Project Assignment 2
Name: Dandan Wang
NetID: dw862

The goal of this part is to predict the residential energy consumption levels of a city when some attributes of the city are given.

- Preprocess

  Before analysis, we extract the useful attributes for analysis. For residential energy data, we choose 'state_id', 'city', 'housing_units', 'total_pop' (total population), 'elec_mwh' (total residential electricity consumption), 'gas_mcf' (total residential natural gas consumption), 'elec_1kdollars' (the expenditure of electricity), 'gas_1kdollars' (the expenditure of natural gas), 'elec_lb_ghg' (greenhouse gas produced by electricity) and 'elec_lb_ghg' (greenhouse gas produced by natural gas). The state, city attributes are used to locate a unique city. The amount of housings, population are both relevant to the electricity and gas consumptions. Besides, according to the expenditures in electricity (elec_1kdollars) and gas (gas_1kdollars), we can derive the unit-price of electricity and natural gas in different areas. The unit-prices might have impacts on the electricity and natural gas consumptions as well. Because people might prefer to use the cheaper energy. In addition, the elec_lb_ghg and gas_lb_ghg attributes are also kept because some people might prefer to use cleaner energy.

- Clean

  Reanalyze the cleaned data in project 1, I found there are some duplicate values of the same city. The reason might be the data is collected with the unit of city. When a city contains more than one zip code, they just use the data of the whole city to fill the blank. However, some different states might have cities with the same names. Therefore, we should drop duplicated rows (with different zip). After dropping all totally same rows (except zip codes), there are also some cities containing more than one row. The reason might be these statistics data are collected based on an area of a city. We should add different rows with same state and city attributes to get the actual energy data of the whole city. The processed data is stored in "cleaned_residential.csv".

- Basic Statistical Analysis and data cleaning insight.

  The mean of housing_units attribute represents the average amount of housings in a city. The mean of total_pop attribute represents the average population in a city. The mean of elec_1kdollars attribute represents the average electricity expenditure in a city. The mean of elec_mwh attribute represents the average electricity consumption in a city. The mean of gas_1kdollars attribute represents the average natural gas expenditure in a city. The mean of gas_mcf attribute represents the average gas consumption in a city. The mean of elec_lb_ghg attribute represents the average

amount of greenhouse gas produced by electricity in a city. The mean of gas_lb_ghg attribute represents the average amount of greenhouse gas produced by natural gas in a city.

```
The mean of each attribute is:
 housing_units     3.954032e+03
total_pop         1.063390e+04
elec_1kdollars    5.065999e+03
elec_mwh          3.820879e+04
gas_1kdollars     1.695562e+03
gas_mcf           1.686994e+05
elec_lb_ghg       5.701948e+07
gas_lb_ghg        2.025113e+07
```

The median of housing_units attribute represents the median of total housings in a city. The median of total_pop attribute represents the median of population in a city The median of elec_1kdollars attribute represents the median of electricity expenditure in a city. The median of elec_mwh attribute represents the median of electricity consumption i in a city. The median of gas_1kdollars attribute represents the median of natural gas expenditure in a city. The median of gas_mcf attribute represents the median of gas consumption in a city. The meidan of elec_lb_ghg attribute represents the median of total greenhouse gas produced by electricity in a city. The median of gas_lb_ghg attribute represents the median of total greenhouse gas produced by natural gas in a city.

```
The median of each attribute is:
housing_units     551.0
total_pop     1446.0
elec_1kdollars     765.0
elec_mwh     6085.0
gas_1kdollars     114.0
gas_mcf     11627.0
elec_lb_ghg     9133653.0
gas_lb_ghg     1395736.0
```

The standard deviation of ousing_units attribute represents the discrete degree of total housings in a city. The standard deviation of total_pop attribute represents the discrete degree of population. The standard deviation of elec_1kdollars attribute represents the discrete degree of electricity expenditure. The standard deviation of elec_mwh attribute represents the discrete degree of electricity consumptions. The standard deviation of gas_1kdollars attribute represents the discrete degree of natural gas expenditure. The standard deviation of gas_mcf attribute represents the discrete degree of gas consumption. The standard deviation of elec_lb_ghg attribute represents the discrete degree of total greenhouse gas produced by electricity. The standard deviation of gas_lb_ghg attribute represents the discrete degree of total greenhouse gas produced by natural gas.

```
The standard deviation of each attribute is:
 housing_units     3.530588e+04
total_pop         9.560821e+04
elec_1kdollars    4.162461e+04
elec_mwh          2.385554e+05
gas_1kdollars     2.233690e+04
gas_mcf           1.969534e+06
elec_lb_ghg       3.581605e+08
gas_lb_ghg        2.364281e+08
```

- Detect Outliers

  Although these cities have different scales, it is not reasonable for a city to have totally different values from most cities. The types of 'housing_units', 'total_pop', 'elec_1kdollars', 'elec_mwh', 'gas_1kdollars', 'gas_mcf', 'elec_lb_ghg', 'gas_lb_ghg' are all continuous numeric. Therefore, I use a simple binning method to select the outliers firstly. I divide the range of each attribute into 100 even bins and output the count of each bin. Please refer to output.txt to get the detailed output. The distributions of all attributes have a same pattern: most attributes gather at the the 1st to 11th bins. Thus, we regard data in other bins as outliers. And we can sum the counter in outlier bins and calculate the percentage of outliers of each attribute. Please refer to output.txt to get the percentage. The percentage shows that the percentage of each attribute is pretty low. Therefore, I record the upper bound of the non-outliers and drop all rows beyond the upper bound. The processed data by binning method is stored into "binned_residential.csv".

- Missing Values

  I have already cleaned the missing values in Project 1. I replaced the missing values with its neighbors. Because the neighbors in the raw data are also close to each other in reality. Therefore, the scale, energy consumption and expenditure should be similar as well. It is reasonable to use the mean of neighbors to fill the missing value. For the raw data, the fraction of missing values in each column is pretty low.

  housing_units 0.0148%
  total_pop 0.0162%
  elec_mwh 0.0148%
  gas_mcf 0.0148%
  elec_1kdollars 0.0148%
  elec_1kdollars 0.0148%
  gas_lb_ghg 0.0148%
  elec_lb_ghg 0.0148%

  The number of rows in the raw data is 67732. There are only tens of rows containing missing values. Of course, deleting these rows directly is also ok. They have few impacts on the final results. Besides, I also design a function named "findMissing" to detect the fraction of missing values in the data frame. The output shows there is no missing value in the data frame.

```
the fraction of missing value is:  city             0.0
housing_units     0.0
total_pop         0.0
elec_1kdollars    0.0
elec_mwh          0.0
gas_1kdollars     0.0
gas_mcf           0.0
elec_lb_ghg       0.0
gas_lb_ghg        0.0
state_id          0.0
```

- Bin Data

  I bin two attributes – 'elec_mwh' and 'gas_mcf'. The reason for binning them is intuitive. Because I want to classify the data based on the electricity consumptions and the natural gas consumptions. And binning them into five even bins also gets the degree of energy consumptions. For example, the city whose elec_degree is 1 has very low electricity consumption level. There are 5 degrees for electricity and natural gas consumptions individually.

degree	representation
1	Very low
2	Low
3	Medium
4	High
5	Very high

  According to the degree, we can know the type of one city in residential energy consumption. For example, Suffield is a very low electricity consumption and a very low natural gas consumption city. The new data is stored in "degreed_residential.csv".

- Additional Part for CS students

  I use the 3 different k [100, 500, 1000] and bin the scores into 10 bins. For different k, the major bin is always the 10^{th} bin. However, the count of data located in the 10^{th} bin changes a little. The 10^{th} bin contains maximal negative outlier factor whose absolute value is smallest. Therefore, data in 10^{th} bin are normal data not outliers. We should abandon data located in other bins. The processed data is stored in "lof_residential.csv". The fraction of outliers is really low. I drop all possible outliers based on the three outlier factor attributes.

```
The k of LOF is  100

Bin count of  lof_score_100  is  [    0     2     0     0     0     0     3     1     7    55 24398]
The k of LOF is  500

Bin count of  lof_score_500  is  [    0     2     0     0     0     2     1     1     5    23 24432]
The k of LOF is  1000

Bin count of  lof_score_1000  is  [    0     2     0     0     0     2     2     0     6    24 24430]
```

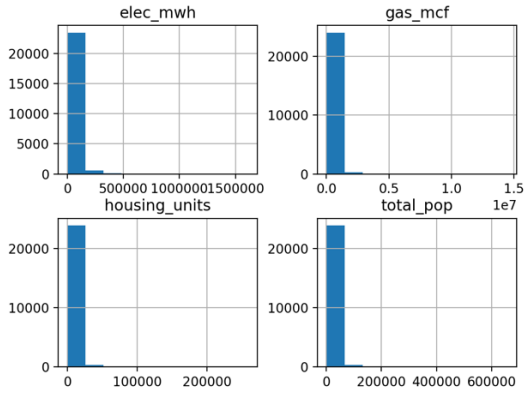
  Compared above binning method with LOF, I bin two class labels into 5 bins and print the counts of all bins. The difference between counts of all bins is really small. Therefore, I use binned data to processe following tasks. In addition, binning method is also necessary to implement before LOF to determine the value of parameter –

contamination. I can estimate this based on the percentage of outliers in binning method.

```
For LOF algorithm:

Bin count of  elec_mwh  is  [    0 24444    14     4     2     2]

Bin count of  gas_mcf  is  [    0 24350    63    22    17     6]
For binning method:

Bin count of  elec_mwh  is  [    0 24430    23     2     2     1]

Bin count of  gas_mcf  is  [    0 24350    63    22    17     6]
```

- Histograms and Correlations

I plot the histograms containing 4 variables – 'housing_units', 'total_pop', ' 'elec_mwh', 'gas_mcf'. These variables all have similar distribution and are all skewed right. The reason is that 'housing_units' and 'total_pop' are positive correlated with 'elec_mwh' and 'gas_mcf'.

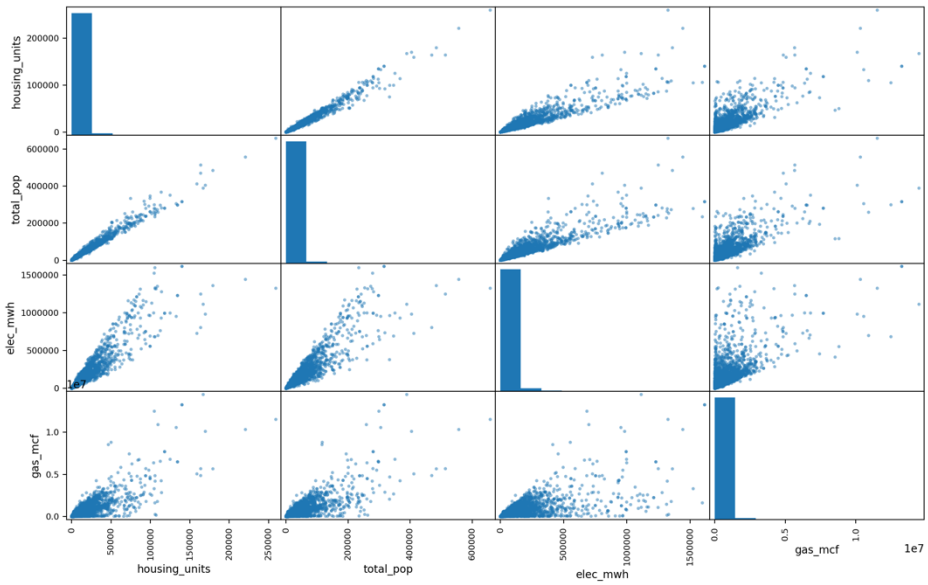


Here is the output table of correlations between the 4 variables.

	housing_units	total_pop	elec_mwh	gas_mcf
housing_units	1.000000	0.990381	0.941415	0.849373
total_pop	0.990381	1.000000	0.930807	0.828372
elec_mwh	0.941415	0.930807	1.000000	0.728769
gas_mcf	0.849373	0.828372	0.728769	1.000000

This table shows that 'housing_units' and 'total_pop' are both strong positive correlated to 'elec_mwh' and 'gas_mcf'. This means the amount of housing and population are strong positive correlated to energy consumptions. Besides, the correlation between housing and population is also strong positive. This is obvious. However, the correlation between 'elec_mwh' and 'gas_mcf' is not as strong as others. This is reasonable that a city whose electricity consumption is high level may not be a city whose natural gas consumption is also high level.

I also plot a set of scatterplot subplots. It also shows the same patter as the correlation matrix.

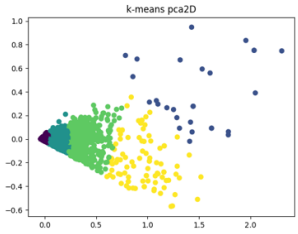


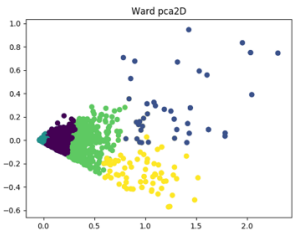
- Cluster Analysis

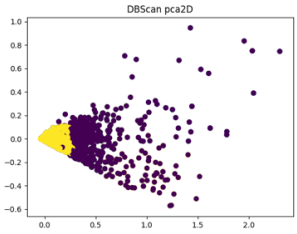
  Here is the clustering result. According to silhouette score, DBSCAN is closer to 1. It performs best. While k-means performs best according to calinski harabasz score.

	silhouette_score	calinski harabasz score
k-means	0.7971056135231366	32574.861914920908.
Ward	0.7881392853479291	29614.521769706418.
dbscan	0.9104714927428617	28221.121175357628.

  Here is the PCA plots. Observing these PCA plots, the clustering results of k-means and ward algorithm are similar except the yellow and dark blue points. The majority of data set is tight. The number of clusters becomes fewer by using dbscan algorithm. The plots show that k-means is more suitable for this data set.







- Association Rules

  I selected 3 columns – 'state_id', 'elec_degree', 'gas_degree' to form the subset. Before calling Apriori algorithm to calculate support, some transformation is necessary. Because the degree of electricity and gas are both 1 to 5. If use the raw subset to calculate, the algorithm would regard two attributes same. Therefore, I transform the two attributes into string type. For example,

	state_id	elec_degree	gas_degree
0	MA	elec_degree_very_low	gas_degree_very_low
1	MA	elec_degree_very_low	gas_degree_very_low
2	MA	elec_degree_very_low	gas_degree_very_low
3	MA	elec_degree_very_low	gas_degree_very_low
4	MA	elec_degree_low	gas_degree_low
5	MA	elec_degree_very_low	gas_degree_very_low
6	MA	elec_degree_very_low	gas_degree_very_low
7	MA	elec_degree_very_low	gas_degree_very_low
8	MA	elec_degree_very_low	gas_degree_very_low
9	MA	elec_degree_very_low	gas_degree_very_low

I use three support levels – 0.6, 0.9, 0.1. When support = 0.6, the output is:

	support	itemsets
0	0.986928	(elec_degree_very_low)
1	0.996640	(gas_degree_very_low)
2	0.986518	(elec_degree_very_low, gas_degree_very_low)

There are only three rules. We can conclude that the degrees of energy consumptions (both electricity and natural gas) of most cities are very low level. Besides, cities who consumed little electricity also consume little natural gas.

When support = 0.1, the output is:

	support	itemsets
0	0.106995	(PA)
1	0.986928	(elec_degree_very_low)
2	0.996640	(gas_degree_very_low)
3	0.106913	(elec_degree_very_low, PA)
4	0.106913	(PA, gas_degree_very_low)
5	0.986518	(elec_degree_very_low, gas_degree_very_low)
6	0.106872	(elec_degree_very_low, PA, gas_degree_very_low)

In addition to above three rules, there are three more rules. We can conclude that most cities in Pennsylvania state use very little energy resources (including both electricity and natural gas).

When support = 0.05, the output is:

	support	itemsets
0	0.054583	(IL)
1	0.106995	(PA)
2	0.986928	(elec_degree_very_low)
3	0.996640	(gas_degree_very_low)
4	0.054215	(elec_degree_very_low, IL)
5	0.054215	(gas_degree_very_low, IL)
6	0.106913	(elec_degree_very_low, PA)
7	0.106913	(gas_degree_very_low, PA)
8	0.986518	(elec_degree_very_low, gas_degree_very_low)
9	0.054133	(elec_degree_very_low, gas_degree_very_low, IL)
10	0.106872	(elec_degree_very_low, gas_degree_very_low, PA)

In addition to Pennsylvania state, Illinois state is also a very low energy consumption state.

The most frequent pattern is 'gas_degree_very_low'. This means that the natural gas consumptions of most cities are very low level. It is reasonable.

- Predictive Analysis

t-test

Firstly, I have a hypothesis – the difference between the means of 'elec_degree' and 'gas_degree' attributes is not obvious. Here is the output of t-test:

```
(10.644534004750708, 1.9768429270561316e-26)
False
```

The first line is the return value of t-test statistical containing t-statistics and p-value. The second line is the return value of a condition statement (p-value > 0.05?). It is False. The original hypothesis fails. Therefore, the difference between degrees of electricity and natural gas is obvious. It represents the distribution of electricity degrees is different from the distribution of natural gas degrees.

Linear Regression

I have another hypothesis – the 'housing_units' and 'total_pop' are positive related with 'elec_degree'. The fit coefficient is 0.6334307274621975. The hypothesis is true.

To predict the class of data, I drop the 'state_id', 'city', 'elec_mwh', 'gas_mcf' attributes and select 'elec_degree' and 'gas_degree' as the class columns. The model becomes to predict the energy consumption levels of a city given some attributes including 'housing_units', 'total_pop', 'elec_1kdollar', 'gas_1kdollar', 'elec_lb_ghg' and 'gas_lb_ghg'. Here are the evaluation and prediction results with different method.

Evaluation of machine learning methods (electricity degree):

	Average mean of accuracy	Average standard deviation of accuracy
Decision Tree	0.996005	0.001801
KNN	0.994109	0.001655
Naïve Bayes	0.287635	0.083137
SVM	0.993853	0.001554
Random Forest	0.996722	0.001452

Prediction (electricity degree):

	Accuracy Score
Decision Tree	0.9967219832001639
KNN	0.9950829748002459
Naïve Bayes	0.32780167998360993
SVM	0.9952878508502356
Random Forest	0.998360991600082

Evaluation of machine learning methods (natural gas degree):

	Average mean of accuracy	Average standard deviation of accuracy
Decision Tree	0.999846	0.000328
KNN	0.998873	0.000820
Naïve Bayes	0.757095	0.007104

SVM	0.993853	0.001554
Random Forest	0.999539	0.000426

Prediction (natural gas degree):

	Accuracy Score
Decision Tree	0.999180495800041
KNN	0.998360991600082
Naïve Bayes	0.7510755992624463
SVM	0.9981561155500922
Random Forest	0.999180495800041

According to above accuracies, all methods except Naïve Bayes perform very well. And the prediction accuracy could be as high as 99.9%. This also shows that the chosen attributes are strong related with the class labels.