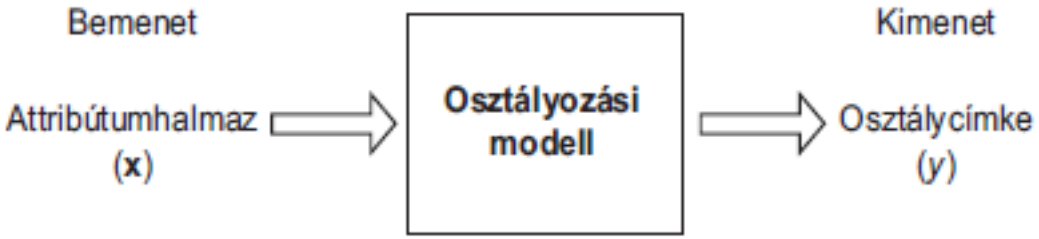
Osztályozás

Osztályozás

- Az **osztályozás**, amely objektumoknak több előre meghatározott kategóriák (osztályok) egyikéhez történő hozzárendelésének a feladata, egy olyan mindent átható probléma, amelyet számos különféle alkalmazás kísér. Ezek a példák magukba foglalják:
- kéretlen elektronikus levelek észlelését az üzenetek fejléce és tartalma alapján,
- sejtek rosszindulatúként vagy jóindulatúként való kategorizálását MRI eredmények alapján,
- galaxisok osztályozását az alakjuk alapján.



Osztályozás

- Egy osztályozási feladatnál **rekordok** egy gyűjteménye alkotja a bemeneti adatokat. Mindegyik rekord (példány, eset), egy (x,y) párral jellemezhető, ahol x az **attribútumok** halmaza és y egy speciális attribútum, amelyet **osztálycímkének** választottunk ki (kategória, célattribútum).
- Minta adatállomány: a gerincesek alábbi kategóriákba való osztályozása: emlős, madár, hal, hüllő, vagy kétéltű. Az attribútumhalmaz a gerincesek olyan tulajdonságait tartalmazza, mint a testhőmérséklet, a bőr függelékei, a szaporodás módja, a repülni tudás és a vízben élés képessége.

Osztályozás

Név	Testhő mérséklet	Bőr függelékei	Eleven szülő	Vízben él	Tud repülni	Van lába	Téli álmot alszik	Osztály-címke
ember	Meleg-vérű	szőr	igen	nem	nem	igen	nem	Emlősök
Óriás-kígyó	Hideg-vérű	pikkely	nem	nem	nem	nem	igen	Hüllők
lazac	Hideg-vérű	pikkely	nem	igen	nem	nem	nem	Halak

Feladat (előrejelző modellezés):

Vipera-gyík	Hideg-vérű	pikkely	nem	nem	nem	igen	igen	?

.

Osztályozás

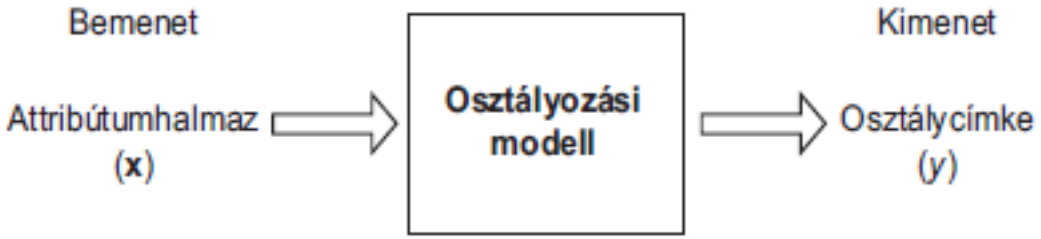
- A bemutatott attribútumok többsége diszkrét, az attribútumhalmaz folytonos jellemzőket is tartalmazhat.
- Az osztálycímkének diszkrét attribútumnak kell lennie. Ez a fő jellemző, ami az osztályozást megkülönbözteti a regressziótól, attól a prediktív modellezési feladattól, amelyben y folytonos attribútum.
- Az **osztályozás** egy olyan f **célfüggvény (target function)** megtanulásának a feladata, amely attribútumértékek minden egyes x halmazához előre definiált osztálycímkék valamelyikét (y) rendeli hozzá.

Osztályozási modellek

- A célfüggvény informálisan **osztályozási modellként (classification model)** is ismert. Az osztályozási modell a következő célokra használható.
- **Leíró modellezés (descriptive modeling)**
- Az osztályozási modell magyarázó eszközként szolgálhat különböző osztályok objektumainak a megkülönböztetésénél. Hasznos lenne például egy olyan leíró modell, amely összegzi és elmagyarázza a táblázat adatait,azaz milyen jellemzők határozzák meg egy gerincesnél, hogy emlős, hüllő, madár, hal, vagy kétéltű.

Osztályozási modellek

- **Előrejelző modellezés (predictive modeling)**
- Egy osztályozási modell arra is használható, hogy megjósoljuk új rekordok osztálycímkéit. Az osztályozási modellt egy olyan fekete dobozként lehet kezelni, amely automatikusan meghatároz egy osztálycímkét, amikor adott egy új rekord attribútumértékeinek halmaza. (pl. a viperagyík esetén)



Osztályozási módszerek

- Az osztályozási módszerek leginkább bináris vagy névleges kategóriákkal rendelkező adatállományok előrejelzésére vagy leírására alkalmasak. Kevésbé hatékonyak rendezett kategóriák esetén (például egy személy osztályozása magas, közepes, vagy alacsony jövedelmű csoport tagjaként), mivel ezek nem veszik figyelembe a kategóriák közötti implicit sorrendet.
- Szintén figyelmen kívül hagyják a kapcsolatok olyan más formáit, mint például a kategóriák közötti alosztály-szuperosztály kapcsolatok (például az ember és az emberszabású majmok főemlősök, ami viszont egy alosztálya az emlősöknek).

Az osztályozási probléma megoldása

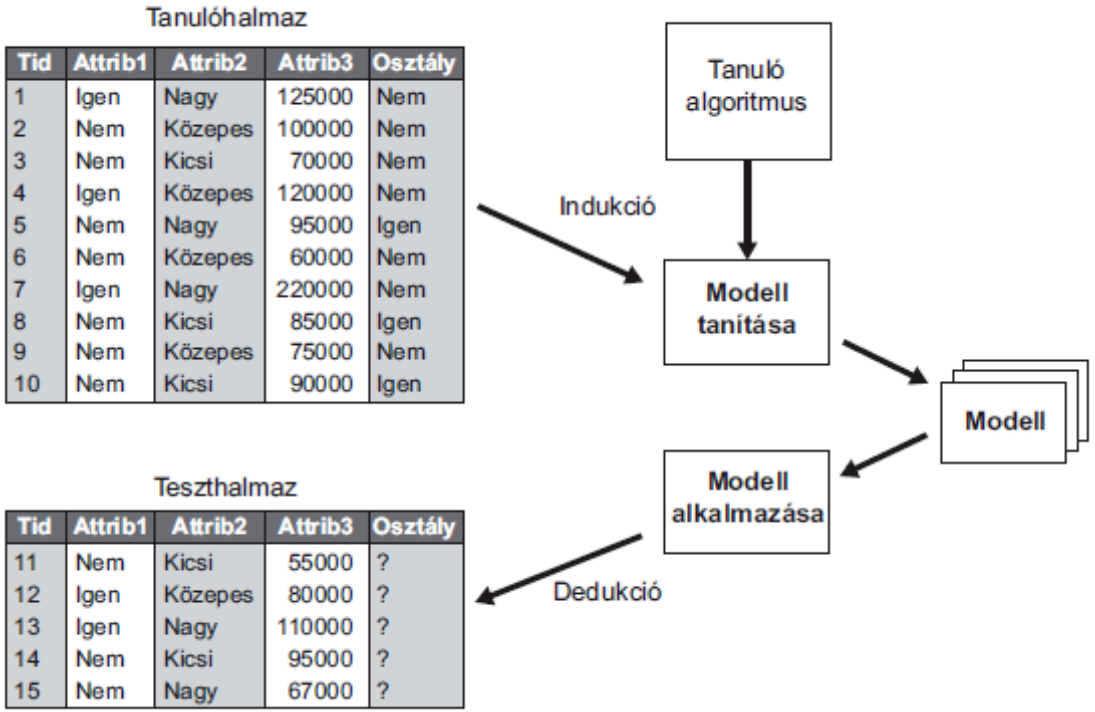
- Egy osztályozási módszer (osztályozó) egy szisztematikus megközelítés osztályozási modellek építésére egy bemeneti adatállományból.
- döntési fák, szabály alapú osztályozók, neurális hálózatok, tartóvektor-gépek és naiv Bayes osztályozók
- Minden módszer **egy tanuló algoritmust (learning algorithm)** alkalmaz annak a modellnek az azonosítására, amely a legjobban illeszkedik a bemenő adatok attribútumai és osztálycímkéje közötti kapcsolatra.

Az osztályozási probléma megoldása

- A tanuló algoritmus által generált modellnek egyszerre kell jól illeszkednie a bemenő adatokra és helyesen megjósolnia korábban soha nem látott rekordok osztálycímkéit.
- Ezért a tanuló algoritmus fő célja jó általánosítási képességgel bíró modellek építése, azaz, hogy a modell pontosan jósolja meg korábban ismeretlen rekordok osztálycímkéit.
- **tanulóhalmaz (training set):** olyan rekordokból áll, amelyeknek ismert az osztálycímkéje.

Az osztályozási probléma megoldása

- A tanulóhalmazt egy osztályozási modell kialakításához használjuk, amelyet ezt követően **a teszthalmazra (test set)** alkalmazunk, amely ismeretlen osztálycímkéjű rekordokból áll.



Osztályozási modellek teljesítménye

- Egy osztályozási modell teljesítményének kiértékelése a modell által helyesen és helytelenül előrejelzett tesztrekordok számán alapszik. Ezeket a számokat egy ún. **tévesztési mátrixba** foglaljuk (confusion matrix).
- egy bináris osztályozási feladat tévesztési mátrixa f_{ij} : azoknak az i osztálybeli rekordoknak a száma, amelyeket a j osztályba jelzünk előre.
- A tévesztési mátrix elemei alapján a modell által tett összes helyes előrejelzés száma (f_{11}+ f_{00}), valamint az összes hibás előrejelzés száma (f_{10} + f_{01}) .

Osztályozási modellek teljesítménye

- Bár a tévesztési mátrix biztosítja annak meghatározásá-hoz szükséges információkat, hogy milyen jól teljesít egy osztályozási modell, ezen információk egyetlen számmá való összegzése még kényelmesebbé tenné a különböző modellek teljesítményének az összehasonlítását.
- Ez olyan **teljesítménymérték (performance metric)** révén tehető meg, mint a **pontosság (accuracy)**:

pontosság = helyes előrejelzések száma/
összes előrejelzés száma = $(f_{11}+ f_{00}) / (f_{11}+ f_{10} + f_{01} +f_{00})$

Osztályozási modellek teljesítménye

- Ezzel egyenértékűen, egy modell teljesítményét kifejezhetjük a **hibaarányával (error rate)** megadva:
  Hibaarány = hibás előrejelzések száma/
  összes előrejelzés száma = $(f_{01}+ f_{10}) / (f_{11}+ f_{10} + f_{01} +f_{00})$
- A legtöbb osztályozási algoritmus olyan modelleket keres, amelyek a legnagyobb pontosságot, vagy azzal egyenértékűen, a legalacsonyabb hibaarányt érik el, amikor a teszthalmazon alkalmazzuk őket.

Modell túlillesztés

- Egy osztályozási modellben elkövetett hibák általában két csoportba sorolhatóak: **tanítási hibák (visszahelyettesítési hiba) és általánosítási hibák**. A tanítási hiba a tanulórekordokon elkövetett helytelen osztályozási hibák száma, míg az általánosítási hiba a modell várható hibája korábban nem látott rekordokon.
- Egy jó modellnek kis tanítási hibával, valamint kis általánosítási hibával kell rendelkeznie. Ez azért fontos, mert egy olyan modell, amely túl jól illeszkedik a tanulóadatokra, rosszabb általánosítási hibával rendelkezhet, mint egy nagyobb tanítási hibájú modell -> **modell túlillesztés**

Osztályozó teljesítményének kiértékelése

- Egy modell általánosítási hibájának becslése a tanítás során a modell kiválasztásban segíti a tanuló algoritmust: a megfelelő bonyolultságú modellnek a megtalálásában, amely nem fogékony a túlillesztésre.
- Miután felépült a modell, a teszthalmazon lehet alkalmazni korábban ismeretlen rekordok osztálycímkéinek az előrejelzésére.
- A teszthalmazon számolt pontosságot (hibaarányt) használhatjuk különböző osztályozók ugyanazon a területen vett relatív teljesítményének az összehasonlítására. (ismernünk kell a tesztrekordok osztálycímkéit)

Visszatartó módszer

- A címkézett esetekből álló eredeti adatokat két diszjunkt halmazra bontjuk, melyeket tanító- és teszthalmaznak nevezünk. Az osztályozási modellt ezután a tanulóhalmazon építjük fel, és teljesítményét a teszthalmazon értékeljük ki.
- A tanulásra és a tesztelésre fenntartott adatok arányát az elemzők általában saját belátásuk szerint állapítják meg (például fele-fele, vagy kétharmad a tanulásra és egyharmad a tesztelésre).
- Az osztályozó pontosságát a felépített modell teszthalmazon mért pontossága alapján lehet becsülni.

Visszatartó módszer ismert korlátjai

- kevesebb címkézett eset áll rendelkezésére a tanuláshoz, mert néhány rekordot a teszteléshez tartunk vissza -> a felépített modell nem lehet olyan jó, mintha az összes címkézett esetet a tanulásra használnánk.
- a modell nagyban függhet a tanuló- és teszthalmazok összetételétől. Minél kisebb a tanulóhalmaz mérete, annál nagyobb a modell szórása. Ha a tanulóhalmaz túl nagy, akkor a kisebb teszthalmazból számított becsült pontosság kevésbé megbízható (konfidencia interv.)
- a tanuló- és teszthalmazok nem függetlenek egymástól. Mindkettő az eredeti adatok részhalmazai, egy olyan osztály, amely felülreprezentált az egyik részhalmazban, alulreprezentált lesz a másikban.

Keresztellenőrzés (cross-validation)

- Minden rekordot ugyanannyiszor használunk tanításra és pontosan egyszer tesztelésre.
- A **k-szoros keresztellenőrzés** módszere az adatokat k egyenlő méretű partícióra osztja fel.
- Mindegyik futás során az egyik partíciót választjuk ki tesztelésre, míg a megmaradókat a tanításra használjuk.
- Ezt a folyamatot k-szor ismételjük meg, így minden egyes partíciót pontosan egyszer használunk tesztelésre.
- teljes hiba: az összes k futás hibájának összege

Keresztellenőrzés (cross-validation)

- A k-szoros keresztellenőrzés módszerének egy speciális esete: k = N , az adatállomány mérete.
- Ebben a hagyj-ki-egyet (leave-one-out) megközelítésben minden teszthalmaz egyetlen rekordot tartalmaz.
- A megközelítés előnye: annyi adatot használ a tanításnál, amennyi csak lehetséges. Ezenkívül a teszthalmazok egymást kölcsönösen kizáróak és ténylegesen lefedik a teljes adatállományt.
- A megközelítés hátránya: számításköltséges N-szer megismételni az eljárást. Mivel mindegyik teszthalmaz csak egy rekordot tartalmaz, a becsült teljesítménymérték varianciája általában nagy.

Osztályozási módszerek

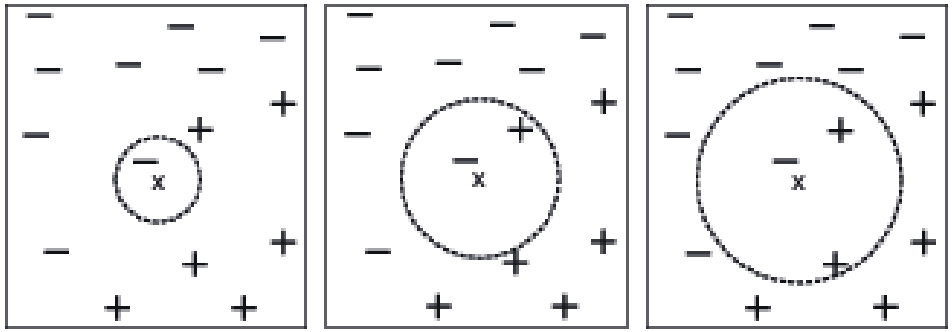
Legközelebbi szomszéd osztályozók

- Az **osztályozás** egy kétlépcsős eljárást foglal magába:
- induktív lépés: az adatokból egy osztályozási modellt alkotunk
- deduktív lépés: a modell tesztesetekre való alkalmazásához
- **Rote osztályozó**: az összes tanulóadatot memorizálja, és csak akkor osztályoz, ha a tesztpéldány attribútumai pontosan illeszkednek egy tanulóesetre.
- A módszer egy nyilvánvaló hátulütője az, hogy bizonyos tesztesetek nem osztályozhatók, mivel nem egyeznek meg a tanulóesetek egyikével sem.

Legközelebbi szomszéd osztályozók

- A módszer rugalmasabbá tétele: a teszteset attribútumaihoz viszonylag hasonló valamennyi tanulóeset megkeresése. Ezek az esetek a **legközelebbi szomszédok (nearest neighbors)**, felhasználhatók a teszteset osztálycímkéjének meghatározásához.
- A legközelebbi szomszédok használatának indoklását legjobban a következő mondás szemlélteti: ''Ha valami úgy totyog, mint egy kacsa, úgy hápog, mint egy kacsa és úgy néz ki, mint egy kacsa, akkor az valószínűleg egy kacsa. ''
- A legközelebbi szomszéd osztályozó minden egyes esetet egy adatpontként reprezentál egy d-dimenziós térben, ahol d az attribútumok száma. Szomszédsági mértékek valamelyikével kiszámítjuk ezek közelségét a tanulóhalmaz összes többi adatpontjához.

Legközelebbi szomszéd osztályozók



(a) 1-legközelebbi szomszéd (b) 2-legközelebbi szomszéd (c) 3-legközelebbi szomszéd

- A többségi szavazási sémával az adatpontot a pozitív osztályhoz rendeljük hozzá.
- Holtverseny esetén az adatpont osztályozásához véletlenszerűen választhatjuk valamelyik osztályt.

Legközelebbi szomszéd osztályozók

- A k érték helyes megválasztásának fontossága:
- Ha k túl kicsi, akkor a legközelebbi szomszéd osztályozó a tanulóadatokban jelenlevő zaj miatt hajlamos lehet a túlillesztésre.
- Ha k túl nagy, akkor a legközelebbi szomszéd osztályozó rosszul osztályozhatja a tesztpéldányt, mivel a legközelebbi szomszédok listája a szomszédságtól messzi adatpontokat is tartalmazhat.
- Az algoritmus kiszámítja minden teszteset és tanulóeset távolságát (hasonlóságát) -> számításköltséges nagy-számú tanulóeset esetén -> hatékony indexelési eljárások, amelyek csökkentik az egyes tesztesetek legközelebbi szomszédainak megkereséséhez szükséges számítási mennyiséget.

Legközelebbi szomszéd osztályozók

- **példányalapú tanulás**: konkrét tanulópéldányokat használ predikció végzéséhez anélkül, hogy az adatokból származó absztrakcióra (modellre) lenne szüksége.
- egy szomszédsági mérték szükséges a példányok hasonlóságának vagy távolságának meghatározásához, valamint egy osztályozási függvény, amely más példányokhoz közelsége alapján visszaadja egy tesztpéldány előrejelzett osztályát.
- Nem igényelnek modellépítést, viszont a tesztesetek osztályozása elég költséges lehet, mivel külön-külön kell kiszámolnunk a teszt- és a tanulóesetek közelségét.

Legközelebbi szomszéd osztályozók

- lokális információk alapján végeznek előrejelzést. A döntéshozatal lokálisan történik a osztályozás során, kis k értékek esetén ezek az osztályozók elég érzékenyek a zajra.
- hibás előrejelzéseket adhatnak, hacsak nem végezzük el a megfelelő előfeldolgozási lépéseket a szomszédsági mértéken és az adatokon.
- magasság (méter) és testsúly (font) alapján osztályozás. A magasság attribútum kis változékonyságú (1,5 és 1,85 között), míg a testsúly attribútum 90 és 250 font között változhat –> az attribútumok skáláját figyelembe kell venni

Bevezetés az adatbányászatba (Pang-Ning Tan, Michael Steinbach, Vipin Kumar) című tananyaga alapján készült (részben)