Assignment 2
ML as a Service

Student Name : Yasaman Mohammadi
Student ID:24612626

Date: 06/10/2023
36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Executive Summary

Project Overview:

The project involves developing two essential models for an American retailer operating ten stores across three different states: California (CA), Texas (TX), and Wisconsin (WI). The retailer sells items from three main categories: hobbies, foods, and household. The objective of this project is to build two predictive models to enhance business operations:

1. **Predictive Sales Revenue Model:**

   Objective: To accurately predict the sales revenue for a specific item in a particular store on a given date.

   Significance: This model is crucial for optimizing inventory management, pricing strategies, and overall store performance. It enables the retailer to make data-driven decisions, reduce wastage, and ensure the availability of popular items.

2. **Forecasting Model:**

   Objective: To forecast the total sales revenue across all stores and items for the next seven days using time-series analysis.

   Significance: This model aids demand forecasting, inventory planning, and budgeting. It allows the retailer to anticipate sales trends, allocate resources efficiently, and respond proactively to market changes.

Problem Statement and Context:

The problem statement revolves around improving the retailer's business operations by harnessing the power of data and advanced analytics. The context is as follows:

- The retailer operates in three states with distinct market characteristics and consumer preferences.
- They offer a wide range of products spanning three categories, which introduces sales forecasting and management complexity.
- Accurate sales revenue prediction at the item and store level is essential for optimizing inventory and pricing.
- The 7-day sales revenue forecast across all stores and items enables proactive decision-making and resource allocation.

2.Business Understanding

2.1. Business Use Cases:

The project applies machine learning models to address specific business needs in retail:

1. **Inventory Optimization:** Predictive models ensure items are stocked appropriately, reducing costs and preventing stockouts.
2. **Pricing Strategy:** Accurate revenue predictions inform pricing decisions, maintaining competitiveness.
3. **Demand Forecasting:** Forecasts help allocate resources efficiently and manage stock levels.
4. **Marketing Effectiveness:** Models evaluate marketing campaigns, optimizing budget allocation.

2.2 Challenges:

1. **Fluctuating Demand:** Customer behaviour varies, making it challenging to predict sales accurately.
2. **Inventory Costs:** Maintaining the right stock levels is crucial; excess inventory ties up capital, while inadequate inventory leads to lost sales.
3. **Complex Data:** Retail data is vast and intricate, with multiple variables affecting sales.

2.3 Opportunities:

1. **Real-time Decisions:** Deploying models as APIs enables quick responses to market shifts, enhancing agility.
2. **Customer Experience:** Accurate pricing and inventory management improve customer satisfaction and loyalty.
3. **Resource Allocation:** Efficient resource allocation, guided by forecasts, ensures profitability and cost savings.

2.4 Key Objectives:

1. **Sales Revenue Prediction:** The primary objective is to accurately predict sales revenue for specific items in each store on any given date. This involves item-level and store-level predictions.
2. **Sales Forecasting:** The project aims to forecast the total sales revenue across all stores and items for the next seven days, providing a forward-looking view of sales trends.

2.5 Stakeholders and Their Requirements:

1. **Retailer:** Accurate and timely sales revenue predictions to optimize inventory, pricing, and resource allocation.

2. **Store Managers:** Item-specific and store-specific sales revenue predictions to manage inventory levels effectively and make informed decisions.

3. **Marketing Team:** Insights into the effectiveness of marketing campaigns and promotions on sales revenue to allocate marketing budgets efficiently.

4. **Finance Department:** Accurate sales forecasts to budget effectively and allocate resources efficiently.

5. **Customers:** Availability of products, competitive pricing, and a positive shopping experience.

2.6 Addressing Stakeholder Requirements:

1. **Sales Revenue Prediction:** By building a predictive model, machine learning could predict each store's sales revenue. This way, retailers can optimize inventory, pricing, and resource allocation.

2. **Sales Forecasting:**We can address the finance department's need for accurate sales forecasts by developing a time-series forecasting model for predicting total sales revenue across all stores and items for the next seven days.

3. **Operational Efficiency:**Deployment as APIs: Deploy both models to provide real-time predictions and insights. This addresses the need for timely information and enhances operational efficiency for store managers, marketing teams, and the retailer.

4. **Customer Satisfaction:**

   - Inventory Management: Accurate sales revenue prediction ensures product availability, enhancing the customer experience.
   - Pricing Optimization: The pricing model helps maintain competitive pricing, contributing to customer satisfaction.

2.8 Tools and technologies for this project:

- Python (EDA and model development)
- Jupyter notebook
- GitHub
- Tableau
- Docker
- Heroku

3.Data Understanding

1. Collect initial data:

Data was gathered from the UTS Canvas platform and presented in tabular format. Five distinct datasets were utilized: Training Data, Evaluation Data, Calendar, Events, and Weekly Item Prices. We successfully generated the final table displayed below by combining these datasets, excluding the Evaluation dataset.

2. Data description:

The combined data frame comprises 47,107,050 columns, each associated with 16 variables. This final dataset was formed by amalgamating four CSV files: Training, Calendar, and Calender Events.:

Table 1-The description of each column/variable

id': A unique identifier for each row or record.
'item_id': Identifier for the specific item being tracked.
'dept_id': The department or category to which the item belongs.
'cat_id': The broader category to which the item belongs.
'store_id': Identifier for the store where the item is sold.
'state_id': Identifier for the state in which the store is located.
'day': The day of the month for the recorded data.
'count': The quantity of items sold on a particular day.
'date': The date corresponding to the recorded data.
'wm_yr_wk': The year and week in which the data was recorded, following the Walmart week format.
'event_name': The name of any special event or occasion associated with the recorded data.
'event_type': The type or category of the special event mentioned in 'event_name'.
'sell_price': The selling price of the item on a given day.
'revenue': The total revenue generated from the sales of the item on a given day.
'year': The year of the recorded data.
'month': The month of the recorded data.

Dataset includes categorical and numerical features. Each column data type can be seen below:

```
Data columns (total 16 columns):
 #   Column      Dtype
---  ------      -----
 0   id          object
 1   item_id     object
 2   dept_id     object
 3   cat_id      object
 4   store_id    object
 5   state_id    object
 6   day         int64
 7   count       int64
 8   date        object
 9   wm_yr_wk    int64
 10  event_name  object
 11  event_type  object
 12  sell_price  float64
 13  revenue     float64
 14  year        int64
 15  month       int64
```

Figure 1-The data type of each column

3. EDA:

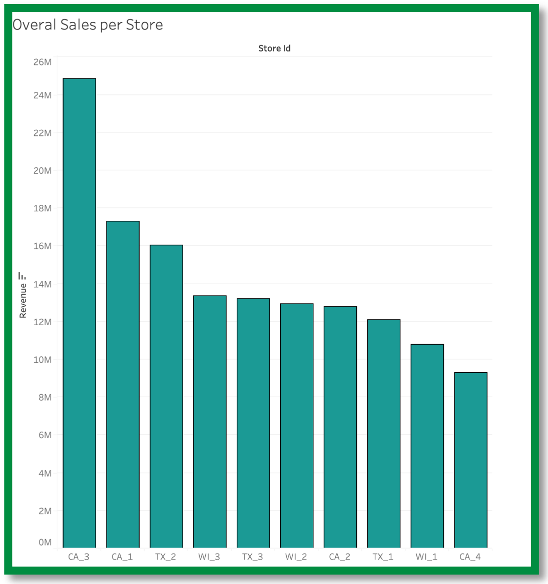

Figure 2- Overal sales per store

The bar chart distinctly reveals that CA_3 store boasts the highest sales, while CA_4 lags behind with the lowest sales. In contrast, stores in Texas (TX) and Wisconsin (WI) exhibit comparable sales levels within their respective state categories.

A declining trend appears evident among the CA stores:

Figure 3-Sales of store in CA state

Tx_1 and TX_3 saw a decline in revenue starting from 2014, while TX_2 experienced a decrease from 2013:

Figure 4-Sales of stores in TX state

WI_1 and WI_3 witnessed a revenue decrease beginning in 2014, whereas WI_2 has declined since 2012:

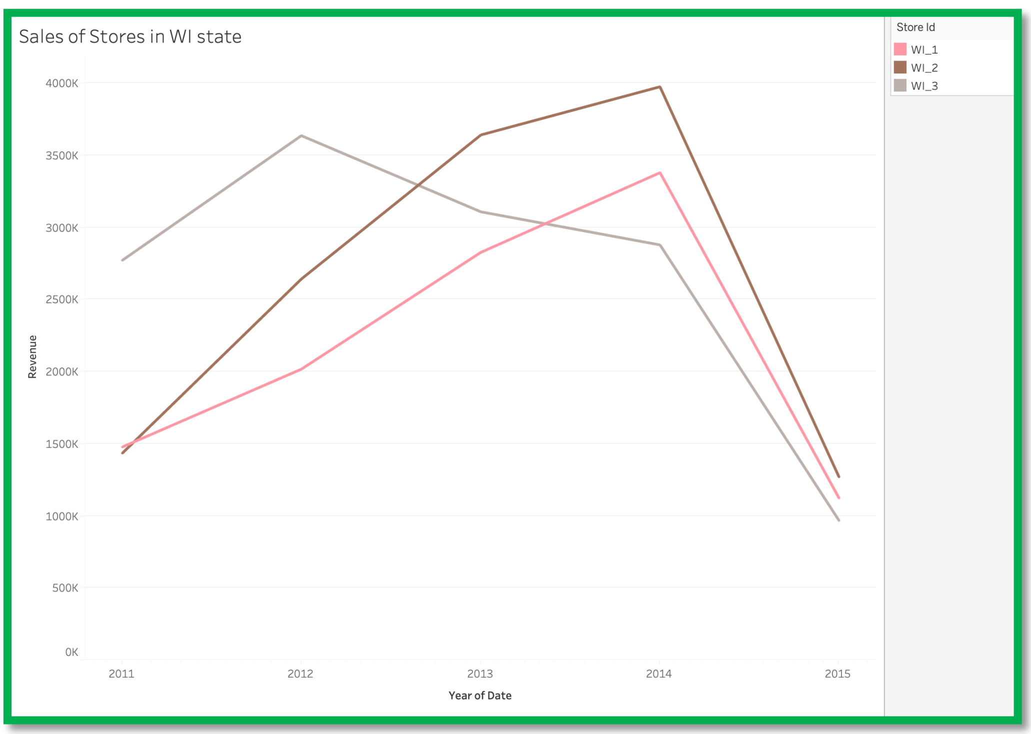

Figure 5-Sales of stores in WI state

Most revenue is derived from the food category, while hobbies contribute the least.

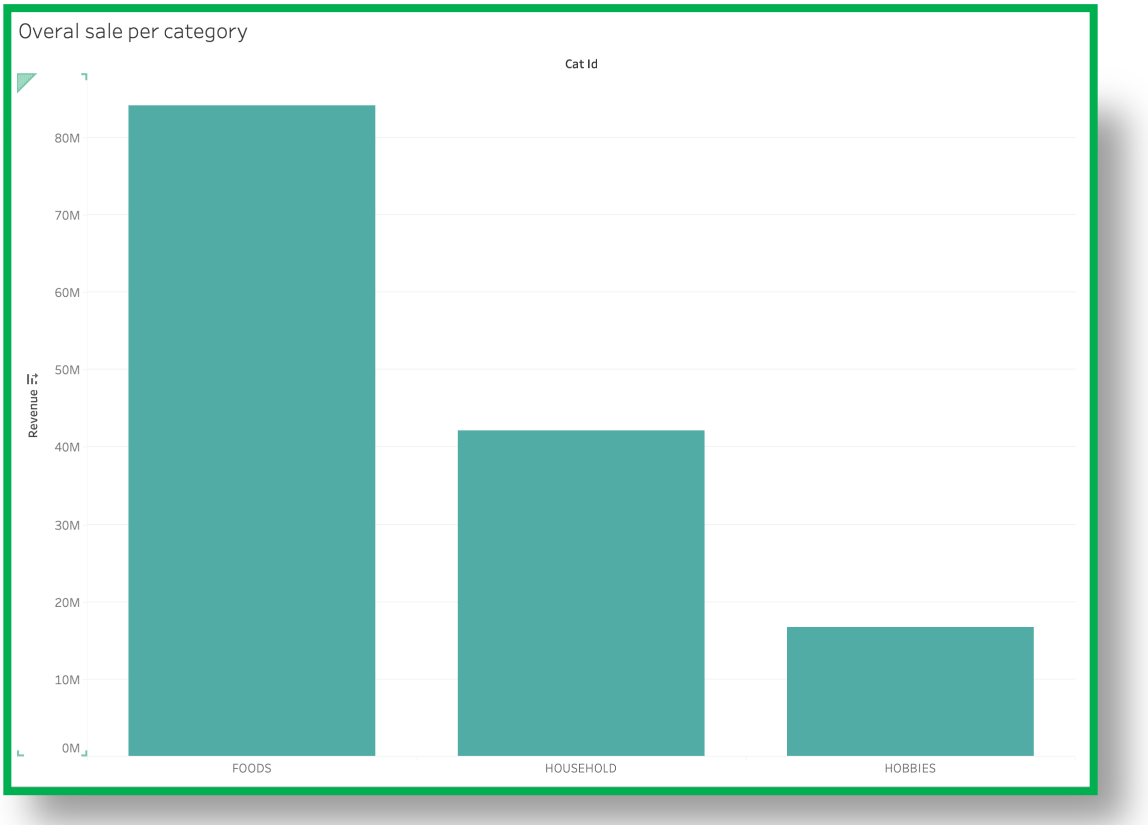

Figure 6-Overal sales per category

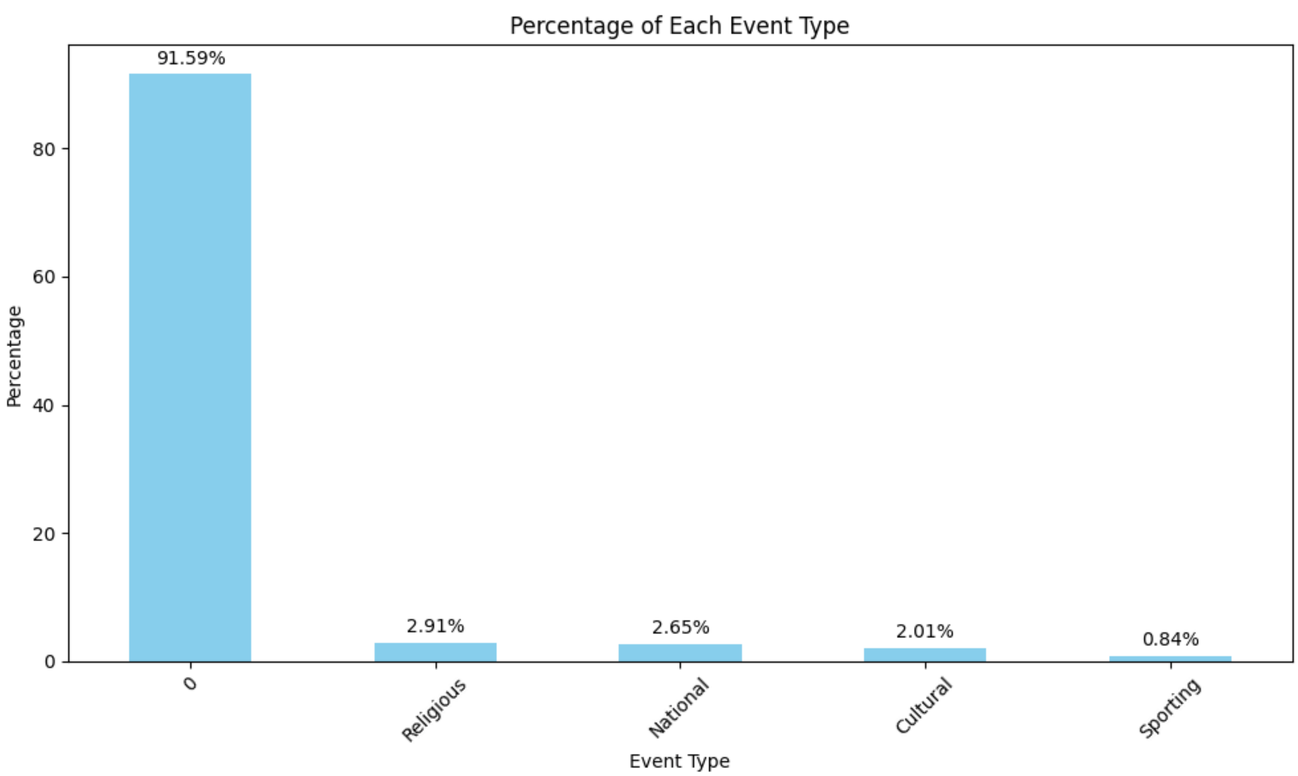

Figure 7-Percent of each event type

Most dates have no scheduled events, and among those with events, religious ones are the most prevalent.

2. Data Preparation

Initially, we had four distinct CSV files, each containing specific information. We amalgamated these CSV files into a final dataset for our analysis to enhance our understanding and achieve more accurate results. During this consolidation process, we introduced a new column in our data, "revenue," by utilizing the count and selling price, aligning with our project's primary focus on revenue-related objectives.

Table 1-Head of sales train table

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...	d_1532	d_1533	d_1534	d_1535	d_1
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1	1	1	0	
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0	0	0	0	
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0	0	1	0	
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	8	2	0	8	
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	0	1	3	

Table 2-Head of calendar events table

	date	event_name	event_type
0	2011-02-06	SuperBowl	Sporting
1	2011-02-14	ValentinesDay	Cultural
2	2011-02-21	PresidentsDay	National
3	2011-03-09	LentStart	Religious
4	2011-03-16	LentWeek2	Religious

Table 3-Head of calendar table

	date	wm_yr_wk	d
0	2011-01-29	11101	d_1
1	2011-01-30	11101	d_2
2	2011-01-31	11101	d_3
3	2011-02-01	11101	d_4
4	2011-02-02	11101	d_5

Table 4-Final table(merged_df)

id	item_id	dept_id	cat_id	store_id	state_id	day	count	date	wm_yr_wk	event_name	event_type	sell_price	revenue	year	month
valuation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
valuation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
valuation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
valuation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
valuation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1

- **Memory optimization**

  Due to the enormity of the final dataset, attempting to process it resulted in kernel crashes. To address this issue, we undertook memory optimization, a pivotal technique for efficiently handling and conserving a data frame's memory resources. This approach yields substantial advantages when dealing with sizable datasets, enhancing performance while minimizing memory consumption.

- **Data Cleaning**

  We utilized a missingno plot to understand the dataset's missing values better. In our analysis, we observed that only the "sell_price" and "revenue" columns contained missing values, and as a corrective measure, we imputed these missing values with zeros.

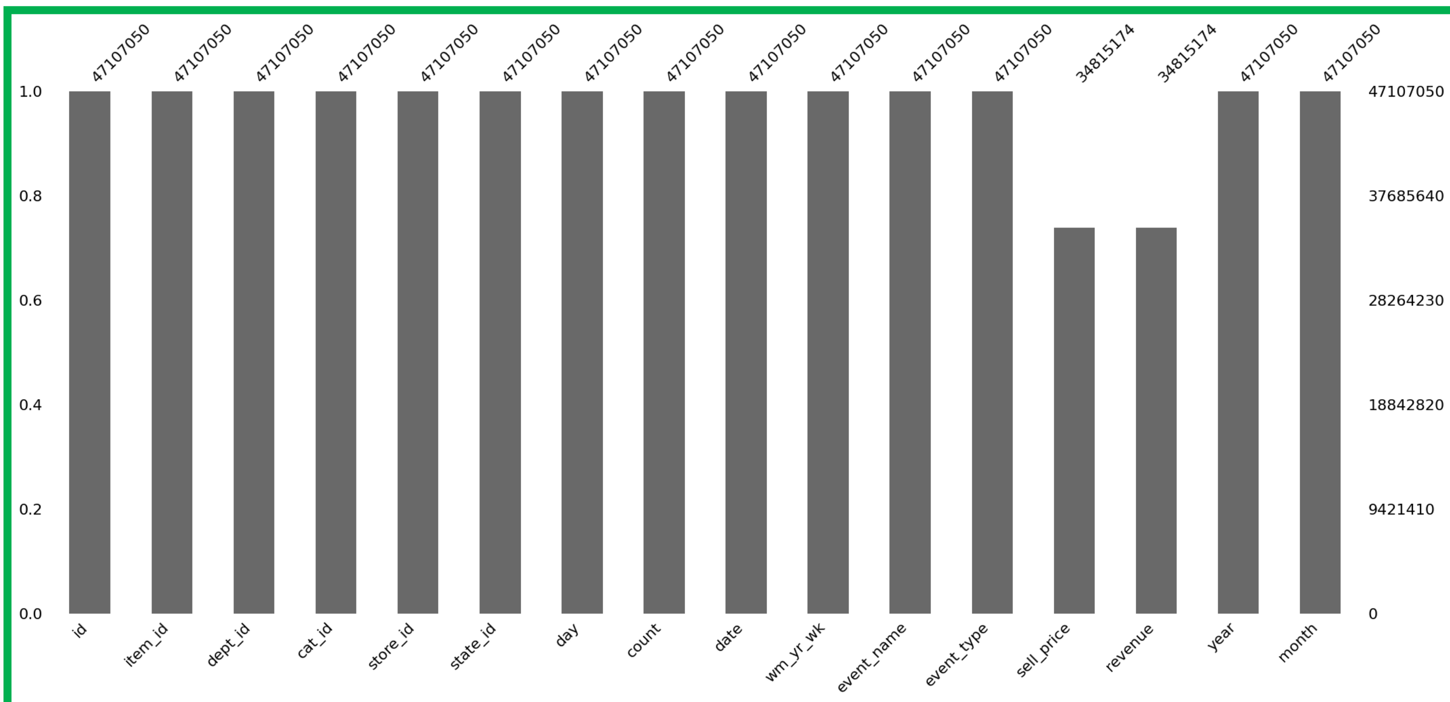

Figure 8-The mnso chart is used to identify missing values.

It is worth mentioning that in our second approach, which focused on the forecasting model, we organized our dataset by grouping it according to date and revenue. This grouping ensured that each date encapsulated the total revenue from all stores and states. As a result, our final dataset for this approach consisted of only two columns.

- **Feature engineering (Predicting Model)**

Binary encoding:

Given the multitude of unique attributes in the event type and name categories, we employed binary encoding as an alternative to one-hot encoding. Binary encoding, a technique that converts categorical data into a binary format of 0s and 1s, offers distinct advantages in machine learning by reducing data dimensionality while retaining critical information. It is worth noting that label encoding, another option for numerical representation, was considered but deemed unsuitable due to the non-ordinal nature of these columns.

Lags:
Lag calculation is a method that involves analysing historical sales data to determine if it exerts any impact on current-day sales. This approach allows for an investigation into whether there exists a connection between the sales figures from the recent past and those of today in a retail store.

It's worth noting that we removed certain columns, including "id," "state_id," "dept_id," "sell_price," and "wm_yr_wk," as their information was redundant and already present in other columns.

Memory-Friendly Machine Learning Subset:
Given the substantial dataset resulting from extensive feature engineering, we opted to employ machine learning on a more manageable subset, explicitly focusing on the data from the last two years to avoid kernel crashes or performance issues.

Splitting the dataset:
The dataset was divided into two subsets: a training set, utilized to train the model, and a test set, employed to assess the model's final performance. The test size parameter was set to 20%, indicating that the last 20% of the dataset was designated as the test data.

3. Modeling

a. *Approach 1*

- **LightGBM regressor.**

  Due to the importance of specific categorical features, namely item_id and store_id, in our dataset, we strategically chose to utilize the LightGBM model. LightGBM is particularly well-suited for handling categorical variables efficiently and effectively. Its gradient boosting framework, designed to handle categorical data, provides speed and predictive accuracy advantages.
  This decision enables us to capitalize on the strengths of LightGBM, ensuring that we can leverage the categorical information in our dataset to improve the model's overall performance and predictive power. By doing so, we aim to obtain more accurate and insightful results considering the nuances of item and store characteristics.

- ***Feature Engineering***

  In this experiment, we employed binary encoding to transform two categorical features into numerical ones. Additionally, we conducted lag calculations to explore potential correlations between sales figures from the recent past and those of the current day within a retail store.

- ***K-fold cross-validation***

  K-fold cross-validation with five folds is employed in the code to evaluate and fine-tune the LightGBM regressor's hyperparameters robustly. This technique ensures that the model's performance is thoroughly assessed across multiple data subsets, reducing the risk of overfitting and providing a more reliable estimate of its effectiveness. GridSearchCV systematically explores hyperparameter combinations, such as the number of estimators ("n_estimators"), using the "neg_mean_absolute_error" metric, resulting in a well-tuned and generalized model.

- ***Hyperparameters:***

N estimator

This is the number of boosting rounds or trees trained in the LightGBM model. It is a crucial hyperparameter because it determines how many iterations the model will go through to learn from the data. Increasing the value of "n_estimators" generally allows the model to learn more complex patterns in the data, but it also makes the training process slower and may risk overfitting if set too high.

With grid search, a n estimator of 250 to 450 was selected for this experiment, and 450 was determined to be the best value.

b. Approach 2

In both this approach and the subsequent one, centered around the forecasting model, we structured our dataset by grouping it based on date and revenue. This arrangement guaranteed that each date encompassed the total revenue across all stores and states. Consequently, our final dataset for these approaches comprised only two columns.

- **Autoregressive integrated moving average (ARIMA)**
For this experiment, as we require time series models, we used ARIMA model with hyperparameter configurations.

- **Hyperparameters:**

  - **d**

Initially, we conducted a stationarity test using the Augmented Dickey-Fuller test. When the p-value exceeded 0.05, it indicated that the time series was non-stationary.

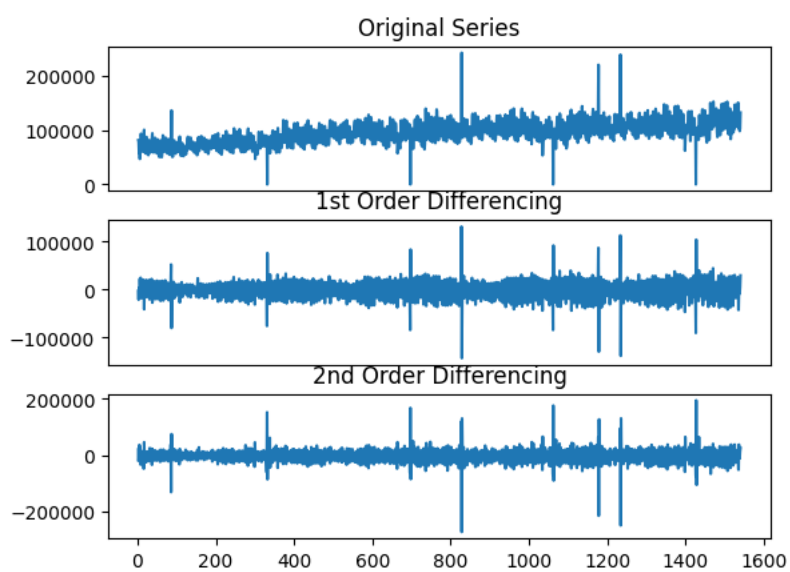

Figure 9-Differencing test

We employed differencing to determine the appropriate value for 'd.' The results indicated that the time series became stationary after the first differencing, confirming that 'd' should be set to 1.

- **p**

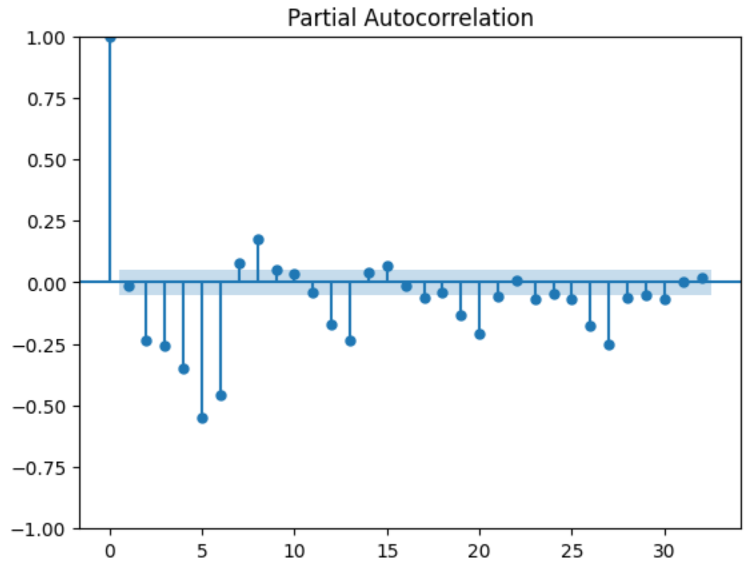

Figure 10-Partial Autocorrelation plot

The partial autocorrelation function (PACF) plot helps establish correlations between a time series and its lags, with the influence of intermediate lags being disregarded. This visualization allows us to identify which lags are unnecessary for the autoregressive component, helping determine the model's appropriate value for 'p'.
This analysis shows that the magnitude of the first lag surpasses the established limit by a considerable margin, while the second lag remains within the acceptable range. Therefore, we can confidently set the order of 'p' to 1.

- **q**

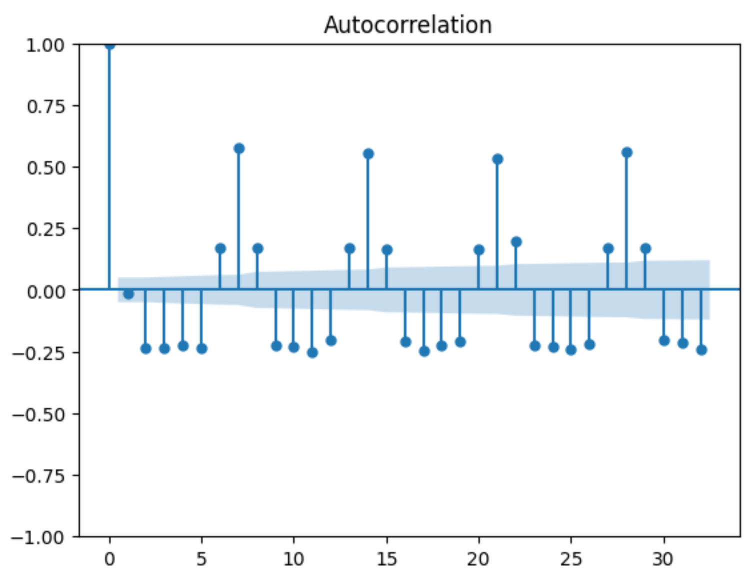

Figure 11-Autocorrelation plot

To determine the value of 'q,' we can turn to the Autocorrelation Function (ACF) plot, which provides insights into the extent of the moving average necessary to eliminate autocorrelation in a stationary time series. Here we can see that 4 of the lags are out of the significance limit so we can say that the optimal value of our q (MA) is 4.

c. Approach 3

- **Seasonal Auto-Regressive Integrated Moving Average**

SARIMA, which stands for Seasonal Auto-Regressive Integrated Moving Average, is an enhanced version of the ARIMA (Autoregressive Integrated Moving Average) model. It goes beyond capturing non-seasonal components and also accounts for seasonality. While ARIMA models are a common choice for time series analysis and forecasting, SARIMA models are tailored to handle data exhibiting seasonal patterns effectively.

Apart from the hyperparameters discussed in the previous section, we included the following in this model:

- **P (Seasonal AutoRegressive Order):** P determines how many past seasonal data points are considered when predicting the current season's value. A higher P captures longer-term seasonal trends but increases model complexity. In the analysis, the optimal P value was found to be 4.
- **Q (Seasonal Moving Average Order):** Q dictates how many preceding forecast errors are used to refine the current prediction with seasonality. A higher Q captures persistent error patterns but also increases model complexity. The optimal Q value in the analysis was 3.
- **D (Differencing Order):** D indicates how often differencing is applied to make the time series stationary. A higher D can remove trends and seasonality but should be used cautiously to avoid information loss. The optimal D value in the analysis was 1.
- **m (Seasonal Period):** "m" represents the number of time steps per seasonal cycle within the data. It defines how often seasonality patterns repeat in the time series. In this analysis, "m" was set to 12 to align with yearly patterns in the data, common for monthly data exhibiting annual seasons.

6. Evaluation

a. Evaluation Metrics

In the project, RMSE (Root Mean Square Error) is used as the evaluation metric for assessing the performance of both the Predictive Sales Revenue and Forecasting Model.
RMSE is suitable for evaluating model performance in this context as it quantifies the average magnitude of errors between predicted values and actual sales revenue. It provides a measure of accuracy and aligns with the project's goals by helping to understand how well the models perform regarding revenue prediction and forecasting.

By employing grid search for the predictive model, it automatically selected hyperparameters that resulted in lower RMSE values on the dataset.

b. Results and Analysis

- *Predictive model:*

Initially, we attempted to utilize a more extensive feature set and the entire dataset for training. However, this approach caused our kernel to crash. Consequently, we adjusted our strategy by using a subset of the data, explicitly the last two years, to ensure all events were included in our training data. This modification resulted in our model functioning smoothly.

Final RMSE on test set:

LGB regressor: 0.77036

- *Forecasting model:*

  As we transformed our data for both models by aggregating daily revenues across all stores, our dataset significantly reduced size, containing only 1541 entries. ARIMA outperformed SARIMA in this context because of the added complexity introduced by SARIMA's seasonal components (P, Q, D) on top of the ARIMA model. When dealing with a relatively short or noisy dataset, including seasonal parameters in SARIMA can increase the risk of overfitting. In contrast, ARIMA's exclusive focus on non-seasonal patterns allowed it to handle the data more effectively.

  Final RMSE on test set:

  ARIMA: 20511.13
  SARIMA: 22406.99

c. *Business Impacts and benefits*

Accurate predictive models are crucial for forecasting sales revenue in multiple stores across different states, as they optimize inventory, pricing strategies, and customer satisfaction. Incorrect predictions can lead to overstocking, stockouts, and financial losses. The forecasting models produced high RMSE values, highlighting the need for more robust models to avoid inventory and financial issues, maintain competitiveness, and ensure customer satisfaction.

d. Data privacy and concerns

1. **Data Collection**: Collecting sales data, especially when it involves transactional details, could potentially contain sensitive customer information, such as purchase history, payment details, and personal identification. This data should be handled with care to avoid any privacy breaches.
2. **Data Usage**: Analyzing sales data can reveal customer preferences and behaviours. If not managed properly, this information can be misused, leading to privacy concerns or even targeted marketing that customers may find intrusive.
3. **Model Deployment**: Deploying predictive models in a real-world setting might involve integrating them with existing systems. This integration must be carefully managed to ensure data privacy and security, mainly if personally identifiable information (PII) concerns exist.

Ethical Concerns:

1. **Bias and Fairness**: There is a risk of introducing bias into the models, leading to unfair outcomes. For example, if historical data contains biases in pricing or inventory management decisions, the models might perpetuate these biases, affecting certain customer groups unfairly.

2. **Consent and Privacy**: If customer data is used in any way, obtaining proper consent and ensuring data privacy is paramount. The retailer must have clear data collection, usage, and retention policies.

Steps Taken to Ensure Data Privacy and Ethical Considerations:

1. **Data Anonymization**: To protect customer privacy, any personally identifiable information (PII) should be anonymized or de-identified during data collection and storage.
2. **Privacy Policies**: Develop and enforce strong data privacy policies and ensure customers are informed about how their data is used, with clear opt-in/opt-out mechanisms.
3. **Fairness Assessment**: Regularly assess the models for bias and fairness. Adjust the models and data if any unfair biases are identified.
4. **Security Measures**: Implement robust security measures to protect customer data throughout its lifecycle, including encryption, access controls, and secure model deployment.
5. **Ethical Review**: Conduct an ethical review of the project's objectives and outcomes to proactively identify and address potential ethical concerns.
6. **Continuous Monitoring**: Continuously monitor and update the project to adapt to changing ethical and privacy considerations and evolving best practices.

7. Deployment

In our Git repository, we have established a "FastAPI" directory. Within this directory, we have included essential files like "requirements.txt," "Dockerfile," and "Docker-compose.yml."

The "main.py" file plays a pivotal role in our application's heart. Here, we have loaded our machine-learning models and meticulously designed and defined various endpoints. These endpoints serve as the interface through which our application interacts with users and processes incoming requests. In this file, the magic happens as our models are put to work, making predictions and handling data as needed.

Once the Docker image was built, we tested its functionality locally using localhost. With a successful deployment, our application is now live and accessible on Heroku.

To ensure your application functions effectively in the real world, it's crucial to proactively monitor its performance and be ready to scale it when required. Equally vital is addressing security, guarding your application against prevalent vulnerabilities.

8.Conclusion

This project offers substantial benefits for various business use cases. However, it is advisable to consider deploying even more robust forecasting models to achieve greater accuracy in results.

GitHub repository:

https://github.com/JYasimo/Machine_Learning_as_a_Service/tree/main

API url:

https://mysterious-ocean-12629-005fa33ae82b.herokuapp.com/

Fast API:

https://mysterious-ocean-12629-005fa33ae82b.herokuapp.com/docs

7.References

1. Article on Time Series and ARIMA:
   Author(s): Nagesh Singh Chauhan
   Title: Machine Learning Part 19: Time Series and Autoregressive Integrated Moving Average Model (ARIMA)
   URL: https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7

2. Stack Overflow Question:
   Title: ValueError: A given column is not a column of the DataFrame
   URL: https://stackoverflow.com/questions/65418378/valueerror-a-given-column-is-not-a-column-of-the-dataframe

3. Medium Article on Machine Learning Pipeline:
   Author(s): Mlearning AI
   Title: Get Column Name After Fitting the Machine Learning Pipeline
   URL: https://medium.com/mlearning-ai/get-column-name-after-fitting-the-machine-learning-pipeline-145a2a8051cc

4. Medium Article on Data Preprocessing:
   Author(s): Mlearning AI
   Title: Neat Data Preprocessing with Pipeline and ColumnTransformer
   URL: https://medium.com/mlearning-ai/neat-data-preprocessing-with-pipeline-and-columntransformer-2a0468865b6b

5. Kaggle Notebook:
   Author(s): Pourya Ayria
   Title: A Complete ML Pipeline Tutorial (Accuracy: 86%)
   URL: https://www.kaggle.com/code/pouryaayria/a-complete-ml-pipeline-tutorial-acu-86#6.3-Stacking

6. Machine Learning Mastery Article on Time Series Forecasting:
   Author(s): Jason Brownlee
   Title: Time Series Forecasting with Prophet in Python
   URL: https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/

7. Apache Spark Documentation on ML Pipeline:
   Title: Apache Spark ML Pipeline
   URL: https://spark.apache.org/docs/latest/ml-pipeline.html

8. PyPI Package:
   Title: Downcast
   URL: https://pypi.org/project/downcast/

9. Analytics India Magazine Article on ARIMA:
   Title: Quick Way to Find p, d, and q Values for ARIMA
   URL: https://analyticsindiamag.com/quick-way-to-find-p-d-and-q-values-for-arima/

10. Medium Article on SARIMA Hyperparameter Tuning:

Author(s): Analytics Vidhya
Title: Fine-Tune SARIMA Hyperparams Using Parallel Processing with Joblib: Step-by-Step Python Code
URL: https://medium.com/analytics-vidhya/fine-tune-sarima-hyperparams-using-parallel-processing-with-joblib-step-by-step-python-code-2037fec1659