دوره آفلاین یادگیری ماشین های بدون نظارت

دوره آموزشی: یادگیری ماشین بدون نظارت
تاریخ گزارش: ۲۴ تیر تا ۳۰ تیر ۱۴۰۵

مدرس: جناب آقای مصطفی توسلی‌پور

نگارنده: سیدعرفان مسعودی

@erfanmasoudiba erfanmasoudiba@gmail.com
linkedin.com/in/erfan-masoudi github.com/ErfanMasoudiBA

فهرست مطالب

۱ تمرین اول - سوال اول: فشرده‌سازی تصویر با K-Means ۲
۲ تمرین اول - سؤال دوم: خوشه‌بندی با DBSCAN ۵
۳ تمرین اول - سؤال سوم: خوشه‌بندی سلسله‌مراتب ۸
۴ تمرین اول - سؤال چهارم: خوشه‌بندی ترکیبی گوسی (GMM) ۱۱
۵ تمرین دوم - سؤال اول: کاهش بعد با PCA و LDA ۱۵
۶ تمرین دوم - سؤال دوم: بیشینه درست‌نمایی (MLE) با توزیع لاپلاس ۱۸
۷ تمرین دوم - سؤال سوم: بررسی نفرین ابعاد ۲۱
۸ تمرین دوم - سؤال چهارم: انتخاب ویژگی افزایشی Selection Forward ۲۳

۱ تمرین اول - سوال اول: فشرده‌سازی تصویر با الگوریتم K-Means

هدف و معرفی مسئله

هدف این تمرین، پیاده‌سازی الگوریتم K-Means Clustering از ابتدا و استفاده از آن برای فشرده‌سازی تصاویر بود. در این روش، هر پیکسل تصویر به عنوان یک نمونه سه‌بعدی شامل مؤلفه‌های رنگی RGB در نظر گرفته شده و با خوشه‌بندی این نمونه‌ها، تعداد رنگ‌های تصویر کاهش می‌یابد. در نهایت هر پیکسل با رنگ مرکز خوشه‌ی متناظر جایگزین می‌شود و تصویری با تعداد رنگ کمتر تولید خواهد شد.

پیاده‌سازی الگوریتم K-Means

الگوریتم K-Means به صورت کامل و بدون استفاده از توابع آماده پیاده‌سازی شد. مراحل اجرای الگوریتم شامل موارد زیر است:

- محاسبه فاصله اقلیدسی هر نمونه تا تمامی مراکز خوشه
- اختصاص هر نمونه به نزدیک‌ترین مرکز خوشه
- محاسبه مجدد مراکز خوشه با استفاده از میانگین اعضای هر خوشه
- تکرار مراحل فوق تا رسیدن به همگرایی یا پایان تعداد تکرارها

برای محاسبه فاصله بین نمونه‌ها و مراکز خوشه از فاصله اقلیدسی استفاده شد:

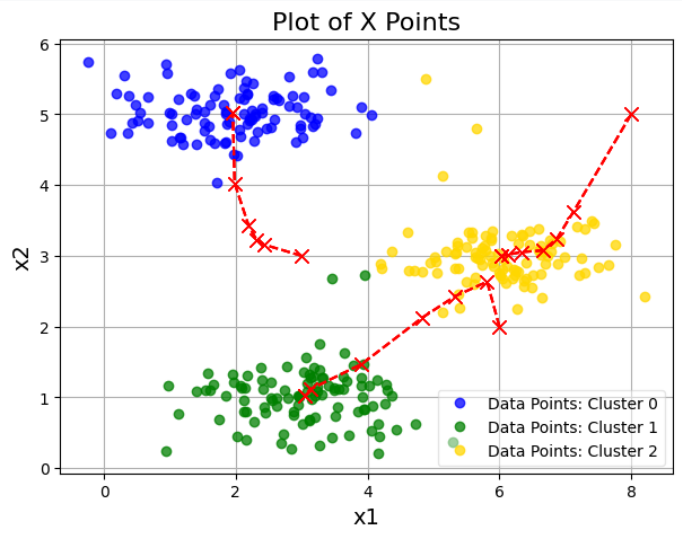
$$d(x, c) = \sum_{i=۱}^{n} (x_i - c_i)^۲$$

که در آن x نمونه داده و c مرکز خوشه است.

اجرای الگوریتم روی مجموعه داده نمونه

در مرحله نخست، عملکرد الگوریتم روی مجموعه داده دوبعدی ارائه‌شده در تمرین بررسی شد. سه مرکز اولیه به صورت دستی انتخاب شدند و الگوریتم طی چند تکرار اجرا گردید. همان‌طور که در شکل مشاهده می‌شود، مراکز خوشه در هر تکرار به

سمت میانگین داده‌های اختصاص یافته حرکت کرده و در نهایت به یک موقعیت پایدار همگرا شدند.



شکل ۱: همگرایی مراکز خوشه در اجرای الگوریتم K-Means

فشرده‌سازی تصویر

برای بررسی کاربرد عملی خوشه‌بندی، یک تصویر رنگی با ابعاد ۱۲۸ × ۱۲۸ انتخاب شد. ابتدا مقادیر رنگی تصویر به بازه $[۰, ۱]$ نرمال شدند و سپس تصویر سه‌بعدی به ماتریسی با ابعاد

$$۱۶۳۸۴ \times ۳$$

تبدیل گردید؛ به طوری که هر سطر نماینده یک پیکسل و هر ستون بیانگر یکی از مؤلفه‌های رنگی RGB بود.

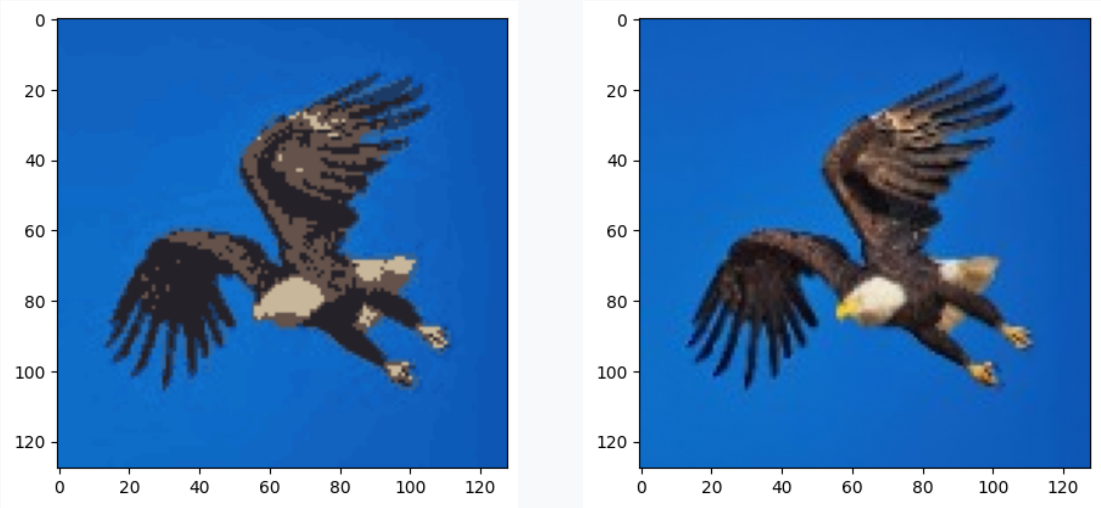
پس از آن، الگوریتم K-Means با تعداد خوشه

$$K = ۱۶$$

اجرا شد. در پایان، هر پیکسل با رنگ مرکز خوشه متناظر جایگزین شد و تصویر فشرده‌شده تولید گردید.

نتایج و تحلیل

شکل زیر تصویر اصلی و تصویر فشرده‌شده را نشان می‌دهد. همان‌طور که مشاهده می‌شود، با وجود کاهش تعداد رنگ‌ها به تنها ۱۶ رنگ، ساختار اصلی تصویر همچنان حفظ شده است و تنها بخشی از جزئیات رنگی از بین رفته‌اند.



شکل ۲: مقایسه تصویر اصلی و تصویر حاصل از فشرده‌سازی با الگوریتم K-Means

کاهش تعداد رنگ‌ها باعث کاهش اطلاعات مورد نیاز برای ذخیره‌سازی تصویر می‌شود. هرچه مقدار K افزایش یابد، کیفیت تصویر نهایی بیشتر خواهد شد؛ اما میزان فشرده‌سازی کاهش پیدا می‌کند. در مقابل، انتخاب مقادیر کوچک‌تر برای K موجب کاهش بیشتر حجم داده‌ها خواهد شد اما افت کیفیت تصویر نیز محسوس‌تر خواهد بود.

جمع‌بندی

در این تمرین، الگوریتم K-Means از پایه پیاده‌سازی و سپس در مسئله فشرده‌سازی تصویر به کار گرفته شد. نتایج نشان دادند که خوشه‌بندی رنگ‌ها می‌تواند بدون از بین رفتن ساختار کلی تصویر، تعداد رنگ‌های مورد استفاده را به طور قابل توجهی کاهش دهد. این آزمایش یکی از کاربردهای مهم الگوریتم‌های یادگیری بدون نظارت در پردازش تصویر را به خوبی نشان می‌دهد.

۲ تمرین اول ـ سؤال دوم: خوشه‌بندی داده‌ها با الگوریتم DBSCAN

هدف و معرفی مسئله

هدف این بخش، پیاده‌سازی الگوریتم خوشه‌بندی مبتنی بر چگالی (DBSCAN) و بررسی عملکرد آن روی یک مجموعه داده دوبعدی بود. برخلاف الگوریتم K-Means که تعداد خوشه‌ها باید از قبل مشخص شود، الگوریتم DBSCAN با استفاده از چگالی نقاط، خوشه‌ها را به صورت خودکار شناسایی کرده و همچنین قادر به تشخیص نقاط نویز نیز می‌باشد.

مجموعه داده مورد استفاده شامل ۴۴۰۶ نمونه و دو ویژگی عددی بود که خروجی الگوریتم t-SNE را نمایش می‌دهد و برای انجام خوشه‌بندی مناسب است.

تعیین پارامترهای الگوریتم

الگوریتم DBSCAN به دو پارامتر اصلی نیاز دارد:

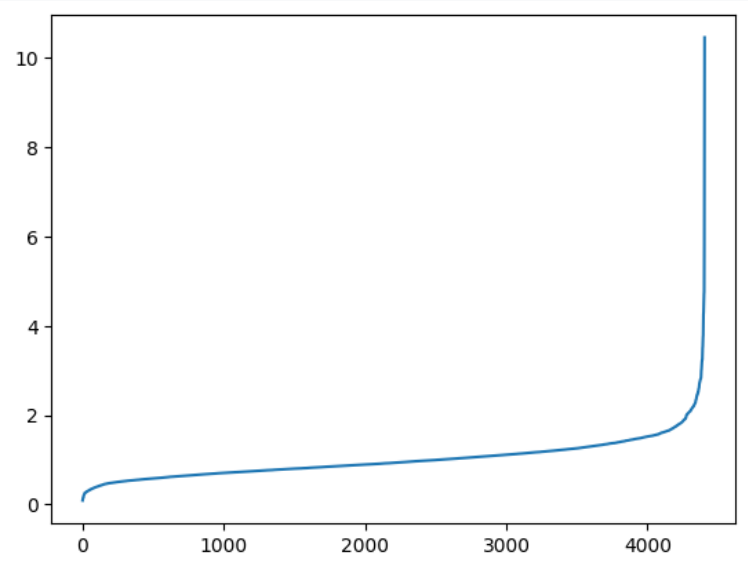
- ε : شعاع همسایگی هر نقطه
- MinPts : حداقل تعداد نقاط لازم برای تشکیل یک ناحیه چگال

مطابق پیشنهاد رایج برای داده‌های دوبعدی، مقدار

$$\text{MinPts} = ۲ \times d = ۴$$

در نظر گرفته شد که در آن d تعداد ابعاد داده است.

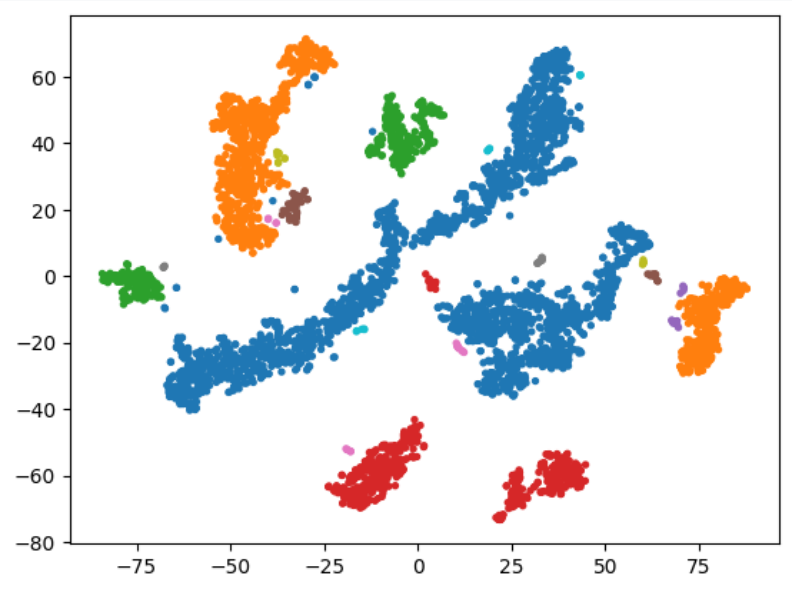
برای تعیین مقدار مناسب ε ابتدا فاصله چهارمین همسایه نزدیک هر نمونه با استفاده از الگوریتم KNN محاسبه شد. سپس با رسم نمودار فاصله‌ها و استفاده از الگوریتم KneeLocator، نقطه زانویی منحنی استخراج و مقدار متناظر آن به عنوان مقدار مناسب ε انتخاب گردید.



شکل ۳: نمودار فاصله چهارمین همسایه و تعیین مقدار مناسب ε

اجرای الگوریتم DBSCAN

پس از تعیین پارامترهای الگوریتم، مدل DBSCAN با استفاده از کتابخانه Scikit-Learn روی مجموعه داده اجرا شد. در پایان، هر نمونه به یکی از خوشه‌ها اختصاص داده شد و نقاطی که در هیچ ناحیه چگالی قرار نگرفتند، به عنوان نویز (Noise) با برچسب ۱− مشخص شدند.



شکل ۴: نتیجه خوشه‌بندی داده‌ها توسط الگوریتم DBSCAN

نتایج و تحلیل

نتایج اجرای الگوریتم نشان داد که DBSCAN موفق به شناسایی

۲۳

خوشه مختلف در مجموعه داده شده است. همچنین تعداد

۳۴

نمونه به عنوان نقاط نویز شناسایی شدند که در هیچ ناحیه با چگالی کافی قرار نداشتند.

مزیت اصلی این الگوریتم نسبت به روش‌هایی مانند K-Means، عدم نیاز به تعیین تعداد خوشه‌ها و همچنین توانایی تشخیص نقاط پرت است. این ویژگی باعث می‌شود DBSCAN برای داده‌هایی با شکل‌های پیچیده و نامنظم عملکرد بسیار مناسبی داشته باشد.

جمع‌بندی

در این تمرین الگوریتم DBSCAN با تعیین خودکار مقدار ε به کمک نمودار فاصله همسایه‌ها و الگوریتم KneeLocator پیاده‌سازی شد. نتایج نشان داد که این روش قادر است بدون اطلاع قبلی از تعداد خوشه‌ها، ساختار طبیعی داده‌ها را استخراج کرده و نقاط نویز را نیز به خوبی شناسایی نماید. به همین دلیل DBSCAN یکی از مناسب‌ترین روش‌های خوشه‌بندی برای داده‌های با توزیع نامنظم و دارای نقاط پرت محسوب می‌شود.

۳ تمرین اول ـ سؤال سوم: خوشه‌بندی داده‌های IRIS با الگوریتم Clustering Agglomerative

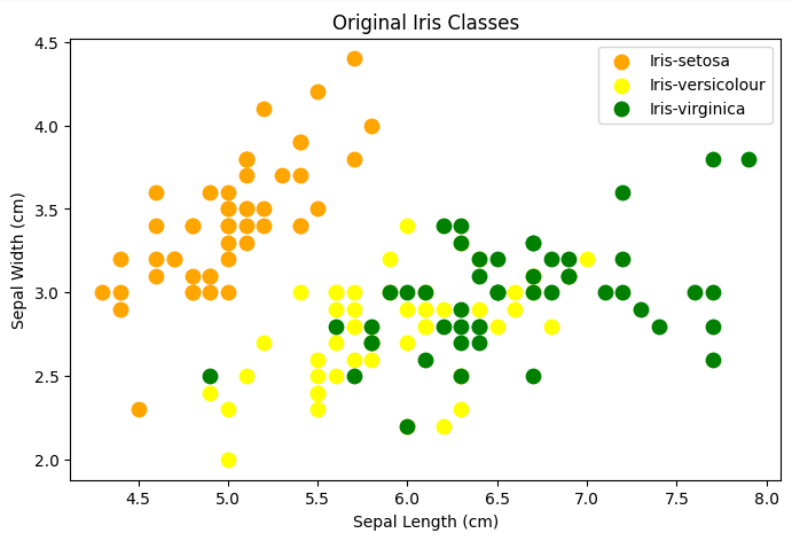
هدف و معرفی مسئله

هدف این بخش، پیاده‌سازی و ارزیابی الگوریتم خوشه‌بندی سلسله‌مراتب تجمعی (Agglomerative Hierarchical Clustering) روی مجموعه داده معروف IRIS بود. این مجموعه داده شامل ۱۵۰ نمونه از سه گونه مختلف گل زنبق (Setosa، Versicolor و Virginica) بر اساس ۴ ویژگی طول و عرض کاسبرگ و گلبرگ است.

در این تمرین، فرآیند خوشه‌بندی به صورت کاملاً بدون نظارت (Unsupervised) و با حذف برچسب‌های واقعی کلاس‌ها انجام شد تا توانایی الگوریتم در کشف ساختار طبیعی داده‌ها سنجیده شود.

تحلیل داده‌ها و بصری‌سازی اولیه

ابتدا داده‌های مربوط به هر یک از سه گونه گل به صورت مجزا فیلتر شده و توزیع ویژگی‌های آن‌ها بررسی گردید. سپس جهت مقایسه اولیه، نمودار پراکندگی ۲ بعدی نمونه‌ها بر اساس طول و عرض کاسبرگ برای تمامی کلاس‌های واقعی ترسیم شد.



شکل ۵: پراکندگی واقعی گونه‌های مختلف گل IRIS در فضای ۲ بعدی

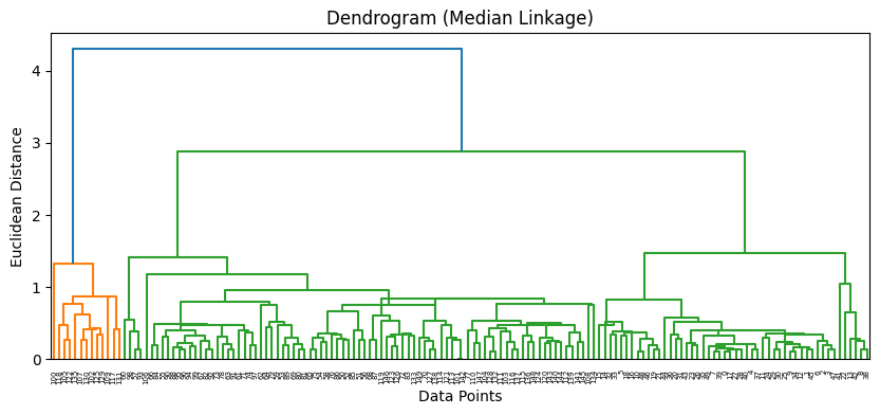
تحلیل دندروگرام و تعیین نوع پیوند (Linkage)

برای تعیین نحوه ادغام خوشه‌ها، دندروگرام مربوط به داده‌ها با چهار روش پیوند مختلف شامل Single، Complete، Average و Median رسم گردید. دندروگرام یک نمودار درختی است که نحوه ادغام تدریجی داده‌ها را در فواصل مختلف نشان می‌دهد.

بر اساس ساختار دندروگرام و دانش اولیه از مسئله، تعداد خوشه‌ها برابر با

$$k = ۳$$

تعیین شد و معیارهای پیوند میان خوشه‌ها ارزیابی گردیدند که روش Average Linkage بهترین پایداری و تفکیک‌پذیری را نشان داد.



شکل ۶: نمودار دندروگرام حاصل از خوشه‌بندی سلسله‌مراتب روی داده‌های IRIS

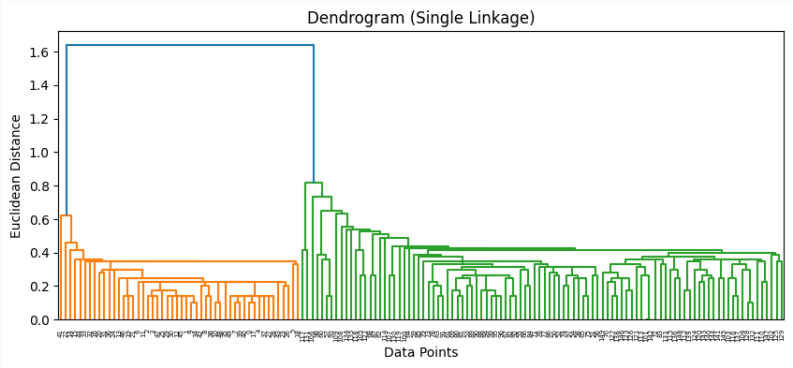
اجرای الگوریتم و ارزیابی با ضریب سیلوئت

مدل AgglomerativeClustering با تعیین $n_clusters = ۳$ و نوع پیوند average روی ویژگی‌های داده اجرا شد. جهت ارزیابی کیفیت خوشه‌بندی، معیار **ضریب سیلوئت (Silhouette Score)** محاسبه گردید.

مقدار ضریب سیلوئت به‌دست‌آمده برابر است با:

$$\boxed{\text{Score Silhouette} \approx ۰/۵۵}$$

این مقدار مثبت و نسبتاً بالا نشان‌دهنده ساختار مناسب خوشه‌ها و جدایش قابل‌قبول آن‌ها از یکدیگر است.



شکل ۷: نتایج حاصل از خوشه‌بندی داده‌های IRIS توسط الگوریتم Agglomerative Clustering

نتایج و تحلیل مقایسه‌ای

تحلیل و مقایسه خوشه‌های تشکیل‌شده توسط الگوریتم با برچسب‌های واقعی داده‌ها نتایج زیر را نشان می‌دهد:

- **گونه Setosa:** به دلیل فاصله زیاد ویژگی‌های آن از دو گونه دیگر، به صورت کاملاً مجزا و با دقت ۱۰۰٪ در یک خوشه مستقل قرار گرفت.
- **گونه‌های Versicolor و Virginica:** به دلیل همپوشانی طیف ویژگی‌های ظاهری در برخی نمونه‌ها، مرز میان این دو گونه دارای اندکی تداخل است که در خروجی الگوریتم خوشه‌بندی نیز این همپوشانی به خوبی مشهود می‌باشد.

جمع‌بندی

در این بخش الگوریتم خوشه‌بندی سلسله‌مراتب روی مجموعه داده IRIS پیاده‌سازی شد. با تحلیل نمودار دندروگرام و اعمال روش پیوند Average، الگوریتم موفق شد بدون دسترسی به برچسب‌های واقعی، سه خوشه اصلی داده‌ها را با ضریب سیلوئت ۰/۵۵ بازسازی کند. نتایج نشان داد که خوشه‌بندی سلسله‌مراتب گزینه‌ای بسیار مناسب برای درک ساختار درختی و ارتباطات میان داده‌ها است.

۴ تمرین اول ـ سؤال چهارم: خوشه‌بندی نرم داده‌ها با مدل ترکیبی گوسی (GMM)

هدف و معرفی مسئله

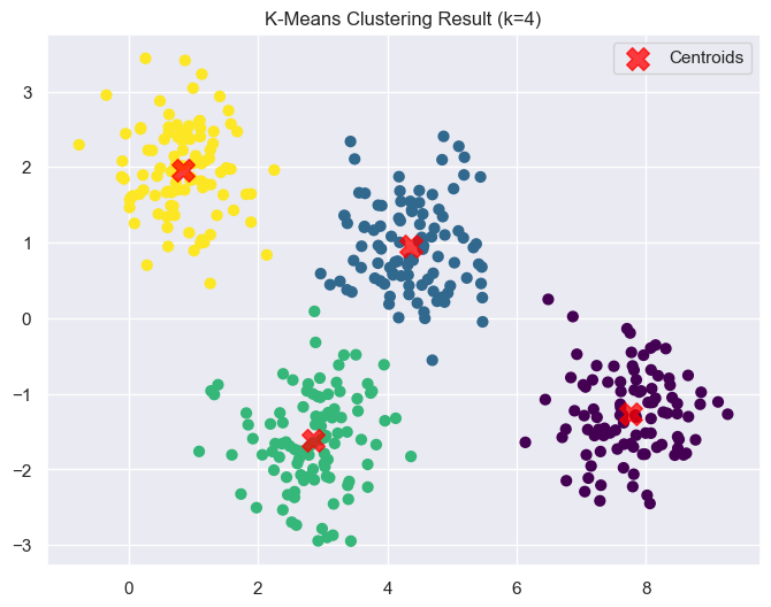
هدف این بخش، پیاده‌سازی و بررسی الگوریتم **مدل ترکیبی گوسی (Gaussian Mixture Model - GMM)** و مقایسه مفاهیم خوشه‌بندی سخت (Hard Clustering) مانند K-Means با خوشه‌بندی نرم (Soft Clustering) می‌باشد.

در الگوریتم‌های خوشه‌بندی نرم، هر نمونه برعکس الگوریتم‌های معمولی به صورت قطعی به یک خوشه متصل نمی‌شود، بلکه یک توزیع احتمالی از میزان تعلق نقطه به هر یک از توابع گوسی (خوشه‌ها) محاسبه می‌گردد. مجموعه داده مورد استفاده در این تمرین شامل ۴۰۰ نمونه دوبعدی تولیدشده توسط تابع make_blobs است.

ارزیابی اولیه و خوشه‌بندی با K-Means

با بررسی نمودار پراکندگی اولیه داده‌ها، وجود ۴ مرکز توزیع واضح در فضای داده مشاهده گردید. در مرحله نخست، الگوریتم K-Means با انتخاب $k = ۴$ اجرا گردید.

الگوریتم K-Means با فرض کروی بودن مرز خوشه‌ها و تخصیص قطعی مرزها عمل می‌کند. نتایج این خوشه‌بندی در شکل زیر آورده شده است.

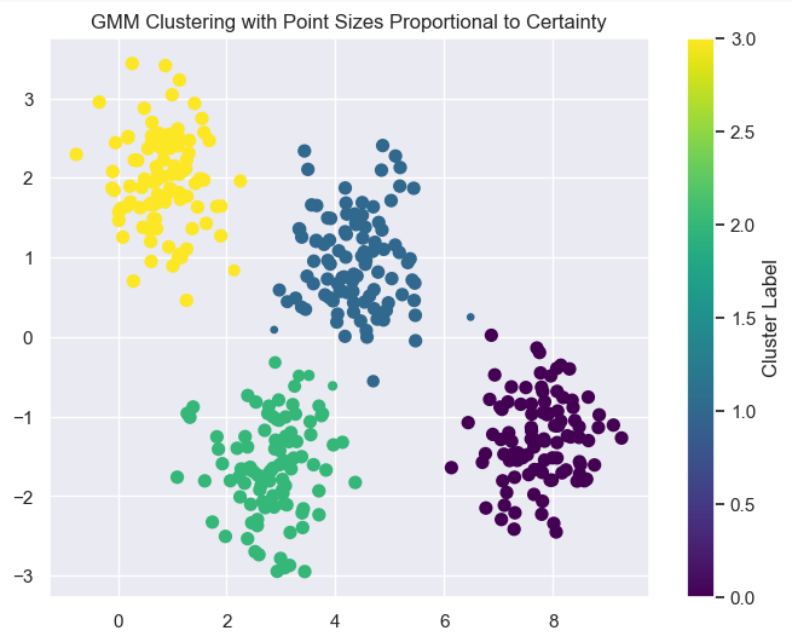


شکل ۸: نتیجه خوشه‌بندی داده‌ها توسط الگوریتم K-Means ($k = ۴$)

آموزش مدل GMM و محاسبه ماتریس احتمالات

در ادامه، مدل GaussianMixture از ماژول sklearn.mixture با تعیین $n_components =$ ۴ روی داده‌ها برازش داده شد. سپس با استفاده از متد predict_proba، ماتریس احتمالات به ابعاد (۴۰۰, ۴) محاسبه گردید.

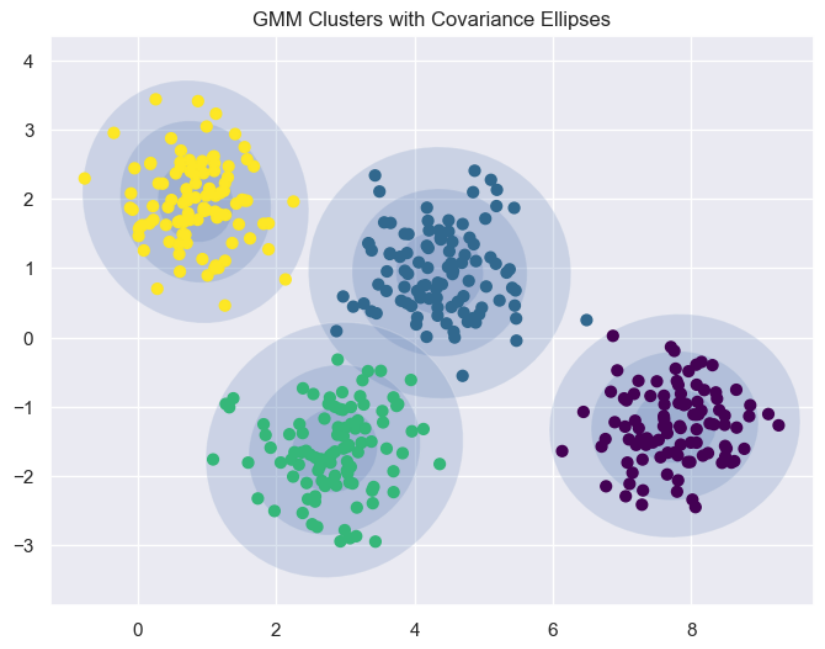
هر سطر از این ماتریس، احتمال حضور یک نمونه داده را در ۴ تابع توزیع گوسی نشان می‌دهد. جهت ملموس‌تر شدن مفهوم عدم قطعیت (Uncertainty)، اندازه نقاط در نمودار متناسب با حداکثر احتمال تعلق آن‌ها تنظیک گردید:



شکل ۹: نمایش میزان قطعیت خوشه‌بندی نمونه‌ها (اندازه بزرگتر نشان‌دهنده احتمال تعلق بالا به خوشه است)

رسم بیضی‌های کواریانس و تحلیل خروجی

با تکمیل توابع draw_ellipse و plot_gmm، بیضی‌های هم‌احتمال توزیع‌های گوسی برای هر خوشه ترسیم شدند. این بیضی‌ها ماتریس کواریانس و نحوه پراکندگی داده‌های هر خوشه را نمایش می‌دهند.



شکل ۱۰: نتیجه خوشه‌بندی GMM به همراه بیضی‌های توزیع کواریانس خوشه‌ها

نتایج و استنباط تحلیلی

تحلیل و استنباط حاصل از خروجی‌های مدل GMM:

- **خوشه‌بندی نرم در برابر سخت:** برعکس K-Means که نقاط مرزی را مجبورا به یک خوشه نسبت می‌دهد، GMM عدم قطعیت نقاط مرزی را بر حسب احتمال بازگو می‌کند.
- **شکل خوشه‌ها:** استفاده از ماتریس کواریانس عمومی به GMM این امکان را می‌دهد تا خوشه‌های بیضوی و کشیده را نیز بهتر از K-Means پوشش دهد.
- **میزان انحراف:** بیضی‌های مرکز خوشه‌ها فشرده‌تر و بیضی‌های بیرونی نشان‌دهنده انحراف معیارهای بالاتر ($\sigma, ۲\sigma, ۳\sigma$) در توزیع گوسی متناظر هستند.

جمع‌بندی

در این بخش، الگوریتم GMM با ۴ مؤلفه گوسی پیاده‌سازی و ارزیابی شد. نتایج نشان داد که این الگوریتم به‌واسطه ارائه خروجی احتمالی (predict_proba) و انعطاف‌پذیری

هندسی به کمک ماتریس کواریانس، ابزاری بسیار قدرتمندتر نسبت به K-Means برای تحلیل داده‌هایی با مرزهای مبهم یا توزیع‌های بیضوی است.

۵ تمرین دوم ـ سؤال اول: پیاده‌سازی دستی و کتابخانه‌ای روش‌های کاهش بعد PCA و LDA

هدف و معرفی مسئله

هدف این بخش، پیاده‌سازی الگوریتم‌های **تحلیل مؤلفه‌های اصلی (PCA)** به عنوان یک روش کاهش بعد بدون نظارت (Unsupervised) و **تحلیل ممیز خطی (LDA)** به عنوان یک روش با نظارت (Supervised) بر روی مجموعه داده دیابت (diabetes.csv) می‌باشد.

در این سؤال، نحوه عملکرد الگوریتم‌ها با پیاده‌سازی ریاضی دستی توسط کتابخانه NumPy انجام شده و خروجی‌های بصری آن به صورت دقیق با خروجی‌های کتابخانه Scikit-Learn مقایسه و تحلیل گردید.

پیش‌پردازش داده‌ها و استانداردسازی

مجموعه داده دیابت شامل ۷۶۸ نمونه و ۸ ویژگی پزشکی (مانند قند خون، انسولین، فشار خون و سن) به همراه متغیر هدف Outcome است.

از آنجا که روش‌های کاهش بعد مبتنی بر ماتریس کواریانس و فاصله‌ها هستند، ویژگی‌ها با استفاده از StandardScaler به میانگین صفر و واریانس یک استانداردسازی شدند:

$$z = \frac{x - \mu}{\sigma}$$

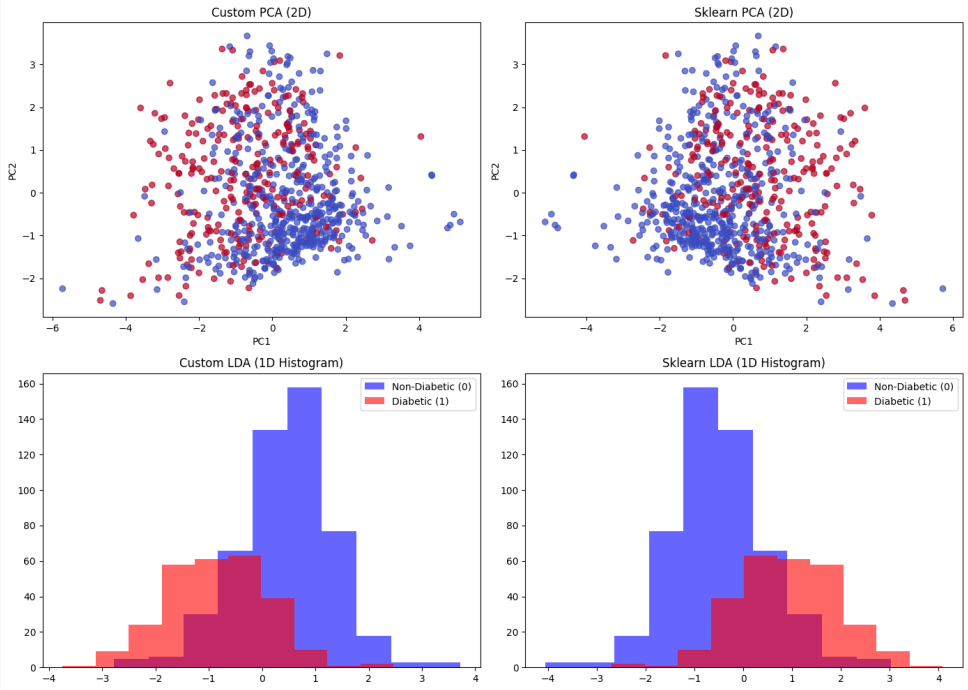
پیاده‌سازی دستی الگوریتم PCA و LDA

مراحل پیاده‌سازی ریاضی دو الگوریتم با NumPy به شرح زیر انجام شد:

- **الگوریتم PCA:** با مرکزیک کردن داده‌ها، محاسبه ماتریس کواریانس و استخراج بردارهای ویژه (Eigenvectors)، ۲ مؤلفه اصلی برتر انتخاب شدند.
- **الگوریتم LDA:** با محاسبه ماتریس‌های پراکندگی درون‌کلاسی (S_W) و بین‌کلاسی (S_B) و حل مسئله مقدار ویژه تعمیم‌یافته برای $S_W^{-1}S_B$ پیاده‌سازی شد. با توجه به $C = ۲$ کلاس، خروجی LDA به $C - ۱ = ۱$ بعد نگاشت گردید.

تحلیل بصری خروجی‌ها و مقایسه پیاده‌سازی‌ها

با بررسی دقیق چهار نمودار تولیدشده در خروجی کد، نتایج تحلیلی زیر استخراج گردید:



شکل ۱۱: مقایسه دقیق خروجی‌های پیاده‌سازی دستی (Custom) و کتابخانه‌ای (Sklearn) برای الگوریتم‌های PCA و LDA

- **تحلیل معکوس شدن علامت در محورها (Sign Inversion):** همان‌طور که در نمودار PCA و هیستوگرام LDA مشهود است، شکل کلی پراکندگی داده‌ها در پیاده‌سازی دستی و اسکلی‌رن کاملاً یکسان است، با این تفاوت که جهت محور افقی (PC1) معکوس (۱۸۰° چرخش) شده است. این موضوع از نظر ریاضی کاملاً طبیعی است؛ زیرا بردارهای ویژه (v) تنها تا ضریب یک ضرب‌کننده منفی (±v) یکتا هستند و هر دو جهت از نظر حفظ واریانس معتبرند.

- **تحلیل عدم تفکیک‌پذیری در PCA:** در نمودارهای دو بعدی PCA (تصاویر بالا)، داده‌های افراد دیابتی (قرمز) و غیردیابتی (آبی) به شدت در یکدیگر تداخل دارند. علت این امر عدم استفاده PCA از برچسب داده‌هاست؛ این الگوریتم صرفاً جهتی را انتخاب می‌کند که بیشترین پراکندگی (واریانس) را داشته باشد، نه جهتی که بهترین تفکیک کلاس را ایجاد کند.

- **تحلیل جدایش در LDA:** در هیستوگرام‌های یک‌بعدی LDA (تصاویر پایین)، توزیع دو کلاس به میزان قابل‌توجهی از هم تفکیک شده‌اند. مرکز توزیع بیماران غیردیابتی و دیابتی در دو سمت مخالف محور قرار گرفته‌اند که نشان‌دهنده موفقیت LDA در بیشینه کردن فاصله میانگین کلاس‌ها نسبت به واریانس درون‌کلاسی است.

جمع‌بندی

در این بخش، صحت پیاده‌سازی ریاضی الگوریتم‌های PCA و LDA اثبات گردید. تحلیل بصری خروجی‌ها نشان داد که جهت‌گیری بردارهای ویژه می‌تواند تفاوت علامت جزئی داشته باشد اما رفتار کلی داده‌ها یکسان است. همچنین مشخص گردید برای مسائل دسته‌بندی و جداسازی کلاس‌ها، روش با نظارت LDA کارایی به مراتب بالاتری نسبت به روش بدون نظارت PCA ارائه می‌دهد.

۶ تمرین دوم ـ سؤال دوم: تخمین بیشینه درست‌نمایی (MLE) و رگرسیون لاپلاس

هدف و معرفی مسئله

هدف این بخش، پیاده‌سازی روش **تخمین بیشینه درست‌نمایی (Maximum Likelihood Estimation - MLE)** بر روی داده‌های زمان فوران آتشفشان (Q2.csv) با فرض تابع توزیع لاپلاس (Laplace Distribution) و مقایسه عملکرد آن با رگرسیون خطی معمولی (OLS / L_2) می‌باشد.

در بسیاری از مسائل دنیای واقعی، داده‌ها ممکن است حاوی نقاط پرت (Outliers) باشند یا توزیع خطا غیرگوسی باشد. تابع توزیع لاپلاس به دلیل استفاده از قدر مطلق خطاها در لوگ-درست‌نمایی، مقاومت به مراتب بالاتری در برابر داده‌های پرت دارد.

تعریف ریاضی لگاریتم منفی درست‌نمایی (NLL)

فرمول منفی لگاریتم درست‌نمایی (Negative Log-Likelihood) برای n نمونه مستقل با توزیع لاپلاس به صورت زیر تعریف می‌شود:

$$\ell(y_۱, y_۲, \ldots, y_n; \mu, \lambda) = -\sum_{i=۱}^{n} \left(-\log(۲\lambda) - \frac{|y_i - \mu|}{\lambda} \right)$$

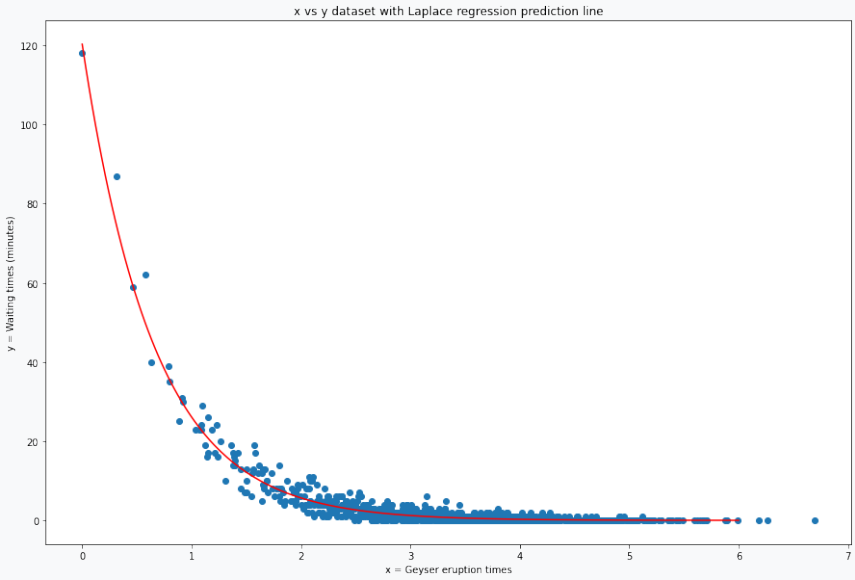
در مدل رگرسیون غیرخطی این تمرین، میانگین توزیع به صورت نمایی وابسته به ویژگی‌ها در نظر گرفته شد:

$$\mu = \exp(\mathbf{X}\mathbf{b})$$

جهت بهینه‌سازی و یافتن ضرایب b، از الگوریتم بهینه‌سازی Powell درون تابع scipy.optimize.minimize استفاده گردید.

برازش مدل لاپلاس بر روی داده‌های واقعی

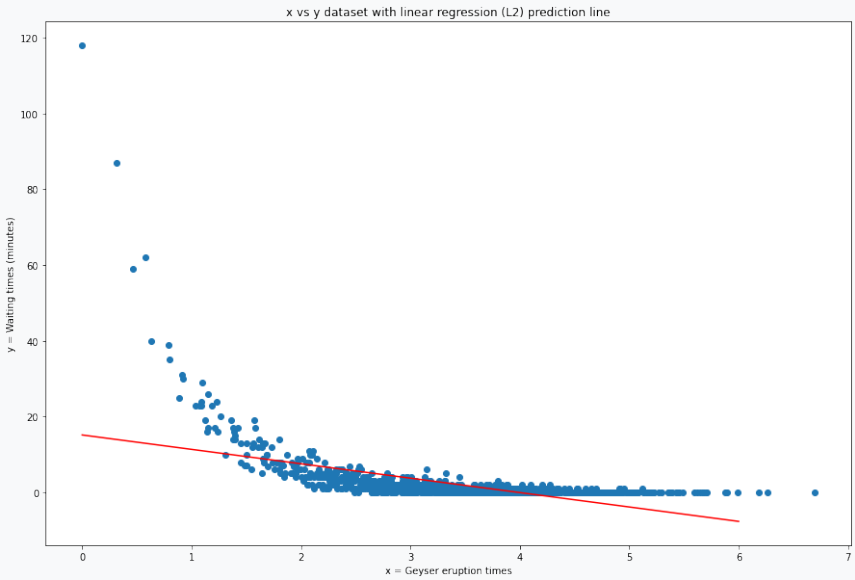
پس از تعریف توابع ‘laplaceRegNegLogLikelihood‘ و ‘modelFit‘، ضرایب بهینه b برای داده‌های مجموعه Q2.csv استخراج شده و منحنی پیش‌بینی برای داده‌های جدید $x \in [۰, ۶]$ ترسیم گردید.



شکل ۱۲: برازش رگرسیون لاپلاس ($\mu = \exp(\mathbf{Xb})$) بر روی داده‌های زمان فوران و انتظار

برازش رگرسیون خطی معمولی (OLS / $L_۲$) و مقایسه

در ادامه، مدل رگرسیون خطی معمولی با فرض توزیع خطا به صورت گوسی (کمترین مربعات) بر روی همان داده‌ها برازش داده شد.



شکل ۱۳: برازش رگرسیون خطی معمولی ($\mathrm{OLS}\ /\ L_2$) بر روی مجموعه داده

تحلیل و پاسخ به خواسته سوال (چرا رگرسیون خطی دچار مشکل است؟)

با مقایسه دو نمودار فوق، نتایج تحلیلی زیر استخراج گردید:

- **ضعف رگرسیون خطی $L_۲$ در برابر داده‌های پرت:** در الگوریتم $L_۲$، خطای هر نقطه به توان ۲ می‌رسد ($|y_i - \hat{y}_i|^۲$). نقاط پرتی که مقدار y آن‌ها بسیار بزرگ است، جریمه فوق‌العاده شدیدی به تابع هزینه تحمیل می‌کنند. این امر باعث می‌شود که خط رگرسیون به شدت به سمت نقاط پرت منحرف شده و توانایی خود را در پیش‌بینی دقیق تنه اصلی داده‌ها از دست بدهد (Underfitting روی داده‌های اصلی).

- **مزیت رگرسیون لاپلاس ($L_۱$):** توزیع لاپلاس از قدر مطلق خطاها ($|y_i - \hat{y}_i|$) استفاده می‌کند. این ویژگی باعث می‌شود که تاثیر نقاط پرت به صورت خطی رشد کند نه نمایی؛ در نتیجه مدل لاپلاس نسبت به نقاط پرت استوار (Robust) بوده و رفتار واقعی روندهای نمایی در داده‌ها را بسیار دقیق‌تر مدل‌سازی می‌کند.

جمع‌بندی

در این تمرین، با پیاده‌سازی دستی تابع درست‌نمایی لاپلاس و بهینه‌سازی آن، نشان داده شد که انتخاب تابع توزیع مناسب متناسب با ماهیت داده‌ها اهمیت بالایی دارد. رگرسیون لاپلاس به دلیل استواری در برابر داده‌های پرت (Outlier Robustness) و فرم نمایی $\mu = \exp(\mathbf{Xb})$، عملکرد بسیار برتری نسبت به رگرسیون خطی معمولی ارائه داد.

۷ تمرین دوم ـ سؤال سوم: بررسی پدیده نفرین ابعاد (Curse of Dimensionality)

هدف و معرفی مسئله

هدف این بخش، بررسی تجربی یکی از چالش‌های بنیادین در یادگیری ماشین و پردازش داده تحت عنوان **نفرین ابعاد (Curse of Dimensionality)** است.

با افزایش ابعاد داده‌ها، فضای حالت به قدری بزرگ و خلوت (Sparse) می‌شود که محاسبات مبتنی بر فاصله (نظیر k-NN یا خوشه‌بندی) دچار اختلال شدید شده و مفهوم «همسایگی نزدیک» معنای فیزیکی و آماری خود را از دست می‌دهد.

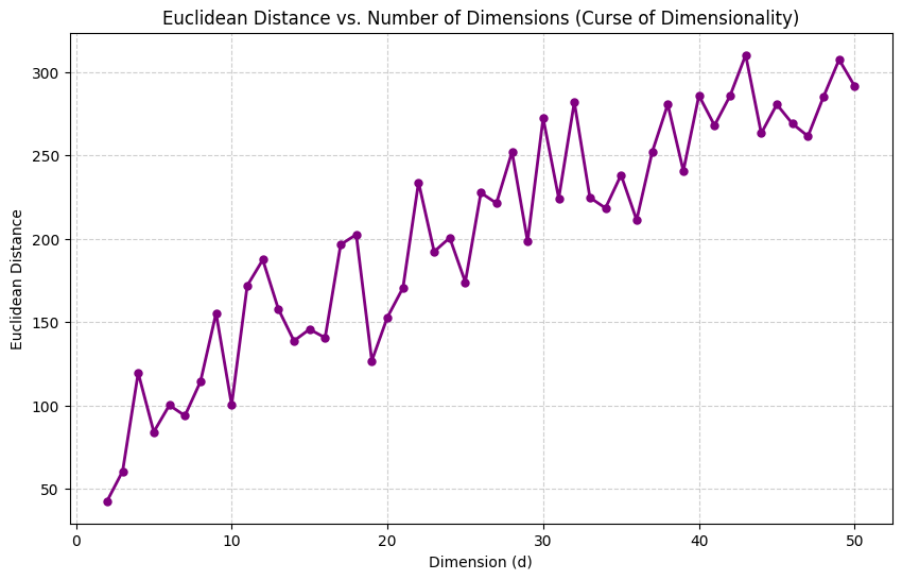
نحوه پیاده‌سازی و فرمول محاسباتی

برای بررسی این پدیده، در هر بعد d از ۲ تا ۵۰ ($d \in [۲, ۵۰]$) دو بردار تصادفی x و y با درایه‌های تصادفی صحیح در بازه $[۰, ۱۰۰]$ با استفاده از تابع `np.random.randint` ایجاد گردید. سپس فاصله اقلیدسی بین آن‌ها به صورت زیر محاسبه شد:

$$D(x, y) = \|x - y\|_۲ = \sqrt{\sum_{i=۱}^{d} (x_i - y_i)^۲}$$

نمودار و تحلیل خروجی

نمودار زیر تغییرات فاصله اقلیدسی دو بردار تصادفی را بر حسب افزایش ابعاد از $d = ۲$ تا $d = ۵۰$ نشان می‌دهد:



شکل ۱۴: تغییرات فاصله اقلیدسی بر حسب تعداد ابعاد (d)

- **روند صعودی فاصله اقلیدسی:** همان‌طور که در نمودار مشخص است، با افزایش ابعاد d، میانگین فاصله اقلیدسی بین دو نقطه تصادفی به شدت رشد کرده و از حدود ۴۰ در فضای ۲بعدی به بیش از ۲۹۰ در فضای ۵۰بعدی می‌رسد.
- **علت ریاضی:** هر بعد جدید، یک جمله مثبت $(x_i - y_i)^2$ به زیر رادیکال اضافه می‌کند که باعث افزایش مجموع مربع خطاهایت تحت جذر می‌شود.

استنباط و نتیجه‌گیری

تحلیل این خروجی، مفاهیم زیر را در یادگیری ماشین روشن می‌سازد:

۱. **دور شدن نقاط از یکدیگر در ابعاد بالا:** در فضاهای با ابعاد بسیار بالا، تمامی نقاط به طرز عجیبی از هم دور شده و فاصله‌های نسبی بین تمام جفت‌نقاط تقریباً یکسان می‌گردد.

۲. **ضرورت الگوریتم‌های کاهش بعد:** همین پدیده نشان می‌دهد چرا در داده‌های واقعی با ابعاد بالا، استفاده از روش‌هایی مانند PCA و LDA قبل از اعمال الگوریتم‌های یادگیری حیاتی و ضروری است.

۸ تمرین دوم ـ سؤال چهارم: پیاده‌سازی انتخاب ویژگی افزایشی (Forward Selection) بر روی سرطان سینه

هدف و معرفی مسئله

هدف این بخش، بررسی مسئله **انتخاب ویژگی (Feature Selection)** به روش **افزایشی (Sequential Forward Selection - SFS)** بر روی مجموعه داده سرطان سینه (cancer.csv) است.

هدف اصلی انتخاب ویژگی، کاهش ابعاد مسئله، حذف ویژگی‌های تکراری و نویز، جلوگیری از بیش‌برازش (Overfitting) و افزایش تفسیرپذیری مدل با حفظ حداکثر دقت دسته‌بندی می‌باشد.

بررسی اولیه داده‌ها و پاکسازی داده‌های پرت

مجموعه داده اولیه شامل ۵۶۹ نمونه، یک ستون شناسه (id)، ستون هدف diagnosis (با برچسب‌های M بدخیم و B خوش‌خیم) و ۳۰ ویژگی پزشکی عددی است.

به منظور پالایش داده‌ها، با استفاده از روش محدوده بین‌چارکی (IQR)، نمونه‌های دارای مقادیر پرت شناسایی و حذف گردیدند. در نهایت ۳۹۸ نمونه پاکسازی‌شده برای مرحله آموزش و ارزیابی باقی ماندند.

پیاده‌سازی طبقه‌بند و انتخاب ویژگی افزایشی (SFS)

از مدل **رگرسیون لوجستیک (Logistic Regression)** به عنوان طبقه‌بند پایه استفاده شد. روند اجرا به شرح زیر است:

۱. **مدل کامل (با ۳۰ ویژگی):** ابتدا مدل با تمامی ۳۰ ویژگی آموزش داده شد که به دقت آموزش ۹۵٫۶٪ و دقت آزمون ۹۵٫۰٪ دست یافت.

۲. **اعمال SFS:** با استفاده از ماژول `SequentialFeatureSelector` از کتابخانه Scikit-Learn، تعداد ۵ ویژگی برتر به صورت مرحله به مرحله انتخاب شدند.

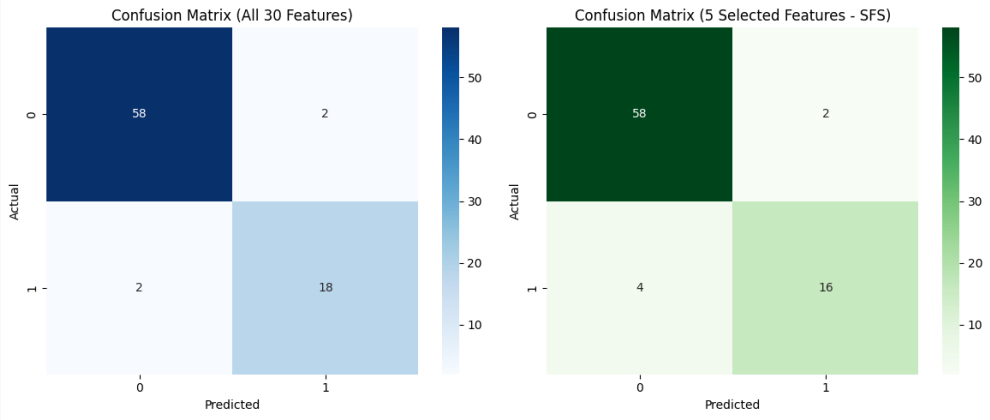
ویژگی‌های منتخب عبارتند از:

- `texture_mean` (میانگین بافت)

- `smoothness_mean` (میانگین همواری)
- `perimeter_se` (خطای معیار محیط)
- `texture_worst` (بدترین بافت)
- `area_worst` (بدترین مساحت)

ارزیابی و مقایسه با ماتریس آشفتگی

ماتریس‌های آشفتگی (Confusion Matrix) برای دو حالت کامل (۳۰ ویژگی) و منتخب (۵ ویژگی SFS) در تصویر زیر رسم شده‌اند:



شکل ۱۵: مقایسه ماتریس آشفتگی مدل با کل ۳۰ ویژگی (سمت چپ) و ۵ ویژگی منتخب SFS (سمت راست)

- **تحلیل ماتریس کامل (۳۰ ویژگی):** از ۸۰ داده آزمون، ۵۸ نمونه منفی واقعی (True Negative) و ۱۸ نمونه مثبت واقعی (True Positive) به درستی پیش‌بینی شده‌اند و تنها ۴ خطای مجموع وجود دارد (دقت ۹۵٪).
- **تحلیل ماتریس SFS (۵ ویژگی):** با تنها ۵ ویژگی، تعداد ۵۸ منفی واقعی و ۱۶ مثبت واقعی به درستی تشخیص داده شده‌اند (دقت ۹۲/۵٪).
- **توجیه علت انتخاب این ویژگی‌ها:** الگوریتم SFS ویژگی‌هایی را انتخاب کرده است که اطلاعات غیرتکراری ارائه می‌دهند؛ ترکیب شاخص‌های بافت

(texture) و اندازه خطیر غده (area_worst و perimeter_se) بیشترین قدرت تفکیک بافت‌های بدخیم و خوش‌خیم را فراهم ساخته است.

جمع‌بندی

نتایج نشان داد که با کاهش ۸۳٪ ابعاد ویژگی‌ها (از ۳۰ به ۵ ویژگی)، دقت مدل تنها ۲٫۵٪ افت داشته است که نشان‌دهنده کارایی بالای الگوریتم Forward Selection در بهینه‌سازی مدل‌های پزشکی می‌باشد.