گزارش عملکردی در مورد تعیین عملکرد الگوریتم‌های یادگیری ماشین نظارت شده KNN و RandomForest در پیش بینی دسته بندی داده‌ها

امروزه برای پردازش و تحلیل داده‌ها مخصوصا در حجم بالا، الگوریتم‌های هوش مصنوعی و یادگیری ماشین، مورد استفاده قرار می‌گیرند. بخش مهم پردازش داده و تبدیل آن به اطلاعات قابل فهم تر، توانایی دسته بندی داده‌های ورودی جدید می‌باشد. از این رو الگوریتم‌های یادگیری برای دسته بندی داده‌های ورودی توسعه داده شده‌اند.

الگوریتم‌های یادگیری به چهار دسته کلی یادگیری‌های نظارت شده[۱]، بدون نظارت[۲]، نیمه نظارتی[۳] و تقویت شده[۴] دسته بندی می‌گردند. الگوریتم‌های مورد مطالعه جزو دسته یادگیری نظارت شده محسوب می‌گردند. [۴]

این متد، مبتنی‌بر تابعی است که پردازش و تبدیل ورودی به خروجی را براساس جفت نمونه‌های ورودی خروجی ترسیم و اجرا می‌کند. برای انجام این کار، این نوع الگوریتم‌ها از داده‌ها و مثال‌های آموزشی برچسب گذاری شده برای استنتاج خروجی‌ها استفاده می‌کند. این امر به این معنی است که دسترسی به اهداف، از طریق تعیین آن‌ها و با کمک برخی از داده‌های ورودی صورت می‌گیرد. از این مدل الگوریتم‌های نزدیک‌ترین همسایه[۵] و جنگل تصادفی[۶] مورد بررسی قرار می‌گیرد که از این به بعد به اختصار به‌صورت KNN و RFC معرفی می‌گردند. [۵]

برای مطالعه و بررسی این الگوریتم‌ها، دیتاست مربوط به گونه‌های گل زنبق[۷] معروف به دیتابیس فیشر مورد مطالعه و بررسی قرار می‌گیرد. این جدول شامل سه نوع گل می‌باشد که از طریق ۴ ویژگی sepallength, sepalwidth, petallength, petalwidth که مرتبط به ابعاد اجزای مختلف گل‌ها به cm هستند. ۳ نوع گل با ویژگی class در دیتاست معرفی شدند که نام علمی گونه آن ها، Iris-sesots, Iris-versicolor, Iris-virginica معرفی می‌گردند.

الگوریتم KNN [۱]

الگوریتم‌های دسته بندی در تلاش هستند تا داده‌های ورودی را با توجه به ویژگی‌های آن، دسته بندی کنند. شیوه این دسته بندی به‌صورت توزیع شرطی از ویژگی‌های مختلف به‌صورت دو به دو و یا مقادیر هدف و ویژگی‌ها می‌باشد که به‌صورت توزیع Y نسبت به X مطرح می‌شوند.

الگوریتم KNN، الگوریتمی بسیار ساده، اما کاربردی برای دسته بندی داده‌ها می‌باشد. KNN براساس شناسایی نزدیک‌ترین همسایه‌های هر داده آزمایشی که با X_0 معرفی می‌گردد ، می‌باشد. در این روش، ابتدا تعداد انواع K نقاطی که در همسایگی X_0

[1] Supervised learning
[2] unsupervised learning (clustering)
[3] Semi-supervised learning
[4] Reinforcement learning
[5] K-nearest neighbors
[6] Random Forest
[7] Iris dataset

قرار دارند شناسایی شده و با عنوان N_0 نشان داده می‌شوند. سپس تابع احتمال شرطی j به عنوان کسری از نقاط N_0 که پاسخ آنها برابر j باشد تخمین زده می‌شود :

$$\Pr(Y = j|X = x_0) = \frac{1}{K}\sum_{i\in N_0} I(y_i = j)$$ رابطه (۱)

در نهایت، KNN مشاهدات آزمایشی X_0 را به کلاسی با بیشترین احتمال از فرمول بالا طبقه بندی می‌کند. با توجه به این موضوع، انتخاب تعداد همسایه‌ها (K) بسیار اهمیت دارد. با توجه به رابطه معکوس K با تابع احتمال، با افزایش این مقدار، خطوط جدا کننده، دسته بندی‌ها به حالت خطی میل می‌کنند و مقادیر خطای پیش بینی افزایش می‌یابد که در دنیای واقعی، دسته بندی خطی دقت بالایی را ارائه نمی‌کند. پس این الگوریتم با انتخاب تعداد همسایه‌های کمتر دسته بندی بهتر و با انعطاف بالاتری را ارائه می‌دهد.

الگوریتم RFC [۳]

RFC، الگوریتم موفق و با سرعت بالا در زمینه دسته بندی داده‌ها محسوب می‌گردد. امکانات این الگوریتم قابلیت انعطاف آن را افزایش داده و می‌توان نویز داده‌ها را در استفاده از RFC نادیده گرفت. جنگل‌های تصادفی، شامل مجموعه ای از احتمال نمونه‌های تصادفی طبقه بندی‌های مشترک $h_1(X),\ h_2(X), \dots, h_k(X)$ می‌باشد. هر طبقه نماینده یکی از کلاس‌ها بوده و نمونه ای که برای طبقه بندی پردازش می‌شود، با کلاس طبقه قوی تر برچسب گذاری می‌شود. در این الگوریتم، n نمونه تصادفی با جایگزینی از n نمونه از مجموعه آموزشی برای ایجاد درخت‌های طبقه بندی ترسیم می‌گردد. با تکرار این فرآیند که با عنوان بوت استرپ[8] معرفی می‌گردد، به‌طور متوسط ۳۶.۸ درصد از نمونه‌های آموزشی برای ساخت هر درخت، مورد استفاده قرار نمی‌گیرد که این نمونه‌های باقی مانده برای محاسبه تخمین داخلی قدرت و میزان همبستگی جنگل مورد استفاده قرار می‌گیرد. مجموعه‌ای از این نمونه‌ها برای طبقه h_k با O_k نمایش داده می‌شوند. تابع $Q(X, y_j)$ برای نمایش نماینده دسته y_j در ورودی X و محاسبه احتمال $P(h(X) = y_j)$ مورد استفاده قرار می‌گیرد :

$$Q(X, y_j) = \frac{\sum_{k=1}^{K} I(h_k(x) = y_j; (X,y) \in O_k}{\sum_{k=1}^{K} I(h_k(X); (X,y) \in O_k)}$$ رابطه (۲)

متغیر I، تابع نمایشگر می‌باشد. تابع لبه‌ای، مقدار بیشینه احتمال نماینده هر کلاس y را نسبت به میانگین دیگر طبقات محاسبه می‌کند.

$$mr(X, y) = P(h(X) = y) - max_{\substack{j=1 \\ j\neq y}}^{c} P(h(X) = y_j)$$ رابطه (۳)

این تابع با استفاده از Q(X,y) و $Q(X, y_i)$ تخمین زده می‌شود. تابع قدرت به عنوان لبه مورد انتظار به‌صورت زیر تعریف می‌گردد که به‌صورت میانگین داده‌های آموزشی مورد استفاده قرار می‌گیرد :

$$s = \frac{1}{n}\sum_{i=1}^{n}(Q(X_i, y) - max_{\substack{j=1 \\ j\neq y}}^{c} Q(X_i, y_j))$$ رابطه (۴)

[8] bootstraping

میانگین همبستگی به عنوان واریانس لبه‌ای برای مربع انحرافات جنگل تولید شده محاسبه می‌گردد :

$$\bar{\rho}\frac{var(mr)}{sd(h(\,))^2} = \frac{\frac{1}{n}\sum_{i=1}^{n}(Q(X_i,y) - \max_{\substack{j=1\\ j\neq y}}^{c} Q(X_i,y_j))^2 - s^2}{(\frac{1}{k}\sum_{t=1}^{k}\sqrt{p_k + \widehat{p_k} + (p_k + \widehat{p_k})^2})^2}$$

رابطه (۵)

که در آن مقادیر تخمین نمونه‌های باقیمانده $P(h_k(X) = y$ و $P(h_k(X) = \widehat{y_j}$ به‌صورت زیر حساب می‌گردند :

$$p_k = \frac{\sum_{(X_i,y)\in O_k} I(h_k(x)=y}{\sum_{(X_i,y)\in O_k} I(h_k(X))}$$

رابطه (۶)

$$\widehat{p_k} = \frac{\sum_{(X_i,y)\in O_k} i(h_k(X)=\widehat{y_j}}{\sum_{(X_i,y)\in O_k} i(h_k(X)}$$

رابطه (۷)

در نهایت برای هر نمونه X از داده‌های آموزشی، با $Q(X,y_j)$ به‌صورت زیر محاسبه می‌گردد :

$$\widehat{y_j} = argmax_{\substack{j=1\\ j\neq y}}^{c} Q(X,y_j)$$

رابطه (۸)

اجرای الگوریتم‌های KNN و RFC

برای اجرا و پیاده سازی این الگوریتم‌ها از زبان پایتون نسخه ۳.۹ استفاده گردید. در پایتون کتابخانه SKLEARN، جزو کتابخانه‌های با ابزارهای بسیار زیاد و کاربردی در حوضه یادگیری ماشین می‌باشد که تمام الگوریتم‌های مورد استفاده برای این تسک در آن قرار داده شده است. که این امر سهولت استفاده از آن را فراهم می‌کند.

فراخوانی کتابخوانه‌ها و الگوریتم‌ها با دستورات زیر صورت گرفت.

```
from sklearn.neighbors import KNeighborsClassifier as knn
from sklearn.ensemble import RandomForestClassifier as rfc
```

پس از بارگذاری و پردازش اولیه داده‌های گل زنبق، برای آموزش الگوریتم‌ها باید داده‌ها به دو بخش آموزش و تست تقسیم می‌شدند. برای تست داده‌ها، یک سوم از کل دیتاست استفاده شد و مابقی برای آموزش داده‌ها استفاده می‌گردید. الگوریتم train_test_split برای این امر مورد استفاده قرار گرفت. متغیر x ، ستون‌های ۴ ویژگی گل زنبق و متغیر y، ستون هدف class که انواع گل زنبق در آن آمده می‌باشد.

```
x_train, x_test, y_train, y_test = tts(x,y,test_size=0.3,random_state=2)
```

پس از تقسیم داده‌ها، نوبت فراخوانی، آموزش الگوریتم و پیش بینی مقادیر تست می‌باشد که برای هر الگوریتم دستورات زیر خروجی مورد نظر را ایجاد کردند.

```
ns = rfc(bootstrap=True,random_state=20)
ns.fit(x_train,y_train)
pred = ns.predict(x_ test)
pred_results = pd.DataFrame({'expected_y':y_test,'predicted_y':pred})
```

```
ns = knn(n_neighbors=3)
ns.fit(x_train,y_train)
pred = ns.predict(x_test)
pred_results = pd.DataFrame({'expected_y':y_test ,'predicted_y':pred})
```

در نهایت خروجی‌های مورد نظر ذخیره گردید.

برای ارزیابی دقت الگوریتم‌ها، ۴ نمره accuracy,precision,IOU,confusion matrix مورد استفاده قرار گرفت. مقادیر خروجی نمره‌ها و ماتریس مورد نظر برای الگوریتم KNN به ترتیب، 0.954 ,0.972 ,0.978 بود و ماتریس در هم ریختگی نشان از وجود یک عدد خطا را داشت.

جدول ۱ ماتریس در هم ریختگی برای الگوریتم KNN

۱۶	۰	۰
۰	۱۷	۱
۰	۰	۰

و برای الگوریتم RFC ، به ترتیب 0.954 ,0.976 ,0.978 محاسبه شد. و ماتریس در هم ریختگی این الگوریتم نیز نشان از وجود یک عدد خطا را داشت.

جدول ۲ ماتریس در هم ریختگی برای الگوریتم RFC

۱۷	۰	۰
۰	۱۴	۱
۰	۰	۱۳

با توجه به امتیاز ارزیابی‌های موجود، دو الگوریتم با تقریب خیلی خوبی مشابه یک دیگر عمل می‌کنند و اختلاف جزئی نمره‌ها را می‌توان به تفاوت داده‌های تقسیم شده که به صورت تصادفی انجام گرفت، نسبت داد.

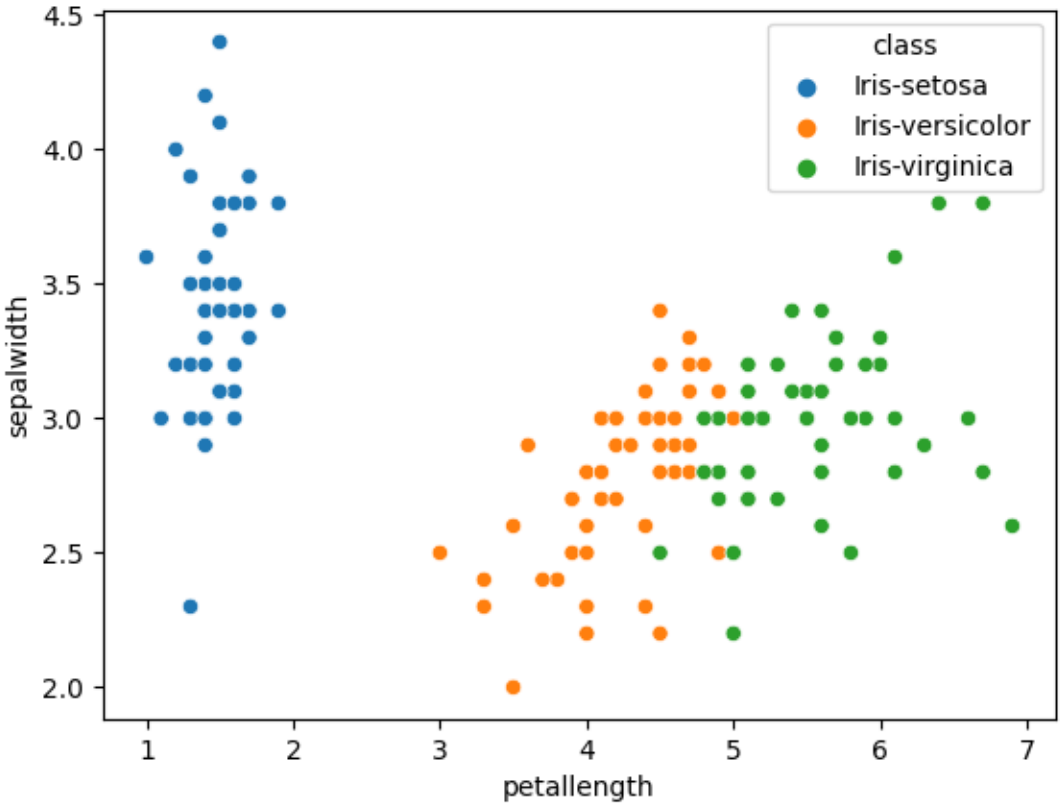
بهبود عملکرد پیش بینی الگوریتم‌ها

در دیتاست‌ها، عملا ویژگی‌های متفاوتی برای یک رکورد ثبت می‌گردد که الگوریتم‌های دسته بندی، به‌صورت پیشفرض وزن این ویژگی‌ها را یکسان می‌گیرد که این امر در مواجه با داده‌های با ابعاد بالا، دقت دسته بندی را کم می‌کند. برخی ویژگی‌ها را می‌توان مهم‌تر از بقیه ویژگی‌ها دانست. عده‌ای از این ستون‌ها مانند یک دیگر تغییر می‌کنند و یا عده‌ای نه تنها ممکن است اثر چندانی بر دقت دسته بندی الگوریتم‌ها نداشته باشند بلکه امکان دارد

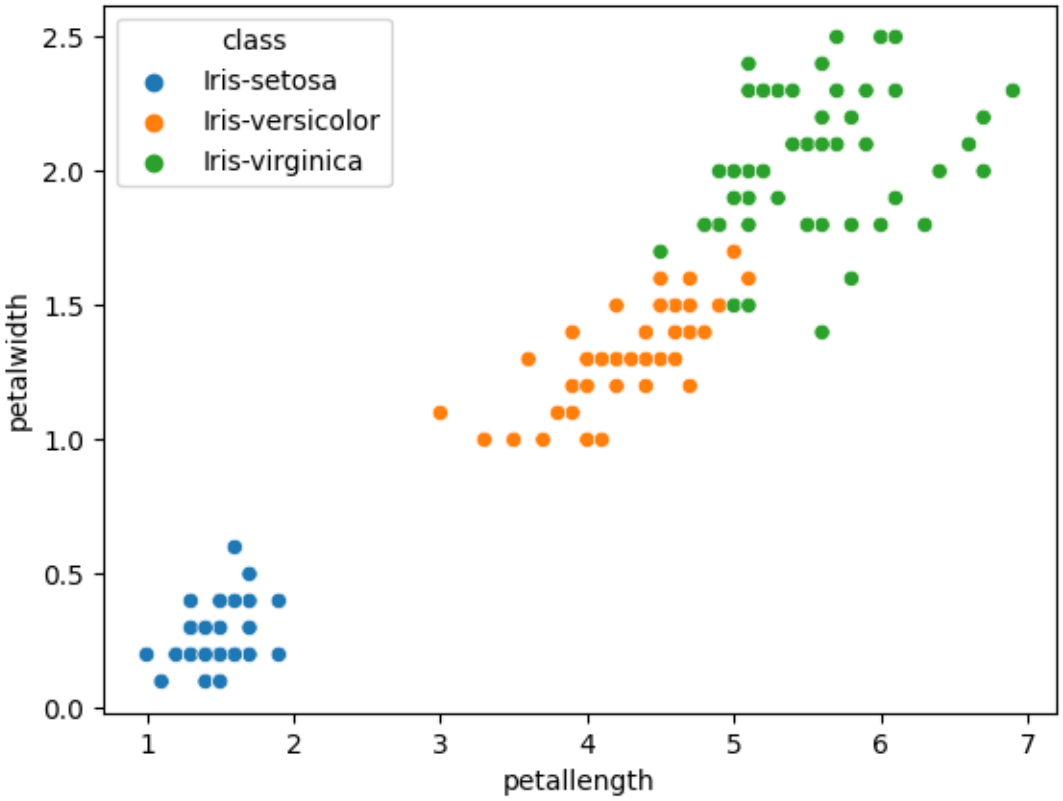
اثر منفی در عملکرد آنها نیز داشته باشند. خوشه بندی، یک کار مهم در داده کاوی محسوب می‌شود که در آن، اجزای مشابه با یک دیگر گروه بندی می‌گردند. [۲]

یکی از روش‌های خوشه بندی، یافتن فضاهای ویژگی است. فضاهای ویژگی با ارزیابی دو به دو ویژگی‌ها سعی در یافتن بهترین جفت ویژگی برای بهبود دقت عملکرد الگوریتم‌های دسته بندی، افزایش سرعت محاسبات و کاهش حجم دیتاست و محاسبات دارد. ضمن آنکه با یافتن ویژگی‌های اصلی، نویز و پراکندگی داده‌ها نیز کاهش می‌یابد.

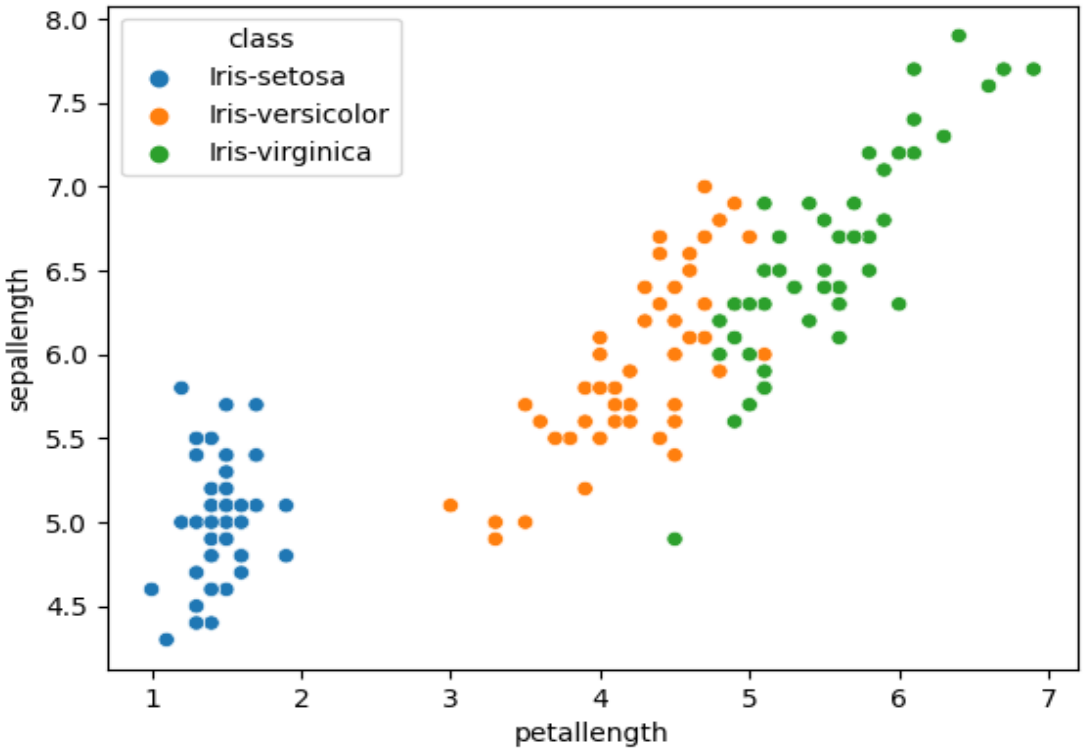
در این دیتا ست نیز، فضاهای ویژگی تشکیل شده و به‌صورت نموداری نمایش داده می‌شوند.



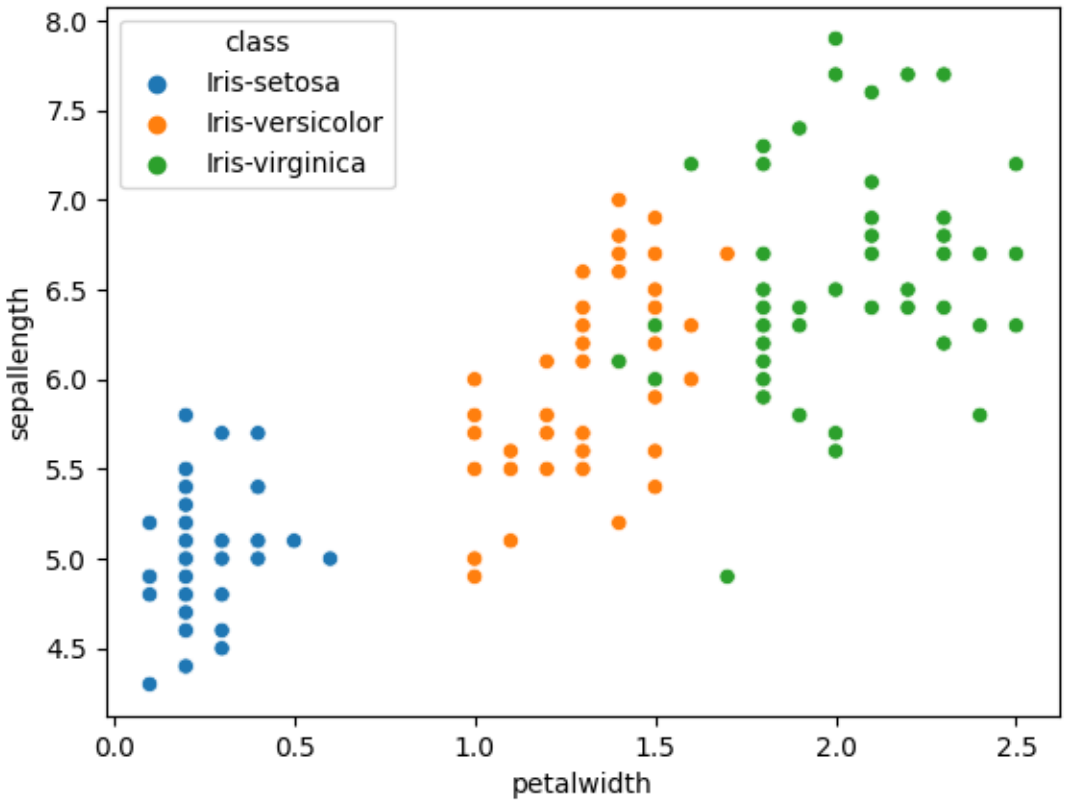
شکل ۱. فضای نمونه ای برای پارامتر sepalwidth و petallength



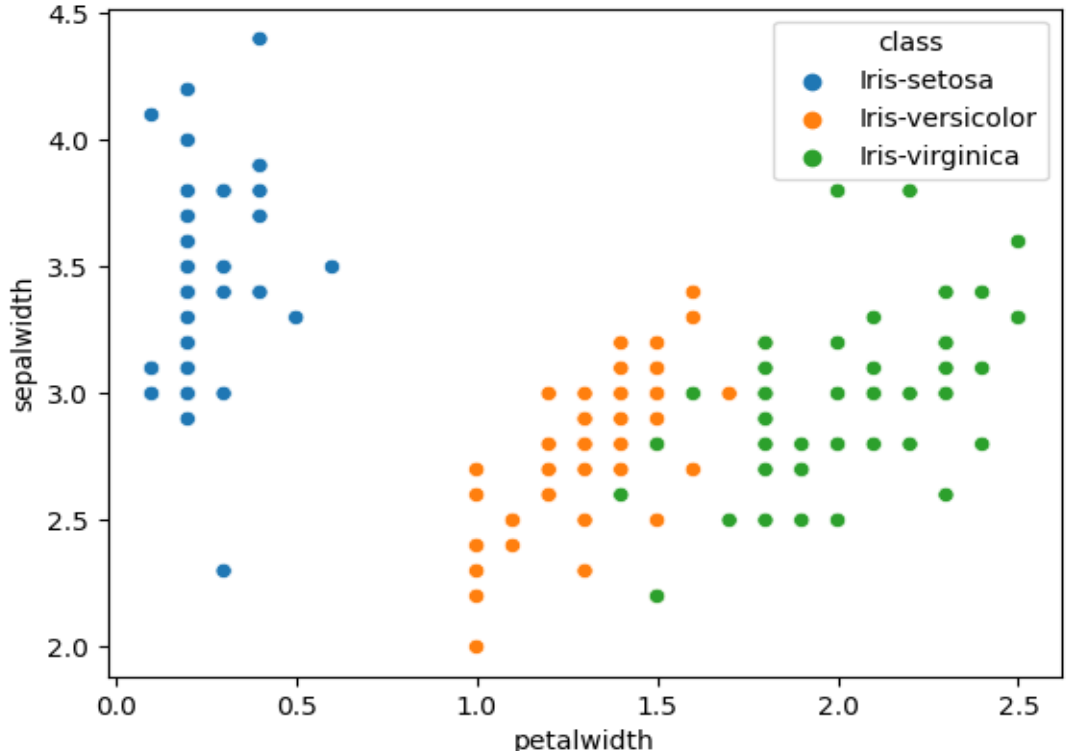
شکل ۲ فضای نمونه ای برای پارامتر petalwidth و petallength



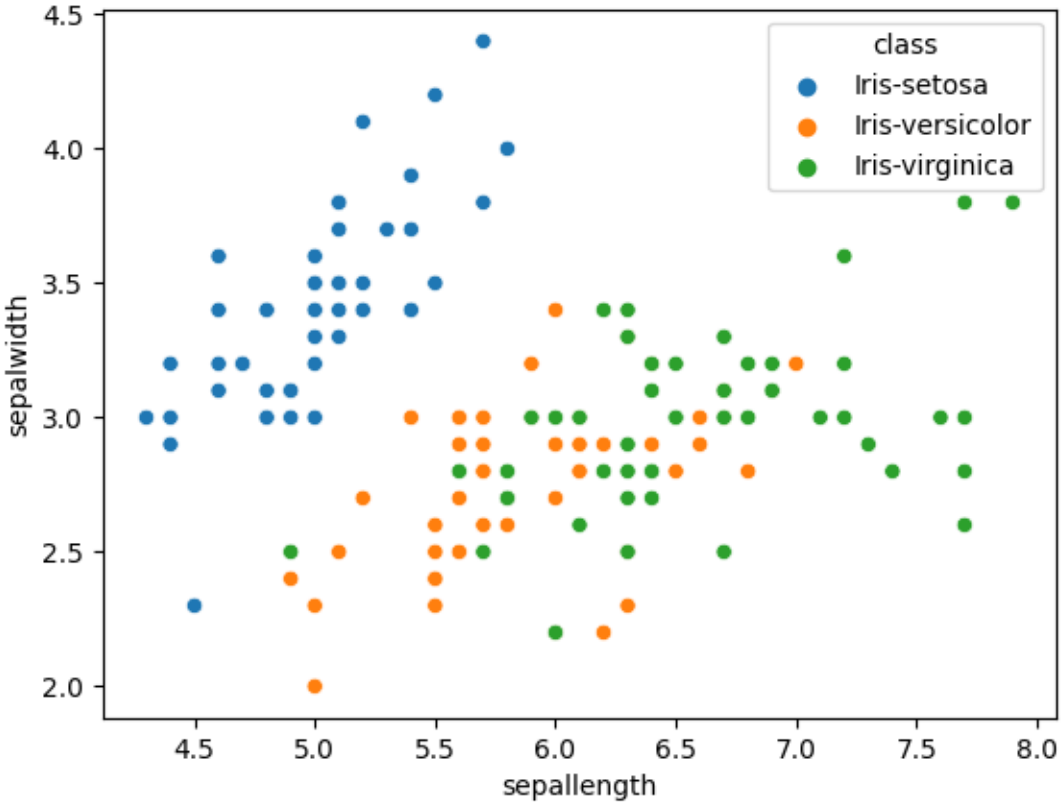
شکل ۳ فضای نمونه ای برای پارامتر sepallength و petallength



شکل ۴ فضای نمونه ای برای پارامتر petalwidth و sepallength



شکل ۵ فضای نمونه ای برای پارامتر sepalwidth و petallwidth



شکل ٦ فضای نمونه ای برای پارامتر sepalwidth و sepallength

در پارامترهای فضاهای نمونه ای اول تا پنجم، قابلیت جداسازی گونه Iris-sesota با ترسیم آستانه‌های جداکننده عمودی به‌صورت کامل قابل جداسازی است. اما در پارامتر فضای ششم، با ترسیم آستانه‌های عمودی یا افقی قابل جداسازی نخواهد بود.

در پارامتر فضای نمونه ای اول، یک نقطه آبی رنگ درانتهای جدول به عنوان داده پرت محسوب می‌گردد. با ترسیم آستانه جداکننده عمودی، نقاط سبز و نارنجی قابلیت تفکیک دارند اما در هر دو دسته بندی از رنگ دیگر نیز وجود دارد. این به این معنا است که احتمال خطای دسته بندی در این شکل بالا است.

در پارامتر فضای ویژگی دوم، با ترسیم آستانه جداکننده افقی از عرض ۱.۶ و طول ۴.۹، تقسیم بندی خوبی صورت می‌گیرد. مشاهده می‌گردد که تنها دو نقطه نارنجی، در کلاس سبز قرار میگیرند. پس در این فضا، در دسته بندی نارنجی و آبی، صد درصد داده‌ها درست تقسیم بندی می‌گردند و دسته بندی سبز با تقریب حدود ۹۵ درصد، انجام می‌گیرد که دقت بالایی است.

در پارامتر فضای ویژگی سوم، ترسیم حد آستانه‌ای جداکننده افقی، قابلیت جداسازی دسته سبز و نارنجی را ندارد اما، در طول ۴.۸، قابلیت تفکیک حدودی دو دسته باقی مانده وجود دارد. اما برخلاف فضای قبلی، تعداد بیشتر از نقاط نارنجی جزو دسته بندی سبز قرار گرفتند که این امر منجر به کاهش دقت دسته بندی سبز و کوچک شدن ابعاد دسته بندی نارنجی می‌گردد. پایین ترین نقطه

سبز رنگ نیز به عنوان داده پرت مشخص میگردد که با این امر، در گروه نارنجی، صد درصد داده همچنان یک دست هستند. اما دقت این فضا از فضای قبلی کمتر است.

پارامتر فضای چهارم، رفتاری مانند فضای سوم دارد اما واریانس هر دسته بزرگتر است. در انتخاب فضای ویژگی مطلوب هرچه واریانس درون کلاسی کمتر باشد، فضای مطلوب تری ایجاد می‌شود. پس بر این اساس، فضای سوم نسبت به این پارامتر انتخاب بهتری محسوب می‌گردد.

پارامتر فضای ویژگی پنجم ، با ترسیم آستانه جداکننده عمودی در طول ۱.۷، مشاهده می‌شود که هم دسته بندی سبز و هم دسته بندی نارنجی، به‌صورت ناخالص تقسیم بندی شده اند. این امر منجر به پایین بودن دقت الگوریتم‌ها نسبت به دیگر پارامترها می‌گردد.

در پارامتر دسته بندی ششم، ترسیم آستانه جداکننده خطوط عمودی یا افقی، قابلیت تقسیم بندی هیچ یک از دسته‌ها را با دقت بالا ندارد. اما با ترسیم خطوط رگرسیونی، دسته آبی به‌صورت کامل قابل تفکیک است. واریانس داده‌های سبز و آبی بسیار در هم آمیخته شده و توانایی ترسیم خطوط تقسیم بندی میان این دو دسته وجود ندارد. به همین علت بیان می‌گردد که قدرت شناسایی دسته بندی‌های این خوشه، تنها جداسازی دسته آبی با دیگر داده‌ها می‌باشد.

پس از بررسی فضاهای نمونه، فضاهای نمونه دوم و سوم، به عنوان فضاهای نمونه مطلوب بررسی شدند.

جدول3 مقادیر نمره های ارزیابی برای پارامترهای فضاهای ویژگی انتخابی

	Knn						RFC					
	petalwidth to petal length			sepal length to petal length			petalwidth to petal length			sepal length to petal length		
accuracy	0.977777778			0.933333333			0.977777778			0.933333333		
percision	0.972222222			0.931578947			0.972222222			0.931578947		
IOU	0.953703704			0.866666667			0.953703704			0.866666667		
conf mat	16	0	0	16	0	0	16	0	0	16	0	0
	0	17	1	0	17	1	0	17	1	0	17	1
	0	0	11	0	2	9	0	0	11	0	2	9

با بررسی فضاهای ویژگی مورد نظر، مشاهده شد که فضای ویژگی دوم، تغییری در دقت الگوریتم ایجاد نکرده اما فضای ویژگی سوم، منجر به کاهش دقت الگوریتم شده است.

فهرست منابع

[1] James, G. and Witten, D. and Hastie, T. and Tibshirani, R. (2021). An Introduction to Statistical

[2] Dash, M et al. (2000). Feature Selection for Clustering. Springer,1805, pp.2-3

[3] Robnik-ˇSikonja, M. (2004). Improving Random Forests. Springer, 3201, pp. 3-5

[4] Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 160(2), p. 2

[5] Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 160(2), p. 4