```
# programmer - Sophia Quinton
# date - 12-15-21
# class - DSC -540
# assignment - Assignment 7

##libraries
import numpy as np
import pandas as pd
from fcmeans import FCM
from matplotlib import pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

##load dataset (Charytoanowicz et. al., 2010)
data = pd.read_csv("C:/Users/sophi/OneDrive/Desktop/Graduate Classes/DSC -
540 Machine Learning/Week 7/Seed_Data.csv")
data

         A      P       C     LK     WK  A_Coef    LKG  target
0    15.26  14.84  0.8710  5.763  3.312   2.221  5.220       0
1    14.88  14.57  0.8811  5.554  3.333   1.018  4.956       0
2    14.29  14.09  0.9050  5.291  3.337   2.699  4.825       0
3    13.84  13.94  0.8955  5.324  3.379   2.259  4.805       0
4    16.14  14.99  0.9034  5.658  3.562   1.355  5.175       0
..     ...    ...     ...    ...    ...     ...    ...     ...
205  12.19  13.20  0.8783  5.137  2.981   3.631  4.870       2
206  11.23  12.88  0.8511  5.140  2.795   4.325  5.003       2
207  13.20  13.66  0.8883  5.236  3.232   8.315  5.056       2
208  11.84  13.21  0.8521  5.175  2.836   3.598  5.044       2
209  12.30  13.34  0.8684  5.243  2.974   5.637  5.063       2

[210 rows x 8 columns]

#kmeans (Pedregosa et. al., 2011)
column_names = data.columns[0:7]
combo_final = []
score_data = pd.DataFrame(columns = ["first", "second", "third"])
for x in range(3):
    scores = []
    for i in column_names:
        for j in column_names:
            if i != j:
                combo = i + " " + j
                combo_final.append(combo)
                Group = data[[i,j]]
                Kmeans_model = KMeans(n_clusters=3)
                label = Kmeans_model.fit_predict(Group)
                accuracy = accuracy_score(data["target"], label)
                scores.append(accuracy)
    if x == 0:
        score_data["first"] = scores
    elif x == 1:
        score_data["second"] = scores
    elif x == 2:
        score_data["third"] = scores

score_data
```

```
       first    second     third
0   0.152381  0.252381  0.252381
1   0.842857  0.342857  0.842857
2   0.342857  0.247619  0.004762
3   0.342857  0.247619  0.342857
4   0.347619  0.290476  0.347619
5   0.342857  0.247619  0.842857
6   0.000000  0.252381  0.252381
7   0.247619  0.366667  0.366667
8   0.371429  0.009524  0.852381
9   0.361905  0.252381  0.361905
10  0.423810  0.085714  0.847619
11  0.380952  0.238095  0.842857
12  0.004762  0.342857  0.842857
13  0.366667  0.366667  0.000000
14  0.414286  0.390476  0.414286
15  0.142857  0.395238  0.819048
16  0.552381  0.423810  0.390476
17  0.647619  0.647619  0.642857
18  0.342857  0.409524  0.342857
19  0.852381  0.852381  0.852381
20  0.414286  0.771429  0.414286
21  0.247619  0.014286  0.014286
22  0.195238  0.409524  0.252381
23  0.538095  0.742857  0.538095
24  0.247619  0.004762  0.247619
25  0.361905  0.252381  0.000000
26  0.142857  0.261905  0.819048
27  0.347619  0.247619  0.347619
28  0.252381  0.180952  0.409524
29  0.423810  0.033333  0.423810
30  0.023810  0.895238  0.347619
31  0.847619  0.423810  0.847619
32  0.385714  0.195238  0.261905
33  0.390476  0.395238  0.566667
34  0.252381  0.409524  0.409524
35  0.214286  0.214286  0.238095
36  0.842857  0.409524  0.004762
37  0.019048  0.238095  0.014286
38  0.647619  0.647619  0.647619
```

```
39  0.085714  0.538095  0.742857
40  0.885714  0.033333  0.885714
41  0.395238  0.395238  0.214286
```

```
combo_final[19]
```

```
'LK P'
```

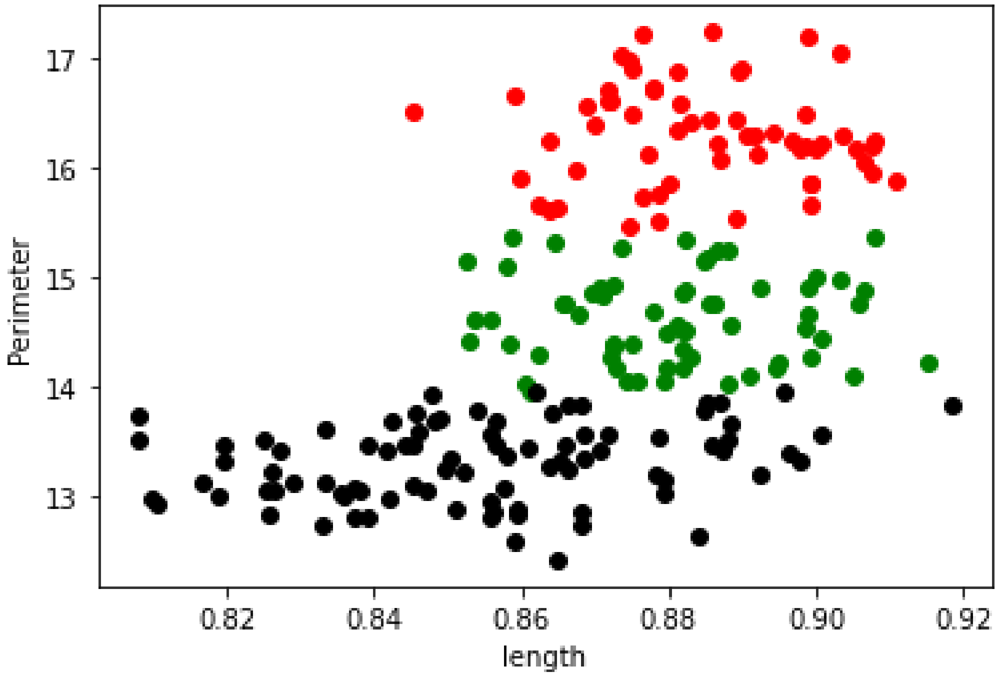
```
combo_final[30]
```

```
'A_Coef A'
```

```
# LK-P (AskPython, 2020) (Hunter, 2007)
Group = data[["LK", "P"]]
Kmeans_model = KMeans(n_clusters=3)
label = Kmeans_model.fit_predict(Group)
accuracy = accuracy_score(data["target"], label)
print("The accuracy of LK-P: ", accuracy)

# (AskPython, 2020) (Hunter, 2007)
tlabel_0 = data[label == 0]
tlabel_1 = data[label == 1]
tlabel_2 = data[label == 2]
plt.scatter(tlabel_0["C"], tlabel_0["P"], color="green")
plt.scatter(tlabel_1["C"], tlabel_1["P"], color="red")
plt.scatter(tlabel_2["C"], tlabel_2["P"], color="black")
plt.xlabel("length")
plt.ylabel("Perimeter")
plt.show()
```

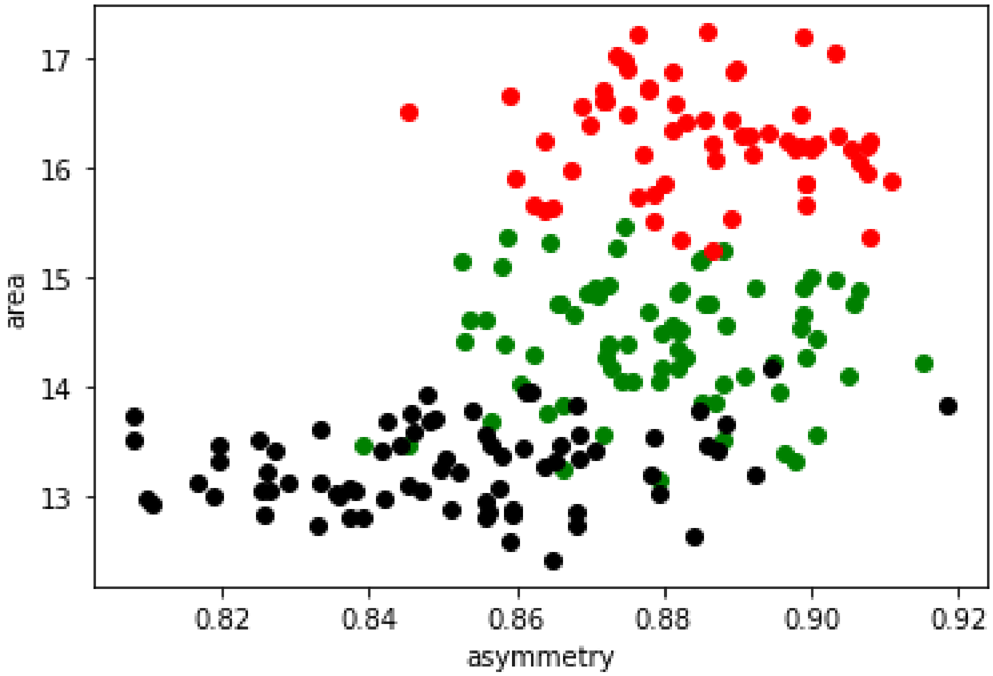
```
The accuracy of LK-P:  0.8523809523809524
```



```
# A_Coef - A (AskPython, 2020) (Hunter, 2007)
Group = data[["A_Coef", "A"]]
Kmeans_model = KMeans(n_clusters=3)
label = Kmeans_model.fit_predict(Group)
accuracy = accuracy_score(data["target"], label)
print("The accuracy of LK-P: ", accuracy)

# (AskPython, 2020) (Hunter, 2007)
tlabel_0 = data[label == 0]
tlabel_1 = data[label == 1]
tlabel_2 = data[label == 2]
plt.scatter(tlabel_0["C"], tlabel_0["P"], color="green")
plt.scatter(tlabel_1["C"], tlabel_1["P"], color="red")
plt.scatter(tlabel_2["C"], tlabel_2["P"], color="black")
plt.xlabel("asymmetry")
plt.ylabel("area")
plt.show()
```

```
The accuracy of LK-P:  0.8952380952380953
```



```
#fuzzy (Dias, 2019)
fcm = FCM(n_clusters=3)
fcm.fit(np.array(Group))
fcm_labels = fcm.predict(np.array(Group))

accuracy = accuracy_score(data["target"], fcm_labels)
print("The fcm is: ", accuracy)
```

```
The fcm is:  0.8952380952380953
```

```
from sklearn.metrics import classification_report
print(classification_report(np.array(data["target"]), label))
```

```
              precision    recall  f1-score   support

           0       0.82      0.87      0.85        70
           1       0.98      0.87      0.92        70
           2       0.89      0.94      0.92        70

    accuracy                           0.90       210
   macro avg       0.90      0.90      0.90       210
weighted avg       0.90      0.90      0.90       210
```

```
Kmeans_model.cluster_centers_
```

```
array([[ 2.63389459, 14.5427027 ],
       [ 3.61348387, 18.68451613],
       [ 4.83916216, 11.93756757]])
```