MSCI 719: Operations Analytics

Assignment 7: End to End Analytics for an Online Retailer, Rue La La: Part II (Price Optimization)

Student: Advait Shah

Faculty: Engineering

Department: Management Sciences

Instructor: Prof. Hossein Abouee Mehrizi

Contents

1. Price Optimization

In Lecture 8, we used a regression tree for demand prediction and developed a model to determine the optimal price of new items so that the firm's revenue is maximized. Instead of a regression tree, use linear regression for demand prediction and answer the following questions.

We have translated Price optimization problem into following linear programming problem:

$$Max \sum_{i\in N}\sum_{j\in M} p_j \hat{D}_{ijk} x_{ij}$$

$$s.t$$

$$\sum_{j\in M} x_{ij} = 1 \qquad \forall i \in N \qquad (1)$$

$$\sum_{i\in N}\sum_{j\in M} p_j x_{ij} = k \qquad (2)$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in N, \forall j \in M \qquad (3)$$

p_j: The j^{th} possible price in set of all prices M

$\hat{D}_{ijk}$: Representing sales of the i^{th} style and j^{th} possible price when the sum of prices of competing styles is k.

x_{ij}: Equals to1 if style i is assigned price p_j and equals to 0 otherwise.

1.1. Compare the results of the regression tree and linear regression methods.

For this we need to divide dataset into train and test dataset, and we will see how our trained models are performing on test dataset with respect to mean absolute errors.

```
# train and test data set split in 80-20 ratio
prediction_data_train = prediction_data[1:(0.8*nrow(prediction_data)),]
prediction_data_test =
prediction_data[(0.8*nrow(prediction_data)+1):nrow(prediction_data),]
```

As you can see below, we have trained both models in R based, with predicted demands as outcome variable and remaining variables as predictor. Note that we had adjusted our demands of stockouts items by doing clustering and making proper adjustments in the sales data, and the same can be referred in my previous assignment report.

Regression Tree model:

```
regression_tree_1 = rpart(formula = true_demand~., data = prediction_data_
train, method = "anova")

mae(prediction_data_test$true_demand,predict(regression_tree_1,prediction_
data_test))

## [1] 873.6247
```

Now, while preparing Linear regression (LR) model, I checked for correlations between variables and found variables to be sufficiently independent for creating LR model.

Linear Regression model:

```
#install.packages("Metrics")
library(Metrics)

## Warning: package 'Metrics' was built under R version 4.1.3

l_regression_model_1 = lm(true_demand~., data = prediction_data_train)

mae(prediction_data_test$true_demand,predict.lm(l_regression_model_1,predi
ction_data_test))

## [1] 991.2431
```

So, we see that, we get MAE of 991.24 in Linear regression model; whereas, it is about 873.62 in regression tree when model is tested on test data set (which is remaining 20% of the given data set).

So, we can say that regression tree model is working well compared to LR model in this case. But, note that we could improve LR model also by doing feature engineering, as some of the variables included in the model are not significant, as their p-value are greater than 0.05.

1.2. Determine the optimal price for the given items using the data set discussed in Lecture and Tutorial 8. Did the optimal prices change?

Now, we will use entire data set to train both of these models (which will be used later to predict demands based on new data and their set prices):

```
regression_tree = rpart(formula = true_demand~., data = prediction_data, m
ethod = "anova")
#rpart.plot(regression_tree)

l_regression_model = lm(true_demand~., data = prediction_data)
#summary(l_regression_model)
```

Importing new data

```
Data_test = read.csv("new_data.csv")
Data_test
```

```
##   ï..Item. Beginning_of_Season Weekend Event_Length Morning Branded
## 1        A                   1       0            2       0       1
## 2        B                   1       0            2       0       0
## 3        C                   1       0            2       0       1
##   Color_Popularity ConcurrentEvents Number_Competing_Styles_in_Event
## 1             0.17                2                                3
## 2             0.65                2                                3
## 3             0.08                2                                3
##   Num_Branded_Events12 Brand_MSRP_Index Bottoms Tops Dresses
## 1                    2             0.87       0    1       0
## 2                    1             0.33       0    1       0
## 3                    1             1.18       0    1       0
```

Optimization:

Price Definition

```
Prices = c(25, 30, 35)
P = rep(Prices, nrow(Data_test))
P

## [1] 25 30 35 25 30 35 25 30 35
```

Preparing Variables

```
Data_test2 = Data_test[rep(seq_len(nrow(Data_test)), each=3), ]
Data_test2

##     ï..Item. Beginning_of_Season Weekend Event_Length Morning Branded
## 1          A                   1       0            2       0       1
## 1.1        A                   1       0            2       0       1
## 1.2        A                   1       0            2       0       1
## 2          B                   1       0            2       0       0
## 2.1        B                   1       0            2       0       0
## 2.2        B                   1       0            2       0       0
## 3          C                   1       0            2       0       1
## 3.1        C                   1       0            2       0       1
## 3.2        C                   1       0            2       0       1
##     Color_Popularity ConcurrentEvents Number_Competing_Styles_in_Event
## 1               0.17                2                                3
## 1.1             0.17                2                                3
## 1.2             0.17                2                                3
## 2               0.65                2                                3
## 2.1             0.65                2                                3
## 2.2             0.65                2                                3
## 3               0.08                2                                3
## 3.1             0.08                2                                3
## 3.2             0.08                2                                3
##     Num_Branded_Events12 Brand_MSRP_Index Bottoms Tops Dresses
## 1                      2             0.87       0    1       0
## 1.1                    2             0.87       0    1       0
## 1.2                    2             0.87       0    1       0
## 2                      1             0.33       0    1       0
## 2.1                    1             0.33       0    1       0
## 2.2                    1             0.33       0    1       0
```

```
## 3                           1             1.18        0    1       0
## 3.1                         1             1.18        0    1       0
## 3.2                         1             1.18        0    1       0
```

Possible Ks

```
possible_k = seq(length(Prices)*min(Prices), length(Prices)*max(Prices), b
y = 5)
possible_k

## [1]  75  80  85  90  95 100 105
```

Initialization

```
Demand_pred = vector(mode = "numeric")
Objectives = vector(mode = "numeric")
Solutions = matrix(nrow = length(possible_k), ncol = length(Prices)*nrow(D
ata_test))
```

Objective: Revenue Optimization:

1.2.1 Optimal Prices when Demands were predicted with **"Regression Tree"** model for each price and set of data:

```
#install.packages("lpSolve", dependencies = TRUE)
library(lpSolve)

## Warning: package 'lpSolve' was built under R version 4.1.3

for (n in 1:length(possible_k)) {
  for (i in 1:length(P)) {
    Data_test2$Price = P[i]
    Data_test2$Relative_Price_of_Competing_Styles = P[i]/(possible_k[n]/3)
    Demand_pred[i] = predict(regression_tree, Data_test2[i, ])
  }
  Obj_coeff = Demand_pred*P
  Cons_coeff = matrix(c(1,1,1,0,0,0,0,0,0,
                        0,0,0,1,1,1,0,0,0,
                        0,0,0,0,0,0,1,1,1,
                        P[1], P[2],P[3],P[4],P[5],P[6],P[7],P[8],P[9]), nr
ow = 4, byrow = TRUE)
  Dir = c("==",
          "==",
          "==",
          "==")
  Rhs = c(1,
          1,
          1,
          possible_k[n])
  Model = lp("max", Obj_coeff, Cons_coeff, Dir, Rhs, all.bin = TRUE)
  Objectives[n] = Model$objval
  Solutions[n,] = Model$solution
}
Demand_pred
```

```
## [1] 5492.087 5492.087 3343.701 6167.724 6167.724 3343.701 5492.087 5492
.087
## [9] 3343.701

Solutions

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    0    0    1    0    0    1    0    0
## [2,]    1    0    0    0    1    0    1    0    0
## [3,]    0    1    0    0    1    0    1    0    0
## [4,]    0    1    0    0    1    0    0    1    0
## [5,]    0    0    1    0    1    0    0    1    0
## [6,]    0    0    1    0    1    0    0    0    1
## [7,]    0    0    1    0    0    1    0    0    1

Objectives

## [1] 428797.5 459636.1 487096.5 514557.0 466823.9 419090.8 351088.6

Solutions[match(max(Objectives), Objectives), ]

## [1] 0 1 0 0 1 0 0 1 0
```

so, we get here optimal value of objective as 514557.0 and corresponding optimal prices for products A, B and C are 30,30, and 30 units.

1.2.2 Optimal Prices when Demands were predicted with **"Linear Regression"** model for each price and set of data:

```
for (n in 1:length(possible_k)) {
  for (i in 1:length(P)) {
    Data_test2$Price = P[i]
    Data_test2$Relative_Price_of_Competing_Styles = P[i]/(possible_k[n]/3)
    Demand_pred[i] = predict(l_regression_model, Data_test2[i, ])
  }
  Obj_coeff = Demand_pred*P
  Cons_coeff = matrix(c(1,1,1,0,0,0,0,0,0,
                        0,0,0,1,1,1,0,0,0,
                        0,0,0,0,0,0,1,1,1,
                        P[1], P[2],P[3],P[4],P[5],P[6],P[7],P[8],P[9]), nrow = 4,
byrow = TRUE)
  Dir = c("==",
          "==",
          "==",
          "==")
  Rhs = c(1,
          1,
          1,
          possible_k[n])
  Model = lp("max", Obj_coeff, Cons_coeff, Dir, Rhs, all.bin = TRUE)
  Objectives[n] = Model$objval
  Solutions[n,] = Model$solution
}
Demand_pred

## [1] 6985.925 5844.846 4703.768 6864.729 5723.651 4582.572 6767.013 5625.934
## [9] 4484.855
```

```
Solutions

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    0    0    1    0    0    1    0    0
## [2,]    0    1    0    1    0    0    1    0    0
## [3,]    0    1    0    0    1    0    1    0    0
## [4,]    0    1    0    0    1    0    0    1    0
## [5,]    0    0    1    0    1    0    0    1    0
## [6,]    0    0    1    0    0    1    0    1    0
## [7,]    0    0    1    0    0    1    0    0    1

Objectives

## [1] 496873.4 499504.8 502103.8 504691.9 497236.0 489653.0 481991.8

Solutions[match(max(Objectives), Objectives), ]

## [1] 0 1 0 0 1 0 0 1 0
```

So, we get here optimal value of objective as 504691.9 and corresponding optimal prices for the products A, B and C are 30,30, and 30 units. So, when we used Linear regression model instead of regression trees for demand prediction and subsequent optimization problem, our optimal prices of products remained unchanged.

1.3. Add the following assumptions and obtain the optimal price for new items (use regression tree for the demand prediction).

- Items C and B cannot be sold at $45.
- Item A cannot be sold at $25.

Now, adding some price related constraints: Items C and B cannot be sold at $35. Item A cannot be sold at $25.

Price Definition

```
Prices = c(25, 30, 35)
# possible prices of A, B and C after applying constraints (each product h
as two possible prices)
P = c(30, 35, 25, 30, 25, 30)
P

## [1] 30 35 25 30 25 30
```

Preparing Variables

```
Data_test2 = Data_test[rep(seq_len(nrow(Data_test)), each=2), ]
Data_test2

##     ï..Item. Beginning_of_Season Weekend Event_Length Morning Branded
## 1          A                   1       0            2       0       1
## 1.1        A                   1       0            2       0       1
## 2          B                   1       0            2       0       0
## 2.1        B                   1       0            2       0       0
## 3          C                   1       0            2       0       1
## 3.1        C                   1       0            2       0       1
##     Color_Popularity ConcurrentEvents Number_Competing_Styles_in_Event
## 1               0.17                2                                3
## 1.1             0.17                2                                3
```

```
## 2              0.65              2                               3
## 2.1            0.65              2                               3
## 3              0.08              2                               3
## 3.1            0.08              2                               3
##      Num_Branded_Events12 Brand_MSRP_Index Bottoms Tops Dresses
## 1                       2             0.87       0    1       0
## 1.1                     2             0.87       0    1       0
## 2                       1             0.33       0    1       0
## 2.1                     1             0.33       0    1       0
## 3                       1             1.18       0    1       0
## 3.1                     1             1.18       0    1       0
```

Possible Ks

```
possible_k = seq(length(Prices)*min(Prices), length(Prices)*max(Prices), b
y = 5)[1:5]
#sliced because 100 and 105 not possible due to contraints
possible_k

## [1] 75 80 85 90 95
```

Initialization

```
Demand_pred = vector(mode = "numeric")
Objectives = vector(mode = "numeric")
Solutions = matrix(nrow = length(possible_k), ncol = 2*nrow(Data_test))
```

Running LP optimization:

```
for (n in 1:length(possible_k)) {
  for (i in 1:length(P)) {
    Data_test2$Price = P[i]
    Data_test2$Relative_Price_of_Competing_Styles = P[i]/(possible_k[n]/3)
    Demand_pred[i] = predict(regression_tree, Data_test2[i, ])
  }
  Obj_coeff = Demand_pred*P
  Cons_coeff = matrix(c(1,1,0,0,0,0,
                        0,0,1,1,0,0,
                        0,0,0,0,1,1,
                        P[1], P[2],P[3],P[4],P[5],P[6]), nrow = 4, byrow =
TRUE)
  Dir = c("==",
          "==",
          "==",
          "==")
  Rhs = c(1,
          1,
          1,
          possible_k[n])
  Model = lp("max", Obj_coeff, Cons_coeff, Dir, Rhs, all.bin = TRUE)
  Objectives[n] = Model$objval
  Solutions[n,] = Model$solution
}
Demand_pred

## [1] 5492.087 3343.701 6167.724 6167.724 5492.087 5492.087
```

```
Solutions

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    0    0    0    0    0
## [2,]    1    0    1    0    1    0
## [3,]    1    0    0    1    1    0
## [4,]    1    0    0    1    0    1
## [5,]    0    1    0    1    0    1

Objectives

## [1]      0.0 456257.9 487096.5 514557.0 466823.9

Solutions[match(max(Objectives), Objectives), ]

## [1] 1 0 0 1 0 1
```

As our prices vector was, P = c(30, 35, 25, 30, 25, 30), from above results we can say that with these constraints cases also our optimal prices solution remains same as it was received without such constraints in part 1.2, i.e. optimal prices for product A, B and C would be 30, 30, and 30 respectively.

References:

[1] H A Mehrizi, eBook: MSCI 719 Winter 2023 Cases Multiple (ID: 9723713) Accessed: Jan. 22, 2023. [Online].

Available: https://www.campusebookstore.com/integration/AccessCodes/default.aspx?permalinkId=ee044bf2-fe82-4db0-ad22-088e81954eef&frame=YES&t=permalink&sid=4u2faw45zyslbp45bbqlpc55