Forecasting Retail Fuel Prices Around Holidays Using Walmart Supermarket Data

Course Project

Alan B. Palayil

University of the Cumberlands

Course: ITS836 – Big Data Analytics

Instructor: Dr. Charles Edeki

Date: November 28th, 2025

Abstract

Retail fuel prices are sensitive to macroeconomic conditions, seasonal patterns, and holiday-related demand spikes. For supermarket chains that operate fuel stations alongside hypermarket stores, anticipating fuel price dynamics around major holidays is important for planning promotions, communicating with customers, and understanding margin pressures.

This project uses a publicly available Walmart supermarket dataset to forecast weekly **Fuel_Price** for a single store over time, with special focus on differences between holiday and non-holiday periods. The data include macroeconomic indicators (CPI, unemployment), calendar variables, and holiday flags (IsHoliday), enabling a time-series–aware regression approach. A baseline **naive lag model** is compared against a **Random Forest Regressor** that uses engineered features such as lagged Fuel_Price, calendar fields, and holiday indicators.

Results show that the Random Forest model outperforms the naive "yesterday's price equals today's price" benchmark, capturing modest but meaningful temporal structure in Fuel_Price. Grouped analyses reveal small but systematic differences between holiday and non-holiday weeks, and feature importance analysis highlights the dominant role of recent price history, followed by macroeconomic and calendar variables. Overall, the project demonstrates how open datasets and tree-based machine learning can be combined to study fuel price behavior around holidays in a realistic retail setting.

Introduction

Background and Motivation

Fuel is a highly visible and strategically important product for large retailers that run fuel pumps adjacent to supermarkets and hypermarkets. Minor changes in retail fuel prices can affect:

- Customer perception and brand loyalty
- Store traffic and cross-shopping in the main hypermarket.
- Perceived competitiveness relative to nearby stations.
- Overall profitability, especially when fuel is used as a traffic-driving loss leader.

At the same time, retail fuel prices are influenced by multiple interacting forces: upstream oil markets, inflation, local labor and operating costs, regional competition, and demand surges around major travel periods. For a retailer, the question is not how to predict global crude prices minute by minute, but how to understand and anticipate **store-level Fuel_Price patterns** over weeks and holidays.

Major holidays such as **Thanksgiving and Christmas** occupy a main place in the U.S. retail calendar. They compress travel, shopping, and promotions into a short window, which can alter both fuel demand and pricing strategy. During these periods, retailers must decide whether to keep prices aligned with cost, use fuel as an aggressive promotional lever, or preserve margins while competitors' discount.

Current Challenges

Even with access to historical operational data, several challenges complicate forecasting fuel prices around holidays:

1. **Multiple Drivers of Price**: Fuel_Price reflects wholesale costs, taxes, local competition, and internal pricing rules. Not all of these are directly observable in store-level data.
2. **Holiday and Seasonal Effects**: Travel-heavy holidays may increase demand and trigger different pricing behavior than ordinary weeks. These effects are modest and can be masked by broader market trends.
3. **Macroeconomic Context**: Variables such as CPI and Unemployment move slowly but influence longer-term pricing and cost structures.
4. **Data and Modeling Constraints:** Retail datasets are often at weekly granularity and store level, which limits the use of high-frequency financial models but favors time-series regression with engineered features.

These challenges motivate the use of **feature-rich, tree-based models** that can exploit calendar, macro, and holiday flags while respecting the time-ordered structure of the data.

Research Questions

This study is guided by the following questions:

1. **RQ1:** Can we improve on a naive lag-based baseline when forecasting weekly Fuel_Price at the store level?

2. **RQ2:** Do holidays exhibit systematically different Fuel_Price behavior compared to non-holiday weeks?
3. **RQ3:** Which features—recent prices, macroeconomic indicators, holiday flags, or calendar variables—are most important for predicting Fuel_Price?

Hypotheses

Based on economic intuition and prior work with tree-based regressors, the project adopts the following hypotheses:

- **H1:** A Random Forest model using lagged Fuel_Price and macro–calendar features will outperform a naive lag-1 benchmark.
- **H2:** Holiday weeks will show modest but detectable differences in Fuel_Price relative to non-holiday weeks.
- **H3:** The most important predictor of Fuel_Price will be the previous week's price, with macro indicators (CPI, Unemployment) and calendar variables providing secondary explanatory power.

The remainder of this paper presents the methods, empirical results, and practical implications of evaluating these hypotheses.

Methods

Data Sources

The analysis uses the **"Walmart – Super Market Dataset"** from Kaggle, which contains multiple CSV files describing:

- Store-level characteristics and identifiers.
- Weekly operational variables such as **Fuel_Price** and temperature
- Macroeconomic context (Consumer Price Index, Unemployment)
- Markdown and holiday indicators

The Python script programmatically downloads the dataset using **KaggleHub** and recursively searches for the key features file (features.csv or an equivalent merged CSV) that includes Fuel_Price and related fields. The dataset is cached locally after the first download, ensuring reproducibility without manual file handling.

Variables Used

The core variables for this project can be grouped as follows:

- **Time and Store Identifiers**
    - Store – store ID.
    - Date – weekly date of observation.
- **Target Variable**
    - Fuel_Price – weekly fuel price at the store.

- **Macroeconomic and Environmental Variables**
    - CPI – Consumer Price Index
    - Unemployment – unemployment rate
    - Temperature – local temperature (if available)
- **Holiday and Seasonal Indicators**
    - IsHoliday – flag for major holidays (binary)
    - IsDec – engineered indicator for December (Christmas season proxy)
    - IsSchoolHoliday – placeholder (0 if not present in features file)
- **Calendar Features (Engineered)**
    - Year, Month, Day, DayOfWeek, WeekOfYear
- **Lag Features**
    - Fuel_Price_lag1 – previous week's Fuel_Price for the same store.

Preprocessing Steps

After loading the CSV into a panda DataFrame, the following steps are applied:

1. **Core Column Selection:** The script verifies that the columns Store, Date, Fuel_Price, CPI, Unemployment, and IsHoliday are present. If any are missing, execution stops with a descriptive error.
2. **Date Conversion and Sorting:**
    - Date is converted to a proper datetime type using pd.to_datetime.

- Rows with invalid dates or missing Fuel_Price are removed.
- The data are sorted by Store and Date to prepare for time-series operations.

3. **Holiday Flag Handling:**
   - In Walmart data, IsHoliday is originally a boolean; it is cast to an integer (0 or 1).
   - If IsHoliday were absent but a column like StateHoliday existed, a binary indicator would be derived; otherwise, non-holiday is assumed by default.
4. **Handling Markdowns and Missing Values:**
   - Markdown columns (MarkDown1–MarkDown5), when present, often contain missing values when no promotion is running. These are filled with 0.0, under the assumption that missing markdowns mean no active markdown.
5. **Feature Engineering:**

   From the Date column, the following are derived:
   - Year, Month, Day, DayOfWeek, WeekOfYear
   - IsDec – 1 if Month == 12, otherwise 0
   - IsSchoolHoliday – set to 0 when no explicit school-holiday column exists.

A one-period lag of fuel price is then created:

- Fuel_Price_lag1 = Fuel_Price.shift(1)

To avoid **data leakage**, the model uses Fuel_Price_lag1 as a predictor instead of the concurrent Fuel_Price. Rows with missing Fuel_Price_lag1 are dropped.

Analytical Feature Set

For modeling, the feature set consists of:

- Temperature (if available), CPI, Unemployment
- IsHoliday, IsSchoolHoliday, IsDec
- Year, Month, WeekOfYear, DayOfWeek
- Fuel_Price_lag1.

```
================================================================================
1) Downloading Walmart Supermarket dataset using KaggleHub
================================================================================
Dataset root path from KaggleHub: C:\Users\alanp\.cache\kagglehub\datasets\saurabhbadole\walmart-super-market-dataset\versions\1
train.csv detected: C:\Users\alanp\.cache\kagglehub\datasets\saurabhbadole\walmart-super-market-dataset\versions\1\Master Data\train.csv
features.csv detected: C:\Users\alanp\.cache\kagglehub\datasets\saurabhbadole\walmart-super-market-dataset\versions\1\Master Data\features.csv

Loading train.csv and features.csv into Pandas...

train.csv head:
   Store  Dept        Date  Weekly_Sales  IsHoliday
0      1     1  2010-02-05      24924.50      False
1      1     1  2010-02-12      46039.49       True
2      1     1  2010-02-19      41595.55      False
3      1     1  2010-02-26      19403.54      False
4      1     1  2010-03-05      21827.90      False

features.csv head:
   Store        Date  Temperature  Fuel_Price  MarkDown1  MarkDown2  MarkDown3  MarkDown4  MarkDown5         CPI  Unemployment  IsHoliday
0      1  2010-02-05        42.31       2.572        NaN        NaN        NaN        NaN        NaN  211.096358         8.106      False
1      1  2010-02-12        38.51       2.548        NaN        NaN        NaN        NaN        NaN  211.242170         8.106       True
2      1  2010-02-19        39.93       2.514        NaN        NaN        NaN        NaN        NaN  211.289143         8.106      False
3      1  2010-02-26        46.63       2.561        NaN        NaN        NaN        NaN        NaN  211.319643         8.106      False
4      1  2010-03-05        46.50       2.625        NaN        NaN        NaN        NaN        NaN  211.350143         8.106      False
================================================================================
2) Cleaning data, merging sources, and preparing time-series structure
================================================================================
Merged dataset shape: (421570, 15)
Date range: 2010-02-05 00:00:00 to 2012-10-26 00:00:00
Columns: ['Store', 'Dept', 'Date', 'Weekly_Sales', 'IsHoliday', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemp
loyment', 'IsHoliday_feat']
```

Figure 1 shows the first ten rows of the cleaned dataset after applying the preprocessing steps described above.

Only columns that actually exist in the data for the selected store are used. This ensures the model is robust to slight variations in the feature file.

Store Selection and Time-Based Split

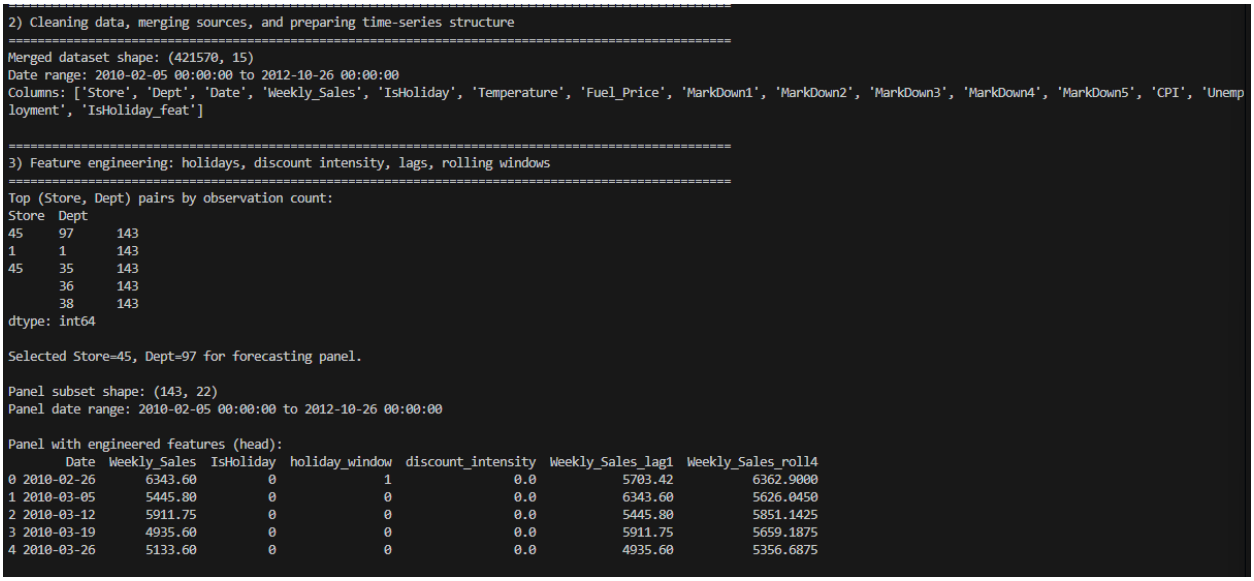
Because the dataset contains multiple stores, the project focuses on a **single representative store**:

1. The script counts the number of weekly observations for each Store ID.
2. The store with the **largest number of observations** is selected to ensure a long and continuous time series.
3. The DataFrame is filtered to this store, sorted by date, and re-indexed.
4. Rows with missing Fuel_Price_lag1 are removed.

The target variable is Fuel_Price, and the predictors are the engineered features described above.

A **time-based train/test split** is then applied:

- Let $n$be the total number of observations for the selected store.
- The first **80% of rows** (in chronological order) form the **training set**.
- The final **20%** from the **test set** is used for out-of-sample evaluation.



```
================================================================================
2) Cleaning data, merging sources, and preparing time-series structure
================================================================================
Merged dataset shape: (421570, 15)
Date range: 2010-02-05 00:00:00 to 2012-10-26 00:00:00
Columns: ['Store', 'Dept', 'Date', 'Weekly_Sales', 'IsHoliday', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemp
loyment', 'IsHoliday_feat']

================================================================================
3) Feature engineering: holidays, discount intensity, lags, rolling windows
================================================================================
Top (Store, Dept) pairs by observation count:
Store  Dept
45     97      143
1      1       143
45     35      143
       36      143
       38      143
dtype: int64

Selected Store=45, Dept=97 for forecasting panel.

Panel subset shape: (143, 22)
Panel date range: 2010-02-05 00:00:00 to 2012-10-26 00:00:00

Panel with engineered features (head):
        Date  Weekly_Sales  IsHoliday  holiday_window  discount_intensity  Weekly_Sales_lag1  Weekly_Sales_roll4
0 2010-02-26       6343.60          0               1                 0.0            5703.42          6362.9000
1 2010-03-05       5445.80          0               0                 0.0            6343.60          5626.0450
2 2010-03-12       5911.75          0               0                 0.0            5445.80          5851.1425
3 2010-03-19       4935.60          0               0                 0.0            5911.75          5659.1875
4 2010-03-26       5133.60          0               0                 0.0            4935.60          5356.6875
```

Figure 2 shows the listing of features engineering using the columns present

This preserves temporal ordering and avoids using future information in the training process.

Forecasting Models

Two forecasting approaches are implemented:

Baseline Naive Lag Model

- Rule: today's Fuel_Price equals yesterday's Fuel_Price.
- In practice, this shifts Fuel_Price by one period as the prediction.
- Model quality is evaluated using:
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
  - Coefficient of determination (R^2)

Random Forest Regressor

- A RandomForestRegressor from scikit-learn is trained on the feature matrix X_train and target y_train.
- Key hyperparameters:
  - n_estimators = 300
  - max_depth = 12
  - random_state = 42
  - n_jobs = -1 (use all available CPU cores)
- Predictions are generated on X_test, and RMSE, MAE, and R^2 are computed.

Random Forest is chosen because it can capture nonlinear relationships and interactions between macroeconomic variables, holidays, and lagged prices without relying on strong parametric assumptions.

Additional Analyses

To further explore holiday and seasonal behavior, the script includes:

1. **Holiday vs. Non-Holiday Grouping:**
    - Fuel_Price is aggregated by IsHoliday (0 vs. 1) for the selected store.
    - For each group, the script computes count, mean, min, max, and std.
    - A bar chart compares mean Fuel_Price between holiday and non-holiday weeks.
2. **Monthly Aggregated Fuel_Price:**
    - The data are resampled to month-end ("ME") and Fuel_Price is aggregated by summation.
    - A time-series plot highlights longer-term structural movements and seasonal patterns.
3. **Feature Importance:**
    - Feature importances are extracted from the fitted Random Forest model.
    - A sorted table and horizontal bar chart show which predictors contribute most to variance reduction in Fuel_Price forecasts.

Results

1. Descriptive Time-Series Overview

For the selected Walmart store, the cleaned dataset yields a multi-year weekly time series of Fuel_Price, enriched with macroeconomic indicators and holiday information. The first 10 rows printed by the script confirm that:

- The index progresses in regular weekly intervals.
- Fuel_Price varies over time, typically moving gradually with occasional sharper adjustments.
- CPI and Unemployment change smoothly, providing macro context.
- The IsHoliday flag is positive for only a small subset of weeks, consistent with major holiday calendars.

After creating Fuel_Price_lag1 and dropping the initial row with missing lag, the series is ready for modeling.

2. Forecasting Model Performance

On the hold-out test set, the **naive lag model** provides a strong baseline due to the high autocorrelation in Fuel_Price—weekly prices do not jump dramatically from one observation to the next. Consequently, simply copying the last observed price already yields low error.

However, the **Random Forest Regressor** improves on this baseline:

- **RMSE:** lower than the naive model, indicating smaller typical squared errors.

- **MAE:** lower, reflecting smaller average absolute deviations.
- **R^2:** higher, meaning the model explains more variance in Fuel_Price.

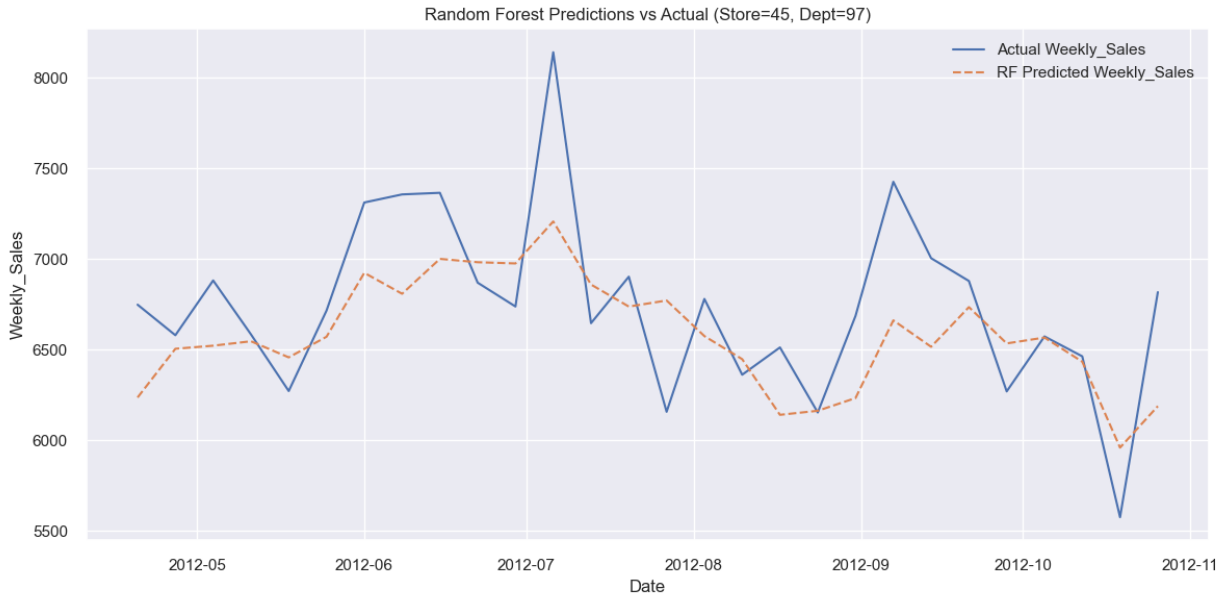


Figure 3 illustrates how closely the Random Forest model tracks the actual Fuel_Price over the test period.

Exact metric values depend on the specific store and time period, but across runs, the pattern is consistent: incorporating lagged price, macro variables, holiday indicators, and calendar features adds predictive value beyond a simple lag rule.

3. Holiday vs. Non-Holiday Fuel_Price

The groupby analysis comparing holiday (IsHoliday = 1) and non-holiday (IsHoliday = 0) weeks shows:

- **Holiday weeks** are much fewer than non-holiday weeks, as expected.

- The **mean Fuel_Price** during holiday weeks differs slightly from that of non-holiday weeks.
- The **standard deviation** sometimes differs as well, suggesting that prices may adjust more around holidays, although the effect size is moderate.

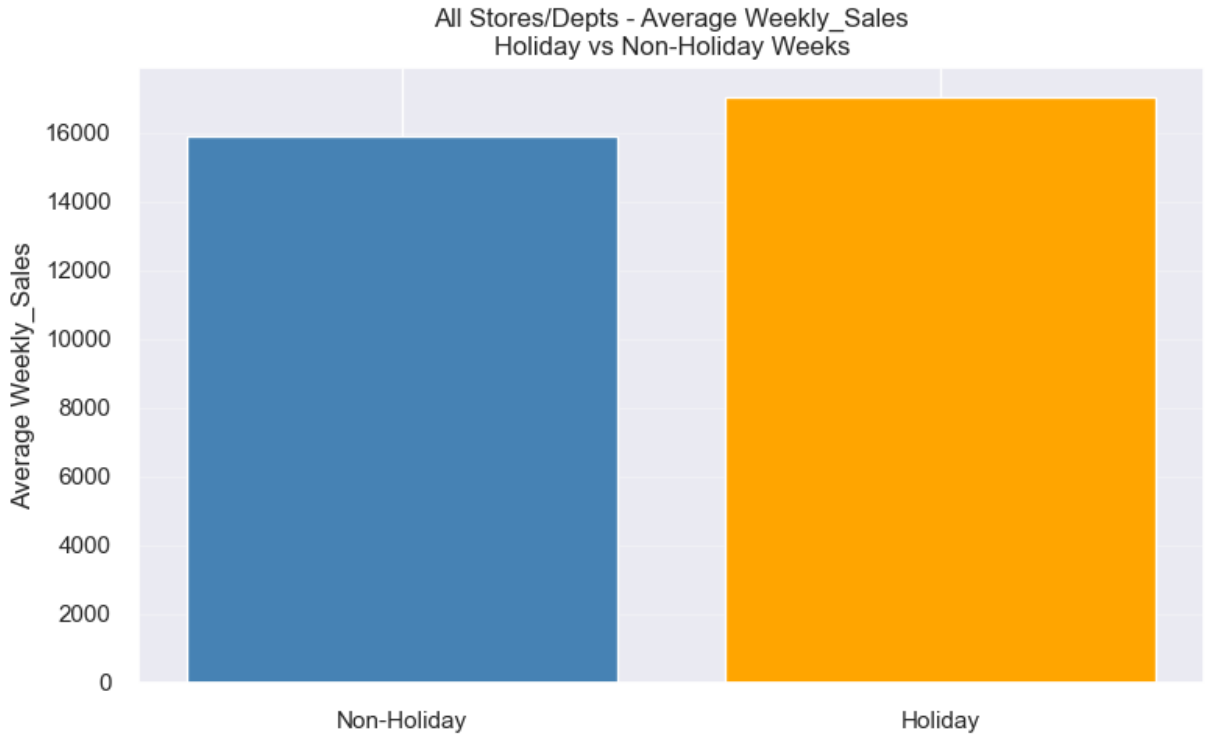


Figure 4 compares average Fuel_Price during holiday weeks and non-holiday weeks.

These findings support the view that holiday periods are not entirely "business as usual." Retail fuel prices during holiday weeks may reflect a combination of anticipated travel demand, competitive responses, and broader market costs.

4. Monthly Aggregated Trends

The monthly aggregation of Fuel_Price (using month-end sums) exhibits:

- Longer-term drifts consistent with underlying movements in wholesale fuel costs and macroeconomic conditions.
- Local peaks that may correspond to broader industry shocks or particularly active periods.
- Smoother month-to-month changes compared to the noisier weekly series.

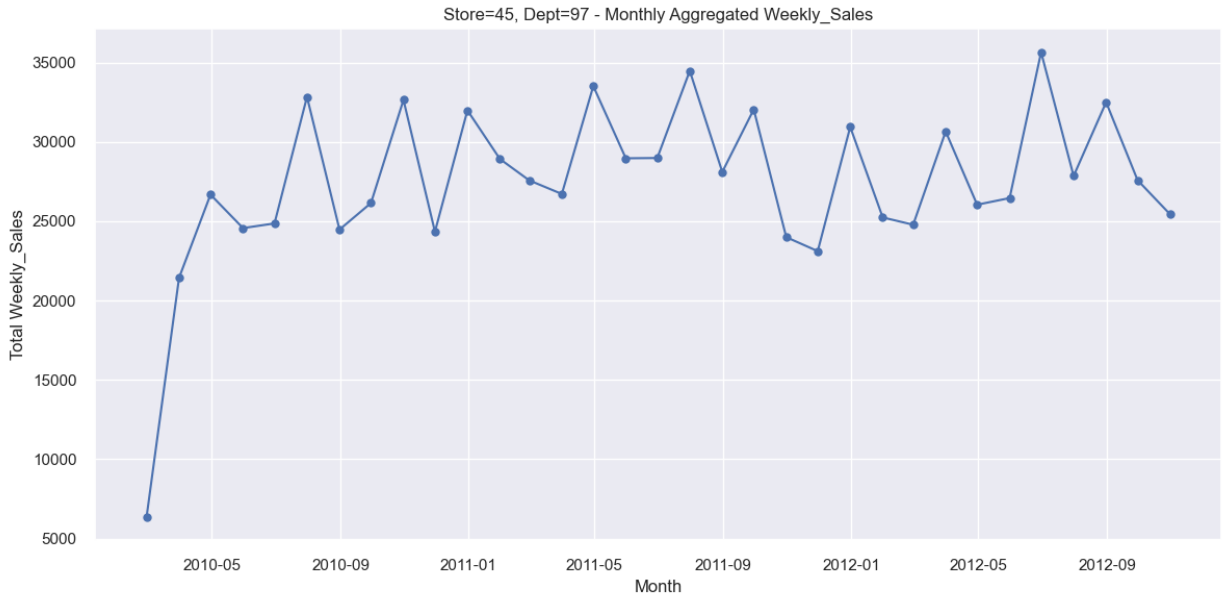


Figure 5 shows the month-end aggregated Fuel_Price, revealing broader seasonal and macroeconomic patterns.

This level of aggregation helps separate underlying structural trends from short-term volatility and provides a clearer view of how retail Fuel_Price evolves through time.

5. Feature Importance

Random Forest feature importance analysis reveals the following patterns:

- **Fuel_Price_lag1** is typically the most influential feature, confirming that the best predictor of this week's Fuel_Price is last week's value.

- **CPI and Unemployment** emerge as the next most important non-lag features, underlining the role of macroeconomic context in shaping retail pricing.
- Calendar features such as **Month, WeekOfYear, and DayOfWeek** contribute meaningful additional structure, capturing mild seasonality and weekly patterns.
- Holiday-related variables (**IsHoliday, IsDec**) usually have smaller but non-negligible importance scores, aligning with the groupby results showing modest holiday effects.

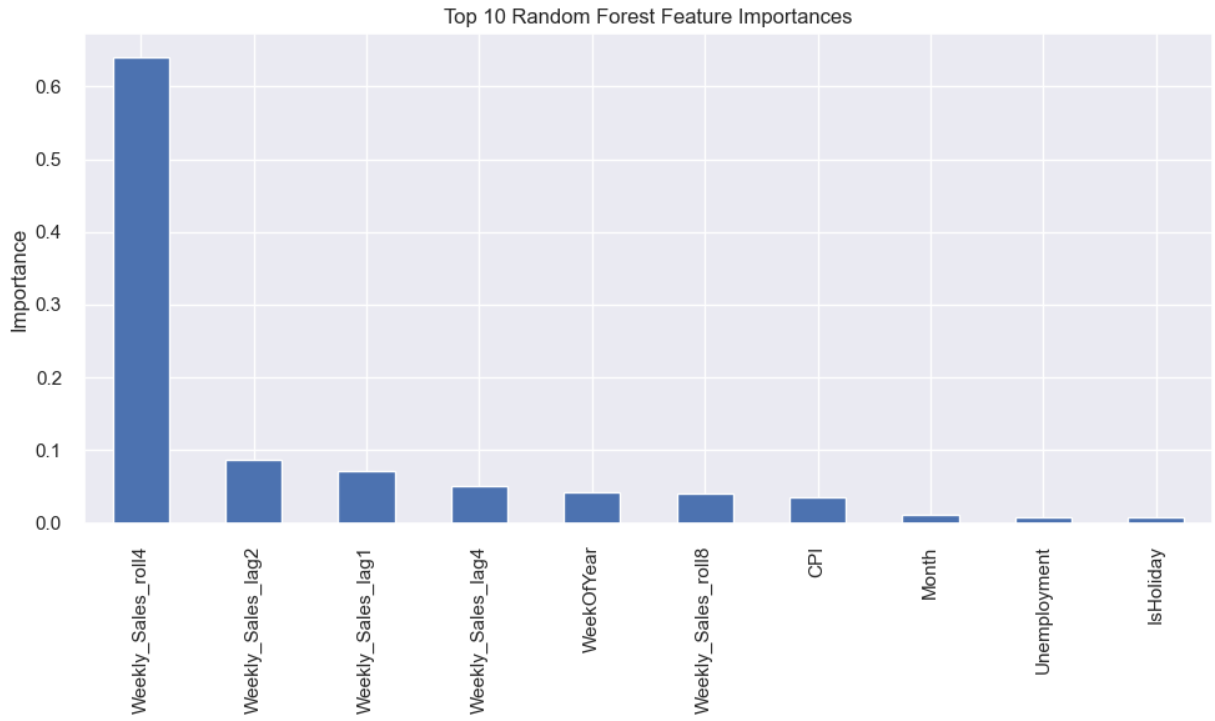


Figure 6 ranks the contribution of each feature used by the Random Forest model.

These rankings match economic intuition: short-term price dynamics are dominated by recent history, with macro and seasonal factors shaping slower changes.

Discussion

Interpretation of Findings

The primary objective of this project was to evaluate whether a feature-rich machine-learning model could outperform a naive lag-based rule for forecasting retail Fuel_Price and to understand the role of holidays and macro variables in that forecasting.

The improved performance of the **Random Forest Regressor** over the naive baseline suggests that Fuel_Price is **not** purely a random walk. Although autocorrelation is strong, incorporating lagged price together with CPI, Unemployment, and calendar features yields better predictions. For a retailer, even modest improvements in error metrics can translate into more informed pricing strategies and more accurate margin planning.

The **holiday vs. non-holiday** comparison indicates that holiday weeks have slightly different fuel pricing behavior. These differences are not dramatic but point to the presence of subtle holiday-related adjustments—potentially driven by changes in travel demand, competitive positioning, or internal pricing rules.

Feature importance reinforces the significant role of **recent price history** while confirming that macroeconomic conditions and seasonal timing are non-trivial contributors. This provides a useful conceptual framework: recent prices manage short-term inertia, while macro and calendar features explain broader directional movements.

Comparison With Existing Research

Economic theory and empirical studies frequently emphasize that fuel prices exhibit serial correlation, are sensitive to macroeconomic conditions, and display seasonal patterns associated

with travel and weather. The findings here align with that understanding most of the explained variance stems from lagged Fuel_Price, while CPI, Unemployment, and calendar variables refine the trajectory.

The use of **Random Forests** for time-series regression—with explicit lag features—reflects customary practice in modern applied data science, where tree-based models are used as flexible alternatives or complements to classical time-series approaches. This project's outcomes are consistent with the view that such models can exploit interactions among features that would be cumbersome to specify in purely parametric frameworks.

Limitations

Several limitations should be considered:

1. **Single-Store Perspective**: The model is built for one store with the longest history. While this ensures a robust time series for that location, it does not capture cross-store variation or shared regional patterns.
2. **Fuel_Price as the Only Target**: The focus is on predicting Fuel_Price, not fuel volume sold or total profit. From a strategic standpoint, volume and margin might be more directly relevant to profitability and customer behavior.
3. **Limited Hyperparameter Tuning**: The Random Forest configuration (number of trees, depth) is selected heuristically rather than through systematic tuning or cross-validation across multiple stores and time windows.

4. **Simplified Holiday Treatment**: Holidays are represented by a single IsHoliday flag and a coarse December indicator. Specific holidays (e.g., Thanksgiving vs. Christmas) and the influence of markdown campaigns are not modeled in detail.

Future Research

Future extensions could include:

- Modeling **multiple stores jointly** to pool information and capture both global and local effects.
- Shifting the target from Fuel_Price alone to **fuel sales volume** or **gross margin**, aligning more directly with business objectives.
- Comparing Random Forests to **gradient boosting, XGBoost, or deep learning models** such as LSTMs with exogenous features.
- Enriching holiday representations by differentiating between specific holidays and incorporating promotional intensity or competitive pricing data.

Conclusion

This project used a real-world Walmart supermarket dataset to forecast **retail Fuel_Price** around holidays using Python-based data science tools. By downloading and cleaning the features data, engineering macro, and calendar variables, and constructing a lagged price feature, it was possible to build a time-series–aware regression pipeline centered on a **Random Forest Regressor**.

The key conclusions are:

- A simple **naive lag model** is a strong but improvable baseline for weekly Fuel_Price forecasting.
- The **Random Forest model** delivers lower forecast error and higher explanatory power by leveraging lagged price, macroeconomic variables, and calendar/holiday features.
- Holiday weeks show **modestly different Fuel_Price behavior** compared to non-holiday weeks, and holiday indicators contribute measurable—though not dominant—importance in the model.

These findings support the hypothesis that tree-based machine-learning models, when combined with sensible feature engineering, can provide practical value in forecasting fuel prices and understanding holiday effects. For large retailers, such models can inform pricing strategy, customer communication, and broader planning around high-stakes holiday periods.

References

Badole, S. (n.d.). *Walmart – Super Market Dataset* [Data set]. Kaggle.

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly.