professor project notebook

March 25, 2018

```
In [10]: import numpy as np
         import matplotlib.pyplot as plt
         import matplotlib.dates as dates
         import pandas as pd
         from sklearn.cluster import KMeans
         from sklearn.model_selection import GridSearchCV
         import sys
         import collections
         import itertools
         import numpy as np
         import matplotlib.pyplot as plt
         from scipy.stats import mode
         from scipy.spatial.distance import squareform
         from numpy import shape
         import random
         from sklearn import metrics
         from sklearn.neighbors import LocalOutlierFactor
         import os
         import glob
```

```
In [11]: path = os.getcwd()
         files = os.listdir(path)

         files1 = files[0:8] + files[10:14]
         print files1
         df = pd.DataFrame()

         for i in range(len(files1)):
             frame  = pd.read_excel(files1[i])
             df = df.append(frame)
         #df = df.append(pd.read_excel(files1))
         print df.head(10)
```

```
['Load Data_12191216 4359239.xls', 'Load Data_12191230 2980063.xls', 'Load Data_12191206 3093584
  Active Energy(-)T1(kWh) Active Energy(-)T2(kWh) Active energy(+)T1H(kWh)  \
0                    ----                    ----             2919300.0000
1                    ----                    ----             2919252.0000
```

```
2                    ----                    ----             2919204.0000
3                    ----                    ----             2919048.0000
4                    ----                    ----             2918772.0000
5                    ----                    ----             2918508.0000
6                    ----                    ----             2918244.0000
7                    ----                    ----             2917968.0000
8                    ----                    ----             2917848.0000
9                    ----                    ----             2917812.0000

  Active energy(+)T2L(kWh) Active energy(-)(kWh)     CT   Customer Address  \
0             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
1             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
2             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
3             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
4             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
5             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
6             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
7             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
8             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   
9             3674376.0000               36.0000  400/1  PLT @ AWASI-BODER   

           Customer Name  Customer No.             Date  \
0  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 12:00   
1  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 11:45   
2  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 11:30   
3  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 11:15   
4  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 11:00   
5  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 10:45   
6  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 10:30   
7  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 10:15   
8  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 10:00   
9  LTD PRIME STEEL MILLS     4359239.0  2017-12-19 09:45   

           ...           Unnamed: 15 Unnamed: 2 Unnamed: 3 Unnamed: 4  \
0          ...                   NaN        NaN        NaN        NaN   
1          ...                   NaN        NaN        NaN        NaN   
2          ...                   NaN        NaN        NaN        NaN   
3          ...                   NaN        NaN        NaN        NaN   
4          ...                   NaN        NaN        NaN        NaN   
5          ...                   NaN        NaN        NaN        NaN   
6          ...                   NaN        NaN        NaN        NaN   
7          ...                   NaN        NaN        NaN        NaN   
8          ...                   NaN        NaN        NaN        NaN   
9          ...                   NaN        NaN        NaN        NaN   

  Unnamed: 5 Unnamed: 6 Unnamed: 7 Unnamed: 8 Unnamed: 9 active energy(+)(kWh)
0        NaN        NaN        NaN        NaN        NaN          6593676.0000
1        NaN        NaN        NaN        NaN        NaN          6593628.0000
```

```
2        NaN        NaN        NaN        NaN        NaN          6593580.0000
3        NaN        NaN        NaN        NaN        NaN          6593424.0000
4        NaN        NaN        NaN        NaN        NaN          6593148.0000
5        NaN        NaN        NaN        NaN        NaN          6592884.0000
6        NaN        NaN        NaN        NaN        NaN          6592620.0000
7        NaN        NaN        NaN        NaN        NaN          6592344.0000
8        NaN        NaN        NaN        NaN        NaN          6592224.0000
9        NaN        NaN        NaN        NaN        NaN          6592188.0000

[10 rows x 32 columns]
```

```
In [12]: newdataset = df.drop(['Active Energy(-)T1(kWh)', 'Active Energy(-)T2(kWh)','Active ener
                             ,'Active energy(+)T2L(kWh)','Active energy(-)(kWh)'],axis = 1)
```

```
In [13]: dataset = newdataset.drop(['Meter No.','Reactive energy(-)(kvarh)','Transformer'],axis
         newdataset2 = dataset.drop(['Unnamed: 3','Unnamed: 4','Unnamed: 5','Unnamed: 6','Unname
                                    'Unnamed: 15','Unnamed: 2','Unnamed: 9',
                                     'Unnamed: 1','Unnamed: 10','Unnamed: 11','Unnamed: 12','Unn
                                     ,'Unnamed: 0','CT', 'Customer Address',
                                     'Customer Name','Reactive energy(+)(kvarh)','PT'],axis = 1)
```

```
In [14]: print newdataset2.head(10)
```

```
   Customer No.               Date active energy(+)(kWh)
0     4359239.0  2017-12-19 12:00           6593676.0000
1     4359239.0  2017-12-19 11:45           6593628.0000
2     4359239.0  2017-12-19 11:30           6593580.0000
3     4359239.0  2017-12-19 11:15           6593424.0000
4     4359239.0  2017-12-19 11:00           6593148.0000
5     4359239.0  2017-12-19 10:45           6592884.0000
6     4359239.0  2017-12-19 10:30           6592620.0000
7     4359239.0  2017-12-19 10:15           6592344.0000
8     4359239.0  2017-12-19 10:00           6592224.0000
9     4359239.0  2017-12-19 09:45           6592188.0000
```

```
In [15]: newdataset2 = newdataset2.replace('----', np.nan)
         newdataset2 = newdataset2.replace('###############', np.nan)
         data1 = newdataset2.dropna(axis = 0)
         data2 = data1.drop_duplicates(subset=['active energy(+)(kWh)','Date'],keep=False)
         print data2.head(10)
```

```
   Customer No.               Date active energy(+)(kWh)
0     4359239.0  2017-12-19 12:00           6593676.0000
1     4359239.0  2017-12-19 11:45           6593628.0000
2     4359239.0  2017-12-19 11:30           6593580.0000
3     4359239.0  2017-12-19 11:15           6593424.0000
4     4359239.0  2017-12-19 11:00           6593148.0000
```

```
5     4359239.0  2017-12-19 10:45          6592884.0000
6     4359239.0  2017-12-19 10:30          6592620.0000
7     4359239.0  2017-12-19 10:15          6592344.0000
8     4359239.0  2017-12-19 10:00          6592224.0000
9     4359239.0  2017-12-19 09:45          6592188.0000
```

```
In [16]: DATE = pd.to_datetime(data2['Date'])
         SRN = pd.Series(data2['Customer No.'])
         data = data2.set_index([SRN,DATE])
         data.head(5)
```

```
Out[16]:                                    Customer No.              Date  \
         Customer No. Date
         4359239.0    2017-12-19 12:00:00     4359239.0  2017-12-19 12:00
                      2017-12-19 11:45:00     4359239.0  2017-12-19 11:45
                      2017-12-19 11:30:00     4359239.0  2017-12-19 11:30
                      2017-12-19 11:15:00     4359239.0  2017-12-19 11:15
                      2017-12-19 11:00:00     4359239.0  2017-12-19 11:00

                                   active energy(+)(kWh)
         Customer No. Date
         4359239.0    2017-12-19 12:00:00          6593676.0000
                      2017-12-19 11:45:00          6593628.0000
                      2017-12-19 11:30:00          6593580.0000
                      2017-12-19 11:15:00          6593424.0000
                      2017-12-19 11:00:00          6593148.0000
```

```
In [17]: import datetime as dt

         total_user_data = pd.DataFrame()

         for srn, DATE in data.groupby(level=0):
             # right now the input is the only for one users with different date
             Input = DATE

             Input2 = DATE.get_values()
             time = pd.DatetimeIndex(Input['Date'])
             timediff = {}
             eng_diff = {}
             for i in range(len(Input)):
                 if i ==0:
                     timediff[0] = np.nan
                     eng_diff[0] = np.nan
                 else:
                     timediff[i] = time[i] - time[i - 1]
                     eng_diff[i] = float(Input['active energy(+)(kWh)'][i]) - float(Input['activ
             diff_eng = pd.Series(eng_diff)
             diff_time =  pd.Series(timediff)
             # cal the time gap as hours
             diff_time = diff_time.dt.seconds/(60 * 60) - 24
             power = {}
             power = diff_eng/diff_time
             power1 = pd.Series(power)
             Input['power'] = power1.values
             # now we output our power and add it to our input
             user_data = Input.drop(['Customer No.','Date'],axis = 1)
             user_data = user_data.dropna()
             total_user_data = total_user_data.append(user_data)
         # add all user data to total user data
         print total_user_data.head(10)
         # finish the data prune
```

```
/home/tzhang/anaconda2/lib/python2.7/site-packages/ipykernel_launcher.py:28: SettingWithCopyWarn
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
```

```
                                 active energy(+)(kWh)  power
Customer No. Date
531109.0     2016-08-19 12:00:00           278504.0000  836.0
             2016-08-19 11:45:00           278299.0000  820.0
             2016-08-19 11:30:00           278093.0000  824.0
             2016-08-19 11:15:00           277885.0000  832.0
             2016-08-19 11:00:00           277681.0000  816.0
             2016-08-19 10:45:00           277473.0000  832.0
             2016-08-19 10:30:00           277264.0000  836.0
             2016-08-19 10:15:00           277054.0000  840.0
             2016-08-19 10:00:00           276843.0000  844.0
             2016-08-19 09:45:00           276633.0000  840.0
```

```
In [87]: from sklearn.ensemble import IsolationForest

         total_prediction = pd.Series()


         for srn, DATE in total_user_data.groupby(level = 0):
             Input = DATE['power']


             fit_data = []
             # use the whole training dataset
```

```
            train_data1 = Input.value_counts()
            train_data_idx = np.array(train_data1.index).reshape(-1,1)
            # the fitting dataset we use the top 20% common values in the training dataset as f
            fit_data_idx = train_data_idx[:len(train_data_idx)/5]
            train_data = Input.values.reshape(-1,1)
            for i in train_data:
                for j in fit_data_idx:
                    if i == j:
                        fit_data.append(i)


            ii =IsolationForest(contamination = 0.01).fit(fit_data).predict(train_data)
            pre1 = pd.Series(ii)

            total_prediction = total_prediction.append(pre1, ignore_index = True)

            inner = []
            outer = []
            error_rate_total = []
            outer_number = []
            for k in range(0,len(ii)):
                if ii[k] ==1:
                    inner.append(ii[k])
                else: outer.append(ii[k])
            error_rate_total = np.append(error_rate_total,float(len(outer))/(len(outer)+len(inn
            outer_number = np.append(outer_number,len(outer))
            normal_case = 1 - error_rate_total

            print 'srn_number %d' %srn
            print 'outlier_rate %f' %error_rate_total
            print 'outlier number %d'%outer_number
            print 'normal_case_rate %f' %normal_case
```

```
srn_number 531109
outlier_rate 0.032995
outlier number 39
normal_case_rate 0.967005
srn_number 2097996
outlier_rate 0.015636
outlier number 624
normal_case_rate 0.984364
srn_number 2113241
outlier_rate 0.015394
outlier number 697
normal_case_rate 0.984606
srn_number 2118864
outlier_rate 0.012632
outlier number 604
```

```
normal_case_rate 0.987368
srn_number 2272639
outlier_rate 0.010176
outlier number 172
normal_case_rate 0.989824
srn_number 2851433
outlier_rate 0.096900
outlier number 4301
normal_case_rate 0.903100
srn_number 2980063
outlier_rate 0.058428
outlier number 2061
normal_case_rate 0.941572
srn_number 2993154
outlier_rate 0.015750
outlier number 51
normal_case_rate 0.984250
srn_number 3093584
outlier_rate 0.071839
outlier number 3791
normal_case_rate 0.928161
srn_number 3819314
outlier_rate 0.167105
outlier number 6866
normal_case_rate 0.832895
srn_number 4359239
outlier_rate 0.080861
outlier number 2982
normal_case_rate 0.919139
srn_number 4814120
outlier_rate 0.193594
outlier number 10058
normal_case_rate 0.806406
```

```
In [78]: total_user_data['pre'] = total_prediction.values
         print total_user_data.head(10)
```

```
                                 active energy(+)(kWh)  power  pre
Customer No. Date
531109.0     2016-08-19 12:00:00           278504.0000  836.0    1
             2016-08-19 11:45:00           278299.0000  820.0    1
             2016-08-19 11:30:00           278093.0000  824.0    1
             2016-08-19 11:15:00           277885.0000  832.0    1
             2016-08-19 11:00:00           277681.0000  816.0    1
             2016-08-19 10:45:00           277473.0000  832.0    1
             2016-08-19 10:30:00           277264.0000  836.0    1
             2016-08-19 10:15:00           277054.0000  840.0    1
             2016-08-19 10:00:00           276843.0000  844.0    1
             2016-08-19 09:45:00           276633.0000  840.0    1
```

```
In [79]: abnormal_value = total_user_data[total_user_data.pre == -1]
         print abnormal_value.head(10)
```

```
                                 active energy(+)(kWh)  power  pre
Customer No. Date
531109.0     2016-08-17 23:30:00           250269.0000   -0.0   -1
             2016-08-17 23:15:00           250269.0000   -0.0   -1
             2016-08-17 23:00:00           250269.0000   -0.0   -1
             2016-08-17 22:45:00           250269.0000   -0.0   -1
             2016-08-17 22:30:00           250269.0000   -0.0   -1
             2016-08-17 22:15:00           250269.0000   -0.0   -1
             2016-08-17 22:00:00           250269.0000   -0.0   -1
             2016-08-17 21:45:00           250269.0000   -0.0   -1
             2016-08-17 21:30:00           250269.0000   -0.0   -1
             2016-08-17 21:15:00           250269.0000   -0.0   -1
```

```
In [80]: # save file
         abnormal_value.to_csv('result.csv')
```