CECS 551 Advanced Artificial Intelligence

Guided by

Dr.Mahshid Fardadi, PhD

Allen Bolourchi, PhD, MBA

Rahul Deo Vishwakarma – TA

Submitted by

Arpitha Hiresadrahalli Dayananda - 029337982

Keval Dharmendra Joshi - 029364333

Soujanya Mulakalapally - 029343182

Varshini Nathala- 029334355

Kaushal Brahmbhatt-029355155

CONTENTS

DELIVERABLES

Resources :

1. Collab link for dataset 1 sprint 1+2.
2. Collab link for dataset 2 sprint 1+2+3.
3. Drive link to access dataset 2.
4. Report for sprint 1, 2 and 3.
5. PPT for presentation.

NAME	INDIVIDUAL CONTRIBUTION
Arpitha Hiresadrahalli Dayananda	Dataset 1- Sprint1-1)b,1)d||Sprint2-1)a-Linear,ARIMA,XGboost,1)c,2)a-Ensemble,RNN,2)b. Sprint3-2)b,c|| Report and PPT preparation.
Keval Dharmendra Joshi	Dataset 1- Sprint1-1)c,1)e|| Sprint2-1)b-Ridge Regression(0.99),1)c. 2)a-CNN,2)b|| Sprint3-2)b,c||Report and PPT preparation.
Soujanya Mulakalapally	Dataset 1- Sprint1- 1)a||Sprint2-1) b-Ridge Regression(17),1)c,2)a-RNN,2)b|| Sprint3-2)b,c|| Report and PPT preparation.
Varshini Nathala	Dataset 2- Sprint1- 1,2|| Sprint2- a,b ||Sprint3-2)a|| Report and PPT preparation.
Kaushal Brahmbhatt	Dataset 2-Sprint2- C||Sprint3-1|| Report and PPT preparation.

INTRODUCTION

Analyzation inventory data of two datasets of around more than 30 stores of an international retail business. The purpose of the analysis is to use the inventory data to improve sales, resulting in a more efficient operation

.

- **dataset 01**: The task is to predict the department-wide sales for each store.
- **dataset 02**: The goal is to predict the *unit* sales of each product for the next 10 days from 10 different stores across various states.

PROBLEM STATEMENT:

Sprint 1: Data Visualization and analysis

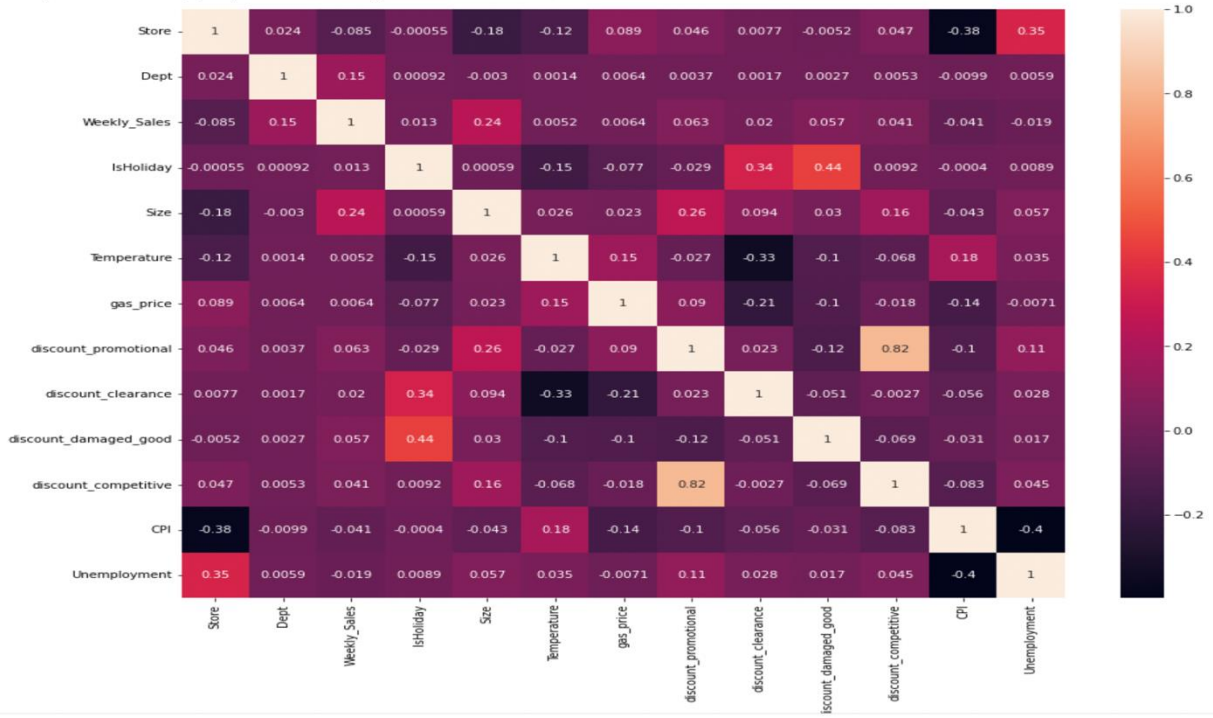
DATASET 01

Perform exploratory data analysis using Python for dataset_01. Please use the right data visualization method for the specific problem statement (choose the right chart type, for example, boxplot, histogram, scatterplot, pie-chart, etc.).

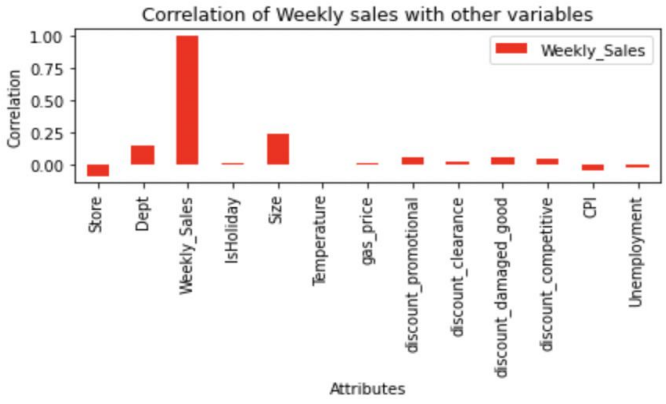
A. Identify the key variables for the model using correlation plots, heatmaps, histograms, and feature importance (SHAP).

Started with a correlation plot and heat map and the graph obtained is as below:

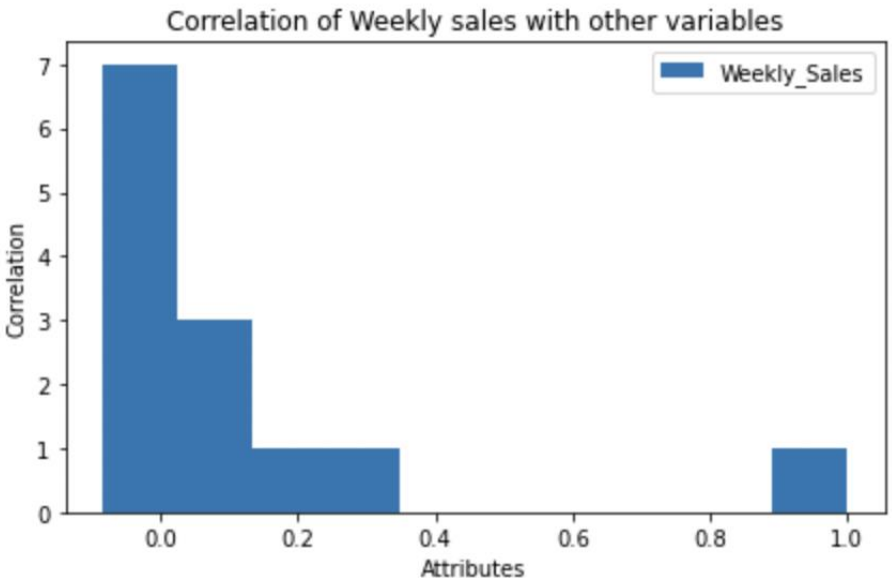
<matplotlib.axes._subplots.AxesSubplot at 0x7fc32c0a7ad0>



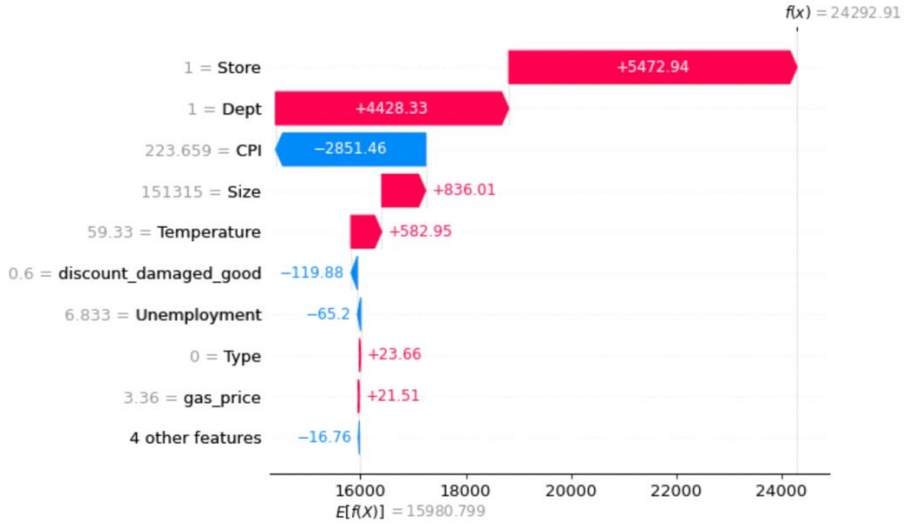
Plotted another view of correlation plot of weekly sales w.r.t other columns as below:



Correlation bar graph view of weekly sales with other columns is as below:

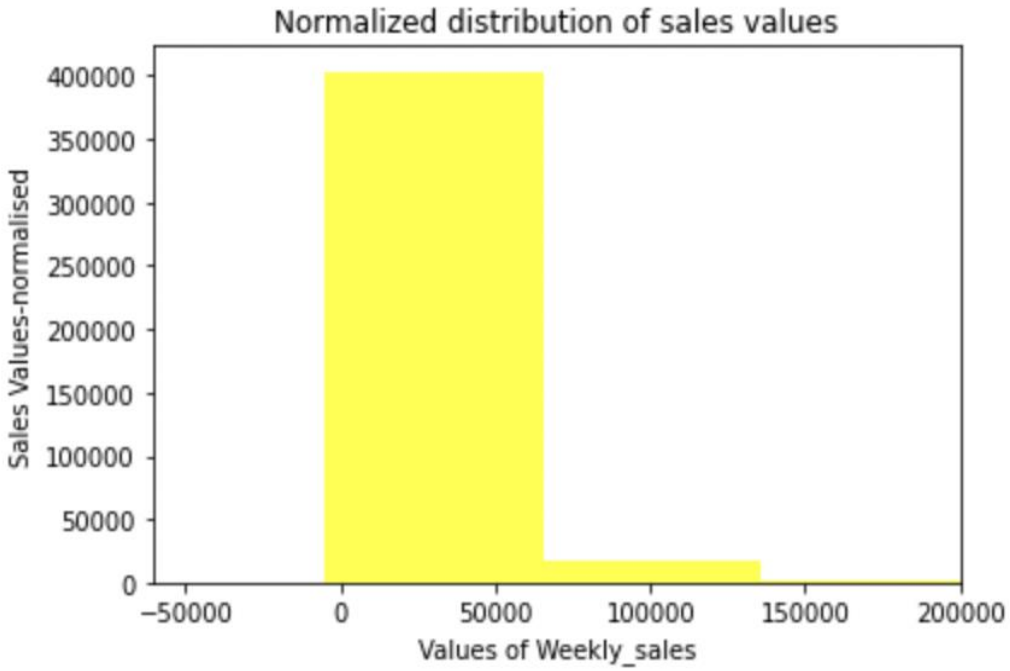


Feature Importance (SHAP) plot:



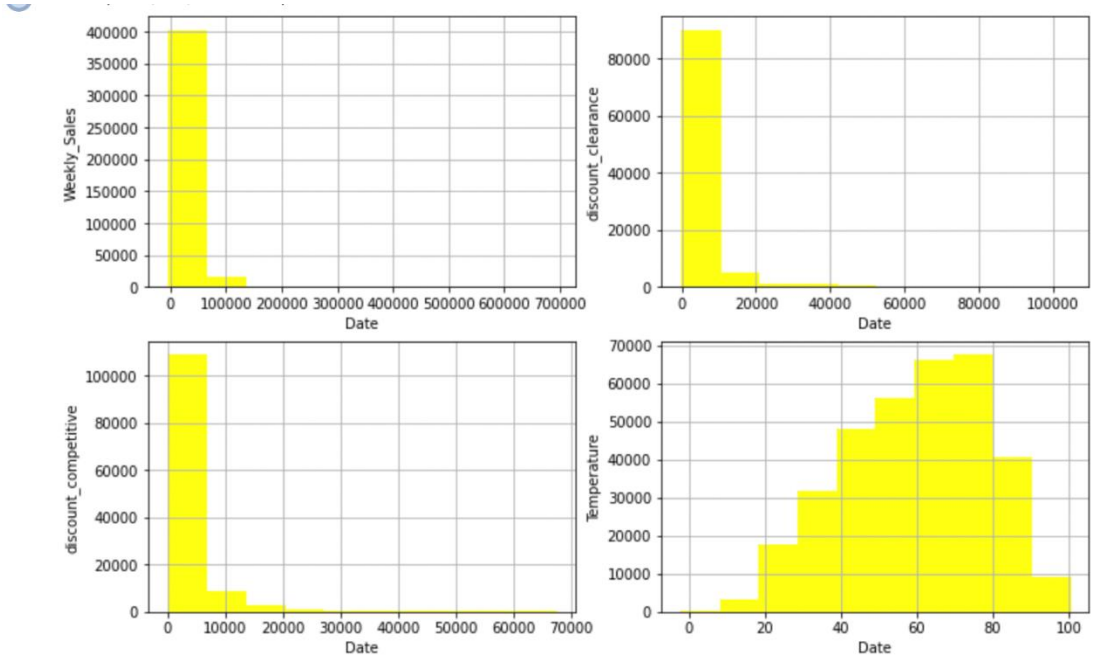
Histogram analysis:

Weekly sales analysis with respect to normalized sales can be seen below:

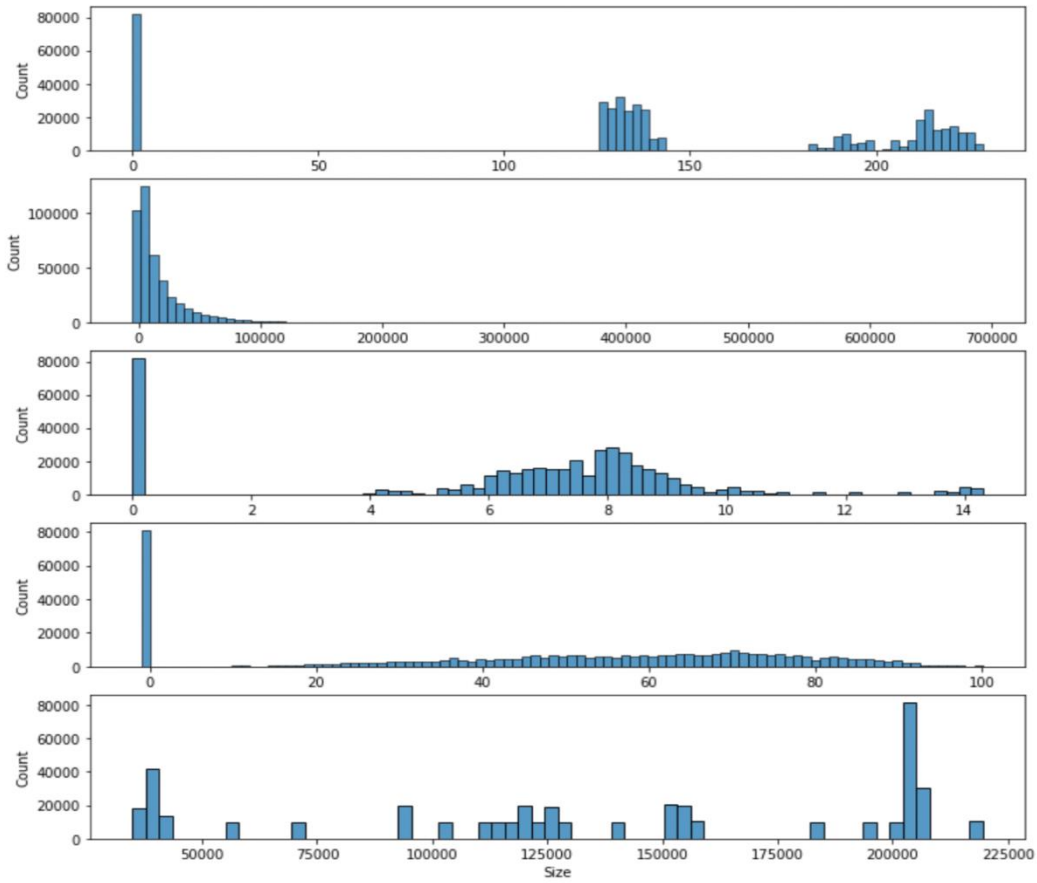


As can be seen, there are some erroneous recordings of sales. The sales can not be negative for a particular month, as illustrated in the graph.

Below is the histogram analysis of the date column with other columns like discount_clearance, discount _competitive, Temperature, gas_price and unemployment:



Below shows the histogram analysis of each column of our dataset:



From Heat map or correlation plot we can infer below key variables:
With respect to weekly-sales-only CPI, unemployment has negative values so we will drop those columns as there is no impact. So, the potential columns are store, dept, weekly_sales, Isholiday,Size,Temperature,gas_price,discount_promotional,discount_clearence,iscount_damaged_good,discount_competitive

From SHAP we can infer below key variables:

From the Shap graph key columns like store, department cpi, size, and Temperature have contributed more and the other columns unemployment, type, gas_price, and other features contributed less which can be dropped.

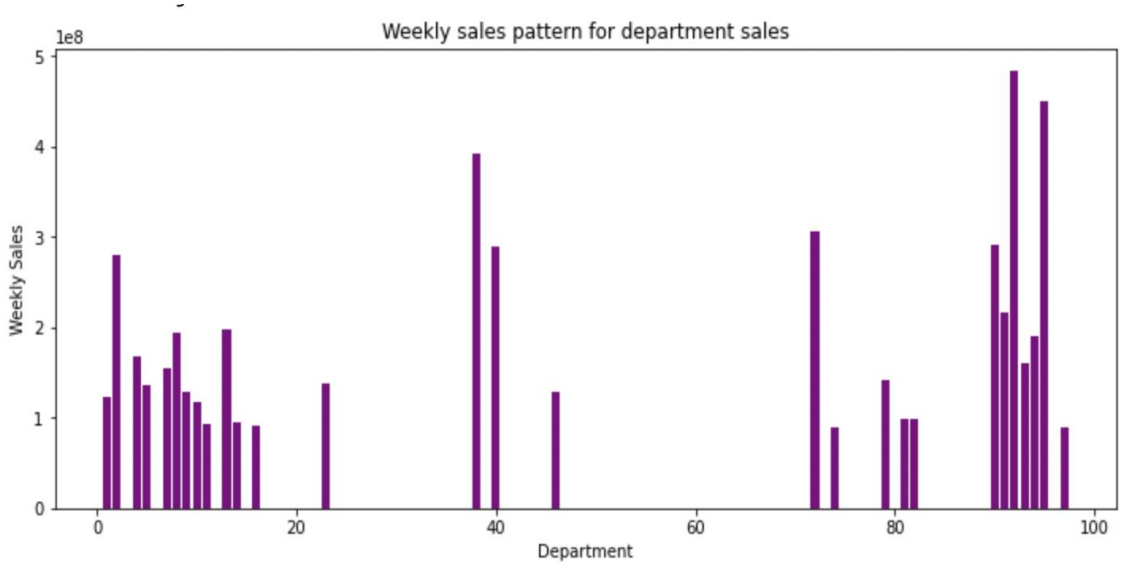
From the Histogram we can infer below key variables:

In the histogram, the columns which are skewed(outliers) cannot be considered as key variables and also the peaked ones across the distribution should also be dropped.
So the columns gas price, CPI, and Temperature should be dropped.

B. For the first 10 stores visualize the weekly and monthly sales patterns for top 35% of the department sales.

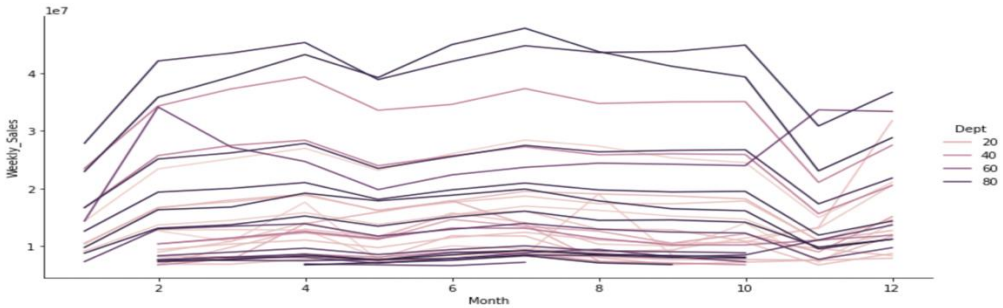
First 10 stores were initially taken as one data frame as analysis for first 10 stores. Then took the top 35% of department sales based on stores which has best performance in terms of sales (top 35% of best performing stores based on sales).

Weekly sales pattern:



As from above graph we can infer the weekly sales are high for departments between 80-100 and sales the next good weekly sales number can be seen between departments 20 and 40.

Monthly sales pattern:



From above graph we can infer that sales are high between month 6- 8
and next good monthly sales are between month 2-4 and 8-10. We can also see the monthly sales are also more for department 80.

B.1.Identify the best department "type" across the first ten stores.

483943341.87

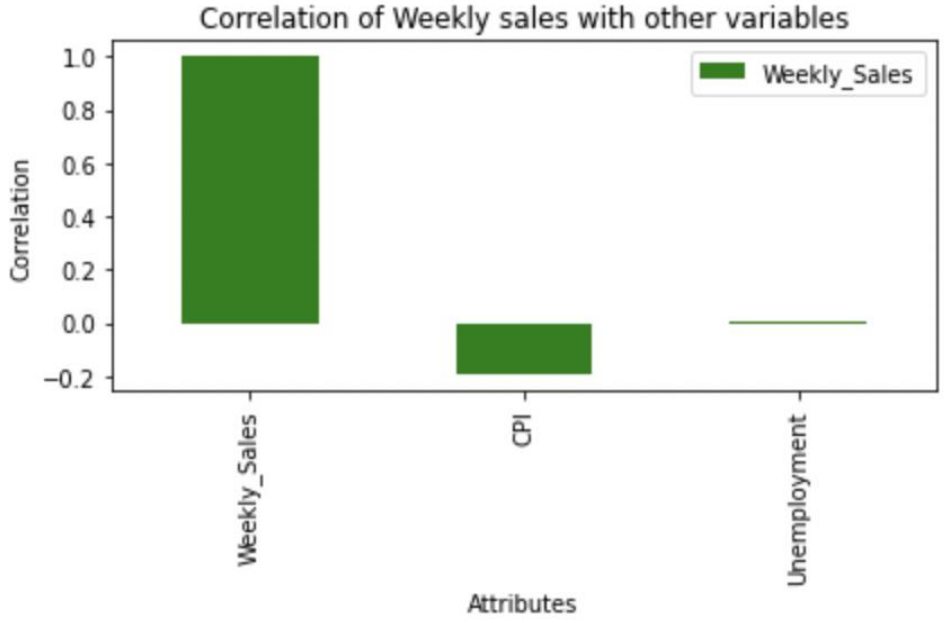
	Dept	Weekly_Sales
73	92	4.839433e+08

Department 92 is the best department among the first 10 stores as it has maximum sales weekly.

C. Investigate the relationship between weekly sales over CPI and unemployment for the first 10 stores. You can explore the what-if scenarios while writing the report.

To identify the relationship between weekly sales over CPI and unemployment for the first 10 stores we selected only those 3 columns and checked the correlation of CPI and unemployment w.r.t weekly_sales and got the graph as below:

	Weekly_Sales	CPI	Unemployment
Weekly_Sales	1.000000	-0.195452	0.002259
CPI	-0.195452	1.000000	-0.241247
Unemployment	0.002259	-0.241247	1.000000



From above correlation matrix, we can see that unemployment has negative or week relationship with weekly sales and CPI also has week relationship with weekly sales for first 10 stores.

To talk about what-if scenario:

If there were more sales, there would have been more job opportunities and then there would have been less unemployment rate and there would have been a positive correlation between them.

If there were more sales, there would be more profit so shop owners can reduce the price for their products and then the CPI would have increased and there would be a positive correlation between them.

D. Investigate the impact of various types of discounts, for example, discount promotional, discount clearance, discount damaged goods, discount competitive and discount employee on the overall sales.

D. 1. Which type of discount is helpful in increasing sales? Consider the top 30% of the best-performing stores (sales per 1000 square feet).

Here 30% of best-performing stores has been taken on the basis of their sales. The stores obtained are: 20, 4, 14, 13, 2, 10, 27, 6, 1, 39, 19, 31, 23.

Plotted graph for all of them to analyze the impact of discount promotional, discount clearance, discount damaged good, discount competitive and discount employee on overall sales as below:


Store2
1e7
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


Store20
1e7
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


Store4
1e7
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


Store14
1e7
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


Store13
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month

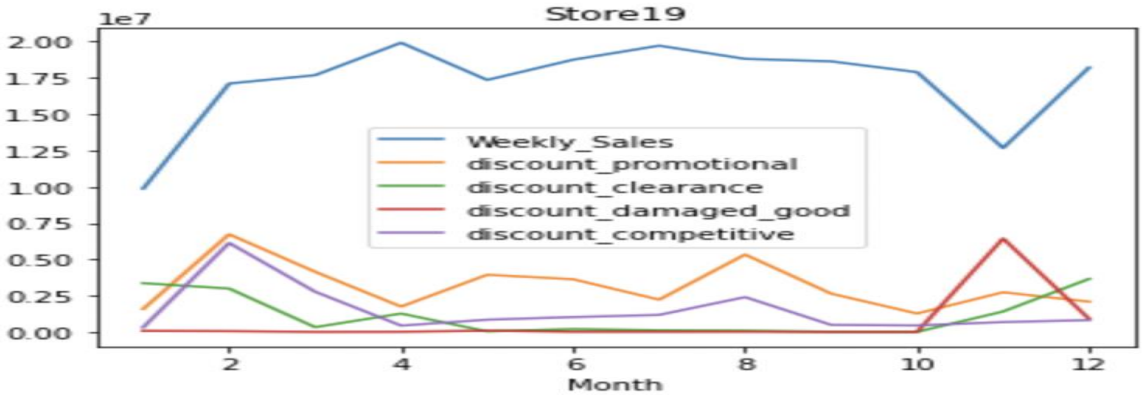

Store10
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store27
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store6
1e7
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store2
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store20
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store19
1e7
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store31
1e7
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month


Store23
1e7
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
2
4
6
8
10
12
Month

From the above graph we can infer that from starting couple of months discount promotional and discount competetive are followring trend with sales and with good discount of these two sales a re high and we can strongly say that increase in discount damaged goods reduced the sales and re cude in discount damaged good sales is increasing sales.

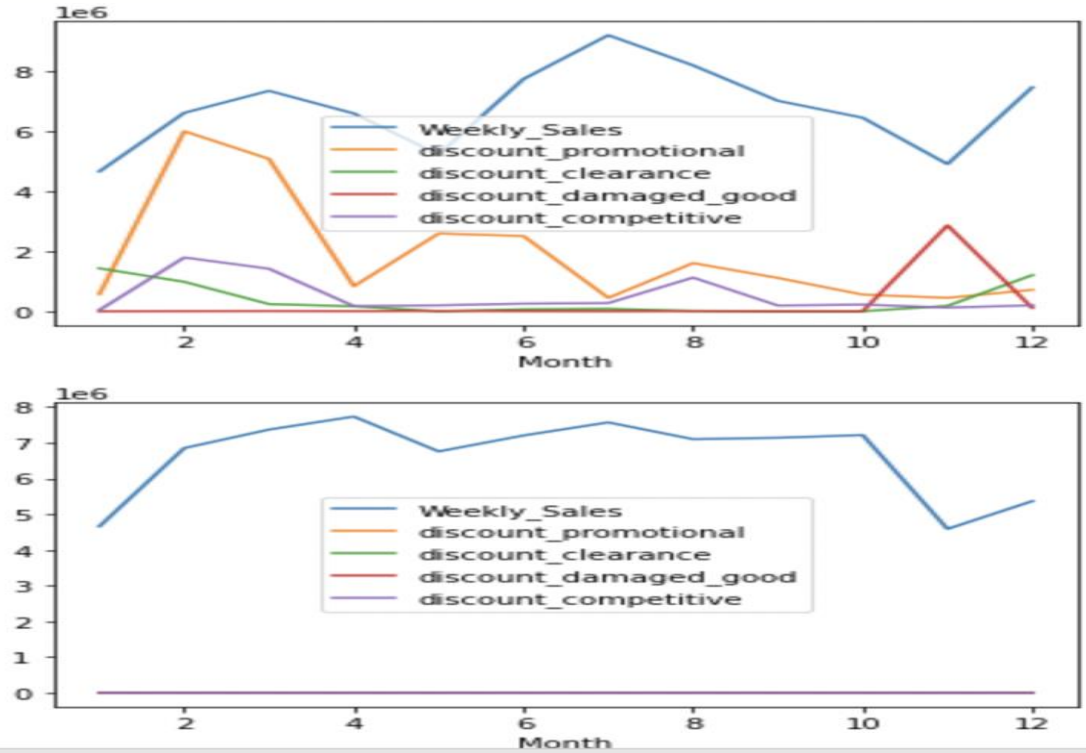
Discount clearance is always below for all the top 30% best performing stores when compared to discount promotional and discount competitive.

So narrowing down to discount promotional and discount competitive it is clearly seen mostly in all the top 30% best performing stores discount promotional is the
winner and this discount(discount promotional) helps in increased sales.

D.2. Does the observed behavior hold true for all the stores? Consider bottom 30% of the least performing store (sales per 1000 square feet).

Yes, the behavior holds good for the bottom 30% of least-performing stores which can be inferred from below graphs:

Here bottom 30% of least performing stores are taken on the basis of weekly sales (the stores with less sales are taken). The stores are:7, 42, 9, 29, 16, 37, 30, 3, 38, 36, 5, 44, 33




1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month

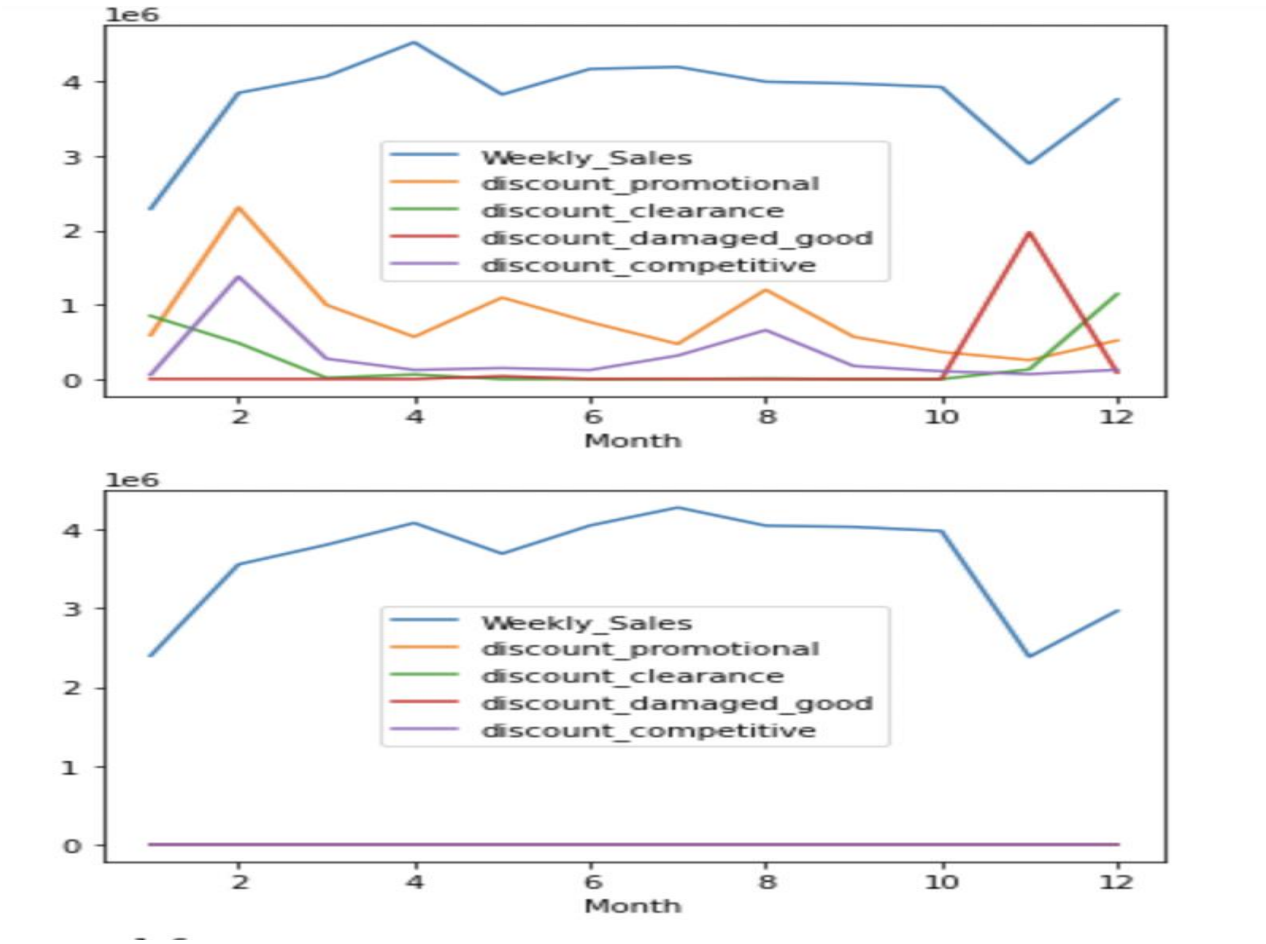

1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month

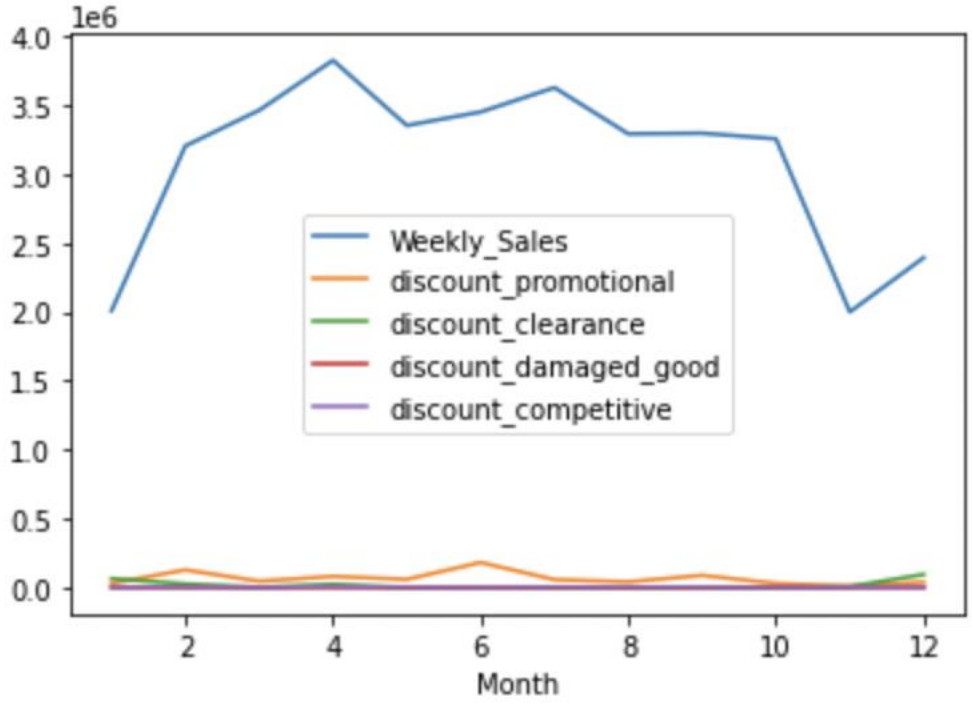

1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month


1e6
Weekly_Sales
discount_promotional
discount_clearance
discount_damaged_good
discount_competitive
Month



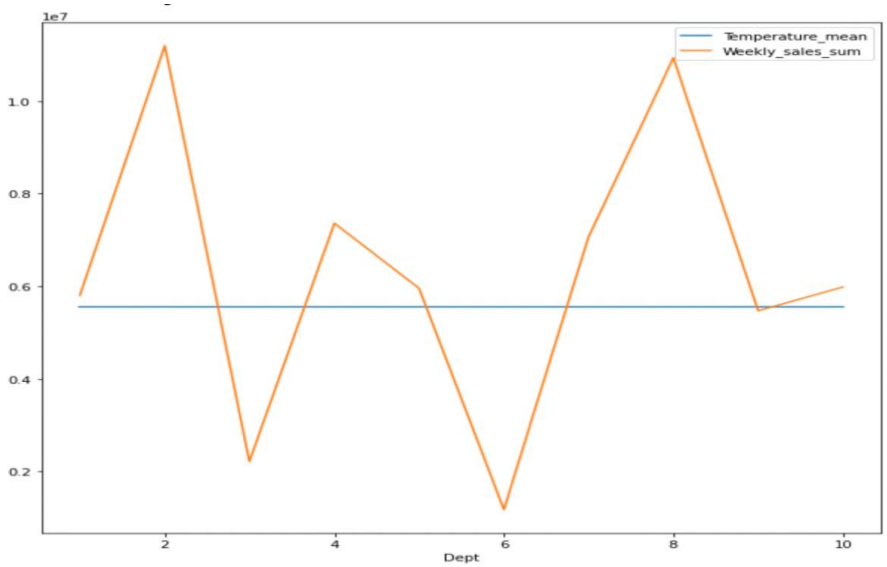
From top 30% of best performing stores we inferred from graphs that discount_promotional help in increased sales this behavior holds good for bottom 30% of least performing stores as well as i t is clearly seen in the above graphs, highest sales is happening only for discount promotional. So can conclude that sales increases with discount promotional kind of discount for both top best 30% and bottom least 30% stores.



E. Identify department which are highly impacted by external factors: “temperature”, “gas price”, and “holiday”. Is there any correlation between overall sales and holiday?

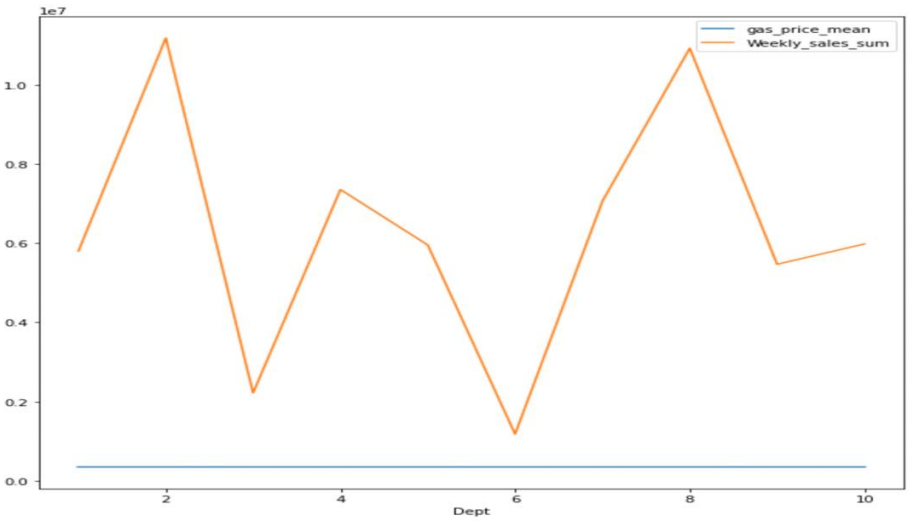
To identify the department which are highly impacted by external factors top 15% of best-performing stores were taken based on sales and then department-wise impact for all these factors were considered. The stores analyzed for temperature and gas price are 20, 4, 14, 13, 2, 10, 27.

Temperature analysis graphs:

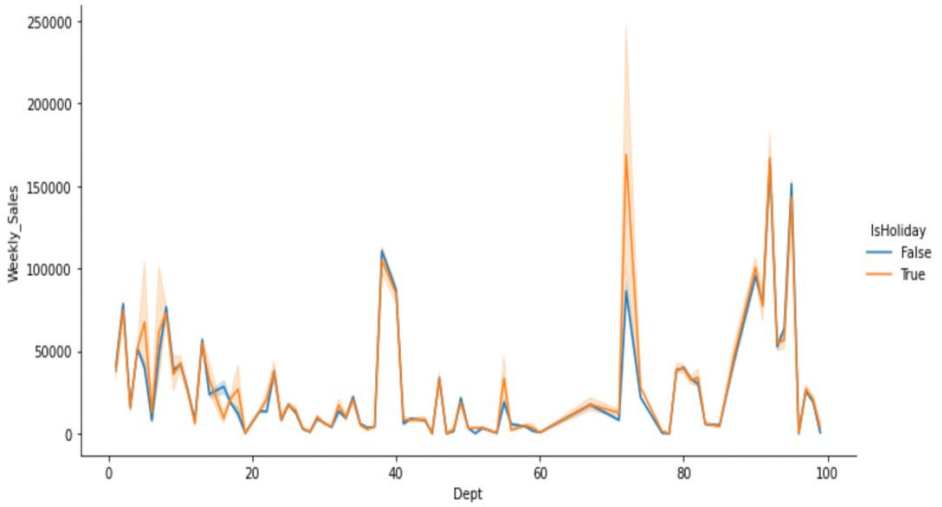


From the above graph we can infer that for any of the department or its sales the temperature has no impact. It does not directly affect or relate to the department or its sales.

For gas_price analysis:

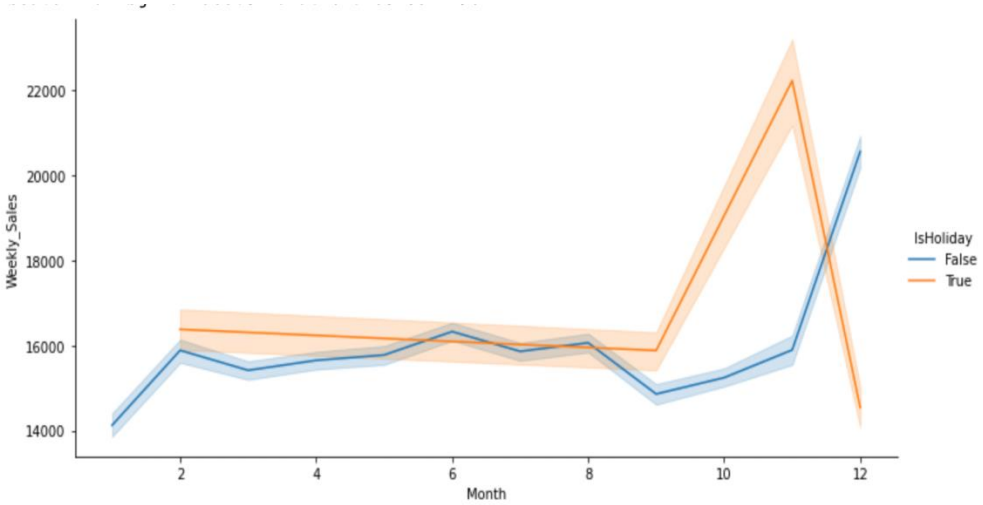


For holiday analysis:



From the above graph we can infer that when there is a holiday all department has better sales.

Is there any correlation between overall sales and holiday?



As you can see from the graph if it is holiday sales are increasing and if there is no holiday the sa les are dropping. So it positively correlated if it is IsHoliday is true and negatively correlated if Is holiday is false.

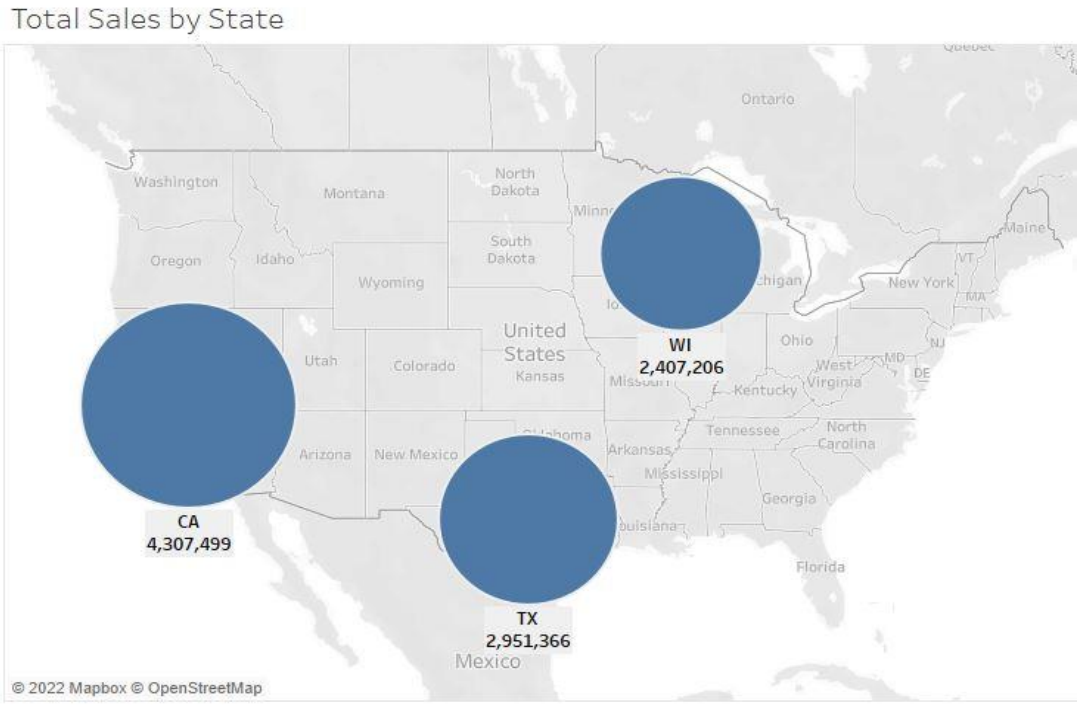
```
[ ] ov=train_store_storeattributes.groupby(by=['IsHoliday'],as_index=False)['Weekly_Sales'].sum()
    os_hol=ov['Weekly_Sales'].corr(train_store_storeattributes['IsHoliday'])
    os_hol

    -0.9999999999999999
```

The above -ve value shows that it is kind of -vely co-related because sales are not increasing for both values of Isholiday.

DATASET 2

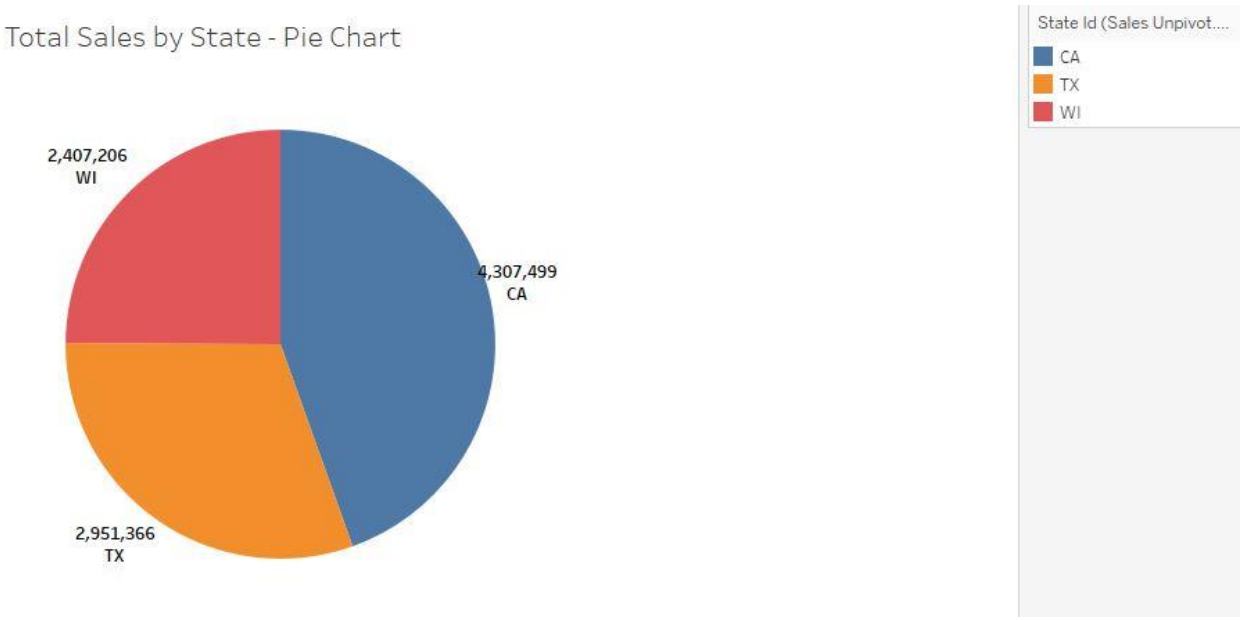
1. Use Tableau to visualize the dataset_02.

Below Image1 represents Total Sales by State:



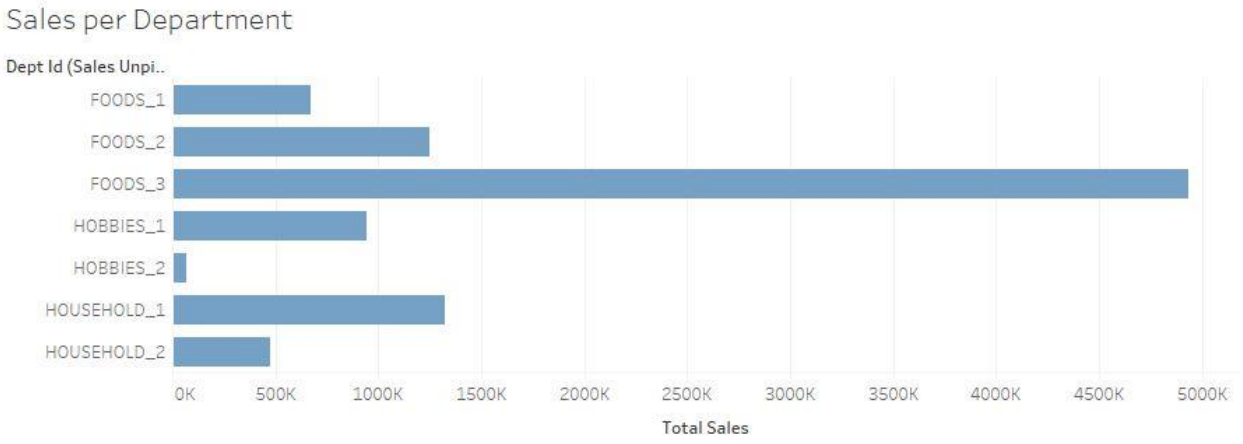
From the above map, we can observe that California has the highest number of sales, with over 4.3M followed by Texas which is 2.9M and Wisconsin which has 2.4M sales.

Below Image 2 represents Total Sales by State-Pie Chart:



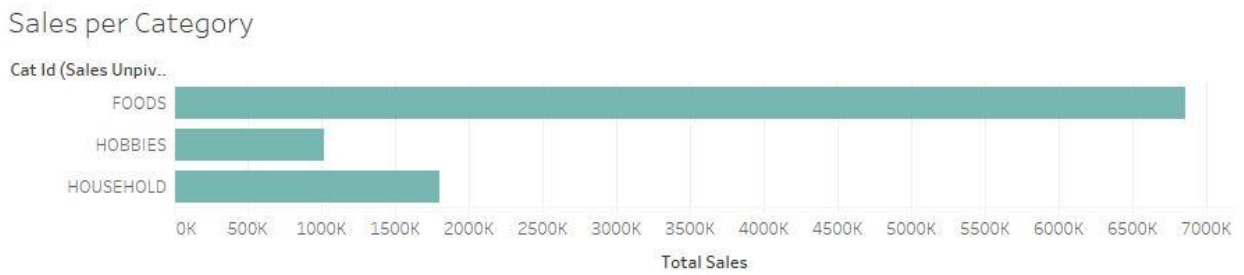
This is the Pie chart depiction of total sales grouped by state id.

Image 3 represents Sales Per Department



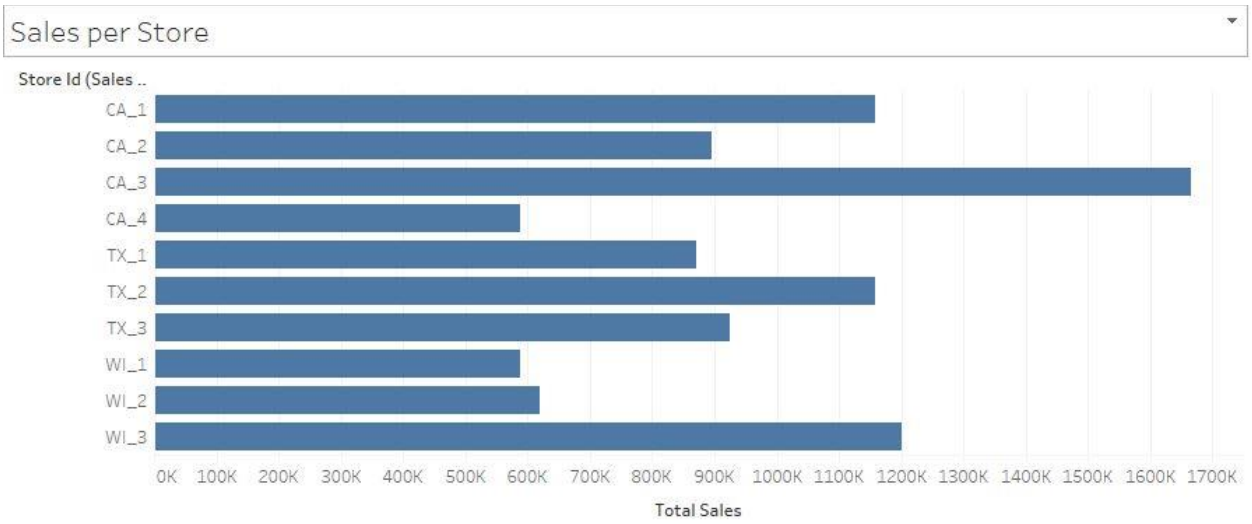
From the above bar graph, we can observe that the department FOODS_3 has the highest number of sales with almost 5000k sales.

Below Image 4 represents Sales Per Category



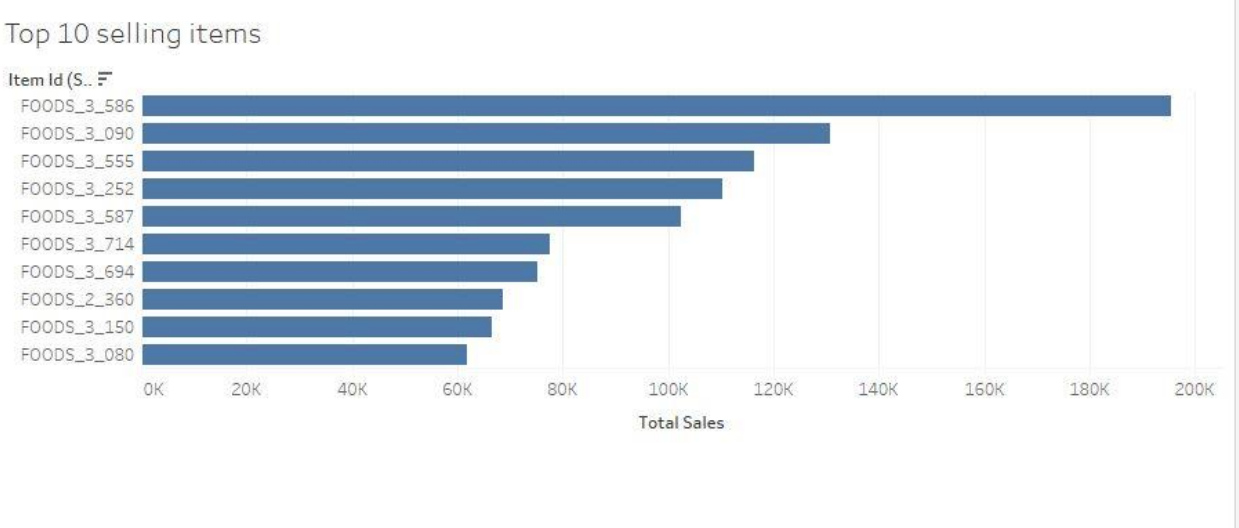
4. From the above bar graph, we can infer that the foods category occupies many of the sales with almost 7M.

Below Image 5 represents Sales Per Store



From the above bar graph, we can observe that the store CA_3 has the highest number of sales with about 1.65M

From the above bar graph, we can observe that the store CA_3 has the highest number of sales with about 1.65M



The above bar graph shows the top 10 selling items in descending order. The item "FOODS_3_586" has been sold the highest number of times at about 190k.

Below Image 7 represents Sales – Time Series

From the above time series graph, we can observe that January month had the highest number of sales at about 890k. Then there was a drastic drop in the month of February with just 720k sales. Although, the sales got picked up from the month of July.

2. **Publish the Tableau dashboard on the public server**

https://10az.online.tableau.com/#/site/tableausiteai/workbooks/1910675/views

SPRINT 2 : Machine Learning Model Building

1. **Design a prediction model to forecast the weekly sales across the first ten stores and use the same model to make predictions for store_11_35. Consider external variables factors and plot-relevant graphs**.

A. Begin with a Linear regression model to forecast the weekly sales using the given features

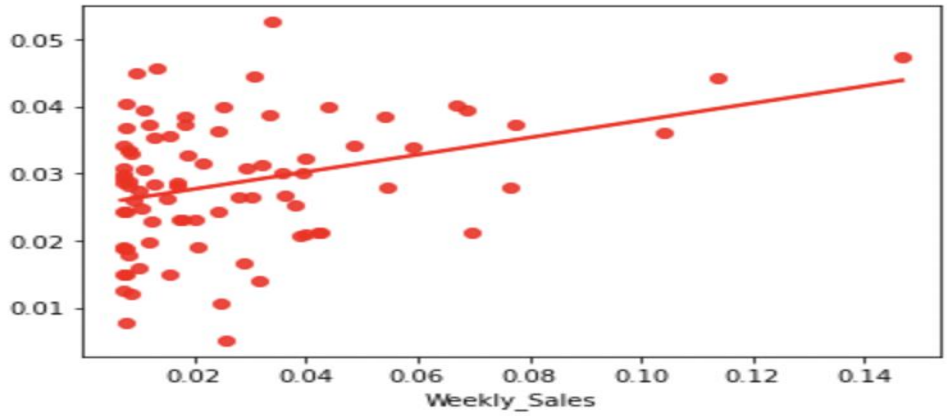
Linear Regression

- Did feature selection and dropped column 'Type' and used feature engineering technique normalization before building the model because when built model without it the performance evaluation metric error rate was very high so used normalization and built model.

- The performance evaluation metric used is:

  The Mean Absolute Error of the model: 0.01831734909788823
  The Mean Squared Error of the model: 0.0005813874775008826

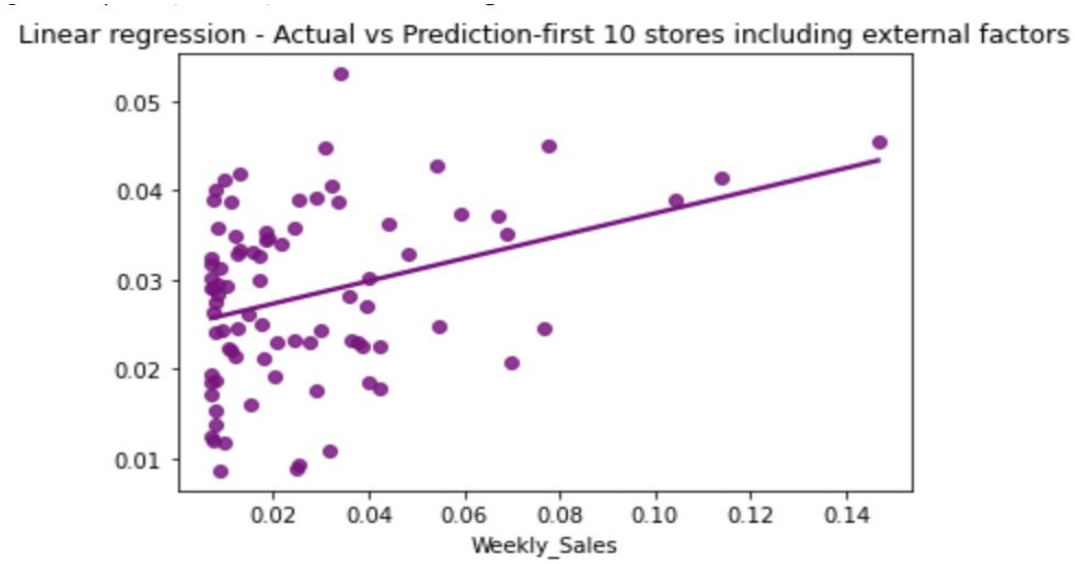
The graph plotted for the predicted and actual value of the regression model is as below:



The less error value and the graph plotted shows that feature selection and feature engineering applied turned in favor as model is predicted at good rate with less error.

Built Linear Regression as below:

Including external factors for first 10 stores



The performance evaluation metric used is:

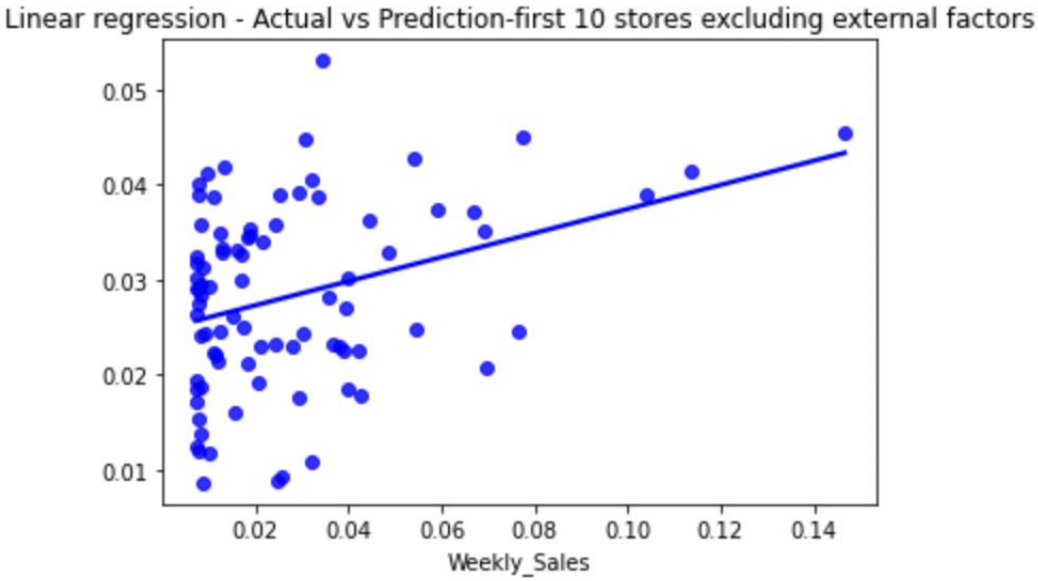
The Mean Absolute Error of model: 0.01855187754044593
The Mean Squared Error of model:0.0005848934202279076
Score of the model: 10.797293492534887

Store 11-35:

Used the built model of Store 1-10 to predict store 11-35

The Mean Absolute Error of model: 16806.548378199703
The Mean Squared Error of model: 870403893.024495

Excluding external factors for first 10 stores



The performance evaluation metric used is:

The Mean Absolute Error of the model:12947.484232170704
The Mean Squared Error of model: 287677820.7140935
score of the model: 9.970060703050343

Store 11-35:

Used the built model of Store 1-10 to predict store 11-35

The Mean Absolute Error of model: 750952802.3878015
The Mean Squared Error of model: 6.664603315736966e+17

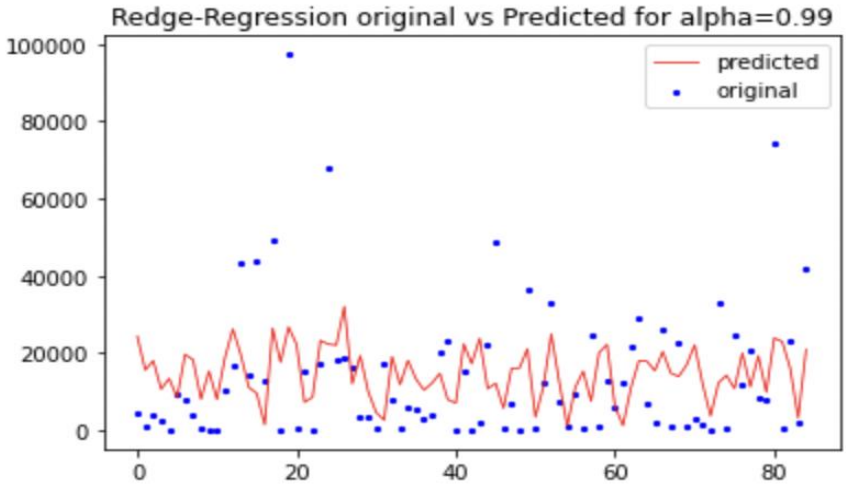
b. Create the following machine learning models: Ridge Regression, Boosting and ARIMA to predict sales.

Ridge Regression(with external factors)

Store 1-10:

- To consider the effect of external factors for first 10 stores with ridge regression considered two different alpha values to check on model error rate. Performed normalization to data before building model because without that got more error rate.

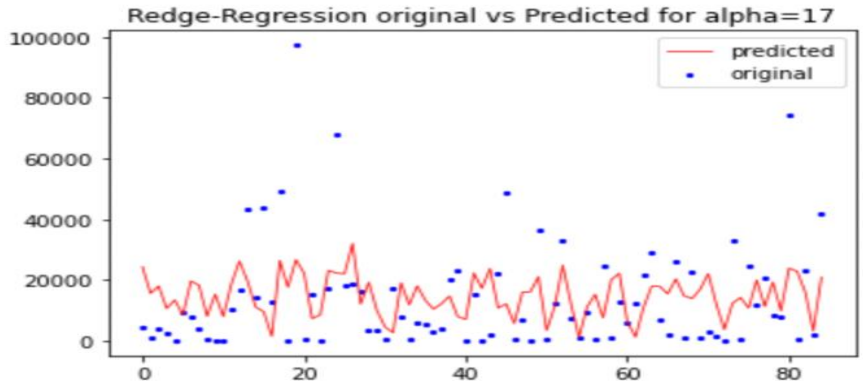
- With alpha=0.99



Here the performance evaluation metric used is:

The Mean Absolute Error of model: 12950.681646965293
The Mean Squared Error of model: 285034767.05530965
The Score of the model: 10.797215051857023

- With alpha=17



Here the performance evaluation metric used is:

The Mean Absolute Error of model: 12948.260201271554
The Mean Squared Error of model: 285033850.9289698
The Score of the model: 10.79750175730606

You can see as alpha value increased the error rate rate decreased for the model.

Store 11-35:

Used the built model of Store 1-10 to predict store 11-35

- With alpha=0.99

  Here the performance evaluation metric used is:

  The Mean Absolute Error of model: 538816648.6142223
  The Mean Squared Error of model: 3.46853855204516e+17

- With alpha=17

  Here the performance evaluation metric used is:

  The Mean Absolute Error of model: 531161862.4061004
  The Mean Squared Error of model: 3.363647237238198e+17

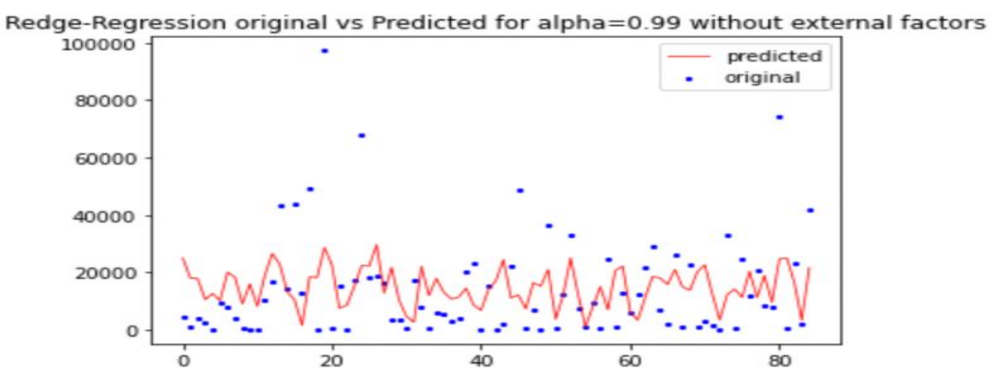
Even here as alpha value increased the error rate decreased for the model.

Ridge Regression(without external factors)

Store 1-10:

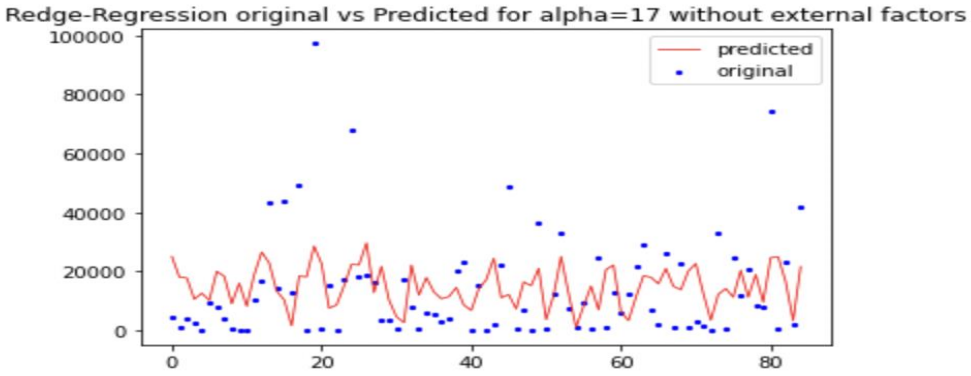
To consider the effect of not including external factors for first 10 stores with ridge regression considered two different alpha values to check on model error rate. Performed normalization to data before building model because without that got more error rate.

- With alpha=0.99



Here the performance evaluation metric used is:
The Mean Absolute Error of model: 12947.425132618266
The Mean Squared Error of model: 287677944.4260363
Score of the model: 9.970021986896727

- With alpha=17



Here the performance evaluation metric used is:
The Mean Absolute Error of the model: 12946.41725218746
The Mean Squared Error of model: 287674303.74608207
A score of the model: 9.971161352435798

You can see as the alpha value increased the error rate decreased for the model even if the external factors are not included.

Store 11-35:

Used the built model of Stores 1-10 to predict stores 11-35

- With alpha=0.99

  Here the performance evaluation metric used is:
  The Mean Absolute Error of model: 749554282.2603292
  The Mean Squared Error of the model: 6.639773187940716e+17

- With alpha=17

  Here the performance evaluation metric used is:
  The Mean Absolute Error of model: 726101146.5481724
  The Mean Squared Error of model: 6.22996448984605e+17

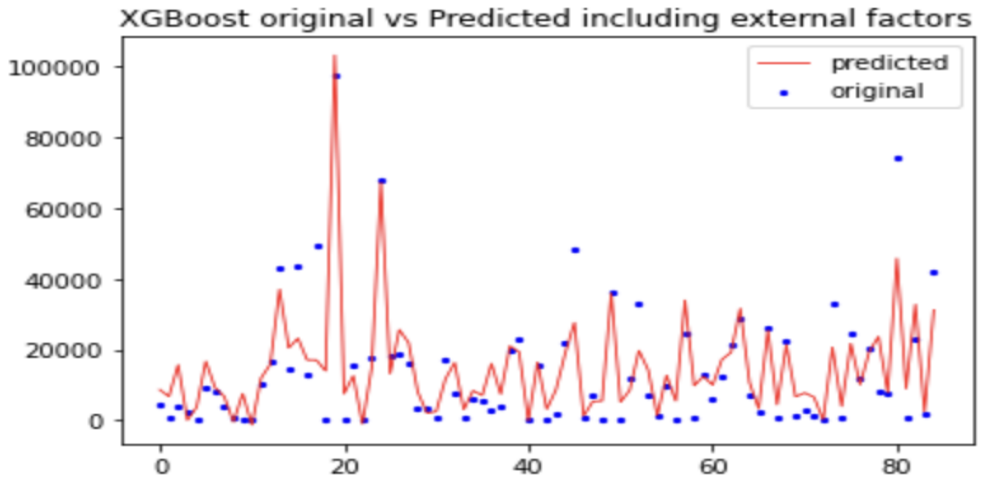
Even here as alpha value increased the error rate decreased for the model when external factors are not included.

BOOSTING ALGORITHM

XGboost(With external factors)

- To consider the effect of external factors for first 10 stores considered XGboost of boosting algorithm. Performed normalization to data before building model because without that got more error rate.
- Performance evaluation metric used is:
  The Mean Absolute Error of model: 5366.568656285903
  The Mean Squared Error of model: 62624069.31912708
  Score of the model: 80.40154383353625

Graph plotted for actual vs predicted sales of XGboost :

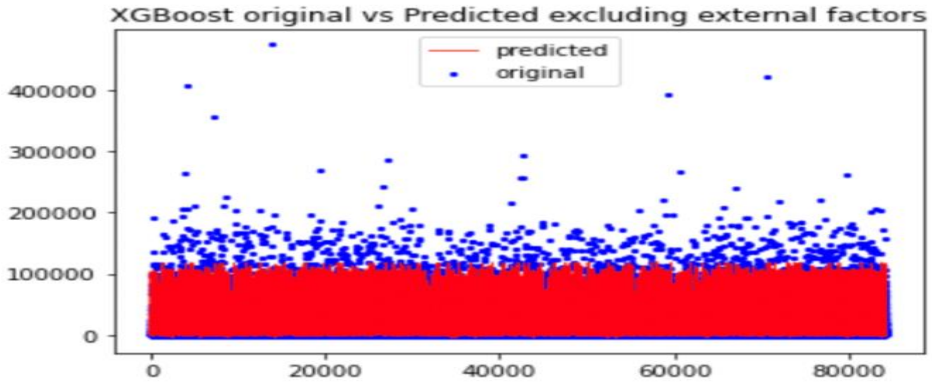


Here the normalization did prior to model building helped to get better score with less error rate as can be seen in performance evaluation used. The graph also shows the predicted values are very accurate with the actual one's so can say model built is the better one.

XGboost(Without external factors)

- To consider the effect of not including external factors for first 10 stores considered XGboost of boosting algorithm.
- Performance evaluation metric used is:
  The Mean Absolute Error of model: 6968.396529056831
  The Mean Squared Error of model: 133086926.60173918
  The Score of the model: 73.90463013091836

  Graph plotted for actual vs predicted sales of XGboost :



Here it is clear from performance metrics and graph that including external factors give better score with less error for the XGboost model to predict sales.

Store 11-35:

Used the built model of Store 1-10 to predict store 11-35:

- With external factors:
  Performance evaluation metric used is:
  The Mean Absolute Error of model: 14637.136522168374
  The Mean Squared Error of model: 713584790.303554

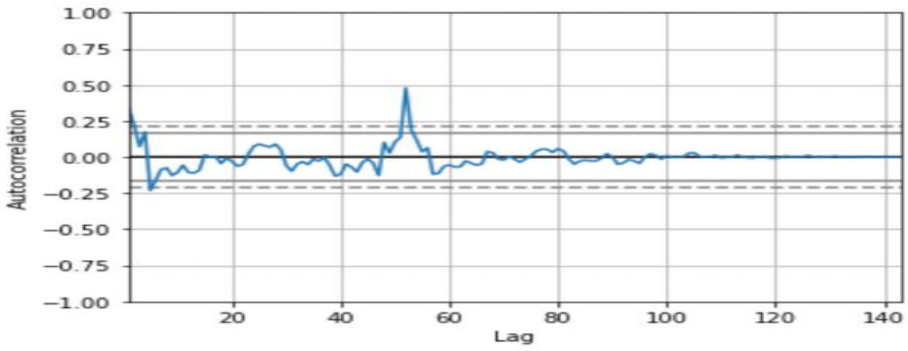
- Without external factors
  Performance evaluation metric used is:
  The Mean Absolute Error of model: 7907.349560543148
  The Mean Squared Error of model: 176413525.03579786
  Score of the model: 70.33766996430388

Here for stores 11-35 when external columns are not included error rate is less.

ARIMA

Stores 1-10

- Here to predict the weekly sales grouped the date and took the weekly sales sum as part of the feature engineering.
- As ARIMA is time-series forecasting extracted date values and checked whether data is stationary or not.
- To check whether data is stationary or not used time series graph and dicky-fuller test.
- When plotted for initial data-frame graph and d-f test results are as below:

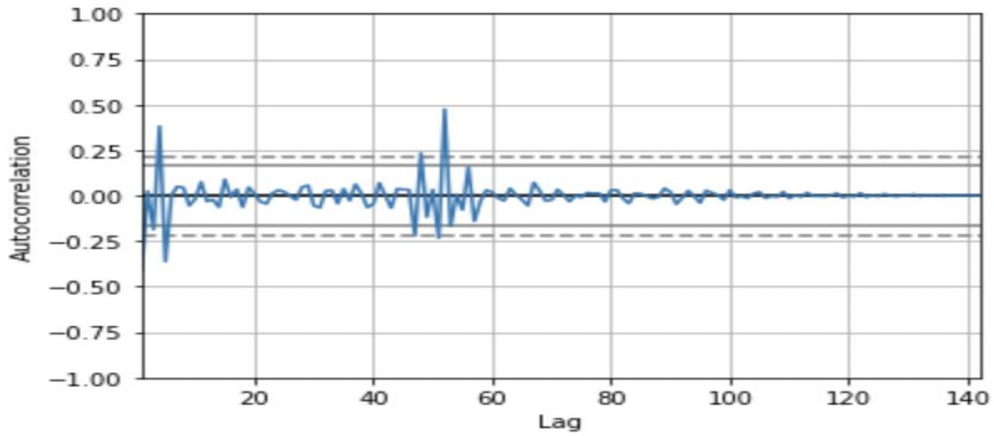



```
<Figure size 720x720 with 0 Axes>
```

```
Test Statistic             -5.908298e+00
p-value                     2.675979e-07
No. of lags used            4.000000e+00
No. of observations used    1.380000e+02
Critical value (1%)        -3.478648e+00
Critical value (5%)        -2.882722e+00
Critical value (10%)       -2.578065e+00
dtype: float64
```

From above graph we can infer that the data is not stationary as it is not linear and d-f test shows test statistic is less than the critical values and mean is increasing with time. So, it's not st ationary data.

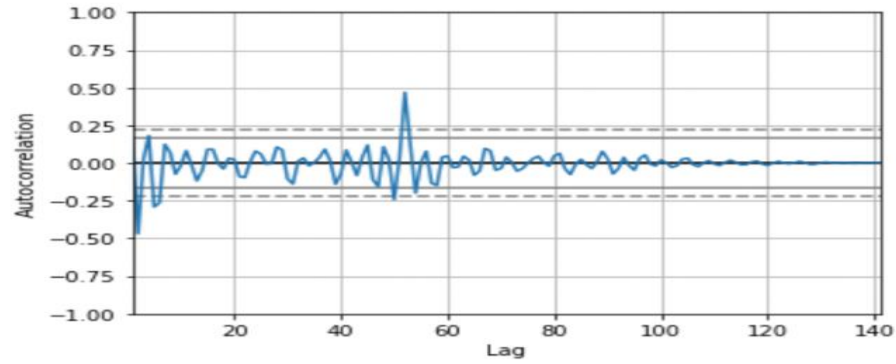
As data was not stationary took diff(1) of data and plotted timeseries and dicky-fuller test got as below:




```
<Figure size 720x720 with 0 Axes>
```

```
Test Statistic              -6.699469e+00
p-value                      3.922579e-09
No. of lags used             7.000000e+00
No. of observations used     1.340000e+02
Critical value (1%)         -3.480119e+00
Critical value (5%)         -2.883362e+00
Critical value (10%)        -2.578407e+00
dtype: float64
```

It is still not stationary so took diff(2) and tested as below:



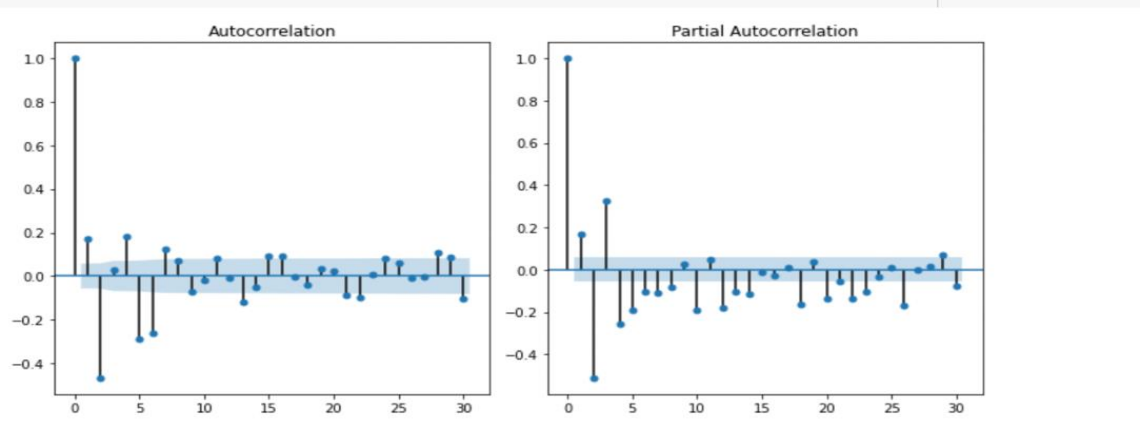

```
<Figure size 720x720 with 0 Axes>
```

```
Test Statistic              -7.276766e+00
p-value                      1.537185e-10
No. of lags used             4.000000e+00
No. of observations used     1.360000e+02
Critical value (1%)         -3.479372e+00
Critical value (5%)         -2.883037e+00
Critical value (10%)        -2.578234e+00
dtype: float64
```

Now the graphs shows linear behaviour and now can be concluded that data is stationary

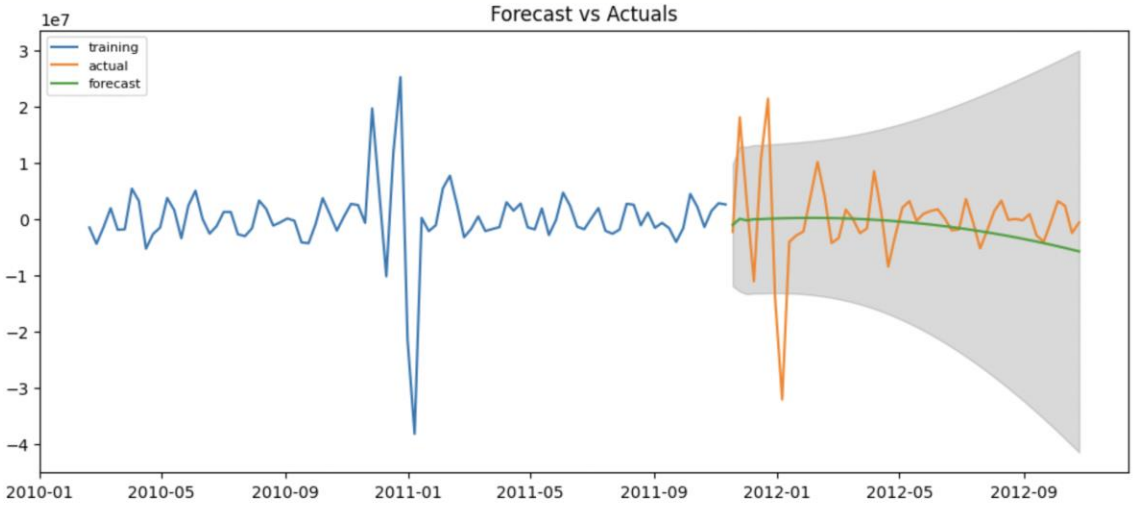
As data is now stationary with difference of **2 the d value is 2**

Found p,q values using ACF and PACF



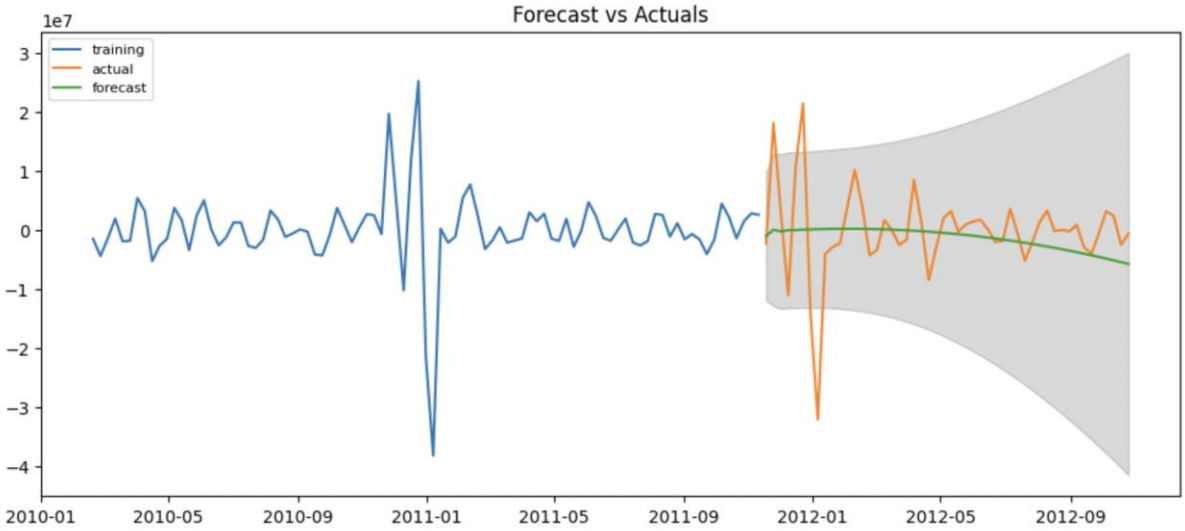
From above partial autocorrelation graph, we can see that two lags are significantly out of limit b ut second one is not that far when compared to one so the order of p can be taken as 1.From abov e autocorelation graph there are two lags which are significant out of limit so the order of q can b e taken as 2. Later Built the model a predicted it.

For the first 10 stores prediction:



The mean absolute error of the model = 5057731.956916306
The mean absolute error of the model = 59566858694429.85
The mean absolute percentage error of the model= 3.9233489986639842
The score of the model is= 92.32999999999998

For the first 11-35 stores prediction :



The performance Metric Evaluation metric used is:

The mean absolute error of the model = 5057731.956916306
The mean absolute error of the model = 59566858694429.85
The mean absolute percentage error of the model= 3.9233489986639842
The score of the model is= 92.32999999999998

C. Communicate the model performance metrics and tabulate the comparison in the report. Support your finding by validating the model accuracy across various stores (first 10 stores and store_11_35).

With External factors:

	First 10 stores (combined data)	**store_11_35**
Ridge Regression alpha = 0.9	Mean Absolute Error: 12950.681646965293 Mean Squared Error: 285034767.05530965	Mean Absolute Error : 538816648.6142223 Mean Squared Error: 3.46853855204516e+17

	Score of the model: 10.797215051857023	
Ridge Regression alpha = 17	Mean Absolute Error: 12948.260201271554 Mean Squared Error: 285033850.9289698 Score of the model: 10.79750175730606	Mean Absolute Error: 531161862.4061004 Mean Squared Error: 3.363647237238198e+17
XGBOOST	The Mean Absolute Error of model: 5366.568656285903 The Mean Squared Error of model: 62624069.31912708 The Score of the model: 80.40154383353625	Mean Absolute Error: 14637.136522168374 Mean Squared Error: 713584790.303554
Linear Regression	The Mean Absolute Error of model: 0.01855187754044593 The Mean Squared Error of model:0.0005848934202279076 Score of the model: 10.797293492534887	The Mean Absolute Error of model: 16806.548378199703 The Mean Squared Error of model: 870403893.024495

Without External factors:

	First 10 stores (combined data)	**store_11_35**
Ridge Regression alpha = 0.9	Mean Absolute Error: 12947.425132618266 Mean Squared Error: 287677944.4260363 Score of the model: 9.970021986896727	Mean Absolute Error: 749554282.2603292 Mean Squared Error: 6.639773187940716e+17
Ridge Regression alpha = 17	Mean Absolute Error: 12946.41725218746 Mean Squared Error: 287674303.74608207	Mean Absolute Error : 726101146.5481724 Mean Squared Error of: 6.22996448984605e+17

	Score of the model: 9.971161352435798	
XGBOOST	The Mean Absolute Error: 6968.396529056831 The Mean Squared Error of model: 133086926.60173918 Score of the model: 73.90463013091836	Mean Absolute Error: 7907.349560543148 Mean Squared Error: 176413525.03579786 Score of the model: 70.33766996430388
Linear Regression	The Mean Absolute Error of the model:12947.484232170704 The Mean Squared Error of model: 287677820.7140935 Score of the model: 9.970060703050343	The Mean Absolute Error of model: 750952802.3878015 The Mean Squared Error of model: 6.664603315736966e+17

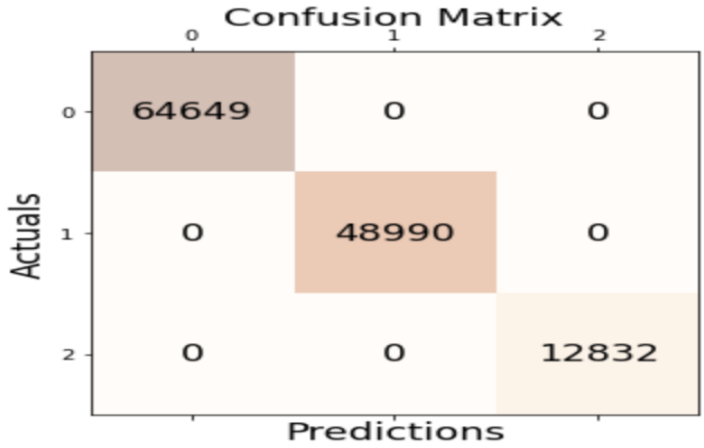
ARIMA

Arima	**First 10 stores (combined data)**	**store_11_35**
Arima	The mean absolute error of the model = 5057731.956916306 The mean absolute error of the model = 59566858694429.85 The mean absolute percentage error of the model= 3.9233489986639842 The score of the model is= 92.32999999999998	The mean absolute error of the model = 5057731.956916306 The mean absolute error of the model = 59566858694429.85 The mean absolute percentage error of the model= 3.9233489986639842 The score of the model is= 92.32999999999998

2. Consider only the first 10 stores for this problem statement.

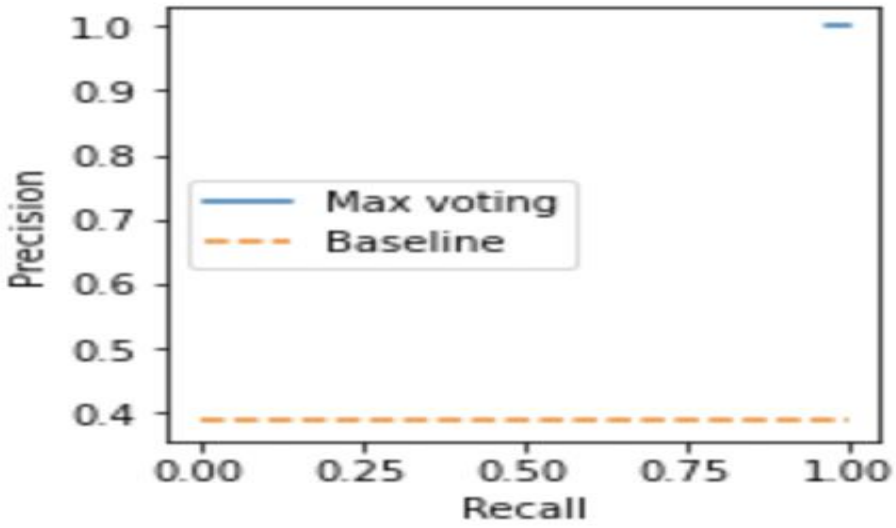
a. Consider the problem statement as multi-label classification problem. Use the below classification algorithms and perform hyper-parameter tuning for the Deep Learning models.

- **Ensemble model**:
  - Here used the max_voting ensemble method, extended ensemble method -bagging classifier, random forest classifier, and XGB classifier to build and predict the model.
  - Used max_voting, extended ensemble method -bagging classifier, random forest classifier and XGB classifier to build and predict the model.
  - Relevant graphs plotted:
    Confusion Matrix, F1 Score, and other metrics

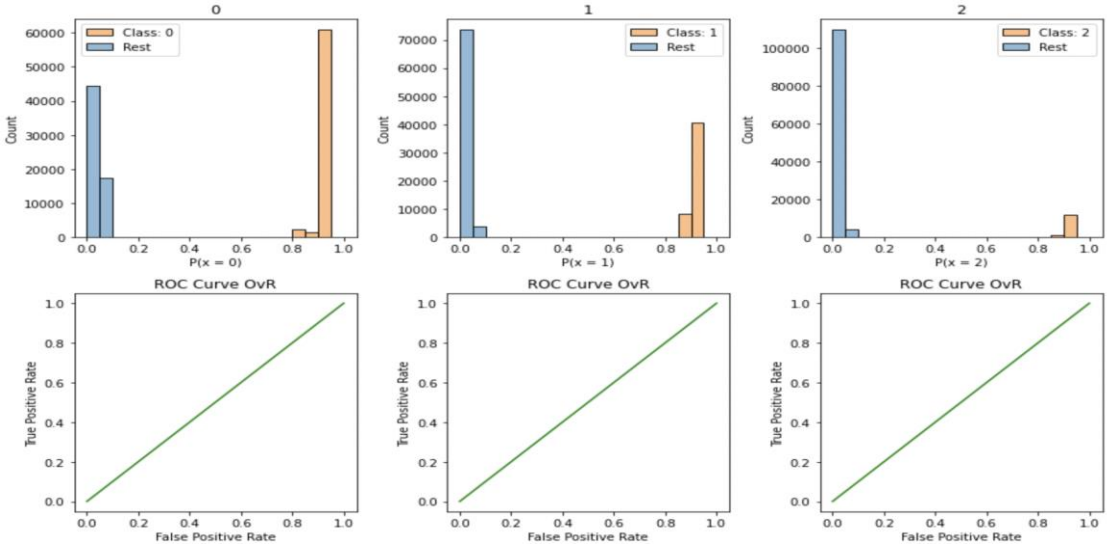



```
Precision of type A 0.9699920478926916
Precision of type B 1.0
Precision of type C 1.0
Prescison of the model 0.9899973492975639
Recall of type A 1.0
Recall of type B 1.0
Recall of type C 1.0
Recision of the model 1.0
The F1 score of the model 0.494949494949495
The F1 Score using f1_score 1.0
The accuracy of the model 1.0
```

Precision recall curve:



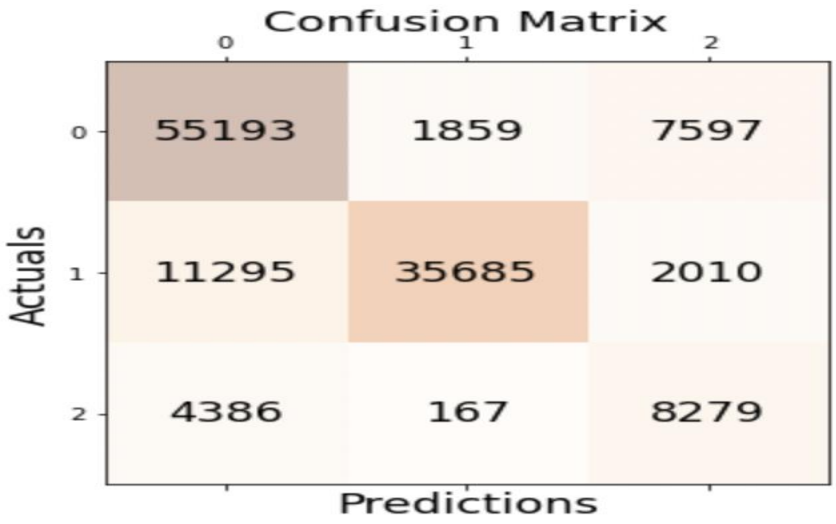
ROC CURVE :



- **Recurrent Neural Network:**

  - RNN model is built with various keras layers to predict the output.
  - Relevant Graphs:

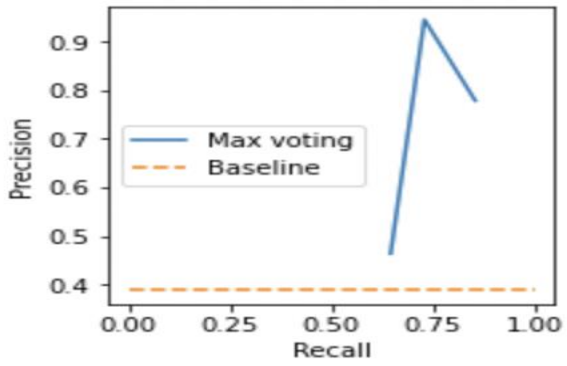
    Confusion matrix:



Precision, Recall, Accuracy, F1 score:

```
Precision of type A 0.8537332363996349
Precision of type A 0.728413962033068
Precision of type A 0.645183915211 97
Precision of model 0.7424437045482243
Recall of type A 0.7787482010328188
Recall of type B 0.9462756224974146
Recall of type c 0.46287599239628757
Recall of model 0.7292999386421738
Accuracy of Type A 0.8012429726972982
Accuracy of Type B 0.8787785342094235
Accuracy of Type C 0.888037573831155
Accuracy of model 0.8560196935792922
The F1 score of the model 0.7354425560842963
The F1 Score using f1_score 0.7840295403689383
```

Precision Recall Curve :



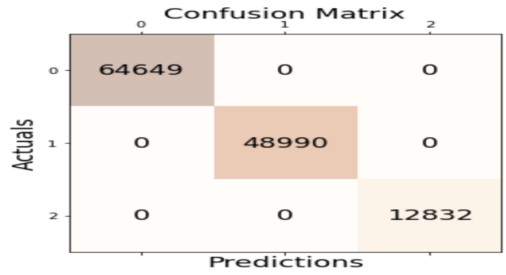
- **Convolutional Neural Network**

    - Built CNN model with various neural layers for the multi-classification problem.
    - Evaluation performed on the model is as below:

```
276/276 [==============================] - 0s 1ms/step
>0.896
276/276 [==============================] - 0s 1ms/step
>0.948
276/276 [==============================] - 0s 1ms/step
>0.854
276/276 [==============================] - 0s 1ms/step
>0.913
276/276 [==============================] - 0s 1ms/step
>0.879
276/276 [==============================] - 0s 1ms/step
>0.951
276/276 [==============================] - 0s 1ms/step
>0.931
276/276 [==============================] - 0s 1ms/step
>0.899
276/276 [==============================] - 0s 1ms/step
>0.918
276/276 [==============================] - 0s 1ms/step
>0.927
276/276 [==============================] - 0s 1ms/step
>0.964
276/276 [==============================] - 0s 1ms/step
>0.931
276/276 [==============================] - 0s 1ms/step
>0.937
276/276 [==============================] - 0s 1ms/step
>0.853
276/276 [==============================] - 0s 1ms/step
>0.957
276/276 [==============================] - 0s 1ms/step
>0.882
276/276 [==============================] - 0s 1ms/step
```

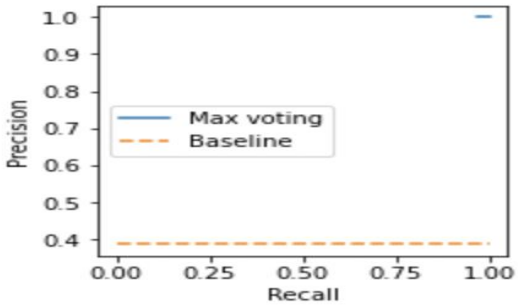
B. Plot the relevant graphs and tabulate the performance metrics (ROC, AUC, Precision-Recall, confusion matrix, F1 score).

Confusion Matrix:

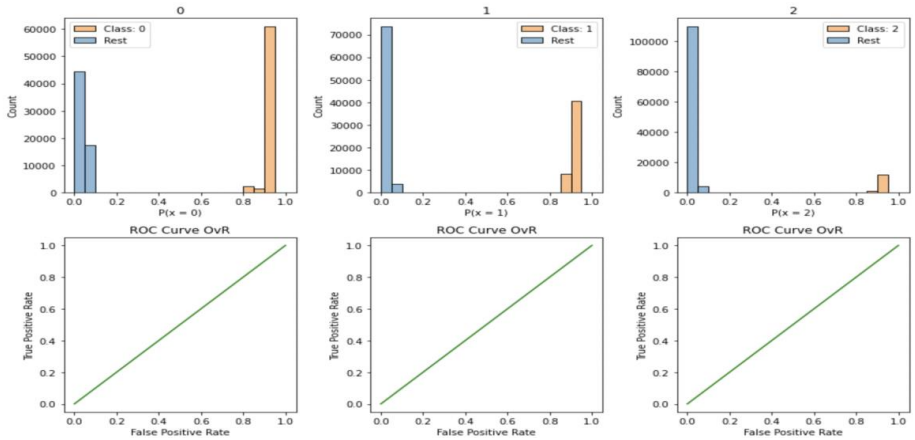



```
Precision of type A 0.9699920478926916
Precision of type B 1.0
Precision of type C 1.0
Prescison of the model 0.9899973492975639
Recall of type A 1.0
Recall of type B 1.0
Recall of type C 1.0
Recision of the model 1.0
The F1 score of the model 0.494949494949495
The F1 Score using f1_score 1.0
The accuracy of the model 1.0
```

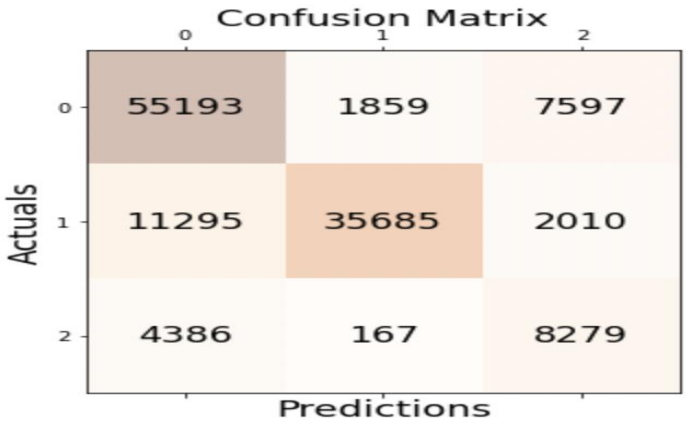
Precision Recall :



ROC Curve:

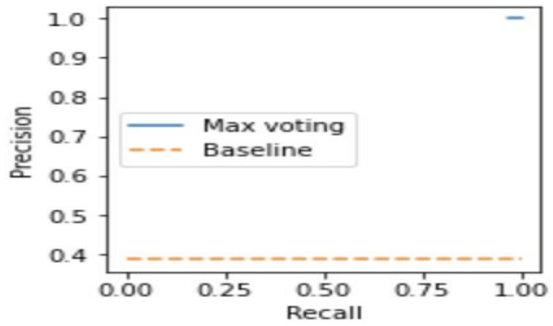


Confusion Matrix :




```
Precision of type A 0.8537332363996349
Precision of type A 0.728413962033068
Precision of type A 0.64518391521197
Precision of model 0.7424437045482243
Recall of type A 0.7787482010328188
Recall of type B 0.9462756224974146
Recall of type c 0.46287599239628757
Recall of model 0.7292999386421738
Accuracy of Type A 0.8012429726972982
Accuracy of Type B 0.8787785342094235
Accuracy of Type C 0.888037573831155
Accuracy of model 0.8560196935792922
The F1 score of the model 0.7354425560842963
The F1 Score using f1_score 0.7840295403689383
```

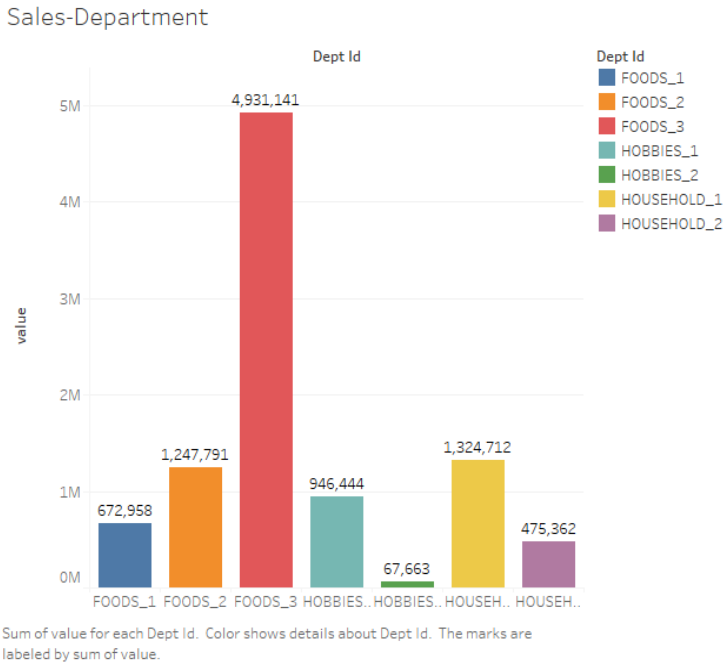
Precision-Recall:



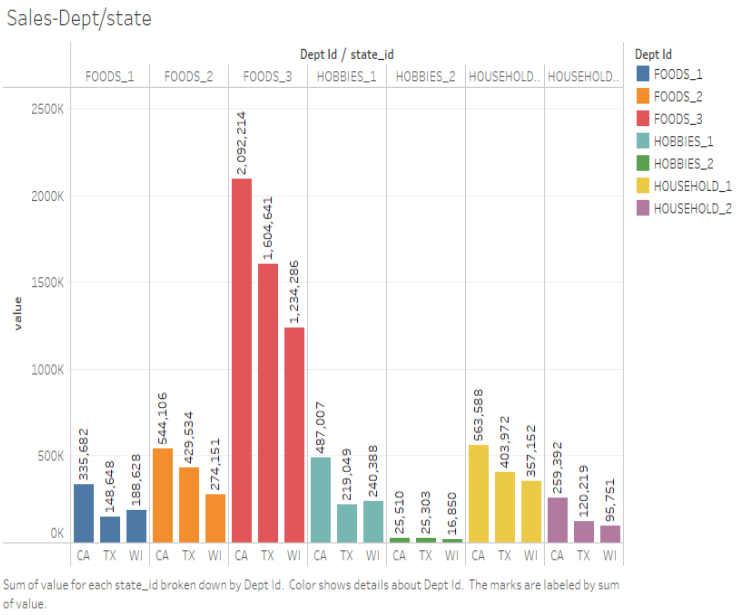
DATASET 2:

Design a machine learning model to make accurate predictions for product sales for next 10 days in advance and compare the performance of different machine learning algorithms.

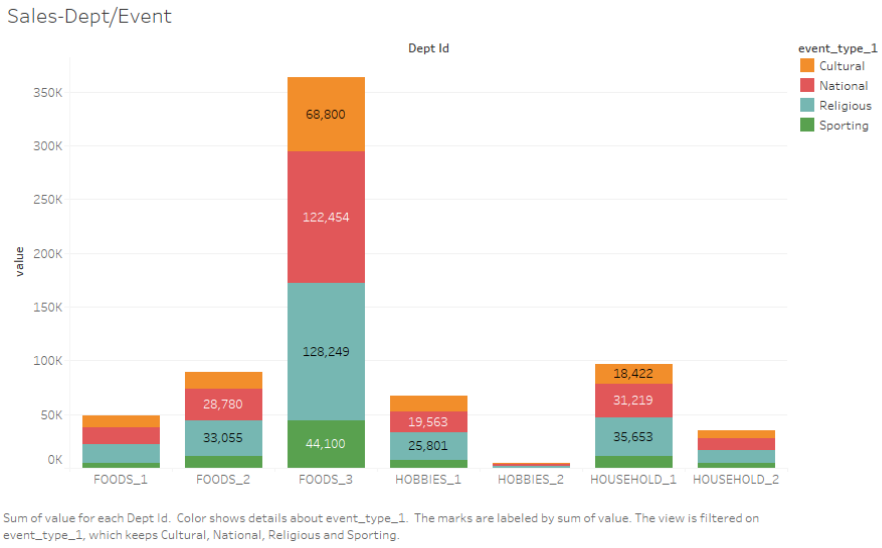
A. Perform data preprocessing and exploratory data analysis.



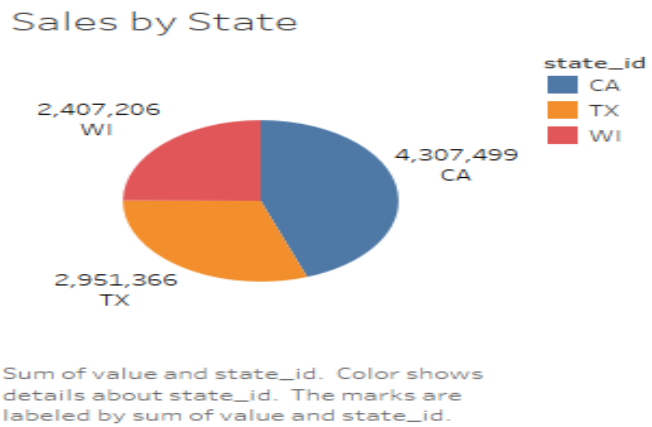
The above graph shows the sales details of different departments where the highest is foods_3 id, which is about 5M.



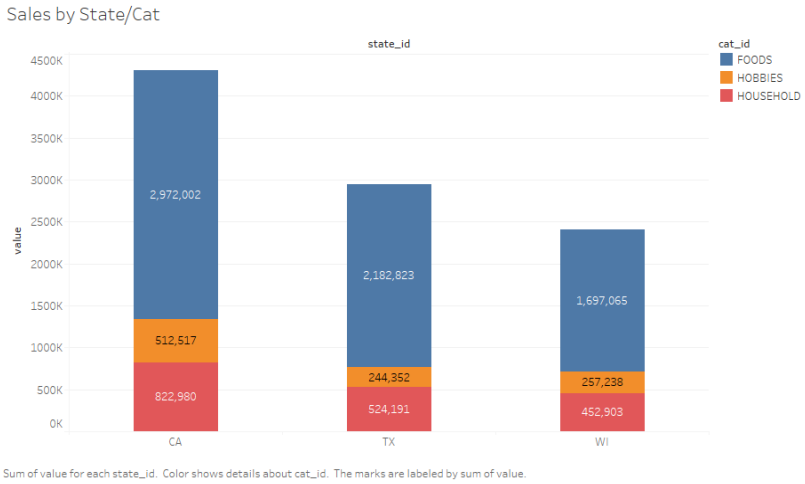
The above graph shows the sales details of different departments state wise where the highest is foods_3 id in the state of California which is about 2M followed by Texas and Wisconsin which are 1.6M and 1.2M respectively.



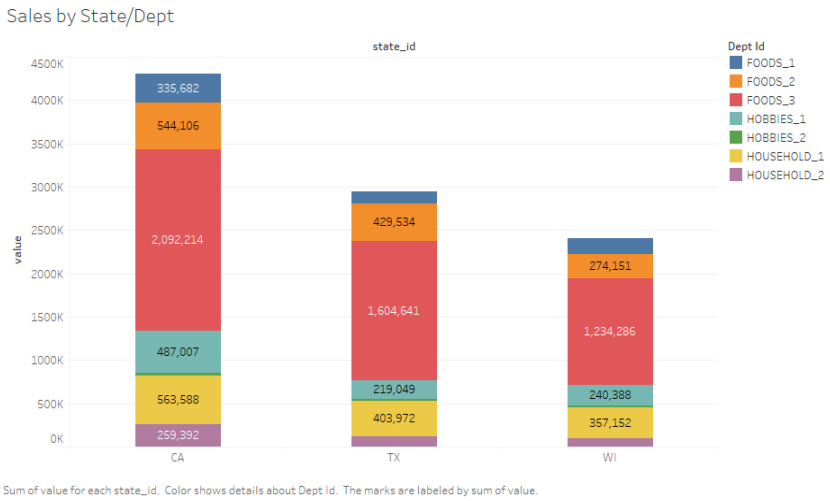
The above graph shows the sales details of different departments based on events like cultural, national, religious and sporting. Where the highest is foods_3 id, which is about 350k in which again the highest is category is religious which is 128k followed by national, which is 122k.



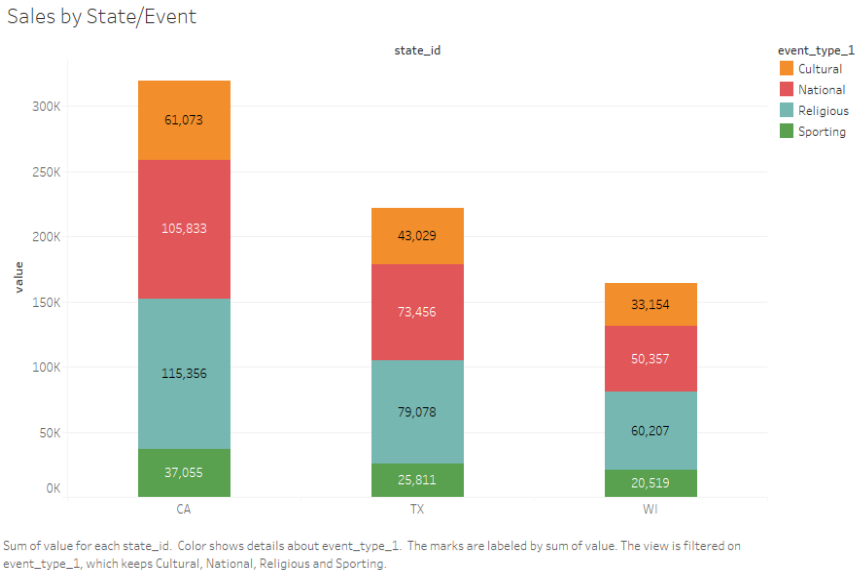
The above graph shows the sales details of different states like California, Texas and Wisconsin where the highest is sales occurred in California, which is about 4.3M followed by Texas with 2.9M and Wisconsin with 2.4M.



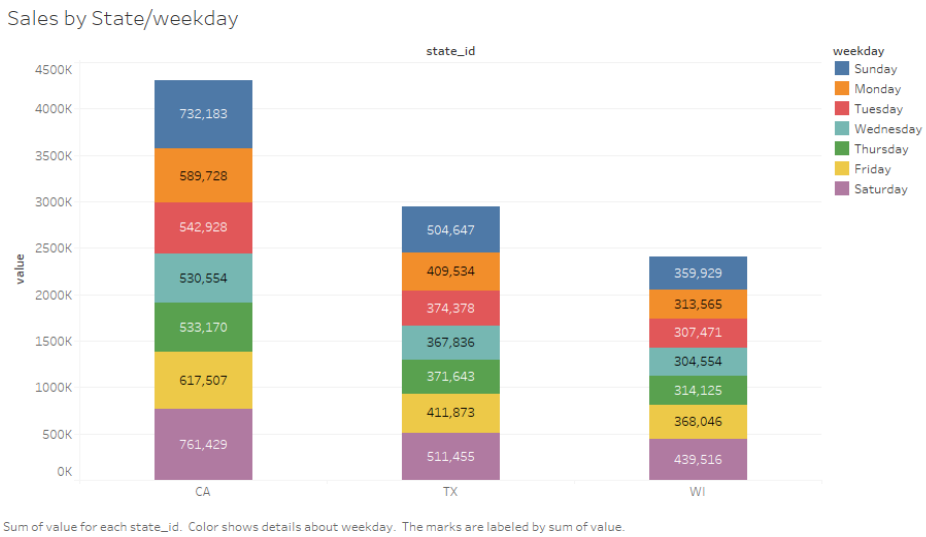
The above graph shows the sales details of different departments like foods, hobbies and household in terms of states where the highest is California with 4500k, followed by Texas with 3000k and Wisconsin 2500k. Sales of foods has the highest value in all the states.



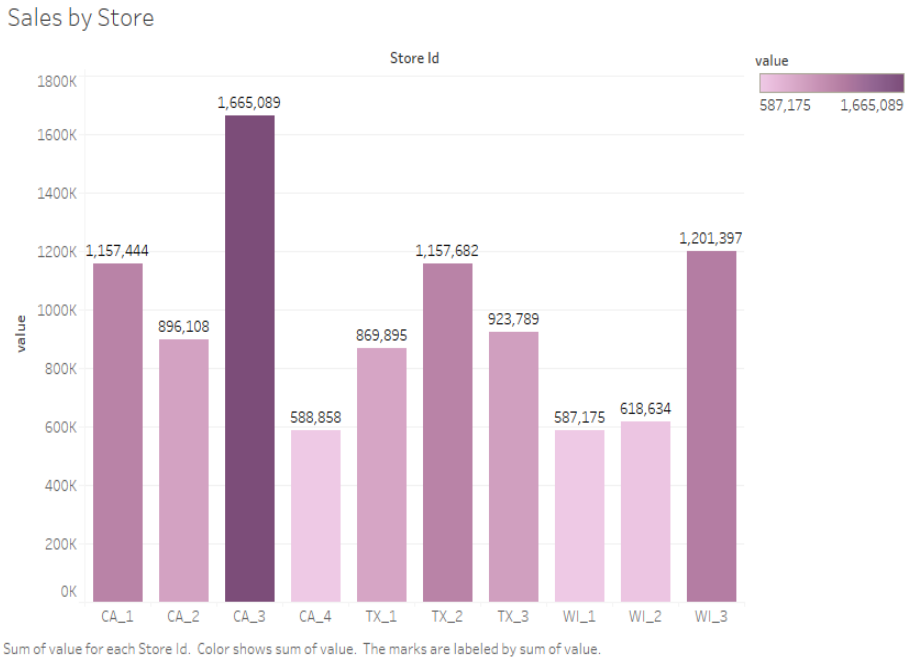
The above graph shows the sales details of different states where the highest sales occurred in California which is about 4000k, followed by Texas and Wisconsin.



The above graph shows the sales details of different states based on events where the highest is California based on events like cultural, national, religious and sporting. Where the highest is California , which is about 300k in religious category and 108k followed in national. Followed by Texas and Wisconsin.

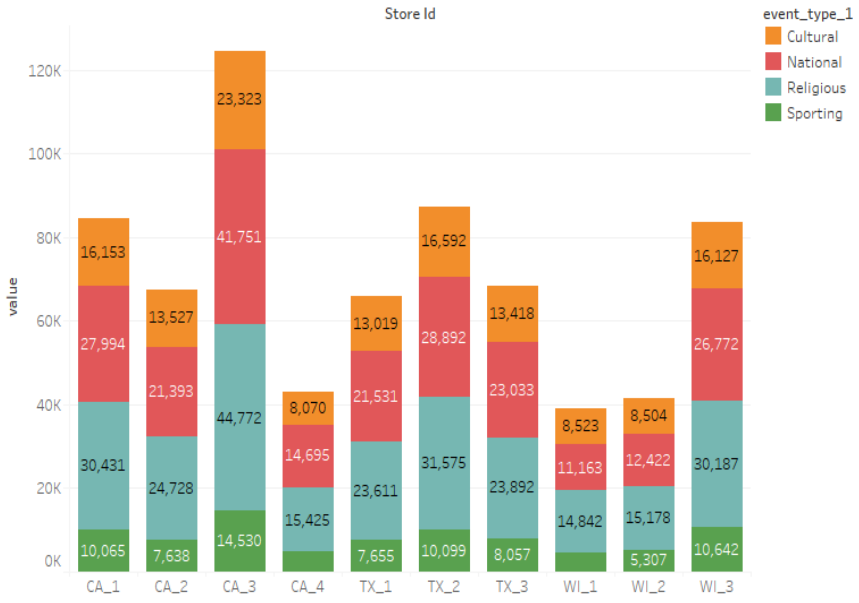


The above graph represents sales in different states and based weekly data. Where the highest is California, which is about 4000k in which again the highest is sales occurred on Saturday. Followed by Texas and Wisconsin.



The above graph represents sales in different states and stores in it. Where the highest is California, which is about 1600k in which again the highest is sales occurred in CA_3 id. Followed by Wisconsin and Texas.

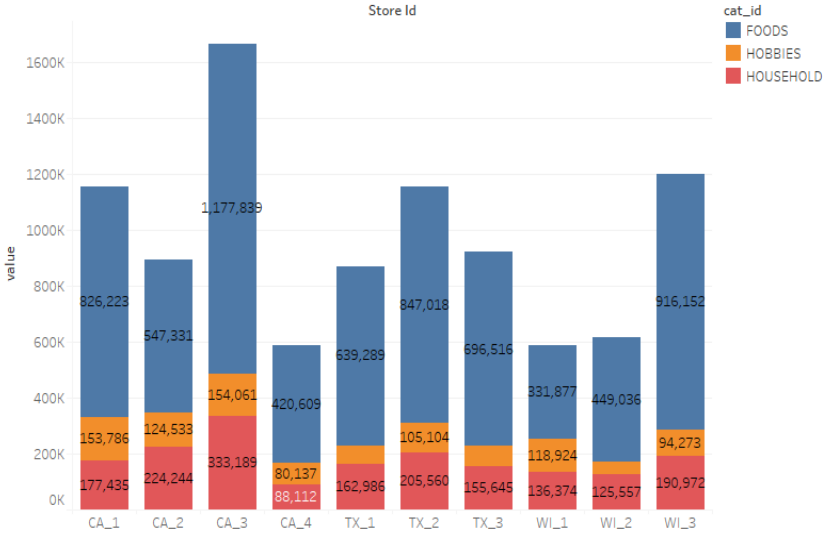
Sales by Store/Event



Sum of value for each Store Id. Color shows details about event_type_1. The marks are labeled by sum of value. The view is filtered on event_type_1, which keeps Cultural, National, Religious and Sporting.

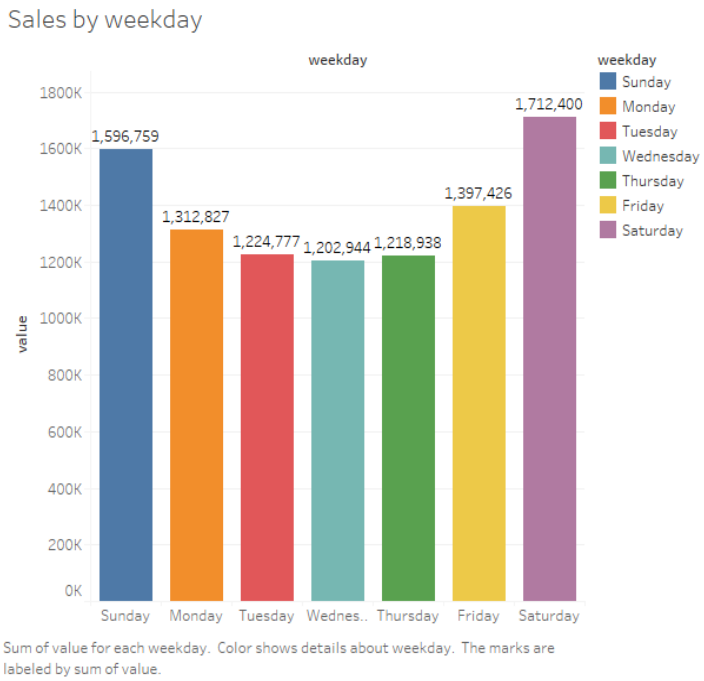
The above graph represents sales in different states and stores in it and based on events like cultural, national, religious and sporting. Where the highest is California, which is about 120k in which again the highest is sales occurred in religious events in CA_3 id. Followed by Texas and Wisconsin.

Sales by Store/cat

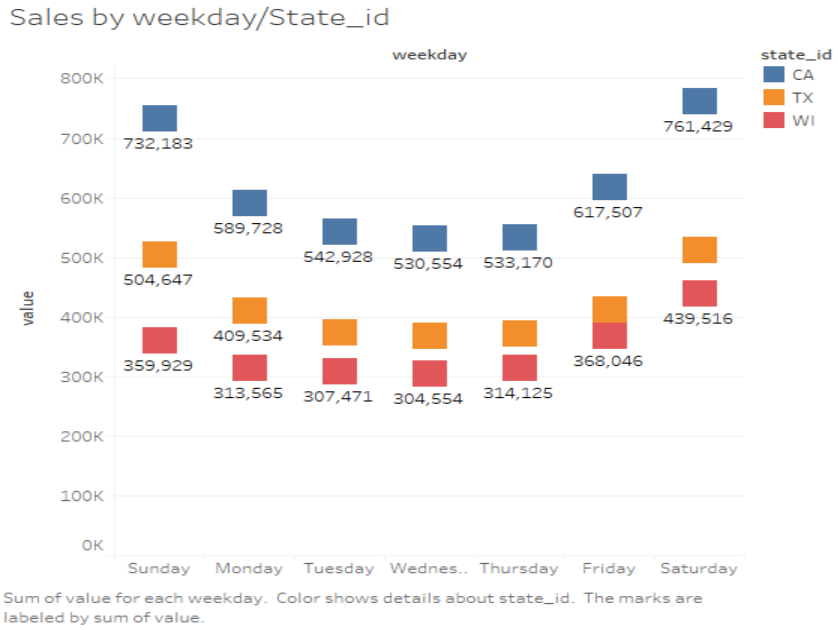


Sum of value for each Store Id. Color shows details about cat_id. The marks are labeled by sum of value.

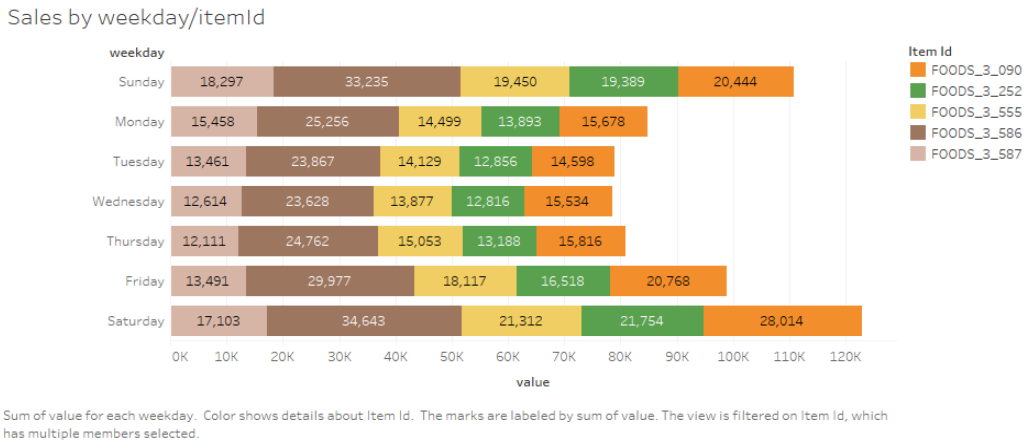
The above graph represents sales in different states and stores in it and based on categories like foods, hobbies and household. Where the highest is California, which is about 1600k in which again the highest is sales occurred in foods department in CA_3 id. Followed by Wisconsin, Texas.



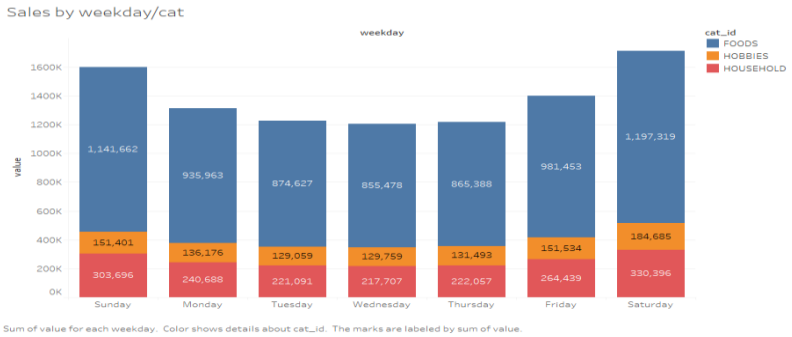
The above graph represents sales based on weekly data. Where the highest sales occurred on Saturday, which is about 1.7M followed by Sunday and Monday.



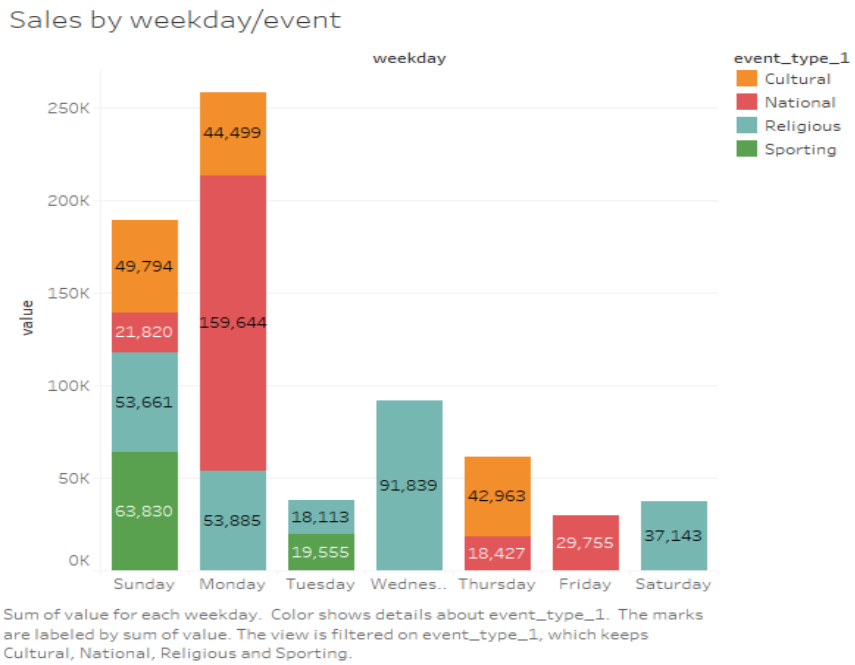
The above graph represents sales in different states and based weekly data. Where the highest is California, which is about 761k in which occurred on Saturday. Followed by Texas and Wisconsin.



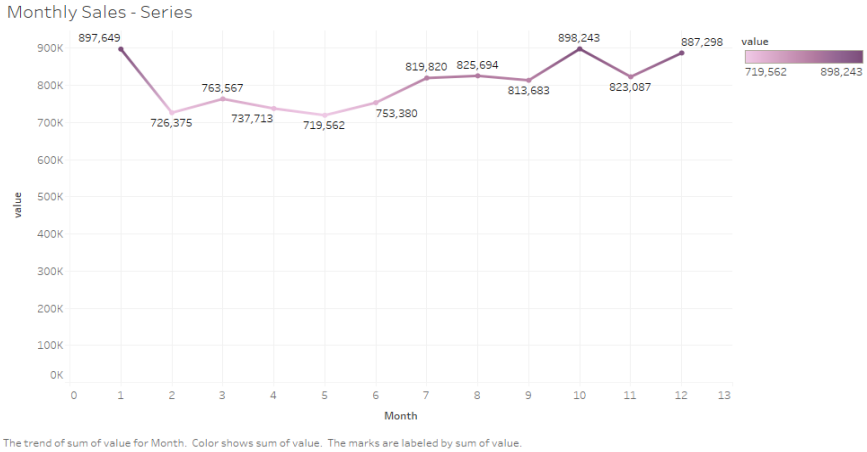
The above graph represents sales in different foods and weekly based data. Where the highest sales occurred on Saturday which is about 120k, where highest sales are of foods_3_586 which is about 34k. Followed by Sunday and Saturday.



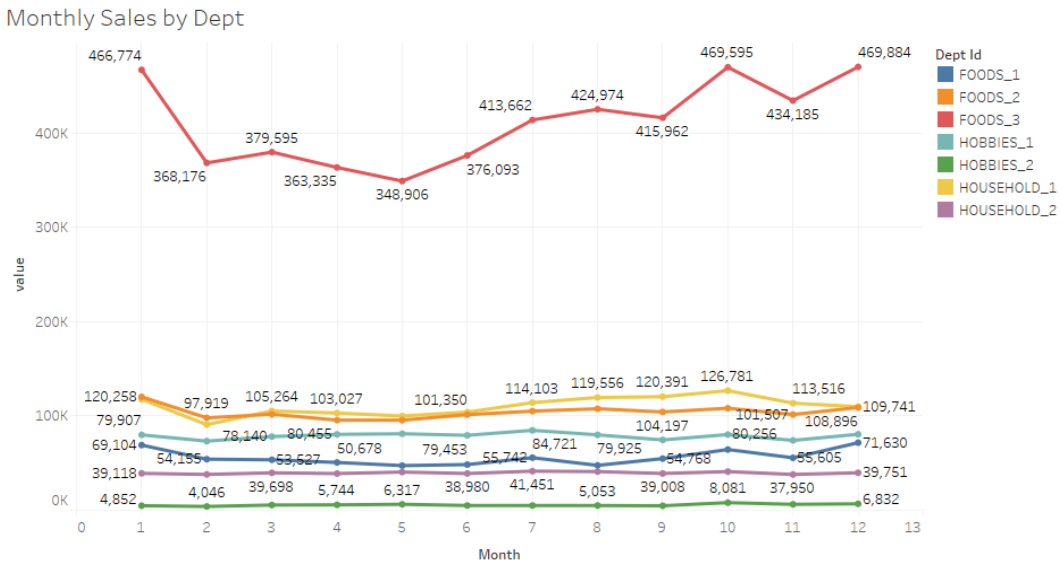
The above graph represents sales in different categories like foods, hobbies and households and based weekly data. Where the highest is Saturday, which is about 1600k in which again the highest is sales is in food category which is 1.1M. Followed by Sunday and Friday.



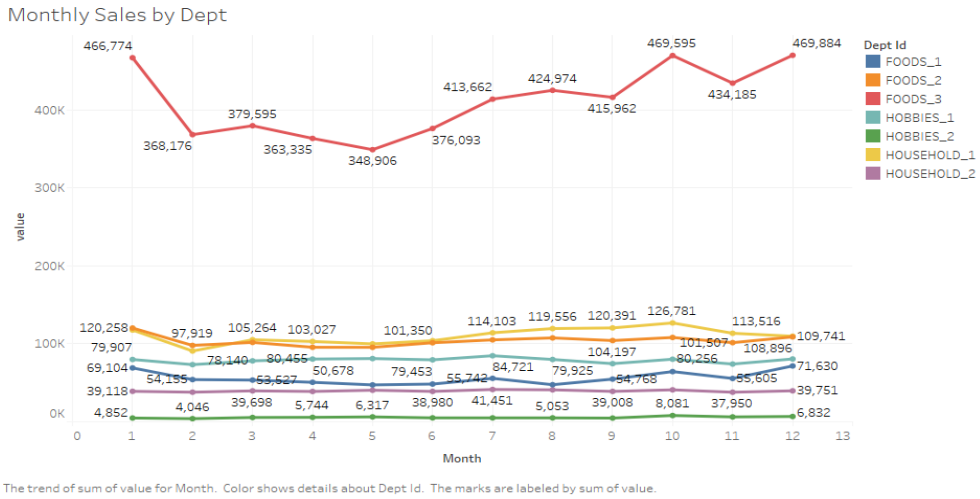
The above graph represents sales weekly based data and event based. Where the highest on Monday, which is about 250k in which again the highest is sales occurred on national events about 159k Followed by Sunday and Wednesday.



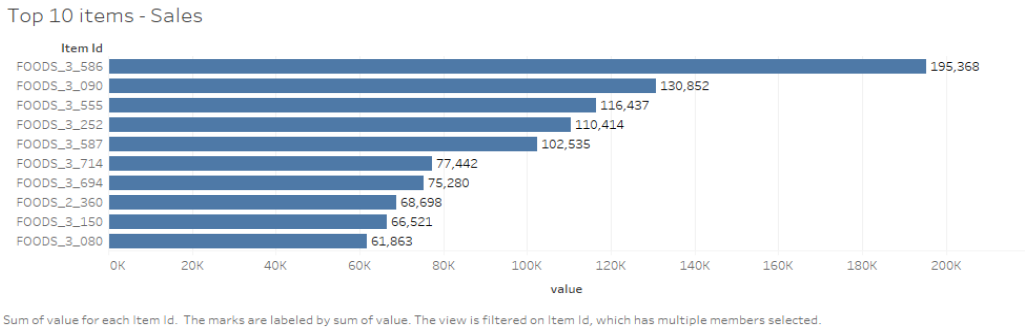
The above graph represents sales in series monthly wise and the highest is about 900k in the first month.



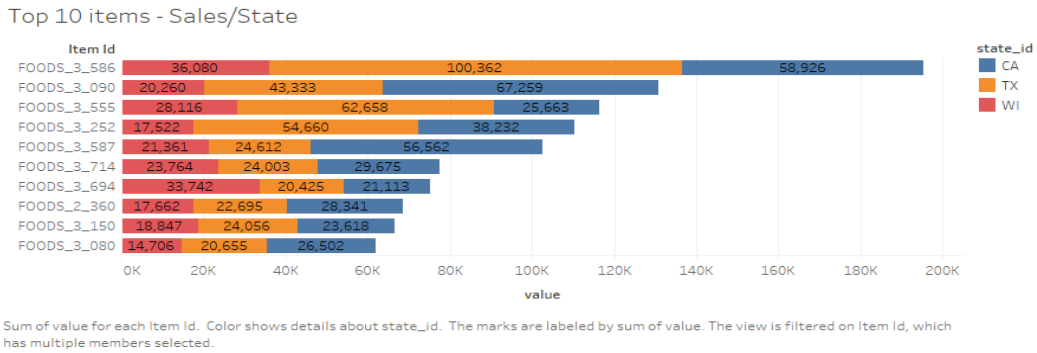
The above graph represents sales in different states and store-based data. Where the highest is California, which is about 157k in which again the highest is sales occurred in CA_3. Followed by Wisconsin and Texas.



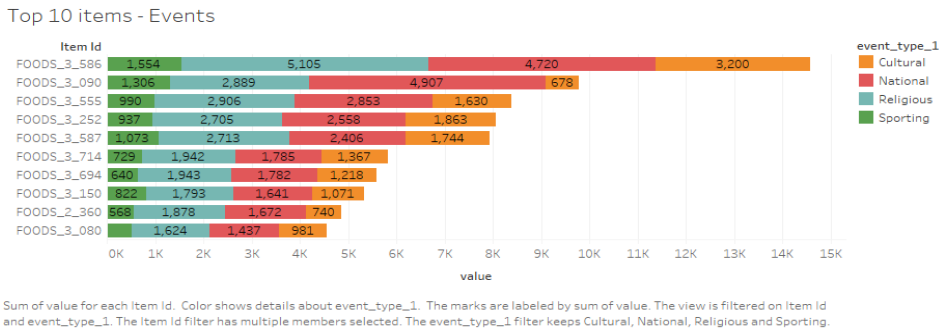
The above graph represents sales in different departments for each month. Where the highest is in first month, which is about 466k in which again the highest is sales occurred in foods_3 dept id. Followed by 10th month in household dept and foods_2 in 1st month.



The above graph represents sales in different items in foods. Where the highest is foods_3_586, which is about 195k.



The above graph represents sales in different states and in foods dept. Where the highest is Foods_3_586 dept, which is about 180k in which again the highest is sales occurred in California. Followed by Texas and Wisconsin.



The above graph represents sales in different events and in foods dept. Where the highest is Foods_3_586 dept, which is about 14k in which again the highest is sales occurred in Religious events. Followed by national and cultural.

Total sales got is : 9,666,071

B. Feature engineering: create two new features using the information provided in Table 1.

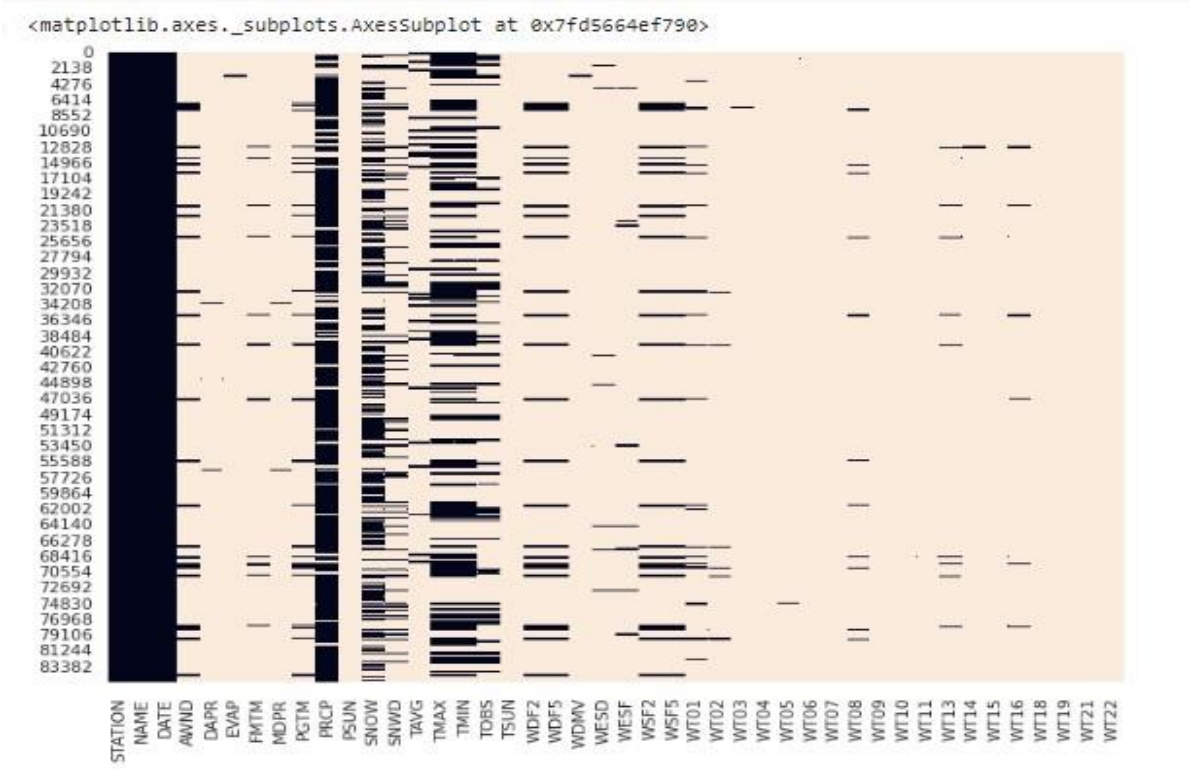
a) weather data
b) median income

Weather data across various stores:

Weather data was obtained from the given URL and was preprocessed and merged to the existing store data.

The store-weather station mapping was also performed.

	DATE	PRCP	SNOW	TMAX	TMIN	store_id
0	2011-01-29	0	0	62.0	46.0	CA_1
1	2011-01-30	1	0	60.0	52.0	CA_1
2	2011-01-31	0	0	63.0	44.0	CA_1
3	2011-02-01	0	0	62.0	46.0	CA_1
4	2011-02-02	0	0	65.0	42.0	CA_1

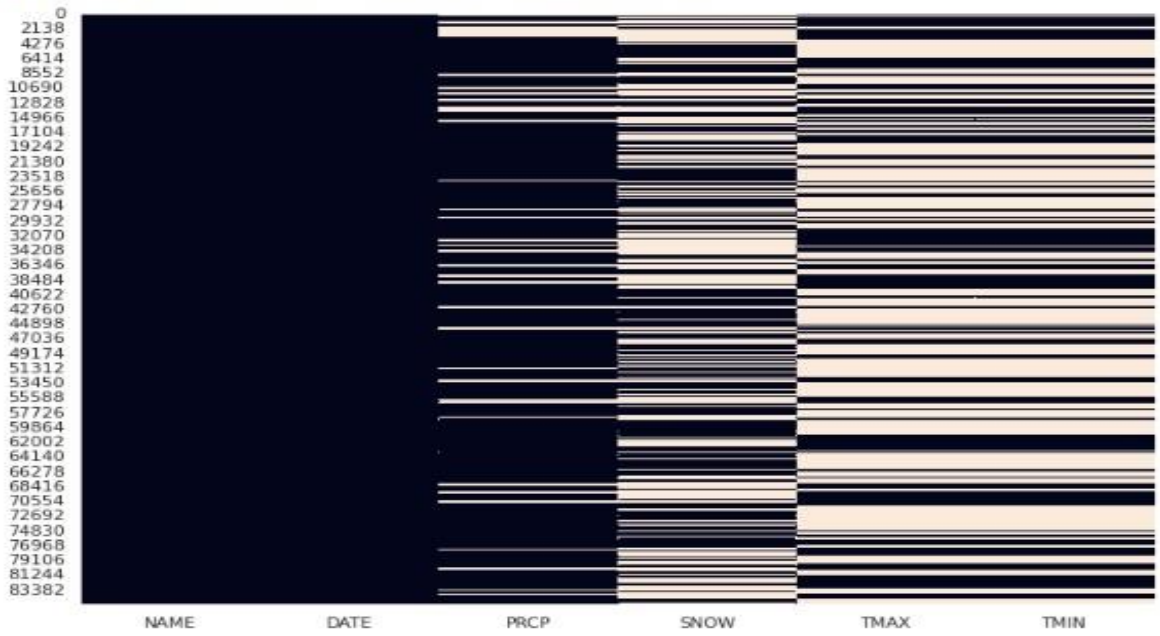


The weather data contains various columns that are not required for our problem statement.

In addition to that, most of the features have very high number of missing values.

```
[ ] sns.heatmap(df_weather.isnull(), cbar=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd567243fd0>
```



The features that had a high number of missing values have been removed.

The snow and precipitation were converted to categorical data.

Store weather station mapping:

	store_id	NAME
0	CA_1	LONG BEACH DAUGHERTY AIRPORT, CA US
1	CA_4	SAN FRANCISCO INTERNATIONAL AIRPORT, CA US
2	CA_2	SAN JOSE, CA US
3	CA_3	SAN DIEGO BROWN FIELD, CA US
4	TX_1	DALLAS REDBIRD AIRPORT, TX US
5	TX_2	DALLAS REDBIRD AIRPORT, TX US
6	TX_3	DALLAS REDBIRD AIRPORT, TX US
7	WI_3	MADISON DANE CO REGIONAL AIRPORT, WI US
8	WI_1	MILWAUKEE MITCHELL AIRPORT, WI US
9	WI_2	MILWAUKEE MITCHELL AIRPORT, WI US

The median income for every city:

	item_id	dept_id	cat_id	store_id	state_id	value	date	sell_price	PRCP	SNOW	TMAX	TMIN	median_income
0	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	2011-01-29	NaN	0.0	0.0	62.0	46.0	41642.0
1	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	2011-01-29	NaN	0.0	0.0	62.0	46.0	41642.0
2	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	2011-01-29	NaN	0.0	0.0	62.0	46.0	41642.0
3	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	2011-01-29	NaN	0.0	0.0	62.0	46.0	41642.0
4	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	2011-01-29	NaN	0.0	0.0	62.0	46.0	41642.0

So to conclude the final data frame contains:

- Store-Data
- Sales-Price
- Calendar Data
- Weather Data
- Median-Income

C. First create model without using any external features, and then create model with the external features.

Begin with ARIMA and compare the RMSE values for each category

Down-casting:- We can see that the memory usage was reduced from 215.5 to 134.7. We converted float64 to float32 and int32 and int64 to int16.

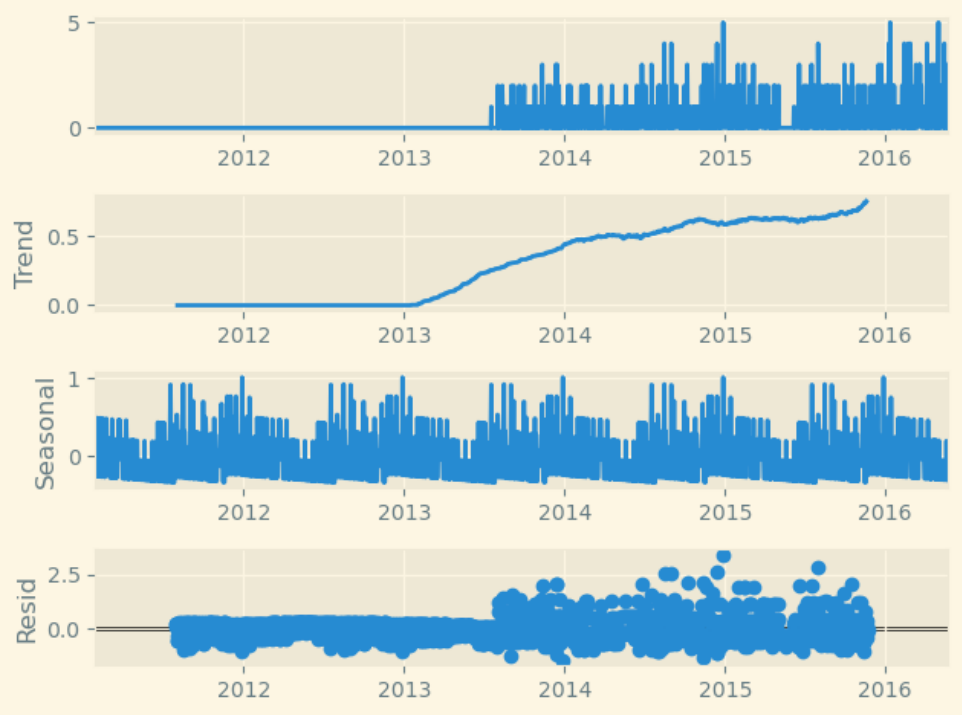
```
df_calendar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1969 entries, 0 to 1968
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   date          1969 non-null   object
 1   wm_yr_wk      1969 non-null   int64 
 2   weekday       1969 non-null   object
 3   wday          1969 non-null   int64 
 4   month         1969 non-null   int64 
 5   year          1969 non-null   int64 
 6   d             1969 non-null   object
 7   event_name_1  162 non-null    object
 8   event_type_1  162 non-null    object
 9   event_name_2  5 non-null      object
 10  event_type_2  5 non-null      object
 11  snap_CA       1969 non-null   int64 
 12  snap_TX       1969 non-null   int64 
 13  snap_WI       1969 non-null   int64 
dtypes: int64(7), object(7)
memory usage: 215.5+ KB
```

```
#Downcasting Calendar
df_calendar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1969 entries, 0 to 1968
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   date          1969 non-null   object
 1   wm_yr_wk      1969 non-null   int16 
 2   weekday       1969 non-null   object
 3   wday          1969 non-null   int16 
 4   month         1969 non-null   int16 
 5   year          1969 non-null   int16 
 6   d             1969 non-null   object
 7   event_name_1  162 non-null    object
 8   event_type_1  162 non-null    object
 9   event_name_2  5 non-null      object
 10  event_type_2  5 non-null      object
 11  snap_CA       1969 non-null   int16 
 12  snap_TX       1969 non-null   int16 
 13  snap_WI       1969 non-null   int16 
dtypes: int16(7), object(7)
memory usage: 134.7+ KB
```

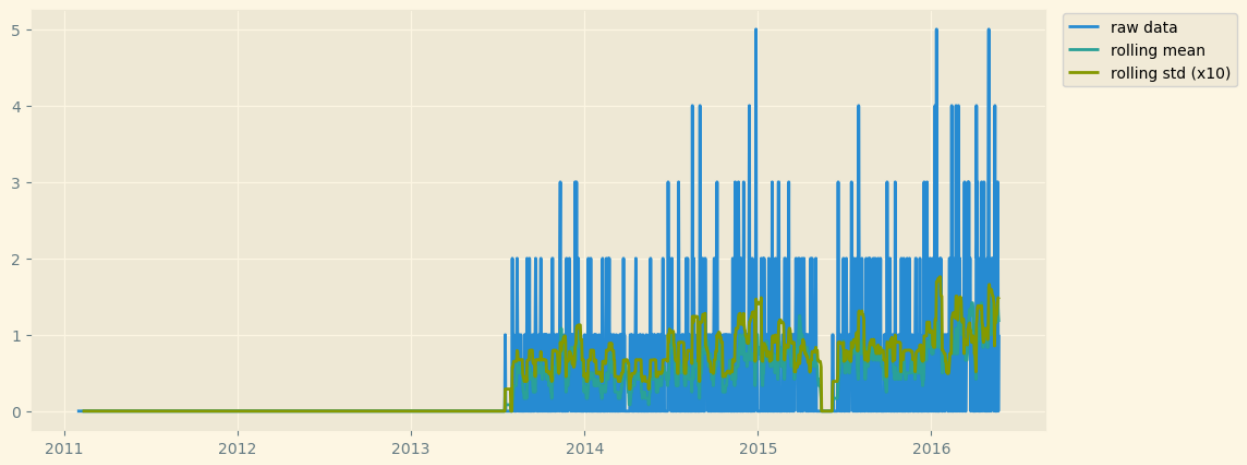
Seasonality and Trend:



Seasonality is a characteristic of a time series in which the data experiences regular and predictable changes that recur every calendar year. Any predictable fluctuation or pattern that recurs or repeats over a one-year period is said to be seasonal.
Seasonality was observed on weekends, special days, and holidays.

Checking if the data is stationary:

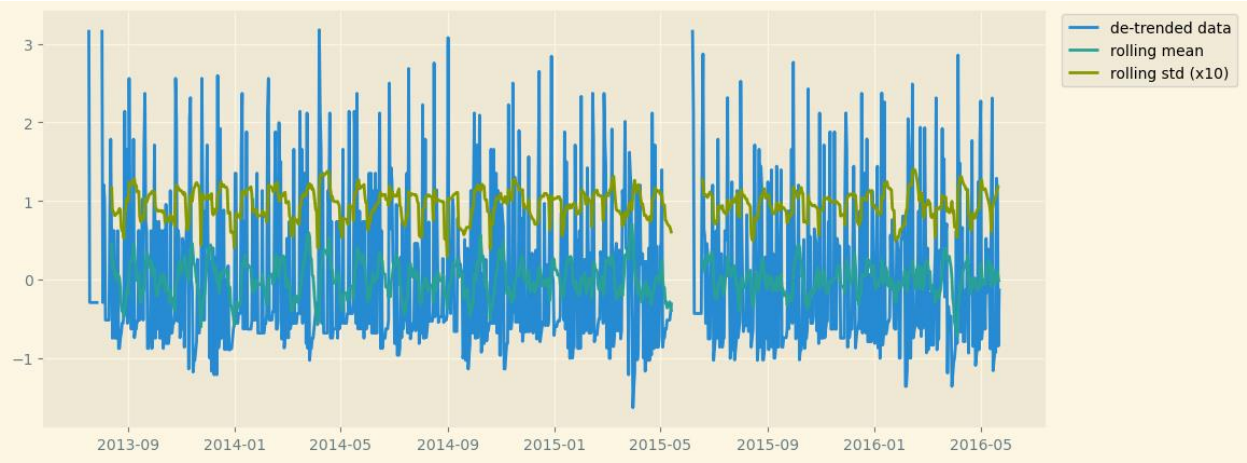
To determine if our series is stationary or non-stationary, plotting the rolling mean and variance is a suitable statistic to use. If rolling statistics exhibit a distinct trend (upward or downward) in the mean and fluctuate in variance (increasing or decreasing amplitude) in the standard deviation, the series may not be stationary



As shown by the figure above, the series is non-stationary, both the mean and variance change. Using the Dickey-fuller test, this test will produce a result known as a "test-statistic", based on which you can determine whether the time series is stationary or not with varying degrees of confidence.

Making data Stationary:

De-trending the series - To eliminate the trend in the series, we will build a de-trended column whose values are split by rolling variance and removed from the rolling mean.



Removing Trend and Seasonality:

Determining whether our de-trended series is stationary by using rolling analysis and the Adfuller test. We will choose the window of 12 as we have 12-period seasonality.

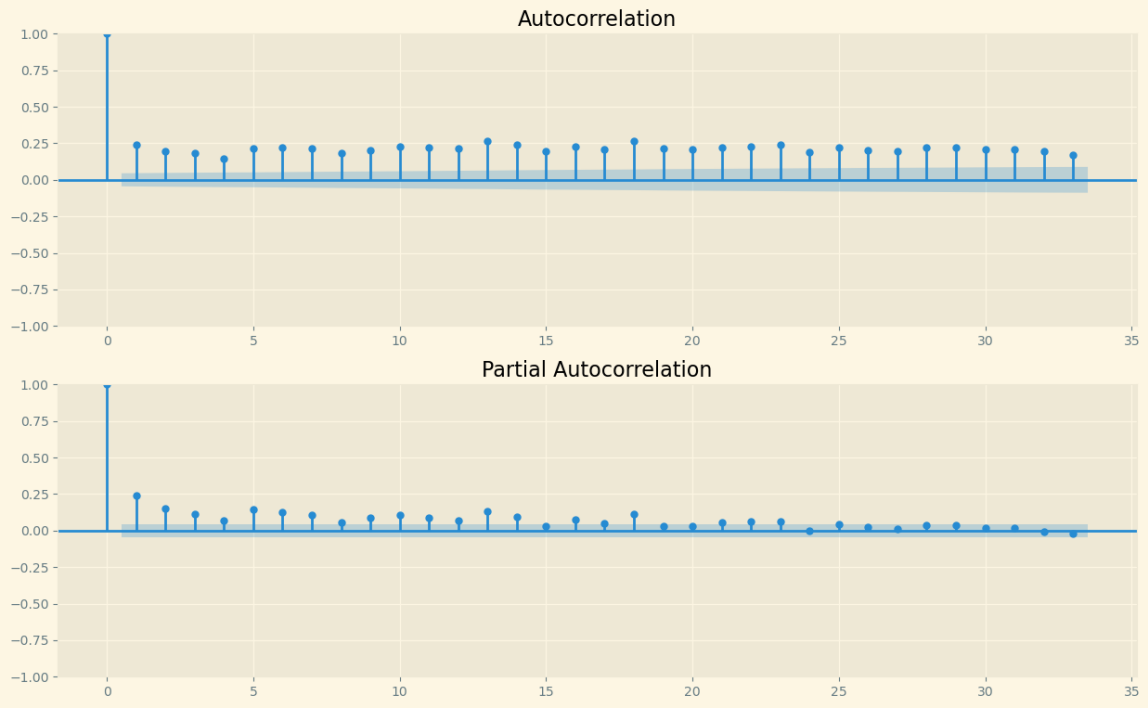
```
adfuller_test(example.deseasonal_sales.dropna())
```

```
Checking if the data is stationary
Test statistic = -12.743
P-value = 0.000
Critical values:
        1%: -3.437109473790722 - The data is  stationary with 99% confidence
        5%: -2.8645242345396436 - The data is  stationary with 95% confidence
        10%: -2.568358964820916 - The data is  stationary with 90% confidence
```

De-seasoning the series - This is accomplished by determining which states, seasonal or cyclic patterns can be removed by subtracting periodical values (this subtraction is performed on detrended series).
The variance/std has an increasing decreasing amplitude.
The above test indicates that our de-seasonal dataset is free of trend and seasonality.

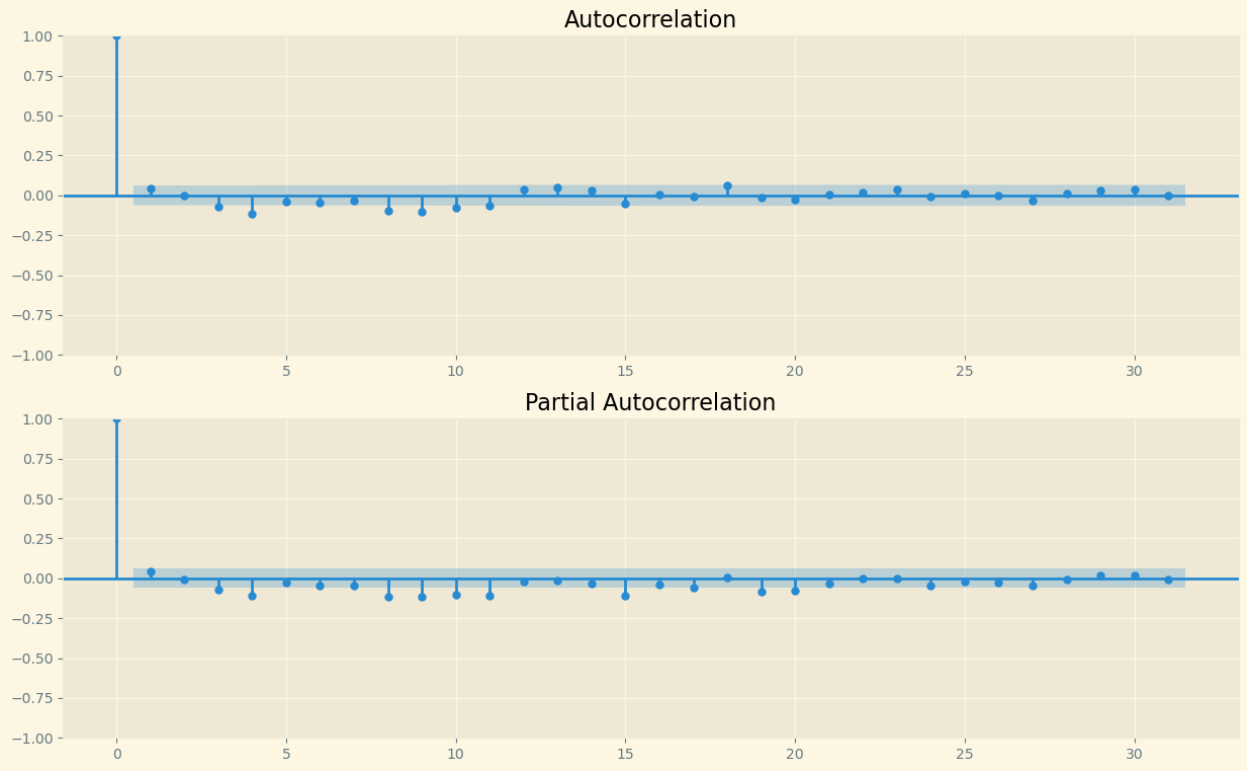
Autocorrelation plots:

Lags = The amount by which we are shifting the time-series data to find correlations of current sales data with previous time-stamp sales data
Auto-Correlation = ACF describes the autocorrelation between an observation and another observation at a prior time step that includes direct and indirect dependence information
Partial Autocorrelation = It finds the correlations in time-series by removing the indirect correlations as in AC



The AR (Auto-Regression) term and MA (Moving Average) term in the ARMA model can be derived using the ACF and PACF plots, respectively. PACF works well for identifying AR parts. The PACF "shuts off" for an AR part after the model order. Theoretically, when anything "shuts off," the PACF is equal to 0 past that value.

ACF is the most effective tool for MA part identification. In the case of an MA part, the PACF does not shut off but instead weavers around 0 in some way. Additionally, the ACF plot shows an exponential decline rather than a sudden termination.
We can set AR to 0 or 1 and MA to 1 or 2 based on our plots.



ARIMA:

ARIMA model - Autoregressive Integrated Moving Average model
It has three parts,
p - AR part
d - differencing part
q - MA part

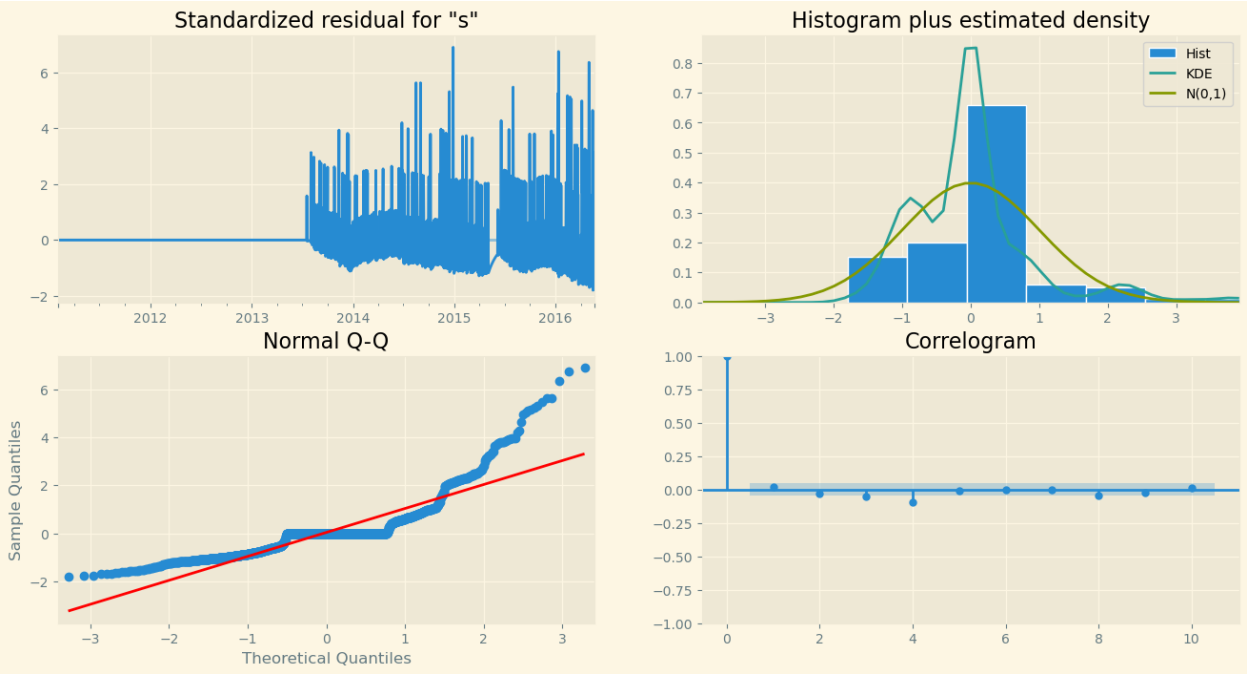
SARIMAX Results

Dep. Variable:	sales	No. Observations:	1941
Model:	SARIMAX(0, 1, 1)	Log Likelihood	-1865.477
Date:	Sat, 19 Nov 2022	AIC	3734.954
Time:	00:07:52	BIC	3746.093
Sample:	01-29-2011 - 05-22-2016	HQIC	3739.050
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-1.0258	0.005	-218.717	0.000	-1.035	-1.017
sigma2	0.3810	0.007	51.300	0.000	0.366	0.396

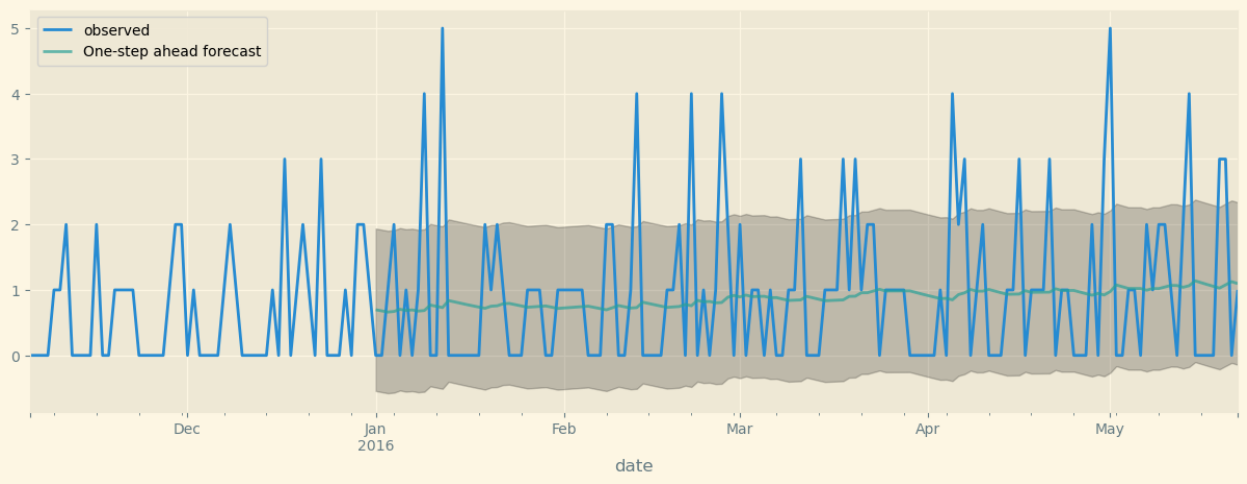
Ljung-Box (L1) (Q):	0.90	Jarque-Bera (JB):	6903.45
Prob(Q):	0.34	Prob(JB):	0.00
Heteroskedasticity (H):	inf	Skew:	2.20
Prob(H) (two-sided):	0.00	Kurtosis:	11.13

SARIMAX(0,1,1) gives minimum and efficient value which we used in the matrix
Below graphs show diagnostics for the model.



Validating Forecast:

To get a better understanding of the data and the forecasts produced by ARIMA, printing the sales for the last 200 days.



Performance metrics of ARIMA:

RMSE:

```
rmse = ((y_forecasted - y_truth)**2).mean()
print('The Mean Squared Error is {}'.format(np.sqrt(rmse), 2))
```

```
The Mean Squared Error is 1.200534809889901
```

The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values.

Long Short-Term Memory (LSTM) networks:

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.LSTM(units =64,input_shape = (np.array(X_train).shape[1], np.array(X_train).shape[2])))
model.add(tf.keras.layers.Dense(30490))

model.compile(
  loss='mean_squared_error',
  optimizer=tf.keras.optimizers.Adam(0.001)
)
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 64)                7822080

 dense (Dense)               (None, 30490)             1981850

=================================================================
Total params: 9,803,930
Trainable params: 9,803,930
Non-trainable params: 0
_________________________________________________________________
```

```
predictions
```

```
array([[0.15734568, 0.03269284, 0.08150573, ..., 0.02733296, 0.08075694,
        0.15835069]], dtype=float32)
```

Performance metrics of LSTM:

```
mae = np.mean(np.abs(inputs - predictions))

mse = np.mean((inputs - predictions)**2)

#mape = np.mean(np.abs(inputs - predictions) / np.abs(inputs))

print("The mean absolute error of the model = ", mae)
print("The mean square error of the model = ", mse)
#print("The mean absolute persentage error of the model = ", mape)
```

```
The mean absolute error of the model =  0.014408902103550805
The mean square error of the model =  0.0011762960917462593
```

LSTM is best fit for dataset 2 to predict sales compared to ARIMA as performance metrics of LSTM gives better accuracy compared to ARIMA.

SPRINT 3: Model deployment and business recommendation

1. Deploy the winning model on Stream lit for dataset-02

Streamlit:

1. Here we will take a dump of ARIMA model in a pickle file, classifier.pkl

```
[ ]  1 !pip install -q streamlit
```

```
|████████████████████████████████| 9.2 MB 6.8 MB/s
|████████████████████████████████| 78 kB 5.9 MB/s
|████████████████████████████████| 237 kB 66.7 MB/s
|████████████████████████████████| 4.7 MB 34.4 MB/s
|████████████████████████████████| 164 kB 56.5 MB/s
|████████████████████████████████| 182 kB 71.5 MB/s
|████████████████████████████████| 62 kB 1.4 MB/s
|████████████████████████████████| 51 kB 5.2 MB/s
Building wheel for validators (setup.py) ... done
```

```
[ ]  1 import streamlit as st
```

```
1 # saving the model
2 import pickle
3 pickle_out = open("classifier.pkl", mode = "wb")
4 pickle.dump(data, pickle_out)
5 pickle_out.close()
```

2. After dumping the model in pickle file, we will download it and load it in wenApp.py, which is our web application file.

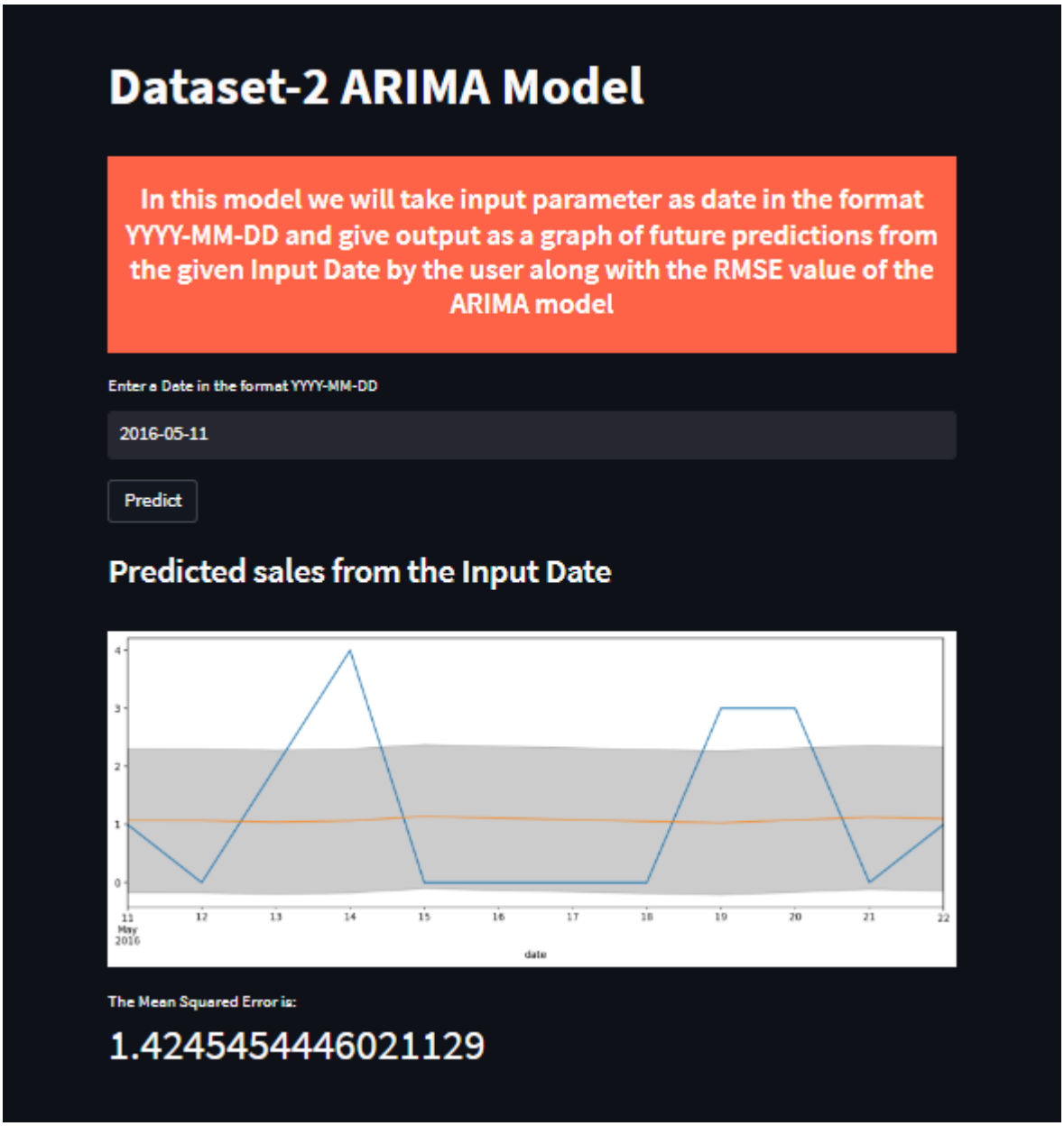
```
pickle_in = open("classifier.pkl","rb")
data = pickle.load(pickle_in)
```

3. We will take user input from the browser, pass it as a parameter to our function for prediction

```python
def main():

    st.title("Dataset-2 ARIMA Model")

    html_temp = """
    <div style="background-color:tomato;padding:10px">
    <h4 style="color:white;text-align:center;">In this model we will take input parameter as dat
    </div>
    """
    st.markdown(html_temp,unsafe_allow_html=True)

    st.write("")
    input_data = st.text_input("Enter a Date in the format YYYY-MM-DD")
    button1 = st.button("Predict")

    if button1:
        sales_prediction(input_data)
```

```python
def sales_prediction(input_data):

    model = sm.tsa.statespace.SARIMAX(data, order=(0,1,1), seasonal_order=(0,0,0,12),
                                      enforce_stationarity=False, enforce_invertibility=False)
    result = model.fit()

    pred = result.get_prediction(start=input_data, dynamic=False)
    pred_ci = pred.conf_int()


    with st.container():
        st.subheader('Predicted sales from the Input Date')
        ax=data[input_data:].plot(label='observed')
        pred.predicted_mean.plot(ax=ax, label='forecast', alpha=0.7, figsize=(15,5))
        ax.fill_between(pred_ci.index,
                        pred_ci.iloc[:, 0],
                        pred_ci.iloc[:, 1], color='k', alpha=.2)
        st.set_option('deprecation.showPyplotGlobalUse', False)
        st.pyplot()

        y_forecasted = pred.predicted_mean
        y_truth = data[input_data:]

        rmse = ((y_forecasted - y_truth)**2).mean()
        rmse_show =  np.sqrt(rmse)
        st.metric('The Mean Squared Error is:', rmse_show)
```

4. After executing the streamlit web application file in localhost, we have a application which take Date in the format YYYY-MM-DD and give prediction graph along with RMSE value of our model



The video recording link of deployment:
https://csulb-my.sharepoint.com/:v:/g/personal/kaushal_brahmbhatt01_student_csulb_edu/Eby5jy8XpNVMsAVA9qnL3UEBGNUcwEahFieMBT5PTHp4TA?e=cshqBX

Deployment code of stream lit:
https://github.com/KAUSHALBRAHMBHATT/arimastreamlit

2. **Logistic performance is impacted by product rotation and demand variability so perform ABC analysis.**
   A. Use first year data of household category to create ABC analysis and interpret the graph.

In this section, we perform the ABC analysis of household category of 1 year data using sales and selling price.

	item_id	sales	sell_price
565	HOUSEHOLD_1_001	3	6.32
566	HOUSEHOLD_1_002	1	6.32
568	HOUSEHOLD_1_004	4	1.98
569	HOUSEHOLD_1_005	0	10.72
575	HOUSEHOLD_1_011	1	4.63

Here, we construct a table in which we have household category and their sales, selling price. As we can observe that each item in this household category has its individual sales and it's own selling price and this selling price is for each item. Then we construct another table.

	item_id	sales	sell_price	revenue
565	HOUSEHOLD_1_001	3	6.32	18.960001
566	HOUSEHOLD_1_002	1	6.32	6.320000
568	HOUSEHOLD_1_004	4	1.98	7.920000
575	HOUSEHOLD_1_011	1	4.63	4.630000
589	HOUSEHOLD_1_025	2	2.98	5.960000

From the above table, we can observe that, revenue has been calculated by multiplying the sales and selling price of each item. The generated revenue is displayed towards the right.

	item_id	sales	sell_price	revenue	RunCumCost	TotSum	RunPerc	Class
3095784	HOUSEHOLD_1_494	57	7.97	454.289988	4.542900e+02	7.471258e+06	0.000061	A
495028	HOUSEHOLD_1_535	57	6.97	397.289988	8.515800e+02	7.471258e+06	0.000114	A
2443080	HOUSEHOLD_1_272	35	9.97	348.950009	1.200530e+03	7.471258e+06	0.000161	A
2269071	HOUSEHOLD_1_053	23	14.97	344.310006	1.544840e+03	7.471258e+06	0.000207	A
2961561	HOUSEHOLD_1_427	56	5.96	333.760002	1.878600e+03	7.471258e+06	0.000251	A
...	...	...	...	...	...	...	...	...
6163495	HOUSEHOLD_2_371	1	0.78	0.780000	7.471255e+06	7.471258e+06	1.000000	C
4257870	HOUSEHOLD_2_371	1	0.78	0.780000	7.471255e+06	7.471258e+06	1.000000	C
10310135	HOUSEHOLD_2_371	1	0.78	0.780000	7.471256e+06	7.471258e+06	1.000000	C
6245818	HOUSEHOLD_2_371	1	0.78	0.780000	7.471257e+06	7.471258e+06	1.000000	C
7388346	HOUSEHOLD_1_057	1	0.50	0.500000	7.471258e+06	7.471258e+06	1.000000	C

736421 rows × 8 columns

In the above table, we have extracted the run cum cost, total sum, run percentage and categorized them into classes for each item in the household category.

	item_id	sales	sell_price	revenue	Class
3095784	HOUSEHOLD_1_494	57	7.97	454.289988	A
495028	HOUSEHOLD_1_535	57	6.97	397.289988	A
2443080	HOUSEHOLD_1_272	35	9.97	348.950009	A
2269071	HOUSEHOLD_1_053	23	14.97	344.310006	A
2961561	HOUSEHOLD_1_427	56	5.96	333.760002	A
...	...	...	...	...	...
6163495	HOUSEHOLD_2_371	1	0.78	0.780000	C
4257870	HOUSEHOLD_2_371	1	0.78	0.780000	C
10310135	HOUSEHOLD_2_371	1	0.78	0.780000	C
6245818	HOUSEHOLD_2_371	1	0.78	0.780000	C
7388346	HOUSEHOLD_1_057	1	0.50	0.500000	C

From the above table, we have narrowed down to only few necessary columns like sales, selling price, revenue and class.

Each item has been categorized into classes. It has been categorized based on revenue generation verses the total number of sales.

```
C    303277
B    234279
A    198865
Name: Class, dtype: int64
```

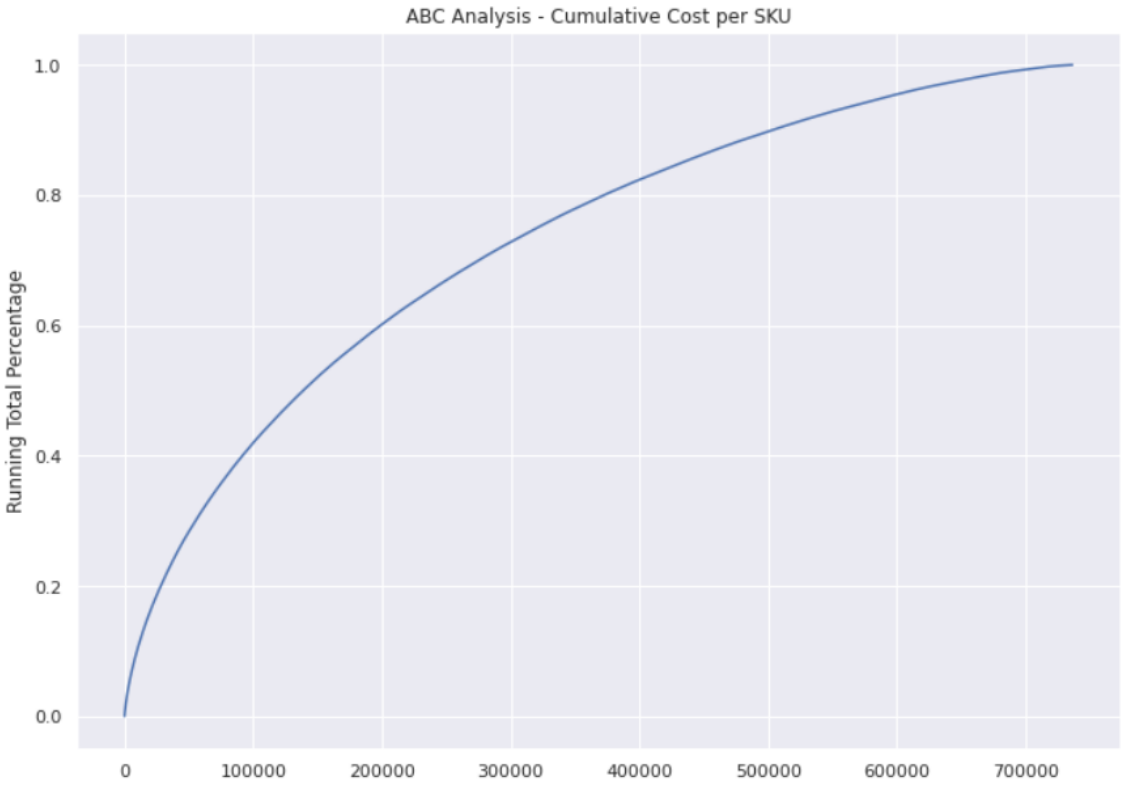
From the above, we can infer that category C has highest sales.

```
Cost of Class A: 4482753.738527
Cost of Class B: 1867811.3132520318
Cost of Class C: 1120692.4770283997
```

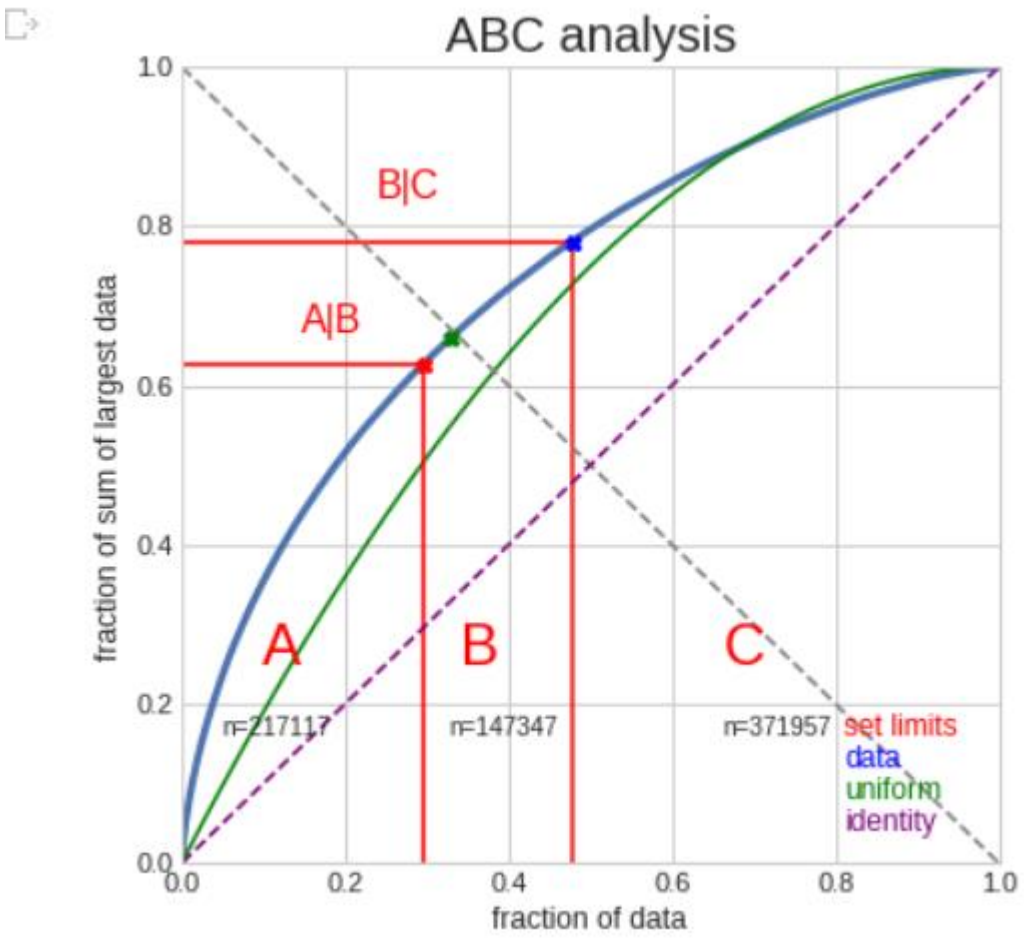
From the above, we can infer that the items from Class A has the highest revenue generation.

```
Percent of Cost of A: 0.5999998957662139
Percent of Cost of B: 0.24999958923249344
Percent of Cost of C: 0.15000051500129263
```

From the above, we can infer that category A has the highest percent of cost with a value of 0.599



The above graph represents the curve which is plotted in running total percentage versus Cumulative cost per SKU. As the cost per SKU increases, the total percentage share of revenue also increases.



The above graph represents ABC analysis.

B. How stable is the customers' demand? (Coefficient of Variation)

Co-efficient of variation for yearly data to check stability that is customer demand in the order of less sales to highest.

Checked demand variability in terms of CV(from mean and standard deviation) for all departments for each day. Sample of one day's customer demand or coefficient of variation is as below:

	item_id	dept_id	cat_id	d	value	wm_yr_wk	wday	month	year	sell_price	...	mean	std	sect	t	t%
0	FOODS_3_298	FOODS_3	FOODS	d_1	8.0	111010.0	10.0	10.0	20110.0	172.319992	...	20.341434	73.179038	16741.0	2.884809e+06	1.170873
1	FOODS_3_083	FOODS_3	FOODS	d_1	18.0	111010.0	10.0	10.0	20110.0	171.319992	...	20.341434	73.179038	16741.0	2.868068e+06	1.164078
2	FOODS_3_242	FOODS_3	FOODS	d_1	12.0	111010.0	10.0	10.0	20110.0	77.739998	...	20.341434	73.179038	16741.0	1.301445e+06	0.528225
3	FOODS_3_535	FOODS_3	FOODS	d_1	13.0	111010.0	10.0	10.0	20110.0	66.799995	...	20.341434	73.179038	16741.0	1.118299e+06	0.453890
4	FOODS_3_469	FOODS_3	FOODS	d_1	43.0	111010.0	10.0	10.0	20110.0	66.799995	...	20.341434	73.179038	16741.0	1.118299e+06	0.453890

d	value	wm_yr_wk	wday	month	year	sell_price	...	mean	std	sect	t	t%	tcs%	sid	sk%	ABC	cv
d_1	8.0	111010.0	10.0	10.0	20110.0	172.319992	...	20.341434	73.179038	16741.0	2.884809e+06	1.170873	1.170873	1.0	0.032798	A	3.597536
d_1	18.0	111010.0	10.0	10.0	20110.0	171.319992	...	20.341434	73.179038	16741.0	2.868068e+06	1.164078	2.334952	2.0	0.065595	A	3.597536
d_1	12.0	111010.0	10.0	10.0	20110.0	77.739998	...	20.341434	73.179038	16741.0	1.301445e+06	0.528225	2.863176	3.0	0.098393	A	3.597536
d_1	13.0	111010.0	10.0	10.0	20110.0	66.799995	...	20.341434	73.179038	16741.0	1.118299e+06	0.453890	3.317066	4.0	0.131191	A	3.597536
d_1	43.0	111010.0	10.0	10.0	20110.0	66.799995	...	20.341434	73.179038	16741.0	1.118299e+06	0.453890	3.770956	5.0	0.163988	A	3.597536

The high value of CV depicts the unstable customer demand that might cause out of stock issues and workload peaks so for example the days like 331, 295 , 305 and others have unstable customer demand as their CV value is low.

C. Discuss a few initiatives and recommendations for improving the retail business for dataset_02.

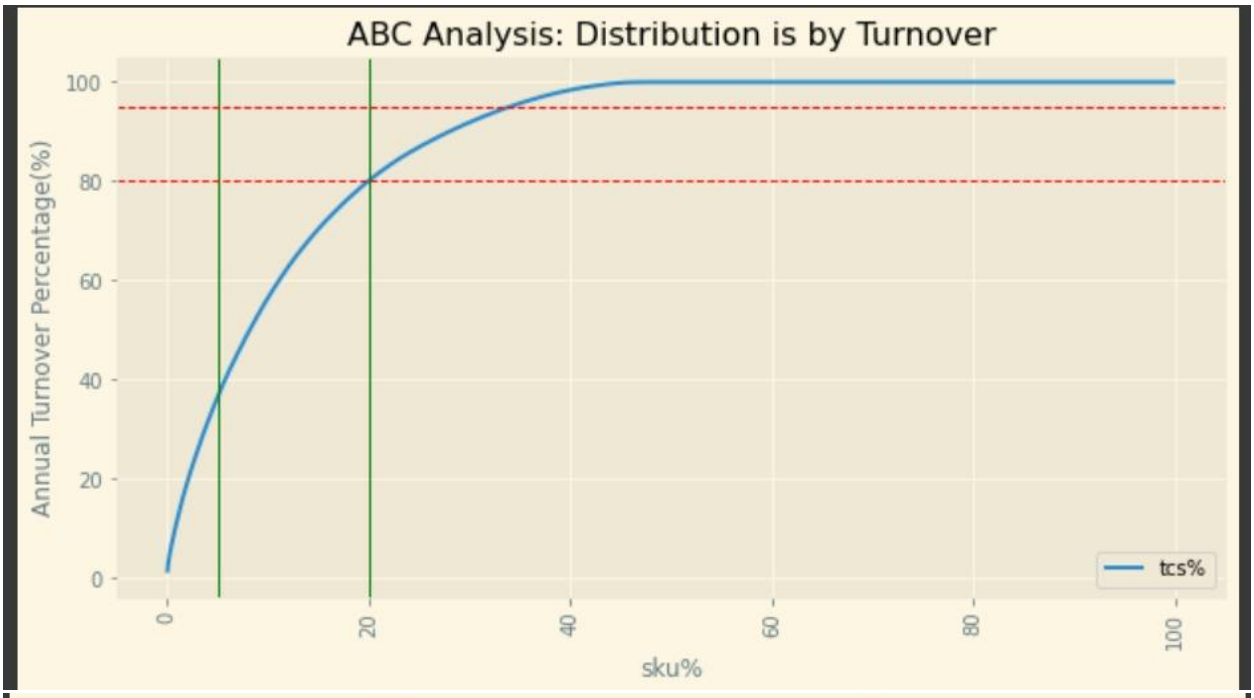
Below is the category ABC analysis example using dataframe for category c:

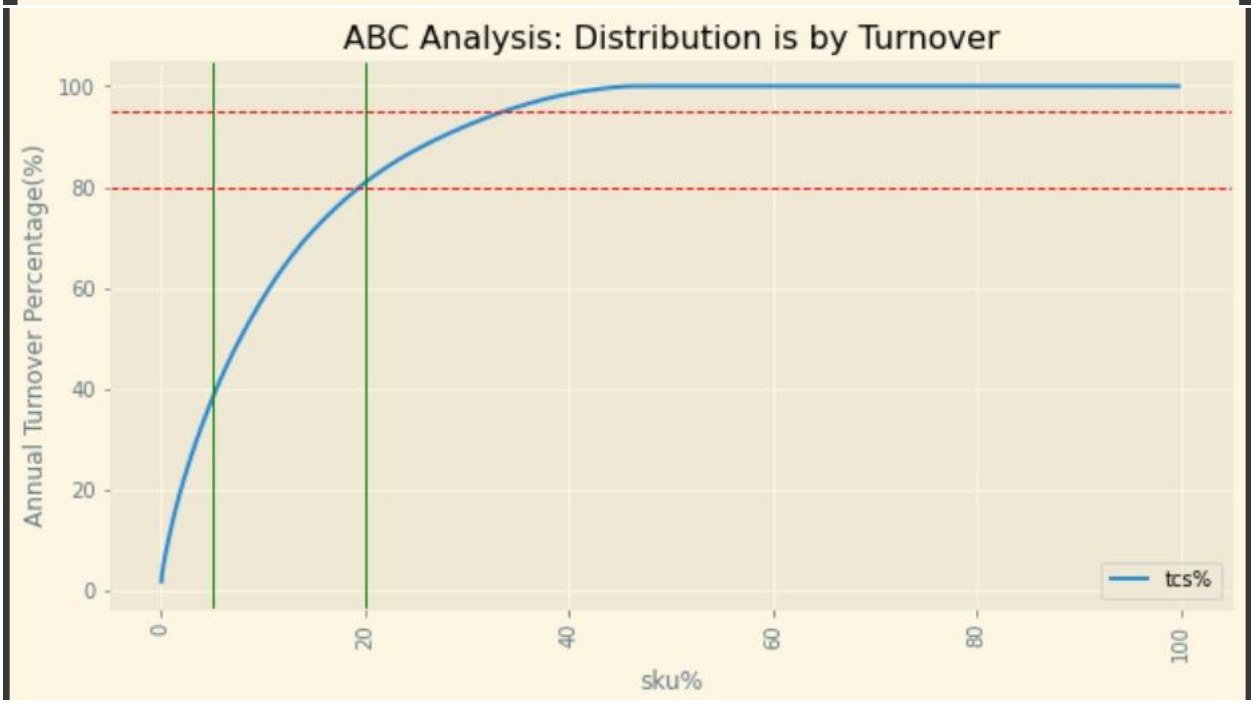
	item_id	dept_id	cat_id	d	value	wm_yr_wk	wday	\
1524	HOUSEHOLD_1_173	HOUSEHOLD_1	HOUSEHOLD	d_1	0.0	111010.0	10.0	
1525	HOUSEHOLD_1_499	HOUSEHOLD_1	HOUSEHOLD	d_1	0.0	111010.0	10.0	
1526	FOODS_2_129	FOODS_2	FOODS	d_1	0.0	111010.0	10.0	
1527	HOUSEHOLD_1_497	HOUSEHOLD_1	HOUSEHOLD	d_1	0.0	111010.0	10.0	
1528	HOUSEHOLD_1_496	HOUSEHOLD_1	HOUSEHOLD	d_1	0.0	111010.0	10.0	

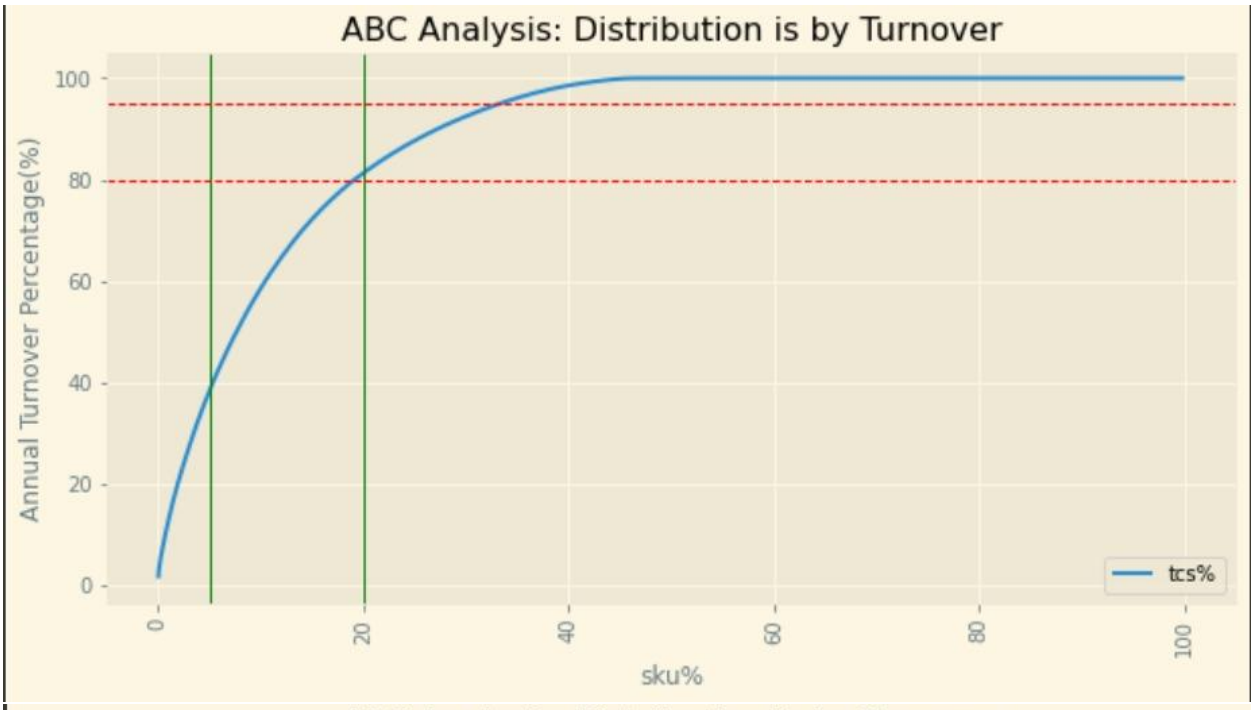
	month	year	sell_price	...	mean	std	sect	t	\
1524	10.0	20110.0	0.0	...	7.716165	15.638702	4105.0	0.0	
1525	10.0	20110.0	0.0	...	7.716165	15.638702	4105.0	0.0	
1526	10.0	20110.0	0.0	...	10.286432	21.351287	4094.0	0.0	
1527	10.0	20110.0	0.0	...	7.716165	15.638702	4105.0	0.0	
1528	10.0	20110.0	0.0	...	7.716165	15.638702	4105.0	0.0	

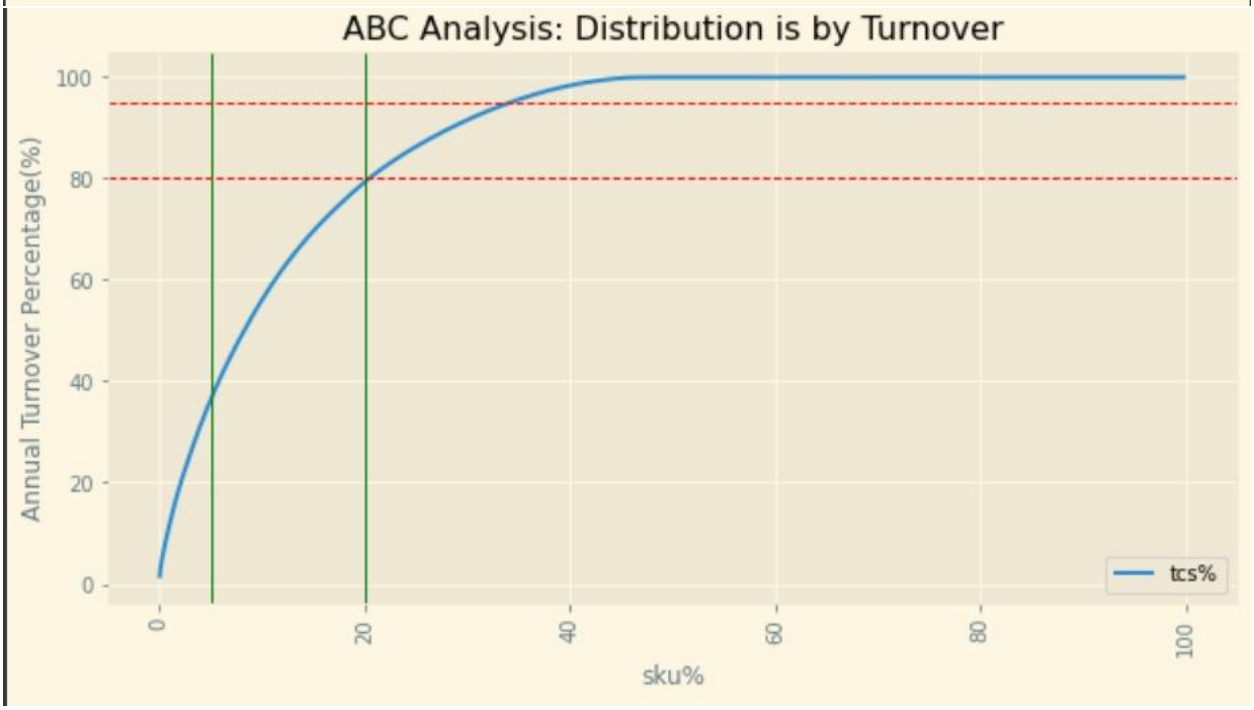
	t%	tcs%	sid	sk%	ABC	cv
1524	0.0	100.0	1525.0	50.016399	C	2.026745
1525	0.0	100.0	1526.0	50.049196	C	2.026745
1526	0.0	100.0	1527.0	50.081994	C	2.075675
1527	0.0	100.0	1528.0	50.114792	C	2.026745
1528	0.0	100.0	1529.0	50.147589	C	2.026745

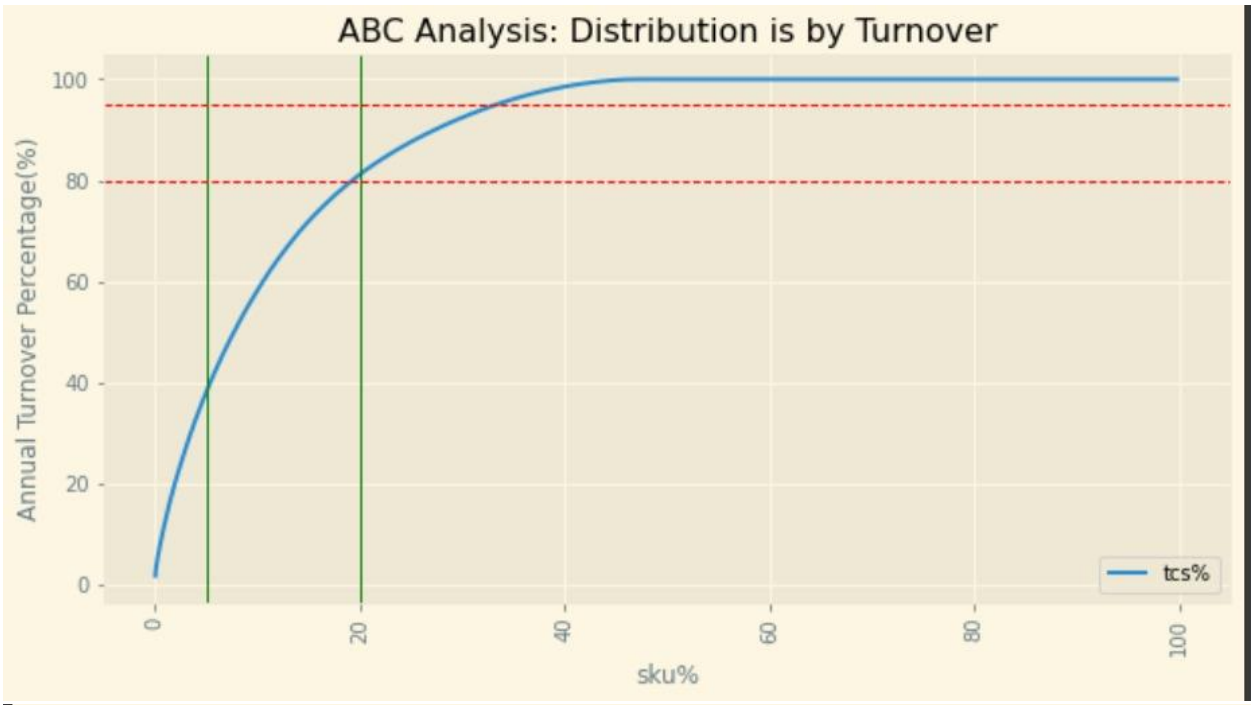
Graph analysis in terms of ABC for days which has low sales(took 10 days which has lowest sales) is as below :

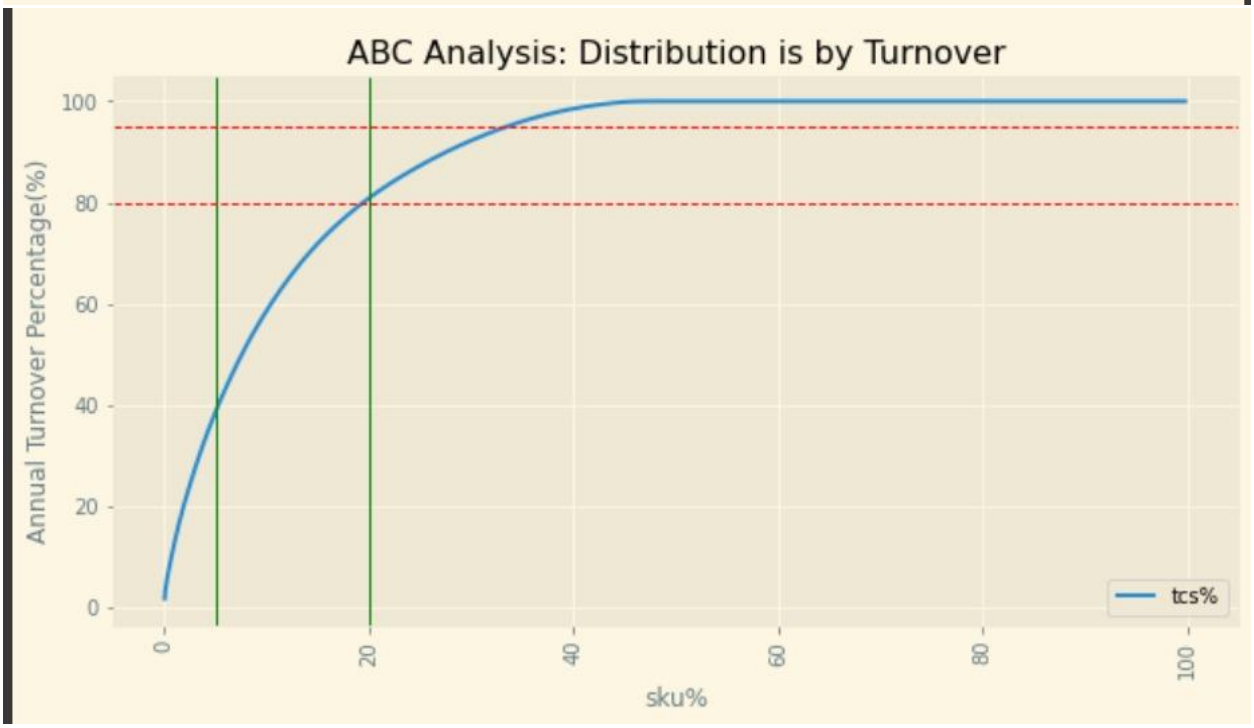

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

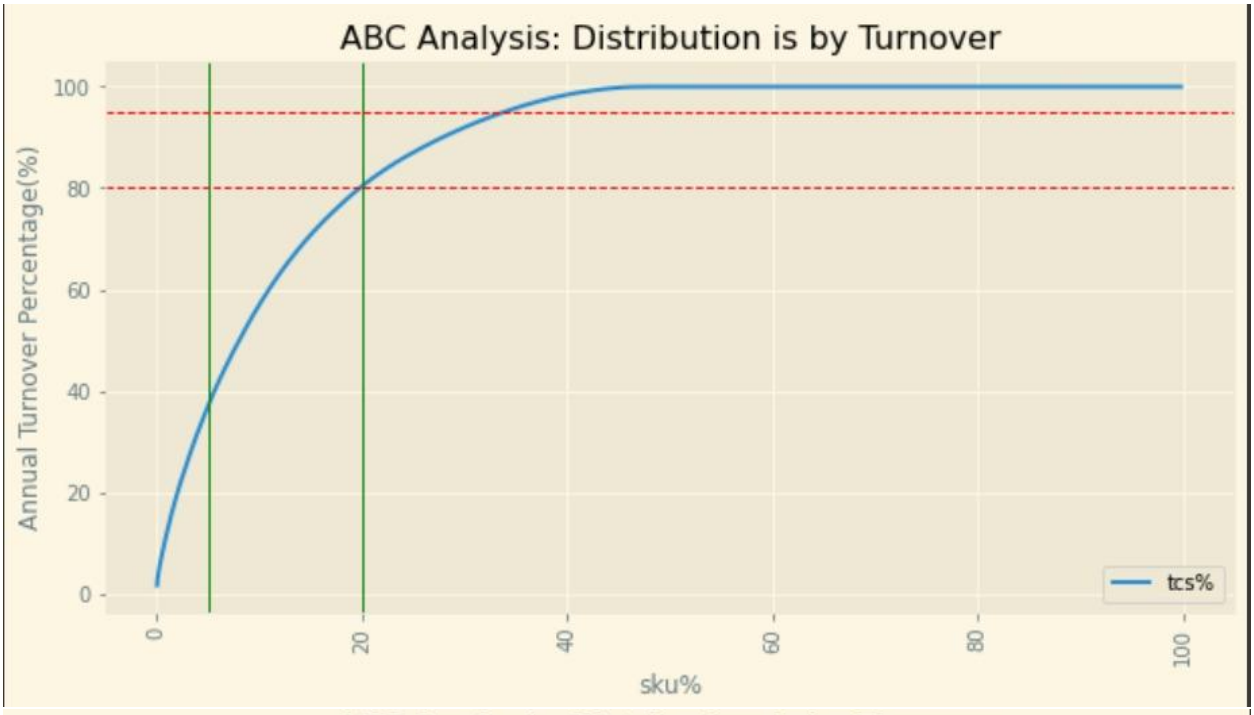

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

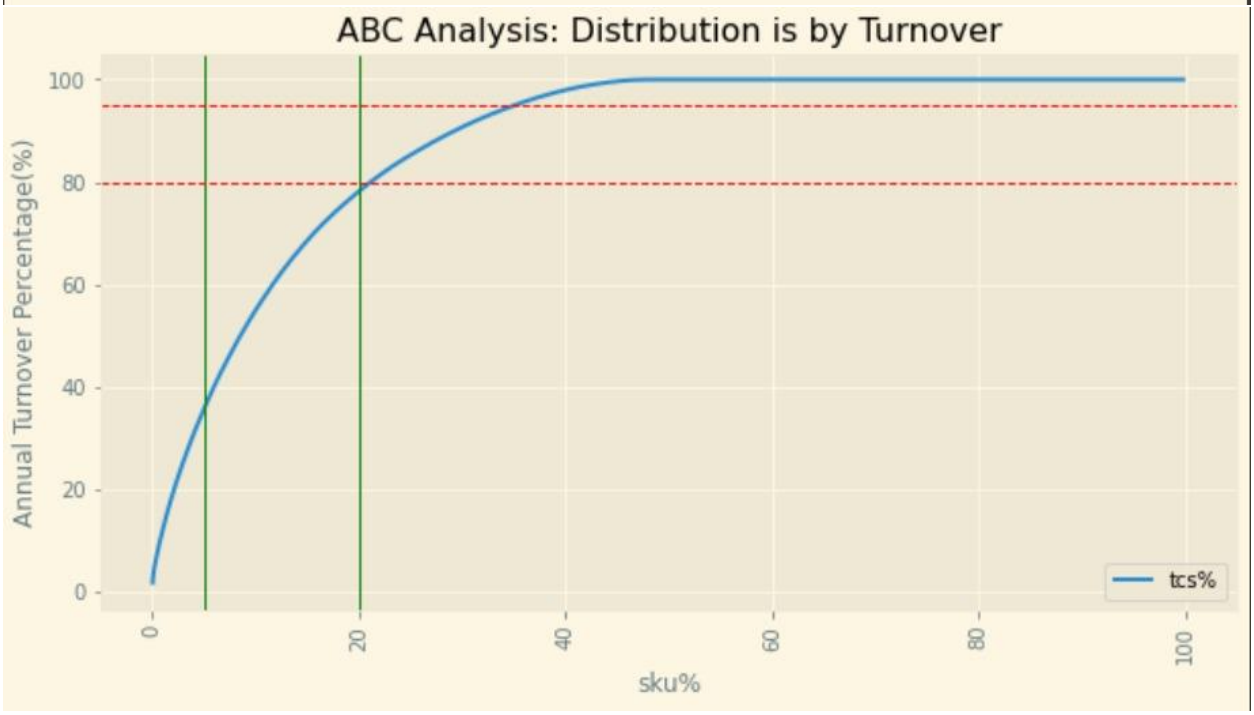

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

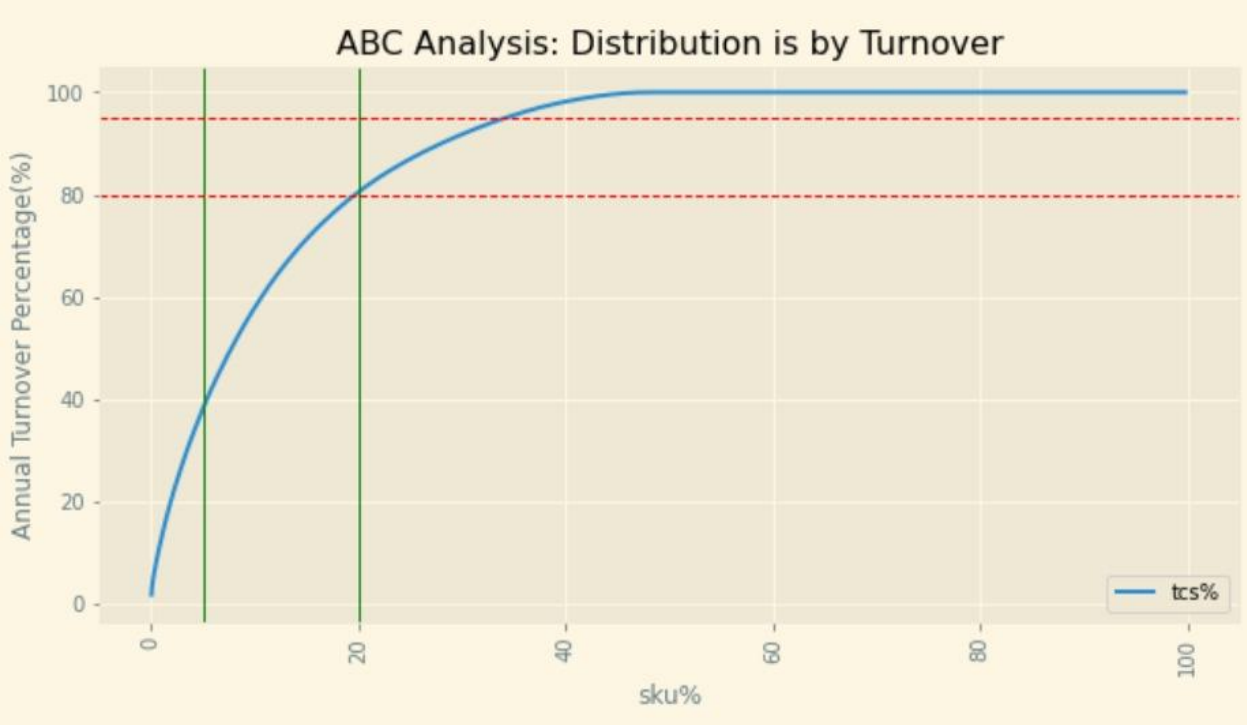

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

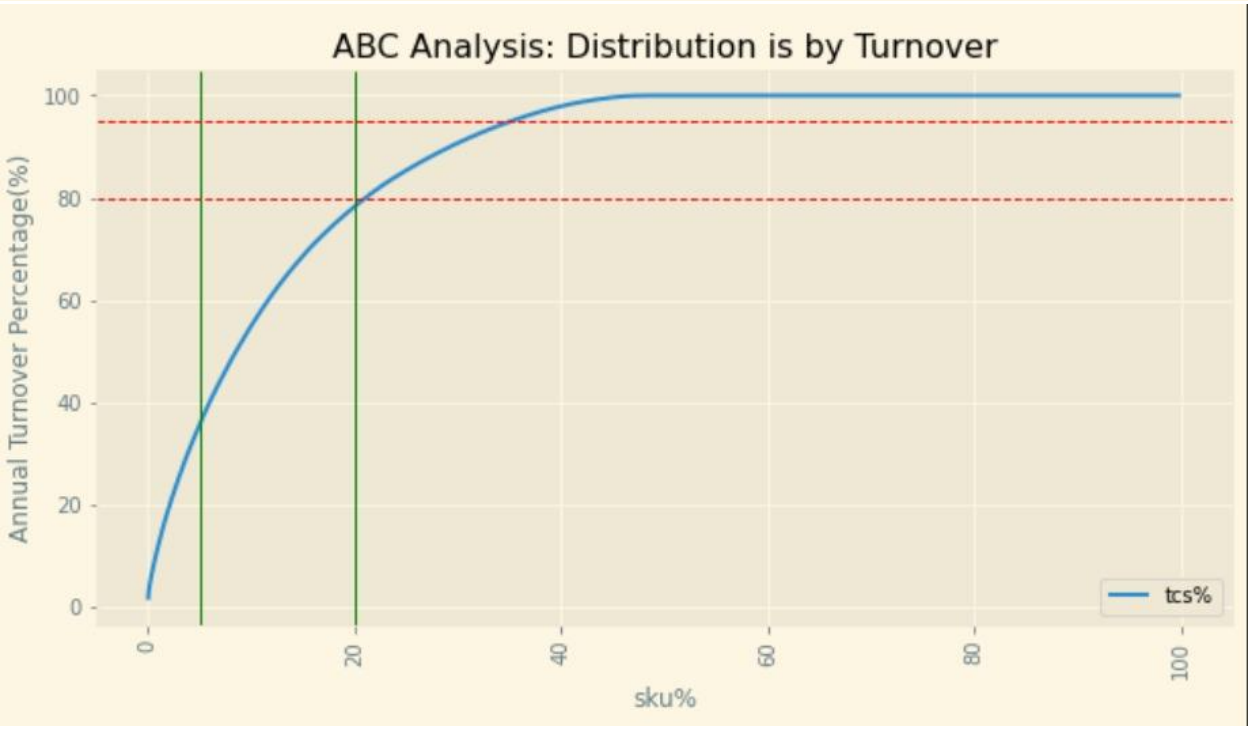

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100


ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

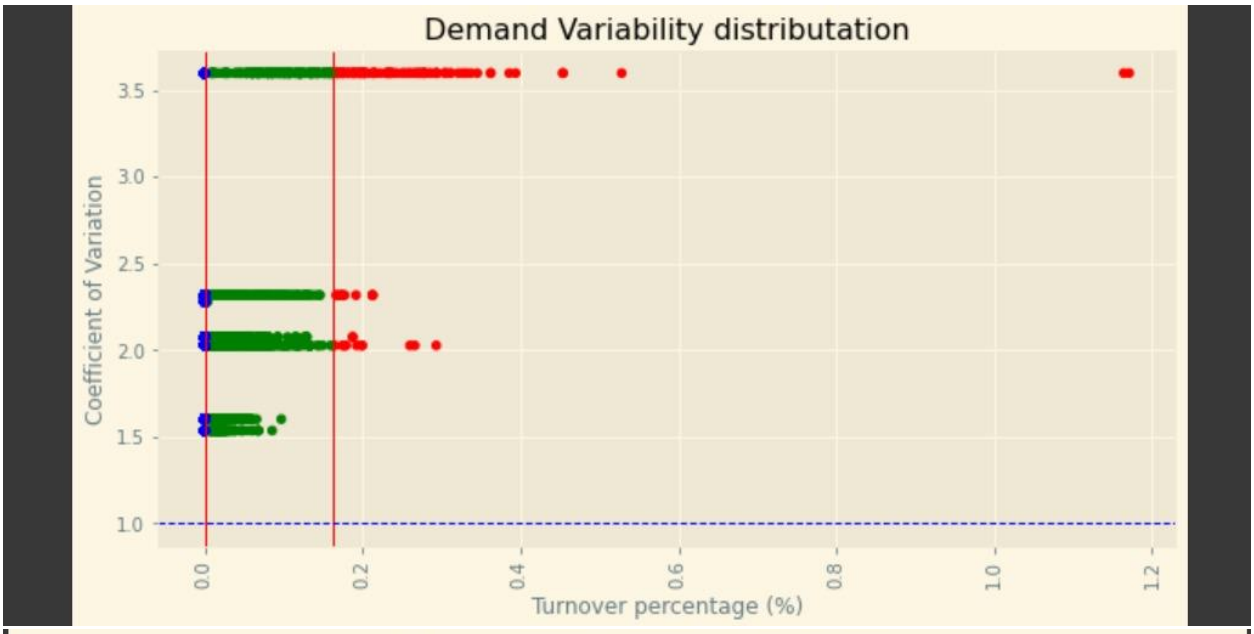

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

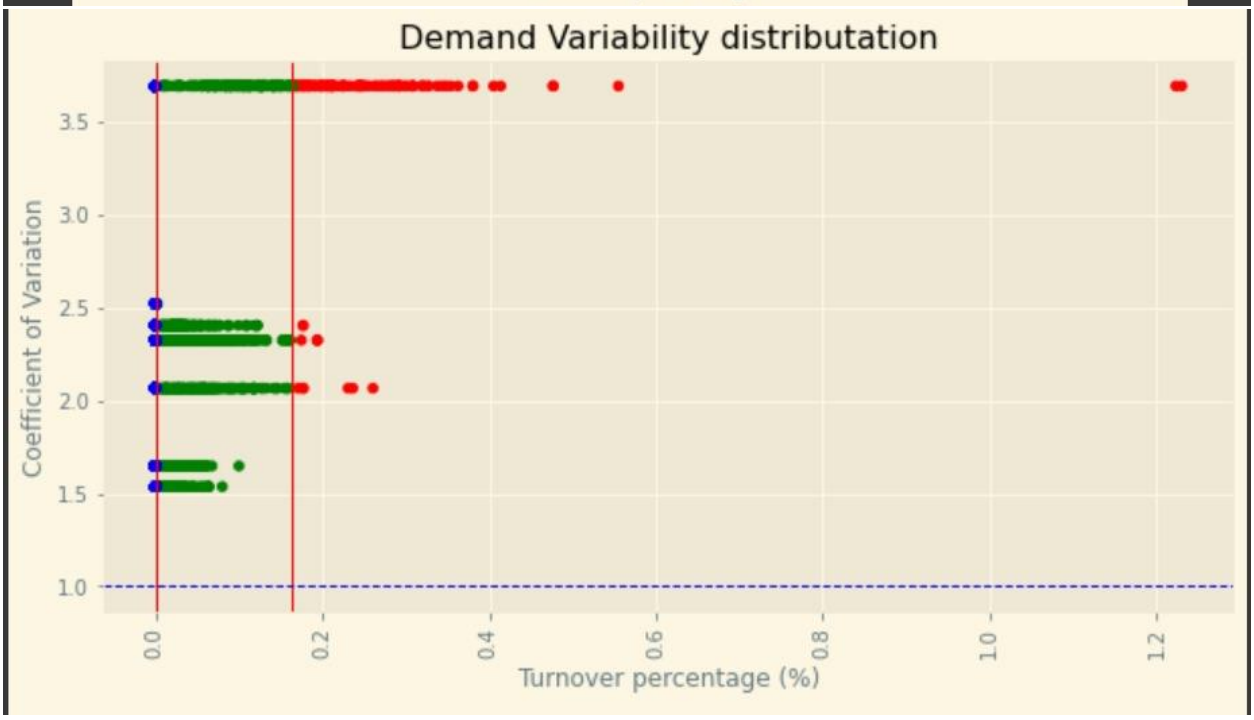

ABC Analysis: Distribution is by Turnover
Annual Turnover Percentage(%)
sku%
tcs%
0
20
40
60
80
100

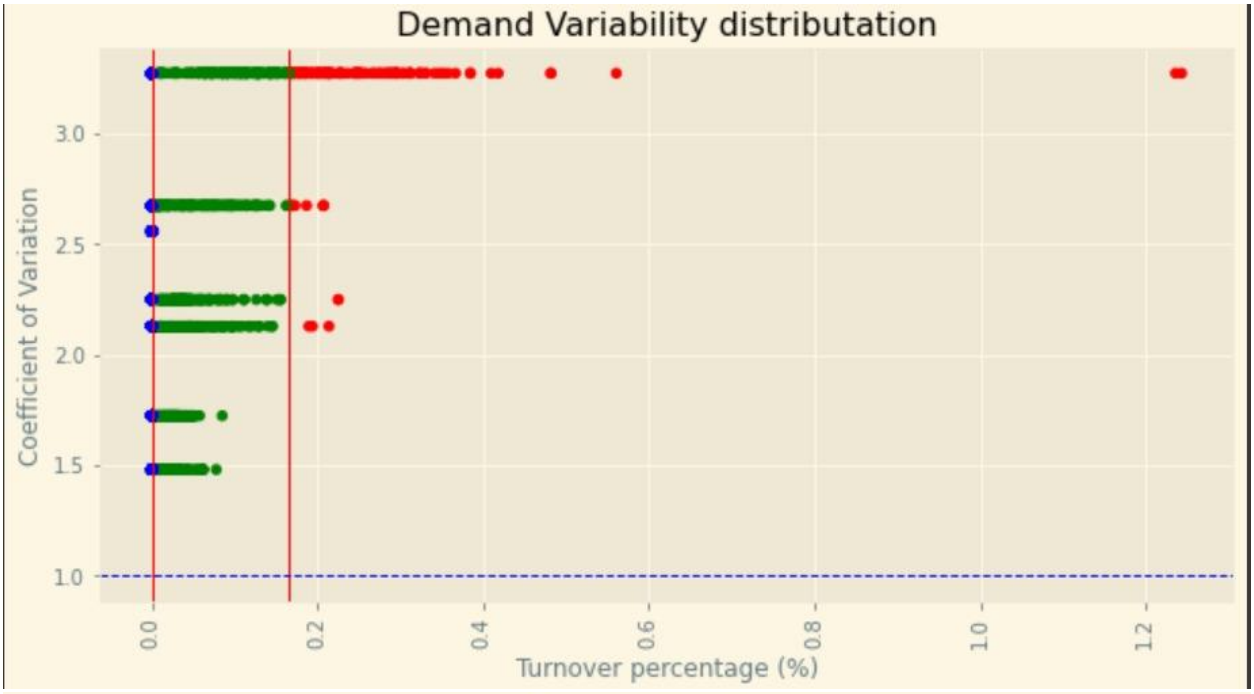


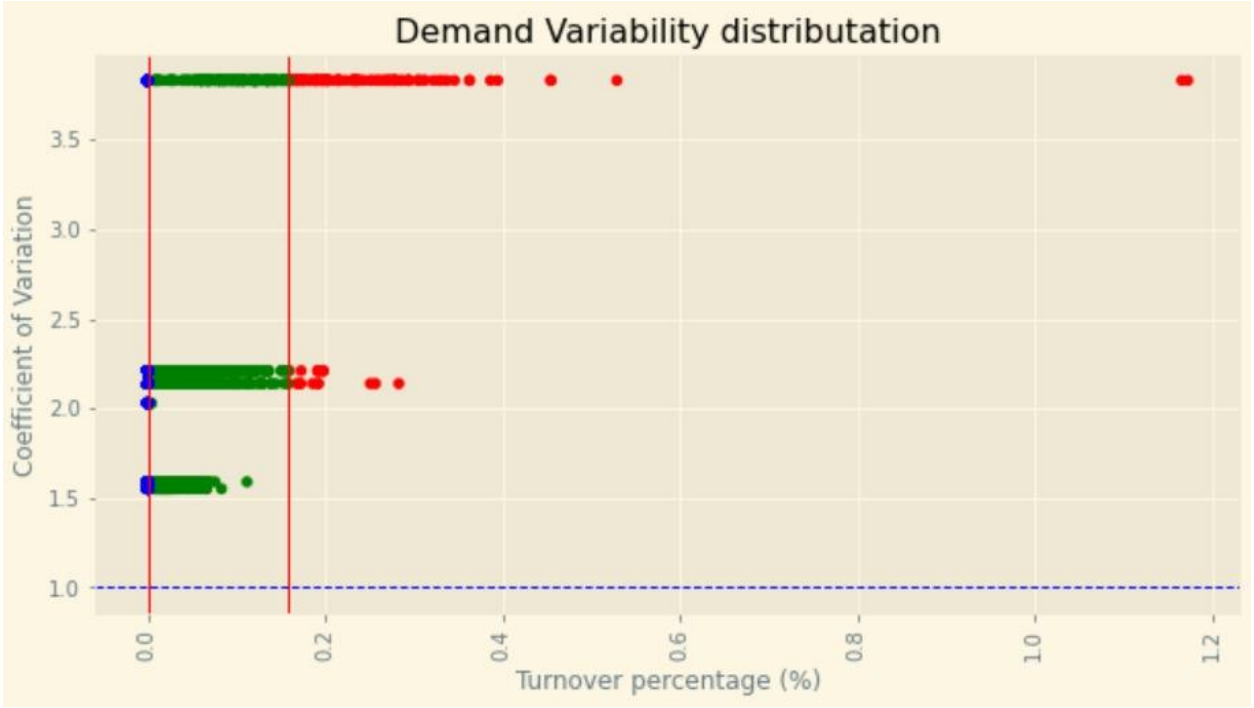


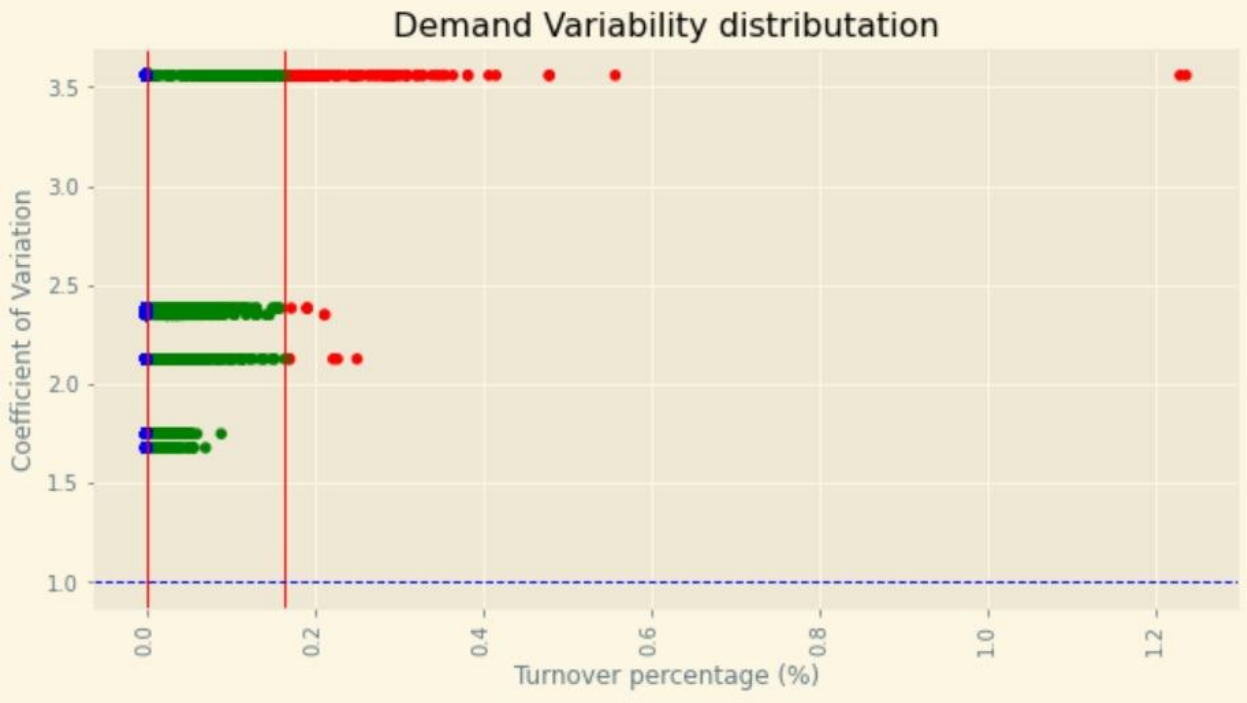
Graph analysis in terms of demand variability distribution (in terms of turnover percentage and coefficient of variation for ABC category) for days which has low sales (took 10 days which has lowest sales) is as below :

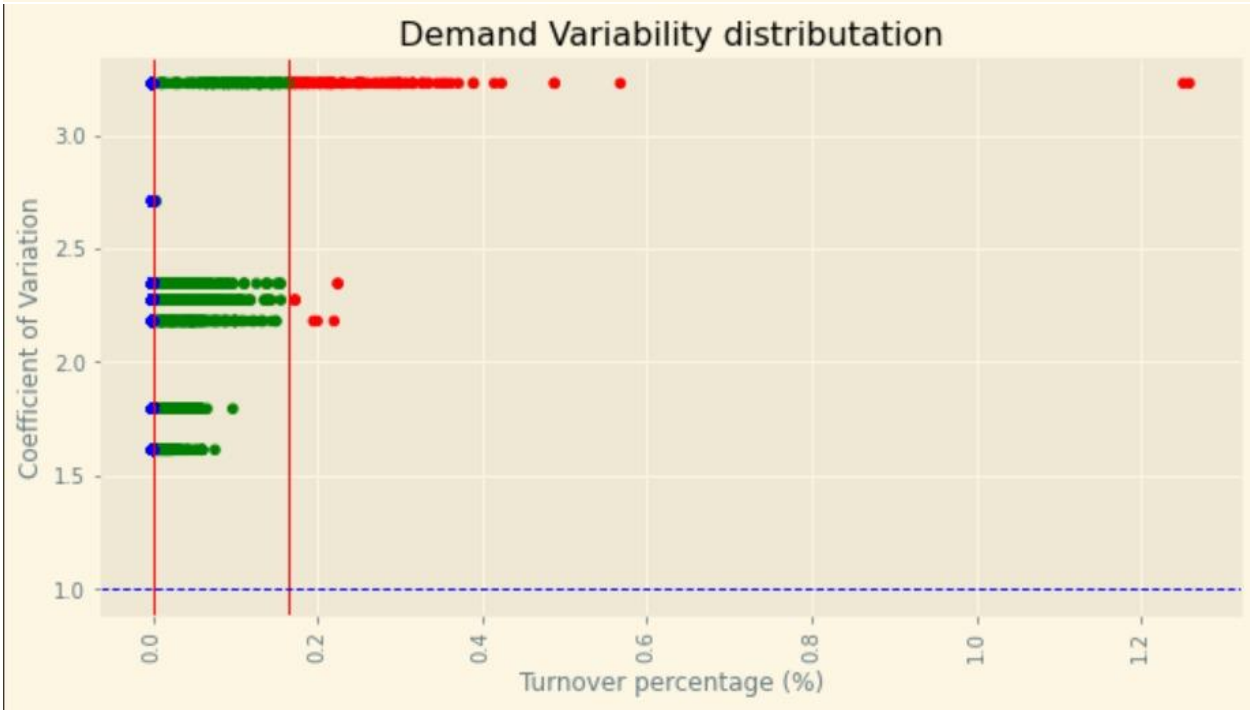

Demand Variability distributation
Coefficient of Variation
Turnover percentage (%)
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

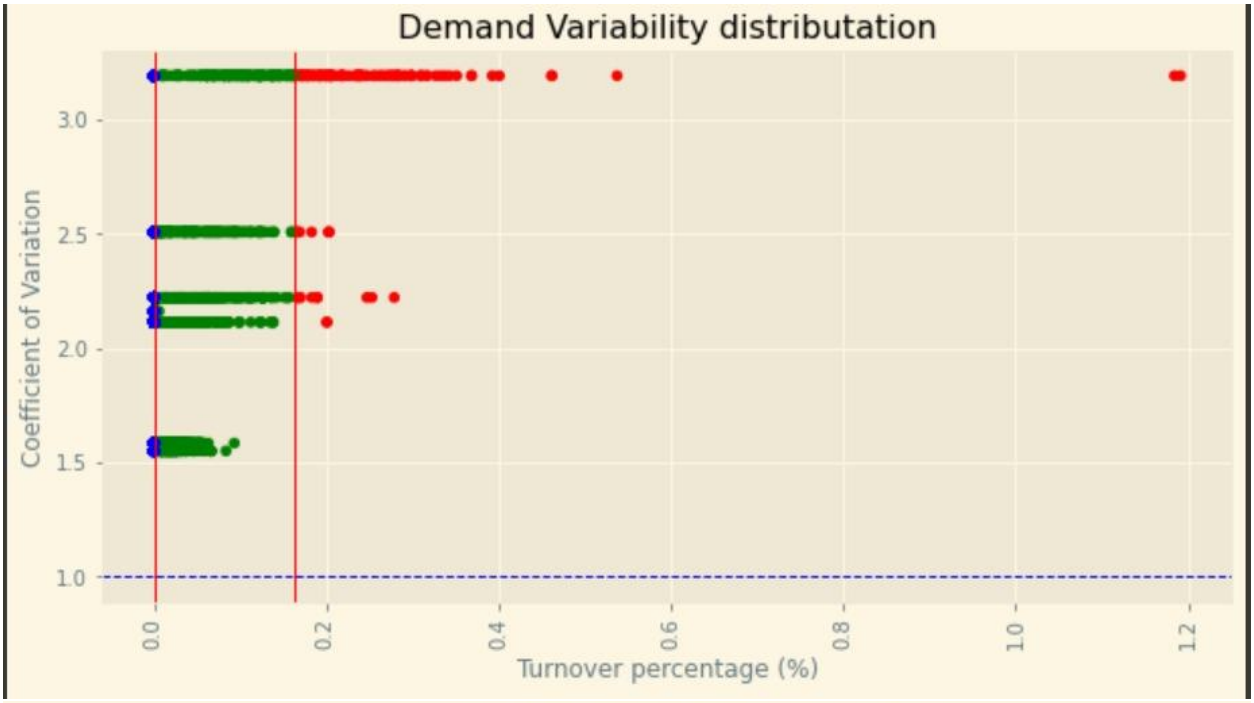

Demand Variability distributation
Coefficient of Variation
Turnover percentage (%)
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2

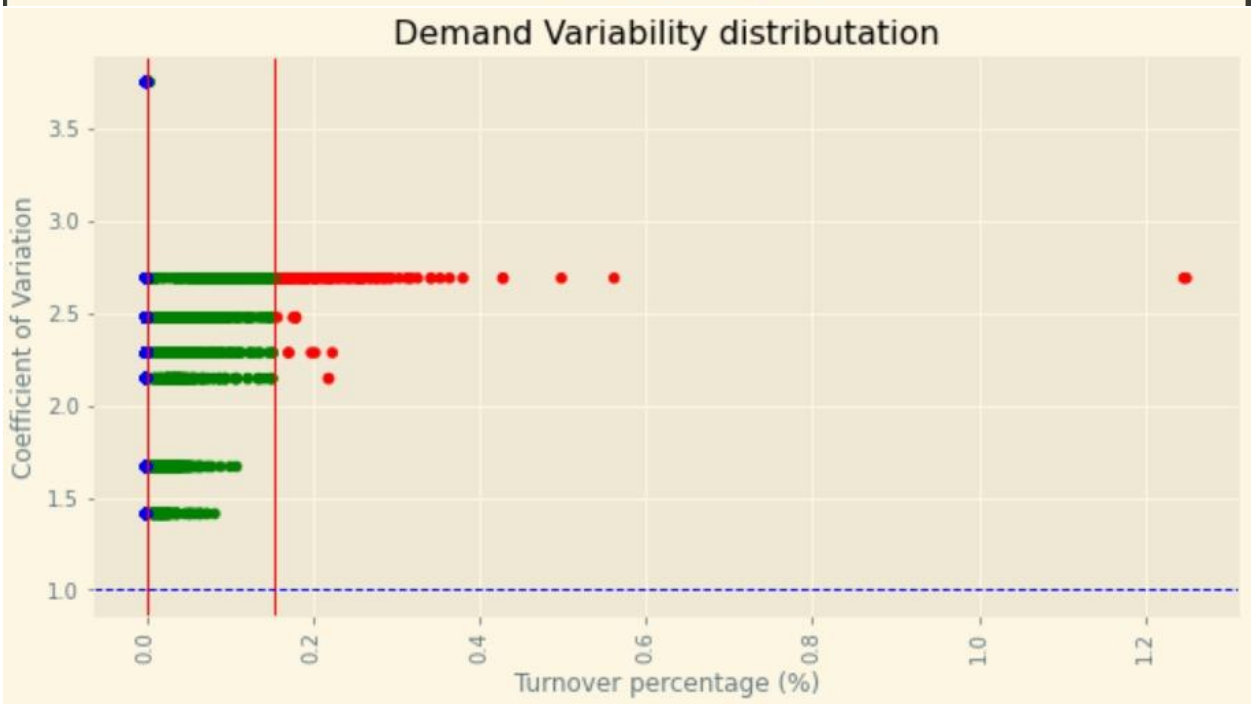

Demand Variability distributation
Coefficient of Variation
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)

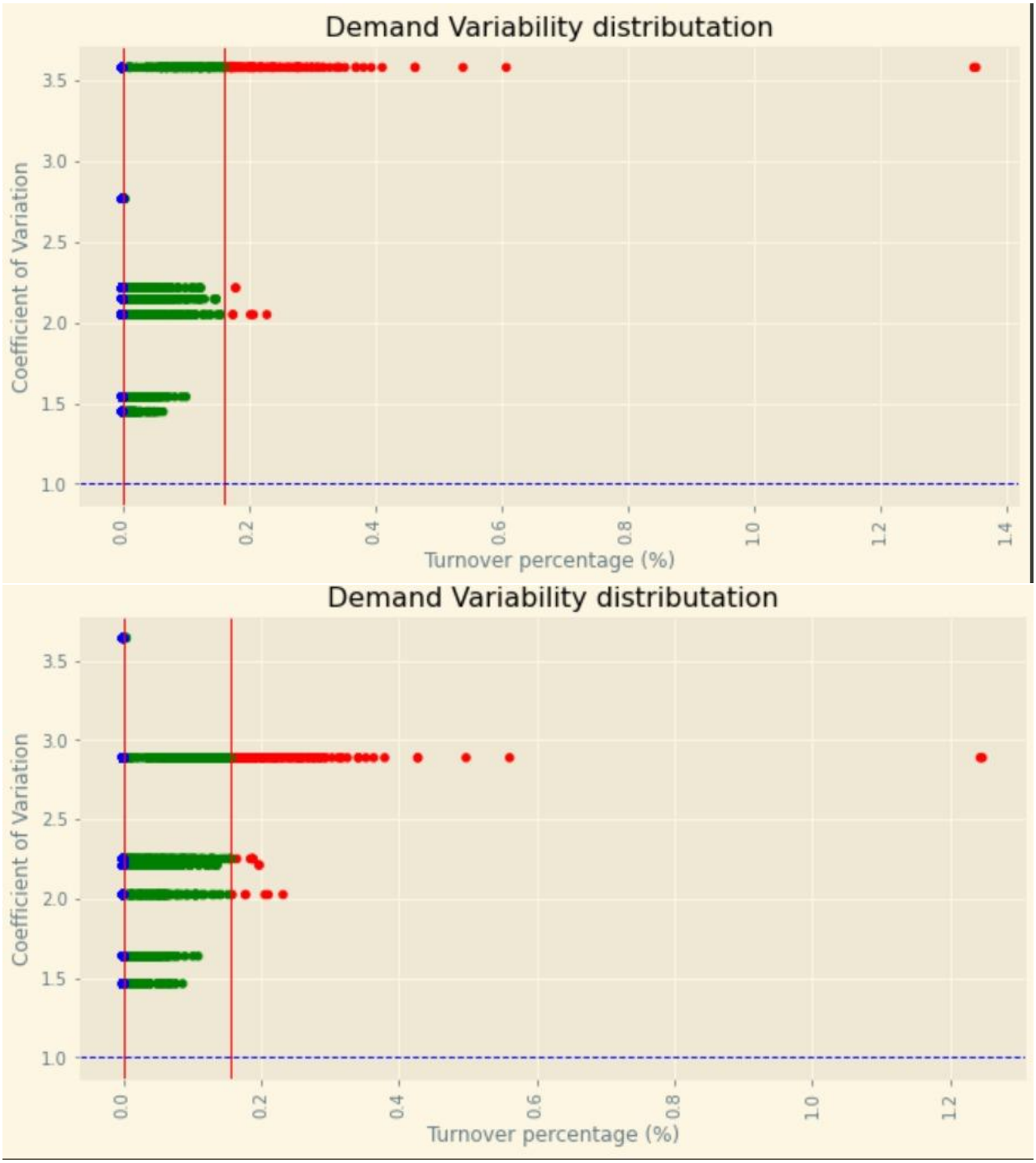

Demand Variability distributation
Coefficient of Variation
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)


Demand Variability distributation
Coefficient of Variation
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)


Demand Variability distributation
Coefficient of Variation
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)


Demand Variability distributation
Coefficient of Variation
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)


Demand Variability distributation
Coefficient of Variation
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Turnover percentage (%)



Based on less demand stability (co efficient of variation)and days(bottom few) which had less sales are taken into consideration to discuss few initiatives and recommendations to improve the sales.

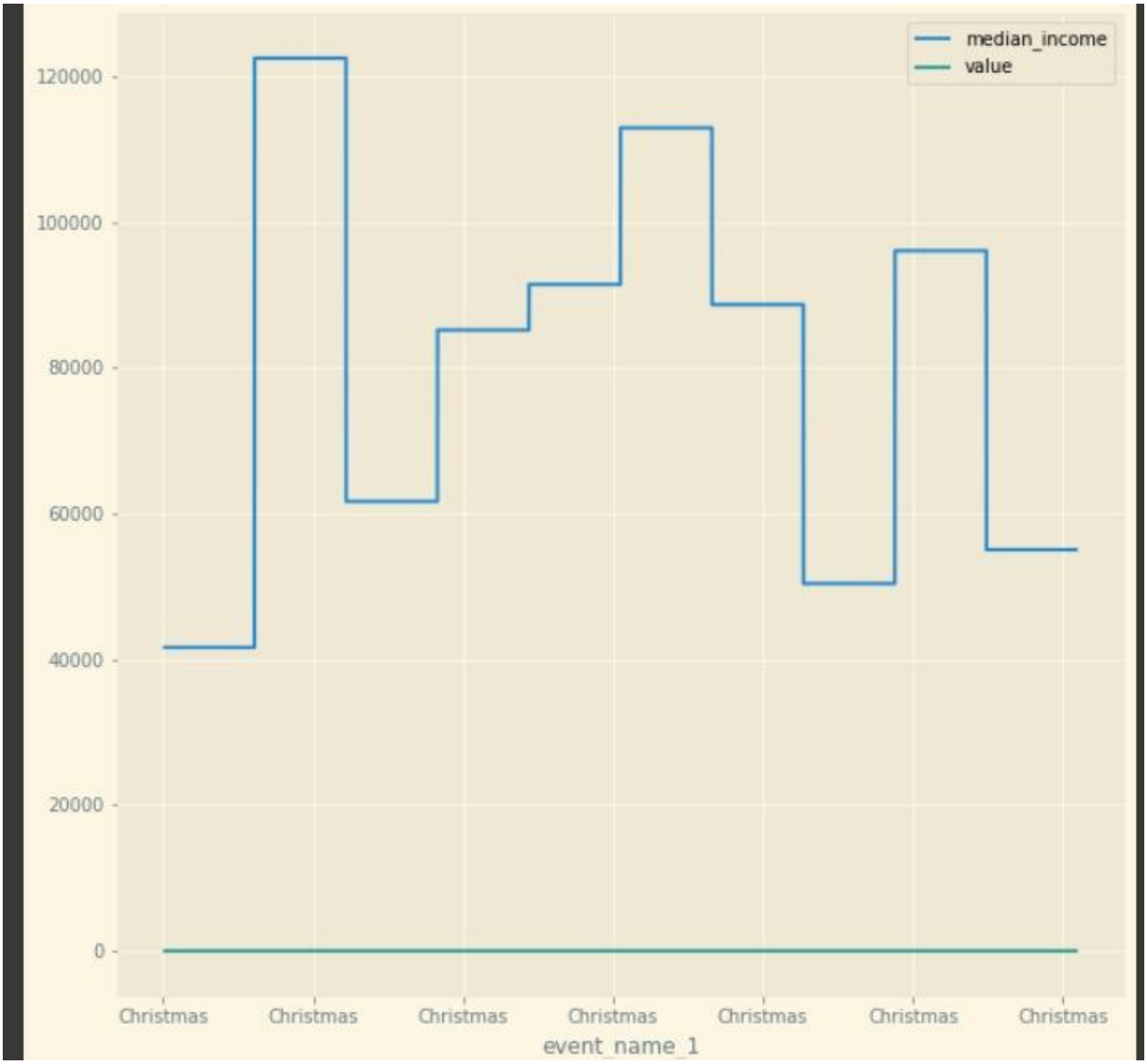
Grouped department for the days considered and performed analysis to give more sale insights for particular departments. Firstly, from ABC analysis it can be inferred that

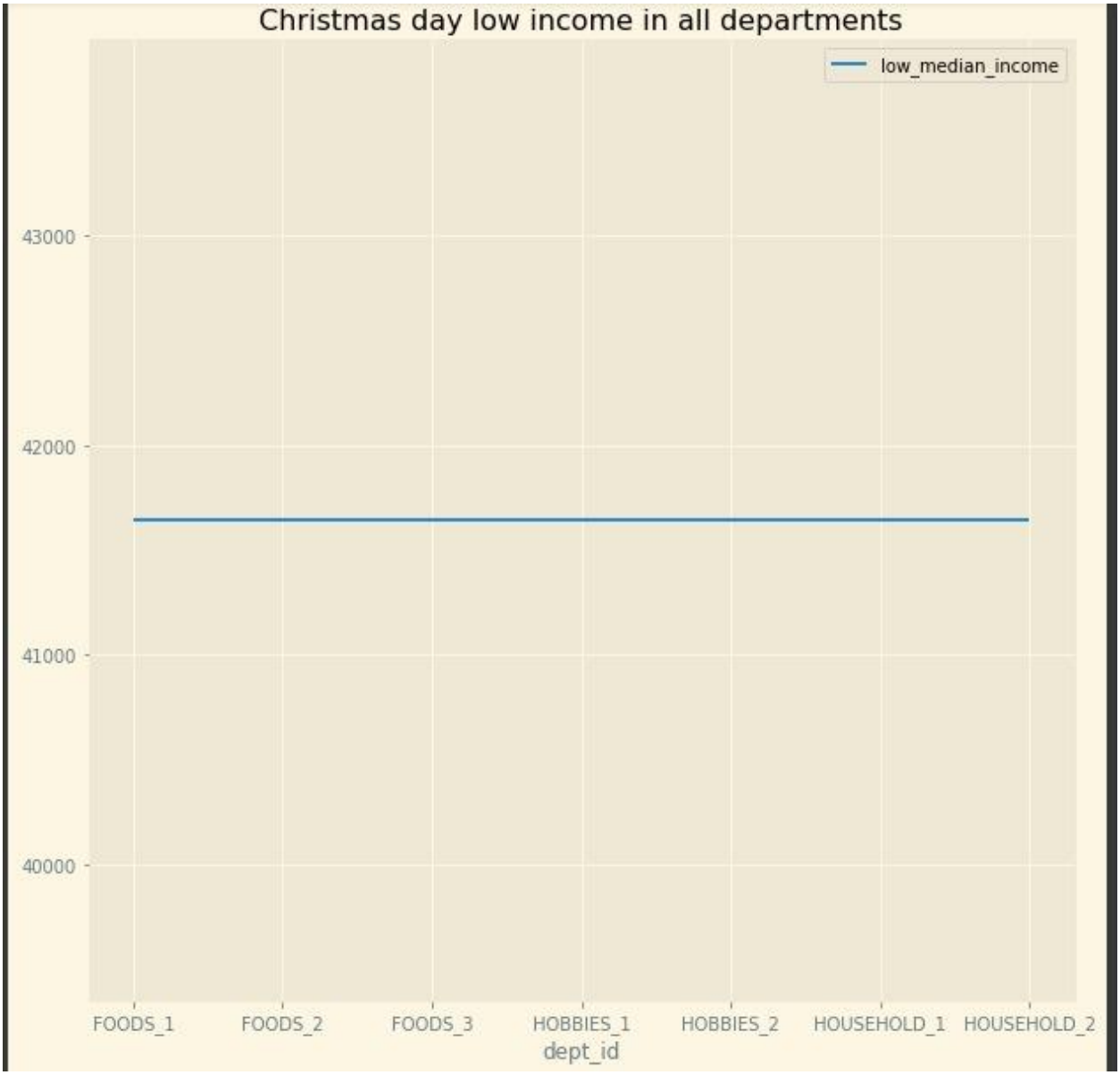
For category C as from analysis section of data-frame and from graphs we can infer that there are almost no or very less sales for days which had less sales so to improve sales under this category departments like household_1, Foods_2, Foods_3 and Hobbies_1 can keep affordable price for their products , some discounts to attract and some useful household and hobbies section as this is the section which has customers who do continuous low budget investment/purchases or section of customers who purchase products which is useful to them which is also low.

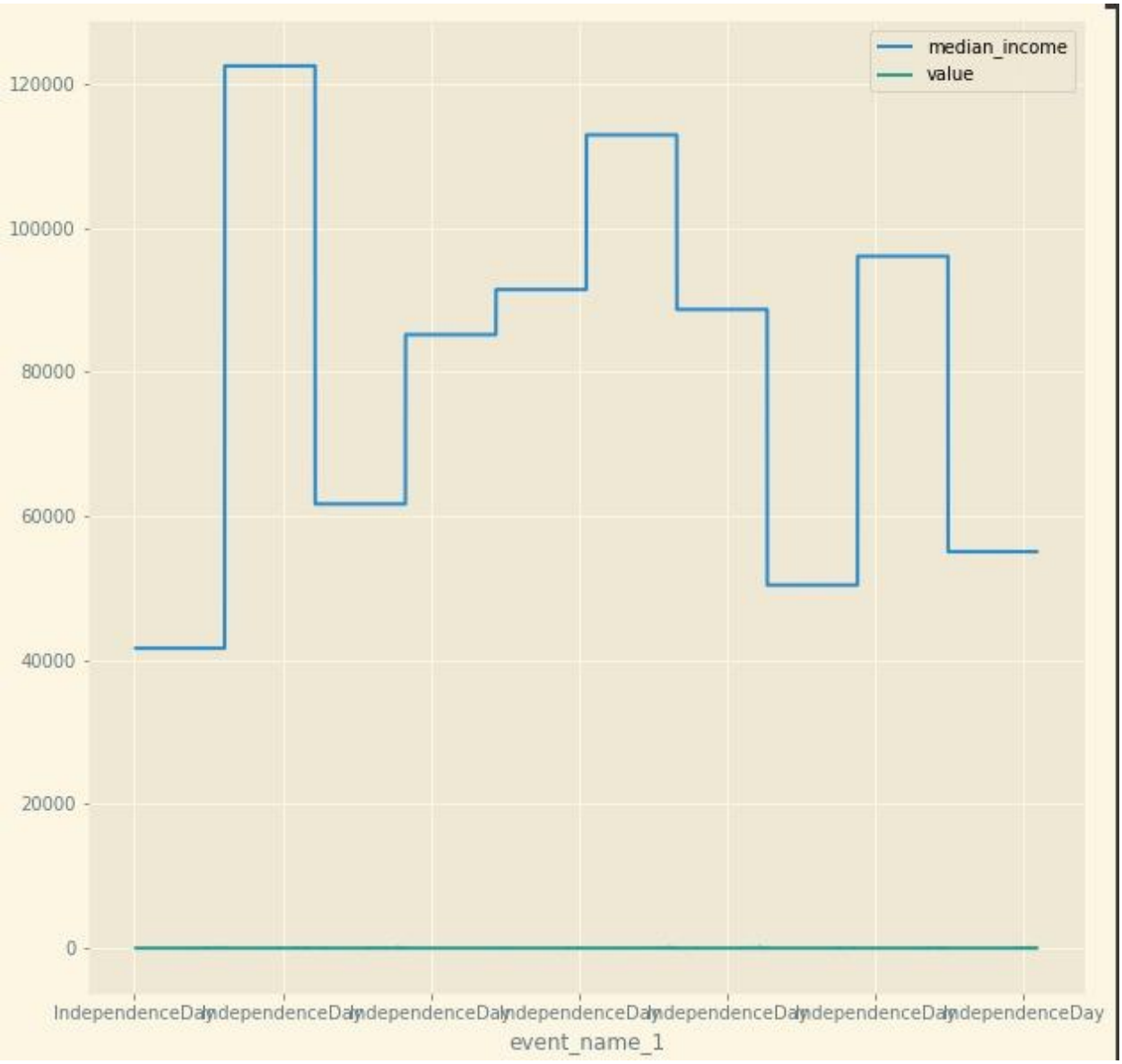
For category B as from analysis section of data-frame and from graphs we can infer that for department Foods_3 sales are good and stable and contribute to the income but Household_1 department is unstable few items in this department has steady income rest are not sold. Category B is helps to boost the business and keeps it steady by either jumping to category A or by dipping to category C so it cannot be neglected. In order to improve the sales of few items which is not doing good one can have like buy one get one offer with increased combined base money, from this customer will feel they are in profit and in terms of sales perspective department will not be in loss of unsold products and will also have marginal profit. Departments can also have increased turn over if they try to give some combo offers with unsold products in it from this no stock will be left un-attended and business will be steady.

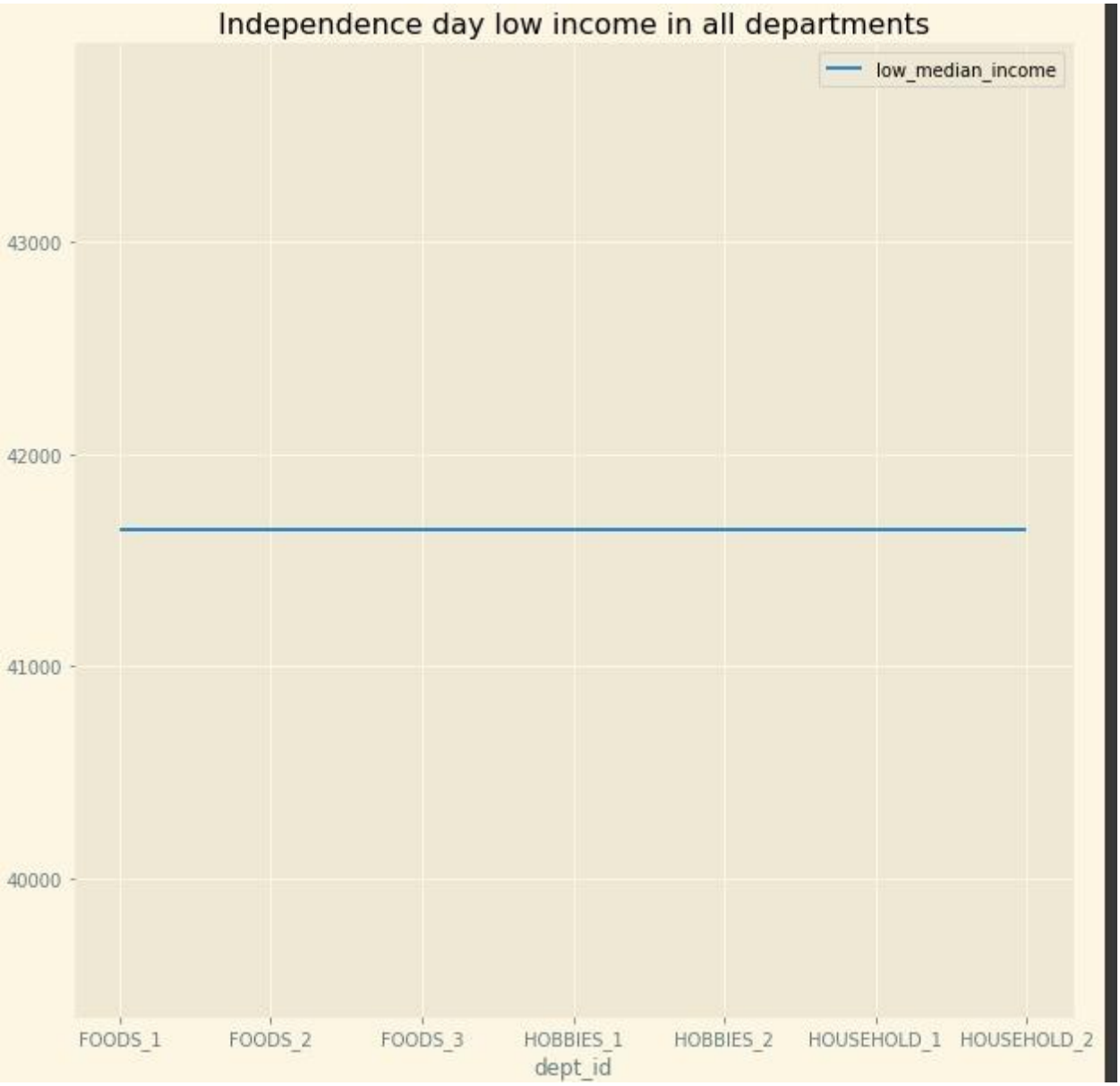
For category C as from analysis section of data-frame and from graphs we can infer that this section is the major contributor to sales. To say Department FOOD in that FOOD_3 item is the major contributor. This section should always be given much importance as here customers will invest lot of money and contribute to companies major profit. So in-order to gear up other categories/departments sales owners can completely nullify the delivery charges or give some useful components free with the order this will build good customer-consumer relationship. As they invest much money here good consumer assistantship and marketing of products also helps in further more boost in sales.

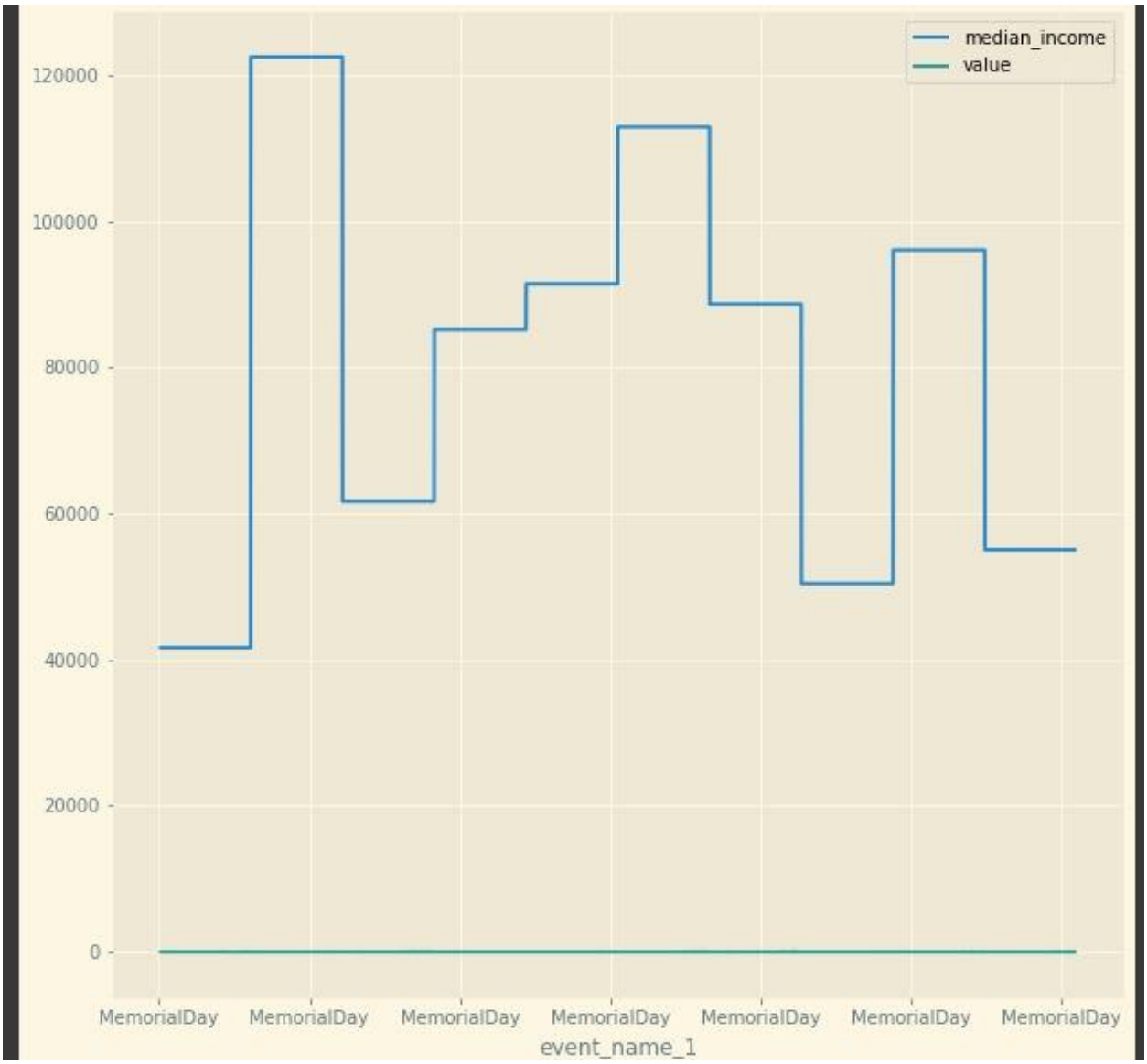
To provide more insights our suggestions to improve sales national holidays type were considered and plotted as below with respect to income and number of sales. Also considered low income median for all department on national holiday types.

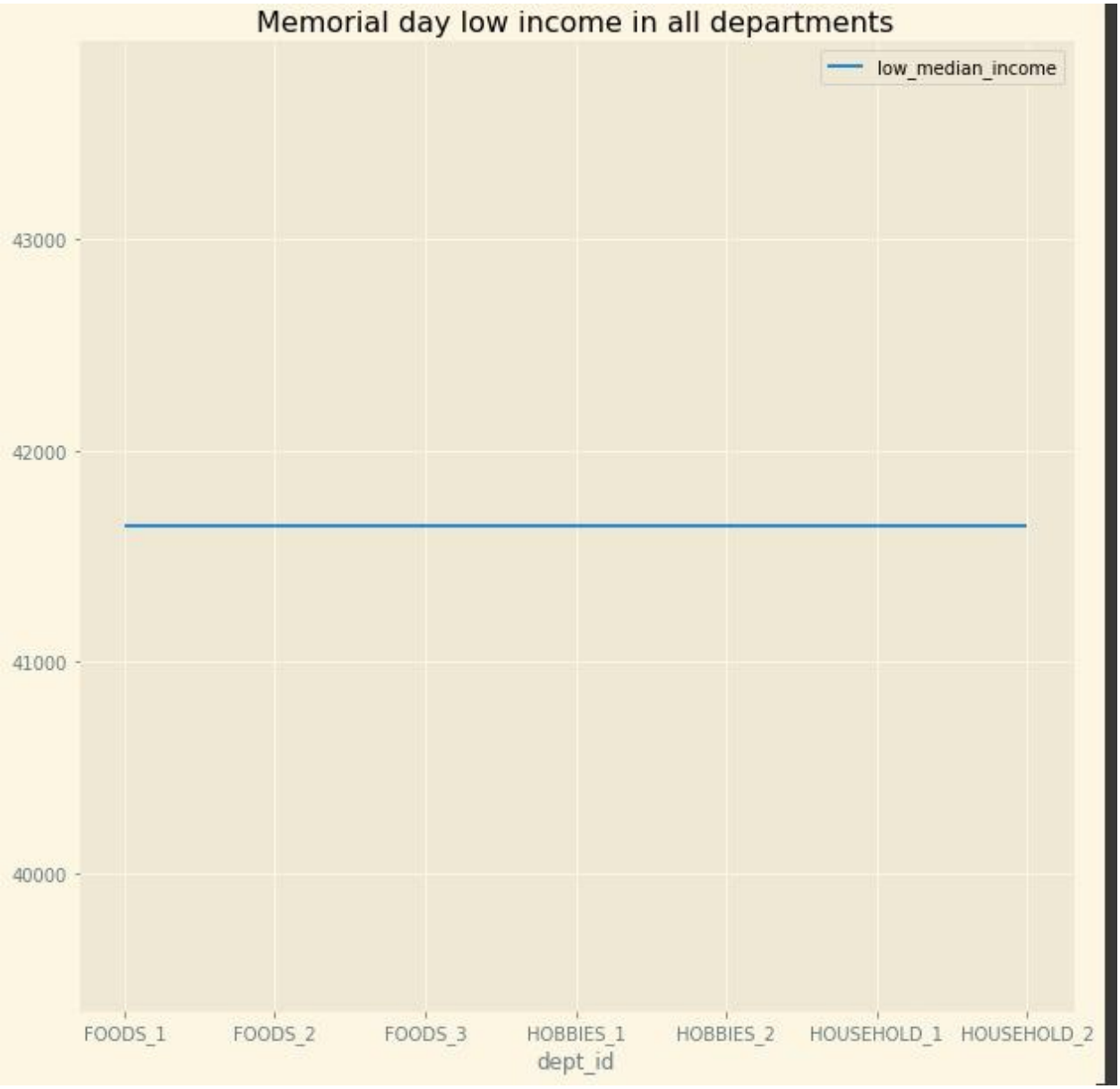

median_income
value
120000
100000
80000
60000
40000
20000
0
Christmas
Christmas
Christmas
Christmas
Christmas
Christmas
Christmas
event_name_1

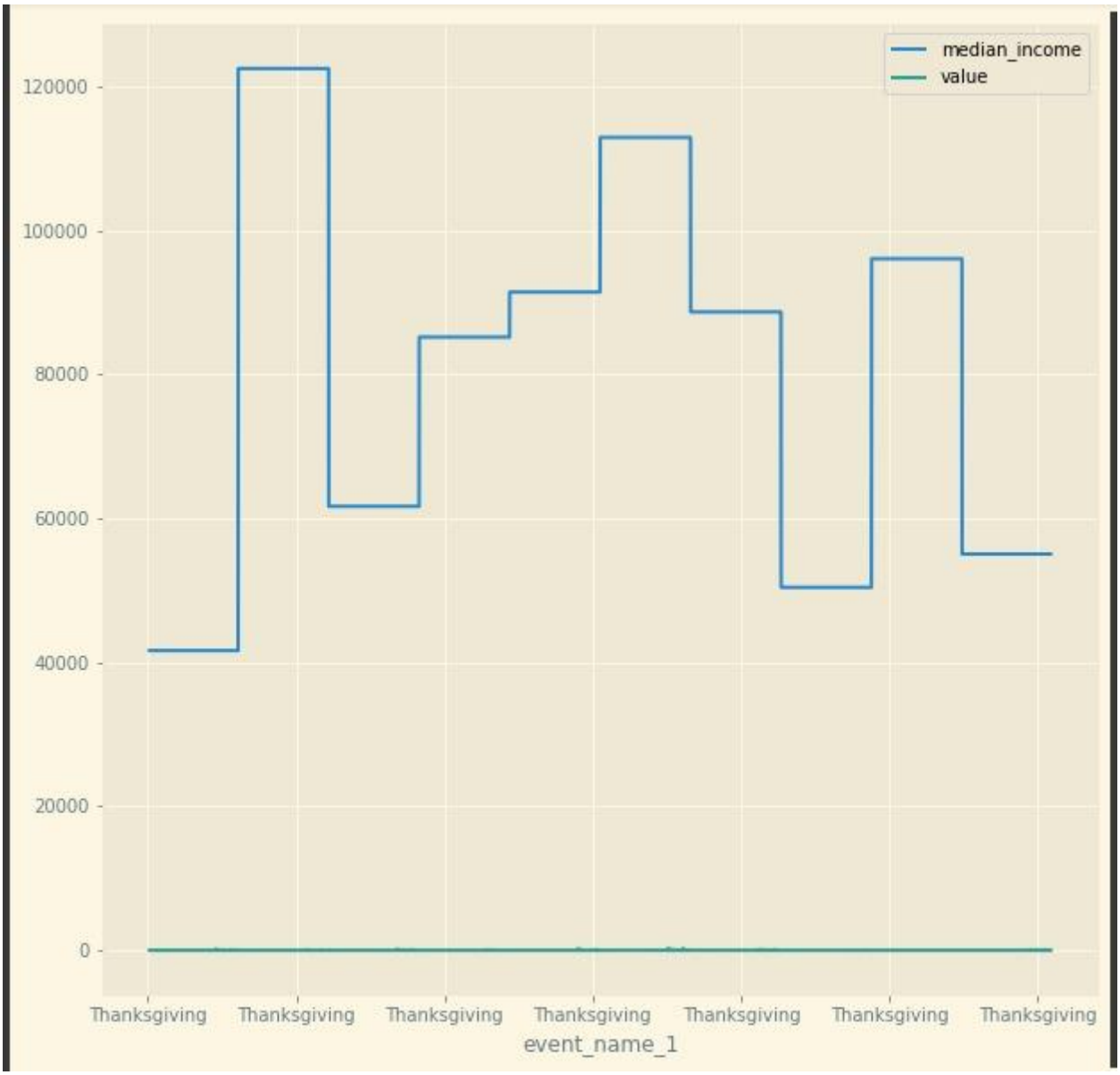

Christmas day low income in all departments
low_median_income
43000
42000
41000
40000
FOODS_1
FOODS_2
FOODS_3
HOBBIES_1
HOBBIES_2
HOUSEHOLD_1
HOUSEHOLD_2
dept_id

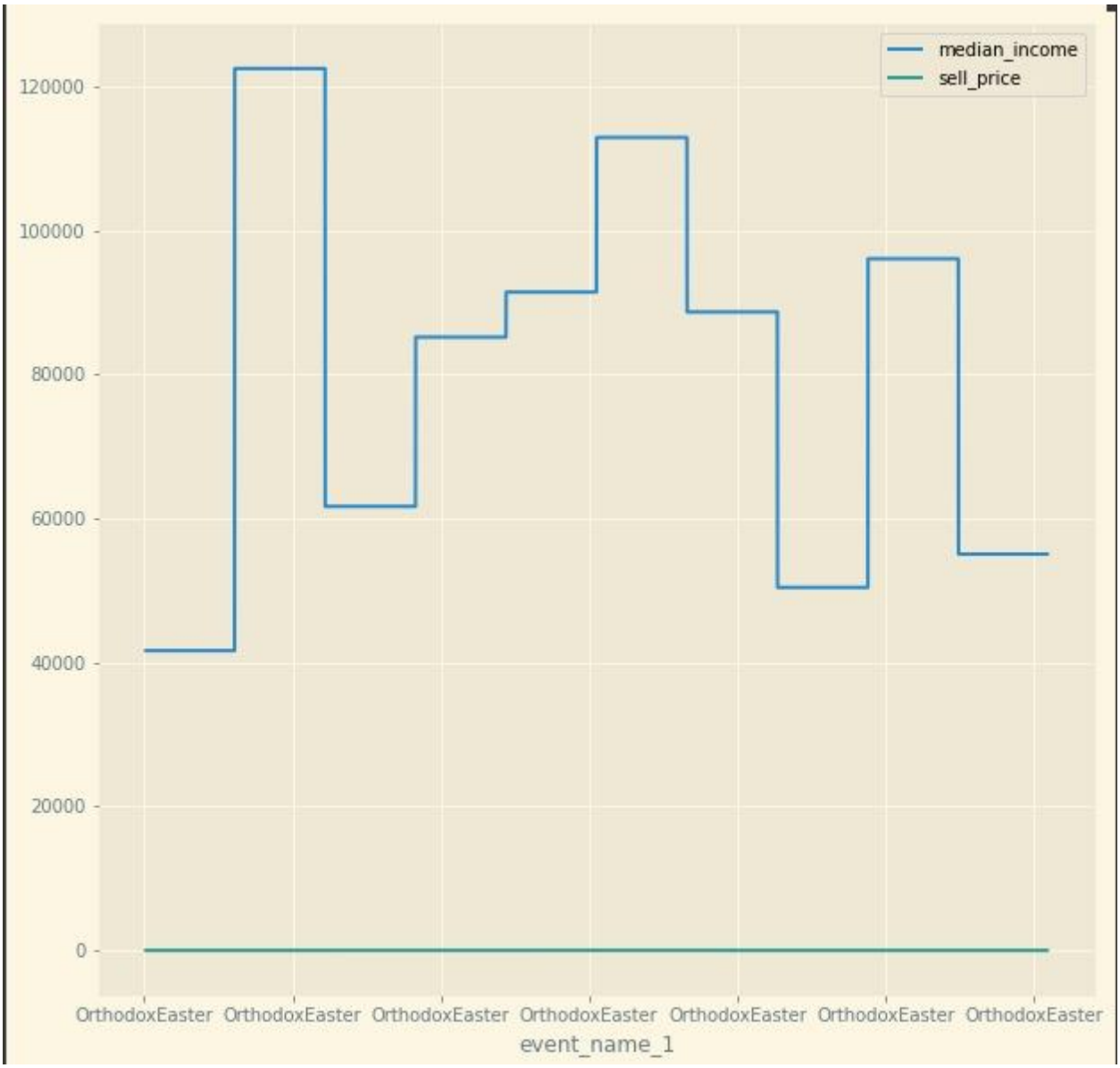

median_income
value
120000
100000
80000
60000
40000
20000
0
IndependenceDay
event_name_1


Independence day low income in all departments
low_median_income
43000
42000
41000
40000
FOODS_1
FOODS_2
FOODS_3
HOBBIES_1
HOBBIES_2
HOUSEHOLD_1
HOUSEHOLD_2
dept_id


median_income
value
120000
100000
80000
60000
40000
20000
0
MemorialDay
MemorialDay
MemorialDay
MemorialDay
MemorialDay
MemorialDay
MemorialDay
event_name_1


Memorial day low income in all departments
low_median_income
43000
42000
41000
40000
FOODS_1
FOODS_2
FOODS_3
HOBBIES_1
HOBBIES_2
HOUSEHOLD_1
HOUSEHOLD_2
dept_id


median_income
value
120000
100000
80000
60000
40000
20000
0
Thanksgiving
Thanksgiving
Thanksgiving
Thanksgiving
Thanksgiving
Thanksgiving
Thanksgiving
event_name_1



From above graphs we can see that for all type-national holidays which has event names like Christmas, Independence Day, Thanksgiving and Easter has low median income for all departments so to improve sales shop owners can give more discount like "special holiday sales discount" or can reduce the delivery charges in this way more customers will tend to buy products in all department and products will not remain unsold.

CONCLUSION

For dataset 1 whose motive is to predict sales we performed data visualization and analysis initially using various graphs to identify key variables we also identified correlation between various columns and visualized the weekly and monthly pattern for top 35% and bottom 35% of department sales. We also investigated the impact of various type of discount on the overall sales by considering top and bottom 30% of best and least performing store respectively. We also identified type of product which are highly impacted by external factors. We then built various ML models like linear regression, ridge regression, XGBoost, Arima, RNN, CNN and Ensemble models. We evaluated their performance using various graphs and performance evaluation metrics.

We merged the data from 3 different files. i.e. calendar.csv, sell_prices.csv and stores.csv into a single data-frame. We down-casted those data files to reduce memory usage. Each record signifies the number of items sold at a particular store on a particular day. It also the shows the selling price of that item on that day. After that we extract weather and median income data from external sources, preprocess them and merge it into our existing data frame. Finally we perform ABC analysis on the existing data-frame and categorize the item in classes A,B,C. Visualizations are provided for all the scenarios wherever required. Before implementing ARIMA model we checked it the data was seasonal, de-trended and made the data seasonality free. However, LSTM proved to be a better fit than ARIMA for this dataset.

We deployed model in stream lit and gave business recommendations or initiatives for dataset 2 using coefficients of variation, ABC analysis and demand variability.

ACKNOWLEDGEMENT

Dr. Allen Bolourchi provided us with a great lot of helpful advice and insight as we worked to solve problems and submit reports for this project. He shared his knowledge with us, and we appreciate that. Throughout the semester, he used his free time to talk with us about how to approach the challenges, share his professional experiences in the MIL/DL/Time Series field, and provide feedback on how to strengthen our results and reporting.
The project is enormous and brand-new to all of us. We are quite grateful for the chance to collaborate with Dr. Bolourchi this semester since it is a fantastic learning opportunity to tackle an industry's actual difficulties.

In the interim, we would want to express our gratitude to Dr. Mahshid Fardadi for providing us with the chance to address this real-world business issue in her class and for connecting us with Dr. Bolourchi. We appreciate her and Rahul Deo Vishwakarma’s assistance and support throughout this semester's project and reporting processes.