SUMMER INTERNSHIP REPORT- 2019

ANALYSIS OF RESTAURANT SALES DATA

Authored by: Anitta Augustine, Britto Sabu, Sandeep M A and Ummar Shaik.

Indian Institute of Information Technology and Management-Kerala, Trivandrum.

Submitted on : 25th July 2019

TRIASSIC SOLUTIONS PVT LTD

Get IT right

Preface and acknowledgement

This report is a part of requirement for fulfilling the summer internship programme-2019 at Triassic Solutions Pvt. Ltd., Technopark, Trivandrum. Triassic started as a humble software provider barely a decade ago and it has grown as a global software provider catering to multiple continents. Apart from core software engineering, their works today span to novel facets like automation and emerging technology.

We, a group of four students from Indian Institute of Information Technology and Management- Kerala, have done this internship as a part of the master programme in Computer Science specialization in Data Analytics. The internship was a two months programme beginning from May 14,2019 to July 13, 2019.

We would like to express our primary gratitude to Dr. Manoj Kumar, the Course Coordinator for Data Analytics at IIITM-K, without whose support we would not have succeeded. We also thank Mrs. Deepa Nair, the then HR Manager at Triassic solutions for her gentle way of communication and making things happen. Most importantly we express our sincere gratitude to Mrs. Sultana Shafi, our lead at the Triassic who shared her valuable time to advice and steer us ahead. Of all above the internship was possible mostly because of guidance from Mr. Ankit Jayaprakash, our friend and a Senior Software Engineer at the Triassic Solutions.

Anitta Augustine
Britto Sabu
Sandeep M A and
Ummar Shaik

Table of contents

Introduction

The analysis of experimental data that have been observed at different points in time leads to new and unique problems in statistical modeling and inference. The obvious correlation introduced by the sampling of adjacent points in time can severely restrict the applicability of the many conventional statistical methods traditionally dependent on the assumption that these adjacent observations are independent and identically distributed. The systematic approach by which one goes about answering the mathematical and statistical questions posed by these time correlations is commonly referred to as time series analysis.

In this report we are analyzing a restaurant sales time-series analysis as a part of two months summer internship programme-2019 at Triassic Solutions, Technopark. The data on which we worked on was from an undisclosed restaurant from Toronto, Canada. Our study can be largely divided into the following manner:

i) Exploratory data analysis
ii) Sales prediction techniques
iii) Market basket analysis and
iv) Future scope of the project

Exploratory data analysis deals with basic understanding of the data by preprocessing it. It can be so helpful to find many hidden patterns in the data so that it might be helpful for further analysis and decision making in a business environment.

In the *Sales prediction techniques* we are exploring the possibilities of different time series prediction algorithms like ARIMA, SARIMA and SARIMAX. We would be comparing the different aspects of each algorithms in detail and theirs results.

Finally, we will be concluding the work by examining the future scope and potential works in the project.

Chapter 1

Exploratory Data Analysis

Before directly going to exploratory data analysis (EDA), it is important us to know which data is given us for wrangling. Here we're provided with two datasets:

I) SalesDTL and
II) SalesHDR

Since the ultimate aim of the analysis is to build a prediction model, we need to carefully explore the data. Both the given files contain sales related data of the restaurant. More precisely, the SalesDTL dataset contains the sales related data of the restaurant and the SalesHDR contains details related to customers. The following are the major insights that we got from the exploratory data analysis. We have used Python (Jupyter notebook, Anaconda) for all the analysis in this project.

EDA for SalesDTL dataset

The following are the attributes (columns) that are given in the SalesDTL dataset.

```
Index(['SDTL_IDL', 'SHDR_IDL', 'RECEIPTNO', 'PROD_IDL',
'PRTYPE_IDL', 'DEPT_IDL', 'BSP', 'RPRICE', 'INCTAXES', 'CPRICE',
'QUANTITY', 'DATEORDER', 'EMPORDERED', 'EMPAUTH', 'DISPORDER',
'PRODGROUP', 'SEAT', 'TYPE', 'TAX1', 'TAX2', 'TAX3', 'TAX4', 'TAX5',
'TAX6', 'TAX7', 'STRATE', 'TRACKSTOCK', 'PCHANNELS', 'POINTS',
'NOTES', 'GROUP_IDL', 'MASTERITEM', 'DEPOSITS', 'ISVOIDED', 'PDFSENT',
'DESCAPPEND', 'MATRIXNUM', 'PREPSENT', 'COURSESORT', 'OMSDTL_IDL',
'ORSDTL_IDL', 'OQUANTITY', 'OCOST', 'OBSP', 'ORETAIL', 'STORENO',
'OTHERFEE', 'HOG_IDL'.
```

In total we have 48 attributes. But we need to know how these can be manipulated for suiting our check if all the columns are fit for the analysis. Therefore, we checked their data types. The following is the results we got.

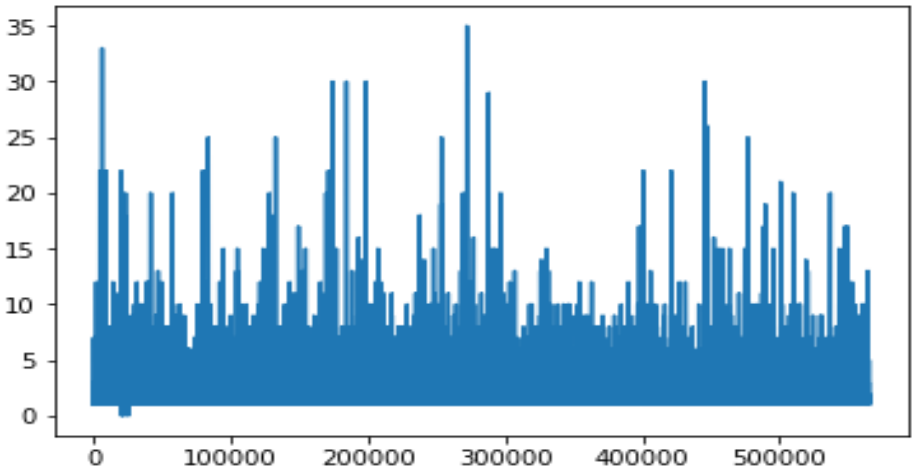
```
SDTL_IDL float64, SHDR_IDL float64, RECEIPTNO float64, PROD_IDL
float64, PRTYPE_IDL float64, DEPT_IDL float64, BSP float64, RPRICE
float64, INCTAXES float64, CPRICE float64, QUANTITY float64, DATEORDER
object, EMPORDERED float64, EMPAUTH float64, DISPORDER int64,
PRODGROUP int64, SEAT int64, TYPE object, TAX1 object, TAX2 object,
TAX3 object, TAX4 object, TAX5 object, TAX6 object, TAX7 object, STRATE
```

```
float64, TRACKSTOCK object, PCHANNELS float64, POINTS float64, NOTES
float64, GROUP_IDL float64, MASTERITEM float64, DEPOSITS float64,
ISVOIDED object, PDFSENT float64, DESCAPPEND float64, MATRIXNUM int64,
PREPSENT object, COURSESORT int64, OMSDTL_IDL float64, ORSDTL_IDL
float64, OQUANTITY float64, OCOST float64, OBSP float64, ORETAIL
float64, STORENO float64, OTHERFEE float64, HOG_IDL float64.
```

But on further checking on these columns individually, we understood that most of the columns containing no information or invalid information. Therefore, out of the 48 attributes we have finalized six attributes. They are: `'SHDR_IDL', 'PROD_IDL','PRTYPE_IDL','RPRICE','QUANTITY','DATEORDER'`. In the following studies we will be going through only these parameters. The *head* of these attributes in Python is printed as given.

	SHDR_IDL	PROD_IDL	PRTYPE_IDL	RPRICE	QUANTITY	DATEORDER
0	1.0	3.0	2.0	7.02	1.0	2005-02-13 03:35:52
1	1.0	4.0	2.0	3.17	1.0	2005-02-13 03:35:52
2	2.0	3.0	2.0	9.95	1.0	2005-03-08 09:18:41
3	3.0	3.0	2.0	9.95	1.0	2005-03-08 09:21:59
4	4.0	28.0	13.0	2.79	1.0	2008-12-19 09:21:48

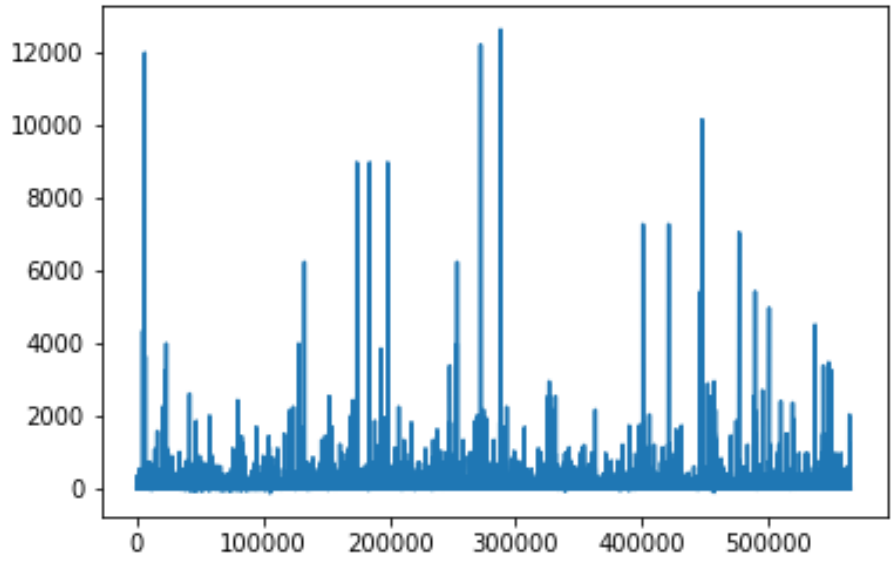
Also note that at this point the dataframe has 565304 rows. To get more insights about the data points, we can plot them. One of the best ways for plotting in Python is importing matplotlib library and it's pyplot. For example, the following is the graphical representation of the magnitude of QUANTITY (histogram plot) and associated points.



The QUANTITY histogram explains the following business insights:

1. The maximum number of quantity bought in a single order is 35.
2. Some of the values in the QUANTITY column has negative value, which might be a clerical error. Since wrong entries may invite invalid analysis, we strongly advice to avoid any clerical mistakes during data input.
3. Although this histogram plot gives total quantity bought Vs order, we cannot conclude any macro decision.

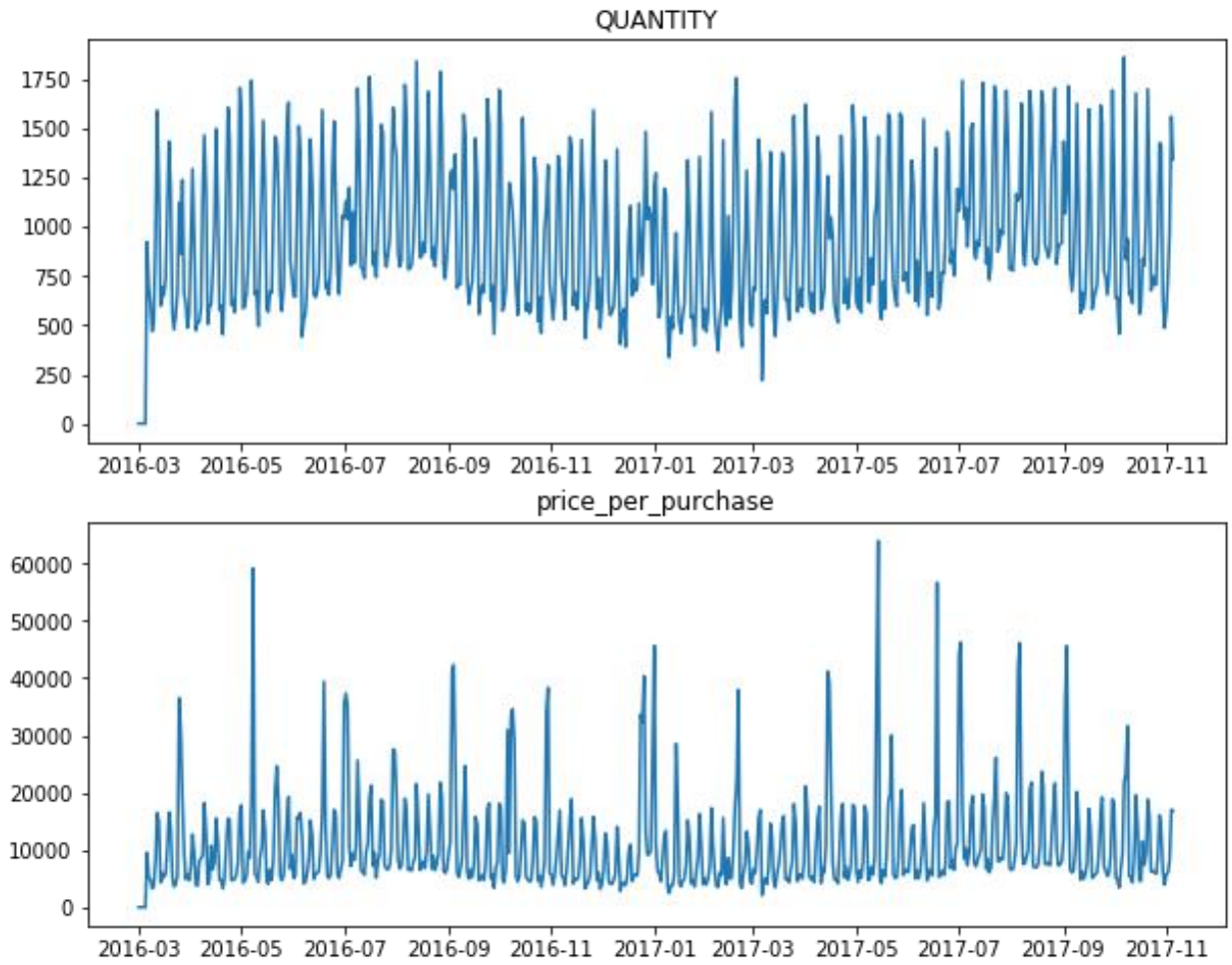
Now let's say we need to know total price per purchase. In that case we can create a new column named 'price_per_purchase' and add to the existing dataframe. While doing this we have found that there are 324 negative entries (possibly clerical error). To resolve the problem, we have taken the liberty to approximate all the entries by taking the absolute value of the all points. Now total price per purchase histogram will be looking as following.



There are some interesting business perspectives that we got from the graph. It is tabulated aside. While the maximum amount spent per purchase is C$12606.59 we cannot expect every purchase at this scale because the average of the purchase stands at C$14.27. Even at 75% quartile, we can only expect C$9.95 purchase. That means higher price purchases are too rare for the restaurant per purchase.

count	565304.000000
mean	14.272679
std	68.495995
min	0.000000
25%	2.500000
50%	7.000000
75%	9.950000
max	12606.590000

For further insights from here we are sampling our data on daily basis and want to know price per purchase per day and quantity per day. The resulting plots are given below.



For plotting the above graphs we have to subset the dataset from March 3,2016 to the end of available data. We have done this to make the data a continuous one. Only with a good continuous data the time series analysis can be meaningful. In addition, the following are the main insights that we can conclude from the plots.

1. There is a clear seasonality for QUANTITY and price_per_purchase. That means if all the factors remain same the restaurant can expect similar pattern of sales in the coming year too.
2. But there is no clear trend that we can conclude form the graph. i.e, no upward or downward trend can be found. This may be due to scarcity of data. That means we have got only two years data. If more years data were available, more trend might have been distinctly observed.

EDA for SalesHDR dataset

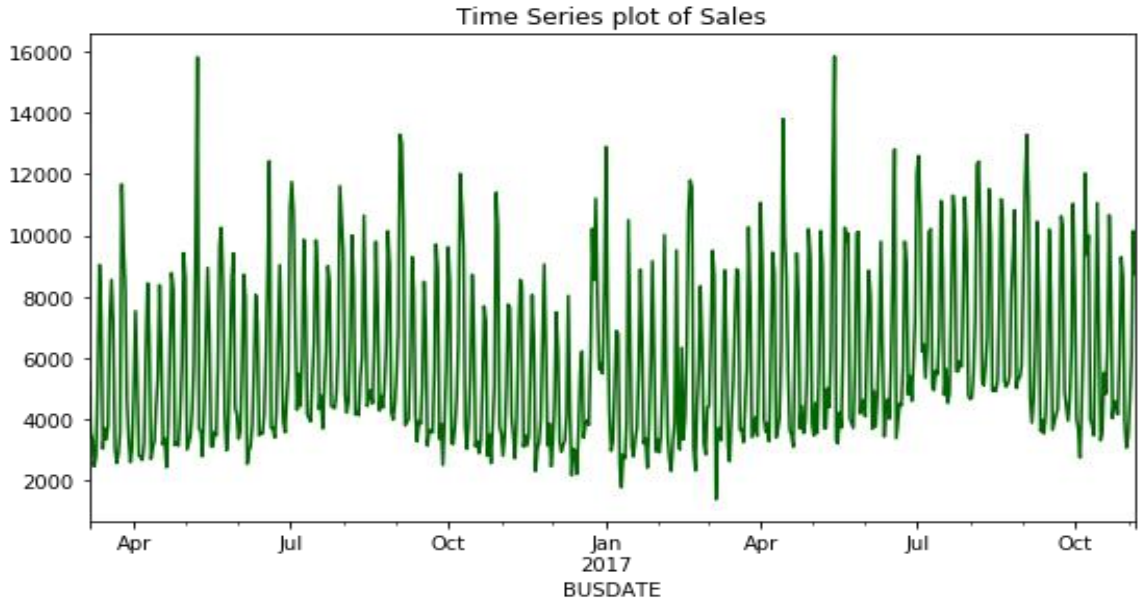
Now let's delve into more insights about SalesHDR dataset. The following are the attributes that we possess in SalesHDR dataset: ['SHDR_IDL', 'HEADERTYPE', 'BUSDATE', 'RECEIPTNO', 'TABLENUM', 'NUMCUST', 'TAX1NET', 'TAX1SALES', 'TAX1TYPE', 'TAX2NET', 'TAX2SALES', 'TAX2TYPE', 'TAX3NET', 'TAX3SALES', 'TAX3TYPE', 'TAX4NET', 'TAX4SALES', 'TAX4TYPE', 'TAX5NET', 'TAX5SALES', 'TAX5TYPE', 'TAX6NET', 'TAX6SALES', 'TAX6TYPE', 'TAX7NET', 'TAX7SALES', 'TAX7TYPE', 'PENNYROUND', 'MDSETOTAL', 'EOPEN_IDL', 'ECLOSE_IDL', 'DATESEATED', 'DATEOPEN', 'DATECLOSED', 'CUST_IDL', 'BPOINTS', 'GCPPOINTS', 'GCPCARDNUM', 'STYPE_IDL', 'STRACK_IDL', 'SECT_IDL', 'STAT_IDL', 'SHIFT_IDL', 'NOTES', 'RCPTPRNTED', 'COSTTOTAL', 'TOTALDUE', 'SALEREPIDL', 'GRATUITY', 'DEPOSITS', 'TAX_IDL', 'PUNCH_IDL', 'ISLOCKED', 'SSP', 'SSETUP', 'WH_IDL', 'PRICELEVEL', 'COMMENT', 'FREIGHT', 'FREIGHTTX1', 'FREIGHTTX2', 'PONUMBER', 'SHIPTO', 'LABEL', 'SEATCOUNT', 'TAXEXEMPT', 'WAITING', 'STORENO', 'OTHERFEE'].

Thus, in total there are 69 attributes. Since the dataframe has many columns, we needed to extract only the relevant columns by plotting a correlation plot, which is given below.

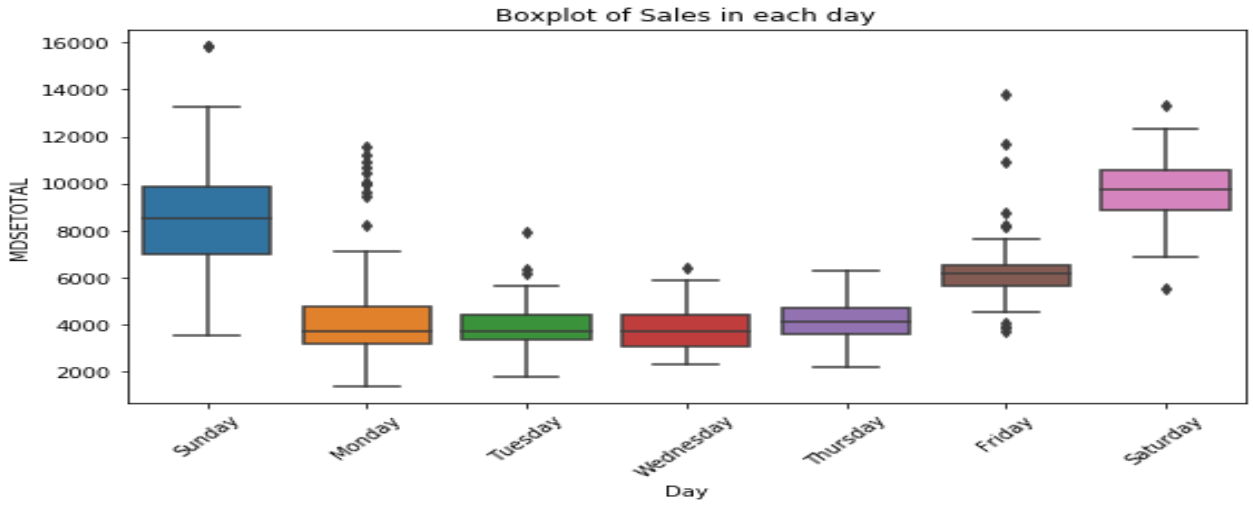
Our focus is on the sales data, so if we check the regarding filed, we can see only a few columns shows and association with the data. So, we can safely ignore those columns for the time being. The data is given on bill by bill basis, so for better results, it is

necessary to resample the data to daily basis for further analysis. From the basic description of the data we can see that the Maximum Number of customers at one time was 85 and maximum sales got was 1500 for one reference bill. Also, there are some inconsistencies in which the minimum value of Sales is a negative value which probably is a clerical error. We can find the negative values and replace them with respective absolute value.

Now that we have extracted few columns and resampled the data on daily basis. The plot of the Total sales per day for one year looks like this:



The data ranges from 6th March 2016 to 5th November 2017. Again, we have subsetted this range so as to get a continuous data. Later, the resampled data were analyzed with respect to the daily statistics and added a column which labels the day correctly. Now let's see how the boxplot for days would look like.

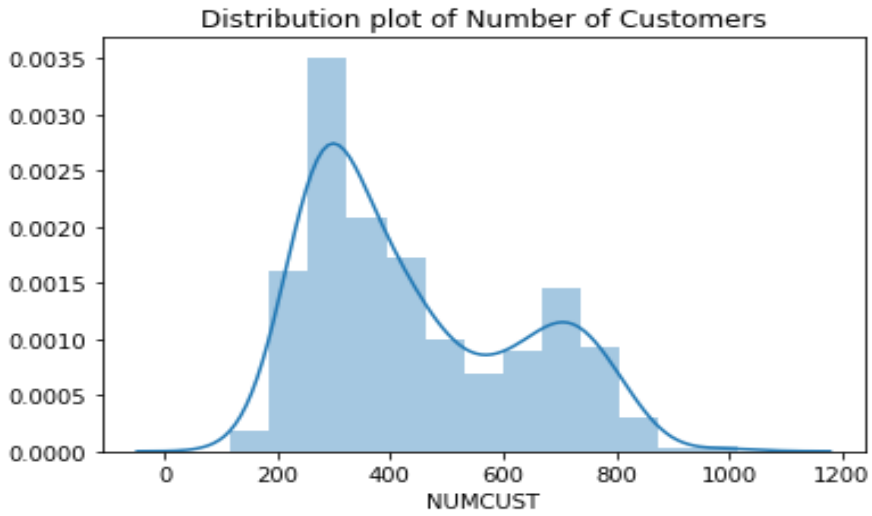


The above boxplot explains that Sundays and Saturdays are experiencing more sales in real terms (ignoring outliers). While the weekdays have more or less same sales pattern with an average around C$4000. There is a clear trend of increase in sales when the weekend approaches. This spike has been understood from Thursday sales onwards with a vivid increase on Friday.

We can recommend two things here:

1. As more people are approaching the restaurant at weekends, publishing more offers, preparing special cuisines etc… could boost the sales.
2. The restaurant should try to boost the weekday sales too.

The average number of customers in a day is 442. But if we examine the distribution plot, we can see that the peak is on the 250-300 region.

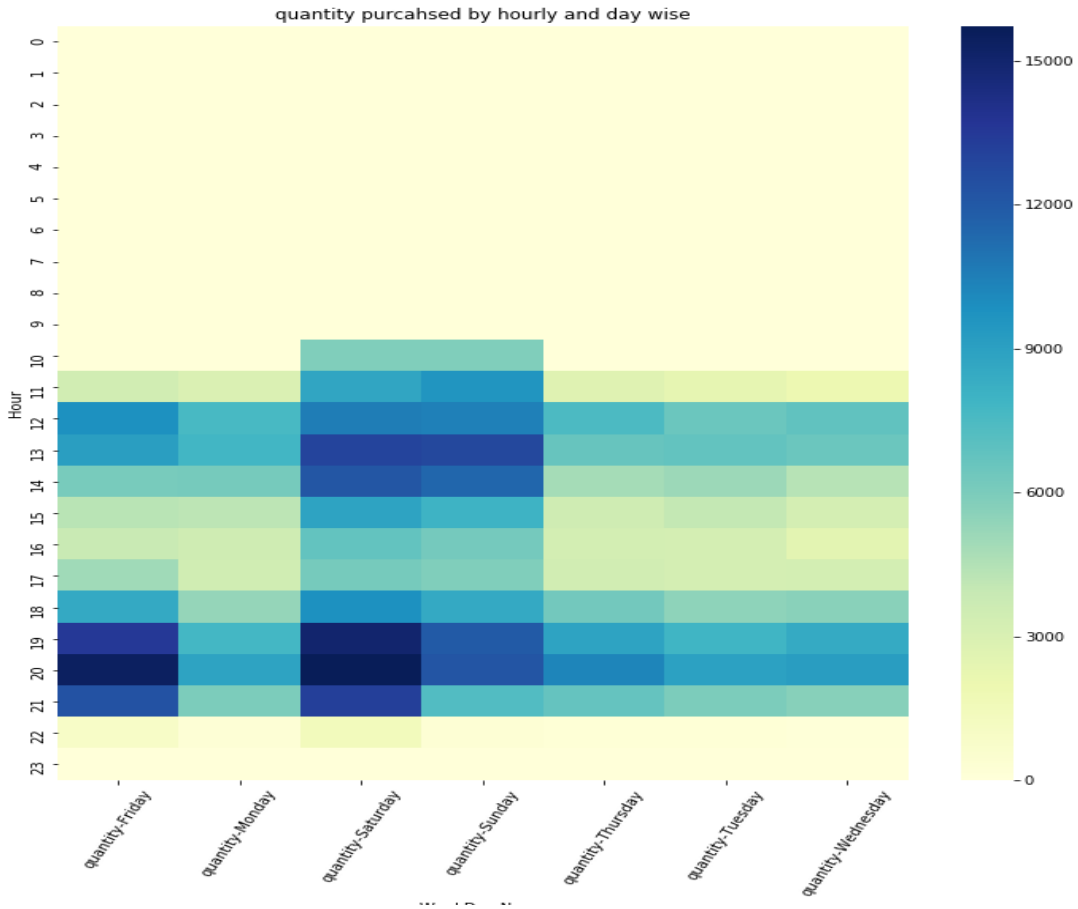


Although the boxplot can explain the distribution of sales on distinct days, it cannot explain the same in hours in a single day. For solving this we require a heatmap (see next page).

The above heatmap explains hourly sales for each day in a week. The colour bar at the right end shows the extend of sales per hour. It seems that the restaurant opens at 10am daily and 9am on Saturdays and Sundays. It might be closing by 11pm. Some of the other evident features from this plot are explained below.

1. It seems the restaurant serves breakfast only on weekends. Also, they are receiving better sales on that business.
2. On Saturdays and Sundays more sales on lunch (12pm – 2pm) is observed. While on weekdays they are average.

3. Friday and Saturday nights witness a heavy rush for dinner during 6pm -9pm.
4. Between 1pm and 5pm, except on Saturdays and Sundays, the sales are in below average range.

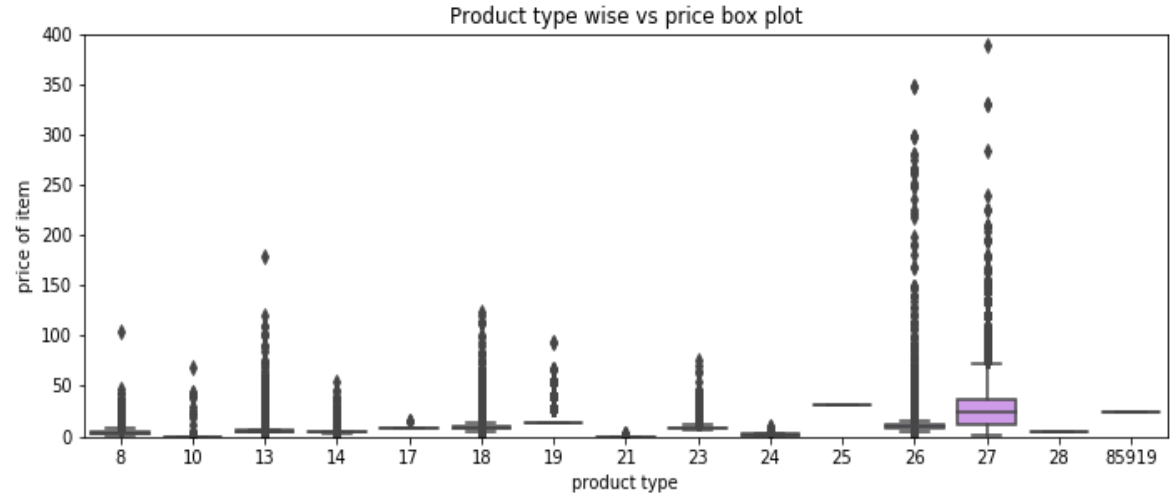


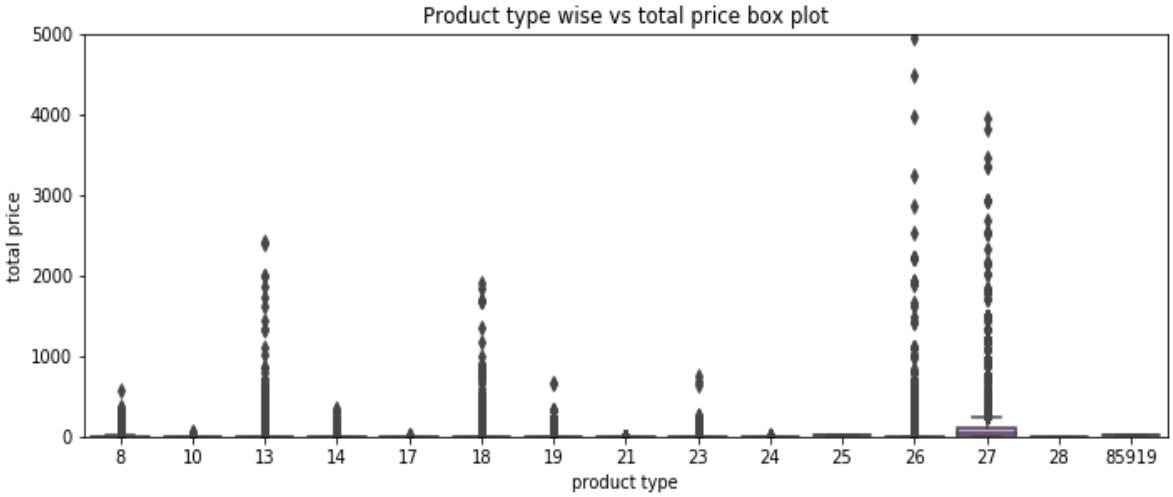
After learning above insights from data we can recommend the following points for better sales:

1. The restaurant should concentrate to fetch more sales during the time 1pm-5pm.
2. They can also use the above said time (the less rush hours) for preparing the coming rush hours (6pm-9pm) on the same day.
3. On an experimental basis, they can serve breakfast on weekdays too.

Exploratory Data Analysis of Product Types and Products

The given data contains 15 product types and 401 products are included in menu. The price distribution of product types is given by:



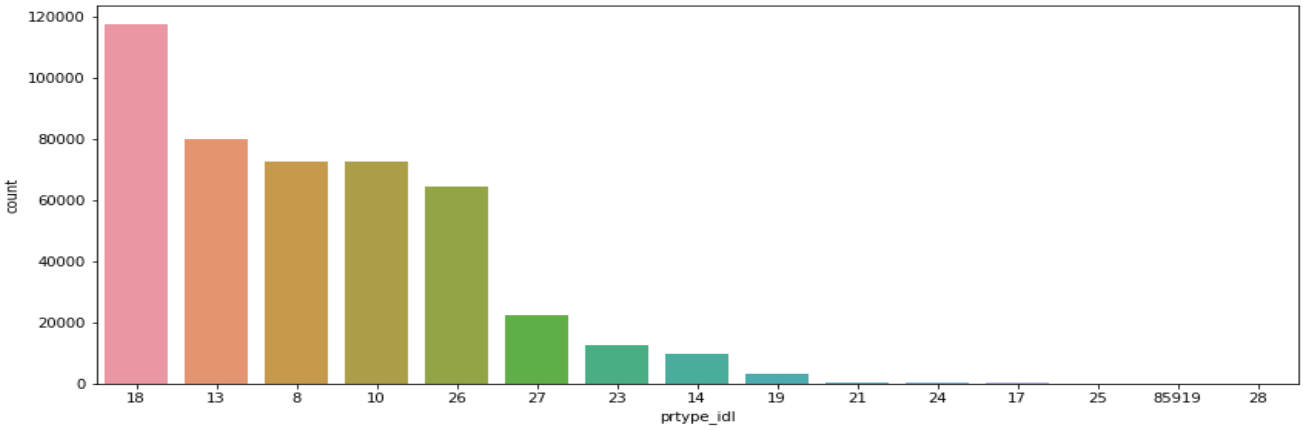


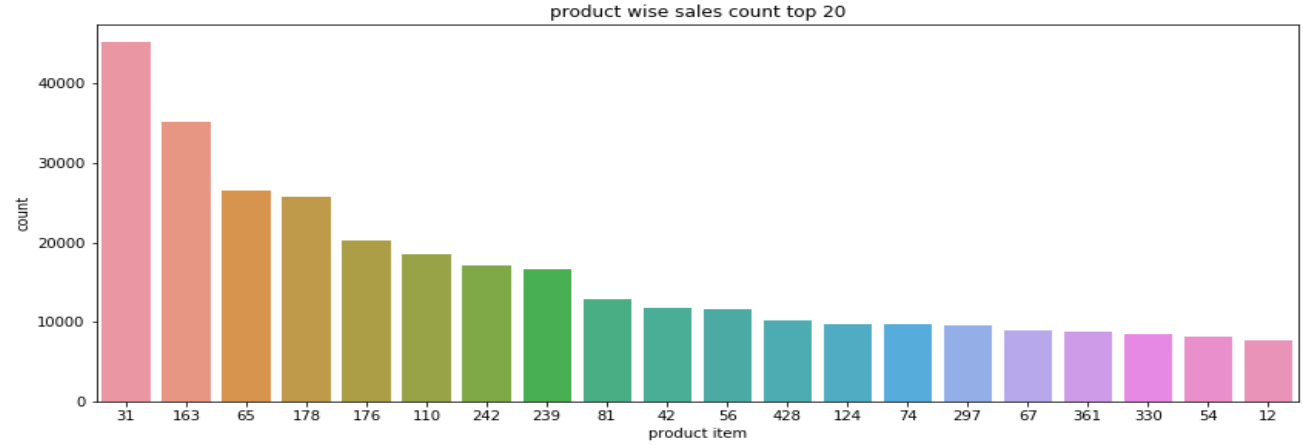
It is clear that the prices of product types 26, 27, 18,13, and 23 are higher and that leads to the total sales of product types are also higher. After grouping the product types according to their total sales in number (count), we found the top 15 products types are 18,13, 8,10,26,27,23,14,19,21,24,17,25,85919,28

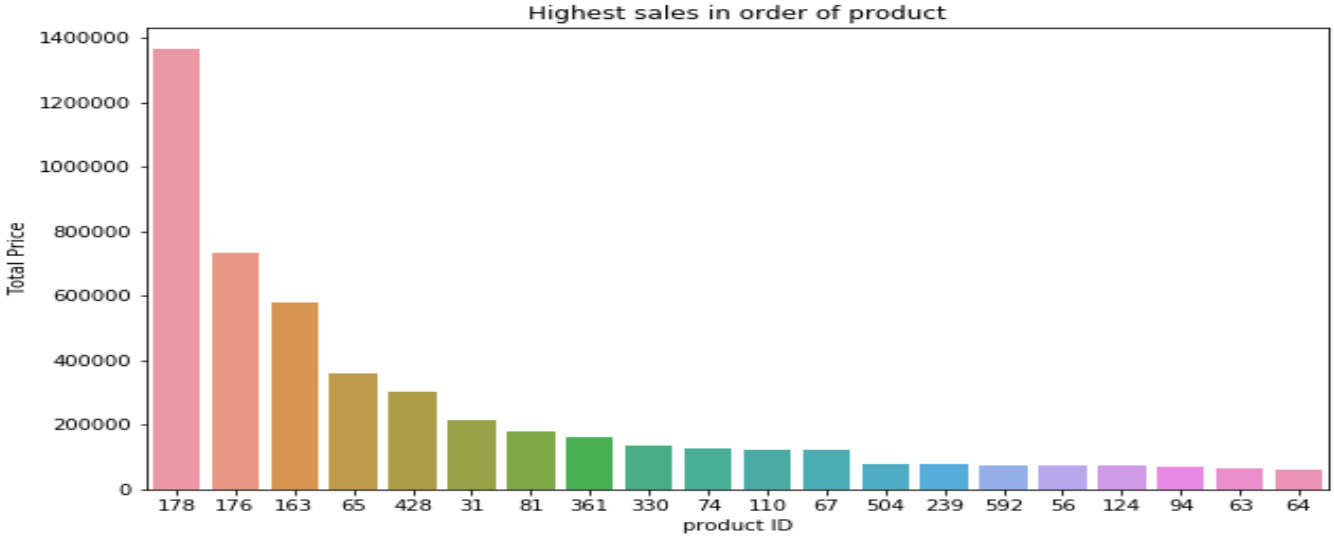
But when grouping with products according to their total count, we found that the top 20 products are 31,163,65,178,176,110,242,239,81,42,56,428,124, 74, 197, 67, 361, 330, 54 and 12. These are the more frequent items ordered.These insights are obtained from the following plots.

In addition, the plot below explains the total sales of different products in descending order. The highest turnover products are 178,176,163,428,31,81.

(PTO)

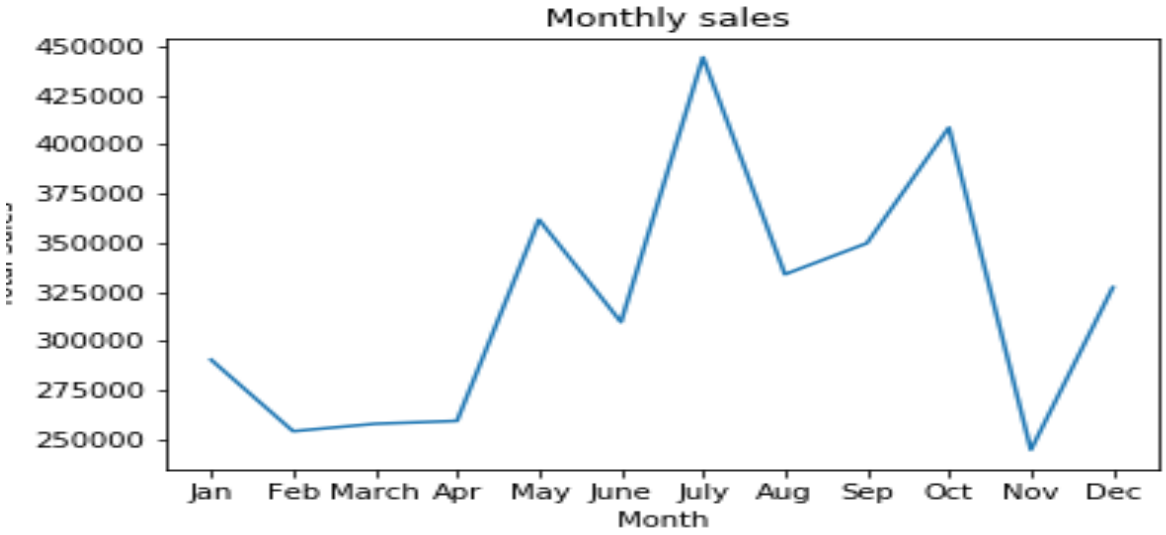






We have observed the sales trend in daily and weekly basis. Now let's check the sales distribution on monthly basis.

(PTO)



It is clear from the plot that the total sales are highest in July, October and lowest in November and February.

Chapter 3

Sales Prediction Techniques

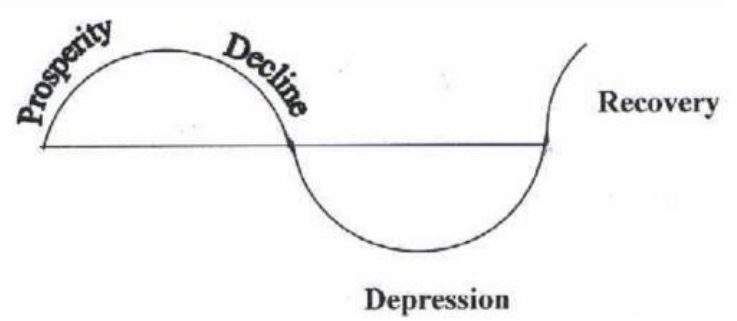
There are many techniques in time series to model a dataset out of which we have selected ARIMA, SARIMA and SARIMAX for our study. As a part of future scope of the same project we will be trying to apply the LSTM technique too. But in this chapter let's understand how the above said three techniques differs each other. To get a better intuition about these techniques without involving inside math of it, the following explanation would help the reader to understand the prediction techniques.

Components of a Time Series

A time series in general is supposed to be affected by four main components, which can be separated from the observed data. These components are: *Trend*, *Cyclical*, *Seasonal* and *Irregular* components. A brief description of these four components is given here. The general tendency of a time series to increase, decrease or stagnate over a long period of time is termed as Secular Trend or simply Trend. Thus, it can be said that trend is a long-term movement in a time series. For example, series relating to population growth, number of houses in a city etc. show upward trend, whereas downward trend can be observed in series relating to mortality rates, epidemics, etc.

Seasonal variations in a time series are fluctuations within a year during the season. The important factors causing seasonal variations are: climate and weather conditions, customs, traditional habits, etc. For example, sales of ice-cream increase in summer, sales of woolen cloths increase in winter. Seasonal variation is an important factor for businessmen, shopkeeper and producers for making proper future plans.

The cyclical variation in a time series describes the medium-term changes in the series, caused by circumstances, which repeat in cycles. The duration of a cycle extends over longer period of time, usually two or more years. Most of the economic and financial time series show some kind of cyclical variation. For example, a business cycle consists of four phases, viz. i) Prosperity, ii) Decline, iii) Depression and iv) Recovery. Schematically a typical business cycle can be shown as aside.



Irregular or random variations in a time series are caused by unpredictable influences, which are not regular and also do not repeat in a particular pattern. These variations are caused by incidences such as war, strike, earthquake, flood, revolution, etc. There is no defined statistical technique for measuring random fluctuations in a time series.

Concept of Stationarity

In the most intuitive sense, stationarity means that the statistical properties of a process generating a time series do not change over time. It does not mean that the series does not change over time, just that the *way* it changes does not itself change over time. The algebraic equivalent is thus a linear function, perhaps, and not a constant one; the value of a linear function changes as x grows, but the *way* it changes remains constant — it has a constant slope; one value that captures that rate of change.

Autocorrelation and Partial Autocorrelation Functions (ACF and PACF)

To determine a proper model for a given time series data, it is necessary to carry out the ACF and PACF analysis. These statistical measures reflect how the observations in a time series are related to each other. For modeling and forecasting purpose it is often useful to plot the ACF and PACF against consecutive time lags. These plots help in determining the order of AR and MA terms.

ARIMA modelling

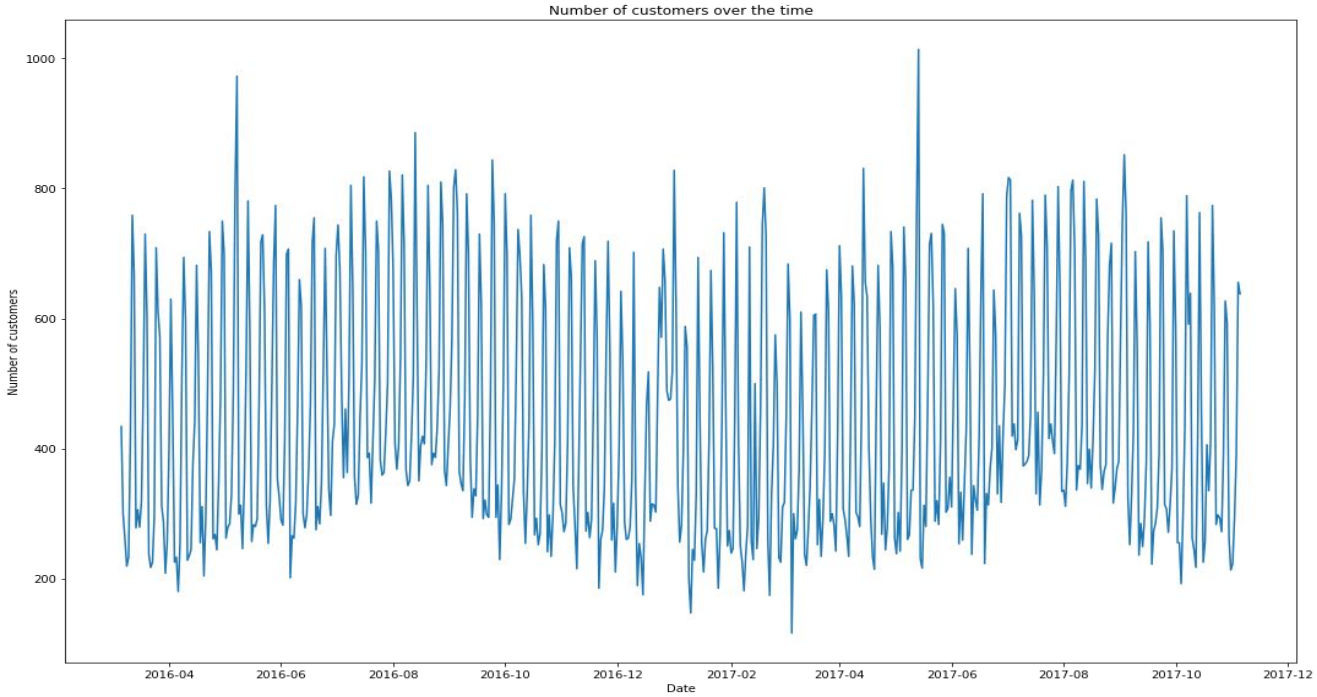
ARIMA (Auto Regressive Integrated Moving Average) whose acronym is descriptive, captures the key aspects of the model itself. Briefly, they are:

- AR: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
- I: *Integrated*. The use of differencing of raw observations (i.e. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- MA: *Moving Average*. A model that uses the dependency between an observation and residual errors from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of *ARIMA(p,d,q)* where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used. The parameters of the ARIMA model are defined as follows:

- *p*: The number of lag observations included in the model, also called the lag order.
- *d*: The number of times that the raw observations are differenced, also called the degree of differencing.
- *q*: The size of the moving average window, also called the order of moving average.

From the dataset SalesHDR, we have taken NUMCUST as the attribute to analyze using ARIMA. NUMCUST signifies number of customers being served at the restaurant per day. This can be understood from the plot below.

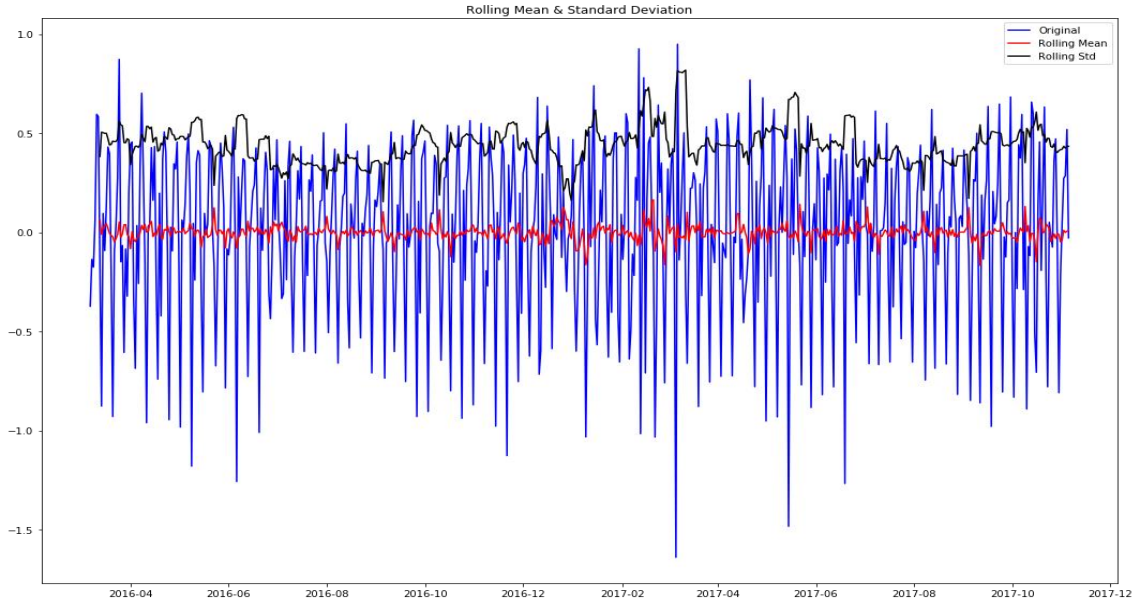


Some of the basic statistical tests that has to be done here are:

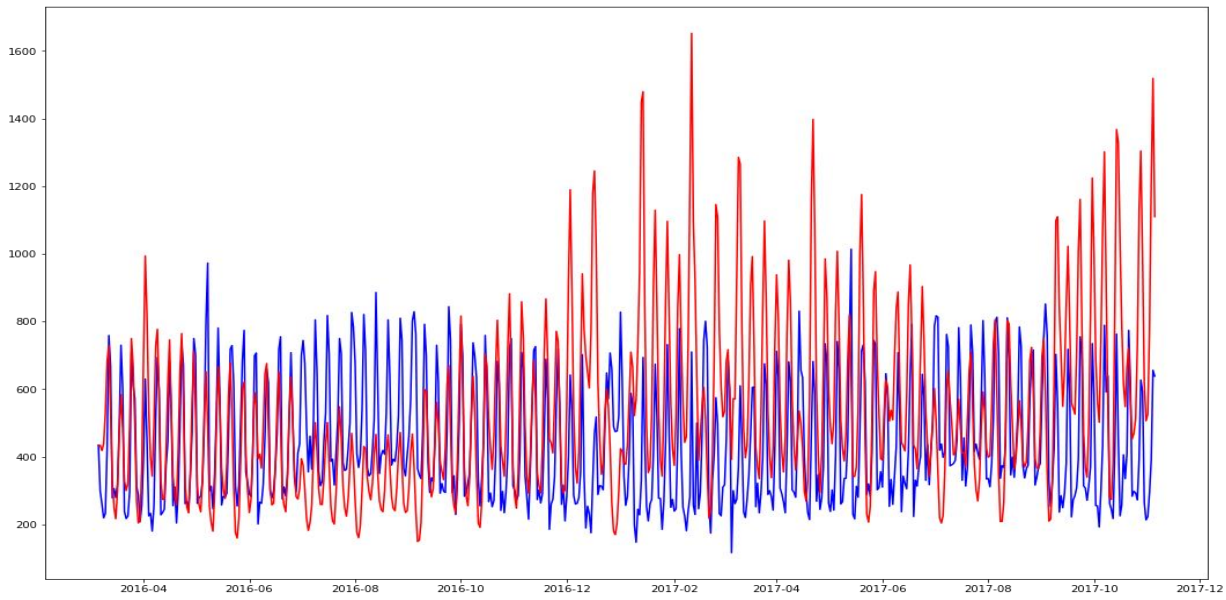
1. Finding rolling statistics (rolling mean, standard deviation and rolling average)
2. Performing Augmented Dickey-Fuller Test.
3. Log transformation and reverse transformation of data.

Rolling statistics result

The following is the log transformed plot in which rolling statistics are observed.



Finally, we've predicted with ARIMA model and the following result were observed. Note that the *(p,d,q)* values are in the range of (0,4). The following plot is the best fit prediction for the above given range (red- predicted, blue-real).



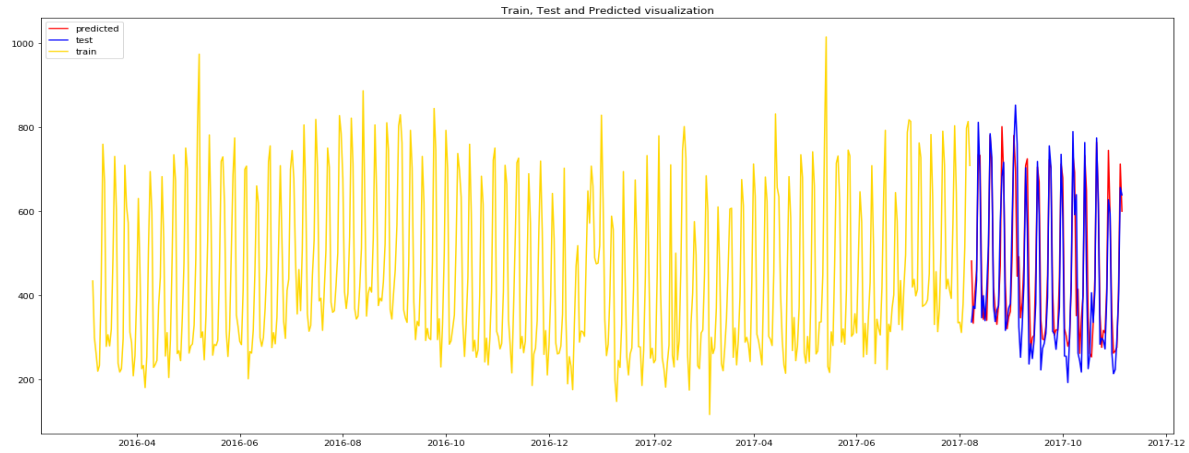
It is evident that the predictions are not at all accurate and has high fluctuations in almost all over the data. Therefore, we cannot consider this method to predict the sales of the restaurant using this model. Hence, we discard ARIMA as a model for prediction for the purpose and go to SARIMA model.

SARIMA modelling

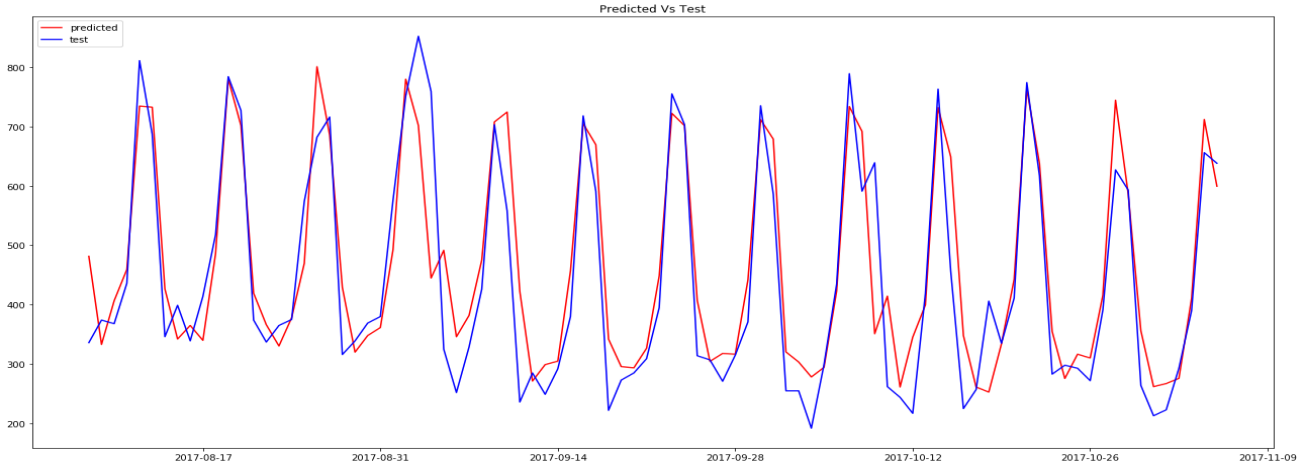
In this model seasonal differencing of appropriate order is used to remove non-stationarity from the series. A first order seasonal difference is the difference between an observation and the corresponding observation from the previous year. Forecasting SARIMA processes is completely analogous to the forecasting of ARIMA processes discussed above in the ARIMA section. But few more parameters are to be encountered here. In total, there are six parameters to deal with:

- (p, d, q) for non-seasonal part.
- (P, D, Q)s for seasonal part, where s is the length of season.

We have taken the same data (NUMCUST) for analysis here. The test, train and predicted data is as shown below.



In order to visualize the predicted data and test data alone, the following plot would be more helpful.



The SARIMA plot seems very promising than that of ARIMA. To get this result, we needed to run the parameters in the order (0,5) and found that order = (0,0,4), seasonal_order = (1,2,4,7) gives the best result. Note that "7" in seasonal_order indicates The test data almost coincides with train data. But this 'almost' can be quantified by MSE (Mean Square Error). After calculating MSE was found to be 7296.248426909384. But our major concern is to decrease MSE in order to increase accuracy.

Therefore, pros and cons in SARIMA implementation here are:

1. We can predict the sales of restaurant with the model, but with less accuracy.
2. The MSE needs to be reduced.
3. If there occur any external factor that have an effect on the sales pattern this model would fail terribly. Hence for multivariate analysis SARIMA is not recommendable.

Now we are going to address this issue. What if any external factor like government laws, market fluctuation, oil price, labour strike etc. have a role in deciding sales of the restaurant? How do we take these unknown parameters into consideration? The stumbling block here is that we do not even now what factors are contributing to the sales of the restaurant in larger picture. Taking more parameters for prediction is known as multivariate analysis. One model for tackling this is implementing SARIMAX.

SARIMAX implementation

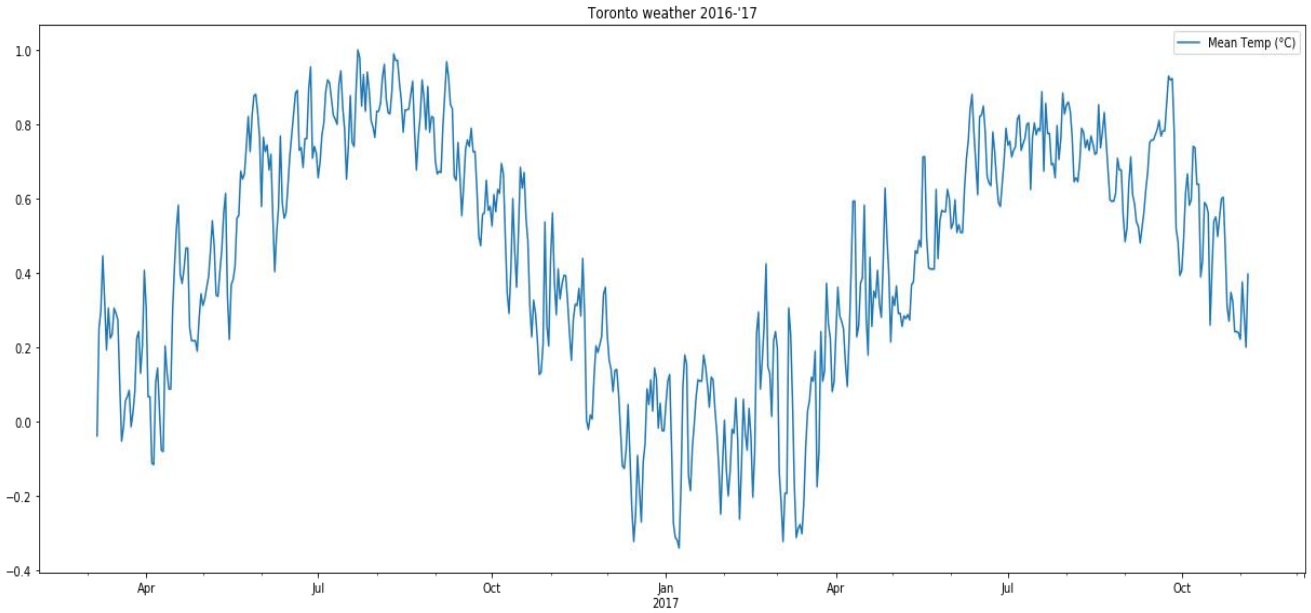
In this section we will see if SARIMAX can solve the issues raised in former session. i.e, i) To increase accuracy.

ii) To decrease MSE.

iii) To accommodate external factors in calculating the prediction model.

SARIMAX is just same as SARIMA added with one feature that we can add an exogenous (external) factor to do multivariate analysis. Here, we were told to take weather data of Toronto city (of 2016 and 2017) as external factor. This is a public data freely downloaded from http://climate.weather.gc.ca.

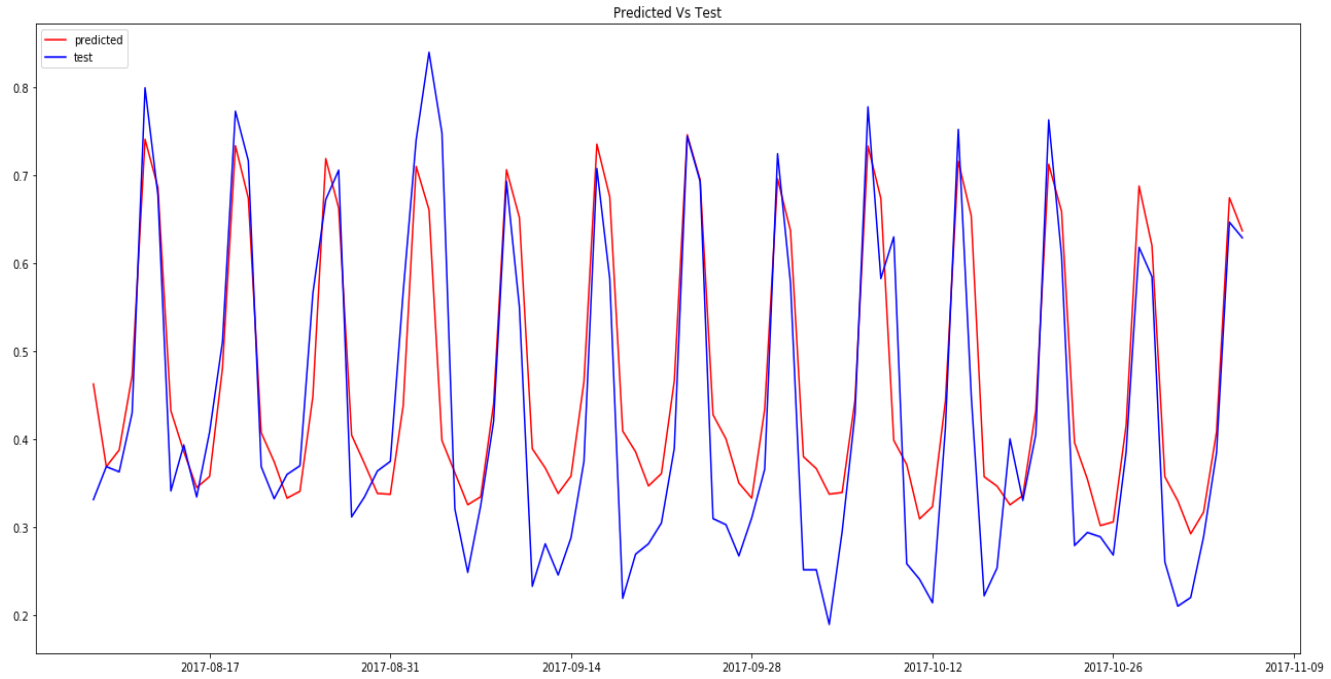
The plot of weather data of Toronto city is given as:



For the algorithm to run, it must be taken utmost care that each timestamp in the original data must coincide with that of weather data. After preprocessing and counterchecking the same, the following result gave us a green signal to go ahead.

```
length of weather data:  610
length of training data:  610
shape of weather data:  (610, 1)
shape of training data:  (610, 1)
```

For training we took first 520 points and rest 90 points to testing. In order to solve the problem of increased MSE we scaled down and the data. At last, test-prediction plot appeared as: -



Note that we have used the same best fit parameters that which we obtained during SARIMA implementation. The above prediction plot is more acceptable. Checking the MSE it was found to be `0.007974099704703067`. This tremendous fall in MSE can be attributed to scaling the data down.

In conclusion we can say that out of ARIMA, SARIMA and SARIMAX, the best model to predict the sales of restaurant is SARIMAX.

Chapter 4

Market Basket Analysis

This portion is too important because we are trying to uncover hidden patterns in the purchasing and sales behavior from the given data. Some the application of basket analysis are: -

1. Market basket analysis identifies the products often purchased together.
2. Market basket analysis uses association rules for finding patterns
3. Given many transactions we could find out which combinations occur frequently etc.

To understand the basket analysis, we need to understand some basic terms involved in it. They are explained below.

- Support (Prevalence):
  How frequent are item sets , consequent or antecedent purchased together
- Confidence (Predictability):
  Given a purchase of the antecedent how likely is the purchase of the consequent
- Lift (Interest):
  How much more likely is the association tan we would expect by chance.
  Generally when the Confidence and lift are higher , the products are more associated. Support can be calculated as follows:
- Support (item) = number of transactions with item / Total number of transactions
  Eg: Confidence can be calculated as follows
- Confidence (item31 => item 164) = number of transactions both (31 and 164) / number of transaction with 31
  Same thing can be written as = Support of (31 and 164) / support of 31
  Lift can be calculated as follows
- Lift (item 31 => item 164) = support of (31 and 164) / support of 34 * support of 164)

Most useful association rules are with high support and high lift. In our data set we applied association rules on 3 types of data sets. First data set contains all the transactions. The second contains transactions from 6th march 2016 to the ending. In the third data set we grouped the weekdays and weekends separately.

For the first dataset we transform the data set for computational convenience. The first few lines of the data set looks like this:

prod_idl	3	4	12	13	14	16	18	19	22	25	...	612	613	614	615	616	617	618	619	620	621
shdr_idl																					
1	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

The rows are shdr_idl means the transactions id. Total number of transactions are 150452. The columns represent prod_idl which is the product purchased. The total number of unique products are 515. The value 0 represent the product is not purchased and 1 is the product purchased in a particular transaction. The library we used in python is *mlxtend* which contains apriori and association rules algorithms. The first 20 items with high support are:

	support	Items
1	0.217604	(31)
59	0.210133	(163)
15	0.157027	(65)
44	0.098889	(110)
67	0.088121	(239)
7	0.086619	(56)
29	0.085363	(81)
68	0.084492	(242)
2	0.078145	(42)
24	0.066207	(74)
62	0.065795	(176)
49	0.065350	(124)
82	0.062764	(297)
64	0.062006	(178)
17	0.060219	(67)
5	0.057247	(54)
104	0.054263	(428)
13	0.051026	(63)
14	0.048979	(64)

Insight: 31 is the most frequent purchased item with a support of 0.217604

After applying association rules algorithm, the following is the result with highest *confidence* of first 20.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
377	(177)	(176)	0.009970	0.065795	0.009830	0.986000	14.985925	0.009174	66.728924
381	(180)	(178)	0.011186	0.062006	0.011027	0.985740	15.897365	0.010333	65.776795
522	(242, 301)	(54)	0.007730	0.057247	0.007557	0.977644	17.077499	0.007115	42.170045
504	(65, 301)	(54)	0.009545	0.057247	0.009299	0.974234	17.017932	0.008752	36.588989
498	(56, 301)	(54)	0.005816	0.057247	0.005636	0.969143	16.929000	0.005303	30.552165
512	(301, 110)	(54)	0.006640	0.057247	0.006414	0.965966	16.873507	0.006034	27.700287
516	(163, 301)	(54)	0.008069	0.057247	0.007730	0.957990	16.734184	0.007268	22.441207
423	(301, 31)	(54)	0.009598	0.057247	0.009159	0.954294	16.669614	0.008610	20.626282
358	(153)	(164)	0.005550	0.018903	0.005291	0.953293	50.430696	0.005186	21.005538
557	(297, 110)	(163)	0.005896	0.210133	0.005444	0.923337	4.394051	0.004205	10.303112
553	(65, 297)	(163)	0.005949	0.210133	0.005450	0.916201	4.360091	0.004200	9.425741
150	(301)	(54)	0.034237	0.057247	0.030422	0.888565	15.521471	0.028462	8.460136
494	(297, 31)	(163)	0.009339	0.210133	0.008248	0.883274	4.203395	0.006286	6.766844
357	(127)	(163)	0.011472	0.210133	0.009345	0.814600	3.876585	0.006935	4.260343
369	(295)	(163)	0.015314	0.210133	0.012403	0.809896	3.854197	0.009185	4.154914
372	(297)	(163)	0.062764	0.210133	0.048068	0.765858	3.644628	0.034880	3.373455
370	(296)	(163)	0.011120	0.210133	0.008169	0.734608	3.495914	0.005832	2.976231
210	(306)	(63)	0.011957	0.051026	0.007976	0.667037	13.072435	0.007366	2.850090
499	(56, 54)	(301)	0.008880	0.034237	0.005636	0.634731	18.539406	0.005332	2.643975
511	(110, 54)	(301)	0.010209	0.034237	0.006414	0.628255	18.350272	0.006064	2.597920

Insight: Person who ordered 177, have more chance of ordering 176 with a confidence of 0.986 which is highest in the data given.

After applying association rules algorithm, the following is the result with highest *lift* of first 20.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
358	(153)	(164)	0.005550	0.018903	0.005291	0.953293	50.430696	0.005186	21.005538
359	(164)	(153)	0.018903	0.005550	0.005291	0.279887	50.430696	0.005186	1.380965
499	(56, 54)	(301)	0.008880	0.034237	0.005636	0.634731	18.539406	0.005332	2.643975
502	(301)	(56, 54)	0.034237	0.008880	0.005636	0.164628	18.539406	0.005332	1.186442
511	(110, 54)	(301)	0.010209	0.034237	0.006414	0.628255	18.350272	0.006064	2.597920
514	(301)	(110, 54)	0.034237	0.010209	0.006414	0.187342	18.350272	0.006064	1.217968
505	(65, 54)	(301)	0.014982	0.034237	0.009299	0.620674	18.128848	0.008786	2.546000
508	(301)	(65, 54)	0.034237	0.014982	0.009299	0.271598	18.128848	0.008786	1.352300
517	(163, 54)	(301)	0.012702	0.034237	0.007730	0.608582	17.775648	0.007295	2.467344
520	(301)	(163, 54)	0.034237	0.012702	0.007730	0.225781	17.775648	0.007295	1.275219
522	(242, 301)	(54)	0.007730	0.057247	0.007557	0.977644	17.077499	0.007115	42.170045
527	(54)	(242, 301)	0.057247	0.007730	0.007557	0.132010	17.077499	0.007115	1.143181
509	(54)	(65, 301)	0.057247	0.009545	0.009299	0.162429	17.017932	0.008752	1.182533
504	(65, 301)	(54)	0.009545	0.057247	0.009299	0.974234	17.017932	0.008752	36.588989
498	(56, 301)	(54)	0.005816	0.057247	0.005636	0.969143	16.929000	0.005303	30.552165
503	(54)	(56, 301)	0.057247	0.005816	0.005636	0.098456	16.929000	0.005303	1.102757
513	(54)	(301, 110)	0.057247	0.006640	0.006414	0.112040	16.873507	0.006034	1.118699
512	(301, 110)	(54)	0.006640	0.057247	0.006414	0.965966	16.873507	0.006034	27.700287
521	(54)	(163, 301)	0.057247	0.008069	0.007730	0.135028	16.734184	0.007268	1.146779
516	(163, 301)	(54)	0.008069	0.057247	0.007730	0.957990	16.734184	0.007268	22.441207

We will consider the most useful associations rules with highest lift.
Insight: If we consider lift is as our metric the order is as above, then the person who purchased 153 have chances of 164 and vice versa with a lift of 50.4.

The second data set is the sliced data set which contains transactions after 6th March 2016. The transformation of the data set as per our convenience looks as follows:

prod_idl	12	13	31	42	43	51	53	54	55	56	...	612	613	614	615	616	617	618	619	620	621
shdr_idl																					
31423	0	0	0	1	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
31424	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31425	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31426	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31427	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

The total number of transactions are 121443 rows × 401 columns. It means 401 unique items are there. The first 20 items with highest support are given aside.

(PTO)

	support	itemsets
1	0.217386	(31)
58	0.200958	(163)
15	0.152730	(65)
67	0.103563	(242)
44	0.101233	(110)
66	0.089268	(239)
7	0.085555	(56)
29	0.082351	(81)
2	0.078728	(42)
63	0.073351	(178)
49	0.064639	(124)
61	0.064079	(176)
24	0.064005	(74)
80	0.063627	(297)
5	0.056973	(54)
17	0.055936	(67)
102	0.055302	(428)
13	0.050089	(63)
14	0.048212	(64)
306	0.047660	(297, 163)

After applying the associations rules on the 2nd dataset the results are as follows the first 20 items with highest lift are:

:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
512	(301)	(56, 54)	0.033999	0.008753	0.005583	0.164204	18.759620	0.005285	1.185992
509	(56, 54)	(301)	0.008753	0.033999	0.005583	0.637817	18.759620	0.005285	2.667165
524	(301)	(110, 54)	0.033999	0.010375	0.006464	0.190119	18.324271	0.006111	1.221938
521	(110, 54)	(301)	0.010375	0.033999	0.006464	0.623016	18.324271	0.006111	2.562443
518	(301)	(65, 54)	0.033999	0.014385	0.008893	0.261565	18.182703	0.008404	1.334734
515	(65, 54)	(301)	0.014385	0.033999	0.008893	0.618203	18.182703	0.008404	2.530139
530	(301)	(163, 54)	0.033999	0.012261	0.007460	0.219424	17.896212	0.007043	1.265397
527	(163, 54)	(301)	0.012261	0.033999	0.007460	0.608462	17.896212	0.007043	2.467195
532	(242, 301)	(54)	0.009568	0.056973	0.009354	0.977625	17.159371	0.008809	42.146043
537	(54)	(242, 301)	0.056973	0.009568	0.009354	0.164186	17.159371	0.008809	1.184990
519	(54)	(65, 301)	0.056973	0.009115	0.008893	0.156092	17.124003	0.008374	1.174162
514	(65, 301)	(54)	0.009115	0.056973	0.008893	0.975610	17.124003	0.008374	38.664098
508	(56, 301)	(54)	0.005772	0.056973	0.005583	0.967190	16.976214	0.005254	28.741816
513	(54)	(56, 301)	0.056973	0.005772	0.005583	0.097991	16.976214	0.005254	1.102237
522	(301, 110)	(54)	0.006694	0.056973	0.006464	0.965560	16.947602	0.006083	27.381456
523	(54)	(301, 110)	0.056973	0.006694	0.006464	0.113456	16.947602	0.006083	1.120424
526	(163, 301)	(54)	0.007757	0.056973	0.007460	0.961783	16.881322	0.007018	24.675867
531	(54)	(163, 301)	0.056973	0.007757	0.007460	0.130944	16.881322	0.007018	1.141748
430	(54)	(301, 31)	0.056973	0.009552	0.009165	0.160861	16.840940	0.008621	1.180315
427	(301, 31)	(54)	0.009552	0.056973	0.009165	0.959483	16.840940	0.008621	23.274703

Insight: If we consider lift is as our metric, then the person who ordered 301 have chances of ordering {56,54}and vice versa with a lift of 18.7.

The Third data set is the sliced data set which contains all transactions and the divided into two parts. The first part is transactions contain weekdays (Monday to Friday) and second part contains transactions with weekends (Saturday and Sunday).

The total number of transactions with weekends are 177052.
The total number of transactions with weekdays are 280410.

The transformed dataset with weekdays as follows. There are 396 unique items present in this.

(PTO)

prod_idl	242	244	252	51	245	248	260	279	253	42	...	613	614	285	590	263	405	621	612	424	268
shdr_idl																					
31622	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31623	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31624	0	0	1	1	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31625	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31626	0	0	0	0	0	0	1	1	0	0	...	0	0	0	0	0	0	0	0	0	0

After applying association rules on the items with highest support the results are shown aside:

	support	itemsets
2	0.221014	(163)
11	0.187668	(31)
7	0.166206	(65)
0	0.099233	(242)
14	0.094189	(81)
5	0.093411	(110)
10	0.089132	(56)
1	0.087161	(42)
3	0.084760	(239)
16	0.069804	(74)
6	0.069750	(124)
22	0.069120	(297)
4	0.062091	(67)
9	0.060602	(428)
12	0.057611	(54)
8	0.053694	(255)
25	0.051950	(297, 163)
15	0.051830	(63)
20	0.051481	(361)
21	0.048168	(64)

Insights: The items 163,31,65,242, 81 are more frequent to be ordered in the restaurant.

(PTO)

The transformed dataset with weekends as follows:

prod_idl	51	81	163	57	76	110	242	278	300	428	...	432	263	424	602	528	249	389	426	556	589
shdr_idl																					
31424	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31425	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31426	0	0	1	1	1	1	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31427	0	0	0	0	0	0	0	1	1	1	...	0	0	0	0	0	0	0	0	0	0
31428	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31429	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
31430	0	1	0	0	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	

There are 365 unique items in weekends data set. After applying association rules the items with highest support are:

Insights: In weekends the more frequent items ordered are 31,163,176,178,65,110

(PTO)

	support	itemsets
6	0.264641	(31)
1	0.169055	(163)
17	0.165899	(176)
19	0.144871	(178)
13	0.131307	(65)
2	0.113670	(110)
3	0.110450	(242)
16	0.096438	(239)
14	0.079868	(56)
21	0.073363	(176, 31)
5	0.065302	(42)
0	0.063532	(81)
7	0.056515	(124)
12	0.055939	(54)
18	0.054894	(297)
11	0.054788	(74)
15	0.048283	(64)
8	0.047324	(63)
4	0.046876	(428)
22	0.046343	(178, 31)

Chapter 5

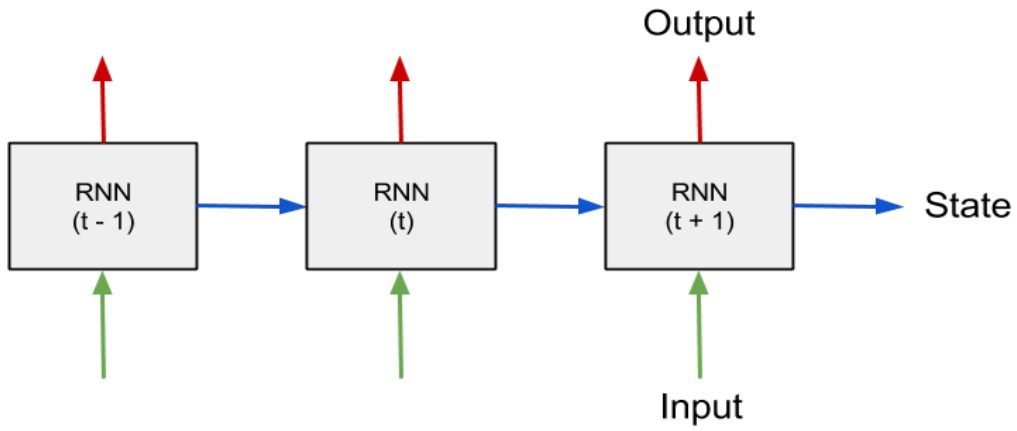
Future Scope

With the recent breakthroughs that have been happening in data science, it is found that for almost all of these sequence prediction problems, Long short Term Memory networks, a.k.a LSTMs have been observed as the most effective solution.

Long short-term memory (LSTM) is an artificial recurrent neural network, (RNN) architecture used in the field of deep learning. LSTMs have an edge over conventional feed-forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time.

The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using backpropagation through time and overcomes the vanishing gradient problem. As such, it can be used to create large recurrent networks that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results. Instead of neurons, LSTM networks have memory blocks that are connected through layers.

Since RNN is a type of neural network where the output from previous steps is fed as input to the current step**,** it is a good option to use LSTM for sales prediction. Also, neural networks are so effective in multivariate analysis because we can input each and every attribute in our data set as single neurons. Through this we can design a deep neural network framework and predict the output. So, this LSTM prediction method will be so effective as compared to other methods.



In the above figure, the output from time t-1 and t becomes the input and predicts the output for time t+1. This is the simple mechanism of RNN.

Also, we can push categorical variables present in our dataset as numerical values through one hot encoding. *One hot encoding* is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. Through one hot encoding we are expected to get a better prediction in LSTM. But before that we have to remove the outliers in data, else the prediction will be biased. So, from different perspectives we can say that LSTM can be determined as the future scope for our sales prediction.

----------THE END----------

References and links

- An Introductory Study on Time Series Modeling and Forecasting, Ratnadip Adhikari and R. K. Agrawal.
- Introduction to Time Series and Forecasting, Second Edition, Peter J. Brockwell, Richard A. Davis, Springer.
- https://www.datacamp.com/courses/manipulating-dataframes-with-pandas
- https://www.datacamp.com/courses/forecasting-using-arima-models-in-python
- https://www.datacamp.com/courses/manipulating-time-series-data-in-python
- https://www.datacamp.com/courses/visualizing-time-series-data-in-python
- https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
- https://www.researchgate.net/publication/299459188_Time_Series_Analysis_Forecasting_and_Control5th_Edition_by_George_E_P_Box_Gwilym_M_Jenkins_Gregory_C_Reinsel_and_Greta_M_Ljung_2015_Published_by_John_Wiley_and_Sons_Inc_Hoboken_New_Jersey_pp_712_ISBN_
- https://machinelearningmastery.com/sarima-for-time--forecasting-iseriesn-python/
- https://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-/python
- Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
- A Beginner's Guide to LSTMs and Recurrent Neural Networks | Skymind
- (PDF) Predicting Activities in Business Processes with LSTM Recurrent Neural Networks
- https://www.youtube.com/watch?v=UNmqTiOnRfg
- https://www.youtube.com/watch?v=2np77NOdnwk&t=319s
- https://www.datacamp.com/courses/introduction-to-time-series-analysis-in-python