Amirkabir University of Technology
(Tehran Polytechnic)

Principles of Data mining

Assignment 2

Sepehr Asgarian

Student_ID: 9531901

2020

In the name of God

Data mining

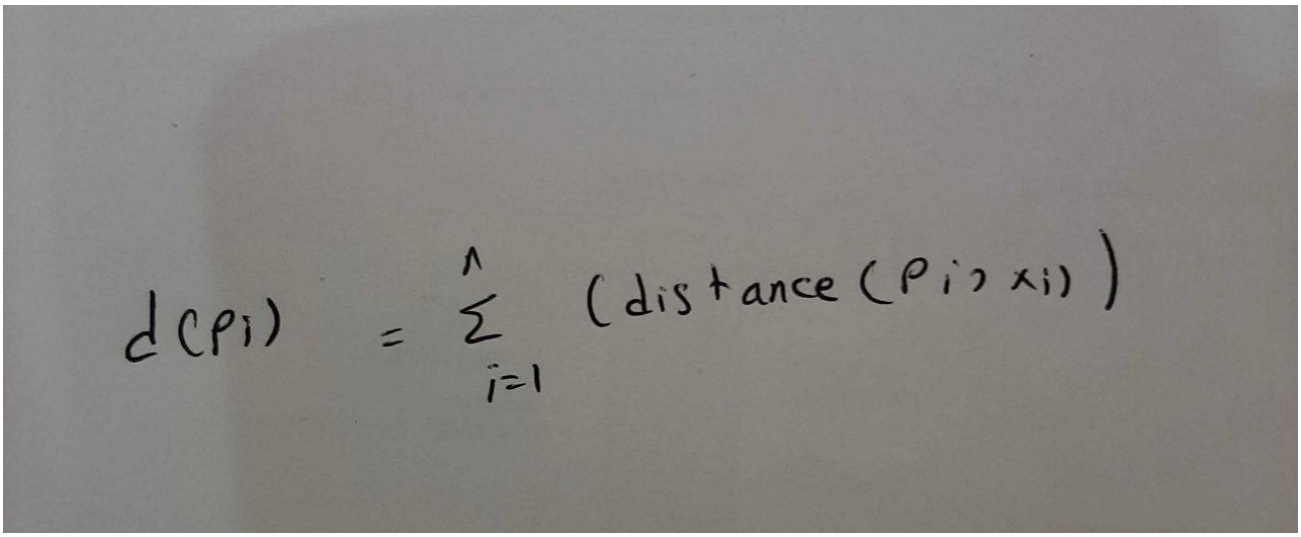
First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**

1

k-means یکی از محبوب ترین الگوریتم های خوشه بندی است که به دلیل عملکرد خوب از لحاظ زمانی بسیار مورد توجه است. هر چند که با افزایش حجم دیتاست، زمان محاسبه ی این الگوریتم افزایش می یابد زیرا لازم دارد که تمام دیتا را در حافظه ی اصلی نگه دارد. به این منظور الگوریتم ها و روش های متعددی برای رفع این مشکل و کاهش هزینه ی آن پیشنهاد شده‌اند که یکی از آن ها Mini batch K-means است. ایده ی اصلی در این الگوریتم این است که از دسته های تصادفی با سایز کوچک و ثابت استفاده می کند تا آن ها را به جای کل دیتاست در حافظه اصلی نگه دارد. در هر دور الگوریتم یک دسته ی تصادفی انتخاب و برای آپدیت کردن خوشه ها استفاده می شود تا همگرایی به خوشه های اصلی حاصل گردد. نتایج نشان می دهند که برای این الگوریتم با آنکه زمان محاسباتی قابل توجهی صرفه جویی می شود اما کیفیت کلاسترها اندکی پایین تر می آید هر چند که نحوه ی دقیق ویژگی های دیتاست که بر این موضوع اثر می گذارند دقیقا اندازه گیری نشده است. بنابراین اگرچه این الگوریتم سریعتر است، اما نتایج خوشه بندی آن ممکن است با مقدار واقعی اندکی متفاوت باشد

2(

ایده ی اول به صورت زیر بیان می شود: فاصله ی هر نقطه از دیتاست از تک تک نقاط را با کمک فرمول زیر حساب می کنیم. سپس داده ها را برحسب این فواصل بدست آمده مرتب سازی می کنیم. سپس کل دیتاست را بر این اساس به k تا دسته ی هم اندازه تقسیم می کنیم. اولین نقطه ی هر دسته را به عنوان centroid در نظر میگیریم.

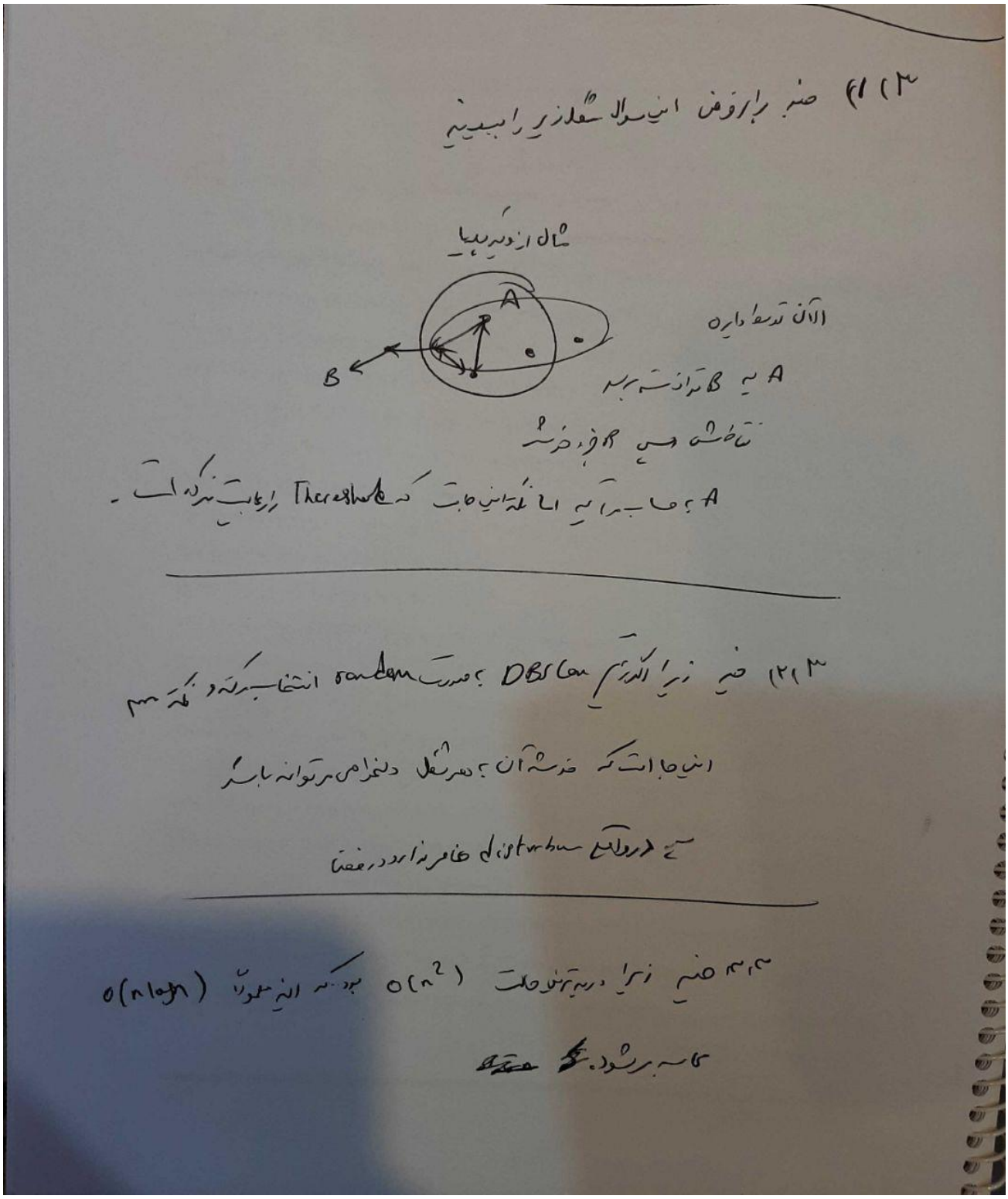


ایده ی دوم به این صورت است: از بین n داده، نقطه ای را حساب می کنیم که مقدارویژگی (attributeمورد نظرش میانگین بقیه ی n داده باشد. در واقع اولین centroid میانگین n داده است. Centroid های بعدی را به گونه ای انتخاب می کنیم که فاصله اقلیدسی شان از سایر centroid های قبلی انتخاب شده، ماکزیمم باشد. حال مرحله قبل را تا جایی ادامه می دهیم که k تا centroid بدست آوریم و بدین ترتیب centroid های خود را برای الگوریتم k means بدست آورده ایم

In the name of God

Data mining

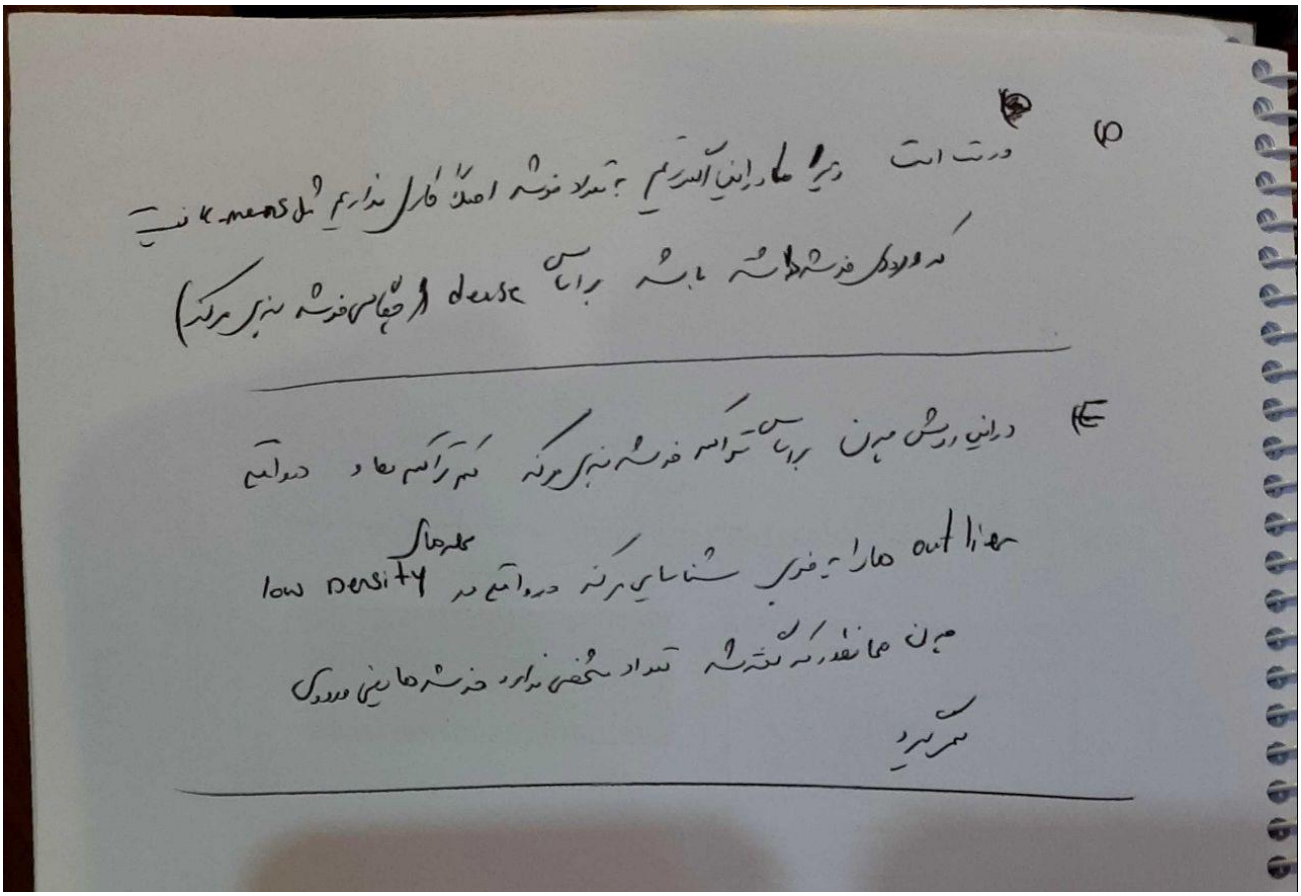
First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**



In the name of God

Data mining

First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**

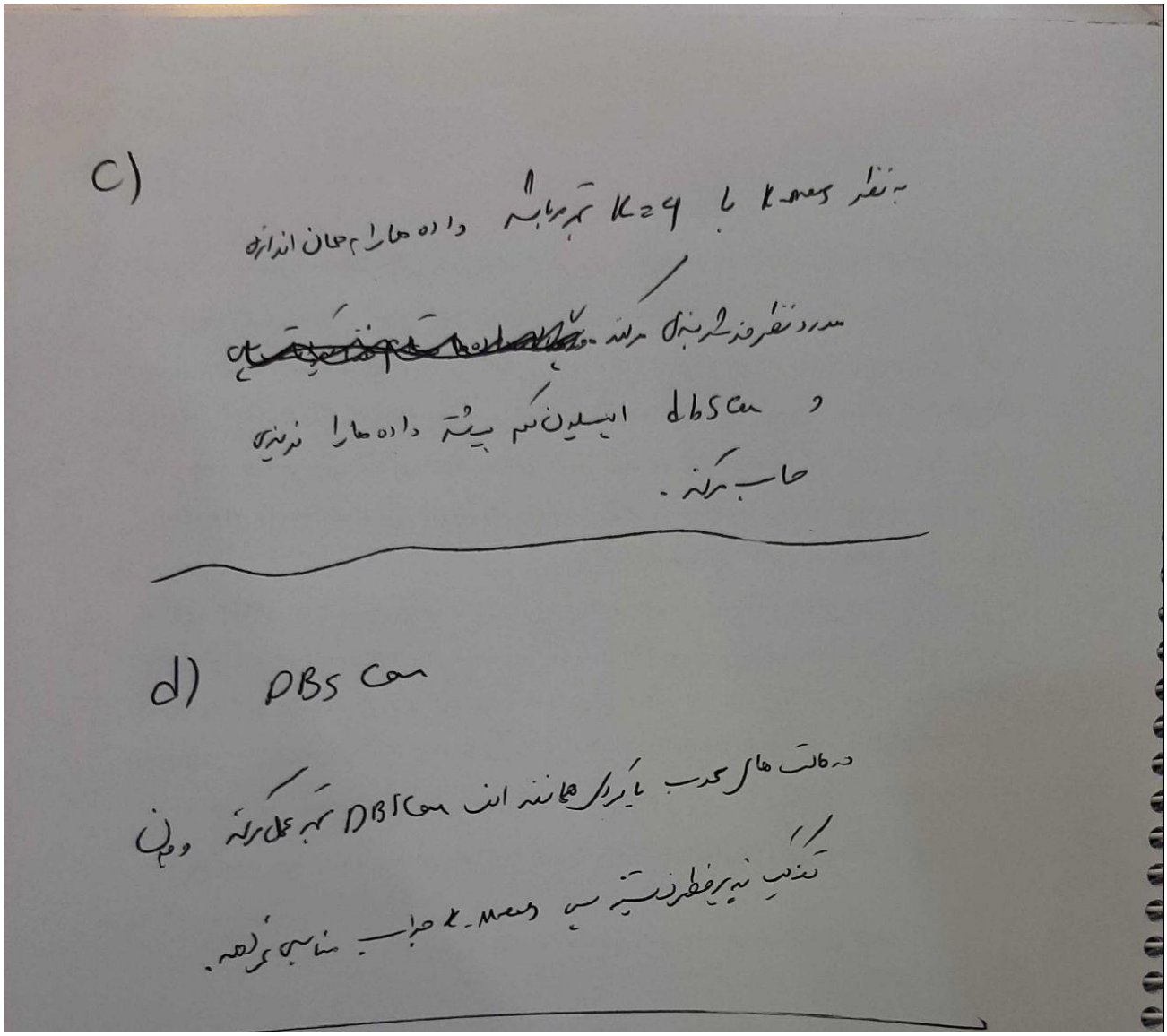


سوال 4

In the name of God

Data mining

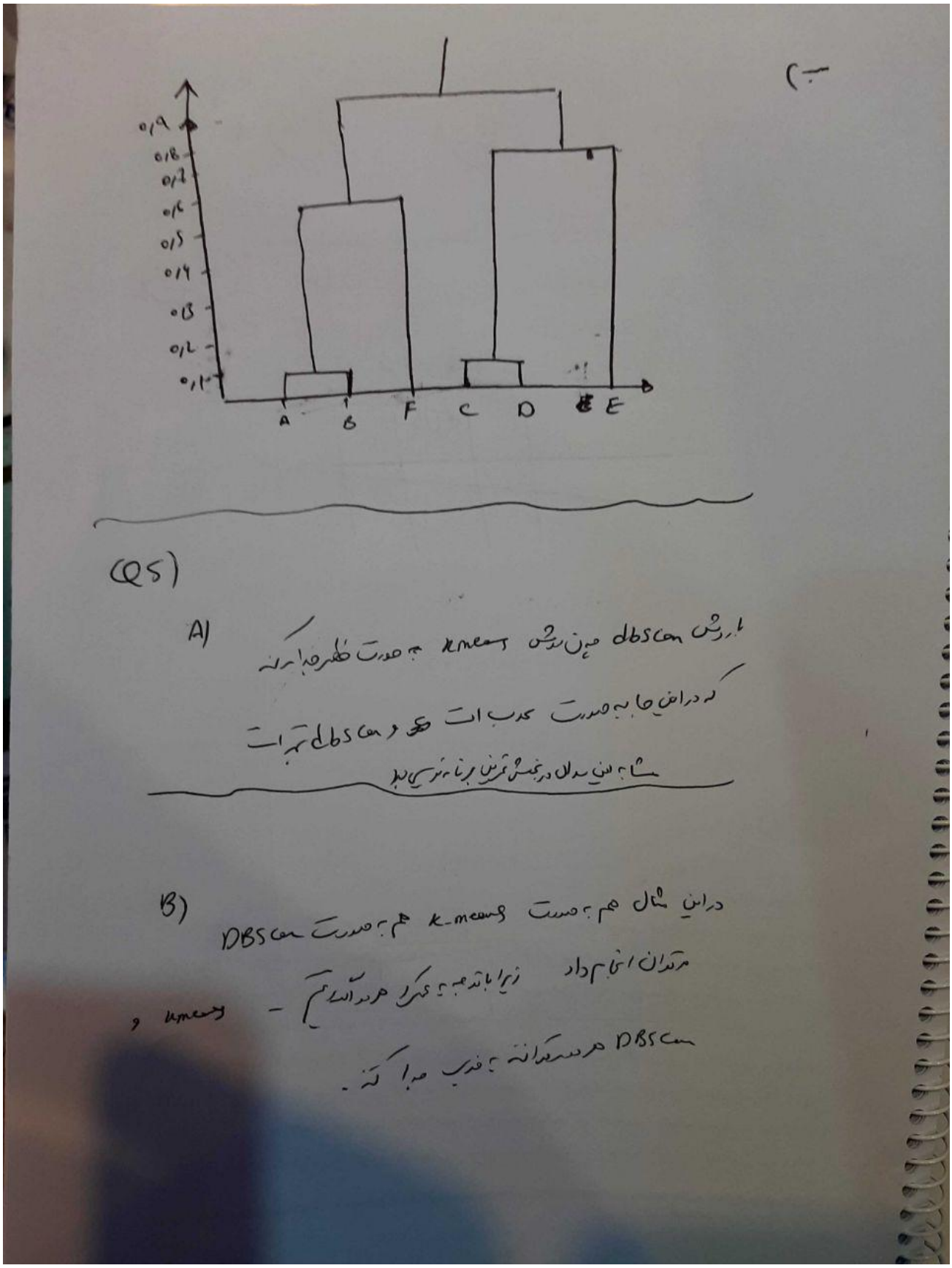
First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**



In the name of God

Data mining

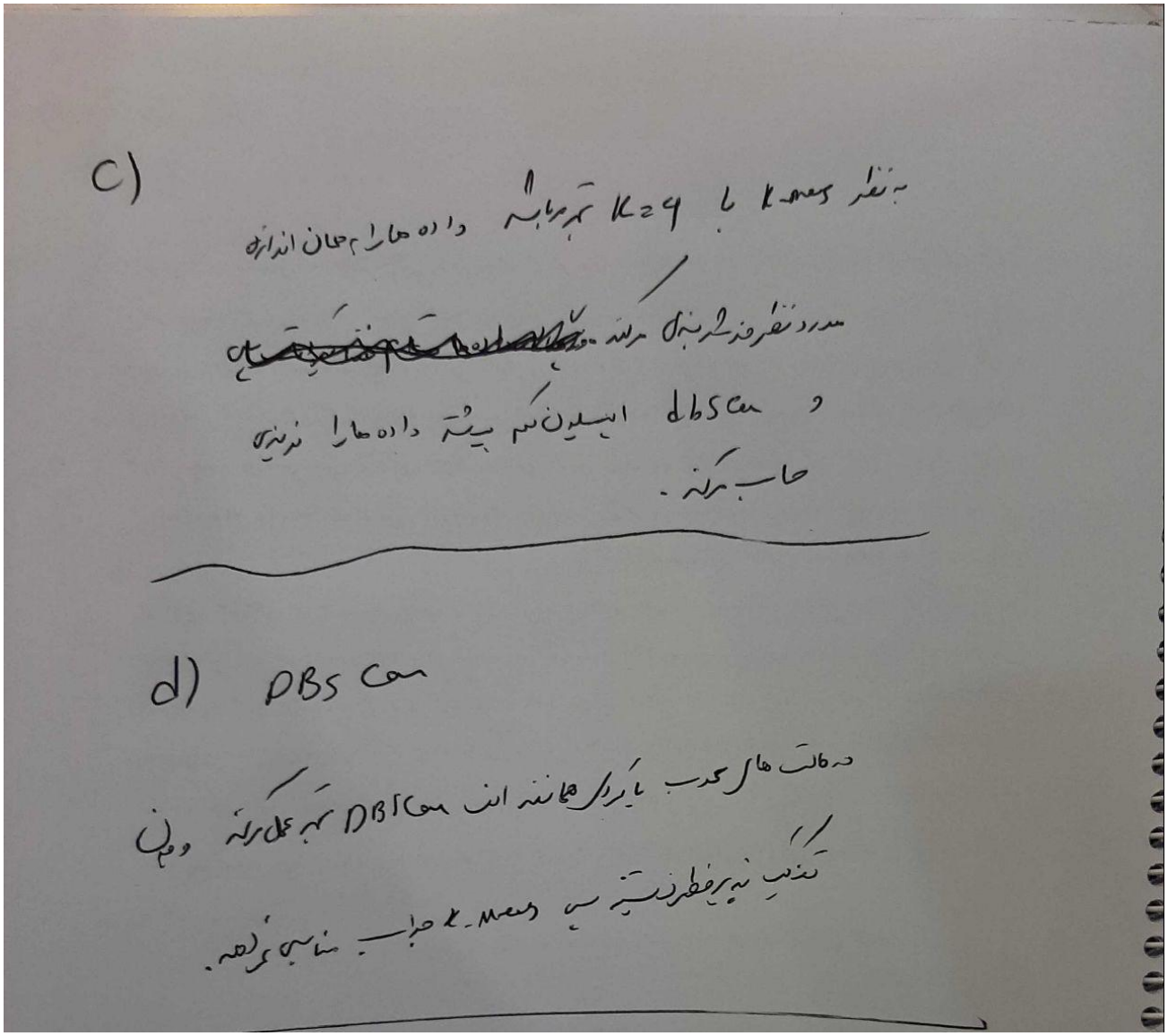
First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**



In the name of God

Data mining

First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**



implementation:

In this I used colab as a Jupyter notebook ☺

1) After completing the algorithm, run it on Dataset1. Set number of iteration to at least 15 and run K-means with k=2, 3, 4. After each run plot the clustered data points with each cluster having a different color. (you can use matplotlib to visualize the data)

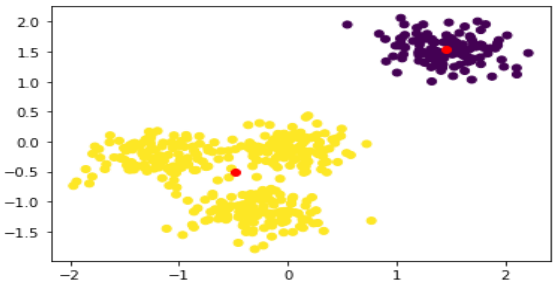
In the name of God

Data mining

First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**

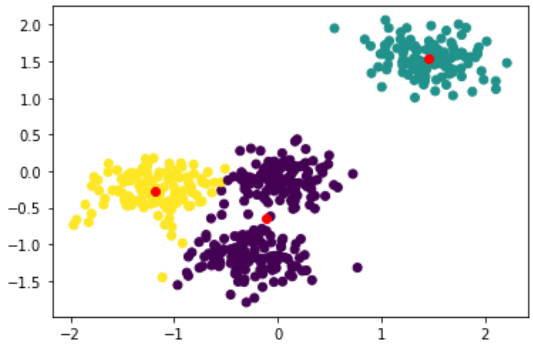
در این مساله از تعداد تکرار 20 استفاده کرده ام و دیتا ست اول پس از 20 تکرار با توجه به موارد خواسته شده در زیر ارائه شده است که اعداد مقابل آن به ترتیب error هر cluster هستند و عدد نهایی میانگین آن است. همچنین با توجه به الگوریتم نوشته شده دقیقا انجام داده ام که در قسمت کد ها کد آن قابل مشاهده است. البته لازم به ذکر است در ایتریشن های پایین ممکن است این شکل ها تفاوت داشته باشند.

برای k=2



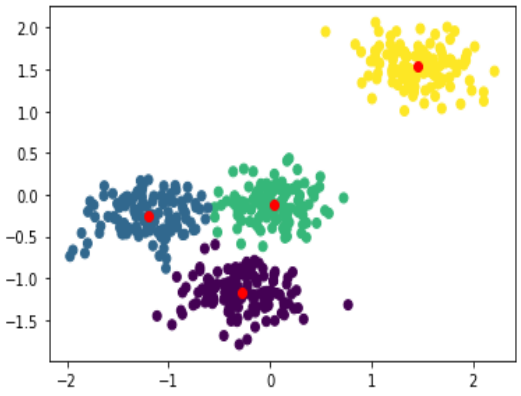
```
[0.3160088195773473, 0.7491160326963793] 0.5325624261368633
```

برای k=3



```
[0.6163639287851186, 0.3160088195773473, 0.33987023164913976]
0.42408099333720184
```

برای k=4



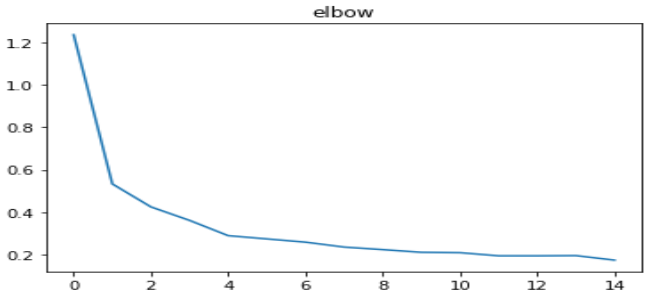
```
[0.33612332094004477, 0.32383689405410077, 0.28865973975955667, 0.3160088195773473] 0.3161571935827624
```

قسمت B و C

در قسمت بالا علاوه بر شکل ها تمامی فواصل از کلاستر ها نمایش داده شده و عدد آخر نیز میانگین آن است برای فرض برای K=2 مقدار 0.5325624261368633 میانگین خطای آن میباش

قسمت D

با ایتریشن 20 حساب شده است یک for روی kmean ای که ساخته ایم لازم بود زده شود که در کد آپلود شده موجود است



D

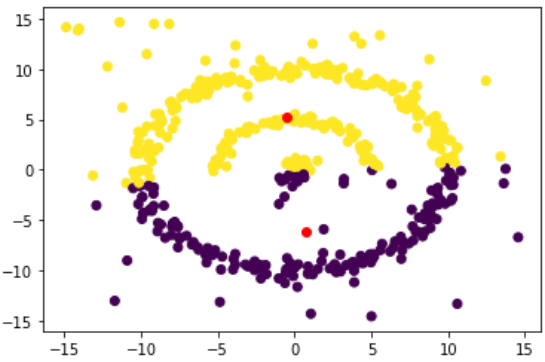
E

با توجه به الگوریتم elbow که در کلاس درس نیز بحث شد که به صورت کلی در زیر بحث شده است که در نمودار بالا جایی است که تغییر بزرگ دیگری در کل خوشهای از اندازه مربع وجود ندارد و به یک k مورد قبول که طبق شکل بالا 4 است میرسیم حتی 5 هم میتواند باشد.

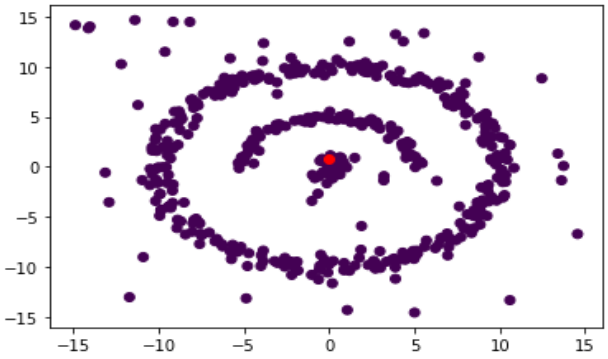
To determine the optimal number of clusters, we have to select the value of k at the “elbow” ie the point after which the distortion/inertia start decreasing in a linear fashion.

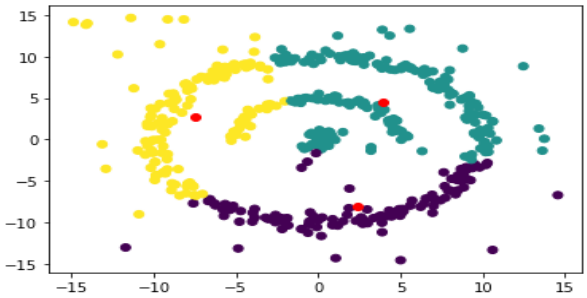
F

مشابه این شکل در بخش تمرین نوشتنی نیز بود و هیمنطور در کلاس نیز در مورد با آن بحث شد و چون محدب است و الگوریتم kmeans درواقع به صورت خطی جدا میکند نمیتواند نتیجه ی خوبی بر بروی این دیتا ست بگیرد و خوشه های آن همانطور که در نتایج پایین میبینید به خوبی خوشه بندی نشده است همانند بخش اول error ها و پس از آن میانگین خطای کلاستر ها در کنار شکل ها آمده است تعداد تکرار برابر با 20 و تعداد کلاستر ها نیز به تعداد رنگ های موجود در شکل میباشد

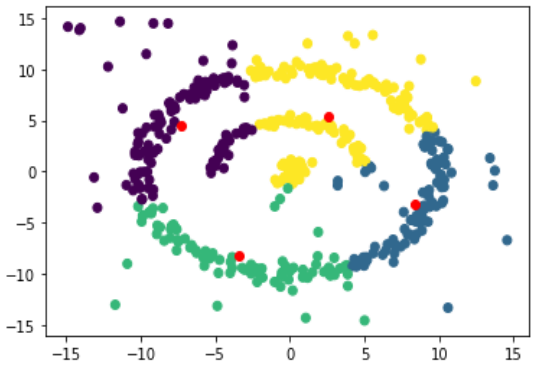


[7.135336441295711, 6.681852305017779] 6.908594373156745

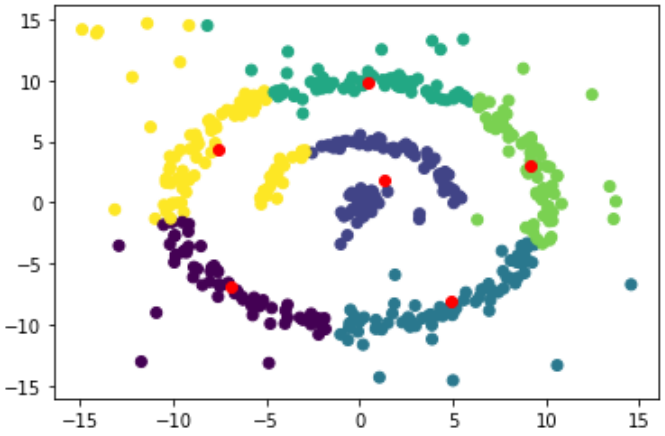




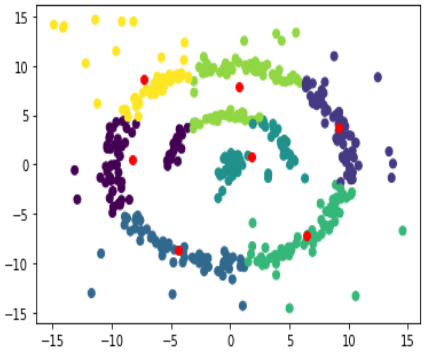
[5.302349082021818, 5.384613569723759, 5.41528542920326] 5.367416026982947



[4.583236343246868, 4.126541221266248, 4.681088492562955, 4.803782632152678] 4.5486621723071865



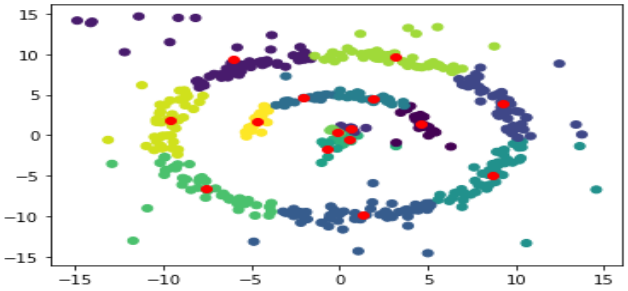
[3.4300301558844795, 2.973304556993111, 3.662230877793421, 3.136407085582292, 3.4449762899763727, 4.098589559972069] 3.4575897543669574

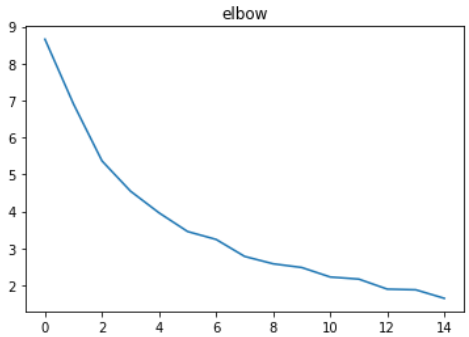


[3.4213581310105616, 3.1362922608085118, 3.414881711700733, 2.579114520160099, 3.371878059079515, 3.5139660650732916, 3.2679059129324997] 3.24362809439503

و در نهایت برای k=15به شکل زیر میباشد و ارور آن به شکل زیر میباشد در بالای عکس آمده و همینطور نموداری که برای elbow استفاده میکنیم نیز نمیتوان elbow ای در نظر گرفت با توجه به صحبت هایی که شد و در نمودار نیز قابل مشاهده است

[1.2156712325839891, 3.2578676260833244, 0.40032970023046094, 2.5146377874055172, 2.695047495491773,

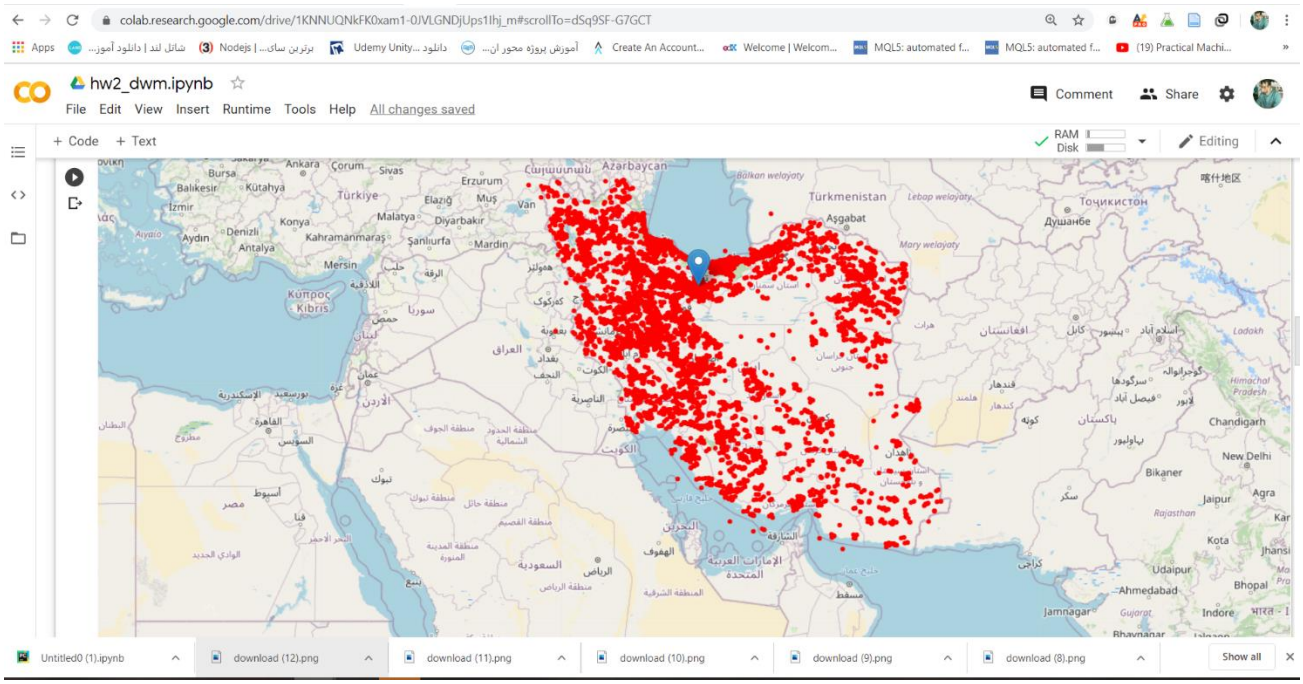




1.0970757360688645, 1.119660900746635, 2.5006864693205344, 0.9369609779747158, 0.5682691645541824, 2.6011403863532463, 0.29261134881610457, 2.309720658323371, 2.247041420851091, 0.9985598376948854] 1.6503520494999129

2)

دقیقا کدی که قرار داده شده بود رو روی دیتا ست covid ران کردم و خروجی مشابه زیر است



2(

دقیقا از خود dbscan library استفاده کردم و import کردم برای مثال عدد دلبخواه

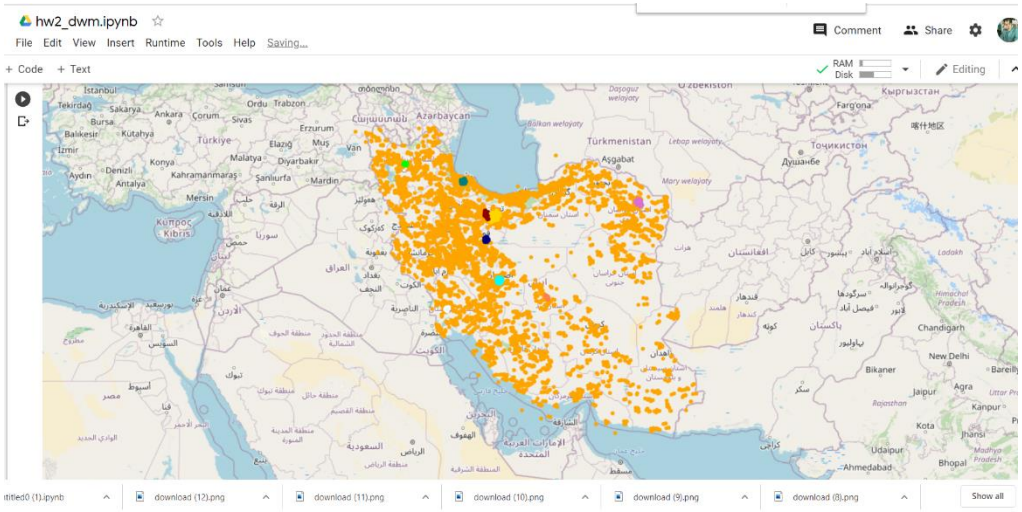
00)3, min_samples=9 predictor = DBSCAN(eps=0.

را استفاده کردم که نتیجه آن به شکل زیر میباشد

In the name of God

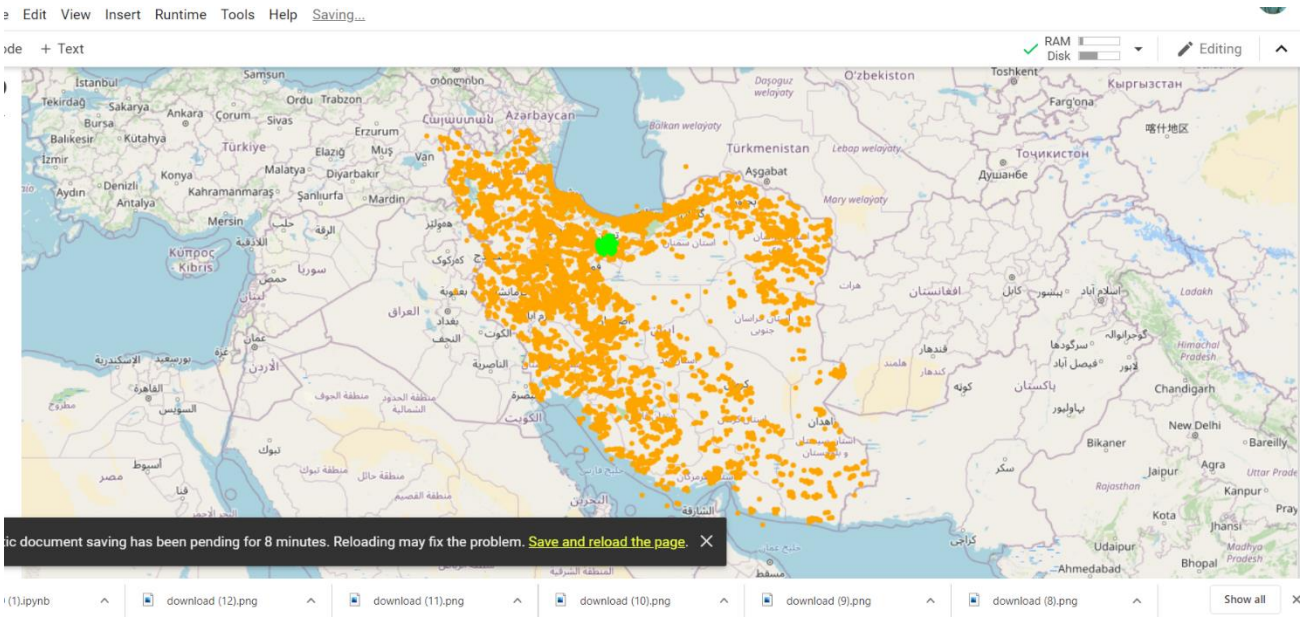
Data mining

First name: Sepehr **Last name: Asgarian** **Student_ID : 9531901**

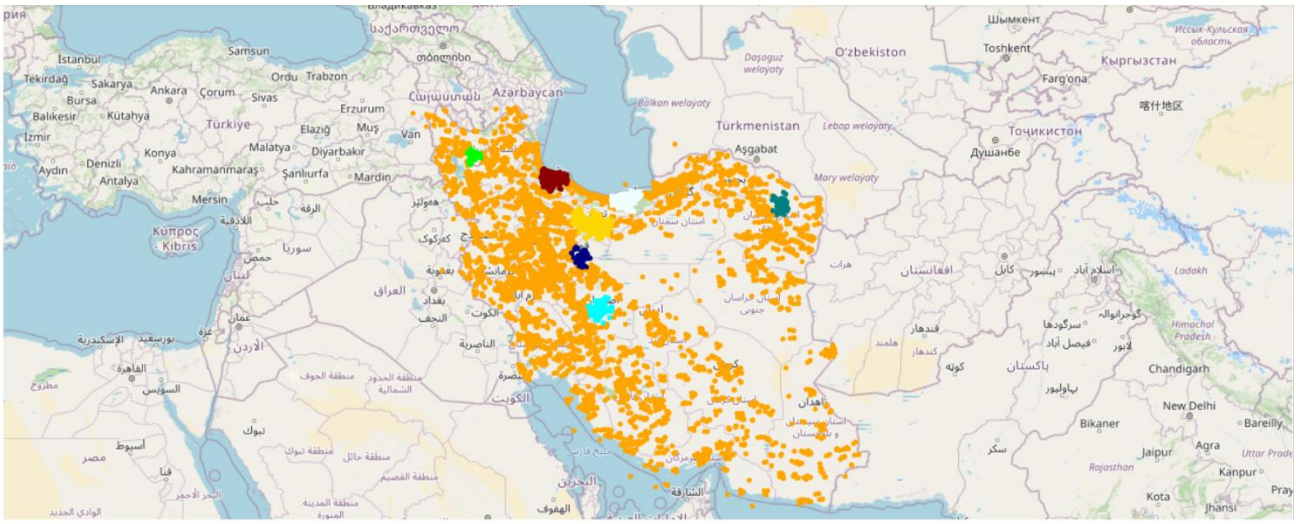


لازم به ذکر است برای انتخاب رنگ ها از سایت https://matplotlib.org/3.1.0/gallery/color/named_colors.html استفاده شده است.

```
predictor = DBSCAN(eps=0.2, min_samples=1000)
```




```
predictor = DBSCAN(eps=0.3, min_samples=300)
```



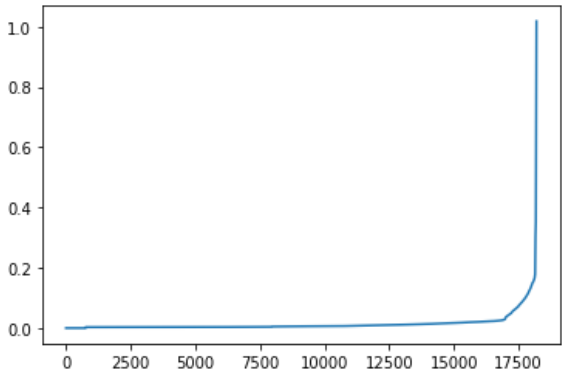
قسمت C

برای fine tune کردن من یک مقاله خوندم که از knn استفاده کرده بود برای این که بتونه بهترین eps را انتخاب کند

برای پیاده سازی این روش

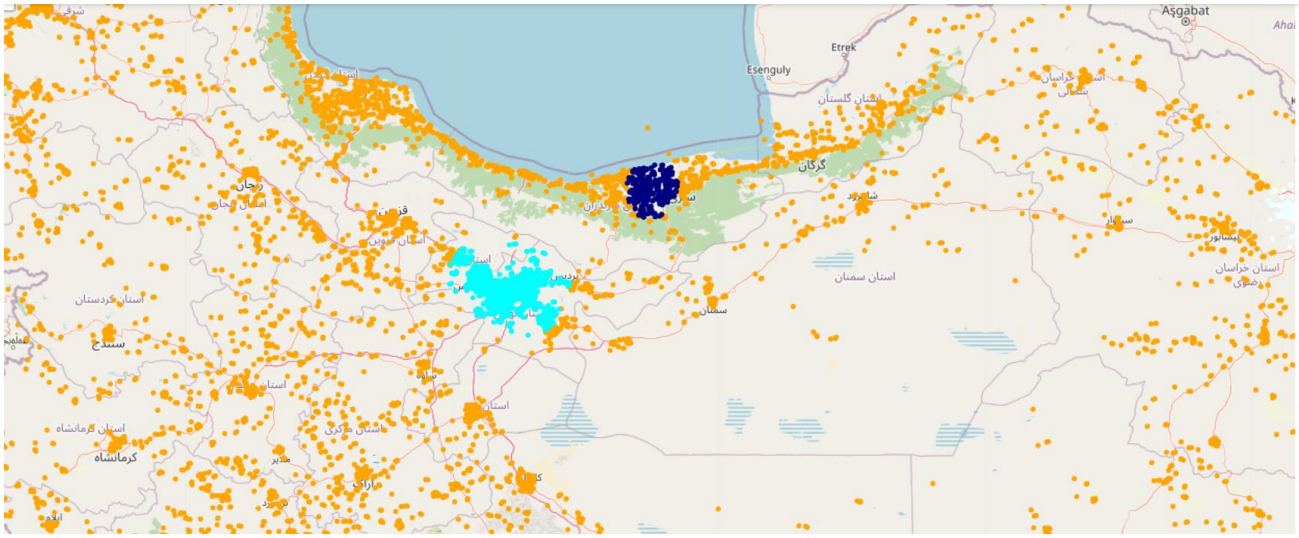
```
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(dataset)
distances, indices = nbrs.kneighbors(dataset)
distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.plot(distances)
```

در این مساله از k=2استفاده کردیم که با توجه به فاصله ای که از یک دیگر داشتند

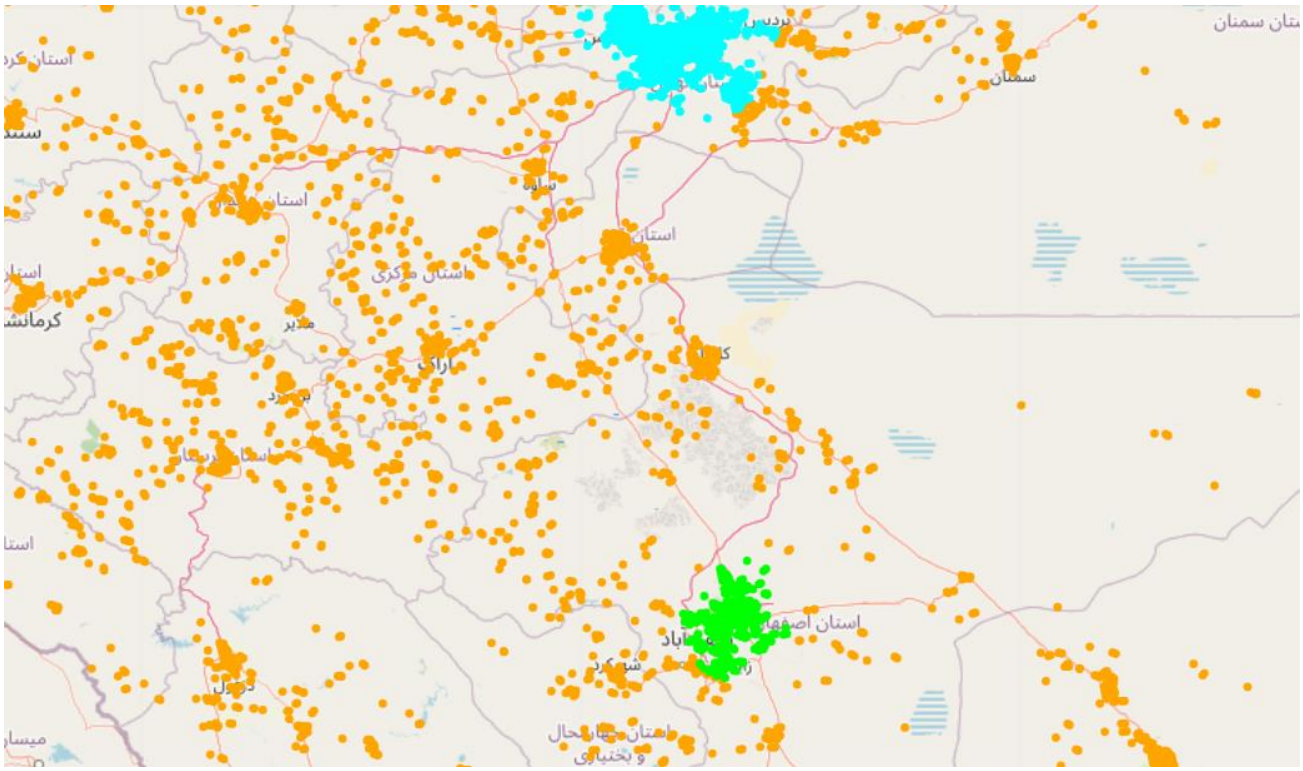


همانطور که در شکل نیز دیده میشود جایی که در واقع این نمودار شکسته حدود0.2

است و برای minsample نیز اعداد مختلف را که بارگزاری کردم 250 تا 400 است که 350 را انتخاب کرده و نمایش آن به صورت زیر میباشد



با این اعداد همانطور که مشاهده میشود cluster هایی برای تهران و گیلان در نظر گرفته شده است که با توجه به مشاهدات فعلی کاملا درست میباشد



همچنان همانطور که مشاهده میشود استان اصفهان نیز به صورت سبز رنگ کرده یک خشه که درست است و همینطور رنگ های زرد outlier هستند

D)

کاملا در موارد بالا قابل مشاهده و توضیح داده شده است

(3

در این سوال با استفاده از الگوریتمی که در سوال اول نوشتیم میایم و در واقع بعد های عکس را کاهش میدهیم و با استفاده این عمل در واقع داریم ویژگی های مهم تر عکس را نگه میداریم

در این سوال کاری که در ابتدا انجام دادم

```
img = image.imread(r'c.png')

img
plt.imshow(img)
plt.show()
```

است که خروجی آن دیتای داده شده است

حال ابتدا اولین کاری که انجام میدهیم اضافه کردن یک بعد دیگر به center ها است که در اینجا 3 بعد است

```
    for i in range(K):
        array.append([random.uniform(0, 1) for x in range(3)])
```

سپس در قسمت حساب کردن avg برای ساختن مراکز برای مرحلهی بعد iteration نیز تغیراتی را در جها سه بعدی کردن آن انجا داده شد

و باقی دقیقا همان کد kmeans میباشد

حال با k=16 و N=4

سنتر های بدست آمده را در image جدید نمایش میدهیم که در واقع حال فشرده آن میشود طبیعی است که هرچقدر میزان k زیاد تر باشد زمان اجرای آن طولانی تر میشود

بخش B را با این خط کد پیاده سازی کردم که در واقع uioتصویر جدید ما میباشد

```
uio[i,j] = comp_pixel_data[index]
```

نتایج به صورت زیر میباشد

ابتدا برای سایز کوچک حساب کردیم

*Figure 1real 200*200*

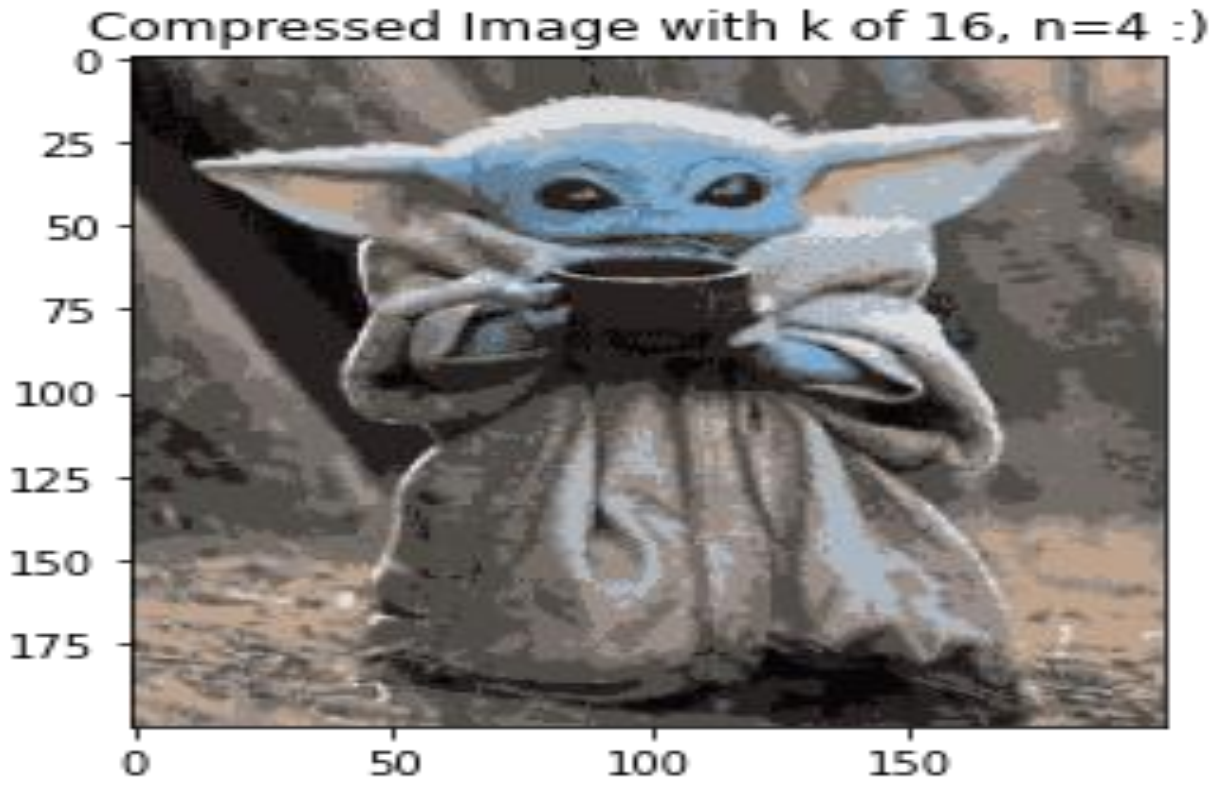
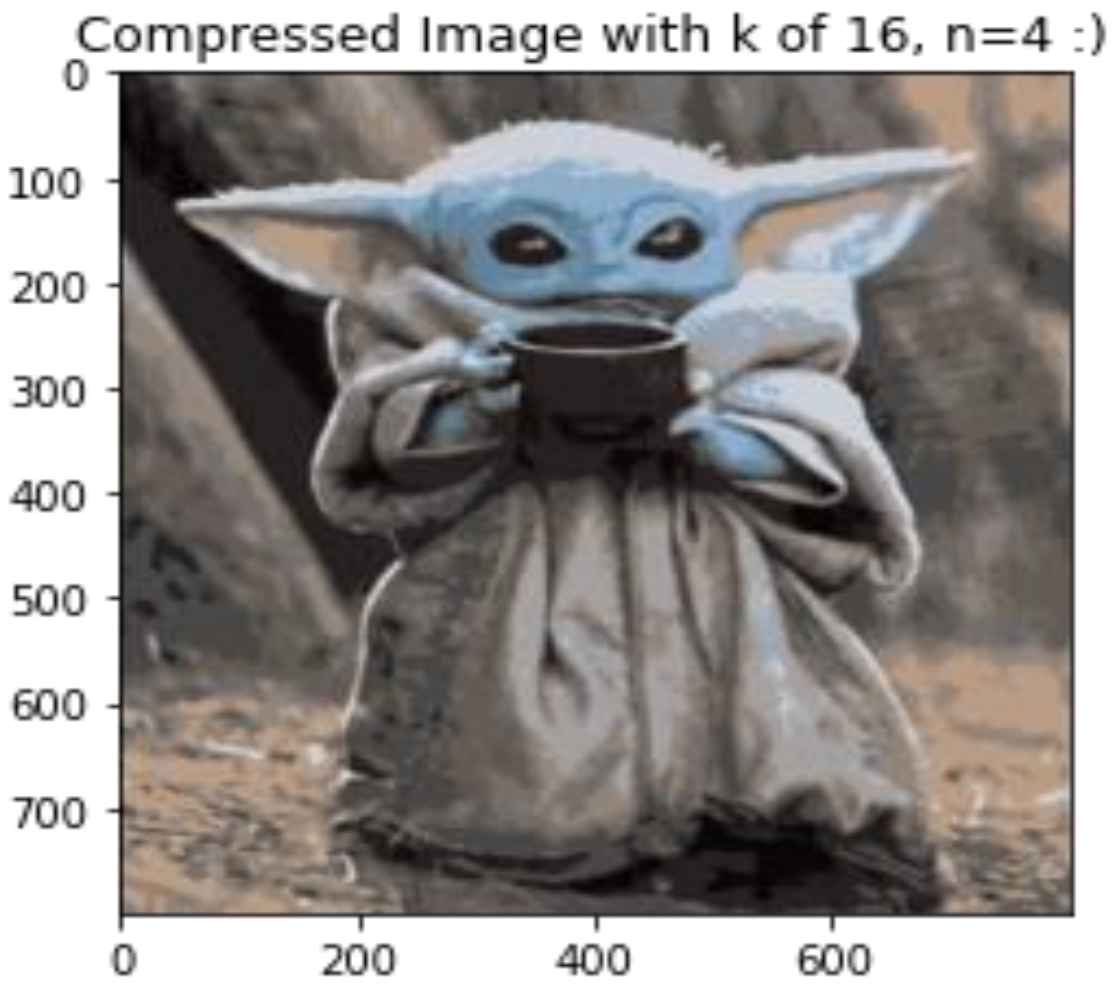


Figure 2سایز 200 در 200 با 16 خوشه

*Figure 3 k=256 size 200*200*

حال با سایز 800 در 800

*Figure 4real 800*800*



*800*800Figure 5*

تمامی کد این قسمت ها در کد موجود است

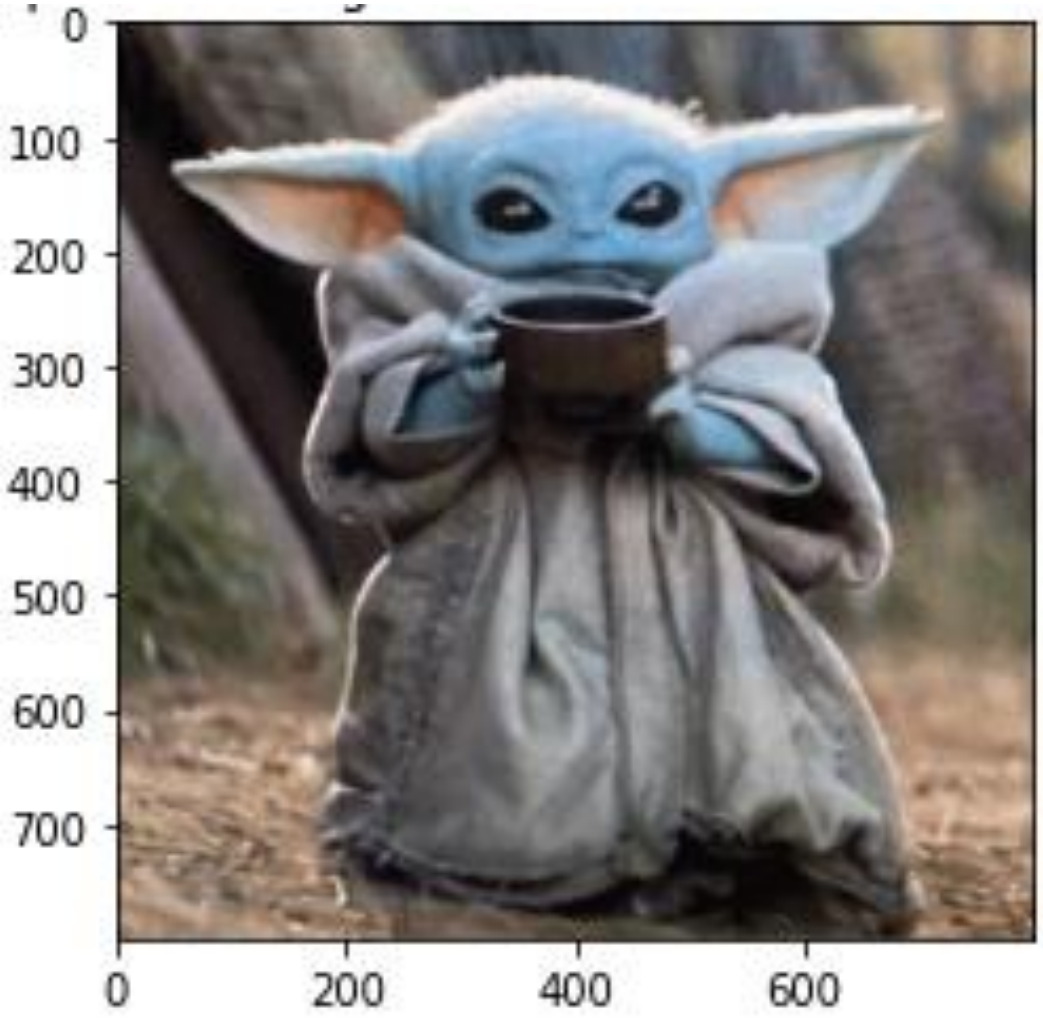
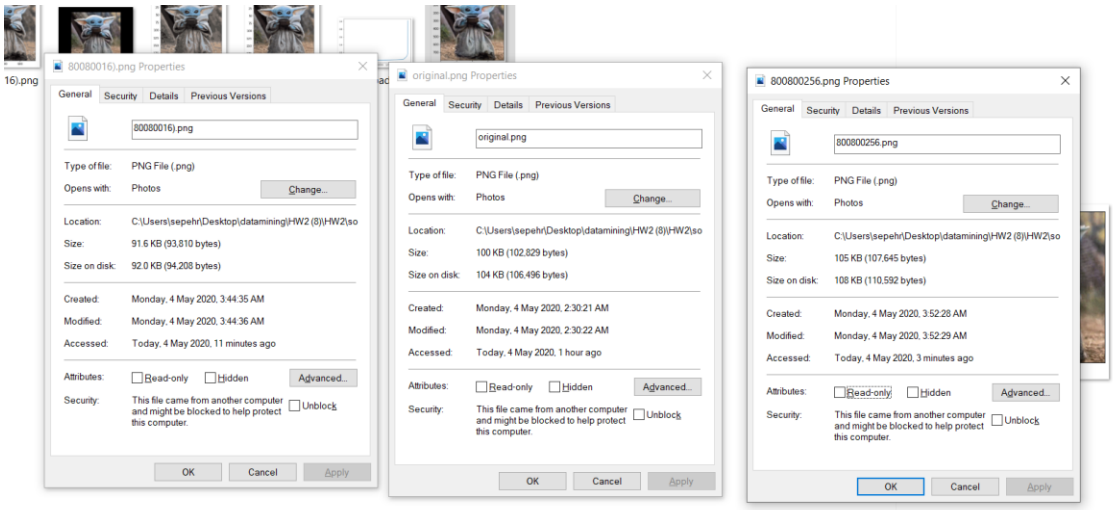


*Figure 6 800 * 800 *256*

In the name of God

Data mining

First name: Sepehr **Last name: Asgarian** **Stud ent_ID : 9531901**



اطلاعات ذخیره در مقایسه با اریجینال آنها مقایسه شده است