

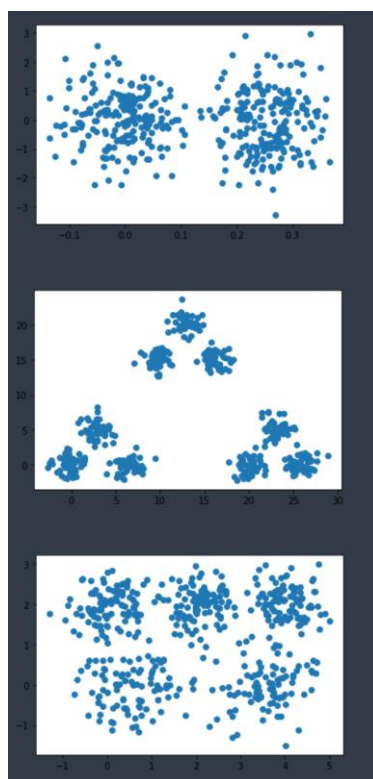
گزارش سوال اول:

	train_score	test_score
0	0.977391	0.958115

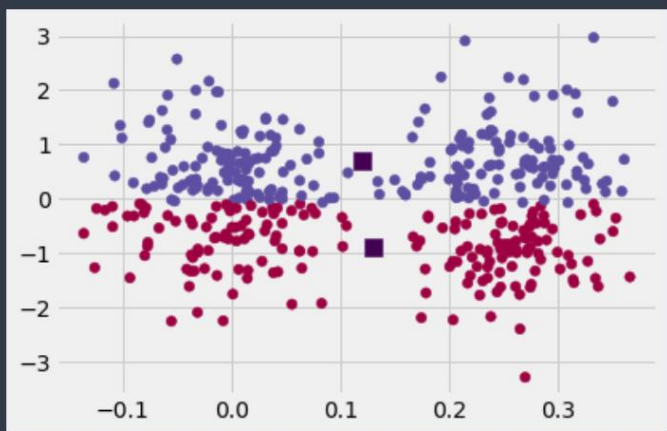
```
array([[ 57,   6],  
       [  2, 126]], dtype=int64)
```

طبق بررسی پارامترهای مختلف بهترین نتیجه مطابق جدول فوق به وجود آمد که در آن اولاً دیتا اسکیل شد. سپس با PCA به 30 متغیر کاهش پیدا کرد. (میزان بهینه) و در آخر بهترین اندازه برای لایه های مخفی شبکه عصبی به ترتیب 10 و 2 بودند. با این پارامترها ما به دقت 95% در تشخیص چهره فرد دست پیدا کردیم.

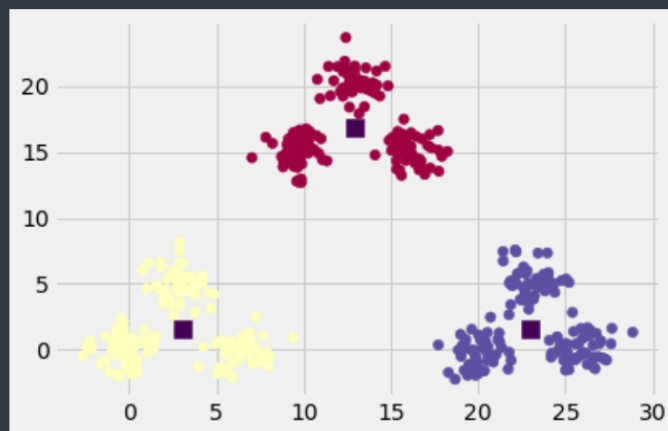
گزارش سوال دوم:



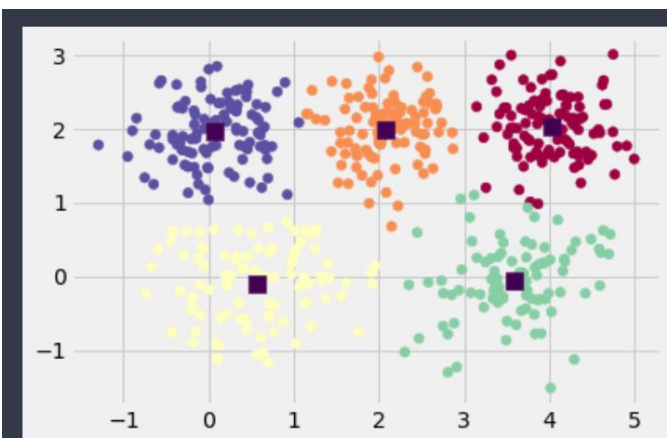
نمای کلی دادگان به صورت فوق است. پیش بینی ما برای K مناسب به ترتیب 2، 3 یا 9 و 5 است. حال الگوریتم k-means را با k حدسی خودمان روی آن پیاده میکنیم:



SSE: 146.8885600533513
silhouette_score: 0.5362906097948957



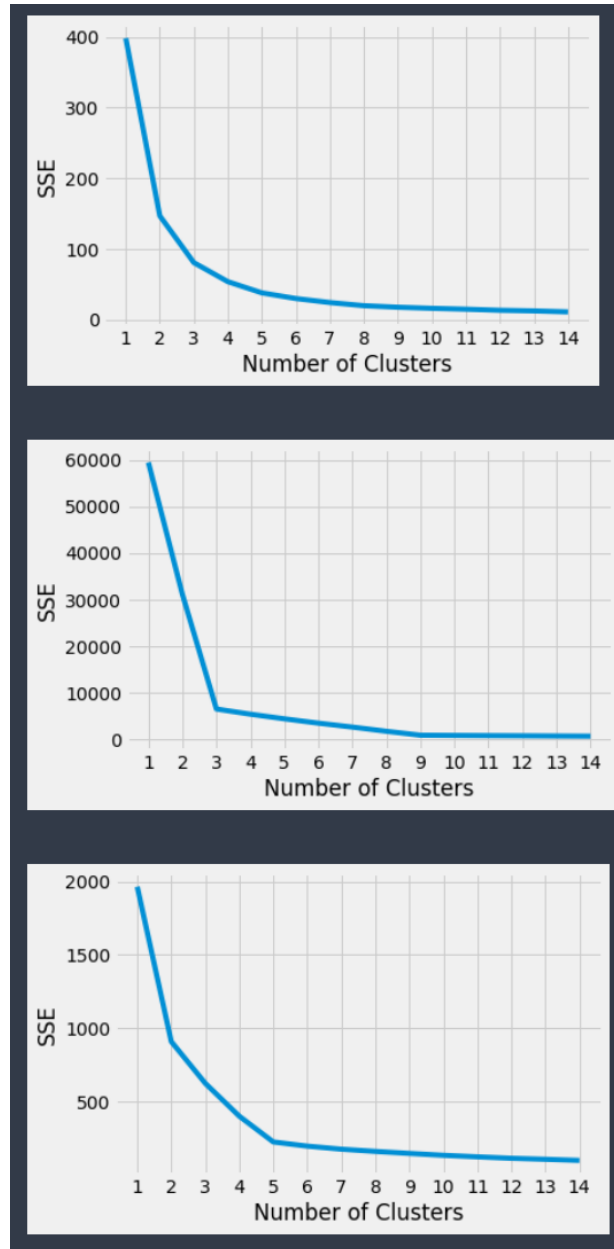
SSE: 6487.004526263209
silhouette_score: 0.7240379233766401



SSE: 222.36598776929108
silhouette_score: 0.5604142087131664

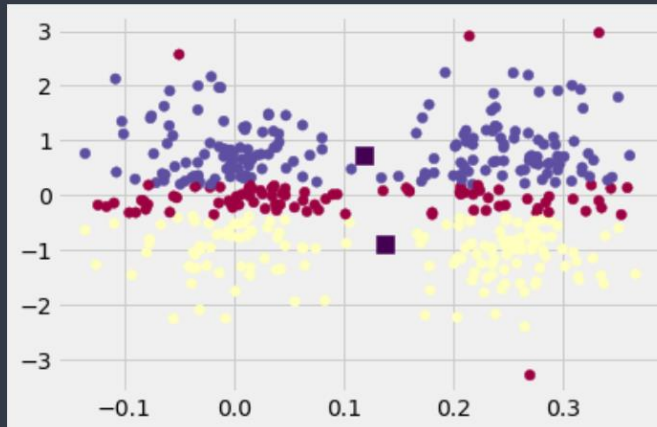
اسکورهای هر کلاسترینگ هم چاپ شده است که بعدا با DBSCAN مقایسه شود.

اما ما k حدسی خودمان را گذاشتیم. بیاید بررسی کنیم با تغییر k میزان SSE چه تغییری خواهد داشت. نمودار آن برای این سه دادگان به ترتیب در پایین آمده است:

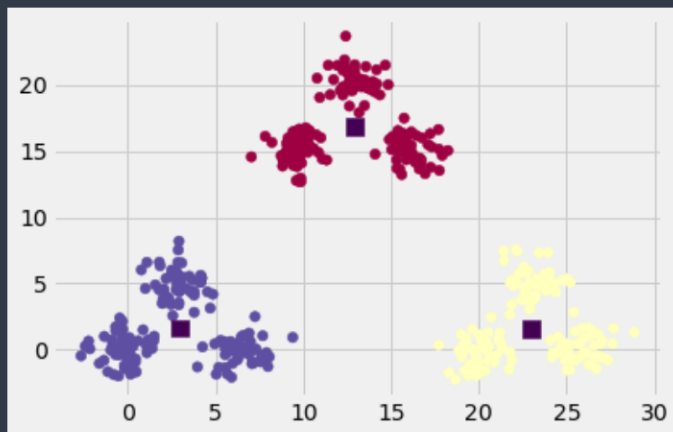


در نقاط زانویی ما بهترین کلاسترینگ را خواهیم داشت. برای داده اول نقاط 2 و 3، برای داده دوم 3 و 9 و برای دادگان آخر 2 و 5 بهترین هستند.

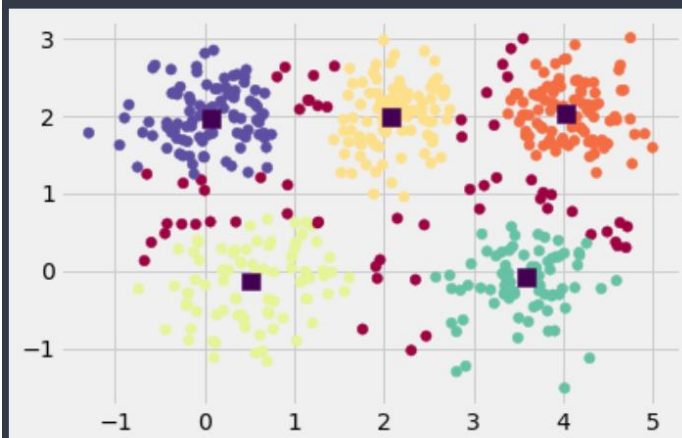
حال به بررسی الگوریتم fuzzy c-means میپردازیم. در نمودارهای اول و سوم رنگ قرمز نشان دهنده نقاطی هستند که به بیش از یک کلاستر از نظر احتمالی تعلق دارند:



SSE: 146.97353305008744
silhouette_score: 0.5362906097948957



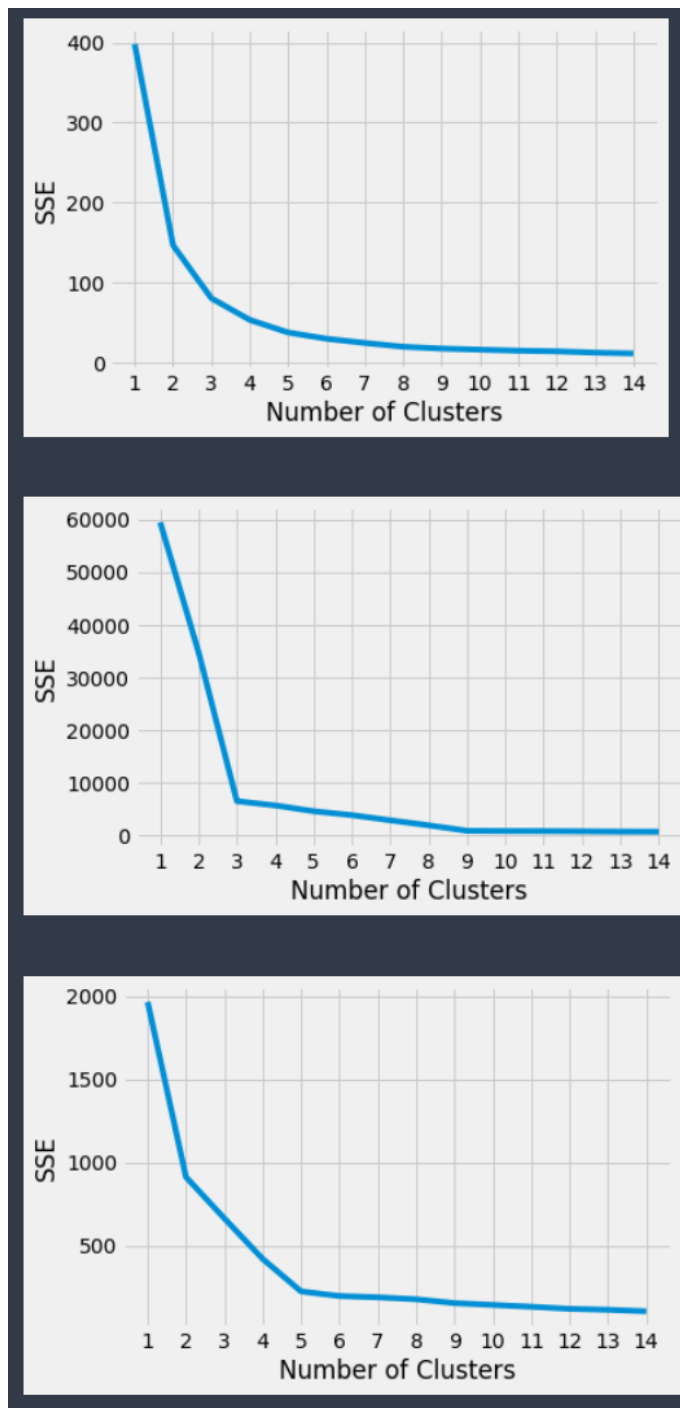
SSE: 6489.403251888046
silhouette_score: 0.7240379233766401



SSE: 222.85309578970023
silhouette_score: 0.5605130818892635

همانطور که مشخص است، عملکرد این الگوریتم خیلی مشابه با الگوریتم k-means است. اما نقطه قوت آن، مشخص کردن دیتایی است که نسبت به آن مطمئن نیستیم و آن را جزو چند دسته حساب کرده است.

نقاط زانویی همانند بخش قبل بدین شکل هستند:

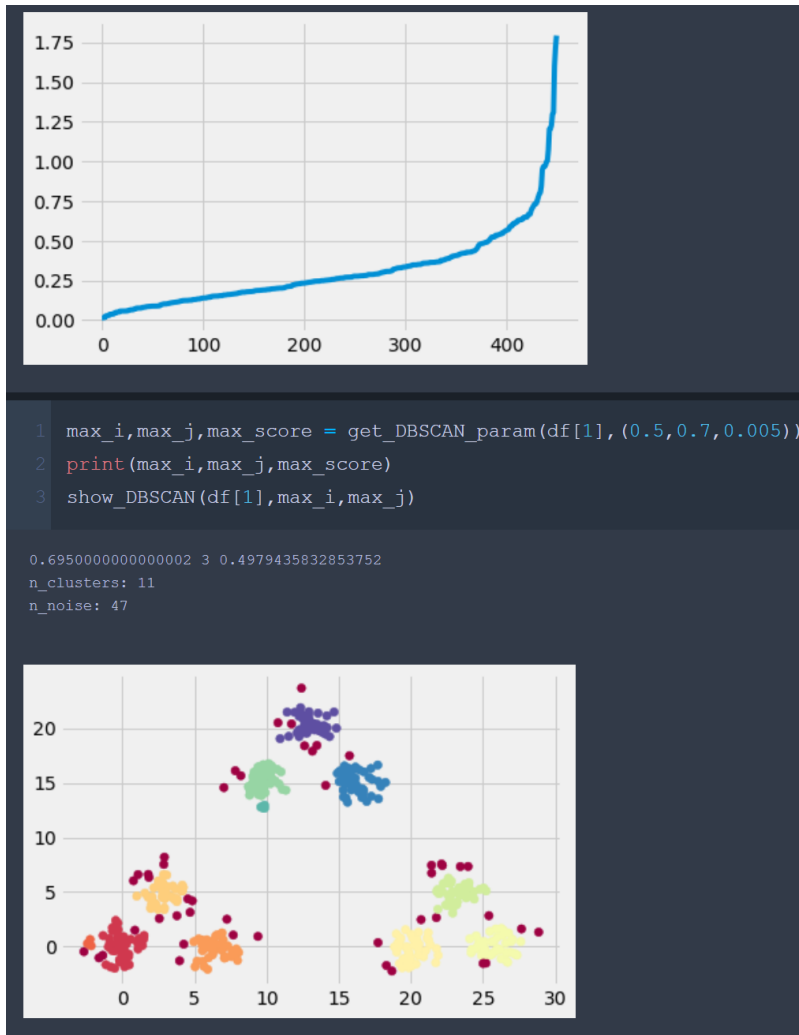


حال به سراغ الگوریتم DBSCAN میرویم. در این الگوریتم ما برای پیدا کردن پارامترها ابتدا نیاز به یک معیار داشتیم که در چه بازه ای اپسیلون را مقداردهی کنیم. رینج دادگان در اینجا مهم میشود. برای فهمیدن آن از الگوریتم نزدیکترین همسایگان استفاده میکنیم که ببینیم همسایگان به صورت میانگین چقدر با هم اختلاف دارند.

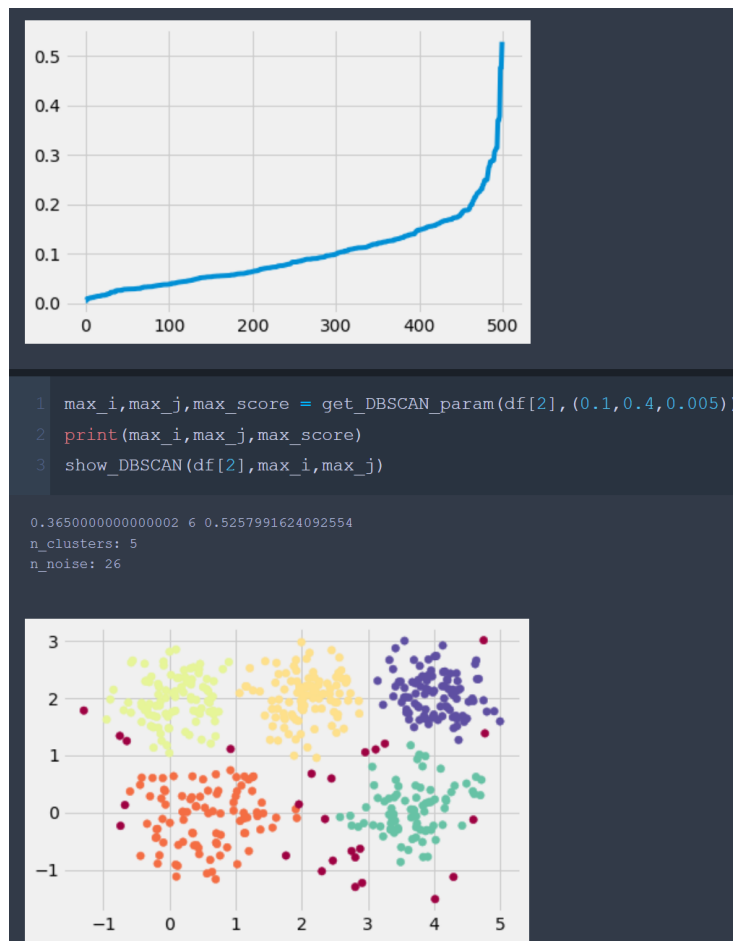
پس از بررسی باز هم نقاط زائیی را به عنوان یک بازه به حلقه فور میدهیم تا در همه نقاط زانویی و برای min_sample بین 1 تا 50 بهترین امتیاز این الگوریتم را بدست آورد.



همانطور که مشخص است نقاط زانویی تقریباً در بازه 0.1 تا 0.22 وجود دارد که توسط ما بررسی میشود. در تمامی این حالات امتیاز بیشتر از 0.41 نمیشود! دلیل آن سخت بودن این مثال برای الگوریتم های کلاسترینگ چگالی محور است. زیرا داده ها پراکندگی بالایی دارند.



در مثال بالا شاهد عملکرد بهتر الگوریتم برای مثال بعدی هستیم که به صورت منظمی کلاسترینگ را انجام داده است.(نقاط قرمز متعلق به هیچ کلاستری نیستند)



در دادگان آخر نیز DBSCAN به خوبی عمل میکند و امتیاز بالایی میگیرد.

در نهایت الگوریتم های k-means و fuzzy-c-means بهترین عملکرد را داشتند اما چون fuzzy-c-means دیتای بیشتری به ما میدهد و یکسری داده را پرت و خارج هرگونه کلاستر گزارش میکند بهتر از k-means است.