Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій

Кафедра програмних систем і технологій

ЗВІТ

з практичної роботи № 5

Тема: «Алгоритми кластеризації за допомогою `Scikit-learn»

Дисципліна «Спеціалізоване програмування автоматизованих систем»

Підготував:
студент гр. ІПЗ-33(1)
Мішак Максим

Перевірила:
Ніколаєнко Анастасія Юріївна

Завдання :

Реалізувати алгоритм кластеризації K Means за допомогою Scikit-learn на мові програмування Python . Використати відстань Мінковського . Використовувати данні з файлу Seeds.csv.

Хід роботи :

Для реалізаці та легшого розуміння завдання можна розділити програму на три умовних блоки :

1. Імпорт данних в программу
2. Обробка данних
3. Вивід результату

Виконувати перший блок будемо за допомогою бібліотеки Pandas .
Фрагмент коду :

```
DataSet = pd.read_csv('dataset/seeds.csv')
```

```
DataSetSize = DataSet.iloc[:,:-1].values
```

```
DataSetClass = DataSet.iloc[:,7].values
```

В цьому коді присутні такі дії : імпорт данних з файлу csv за допомогою функції read_csv , в параметри якого передаємо шлях та назву файлу . Наступними двома командами розділяємо на два класа

Наступним кроком буде обробка данних , для якої будемо використовувати бібліотек для машинного навчання Scikit-learn . а саме : клас metrics для аналізу данних та клас cluster, а саме метод KMeans .

Фрагмент коду :

```
ClusterFunction = KMeans(n_clusters=6)
```

```
ClusterFunction.fit(DataSetSize)
```

```
labels = ClusterFunction.labels_
```

```
metrics.silhouette_score(DataSetSize,labels,metric='
minkowski')
```

```
Predictions = ClusterFunction.predict(DataSetSize)
```

```
DataSet['cluster']=Predictions
```

`ClusterFunction = KMeans(n_clusters=6)`	Створюємо змінну `ClusterFunction`, в якій викликаємо метод `KMeans` , в параметри якого передаємо аргумент `n_clusters` з кількістью кластерів
`ClusterFunction.fit(DataSetSize)`	Викликаємо метод `fit` який відповідає за навчання моделі , для цього в параметри функції передаємо параметр `DataSetSize` , який репрезентує частину нахих данних
`metrics.silhouette_score(DataSetSize,labels,metric='minkowski')`	Обчислюємо середню відстань між елементами за допомогою цієї функції . В параметри функції передаємо об’єкт з данними `DataSetSize` , заголовки `labels` , та вид метрика , по якому обчислюємо данні `metric='minkowski'`
`Predictions = ClusterFunction.predict(DataSetSize)`	Створюємо змінну , в якій викликаємо функцію `predict` в параметри якого передаємо `DataSetSize` для передбачення данних
`DataSet['cluster']=Predictions`	Додаємо до данних ще одну колонку , яка має назву `'cluster'` , яка репрезентує кластер , до якого алгоритм відніс екземпляр данних

Для виводу данних нам знадобиться бібліотека matplotlib , а саме клас puplot та клас ListedColormap .

Фрагмент коду останнього компоненту програми :

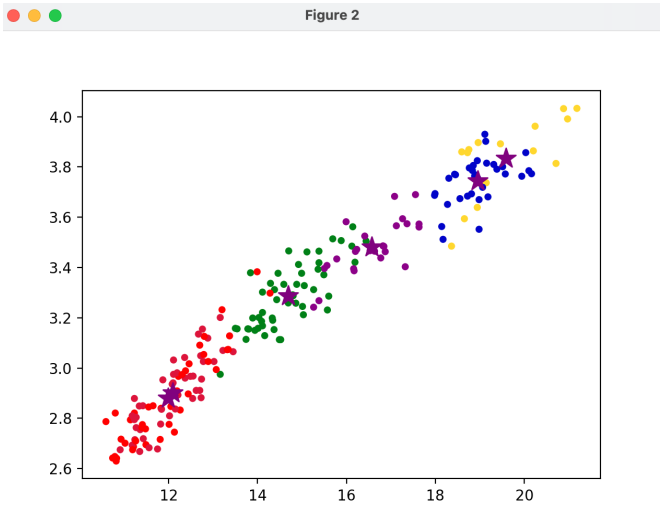
```
Centroid = ClusterFunction.cluster_centers_
```

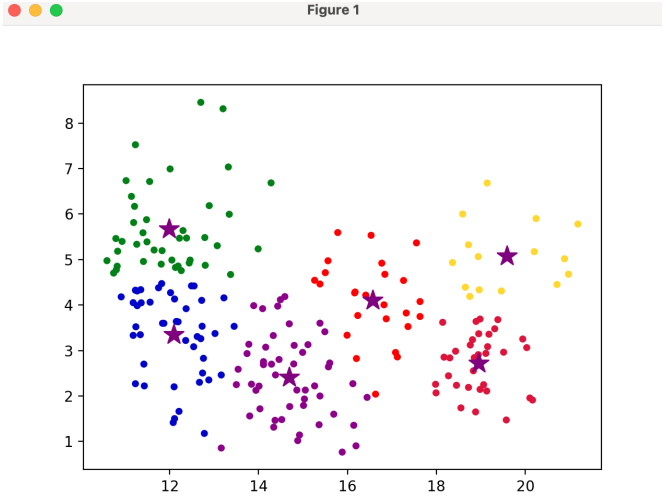
```
count_clusters = Counter(labels)
fig , ax = plt.subplots()
ColorMap = ListedColormap(['crimson', 'mediumblue',
'darkmagenta','yellow','red','green','gold'])
ScatterMatrixResults =
ax.scatter(DataSetSize[:,0],DataSetSize[:,5],c=Predi
ctions,s=15,cmap =ColorMap)
ScatterMatrixCentroids =
ax.scatter(Centroid[:,0],Centroid[:,5],marker='*',c=
'purple',s=200,label='centroids')
plt.show()
print(DataSet)
print(Centroid)
print(count_clusters)
```

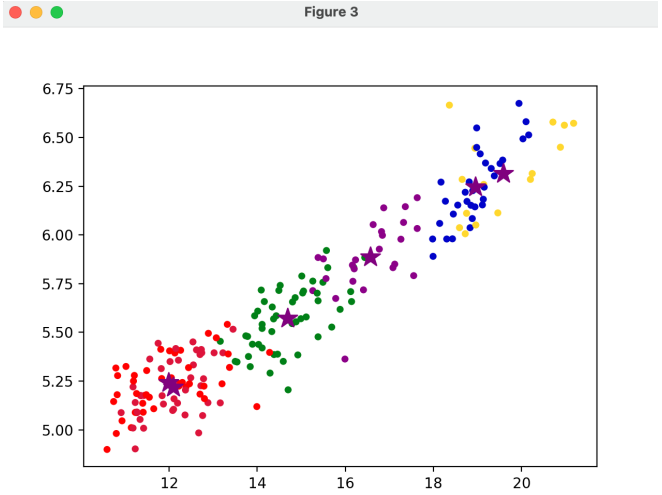
`Centroid = ClusterFunction.cluster_centers_`	Створюємо змінну `Centroid` , яку передаємо координати центроїд
`ColorMap = ListedColormap(['crimson', 'mediumblue', 'darkmagenta','yellow','red','green','gold'])`	Створюємо карту кольорів , за допомогою якої буде зручно та зрозуміліше виводити данні
`ScatterMatrixResults = ax.scatter(DataSetSize[:,0],DataSetSize[:,5],c=Predictions,s=15,cmap =ColorMap)` `ScatterMatrixResults1 = ax1.scatter(DataSetSize[:`	Виводимо данні за допомогою `scatter` в параметри якого передаємо данні для виводу та кольорову мапу

``,0],DataSetSize[:,4],c=Predictions,s=15,cmap=ColorMap)``` ```ScatterMatrixCentroids1 = ax1.scatter(Centroid[:,0],Centroid[:,4],marker='*',c='purple',s=200,label='centroids')```	
``print(DataSet)``` ```print(Centroid)``` ```print(count_clusters)``` ```ScatterMatrixResults2 = ax2.scatter(DataSetSize[:,0],DataSetSize[:,3],c=Predictions,s=15,cmap=ColorMap)``` ```ScatterMatrixCentroids2 = ax2.scatter(Centroid[:,0],Centroid[:,3],marker='*',c='purple',s=200,label='centroids')```	Виводимо інформацію в консоль

Результати роботи програми :







```
/Users/macbook/Desktop/KHY/SPAS./spas5/venv/bin/python /Users/macbook/Desktop/KHY/SPAS./spas5/main.py
        V1     V2      V3     V4     V5     V6     V7  class  cluster
0    15.26  14.84  0.8710  5.763  3.312  2.221  5.220      1        2
1    14.88  14.57  0.8811  5.554  3.333  1.018  4.956      1        2
2    14.29  14.09  0.9050  5.291  3.337  2.699  4.825      1        2
3    13.84  13.94  0.8955  5.324  3.379  2.259  4.805      1        2
4    16.14  14.99  0.9034  5.658  3.562  1.355  5.175      1        2
..     ...    ...     ...    ...    ...    ...    ...    ...      ...
205  12.19  13.20  0.8783  5.137  2.981  3.631  4.870      3        1
206  11.23  12.88  0.8511  5.140  2.795  4.325  5.003      3        1
207  13.20  13.66  0.8883  5.236  3.232  8.315  5.056      3        4
208  11.84  13.21  0.8521  5.175  2.836  3.598  5.044      3        1
209  12.30  13.34  0.8684  5.243  2.974  5.637  5.063      3        4

[210 rows x 9 columns]
[[18.95454545 16.38878788  0.8868      6.24748485  3.74469697  2.72354545
   6.11945455]
 [12.09045455 13.30977273  0.85708636  5.21740909  2.90065909  3.34375
   5.00531818]
 [14.69294118 14.47411765  0.88094314  5.57213725  3.28643137  2.40790588
   5.15882353]
 [16.562      15.3916      0.878244    5.88816     3.4808      4.10948
   5.7252    ]
 [11.9847619  13.29357143  0.85079524  5.24138095  2.8797381   5.6732619
   5.12197619]
 [19.58333333 16.646       0.88772667  6.31586667  3.83506667  5.08153333
   6.1444    ]]
Counter({2: 51, 1: 44, 4: 42, 0: 33, 3: 25, 5: 15})

Process finished with exit code 0
```

Висновок :

Проаналізувавши результати виконання програми , можемо зробити висновок , що програма має велику точність , вірно розділяє данні на кластери . Також з графіка видно , наскільки точно центроїди знаходяться в центрі кластерів .