به نام خدا

پروژه درس داده‌کاوی: فصل ۸، خوشه‌بندی

استاد درس: دکتر رضا رمضانی

مهلت تحویل: ۱۵ تیر ۱۴۰۰

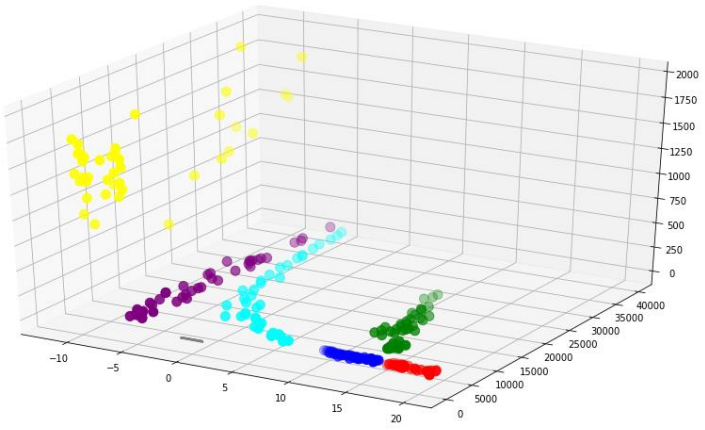
پروژه بصورت ۱ نفره یا ۲ نفره است

در این پروژه، یک دیتاست شامل اطلاعات حدود ۲۰۰ ستاره و ویژگی‌های آن‌ها در اختیار شما قرار گرفته است. هدف، خوشه‌بندی کردن این ستاره‌ها در ۶ دسته است. همچنین کلاس و دسته‌ی صحیح هر ستاره در ستون type با اعداد ۰ تا ۵ مشخص شده است که فعلا برای فرآیند خوشه‌بندی با این ستون کاری نداریم اما هنگام ارزیابی برای ما مهم است. ۴ ستون اول مقادیر عددی پیوسته هستند که به ترتیب دما، درخشش، شعاع و قدر مطلق ستاره است. دو ستون بعدی، رنگ و نوع خاص ستاره است که هر دو کمیت‌هایی nominal هستند. حال می‌خواهیم این داده‌ها را بدون استفاده و توجه به ستون type خوشه‌بندی نماییم. مراحلی که در ادامه مشخص شده‌اند را انجام دهید و نتایج کار را به صورت کامل مستند کنید. استفاده از کتابخانه‌هایی مانند scikit-learn مجاز است.

۱. ابتدا داده‌ها را بدون هیچگونه پیش پردازشی و بدون استفاده از داده‌های ستون type خوشه‌بندی کنید. خوشه‌بندی را به کمک یک الگوریتم Partitional و یک الگوریتم Hierarchical انجام داده و تعداد خوشه را ۶ در نظر بگیرید. در نهایت دقت خوشه‌بندی خود را حساب کنید.

> **نحوه محاسبه دقت خوشه‌بندی:** راه‌های زیادی برای ارزیابی دقت خوشه‌بندی وجود دارد. برای این مساله، بعد از خوشه‌بندی، ببینید برچسب (type) اکثریت داده‌های موجود در یک خوشه چیست. سپس برچسب تمام داده‌های آن خوشه را همین برچسب اکثریت در نظر بگیرید. این کار را برای تمام خوشه‌ها انجام دهید. لذا تمام داده‌های خوشه‌بندی شده، یک برچسب خوشه دارد. در نهایت ببینید چند درصد از کل داده‌ها برچسب صحیح دریافت نموده‌اند.

> شمای کلی داده‌ها بر اساس سه بعد Temperature, A_M, R به شکل زیر است که بهترین نمایشی است که می‌توان از داده‌ها نشان داد. هر رنگ نشان‌دهنده یک طبقه در داده‌ها می‌باشد و بر اساس type که در دیتافریم است نمایش داده شده است.



در ابتدا همه‌ی ستون‌ها (۵ ستون) به الگوریتم k-means داده شده. پس از انجام خوشه بندی، طبق معیاری ارزیابی صورت سوال تعداد هر برچسب در خوشه‌ها به شکل زیر می‌باشد که در هر تاپل عداد اول نماینده برچسب و عدد دوم نماینده تعداد آن برچسب در خوشه است.

```
([(4, 10), (5, 8)],
 [(0, 40), (1, 40), (2, 40), (3, 34)],
 [(4, 3), (5, 3)],
 [(4, 18), (5, 14), (3, 5)],
 [(5, 4), (4, 2)],
 [(5, 11), (4, 7), (3, 1)])
```

چون لیبل تکراری داریم، از بزرگ‌ترین خوشه شروع می‌کنیم و برچسب می‌زنیم.

خوشه با ۱۵۴ داده برچسب ۰

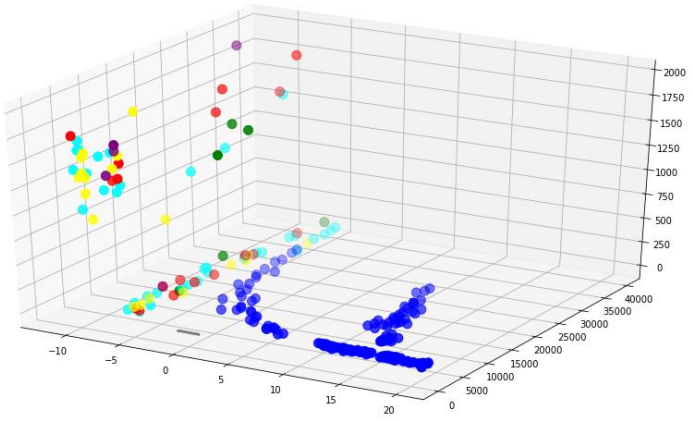
خوشه با ۳۷ داده برچسب ۴

خوشه با ۱۹ داده برچسب ۵

خوشه با ۱۳ داده ۵؟۴

چون این روش هم اعداد دقیقی به ما نمی‌دهد در ادامه سعی می‌شود بر اساس مصور سازی و با توجه به همین اعداد تولید شده کیفیت خوشه بندی ارزیابی شود.

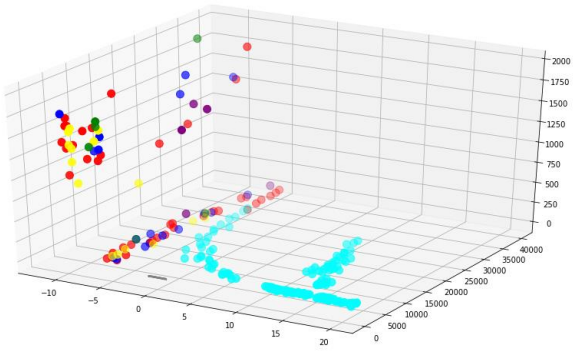
پس از خوشه بندی k-means دسته بندی که در قسمت قبل از داده‌ها نمایش داده شد به شکل زیر تغییر می‌کند.



پس از k-means همه‌ی ستون‌ها (۵ ستون) به الگوریتم AgglomerativeClustering که یک الگوریتم Hierarchical است داده شد. پس از انجام خوشه بندی، طبق معیاری ارزیابی صورت سوال تعداد هر برچسب در خوشه‌ها به شکل زیر می‌باشد که در هر تاپل عداد اول نماینده برچسب و عدد دوم نماینده تعداد آن برچسب در خوشه است. همانطور که مشاهده می‌شود این الگوریتم تعداد خوشه‌ها را یکی کمتر از تعداد اصلی خوشه‌ها در نظر می‌گیرد.

```
([(4, 18), (5, 17), (3, 6)],
 [(4, 10), (5, 7)],
 [(5, 4), (4, 2)],
 [(0, 40), (1, 40), (2, 40), (3, 34)],
 [(5, 12), (4, 10)],
 [])
```

پس از خوشه بندی AgglomerativeClustering دسته بندی که در قسمت ابتدا بر اساس برچسب‌های اصلی از داده‌ها بود به شکل زیر تغییر می‌کند.



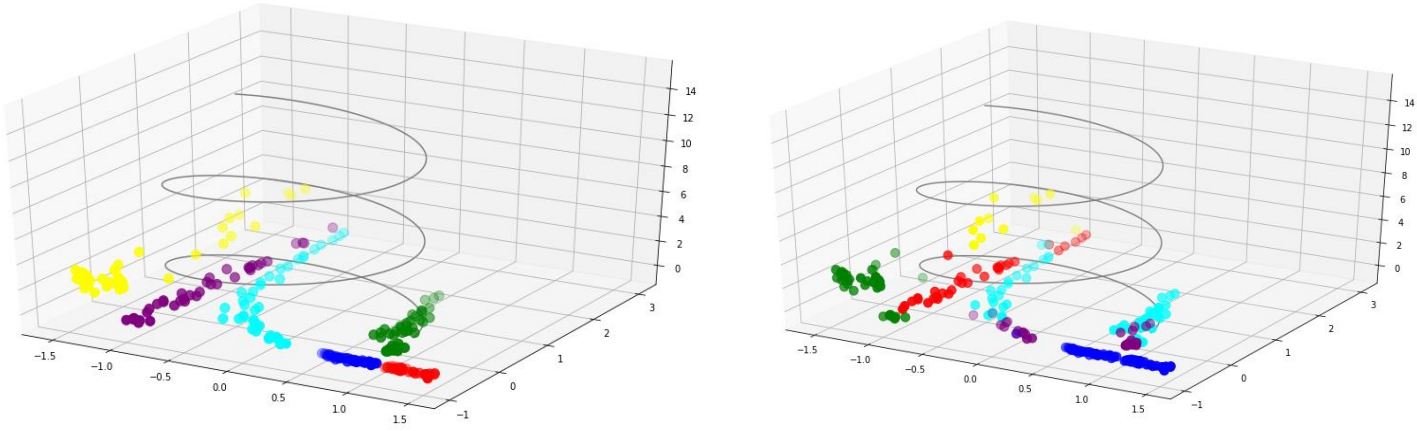
۲. در گام بعدی داده‌ها را پیش پردازش کنید (مانند گسسته سازی، مقیاس‌بندی و ... - با مکتوب کردن دقیق پیش پردازش) و مجددا خوشه‌بندی بیان شده در مرحله ۱ را انجام دهید و دقت را حساب کنید. بایستی سعی کنید پیش پردازش را به نحوی انجام دهید که دقت خوشه‌بندی افزایش یابد.

از مواردی که می‌توان برای پیش پردازش استفاده کرد، scale کردن داده‌هاست. چون الگوریتمی مثل k-means بر اساس فاصله کار می‌کند (یا اقلیدسی یا منهتن که اینجا اقلیدسی در نظر گرفته شده است) و اعداد در اسکیل‌های بسیار بزرگ

وجود دارند و با عث میشود تاثیر آن‌ها بسیار زیاد شود در نتیجه از معیارهای scaling می‌توان استفاده کرد که در یک رنج متعادل قرار بگیرند. که در اینجا از standard scaler استفاده شده است.

همچنین داده‌های categorical که شامل color و Spectral_Class می‌شود نیز به عدد تبدیل شده‌آند تا بتوان آن‌ها را به ورودی الگوریتم‌ها داد.

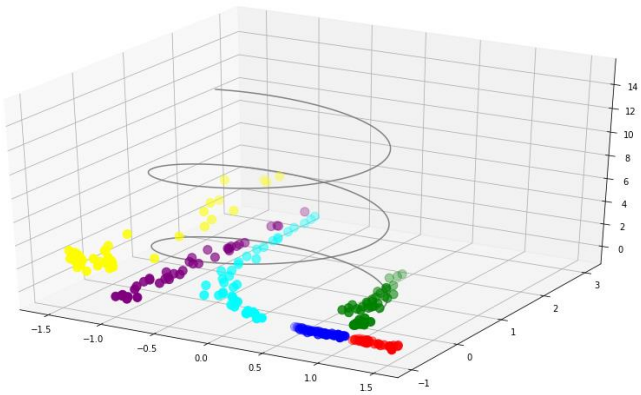
پس از scale کردن داده‌ها برچسب‌های داده‌ی اصلی در سمت چپ و خروجی K-means در سمت راست نمایش داده شده است.

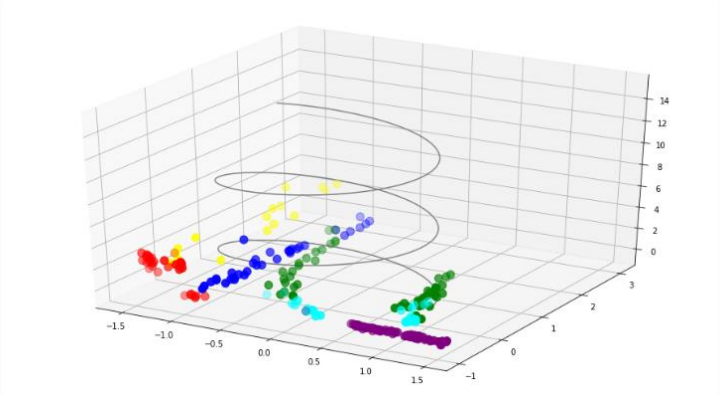


معیار دقت‌را دوباره برای هر خوشه بدست می‌اوریم. مشاهده می‌شود که خوشه‌ها تفکیک پذیری بهتری پیدا کرده‌اند و تعداد برچسب‌های یکسان کمتر و تعداد نقاط در هر خوشه متعادل‌تر شده است. اما با توجه به تصاویر بالا بازم هم با واقعیت تفاوت دارد. اما با توجه به شکلی که در سوال یک بدون پیش پردازش و scale کردن انجام شد بسیار بهتر است.

```
([(4, 28), (3, 5), (5, 1)],
 [(0, 40), (1, 40), (3, 1)],
 [(5, 29), (4, 9)],
 [(2, 27), (3, 21), (4, 2)],
 [(2, 13), (3, 13)],
 [(5, 10), (4, 1)])
```

پس از scale کردن داده‌ها برچسب‌های داده‌ی اصلی در سمت چپ و خروجی AgglomerativeClustering در سمت راست نمایش داده شده است. چون داده‌ها scale شده‌اندکیفیت خروجی خیلی بهتر از آن چیزی که در سوال اول نمایش داده شد، شده است.





```
([(5, 23), (4, 9)],
 [(4, 29), (3, 5), (5, 1)],
 [(2, 27), (3, 21), (4, 2)],
 [(2, 13), (3, 13)],
 [(0, 40), (1, 40), (5, 16), (3, 1)],
 [])
```

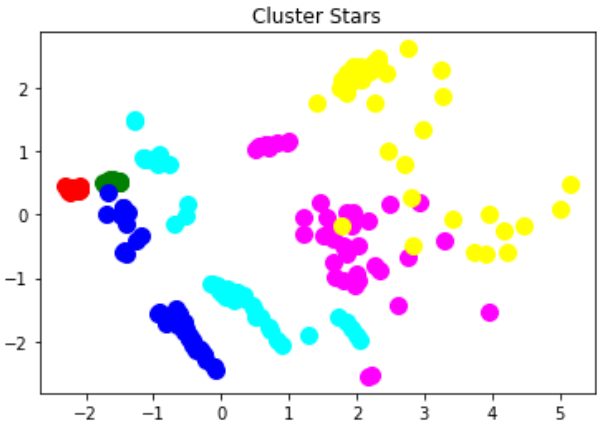
۳. با استفاده از روش PCA ابعاد داده‌ها (به غیر از type) را به ۲ بعد کاهش دهید (در صورت نیاز تمام داده‌های خود را عددی کنید) و آن‌ها را ترسیم کنید. سپس، با کمک الگوریتم DBScan و دو نوع الگوریتم قبلی این داده‌های دوبعدی را خوشه‌بندی کنید و نتیجه‌ی خوشه‌بندی‌ها را بصورت مصور نشان دهید (در هر تصویر، هر خوشه با یک رنگ منحصربه‌فرد مشخص شود). خوشه‌های بدست آمده را تحلیل کنید.

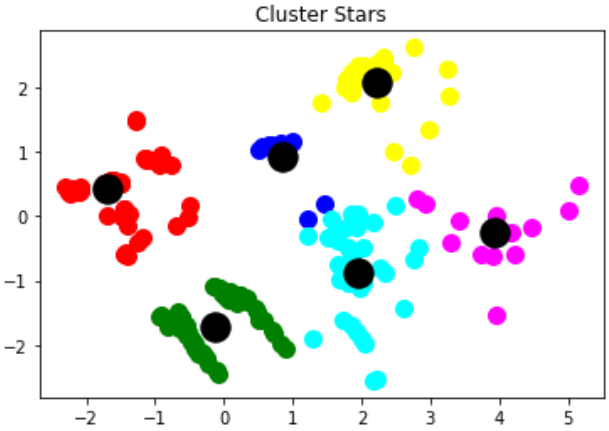
پس از کاهش به ۲ بعد هر ۳ الگوریتم اجرا شدند. با توجه به این از standard scaler برای scale کردن داده‌ها استفاده شده و داده‌ها اکثرا بین رنج‌های -۲ تا ۳ قرار می‌گیرند، مقدار eps=0.5 در نظر گرفته شد.(مقدارهای مختلفی امتحان شد که به نظر 0.5 عدد خوبی برای eps بود.)

در شکل زیر برای درک بهتر نمودار سمت چپ نماینده‌ی لیبیل‌های اصلی داده‌ها و سمت راست نماینده خروجی الگوریتم است که با یک دیگر مقایسه می‌شود.

۱. K-means

الگوریتم K-means برمبنای فاصله عمل می‌کند. همانطور که در تصاویر نقاط مشکی نشان دهنده مرکزهای دسته هستند و اگر با فاصله‌ی خوبی نقاط از یک دیگر قرار گرفته باشند الگورتیم مرزها را تشخیص داده است اما برای جاهایی که داده‌ها بسیار به یک دیگر نزدیک بوده اند الگوریتم آن‌ها را یک خوشه در نظر گرفته است.



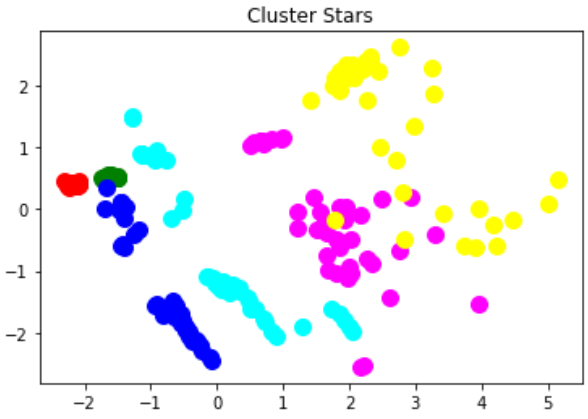


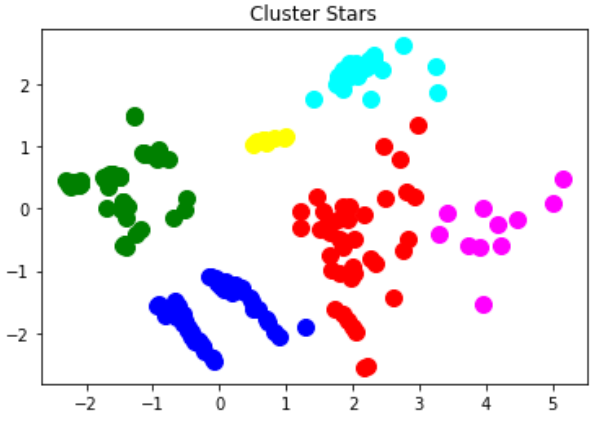
در شکل زیر هم تعداد هر نوع برچسب برای هر خوشه با توجه به برچسب‌هایی که از الگوریتم k-means بدست امده است قابل مشاهده است که اشتراک و همپوشانی بسیاری و جود دارد که به نادرستی تشخصی داده شده است و دچار برچسب تکراری برای هر خوشه می‌شود.

```
([(0, 40), (1, 40), (3, 14), (2, 13)],
 [(2, 27), (3, 20)],
 [(4, 11)],
 [(4, 26), (3, 6), (5, 2)],
 [(5, 10), (4, 3)],
 [(5, 28)])
```

۲. AgglomerativeClustering

این الگوریتم هم پس از کاهش بعد و ترسیم آن عملکردی مشابه با k-means را نشان می‌دهد. که فقط در تعدادی از نقاط زرد و قرمز متفاوت هستند.



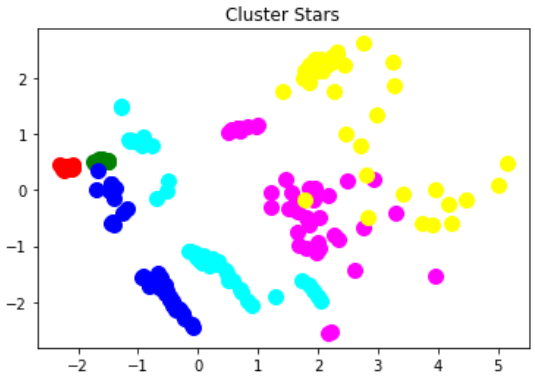


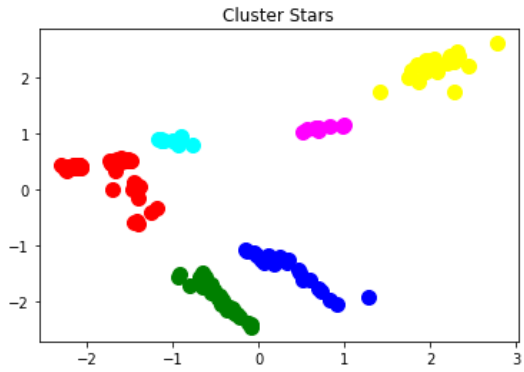
در شکل زیر هم تعداد هر نوع برچسب برای هر خوشه با توجه به برچسب‌هایی که از الگوریتم AgglomerativeClustering بدست امده است قابل مشاهده است که اشتراک و همپوشانی بسیاری و جود دارد که به نادرستی تشخصی داده شده است و دچار برچسب تکراری برای هر خوشه می‌شود.

```
([(4, 29), (5, 6), (3, 5)],
 [(0, 40), (1, 40), (3, 14), (2, 13)],
 [(2, 27), (3, 21)],
 [(5, 25)],
 [(4, 11), (5, 9)],
 [])
```

۳. DBscan

به دلیل این که الگوریتم DBscan مبتنی بر چگالی است، همانطور که از خروجی مشخص است(سمت راست) داده‌های قرمز، سبز، بخشی از آبی کم‌رنگ و پررنگ که در سمت چپ به صورت مجزا هستند را به صورت یک خوشه در نظر گرفته است. در این الگوریتک ۳ خوشه‌ی آبی پررنگ و کم‌رنگ صورتی و زرد تا حدودی به درستی دسته بندی شده اند.





در شکل زیر هم تعداد هر نوع از برچسب‌ها در خوشه‌ها مشاهده می‌شود که خوشه‌ی ابتدایی از سه نوع برچسب تشکیل شده و بازهم شاهد هم‌پوشانی در خوشه‌ها هستیم.

```
([(0, 40), (1, 40), (2, 13)],
 [(2, 27)],
 [(3, 21)],
 [(3, 9)],
 [(4, 9)],
 [(5, 23)])
```

نتیجه‌گیری کلی

با توجه به مراحل طی شده و مصورسازی‌ها می‌توان دریافت که scale کردن داده‌ها و پیش‌پرردازش برای این که داده‌ها در خوشه‌های درستی قرار بگیرند بسیا اهمیت دارد و انجام این عمل به شکل قابل توجهی خوشه‌ها نزدیک‌تر به خوشه‌های قبلی داده‌ها شد. اما با توجه به دقت الگوریتم‌های کلاسترینگ که بر اساس فاصله هستند خیی نمی‌توان در فاصله‌های نزدیک به دقت داده‌ی اصلی رسید.

مستندات پروژه و کدها در آدرس زیر در دسترس است.

https://github.com/amirsartipi13/DM-P3-Clustering