به نام خدا

تمرین سری 5 کلاسترینگ

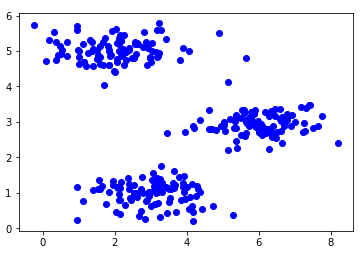
حسین غلامی 97123021

در ابتدا به دلیل این که فایل اطلاعات .mat است ، توسط matlab باز کرده ، ماتریس (جدول) را در excel کپی کرده و با فرمت csv ذخیره میکنیم. در پوشه قرار گرفت.

سوال 1

الف ) در این بخش رسم داده ها را مد نظر دارد که به شرح زیر است.

همانطور که مشاهده میشود ، به سه بخش تقسیم میشوند:

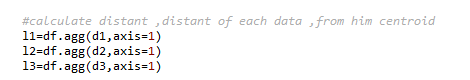
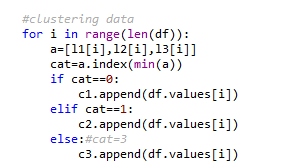


ب) در این بخش از ما الگوریتم kmeans را میخواهد که بدون کتابخانه ، پیاده سازی کنیم.(نقطه شروع رندم)

برای این کار ابتدا : 3 نقطه رندم به عنوان centroid در نظر میگیریم

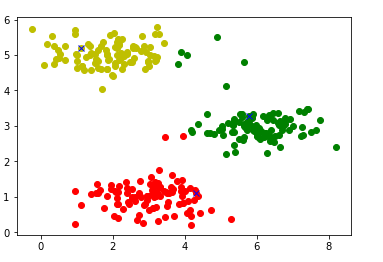


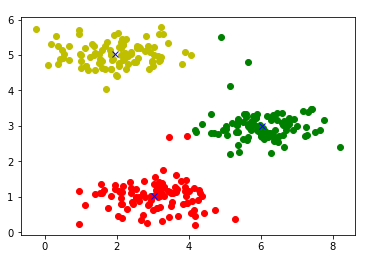
و centroid جدید را مقدار دهی اولیه میکنیم حال وارد الگوریتم میشویم ، برای این کار از یک حلقه while که شرط توقف آن تغییر نکردن cs ها باشد (و همچنین ittration ) پیاده سازی میکنیم

حال فاصله همه ی نقاط را تا cs ها محاسبه میکنیم

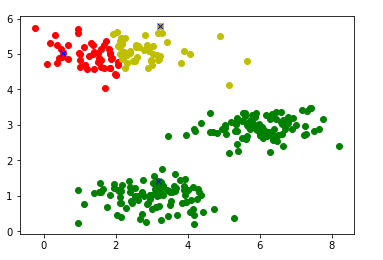
و کلاسترینگ اولیه را انجام میدهیم

حال مرکز جدید را آپ دیت کرده و مجدد همین روند را اجرا میکنیم

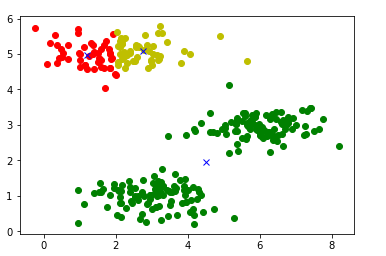
و در پایان هر بار رسم میکنیم برای مثال :

که در 3 مرحله به شکل زیر ختم شد:

ولی نقاط همواره به شکل فوق تقسیم نمیشود برای مثال :



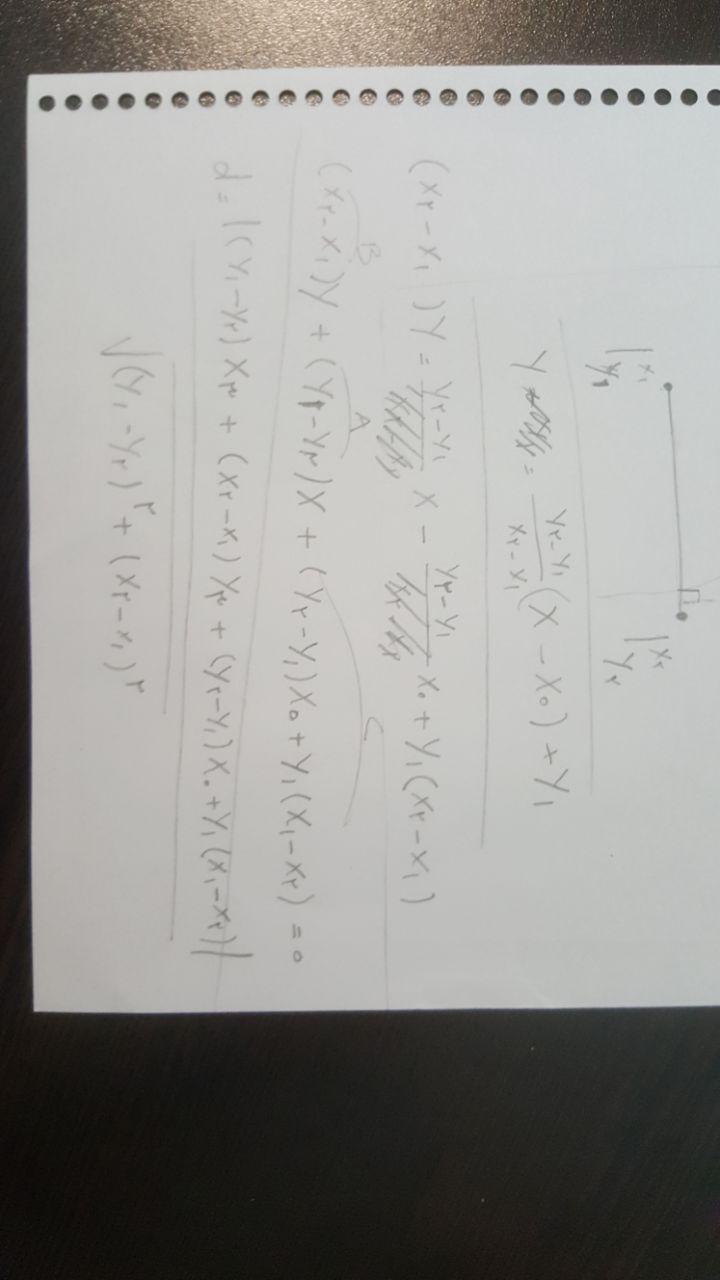
که با 5 بار اجرا به شکل زیر تبدیل شد

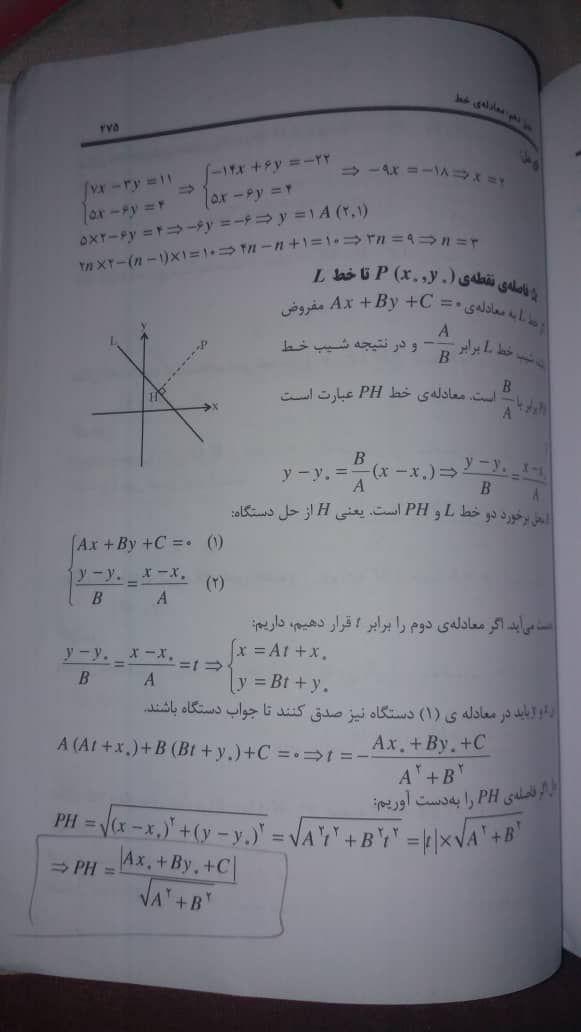


برای حل مشکل به پیاده سازی بخش پ ، پرداخته شد

پ)برای رفع مشکل فوق اگر فاصله دور ترین نقاط را از هم محاسبه کنیم ، میتوانیم مشکل را حل کنیم

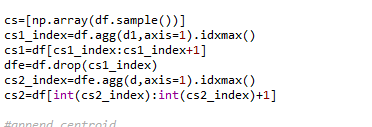
به این شکل که ابتدا نقطه ای رندم انتخاب نموده ، فاصله ی همه ی نقاط را تا آن حساب میکنیم دور ترین نقطه را نقطه شروع دوم در نظر میگیریم و آن را حذف میکنیم ، - تا اینجا 2 نقطه داریم ، با نقطه فوق خطی تشکیل داده وفاصله همه نقاط را تا آن خط محاسبه میکنیم ، دورترین نقطه را نقطه سوم می نامیم ، و الگوریتم قبلی را شروع میکنیم

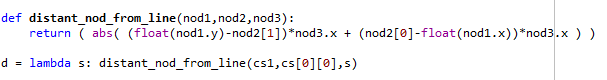
محاسبه تابع برای فاصله نقطه از خط :



با توجه به این که مخرج کسر ثابت است :

تنها صورت پیاده سازی میشود.

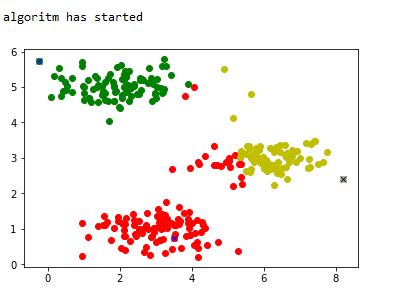




که برای مثال:

نقطه ای که در ناحیه قرمز است به عنوان

نقطه رندم انتخاب شده، نقطه ای که در ناحیه سبز



است به عنوان دور ترین نقطه انتخاب شده

و نقطه ای که در ناحیه زرد است به عنوان

دورترین نقطه نسب به خط است.

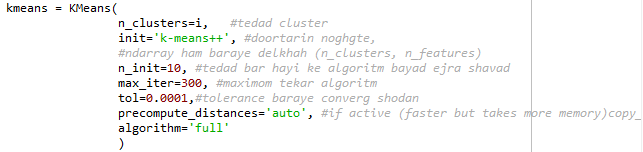
بقیه الگوریتم مشابه قبل است.

سوال2

در این بخش به دلیل تعدد ابعاد اطلاعات ، و تنوع تعداد کلاسترینگ از 

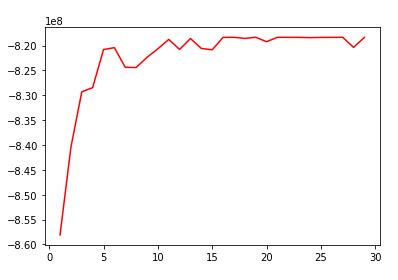
استفاده شد .

الف)تابع هندلر kmeans به شکل زیر است که توضیحات آن نوشته شد:

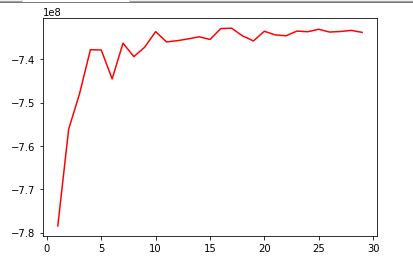


با مقدار دهی n\_init 10 بار اجرا شده و بهترین نتیجه را اعلام میکند ، init وقتی به صورت k-means++ است مقدار دهی اولیه با همان معیار دورترین عمل میکند. Max\_itter تعداد دفعات اجرای الگوریتم را محدود میکند

ب) در این تابع تابعی تحت عنوان score دارد که مجموع فواصل را اعلام میکند ، با توجه به آنچه مد نظر صورت سوال است ، میزان تغییرات score را به برای هر بار iteration میخواهد پس max\_itteration را هر بار یک واحد اضافه نموده و منحنی را رسم میکنیم:

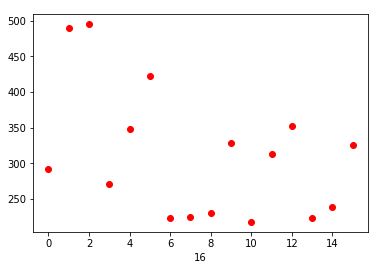
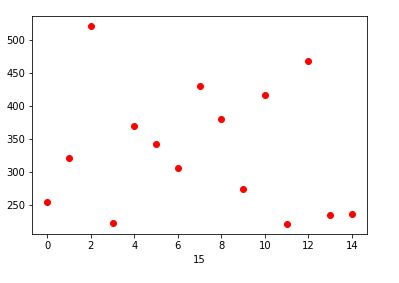
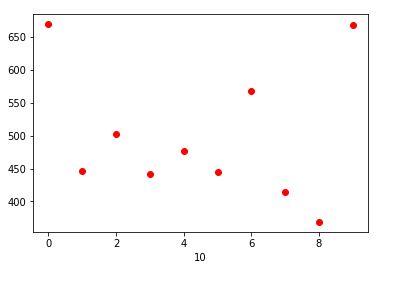
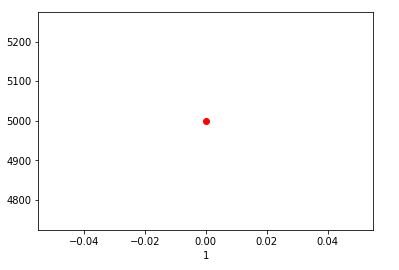
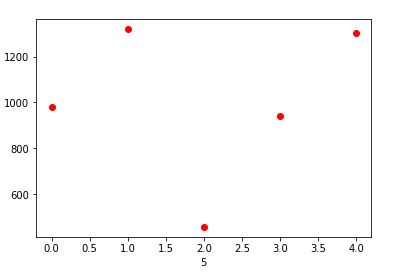
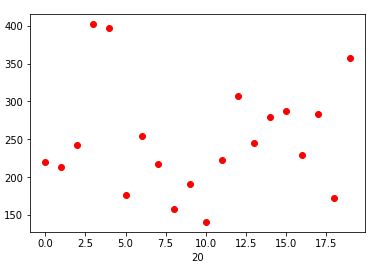


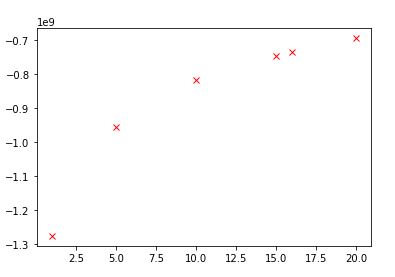
پ) همین منحنی را به ازای k=16 میخواهد



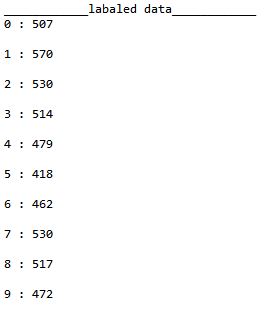
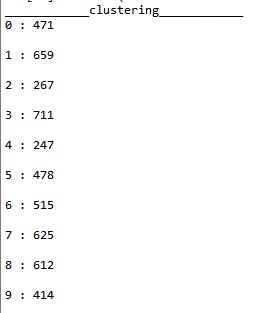
K=16 k=10

همانطور که مشاهده میشود میزان روند کلی نزولی(کاهش فاصله = افزایش امتیازscore) است و با افزایش میزان کلاستر عدد همگرا شده ، کاهش میابد.

ت) اجرای الگوریتم به ازای k=1, 5, 10, 15, 16,20 و نمونه های موجود در آن

میزان score یا همان قرینه فاصله within distant به ازای k=1, 5, 10, 15, 16,20 

و اما در نهایت بررسی دقیق به ازای k=10



بیشترین میزان اطلاعات در داده های label خوره متعلق به 1 است پس نتیجه میگیریم که کلاستر 3 متعلق به عدد 1 است

و به همین ترتیب پیش میرویم