دانشگاه سمنان
Semnan University
پردیس فرزانگان

بسمه تعالی

فصل چهارم داده کاوی

دسته بندی
Classification

مدرس

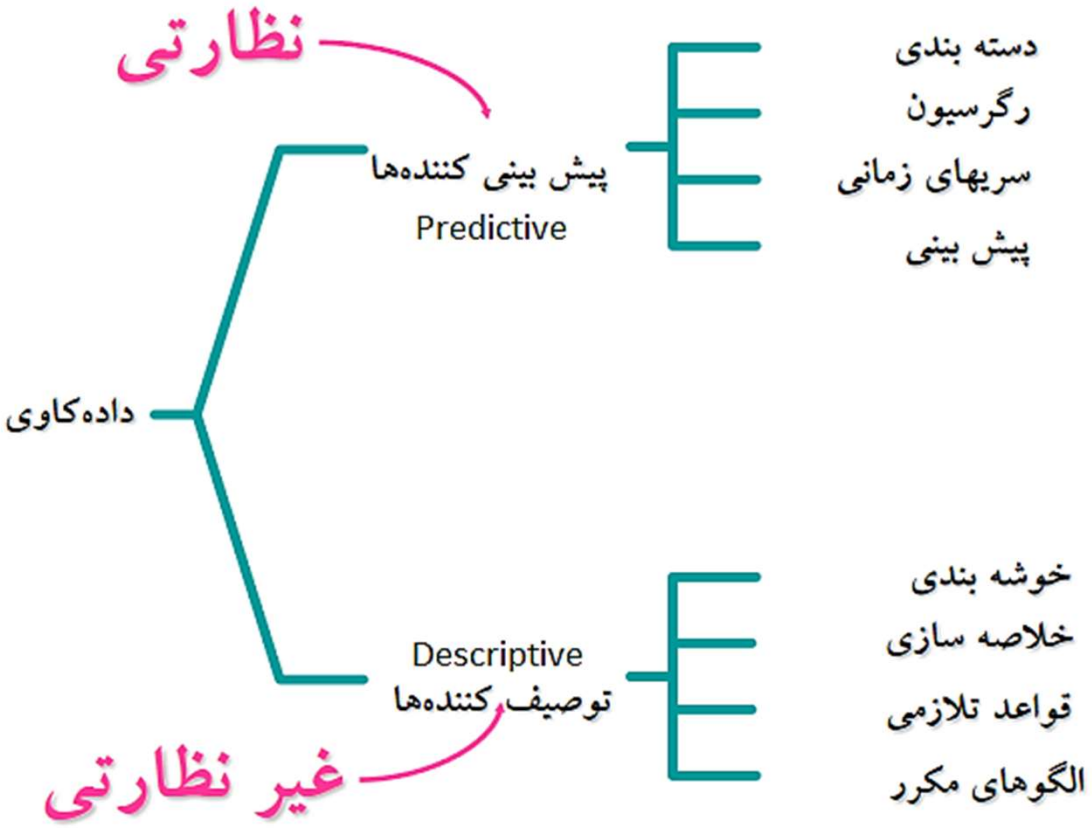
فاطمه دارائی

f_daraei@semnan.ac.ir

https://fdaraei.profile.semnan.ac.ir

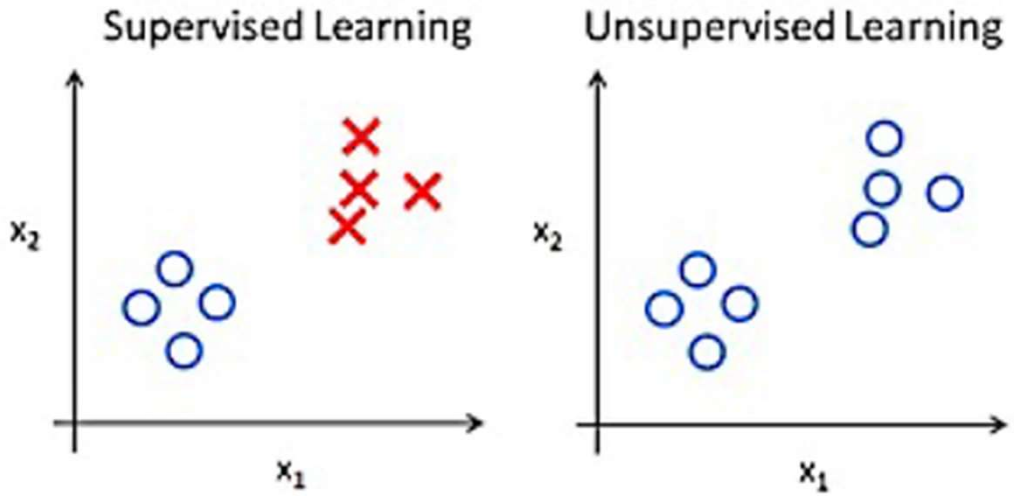
دانشگاه سمنان
Semnan University
پردیس فرزانگان

تکنیک های داده کاوی



یادگیری با نظارت Supervised learning

- این نوع یادگیری، نگاشت داده‌های ورودی به اهداف شناخته شده با توجه به مجموعه‌ای از نمونه‌ها (اغلب توسط انسان‌ها برچسب گذاری شده) است.
- داده های آموزشی همراه با برچسب هستند که کلاس آن داده ها را نشان می‌دهند.
- بیشترین کاربرد و بالاترین دقت در این نوع یادگیری است.



دانشگاه سمنان
Semnan University
پردیس فرزانگان

دسته بندی Classification

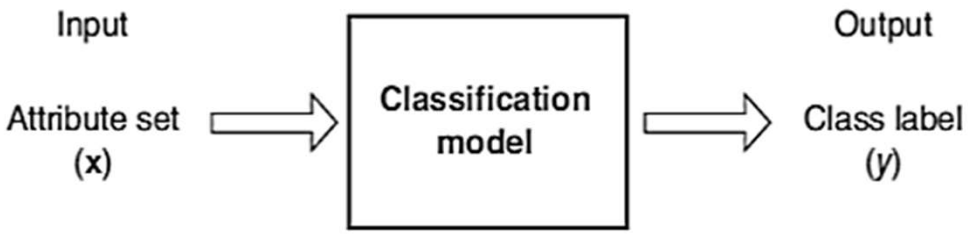
دسته بندی وظیفه یادگیری تابع هدف f است که مجموعه x را به یکی از برچسب های کلاس از پیش تعریف شده y نگاشت می کند.
مجموعه آموزشی شامل رکوردهایی با برچسب های کلاس شناخته شده است و هر رکورد شامل مجموعه‌ای از ویژگی‌ها است که یکی از آن‌ها کلاس است.

وظیفه: یافتن مدلی برای ویژگی کلاس به عنوان تابعی از مقادیر سایر ویژگی‌ها.

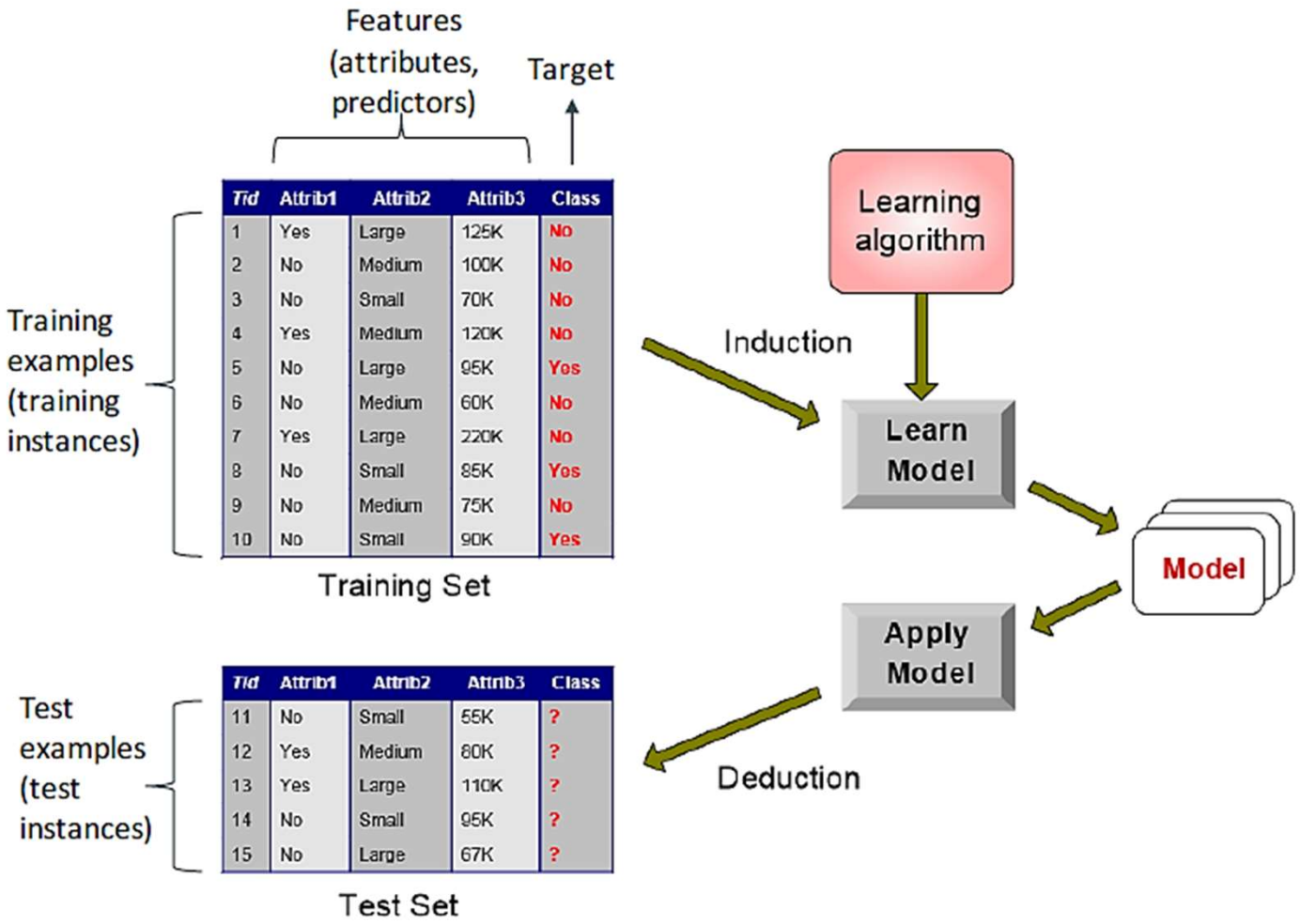
هدف: رکوردهای قبلاً دیده نشده باید با دقت بالا به یک کلاس اختصاص داده شوند.

یک مجموعه تست برای تعیین دقت مدل استفاده می‌شود. معمولاً، مجموعه داده‌های داده شده به مجموعه‌های آموزشی و تست تقسیم می‌شود،

با این تفاوت که مجموعه آموزشی برای ساخت مدل و مجموعه تست برای اعتبارسنجی آن استفاده می‌شود.



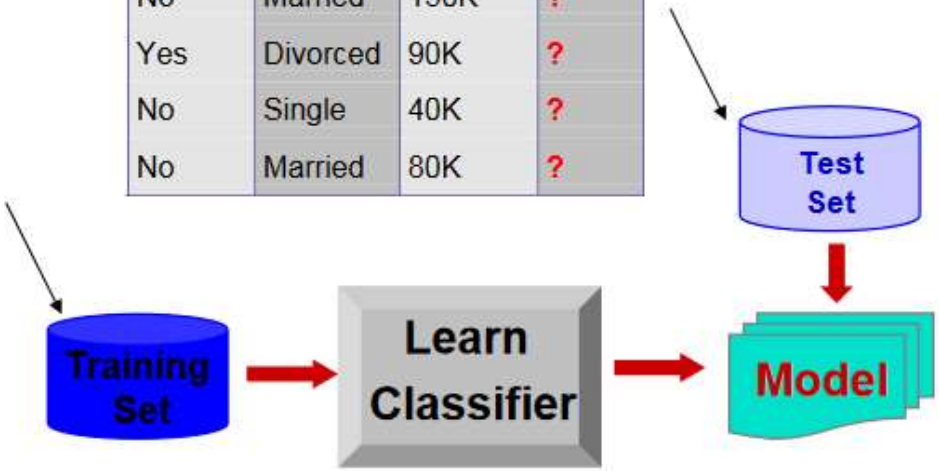
مثال دسته بندی Classification



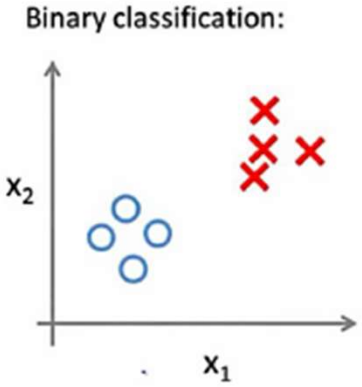
Two class labels (or classes): Yes (1), No (0)

Tid	Refund (categorical)	Marital Status (categorical)	Taxable Income (continuous)	Cheat (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?

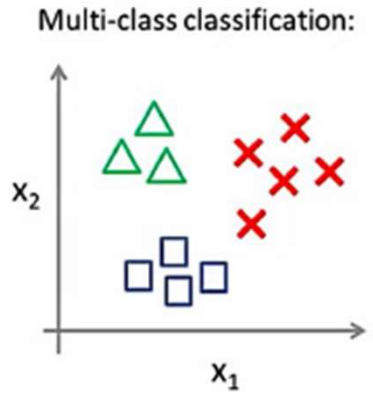


دسته بندی باینری و دسته بندی چند کلاسه



Binary classification:

وظایف طبقه‌بندی با تنها دو کلاس، که معمولاً با {+،−}، {+۱،−۱} یا {مثبت، منفی} نشان داده می‌شوند.
مثال: تشخیص مثبت بودن بیماری، تحلیل احساسات (مثبت/منفی).



Multiclass classification:

وظایف طبقه‌بندی با بیش از دو کلاس.
مثال: شناسایی موضوع ایمیل، تحلیل احساسات (مثبت/خنثی/منفی).

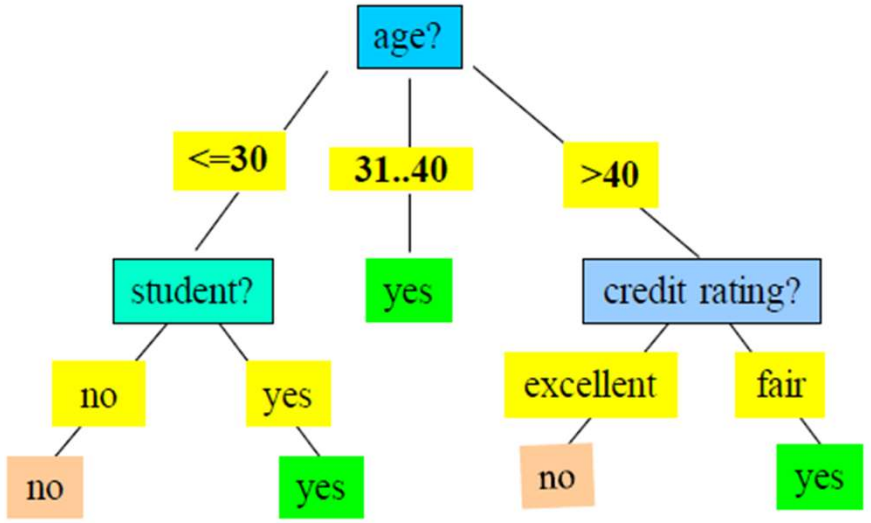
دانشگاه سمنان
Semnan University
پردیس فرزانگان

Decision Tree درخت تصمیم

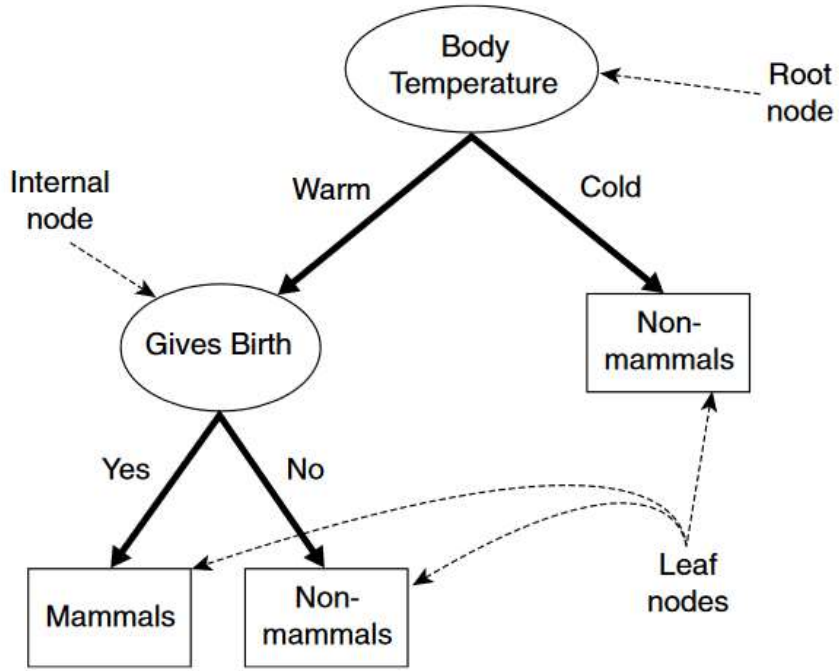
درخت تصمیم، یکی از ابزارهای قوی و متداول برای دسته بندی و پیش بینی می‌باشد.

درخت تصمیم یک درخت با ساختار فلوچارت مانند است.

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



- هر گره غیرپایانی، یک تست (آزمون و سوال) بر روی ویژگی انجام می‌دهد.
- در هر یک از گره های داخلی، با توجه به یک یا چند صفات خاصه تصمیم‌گیری صورت می‌گیرد.
- هر شاخه یا یال یک نتیجه از تست است.
- هر گره برگ، یک برچسب کلاس است.
- بالاترین گره در درخت، گره ریشه است.



نحوه استفاده از درخت تصمیم

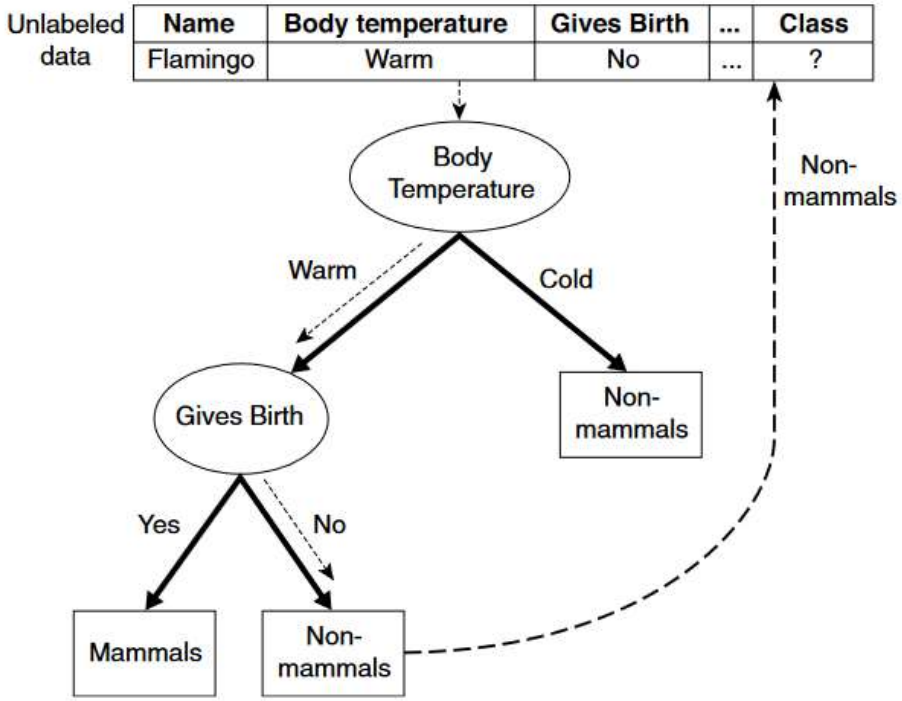
- برای یک داده جدید که مقادیر ویژگی آن معلوم است اما برچسبی ندارد.
- بر اساس درخت یک مسیر از ریشه به برگ طی می شود.
- برچسب برگ نشان دهنده کلاس پیش‌بینی شده برای آن موجودیت است.
- با این تفاسیر درخت تصمیم را به راحتی می توان به قوانین دسته بندی تبدیل نمود.

در درخت تصمیم یکسری سوال وجود دارد و با مشخص شدن

پاسخ هر سوال، یک سوال دیگر پرسیده می شود.

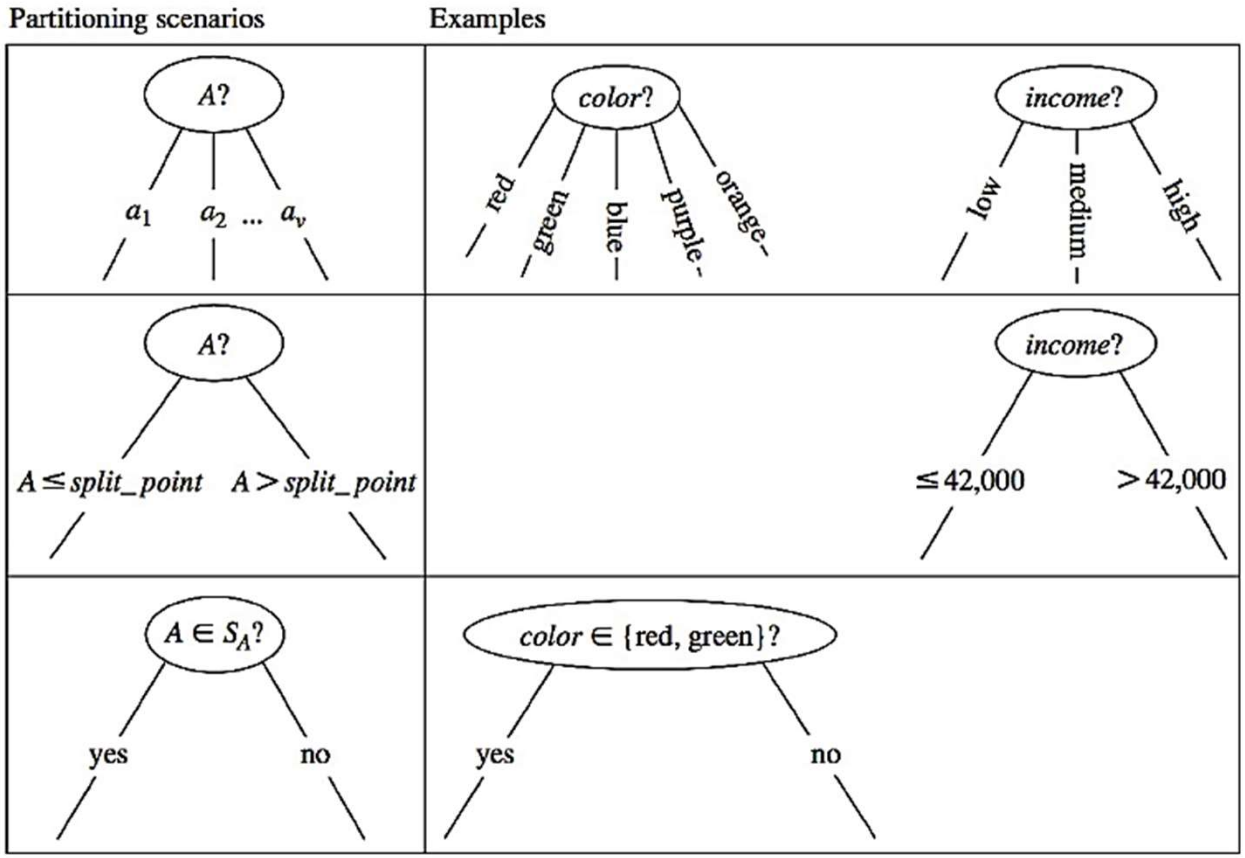
اگر سوالها درست و خوب پرسیده شوند،

با یک مسیر کوتاه پیش بینی دسته رکورد جدید انجام می شود.



دانشگاه سمنان
Semnan University
پردیس فرزانگان

انواع تقسیم بندی گره میانی

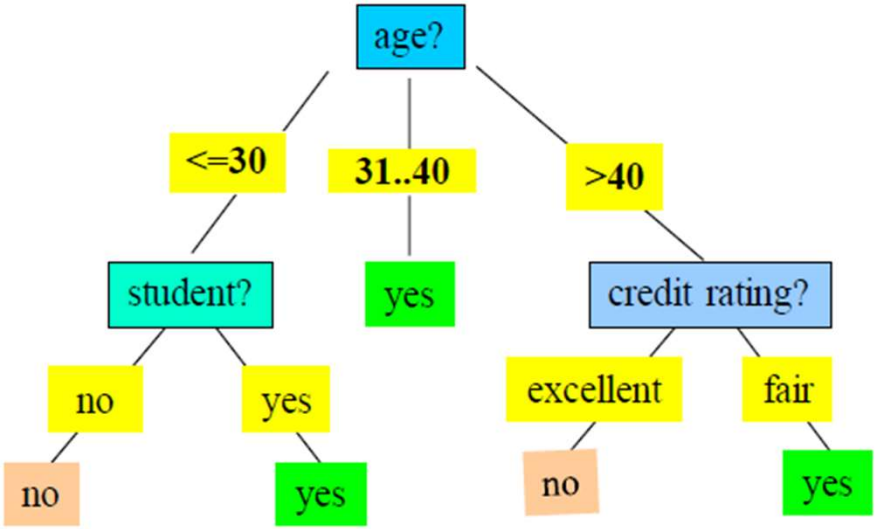


نحوه استفاده از درخت تصمیم

چگونه درخت‌های تصمیم برای طبقه‌بندی استفاده می‌شوند؟

1. مسیری از ریشه تا یک گره برگ طی می‌شود که پیش‌بینی مربوط به آن صفت را در خود دارد.
2. ویژگی‌های یک صفت با درخت تصمیم مقایسه می‌شوند.

RID	age	income	student	credit-rating	Class
1	youth	high	no	fair	?



چرا درخت تصمیم؟

۱) ایجاد درخت تصمیم به هیچ دانش تخصصی یا تنظیم پارامتری احتیاج ندارد، به همین دلیل برای کشف دانش مناسب است.

۲) درخت‌های تصمیم‌گیری می تواند داده های با ابعاد زیاد را پوشش دهند

۳) نمایش دانش بدست آمده از ساختار درختی شهودی و به طور کلی آسان است و توسط انسان قابل درک است.

۴) آموزش و دسته بندی با استفاده از درخت تصمیم‌گیری ساده و سریع می باشد.

۵) به طور کلی، دسته بندی با درخت تصمیم‌گیری از دقت خوبی برخوردار است.

۶) الگوریتم درخت تصمیم‌گیری برای دسته بندی در بسیاری از زمینه های کاربردی قابل استفاده است.

الگوریتم درخت تصمیم: یادگیری

الگوریتم های مختلفی برای ساخت درخت تصمیم وجود دارد:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification And Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector)
- MARS (extends decision trees to handle numerical data better)

- در این الگوریتم‌ها یک روش حریصانه، بدون بازگشت به عقب، استفاده می شود.
- همچنین راهکار به صورت بالا به پایین و تقسیم و حل است.

الگوریتم حریصانه (الگوریتم پایه)

- درخت به صورت بازگشتی با روش تقسیم و حل، از بالا به پایین ساخته می‌شود.
- در آغاز، همه نمونه های آموزشی در ریشه هستند.

ویژگی‌ها بر اساس یک معیار آماری انتخاب می‌شوند.

- ویژگی‌ها چند مقداری هستند (اگر مقادیر پیوسته باشد، گسسته می‌شوند).
- داده ها به صورت بازگشتی و بر اساس ویژگی‌های انتخاب شده پارتیشن بندی می‌شوند.
- ویژگی‌های آزمون بر اساس معیارهای اکتشافی و یا آماری انتخاب می‌شوند.

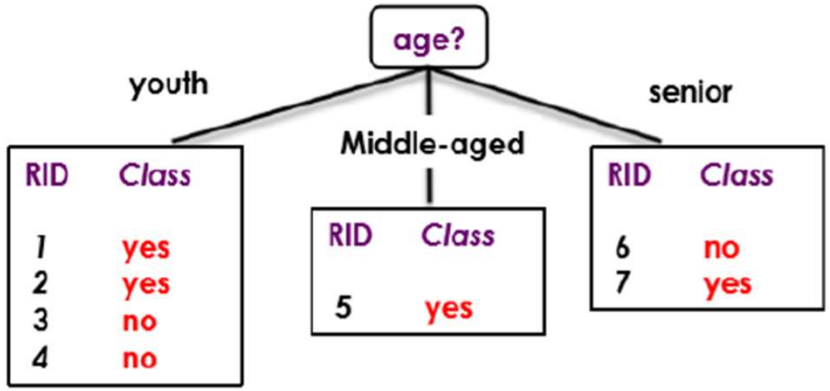
شرایط برای توقف پارتیشن‌بندی

- همه نمونه ها برای یک گره داده شده متعلق به یک کلاس باشند.
- هیچ ویژگی‌ای برای پارتیشن‌بندی بیشتر، باقی نمانده باشد- در این شرایط برچسب اکثریت نمونه‌های برگ برای دسته‌بندی به کار گرفته می‌شود. (ناشی از ویژگی‌های ناکافی یا وجود نویز)
- هیچ نمونه بیشتری وجود نداشته باشد.

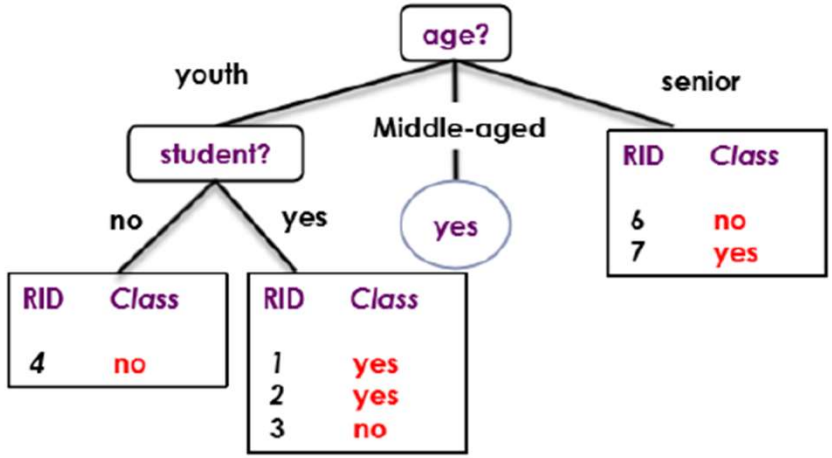
دانشگاه سمنان
Semnan University
پردیس فرزانگان

مثال

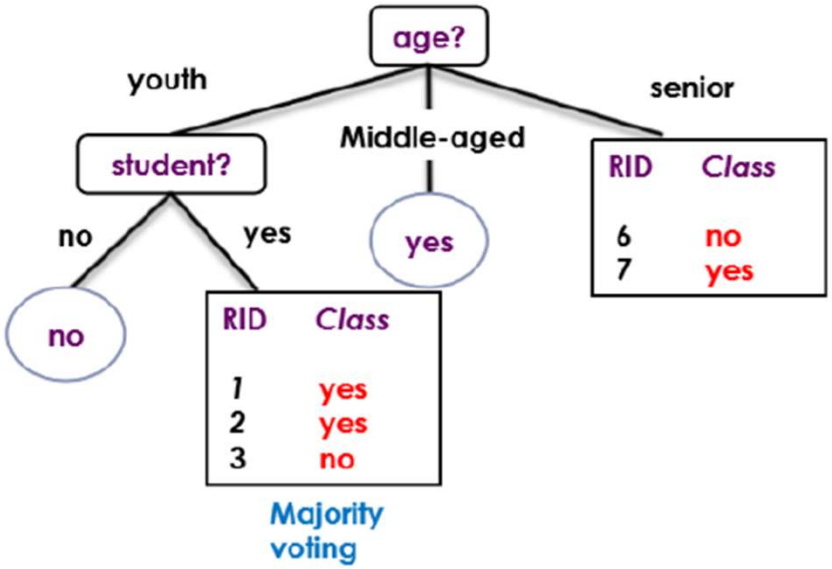
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



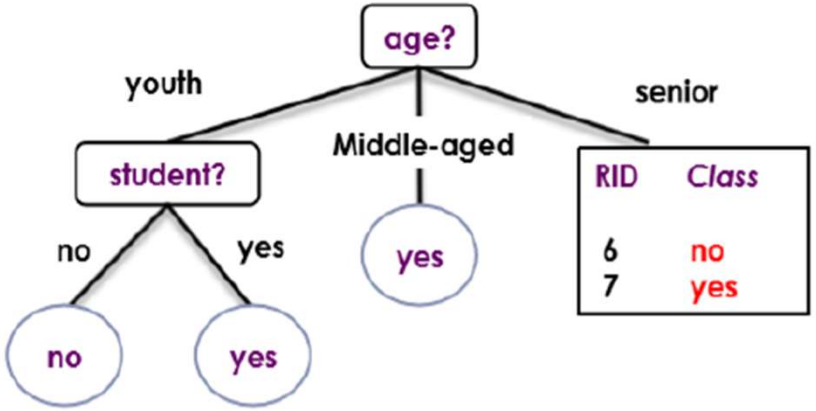
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



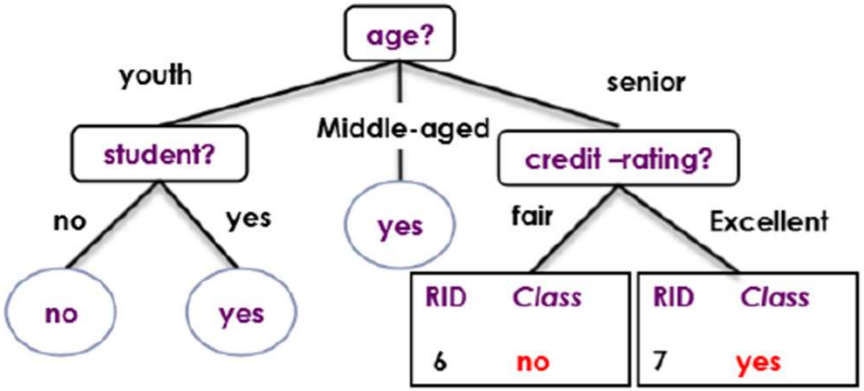
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



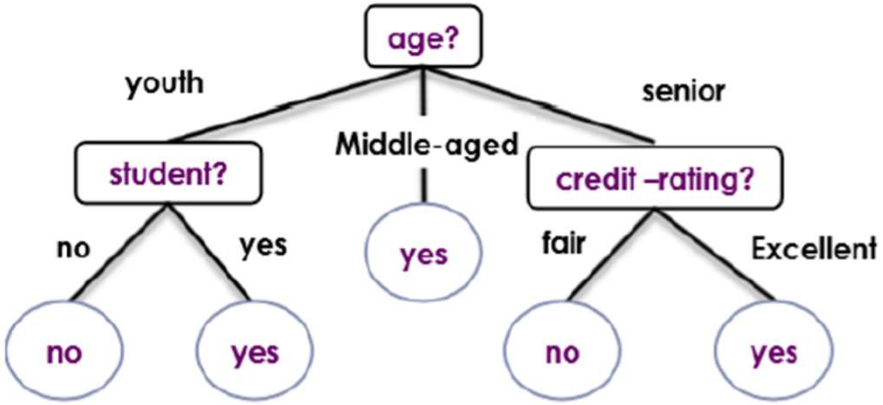
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes



معیار انتخاب صفات و ویژگی

معیار انتخاب ویژگی (قاعده‌ی تقسیم) یک روش ابتکاری برای انتخاب معیار تقسیم‌بندی است که به بهترین شکل، یک بخش داده را جدا می‌کند. ایده‌آل این است که:

- هر بخش حاصل باید خالص باشد (شامل نمونه‌هایی که همگی به یک کلاس تعلق دارند).
- برای هر ویژگی، امتیازی تعیین شده و ویژگی با بالاترین امتیاز انتخاب می‌شود.
- می‌تواند یک نقطه‌ی تقسیم برای ویژگی‌های پیوسته یا یک زیرمجموعه‌ی تقسیم برای درخت‌های دودویی تعیین کند.

به بررسی معیارهای **Information Gain**، **Gain Ratio** و **Gini Index** خواهیم پرداخت.

معیار Information Gain

بهره اطلاعات همان مفهوم آنتروپی اطلاعات است.

آنتروپی اطلاعات مفهومی از نظریه اطلاعات است. این مفهوم بیان می‌کند که در یک رویداد چه میزان اطلاعات وجود دارد.

به‌طور کلی، هر چه یک رویداد نامطمئن‌تر یا تصادفی‌تر باشد، میزان اطلاعات موجود در آن بیشتر خواهد بود.

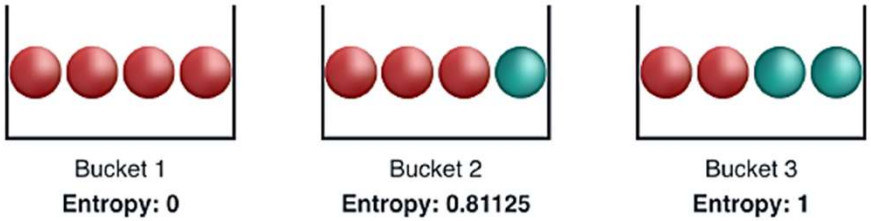
فرمول زیر محاسبه آنتروپی را نشان می دهد که H همان آنتروپی اطلاعات می باشد:

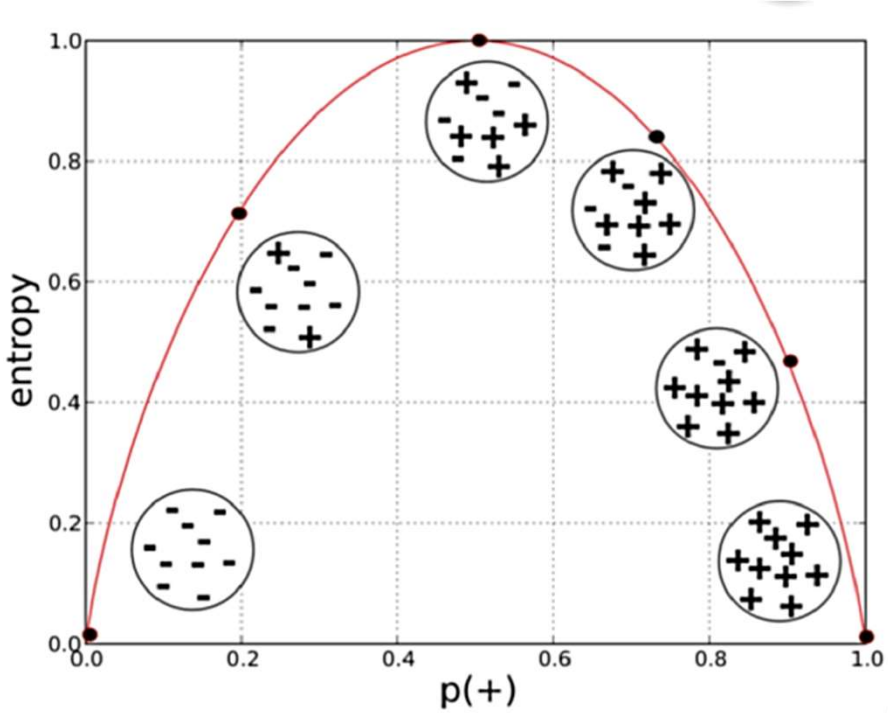
$$X \in \{x_1, x_2, \dots, x_m\}$$
$$\{p_1, p_2, \dots, p_m\}$$

$$H(X) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

دنبال معیاری هستیم که آنتروپی آن کمتر باشد.

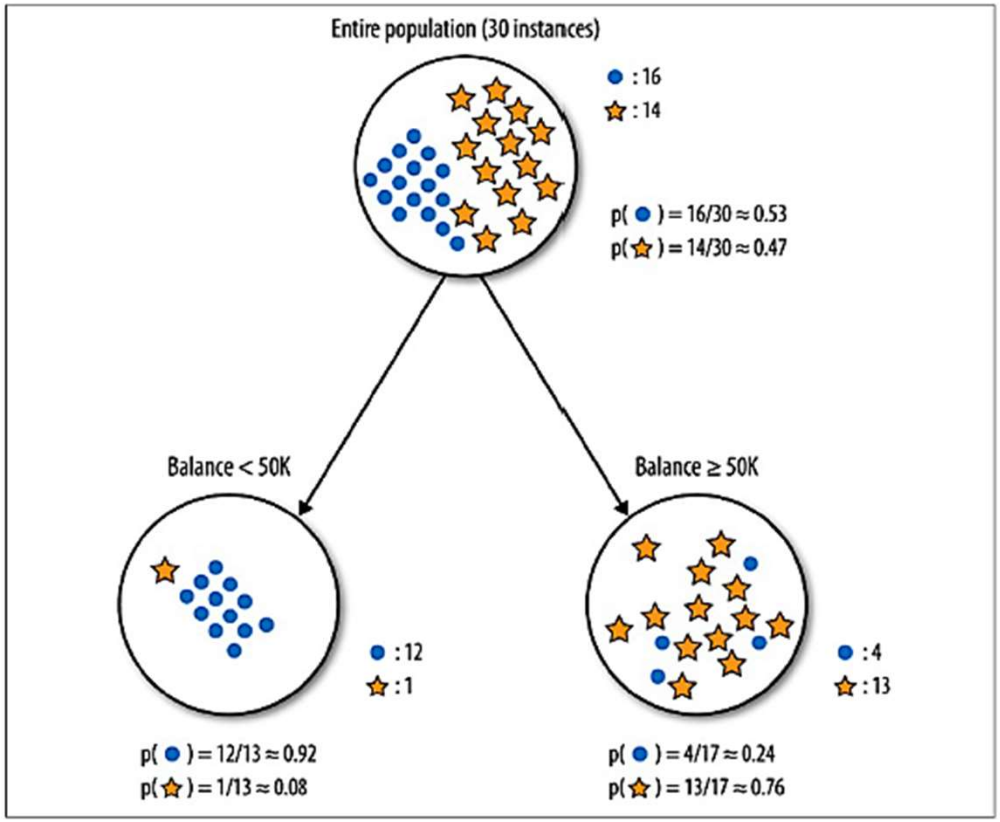
آنتروپی = عدم قطعیت





https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4

Source: Data Science for Business



Source: Data Science for Business

آنتروپی بالا

- از یک توزیع یکنواخت (هیستوگرام مسطح) آمده است.
- مقادیر نمونه‌گیری شده از آن کمتر قابل پیش‌بینی هستند.

آنتروپی پایین

- از یک توزیع متغیر (هیستوگرام دارای قله‌ها و دره‌های متعدد) آمده است.
- مقادیر نمونه‌گیری شده از آن بیشتر قابل پیش‌بینی هستند.

محاسبه Information Gain

مرحله ۱: محاسبه آنتروپی مجموعه D

آنتروپی، میانگین مقدار اطلاعات مورد نیاز برای شناسایی برچسب کلاس یک نمونه در D را نشان می‌دهد.

m: تعداد کلاس‌ها

pi: احتمال این که یک نمونه دلخواه در D متعلق به کلاس i باشد. این احتمال با رابطه زیر تخمین زده می‌شود:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

یک تابع لگاریتم به پایه ۲ استفاده می‌شود زیرا اطلاعات به صورت بیت کدگذاری می‌گردد.

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

9 tuples in class yes
5 tuples in class no

$$Info(D) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940 \text{ bits}$$

دانشگاه سمنان
Semnan University
پردیس فرزانگان

مرحله ۲: محاسبه آنتروپی وزنی برای هر ویژگی پس از تقسیم‌بندی با استفاده از آن ویژگی

فرض کنید می‌خواهیم نمونه‌های موجود در D را بر اساس یک ویژگی A تقسیم‌بندی کنیم.
برای این کار:

تقسیم‌بندی بر اساس ویژگی A: هر مقدار ممکن از A یک بخش جدید ایجاد می‌کند. این بخش‌ها مجموعه‌های فرعی مختلفی از D خواهند بود.

محاسبه آنتروپی وزنی برای هر بخش: برای هر بخش ایجادشده توسط مقدارهای مختلف ویژگی A، آنتروپی محاسبه می‌شود و هر آنتروپی متناسب با اندازه بخش خود وزن‌دهی می‌شود.

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

الگوریتم *ID3*

این الگوریتم یکی از ساده‌ترین الگوریتم‌های درخت تصمیم است که از معیار *Information Gain* استفاده می‌کند. در اجرای این الگوریتم دو شرط توقف وجود دارد. یکی این که کلیه‌ی نمونه‌های باقیمانده متعلق به یک کلاس باشند و یا اینکه پس از محاسبه‌ی مقدار معیار *Information Gain* بهترین آن بزرگتر از صفر نباشد. هیچگونه روش هرس کردنی در آن موجود نیست و می‌تواند صفات‌خاصه‌ی عددی و داده‌های ناقص را به عنوان ورودی بپذیرد.

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694 \text{ bits.}$$

مرحله ۳: محاسبه بهره اطلاعاتی **Information Gain**

$$Gain(A) = Info(D) - Info_A(D)$$

ویژگی با بیشترین Information Gain به عنوان ویژگی تقسیم انتخاب می‌شود.

این ویژگی بالاترین قدرت را در جداسازی داده‌ها دارد و با کاهش بیشترین میزان ابهام، بیشترین اطلاعات را فراهم می‌کند.

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$
$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

مثال

جدول ۲-۷: نمونه‌ای از یک داده‌ی آزمایشی

ID	Age	Income	Job	Computer
1	Old	Medium	Student	No
2	Middle	High	Teacher	No
3	Old	Low	Teacher	No
4	Young	Medium	Teacher	Yes
5	Young	Low	Teacher	Yes
6	Old	Medium	Student	Yes
7	Middle	Medium	Student	Yes
8	Young	High	Teacher	No
9	Old	High	Student	No
10	Middle	High	Student	No

$$Entropy(D) = -\frac{4}{10} Log_2(\frac{4}{10}) - \frac{6}{10} Log_2(\frac{6}{10}) = 0.970$$

$$Domain(Age) = \{Old, Middle, Young\}$$

$$Domain(Income) = \{High, Medium, Low\}$$

$$Domain(Job) = \{Teacher, Student\}$$

$$Entropy_{Age}(D) = \frac{4}{10} \times (-\frac{3}{4} Log_2(\frac{3}{4}) - \frac{1}{4} Log_2(\frac{1}{4})) +$$
$$\frac{3}{10} \times (-\frac{2}{3} Log_2(\frac{2}{3}) - \frac{1}{3} Log_2(\frac{1}{3})) +$$
$$\frac{3}{10} \times (-\frac{1}{3} Log_2(\frac{1}{3}) - \frac{2}{3} Log_2(\frac{2}{3})) = 0.875$$

$$Entropy_{Income}(D) = \frac{4}{10} \times (-\frac{4}{4} Log_2(\frac{4}{4}) - \frac{0}{4} Log_2(\frac{0}{4})) +$$
$$\frac{4}{10} \times (-\frac{1}{4} Log_2(\frac{1}{4}) - \frac{3}{4} Log_2(\frac{3}{4})) +$$
$$\frac{2}{10} \times (-\frac{1}{2} Log_2(\frac{1}{2}) - \frac{1}{2} Log_2(\frac{1}{2})) = 0.524$$

$$Entropy_{Job}(D) = \frac{5}{10} \times (-\frac{3}{5} Log_2(\frac{3}{5}) - \frac{2}{5} Log_2(\frac{2}{5})) +$$
$$\frac{5}{10} \times (-\frac{3}{5} Log_2(\frac{3}{5}) - \frac{2}{5} Log_2(\frac{2}{5})) = 0.970$$

$$Entropy(D) = -\frac{4}{10} Log_2(\frac{4}{10}) - \frac{6}{10} Log_2(\frac{6}{10}) = 0.970$$

$$InformationGain(Age) = 0.970 - 0.875 = 0.095$$

$$InformationGain(Income) = 0.970 - 0.524 = 0.446$$

$$InformationGain(Job) = 0.970 - 0.970 = 0$$

به دلیل اینکه صفت‌خاصه‌ی درآمد دارای بیشترین مقدار است، برای ریشه‌ی درخت انتخاب می‌شود. چون در دامنه‌ی این صفت‌خاصه می‌توان ۳ مقدار متمایز یافت، بنابراین سه شاخه از این گره منشعب می‌شود که هر یک با مقادیر سه گانه‌ی این صفت‌خاصه برچسب خورده‌اند

درواقع داده‌های آموزشی به ۳ زیرمجموعه افراز می‌شوند. برای هر یک از این زیرمجموعه‌ها همانند قبل بهترین صفت‌خاصه جهت انشعاب محاسبه می‌شود. این بار صفت‌خاصه‌ی درآمد در محاسبات شرکت نمی‌کند و از میان دیگر صفات یکی انتخاب می‌شود. این کار تا هنگامی که یکی از شروط توقف الگوریتم محقق شود، ادامه می‌یابد.

Gain Ratio

Information Gain به ویژگی‌هایی که دارای تعداد زیادی مقادیر هستند تمایل دارد.

به عنوان مثال، این اندازه‌گیری تمایل به انتخاب ویژگی‌هایی مانند شماره دانشجویی دارد که اصلاً ویژگی مفیدی برای طبقه‌بندی نیست.

الگوریتم C4.5 از نسبت بهره Gain Ratio استفاده می‌کند تا این مشکل را برطرف کند (نرمال‌سازی بهره اطلاعات).

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$\text{GainRatio(A)} = \frac{Gain(A)}{SplitInfo(A)}$$

مثال Gain Ratio

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

$$Gain(income) = 0.029 \quad \Rightarrow \quad GainRatio(income) = 0.029/1.557 = 0.019$$

معیار Gini Index

شاخص جینی در الگوریتم CART استفاده می‌شود و میزان ناخالصی (یا عدم خلوص) یک مجموعه داده را اندازه‌گیری می‌کند.

اگر مجموعه داده D شامل نمونه‌هایی از m کلاس مختلف باشد، gini(D) به صورت زیر تعریف می‌شود:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

اگر مجموعه داده D بر اساس ویژگی A به دو زیرمجموعه D1 و D2 تقسیم شود، به صورت زیر تعریف می‌شود:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$
\begin{aligned}
&Gini_{income \in \{low,medium\}}(D) \\
&\quad = \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\
&\quad = \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) \\
&\quad = 0.443 \\
&\quad = Gini_{income \in \{high\}}(D).
\end{aligned}
$$

$Gini_{\{low, high\}}$ is 0.458; $Gini_{\{medium, high\}}$ is 0.450. Thus, split on the {low, medium} (and {high}) since it has the lowest Gini index

جدول ۳-۷ اطلاعات مشتریانی که درخواست وام داشتند را نشان می‌دهد.

جدول ۳-۷: اطلاعاتی جهت تصمیم‌گیری برای اعطای وام

ID	Age	Job	House	Credit	Class
1	*Old*	*False*	*True*	*Excellent*	*Yes*
2	*Old*	*False*	*True*	*Good*	*Yes*
3	*Middle*	*False*	*False*	*Fair*	*No*
4	*Middle*	*True*	*True*	*Good*	*Yes*
5	*Young*	*False*	*False*	*Fair*	*No*
6	*Old*	*False*	*False*	*Fair*	*No*
7	*Middle*	*False*	*True*	*Excellent*	*Yes*
8	*Young*	*True*	*False*	*Good*	*Yes*
9	*Young*	*True*	*True*	*Fair*	*Yes*
10	*Middle*	*False*	*False*	*Good*	*No*

سن، داشتن شغل و منزل مسکونی و همچنین اعتبار مشتریان در تصمیم‌گیری برای اعطای وام موثر هستند.

ازآنجا که ۱۰ نمونه داده به دو کلاس *Yes* و *No* به ترتیب به نسبت ۶ و ۴ توزیع شده است، داریم:

$$Gini(D) = 1 - (\frac{6}{10})^2 - (\frac{4}{10})^2 = 0.48$$

جهت یافتن بهترین صفت‌خاصه، معیار *Gini Index* برای کلیه‌ی صفات‌خاصه محاسبه می‌شود. در مجموعه دامنه‌ی دو صفت‌خاصه‌ی شغل و منزل مسکونی می‌توان دو مقدار *True* و *False* را یافت و به همین دلیل محاسبه‌ی معیار خیلی سخت نیست.

$$Gini_{Job}(D) = \frac{7}{10} \times (1 - (\frac{4}{7})^2 - (\frac{3}{7})^2) + \frac{3}{10} \times (1 - (\frac{0}{3})^2 - (\frac{3}{3})^2) = 0.343$$

$$Gini_{House}(D) = \frac{5}{10} \times (1 - (\frac{4}{5})^2 - (\frac{1}{5})^2) + \frac{5}{10} \times (1 - (\frac{0}{5})^2 - (\frac{5}{5})^2) = 0.160$$

اما صفات‌خاصه‌ی سن و اعتبار هر یک دارای سه مقدار هستند و از آنجا که معیار *Gini Index* یک انشعاب دودویی را برای هر یک از این صفات‌خاصه می‌سازد، باید کلیه‌ی حالات تقسیم مقادیر برای این صفات‌خاصه بررسی شوند.

$\text{Gini}_{\text{Age}}(D)=0.476$ وقتی $\{Old\}\ ,\ \{Middle, Young\}$

$\text{Gini}_{\text{Age}}(D)=\underline{0.467}$ وقتی $\{Middle\}\ ,\ \{Old, Young\}$

$\text{Gini}_{\text{Age}}(D)=0.476$ وقتی $\{Young\}\ ,\ \{Middle, Old\}$

$\text{Gini}_{\text{Credit}}(D)=0.400$ وقتی $\{Excellent\}\ ,\ \{Good, Fair\}$

$\text{Gini}_{\text{Credit}}(D)=0.450$ وقتی $\{Good\}\ ,\ \{Excellent, Fair\}$

$\text{Gini}_{\text{Credit}}(D)=\underline{0.317}$ وقتی $\{Fair\}\ ,\ \{Excellent, Good\}$

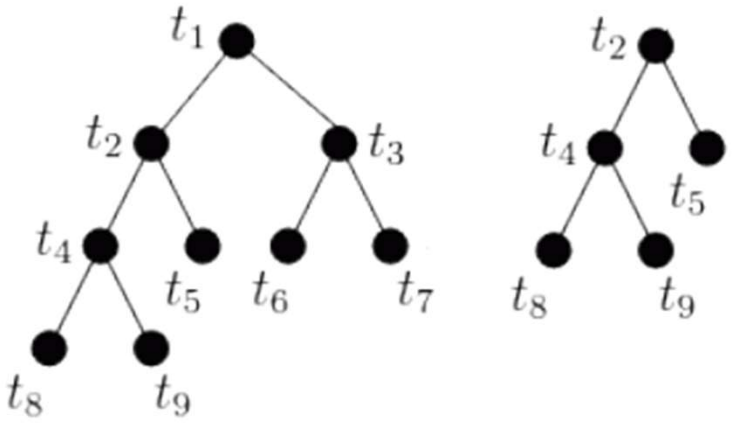
همانطور که ملاحظه می‌کنید بهترین انشعاب برای صفت‌خاصه‌ی سن هنگامی است که مقادیر پیر و جوان در یک گروه و میانسال در گروه دیگر قرار می‌گیرند و مقدار *Gini Index* برابر با ۰/۴۶۷ است. بطور مشابه مقدار این معیار برای صفت‌خاصه‌ی اعتبار برابر با ۰/۳۱۷ می‌باشد. در مرحله‌ی اول از میان صفات‌خاصه، منزل مسکونی با مقدار ۰/۱۶ انتخاب می‌شود. با تکرار عملیات فوق در نهایت ما با یک درخت تصمیم دودویی روبرو خواهیم بود.

هرس کردن Pruning

برای ساده‌تر کردن مدل و کاهش احتمال بیش‌برازش overfitting

اصل تیغ اوکام: Occam's razor

از بین دو توضیح برای چیزی، توضیحی که به احتمال زیاد درست است، ساده‌ترین آن‌هاست.



پیش هرس و پس هرس
Pre-pruning vs. Post-pruning

پیش-هرس کردن
قبل از اینکه الگوریتم به یک درخت کامل رشد کند، متوقفش کنید.

پس-هرس کردن
درخت تصمیم را به‌طور کامل رشد دهید.
گره‌های درخت تصمیم را به‌صورت از پایین به بالا هرس کنید.
اگر پس از هرس کردن، خطای تعمیم بهبود یافت، زیردرخت را با یک گره برگ جایگزین کنید.
برچسب کلاس گره برگ از کلاس اکثریت نمونه‌ها در زیردرخت تعیین می‌شود.

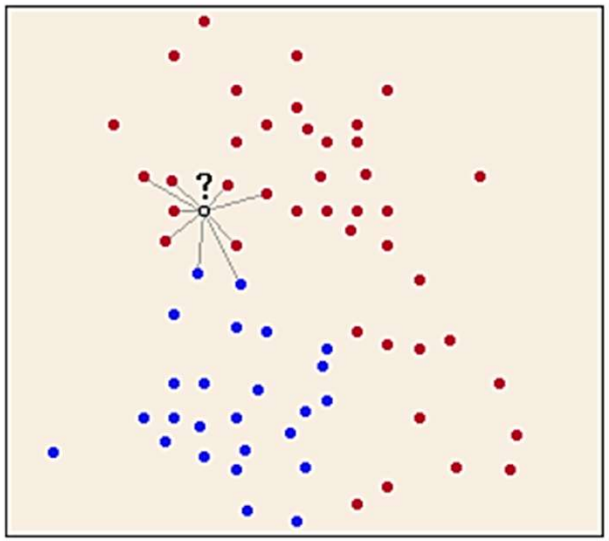
پیش-هرس کردن سریع‌تر است اما دقت کمتری دارد.

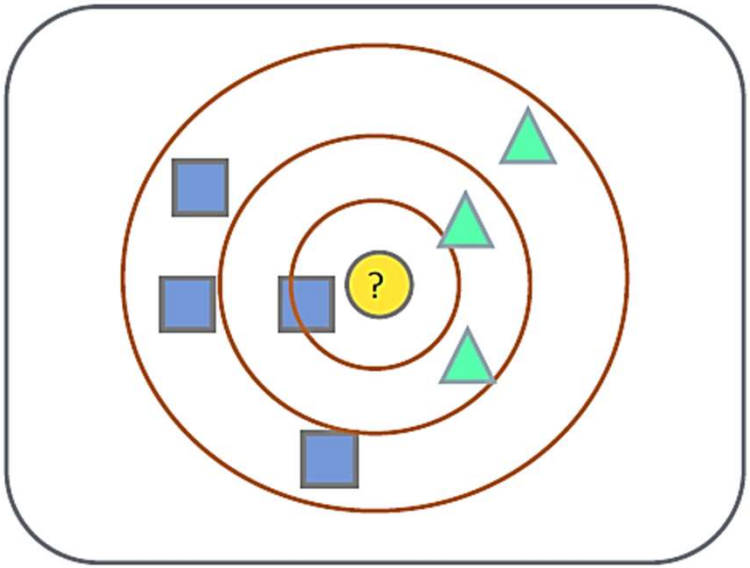
دانشگاه سمنان
Semnan University
پردیس فرزانگان

K نزدیک‌ترین همسایه
K-Nearest Neighbors (KNN)

از ساده‌ترین الگوریتم‌های داده‌کاوی است.
نیازمند سه چیز است:

- فضای ویژگی‌ها (داده‌های آموزشی)
- معیار فاصله
- مقدار k





- k = 1:
  - Belongs to square class
- k = 3:
  - Belongs to triangle class
- k = 7:
  - Belongs to square class

KNN

- قابل استفاده برای مسائل طبقه‌بندی و رگرسیون
- ترکیب برچسب‌های k نزدیک‌ترین همسایه:
- گرفتن رأی اکثریت (میانگین) برچسب‌ها بین همسایه‌ها
- وزن‌دهی: $w = \frac{1}{d} \text{ or } \frac{1}{d^2}$
- مثال: فرض کنید این‌ها سه نزدیک‌ترین همسایه هستند.

Value	5	8	9
Distance	3	2	5

$$\text{prediction} = \frac{\frac{1}{3}\times 5 + \frac{1}{2}\times 8 + \frac{1}{5}\times 9}{\frac{1}{3}+\frac{1}{2}+\frac{1}{5}}$$

انتخاب مقدار K :

- اگر k خیلی کوچک باشد، مدل به نقاط نویزی حساس می‌شود.
- اگر k خیلی بزرگ باشد، همسایگی ممکن است شامل نقاط نامربوط شود.
- مقدار k را فرد انتخاب کنید تا از تساوی آرا جلوگیری شود.

1. دسته‌بندهای نزدیک‌ترین همسایه، یادگیرندگان تنبل هستند:
همه محاسبات تا زمان طبقه‌بندی به تعویق می‌افتد.

2. مرحله آموزش سریع است، اما مرحله اعمال (پیش‌بینی) بسیار کند است.
3. نیاز به تعداد زیادی نمونه آموزشی دارد.
4. روش KNN یک روش یادگیری مبتنی بر نمونه (غیرپارامتری) است.

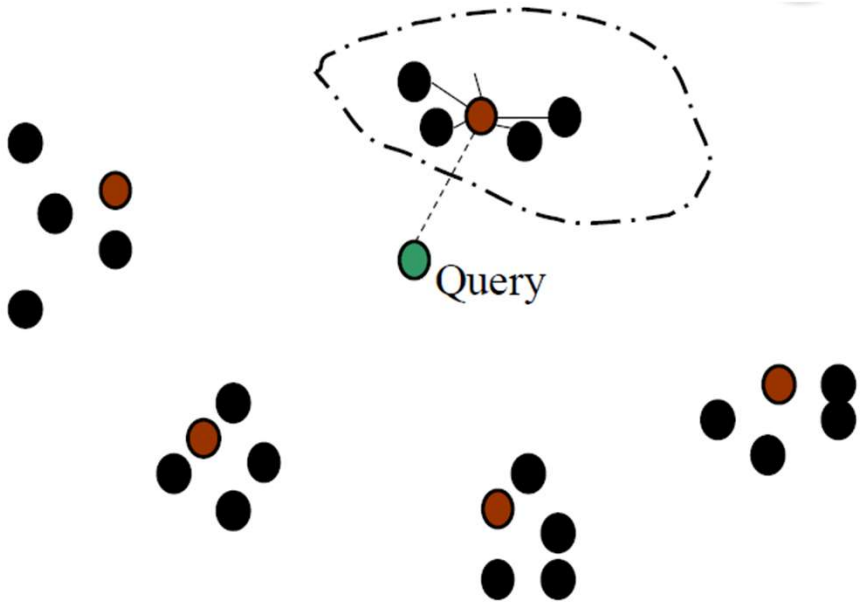
5. نیاز است ویژگی‌ها مقیاس‌بندی شوند تا از غالب شدن یک ویژگی بر معیارهای فاصله جلوگیری شود.

افزایش سرعت KNN

خوشه‌بندی داده‌های آموزشی

- جستجوی نزدیک‌ترین همسایه‌ها به نقطه تست در نزدیک‌ترین خوشه
- مصالحه بین دقت و سرعت:

نزدیک‌ترین **i** خوشه‌ها را در نظر بگیرید.



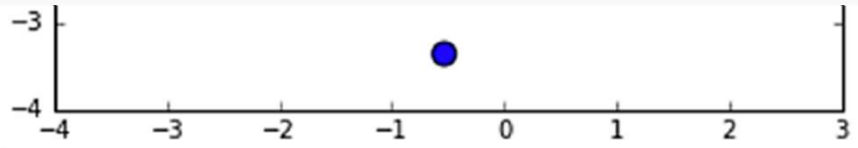
مثال پایتون

```
from sklearn.neighbors import KNeighborsClassifier

clf = KNeighborsClassifier(n_neighbors=1)
clf.fit(X,y)
print('With k=1: ', clf.predict([[-2,-2], [1,1]]))

clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X,y)
print('With k=3: ', clf.predict([[-2,-2], [1,1]]))
```

```
With k=1:  [1 1]
With k=3:  [0 1]
```



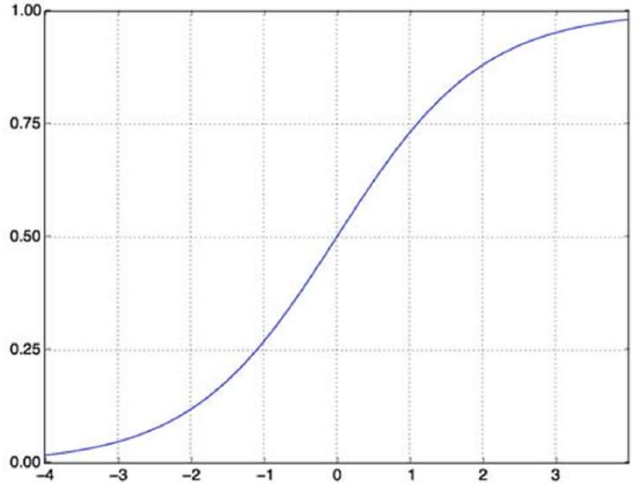
Logistic regression

- احتمال کلاس‌های مختلف را ارائه می‌دهد، به جای اینکه فقط کلاس را پیش‌بینی کند.
- فقط برای ویژگی‌های عددی کار می‌کند.
- ترکیب خطی از ویژگی ها را می سازد.

$$Z = w_0 + w_1X_1 + w_2X_2 + \cdots + w_nX_n$$

- ما تابع سیگموید (لجستیک) را بر روی z اعمال می‌کنیم تا مقداری به‌دست آوریم که بتوان آن را به‌عنوان احتمال تفسیر کرد.

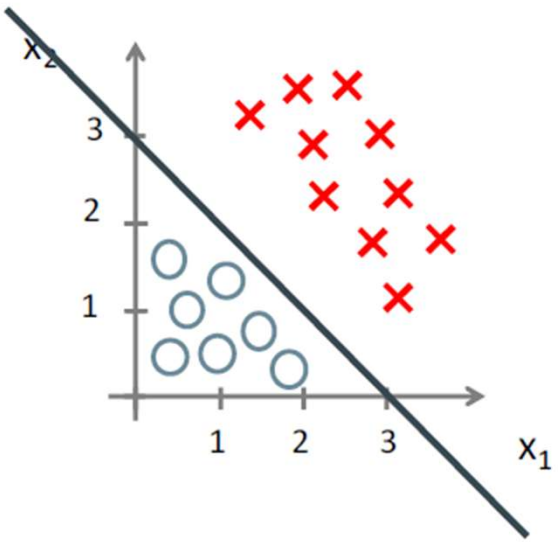
$$\sigma(Z) = \frac{1}{1+e^{-Z}}$$



فرض کنید آستانه عددی مانند ۰.۵ است.

Predict y=1 if $\sigma(z) \geq 0.5$

That is, predict y=1 if $w_0 + w_1 X_1 + w_2 X_2 + \cdots \geq 0$

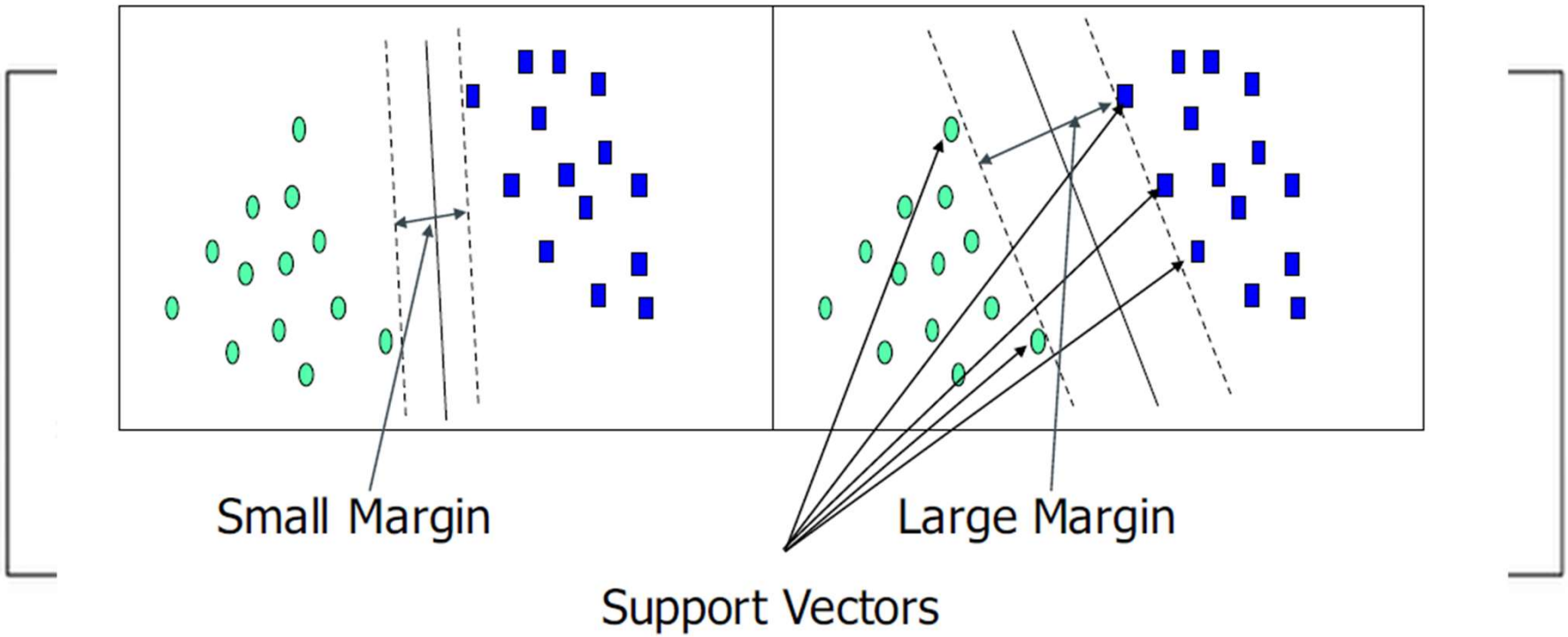


برای مثال پیش بینی می کند که y=1 اگر:

$$-3 + X_1 + X_2 \geq 0$$

یک دسته‌بند خطی است:
دارای یک مرز تصمیم‌گیری خطی است.

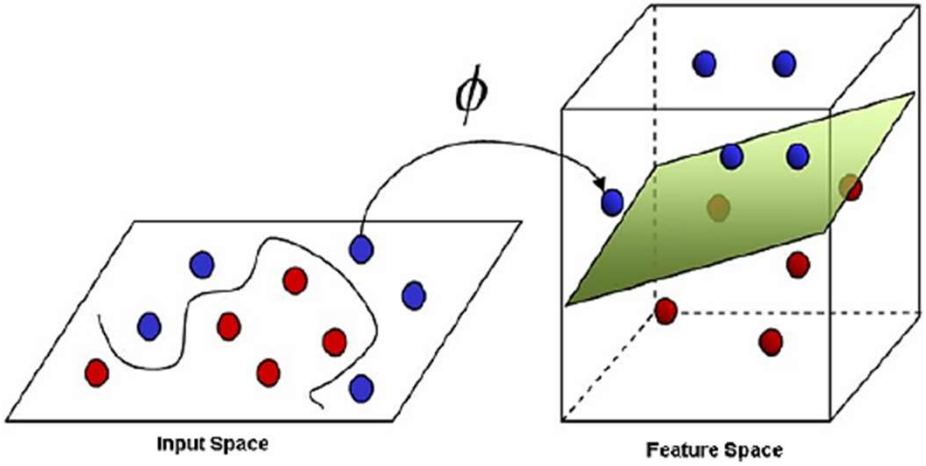
ماشین بردار پشتیبان SVM



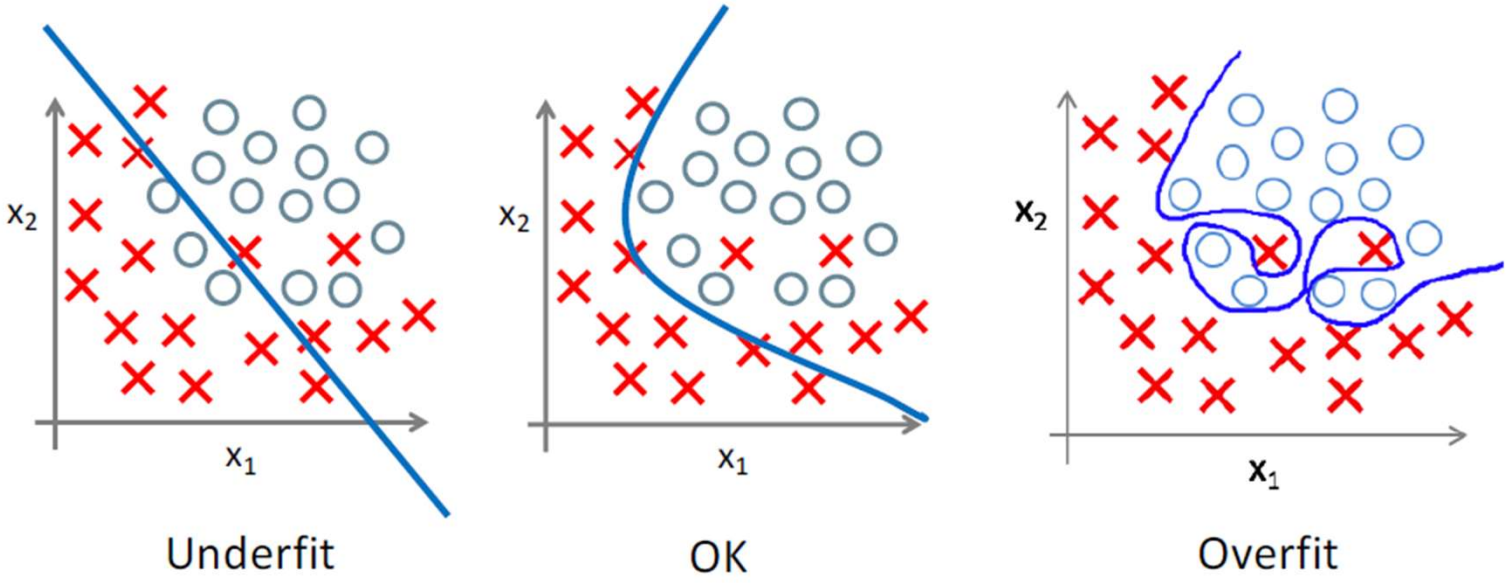
ماشین بردار پشتیبان SVM

با نگاشت غیرخطی مناسب به یک بُعد به اندازه کافی بالا (ترفند کرنل)، داده‌های دو کلاس همیشه می‌توانند با یک اَبَرصَفحه از هم جدا شوند.

در بُعد جدید، SVM به دنبال یافتن مرز تصمیم‌گیری خطی بهینه است.



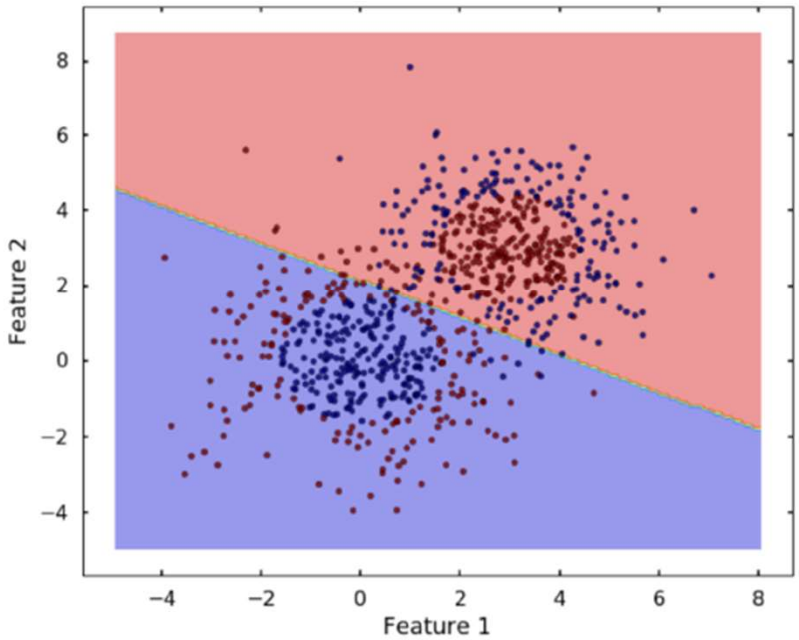
http://thecaffeinedev.com/2-support-vector-machine-learning-math-behind-part2/



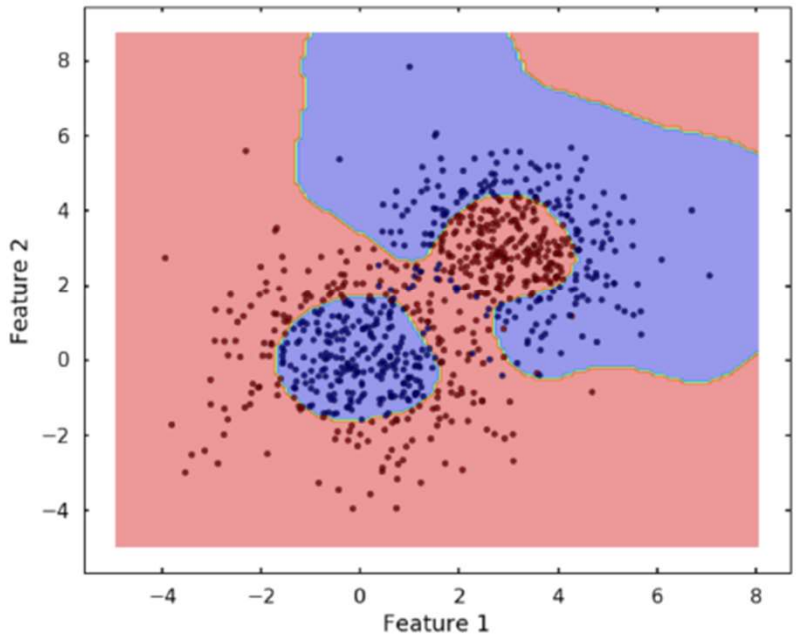
در کتابخانه‌ی sklearn، این مورد توسط پارامتر C در SVM و پارامتر gamma در کرنل rbf کنترل می‌شود.

10

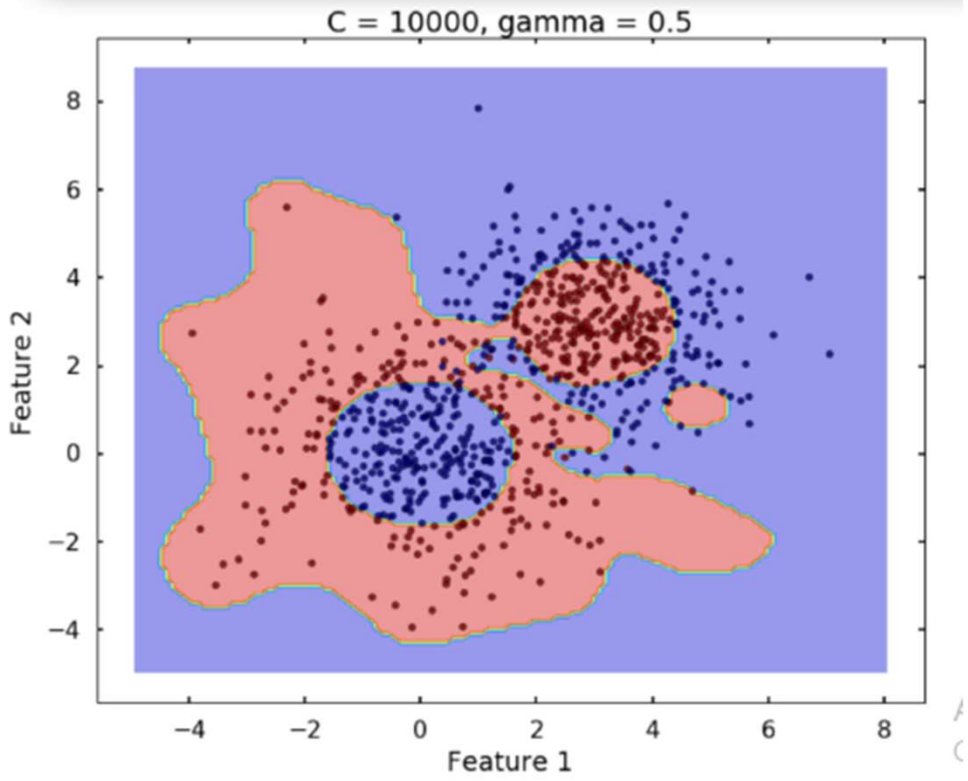
```
clf = svm.SVC(kernel='linear')
clf.fit(X,y)
```




```
clf = svm.SVC(kernel='rbf')
clf.fit(X,y)
```



```
clf = svm.SVC(kernel='rbf', C = 10000, gamma = 0.5)
clf.fit(X,y)
```



مقدار بزرگ برای C هزینه‌ی خطای دسته‌بندی اشتباه را بالا می‌برد.

این باعث می‌شود که الگوریتم برای برازش داده، از مدل انعطاف‌پذیرتری استفاده کند.

اما مقادیر خیلی بزرگ باعث می‌شود که SVM بیشتر مستعد بیش‌برازش overfit داده شود.

Evaluation ارزیابی

برای مقایسه‌ی مدل‌های مختلف

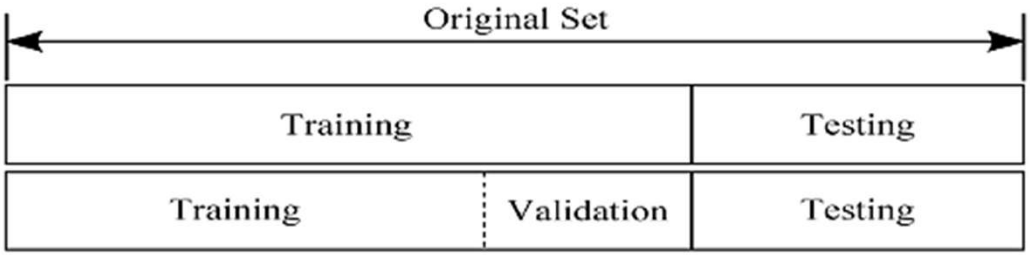
برای تنظیم اَبَرمُتغیرها مانند:

K در KNN، تعداد لایه‌ها در شبکه‌های عصبی، بهترین هرس در درخت تصمیم و غیره.

هدف اصلی در یادگیری ماشین، **تعمیم** generalization است. ما می‌خواهیم توانایی تعمیم مدل خود را اندازه‌گیری کنیم.

روش نگه‌داشتن Hold-out: شما مدل را روی داده‌های آموزش، آموزش می‌دهید و آن را روی داده‌های اعتبارسنجی validation ارزیابی می‌کنید. وقتی مدل آماده شد، یک‌بار دیگر آن را روی داده‌های آزمون نهایی تست می‌کنید.

داده‌ها را قبل از تقسیم به هم بزنید (Shuffleکنید)



ارزیابی Evaluation

چرا فقط آموزش و تست را انجام ندهیم؟

ما این تنظیمات را با استفاده از عملکرد مدل روی داده‌های اعتبارسنجی به عنوان مرحله بازخورد انجام می‌دهیم.

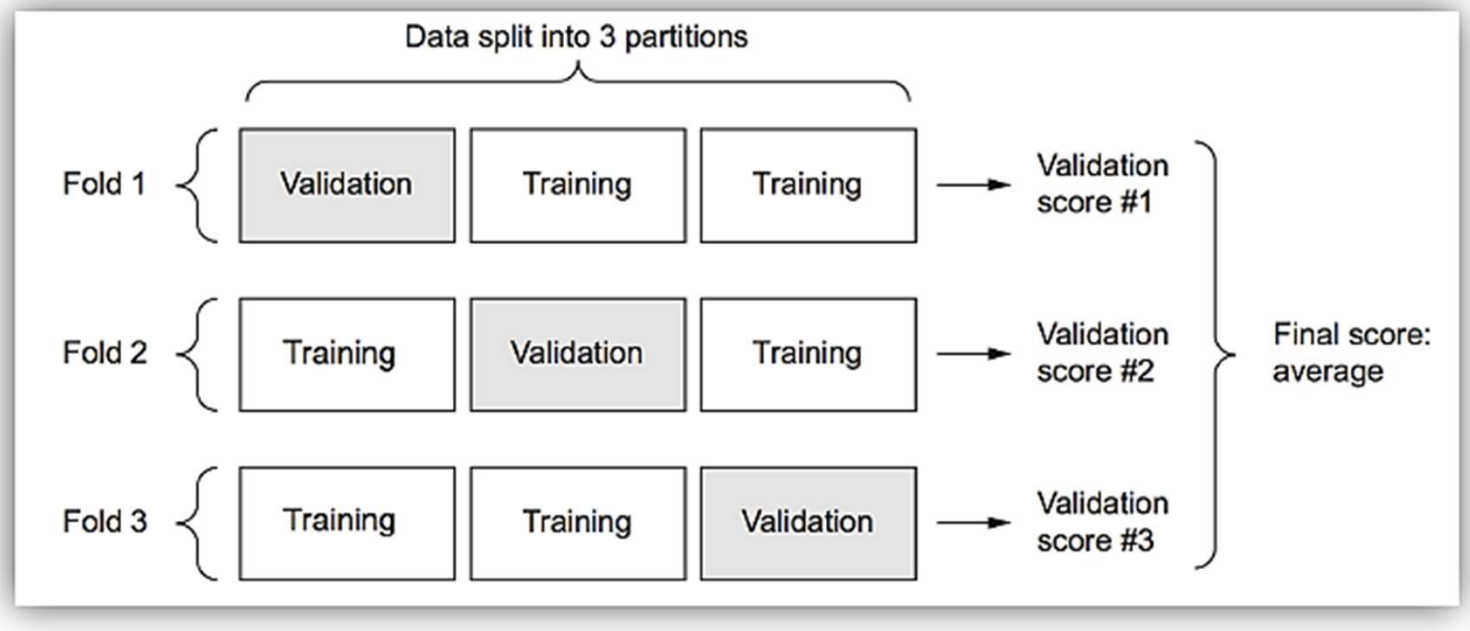
اما این کار می‌تواند به سرعت منجر به بیش‌برازش overfitting نسبت به مجموعه‌ی اعتبارسنجی شود (نشت اطلاعات).

در نهایت، ممکن است مدلی داشته باشید که روی داده‌های اعتبارسنجی به صورت مصنوعی عملکرد خوبی دارد، اما روی داده‌های کاملاً جدید خوب عمل نمی‌کند.

k-fold Cross Validation

وقتی داده‌های کمی دارید، مجموعه‌ی اعتبارسنجی بسیار کوچک خواهد بود. این موضوع مانع از ارزیابی قابل اعتماد مدل می‌شود. بنابراین، از **k-fold Cross Validation** استفاده می‌کنیم.

مقادیر معمول برای k ۵ و ۱۰ هست.



معیارهای ارزیابی مدل دسته بندی

تعداد سوابق آزمونی که به درستی (یا نادرست) توسط مدل طبقه بندی پیش بینی شده‌اند.

ماتریس سردرگمی: Confusion matrix

مثبت صحیح: True Positive شخص بیمار، به درستی بیمار تشخیص داده شود.
مثبت کاذب: False Positive شخص سالم، به اشتباه بیمار تشخیص داده شود.
منفی صحیح: True Negative شخص سالم، به درستی سالم تشخیص داده شود.
منفی کاذب: False Negative شخص بیمار، به اشتباه سالم تشخیص داده شود.

Predict class / True class	Positive	Negative
Positive	TP	FN
Negative	FP	TN

اگر m کلاس داشته باشیم، CM(i,j) در یک ماتریس سردرگمی، تعداد تاپل های کلاس i را نشان می دهد که توسط طبقه بندی کننده به عنوان کلاس j برچسب گذاری شده اند.

معیارهای ارزیابی مدل دسته بندی

Accuracy صحت: درصد نمونه هایی که درست دسته بندی شده اند.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Error rate: *1 – accuracy*, or

Error rate = (FP + FN)/All

دقت $$\text{Accuracy} = \frac{\#\text{ correct predictions}}{\text{total }\#\text{ of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

نرخ خطا $$\text{Error rate} = \frac{\#\text{ wrong predictions}}{\text{total }\#\text{ of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

حساسیت و تشخیص

- مشکل عدم تعادل کلاس: یک کلاس ممکن است نادر باشد، به عنوان مثال HIV مثبت
- اکثریت قابل توجه طبقه منفی و اقلیت طبقه مثبت یا برعکس
- در این حالتها از معیارهای زیر استفاده می‌کنیم:

- **Sensitivity**: *True Positive recognition rate*
  - **Sensitivity = TP/P**
- **Specificity**: *True Negative recognition rate*
  - **Specificity = TN/N**

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

به بیان ریاضی، حساسیت حاصل تقسیم موارد مثبت واقعی به حاصل جمع موارد مثبت واقعی و موارد منفی کاذب است.

تشخیص حاصل تقسیم موارد منفی واقعی به حاصل جمع موارد منفی واقعی و مثبت کاذب است.

Precision

دقت (Precision): چند درصد از تاپل هایی که طبقه بندی کننده به عنوان مثبت برچسب گذاری کرده است در واقع مثبت هستند.

$$precision = \frac{TP}{TP + FP}$$

Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	90	210	300
cancer = no	140	9560	9700
Total	230	9770	10000

Precision = 90/230 = 39.13%

Recall

فراخوانی(Recall): چند درصد از تاپل های مثبت طبقه بندی کننده به عنوان مثبت برچسب زده است؟

$$Recall = \frac{TP}{TP + FN}$$

Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	90	210	300
cancer = no	140	9560	9700
Total	230	9770	10000

$Recall = 90/300 = 30\%$

F-measures

F-measures: (F1 , F score) میانگین هارمونیک بین Precision و Recall

$$F = \frac{2 \times precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

معیار وزنی Precision وRecall

β بار از وزن را به درست بودن اختصاص می‌دهد

$$F_\beta = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

دانشگاه سمنان
Semnan University
پردیس فرزانگان

معیارهای ارزیابی Regression

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|y^{(i)} - \hat{y}^{(i)}\right|$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$