Store Item Sales Forecasting
Group – 4

Group 4:

Bikka Vijay, Abinesh Ganesan, Sridhar Hallyala, Sai Yeseswini Chadalawada

1. Introduction

The focus of the project is to help businesses to optimize store inventory management and supply chain planning by predicting demand for products that customers are expected to buy in the future in a specific duration, which would help anticipate consumer demand and prepare for any changes in consumer behavior. The project aims to forecast sales of items which could be leveraged to forecast future sales and help businesses optimize their inventory based on the predicted sales. This could further help businesses in preparing data driven and informed marketing strategies. Furthermore, it would help organizations in observing and analyzing trends, account for market conditions like inflation, competitors and make financial decisions accordingly.

Through this project, we predicted sales of a particular item by training models to forecast sales in the next 4 months in a particular store. The intent is to develop a framework that could be extrapolated to other items and store and train models to forecast demand.

1.1 Member Contribution

Each member of our team contributed to various sections of the analysis. However, we came up with the idea together after multiple discussions and project proposals. In addition, each member contributed equally to the presentation and report preparation. Below are the tasks performed by each member:

- Abinesh Ganesan: Data preparation, Exploratory data analysis, Stationary check
- Bikka Vijay: Exploratory data analysis, stationarize data, Performing and Analyzing ACF and PACF plots, Model Training, and evaluation
- Sridhar Hallyala: Exploratory data analysis, Model Training, Hyperparameter Tuning and evaluation.
- Sai Yeseswini Chadalawada: Exploratory data analysis, series decomposition, Analyzing trends.

2. Analysis

The following section describes the dataset, analysis and data preparation.

2.1 Dataset

The data is obtained from a Store Item Demand Forecasting Challenge from Kaggle (https://www.kaggle.com/competitions/demand-forecasting-kernels-only/data). As part of the challenge, 3 files are provided namely sample_submission.csv, train.csv and test.csv. We have used train.csv to train and test our models.

- The training data consists of 4 columns including date, store, item and sales.
- Data consists of information from January 1, 2013 to December 31, 2017.
- It consists of 10 unique stores and 50 unique items.
- There are 913K observations in the data.

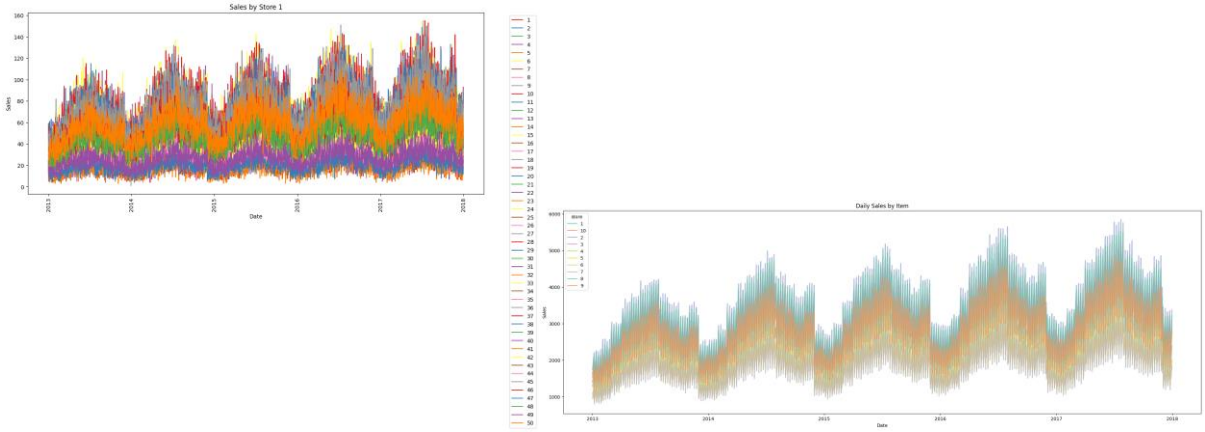
For this project, we selected store 1 and item 1 to train the model and develop a framework since the sales of items can vary significantly from each other and also vary according to the store. The framework can be leveraged to train more models for other items and stores.

2.2 Exploratory Data Analysis

Below exploratory data analysis is conducted to study trends in the data and prepare the data for training models.

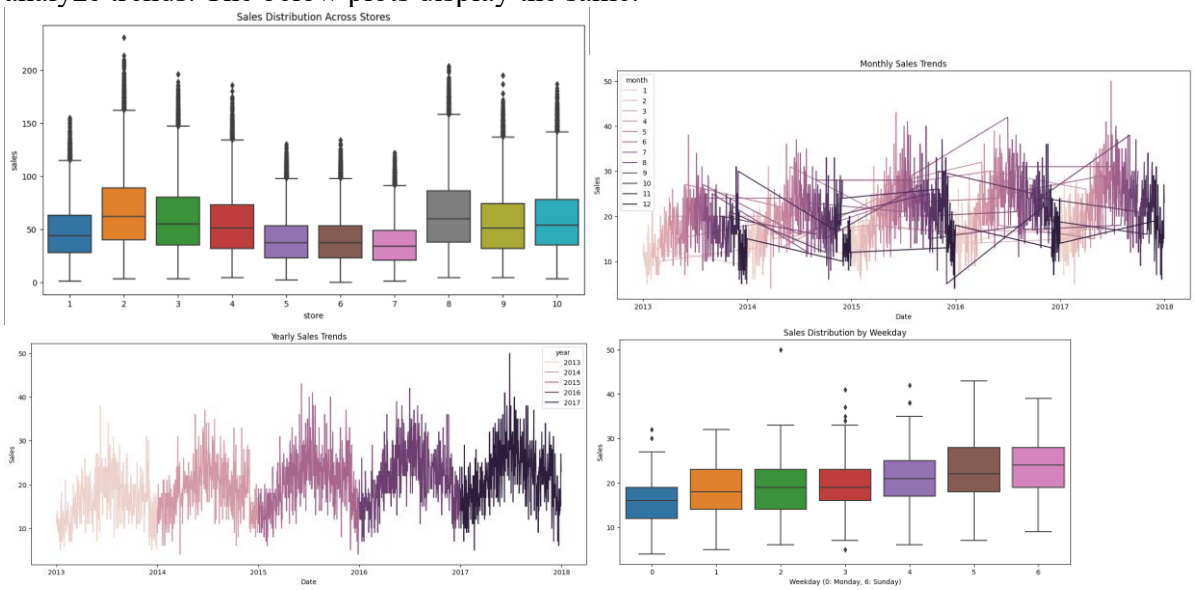
2.2.1 Visualizing Trends

We plotted item sales by stores and items, daily sales by store and daily sales by items as in the below graphs. It is observed that there is **seasonality, cyclic pattern and a slight increasing trend** in all the graphs.



2.2.2 Analyzing Trends over Time

The time series is further plotted by year, day of the week and over a period of 30 days to analyze trends. The below plots display the same.



Yearly Trend

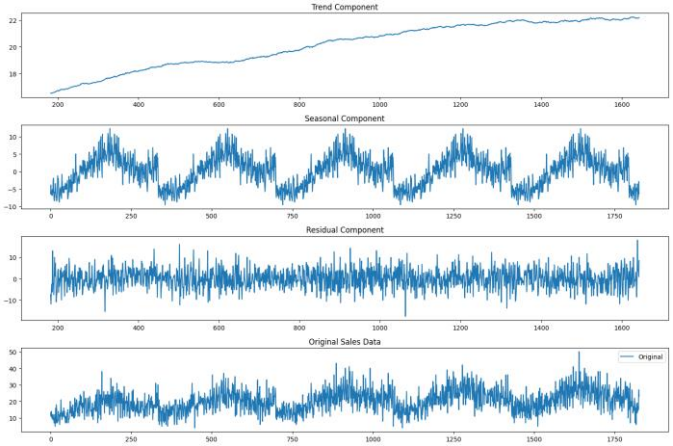
From the above graphs, it is observed that **there is a yearly repeating pattern** in the data (cyclicity). The sales increase every year in the middle of the year and decrease towards the year's end. The pattern is repeated from 2013-2018.

Monthly Trend

There is **no clear pattern** observed as observed in the second graph. There is no evident trend in the data over a period of 30 days in January 2013.

Weekday Trend
It is observed that the sales are higher on weekends in comparison to weekdays. As depicted in the graph, 0 represents Monday and 6 represents Sunday. In addition, there are outliers in the data observed on weekdays. A **slight increase in trend is observed over weekdays**.

2.2.3 Decomposing Series to Analyze Trends
The series is decomposed to visualize the above patterns more clearly. For the same reason, the series is decomposed into various components (trend, seasonality, and residuals).



The below observations are made based on the above plot:
1. The first subplot shows the distribution of sales, which is in sync with the plots discussed earlier.
2. The second subplot displays a clear increase in trend over time.
3. The third subplot shows a seasonal pattern observed as discussed earlier.
4. The fourth subplot shows how residuals, or the noise, are plotted over time. It can be observed that there is no evident pattern in this plot.

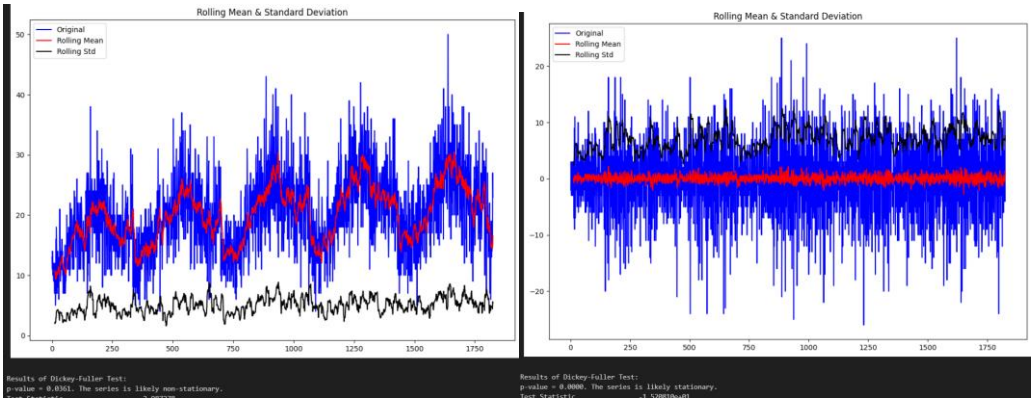
The above observations indicate that the series is **non-stationary.**

2.2.4 Dickey-Fuller Test
To further analysis stationarity of the series using statistical method, we used Dickey-Fuller test and observed the test statistic to make conclusions. Below are the results of the test. It can be observed that the value of test statistic (-2.99) is less than the Critical value at 5% (-2.86).

This indicates that the data is not stationary at a significance level of 5% or 0.05.
This indicates that further transformations must be applied to the data to make the series stationary and prepare data for model training.

2.2.5 First Order Differencing to make the series Stationary.
As concluded from above, since the series is non-stationary, the next step is to transform the series to stationary series. We applied differencing of order 1 on the series and obtained the following results on applying Dickey-Fuller test on the transformed series.

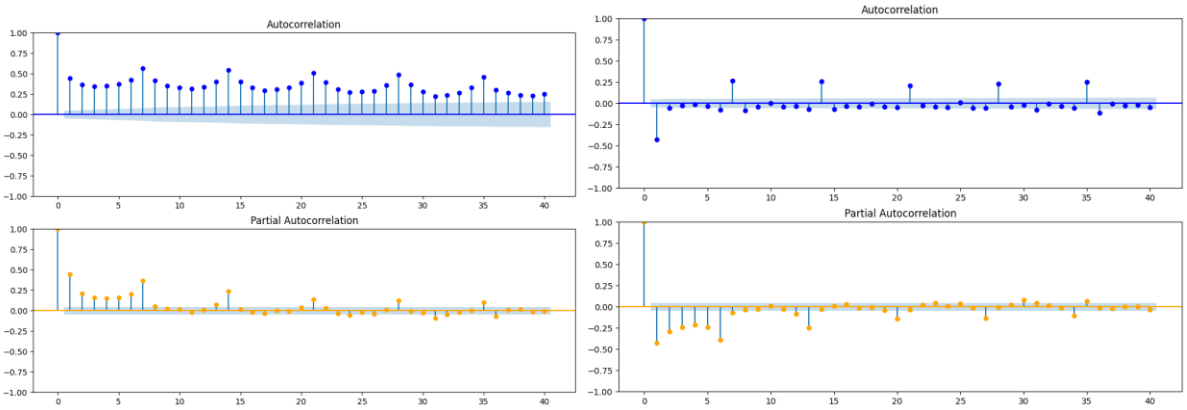
As observed the transformed series seems to be stationary at a significance level of 1% as the value of test statistic is greater than all the critical values observed. In addition, we plotted the original series and the transformed series as below. This further helps in visually analyzing that the transformed series is stationary.



2.2.6 ACF (Autocorrelation Function) and (Partial Autocorrelation Function)

What is ACF?
For example, if we anticipate today's stock price based on yesterday's stock price, the ACF will indicate how strongly these two variables are associated. Similarly, if we predict today's value based on yesterday's value, the ACF will indicate how strongly these variables are related and how many days are needed to predict today's value.

What is PACF?
Because today's value depends on yesterday's time, we must utilize the corelation of the day before yesterday when calculating the correlation between today and yesterday. Thus, PACF is utilized for this purpose.



Here we can see the acf and pacf both have a recurring pattern every 7 periods. Indicating a weekly pattern exists. So, we can suspect that there is some sort of significant seasonal thing going on. Hence, we consider SARIMA to take seasonality into account.

3. Modeling

3.1 ARIMA

ARIMA is short for Auto-Regressive Integrated Moving Average, which is a forecasting algorithm based on the assumption that previous values carry inherent information and can be used to predict future values. The ARIMA model takes in three parameters: p is the order of the AR term, q is the order of the MA term, d is the number of differencing.

Determining p, d, q

Since, we used first order differencing to make the time series stationary. I = 1. Also, within 6 lags the AR is significant. Which means, we can use AR = 6. MA is first tried then the MA order is being set to 0.

```
==============================================================================
Dep. Variable:                  sales   No. Observations:                 1826
Model:                 ARIMA(6, 1, 0)   Log Likelihood               -5597.679
Date:                Tue, 28 Nov 2023   AIC                          11209.359
Time:                        22:39:19   BIC                          11247.924
Sample:                             0   HQIC                         11223.585
                                 - 1826
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8174      0.021    -39.063      0.000      -0.858      -0.776
ar.L2         -0.7497      0.025    -30.480      0.000      -0.798      -0.702
ar.L3         -0.6900      0.026    -26.686      0.000      -0.741      -0.639
ar.L4         -0.6138      0.027    -22.743      0.000      -0.667      -0.561
ar.L5         -0.5247      0.025    -21.199      0.000      -0.573      -0.476
ar.L6         -0.3892      0.021    -18.819      0.000      -0.430      -0.349
sigma2        26.9896      0.817     33.037      0.000      25.388      28.591
===================================================================================
Ljung-Box (L1) (Q):                   1.41   Jarque-Bera (JB):                19.53
Prob(Q):                              0.23   Prob(JB):                         0.00
Heteroskedasticity (H):               1.41   Skew:                             0.15
Prob(H) (two-sided):                  0.00   Kurtosis:                         3.40
===================================================================================
```

3.1.2 Analyze the result

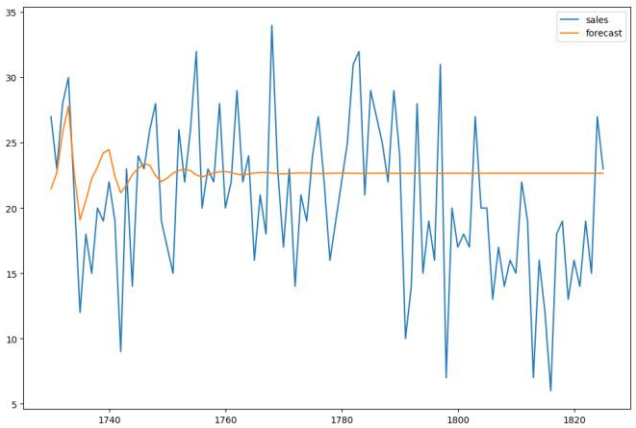
The normaltest function from the scipy.stats library is used to test the normality of the residual distribution. The results of the normal test are:

NormaltestResult(statistic=16.742690147718932, p value=0.00023140408872260922)

It failed the chi-squared statistic test, p-value is very small, meaning the residual is not a normal distribution.

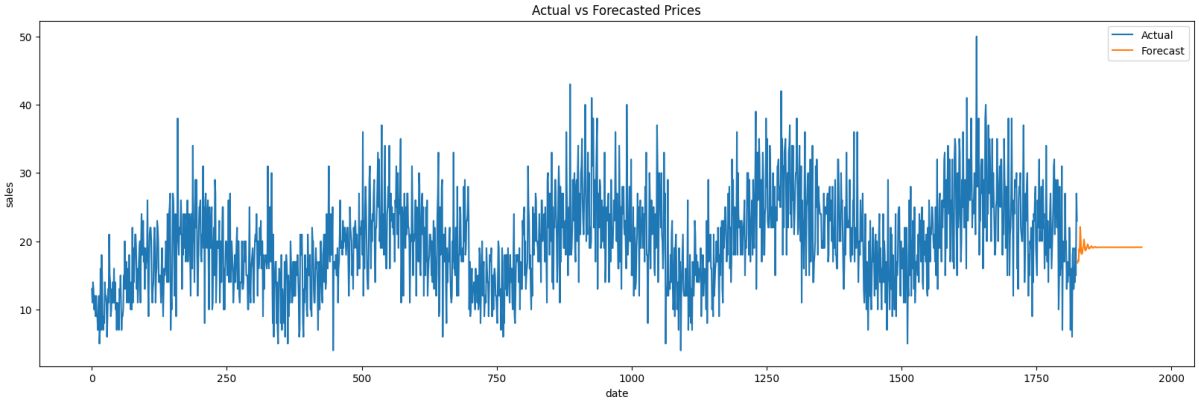
3.1.3 Prediction and Evaluation

Taking the last 30 days in the training set as validation data, we predict the sales. The prediction is not very accurate.



The error results for the ARIMA model are as follows:
MAPE: 33.01%
SMAPE: 25.07%
RMSE: 6.15
MSE: 37.81
MAE: 4.96

3.1.4 Forecasting Next 3 months sales



3.2 SARIMAX

SARIMAX is similar to ARIMA and stands for seasonal auto regressive integrated moving average with exogenous factors, and is used on data sets that have seasonal cycles. SARIMAX requires not only the p, d, and q arguments that ARIMA requires, but it also requires another set of P, D, and Q arguments for the seasonality aspect as well as an argument called “s” which is the periodicity of the data’s seasonal cycle.

The parameters are selected as**:**
p,d,q are 1
P = 7
D = 1
Q =6
s is considered 12, it specifies monthly data suggests a yearly seasonal cycle.

SARIMAX Results

Dep. Variable:					sales	No. Observations:	1826
Model:			SARIMAX(1, 1, 1)x(7, 1, [1, 2, 3, 4, 5, 6], 12)			Log Likelihood	-5495.934
Date:					Tue, 28 Nov 2023	AIC	11023.868
Time:					22:53:39	BIC	11111.911
Sample:					0	HQIC	11056.356
					- 1826		
Covariance Type:					opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0330	0.025	-1.340	0.180	-0.081	0.015
ma.L1	-0.8949	0.011	-78.772	0.000	-0.917	-0.873
ar.S.L12	-1.1909	0.077	-15.548	0.000	-1.341	-1.041
ar.S.L24	-0.6121	0.156	-3.911	0.000	-0.919	-0.305
ar.S.L36	0.3791	0.198	1.919	0.055	-0.008	0.766
ar.S.L48	1.0263	0.169	6.076	0.000	0.695	1.357
ar.S.L60	0.8412	0.097	8.678	0.000	0.651	1.031
ar.S.L72	-0.0257	0.045	-0.572	0.567	-0.114	0.062
ar.S.L84	0.0552	0.029	1.908	0.056	-0.002	0.112
ma.S.L12	0.2272	0.073	3.120	0.002	0.085	0.370
ma.S.L24	-0.5921	0.124	-4.780	0.000	-0.835	-0.349
ma.S.L36	-1.0498	0.076	-13.874	0.000	-1.198	-0.902
ma.S.L48	-0.6755	0.168	-4.009	0.000	-1.006	-0.345
ma.S.L60	0.1889	0.084	2.252	0.024	0.025	0.353
ma.S.L72	0.9086	0.128	7.074	0.000	0.657	1.160
sigma2	21.9937	2.390	9.202	0.000	17.309	26.678

Ljung-Box (L1) (Q):	0.12	Jarque-Bera (JB):	15.31
Prob(Q):	0.73	Prob(JB):	0.00
Heteroskedasticity (H):	1.34	Skew:	0.15
Prob(H) (two-sided):	0.00	Kurtosis:	3.34

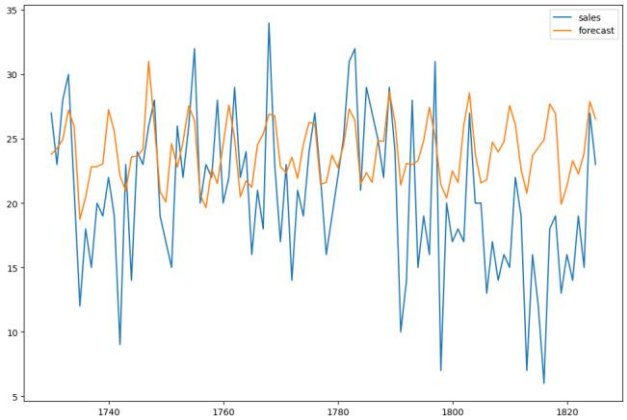
3.2.1 Analyze the result

The normaltest function from the scipy.stats library is used to test the normality of the residual distribution. The results of the normal test are:

NormaltestResult(statistic=13.640194651882004, pvalue=0.0010916146746838464)

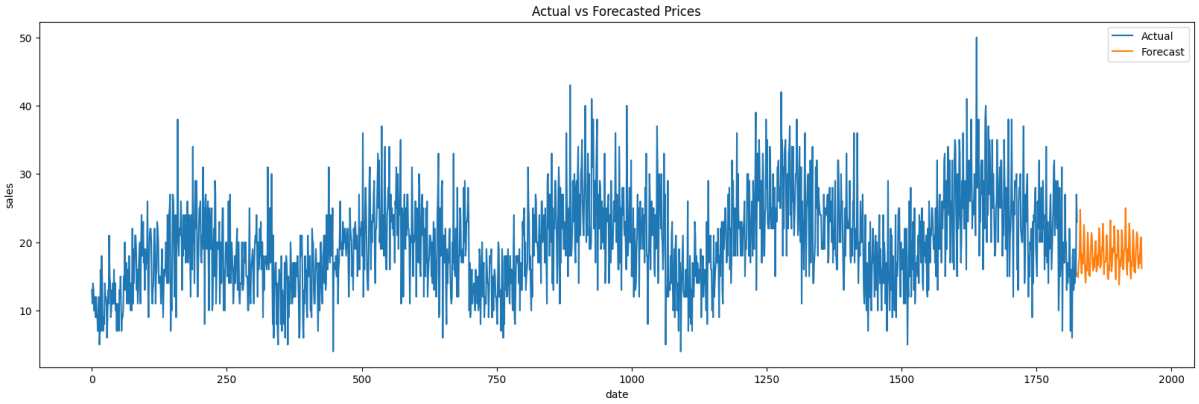
3.2.3 Prediction and Evaluation

Taking the last 30 days in training set as validation data, we predict the sales. This better captures the pattern in sales rather than generalising into a straight line like ARIMA.



The error results for the SARIMAX model are as follows:
MAPE: 34.94 %
SMAPE: 25.78 %
RMSE: 6.40
MSE: 40.93
MAE: 5.17

3.2.4 Forecasting Next 3 months sales



4. Model Tuning

4.1 Hyperparameters in ARIMA model

The ARIMA (Autoregressive Integrated Moving Average) model has three key hyperparameters:
1. p: The order of the autoregressive (AR) component of the model, which is the number of lags of the dependent variable included in the model. A higher value of p means that the model considers more past values of the dependent variable.
2. d: The degree of differencing required to make the time series stationary, which is the number of times the data needs to be differenced to make it stationary. Stationary data has constant mean and variance over time.
3. q: The order of the moving average (MA) component of the model, which is the number of lagged forecast errors included in the model. A higher value of q means that the model takes

into account more past errors in forecasting the current value.

4.2 Hyperparameters in SARIMAX model

The key hyperparameters in a SARIMAX model include:

1. p, d, and q: These hyperparameters are the same as in the ARIMA model and specify the order of the autoregressive, differencing, and moving average components of the model, respectively.
2. P, D, and Q: These hyperparameters specify the seasonal order of the autoregressive, differencing, and moving average components of the model, respectively. Seasonality refers to patterns that repeat over fixed time intervals, such as daily, weekly, or yearly cycles.
3. m: The number of time steps in a single seasonal period. This hyperparameter specifies the frequency of the seasonality in the data, such as 12 for monthly data or 7 for weekly data.
4. Exogenous variables: SARIMAX models can include exogenous variables, which are external factors that may influence the time series data. Hyperparameters related to exogenous variables include the lag order of the exogenous variables and the regularization strength.

4.3 Hyperparameter tuning ARIMA

4.3.1 Model tuning

We used the auto_arima function from the pmdarima library to automatically tune the hyperparameters for an ARIMA model.

The auto_arima_params dictionary defines the ranges for the hyperparameters that auto_arima will search over. The start_p, start_d, and start_q parameters set the minimum values for the autoregressive, differencing, and moving average orders, respectively. The max_p, max_d, and max_q parameters set the maximum values for these orders.

The seasonal parameter is set to False because the model being fitted is not a seasonal ARIMA model. The stepwise parameter is set to True to enable a more efficient search algorithm. The suppress_warnings parameter is set to True to suppress warning messages that may arise during the fitting process. The error action parameter is set to 'ignore' to ignore any errors that may occur during the fitting process.

The auto_arima function is then called with the training data and the hyperparameter ranges specified in auto_arima_params. This function performs an exhaustive search over the specified hyperparameter ranges to find the best-performing model based on a given metric, such as AIC or BIC.

Finally, the ARIMA model is fitted using the optimal hyperparameters found by auto_arima, and a summary of the model is printed using the summary() function.

Dep. Variable:		sales		No. Observations:		1826
Model:		ARIMA(7, 1, 1)		Log Likelihood		-5572.990
Date:		Tue, 28 Nov 2023		AIC		11163.980
Time:		23:41:18		BIC		11213.564
Sample:		0		HQIC		11182.271
		- 1826				
Covariance Type:		opg				
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0448	0.037	1.215	0.224	-0.027	0.117
ar.L2	-0.0597	0.032	-1.841	0.066	-0.123	0.004
ar.L3	-0.0665	0.032	-2.072	0.038	-0.129	-0.004
ar.L4	-0.0488	0.032	-1.508	0.131	-0.112	0.015
ar.L5	-0.0317	0.031	-1.022	0.307	-0.092	0.029
ar.L6	0.0199	0.029	0.684	0.494	-0.037	0.077
ar.L7	0.2757	0.028	9.861	0.000	0.221	0.331
ma.L1	-0.9143	0.025	-36.138	0.000	-0.964	-0.865
sigma2	26.2645	0.811	32.372	0.000	24.674	27.855
Ljung-Box (L1) (Q):		0.00	Jarque-Bera (JB):		14.77	
Prob(Q):		0.99	Prob(JB):		0.00	
Heteroskedasticity (H):		1.41	Skew:		0.14	
Prob(H) (two-sided):		0.00	Kurtosis:		3.33	

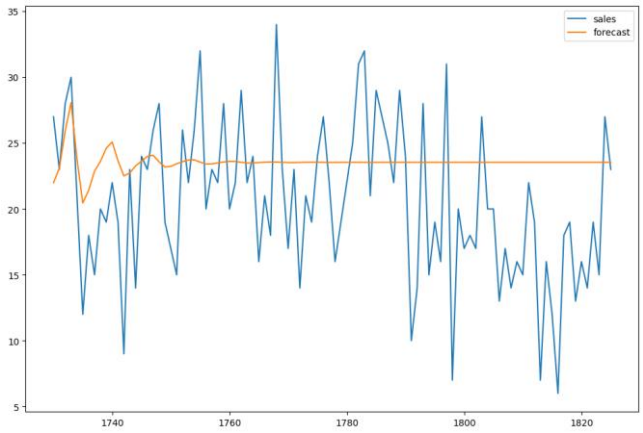
4.3.2 Plotting Residual Distribution

The normaltest function from the scipy.stats library is used to test the normality of the residual distribution. The results of the normal test are:

NormaltestResult(statistic=13.425420916365283, pvalue=0.0012153654589205209)

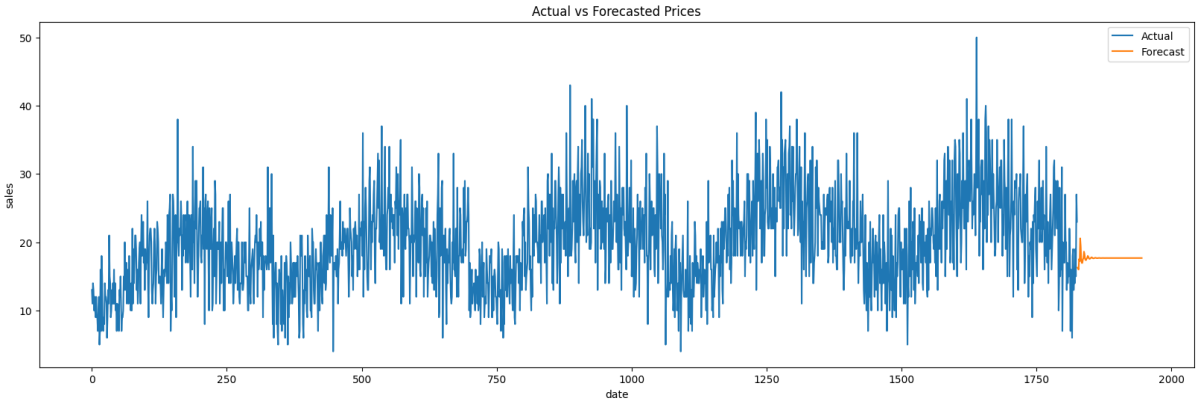
4.3.3 Prediction and Evaluation

The error results for the hypertuned model are calculated and they are as follows:
MAPE: 35.83%
SMAPE: 26.46%
RMSE: 6.52
MSE: 42.56
MAE: 5.30



We generated a forecast of the time series using an ARIMA model fit to the training data. The forecast is then plotted along with the actual values.

4.3.4 Forecasting next 3 months of sales



4.4 Hyperparameter tuning SARIMAX

4.4.1 Model tuning

We used the auto_arima function from the pmdarima library to tune the hyperparameters of a seasonal SARIMAX model for a time series. The function searches for the optimal hyperparameters using a grid search approach with a maximum number of iterations specified by the n_fits parameter. The range of hyperparameters to search over is specified using the various start_ and max_ parameters. The error_action parameter is set to 'ignore' to ignore any warnings or errors that may occur during the hyperparameter search. Once the optimal hyperparameters are found, the SARIMAX model is fit to the training data using the SARIMAX and fit functions from the statsmodels.tsa.statespace module.

SARIMAX Results

Dep. Variable:	sales	No. Observations:	1826
Model:	SARIMAX(2, 1, 1)x(2, 0, [], 12)	Log Likelihood	-5647.306
Date:	Tue, 28 Nov 2023	AIC	11306.613
Time:	23:51:02	BIC	11339.669
Sample:	0	HQIC	11318.807
	- 1826		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.0327	0.027	1.232	0.218	-0.019	0.085
ar.L2	-0.0722	0.026	-2.831	0.005	-0.122	-0.022
ma.L1	-0.8988	0.013	-69.966	0.000	-0.924	-0.874
ar.S.L12	-0.0617	0.023	-2.645	0.008	-0.108	-0.016
ar.S.L24	-0.1201	0.024	-5.096	0.000	-0.166	-0.074
sigma2	28.4964	0.872	32.696	0.000	26.788	30.205

Ljung-Box (L1) (Q):	0.07	Jarque-Bera (JB):	16.13
Prob(Q):	0.80	Prob(JB):	0.00
Heteroskedasticity (H):	1.40	Skew:	0.14
Prob(H) (two-sided):	0.00	Kurtosis:	3.37

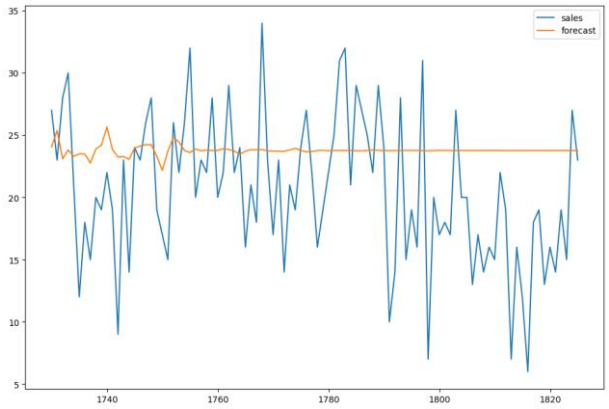
4.4.2 Plotting Residual Distribution

The normaltest function from the scipy.stats library is used to test the normality of the residual distribution. The results of the normal test are:

NormaltestResult(statistic=13.917196665537606, pvalue=0.0009504278248756773)

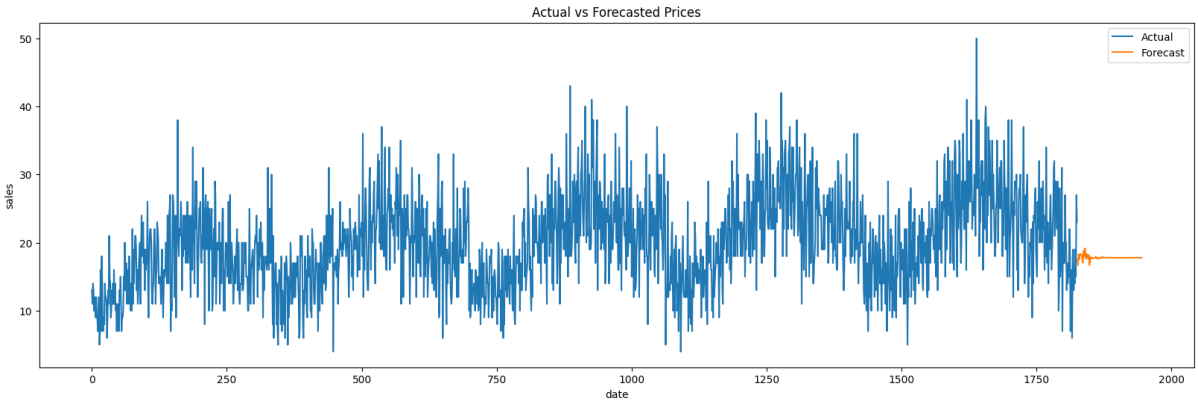
4.4.3 Prediction and evaluation

The error results for the hypertuned model are calculated and they are as follows:
MAPE: 37.12 %
SMAPE: 27.25 %
RMSE: 6.71
MSE: 45.05
MAE: 5.50



We generated a forecast of the time series using a SARIMAX model fit to the training data. The forecast is then plotted along with the actual values.

4.6.4 Forecasting next 3 months of sales



5. Model comparison

The models are compared to check and compare their performance on the basis of Log Likelihood, AIC, BIC and HQIC.

Model	Parameters	Log likelihood	AIC	BIC	HQIC
ARIMA	(6, 1, 0)	-5597.679	11209.359	11247.924	11223.585
SARIMAX	(1, 1, 1) x (7, 1, 6, 12)	-5495.319	11022.639	11110.682	11055.127
Tuned ARIMA	(7, 1, 1)	-5572.990	11163.980	11213.564	11182.271
Tuned SARIMAX	(2, 1, 1)x(2, 0, 6, 12)	-5647.306	11306.613	11339.669	11339.669

6. Conclusion

The SARIMAX(1,1,1)x(7,1,6,12) model has the highest log-likelihood (-5495.319), hence is a better fit. It also has the lowest AIC in comparison to other models.

Summing up, we trained different models ARIMA, and SARIMAX and also performed hyperparameter tuning on them. From our analysis, we conclude that SARIMAX (1,1,1)x(7,1,6,12) is the champion model which can be leveraged to forecast sales for the next 4 months.

Scope: Similarly, we can train models to forecast sales of other stores and items.