# **EXPERIMENT REPORT**

|  |  |
| --- | --- |
| **Student Name** | Sudarat Sukjaroen |
| **Project Name** | Assignment 2 |
| **Date** | 10 October 2023 |
| **Deliverables** | sukjaroen\_sudarat-24667255.ipynb  https://github.com/sudarat-pom/AdvanceML\_AT2 |

|  |  |
| --- | --- |
| 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** | As I am a data scientist. I am assigned to create 2 applications.  1. Predictive model using the best machine learning algorithm with the best metrics value and creates an application for user input item, store and date. Then, the application predicts the sales revenue.  2. Forecasting model using the best machine learning algorithm with the best metrics value and creating an application for the user to forecast the total by store and item in the next 7 days.  The business objectives are focused on improving decision-making and operational efficiency by leveraging predictive and forecasting models to optimize inventory, pricing, and resource allocation, ultimately leading to improved profitability and customer satisfaction. |
| **1.b. Hypothesis** | Predictive Model Hypothesis:  H0: No significant relationship between features and sales revenue.  H1: Significant relationship; ML predicts sales revenue accurately.  Forecasting Model Hypothesis:  H0: Inaccurate total revenue forecasts using time-series analysis.  H1: Accurate forecasts using time-series analysis. |
| **1.c. Experiment Objective** | Experiment Objective: Assess two ML models' performance (predictive and forecasting) with the expectation of high prediction accuracy.  Expected Outcome: Accurate models deployable as production APIs for optimized business decisions.  Possible Scenarios:  - Both models succeed (high accuracy).  - One model succeeds; the other may need improvements.  - Neither model meets expectations, requiring further refinement. |

|  |  |
| --- | --- |
| 1. **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** | Data exploration  1.Number of sales\_train data set = 30,490 (total 1,547 columns) Total = 47,168,030  2.Number of sales\_test data set = 30,490 (total 400 columns) Total = 12,196,000  3.Number of sell\_prices data set = 6,841,121 (total 4 columns) Total = 27,264,484  4. Number of calendar data set = 1,969 (total 3 columns)  5.Number of calendar\_events data set = 167 (total 3 columns)  Data preparation  - No Null value  - Check duplicate record  - Display the top 5/10 for each dataframe to check how data is stored. |
| **2.b. Feature Engineering** | 1. Create store\_item\_id dataframe to map store and item from text to number. (Total 30,490 records)    2. Add event\_name\_num and event\_type\_num for map data from text to number in calendar\_events dataframe.    3. Concat sales\_train and sales\_test together (do in the first time, but remove this step later).  4. Transposed data from vertical to horizontal.  Data was stored in Vertical    Data is stored in Horizontal     1. Merge with calendar dataframe to get date and wm\_yr\_wk.      1. Merge with sell\_prices dataframe to get sell\_price and calculate sales\_revenue.      1. Merge with store\_item\_id dataframe to get store\_id\_num and item\_id\_num.      1. Merge with calendar\_event to get event\_name, event\_name\_id, event\_type and event\_type\_id and replace Null value that can not merge by ‘None’ and 999.     Number of Records: 46,976,160  Number of columns: 18 |
| **2.c. Modelling** | **Prediction model**   1. Calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error) for a baseline that uses all sales\_train dataframe.   Baseline MAE: 4.4718985282519395  Baseline MSE: 69.34113576036032  2. Run Linear Regression model with sales\_train and calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error).  Linear Regression Training MAE: 4.4731755981207195  Linear Regression Training MSE: 68.39213463021729  3. Run Random Forest model with sales\_train dataframe and calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error).  Random Forest Training MAE: 3.590928695305071  Random Forest Training MSE: 47.64180952426532  4. Run Decision Tree model with sales\_train dataframe and calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error).  Decision Tree Training MAE: 3.6000990680736082  Decision Tree Training MSE: 48.039007642660216  5. Run Gradient Boosting model with sales\_train dataframe and calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error).  Gradient Boosting Training MAE: 4.044916029251678  Gradient Boosting Training MSE: 55.37447005080774  6. Run XGBoost model with sales\_train dataframe and calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error).  XGBoost Training MAE: 3.6271729932970085  XGBoost Training MSE: 46.870818579955085  Note   1. To compare MAE and MSE for each model. Filter data dept=’FOODS\_1’ then Number of Records: 2,734,536 and Number of columns: 18. 2. Selected features are store\_id\_num, item\_id\_num, event\_name\_id and event\_type\_id. 3. Run experiments with train test split 80%, 20% and random state = 42   **Forecasting model**   1. SARIMA model   From the maximum date in sales\_train dataframe, 18/04/2015, run the SARIMA model to forecast sales revenue in the next 7 days by store\_id and item\_id. Run experiment on store\_id = 'CA\_1' and item\_id = 'FOODS\_1\_001'. |

|  |  |
| --- | --- |
| 1. **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** | Summarise the metrics from 6 prediction models; the best metric came from the XGBoost model.      The final XGBoost model runs from all data in the sales\_train dataframe.  Number of Records: 34,815,174  Number of columns: 18 |
| **3.b. Business Impact** | Predictive Model:   * Positive Impact: Accurate predictions enable better inventory management, reducing overstock or understock situations. This can result in cost savings and improved customer satisfaction. * Negative Impact: Incorrect predictions may lead to stockouts or excess inventory, resulting in financial losses and customer dissatisfaction.   Forecasting Model:   * Positive Impact: Accurate forecasts aid in demand planning, helping the business allocate resources efficiently. It allows for timely adjustments to production, staffing, and inventory levels. * Negative Impact: Inaccurate forecasts can lead to overproduction, underutilized resources, or stockouts, affecting profitability and operational efficiency. |
| **3.c. Encountered Issues** | Can not concat sales\_train and sales\_test  Firstly, I concat them together, transposed and used data in sales\_test to calculate the baseline. After running the forecasting model, the results were negative amounts that are abnormal. Then I investigated and found data in the sales column (Number of sales items) were missing or incomplete. It made all calculations inaccurate.  Then I do not concat them and rerun all models. |

|  |  |
| --- | --- |
| 1. **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** | Predictive Model (Machine Learning Algorithm):   * Machine learning models offer accurate sales predictions with algorithm and feature choices impacting performance. * Metrics like MAE, MSE assess model accuracy. * Experimentation and feature engineering enhance predictions. * Valuable for inventory, demand forecasting, and pricing at store/item levels.   Forecasting Model (Time-Series Analysis Algorithm):   * Time-series models forecast total sales revenue for the next 7 days, capturing seasonality and trends. * Forecast accuracy depends on data quality and parameter choices (e.g., seasonal orders). * Regular model retraining with fresh data is crucial for adapting to changing sales patterns. * Combining forecasts across stores/items aids in overall sales trend analysis and demand planning. |
| **4.b. Suggestions / Recommendations** | 1. Run more models on forecasting models.  2. Add 1 layer between the user interface and API.  I have found the selected features will be the input from the user, but before running the model, I do the feature engineering. Then, the selected feature becomes a number that inconveniences the user to input them directly.  We should add 1 layer between the user interface and API. To receive the data from Store name, Item name, and Date. We could design them as a drop-down list that is easy to input and validate data.  After that, we provide code to map and convert to a number store\_id\_num, item\_id\_num, event\_name\_id and event\_type\_id before sending input for API. I also created the prototype Python code to support this layer. |