دانشگاه صنعتی خواجه نصیرالدین طوسی

داده کاوی

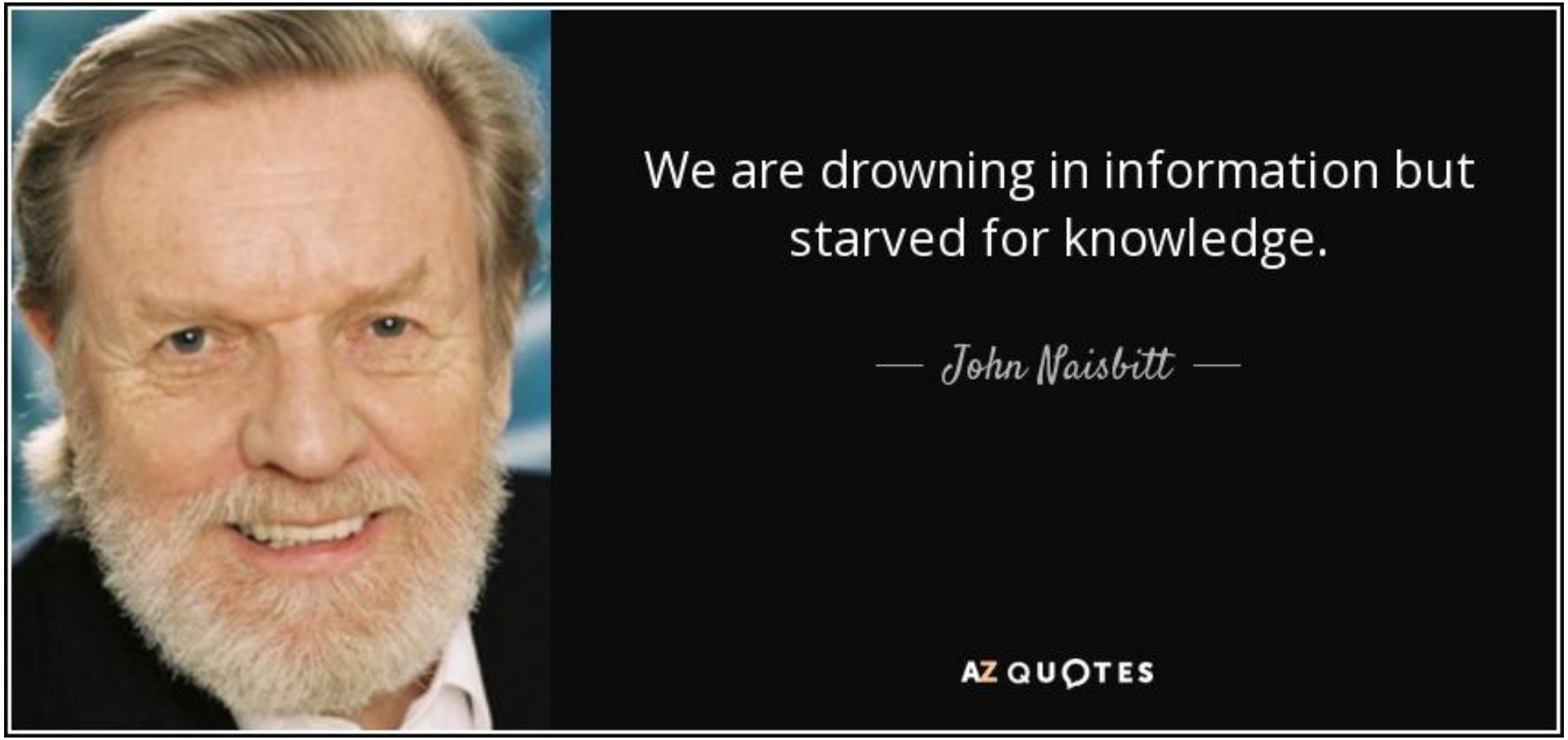
علیرضا فریدونیان

محسن کجوری نفت‌چالی

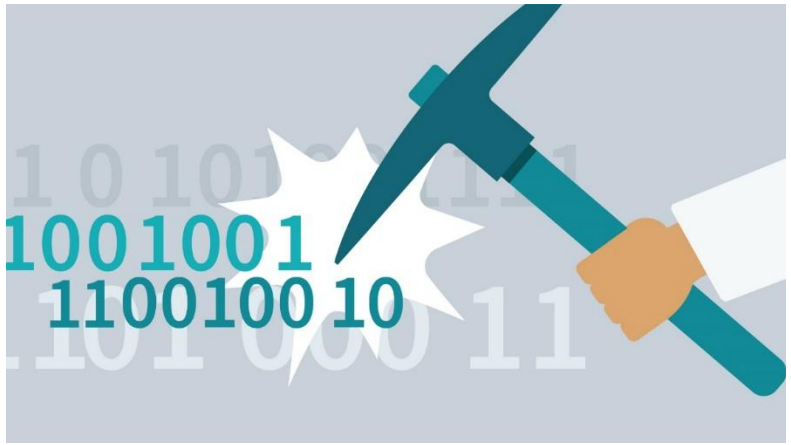
فهرست مطالب

فصل اول

مقدمه و مفاهیم


We are drowning in information but starved for knowledge.
— John Naisbitt —
AZ QUOTES

تعریف داده‌کاوی

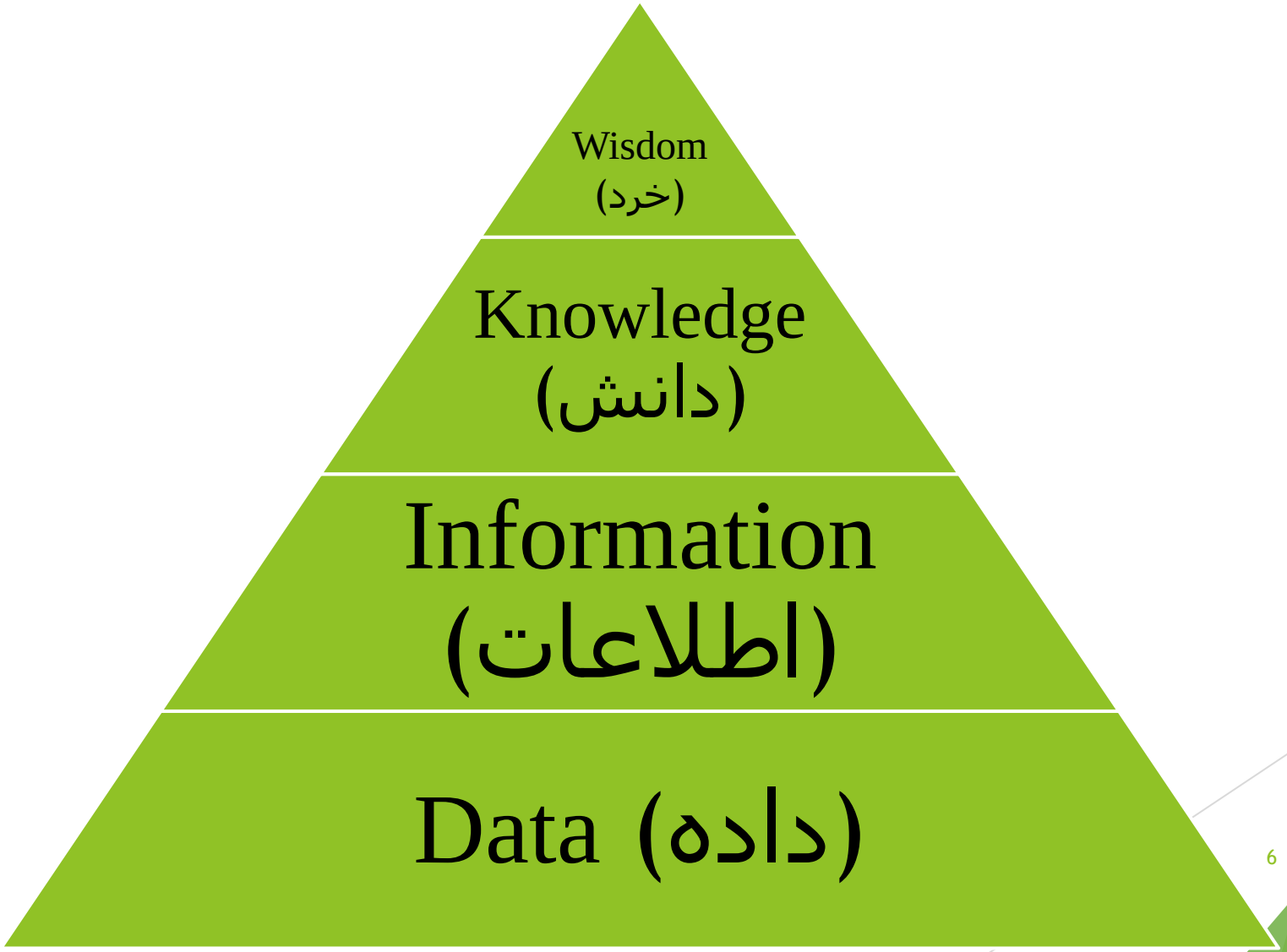


داده‌کاوی عبارتست از:
“ استخراج اطلاعات و دانش و کشف الگوهای پنهان از پایگاه داده‌های بسیار بزرگ “
-- ژیاوی هان

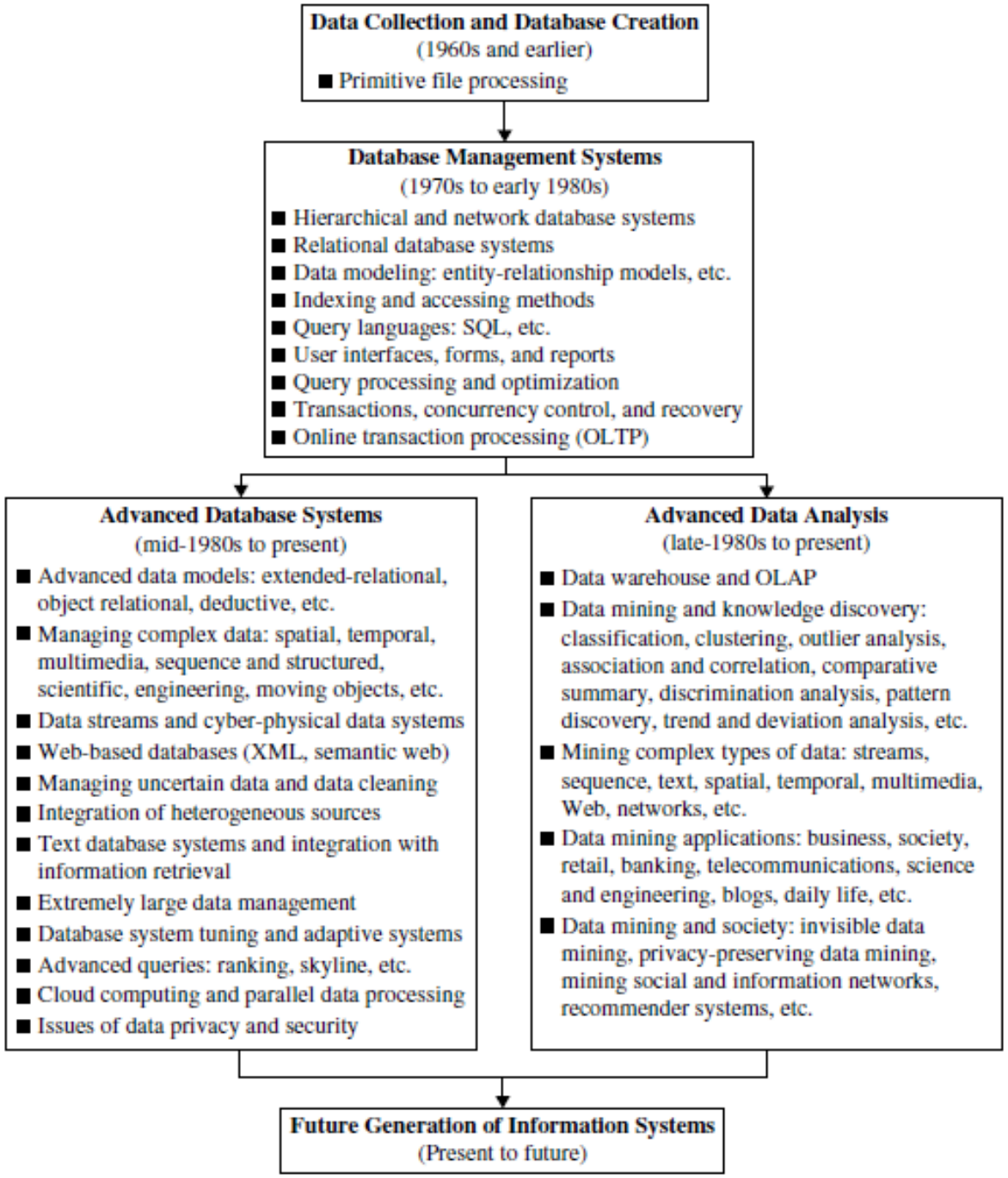
دیگرواژه‌های با معنای داده‌کاوی:

Knowledge mining from data,
Knowledge extraction,
Knowledge Discovery in Databases
Data analysis / Pattern analysis,
Data archaeology,
Data dredging

هرم سلسله‌مراتب داده تا خرد

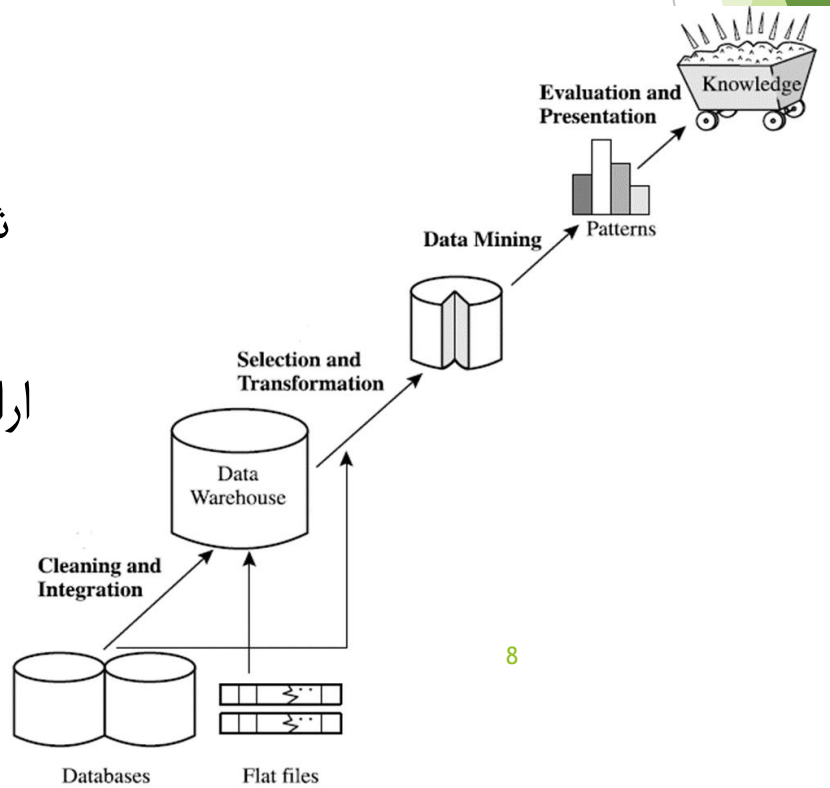


تحول داده‌کاوی بر مبنای تکامل فناوری اطلاعات (IT)



فرایند اکتشاف دانش KDD

1. Data cleaning: پاکسازی و حذف نویز و داده‌های نامربوط
2. Data integration: ترکیب داده‌ها از چند منبع
3. Data selection: بازیابی داده‌های مرتبط با تحلیل مورد نظر
4. Data transformation: تبدیل و یک‌کاسه کردن داده‌ها به فرم مناسب برای کاوش با خلاصه‌سازی یا گردآوری
5. Data mining: فرایند استخراج الگوهای داده‌ای با روش‌های هوشمند
6. Pattern evaluation: شناسایی الگوهای بازنماینده‌ی دانش واقعا جالب (سودمند) بر مبنای معیارهای جالب بودن
7. Knowledge presentation: ارایه دانش کاوش (استخراج) شده با روش‌های تجسم و بازنمایی دانش



پایگاه‌داده‌ها Database

- یک سیستم پایگاه‌داده، شامل مجموعه‌ای از داده‌های مرتبط‌به‌هم interrelated است و مجموعه‌ای از برنامه‌های نرم‌افزاری برای مدیریت و دسترسی به داده‌ها.
- برنامه‌های نرم‌افزاری سازوکار تعریف ساختارهای پایگاه‌داده‌ها و ذخیره داده‌ها را برای تخصیص و مدیریت دسترسی همزمان به داده‌ها را تعیین می‌کنند.
- یک پایگاه‌داده‌ی رابطه‌ای مجموعه‌ای از جدول‌هاست، که هر جدول شامل مجموعه‌ای از ویژگی‌ها (ستون جدول یا فیلد) و مجموعه‌ای از رکوردها (سطر جدول یا چندتایی یا tuple) است.
- هر رکورد در یک جدول رابطه‌ای نماینده‌ی یک شیء (object) است که با یک کلید (key) معین می‌شود و به وسیله‌ی مجموعه‌ای از ویژگی‌ها تشریح می‌شود.

پایگاه‌داده‌ها Database

- رابطه‌ی *customer* شامل مجموعه‌ای از ویژگی‌های تشریح کننده‌ی اطلاعات مشتری شامل یک شماره مشخصه، نام، آدرس، سن، شغل، درآمد سالیانه، اطلاعات اعتباری و بخش است.
- بهمین ترتیب، هر یک از رابطه‌های *item, employee, branch* شامل مجموعه‌ای از ویژگی‌هایی هستند که مشخصه‌های آنها را نشان می‌دهند.

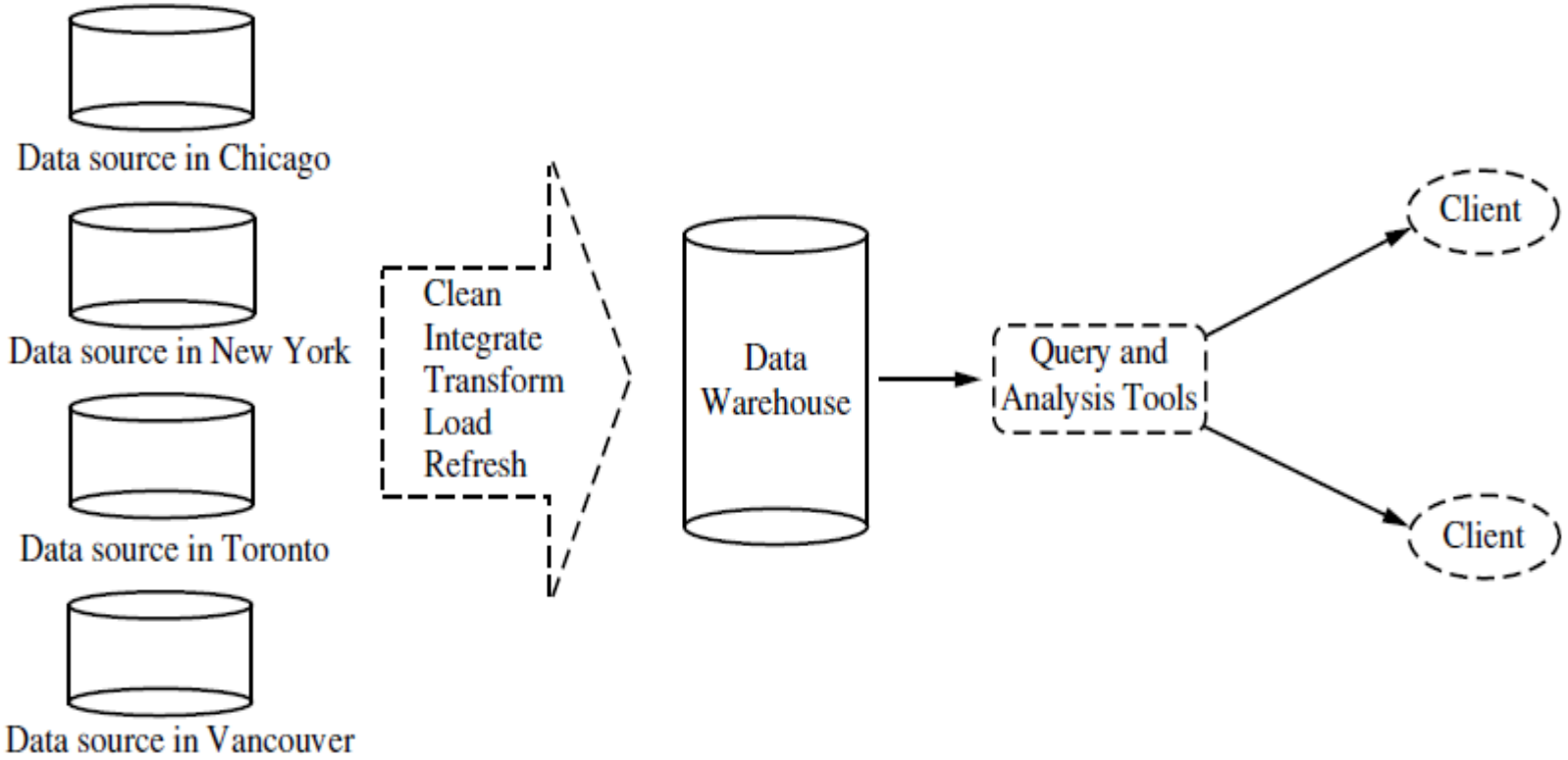
customer	(*cust_ID, name, address, age, occupation, annual_income, credit_information, category, ...*)
item	(*item_ID, brand, category, type, price, place_made, supplier, cost, ...*)
employee	(*empl_ID, name, category, group, salary, commission, ...*)
branch	(*branch_ID, name, address, ...*)
purchases	(*trans_ID, cust_ID, empl_ID, date, time, method_paid, amount*)
items_sold	(*trans_ID, item_ID, qty*)
works_at	(*empl_ID, branch_ID*)

Database پایگاه‌داده‌ها

- *purchases* مشخص کننده‌ی خریدهای *customer* است که یک تراکنش فروش توسط *employee* را ایجاد می‌کند
- *items_sold* آیتم‌های فروخته شده در یک تراکنش خاص
- *works_at* مشخص‌کننده‌ی *branch*ی از فروشگاه که *employee* در آن کار می‌کند.

customer	(*cust_ID, name, address, age, occupation, annual_income, credit_information, category, ...*)
item	(*item_ID, brand, category, type, price, place_made, supplier, cost, ...*)
employee	(*empl_ID, name, category, group, salary, commission, ...*)
branch	(*branch_ID, name, address, ...*)
purchases	(*trans_ID, cust_ID, empl_ID, date, time, method_paid, amount*)
items_sold	(*trans_ID, item_ID, qty*)
works_at	(*empl_ID, branch_ID*)

Data Warehouse انباره داده



کارکردهای داده‌کاوی

- توصیف و تفکیک:

characterization and discrimination

- اکتشاف الگوهای پرشمار، مشارکت‌ها و همبستگی‌ها:

mining of frequent patterns, associations, and correlations

- دسته‌بندی و رگرسیون:

classification and regression

- تحلیل خوشه‌بندی: clustering
- تحلیل داده‌های پرت: outlier analysis

کارکردهای داده‌کاوی

توصیف و تفکیک:

characterization and discrimination

- توصیف:

خلاصه کردن ویژگی مشترک کلاس هدف

مانند: مشخصات مشترکین پرمصرف در شرکت توزیع

- تفکیک:

مقایسه ویژگی عمومی نمونه‌هایی از یک کلاس با کلاس دیگر

مانند: مقایسه‌ی ویژگی‌های مشترکین با مصرف بالا در پیک بار و مشترکین با مصرف متوسط در پیک بار

کارکردهای داده‌کاوی

اکتشاف الگوهای پرشمار، مشارکت‌ها و همبستگی‌ها:

mining of frequent patterns, associations and correlations

- پیدا کردن اقلام یا توالی‌هایی که با هم ظاهر می‌شوند.

مانند: شیر و نان که در سبدهای خرید در کنار هم هستند

یا

همبستگی بین مرتکبین برق‌دزدی با مشترکینی که در الگوی مصرفشان تغییرات ناگهانی مشاده می‌شود.

کارکردهای داده‌کاوی

دسته‌بندی و رگرسیون:

classification and regression

- دسته‌بندی (گسسته): یافتن مدل یا تابعی برای دسته‌بندی، بر اساس یک مجموعه آموزشی از داده‌ها.

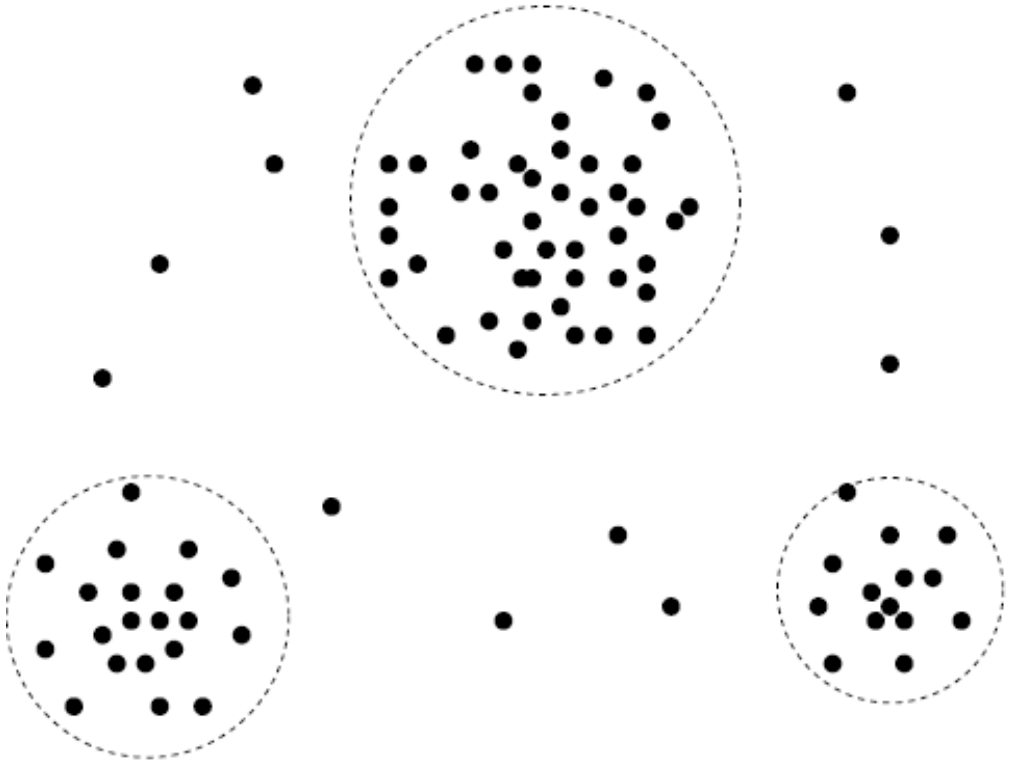
بازنمایی مدل یا تابع به صورت: قواعد اگر–آنگاه، درخت تصمیم، شبکه عصبی مصنوعی، دسته‌بندی بیز، ماشین بردار پشتیبان، و نزدیک‌ترین همسایه

- رگرسیون (پیوسته): یافتن مدل یا تابع پیوسته برای یک مجموعه پیوسته داده‌ها، بر اساس یک مجموعه آموزشی از داده‌ها.

کارکردهای داده‌کاوی

- تحلیل خوشه‌بندی: clustering

بر خلاف دسته‌بندی و رگرسیون که مجموعه داده‌هایشان دارای برچسب کلاس هستند، در تحلیل خوشه‌بندی برچسب کلاس وجود ندارد، یعنی یادگیری بدون سرپرستی است.



A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

کارکردهای داده‌کاوی

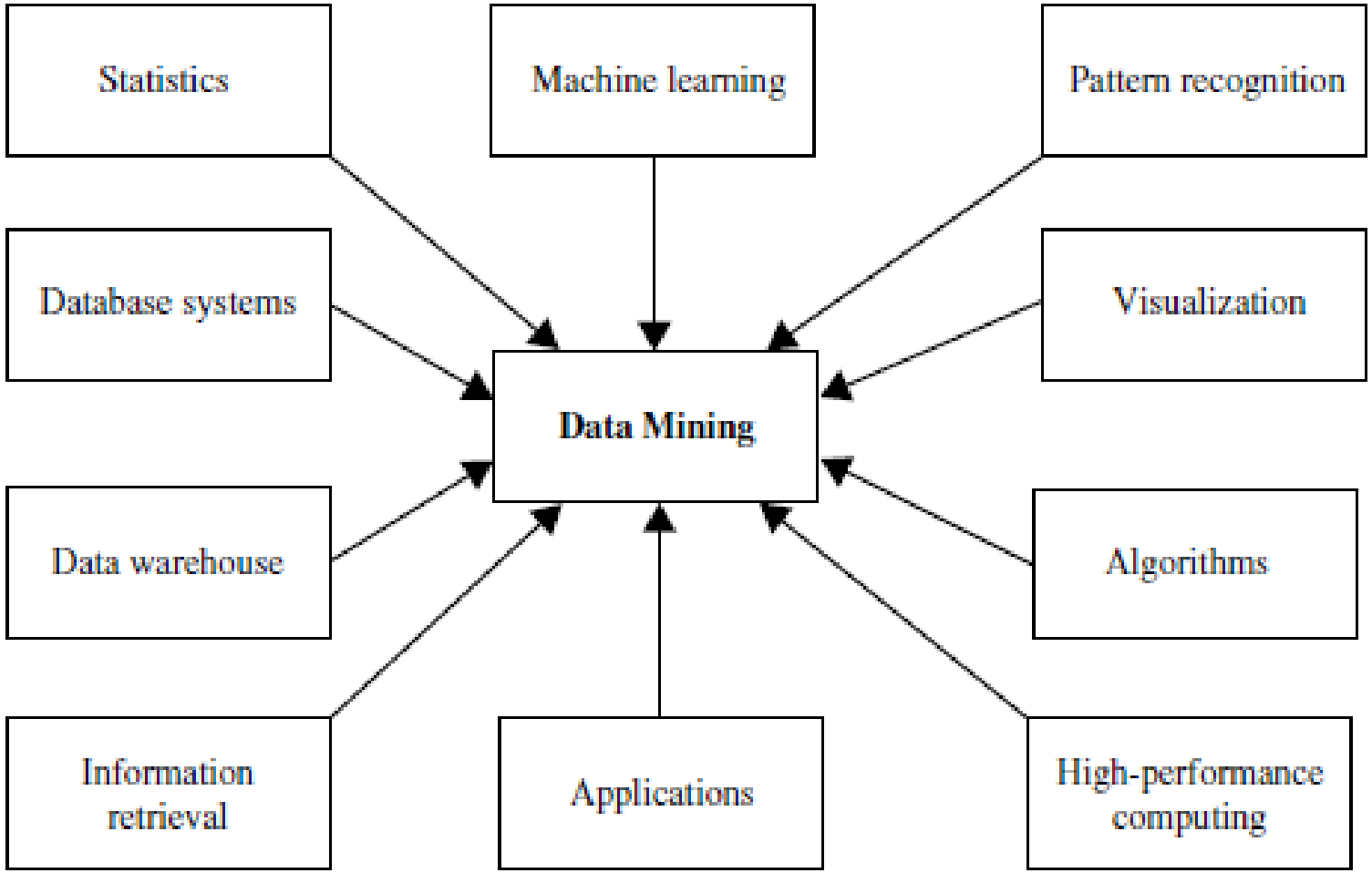
- تحلیل داده‌های پرت: outlier analysis

هدف، یافتن نمونه‌هایی است که رفتار آنها با رفتار عمومی دیگرنمونه‌ها و یا مدل داده‌ها هم‌راستا نباشد.

معمولا به عنوان نویز و استثنا حذف می‌شوند.

اما مثلا در تشخیص تقلب و برق‌دزدی می‌تواند به کار برود.

داده کاوی


Statistics
Machine learning
Pattern recognition
Database systems
Visualization
Data Mining
Data warehouse
Algorithms
Information retrieval
Applications
High-performance computing

فصل اول

پیش پردازش داده ها

پیش پردازش داده ها

- پاکسازی داده ((data cleaning

مهمترین فعالیت های این بخش عبارت است تخمین مقادیر ناموجود در پایگاه داده ها، از بین بردن اختلال (noiseدر داده ها، حذف کردن داده های پرت و نامربوط، از بین بردن ناسازگاری در داده ها.

- یکپارچه سازی داده ((data integration

در بسیاری از موارد ممکن است داده ها در فایل ها و منابع مختلف نگهداری شوند و در این صورت نیاز است تا داده ها پیش از اجرای تکنیک های داده کاوی با یکدیگر یکپارچه شوند. یکپارچه سازی هم فعالیتی سنگین است و هم چالش های فراوانی را به همراه دارد

- کاهش داده ((data reduction

ممکن است همیشه، همه داده ها مورد نیاز نباشند و تنها بخشی از داده ها که مورد نیاز است باید مورد پردازش قرار بگیرد. کاهش داده (data reductionبه این مباحث می پردازد.

- تبدیل داده ((data transformation

فعالیت های مانند نرمال سازی داده ها و گسسته سازی داده ها در این حوزه جای میگیرند.

پیش پردازش داده ها– یکپارچه سازی داده ها

ID	نام	شماره ملی	تاریخ تولد
1	سعود	127001	—
2	محسن	127015	65/7/7
3	کامران	126111	—
4	سعید	127101	—
5	علی	127551	60/8/7
6	کوروش	125111	—
7	علی	۱۲۷۵۵۱	61/5/1

ID	مشتری	کالا	تاریخ
1	1	23	6/5
2	1	25	7/7
3	2	120	3/8
4	5	255	9/7
5	5	750	9/1
6	4	23	8/6

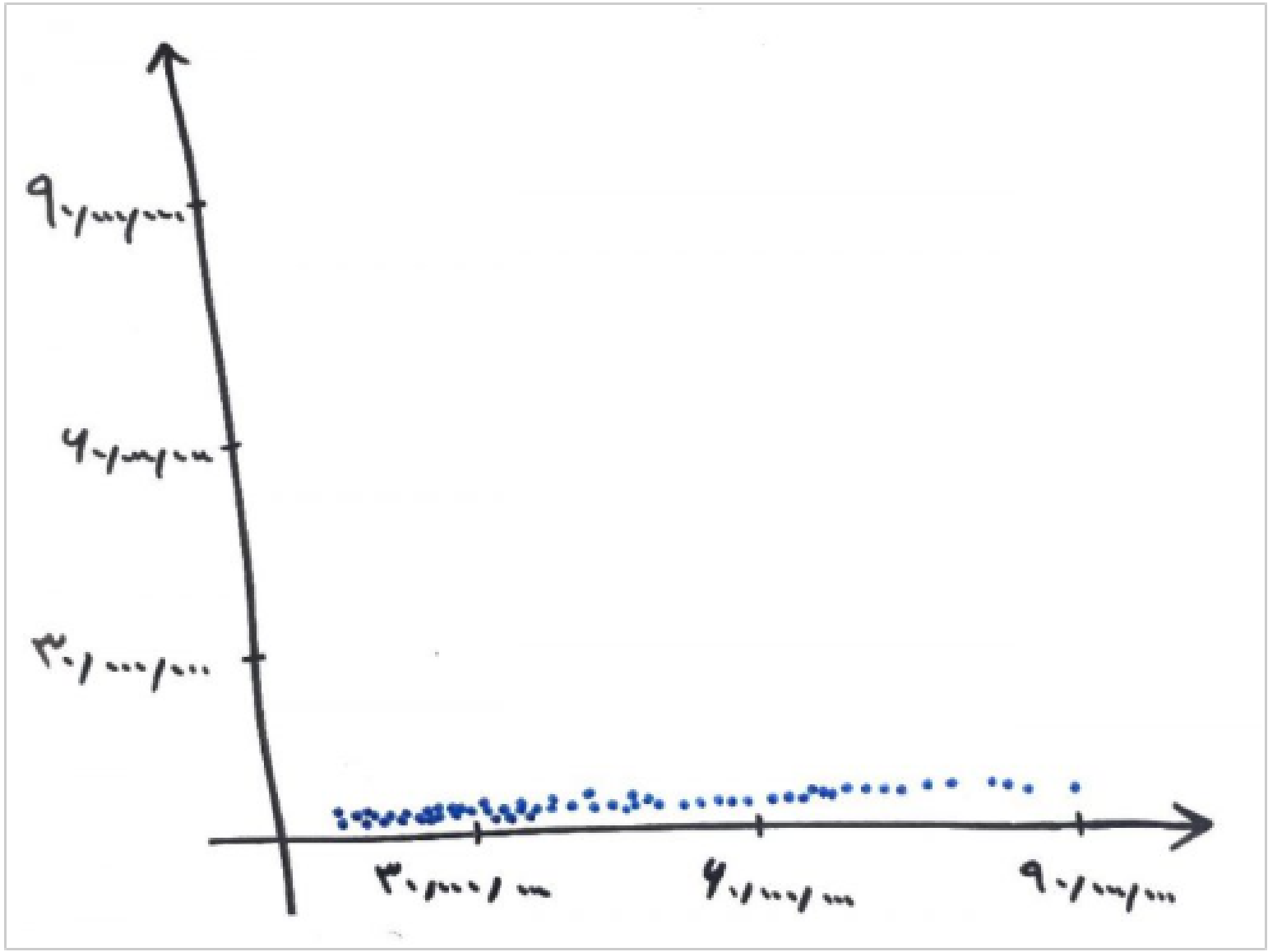
پیش پردازش داده ها– یکپارچه سازی

- در جدولِ خریدها، تمامِ آن‌ها بایستی یک شناسه‌ی مشتری را داشته باشند. در غیر اینَ صورت یکپارَچگی داده‌ها دچار مشکل می‌شود.
- مورد دیگری که باعث نقضِ یکپارچگی در مثالِ بالا می‌شود، وجود چندباره‌ی یک مشتری است.
- این دست از داده‌ها و مسائل این‌چنینی، می‌توانند باعث کثیف شدنِ داده‌ها شوند و تاثیر منفی بر روی الگوریتم‌های داده‌کاوی(مرحله‌ی بعد از َپیش پردازش) و به تبعِ آن، نتایج و تحلیل‌های حاصل داشته باشند.

پیش پردازش داده ها– یکپارچه سازی

- یکپارچگی داده‌ها یک صفت خاص یا یک ویژگی از اطلاعات است که نشان می‌دهد داده‌ها در بین تغییرات و یا رویدادهای فنی سالم و بدون تغییر بمانند. یکپارچگی داده‌ها ارتباط قوی‌ای با سرورها و پایگاه‌داده‌ها دارد، چون سرورها و پایگاه داده‌ها جایی هستند که بیشتر داده‌ها در آن ذخیره می‌شوند.
- موارد مطروحه در بحث یکپارچگی داده ها:
- ۱– ارتباط امن: اطلاعات به درستی و امن از ارسال کننده به دریافت کننده ارسال شوند.
- برای مثال: داده‌های ارسال شده از فرم مشتریان به سمت پایگاه داده ای که اطلاعات سایر مشتریان را در خود نگاه می‌دارد.
- ۲– ذخیره سازی امن : داده‌هایی که در سرورها قرار دارند، تغییر داده یا اصلاح نشده‌اند و از آنها می‌توان برای مقاصد اصلی استفاده کرد.
- ۳- داده‌ها می‌توانند بررسی شوند: به این معنی که در هر مرحله از تغییراتی که روی داده‌ها انجام شده است می‌توان داده‌ها را بررسی کرد. این نکته بسیار مهم است مخصوصا برای سازمانهایی که با داده‌های حساسی در ارتباط هستند.
- تهدید یکپارچگی داده‌ها:
- ۱– خطاهای فنی
- ۲– حمله‌های سایبری و سایر نقص‌های امنیتی

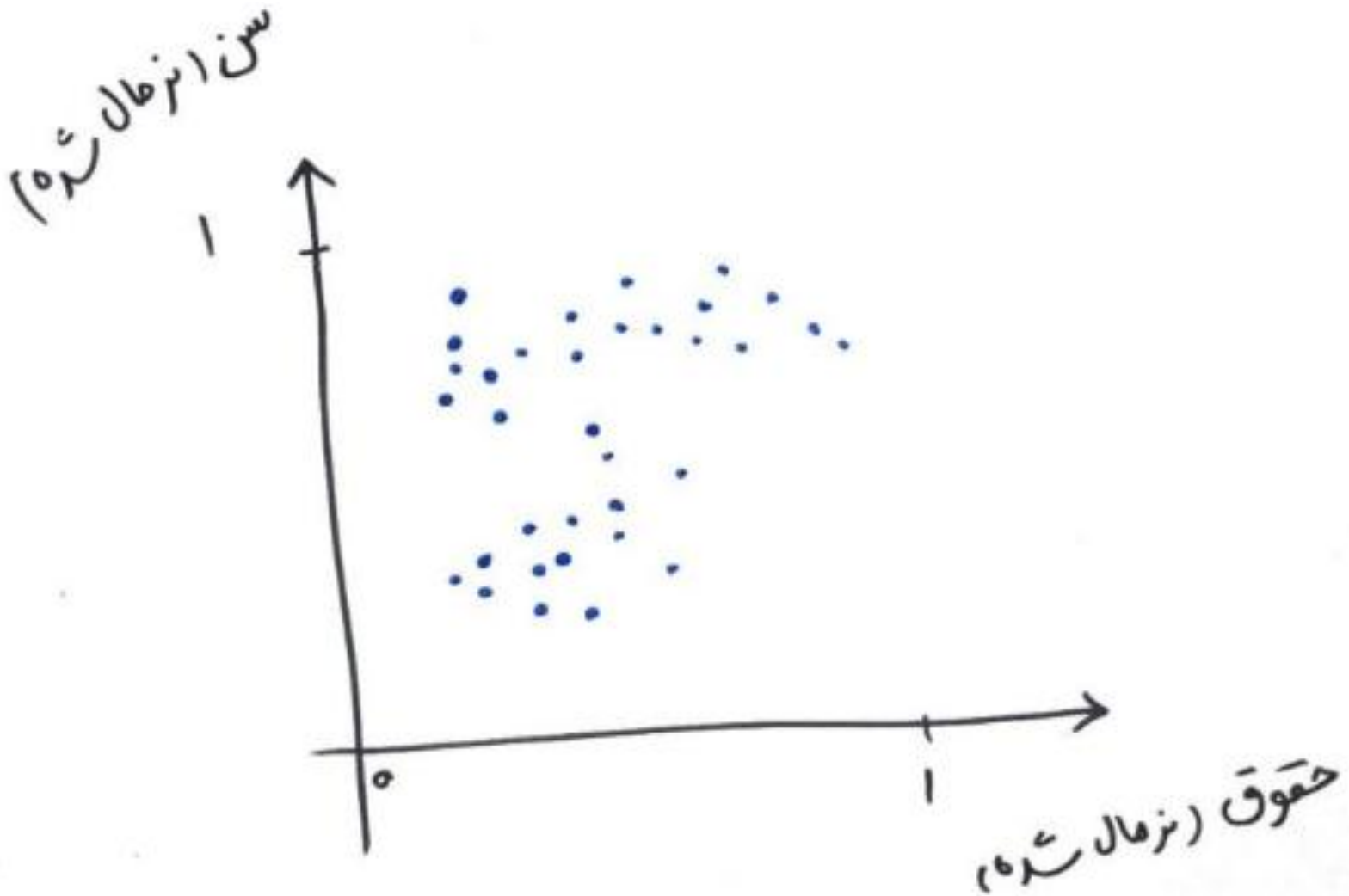
پیش پردازش داده ها– نرمالیزه نمودن



پیش پردازش داده ها– نرمالیزه نمودن

- داده‌ها در ۲بُعد گسترش یافته‌اند: بُعدِ اول(محور عمودی)، سن(که معمولاً بین ۲۰ تا ۹۰سال است) و بُعدِ دوم(محور افقی) حقوقِ ماهیانه که معمولاً بین ۹،۰۰۰،۰۰۰ تا ۱۰۰،۰۰۰،۰۰۰ریال متغیر است.
- ویژگی حقوقِ ماهیانه(محورِ افقی)، تاثیر بسیار زیادی بر روی الگوریتم می‌گذارد(به خاطر اینکه بازه‌ی بزرگ‌تری از اعداد را در بر می‌گیرد و در اصطلاح Scaleبیشتری دارد).
- این یکی از مواقعی است که داده‌ها در بازه‌ی تغییراتِ متفاوت می‌توانند تاثیر غیرِ دلخواهی بر روی همدیگر و به تبعِ آن بر روی الگوریتم، قرار دهند.
- داده‌ها باید در یک بازه‌ی(Rangeمساوی نسبت به یکدیگر قرار بگیرند(مثلاً همه در یک بازه‌ای مانند ۰ تا ۱قرار داشته باشند)
- به این کار نرمال‌سازی داده‌ها یا Data Normalizationگفته می‌شود.

پیش پردازش داده ها– نرمالیزه نمودن



پیش پردازش داده ها– تبدیل داده

- همیشه داده‌ها به صورتِ عددی آماده نیستند و بعضاً نیاز دارند تا به فرمتِ دلخواهِ الگوریتم(یعنی همان فرمتِ ماتریسِ عددی) تبدیل شوند.
- این دست از داده‌ها بایستی قبل از تزریق به الگوریتم، به فُرمتِ مناسب تبدیل(Transform)شوند.

پیش پردازش داده ها– تبدیل داده

	سن	معدل	قد	جنسیت
Student #1	19	18.5	185	مرد
Student #2	21	17.0	162	زن
student #3	22	15.5	177	زن
student #4	20	18	156	مرد
⋮				

پیش پردازش داده ها– تبدیل داده

	سن	معدل	قد	مرد	زن
Student #1	19	18.5	185	1	0
Student #2	21	17.0	162	0	1
Student #3	22	15.5	177	0	1
Student #4	20	18	156	1	0
⋮					

پیش پردازش داده ها– داده های گم شده

- فقدانِ داده‌ها معمولاً یکی از مسائلِ اصلی در حوزه‌ی جمع آوری داده می‌باشَد.
- راه حل ها:
- یکی از راه‌حل‌ها غلبه بر داده‌ی گم شده با پر کردنِ آن توسط داده‌ی واقعی است.
- اول این‌که نمونه های مربوطه را از بین داده‌ها حذف کنیم.
- در واقع حذفِ چند رکورد، عموماً در این معادلات خللی وارد نمی‌کند.
- راه‌حلِ دیگر این است که ستون ویژگی مورد نظر را از بینِ داده‌ها حذف کنیم در بعضَی از مواقع می‌توان میانَگین(Mean) یا میانه‌ی(Median) اعدادِ موجودِ رکوردهای دیگر را محاسبه کرده و به جای مقادیرِ مفقود شده قرارَ داد.

پیش پردازش داده ها– داده های مفقوده

- استفاده از الگوریتم KNN
- این الگوریتم به دنبالِ نزدیک‌ترین همسایه در بین داده‌های موجود می‌گردد و برای حلِ مشکلِ دادَه‌های مفقود شده نیز می‌توان از این روش استفاده کرد.
- می توان ویژگی های مفقوده را بر اساس ویژگی های نمونه های نزدیک به آن تخمین زد.

پیش پردازش داده ها– داده های پرت

- به این دست از داده‌ها که معمولاً با بقیه‌ی داده‌ها ناسازگار هستند داده‌های پرت(Outliers)می‌گویند و مجموعه‌ی داده را دارای اغتشاش یا نویز(Noise)می‌دانند.
- نویزها که به داده‌های غیرطبیعی(Anomalies)نیز شهرت دارند و باعث خراب شدنِ آمارها و داده‌های مجموعه‌ی داده می‌شوند.
- یکی از روش‌های مواجه با این داده های پرت، حذفِ مقادیر بالا و پایین داده‌ها به تعداد مشخص است.
- با این‌کار داده‌ها در یک بازه‌ی مشخص و معقول قرار می‌گیرند.

پیش پردازش داده ها– کاهش داده ها

- برخی از ویژگی ها روی نتیجه مورد نظر از تحلیل اثرگذاری ندارند.
- این ویژگی ها نیز یک ویژگیِ نویز به حساب می‌آید. یعنی برخی اوقات یک ویژگی(بُعد) نیز می‌تواند نویَز باشد به این صورت که در تصمیم‌گیریِ نهایی تاثیرِ چندانی نداشته باشد.
- برای تشخیص ویژگی‌های نویز می توان ویژگی‌هایی با تاثیرِ کم را از میان ویژگی‌های موجود تعیین و حذف نمود.

پیش پردازش داده ها– انتخاب ویژگی

- با بزرگ شدن مجموعه‌ی داده و تعدادِ ویژگی‌ها، نمی‌توان شناسایی ویژگی های با اهمیت کمتر را راحت انجام داد.
- برای همین در این دست از موارد می‌توان از الگوریتم‌های کاهشِ ویژگی استفاده کرد و تعداد ویژگی‌ها یا همان ابعاد را کاهش داد. به این الگوریتم‌های در اصطلاح الگوریتم‌های کاهشِ ابعاد یا Dimensionality Reduction نیز می‌گویند.
- در بعضی از موارد تعدادِ ویژگی‌های مجموعه‌ی داده بسیار زیاد است و الگوریتم‌های داده‌کاوی(مانند طبقه‌بندی یا خوشه‌بندی) در ابعادِ زیاد دچار خطا می‌شوند و یا سرعت انجامِ عملیات در آن‌ها کاهش پیدا می‌کَند.
- همچنین در بعضی از موارد می‌خواهیم با کاهشِ تعداد ابعاد، آن‌ها را در یک نمودار یا چارت رسم کنیم. برای همین بایستی دادَه‌ها را به تعداد ۲ یا ۳ بُعد تبدیل کرده تا قابل نمایش باشند.

ویژگی‌ها (ابعاد یا متغیرها)

- ✓ یک فیلد داده، که نشان دهنده خصوصیت یا ویژگی یک داده است.
- ✓ بعنوان مثال: نام، وزن، رنگ و...

- ✓ انواع ویژگی‌ها
  - ✓ اسمی(کیفی)
  - ✓ باینری
  - ✓ عددی(کمی)
    - ✓ فاصله‌ای
    - ✓ نسبی

36

انواع ویژگی ها

- ✓ اسمی(کیفی)
  - صرفاً نام‌های متفاوتند و فقط اطلاعاتی برای تمایز اشیاء فراهم می‌کنند(کد پستی، جنسیت شماره پرسنلی، رنگ چشم)
  - اطلاعات کافی برای مرتب کردن اشیاء فراهم میکند(رتبه‌ای)({خوب، بهتر، بهترین}، سطح تحصیلات)
- ✓ باینری
  - ویژگی‌های اسمی تنها با دو حالت(۰ یا ۱)
  - باینری‌های متقارن: هر دو خروجی اهمیت یکسان دارند(مانند جنسیت)
  - باینری نامتقارن: خروجی‌ها اهمیت یکسان ندارند(مانند مثبت یا منفی بودن تست پزشکی)
  - عرف: اختصاص ۱ به مهمترین خروجی(مانند مثبت بودن تست HIV)

37

انواع ویژگی

- عددی(کمی)
  - فاصله‌ای(Interval)
    - با مقیاسی از واحدهای مساوی اندازه‌گیری می‌شوند
    - نقطه صفر ذاتی ندارند
    - مقادیر ترتیب دارند(مانند درجه دما به سانتیگراد یا فارنهایت، تاریخ‌های تقویم)
  - نسبی
    - نقطه صفر ذاتی دارند
    - هم تفاوت و هم نسبت با معنی است(۱۰ متر دو برابر ۵ متر است)
    - مانند درجه دما به کلوین، طول، مقادیر پولی

انواع ویژگی

نوع ویژگی		توصیف مقادیر ویژگی	مثال	عملیات
طبقه‌ای (کیفی)	اسمی	صرفاً نامهای متفاوتند و فقط اطلاعاتی برای تمایز اشیاء فراهم می‌کنند (=، ≠).	کد پستی، جنسیت، شماره پرسنلی، رنگ چشم	مُد، آنتروپی، همبستگی توافقی، آزمون کای مربع
	رتبه‌ای	اطلاعات کافی برای مرتب کردن اشیاء فراهم می‌کند (>،<).	{خوب، بهتر، بهترین}، سطح تحصیلات	میانه، دهک، همبستگی رتبه‌ای، آزمونهای ردیف[1]، آزمونهای علامت
عددی (کمی)	فاصله‌ای	تفاوت بین مقادیر با معنی است یعنی واحد اندازه‌گیری وجود دارد (+،−).	تاریخ تقویم، درجه سانتیگراد	میانگین، انحراف معیار، همبستگی پیرسون، آزمون t و F
	نسبتی	هم تفاوت و هم نسبت با معنی است (*، /).	درجه کلوین، مقدار پول، سن، جرم، طول	میانگین هندسی، میانگین موزون، درصد تغییر

انواع ویژگی

تبدل داده‌های کیفی

- ✓ فرض: ویژگی رنگ برای ماشین شامل سه مقدار {قرمز، سفید، مشکی} باشد.
- ✓ برای می‌توانیم این ویژگی را تبدیل به سه ویژگی باینری کنیم:

Red = [1,0,0]
White = [0,1,0]
Black = [0,0,1]

- ✓ چرا برای جلوگیری از افزایش ابعاد مساله آن‌ها را تبدیل به اعداد ۱، ۲ و ۳ نکنیم؟
- ✓ این بدان معنا خواهد بود که بین مقادیر ترتیب خاصی وجود دارد.
- ✓ این ترتیب در تصمیم‌گیری الگوریتم مهم خواهد بود.

انواع ویژگی

اگر ترتیب در مقدار کیفی مهم باشد:

- ✓ **می‌توان با تبدیل مقادیر آن ویژگی ادامه داد.**
- ✓ **مثال: اگر مقادیر کیفیت یک کالا را نشان دهند:**

{بد، متوسط، خوب، عالی}

- ✓ **می‌توان مقادیر آن را با اعداد {۱،۲،۳،۴} جایگزین کرد.**

انواع ویژگی

- **داده‌ها در دنیای واقعی کثیف هستند!**

- ✓ **داده‌های شامل ویژگی‌های پر نشده یا کامل نشده**
- ✓ **داده‌های نویزی: شامل نویز، خطا یا داده‌های پرت**
  - ✓ **میزان درآمد = ۱۰−**
- ✓ **داده‌های متناقض: وجود اختلافاتی در بین داده‌ها**
  - ✓ **سن = ۴۲ تولد = ۱۳۹۵/۲/۵**
  - ✓ **رتبه‌بندی گاهی با (۱،۲،۳) انجام شده گاهی با** a,b,c
- ✓ **عمدی(پر کردن داده‌های گم شده)**
  - ✓ **تولد برای همه ۱ فروردین ثبت شده است**

چگونه با این داده‌ها رفتار کنیم

- ✓ حذفشان کنیم
- ✓ در صورتی که حجم کل داده‌ها زیاد باشد و تعداد این داده‌ها کم
- ✓ به صورت دستی پر کنیم
- ✓ خسته کننده و غیر قابل اجرا
- ✓ بصورت اتوماتیک پر کنیم
- ✓ با یک ثابت سراسری (مانند "مجهول")
- ✓ با مقدار متوسط آن ویژگی پرشود
- ✓ با مقدار متوسط آن ویژگی در بین داده‌های آن کلاس پرشود (هوشمندانه‌تر)
- ✓ با محتمل‌ترین مقدار پرشود(با روش‌های استنتاجی یا رگرسیونی)

فصل دوم

کلاس بندی

کلاس‌بندی(Classification)

کلاس‌بندی برای تخصیص یک برچسب کلاس به مجموعه‌ای از داده‌ها که هنوز کلاس‌بندی نشده‌اند استفاده می‌شود.
بعبارتی داده‌ها براساس ویژگی‌هایشان به دسته‌هایی که نام آن‌ها از قبل مشخص است تخصیص داده می‌شوند.

روش‌های مختلف کلاس‌بندی

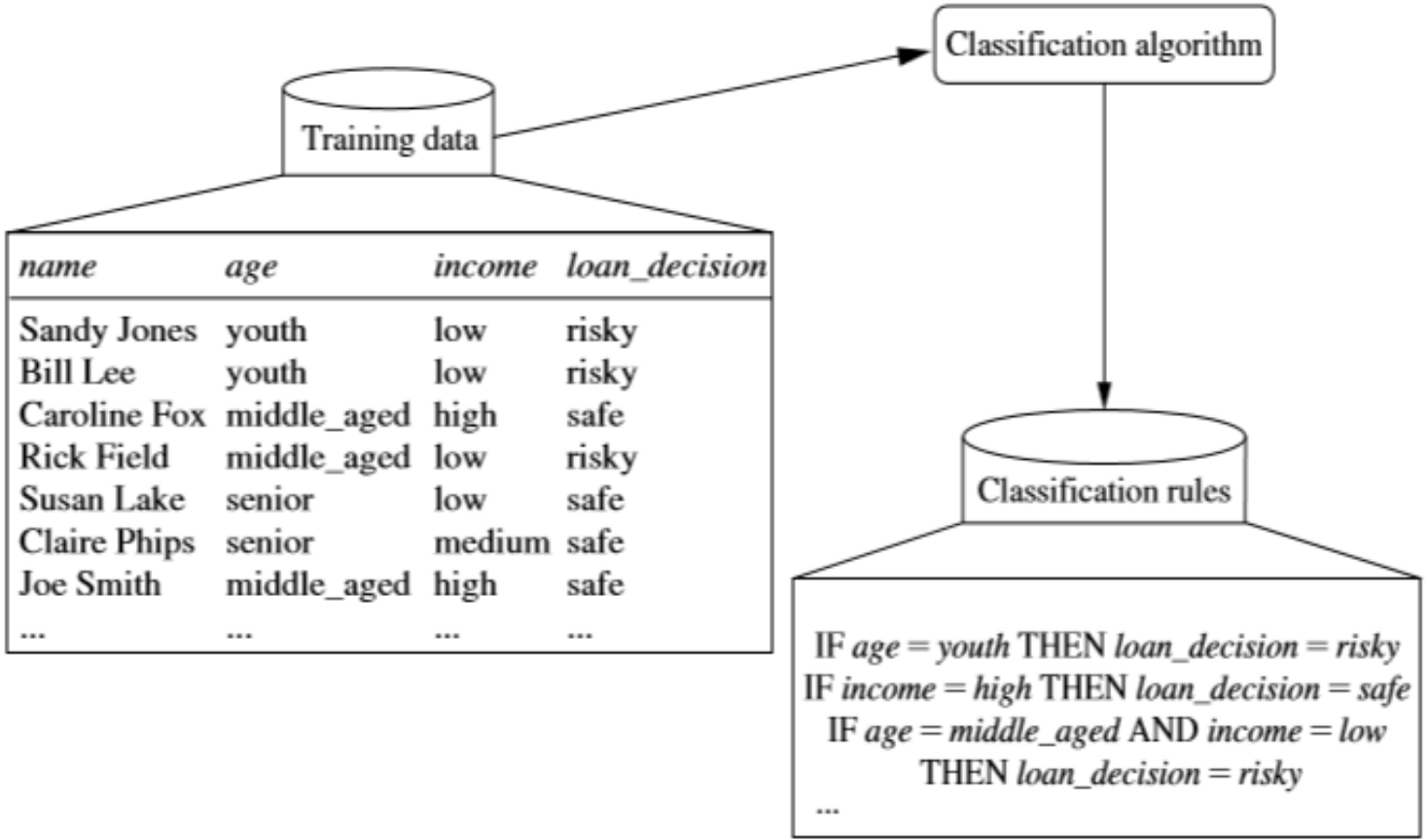
- **درخت تصمیم**
- **شبکه‌های عصبی**
- **بیزین ساده و شبکه‌های بیزین**
- **نزدیکترین همسایگی**

یادگیری تحت نظارت در مقابل یادگیری بدون نظارت

- یادگیری تحت نظارت (کلاس بندی)
  - نظارت: داده های آموزشی (مشاهدات، اندازه گیری ها و ...) به وسیله برچسبها همراه می شوند که مشخص کننده کلاس مشاهدات هستند.
  - داده های جدید براساس مجموعه آموزشی کلاس بندی می شوند.
- یادگیری بدون نظارت (خوشه بندی)
  - برچسب کلاس داده های آموزشی شناخته شده نیست.
  - با داشتن مجموعه ای از اندازه گیری ها، مشاهدات و غیره، هدف ساخت کلاس ها یا خوشه ها در داده ها می باشد.

کلاس بندی – یک فرآیند دو مرحله ای

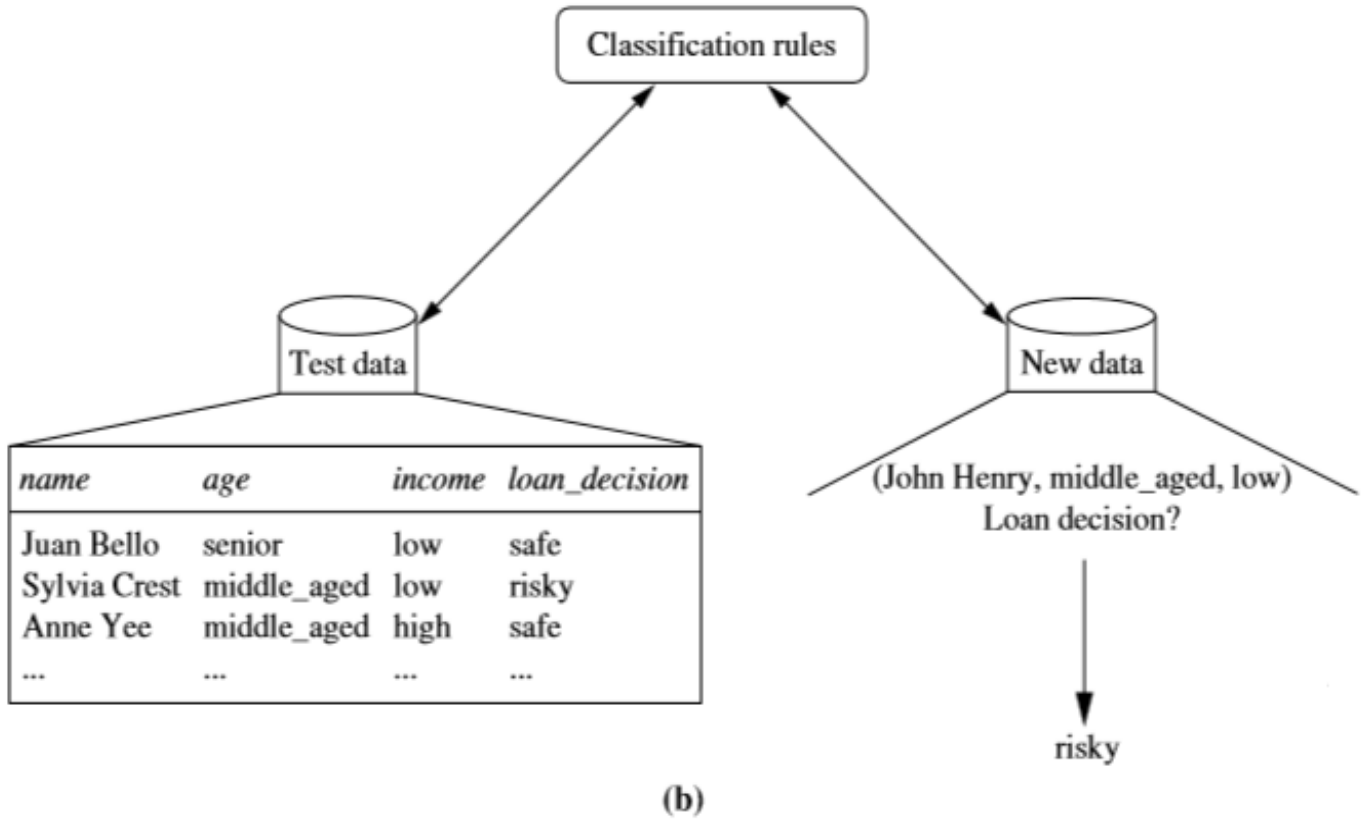
a) **یادگیری: داده های آموزشی توسط الگوریتم کلاس بندی تحلیل می شوند. در اینجا صفت برچسب کلاس** loan_decision **است و مدل آموخته یا** classifier **به فرم قوانین کلاس بندی ارائه شده است.**



name	age	income	loan_decision
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

کلاس بندی – یک فرآیند دو مرحله ای

b) **کلاس بندی:** داده تست برای تخمین دقت قوانین کلاس بندی به کار گرفته شده است. اگر دقت قابل قبول باشد قوانین می توانند برای کلاس بندی به کار گرفته شوند.

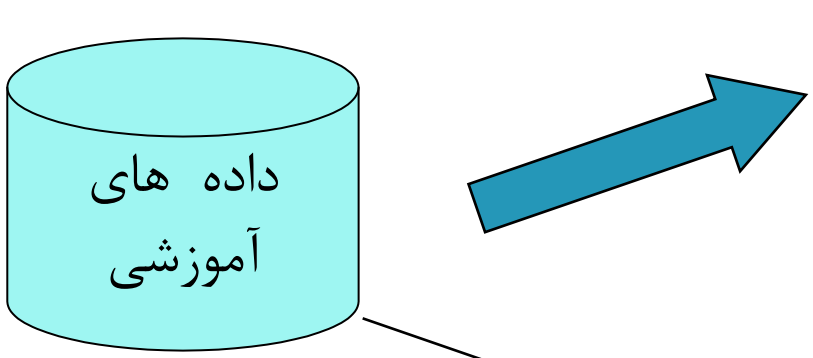


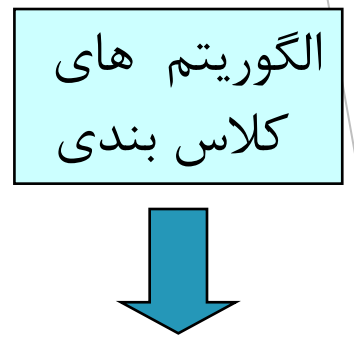
(b)

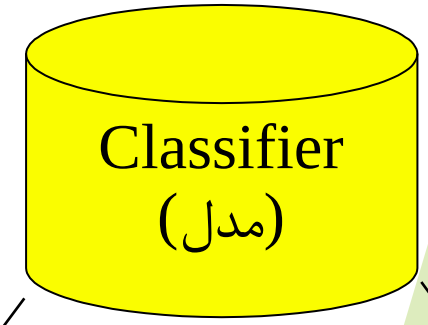
کلاس بندی - یک فرآیند دو مرحله ای

- ساخت مدل: توصیف مجموعه ای از کلاسهای از پیش تعیین شده
  - فرض می شود هر تاپل / نمونه متعلق به یک کلاس از پیش تعریف شده است که به آن صفت برچسب کلاس گویند.
  - مجموعه تاپل هایی که برای ساخت مدل استفاده می شوند را مجموعه آموزشی گویند.
  - مدل به صورت قوانین کلاس بندی، درخت های تصمیم یا فرمول های ریاضی ارائه میشود.
- استفاده از مدل: برای کلاس بندی اشیاء بعدی یا ناشناخته
  - تخمین دقت مدل
    - برچسب شناخته شده از نمونه تست با نتایج کلاس بندی شده ی مدل مقایسه میشوند.
    - نرخ دقت، درصد نمونه های مجموعه تست است که به طور صحیح به وسیله مدل کلاس بندی شده اند.
    - مجموعه تست مستقل از مجموعه آموزشی است(درغیراینصورت منجر به overfitting میشود.)
  - اگر دقت قابل قبول باشد، از مدل برای کلاس بندی داده های جدید استفاده میشود.

مرحله اول: ساخت مدل



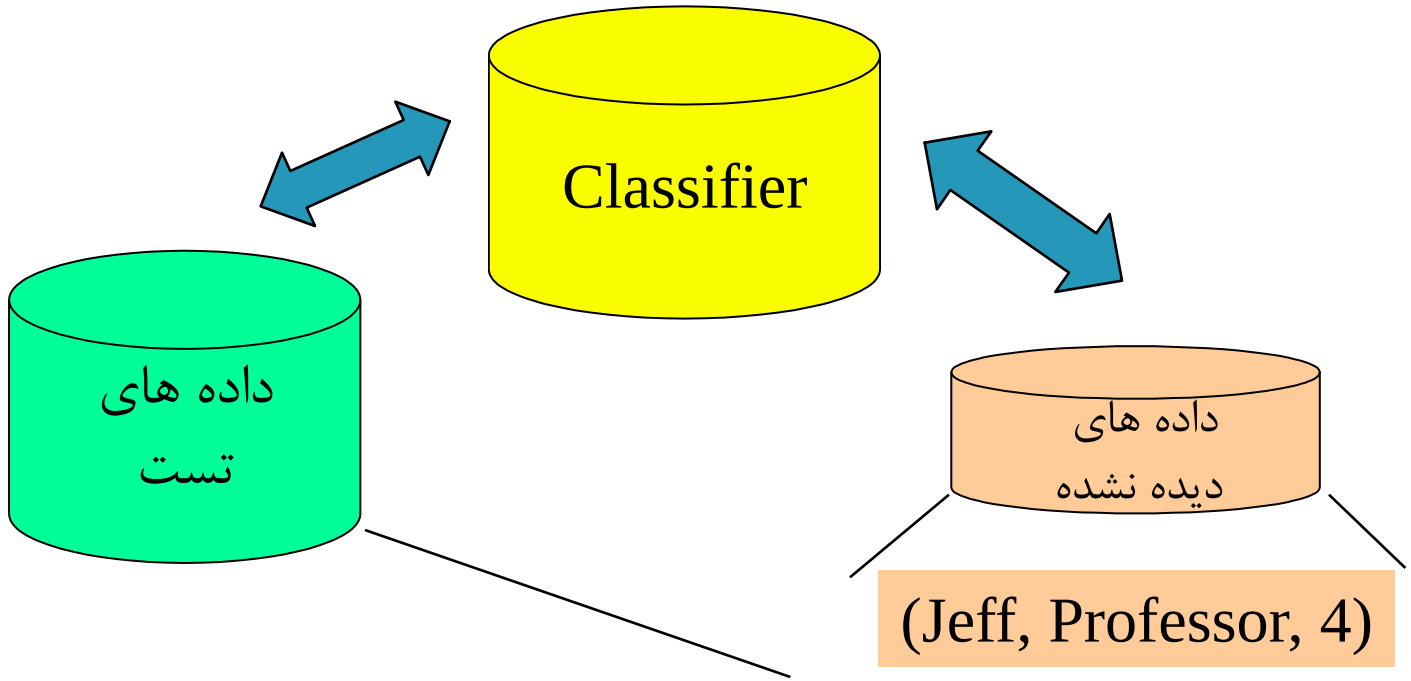




NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

مرحله دوم: استفاده از مدل در پیش بینی



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

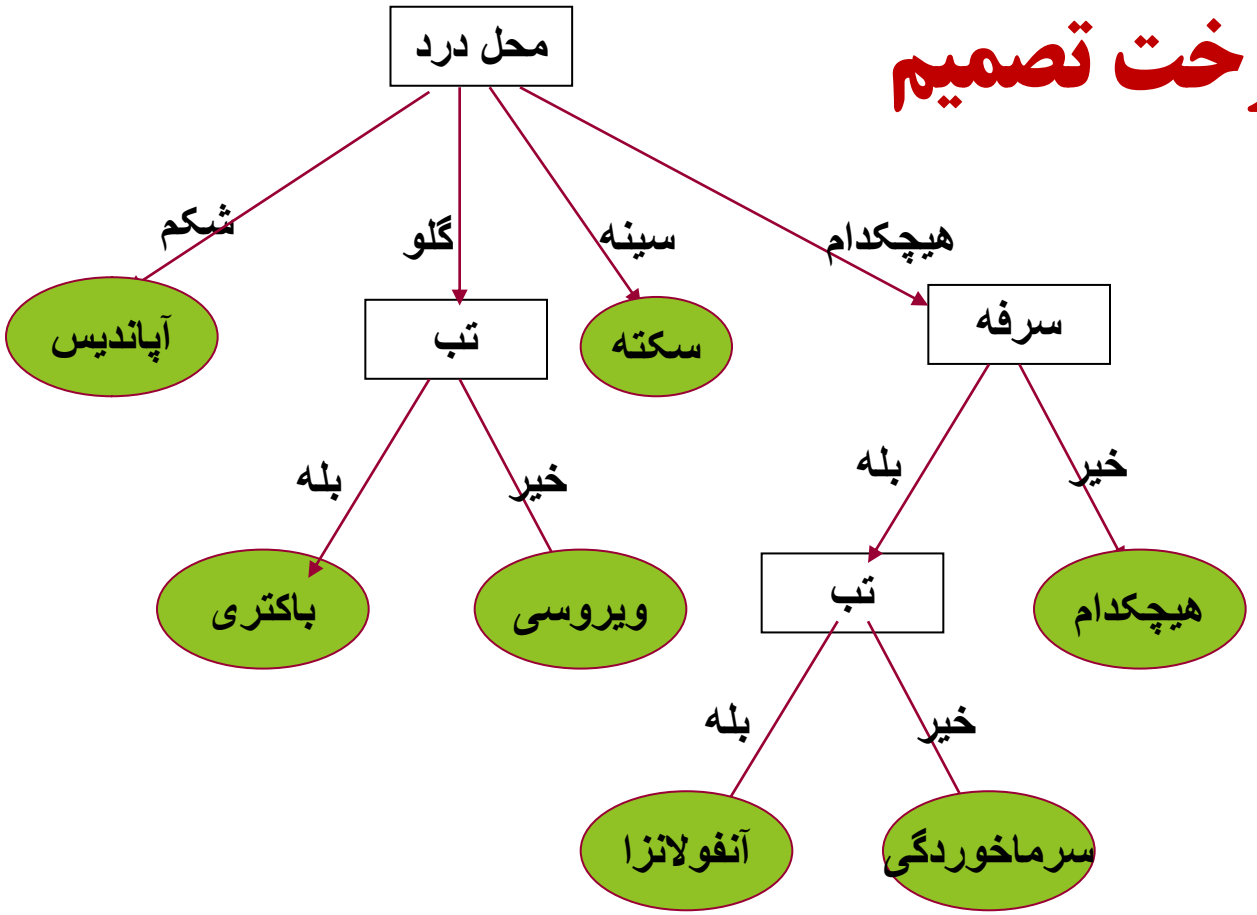
درخت تصمیم

- در یک مسئله یادگیری با دو جنبه مختلف روبرو هستیم:
  - نحوه نمایش فرضیه ها
  - روشی که برای یادگیری برمی گزینیم
- در این فصل برای نمایش فرضیه ها از درخت تصمیم استفاده میکنیم و برای یادگرفتن این درخت از روش ID3 استفاده میکنیم.

نمایش درخت تصمیم

- درخت تصمیم درختی است که در آن نمونه ها را به نحوی دسته بندی میکند که از ریشه به سمت پائین رشد میکنند و در نهایت به گره های برگ میرسد:
  - هر گره داخلی یاغیر برگ(non leaf) با یک ویژگی (attribute) مشخص میشود. این ویژگی سوالی را در رابطه با مثال ورودی مطرح میکند.
  - درهر گره داخلی به تعداد جوابهای ممکن با این سوال شاخه (branch) وجود دارد که هر یک با مقدار آن جواب مشخص میشوند.
  - برگهای این درخت با یک کلاس و یا یک دسته از جوابها مشخص میشوند.

مثالی از یک درخت تصمیم



- هر برگ این درخت یک کلاس یا دسته را مشخص میکند.
- یک مثال آموزشی در درخت تصمیم به این صورت دسته بندی میشود:
- از ریشه درخت شروع میشود.
- ویژگی معین شده توسط این گره تست می گردد.
- و سپس منطبق با ارزش ویژگی در مثال داده شده در طول شاخه ها حرکت رو به پائین انجام می دهد.
- این فرآیند برای گره های زیردرختان گره جدید تکرار می شود.

بایاس درخت تصمیم

بایاس درخت تصمیم بر این ایده است که درختهای کوچکتر بر درختهای بزرگتر ترجیح داده شود.

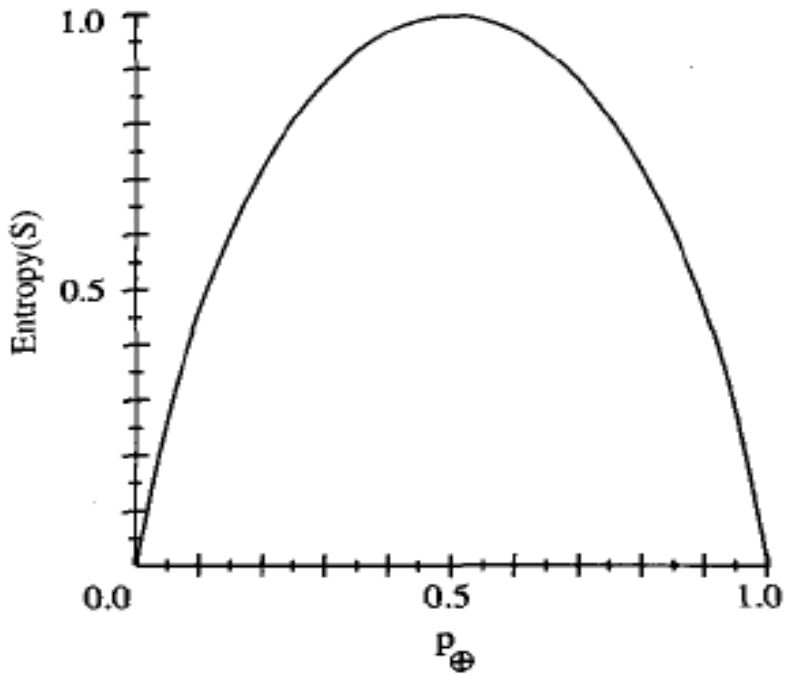
در مسئله ای با تعداد m ویژگی باشد، پس حداکثر ارتفاع درخت m خواهد بود.

چرا که:

درخت تصمیم دارای یک ریشه است که آن خود یک ویژگی است،

در سئوال از آن ویژگی به پاسخی می رسیم که آن خود نیز، ویژگی است.

- **اگر اعضای S نیمی مثبت و نیمی منفی باشد آنتروپی برابر با یک است**



بهره اطلاعات(Information Gain)

- بهره اطلاعات یک ویژگی عبارت است از مقدار کاهش آنتروپی که بواسطه جداسازی مثالها از طریق این ویژگی حاصل میشود.
- بعبارت دیگر بهره اطلاعات Gain(S,A) برای یک ویژگی نظیر A نسبت به مجموعه مثالهای S بصورت زیر تعریف میشود:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- که در آن Values(A) مجموعه همه مقدار ویژگی های A بوده و S_V زیرمجموعه ای از S است که برای آن A دارای مقدار V است.
- در تعریف فوق عبارت اول مقدار آنتروپی داده ها و عبارت دوم مقدار آنتروپی مورد انتظار بعد از جداسازی داده هاست.

در نظر گرفتن ویژگی های با مقادیر پیوسته

- درخت یادگرفته شده توسط ID3 محدود به توابع و ویژگی های با مقدار گسسته است.
- برای اینکه این الگوریتم ویژگی های با مقدار پیوسته را نیز شامل شود، میتوان برای یک ویژگی پیوسته مثل A یک ویژگی بولی مثل Ac تعریف گرد که Ac درست است اگر A<C باشد و در غیر اینصورت نادرست است.
- C باید طوری انتخاب شود که بهره اطلاعات را حداکثر کند. اینکار میتواند با مرتب کردن مقادیر ویژگی A وانتخاب نقاطی که مقادیر مثالهای مجاور تغییر میکنند انجام شود. در چنین حالتی میانگین دومثال مجاور میتواند بعنوان آستانه انتخاب شود.

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

(48 + 60)/2 (80 + 90)/2

معیار نسبت بهره یا gain ratio

- میتوان از معیار دیگری با نام نسبت بهره و یاgain ratio استفاده نمود که خاصیت آن حساسیت داشتن به این است که یک ویژگی با چه گستردگی و یکنواختی داده ها را جدا میکند.
- برای اینکار عبارتی بصورت زیر تعریف میشود:

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- با استفاده از عبارت فوق نسبت بهره بصورت زیر تعریف میشود:

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

معیار نسبت بهره یا gain ratio

- SI باعث میشود تا ویژگی هائی که مقادیر زیادی با توزیع یکنواخت دارند حذف گردند.
- برای مثال یک ویژگی نظیر تاریخ برای تک تک مثالها توزیع یکسانی دارد از اینرو $SI=\log^{n}_{2}$ خواهد شد در حالیکه اگر یک ویژگی مثالها را به دو دسته تقسیم کند SI=1 خواهد شد.
- یک مشکل عملی استفاده ازمعیار نسبت بهره این است که ممکن است مخرج این عبارت صفر و یا خیلی کوچک شود.

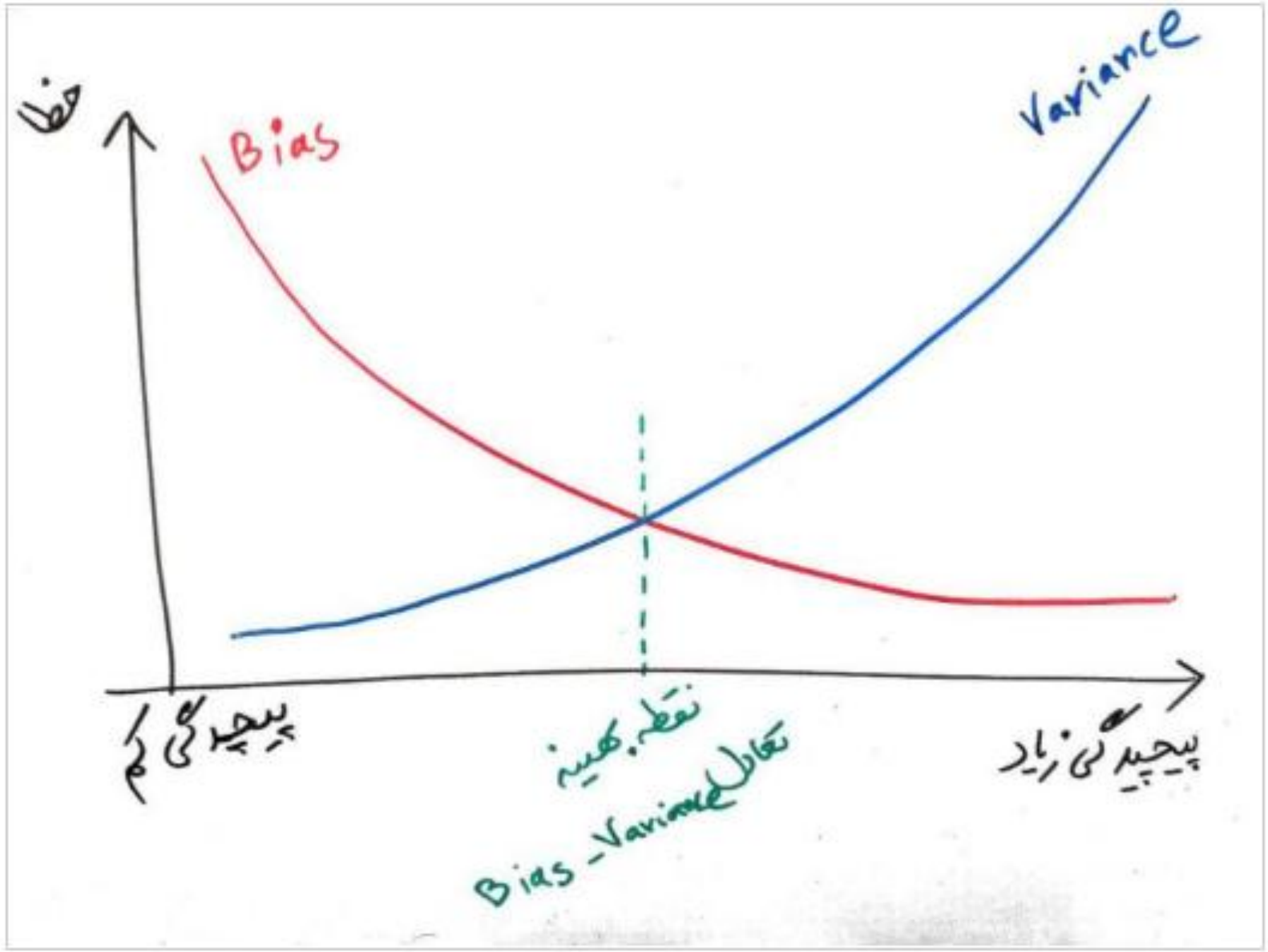
کلاس بندی– بایاس و واریانس

- زمانی که خطِ طبقه بند دارای پیچیدگی(Complexity)زیادی است و معنی این خطِ پیچیده این استَ که طبقه‌بندی که این خط را رسم کرده است بسیار Overfit شَده.
- یعنی کاملا بر روی داده های آموزشیِ موجود درست عمل می کند ولی اگر یک یا چند داده‌ی جدید ببیند، نمی‌تواند به درستیَ این داده‌ی جدید را طبقه‌بندی کند(برچسب بزند).
- در اصطلاح این‌جا Variance بالایی داریم به این معنی که خط طبقه‌بند دارای پیچیدگیِ بالایی می‌باشد و در عینِ حال که می‌تواند طبقه‌بندی راَ بر روی داده‌های آموزشیِ موجود بسیار درست انجاَم دهد، ولی اگر یک داده‌ی جدید را مشاهده کند، ممکن استَ خطا داشته باشد.
- در حالتی که خط طبقه بندی بسیار ساده رسم شده است. این خط ساده بر روی داده های آموزشی(داده‌های موجود) نیز خطا دارد ولی به هر حال نسبت به داده های جدید هم خطای خیلی بالایی نخواهد داشت.
- یعنی هم بر روی داده های آموزشی و هم بر روی داده‌های جدید یک خطای نسبی دارد. در این جا مقدار Bias زیادی داریم یعنی خطِ ساده ما، چه در داده‌های آموزشی و چه در داده‌های جدید یک خطای نسبی بالایی داردِ. در واقع در اینجا Underfitting رخ داده است.

کلاس بندی– بایاس و واریانس

- Biasبالا سادگیِ زیادِ طبقه‌بند را در نتیجه دارد و Varianceبالا نیز پیچیدگیِ بیش از حد را برای ما به همراه دارد.
- باید یک تعادلِ معقولی بین Bias و Varianceیعنی بین سادگیِ زیادِ طبقه‌بند و پیچیدگیِ زیادِ آن برقرار شود تا یک طبقه بند خوب داشته باشیم.
- با زیاد شدن پیچیدگیِ یک مُدل(مثلا یک الگوریتمِ طبقه‌بندی)، Varianceزیاد می شود(خط آبی) و با کم شدن پیچیدگیْ Biasزیاد می رود(خط قرمز). و زیاد شدن هر دو باعث بالا رفتن مقدار خطای کلْ می شود(محور عمودی).
- در نقطه سبز رنگ(که یک پیچیدگی معقول دارد) مقدار Bias و Varianceدر حد معقول و نرمالی قرار میگیرد. از یک طبقه‌بند خوب انتظار می رود که مدلی بسازد که یک پیچیدگی معقول(در نقطه ای نزدیک به نقطه سبز) به دست بیاورد. یعنی نه زیاد پیچیده باشد و نه زیاد ساده.

کلاس بندی– بایاس و واریانس



دو روش برای جلوگیری از بیش برازش وجود دارد:

۱) پیش هرس (Prepruning)
۲) پس هرس (Postpruning)

ابتدا به درخت اجازه داده میشود تا به اندازه کافی رشد کند. سپس گره هائی را که باعث افزایش دقت دسته بندی نمیشوند حرس میگردند

پیش هرس (Prepruning):

- وقتی به میزان تعیین شده از معیاری از صحت دسته بندی رسیدیم از ادامه رشد درخت جلوگیری می‌شود.
- تعیین میزان معیار از قبل دشوار می‌باشد.

پس هرس (Postpruning):

- **هرس کردن درخت پس از رشد کامل**
  - ✓ **داده‌ها به دو مجموعه آموزش و آزمون تقسیم می‌شوند.**
  - ✓ **هر زیر شاخه با یک برگ جایگزین می‌شود. به این برگ، دسته مثال‌های اکثریت، یعنی دسته‌بندی اکثر مثال‌های قرار گرفته تحت این شاخه، نسبت داده می‌شود.**
  - ✓ **عملکرد درخت برروی مثال‌های آزمون، بررسی میشود: اگر درخت هرس شده، عملکرد بهتر و یا مساوی با درخت فعلی داشت، از درخت هرس شده، استفاده می‌شود.**
  - ✓ **هرس کردن آنقدر ادامه می‌یابد تا هرس بیشتر، سودی نداشته باشد.**

دلایل بروز Overfitting

- **الگوریتم ID3 هر شاخه از درخت را آنقدر به عمق میبرد که بتواند بطور کامل مثالهای آموزشی را دسته بندی کند. این امر میتواند منجر به Overfitting شود. دلایل بروز overfitting عبارتند از:**
- **وجود نویز در داده های آموزشی**
- **تعداد کم مثالهای آموزشی**
  - **برای مثال اگر فقط دو بار پرتاب سکه داشته باشیم و هر دو بار شیر آمده باشد چه نتیجه ای در مورد این آزمایش میتوان گرفت؟**

Overfitting

پدیده overfitting **منحصر به درخت های تصمیم نیست و سایر روشهای یادگیری ماشینی نیز با آن مواجه هستند. این پدیده غالبا وقتی اتفاق می افتد که:**

- the hypothesis space is very large
- the hypothesis search is not biased toward simple models
- there is little training data
- there is a lot of noise in the training data

در عمل با دیدن شرایط زیر میتوانیم بگوئیم که overfitting **رخ داده است:**

- **اختلاف زیاد بین دقت دسته بندی داده های آموزشی و داده های تست**
- **رسیدن به فرضیه و یا مدلهای خیلی پیچیده (مثلا رسیدن به یک درخت تصمیم خیلی بزرگ)**

پرهیز از Overfitting

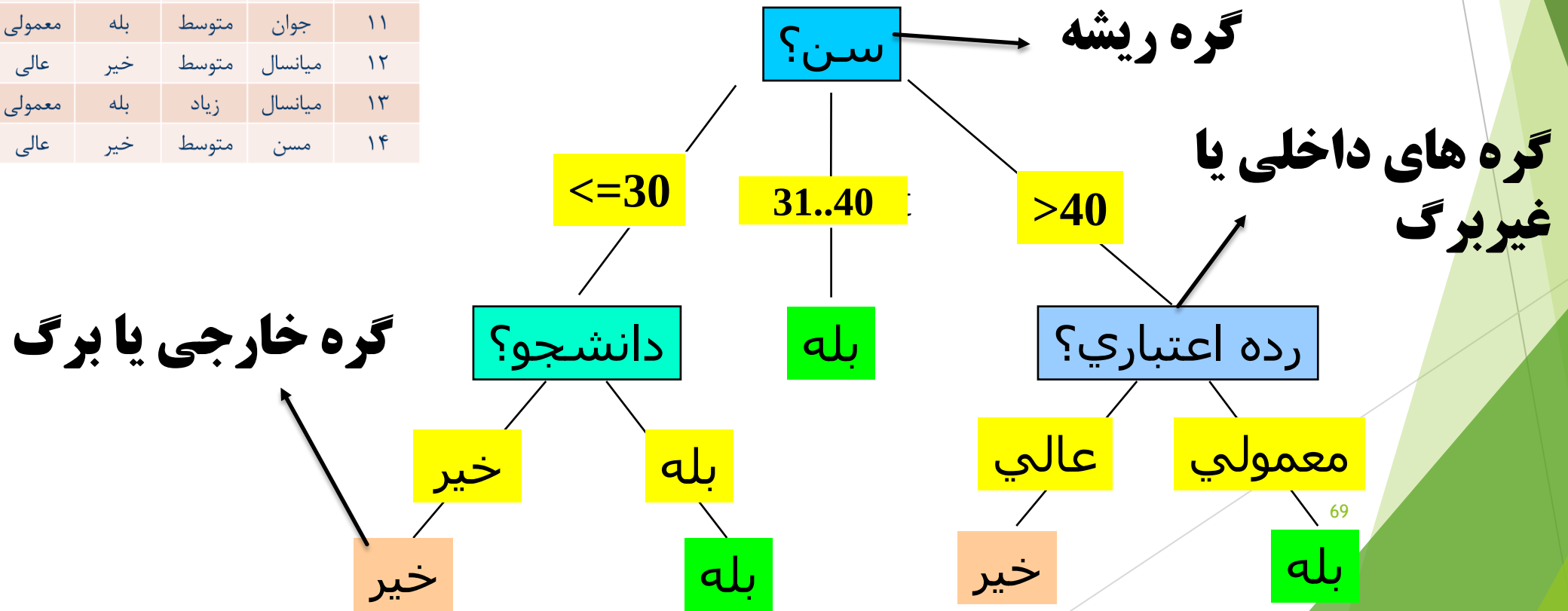
1. **جلوگیری از رشد درخت قبل از رسیدن به مرحله ای که بطور کامل داده های آموزشی را دسته بندی نماید.**
2. **اجازه به رشد کامل درخت و سپس حرس کردن شاخه هائی که مفید نیستند.** (post pruning)

در عمل روش دوم بیشتر استفاده شده است زیرا تخمین اندازه صحیح درخت کار ساده ای نیست.

ساخت درخت تصمیم

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

- آیا این مشتری کامپیوتر خریداری می‌کند؟



الگوریتم های درخت تصمیم

الگوریتم‌های درخت تصمیم مختلفی وجود دارند

C4.5	CART	ID3	CHAID	....

- ✓ **یکی از تفاوت‌های این الگوریتم‌ها سنجه یا معیار انتخاب خصوصیت جداساز است.**
- ✓ **معیارهای مانند بهره اطلاعاتی**(Information Gain)**،** **نسبت بهره**(Gain Ratio)**، شاخص جینی** (Gini Index)**و ...**
- ✓ این الگوریتم‌ها در درجه اول از حیث راهی که

شاخص های ساخت درخت تصمیم

شاخص جینی (Gini Index):

برای اندازه‌گیری تنوع جمعیت استفاده می‌شود. همان مفهوم می‌تواند برای تعیین خلوص یک کلاس خاص به عنوان نتیجه یک تصمیم برای شاخه زدن در طول یک صفت یا متغیر خاص استفاده شود. بهترین تقسیم آن است که خلوص مجموعه‌های ناشی از تقسیم پیشنهادی را افزایش می‌دهد.

بهره اطلاعات (Gain Ratio):

مکانیسم تقسیم مورد استفاده در روش ID3 است، که شاید به طور گسترده الگوریتم درخت تصمیم‌گیری شناخته شده است. این روش در سال ۱۹۶۶ توسعه داده شد و از آن پس او این الگوریتم را به الگوریتم C4.5 و C5تکامل داد. ایده پایه ID3 این است که یک مفهوم به نام آنتروپی در محل شاخص جینی استفاده شود.

آنتروپی (Entropy):

میزان عدم قطعیت و یا اتفاقی در یک مجموعه داده را اندازه می‌گیرد. اگر تمام داده‌ها در یک زیر مجموعه متعلق به فقط یک کلاس باشد، هیچ عدم قطعیت و یا اتفاقی در مجموعه داده وجود ندارد. بنابراین آنتروپی صفر است. هدف از این روش این است که برای ساخت درختان زیرمجموعه به طوری باشد که آنتروپی هر زیر مجموعه آخر صفر باشد (یا نزدیک به صفر). این محاسبات بارها و بارها برای هر صفت تکرار می‌شوند، و آن یکی که بالاترین بهره اطلاعات را دارد، به عنوان ویژگی تقسیم انتخاب شده است.

تحلیل شاخص ها

هر سه شاخص، عملکرد نسبتاً مطلوبی دارند، در عین حال:

- ✓ **بهره اطلاعات** (Information Gain)**:**
  - **به سمت ویژگی‌هایی با چندین مقدار، بایاس دارد.**

- ✓ **نسبت بهره** (Gain Ratio)**:**
  - **گرایش بسوی انجام انشعاباتی دارد که طی آنها، یک بخش، بسیار کوچکتر از بقیه باشد.**

- ✓ **شاخص جینی** (Gini Index)**:**
  - **به طرف ویژگیهای چند مقداره، بایاس دارد.**
  - **منجر به تولید بخشبندی‌هایی با اندازه نسبتاً برابر دارد که هر کدام، نسبتاً خالص باشند.**

مثالی برای ساخت درخت تصمیم

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

زیاد

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

– نمی‌توان بـا قطعیـت برچسبی برای نمونه های افتاده در این گره انتخاب کرد.

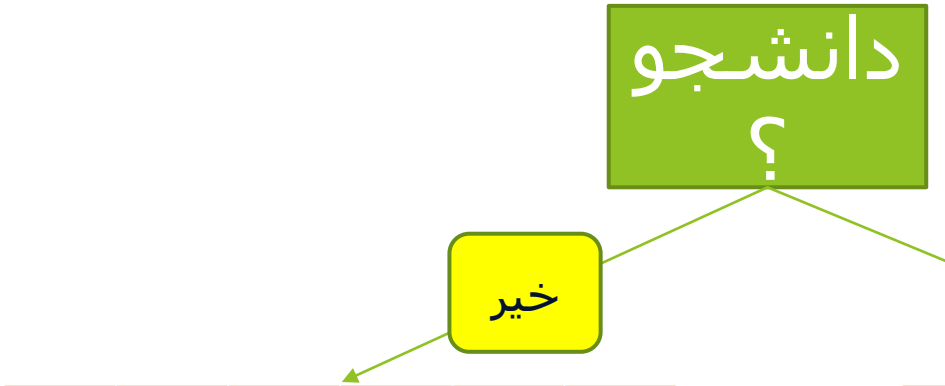
متوسط

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

کم

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۹	جوان	کم	بله	معمولی	بله

– احتمال خرید رایانه در این گروه بالا است
– ما به دنبال قطعیت در برچسب یک گره جدید هستیم



شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

جوان

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله

میانسال

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۳	میانسال	زیاد	خیر	معمولی	بله
۷	میانسال	کم	بله	عالی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

مسن

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

این یک گره ایدآل است

آنتروپی

میزان عدم قطعیت یا تصادفی بودن در یک داده را نشان می‌دهد

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

ما به دنبال خصوصیتی هستیم که با جداسازی داده‌ها به کمک آن به قطعیت بیشتر در برچسب گره‌های جدید برسیم

آنتروپی بیشتر ⟵ عدم قطعیت بیشتر

آنتروپی کمتر ⟵ عدم قطعیت کمتر

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0– 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6)– (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6)– (4/6) $\log_2$ (4/6) = 0.92

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

اطلاعات مورد انتظار(آنتروپی)

جوان

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله

$$-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}=0.97$$

میانسال

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۳	میانسال	زیاد	خیر	معمولی	بله
۷	میانسال	کم	بله	عالی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

$$-\frac{4}{4}\log_2\frac{4}{4}=0$$

مسن

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

$$-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}=0.97$$

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

$$-\frac{1}{7}\log_2\frac{1}{7}--\frac{6}{7}\log_2\frac{6}{7}=0.591$$

$$-\frac{4}{7}\log_2\frac{4}{7}--\frac{3}{7}\log_2\frac{3}{7}=0.985$$

جمع بندی برای یک انشعاب

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

اطلاعات مورد انتظار پس از انشعاب با ویژگی A

انشعاب بر اساس سن:

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۳	میانسال	زیاد	خیر	معمولی	بله
۷	میانسال	کم	بله	عالی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

$$\frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) = 0.694$$

$$\frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) = 0.694$$

نسبت بهره Gain Ratio

- تعمیم یافته بهره اطلاعاتی است.
- شاخص بهره اطلاعاتی، بطرف ویژگیهای با تعداد زیادی از مقادیر، بایاس دارد.
- اگر شماره ردیف را یک ویژگی در نظر می‌گرفتیم، انتخاب بهره اطلاعاتی شماره ردیف بود.
- درخت C4.5 از شاخص نسبت بهره برای انتخاب ویژگی انشعاب، استفاده می‌کند.

$$GainRation(A) = Gain(A) / SplitInfo(A)$$

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

Gain(**سن**) = 0.246

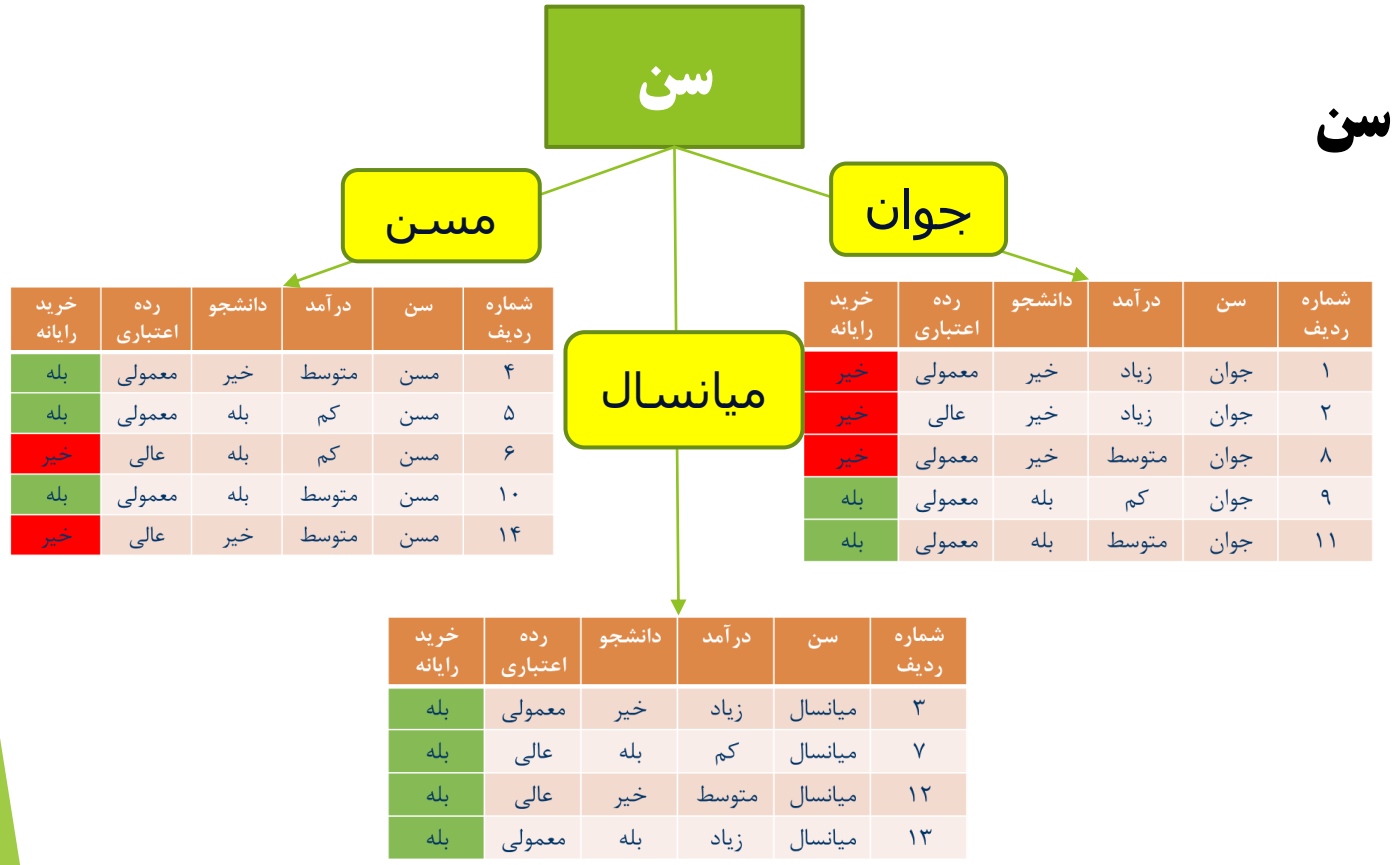
بیشترین مقدار اطلاعات بدست آمده

Gain(**درآمد**) = 0.029

Gain(**دانشجو**) = 0.151

Gain(**رده اعتباری**) = 0.048

مرحله اول تفکیک بر اساس سن انجام می‌شود



شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۳	میانسال	زیاد	خیر	معمولی	بله
۷	میانسال	کم	بله	عالی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

کم

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۹	جوان	کم	بله	معمولی	بله

زیاد

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله

متوسط

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۴	مسن	متوسط	خیر	معمولی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

$$\text{Gain(درآمد)} = 0.029$$

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

$$= 0.926$$

$$\text{GainRatio(income)} = \frac{0.029}{0.926} = 0.031$$

اطلاعات بدست آمده(Information Gained) پس از تفکیک با ویژگی A

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

اطلاعات مورد انتظار در گره بالایی

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

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

$$Gain(\text{سن}) = 0.940 - 0.694 = 0.246$$

شاخص جینی (Gini Index)

– اگر مجموعه داده D شامل نمونه‌هایی از n کلاس مختلف باشد:

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

– که در آن p_j برابر با فرکانس نسبی عناصر کلاس j در D است.

– اگر ویژگی A داده را به دو بخش D_1 و D_2 تفکیک کند، مقدار جینی D عبارتست از:

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

ویژگی با کمینه شاخص جینی (بیشینه میزان کاهش ناخالصی)، جهت تفکیک، انتخاب می‌شود.

شماره ردیف	سن	درآمد	دانشجو	رده اعتباری	خرید رایانه
۱	جوان	زیاد	خیر	معمولی	خیر
۲	جوان	زیاد	خیر	عالی	خیر
۳	میانسال	زیاد	خیر	معمولی	بله
۴	مسن	متوسط	خیر	معمولی	بله
۵	مسن	کم	بله	معمولی	بله
۶	مسن	کم	بله	عالی	خیر
۷	میانسال	کم	بله	عالی	بله
۸	جوان	متوسط	خیر	معمولی	خیر
۹	جوان	کم	بله	معمولی	بله
۱۰	مسن	متوسط	بله	معمولی	بله
۱۱	جوان	متوسط	بله	معمولی	بله
۱۲	میانسال	متوسط	خیر	عالی	بله
۱۳	میانسال	زیاد	بله	معمولی	بله
۱۴	مسن	متوسط	خیر	عالی	خیر

در کل داده‌ها ۹ خیر و ۵ بله وجود دارد:

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

فرض کنید ویژگی درآمد داده را به دو بخش شامل ۱۰ نمونه {کم، متوسط} D_1 : و ۴ نمونه شامل {زیاد} : D_2 تفکیک شده است.

$$gini_{income \in \{low, medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2) = 0.443$$

$$Gini(D_1) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2 = 0.42 \quad Gini(D_2) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

به همین ترتیب:

Gini{low,high} = 0.458; Gini{medium,high} = 0.450

- ✓ **با توجه به مینیمم بودن مقدار شاخص جینی برای تفکیک به دو گروه {کم، متوسط} و {زیاد}، این تقسیم بندی برای ادامه درخت انتخاب خواهد شد.**

الگوریتم درخت تصمیم ID3 و ساختار Entropy و Gain

- الگوریتم ID3 یکی از الگوریتم‌های پایه جهتِ ساختِ درخت‌های تصمیم است. در یک درخت تصمیم، مهم است که کدام یک از ویژگی‌ها(یا همان ابعاد) را در سطوح بالاتری از درخت انتخاب کنیم تا به طبقه‌بندی کمک کند.
- می توان فقط با انتخاب درستِ یک ویژگی(بُعد) درخت تصمیمی ساخت که فقط یک سطح داشته باشد(به جای دو سَطح) و خیلی ساده‌تر تصمیم بگیرد. برای شناسایی این ویژگیِ های بهینه نیاز داریم با مفاهیم Entropy و Gain آشنا شویم.
- Entropy در واقع نشان دهنده کم بودن اطلاعات است. یعنی در مجموعه‌ی داده‌ی موردنظر، از روی یک ویژگی(بُعد) چقدر می‌توانید تشخیص دهید که کلاسِ نهایی چیست. هر اندازه که یک ویژگی دارای مجموع Entropy بالاتری باشد، حاوی اطلاعات کمتری است.

الگوریتم درخت تصمیم ID3 و ساختار Entropy و Gain

- هر اندازه که Entropy کمتر (فضا را محدودتر کرده) باشد اطلاعات بیشتری به ما داده می شود. چرا که در یک محدوده‌ی مشخص‌تر قرار دارد.
- Gain که در واقع همان Information Gain می باشد، از Entropy هر مقدار از ویژگی‌ها کمک گرفته و به میزانِ اطلاعاتی که می‌توان از یک ویژگی (بُعد) به دست آورد، گفته می شود. یعنی یک ویژگیِ خاص چقَدر می‌تواند اطلاعاتِ زیادتری به ما بدهد.
- الگوریتم ID3 در واقع وظیفه پیدا کردنِ ویژگی‌هایی دارای اطلاعات زیادتر (Gain بیشتر) را دارد و آن‌ها را باید در سطوحِ بالاتری از درختَ قرار دهد. هر بار که یک ویژگی در سطحی از درخت انتخاب شد، زیرْ درخت‌های آنَ نیز دقیقا به همان صورت (ویژگی‌هایی با اطلاعات بالا) انتخاب می‌شوند و در سطوح و گره‌های بعدی قرار می‌گیرند. البته وقتی یک گره از درخت انتخاب شد، برای ساختِ زیر درخت‌های دیگر، مجموعه داده‌ها بر اساس مقدارِ گره‌ی انتخاب شده در بالا، کوچکتر می‌شوندَ و هر چه در درخت پایین‌تر می‌رویم (به برگ‌ها نزدیک‌تَر می‌شویم)، مجموعه داده‌ها برای محاسبه‌ی مقدار اطلاعاتْ کمتر می‌شود

مثالی از آنتروپی و Gain در ID3

Example 1

If S is a collection of 14 examples with 9 YES and 5 NO examples then

Entropy(S) = - (9/14) Log2 (9/14) - (5/14) Log2 (5/14) = 0.940

Notice entropy is 0 if all members of S belong to the same class (the data is perfectly classified). The range of entropy is 0 ("perfectly classified") to 1 ("totally random").

Gain(S, A) is information gain of example set S on attribute A is defined as

$$Gain(S, A) = Entropy(S) - \Sigma ((|S_v| / |S|) * Entropy(S_v))$$

Where:

Σ is each value v of all possible values of attribute A

S_v = subset of S for which attribute A has value v

$|S_v|$ = number of elements in S_v

$|S|$ = number of elements in S

مثالی از آنتروپی و Gain در ID3

Example 2

Suppose S is a set of 14 examples in which one of the attributes is wind speed. The values of Wind can be *Weak* or *Strong*. The classification of these 14 examples are 9 YES and 5 NO. For attribute Wind, suppose there are 8 occurrences of Wind = Weak and 6 occurrences of Wind = Strong. For Wind = Weak, 6 of the examples are YES and 2 are NO. For Wind = Strong, 3 are YES and 3 are NO. Therefore

$$Gain(S,Wind)=Entropy(S)-(8/14)*Entropy(S_{weak})-(6/14)*Entropy(S_{strong})$$

$$= 0.940 - (8/14)*0.811 - (6/14)*1.00$$

$$= 0.048$$

$$Entropy(S_{weak}) = - (6/8)*log2(6/8) - (2/8)*log2(2/8) = 0.811$$

$$Entropy(S_{strong}) = - (3/6)*log2(3/6) - (3/6)*log2(3/6) = 1.00$$

For each attribute, the gain is calculated and the highest gain is used in the decision node.

مثالی از آنتروپی و Gain در ID3

Example of ID3

Suppose we want ID3 to decide whether the weather is amenable to playing baseball. Over the course of 2 weeks, data is collected to help ID3 build a decision tree (see table 1).

The target classification is "should we play baseball?" which can be yes or no.

The weather attributes are outlook, temperature, humidity, and wind speed. They can have the following values:

outlook = { sunny, overcast, rain }

temperature = {hot, mild, cool }

humidity = { high, normal }

wind = {weak, strong }

مثالی از آنتروپی و Gain در ID3

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

مثالی از آنتروپی و Gain در ID3

We need to find which attribute will be the root node in our decision tree. The gain is calculated for all four attributes:
Gain(S, Outlook) = 0.246
Gain(S, Temperature) = 0.029
Gain(S, Humidity) = 0.151
Gain(S, Wind) = 0.048 (calculated in example 2)
Outlook attribute has the highest gain, therefore it is used as the decision attribute in the root node.

مثالی از آنتروپی و Gain در ID3

Since Outlook has three possible values, the root node has three branches (sunny, overcast, rain). The next question is "what attribute should be tested at the Sunny branch node?" Since we=92ve used Outlook at the root, we only decide on the remaining three attributes: Humidity, Temperature, or Wind.

S_{sunny} = {D1, D2, D8, D9, D11} = 5 examples from table 1 with outlook = sunny

Gain(S_{sunny}, Humidity) = 0.970

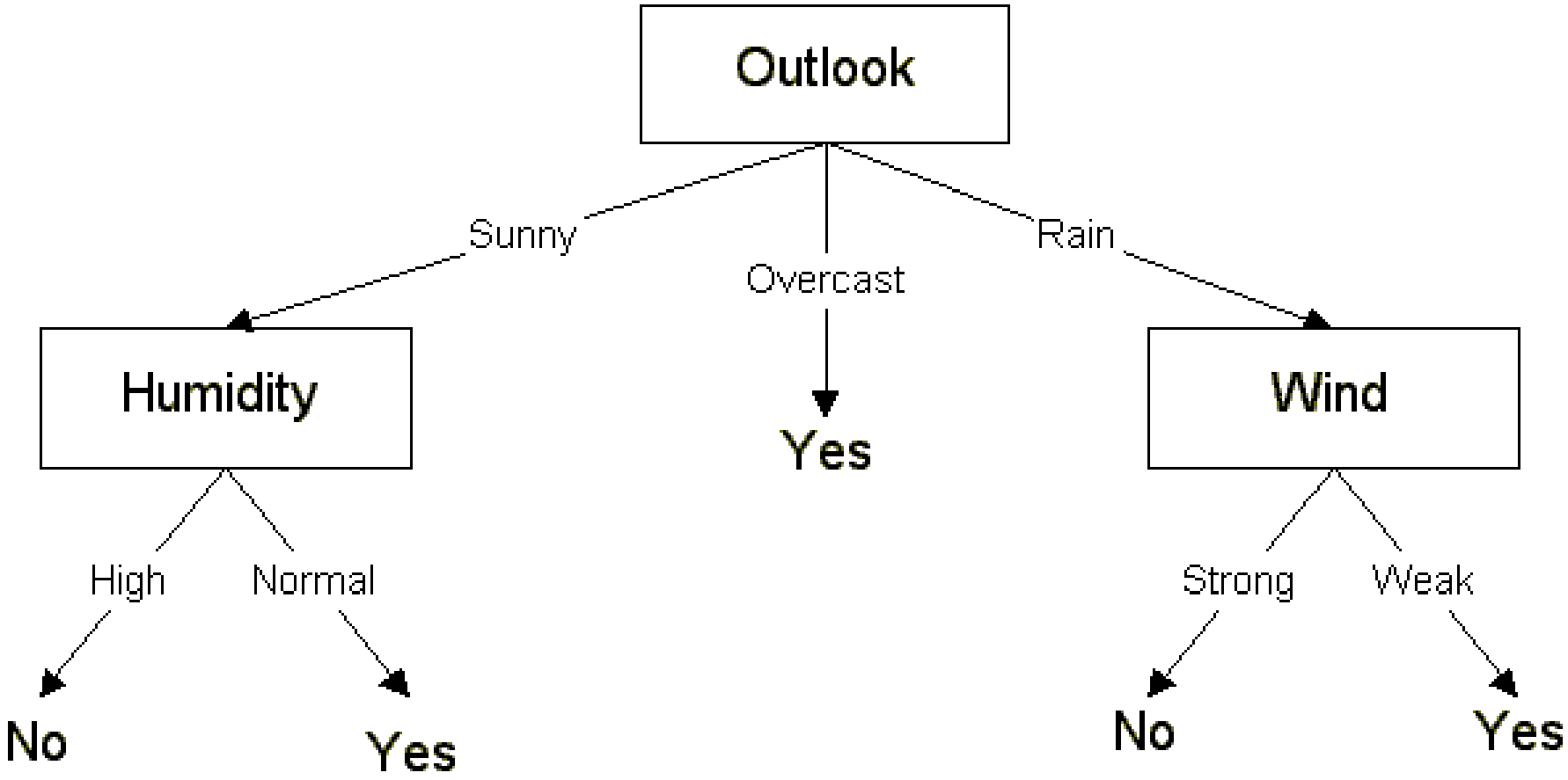
Gain(S_{sunny}, Temperature) = 0.570

Gain(S_{sunny}, Wind) = 0.019

Humidity has the highest gain; therefore, it is used as the decision node.

This process goes on until all data is classified perfectly or we run out of attributes.

مثالی از آنتروپی و Gain در ID3



مثالی از آنتروپی و Gain در ID3

The final decision = tree
The decision tree can also be expressed in rule format:
IF outlook = sunny AND humidity = high THEN playball = no
IF outlook = rain AND humidity = high THEN playball = no
IF outlook = rain AND wind = strong THEN playball = yes
IF outlook = overcast THEN playball = yes
IF outlook = rain AND wind = weak THEN playball = yes
ID3 has been incorporated in a number of commercial rule-induction packages. Some specific applications include medical diagnosis, credit risk assessment of loan applications, equipment malfunctions by their cause, classification of soybean diseases, and web search classification.

کلاس بندی– C4.5

- از نقاط ضعف الگوریتم ID3 که در C4.5 رفع شده است می‌توان به موارد زیر اشاره کَرد:
- ۱. الگوریتم C4.5 میتواند مقادیر گسسته یا پیوسته را در ویژگی‌ها درک کند. الگوریتمِ ID3نمی‌تواند تفاوتِ مقادیرِ عددیِ پیوسته را درک کند. برای مثال نمی‌تواندَ تفاوت بین معدل‌ها رَا درک کند. ولَی الگوریتمِ C4.5می‌تواند این کار را انجام دهد و مقادیرِ پیوسته را هم درک کرده و بر اساسَ آن درخت تصمیم را بسازد.
- الگوریتمِ ID3 نمی‌تواند درختی را با مقادیر پیوسته بسازد زیرا ساختِ این درخت نیازمند اَین است که الگوریتم بتواند ویژگی ها را درست شناسایی نمودَه و بر اساس آَن شاخه‌های زیر درخت‌های چپ و راست را بسازد. ولی این کار از توسطِ الگوریتم C4.5 قابل انجام است.

کلاس بندی– C4.5

- در برخی موارد، تعدادی از داده‌ها وجود ندارند. الگوریتم C4.5 می‌تواند این مقادیر را مدیریت نموده و با وجود مقادیری که ناموجود است، درخت تصمیم خود را بسازد. در حالی که الگوریتمی مانند ID3 و بسیاری دیگر از الگوریتم‌های طبقه‌بندی نمی‌توانند با وجود مقادیر ناموجود، مدلِ خود را بسازند.
- مورد دیگری که باعث بهینه شدن الگوریتم C4.5(نسبت به ID3) می‌شود، عملیاتِ هرس کردن(Prunning)جهت جلوگیری از بیش برازش می‌باشد. الگوریتم‌هایی مانند ID3به خاطر اینکه سعی دارند تا حد امکان شاخه و برگ داشته باشند(تا به نتیجه مورد نظر برسند) با احتمال بالاتری دارای پیچیدگی در ساخت مدل می‌شوند و این پیچیدگی در بسیاری از موارد الگوریتم را دچار Overfittingو خطای بالا می‌کند. اما با عملیات هرس کردن درخت که در الگوریتم C4.5انجام می‌شود، می‌توان مدل را به یک نقطه بهینه رساند که زیاد پیچیده نباشد(و البته زیاد هم ساده نباشد) و Overfitting یا Underfitting رخ ندهد.
- مورد بعدی بحثِ وزن‌دهی(Weighting)به ویژگی‌ها است.

مثالی از C4.5

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

مثالی از C4.5

We will do what we have done in ID3 example. Firstly, we need to calculate global entropy. There are 14 examples; 9 instances refer to yes decision, and 5 instances refer to no decision.

Entropy(Decision) = $\sum$ – p(I) . $\log_2$p(I) = – p(Yes) . $\log_2$p(Yes) – p(No) . $\log_2$p(No) = – (9/14) . $\log_2$(9/14) – (5/14) . $\log_2$(5/14) = 0.940

In ID3 algorithm, we've calculated gains for each attribute. Here, we need to calculate gain ratios instead of gains.

GainRatio(A) = Gain(A) / SplitInfo(A)

SplitInfo(A) = -$\sum$ |Dj|/|D| x $\log_2$|Dj|/|D|

مثالی از C4.5

Wind Attribute

Wind is a nominal attribute. Its possible values are weak and strong.

Gain(Decision, Wind) = Entropy(Decision) – ∑ (p(Decision|Wind) . Entropy(Decision|Wind))

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)] + [p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)]

There are 8 weak wind instances. 2 of them are concluded as no, 6 of them are concluded as yes.

Entropy(Decision|Wind=Weak) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) = – (2/8) . $\log_2$(2/8) – (6/8) . $\log_2$(6/8) = 0.811

Entropy(Decision|Wind=Strong) = – (3/6) . $\log_2$(3/6) – (3/6) . $\log_2$(3/6) = 1

Gain(Decision, Wind) = 0.940 – (8/14).(0.811) – (6/14).(1) = 0.940 – 0.463 – 0.428 = 0.049

There are 8 decisions for weak wind, and 6 decisions for strong wind.

SplitInfo(Decision, Wind) = -(8/14).$\log_2$(8/14) – (6/14).$\log_2$(6/14) = 0.461 + 0.524 = 0.985

GainRatio(Decision, Wind) = Gain(Decision, Wind) / SplitInfo(Decision, Wind) = 0.049 / 0.985 = 0.049

مثالی از C4.5

Outlook Attribute

Outlook is a nominal attribute, too. Its possible values are sunny, overcast and rain.

Gain(Decision, Outlook) = Entropy(Decision) – ∑ (p(Decision|Outlook) . Entropy(Decision|Outlook)) =

Gain(Decision, Outlook) = Entropy(Decision) – p(Decision|Outlook=Sunny) . Entropy(Decision|Outlook=Sunny) – p(Decision|Outlook=Overcast) . Entropy(Decision|Outlook=Overcast) – p(Decision|Outlook=Rain) . Entropy(Decision|Outlook=Rain)

There are 5 sunny instances. 3 of them are concluded as no, 2 of them are concluded as yes.

Entropy(Decision|Outlook=Sunny) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) = -(3/5).$\log_2$(3/5) – (2/5).$\log_2$(2/5) = 0.441 + 0.528 = 0.970

Entropy(Decision|Outlook=Overcast) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) = -(0/4).$\log_2$(0/4) – (4/4).$\log_2$(4/4) = 0

Notice that $\log_2$(0) is actually equal to -∞ but assume that it is equal to 0. Actually, lim (x->0) x.$\log_2$(x) = 0. If you wonder the proof, please look at this post.

Entropy(Decision|Outlook=Rain) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) = -(2/5).$\log_2$(2/5) – (3/5).$\log_2$(3/5) = 0.528 + 0.441 = 0.970

Gain(Decision, Outlook) = 0.940 – (5/14).(0.970) – (4/14).(0) – (5/14).(0.970) – (5/14).(0.970) = 0.246

There are 5 instances for sunny, 4 instances for overcast and 5 instances for rain

SplitInfo(Decision, Outlook) = -(5/14).$\log_2$(5/14) -(4/14).$\log_2$(4/14) -(5/14).$\log_2$(5/14) = 1.577

GainRatio(Decision, Outlook) = Gain(Decision, Outlook)/SplitInfo(Decision, Outlook) =

مثالی از C4.5

Humidity Attribute
As an exception, humidity is a continuous attribute. We need to convert continuous values to nominal ones. C4.5 proposes to perform binary split based on a threshold value. Threshold should be a value which offers maximum gain for that attribute. Let's focus on humidity attribute. Firstly, we need to sort humidity values smallest to largest.

Day	Humidity	Decision
7	65	Yes
6	70	No
9	70	Yes
11	70	Yes
13	75	Yes
3	78	Yes
5	80	Yes
10	80	Yes
14	80	No
1	85	No
2	90	No
12	90	Yes
8	95	No
4	96	Yes

مثالی از C4.5

Now, we need to iterate on all humidity values and separate dataset into two parts as instances less than or equal to current value, and instances greater than the current value. We would calculate the gain or gain ratio for every step. The value which maximizes the gain would be the threshold.

Check 65 as a threshold for humidity

Entropy(Decision|Humidity<=65) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) = -(0/1).$\log_2$(0/1) – (1/1).$\log_2$(1/1) = 0

Entropy(Decision|Humidity>65) = -(5/13).$\log_2$(5/13) – (8/13).$\log_2$(8/13) =0.530 + 0.431 = 0.961

Gain(Decision, Humidity<> 65) = 0.940 – (1/14).0 – (13/14).(0.961) = 0.048

** The statement above refers to that what would branch of decision tree be for less than or equal to 65, and greater than 65. It **does not** refer to that humidity is not equal to 65!*

SplitInfo(Decision, Humidity<> 65) = -(1/14).$\log_2$(1/14) -(13/14).$\log_2$(13/14) = 0.371

GainRatio(Decision, Humidity<> 65) = 0.126

Check 70 as a threshold for humidity

Entropy(Decision|Humidity<=70) = – (1/4).$\log_2$(1/4) – (3/4).$\log_2$(3/4) = 0.811

Entropy(Decision|Humidity>70) = – (4/10).$\log_2$(4/10) – (6/10).$\log_2$(6/10) = 0.970

Gain(Decision, Humidity<> 70) = 0.940 – (4/14).(0.811) – (10/14).(0.970) = 0.940 – 0.231 – 0.692 = 0.014

مثالی از C4.5

SplitInfo(Decision, Humidity<> 70) = -(4/14).$\log_2$(4/14) - (10/14).$\log_2$(10/14) = 0.863

GainRatio(Decision, Humidity<> 70) = 0.016

Check 75 as a threshold for humidity

Entropy(Decision|Humidity<=75) = – (1/5).$\log_2$(1/5) – (4/5).$\log_2$(4/5) = 0.721

Entropy(Decision|Humidity>75) = – (4/9).$\log_2$(4/9) – (5/9).$\log_2$(5/9) = 0.991

Gain(Decision, Humidity<> 75) = 0.940 – (5/14).(0.721) – (9/14).(0.991) = 0.940 – 0.2575 – 0.637 = 0.045

SplitInfo(Decision, Humidity<> 75) = -(5/14).$\log_2$(4/14) - (9/14).$\log_2$(10/14) = 0.940

GainRatio(Decision, Humidity<> 75) = 0.047

I think calculation demonstrations are enough. Now, I skip the calculations and write only results.

Gain(Decision, Humidity <> 78) =0.090, GainRatio(Decision, Humidity <> 78) =0.090

مثالی از C4.5

Gain(Decision, Humidity <> 80) = 0.101, GainRatio(Decision, Humidity <> 80) = 0.107

Gain(Decision, Humidity <> 85) = 0.024, GainRatio(Decision, Humidity <> 85) = 0.027

Gain(Decision, Humidity <> 90) = 0.010, GainRatio(Decision, Humidity <> 90) = 0.016

Gain(Decision, Humidity <> 95) = 0.048, GainRatio(Decision, Humidity <> 95) = 0.128

Here, I ignore the value 96 as threshold because humidity cannot be greater than this value.

As seen, gain maximizes when threshold is equal to 80 for humidity. This means that we need to compare other nominal attributes and comparison of humidity to 80 to create a branch in our tree.

Let’s summarize calculated gain and gain ratios. Outlook attribute comes with both maximized gain and gain ratio. This means that we need to put outlook decision in root of decision tree.

مثالی از C4.5

Attribute	Gain	GainRatio
Wind	0.049	0.049
Outlook	0.246	0.155
Humidity <> 80	0.101	0.107

After then, we would apply similar steps just like as ID3 and create following decision tree. Outlook is put into root node. Now, we should look decisions for different outlook types.

مثالی از C4.5

Outlook = Sunny

We've split humidity for greater than 80, and less than or equal to 80. Surprisingly, decisions would be no if humidity is greater than 80 when outlook is sunny. Similarly, decision would be yes if humidity is less than or equal to 80 for sunny outlook.

Day	Outlook	Temp.	Hum. > 80	Wind	Decision
1	Sunny	85	Yes	Weak	No
2	Sunny	80	Yes	Strong	No
8	Sunny	72	Yes	Weak	No
9	Sunny	69	No	Weak	Yes
11	Sunny	75	No	Strong	Yes

مثالی از C4.5

Outlook = Overcast

If outlook is overcast, then no matter temperature, humidity or wind are, decision will always be yes.

Day	Outlook	Temp.	Hum. > 80	Wind	Decision
3	Overcast	83	No	Weak	Yes
7	Overcast	64	No	Strong	Yes
12	Overcast	72	Yes	Strong	Yes
13	Overcast	81	No	Weak	Yes

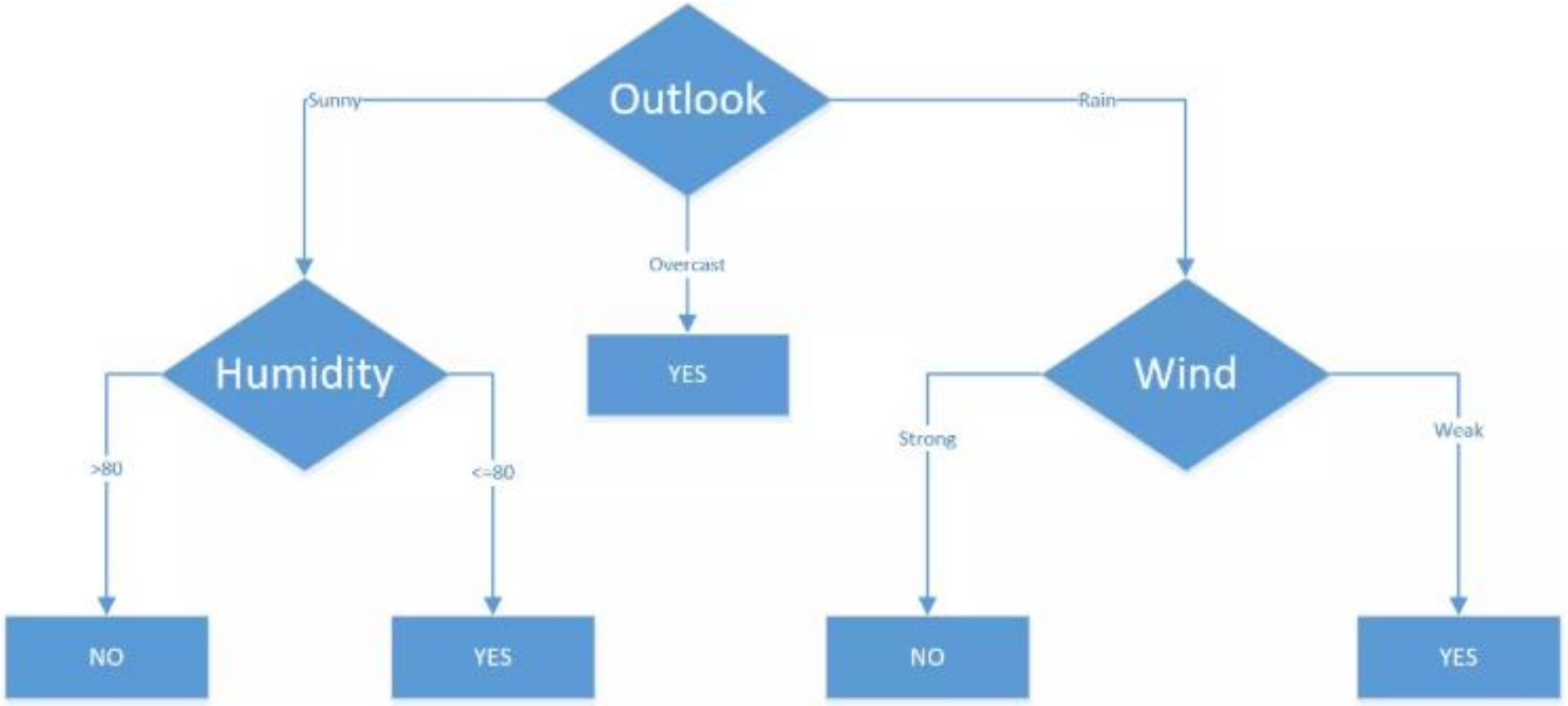
مثالی از C4.5

Outlook = Rain

We’ve just filtered rain outlook instances. As seen, decision would be yes when wind is weak, and it would be no if wind is strong.
Final form of decision table is demonstrated below.

Day	Outlook	Temp.	Hum. > 80	Wind	Decision
4	Rain	70	Yes	Weak	Yes
5	Rain	68	No	Weak	Yes
6	Rain	65	No	Strong	No
10	Rain	75	No	Weak	Yes
14	Rain	71	No	Strong	No

مثالی از C4.5


Outlook
Sunny
Overcast
Rain
Humidity
YES
Wind
>80
<=80
Strong
Weak
NO
YES
NO
YES

مثالی از C4.5

Conclusion

So, C4.5 algorithm solves most of problems in ID3. The algorithm uses gain ratios instead of gains. In this way, it creates more generalized trees and not to fall into overfitting. Moreover, the algorithm transforms continuous attributes to nominal ones based on gain maximization and in this way it can handle continuous data. Additionally, it can ignore instances including missing data and handle missing dataset. On the other hand, both ID3 and C4.5 requires high CPU and memory demand. Besides, most of authorities think decision tree algorithms in data mining field instead of machine learning.

مثالی از درخت رگرسیون

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30

مثالی از درخت رگرسیون

Standard deviation

Golf players = {25, 30, 46, 45, 52, 23, 43, 35, 38, 46, 48, 52, 44, 30}

Average of golf players = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30)/14 = 39.78

Standard deviation of golf players = $\sqrt{[((25 – 39.78)^2 + (30 – 39.78)^2 + (46 – 39.78)^2 + … + (30 – 39.78)^2)/14]}$ = 9.32

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Outlook

Outlook can be sunny, overcast and rain. We need to calculate standard deviation of golf players for all of these outlook candidates.

مثالی از درخت رگرسیون

Golf players for sunny outlook = {25, 30, 35, 38, 48}
Average of golf players for sunny outlook = (25+30+35+38+48)/5 = 35.2
Standard deviation of golf players for sunny outlook = $\sqrt{(((25 - 35.2)^2 + (30 - 35.2)^2 + \dots)/5)} = 7.78$

Overcast outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Golf players for overcast outlook = {46, 43, 52, 44}
Average of golf players for overcast outlook = (46 + 43 + 52 + 44)/4 = 46.25
Standard deviation of golf players for overcast outlook = $\sqrt{(((46-46.25)^2+(43-46.25)^2+\dots)} = 3.49$

مثالی از درخت رگرسیون

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

Rainy outlook

Golf players for overcast outlook = {45, 52, 23, 46, 30}

Average of golf players for overcast outlook = (45+52+23+46+30)/5 = 39.2

Standard deviation of golf players for rainy outlook = $\sqrt{(((45 - 39.2)^2+(52 - 39.2)^2+\ldots)/5)}=10.87$

مثالی از درخت رگرسیون

Summarizing standard deviations for the outlook feature

Outlook	Stdev of Golf Players	Instances
Overcast	3.49	4
Rain	10.87	5
Sunny	7.78	5

Weighted standard deviation for outlook = (4/14)x3.49 + (5/14)x10.87 + (5/14)x7.78 = 7.66
You might remember that we have calculated the global standard deviation of golf players 9.32 in previous steps. Standard deviation reduction is difference of the global standard deviation and standard deviation for current feature. In this way, maximized standard deviation reduction will be the decision node.
Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

مثالی از درخت رگرسیون

Temperature
Temperature can be hot, cool or mild. We will calculate standard deviations for those candidates.
Hot temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
13	Overcast	Hot	Normal	Weak	44

Golf players for hot temperature = {25, 30, 46, 44}
Standard deviation of golf players for hot temperature = 8.95

مثالی از درخت رگرسیون

Cool temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38

Golf players for cool temperature = {52, 23, 43, 38}
Standard deviation of golf players for cool temperature = 10.51

مثالی از درخت رگرسیون

Mild temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for mild temperature = {45, 35, 46, 48, 52, 30}
Standard deviation of golf players for mild temperature = 7.65

مثالی از درخت رگرسیون

Summarizing standard deviations for temperature feature

Temperature	Stdev of Golf Players	Instances
Hot	8.95	4
Cool	10.51	4
Mild	7.65	6

Weighted standard deviation for temperature = (4/14)x8.95 + (4/14)x10.51 + (6/14)x7.65 = 8.84
Standard deviation reduction for temperature = 9.32 – 8.84 = 0.47

مثالی از درخت رگرسیون

Humidity

Humidity is a binary class. It can either be normal or high.

High humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for high humidity = {25, 30, 46, 45, 35, 52, 30}

Standard deviation for golf players for high humidity = 9.36

مثالی از درخت رگرسیون

Normal humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
13	Overcast	Hot	Normal	Weak	44

Golf players for normal humidity = {52, 23, 43, 38, 46, 48, 44}
Standard deviation for golf players for normal humidity = 8.73

مثالی از درخت رگرسیون

Summarizing standard deviations for humidity feature

Humidity	Stdev of Golf Player	Instances
High	9.36	7
Normal	8.73	7

Weighted standard deviation for humidity = (7/14)x9.36 + (7/14)x8.73 = 9.04
Standard deviation reduction for humidity = 9.32 – 9.04 = 0.27

مثالی از درخت رگرسیون

Wind

Wind is a binary class, too. It can either be Strong or Weak.

Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for strong wind= {30, 23, 43, 48, 52, 30}

Standard deviation for golf players for strong wind = 10.59

مثالی از درخت رگرسیون

Weak Wind

1	Sunny	Hot	High	Weak	25
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
13	Overcast	Hot	Normal	Weak	44

Golf players for weakk wind= {25, 46, 45, 52, 35, 38, 46, 44}
Standard deviation for golf players for weak wind = 7.87

مثالی از درخت رگرسیون

Summarizing standard deviations for wind feature

Wind	Stdev of Golf Player	Instances
Strong	10.59	6
Weak	7.87	8

Weighted standard deviation for wind = (6/14)x10.59 + (8/14)x7.87 = 9.03
Standard deviation reduction for wind = 9.32 – 9.03 = 0.29
So, we've calculated standard deviation reduction values for all features.
The winner is outlook because it has the highest score.

مثالی از درخت رگرسیون

Feature	Standard Deviation Reduction
Outlook	1.66
Temperature	0.47
Humidity	0.27
Wind	0.29

We’ll put outlook decision at the top of decision tree. Let’s monitor the new sub data sets for the candidate branches of outlook feature.

مثالی از درخت رگرسیون

Sunny Outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}
Standard deviation for sunny outlook = 7.78
Notice that we will use this standard deviation value as global standard deviation for this sub data set.

مثالی از درخت رگرسیون

Sunny outlook and Hot Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30

Standard deviation for sunny outlook and hot temperature
= 2.5

Sunny outlook and Cool Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and cool temperature = 0

مثالی از درخت رگرسیون

Sunny outlook and Mild Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
8	Sunny	Mild	High	Weak	35
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and mild temperature = 6.5

Summary of standard deviations for temperature feature when outlook is sunny

Temperature	Stdev for Golf Players	Instances
Hot	2.5	2
Cool	0	1
Mild	6.5	2

Weighted standard deviation for sunny outlook and temperature = (2/5)x2.5 + (1/5)x0 + (2/5)x6.5 = 3.6
Standard deviation reduction for sunny outlook and temperature = 7.78 – 3.6 = 4.18

مثالی از درخت رگرسیون

Sunny outlook and high humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35

Standard deviation for sunny outlook and high humidity = 4.08

Sunny outlook and normal humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and normal humidity = 5

مثالی از درخت رگرسیون

Summarizing standard deviations for humidity feature when outlook is sunny

Humidity	Stdev for Golf Players	Instances
High	4.08	3
Normal	5.00	2

Weighted standard deviations for sunny outlook and humidity = (3/5)x4.08 + (2/5)x5 = 4.45
Standard deviation reduction for sunny outlook and humidity = 7.78 – 4.45 = 3.33

Sunny outlook and Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and strong wind = 9

مثالی از درخت رگرسیون

Sunny outlook and Weak Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and weak wind = 5.56

Wind	Stdev for Golf Players	Instances
Strong	9	2
Weak	5.56	3

Weighted standard deviations for sunny outlook and wind = (2/5)x9 + (3/5)x5.56 = 6.93
Standard deviation reduction for sunny outlook and wind = 7.78 – 6.93 = 0.85

مثالی از درخت رگرسیون

We've calculated standard deviation reductions for sunny outlook. The winner is temperature.

Feature	Standard Deviation Reduction
Temperature	4.18
Humidity	3.33
Wind	0.85

Pruning

Cool branch has one instance in its sub data set. We can say that if outlook is sunny and temperature is cool, then there would be 38 golf players. But what about hot branch? There are still 2 instances. Should we add another branch for weak wind and strong wind? No, we should not. Because this causes over-fitting. We should terminate building branches, for example if there are less than five instances in the sub data set. Or standard deviation of the sub data set can be less than 5% of the entire data set. I prefer to apply the first one. I will terminate the branch if there are less than 5 instances in the current sub data set. If this termination condition is satisfied, then I will calculate the average of the sub data set. This operation is called as pruning in decision tree trees.

مثالی از درخت رگرسیون

Overcast outlook

Overcast outlook branch has already 4 instances in the sub data set. We can terminate building branches for this leaf. Final decision will be average of the following table for overcast outlook.

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

If outlook is overcast, then there would be (46+43+52+44)/4 = 46.25 golf players.

مثالی از درخت رگرسیون

Rainy Outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

We need to find standard deviation reduction values for the rest of the features in same way for the sub data set above.
Standard deviation for rainy outlook = 10.87
Notice that we will use this value as global standard deviation for this branch in reduction step.

مثالی از درخت رگرسیون

Rainy outlook and temperature

Temperature	Standard deviation for golf players	instances
Cool	14.50	2
Mild	7.32	3

Weighted standard deviation for rainy outlook and temperature = (2/5)x14.50 + (3/5)x7.32 = 10.19
Standard deviation reduction for rainy outlook and temperature = 10.87 – 10.19 = 0.67

Rainy outlook and humidity

Humidity	Standard deviation for golf players	instances
High	7.50	2
Normal	12.50	3

Weighted standard deviation for rainy outlook and humidity = (2/5)x7.50 + (3/5)x12.50 = 10.50
Standard deviation reduction for rainy outlook and humidity = 10.87 – 10.50 = 0.37

مثالی از درخت رگرسیون

Rainy outlook and wind

Wind	Standard deviation for golf players	instances
Weak	3.09	3
Strong	3.5	2

Weighted standard deviation for rainy outlook and wind = (3/5)x3.09 + (2/5)x3.5 = 3.25
Standard deviation reduction for rainy outlook and wind = 10.87 – 3.25 = 7.62

Summarizing rainy outlook

As illustrated below, the winner is wind feature.

Feature	Standard deviation reduction
Temperature	0.67
Humidity	0.37
Wind	7.62

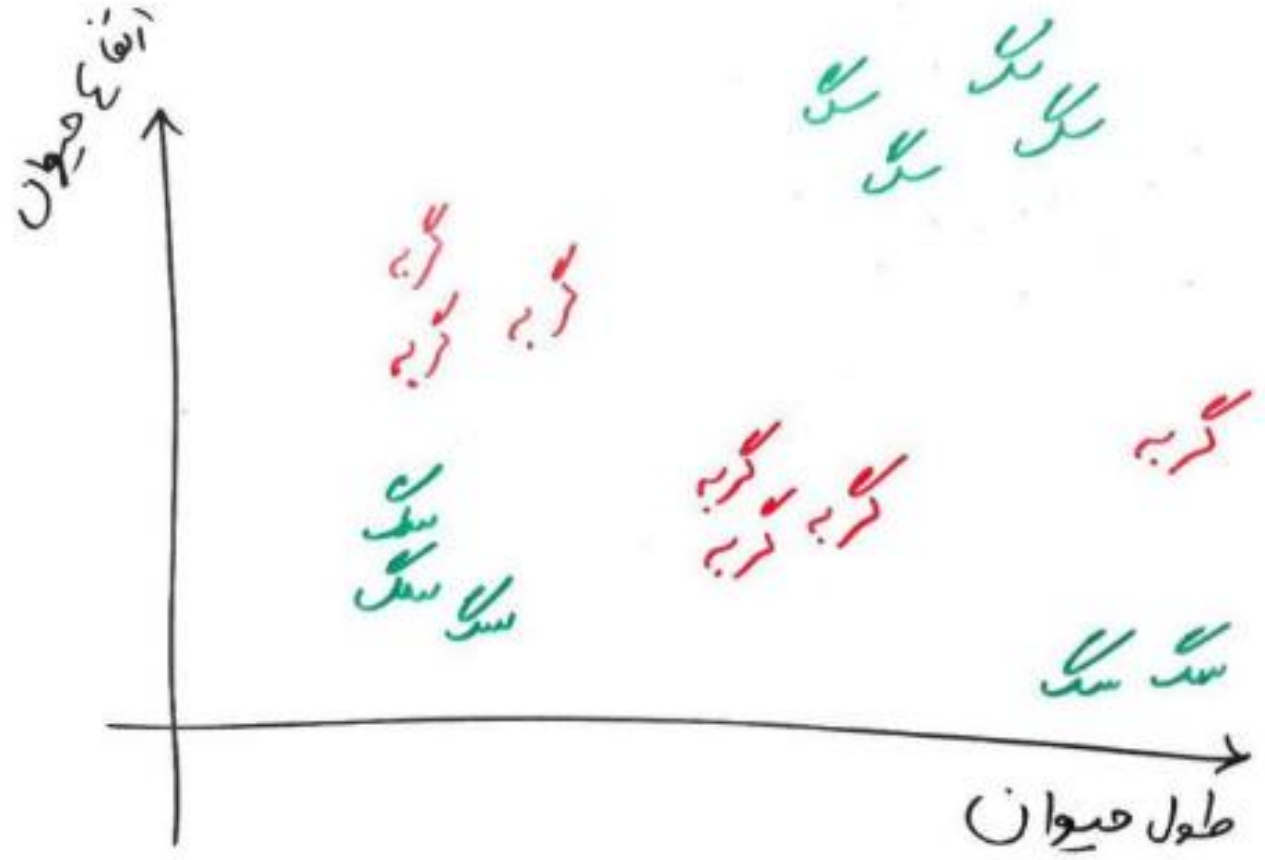
مثالی از درخت رگرسیون

So, we have mentioned how to build decision trees for regression problems. Even though, decision trees are powerful way to classify problems, they can be adapted into regression problems as mentioned. Regression trees tend to over-fit much more than classification trees. Termination rule should be tuned carefully to avoid over-fitting. Finally, lecture notes of Dr. Saed Sayad (University of Toronto) guides me to create this content.

کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

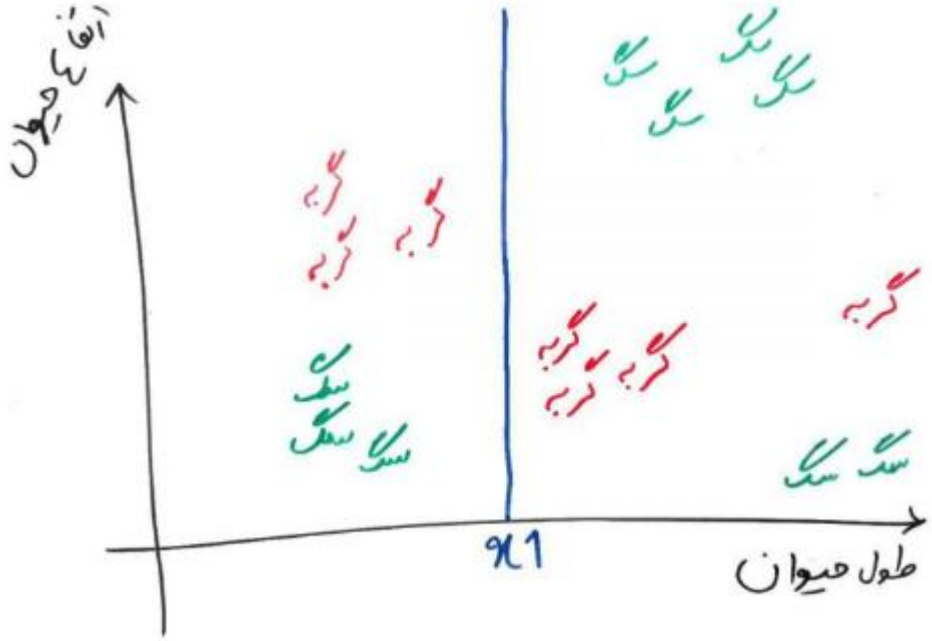
- یکی از محبوب‌ترین و در عین حال از ساده‌ترین الگوریتم‌های درخت‌های تصمیم، درخت تصمیمِ CARTاست که کاربردهای زیادی در طبقه بندی و رگرسیون دارد. CARTکه خود مَخفف Classification And Regression Treeاست بر اساس درخت های دودویی(باینری) بنا نهاده شده است.
- این درخت(و البته درخت‌های دیگر) می‌تواند پایه‌ای برای الگوریتم‌های پیچیده‌تر مانند جنگلِ تصادفی(Random Forest)باشد.
- الگوریتم درختِ تصمیمِ CARTبرای ساختِ درختِ تصمیم، داده‌ها را به قسمت‌های دوتایی تَقسیم کرَده و بَر اساس آن‌ها درختَ دودویَی(باینری) را می‌سازد.
- فرض کنید تعدادی حیوان داریم(سگ و گربه) که هر کدام دو ویژگیِ طولِ حیوان و ارتفاعِ حیوان را دارند. بر اساس این دو ویژگی می‌خواهیم سگ‌هَا و گربه‌ها را از هم دیگر جداَ کنیم. ۱۶ حیوان داریم که هر کدام ۲ویژگی دارند.

کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART



کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

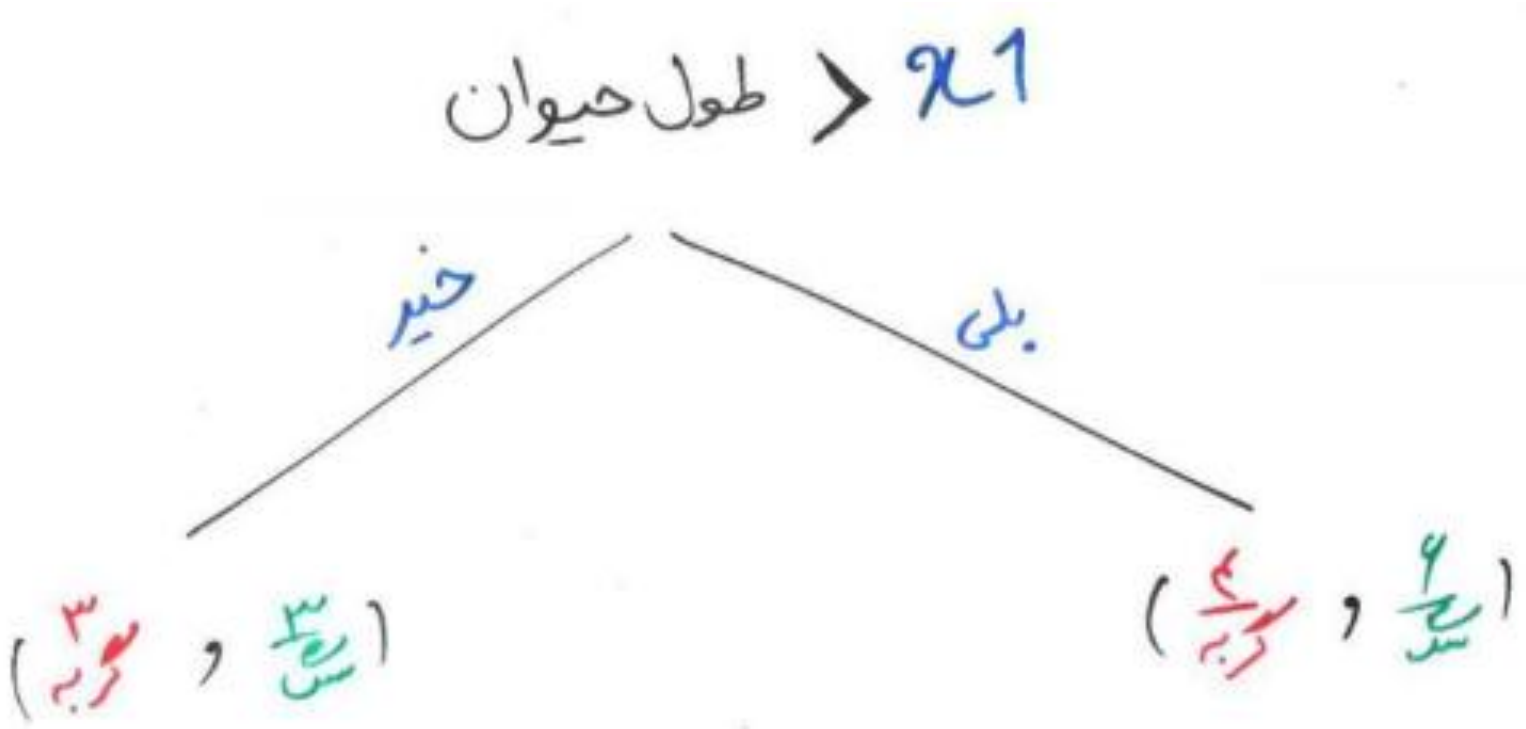
- **حال درخت تصمیم CART میخواهد یک طبقه بند ایجاد کند تا بتواند با دقت بالا تمایز بین سگ ها و گربه ها را تشخیص دهد. درخت CART این کار را در مرحله های مختلفی انجام می دهد. مرحله اول مانند شکل زیر انجام است:**



- **الگوریتمِ CARTبه این صورت عمل می‌کند. در تصویر بالا مشاهده می‌کنید که داده‌ها را به دو قسمتِ تصمیم گردیم. این کار با با یکخط آبی در محورِ طول حیوان انجام شده است و حیوانات را به دو قسمت(که طول آن ها بزرگتر یا کوچکترِ از است تقسیم می‌کنیم.**

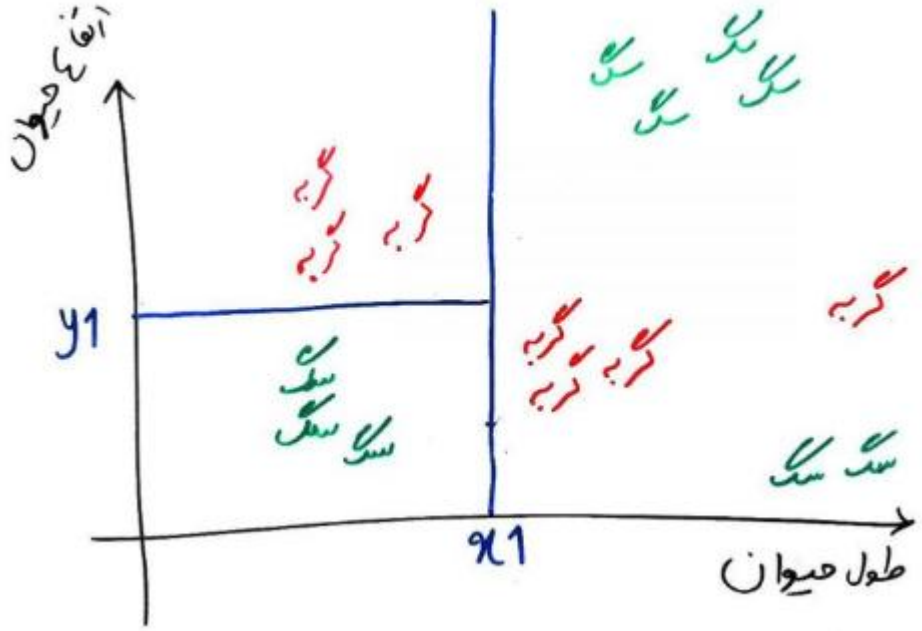
کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

- حال با این کار یک درخت دودویی ایجاد می شود. مانند تصویر زیر:



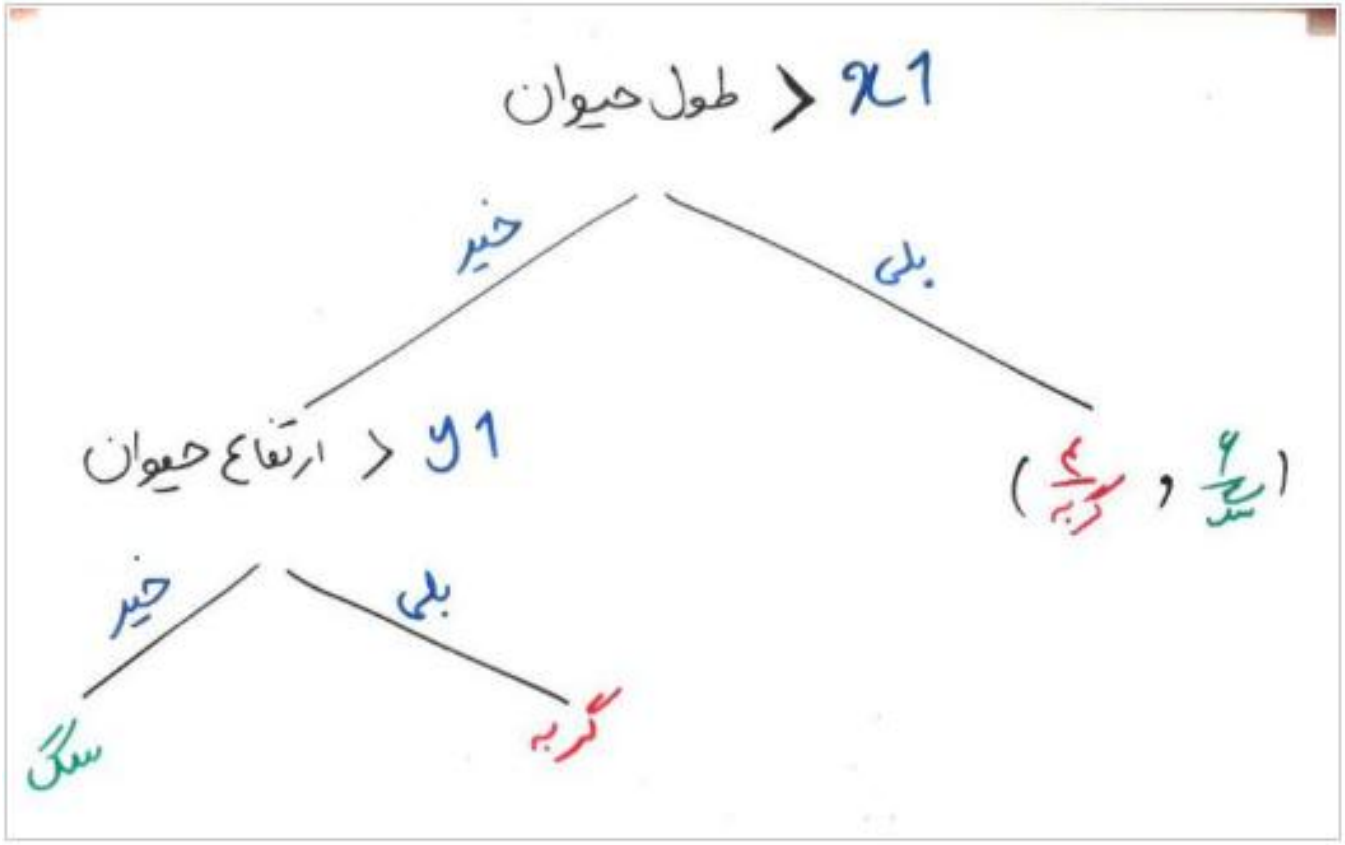
کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

- این درختِ دودویی که بر اساس خط X1 کشیده شده است دو قسمت دارد. اگر طول حیوان بزرگتر از X1 بود، سمت راست X1 مورد نظر است که در آن قسمت ۶ سگ و ۴ گربه داریم. اگر طولِ حیوان کوچکتر از X1 بود سمت چپ این خط مورد نظر است که ۳ سگ و ۳ گربه داریم. حال می‌توان هر کدام از این دو قسمت را نیز، به قسمت های کوچکتری تقسیم کرد:



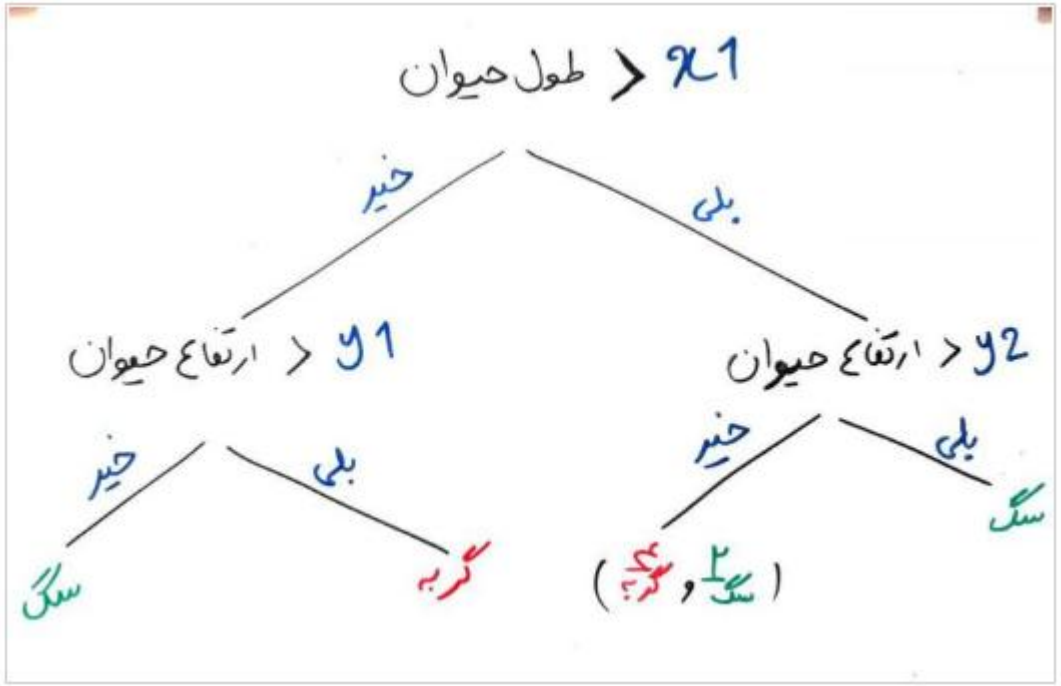
کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

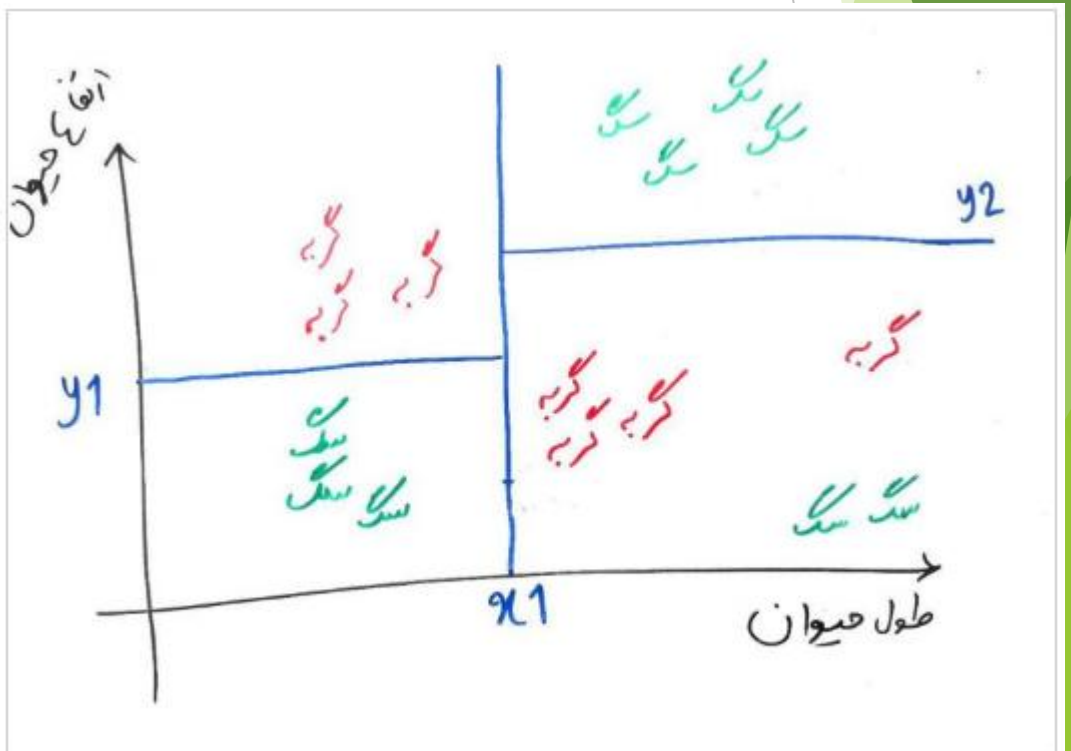
- همان طور که میبینید، قسمت سمت چپ خط X1 را با یک خطِ دیگر جدا کردیم. این خط که همان خط Y1است توانست گربه ها و سگ ها را در قسمت سمت چپ X1از همدیگر جدا کند. درخت CARTمناسب تا به اینجا به این صورت است:
-



کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

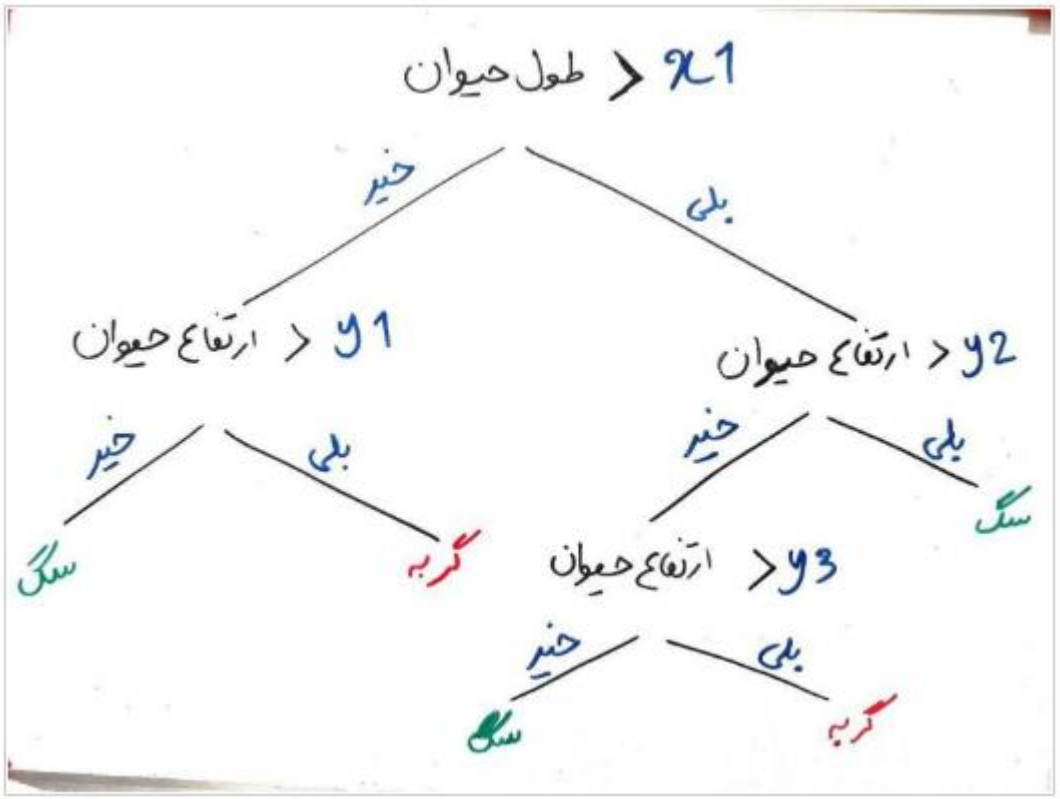
- همان طور که می‌بینید، قسمت سمت چپ درخت را بسط دادیم. یعنی هنگامی که طولِ حیوان کوچکتر از X1است را بسط داده و در این شرایطَ اگر ارتفاع حیوان کوچکتر از Y1بود این حیوان را سگ و در اگر ارتفاع حیوان بزرگتر از Y1بود، این حیوان گربه است.

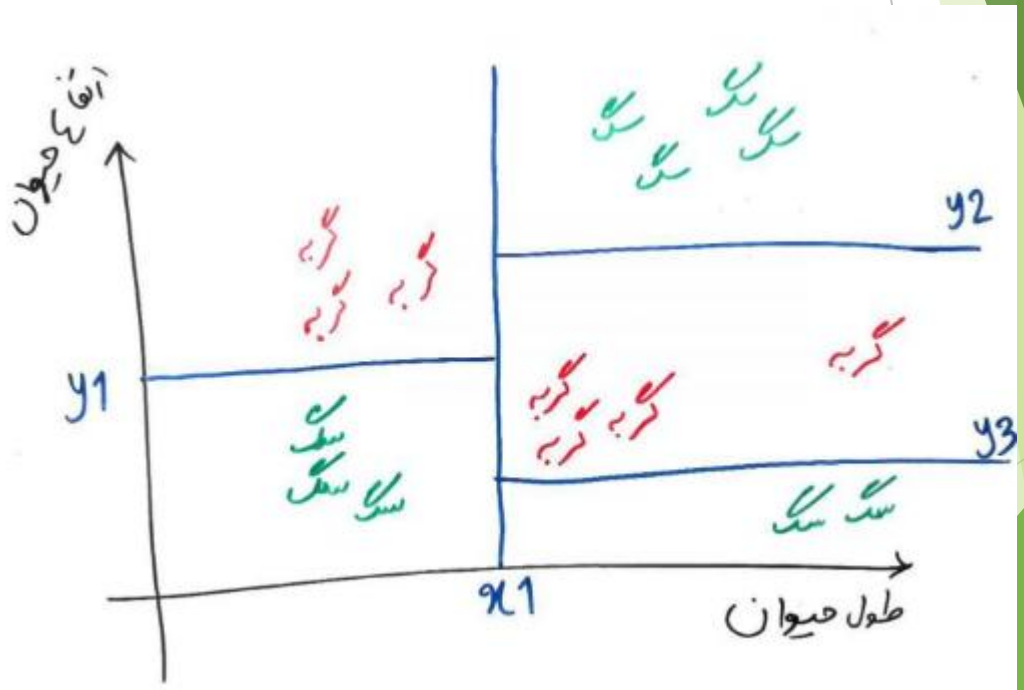




کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

- **و در آخر خطی مانند زیر میتواند طبقه بند را با دقت بالایی مدل سازی کند:**





کلاس بندی– الگوریتم طبقه بند درخت تصمیم CART

- درخت تصمیمِ CARTبرای انتخاب گره های درخت از معیاری به نام معیار شاخص Giniَاستفادهَ می کند.
- همان طور که درخت های ID3وC4.5 ازEntropyو Gain استفاده می‌کرند.
- هر چقدر یک ویژگی(بُعد) شاخصِ Giniکمتری داشته باشد آن ویژگی اطلاعات بیشتری دارد و میتواند در درخت سَاخته شده بالاتر قرار بگیرد.

کلاس بندی – الگوریتم طبقه بند درخت تصمیم CART

- تفاوتِ شاخص Gini و Entropy :
- معمولاً شاخص Gini برای داده‌هایی که دارای قسمتِ بزرگتر هستند به درد می‌خورد این دَر حالی است که Entropy به درد داده‌هایی می‌خورد که قسمت های کوچک زیادی دارند که مقادیر یکتا در آن ها بیشتر است.
- جهت جلوگیری از Overfit شدن درخت تصمیم CART میتوان از یک شرطِ توقف استفاده کرد.
- یکی از این روش‌ها استفاده از تعداد مشخص نمونه در زیر درخت خاص است به گونه ای که اگر تعداد نمونه‌ها در یک زیر درختْ از یک حد آستانه کمتر شد، دیگر درخت ریشه‌سازی را ادامه نمی دهد.

مثالی از CART

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

مثالی از CART

Gini index

Gini index is a metric for classification tasks in CART. It stores sum of squared probabilities of each class. We can formulate it as illustrated below.

Gini = 1 – Σ $(Pi)^2$ for i=1 to number of classes

Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. I will summarize the final decisions for outlook feature.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$

Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$

Gini(Outlook=Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$

Then, we will calculate weighted sum of gini indexes for outlook feature.

Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = 0.342

مثالی از CART

Temperature
Similarly, temperature is a nominal feature and it could have 3 different values: Cool, Hot and Mild. Let's summarize decisions for temperature feature.

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

Gini(Temp=Hot) = 1 – (2/4)2 – (2/4)2 = 0.5
Gini(Temp=Cool) = 1 – (3/4)2 – (1/4)2 = 1 – 0.5625 – 0.0625 = 0.375
Gini(Temp=Mild) = 1 – (4/6)2 – (2/6)2 = 1 – 0.444 – 0.111 = 0.445
We'll calculate weighted sum of gini index for temperature feature
Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = 0.439

مثالی از CART

Humidity
Humidity is a binary class feature. It can be high or normal.

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

Gini(Humidity=High) = 1 – (3/7)2 – (4/7)2 = 1 – 0.183 – 0.326 = 0.489
Gini(Humidity=Normal) = 1 – (6/7)2 – (1/7)2 = 1 – 0.734 – 0.02 = 0.244
Weighted sum for humidity feature will be calculated next
Gini(Humidity) = (7/14) x 0.489 + (7/14) x 0.244 = 0.367

مثالی از CART

Wind

Wind is a binary class similar to humidity. It can be weak and strong

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

Gini(Wind=Weak) = 1 – (6/8)2 – (2/8)2 = 1 – 0.5625 – 0.062 = 0.375
Gini(Wind=Strong) = 1 – (3/6)2 – (3/6)2 = 1 – 0.25 – 0.25 = 0.5
Gini(Wind) = (8/14) x 0.375 + (6/14) x 0.5 = 0.428

مثالی از CART

Time to decide

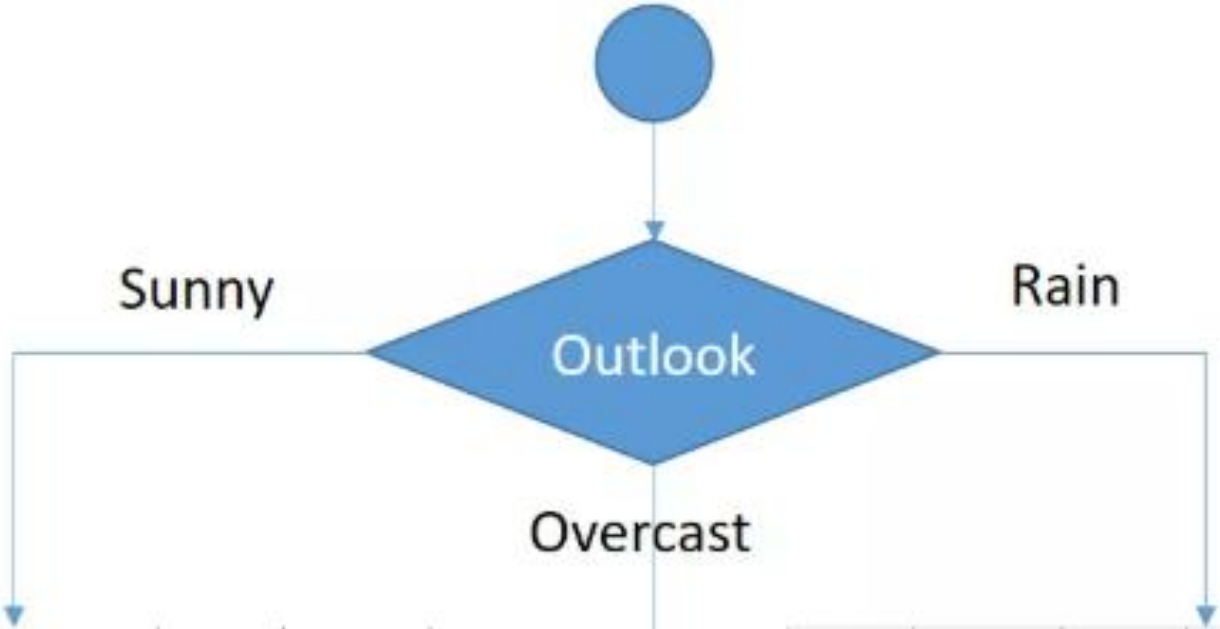
We've calculated gini index values for each feature. **The winner will be outlook feature because its cost is the lowest.**

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428

We'll put outlook decision at the top of the tree.

You might realize that sub dataset in the overcast leaf has only yes decisions. This means that overcast leaf is over.

مثالی از CART



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

مثالی از CART

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

مثالی از CART

We will apply same principles to those sub datasets in the following steps.
Focus on the sub dataset for sunny outlook. We need to find the gini index scores for temperature, humidity and wind features respectively.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

مثالی از CART

Gini of temperature for sunny outlook

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini(Outlook=Sunny and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$
Gini(Outlook=Sunny and Temp.=Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$
Gini(Outlook=Sunny and Temp.=Mild) = $1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$
Gini(Outlook=Sunny and Temp.) = $(2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2$

مثالی از CART

Gini of humidity for sunny outlook

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini(Outlook=Sunny and Humidity=High) = $1 – (0/3)^2 – (3/3)^2 = 0$
Gini(Outlook=Sunny and Humidity=Normal) = $1 – (2/2)^2 – (0/2)^2 = 0$
Gini(Outlook=Sunny and Humidity) = $(3/5)x0 + (2/5)x0 = 0$

Gini of wind for sunny outlook

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

Gini(Outlook=Sunny and Wind=Weak) = $1 – (1/3)^2 – (2/3)^2 = 0.266$
Gini(Outlook=Sunny and Wind=Strong) = $1- (1/2)^2 – (1/2)^2 = 0.2$
Gini(Outlook=Sunny and Wind) = $(3/5)x0.266 + (2/5)x0.2 = 0.466$

مثالی از CART

Decision for sunny outlook

We’ve calculated gini index scores for feature when outlook is sunny.

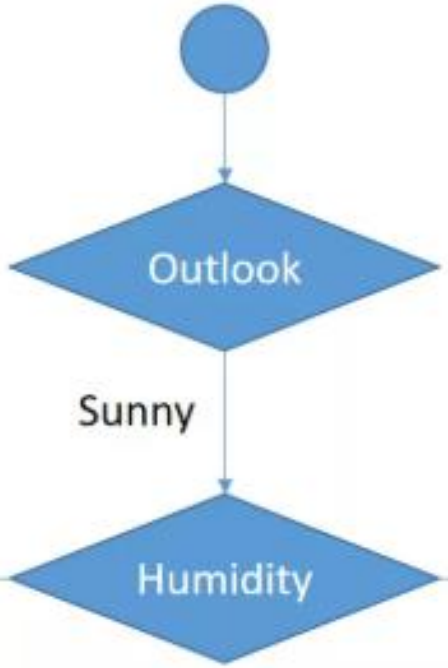
The winner is humidity because it has the lowest value.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466

We’ll put humidity check at the extension of sunny outlook.

As seen, decision is always no for high humidity and sunny outlook. On the other hand, decision will always be yes for normal humidity and sunny outlook. This branch is over.

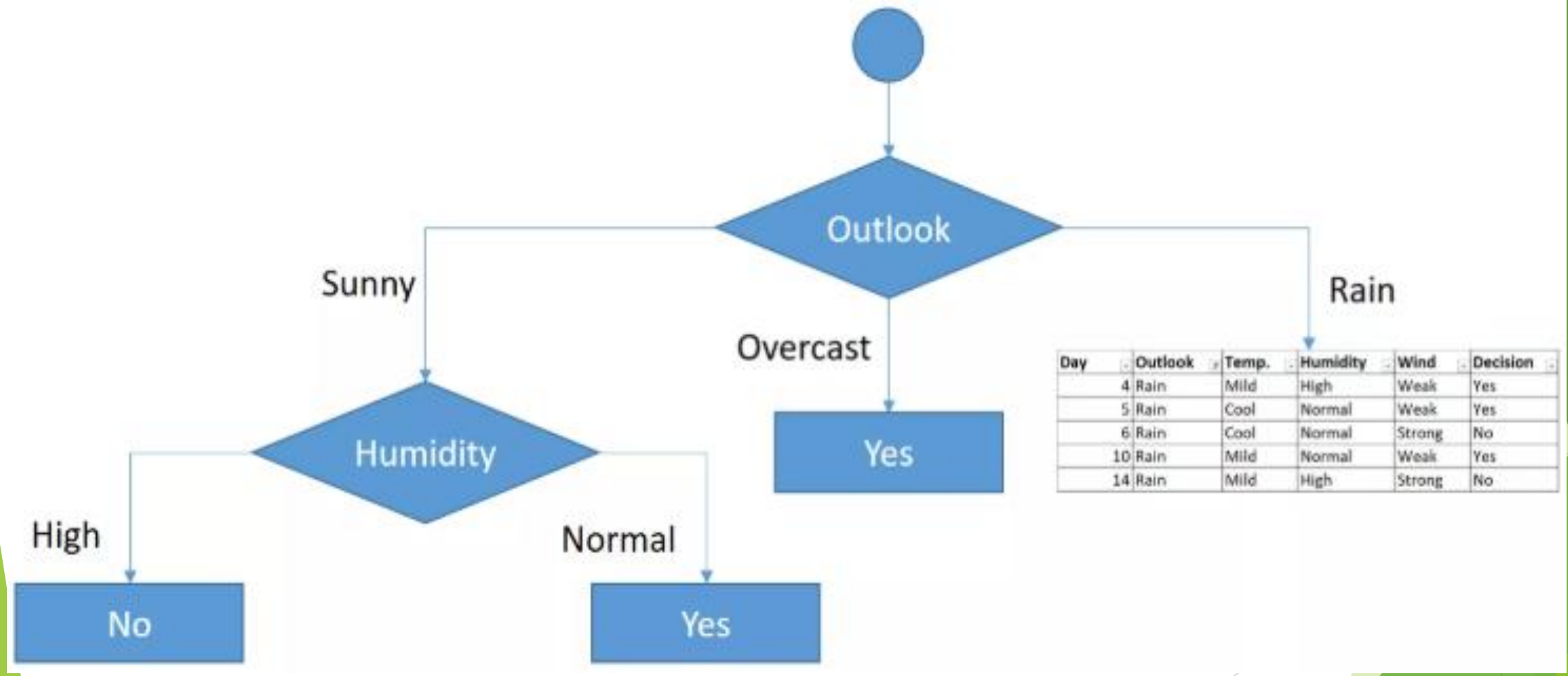
مثالی از CART



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

مثالی از CART



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

مثالی از CART

Rain outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

We'll calculate gini index scores for temperature, humidity and wind features when outlook is rain.

مثالی از CART

Gini of temperature for rain outlook

Temperature	Yes	No	Number of instances
Cool	1	1	2
Mild	2	1	3

Gini(Outlook=Rain and Temp.=Cool) = $1 – (1/2)^2 – (1/2)^2 = 0.5$
Gini(Outlook=Rain and Temp.=Mild) = $1 – (2/3)^2 – (1/3)^2 = 0.444$
Gini(Outlook=Rain and Temp.) = $(2/5)x0.5 + (3/5)x0.444 = 0.466$

Gini of humidity for rain outlook

Humidity	Yes	No	Number of instances
High	1	1	2
Normal	2	1	3

Gini(Outlook=Rain and Humidity=High) = $1 – (1/2)^2 – (1/2)^2 = 0.5$
Gini(Outlook=Rain and Humidity=Normal) = $1 – (2/3)^2 – (1/3)^2 = 0.444$
Gini(Outlook=Rain and Humidity) = $(2/5)x0.5 + (3/5)x0.444 = 0.466$

مثالی از CART

Gini of wind for rain outlook

Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

Gini(Outlook=Rain and Wind=Weak) = $1 – (3/3)^2 – (0/3)^2 = 0$
Gini(Outlook=Rain and Wind=Strong) = $1 – (0/2)^2 – (2/2)^2 = 0$
Gini(Outlook=Rain and Wind) = $(3/5)x0 + (2/5)x0 = 0$

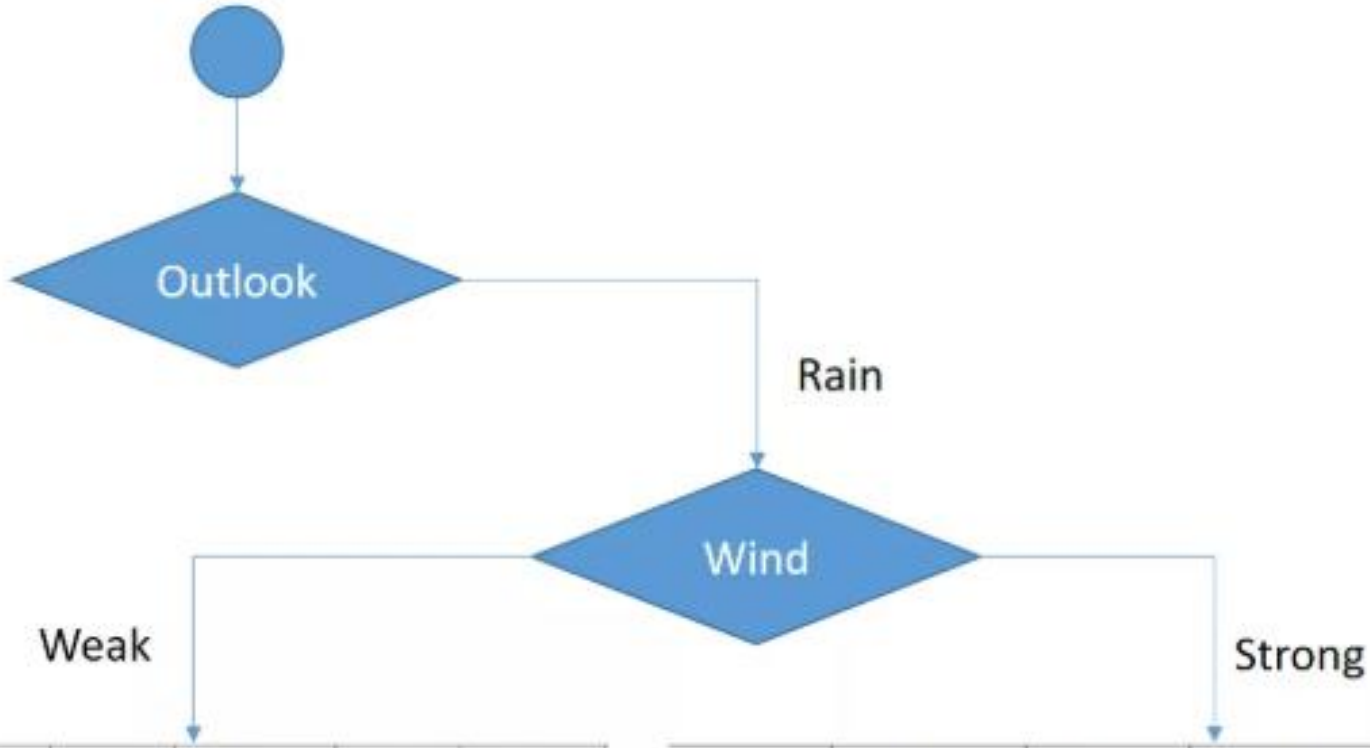
Decision for rain outlook

The winner is wind feature for rain outlook because it has the minimum gini index score in features.

Feature	Gini index
Temperature	0.466
Humidity	0.466
Wind	0

Put the wind feature for rain outlook branch and monitor the new sub data sets.

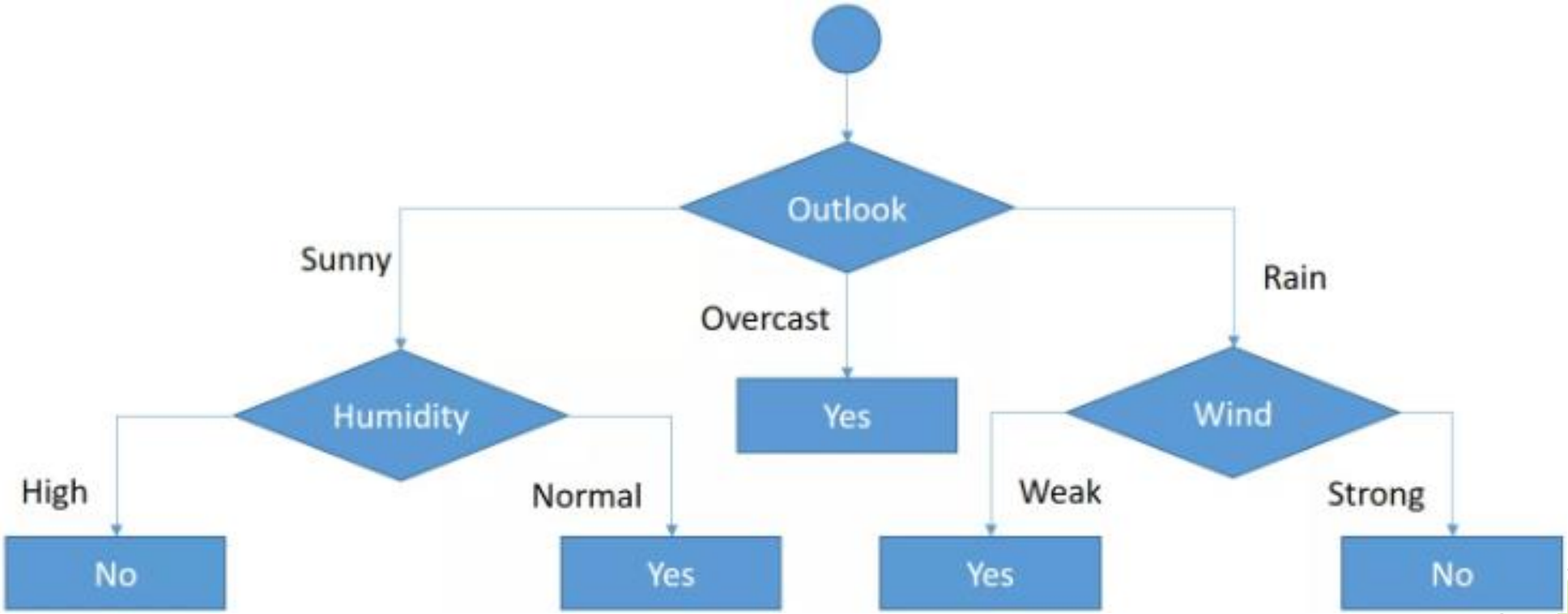
مثالی از CART



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

مثالی از CART



مثالی از CART

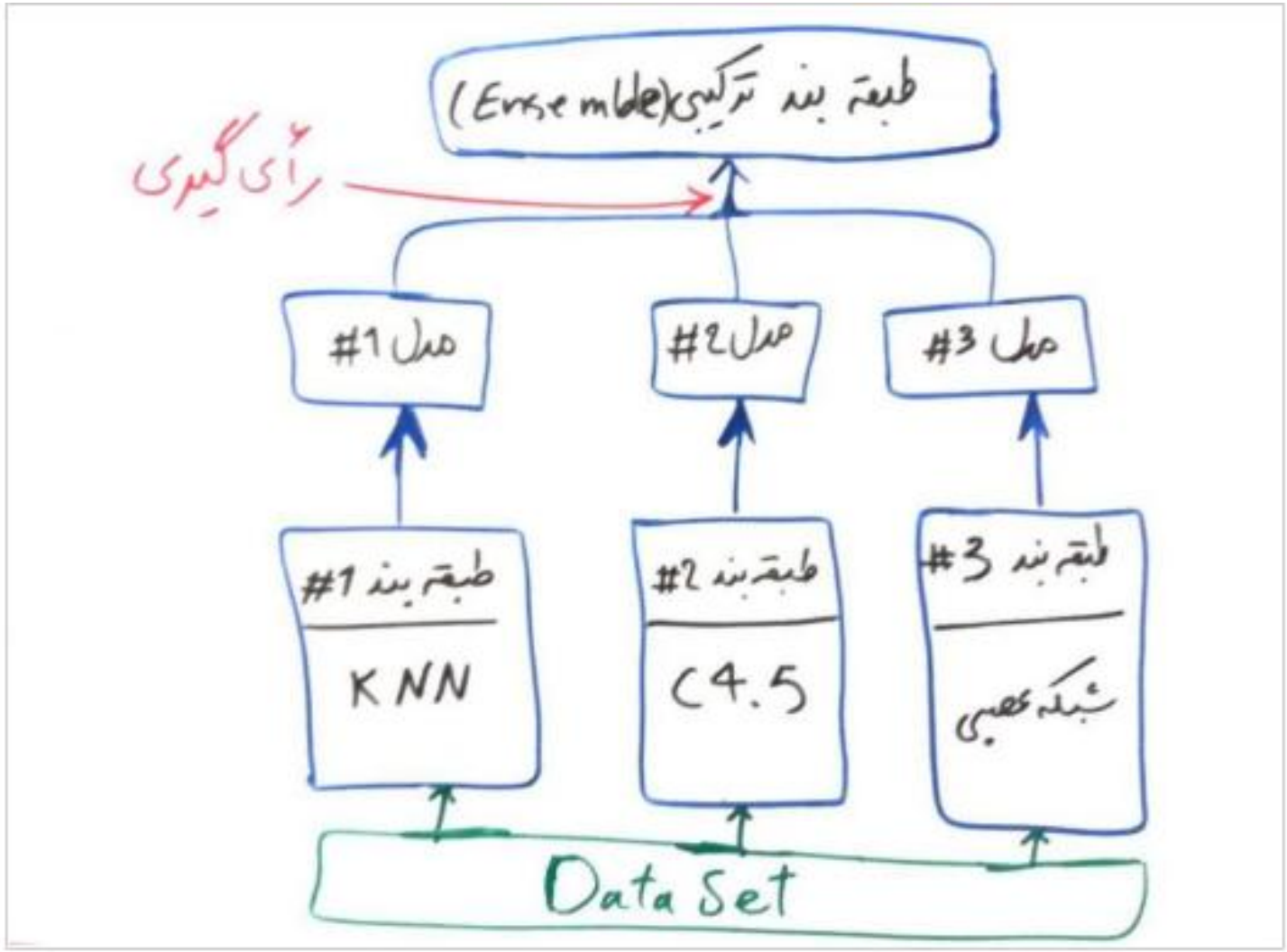
As seen, decision is always yes when wind is weak. On the other hand, decision is always no if wind is strong. This means that this branch is over.

So, decision tree building is over. We have built a decision tree by hand. BTW, you might realize that we've created exactly the same tree in ID3 example. This does not mean that ID3 and CART algorithms produce same trees always. We are just lucky. Finally, I believe that CART is easier than ID3 and C4.5, isn't it?

طبقه بند ترکیبی(Ensemble Classifierو مبحث Baggingو Boosting

- هر طبقه‌بند یک مدل را بر روی داده‌های آموزشی می‌سازد تا به وسیله‌ی آن بتواند تفاوت‌ها را در طبقه‌های مختلف درک کند. طبقه‌بندِ ترکیبی، اما به جای اینکه خود یک مدل بسازد از مدل‌های ساخته شده توَسطِ بقیه‌ی طبقه‌بندها استفاده کرده و با یک آمارگیری، مشخص می‌کند کَه کدام طبقه را برای نمونه‌ی جاری باید برگزینند.

طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting



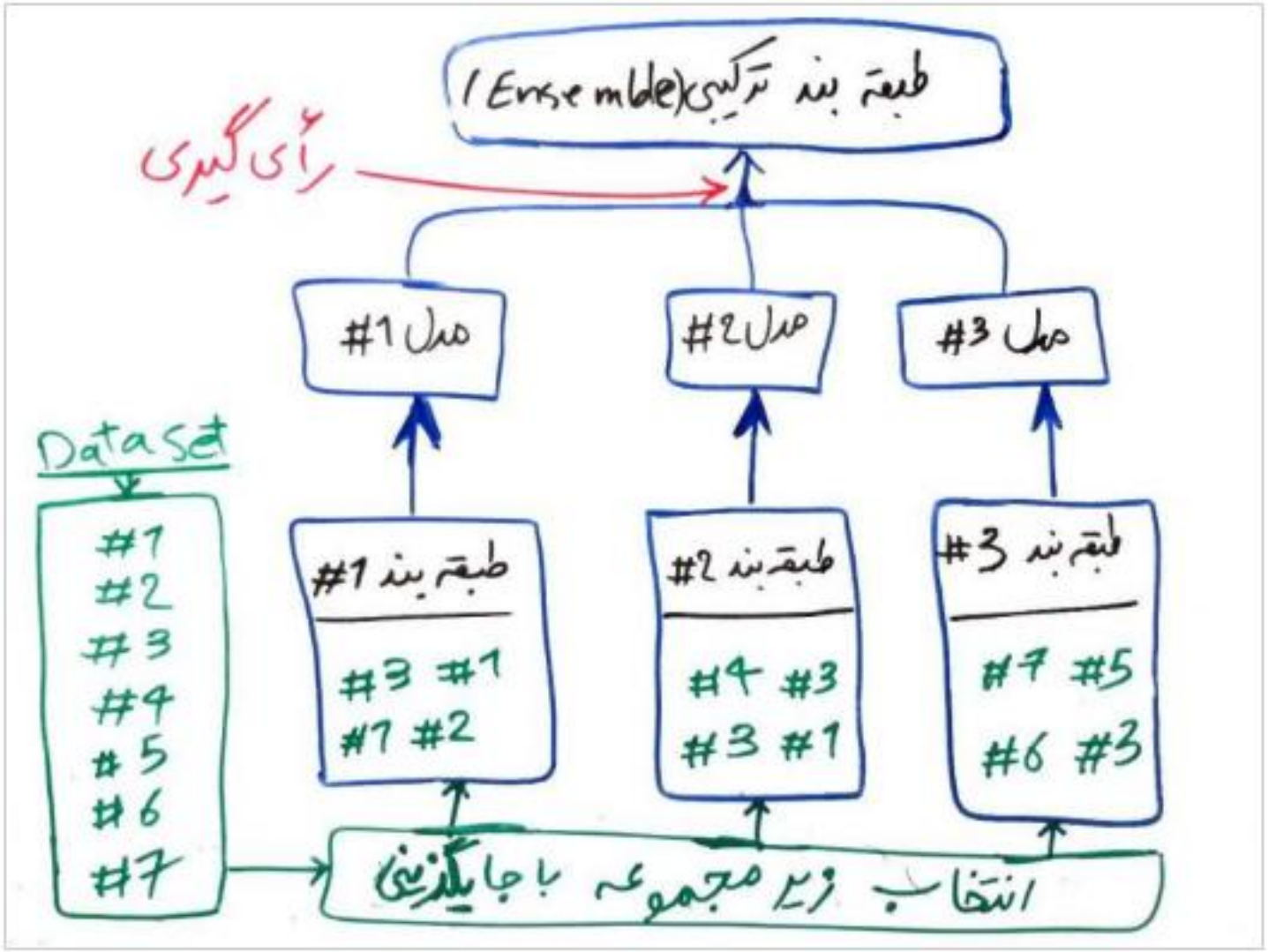
طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting

- در شکل بالا، از سه الگوریتم پایه به نام‌های (KNNنزدیک ترین همسایه)، درخت تصمیم C4.5و شبکه‌های عصبی استفاده شده است. هر کدام از آن‌ها از روی مجموعه‌ی داده، یاد گرفته و مدل خود را می‌سازند. فرض کنید یک مجموعه داده از دو نمونه الف و ب را به این سه الگوریتم داده‌ایم و هر کدام از این الگوریتم‌ها، مدل خود را بر روی این مجموعه داده ساخته‌اند. حال در این مثال یک نمونه‌ی جدید که نمی‌دانیم الف است یا ب به این سه الگوریتم داده می‌شود و دو مدل ۱# و ۲# این نمونه را الف طبقه بندی می‌کنند، این در حالی است که مدل ۳# این نمونه را ب طبقه‌بندی می کند. پس الگوریتم ترکیبی‌نهایی، این مدل را بر اساس نظر اکثریت (در اینجا ۲ به ۱)، در نهایت الف طبقه‌بندی می‌کند.

طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting

- همانطور که در شکل بالا مشاهده کردید، هر کدام از طبقه‌بندها، به مجموعه‌ی داده(Dataset) دسترسی دارند. در روش Bagging یک زیر مجموعه از مجموعه داده‌ی اصلی به هر کدام از طبقه‌بندها داده می‌شود. یعنی هر طبقه‌بند یک قسمت از مجموعه‌ی داده را مشاهده کرده و باید مدل خود را بر اساس همان قسمت از داده‌ها که در اختیارش قرار گرفته است، بسازد(یعنی کلِ دیتاست به هر کدام از طبقه‌بندها داده نمی‌شود). برای مثال شکل زیر را نگاهی بیندازید:

طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting



طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting

- در این شکل، مجموعه داده‌ی ما دارای ۷نمونه است. برای هر کدام از طبقه‌بندها، یک زیرمجموعه از داده های اصلی انتخاب می شود. انتخاب این زیر مجموعه با جایگزینی خواهد بود. یعنی یک نمونه می‌تواند چند بار هم انتخاب شود. برایَ مثال به طبقه‌بند شماره ۱# نمونه های ۳، ۱، ۱ و ۲ داده شده است. همان طور که می‌بینید نمونه ۱ دو مرتبه به طبقه‌بند شماره‌ی ۱# داده شده است. طبق روال، هر طبقه بند با استفاده از داده‌هایی که در اختیار دارد یَک مدل می‌سازد و بقیه‌ی کار مانند مثالِ قبل انجام می‌شود.
- تحقیقاتْ نشان داده است که روشِ Baggingبرای الگوریتم‌هایی مانند شبکه‌های عصبی یا درخت‌های تصمیم که به با تغییرِ کمِ نمونه‌ها ممکن است طبقه‌های مختلفَی ایجاد کنند(این الگوریتم ها به الگوریتم‌های غیرثابت(Unstableنیز خوانده می‌شوند) می‌تواند مفید باشد.

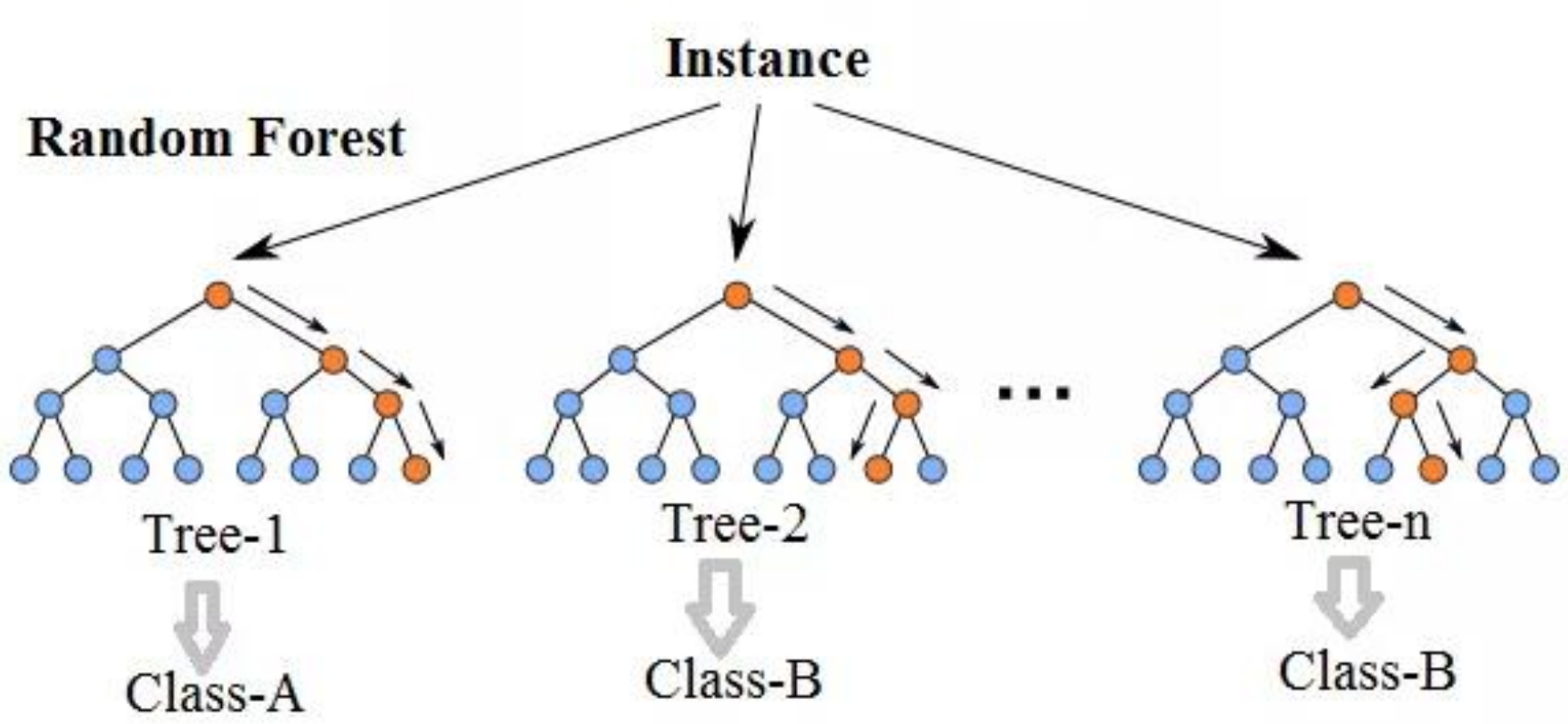
طبقه بند ترکیبی(Ensemble Classifierو مبحث Baggingو Boosting

- مثالی برای توضیحِ روش)Boostingتقویتی)
- فرض کنید شما یک سری نمونه سوال امتحانی دارید. از ابتدا آن ها را همراه پاسخ داده شده می‌خوانید و یاد می‌گیرید. در حالِ خَواندنْ آن‌هایی که برایتان مشکل‌تر است را عَلامت‌گذاری می‌کنید تا بعدا دوباره مرور کنید.
- همین روش هم در Boosting به کار گرفته می‌شود. در این روشِ طبقه‌بندِ اول که در مثالِ بالا طبقه‌بندیِ KNN با شماره ۱# بود، یک نمونه از داده‌ها را(مانند روَش Bagging)مشاهده مَی‌کند و طبقه‌بندَ خود را می‌سازد. سپس از همان نمونه‌های مجموعه‌ی آموزشی به او داده می‌شود و این طبقه‌بند احتمالاً برخی از نمونه ها را اشتباه ارزیابی می‌کند.
- حال برای انتخاب زیر مجموعه‌ی داده‌ها برای طبقه‌بند دوم(#۲) در روش Boosting، آن نمونه‌هایی که طبقَه‌بندِ اول نتوانسته است به درستی طبقه‌بندی کند، با احتمالِ بیشتری برای طبقه‌بندِ دوم انتخاب مَی‌شود. در واقع نمونه‌های سخت‌تر احتمال انتخاب بیشترَی دارند تا آن‌ها که ساده‌تر هستند. به همین ترتیب برای ایجاد یک زیر مجموعه‌ی داده برای طبقه‌بندِ سوم، آن‌هایی که در طبقه‌بندهای اول و دوم مشکل‌تر به نظر می‌رسیدند، با احتمال بیشتری انتخابَ می‌شوند.
- طبقه‌بندهای ترکیبی عموماً از بیش برازش شدنِ مدل یادگرفته شده توسط الگوریتمْ جلوگیری می‌کنند و در بسیاری از موارد نتایج بهتری نسبتَ به الگوریتم‌های دیگر تولَید می‌کنند.

الگوریتم جنگل تصادفی (Random Forest)

- الگوریتمِ جنگلِ تصادفی یا همان Random Forestهم یک الگوریتمِ ترکیبی(Ensemَble)می‌باشد که از درخت‌های تصمیم، برای الگوریتم‌های ساده و ضعیفِ خود استفاده می‌کند. حتما درس درخت‌های تصمیم(Decision Trees)را مطالعهَ کرده‌اید و می‌دانید که یک الگوریتمِ درختِ تصمیم، می‌تواند به راحتی عملیاتِ طبقه بندی را بر روی داده‌ها انجام دَهد. حَال در الگوریتمِ جنگل تصادفی از چندینِ درخت تصمیم(برای مثال ۱۰۰ درخت تصمیم) استفاده می‌شَود. در واقع مجموعه‌ای از درخت‌های تصمیم، با هم یک جنگل را تولید می‌کنند و این جنگل می‌تواند تصمیم‌های بهتری را(نسبت به یک درخت) اتخاذ نماید.
- در الگوریتم جنگل تصادفی به هر کدام از درخت‌ها، یک زیرمجموعه‌ای از دادگان داده می‌شود. برای مثال اگر دیتاست شما دارای ۱۰۰۰سطر(۱۰۰۰نمونه) و ۵۰ستون(یعنی ۵۰ ویژگی) بود(درس ویژگی‌ها و ابعاد را خوانده باشید)، الگوریتمِ جنگلِ تصادفی به هر کدام از درخت‌ها، ۱۰۰سطر و ۲۰ستون(که به صورت تصادفی انتخابَ شده‌اند) که زیر مجموعه‌ای از مجموعه‌ی دادگان هست، می‌دهد. این درخت‌ها با همین دیتاستِ زیر مجموعه، می‌توانند تصمیم بگیرند و مدلِ طبقه‌بندِ خود را بسازند. برای نمونه شکَل زیر را در نظر بگیرید:

رندوم فارست

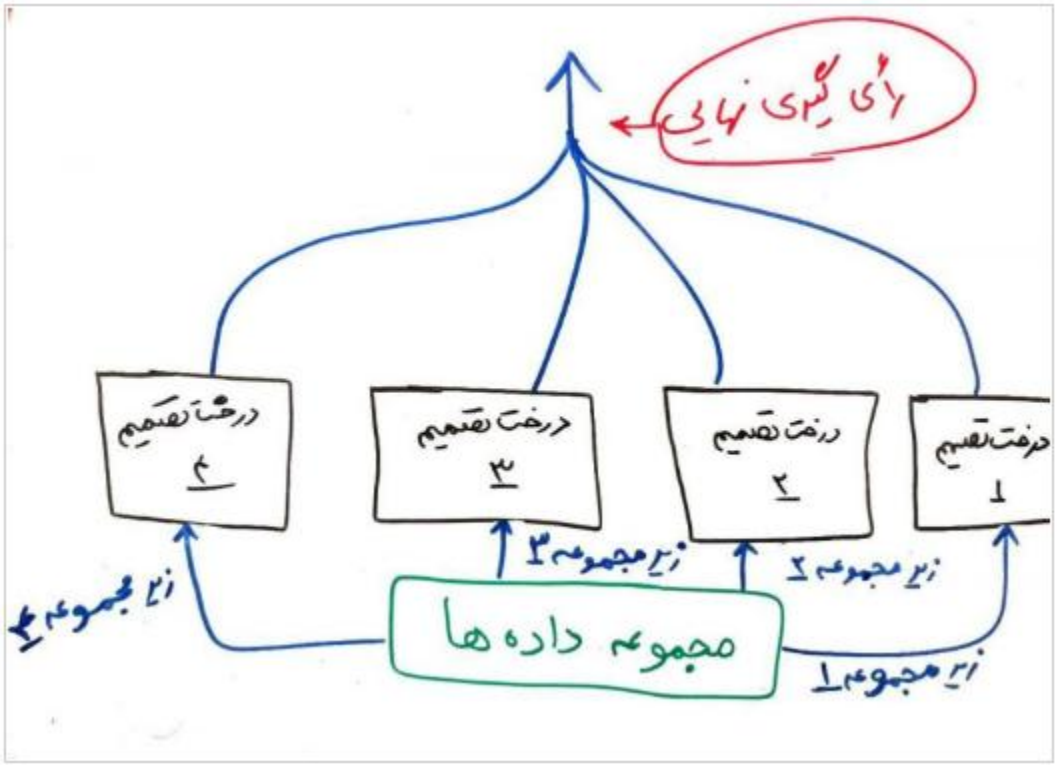

Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B

رندوم فارست

Herein, random forest is a new algorithm derived from decision trees. Instead of applying decision tree algorithm on all dataset, dataset would be seperated into subsets and same decision tree algorithm would be applied to these subsets. Decision would be made by the highest number of subset results.

- So, why traditional decision tree algorithm evolved into random forests? Working on all dataset may cause to overfitting. In other words, it might cause memorizing instead of learning. In this case, you would have high accuracy on training set, but you would fail on newly instances.
- What's more, random forests work on multiple and small datasets. Increasing the dataset size would increase the learning time exponentially in decision tree. So, you can parallelize the learning procedure in random forest. In this way, learning time would last less than decision tree.
- If decision tree algorithm were the wise / sage person of around you, then random forests would be multiple smart people. Wise one might know every domain but each smart person can be expert on different domains. Wise one would most probably respond the correct answer but you might not always have a opportunity to ask him. However, bringing smart people to gather would most probably be acceptable.

طبقه بند ترکیبی(Ensemble Classifier)و مبحث Baggingو Boosting



- همان‌طور که مشاهده می‌شود، مجموعه‌ای از درخت‌های تصمیم وجود دارند که به هر کدام از آن‌ها یک زیر مجموعه‌ای از دادگان داده می‌شود.
- هر کدام از الگوریتم‌ها عملیاتِ یادگیری را انجام می‌دهند.
- در هنگام پیش‌بینی، یعنی وقتی که یک سریِ داده‌ی جدید به الگوریتمْ جهت پیش‌بینی دادَه می‌شود، هر کدام از این الگوریتم‌های یادگرفته شده، یک نتیجه را جهت پیش‌بینی برمی‌گردانند. الگوریتمِ جنگلِ تصادفی در نهایت، می‌تواند با استفادَه از رَای‌گیری، آن طبقه‌ای را که بیشترین رای را آورده است انتخاب کند و به عنوانِ طبقه‌ی نهایی جهت انجامِ عملیات طبقه‌بندیَ قرار دهد.

فصل چهارم

ارزیابی کلاس بندی

بررسی معیارهای سنجش دسته‌بندی

- به این جدول مهم که اساس تحلیل و ارزیابی کارآیی یک مدل در مباحث دسته‌بندی است، ماتریس درهم‌ریختگی یا اغتشاش گفته می-شود

		مقادیر واقعی	
		مثبت	منفی
مقادیر پیش بینی شده	مثبت	درست مثبت TP	نادرست مثبت FP خطای نوع یک
	منفی	نادرست منفی FN خطای نوع دو	درست منفی TN

بررسی معیارهای سنجش دسته‌بندی

- با فرض اینکه هدف ما پیش‌بینی دیابت یک بیمار باشد، اگر پیش‌بینی مثبت باشد یعنی بیمار، مبتلا به دیابت است و اگر پیش بینی منفی باشد، یعنی بیمار به دیابت مبتلا نیست، به تحلیل سلول‌های این ماتریس میپردازیم:
- درست مثبت : (True Positive-TP) اگر بیمار واقعا دیابت داشته باشد و مقدار پیش‌بینی شده هم دیابت را نشان دهد.
- نادرست مثبت : (FP) اگر بیمار دیابت نداشته باشد اما نتیجه پیش بینی ما، نشانگر دیابت بیمار باشد.
- نادرست منفی : (FN) اگر بیمار دیابت داشته باشد اما پیش بینی ما، دیابت را منفی نشان دهد.
- درست منفی : (TN) اگر بیمار دیابت نداشته باشد و پیش بینی ما هم همین را نشان بدهد.
- حالت ایده آل این است که موارد نادرست (نادرست مثبت و نادرست منفی) صفر باشند اما در عمل این اتفاق نمی‌افتد.

بررسی معیارهای سنجش دسته‌بندی

دقت - صحت - بازخوانی

- اولین معیار، معیار دقت یا میزان تشخیص درست مدل است. یعنی نسبت تشخیص های درست (TP+TN)به کل داده ها:

$$\text{دقت (Accuracy)} = \frac{TP+TN}{N} = \frac{\text{تشخیص‌های درست}}{\text{کل داده‌ها}}$$

- به عنوان یک مثال:

n=165	پیش بینی منفی	پیش‌بینی مثبت	
واقعی منفی	TN = 50	FP = 10	60
واقعی مثبت	FN = 5	TP = 100	105
	55	110	

بررسی معیارهای سنجش دسته‌بندی

- همانطور که مشاهده می کنید در مثال فوق که یک جمعیت ۱۶۵ تایی از داده‌ها را شامل می‌شود، تعداد داده‌های درست تشخیص داده شده یعنی آنهایی که در واقعیت و در پیش بینی، یک مقدار داشته اند (ناحیه سبزرنگ)، به نسبت سایر داده ها، بسیار بزرگ‌تر است بنابراین انتظار می رود دقت این مدل ما بالا باشد. آنرا محاسبه می کنیم:

$$\text{دقت} = \frac{۱۰۰+۵۰}{۱۶۵} = ۰.۹۱$$

- این عدد دقت بسیار خوبی است. برای بسیاری از مسائل دسته‌بندی دنیای واقعی این معیار، بسیار کارآمد است چون هم داده‌های در نظر نگرفته شده را لحاظ کرده است (مخرج کسر) و هم داده‌های شناسایی شده را (صورت کسر).

بررسی معیارهای سنجش دسته‌بندی

- فرض کنید هزار نمونه آزمایش خون داریم که تنها دو نفر از آنها به ایدز مبتلا هستند. می خواهیم مدلی (الگوریتم) پیشنهاد کنیم افراد دارای ایدز و افراد سالم را شناسایی کند. این مدل اگر نمونه ای را مثبت اعلام کرد، آن فرد به ایدز مبتلاست.
- با این مدل ما ۹۹۸ نمونه منفی درست تشخیص داده شده داریم (TN = 998) و نمونه مثبت هم کلا نداریم (TP=0) بنابراین طبق فرمول فوق، دقت الگوریتم پیشنهادی برابر است با

$$\text{دقت مدل پیشنهادی برای تشخیص ایدز} = \frac{۹۹۸+۰}{۱۰۰۰} = ۰,۹۹۸$$

بررسی معیارهای سنجش دسته‌بندی

- مشکل اصلی هم نامتعادل بودن داده ها و تفاوت معنی دار تعداد نمونه های هر دسته است که باعث می‌شود یک مدل متمایل به دسته پرتعداد، دقت کلی را بالا نشان دهد. بنابراین نیاز به معیاری دقیق تر برای سنجش دقت و کارآیی الگوریتم های پیشنهادی دسته بندی هستیم.
- در این گونه مسایل بهتر است بر تعداد نمونه های مثبت شناسایی شده به کل نمونه های مثبت تمرکز کنیم.

$$\text{بازخوانی (Recall)} = \frac{\text{تعداد های نمونه تشخیص درست مثبت}}{\text{کل های نمونه واقعاً مثبت}} = \frac{TP}{TP+FN}$$

- توضیح اینکه کل نمونه های واقعاً مثبت شامل نمونه هایی است که درست، مثبت شناسایی شده اند (TP)و نمونه هایی که مثبت بوده اند اما نادرست، منفی شناسایی شده اند.(FN)
- سنجه بازخوانی برای روش پیشنهادی صفر است (چون هیچ نمونه مثبتی را شناسایی نکرده ایم - صورت کسر برابر صفر است) که نشانگر ضعیف بودن مدل پیشنهادی است و بنابراین آنرا می توانیم به راحتی رد کنیم.

بررسی معیارهای سنجش دسته‌بندی

- در کنار معیار بازخوانی معیار دیگری را به نام صحت(Precision) ، برابر تعداد نمونه های تشخیصی درست مثبت به کل نمونه های مثبت اعلام شده به صورت زیر تعریف می کنیم تا میزان مثبت های اشتباه را هم در نظر گرفته باشیم:

$$\text{صحت (Precision)} = \frac{\text{تعداد های نمونه تشخیصی درست مثبت}}{\text{تعداد کل های نمونه تشخیصی مثبت}} = \frac{TP}{TP+FP}$$

بررسی معیارهای سنجش دسته‌بندی

- اگر بتوانیم معیاری ترکیبی از این دو معیار برای سنجش الگوریتم های دسته‌بندی به دست آوریم، تمرکز بر آن معیار به جای بررسی همزمان این دو، مناسب‌تر خواهد بود.
- اگر بخواهیم میانگین معمولی دو معیار بازخوانی و صحت را ملاک کار درنظر بگیریم، برای حالت هایی که صحت بالا و بازخوانی پایینی داریم (و یا بالعکس)، میانگین معمولی عددی قابل قبول خواهد بود در صورتی که الگوریتم پیشنهادی نباید نمره قبولی بگیرد. برای رفع این نقیصه و تولید یک معیار واحد که متمایل به عدد کوچکتر باشد، از میانگین هارمونیک و با فرمول زیر استفاده می کنیم:

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} 1/x_i}, \qquad x_i > 0 \text{ for all } i.$$

بررسی معیارهای سنجش دسته‌بندی

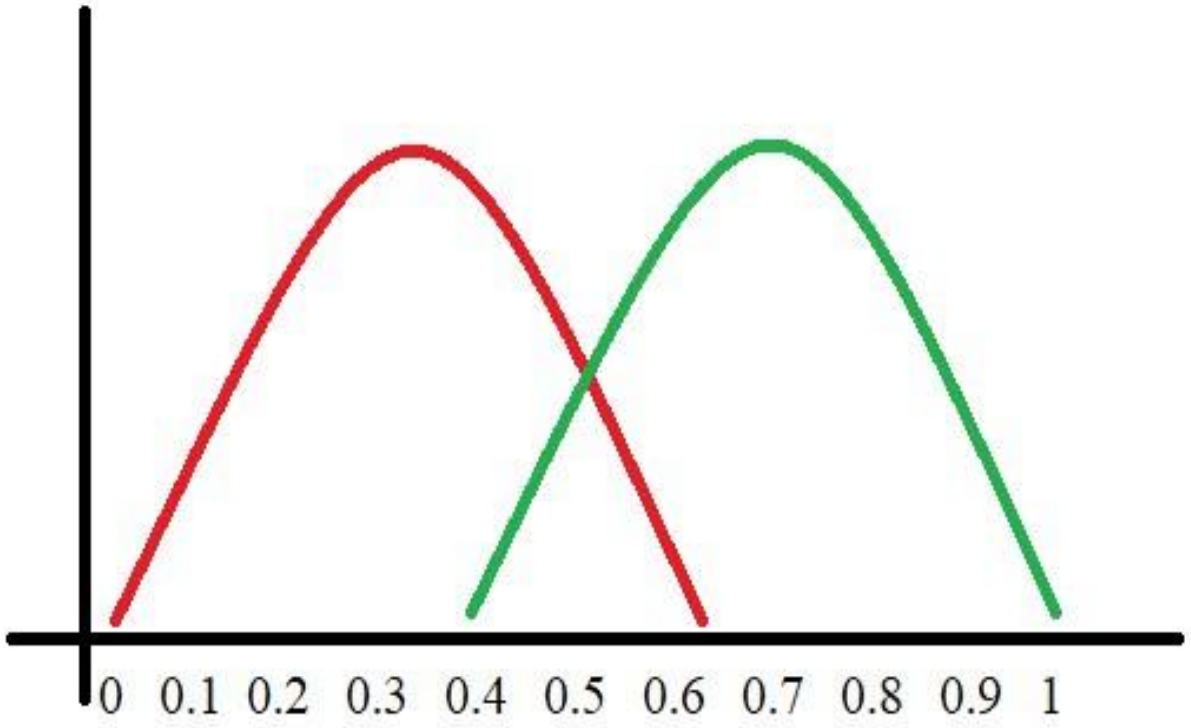
- این میانگین هارمونی برای دو مقدار بازخوانی و صحت را با نام F1-Scoreمی شناسیم که طبق فرمول فوق برابر است با:

$$\text{F1-Score} = \frac{۲}{\frac{۱}{Recall} + \frac{۱}{Precision}} = ۲ \times \frac{Precision * Recall}{Precision + Recall}$$

- در این فرمول، اگر یکی از دو مقدار عددی کوچک باشد، یا حتی صفر باشد، نتیجه نهایی عددی کوچک و یا صفر خواهد بود. چون دو معیار بازخوانی و صحت اعدادی بین صفر تا یک هستند و در صورت کسر در هم دیگر ضرب شده اند بنابراین نتیجه نهایی به سمت عدد کوچکتر، متمایل خواهد بود و اگر هر دو با هم، عددی بزرگ (نزدیک ۱) باشند، نتیجه نهایی به سمت یک حرکت خواهد کرد

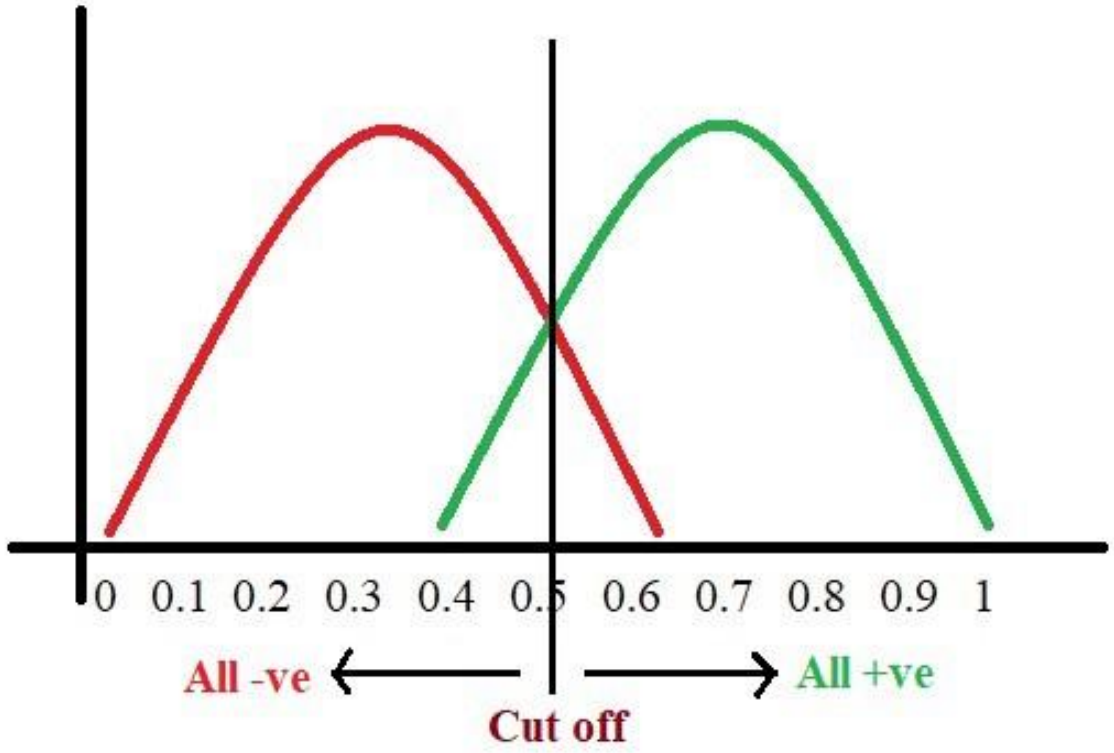
بررسی معیارهای سنجش دسته‌بندی

- مدل به هر فرد احتمالی بین ۰ تا ۱ اختصاص می‌دهد که با توجه به آن، بیمار بودن یا سالم بودن شخص را حدس خواهیم زد.
- اگر نمودار توزیع این احتمال را بر اساس درصد احتمال به عضویت در گروه بیماران یا افراد سالم رسم کنیم به نمودار ساده زیر می رسیم که در آن نمودار سبزرنگ، احتمال بیمار بودن و نمودار قرمز رنگ، احتمال سالم بودن یک شخص را نشان می‌دهد.



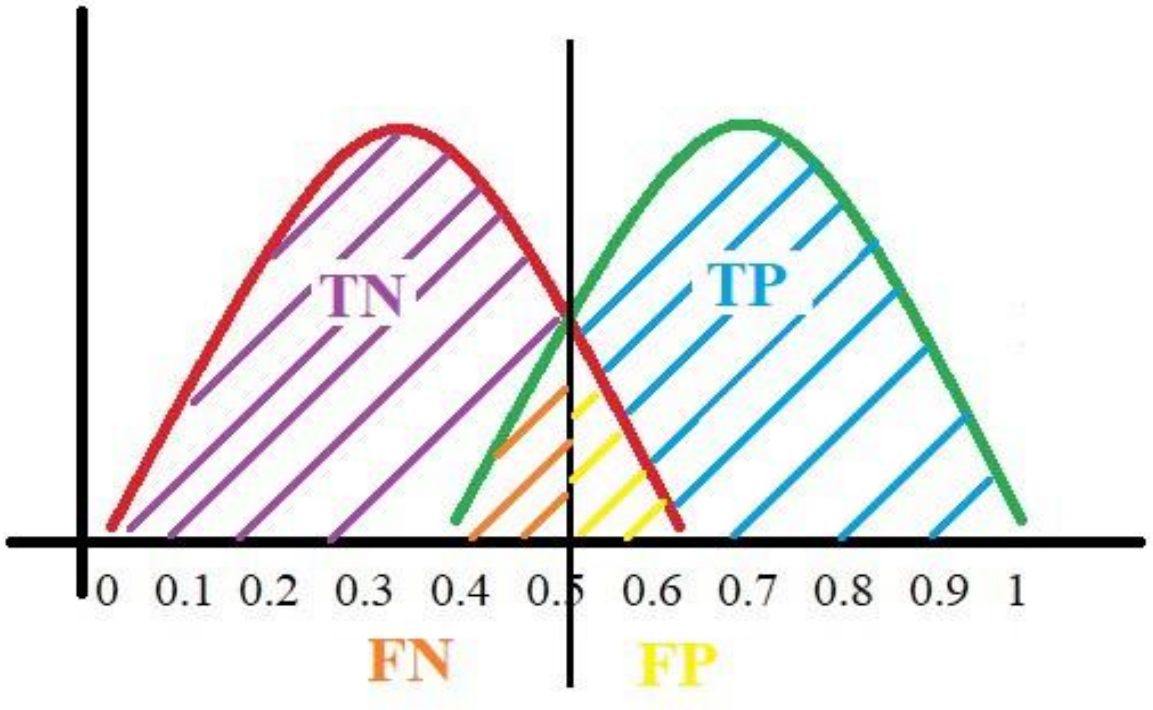
بررسی معیارهای سنجش دسته‌بندی

- اگر خروجی مدل ما زیر عدد ۰,۴ باشد، شخص مورد نظر قطعا سالم است و اگر عدد خروجی مدل ما بالای ۰,۶ باشد، نشان دهنده بیمار بودن شخص است اما اگر عددی بین این دو تولید شد، مثلاً عدد ۰,۵ ، با قطعیت نمی‌توانیم بیان کنیم که شخص بررسی شده، سالم است یا نه. اگر بین ۰,۴ تا ۰,۵ باشد، احتمال سالم بودن شخص بیشتر است و اگر بین ۰,۵ تا ۰,۶ باشد، احتمال بیمار بودن شخص ،قوت می‌گیرد که این امر، باعث می‌شود دقت مدل کمی پایین بیاید و ناخواسته، نتایج اشتباهی حاصل شود.



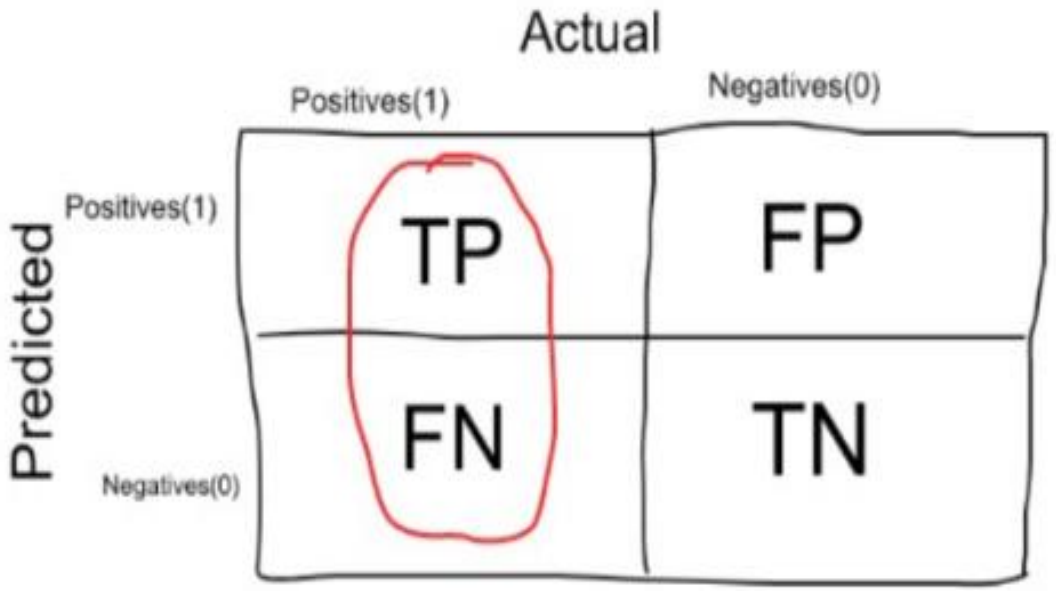
بررسی معیارهای سنجش دسته‌بندی

- تعیین این نقطه عدد ۰٫۵ و در مثالهای واقعی کاملا بسته به شرایط عددی بین ۰ تا ۱ است.
- ناحیه زرد رنگ بیانگر افرادی است که اشتباهاً بیمار تشخیص داده خواهند شد (False Positive) نادرست مثبت (و ناحیه نارنجی رنگ هم بیانگر افرادی است که به اشتباه سالم تشخیص داده شده اند) نادرست منفی.(False Negative)
- هر چه مدل ما دقیق‌تر باشد، این دو خط قرمز و سبز باید اشتراک کمتری داشته باشند یعنی بتوانیم با قطعیت بیشتری دسته‌بندی داده‌ها را انجام دهیم.



بررسی معیارهای سنجش دسته‌بندی

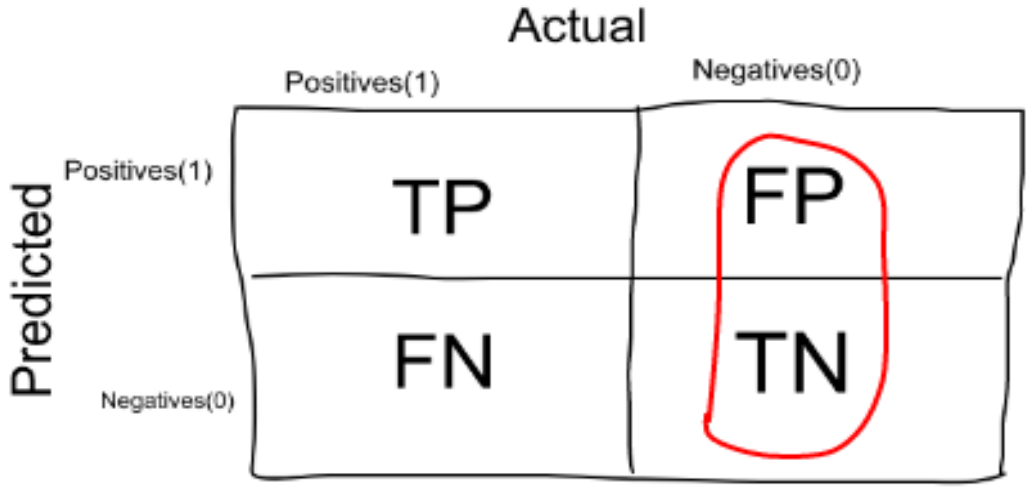
- **انتخاب درست نقطه تقسیم یا تعیین آستانه تقسیم در یک مدل، تصمیم مهمی است چون تغییر آن باعث افزایش یا کاهش خطا خواهد شد. برای سنجش خطاهای تولید شده، دو معیار** Sensitivity (Recall) **و** Specificity **را به صورت زیر تعریف می کنیم:**



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

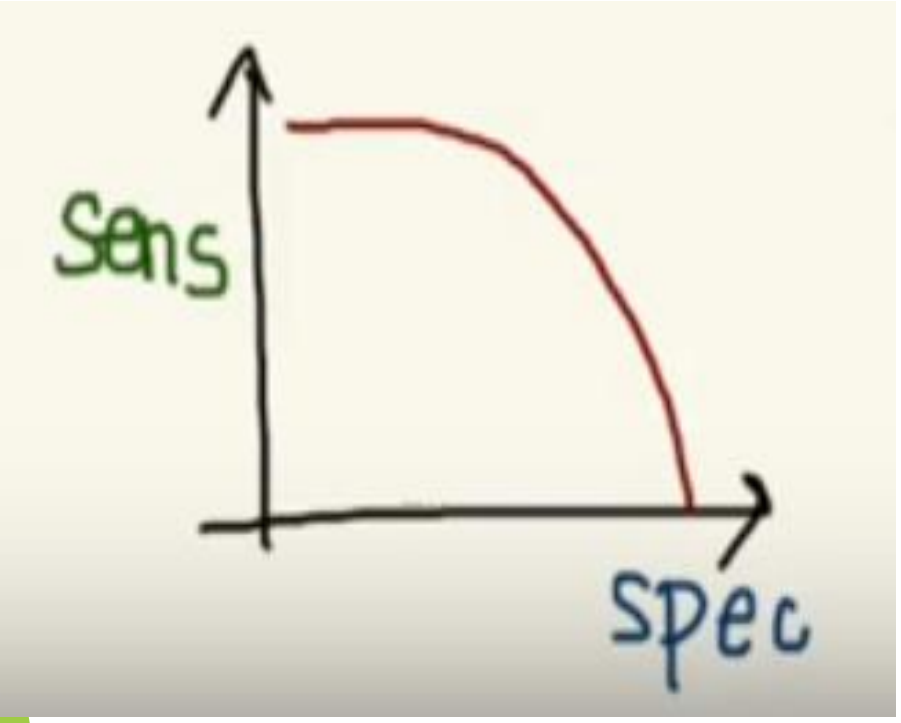
بررسی معیارهای سنجش دسته‌بندی

- معیار بازخوانی یا همان Sensitivity (حساسیت) نشان می‌دهد چقدر از بیماران واقعی (دسته مثبت) را نسبت به کل جامعه بیماران، شناسایی کرده‌ایم. یعنی نسبت آنهایی که درست شناسایی شده‌اند به مجموع تمام بیماران (آنهایی که به درستی بیمار شناخته شده اند + آنهایی که اشتباهاً سالم تشخیص داده شده‌اند). هدف ما این است که حساسیت مدل ما بالا باشد یعنی تعداد بیشتری از بیماران را شناسایی کند. معیار Specificity همین مفهوم را برای افراد سالم (یا دسته منفی) نشان می‌دهد یعنی چند نفر از افراد واقعا سالم را از کل افراد سالم، درست تشخیص داده‌ایم:



$$\text{Specificity} = \frac{TN}{TN + FP}$$

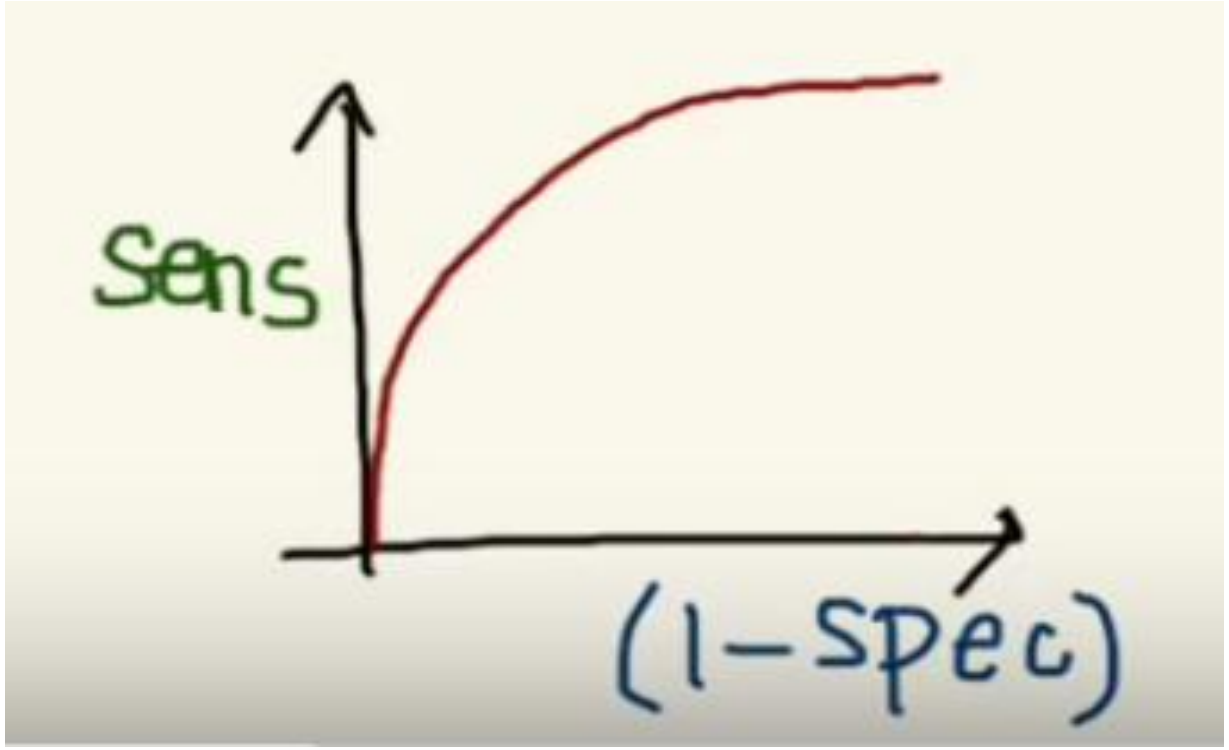
بررسی معیارهای سنجش دسته‌بندی



- میزان افرادی که بیمار نیستند، درست منفی (TN) به کل افراد سالم (آنهایی که سالم تشخیص داده شده‌اند و آنهایی که اشتباهاً بیمار فرض شده‌اند)، Specificity مدل را تشکیل می‌دهد.
- اگر حد آستانه را روی ۰٫۴ تنظیم کنیم و بالاتر از آنرا بیمار اعلام کنیم، طبق شکل متوجه می‌شویم که تمام بیماران را تشخیص خواهیم داد یعنی حساسیت مدل بالاست اما میزان زیادی از افراد سالم را هم بیمار اعلام خواهیم کرد یعنی Specificity ما پایین خواهد آمد.
- بالعکس اگر حد آستانه را بالا ببریم، مثلا آنرا روی ۰٫۶ تنظیم کنیم، تمام افراد سالم را درست تشخیص خواهیم داد اما بیماران زیادی را هم به اشتباه، سالم اعلام خواهیم کرد یعنی Specificity مدل بالا و حساسیت آن کم خواهد شد. با تغییر این آستانه به شکل زیر برای بیان نسبت میان حساسیت و Specificity خواهیم رسید

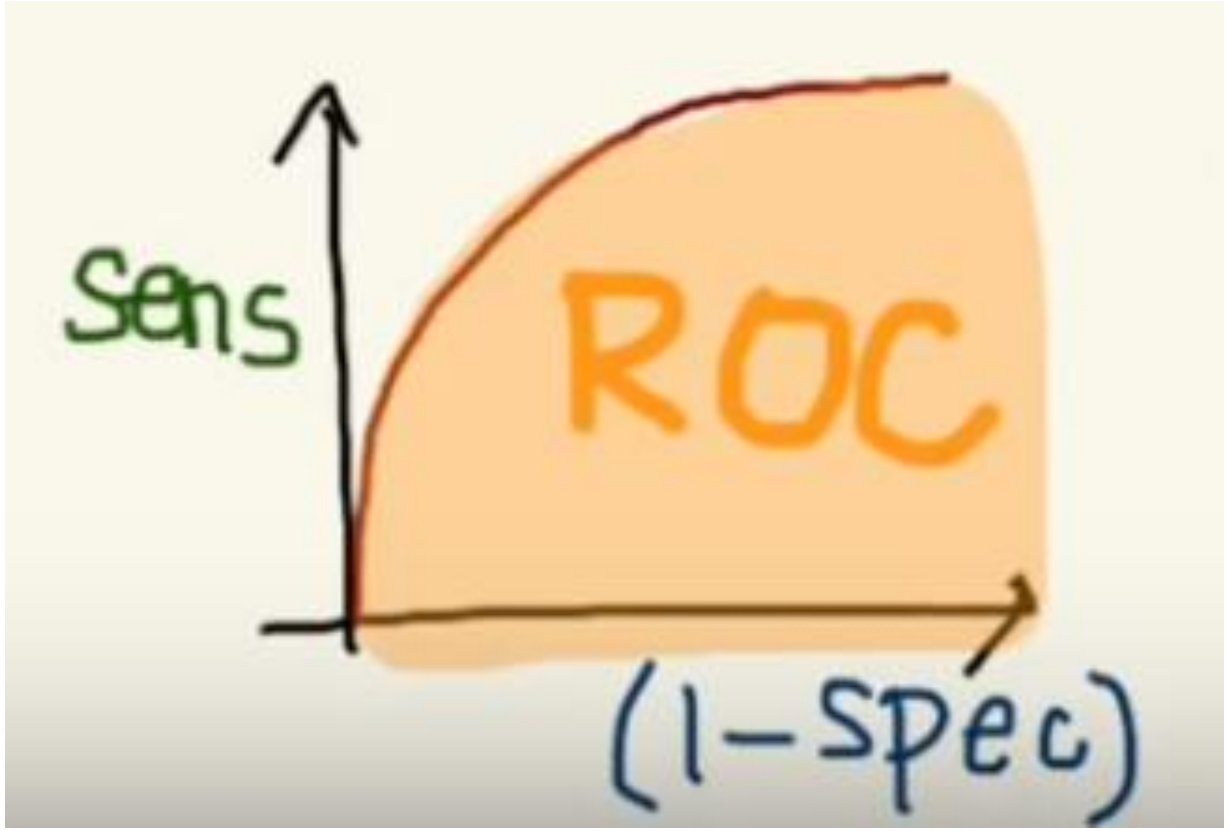
بررسی معیارهای سنجش دسته‌بندی

- **برای اینکه بهتر بتوانیم از این نمودار استفاده کنیم و مقادیر هر دو محور با هم رشد یا کاهش پیدا کنند به جای** Specificity **از ۱ منهای** Specificity **استفاده می‌کنیم:**



بررسی معیارهای سنجش دسته‌بندی

با این ترتیب، نموداری حاصل می‌شود که به آن نمودار ROC - Receiver Operating Characteristics **و یا منحنی** ROC **می‌گوییم.**



بررسی معیارهای سنجش دسته‌بندی

- برای داشتن میزان جداکنندگی و دقت کار مدل به روابط زیر می رسیم:
- 1–Specificity: نرخ تولید خطای دسته‌بندی (برای دسته مثبت) را نشان می‌دهد. تعداد افراد سالمی که بیمار تشخیص داده شده‌اند به کل افراد سالم. به این معیار FPR هم گفته می‌شود:

$$Specificity = \frac{TN}{TN + FP}$$

$$1 - Specificity = 1 - \frac{TN}{TN + FP}$$

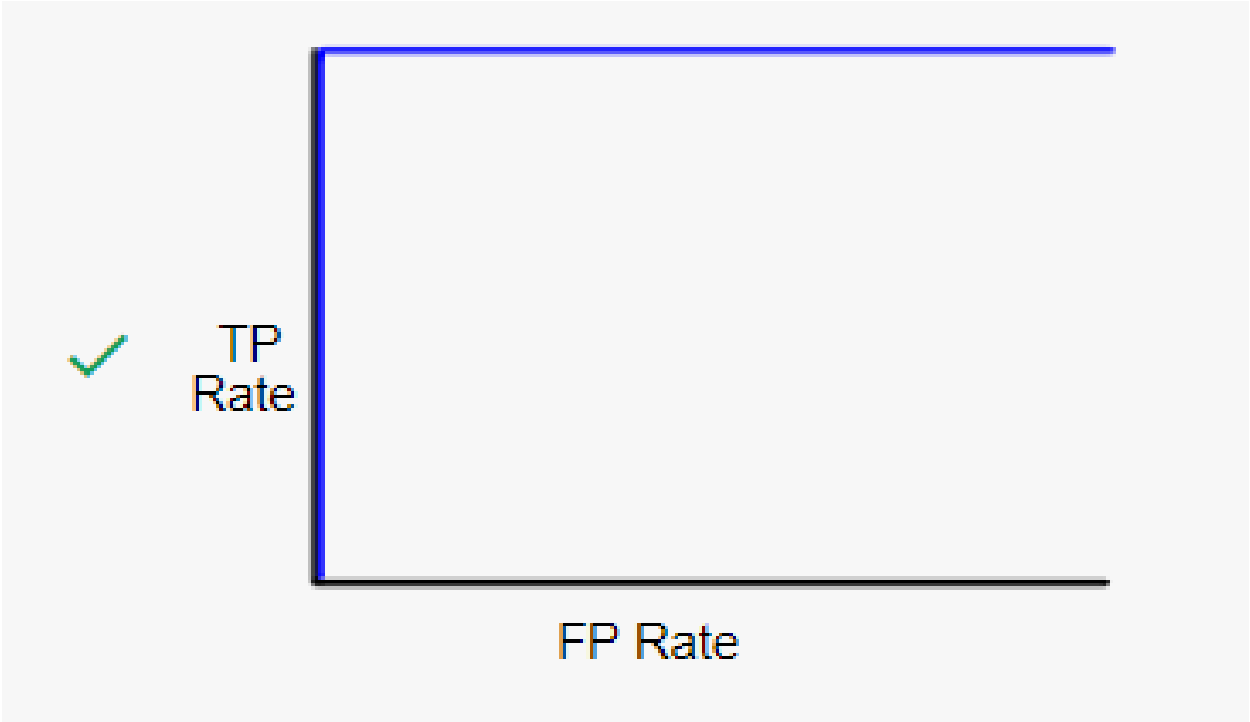
$$1 - Specificity = \frac{TN + FP - TN}{TN + FP}$$

$$1 - Specificity = \frac{FP}{TN + FP}$$

- در نمودار ROC نرخ تولید داده‌های درست یعنیTPR ،محور Y را نشان می‌دهد و نرخ تولید خطا برای داده‌های مثبت هم (FPR) محور X را تشکیل می‌دهد.

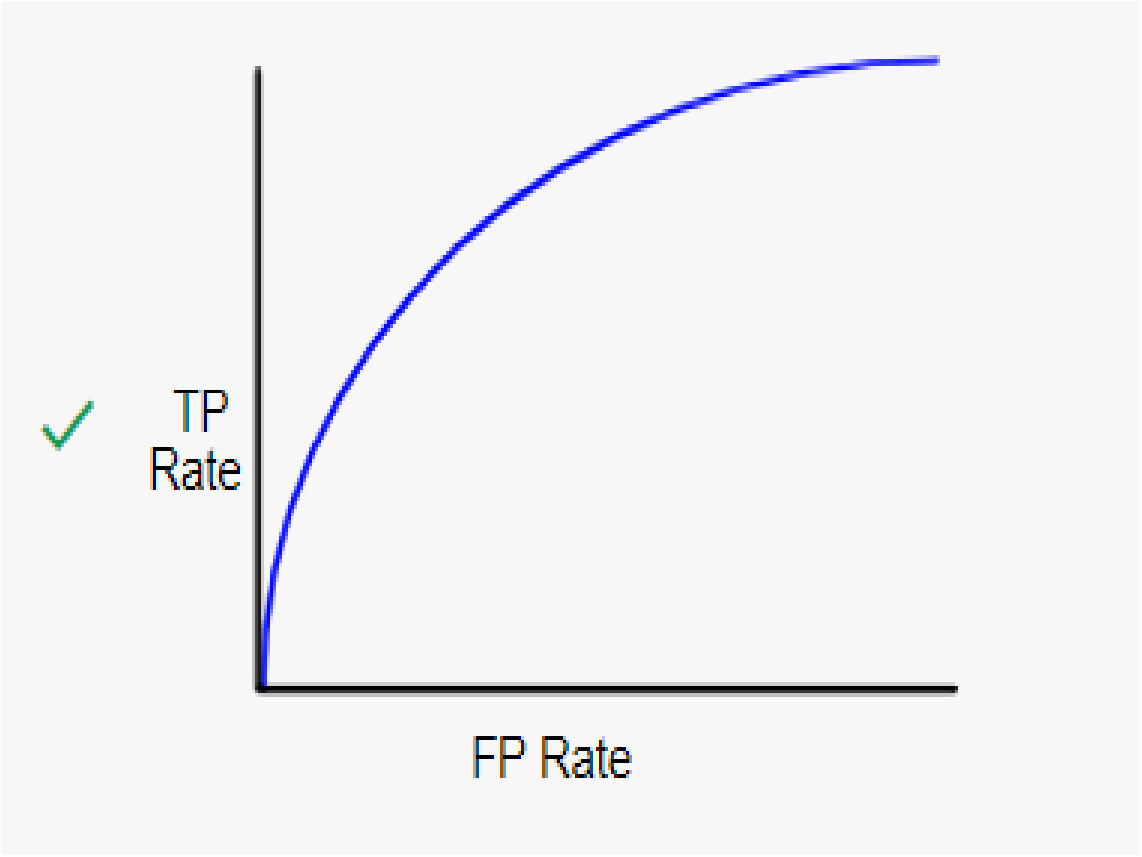
بررسی معیارهای سنجش دسته‌بندی

با این توصیف نموداری مناسب تر خواهد بود که محور Y آن به یک نزدیک باشد و محور X آن یعنی میزان تولید خطای آن، به صفر نزدیک باشد:



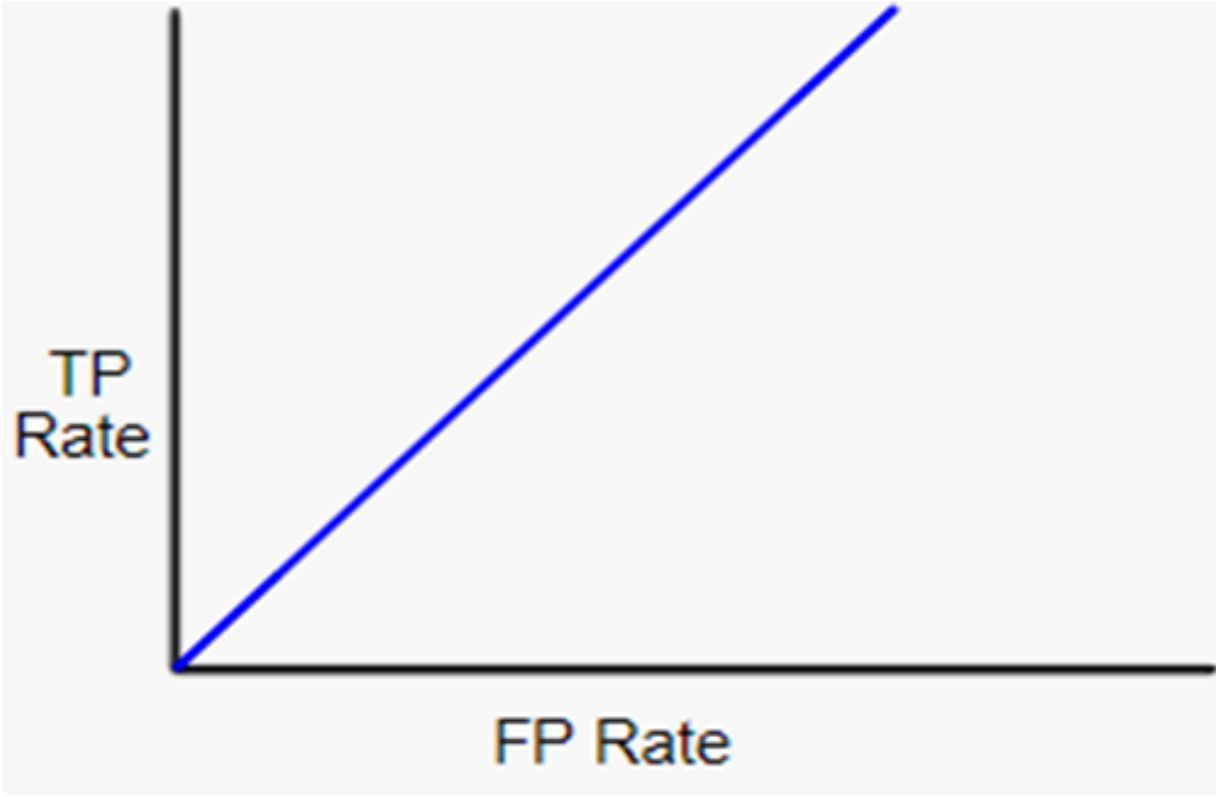
بررسی معیارهای سنجش دسته‌بندی

اما در دنیای واقعی، نمودار ما بیشتر شبیه شکل زیر خواهد بود:



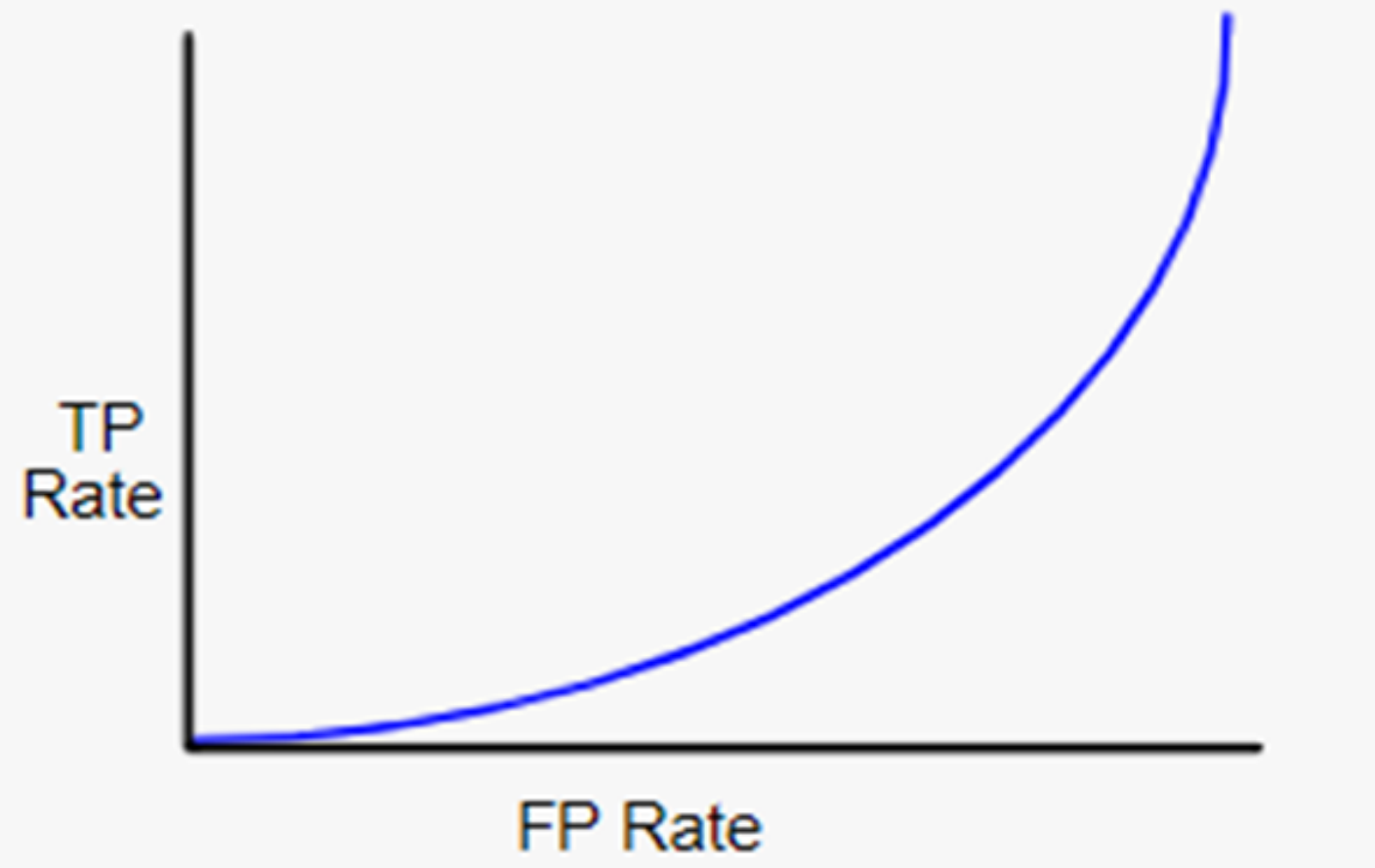
بررسی معیارهای سنجش دسته‌بندی

- که اگر آنرا نسبت به حالت تصادفی یعنی حالتی که کاملا تصادفی اشخاص را به دو دسته بیمار و سالم تقسیم کنیم (نمودار زیر)، بهبود مدل کاملاً مشخص است:



بررسی معیارهای سنجش دسته‌بندی

مطمئنا ایجاد نموداری به شکل زیر نشان دهنده خطای محرز در مدل است چون حتی از حالت تصادفی هم بدتر عمل کرده است:



بررسی معیارهای سنجش دسته‌بندی

نحوه رسم نمودار ROC

- به عنوان مثال، فرض کنید برای صد نفر که نصف آنها بیمار و نصف آنها سالم هستند، مدلی ساخته‌ایم که اگر حد آستانه تشخیص یک دسته از صفر تا یک تغییر بدهیم، اعداد زیر حاصل می‌شوند:

Threshold	TP	FP	TN	FN
0.0	50	50	0	0
0.1	48	47	3	2
0.2	47	40	9	4
0.3	45	31	16	8
0.4	44	23	22	11
0.5	42	16	29	13
0.6	36	12	34	18
0.7	30	11	38	21
0.8	20	4	43	33
0.9	12	3	45	40
1.0	0	0	50	50

بررسی معیارهای سنجش دسته‌بندی

- **نحوه رسم نمودار ROC**
- **برای حد آستانه ۰٫۵ جدول پراکنش زیر را خواهیم داشت:**

Threshold =0.5	Actual Positives	Actual Negatives
Predicted Positives	42 (TP)	16 (FP)
Predicted Negatives	13 (FN)	29 (TN)

بررسی معیارهای سنجش دسته‌بندی

- با اعداد فوق برای حد آستانه ۰٫۵ سه معیار صحت، بازخوانی و F1 را به صورت زیر محاسبه می‌کنیم:

$$recall = \frac{TP}{TP+FN} = \frac{42}{42+13} = 0.76 \quad precision = \frac{TP}{TP+FP} = \frac{42}{42+16} = 0.724 \quad F1\ Score = 2 * \frac{precision*recall}{precision+recall} = 0.74$$

- نرخ تولید خطا و نرخ تولید داده‌های درست هم به صورت زیر محاسبه می‌شود

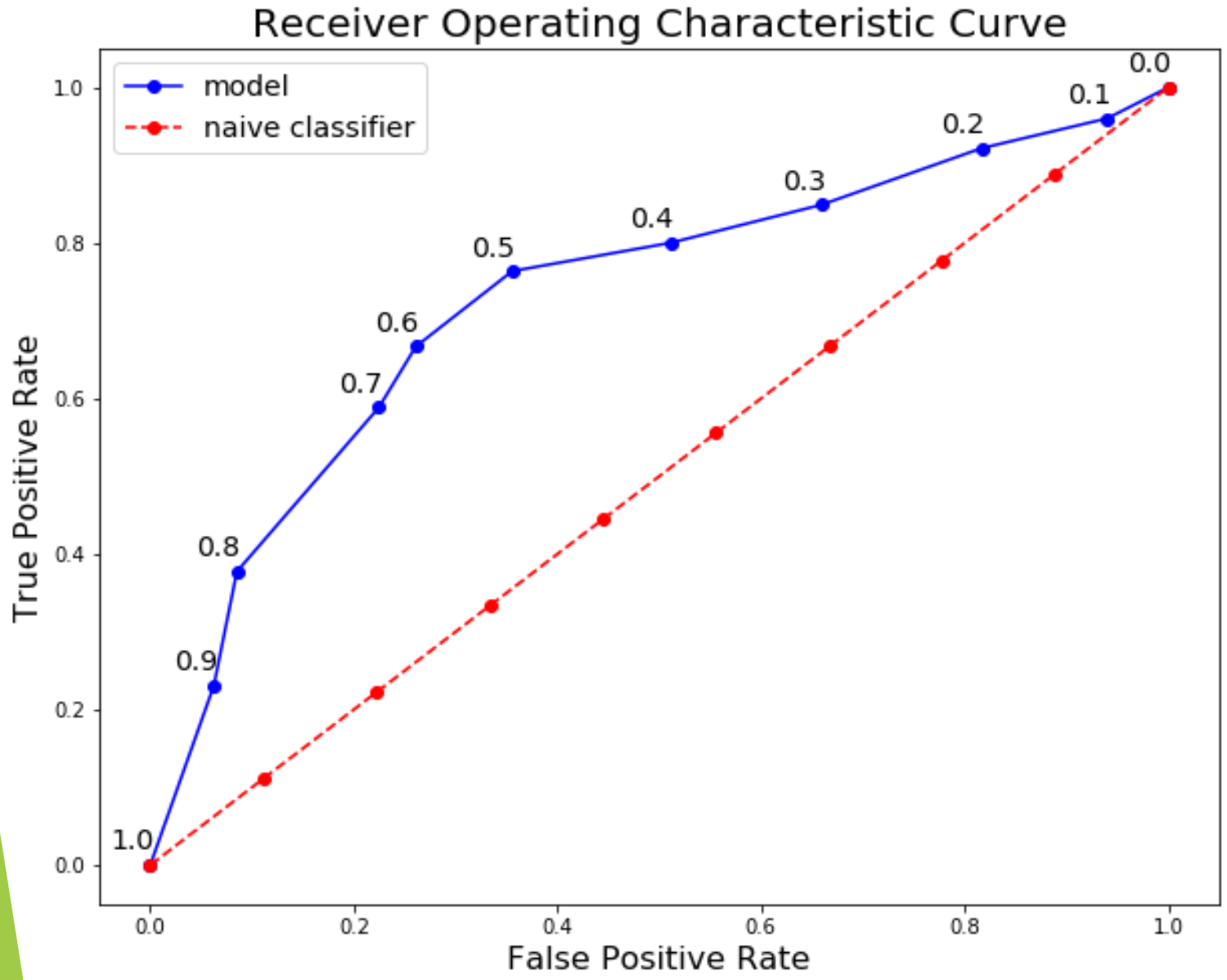
$$true\ positive\ rate = \frac{TP}{TP+FN} = \frac{42}{42+13} = 0.76 \quad false\ positive\ rate = \frac{FP}{FP+TN} = \frac{16}{16+29} = 0.36$$

مقادیر شاخص های سنجش دسته‌بندی

حال با محاسبه این اعداد برای مقادیر مختلف حد آستانه، جدول زیر را خواهیم داشت:

threshold	recall	precision	f1	tpr	fpr
0.0	1	0.5	0.666667	1	1
0.1	0.96	0.505263	0.662069	0.96	0.94
0.2	0.921569	0.54023	0.681159	0.921569	0.816327
0.3	0.849057	0.592105	0.697674	0.849057	0.659574
0.4	0.8	0.656716	0.721311	0.8	0.511111
0.5	0.763636	0.724138	0.743363	0.763636	0.355556
0.6	0.666667	0.75	0.705882	0.666667	0.26087
0.7	0.588235	0.731707	0.652174	0.588235	0.22449
0.8	0.377358	0.833333	0.519481	0.377358	0.0851064
0.9	0.230769	0.8	0.358209	0.230769	0.0625
1.0	0	0	0	0	0

بررسی معیارهای سنجش دسته‌بندی

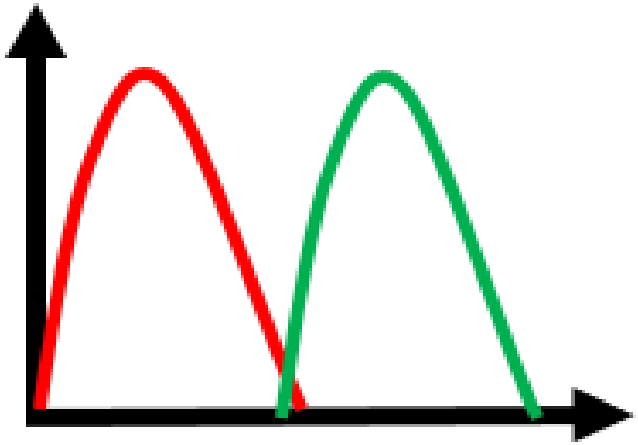


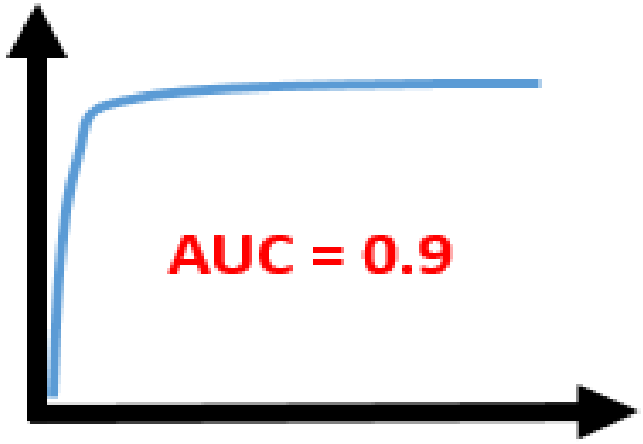
نمودار ROC

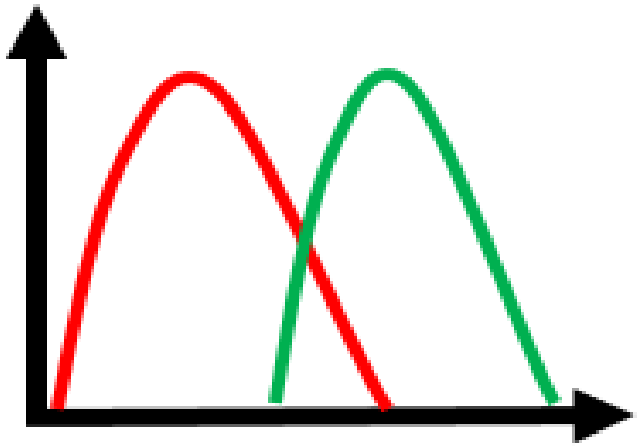
بررسی معیارهای سنجش دسته‌بندی

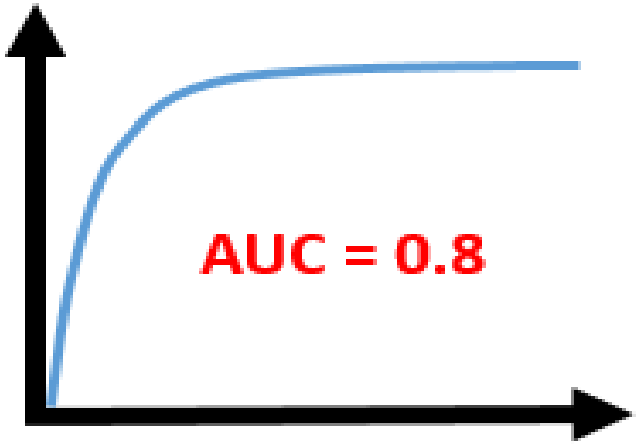
- **سطح زیر نمودار AUC(Area Under Curve)**
- **در حالت ایده‌آل شکل ایده‌آل نمودار (ROC) ، مساحت زیر نمودار عدد یک را نشان می‌دهد و در حالت تصادفی عدد ۰٫۵ و در بیشتر موارد، عددی بین این دو خواهد بود که هر چه به یک نزدیک‌تر باشد نشان از دقت بیشتر مدل ما در تشخیص داده‌های مثبت است.**
- **این مساحت که با معیار AUC نشان داده می‌شود، معیار دیگری است برای سنجش میزان کارآیی یک مدل که هر چه مدل دقیق‌تری داشته باشیم عدد آن به یک نزدیک و هر چه عملکرد ضعیف‌تری در تشخیص دسته‌ها داشته باشد به عدد صفر نزدیک خواهد بود.**

بررسی معیارهای سنجش دسته‌بندی

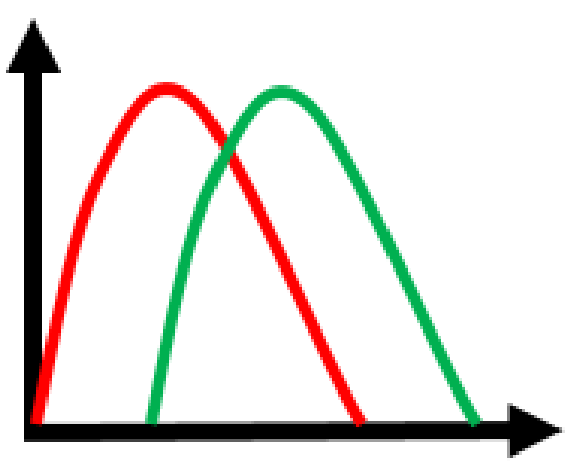


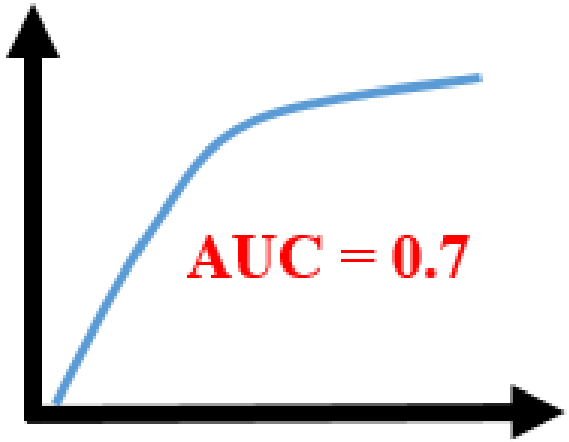


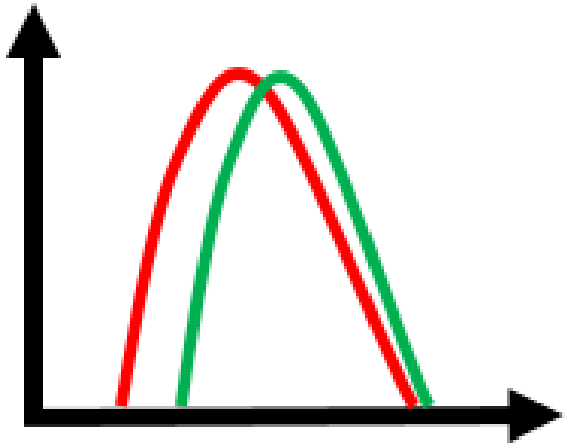


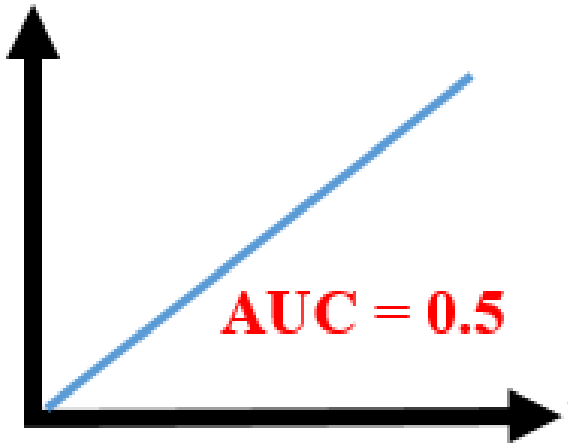


بررسی معیارهای سنجش دسته‌بندی









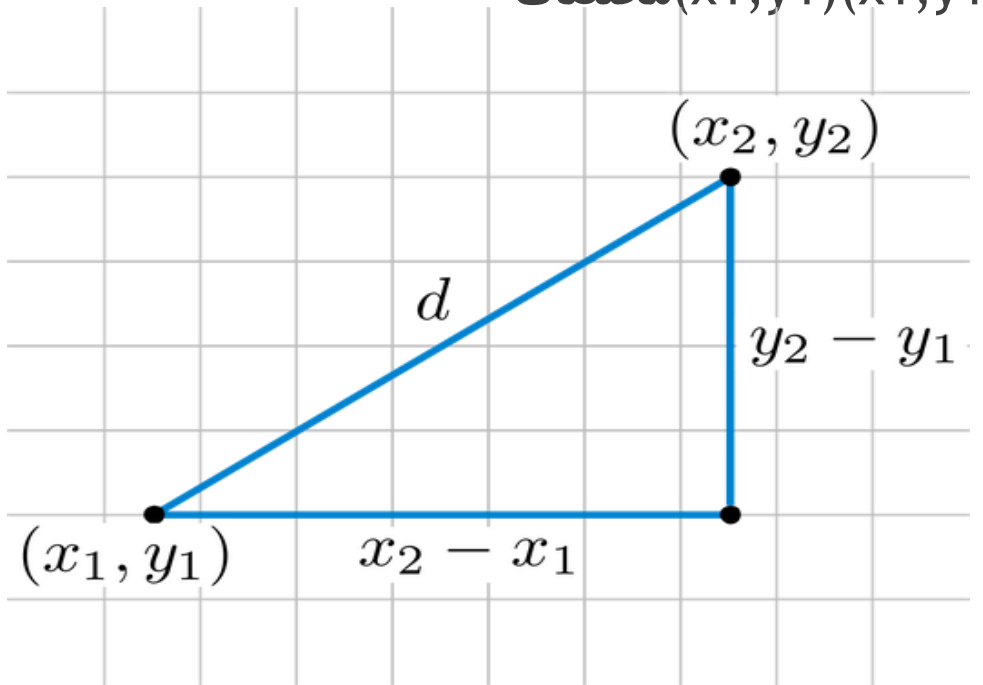
فصل سوم

خوشه بندی

فاصله ها در داده کاوی

تابع فاصله اقلیدسی

- با استفاده از «فاصله اقلیدسی (Euclidean Distance) »کوتاه‌ترین فاصله بین دو نقطه برطبق رابطه فیثاغورث، محاسبه می‌شود. اگر x و y دو نقطه با p مولفه باشند، فاصله اقلیدسی بین این دو به صورت زیر قابل محاسبه است:
- Deuc=(pΣi=1(xi−yi)2)12Deuc=(Σi=1p(xi−yi)2)12
- در تصویر ۱، فاصله اقلیدسی بین دو نقطه با نشان داده شده است (x2,y2)(x2,y2) با (x1,y1)(x1,y1) مختصات



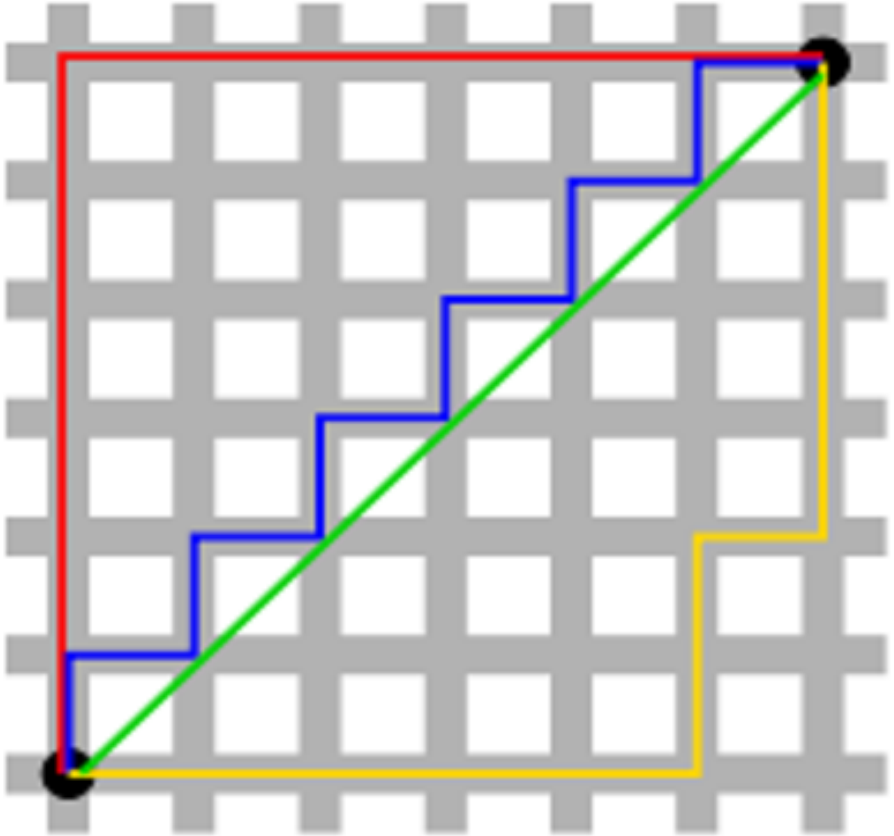
فاصله ها در داده کاوی

تابع فاصله منهتن

- اگر به جای مربع فاصله بین مولفه‌ها، از قدر مطلق فاصله بین مولفه‌های نقاط استفاده شود، تابع فاصله را «منهتن (Manhatan)» می‌نامند. این نام به علت تقاطع منظم خیابان‌ها در محله منهتن نیویورک انتخاب شده است. البته این فاصله گاهی به نام «فاصله تاکسی (Taxicab)» یا «بلوک شهری (City Block)» نیز نامیده می‌شود.
- اگر x و y دو نقطه با p مولفه (p بعدی) باشند، شیوه محاسبه فاصله منهتن به صورت زیر خواهد بود:
- $Dman=p\Sigma i=1|xi-yi|Dman=\Sigma i=1p|xi-yi|$
- همانطور که در تصویر ۳ دیده می‌شود، اگر محور افقی و عمودی به طول‌های برابر و مساوی با ۱ تقسیم شده باشند، بین دو نقطه سیاه رنگ، مسیرهای مختلفی با فاصله منهتن یکسان وجود دارد. خط‌های قرمز و آبی و زرد، مسیرهایی با فاصله منهتن ۱۲ بین این دو نقطه هستند، در حالیکه خط سبز نشان دهنده فاصله اقلیدسی است که با طول برابر با $6\sqrt{262}$ کوتاه‌ترین مسیر را (در صورت وجود) نشان می‌دهد.

فاصله ها در داده کاوی

- براساس نقاط مربوط به مثال ۱، فاصله منهتن بین دو نقطه A و B به صورت زیر محاسبه می‌شود:
- Dman(A,B)=(|2−4|+|4−5|)=2+1=3Dman(A,B)=(|2−4|+|4−5|)=2+1=3
- همانطور که دیده می‌شود، فاصله منهتن برای این دو نقطه بزرگتر از فاصله اقلیدسی است. این واقعیت در تصویر ۳ نیز براساس مقایسه خط سبز و آبی قابل بررسی است. از آنجایی که خط سبز، نشانگر قطر هر مربع با اضلاع خاکستری است، بنا به نامساوی مثلثی از مجموع دو ضلع دیگر کوچکتر هستند.

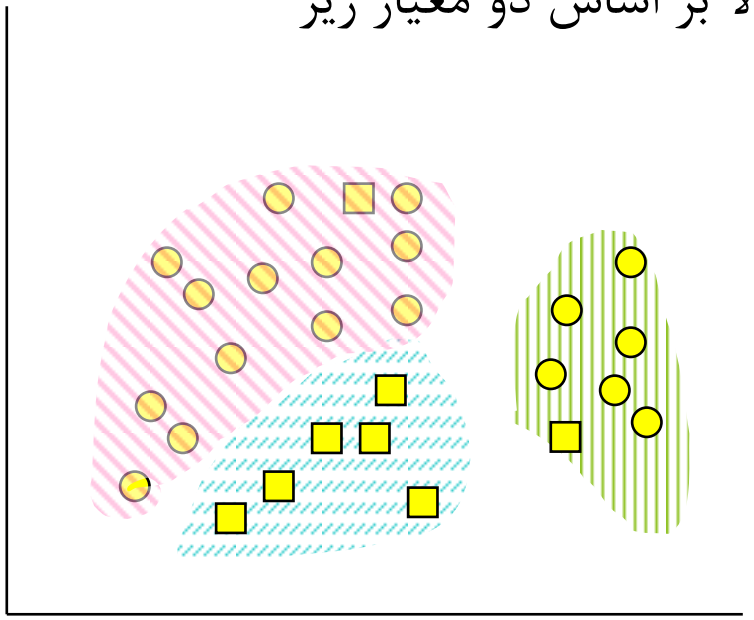


خوشه‌بندی(Clustering)

خوشه‌بندی، گروه بندی نمونه‌های مشابه با هم در یک حجم داده‌ها می‌باشد. به گونه‌ای که نمونه‌های داخل هر گروه با یکدیگر همگن و داده‌های گروه‌های مختلف تا حد ممکن ناهمگن باشند.

یک خوشه‌بندی خوب خوشه‌هایی با کیفیت بالا بر اساس دو معیار زیر تولید می‌کند:

- ✓ شباهت زیاد بین نقاط داخلی هر خوشه
- ✓ شباهت کم بین نقاط خوشه‌های مختلف



از خوشه‌بندی: کاربردهایی

- ✓ **بازاریابی (Marketing)**: دسته‌بندی مشتری‌ها به دسته‌هایی برحسب رفتارها و نیازهای آنها از طریق مجموعه زیادی از ویژگی‌ها و آخرین خریدهای آن‌ها
- ✓ **زیست‌شناسی (Biology)**: دسته‌بندی حیوانات و گیاهان از روی ویژگی‌های آن‌ها
- ✓ **کتابداری** : دسته‌بندی کتاب‌ها
- ✓ **نقشه‌برداری شهری (City-Planning)**: دسته‌بندی خانه‌ها بر اساس نوع و موقعیت جغرافیایی آن‌ها
- ✓ **مطالعات زلزله‌نگاری (Earthquake studies)**: تشخیص مناطق حادثه‌خیز بر اساس مشاهدات قبلی
- ✓ ....

تعداد بهینه خوشه ها

روش های تعیین تعداد بهینه خوشه ها

- **روش های مستقیم یا** Direct Methods **و روش های مبتنی بر آزمون های آماری یا** Statistical Testing Methods
- **روش های مستقیم تعیین تعداد بهینه خوشه ها در الگوریتم های خوشه بندی: این روش ها به دنبال بهینه سازی یک معیار به‌خصوص، مانند مجموع مربعات فواصل درون خوشه ای (**Within-cluster Sum of Square **یا** WSS**) یا سیلوئت میانگین (** Average Silhouette**) هستند. از جمله این متدها می‌توان به متد** elbow **و روش های مبتنی بر معیار** silhouette **اشاره کرد.**
- **روش های مبتنی بر آزمون های آماری در تعیین تعداد بهینه خوشه ها در الگوریتم های خوشه بندی: این متدها به دنبال تطبیق مشاهدات با فرض صفر یک آزمون آماری هستند. از جمله این روش ها می‌توان به** Gap Statistics **اشاره کرد.**

تعداد بهینه خوشه ها

خوشه بندی X-means

- در آمار و داده کاوی، X-means clustering نوعی از خوشه بندی k-means است که خوشه‌ها را براساس عملیات مکرر تقسیم و دسته‌بندی، و نگه داشتن بهترین نتیجه تا زمانی که معیاری مانند معیار اطلاعاتی آکائیکه (AIC) یا معیار اطلاع بیزی–شوارتز (BIC) بدست آید.

تعداد بهینه خوشه ها

- رویکرد معیارهای اطلاعاتی (Information criterion)
- یکی دیگر از مجموعه روش‌ها برای تعیین تعداد خوشه‌ها، اطلاعات معیاری مانند (Akaike information criterion (AIC, اطلاعات معیار Bayesian (BIC) یا انحراف اطلاعات معیار (DIC) – اگر ایجاد تابع برای مدل خوشه بندی ممکن باشد. به عنوان مثال: مدل k–means مدل «تقریباً» یک Gaussian mixture model است و ساخت احتمالی برای مدل مرکب گوسی مقدور است و نتیجه نیز تعیین مقادیر معیار است.[۵]

تعداد بهینه خوشه ها

رویکرد نظری اطلاعات (information-theoretic)

- نظریه نرخ اعوجاج برای انتخاب k به نام روش «پرش» نامگذاری شده‌است که تعداد خوشه‌ها را برای دستیابی به حداکثر راندمان و همزمان به حداقل رساندن خطا توسط استانداردهای اطلاعات نظریتعیین می‌کند.[۶] استراتژی الگوریتم این است که اعوجاج منحنی تولید می‌کند برای داده‌های ورودی در حال اجرا توسط استاندارد الگوریتم خوشه بندی مانند k-means برای تمام مقادیر k بین ۱ و n و اعوجاج را (در زیر توضیح داده شده) و در نتیجه خوشه بندی مشخص می‌کند. اعوجاج منحنی معیاری منفی برای تعیین ابعاد داده‌است. روش «پرش» مقادیری را برای انتخاب درست k *خروجی می‌دهد*. بزرگترین پرش به عنوان بهترین انتخاب است.

تعداد بهینه خوشه ها

روش نیمرخ (silhouette)

- متوسط نیمرخ (silhouette) از اطلاعات معیار مفید دیگری برای ارزیابی طبیعی تعداد خوشه هاست. نیمرخ (silhouette) یک نمونه داده میزان نزدیک شدن داده‌ها درون خوشه و میزاَن آزادی و فاصله داده‌ها از خوشه همسایه است. یعنی خوشه ای که میانگین فاصله از مرجع "datum" پایین‌ترین است.[۷] یک نیمرخ نزدیک به ۱ به معنی است که datum در خوشه ای مناسب است در حالی که یک نیمرخ نزدیک به –۱ به این معنی است که datum در خوشه اشتباه قرار گرفته‌است. تکنیک‌های بهینه‌سازی مانند الگوریتم ژنتیک در تعیین تعداد خوشه‌ها مفیدند که بزرگترین نیمرخ را بدست می‌دهند.

تعداد بهینه خوشه ها

اعتبارسنجی متقاطع Cross-validation

- همچنین می‌توان از فرایند «اعتبارسنجی متقاطع» cross-validation به تجزیه و تحلیل تعداد خوشه پرداخت. در این فرایند، داده‌ها به v قطعه تقسیم می‌شوند. هر یک از قطعات به نوبه خود به عنوان یک مجموعه تست هستند. مدل خوشه بندی محاسبه شده در $v - 1$ مجموعه دیگر و مقدار تابع هدف (برای مثال مجموع مربعات فاصله به centroids برای k–means) برای مجموعه تست محاسبه می‌شوند. این v مقادیر محاسبه و برای هر تعداد خوشه جایگزینی محاسبه و میانگین گرفته می‌شود. تعداد خوشه انتخاب شده به طوری که افزایش در تعداد خوشه منجر به کوچک سازی در تابع هدف شود.

تعداد بهینه خوشه ها

تجزیه و تحلیل هسته ماتریس

- هسته ماتریس، مجاورت اطلاعات ورودی را تعریف می‌کند. برای مثال در تابع پایه شعاعی گوسی، نقطه ورودی‌ها را در فضایی با ابعادی بالاتر به نام «فضای ویژگی‌ها» تعیین می‌کند. اعتقاد بر این است که داده در فضای ویژگی‌ها به خطوطی قابل تفکیک تبدیل می‌شود. از این رو الگوریتم‌های خطی را می‌توان بر روی داده‌ها با موفقیت بالاتری بکار گرفت.
- هسته ماتریس می‌تواند در جهت پیدا کردن تعداد بهینه خوشه‌ها تحلیل و پردازش شود.[۱۴] این روش با تجزیه مقدار ویژه (eigenvalue) هسته ماتریس کار می‌کند. سپس آن مقادیر ویژه و بردارهای ویژه (eigenvectors) برای به دست آوردن میزان فشردگی در توزیع ورودی‌ها تحلیل می‌شود. در نهایت یک طرح کشیده خواهد شد که در آن آرنج این طرح نشان دهنده تعداد بهینه خوشه‌ها در مجموعه داده‌است. بر خلاف روش‌های قبلی این روش نیازی به انجام هر خوشه یک–پیشینی (سابقه داده) ندارد. این روش به‌طور مستقیم تعداد خوشه‌ها را از داده‌ها می‌یابد.

تعداد بهینه خوشه ها

متد elbow

- روش elbow، مجموع فواصل درون خوشه ای داده ها را به عنوان تابعی از تعداد خوشه ها در نظر می‌گیرد. به این ترتیب تعداد خوشه ها به نحوی انتخاب می‌شوند که افزودن یک خوشه دیگر، بهبودی در حداقل‌سازی WSS ایجاد نکند.
- روش آرنج درصد واریانس را به عنوان تابعی از تعداد خوشه‌ها توضیح می‌دهد: یکی باید به عنوان تعداد خوشه‌ها انتخاب شود به طوری که با اضافه کردن خوشه ای دیگر مدل‌سازی داده بهتری بدست نیاید. دقیق تر، اگر یک ترسیم (plot) درصد واریانس را تشریح کند طوریکه مخالف تعداد خوشه‌ها باشد اولین خوشه‌ها اطلاعات زیادی (توضیح بسیاری از واریانس) را اضافه می‌کنند، اما در بعضی نقطه‌ها حاشیه سود کاهش خواهد یافت و یک زاویه در نمودار به وجود می‌آورد. تعداد خوشه‌ها در این نقطه انتخاب شده‌اند یعنی همان «معیار آرنج». این «آرنج» نمی‌تواند همیشه به روشنی مشخص شود.[۱] درصد از واریانس نسبت واریانس بین—گروهی به کل واریانس را توضیح داده، همچنین به عنوان یک آزمون F شناخته شده‌است.

تعداد بهینه خوشه ها طبق الگوریتم زیر به‌دست می‌آید:

- ۱– اجرای الگوریتم خوشه بندی مانند k-means برای مقادیر متفاوت k (به‌طور مثال با در نظر گرفتن مقدار k در بازه ۱ تا ۱۰)
  ۲– محاسبه مقدار WSS برای هر مقدار k

تعداد بهینه خوشه ها

متد silhouette میانگین

- این معیار مشخص می‌کند که پراکندگی داده ها در خوشه ها به چه صورت است. هر چه مقدار سیلوئت بالاتر باشد، کیفیت خوشه بندی نیز بالاتر است.
- در متد سیلوئت میانگین ، الگوریتم خوشه بندی به ازای مقادیر مختلف k اجرا شده و به‌ازای هر اجرا، معیار سیلوئت برای هر یک از اعضای خوشه ها محاسبه می‌شود. سپس از سیلوئت های به‌دست آمده معدل گرفته می‌شود. مقدار بهینه k مقداری است که به ازای آن، سیلوئت میانگین ماکزیمم شود.
الگوریتم این متد مشابه روش elbow بوده و می‌توان آن را به صورت زیر نوشت:
- ۱– اجرای الگوریتم خوشه بندی مانند k-means برای مقادیر متفاوت k (به طور مثال با در نظر گرفتن مقدار k در بازه ۱ تا ۱۰)
۲– محاسبه مقدار سیلوئت هر یک از مشاهدات برای هر مقدار k و محاسبه میانگین آن‌ها
۳– رسم مقدار میانگین سیلوئت بر حسب مقادیر مختلف k
۴– نقطه ماکزیمم نمودار رسم شده، تعداد بهینه خوشه ها را نشان می‌دهد.

تعداد بهینه خوشه ها

متد Gap Statistics

- این روش می‌تواند بر انواع مختلف الگوریتم های خوشه بندی اعمال شود. متد Gap Statistics به‌ازای هر یک از مقادیر در نظر گرفته شده برای k، مجموع تفاضلات درون خوشه ای داده ها را با مقادیر مورد انتظار آن‌ها (توزیع داده ها با فرض درست بودن فرض صفر) مقایسه می‌کند. مقدار بهینه خوشه ها، مقداری است که آماره gap را ماکزیمم کند. این به آن معناست که ساختار خوشه بندی از توزیع یکنواخت تصادفی داده ها دور است.
- الگوریتم این روش را می‌توان به فرم زیر نوشت:
- ۱– خوشه بندی داده ها با در نظر گرفتن مقدار k در بازه ۱ تا k_{max} . محاسبه مقدار تفاضلات درون خوشه ای برای هر یک از اجراها و قرار دادن آن در متغیر W_k
  ۲– تولید تعداد B مجموعه داده از مجموعه داده های اصلی، به نحوی که دارای توزیع یکنواخت تصادفی باشند. خوشه بندی هر یک از این B مجموعه داده به‌ازای مقادیر مختلف k (در بازه ۱ تا k_{max}). محاسبه مقدار تفاضلات درون خوشه ای برای هر یک از اجراها و قرار دادن آن در متغیر W_{kb}

$$Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}^{*}) - log(W_k)$$

تعداد بهینه خوشه ها

۳– محاسبه آماره gap به صورت تفاضل مقادیر مشاهده شده W_k از مقادیر مورد انتظار آن‌ها تحت فرض صفر W_{kb} و نیز محاسبه انحراف معیار آماره به‌دست آمده:

- **۴– انتخاب تعداد بهینه خوشه ها به صورت کم‌ترین مقدار k به‌طوری‌که** $Gap(k) \geq Gap(k+1) - s_{k+1}$
- **ذکر این نکته ضروری است که مقدار B = 500 نتایج دقیقی ارائه می‌دهد، به‌طوریکه نمودار آماره gap با اجراهای مجدد الگوریتم بدون تغییر باقی می‌ماند.**

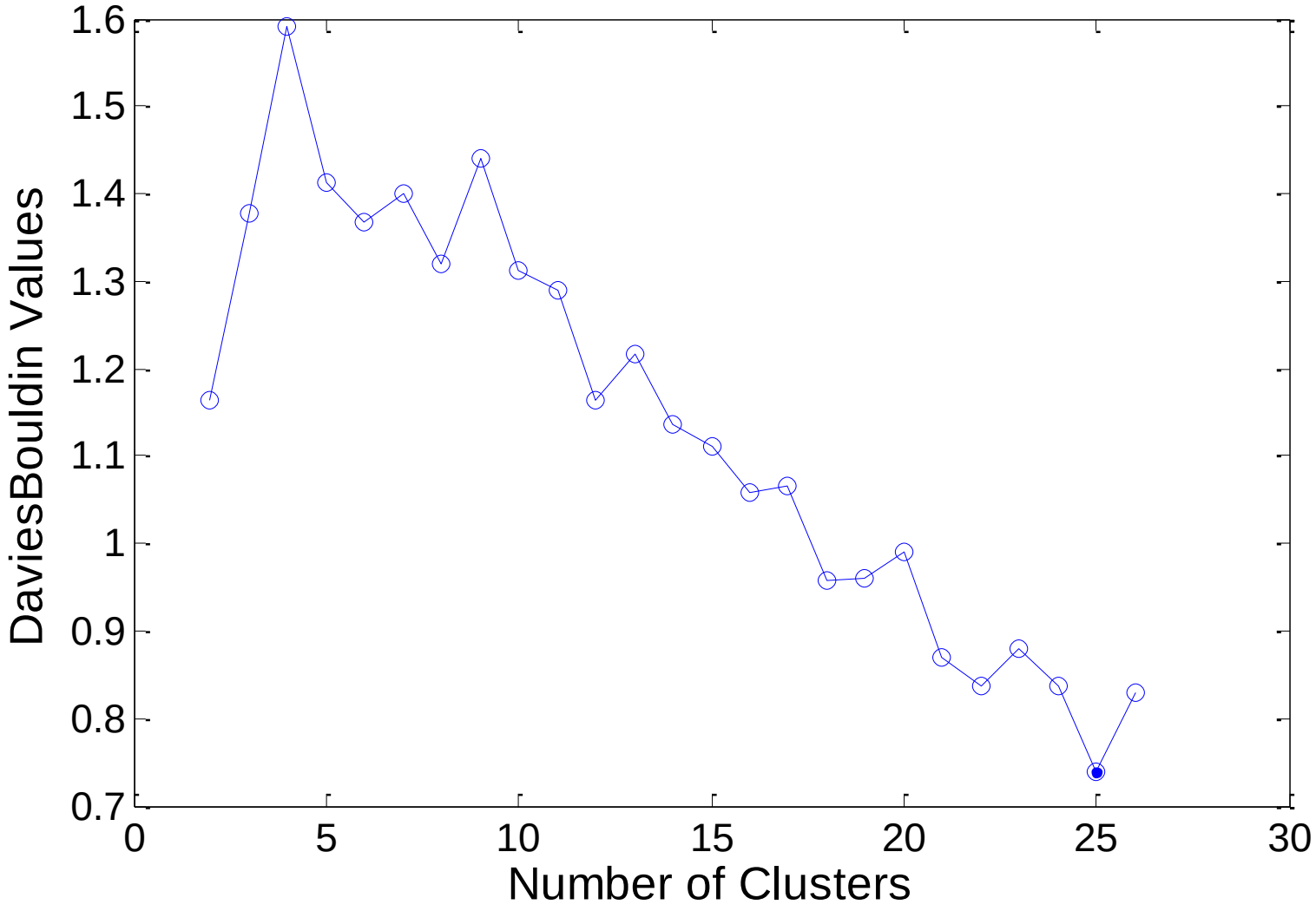
$$Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}^{*}) - log(W_k)$$

شاخص دیویس– بولدین

$$DB = \frac{1}{N_c}\sum_{J=1}^{N_C} \max_{l \neq j} \frac{S(Q_j)+S(Q_l)}{d(Q_j,Q_l)}$$

- برای تعیین تعداد بهینه خوشه ها
- فاصله درون خوشه $S(Q_K)$
- خوشه-های مختلف $Q_i, i = 1, 2, , ..., N_c$
- فاصله میان خوشه-های $d(Q_j, Q_l)$

شاخص دیویس– بولدین



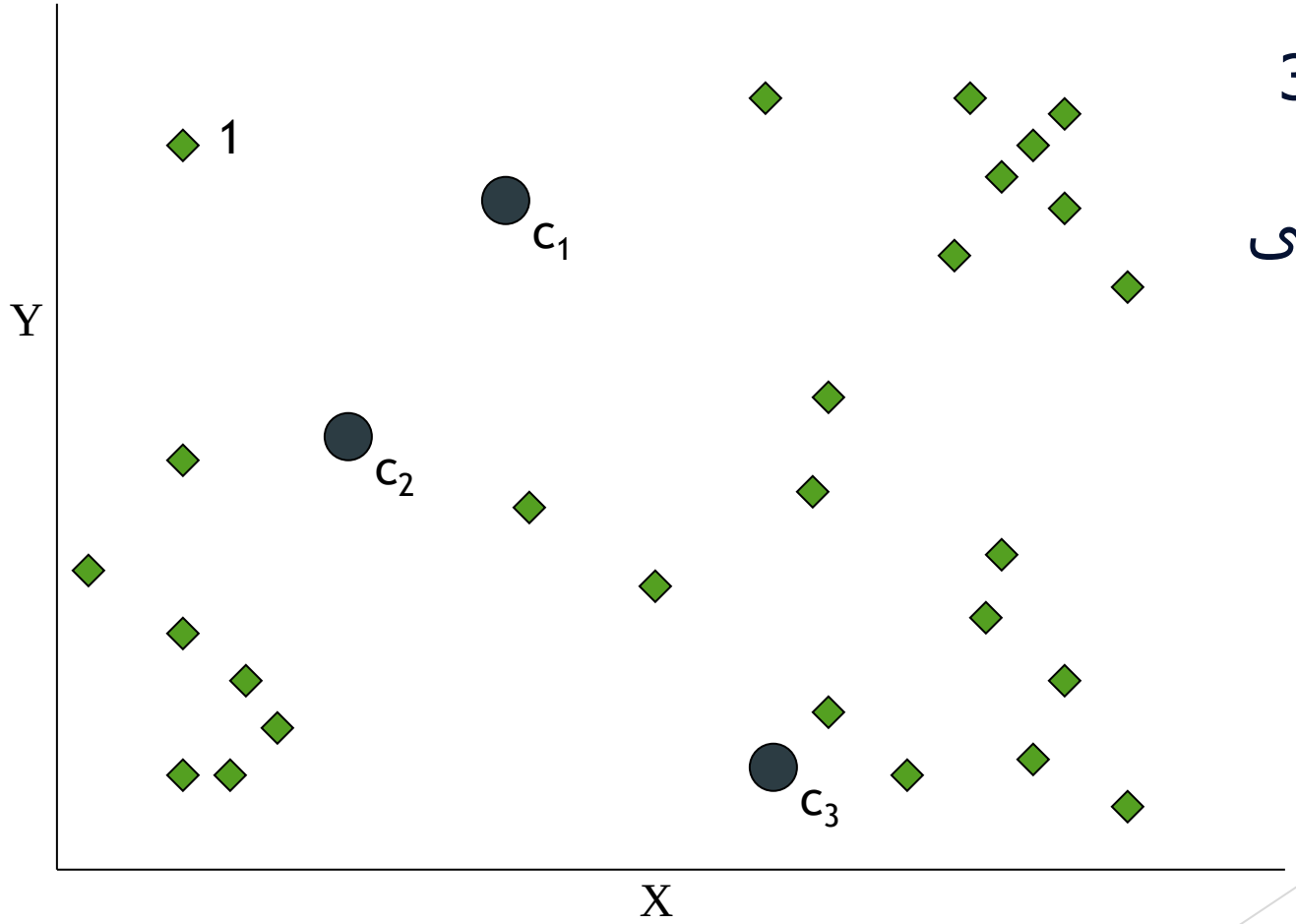
الگوریتم خوشه‌بندی K-Means:

الگوریتم خوشه‌بندی K-Means یکی از ساده-ترین و مشهورترین الگوریتم‌های یادگیری بدون نظارت است. در K-Means عملاً مجموعه داده-ها به تعداد خوشه-های از پیش تعیین شده تقسیم می‌شوند.

الگوریتم K-Means به شرح زیر می-باشد:

- ✓ انتخاب چند نقطه به عنوان مراکز خوشه ها(بصورت تصادفی یا روش‌های دیگر)
- ✓ تعیین فواصل بقیه داده-ها با مرکز خوشه-ها
- ✓ قرارگیری داده‌هایی که به مرکز هر خوشه نزدیکترند در آن خوشه
- ✓ محاسبه میانگین هر خوشه به عنوان مرکز جدید خوشه
- ✓ تکرار مرحله دوم تا چهارم جهت رسیدن به نتیجه مطلوب

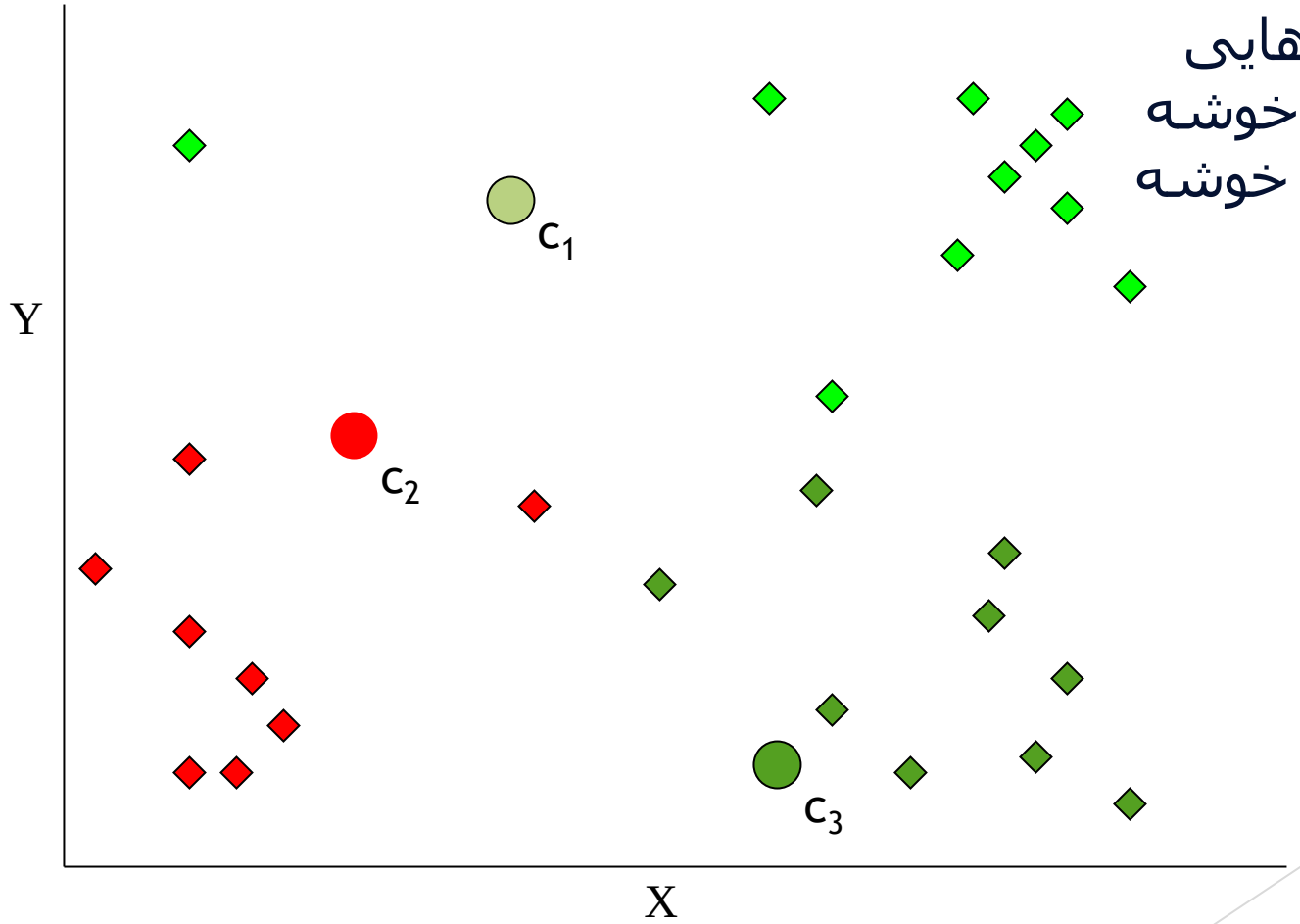
مثال k-means :



انتخاب اولیه 3 مرکز خوشه بصورت تصادفی

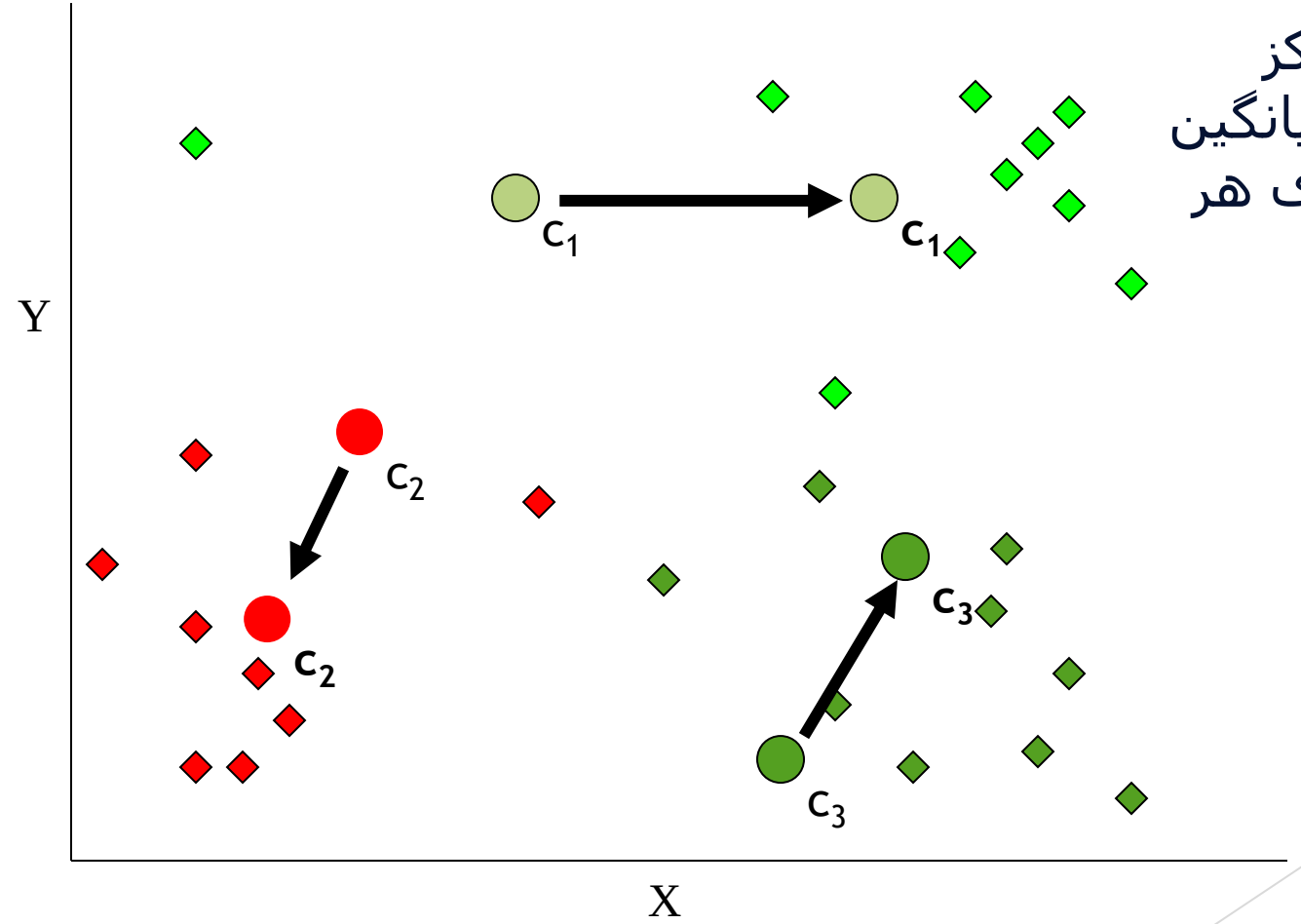
مثال k-means :

قرار گیری داده‌هایی
که به مرکز هر خوشه
نزدیکترند در آن خوشه

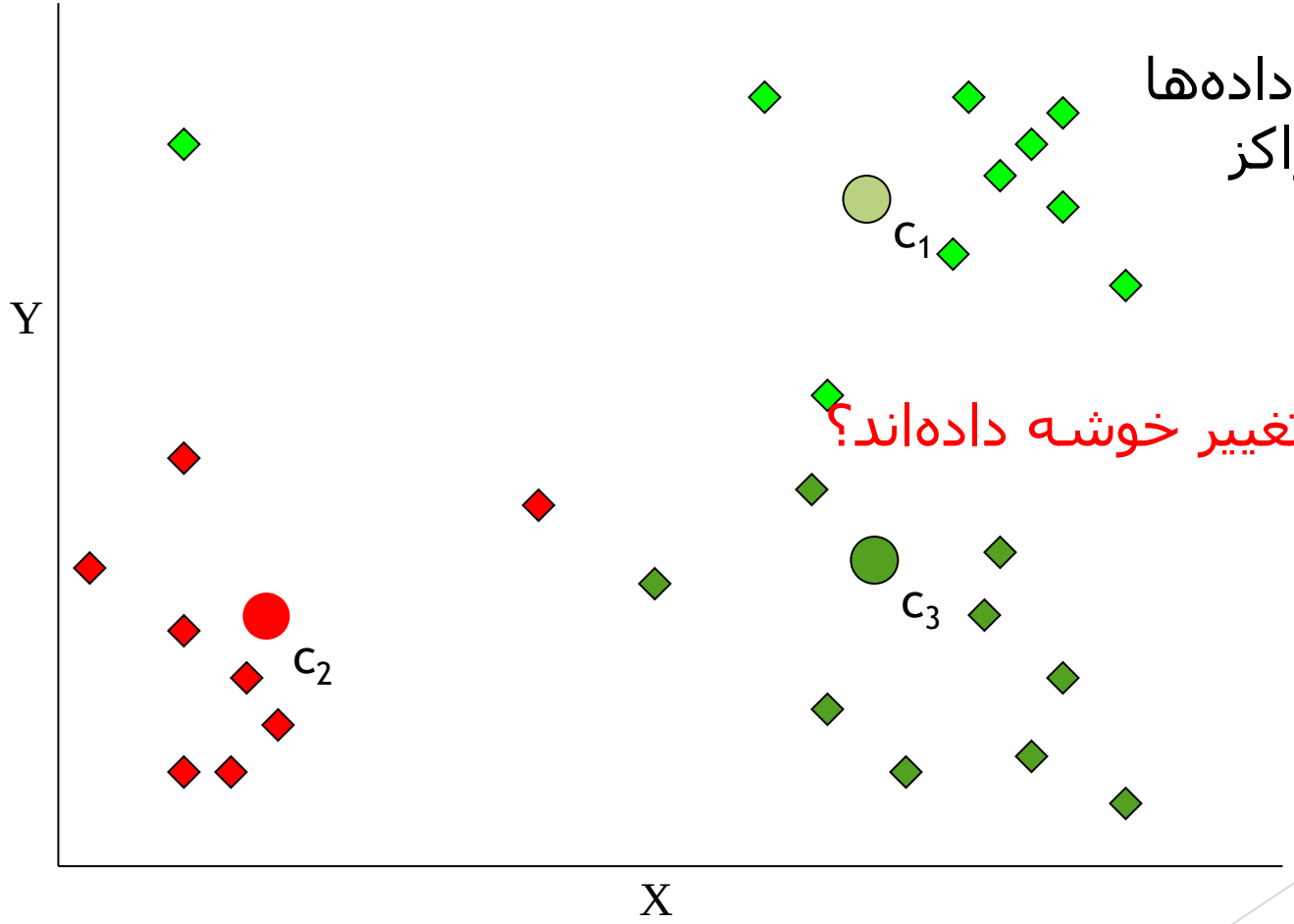


مثال k-means :

تغییر مکان مراکز خوشه‌ها به میانگین مکانی داده‌های هر خوشه



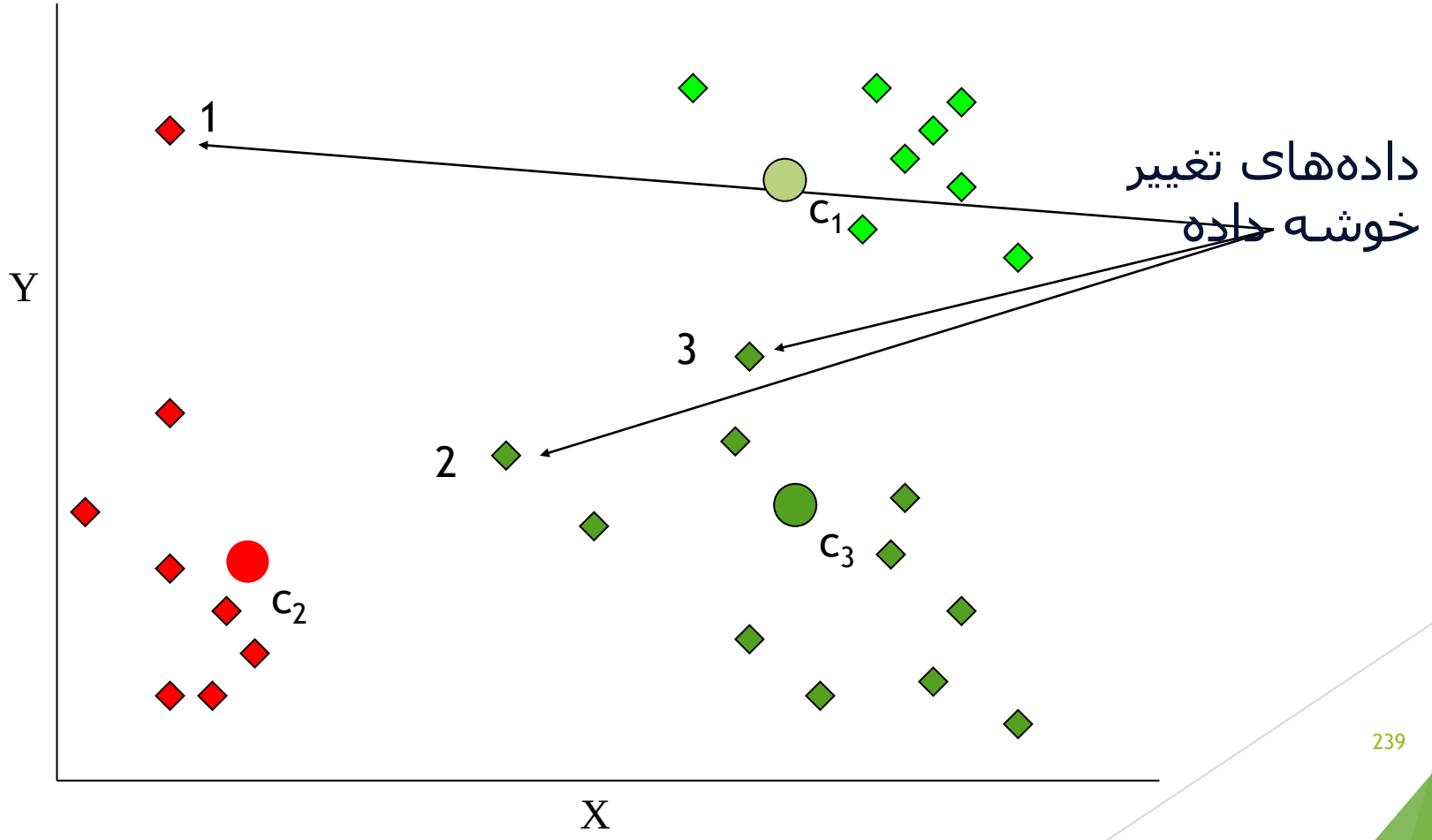
مثال k-means :



تخصیص دوباره داده‌ها
به نزدیکترین مراکز
خوشه جدید

چه داده‌هایی تغییر خوشه داده‌اند؟

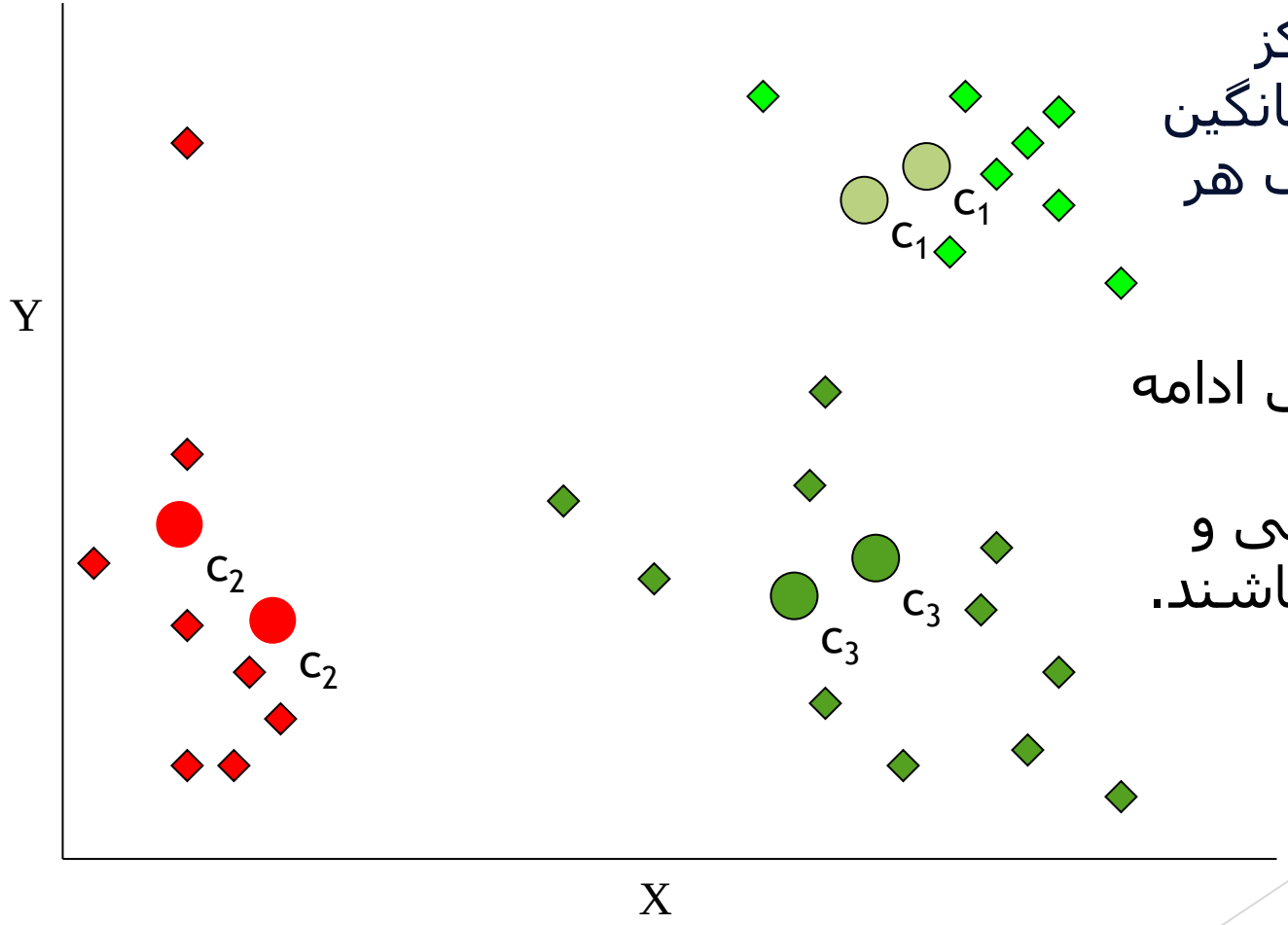
مثال k-means :



مثال k-means :

تغییر مکان مراکز خوشه‌ها به میانگین مکانی داده‌های هر خوشه

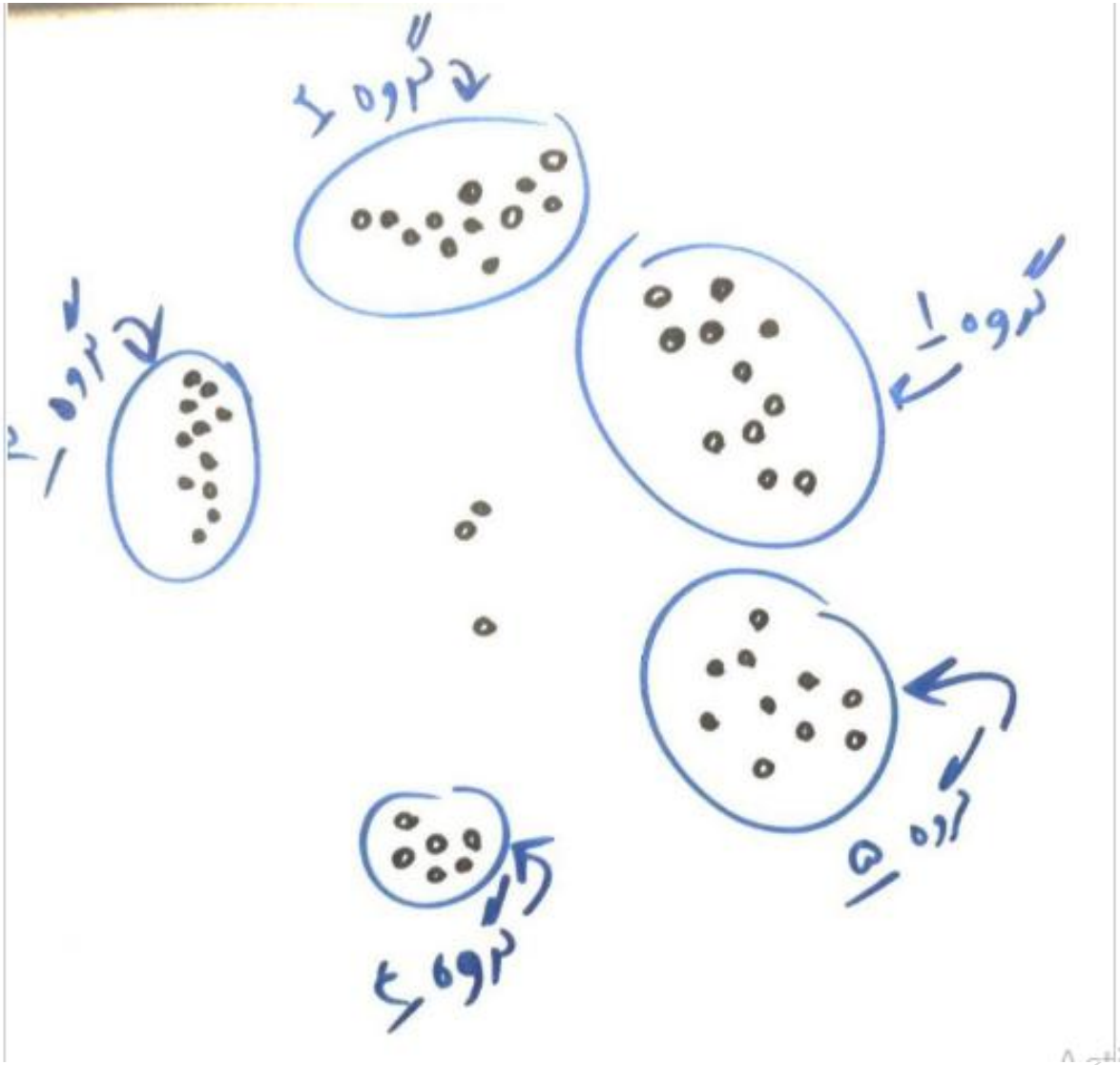
این کار تا زمانی ادامه پیدا می‌کند که خوشه‌های قبلی و بعدی یکسان باشند.



خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

- بعد از الگوریتم خوشه‌بندی KMeans، الگوریتمِ DBSCAN را بتوان معروف‌ترین الگوریتمْ در حوزه‌ی خوشه‌بندیِ داده‌ها دانست.
- یکی از تفاوت‌های اصلی این الگوریتم با KMeansاین است که الگوریتم DBSCANنیاز به تعیین تعدادِ خوشه توسط کاربر ندارد و خودِ الگوریتم می‌تواند خوشه‌ها را مبتنی بر غلظتِ آن‌ها شناسایی کند.

خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

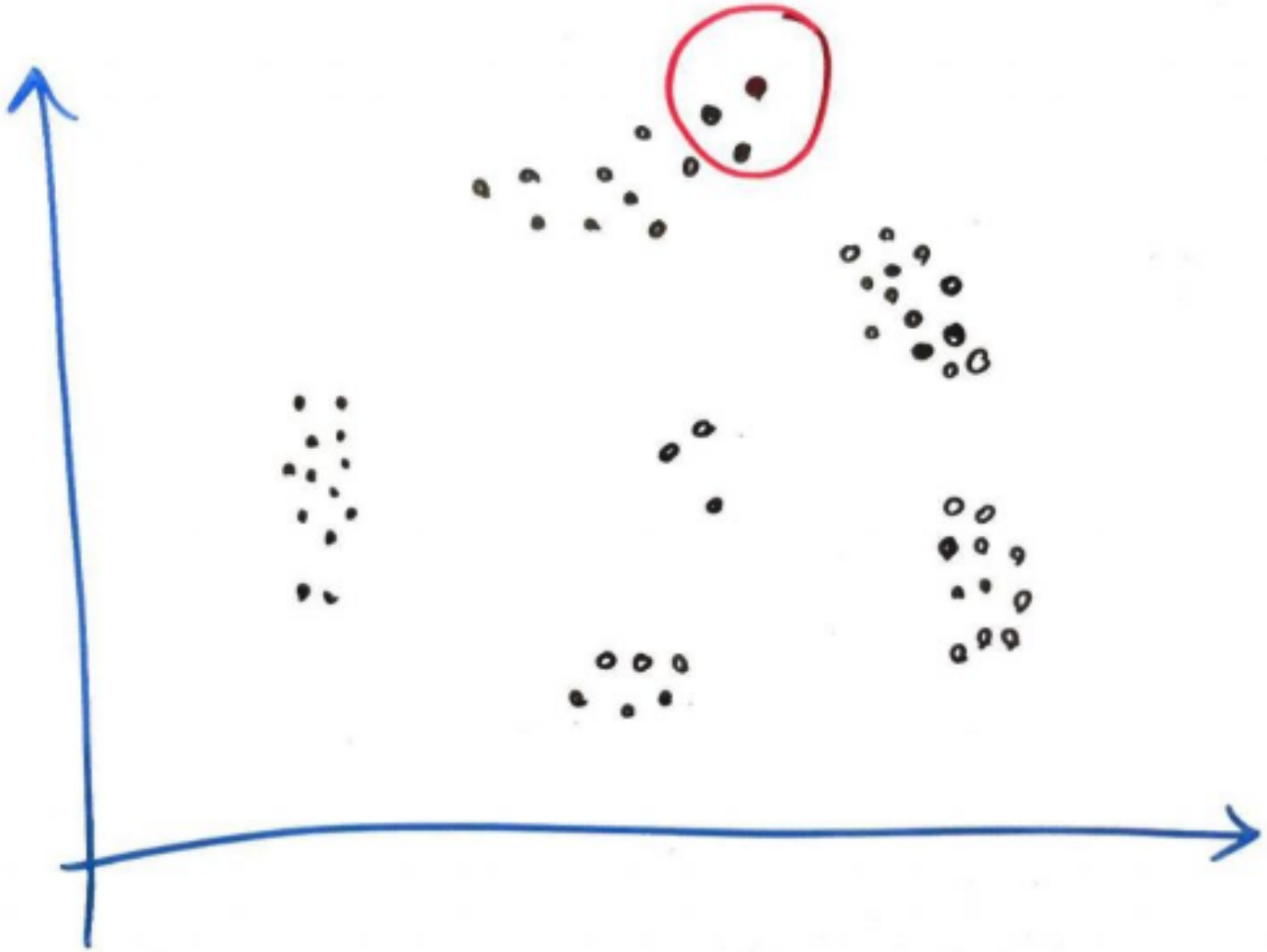


خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

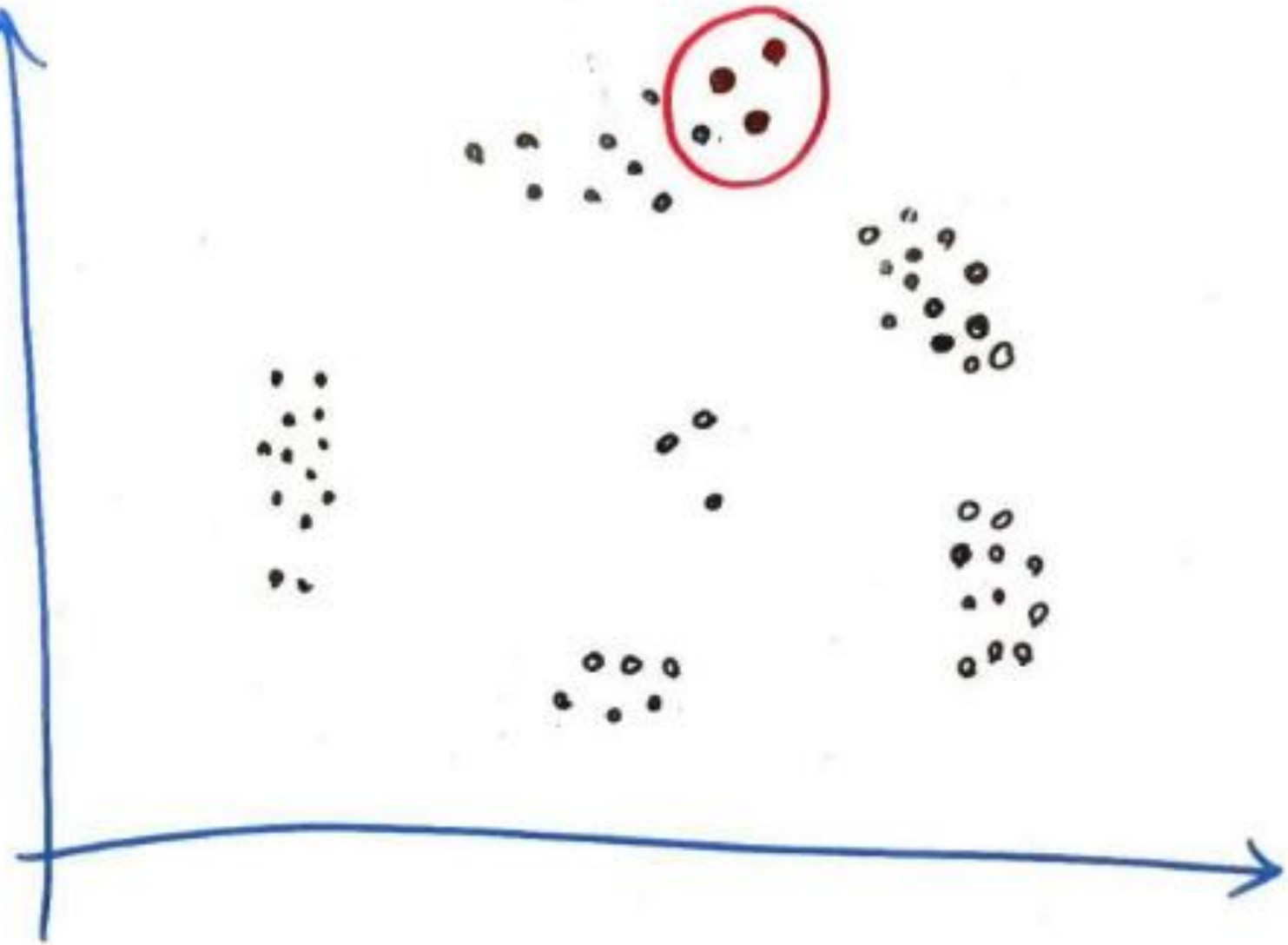
- در این روش با استفاده از تراکم و غلظتی که دیده می شود، نمونه ها خوشه‌بندی می شوند.
- در الگوریتمِ DBSCANدو پارامتر وجود دارد:
- ۱– یکی از آن‌ها شعاع است که به آن Epsilon نیز می‌گویند
- ۲– دومی حداقل نقاط موجود در یک خوشه است که به آن MinPointsمی‌گویند.
- این الگوریتم ابتدا یک نمونه را انتخاب می‌کند و با توجه به شعاعِ Epsilon به دنبال همسابه برای این نقطه در فضا می‌گردد.
- اگر الگوریتم در آن شعاع مشخص Epsilon حداقل توانست به تعدادِ MinPoint نقطه پیدا کند، آن‌گاه همه‌ی آن نقطه‌ها با هم به یک خوشه تعلق می‌گیرند.
- الگوریتمْ سپس به دنبال یکی از نقطه‌های همجوار نقطه فعلی می‌رود تا دوباره با شعاعِ Epsilon در آن نقطه به دنبالِ نقاط همسایه دیگر بگردد
- اگر تعداد نقاطِ همسایه‌ی جدید بازهم پیدا شوند، این الگوریتم دوباره همه آن نقاط جدید را با نقاط قبلی به یک خوشه متعلق می‌کند و اگر نقطه‌ی جدیدی در همسایگی پیدا نکرد این خوشه تمام شده است و برای پیدا کردنِ خوشه‌های دیگر در نقاط دیگر، به صورت تصادفی یک نقطه دیگر را انتخاب کرده و شروع به یافتنِ همسایه و تشکیلِ خوشه‌ی جدید برای آن نقطه می‌کند. این کار آنقدر ادامه پیدا می کند تا تمامی نقاط بررسی شوند.

خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

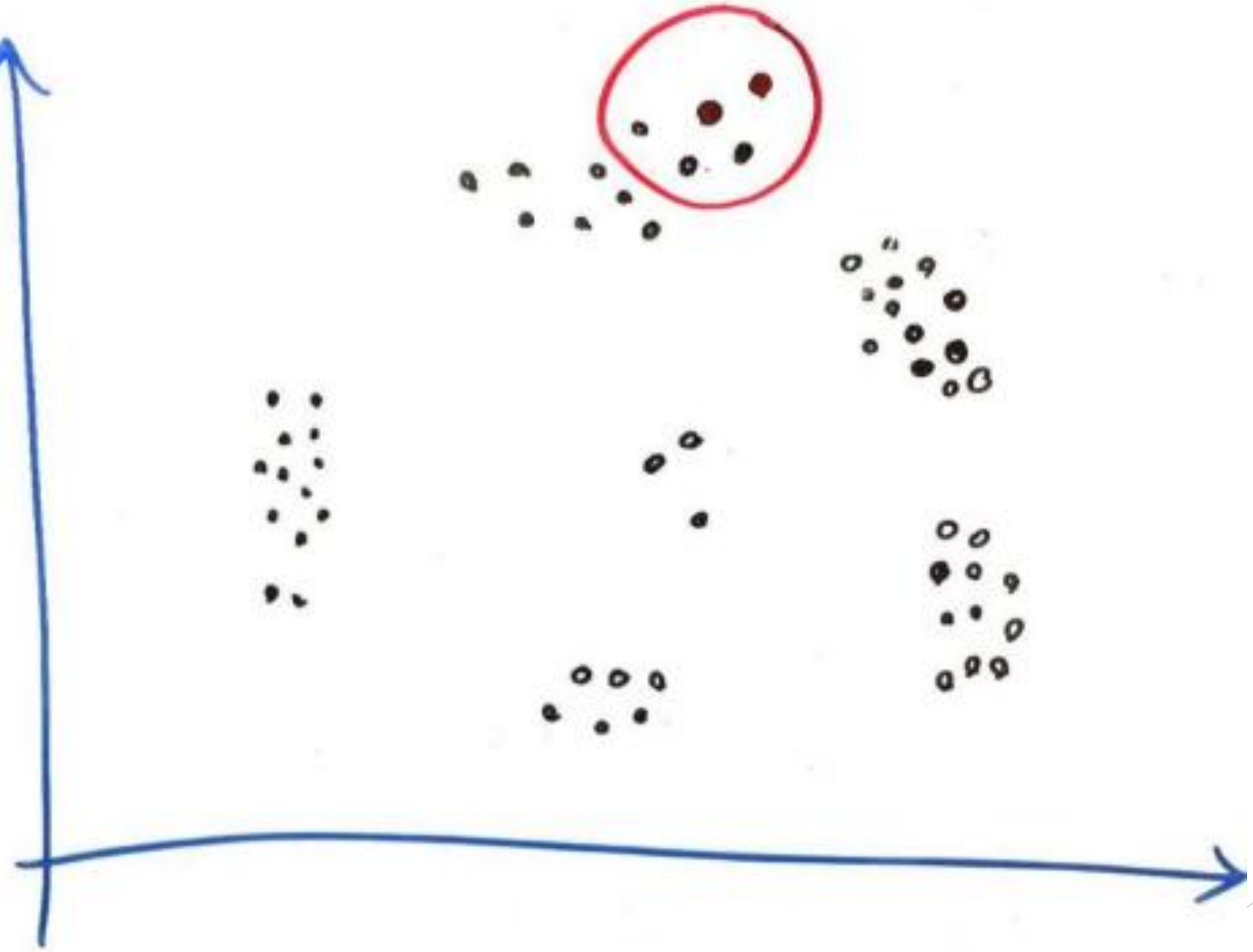
- در مثال زیر با فرض مقدار شعاع آستانه برابر با ۱ و تعداد حداقل اعضاء خوشه برابر با ۳ داریم:



خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

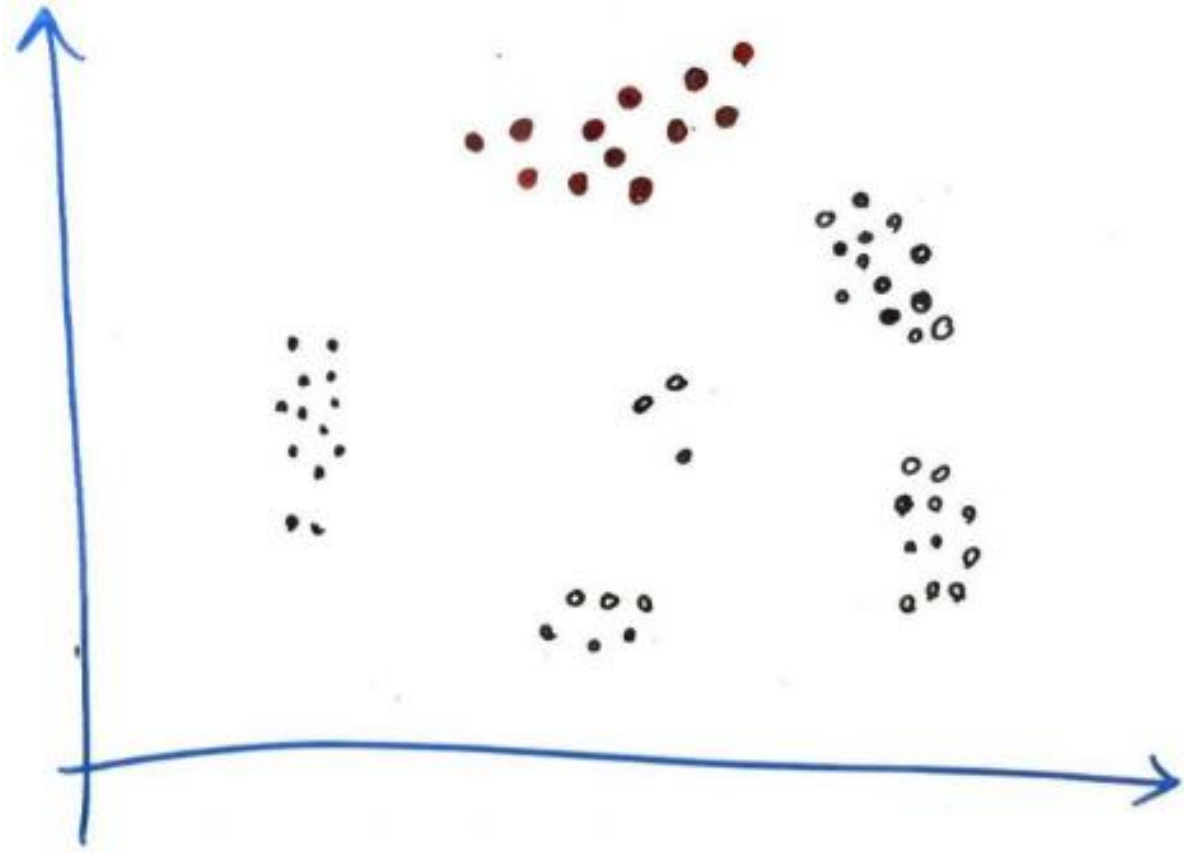


خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت



خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

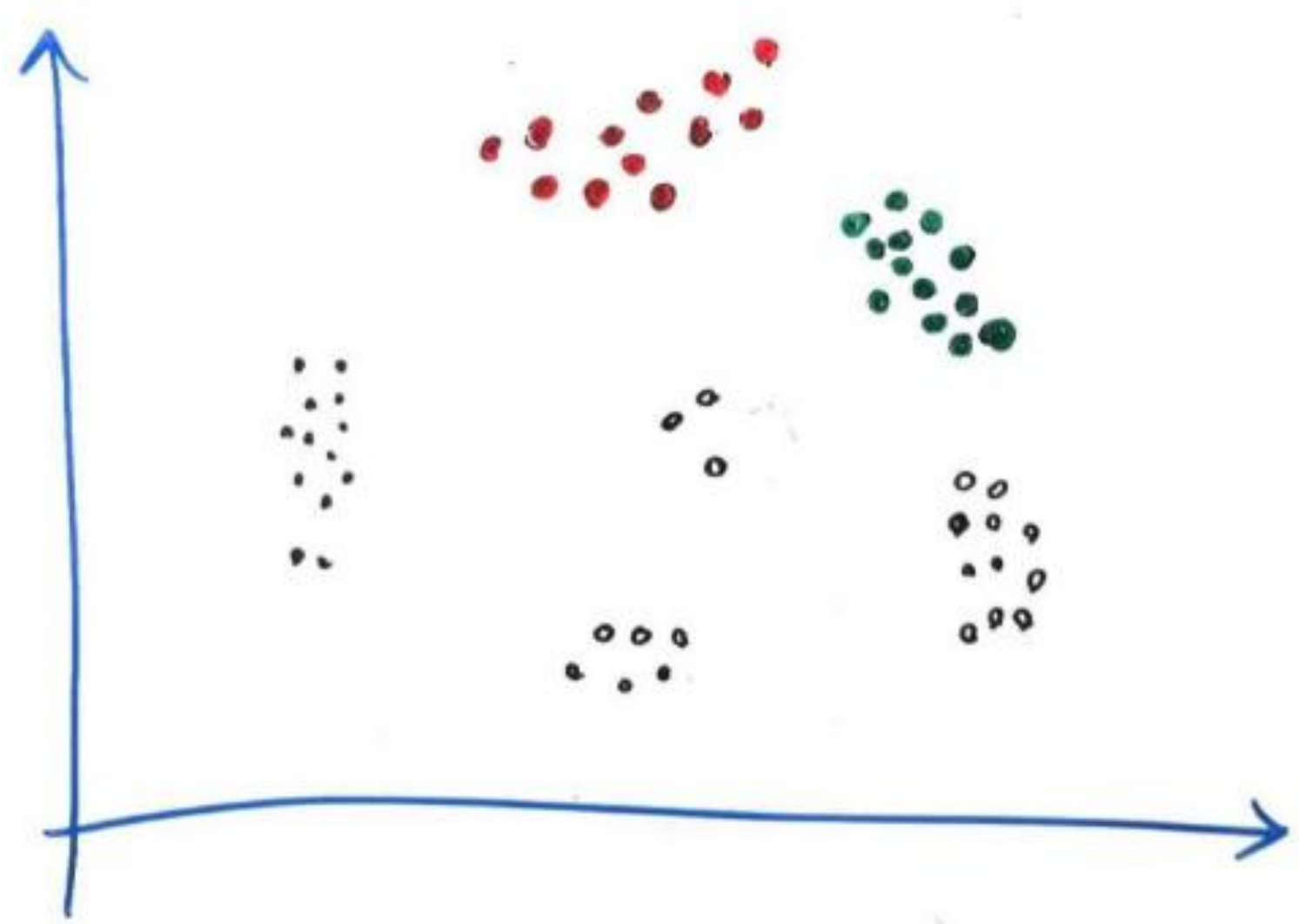
- این کار ادامه پیدا می‌کند تا زمانی که دیگر نقطه‌ای در میان خوشه‌های نزدیکِ همسایه‌ها نباشد.
- در شکل زیر، که تمامیِ نمونه‌های بالای شکل به خوشه‌های قرمز نسبت داده شده است:



خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

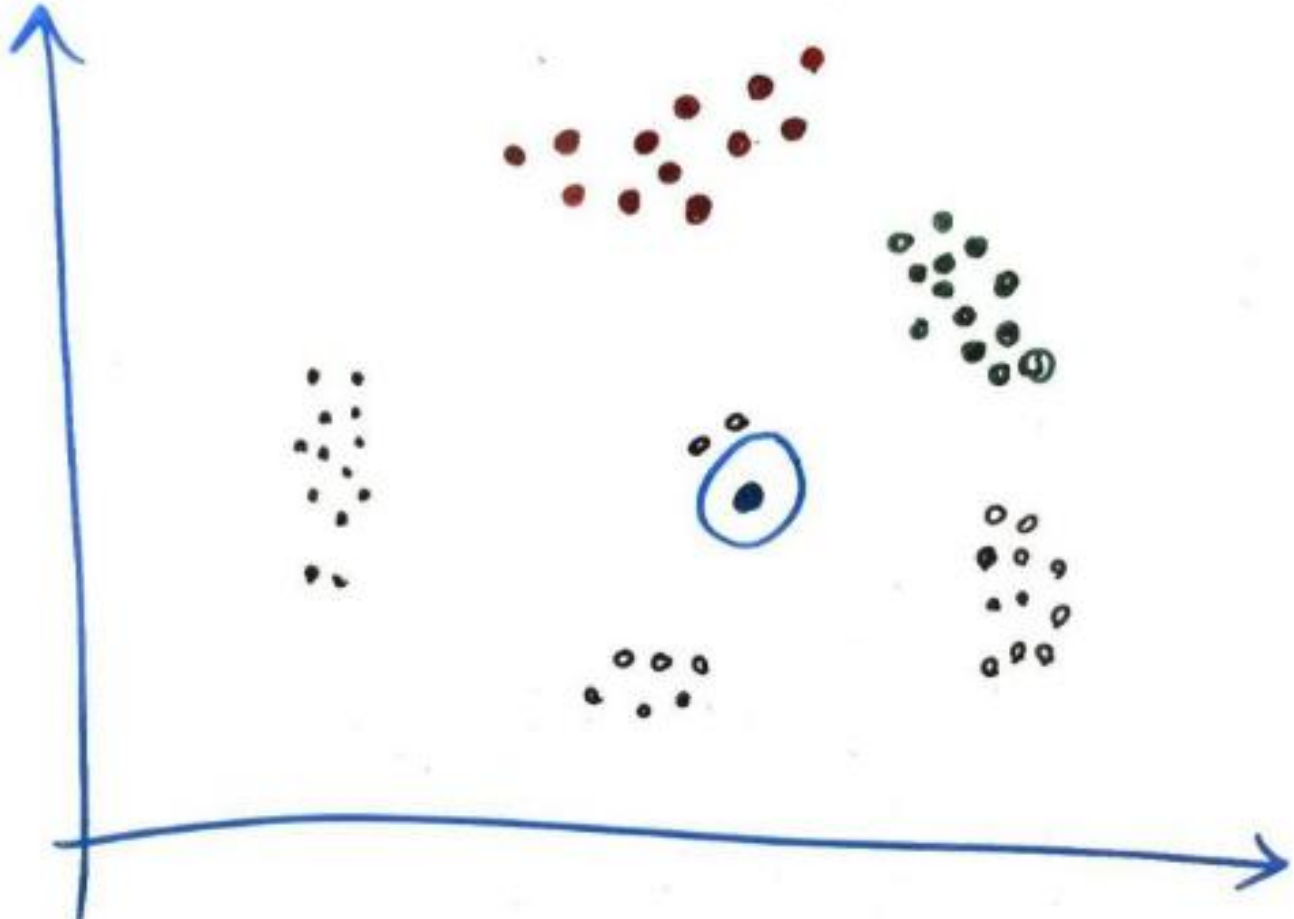
- ممکن است تمامیِ نمونه‌ها در همسایگی، دارای حداقل ۳همسایه نباشند. به این نمونه‌ها،نقطه ِهای مرزی یا حاشیه(borderگفته می‌شوند.
- مثلا در همان شکل بالا، در میان خوشه‌ی قرمز رنگ، ممکن است نمونه‌ای که انتهای سمت چپ قرار دارد، یک نقطه مرزی باشد.
- الگوریتم به همین ترتیب ادامه پیدا میکند. هنگامی که یک خوشه(مانند خوشه‌ی قرمز بالا) تشکیل شد، یک نقطه‌ی تصادفی دیگر را انتخاب می‌کنیم.

خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت



خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

- فرض کنید یک نمونه مانند شکل زیر پیدا شد که در شعاعِ مورد نظر الگوریتم، به تعداد کافی نمونه پیدا نکرد:

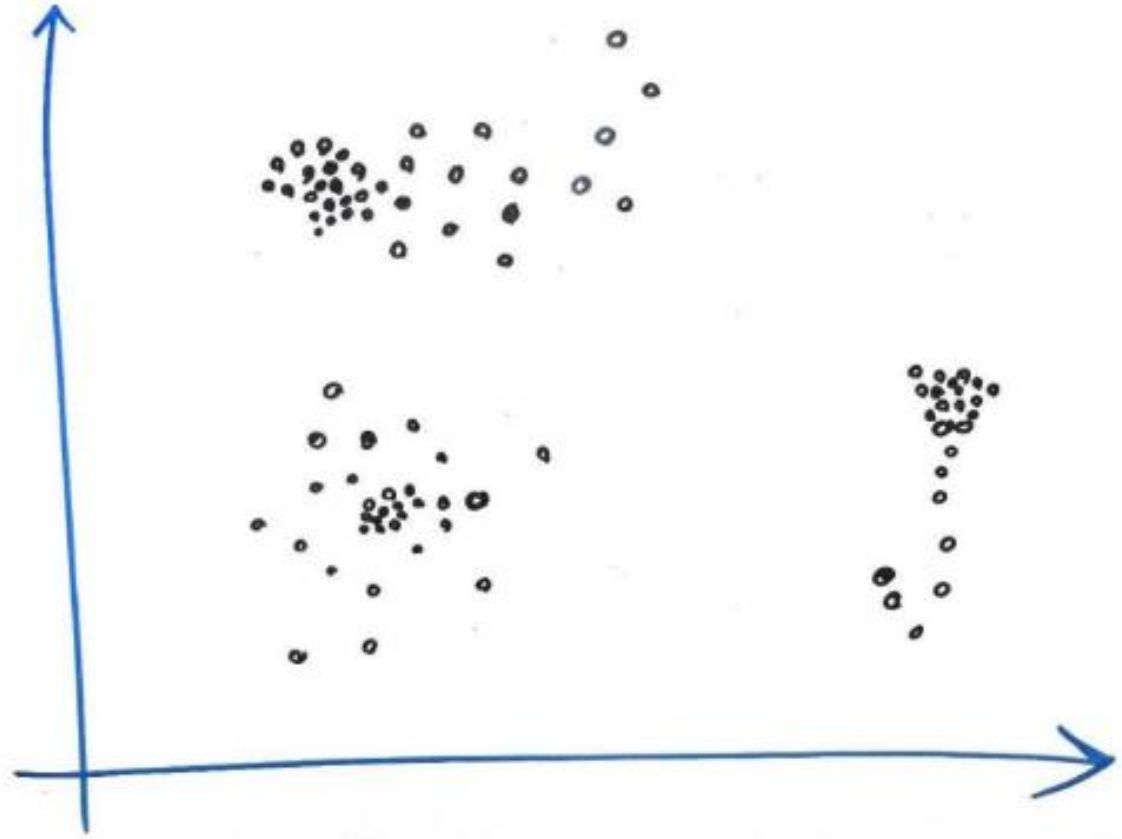


خوشه بندی– الگوریتم خوشه بندی DBSCANمبتنی بر غلظت

- این نمونه در شعاع خود، کمتر از ۳مورد نمونه دارد. الگوریتم DBSCAN این نمونه را به عنوان داده‌ی پرت یا Outlier شناسایی می‌کند و به هیچ خوشه‌ای نسبت نمی‌دهد.
- البته الگوریتم بایستی تمامیِ خوشه‌ها را بسازد و تمامیِ نقاط را بررسی کند تا بتواند بفهمد یک نقطه پرت است یا َخیر.
- هر چه شعاع کوچکتر در نظر گرفته شود، خوشه‌های بیشتر و کوچکتری تشکیلی می‌شود.
- هر چه قدر MinPoint بزرگ‌تر در نظر گرفته شود، احتمال ایجادِ خوشه‌ها، کمتر می‌شود.

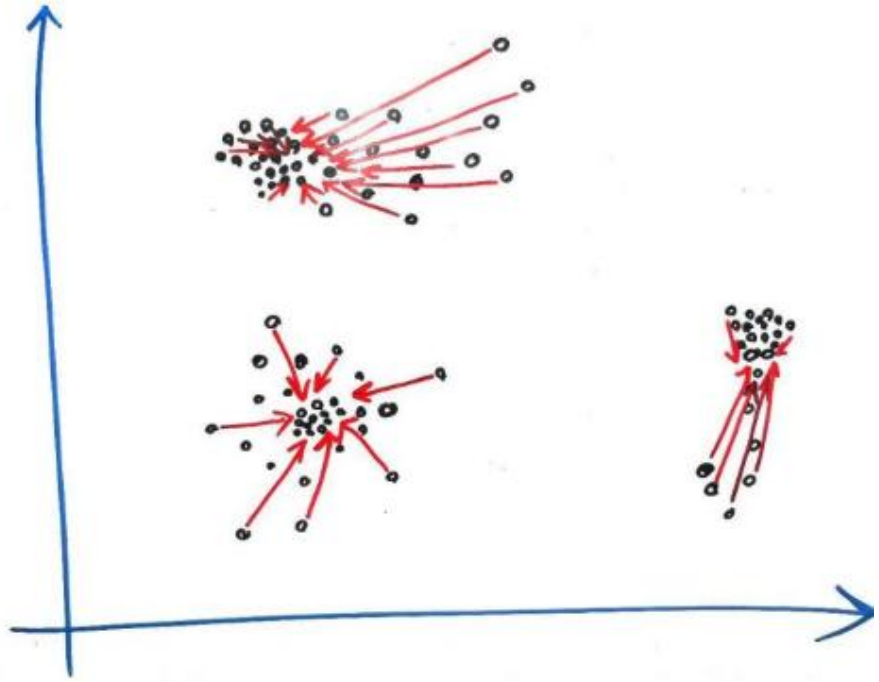
الگوریتم خوشه بندی MeanShift

- در الگوریتمِ Kmeans، کاربر در ابتدا باید تعداد خوشه‌های موجود را در این الگوریتم مشخص نماید.
- الگوریتمِ KMeans بایستی تعدادِ نقاطِ اولیه‌ی خود را بداند تا به وسیله‌ی آن بتواند خوشه‌ها را بسازد.
- برای غلبه بر این مشکل می‌توان از الگوریتمِ MeanShift استفاده کرد.

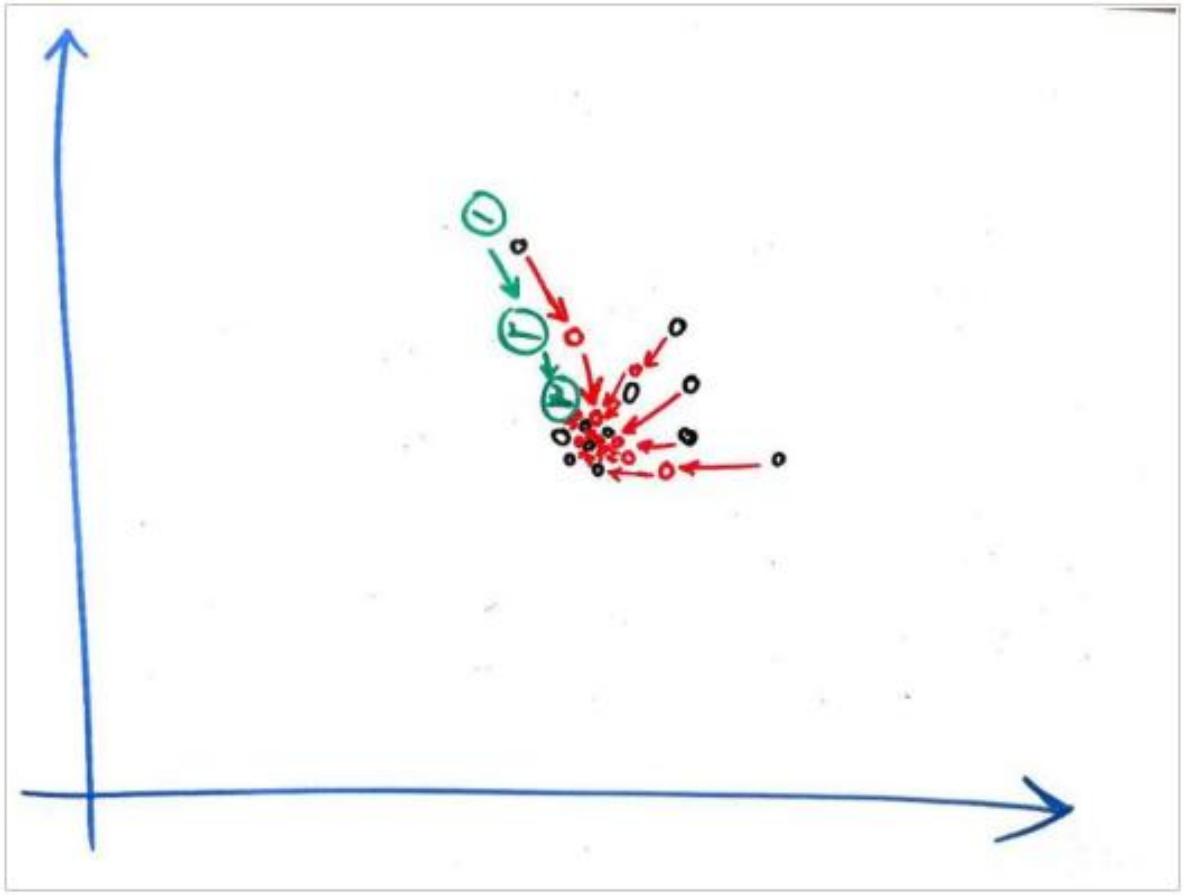


الگوریتم خوشه بندی MeanShift

- در هر گروه یک سری نقاط وجود دارند که تجمُعِ بالاتری در آن‌ها است(غلظت بالاتری دارند).
- کارِ اصلیِ الگوریتمِ MeanShift پیدا کردنِ این نقاطِ با تجمُعِ بالا در هر خوشه است تا با کمکِ آن بتواند خوشه‌های مختلف را پیدا کند.
- عملکرد الگوریتمِ MeanShift مانند این است که هر نمونه(نقطه) در فضا را به آرامی و در تکرارهای مختلف، به سمت نقطه‌ای با تجمُعِ بیشتر در نزدیکی خود حرکت می‌دهد تا سرانجام همه‌ی نقاط در یک‌جا(یا تقریبا یک‌جا) جمع شوند. برای مثال شکل زیر را نگاه کنید:



الگوریتم خوشه بندی MeanShift



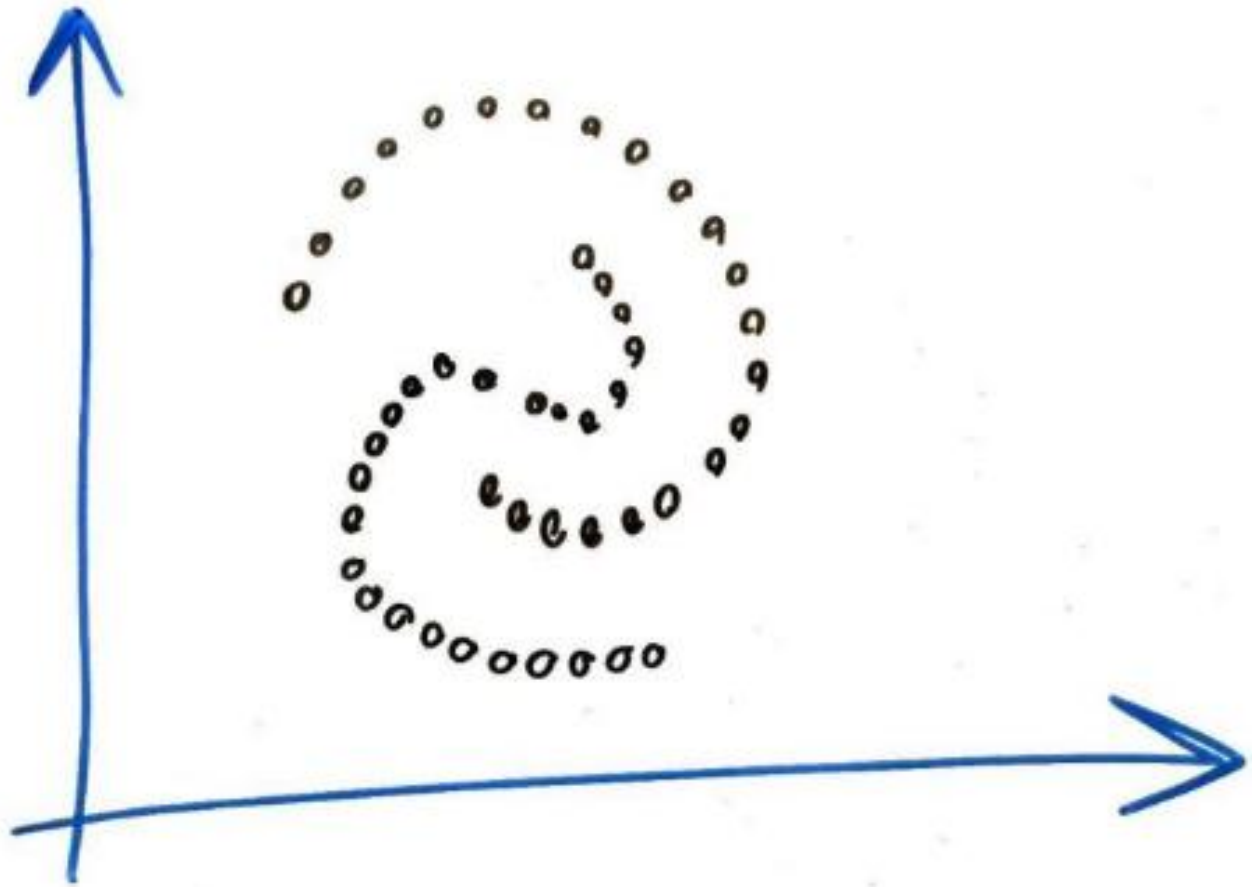
- همان‌طور که می‌بینید، هر کدام از نقاط در هر گروه تمایل دارند به نقطه‌ای که تجمُع بیشتری در آن قرار دارد(یعنی غَلیظ‌تر است) حرکت کنند.
- در نقطه ای که تجمُع بیشتر است، مقدار PDF(Probability Density Function) بالاتری داریم و الگوریتم می‌خواهد نمونه‌ها(نقاط) را آرام آرام به این نقطه که تجمُعِ زیادی دارد همگرا کند.

الگوریتم خوشه بندی MeanShift

- **الگوریتم آنقدر ادامه پیدا می‌کند که تمامیِ نقاط به یک نقطه در فضا همگرا شوند و یا اینکه به تعدادِ مشخصی تکرار(دور) داشته باشد.**
- **الگوریتمِ** MeanShift **عملیاتِ خوشه‌بندی را با استفاده از میانگینِ وزنی یا همان** Weighted Arithmetic Mean **نجام می‌دهد.**
- **الگوریتم** MeanShift **در سرعت کندتر از الگوریتم** KMeans **است. زیرا پیچیدگیِ زمانی محساباتیِ این الگوریتم بیشتر از الگوریتم** KMeans **است.**

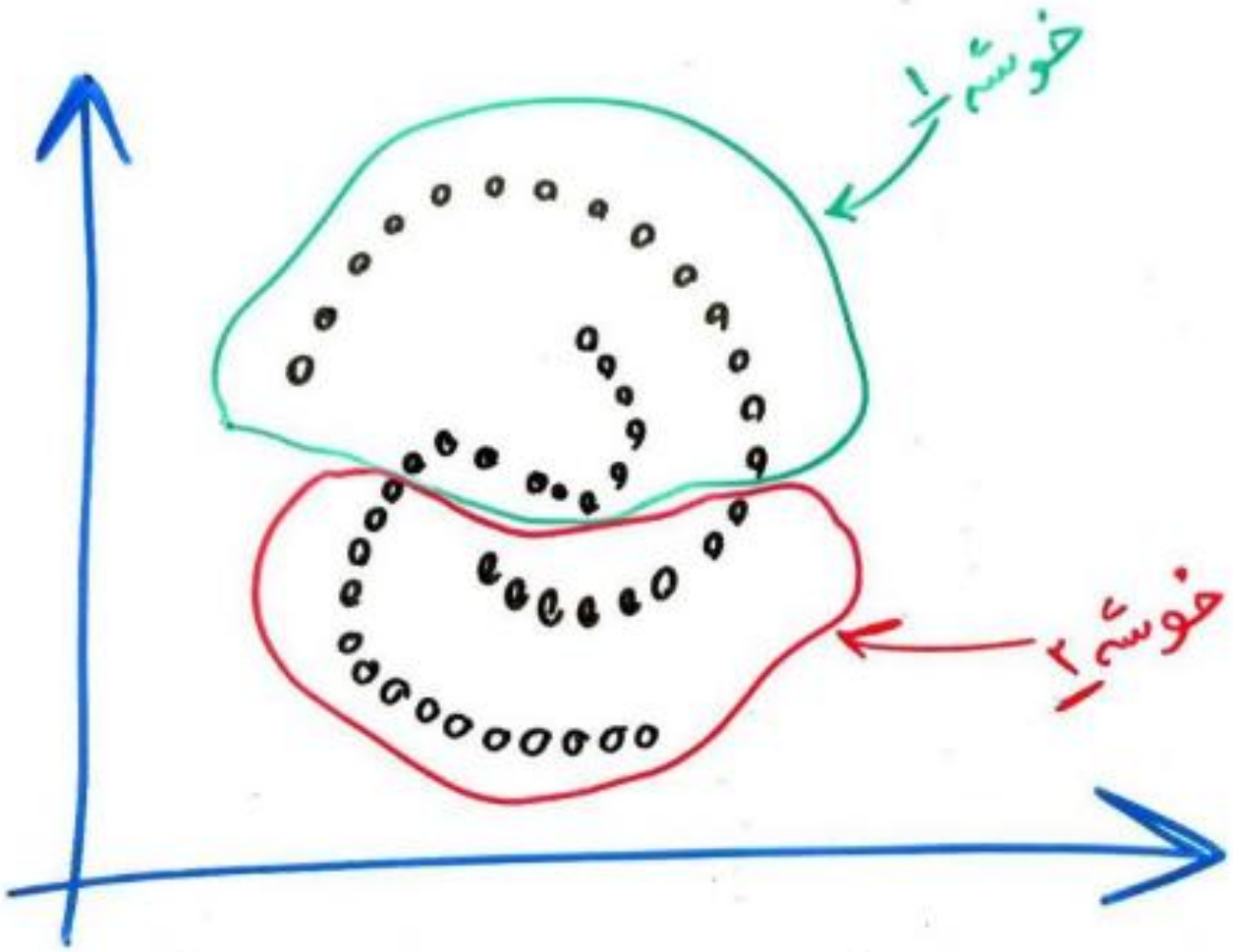
الگوریتم خوشه بندی طیفی(Spectral Clustering)

- این خوشه‌بندیْ خود در نهایت از الگوریتمی مانند KMeans استفاده می‌کند، ولی قبل از آن یک سری تغییر در ساختار داده‌ها و در واقع تغییر در نگاه خود به داده‌ها به وجود می‌آورد.



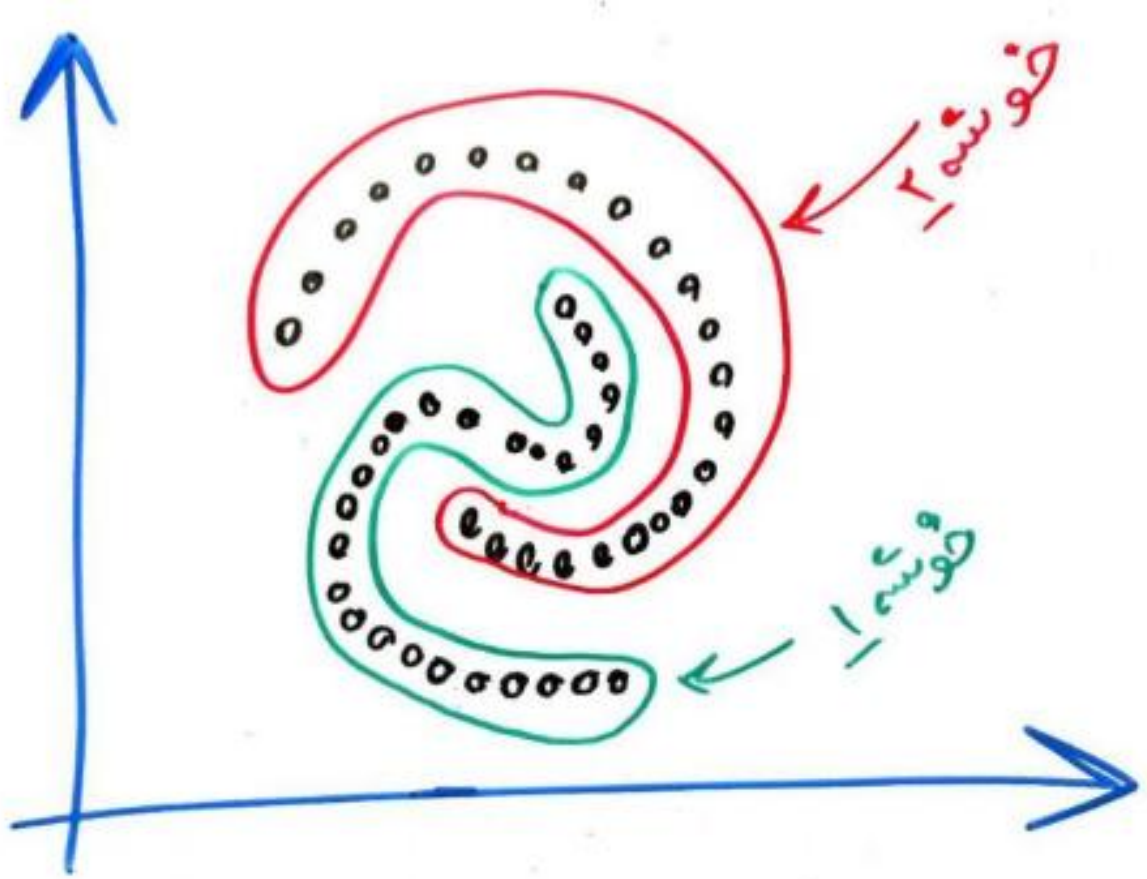
الگوریتم خوشه بندی طیفی (Spectral Clustering)

- اگر بخواهیم شکل بالا را با استفاده از الگوریتم KMeans به دو خوشه تبدیل کنیم، احتمالاً به دو خوشه‌ی زیر می‌رسیم:

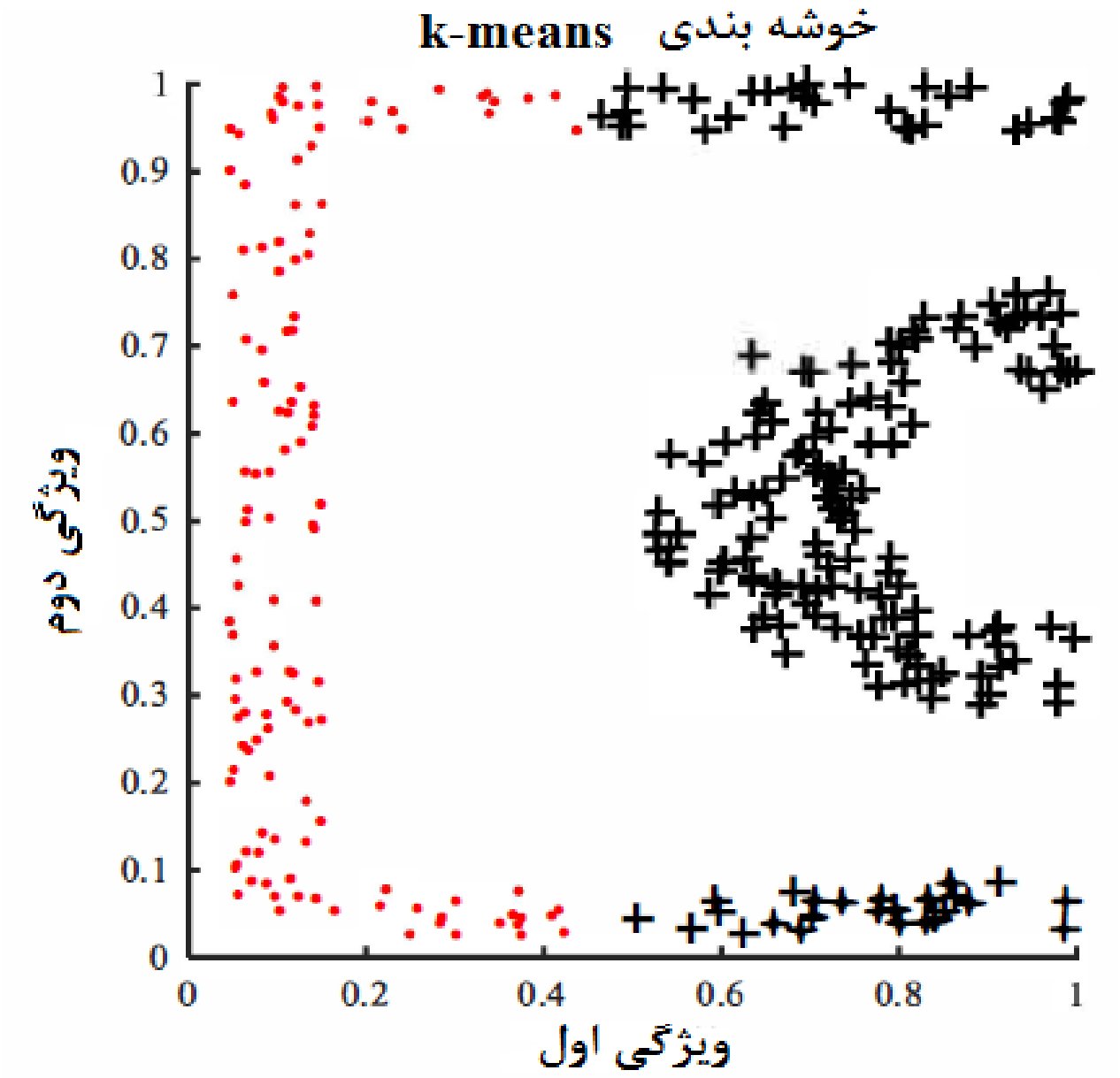


الگوریتم خوشه بندی طیفی(Spectral Clustering)

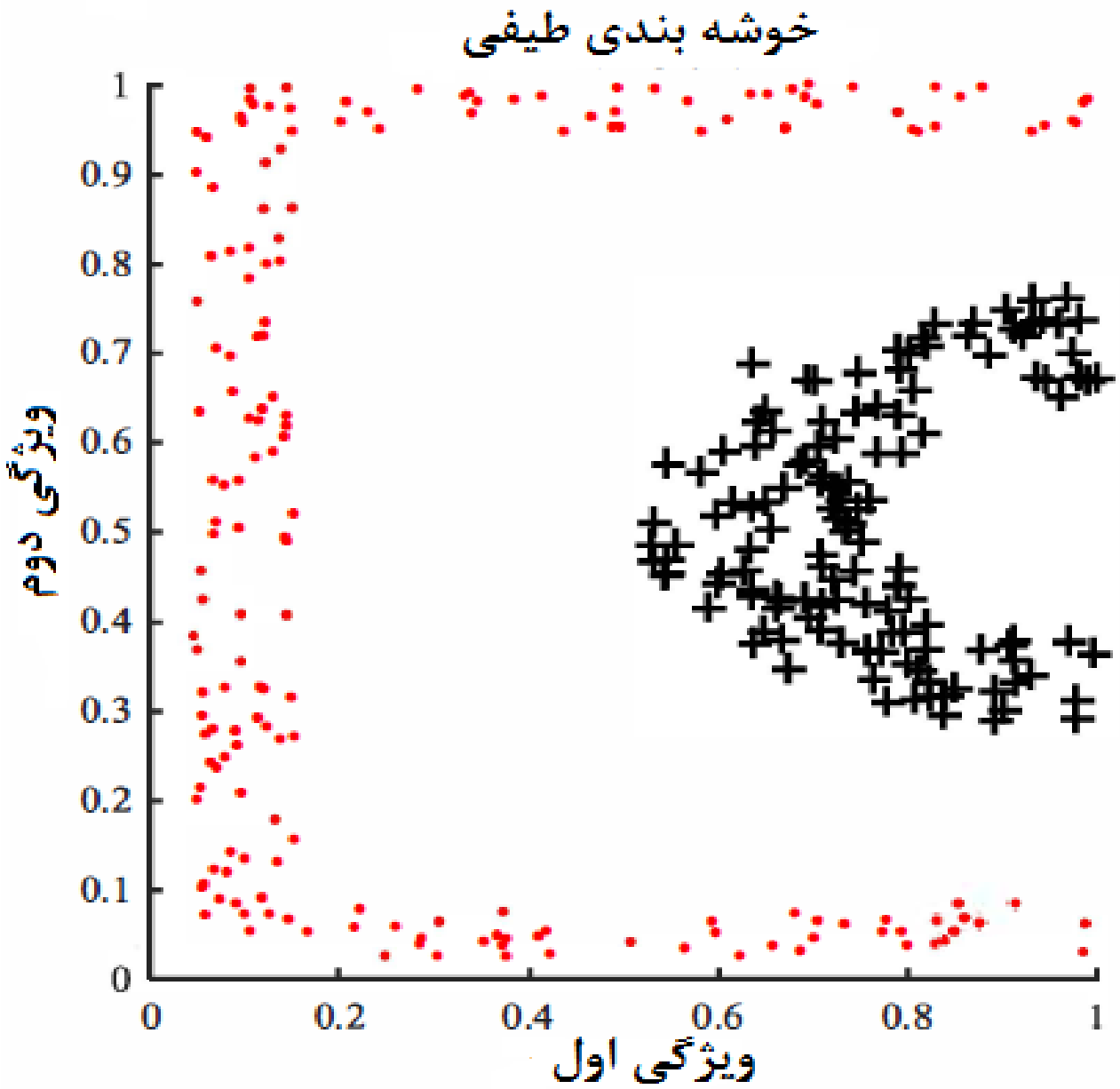
- البته می توان شکل زیر را به صورت زیر خوشه‌بندی کنیم:



الگوریتم خوشه بندی طیفی(Spectral Clustering)



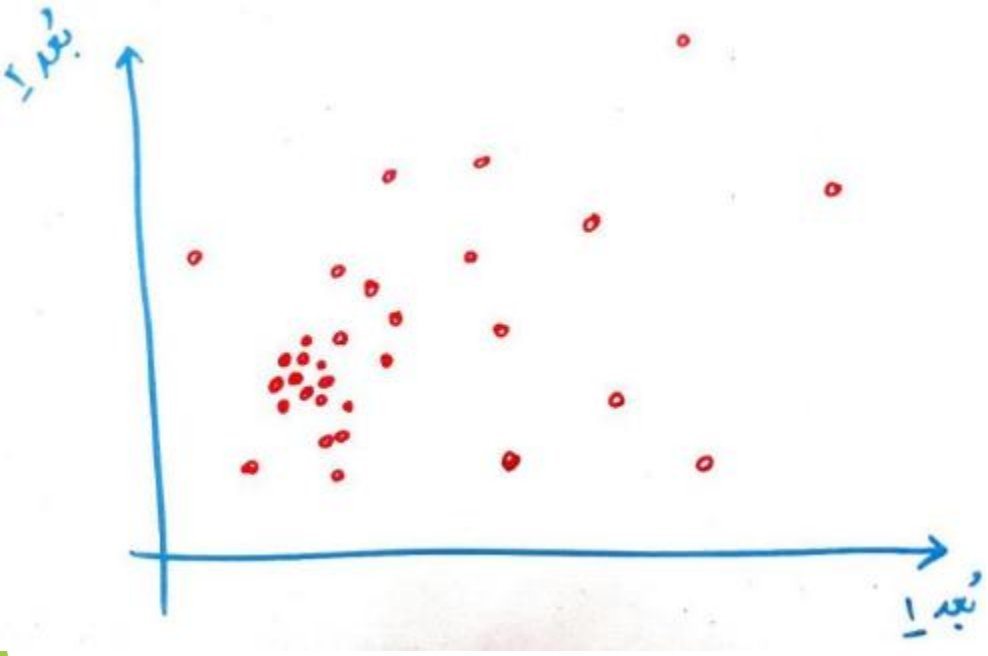
الگوریتم خوشه بندی طیفی(Spectral Clustering)



الگوریتم خوشه بندی طیفی(Spectral Clustering)

- این الگوریتمِ خوشه‌بندی باعث می‌شود که خوشه‌ها به صورت شکل‌هایی ساخته شوند که نقاطِ نزدیک و متصل به هم در یک خوشه قرار گیرند.
- این الگوریتم ابتدا یک ماتریس وابستگی(Affinity Matrix) می‌سازد و با ساختِ این ماتریسِ وابستگی، در واقع مسئله‌ی ما به یک گرافْ تبدیل می‌شود که اجزای به هم متصلِ گرافْ تشکیلِ یک خوشه را با هم می‌دهند.
- در این گراف، یال هایی که عناصر آن‌ها در یک خوشه هستند وزن زیادی دارند، و برعکس یال هایی که عناصرِ آن‌ها در یک خوشه نیستند، وزن کمتری را دارند.
- بعد از آن لاپلاسینِ گراف را ایجاد کرده بردارهای ویژه آن را انتخاب می‌کنیم.
- در آخر با الگوریتمی مانند KMeans از میانِ بردارهای ویژه می‌توان به خوشه‌بندی‌های مورد نظر دست پیدا کرد.
- البته در نهایت این الگوریتم نیز نیاز به گرفتن تعدادِ موردِ انتظار خوشه‌ها از کاربر دارد.

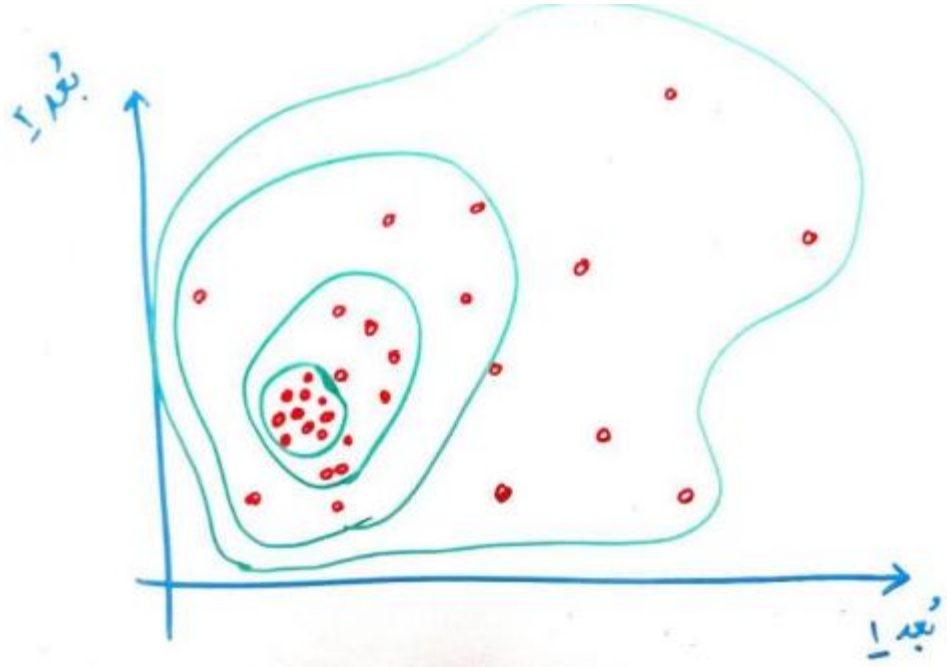
خوشه بندی با Gaussian Mixture Model و الگوریتم EM



- این الگوریتم خودْ از روشِ بیشینه سازی انتظار(Expectation Maximization) یا همان EM استفاده می‌کند .
- با رسم نمودار هیستوگرام، مشاهده می نماییم که اکثر پدیده ها دارای توزیع گوسی هستند. یعنی فراوانی حول یک مقدار بیشتر و با دور شدن از آن مقدار این فراوانی نیز کمتر می شود.
- در این تصویر یک منطقه داریم که در این منطقه(که دو بُعدی هست - یعنی دارای دو متغیر هستیم) توزیع گوسی وجود دارد. یعنی غلظت در جایی زیاد است و هر چه از آن منطقه دورتر شویم غلظت(و تعداد نمونه‌های نزدیک به آن نیز) کمتر می‌شود.

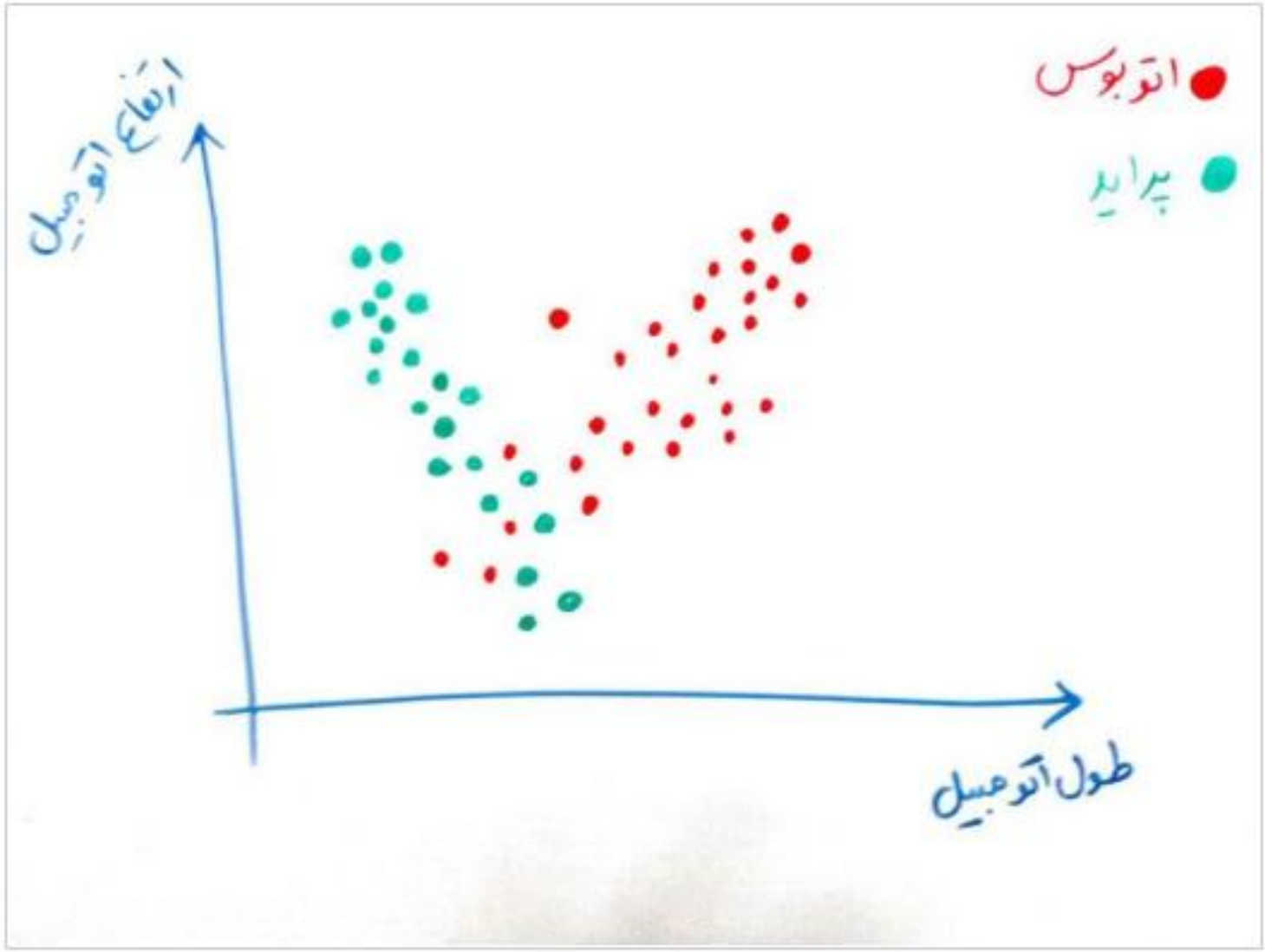
خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- دایره هایی که بیشتر در داخل قرار دارند، مناطقی هستند که تراکم(غلظت) داده‌ها در آن ها بیشتر است. مشاهده می‌کنید که هر چه از دایره‌های داخلی دورتر می‌شویم، تراکمِ داده‌ها کمتر می‌شود. در این‌جا در واقع یک ناحیه مرکزیت دارد و هر چه از این ناحیه دورترَ می‌شویم غلظتْ کمتر می‌شود.
- این یک نوع توزیعِ گوسی در ۲بُعد می‌باشد. به توزیع گوسی، توزیع نرمال نیز می‌گویند.



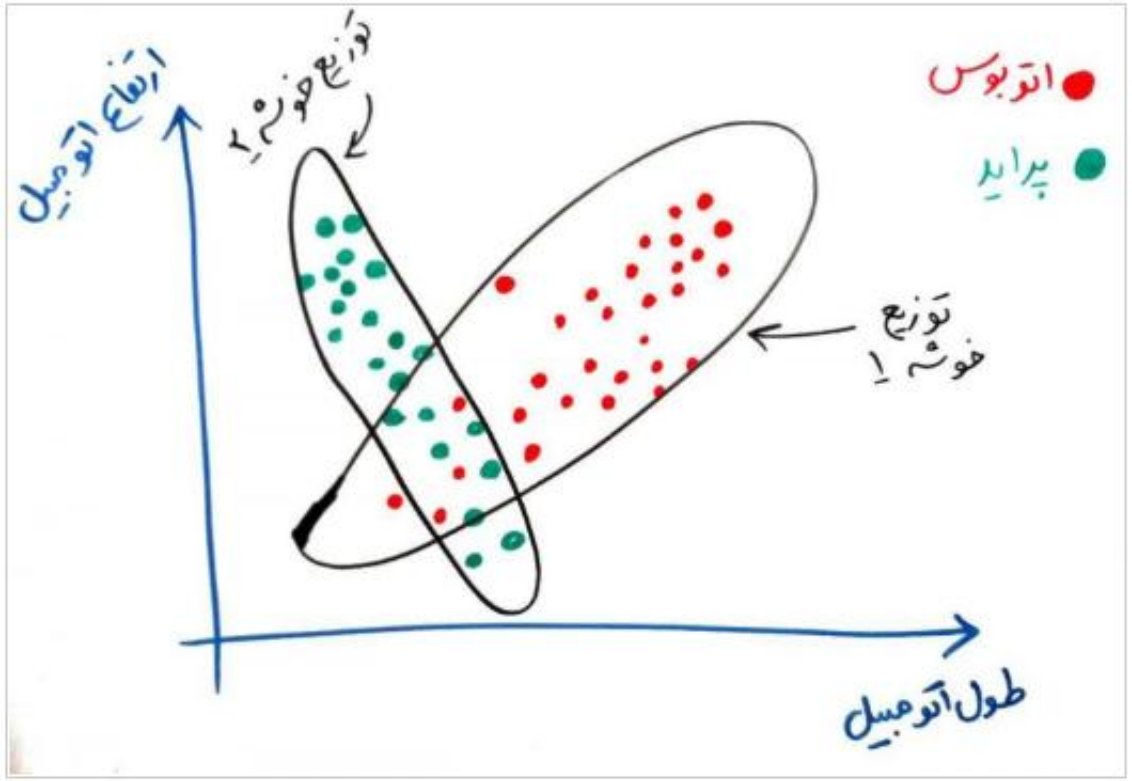
خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- فرض می شود که هدف قرار دادن داده‌های زیر را در دو خوشه باشد



خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- در این حالت ممکن است قسمتی از داده‌های خوشه اول، کاملا در میان داده‌های خوشه دوم باشد.
- الگوریتم‌های قبلی مانند KMeans و DBSCAN قادر به فهم این خوشه داده‌ها نیستند. ولی توسط مدلِ ترکیبیِ گوسی یا همان GMM می‌توان این خوشه‌ها را پیدا کرد.
- در واقع در این الگوریتم ابتدا فرض می‌کنیم که مثلا ۲خوشه داده داریم که به صورت زیر از توزیعِ گوسی پیروی می‌کنند.



خوشه بندی با Gaussian Mixture Modelو الگوریتم EM

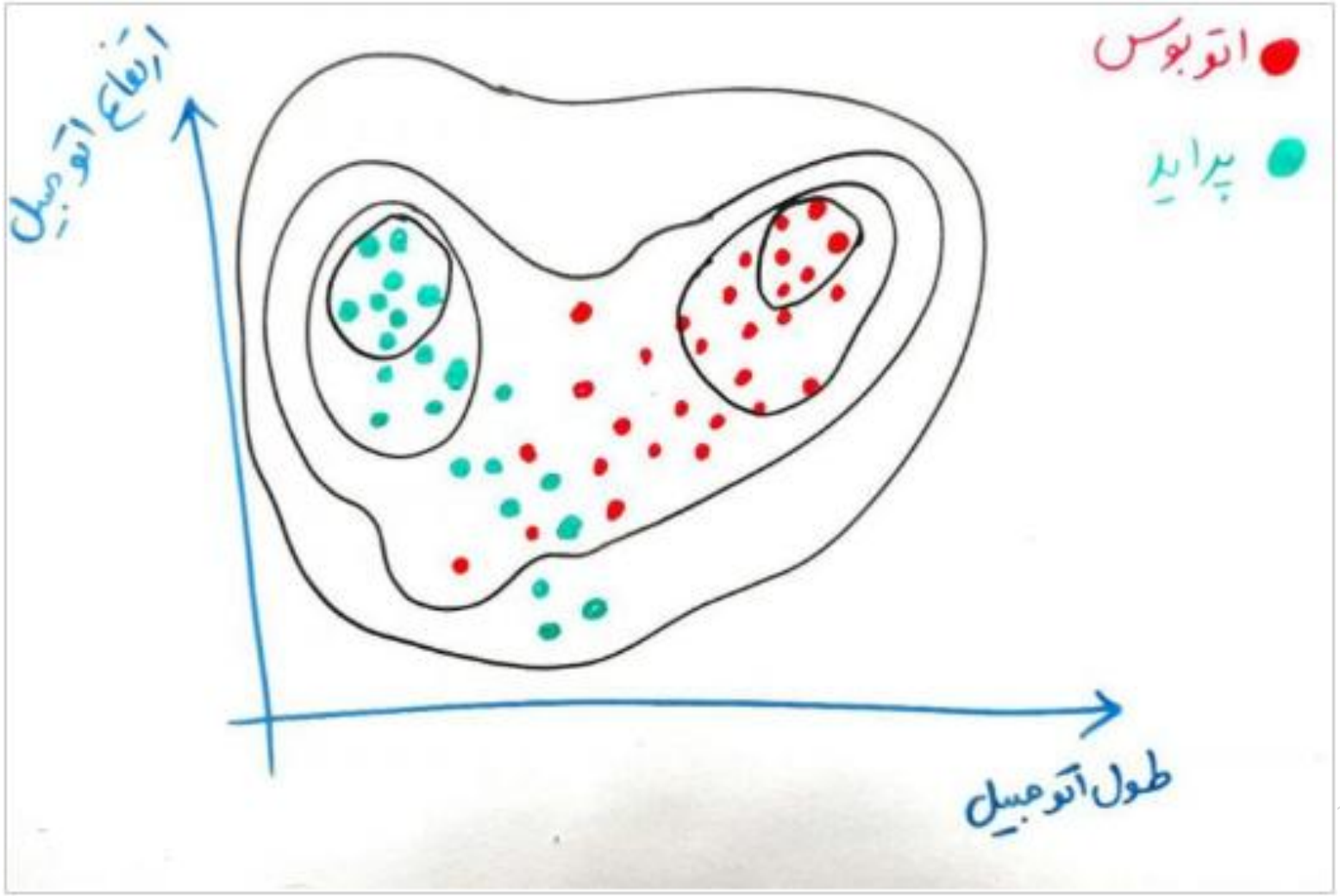
- برای پیدا کردن دو خوشه که از توزیع گوسی استفاده می‌کنند و با استفاده از مدلِ مُولد به دنبال پیدا کردنِ پارامترهایی برای برای توصیف گوسیِ داده‌ها می‌گردیم.
- این پارامترها با استفاده از الگوریتم Expectation Maximization تعیین می شود.
- این الگوریتم ۲ بخش دارد:
- بخش اول Expectation یا همان انتظار است، که در این قسمت، الگوریتم می‌خواهد ببیند که هر کدام از نمونه ها(نقاط) به کدام توزیعِ گوسی بیشتر نزدیک هستند و در واقع احتمالِ عضویتِ یک نمونه(نقطه) را به تابع گَوسی پیدا کند.
- بحث EMدر الگوریتم KMeans نیز وجود دارد.
- در هر دوراز این الگوریتم، هر کدام از نمونه‌ها به نزدیک ترین مرکز خوشه تعلق پیدا می‌کردند.
- این یعنی الگوریتم انتظار دارد که این نمونه(نقطه) به یک خوشه‌ی خاص تعلق داشته باشد.

خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- حال در GMM در هر بار تلاش، الگوریتم انتظار دارد تا احتمالِ عضویتِ یک نمونه(نقطه) را به هر کدام از توزیع های گوسی مورد نظر نسبتَ دهد.
- بخش دومِ روشِ EM در واقع Maximization یا بیشینه‌سازی است.
- در الگوریتمِ KMeans در هر دور نیاز بود که مرکزِ خوشه را تغییر دهیم تا شباهتِ مرکزَ خوشه با تمامی نمونه ها(نقاط) داخل آن خوشه بیشینه شود. در الگوریتَم GMMنیز پارامترها که شامل وزن، میانگین و covariance است در هر دور به روز رسانی می شود تا در نهایت توزیع گوسی برای هر کدام از خوشه ها تشکیل شود.
- در واقع شباهت توزیع داده‌ها به صورت گوسی بیشینه شود.

خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- بعد از چندین بار تکرار مطلوب است که توابع گوسی مانند شکل زیر ایجاد شود:



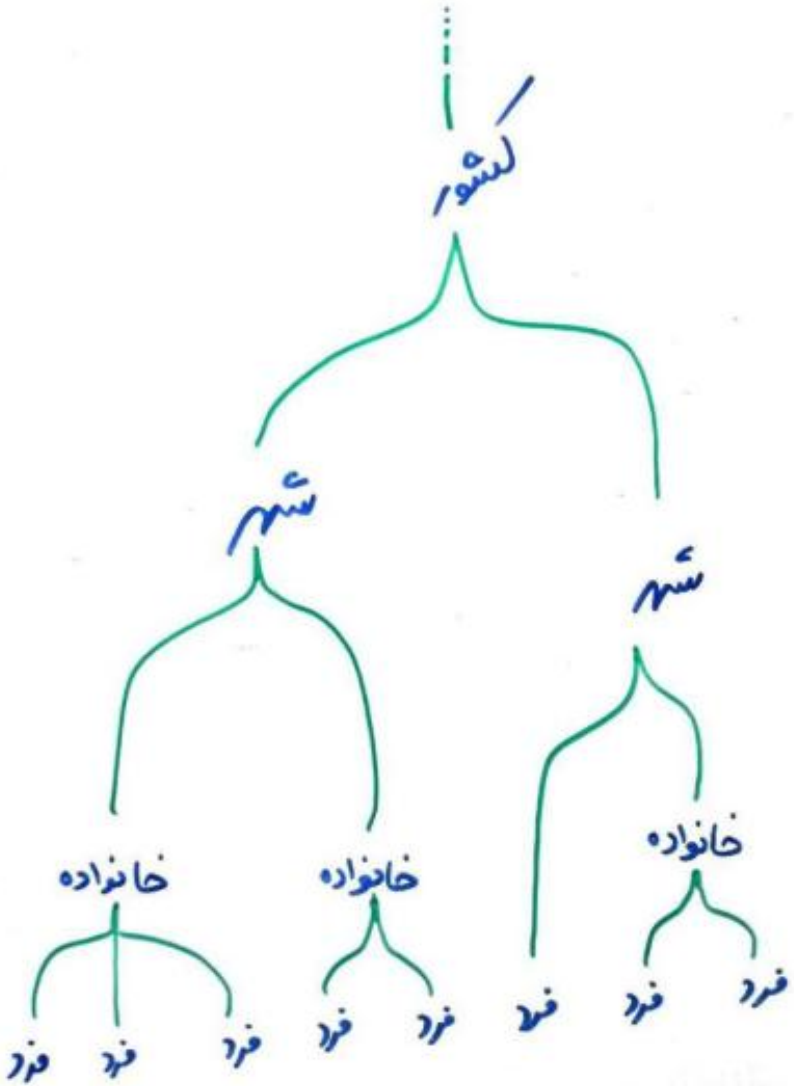
خوشه بندی با Gaussian Mixture Model و الگوریتم EM

- الگوریتم GMM دوخاصیت بسیار مهم نسبت به الگوریتمی مانند KMeans دارد:
- این الگوریتم می‌تواند خوشه هایی غیر کروی(دایره ای) را پیدا کند در حالیکه الگوریتم KMeans می‌تواند خوشه هایی کروی(دایره ای) را بیابد و این می‌تواند نقطه ضعف KMeans باشد، ولی در GMM با توجه به ساختار و توزیعِ گوسیْ عملیاتِ خوشه بندی انجام می‌شود.
- همچنین در الگوریتم GMM هر نمونه(نقطه) می‌تواند به چند خوشه تعلق پیدا کند(به نسبت عضویت در ساخت توزیع گوسی آن خوشه).
- در GMM ما نوعی خوشه بندی نرم(Soft Clustering) داریم که هر نمونه(نقطه) می‌تواند به بیش از یک خوشه تعلق داشته باشد. ولی در KMeans خوشه‌بندی یک خوشه‌بندیِ سخت است((Hard Clustering، به این معنی که هر نمونه(نقطه) فقط می‌تواند بهَ یک خوشه تعلق داشته باشد.

خوشه بندی سلسله مراتبی(Hierarchical (Clustering)

- **این الگوریتم‌ها به دو دوسته‌ی** agglomerative **و** partitioning **تقسیم‌بندی می‌شوند:**
- **در روش** agglomerative **که به آن روش پایین به بالا نیز گفته می‌شود، هر نمونه ابتَدا خود یک خوشه است و در هر مرحله نمونه‌ها به هم دیگر می‌چسبند تا خوشه‌های بزرگ‌تر را تولید کنند و در نهایت همه‌ی نمونه‌ها با هم یک خوشه‌ی بزرگ را تشکیل می‌دهند.**
- **در** partitioning **بر عکس است، یعنی ابتدا تمامیِ نمونه‌ها با هم یک خوشه‌ی بزرگ در نظر گرفته می‌شوند و بعد در هر مَرحله به خوشه‌های کوچک‌تر تقسیم می‌شوند تا جایی که هر نمونه یک خوشه باشد.**

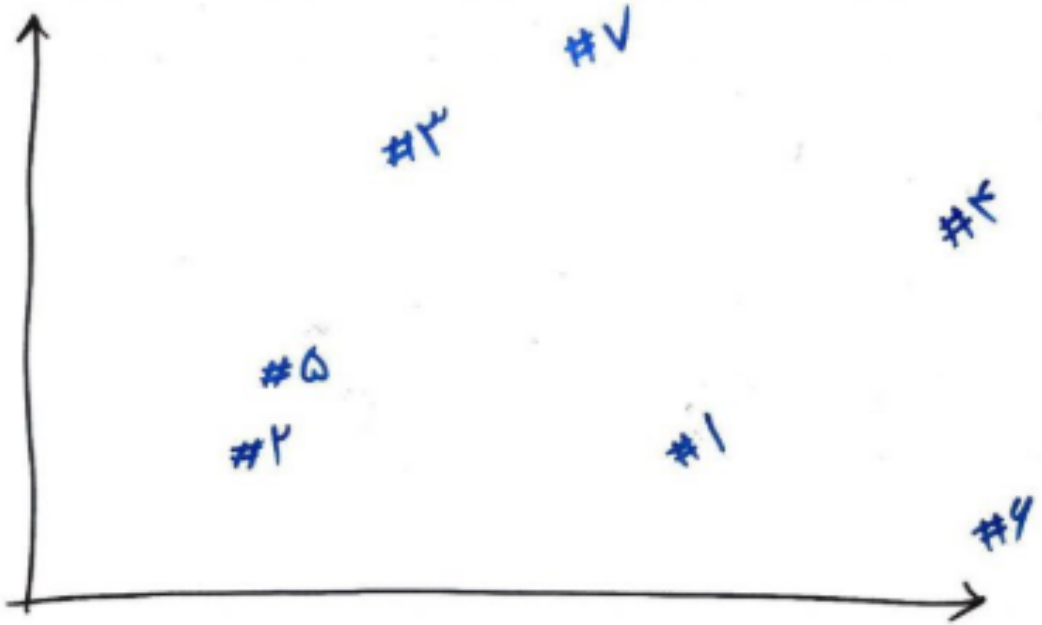
خوشه بندی سلسله مراتبی(Hierarchical Clustering)



- خوشه ها از پایین به بالا بزرگ‌تر ساخته می‌شوند و از بالا به پایین، بیشتر قسمت‌بندی می‌شوند.

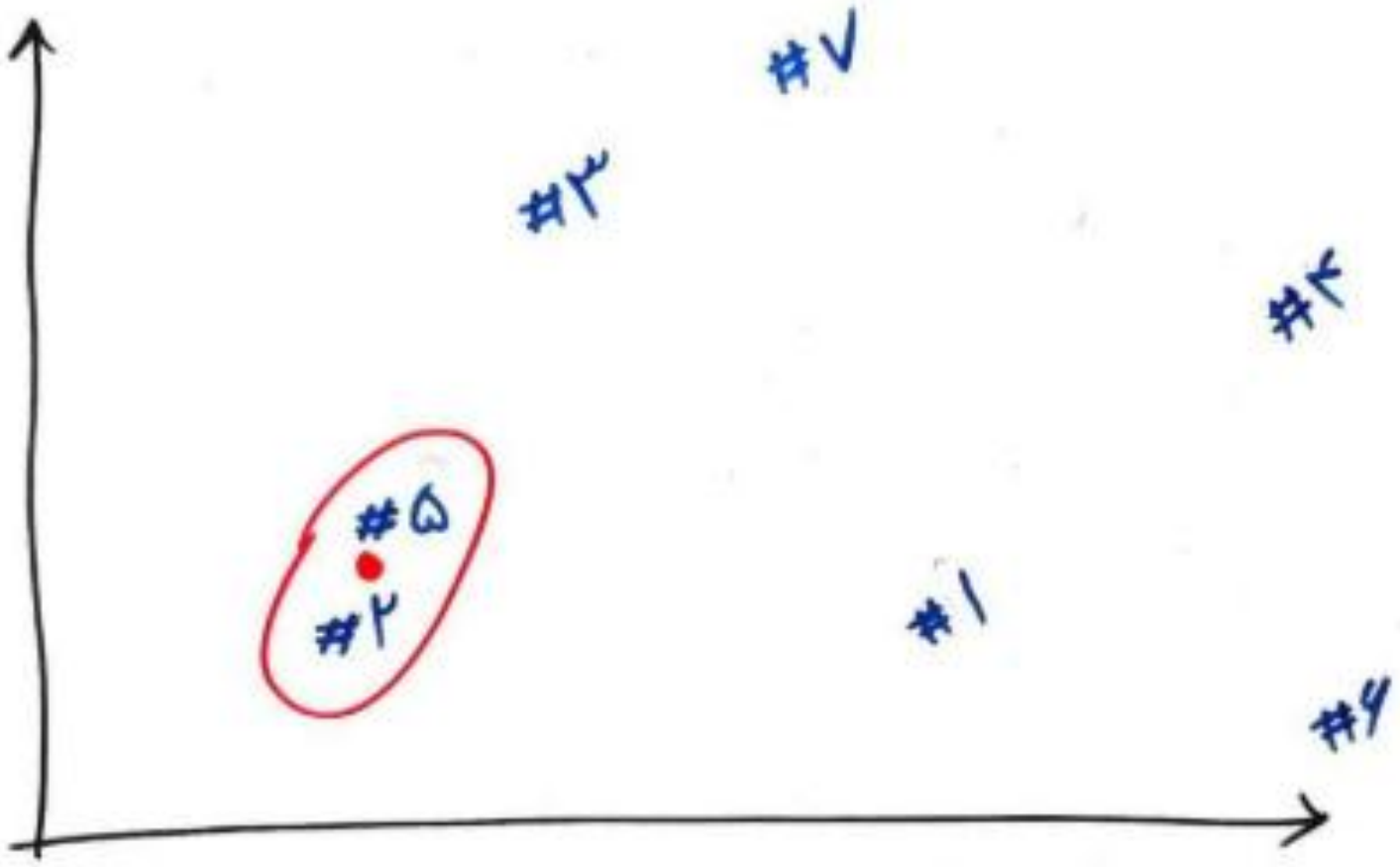
خوشه بندی سلسله مراتبی(Hierarchical Clustering)

- روند کار:
- ابتدا دو نقطه‌ای که بیشتر از همه به هم نزدیک هستند را پیدا می‌کنیم و این دو نقطه را با هم یک خوشه در نظر می گیریم
- دوباره میانگین این دو نقطه را یک نقطه در مرکز خوشه در نظر می‌گیریم و به دنبال دو خوشه نزدیک به هم می‌گردیم تا آن‌ها را تبدیل به یک خوشه کنیم.
- این کار را آنقدر انجام می‌دهیم تا تمامیِ نقاط در نهایت یک خوشه واحد شوند.



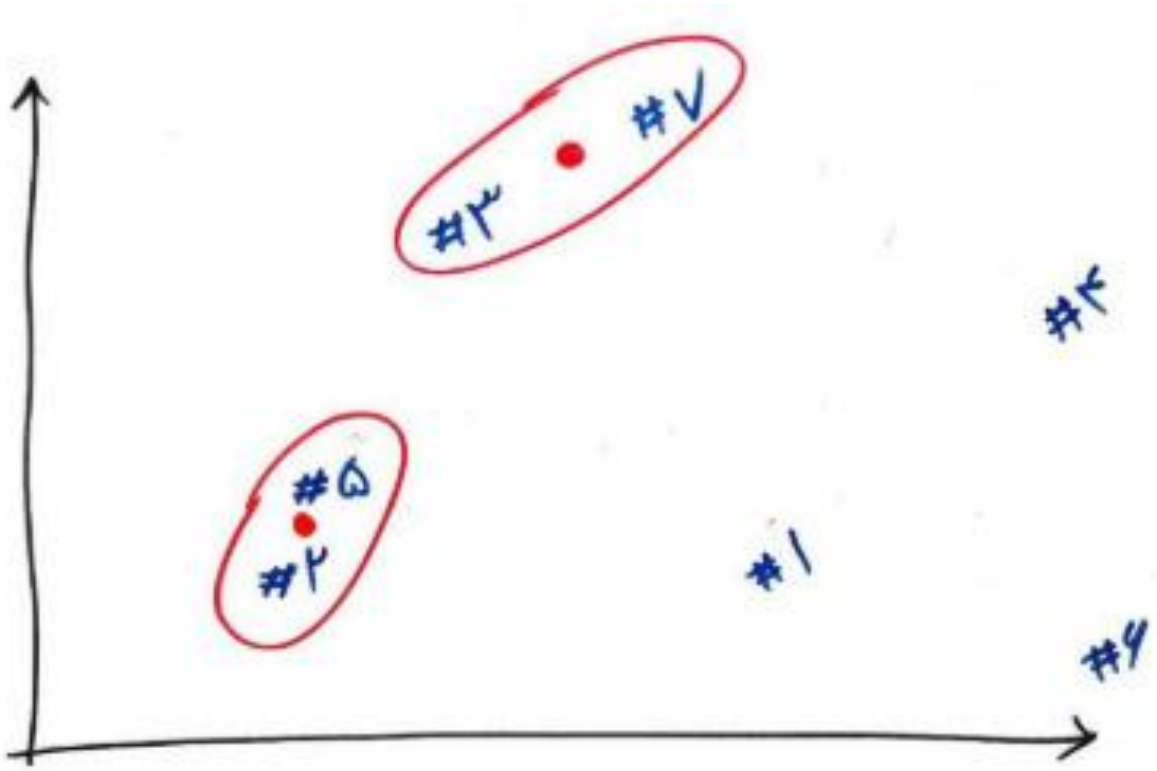
خوشه بندی سلسله مراتبی(Hierarchical Clustering)

- در این شکل دو نمونه ۲ و ۵ به همدیگر نزدیک‌تر هستند. پس این دو را یک خوشه در نظر می‌گیریم و یک نقطه در میان این خوشه به عنوان مرکز خوشه(جهت مقایسه های بَعدی) می‌سازیم. مانند شکل زیر:



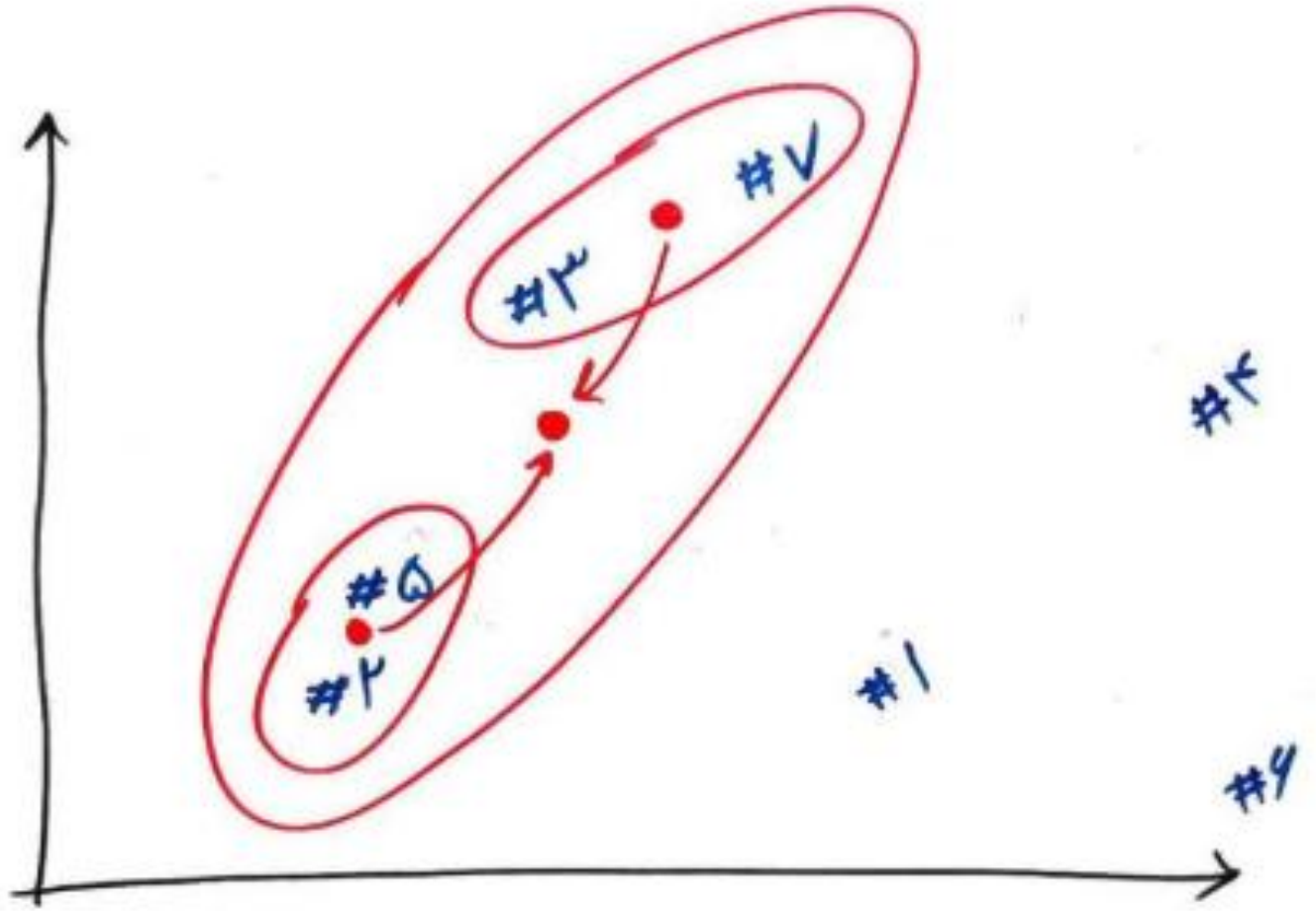
خوشه بندی سلسله مراتبی(Hierarchical Clustering)

- حال این نقطه جدید(مرکز خوشه‌ای که از دو نمونه ۲ و ۵ ساخته شده بود) را با نقاط دیگر مقایسه می‌کنیم. دوباره به دنبال نقاط نزدیک به هم(با توجه به خوشه جدید ایجاد شده) می‌گردیم.
- مشاهده می‌شود که نقاط ۳ و ۷ به یکدیگر نزدیک‌تر هستند. این دو نقطه را با هم به یک خوشه تبدیل کرده و نقطه‌ی مرکزیِ آن‌ها را محاسبه می‌کنیم.



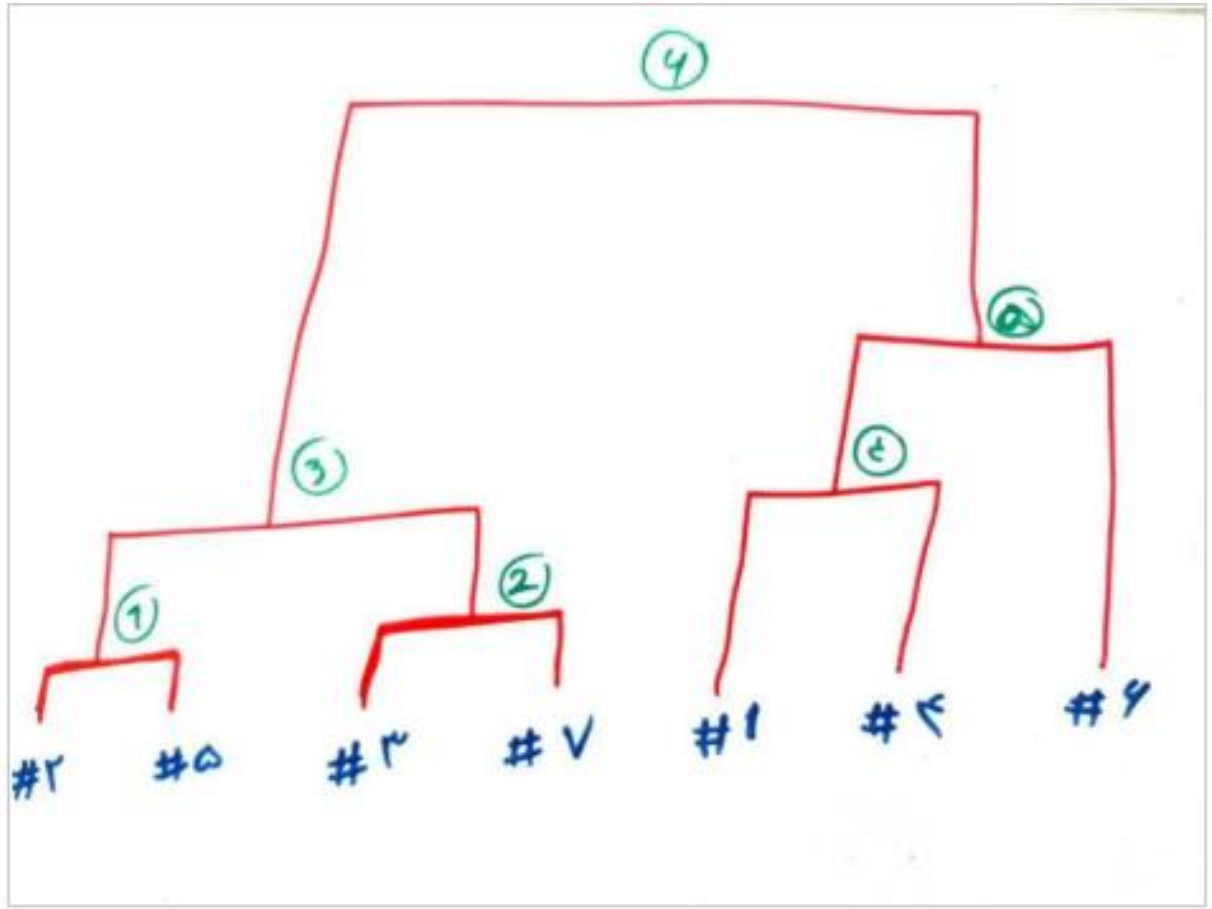
خوشه بندی سلسله مراتبی(Hierarchical Clustering)

- مشاهده می شود که برای دور بعد می‌توانیم مرکزهای خوشه های ایجاد شده را با هم ترکیب کنیم و یک خوشه جدیدتر بسازیم. شکل زیر تجمیع شده‌ی دو خوشه قبلی(خوشه های ۲ - ۵ و ۳ - ۷) برای خوشه جدید است:



خوشه بندی سلسله مراتبی(Hierarchical Clustering)

- در ابتدا ۷خوشه داریم، در مرحله‌ی اول نمونه‌های ۲ و ۵ با هم تجمیع می‌شوند و در مرحله دوم نمونه‌های ۳ و ۷. به همین ترتیب جلو می‌رویم تا در مرحله ۶ تمامیِ خوشه‌ها با یکدیگر تجمیع شده و یک خوشه‌ی واحد را می‌سازند.
- معمولاً شرط پایان می‌تواند این باشد که الگوریتم به یک تعداد مشخص خوشه برسد.

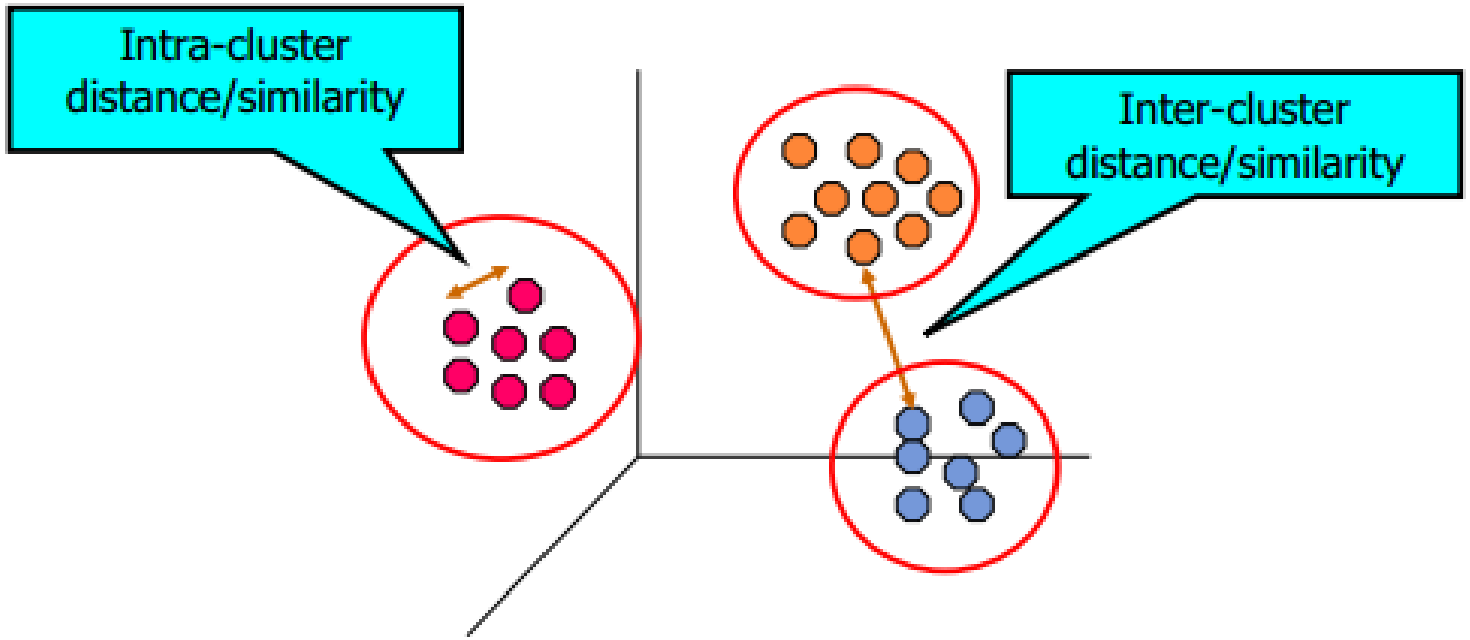


فصل چهارم

ارزیابی خوشه بندی

بررسی معیارهای ارزیابی خوشه‌بندی

- **معیارهای داخلی**
- **داده های هر خوشه باید بیشترین شباهت را به یکدیگر داشته باشند.**
- **داده های هر خوشه باید کمترین شباهت را به داده های سایر خوشه ها داشته باشند.**



بررسی معیارهای ارزیابی خوشه‌بندی

- **معیارهای خارجی**
  - از یک مجموعه آزمون دارای پرچسلب استفاده می کند
  - ارزیابی بر اساس اینکه داده های هر دسته در یک خوشه قرار گرفته اند یا نه؟
  - استفاده از آنتروپی یا معیارهای ارزیابی دسته بندی
- **معیارهای وابسته به کاربرد**
  - اگر هدف از خوشه بندی بهبود سرعت جستجو است چقدر برای این هدف موفق بوده است
  - پر هزینه است

بررسی معیارهای ارزیابی خوشه‌بندی

- **شباهت درون خوشه ای**
- **مطلوب ان است که داد۹اب دانشیار باهت ببسزی**
- **بیانگر cluster cohesion است.**
- **برای شباهت می توان از توابع فاصله استفاده نمود.**

$$Cohesion(C_k) = \sum_{x,y \in C_k} Similrity(x, y)$$

نمونه های متعلق به یک خوشه

بررسی معیارهای ارزیابی خوشه‌بندی

- معیار sum of square erroe (SEE)
- جمع مربعات فاصله بین همه نمونه های یک خوشه با مرکز آن خوشه
- هرچه فاصله (SEE)کمتر باشد شباهت بیشتر است
- فاصله می تواند هر معیاری از جمله فاصله اقلیدوسی نیز باشد

$$SSE(C_k) = \sum_{x \in C_k} (Dist(\mu_k, x))^2$$

مرکز خوشه

$$SSE = \sum_{k=1}^{K} SSE(C_k) = \sum_{k=1}^{K} \sum_{i=1}^{|C_k|} (Dist(\mu_k, x_i))^2$$

اندازه خوشه = تعداد نمونه ها

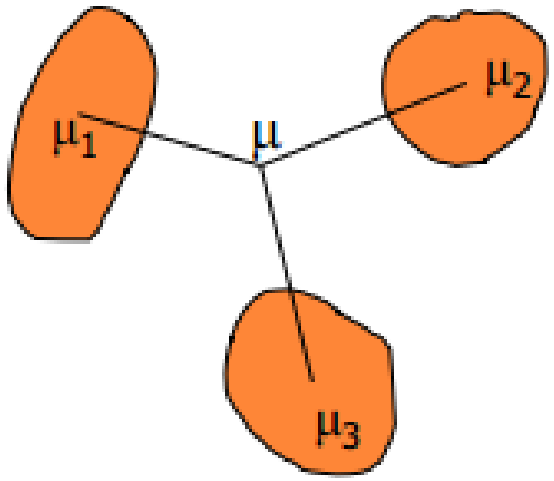
بررسی معیارهای ارزیابی خوشه‌بندی

- **شباهت بین خوشه ای**
- **مطلوب ان است که شباهت کمتر باشد**
- **بیانگر cluster separation است.**
- **برای شباهت می توان از توابع فاصله استفاده نمود.**

$$Separation(C_i, C_j) = \sum_{\substack{x \in C_i, \\ y \in C_j}} Similarity(x, y)$$

نمونه های متعلق به دو خوشه مختلف

بررسی معیارهای ارزیابی خوشه‌بندی



- معیار total sum of squares (TSS)
  - محاسبه مرکز کلی خوشه ها
  - بیانگر فاصله همه نقاط از مرکز کلی

$$TSS = \sum_{i=1}^{N} \left(Dist(\mu, x_i)\right)^2$$

- معیار sum of square between (SSB)
  - فاصله مراکز خوشه ها از مرکز کلی
  - هر اندازه بزرگتر باشد بهتر است

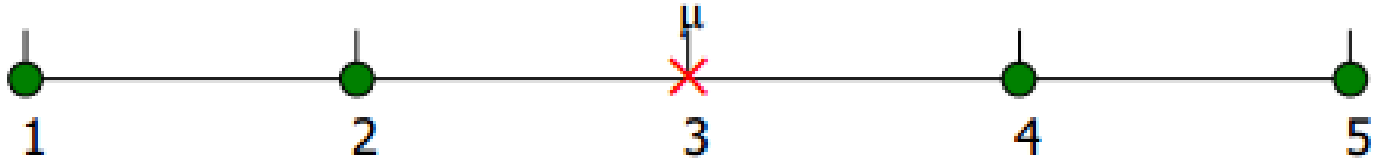
$$SSB = \sum_{k=1}^{K} |C_k| \left(Dist(\mu_k, \mu)\right)^2$$

- معیار TSS=SSE+SSB
  - برای یک مجموعه داده TSS ثابت است
  - اگر فاصله درون خوشه ای (SSE) افزایش یابد، SSB کاهش می یابد و برعکس

بررسی معیارهای ارزیابی خوشه‌بندی

مثال

- ۴ نمونه داده
- با یک خوشه



- یک خوشه با مرکز ۳

$$TSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$SSE = (3-1)^2 + (3-2)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$SSB = 4 \times (3-3)^2 = 0$$

- با دو خوشه

- دو خوشه با مراکز ۱.۵ و ۴.۵
- مرکز کلی = ۳

$$TSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$SSB = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

- داریم $TSS = SSE + SSB$

بررسی معیارهای ارزیابی خوشه‌بندی

ضریب نیمرخ (Silhouette coefficient)

ترکیب شباهت درون خوشه ای و بین خوشه ای

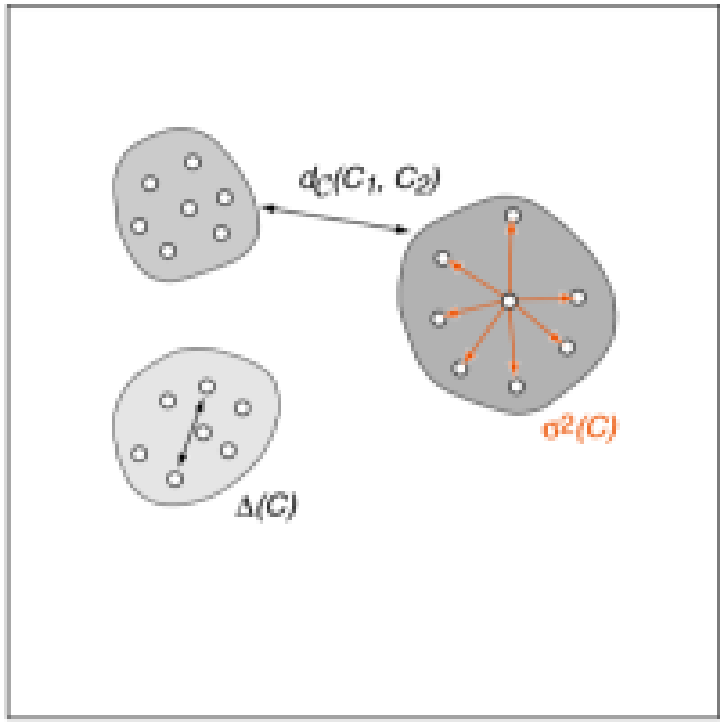
- محاسبه برای یک نمونه داده مانند x_i
  - گام ۱: محاسبه متوسط فاصله داده x_i از تمام داده‌های دیگر در خوشه خودش = a_i
  - گام ۲: محاسبه متوسط فاصله داده x_i از تمام داده‌های دیگر در $K-1$ خوشه دیگر (برای هر خوشه یک مقدار بدست می‌آید). کمترین مقدار بدست آمده از بین $K-1$ متوسط فاصله محاسبه شده را انتخاب کن = b_i
  - گام ۳: ضریب نیمرخ $s_i=(b_i-a_i)/\max(b_i,a_i)$

- داریم $-1< s_i <1$
  - مقدار منفی حالت نامناسب است (فاصله نمونه از سایر خوشه ها از خوشه خودش کمتر است)
  - حالت ایده آل: مقدار $a_i=0$ و در نتیجه $s_i=1$

- بررسی مناسب بودن یک روش خوشه‌بندی: محاسبه متوسط s_iها برای کل داده‌ها

بررسی معیارهای ارزیابی خوشه‌بندی

- **معیار همبستگی**
  - **هر اندازه مقدار I(C) بیشتر باشد بهتر است**



$$I(C) = \frac{\min_{i \neq j}\{d_C(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}}$$

بررسی معیارهای ارزیابی خوشه‌بندی

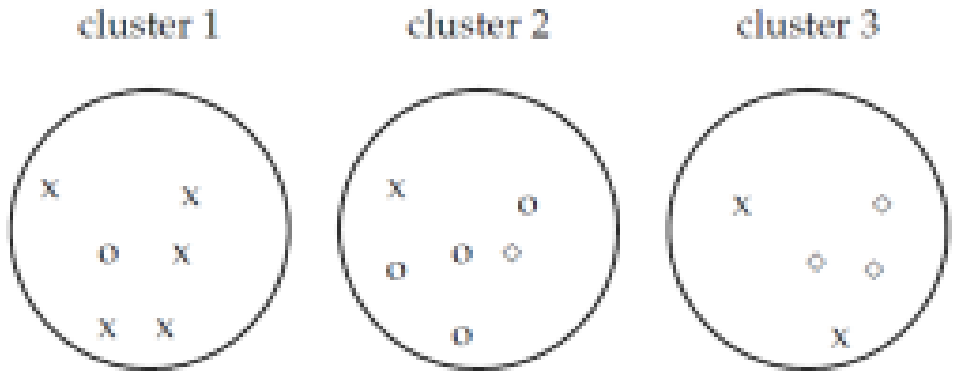
- **خالص بودن خوشه ها**
- دسته های واقعی هر نمونه داده مشخص اشت
- پس از خوشه بندی به هر نمونه یک برچسب می زنیم، برچسب دسته ای که بیشترین تعداد داده از آن دسته در یک خوشه قرار گرفته است
- محاسبه درستی انتساب نمونه ها به خوشه ها
- محاسبه خالص بودن: شمارش تعداد نمونه های درست هر دسته، جمع نمودن آنها با هم و تقسیم بر تعداد کل نمونه ها
- مقدار خاالص بوودن بین۱ (خوشه بندی بهینه) و ۰(خوشه بندی بد) می باشد
- وقتی تعداد خوشه ها زیاد باشد

تعداد خوشه ها

$$Purity\ (Cluster, Class) = \frac{1}{N}\sum_{k=1}^{K} \max_{j} |Cluster_k \cap Class_j|$$

بررسی معیارهای ارزیابی خوشه‌بندی

- خالص بودن خوشه ها
- ۱۷ نمونه داریم که متعلق به سه کلاس هستند



$\max_j |Cluster_1 \cap Class_j| = 5$ $\max_j |Cluster_1 \cap Class_j| = 4$ $\max_j |Cluster_1 \cap Class_j| = 5$

$$Purity\ (Cluster, Class) = \sum_{k=1}^{K} \max_j |Cluster_k \cap Class_j| = \frac{1}{N}(5+4+3) = 0.71$$

بررسی معیارهای ارزیابی خوشه‌بندی

- **خالص بودن خوشه ها**
- **اطلاعات متقابل نرمال شده (Normalized Mutual Information)**

اطلاعات متقابل

$$NMI(Cluster, Class) = \frac{I(Cluster, Class)}{[H(Cluster) + H(Class)]/2}$$

آنتروپی

تعداد خوشه ها

تعداد دسته ها

$$I(Cluster, Class) = \sum_{k=1}^{K}\sum_{j=1}^{J} P(Cluster_k \cap Class_j) log \frac{P(Cluster_k \cap Class_j)}{P(Cluster_k)P(Class_j)}$$

$$= \sum_{k=1}^{K}\sum_{j=1}^{J} \frac{|Cluster_k \cap Class_j|}{N} log \frac{|Cluster_k \cap Class_j|\, N}{|Cluster_k||Class_j|}$$

$$H(Cluster) = -\sum_{k=1}^{K} P(Cluster_k) log P(Cluster_k) = \sum_{k=1}^{K} \frac{|Cluster_k|}{N} log \frac{|Cluster_k|}{N}$$

بررسی معیارهای ارزیابی خوشه‌بندی

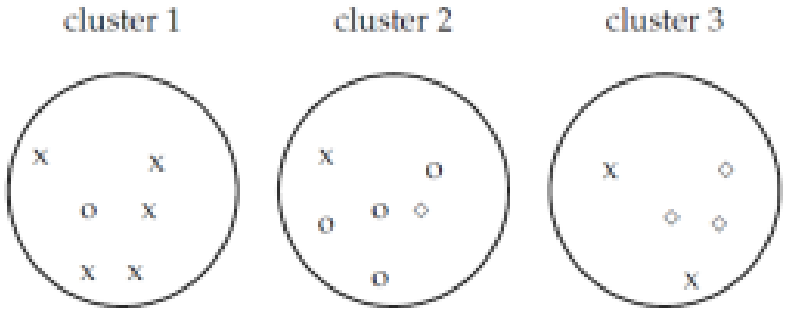
- خالص بودن خوشه ها
- اطلاعات متقابل نرمال شده (Normalized Mutual Information)
- مقدار اطلاعات متقابل

$$NMI(Cluster, Class) = \frac{I(Cluster, Class)}{[H(Cluster) + H(Class)]/2}$$

- مقدار NMI بین صفر و یک است
- بیانگر افزایش میزان اطلاعات ما از دسته ها با دیدن خوشه ها
- عددی بین صفر و یک
- صفر= خوشه بندی تصادفی: دانستن خوشه کمکی به اطلاعات ما نمی کند
- یک= هر خوشه دقیقاً بیانگر یک خوشه است

بررسی معیارهای ارزیابی خوشه‌بندی

- خالص بودن خوشه ها
- اطلاعات متقابل نرمال شده (Normalized Mutual Information)
- ۱۷ نمونه داریم که متعلق به ۳ خوشه است



	دسته ×	دسته ◇	دسته ○	P(Cluster)
خوشه ۱	5/17	0/17	1/17	6/17
خوشه ۲	1/17	1/17	4/17	6/17
خوشه ۳	2/17	3/17	0/17	5/17
P(Class)	8/17	4/17	5/17	

بررسی معیارهای ارزیابی خوشه‌بندی

- خالص بودن خوشه ها
- اطلاعات متقابل نرمال شده (Normalized Mutual Information)
- ۱۷ نمونه داریم که متعلق به ۳ خوشه است

$$I(Cluster, Class) = \sum_{k=1}^{K}\sum_{j=1}^{J} P(Cluster_k \cap Class_j) log \frac{P(Cluster_k \cap Class_j)}{P(Cluster_k)P(Class_j)}$$

$$= \left[\frac{5}{17} log \frac{\frac{5}{17}}{\frac{6}{17} * \frac{8}{17}}\right] + \left[\frac{0}{17} log \frac{\frac{0}{17}}{\frac{6}{17} * \frac{4}{17}}\right] + \cdots + \left[\frac{3}{17} log \frac{\frac{3}{17}}{\frac{4}{17} * \frac{5}{17}}\right] + \left[\frac{0}{17} log \frac{\frac{0}{17}}{\frac{5}{17} * \frac{5}{17}}\right] = 0.565$$

$$H(Cluster) = -\sum_{k=1}^{K} P(Cluster_k) log P(Cluster_k) = -\left(\left[\frac{6}{17} log \frac{6}{17}\right] + \left[\frac{6}{17} log \frac{6}{17}\right] + \left[\frac{5}{17} log \frac{5}{17}\right]\right) = 1.58$$

$$H(Class) = -\sum_{j=1}^{J} P(Class_j) log P(Class_j) = -\left(\left[\frac{8}{17} log \frac{8}{17}\right] + \left[\frac{4}{17} log \frac{4}{17}\right] + \left[\frac{5}{17} log \frac{5}{17}\right]\right) = 1.52$$

$$NMI(Cluster, Class) = \frac{I(Cluster, Class)}{[H(Cluster) + H(Class)]/2} = \frac{0.565}{[1.58 + 1.52]/2} = 0.365$$

بررسی معیارهای ارزیابی خوشه‌بندی

- **خالص بودن خوشه ها**
- **آنتروپی**
  - **محاسبه آنتروپی هر خوشه**
  - **میانگین گیری وزندار روی آنتروپی همه خوشه ها**

احتمال تعلق یک عضو خوشه k به دسته j

$$H(Cluster_k) = -\sum_{j=1}^{J} p_{jk} log p_{jk}$$

$$H = \sum_{k=1}^{K} \frac{|Cluster_k|}{N} H(Cluster_k)$$

بررسی معیارهای ارزیابی خوشه‌بندی

- خالص بودن خوشه ها
- معیار Rand Index و F-Measure
- TP: نمونه به خوشه خودش انتساب داده شده است
- TN: دو نمونه غیر مشابه به دو خوشه مختف انتساب داده شده است
- FP: نمونه به خوشه دیگری انتساب داده شده است
- FN: دو نمونه مشابه به دو خوشه مختلف انتساب داده شده است
- محاسبه دقت انتساب

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

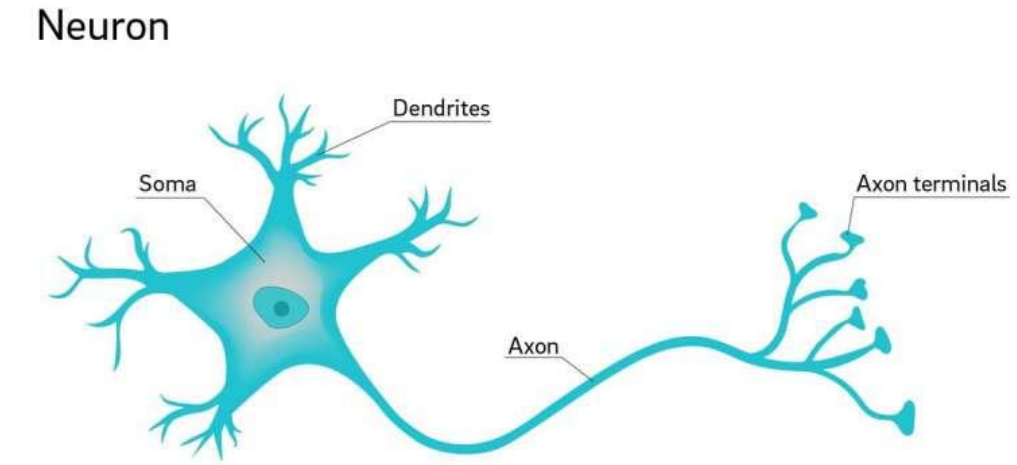
- محاسبه F-Measure

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

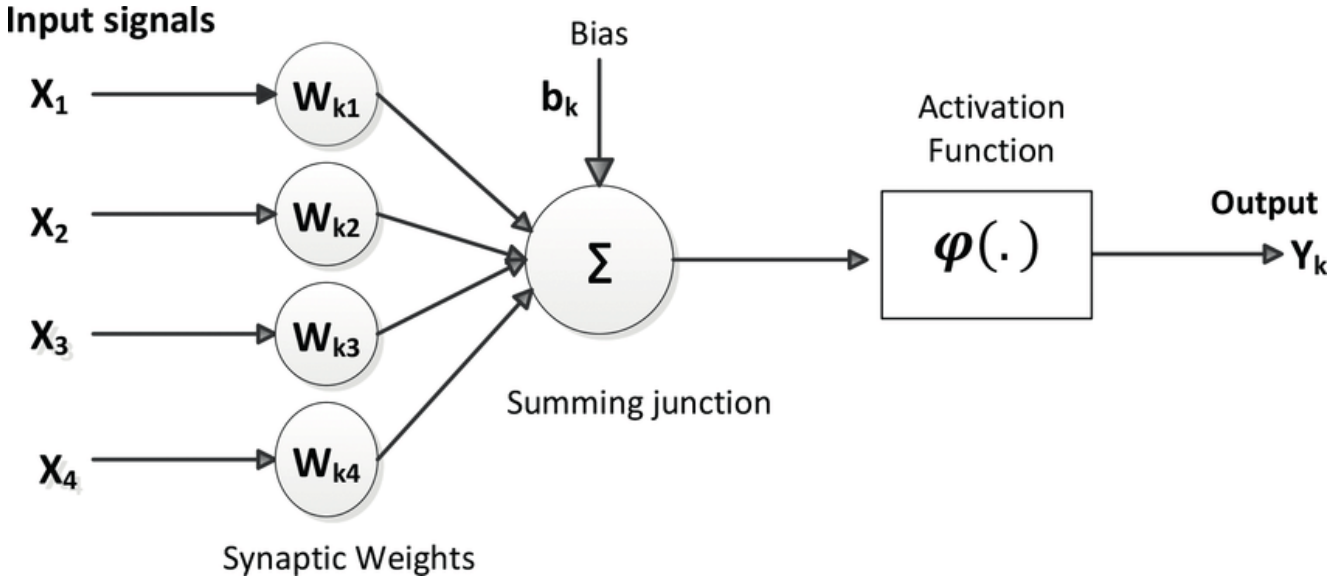
شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی

منبع الهام: نـورون بیولوژیک



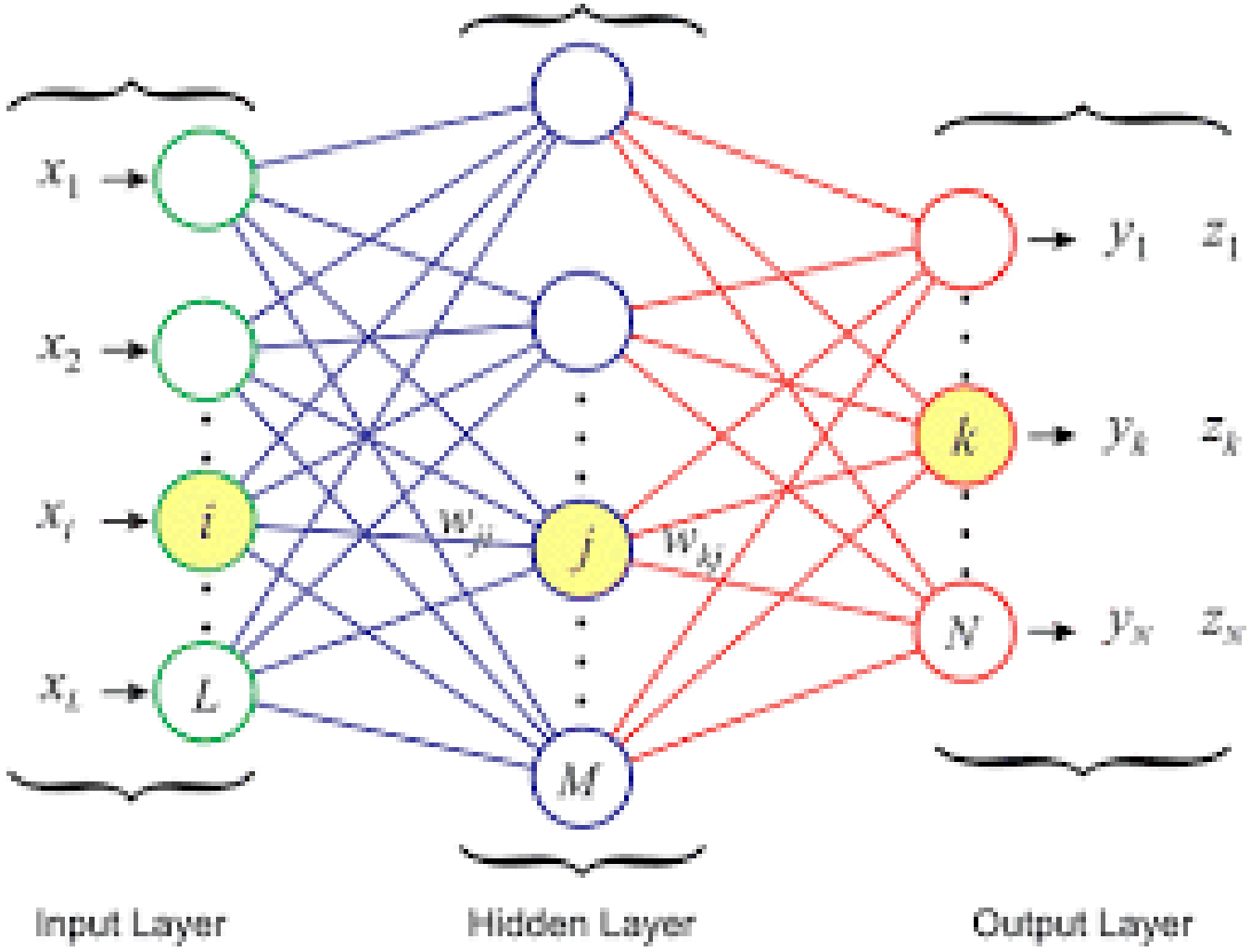
مـــدل ریاضـــی نورون مک‌کلاچ-پیتس



شبکه‌های عصبی مصنوعی

شـــــــــــبکه پرســـــــپترون چندلایه

Multi-Layer Perceptron



شبکه‌های عصبی مصنوعی

آموزش پس‌انتشار خطا error back propagation

- مبنای روش، بهینه‌سازی با گرادیان کاهشی است، برای کمینه کردن خطای خروجی شبکه عصبی. در سه گام:

۱- انتشار به جلو feed forward

$$o_j = \varphi(\mathrm{net}_j) = \varphi\left(\sum_{k=1}^{n} w_{kj} o_k\right)$$

۲- محاسبه گرادیان خطا compute gradient

$$E = \frac{1}{2}\sum_{i=1}^{p} \|\mathbf{o}_i - \mathbf{t}_i\|^2$$

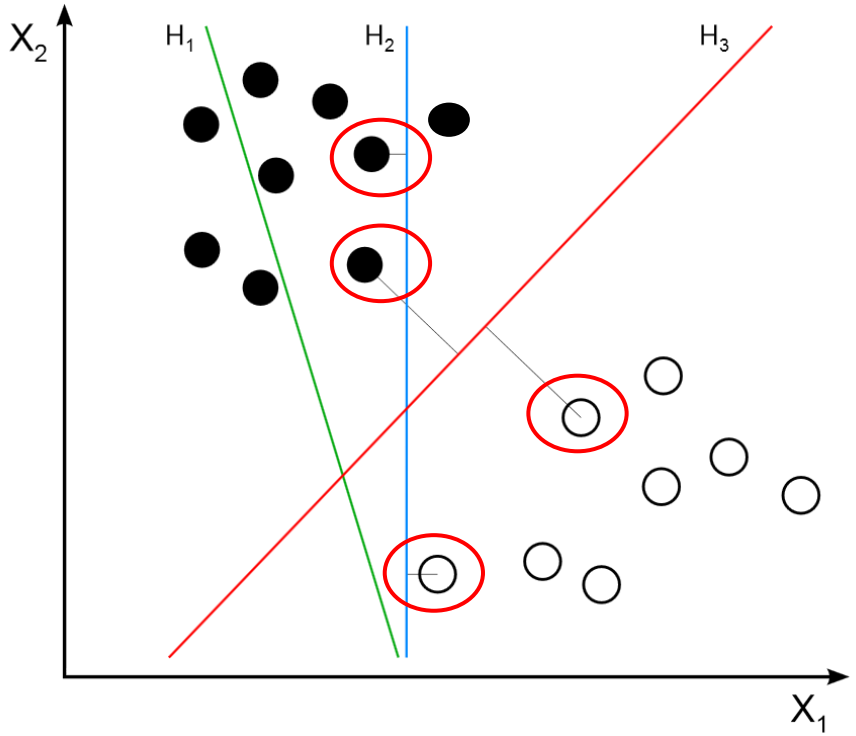
$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_\ell}\right).$$

۳- به‌روزرسانی وزن‌ها update weights

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \quad \text{for } i = 1, \ldots, \ell,$$

Support Vector Machine(SVM)

- معرفی شده در سال ۱۹۹۲ توسط Cortes و Vapnik
- بر خلاف MLP، RBF و ... بجای اینکه خطای کلاسه کردن را کمینه کند، ریسک عملیاتی را بهینه می کند.



یک فضای دو بعدی(خط)

$$w_1 x_1 + w_2 x_2 + b = 0$$

یک فضای سه بعدی(صفحه)

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$$

فوق صفحه یا Hyperplane

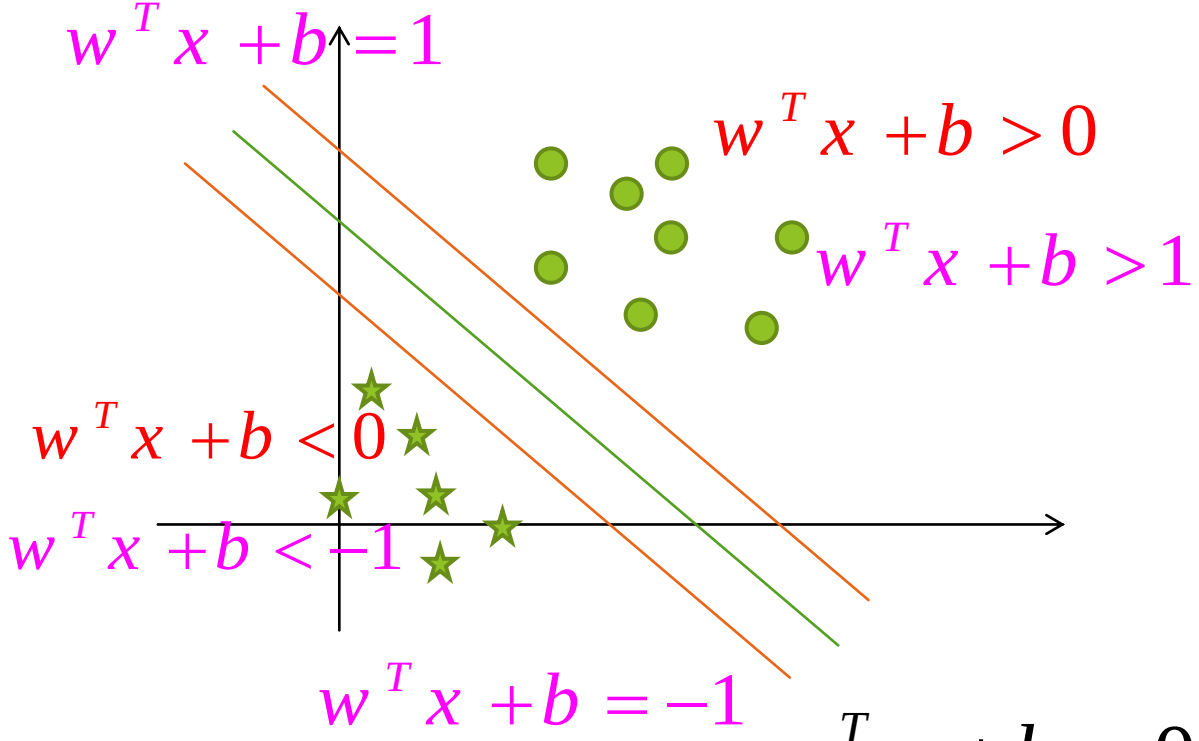
$$\sum_i w_i x_i + b = 0$$

بیان به شکل برداری:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

معادله کلی یک تفکیک کننده

$$w^T x + b = 0$$



تقسیم فضا به دو قسمت

فرض برای داده ها:

$\{x_i, y_i\}$ $\quad i = 1, 2, ..., n$

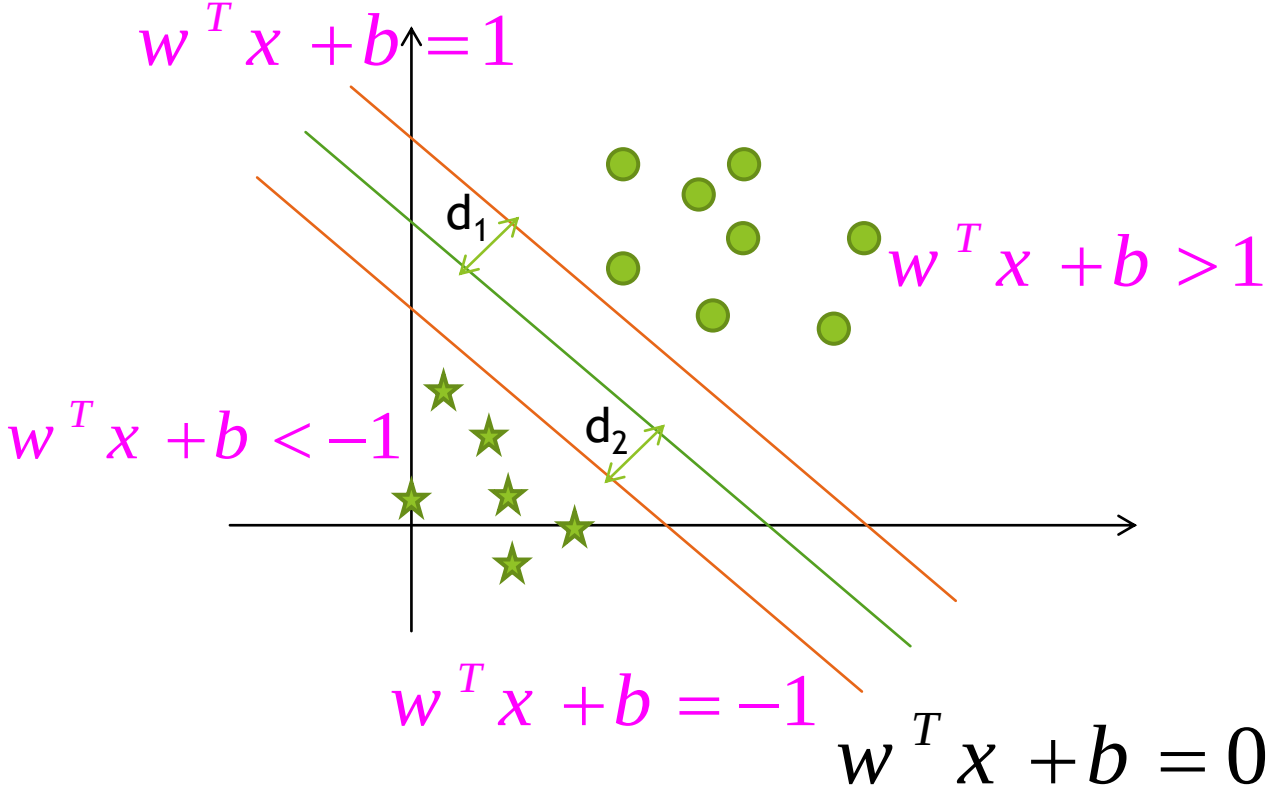
$x_i \in R^d$ یکسری بردار در فضای d بعدی

$y_i \in \{1, -1\}$ Label هستند(نشان دهنده کلاس)

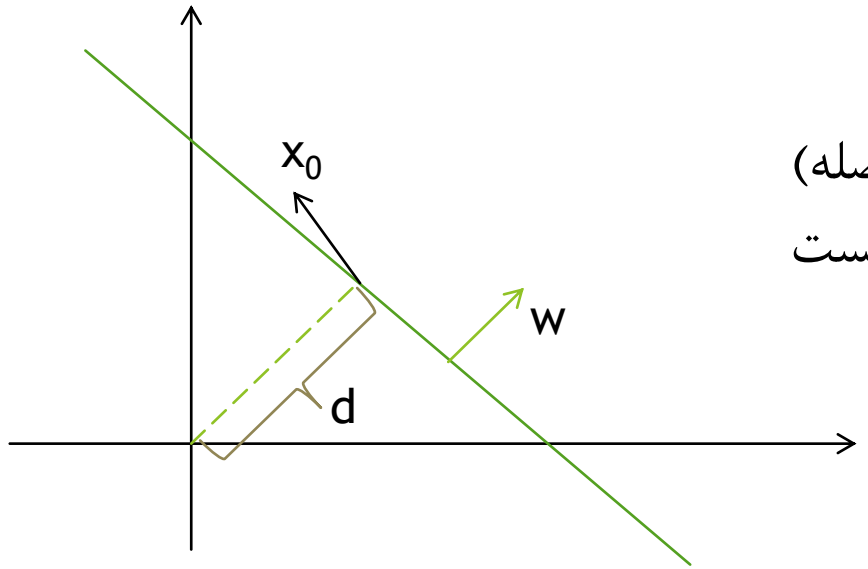
if y_i=1 $\Longrightarrow$ $w^T x_i + b > 1$

if y_i=-1 $\Longrightarrow$ $w^T x_i + b < -1$

$$y = sign(\mathrm{w}^T \mathrm{x} + \mathrm{b})$$



- ✓ تفکیک کننده ای بهتر است که d_1+d_2 آن بیشتر شود.
- ✓ هیچکدام از این کلاس ها از حق خودشان نمی گذرند، پس خط سبز دقیقا بین دو خط آبی قرار می گیرد.
- ✓ بدست آوردن یکی از dها کافیست تا طول حاشیه بدست آید.



- ما می خواهیم d را بدست آوریم(فاصله)
- برداری که عمود بر تفکیک کننده هست همان بردار w است(بردار نرمال در هندسه تحلیلی)

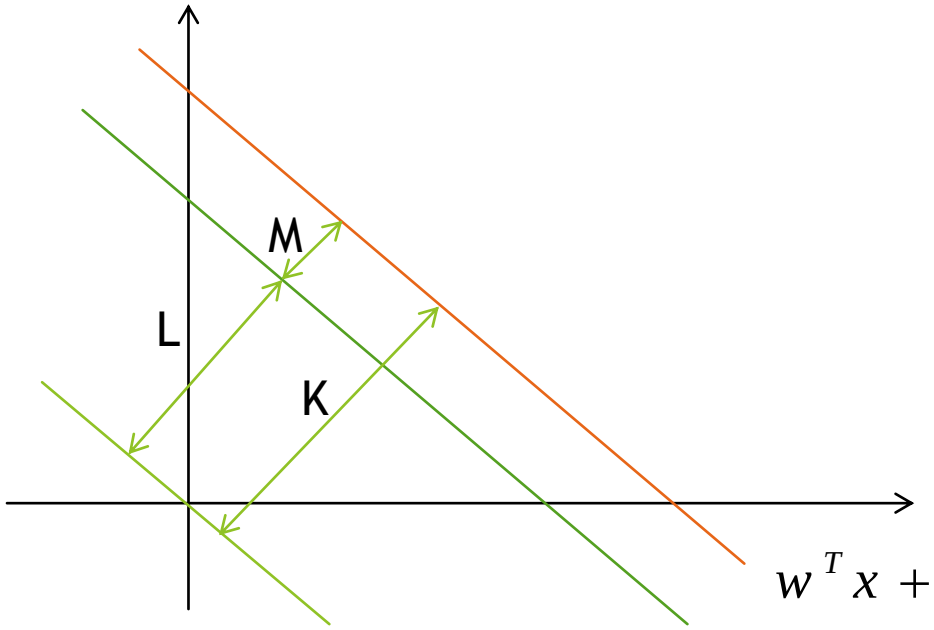
$$w^T x + b = 0$$

$$\begin{cases} x_0 = t.w \\ w^T x_0 + b = 0 \end{cases} \Rightarrow tw^T w + b = 0$$

$$t = \frac{-b}{w^T w} = \frac{-b}{\|w\|^2}$$

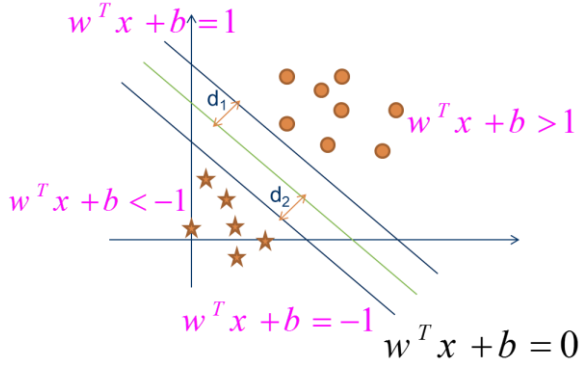
$$\|x_0\| = t\|w\| = \frac{-b}{\|w\|^2}\|w\| = \frac{-b}{\|w\|}$$

فاصله از مبدا: $\frac{|b|}{\|w\|}$



$w^T x + b = 1$

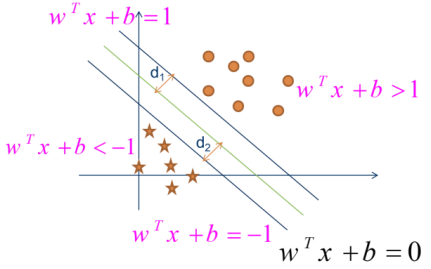
$w^T x + b = 0$



$$M = |K - L| \qquad M = \left| \frac{|b-1|}{\|w\|} - \frac{|b|}{\|w\|} \right| = \frac{1}{\|w\|}$$

Max $$d_1 + d_2 = \frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$$

Min $$\frac{1}{2}\|w\|^2 = \frac{1}{2} w^T w$$



تابع هدف نهایی:

Min $$\frac{1}{2} w^T w$$

S.T

if $y_i=1$ $$\Longrightarrow \quad w^T x_i + b \geq 1$$

if $y_i=-1$ $$\Longrightarrow \quad w^T x_i + b \leq -1$$

$$y_i \left(w^T x_i + b \right) \geq 1$$

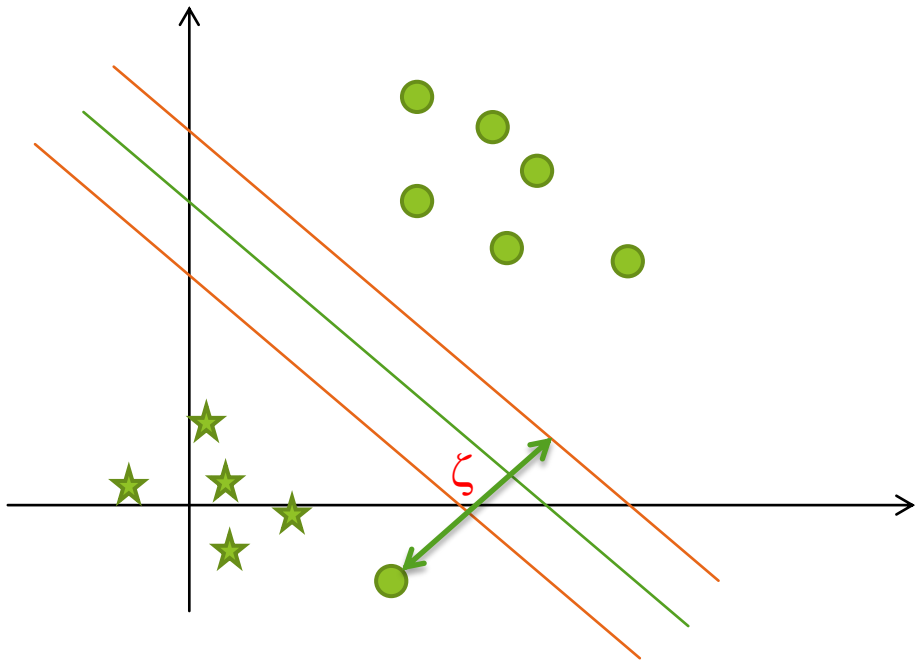
$$\text{Min} \quad \frac{1}{2} w^T w$$

S.T

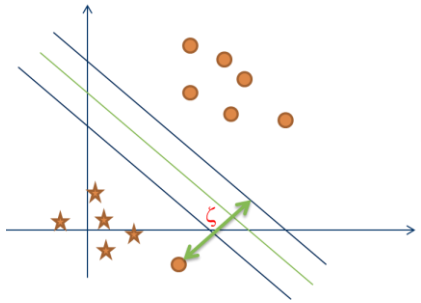
$$y_i \left(w^T x_i + b \right) - 1 \geq 0 \qquad \forall i$$

- از تکنیک لاگرانژ و مساله دوال (Dual Problem) براي حل اين مساله استفاده مي‌شود.

- ✓ آنچه تا کنون داشتیم SVM بـا حاشـیه سـخت بـود(SVM hard margin)، یعنی هیچگونه تخطی را قبول نمی کرد.
- ✓ نوع دیگری از بردار پشتیبان ماشین، SVM با حاشیه نرم اسـت (SVM soft margin)، که تخطی را قبول و برای آن جریمه در نظر می گیرد.



$\zeta \geq 0$ میزان تخطی



if $y_i=1$ $\Rightarrow$ $w^T x_i + b \geq 1$

$\Downarrow$

$w^T x_i + b + \zeta \geq 1$ $\Rightarrow$ $w^T x_i + b \geq 1 - \zeta$

if $y_i=-1$ $\Rightarrow$ $w^T x_i + b \leq -1$

$\Downarrow$

$w^T x_i + b - \zeta \leq -1$ $\Rightarrow$ $w^T x_i + b \leq -1 + \zeta$

$$y_i \left(w^T x_i + b \right) \geq 1 - \zeta$$ شکل واحد

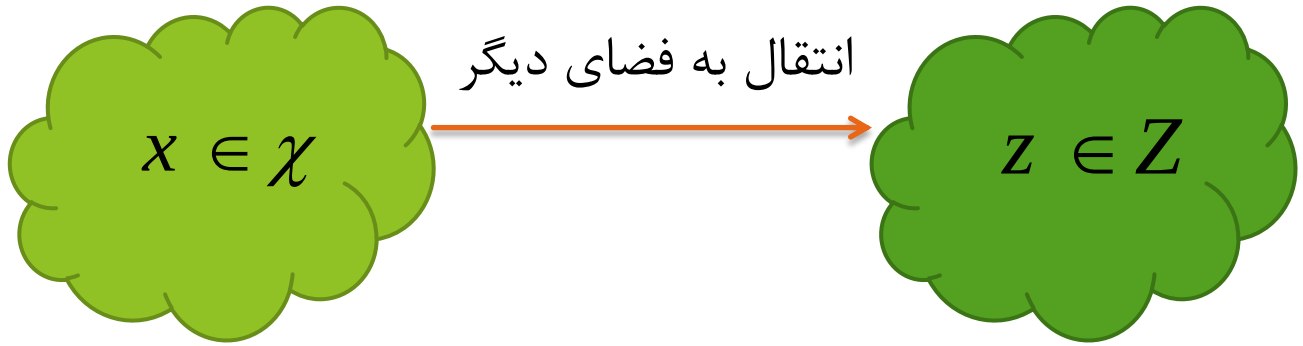
به ازای داشتن ζ باید جریمه پداخت کنن.

$$\text{Min} \quad \frac{1}{2}w^T w + C\sum_i \zeta_i$$

$$\text{S.T} \quad y_i\left(w^T x_i + b\right) \geq 1 - \zeta \qquad \zeta_i \geq 0$$

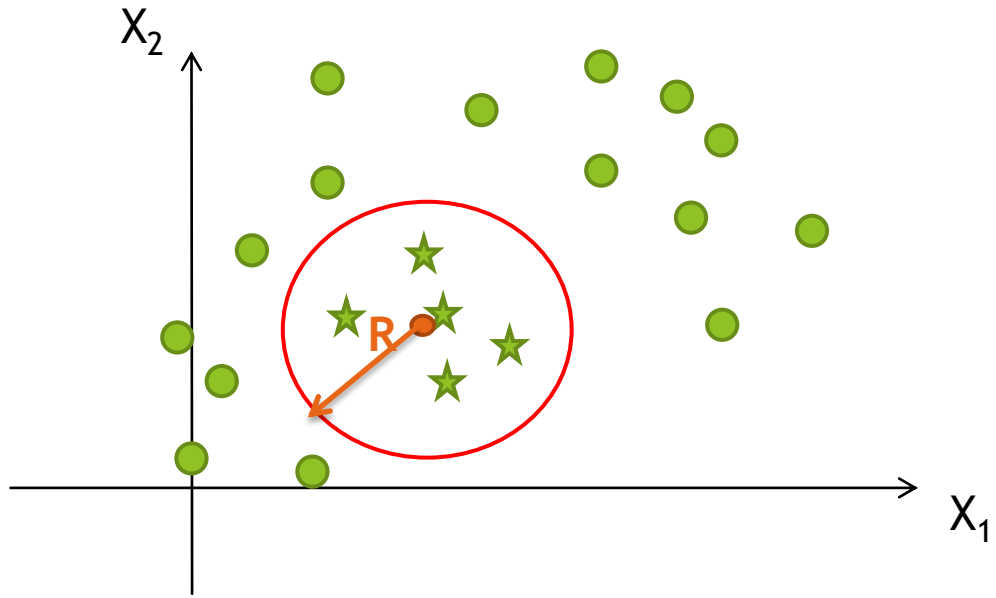
آیا همیشه می‌شود با توابع خطی تفکیک خوبی ایجاد کرد

Kernel Trick



$$Q : X \rightarrow Z \qquad z = Q(\mathrm{x})$$

$$w^T x + b = 0 \longrightarrow w^T Q(\mathrm{x}) + b = 0$$



$$z = \sqrt{(x_1 - x_{01})^2 - (x_2 - x_{02})^2}$$

$z \leq R$ داخل دایره

$z > R$ خارج دایره

$z - R = 0$ تفکیک کننده