DataMining1

Harika

2/27/2020

#Libraries used in this task

library(ISLR)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(glmnet)

## Warning: package 'glmnet' was built under R version 3.6.2

## Loading required package: Matrix

## Loaded glmnet 3.0-2

#SafeBabies is the filtered dataset from Carseats with only three variables like Sales,Price,ShelveLoc

SafeBabies <- Carseats %>% select("Sales", "Price", "ShelveLoc")

#To know data type of each variable

str(SafeBabies)

## 'data.frame': 400 obs. of 3 variables:  
## $ Sales : num 9.5 11.22 10.06 7.4 4.15 ...  
## $ Price : num 120 83 80 97 128 72 108 120 124 124 ...  
## $ ShelveLoc: Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...

#Filter the data according to ShelveLoc location catagory good and bad.

#Good ShelveLoc data  
good <- filter(SafeBabies,ShelveLoc == "Good")  
#Bad ShelveLoc data  
bad <- filter(SafeBabies,ShelveLoc=="Bad")

#To maximize the profit we based on ShelveLoc following mathematical analysis is pre-required Profit = Sales \* (Price - ProductionCost) Profit = S*(P-C)————-(1) S = -b1*P+b0 By doing regression we can find values of b0,b1 So equation(1) becomes Profit = (-b1*P+b0)*(P-C) Profit = -b1*P^2+b1*P*C+b0*P-b0\*C ——–(2) Now Profit is quadratioc equation to find value of P(Price) by doing derivative of eq(2) to 0.

#Regression for finding b0,b1 values for good and bad ShelvLOc

good\_model <- lm(Sales ~ Price,data=good)  
summary(good\_model)

##   
## Call:  
## lm(formula = Sales ~ Price, data = good)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.721 -1.351 -0.098 1.483 4.353   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17.968864 0.988008 18.187 < 2e-16 \*\*\*  
## Price -0.065785 0.008199 -8.023 5.85e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.888 on 83 degrees of freedom  
## Multiple R-squared: 0.4368, Adjusted R-squared: 0.43   
## F-statistic: 64.37 on 1 and 83 DF, p-value: 5.848e-12

bad\_model <- lm(Sales ~ Price,data=bad)  
summary(bad\_model)

##   
## Call:  
## lm(formula = Sales ~ Price, data = bad)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4622 -1.0617 -0.2014 1.2050 4.6412   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.832984 0.990317 11.949 < 2e-16 \*\*\*  
## Price -0.055220 0.008486 -6.507 3.7e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.967 on 94 degrees of freedom  
## Multiple R-squared: 0.3105, Adjusted R-squared: 0.3032   
## F-statistic: 42.34 on 1 and 94 DF, p-value: 3.702e-09

#Finding Derivative of equation(2) by substituting b1,b0,C=55 values

Profit\_good <- expression(-0.065785\*P^2 + 17.968864\*P\*55 + (17.968864)\*P +55\*0.065785)   
D1 <- D(Profit\_good,'P')  
P\_good <- (-0.065785 \*55 - 17.968864)/(2 \* -0.065785)  
P\_good

## [1] 164.0727

Profit\_bad <- expression(-0.055220\*P^2 + 11.832984\*P\*55 + (11.832984)\*P +55\*0.055220)   
D2 <- D(Profit\_bad,'P')  
P\_bad <- (-0.055220 \*55 - 11.832984)/(2 \* -0.055220)  
P\_bad

## [1] 134.644

# OPtimal Price for good shelvelocation is 164.0727

#OPtimal Price for bad shelvelocation is 134.644

# When cost “C” is range from 40 to 85

c <- 40:85  
Price\_good <- numeric(length=length(c))  
for (i in seq\_along(c)){  
 Price\_good[i] <- (-0.065785 \*c[i] - 17.968864)/(2 \* -0.065785)  
 optimal\_good <- cbind.data.frame(Varying\_cost = c(40:85),Price\_good)  
}  
Price\_good

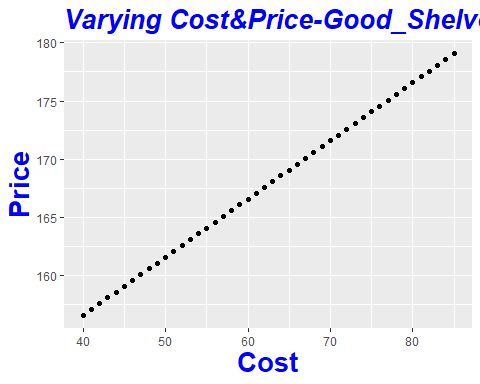
## [1] 156.5727 157.0727 157.5727 158.0727 158.5727 159.0727 159.5727  
## [8] 160.0727 160.5727 161.0727 161.5727 162.0727 162.5727 163.0727  
## [15] 163.5727 164.0727 164.5727 165.0727 165.5727 166.0727 166.5727  
## [22] 167.0727 167.5727 168.0727 168.5727 169.0727 169.5727 170.0727  
## [29] 170.5727 171.0727 171.5727 172.0727 172.5727 173.0727 173.5727  
## [36] 174.0727 174.5727 175.0727 175.5727 176.0727 176.5727 177.0727  
## [43] 177.5727 178.0727 178.5727 179.0727

c <- 40:85  
Price\_bad <- numeric(length=length(c))  
for (i in seq\_along(c)){  
 Price\_bad[i] <- (-0.055220 \*c[i] - 11.832984)/(2 \* -0.055220)  
 optimal\_bad <- cbind.data.frame(Varying\_cost = c(40:85),Price\_bad)  
}  
Price\_bad

## [1] 127.144 127.644 128.144 128.644 129.144 129.644 130.144 130.644  
## [9] 131.144 131.644 132.144 132.644 133.144 133.644 134.144 134.644  
## [17] 135.144 135.644 136.144 136.644 137.144 137.644 138.144 138.644  
## [25] 139.144 139.644 140.144 140.644 141.144 141.644 142.144 142.644  
## [33] 143.144 143.644 144.144 144.644 145.144 145.644 146.144 146.644  
## [41] 147.144 147.644 148.144 148.644 149.144 149.644

# Plots between varying cost and respective price for good and bad shelveloc

ggplot(optimal\_good,aes(x=Varying\_cost,y=Price\_good))+geom\_point()+ ggtitle("Varying Cost&Price-Good\_ShelveLoc") +xlab("Cost") + ylab("Price") + theme(plot.title = element\_text(color="blue", size=20, face="bold.italic"),  
axis.title.x = element\_text(color="blue", size=20, face="bold"),  
axis.title.y = element\_text(color="blue", size=20, face="bold")  
)



ggplot(optimal\_bad,aes(x=Varying\_cost,y=Price\_bad))+geom\_point()+ ggtitle("Varying Cost&Price-Bad\_ShelveLoc") +xlab("Cost") + ylab("Price") + theme(plot.title = element\_text(color="brown", size=20, face="bold.italic"),  
axis.title.x = element\_text(color="brown", size=20, face="bold"),  
axis.title.y = element\_text(color="brown", size=20, face="bold")  
)

