EXPERIMENT REPORT

Student Name	Saumya Bhutani
Project Name	AT 2
Date	10/10/2023
Deliverables	<SaumyaBhutani_AT2_14360820> <LGBMBoost Hyperparameter Tuning>

---

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 predictive model using a Machine Learning algorithm that accurately predicts the sales revenue for specific items 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: The Random Forest model can effectively predict sales revenue to improve business outcomes. Hypothesis: The XGBoost model can efficiently forecast sales revenue, leading to improved business outcomes. Hypothesis: The LightGBM (Light Gradient Boosting Machine) Regressor 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. After conducting a series of hypothesis tests, I will conclude which model performs the best.

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, four datasets were provided, including the train set, calendar set, event set, and item_weekly_sell_price set. In the train set, I transformed 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. Lastly, a new column labeled 'event' with values 'Yes' and 'No' was added to the merged dataset, indicating whether an event occurred on a specific date, based on the events dataset. 2. Checked for NAN values across all the features. It was identified that the features 'sell_price', 'revenue, 'event_name' and 'event_type' had NAN values. The columns 'event_name' and 'event_type' had a high percentage of missing values which is more than 91.5%. Therefore, I did not consider these features for my analysis and were removed. For Nan in columns, 'sell_price', 'revenue' were filled with 0. 3. Data splitting I separated the dependent feature ('revenue') and independent features into y and X respectively in a 70:30 ratio. The 2 datasets: train and validation sets in the ratio 70:30. I split in this way to ensure that there is enough data to train the model.
2.b. Feature Engineering	The features 'sell_price', 'revenue, 'event_name' and 'event_type' had NAN values. The columns 'event_name' and 'event_type' had a high percentage of missing values which is more than 91.5%. Therefore, I did not consider these features for my analysis and were removed. For Nan in columns, 'sell_price', and 'revenue' were filled with 0. The date column has been split into 4 columns namely, day, month, day_of_week and year. Dropped features: I have removed the following columns from my dataset: ● `cat_id`: This column was dropped because the information it contained is now captured in the `item_id`. ● `Sales`: The `Sales` column was excluded because the focus is on predicting revenue, not sales. ● `date`: Although the `date` feature was used to extract day, month, day_of the week, and year, it was dropped because a datetime column cannot be used directly in modelling. ● `wm_yr_wk`: This column was initially used for merging tables but is no longer needed. ● `sell_price`: The per-item cost information was discarded as the primary

interest is in revenue.

- ‘event’ : as there are a lot of fdates having no event. The column is not useful for modelling.
- `day`: This column was removed due to the high number of values. Instead, the `month`, `day_of_week`, and `year` will be used in modeling.

I have utilized a pipeline to execute the following sequence of actions:

1. Item_id Transformation: The `item_id` column (having 3049 categories) has undergone a transformation using Target Encoding. This choice was made because Target Encoding is effective in handling categorical variables and capturing their relationships with the target variable. It is typically useful when there are a large number of categories.

2. Categorical Column Transformation: The columns 'store_id,' and 'day_of_week' have been encoded using One-Hot Encoding. This transformation was applied to convert object-type features into a numeric format. This step is crucial before modelling because many machine learning algorithms require numerical inputs to perform mathematical computations.

Categorical columns	No. of categories
store_id	10
day_of_week	7

3. Numerical Column Transformation: All the numerical columns have been processed using Min-Max Scaling. This transformation ensures that numerical variables are appropriately scaled within a specific range, which is essential for accurate model predictions.

These transformations were performed to ensure that the data is adequately prepared for modelling, allowing machine learning algorithms to capture patterns and make accurate predictions effectively.

2.c. Modelling	Decision Tree Regressor Decision tree regressors works by recursively splitting the data into subsets based on the most significant attribute at each node, eventually forming a tree-like structure. These trees are used to make predictions on numerical target variables. Decision trees can capture non-linear relationships in the data but are prone to overfitting. However, they can be regularized to mitigate this issue. Decision tree regressors are interpretable and easy to visualize, making them useful for understanding feature importance. XGBoost Regressor XGBoost (Extreme Gradient Boosting) is an ensemble learning method that can be used for regression tasks. It builds multiple decision trees sequentially, each tree correcting the errors of the previous one. XGBoost uses a gradient-boosting framework and incorporates regularization techniques to prevent overfitting. It's known for its high predictive accuracy and speed. XGBoost can capture complex relationships between features and target variables, making it effective for a wide range of regression problems. LightGBM (LGBM) Regressor LightGBM is another gradient-boosting framework that excels in regression tasks. It's similar to XGBoost but offers improved efficiency and speed due to its histogram-based approach to tree building. LightGBM uses a leaf-wise strategy to grow trees, which can result in a more balanced tree structure. It also supports parallel and GPU learning, making it suitable for large datasets. LGBM regressors are known for their high predictive performance and are particularly effective when dealing with complex, high-dimensional data. Dependent feature - revenue Independent feature - store_id , item_id , month , day_of_week , year Less number of Independent is used it is as per business requirement and also since the size of the dataset is large I am considering only the essential features.

3. EXPERIMENT RESULTS

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.

3.a. Technical Performance

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.

Decision Tree Regressor
Training set RMSE score :6.99
Validation set RMSE score :7.03

The Decision Tree Regressor performs well with an RMSE score of 6.99 on the training set and 7.03 on the validation set. These scores suggest that the model's predictions are close to the actual values for both datasets, indicating good generalization.

XGBoost Regressor
Training set RMSE score : 7.58
Validation set RMSE score : 7.60

The XGBoost model shows reasonable performance with similar RMSE scores for both training and validation sets, indicating a good fit and generalization.

LightGBM (LGBM) Regressor
Training set RMSE score : 7.55
Validation set RMSE score : 7.56

Hyperparameter: n_estimators = 100

The LightGBM (LGBM) Regressor performs well with a training set RMSE score of 7.55 and a validation set RMSE score of 7.56. This suggests that the model has good predictive capabilities and generalizes well to unseen data.

Overall, all three models exhibit similar performance, with RMSE scores in the same range.
Thus, other factors which include model interpretability, training time, and resource requirements are considered in choosing the best model for deployment.

LightGBM outperforms the other two models due to the following:

- Speed: It is faster and memory-efficient.
- Memory Usage: It uses less memory.
- Parallel Processing: It supports parallel and distributed computing.
- Hyperparameter Tuning: Requires fewer tuning iterations.
- Ensemble Nature: Provides good predictive performance.

3.b. Business Impact	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 LightGBM (LGBM) 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.
3.c. Encountered Issues	Data Merging: One of the challenges was merging various datasets with different formats, particularly when some were in wide format. Careful analysis and data preprocessing were necessary to align the data correctly. Large Dataset: The final dataset size was substantial, leading to extended modelling times and memory issues. This required efforts to optimize memory usage. This is also one of the reasons why cross-validation was not working. Tried to: Convert the dataset to HDF5 format using Vaex helped reduce memory usage, but it wasn't entirely sufficient. Future Consideration: Exploring distributed computing or cloud-based solutions for handling large datasets could be beneficial in future experiments. Memory Issues: Despite converting to HDF5, memory issues persisted, particularly when running resource-intensive models like Random Forest Regressor. Solution: Implementing a data pipeline helped streamline the process, but limitations in handling heavy models remained. Future Consideration: Further research into memory-efficient algorithms and distributed computing frameworks may be necessary for large-scale experiments. Model Performance: Some models, such as Random Forest Regressor, did not perform as expected, possibly due to dataset size and complexity. Solution: Experimenting with alternative models and hyperparameter tuning could help 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. Challenges with Large Datasets: Handling large datasets presents computational challenges, including memory limitations and extended modeling times. While some issues were mitigated, the need for scalable solutions remains. 3. Model Performance Varies: Different machine learning models exhibited varying levels of performance. While LightGBM showed promise, other models like Random Forest Regressor faced limitations due to dataset size. 4. 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. 5. Consideration of Scalability: As the dataset size grows, scalability becomes a concern. Future experiments should explore cloud-based solutions and distributed computing to ensure efficient processing. 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. The performance of the regressors are highly dependent on the values of hyperparameters. The model may overfit the training data if these hyperparameters are not correctly tuned. I tried manual and automated hyper parameterization. I would want

	to explore boosting techniques to predict.
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 gradient boosting ensembles like XGBoost and CatBoost, as well as 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. Data Sampling: Explore data sampling techniques, such as down-sampling or stratified sampling, to handle class imbalance and potentially reduce the dataset size for faster experimentation. Expected uplift: Moderate. 5. 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. 6. 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). 7. 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. 8. Cross-Validation: Implement robust cross-validation techniques to ensure model stability and generalize well to unseen data. Expected uplift: Moderate. 9. 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