Anomaly_Detection

Group B

20/02/2021

Abstract

Anomaly detection or Outlier detection identifies data points, events or observations that deviate from dataset's normal behavior. Anomalous data indicate critical incidents or potential opportunities. In order to take advantage of opportunities or fix costly problems anomaly detection has to be done in real time. Unsupervised machine learning models can be used to automate anomaly detection. Unsupervised anomaly detection algorithms scores data based on intrinsic properties of the dataset. Distances and densities are used to give an estimation what is normal and what is an outlier. Anomaly detection monitor is a tool developed for an online retailer to check product quality issues like profit opportunities and sales glitches. The application is built using R and Shinyapp following CRISP-DM framework.

Business Case

Objectives

Detect point anomalies from superstore dataset using K-NN and clustering methods

Import data

```
#load libraries
library(readxl)
library(tidyr)
library(dplyr)
library(ggplot2)
library(anomalize)
library(lemon)
library(DMwR)
#library(CORElearn)
library(outForest)
```

```
#read data from file
superstore<-read_excel("superstore.xls")</pre>
```

Data Understanding

US Superstore dataset is sourced from US superstore dataset . The dataset have online orders for Superstores in U.S. from 2014-2018. Tableau community is the owner of the dataset. The dataset has 9994 records and 21 attributes.

Table 1: Dataset description

Attribute	Data Type	Description
Row ID	numeric	row number
Order ID	character	unique order number
Order Date	numeric	order placed date
Ship Date	numeric	order shipping date
Ship Mode	character	shipping mode of order
Customer ID	character	unique customer id for order
Customer Name	character	name of customer
Segment	character	section of product
Country	character	country based on order
City	character	city based on order
State	character	state based on order
Postal Code	numeric	pin code
Region	character	region based on order
Product ID	character	product id of product
Category	character	category of product
Sub-Category	character	sub-category of product
Product Name	character	name of product
Sales	numeric	selling price of product
Quantity	numeric	order quantity
Discount	numeric	discount on product
Profit	numeric	profit from product

Data Preparation

Descriptive Analysis

Continous variables summary

```
superstore %>%
  select_if(is.numeric)%>%
  summary()
```

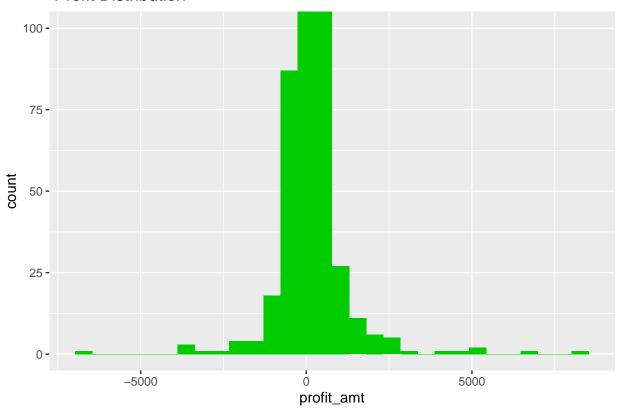
```
##
   postal_code
                  sales_amt
                                     quantity
                                                  discount
## Min. : 1040 Min. : 0.444
                                  Min. : 1.00
                                              Min. :0.0000
## 1st Qu.:23223 1st Qu.:
                          17.280
                                  1st Qu.: 2.00
                                               1st Qu.:0.0000
                          54.490
                                  Median: 3.00
                                               Median :0.2000
## Median:56431 Median:
## Mean :55190 Mean : 229.858
                                  Mean : 3.79
                                                Mean :0.1562
                 3rd Qu.: 209.940
                                  3rd Qu.: 5.00
##
   3rd Qu.:90008
                                                3rd Qu.:0.2000
## Max.
        :99301
               Max. :22638.480
                                  Max. :14.00 Max. :0.8000
##
   profit_amt
## Min. :-6599.978
## 1st Qu.: 1.729
## Median :
            8.666
## Mean :
            28.657
## 3rd Qu.:
            29.364
## Max. : 8399.976
```

Profit

```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=profit_amt),fill="green3")+
  coord_cartesian(ylim = c(0, 100))+
  labs(title=" Profit Distribution")
```

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

Profit Distribution

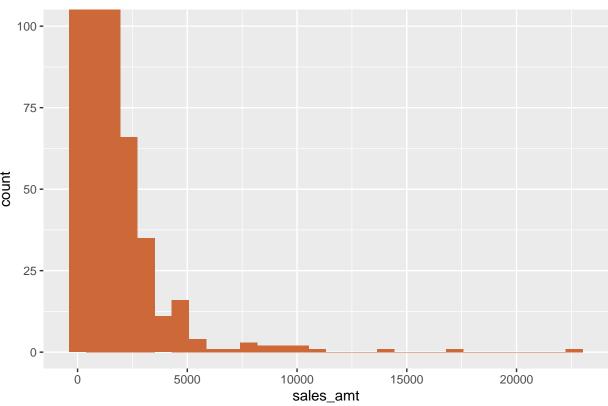


Sales

```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=sales_amt),fill="sienna3")+
  coord_cartesian(ylim = c(0, 100))+labs(title=" Sales Distribution")
```

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

Sales Distribution

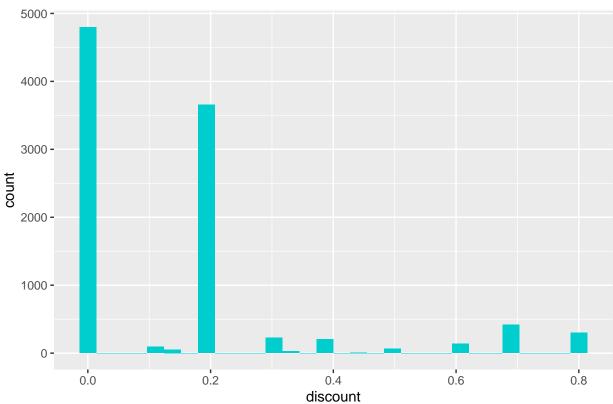


Discount

```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=discount),fill="cyan3")+
  labs(title=" Discount Distribution")
```

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

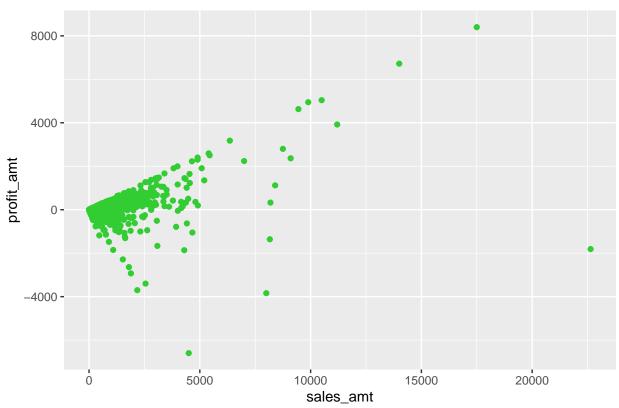
Discount Distribution



Sales Profit

```
ggplot(data = superstore) +
  geom_point(mapping = aes(x = sales_amt, y = profit_amt),colour="limegreen")+
  labs(title=" Sales Profit Distribution")
```

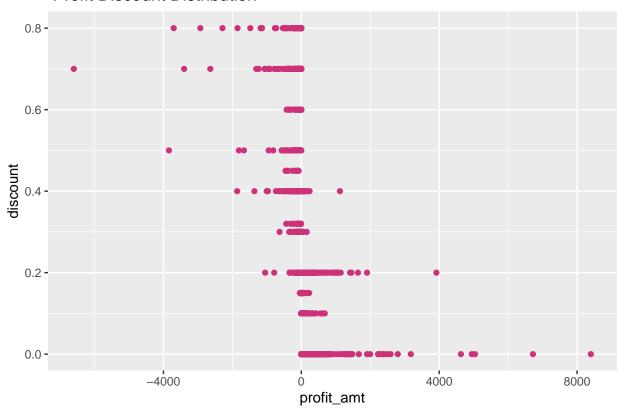
Sales Profit Distribution



Profit Discount

```
ggplot(data = superstore) +
  geom_point(mapping = aes(x = profit_amt, y = discount),colour="violetred3")+
  labs(title=" Profit Discount Distribution")
```

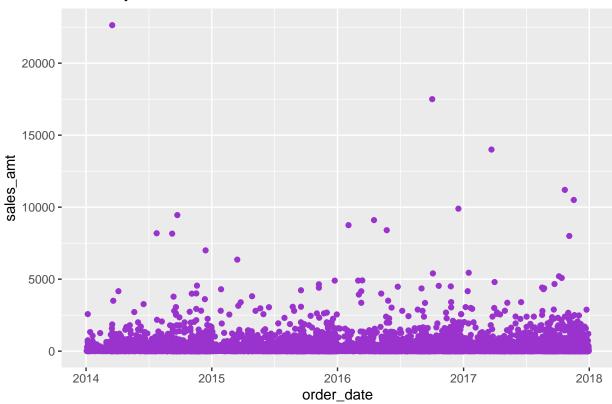
Profit Discount Distribution



Sales by Year

```
ggplot(data=superstore,aes(x = order_date, y =sales_amt)) +
    geom_point(color = "darkorchid3") +
    labs(title=" Sales by Year")
```

Sales by Year

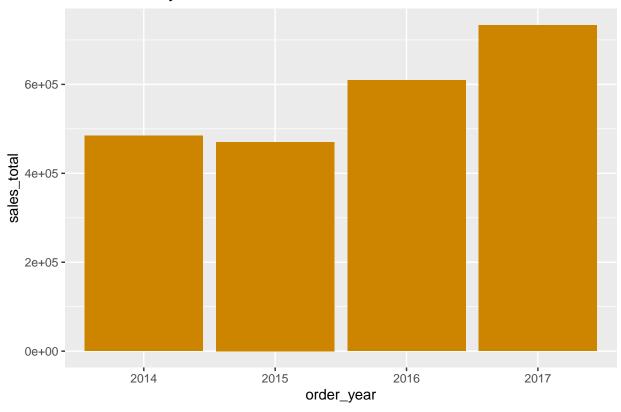


Total Sales by Year

```
sales_year<-aggregate(superstore$sales_amt,by=list(year=format(superstore$order_date, "%Y")),FUN=sum)
names(sales_year)<-c("order_year","sales_total")

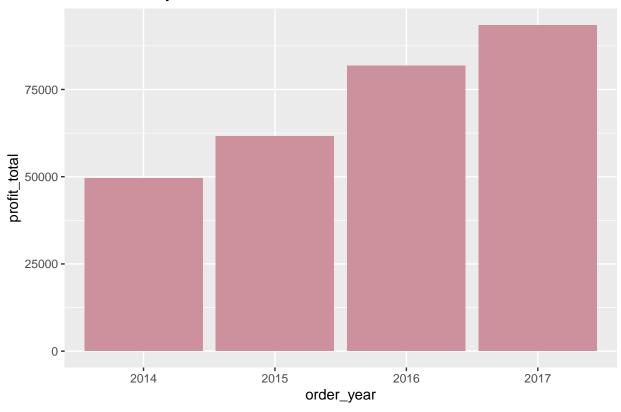
ggplot(data=sales_year,aes(x = order_year, y =sales_total)) +
    geom_bar(stat="identity",fill = "orange3") +
    labs(title=" Total Sales by Year")</pre>
```

Total Sales by Year



Profit by Year

Total Profit by Year



```
# total product id
count_product_id<-unique(superstore$product_id)
length(count_product_id)</pre>
```

[1] 1862

```
#total product name
count_product_name<-unique(superstore$product_name)
length(count_product_name)</pre>
```

[1] 1850

```
#product name and product id mismatch
superstore %>%
  distinct(product_name,product_id) %>%
  group_by(product_id) %>%
  filter(n()>1) %>%
  select(product_id)
```

```
## # A tibble: 64 x 1
## # Groups: product_id [32]
## product_id
## <chr>
## 1 FUR-FU-10004848
```

```
## 2 FUR-CH-10001146
## 3 OFF-BI-10004654
## 4 FUR-CH-10001146
## 5 OFF-PA-10002377
## 6 OFF-AR-10001149
## 7 OFF-PA-10000659
## 8 TEC-MA-10001148
## 9 FUR-FU-10004017
## 10 TEC-AC-10003832
## # ... with 54 more rows
#total category and subcategory
count_category<-unique(superstore$category)</pre>
length(count_category)
## [1] 3
count_subcategory<-unique(superstore$sub_category)</pre>
length(count_subcategory)
## [1] 17
superstore %>%
 distinct(category, sub_category)
## # A tibble: 17 x 2
##
      category sub_category
##
      <chr>
                     <chr>
## 1 Furniture
                    Bookcases
## 2 Furniture
                     Chairs
## 3 Office Supplies Labels
## 4 Furniture
                      Tables
## 5 Office Supplies Storage
## 6 Furniture
                      Furnishings
## 7 Office Supplies Art
## 8 Technology
                      Phones
## 9 Office Supplies Binders
## 10 Office Supplies Appliances
## 11 Office Supplies Paper
## 12 Technology
                      Accessories
## 13 Office Supplies Envelopes
## 14 Office Supplies Fasteners
## 15 Office Supplies Supplies
## 16 Technology
                      Machines
## 17 Technology
                      Copiers
superstore_sales<-superstore %>%
                  select(order_date,sales_amt)
superstore_sales<-as_tibble(superstore_sales)</pre>
```

```
# superstore_sales_anomalized <- superstore_sales %>%
# time_decompose(sales_amt, merge = TRUE) %>%
# anomalize(remainder) %>%
# time_recompose()
```

Model

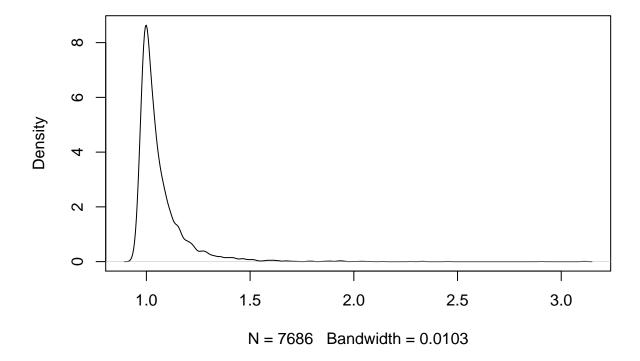
Local Outlier Factor Algorithm -Nearest neighbour method

LOF uses density based methods to calculate degree of outlying.LOF is a unsupervised anomaly detection technique, evry point in the datase is assigned LOF score based on the threshold value it classify the datapoints as outlier or non-outlier.

```
#remove duplicates rows
superstore_unq<-superstore[!duplicated(superstore[c("sales_amt","profit_amt","quantity","discount")]),]
#select numerical variables
superstore_lof<-superstore_unq[,c("sales_amt","profit_amt","quantity","discount")]

# for k=10
lof_scores <- lofactor(superstore_lof, k=10)
plot(density(lof_scores))</pre>
```

density.default(x = lof_scores)



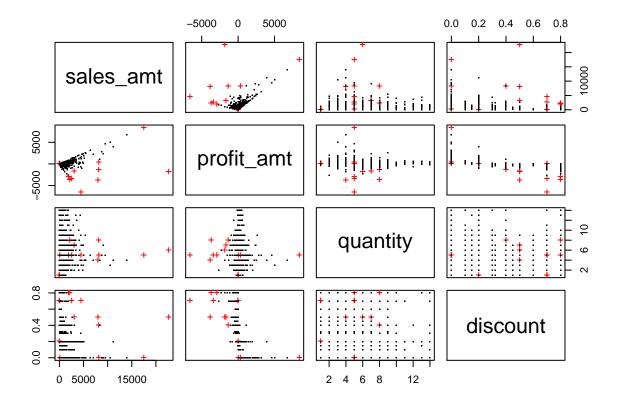
Manual Evaluation

```
#top 5 outliers transactors
 lof_outliers <- order(lof_scores >2, decreasing=T)[1:13]
 superstore_unq[lof_outliers,]
## # A tibble: 13 x 18
##
      orderid order_date ship_date ship_mode customer_id segment city state
##
      <chr>
              <date>
                         <date>
                                    <chr>
                                              <chr>>
                                                          <chr>
                                                                  <chr> <chr>
   1 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
##
## 2 CA-201~ 2014-09-08 2014-09-12 Standard~ BM-11140
                                                          Consum~ San ~ Texas
## 3 US-201~ 2017-11-04 2017-11-04 Same Day GT-14635
                                                          Corpor~ Burl~ Nort~
## 4 CA-201~ 2014-07-25 2014-07-27 Second C~ KL-16645
                                                          Consum~ San ~ Cali~
## 5 CA-201~ 2014-03-18 2014-03-23 Standard~ SM-20320
                                                          Home O~ Jack~ Flor~
## 6 CA-201~ 2017-04-17 2017-04-23 Standard~ SR-20425
                                                          Home O~ Loui~ Colo~
## 7 US-201~ 2017-12-07 2017-12-13 Standard~ HG-14965
                                                          Corpor~ Chic~ Illi~
## 8 CA-201~ 2016-01-04 2016-01-08 Standard~ BP-11185
                                                          Corpor~ Phil~ Penn~
## 9 CA-201~ 2015-09-06 2015-09-08 Second C~ AT-10435
                                                          Home O~ Tama~ Flor~
## 10 CA-201~ 2016-10-02 2016-10-09 Standard~ TC-20980
                                                          Corpor~ Lafa~ Indi~
## 11 CA-201~ 2016-11-25 2016-12-02 Standard~ CS-12505
                                                          Consum~ Lanc~ Ohio
## 12 US-201~ 2017-12-10 2017-12-13 First Cl~ WB-21850
                                                          Consum~ Phil~ Penn~
## 13 CA-201~ 2014-07-26 2014-07-30 Standard~ LF-17185
                                                          Consum~ San ~ Texas
## # ... with 10 more variables: postal_code <dbl>, region <chr>,
      product_id <chr>, category <chr>, sub_category <chr>, product_name <chr>,
## #
      sales_amt <dbl>, quantity <dbl>, discount <dbl>, profit_amt <dbl>
#new column for outlier status
outlier_orderid<-superstore_unq[which(lof_scores >2),1]
vec<-as.vector(outlier orderid$orderid)</pre>
superstore_unq<-mutate(superstore_unq,outlier_status=ifelse(orderid %in% (vec) ,"Yes","No"))</pre>
x<-subset(superstore_unq,superstore_unq$outlier_status=="Yes")</pre>
## # A tibble: 49 x 19
##
      orderid order_date ship_date
                                    ship_mode customer_id segment city state
##
      <chr>
              <date>
                         <date>
                                    <chr>
                                              <chr>>
                                                          <chr>
                                                                  <chr> <chr>
## 1 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
   2 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 3 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 4 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 5 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 6 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 7 US-201~ 2015-09-17 2015-09-21 Standard~ TB-21520
                                                          Consum~ Phil~ Penn~
## 8 CA-201~ 2014-09-08 2014-09-12 Standard~ BM-11140
                                                          Consum~ San ~ Texas
## 9 CA-201~ 2014-09-08 2014-09-12 Standard~ BM-11140
                                                          Consum~ San ~ Texas
## 10 CA-201~ 2014-09-08 2014-09-12 Standard~ BM-11140
                                                          Consum~ San ~ Texas
## # ... with 39 more rows, and 11 more variables: postal_code <dbl>,
## # region <chr>, product_id <chr>, category <chr>, sub_category <chr>,
```

```
## # product_name <chr>, sales_amt <dbl>, quantity <dbl>, discount <dbl>,
## # profit_amt <dbl>, outlier_status <chr>
```

Plot LOF outliers

```
pch <- rep(".", 7000)
pch[lof_outliers] <- "+"
col <- rep("black",7000)
col[lof_outliers] <- "red"
pairs(superstore_lof, pch=pch, col=col)</pre>
```



Random Forest Algorithm

outForest is a random forest based implementation of the method. Each numeric variable is regressed onto all other variables using a random forest. If the scaled absolute difference between observed value and out-of-bag prediction is suspiciously large (e.g. more than three times the RMSE of the out-of-bag predictions), then a value is considered an outlier. After identification of outliers, they can be replaced e.g. by predictive mean matching from the non-outliers.

```
#dataframe for outforest
superstore_out<-superstore_unq[,c("sales_amt","profit_amt","quantity","discount")]</pre>
```

```
#fit outforest
outforest_outForest(superstore_out)
```

```
##
## Outlier identification by random forests
##
## Variables to check: sales_amt, profit_amt, quantity, discount
## Variables used to check: sales_amt, profit_amt, quantity, discount
##
## Checking: sales_amt profit_amt quantity discount
```

Manual Evaluation

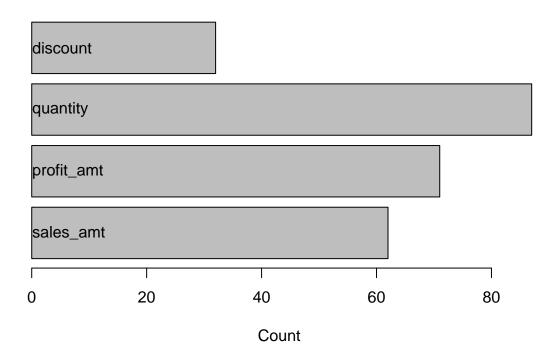
Observed same set of top outliers as LOF algorithm

```
#outlier rows, observed values, predicted and RMSE
head(outliers(outforest_outlier),13)
```

```
##
       row
                  col observed predicted
                                               rmse
                                                       score threshold
## 18 2452 sales_amt 22638.480 1995.8288 431.5942
                                                    47.82884
                                                                     3
## 113 5613 profit_amt 8399.976 2721.1453 172.6538
                                                                     3
                                                    32.89144
## 117 6269 profit_amt -6599.978 -1018.6539 172.6538 -32.32669
                                                                     3
## 40 5613 sales_amt 17499.950 6213.8618 431.5942
                                                    26.14977
                                                                     3
## 70
       665 profit_amt -3839.990
                                                                     3
                                324.9757 172.6538 -24.12323
## 120 6517 profit_amt 6719.981 3178.4311 172.6538 20.51244
                                                                     3
## 81 2452 profit_amt -1811.078 1696.5543 172.6538 -20.31599
                                                                     3
## 48 6517 sales_amt 13999.960 6328.6396 431.5942 17.77438
                                                                     3
                                                                     3
## 15 2290 sales_amt 8187.650 1115.6691 431.5942 16.38572
## 133 7555 profit_amt -3701.893 -954.7349 172.6538 -15.91137
                                                                     3
       164 sales amt 8159.952 1555.9688 431.5942 15.30137
                                                                     3
## 2
## 94 3679 profit_amt 5039.986 2490.2289 172.6538 14.76803
                                                                     3
## 24 3679 sales amt 10499.970 4204.6149 431.5942 14.58628
                                                                     3
##
      replacement
## 18
        2104.5500
## 113
        1351.9896
## 117 -1002.7836
        4643.8000
## 40
## 70
         503.7822
## 120
       1351.9896
## 81
         610.8624
## 48
        5443.9600
## 15
         836.5920
## 133
       -1002.7836
## 2
        2036.8600
## 94
        1013.1270
## 24
        2999.9500
```

```
# Outliers per variable
plot(outforest_outlier)
```

Number of outliers per variable



Responsible ML Framework

Conclusion

Bibliography