Ecommerce Cap Stone Project Student Working Handbook - 1st Sept, 2018

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

Packages that you would need for analysis

library(dplyr), library(plyr), library(lubridate), library(MASS), library(car), library(DataCombine), library(zoo), library(glmnet), library(DAAG), library(ggplot2)

Data Sanity Check

First, you should make proper names for the columns after reading the data file.

```
colnames(ecom) <-
c("FSN_ID","Order_date","Year","Month","Order_id","Order_item_id","gmv","Units","deliverybdays","deliverycdays",
"payment_mode","SLA","cust_id","pincode","P_super_category","P_analytic_category","P_sub_category","P_analytic_vertical","
MRP","Procurement_SLA")
```

Step 1 - Data sanity Check: Data should be from July-2015 to June -2016

```
ecom <- subset(ecom, !(Month ==5 & Year==2015|Month ==6 & Year==2015|Month ==7 & Year==2016))
```

Step 2 – Add a week column in the dataset. We have to convert the data set in a weekly level. Check if the date formats are correct or not.

Step 3 - Inspect Data quality issues, for example - Jan 2016 should be week 54, not week 1 etc.

```
ecom$week_year<- ifelse(ecom$week_year<=26  & ecom$Year==2016,ecom$week_year+53,ecom$week_year)
```

Step 4 – We should not considering free products

```
ecom <- subset(ecom,MRP!=0)
```

Step 5 - removing rows with NA values

Step 6 – Make “gmv” =1 if they are 0

Step 7 - gmv should not be more than MRP*units since we can offer discounts but should not charge higher. So, we will take a subset such that (MRP*Units)>=gmv

Step 8 – Divide the data in 3 buckets, for example -

```
GamingAccessory <- ecom[ecom$P_sub_category=="GamingAccessory" ,]
```

You need to remember all of the below exercise need to be performed 3 times – for Home Audio, Gaming and Camera. Therefore, it is a good idea to write a function where you will do the following –

1. Prepare KPIs
2. Perform Clustering
3. Data aggregation as and when required
4. Analyzing NPS data
5. Using Holiday information
6. Merging master data with holiday data

Preparing the KPIs

1. KPI - List price for all the products

```
dataset$list_price <- dataset$gmv/dataset$Units
```

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

2. KPI - Promotional Offer for all the products

```
dataset$promotional_offer <- (dataset$MRP-dataset$list_price)/dataset$MRP
```

3. We will perform a clustering exercise - It divides products into three categories based on MRP and num units (Units) and List Price - mass market, medium market and premium product.

```
dataset$P_analytic_vertical <- factor(dataset$P_analytic_vertical)

cluster<- aggregate(cbind(Units,list_price,MRP)~P_analytic_vertical,dataset,mean)
```

Perform a K-Means clustering by scaling all above 3 variables.

```
cluster$list_price_1 <- scale(cluster$list_price)

cluster$MRP_1<- scale(cluster$MRP)

cluster$Units_1 <- scale(cluster$Units)
```

Bring the 3-Cluster solution cluster # in the master dataset by merging with P_analytic_vertical as the key. Name "Mass_p", "Premium_p", "Middle_p" by using below formula -

```
clust <- kmeans(k1[,-1], centers = 3,iter.max = 50,nstart = 50)

cluster$P_tag <- as.factor(clust$cluster)

dataset <-merge(dataset,cluster,by=c("P_analytic_vertical"),all.x=TRUE)

k2 <- table(dataset$P_tag)

levels(dataset$P_tag)[which(k2==max(table(dataset$P_tag)))] <- "Mass_p"

levels(dataset$P_tag)[which(k2==min(table(dataset$P_tag)))] <- "Premium_p"

levels(dataset$P_tag)[which(k2!=max(table(dataset$P_tag))& k2!=min(table(dataset$P_tag)))] <- "Middle_p"
```

4. KPI - Payment model indicator

```
dataset$order_pay_ind<-ifelse(dataset$payment_mode=="Prepaid",1,0)
```

5. Create Percentage Online Order = (Online_order/Total Order) KPI

```
total_order <-aggregate(order_pay_ind ~ week_year,data=dataset,FUN=NROW)
```

Total online order

```
Online_order<-aggregate(order_pay_ind ~ week_year,data=dataset,FUN=sum)
```

Merge the both the file

```
merged <-merge(total_order,Online_order,by=c("week_year"),all.x=TRUE)
```

Create new column of percentage online

```
merged$per_order <- merged$order_pay_ind.y/merged$order_pay_ind.x
```

Add this " per_order" information to the main dataset.

```
dataset<- merge(dataset,merged,by=c("week_year"),all.x=TRUE)
```

Drop columns that are redundant.

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

Understanding NPS data

Read this file and bring "nps" information in the master data. If you need to convert "Month" into characters, please do that.

Merge "NPS" data –

```
dataset<-merge(dataset,nps,by=c("Month","Year"),all.x=TRUE)
```

Start Merging

Prepare a holiday calendar based on the holiday list that you have. You can use the below list to prepare the holiday data.

```
holiday_list<-c("2015-07-18","2015-07-19","2015-08-15",
                "2015-08-16","2015-08-17","2015-08-28",
                "2015-08-29","2015-08-30","2015-10-15",
                "2015-10-16","2015-10-17","2015-11-07","2015-11-08","2015-11-09","2015-11-10",
                "2015-10-11","2015-10-12","2015-11-13","2015-11-14","2015-12-25","2015-12-26",
                "2015-12-27","2015-12-28","2015-12-29","2015-12-30","2016-01-01","2016-01-02",
                "2016-01-03","2016-01-20","2016-01-21","2016-01-22","2016-02-01","2016-02-02",
                "2016-02-20","2016-02-21","2016-02-14","2016-02-15","2016-03-07","2016-03-08",
                "2016-03-09","2016-05-25","2016-05-26","2016-05-27")
```

You should aggregate this data at a **weekly** level with a holiday frequency (sum would be used to aggregate).

```
holiday <- aggregate( holiday_freq ~ week_year,holiday,sum)
```

Prepare a "Product" table. This is a **weekly** distribution of "Premium", "Mass Market" and "Aspiring" products.

```
products <- as.data.frame.matrix(t(table(dataset$P_tag,dataset$week_year)))

products$week_year <- row.names(products)
```

Bring your master data in a **weekly** level by 2 steps -

```
dataset_1 <- aggregate(gmv~week_year,dataset,sum)

dataset<- aggregate(cbind(list_price,MRP,Units,SLA,promotional_offer,Procurement_SLA,per_order,NPS)~
week_year,data=dataset,FUN = mean)
```

And now merge your data.

```
dataset <- merge(dataset,products,by="week_year",all.x=TRUE)

dataset <- merge(dataset,holiday,by="week_year",all.x=TRUE)

dataset$holiday_freq[is.na(dataset$holiday_freq)] <-0
```

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

dataset <- merge(dataset,dataset_1,by="week_year",all.x=TRUE)

Your data is ready for further processing. Your data should look like below for all 3 categories -

Camera

Filter

week_year	list_price	MRP	Units	SLA	promotional_offer	Procurement_SLA	per_order	NPS	Middle_p	Premium_p	Mass_p	holiday_freq	gmv
27	1284.2710	2333.700	1.046058	6.245350	0.4551645	2.777679	0.3427812	54.60000	0	0	1129	0	1473817
28	1416.5405	2517.749	1.042819	7.173913	0.4486012	2.737813	0.3306983	54.60000	0	0	3036	0	4358995
29	1497.3903	2610.180	1.029946	6.224764	0.4486992	2.843203	0.3358008	54.60000	0	0	2972	2	4501054
30	1450.6863	2527.312	1.042764	6.362641	0.4424304	2.800889	0.3185084	54.60000	0	0	2923	0	4327606
31	1366.6184	2458.444	1.055627	6.525226	0.4492055	2.750323	0.2820181	54.61397	0	0	773	0	1075220
32	699.0000	1450.000	1.000000	5.000000	0.5179310	4.000000	0.0000000	60.00000	0	0	2	0	1398
33	299.0000	1990.000	1.000000	6.000000	0.8497487	1.000000	0.0000000	60.00000	0	0	2	3	598
35	1312.5791	2383.703	1.035889	6.588363	0.4501227	2.789016	0.3001631	47.09233	4	0	1835	3	2450575
36	1341.8001	2413.950	1.044896	6.549061	0.4523243	2.875476	0.3084851	46.90000	30	0	7053	0	9623138
37	1475.5241	2685.741	1.028378	6.259716	0.4551910	2.914559	0.3001234	46.90000	10	0	6474	0	9702998
38	1432.7277	2557.777	1.035714	6.684630	0.4558877	2.907844	0.2975128	46.90000	50	0	6222	0	9159720
39	1163.1088	2526.245	1.026615	5.561181	0.5779971	2.802986	0.2188683	46.90000	40	0	9203	0	10869781
40	1148.6627	2260.168	1.031737	6.251298	0.5006382	2.693306	0.2437969	44.40000	25	0	3441	0	4069029
41	1334.8419	2396.715	1.024652	6.265541	0.4618820	2.762326	0.2384780	44.40000	27	0	3705	2	5038194
42	1226.9765	2436.153	1.030812	6.459510	0.5396304	3.003950	0.2901442	44.40000	29	0	10097	3	12553013
43	1207.7712	2188.498	1.034015	6.513445	0.4721591	2.954493	0.2254654	44.40000	17	0	4334	0	5358427
44	1248.0832	2344.217	1.031685	6.746244	0.4629146	2.776291	0.2198853	45.85731	29	0	3632	0	4638693
45	1218.5492	2259.335	1.023894	6.163524	0.4898203	2.838156	0.2329662	47.00000	22	0	5335	4	6591549
46	1264.6713	2374.426	1.022222	6.463272	0.4718895	2.766358	0.2305556	47.00000	29	0	3211	2	4129894
47	1348.9257	2483.049	1.034215	6.096423	0.4603084	2.728149	0.2438569	47.00000	24	0	3191	0	4415533
48	1540.2064	2800.291	1.017908	6.333428	0.4477852	2.830586	0.2345082	46.62103	17	0	3501	0	5453831
49	1565.1460	2789.050	1.025217	6.089835	0.4769302	2.752824	0.2503283	45.80000	39	0	3768	0	6025603

Gaming Accessory

week_year	list_price	MRP	Units	SLA	promotional_offer	Procurement_SLA	per_order	NPS	Middle_p	Premium_p	Mass_p	holiday_freq	gmv
26	937.3333	3232.3333	1.000000	6.000000	0.5902928	3.333333	1.0000000	54.60000	0	0	3	0	2812
27	779.5393	1808.1081	1.079195	4.179761	0.4487434	3.261471	0.2840981	54.60000	504	8	1079	0	1309103
28	971.4813	1879.7360	1.033571	6.314729	0.3509990	2.611414	0.2429710	54.60000	608	28	1747	0	2376077
29	821.9946	1945.1622	1.092287	4.944559	0.4352054	2.526860	0.2431129	54.60000	585	18	2301	2	2463306
30	1051.2692	3564.3825	1.045269	5.349103	0.4086302	2.588499	0.2100326	54.60000	745	17	1690	0	2657359
31	1017.9913	4849.1647	1.046854	5.842035	0.4411685	2.990629	0.2329317	54.62892	284	7	456	0	798571
32	986.0000	2051.3333	1.000000	6.500000	0.4672656	2.666667	0.3333333	60.00000	1	0	5	0	5916
34	292.3333	657.3333	1.000000	3.833333	0.5357355	1.833333	0.0000000	60.00000	0	0	6	0	1754
35	828.8533	1784.8292	1.053089	5.101351	0.3925386	2.563707	0.2084942	47.07703	548	14	1510	3	1781232
36	862.3750	1947.8131	1.042870	5.254916	0.3843096	2.707174	0.2177148	46.90000	1290	22	4333	0	4971043
37	898.5267	1802.3998	1.047190	5.366392	0.3980831	2.570305	0.2230961	46.90000	1418	28	3767	0	4768542
38	1008.9675	2883.3848	1.042660	5.956085	0.3915196	2.680887	0.2183187	46.90000	1414	30	3338	0	4958599
39	933.6036	2648.2866	1.033868	5.912299	0.4208391	2.552941	0.1853832	46.90000	1958	30	3622	0	5336080
40	927.2027	2331.8775	1.041374	5.601572	0.3712698	2.697145	0.1795614	44.40000	561	15	1841	0	2325025
41	788.5406	1743.5583	1.036594	5.788043	0.3983330	2.456159	0.1692029	44.40000	856	8	1896	2	2240460
42	654.1115	4865.4254	1.037365	5.248086	0.5305525	3.480487	0.2116431	44.40000	4184	19	6636	3	7237460
43	798.4464	1750.2194	1.032851	5.863773	0.4329335	2.280892	0.1720916	44.40000	1383	12	1923	0	2721489
44	697.8961	1580.7222	1.054129	5.545706	0.4488646	2.182546	0.1648716	45.96388	1281	5	2335	0	2629582
45	769.3689	1668.7261	1.039908	5.266514	0.4447838	2.489220	0.1919725	47.00000	1301	8	3051	4	3461817
46	736.0797	1592.4217	1.026497	6.123324	0.4200532	2.322538	0.1642133	47.00000	1247	10	1800	2	2314388
47	771.6115	1771.5656	1.033937	5.779573	0.4329133	2.479315	0.1706529	47.00000	1268	10	1816	0	2465247
48	726.9584	1850.2125	1.030114	5.882778	0.4368927	2.529368	0.1637631	46.63833	1644	16	2358	0	2974716

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

Home Audio

week_year	list_price	MRP	Units	SLA	promotional_offer	Procurement_SLA	per_order	NPS	Premium_p	Middle_p	Mass_p	holiday_freq	gmv
26	2746.333	3249.667	1.000000	4.333333	0.1565662	4.666667	1.0000000	54.60000	0	0	3	0	8239
27	2338.035	3816.198	1.006658	6.517976	0.3329584	2.727031	0.2410120	54.60000	0	3	748	0	1773933
28	2164.772	3452.095	1.010471	7.631494	0.3312229	2.683447	0.2464760	54.60000	0	7	2476	0	5401829
29	2213.912	3634.609	1.008207	7.063499	0.3387710	2.688553	0.2228942	54.60000	0	8	2307	2	5151645
30	2146.182	3520.043	1.005533	7.202858	0.3314970	2.704472	0.2134624	54.60000	0	2	2167	0	4681706
31	2215.802	3728.333	1.007776	7.237947	0.3417192	2.603421	0.2363919	54.60840	0	1	642	0	1430799
35	2032.210	3704.846	1.008734	5.864629	0.3994524	2.697234	0.2154294	47.05255	0	4	1370	3	2817374
36	2135.183	3729.465	1.004924	5.411128	0.3677363	2.767110	0.2348597	46.90000	0	10	4052	0	8722150
37	1980.668	3529.194	1.013934	5.585745	0.3776464	2.745981	0.2025723	46.90000	0	18	3714	0	7472222
38	1954.240	3327.583	1.007052	6.420780	0.3661542	2.638928	0.2233192	46.90000	0	16	4238	0	8332486
39	1897.154	3173.260	1.009311	6.232296	0.3411350	2.532468	0.2234746	46.90000	0	6	4075	0	7794543
40	1939.658	3215.860	1.010204	6.285204	0.3439507	2.510204	0.2091837	44.40000	0	4	1956	0	3829763
41	2360.912	4412.898	1.009699	5.587291	0.4000720	2.500669	0.2227425	44.40000	1	3	2986	2	7121050
42	2384.808	4533.333	1.011123	6.198971	0.4146324	2.315211	0.2442992	44.40000	0	12	7180	3	17293939
43	2208.124	3838.624	1.011654	6.588346	0.3674964	2.519925	0.1917293	44.40000	0	0	2660	0	5924864
44	2083.021	3392.043	1.012320	6.349897	0.3404685	2.492813	0.1765914	45.78168	0	1	2434	0	5117172
45	2269.559	4738.259	1.010772	5.046679	0.4540607	2.534496	0.1856886	47.00000	0	4	3895	4	8925261
46	2069.580	3638.402	1.014392	6.450426	0.3545211	2.390725	0.1934968	47.00000	0	4	1872	2	3912077
47	2159.753	3675.570	1.006904	6.033988	0.3495621	2.465746	0.2002124	47.00000	0	3	1880	0	4084870
48	2068.733	3653.837	1.010793	6.144533	0.3752468	2.463163	0.1825434	46.60582	0	3	2128	0	4440784
49	2230.547	4756.762	1.007148	4.956755	0.4683486	2.893495	0.1669049	45.80000	0	3	2795	0	6270658
50	2183.965	4185.480	1.004228	5.674419	0.4143772	2.742495	0.1708245	45.80000	0	3	2362	0	5179731

Bring Ad Stock Information

```
Adstock <- read.csv("D:\\Mindtree- Personal Docs-13 Feb 2016\\TapChief\\Solutions - Ecommerce\\Final_adstock.csv")

GamingAccessory <- merge(GamingAccessory , Adstock,by.x = "week_year")
```

Next, we have to create moving average variables. We will use 2, 3, 4 point moving average. Variables in consideration will be “list price” and “promotional offer”. You can calculate this by whatever way you want. But at the end, you should have (LP_MA1, PO_MA1), (LP_MA2, PO_MA2), (LP_MA3, PO_MA3) – 2, 3 and 4 point Mas respectively for list price and promotional offer. You need to merge these variables as well.

```
x=dataset[,c("week_year","list_price","promotional_offer")]
```

After this, we need to calculate the incremental List Price lift by using -

```
dataset$inc_LP_MA1<-(dataset$list_price - dataset$LP_MA1)/dataset$LP_MA1

dataset$inc_LP_MA2<-(dataset$list_price - dataset$LP_MA2)/dataset$LP_MA2

dataset$inc_LP_MA3<-(dataset$list_price - dataset$LP_MA3)/dataset$LP_MA3
```

In the same way, you have to create the Promotional Offer lifts (Not shown here). You can drop these MA variables as you main concern are the lifts.

Finally we will also compute lag variables and we can use - library(DataCombine) to do this

1) Lag of List price by 1 week, 2 week and 3 week

2) Lag of Promotional offer by 1 week,2 week, 3 week

3) Lag of Promotional NPS by 1 week,2 week, 3 week

Note – This is not a solution material. This document is created to give you some detailed understanding on the approach and some scripts to help you write your own solution

Prepared By – Anirban Pramanik / Date – 31st Aug 2018

4) Lag of Promotional Holiday Frequency by 1 week,2 week, 3 week

```
data_dum <- slide(dataset, Var = "list_price",slideBy = -1)

data_dum <- slide(data_dum, Var = "list_price",slideBy = -2)

data_dum <- slide(data_dum, Var = "list_price", slideBy = -3)
```

Performing EDA

```
plot1 <- ggplot(dataset,aes(TV,gmv))+geom_point()+ geom_smooth(aes(method="lm"))+ggtitle(name) + labs(x = "TV AdStock", y = "GMV")

print(plot1)
```