دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

گزارش تمرین شماره 2

درس سیستم های هوشمند

پاییز 1401

امیرحسین بیرژندی

...

810198367

...

سوال 1 – درخت تصمیم (تحلیلی)

الف) طراحی طبقه‌بند

برای طراحی درخت تصمیم ابتدا برای هرکدام از ویژگی ها Information Gain را محاسبه کرده و ماکسیمم را به عنوان نود اصلی در نظر می‌گیریم.

برای محاسبه Information Gain از رابطه زیر استفاده می‌کنیم.

$$Information\ Gain(S,A) = Entropy(S) - \frac{|S_{yes}|}{|S|}Entropy(S_{yes}) - \frac{|S_{no}|}{|S|}Entropy(S_{no})$$

$$IG(S,Color) = 1 - \frac{6}{10} * 0.9183 - \frac{4}{10}0.8113 = 0.1245$$

$$IG(S,number\ of\ legs) = 1 - \frac{7}{10} * 0.8631 = 0.3958$$

$$IG(S,height) = 1 - 0.4 - 0.6 = 0$$

$$IG(S,Habitat) = 1 - 0.4 - 0.6 = 0$$

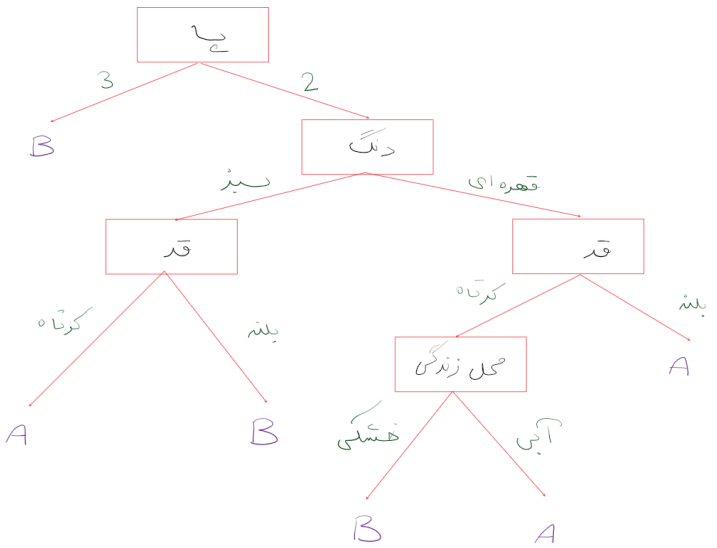
در نتیجه نود اول را تعداد پا ها قرار می‌دهیم حال دوباره Information Gain را برای دیگر ویژگی ها محاسبه می‌کنیم در حالتی که تعداد پاها برابر 2 است. زیرا در حالتی که تعداد پاها برابر 3 است همه جانداران از نوع B هستند.

$$IG(S_2,Color) = 0.8631 - \frac{5}{7} * 0.7219 - \frac{2}{7} = 0.0617$$

$$IG(S_2,height) = 0.8631 - \frac{4}{7}0.8113 - \frac{3}{7}0.9183 = 0.0059$$

$$IG(S_2,Habitat) = 0.8631 - \frac{4}{7}0.8113 - \frac{3}{7}0.9183 = 0.0059$$

نتیجه می‌گیریم در سطح دوم از ویژگی رنگ استفاده می‌کنیم. و این مرحله را دوباره تکرار می‌کنیم. دقت شود در اینجا در هر دو سمت شاخه باید این مسیر را تکرار کنیم ابتدا در سمت قهوه‌ای این کار را را انجام می‌دهیم. در نهایت نمودار درخت تصمیم به صورت زیر خواهد بود.



ب) آزمون طبقه‌بند

با تست کردن عملکرد درخت طراحی شده متوجه می‌شویم که شماره 2 و4 به درستی عمل نمی‌کند در نتیجه ماتریس آشفتگی به صورت زیر خواهد بود.

	A	B
A	**2**	**1**
B	**1**	**2**

Confusion Matrix

ج) رویکرد حریصانه الگوریتم *ID3*

اگر بخواهیم این طبقه بندی را به کمک تنها دو ویژگی انجام دهیم موفق نخواهیم بود. زیرا هیچ دو ویژگی وجود ندارند که جانداران نوع A و B را به طور کامل از همدیگر جدا می‌کنند. برای بررسی این موضوع یکی از جفت ویژگی ها را بررسی می‌کنیم. برای مصال اگر تعداد پا و قد را بررسی کنیم مشاهده می‌کنیم پس از جداسازی به وسیله تعداد پا آن هایی که تعداد پاهای برابر با 2 دارند اگر به وسیله قد آن ها را جدا کنیم در سمت جانداران بلند هم جانداران نوع A وجود خواهند داشت و هم نوع B در نتیجه درصد خطا روی داده های آموزش برابر صفر نخواهد بود. با بررسی بقیه جفت ویژگی ها این موضوع را خواهیم دید.

د) افزایش قوام طبقه بند

در درخت های تصمیم گیری به دلیل اینکه این روش بر روی داده های آموزش به خوبی فیت می‌شوند یعنی به نوعی به نویز های داده ها حساس می‌شود در نتیجه فرابرازش رخ می‌دهد و این باعث می‌شود روی داده های ندیده عملکرد خیلی مطلوبی نداشته باشد.

برای بهبود درخت تصمیم می‌توانیم از *pre-pruning* استفاده کنیم در این روش به جای اینکه شاخه را به طور کامل گسترش دهیم عمق درخت را محدود کرده تا خیلی به تک تک داده ها حساس نشود و برگ های خیلی کوچک تشکیل نشود.

روش دیگر برای بهبود درخت تصمیم و جلوگیری از بیش‌برازش آن *post-pruning* که پس از تشکیل یک درخت کامل بعضی از برگ های آن را جدا کرده

سوال 2 – پیاده‌سازی الگوریتم درخت تصمیم

الف) پیاده سازی مدل درخت

ابتدا بررسی می‌کنیم چه ویژگی هایی می‌تواند تاثیر بسازیی داشته باشد.

سن : مشخصاً سن افراد می‌تواند موثر باشد زیرا افراد مسن شاید جنب و جوش افراد جوان را نداشته باشند.

جنسیت : در کشتی تایتانیک ابتدا خانم‌ها سوار کشتی های کمکی شدند و در نتیجه سریع‌تر امکان نجات داشتند.

طبقه : افراد ساکن در طبقه پایین امکان رسیدن به عرشه کشتی در مدت زمان کمتر دارند.

معیار انتخاب ویژگی را بهره اطلاعات در نظر می‌گیریم.

توضیح پیاده سازی:

برای پیاده‌سازی این بخش دو کلاس که یکی نود است و دیگری خود درخت است. توابع مورد نیاز به چند بخش تبدیل می‌شوند که یا توابعی هستند که بررسی می‌کنند که چه ویژگی بهره اطلاعات بیشتری دارد پس از انتخاب آن با تابع جدا کننده با توجه به مرز تصمیم گیری داده ها را تقسیم می‌کنیم و این کار را تا آنجایی ادامه می‌دهیم که به یک برگ برسیم. در زیر منظور از K همان عمق درخت است.

K = 3

```
The accuracy for k=3 is:  0.8461538461538461

The confusion matrix:

array([[51,  6],
       [ 8, 26]], dtype=int64)
```

K = 4

```
The accuracy for k=4 :  0.8571428571428571

The confusion matrix:

array([[50,  7],
       [ 6, 28]], dtype=int64)
```

K = 5

```
The accuracy for k=5 :  0.8571428571428571

The confusion matrix:

array([[50,  7],
       [ 6, 28]], dtype=int64)
```

K = 6

```
The accuracy for k=6 :  0.8351648351648352

The confusion matrix:

array([[50,  7],
       [ 8, 26]], dtype=int64)
```

K = 7

```
The accuracy for k=7 :  0.8461538461538461

The confusion matrix:

array([[50,  7],
       [ 7, 27]], dtype=int64)
```

با افزایش تعداد درخت از 3 به 4 و 5 دقت درخت افزایش کمی پیدا می‌کند زیرا با عمق 3 هنوز دسته بندی خیلی دقیق صورت نگرفته است در نتیجه عمق 4 و 5 نتیجه بهتری دارد. اما با افزایش دوباره به دلیل بیش برازش روی داده های آموزش نتیجه کمی افت می‌کند.

ب) بهبود بخشی الگوریتم درخت تصمیم

ایرادات:

1- با تغییر کوچکی در داده‌ها امکان تغییر خیلی زیاد در ساختار درخت وجود دارد.

2- زمان و هزینه یادگیری بالایی نسبت به روش های دیگر دارد.

3- برای مقادیر پیوسته نسبت به دیگر الگوریتم ها نتایج ضعیف‌تری دارد.

4- امکان بیش برازش در آن زیاد است.

با استفاده از جنگل های تصادفی و یا *Bagging* به دلیل اینکه در هر درخت از نمونه های خاصی از کل دیتاست استفاده می‌شود و همچنین اجماع چند درخت و در نهایت رای اکثریت بین آن ها از واریانس یک درخت تصمیم ساده می کاهیم. به عبارتی همانطور که گفته شد در درخت تصمیم با تغییر کمی از داده ها ممکن است کل مدل تغییر پیدا کند در صورتی که با جنگل های تصادفی و *bagging* این اتفاق رخ نمی‌دهد.

ج) استفاده از جنگل تصادفی

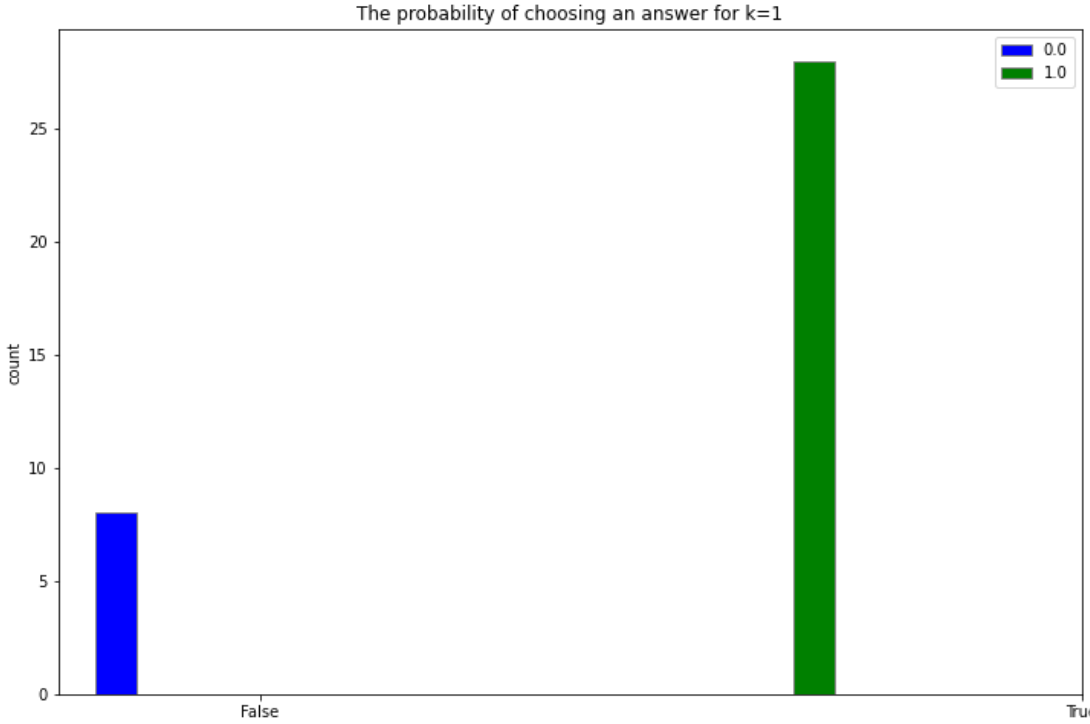
```
The accuracy for 10 trees with max depth of 5 is:  0.8681318681318682

                    The confusion matrix:

                    array([[54,  3],
                           [ 9, 25]], dtype=int64)
```

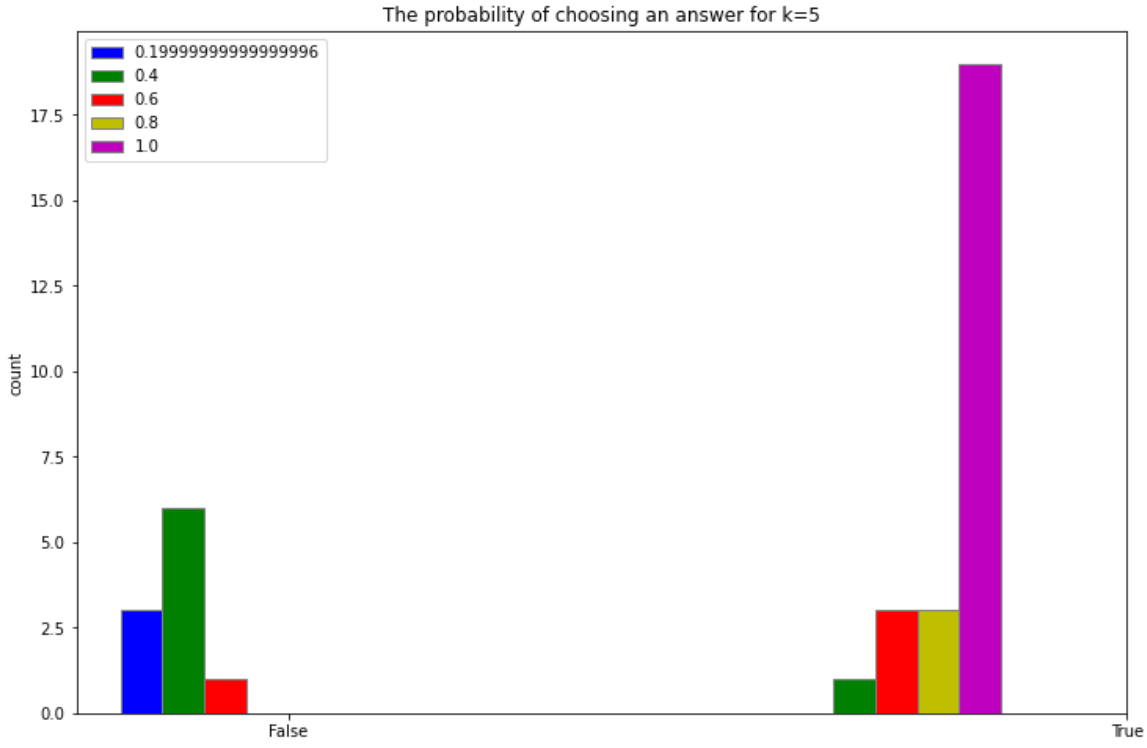
سوال 3 – یادگیری بر اساس معیار

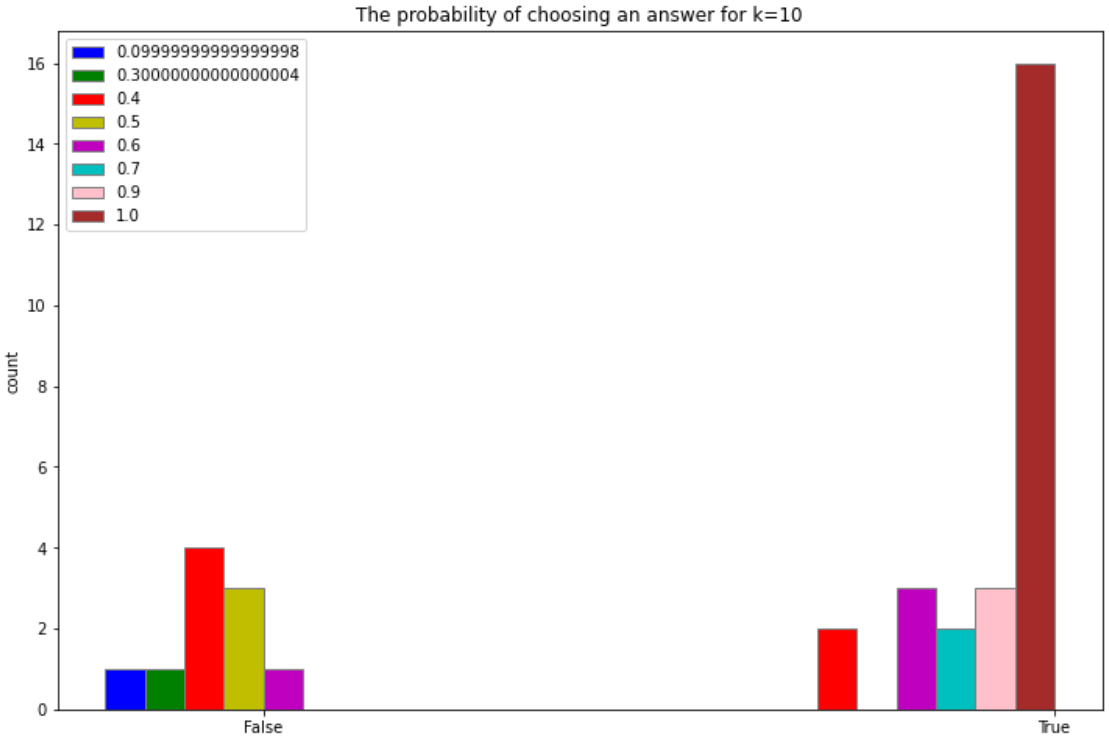
الف) طراحی طبقه بند

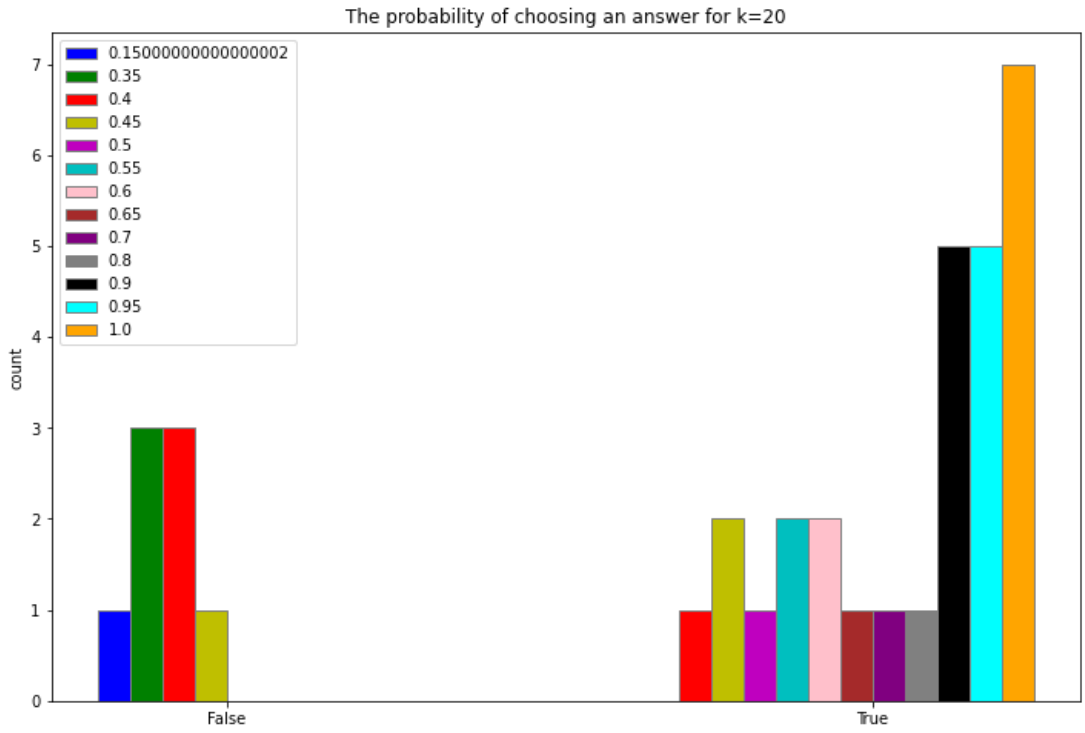
```
The accuracy score for k=1 is:  0.7777777777777778

The confusion matrix for k=1 is:

array([[12.,  0.,  2.],
       [ 3., 11.,  0.],
       [ 1.,  2.,  5.]])


The accuracy score for k=5 is:  0.7222222222222222

The confusion matrix for k=5 is:

array([[12.,  0.,  2.],
       [ 0., 11.,  3.],
       [ 2.,  3.,  3.]])


The accuracy score for k=10 is:  0.7222222222222222

The confusion matrix for k=10 is:

array([[14.,  0.,  0.],
       [ 0.,  9.,  5.],
       [ 3.,  2.,  3.]])


The accuracy score for k=20 is:  0.7777777777777778

The confusion matrix for k=20 is:

array([[14.,  0.,  0.],
       [ 0.,  9.,  5.],
       [ 1.,  2.,  5.]])
```

ب) محاسبه توزیع احتمال تعلق به هر کلاس






The probability of choosing an answer for k=10
0.09999999999999998
0.30000000000000004
0.4
0.5
0.6
0.7
0.9
1.0
count
0
2
4
6
8
10
12
14
16
False
True


The probability of choosing an answer for k=20
0.15000000000000002
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.8
0.9
0.95
1.0
count
0
1
2
3
4
5
6
7
False
True

یادگیری بر اساس معیار

الف) بررسی کارکرد روش یادگیری:

در روش *LMNN* مسئله بهینه سازی به صورت زیر است.

$$\min_{\mathbf{M}} \sum_{i,j \in N_i} d(\vec{x}_i, \vec{x}_j) + \lambda \sum_{i,j,l} \xi_{ijl}$$

که در آن سعی می‌کنیم داده های هم برچسب را تا حد امکان به یکدیگر نزدیک و داده های با برچسب متفاوت را از هم دور کنیم. شروط آن در زیر به اختصار گفته می‌شود.

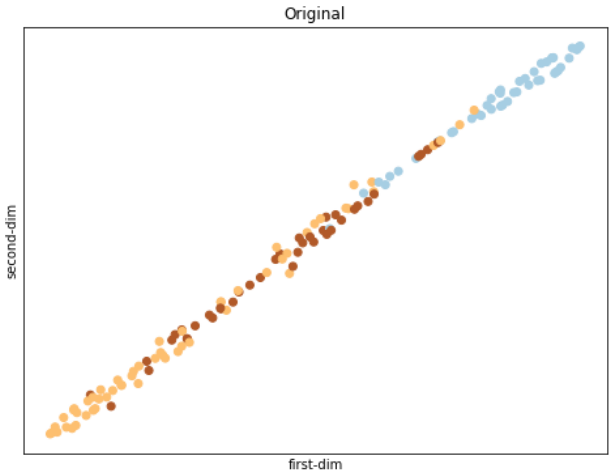
$$\begin{aligned}
&\forall_{i,j \in N_i, l, y_l \neq y_i} \\
&d(\vec{x}_i, \vec{x}_j) + 1 - d(\vec{x}_i, \vec{x}_l) \leq \xi_{ijl} \\
&\xi_{ijl} \geq 0 \\
&\mathbf{M} \succeq 0
\end{aligned}$$

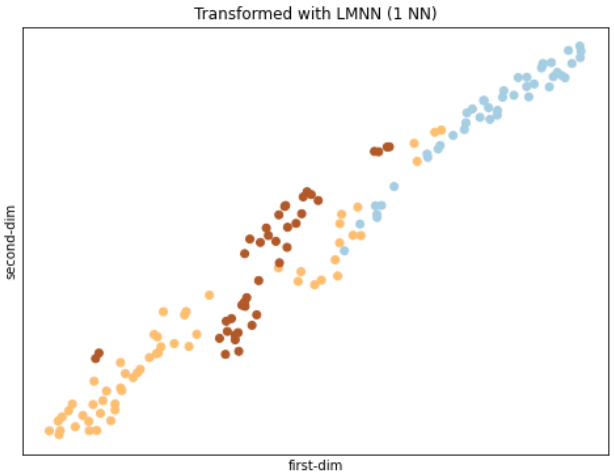
شروط آن برای این است که به ازای نزدیک شدن به داده غیر هم برچسب جریمه شویم.

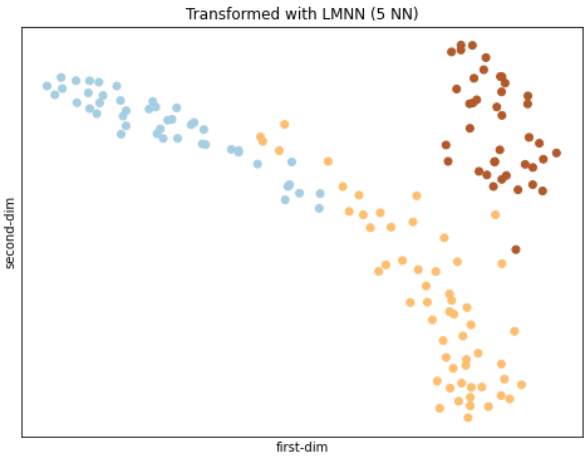
در *LFDA* هدف این است که زمانی که داده ها در یک صفحه چندین برچسب دارند را بتوانیم آن ها را به نحوی از یکدیگر جدا کنیم. که داده های هم برچسب نزدیک یکدیگر و از آن سو داده های یر هم برچسب در سوی دیگر قرار بگیرند.

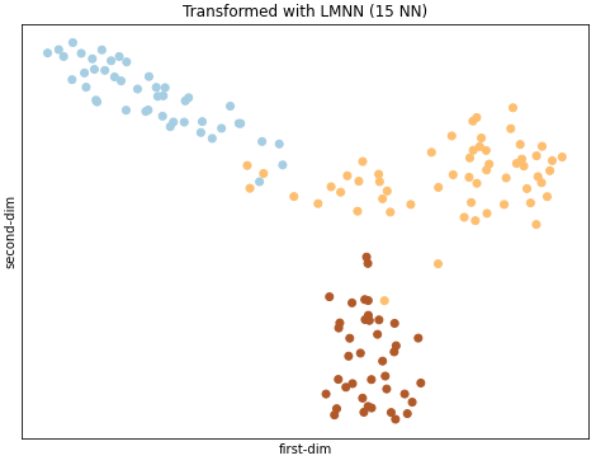
ب) ترسیم دادگان انتقال یافته به فضای جدید:

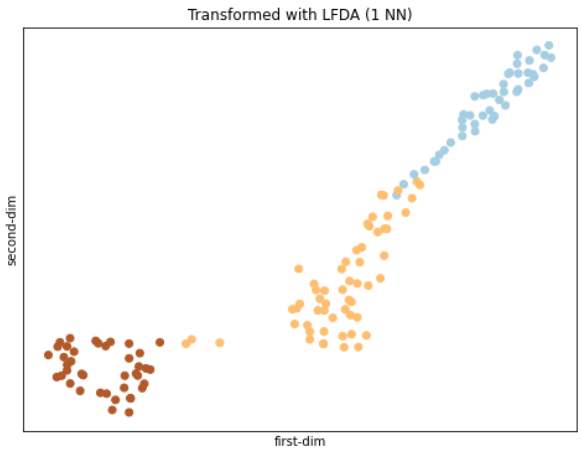
همانطور که گفته شد در این دو روش سعی بر این است که داده های هم برچسب را به یکدیگر نزدیک تر و داده های با برچسب متفاوت را از یکدیگر دور کنیم. پارامتر K نیز در واقع بدین منظور است که با آن تعداد داده همسایه ای که با آن داده های هم لیبل را جدا کنیم مشخص می‌کنیم و برای همین است که زمانی که $k=15$ بود نسبت به دو مقدار دیگر همسایه های هم لیبل بیشتری با یکدیگر جدا می‌کنیم در نتیجه در نهایت پاسخ بهتری نیز دریافت می‌کنیم.

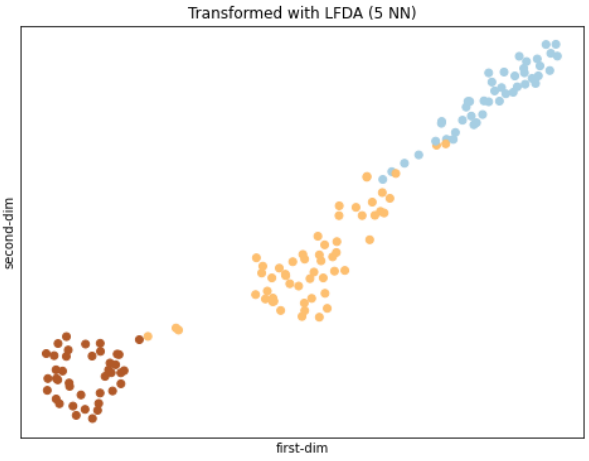

Original
second-dim
first-dim

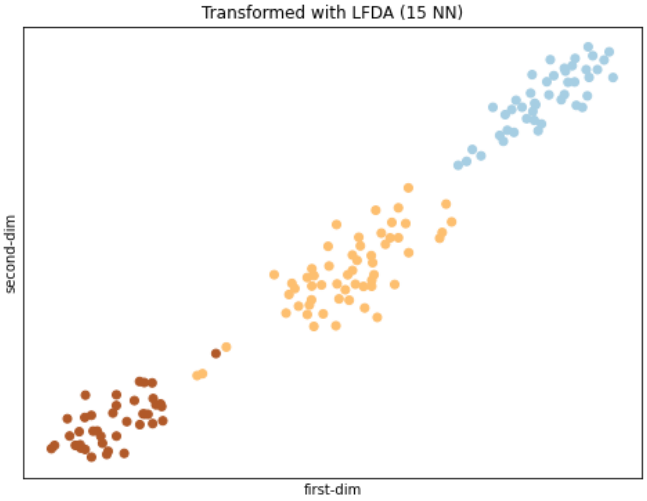

Transformed with LMNN (1 NN)
second-dim
first-dim


Transformed with LMNN (5 NN)
second-dim
first-dim


Transformed with LMNN (15 NN)
second-dim
first-dim


Transformed with LFDA (1 NN)
second-dim
first-dim

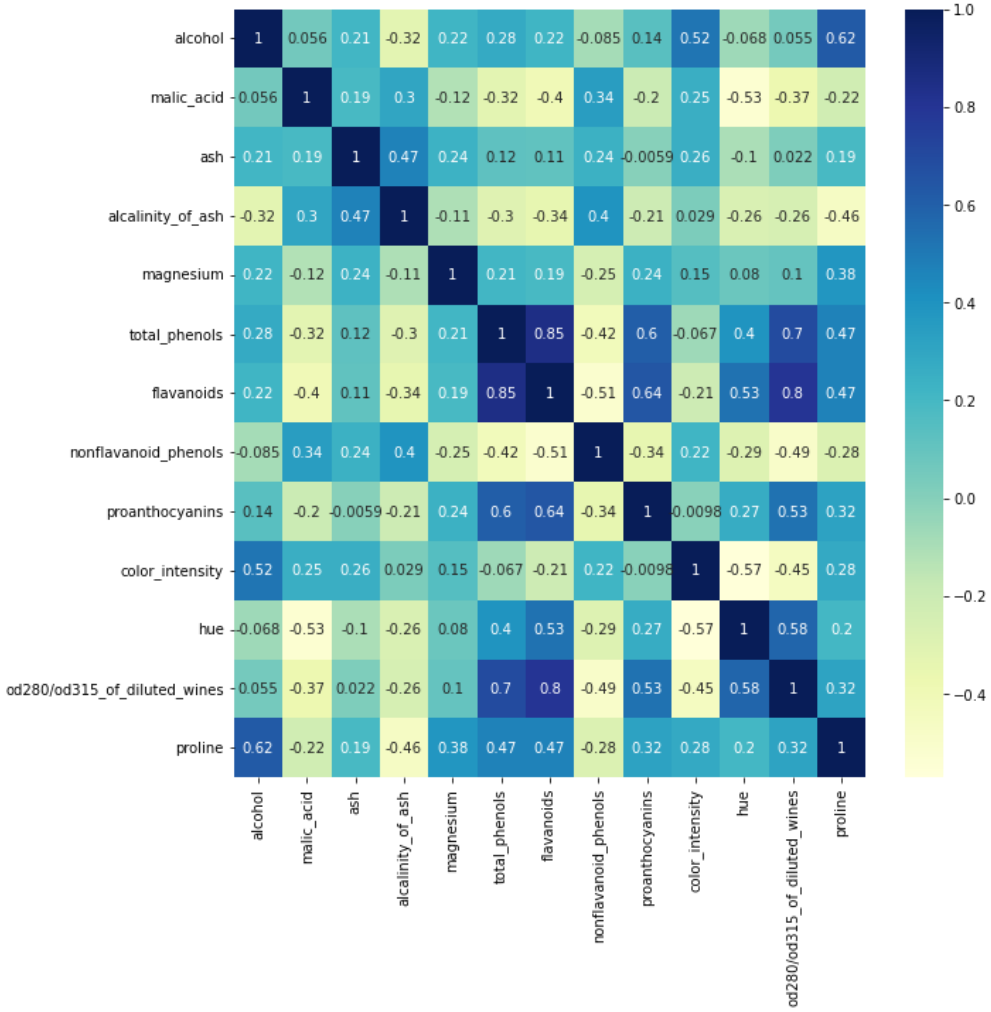

Transformed with LFDA (5 NN)
second-dim
first-dim



ج) مقایسه عملکرد طبقه بند

```
The accuracy score for k=15 in lmnn is:  1.0
The confusion matrix for k=15 in lmnn is:
 [[14.  0.  0.]
 [ 0. 14.  0.]
 [ 0.  0.  8.]]


The accuracy score for k=15 in lfda is:  1.0
The confusion matrix for k=15 in lfda is:
 [[14.  0.  0.]
 [ 0. 14.  0.]
 [ 0.  0.  8.]]
```

د) ضریب همبستگی



ه) GMML

در این مقاله هدف این است که همانند مقاله های دیگر با یک معیار شباهت و یک معیار تفاوت در نظر بگیریم که در صورت مشروط شدن تابع هدف روی آن ها بتوانیم داده ها را از هم جدا کنیم.

$$\min_{\boldsymbol{A}\succeq 0} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{S}} \max\big(0\,,\; l - d_{\boldsymbol{A}}(\boldsymbol{x}_i,\boldsymbol{x}_j)\big)^2 + \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{D}} \max\big(0\,,\; d_{\boldsymbol{A}}(\boldsymbol{x}_i,\boldsymbol{x}_j) - u\big)^2,$$

تابع هدف این روش با روش LMNN تفاوت دارد