PRESENTATION

ROSSMAN STORE

P233351 QIU YI

01 INTRODUCTION

Background:
Rossmann operates over 3,000 drug stores in 7 European countries. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality.

1 The reasons we chose the dataset
This dataset are based on time series sales, the influence of other factors, and models supported by common sense (discounts and events promote sales, and sales can be evaluated each year based on the month of last year).
2 Problem formulation- Perform extensive Time Series Analysis (seasonal decomposition, trends, autocorrelation).

- Applying time series to try to find out any trends or patterns about sales, especially seasonal patterns,
- we can use the Arima model based on ACF and PACF result,
- Finally make a linear regression forecast of sales.

02 ARIMA MODEL -PRINCIPAL

ARIMA- WHAT?

- ARIMA models are a class of linear models that is capable of representing stationary as well as non-stationary time series.
- ARIMA models do not involve independent variables in their construction. They make use of the information in the series itself to generate forecasts.
- ARIMA models rely heavily on autocorrelation patterns in the data.

ARIMA - IS IT SUITABLE?

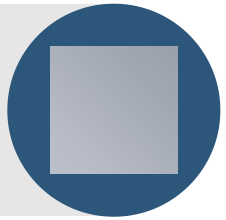
- Do the data exhibit a discernible pattern?
  - Any trend? Pattern?
- Can this be exploited to make meaningful forecasts?
- Is the time series stationary?

Contoso

- Model identification

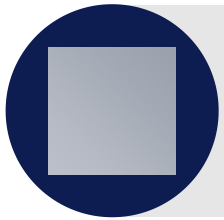
Autoregressive Moving Average (ARMA) models.

AR



- If ACF decay or die down to zero and PACF cut off to zero, AR model is suitable.
- The order of the AR is determined by the number of significant PACF values.

- A time series model which uses past errors as explanatory variable
- If PACF decay of die down to zero gradually and ACF cut off to zero, MA model is suitable.
- The order of MA is determined by the number of significant ACF.



MA

Contoso

WORKING ON PYTHON

GUIDELINE IN PYTHON

DATA PROCESSING

- Convert date format
- Add new feature(SPC)
- Abnormal data processing

EDA

- ECDF diagrams
- Mean-median for Sales and Customers
- Storetype-A analysis

('StoreType', 'Sales', 'Customers', 'PromoOpen', 'CompetitionOpen')

ARIMA MODEL

- seaborn heatmap
- Yearly trend by week
- Stationarize the data
- ACF and PACF charts
- 1st difference
- Arima model

LINEAR REGRESSION

- *seaborn* heatmap
- *Fit linear regression model*
- Evalation with result.rsquared
- Test (RMSE)

Contoso

01 DATASET

Size: (1017209, 8) for Train dataset

- Sales: the turnover for any given day .
- Customers: the number of customers on a given day.
- Open: 0 = closed, 1 = open.
- Promo: indicates whether a store is running a promo.
- StateHoliday: indicates a state holiday.
- SchoolHoliday: indicates if the (Store, Date) was affected by the closure of public schools.

Data processing

Convert date format : Year/ Month/Day/ Week

Add new feature(SPC): Sales Per Customer

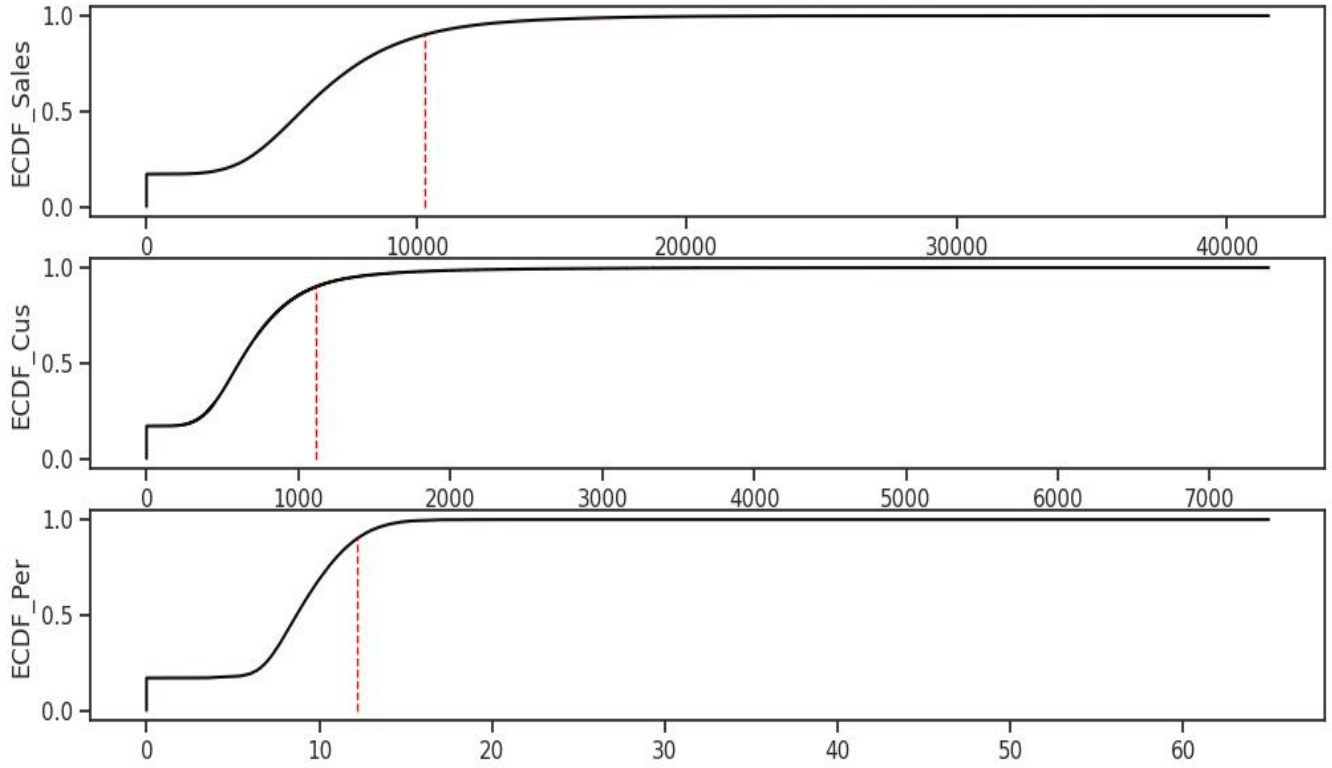
Abnormal data removing: (['Sales']==0) & (train['Open']==0), nearly 17%)

Merged the train data with store informations, we got (844338, 22) shape file

Split data into train and test 80:20

02 ECDF DIAGRAMS

Empirical Cumulative Distribution Function (CDF) Plots



Mean and Median for sales and Customers

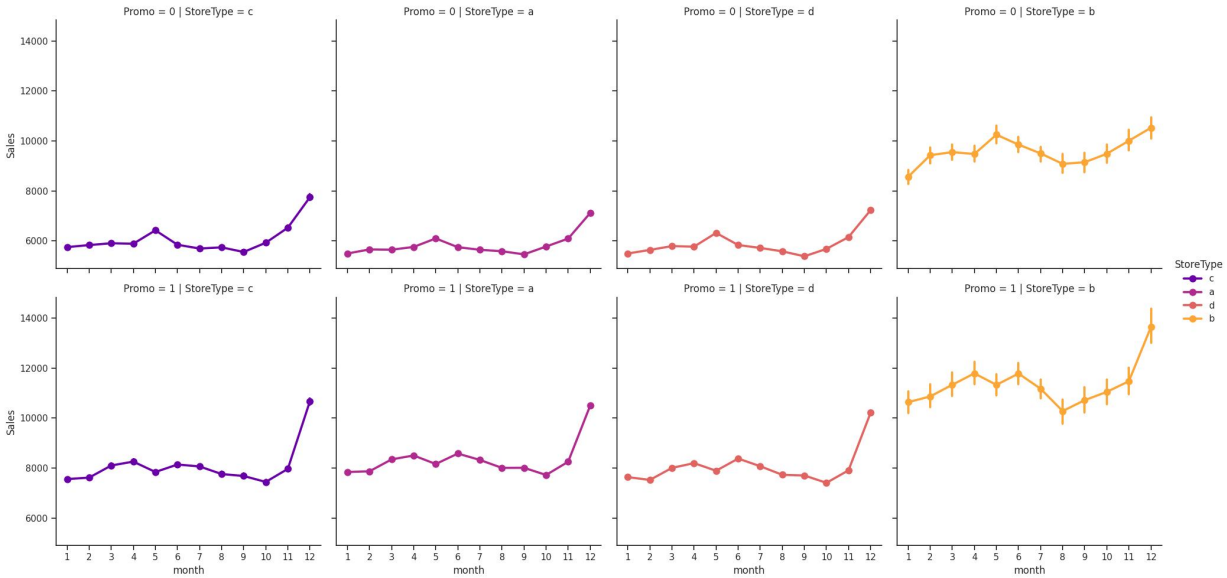
	Sales	Customers
Mean	6955.96	762.78
Median	6369.0	676.0

ECDF diagrams usually appear as a stepped polyline. Each step corresponds to an observation in the data set (or all observations in an interval). The line segments rise step by step from left to right, and the height indicates what proportion of the data points fall at this value or smaller at the current x value.

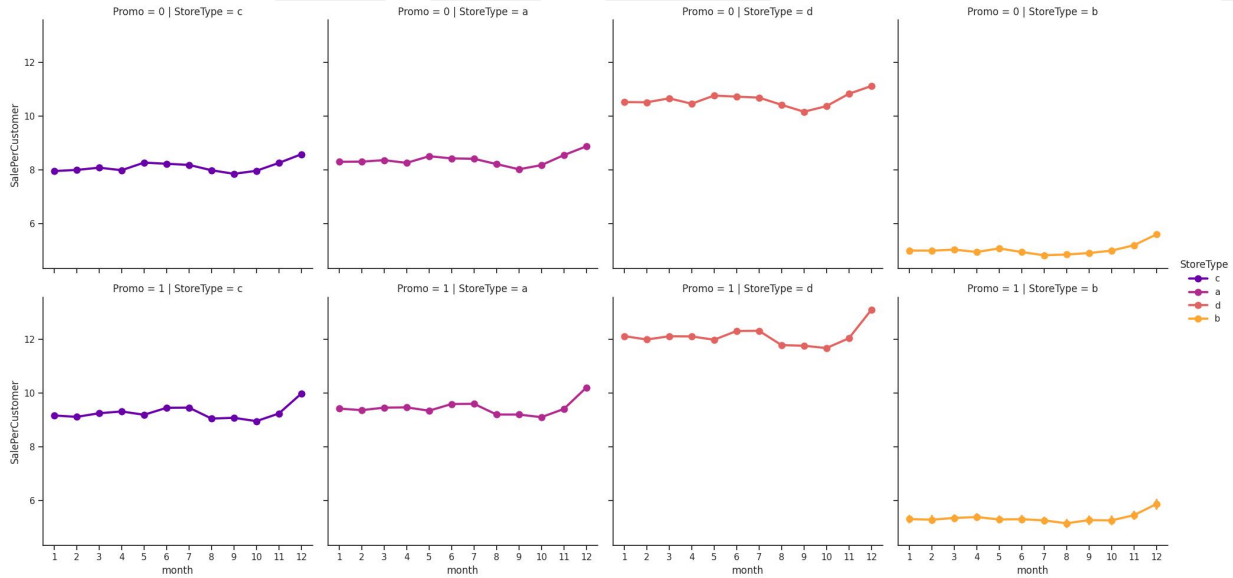
Contoso

STORETYPE-A ANALYSIS

Sales by month



SPC by month

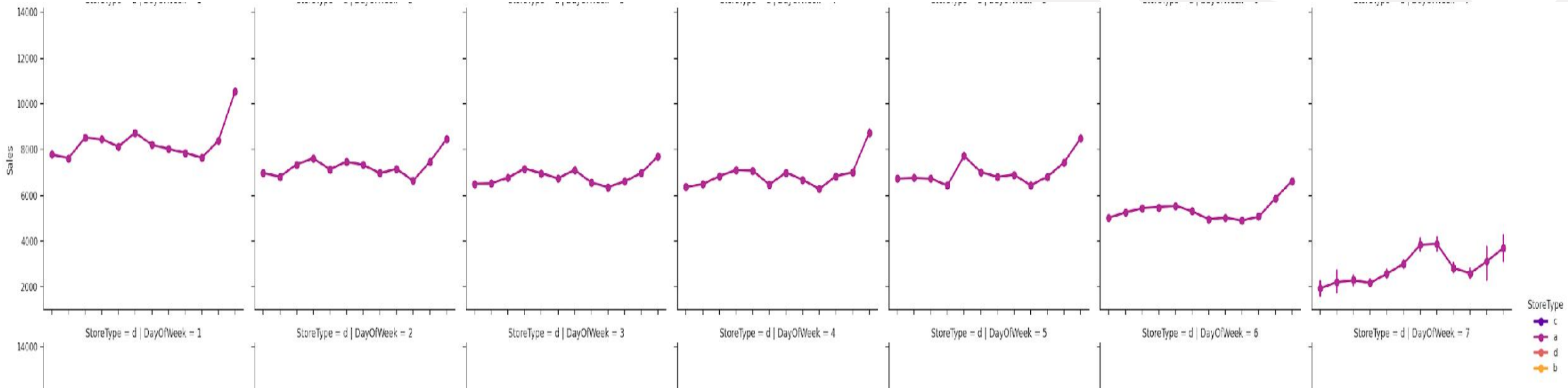


- All store types follow the same trend but at different scales depending on the presence of the (first) promotion Promo and Store Type itself
  Type B is small in number but has a high average of customers and sales.
- The highest SalePerCustomer amount is observed at the Store Type D not B,

STORETYPE-A ANALYSIS- BY DAY OF WEEK

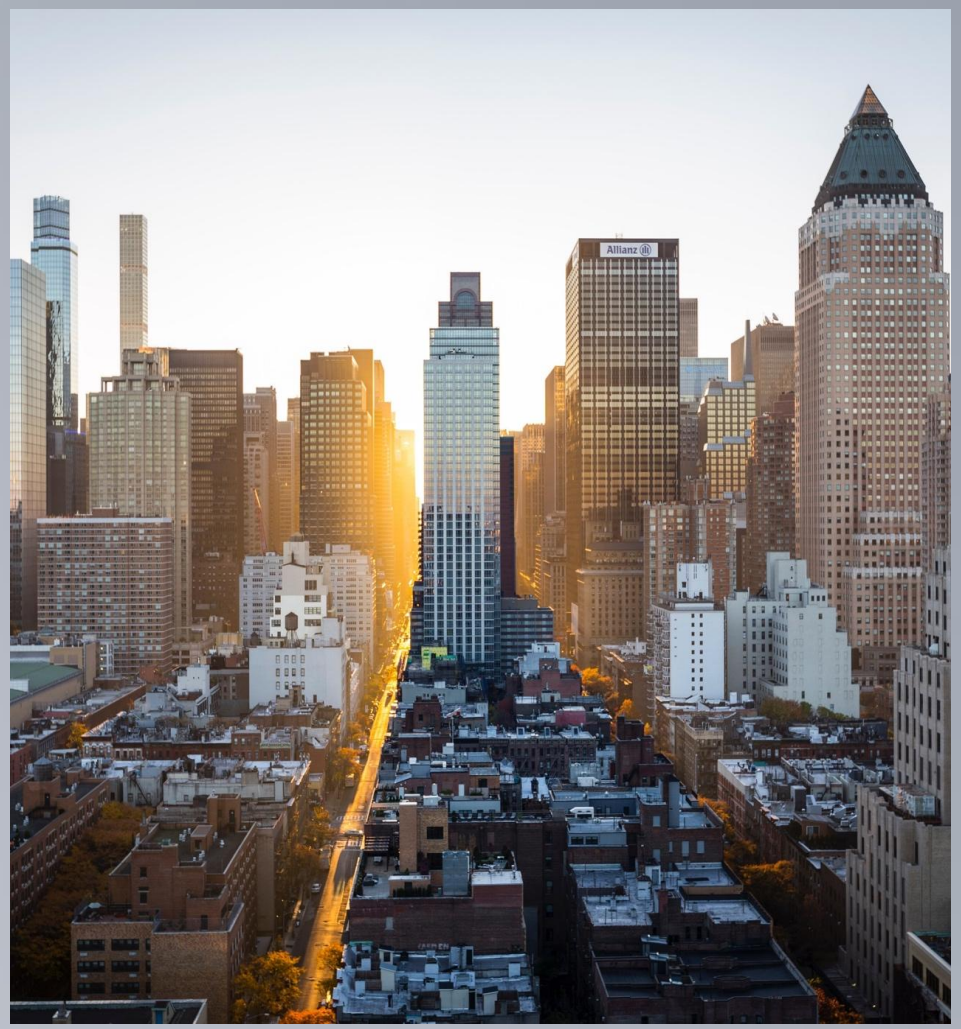
StoretypeC and A follow same trend

High sale on Monday , lowerest on Sunday, highest on DEC.



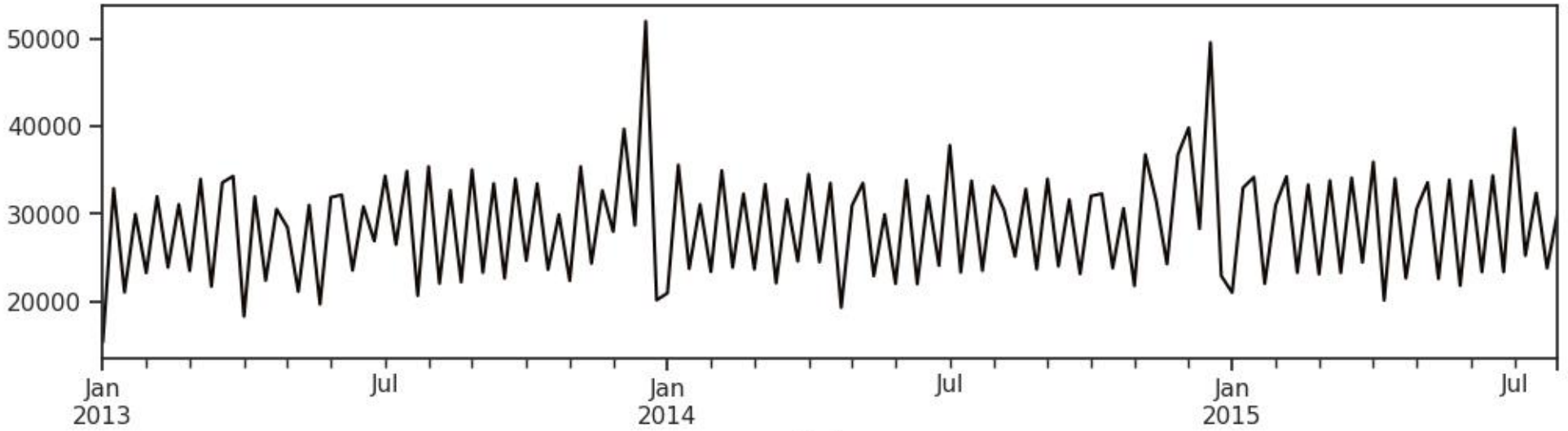
03 Arima model

- Seaborn heatmap for Correlation
- Yearly trend by week
- Stationarize the data
- Plot the ACF and PACF charts
- 1st difference
- Arima model

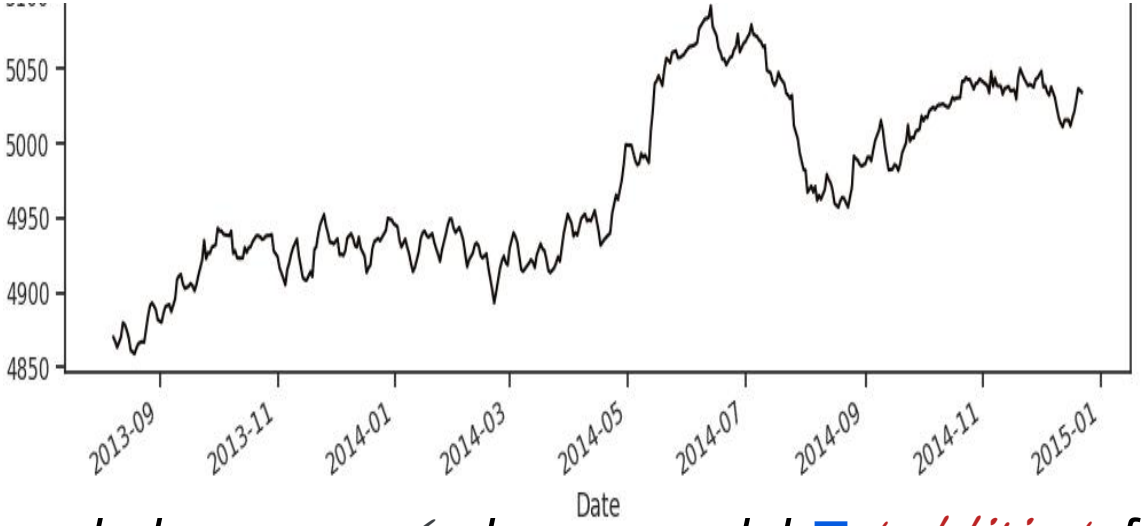


Contoso

TREND CHART-Seasonality



- It always peaks at Christmas time;
- After eliminating seasonal eff ects, we can see that the overall development trend of this store is on the rise

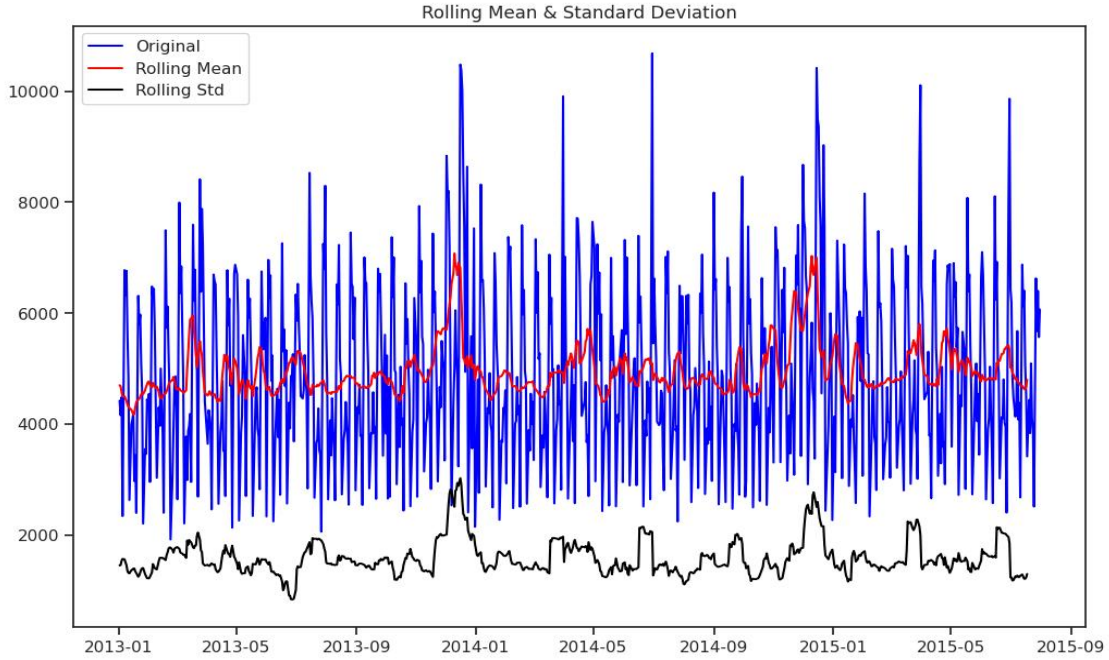


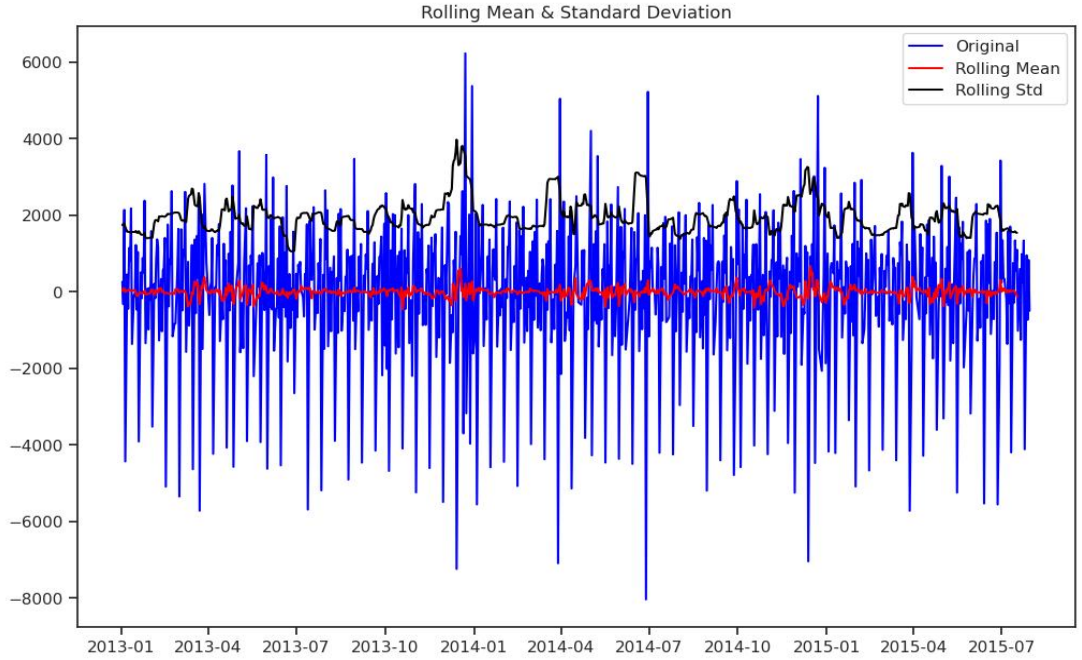

```
decomposition_a = seasonal_decompose(sales_a, model = 'additive', freq = 365)
```

TREND CHART-Rolling Mean & Standard Deviation

Dickey-Fuller Test for original data:
p-value = 0.0000. The series is likely stationary.
Test Statistic -5.292708
p-value 0.000006
#Lags Used 17.000000
Number of Observations Used 766.000000
Critical Value (1%) -3.438916
Critical Value (5%) -2.865321
Critical Value (10%) -2.568783

Results of Dickey-Fuller Test with 1st difference
p-value = 0.0000. The series is likely stationary.
Test Statistic -1.219964e+01
p-value 1.231658e-22
#Lags Used 1.600000e+01
Number of Observations Used 766
Critical Value (1%) -3.438916e+00
Critical Value (5%) -2.865321e+00
Critical Value (10%) -2.568783e+00





Remark:

statsmodels.tsa.seasonal.seasonal_decompose(x, model='additive', filt=None, period=None, two_sided=True, extrapolate_trend=0)

model='additive' means to use an additive model for decomposition.
Additive modeling is a method of decomposing a time series ***into trend, seasonality, and residuals.***
In an additive model, each component is obtained by linear addition of the original sequence, so they are independent of each other.

trend = decomposition.trend
seasonality = decomposition.seasonal
residual = decomposition.resid

In statistics, the ***Dickey–Fuller test*** *tests the null hypothesis that a unit root is present in an autoregressive (AR) time series model. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity.*

Remark:

The Augmented Dickey-Fuller test for stationarity on a given time series. The test returns several statistics and p-values that can be used to analyze the stationarity of the time series. Here is an explanation of the key results and how to interpret them:

1. adf: This is the test statistic of the Augmented Dickey-Fuller test. **The more negative the value, the more likely it is that the time series is stationary.**

2. pvalue: This is the p-value associated with the test statistic. **if the p-value is below the significance level, the time series can be considered stationary.**

3. critvalues: This is a dictionary of critical values for different significance levels. The keys in the dictionary represent the significance levels (e.g., 1%, 5%, 10%), The AIC and BIC values are reported in the SARIMAX results as follows:

AIC: The AIC value for the model is 13421.593. The AIC is calculated as -2 * log-likelihood + 2 * number of parameters. **It penalizes models with more parameters, so a lower AIC value indicates a better fit.** In this case, the AIC value suggests that the ARIMA(11, 1, 0) model is a good fit for the data.

BIC: The BIC value for the model is 13477.551. The BIC is similar to the AIC but penalizes models with more parameters more heavily. It is calculated as -2 * log-likelihood + log(number of observations) * number of parameters. **Like the AIC, a lower BIC value indicates a better fit.** In this case, the BIC value also suggests that the ARIMA(11, 1, 0) model is a good fit for the data.

The reason for comparing AIC and BIC values is to s**trike a balance between model fit and complexity.** A model with a better fit to the data will have a lower AIC and BIC value. However, including more parameters in the model can lead to overfitting, where the model captures noise or random fluctuations in the data instead of the underlying patterns. **The AIC and BIC penalize models with more parameters, encouraging the selection of simpler models that still provide a good fit to the data.**

Difference

Electronic Product

V1-Y

V2-1st difference

V3-2nd difference

V4-3nd difference

Advertisment

Newspaper

On TV

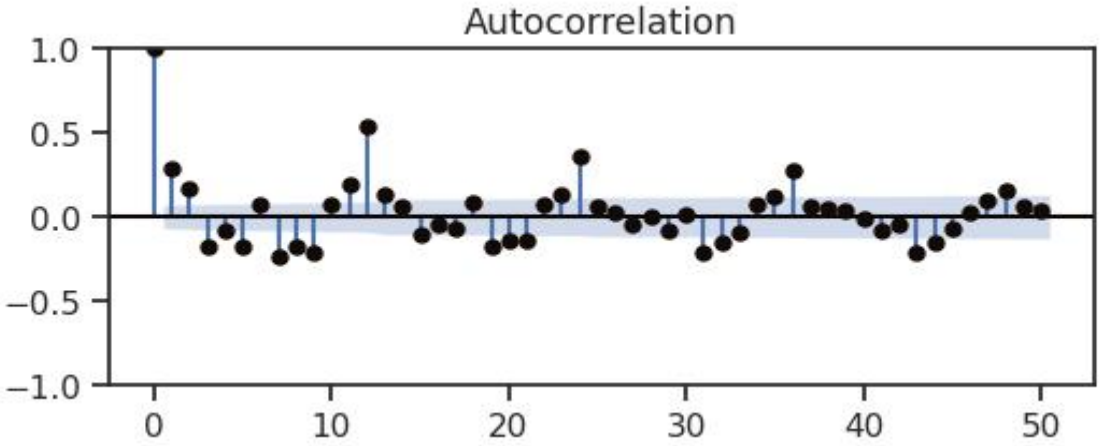
On cellphone

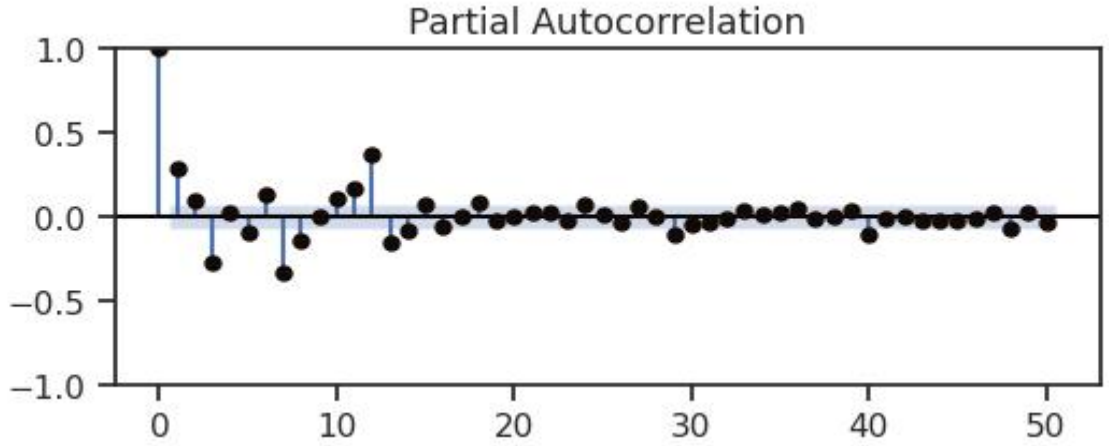
Short Video

The differenced series is defined as: yt(d)=yt-yt-1

Contoso

Plot the ACF and PACF charts and find the optimal parameters

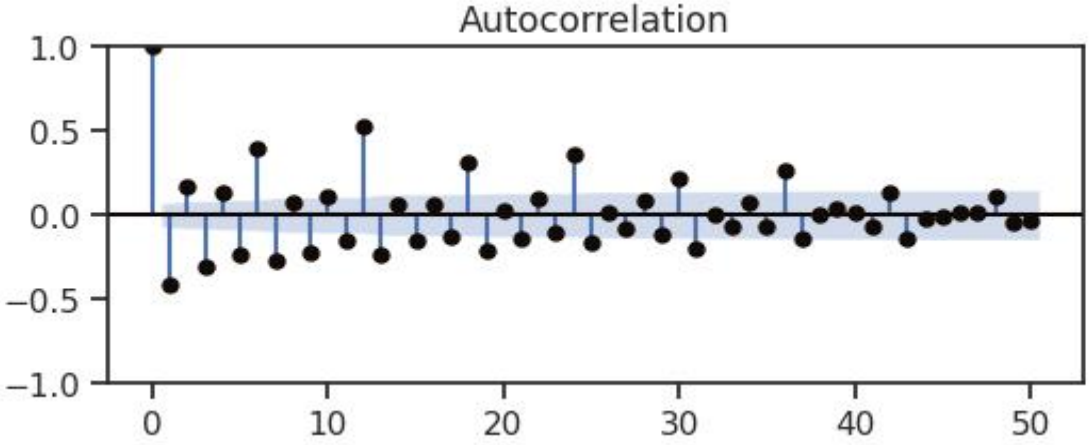


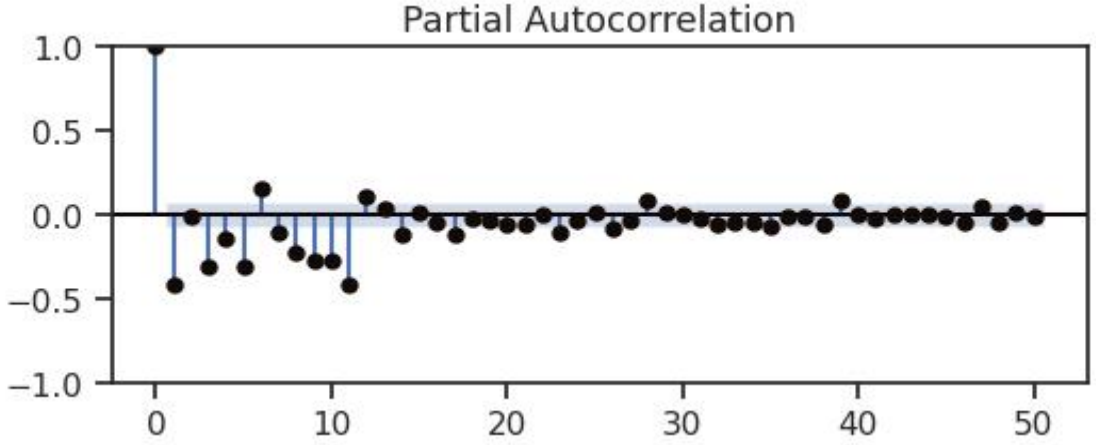


non randomnes of the time series and high lag-1 (which will probably need a higher order of differencing d/D).

Type A shows seasonalities at certain lags. It is each 12th observation with positives spikes at the 12 (s) and 24(2s) lags and so on.

Below is chart based on 1st difference





Arima VS Sarima

2 models are suitable for non-seasonal time series data, while SARIMA models are suitable for time series data that exhibit seasonal patterns. SARIMA models include additional parameters to capture the seasonal component, such as the seasonal order (p, d, q, P, D, Q, s).

P: The seasonal autoregressive order, which represents the number of lagged observations at the seasonal frequency.

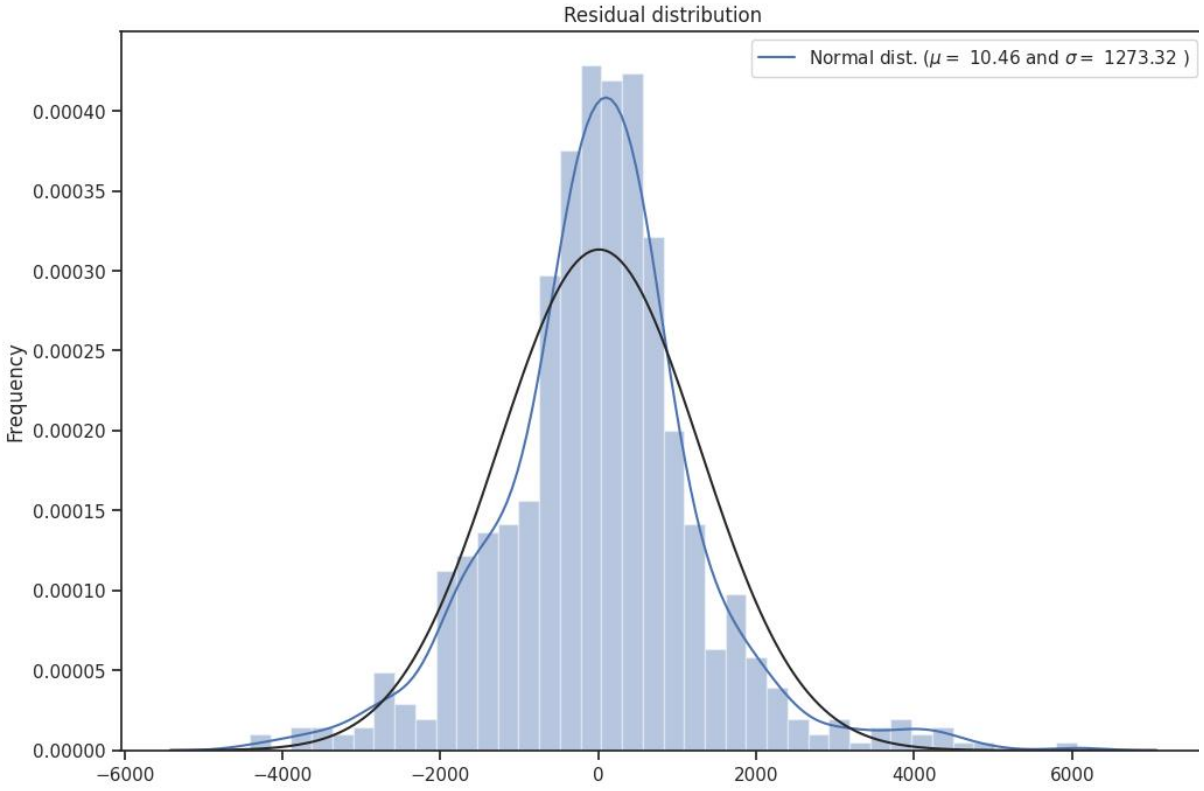
D: The seasonal differencing order, which represents the number of times the seasonal time series needs to be differenced to achieve stationarity.

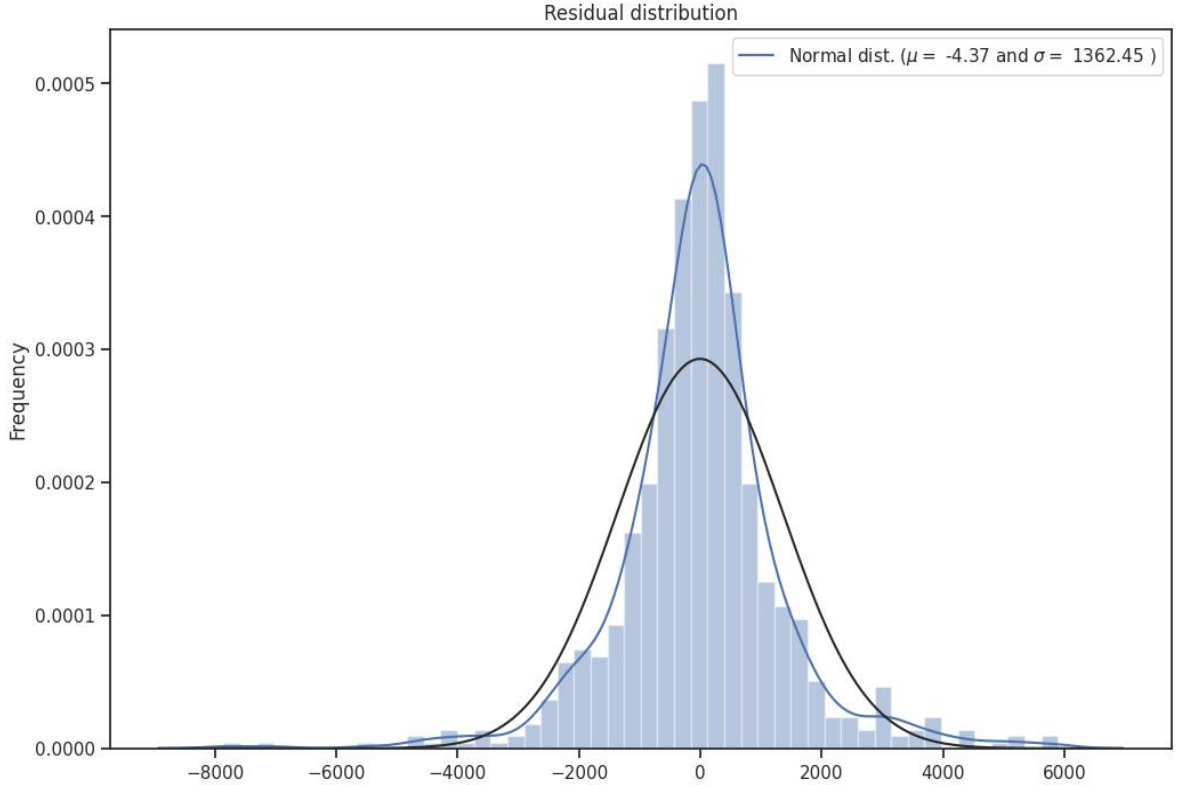
Q: The seasonal moving average order, which represents the number of lagged forecast errors at the seasonal frequency.

s: The seasonal period, which represents the number of time steps in each seasonal cycle (e.g., 12 for monthly data with yearly seasonality).

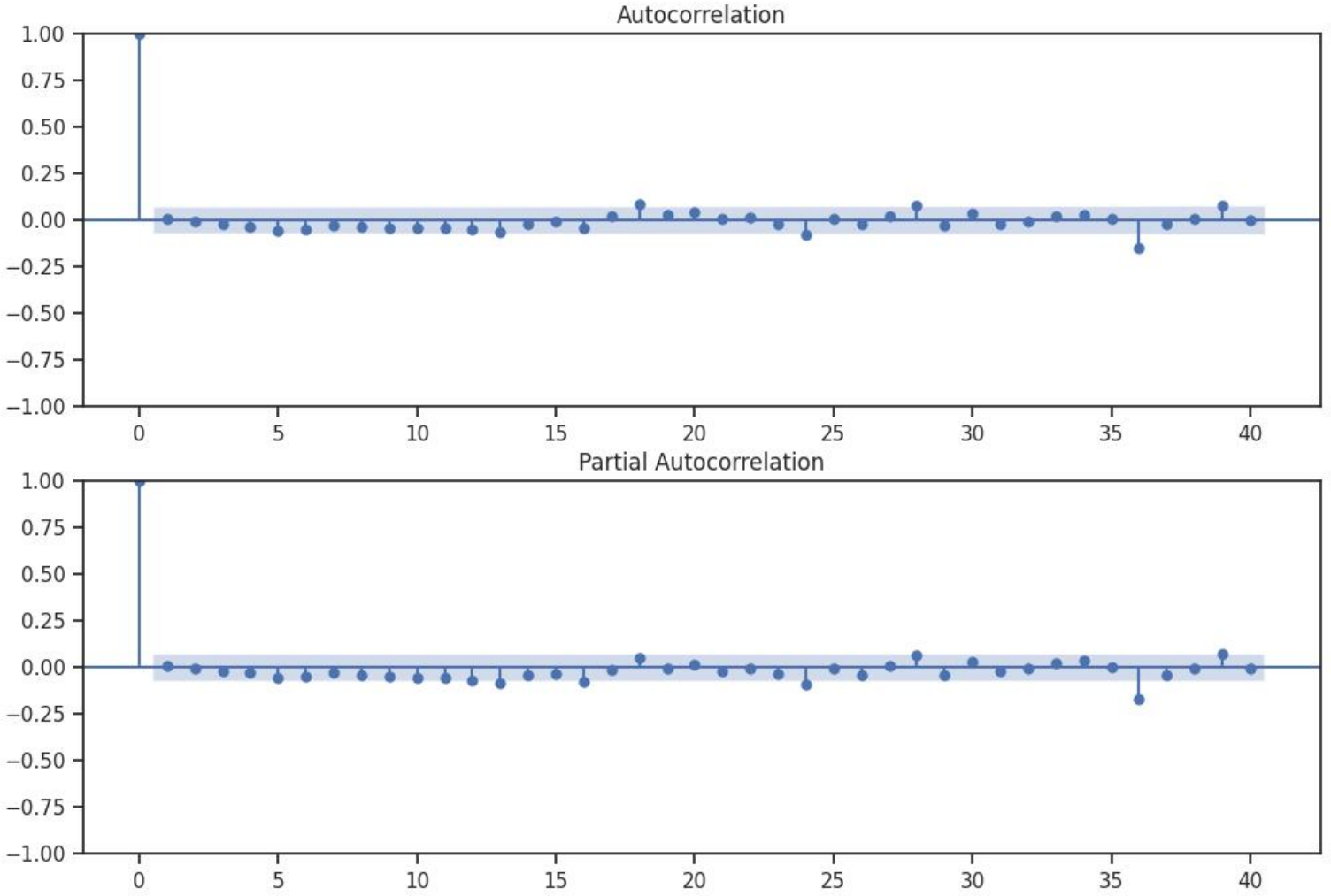
When selecting the values for these parameters, it is common to use techniques such as grid search or automated model selection algorithms (e.g., auto.arima in R) to find the best combination of parameters that minimizes a model selection criterion (e.g., AIC or BIC).

Arima VS Sarima Residual



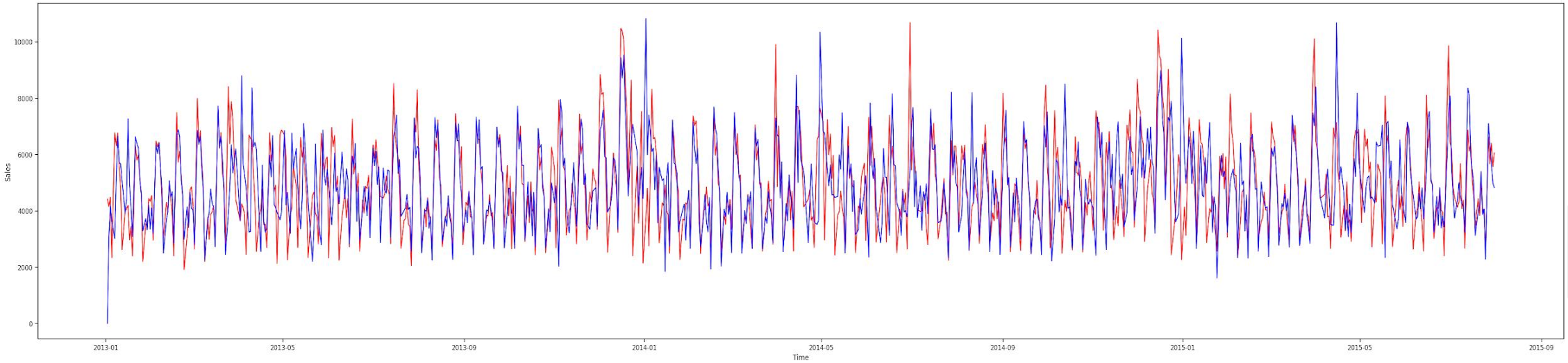


Result

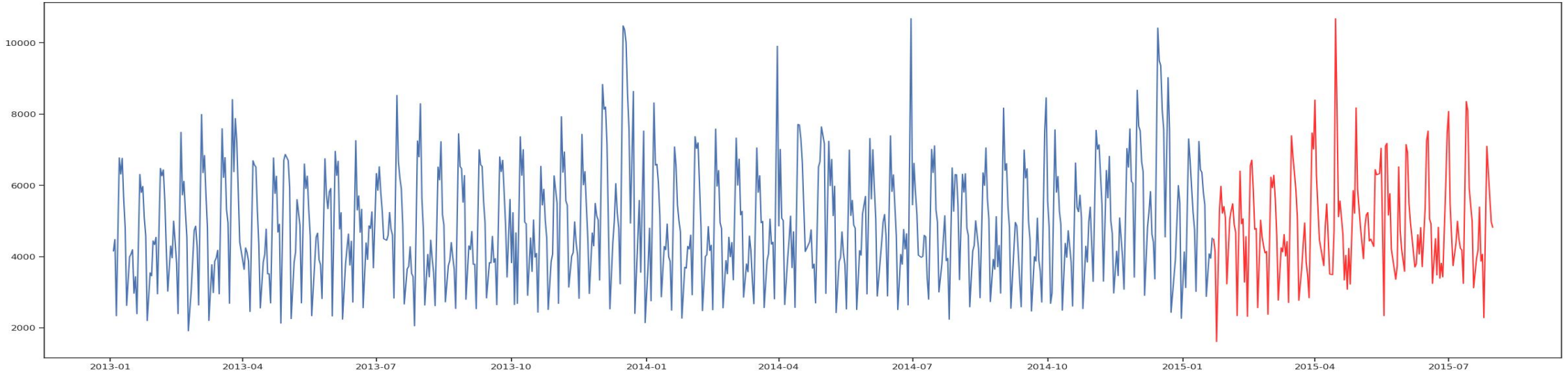


Contoso

Fitted value by sarima

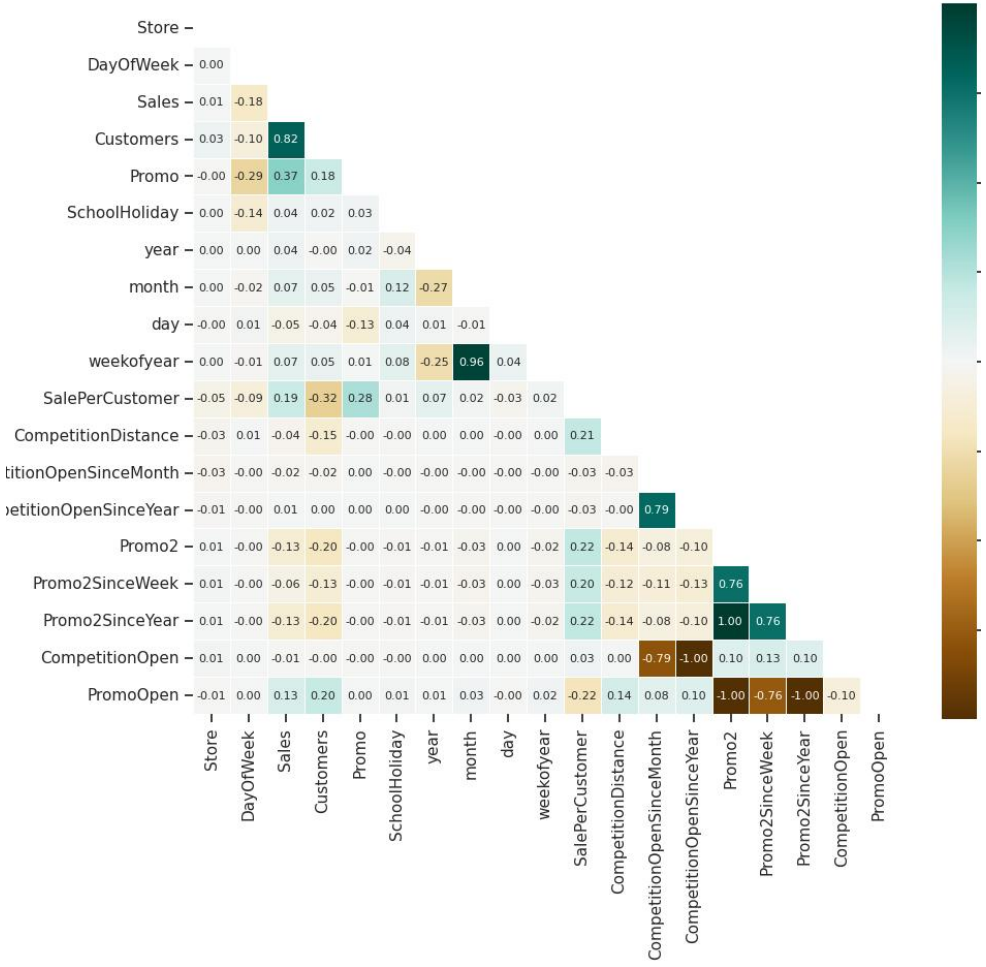


Forecast by sarima

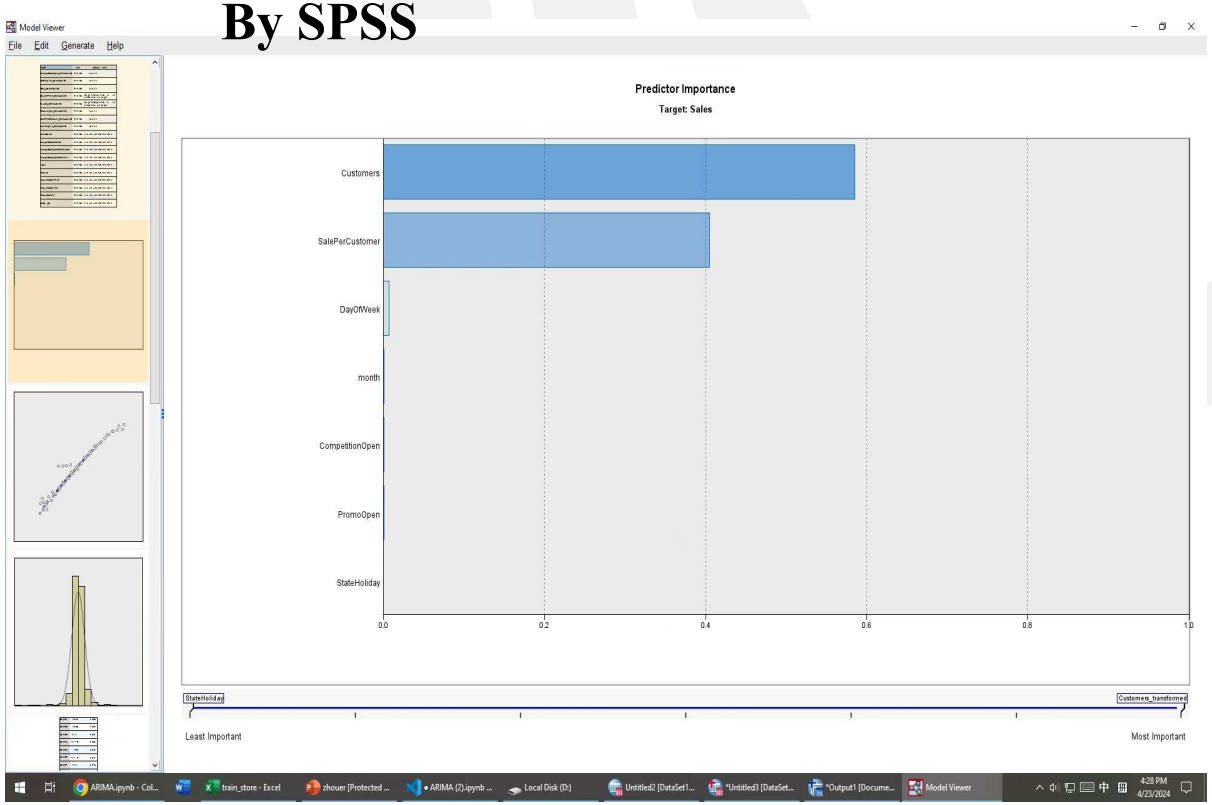


04 Linear regression

Seaborn heatmap for Correlation



By SPSS



A strong relationship between sales and the number of customers.

Linear regression (all store)

model = sm.OLS(y,X)

result = model.fit()

Result -Pvalues

Store	4.026824e-05
DayOfWeek	0.000000e+00
Customers	0.000000e+00
Promo	0.000000e+00
SchoolHoliday	3.048130e-09
year	3.380106e-29
day	8.827041e-01
weekofyear	1.596532e-285
SalePerCustomer	0.000000e+00
CompetitionDistance	1.215879e-12
CompetitionOpen	0.000000e+00
PromoOpen	0.000000e+00
Intercept	1.264685e-37

Result- rsquared : 0.9087089053130488

Result- rmse : 938.8404638845433

```
result.pvalues

Store                   4.026824e-05
DayOfWeek               0.000000e+00
Customers               0.000000e+00
Promo                   0.000000e+00
SchoolHoliday           3.048130e-09
year                    3.380106e-29
day                     8.827041e-01
weekofyear             1.596532e-285
SalePerCustomer         0.000000e+00
CompetitionDistance     1.215879e-12
CompetitionOpen         0.000000e+00
PromoOpen               0.000000e+00
Intercept               1.264685e-37
dtype: float64
```

```
[ ] result.rsquared

0.9087089053130488
```

```
[ ] np.sqrt(np.mean(result.resid**2))#rmse

938.8404638845433
```

```
[ ] result.params

Store                    -0.014601
DayOfWeek               -44.608425
Customers                 7.430991
Promo                   320.732985
SchoolHoliday            17.384098
year                     17.138981
day                       0.019615
weekofyear                2.988610
SalePerCustomer         676.768069
CompetitionDistance       0.001093
CompetitionOpen          -0.004182
PromoOpen                 0.008930
Intercept            -39447.975825
```

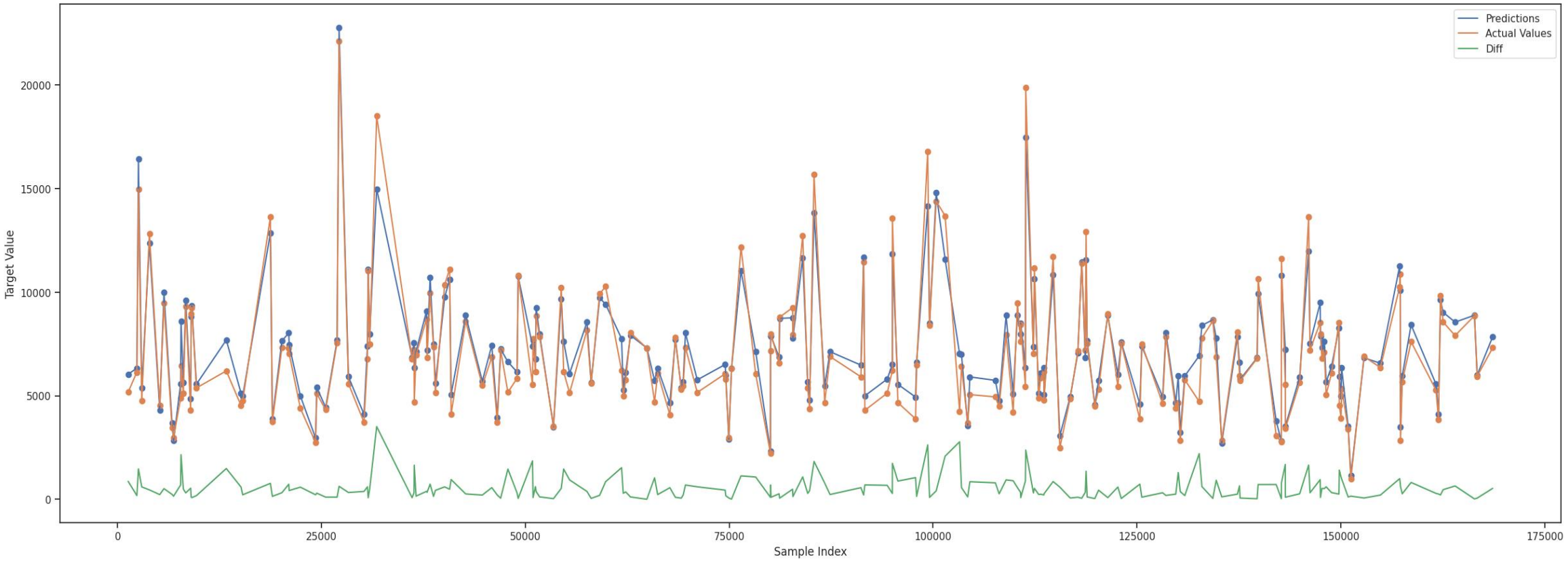
Regression Result On SPSS (only storetypeA)

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0*	*Upper 95.0%*
Intercept	-185.4824601	660.6098242	-0.280774601	0.77895832	-1482.28	1111.317	-1482.28	1111.316855
DayOfWeek	-142.3978541	14.03551545	-10.14553791	8.57984E-23	-169.95	-114.846	-169.95	-114.845665
Customers	6.183027831	0.191109179	32.35337977	6.3667E-146	5.807874	6.558182	5.807874	6.55818158
Promo	1096.30838	51.67254315	21.21645874	4.22759E-79	994.8734	1197.743	994.8734	1197.743321
year	0	0	65535	#NUM!	0	0	0	0
month	0	0	65535	#NUM!	0	0	0	0
day	8.183641531	2.351229168	3.480580133	#NUM!	3.5681	12.79918	3.5681	12.79918351
weekofyea	6.494078325	1.519748006	4.273128373	2.16774E-05	3.510762	9.477395	3.510762	9.477394803
Competiti	0	0	65535	#NUM!	0	0	0	0
Competiti	0	0	65535	#NUM!	0	0	0	0
Competiti	31.86203004	21.41077213	1.488130827	#NUM!	-10.168	73.8921	-10.168	73.89209624
PromoOper	-24.65559167	21.34348539	-1.155181135	0.248372752	-66.5536	17.24239	-66.5536	17.24238838

Delete step by : SalePerCustomer、SchoolHoliday

Contoso

Result -Python



06 Summary

Advantages¶

- Arima module is a powerful tool for the time series forecasting as it accounts for time dependencies, seasonalities and holidays
- SPSS: Multiple model selection, automatic adjustment, multiple prediction methods, visual analysis and other advantages

Drawbacks¶

- Arima ans Sarima doesn't catch interactions betwaen external features, which could improve the forecasting power of a model. In our case, these variables are Promo and CompetitionOpen.
- Fitting seasonal ARIMA model needs 4 to 5 whole seasons in the dataset, which can be the biggest drawback for new companies.
- Seasonal ARIMA in Python has 7 hyper parameters which can be tuned only manually affecting significantly the speed of the forecasting process.

.

Contoso

Contoso

THANK YOU