EXPERIMENT REPORT

Student Name : Yasaman Mohammadi
Project Name: Machine Learning as service

Date:06/10/2023
Deliverables:
notebook name: Mohammadi_Yasaman_24612626-forcasting_ARIMA.ipynb

1.EXPERIMENT BACKGROUND

1.a. Business Objective

This project aims to build and deploy two predictive models as APIs for an American retailer with ten stores across three states. These models will forecast total sales revenue for the next seven days, aiding inventory management, staffing, and marketing decisions.

Performance Metrics:

Root Mean Squared Error (RMSE)

Use of Results:

- Optimize inventory levels.
- Efficiently schedule staff.
- Tailor marketing strategies based on forecasts.

Impact of Accurate/Incorrect Results:

Accurate results lead to cost savings, improved customer experience, and increased revenue. Incorrect results can result in overstocking, understocking, or inefficient staffing, negatively affecting the business's efficiency and profitability.

1.b. Hypothesis

- Can analyzing historical sales data help identify trends and seasonality patterns, ultimately enabling better inventory planning and more effective targeted marketing efforts?

  Effective inventory management can minimize stockouts (when products are unavailable when customers want to buy) and overstock situations (excess inventory leading to storage costs and potential losses due to obsolescence). Analyzing historical sales data can help maintain optimal stock levels, thus improving customer satisfaction and reducing costs.

1.c. Experiment Objective

The project's overarching objective is to develop predictive models for sales revenue forecasting in an American retailer, ultimately delivering highly accurate and dependable forecasts for the upcoming seven days. The primary focus is on bolstering decision-making across inventory management, staffing, and marketing strategies. While specific numerical targets may be contingent on the retailer's historical data, the overarching aim remains consistent: to enhance operational efficiency and boost profitability. The project assesses model performance using the Root Mean Square Error (RMSE), where lower RMSE values indicate superior model performance.

Possible Scenarios for Successful Implementation:

- **Inventory Cost Optimization:** Effective implementation can decrease expenses associated with inventory management by minimizing overstocking and understocking situations, ultimately translating into significant cost savings for the retailer.

- **Strategic Marketing Effectiveness:** Precise sales forecasts can empower retailers to strategize and execute more impactful marketing campaigns, potentially enhancing their marketing initiatives' return on investment (ROI).

Potential Outcomes of Unsuccessful Implementation:

- **Inaccurate Forecasts:** If the predictive models fail to perform effectively, the retailer may continue to grapple with inaccurate sales forecasts, potentially resulting in inventory imbalances and operational difficulties.

- **Competitive Disadvantage:** Within a fiercely competitive retail environment, the inability to harness precise forecasts may leave the retailer disadvantaged compared to competitors who can more adeptly respond to market fluctuations.

2.EXPERIMENT DETAILS

2.a. Data Preparation

	the steps taken for preparing the data	Explanation
1	Data collection	The process of constructing a machine learning model commences with gathering data from diverse sources. In this instance, the data was obtained from Canvas UTS, consisting of four separate CSV files. Then they merged.
2	Memory optimization	Memory optimization is a crucial technique for effectively managing and preserving the memory resources of a data frame, offering significant benefits when working with large datasets. It boosts performance while keeping memory usage to a minimum.
3	Explore Dataset	A general understanding is derived from the data in this step.
4	Data cleaning	Data was preprocessed to remove any irrelevant, missing, or corrupted data points and any discrepancies in the data.
5	Data visualization	Visual understanding of data distribution is gained by plotting different charts in both Python and Tableau.
6	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.

Table 1- the steps taken for preparation of the data.

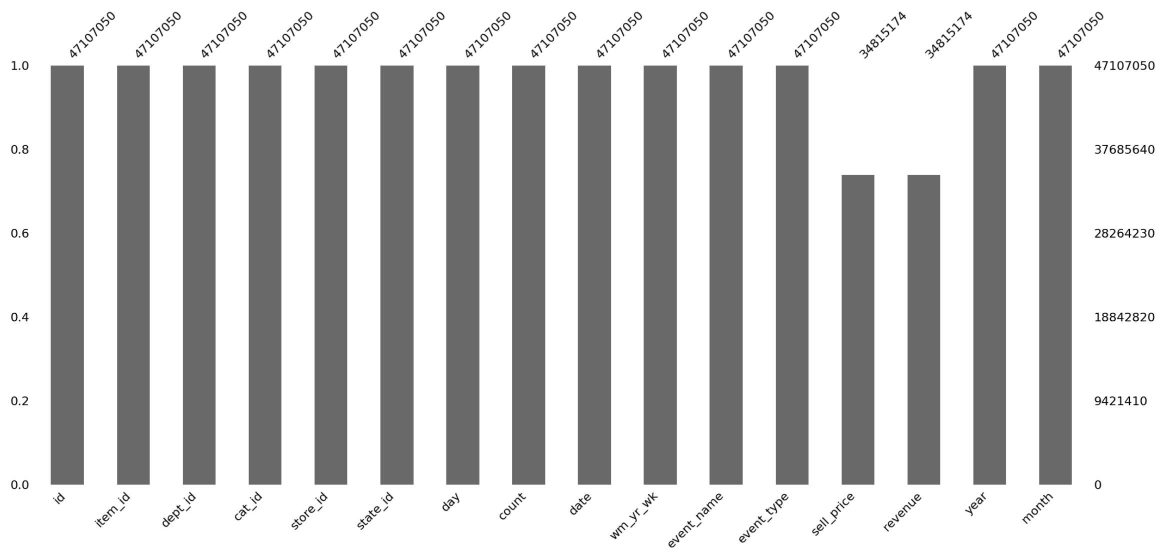
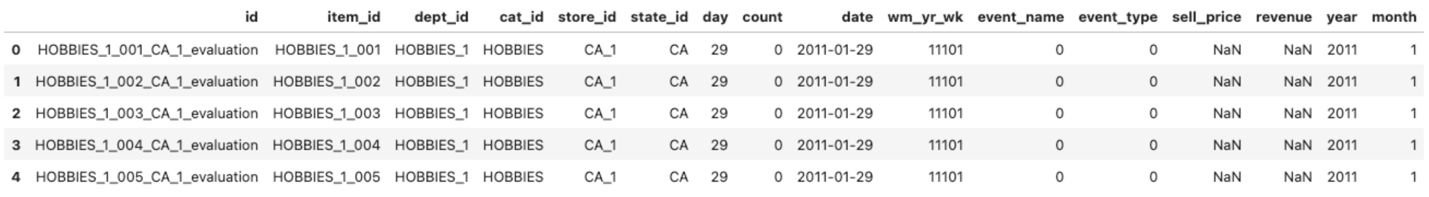


Figure2-The mnso bar chart for finding missing values

2.b. Feature Engineering



	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
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	29	0	2011-01-29	11101	0	0	NaN	NaN	2011	1

We calculate revenue by multiplying the count of each sold item by its selling price. For the SARIMA model, we exclusively utilize the 'revenue' and 'date' columns. Since SARIMA is a univariate time series model, we do not need other features, and the performance of this model is highly dependent on choosing the hyperparameters correctly.

2.c. Modelling

For this experiment, as we require time series models, we conducted runs of ARIMA and SARIMA models with various 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.

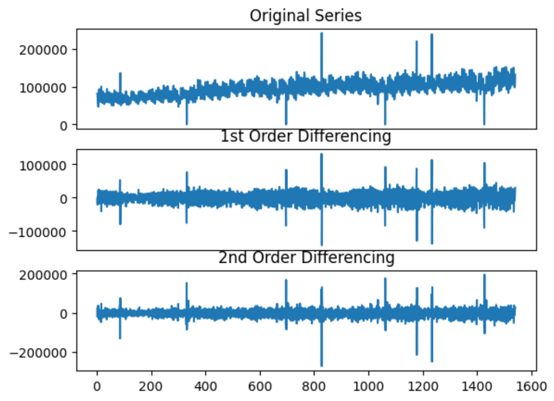


Figure 2-Differencing plot

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**

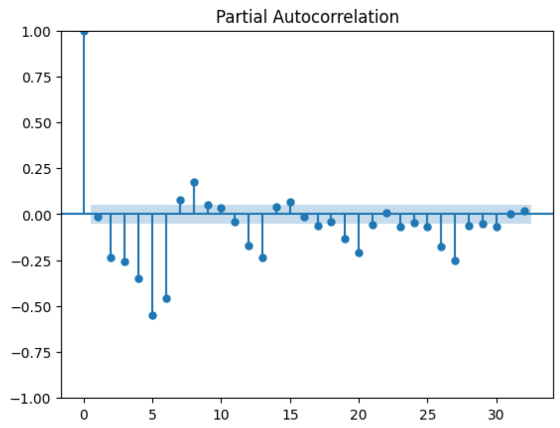
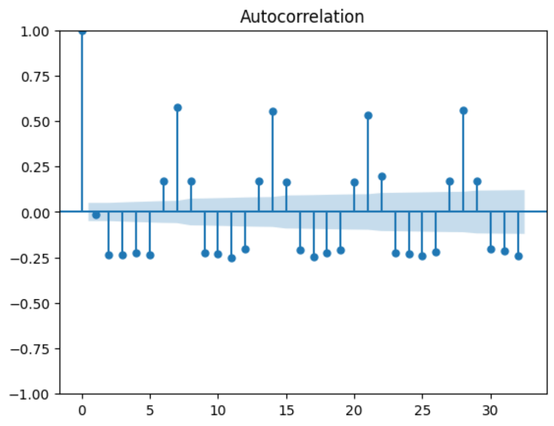


Figure 3-Partial Autocorrelation plot

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**

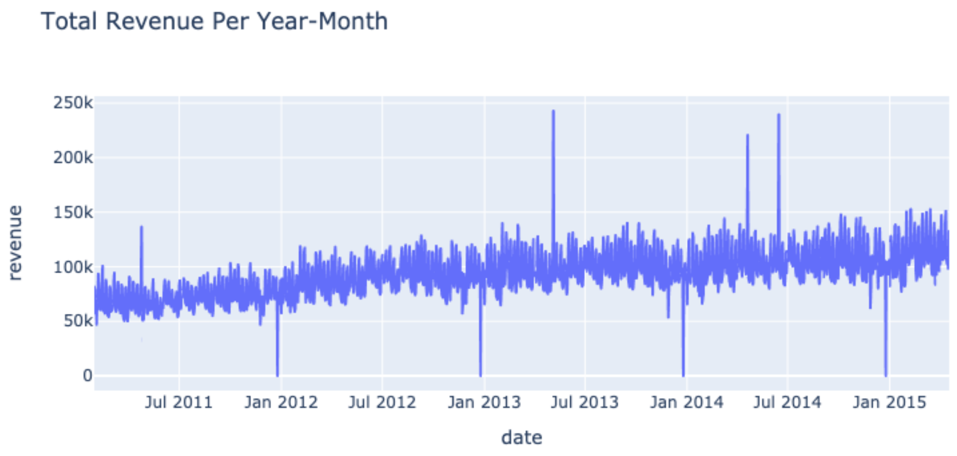


To determine the value of 'q,' we can turn to the Autocorrelation Function (ACF) plot, which provides insights into the extent of the moving average necessary to eliminate autocorrelation in a stationary time series. Here we can see that 4 of the lags are out of the significance limit so we can say that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

- **P**, or the Seasonal AutoRegressive Order, signifies the number of previous seasonal data points considered when predicting the current season's value. A higher P value indicates a greater reliance on past seasonal patterns, enabling the model to capture longer-term seasonal trends. However, it's important to note that as P increases, so does the complexity of the model. For our analysis, we experimented with P values ranging from 0 to 6 and found that the optimal value was 4.

- **Q**, known as the Seasonal Moving Average Order, dictates the number of preceding forecast errors considered for refining the current prediction with seasonality. A higher Q value signifies a greater incorporation of past forecast errors to fine-tune the current prediction. This facilitates the capture of any persisting patterns in the errors, albeit at the cost of heightened model complexity. In our analysis, we conducted experiments with Q values from 0 to 6, ultimately identifying the optimal value as 3.

- **D**, or the Differencing Order, denotes the degree of differencing that transforms the time series into a stationary form. Increasing the D value implies using differencing multiple times to achieve stationarity. A higher D value can eliminate trends and seasonality, making the data amenable to modelling. However, it's essential to exercise caution, as excessive differencing can result in information loss. In our analysis, we conducted experiments with D values ranging from 0 to 6 and determined the optimal value to be 1.



- **m (Seasonal Period)**: In SARIMA modelling, "m" signifies the seasonal period, representing the number of time steps per seasonal cycle within the data. This parameter defines how frequently the seasonality patterns repeat in the time series. For example, when working with monthly data exhibiting annual seasonality, "m" is commonly set to 12, corresponding to the 12 months in a year.In our specific analysis, we selected an "m" value of 12, aligning with the evident yearly patterns in the data.

In future experiments, the inclusion of different machine learning models like Prophet is valuable.

3. EXPERIMENT RESULTS

3.a. Technical Performance

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.

RMSE for ARIMA on test set: 20511.13

RMSE for SARIMA on test set: 22406.99

3.b. Business Impact

Interpreting the results of sales revenue forecasting experiments is vital for informed decision-making. Incorrect forecasts can have significant consequences, including inventory shortages, financial losses, and customer dissatisfaction. These errors affect inventory management, resource allocation, marketing, and overall business performance. Given that both models produced high Root Mean Square Error (RMSE) values, it is imperative to consider more robust and accurate forecasting models to prevent such problems from recurring. To maintain a competitive edge, businesses must continuously refine their forecasting models to mitigate these potential negative impacts.

3.c. Encountered Issues

Initially, we encountered identical values for each date in our forecasts due to incorrect hyperparameter settings. However, after fine-tuning these hyperparameters, we successfully resolved this issue. In future experiments, considering more suitable hyperparameters, exploring more robust models, or working with larger datasets may improve our forecasting accuracy and help reduce the Root Mean Square Error (RMSE).

4.FUTURE EXPERIMENT

4.a. Key Learning

Both models exhibit significant RMSE values, but deploying a more robust model like a neural network is worthwhile. Achieving higher accuracy can substantially boost business profits. Leveraging plots for hyperparameter optimization in both models proves to be highly beneficial.

4.b. Suggestions / Recommendations

Potential Next Steps and Experiments:

1. Experiment with Advanced Time Series Models:
    - Implement state-of-the-art time series forecasting models such as Prophet or LSTM neural networks.
    - Expected Uplift: These models are known for capturing complex temporal patterns, potentially reducing RMSE and improving accuracy significantly.
2. Hyperparameter Tuning:
    - Fine-tune the hyperparameters of selected models using cross-validation.
    - Expected Uplift: Optimizing hyperparameters can significantly improve model performance.
3. Cross-Validation Strategies:
    - Implement more advanced cross-validation methods such as time series cross-validation (TSCV) to evaluate model robustness.
    - Expected Uplift: Proper cross-validation can better estimate model generalization performance.
4. Continuous Monitoring and Model Updating:
    - Implement a system for continuous monitoring of model performance.
    - Set up processes to retrain models periodically with new data.
    - Expected Uplift: Continuous monitoring ensures that the model remains effective as data patterns evolve.

Ranking of Experiments:

1. Experiment with Advanced Time Series Models
2. Hyperparameter Tuning
3. Cross-Validation Strategies
4. Continuous Monitoring and Model Updating