



Time Series prediction of Retail dataset Report

MCDA 5580



Team member:

Mitkumar Patel A00444857

Wensho Li A00445457

Chirag Panasuriya A00442907

Contents

1.	Executive Summary	1
2.	Objective	1
3.	About Data	1
4.	Data Preparation	2
5.	Design/Methodology/Approach.....	4
6.	Statistical	5
7.	Neural Network	7
8.	Support Vector Regression.....	7
9.	Comparison of techniques.....	8
10.	Appendix	8
11.	Reference:.....	10

1. Executive Summary

A prediction and projection of different products' daily consumption which will help in business forecasting, future planning and understanding of past behavior. Time series analysis is done based on past pattern of data and attempt to predict the future based upon the underlying patterns contained within those data.

This report is based on time series analysis of top six bestselling (by quantity) products from the retail dataset to predict the daily consumption. Various trend models of time series analysis were discussed with a view of showing understanding to the appropriate method to be used for forecasts. Build that information into future for everything from product demand to inventory planning and staffing.

2. Objective

“Prediction is truly very difficult, especially if it’s about the unknown future” – Nils Bohr, Nobel laureate in Physics

Forecasting is the science of predicting the future. By using data historical data, businesses can understand trends, make a call on what might happen and when. Given the consequences of forecasting, accuracy matters. If the forecast is much high, then company may over invest in those products which can be wrong investment. But If forecast is too low then company may under invest and that can lead to shortage of raw material and human power.

We want to predict daily consumption in the future based on given data of past, for this we require some trainable model.

Forecasting is a hard problem where accuracy really matters and exogenous factors that come into play that are hard to account for. It looks in the dataset for features such as trends, cyclical fluctuations, seasonality, and behavioral patterns.

3. About Data

The original dataset is stored in dataset01.sales219, which includes information about order date, customer information and products quantity etc.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 TRANSACTION_RK	varchar(10)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	2 CALENDAR_DT	varchar(9)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 date	date			No	None			Change Drop More
<input type="checkbox"/>	4 time	time			No	None			Change Drop More
<input type="checkbox"/>	5 TRANSACTION_TM	varchar(8)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 ITEM_SK	varchar(20)	utf8_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 RETAIL_OUTLET_LOCATION_SK	int			Yes	NULL			Change Drop More
<input type="checkbox"/>	8 POS_TERMINAL_NO	int			Yes	NULL			Change Drop More
<input type="checkbox"/>	9 CASHIER_NO	int			Yes	NULL			Change Drop More
<input type="checkbox"/>	10 ITEM_QTY	int			Yes	NULL			Change Drop More
<input type="checkbox"/>	11 ITEM_WEIGHT	decimal(4,3)			Yes	NULL			Change Drop More
<input type="checkbox"/>	12 SALES_UOM_CD	varchar(1)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	13 SELLING_RETAIL_AMT	decimal(6,5)			Yes	NULL			Change Drop More
<input type="checkbox"/>	14 PROMO_SALES_IND_CD	varchar(1)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	15 STAPLE_ITEM_FLG	varchar(1)	utf8_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	16 REGION_CD	int			Yes	NULL			Change Drop More
<input type="checkbox"/>	17 CUSTOMER_SK	varchar(20)	utf8_unicode_ci		Yes	NULL			Change Drop More

Diagram 1 Sales219 Column

4. Data Preparation

Extract dataset for time series analysis:

- 1) Query the top 6 selling products based on quantity.

```
SELECT ITEM_SK, SUM(ITEM_QTY) FROM `sales219`
GROUP BY ITEM_SK
ORDER BY SUM(ITEM_QTY) DESC
LIMIT 6;
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT ITEM_SK, SUM(ITEM_QTY) FROM `sales219`
2 GROUP BY ITEM_SK
3 ORDER BY SUM(ITEM_QTY) DESC
4 LIMIT 6;
```

The results are displayed in a table:

ITEM_SK	sum(ITEM_QTY)
11740941	90239
11740923	61482
11741274	28261
11636550	23839
11629829	22338
11743201	20716

- 2) Based on result of step 1, query product separately and create table.
As sum of last A number of all team members is odd number, we take the first, third and fifth selling products as source of data. Following are SQL statement of creating table:




```
CREATE TABLE w_li.firstproduct AS SELECT
    date,
    ITEM_SK,
    SUM(ITEM_QTY) AS quant
FROM
    dataset01.sales219
WHERE
    ITEM_SK = '11740941'
GROUP BY
    date,
```

```
ITEM_SK;
```

```
CREATE TABLE w_li.thirdproduct AS SELECT
  date,
  ITEM_SK,
  SUM(ITEM_QTY) AS quant
FROM
  dataset01.sales219
WHERE
  ITEM_SK = '11741274'
GROUP BY
  date,
  ITEM_SK
ORDER BY
  ITEM_SK,
  date;
```

```
CREATE TABLE w_li.fifthproduct AS SELECT
  date,
  ITEM_SK,
  SUM(ITEM_QTY) AS quant
FROM
  dataset01.sales219
WHERE
  ITEM_SK = '11629829'
GROUP BY
  date,
  ITEM_SK
ORDER BY
  ITEM_SK,
  date;
```

- 3) Extract datasets
Exporting selling data separately and got three datasets.

Name			
 fifthproduct.csv			
 firstproduct.csv			
 thirdproduct.csv			

	A	B	C
1	2015-01-02	11740941	430
2	2015-01-03	11740941	433
3	2015-01-04	11740941	305
4	2015-01-05	11740941	469
5	2015-01-06	11740941	533
6	2015-01-07	11740941	273
7	2015-01-08	11740941	279
8	2015-01-09	11740941	292
9	2015-01-10	11740941	399
10	2015-01-11	11740941	362
11	2015-01-12	11740941	315
12	2015-01-13	11740941	579
13	2015-01-14	11740941	364
14	2015-01-15	11740941	388

5. Design/Methodology/Approach

There are different types of Prediction Methods like Qualitative techniques, Time series analysis and Causal models.

Time series involve historical data, finding structure of the data like trends, growth rates and cyclical patterns.

We will focus on the time series analysis approach which has been the motivation behind conventional forecasting methods, and it can give an extensive formation of the forecasting aspect.

There are 5 most used algorithms in the different industry. All algorithms work towards getting important characteristics for forecasting purposes.

1. Autoregressive (AR)

- Autoregressive models extract the behavioral pattern of the past data in order to do time series forecasting of future trends.

2. Moving Average (MA)

- The Moving Average uses past forecasted errors (or noise) in a regression-like model to explain an averaged trend in the data.

3. Autoregressive Moving Average (ARMA)

- It is simply the combination of Autoregressive and Moving Average where AR extract the trends from past data and MA collect the Noise effects.

4. Autoregressive Integrated Moving Average (ARIMA)

- ARIMA is the most used algorithms in Time Series prediction. While other models describe the trend and seasonality of the data points, ARIMA aims to explain the autocorrelation between the different data points.

5. Exponential Smoothing (ES)

- It is the technique for smoothing time series data using the exponential window function. It should only be used for time series with no systematic trend and/or seasonal components. Holt-Winters is a model of time series behavior

The three key primary ideas that are fundamental to consider, when dealing with a forecasting problem handled with a time series analysis, are:

1. Repeating patterns
 - One element that we are looking for is a pattern that repeats in time. One key concept related to this idea is similarity between observations as a function of the time lag between them
 - It is like finding the correlation between two different time points which is basic tool of finding repeating patterns
2. Static patterns
 - The aim behind static pattern relates to the something that does not change with time
3. Trends
 - A trend represents a likelihood identified in our data. Identifying a trend help us to know the direction that our time-series is heading, which is basic for predicting the future of sales

6. Statistical

In this case, we use liner regression as statistical time series prediction.

The R scripts is showing as following diagram, it works for all three datasets.

```
> # Linear regression
> glmFitTime <- train(V8 ~ .,
+                      data = xy,
+                      method = "glm",
+                      preProc = c("center", "scale"),
+                      tuneLength = 10,
+                      trControl = myCvControl)
> glmFitTime
```

Diagram Liner Regression

For the first product, the model generates 31 samples and 7 predictors. It also got root mean square error of 102.3097, Rsquared 0.563481 and Mean Absolute Error of 81.85727.

Following table shows the accuracy range of three different top selling products.

	First Selling Product	Third Selling Product	Fifth Selling Product
Accuracy Range	300.5261	367.1792	58.5807

Generalized Linear Model

31 samples

7 predictor

Pre-processing: centered (7), scaled (7)

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 28, 28, 28, 28, 29, 28, ...

Resampling results:

RMSE	Rsquared	MAE
102.3097	0.563481	81.85727

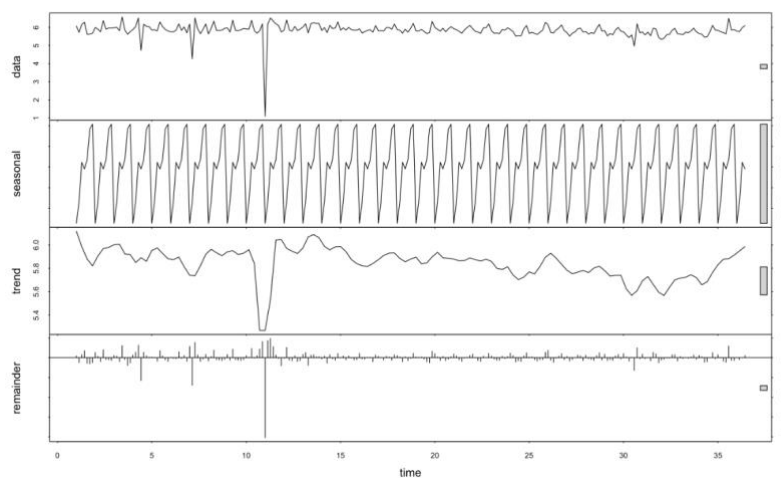
Result of linear regression prediction

And we can also get which point contribute most to the prediction:

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	349.94	16.04	21.814	< 2e-16	***
V1	59.05	18.89	3.125	0.00476	**
V2	49.32	18.30	2.696	0.01290	*
V3	-30.60	19.48	-1.571	0.12989	
V4	-12.72	17.63	-0.721	0.47794	
V5	-16.86	19.58	-0.861	0.39803	
V6	-12.41	20.62	-0.602	0.55325	
V7	-17.99	18.80	-0.957	0.34861	

And we also got the seasonal decomposition diagram:



Seasonal Decomposition Diagram of Top1 Product

7. Neural Network

We also implement neural network for this time series prediction. R scripts showing as following diagram. Since the structure of three datasets are all the same, this script is work on top 1,3,5 selling products.

```
# Neural Network
nnFitTime <- train(V8 ~ .,
  data = xy,
  method = "avNNet",
  preProc = c("center", "scale"),
  trControl = myCvControl,
  tuneLength = 10,
  linout = T,
  trace = F,
  MaxNWts = 10 * (ncol(xy) + 1) + 10 + 1,
  maxit = 500)
```

Neural Network Rscript

```
> y_hat = predict(nnFitTime, newdata = x)
> mean(100*abs(y_hat-y)/y)
[1] 189.6344
```

Output of Neural Network Prediction

And following table shows the result of three datasets prediction:

	First Selling Product	Third Selling Product	Fifth Selling Product
Accuracy Range	189.6344	379.0096	59.65719

8. Support Vector Regression

We also implement support vector regression for this time series prediction. R scripts showing as following diagram. Since the structure of three datasets are all the same, this script is work on top 1,3,5 selling products.

```
# Support Vector Regression
svmFitTime <- train(V8 ~ .,
                    data = xy,
                    method = "svmRadial",
                    preProc = c("center", "scale"),
                    tuneLength = 10,
                    trControl = myCvControl)

svmFitTime
summary(svmFitTime)
y_hat = predict(svmFitTime, newdata = x)
mean(100*abs(y_hat-y)/y)
```

Following table shows the accuracy of prediction on these three top selling products.

	First Selling Product	Third Selling Product	Fifth Selling Product
Accuracy Range	332.9933	327.1906	41.48291

9. Comparation of techniques

Here we already implement all three Time Series prediction, it is time to make comparison among these three methods.

	First Selling Product	Third Selling Product	Fifth Selling Product
Statistic Accuracy Range	300.5261	367.1792	58.5807
Neural Network Accuracy	189.6344	379.0096	59.65719
Support Vector Regression Accuracy	332.9933	327.1906	41.48291

From above table we could know that for top1 selling product, Neural Network has the best performance. For the third selling product, Support Vector Regression has the best performance. And for the fifth selling product, Support Vector Regression has the best performance.

If we do not take the dataset difference in consideration, Neural Network has the best performance among the product selling prediction.

10. Appendix

SQL Statement for extracting dataset:

```
SELECT ITEM_SK, sum(ITEM_QTY) FROM `sales219`
GROUP BY ITEM_SK
ORDER BY SUM(ITEM_QTY) DESC
LIMIT 6;
```

```

CREATE TABLE w_li.thirdproduct AS SELECT
    date,
    ITEM_SK,
    SUM(ITEM_QTY) AS quant
FROM
    dataset01.sales219
WHERE
    ITEM_SK = '11741274'
GROUP BY
    date,
    ITEM_SK
ORDER BY
    ITEM_SK,
    date;

```

R Script:

```

library(caret)

xy=read.table("firstproduct_hs.csv",sep=',',header=F,fileEncoding="UTF-8-BOM")
y=xy[,8]
head(y)
x=xy[,1:7]

# Using pre-sliced data
myCvControl <- trainControl(method = "repeatedCV",
                           number=10,
                           repeats = 5)

# Linear regression
glmFitTime <- train(V8 ~ .,
                   data = xy,
                   method = "glm",
                   preProc = c("center", "scale"),
                   tuneLength = 10,
                   trControl = myCvControl)
glmFitTime
summary(glmFitTime)
y_hat = predict(glmFitTime, newdata = x)
mean(100*abs(y_hat-y)/y)

# Support Vector Regression
svmFitTime <- train(V8 ~ .,
                   data = xy,
                   method = "svmRadial",
                   preProc = c("center", "scale"),
                   tuneLength = 10,
                   trControl = myCvControl)
svmFitTime
summary(svmFitTime)
y_hat = predict(svmFitTime, newdata = x)
mean(100*abs(y_hat-y)/y)
# Your error with support vector regression

# Neural Network
nnFitTime <- train(V8 ~ .,
                  data = xy,
                  method = "avNNet",
                  preProc = c("center", "scale"),
                  trControl = myCvControl,
                  tuneLength = 10,

```

```

        linout = T,
        trace = F,
        MaxNWts = 10 * (ncol(xy) + 1) + 10 + 1,
        maxit = 500)
nnFitTime
summary(nnFitTime)
y_hat = predict(nnFitTime, newdata = x)
mean(100*abs(y_hat-y)/y)
# Your error with neural networks

# You can experiment with other methods, here is where you can find the methods caret supports:
# https://topepo.github.io/caret/available-models.html

# Compare models
resamps <- resamples(list(lm = glmFitTime,
                        svn = svmFitTime,
                        nn = nnFitTime))
summary(resamps)

# Now working with the time-series modeling

t=read.csv("firstproduct.csv",header=T )
tSeries<-ts(t[,1],frequency = 7)
head(tSeries)
plot(t)
head(tSeries)

library(forecast)
hw = ets(tSeries,model="MAM")
mean(100*abs(fitted(hw) - tSeries)/tSeries)
# Your Holt-Winters error

ar <- Arima(tSeries,order=c(7,0,7))
mean(100*abs(fitted(ar) - tSeries)/tSeries)
# Your Arima error

```

11. Reference:

<http://dev.cs.smu.ca/~pawan/5580/assignments/assign3PredictionSample.txt>

<https://youtu.be/3kq0QTTeqOk>

<https://youtu.be/5eDEF4LOJEI>

<https://smu.brightspace.com/d2l/le/content/99661/viewContent/832210/View>