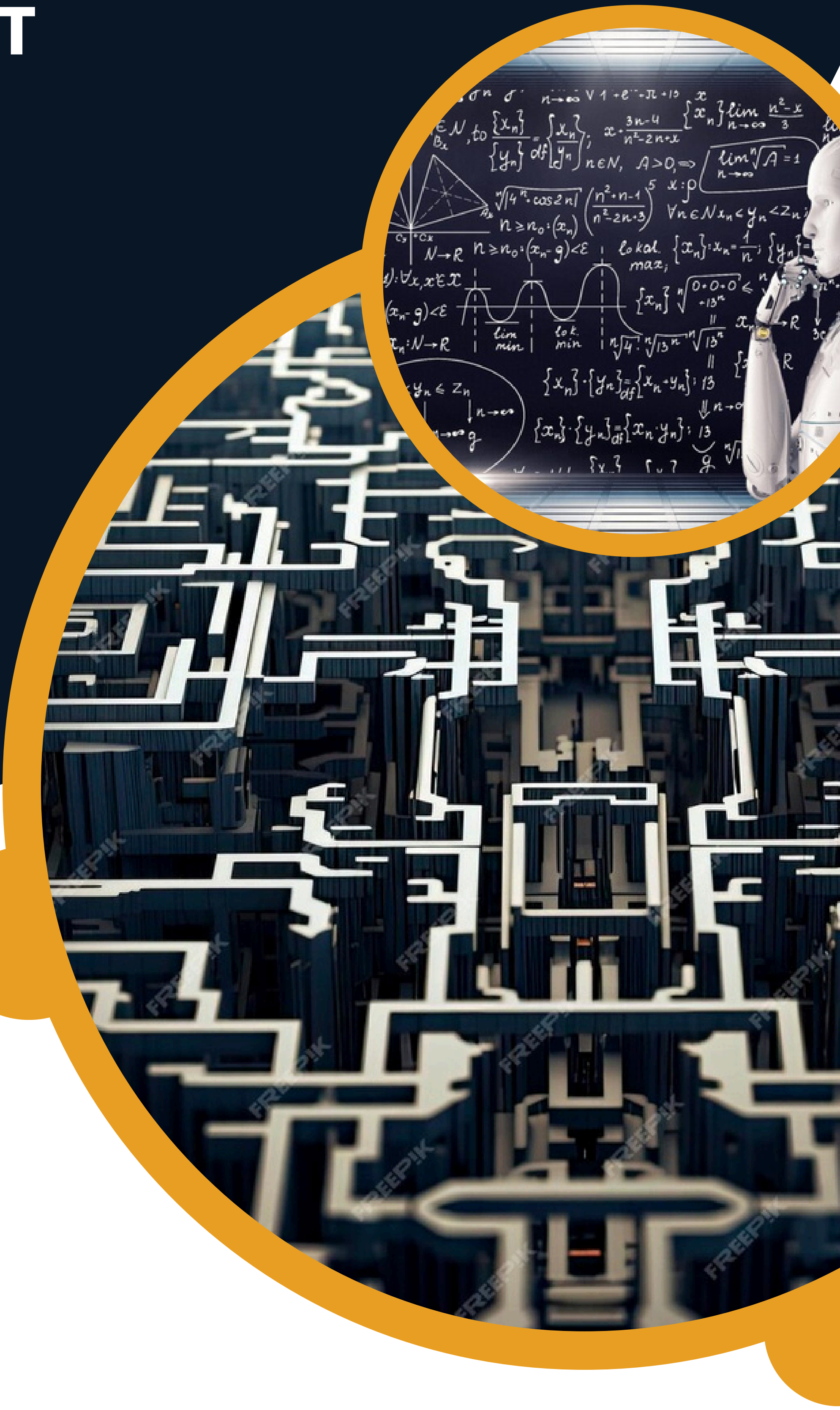
DATA SCIENCE – INTERNSHIP ASSIGNMENT

EXPERNETIC LLC.



Prepared By :

Janith Ramanayake

Table of Contents

1. Analysis of Supermarket Data

1.1 Description of the Data Set

The study was conducted using four datasets related to supermarket transactions: **Item**, **Promotion**, **Sales**, and **Supermarkets**, each provided in a .csv format. The problem description included brief overviews of each dataset but did not provide detailed descriptions of the variables within them. To conduct a thorough analysis, it is crucial to have an in-depth understanding of each variable. Therefore, **general knowledge** and the **OpenAI ChatGPT-4o** model were utilized to gain a more comprehensive description of each variable. The detailed descriptions of these variables are as follows.

Item Dataset

Variable	Description	Data Type
code	Unique code for identify item.	int64
description	Description of the item.	object
type	Item type (Type 1, Type 2, Type 3, Type 4)	object
brand	Brand name of the item	object
size	Weight of the item in Oz or LB	object

Promotion Dataset

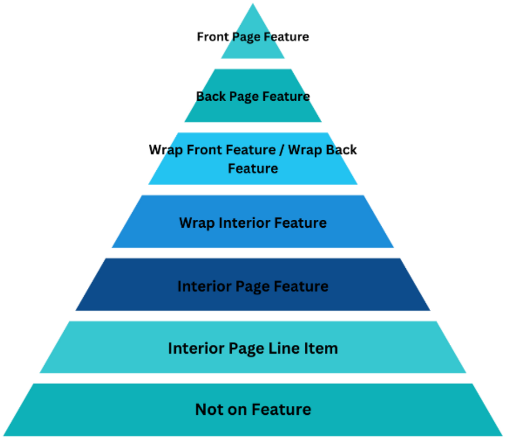
Variable	Description	Data Type
code	Unique code for identify item	int64
supermarkets	Unique code for identify supermarket	int64
week	Week number of the promotion happen	int64
feature	Visibility that products receive in promotional catalog	object
display	Display locations of products within the store	object
province	Province of the supermarket (1, 2)	int64

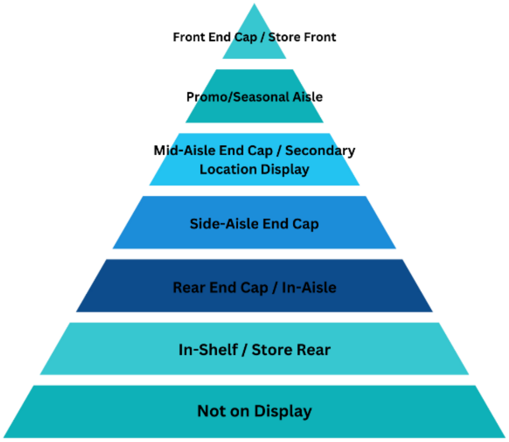
The "**Feature"** column represents various types of promotional placements or visibility that products may receive in marketing materials, such as promotional catalogs. In this dataset, there are 8 distinct types of features. Based on the research conducted, a detailed description of each feature type is provided below.

- **Not on Feature:** Product is not in the promotional catalog.
- **Interior Page Feature:** Product is featured inside the promotional catalog.
- **Wrap Interior Feature:** Product is highlighted on the interior wraparound page of the promotional catalog.
- **Wrap Back Feature:** Product is featured on the back of a wraparound page of the promotional catalog.
- **Interior Page Line Item:** Product is listed as a line item within the interior pages of the promotional catalog.
- **Wrap Front Feature:** Product is highlighted on the front of a wraparound page of the promotional catalog.
- **Front Page Feature:** Product is prominently displayed on the front page of the promotional catalog.
- **Back Page Feature:** Product is featured on the back page of a promotional catalog.

The **Display** column provides information about the physical placement or display locations of products within the store. This dataset includes 11 distinct display values, each corresponding to a specific location where a product can be positioned to enhance visibility or serve promotional purposes. A detailed explanation of each display type is provided below.

- **Mid-Aisle End Cap:** Display located at the end of a store aisle, in the middle of the store.
- **Not on Display:** Product is not display in the store.
- **Rear End Cap:** Product is displayed on an end cap at the back of the store.
- **Store Rear:** Product is placed towards the back of the store.
- **Front End Cap:** Display at the front of an aisle near the store's entrance or main walkways.
- **In-Shelf:** Product is located on a regular shelf, alongside other products.
- **Store Front:** Product is placed near the front of the store.
- **Secondary Location Display:** Product is placed in an additional display location beyond its regular shelf spot.
- **In-Aisle:** Product is displayed directly within an aisle.
- **Promo/Seasonal Aisle:** Product is featured in a dedicated aisle for promotional or seasonal items.
- **Side-Aisle End Cap:** Refers to an end cap on the side aisles.





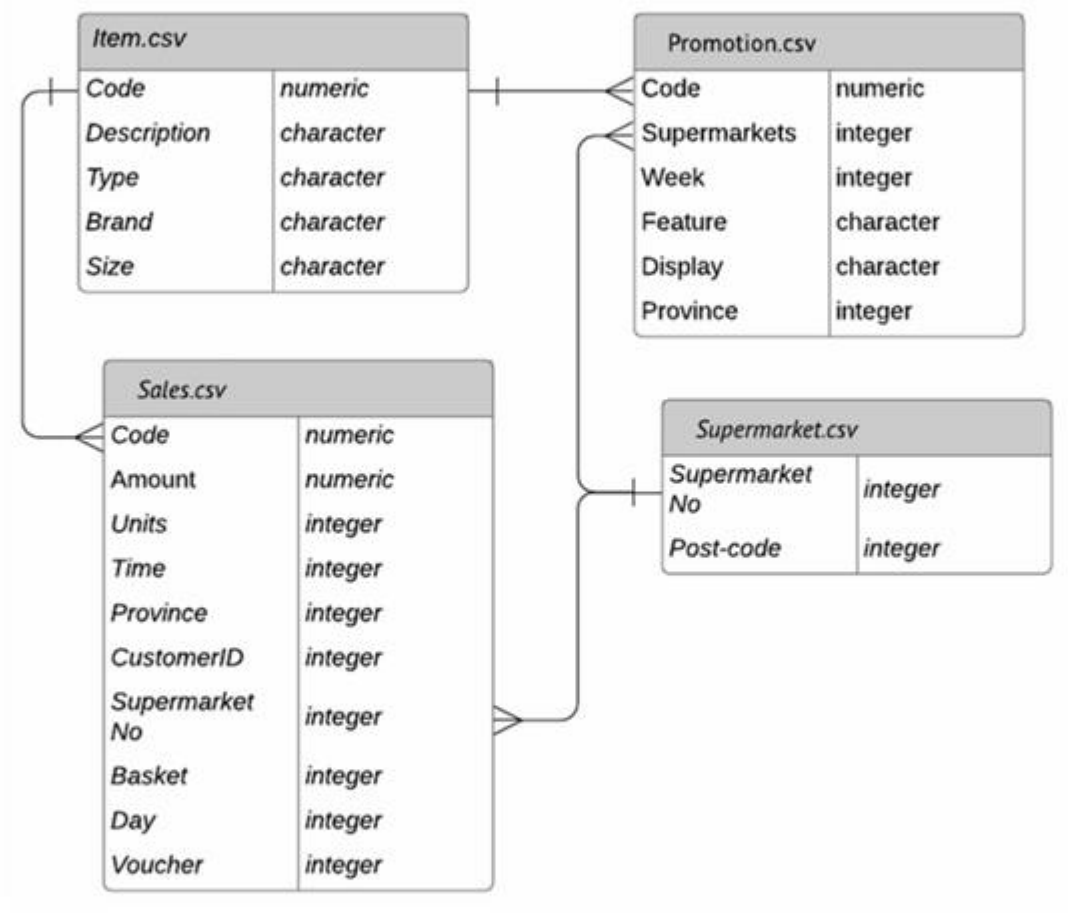
Sales Dataset

Variable	Description	Data Type
code	Unique code for identify item	int64
amount	Total amount that customer pay	float64
units	Number of units that customer buy	int64
time	Purchase time of the order	int64
province	Province of the supermarket (1, 2)	int64
week	Week number that purchase happens	int64
customerId	Unique code for identify Customer	int64
supermarket	Unique code for identify supermarket	int64
basket	Unique code for identify items that purchased together by customer	int64
day	Day number that purchase happens	int64
voucher	Any discount applied or not (0,1)	int64

Supermarket Dataset

Variable	Description	Data Type
supermarket_No	Unique code for identify supermarket	int64
postal-code	Postal code of region where supermarket located	int64

According to the problem statement relationships among the datasets represented as follows.

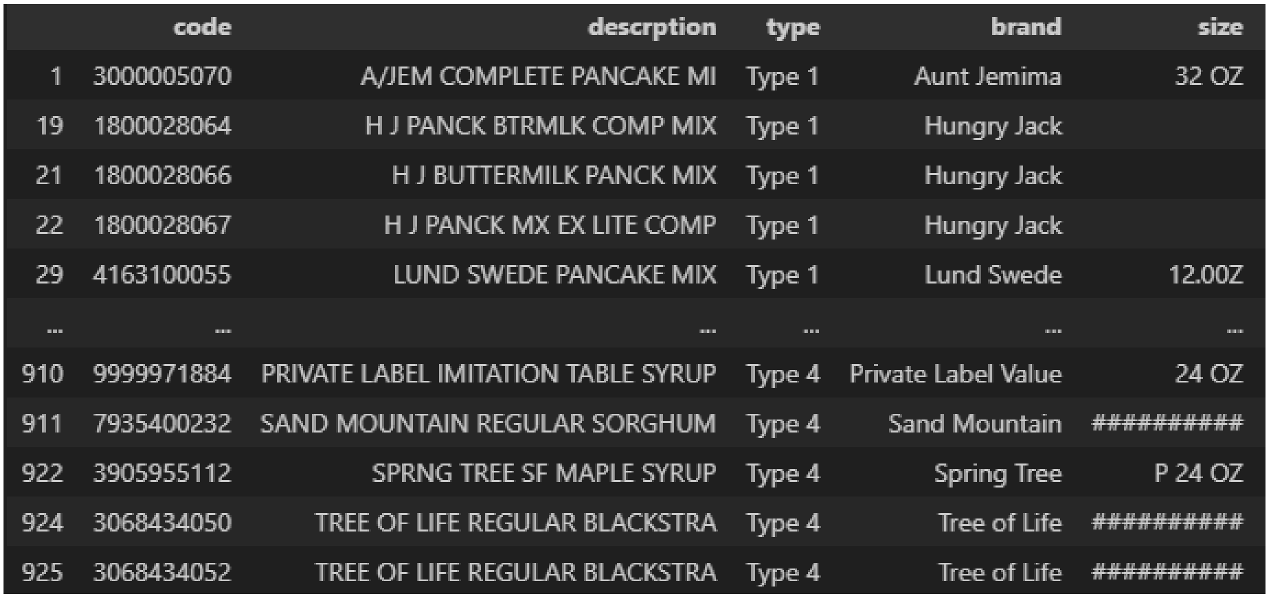


1.2 Data Cleaning and Pre processing

There are no missing or duplicate values in the datasets. However, unrealistic values, duplicate classes, and incorrect data types used in certain variables were identified in some instances. These issues are addressed as follows.

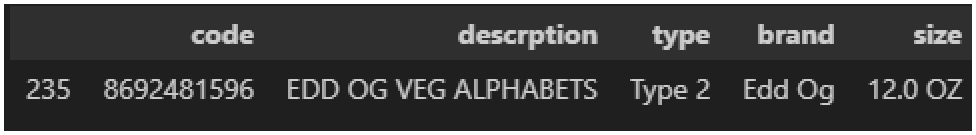
1.2.1 Cleaning in Item dataset

First, attention is directed to the *size* variable. The expected format of this variable is **"[number]<space> [unit]"**. However, it has been observed that 138 instances deviate from this expected format. Consequently, these instances must be corrected to adhere to the proper structure. Given the variations in the formats of these values, manual adjustments are required.

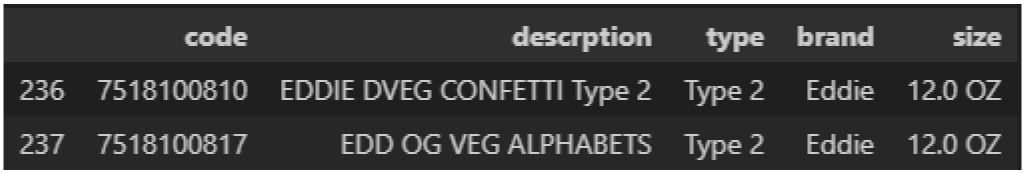


	code	descrption	type	brand	size
1	3000005070	A/JEM COMPLETE PANCAKE MI	Type 1	Aunt Jemima	32 OZ
19	1800028064	H J PANCK BTRMLK COMP MIX	Type 1	Hungry Jack	
21	1800028066	H J BUTTERMILK PANCK MIX	Type 1	Hungry Jack	
22	1800028067	H J PANCK MX EX LITE COMP	Type 1	Hungry Jack	
29	4163100055	LUND SWEDE PANCAKE MIX	Type 1	Lund Swede	12.0OZ
...	...	...	...	...	...
910	9999971884	PRIVATE LABEL IMITATION TABLE SYRUP	Type 4	Private Label Value	24 OZ
911	7935400232	SAND MOUNTAIN REGULAR SORGHUM	Type 4	Sand Mountain	##########
922	3905955112	SPRNG TREE SF MAPLE SYRUP	Type 4	Spring Tree	P 24 OZ
924	3068434050	TREE OF LIFE REGULAR BLACKSTRA	Type 4	Tree of Life	##########
925	3068434052	TREE OF LIFE REGULAR BLACKSTRA	Type 4	Tree of Life	##########

Next, the focus shifts to the *brand* variable, which contains 131 unique brand names within the dataset. An examination was conducted to determine whether the same brand name appeared in different formats, resulting in the identification of one such instance.

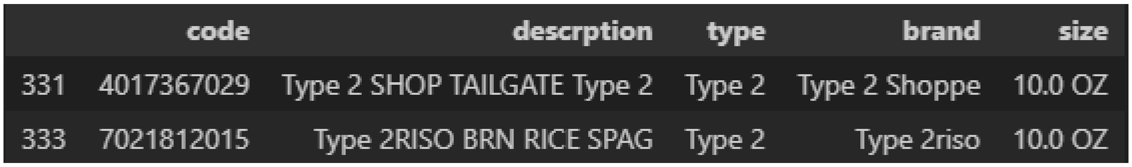


	code	descrption	type	brand	size
235	8692481596	EDD OG VEG ALPHABETS	Type 2	Edd Og	12.0 OZ



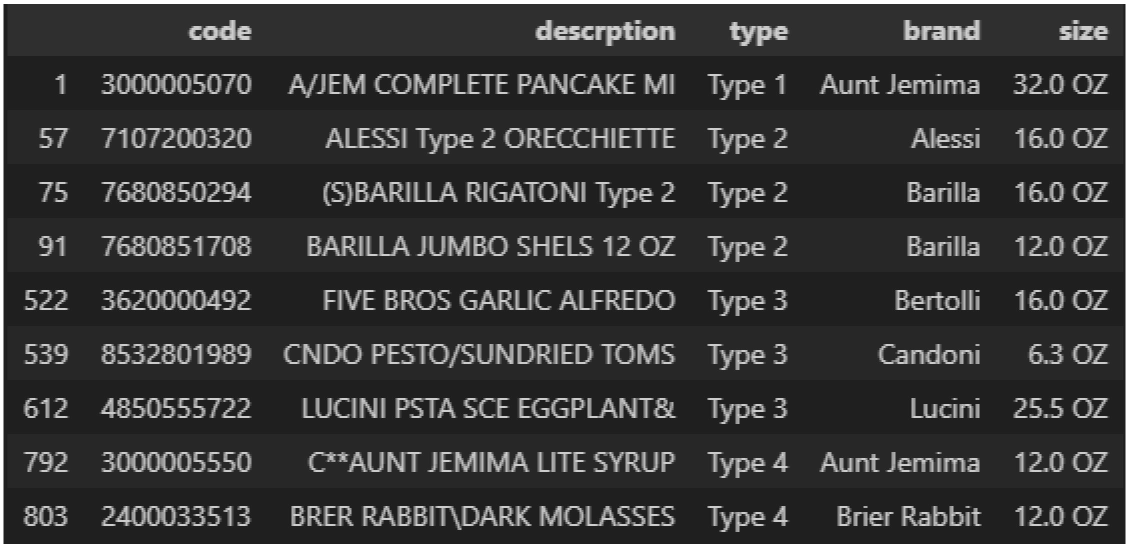
	code	descrption	type	brand	size
236	7518100810	EDDIE DVEG CONFETTI Type 2	Type 2	Eddie	12.0 OZ
237	7518100817	EDD OG VEG ALPHABETS	Type 2	Eddie	12.0 OZ

Consequently, "Edd Og" is amended to "Eddie." Additionally, it was observed that several brand names were prefixed with "Type 2." This discrepancy is clearly a data entry error, as these values pertain to the *type* variable located adjacent to the *brand* variable. Therefore, these entries have been rectified accordingly.



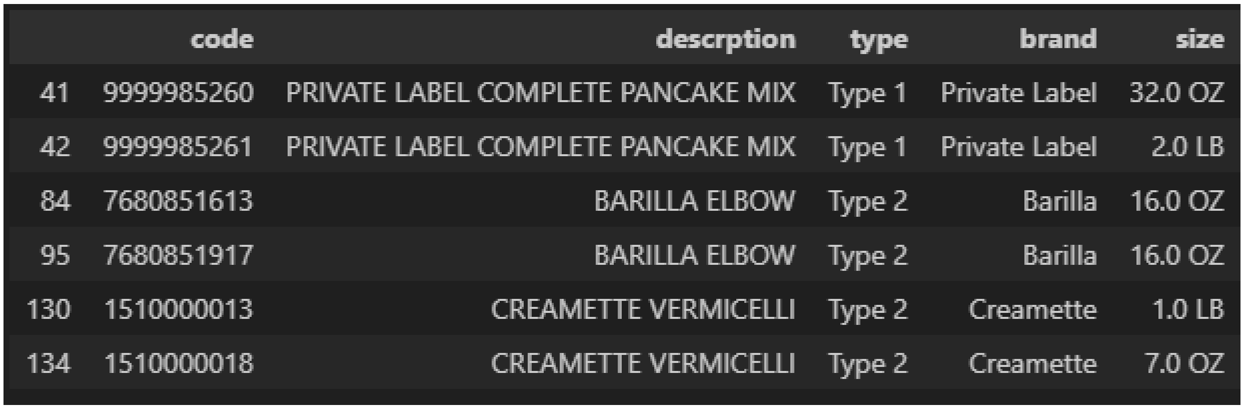
	code	descrption	type	brand	size
331	4017367029	Type 2 SHOP TAILGATE Type 2	Type 2	Type 2 Shoppe	10.0 OZ
333	7021812015	Type 2RISO BRN RICE SPAG	Type 2	Type 2riso	10.0 OZ

Subsequently, the focus shifts to the *description* variable. It is observed that the typical structure of the description follows the format: **"<Company Name><Product Name>"**. However, several descriptions deviate from this pattern. To maintain a consistent structure, these descriptions are adjusted accordingly.



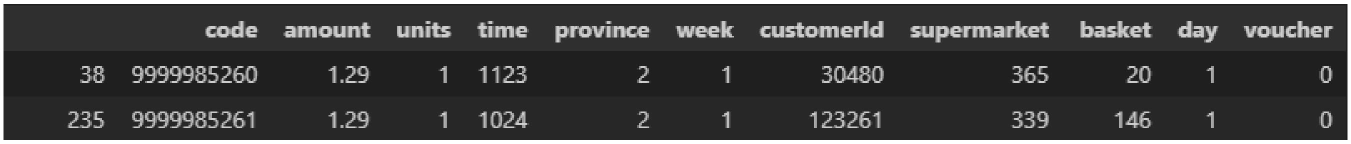
	code	descrption	type	brand	size
1	3000005070	A/JEM COMPLETE PANCAKE MI	Type 1	Aunt Jemima	32.0 OZ
57	7107200320	ALESSI Type 2 ORECCHIETTE	Type 2	Alessi	16.0 OZ
75	7680850294	(S)BARILLA RIGATONI Type 2	Type 2	Barilla	16.0 OZ
91	7680851708	BARILLA JUMBO SHELS 12 OZ	Type 2	Barilla	12.0 OZ
522	3620000492	FIVE BROS GARLIC ALFREDO	Type 3	Bertolli	16.0 OZ
539	8532801989	CNDO PESTO/SUNDRIED TOMS	Type 3	Candoni	6.3 OZ
612	4850555722	LUCINI PSTA SCE EGGPLANT&	Type 3	Lucini	25.5 OZ
792	3000005550	C**AUNT JEMIMA LITE SYRUP	Type 4	Aunt Jemima	12.0 OZ
803	2400033513	BRER RABBIT\DARK MOLASSES	Type 4	Brier Rabbit	12.0 OZ

It was observed that certain item codes share identical descriptions, as outlined below.



	code	descrption	type	brand	size
41	9999985260	PRIVATE LABEL COMPLETE PANCAKE MIX	Type 1	Private Label	32.0 OZ
42	9999985261	PRIVATE LABEL COMPLETE PANCAKE MIX	Type 1	Private Label	2.0 LB
84	7680851613	BARILLA ELBOW	Type 2	Barilla	16.0 OZ
95	7680851917	BARILLA ELBOW	Type 2	Barilla	16.0 OZ
130	1510000013	CREAMETTE VERMICELLI	Type 2	Creamette	1.0 LB
134	1510000018	CREAMETTE VERMICELLI	Type 2	Creamette	7.0 OZ

While the item codes and sizes may differ, the codes are assigned uniquely to distinguish between them. A closer examination of the sizes reveals that they are often equivalent, but represented in different units. For instance, in the first row above, 32 ounces (Oz) is equivalent to 2 pounds (LB). This suggests that these entries represent the same item but were recorded under different codes. This conclusion is further supported by analyzing the prices, which appear to be identical on specific days, confirming the initial assumption.



	code	amount	units	time	province	week	customerId	supermarket	basket	day	voucher
38	9999985260	1.29	1	1123	2	1	30480	365	20	1	0
235	9999985261	1.29	1	1024	2	1	123261	339	146	1	0

It is important to note **that some items share the same description but differ in size**. Therefore, it is crucial to accurately distinguish these items and avoid making any changes to them. Therefore, the initial step involves converting all values from pounds (LB) to ounces (OZ). Subsequently, similar items are identified and consolidated under a single code. (code which first occurred) Then it is necessary to update these codes in both the *sales* and *promotion* datasets, as the *code* variable is present in both datasets.

1.2.2 Cleaning in Promotion dataset

The *promotion* dataset does not require any cleaning. However, it is essential to verify whether there are any values for the *code* variable that are not present in the *item* dataset. Upon conducting this check, it was determined that no such discrepancies exist.

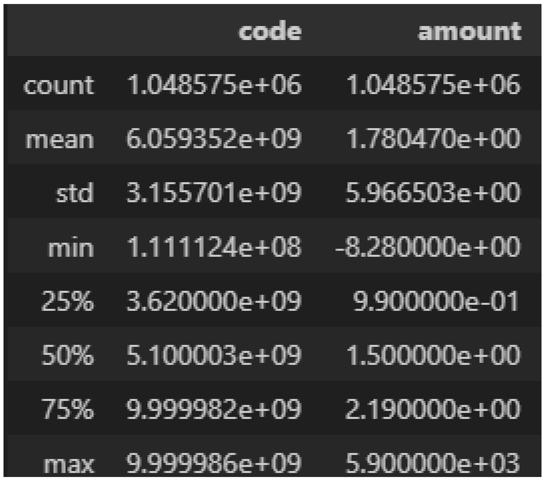
1.2.3 Cleaning in Supermarket dataset

The *supermarket* dataset does not require any cleaning.

1.2.4 Cleaning in Sales dataset

Initially, a verification was conducted to determine whether any *code* values existed in the dataset that were absent from the *item* dataset, as well as whether any *supermarket* values were missing from the *supermarkets* dataset. The results indicated that no such discrepancies were present.

Next, an observation was made regarding the presence of negative values in the *amount* variable. Given that this variable represents currency, negative values are not permissible. Consequently, it is necessary to impute these values to ensure accuracy and consistency within the dataset.



	code	amount
count	1.048575e+06	1.048575e+06
mean	6.059352e+09	1.780470e+00
std	3.155701e+09	5.966503e+00
min	1.111124e+08	-8.280000e+00
25%	3.620000e+09	9.900000e-01
50%	5.100003e+09	1.500000e+00
75%	9.999982e+09	2.190000e+00
max	9.999986e+09	5.900000e+03

There are 2,426 instances with negative amount values. It is observed that the amount values can vary due to the influence of factors such as day, province, and voucher variables.

Therefore, when selecting a replacement for these negative values, a systematic approach was employed: first, the unit amounts for a given item were identified based on the specific day, province, and the presence or absence of a voucher. The mode of these values was then selected and multiplied by the number of units to replace the negative values. If no suitable replacement value satisfying the above conditions could be identified, the negative value was replaced with NaN.

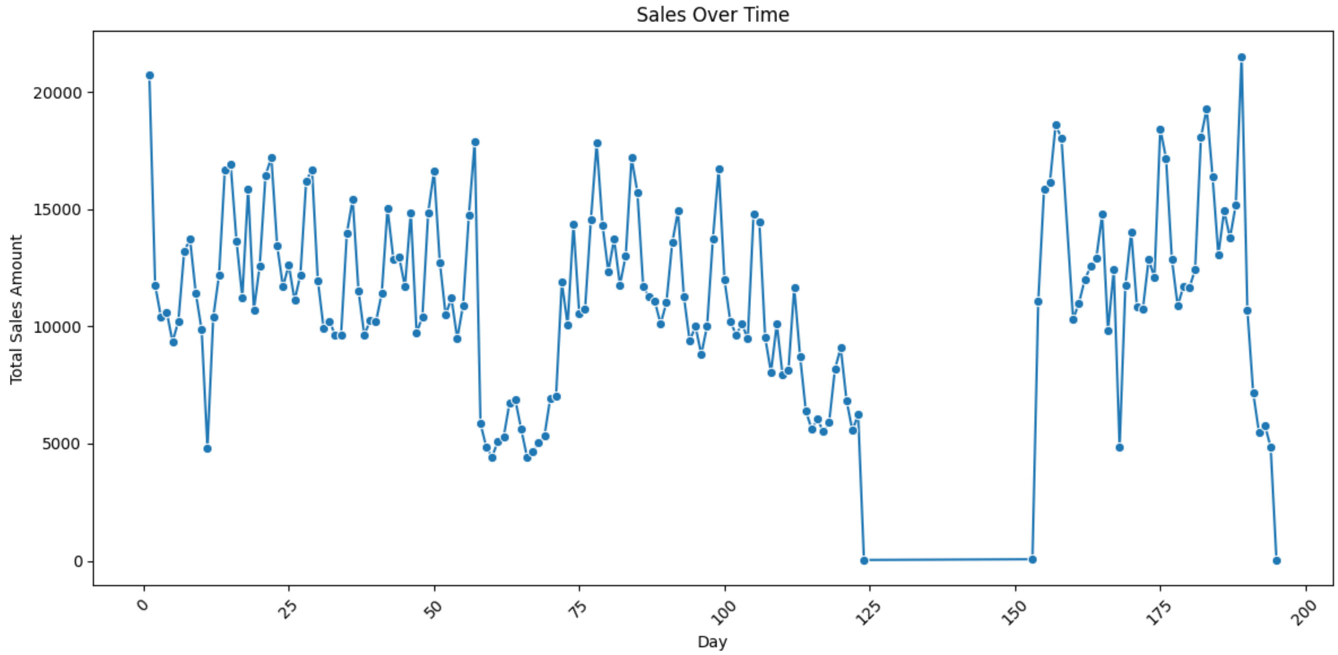
Ultimately, it was determined that there was insufficient information to impute 471 values, representing 0.05% of the entire dataset. As a result, it was decided to remove these instances.

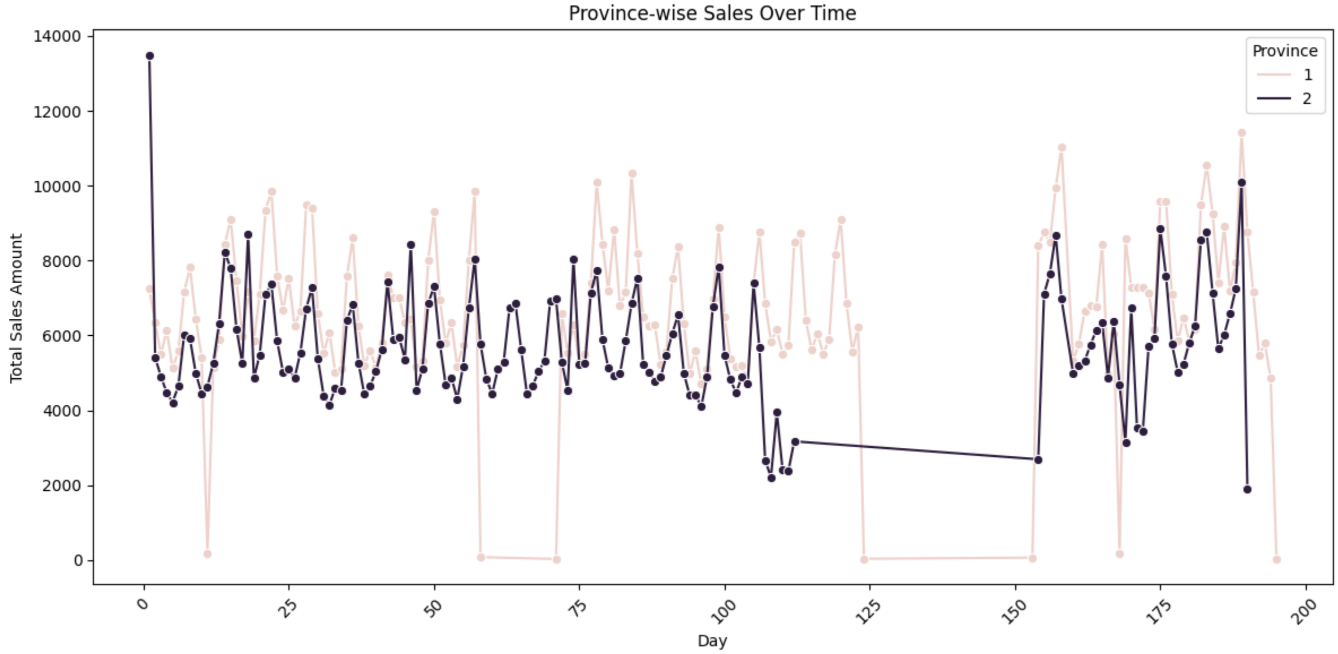
1.3 Business-valued solutions

1.3.1 Sales Forecasting Model

1.3.1.1 Cleaning and Preprocessing

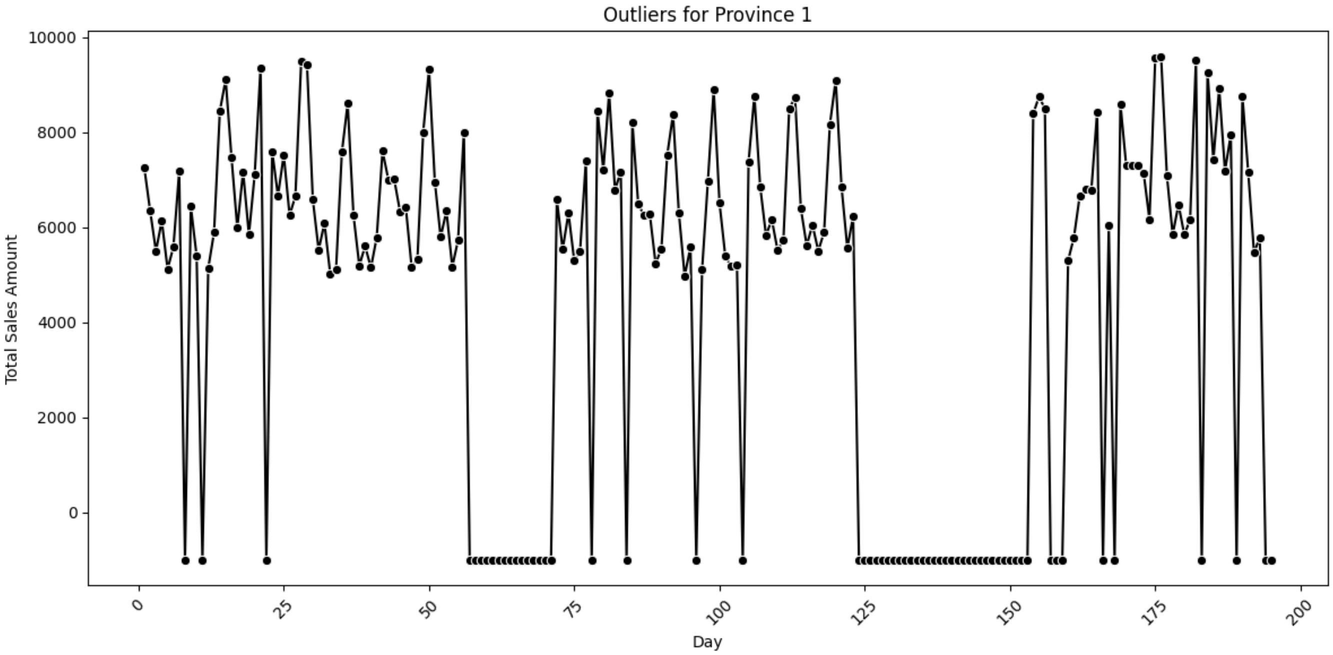
The main objective is to create sales forecasting model to predict sales for the upcoming month. The following graphs illustrate total sales and the variation in sales across provinces over time.





It can be observed that there are days with missing sales data. Since this may impact the prediction model, it is important to impute these missing values. Additionally, some clear outliers are identifiable in the data. Therefore, identifying and handling other potential outliers is equally important.

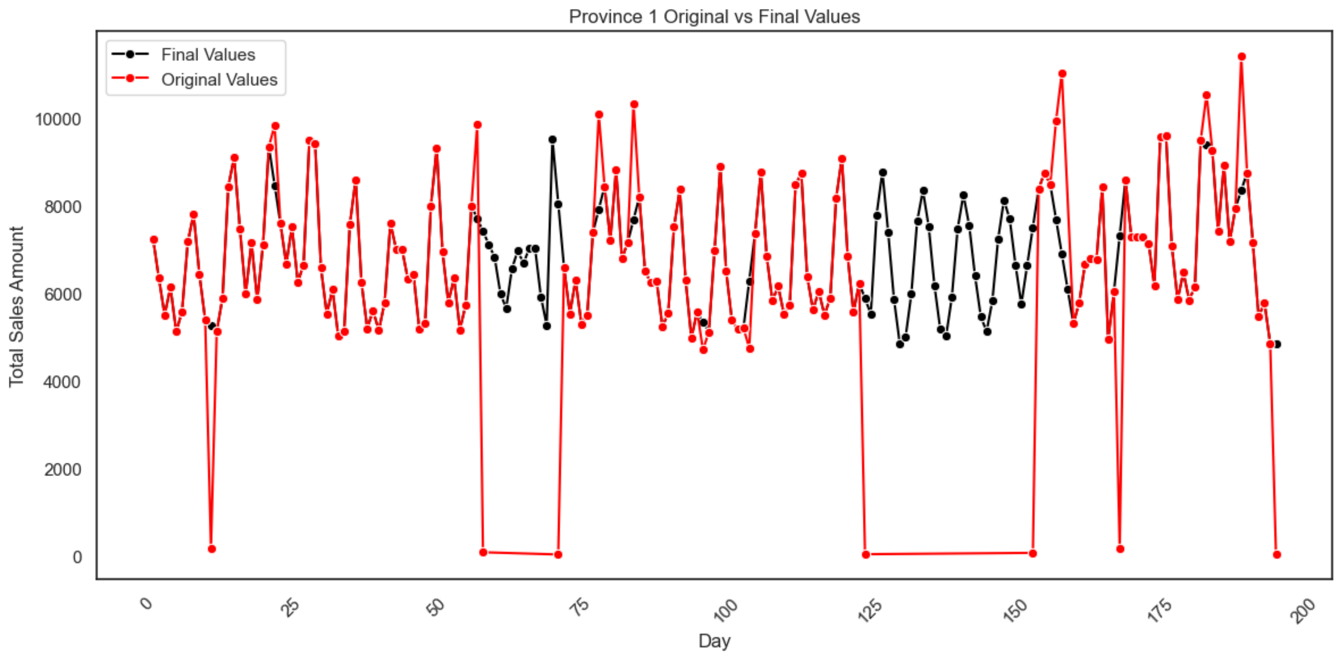
Outliers were identified using the **Isolation Forest (IForest)** method. Below, the detected outliers are recorded with values marked as -999. This method is applied province-wise.

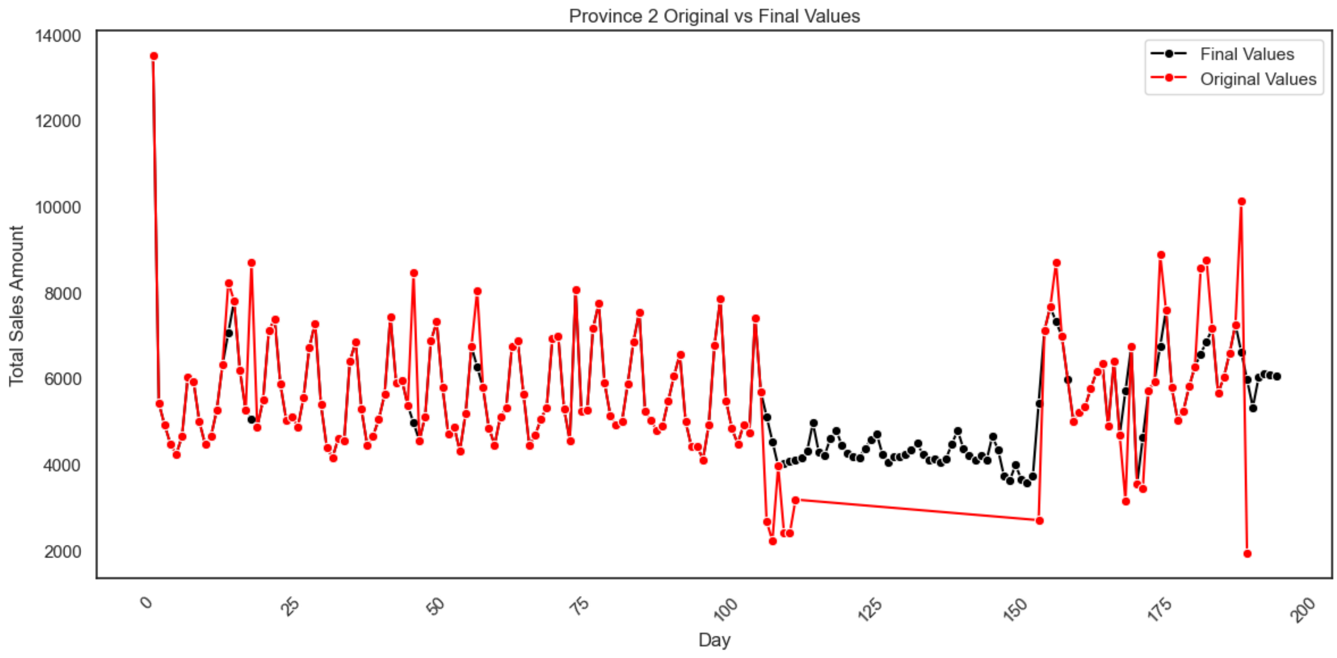


For handle those outliers we use 2 techniques.

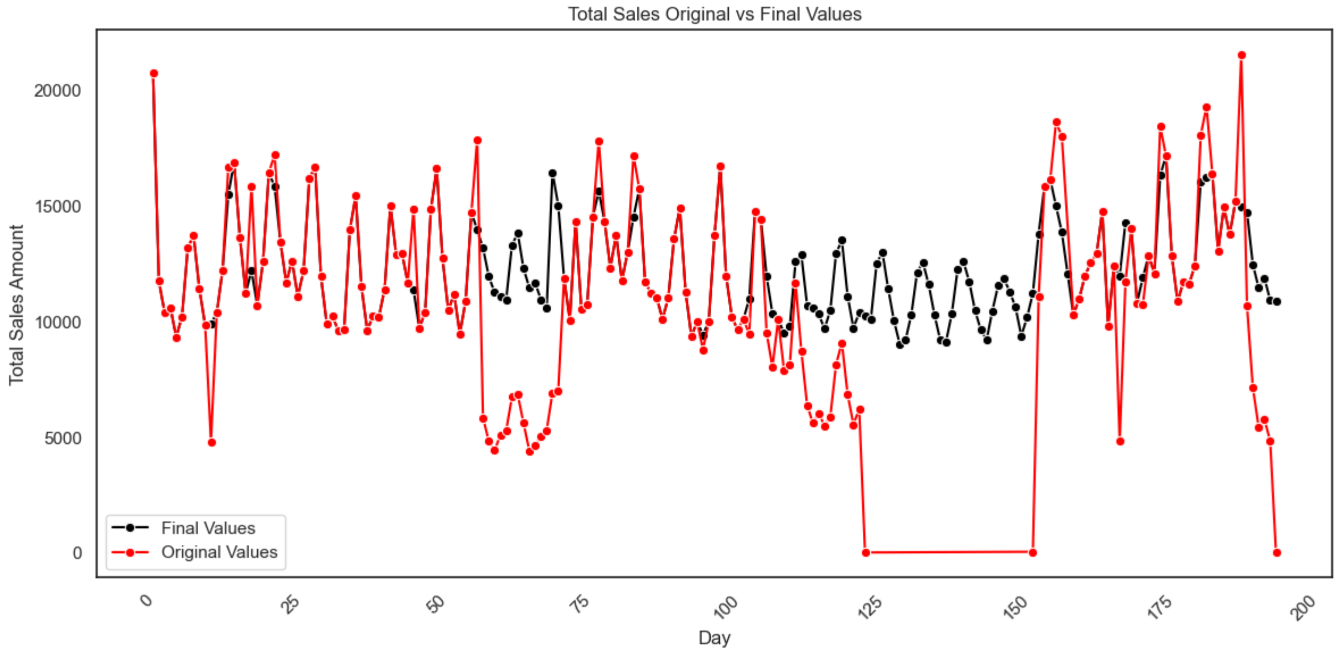
1) Use time-based interpolation for handle isolated outliers
2) Use ARIMA model for handle grouped outliers.

After handling the outliers, the province-wise data appears as follows. It can be observed that the dataset is now more generalized compared to the earlier version.


Province 1 Original vs Final Values
Final Values
Original Values
Total Sales Amount
10000
8000
6000
4000
2000
0
0
25
50
75
100
125
150
175
200
Day

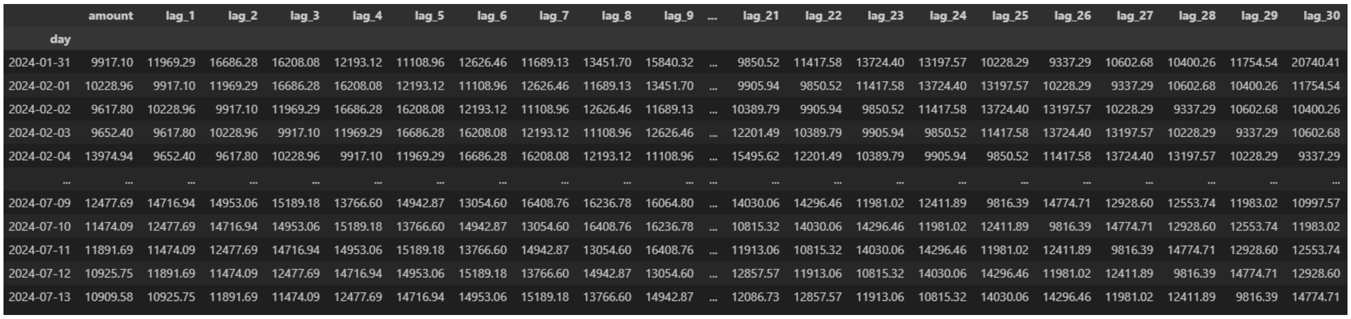

Province 2 Original vs Final Values
Final Values
Original Values
Total Sales Amount
14000
12000
10000
8000
6000
4000
2000
0
25
50
75
100
125
150
175
200
Day

The final dataset used for the forecasting model is presented as follows



1.3.1.2 Model Building

Since the dataset only contains total sales as a variable, 30 lag features were created during the feature engineering step to enhance the forecasting model.



	amount	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	...	lag_21	lag_22	lag_23	lag_24	lag_25	lag_26	lag_27	lag_28	lag_29	lag_30
day																					
2024-01-31	9917.10	11969.29	16686.28	16208.08	12193.12	11108.96	12626.46	11689.13	13451.70	15840.32	...	9850.52	11417.58	13724.40	13197.57	10228.29	9337.29	10602.68	10400.26	11754.54	20740.41
2024-02-01	10228.96	9917.10	11969.29	16686.28	16208.08	12193.12	11108.96	12626.46	11689.13	13451.70	...	9905.94	9850.52	11417.58	13724.40	13197.57	10228.29	9337.29	10602.68	10400.26	11754.54
2024-02-02	9617.80	10228.96	9917.10	11969.29	16686.28	16208.08	12193.12	11108.96	12626.46	11689.13	...	10389.79	9905.94	9850.52	11417.58	13724.40	13197.57	10228.29	9337.29	10602.68	10400.26
2024-02-03	9652.40	9617.80	10228.96	9917.10	11969.29	16686.28	16208.08	12193.12	11108.96	12626.46	...	12201.49	10389.79	9905.94	9850.52	11417.58	13724.40	13197.57	10228.29	9337.29	10602.68
2024-02-04	13974.94	9652.40	9617.80	10228.96	9917.10	11969.29	16686.28	16208.08	12193.12	11108.96	...	15495.62	12201.49	10389.79	9905.94	9850.52	11417.58	13724.40	13197.57	10228.29	9337.29
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-07-09	12477.69	14716.94	14953.06	15189.18	13766.60	14942.87	13054.60	16408.76	16236.78	16064.80	...	14030.06	14296.46	11981.02	12411.89	9816.39	14774.71	12928.60	12553.74	11983.02	10997.57
2024-07-10	11474.09	12477.69	14716.94	14953.06	15189.18	13766.60	14942.87	13054.60	16408.76	16236.78	...	10815.32	14030.06	14296.46	11981.02	12411.89	9816.39	14774.71	12928.60	12553.74	11983.02
2024-07-11	11891.69	11474.09	12477.69	14716.94	14953.06	15189.18	13766.60	14942.87	13054.60	16408.76	...	11913.06	10815.32	14030.06	14296.46	11981.02	12411.89	9816.39	14774.71	12928.60	12553.74
2024-07-12	10925.75	11891.69	11474.09	12477.69	14716.94	14953.06	15189.18	13766.60	14942.87	13054.60	...	12857.57	11913.06	10815.32	14030.06	14296.46	11981.02	12411.89	9816.39	14774.71	12928.60
2024-07-13	10909.58	10925.75	11891.69	11474.09	12477.69	14716.94	14953.06	15189.18	13766.60	14942.87	...	12086.73	12857.57	11913.06	10815.32	14030.06	14296.46	11981.02	12411.89	9816.39	14774.71

Several machine learning and deep learning models were fitted to forecast sales. The results of these models are presented below.

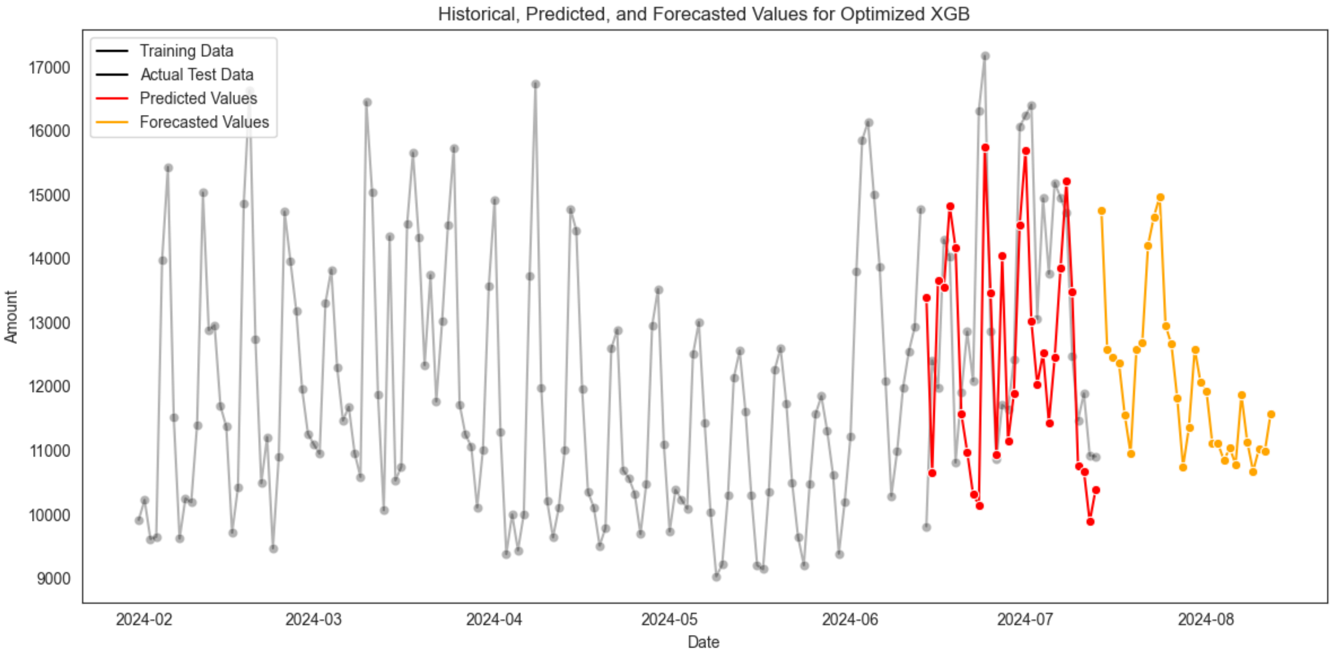
The primary models used for forecasting include Linear Regression, K-Nearest Neighbors, Support Vector Machines, Random Forest, and XGBoost.

Model	Training MAE	Test MAE
SVM	1573.19	2061.69
Linear Regression	932.03	1657.21
XGB	1142.12	1643.20
Random Forest	1056.85	1615.15
KNN	939.20	1257.75

The three best models, based on Test MAE, were selected, and their hyperparameters were optimized using **Optuna**.

Optimized Model	Training MAE	Test MAE
KNN	999.41	1585.90
Random Forest	1051.31	1507.61
XGB	1133.82	1346.22

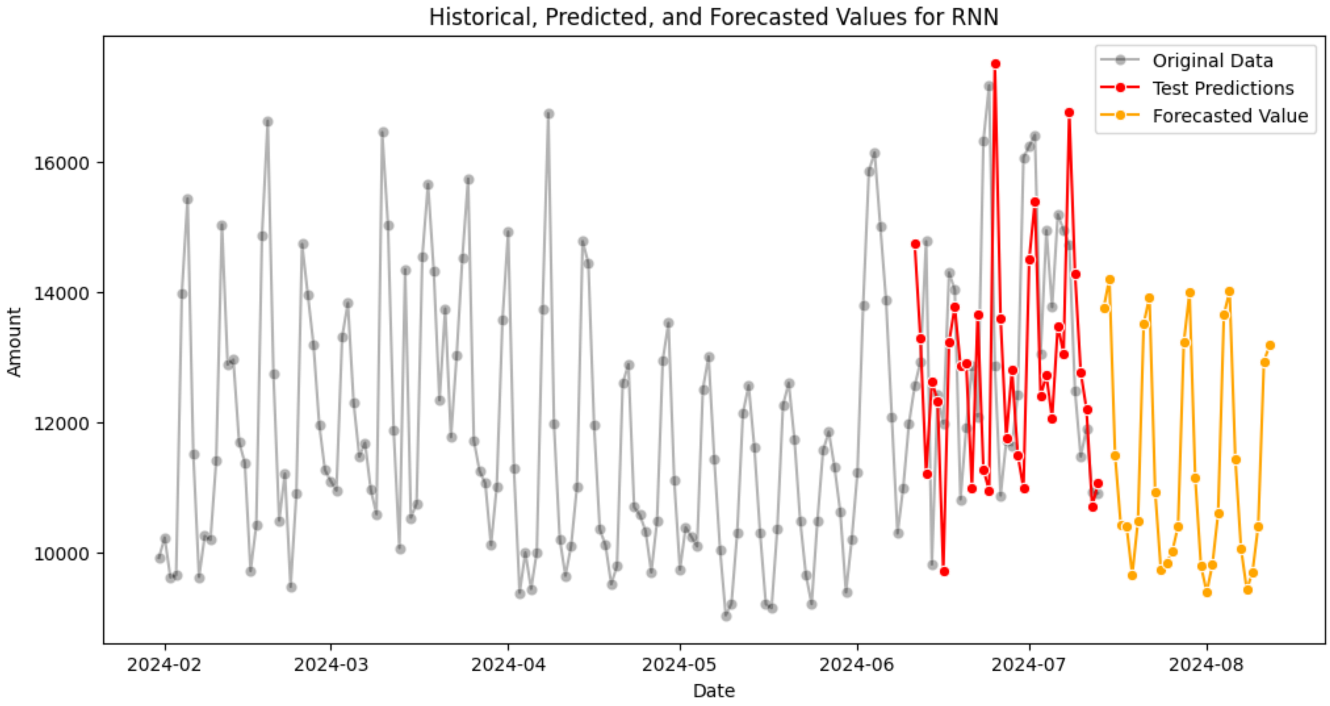
According to the results, the base KNN model has the lowest Test MAE. However, the optimized XGBoost model shows the smallest difference between train and test MSE, along with the second lowest Test MAE. Therefore, the optimized XGBoost model is selected as the best model.



The deep learning models utilized include MLP, RNN, LSTM, GRU, CNN, Transformer, and LSTM+CNN.

Model	Training MAE	Test MAE
LSTM	756.56	2194.71
LSTM + CNN	423.57	2084.12
Transformer	1568.53	1875.27
CNN	305.62	1785.14
MLP	295.57	1732.75
GRU	437.14	1716.16
RNN	243.03	1433.34

According to the results, the RNN model has the lowest Test MAE and the smallest difference between train and test MSE. Therefore, the RNN model is selected as the best model Deep Learning Model.

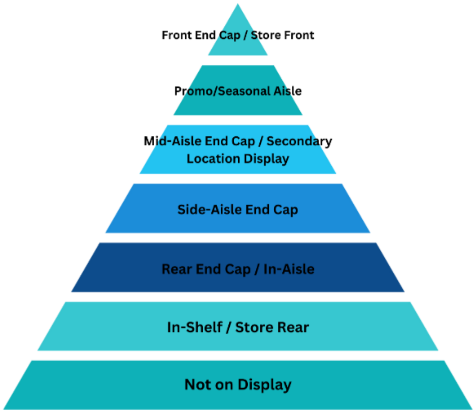


After reviewing the prediction patterns of the two best models, it can be concluded that the optimized XGBoost model captures irregular patterns more effectively compared to the RNN model. Therefore, the optimized XGBoost model is selected as the final model.

1.3.2 Effect on Display Variable

The goal is to determine whether the variable 'display' have an effect on sales. First, it is necessary to combine the sales and promotion datasets. However, it is important to note that the promotion dataset contains data from the 43rd to the 104th week, while the sales dataset contains data from the 1st to the 28th week. Despite this, it was observed that the values of the 'display' variable do not depend on the week (which is somewhat an unlikely scenario). Therefore, the time aspect is ignored, and the two datasets are combined based on the item code and province.

This means it is assumed that every item is displayed in the store is same (as discussed in the dataset description) throughout the entire period.



First, the display classes are divided into three main groups.

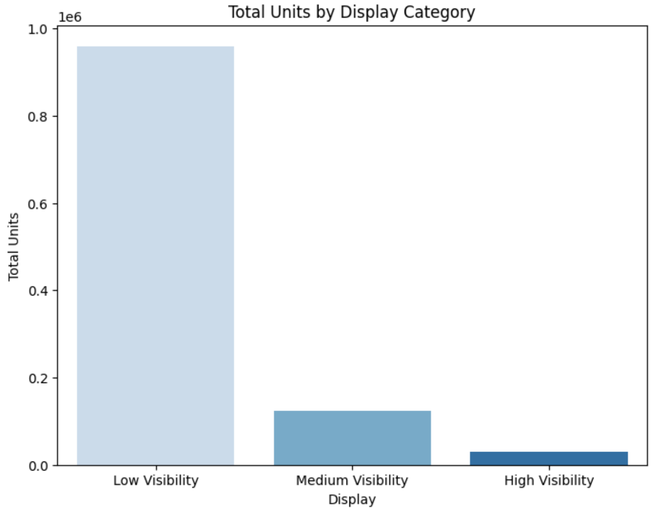
1. High Visibility: Front End Cap, Store Front, Promo/Seasonal Aisle

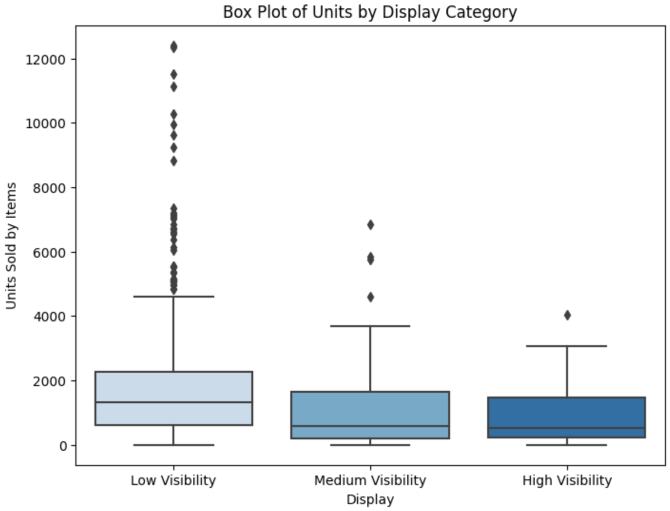
2. Medium Visibility: Mid-Aisle End Cap, Secondary Location Display, Side-Aisle End Cap, Rear End Cap

3. Low Visibility: Not on Display, In-Shelf, Store Rear, In-Aisle

Now, following three hypotheses will be tested.

1. Displaying products at High visible places will lead to higher unit sales or sales amount compared to displaying them at the Medium or Low visible places.
2. Displaying products at Medium visible places will not have any difference on unit sales or sales amount compared to displaying them at the High or Low visible places.
3. Displaying products at Low visible places will lead to lower unit sales or sales amount compared to displaying them at the Medium or High visible places.





According to the box plot and bar plot above, it can be observed that the lower visible class has higher unit sales, which is somewhat implausible. This discrepancy may be due to the imbalance in the dataset. Therefore, conducting hypothesis testing is crucial.

Hypothesis	P Value		
	≤	≠	≥
Hypothesis 01 (Amount)	0.005	7.40E-05	1.0
Hypothesis 01 (Units)	0.009	3.16E-04	0.99
Hypothesis 02 (Amount)	0.002	0.003	1.0
Hypothesis 02 (Units)	6.3E-04	1.2E-04	1.0
Hypothesis 03 (Amount)	1.0	1.8E-05	3.78E-05
Hypothesis 03 (Units)	1.0	4.92E-07	1.52E-05

According to these results we can conclude following thing at 95% confidence,

1. Displaying products at High visible places will lead to lower unit sales and sales amount compare to other places.
2. Displaying products at Medium visible places will lead to lower unit sales and sales amount compared to other places.
3. Displaying products at Low visible places will lead to higher unit sales and sales amount compared to other places

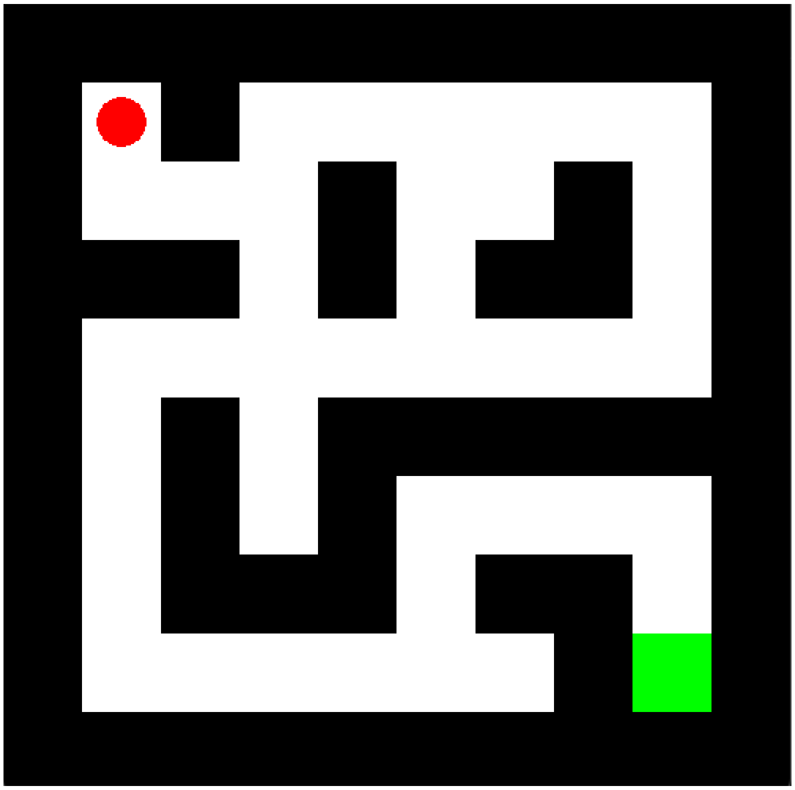
Therefore, the placement of items displayed in the store does not have a significant impact on sales, according to these findings.

2. Reinforcement Learning for Maze Game

2.1 Maze Layout

The maze is displayed using the pygame library, and the game window size is set to 500x500 pixels. The tiles are dynamically scaled based on the number of rows and columns. The game uses different colors to differentiate the elements:

- White for background
- Black for walls
- Green for the goal
- Red for the agent



2.2 Game Logic

The agent navigates the maze using actions that correspond to moving up, down, left, or right. The goal is to reach the designated endpoint while avoiding walls. The maze's boundary acts as walls, preventing the agent from going out of bounds.

2.3 DQN Agent

The agent is implemented in the Jupyter notebook Agent.ipynb, where a DQN model is used. The model is created using stable_baselines3, which simplifies the process of setting up and training the agent.

Key settings for the agent include:

- **Policy**: CnnPolicy is used, indicating that a convolutional neural network is applied to process the environment's observations.
- **Buffer Size**: 1000
- **Learning Starts**: 0, meaning learning starts immediately without waiting for any number of steps.
- **Training Environment**: The maze environment is wrapped using Monitor and DummyVecEnv to ensure compatibility with stable_baselines3.

2.4 Training the Agent

During training, the agent interacts with the environment and learns through a series of episodes. The DQN model logs progress into Tensorboard for monitoring, and during execution, it uses the cuda device (indicating GPU acceleration).

2.5 Problems and Challenges

- The algorithm's execution speed is a significant issue. The combination of the algorithm's computational demands and the environment’s complexity results in long training times.
- The available computing power is insufficient to train the agent fully. As a result, the agent was unable to complete its training, and the learning process was cut short.