anomaly

Whiterose

2022-06-13

Exploratory data analysis

Define the question

You are a Data analyst at Carrefour Kenya and are currently undertaking a project that will inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

defining the metric for success

explaining the context

Carre-four is a French multinational retail corporation headquartered in Massy, France. The eighth-largest retailer in the world by revenue, it operates a chain of hypermarkets, groceries stores and convenience stores, which as of January 2021, comprises its 12,225 stores in over 30 countries. Kenya been one of them a statistical analysis is needed to improve sales in this country.

experimental design

1.Problem Definition 2.Data Sourcing 3.Check the Data 4.Perform Data Cleaning 5.Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate) 6.Implement the Solution 7.Challenge the Solution 8.Follow up Questions

data source validation

loading packages

```
library(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(stats)
library(readr)
library(rmarkdown)
library(tidyr)
library(tibble)
library(caret)
```

```
## Loading required package: lattice
```

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

loading dataset

```
df <- read.csv("Supermarket_Sales_Dataset II.csv", row.names=NULL)
```

viewing dataset

```
view(df)
```

verifying object's class

```
class(df)
```

```
## [1] "data.frame"
```

Data cleaning

missing values

```
colSums(is.na(df))
```

```
##            shrimp           almonds           avocado    vegetables.mix 
##                 0                 0                 0                 0 
##      green.grapes  whole.weat.flour              yams    cottage.cheese 
##                 0                 0                 0                 0 
##      energy.drink      tomato.juice    low.fat.yogurt         green.tea 
##                 0                 0                 0                 0 
##             honey             salad     mineral.water            salmon 
##                 0                 0                 0                 0 
## antioxydant.juice   frozen.smoothie           spinach         olive.oil 
##                 0                 0                 0              7500
```

dealing with missing values

```
na.omit(df)
```

```
##  [1] shrimp            almonds           avocado           vegetables.mix   
##  [5] green.grapes      whole.weat.flour  yams              cottage.cheese   
##  [9] energy.drink      tomato.juice      low.fat.yogurt    green.tea        
## [13] honey             salad             mineral.water     salmon           
## [17] antioxydant.juice frozen.smoothie   spinach           olive.oil        
## <0 rows> (or 0-length row.names)
```

```
nrow(df) # test how many rows
```

```
## [1] 7500
```

```
Not0 <- which(df$LATITUDE == 0) #output which rows = 0
data <- df[-Not0,] # new data = old data with rows != 0
nrow(df) # test how many rows
```

```
## [1] 7500
```

```
write.csv(df, file = "sales.csv") #output new file
```

aprior algorithm

```
# Installing Packages
#install.packages("arules")
#install.packages("arulesViz")

# Loading package
library(arules)
library(arulesViz)
library(RColorBrewer)
```

```
# Fitting model
# Training Apriori on the dataset
'set.seed = 100 # Setting seed
associa_rules = apriori(data = df,
                        parameter = list(support = 0.004,
                                         confidence = 0.2))'
```

```
## [1] "set.seed = 100 # Setting seed\nassocia_rules = apriori(data = df, \n                        para
```

plotting

```
# Plot
#itemFrequencyPlot(df, topN = 10)
```

Visualising the results

```
#inspect(sort(associa_rules, by = 'lift')[1:10])
'plot(associa_rules, method = "graph",
     measure = "confidence", shading = "lift")'
```

```
## [1] "plot(associa_rules, method = \"graph\", \n     measure = \"confidence\", shading = \"lift\")"
```

Anomaly detection

loading dataset

```
df1 <- read_csv("Supermarket_Sales_Forecasting - Sales.csv")
```

```
## Rows: 1000 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (1): Sales
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(df1)
```

```
## Rows: 1,000
## Columns: 2
## $ Date  <chr> "1/5/2019", "3/8/2019", "3/3/2019", "1/27/2019", "2/8/2019", "3/~
## $ Sales <dbl> 548.9715, 80.2200, 340.5255, 489.0480, 634.3785, 627.6165, 433.6~
```

Installing anomalize package

```
#install.packages("anomalize")
```

Load tidyverse and anomalize

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v purrr   0.3.4     v forcats 0.5.1
## v stringr 1.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x Matrix::expand() masks tidyr::expand()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## x Matrix::pack()   masks tidyr::pack()
## x arules::recode() masks dplyr::recode()
## x Matrix::unpack() masks tidyr::unpack()
```

```
library(anomalize)
```

```
## == Use anomalize to improve your Forecasts by 50%! =============================
## Business Science offers a 1-hour course - Lab #18: Time Series Anomaly Detection!
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```
#Install the devtools package then github packages
#install.packages("devtools")
#install.packages("Rcpp")
library(devtools)
```

```
## Loading required package: usethis
```

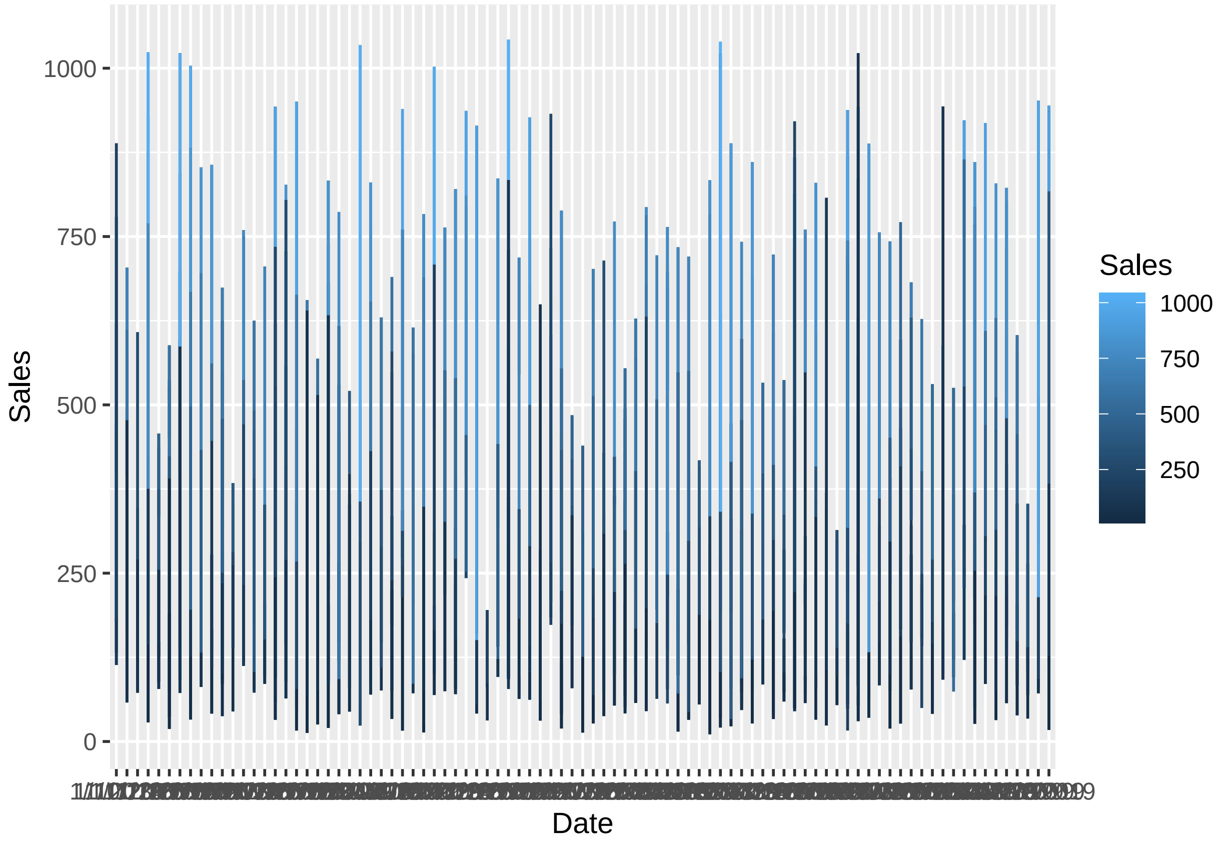
```
#install_github("petermeissner/wikipediatrend")
#install_github("twitter/AnomalyDetection")

#Loading the libraries
library(Rcpp)
library(wikipediatrend)
```

```
##
##   [wikipediatrend]
##
##   Note:
##
##     - Data before 2016-01-01
##       * is provided by petermeissner.de and
##       * was prepared in a project commissioned by the Hertie School of Governance (Prof. Dr. Simon Mu
##       * and supported by the Daimler and Benz Foundation.
##
##     - Data from 2016-01-01 onwards
##       * is provided by the Wikipedia Foundation
##       * via its pageviews package and API.
##
```

```
library(AnomalyDetection)
```

```
#Plotting data
library(ggplot2)
ggplot(df1, aes(x=Date, y=Sales, color=Sales)) + geom_line()
```



There some huge pikes at different levels. however our data seems not to have a lot of anomalies.

```
#Apply anomaly detection and plot the results
#anomalies = AnomalyDetectionTs(df1[2], direction="both", plot=TRUE, period = 24)
#anomalies$plot
```