



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. A kognitív tesztről	1
3. Adatprepáció	2
3.1. Adatok jellemzése, adattisztítás	2
4. Felderítő adatelemzés a két évben	4
4.1. Általános ábrák	4
4.2. Folyamatábrák (Sankey-diagramok)	9
4.3. Klaszterezés	16
4.3.1. K-közép algoritmus működése	17
4.3.2. DBSCAN algoritmus működése	17
4.3.3. Ward féle hierarchikus klaszterező algoritmus működése	18
4.3.4. K-közép algoritmus eredménye	18
4.3.5. DBSCAN eredménye	21
4.3.6. Ward féle algoritmus eredménye	21
5. Prediktív analitika	25
5.1. Modellek és metodológia	25
5.2. Osztályozó algoritmusok és optimalizálásuk	28
5.2.1. Lineáris regresszió	28
5.2.2. Naive Bayes	29
5.2.3. Gradient Tree Boosting	29
5.2.4. Logisztikus regresszió	29
5.2.5. SVM	30
6. Modellek kiértékelése	31
7. Diszkusszió, következtetések	37
8. Összefoglalás	37

1. Bevezetés

....

2. A kognitív tesztről

(ez a rész nem tudom, hogy kell-e)

3. Adatprepació

3.1. 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 és/vagy természettudomány 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-ján 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 oszlo-

pokra vonatkozó mezőket a hallgatók töltötték ki, így a kategorikus változók nem voltak rendszerezve. Először egységesítettük a szakmegnevezést ("Vegyésmérnöki", "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 230 és 202 adatrekordot tartalmaztak, melyekben a vegyésmérnök, biomérnök és környezetmérnök hallgatók száma rendre 119,81,28 illetve 104,71,21 volt.

4. Felderítő adatelemzés a két évben

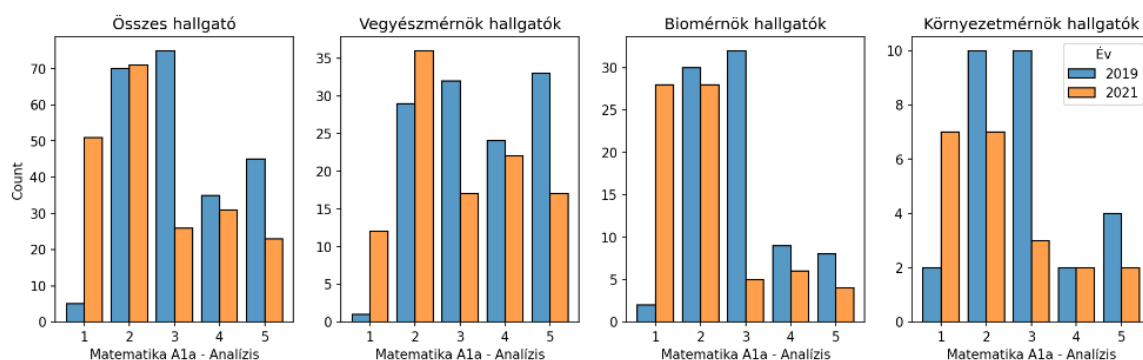
Egy részben feltáró adatelemzés fontos szakasza a felderítő elemzés. Célja az adatok ábrázolása egyszerűen és gyorsan, annak érdekében, hogy az adattípusok egyszerű viszonyai szemléletesen áttekinthetőek legyenek. Erre a célra alkalmasak a oszlopdiagramok, szórásvázlatok, folyamatábrák és más grafikonok, továbbá klaszterezéssel kevésbé triviális összefüggések is kinyerhetők az adatból.

4.1. Általános ábrák

Ebben az alfejezetben tekintjük át a számos készült ábrából a legérdekesebbeket. Viszonylag sok változóval kellett dolgozni, így célszerű volt több részabrást egységekbe gyűjteni a hasonló felépítésű grafikonokat. A megfelelő ábrák elkészítéséhez a Python *matplotlib* és *seaborn* csomagjait használtuk. A változók értékeinek eloszlását oszlopdiagramokkal, a változópárok egymás közötti viszonyait szórásvázlatokkal szemléltettük.

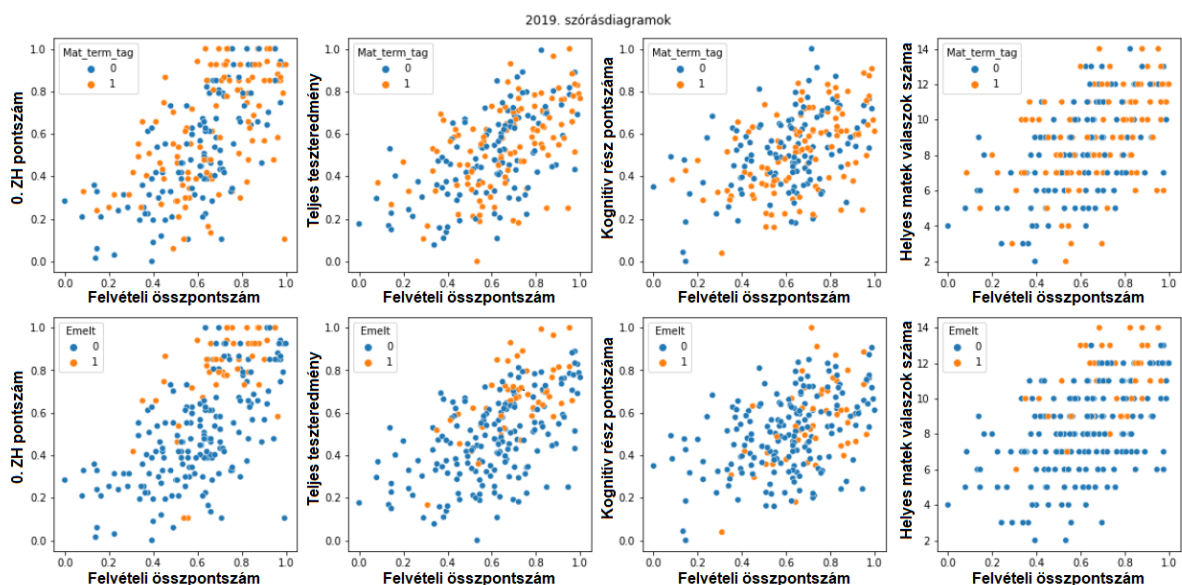
A *seaborn pairplot* függvény előnye, hogy csak az általunk vizsgált változók oszlopait kell megadni, és az összes kombinációt egyszerre számítja ki és ábrázolja egy megfelelő méretű mátrixban. Először tekintsük a *matplotlib figure* alakzatokba összesített ábrákat. Ezeknek az az előnye a *seaborn pairplot*-tal szemben, hogy tetszőleges ábrákat lehet belehelyezni, így például színezéssel bevihető egy harmadik dimenzió a szórásvázlatba. Sőt negyedik dimenziót is lehetne ábrázolni a pontok kinézetének módosításával, de az már nehezen értelmezhető.

Az 1. ábrán az két évben szerzett Matematika A1a -Analízis jegyek eloszlása látható összesítve és szakokra lebontva.



1. ábra. Az elsőéves matematikajegyek eloszlása az egyes években

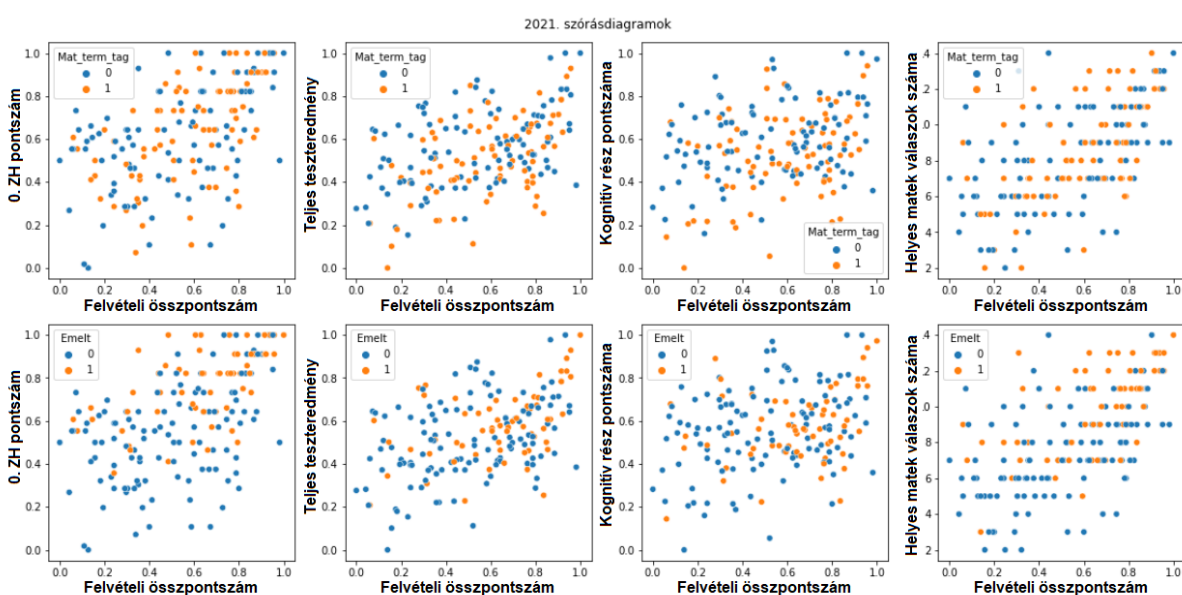
A legszembetűnőbb különbség a két év között az elégtelen, elégséges és közepes érdemjegyek eloszlásának tekintetében van. Az összesített ábrán jól látható, hogy míg 2019-ben a 2-es és 3-as érdemjegyek vannak túlsúlyban, addig 2021-ben az 1-es és 2-es érdemjegyek kerültek többségbe. Ugyanez a jelenség figyelhető meg a biomérnök és környezetmérnök hallgatók esetén is.



2. ábra. 2019. Szórásdiagramok

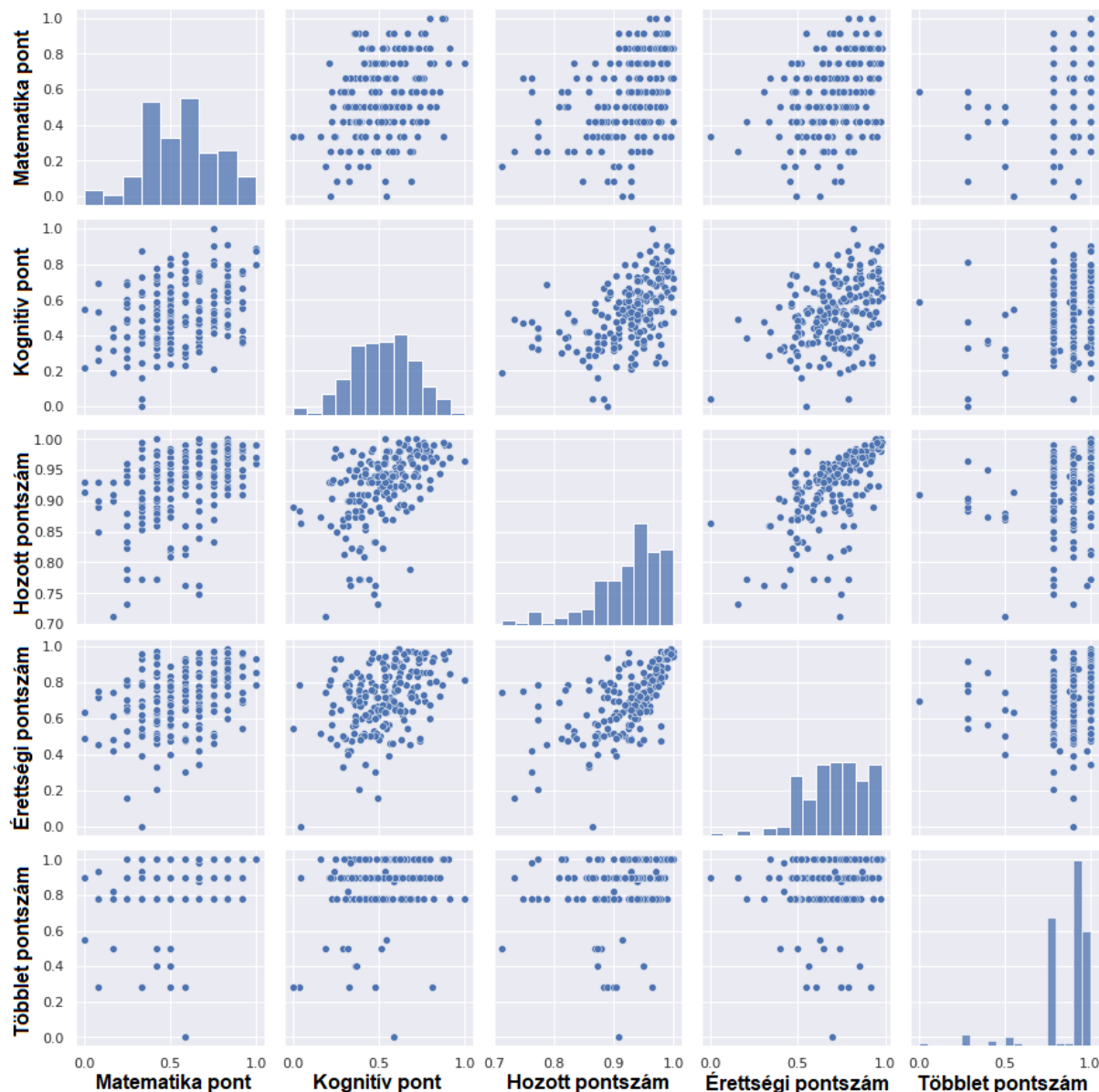
A 2. ábrán egyrészt a 0. ZH, a teszt összpontszáma és a két külön rész pontszámai vannak ábrázolva pontfelhőként a felvételi összpontszám függvényében, de minden eset kétszer. A felső sorban a pontok az alapján vannak színezve, hogy járt-e természettudományi tagozatra a hallgató (1 igen, 0 nem), az alsóban pedig, hogy emelt szinten érettségizett-e (1 igen, 0 nem).

Egyrészt rögtön látszik, hogy minden ábrán a bal felső és a jobb alsó sarkokban többnyire nincs pont, ez a pozitív korreláció, azaz összefüggés jele. Másrészt a színezésből arra következtethetünk, hogy az emelt érettségi megléte erősen fölfelé húzza a többi eredményt, míg a természettudományi tagozatosok esetében nem egyértelmű a helyzet, észrevehető elkülönülés nincs.



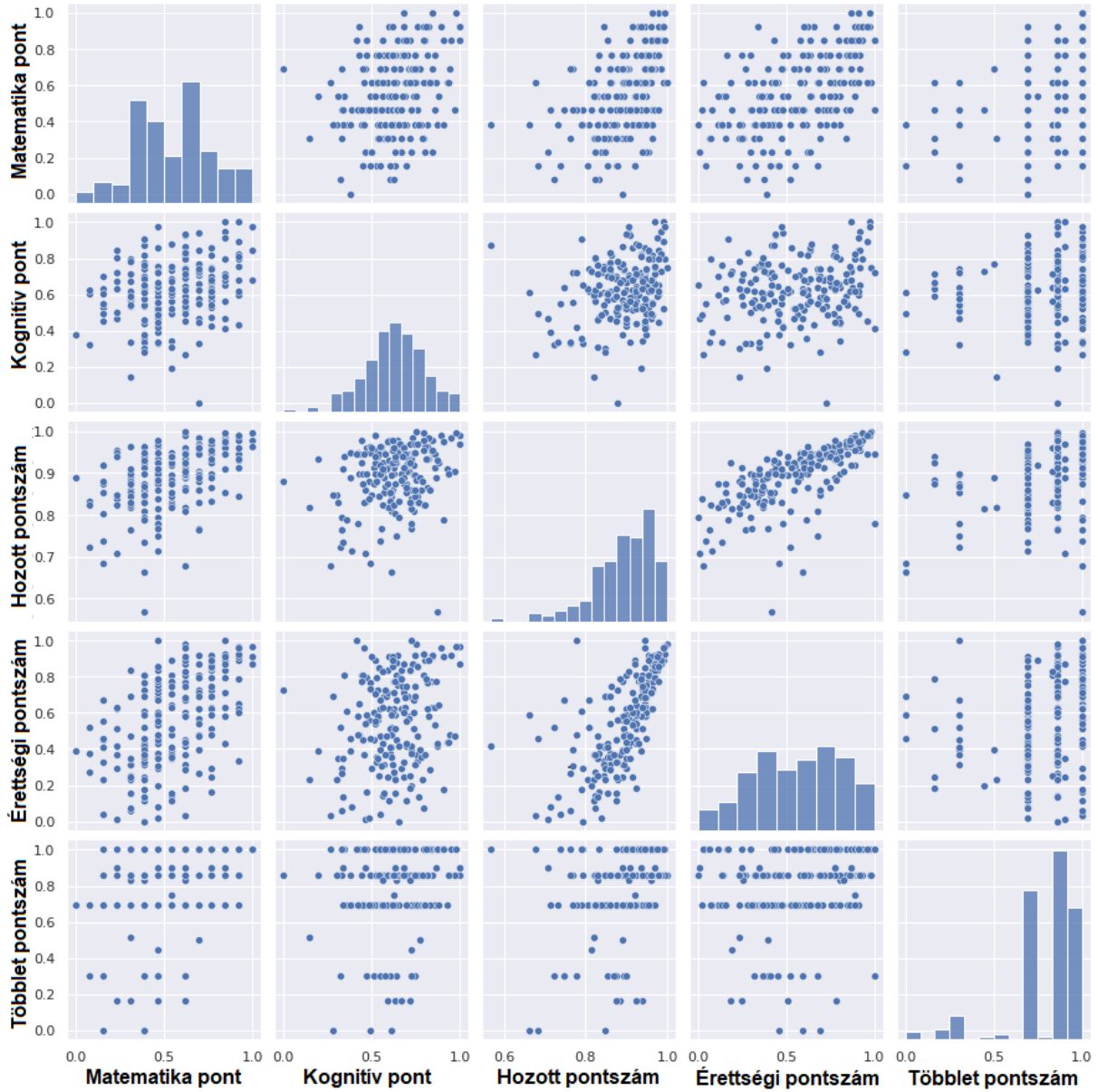
3. ábra. 2021. Szórásdiagramok

A 2019-es ábrához képest az mondható, hogy nőtt a változók szórása, ugyanis a pontfelhők jobban szétterülnek. A természettudományi tagozaton való tanulás továbbra sem fejtett ki hatást a többi eredményre. Az emelt szintű érettségi megletének hatása viszont 2019-hez képest csökkent, ez a teljes teszteredménynél a legszemléletesebb.



4. ábra. 2019. Teszt és felvételi pontszám részei

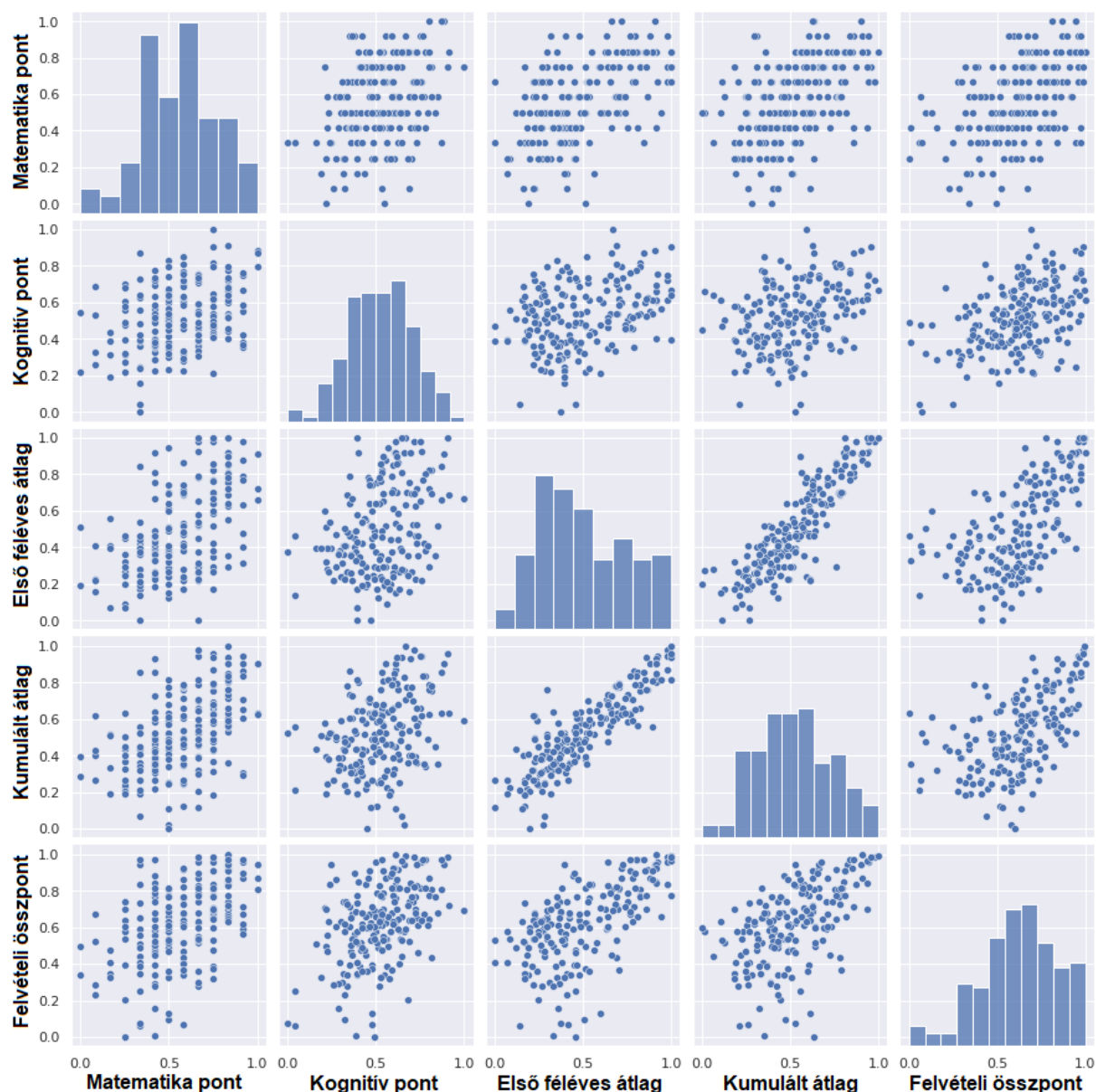
Ezen az ábrán látható a *seaborn pairplot* függvény egyik eredménye. Ebben az esetben volt két hallgató, aki pont duplázással számította a teljes felvételi pontszámát, így a hozott pontszámánál 0 állt értékként. Ez természetesen torzítást okozott a megfelelő részábrákon, így ki lettek véve. A változók itt min-max skálázva vannak, így minden tengelyen 0-tól 1-ig vannak értékek.



5. ábra. 2021. Teszt és felvételi pontszám részei

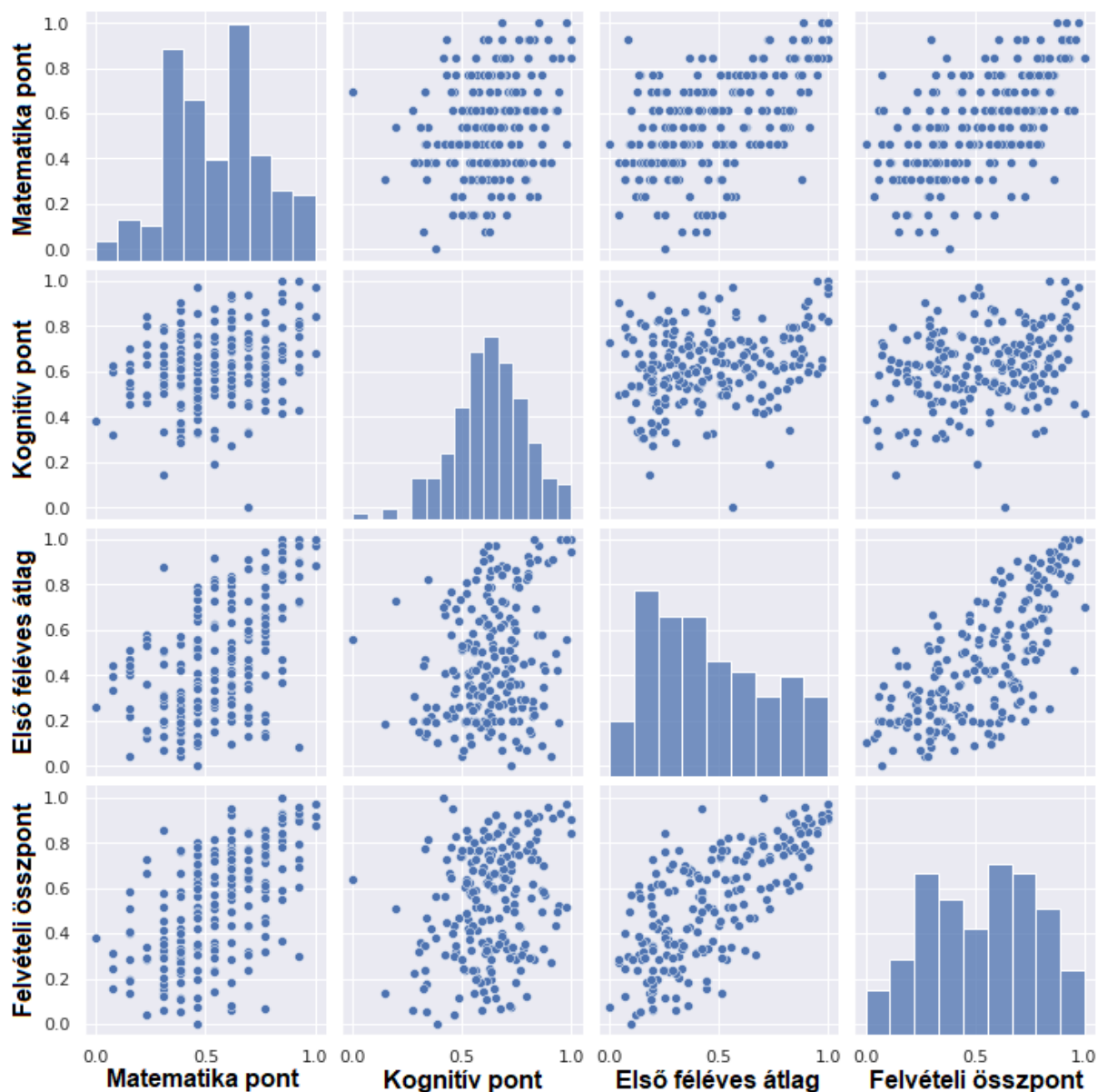
Egészen hasonló képet látunk, mint a 2019 esetében, de itt is érvényesül az erősebb szórás. Továbbá szembevető az érettségi pontszámok eloszlásának megváltozása. Míg 2019-ben a skálázott adatok nagy része 0.5-1 között van, 2021-ben már az egész intervallumra széthúzódnak.

Eddig a pontfelhők nem mutatnak jelentős korrelációt, illetve a kétdimenziós vetületekben nem emelkednek ki csoportok. Mivel más változókat is szerepeltek a vizsgálatban, azokról is készültek ábrák, amelyek alább tekinthetők meg.



6. ábra. 2019. Teszt részek, átlagok és felvételi összpontszám

Itt a felvételi pontszám össze lett vonva és be lett véve az első féléves és a későbbi kumulált átlag. Nyilván az utóbbi kettő erősen összefügg. Enyhén kirajzolódik egy összefüggés a felvételi pontszám és az átlag között. Érdekes, hogy a kumulált átlag eloszlása jobban közelít a normális eloszláshoz, mint az első féléves átlagé, ami eltolódik a skála alja felé.



7. ábra. 2021. Teszt részek, átlag és felvételi összpontszám

Ebben az esetben nem volt értelme a kumulált átlagot bevenni, ugyanis a kutatás idejében megegyezett az első féléves átlaggal. Itt is megmaradt a felvételi összpontszám és az átlag közötti összefüggés, de a pontfelhő kevésbé zajos. A matematikai rész pontszáma és az átlag között is van gyenge, pozitív összefüggés.

4.2. Folyamatábrák (Sankey-diagramok)

A Sankey-diagramok olyan folyamatábrák, amelyekben az ábrázolt folyamatok szélessége arányos a megfelelő folyamértékekkel. Ezáltal könnyen ábrázolhatóak adott állapotok közötti folyamatok, 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 rajta. Emiatt lesz némi redundancia a lentebb 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 diszkrétizálására. 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.

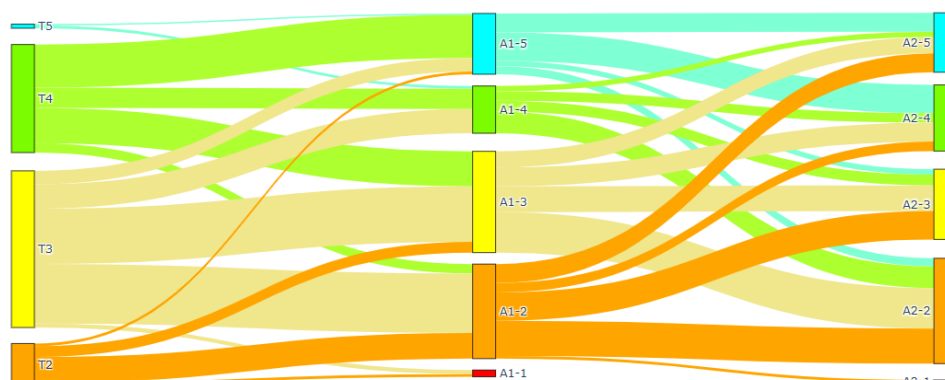
A *felvételi pontszám - nulladik ZH - kognitív teszt eredmény* változatban a nulladik zh és a teszt eredményei nullától az elérhető pontszámig lettek felosztva 20%-os lépésközökkel. Továbbá a felvételi pontszám a legalacsonyabb értéktől 500 pontig. A többi változat a kutatás későbbi szakaszában készült, így a folytonos mutatók az értékkészletük terjedelme szerint lettek felosztva, nem az elérhető pontszámok alapján.

Ezután *for ciklus* használatával elkészültek a tranzíciós mátrixok. Ezen mátrixok *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, továbbá kellett egy harmadik lista, amelynek a megfelelő indexű eleme a két állapot közötti folyamat 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 még 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 mozgatásával a hozzátartozó sávok automatikusan mozogtak együtt.



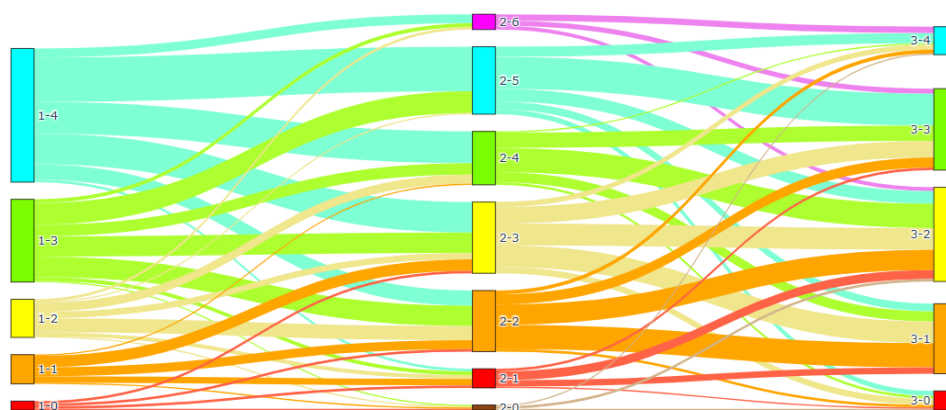
8. ábra. 2019. Kognitív teszt eredmény - A1 jegy - A2 jegy

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. Ehhez képest az tantárgyakon elért jegyek eloszlása egyenletesebb.

Látható, hogy akik a teszten gyengébben teljesítettek, azok a későbbiekben többnyire rosszabb jegyeket szereztek, de nem kevesen javítottak is. A jobban teljesítők szintén tartották általában a szintet, de itt sem elhanyagolható azok száma, aki rontott. Meglepő a kettesről négyes-ötösre javítók és az ötösről rosszabb jegyekre rontó hallgatók aránya a teljes viszonylatban.

Összességében sejthető, hogy egy hallgató teljesítménye a teszt során összefügg a to-vábbi teljesítményével, de viszonylag sok esetben van jelentős romlás, illetve javulás.

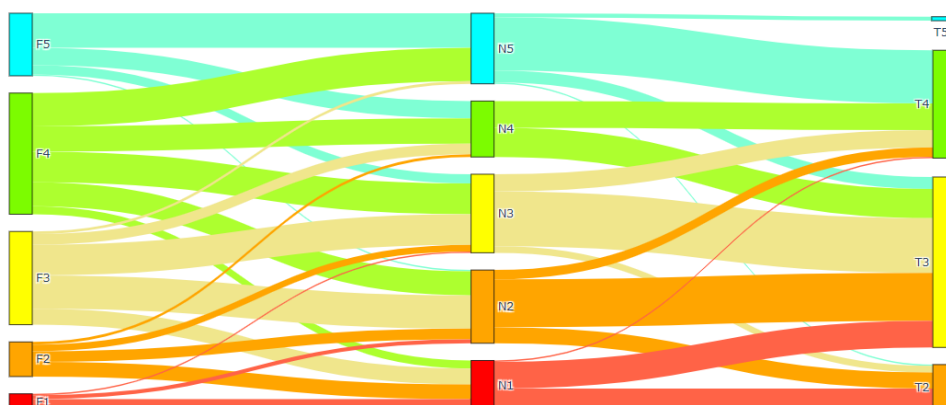


9. ábra. 2019. Matematikai teszt 1. blokk - 2. blokk - 3. blokk helyes válaszok száma

A teszt matematikai részének a komplexitása a blokkonként nőtt, így érdekes volt megvizsgálni, hogyan teljesítettek a hallgatók, és hogyan viszonyul egymáshoz a helyes válaszaik száma a különböző blokkokban.

Rögtön látszik, hogy az első blokkban a hibátlan (négyes) osztály a legnagyobb, és a rosszabb eredmények száma csökken. Ez várható, hiszen itt alapismeretek voltak felmérve. A második és harmadik blokknál teljesen más a kép, a közepes osztályok körülbelül azonosak, a szélsők pedig kicsik, jobban hasonlít a normális eloszlásra.

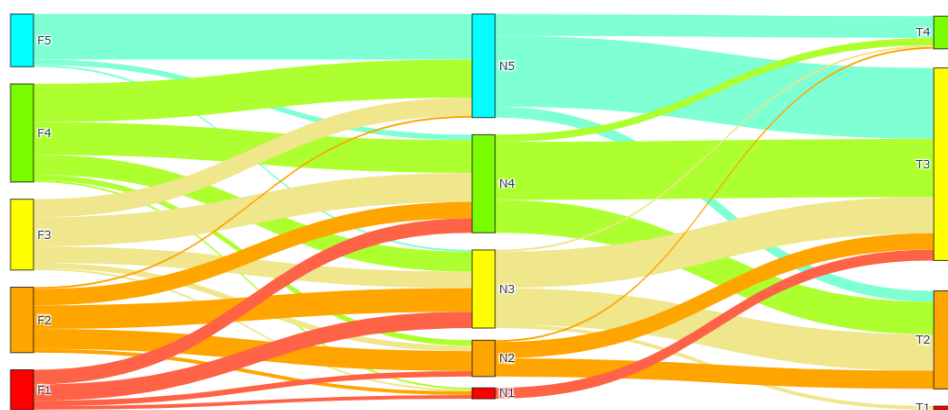
A folyamatok egészen változatosak. Az ábra közepén erősen szétszlanak, de a széleikhez közelítve kisebb trendek megfigyelhetők. Például a második blokkban minden kérdésre helyesen válaszolt hallgatók az első blokkban is többnyire szinte csak helyesen válaszoltak.



10. ábra. 2019. Felvételi - 0. zh - kognitív teszt pontszámok

Az ábrán főleg a 0. zh-ból a tesztbe menő folyamatokon látszik egy erős tendencia. Igaz itt még a régebbi, teljes szerezhető pontszám szerinti felosztás van, így csak négy osztály jött létre, de nagyon kevés többosztálynyi ugrás van, tehát feltehető, hogy erősen összefügg a 0. zh és a kognitív teszt eredménye.

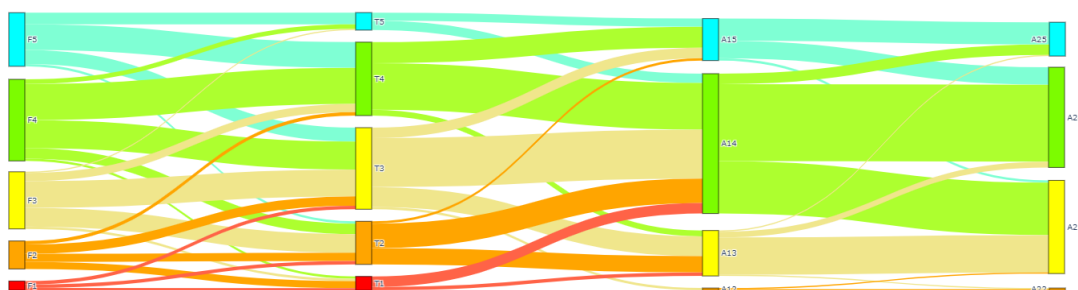
Ami a felvételi és 0. zh pontszámok kapcsolatait illeti, a folyamatok egészen szerteágazóak, de itt is fennáll, hogy a közel hasonló teljesítmények között szélesebbek, mint a nagyobb különbségek esetén.



11. ábra. 2021. Felvételi - 0. zh - kognitív teszt pontszámok

Annak ellenére, hogy a tesztes oszlop a 2019-es tesztes oszlopra hasonlít, csak fejjel lefelé, ez a valós eloszlás. Valóban, míg 2019-ben nem volt 20% alatti eredmény, itt fordítva, nem volt 80% fölötti. A felvételi pontszámok arányai romlottak, arányosan nagyobbak lettek az alsó osztályok. Ellenkezőleg a 0. zh eredményei javultak, nagyobbak a magasabb pontszámhoz tartozó osztályok arányai.

Ami a folyamatokat illeti, többnyire hasonlóak a tendenciák, bár érdekes, hogy akiknek a legrosszabb lett a 0. zh, a kognitív teszt közepesen ment. Itt is mondható, hogy a jobban teljesítő hallgatók többnyire mindenből jól teljesítettek, illetve a gyengébbek között ritkábbak voltak a jobb eredmények. Ennek ellenére itt is szerteágazóak a folyamatok, tehát összefüggés van, de nem nagyon erős.

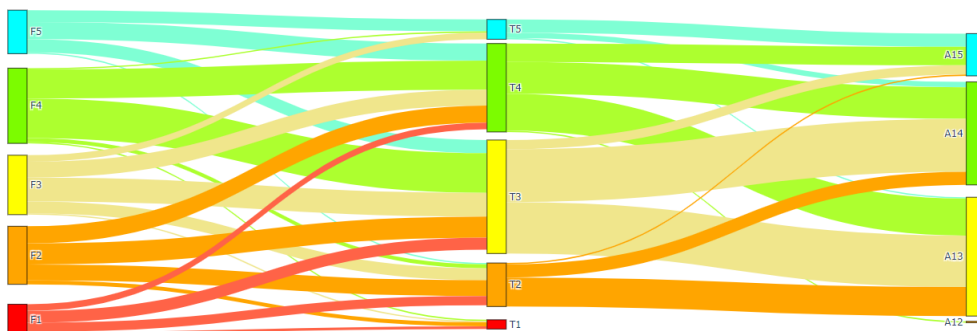


12. ábra. 2019. Felvételi - kognitív teszt pontszám - első féléves átlag - kumulált átlag

Ezen az ábrán már közvetlenül láthatóak a felvételi és kognitív teszt eredményeinek viszonyai, azonban itt már az értékkészlet van felosztva ekvidisztánsan. Ez abban is megnyilvánul, hogy a teszt pontszámosztályainak mások az arányai.

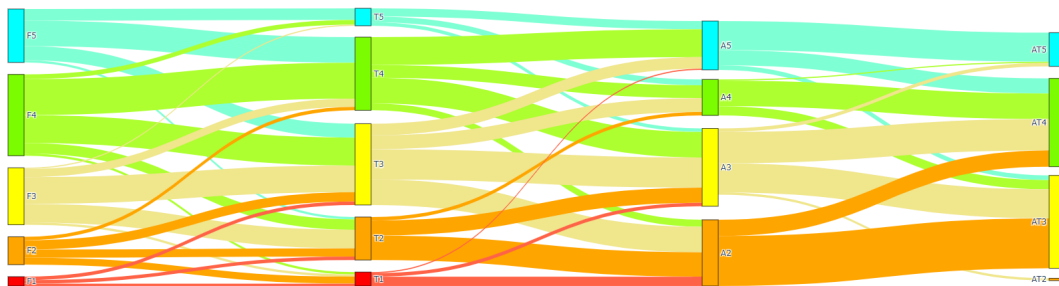
Két előnye volt az értékkészlet szerinti osztályozásnak: gyorsabb volt (egy függvényre volt csak szükség), illetve így az eredmények egymáshoz képest való relatív eloszlása jobban látszik. Míg a régebbi ábrán nagyon nagy a hármas osztály, mert a legtöbben 40-60% körüli eredményeket értek el, itt az látszik, hogy a legjobb és legrosszabb dolgozók pontszámai között, a terjedelem 20%-ával lépkedve milyen a hallgatók eloszlása. Így az lesz valóban "középtájban", aki a többi hallgató pontszámához képest átlaghoz közeli pontszámot szerzett, nem pedig az a sok tanuló, aki 40-60%-ot ért el.

Visszatérve az ábrára, a fentebb megfigyelt tendenciák lényegében ismétlődnek, jó eredményeket jók, rosszakat rosszak követik nagyrészt, de továbbra is változatosak a folyamatok. Összefüggőség szempontjából jó jel, hogy a legjobb teszteredményt elérők négyesnél rosszabb átlagot nem értek el. Sőt nagyon sok hallgatónak volt négyes az átlaga az első félévben, ami nagyrészt kétfelé oszlott az idő elteltével, többen megtartották, de sokan rontottak hármasra, páran javítottak. A többi átlagértékkel hasonló a helyzet, inkább megmaradt, mint javult vagy romlott.



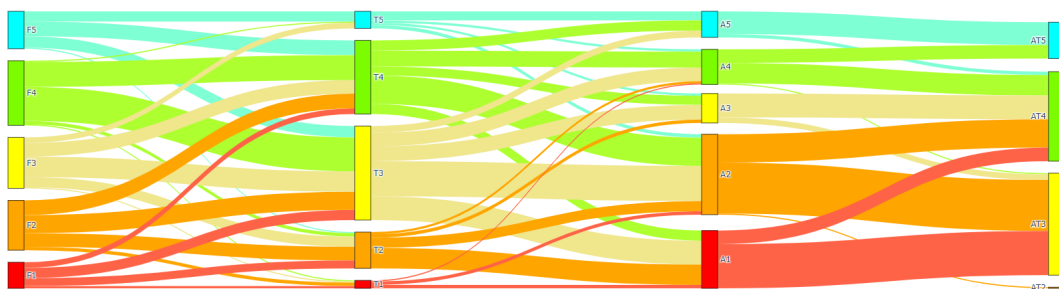
13. ábra. 2021. Felvételi - kognitív teszt pontszám - első féléves átlag

A kutatás során ez az évfolyam még az első évében volt, így csak első féléves átlagokról volt adatunk. Ettől eltekintve 2019-hez képest több hallgatónak volt relatíve gyengébb a felvételi pontszáma, de ennek megfelelően többen is "javítottak". Az átlagok eloszlása romlott, aránylag több hármas és kevesebb négyes, illetve ötös lett.



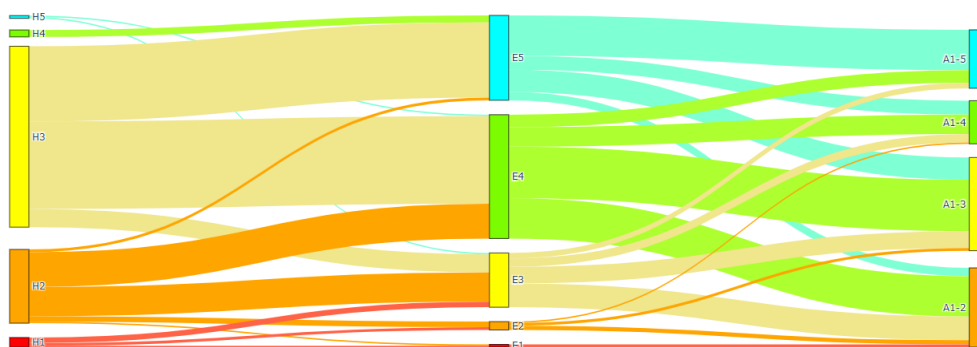
14. ábra. 2019. Felvételi - kognitív teszt pontszám - A1 jegy - kumulált átlag

Az első két oszlop viszonyát már láttuk, az újdonság az A1 jegy az első féléves átlag helyett. Ettől függetlenül a tendenciák változatlanok, azok szereztek jobb, illetve rosszabb jegyet, akik rendszerint jobban, illetve rosszabban írták meg a tesztet. A trendet továbbra erősíti a jegyek és átlagok viszonya, bár értelemszerűen nem meglepő, hogy a két dolog összefügg.



15. ábra. 2021. Felvételi - kognitív teszt pontszám - A1 jegy - első féléves átlag

Szembetűnő, hogy az A1 jegyek valamilyen okból kifolyólag erősen romlottak. Míg 2019-ben nem bukott egy hallgató sem, itt majdnem a csoport negyedének nem sikerült elvégezni a tárgyat, sőt olyanoknak is, akik a legjobbak között voltak a teszt alapján. Továbbá a csoportnak több, mint a fele nem tudott kettesnél jobb jegyet szerezni. Fentebb láttuk, hogy az átlagok is romlottak, de ezek szerint nem olyan mértékben, mint az A1 jegyek. Érdekes, hogy az egyest és kettest szerzett hallgatók mégis nagyrészt hármas, sőt nem kevesen négyes átlagot szereztek. Ettől függetlenül a szokásos trendek itt is megmutatkoznak.

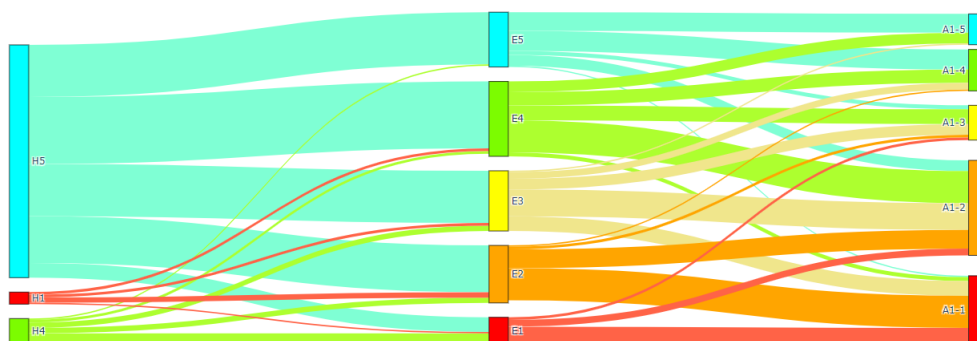


16. ábra. 2019. Hozott és - érettségi pontszám - A1 jegy

Ahogy később látni fogjuk, a modellek alapján a hozott és érettségi pontszámok erősen befolyásolták a jegyekre vett predikciókat, így utólag elkészült ez és a következő ábra.

Jellegzetes, hogy a hallgatók nagy részének hasonló volt a hozott pontszáma, nagyon kevésnek volt kiemelkedő vagy relatíve kicsi. Aránylag kevés hallgatónak volt a legjobb pontszám 60%-a alatt az érettségije. Még ezek mellett is megőrződött az a tendencia, hogy egy-egy hallgató nagyrészt hasonlóan teljesít a különböző mutatók alapján.

Az ábráról leolvasható, hogy míg az utolsó oszlop felső felébe szinte csak kék és zöld folyamatok érkeznek, mégis szerteágaznak, és például a legjobb érettségi pontszámokkal rendelkezők közül is többen kettest szereztek a tárgyból.



17. ábra. 2021. Hozott és - érettségi pontszám - A1 jegy

Az ábrán rögtön szembetűnik egy meglepő dolog, a hozott pontszámoknál van egy óriási ötös osztály. Az értékeket leellenőrizve, tényleg nagyon sok esetben 180 és 190 között mozgott a hozott pontszám.

Ehhez, és a 2019-es ábrához képest az érettségi pontszám viszont egészen egyenletesen oszlott el. Az ábra bal felétől eltekintve, a teljesítmény megmaradására vonatkozó tendencia igaz gyengébben, de ismét jelen van.

4.3. Klaszterezés

A klaszterezés célja általában esetleges nem triviális összefüggések feltárása, illetve az adatok csoportokba való sorolása gépi tanulási módszerekkel úgy, hogy az azonos csoportban lévő adatok "hasonlóbbak" legyenek egymás közt, mint a különböző csoportokból vett adatok.

Egyes esetekben a klaszterezés eredménye rámutathat néhány változóra, amelyeknek erős a magyarázóereje, így a további vizsgálatokból el lehet hagyni a többi változót. Ennek ellenére gyakoriak azok az esetek, amikor nehéz vagy nem lehet megállapítani ezeket a változókat, mert minden változó fontos. Ilyenkor az adatok két-három dimenziós vetületeinek szórásdiagramjai többnyire homogének és nem csoportosíthatóak szemre.

A 4.1 alfejezetben szereplő ábrákat áttekintve, első ránézésre nem rajzolódott ki olyan kétdimenziós vetület, amelyben láthatóan megjelennének csoportok. Ennek ellenére három gépi tanulási módszer lett kipróbálva:

1. K-közép algoritmus
2. DBSCAN algoritmus
3. Ward féle hierarchikus klaszterező algoritmus.

Számos optimalizációs kísérlet volt: hiperparaméterek beállítása, változók válogatása, illetve többféle skálázás. Volt egy kísérlet PCA alkalmazására, azonban minden esetben csak az összes megadott változó reprezentálta eléggé jól az adatot, így ez el lett vetve.

Sok modell és klaszterkép készült, a legérdekesebbek lentebb lesznek bemutatva. A szemléltetéshez a bevett változók közötti páros szórásdiagramok lettek legenerálva és a pontok az alapján lettek színezve, amely klaszterbe kerültek.

4.3.1. K-közép algoritmus működése

Az algoritmus működési elve nagyon egyszerű. Az alapvető paraméter, amelyet meg kell adni, a létrehozandó osztályok kívánt száma, legyen ez k . Első lépésben választ a térben véletlenszerűen k pontot, ezek a klaszter középpontok, majd az összes pontot ahhoz a középponthoz rendeli, amelyhez a legközelebb van. A második lépésben a kialakult osztályok középpontjait újraszámolja és azokkal a középpontokkal megismétli az első lépést. Ezt a két lépést iterálja addig, amíg két egymást követő osztályozás között már nem lesz jelentős változás.

Így alapvetően a legfontosabb feladat a megfelelő klaszterszám meghatározása. Ehhez létezik egy könyökszabály, melynek a lényege a következő: lefuttatjuk a klaszterezési algoritmust 1-től l -ig megadva a klaszterszámokat, majd ábrázoljuk ezek esetében a pontoknak az átlagos négyzetes eltérését a saját klaszter-középpontjuktól. Természetesen ahogy több csoportra bontjuk a pontokat, úgy csökken ez az eltérés, így az a klaszterszám lesz optimális, ahol jelentős a javulás, de utána már nem. Ez megfelelő esetben vizuálisan egy behajlított karra emlékeztet, innen ered a neve. Nehezen klaszterezhető adatok esetén ez a görbe "simább", nehezen állapítható meg a "könyök".

4.3.2. DBSCAN algoritmus működése

A DBSCAN (Density-based spatial clustering of applications with noise) sűrűség alapú klaszterező algoritmus. Ebben az esetben a sűrűség egy bizonyos ϵ sugáron belüli pontok száma. Ez alapján az algoritmus három kategóriára osztja az adatpontokat:

- Magpont: olyan adatpont, melynek egy meghatározott számúnál (μ) több pont van az ϵ sugarú környezetében.
- Határpont: olyan adatpont, amelynek μ -nél kevesebb pont van az ϵ sugarú környezetében, de ő maga egy magpont ϵ sugarú környezetébe esik.
- Zaj (outlier): olyan adatpont, amely egyike sem az előzőknek.

Működése során először felosztja az adatpontokat a három leírt típus szerint, majd a zajos pontokat figyelmen kívül hagyja. A többi pont esetén a sűrűn összefüggő részek határozzák meg a klasztereket. Előnye, hogy kiszűri a zajos pontokat és érzéketlen rájuk, illetve jól kezeli a különböző méretű osztályokat. Változó sűrűségű csoportok esetén gyengébben teljesít.

A felhasználó feladata meghatározni a μ és ϵ értékeket, illetve a megfelelő távolságfogalmat. Ehhez szintén létezik egy könyökszabály. Ennél az algoritmusnál növekvő

sorrendben kell ábrázolni az egyes adatpontoknak a legközelebbi szomszédjuktól vett távolságot. Mivel sok adatpont van, a könyök simább, így az ϵ -t csak becsülni tudjuk.

Ezután a μ paramétert kell hozzáigazítani oly módon, hogy a kialakuló klaszterszám megfelelő legyen. Ez úgy történik, hogy az adott paraméterárral lefuttatjuk az algoritmust és megnézzük a keletkezett klaszterek számát. Azt a párt választjuk, aminél megfelelő a klaszterszám.

4.3.3. Ward féle hierarchikus klaszterező algoritmus működése

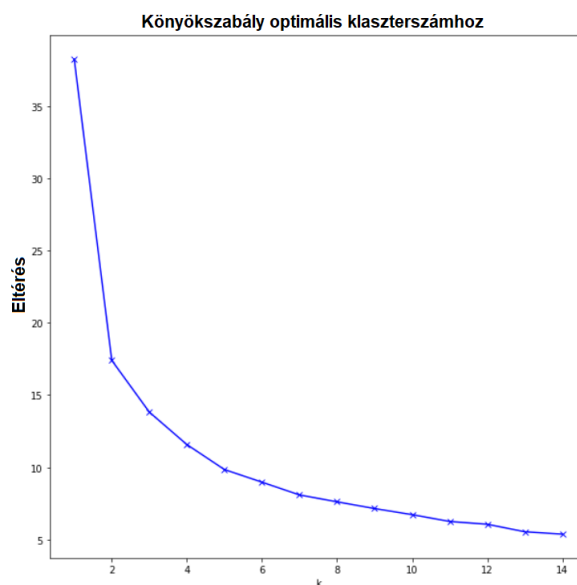
Az általunk használt hierarchikus klaszterező algoritmus agglomeratív módon hozza létre a klasztereket. Azaz először minden pontot külön klaszternek tekint, majd ezeket egyesíti egy kiválasztott módszer alapján addig, amíg az összes pont egy klaszterbe nem kerül. Az iterációkat dendrogrammal lehet ábrázolni, és az alapján lehet dönteni, hány klasztert akarunk létrehozni. Gyakorlatban a dendrogramon meg kell keresni a legszélesebb vízszintes sávot, amelyben csak függőleges szakaszok vannak, és venni azok számát.

Az alapvető egyesítő módszerek a single, average, complete linkage nevet hordozzák, de mi a Ward féle módszert alkalmaztuk. Ez a módszer az összes klaszteren belüli szórásnégyzet minimalizálására törekszik, így valamelyest hasonlít a k-közép algoritmusra.

4.3.4. K-közép algoritmus eredménye

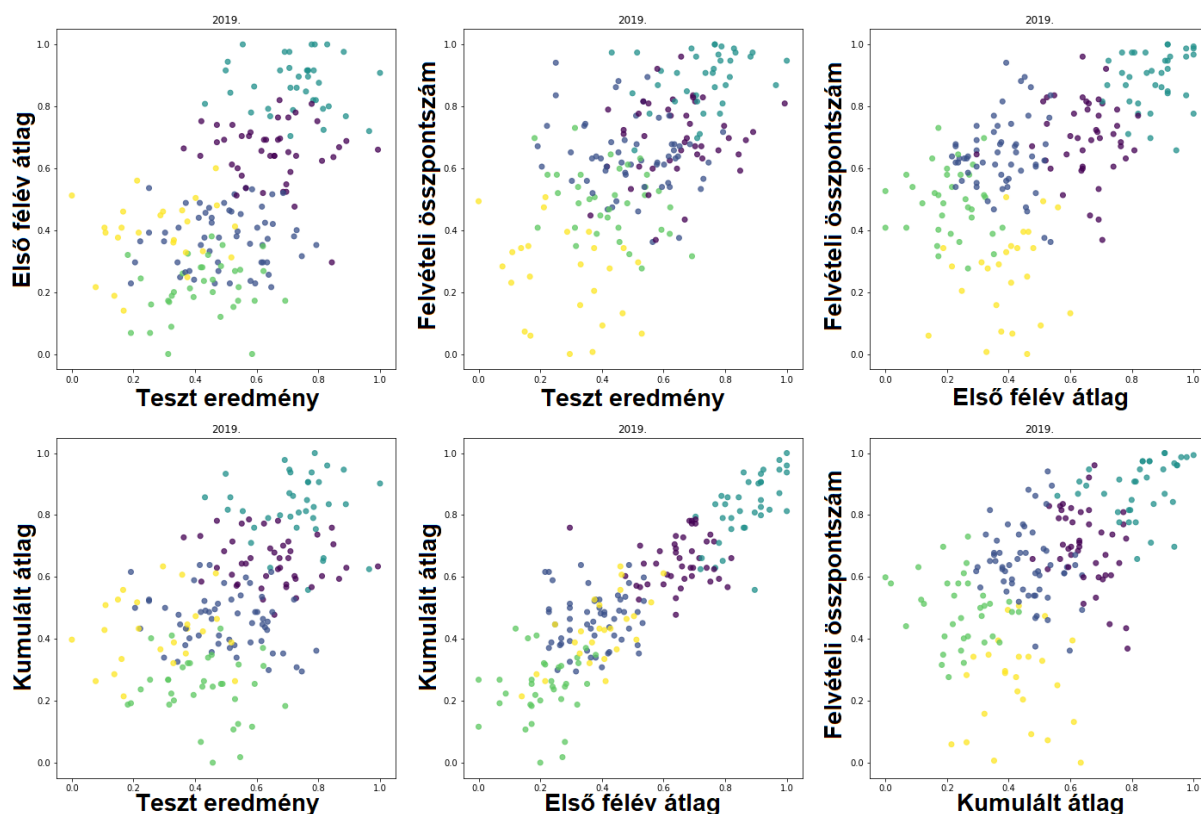
Ami a szórásvázlatokból sejthető volt, a könyökszabály alkalmazása során igazolódott be. Szinte minden próbálkozás esetén sima volt a görbe, és nem volt egyértelműen jó klaszterezés. Ezekben az esetekben a klaszterkép is elmosott volt.

A k-közép algoritmus eredményei közül kettőt emelhetünk ki. Az egyikben a 2019-es adatokból négy változó szerepelt, min-max skálázva: egyesített teszteredmény, első féléves átlag, kumulált átlag és a felvételi összpontszám. A másikban ugyanez a felállítás 2021-re, kivéve a kumulált átlagot, mert az megegyezett az első félévessel. Az első változatnak a könyökszabály ábrája tekinthető meg alább:



18. ábra. 2019. K-közép, 4 változós könyökszabály

Természetesen ezen az ábrán sincs teljesen egyértelmű törés, de 5 klaszter esetén mondható közel optimálisnak. Ezzel az alábbi klaszterképet generálta az algoritmus:

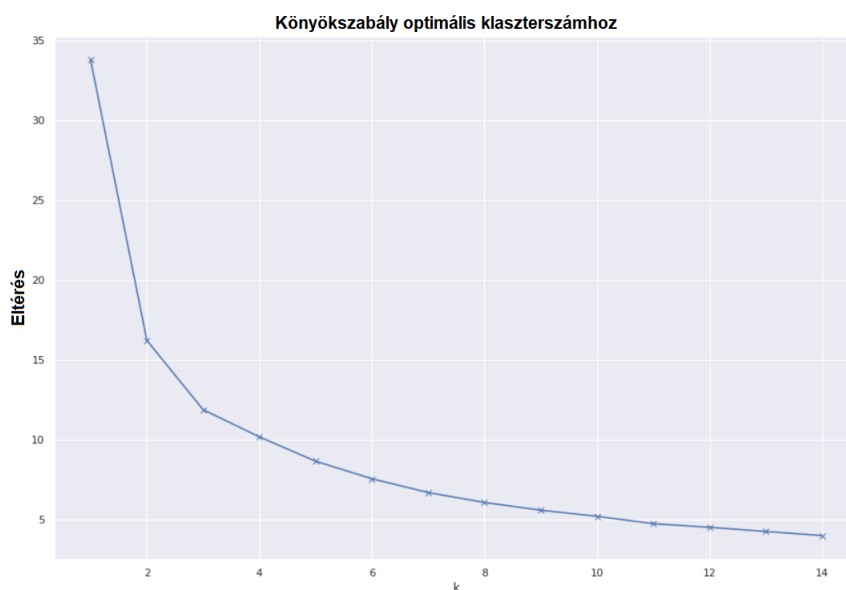


19. ábra. 2019. K-közép, 4 változós klaszterkép

Az ábrák közül a legtisztábban a felvételi összpontszám és az első féléves átlag szórásdiagramján látszanak az osztályhatárok. Megfigyelhető az algoritmus jellegzetessége, a hasonló méretű, "gömböses" klaszterek kialakítása. A kékes zöld klaszter a legstabilabb,

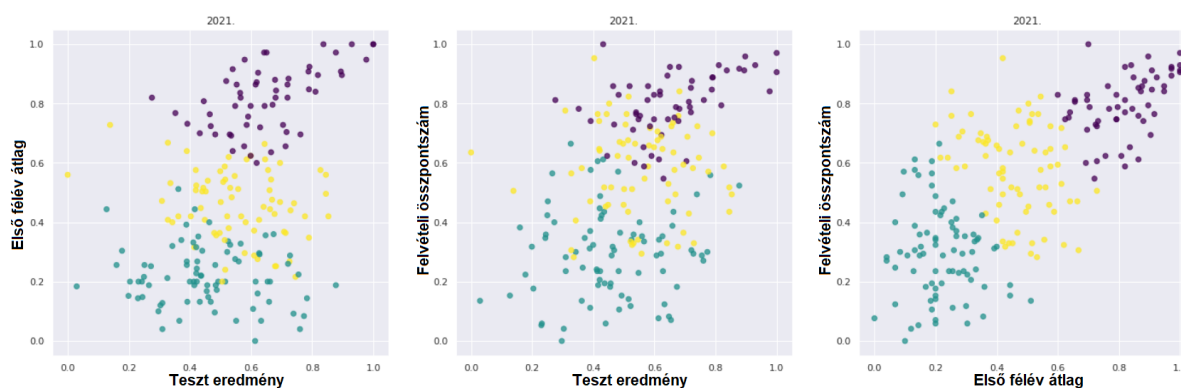
annak az adatpontjai mindig a jobb felső sarokban vannak, azaz minden változóban magas értékeik vannak. A lila hasonlóan viselkedik, csak mérsékeltebb, minden változóban átlag feletti, de nem a legjobb. Ellenben a zöld klaszter azokból a hallgatókból áll, akik átlagosan teljesítettek a felvételi és a teszt során, de más mutatókban lecsúsztak. A kék a legbizonytalanabb, ugyanis többnyire középtűt van mindig, de mélyen belemetsz más klaszterokba. A sárga klaszter pontjai minden változó esetén a skála alsó felén vannak, kivéve a kumulált átlagot, ahol középtűt összpontosulnak.

Most tekintsük a 2021-es eredményt:



20. ábra. 2021. K-közép, 3 változós könyökszabály

Ez az ábra jobb az előzőnél, ugyanis 3-nál még érzékeny a törés, de utána szinte kisimul.



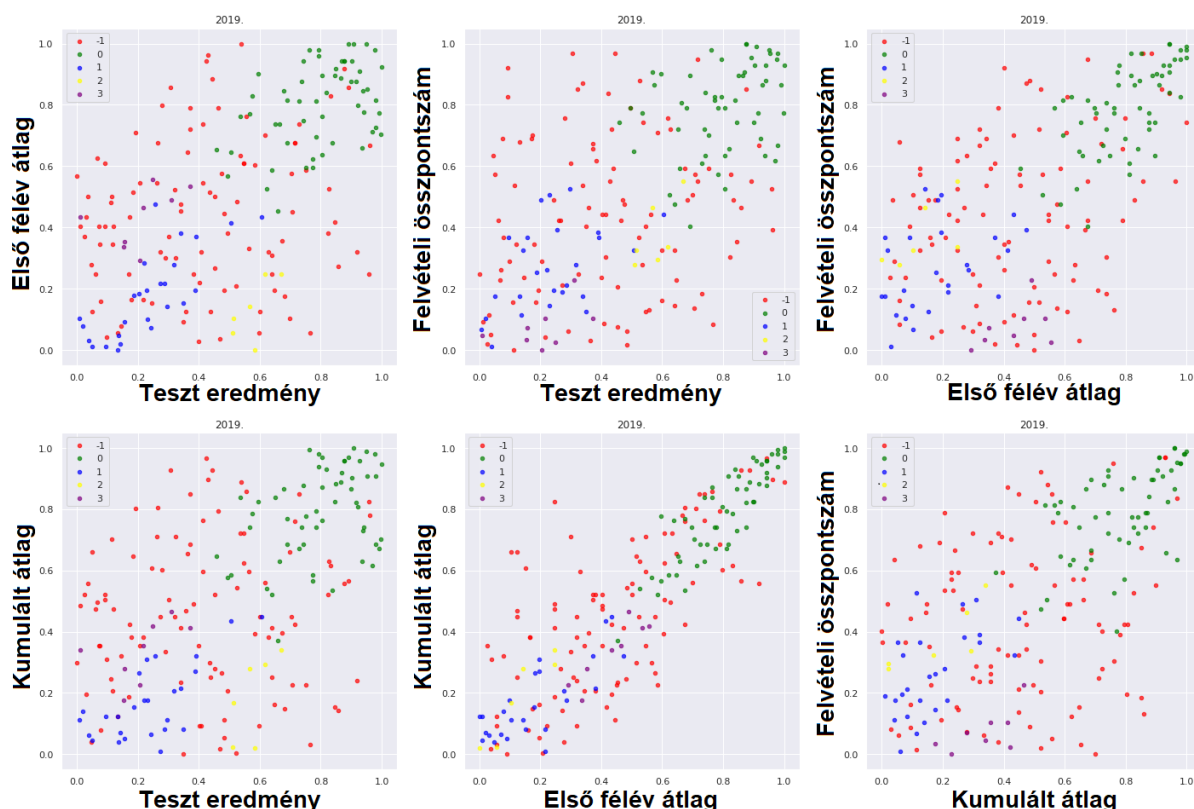
21. ábra. 2021. K-közép, 3 változós klaszterkép

Ebben az esetben még tisztább az elkülönülés az első féléves átlag és a felvételi összpontszám esetén, tehát feltehetőleg ezek jobban befolyásolták a döntést. Itt is gömbszerű klaszterek lettek, ugyanis a pontfelhő egészen homogén.

4.3.5. DBSCAN eredménye

A DBSCAN algoritmus akkor működik igazán jól, ha az adatok pontfelhője közel azonos sűrűségű részekből áll, amelyek között nagyobb távolság van. Az elkészített, és fentebb bemutatott ábrákon nem volt jellemző az előbb említett tulajdonság. Ennek következményeként az algoritmussal nem sikerült jó klaszterezést megvalósítani.

A futtatások során az eredmények két csoportba voltak kategorizálhatóak. Az egyikben létrejött egy nagy klaszter és elenyésző volt a többi, a másikban a pontok nagy része "outlier" volt és néhány nagyobb, de továbbra is kis méretű klaszter jelent meg. Egy ilyen tekinthetünk meg alább.



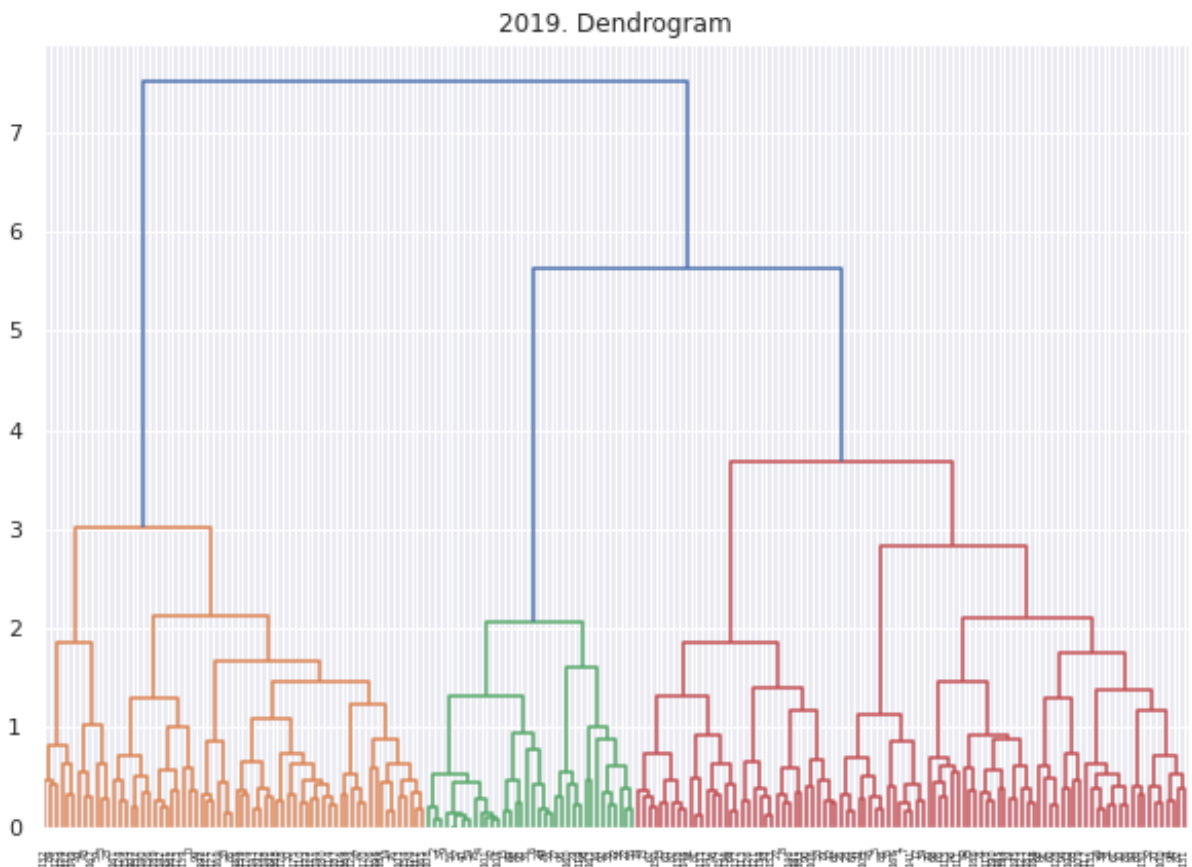
22. ábra. 2019. DBSCAN, 4 változós klaszterkép

A klaszterezés 2019-es adatokon készült, Mahalanobis távolságot használva és kvantilizáció mentén skálázva. Ismét ugyanaz a négy változó került be: egyesített teszt eredmény, első féléves átlag, kumulált átlag és a felvételi összpontszám. Látszik, hogy piros pontokból van a legtöbb, amelyekhez a -1-edik klasztert rendelte az algoritmus, ezek az outlierok. A zöld és a kék klaszter még viszonylag nagy, de a többi csak pár pontot tartalmaz.

4.3.6. Ward féle algoritmus eredménye

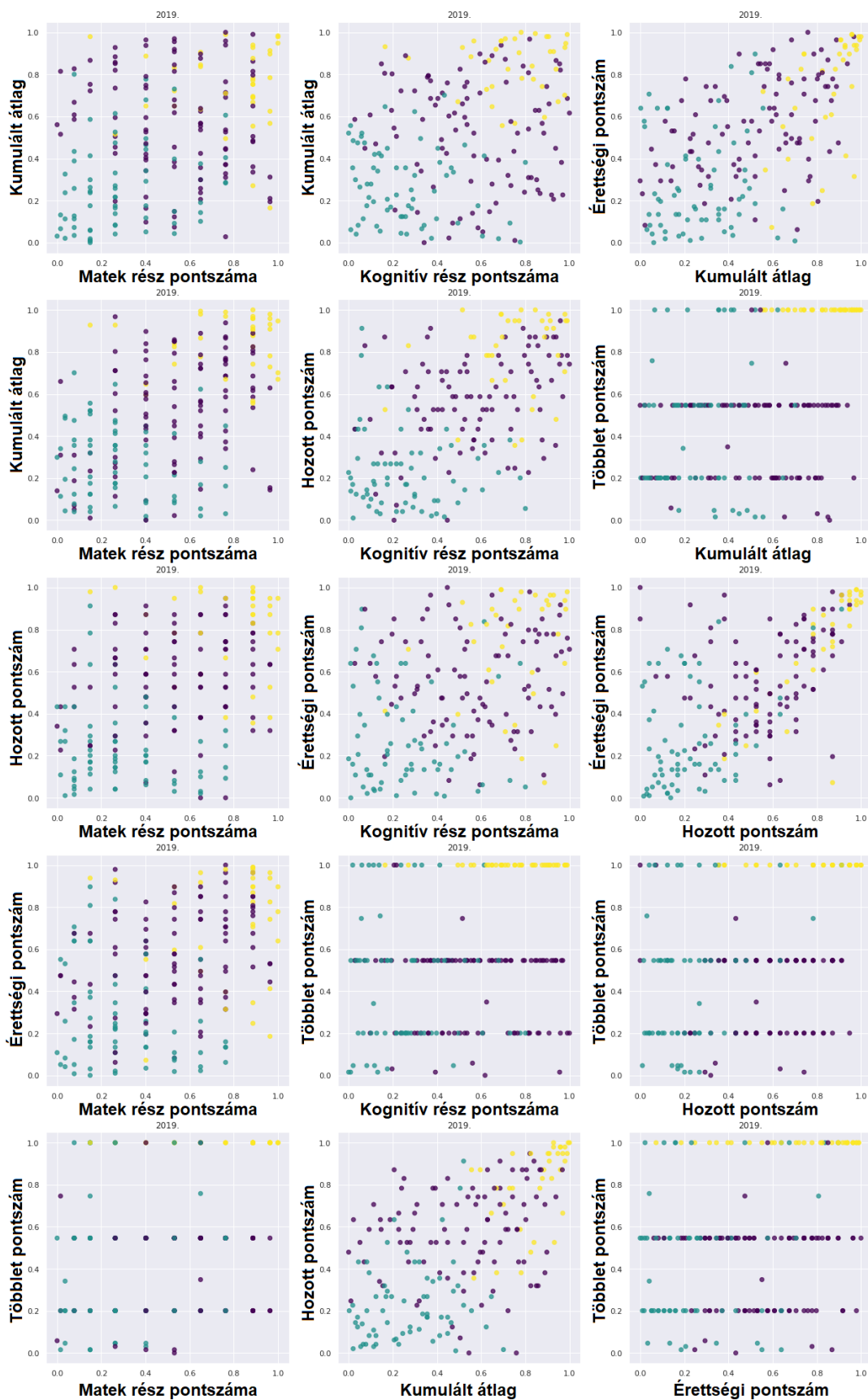
Mint ahogy a másik két módszernél a könyökszabályokban, itt a dendrogramokban volt nehéz eldönteni az optimális paraméter értéket, jelen esetben a klaszterszámot. Ha egy

szám optimális, akkor az a síkrész, amelyben annyi függőleges vonal megy, kiemelkedően széles. Ha nehezen klaszterezhető az adat, akkor kisebbek a sávok, és azok közül a legnagyobbak közel azonos méretűek. Ezt lehet látni az alábbi ábrán:



23. ábra. Dendrogram

Itt a kettő és a három függőleges szálát tartalmazó sávok szélessége közel azonos. A választás három klaszterre esett. Kevesebb változó választásánál a klaszterek száma minden esetben legfeljebb kettő esetén lett volna optimális, így egészen sokat kellett használni: matematikai rész pontszáma, kognitív rész pontszáma, kumulált átlag, hozott pontszám, szerzett érettségi pontszám, többlet pontszám.



24. ábra. Ward féle algoritmus eredménye

A változók nagy száma miatt lett ilyen nagy a párosítási ábra. Mindez a 2019-es

adatokból van, kvantilisek mentén skálázva. Minden erőfeszítés ellenére nem alakult ki tisztább klaszterkép. Itt is a k-középhez hasonlóan a három klaszter a "jók", "közepesek" és a "gyengébbek" osztályait jelenti, de nincs vetület, amin olyan sima lenne a klaszterhatár.

5. Prediktív analitika

5.1. Modellek és metodológia

Következő lépésként azt vizsgáltuk, hogy a két évben külön a bemeneti adatok alapján mennyire pontosan lehet előrejelezni egyes teljesítménymutatók értékét, 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[8]. Ekkor a gépi tanulás során a teljes adathalmazt tanító és tesztalmazra 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 legoptimálisabb 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 tesztalmazon is az általánosítóképesség ellenőrzése végett.

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 pedig lineáris regressziót és *Gradient Tree Boosting*-ot (ezen algoritmusok 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.[2, 3, 4, 5] 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 vonatkozólag illetve szétbontva vegyészmérnök és biomérnök hallgatókra egy-

aránt (a környezetmérnökökről nem állt rendelkezésre elég adat, így őket a szakonkénti bontásban kihagytuk).

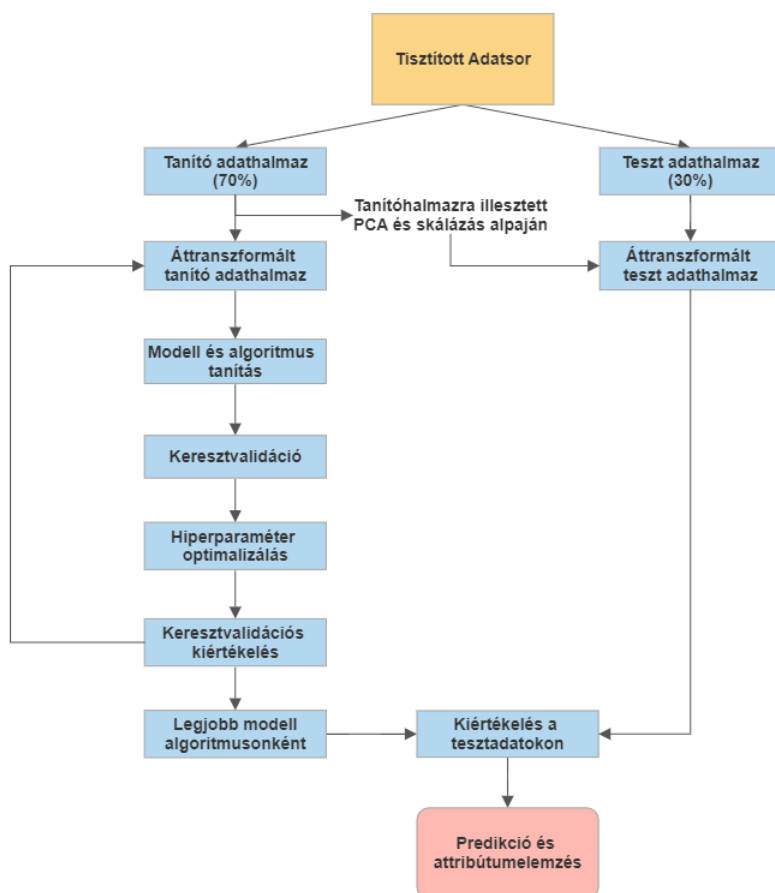
A használandó osztályozó algoritmusok közül a Naive Bayes és Gradient Tree Boosting algoritmusok képesek kezelni a többosztályos feladatokat, a lineáris regresszió azonban folytonos célváltozóérték prediktáláshoz használható, így ott a prediktált értéket kerekítettük a legközelebbi címkeértékhez. A logisztikus regresszió és az SVM 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[10]. Az új változók, amelyeket főkomponenseknek nevezünk, a kisebb dimenziós térben az eredeti változók tapasztalati 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

A vizsgálatokat Python-ban végeztük el az *Sklearn*[6] csomag használatával. A *jegycsoportok* prediktálására épített modellezési struktúrát a 25. ábra mutatja. Az egyes csoportmodelleknél 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 kvantilisek mentén a $[0, 1]$ intervallumba transzformáltuk át, valamint az érdemjegy csoportok prediktálásához a 2 és 3 csoportmodelleknél a célváltozó-értékeket rendre 1, 0-ra illetve 3, 2, 1-re módosítottuk.

Valamennyi algoritmus hiperparamétereinek optimális megválasztására 5-szörös keresztvalidációt alkalmaztunk, ahol törekedtünk arra, hogy az adatrekordok címkéjének eloszlása egyenletes legyen a felosztott részek között. A keresztvalidációnál használt, jó-ságot mérő metrikának a kiegyensúlyozott pontosságot (*'balanced accuracy'*) választottuk,



25. ábra. A jegycsoportokra irányuló modellezési struktúra sematikus ábrája

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 többsztályos 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 két valamint több osztály esetén:

- **TP** a pozitívnak (osztálybelinek) osztályozott, valóban pozitív (osztálybeli) adatrekordok száma
- **FP** a pozitívnak (osztálybelinek) osztályozott de valójában negatív (nem osztálybeli) adatrekordok száma
- **FN** a negatívnak (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 teszt-halmazon visszamértünk, valamint kiértékeljük az így módon választott modellek esetén az egyes változók fontosságát is.

A *kumulált átlag* predikciójánál a tisztított adathalmazt ugyanúgy 70-30 arányban osztottuk fel. Adattranszformálásra csak skálázást alkalmaztunk, PCA-t nem, majd valamennyi osztályozó hiperparamétereinek optimalizálása hasonlóan 5-szörös keresztvalidációval történt. Az illesztett legjobb modelleknél mindkét évben feljegyeztük a reziduális tagok szórásdiagramját, az egyes változók prediktív erejét, valamint az alábbi statisztikák értékét[9]

- R^2 : A modell hatásfokát mérő mutató, értéke $[-\infty, 1]$ közötti:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

ahol y_i és \hat{y}_i rendre az i -edik adatrekord valódi és prediktált célváltozóértéke, \bar{y} pedig a valódi célváltozóértékek átlaga.

- MAE : Átlagos abszolút eltérés (A keresztvalidáció során ez volt a használt mérőszám is):

$$MAE = \frac{1}{n} \sum_i^n |y_i - \hat{y}_i|$$

- $RMSE$: Gyököt vont átlagos négyzetes eltérés:

$$RMSE = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2}$$

5.2. Osztályozó algoritmusok és optimalizálásuk

5.2.1. 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 célváltozóérték és a prediktált érték közötti négyzetes hibát.

Mivel az algoritmus implementálása során a legkisebb négyzetek elvét alkalmazza, amellyel az optimális megoldás analitikusan elérhető, ezért nem történt hiperparaméter-optimalizáció.

5.2.2. 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.

Az algoritmus a jellegéből adódóan nem igazán optimalizálható, így a Naive Bayes esetén nem történt keresztvalidáció.

5.2.3. 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.[1] Gradient Tree Boosting esetén a gyenge prediktorok *döntési fák*, amelyek lefelé irányított, legtöbbször bináris fák, továbbá minden belső csúcsban egy attribútumra vonatkozó feltétel szerepel, a levelek pedig valamilyen célváltozóértékkel címkézettek. Az adatrekordok osztályozása intuitívan a fán való végigvezetéssel történik, végül azt a címkét prediktálva nekik, amilyen címkéjű levélbe jutottak.

A boosting eljárás során minden fázisban egy új döntési fát építünk, amely megpróbálja az előző fázisban épített fa hibáit csökkenteni az úgynevezett reziduálisokra építve. A cél egy erős prediktorként funkcionáló döntési fa létrehozása ráerősítések sorozata révén. Az algoritmus alkalmas osztályozási és regressziós problémák megoldására is, előbbi esetben a cél a levelekben az adatrekordok címke szerinti homogenitásának maximalizálása, utóbbinál pedig a szórás minimalizálása az egy levélbe kerülő rekordok célváltozójára nézve.

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)
- Facsúcsokban használt vágási feltétel (négyzetes hiba, Friedman MSE)
- Fák maximális mélysége (3,4,5 és 6 között változtatva)

5.2.4. 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 szigmoid¹ függvény bemeneteként, amelyet a függvény leképez a (0, 1) intervallumra,

¹A 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.

és amennyiben az output 0.5-nél nagyobb, úgy a pozitív osztályba soroljuk az adott rekordot, különben a negatívba.

A keresztvalidálás során optimalizált paraméterek:

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

5.2.5. SVM

Az SVM (Support Vector Machine) algoritmus a logisztikus regresszióhoz hasonlóan egy lineárisan szeparáló hipersíkot akar meghatározni[7]. Lényege, hogy különböző magfüggvények segítségével az adatrekordokat egy magasabb dimenziós térbe képi le, ahol olyan szeparáló hipersíkot keres, amely maximalizálja a vele párhuzamos hipersíkok által meghatározott olyan térrészt, amely adatpontot nem tartalmaz. A cél a megfelelő magfüggvény és a legoptimálisabb 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)

6. Modellek kiértékelése

		Osztályozó algoritmusok					
		Grad. Boost.	Naive B.	Log. reg.	SVM	Lin. reg.	
3 csoport	Összesített	2 PC	66.67	62.67	52.00	58.67	61.33
		4 PC	70.67	60.00	57.33	53.33	65.33
		6 PC	65.33	62.67	54.67	68.00	64.00
		8 PC	64.00	68.00	53.33	62.67	64.00
	Szakonként	2 PC	78.71	80.36	73.85	67.24	80.36
		4 PC	75.41	80.36	47.80	65.59	78.71
		6 PC	82.02	80.36	75.47	67.21	82.02
		8 PC	80.36	78.71	37.42	75.44	78.71
2 csoport	Összesített	2 PC	59.42	69.57	69.57	72.46	69.57
		4 PC	59.42	71.01	73.91	69.57	73.91
		6 PC	63.77	69.57	73.91	71.01	73.91
		8 PC	68.12	69.57	71.01	73.91	72.46
	Szakonként	2 PC	67.31	67.31	65.69	62.41	64.03
		4 PC	64.00	64.00	67.31	67.31	63.97
		6 PC	64.03	65.69	64.07	65.65	67.27
		8 PC	64.07	62.51	57.49	67.34	68.96

2. táblázat. A 2019-es adatsor eredményei

Először a **jegycsoportok** prediktálásának eredményeit ábrázoljuk. 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 átlagai súlyozva az egyes szakokon tanuló hallgatók számával. Az algoritmusok többségének hatásfoka 60-80% közé tehető. 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.

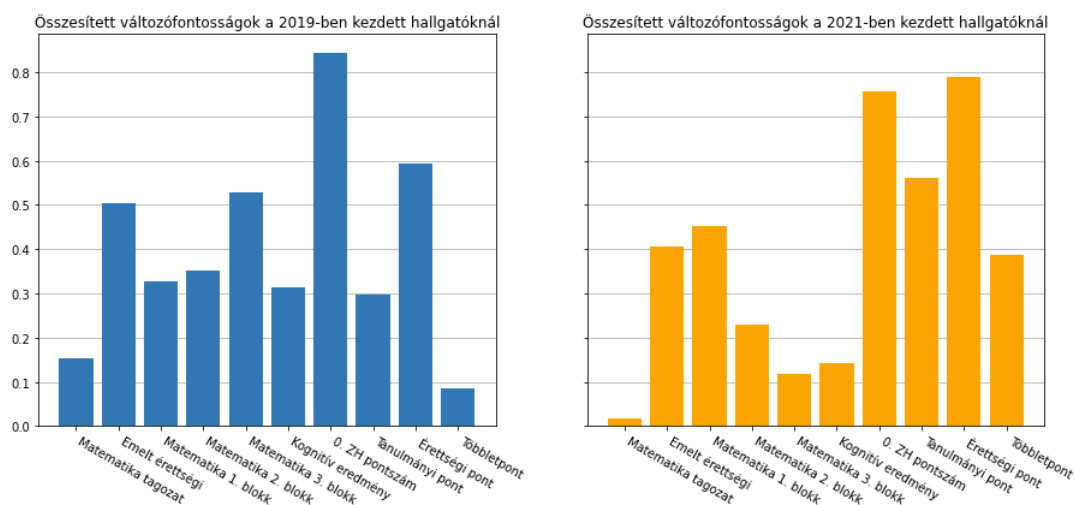
A 3. táblázat a 2021-es adatokon optimalizált algoritmusok eredményét szemlélteti az előző ábráival megegyező metodológia szerint. Ezen adathalmazon az osztályozók teljesítménye jobbnak mondható, mint a 2019-es adatsoron, átlagosan a legeredményesebb algoritmusnak a Naive Bayes nevezhető, amely a szakonkénti 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 értek el. A 2019-es eredményekkel ellentétben a 2021-es adatokon a 2 csoport modellek értékei jobbak, mint a 3 csoport modelleké, ugyanakkor nem szignifikánsan.

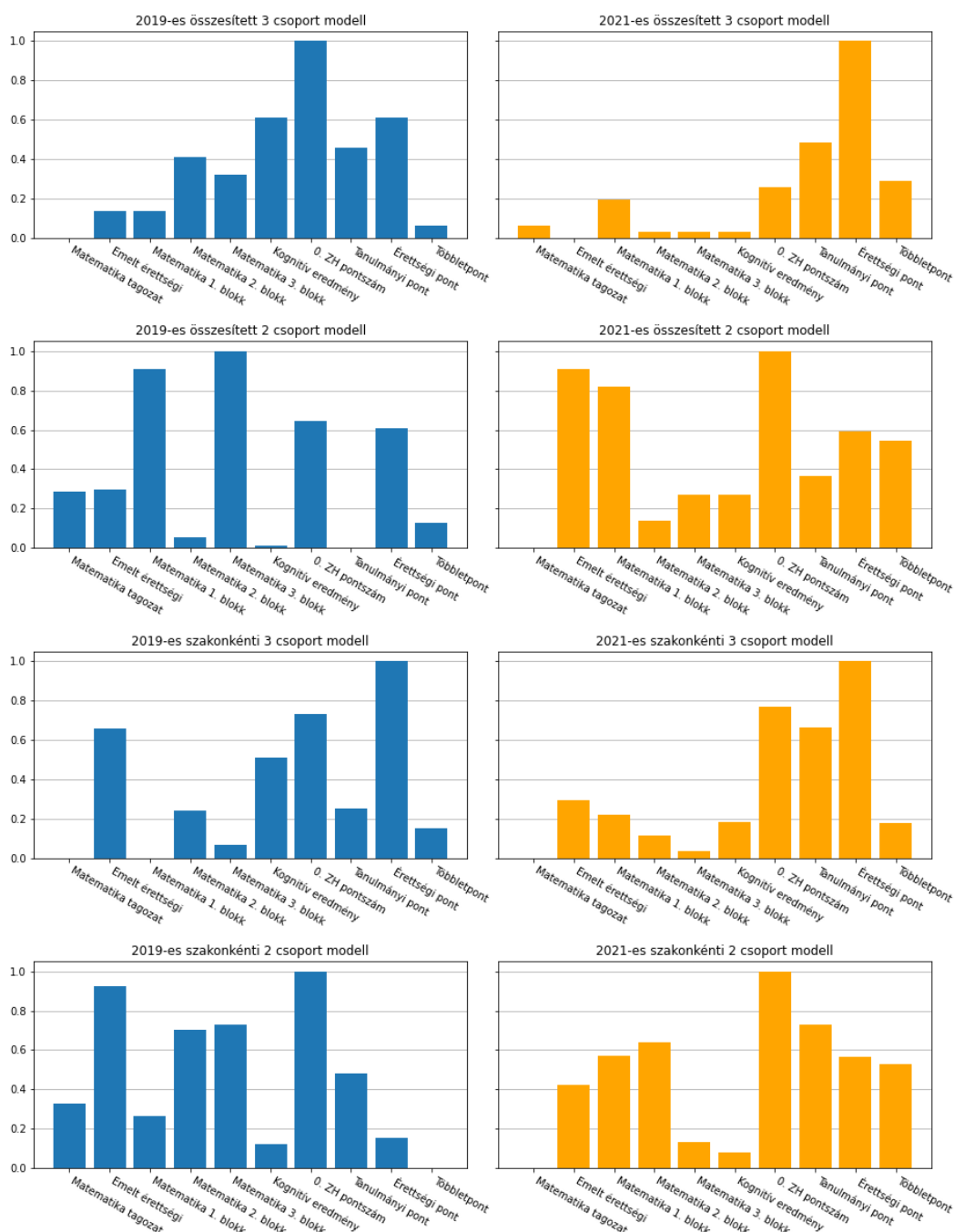
		Osztályozó algoritmusok					
		Grad.	Boos.	Naive B.	Log. reg.	SVM	Lin. reg.
3 csoport	Összesített	2 PC	54.24	80.39	76.29	78.81	60.92
		4 PC	54.89	72.05	72.13	75.79	70.40
		6 PC	66.24	76.44	70.04	76.94	70.62
		8 PC	63.00	75.29	68.18	73.06	62.07
	Szakonként	2 PC	64.84	81.89	59.23	64.77	49.35
		4 PC	55.55	83.07	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	83.24	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	81.53
		8 PC	68.32	77.84	77.41	69.75	79.68

3. táblázat. A 2021-es adatsor eredményei

A két év modelljeinek összesített változófontosságait a 26. ábra, míg az egyes modellekhez tartozó attribútumszignifikanciákat a 27. ábra szemlélteti.



26. ábra. Az változók összesített fontossága a két évben

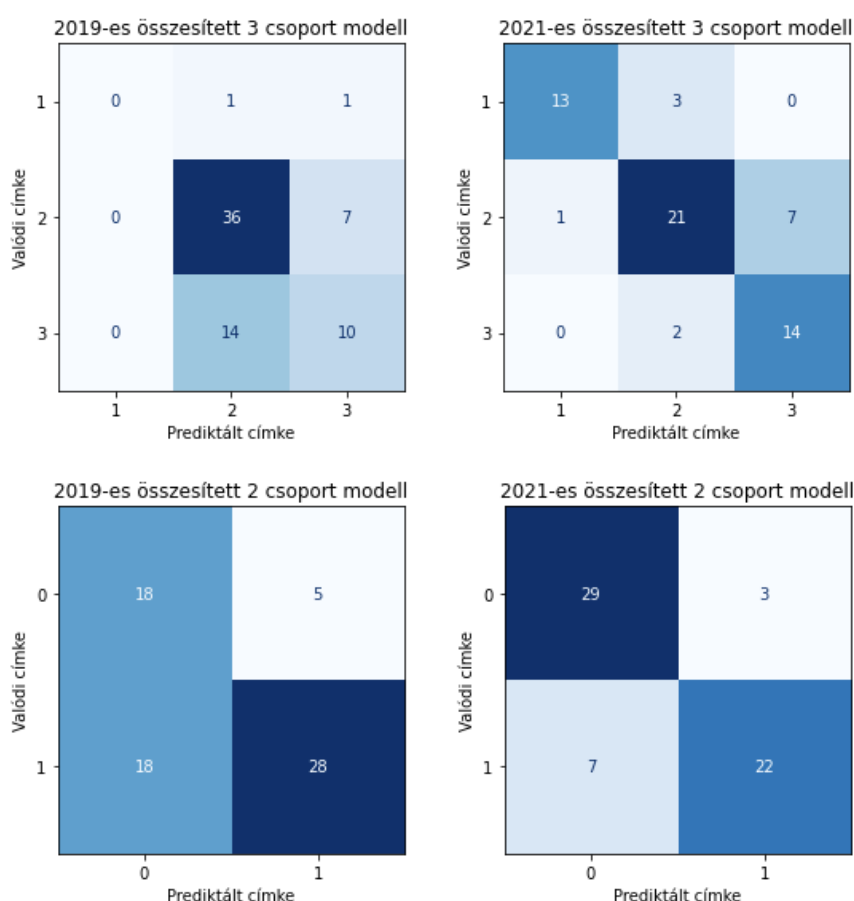


27. ábra. Az attribútumok prediktív ereje a legjobban teljesítő algoritmusoknál

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. Az összesített ábrán az egyes évek legfontosabb attribútumai a megfelelő évek különböző modelljein is többnyire jelentős szignifikanciával bírnak, a többi változó fontossága viszont modellenként eltérő. A 0.ZH pontszám illetve az érettségi pont prediktív ereje mindkét évben kiemelkedő. A 2019-es évben az emelt matematika érettségi megléte és a kognitív teszten elért eredmény is nem elhanyagolható szignifikan-

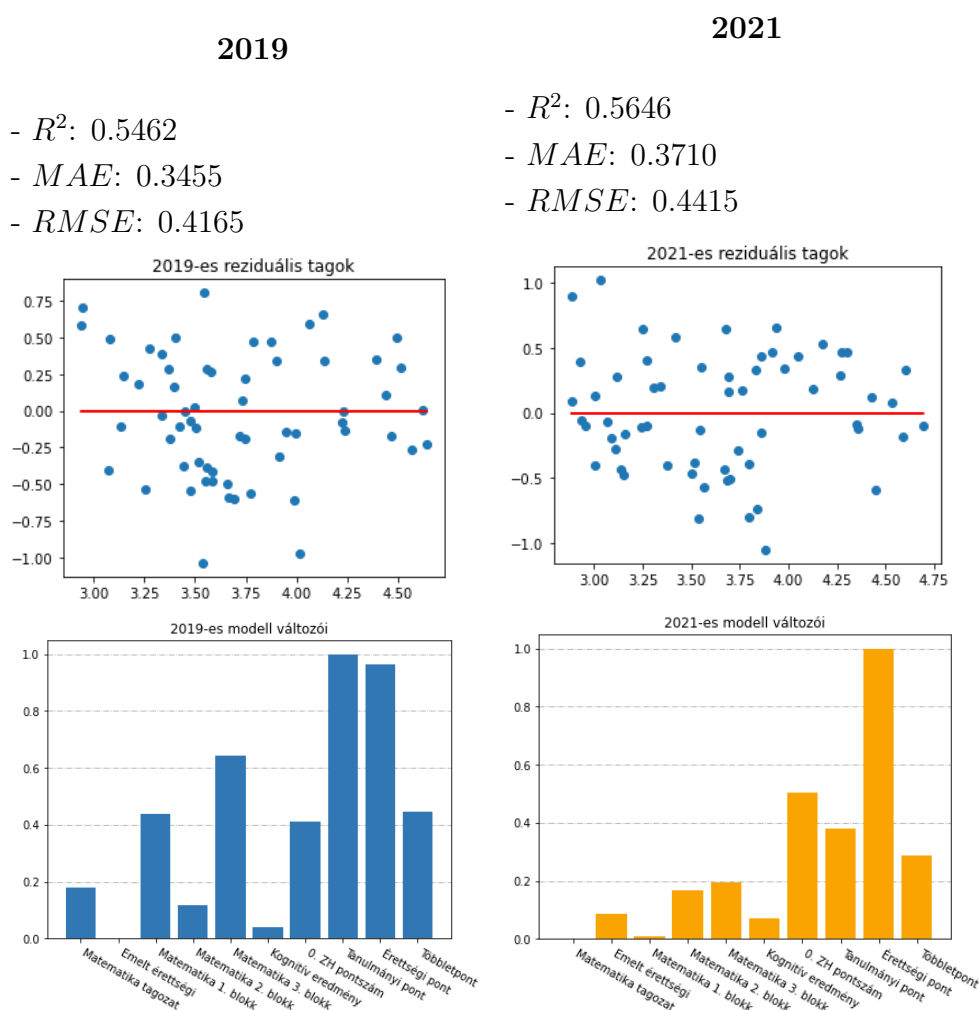
ciával bírt, viszont 2021-ben ezen tényezők prediktáló ereje csökkent, ugyanakkor az elért tanulmányi- és többletpontok jelentősége a matematika jegyre nézve közel kétszeresére nőtt.

A félreosztályozások jellegéről több információval bírnak a *tévesztési mátrixok*, amelyek a prediktált és valódi címke vonatkozásában ábrázolják az egyes adatrekordok számát. A 28. ábrán az összesített modelleknél legjobban teljesítő algoritmusok tévesztési mátrixai láthatóak, ahol a 3 csoport modellnél a 3, 2, 1 címkék rendre az $\{5,4\}$, $\{3,2\}$, $\{1\}$ osztályokat, míg a 2 csoport modellnél az 1 és 0 címkék pedig az $\{5,4,3\}$ és $\{2,1\}$ osztályokat reprezentálják. Az ábrán jól látható módon míg 2021-ben a félreosztályozás mértékében az egyes osztályok között nincsenek számottevő különbségek, addig 2019-ben a 3 csoport modellnél a 4-5-ös osztály nagyobb része 2-3-as osztályba lett besorolva, valamint a 2 csoport modell esetén magas a hamisan lemorzsolódásnak osztályzott rekordok aránya. Ez azt mutatja, hogy az illesztett modellek sok esetben alulbecsülik a hallgatók teljesítményét, amelynek az egyik oka az lehet, hogy az osztályok között vékonyak a határok a bemeneti adatok tekintetében.



28. ábra. Tévesztési mátrixok az összesített modellek esetén

A **kumulált átlag** lineáris regresszióval való prediktálásának eredményei (statisztikák, reziduális szórásdiagramok és az egyes változók prediktív ereje egymáshoz viszonyítva) a 30. ábrán láthatóak.



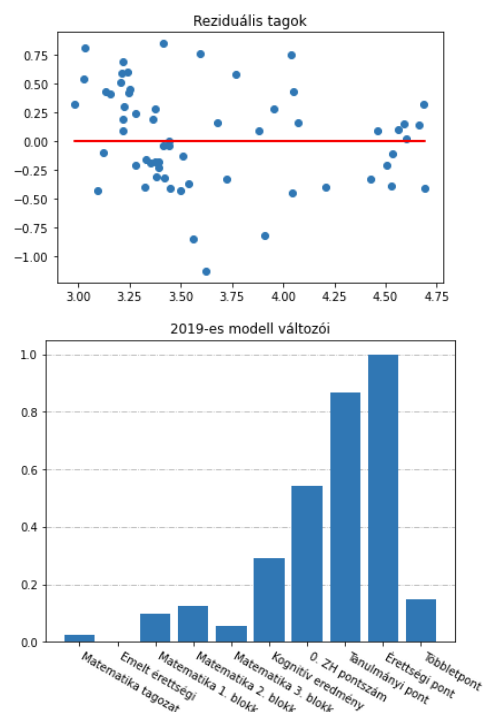
29. ábra. Lineáris regresszió eredménymutatói a két évben

Statisztikák tekintetében mindkét évben hasonló eredményeket kaptunk, amely azt mutatja, hogy a két évben hasonló hatékonysággal lehet a kumulált átlag eloszlását modellezni lineáris modellekkel, ugyanakkor csak közepes hatékonysággal. Ami kiemelendő, az a reziduális tagok eloszlása, amelyek mindkét évben hasonló alakzatot vesz fel, bár 2019-ben kicsivel nagyobb szórással. A változók prediktív erejét tekintve itt is fennáll az az érdemjegycsoportok prediktálásánál megállapított jelenség, miszerint a kognitív teszt matematika részén elért eredmény 2019-ben magas szignifikanciával bírt, azonban 2021-ben ez a szignifikancia drasztikusan csökkent. Az érettségi pont mindkét évben igen fontos determináló tényező, ugyanakkor 2019-ben a hozott tanulmányi pontok is nagy szerepet játszanak az első féléves kumulált átlag meghatározásában.

A Gradeint Tree Boosting-gal való előrejelzés eredményei a lineáris regressziónál használt struktúra szerint a 30. ábrán láthatóak.

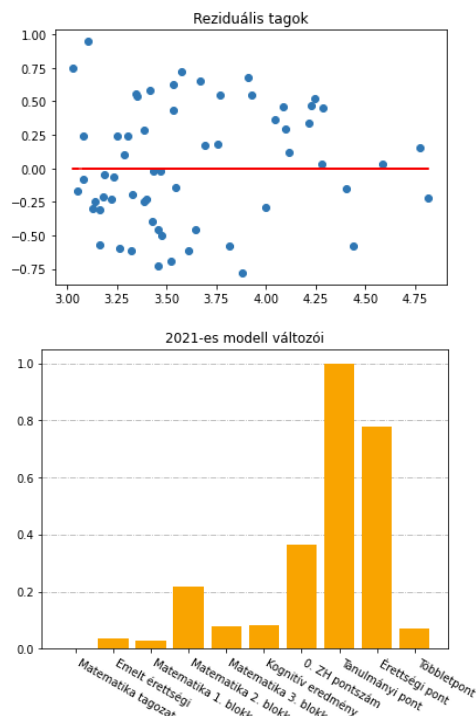
2019

- R^2 : 0.7485
- MAE : 0.3059
- $RMSE$: 0.4253



2021

- R^2 : 0.4696
- MAE : 0.3770
- $RMSE$: 0.4407



30. ábra. Gradient Tree Boosting eredménymutatói a két évben

GBT-t alkalmazva a 2019-es modell statisztikái drasztikusan jobbak, mint a lineáris regressziónál számított értékek, ugyanakkor a 2021-es modell hatásfoka valamelyest csökkent. A reziduális tagok szórásképe a két évben igencsak eltérő. Ami megjegyzendő, hogy az egyes attribútumok egymáshoz viszonyított prediktív ereje a két évben közel azonos, a két legfontosabb változó a tanulmányi és érettségi pont.

7. Diszkusszió, következtetések

A matematika érdemjegyek prediktálásánál a legjobb modellekkel is csak 80-85% közötti hatásfok érhető el, amely már jelentősen jobb, mint egy véletlen osztályzó által adott 33%-os illetve 50%-os eredmény, de még bizonyosan tovább növelhető. A teljesítmény több adatponttal és több, új bemeneti változó (középiskola földrajzi lokációja; nem; életvitellel kapcsolatos adatok stb.) vizsgálatával mindenképpen javítható lenne, ugyanakkor az új érettségi rendszer 2024-re esedékes bevezetésével érdemes ezek mellett új modellezési struktúrákat is kialakítani. Egy másik megoldás a teljesítmény növelésére egy Ensemble modell lenne, amely az egyes kategóriákban legjobb teljesítő, előre meghatározott számú algoritmus eredményét vetné össze és súlyozná az algoritmusok hatásfokával, majd ezek alapján adna egy végső predikciót, ugyanakkor ez több időt és magasabb szintű optimalizációt venne igénybe. A bemeneti változók tekintetében a legfontosabb észrevétel, hogy 2021-ben drasztikusan megnőtt a felvételi pontszám jelentősége a matematika érdemjegyre nézve, viszont a kognitív készségek érdemjegyre gyakorolt hatásaiban csökkentek a különbségek a hallgatók között.

A kumulált átlag predikciójánál látható, hogy a két legjobban domináló tényező mindkét éven a tanulmányi és az érettségi pont, a többi attribútum pedig változó, de többnyire elhanyagolható szignifikanciával bír. Az is észrevehető, hogy egy "black box" modell, mint a Gradient Tree Boosting algoritmus, sokkal több információt tud kinyerni az érettségi és tanulmányi pontokból a kumulált átlagra nézve, mint a lineáris regresszió, így további modellezések során célszerű az előbbit használni, esetleg más ensemble módszereket alkalmazni. Ugyanakkor a lineáris regressziót érdemes lehet kibővíteni harmad- vagy negyedfokú polinomiális regresszióvá, vagy kombinálni az osztályozófüggvényt valamilyen trigonometrikus függvénnyel, hisz a lineáris regressziónál kapott két szórás kép hasonló görbét ír le, s így pontosabb eredményekhez juthatunk mindkét évben. Természetesen a teljesítmény több adatponttal és több változóval itt is potenciálisan tovább javítható lenne.

A legfontosabb kérdés az, hogy ezeket az eredményeket milyen módon és mennyire etikus formában lehet felhasználni. Korábban már készültek kutatások, amelyek a magyar felsőoktatási felvételi rendszerhez használt bemeneti adatok értelmezhetőségének javítására irányultak a későbbi egyetemi teljesítmény ismeretében. (cite Molontay)

8. Összefoglalás

Hivatkozások

- [1] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 1189-1232.
- [2] Jovial Niyogisubizo, Lyuchao Liao, Eric Nziyumva, Evariste Murwanashyaka, Pierre Claver Nshimyumukiza. Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization. *Computers and Education: Artificial Intelligence*, Volume 3, 100066, 2022.
- [3] Marina Segura, Jorge Mello, Adolfo Hernandez. (2022). Machine Learning Prediction of University Student Dropout: Does Preference Play a Key Role?. *Mathematics*, Volume 10, 3359, 2022.
- [4] Ahajjam Tarik, Haidar Aissa, Farhaoui Yousef. Artificial Intelligence and Machine Learning to Predict Student Performance during the COVID-19. *Procedia Computer Science*, 184:835-840, 2021.
- [5] D. K. Dake, D. D. Essel, J. E. Agbodaze. Using Machine Learning to Predict Students' Academic Performance During Covid-19. *2021 International Conference on Computing, Computational Modelling and Applications (ICCMA)*, pages 9-15, 2021.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011.
- [7] C. Cortes, V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.
- [8] Roland Molontay. *Lecture notes in Introduction to Data Science*. Budapest University of Technology and Economics, 2021.
- [9] Bolla Marianna, Krámlí András. *Statisztikai következtetések elmélete*. Elméleti matematika. Typotex Kft., 2012.
- [10] Jonathon Shlens. A Tutorial on Principal Component Analysis. *CoRR*, abs/1404.1100, 2014.