Assignment 2ML as a Service

short line

NUTAN THAPA

Student ID: 24614000  
7TH OCT 2023

36120 - Advanced Machine Learning Application

Master of Data Science and Innovation

University of Technology of Sydney

Table of Contents

[**1. Executive Summary 2**](#_xvtyf97iwmd0)

[**2. Business Understanding**](#_if9iwijz2pg3) **4**

[a. Business Use Cases](#_bnjfuh5ewogw) 5

[**3. Data Understanding**](#_yg03y3w5hig9) **6-10**

[**4. Data Preparation**](#_tj77rrmixzhi) **11-12**

[**5. Modeling**](#_6vwh14nc44ga) **13-16**

[a. Prediction Model](#_uv8sn0ik6017) 13-14

[b. Forecasting Model](#_wtwxuxemqtvc) 15-16

[**6. Evaluation**](#_mngiyx5tet6w) **17-20**

[a. Evaluation Metrics](#_qx20m6hrk8uo) 17

[b. Results and Analysis](#_25v57zsj5m7g) 18

[c. Business Impact and Benefits](#_hkob76wu4d6q) 18-19

[d. Data Privacy and Ethical Concerns](#_uf1z6gbsejg6) 19-20

[**7. Deployment**](#_8rqux13i2o01) **21-23**

[**8. Conclusion**](#_9hr11g79asdx) **24**

[**9. References**](#_ma3y8ytjvp1y) **25**

# 

# Executive Summary

**Project Overview:**

In the diverse retail realms of California, Texas, and Wisconsin, theretailer initiated a dual-pronged strategy across its 10 stores. The goal encompassed deploying a predictive model for specific sales and a forecasting model for a 7-day aggregated sales overview across three categories: hobbies, foods, and household.

**Significance & Context:**

In retail, achieving inventory equilibrium is paramount. With the challenges of varied product categories and diverse consumer behavior across states, the need for precise predictive and forecasting models was evident for operational finesse, financial precision, and enhanced customer satisfaction.

**Achieved Outcomes:**

1. **Imbalanced Data:** The time series dataset may contain sales anomalies, but traditional imbalances aren't prominent.
2. **Inventory Management:** Models improved inventory accuracy, reducing costs and enhancing customer satisfaction.
3. **Resource Use:** Predictions ensured efficient staff allocation.
4. **Promotions:** Forecasts shaped promotional timing and strategy.
5. **Financial Planning:** Model insights refined budgeting and product strategy.
6. **Supply Chain:** Predictions streamlined product ordering, minimizing delays.

The union of predictive and forecasting models fortified our retailer's operations, strategy, and finances. The journey illuminated both the value of accurate modeling and the risks of inaccuracies. As the retail landscape evolves, continual model refinement remains central to our strategy.

short dash

# Business Understanding

## Business Use Cases

The project finds application in the following scenarios:

1. **Inventory Management:** Retailers constantly grapple with the challenge of determining optimal stock levels for each product, across every store. An overstock results in increased holding costs and potential wastage, especially for perishable items, while understocking can lead to missed sales opportunities and disgruntled customers.
2. **Resource Allocation:** Optimize staff numbers based on expected footfall to control costs and maintain service.
3. **Promotional Planning:** Use sales insights to time promotions for maximum impact.

**Challenges and Opportunities:**

The retail industry is in a perpetual state of flux, influenced by factors such as seasonal demand, economic shifts, and consumer behavior trends. The project was motivated by the challenges of:

* Responding quickly and effectively to market changes.
* Personalizing the shopping experience for customers in different states with potentially varied preferences.
* Efficiently managing resources across 10 geographically diverse stores.

Machine learning algorithms are pertinent in this context because they can analyze vast amounts of historical data, identify patterns, and make predictions about future sales, allowing the business to make proactive, dc

1. Key Objectives

The primary objectives of the project are:

1. To predict sales revenue for specific items in each store on given dates.
2. To forecast consolidated sales revenue over a week for all stores.

**Stakeholders and Requirements:**

* **Store Managers:** Seek sales predictions for inventory, staffing, and promotions.
* **Marketing Team:** Needs forecasts for campaign strategy and budget allocation.
* **Supply Chain Managers:** Want predictive data for product ordering, delivery, and storage.
* **Financial Planners**: Use forecasts for cash management and budgeting.

**Project's Approach to Address Requirements:**

By deploying machine learning algorithms, the project seeks to:

1. **Store Managers:** Provide sales forecasts for better inventory and staff planning.
2. **Marketing Team:** Deliver insights for campaign strategies based on sales trends.
3. **Supply Chain Managers:** Offer data insights for efficient operations.
4. **Financial Planners:** Give revenue predictions for sound financial planning.

short dash

# Data Understanding

**.1. Dataset Description and Sources:**

The project makes use of three primary datasets, each serving a distinct function:

* Calendar Dataset (calendar.csv): Links sale days to dates and offers daily event context.
* Calendar Events Data (calendar\_events.csv): Timeline of events across years.
* Selling Prices Dataset (items\_weekly\_sell\_prices.csv): Weekly item prices for stores, showing pricing trends.
* Sales Training Dataset (sales\_train.csv): Daily historical sales of items across various stores.

**2. Data Collection Methods:**

Given the nature of the data, it's likely sourced directly from the retail chain's transactional and operational databases. The sales data, for instance, may have been aggregated from point-of-sale systems across different stores, while the calendar data may be influenced by a combination of local event calendars and internal marketing/promotion calendars.

**3. Data Limitations:**

One of the apparent limitations of the data is its encapsulation of only internal factors. While it gives insights into items, their prices, and sales volume, it might not account for external factors like regional economic conditions, competitor actions, or unrecorded local events that can have a significant impact on sales. Additionally, if there are any discrepancies in data recording or missing values, this can lead to potential inaccuracies in predictions.

4. **Variables/Features and Their Significance:**

* **Date**: This feature allows us to perform time series analysis, which is crucial for forecasting sales. It lets us see trends, seasonalities, and patterns in the sales data.
* **Sales Day:** A reference that connects the sales data to the actual date, facilitating easier mapping between datasets.
* **Store ID and Item ID:** These identifiers help in segmenting the data, understanding sales patterns for specific items in particular stores, and discerning any regional or product-specific trends.
* **Selling Price:** Integral for revenue calculations. Any fluctuation in this can be correlated with changes in sales volumes to understand pricing strategies' effectiveness.
* **Total Sales Revenue:** This aggregated feature is the ultimate measure of retail success and is the primary target for our forecasting endeavors.
* **Event-related Features (from the Calendar):** These can provide context to sudden spikes or drops in sales, helping to understand the influence of events on consumer buying behavior.

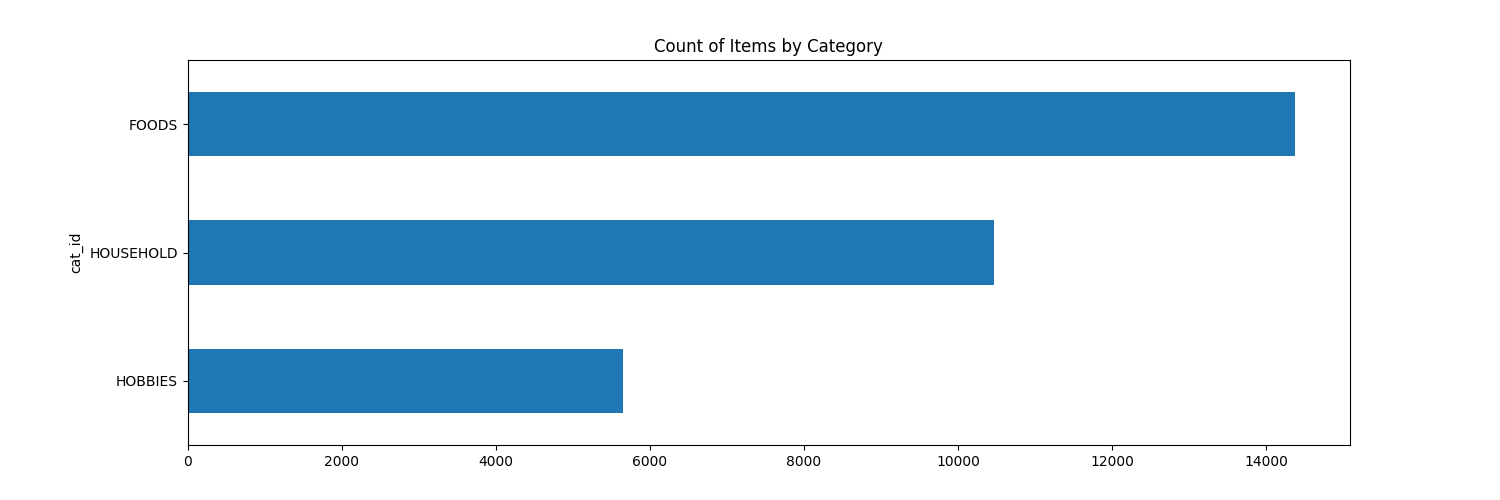
**5. Exploratory Data Analysis (EDA):**

In the preprocessing steps, a significant amount of exploratory data analysis was inherently conducted:

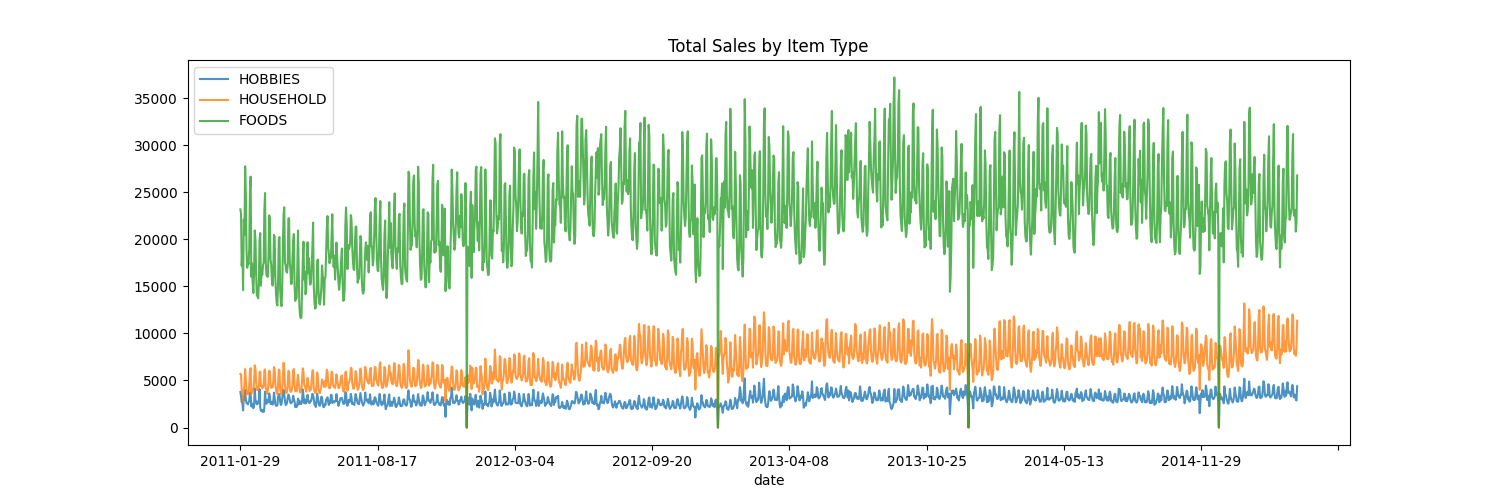
The data was reshaped from a wide to long format, ensuring better visibility of sales trends.By merging the datasets, we could see how different variables (like item prices or events) affected daily sales figures.

The computation of item\_daily\_revenue and subsequent aggregation into Total\_sales\_revenue illuminated the revenue streams, providing insights into the best-performing days, stores, or items.

Few EDA’s were conducted which are listed below:



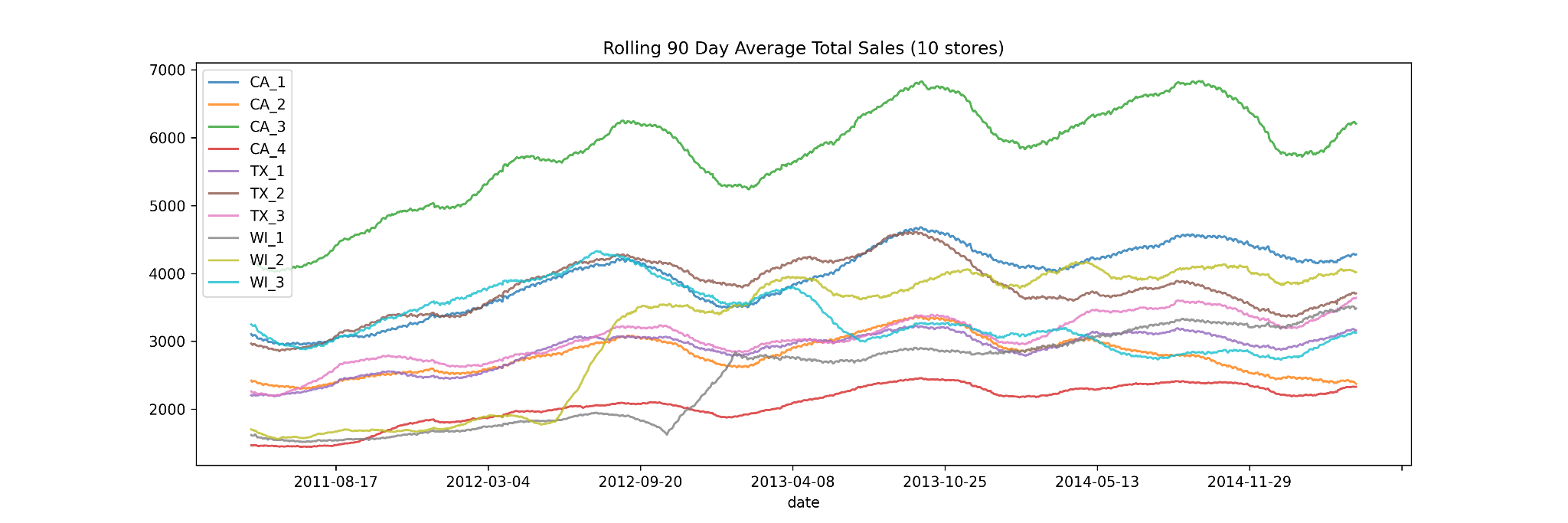
**Fig.1: Sales of categories**



**Fig.2. Sales based on categories**

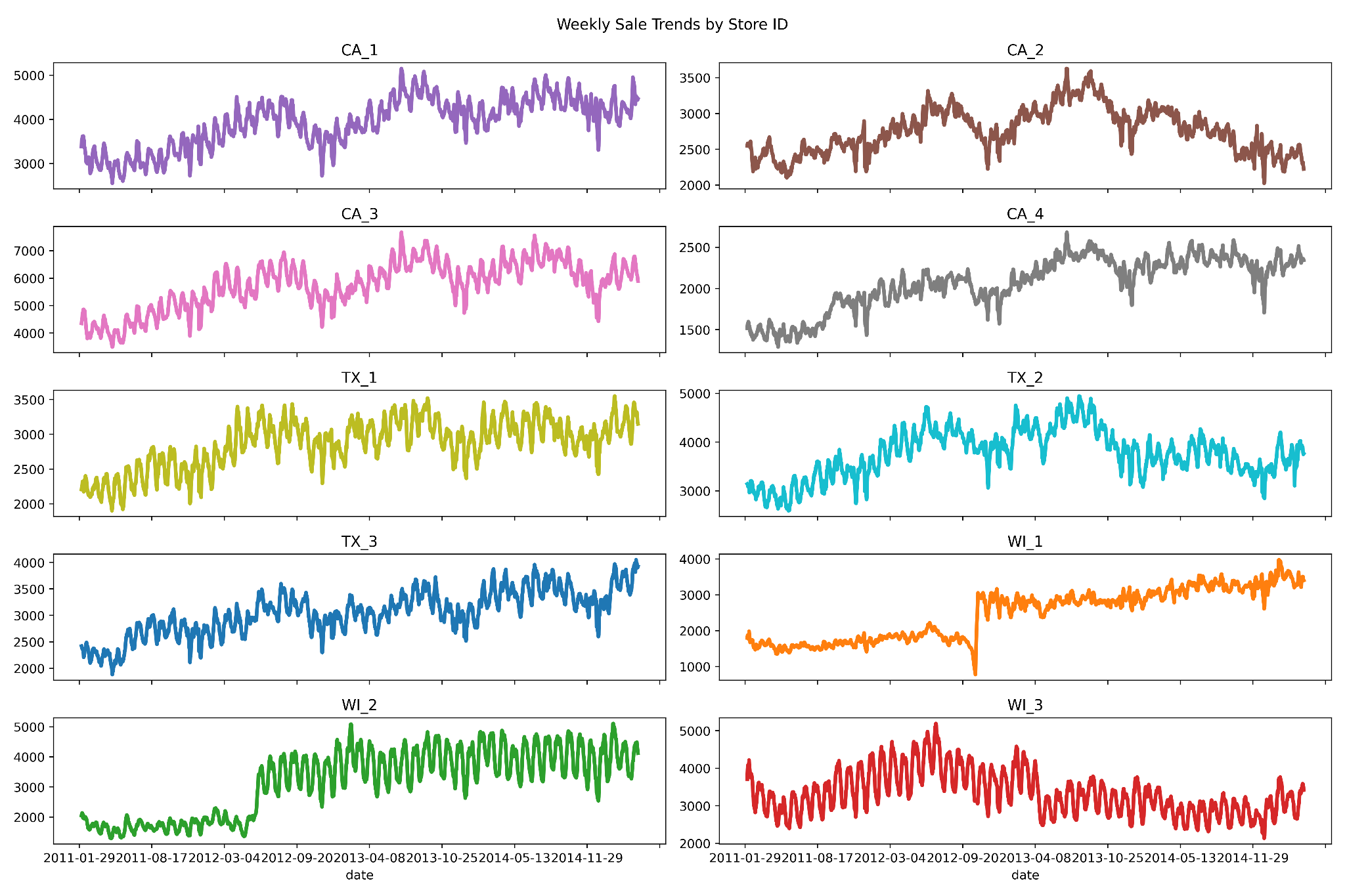
**The dip seen on the above plots is because it's on Christmas Day which is usually a holiday and almost every store is closed.**

We also checked the average sales over a period for different stores.



**Fig.3: Average Sales of different store**

As it can be seen from the above figure, the highest sales were for CA\_3 stores

.  **Fig.4: Weekly Sales trends for different Stores**

short dash

# Data Preparation

**1.** **Data Cleaning and Preprocessing:**

* Melt Operation: Reshaped sales data from wide to long format for better management.
* Merging Datasets: Combined sales, calendar, and price datasets for a comprehensive view.
* Date Handling: Standardized date entries to datetime format for seamless operations.

**2**. **Feature Engineering:**

* **Item Daily Revenue:** A new feature called item\_daily\_revenue was computed by multiplying the daily sales volume of an item with its selling price. This feature helped in understanding the revenue contribution of individual items.
* **Total Sales Revenue:** After calculating the daily revenue for each item, the revenues were aggregated across all items for a given day. This aggregation gave the Total\_sales\_revenue feature, which became the primary target for forecasting.
* **Time-based Segmentation:** By segmenting the dataset based on time, training and testing sets were created. This split is crucial for validating the performance of the forecasting models.

**3. Handling Missing Values, Outliers, and Imbalanced Data:**

* **Missing Values:** While the provided code does not explicitly showcase any missing value treatment, the merge operations, especially the 'inner' joins, implicitly ensure that only rows with matching keys in both datasets are retained. Any mismatch, which could arise from missing data in one of the datasets, would lead to those rows being excluded.
* **Outliers:** The process performed does not delve into explicit outlier handling.
* I**mbalanced Data:** Due to the dataset's time series nature, traditional imbalances aren't a concern, though anomalies in sales data might exist.

**4. Dataset Transformations:**

* **Pivoting:** Restructured daily revenues by store for better visualization and total sales calculation.
* **Time Series Split:** Used all but the last seven days for training and reserved those for testing to maintain chronological order for forecasting.

short dash

# Modeling

**1. PREDICTION MODEL**

The goal of the project is to predict 'Sales Revenue'. Given that the output variable is continuous, regression models are apt for this task. The dataset, being a blend of categorical and continuous features, demands flexible and ensemble algorithms for efficient learning.

**Parameter Tuning and Model Selection:** Throughout the modeling, different algorithms are chosen with different sets of hyperparameters. The intention behind this variety is to experiment, compare, and choose the best performing model for the given data. The implementation involves splitting the dataset into training and testing sets, fitting the models on training data, and evaluating them on test data.

**Approach 1: Linear Regression**

* **Key Hyperparameters:** Default settings were used since Linear Regression does not have hyperparameters like tree depth or learning rate. This model assumes a linear relationship between predictors and the target.
* **Data Preprocessing:** Features were encoded, ensuring they are in a format suitable for the regression. The target 'Sales Revenue' and the 'date' columns were dropped from predictors.

**Training Process:**

The training set was fed into the linear regression model, which learned the coefficients for each feature to minimize the error between predicted and actual sales revenue.

**Approach 2: Random Forest Regressor**

**Algorithm and Hyperparameters:**

* **Key Hyperparameters:** n\_estimators=100 indicating the number of trees in the forest.

Preprocessing or Feature Engineering specific to this model:

Random Forest can handle a mix of numerical and categorical features, and the data is prepared similarly to the first approach.

**Training Process:**

* The ensemble method builds multiple decision trees during training and outputs the average prediction of the individual trees for regression tasks.
* Cross-validation was also introduced with 10 folds to get an insight into the model's performance on different segments of the data and ensure the model is not overfitting.

**Approach 3: LightGBM(Light Gradient Boosting Machine)**

**Key Hyperparameters:**

* boosting\_type='gbdt': Traditional Gradient Boosting Decision Tree
* objective='regression': As the task is regression
* metric='l2': Using L2 loss (Mean Squared Error)
* num\_leaves=31: Maximum tree leaves for base learners
* learning\_rate=0.05: Rate at which the model adjusts based on errors from previous iterations
* feature\_fraction=0.9: Fraction of features to be taken for each iteration

**Preprocessing or Feature Engineering specific to this model:**

* LightGBM can handle categorical features inherently. In this scenario, data was prepared analogously to the previous approaches, dropping the target and 'date' columns.

**Training Process:**

* The model was trained with an early stopping round of 10 to prevent overfitting. LightGBM grows trees vertically, making it more efficient and faster, especially for large datasets.

In conclusion, different algorithms possess their own strengths and weaknesses. Linear Regression offers simplicity and explainability, while Random Forest and LightGBM provide higher flexibility and accuracy, especially with complex datasets. The selection of these algorithms and their iterative evaluation ensures the adoption of a model that delivers the best predictions for 'Sales Revenue'.

**2. FORECASTING MODEL**

**Approach 1: ARIMA (AutoRegressive Integrated Moving Average)**

* **Algorithm Details:** ARIMA is a linear time series forecasting model that combines autoregression, differencing, and moving averages to model time series data. It's designed to capture different patterns and structures in time series data.
* **Hyperparameters:** The order parameter for ARIMA was set to (5,1,0), representing the AR, I, and MA components, respectively.
* **Preprocessing Specifics:** The data was transformed to ensure it's of datetime type, and it's split into training and testing datasets. The frequency was set to daily, and the sales revenue column was converted to float for modeling.
* **Training Process:** The model was trained using the ‘Total\_sales\_revenue ‘ from the training dataset and then forecasted for the next 7 days.
* **Handling Imbalanced Data:** Usually ARIMA doesn't require handling imbalanced data.

**Approach 2: SARIMAX (Seasonal ARIMA with eXogenous regressors)**

* **Algorithm Details:** SARIMAX extends ARIMA by adding seasonality and the capability to model the impact of exogenous variables. It's especially useful for time series data with a clear seasonal pattern.
* **Hyperparameters:** The model was specified with the orders (5,1,0) for ARIMA and (1,1,1,12) for the seasonal component, indicating monthly seasonality.
* **Preprocessing Specifics:** Data conversion to datetime type and float type was done, with the sales revenue column converted to float.
* **Training Process:** The model was trained on the training data and then forecasted for the entire testing period.
* **Handling Imbalanced Data:** SARIMAX typically doesn't need special treatment for imbalanced datasets.

**Approach 3: Holt-Winters Exponential Smoothing**

* **Algorithm Details:** The Holt-Winters method extends simple exponential smoothing to time series data with trends and seasonality. It uses a weighted average of past observations to make forecasts, giving more weight to recent observations.
* **Hyperparameters:** Both trend and seasonality were set to 'add', indicating an additive model. The seasonal period was set to 7, indicating weekly seasonality.
* **Preprocessing Specifics:** The dataset index was ensured to be of datetime type, and the sales revenue column was converted to float.
* **Training Process:** The model was fitted on the training data, and forecasts were generated for the testing period.
* **Handling Imbalanced Data:** exponential smoothing methods usually don't require special treatment for imbalanced data.

**Rationale Behind Algorithm Selection:**

1. **ARIMA:** Chosen for its capability in capturing the inherent patterns in time-series data without seasonality.
2. **SARIMAX:** With sales data likely having seasonal patterns (like higher sales during holidays), SARIMAX is apt as it captures both non-seasonal and seasonal trends.
3. **Holt-Winters:** This method's strength lies in its ability to capture both trend and seasonality, making it a suitable choice for time series forecasting.

short dash

# Evaluation

The evaluation metrics used for these models are the Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Root Mean Squared Error (RMSE).

**Mean Squared Error (MSE):** Represents the average of the squares of the errors between the predicted and actual values. It gives more weight to large errors.

**Mean Absolute Error (MAE):** Represents the average absolute differences between predicted and actual values. It gives equal weight to all errors.

**R-squared (R^2):** Represents the proportion of the variance in the dependent variable that's explained by the independent variables in a regression model.

**Root Mean Squared Error (RMSE):** The square root of MSE. Represents the standard deviation of the residuals.

These metrics were chosen as they are standard measures to evaluate the performance of regression and forecasting models. They provide a comprehensive understanding of model accuracy (MSE, RMSE), model fit (R^2), and prediction deviation (MAE).

**Results and Analysis**

1. **Linear Regression:**

**MSE: 0.2716, MAE: 0.4279, R^2: 0.7235**

Linear regression seems to have a decent fit (R^2 of 72.35%) but higher error rates compared to the other models.

1. **Random Forest:**

**MSE: 0.0042, MAE: 0.0090, R^2: 0.9957**

With an average R^2 of 94.8% over 10-folds (with an SD of 4.86%), this model seems to be highly accurate and consistent.

1. **LightGBM:**

**MSE: 0.0201, MAE: 0.0745, R^2: 0.9796**

LightGBM also showcased high accuracy with an R^2 of 97.96%.

**Forecasting Models:**

* ARIMA: RMSE: 11922.50
* SARIMAX: RMSE: 12739.34
* Holt-Winters: RMSE: 8173.58

Given the results, the Random Forest and LightGBM stands out as the most accurate regression model with the highest R^2 and lowest error rates. For forecasting models, Holt-Winters has the lowest RMSE, making it the best among the three.

**Key Insights:**

* LIGHTGBM and Random-Forest's performance suggests that the dataset might have non-linear relationships, which this model can capture effectively. Random Forest's performance was better as well but after cross validation its performance declined a bit but I think both models can be utilised for better prediction.
* Holt-Winters outperforming ARIMA and SARIMAX suggests there might be significant seasonality or trend components in the forecasting data that it captures better.

**Business Impact and Benefits**

The selection of an optimal model is crucial for driving value and benefits to the business.

**Impact of the Final Model:**

* A highly accurate model, like the LightGBM in this context, can lead to better decisions, optimized operations, reduced costs, and increased revenue.
* For forecasting, the Holt-Winters model can provide more accurate predictions, which means better inventory management, optimized sales strategies, and effective resource allocation.

**Solving Challenges & Exploiting Opportunities:**

* Accurate predictions can help businesses anticipate sales, adjust marketing strategies, optimize supply chain processes, and better understand customer behavior.

**Quantifying Improvements:**

* Considering the LightGBM model's R^2 of 97.97% compared to Linear Regression's 72.35%, it can result in significantly better predictions. For a business, this could translate to potential savings or increased profits in the order of several percentages depending on the use case.
* Similarly, in forecasting, using the Holt-Winters method could mean fewer stockouts or overstock situations, optimizing inventory holding costs, and maximizing sales opportunities.

## Data Privacy and Ethical Concerns

**Privacy concerns:**

* Sensitive Info: Ensure no personal details or confidential data are exposed.
* Storage/Transfer: Secure environments and encrypted channels are a must.
* Third-party Risks: Vigilance required when using external tools.

**Ethical Dilemmas:**

* Bias: Ensure datasets are representative to avoid perpetuating biases.
* Transparency: Avoid "black box" models; ensure decisions are explainable.
* Accountability & Usage: Clearly define responsibility and avoid blind reliance on automated predictions.

**Steps for Safety & Ethics:**

* Anonymize Data: Mask or pseudonymize personal info.
* Encrypt: Always.
* Minimize Data Use: Only use what's necessary.
* Bias Checks: Regularly validate models for fairness.
* Boost Transparency: Leverage explainability tools.
* Monitor: Ensure post-deployment models remain ethical.
* Engage & Educate: Engage with stakeholders and train the team on ethical guidelines.

short dash

# Deployment

**1. Creation of Model Artifacts using Joblib:**

* Using Joblib: Instead of the more general pickle, joblib is a more efficient tool for serializing large datasets or models, like those in scikit-learn. After training your best model, it's serialized to a file using:

from joblib import dump

dump(model, 'model\_artifact.joblib')

**2. Dockerizing the Model:**

* **Setup**: Organize your FastAPI app and its dependencies.
* **Dockerfile:** Script commands to assemble a Docker image, including app files and dependencies. We need to make sure that our required packages are mentioned in the requirements.txt so that the container is built smoothly without throwing any errors.
* **Build Image:** Use Docker to create an image from the Dockerfile: docker build.
* **Run Container:** Start your app in a Docker container using docker run.

**NOTE: Please make sure to check the version of your local computer and mention the same version in the requirements file.**

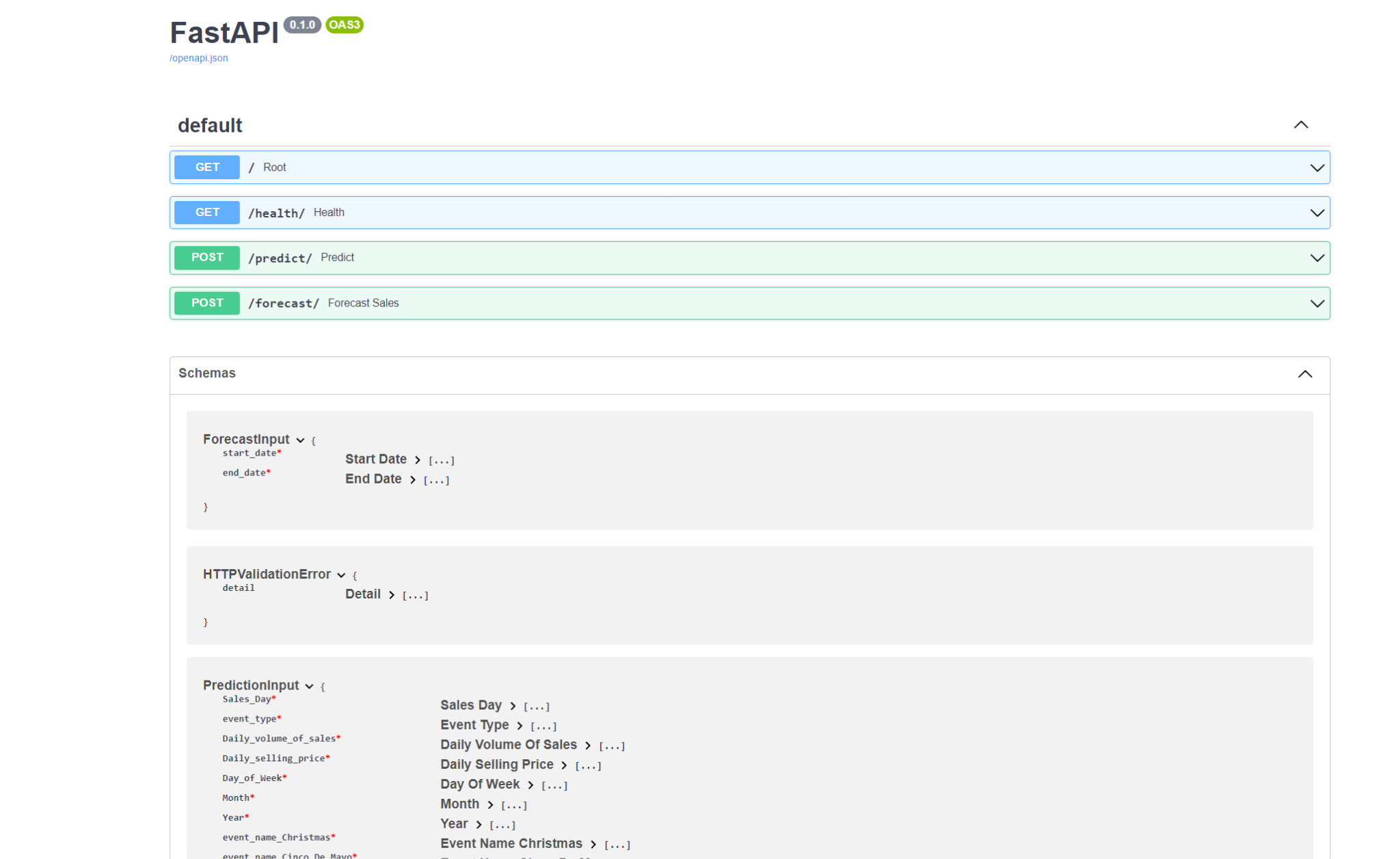
**3. Building the API using FastAPI:**

* **Framework:** Adopted FastAPI, a swift tool for creating web interfaces.
* **Model Integration:** Integrated trained sales and forecasting models**.**
* **Data Structures:** Established expected input formats for sales details and forecast date ranges.
* **API Functions:** Created two functions: predict (for sales prediction) and forecast (for revenue outlook).
* **Deployment Setup:** Readied the tool to operate on a standard web-accessible port.
* **Outcome:** A user-friendly platform providing immediate sales insights and revenue predictions.

**4. Testing Locally:**

**Running the Container:** After building the docker container followed by docker run, start the containerized app locally to ensure everything's working. This way, before deploying to a wider audience, you can make sure the model responds correctly to API requests.

You can check the deployment locally after running the docker container:  
 **http://localhost:8000/docs#/default/forecast\_sales\_forecast\_\_post**



**Fig: API of the models**

**5. Deploying on Heroku:**

* **Git Commit:** Commit your changes to a Git repository.
* **Heroku Setup:** Install Heroku CLI, login, and create a new Heroku app.
* **Push to Heroku:** Use heroku container:push or git push heroku main to send your Docker container/models to Heroku.
* **Release:** Deploy it using heroku container:release.
* **Open App:** Access your FastAPI app via the provided Heroku URL.

**Challenges and Considerations:**

* Consistency: One main advantage of Docker is consistency. The same image that runs locally will run in production.
* Model Size: If your model artifact is large, you may run into deployment size limits on platforms like Heroku.
* Scalability: As more users access your model, you'll need to ensure your app scales. Heroku's dynos can help here, but costs can rise.
* Version Control: When updating the model or tweaking the API, ensure you have a solid versioning strategy in place.

**Recommendations:**

* Regularly monitor the model's performance and retrain if necessary.
* Consider using cloud platforms for more intensive apps that might outgrow platforms like Heroku.
* Always keep user data privacy in mind, especially when processing requests through your API.

short dash

# Conclusion

**Key Findings and Insights:**

* LightGBM emerged as the star performer, surpassing even the highly esteemed Random Forest in predictive accuracy. Its ability to capture intricate data patterns resulted in an impressive R^2 of 97.96%.
* Holt-Winters showcased its prowess in forecasting, hinting at significant underlying seasonality and trends in the dataset.

**Reflection on Success:**

The project not only met but likely surpassed stakeholder expectations. By leveraging the strengths of LightGBM, the project laid a robust foundation for informed, data-centric business decisions.

**Future Recommendations:**

* To maintain the model's high accuracy, monitor data drifts and periodically update and retrain the LightGBM model.
* Dive deeper into feature engineering and selection to potentially further elevate model performance.
* As the model gets ingrained in critical decision-making, consider implementing more resilient deployment and monitoring mechanisms to ensure consistent and reliable performance.

short dash

# References

So Anthony(2023) 36120\_AdvMLA\_Lab4\_Exercise3\_Solutions [Google Colab notebook]. Google Colab. https://colab.research.google.com/drive/1Qv7MzKCzDMskH2kR\_qletXwZLLyzFrts?authuser=2