DEA

Yashwanth k

2023-11-05

#summary

In summary,your data centers exhibit a range of efficiency levels. To improve energy consumption and performance, it's essential to focus on those with lower efficiency scores, especially the ones falling below the first quartile. Implementing innovative and transparent energy policies and corrective actions in these data centers can lead to significant cost reduction and sustainability improvements. For data centers with higher efficiency scores, the emphasis should be on maintaining their current efficiency levels.

Analyze the results

Efficiency Scores: - An efficiency score of 1 indicates perfect efficiency, while 0 means complete inefficiency.

- Data Centers with Maximum Efficiency:
- Data Centers 1, 2, 5, 7, 10,13 and 15 are performing exceptionally well with efficiency scores of 1. They are optimizing their resources and energy consumption effectively.
- High Efficiency Data Centers:
- Data Centers 3,14, and 16 have efficiency scores close to 1 (around 0.99), indicating that they are also operating efficiently. However, there's still room for improvement.
- Data Centers with Room for Improvement:
- Data Centers 8,9 and 11 have lower efficiency scores, suggesting inefficiencies in their operations. They may benefit from implementing energy-saving measures and process improvements.
- Data Centers with Significant Inefficiency:
- Data Centers 4,6,12, 17 and 18 have the lowest efficiency scores (around 0.47), indicating significant inefficiencies. They should prioritize energy-saving technologies and operational improvements to reduce energy consumption and enhance performance.

-Peers:

This describes the connection between Decision Making Units (DMUs) that share similar input and output characteristics, allowing for efficiency comparisons. It serves as a tool to evaluate the performance of a specific DMU and offers insights for enhancing its efficiency.

Example: In this context, the 11th DMU is linked with DMUs 2, 13, and 15.

- Lambda: In the context of DEA, when assessing the efficiency score of a DMU, the "lambda" function is used to determine the weights or multipliers assigned to its inputs and outputs. These weights signify the relative importance or contribution of each input and output to the overall efficiency of the DMU.

Example: If the lambda value is 1, it means that particular attribute carries greater weight in determining the efficiency score of the Decision Making Unit. If the value is less than 1, it still contributes to the score, but it is not the sole determining factor.

- Conclusion:
- It's crucial to continuously monitor and assess the efficiency of your data centers. Use DEA to identify areas of inefficiency and take corrective actions. Prioritize energy-saving initiatives to make your data centers more sustainable and cost-effective. One method to enhance process efficiency is to identify and remove bottlenecks. Another approach is to improve efficiency by investing in new technology or equipment.

```r
df = read.csv('energy.csv')
head(df)
```

```
##   X Energy.Policy Scheduling.Model Work.Load D.C..Size Shut.Downs
## 1 1        Always       Monolithic      High      1000      37166
## 2 2        Margin            Mesos      High      1000      13361
## 3 3         Gamma            Omega      High      1000      14252
## 4 4        Always            Mono.       Low      1000      36404
## 5 5   Exponential            Mesos       Low      1000      19671
## 6 6          Load            Omega       Low      1000      32407
##   Computing.Time..h. MWh.Consumed Queue.Time..ms.
## 1             104.42        49.01            90.1
## 2             104.26        49.65          1093.0
## 3             104.17        49.60             0.1
## 4              49.25        23.92            78.3
## 5              49.63        24.65          1188.7
## 6              49.34        24.19             1.1
```

```r
str(df)
```

```
## 'data.frame':    18 obs. of  9 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Energy.Policy     : chr  "Always" "Margin" "Gamma" "Always" ...
##  $ Scheduling.Model  : chr  "Monolithic" "Mesos" "Omega" "Mono." ...
##  $ Work.Load         : chr  "High" "High" "High" "Low" ...
##  $ D.C..Size         : int  1000 1000 1000 1000 1000 1000 5000 5000 5000 5000 ...
##  $ Shut.Downs        : int  37166 13361 14252 36404 19671 32407 6981 9877 33589 8578 ...
##  $ Computing.Time..h.: num  104.4 104.3 104.2 49.2 49.6 ...
##  $ MWh.Consumed      : num  49 49.6 49.6 23.9 24.6 ...
##  $ Queue.Time..ms.   : num  90.1 1093 0.1 78.3 1188.7 ...
```

```r
# Load the Benchmarking package
library(Benchmarking)
```

```
## Warning: package 'Benchmarking' was built under R version 4.3.2

## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog
```

```r
# Prepare your data
inputs <- df[, c("D.C..Size", "Shut.Downs")]
outputs <- df[, c("Computing.Time..h.", "MWh.Consumed", "Queue.Time..ms.")]

# Create a DEA model with Constant Returns to Scale (CRS)
dea_model <- dea(inputs, outputs, RTS = "crs")

# Assess the efficiency
efficiency_scores <- eff(dea_model)
```

```r
efficiency_scores
```

```
##  [1] 1.0000000 1.0000000 0.9991215 0.4817724 1.0000000 0.4872105 1.0000000
##  [8] 0.9826417 0.9577554 1.0000000 0.9805683 0.4753953 1.0000000 0.9943765
## [15] 1.0000000 0.9970310 0.5290248 0.4783408
```

```r
# Analyze the results
summary(efficiency_scores)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4754  0.6362  0.9957  0.8535  1.0000  1.0000
```