R Notebook

Name: Gabriel Bentley

Date: 10/05/22

Dataset: Gas Turbine Metric Measurements

https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set

Data set information

The dataset contains 14795 instances of 11 sensor measures aggregated over one hour (by means of average or sum) from a gas turbine located in Turkey's north western region for the purpose of studying flue gas emissions, namely CO and NOx (NO + NO2). This data is collected in another data range (01.01.2011 - 31.12.2011) and (01.01.2015 - 31.12.2015), includes gas turbine parameters (such as Turbine Inlet Temperature and Compressor Discharge pressure) in addition to the ambient variables.

Attribute information

The explanations of sensor measurements and their brief statistics are given below.

Variable (Abbr.) Unit Min Max Mean Ambient temperature (AT) C â€“6.23 37.10 17.71 Ambient pressure (AP) mbar 985.85 1036.56 1013.07 Ambient humidity (AH) (%) 24.08 100.20 77.87 Air filter difference pressure (AFDP) mbar 2.09 7.61 3.93 Gas turbine exhaust pressure (GTEP) mbar 17.70 40.72 25.56 Turbine inlet temperature (TIT) C 1000.85 1100.89 1081.43 Turbine after temperature (TAT) C 511.04 550.61 546.16 Compressor discharge pressure (CDP) mbar 9.85 15.16 12.06 Turbine energy yield (TEY) MWH 100.02 179.50 133.51 Carbon monoxide (CO) mg/m3 0.00 44.10 2.37 Nitrogen oxides (NOx) mg/m3 25.90 119.91 65.29

Read in and seperate the data set into train and test

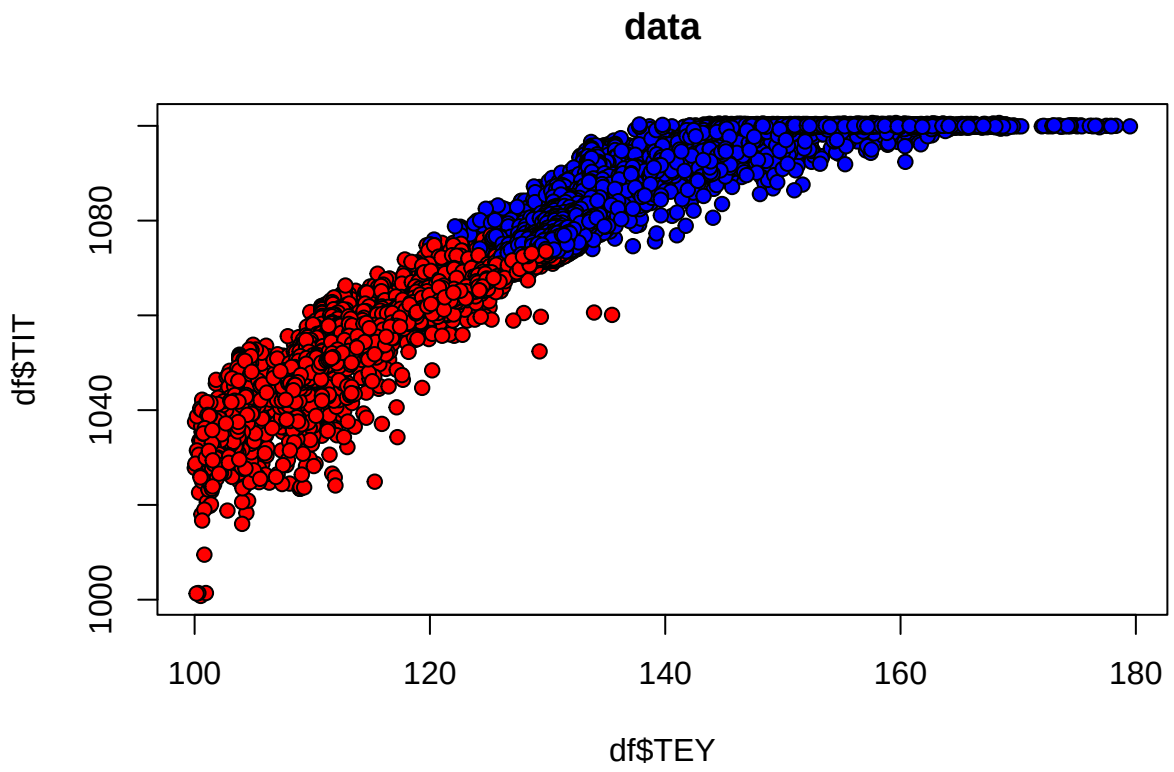
```
df <- read.csv("gt_2011.csv")

df$year <- rep(c("2011"), each = 7411)


df2 <- read.csv("gt_2015.csv")
df2$year <- rep(c("2015"), each = 7384)



df <- rbind(df, df2)


df <- na.omit(df)

df$year <- factor(df$year)
```

```
set.seed(8788)
kmCluster <- kmeans(df, 2, nstart = 20)
```

```
table(kmCluster$cluster, df$year)
```

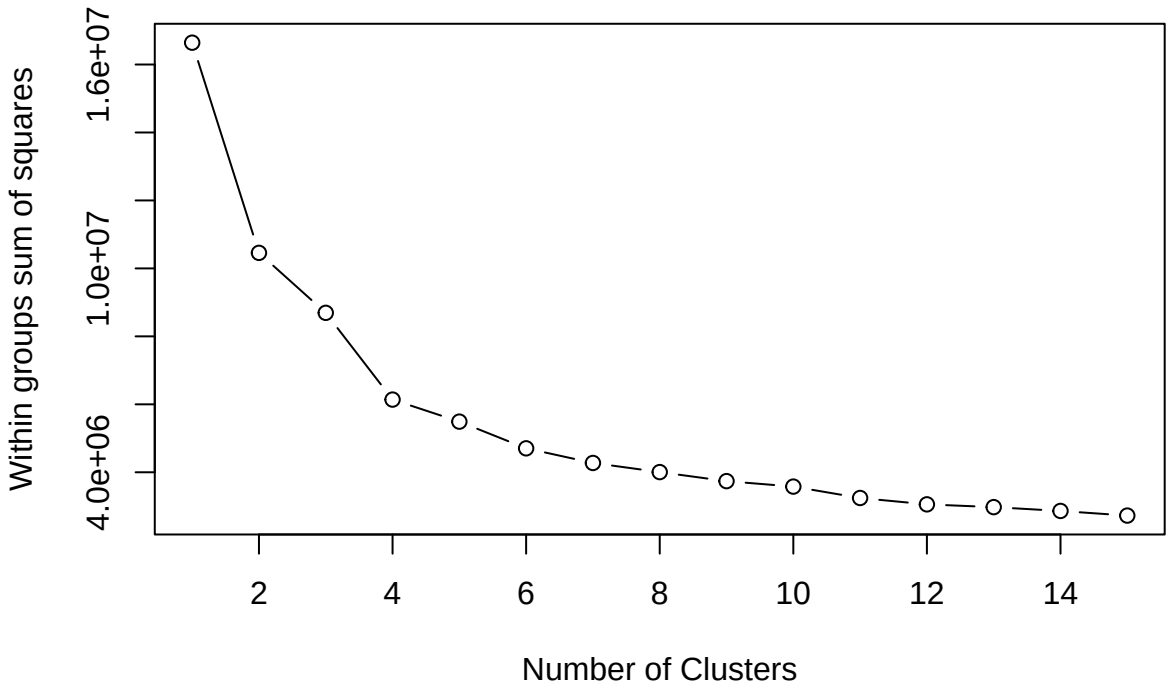
```
##
##     2011 2015
##   1 1677 2075
##   2 5734 5309
```

```
plot(df$TEY, df$TIT, pch=21, bg=c("red","blue")[unclass(kmCluster$cluster)],main = "data")
```



Determine number of clusters for K means

```
wss <- (nrow(df)-1)*sum(apply(df,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(df,
   centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```

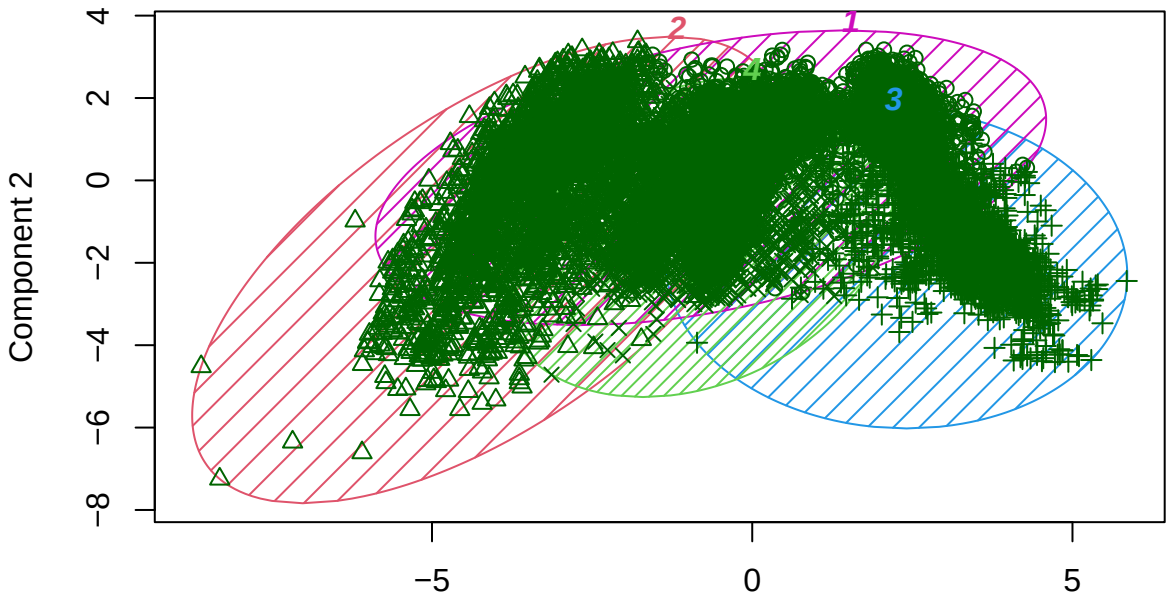


K Means Clustering

```
library(flexclust)
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: modeltools
## Loading required package: stats4
```

```
library(cluster)
set.seed(8788)
kmCluster <- kmeans(df, 4, nstart = 20)

clusplot(scale(df[,1:11]), kmCluster$cluster, color=TRUE, shade=TRUE, labels = 4, lines = 0)
```




```
ct.km <- table(df$year, kmCluster$cluster)
ct.km
```

```
##
##            1    2    3    4
##   2011 1229 1486 1768 2928
##   2015 2376 1659 1112 2237
```

```
cat("Cluster sizes: \n")
```

```
## Cluster sizes:
```

```
kmCluster$size
```

```
## [1] 3605 3145 2880 5165
```

```
cat("Cluster centers: \n")
```

```
## Cluster centers:
```

```
kmCluster$centers
```

```
##          AT       AP       AH     AFDP     GTEP      TIT      TAT      TEY
## 1 25.38734 1011.478 56.80470 4.076833 28.29355 1093.425 546.6053 140.9052
## 2 16.85406 1013.753 77.12357 3.018045 20.15382 1052.068 548.3190 111.1711
## 3 13.63272 1016.867 78.25835 4.688731 31.86953 1099.697 533.7539 157.6086
## 4 13.59548 1015.281 81.49857 3.717066 24.38966 1081.982 549.7646 132.4122
##         CDP       CO      NOX     year
## 1 12.67644 1.599857 58.16120 2013.636
## 2 10.56393 4.740826 66.82175 2013.110
## 3 13.77764 1.356155 60.45065 2012.544
## 4 11.84764 1.971422 67.59135 2012.732
```

```
aggregate(scale(df[,1:11]),by=list(kmCluster$cluster),FUN=mean)
```

```
##   Group.1          AT          AP         AH       AFDP       GTEP          TIT
## 1       1  1.05794273 -0.43319607 -1.1810100  0.3392972  0.5440033  0.633191341
## 2       2 -0.04049601 -0.08860812  0.2209881 -1.2121603 -1.3032912 -1.631092735
## 3       3 -0.45515991  0.38302510  0.2992877  1.2359198  1.3555642  0.976612595
## 4       4 -0.45995413  0.14273613  0.5228625 -0.1878744 -0.3419759  0.006677172
##          TAT        TEY        CDP         CO        NOX
## 1  0.1454362  0.3720737  0.4585194 -0.3421249 -0.4823257
## 2  0.3863641 -1.4613953 -1.3898587  1.0907618  0.2664438
## 3 -1.6613859  1.4020436  1.4220281 -0.4533001 -0.2843861
## 4  0.5896183 -0.1516211 -0.2666598 -0.1726198  0.3329816
```

```
ran <- randIndex(ct.km)

cat("Rand Index: ", ran)
```

```
## Rand Index:  0.02048952
```

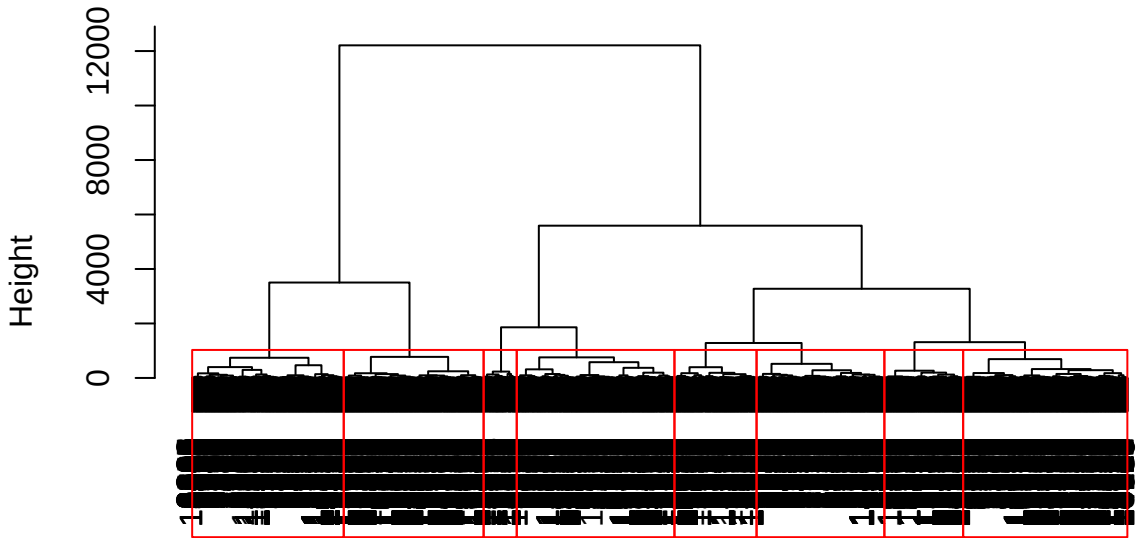
Since the rand Index for the kMeans cluster was close to zero means that the clustering was practically random.

Hierarchical Clustering

```
# Ward Hierarchical Clustering
df.scaled = scale(df[,1:11])

d <- dist(df.scaled, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
plot(fit) # display dendogram

groups <- cutree(fit, k=8)

rect.hclust(fit, k=8, border="red")
```




```
for (c in 3:11) {
  cluster_cut <- cutree(fit, k=c)
  table_cut <- table(cluster_cut, df$year)
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand index = ", ri))
}
```

```
##
## cluster_cut 2011 2015
##           1 3587 3578
##           2 2307 2304
##           3 1517 1502
## [1] "cut= 3 Rand index =  -8.43383788854647e-05"
##
## cluster_cut 2011 2015
##           1 3587 3578
##           2 1691  707
##           3 1517 1502
##           4  616 1597
## [1] "cut= 4 Rand index =  0.0175508385373762"
##
## cluster_cut 2011 2015
##           1 2030 1291
##           2 1691  707
##           3 1517 1502
##           4 1557 2287
##           5  616 1597
## [1] "cut= 5 Rand index =  0.0273953383727107"
```

```
##
## cluster_cut 2011 2015
##           1 2030 1291
##           2 1691  707
##           3 1323 1172
##           4  194  330
##           5 1557 2287
##           6  616 1597
## [1] "cut= 6 Rand index =  0.0277691567469568"
##
## cluster_cut 2011 2015
##           1 2030 1291
##           2 1691  707
##           3 1323 1172
##           4  194  330
##           5 1038 1560
##           6  519  727
##           7  616 1597
## [1] "cut= 7 Rand index =  0.0257811397138855"
##
## cluster_cut 2011 2015
##           1 1984   38
##           2 1691  707
##           3 1323 1172
##           4   46 1253
##           5  194  330
##           6 1038 1560
##           7  519  727
##           8  616 1597
## [1] "cut= 8 Rand index =  0.0687050281692467"
##
## cluster_cut 2011 2015
##           1 1984   38
##           2 1691  707
##           3 1323 1172
##           4   46 1253
##           5  194  330
##           6 1038 1560
##           7  519  727
##           8  339  724
##           9  277  873
## [1] "cut= 9 Rand index =  0.0645100913715758"
##
## cluster_cut 2011 2015
##          1  1984   38
##          2  1691  707
##          3   415  497
##          4    46 1253
##          5   194  330
##          6   908  675
##          7  1038 1560
##          8   519  727
##          9   339  724
##          10  277  873
```

```
## [1] "cut= 10 Rand index =  0.0648576860527577"
##
## cluster_cut 2011 2015
##           1  1984   38
##           2  1132  239
##           3   415  497
##           4   559  468
##           5    46 1253
##           6   194  330
##           7   908  675
##           8  1038 1560
##           9   519  727
##           10  339  724
##           11  277  873
## [1] "cut= 11 Rand index =  0.0633709760296745"
```

Model based clustering

```
library(mclust)
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
fit <- Mclust(df)

summary(fit) # display the best model
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 6 components:
##
##  log-likelihood     n  df       BIC       ICL
##       -250125.8 14795 490 -504956.6 -505153.4
##
## Clustering table:
##    1    2    3    4    5    6
## 3291 2376 4483  702 2470 1473
```

Analysis of results

K Mean Result: The clustering of the data set using K Mean was close to random as the rand index produced was 2% close to zero.

Hierarchical Result: The clustering of the data using a Hierarchical result was also close to random but slightly better than the K Mean results. The best cut of the hierarchical result was a cut of 8 to the tree resulting in a rand index of 6.8%.

Model Results: The model I used for this clustering technique was a classification model and the summary of the fit for the model used 5 different clusters. My computer was not capable of kniting the model plot to pdf so I removed it from the notebook.

The fact that the results of clustering was random shows that the data points for turbine values in 2011 are very similar to turbine values of 2015, or that they are close together.