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.

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

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)

	store_id	item_id	store_id_num	item_id_num
0	CA_1	HOBBIES_1_001	0	0
154	CA_1	HOBBIES_1_002	0	1
416	CA_1	HOBBIES_1_003	0	2
541	CA_1	HOBBIES_1_004	0	3
818	CA_1	HOBBIES_1_005	0	4

2. Add event_name_num and event_type_num for map data from text to number in calendar_events dataframe.

	date	event_name	event_type	event_name_id	event_type_id
0	2011-02-06	SuperBowl	Sporting	0	0
1	2011-02-14	ValentinesDay	Cultural	1	1
2	2011-02-21	PresidentsDay	National	2	2
3	2011-03-09	LentStart	Religious	3	3
4	2011-03-16	LentWeek2	Religious	4	3
..	...	...	...	...	...
162	2016-05-30	MemorialDay	National	12	2
163	2016-06-02	NBAFinalsStart	Sporting	13	0
164	2016-06-07	Ramadan starts	Religious	17	3
165	2016-06-19	Father's day	Cultural	15	1
166	2016-06-19	NBAFinalsEnd	Sporting	14	0

[167 rows x 5 columns]

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

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...	d_1532
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	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
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	8
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2
5	HOBBIES_1_006_CA_1_evaluation	HOBBIES_1_006	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2
6	HOBBIES_1_007_CA_1_evaluation	HOBBIES_1_007	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0
7	HOBBIES_1_008_CA_1_evaluation	HOBBIES_1_008	HOBBIES_1	HOBBIES	CA_1	CA	12	15	0	0	...	0
8	HOBBIES_1_009_CA_1_evaluation	HOBBIES_1_009	HOBBIES_1	HOBBIES	CA_1	CA	2	0	7	3	...	2
9	HOBBIES_1_010_CA_1_evaluation	HOBBIES_1_010	HOBBIES_1	HOBBIES	CA_1	CA	0	0	1	0	...	0

Data is stored in Horizontal

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
0	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_1	3.0
1	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_2	0.0
2	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_3	0.0
3	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_4	1.0
4	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_5	4.0
5	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_6	2.0
6	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_7	0.0
7	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_8	2.0
8	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_9	0.0
9	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_10	0.0

5. Merge with calendar dataframe to get date and wm_yr_wk.

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	date	wm_yr_wk
0	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_1	3.0	2011-01-29	11101
1	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_2	0.0	2011-01-30	11101
2	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_3	0.0	2011-01-31	11101
3	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_4	1.0	2011-02-01	11101
4	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_5	4.0	2011-02-02	11101
5	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_6	2.0	2011-02-03	11101
6	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_7	0.0	2011-02-04	11101
7	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_8	2.0	2011-02-05	11102
8	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_9	0.0	2011-02-06	11102
9	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_10	0.0	2011-02-07	11102

6. Merge with sell_prices dataframe to get sell_price and calculate sales_revenue.

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	date	wm_yr_wk	sell_price	sales_revenue
0	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_1	3.0	2011-01-29	11101	2.0	6.0
1	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_2	0.0	2011-01-30	11101	2.0	0.0
2	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_3	0.0	2011-01-31	11101	2.0	0.0
3	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_4	1.0	2011-02-01	11101	2.0	2.0
4	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_5	4.0	2011-02-02	11101	2.0	8.0
5	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_6	2.0	2011-02-03	11101	2.0	4.0
6	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_7	0.0	2011-02-04	11101	2.0	0.0
7	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_8	2.0	2011-02-05	11102	2.0	4.0
8	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_9	0.0	2011-02-06	11102	2.0	0.0
9	FOODS_1_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_10	0.0	2011-02-07	11102	2.0	0.0

7. Merge with store_item_id dataframe to get store_id_num and item_id_num.

id	item_id	dept_id	cat_id	store_id	state_id	d	sales	date	wm_yr_wk	sell_price	sales_revenue	store_id_num	item_id_num
_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_1	3.0	2011-01-29	11101	2.0	6.0	0	1612
_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_2	0.0	2011-01-30	11101	2.0	0.0	0	1612
_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_3	0.0	2011-01-31	11101	2.0	0.0	0	1612
_001_CA_1_evaluation	FOODS_1_001	FOODS_1	FOODS	CA_1	CA	d_4	1.0	2011-02-01	11101	2.0	2.0	0	1612

8. 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.

store_id	state_id	d	sales	date	wm_yr_wk	sell_price	sales_revenue	store_id_num	item_id_num	event_name	event_name_id	event_type	event_type_id
CA_1	CA	d_1	3.0	2011-01-29	11101	2.0	6.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_2	0.0	2011-01-30	11101	2.0	0.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_3	0.0	2011-01-31	11101	2.0	0.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_4	1.0	2011-02-01	11101	2.0	2.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_5	4.0	2011-02-02	11101	2.0	8.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_6	2.0	2011-02-03	11101	2.0	4.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_7	0.0	2011-02-04	11101	2.0	0.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_8	2.0	2011-02-05	11102	2.0	4.0	0	1612	None	999.0	None	999.0
CA_1	CA	d_9	0.0	2011-02-06	11102	2.0	0.0	0	1612	SuperBowl	0.0	Sporting	0.0

Number of Records: 46,976,160
Number of columns: 18

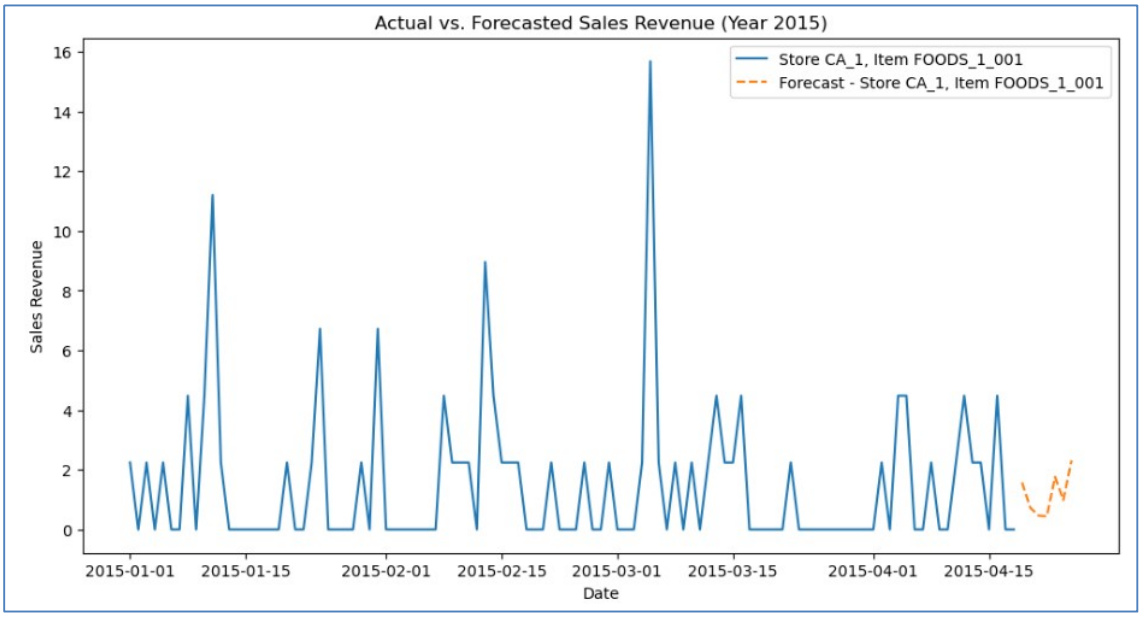
2.c. Modelling	**Prediction model** 1. Calculate MAE (Mean Absolute Error) and MSE (Mean Squared Error) for a baseline that uses all sales_train dataframe. Baseline MAE: 4.4718985282519395 Baseline MSE: 69.34113576036032 2. Run Linear Regression model with sales_train and calculate MAE (Mean Absolute 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 <u>Note</u> 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'.

Forecasted Sales Revenue for the Next 7 Days:

	store_id	item_id	date	forecasted_sales
0	CA_1	FOODS_1_001	2015-04-19	1.588005
1	CA_1	FOODS_1_001	2015-04-20	1.017585
2	CA_1	FOODS_1_001	2015-04-21	0.941780
3	CA_1	FOODS_1_001	2015-04-22	1.177714
4	CA_1	FOODS_1_001	2015-04-23	0.880150
5	CA_1	FOODS_1_001	2015-04-24	1.678179
6	CA_1	FOODS_1_001	2015-04-25	1.924820



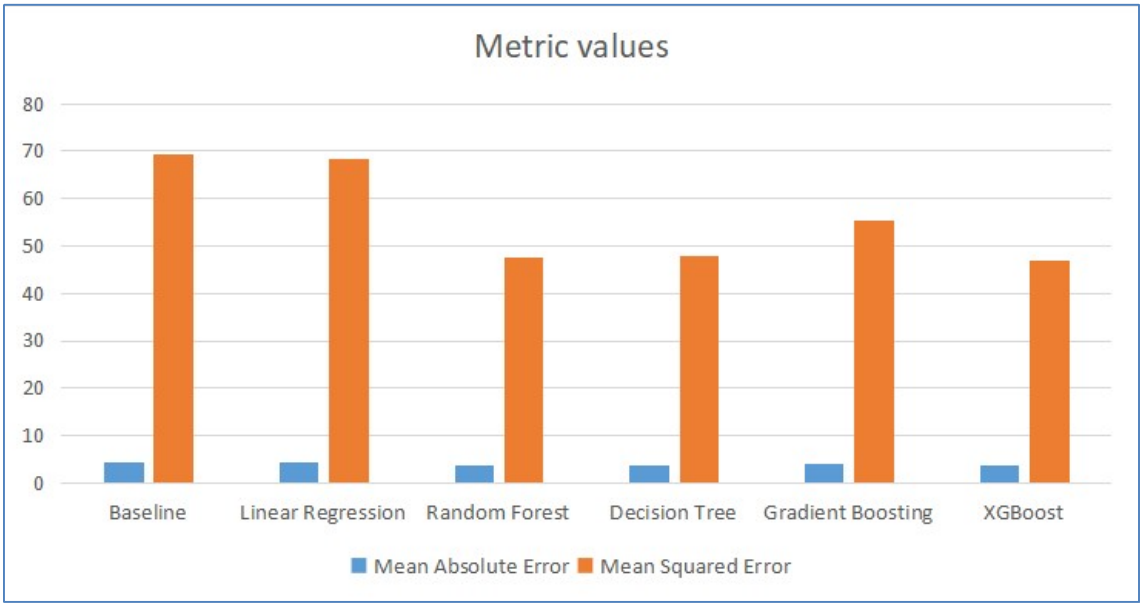
3. EXPERIMENT RESULTS

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

Summarise the metrics from 6 prediction models; the best metric came from the XGBoost model.

No.	Model Name	Mean Absolute Error	Mean Squared Error
1	Baseline	4.471898528	69.34113576
2	Linear Regression	4.473175598	68.39213463
3	Random Forest	3.590928695	47.64180952
4	Decision Tree	3.600099068	48.03900764
5	Gradient Boosting	4.044916029	55.37447005
6	XGBoost	3.627172993	46.87081858



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.

---

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	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.