|  |  |
| --- | --- |
| Miskolci Egyetem  Gépészmérnöki és Informatikai Kar  Általános Informatikai Intézeti  *3515 Miskolc-Egyetemváros* |  |

**SZAKDOLGOZAT**

*Feladat címe:*

Mesterséges intelligencia alapú oktatást támogató alkalmazás tervezése

*Készítette:*

Szilvási Péter

*MSc szintű mérnökinformatika szakos*

*Alkalmazásfejlesztő szakirányos hallgató*

*Témavezető:*

Dr. Kovács László

*tanszékvezető, egyetemi tanár*

**Miskolc, 2020.**

# Bevezetés

## Téma választás ismertetése

placeholder

## Miért kell támogatni az oktatást?

placeholder

## Oktatás fontossága

placeholder

## Mesterséges intelligencia és az oktatás

placeholder

# Gépi tanulás menete

## Az adat az új üzemanyag

**Gépi tanulás**

Ha elmélyülünk a gépi tanulás definíciójában, akkor azt találjuk, hogy „*A gépi tanulás az algoritmusok és statisztikai modellek tudományos vizsgálata, amelyet a számítógépes rendszerek egy adott feladat végrehajtásához használnak kifejezett utasítások nélkül, mintákra és következtetésekre támaszkodva*”.

Ha elemezzük az előző definíciót, találunk néhány kiemelt szempontot. Hatalmas tudományos kutatás és erőfeszítés van a növekedés és fejlődés támogatására.

Algoritmusokra és statisztikai modellekre támaszkodik, amelyek azt mondják nekünk, hogy ha el akarjuk sajátítani a gépi tanulást, akkor az algoritmusok, a statisztikák és a valószínűségek átfogó megértését kell kialakítanunk. Célja egy meghatározott feladat végrehajtása, amely elmondja, hogy a gépi tanulási megoldásnak van egy bizonyos hatóköre, például előrejelzés és ajánlás.

Kifejezett utasítások nélkül, ami azt mondja nekünk, hogy a gépi tanulási megoldást nem szabad pontosan beprogramozni a dolgok megtanítására. Mintákra és következtetésekre támaszkodva, amely azt mondja nekünk, hogy gépi tanulási megoldásunk a történeti adatokból bizonyos minták megemésztésére támaszkodik a helyes működés megtanulásához.

**Megoldható problémák gépi tanulással**

Most tárgyaljuk meg a gépi tanulás néhány érdekes, valós alkalmazását a mindennapokban.

* A képfelismerés. Hogyan tudja a mobiltelefon feloldani magát azzal, hogy ha a gazdája megmutatja az arcát a kamerának, miközben nem oldja fel magát más emberek számára? Egy képfelismerési technológiával érik el ezt a funkcionalitást. Ahhoz, hogy fel tudjuk ismerni a kívánt objektumot, a képi adatok (pixeleket) egy gépi tanulási algoritmuson kell kiterjeszteni.
* A rosszindulatú programok észlelése, ahol arra vagyunk kíváncsik, hogy az antivírusok hogyan tudják hatékonyan megismerni az új vírusokat, mielőtt frissítéseket kapnának. Ismét a gépi tanulás a motor, amely ezt működteti.
* A hangfelismerés. Egy másik jól ismert példa, amely a gépi tanulást használja. Legyen szó mobiltelefonról vagy okos otthonról, a gépi tanulás lehetővé teszi a felhasználó hatékony azonosítását.

A gépi tanulás terén szerzett kompetenciák és készségek fejlesztésével fantasztikus megoldásokat tudunk kidolgozni, amelyek a valós problémákkal foglalkoznak és értéket képviselnek az emberiség számára.

Végül érdemes megvitatni, hogy a gépi tanulás miért vált manapság ilyen vonzóvá. Nem is gondolná az ember, hogy már a 60-as évek óta létezik. Azt állíthatjuk, hogy az alapstatisztika és a valószínűség valamiféle gépi tanulás. Az elmúlt években azonban volt néhány mozgatóerő.

**Adat túlcsordulás**

Először is, a problémák nagy méretűvé váltak: ***nagy dimenziószám***. A sok dimenzióval olyan adat, amelynek túl sok jellemzője van, ami szinte lehetetlenné teszi a hagyományos programozással történő elemzést. A nagyon dimenziós adatra példa a páciens egészségi állapota, ahol számos jellemző, például az immunrendszer állapota, a genetikai háttér, a táplálkozás, az operációk, a drog- és dohányfogyasztás.

Hatalmas adat túlcsordulás van, például hírcsatornák, webszolgáltatások, adatbázisok, e-mailek, felmérések adatforrásain. A vállalkozások a közelmúltban észrevették, hogy hatékony módszert kell találni ezen hatalmas adatok megértésére, amelyekből értéket kell termelniük.

A számítástechnika fejlődése, valamint a számítási és tárolási erőforrások folyamatos csökkenése, például a felhőalapú számítástechnikában, hatékonyabbá tette a gépi tanulási algoritmusok megvalósítását. Általában a gépi tanulási algoritmusok erőforrás-igényesek. A kutatások jelentősen megnövekedtek az elmúlt évtizedben. A vállalkozások elkezdték megérteni ennek értékét, ezért számos algoritmust kutattak és fejlesztettek ki.



Tehát, figyelembe véve korábbi információkat, magabiztosan kijelenthetjük, hogy:

*„Az adatok az új üzemanyag.”*

## Miben különbözik a gépi tanulás?

Nos, egy jó kérdés merülhet fel a bennünk, hogy miért nem programozzuk be üzleti szabályainkat és tartomány (domain) ismeretünket a gépi tanulási modelljeinkbe úgy, mint a hagyományos programozásnál? Más szóval, mi a tényleges különbség a gépi tanulás és a hagyományos programozás között?

**Hagyományos programozás és gépi tanulás**

A hagyományos programozás során a teljes tartományú üzleti logikát olyan üzleti szabályok formájában rögzítjük, amelyek kifejezetten programozhatók a kódunkba. Például, ha egy adott vállalat fizetését szeretnénk kiszámítani, akkor minden forgatókönyvre világos üzleti szabályunk lesz, és pontosan tudjuk, hogy mi fog történi. Teljes mértékben megértjük az üzleti területet.

Míg a gépi tanulás során nem teljesen értjük a tartománymodellt, valamennyire tudjuk, hogyan reagál a rendszer bizonyos bemenetekre és kimenetekre. Nem igazán értjük teljesen a mögöttes bonyolult kapcsolatokat a bemenet és a kimenet között. Például egy gépi tanulási algoritmus a vásárlási viselkedés hasonlósága alapján különböző csoportokba sorolhatja az ügyfeleket, annak ellenére, hogy nem értjük teljesen az alapul szolgáló komplex összefüggést. A feladat egy gépi tanulási algoritmusra van delegálva.

A hagyományos programozásban a rendszer esztétikus. A változtatásokat kifejezetten a forráskód vagy a konfiguráció megváltoztatásával kell végrehajtani. Például, ha új szabályt szeretnénk hozzáadni a fizetési rendszerünkhöz, módosítanunk kell a forráskódot. De a gépi tanulásban a rendszer dinamikus és folyamatosan tanul, és a bemenetek és az új tanúsítványok alapján igazodik. Az ügyfelek ápolására szolgáló algoritmusunk idővel jobbá válhat, mivel több trendet tanul meg.

Egy másik különbség az, hogy a hagyományos programozásnál nincs szükségünk semmilyen történelmi adatra vagy tanulási fázisra. A logika pontosan tudja, mit kell tennie. Bérszámítási rendszerünk megfelelően működhet anélkül, hogy úgymond képzési fizetést biztosítanánk. Míg a gépi tanulás során sok történelmi adatra van szükségünk ahhoz, hogy tanulhassunk belőle. Például ügyfeleink megfelelő osztályozásához, a gépi tanulási algoritmusunknak elegendő mintát kell látnia az alapul szolgáló kapcsolatok megértéséhez. Ügyfélápoló algoritmusunknak történelmi adatokra van szüksége, hogy tanuljanak belőlük. Az embernél is hasonló a tanulás, múltbeli eseményekből vonnunk le következtetéseket, majd próbáljuk megjósolni a jövőt.

**Keretrendszerek**

A hagyományos programozásban a logika egyszerű és egyértelmű. Könnyen elolvashatjuk a forráskódot, hogy megtaláljuk az igazságot. A gépi tanulás homályos és bonyolult bizonyos matematikai algoritmusok alapján. Általában könyvtárak vagy olyan keretrendszerek, például a scikit-learn, tensorflow és a Keras segítségével elvonatkoztathatunk a gépi tanulás mögötti bonyolultságtól. A keretrendszerek absztrakciós rétegek, amelyek hozzájárulnak a gyors és egyszerű fejlesztéshez.

## Főbb tanulási típusok

Miután megismertük, hogy mi a gépi tanulás és miben különbözik a tradicionális programozási megközelítéstől, fontos tisztában lenni a gépi tanulási algoritmusok különböző típusaival. Minden algoritmuscsoportnak megvan a maga módszere a tanulásra. Alapvetően három csoportot különíthetünk el: a felügyelt tanulási algoritmusok, a felügyelet nélküli tanulási algoritmusok és a megerősítő tanulási algoritmusok. Nézzük meg, hogyan működik mindegyikük!

**Felügyelt tanulás**

Kezdjük a felügyelt tanulással. A felügyelt tanulási algoritmusok úgy működnek, hogy egy adott bemenetre () egy megfelelő kimenetet () eredményez. A bemeneti érték egy függvényen () keresztül leképezésre, ami egy kimeneti értéket produkál.

Tehát a kimenet függ a bemenettől. A cél az, hogy felfedezzük a (bemenet) és (kimenet) közötti összefüggéseket. Az függvényt leképezési függvénynek (mapping function) is nevezik, ami leírja az és kapcsolatát.

**Itt járok->**Vegyünk egy konkrét példát. Tegyük fel, hogy van egy olyan üzletünk, ahol számos vásárlás, vásárlás dátuma és elköltött összege van. Felügyelt gépi tanulással szeretnénk megtudni a várható értékesítési és vásárlói profilokat. A felügyelt algoritmus azokat a történelmi adatokat veszi fel, amelyek tartalmazzák az X bevitelt, azaz a vásárlások számát, a vásárlás dátumát és az elvégzett összegeket, valamint a kimenetet, amely a problémás tartomány várható értékesítési vagy ügyfélprofilja. A felügyelt gépi tanulási algoritmus rétegek, és ennek eredménye egy gépi tanulási modell. Ezt követően a betanult modellnek új inputot adhatunk, és az új input alapján előrejelezhetjük a jövőbeni értékesítési és vásárlói profilokat. A felügyelt tanulásban előzetesen tudjuk a helyes választ, ami azt jelenti, hogy megfelelő eladásokkal és kategóriákkal rendelkezünk, amelyeket algoritmusunk tanulhat és fejleszthet belőle. A felügyelt gépi tanulási algoritmusok két jól ismert típusa a regressziós algoritmus, amikor az előrejelzett érték folyamatos jellegű, bármi egy bizonyos tartományban. Az eladási előrejelzések bármilyen értéket feltételezhetnek, 13 666, 100, 150 vagy akár milliót.

A másik típus az osztályozási algoritmusok, amikor az előrejelzett érték diszkrét jellegű, ami azt jelenti, hogy az előrejelzett érték lehetséges opcióival rendelkezik. Például egy kiskereskedelmi üzlet minősítheti az ügyfeleket kiemelt, közepes és normális vásárlóknak.

**Felügyelet nélküli tanulás**

Felügyelet nélküli algoritmusok azok, amikor csak bemeneti adatok vannak, és nincsenek kimeneti adatok. Azaz. nincs megalapozott igazság. A cél általában az alapul szolgáló adatszerkezet megértése az adatok helyes felhasználása érdekében. Ezeket az algoritmusokat önszerveződő algoritmusoknak is nevezzük. Tegyük fel, hogy van egy bizonyos adatkészletünk, amelyet A – L betűkkel jelölünk, és szeretnénk megismerni ezeknek az adatkészlet-tagoknak a kapcsolatát vagy asszociációit egy felügyelet nélküli tanulási algoritmus segítségével. Az algoritmus a következő csoportot hozhatja létre az általa kitalált más kapcsolatok vagy asszociációk alapján. A, I, J; B, K, H; C, H, F; E, G, L. Az, hogy miként szervezzük ezeket a kapcsolatokat és asszociációkat, az általunk használt, felügyelet nélküli tanulási algoritmus típusától függ. Vegyünk egy példát. Tegyük fel, hogy három különböző autóból állunk, hétköznapi autókból, sportautókból és elektromos autókból. Az autókat egy felügyelet nélküli tanulási algoritmus segítségével tápláljuk, amely kiszámítja az egyes autókkal kapcsolatos összefüggéseket. És akkor ebből lesz egy gépi tanulási modellünk. A gépi tanulási modell lehet jövőbeli autók számára, és felhasználható a jövőbeli minták előrejelzésére. Mint láttuk, a felügyelet nélküli algoritmusok ahelyett, hogy a visszajelzésekre támaszkodnának, mint a felügyelt algoritmusok, a felügyelet nélküli algoritmusok a közös vonások azonosításán alapulnak, de akár az adatkészlet közötti különbségeken is, bizonyos tulajdonságok hiányában vagy jelenlétében.

A gépi tanulási algoritmusok két jól ismert típusa, amelyek a felügyelet nélküli tanulási megközelítéseket követik, csoportosulnak. Ezt a típusú algoritmust akkor használják, ha egy adott adatbevitelt csoportosítani kívánunk. Például az ügyfelek csoportok szerinti szegmentálása, hogy a hirdetések megfelelő módon foglalkozzanak velük. A klaszterező algoritmus felismeri az ügyfél jellemzőit, mint például az életkor, a lakóterület, a jövedelmi szint és a nem. Számos különböző fürtözési algoritmus létezik, mint például a k-csoportosítás. Társítás, az ilyen típusú algoritmusok célja, hogy asszociációkat hozzanak létre a nagy adatbázisok adatelemei között. Célja a változók közötti hasznos kapcsolatok feltárása. A felügyelet nélküli tanulást azonban nem tárgyalják ezen a tanfolyamon.

**Megerősítő tanulás**

A gépi tanulási algoritmus harmadik típusa a megerősítő tanulás, más néven célorientált tanulás. Az ilyen típusú algoritmusok célja szoftverügynökök előállítása, ami a megerősítés másik neve. tanulási algoritmusok, amelyek a kommutatív jutalmak fogalmának felhasználásával megtanulják a környezetükből tanult feladatot. Az ügynök úgy dönt, hogy végrehajt egy műveletet a környezetben, majd az ügynök tanul ebből a műveletből. A tanulás visszavezethető az ügynökhöz környezeti állapot és jutalom formájában. Ennek felhasználásával az ügynök törli és frissíti belső házirendjét, és több tapasztalatot szerez. Az ilyen típusú algoritmusok célja a sok lépés komplex problémájának megoldása, például az összetett játék megoldása. A végrehajtási képzési algoritmusok üres állapotból indulnak ki, és megfelelő ösztönzők alkalmazásával fokozatosan javítják a teljesítményt. A megerősítő algoritmusokat általában olyan autonóm rendszerekben használják, amelyek emberi útmutatás nélkül hoznak döntéseket. A megerősítő algoritmusokat szintén fejlett témának tekintik, ezért ezen a tanfolyamon nem fogjuk megtenni. Végül hasznos összehasonlítani és szembeállítani a tanulási típusokat két különböző dimenzióból, objektív és tanulási forrásból. A felügyelt algoritmusok célja a jövőbeli értékek vagy kategóriák előrejelzése, és tanulási forrásuk a kimeneti adatkészlet. A felügyelet nélküli algoritmusok célja az adatok rendszerezése az alapul szolgáló struktúra alapján, és a tanulási forrás az az adat, amelyet az adatok visszaadnak. A megerősítő algoritmusok a környező környezetből származó haszon és állapot alapján adaptálódnak. A tanulási forrás természetesen a környező környezet és a haszon. Most nézzük meg a gépi tanulás módjait. Kötegelt vagy offline tanulási módszerek. Ezek gépi tanulási módszerek, amelyeket végpontok közötti gépi tanulási rendszerekben alkalmaznak, ahol a modellt az összes rendelkezésre álló képzési adat felhasználásával képezik egy lövés alatt. Az online képzési módszerek a kötegelt tanuláshoz képest más módon működnek. Az edzésadatokat általában több növekményes kötegben teszik az algoritmusba. Ezeket a tételeket mini kötegeknek nevezzük. Dióhéjban az a különbség a kötegelt tanulás és az online tanulás között, hogy miként tápláljuk az edzés adatait.

## A gépi tanulás folyamata

Machine learning pipeline

Adat pipeline

(Többi pipeline)

## Az adatforrás a kezdetek kezdete

Mik lehetnek az adat

Honnan szerezhető

## Saját adatforrásom ismertetése

Adat alapvetően: Ált leírás: null érték, zajok

Adathalmaz ismertetése

Átvezetés:

*Többet mond egy kép*

*mint száz bekezdés*

# **Reprezentációs módszerek**

## Miért hasznos az adatok megjelenítése?

Az adattudományban az egyik legfontosabb készség az adatok eloszlásának és összetettségének vizualizálása és megértése. Manapság azonban az online tanfolyamok nagy része elsősorban a gépi tanulásra és az algoritmus működésére összpontosít.

Ez nem azt jelenti, hogy a gépi tanulás ismerete elhanyagolandó. Nyilvánvalóan a legjobb adattudós könnyedén eligazodik a gépi tanulási algoritmusok nagy részével. Mindenesetre az adattudomány nem korlátozódik le a gépi tanulásra, inkább az egy képesség. A készség alatt azt kell érteni, hogy mennyire érted a probléma területet (domain), mennyire tudsz elmélyülni az adatokban. Elkapni a rejtett összefüggéseket és megtalálni az adatokban rejlő üzenetet. Az adat beszél magáról. A legjobb mód arra, hogy szóra bírjuk az ***adatmegjelenítés***.

**Mi az adatmegjelenítés?**

Az adatmegjelenítés az információ (adatok) felvételének és vizuális kontextusba, például térképbe vagy grafikonba helyezésének módszere. A fő cél a nagy adatkészletek vizuális grafikába történő szűrése, hogy lehetővé tegye az adatokon belüli összetett kapcsolatok könnyű megértését.

Tehát a vizualizáció választ ad azokra a kérdésekre, amelyeket nem tudunk.

**Miért fontos az adatmegjelenítés?**

Az adatmegjelenítést egyre inkább minden sikeres adatközpontú elemzési stratégia alapvető utolsó lépésének tekintik. Amikor az adattudósok egy összetett projekt közepén vannak, szükségük van egy módra az összegyűjtött adatok megértésére. Figyelemmel kell kísérni és módosítani folyamatot annak biztosítása érdekében, hogy megfelelően funkcionáljon. Az adatmegjelenítés megkönnyíti az adatcsoportok mintáinak, előítéleteinek és kiugró értékeinek észlelését is.

A célközönség vezetése arra, hogy az üzleti felismerésekre összpontosítson, hogy felfedezzék a figyelmet igénylő területeket. Korábban észrevétlen kulcsfontosságú tények feltárása az adatforrásokkal kapcsolatban, hogy a döntéshozók segítsenek adatelemzési jelentések elkészítésében. Segít az érdekelt feleknek és a csapat többi tagjának minőségi információkkal szolgálni azáltal, hogy hatalmas mennyiségű adatot könnyen érthető képpé és grafikává alakít.

**Az adatmegjelenítés előnyei**

Figyelembe véve, hogy az adatok milyen hatást gyakorolnak a vállalkozás növekedésére, íme néhány előny

* Segít felismerni a legújabb fejlesztéseket a termék fejlesztése és az üzleti nyereség növelése érdekében.
* Az adatmegjelenítések megkönnyítik a kis és nagy adatok megértését az emberi agy számára, ami jobb elemzéshez vezet.
* Segít megérteni a történetet - Az emberi agy nem képes egyszerre nagy mennyiségű számot vagy szöveget megérteni, sőt csak elképzelni. Szüksége van egy vizuális ábrázolásra, hogy értelmezze őket, és ennek következtében a nyers adatokat kézzelfogható fogalommá alakítsa.

## Ábrázolási formák

**bevezetés**: történet, gépek előtt, hogy ábrázoltunk, mit ábrázoltunk (barlangokra strigulát stb.). Az ember maximum három dimenziót képes értelmezni. Az adatelemzésnél a dimenziót úgy kell felfogni, mint a tér leírására használtat. Szélesség, hosszúság és magasság helyett adatok tulajdonsága mentén ábrázolunk. Egy csillag esetében például: hőmérséklet, tömeg és fényesség. Az adatok természetéből kifolyólag számtalan dimenzió megjelenítés lehetséges.

**különböző tulajdonságok**at ugyanazon a koordinátán való ábrázolása

**2d**

**3d**

**Hogyan tudunk háromnál több dimenziót ábrázolni?**

...

## Saját adathalmazon bemutatása

**Saját adathalmazon ábrázolása**: Milyen megjelenítést használtam, több adathalmaz bemutatása, észrevételek, következtetések stb.

**Megjelenítési forma bemutatása**: Miért ezt választottam. Mi mit jelent rajta. Hány db adat. Színek, csoportok leírása stb.

**Mit mond az ábra?** Ábráról leszűrt következtetések. Célszerű stratégiák megfogalmazása: outlier eltávolítása, dimenzió csökkentés/növelés, adat csökkentés/növelés stb.

**Az ábrázolás az érme egyik oldala**: ábrázolás + statisztika. Az ábrázolás lehet megtévesztő. Mikor lehet megtévesztő és miért? Átvezetés: Nem szabad hinni mindig a szemünkben. Az ember képes beleesni az optikai illúziók csapdáiban. Ennek érdekében nyújthat segítséget az adatok statisztikai elemzése.

# Adatelemzés lépései és adathalmazok ismertetése

placeholder