Untitled

2023-04-17

`{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)`

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

`{r cars} summary(cars)`

Including Plots

You can also embed plots, for example:

`{r pressure, echo=FALSE} plot(pressure)`

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

data <- read.csv("/Users/divyarobin/Desktop/DivyaSimplilearn/4) PGP with R/Project/Walmart_Store_sales.csv") View(data)

library("dplyr") #Calling dplyr function for data manipulation library("ggplot2") # for data visualisation library("scales") #for change of scales in data visualisation library("zoo") library("tidyverse") library("tidyr") library("lubridate") library(car) #Companion to Applied Regression for Regression Visualisations require(stats) library(corrplot) library(caTools) library(MLmetrics) library("repr")

head(data) dim(data) str(data) summary(data) class(data)

#Checking NULL values colSums(is.na(data)) #Observed no NULL values

Basic Statistics tasks

1) Which store has maximum sales

res1 <- aggregate(data$Weekly_Sales, list ¿$Store), sum) View(res1) store_with_max_sales <- which.max(res1$x) View(store_with_max_sales)

1) Result : Store Number 20, has maximum Weekly sales of 301397792.

2) Which store has maximum standard deviation i.e., the sales vary a lot. Also, find out the coefficient of mean to standard deviation

res2 <- aggregate(data$Weekly_Sales,list$¿Store), sd) View(res2) store_with_max_sd <- which.max(res2$x) View(store_with_max_sd)

res3 <- aggregate(data$Weekly_Sales,list$¿Store), mean) View(res3)

coeff_mean_to_sd <- (res2[14,2] / res3[14,2]) * 100 coeff_mean_to_sd

coeff_mean_to_sd_max <- function(a,b){ output <- (a/b)*100 return(output) }

res4 <- coeff_mean_to_sd_max(res2x,res3x) max(res4) which.max(res4)

2) Result : Store Number 14, has maximum Standard Deviation which is, 317569.95,

with coefficient of Mean to Standard Deviation = 15.71367.

Also, Store Number 35, has maximum coefficient of mean to standard deviation, which is 22.96811.

3) Which store/s has good quarterly growth rate in Q3’2012

data2<-data data2$month_Year=substr$¿Date, 4, 10) View(data2)

Q3_2012 <- filter(data2, data2$month_Year == "07-2012" | data2$month_Year == “08-2012” | data2$month_Year == "09-2012") Q2_2012 <- filter(data2, data2$month_Year == “04-2012” | data2$month_Year == "05-2012" | data2$month_Year == “06-2012”)

Q3_2012_Sales <- aggregate(Q3_2012$Weekly_Sales,list$¿Store), sum) colnames(Q3_2012_Sales)[1] <- “Store” colnames(Q3_2012_Sales)[2] <- “Q3_2012_Sales_by_Store” View(Q3_2012_Sales)

Q2_2012_Sales <- aggregate(Q2_2012$Weekly_Sales,list$¿Store), sum) colnames(Q2_2012_Sales)[1] <- “Store” colnames(Q2_2012_Sales)[2] <- “Q2_2012_Sales_by_Store” View(Q2_2012_Sales)

Q3_2012_Growthrate <- merge (Q2_2012_Sales , Q3_2012_Sales , by = ‘Store’) # Merging
View(Q3_2012_Growthrate)

Q3_2012_Growthrate$Growth_Rate <- ¿$Q3_2012_Sales_by_Store - Q3_2012_Growthrate
$Q2_{2012_S} ales_b y_S tore$ ¿$*100$¿$/Q3_{2012_G} rowthrate$Q2_2012_Sales_by_Store
View(Q3_2012_Growthrate)

positive_growthrate <- filter(Q3_2012_Growthrate, Growth_Rate > 0)
positive_growthrate<-arrange(positive_growthrate, desc(Growth_Rate))
View(positive_growthrate)

3) Result : “The positive growth rate Stores are 7 16 35 26 39 41 44 24 40 23”

“Store 7 has highest growth rate & it is 13.330776030738”

4) Some holidays have a negative impact on sales. Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together.

Holiday_date <- c(“12-02-2010”, “11-02-2011”, “10-02-2012”, “08-02-2013”,“10-09-2010”, “09-09-2011”, “07-09-2012”, “06-09-2013”,“26-11-2010”, “25-11-2011”, “23-11-2012”, “29- 11-2013”,“31-12-2010”, “30-12-2011”, “28-12-2012”, “27-12-2013”) Events <- c(rep(“Super Bowl”, 4), rep(“Labour Day”, 4),rep(“Thanksgiving”, 4), rep(“Christmas”, 4))
Holidays_Data <- data.frame(Events,Holiday_date)

data3<-merge(data,Holidays_Data, by.x= “Date”, by.y=“Holiday_date”, all.x = TRUE) data3
$Events=as.character$¿Events) data3$Events$¿Events)]= “No_Holiday” head(data3)

Holiday_Sales<-aggregate(data3$Weekly_Sales,list$¿Events), mean)
colnames(Holiday_Sales) <- c(“Events”, “Mean_Sales_by_Event”)
Holiday_Sales$Positive_Sales_Impact <- Holiday_Sales[,2] >= Holiday_Sales[3,2]
View(Holiday_Sales)

4) Result : Super Bowl, Thanksgiving and Labour day have sales higher than the mean sales of a Non Holiday and creating positive impact on sales.

Christmas Event has negative impact on Sales.

5) Provide a monthly and semester view of sales in units and give insights.

Monthly View

x <- as.factor(data2$Date) y <- strptime(x,format="%d-%m-%Y")

data2$Mon_Year<-as.Date(y,format="%Y-%m-%d") data2$Mon_Year = as.yearmon(data2$Mon_Year)

Month_Year_Sales<-summarise(group_by(data2,Mon_Year),sum(Weekly_Sales)) colnames(Month_Year_Sales)[2] <- "Sales_by_Month" Month_Year_Sales<- as.data.frame(Month_Year_Sales)

Month_Year_Sales$Mon_Y ear <- as.character$¿Mon_Year) Month_Year_Sales $Mon_Y ear <- factor$¿Mon_Year, levels=Month_Year_Sales$Mon_Year)

p <- ggplot(data=Month_Year_Sales, aes(x=Mon_Year, y=Sales_by_Month, group=1)) + geom_line(color="red")+ geom_point()+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+ scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6))+ ggtitle('Monthly Sales - 2010 to 2012')+ theme(plot.title = element_text(hjust = 0.5))+ xlab("Month") + ylab("Total Sales in a Month") p

Semester View

data2$Date <- dmy$¿Date) data2$sem <- semester$¿Date, with_year=TRUE)

sem_df <- aggregate(Weekly_Sales~sem,data=data2, sum) sem_df$sem_y ear <- paste$¿ sem,1,4),'-S',substr(sem_df$sem,6,6),sep = '')

q <- ggplot(data=sem_df, aes(x=sem_year, y=Weekly_Sales, group=1)) + geom_line(color="green")+ geom_point()+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+ scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6))+ ggtitle('Semester Sales - 2010 to 2012')+ theme(plot.title = element_text(hjust = 0.5))+ xlab("Semester") + ylab("Total Sales in a Semester") q

5) Result : From the Monthly sales plot, it is evident that the sales are higher during the month of December and lowest during January.

From the Semester sales plot, it is evident that the sales during Second Semester of 2010 and 2011 are higher and Second Semester of 2012 is the lowest.

Also, the sales during the First Semester of 2011, has seen a decrease.

#Statistical Model #For Store 1 – Build prediction models to forecast demand #Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010 (starting from the earliest date in order). Hypothesize if CPI, unemployment, and fuel price have any impact on sales. #Change dates into days by creating new variable. #Select the model which gives best accuracy.

data_lm <- data View(data_lm)

#missing values, mean, length, min, sd quantile, percentile mystats <- function(x){ nmiss <- sum(is.na(x)) a <- x[!is.na(x)] m <- mean(a) n <- length(a) s <- sd(a) min <- min(a) p1 <- quantile(a, 0.01) #this is the 1st percentile p5 <- quantile(a, 0.05) p10 <- quantile(a, 0.10) q1 <- quantile(a, 0.25) q2 <- quantile(a, 0.5) q3 <- quantile(a, 0.75) p90 <- quantile(a, 0.90) p95 <- quantile(a, 0.95) p99 <- quantile(a, 0.99) max <- max(a)

UC <- m + 3*s LC <- m - 3*s

outlier_flag <- max > UC | min < LC

return(c(nmiss = nmiss, s= s,n=n, mean=m,min = min, p1=p1,p5=p5,p10=p10,q1=q1, q2=q2,q3=q3,p90=p90,p95=p95,p99=p99,max=max, Upper_region = UC, lower_region = LC, outlier = outlier_flag)) }

num_ind <- sapply(data_lm, is.numeric) num_ind

num_col <- data_lm[num_ind] num_col

test <- apply(num_col,2,mystats) View(test)

diag <- t(data.frame(test)) View(diag)

Hypothesize if CPI, unemployment, and fuel price have any impact on sales.

1) CPI vs Weekly Sales

Ho -> CPI does not influence Weekly Sales

Ha -> CPI affects Weekly Sales

a <- chisq.test(data_lm$CPI, data_l m$Weekly_Sales) a

if (a[3] < 0.05) { print("Reject Null Hypothesis") }else{ print(" Accept Null Hypothesis") }

1) Result : Accept Null Hypothesis. Hence, CPI does not influence Weekly Sales.

2) Unemployment vs Weekly Sales

Ho -> Unemployment does not influence Weekly Sales

Ha -> Unemployment affects Weekly Sales

a <- chisq.test(data_lm$Unemployment, data_l m$Weekly_Sales) a

if (a[3] < 0.05) { print("Reject Null Hypothesis") }else{ print(" Accept Null Hypothesis") }

2) Result : Accept Null Hypothesis. Hence, Unemployment does not influence Weekly Sales.

3) Fuel Price vs Weekly Sales

Ho -> Fuel Price does not influence Weekly Sales

Ha -> Fuel Price affects Weekly Sales

a <- chisq.test(data_lm$Fuel_{P}rice, data_{l}m$Weekly_Sales) a

if (a[3] < 0.05) { print("Reject Null Hypothesis") }else{ print(" Accept Null Hypothesis") }

3) Result : Accept Null Hypothesis. Hence, Fuel Price does not influence Weekly Sales.

Changing dates into days.

data_day <- data_lm View(data_day) data_day$Date <- as.Date¿$Date, "%d-%m-%Y") str(data_day$Date¿data_{d}ay$Day <- weekdays(data_day$Date) View(data_day)

Creating a dataframe with required columns

data4 <- data

#selecting only first store as prediction Required only for first Store data4<- dplyr::filter(data4, Store ==1)

#changing date column in dataframe to date format & arranging in ascending order as per dates data4$Date <- lubridate::dmy¿$Date) data4 <- dplyr::arrange(data4,Date)

#Creating a week number,month,quarter column in dataframe data4 $Week_{N}umber <- seq¿$Date)))

#adding quarter & month columns data4$month <- lubridate::month¿$Date) data4 $quarter <- lubridate::quarter¿$Date)

##Creating a event type dataframe##

creating Holiday_date vector

Holiday_date <- c("12-02-2010", "11-02-2011", "10-02-2012", "08-02-2013","10-09-2010", "09-09-2011", "07-09-2012", "06-09-2013","26-11-2010", "25-11-2011", "23-11-2012", "29-11-2013","31-12-2010", "30-12-2011", "28-12-2012", "27-12-2013")

#assigning date format to Holiday_date vector Holiday_date <- lubridate::dmy(Holiday_date)

#Creating Events vector Events <-c(rep("Super Bowl", 4), rep("Labour Day", 4),rep("Thanksgiving", 4), rep("Christmas", 4))

#Creating dataframe with Events and date Holidays_Data <- data.frame(Events,Holiday_date)

#merging both dataframes data4<-merge(data4,Holidays_Data, by.x= "Date", by.y="Holiday_date", all.x = TRUE)

#Replacing null values in Event with No_Holiday data4$Events=as.character$¿Events) data4$Events$¿Events)]= "No_Holiday"

Removing outliers using Box Plots

par(mfrow=c(1,1))

#Creating a dataframe for outlier treatment data5 <- data4

#As we are predicting sales, Thought of removing outliers in Sales based on Various parameters #Temperature Outlier treatment – found 5 outlier and removed them boxplot(data5$Weekly_Sales cut$¿Temperature, pretty(data5$Temperature)), main="Temperature vs Weekly Sales", xlab ="Temperature", ylab="Weekly Sales", cex.axis=0.5, col="Steel Blue") outliers_temp <- boxplot(data5$Weekly_Sales ~ cut(data5 $Temperature, pretty$¿Temperature)), main="Temperature vs Weekly Sales", cex.axis=0.5,plot=FALSE)$out data5<-data5$¿Weekly_Sales %in% outliers_temp),]

#CPI Outlier treatment-found 1 outlier and removed them boxplot(data5 $Weekly_Sales cut$¿CPI, pretty(data5$CPI)), main="CPI vs Weekly Sales",xlab ="CPI", ylab="Weekly Sales", cex.axis=0.5,col="Steel Blue") outliers_CPI <- boxplot(data5$Weekly_Sales ~ cut(data5$CPI, pretty$¿CPI)), main="CPI vs Weekly Sales", cex.axis=0.5,plot=FALSE)$out data5<-data5$¿Weekly_Sales %in% outliers_CPI),]

#Unemployment outlier treatment–found 3 outlier and removed them boxplot(data5 $Weekly_Sales cut$¿Unemployment, pretty(data5$Unemployment)), main="Unemployment vs Weekly Sales",xlab ="Unemployment", ylab="Weekly Sales", cex.axis=0.5,col="Steel Blue") outliers_Unemployment <- boxplot(data5$Weekly_Sales ~ cut(data5$Unemployment, pretty$¿Unemployment)), main="Unemployment vs Weekly Sales", cex.axis=0.5,plot=FALSE)$out data5<-data5$¿Weekly_Sales %in% outliers_Unemployment),]

#fuel price outlier treatment – found 2 outliers and removed boxplot(data5 $Weekly_Salescut¿$Fuel_Price, pretty(data5$Fuel_Price)), main="Fuel_Price vs Weekly Sales", xlab ="Fuel Price", ylab="Weekly Sales", cex.axis=0.5,col="Steel Blue") outliers_fuel_price <- boxplot(data5$Weekly_Sales ~ cut(data5$Fuel_Price, pretty¿$ Fuel_Price)), main="Fuel_Price vs Weekly Sales", cex.axis=0.5,plot=FALSE) $outdata5<-data5¿$Weekly_Sales %in% outliers_fuel_price),]

#Outlier treatment for Holiday Flag - No outliers found boxplot(data5 $Weekly_Salesdata5$Holiday_Flag, main = 'Weekly Sales - Holiday_Flag',xlab ="Holiday Flag", ylab="Weekly Sales",col="Steel Blue")

#outlier treatment for month - 4 outliers found and removed boxplot(data5 $Weekly_Salesdata5$month, main = 'Weekly Sales - month', xlab ="Month", ylab="Weekly Sales", col="Steel Blue") outliers_month <- boxplot(data5$Weekly_Salesdata5$month, main = 'Weekly Sales - month',plot=FALSE)$outdata5<-data5¿$Weekly_Sales %in% outliers_month),]

#outlier treatment for quarter - 2 outliers found and removed outliers_quarter <- boxplot(data5$Weekly_Salesdata5$quarter, main = 'Weekly Sales - quarter',xlab ="Quarters", ylab="Weekly Sales", col="Steel Blue")$outdata5<-data5¿$Weekly_Sales %in% outliers_quarter),]

#Removing unnecessary columns and changing structure of Events data5 $Date<-NULLdata5$Store <- NULL data5$Events<-as.factor¿$Events) str(data5)

data5$Holiday_Flag<-as.numeric¿$Holiday_Flag) data5 $Week_Number<-as.numeric¿$Week_Number) data5$quarter<-as.numeric¿$quarter)

Finding Multi-collinearity between X variables:

#correlation matrix and corr plot corr = cor(data5[, c(1:9)]) View(corr) corrplot(corr, method = "color", cl.pos = 'n', rect.col = "black", tl.col = "indianred4", addCoef.col = "black", number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c("blue","white","red"))(100))

Observation: Very low correlation between Temperature and Weekly Sales. Hence can be omitted.

#Creating Dummy Variables for categorical variables as continuous variables

Events <- as.factor(data5$Events) dummy_Events <- data.frame(model.matrix(~Events))[,-1]

quarter <- as.factor(data5$quarter) dummy_quarter <- data.frame(model.matrix(~quarter))[,-1] month <- as.factor(data5$month) dummy_month <- data.frame(model.matrix(~month))[,-1]

data5 <- cbind(data5,dummy_Events,dummy_quarter,dummy_month)

View(data5)

corr = cor(data5[, c(1,2,3,4,5,6,7,8,9,11,12,13)]) View(corr) corrplot(corr, method = “color”, cl.pos = ‘n’, rect.col = “black”, tl.col = “indianred4”, addCoef.col = “black”, number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c(“blue”,“white”,“red”))(100))

OBSERVATION: Very low correlation between Temperature and Holiday Flag with Sales. Hence dropping Temperature and Holiday Flag.

Also, CPI has higher co-linearity with Week Number and Fuel Price, hence dropping CPI column.

Building model with the listed parameters:

Weekly Sales, Fuel Price, Unemployment, Week Number, Dummy_Event and Month.

final_dataset <- data5[, c(1,4,6,7,11:12, 17:27)]

Splitting data into Train and Test:

train_ind = sample(1:nrow(final_dataset),size = floor(0.80*nrow(final_dataset)))

final_dataset[train_ind,]

Training = final_dataset[train_ind,] #here this is my training data Testing = final_dataset[-train_ind,] #here this is my testing data

View(Testing) dim(Training) dim(Testing)

#Lets build the model for training the data #lm() is for linear model - Linear Regression(Y ~ X(trainingdata)) names(final_dataset)

model <- lm(Weekly_Sales ~ Fuel_Price + Unemployment + Week_Number + EventsLabour.Day + EventsNo_Holiday + month2 + month3 + month4 + month5 + month6

+ month7 + month8 + month9+ month10 + month11 + month12, data = Training)
summary(model)

#Now test and validate the model y_pred_test <- predict(model,newdata = Testing)
y_pred_test

testing = cbind(Testing,Prediction_Y_cap = y_pred_test) View(testing)

testing$error <- testing$Weekly_Sales - testing$Prediction_{Y_c}aphead¿$error)
View(testing)

#RMSE - Root Mean Square Error rmse <- sqrt(mean(testing$error^2)) rmse #The lesser the error the better the model max(testing$error) min(testing$error)

print(paste(“RMSE :”, rmse, ” Adjusted R-squared value: 0.3438”))