CECS 551 Artificial Intelligence

Artificial Intelligence (AI) approach in Retail Market Analysis and Growth

---

Student
Mudra Chaudhary
Santosh
Antony
Misha
Vruddhi
Jeremy

Supervisors
Dr. Mahshid Fardadi
Dr. Allen Bolourchi

Fall 2022

Contents

Abstract

International Retail Business is a hypothetical store which has multiple products sold across various geographical locations. There are inventory data of two dataset of around more than 30 stores

Department stores have uncountable product and money transactions every day. Because of their rapid transaction rates, keeping a balance between the inventory and customer demand is the most important decision for the managers. Therefore, making an accurate sales prediction for different prod- ucts becomes an essential need for stores to optimize their profits. Most of the existing sales predictions only depend on extrapolating the statistical trend. The previous studies on market sales prediction require a lot of extra informa- tion like customer and product analysis. A more simplified model is needed by the department store to predict the product sales based on only the historical sales record. The new emerging machine learning methods enable us to make more accurate predictions like that

We analyze the data for stores and provide few initiative which help maximize the overall sales and profit. This document summarizes our finding.

1 Introduction

In a team of six, three members mainly focus on analyzing the data for dataset 01: the task is to predict the department-wide sales for each store and and three persons mainly focuses on the tasks related to 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.Results are communicated between team members to help them with their individual analysis and ensure consistency.

1.1 Problem Statement

The final project is designed to implement three-week sprints of the scrum process, mimicking a real tech company machine learning or software development team environment. Please see the timetable below for more details. It is suggested that one person plays the role of the scrum master to coordinate the communications between team members and ensure on-time delivery at the end of each sprint.

1.2 Dataset Description

1.2.1 Dataset 01

Features	Description
store	The number of stores date - MMDDYYY format
temperature	Temperature in Fahrenheit
gas price	Price per gallon in
dollar discount promotional discounts	
discount clearance	
discounts discount damaged good	
discounts discount competitive	
discounts discount employee	
discounts	
IsHoliday	Yes or No
CPI	The Consumer Price Index (CPI) Unemployment - Unemployment rate in the region where store is present

Table 1: store 01-10.csv

Features	Description
store	The number of stores
type	Stores segregated into three types, i.e., A, B, and C
size	Size of the store

Table 2: stores.csv

Features	Description
store	The number of stores
dept	Department ID
date	MM/DD/YYYY
IsHoliday	Yes or No

Table 3: test.csv

Features	Description
store	The number of stores
dept	Department ID
date	MM/DD/YYYY
weekly sales	Sales per week
IsHoliday	Yes or No

Table 4: train.csv

1.2.2 Dataset 02

Features	Description
date	The date in a “y-m-d” format
wm_ yr_ wk	The id of the week the date belongs to
weekday	The type of the day (Saturday, Sunday, . . . , Friday)
wday	The id of the weekday, starting from Saturday
month	The month of the date
year	The year of the date
event_ name_ 1	If the date includes an event, the name of this event
event_ name_ 1	If the date includes an event, the type of this event
event_ name_ 2	If the date includes a second event, the name of this
event event_ name_ 2	If the date includes a second event, the type of this event
snap_ CA, snap_ TX, and snap_ WI	A binary variable (0 or 1) indicating whether the stores of CA, TX or WI allow SNAP purchases on the examined date. 1 indicates that SNAP purchases are allowed.

Table 5: calendar.csv

Features	Description
store_ id	The id of the store where the product is sold
item_ id	The id of the product
wm_ yr_ wk	The id of the week
sell_ price	The price of the product for the given week/store. Price is provided per week (average across seven days). If not available, this means that the product was not sold during the examined week

Table 6: sell_ strain.csv

Features	Description
item_ id	The id of the product
dept_ id	The id of the department the product belongs to
cat_ id	The id of the department the product belongs to
store_ id	The id of the store where the product is sold
state_ id	The State where the store is located
d_ 1, d_ 2, . . . , d_ i, . . . d_ 1941	The number of products sold at day i, starting from 2011-01-29

Table 7: sale_ strain.csv

1.3 Proposed Workflow

We perform the analysis in three phases.

1. Data visualization.
2. Create a machine leaning model for sales prediction.
3. Model deployment and business recommendation.

1.3.1 Phase 1

Each phase is covered in individual chapters. The first phase tires to understand the data distribution, relationship among them using correlation matrix and SHAP feature interactions for Dataset 01.

1. First level item Identify the key variables for the model using correlation plots, heatmaps, histograms, feature importance (SHAP).
2. First level item For the first 10 stores visualize the weekly and monthly sales patterns for top 35 % of the department sales.
   (a) Second level item Identify the best department across the first ten stores. (hint – department 38)]Identify the best department across the first ten stores. (hint – department 38)

3. 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.

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

   (a) Second level item Which type of discount is helpful in increasing the sales? Consider the top 30% of the best performing stores (sales per 1000 square feet).

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

5. Identify the departments which are highly impacted by external factors: "tem- perature", "gas price", and "holiday". Is there any correlation between overall sales and holidays?

Analysis of DataSet 02 :

1. Use Tableau to visualize the dataset 02.

2. Publish the Tableau dashboard on public server.

1.3.2 Phase 2

The purpose of second phase is to predict sales of the store in a week. The data consists of 45 stores including store information and monthly sales.As in dataset, size and time related data are given as feature, so analyze if sales are impacted by external factors, for example, how inclusion of holidays in a week soars the sales in store? .

1. Dataset 01

   (a) 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. There are external variables such as gas price, holidays, unemployment, and temperature for the given dataset. Evaluate the impact of these external variables on the accuracy of the model (do they help improve the accuracy?). Plot the relevant graphs.

      i. Begin with Linear regression model to forecast the weekly sales using the given features. Hints: Feature selection, feature engineering (new features, if any), PCA.

      ii. Create the following machine learning models: ARIMA, Ridge Re- gression and Boosting to predict sales.

      iii. Communicate the model performance metrics and tabulate the com- parison in report. Support your finding by validating the model accuracy across various stores (first 10 stores and $store_1 1_3 5$).

(b) In stores.csv, we have a feature “type”, i.e., three types of stores – A (Super center), B (Discount center), and C (Neighborhood markets). Now, based on the given features in the dataset, can we predict the store type? Please consider all only the first 10 stores for this problem statement.

i. Consider the problem statement as multi-label classification problem. Use the below classification algorithms and perform hyper-parameter tuning for the Deep Learning models.
   A. Ensemble models (3 statistical methods)
   B. Recurrent neural network (RNN)
   C. Convolutional Neural Networks (CNN)

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

2. Dataset 02

(a) Perform data preprocessing and exploratory data analysis. Hints: Does seasonality and trend exist in the dataset? Drop any column? Extra
credit: Perform down-casting (shrink dataset size)

(b) Feature engineering: create two new features using the information provided in Table 1.

i. weather data
ii. median income

(c) Use the machine learning algorithms below to model n-step ahead fore- casting (n = 10).

Note: First create a model without using any external features, and then create a model with the external features.

i. Begin with ARIMA and compare the RMSE values for each category.
ii. Long short-term memory (LSTM) – Perform hyper-parameter to im- prove the model.
iii. Tabulate the performance metrics of each model.

1.3.3 Phase 3

One of the goal is to provide insight and recommend few initiatives based on the sales predictions model from the previous phases and provide minimum of three inventory optimization initiatives and examine them on the data. Also deploy the model on a dashboard for visualization.

(a) Deploy the machine learning model using Streamlit for dataset 02 (ARIMA)

(b) Product segmentation based on demand variability (ABC Analysis). The references that are driving most of the sales.

- Class A: Very Fast Movers: top 5%
- Class B: The following 15% of fast movers
- Class C: The remaining 80% of very slow movers Task

i. Use first year data of Household category to create ABC Analysis and interpret the graph.

ii. How stable is the customers' demand? (Coefficient of Variation) To understand which products will bring planning and distribution challlenges, compute the coefficient of variation of the yearly distribution of sales of each reference.

iii. Discuss a few initiatives and recommendations for improving the re- tail business for dataset 02.
Hints:

A. Provide more discount for locations which have low median in- come based on specific event date, for example, Christmas

B. Potential interpretation based on ABC Analysis and Demand variability.

2 Phase 1

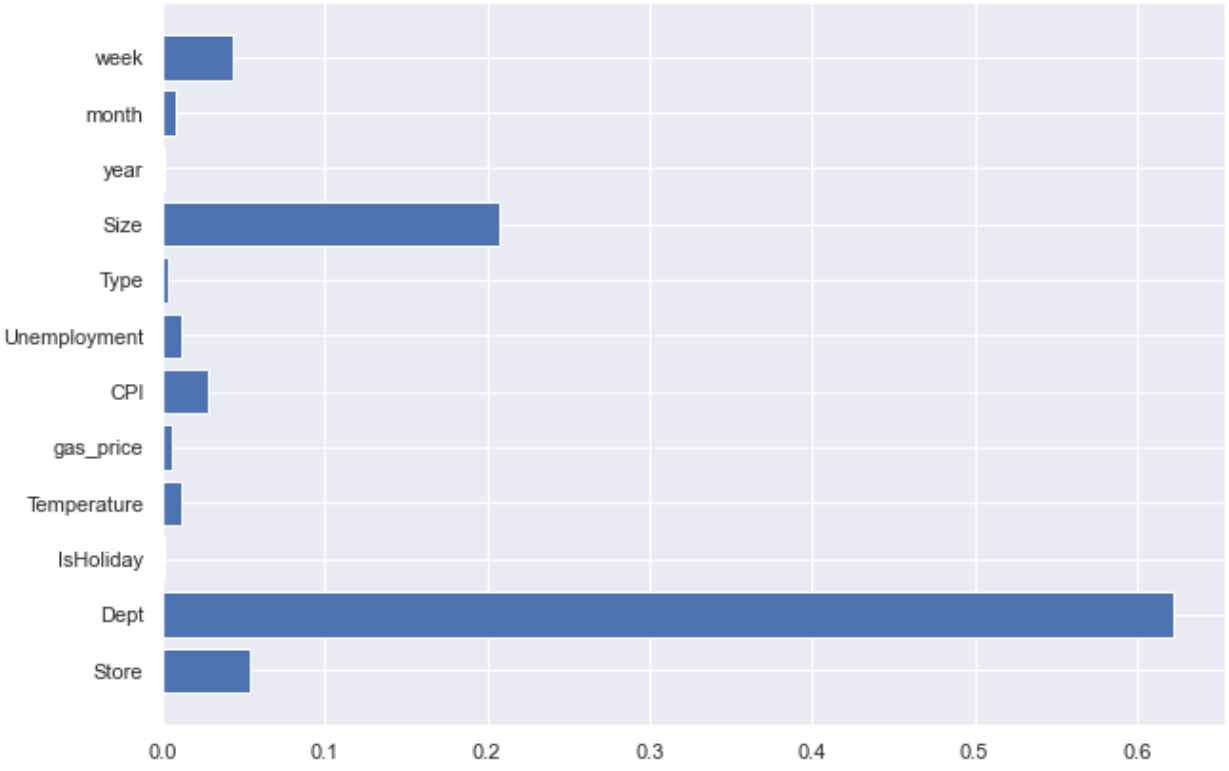
A snapshot of the data and understand the current business from the given dataset using Tableau for the visualization for the second dataset. The goal is to create a Tableau dashboard and publish the results to present the infor- mation to an upper higher-level management of an organization.

2.1 Dataset 01

The exploratory data analysis using Python for dataset 01 with the right data visualization method for specific problem statements (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, feature importance (SHAP).

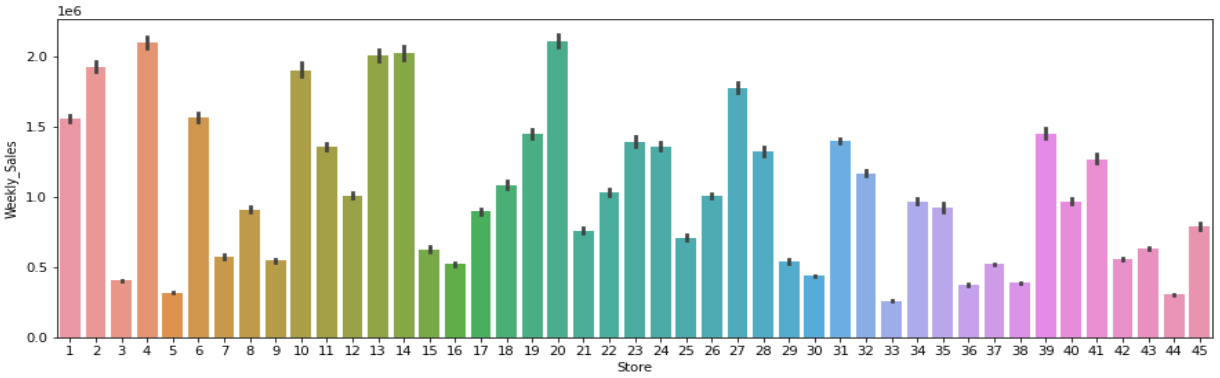
Here we are using the mean SHAP value to measure feature importance with a larger value indicating a larger impact on the model's prediction using that feature.
Which store has the highest weekly sales – Store 20



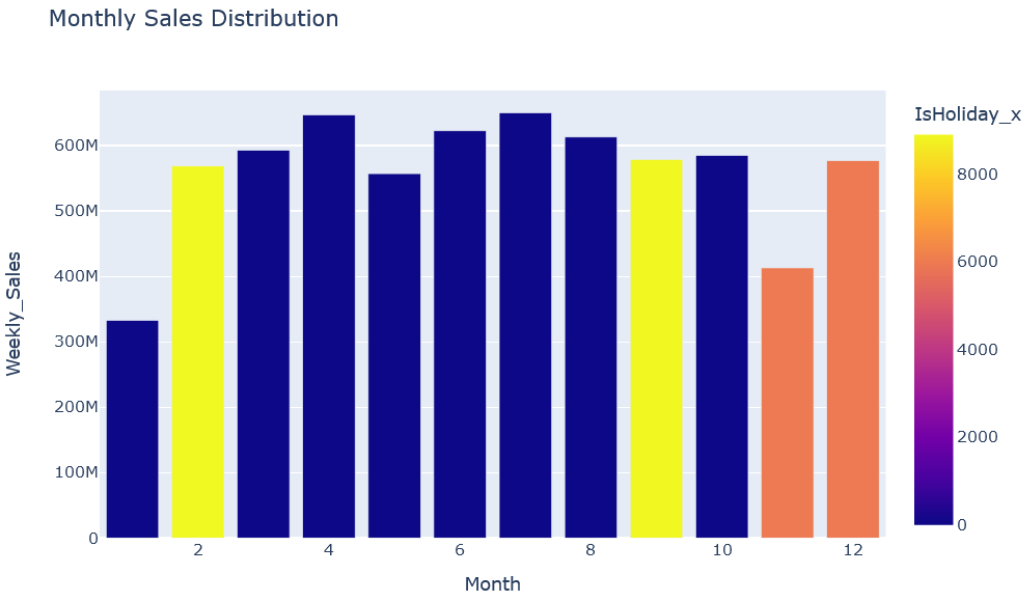
Mean SHAP plot:

Here we are using the mean SHAP value to measure feature importance with a larger value indicating a larger impact on the model's prediction using that feature.

Which store has the highest weekly sales – Store 20



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



From the above bar graph, we can see that :

The Sales are maximum in the month of April, July, September , October which are not

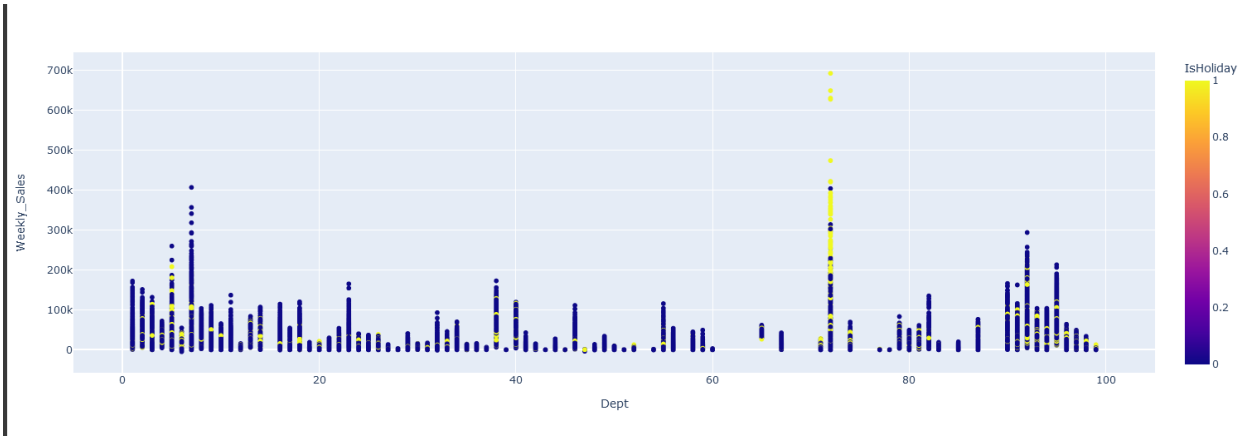
Holiday Months.

We can also see that good Sales happen in the month of Feb, Sep , December which are Holiday Month.

Least Sales takes place in the month of January which is quite expected as it falls just after the Holiday Month December.

All of this based on the top 35% of the dept sales for the first 10 stores.

i. Identify the best department across the first ten stores. (hint – department 38)]Identify the best department across the first ten stores. (hint – department 38)



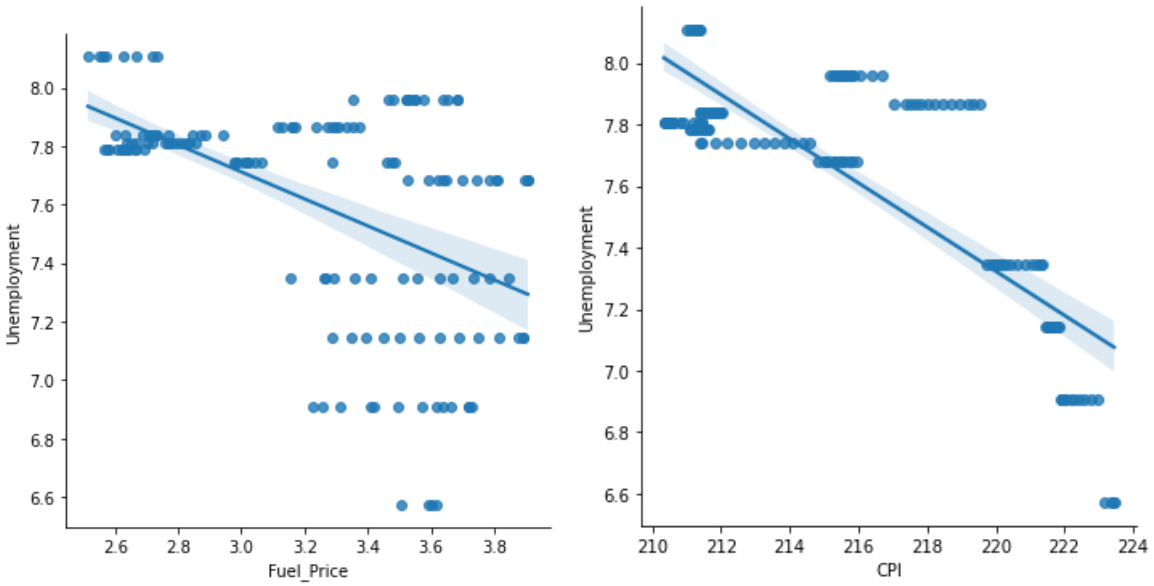
According to this plot, we can highlight that the dept with the highest sales for one week is the Dept 72. It has the highest sales and it is mainly during holidays.

Row Labels	Sum of Weekly_Sales
2	280611174.4
38	393118136.9
40	288936022
72	305725152.2
90	291068463.7
91	216781705.7
92	483943341.9
95	449320162.5
Grand Total	**2709504159**

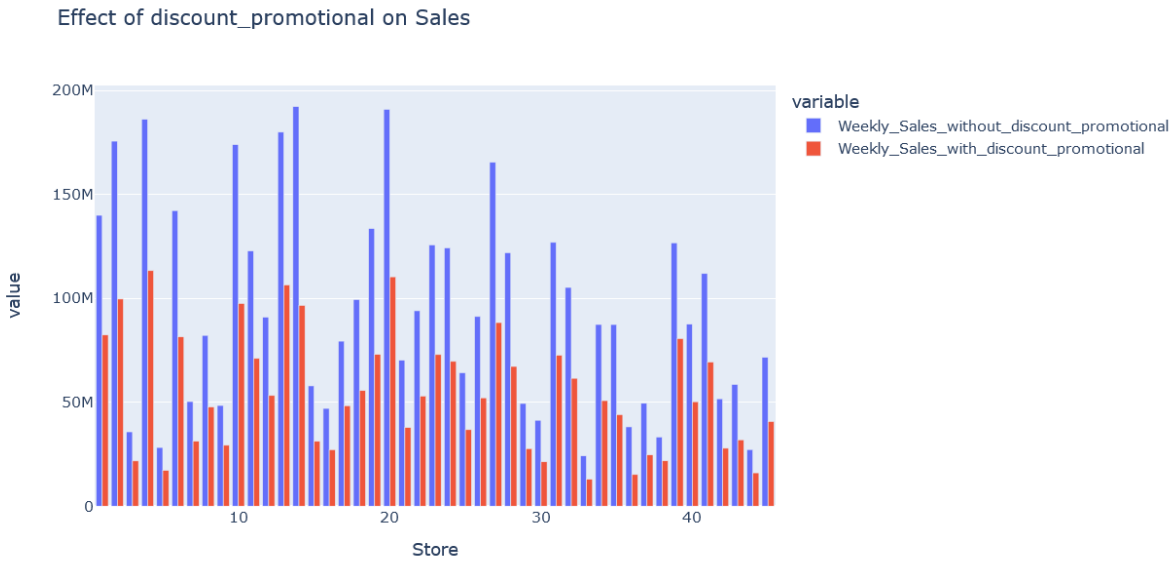
This extract from Excel shows that dept 92 is the best seller if we just sum all the weekly sales.

(c) Investigate the relationship between weekly sales over CPI and unem- ployment for the first 10 stores. You can explore the what-if scenarios while writing the report.

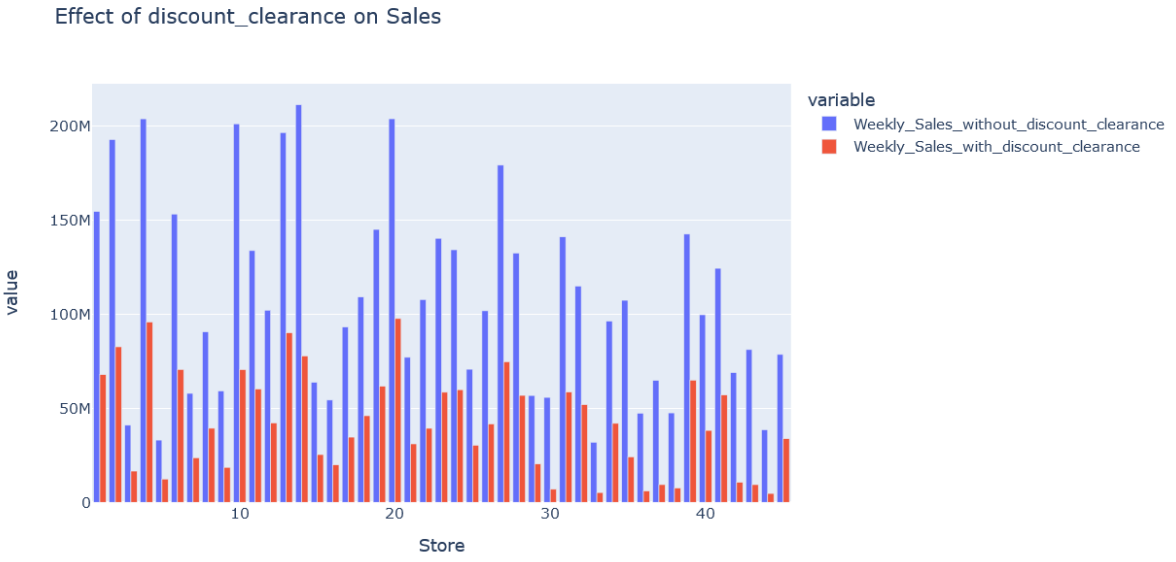
Here we can see using Seaborn plots that as the gas_price and CPI goes high, rate of unemployment decreases.



(d) Investigate the impact of various types of discounts, for example, dis- count promotional, discount clearance, discount damaged good, discount competitive and discount employee on the overall sales.



From the above chart, we can see that the sales are high without discount_promotional values which makes sense as the discount_promotional values are reduced in price. Hence its likely to have lower sales for weeks when discount_promotional is applied. The results is the same for others discount as we can see below



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

According to the different plot we made no discount is helpful because the price will be lower.

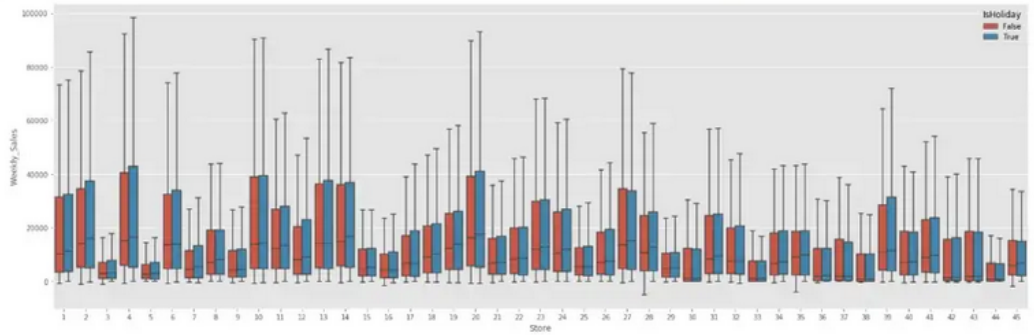
ii. **Does the observed behavior hold true for all the**

stores? Consider the bottom 30% of the least performing store (sales per 1000 square feet).

The observed behavior seems to be true for all the stores with the graph plotted.

(e) Identify the departments which are highly impacted by external factors: "temperature", "gas price", and "holiday". Is there any correlation between overall sales and holidays?

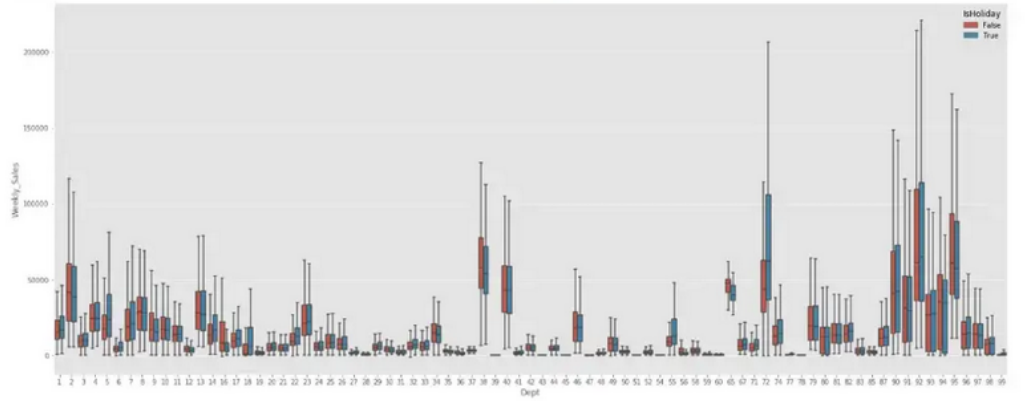
```
In [17]: data = pd.concat([train['Store'], train['Weekly_Sales'], train['IsHoliday']], axis=1)
         f, ax = plt.subplots(figsize=(25, 8))
         fig = sns.boxplot(x='Store', y='Weekly_Sales', data=data, showfliers=False, hue="IsHoliday")
```



Holiday and Store do not show significant relations but just small higher sales soaring when holiliday

We wanted to highlight if there is any relation between sales and holiday.

```
In [20]: data = pd.concat([train['Dept'], train['Weekly_Sales'], train['IsHoliday']], axis=1)
         f, ax = plt.subplots(figsize=(25, 10))
         fig = sns.boxplot(x='Dept', y='Weekly_Sales', data=data, showfliers=False, hue="IsHoliday")
```



Unlike store and holiday relation, department and holiday do not explain any relation 72 department shows the highest surge in sales during holiday However others don't and even more in some departments non-holidays' sales are higher. That means the character of the product (department) is different from sales.

2.2 Dataset 02

Figure 1 depicts the distribution of hobbies, home items, and food. Food is distributed properly and peaks at 1.5. The hobbies have a tendency where a few inexpensive goods sell the most frequently. Moreover, A normal distribution bell curve for the household is crested at 2.0 with highest frequency, with the exception of items noted at 0.5-0.75.

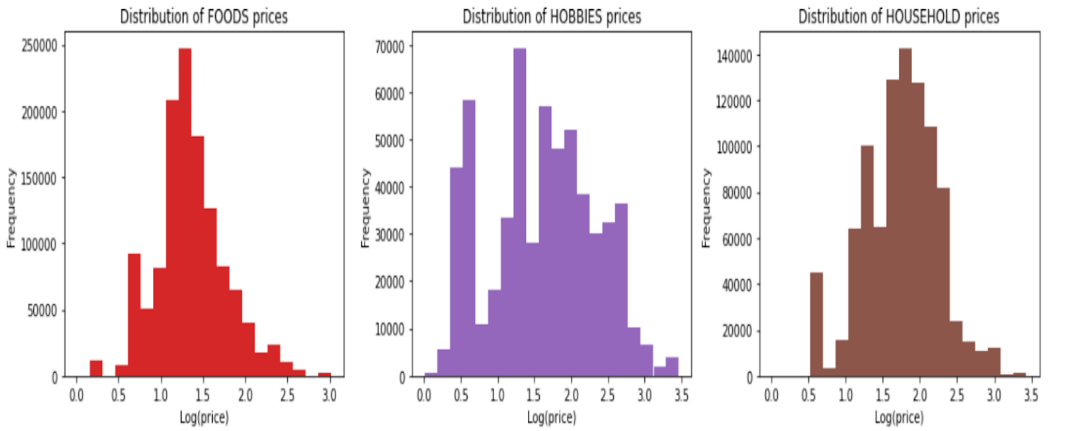


Figure 1

Aggregate Sales by State are shown in Figure 2. In all 3 states, the same seasonal trend can be seen, followed by a significant drop in sales on December 25 (Christmas), when businesses are closed. Additionally, there is a discrepancy in Texas' overall sales trend. Only on June 15, 2015, sales saw an unexpected rise, which doesn't appear to apply to other states.

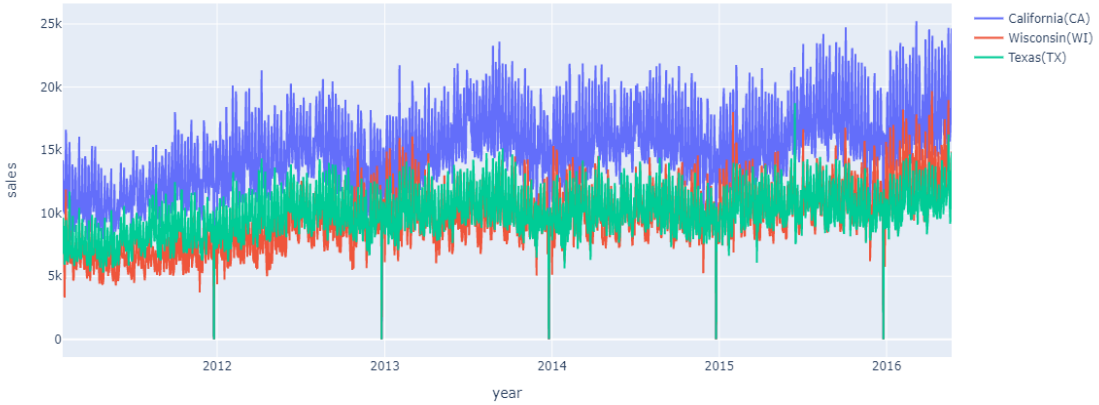


Figure 2

Figure 3 shows that while average aggregate sales for stores in both California and Wisconsin are on the rise, Wisconsin retailers are growing more steadily than those in California. However, After July 2012, Texas shops exhibited a consistent trend in their sales.

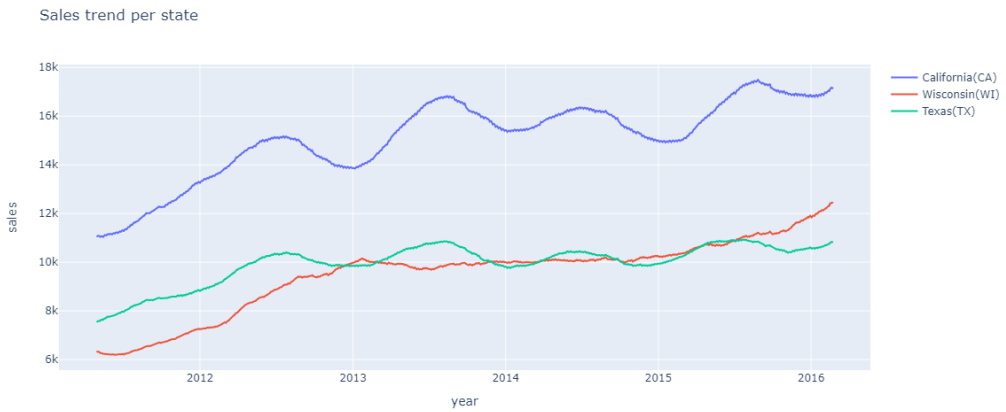


Figure 3

Figure 4 shows that average sales are substantially higher on Saturday and Sunday than on other days of the week in all three states. This pattern demonstrates how the day of the week might be a crucial variable to take into account when predicting unit sales.

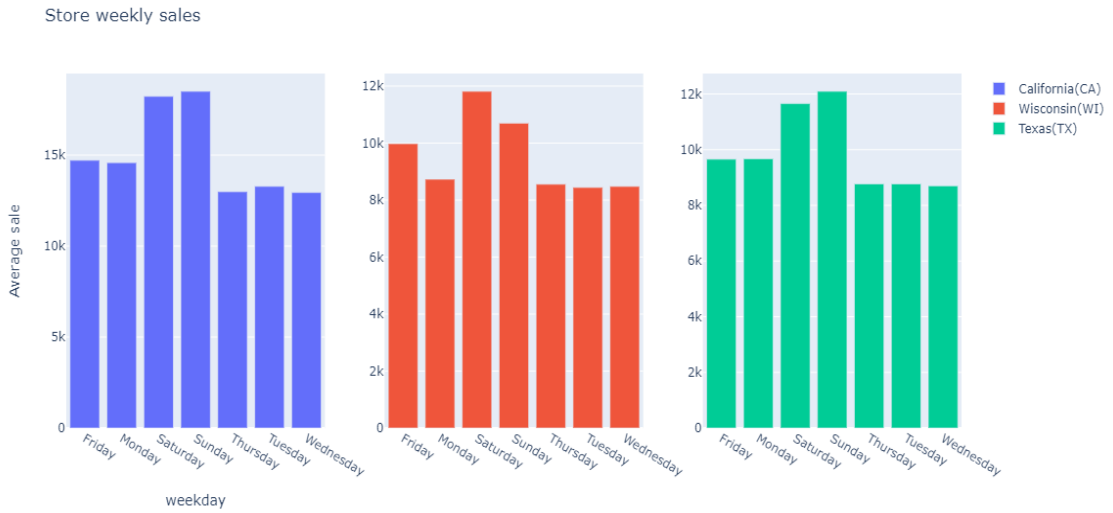


Figure 4

Figure 5 demonstrates that Wisconsin has the greatest average sales in February, whereas California and Texas have the highest average sales in August. May average sales in all 3 states indicate a significant decline. Monthly average sales in each state vary greatly. This pattern demonstrates the significance of the month of the year in predicting unit sales.

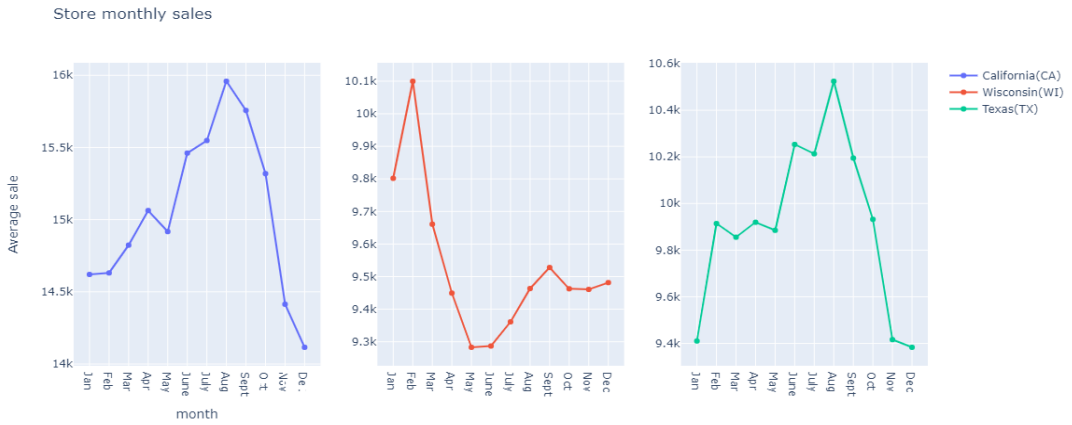


Figure 5

Figure 6 graph may be the most impactful in understanding that there is a noticeable decrease in the average sales for all categories during national events. Sports events have an impact on the average sales of the FOOD category. This graph demonstrates how calendar events may have a significant role in predicting unit sales.

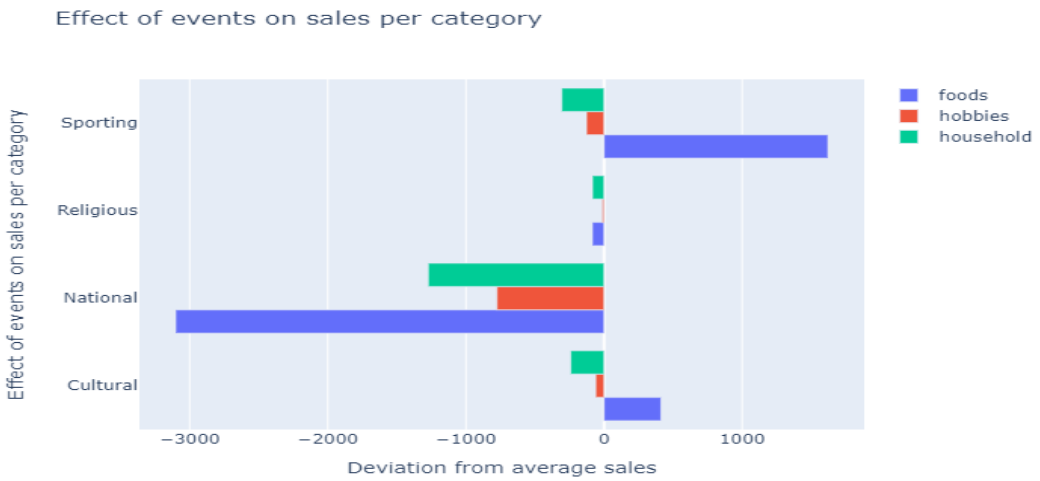


Figure 6

3 Phase 2

A snapshot of the data and understand the current business from the given dataset using Tableau for the visualization for the second dataset. The goal is to create a Tableau dashboard and publish the results to present the information to an upper higher-level management of an organization.

3.1 Dataset 01

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. There are external variables such as gas price, holidays, unemployment, and temperature for the given dataset. Evaluate the impact of these external variables on the accuracy of the model (do they help improve the accuracy?). Plot the relevant graphs.

   (a) Begin with Linear regression model to forecast the weekly sales using the given features. Hints: Feature selection, feature engineering (new features, if any), PCA.

   Linear Regression to forecast weekly sales for Dataset 01

   After thorough analysis of the correlation matrix, the columns such as IsHoliday, Temperature, gas_price, Date were dropped and the rest were plotted to attain the following results.

   The Mean Square Error(MSE) or J(theta) is: 43975850461.28765
   R square obtain for scikit learn library is : 0.9120388635519361

```
In [120]: from sklearn.metrics import mean_squared_error
          J_mse_sk = mean_squared_error(y_pred_sk, y_test)
```

```
In [121]: R_square_sk = lin_reg.score(X_test,y_test)
          print('The Mean Square Error(MSE) or J(theta) is: ',J_mse_sk)
          print('R square obtain for scikit learn library is :',R_square_sk)

          The Mean Square Error(MSE) or J(theta) is:  43975850461.28765
          R square obtain for scikit learn library is : 0.9120388635519361
```

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

As you can see, the rolling mean and rolling standard deviation increase with time. Therefore, we can conclude that the time series is not station- ary.
To get it Stationary :

ADF : an augmented Dickey–Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. A p-value is a statistical measurement used to validate a hypothesis against observed data.

A critical value defines regions in the sampling distribution of a test statis- tic.

The ADF Statistic is far from the critical values and the p-value is greater than the threshold. Thus, we can conclude that the time series is not sta- tionary.

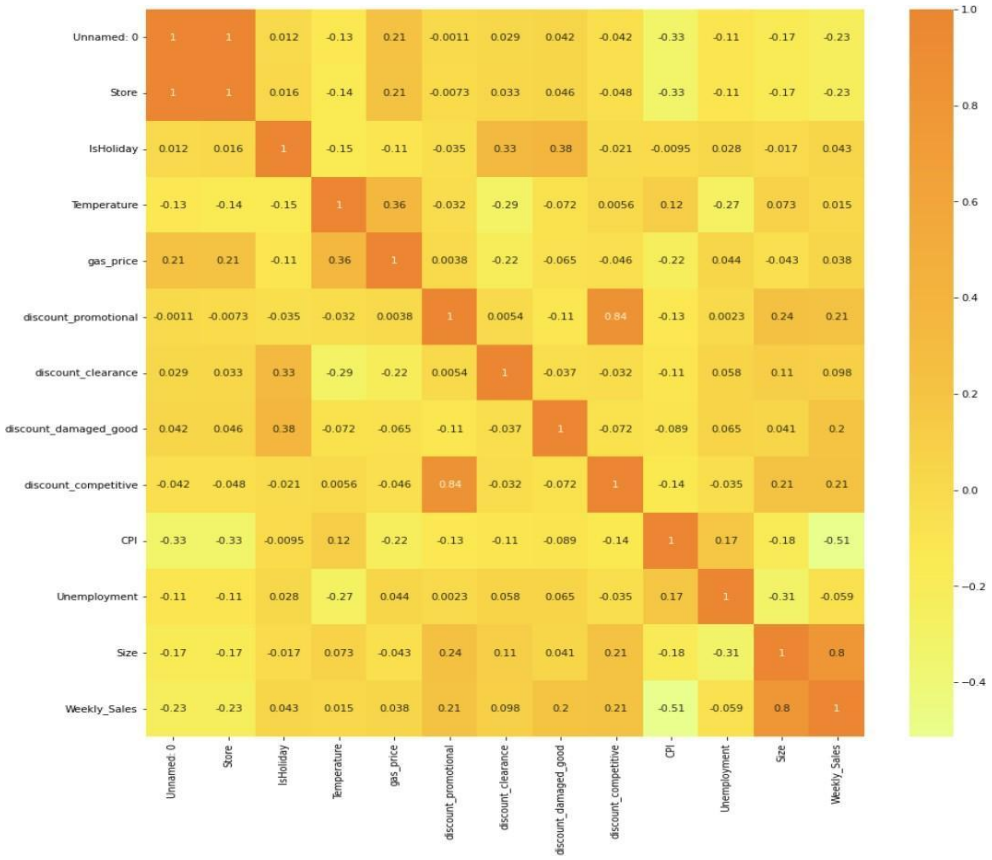


Figure 2

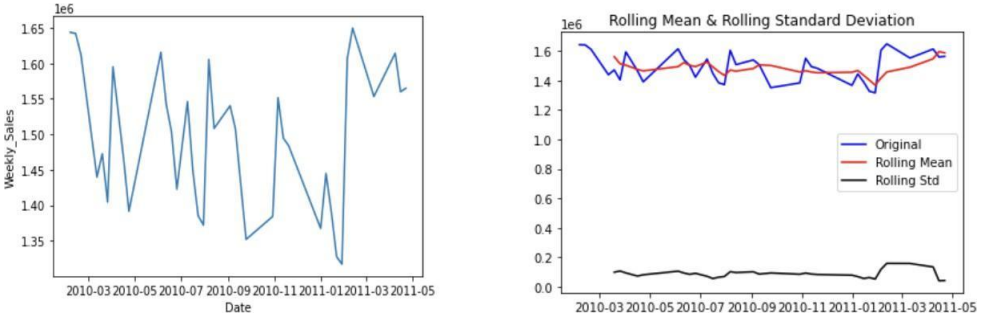


Figure 3

Taking the log of the dependent variable is a simple way of lowering the rate at which rolling mean increases.

Plotting the df log

Calculated the rolling statistics and did the Dickey–Fuller test and the plot looks like below:

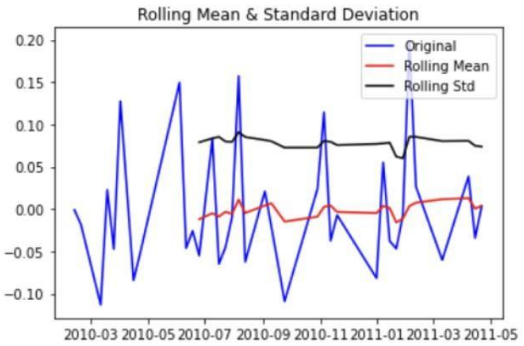
In [218]:
```
def get_stationarity(timeseries):

    # rolling statistics
    rolling_mean = timeseries.rolling(window=12).mean()
    rolling_std = timeseries.rolling(window=12).std()

    # rolling statistics plot
    original = plt.plot(timeseries, color='blue', label='Original')
    mean = plt.plot(rolling_mean, color='red', label='Rolling Mean')
    std = plt.plot(rolling_std, color='black', label='Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    # Dickey-Fuller test:
    result = adfuller(timeseries['Weekly_Sales'])
    print('ADF Statistic: {}'.format(result[0]))
    print('p-value: {}'.format(result[1]))
    print('Critical Values:')
    for key, value in result[4].items():
        print('\t{}: {}'.format(key, value))
```

In [219]:
```
rolling_mean = df_log.rolling(window=16).mean()
df_log_minus_mean = df_log - rolling_mean
df_log_minus_mean.dropna(inplace=True)
get_stationarity(df_log_minus_mean)
```

Figure 4

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

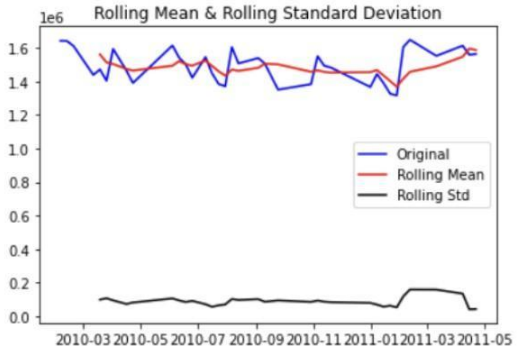
In [252]:
```
df_log_shift = df_log - df_log.shift()
df_log_shift.dropna(inplace=True)
get_stationarity(df_log_shift)
```




```
ADF Statistic: -5.303296074937923
p-value: 5.390224047672758e-06
Critical Values:
        1%: -3.6461350877925254
        5%: -2.954126991123355
        10%: -2.6159676124885216
```

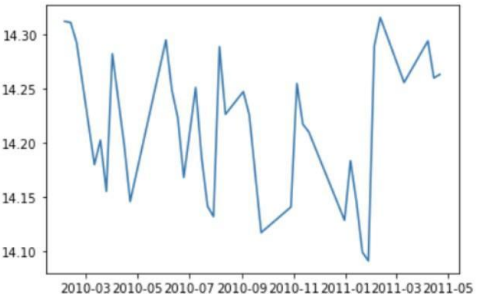
Figure 5

In [215]:
```
rolling_mean = dfb.rolling(window = 5).mean()
rolling_std = dfb.rolling(window = 5).std()
plt.plot(dfb, color = 'blue', label = 'Original')
plt.plot(rolling_mean, color = 'red', label = 'Rolling Mean')
plt.plot(rolling_std, color = 'black', label = 'Rolling Std')
plt.legend(loc = 'best')
plt.title('Rolling Mean & Rolling Standard Deviation')
plt.show()
```

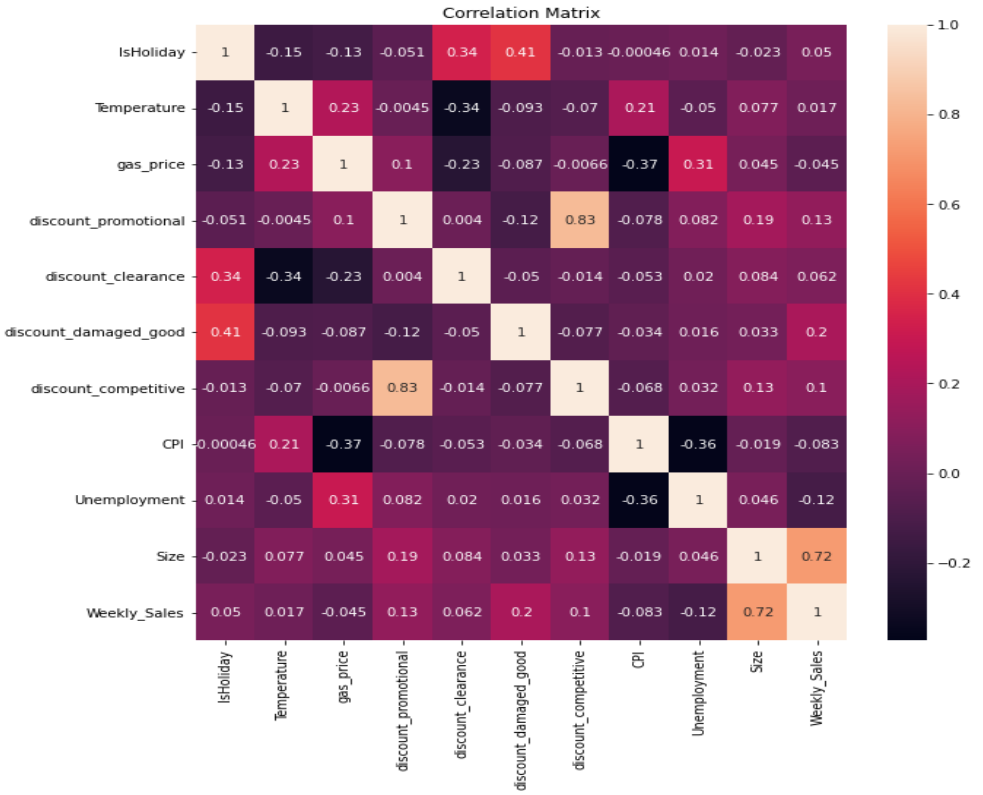


In [251]:
```
df_log = np.log(dfb)
plt.plot(df_log)
```

Out[251]:
```
[<matplotlib.lines.Line2D at 0x7fdb3b640d90>]
```



2. In stores.csv, we have a feature "type", i.e., three types of stores – A (Super center), B (Discount center), and C (Neighborhood markets). Now, based on the given features in the dataset, can we predict the store type? Please consider all 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.

i. Ensemble models (3 statistical methods)

Linear Regression Score for store type prediction

	precision	recall	f1-score	support
0	0.93	0.97	0.95	178
1	0.96	0.90	0.93	137
accuracy			0.94	315
macro avg	0.94	0.93	0.94	315
weighted avg	0.94	0.94	0.94	315

XGBRegressor Score for store type prediction

	precision	recall	f1-score	support
0	1.00	0.99	1.00	178
1	0.99	1.00	1.00	137
accuracy			1.00	315
macro avg	1.00	1.00	1.00	315
weighted avg	1.00	1.00	1.00	315

RandomForestRegressor Score for store type prediction

	precision	recall	f1-score	support
0	1.00	0.99	1.00	178
1	0.99	1.00	1.00	137
accuracy			1.00	315
macro avg	1.00	1.00	1.00	315
weighted avg	1.00	1.00	1.00	315

Ensemble Method Score for store type prediction

	precision	recall	f1-score	support
0	1.00	0.99	1.00	178
1	0.99	1.00	1.00	137
accuracy			1.00	315
macro avg	1.00	1.00	1.00	315
weighted avg	1.00	1.00	1.00	315

Ensemble Method AVG Score for store type prediction

	precision	recall	f1-score	support
0	1.00	0.99	1.00	178
1	0.99	1.00	1.00	137
accuracy			1.00	315
macro avg	1.00	1.00	1.00	315
weighted avg	1.00	1.00	1.00	315

Results:

We can see that the ensemble method predicts very well with a 99% chance to predict the type of store.

100% to predict A store
99% to predict B store

The prediction is better with ensemble than the one used so here this model works really well.

ii. Recurrent neural network (RNN)

For the RNN we decided to use a few features from the dataset. We used the ones that were really correlated with the Type of store.
Below the correlation matrix :

	Store	discount_promotional	discount_competitive	Size	Weekly_Sales	Types
Store	1.000000	0.064992	0.042552	0.129496	-0.177068	0.014893
discount_promotional	0.064992	1.000000	0.827387	0.190989	0.130576	-0.106679
discount_competitive	0.042552	0.827387	1.000000	0.135965	0.105474	-0.063823
Size	0.129496	0.190989	0.135965	1.000000	0.719277	-0.849388
Weekly_Sales	-0.177068	0.130576	0.105474	0.719277	1.000000	-0.609449
Types	0.014893	-0.106679	-0.063823	-0.849388	-0.609449	1.000000

The features remaining here are correlated and make a sense compare to some score with other features (with other discount the correlation is 0.001)

We created a new feature “Type” which is 0, 1 or 2 depending on A, B or C type stores.
Then the y_test split this feature as : ((1;0;0),(0;1;0)….(0;0;1))
This is for putting A, B and C stores in the same place as labels.

For RNN we picked the optimizer Adam because it is the most efficient optimizer in this situation.

```
Epoch 1/10
51/51 [==============================] - 2s 15ms/step - loss: 0.7807 - accuracy: 0.5327 - val_loss: 0.6519 - val_accuracy: 0.7470
Epoch 2/10
51/51 [==============================] - 0s 9ms/step - loss: 0.6316 - accuracy: 0.6637 - val_loss: 0.5730 - val_accuracy: 0.8617
Epoch 3/10
51/51 [==============================] - 1s 10ms/step - loss: 0.4019 - accuracy: 0.8214 - val_loss: 0.4847 - val_accuracy: 0.8221
Epoch 4/10
51/51 [==============================] - 0s 10ms/step - loss: 0.2740 - accuracy: 0.8938 - val_loss: 0.2701 - val_accuracy: 0.9170
Epoch 5/10
51/51 [==============================] - 0s 10ms/step - loss: 0.2514 - accuracy: 0.8919 - val_loss: 0.3660 - val_accuracy: 0.8340
Epoch 6/10
51/51 [==============================] - 0s 9ms/step - loss: 0.2039 - accuracy: 0.9077 - val_loss: 0.2133 - val_accuracy: 0.9802
Epoch 7/10
51/51 [==============================] - 1s 10ms/step - loss: 0.1933 - accuracy: 0.9266 - val_loss: 0.2232 - val_accuracy: 0.9802
Epoch 8/10
51/51 [==============================] - 0s 9ms/step - loss: 0.2033 - accuracy: 0.9028 - val_loss: 0.2743 - val_accuracy: 0.9684
Epoch 9/10
51/51 [==============================] - 0s 9ms/step - loss: 0.1388 - accuracy: 0.9435 - val_loss: 0.2297 - val_accuracy: 0.9802
Epoch 10/10
51/51 [==============================] - 1s 10ms/step - loss: 0.1047 - accuracy: 0.9583 - val_loss: 0.2335 - val_accuracy: 0.9802
8/8 [==============================] - 0s 5ms/step
```

With parameters set as batch_size=20,epochs=10,verbose=1 we will check the accuracy compared to the CNN.
This score is supposed to be high as we used the Adam optimizer.

iii. Convolutional Neural Networks (CNN)

CNN is known for having a better score when we apply this neural network.
With the same parameters and the Adam model this is what we have :

```
Epoch 1/10
51/51 [==============================] - 1s 13ms/step - loss: 0.7599 - accuracy: 0.5427 - val_loss: 0.6027 - val_accuracy: 0.7579
Epoch 2/10
51/51 [==============================] - 1s 10ms/step - loss: 0.6347 - accuracy: 0.6511 - val_loss: 0.4579 - val_accuracy: 0.9405
Epoch 3/10
51/51 [==============================] - 0s 10ms/step - loss: 0.3949 - accuracy: 0.8380 - val_loss: 0.1785 - val_accuracy: 0.9881
Epoch 4/10
51/51 [==============================] - 1s 10ms/step - loss: 0.3248 - accuracy: 0.8559 - val_loss: 0.2377 - val_accuracy: 0.8571
Epoch 5/10
51/51 [==============================] - 1s 10ms/step - loss: 0.2584 - accuracy: 0.8897 - val_loss: 0.3073 - val_accuracy: 0.8373
Epoch 6/10
51/51 [==============================] - 0s 10ms/step - loss: 0.2253 - accuracy: 0.9066 - val_loss: 0.1791 - val_accuracy: 0.8968
Epoch 7/10
51/51 [==============================] - 0s 9ms/step - loss: 0.1862 - accuracy: 0.9314 - val_loss: 0.5597 - val_accuracy: 0.8294
Epoch 8/10
51/51 [==============================] - 0s 9ms/step - loss: 0.2456 - accuracy: 0.8917 - val_loss: 0.0873 - val_accuracy: 0.9921
Epoch 9/10
51/51 [==============================] - 0s 9ms/step - loss: 0.1619 - accuracy: 0.9284 - val_loss: 0.1733 - val_accuracy: 0.9286
Epoch 10/10
51/51 [==============================] - 0s 9ms/step - loss: 0.1248 - accuracy: 0.9463 - val_loss: 0.0992 - val_accuracy: 0.9881
8/8 [==============================] - 0s 4ms/step
```

Now that we runned both neural network models, let’s compare the accuracy of each.

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

Accuracy score :

```
print("RNN accuracy :",round(accuracy,2))
print("CNN accuracy :",round(accuracy2,2))
```

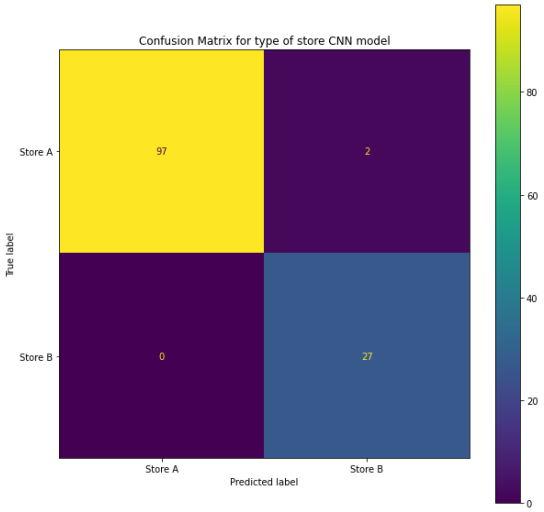
```
RNN accuracy : 96.83
CNN accuracy : 98.41
```

Here the accuracy is 98.41 for the CNN model, so almost 2 % better than with RNN (96.83) this highlights the efficiency of CNN model.

At this scale, 2% is a big deal, we can't deny it.

Furthermore, both models show their efficiency with such a score.

Confusion matrix:



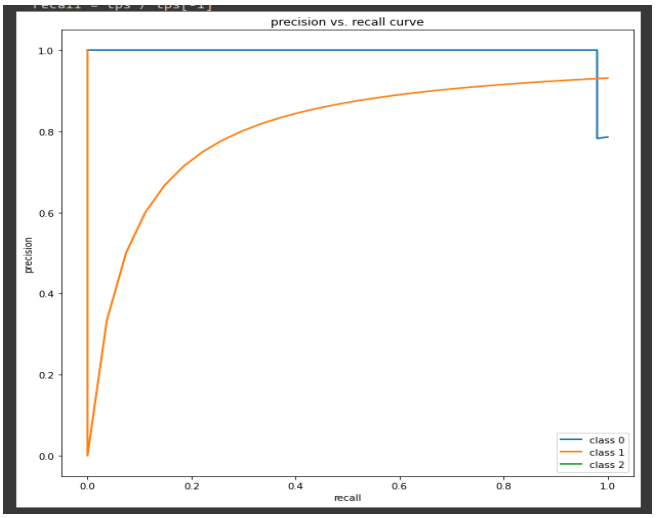
This plot correspond to CNN confusion matrix (the best model of all three)
The few mistakes are because the model thought it was a B store but it was an A store.
The other case does not appear here.

F1 score:

```
print("f1 score for CNN model : \n",f1_score(y_label, predict_label))
```

```
f1 score for CNN model :
 0.9642857142857143
```

For our best model (CNN), the f1 score is 0.964.
Precision & recall:



3.2 Dataset 02

1. **Perform data preprocessing and exploratory data analysis. Hints: Does seasonality and trend exist in the dataset? Drop any column? Extra credit: Perform down-casting (shrink dataset size)**

Exploratory Data Analysis and its Observation

EDA will help us to understand data more clearly and we can add important features using Feature Engineering techniques which helps us to decrease RMSE score or we can remove unimportant features which helps us to reduce the size of data.

Using EDA, we have found two important things:

- whether Seasonality exists in feature
- whether Trend exists in feature

Data Preprocessing:

Every Modeling Strategy has to include a crucial step called data cleaning and preprocessing. Our model could operate improperly if there is inaccurate data present. Errors in calculation, human mistake, and other factors can all contribute to inaccurate data. Here, we have nan values for event name 1, event name 2, and event type 1, event type 2. With no events, we have swapped out these nan values. Additionally, we are attempting to decrease the amount of RAM used by all categorical data, such as item id, cat id, store id, dept id, year, event name 1, event name 2, event type 1, event type 2, year, etc.

Removing Unnecessary columns

```
%%time
# Converting snap_CA,snap_WI,snap_TX into one feature named snap
train.loc[train['state_id'] == 'CA', 'snap'] = train.loc[train['state_id'] == 'CA']['snap_CA']
train.loc[train['state_id'] == 'TX', 'snap'] = train.loc[train['state_id'] == 'TX']['snap_TX']
train.loc[train['state_id'] == 'WI', 'snap'] = train.loc[train['state_id'] == 'WI']['snap_WI']
train.drop(['snap_CA','snap_TX','snap_WI'],axis=1,inplace=True)


test.loc[test['state_id'] == 'CA', 'snap'] = test.loc[test['state_id'] == 'CA']['snap_CA']
test.loc[test['state_id'] == 'TX', 'snap'] = test.loc[test['state_id'] == 'TX']['snap_TX']
test.loc[test['state_id'] == 'WI', 'snap'] = test.loc[test['state_id'] == 'WI']['snap_WI']
test.drop(['snap_CA','snap_TX','snap_WI'],axis=1,inplace=True)

final_test.loc[final_test['state_id'] == 'CA', 'snap'] = final_test.loc[final_test['state_id'] == 'CA']['snap_CA']
final_test.loc[final_test['state_id'] == 'TX', 'snap'] = final_test.loc[final_test['state_id'] == 'TX']['snap_TX']
final_test.loc[final_test['state_id'] == 'WI', 'snap'] = final_test.loc[final_test['state_id'] == 'WI']['snap_WI']
final_test.drop(['snap_CA','snap_TX','snap_WI'],axis=1,inplace=True)
```

Drop Column :

```
%%time
# Weekday as wday are similar features so we remove it
#no use of  wm_yr_wk feature
train.drop('weekday',axis=1,inplace=True)
train.drop('wm_yr_wk',axis=1,inplace=True)

test.drop('weekday',axis=1,inplace=True)
test.drop('wm_yr_wk',axis=1,inplace=True)

final_test.drop('weekday',axis=1,inplace=True)
final_test.drop('wm_yr_wk',axis=1,inplace=True)
```

```
CPU times: user 979 ms, sys: 878 ms, total: 1.86 s
Wall time: 3.09 s
```

All additional categorical characteristics found in the data have also been subjected to our Label Encoder application. In addition, we are consolidating snap ca, snap tx, and snap wi into one column called snap.

Exploratory Data Analysis:

For the aim of analysis, exploratory data analysis is a critical component. We may use it to discover patterns in the data and acquire a broad picture of the

information. The two primary Python libraries Seaborn & Matplotlib were used to perform EDA on the Walmart sales data. The following plots are a few that we have created.

Sales Variation According to Categories of Items

Here in Figure 1, we've charted the typical sales of items according to their categories. With the use of this plot, we were able to determine that FOODS category items had higher average sales than the other two, while HOBBIES category items have the lowest average sales of the three. Furthermore, FOODS category average sales are significantly greater than those of the other two categories.

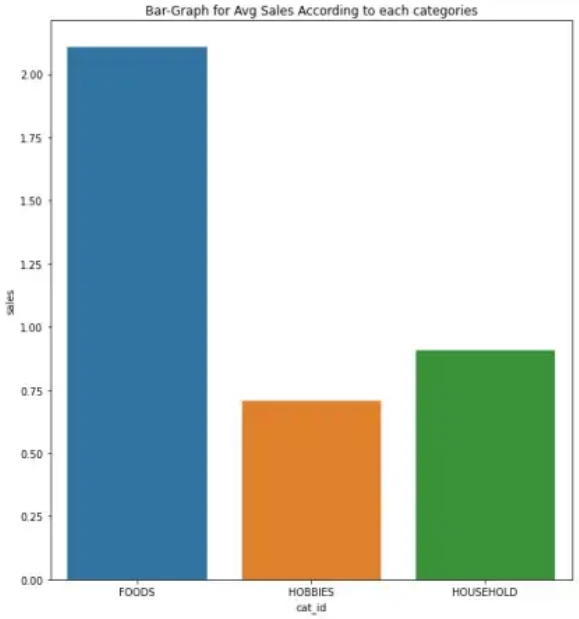


Figure1

Sales Variation According to States where Items are Sold

The bar graph illustrates the differences in average sales between the three states of Wisconsin, Texas, and California. We learn from this plot that California has the highest average sales, but there is not a significant difference between the other two states.

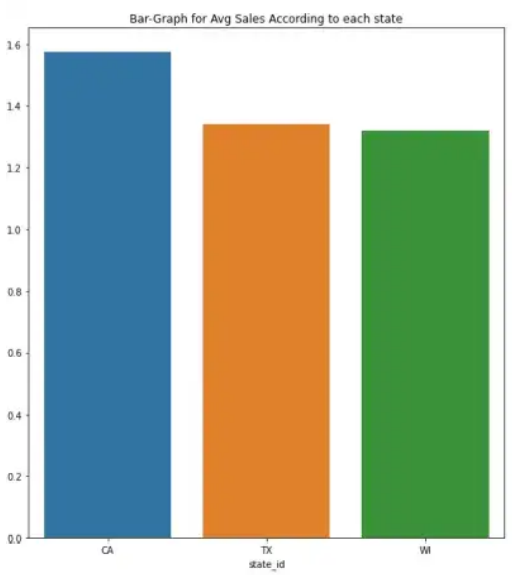


Figure2

Sales Variation According to Stores where Items are Sold

The Bar Plot depicts how typical sales of products differ depending on the store where they are sold. This curve indicates that the store with store id CA 3 has the highest average sales and the store with store id CA 4 has the lowest average sales.

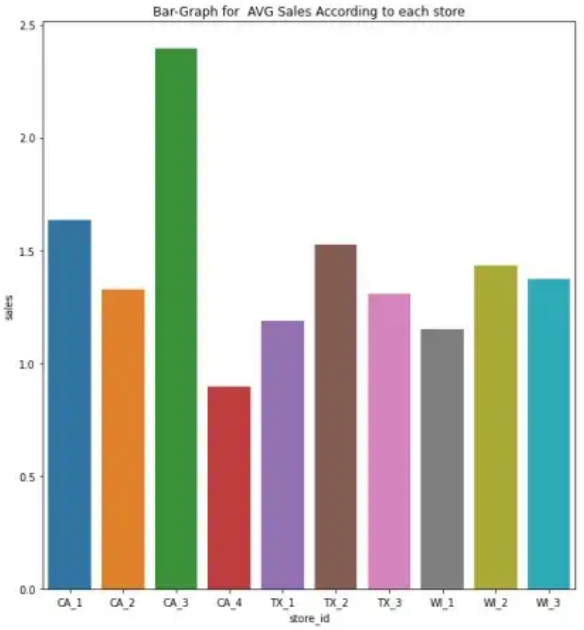


Figure3

Average Sales of Products according to day of week

The Bar Plot displays how sales fluctuate depending on the day of the week. As you can see, the numbers 1 and 2 correspond to Saturday, Sunday and Monday, respectively. We learn from this story that weekends have higher average sales (Saturday and Sunday).

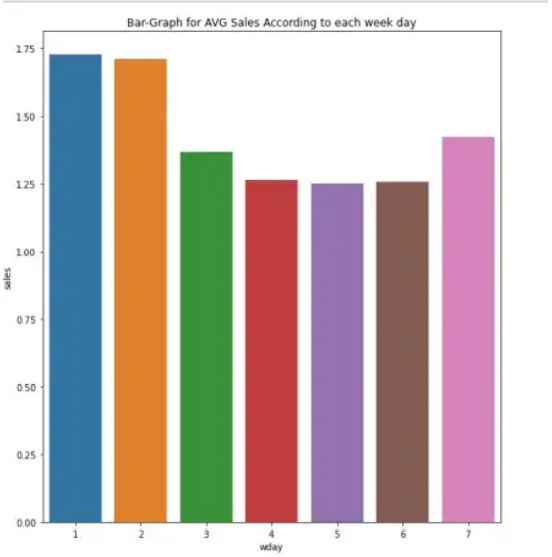


Figure4

Sales For Last 30 days in train Data For Each State

The sales of various items during the past 30 days are depicted in the three graphs below for the states of Texas, California, and Wisconsin.

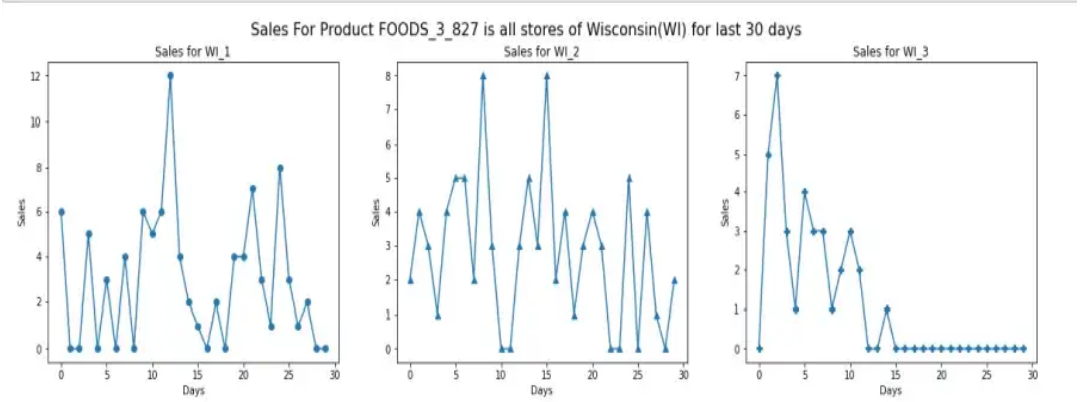


Figure 5.1 Sales of FOODS 3 827 during the previous 30 days across all Wisconsinxas locations.

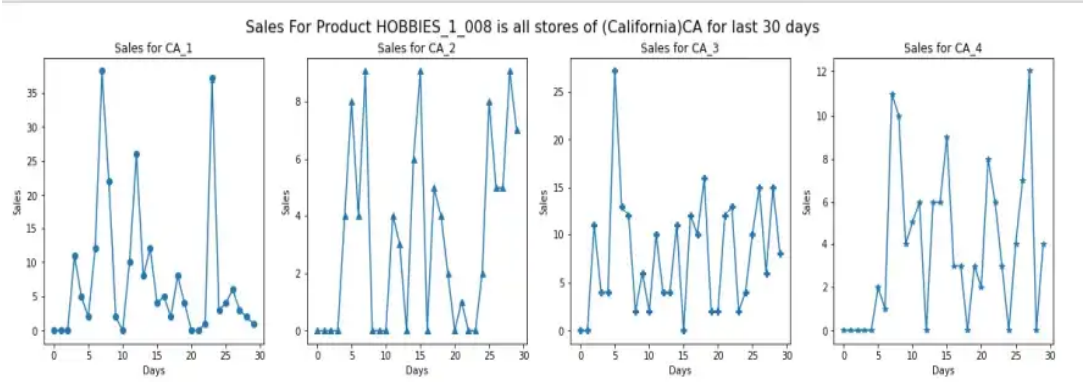


Figure 5.2 HOBBIES_1_008 sales for the last 30 days across all Californian retailers

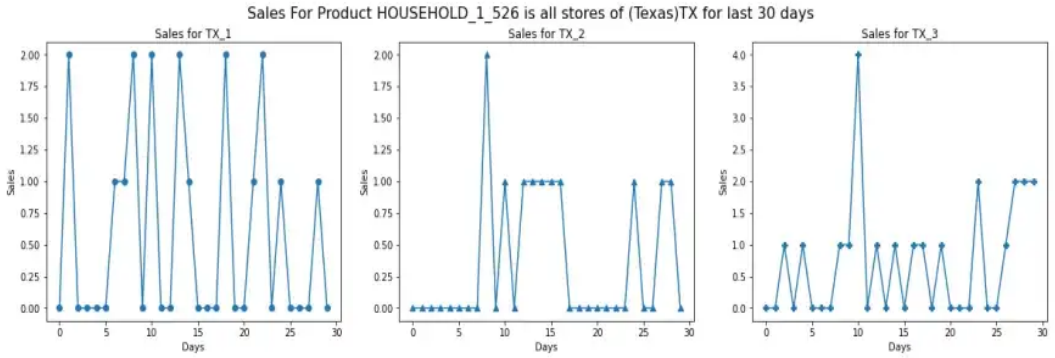


Figure 5.3 Sales of HOUSEHOLD_1_526 at Texas's shops during the previous 30 days

Time Series for Total Sales in all 3 States

The time series depicts the state-by-state trend in sales. We may infer that California always has more sales than the other two states from this.

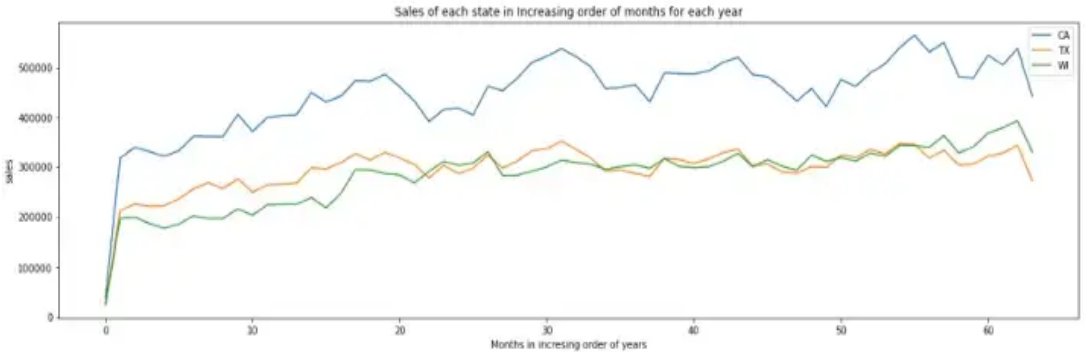


Figure6

We discover that the datasets are significantly larger in size when we utilize the.info() function, which can lead to a problem or, to put it another way, a memory error as we proceed. We alter the datatypes of the various columns in the datasets to avoid this. When reading a dataframe, Pandas automatically assigns the types int64 and float64, even though the integer can also be expressed in the int8 format. Due to the dataframe's bigger size, this causes it to use more RAM. Therefore, we downcast the datasets to ensure that the least amount of RAM is used.

Downcasting is the process to reduce the memory usage by a dataframe and speed up the operations that are to be performed on them.

```
def downcast(df):
    cols = df.dtypes.index.tolist()
    types = df.dtypes.values.tolist()
    for i,t in enumerate(types):
        if 'int' in str(t):
            if df[cols[i]].min() > np.iinfo(np.int8).min and df[cols[i]].max() < np.iinfo(np.int8).max:
                df[cols[i]] = df[cols[i]].astype(np.int8)
            elif df[cols[i]].min() > np.iinfo(np.int16).min and df[cols[i]].max() < np.iinfo(np.int16).max:
                df[cols[i]] = df[cols[i]].astype(np.int16)
            elif df[cols[i]].min() > np.iinfo(np.int32).min and df[cols[i]].max() < np.iinfo(np.int32).max:
                df[cols[i]] = df[cols[i]].astype(np.int32)
            else:
                df[cols[i]] = df[cols[i]].astype(np.int64)
        elif 'float' in str(t):
            if df[cols[i]].min() > np.finfo(np.float16).min and df[cols[i]].max() < np.finfo(np.float16).max:
                df[cols[i]] = df[cols[i]].astype(np.float16)
            elif df[cols[i]].min() > np.finfo(np.float32).min and df[cols[i]].max() < np.finfo(np.float32).max:
                df[cols[i]] = df[cols[i]].astype(np.float32)
            else:
                df[cols[i]] = df[cols[i]].astype(np.float64)
        elif t == np.object:
            if cols[i] == 'date':
                df[cols[i]] = pd.to_datetime(df[cols[i]], format='%Y-%m-%d')
            else:
                df[cols[i]] = df[cols[i]].astype('category')
```

As you can see from the above code, we are reducing the datatypes of the columns based on the values present in those columns. If we look at the datasets after performing downcasting, we observe the new size to be almost 1/4th of what it originally was. The sales data was 454.5MB and then it became 97.1MB only. This helps in faster operations and saving memory.

i. weather data

We pulled up the data from the website provided pertaining to our zip codes and dates in the calender.csv file. The data seems to have precipitation and snow values which helped us understand the seasonal patterns in the area affecting the sales.

	store_id	DATE	DAPR	MDPR	PRCP	SNOW	item_id	wm_yr_wk	sell_price
38998630	CA_1	5/7/2016	NaN	NaN	0.01	NaN	HOUSEHOLD_2_238	11224	10.97
38998631	CA_1	5/7/2016	NaN	NaN	0.01	NaN	HOUSEHOLD_2_238	11225	10.97
38998632	CA_1	5/7/2016	NaN	NaN	0.01	NaN	HOUSEHOLD_2_238	11226	10.97
38998633	CA_1	5/7/2016	NaN	NaN	0.01	NaN	HOUSEHOLD_2_238	11227	10.97
38998634	CA_1	5/7/2016	NaN	NaN	0.01	NaN	HOUSEHOLD_2_238	11	NaN

ii. median income

Data are based on a sample and are subject to sampling variability. The degree of uncertainty for an estimate arising from sampling variability is represented through the use of a margin of error. The value shown here is the 90 percent margin of error. The margin of error can be interpreted roughly as providing a 90 percent probability that the interval defined by the estimate minus the margin of error and the estimate plus the margin of error (the lower and upper confidence bounds) contains the true value. In addition to sampling variability, the ACS estimates are subject to nonsampling error The effect of nonsampling error is not represented in these tables.

Label (Grouping)	ZCTA5 908	ZCTA5 908	ZCTA5 908	ZCTA5 908	ZCTA5 908	ZCTA5 908
HOUSEHOLD INCOME BY RACE AND HISPANIC OR LATINO ORIGIN OF HOUSEHOLDER						
Households	15,027	±603	15,027	±603	52,122	±2,348
One race--						
White	8,533	±569	56.80%	±3.1	54,828	±3,708
Black or African American	2,412	±379	16.10%	±2.3	36,463	±9,438
American Indian and Alaska Native	126	±63	0.80%	±0.4	-	**
Asian	1,996	±230	13.30%	±1.6	51,625	±5,167
Native Hawaiian and Other Pacific Islander	10	±18	0.10%	±0.1	-	**
Some other race	1,502	±278	10.00%	±1.8	55,411	±5,186
Two or more races	448	±153	3.00%	±1.0	64,815	±13,183
Hispanic or Latino origin (of any race)	5,379	±544	35.80%	±3.0	52,069	±2,673
White alone, not Hispanic or Latino	5,024	±411	33.40%	±2.8	61,404	±10,727
HOUSEHOLD INCOME BY AGE OF HOUSEHOLDER						
15 to 24 years	911	±243	6.10%	±1.6	22,539	±2,855
25 to 44 years	7,228	±470	48.10%	±3.1	56,570	±6,815
45 to 64 years	5,290	±584	35.20%	±3.1	53,994	±2,989
65 years and over	1,598	±264	10.60%	±1.9	26,894	±10,904

2. Use the machine learning algorithms below to model n-step ahead fore- casting (n = 10).

Note: First create a model without using any external features, and then create a model with the external features.

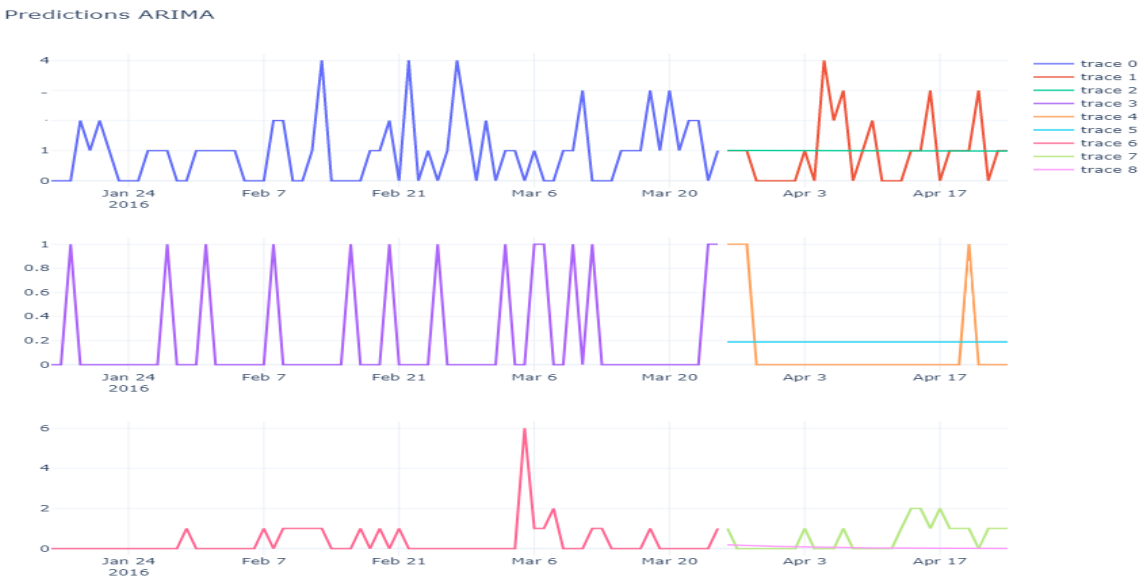
- **Begin with ARIMA and compare the RMSE values for each category**.

ARIMA

Autoregressive Moving Average is known as ARIMA. It results from the combination of two less complex models, the Autoregressive (AR) and the Moving Average (MA). The "MA" section occurs second because, in analysis, we frequently add the residuals to the end of the model equation. Naturally, this will become clear when we look at the equation.

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(1,1,1)[12]             : AIC=inf, Time=0.94 sec
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=1057.977, Time=0.03 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=1058.002, Time=0.10 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=1056.895, Time=0.14 sec
 ARIMA(0,1,1)(0,1,0)[12]             : AIC=1060.491, Time=0.04 sec
 ARIMA(0,1,1)(1,1,1)[12]             : AIC=1057.128, Time=0.35 sec
 ARIMA(0,1,1)(0,1,2)[12]             : AIC=1057.901, Time=0.33 sec
 ARIMA(0,1,1)(1,1,0)[12]             : AIC=1058.006, Time=0.10 sec
 ARIMA(0,1,1)(1,1,2)[12]             : AIC=inf, Time=0.89 sec
 ARIMA(0,1,0)(0,1,1)[12]             : AIC=1060.911, Time=0.10 sec
 ARIMA(1,1,1)(0,1,1)[12]             : AIC=inf, Time=0.67 sec
 ARIMA(0,1,2)(0,1,1)[12]             : AIC=1059.114, Time=0.17 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=1057.048, Time=0.13 sec
 ARIMA(1,1,2)(0,1,1)[12]             : AIC=1061.318, Time=0.20 sec
 ARIMA(0,1,1)(0,1,1)[12] intercept   : AIC=1058.375, Time=0.17 sec

Best model:  ARIMA(0,1,1)(0,1,1)[12]
Total fit time: 4.363 seconds
```

Prediction :



Feature INFO	RSME SCORE
Adding Date based Features	11.05074764647
Adding Lag Features	4.07227605668
Adding rolling window features	3.04637774892

Observation:

- After looking at the result I can say that the Lag feature helps a lot to reduce RMSE.

- **Long short-term memory (LSTM) – Perform hyper-parameter to im- prove the model.**

Long Short Term Memory (LSTM)

There are a few crucial steps to emphasize during the LSTM model training procedure. beginning with the fact that the dataset was divided into training and testing datasets with an 80% to 20% ratio. The MinMaxScaler is then used to rescale them in accordance with the training data. A loop is formed based on the sequence length that describes how long each sequence should be and moves to the next number of data using the sliding windows on the daily sales of the product. There are two levels in the prediction class function: the LSTM layer, which is essentially stackable with other layers, and the output layer, which is contained within the linear layer.

The LSTM layer's final time stamp is passed to the linear layer through the forward method inside the function, which accepts the inputs, all of which are sequences. As a result, there would be just one output in the value's final forecast.

All of our characteristics aside from the event type 1 and event type 2 features and the date-related features are used in this model. The model's data transmission structure is as follows:

Category Features: Each categorical variable was given its own embedding layer, which was subsequently passed via LSTM layers.
Passing all of these numerical features through an LSTM unit

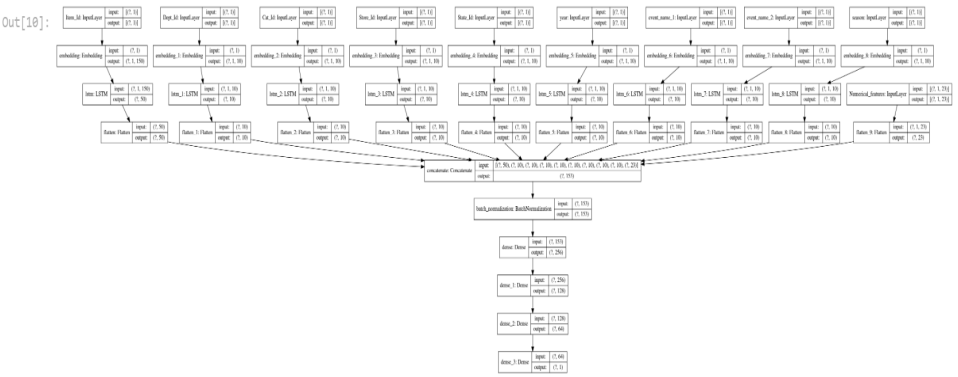
I concatenated the results after obtaining them for all LSTM units and ran them through a number of dense layers.

Mean Squared Error is the loss during model training, while Nadam is the optimizer.

The model's structure is as follows:

In [10]:
```
tf.keras.utils.plot_model(model,show_shapes=True)
```

Out[10]:



Code Snippet :

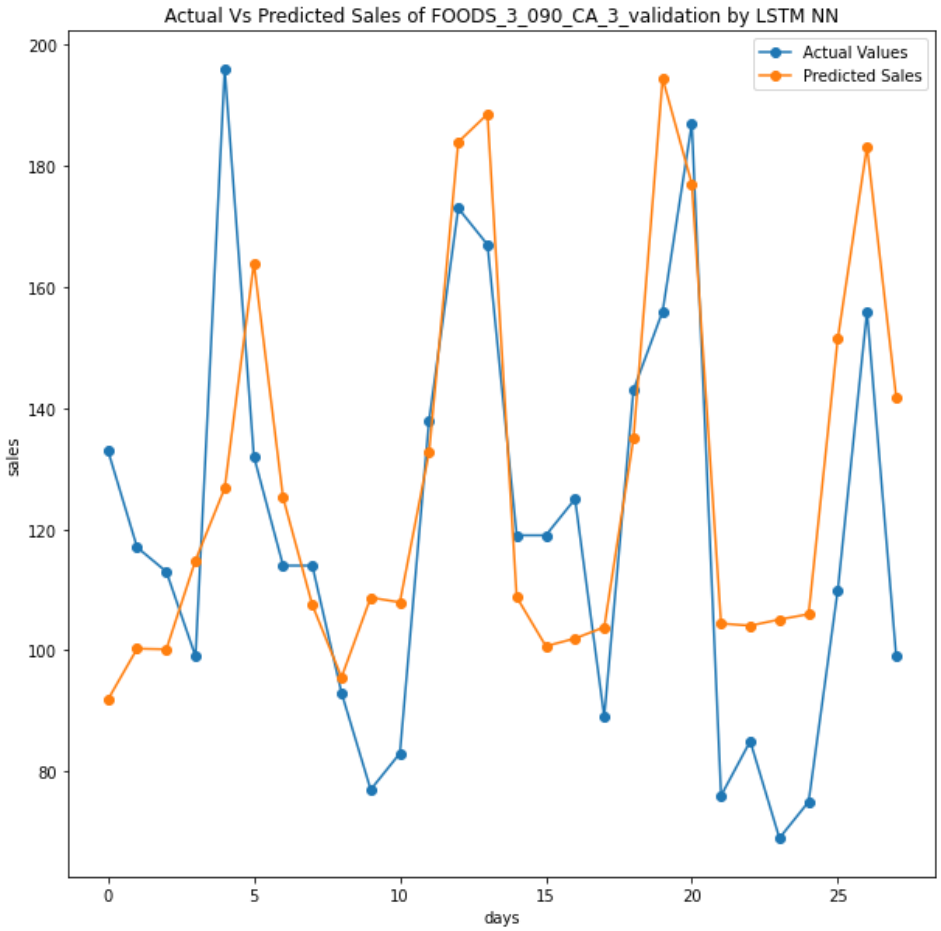
```
tf.keras.backend.clear_session()
input1=tf.keras.layers.Input(shape=1,name='Item_Id')
input2=tf.keras.layers.Input(shape=1,name='Dept_Id')
input3=tf.keras.layers.Input(shape=1,name='Cat_Id')
input4=tf.keras.layers.Input(shape=1,name='Store_Id')
input5=tf.keras.layers.Input(shape=1,name='State_Id')
input6=tf.keras.layers.Input(shape=1,name='year')
input7=tf.keras.layers.Input(shape=1,name='event_name_1')
input8=tf.keras.layers.Input(shape=1,name='event_name_2')
input9=tf.keras.layers.Input(shape=1,name='season')
input10=tf.keras.layers.Input(shape=(1,23),name='Numerical_features')

emb1=tf.keras.layers.Embedding(3050,output_dim=150)(input1)
emb2=tf.keras.layers.Embedding(8,output_dim=10)(input2)
emb3=tf.keras.layers.Embedding(4,output_dim=10)(input3)
emb4=tf.keras.layers.Embedding(11,output_dim=10)(input4)
emb5=tf.keras.layers.Embedding(4,output_dim=10)(input5)
emb6=tf.keras.layers.Embedding(2017,output_dim=10)(input6)
emb7=tf.keras.layers.Embedding(32,output_dim=10)(input7)
emb8=tf.keras.layers.Embedding(6,output_dim=10)(input8)
emb9=tf.keras.layers.Embedding(5,output_dim=10)(input9)

lstm1=tf.keras.layers.LSTM(50)(emb1)
lstm2=tf.keras.layers.LSTM(10)(emb2)
lstm3=tf.keras.layers.LSTM(10)(emb3)
```

```
Epoch 1/15
45174237/45174237 [==============================] - 636s 14us/sample - loss: 8.7152 - val_loss: 4.9377
Epoch 2/15
45174237/45174237 [==============================] - 609s 13us/sample - loss: 7.5126 - val_loss: 4.9375
Epoch 3/15
45174237/45174237 [==============================] - 608s 13us/sample - loss: 7.2318 - val_loss: 4.8146
Epoch 4/15
45174237/45174237 [==============================] - 607s 13us/sample - loss: 7.0534 - val_loss: 4.7975
Epoch 5/15
45174237/45174237 [==============================] - 608s 13us/sample - loss: 6.9097 - val_loss: 4.8065
Epoch 6/15
45174237/45174237 [==============================] - 610s 13us/sample - loss: 6.8050 - val_loss: 4.8488
Epoch 7/15
45174237/45174237 [==============================] - 609s 13us/sample - loss: 6.7098 - val_loss: 4.8216
Epoch 8/15
45174237/45174237 [==============================] - 607s 13us/sample - loss: 6.6371 - val_loss: 4.8313
Epoch 9/15
45174237/45174237 [==============================] - 608s 13us/sample - loss: 6.5786 - val_loss: 4.7672
Epoch 10/15
```

Prediction :

In [69]:

```
#see how predicted sales varies form actual for FOODS_3_090_CA_3_validation
nn=pd.read_csv('NN.csv')
test.sort_values(['id','date'],inplace=True)
tt=test.pivot_table(index='id',values='sales',columns='d')
tt.reset_index(level=0,inplace=True)
tt['id']=tt['id'].apply(lambda x:x.replace('evaluation','validation'))
actual_sales=tt[tt['id']=='FOODS_3_090_CA_3_validation'].values.flatten()[1:]
pred_sales=nn[nn['id']=='FOODS_3_090_CA_3_validation'].values.flatten()[1:]
plt.figure(figsize=(10,10))
plt.plot(actual_sales,label='Actual Values',marker='o')
plt.plot(pred_sales,label='Predicted Sales',marker='o')
plt.title("Actual Vs Predicted Sales of FOODS_3_090_CA_3_validation by LSTM NN")
plt.ylabel('sales')
plt.xlabel('days')
plt.legend()
plt.show()
```



The Figure shows how this model performs on test data. It shows variation between actual & predicted sales of id FOODS_3_090_CA_3_validation.

Hyper-parameters :

```
def GridSearch(x_train,y_train,x_CV,y_CV,parameter):

  learning_rate=parameter['learning_rate']
  n_estimators=parameter['n_estimators']
  train_scores =[]
  CV_scores =[]
  parameter_str=[]
  for i in tqdm(n_estimators):
    for j in tqdm(learning_rate):
      clf=LSTM(objective='poisson' ,learning_rate=j,n_estimators=i)
      clf.fit(x_train,y_train)
      ## predicting CV data
      y_predict=clf.predict(x_CV)
      RMSE=mean_squared_error(y_CV, y_predict, squared=False)
      CV_scores.append(RMSE)
      ## predicting training data
      y_predict_=clf.predict(x_train)
      RMSE_=mean_squared_error(y_train, y_predict_, squared=False)
      train_scores.append(RMSE_)

      parameter_str.append('n_est:'+str(i)+'__'+'LR:'+str(j))
  return train_scores,CV_scores,parameter_str

parameters = {'learning_rate':[0.0001, 0.001, 0.01, 0.1,0.2,10] , 'n_estimators': [10,50,75,100,150] }
train_scores,CV_scores,parameter_str=GridSearch(x_train,y_train,x_CV,y_CV,parameters)
```

We have used different activation functions, hidden layer sizes., Learning rates, Dense layers.

n_estimeter:100 and learning rate:0.2
train_scores:1.765594
CV_train=1.716439

- **Tabulate the performance metrics of each model.**

Performance of Each Model :

Model	MAE	RSME	Accuracy
ARIMA	1887	5684	96.42
LSTM	2291	5205	97.23

4 Phase 3

One of the goal is to provide insight and recommend few initiatives based on the sales predictions model from the previous phases and provide minimum of three inventory optimization initiatives and examine them on the data. Also deploy the model on a dashboard for visualization.

4.1 Deploy the machine learning model using Streamlit for dataset 02 (ARIMA)

Deployment Screen Recroding Video Link :
https://drive.google.com/file/d/1k31uX49_1LOU1Z13_WrmMYAs9HJXpL64/view?usp=sharing

4.2 Product segmentation based on demand variability (ABC Analysis)

The references that are driving most of the sales.

- Class A: Very Fast Movers: top 5%
- Class B: The following 15% of fast movers
- Class C: The remaining 80% of very slow movers Task

1. Use first year data of the Household category to create ABC Analysis and interpret the graph.

Product segmentation refers to the activity of grouping products that have similar characteristics and serve a similar market. It is usually related to marketing (Sales Categories) or manufacturing (Production Processes).

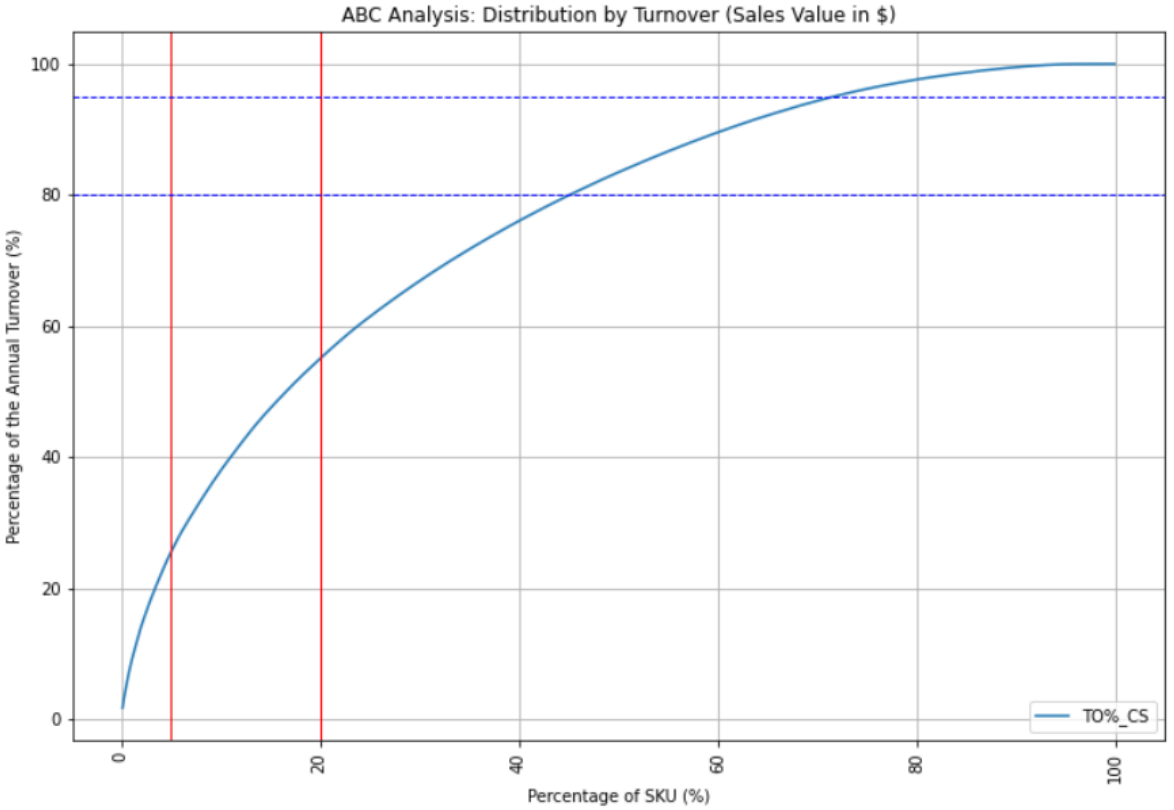
Mainly focused on the sales volumes distribution (fast/slow movers), demand variability and delivery lead time.
You want to put efforts into managing products that have:

- The highest contribution to your total turnover: ABC Analysis
- The most unstable demand: Demand Variability

Simple statistical tools to combine ABC Analysis and Demand Variability to perform product segmentation.

This analysis will be done for the SKU in the HOUSEHOLDS category.

ABC Analysis



Class A: the top 5%

- Number of SKU: 16

- Turnover (%): 25%

Class B: the following 15%

- Number of SKU: 48
- Turnover (%): 31%

Class C: the 80% slow movers

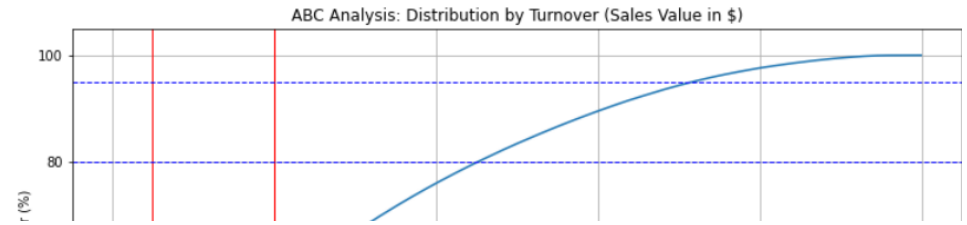
- Number of SKU: 253
- Turnover (%): 43%

In this example, we cannot clearly observe the Pareto Law **(20% of SKU making 80% of the turnover)**.

However, we still have 80% of our portfolio making less than 50% of the sales.

Code :

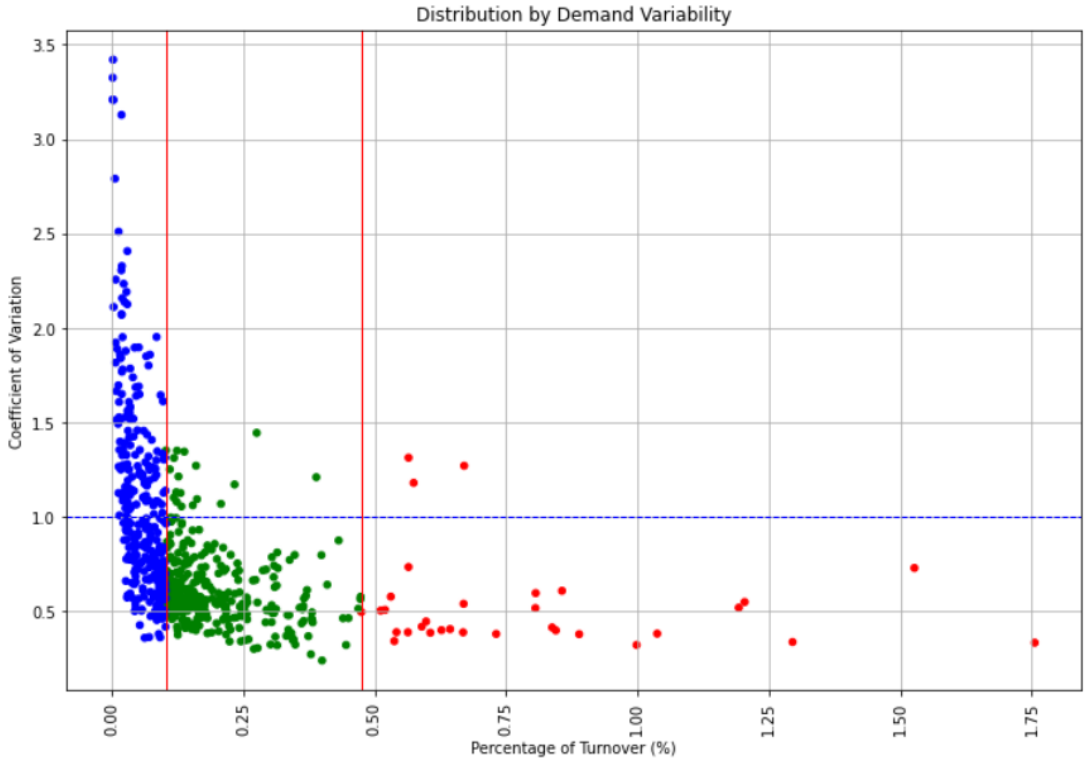
```
ax = plt.gca()

df_abc.plot(figsize=(12, 8), x='SKU_%', y='TO%_CS', ax =ax, grid = True)

ax.axvline(5 , color="red", linestyle="-", linewidth = 1.0)
ax.axvline(20 , color="red", linestyle="-", linewidth = 1.0)
ax.axhline(80 , color="blue", linestyle="--", linewidth = 1.0)
ax.axhline(95 , color="blue", linestyle="--", linewidth = 1.0)

plt.xlabel('Percentage of SKU (%)')
plt.xticks(rotation=90)
plt.ylabel('Percentage of the Annual Turnover (%)')
plt.title('ABC Analysis: Distribution by Turnover (Sales Value in $)')

plt.show()
```



2. How stable is the customers' demand? (Coefficient of Variation) To understand which products will bring planning and distribution challenges, compute the coefficient of variation of the yearly distribution of sales of each reference.

Demand Stability: Coefficient of Variation

In order to understand which products will bring planning and distribution challenges, we will compute the coefficient of variation of the yearly distribution of sales of each reference.



Code :

```
ax = plt.gca()

colors = {'A':'red', 'B':'green', 'C':'blue'}

df_plot = df_abc[df_abc['CV']<4].copy()
df_plot.plot.scatter(figsize=(12, 8), x='TO%', y='CV', color=df_plot['ABC'].map(colors), ax =ax, grid = True)

ax.axvline(to_a , color="red", linestyle="-", linewidth = 1.0)
ax.axvline(to_b , color="red", linestyle="-", linewidth = 1.0)
ax.axhline(1 , color="blue", linestyle="--", linewidth = 1.0)

plt.xlabel('Percentage of Turnover (%)')
plt.xticks(rotation=90)
plt.ylabel('Coefficient of Variation')
plt.title('Distribution by Demand Variability')

plt.show()
```

3. Discuss a few initiatives and recommendations for improving the retail business for dataset 02.
   Hints:

   (a) Provide more discount for locations which have low median income based on specific event date, for example, Christmas

   (b) Potential interpretation based on ABC Analysis and Demand variability.

- Inventory Optimization: This involves demand forecasting to better be able to predict what products can be maintained (especially that brings in more profits) and thereby avoid stockouts and improve customer satisfaction.
- Supplier Relations: By understanding where the products lie both regionally and temporarily (in time), it's easy to see which suppliers bring in profits and therefore maintain a good relationship and have an adequate supply of those products. Win-Win strategy.
- Customer Service: Maintaining sufficient stock to meet demand increases customer satisfaction. If the customer gets what they want at a reasonable price, they're happy and that's the key.
- Strategic Pricing: ABC analysis study suggests to alter/adjust the prices of the products as per demand in a better way and thereby maintain the inventory in an almost clean state.
- Better Resource Allocation: Continually evaluating the allocated resources, it ensures that the store have the much-needed items categorically prioritized inline with customer demand.

Conclusion

Our team worked on two datasets pertaining to Stores & features to identify the most important Retail store sales information. Our work includes analyzing the datasets for any anomalies, studying/understanding the graphs by creating different visualizations, build the neural network models, and lastly making public dashboards using Tableau. We were able to calculate sales weekly, monthly & also identify the top performing stores & products.

Next, we deployed various machine learning algorithms to predict sales for the stores and generate analysis based on weather, seasons, income of the people, etc.

Lastly, we created ABC analysis for the dataset and provided some recommendations for the same and deployed our model on to Streamlit. Also, not to mention, there is a huge scope for improvement with the accuracy of the models.

Acknowledgement

Throughout this project's problem-solving and reporting process, we received a great deal of invaluable insight and guidance from Dr. Allen Bolourchi. We would like to thank him for sharing his expertise with us. He utilized his private time throughout the semester to discuss with us how to approach the problems, providing his professional insights in the ML/DL/Time Series field, and giving us feedback to improve our findings and reporting skills. The project, to all of us, is huge and new. This is a tremendous learning experience to tackle an industry's real problems and we appreciate very much the opportunity to work with Dr. Bolourchi throughout this semester. Meantime, we thank Dr. Mahshid Fardadi for giving us this opportunity in her class to tackle this real-world industry problem, and for connecting us with Dr. Bolourchi. We thank her support and assistance during the project and reporting process in this semester