Assignment 2 ML as a Service
https://github.com/sudarat-pom/AdvanceML_AT2

Sudarat Sukjaroen
Student ID: 24667255
10 October 2023

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

1. Executive Summary

The project aims to develop two essential models for sales revenue prediction and forecasting, each with its own specific objective and significance:

Predictive Model (Machine Learning Algorithm):

Objective: To build a predictive model using machine learning algorithms that accurately forecast sales revenue for a specific item in a particular store on a given date.

Significance: Accurate sales revenue predictions at the item-store-date level are crucial for inventory management, demand forecasting, and pricing strategies. This model enables data-driven decision-making at a granular level, optimizing operations and maximizing revenue.

Forecasting Model (Time-Series Analysis Algorithm):

Objective: To develop a forecasting model using time-series analysis algorithms that provide forecasts for the total sales revenue across all stores and items for the next 7 days.

Significance: Accurate sales revenue forecasting helps in effective resource allocation, production planning, and staffing. It allows the business to adapt to short-term changes in demand, ultimately improving operational efficiency and customer satisfaction.

2. Business Understanding

a. Business Use Cases

Inventory Management:

Use Case: Accurate sales revenue predictions at the item-store-date level enable optimized inventory management. Businesses can maintain optimal stock levels, reducing the risk of overstock or stockouts.

Challenges/Opportunities: Seasonal demand fluctuations, changing customer preferences, and market trends make inventory management challenging. Predictive models help businesses adapt to these changes efficiently.

Demand Forecasting:

Use Case: Short-term sales forecasts are crucial for demand planning and resource allocation. Accurate forecasts enable businesses to adjust production, staffing, and marketing strategies in response to changing demand.

Challenges/Opportunities: Fluctuations in demand, promotions, and external factors (e.g., holidays) pose challenges in forecasting. Time-series analysis algorithms capture these patterns and help businesses respond effectively.

Pricing Strategies:

Use Case: Data-driven pricing strategies can be based on accurate sales predictions. Businesses can optimize prices to maximize revenue while remaining competitive.

Challenges/Opportunities: Setting optimal prices requires considering various factors, including demand elasticity and competitor pricing. Predictive models assist in pricing decisions by providing insights into expected sales.

b. Key Objectives

Sales Revenue Prediction:

Objective: Develop a predictive model to accurately forecast sales revenue for specific items in individual stores on given dates.

Stakeholders: Retail Managers, Inventory Managers, Pricing Analysts.

Requirements: Stakeholders require accurate sales predictions to optimize inventory levels, pricing strategies, and demand planning. They need timely insights to avoid overstocking or stockouts and reduce operational costs.

Project Approach: The project addresses this objective by leveraging machine learning algorithms to analyze historical sales data, capture patterns, and make accurate sales revenue predictions at the item-store-date level.

Sales Revenue Forecasting:

Objective: Create a forecasting model that provides forecasts for total sales revenue across all stores and items for the next 7 days.

Stakeholders: Operations Managers, Resource Allocation Teams, Marketing Teams.

Requirements: Stakeholders require short-term sales forecasts to optimize resource allocation, production planning, and marketing efforts. They need the ability to adapt to changing market dynamics and customer demand patterns.

Project Approach: The project addresses this objective by employing time-series analysis algorithms to capture seasonality and trends, enabling the generation of accurate forecasts for total sales revenue over a 7-day horizon.

3. Data Understanding

Table 1: Overall data in the project

No	Data set name	Number of records	Number of columns	Total
1	sales_train	30,490	1,547	47,168,030
2	sales_test	30,490	400	12,196,000
3	sell_prices	6,841,121	4	27,364,484
4	calendar	1,969	3	5,907
5	calendar_events	167	3	501

Figure 1: Overview of sales train data

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...	d_1532	d_1533	d_1534
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1	1	1
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	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	0	1
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	8	2	0
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	0	1
5	HOBBIES_1_006_CA_1_evaluation	HOBBIES_1_006	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	0	0
6	HOBBIES_1_007_CA_1_evaluation	HOBBIES_1_007	HOBBIES_1	HOBBIES	CA_1	CA	0	0	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	12	14
8	HOBBIES_1_009_CA_1_evaluation	HOBBIES_1_009	HOBBIES_1	HOBBIES	CA_1	CA	2	0	7	3	...	2	0	0
9	HOBBIES_1_010_CA_1_evaluation	HOBBIES_1_010	HOBBIES_1	HOBBIES	CA_1	CA	0	0	1	0	...	0	1	0

Figure 2: Overview of sales test data

	d_1542	d_1543	d_1544	d_1545	d_1546	d_1547	d_1548	d_1549	d_1550	d_1551	...	d_1932	d_1933	d_1934	d_1935	d_1936	d_1937
0	0	1	0	2	1	0	2	0	1	0	...	2	4	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	...	0	1	2	1	1	0
2	0	0	0	0	0	1	0	0	0	0	...	1	0	2	0	0	0
3	4	1	0	1	3	5	2	3	0	2	...	1	1	0	4	0	1
4	3	0	0	1	1	0	2	0	2	1	...	0	0	0	2	1	0
5	2	0	0	0	1	0	0	1	1	3	...	2	1	0	0	1	0
6	1	0	1	0	0	0	0	0	0	0	...	0	1	0	0	0	1
7	4	0	5	14	2	2	10	11	7	30	...	7	0	6	0	15	5
8	0	0	0	0	1	0	1	0	0	0	...	1	0	0	0	0	0
9	3	3	0	0	0	1	0	2	1	0	...	0	0	1	0	2	1

Figure 3: Overview of Sell price items data

	store_id	item_id	wm_yr_wk	sell_price
0	CA_1	HOBBIES_1_001	11325	9.58
1	CA_1	HOBBIES_1_001	11326	9.58
2	CA_1	HOBBIES_1_001	11327	8.26
3	CA_1	HOBBIES_1_001	11328	8.26
4	CA_1	HOBBIES_1_001	11329	8.26
5	CA_1	HOBBIES_1_001	11330	8.26
6	CA_1	HOBBIES_1_001	11331	8.26
7	CA_1	HOBBIES_1_001	11332	8.26
8	CA_1	HOBBIES_1_001	11333	8.26
9	CA_1	HOBBIES_1_001	11334	8.26

Figure 4: Overview of calendar data

	date	wm_yr_wk	d
0	2011-01-29	11101	d_1
1	2011-01-30	11101	d_2
2	2011-01-31	11101	d_3
3	2011-02-01	11101	d_4
4	2011-02-02	11101	d_5
5	2011-02-03	11101	d_6
6	2011-02-04	11101	d_7
7	2011-02-05	11102	d_8
8	2011-02-06	11102	d_9
9	2011-02-07	11102	d_10

Figure 5: Overview of calendar events data

	date	event_name	event_type
0	2011-02-06	SuperBowl	Sporting
1	2011-02-14	ValentinesDay	Cultural
2	2011-02-21	PresidentsDay	National
3	2011-03-09	LentStart	Religious
4	2011-03-16	LentWeek2	Religious
5	2011-03-17	StPatricksDay	Cultural
6	2011-03-20	Purim End	Religious
7	2011-04-24	Easter	Cultural
8	2011-04-24	OrthodoxEaster	Religious
9	2011-04-26	Pesach End	Religious

Figure 6: Summarise sales revenue monthly by product categories

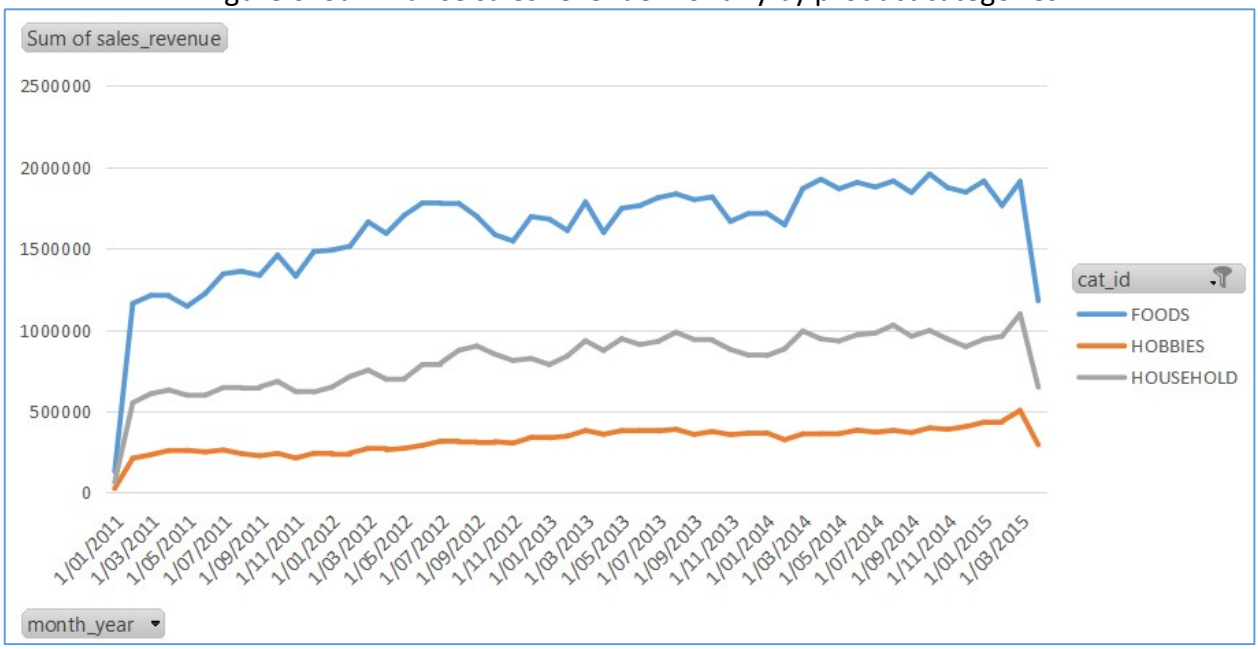


Figure 7: Summarise sales revenue monthly by states

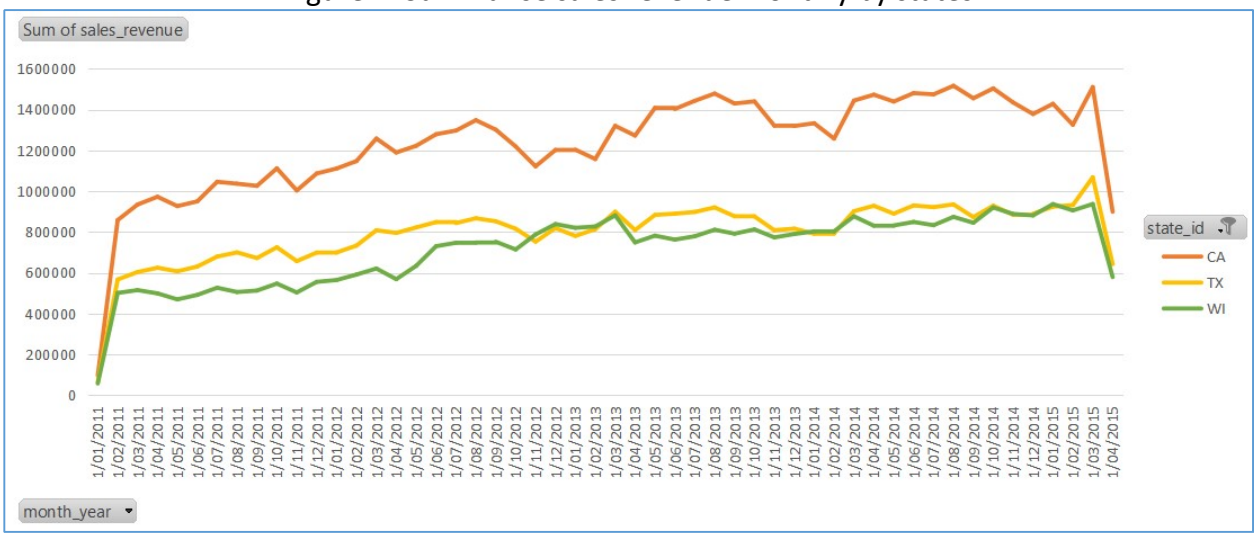
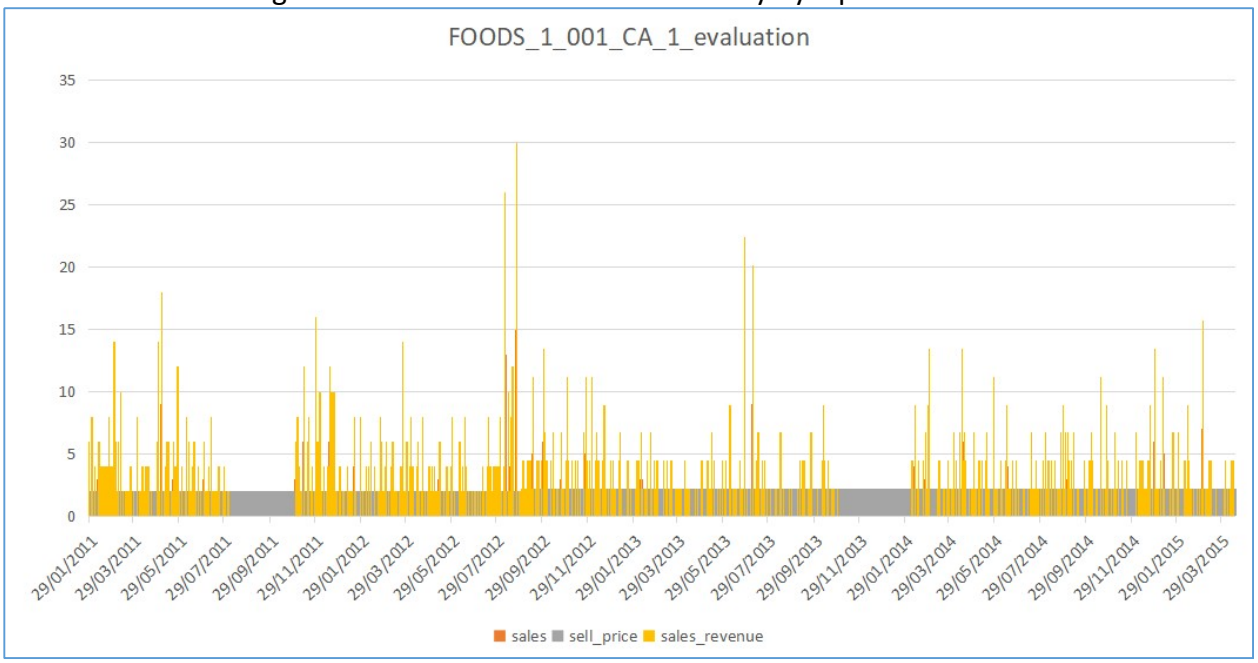


Figure 8: Summarise sales revenue daily by 1 product item



4. Data Preparation

- No Null value
- Check duplicate record and found 7 duplicated records in sales_test but can not drop because data is store in vertical. If we dropped them, data will be missing.

1. Create store_item_id dataframe to map store and item from text to number. (Total 30,490 records)

Figure 9: Result from data preparation step 1

	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.

Figure 10: Result from Data preparation step 2

	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.

Figure 11: 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

Figure 12: 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.

Figure 13: Result from Data preparation step 5

	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.

Figure 14: Result from Data preparation step 6

	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.

Figure 15: Result from Data preparation step 7

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.

Figure 16: Result from Data preparation step 8

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

5. Modelling

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.

Prediction Model

1. Baseline Calculation

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. Linear Regression Model

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. Random Forest Model

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. Decision Tree Model

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. Gradient Boosting Model

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. XGBoost Model

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

Forecasting Model

1. SARIMA model

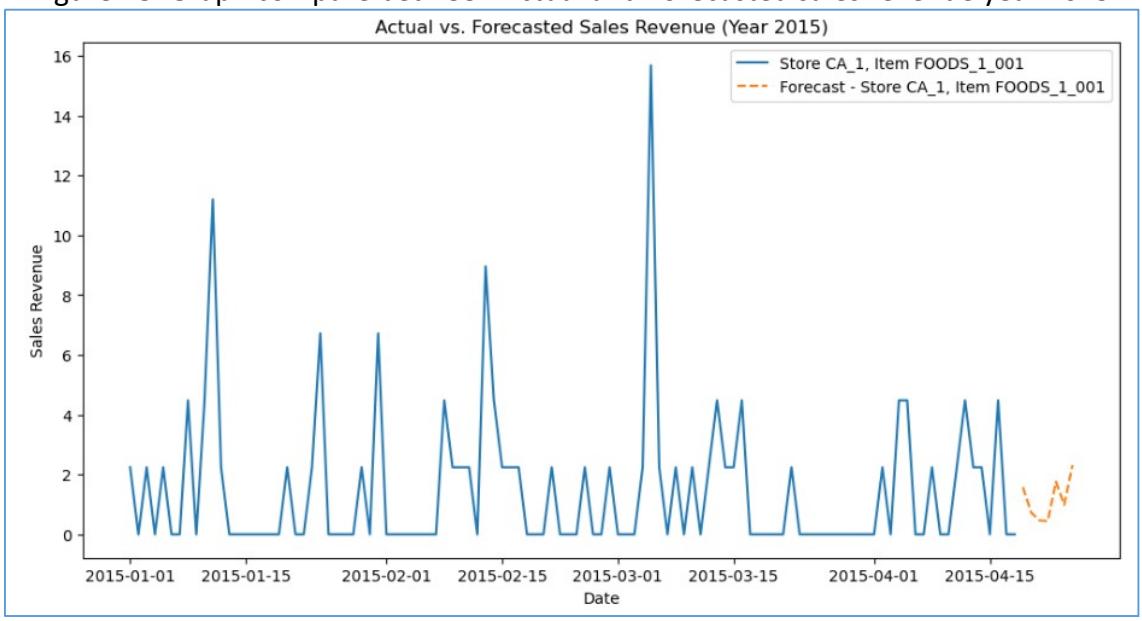
From the maximum date in sales_train dataframe, 18/04/2015, run the SARIMA model to forecast sales revenue in the next 7 days by store_id and item_id. Run experiment on store_id = 'CA_1' and item_id = 'FOODS_1_001'.

Figure 17: Result from Forecasting Model

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

Figure 18: Graph compare between Actual and Forecasted sales revenue year 2015



9. Evaluation

Evaluation Metrics

Mean Absolute Error (MAE): MAE is a metric used to measure the average absolute differences between predicted values and actual values in a dataset. It quantifies the magnitude of errors in predictions, where lower MAE values indicate better accuracy. MAE is less sensitive to outliers compared to MSE, making it useful for models where large errors should not be excessively penalized.Explain why these metrics were chosen and how they relate to the project goals.

Mean Squared Error (MSE): MSE is a metric that calculates the average of the squared differences between predicted values and actual values. It gives higher weight to larger errors, making it sensitive to outliers. MSE is commonly used in regression problems and is helpful for evaluating the overall quality of predictions. However, it may not provide a clear intuitive understanding of the magnitude of errors due to the squaring of differences.

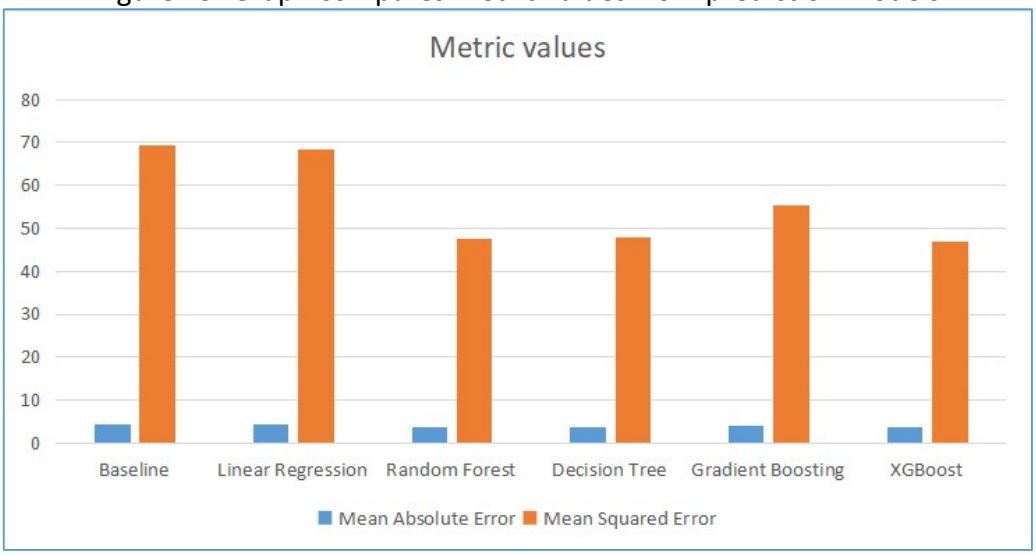
Results and Analysis

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

Table 2: Compare metric values from prediction models

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

Figure 19: Graph compares metric values from prediction models



The final XGBoost model runs from all data in the sales_train dataframe.
Number of Records: 34,815,174
Number of columns: 18

Business Impact and Benefits

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.

Data Privacy and Ethical Concerns

Data Privacy Implications:

Data privacy is a critical concern in this project, as it involves the collection and analysis of sales data, which may include sensitive information about customers, stores, and items.

Ethical Concerns:

Customer Privacy: There is a potential ethical concern related to the privacy of customer data, especially if the data includes personally identifiable information (PII). Protecting customer privacy and ensuring compliance with data protection regulations (e.g., GDPR) is paramount.

Bias and Fairness: Models developed for predicting sales revenue and demand forecasting should be carefully monitored for bias and fairness. Biased models can lead to unfair outcomes, such as unequal pricing or inventory allocation.

Transparency: Ethical concerns also revolve around model transparency. Stakeholders should have a clear understanding of how the models make predictions and what data they use.

Steps Taken to Ensure Data Privacy and Ethical Considerations:

Data Anonymization: Personal and sensitive customer information should be anonymized or pseudonymized to protect customer privacy while still allowing for meaningful analysis.

Data Encryption: Data should be encrypted both at rest and in transit to safeguard it from unauthorized access.

Data Access Controls: Implement strict access controls and authentication mechanisms to ensure that only authorized personnel can access the data.

10. Deployment

1. Plan to deploy by using FastAPI, Docker and Heroku.

FastAPI is a modern, high-performance web framework for building APIs with Python. It's known for its simplicity, automatic documentation generation, and fast execution speed.

Docker is a platform for developing, shipping and running applications in containers, which are lightweight, portable, and isolated environments that simplify application deployment and management. It enables consistent and efficient software deployment across different environments.

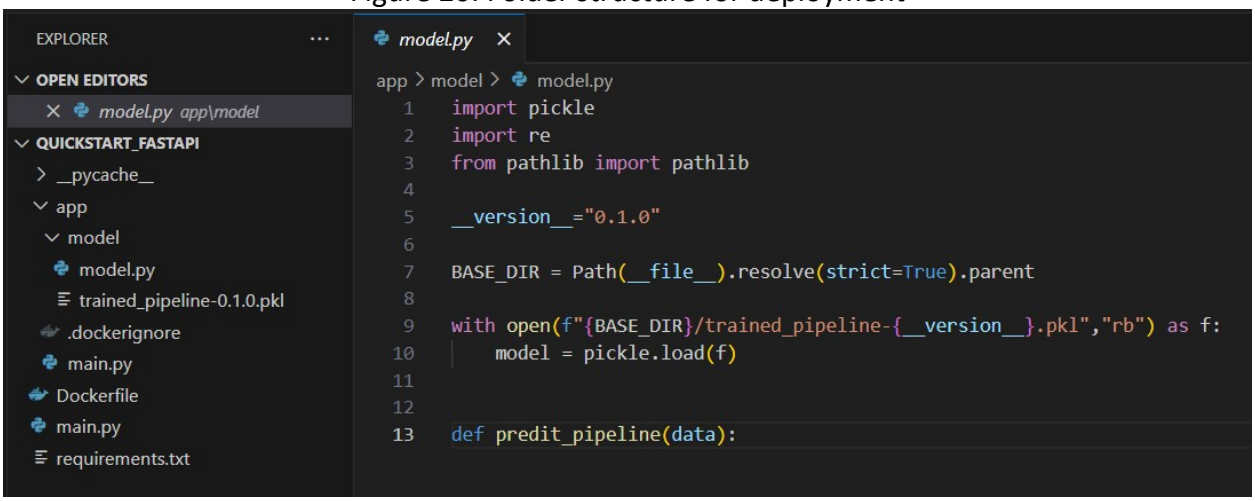
Heroku is a cloud platform as a service (PaaS) that allows developers to deploy, manage, and scale web applications and services quickly and easily. It abstracts infrastructure management.

2. Create the Pipe Line for the selected model and then create the pickle and download for deployment.

3. Create the environment for deployment using FastAPI framework. Start from create App folder, Model folder and put the pickle file and create model.py file inside, create main.py file.

4. In main root, create docker and dockerignore, requirements file.

Figure 20: Folder structure for deployment



5. Download FastAPI in Container - Docker code from FastAPI website to use in docker file.

6. For requirement file, put the name of dependency and version in the model. For example scikit-learn ==1.0.2.

7. Buid the model image to use as API.

Figure 21: Result from buid model image

```
PS C:\Users\sudar\anaconda3\Lib\site-packages\xlwings\quickstart_fastapi> docker build -t sales-revenue-prediction-app .
>>
[+] Building 2.6s (10/10) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 331B
 => [internal] load metadata for docker.io/library/python:3.9
 => [auth] library/python:pull token for registry-1.docker.io
 => [1/4] FROM docker.io/library/python:3.9@sha256:031b2a713079b2436e6c2102dbffbce109c9c2582ad0867c380e3c8d7fed477f
 => [internal] load build context
 => => transferring context: 251B
 => CACHED [2/4] COPY ./requirements.txt /app/requirements.txt
 => CACHED [3/4] RUN pip install --no-cache-dir --upgrade -r /app/requirements.txt
 => CACHED [4/4] COPY ./app /app/app
 => exporting to image
 => => exporting layers
 => => writing image sha256:fd0350e9c4c73956fb891fc6f55d614f560c6105125efada058b7876883322d2
 => => naming to docker.io/library/sales-revenue-prediction-app

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\sudar\anaconda3\Lib\site-packages\xlwings\quickstart_fastapi>
```

8. But there is the problem in running docker step. After run command line, it returned nothing. I tried to figure out by searching websites and consult people. I could not solve this problem.

9. For workaround to check the model by manually, you can run from experiment file names “sukjaroen_sudarat-24667255-predictive_XGBoost.ipynb” which is designed to be the same as user input.

Figure 22: Aspect input variables

```
Add 1 layer between the user interface and API

In [129]: new_data = {
              'store_id': 'CA_1',
              'item_id': 'FOODS_1_001',
              'date': '2020-06-19'
          }
```

Table 3: Sample Store ID and item ID to input

store_id	item_id
CA_1	HOBBIES_1_008
CA_1	HOBBIES_1_009
CA_1	HOBBIES_1_010
CA_1	HOBBIES_1_012
CA_1	HOBBIES_1_015
CA_1	HOBBIES_1_016
CA_1	HOBBIES_1_022
CA_1	HOBBIES_1_023
CA_1	HOBBIES_1_028
CA_1	HOBBIES_1_029

store_id	item_id
TX_1	HOBBIES_1_004
TX_1	HOBBIES_1_008
TX_1	HOBBIES_1_009
TX_1	HOBBIES_1_010
TX_1	HOBBIES_1_012
TX_1	HOBBIES_1_015
TX_1	HOBBIES_1_016
TX_1	HOBBIES_1_020
TX_1	HOBBIES_1_022
TX_1	HOBBIES_1_023

store_id	item_id
WI_1	HOBBIES_1_004
WI_1	HOBBIES_1_008
WI_1	HOBBIES_1_010
WI_1	HOBBIES_1_012
WI_1	HOBBIES_1_015
WI_1	HOBBIES_1_016
WI_1	HOBBIES_1_017
WI_1	HOBBIES_1_020
WI_1	HOBBIES_1_022
WI_1	HOBBIES_1_023

Figure 23: Aspect predicted sales revenue

```
[4.3929057]
```

	store_id	item_id	date	store_id_num	item_id_num	event_name	event_name_id	event_type	event_type_id	predicted_sales
0	CA_1	FOODS_1_001	2020-06-19	0	1612	Father's day	15	Cultural	1	4.392906

11. Conclusion

Summarize the key findings

the project has successfully developed and validated both predictive and forecasting models. These models can play a vital role in optimizing inventory management, demand forecasting, and pricing strategies for the business. Further refinement and continuous model updates will ensure accurate predictions and meaningful insights for better decision-making.

Reflect on the project's success

If the project can deploy and calculate the predictive and forecating value correctly, the project will be success in achieving its objectives and delivering valuable solutions to meet stakeholders' requirements. The accurate predictive and forecasting models are poised to make a positive impact on inventory management, pricing strategies, and demand planning within the business, and they are positioned for further improvements and adaptations as needed.

future work, recommendations

User-Friendly Interface: Develop a user-friendly interface or dashboard for stakeholders to interact with the models, visualize results, and access insights easily.

Model Refinement: Continue to refine and optimize the machine learning algorithms used in the predictive model. Explore more advanced techniques, such as deep learning or ensemble methods, to improve prediction accuracy further.

Feature Engineering: Invest in in-depth feature engineering to capture additional patterns and insights from the data. Incorporate external factors like holidays, promotions, and economic indicators that may influence sales revenue.