FML_FINAL

Sania Fatima

2023-12-02

#SUMMARY:

##OBJECTIVE:

Understanding the structure,patterns of the given dataset using DBSCAN clustering algorithm.

##APPROACH:

Organising the datapoint into similar groups where our focus is to minimize the variations within the clusters and to maximize the variations between the clusters.

##METHOD:

Applying the DBSCAN method to form clusters: It is based on two things 1) Epsilon(Closeness of datapoints) 2) Density(Minimum Points) *** ##INTERPRETATION:

#1) Using k=2 in KNNdistplot, determine epsilon.What was your epsilon.

After constructing the knee method, the epsilon value which I took was 0.3.Epsilon shows the closeness of datapoints.The datapoints within the clusters should have high similarity and should have less similarity between the clusters.The choice of epsilon has significant impact on clusters forming.

DBscan is a non-linear density based clustering algorithm that groups the data into dense clusters by measuring the closeness and minimum points and filtering the noise points that lie in the low density region.Density here is given by the minimum points(5) ,the closeness of the points which is given by epsilon(0.3) DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.

#2)How many clusters did you decide?

The number of cluster are 3. DBSCAN doesnot require to specify the number of clusters in the data intially as in the case of kmeans.Based on the Core,Boundary and Noise points we can say that the three clusters with epsilon(0.3) and minimum points(5) have formed.

#3)Describe your clusters.Provide relevant tables and graphs to support your conclusion.

CLUSTER NUMBER 1:COST EFFECTIVE :The cluster number 1 consists of 391 datapoints.It is showing the 1.16 of sulphur content in it with ash content of 8.77 where the fuel cost (mmbtu)is 2.11.As per the domain knowledge,we can say that the the sulphur content has a minimal and decent amount with little high ash prices but the cost of fuel is low.

CLUSTER NUMBER 2:LOW EMISSION:The cluster Number 2 consists of 263 datapoints.It contains the 0.00 of sulphur content and ash content where fuel cost is 8.07.This clusters shows that it is less polluted to environment as there is no carbon emmission present such as sulphur and ash.

CLUSTER NUMBER 3:LOW POLLUTION:This cluster consists of 64 datapoints where it has the sulphur content of 0.13 with no ash content and 19.53 fuel cost.This cluster consists of no ash content which is less harmful to environment and has a minimal number of sulphur.

#4)What can you say about the relative composition of the different fuels types in relation to your clusters?

According to the graph,there is the pattern between the fuel type and clusters including sulfur content,Ash content, and Fuel cost.we can say that the relative fuel(fuel_type_code_pudl) composition in cluster number 1 consists of 391 datapoints which only have coal present in it with minimal sulphur() ,ash and cost price of fuel.Moreover, it has supplier involved for the demand of coal is interocean coal and d & e mining .The cluster 2 covers the vast majority of points with gas which has a count of 259 and has 5 points of gas present with the zero sulphur,ash content and has a price of 8.63.Moreover, it has supplier invovled for gas is bay gas pipeline whereas the cluster number 3 has 59 datapoints of oil and 4 datpoints of gas.

```
knitr::opts_chunk$set(echo = TRUE, comment = NULL)
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
```

Data Importing and Cleaning:

1.Loading the required libraries.

```
library(tidyverse, warn.conflicts = FALSE)
library(factoextra, warn.conflicts = FALSE)
library(caret, warn.conflicts = FALSE)
library(e1071, warn.conflicts = FALSE)
library(cluster, warn.conflicts = FALSE)
library(dplyr, warn.conflicts = FALSE)
library(tinytex, warn.conflicts = FALSE)
library(dbscan, warn.conflicts = FALSE)
library(fpc, warn.conflicts = FALSE)
library(ellipse, warn.conflicts = FALSE)
```

2.Importing and reading the dataset.

```
library(readr)
Fuel.receipt<- read.csv("C:/Users/Sania fatima/Desktop/Final Exam FML/fuel_receipts1.csv")
dim(Fuel.receipt)
```

```
[1] 756  12
```

```
str(Fuel.receipt)
```

```
'data.frame':   756 obs. of  12 variables:
 $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ rowid              : int  1 2 3 4 5 6 7 8 9 10 ...
 $ plant_id_eia       : int  3 3 3 7 7 7 7 8 8 8 ...
 $ energy_source_code : chr  "BIT" "BIT" "NG" "BIT" ...
 $ fuel_type_code_pudl: chr  "coal" "coal" "gas" "coal" ...
 $ fuel_group_code    : chr  "coal" "coal" "natural_gas" "coal" ...
 $ supplier_name      : chr  "interocean coal" "interocean coal" "bay gas pipeline" "alabama coal" ...
 $ fuel_received_units: int  259412 52241 2783619 25397 764 603 2341 8869 75442 206741 ...
 $ fuel_mmbtu_per_unit: num  23.1 22.8 1.04 24.61 24.45 ...
 $ sulfur_content_pct : num  0.49 0.48 0 1.69 0.84 1.54 0 2.16 1.24 1.9 ...
 $ ash_content_pct    : num  5.4 5.7 0 14.7 15.5 14.6 0 15.4 11.9 15.4 ...
 $ fuel_cost_per_mmbtu: num  2.13 2.12 8.63 2.78 3.38 ...
```

There are total 756 observation with 12 variables which includes the categories as integer,chatacter,numerical.

3.Dropping the categorical variables.

```
set.seed(1)
fuel<- Fuel.receipt[,-c(2,3,4,5,6,7,8,9)]
```

```
row.names(fuel) <- fuel[,1]
fuel <- fuel[,-1]
head(fuel)
```

```
  sulfur_content_pct ash_content_pct fuel_cost_per_mmbtu
1               0.49             5.4               2.135
2               0.48             5.7               2.115
3               0.00             0.0               8.631
4               1.69            14.7               2.776
5               0.84            15.5               3.381
6               1.54            14.6               2.199
```

The reason for excluding the specific columns(2,3,4,5,6,7) is that they contain categorical variables that are not suitable for forming the clusters.It's a preprocessing step before applying the clustering techniques.

All the categorical variables have been dropped.

4.Normalizing the data by using the scale function.

```
Fuel.norm.df <- scale(fuel)
head(Fuel.norm.df)
```

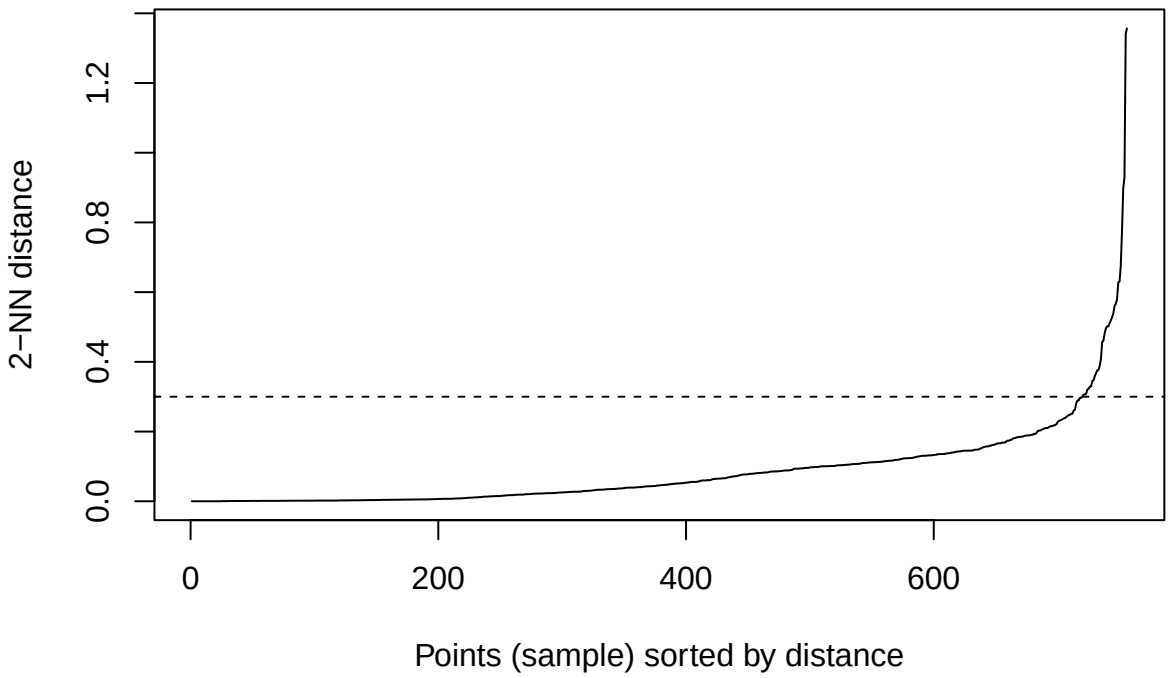
```
  sulfur_content_pct ash_content_pct fuel_cost_per_mmbtu
1         -0.2306101      0.07903011          -0.6846187
2         -0.2401646      0.13762069          -0.6883690
3         -0.6987812     -0.97560030           0.5334835
4          0.9159315      1.89533804          -0.5644211
5          0.1037979      2.05157959          -0.4509741
6          0.7726138      1.87580785          -0.6726176
```

As there might be variables which is unit less,the scaling is done to bring all variables to a common scale,which is very important for machine learning algorithms.It transforms the data so that each variable has a mean of 0 and standard deviation of 1.By standardizing the data,the variables will become comparable and the scale of variable will have no longer influences on the results of certain analyzes.This converts all measurements to the same scale.

DBScan Clustering: #1) Using k=2 in KNNdistplot, determine epsilon.What was your epsilon.

After constructing the knee method, the epsilon value which I take was 0.3. Finding the epsilon value.

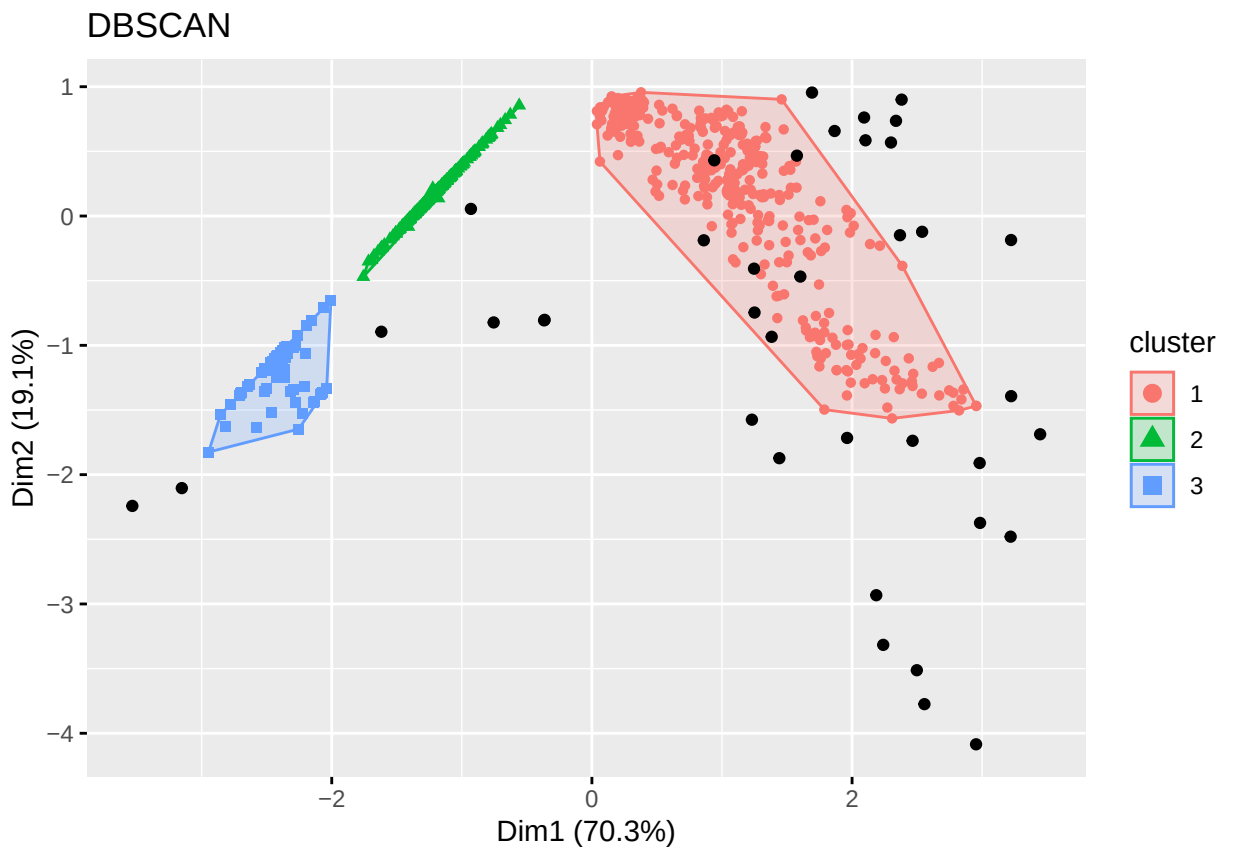
```
dbscan::kNNdistplot(Fuel.norm.df, k=2)
abline(h=0.3, lty=2)
```



#2)How many clusters did you decide?

Clustering the data using the DBSCAN Algorithm.

```
DBscan.fuel <- fpc::dbscan(Fuel.norm.df, eps = 0.3, MinPts = 5)
fviz_cluster(DBscan.fuel, Fuel.norm.df, stand = FALSE, main = "DBSCAN", frame = TRUE,
             geom = "point", ellipse.type = "convex", ellipse.level = 0.3)
```




```
# Cluster Sizes
table(DBscan.fuel$cluster)
```

```
  0   1   2   3
 38 391 263  64
```

DBSCAN doesnot require to specify the number of clusters in the data intially as in the case of kmeans.DBscan is a non-linear density based clustering algorithm that groups the data into dense clusters by measuring the closeness and minimum points and filtering the noise points that lie in the low density region.Density here is given by the minimum points(5) ,the closeness of the points which is given by epsilon(0.3)It does not perform well on large dimensions and sparse data. It shows outliers and handles complicated structures. It can even find a cluster completely surrounded by complete different cluster. Due to the MinPts parameter,DBSCAN has a notion of noise, and is robust to outliers. DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.It is deterministic.

```
# Aggregate means for each variable within each cluster
cluster.data.2 <- cbind(fuel, Cluster = DBscan.fuel$cluster)
cluster.avg.2 <- data.frame(cluster.data.2 %>%   group_by(Cluster) %>%   summarise_all(mean,na.rm=TRUE))
print(round(cluster.avg.2,2))
```

```
  Cluster sulfur_content_pct ash_content_pct fuel_cost_per_mmbtu
1       0              2.41           9.16               4.68
2       1              1.16           8.77               2.11
3       2              0.00           0.00               8.07
4       3              0.13           0.00              19.53
```

The number of cluster are 3

CLUSTER NUMBER 1:COST EFFECTIVE :The cluster number 1 consists of 391 datapoints.It is showing the 1.16 of sulphur content in it with ash content of 8.77 where the fuel cost (mmbtu)is 2.11.As per the domain knowledge,we can say that the the sulphur content has a minimal and decent amount with little high ash prices but the cost of fuel is low.

CLUSTER NUMBER 2:LOW EMISSION:The cluster Number 2 consists of 263 datapoints.It contains the 0.00 of sulphur content and ash content where fuel cost is 8.07.This clusters shows that it is less polluted to environment as there is no carbon emmission present such as sulphur and ash.

CLUSTER NUMBER 3:PURE PREMIUM FUEL:This cluster consists of 64 datapoints where it has the sulphur content of 0.13 with no ash content and 19.53 fuel cost.This cluster consists of no ash content which is less harmful to environment and has a minimal number of sulphur.

#4)What can you say about the relative composition of the different fuels types in relation to your clusters?

Dropping the first two variables.

```
set.seed(2)
fuel.Pattern <- Fuel.receipt[,-c(1,2)]
head(fuel.Pattern)
```

```
  plant_id_eia energy_source_code fuel_type_code_pudl fuel_group_code
1            3               BIT                coal            coal
2            3               BIT                coal            coal
3            3                NG                 gas     natural_gas
4            7               BIT                coal            coal
5            7               BIT                coal            coal
6            7               BIT                coal            coal
     supplier_name fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct
1  interocean coal              259412              23.100               0.49
2  interocean coal               52241              22.800               0.48
3 bay gas pipeline             2783619               1.039               0.00
4     alabama coal               25397              24.610               1.69
5     d & e mining                 764              24.446               0.84
6     alabama coal                 603              24.577               1.54
  ash_content_pct fuel_cost_per_mmbtu
1             5.4               2.135
2             5.7               2.115
3             0.0               8.631
4            14.7               2.776
5            15.5               3.381
6            14.6               2.199
```

Converting the variable 'plant_id_eia' into factor.

```
fuel.Pattern$plant_id_eia <- as.factor(fuel.Pattern$plant_id_eia)
```

DBScan Clustering:- Finding the pattern in the clusters with respect to the numerical variables for those not used in clustering.

```
Cluster.Pattern <-  fuel.Pattern %>% select(c(3)) %>% mutate(Cluster = DBscan.fuel$cluster)
head(Cluster.Pattern)
```

```
  fuel_type_code_pudl Cluster
1                coal       1
2                coal       1
3                 gas       2
4                coal       1
```

```
5                coal       0
6                coal       1
```

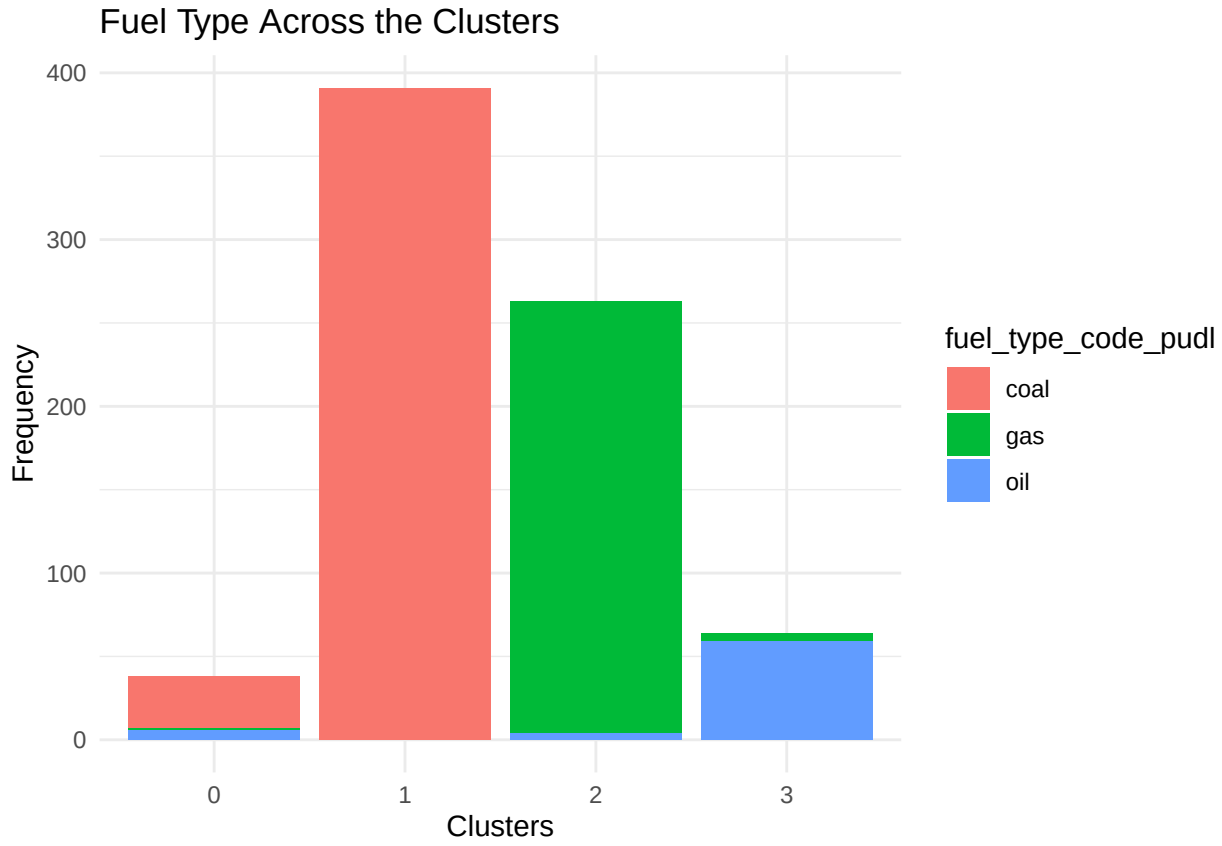
Using categorical variables to find patterns.

```
Categorical.2 <-  DBscan.fuel$cluster
# Create a table with counts of each category within each cluster
categorical.count.3  <- table(Categorical.2, Fuel.receipt$fuel_type_code_pudl)
# Print the table
 categorical.count.3
```

```
Categorical.2 coal gas oil
            0   31   1   6
            1  391   0   0
            2    0 259   4
            3    0   5  59
```

Visualizing the distribution of firms grouped by clusters by using the bar charts.

```
# Bar Chart for Fuel Type
Fuel_Type <- ggplot(Cluster.Pattern, aes(x = factor(Cluster), fill = fuel_type_code_pudl)) +
             geom_bar() + labs(x = 'Clusters', y = 'Frequency',
             title = 'Fuel Type Across the Clusters') +
             theme_minimal()
Fuel_Type
```



According to the graph, we can say that the relative fuel(fuel_type_code_pudl) composition in cluster number 1 consists of 391 datapoints which only have coal present in itwith minimal sulphur() ,ash and cost

price of fuel.Additinally, it has supplier involved for the demand of coal is interocean coal and d & e mining .The cluster 2 covers the vast majority of points with gas which has a count of 259 and has 5 points of gas present with the zero sulphur,ash content and has a price of 8.63.Moreover, it has supplier invovled for gas is bay gas pipeline whereas the cluster number 3 has 59 datapoints of oil and 4 datpoints of gas.