Final Project

Srilaya Valmeekam

2023-05-06

```
library(dplyr)
library(caret)
library(factoextra)
library(leaps)
library(dbscan)
library(esquisse)
```

Data Cleansing:

```
library(readr)
Data <- read.csv("Downloads/fuelcostpudl .csv")

#Using NA to replace missing values

Na<- Data %>% replace(.=="",NA)

#Obtaining the percentages of null values in each column
missing_values<- (colMeans(is.na(Na))*100)

#Removing variables with null values and percentages greater than 50%, as wel
l as a few other variables that do not contribute significantly to the analys
is

Data_1<- subset(Data,select=-c(1:5,7:8,12:14,22:25,26:30))

#Random sampling of 2% data
set.seed(2467)
data_2<-sample_n(Data_1,12000)
```

Data Exploration:

```
#Creating dummy variables to convert fuel_type_code_pudl into numerical data

fuel_type_coal <- ifelse(data_2$fuel_type_code_pudl=="coal" ,1,0)
fuel_type_gas <- ifelse(data_2$fuel_type_code_pudl=="gas" ,1,0)
fuel_type_oil <- ifelse(data_2$fuel_type_code_pudl=="oil" ,1,0)

#Adding these new columns to the existing dataframe

New_data<- cbind(data_2[,-3],fuel_type_coal,fuel_type_gas,fuel_type_oil)
```

Data Preparation:

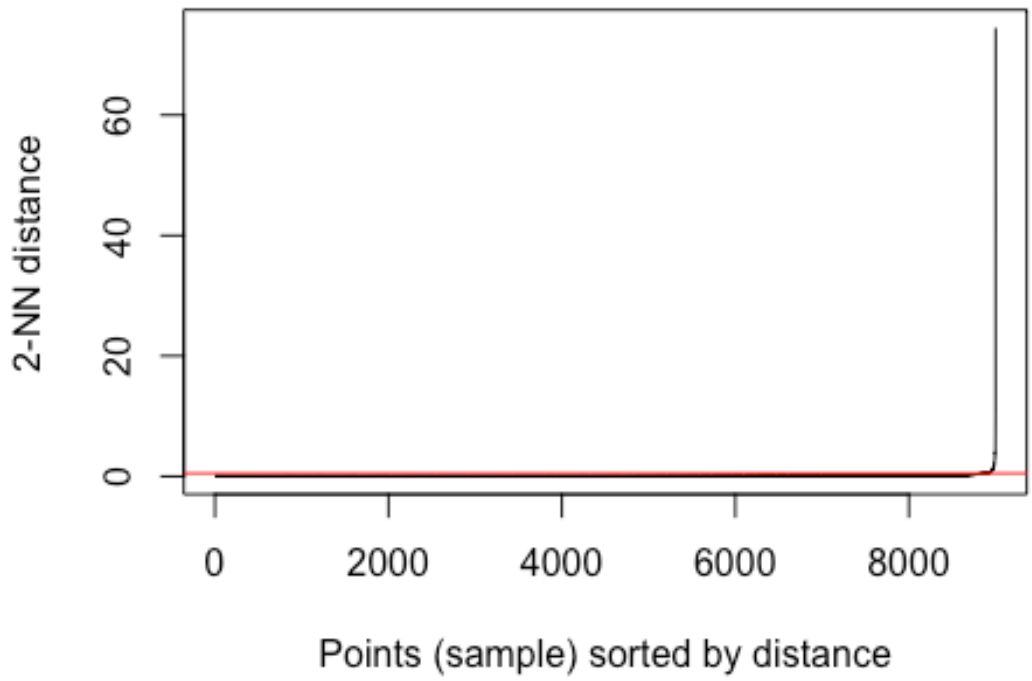
```
#Splitting data into training and test

Split_data<-createDataPartition(New_data$fuel_received_units,p=.75,list=FALSE
)
Training<-New_data[Split_data,]
Test<-New_data[-Split_data,]

Training[is.na(Training)] <- 0
Test[is.na(Test)] <- 0
```

Modeling Strategy:

**The first method was to use the k-means algorithm. However, after observing that the k-means clusters produced were overlapped, it became clear that the data contains border points and outliers.Since the variation between the clusters was too small, I did not prefer to continue my analysis with those clusters.*

The next immediate idea was to perform DBSCAN algorithm as it handles border points and outliers.

```
#Selection of numerical data for clustering

Training_numerical<-Training[,c(4:9,11:13)]

#Normalizing the data
Training_norm<-scale(Training_numerical)

dbscan::kNNdistplot(Training_norm, k =  2)
abline(h = 0.5,col="red")
```

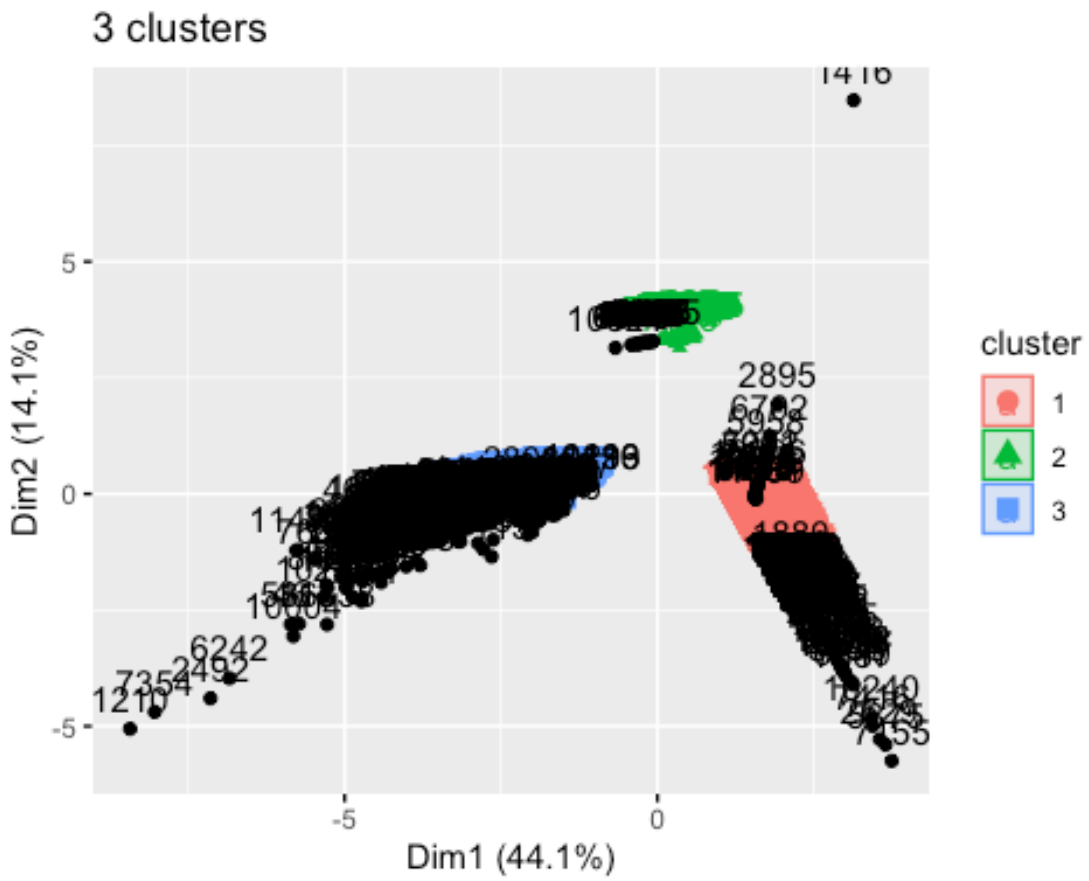


Based on the plot above, set the epsilon value to 0.5. After a few attempts with different values, the option of minPts was made. When minPts was set to 100, I got three perfect clusters with greater variation between them.

```
db <- dbscan::dbscan(Training_norm, eps = 0.5, minPts = 100)
db

## DBSCAN clustering for 9000 objects.
## Parameters: eps = 0.5, minPts = 100
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 3 cluster(s) and 868 noise points.
##
##    0    1    2    3
##  868 4719  741 2672
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

There are 914 boundary points in the data, and three clusters have been formed, with 4732,740, and 2614 data points in each cluster.

```
 #Plotting clusters for better data visualization:
fviz_cluster(db,Training_numerical,main="3 clusters")
```




```
#Assigning clusters to the original data:
assigned_data<-cbind(Training_numerical,db$cluster)


#Finding mean within each cluster to interpret the clusters:
mean_k3 <- Training_numerical %>% mutate(Cluster = db$cluster) %>% group_by(C
luster) %>% summarise_all("mean")
head(mean_k3)
```

```
## # A tibble: 4 × 10
##   Cluster fuel_receive…¹ fuel_…² sulfu…³ ash_c…⁴ mercu…⁵ fuel_…⁶ fuel_…⁷ f
uel_…⁸
##     <int>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
## 1       0        849810.   15.3    1.47    13.3  4.13e-2   18.4    0.771
0.205
## 2       1        281963.    1.03   0        0    0          3.31   0
1
## 3       2          6320.    5.81   0.130    0    0         10.9    0
0
## 4       3         43587.   21.6    1.23     8.17 5.61e-6    1.88   1
0
## # … with 1 more variable: fuel_type_oil <dbl>, and abbreviated variable na
```

```
mes
## #   ¹fuel_received_units, ²fuel_mmbtu_per_unit, ³sulfur_content_pct,
## #   ⁴ash_content_pct, ⁵mercury_content_ppm, ⁶fuel_cost_per_mmbtu,
## #   ⁷fuel_type_coal, ⁸fuel_type_gas
```

Each cluster's interpretation

It can be observed that each of the fuel type falls under each cluster. Therefore, my analysis of each cluster is based on fuel types.

Names of each cluster:

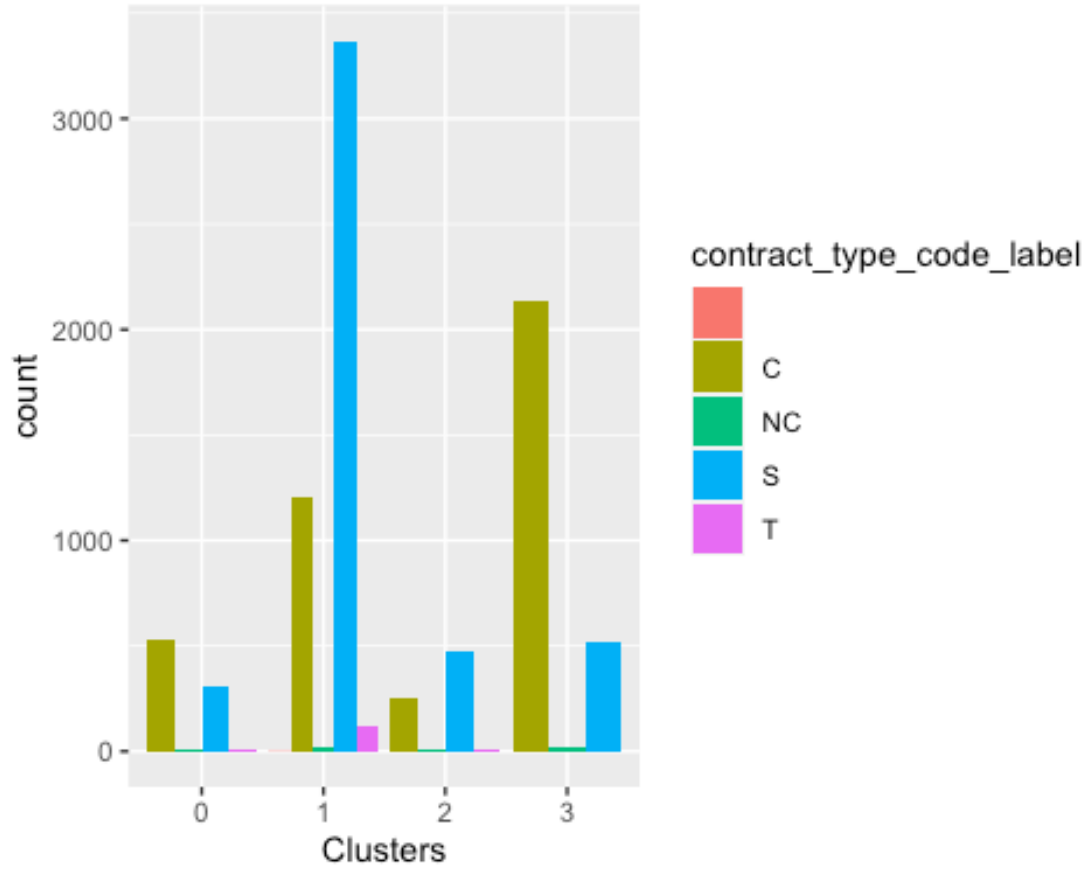
Cluster 1: Gas

Cluster 2: Oil

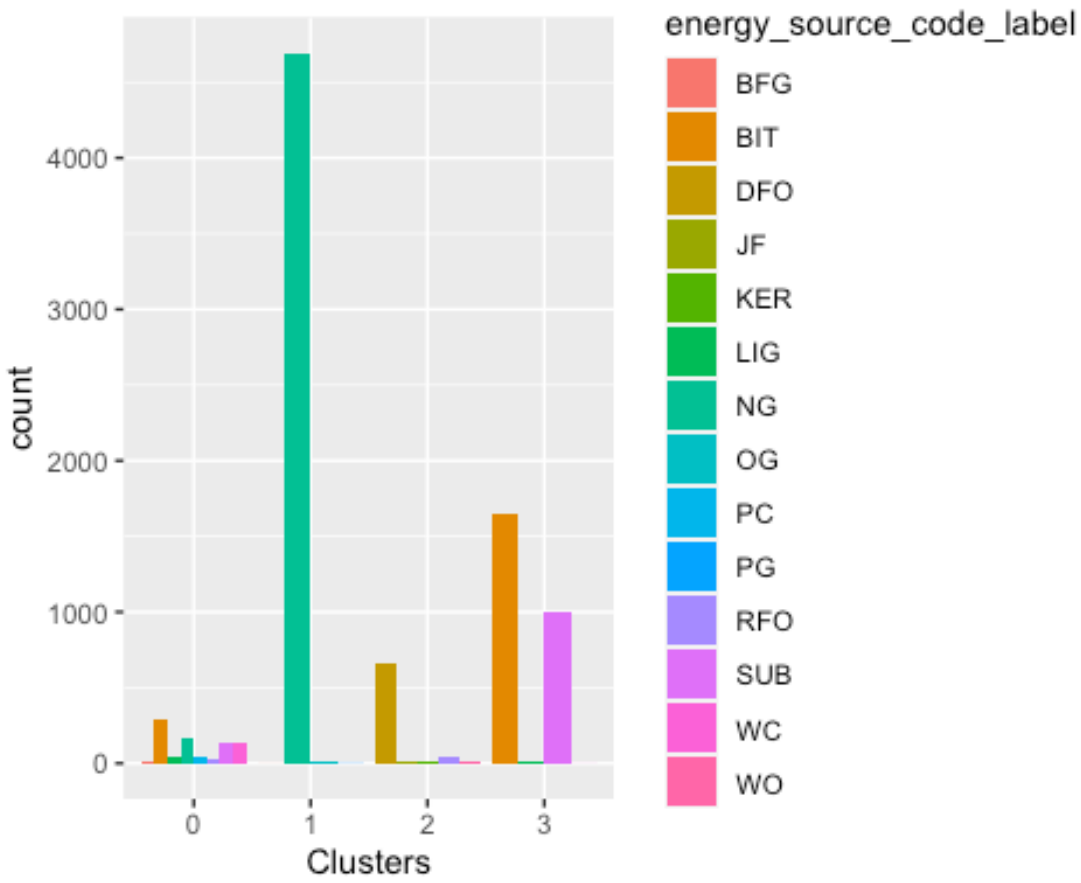
Cluster 3: Coal

Interpreting the pattern in the clusters with respect to the Categorical variables:

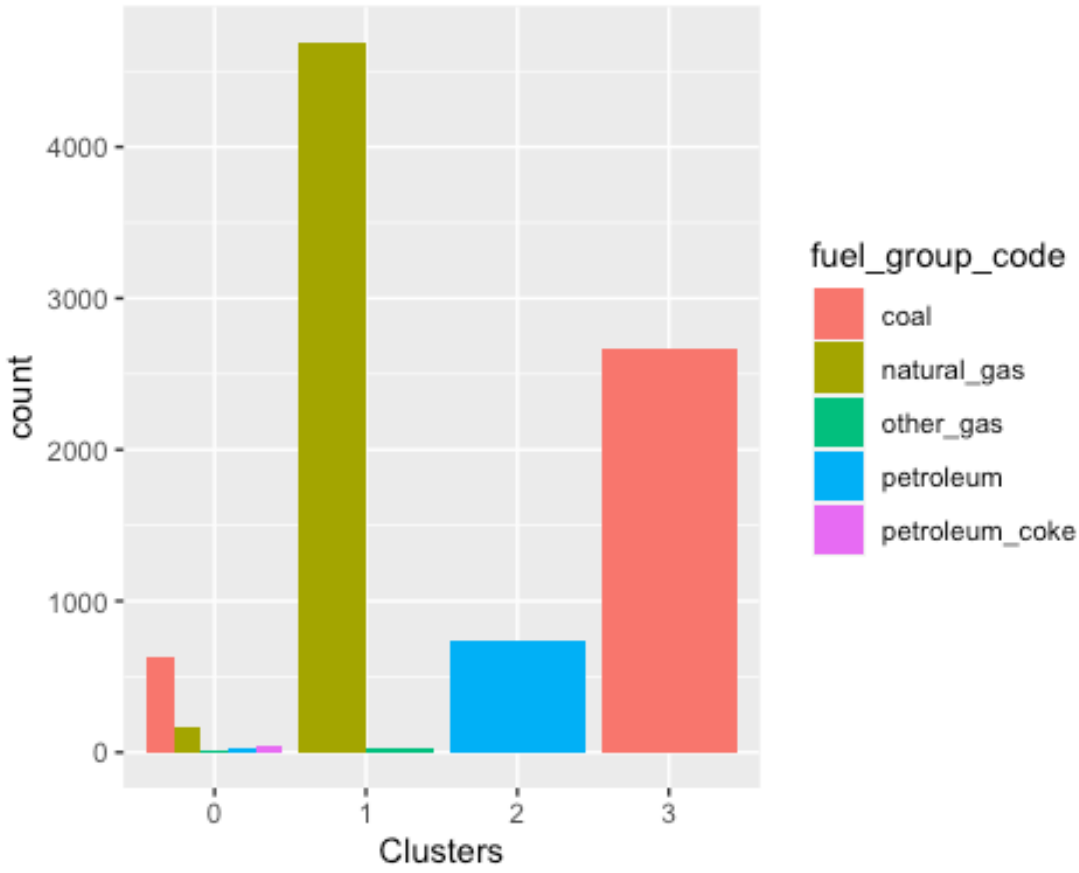
```
plots <- Training[,c(1:3,10)] %>% mutate(Clusters=db$cluster)

ggplot(plots, mapping = aes(factor(Clusters), fill =contract_type_code_label)
)+geom_bar(position='dodge')+labs(x ='Clusters')
```



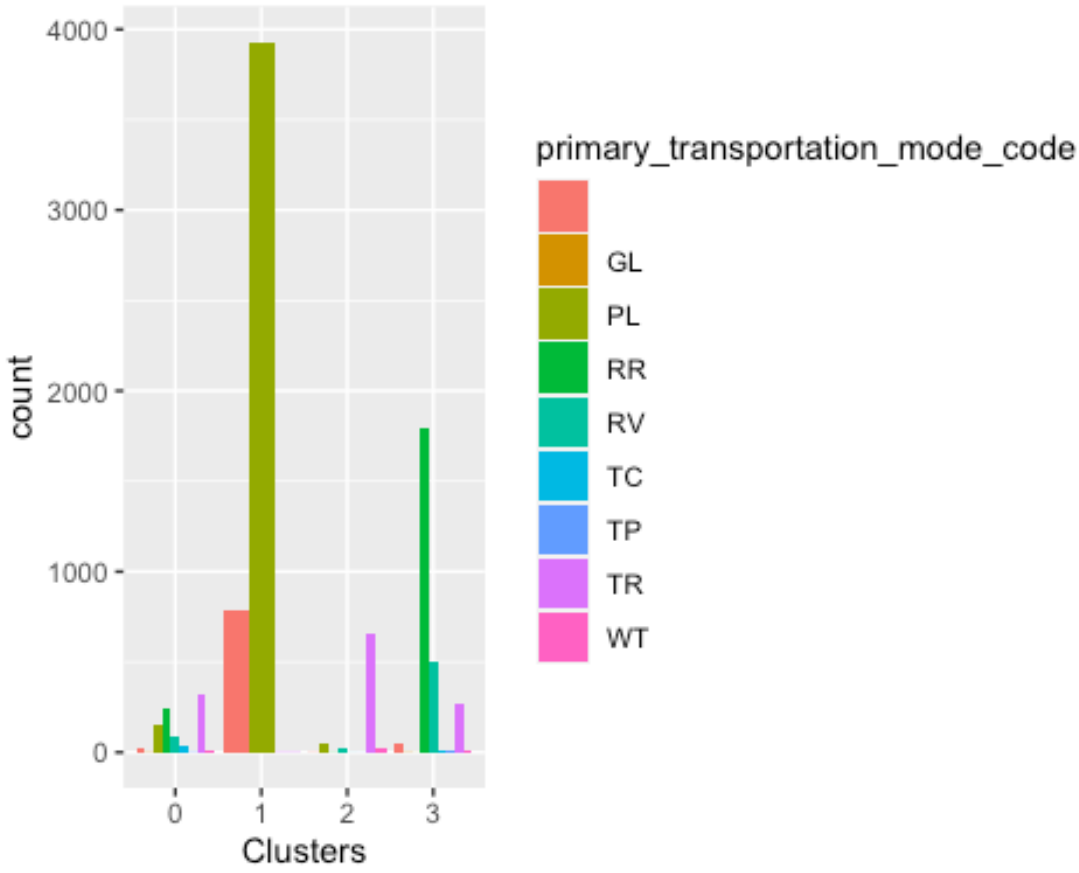
```
ggplot(plots, mapping = aes(factor(Clusters), fill =energy_source_code_label)
)+geom_bar(position='dodge')+labs(x ='Clusters')
```




```
ggplot(plots, mapping = aes(factor(Clusters), fill =fuel_group_code))+geom_ba
r(position='dodge')+labs(x ='Clusters')
```




```
ggplot(plots, mapping = aes(factor(Clusters), fill =primary_transportation_mo
de_code))+geom_bar(position='dodge')+labs(x ='Clusters')
```



Analysis of Cluster 1: Gas

Gas has the average lowest fuel cost per mmbtu. That also explains why is it supplied the most number of avergae units of fuel. Gas does not contain any ash,sulfur and mercury content which makes it a good type of fuel that can be used. It also contains the lowest average of fuel mmbtu per unit. which means that the heat content generated by fuel is less Based on the graph, most gas type fuel is purchased on spot and a relatively lesser amount is purchased on contract. Energy source code is Natural gas and the most commonly used transportation type to supply this type of fuel is through pipelines(PL).

Analysis of Cluster 2: Oil

The average cost of oil per mmbtu is 10.49, making it the most expensive type of fuel in the USA. The average units of oil received in comparision to gas and coal is very less, probably because it is the most expensive fuel type. Even oil doesnot contain ash and mercury percentage but do has a little percent of sulfur content. From the graphs, oil is only purchased on spot. No contract based purchases have been recorded. The energy source code for this type of fuel is DFO which means Distillate Fuel Oil which also includes

Analysis of Cluster 3: Coal *Coal is the least expensive type of fuel and is also widely supplied in the USA. Unlike other two fuels, it contains ash,sulfur and mercury content. The average heat energy received from coal is 21.5612. From the graphs of categorical variables, coal is*

purchased mostly on spot. The energy source code for this type of fuel is BIT and SUB, which indicates that conventional Steam Coal is supplied the most in the U.S.A

Extra-Credit:

```
#Running the multiple linear regression model to determine the best set of va
riables to predict fuel_cost_per_mmbtu by considering variables which were us
ed to form clusters:

Model<- lm(Training_numerical$fuel_cost_per_mmbtu~.,data=Training_numerical)
summary(Model)

##
## Call:
## lm(formula = Training_numerical$fuel_cost_per_mmbtu ~ ., data = Training_n
umerical)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -10.8   -4.2   -1.7    0.5 7125.9
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.002e+01  3.733e+00   2.683  0.00731 **
## fuel_received_units -1.923e-06  1.211e-06  -1.587  0.11254
## fuel_mmbtu_per_unit  1.182e-01  4.270e-01   0.277  0.78197
## sulfur_content_pct  -3.568e-01  1.358e+00  -0.263  0.79275
## ash_content_pct     -8.968e-03  1.879e-01  -0.048  0.96193
## mercury_content_ppm -2.236e+00  2.742e+01  -0.082  0.93499
## fuel_type_coal      -1.014e+01  7.391e+00  -1.372  0.17002
## fuel_type_gas       -3.031e+00  3.683e+00  -0.823  0.41064
## fuel_type_oil               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.92 on 8992 degrees of freedom
## Multiple R-squared:  0.001501,   Adjusted R-squared:  0.0007234
## F-statistic: 1.931 on 7 and 8992 DF,  p-value: 0.06066

#Fuel received units,fuel_type_coal and fuel_type_oil best determine the fuel
_cost_per_mmbtu variable.


#Checking the prediction of the above model on Test data
Test_data<- Test[,c(4:9,11:13)]
Test_Model<-predict(Model, data = Test_data)

#Predicting clusters for Test data:
Test_norm<-scale(Test_data)

Testing_clusters<- predict(db,newdata = Test_norm,data=Training_norm)
```

```
#Appending cluster information and above predicted fuel cost per unit values
to the test data:
Test_predicted_data<- cbind(Test_data,Test_Model,Testing_clusters)
head(Test_predicted_data)
```

```
##   fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_p
ct
## 1                  74              14.147               0.69             6
.5
## 2               12527               1.020               0.00             0
.0
## 3               16867              16.690               0.27             5
.6
## 4               53176              17.718               0.26             5
.0
## 5                  63               1.030               0.00             0
.0
## 6                 650              13.214               1.33            40
.4
##   mercury_content_ppm fuel_cost_per_mmbtu fuel_type_coal fuel_type_gas
## 1                   0               1.475              1             0
## 2                   0               0.000              0             1
## 3                   0               1.092              1             0
## 4                   0               2.501              1             0
## 5                   0               1.454              0             1
## 6                   0               0.000              1             0
##   fuel_type_oil Test_Model Testing_clusters
## 1             0  7.1025234                3
## 2             0  6.6943801                0
## 3             0 10.0007631                3
## 4             0  7.1073848                3
## 5             0  0.6380597                1
## 6             0 10.6768884                0
```

```
#Finding out the averages to see how close the predicted values are to the ac
tual fuel_cost values:
mean_Predicted_Test <- Test_predicted_data %>% mutate(Cluster = Testing_clust
ers) %>% group_by(Cluster) %>% summarise_all("mean")
head(mean_Predicted_Test)
```

```
## # A tibble: 4 × 12
##   Cluster fuel_receive…¹ fuel_…² sulfu…³ ash_c…⁴ mercu…⁵ fuel_…⁶ fuel_…⁷ f
uel_…⁸
##     <int>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
## 1       0        346646.    8.21  0.520     3.91 0.00869    3.12   0.316
0.487
## 2       1        174687.    1.03  0         0    0          4.27   0
1
## 3       2         15418.    5.90  0.0286    0    0          7.65   0
```

```
0
## 4       3         44068.   21.6   1.29      8.50 0          2.49   1
0
## # … with 3 more variables: fuel_type_oil <dbl>, Test_Model <dbl>,
## #   Testing_clusters <dbl>, and abbreviated variable names
## #   ¹fuel_received_units, ²fuel_mmbtu_per_unit, ³sulfur_content_pct,
## #   ⁴ash_content_pct, ⁵mercury_content_ppm, ⁶fuel_cost_per_mmbtu,
## #   ⁷fuel_type_coal, ⁸fuel_type_gas
```

Observations: We could see that the averages of predicted values in each cluster is comparitively closer to the averages of actual fuel cost values. This shows that by choosing variables with significant relationship and cluster information leads to better prediction.