Anomaly Inclass

October 29, 2020

`

Anomaly vs Outlier Detection

Recall that the distinction between Anomaly and Outlier detection is not so much in the methodological approach taken, but more in the process that generated the data. Detecting the potential existence of an different (anomalous) process that generates some subset of your data is the goal of anomaly detection, which means that sometimes the underlying analytical challenge is anomaly detection. In the walk through we will not distinguish between outlier and anomaly detection until the last section, where we will focus on anomalous pattern detection which by definition assumes there is a anomalous process (or an even which shifts the normal behavior) generating a subset of data.

Local Outlier Factor

We will first investigate using a density based outlier detection approach known as Local Ouliter Factor (LOF), which we dicussed in class. We are going to use a dataset of 400 call center employees, and our goal is to identify outliers among this group given their job performance data. After loading the data, do some investigate the data structure and conduct some simple explorations # `{r} #` {r message = FALSE, results = FALSE, results = 'hide'} # Read source data

Remember density based ouliter detection is based on distances, and this means we should normalize our data. Recall tha the `scale` function will be useful for this

```
#Lets look at summary stats
skim(callCenter[,2:11])
```

Data summary

Name	callCenter[, 2:11]
Number of rows	400
Number of columns	10
_______________________	
Column type frequency:	
numeric	10
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Avg.Tix…Day	0	1	156.09	4.42	143.10	153.07	156.05	159.10	168.70	▁▅▇▅▁
Customer.rating	0	1	3.50	0.46	2.07	3.21	3.50	3.81	4.81	▁▃▇▅▁
Tardies	0	1	1.47	0.97	0.00	1.00	1.00	2.00	4.00	▃▇▆▂▁
Graveyard.Shifts.Taken	0	1	1.99	0.79	0.00	1.00	2.00	2.00	4.00	▁▃▇▃▁
Weekend.Shifts.Taken	0	1	0.95	0.55	0.00	1.00	1.00	1.00	2.00	▂▁▇▁▂
Sick.Days.Taken	0	1	1.88	1.67	0.00	0.00	2.00	3.00	7.00	▇▃▅▁▁
X..Sick.Days.Taken.on.Friday	0	1	35.22	39.30	0.00	0.00	25.00	67.00	100.00	▇▁▁▁▂
Employee.Dev..Hours	0	1	11.97	7.47	0.00	6.00	12.00	17.00	34.00	▆▇▇▂▁
Shift.Swaps.Requested	0	1	1.45	1.00	0.00	1.00	1.00	2.00	5.00	▇▅▂▁▁
Shift.Swaps.Offered	0	1	1.76	1.81	0.00	0.00	1.00	3.00	9.00	▇▅▂▁▁

```
# Normalize the data using the scale() function -- for each observation, this subtracts the column's mean and divides by the
column's standard deviation
#column 1 is employer ID so it is not notmalized.
#scale returns a matrix, so data.frame converts that matrix to a data frame, which is stored in callCenterSc
callCenterSc <- data.frame(scale(callCenter[2:11]))
skim(callCenterSc)
```

Data summary

Name	callCenterSc
Number of rows	400
Number of columns	10
_______________________	
Column type frequency:	
numeric	10
________________________	
Group variables	None

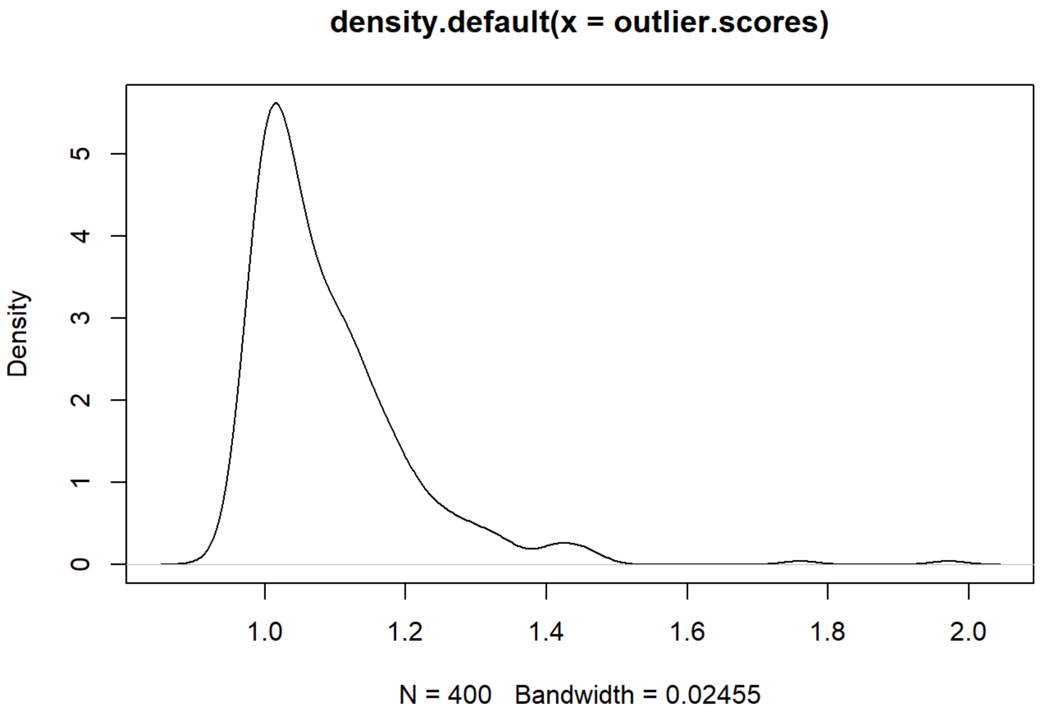
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Avg.Tix…Day	0	1	0	1	-2.94	-0.68	-0.01	0.68	2.86	▁▅▇▅▁
Customer.rating	0	1	0	1	-3.09	-0.62	0.02	0.68	2.85	▁▃▇▅▁
Tardies	0	1	0	1	-1.51	-0.48	-0.48	0.55	2.61	▃▇▆▂▁
Graveyard.Shifts.Taken	0	1	0	1	-2.50	-1.24	0.02	0.02	2.54	▁▃▇▃▁
Weekend.Shifts.Taken	0	1	0	1	-1.74	0.09	0.09	0.09	1.91	▂▁▇▁▂
Sick.Days.Taken	0	1	0	1	-1.12	-1.12	0.07	0.67	3.06	▇▃▅▁▁
X..Sick.Days.Taken.on.Friday	0	1	0	1	-0.90	-0.90	-0.26	0.81	1.65	▇▂▂▂▃
Employee.Dev..Hours	0	1	0	1	-1.60	-0.80	0.00	0.67	2.95	▆▇▇▂▁
Shift.Swaps.Requested	0	1	0	1	-1.45	-0.45	-0.45	0.55	3.55	▇▅▂▁▁
Shift.Swaps.Offered	0	1	0	1	-0.97	-0.97	-0.42	0.68	3.99	▇▅▂▁▁

Next we want to use the `lof` function to calculate the local outlier factor (lof). Let us choose k = 5 Nearest Neighbors and plot the density of the scores.

```
#Calculate the outlier scores
#lofactor - the parameter 5 indicates 5 nearest neight bours
#outlier.scores  contains the lof for ecah row in the lofactor dataframe
outlier.scores <- lofactor(callCenterSc,5) #there is also a fucntion called lof() from the Rlof library.

#plot score density
plot(density(outlier.scores))
```



Consider the employees for which lof > 1.5. A value of 1 signifies that a node and its neighbors are similarly distant from each other.

```
callCenter[which(outlier.scores > 1.5),]
```

```
##     Employee.ID Avg.Tix...Day Customer.rating Tardies Graveyard.Shifts.Taken
## 299      137155         165.3            4.49       1                      3
## 374      143406         145.0            2.33       3                      1
##     Weekend.Shifts.Taken Sick.Days.Taken X..Sick.Days.Taken.on.Friday
## 299                    2               1                            0
## 374                    0               6                           83
##     Employee.Dev..Hours Shift.Swaps.Requested Shift.Swaps.Offered
## 299                  30                     1                   7
## 374                  30                     4                   0
```

```
# or
#this is the deep layer way of doing this - BETTER
#here the employer ID is joined to the result of filter(outlier.scores > 1.5)
callCenter %>% filter(outlier.scores > 1.5)
```

```
##   Employee.ID Avg.Tix...Day Customer.rating Tardies Graveyard.Shifts.Taken
## 1      137155         165.3            4.49       1                      3
## 2      143406         145.0            2.33       3                      1
##   Weekend.Shifts.Taken Sick.Days.Taken X..Sick.Days.Taken.on.Friday
## 1                    2               1                            0
## 2                    0               6                           83
##   Employee.Dev..Hours Shift.Swaps.Requested Shift.Swaps.Offered
## 1                  30                     1                   7
## 2                  30                     4                   0
```

```
#an equivalent is (old way of doing the above)
filter(callCenter, outlier.scores > 1.5)
```

```
##   Employee.ID Avg.Tix...Day Customer.rating Tardies Graveyard.Shifts.Taken
## 1      137155         165.3            4.49       1                      3
## 2      143406         145.0            2.33       3                      1
##   Weekend.Shifts.Taken Sick.Days.Taken X..Sick.Days.Taken.on.Friday
## 1                    2               1                            0
## 2                    0               6                           83
##   Employee.Dev..Hours Shift.Swaps.Requested Shift.Swaps.Offered
## 1                  30                     1                   7
## 2                  30                     4                   0
```

There should be two outlier employees, based on their values give some intuitions for why you think they they are outliers?

** Answer: ** One has especially generous and flexible performance and the other’s performance lags across multiple dimensions.

```
# we compare the dimensions of the outliers (eg sick dateys taken on fridays, emp dev hours etc) to the averages from teh co
mmand below
skim(callCenter)
```

Data summary

Name	callCenter
Number of rows	400
Number of columns	11
_______________________	
Column type frequency:	
numeric	11
________________________	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Employee.ID	0	1	137946.04	4240.88	130564.00	134401.50	137906.50	141771.25	145176.00	▇▇▇▇
Avg.Tix…Day	0	1	156.09	4.42	143.10	153.07	156.05	159.10	168.70	▁▃▇▃▁
Customer.rating	0	1	3.50	0.46	2.07	3.21	3.50	3.81	4.81	▁▂▇▅▁
Tardies	0	1	1.47	0.97	0.00	1.00	1.00	2.00	4.00	▂▇▆▁
Graveyard.Shifts.Taken	0	1	1.99	0.79	0.00	1.00	2.00	2.00	4.00	▁▃▇▃▁
Weekend.Shifts.Taken	0	1	0.95	0.55	0.00	1.00	1.00	1.00	2.00	▂▁▇▁
Sick.Days.Taken	0	1	1.88	1.67	0.00	0.00	2.00	3.00	7.00	▇▂▅▁
X..Sick.Days.Taken.on.Friday	0	1	35.22	39.30	0.00	0.00	25.00	67.00	100.00	▇▁▁▁
Employee.Dev..Hours	0	1	11.97	7.47	0.00	6.00	12.00	17.00	34.00	▆▇▇▁
Shift.Swaps.Requested	0	1	1.45	1.00	0.00	1.00	1.00	2.00	5.00	▇▅▂▁
Shift.Swaps.Offered	0	1	1.76	1.81	0.00	0.00	1.00	3.00	9.00	▇▅▂▁