

مبانی هوش محاسباتی

گزارش تمرین اول

شرح کلی:

برای clustering این دیتاست به ۳ روش dbscan , kmeans و agglomerative از کتابخانه scikit learn پایتون کمک می‌گیریم. پیاده‌سازی الگوریتم‌های فوق به کمک توابع این کتابخانه قابل انجام است. برای سنجش نتایج به دست آمده توسط هر الگوریتم بایستی مقدار rand index را محاسبه کرد که خود این مقدار وابسته به ۴ مولفه TP , TN , FP, FN می‌باشد.

نحوه محاسبه rand index :

این میزان با استفاده از فرمول زیر محاسبه می‌شود.

$$\text{rand index} = \frac{TP+TN}{TP+FP+TN+FN}$$

Rand Index

```
[4] def RandIndex(clusteredData):
    TP_and_FP = calculateTP_FP(clusteredData)
    TP = calculateTP(clusteredData)
    FN = 1800 - TP
    totalClusteredData = combination(410 - np.count_nonzero(clusteredData == -1))
    TN = totalClusteredData - (FN + TP_and_FP)
    randIndex = (TP + TN) / totalClusteredData
    print(f"rand index is : {randIndex}")
```

نتایج معیار rand index :

• Kmeans

Algorithm Evaluation

```
1s ✓ RandIndex(result)
rand index is : 0.9537837676665275
```

• Dbscan

Algorithm Evaluation

```
0s ✓ RandIndex(result)
rand index is : 0.3277857484457197
```

Agglomerative •

Single Link ○

Algorithm Evaluation

✓
0s



RandIndex(result)

rand index is : 0.7033573856520962

Complete Link ○

Algorithm Evaluation

✓
0s



RandIndex(result)

rand index is : 0.9448506172103286

Average Link ○

Algorithm Evaluation

✓
0s



RandIndex(result)

rand index is : 0.9244677679050629

همانطور که مشاهده می‌کنید الگوریتم dbscan امتیاز بسیار پایینی در این معیار دریافت کرده زیرا بسیاری از داده‌ها را به شکل نویز تشخیص داده است.

در ضمن، نتایج کلی کلاسترینگ پس از اتمام الگوریتم در پوشه هایی با نام هر کلاستر به شکل زیر ذخیره می شود. Paths شامل آدرس تمامی عکس هاست. و result نتایج کلاسترینگ را شامل می شود.

Collecting Images

```
[ ] for i in range(num_of_clusters):  
    os.makedirs("cluster" + str(i))  
  
    for i in range(len(paths)):  
        shutil.copy2(paths[i], "cluster" + str(result[i]))
```

