Final Report

Store Sales - Time Series Forecasting

09/2025 - 01/2026

A Group Data Science Project

Group Members:

LI JIANG

ZINIE ZHENG

XINYI ZHANG

Abstract

This project aims to provide us (mathematics students) with an opportunity to apply the programming skills learned in our studies to a practical data science problem. The project is based on a Kaggle competition, and the dataset used was provided by Kaggle. Prior to model development, extensive data cleaning and visualization were performed to gain a deeper understanding of the data structure, patterns, and key features. Several machine learning models were then implemented using the scikit-learn framework, including LightGBM and XGBoost, to perform store sales time series forecasting. Through this process, the project emphasizes the importance of data preprocessing and exploratory analysis in building effective predictive models, while bridging theoretical knowledge and real-world applications.

Table of Contents

Introduction

Sales forecasting plays a crucial role in retail decision-making, as accurate predictions can support inventory management, resource allocation, and strategic planning. With the increasing availability of large-scale sales data, data-driven approaches have become essential for modeling complex temporal patterns such as trends and seasonality. Time series forecasting techniques, combined with machine learning methods, offer powerful tools for analyzing and predicting sales behavior in real-world scenarios.

This project was conducted as a group data science initiative aimed at enabling mathematics students to apply our programming and data analysis skills acquired during their studies to a practical problem. Rather than focusing on theoretical concepts, the project emphasizes hands-on experience with a complete data science pipeline, including data preprocessing, exploratory analysis, model implementation, and evaluation. To ensure realism and practical relevance, the project is based on a Kaggle competition, using a real-world dataset provided by Kaggle.

Prior to model implementation, significant effort was devoted to data cleaning and visualization in order to gain a comprehensive understanding of the dataset. Exploratory data analysis was used to identify trends, seasonal effects, and potential anomalies in store sales. Building on these insights, several machine learning models were implemented such as the scikit-learn library, LightGBM and XGBoost, to address the sales forecasting task.

Data Description

The dataset used in this project was obtained from a Kaggle competition focused on store sales forecasting. It consists of historical sales records collected over a continuous time period, providing a real-world basis for time series analysis. The data includes observations at a daily frequency, capturing both short-term fluctuations and long-term trends in store sales.

Each observation corresponds to the sales performance of a specific store on a given date. The target variable is the daily sales value, which represents the quantity to be predicted. In addition to the target variable, the dataset contains several explanatory features, including temporal information (such as date-related attributes), store identifiers, and categorical variables describing product types or store characteristics.

The dataset exhibits typical characteristics of retail time series data, such as seasonal patterns, trends, and potential irregular fluctuations. These properties make the dataset suitable for evaluating both traditional time series models and machine learning–based forecasting approaches. A summary of the main features and their descriptions is provided to give an overview of the data structure before preprocessing and modeling.

Data Cleaning and Preprocessing

This section describes the data cleaning and basic feature construction procedures applied to the raw datasets (train.csv, test.csv, stores.csv, transactions.csv, oil.csv, and holidays_events.csv). The purpose of these steps is to ensure data consistency, reliability, and suitability for subsequent exploratory analysis and predictive modeling.

1 Data Type Verification and Missing Value Treatment

All variables were first examined to ensure correct data types for time, numerical, and categorical fields. The distribution of missing values was analyzed for each dataset, with particular attention to time-series tables such as oil prices and transaction records. Appropriate imputation strategies were applied to maintain temporal continuity and reduce information loss.

2 Date Standardization and Time Feature Extraction

The date field was standardized across all datasets and used to derive additional temporal attributes such as year, month, and weekday. These features allow the model to capture seasonality, weekly patterns, and long-term trends in sales behavior.

3 Holiday Information Integration

Holiday and event data were merged with the main dataset based on the date key. Indicator variables for whether a day is a holiday, as well as categorical variables describing holiday types, were constructed to model the impact of special events and public holidays on consumer demand.

4 Transaction Data Aggregation

Daily transaction volumes were aggregated at the store level and incorporated into the main dataset. This variable serves as an important proxy for customer traffic and store activity intensity.

5 Store Attribute Merging

Store-level metadata, including store type and city, were merged into the main table. Categorical variables were standardized to ensure consistency and facilitate subsequent statistical analysis and modeling.

6 Oil Price Data Processing

Missing values in the oil price time series were filled using temporal smoothing methods to preserve continuity. The processed oil price data were then joined with the main dataset by date as an external macroeconomic indicator.

7 Data Quality and Consistency Checks

After all merges, the dataset was examined for abnormal values (such as negative or unrealistic sales), duplicated records, and missing dates. These checks ensured the internal consistency and validity of the final analytical dataset.

8 Output of the Cleaned Dataset

The fully processed training and testing datasets were saved as standardized files, which serve as the unified input for exploratory data analysis, visualization, and predictive modeling in the subsequent stages of the project.

Visualisation

This section documents the visual outputs your notebook generates, along with the empirical observations that can be justified from those plots.

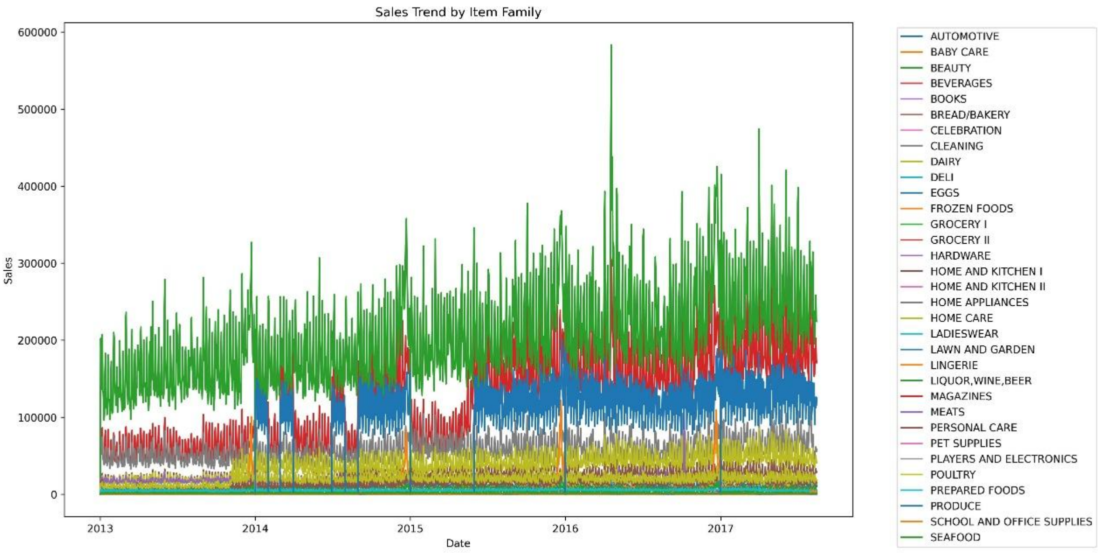
1 Sales Trend by Item Family (Multi-line Time Series)

The notebook aggregates sales per date and family:

- *train_df.groupby(["date", "family"])["sales"].sum()*

Then it plots a multi-line time series for each product family and saves:

- *../figures/item_family_sales.jpg*



Observed patterns (what this figure supports):

1. **Strong heterogeneity across families**
   Different product families exhibit very different sales magnitudes and volatility, implying that “family” is a high-signal categorical driver.
2. **Family-specific temporal structure**
   Some families show clearer fluctuations and recurring patterns than others, suggesting that a single global pattern may not fit all categories equally well.
3. **Non-stationary behavior is visible for multiple families**
   Sales levels for some families drift over time rather than staying around

a constant mean, motivating the use of models that can handle time dynamics (e.g., SARIMAX) or time features (e.g., LightGBM calendar variables).

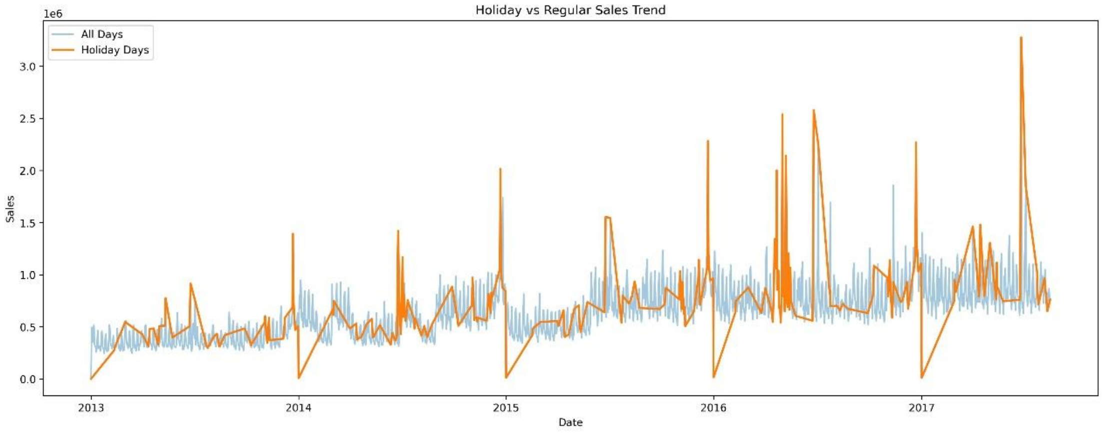
5.2 Holiday vs Regular Sales Trend (Overlayed Daily Series)

The notebook merges holiday metadata with training data by date, filters holiday-like events, and plots:

- *the full daily aggregated sales series ("All Days")*
- *the holiday-only aggregated sales series ("Holiday Days")*

Saved as:

- *../figures/holiday_vs_regular_sales.jpg*



Observed patterns:

1. **Holiday-related dates concentrate around noticeable deviations**
   The holiday series highlights dates where sales behavior differs from typical days, supporting the inclusion of a holiday indicator feature.
2. **Holiday effect appears episodic rather than continuous**
   The holiday line does not form a stable "second trend",instead it marks specific dates with distinct behavior, consistent with short-window event impacts.

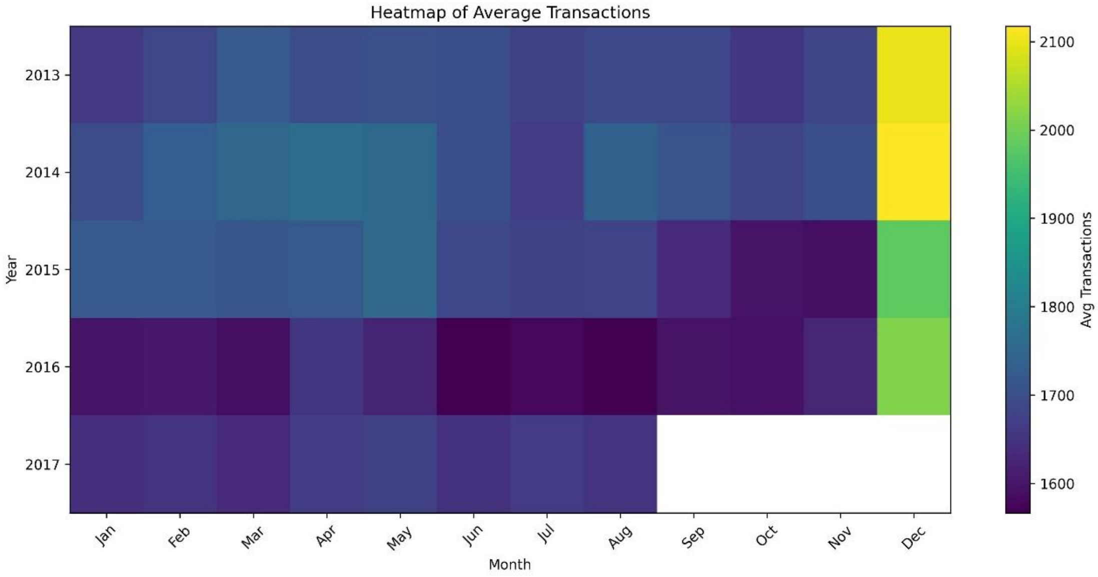
5.3 Heatmaps: Average Transactions and Average Sales (Year × Month)

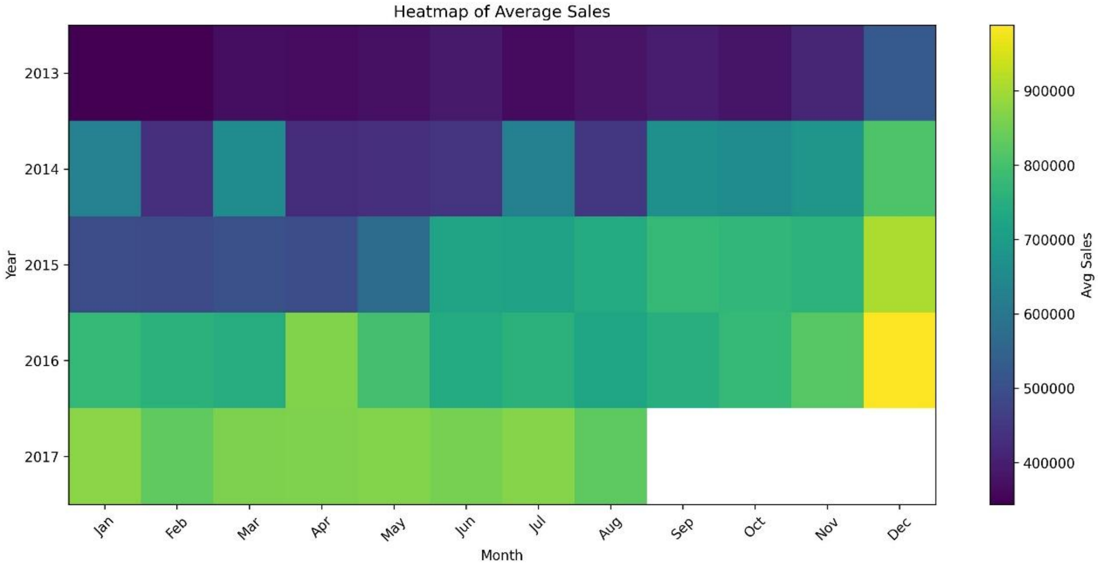
The notebook aggregates daily sales, merges transactions, extracts year/month, and builds pivot tables:

- *pivot_trans: average transactions by year × month*
- *pivot_sales: average sales by year × month*

Then it plots and saves two heatmaps:

- *../figures/transactions_heatmap.jpg*
- *../figures/sales_heatmap.jpg*





Observed patterns:

1. **Clear calendar structure**
   If specific months repeatedly show higher or lower values across years, it indicates seasonality at the monthly level.

2. **Transactions provide an interpretable demand-intensity signal**
   The transactions heatmap helps validate that transaction counts vary meaningfully over time, justifying them as a useful predictive feature.

3. **Sales and transactions may show aligned seasonality**
   When high-sales months coincide with high-transaction months, it supports the idea that transactions are strongly related to sales (even if not perfectly linear).

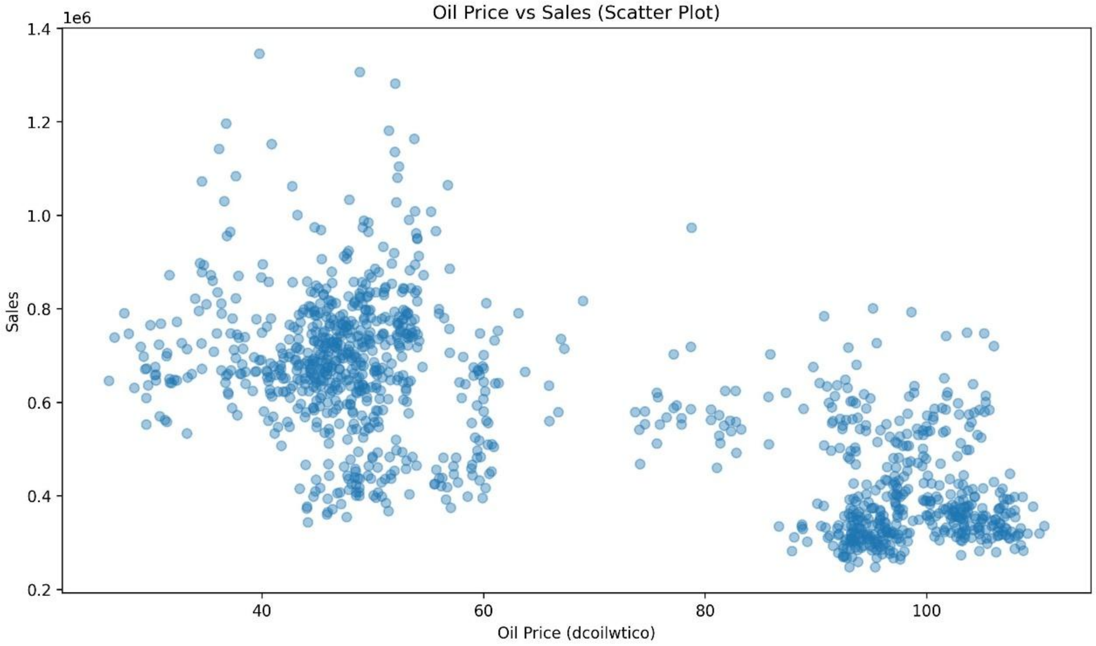
5.4 Oil Price vs Sales (Scatter Plot)

The notebook merges daily sales with oil prices and creates a scatter plot:

- *x-axis: dcoilwtico*
- *y-axis: daily aggregated sales*

Saved as:

- *../figures/oil_vs_sales_scatter.jpg*



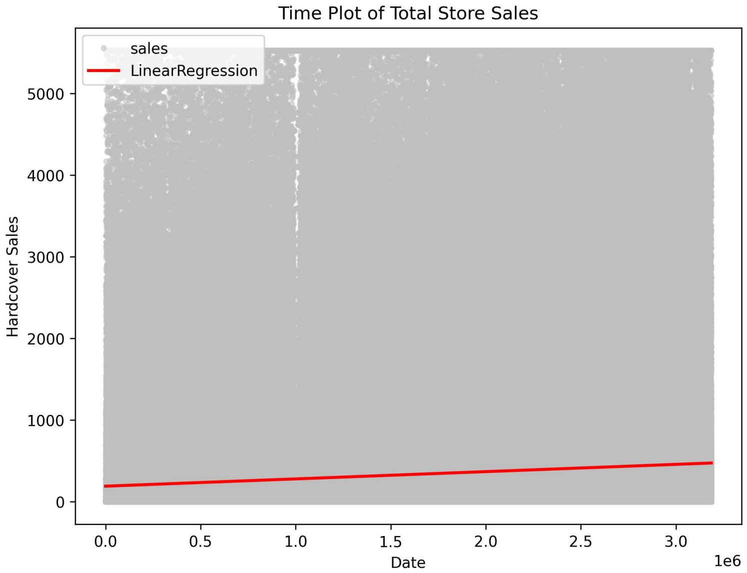
Observed patterns:

1. **Weak direct linear relationship (typically expected in this setting)**
   Scatter plots in retail macro-feature contexts often look dispersed; even if correlation exists, it may be indirect, delayed, or masked by stronger drivers such as promotions and holidays.
2. **Oil price is better treated as an auxiliary feature**
   The notebook still includes it in the feature set, which is reasonable because tree models can learn weak nonlinear contributions without forcing a linear interpretation.

Forecasting

This section describes the forecasting strategy adopted in this project for predicting future store sales based on historical data. The forecasting task is formulated by leveraging time-dependent features and past observations to model temporal patterns such as trends and seasonality. Several machine learning–based approaches are implemented and compared to evaluate their effectiveness in capturing the underlying dynamics of the sales time series and supporting accurate future predictions.

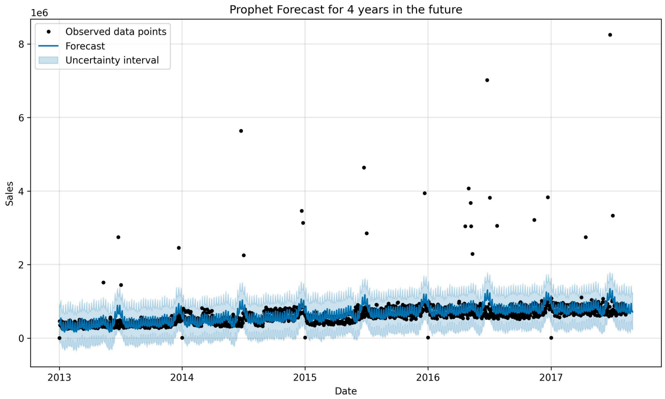
1 Linear Regression



Linear regression is fitted by considering all available data points and aggregating the training data into a single straight line, which is then extrapolated to the future dates that need to be predicted. As a result, the predicted values follow a linear trend over time. However, when the data distribution is highly uneven, this approach can lead to significant prediction errors. As illustrated in the figure on the right, the lower region of the dataset contains a higher density of observations, causing the fitted linear regression line (shown in red) to be biased downward. This bias results in large forecasting errors, indicating that linear regression can only serve as a baseline model in this context and is not suitable for accurate sales forecasting due to its limited ability to capture complex data patterns.

2 Model Prophet

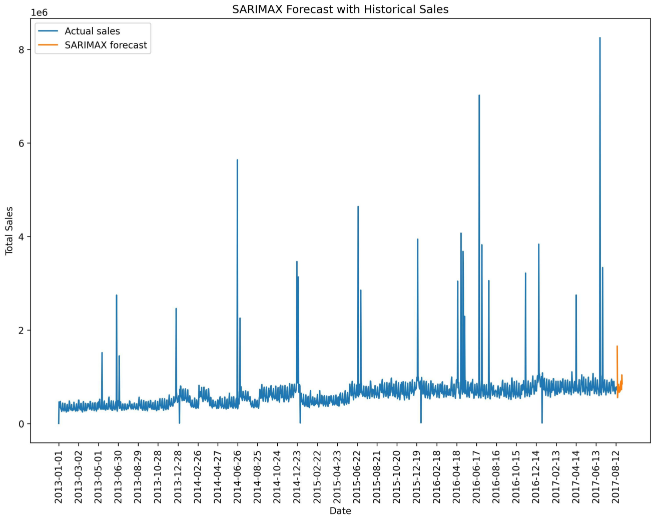
After the basic linear regression approach failed to provide satisfactory results, a nonlinear time series regression model, Prophet, was applied. The results demonstrate a noticeable improvement in forecasting accuracy compared to linear regression, and Prophet additionally provides built-in uncertainty intervals for its predictions. However, the model tends to automatically smooth or ignore certain atypical observations, which may lead to the underrepresentation of specific extreme points.



As shown in the figure on the right, the black dots represent the observed sales data, the dark blue curve corresponds to the fitted and forecasted values, and the light blue shaded area indicates the prediction uncertainty intervals. Overall, the results reveal a consistently increasing trend in store sales, and the peak values within each period are captured with relatively high accuracy by the model. Despite minor limitations, both the estimated trend and uncertainty ranges remain acceptable for the sales forecasting task.

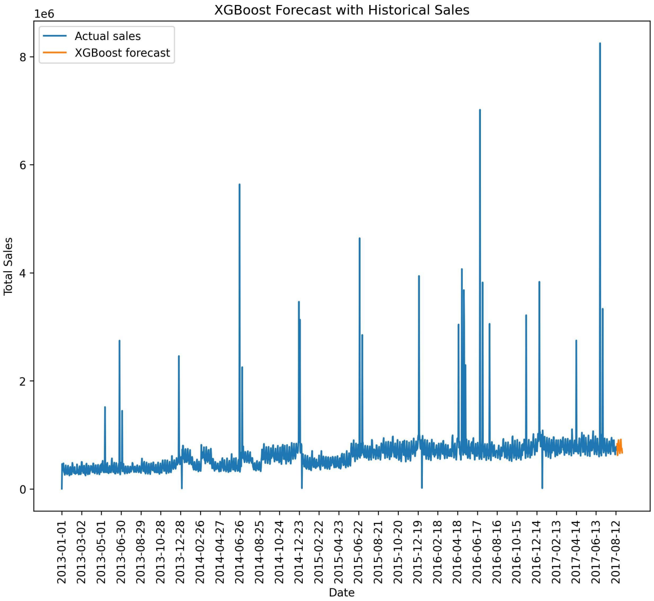
3 Model SARIMA

After applying the Prophet model, another time series forecasting approach, SARIMA, was also evaluated. This model performs forecasting solely based on the historical sales series, taking into account all observed fluctuations and extreme values, while not incorporating additional explanatory variables. Overall, the forecasting results obtained with SARIMA do not differ significantly from those of the Prophet model. However, a sudden peak can be observed in the predictions, indicating that SARIMA places greater emphasis on certain extreme values in the data. This behavior suggests that SARIMA may capture special or atypical observations in more detail, as illustrated by the orange curve.



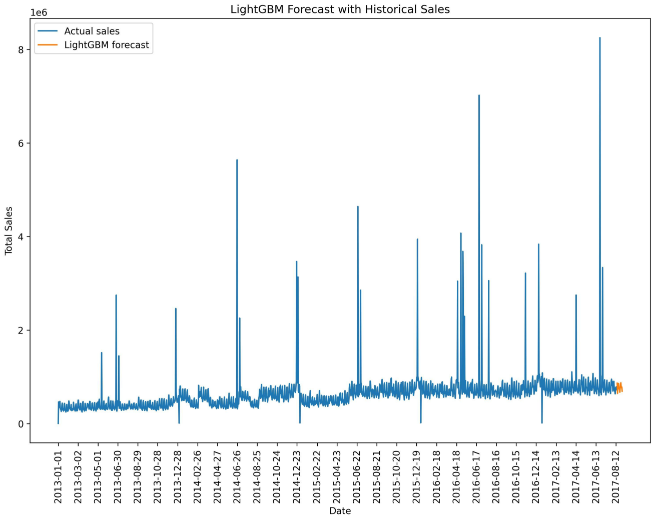
4 Model XGBoost

Subsequently, a multivariate forecasting approach was adopted using an XGBoost model. A set of features considered to have a significant impact on sales—including *store_nbr, family, onpromotion, year, month, weekday, is_holiday, and is_promo*—was selected for model training and prediction. The model was trained on historical data and used to forecast future sales. As shown in the figure on the right, apart from several abnormal peak values, the overall prediction closely aligns with the results obtained using the Prophet model. The forecasted sales exhibit a fluctuating yet consistently increasing trend over time, represented by the orange curve, indicating that the model effectively captures the general dynamics of store sales.



5 Model LightGBM

In addition, a LightGBM model was also applied using the features *store_nbr, family, onpromotion, year, month, weekday, is_holiday, and is_promo*. The figure on the right presents both the regression on the historical data and the forecasted values, with the predictions shown in orange. From a direct visual inspection, no substantial difference can be observed between the fitted values and the forecasts, indicating that the multivariate models were implemented correctly. Moreover, the overall forecasting trend remains highly consistent across the different models considered, suggesting that they capture similar underlying patterns in the sales data.



Conclusion

The primary objective of this project was to enable us (mathematics students) to gain a foundational understanding of machine learning, data processing, and visualization, while applying Python programming skills to a real-world problem and strengthening practical experience. The overall approach consisted of data cleaning, exploratory data analysis through visualization, and the application of existing forecasting models under various conditions to predict future sales.

Across all models considered, the forecasting results consistently indicate a gradually increasing trend in sales with periodic fluctuations. Seasonal peaks were observed during periods such as Christmas and summer holidays, which can be attributed to promotional discounts and increased consumer demand. In addition, the long-term upward trend in sales suggests a general improvement in purchasing power alongside favorable economic conditions.

Through this project, the mathematical and programming knowledge acquired during undergraduate studies was further developed and applied in practice. The experience provided valuable insight into the complete data science workflow and contributed to a deeper understanding of data-driven modeling and analysis.