Test_Pcorrelation

February 7, 2020

An Introduction to the Pcorrelation program

I. Introduction

Correlation is one of the most popular methods to detect the linear relation between two features. One might have difficulty to find a strong correlation because of high data dimensionality. To this end, I developed a program, namely: Pcorrelation. Several vital outcomes are produced by using this program, such as

1. seek a number of attributes that have a correlation of over specific threshold.
2. List names of attributes that have a correlation of over specific threshold.
3. Show pair of attributes that have a correlation of over specific threshold.
4. Visualize the correlation matrix

II. A simple illustration

This program is particularly useful for the data with very high dimensionality. A dataset with a small number of features might not be needed. However, for simplicity, I use the data on candy competition (12 features). The dataset is available at the two links below

https://github.com/phuongvnguyen/Pfeature-Selection/blob/master/candy-data.csv

https://github.com/fivethirtyeight/data/tree/master/candy-power-ranking

Thus, the program below is an excellent example of showing the usefulness of my Pcorrelation program.

1 Checking the working directory

[1]:
```
import os
os.getcwd()
```

[1]:
```
'/Users/phuong/Dropbox/Machine Learning/My Python Functions'
```

[2]:
```
#os.chdir("C:\\Users\\Phuong_1\Dropbox\Machine␣
 ↪Learning\\Lidl\\candy-power-ranking") # Window format
os.chdir("/Users/phuong/Dropbox/Machine Learning/Lidl/candy-power-ranking") #␣
 ↪Mac iOS format
#print(Bold + Blue + 'Your current working directory is:' + End)
print(os.getcwd())
```

```
/Users/phuong/Dropbox/Machine Learning/Lidl/candy-power-ranking
```

2 Loading the dataset

[3]:
```
import pandas as pd
candy_data=pd.read_csv('candy-data.csv')
print(candy_data.head(3))
print(candy_data.shape)
```

```
  competitorname  chocolate  fruity  caramel  peanutyalmondy  nougat  \
0      100 Grand          1       0        1               0       0
1   3 Musketeers          1       0        0               0       1
2       One dime          0       0        0               0       0

   crispedricewafer  hard  bar  pluribus  sugarpercent  pricepercent  \
0                 1     0    1         0         0.732         0.860
1                 0     0    1         0         0.604         0.511
2                 0     0    0         0         0.011         0.116

   winpercent
0   66.971725
1   67.602936
2   32.261086
(85, 13)
```

[4]:
```
os.chdir("/Users/phuong/Dropbox/Machine Learning/My Python Functions")
```

[5]:
```
print(os.getcwd())
```

```
/Users/phuong/Dropbox/Machine Learning/My Python Functions
```

3 Calling the program

Please make sure, the python file "Pcorrelation" is located in your current working directory. Otherwise, the codes below can not be implemented.

[6]:
```
from Pcorrelation_Alone import Pcorrelation
```

[7]:
```
# check
Pcorrelation
```

[7]: <function Pcorrelation_Alone.Pcorrelation(data, threshold, figsizes)>

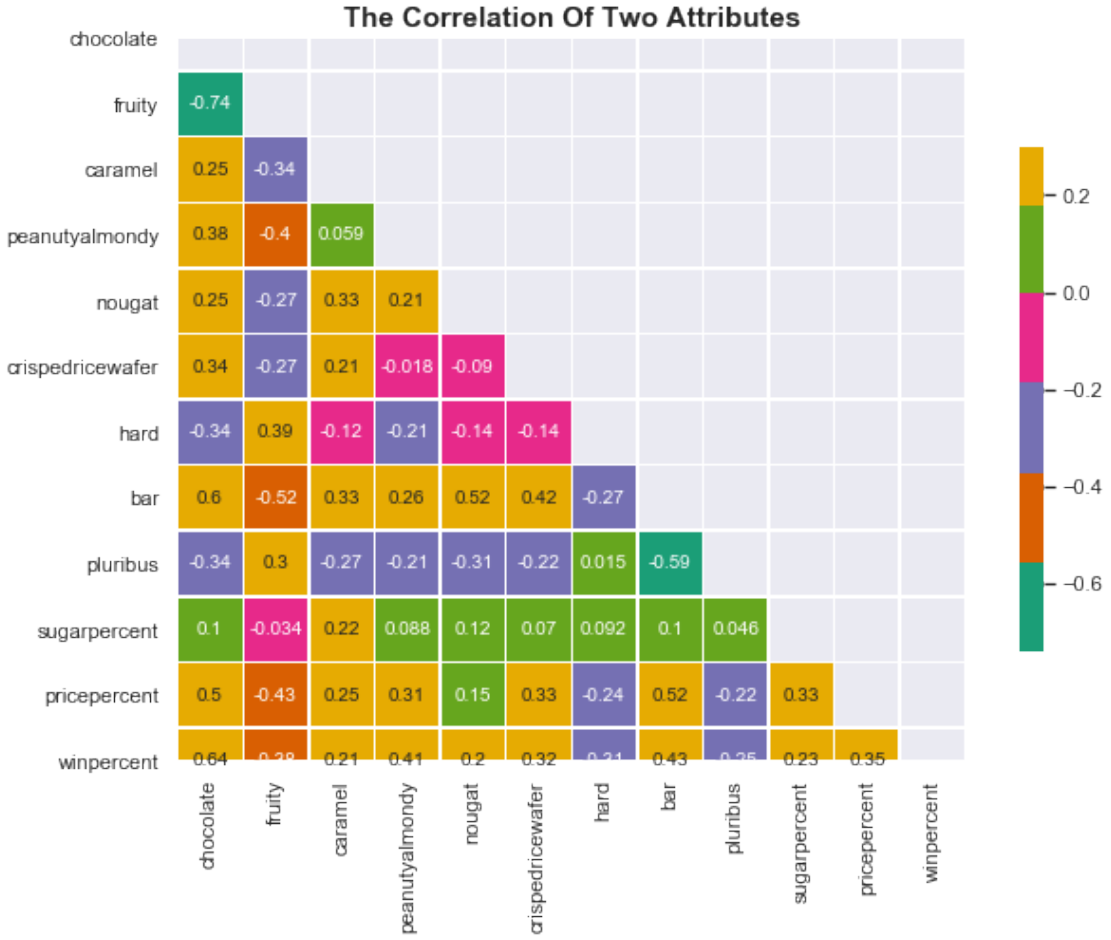
4 Fitting data

[9]:
```
corr_result=Pcorrelation(data=candy_data.iloc[:,1:13],
                         threshold=0.3,
                         figsizes=(10,7))
```

```
------------------------------------------------------------------
This program was developed by Phuong Van Nguyen
------------------------------------------------------------------
The number of the attributes has the higher-threshold correlation: 10
------------------------------------------------------------------
The list of the attributes has the higher-threshold correlation:
['fruity', 'caramel', 'peanutyalmondy', 'nougat', 'crispedricewafer', 'hard',
'bar', 'pluribus', 'pricepercent', 'winpercent']
------------------------------------------------------------------
------------------------------------------------------------------
Recording the higher-threshold correlations:
------------------------------------------------------------------
```

	Attribute_1	Attribute_2	Correlation_Value
0	chocolate	fruity	-0.741721
1	fruity	caramel	-0.335485
2	chocolate	peanutyalmondy	0.377824
3	fruity	peanutyalmondy	-0.399280
4	caramel	nougat	0.328493
5	chocolate	crispedricewafer	0.341210
6	chocolate	hard	-0.344177
7	fruity	hard	0.390678
8	chocolate	bar	0.597421
9	fruity	bar	-0.515066
10	caramel	bar	0.333960
11	nougat	bar	0.522976
12	crispedricewafer	bar	0.423751
13	chocolate	pluribus	-0.339675
14	nougat	pluribus	-0.310339
15	bar	pluribus	-0.593409
16	chocolate	pricepercent	0.504675
17	fruity	pricepercent	-0.430969
18	peanutyalmondy	pricepercent	0.309153
19	crispedricewafer	pricepercent	0.328265
20	bar	pricepercent	0.518407
21	sugarpercent	pricepercent	0.329706
22	chocolate	winpercent	0.636517
23	fruity	winpercent	-0.380938
24	peanutyalmondy	winpercent	0.406192

```
25  crispedricewafer        winpercent           0.324680
26              hard        winpercent          -0.310382
27               bar        winpercent           0.429929
28      pricepercent        winpercent           0.345325
------------------------------The End----------------------
```



Thank you for reading my code

Any comments are warmly welcome at phuong.nguyen@summer.barcelonagse.eu