EXPERIMENT REPORT

Student Name	Saumya Bhutani
Project Name	AT 2
Date	10/10/2023
Deliverables	<SaumyaBhutani_AT2_14360820> <SARIMA_Forecasting>

---

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	**Business Goal:** The primary goal of this project for the business is to develop a forecasting model using a Machine Learning algorithm that accurately predicts the sales revenue for next 7 dates at a specific date. This model will provide insights into sales patterns and trends for each item in each store on any given date. **Use of Results:** Inventory Management: Accurate sales revenue predictions will help optimize inventory management. The business can stock the right quantity of each item to meet demand while minimizing overstock or understock situations. Demand Forecasting: The model's predictions can assist in demand forecasting for individual items. This information can be used to plan production, distribution, and marketing strategies more effectively. Pricing Strategies: By understanding the sales trends for specific items, the business can adjust pricing strategies to maximize revenue and profit margins. Store Performance Evaluation: The model can provide insights into the performance of each store, identifying which items are top sellers in different regions or states. Marketing Campaigns: The results can inform marketing campaigns, helping the business target specific items or categories for promotion in different stores.

	Impact of Accurate Results: Accurate sales predictions have several key applications: 1. Inventory Optimization: Ensure the right stock levels, reducing overstock and understock situations. 2. Demand Forecasting: Plan production, distribution, and marketing effectively. 3. Pricing Strategy: Adjust pricing for higher revenue and margins. 4. Store Performance: Identify top-selling items in different regions. 5. Marketing Campaigns: Tailor promotions for specific items or categories. **Impact of Incorrect Results:** 1. Inventory Problems: Overstocking ties up capital, while understocking leads to lost sales. 2. Customer Dissatisfaction: Out-of-stock situations frustrate customers. 3. Revenue Loss: Inaccurate forecasts can lead to revenue losses. 4. Resource Inefficiencies: Misallocation of resources causes inefficiencies and increased costs.
1.b. Hypothesis	Hypothesis: SARIMA model can effectively predict sales revenue to improve business outcomes. It is worthwhile to consider this hypothesis as it can provide valuable insights for inventory optimization, improved demand forecasting, pricing strategies, store performance evaluation, and optimized marketing. These benefits can lead to cost reduction, increased profitability, and enhanced customer satisfaction, justifying the importance of testing it.
1.c. Experiment Objective	I expect the model to give results in favour of my Hypothesis. However, it may have the following possible outcomes: Positive outcome: it can identify significant predictors. The model's rmse score is low. The model can be used to make predictions. Negative outcome: it cannot identify significant predictors. The model's rmse score is high. The model cannot be used by the business . Inconclusive outcome: it identifies some significant predictors but the model's rmse score is high to be actionable.

---

2. EXPERIMENT DETAILS	
Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.	
2.a. Data Preparation	For this experiment, I performed the following data preparation steps: 1. Dataset Preparation: Initially, five datasets were provided, including the train set, test set, calendar set, event set, and item_weekly_sell_price set. At first I concatenated the cols in train with test set. In the train set, I transposed the date columns to make it interpretable, providing information about items sold per day. Then, using the day information from the calendar set, I performed a left join between the train and calendar sets to incorporate details about the date of item sales and the corresponding week. Next, I merged the previously joined dataset with the item_weekly_sold dataset, using the week as the common identifier, and calculated revenue by multiplying the quantity sold by the sell price. From te final dataset, I took two columns date and revenue which were then grouped datwise and summed the revenue values. The final dataset has 1941 rows. 2. Data splitting The date colum was set as index as in time series forecasting it provides benefits like easier data manipulation, better visualization, and compatibility with forecasting models. It allows for time-based operations and simplifies analysis. The dataset was split into df_train_data (rows from 1 to 1541) and df_test_data (1542 to 1941) . This was done according to the original training and testing set.

2.b. Feature Engineering	The Dickey-Fuller test was initially applied to the revenue dataset to assess its stationarity. It was observed that the dataset did not meet the stationarity criteria, with a p-value exceeding the standard significance level of 0.05. In an attempt to address this issue, a first-order difference (revenue first shift) was computed to make the data more stationary. However, this transformation did not yield the desired outcome, as the p-value remained above 0.05, indicating non-stationarity. Subsequently, a weekly first shift was introduced by shifting the data by 7 periods, resulting in a p-value of 0.00, which still indicated non-stationarity. To achieve stationarity, a weekly second shift was executed by shifting the data by 14 periods. This final transformation successfully rendered the dataset stationary, as confirmed by the Dickey-Fuller test. However, it also revealed the presence of seasonality in the data. These sequential transformations were pivotal in preparing the dataset for modelling. They were aimed at ensuring that the data was suitable for machine learning algorithms, enabling them to capture underlying patterns and facilitate accurate predictions while complying with the p-value threshold for stationarity assessment.

2.c. Modelling	The SARIMA (Seasonal Autoregressive Integrated Moving Average) model was chosen for modelling because it effectively addresses the seasonality present in the dataset. SARIMA is an extension of the ARIMA (Autoregressive Integrated Moving Average) model, specifically designed to handle time series data with seasonal patterns. It was observed that the sales revenue data exhibited clear seasonality, which means that there were recurring patterns or trends that repeated over specific time intervals. The key components of a SARIMA model include: 1. Seasonal Autoregressive (SAR) terms: These capture the relationship between the current value and previous values within each seasonal cycle. 2. Seasonal Integrated (I) term: This represents the differencing required to make the data stationary across seasonal cycles. 3. Seasonal Moving Average (SMA) terms: These account for the correlation between the current value and past forecast errors within each seasonal cycle. By incorporating these components, SARIMA models can effectively model and predict time series data with seasonality, making them a suitable choice for this project. The aim was to leverage SARIMA's ability to capture and account for the observed seasonal patterns in sales revenue data, ultimately leading to more accurate and reliable sales forecasts. The SARIMA model is applied to the ‘weekly second difference’ column which is now stationary. It has the following hyperparameters: p: Specifies the order of the seasonal autoregressive (AR) component. d: Indicates the number of differences to make the data stationary. q: Sets the order of the seasonal moving average (MA) component. P: Specifies the order of the non-seasonal AR component. D: Indicates the order of non-seasonal differencing. Q: Sets the order of the non-seasonal MA component. s: Defines the length of the seasonal cycle. These parameters help tailor the model to capture seasonality and trends in time series data.

<table>
<tr><th colspan="2">3. EXPERIMENT RESULTS</th></tr>
<tr><td colspan="2">Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.</td></tr>
<tr><td>3.a. Technical Performance</td><td>RMSE (Root Mean Squared Error) is a preferred metric for evaluating regression models due to its practical interpretability, sensitivity to errors, and mathematical properties. It measures prediction accuracy in the same units as the target variable, emphasizing the impact of larger errors and effectively handling outliers. RMSE is consistent across datasets and facilitates model comparisons.

Root Mean Squared Error (RMSE): 18985

A high Root Mean Squared Error (RMSE) score of 18,985 for a SARIMA model can be attributed to several reasons:

Complexity of the Data: SARIMA models may struggle with highly complex or noisy data. If the underlying patterns in the data are intricate or irregular, it can be challenging for SARIMA to capture them accurately.

Inadequate Model Tuning: The SARIMA model requires careful selection of hyperparameters, such as the order of differencing (d), seasonal order (p, q, P, Q), and the lag values. If these hyperparameters are not tuned properly, the model may not perform optimally.

Outliers or Anomalies: Outliers or anomalies in the data can significantly affect the performance of time series models like SARIMA. If not properly handled or removed, outliers can lead to inflated RMSE values.</td></tr>
<tr><td>3.b. Business Impact</td><td>Accurate sales predictions have a significant impact on various aspects of the business, including inventory optimization, demand forecasting, pricing strategies, store performance evaluation, and marketing campaigns.
The SARIMA model has shown strong predictive performance with low RMSE scores.

Deploying this model could benefit the business by improving sales predictions, optimizing pricing, enhancing marketing campaigns, and streamlining resource allocation.

Incorrect results can lead to inventory problems, customer dissatisfaction, revenue loss, resource inefficiencies, and suboptimal pricing and marketing strategies. Accurate predictions are crucial for achieving business goals and maximizing profitability.

The model can be put into deployment.</td></tr>
</table>

3.c. Encountered Issues	Model Performance: Some models, such as ARIMA, did not perform as expected, possibly due to seasonality. Solution: Experimenting with alternative models and hyperparameter tuning helped improve performance. Future Consideration: Evaluating ensemble methods or deep learning approaches for better model performance. Scalability: Ensuring that the experiments can scale efficiently with even larger datasets in the future is a challenge. Solution: Considering cloud-based solutions and parallel processing frameworks for scalability.

---

4. FUTURE EXPERIMENT	
Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.	
4.a. Key Learning	The experiment yielded valuable insights and raised several considerations for future experimentation: 1. Data Preparation is Crucial: The process of data preparation, including merging, cleaning, and transforming datasets, is critical. Careful attention to data quality and format significantly impacts the success of machine learning experiments. 2.. Hyperparameter Tuning is Essential: Hyperparameter tuning can have a substantial impact on model performance. Further experimentation with hyperparameter optimization is warranted to fine-tune models. 6. Ensemble Methods and Deep Learning: Exploring ensemble methods and deep learning approaches may lead to better predictive performance, especially when dealing with complex datasets. 7. Continuous Optimization: The need for continuous optimization and exploration of advanced techniques for handling large and complex datasets is evident. It's crucial to adapt and evolve the approach to meet the demands of future experiments. Considering the insights gained and the identified challenges, it is not a dead end but rather a stepping stone for further experimentation. The current approach provides a foundation for future research, focusing on improving model performance, scalability, and efficiency. Experimenting with different models, hyperparameter tuning, and

	exploring advanced techniques will contribute to better predictive capabilities and valuable business insights.
4.b. Suggestions / Recommendations	Potential Next Steps and Experiments: 1. Feature Engineering: Explore additional feature engineering techniques, such as lag features, rolling statistics, and one-hot encoding for categorical variables, to enhance model performance. Expected uplift: Moderate. 2. Advanced Models: Experiment with more advanced models, including deep learning architectures such as LSTM and Transformer models. These models may capture complex patterns in the data better. Expected uplift: High. 3. Hyperparameter Tuning: Conduct an extensive hyperparameter tuning process using techniques like grid search, random search, or Bayesian optimization to fine-tune model parameters and achieve optimal performance. Expected uplift: Moderate to High. 4. Scaling and Distributed Computing: Deploy the experimentation environment on a cloud platform that supports scalable infrastructure, such as AWS, Azure, or Google Cloud. Leverage distributed computing frameworks like Dask to process large datasets efficiently. Expected uplift: High. 5. Deployment and Monitoring: If a model meets the required business outcome, deploy it into a production environment. Set up monitoring to track model performance and retrain as needed. Expected uplift: High (if successful deployment). 6. Continuous Improvement: Establish a process for continuous model improvement. This includes retraining models with fresh data, re-evaluating feature engineering, and periodically reassessing model performance. Expected uplift: Ongoing. 7. Cross-Validation: Implement robust cross-validation techniques to ensure model stability and generalize well to unseen data. Expected uplift: Moderate. 8. Explainability: Incorporate model explainability techniques to understand the factors driving predictions. This can provide valuable insights for decision-making. Expected uplift: Moderate. To deploy a successful solution into production: 1. Model Evaluation: Thoroughly evaluate the model's performance on a holdout dataset or through cross-validation. Ensure it meets predefined business objectives and metrics. 2. Scalable Infrastructure: Deploy the model on a production-ready infrastructure capable of handling real-time or batch predictions, depending on business needs. 3. API Development: Develop APIs for integrating the model into existing business systems and applications. Ensure data input and output are well-defined.

	4. Monitoring and Alerts: Implement monitoring and alerting systems to detect and respond to model performance degradation or anomalies. 5. Feedback Loop: Establish a feedback loop for continuous improvement, where model performance is regularly reviewed and retraining is scheduled. 6. Documentation: Document the model, its architecture, and dependencies for future reference and maintenance. 7. Security: Implement security measures to protect data and model access. 8. User Training: Train relevant business stakeholders on how to interpret and use model predictions effectively. 9. Compliance: Ensure compliance with data privacy and regulatory requirements. 10. Testing: Rigorously test the deployed solution in a staging environment before releasing it to production. 11. Deployment Plan: Create a deployment plan that outlines the steps, responsibilities, and timeline for the production release. 12. Post-Deployment Evaluation: Continuously monitor the model's performance in the production environment and be prepared to make adjustments as necessary. These steps, when executed diligently, can lead to a successful deployment of the predictive model, driving business growth and revenue optimization. Steps to be performed in future: Deep learning techniques to be applied.

Appendix:

Note: Coding references has bee taken from lecture sources

NumPy. (2021). NumPy: The fundamental package for scientific computing with Python. Retrieved from https://numpy.org/(Accessed April 28, 2023)

pandas. (2022). pandas 1.3.3 documentation. Retrieved from https://pandas.pydata.org/docs/1.4.0/index.html(Accessed April 28, 2023)

scikit-learn. (2021). Scikit-learn: Machine learning in Python. Retrieved from https://scikit-learn.org/stable/index.html(Accessed April 28, 2023)

Lecture Tutorials

Google colaboratory. (n.d.). Retrieved April 28, 2023, from https://colab.research.google.com/drive/1gPfMZWhkCPXZvEabRIgTKC5ldPP66h9L?usp=share_link

Google colaboratory. (n.d.). Retrieved April 28, 2023, from https://colab.research.google.com/drive/1XygeQMupiP8CQSwJQ0f53Zg81rpLJIx2?usp=share_link

Google colaboratory. (n.d.). Retrieved April 28, 2023, from https://colab.research.google.com/drive/10Wf8BKyWJQA4oZfuz8NSCxylv5xEln9H

Github link: https://github.com/bhutanisaumya/Adv_MLA_AT2