



# TDK DOLGOZAT

Elsőéves hallgatók pandémia előtti és alatti bemeneti  
adatainak elemzése modern adattudományi  
eszközökkel

Köller Donát Ákos & Vlaszov Artúr  
BME Matematikus MSc

Témavezető: Szilágyi Brigitta  
Geometria Tanszék



BME Matematika Intézet  
Budapest  
2022

# Tartalomjegyzék

1. Bevezetés	1
2. Fogalmak és definíciók	1
3. A kognitív tesztről	2
4. Adatprepáció	3
4.1. Adatok jellemzése, adattisztítás . . . . .	3
5. Felderítő adatelemzés a két évben	5
5.1. Általános ábrák . . . . .	5
5.2. Folyamatábrák (Sankey-diagramok) . . . . .	5
5.3. Klaszterezés . . . . .	7
6. Prediktív analitika	8
6.1. Modellek és metodológia . . . . .	8
6.2. Osztályozó algoritmusok és optimalizálásuk . . . . .	10
6.2.1. Lineáris regresszió . . . . .	10
6.2.2. Naive Bayes . . . . .	11
6.2.3. Gradient Tree Boosting . . . . .	11
6.2.4. Logisztikus regresszió . . . . .	11
6.2.5. SVM . . . . .	11
7. Modellek kiértékelése	13

## 1. Bevezetés

....

## 2. Fogalmak és definíciók

Ez a szekció csak egy placeholder azon definícióknak, amiket még nem tudtam hova beépíteni

- **One-vs-Rest elv:** Bináris osztályozó algoritmusok esetén használt eljárás, amely többosztályos osztályozási feladat esetén használatos. Lényege, hogy minden osztály esetén képezünk egy új virtuális osztályt, amely az összes többi osztályt tartalmazza, majd minden ilyen osztálypár esetén a bináris osztályozó meghatározza a kérdéses adatrekord esetén az osztályba tartozási valószínűségeket. Végül azt a címkét prediktáljuk a rekordnak, amely osztály esetén a legnagyobb ez a valószínűség.

### **3. A kognitív tesztről**

(ez a rész nem tudom, hogy kell-e)

## 4. Adatprepació

### 4.1. Adatok jellemzése, adattisztítás

Az adatokat az EduBase és Neptun rendszerből lekérve nyertük több adattábla formájában. A táblákból a kutatáshoz használt adatok egyrészt olyan bemeneti adatok, amelyeket a szemeszter első hetéig bezárólag érnek el a hallgatók, másrészt olyan teljesítménymutatók, amelyekről csak a félév végén van információnk (utóbbiak prediktálására építünk modelleket a prediktív analitikai részben). Ezek egy része alapvetően rendelkezésünkre állt a táblákból, másokat a meglévő sorokból és oszlopokból megfelelő *feature engineering* segítségével hoztunk létre. A változók pontos megnevezése és jellege az 1. táblázatban látható.

Emelt érettségi	Bináris változó arra vonatkozóan, hogy a hallgató matematikából emelt érettségit tett-e.
Matematika tagozat	Bináris változó arra vonatkozóan, hogy a hallgató matematika tagozatos volt-e.
Szak	Kategorikus változó, a hallgató szaka a VBK-n.
Matematika 1. blokk	Az elsőéves VBK hallgatók által írt kognitív teszt matematika részén az 1-4. kérdésekre adott helyes válaszok száma.
Matematika 2. blokk	A kognitív teszt matematika részén az 5-10. kérdésekre adott helyes válaszok száma.
Matematika 3. blokk	A kognitív teszt matematika részén a 11-14. kérdésekre adott helyes válaszok száma.
Kognitív eredmény	A kognitív teszt kognitív készségeket mérő részén elért százalékos teljesítmény (0-100-as skálán).
0. ZH pontszám	A BME központi 0. ZH-n elért pontszám.
Tanulmányi pont	A felvételi pontszám tanulmányi pontokból származó része.
Érettségi pont	A felvételi pontszám érettségi pontokból származó része.
Többletpont	A felvételi pontszám többletpontokból származó része.
Matematika A1a	A Matematika A1a tárgyból szerzett érdemjegy.
Kumulált átlag	Az első félév végén megállapított kumulált átlag.

1. táblázat. A vizsgált változók a két évben

A felderítő és modellezési fázisban használt adattáblát a rendelkezésre álló táblák megfelelő mértékű tisztításából, majd ezt követő Neptun-kód alapú összeillesztéséből nyertük. Ezen műveleteket Python-ban valamint R-ben végeztük el.

A kezdeti adattáblák közül a matematika és kognitív eredményeket tartalmazó adatsor

tisztítására volt a legnagyobb szükség, ugyanis az EduBase rendszerben az egyes oszlopokra vonatkozó mezőket a hallgatók töltötték ki, így a kategorikus változók nem voltak rendszerezve. Először egységesítettük a szakmegnevezést ("Vegyészmérnöki", "Biomérnöki", "Környezetmérnöki"), ahol pedig nem volt egyértelmű a szak, azt "UNKNOWN"-ra állítottuk. Ezt követően a teszten elért matematika és kognitív eredményt százalékos formátumról 0-1 közötti tizedestört formátumra változtattuk, valamint létrehoztunk 3 új oszlopot, amely a matematikateszt 3 blokkjában helyesen megválaszolt kérdések számát mutatja. Végül a vizsgálataink szempontjából irreleváns oszlopokat eltávolítottuk, valamint azon oszlopokat is kiszűrtük, amelyek értéke a többiből egyértelműen kikövetkeztethető volt.

A kumulált átlagokat, felvételi pontszámokat illetve 0. ZH eredményeket tartalmazó tábláknál egy-egy hiányos és/vagy anomáliás sor, illetve az irreleváns oszlopok eltávolításán kívül nem volt szükség adattisztításra, a matematika jegyeket tartalmazó tábla pedig minden hallgató minden Matematika A1a tárgyból tett vizsgaalkalmáról és az azon szerzett érdemjegyről tartalmazott adatrekordot, így ezekből meg kellett határozni azt a végső érdemjegyet, amellyel a hallgató a tárgyat elvégezte. A végső összeillesztés során 10-20 fős sorvesztéssel is kellett számolnunk mindkét évben, ugyanis voltak olyan hallgatók, akikről nem minden táblában volt adat (vagy azért, mert nem írtak kognitív tesztet, vagy azért, mert az első félév vége előtt elhagyták a szakot). Az így kapott adattáblák, amelyek a 2019-es és 2021-es évet reprezentálják, rendre 200 és 220 adatrekordot tartalmaztak. ...

## 5. Felderítő adatelemzés a két évben

A felderítő elemzés célja az adatok ábrázolása egyszerűen és gyorsan, annak érdekében, hogy az adattípusok egyszerű viszonyai szemléletesen áttekinthetők legyenek. Erre a célra alkalmasak a oszlopdiagramok, szórásdiagramok, folyamatábrák és más grafikonok.

### 5.1. Általános ábrák

### 5.2. Folyamatábrák (Sankey-diagramok)

A Sankey-diagramok olyan folyamatábrák, amelyekben az ábrázolt folyamatok szélessége arányos a megfelelő folyamértékekkel. Ezáltal könnyen ábrázolhatóak adott állapotok közötti vándorlások, illetve ezek egymással vett aránya. Egy hátránya van, hogy egy ábrán csak két osztálycsoport között olvashatjuk le az áramlásokat. Ha egy harmadik osztálycsoportot is belehelyezünk, és csak a másodikkal kötjük össze, akkor az első és a harmadik közötti vándorlások nem jelennek meg az ábrán. Emiatt lesz némi redundancia az alább ismertetett ábrákban.

Dolgozatunkban hat opciót vizsgáltunk a két évben az elérhető adatoknak megfelelően. Mindkét esetben az állapotok a hallgatók valamely eredményét jelentették, például felvételi pontszámot.

- 2019:
  1. matematikai teszt 1.,2. és 3. részeiben helyesen megoldott feladatok száma.
  2. kognitív teszt eredmény -  $A1$  jegy -  $A2$  jegy.
- 2019 és 2021:
  1. tanulmányi hozott pontszám - érettségi pontszám -  $A1$  jegy.
  2. felvételi pontszám - nulladik ZH - kognitív teszt eredmény.
  3. felvételi pontszám - kognitív teszt eredmény -  $A1$  jegy - kumulált átlag.
  4. felvételi pontszám - kognitív teszt eredmény - első féléves átlag - kumulált átlag.

Az első két esetben csak a 2019-es évfolyamról volt adat. Az utolsó esetben 2021-ben megegyezett a kutatás idejében az első féléves és a kumulált átlag, így ennél az utóbbi értelemszerűen nem lett még egyszer beletéve.

A matematikai teszt részeinek eredményein és az  $A1$ ,  $A2$  jegyeken kívül a többi változó folytonos volt, így szükség volt ezek diszkretizálására. Öt-öt osztály lett létrehozva minden

esetben, de nem feltétlenül került mindegyikbe hallgató. Az átlagok kerekítve lettek, a felvételi, kognitív teszt és nulladik zh pontszámok ekvidisztáns módon lettek felosztva.

A *felvételi pontszám - nulladik ZH - kognitív teszt eredmény* változatban a nulladik zh és a teszt eredményei nullától az elérhető pontszámig lettek felosztva 20%-os lépésközökkel. A többi változat a kutatás későbbi szakaszában készült, így a két mutató a terjedelmük szerint lett felosztva, nem az elérhető pontszám alapján.

Ezután *for ciklus* használatával elkészültek a tranzíciós mátrixok. Egy ilyen mátrix  $i$ -edik sorának  $j$ -edik eleme azt mutatta, hogy az egyik változó  $i$ -edik osztályában hány olyan hallgató volt, aki a másik változó  $j$ -edik osztályába került. Ez a folyamatok szélességének megadásához volt szükséges.

A *plotly.graph\_objects* könyvtár *go.Sankey* függvényével lettek elkészítve egyenként az ábrák. Minden esetben meg kellett adni számozva a kiinduló és beérkező állapotokat, így a középső csoportok mindkét listában szerepeltek. A program index alapján kapcsolta össze a két listából az állapotokat, így kellett egy harmadik lista, amelynek a megfelelő indexű eleme a két állapot közötti folyam mérete. Ezen kívül a folyamatokhoz és magukhoz az állapotokhoz is színeket kellett még rendelni, ismét egy-egy listában, figyelve az indexeket.

Végül a kirajzolódott ábrát kellett kézzel igazítani, ugyanis az eredmények csoportjai nem feltétlenül álltak megfelelő sorrendben a függőleges tengelyen, illetve azokban az esetekben, amikor nem volt mind az öt osztályban hallgató, egy-egy csoport átcsúszott egy másik eredmény oszlopába. Ezek nem okoztak gondot, ugyanis a csoport mozgásával a hozzátartozó sávok automatikusan mozogtak együtt.

A 2021-es gólyák a kutatás idejében még nem vehették fel az *A2* tárgyat, így értelem-szerűen csak a 2019-es évfolyamra készült ez az ábra.

Az ábrán rögtön látható, hogy a kognitív tesztben mindenki 20% felett teljesített, de nagyon kevés hallgatónak sikerült 80%-nál magasabb eredményt elérni. Legtöbbször 40% és 60% közötti pontszámot szereztek, a második legnagyobb osztály a 60-80%-os.

Ehhez képest az *A1* tárgyon elért jegyek eloszlása egyenletesebb. Néhány hallgatónak nem sikerült elvégezni a tárgyat, nagyobb részük kettést vagy hármaszt szerzett, és körülbelül a harmaduk 45/55 arányban négyest és ötöst szereztek.

A folyamatokat illetően, azok, akiknek nem sikerült a tárgy, a kettes és a hármas tesztosztályban voltak, azaz közülük valakinek nem sikerült 40%-ot elérni, de egyikőjük sem teljesített 60%-nál jobban. A tesztet legjobban megírok pedig többségben ötöst vagy négyest szereztek, azonban a négyes tesztosztályból több hallgató is csak hármaszt tudott szerezni. A hármas osztályból minden jegyhez megy folyam, a legtöbb ketteshez és hármaszhoz, illetve körülbelül a negyede egyenletesen oszlik el a négyes és ötös között. Egy érdekes eset van, a kettes tesztosztályból valakinek sikerült ötöst szerezni a tárgyból. Ennek ellenére ebből az osztályból a hallgatók kétharmada legfeljebb kettést ért el.

Ezekből sejthető, hogy egy hallgató teljesítménye a teszt során összefügg a további teljesítményével, de lehet jelentős romlás és javulás is.

Az *A2* tárgy jegyeinek eloszlása ismét más, több ötös és négyes, kevesebb hármas és körülbelül ugyanannyi kettes lett, és egy hallgatónak nem sikerült elvégezni. Érdekes, hogy az első körben kettest szerző hallgatók majdnem harmada javított négyesre vagy ötösre, a négyest szerző hallgatóknak pedig majdnem a fele kettesre rontotta. Hármából is sokan javítottak, körülbelül a negyedük továbbra is hármast szerzett, viszont több, mint harmaduk rontott. Ötöst szerzőkből hatan kettesre rontottak (kb. 13%).

### **5.3. Klaszterezés**



## 6. Prediktív analitika

### 6.1. Modellek és metodológia

Következő lépésként azt vizsgáltuk, hogy a két évben külön a bemeneti adatok alapján mennyire pontosan lehet előrejelezni egyes teljesítménymutatók értéket, illetve hogy a predikciókra nézve a különböző bemeneti változók, más néven *attribútumok* milyen és mekkora szereppel bírnak. Ehhez többféle modellen többféle *gépi tanuláson alapuló osztályozó algoritmus* használatára volt szükség. A kutatás során kétféle eredmény alaposabb vizsgálatára összpontosítottunk mindkét évben: a "Matematika A1a - Analízis" tárgyból kapott érdemjegyre illetve az első félév végén megállapított kumulált tanulmányi átlagra. Ezen feladatok a felügyelt gépi tanulási problémák közé esnek, ahol a kitüntetett célváltozót szeretnénk a többi bemeneti változóval előrejelezni. Ekkor a gépi tanulás során a teljes adathalmazt tanító és teszhalmazra bontjuk, a tanító adathalmazon betanítjuk az algoritmusokat úgy, hogy a tanítóhalmazbeli adatok célváltozóját minél pontosabban prediktálják, majd valamilyen metrika szerint kiválasztjuk a betanított algoritmusok közül a legjobbat. A leghatékosabb osztályozó kiválasztásához többnyire *keresztvalidációt* alkalmazunk. Keresztvalidáció során a tanítóhalmazt felbontjuk  $K$  egyenlő részre, melyek közül az egyiket kinevezzük validációs halmaznak. Ezt követően az algoritmusokat betanítjuk a maradék  $K-1$  részen, amelyeket aztán a validációs halmazon kiértékelünk. Ezt összesen  $K$  alkalommal ismételjük, mindig másik részt választva validációs halmaznak, majd azt az algoritmust dedikáljuk a legjobbnak, amelynek az aggregált teljesítménye a  $K$  darab iteráció során a legjobb. Az így kapott osztályozót még visszamérjük a teszhalmazon is az általánosítóképesség ellenőrzése végett. A kutatás során a cél a teszhalmazon való minél hatásosabb előrejelzés mellett az egyes éveknél használt és optimalizált algoritmusok esetén a különböző attribútumok prediktív erejének összehasonlítása volt.

A két vizsgált probléma közül az előbbi alapvetően egy osztályozási probléma öt osztállyal, míg az utóbbi egy regressziós feladat. Az előbbi feladathoz ötféle algoritmust, *Gradient Tree Boosting*-ot, *Naive Bayes*-t, *logisztikus regressziót*, *SVM*-et és *lineáris regressziót* használtunk, az utóbbihoz csupán lineáris regressziót (ezen algoritmusok pontos működése és optimalizálása a következő fejezetben lesz ismertetve). Azért esett a választás ezen algoritmusokra, ugyanis számos egyetemi teljesítményt és lemorzsolódást vizsgáló tanulmány során teljesítettek kiváló eredménnyel. (IDE REFERENCIÁK) Mivel azonban viszonylag kevés adat állt a rendelkezésünkre, ezért az érdemjegyek prediktálásánál az adatrekordok esetén a pontos érdemjegy helyett érdemjegy csoportok előrejelzésére koncentráltunk. A matematika érdemjegycsoportok prediktálására így kétféle modell került felvázolásra: egy *3 csoport modell*, illetve egy *2 csoport modell*.

A 2 csoport modell esetén a két csoportot a  $\{5,4,3\}$  illetve  $\{2,1\}$  osztályok adják, míg a 3 csoport modellnél az osztályok  $\{5,4\}$ ,  $\{3,2\}$  illetve  $\{1\}$  módon alakultak. Az előbbi

esetben az osztályok intuitívan a lemorzsolódási veszélyeztetettség szerint formálódtak, míg az utóbbiban egy általánosabb "jól teljesítő", "rosszul teljesítő", "lemorzsolódott" csoporthármast kívántunk elérni. Mindkét modell esetén külön vizsgáltuk a teljesítményt összes hallgatóra vonatkozóan illetve szétbontva vegyész-mérnök és biomérnök hallgatókra egyaránt (a környezetmérnökökről nem állt rendelkezésre elég adat, így őket a szakonkénti bontásban kihagytuk).

Az egyes osztályozó algoritmusok közül a Naive Bayes és Gradient Tree Boosting algoritmusok képesek kezelni a többosztályos feladatokat, a lineáris regresszió azonban folytonos célváltozóérték prediktáláshoz használható, így ott a prediktált értéket kerekítettük a legközelebbi címkeértékhez, a logisztikus regresszió és az SVM pedig alapjáraton csak bináris osztályozásra alkalmas, így a náluk *One-vs-Rest* elvű osztályozást használtunk. Az elv lényege, hogy minden osztály esetén képezünk egy új virtuális osztályt, amely az összes többi osztályt tartalmazza, majd minden ilyen osztálypár esetén a bináris osztályozó meghatározza a kérdéses adatrekord esetén az osztályba tartozási valószínűségeket. Végül azt a címkét prediktáljuk a rekordnak, amely osztály esetén a legnagyobb ez a valószínűség.

Az egyes modellek és almodellek esetén a tanítás előtt főkomponens analízist (Principal Component Analysis) is alkalmaztunk. Az eljárás lényege, hogy az adatpontokat egy kisebb dimenziós térre vetítjük le oly módon, hogy a változók közötti variancia minél nagyobb részét tartsák meg, így minimalizálva az információvesztést. Az új változók, amelyeket főkomponensnek nevezünk, a kisebb dimenziós térben az eredeti változók kovarianciamátrixának sajátvektorai lesznek. PCA használata során az algoritmusok gyorsabban tanulnak a kisebb dimenziószám miatt, és olykor jobb eredményt is érnek el. Jelen kutatásban másrészt azért is döntöttünk a PCA alkalmazása mellett, mert a korábbi, ugyanezen problémakört vizsgáló, általunk átnézett tanulmányok nem használtak PCA-t még nagyobb attribútum szám esetén sem, ugyanakkor mi a korai tanító-tesztelési fázisnál azt tapasztaltuk, hogy az áttanszformált adatokon jobban teljesítenek az osztályozó algoritmusok, így potenciálisan megéri alkalmazni.

## Implementálás

Valamennyi modellnél a teljes adathalmazt felbontottuk tanító- és teszhalmazra 70-30 arányban, majd a tanítóhalmazra illesztett PCA modellt alkalmaztuk a teszhalmazra is, ahol a főkomponensek számát minden modellnél 2 és 8 között iterálva változtattuk 2-es lépésközzel. Ezt követően a változókat kvantilis alapú 0-1 skálázásnak vetettük alá, ahol az algoritmusok jobb teljesítőképessége és a változók kiértékelés utáni összehasonlíthatósága végett a különböző attribútumok értékeit a kvantilisok mentén a  $[0, 1]$  intervallumba transzformáltuk át, valamint az érdemjegy csoportok prediktálásához a 2 és 3 csoport modelleknél a célváltozó-értékeket rendre 1, 0-ra illetve 2, 1, 0-ra módosítottuk. Az al-

goritmusok hiperparamétereinek optimális megválasztására *5-szörös keresztvalidációt* alkalmaztunk. A keresztvalidációnál használt, jóságot mérő metrikának a *kiegyensúlyozott pontosságot* (Balanced Accuracy) választottuk, amelynek képlete az alábbi:

$$\text{balanced-accuracy} = \frac{1}{2} \left( \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right)$$

Ezen metrika alapvetően bináris, pozitív-negatív osztályú osztályozási problémákhoz alkalmas, de többosztályos osztályozás esetén is használható, ahol az egyes osztályokhoz tartozó  $TP$ ,  $FP$ ,  $FN$ ,  $TN$  értékek által kiszámított kiegyensúlyozott pontosságok számtani közepét nézzük. A képlet egyes jelölései 2 valamint több osztály esetén:

- **TP** a pozitívnek (osztálybelinek) osztályozott, valóban pozitív (osztálybeli) adatrekordok száma
- **FP** a pozitívnek (osztálybelinek) osztályozott de valójában negatív (nem osztálybeli) adatrekordok száma
- **FN** a negatívnek (nem osztálybelinek) osztályozott de valójában pozitív (osztálybeli) adatrekordok száma
- **TN** a negatív (nem osztálybelinek) osztályozott, valóban negatív (nem osztálybeli) adatrekordok száma

A választott metrika mellett meghatároztuk a legjobb algoritmusokat, amelyeket a teszhalmazon visszamértünk, valamint kiértékeltek az így módon választott modellek esetén az egyes változók fontosságát is. A pontos modellezési struktúrát a ?? ábra mutatja.

## 6.2. Osztályozó algoritmusok és optimalizálásuk

(ez a szekció még félkész)

### 6.2.1. Lineáris regresszió

A lineáris regresszió jellegéből adódóan alapvetően folytonos célváltozó prediktálására alkalmas, ugyanakkor kategorikus, ordinális címkéjű adatok osztályozására is fel lehet használni. Az algoritmus lényege, hogy a magyarázóváltozók mindegyikéhez egy-egy súlyt rendelünk, majd az adatpont attribútumértékeinek vesszük a súlyokkal való súlyozott összegét, és esetleg egy bias tagot hozzáadva az így kapott szumma lesz a prediktált címkeérték. A cél a tanítás során a súlyok optimális megtalálása, miközben a tanítóhalmazon minimalizáljuk a predikciós hibát.

### 6.2.2. Naive Bayes

A Naive Bayes algoritmus működése mögött álló alapelv az, hogy feltesszük az attribútumok feltételes függetlenségét, amennyiben a célváltozó értéke ismert. Osztályozás során azt vizsgáljuk, hogy mely címkeérték mellett a legnagyobb a valószínűsége annak, hogy az adott adatrekord attribútumai éppen a felvett értékeket kapták. A megfelelő valószínűségeket a tanítóhalmazbeli adatok attribútumértékeinek különböző címkék melletti relatív gyakoriságai adják.

### 6.2.3. Gradient Tree Boosting

A Gradient Boosting algoritmus egy Ensemble típusú osztályozó, amelyek lényege, hogy sok gyenge teljesítményű prediktor (*weak learner*) eredményét felhasználva hoz egy erős predikciót. Gradient Tree Boosting esetén a gyenge prediktorok *döntési fák*, amelyek lefelé irányított, legtöbbször bináris fák, amelyeknek minden belső csúcsában egy attribútumra vonatkozó feltétel szerepel, a levelei pedig valamilyen célváltozóértékkel címkézettek.

A keresztvalidálás során optimalizált paraméterek:

- Tanulórata (0.01 és 5 között változtatva 0.1-es lépésközzel)
- Boosting fázisok száma (5 és 100 között változtatva 5-ös lépésközzel)
- Vágási feltétel (négyzetes hiba, Friedman MSE)

### 6.2.4. Logisztikus regresszió

A logisztikus regresszió alapvetően bináris osztályozási problémák megoldására alkalmas, de kiterjeszthető többosztályos feladatok megoldására is. Lényege, hogy a lineáris regresszióhoz hasonlóan az adatrekord attribútumértékeinek súlyozott összegét használjuk egy szigmoid<sup>1</sup> függvény bemeneteként, amely azt utána leképzi az (0, 1) intervallumra, és amennyiben az output 0.5-nél nagyobb, úgy a pozitív osztályba soroljuk az adott rekordot. Többosztályos osztályozás esetén

A keresztvalidálás során optimalizált paraméterek:

- Regularizációs paraméter (0.05 és 5 között változtatva 0.05-ös lépésközzel)
- Önoptimalizálási módszer ("SAG", "SAGA")

### 6.2.5. SVM

Az SVM (Support-Vector-Machine) egy lineáris szeparálást használó bináris osztályozó algoritmus, amely egyéb problémákra is kiterjeszthető. Lényege, hogy különböző mag-

---

<sup>1</sup>A szigmoid függvény:  $\sigma(z) = \frac{1}{e^{-z} + 1}$ , ahol  $z = x_1w_1 + x_2w_2 + \dots + x_nw_n + b$  a súlyozott összeg.

függvénye segítségével az adatrekordokat egy magasabb dimenziós térbe képi le, ahol olyan szeparáló hipersíkot keres, amely maximalizálja a vele párhuzamos, adatpontot nem tartalmazó térrészt. A cél a megfelelő magfüggvény és a szeparáló hipersík megtalálása, amellyel a különböző címkéjű adatpontok lineárisan szeparálhatóak. ...

A keresztvalidálás során optimalizált paraméterek:

- Regularizációs paraméter (0.1 és 5 között változtatva 0.1-es lépésközzel)
- Magfüggvény (lineáris, legfeljebb 3-ad fokú polinomiális, RBF)

## 7. Modellek kiértékelése

		Osztályozó algoritmusok				
		Grad. Boost.	Naive B.	Log. reg.	SVM	Lin. reg.
3 csoport	Összesített	2 PC	66.67	62.67	52.00	61.33
		4 PC	<b>70.67</b>	60.00	57.33	65.33
		6 PC	65.33	62.67	54.67	64.00
		8 PC	64.00	68.00	53.33	64.00
	Szakonként	2 PC	78.71	80.36	73.85	80.36
		4 PC	75.41	80.36	47.80	78.71
		6 PC	<b>82.02</b>	80.36	75.47	82.02
		8 PC	80.36	78.71	37.42	78.71
2 csoport	Összesített	2 PC	59.42	69.57	69.57	69.57
		4 PC	59.42	71.01	<b>73.91</b>	73.91
		6 PC	63.77	69.57	73.91	73.91
		8 PC	68.12	69.57	71.01	72.46
	Szakonként	2 PC	67.31	67.31	65.69	64.03
		4 PC	64.00	64.00	67.31	63.97
		6 PC	64.03	65.69	64.07	67.27
		8 PC	64.07	62.51	57.49	<b>68.96</b>

2. táblázat. A 2019-es adatsor eredményei

A 2. táblázatban a 2019-es adatokra optimalizált modellek teljesítménye látható, ahol az első három oszlop a modellt és a használt főkomponensek (PC) számát mutatja, míg a jobb oldali öt oszlop az optimalizált algoritmusok teljesítményét szemlélteti. A szakonkénti bontásban szereplő értékek a biomérnöki és vegyészmérnöki adatokon kapott értékek súlyozott átlagai. A 3 csoport modellek esetén a Gradient Boosting algoritmus, míg a 2 csoport modellek esetén a regressziós algoritmusok teljesítettek a legjobban, ugyanakkor a 2 csoport modellek teljesítménye többségében alulmúlja a 3 csoport modellekét. Mivel a 2 csoport modellben az osztályok eloszlása kiegyensúlyozottabb, ez arra enged minket következtetni, hogy a 2-es és 3-as érdemjegyet szerzők között a bemeneti adatok tekintetében nincsenek nagy különbségek.

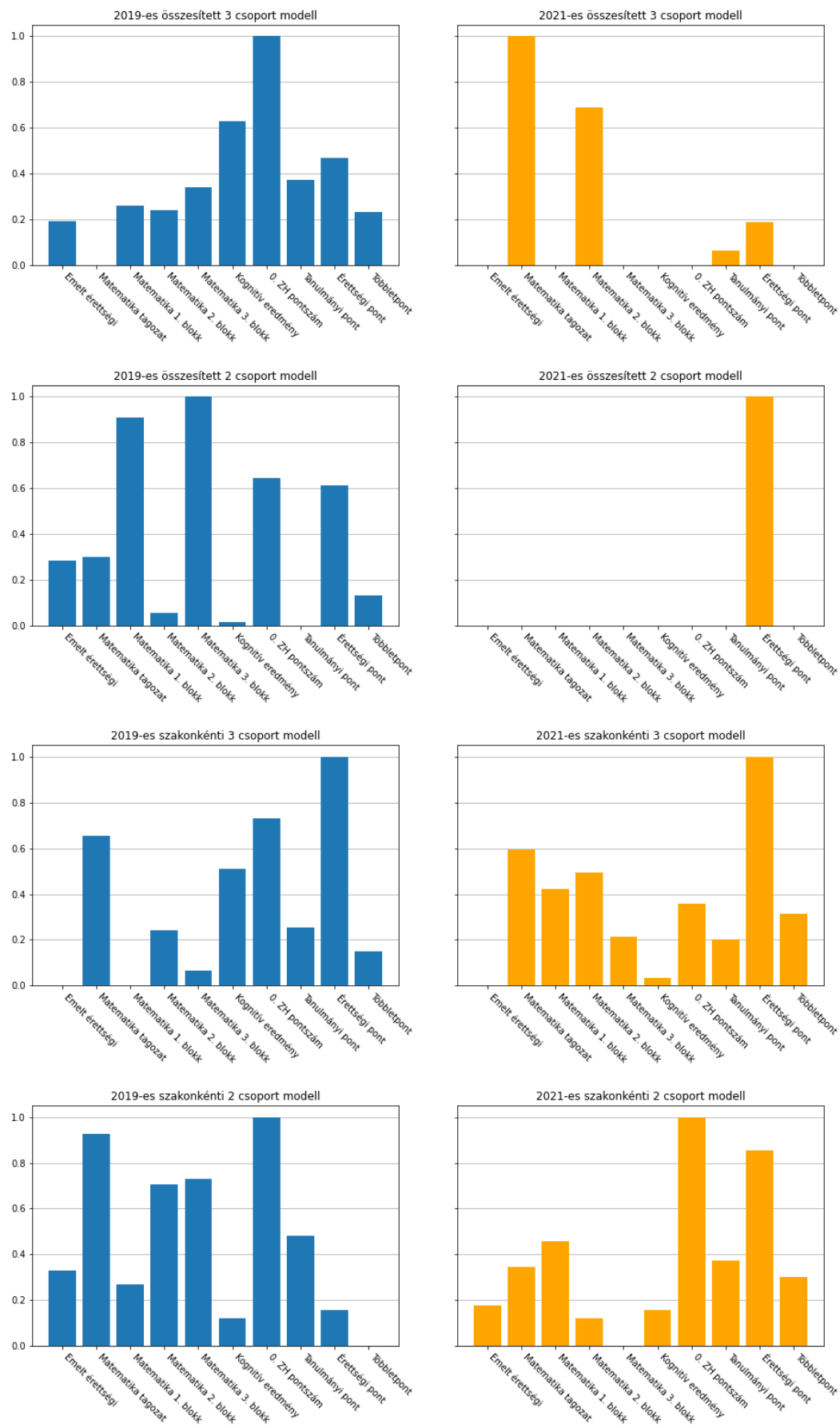
A 3. táblázat a 2021-es adatokon optimalizált algoritmusok eredményét szemlélteti az előző ábráival megegyező metodológia szerint. Ezen adathalmazon az osztályozók teljesítménye jobbnak nevezhető, mint a 2019-es adatsoron, átlagosan a legeredményesebb algoritmusnak a Naive Bayes nevezhető, amely a szakonkénti bontású 3 csoport modellen ért el minden főkomponensszám mellett 80% feletti teljesítményt, ugyanakkor a 2 csoport modell esetén a regressziós és SVM algoritmusok is 80% közeli vagy afölötti eredményt

érték el. A 2019-es eredményekkel ellentétben a 2021-es adatokon a 2 csoport modellek értékei jobbak, mint a 3 csoport modelleké, ugyanakkor nem szignifikánsan.

		Osztályozó algoritmusok					
		Grad.	Boos.	Naive B.	Log. reg.	SVM	Lin. reg.
3 csoport	Összesített	2 PC	53.74	72.97	67.75	<b>73.30</b>	56.80
		4 PC	61.86	70.49	65.82	66.63	51.91
		6 PC	68.27	68.53	67.73	70.80	64.00
		8 PC	66.31	70.20	67.21	67.17	64.00
	Szakonként	2 PC	64.84	81.89	59.23	64.77	49.35
		4 PC	55.55	<b>83.07</b>	66.69	66.66	67.91
		6 PC	66.69	82.85	66.65	53.67	53.31
		8 PC	59.26	82.96	68.51	62.95	55.53
2 csoport	Összesített	2 PC	69.45	67.89	69.77	69.77	67.89
		4 PC	72.74	72.90	78.23	76.51	77.91
		6 PC	67.89	72.90	79.80	79.80	76.19
		8 PC	68.05	74.62	79.96	<b>83.24</b>	79.63
	Szakonként	2 PC	60.23	77.41	72.87	73.01	79.68
		4 PC	63.92	74.86	72.59	69.60	78.27
		6 PC	67.61	77.84	81.08	76.42	<b>81.53</b>
		8 PC	68.32	77.84	77.41	69.75	79.68

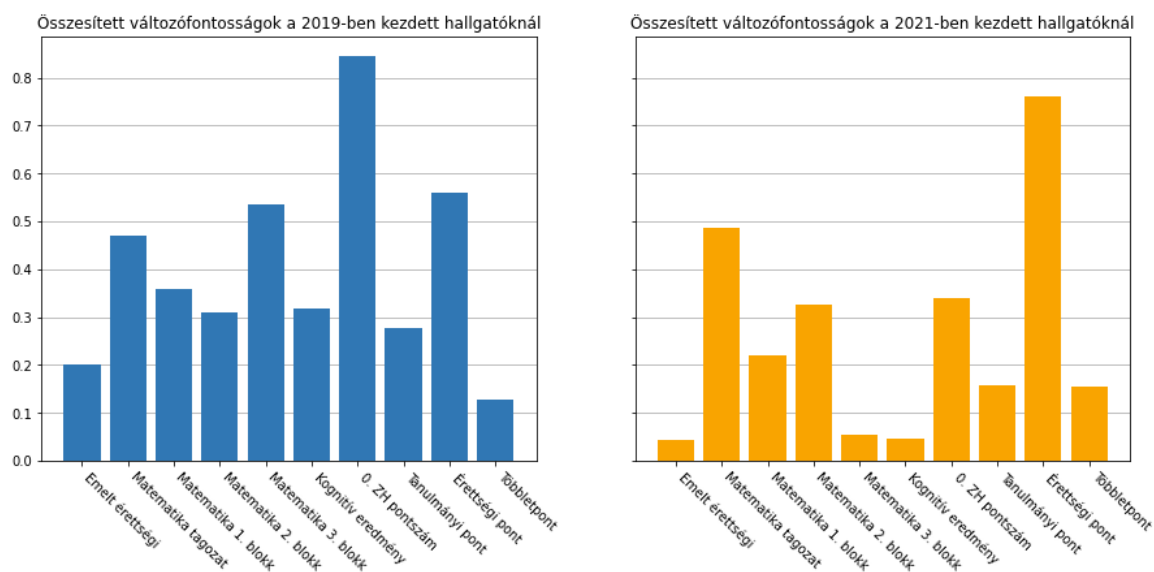
3. táblázat. A 2021-es adatsor eredményei

A két év négy modelljén külön-külön legjobban teljesítő algoritmusok esetén az egyes változók fontosságának egymáshoz viszonyított arányait a 1. ábra, míg az évenként összesített és leátlagolt változófontosságokat a 2. ábra mutatja. A diagramokon szereplő értékek az egyes algoritmusoknál megállapított változófontosságok *min-max* skálázott értékei, amelyeket regressziós algoritmusoknál az attribútumokhoz rendelt súlyokból, a többi osztályozónál pedig az Sklearn '*inspection*' csomagjának segítségével nyertünk ki. Mindkét évben többnyire ugyanazok a változók a legdominánsabbak: 0.ZH pontszám, érettségi pont és a matematika tagozat, ugyanakkor szignifikanciájuk a két évben merőben eltérő. A másik fontos észrevétel a többi változó fontossága a dominánsokhoz képest. Míg 2019-ben a többi változó közepes mértékű fontossággal bír, addig 2021-ben ezek a változók igencsak csekély szignifikanciával rendelkeznek.



1. ábra. Az attribútumok prediktív ereje a legjobban teljesítő algoritmusoknál





2. ábra. Az változók összesített fontossága a két évben

A kumulált átlag prediktálására vonatkozó  $R^2$  statisztikát, reziduális szórásnégyzetet és változófontosságot rendre a ...táblázat, ...ábra és ...ábra szemlélteti.

## Hivatkozások