storage effectively.
3. **Weather Impact on Pricing**:
    - Our system reveals the impact of weather conditions (e.g., temperature, rainfall) on crop prices, allowing AgroChill to prepare for fluctuations due to environmental factors.

7. Deployment Strategy

Our solution is packaged into a Docker container, ensuring it is portable and scalable. The following components are deployed:

- **Model**: The time-series model is included in the Docker container, ensuring it is always up-to-date with new data.
- **APIs**: Exposed APIs allow external systems to retrieve forecasted prices, set new weather data, and submit price data.
- **Cloud Deployment** (Optional): For bonus points, the solution can be deployed on a cloud platform like AWS or GCP for scalability and public access.

8. Challenges & Future Improvements

- **Data Completeness**: Missing weather data was a challenge, but we handled this with imputation techniques. In the future, we can use more sophisticated models for imputation.
- **Model Improvement**: While **Linear Regression** performed adequately, integrating more complex models like **LSTMs** might yield better results in capturing non-linear relationships.

9. Conclusion

The **AgroChill** forecasting system provides a highly effective solution for predicting agricultural commodity prices. Through the integration of weather data, time-series forecasting models, and real-time data processing, the system enables AgroChill to make better decisions on when to sell or store produce, ensuring maximum profitability and reducing waste.