



**SZÉCHENYI**  
ISTVÁN  
**EGYETEM**



# **SZAKDOLGOZAT**

## **Termékajánlás valós példán demonstrálva**

**Göntér Anna**

**Mérnök Informatikus BSc szak**

**2019**

## Nyilatkozat

Alulírott, Göntér Anna (KHRV0P) Mérnök Informatikus, BSc szakos hallgató kijelentem, hogy a Termékajánlás valós példán demonstrálva című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, 2019.04.23.

---

hallgató

# Kivonat

## Termékajánlás valós példán demonstrálva

A termékajánló rendszerek egyre népszerűbbek lesznek a mai internetes világban. Amikor világhálón megtekintünk egy könyvet, filmet, zenét vagy akár szállást keresünk a termékajánló rendszerek folyamatosan dolgoznak a háttérben, hogy találjanak számunkra egy még jobb könyvet vagy filmet.

A termékajánló rendszerek háttérében komoly gépi tanulási algoritmusok állnak, amik nagyon sokrétűek lehetnek. Különböző algoritmusok, nagyon különböző eredményeket tudnak generálni. Az algoritmus hatékonysága függhet attól is, hogy milyen adatokat kap a rendszer a felhasználótól, vagy, hogy a felhasználó, mikor regisztrált, vannak-e előzmények. A dolgozatom első felében ezeket a lehetőségeket fogom körül járni és bemutatni.

A dolgozatom második felében pedig ki fogom próbálni ezeket az algoritmusokat egy adott adatkészleten, amit a Kaggle.com oldalon találtam. Az oldal prediktív és elemzési versenyek és kihívások számára lett kitalálva, ahova különböző kisebb-nagyobb vállalatok tölthetik fel az adatkészletüket és indíthatnak versenyeket.

Munkám célja egy a Kaggle.com oldalon talált adatkészleten kipróbálni a lehető legtöbb algoritmust és megtalálni azt, amelyik a legjobb ajánlást tudja készíteni az adott adatkészletre. A kiválasztott adatkészlet az Expedia szálláskereső oldal által felrakott verseny adatai. A verseny már lezárult, de az eredményeket továbbra is lehet látni, illetve az oldal továbbra is kiértékeli az általam feltöltött eredményeket.

# **Abstract**

## **Recommendation Systems Demonstrating on Real Example**

Nowadays the Recommender Systems is becoming increasingly popular, because of the world of internet, and social network. Whenever we search for books, movies or accommodations on the internet. The recommender systems are always working in the background to find a better book or a better movie for us.

There are serious machine learning algorithms in the background of the recommendation systems. Different algorithm can produce very different outcoming results. The efficiency of the algorithms depends on several parameters, like the data from the users, the date of the registration or the history of the user.

In the first part of my thesis I will present these algorithms and the background of the recommender system.

The second part of the thesis I will try these algorithms on a dataset what I found on the Kaggle.com website. On this website there are a lot of predictive and analytical competitions and challenges. Various smaller or larger companies can upload their datasets and launch competitions, which will help them improve their algorithms.

The aim of my work is to try out as many algorithms as possible on the dataset what I found on Kaggle.com and find the best recommendation algorithm for that dataset. The selected dataset is uploaded by Expedia accommodation search page. The competition is over, but the results are still seen, and the page can evaluate the results I will upload.

# Tartalomjegyzék

Nyilatkozat .....	2
Kivonat .....	3
Abstract.....	4
Tartalomjegyzék .....	5
Bevezetés .....	1
Termékajánlórendszerek .....	2
Termékajánló rendszerek a gyakorlatban .....	5
Termékajánló módszerek .....	7
Kollaboratív ajánlás.....	7
Szomszéd alapú megközelítés .....	8
Tartalom alapú termékajánló rendszerek .....	9
Előnyök és hátrányok .....	10
Tudás alapú termékajánló rendszerek.....	10
Kényszer alapú ajánlórendszer .....	11
Eset alapú ajánlórendszer .....	11
Társadalom alapú termékajánló rendszerek .....	12
Demográfiai termékajánló rendszerek.....	12
Gépi tanulás alapú módszerek .....	13
Felügyelet nélküli tanulás.....	13
Felügyelt tanulás.....	14
Legközelebbi szomszédok.....	15
Döntési fa.....	16
Véletlen erdők.....	17
Adaptív erősítés .....	17
Támogató vektor gépek .....	18
Mesterséges neurális hálózatok .....	18
Bayes-i osztályozó.....	19
Irodalomjegyzék .....	20

## Bevezetés

Mindenki, aki vásárolt már online találkozhatott az alábbi jelenséggel. Berakott egy terméket (például könyv) a virtuális kosarába és ennek következményeképp az oldal azonnal ajánlásokat ad, hogy miket érdemes még megvennie mellé, vagyis feltűnik a jól ismert „Mit vettek még, akik ezt vették?” mondat. Az oldálnak egy algoritmus segít megtalálni a jó párosítást, az alapján, hogy a felhasználó eddig miket tekintett meg, miket vásárolt, miket vásároltak azok, akiknek hasonló érdeklődési körük van vagy ugyanazt a terméket vásárolták. Ma már nagyon sok nagy cég használja ennek az algoritmusnak az előnyeit és sok esetben jelentősen meg tudja növelni a bevételét. Ezeknek a rendszereknek előnyét, hátrányát és megvalósítását fogom bemutatni az alábbiakban. Az informatikusok először az 1990-es évek végén kezdtek el foglalkozni a gépi tanulással. [1] Majd 2002-ben megjelent a ***Torch: a modular machine learning software library*** című dokumentum, ami még jobb alapot adott neki. [2] De az ajánlórendszerek köztudatba robbanása 2006-ban történt, amikor a Netflix kihirdetett egy versenyt, aminek a főnyereménye 1.000.000\$ volt, célja pedig, minél jobb becslést adni, hogy az adott illető mennyire kedvelne egy adott filmet, ezt gépi tanulással és az ajánlórendszer készítésével törekedtek a lehető legjobb eredményre, ehhez adott volt egy adatbázis, aminek az adatai alapján kellett a becsléseket készíteni. Több ezer csapat jelentkezett a világ minden tájáról, még Magyarországról is, akik 14. helyezést értek el. [3]

## Termékajánlórendszerek

A termékajánló rendszer egy olyan rendszerre utal, ami a felhasználó számára megtalálja a személyre szabott ajánlatokat és csak azokat ajánlja. Azért van erre szükség, mert a digitális vásárlási folyamatnak köszönhetően túl sok választási lehetőségük van és ezeket szűkíteni kell. Régen az emberek rendes áruházban vásároltak, ahol a rendelkezésre álló termékek csak korlátozott számban voltak elérhetőek. Például egy könyvesboltban található könyvek száma csak annyi lehet, amennyi a boltban elfér, viszont az interneten nincsenek ilyen korlátok. Mivel nőtt a rendelkezésre álló termékek száma felmerült a probléma, hogy az emberek nagyon nehezen tudják kiválasztani, azokat a termékeket, amikre szükségük van, szeretnének nézni/olvasni és még tetszik is nekik. Ez a probléma szülte az ajánlórendszer létrejöttét. [4] Például termékajánlás az is amikor bemegyünk egy könyvesboltba és vannak könyvek, amik a borítójukkal 'felénk néznek', de vannak olyanok is amelyeknek csak a gerincét látjuk. Így akarja a bolt ajánlani azokat a könyveket, amiknek a borítóját látjuk. Az ajánló rendszert többféleképpen hozhatjuk létre, például tartalomalapú, vagy együttműködésen alapuló szűréssel.



*A képen egy karikatúra látható a termékajánló rendszerekről. [5]*

A kép nagyon jól szemléltet, hogy miről szólnak ezek a rendszerek. A termékajánlást minden boltban megtalálhatjuk, mivel nem véletlenül vannak a csokik a kassza mellett vagy a tejtermékek közel a kenyérfélékhez, mivel ezzel a bolt üzenni akar, hogy nekünk arra is szükségünk van.

A termékajánló rendszerek nagyon hasznosak tudnak lenni a vállalatok számára, mivel ezáltal jobban tudják reklámozni a termékeiket és több emberhez eljutnak. Ugyanakkor a felhasználó szemszögéből is hasznos lehet, mivel nem kell órákig keresgélnie éppen, amit szeretne, vagy ha szállást foglal a termékajánló rendszer azonnal kiadja neki azt amelyiket nagy valószínűséggel jóra értékelné, és nem kap sok felesleges, irreleváns ajánlatot és

információt.

Az ajánlórendszerek előnyei az eladó szemszögéből:

- Növeli az eladott termékek számát – Ez a fő oka a termékajánló rendszerek létrejöttének, hogy a szolgáltató minél több terméket el tudjon adni. Ami annak köszönhető, hogy a rendszer feldob más árukat is a vevőnek, hogy ez is érdekelheti és lehet, hogy a vevő magától nem keresett volna rá az ajánlott termékre, de mivel a rendszer megmutatta, rájön, hogy szüksége van rá. Például ilyen, amikor a boltokban direkt olyan sorrendben rakják a termékeket, hogy az ember minden soron végig menjen és több dolgot megkívánjon.
- Segít több különböző terméket eladni – A termékajánló rendszer képes olyan ajánlatot adni a felhasználónak, amit a rendszer nélkül nehezen talált volna meg. Például egy filmes ajánlórendszerben lehet, hogy a felhasználó előnyben részesíti az újabb filmeket és nem tud a régiekről, de a rendszer megtalálja neki azok között is ami tetszhet, úgy, hogy a felhasználónak eszébe se jutna ott keresni.
- Növeli a felhasználó elégedettségét és hűségét a termék/oldal iránt – Ha jól működik az ajánlórendszer akkor a felhasználó már csak azért is meglátogatja az oldalt, mert kíváncsi, hogy éppen mit ad fel neki a rendszer, hogy érdekelheti. Ezt egy személyes példával tudom alátámasztani, én az IMDB oldalát nagyon gyakran szoktam nézni, hogy ajánljon nekem hasonló filmeket, amiket ezelőtt néztem, néha még akkor ránézek, amikor éppen nem is akarok filmet nézni.
- Segít megérteni, mit akar a felhasználó – Például, ha a keresési előzmények alapján működik az ajánlórendszer, akkor a rendszer jobban emlékszik az előzményekre, mint a felhasználó, így sokkal könnyebben, jobbat tud ajánlani, mintha a felhasználó kezdett volna el keresgélni millió film között.

Az ajánlórendszerek előnyei a felhasználó/vásárló szemszögéből:

- Megtalálja az összes releváns terméket a felhasználó számára – Az ajánlórendszer az összes lehetséges ajánlatot megtalálja a felhasználónak és generál egy várható értékelést, amit a felhasználó adna rá.
- Csomag ajánlása – Például sok nagyobb városban megtalálható város nevével ellátott kártya, amivel a felhasználóknak kedvezményeket biztosítanak különböző múzeumokba, éttermekbe és sokszor kínálnak ingyenes tömegközlekedést is. Ez azért jó, mert lehet, hogy az ember nem akar elmenni egy múzeumba, de látja, hogy a kártyával megéri meglátogatni, és ezáltal kihagyhatatlan élményben lesz része.



- Böngészheti a neki ajánlott termékeket – Ebben az esetben a felhasználó többi ajánlatot is kap, de nem direktben mindenféle felszólítással, hanem amikor böngészik, akkor a számára legideálisabb termékeket mutatja az oldal először.
- Másokat segít, illetve befolyásol – Az értékeléseket olvasva a felhasználók könnyebben fel tudják ismerni, ha egy termék nagyon jó vagy nagyon rossz minőségű, esetleg tudnak találni összehasonlításokat is a vélemények között. A probléma ezzel, hogy a felhasználó az összes másik felhasználó által írt véleményeket olvashatja, nem csak azokét a felhasználókét, akik hasonlóak hozzájuk. [6]

Az ajánlást több más fontos tényező is befolyásolhatja, amik által javulhat, de akár romolhat is az ajánlórendszer pontossága, ezek a következő tényezők lehetnek:

- Sokszínűség – A felhasználó elégedettebb, ha az ajánlat, amit kap az sokszínűbb. Például, ha tetszett neki egy film egy bizonyos rendezőtől, akkor a rendszer nem csak annak a rendezőnek a műveit ajánlja neki, hanem más rendezőktől, hasonló filmeket is mutat.
- Ajánlás megtartása – Előfordul, hogy hasznos az ajánlást újból felmutatni a felhasználónak, nem pedig újakat mutatni, mert lehet, hogy elkerülte a figyelmét vagy éppen nem olyan hangulatban volt, hogy azt a filmet nézze meg vagy zenét hallgassa meg viszont, ha újra ajánljuk, akkor előfordulhat, hogy pár nap múlva már ahhoz lesz kedve. [7]
- Bizalom – Az ajánló rendszer nem ér semmit, ha a felhasználó nem bíz benne. Ez nagyon logikus dolog, mivel a felhasználó nem tudja, hogy a program honnan tudja róla, hogy mi tetszik neki, ezt úgy lehet kiküszöbölni, hogy a rendszer kiírja azt is, hogy miért ajánlotta az adott könyvet. Például gyakran lehet látni, hogy az van a weboldalon, hogy megtekintetted az x terméket ezért y is tetszeni fog. [8]
- Manipuláció – A termékajánló rendszerek kezdeti időszakában volt jellemző, hogy a felhasználók is részt vehettek az ajánlat elkészítésében, de hamar rájöttek, hogy így sok csalást végbe lehet vinni, azzal, ha valaki téves információkat ír a véleményében, és így akár hátrányba tudja szorítani a konkurenciáját. [9] Például bevett marketing fogás, hogy a nagy cégek felbérelnek több száz embert is akár, hogy írjanak pozitív véleményt egy termékükről így több vásárlóhoz juthatnak, vagy ugyanez fordítva is igaz lehet, hogy negatív véleményt írnak a vetélytárs termékéről.

- Túl egyértelmű ajánlások – Nagyban ronthat a pontosságon, ha az ajánlás túl egyértelmű, mert akkor a felhasználó egyből maximum pontot fog neki adni és így hamisan emeli a pontosságot. Nagyon jó példa erre, hogy valaki bemegy az élelmiszerboltba és a rendszer ajánlja neki, hogy vegyen kenyeret, persze ez nagyon jó ajánlás, mert tényleg fontos, hogy legyen kenyér, viszont ugyanakkor nagyon rossz ajánlás is, mivel a felhasználó pont ezért megy be a boltba, hogy kenyeret vegyen. [10]

#### *A segédmátrix*

A termékajánló rendszerek létrehozásánál nagy segítségünkre lehetnek a segédmátrixok, amik tartalmazzák, hogy különböző felhasználó, hogyan értékelték a különböző termékeket, és ez alapján dolgozik nagyon sok algoritmus, hogy megkeresi a hasonló felhasználókat és összehasonlítja őket.

Mind a tartalom alapú, mind az együttműködésen alapuló rendszer egy segédmátrixot használ, amiben minden érték egy termék előnyének a mértékét jelzi a felhasználó számára. Például a Netflix és sok más nagy oldal esetében a felhasználók egy 1-5-ig tartó skálán tudják jelezni, hogy mennyire tetszett nekik az adott film. Szóval a mátrix értékei 1 és 5 között fognak mozogni. [11]

#### ***Termékajánló rendszerek a gyakorlatban***

Szinte az összes B2C elven működő nagyvállalat rendelkezik termékajánló rendszerrel, mint a Netflix, Spotify, YouTube, Amazon, Apple Music, Google, Booking.com, Expedia csak pár a sok közül.

A Netflix egy olyan oldal, ahol körülbelül 17 ezer film és sorozat található. A felhasználónak nagyon nehéz dolga van, ha nincs egy konkrét film, amit meg szeretne nézni, hanem csak úgy böngészne hátha talál valami olyat, ami tetszik neki. 17 ezer filmet végig nézni nem kis időbe telik a felhasználónak, így kitalálták, hogy létrehoznak egy termékajánló rendszert, ami különböző módokon ajánl filmeket ezzel megkönnyítve a felhasználók életét. A Netflixnek nagyon fontos, hogy jó legyen az ajánló rendszere, mivel havi előfizetések vannak és ezt a felhasználó bármikor lemondhatja, így ezekkel az ajánlásokkal is igyekszik megtartani a felhasználóit, és ezt nem is titkolják. [12]

A Netflix weboldalán található egy leírás az ajánlórendszerükről. Azért, hogy a legjobb ajánlást tudják adni a felhasználó számára a következő dolgokat használják fel: a felhasználó

kapcsolatát a szolgáltatással, ajánlanak az megtekintési előzmények és a megtekintett értékelések alapján. Segítségükre van az az elv, hogy hasonló embereknek hasonló az ízlése, ezt hívják társadalom alapú szűrésnek. Továbbá felhasználják a felhasználó adataiból, hogy mikor nézte a filmet, milyen eszköztől nézte, és milyen sokáig nézte. Ezek az adatok, mind segítik az ajánlórendszer hatékonyságát. Viszont nem veszik figyelembe a demográfiai adatokat, mint például a kort és a nemet. Nem csak így segítik a megfelelő film megtalálását, hanem a keresővel is ajánlanak. Például, ha keresünk egy filmet, akkor nem csak az az egy fog megjelenni, hanem több ahhoz hasonló is. Amikor a felhasználó regisztrál, akkor opcionálisan kiválaszthat filmeket, amik tetszettek neki. Később az ajánlórendszer ezekhez a filmekhez hasonlókat fog ajánlani. Ha valaki nem választja ki ezeket a filmeket, akkor a rendszer a legnépszerűbb filmeket fogja neki feldobni az elején. Később pedig minden esetben tud alapozni a megtekintési előzményekre. A felhasználó úgy kapja ezeket az ajánlásokat, hogy amikor belép a profiljába a kezdőképernyőjén személyre szabott ajánlásokat kap. Több sorban vannak filmek és mindegyik sorban balról jobbra vannak rendezve a filmek, kezdve a legerősebb ajánlással a gyengébbekig. Ilyen sorok a Videó folytatása, Népszerű filmek, Díjnyertes filmek stb. [13]

2006-ban a Netflix egy nyílt versenyt hirdetett ki, azzal a céllal, hogy megtalálják a legjobb algoritmust a vélemények, értékelések alapján, úgy, hogy nincs információjuk a filmről vagy a felhasználóról. Az induló csapatok kaptak egy három összetevőből álló adathalmazt, ami tartalmazta a felhasználót, a filmet, és az értékelés dátumát. Viszont a zsűri tudott egy negyediket is, amit a csapatoknak kellett kitalálnia a lehető legnagyobb pontossággal, ez volt maga az értékelés, amit egy 1-5 -ig skálán tudtak megtenni. [14]

A lényeg ugye az volt, hogy a lehető legkisebb eltérés legyen az eredményük és a validációs adatok között. A győztes csapat, BellKor's Pragmatic Chaos névvel 0,8567-es pontosságot tudtak elérni, ezzel 10,06%-ot javítottak a Netflix ajánlórendszerén. Az eredménytáblán látszik, hogy csak az eredmények között csak ezred vagy tízezred különbségek vannak és azok is nagyon sokat tudnak számítani. Érdekesség, hogy a 14. helyen lévő Gravity néven futó csapat magyar informatikusokból állt, akik 2007 januárjától 2007 májusáig az első helyen álltak. És ma már termékajánló rendszerekkel foglalkozó vállalkozást indítottak. [3]

## Termékajánló módszerek

### *Kollaboratív ajánlás*

A kollaboratív ajánlásokat az első termékajánló rendszerek között tartják számon, amit a GroupLens nevezetű csoport talált ki először. A GroupLens Minnesotai Egyetem egy csoportja, akik úttörőnek számítanak az ajánlórendszerek területén. [15] A kollaboratív ajánlási módszerek a minden egyes felhasználónak egyedi ajánlást készít, a felhasználó értékelései, véleményei és korábbi vásárlásai alapján. Ehhez nem kell külső információ, mint például kor, lakhely a felhasználóról sem a termékről. [6] A kollaboratív ajánlás alapja, hogy a hasonló felhasználóknak azonos az érdeklődési körül és hasonló termékeket vásárolnak. A kollaboratív ajánlási rendszerek úgy működnek, hogy felépítenek egy adatbázist, ami felhasználó-termék mátrix formájába rendezi a termékeket és felhasználókat. Ahogy a különböző algoritmusok végig járnak ezen a mátrixok összegyűjtik a hasonló profillal rendelkező felhasználókat és egy csoportba helyezi őket, amit szomszédoknak neveznek. Egy felhasználó arról a termékről fog ajánlást kapni, amelyik ő még nem értékelt vagy nem látott viszont a csoportjában már valaki értékelte és tetszett neki, így valószínűleg a felhasználónak is tetszeni fog. [16]

Két véleményfajta különböztetünk meg:

- Explicit értékelés: Azt mutatja meg közvetlenül, hogy a felhasználó hogyan értékelt az adott terméket (ez lehet film, könyv, kávézó stb.). Ez egyértelmű vélemény, biztosan tudjuk, hogy a felhasználó, mit gondol a termékről.
- Implicit értékelés: Nevezhetjük ezt 'sütinek' is, ez szolgáltatja az információt arról, hogy mire kattintott rá, milyen oldalt/terméket nézett meg a felhasználó, mire keresett rá, mit kedvelt a közösségi médiában, mennyi időt töltött egy oldalon. Szóval ez csak helyettesként szolgál és heurisztikát nyújt arról, hogy a felhasználó mennyire kedvel egy terméket. Mondhatjuk, hogy ez egy közvetett vélemény. Például van egy zeneszám, amit a felhasználó csak egyszer hallgat meg és nem ad róla véleményt. Ilyenkor lép életbe az implicit vélemény alkotás, mivel nem tudhatjuk, hogy azért hallgatta egyszer mert éppen nincs ideje vagy azért, mert nem tetszett neki. [6]
- Hibrid értékelés: Az explicit, és az implicit vélemények keveréke, ami tartalmazza mindkettő erősségeit, és csökkenti a gyengeségeiket, hogy egy még jobb rendszer jöheszen létre. Ezt úgy érhetjük el, hogy ha implicit adatokat használunk, akkor ellenőrizzük az explicit véleményeket is. [16]

Összevetve a kettőt az explicit vélemény sokkal egyszerűbb és egyértelműbb, mint az implicit, mivel nem kell kitalálnunk, hogy mit szeret a felhasználó, hanem ő maga megmondja, míg implicit vélemény esetében csak sejtésünk lehet.

## **Szomszéd alapú megközelítés**

Másnéven memória alapú algoritmusok a legkorábban fejlesztett kollaboratív algoritmusok között találhatók. Ezeknek az algoritmusoknak az alapja, hogy a hasonló felhasználók, hasonlóan értékelnek és a hasonló termékek hasonló értékeléseket kapnak.

### *Felhasználó alapú modellek*

Ez az úgynevezett „Hozzád hasonló felhasználók ezt a terméket is kedvelték.” Ebben az esetben az algoritmusok megkeresik a cél felhasználóhoz hasonló felhasználókat és megvizsgálja, hogy a hasonló felhasználók által jóra értékelt termékek közül, melyiket nem értékelte még a cél felhasználó és ezeket a termékeket fogja a rendszer ajánlani neki. [15]

#### Pearson – féle korreláció

A Pearson korrelációt Karl Pearson, brit matematikus alkotta. A Pearson – féle korrelációt, akkor használjuk, ha meg szeretnénk találni a lineáris kapcsolatot kettő elem között. Az értéke negatív 1 és pozitív 1 között mozoghat.

Ez az egyik mérték, amivel meg tudjuk határozni két elem közötti hasonlóságot. Először ki kell számolni az átlagát mindegyik felhasználónak, majd utána használhatjuk a Pearson korrelációt, ami a következőképpen néz ki:

$$U_{ik} = \frac{\sum_j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_j (v_{ij} - v_i)^2 \sum_j (v_{kj} - v_k)^2}}$$

Az egyenletben az  $u$  jelenti a felhasználókat,  $v$  az értékeléseket, az  $i$  és  $k$  különbözteti meg a két felhasználót. [17]

### *Termék alapú modellek*

Ezt a módszer úgy lehet egyszerűen megfogalmazni, hogy „Felhasználók, akik ezt a terméket kedvelték, ezt a terméket is kedvelték.” Ez a fajta szűrő nem a felhasználók közötti hasonlóságot nézi, hanem a termékek közöttit, amiket a felhasználó értékelt. A hasonlóságot ebben az esetben is a fent említett két módon tudjuk kiszámítani. A fő különbség a felhasználó és a termék alapú szűrők között, hogy a termék-alapú szűrők függőlegesen töltik ki a sorokat a táblázatban, amíg a felhasználó alapúak vízszintesen. [6]

#### Cosinus hasonlóság

Az úgynevezett nyers értékeléseknél használják ezt a számolási módszert. Általában ezt a

számolást nem a hasonlóan értékelt termékekre végzik, hanem az összes lehetséges elemet megvizsgálják vele. [15]

$$AdjustedCosine(i,j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i} s_{uk}^2} \sqrt{\sum_{u \in U_j} s_{uj}^2}}$$

### ***Tartalom alapú termékajánló rendszerek***

*„Ha van 4,5 millió vásárlónk, nem elég, ha egy boltunk van. Akkor 4,5 millió bolt kell.”*

Mondta Jeff Bezos az Amazon alapítója a Washington Postnak. [18]

A tartalom alapú ajánló rendszereknek ez az alapja. Valóban igaz, hogy minden felhasználónak más az érdeklődése, például egy fiatal anyuka baba játékokat keres és nem veszi jó néven, ha csak a programozást ismertető könyveket találja, ezért nagy eséllyel legközelebb nem fogja meglátogatni az oldalt.

A tartalom alapú rendszer megtanulja, hogy olyan elemeket ajánljon, amelyek hasonlóak, mint amiket a felhasználó a múltban kedvelt. Az elemek hasonlóságának kiszámolása, úgy történik, hogy a rendszer összehasonlítja a jellemzőit a két elemnek. Például, ha a felhasználó pozitívan értékeli egy filmet, ami a komédia műfajba tartozik, akkor az ajánló rendszer meg tudja tanulni, hogy milyen másikat tud ajánlani ugyanebben a kategóriában.

Az ajánlási folyamat három lépésben történik, melyek mindegyikét külön komponens kezeli.

**Tartalomelemző:** Ha az információnak nincs megfelelő felépítése, szükséges egy előfeldolgozási lépés, hogy ki tudjuk szűrni a nekünk fontos információt. Szóval ez a lépés fontos, hogy a következő lépésekben olyan információt tudjunk tovább adni, amivel könnyebben lehet dolgozni. Ez szolgáltatja a bemenetet a következő két lépéshez.

**Profilelemző:** Ez a modul összegyűjti a felhasználó által előnyben részesített adatokat, és megpróbálja általánosítani ezeket az adatok, a felhasználó profiljának felépítésének érdekében. Általában ennek a realizációnak az alapja a gépi tanulás, ami képes arra, hogy felállítsa a felhasználó érdeklődési körét arra támaszkodva, hogy ismeri a felhasználó múltjából kedvelt/nem kedvelt termékeket, és tárolja a felhasználó korábbi visszajelzéseit.

**Szűrő komponens:** Ez a modul hasznosítja a felhasználó profilját, ahhoz, hogy megtalálja a megfelelő terméket, azzal, hogy összehasonlítja a profilon található elemeket az ajánlandó elemmel. Ha az elem megfelelő, akkor hozzáadja egy lehetséges érdeklődési listához. [6]

## **Előnyök és hátrányok**

Előnyei:

- Felhasználói függetlenség: Csak azokat az ajánlásokat használja, amiket a felhasználó profilja biztosít, mivel ezek a saját előzményei alapján történnek nem pedig a Kollaboratív ajánlás esetében megismert hasonló felhasználók alapján.
- Átlátszóság: Egyértelműen fel tudja sorolni a termékek jellemzőit vagy leírásukat az ajánlások listájáról. Ezek a jellemzők alapján tudjuk eldönteni, hogy megéri-e ajánlani a terméket. Amíg ez a Kollaboratív ajánlás esetében egy fekete doboz, mivel csak a hasonló felhasználókat ismerjük.
- Új termék: Képes olyan terméket is ajánlani, amit még előtte senki nem értékelt.

Hátrányai:

- Korlátozott a tartalom elemzés: Ahhoz, hogy rendesen működjön a rendszer ismerni kell a termék bizonyos jellemzőit, például film esetében a rendezőt, színészeket. Fontos, hogy legyen elég információnk.
- Túlműködés: Ez azt jelenti, hogy a rendszer csak hasonló értékeléseket keres és emiatt nehéz az újakat megtalálni.
- Új felhasználó: Ahhoz, hogy ajánlást tudjunk csinálni először kellene előzmények, amik alapján meg tudjuk csinálni az ajánlást, viszont egy új felhasználó esetében nincsenek még értékelések, szóval a rendszer még nem tud megbízható ajánlást létrehozni. [6]

## ***Tudás alapú termékajánló rendszerek***

A tudás alapú ajánlórendszerekkel a nem mindennapi termékeket szokás ajánlani, hanem ha az ember valami nagyobb szeretne vásárolni. Például autót, utazást, lakást vagy luxuscikkeket. Amiknél nagyon fontos, hogy az ajánlórendszer jól ismerje a vevőt, hogy tudjon neki megfelelő termékeket ajánlani. A felhasználó, ha egy házat vesz sokkal jobban körül kell járnia a témát, mint egy film megnézésénél, aminek nem lehet semmi anyagi következménye, hogy megnézte-e vagy sem. Viszont, ha vesz egy kocsit, ami fél év múlva már nem indul el, akkor annak nagyon rossz következményei lehetnek. Általában, ha az ember ilyen terméket vesz, nagyobb eséllyel ír véleményt a termékről, vagy az ajánlásról, mint egy film esetében. Ezért, ha úgy nézzük nincs nehéz dolga az ajánlórendszernek, mivel explicit értékeléseket tudnak használni. Mivel, ha a felhasználó konkrétan leírja, hogy milyen terméket szeretne, ezért a tudás alapú ajánlórendszereknek az interaktivitás kulcsfontosságú eleme. Nehéz egyértelműnek venni az értékeléseket, mivel általában

nagyon összetett termékekről van szó. Például egy autó esetében előfordulhat az, hogy egy a felhasználó számára minden paraméter megfelelő, kivéve a színe. Ilyen esetben nehéz eldönteni, hogy ajánlhatjuk-e ezt az autót vagy sem. Még egy probléma a tudás alapú ajánlásokkal, hogy ezek a rendszerek idő-érzékenyek, azaz, ha van egy 10 éves laptop, amiről nagyon jó vélemények vannak, az nem jelenti azt, hogy ez a laptop még mai szemmel is ugyanolyan jónak számít, mint 10 évvel ezelőtt. [15]

A tudás alapú módszerek két fő csoportba sorolhatók aszerint, hogy milyen interakciót használnak, ezek a kényszer alapú és az eset alapú ajánlórendszerek.

### **Kényszer alapú ajánlórendszer**

A kényszer alapú ajánlórendszerek megengedik a felhasználónak, hogy specifikálja az igényeit, szóval egy ilyen ajánlás több szűrőt igényel. Nehézsége, hogy a felhasználó sokszor nem ugyanazokat a szűrőket emeli ki, mint amik a rendszerben megtalálhatóak, így az algoritmusoknak ez egy plusz lépés, hogy hasonló formára hozza a szűrőket. Egy ház ajánlásnál a következő szűrőket használhatjuk: családiállapot, család mérete, város vagy falu, minimum hálószoba, maximum érték, négyzetméter stb. Ha ezek közül párat megad a felhasználó, vagy implicit módon lehet következtetni, akkor abból a többbit a rendszer ki tudja következtetni. Például, ha a felhasználó megadta, hogy egyedülálló, akkor az ajánlórendszer valószínűleg egy kisebb házat fog ajánlani, mint egy 3 gyerekes családnak. [15]

### **Eset alapú ajánlórendszer**

Eset alapú ajánlórendszerek esetében keresnek hasonló eseteket és azokat veszik alapul az ajánlás elkészítéséhez. Hasonlóan a kényszer alapú ajánlásokhoz, Ennél a módszernél is szűrőkre van szüksége az algoritmusnak, hogy ajánlást tudjon készíteni. Csak ennél a módszernél nincsenek annyira kötött értékek, mint például a szobák száma stb. Hanem itt előre legyártott szűrők vannak. Ha az algoritmus nem talál egy olyan terméket, ami a felhasználó minden igényét kielégíti, akkor keres egy hasonlót és azt ajánlja. Szóval itt nem probléma, ha a felhasználó és a rendszer nem pontosan ugyanazokat a szűrőket alkalmazzák. [15]

Az első tudás alapú ajánlórendszer az Entree Chicago, amit a chicagói egyetem informatikusai fejlesztettek 1996-ban és 1997-ben. Az oldalon étterem ajánlásokat lehetett kérni. Kétféleképpen lehetett ajánlást kérni. Az első verzió a kényszer alapú ajánlást használta, azaz több paraméterre rákérdezett, étel típusa, árak, stílus, légkör, események és ezek alapján keresett egy éttermet, ami a lehető legtöbb kérésnek eleget tudott tenni. A másik



verzió pedig az eset alapú ajánlást használja, aminél csak egy éttermet kér és egy várost és ehhez az étteremhez fog hasonlót keresni az adott városban. Majd a találatokat szépen kilistázza és a felhasználó ki tudja választani a neki legjobban tetszőt. [19]

### ***Társadalom alapú termékajánló rendszerek***

Ez a fajta a termékajánló rendszereknek a felhasználó barátaira építi az ajánlásait. Követi a jól ismert mondatot: „Mondd meg kik a barátaid, és én megmondom ki vagy te.” Ez a rendszer felhasználja az embereknek azt a tulajdonságát, hogy jobban adnak a barátaik véleményére, mint idegen emberek által írt értékelésekre. Az ajánlórendszert nagyban segíti a közösségi oldalak virágzása, ahol az emberek akarva, akaratlanul is látják, hogy ha egy ismerősük értékelt valamit. Szerintem mindenki így van, hogy ha rákeres egy termékre valamelyik közösségi platformon és látja, hogy egy ismerőse jó véleményt írt róla, akkor szívesebben veszi meg, mintha idegenek véleményeit olvasgatná. [6]

### ***Demográfiai termékajánló rendszerek***

Ez a típusa az ajánló rendszereknek a felhasználó demográfiai alapjain nyugszik. Azt feltételezi, hogy különböző ajánlásokat kell gyártani a különböző demográfiai adatokkal rendelkező felhasználóknak. Például ilyen demográfiai adat az életkor, lakhely, nyelv. A marketingben ezt a fajta megközelítés nagyon népszerű, viszont az termékajánló rendszerek esetében már nem örvend akkora népszerűségnek. [6]

A marketing esetében ezt úgy lehet észrevenni, hogy a boltokban, mint az Aldiban, Praktikerben vagy az Ikeában megkérdezik a vásárlóktól, hogy mi az irányítószámuk. Ezt azért, kérdezik mert így fogják tudni, hogy melyik irányítószámú faluba, városrészre milyen újságot érdemes küldeni. Például észrevették, hogy akik a belvárosban laknak azok kisebb valószínűséggel fognak fűnyírót vásárolni, mint a falusiak, így nekik nincs is értelme bedobni a prospektust az új fűnyírókról. Amíg régen erre nem figyeltek, hanem mindenki kapott mindenféle szóróanyagot, aminek az emberek nem örültek, nagyon sok szemét keletkezett vele és még a boltoknak is plusz kiadást jelentett.

## Gépi tanulás alapú módszerek

A gépi tanulást, mint fogalmat először Arthur Samuel írta le 1959-ben, amit a következőképpen definiált: „*A tudomány azon területe, ami lehetővé teszi, hogy a számítógépek programozás nélkül tanuljanak*”. [20]

A gépi tanulás a mesterséges intelligenciának egy olyan ága, ami az önállóan tanulni képes rendszereket foglalja magába, ez a tanulás egy olyan algoritmust követ, ami becsléseket készít már meglévő adathalmazból. Például gyakran alkalmazzák szakértői rendszerek létrehozásakor, illetve az adatbányászat területén is. [21]

### ***Felügyelet nélküli tanulás***

A felügyelet nélküli tanulást, akkor célszerű használni, amikor nagyon nagy az adatkészletünk, ami még jelöletlen adatokat tartalmaz. Ebben az esetben az adatok megértéséhez olyan algoritmusokat használhatunk, amelyek képesek megtalálni az adatok közötti mintákat és el tudják látni címkével az adatokat. Lényegében lehetővé teszi, hogy később felügyelt tanulást lehessen használni. Miután megtörténik a címkézés a felügyelt tanuláshoz hasonlóan a felügyelet nélküli tanulás is mintákat keres

A kettő módszer között csak annyi a különbség, hogy a felügyelet nélküli tanuló algoritmusok nem értik az adatokat. Sok esetben a felügyelet nélküli tanulás gyorsabban képes, pontosabb eredményt megadni, mint a felügyelt tanulás, mivel maga az algoritmus építi fel a címkéket és így könnyebben tud vele dolgozni, mintha meg kéne értenie a minták elhelyezkedését.

Például ezeket az algoritmusokat használják a spam felismerő rendszerek is, mivel amikor beérkezik egy e-mail nincs azonnal címkével ellátva, hanem kell egy algoritmus, ami fel tudja címkézni és ezután a címke alapján eldönti a levélről, hogy spam vagy nem spam. [21]

### ***Felügyelt tanulás***

A felügyelt tanulás egy adatkészlettel dolgozik, amiről tudjuk, hogy milyen osztálycímkékkel vannak ellátva benne lévő adatok. Szóval minden adathoz tartozik egy osztálycímke. A felügyelt tanulás lényege, hogy megtalálja, hogy a még nem ismert példákhoz milyen osztálycímke rendelhető a különböző minták alapján. Ezeket a mintákat jellemzőkkel látjuk el, amikből következtethetünk egy még ismeretlen minta osztályára. [22]

Például vegyünk egy tanuló adathalmazt, ami zenéket tartalmaz. Ezt felruházzuk jellemzőkkel, lehet ez műfaj, ritmus, nyelv, kiadás éve, jelölések stb. Ezekhez rendelünk egy-egy címkét, amivel megnézzük, hogy egy adott személynek tetszettek-e az adathalmazban található számok vagy nem. Ezek után vizsgáljuk a teszt adathalmazban található zenéket és összehasonlítjuk a jellemzőiket a tanuló adathalmazban található zenék jellemzőivel és ez alapján tudunk ezekhez is címkéket rendelni a tanuló adathalmazban lévő zenékhez, amik egy becslést adnak.

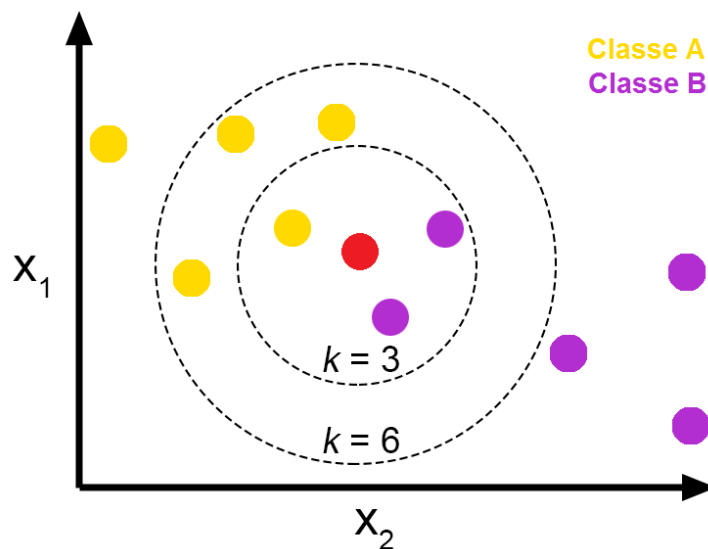
A felügyelt tanulásnak kettő verziója létezik. Amikor a címkék folyamatosak, akkor regresszióknak hívjuk. Amikor az adatok egy véges készletnek az értékei, akkor pedig klasszifikációt használunk.

Számos probléma megoldására használják a felügyelt tanulási módszereket, mint például a beszéd felismerés, kockázat elemzés és amire én is használni fogom, ajánló rendszerek. [21]

## Legközelebbi szomszédok

Ez egy felügyelt tanuló algoritmus, szóval az elemei címkékkel vannak ellátva. Nagyon egyszerű logikára épül az algoritmus. Rendelkezésünkre áll egy nagy adatkészlet, aminek az elemei címkézve vannak, ez lesz a tanuló adatkészlet. Az adatkészlet elemei lehetnek címkézve diszkrétan, akkor klasszifikációról beszélünk, illetve címkézhetjük őket folytonosan ezt pedig regresszióknak hívjuk.

Az adatkészletet képzelhetjük sok kicsi pontnak, amik nagyjából címkék szerint vannak rendezve. Kapunk egy új adatot, aminek nincs még címkéje, de el kell róla döntenünk, hogy melyik csoportba tartozik. Az algoritmusnak van egy argumentuma, a  $k$ , ez azt határozza meg, hogy hány közeli szomszédot vizsgálunk. Az algoritmus úgy dönti el, hogy milyen címkével lássa el az új elemet, hogy megkeresi a  $k$  db legközelebbi elemet és összegzi, hogy melyik címkéből mennyi van, az új adatot pedig felruházza azzal a címkével, amelyik legtöbbször előfordul a  $k$  darab szomszéd között. [23]



*A kép a legközelebbi szomszéd algoritmust szemlélteti. [24]*

A képen látszik, hogy sokat tud változtatni az eredményen, ha kisebb vagy nagyon  $k$ -t választunk. A képen, ha  $k$  egyenlő hárommal, akkor egyértelműen a lila csoportba fog tartozni a piros pötty, de ha  $k$ -t hatnak vesszük, akkor már torzulhat és a sárga címke lesz az erősebb.

Az eredményt befolyásolhatja, hogy milyen számítást alkalmazunk a szomszédok kereséséhez, azaz, milyen módszerrel számoljuk ki a távolságokat.

## Döntési fa

Ez az algoritmus az egyik legrégebbi és legerősebb algoritmus. Különlegessége, hogy nemlineáris döntéseket tudunk vele hozni lineáris felületen. A döntési fák olyan nem paraméteres felügyelt tanulási metódusok, amelyeket klasszifikációra és regresszióra használhatunk. A cél egy olyan modell létrehozása, ami megjósolja egy célváltozó értékét azáltal, hogy megtanulja az egyszerű döntéseket az adatok jellemzőiből. A döntési fa algoritmust, úgy kell elképzelni, mint egy családfát, aminek van egy gyökere és onnan ágaznak szét a további ágak, levelek. Azért is hívják döntési fának, mivel van egy alap állítás és az algoritmusnak el kell döntenie, hogy az állítás igaz-e vagy sem és ettől függően lép tovább a következő ágra, ahol újabb állítást talál, és ez így megy addig, amíg el nem éri az algoritmus az utolsó levelet.

Ajánlórendszerekénél úgy lehet ezt elképzelni például, hogy van egy felhasználó, akinek szeretnénk ajánlani egy olyan filmet, amit még nem látott, de valószínűleg tetszene neki. Az algoritmus kérdéseket tesz fel magának, mint például látta már a felhasználó ezt a filmet, tetszenek neki az ehhez hasonló filmek stb. És ha mindegyik kérdésre igennel tud felelni az algoritmus akkor ajánlani fogja a felhasználónak a filmet. [25]

Előnyei: könnyű megérteni és interpretálni, a fákat vizualizálhatjuk. Képes több-kimenet problémákat is megoldani. Jobban kezeli a kiugró értékeket, mint az SVM. Ezeken kívül még van több előnye is.

Hátrányai: létrejöhet egy olyan jelenség, aminek overfitting (túltanulás) a neve. Ez úgy jön létre, hogy az algoritmus túl mélyen beleássa magát a döntésekbe és felesleges lépéseket tesz, amik már nem segítik a becslést. A működésének egy fa az alapja, ami döntések alapján választja szét az adatokat. Akkor célszerű ezt használni, amikor nem tudunk egy egyszerű vektort rakni az adatainkra, mivel nem helyezkednek el olyan szabályosan. Ilyenkor kérdéseket kell feltenni és a válaszok alapján felosztani az adatokat. Így több halmaz jön létre. Például adott egy olyan adatkészlet, aminek a jobb felső sarkában vannak egy fajta értékek a többi részében pedig egy másik fajta, ilyenkor célszerű először mondjuk az x-tengelyen elválasztani őket egy kérdéssel, hogy egy kiválasztott pont fölött vagy alatt vannak-e, majd ugyanezt megcsinálni az y-tengellyel is. [26]

## Véletlen erdők

A Random Forest Classifier kitalálása Leo Breiman, a Berkeley egyetem professzorához köthető, észrevételeit a Random Forests című publikációjában írta le 2001-ben. [27]

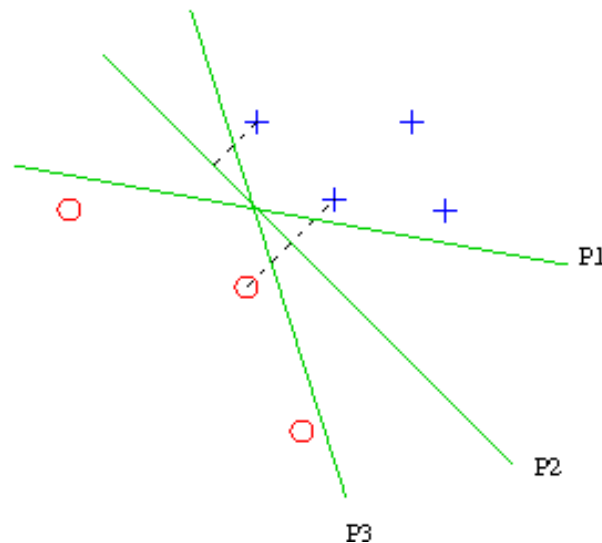
Ez egy olyan átlagolási algoritmus, amely véletlenszerű döntési fákon alapul. Ez azt jelenti, hogy sokféle osztályozót hoznak létre a véletlenszerűség bevezetésével az osztályozó konstrukcióban. Az együttes előrejelzése az egyes osztályozók átlagolt előrejelzése. olyan metaadat becslés, amely számos döntési fa osztályozóhoz illeszkedik az adatminták különböző részmintáihoz, és átlagolással használja a prediktív pontosságot és a túlszabályozás ellenőrzését. Az alminták mérete mindig megegyezik az eredeti beviteli minta méretével. Amikor hívásra kerül az algoritmus, bele kell írni, hogy hány darab fát szeretnénk benne használni, ezt jelenti a kódban látható *n\_estimators=10* paraméter. A tíz az alapértelmezett érték, viszont ezt lehet változtatni, bármennyire, amennyire szeretnénk, hasonlóan, mint a döntési fáknál. [28]

## Adaptív erősítés

Az Adaptive Boosting legfontosabb alapelve az, hogy egy sor gyenge tanulók sorozata, vagyis olyan modellek, amelyek csak kismértékben jobbak, mint a véletlenszerű találgatás, mint például a kicsi döntési fák. Mindegyikük előrejelzéseit súlyozott többségi szavazással kombinálják a végső előrejelzés elkészítéséhez. Minden egymást követő iterációnál a minta súlyait egyedileg módosítják, és a tanulási algoritmust újra alkalmazzák az újra számolt adatokra. Egy adott lépésnél azokat a képzési példákat, amelyeket az előző lépésben indukált modell által helytelenül előre jelöltek, súlyuk megnövekedett, míg a súlyok csökkentek a korábban előre jelzettekénél. Az iterációk folytatásaként a nehezen megjósolható példák egyre növekvő befolyással bírnak. Minden későbbi gyenge tanuló kénytelen koncentrálni a példákra, amelyeket a sorozatban az előzők nem fogadtak el. Ennek az algoritmusnak is van paramétere, hasonlóan, mint az előzőnek. Csak itt a becslések maximális számát jelenti, amelyen a feljavítás megszűnik. A tökéletes illeszkedés esetén a tanulási folyamat korai szakaszban leáll. [29]

## Támogató vektor gépek

Rövidebb nevén SVM. Az Support Vector Machine olyan felügyelt tanulási módszereket takar, amiket osztályozáshoz, regresszióhoz vagy kiugró érték felfedezéséhez használhatunk. Több dimenziós terekben is hatékonyan működik, illetve akkor is ha több a dimenziók száma, mint a példák száma. A döntési funkció egy részhalmazát, így nincs szüksége annyi memóriára. Egy hátránya, ha a jellemzők száma sokkal nagyobb, mint a minták száma, elkerüljük a kernelfunkciók kiválasztását és a regularizációs kifejezést. Több fajta klasszifikációja létezik. A működése, a nevéből is lehet gondolni, hogy egy vektoron alapszik. Az algoritmus létrehoz egy vektort, amivel két részre bontja az adatokat, ezt nevezi hyperplane-nek (hipersík). Egy adatkészletben a hyperplane-t nagyon sokféleképpen lehet elhelyezni, ha csak azt vesszük figyelembe, hogy ketté bontsuk az adatkészletet. Ezért találták ki, hogy legyen egy margin (margó) nevezetű 'támasz'. Ennek az a lényege, hogy mindegyik oldalról a legközelebbi pontoknak a lehető legnagyobb távolságban kell lenniük. [30] Az ábrán láthatók a különböző lehetséges vektorok és a pontok, amik között a legjobb elválasztó vektort keressük.



*A kép az SVM algoritmus működését szemlélteti. [31]*

Az ábra nagyon jól mutat három lehetséges vektort is, de látszik, hogy a P1-es és a P3-as nagyon közel található az adatokhoz, ez zajokat okozhat a későbbiekben. Viszont a P2-es vektor az egy nagyon jó hipersík, mivel az van legtávolabb a két legközelebbi ponttól.

## Mesterséges neurális hálózatok

Az artificial neural networks az emberi agy működése alapján dolgozik, ahol szintén megfigyelhetők a hálózatok, amiken keresztül az információ eljut egyik helyről a másikra.

Minden neurális hálózat felépítési alapjai a neuronok, ezek általában egyszerű matematikai függvények, amik szorzatokból és összegekből állnak. Ha kettő ilyen neuront összekötünk, akkor már van egy neurális hálózatunk. Ezeket a hálózatokat sokféleképpen fel lehet építeni, például lehet előrecsatolt, de léteznek olyan hálók is, amiknek vannak visszacsatolt elemei is. [32]

### **Bayes-i osztályozó**

A Bayesian Classifier neve is mutatja ez a módszer Thomas Bayes valószínűségszámításban használt tételén alapszik. A Bayes tételén alapuló algoritmus a leghatékonyabb osztályozó algoritmus. Ez az algoritmus megmutatja az összes osztálynak az egyetlen valószínűségi összegét.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

A Bayes tétel alapja, hogy ha ismerjük az A és B események valószínűségét, akkor meg tudjuk mondani, hogy mennyi a valószínűsége annak, ha bekövetkezik A, akkor B is be fog következni. [33]



## Irodalomjegyzék

- [1] G. Takács, I. Pilászy, B. Németh és D. Tikk, „Scalable Collaborative Filtering Approaches,” 2009. [Online]. Available: <http://www.jmlr.org/papers/volume10/takacs09a/takacs09a.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [2] R. Collobert, S. Bengio és J. Mariéthoz, „Torch: a modular machine learning software library,” 30. október 2002. [Online]. Available: <http://www.idiap.ch/ftp/reports/2002/rr02-46.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [3] „Netflix Prize,” Netflic Inc., 1997-2009. [Online]. Available: <https://www.netflixprize.com/leaderboard.html>. [Hozzáférés dátuma: 23. április 2019.].
- [4] R. Singh, K. Chuchra és A. Rani, „A Survey on the Generation of Recommender Systems,” 3. március 2017. [Online]. Available: <https://pdfs.semanticscholar.org/7c38/fa13e9a843e73ede4a935a10565d2594a249.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [5] „kaggle.com,” [Online]. Available: <https://www.kaggle.com/rounakbanik/movie-recommender-systems>. [Hozzáférés dátuma: 14. április 2019].
- [6] F. Ricci, L. Rokach, B. Shapira és P. Kantor B., Recommender Systems Handbook, Springer, 2010, p. 845.
- [7] J. Beel, S. Langer, M. Genzmehr és A. Nürnberger, „Persistence in Recommender Systems: Giving the Same Recommendations to the Same Users Multiple Times,” 2013. [Online]. Available: [http://docear.org/papers/persistence\\_in\\_recommender\\_systems\\_-\\_giving\\_the\\_same\\_recommendations\\_to\\_the\\_same\\_users\\_multiple\\_times.pdf](http://docear.org/papers/persistence_in_recommender_systems_-_giving_the_same_recommendations_to_the_same_users_multiple_times.pdf). [Hozzáférés dátuma: 23. április 2019.].
- [8] M. Montaner, B. López és J. Lluís de la Rosa, „Developing Trust in Recommender Agents,” [Online]. Available: <http://eia.udg.es/arl/ramonycajal/156-montaner.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [9] J. A. Konstan és J. Riedl, „Recommender systems: from algorithms,” Springer, 2012.
- [10] D. Goncalves, M. Costa és F. M. Couto, „A Flexible Recommendation System for Cable TV,” 2016.
- [11] D. Jannach, P. Resnick, A. Tuzhilin és M. Zanker, „Recommender Systems - Beyond Matrix Completion,” 2016.
- [12] L. E. Molina Fernández, *Recommendation System for Netflix*, 2018.
- [13] „How Netflix’s Recommendations System Works,” Netflix Inc., [Online]. Available: <https://help.netflix.com/en/node/100639>. [Hozzáférés dátuma: 23. április 2019.].
- [14] „Kaggle.com,” Kaggle Inc., [Online]. Available: <https://www.kaggle.com/netflix-inc/netflix-prize-data>. [Hozzáférés dátuma: 23. április 2019.].
- [15] C. C. Aggarwal, Recommender Systems The Textbook, Springer, 2016.
- [16] F. Isinkaye, Y. Folajimi és B. Ojokoh, „Recommendation systems: Principles, methods and evaluation,” 2015.
- [17] J. W. Grice, „Pearson’s correlation coefficient,” november 2013. [Online]. Available:

- [http://psychology.okstate.edu/faculty/jgrice/psyc5314/pearsons\\_r.pdf](http://psychology.okstate.edu/faculty/jgrice/psyc5314/pearsons_r.pdf). [Hozzáférés dátuma: 23. április 2019.].
- [18] L. Walker, „Amazon Gets Personal With E-Commerce,” *Washington Post*, 8 november 1998.
- [19] R. Burke, „Knowledge-based recommender systems,” [Online]. Available: <https://pdfs.semanticscholar.org/1b50/ff4e5d8420f3ffae7de2a060b5a6fd4b8023.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [20] A. Munoz, „Machine Learning and Optimization,” [Online]. Available: <https://pdfs.semanticscholar.org/7fbb/a79630b5a09dd66ab13f00c3aefaa56cf268.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [21] J. Hurwitz és D. Kirsch, *Machine Learning For Dummies*, John Wiley & Sons, Inc., 2018.
- [22] „Felügyelt tanulás,” Szegedi Tudományegyetem, [Online]. Available: <http://www.inf.u-szeged.hu/~ihegedus/teach/04-KNN.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [23] N. Kumar, L. Zhang és S. Nayar, „What is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images,” [Online]. Available: [http://www1.cs.columbia.edu/CAVE/publications/pdfs/Kumar\\_ECCV08\\_2.pdf](http://www1.cs.columbia.edu/CAVE/publications/pdfs/Kumar_ECCV08_2.pdf). [Hozzáférés dátuma: 23. április 2019.].
- [24] „towardsdatascience.com,” [Online]. Available: <https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d>. [Hozzáférés dátuma: 14. április 2019.].
- [25] L. Rokach és O. Maimon, „Decision Trees,” in *Data Mining and Knowledge Discovery Handbook*.
- [26] „Decision Trees,” scikit-learn, [Online]. Available: <https://scikit-learn.org/stable/modules/tree.html>. [Hozzáférés dátuma: 23. április 2019.].
- [27] L. Breiman, „Random Forests,” január 2001. [Online]. Available: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [28] C. Tomasi, „Random Forest Classifiers,” [Online]. Available: <https://pdfs.semanticscholar.org/3052/b67a414adecd7545dad635ca3c8495be9314.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [29] R. E. Schapire, „Explaining AdaBoost,” [Online]. Available: <http://rob.schapire.net/papers/explaining-adaboost.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [30] A. Ben-Hur és J. Weston, „A User’s Guide to Support Vector Machines,” [Online]. Available: <http://pymml.sourceforge.net/doc/howto.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [31] „cs.nyu.edu,” [Online]. Available: <https://cs.nyu.edu/faculty/davise/ai/linearSeparator.html>. [Hozzáférés dátuma: 4. április 2019.].
- [32] A. Krenker, J. Bešter és A. Kos, „Introduction to the Artificial Neural Networks,” [Online]. Available: <http://cdn.intechweb.org/pdfs/14881.pdf>. [Hozzáférés dátuma: 23. április 2019.].
- [33] P. Langley és S. Sage, „Induction of Selective Bayesian Classifiers,” [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1302/1302.6828.pdf>. [Hozzáférés dátuma: 23. április 2019.].

- [34] „Kaggle.com,” [Online]. Available: <https://www.kaggle.com/rounakbanik/movie-recommender-systems>. [Hozzáférés dátuma: 14. április 2019].
- [35] F. h.-r.-s. (. n. 2019-04-14). [Online].