به نام خدا

سیستم‌های هوشمند (دکتر رشاد حسینی)

تمرین سری **دوم**

نیم‌سال اول ۱۴۰۲-۱۴۰۱

عرفـان پـنـــاهی ۸۱۰۱۹۸۳۶۹

این فایل شامل گزارش و نتایج شبیه سازی های انجام شده است.

***** فایل شبیه سازی با پایتون مربوط به هر قسمت این تمرین با عنوان HW2_Q#x_810198369.ipynb پیوست شده است.**

سوال ۱: درخت تصمیم (تحلیلی) (لینک گزارش)

سوال ۲: پیاده‌سازی الگوریتم درخت تصمیم (لینک گزارش)

سوال ۳: یادگیری براساس معیار (لینک گزارش)

سوال ۱: درخت تصمیم (تحلیلی)

قسمت الف: طراحی طبقه بند

در الگوریتم ID3، ابتدا می‌خواهیم ریشه (گره مادر – مرجع) را برای درخت تصمیم بدست آوریم. به همین منظور هر ویژگی را بعنوان یک گره در نظر می‌گیریم و سپس در این حالت میزان ابهامی از برچسب داده که برطرف می‌شود (آنتروپی) را بدست می‌آوریم. بدیهی است که در هر مرحله هر ویژگی که آنتروپی کمتری دارد (ابهام بیشتری را از ابهام گره بالایی برطرف کرده است) به‌عنوان گره در نظر می‌گیریم. جدول ۱، کلیه داده‌های آموزش را نشان می‌دهد.

شماره	رنگ	تعداد پا	قد	محل زندگی	جاندار
۱	قهوه‌ای	۲	بلند	خشکی	A
۲	قهوه‌ای	۳	کوتاه	خشکی	B
۳	سبز	۲	بلند	آب	B
۴	سبز	۳	بلند	آب	B
۵	قهوه‌ای	۲	کوتاه	آب	A
۶	قهوه‌ای	۲	بلند	آب	A
۷	قهوه‌ای	۲	کوتاه	خشکی	B
۸	سبز	۲	کوتاه	آب	A
۹	سبز	۳	بلند	آب	B
۱۰	قهوه‌ای	۲	بلند	خشکی	A

جدول ۱: داده های آموزش

ابتدا آنتروپی باینری را به صورت زیر در نظر می‌گیریم:

$$H_{binary} = h_b(p) = p\log(p) + (1-p)\log(1-p)$$

به این ترتیب گره ریشه، ویژگی‌ای خواهد بود که کمترین آنتروپی را دارد.

$$H_{\text{رنگ}} = p_{\text{سبز}} H_{\text{سبز}} + p_{\text{قهوه‌ای}} H_{\text{قهوه‌ای}}$$

$$H_{\text{سبز}} = h_b\left(p_{\text{سبز},A}\right) = h_b\left(\frac{1}{4}\right) = 0.8113$$

$$H_{\text{قهوه‌ای}} = h_b\left(p_{\text{قهوه‌ای},A}\right) = h_b\left(\frac{1}{3}\right) = 0.9183$$

$$\rightarrow \boldsymbol{H}_{\text{رنگ}} = \frac{\mathbf{4}}{\mathbf{10}}\boldsymbol{H}_{\text{سبز}} + \frac{\mathbf{6}}{\mathbf{10}}\boldsymbol{H}_{\text{قهوه‌ای}} = \mathbf{0.8775}$$

$$H_{\text{تعداد پا}} = p_2H_2 + p_3H_3$$

$$H_2 = h_b(p_{2,A}) = h_b\left(\frac{2}{7}\right) = 0.8631$$

$$H_3 = h_b(p_{3,A}) = h_b(1) = 0$$

$$\rightarrow \boldsymbol{H}_{\text{تعداد پا}} = \frac{\mathbf{7}}{\mathbf{10}}\boldsymbol{H}_{\mathbf{2}} + \frac{\mathbf{3}}{\mathbf{10}}\boldsymbol{H}_{\mathbf{3}} = \mathbf{0.6042}$$

$$H_{\text{قد}} = p_{\text{بلند}}H_{\text{بلند}} + p_{\text{کوتاه}}H_{\text{کوتاه}}$$

$$H_{\text{بلند}} = h_b\left(p_{\text{بلند},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$H_{\text{کوتاه}} = h_b\left(p_{\text{کوتاه},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$\rightarrow \boldsymbol{H}_{\text{قد}} = \frac{\mathbf{4}}{\mathbf{10}}\boldsymbol{H}_{\text{کوتاه}} + \frac{\mathbf{6}}{\mathbf{10}}\boldsymbol{H}_{\text{بند}} = \mathbf{1}$$

$$H_{\text{محل زندگی}} = p_{\text{آب}}H_{\text{آب}} + p_{\text{خشکی}}H_{\text{خشکی}}$$

$$H_{\text{آب}} = h_b\left(p_{\text{آب},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

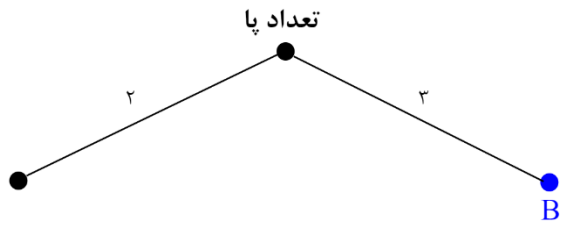
$$H_{\text{خشکی}} = h_b\left(p_{\text{خشکی},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$\rightarrow \boldsymbol{H}_{\text{محل زندگی}} = \frac{\mathbf{6}}{\mathbf{10}}\boldsymbol{H}_{\text{آب}} + \frac{\mathbf{4}}{\mathbf{10}}\boldsymbol{H}_{\text{خشکی}} = \mathbf{1}$$

در نتیجه ویژگی ریشه درخت تصمیم، تعداد پا خواهد بود.

تعداد پا = گره ریشه

در این صورت درخت بعد از مرحله اول مطابق با تصویر ۱-۱ خواهد شد.



تصویر ۱-۱: درخت تصمیم بعد از اولین مرحله (تشخیص گره مادر)

حال داده های مربوط به هر شاخه را جدا می‌کنیم. همانطور که در تصویر ۱-۱ نیز مشاهده می‌شود، جانداران با ۳ پا همه از نوع B هستند. حال ادامه درخت را برای جانداران ۲ پا بدست می‌آوریم. به این ترتیب جدول ۲ جانداران ۲ پا را نشان می‌دهد.

شماره	رنگ	قد	محل زندگی	جاندار
۱	قهوه‌ای	بلند	خشکی	A
۳	سبز	بلند	آب	B
۵	قهوه‌ای	کوتاه	آب	A
۶	قهوه‌ای	بلند	آب	A
۷	قهوه‌ای	کوتاه	خشکی	B
۸	سبز	کوتاه	آب	A
۱۰	قهوه‌ای	بلند	خشکی	A

جدول ۲: داده های آموزش با ۲ پا

$$H_{\text{رنگ}} = p_{\text{سبز}} H_{\text{سبز}} + p_{\text{قهوه‌ای}} H_{\text{قهوه‌ای}}$$

$$H_{\text{سبز}} = h_b\left(p_{\text{سبز},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$H_{\text{قهوه‌ای}} = h_b\left(p_{\text{قهوه‌ای},A}\right) = h_b\left(\frac{4}{5}\right) = 0.7219$$

$$\rightarrow \boldsymbol{H_{\text{رنگ}} = \frac{2}{7} H_{\text{سبز}} + \frac{5}{7} H_{\text{قهوه‌ای}} = 0.8014}$$

$$H_{\text{قد}} = p_{\text{بلند}} H_{\text{بلند}} + p_{\text{کوتاه}} H_{\text{کوتاه}}$$

$$H_{\text{بلند}} = h_b\left(p_{\text{بلند},A}\right) = h_b\left(\frac{1}{4}\right) = 0.8113$$

$$H_{\text{کوتاه}} = h_b\left(p_{\text{کوتاه},A}\right) = h_b\left(\frac{1}{3}\right) = 0.9183$$

$$\rightarrow \boldsymbol{H_{\text{قد}} = \frac{4}{7} H_{\text{کوتاه}} + \frac{3}{7} H_{\text{بند}} = 0.8571}$$

$$H_{\text{محل زندگی}} = p_{\text{آب}} H_{\text{آب}} + p_{\text{خشکی}} H_{\text{خشکی}}$$

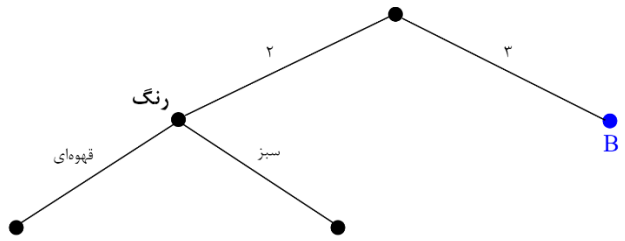
$$H_{\text{آب}} = h_b\left(p_{\text{آب},A}\right) = h_b\left(\frac{1}{4}\right) = 0.8113$$

$$H_{\text{خشکی}} = h_b\left(p_{\text{خشکی},A}\right) = h_b\left(\frac{1}{3}\right) = 0.9183$$

$$\rightarrow \boldsymbol{H_{\text{محل زندگی}} = \frac{4}{7} H_{\text{آب}} + \frac{3}{7} H_{\text{خشکی}} = 0.8571}$$

رنگ = گره میانی اول

در این صورت درخت بعد از مرحله اول مطابق با تصویر ۱-۲ خواهد شد.



تصویر ۱-۲: درخت تصمیم بعد از تشخیص گره میانی اول

حال داده های مربوط به هر شاخه را جدا می‌کنیم. مطابق با تصویر ۱-۲، ابتدا درخت در ادامه شاخه قهوه‌ای را مشخص می‌کنیم.

جدول ۳، داده های آموزش مربوط به جانداران ۲ پا و قهوه‌ای رنگ را نشان می‌دهد.

شماره	قد	محل زندگی	جاندار
۱	بلند	خشکی	A
۵	کوتاه	آب	A
۶	بلند	آب	A
۷	کوتاه	خشکی	B
۱۰	بلند	خشکی	A

جدول ۳: داده های آموزش با ۲ پا و قهوه‌ای رنگ

$$H_{\text{قد}} = p_{\text{بلند}} H_{\text{بلند}} + p_{\text{کوتاه}} H_{\text{کوتاه}}$$

$$H_{\text{بلند}} = h_b\left(p_{\text{بلند},A}\right) = h_b(1) = 0$$

$$H_{\text{کوتاه}} = h_b\left(p_{\text{کوتاه},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$\rightarrow \boldsymbol{H}_{\text{قد}} = \frac{\mathbf{2}}{\mathbf{5}}\boldsymbol{H}_{\text{کوتاه}} + \frac{\mathbf{3}}{\mathbf{5}}\boldsymbol{H}_{\text{بند}} = \mathbf{0.4}$$

$$H_{\text{محل زندگی}} = p_{\text{آب}} H_{\text{آب}} + p_{\text{خشکی}} H_{\text{خشکی}}$$

$$H_{\text{آب}} = h_b\left(p_{\text{آب},A}\right) = h_b(1) = 0$$

$$H_{\text{خشکی}} = h_b\left(p_{\text{خشکی},A}\right) = h_b\left(\frac{1}{3}\right) = 0.9183$$

$$\rightarrow \boldsymbol{H}_{\text{محل زندگی}} = \frac{\mathbf{4}}{\mathbf{7}}\boldsymbol{H}_{\text{آب}} + \frac{\mathbf{3}}{\mathbf{7}}\boldsymbol{H}_{\text{خشکی}} = \mathbf{0.5510}$$

قد = گره میانی $\boldsymbol{I}$

حال مطابق با جدول ۴ داده های مربوط به رنگ سبز را جدا می‌کنیم و ویژگی گره II را بدست می‌آوریم.

شماره	قد	محل زندگی	جاندار
۳	بلند	آب	B
۸	کوتاه	آب	A

جدول ۴: داده های آموزش با ۲ پا و سبز رنگ

$$H_{\text{قد}} = p_{\text{بلند}} H_{\text{بلند}} + p_{\text{کوتاه}} H_{\text{کوتاه}}$$

$$H_{\text{بلند}} = h_b\left(p_{\text{بلند},A}\right) = h_b(1) = 0$$

$$H_{\text{کوتاه}} = h_b\left(p_{\text{کوتاه},A}\right) = h_b(1) = 0$$

$$\rightarrow \boldsymbol{H}_{\text{قد}} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{H}_{\text{کوتاه}} + \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{H}_{\text{بند}} = \mathbf{0}$$

$$H_{\text{محل زندگی}} = p_{\text{آب}} H_{\text{آب}} + p_{\text{خشکی}} H_{\text{خشکی}}$$

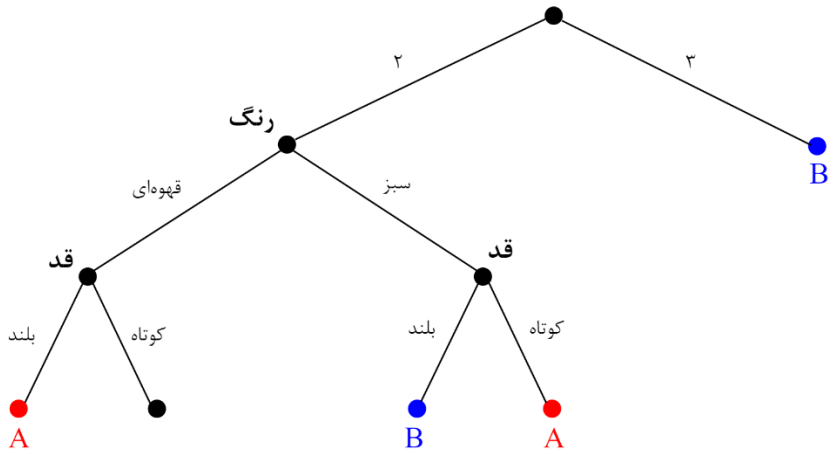
$$H_{\text{آب}} = h_b\left(p_{\text{آب},A}\right) = h_b\left(\frac{1}{2}\right) = 1$$

$$H_{\text{خشکی}} = h_b\left(p_{\text{خشکی},A}\right) = h_b(1) = 0$$

$$\rightarrow \boldsymbol{H}_{\text{محل زندگی}} = \boldsymbol{H}_{\text{آب}} + \mathbf{0}\boldsymbol{H}_{\text{خشکی}} = \mathbf{1}$$

قد = گره میانی II

تصویر ۱-۳، درخت تصمیم بعد از مشخص شدن گره میانی های دوم را نشان می‌دهد.



تصویر ۱-۳: درخت تصمیم بعد از مشخص شدن گره میانی های دوم

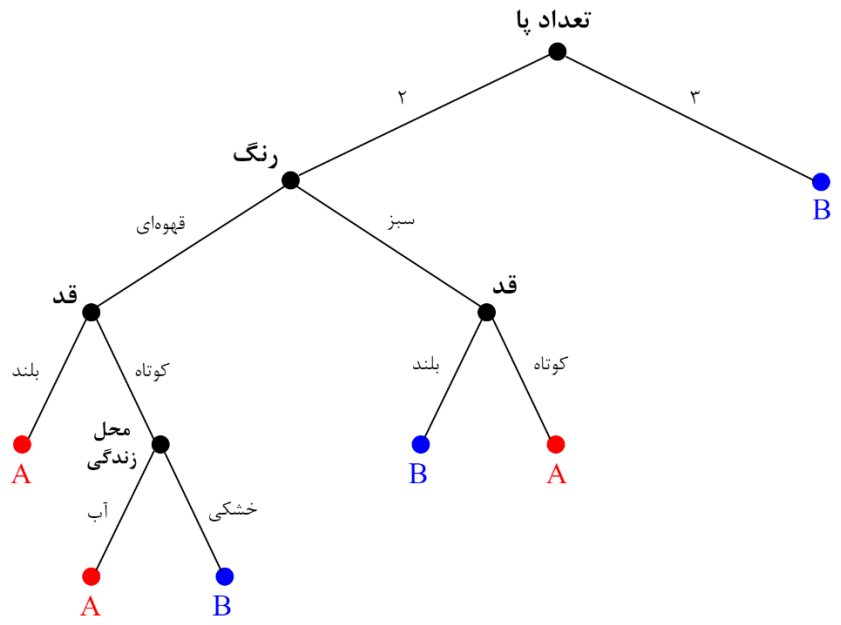
جدول ۵، داده های آموزش برای جانداران ۲ پا، قهوه‌ای رنگ و کوتاه را نشان می‌دهد.

شماره	محل زندگی	جاندار
۵	آب	A
۷	خشکی	B

جدول ۵: داده های آموزش برای جانداران ۲ پا، قهوه‌ای رنگ و کوتاه

به همین ترتیب ویژگی گره آخر نیز که محل زندگی است مشخص می‌شود و همه داده های آموزش با این درخت تصمیم به درستی طبقه بندی می‌شوند.

تصویر ۱-۴، درخت تصمیم آموزش داده شده با الگوریتم ID3 را نشان می‌دهد.



تصویر ۱-۴: درخت تصمیم آموزش داده شده با الگوریتم ID3

قسمت ب: آزمون طبقه بند

در این قسمت می‌خواهیم ماتریس آشفتگی (confusion matrix) را برای داده های آزمون جدول ۶ بدست آوریم.

شماره	رنگ	تعداد پا	قد	محل زندگی	جاندار
۱	قهوه‌ای	۳	بلند	خشکی	B
۲	سبز	۲	بلند	خشکی	A
۳	سبز	۲	کوتاه	خشکی	A
۴	قهوه‌ای	۲	کوتاه	آب	B
۵	قهوه‌ای	۲	بلند	خشکی	A

جدول ۶: داده های آزمون

برای هرکدام از داده های بالا با استفاده از درخت تصمیم قسمت اول برچسب جاندار را مشخص می‌کنیم و سپس با نوع جاندار (برچسب واقعی) مقایسه می‌کنیم.

1 شماره : تعداد پا $= 3 \rightarrow$ برچسب : $B \rightarrow$ درست

2 شماره : تعداد پا $= 2 \rightarrow$ رنگ : سبز $\rightarrow$ قد : بلند $\rightarrow$ برچسب : $B \rightarrow$ غلط

3 شماره : تعداد پا $= 2 \rightarrow$ رنگ : سبز $\rightarrow$ قد : کوتاه $\rightarrow$ برچسب : $A \rightarrow$ درست

4 شماره : تعداد پا $= 2 \rightarrow$ رنگ : قهوه‌ای $\rightarrow$ قد : کوتاه $\rightarrow$ محل زندگی : آب $\rightarrow$ برچسب : $A \rightarrow$ غلط

4 شماره : تعداد پا $= 2 \rightarrow$ رنگ : قهوه‌ای $\rightarrow$ قد : بلند $\rightarrow$ برچسب : $A \rightarrow$ درست

ماتریس آشفتگی (confusion matrix) درخت تصمیم برای داده های آزمون مطابق با تصویر ۵ خواهد بود.

Predict \ True	A	B
A	2	1
B	1	1

تصویر ۱-۵: ماتریس آشفتگی (confusion matrix)

همانطور که در تصویر ۱-۵ مشاهده می‌شود، عملکرد درخت تصمیم برای داده آزمون دقت معمولی دارد و ممکن است نتواند به خوبی برچسب جاندار را مشخص کند.

قسمت ج: رویکرد حریضانه الگوریتم ID3

نمی‌توان از دو ویژگی برای طبقه بندی جانداران استفاده کرد به طوری که خطا صفر شود. همانطور که در قسمت الف نیز بدست آمد، ۲ ویژگی بهتر اول تعداد پا و رنگ هستند و بعد از گره آن‌ها هنوز نوع جاندار مشخص نمی‌شود. زیرا احتمال مشترک بودن ۲ ویژگی برای ۲ جاندار متفاوت زیاد است. برای مثال جانداران شماره ۷ و ۱۰ از نظر رنگ و تعداد پا یکسان هستند اما برچسب یکسانی ندارند.

قسمت د: افزایش قوام طبقه بند

در هنگام آموزش طبقه بند درخت تصمیم بخاطر زیاد بودن داده های آموزش ممکن است علی رغم اینکه دقت طبقه بند روی داده های آموزش ۱۰۰٪ شود، دچار بیش برازش (overfitting) شود. دلیل این امر تعداد دفعات تقسیم کردن (splitting) و استفاده از داده های زیاد برای آموزش طبقه بند است.

از راهکار های مقابله با این مشکل می‌توان به دو مورد زیر اشاره کرد:

۱- صرف نظر از استفاده از همه داده های آموزش در فرایند تقسیم کردن (splitting)

۲- طراحی کلی درخت تصمیم و سپس حذف تعدادی از شاخه ها و گره های میانی

منبع:

- https://towardsdatascience.com/construct-a-decision-tree-and-how-to-deal-with-overfitting-f907efc1492d

---

سوال ۲: پیاده‌سازی الگوریتم درخت تصمیم

دادگان کشتی تایتانیک: از داده های تحت عنوان train-data.csv، ۸۰٪ را به عنوان دادهٔ آموزشی و ۲۰٪ را به عنوان دادهٔ آموزش انتخاب می‌کنیم. به این منظور از تابع train_test_split (کتابخانهٔ sklearn.model_selection) استفاده می‌کنیم.

قسمت الف: پیاده‌سازی مدل درخت تصمیم

ویژگی های مؤثر در تشخیص نجات یافتگان کشتی تایتانیک: با توجه به ۷ ویژگی داده شده، مهم ترین ویژگی ها جنسیت، سن و همچنین تعداد افراد همراه می‌باشد. سایر ویژگی ها مثل بندر سوار شدن در تشخیص نجات یافتگان ممکن است خیلی مهم نباشند. برای طراحی درخت تصمیم سعی می‌کنیم از هر ۷ ویژگی استفاده کنیم. در نهایت به این نتیجه می‌رسیم که زیاد بودن ویژگی ها و وجود برخی ویژگی های غیر مؤثر باعث فرابرازش (overfitting) و پایین آمدن دقت درخت تصمیم می‌شود.

پیش پردازش داده‌ها: در این قسمت داده های پارامتری را به داده های عددی تبدیل می‌کنیم. برای مثال ۳ بندر سوار شدن به کشتی که S، Q و C هستند را با اعداد ۰، ۱ و ۲ مدل می‌کنیم. همچنین برای داده های از بین رفته در یک ویژگی (NaN) از میانه داده های آن ویژگی استفاده می‌کنیم.

الگوریتم طراحی درخت تصمیم: برای اینکه بتوانیم ویژگی ها را از حالت پیوسته و گسسته به حالت باینری ببریم، سعی می‌کنیم بین تمامی داده های هر ویژگی، مرزی را انتخاب کنیم که کمترین آنتروپی را نتیجه دهد. داده های کوچکتر از مرز را با 0 و داده های بزرگتر از مرز را با 1 مشخص می‌کنیم. سپس به‌مانند بخش اول تمرین، در هر گره از درخت، سعی می‌کنیم ویژگی ای را انتخاب کنیم که منجر به کمترین آنتروپی شود. به‌عبارت دیگر بیشترین ابهام نسبت به برچسب داده برطرف شود. در نهایت گره های تعیین شده در هر طبقه را به‌هم متصل کرده و طبقه بندی را با استفاده از مرزهای مشخص شده انجام می‌دهیم. در الگوریتم کلی تعداد ماکزیمم طبقات درخت را به عنوان یکی از شروط توقف و رسیدن به برگ های درخت در نظر می‌گیریم و به همین منظور دقت و ماتریس آشفتگی (confusion matrix) را برای مقادیر مختلف تعداد طبقات درخت بدست می‌آوریم.

تصویر ۲-۱ دقت و ماتریس آشفتگی را برای تعداد طبقات مختلف درخت نشان می‌دهد. همانطور که مشاهده می‌شود در صورتی که تعداد طبقات را کم در نظر بگیریم چون از ویژگی های کمتری در طبقه بندی استفاده کرده‌ایم دقت کمی داریم. از طرف دیگر اگر تعداد طبقات خیلی زیاد شود، طبقه بند مقاومت خود را در برابر فرابرازش (overfitting) از دست می‌دهد و دقت آن روی داده های تست کمتر می‌شود.

```
Maximum Depth : 4
 accuracy = 81.00558659217877 %
 confusion matrix:
 [[109  32]
 [  2  36]]

Maximum Depth : 3
 accuracy = 79.3296089385475 %
 confusion matrix:
 [[109  35]
 [  2  33]]

Maximum Depth : 6
 accuracy = 82.12290502793296 %
 confusion matrix:
 [[110  31]
 [  1  37]]

Maximum Depth : 5
 accuracy = 81.56424581005587 %
 confusion matrix:
 [[110  32]
 [  1  36]]

Maximum Depth : 7
 accuracy = 81.56424581005587 %
 confusion matrix:
 [[109  31]
 [  2  37]]
```

تصویر ۲-۱: ماتریس آشفتگی (confusion matrix) و دقت برای تعداد طبقات مختلف درخت تصمیم

قسمت ب: بهبود بخشی الگوریتم درخت تصمیم

همانطور که در بخش اول نیز گفته شد یکی از عواملی که با استفاده از آن می‌توان مانع از فرابرازش درخت تصمیم شد حذف برخی ویژگی ها است. می‌توانیم با حذف برخی ویژگی هایی که آنتروپی بیشتری دارند، دقت درخت تصمیم را افزایش دهیم. یکی دیگر از راهکار های مناسب برای افزایش دقت درخت تصمیم، طراحی جنگل تصادفی است. در الگوریتم جنگل تصادفی هنگامی که می‌خواهیم درخت را آموزش دهیم، از تعداد دلخواهی دادهٔ آموزشی به‌صورت رندوم استفاده می‌کنیم. به این ترتیب تعدادی درخت (مثلاً ۱۰ درخت) را طراحی می‌کنیم. سپس برای طبقه بندی، بین برچسب های خروجی از هر درخت جنگل، رأی اکثریت گرفته و برچسب نهایی را تعیین می‌کنیم. در **قسمت ج** این الگوریتم را پیاده سازی می‌نمائیم.

قسمت ج: استفاده از جنگل تصادفی

مطابق با الگوریتم گفته شده در قسمت ب، جنگل تصادفی را با ۱۰ درخت تعریف می‌کنیم. تصویر ۲-۲ ماتریس آشفتگی (confusion matrix) را به‌همراه دقت جنگل تصادفی نشان می‌دهد.

```
Random Forest with 10 trees :
 accuracy = 82.68156424581005 %
 confusion matrix:
 [[109  29]
 [  2  39]]
```

تصویر ۲-۲: ماتریس آشفتگی (confusion matrix) و دقت جنگل تصادفی

سوال ۳: یادگیری براساس معیار

*** طبقه بند k همسایه نزدیک

قسمت الف: طراحی طبقه بند

در این قسمت ابتدا به کمک ۸۰٪ داده‌ها (۱۳۶ داده)، تابع مربوط به طبقه بند kNN را می‌نویسیم. سپس ماتریس آشفتگی (confusion matrix) و دقت طبقه بند را برای مقادیر مختلف k بدست می‌آوریم. تصویر ۳-۱، ماتریس آشفتگی (confusion matrix) و دقت طبقه بند را برای مقادیر $k = 1,5,10,20$ نشان می‌دهد.

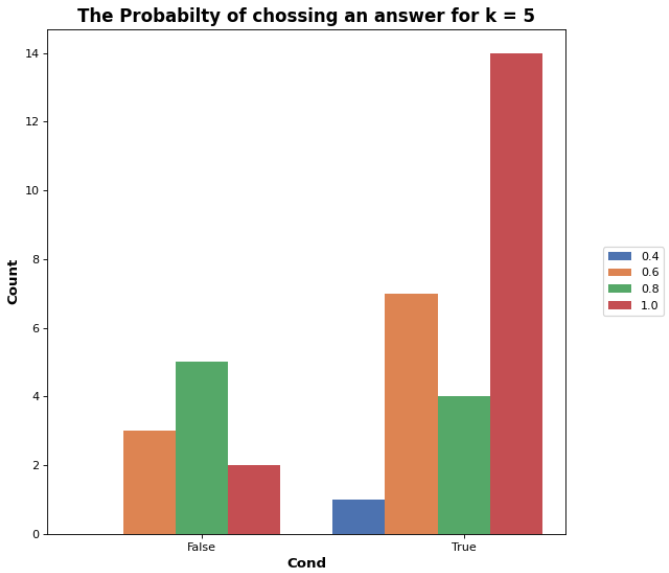
```
(k = 1)
 accuracy = 69.44444444444444 %
 cofusion matrix:
 [[10  2  0]
 [ 0  8  4]
 [ 1  4  7]]

(k = 5)
 accuracy = 69.44444444444444 %
 cofusion matrix:
 [[10  3  0]
 [ 0  9  5]
 [ 1  2  6]]
```

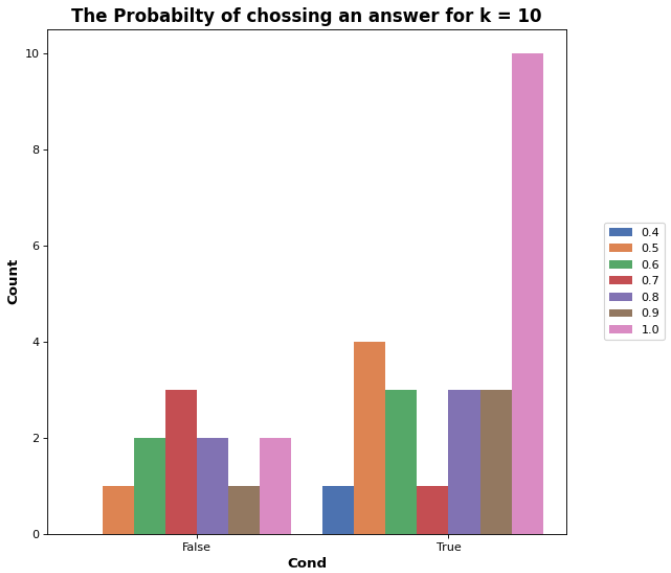
```
(k = 10)
 accuracy = 63.888888888888886 %
 cofusion matrix:
 [[11  3  1]
 [ 0  9  7]
 [ 0  2  3]]

(k = 20)
 accuracy = 80.55555555555556 %
 cofusion matrix:
 [[11  3  1]
 [ 0 10  2]
 [ 0  1  8]]
```

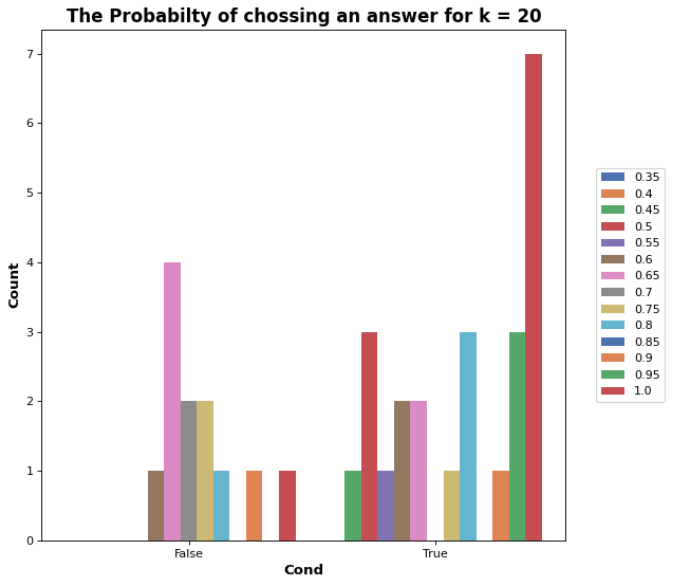
تصویر ۳-۱: ماتریس آشفتگی (confusion matrix) و دقت طبقه بند برای مقادیر k مختلف

قسمت ب: محاسبه توزیع احتمال تعلق به هر کلاس

در این قسمت مطابق با خواسته مسئله تابع Prob_knn نوشته شده است. تصاویر ۳-۲ نمودار مربوط به احتمال تعلق به هر کلاس را به ازای $k = 5,10,20$ نشان می‌دهد.







تصویر ۳-۲: نمودار مربوط به احتمال تعلق به هر کلاس را به ازای $k = 5,10,20$

همانطور که در تصویر ۳-۱ و ۳-۲ مشاهده می‌شود، برای مقادیر خیلی کوچک k ($k = 1$)، دقت خیلی مناسب نیست. زیرا احتمال اینکه یک داده تست نزدیک به یک دادهٔ آموزش غیرهمکلاس باشد وجود دارد. علاوه بر آن اگر k خیلی بزرگ شود، ممکن است دچار خطا شویم. برای مثال در صورتی که تعداد داده های آموزش از یک کلاس در یک موقعیت خاص کم باشد، برای یک k بزرگ، از کلاس مخالف داده های زیادی در محدوده داده تست قرار می‌گیرد و خطا رخ می‌دهد.

*** یادگیری بر اساس معیار

قسمت الف: بررسی کارکرد روش یادگیری

هدف از این دو روش، انتقال دادگان به فضای جدیدی است که داده های مربوط به هر دو کلاس مختلف فاصله بیشتری تا هم داشته باشند. به این منظور در هر روش یک مسئله بهینه سازی به همراه قیود تعریف می‌شود.

در ابتدا معیار Mahalanobis metric را بصورت زیر تعریف می‌کنیم:

$$d(\vec{x_i}, \vec{x_j}) = (\vec{x_i} - \vec{x_j})^T M (\vec{x_i} - \vec{x_j})$$

*** روش LMNN (Largest Margin Nearest Neighbor):

در این روش برای تعریف دادگان در فضای جدید، ابتدا برای یک داده تعداد k داده نزدیک را مشخص می‌کنیم و سپس داده های هم‌کلاس را نزدیک و داده های کلاس های مخالف را دور می‌کنیم.

حال مسئله بهینه سازی را به صورت زیر تعریف می‌کنیم. (λ یک ثابت مثبت و M یک ماتریس معین مثبت (قید III) است.)

$$\min_{m} \sum_{i,j \in N_i} d(x_i, x_j) + \lambda \sum_{i,j,l} \xi_{ijl}$$

$$(I)\ d(x_i, x_l) - d(x_i, x_j) \geq 1 - \xi_{ijl}\,, \qquad \forall i,j \in N_i, \qquad l: y_i \neq y_l$$

$$(II)\ \xi_{ijl} \geq 0$$

$$(III)\ M \geq 0$$

قید (I) و (II) بیان می‌کند که می‌خواهیم علاوه بر کم کردن فاصله داده های کلاس مشابه به هم، داده های کلاس های مخالف را دور کنیم.

*** روش LFDA (Local Fisher Discriminant Analysis):

در این روش برای تعریف دادگان در فضای جدید، سعی می‌کنیم به داده های هم‌کلاس وزن دهیم و به همین ترتیب آنها را در موقعیتی مستقل تر قرار دهیم. (در این روش همبستگی ویژگی های داده های دو کلاس مختلف تقریباً صفر می‌شود.)

به این ترتیب دو ماتریس را برای مسئله بهینه سازی به صورت زیر تعریف می‌کنیم.

$$A_{i,j}^{(\omega)} = \begin{cases} \dfrac{A_{i,j}}{n_c}, & if \quad y_i = y_j = c \\ 0\,, & if \quad y_i \neq y_j \end{cases}$$

$$A_{i,j}^{(\omega)} = \begin{cases} A_{i,j}\left(\dfrac{1}{n} - \dfrac{1}{n_c}\right), & if \quad y_i = y_j = c \\ \dfrac{1}{n}\,, & if \quad y_i \neq y_j \end{cases}$$

n_c تعداد داده های کلاس c و n تعداد کل داده ها را نشان می‌دهد.

$$S^{(W)} = \frac{1}{2} \sum_{i,j=1}^{n} A_{i,j}^{(\omega)} (x_i - x_j)(x_i - x_j)^T$$

$$S^{(B)} = \frac{1}{2} \sum_{i,j=1}^{n} A_{i,j}^{(b)} (x_i - x_j)(x_i - x_j)^T$$

حال مسئله بهینه سازی را به صورت زیر تعریف می‌کنیم.

$$T_{LFDA} = \underset{T}{\operatorname{argmax}} \left[trace\left(\left(T^T S^{(W)} T\right)^{-1} T^T S^{(W)} T \right) \right]$$

در نهایت با استفاده از T_{LFDA} به دست آمده، داده ها را به صورت زیر به فضای جدید تصویر می‌کنیم. (z_i ها بیانگر داده های تصویر شده اند.)

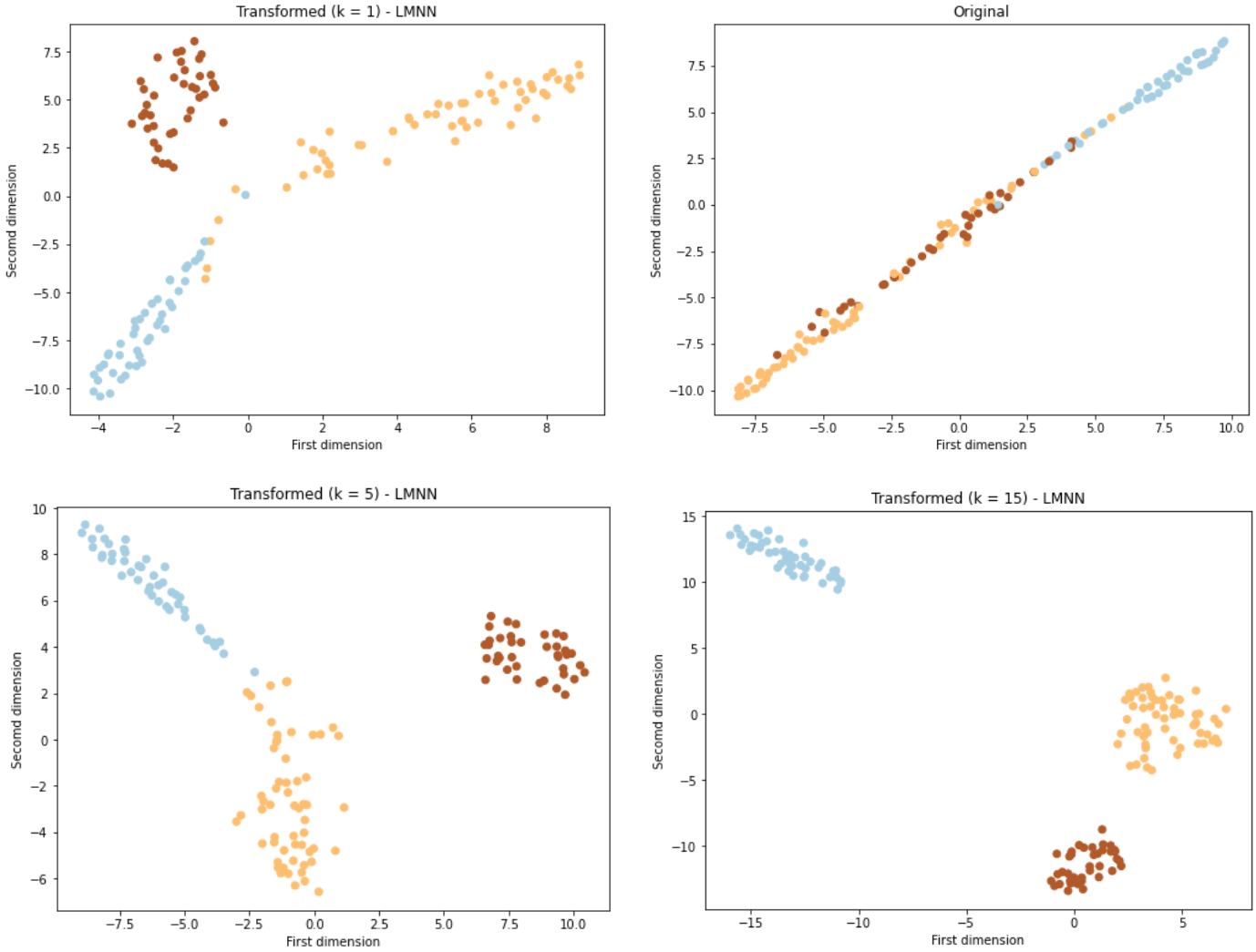
$$z_i = T_{LFDA}^T x_i$$

قسمت ب: ترسیم دادگان انتقال یافته در فضای جدید

سوال ۱. در دو روش یادگیری بحث شده، k تعداد نزدیک‌ترین همسایه های به داده منتخب هستند که سعی می‌کنیم داده های هم‌کلاس را به هم نزدیک و داده های کلاس های مخالف را از آن دور کنیم. اما در طبقه بند kNN سعی می‌کنیم با استفاده از اکثریت کلاس های k داده نزدیک، کلاس داده تست را مشخص کنیم.

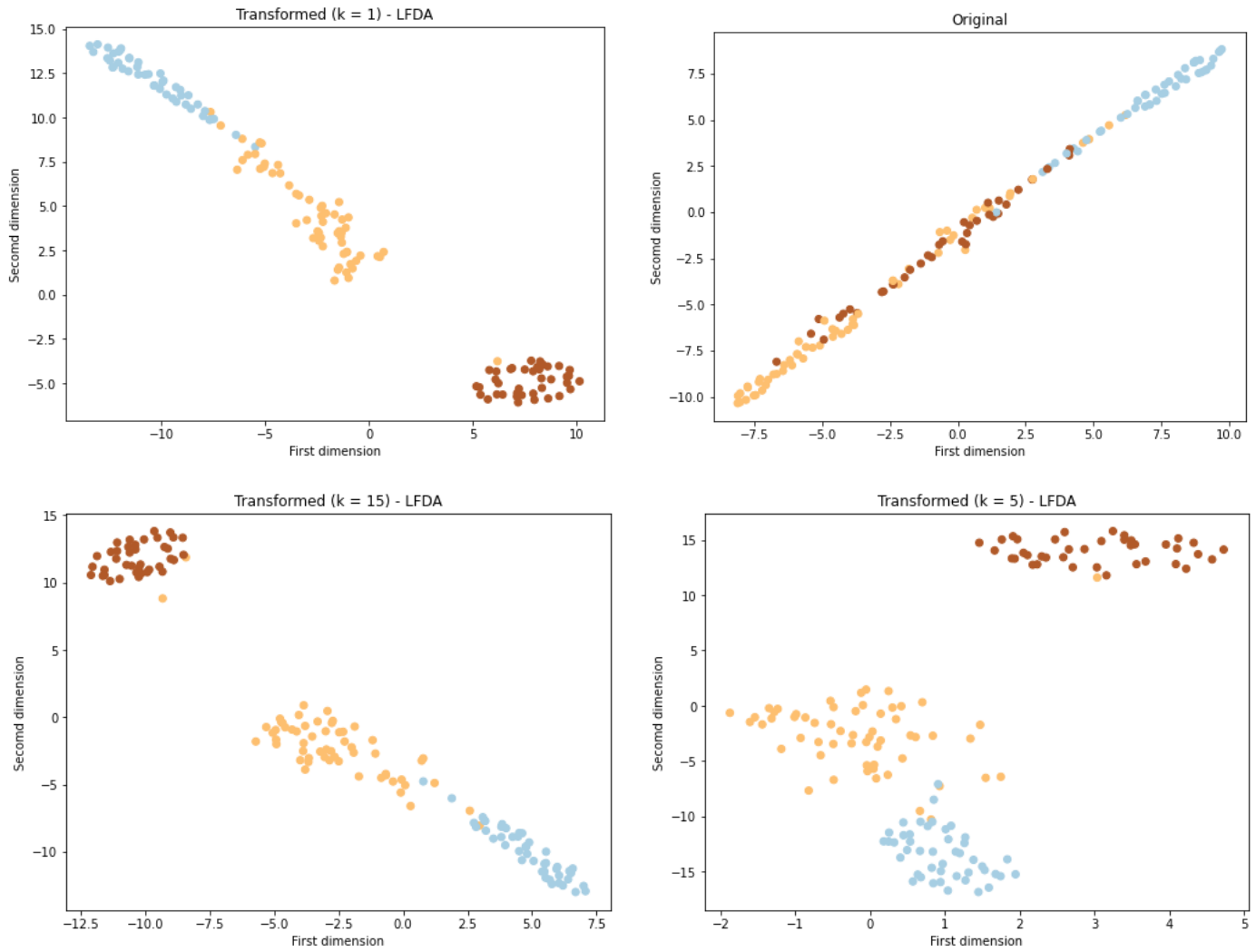
سوال ۲. ابتدا داده های اصلی (با ۱۳ بعد) را با استفاده از روش PCA، به فضای با تعداد بعد پایین تر (۲ بعد) انتقال می‌دهیم. سپس با استفاده از روش های یادگیری LMNN و LFDA، داده ها را به فضای جدید و با تعداد کمتر (به کمک آرگومان n_components در دستور مربوط به روش LMNN یا LFDA می‌بریم.

تصاویر ۳-۳، نمودار های دادگان در فضای با ۲ بعد با روش LMNN را نشان می‌دهد.



تصویر ۳-۳: نمودار دادگان در فضای تقلیل یافته با استفاده از روش LMNN

تصاویر ۳-۴، نمودار های دادگان در فضای با ۲ بعد با روش LFDA را نشان می‌دهد.



تصویر ۳-۴: نمودار دادگان در فضای تقلیل یافته با استفاده از روش LFDA

همانطور که در تصاویر ۳-۳ و ۳-۴ مشاهده می‌شود، بهترین مقدار k، برای هر دو روش LMNN و LFDA، مقدار $k = 15$ می‌باشد. زیرا در این حالت تعداد نقاط تصمیم گیری برای نزدیک شدن به کلاس دادهٔ منتخب یا دور شدن از آن زیادتر است و دسته ها جدا تر نسبت به هم قرار می‌گیرند.

قسمت ج: مقایسه عملکرد طبقه بند

حال داده های تست را با استفاده از ماشین آموزش داده شده با روش های LFDA و LMNN به فضای جدید انتقال می‌دهیم و از طبقه بند kNN در قسمت قبل برای مشخص کردن برچسب آن‌ها استفاده می‌کنیم.

تصویر ۳-۵، دقت و ماتریس آشفتگی را برای ماشین آموزش داده شده با استفاده از روش LMNN را نشان می‌دهد.

```
LMNN: (k = 1)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

```
LMNN: (k = 5)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

```
LMNN: (k = 10)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

```
LMNN: (k = 20)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

تصویر ۳-۵: دقت و ماتریس آشفتگی را برای ماشین آموزش داده شده با استفاده از روش LMNN

تصویر ۳-۶، دقت و ماتریس آشفتگی را برای ماشین آموزش داده شده با استفاده از روش LFDA را نشان می‌دهد.

```
LFDA: (k = 1)
 accuracy = 97.22222222222221 %
 cofusion matrix:
 [[12  1  0]
 [ 0 15  0]
 [ 0  0  8]]
```

```
LFDA: (k = 10)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

```
LFDA: (k = 5)
 accuracy = 97.22222222222221 %
 cofusion matrix:
 [[11  0  0]
 [ 1 16  0]
 [ 0  0  8]]
```

```
LFDA: (k = 20)
 accuracy = 100.0 %
 cofusion matrix:
 [[12  0  0]
 [ 0 16  0]
 [ 0  0  8]]
```

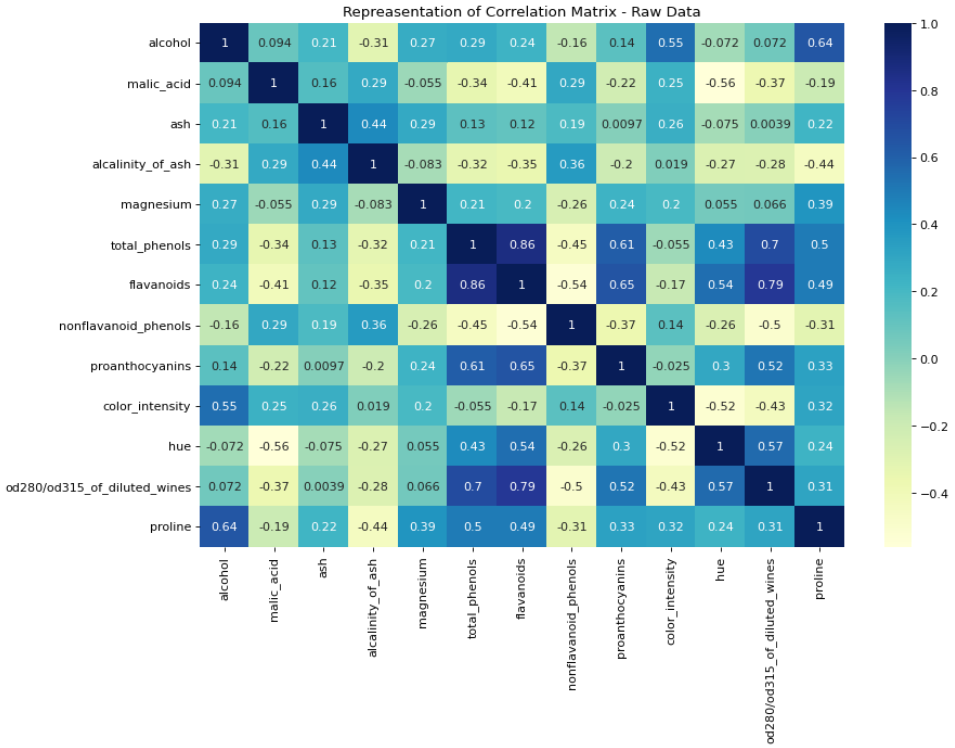
تصویر ۳-۶: دقت و ماتریس آشفتگی را برای ماشین آموزش داده شده با استفاده از روش LFDA

همانطور که در تصاویر ۳-۵ و ۳-۶، مشاهده می‌شود بعد از انتقال دادگان به فضای دو بعدی با استفاده از روش های یادگیری LFDA و LMNN داده ها از هم جدا شده و موقعیت بهتری نسبت به هم دارند. درنتیجه طبقه بند kNN عملکرد بهتری خواهد داشت و دقت آن بسیار بهتر خواهد شد.

قسمت د: ضریب همبستگی

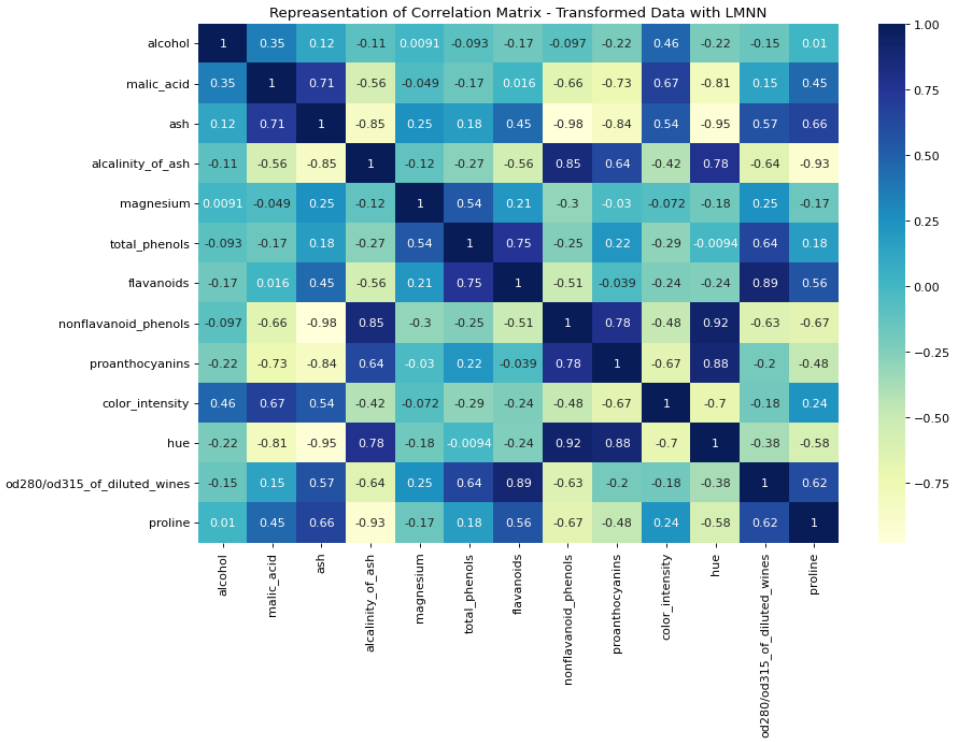
در این قسمت ابتدا برای داده های اولیه (خام) و انتقال یافته به هر دو روش یادگیری، ماتریس همبستگی را بدست می‌آوریم. به این منظور ابتدا داده ها و ویژگی های آن‌ها را به یک DataFrame تبدیل کرده و سپس از دستور corr(). برای بدست آوردن ماتریس هبستگی استفاده می‌کنیم. در نهایت با استفاده از دستور heatmap (کتابخانه seaborn) این ماتریس را رسم می‌کنیم.

تصویر ۳-۷ ماتریس همبستگی برای داده های خام را نشان می‌دهد.



تصویر ۳-۷: ماتریس هبستگی ویژگی ها برای داده های خام

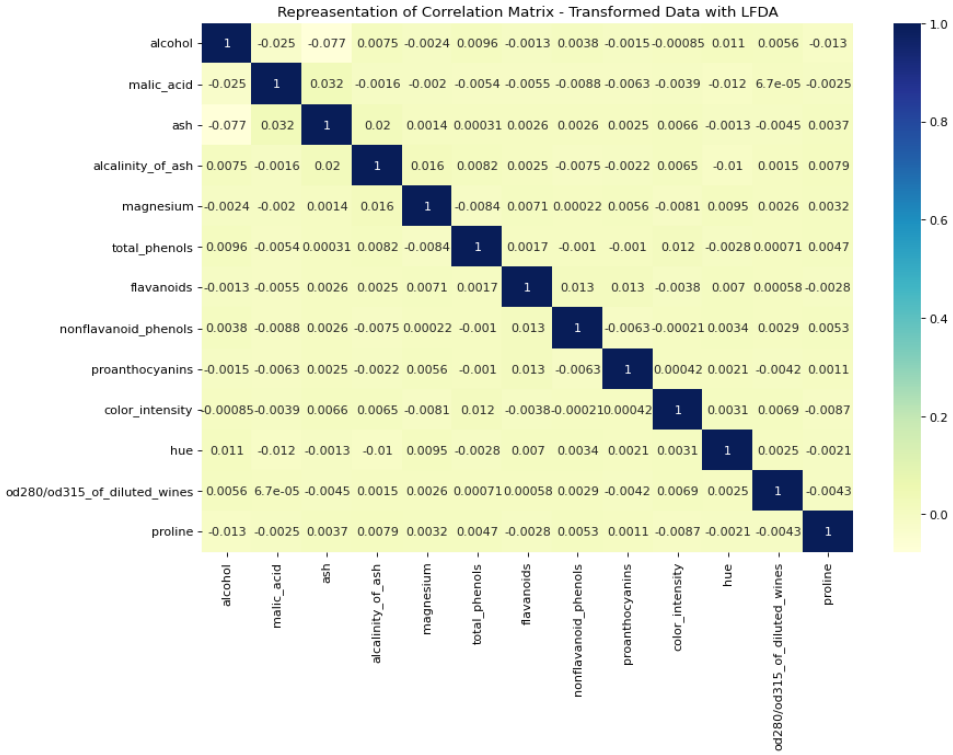
تصویر ۳-۸ ماتریس همبستگی برای داده های انتقال یافته به فضای جدید به روش LMNN را نشان می‌دهد.



تصویر ۳-۸: ماتریس هبستگی ویژگی ها برای داده های انتقال یافته به فضای جدید به روش LMNN

همانطور که در تصاویر ۳-۸ مشاهده می‌شود، مقادیر ماتریس همبستگی بعد از انتقال تصاویر به فضای جدید (۱۳ بعدی) با روش LMNN، به نسبت قبل بزرگتر شده است. در نتیجه می‌توان گفت با این روش یادگیری با افزایش همبستگی ویژگی ها به هم، جداسازی کلاس های مختلف آن ها را فراهم می‌کند.

تصویر ۳-۹ ماتریس همبستگی برای داده های انتقال یافته به فضای جدید به روش LFDA را نشان می‌دهد.



تصویر ۳-۹: ماتریس هبستگی ویژگی ها برای داده های انتقال یافته به فضای جدید به روش LFDA

همانطور که در تصاویر ۳-۹ مشاهده می‌شود، مقادیر ماتریس همبستگی بعد از انتقال تصاویر به فضای جدید (۱۳ بعدی) با روش LFDA، روی قطر اصلی بزرگتر است. در نتیجه می‌توان گفت با این روش یادگیری توانسته ایم تا حدودی ویژگی ها را از یکدیگر مستقل کنیم.

قسمت ه: GMML

در این روش سعی می‌شود با تغییر در تعریف مسئله بهینه سازی LMNN در قسمت الف، بتوانیم داده ها را بهتر در فضای جدید تصویر کنیم. به این منظور مسئله بهینه سازی به صورت زیر تعریف می‌شود. (S بیانگر داده های هم‌کلاس و D بیانگر داده های کلاس مخالف است.)

$$A = \underset{A}{\operatorname{argmin}} \sum_{(x_i,x_j)\in S} tr\left[A(x_i - x_j)(x_i - x_j)^T\right] + \sum_{(x_i,x_j)\in N} tr\left[A^{-1}(x_i - x_j)(x_i - x_j)^T\right]$$