Eötvös Loránd Tudományegyetem
Természettudományi Kar

Mobiltelefon-alkalmazások naplófájljainak elemzése

Szakdolgozat

Császár Kristóf

Matematika alapszak
Matematikai elemző szakirány

Témavezető:

Lukács András
Számítógéptudományi Tanszék

Budapest

2017

Tartalomjegyzék

Ábrák jegyzéke

Táblázatok jegyzéke

1. Bevezetés

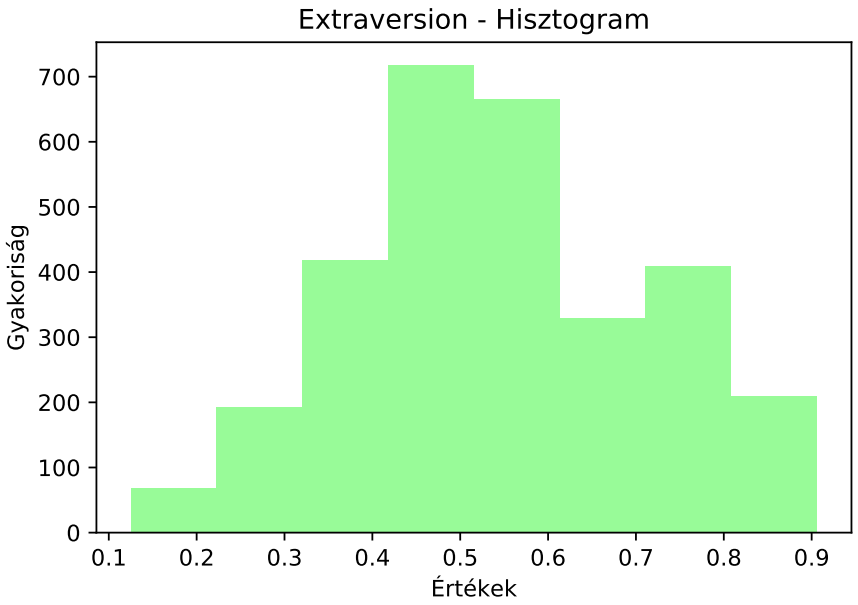
A mai világban a mobiltelefon elengedhetetlen a napi rutin során. Elsődleges rendeltetését - az elérhetőség biztosítását - rég túlnőtte, ez sokszor háttérbe is szorul. A technika fejlődésével a telefonok olyan plusz funkciókat kaptak, melyek segíthetnek felmérések, kutatások lebonyolításában. Az okoseszközök egyik fő tulajdonsága az alkalmazás-centrikusság, áruházanként több millió applikáció áll rendelkezésre, a legtöbb ingyenesen. Ilyen alkalmazásokat ma már bárki írhat, ezek segítségével felméréseket is készíthetünk. Diplomamunkámban is egy ilyen applikációval gyűjtött adathalmazt fogok feldolgozni, célomnak azt tűztem ki, megvizsgálom, hogy a telefonok naplófájljaiból kinyerhető információk alapján meghatározható-e az adott ember személyisége, tehát jellemez-e minket és egyéniségünket telefonhasználatunk.

A személyiség nem egyértelmű, egzakt fogalom, mindenképp pontosítanunk kell, hogy mire is gondolunk. Ahhoz, hogy mérhető tulajdonságokat kapjunk, egy olyan szemléletet választottam, mely dimenzionálisan közelíti meg a személyiséget, öt faktorba sorolva tulajdonságainkat. Ezek a faktorok már mérhetőek olyan tekintetben, hogy meghatározható, egy-egy adott faktor mennyire igaz az adott emberre [1].

A modellezéshez adatbányászati eszközöket fogok felhasználni. Az ilyen feladatoknál, ahol egy ismeretlen paramétert kell becsülni az eddigi ismereteink, adataink segítségével, és ezeken az adatokon valamilyen modell megalkotásával, prediktív modellezésnek nevezzük. Ennek két fő típusa, amit az ismeretlen paraméter fajtája alapján el kell különítenünk, a klasszifikáció, illetve a regresszió. A klasszifikáció olyan címkézett adatokon értelmezhető, amikor a címkék lehetséges értékei diszkrétek, az összes lehetséges értéket ismerjük. A regresszióról akkor beszélünk, ha a paraméterértékek folytonosak. Jelen feladatban a regressziót fogom segítségül hívni.

1.1. Bővebben az öt faktoros személyiségmodellről

A modellt a ma is használt formájában először Ernest Tupes és Raymond Christal használta, melynek mérésére Goldberg 1993-ban fejlesztett ki egy 44 kérdésből álló tesztet. Az öt faktort nehéz egy-egy szóval jellemezni, így magyarra fordítani még nehezebb, általában több, jellemző tulajdonsággal írjuk körül őket. Magyarul



1. ábra. Az extraverzióhoz tartozó hisztogram

így hangoznak (a dolgozatban az egyszerűség és egyértelműség kedvéért az eredeti, angol nyelvű változatokat fogom használni):

- Extraverzió - Extraversion
- Barátságosság - Agreeableness
- Lelkiismeretesség - Conscientiousness
- Érzelmi stabilitás - Neuroticism
- Kultúra/Intellektus - Openness

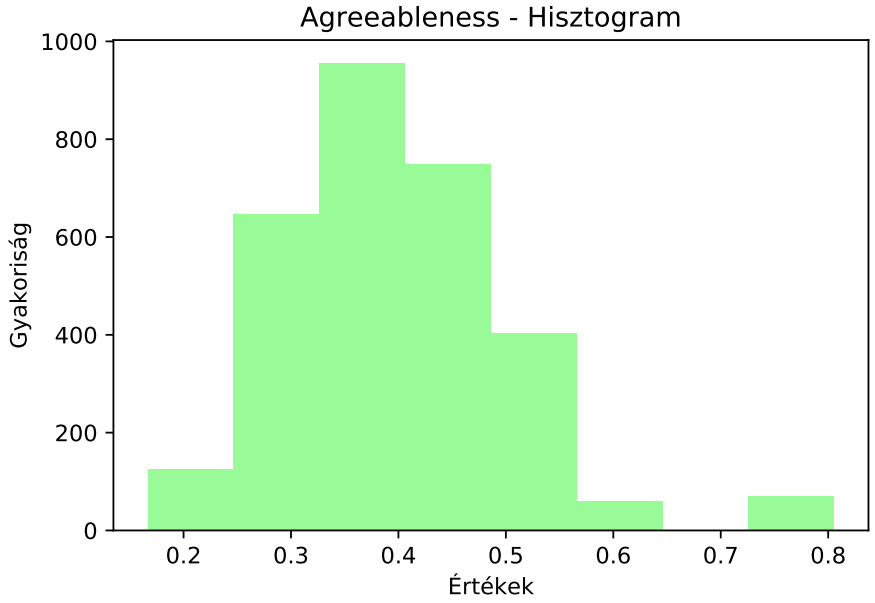
Azonban az öt kifejezés sokkal többet hordoz magában ennél [2].

Az extravertált emberre jellemző, hogy magabiztos, domináns, általában társasági személy, aki szabadon ki tudja fejezni impulzusait. A társasági eseményektől feltöltődik energiával, szeret a figyelem középpontjában lenni, széles baráti és ismerettségi köre van, szívesen ismer meg új embereket. Ennek ellenpólusa a visszahú-

zódó, félénk, önbizalomhiányos személy. Társasági események alkalmával kimerül, nehezen kezdeményez beszélgetést, nem szereti a csevegést.

A barátságosság leginkább az emberi kapcsolatok fenntartásának képességét méri, emellett a gondoskodást és az érzelmi támogatást is. Jellemző a jó együttműködő képesség, és egyéb proszociális viselkedés. Szembeállítható az ellenséges, önző személyiség, az érzelmi hidegség és bizalmatlanság.

A lelkiismeretesség kissé félrevezető magyarul, az eredeti jelentése az átgondolt, mérsékelt, célirányos viselkedés. Aki erősen lelkiismeretes, arra jellemző, hogy sok időt tölt a dolgok előkészítésével, a fontos feladatokat helyezi előtérbe, ezekkel azonnal végez, ügyel a részletekre, ütemezés szerint könnyebben végzi munkáját. Aki kevésbé lelkiismeretes, nem ügyel a dolgokra, rendetlenség van körülötte, összekeveri a dolgokat, halogatja a fontos dolgokat, az egyszerűeket helyezi előtérbe, nem tudja időben befejezni a feladatokat.

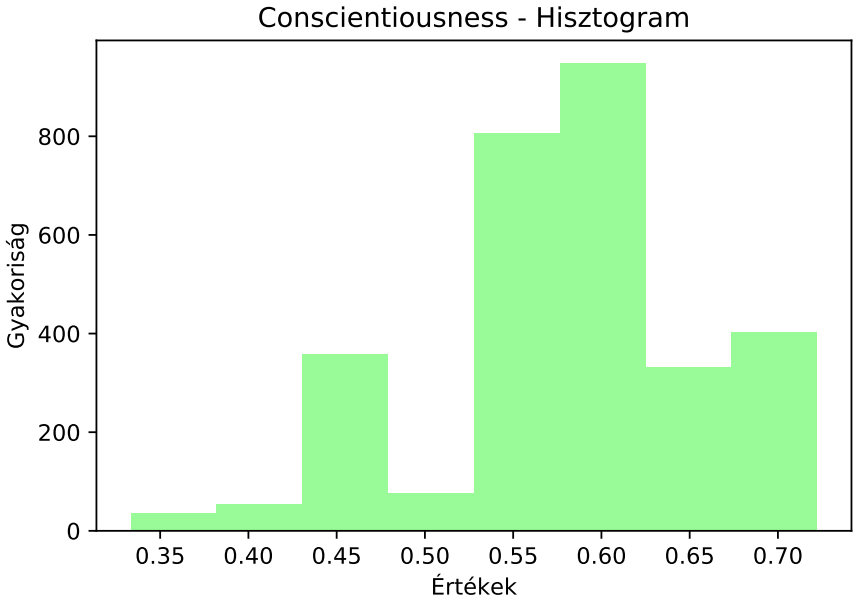


2. ábra. A barátságossághoz tartozó hisztogram

Az érzelmi stabilitás is komplex jelentéssel bír, bár itt a magyar megfelelő pontos. Akinél magas ez a faktor, érzelmileg instabil, sok stressz éri, sokat aggódik, könnyen felhúzza magát, ingadozik a hangulata. Aki kevésbé instabil, ritkán szomorú és depressziós, nem aggódja túl a dolgokat, kipihent.

Utolsóként pedig a kultúra, intellektus, mely az új dolgokra való nyitottságot, a

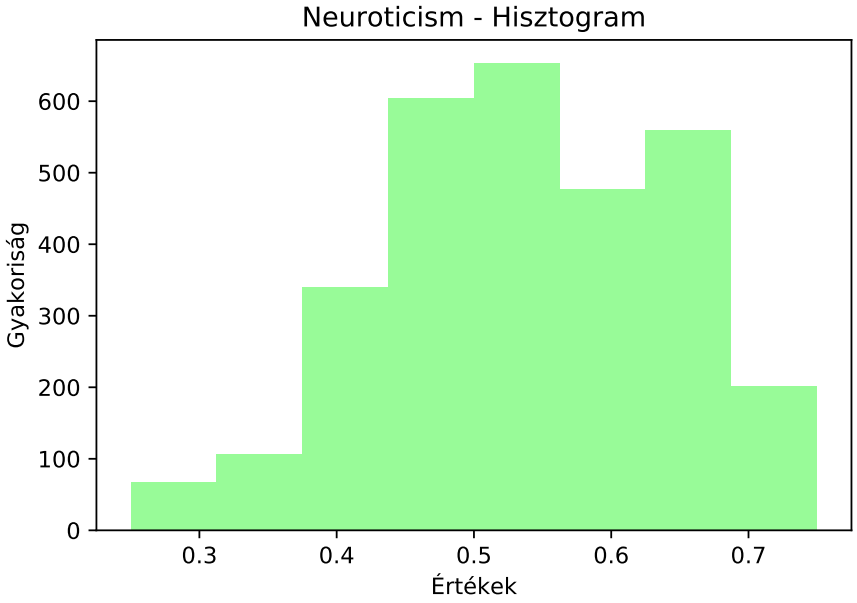
kreativitást jelenti. Akinél ez jellemző, az szereti az új kihívásokat, sokszor gondolkozik elvont dolgokon. Akire nem jellemző, az nem szereti a változást, elutasítja az új dolgokat, kis képzelőerővel bír, nem szeret elgondolkozni bonyolult koncepciókon.



3. ábra. A lelkiismeretességhez tartozó hisztogram

A kutatás szerint ezek a faktorok viszonylag stabilak, azonban pár évente változnak bizonyos mértékben, attól függően, hogy milyen fontos életeseményeken megyünk keresztül. Ahogy öregszünk egyre több tapasztalatunk van az életről, ez megnyilvánul személyiségünkben is. Idővel kevésbé leszünk extravertáltak, kevésbé vagyunk érzelmileg instabilak, és kevésbé nyitunk új dolgok felé, inkább az állandóra vágyunk, azonban egyre inkább jellemző a barátságosság és a lelkiismeretesség.

A célváltozók eloszlása alapvetően normális eloszlást kellene, hogy kövessen, azonban látható, hogy minden változónál előfordul ugrás az értékekben. Ez annak tudható be, hogy kevés emberrel végezték a kísérletet, és emiatt tozultak az eloszlásfüggvények. A lelkiismeretesség és kultúra, intellektus kivételével jól megfigyelhetők a normális eloszlásra utaló jegyek.

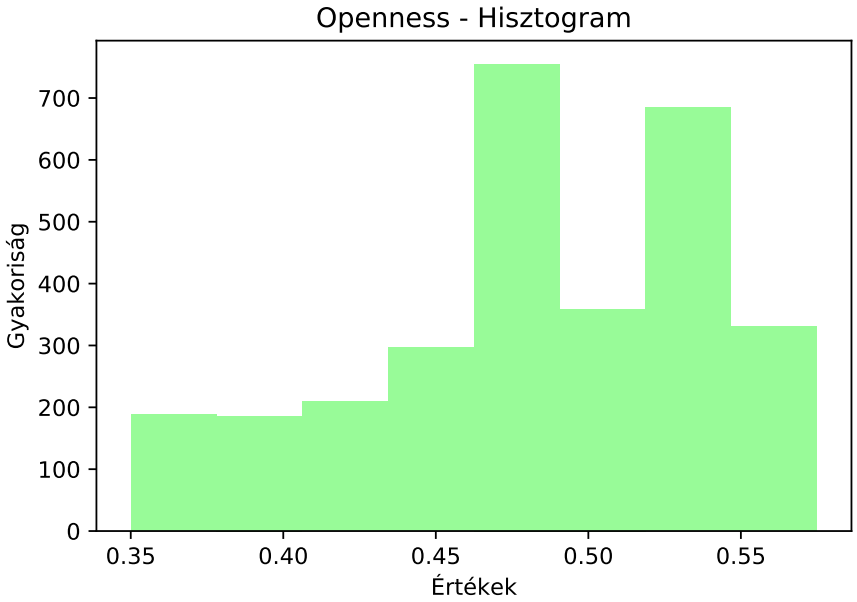


4. ábra. Az érzelmi stabilitáshoz tartozó hisztogram

1.2. Motiváció

A rendelkezésünkre álló adatok - mobiltelefon-alkalmazások naplófájljai - olyan információkat tartalmaznak, melyeket számos területen felhasználhatunk. Egy kutatás szerint [3] a telefonon használt alkalmazások sokat elmondanak rólunk: nemünket, életkorunkat, fő személyiségjegyeinket - például a korábban már említett öt faktoros modellt. Ezeknek az adatoknak a birtokában a cégek a marketing tevékenységüket sokkal kifinomultabban és pontosabban tudnák végezni például alkalmazásokon belüli hirdetésekkel, vagy az alkalmazás-áruházak ezek alapján tudnak ajánlani hasonló alkalmazásokat, melyeket hasznosnak találhatunk - ezt meg is teszik, sokszor találkozunk ilyen ajánlásokkal. Mi most nemcsak az alkalmazások alapján próbáljuk meghatározni az öt faktort, hanem felhasználunk egyéb naplózott információkat is. A marketing témájú üzenetek előkészítéséhez előszeretettel használják az öt nagy faktort. Ezek alapján az embereket körülbelül tíz-tizenöt csoportba osztják, attól függően, hogy hányan vannak, mekkora a célközönség, és minden csoportnak külön írják meg a levelet, aminek a lényege ugyan az, a termék nem változik, csak a célközönséget másik oldalról próbálják megragadni. Ezekkel a csoportra szóló megkeresésekkel lényegesen növelik annak valószínűségét, hogy

felkeltik az érdeklődést adott termék iránt.



5. ábra. A kultúra, intellektushoz tartozó hisztogram

Egy további alkalmazási terület, az emberi erőforrás-kiválasztás illetve a pályaválasztási tanácsadás, hiszen személyiségünk határozza meg érdeklődésünket is. Ezekhez hosszú, sok kérdésből álló tesztet kell kitölteni, a minél pontosabb eredmény elérésének édekében. Ezeket aztán ki kell értékelni, így az egész folyamat akár több órát, napot is igénybe vehet. Ha sikerül egy olyan modellt találni, mely aránylag pontosan meg tudja határozni a faktorokat, akkor ezzel rengeteg időt és energiát tudnánk spórolni, és ki lehetne váltani ezeket a több szempontból is költséges teszteket.

Az emberi erőforrásoknál maradva napjainkban a nagyobb munkahelyeken évente ismétlődő felméréssel vizsgálják a teljesítményt, például 360 fokos értékeléssel. Ezeknek a lényege, hogy az értékelt személyt felettesei, beosztottjai és azonos szinten dolgozó kollegái által értékeltetik, innen ered a kifejezés is. Ebben a felmérésben fontos, hogy a vezetők milyen értékelést kapnak, és ez szoros összefüggésben van az öt személyiségjegyükkel [4]. A kutatás szerint 22 kisebb faktort lehet elkülöníteni, amikkel a vezetők rendelkezhetnek, ezeket pedig be lehet sorolni az öt nagy faktorba, és ez alapján jól meghatározhatóak egymásból.

Utolsó példának pedig tekintsük a pszichológiát - átlátni egy ember személyét,

megérteni a gondolatait nem egyszerű feladat, ebben is segítséget nyújthat, ha például a páciens telefonján használt alkalmazások alapján egyből kiderülnek a fontos személyiségjegyek, akár olyan részletek is előkerülhetnek, melyet tudatosan nem árulnánk el magunkról, viszont tudat alatt, cselekedeteink, vagy a használt alkalmazások alapján kiderülnek.

2. Az adatok

Az adatokat szolgáltató eredeti kutatás célja elsősorban az volt, hogy megértsék, miért teljesítenek jobban bizonyos diákok a másiknál, vagy mik vezetnek ahhoz, hogy valaki abbahagyja a tanulmányait, felismerhető e valamilyen mintázat a diákok életritmusában, ami alapján előre látható a viselkedés. Dolgozatomban a Dartmouth Egyetem egy kutatásának részeként összegyűjtött adatokat fogom felhasználni [5]. Sokszor a tanulók életében a stressz és a feszültség rejtve marad. Valójában az egyetemi dolgozók, a tanárok keveset tudnak a diákjaikról az iskolai környezeten kívül. A tanulók valószínűleg ismerik saját magukat, korlátaikat, viszont hallgatótársaikról már kevesebbet tudnak. Annak érdekében, hogy kicsit jobban megismerhessék a diákéletet, kifejlesztettek egy alkalmazást, a StudentLife-ot, mely a telefon érzékelői segítségével automatikusan rögzít számos tevékenységet. Miért égnek ki a diákok, halasztanak féléveket, vagy éppen miért hagyják abba tanulmányaikat? Mennyire befolyásolja a stressz, a hangulat, a munkaterhek, a társasági élet, az alvásmennyiség és a mentális egészség a teljesítőképességet? Ezekre a kérdésekre is keresi a választ a tanulmány, melyet 48 diákkal végeztek nagyjából 10 héten keresztül. Minden résztvevőnek ki kellett tölteni a kutatás előtt és után bizonyos teszteket, köztük az öt faktoros személyiségmodell meghatározásához használt, 44 kérdésből álló tesztet is.

Személyiségi jogok miatt minden olyan adatot eltávolítottak, illetve kódoltak a forrásban, amiből felismerhető lett volna bármelyik résztvevő. Például a csatlakozott Bluetooth vagy Wifi nevek illetve címek is eltávolításra kerültek - voltak, akik saját nevüket használták az eszközök nevében. A böngészési előzmények is eltávolításra kerültek hasonló megfontolásból.

2.1. Hardveres és szoftveres környezet

A dolgozat, illetve a programkódok és a modellek létrehozása során egy egyszerű notebook volt segítségemre, melyben 2 Gigabájt rendszermemória és 60 Gigabájt szabad tárhely állt rendelkezésre. A számítógépen a Windows 10 operációs rendszer Home verziójának 64 bites típusa volt telepítve.

Elemzési eszközömnek a Python programozási nyelvet választottam, mert ezt találtam a legmegfelelőbb az adatbányaszati modellek megalkotásához, mivel nagy támogatottsággal bír ilyen típusú kutatásoknál [6]. A nyelv 3.6.1 verzióját használtam, a dolgozat megírásakor ez volt a legfrissebb. A Python nyelvet az Anaconda nyílt forráskódú platformon keresztül használtam, mely szerteágazó eszközkészlettel rendelkezik, matematikai és adatbányászati témákban is [7]. Ez mindenki számára hozzáférhető, és nem igényel mást, mint egy böngészőt, ezen belül futtatható. A böngészőm a Google Chrome volt. Az Anaconda az IPython és ezen belül a Jupyter notebook segítségével biztosítja a böngészőben futtatható környezetet [8].

A kódok futása gyorsnak mondható, a modellek tanításának egy-egy köre átlagosan tíz percig tartott, míg a teljes adathalmaz átkonvertálása felhasználható formátumba, körülbelül egy órát vett igénybe. Ezen kódok megírása nagyjából 2 hónapot vett igénybe, a többszöri tesztelések és próbafutások miatt.

A nyelv előnye az egyszerűsége, mind tanulhatóságában, mint használhatóságában, könnyű benne programozni, és rendkívül sokrétű az eszközkészlete, rengeteg csomaggal bővíthető, mint a Numpy, mely a többdimenziós mátrixokat is jól kezeli, a Pandas szintén jól kezeli a nagy adathalmazokat és ezeket tudja elemezni, a Mathplotlib a hisztogramok, illetve a kétdimenziós ábrázoláshoz, a scikit-learn pedig a modellek algoritmusait tartalmazza [9][10][11][12][13].

A fájlok CSV kiterjesztésűek, ezeket programmal könnyű feldolgozni, a legtöbb programozási nyelven van automatikus feldolgozó mechanizmus, a kihívás ez esetben a széttagoltság, mivel minden tanuló minden különböző adata külön fájlban van tárolva, így ezeket egyesével kell feldolgozni is. Ezek a fájlok általában 30 ezer sor körül vannak, de előfordul olyan is, mely több százezer soros. Ezeknél kézi megoldás nem lehetséges, mindenképp valamilyen algoritmikai megoldás kell a beolvasáshoz. A fájlok mappákba vannak rendezve, magyarázó változónként, tehát célszerű így feldolgozni őket. Az Anaconda egyik csomagja, nevezetesen a Glob képes azt a funkciót ellátni, hogy egy adott mappaszerkezeten belül kilistázza az összes előre megadott kiterjesztésű fájlt, akár úgy is, hogy megadjuk, milyen szövegrészletet

tartalmazzon a neve.

Az adatok tanulónként és témánként (attribútumonként) állnak rendelkezésünkre, azok a magyarázó és célváltozók, melyeket felhasználtam a modellalkotás során a következők:

- Bizonyos alkalmazások használatának darabszáma: Gmail, Chrome, Térkép, Kamera, Beállítások
- Küldött és fogadott SMS-ek száma
- Kezdeményezett és fogadott hívások hossza
- Naptárban vezetett események száma
- Helyzet, pozíció
- Mennyi ideig volt lezárva összesen a telefon
- Csatlakoztatott eszközök darabszáma
- Az öt személyiségfaktor

2.2. Elemzés, tisztítás, feldolgozás

Azt vettem észre az adatok áttekintésekor, hogy vannak olyan sorok, melyek teljesen üresek eltekintve az időbélyegzőtől, mely minden sort jellemez (az összes magyarázó adatunknál fontos az időbelisége). Ezeket a sorokat kiszűrtem, csak az adatokat tartalmazó sorokkal foglalkoztam.

Előfordult még, hogy adott sorok akár többször is előfordultak, mint duplikátum, ezeket szintén töröltem, mivel egy ember egy pillanaton belül nem tud egyszerre például több alkalmazást elindítani.

A mérések négyféle adatot tartalmaznak: érzékelt adatokat, melyeket a telefon érzékelői gyűjtöttek, naplózott eseményeket, mint például a hívások, üzenetek, illetve, hogy a telefonnal milyen egyéb interakciókat végeztek, elő- és utófelmérések, amiket a résztvevők töltöttek ki, és végül tanulmányi adatok, jegyek átlaga.

A 44 kérdésből összeállítottam az öt faktort, ehhez a megadott értékelést használtam. A faktorokhoz bizonyos kérdések tartoznak, ezek összege adja, hogy mennyire jellemző a tulajdonság az egyénre. Ezt normáltam, hogy minden tulajdonság 0 és 1 közötti szám legyen, így átláthatóbb százalékos formában. Ennek legenerálását csináltam meg először, egy külön fájlba kiírva, majd ebből olvastam be később az adattábla létrehozásakor. A tábla összeállítását több lépcsőben végeztem, miután elkészültek a faktorok, elkezdem feldolgozni a többi attribútumot. Mivel összesen 48 ember adatai állnak rendelkezésre, így a 10 heti adatot heti bontásban aggregál-

	call	sms	calendar	conversation	phonelock
call	1.000000	0.998374	0.089745	0.305533	0.364086
sms	0.998374	1.000000	0.088874	0.304185	0.363689
calendar	0.089745	0.088874	1.000000	0.131137	0.047087
conversation	0.305533	0.304185	0.131137	1.000000	0.363317
phonelock	0.364086	0.363689	0.047087	0.363317	1.000000

1. táblázat. A magyarázó változók korrelációs mátrixa - részlet

tam, ezzel elősegítve a modell pontosítását. Így keletkezett összesen 3011 sor adat. Sok fájl feldolgozására került sor, minden attribútum és személy külön fájlban van tárolva, ezek automatikus beolvasására a Glob csomagot használtam, mely adott mappában minden fájlt fel tud dolgozni egy parancs kiadásával. Attribútumonként külön listában és szettben tárolva oldottam meg az aggregálást, így a kiírásnál csak a szettben lévő elemeket kellett megszámolni a listában, és nem kellett különféle módokon megkeresni a különböző elemeket.

Miután elkészült a modellezésre még alkalmatlan fájl, megnéztem a korrelációs mátrixot, melynek egy kisebb részletét az 1. ábrán is láthtjuk. Az SMS és Call változók korrelációja nagyobb, mint 99%, tehát nem kapunk plusz információt mindkét oszlopból, elég az egyiket használni, így ezt el is távolítottam. A magyarázó változók korrelációja a legtöbb esetben 0,2 alatti, egy-egy esetben pedig 0,4 közeli, e fölött már nincsenek értékek. Összességében az adatról elmondható, hogy a változók egymással nincsenek, vagy csak kis mértékben vannak összefüggésben.

A 2. táblázat mutatja, hogy a célváltozók egymással vett korrelációja milyen értékeket vesz fel. A barátságosság és az érzelmi stabilitás mutat közepes korrelációt, a többi csak kis mértékben korrelál. Ennek oka pszichológiai, viszont kétségkívül a két faktor nem teljesen független egymástól, hiszen egy gondoskodó ember, aki törődik másokkal, viszonylag kisebb valószínűséggel szomorú, depressziós. Ugyan így belátható, hogy az extraverzió és az érzelmi stabilitás sem függetlenek teljes mértékben, mivel egy társaságot kedvelő személy, aki magabiztos, domináns, hasonlóképp valószínűleg kevésbé depressziós. Ezek és a további összefüggések az adatok elemzésével kevésbé magyarázhatóak, így ezt nem vizsgálom tovább.

Két változó korrelációját a következő képlet szerint számítottam ki:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y},$$

	Extrav.	Agree.	Conscient.	Neurot.	Openn.
Extraversion	1.000000	0.204453	0.150842	0.354009	-0.132489
Agreeableness	0.204453	1.000000	0.084001	0.406082	-0.003877
Conscientiousness	0.150842	0.084001	1.000000	0.098793	0.029990
Neuroticism	0.354009	0.406082	0.098793	1.000000	-0.316286
Openness	-0.132489	-0.003877	0.029990	-0.316286	1.000000

2. táblázat. A célváltozók korrelációs mátrixa

ahol x és y a két változónk, $\bar{x}$ és $\bar{y}$ a változó értékeinek átlaga, s_x és s_y pedig a tapasztalati korrigált szórás.

Összességében az adatfeldolgozás után kaptam 3011 sort, melyek mindegyike egy-egy személy egy hetének összesített adatait tartalmazta. A 3. táblázattal dolgoztam a dolgozat során, az a két táblázat az adatok egy rövid részlete. Jellemzőek a kiugró értékek, mely mérési hibaként is felléphetnek, vagy előfordulhat, hogy egy személy nem engedélyezte bizonyos adatok felhasználását, de az is lehetséges, hogy csak egyszerűen nem használta az adott alkalmazást. Mivel ez utálag nem deríthető ki, így felhasználtam ezeket az adatokat is, eredeti formájukban.

id	call	cal.	conv.	phone.	gmail	settings	chrome	phone	maps	cam.	Extra.	Agree.	Cons.	Neuro.	Open.
u4425	24	0	4841	28470	72	72	72	0	72	72	0.468	0.388	0.583	0.437	0.450
u4426	24	0	15320	35280	72	72	72	0	72	72	0.468	0.388	0.583	0.437	0.450
u3030	24	0	38816	20516	0	71	71	0	71	71	0.250	0.277	0.694	0.250	0.550
u5761	24	0	25436	35595	1	0	63	72	0	1	0.718	0.361	0.611	0.468	0.475
u3024	24	1	26361	86578	0	72	72	0	72	72	0.250	0.277	0.694	0.250	0.550
u1824	24	0	33904	81907	72	72	72	0	0	72	0.531	0.638	0.555	0.562	0.475
u0821	24	0	17646	42672	71	51	51	51	51	4	0.531	0.361	0.583	0.687	0.475
u3027	24	0	49062	71207	0	72	72	0	72	72	0.250	0.277	0.694	0.250	0.550
u0019	25	1	31550	33199	67	72	72	72	72	31	0.437	0.472	0.611	0.656	0.475
u1920	22	1	39276	26225	27	57	29	0	0	32	0.812	0.527	0.555	0.656	0.500
u5024	24	0	11531	94049	72	0	72	0	0	72	0.343	0.416	0.583	0.500	0.475
u4422	22	0	2702	4095	62	1	62	0	62	9	0.468	0.388	0.583	0.437	0.450
u1830	24	0	54552	21714	72	72	72	0	0	72	0.531	0.638	0.555	0.562	0.475
u1829	24	0	11877	16071	72	64	72	0	0	72	0.531	0.638	0.555	0.562	0.475
u0013	24	1	22603	61541	65	72	72	72	0	72	0.437	0.472	0.611	0.656	0.475
u5026	24	0	9327	18196	72	0	72	0	0	72	0.343	0.416	0.583	0.500	0.475
u3028	24	1	21533	84088	0	72	72	0	72	72	0.250	0.277	0.694	0.250	0.550
u5771	24	0	0	0	72	8	72	72	0	1	0.718	0.361	0.611	0.468	0.475
u0124	24	0	39884	76222	0	50	72	0	0	66	0.468	0.416	0.638	0.500	0.525
u3221	24	1	34132	0	59	0	4	0	0	7	0.312	0.250	0.666	0.437	0.475

3. táblázat. Az adatok első pár sora

3. Modellek és eredményeik

Az adatok feldolgozását követően tudtam elkezdeni modelleket illeszteni és megjósolni a célváltozókat. Mivel a korreláció nem mutat a célváltozók között lényeges összefüggést, ezeket külön-külön vizsgáltam, így minden modellcsalád esetében öt különböző modellel dolgoztam. A kevés adatra tekintettel keresztvalidációt használtam a tanításhoz, így próbáltam meg a pontosságot és a modellek stabilitását növelni a viszonylag kisebb halmazon.

A keresztvalidáció esetén meg kell adnunk, hogy hány, körülbelül egyenlő részre ossza fel az algoritmus a halmazunkat, ennek leggyakrabban használt változata a tízrészes felbontás. Ekkor az adatból 9 részen tanul a modell, 1 részen pedig értékel, ezt minden lehetséges módon megteszi, azaz tízszer, és ezeknek az értékeléseknek az átlagát fogjuk kapni, mint a modell hibáját. Ekkor az építés így kilencvenszer ismétlődik, és tízszer értékel, ami időigényes folyamat is lehet, minél bonyolultabb modelleket építünk.

Az eddig felsoroltak alapján tehát az öt célváltozónkra külön, mindegyik esetén keresztvalidációt alkalmazva építünk modellt, ami a legegyszerűbb esetben is négyszázötven egyszerűbb komponens felépítését jelenti. Ezt muszáj optimalizálni, különben belefuthatunk olyan hosszú programfutásokba, amiket nem lehet kivárni, főleg nem otthoni körülmények között, még egy jól felszerelt notebook vagy asztali számítógép használatával sem.

Az optimalizálás során az egyik legelterjedtebb módszer a Line Search folyamat. A modellek finomhangolásához több paraméter áll rendelkezésünkre, ezek beállításához ezt a folyamatot használom, melynek lényege, hogy egyetlen futó és a többi rögzített változó mellett megtalálja a hiba lokális minimumát. Időigényét tekintve ez lineáris idejű a változókban, mivel mindegyikre egyszer kell futtatni, és minden futás után csökken a fennmaradó dimenziók száma. A folyamatot minden paraméterre megismételve találhatunk egy olyan minimumot, mely jól közelíti a globális

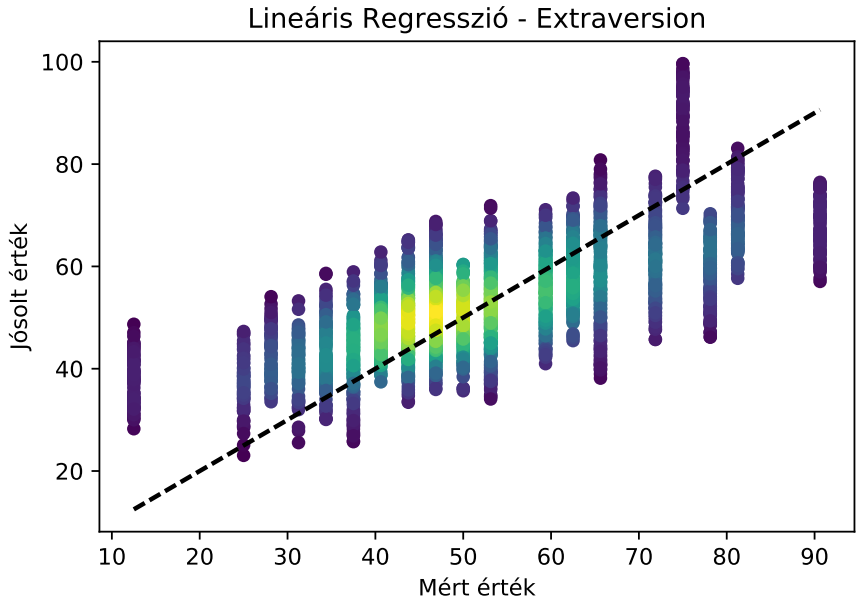
hibát. Mivel az összes lehetséges eset vizsgálata rengeteg időt venne igénybe, így ennek a módszernek a használata kihagyhatatlan.

A modellek kiértékelésére a hibájukat használtam, ami azt mutatja meg, hogy a pontos, mért értéktől mennyire térnek el az adatok. Ennek egyik módja az átlagos négyzetes eltérés, melynek képlete:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Ez a hiba a modell minőségét méri, minél pontosabb a jóslat, annál kisebb a hiba. Ha ennek a gyökét vesszük, akkor az eredeti adatok minimuma és maximuma közötti értéket kapunk, melyet ha normálunk nulla és egy közé, akkor százalékosan kapjuk meg az eltérést. Esetemben az adatok nulla és száz közöttiek, ezért normálás nélkül is jól mutatják a hibát.

3.1. Lineáris regresszió

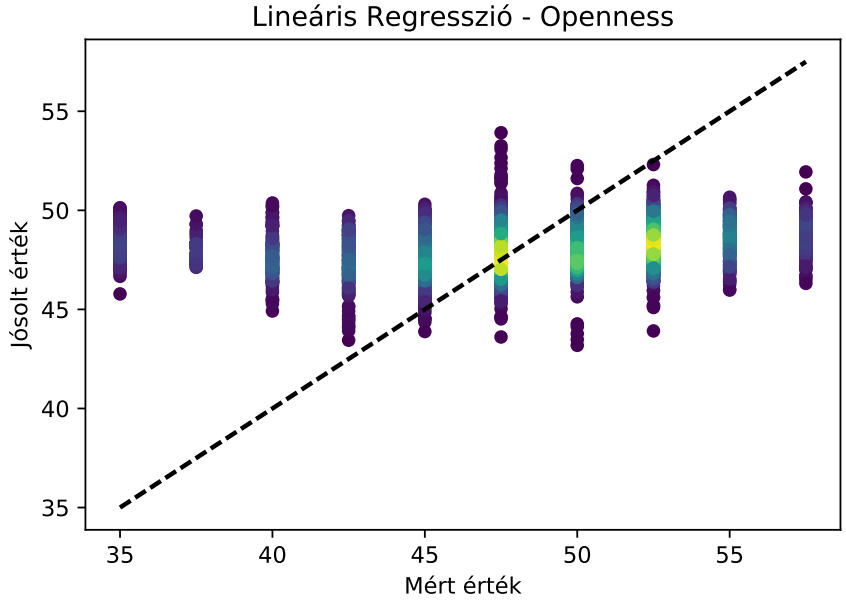


6. ábra. Lineáris regresszióval meghatározott célértékek ábrázolása hőtérképpel

A lineáris regresszió előtt megvizsgáltam, hogy milyen hibát kapok akkor, ha célváltozóknak konstans értéket adok. Ezt vettem alapul a későbbiek során az össze-

hasonlítások alkalmával, hogy ehhez képest mennyivel jobb eredményt tudok elérni bizonyos modellekkel.

Az egyszerű lineáris modellnek várhatóan nagy lesz a hibája, mivel itt nincs paraméter, melyet hangolhatnánk, csak egyszerűen együtthatókat illeszt az algoritmus a változókra.



7. ábra. Lineáris regresszióval meghatározott célértékek ábrázolása hőtérképpel

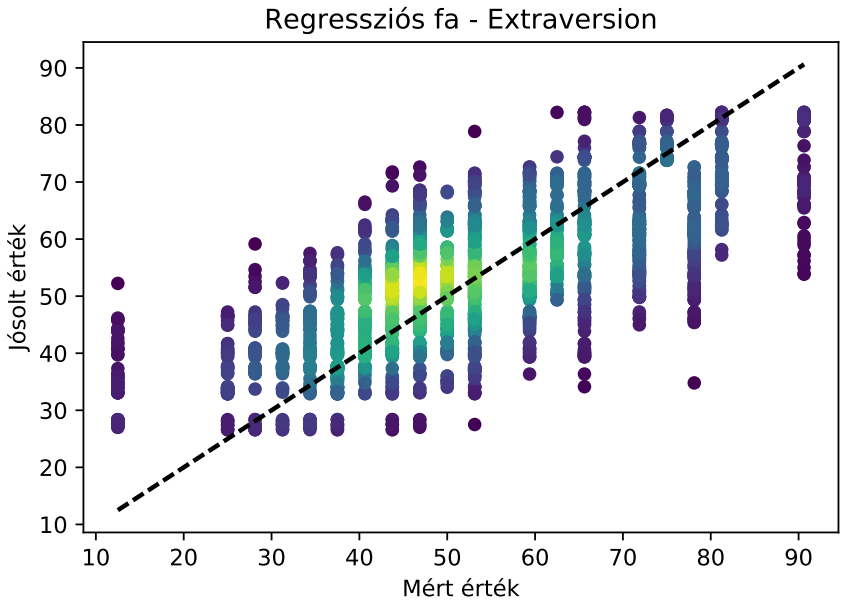
A mért és jósolt adatok szemléltetésére minden modellhez elkészítettem egy kétdimenziós ábrát, melynek egyik tengelyén a mért, másikon a jósolt értékek vannak, illetve a pontok sűrűségének érzékeltetéséhez hőtérképet használtam, mely a ritka területeken sötét lila, a sűrű területeken pedig sárga.

A lineáris regresszió, ahogy a képen is látszik, nem egy erős modell, az adott ponthalmazra egy egyenest tud csak illeszteni, melynek hibája nem lineáris kapcsolat esetén nagy. Mivel a korreláció nem mutatott jelentős összefüggést, így várható volt, hogy nem lesz elegendően pontos.

3.2. Regressziós fa

A regressziós fáknál [14] egyetlen fa megalkotásával döntjük el, hogy az adatok milyen vágásokkal, milyen döntésekkel legyenek besorolva adott értékekhez, ehhez

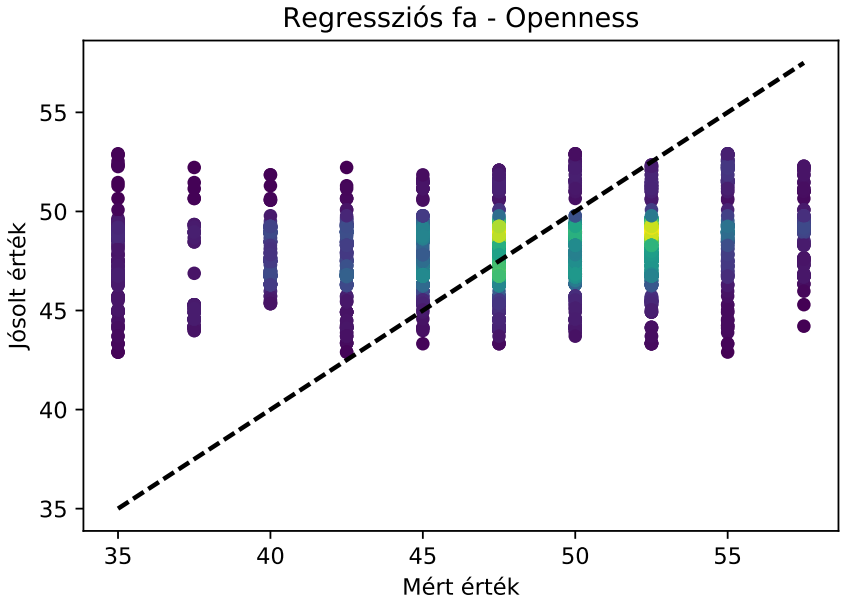
egyszerű döntések sorozatával jutunk el. A gyökérből indulva, minden szinten szétválasztja az adatokat az algoritmus egy feltétel alapján, melyet a tanuló adatok vizsgálatával határoz meg.



8. ábra. Regressziós fával meghatározott célértékek ábrázolása hőtérképpel

A fákat számos paraméterrel tudjuk finomhangolni. Ezek nem mindig hozzák meg a várt javulást, de némelyik esetben jelentős eredményt lehet velük elérni. Most is így történt, a Line Search módszert alkalmazva először a fa maximális mélységére alkalmaztam, a többi paramétert pedig változatlanul, az alap beállításon hagytam. Ekkor azt kaptam eredményül, hogy a maximum 5 mélységig épített fák jobb eredményt produkálnak, mintha semmilyen megkötést nem teszünk a maximális mélységre. Ezt felhasználva, a következő keresést már ennek beállításával, a maradék változtatása nélkül futtattam a levelekben lévő minimális darabszámra. Korábbi tanulmányaim során már tapasztaltam, hogy ezt érdemes vizsgálni, mivel ha egy fa levelébe túl kevés elem kerül, akkor ez túltanuláshoz vezethet, és ronthatja a modell teljesítőképességét. A keresés után azt az eredményt kaptam, hogy az 50 körüli, pontosabban a jelenlegi adatokon az 55-56 darab esetén kaphatunk szignifikánsan jobb eredményt. Ezeken kívül még egy tulajdonságot vizsgáltam, hogy összesen a fának hány levele legyen, ezt hasonló megfontolásból, mint a minimum levél-egyedszámot, ez is túltanuláshoz vezethet, azonban, ha a maximum túl kicsi

szám, akkor pedig alultanuláshoz. Ez a keresés nem hozott megszorítást, ha nem maximalizáljuk a darabszámot, akkor a folyamatosan javuló értékek beállnak egy minimumra, és onnantól nem romlanak, tehát ezt a tulajdonságot nem érdemes korlátozni, maradhat az alapbeállítás.



9. ábra. Regressziós fával meghatározott célértékek ábrázolása hőtérképpel

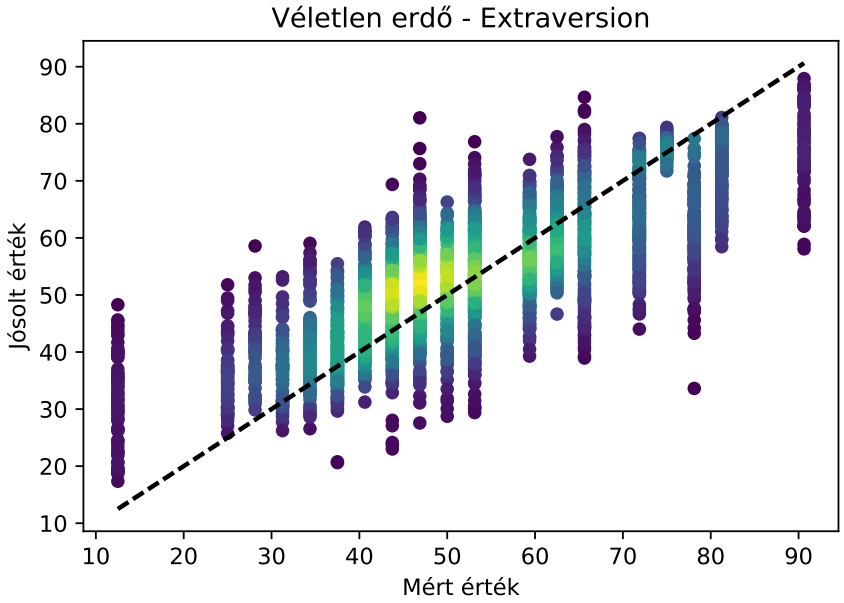
A hangolás után elkészült az első modell, mely már jobban teljesít, mint a skálázott érték, sőt jobban, mint a lineáris regresszió. Miután meghatároztuk a prediktált célváltozók értékeit, ábrázoltam őket egy-egy kétdimenziós ábrán, jelölve az $y = x$ tengelyt, amihez közelítenie kell a pontoknak.

Ahogy az 8. ábrán is jól látható, a zöldes színű pontok közel vannak a tengelyhez, ezeket a modell jól eltalálja, a sötétkéket viszont általában nem. A hőtérkép is a tengely mentén mutat nagyobb intenzitást. A széttagoltság a fában lévő vágásokkal magyarázható.

3.3. Véletlen erdő

A véletlen erdő [15] egy hibrid tanulási módszer, melyek több gyenge prediktorból állnak, és ezeket kombinálva egy erősebb, stabilabb, pontosabb modellt eredményeznek. Tanulási folyamatuk lassúbb, viszont jelentősen kisebb hibát produkálnak.

A véletlen erdőknél ezek a gyenge módszerek a fák, jelen esetben a regressziós vál-



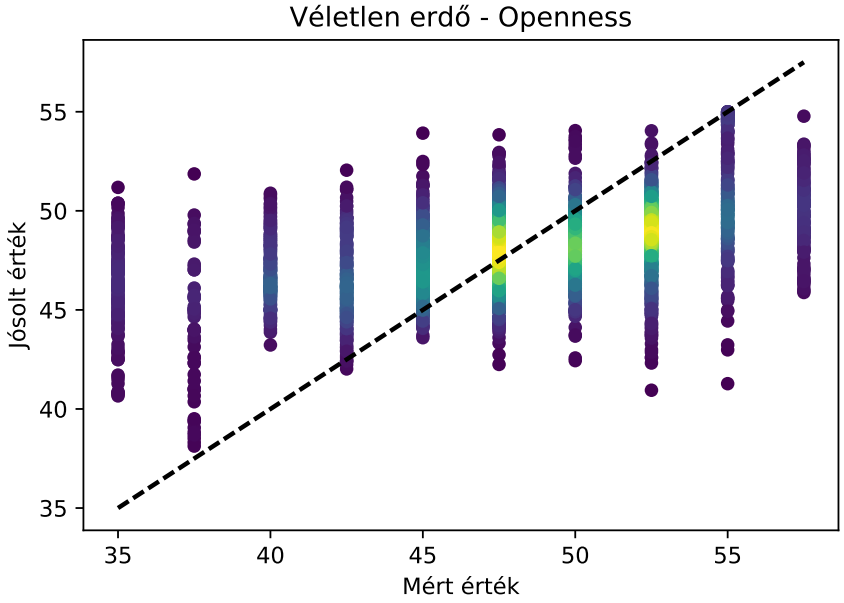
10. ábra. Véletlen erdővel meghatározott célértékek ábrázolása hőtérképpel

tozatuk. Fontos, hogy a különböző regressziós fák korrelációja kicsi maradjon, ha korrelálnak egymással, akkor nem kapunk új információt a különböző fákból, így nincs értelme többet felhasználni belőlük.

A véletlen erdők több általában sok fán alapulnak, mindig bizonyos számú fa kerül előállításra az adatok különböző részhalmazain, melyeknek aztán az átlaga határozza meg a véletlen erdő hibáját, így a túltanulást is viszonylag jól ki lehet küszöbölni. A pontosabb eredményhez minél több fát állítunk elő, annál jobb eredmény várható, azonban az időigény is gyorsan növekszik, mivel külön-külön, egyesével kell előállítani a fákat. A részhalmazok mérete mindig ugyanakkora, mint a tanítóhalmaz mérete, mert a mintavételezés visszatevéssel történik, tehát egy rekord többször is előfordulhat egy-egy fa építésénél.

A véletlen erdő hibájának csökkentését erősen befolyásolja az egyéni fák hibája, ha ezek nagyon gyenge eredményt produkálnak, akkor az erdőnek sem lesz nagyon jó az eredménye. Mivel az erdő fákra épül, ezért a fák jósága erősen befolyásolja a végeredményt, ezen kívül pedig a fák közti korreláció számít igazán, ha sikerül kis korrelációt elérni, akkor lehetséges a nullához közelítő hiba.

Ahogy a korábbi modellek esetében, itt is a Line Search technikát alkalmaztam a

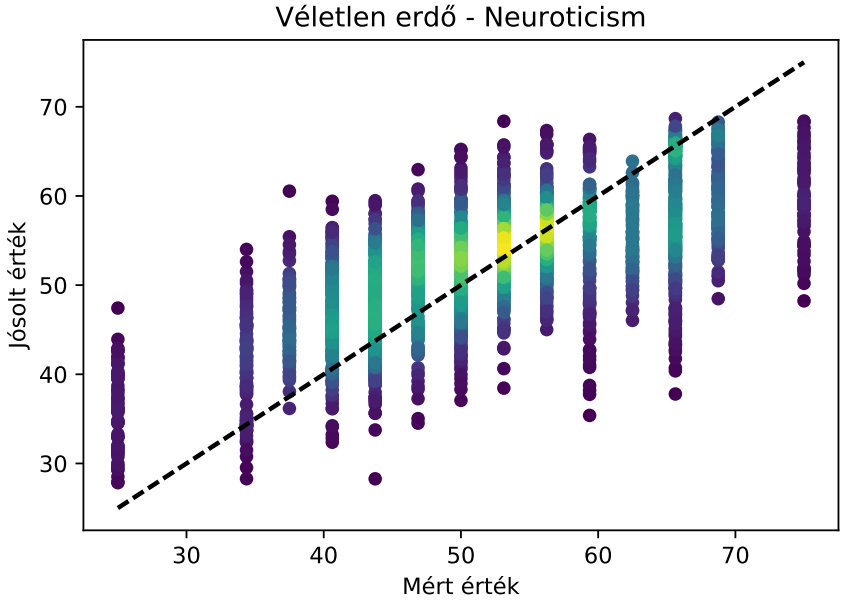


11. ábra. Véletlen erdővel meghatározott célértékek ábrázolása hőtérképpel

paraméterértékek beállításához. Mivel az erdők esetében a fák száma is választható, és ha ezt már az elején magas számnak választjuk, akkor időben nagyon elhúzódhat a program futása, így ezt hagytam a végére, kezdetnek beállítottam 20 fára, mivel 1 és 20 között itt kaptam a legkisebb hibát, majd a többi értéket hangoltam.

Elsőként a mélységre vonatkozó kritériumot vizsgáltam, amely nem hozott javulást semmilyen korlátra, ezért nem is állítottam be maximális értéket. Annál inkább javított a modellen a levelekben minimálisan besorolandó elemek száma, itt 20 körüli értékekre rendelkezett a legalacsonyabb hiba, ezt fixáltam. A levelek maximális számára sem érdemes korlátot beállítani, ezt kaptam a keresés során. Miután az eddigieket beírtam a paramétereknek, csak a fák darabszáma maradt hátra, mely - a fák egyenkénti előállítása miatt - hosszú folyamatnak bizonyult, 1 és 20 között már nem vizsgáltam, csak 20 és 100 között. A hosszas futás után (több, mint 12 ezer fa előállítása volt az algoritmus feladata az 5 célváltozóra összesen) körülbelül az 50 darab (célváltozónként eltérő, 49-52 közötti darab) fából álló erdőre minimális a négyzetes eltérés.

Ahogy azt a 10. ábra mutatja, nemcsak a zöldes színű pontok vannak közel a tengelyhez, hanem a szélső sávokban lévő pontok is egyre inkább elkezdtek konvergálni a középtengely felé. A széttagoltság már szinte egyáltalán nem jellemző,



12. ábra. Véletlen erdővel meghatározott célértékek ábrázolása hőtérképpel

legfeljebb a szélső esetekben fordul elő teljesen izolált pont.

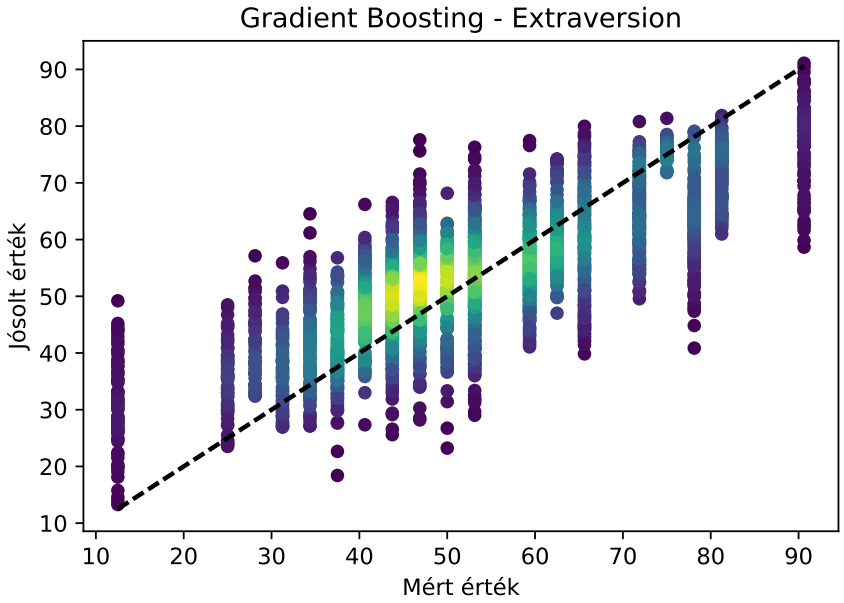
Elmondható, hogy jelentősen javultak az eredmények mind az öt változóra, a 5. táblázat mutatja, hogy hány százalékkal lettek jobbak (esetünkben csökkentek) a hibák.

A legrosszabb eredményt továbbra is az Openness (11. ábra) faktor produkálja, ennek több lehetséges oka is lehet, elsősorban az adatok kicsi változatossága állhat emögött. Míg a többi célváltozó esetében a skála szélesebb, és jobban szóródnak az adatok, itt ez nem figyelhető meg bizonyos értékekből sok van, és erre tud a módszer tanulni, a többit pedig csak találgatja. Azt gondolom, hogy ezt a tulajdonságot nehéz megjósolni, adataim alapján az emberek nem túl változatosak ilyen tekintetben, nagy a hasonlóság köztük.

3.4. Gradient Boosting

A Gradient Boosting [16] szintén egy hibrid módszer, több, gyenge regresszor felhasználásával kapjuk meg az összetett modellt. A regressziós döntési fák alkotják az alapot, először a teljes adathalmazon végezzük a vizsgálatot, majd az eredmény alapján több részre osztja az adatot az algoritmus, annak érdekében, hogy a célvál-

tozónknak kisebb legyen a szórása. A részeken külön elvégezzük újra a partícionálást, és ezt az iteratív folyamatot addig folytatjuk, amíg el nem érünk valamelyik megállító kritériumig. Ez a leállás lehet azért, mert minden elem egy partícióba tartozik, és nem tudjuk tovább bontani az adatot, vagy a paraméterként megadott elemszámot vagy mélységet elértük. A költségfüggvényünk itt is az átlagos négyzetes eltérés, és ennél a módszernél ennek segítéségvel „bünteti" a rosszul prediktált rekordokat.



13. ábra. Gradient Boosting technikával meghatározott célértékek ábrázolása hőtérképpel

Annak érdekében, hogy most se tanuljunk túl a tanító adathalmazon, a fa maximális mélységét beállítottam, a skálázás után hét mélységre kaptam a legkedvezőbb eredményt, így remélhetőleg az általánosító képessége javul a modellnek és ismeretlen adatokra is jól prediktál.

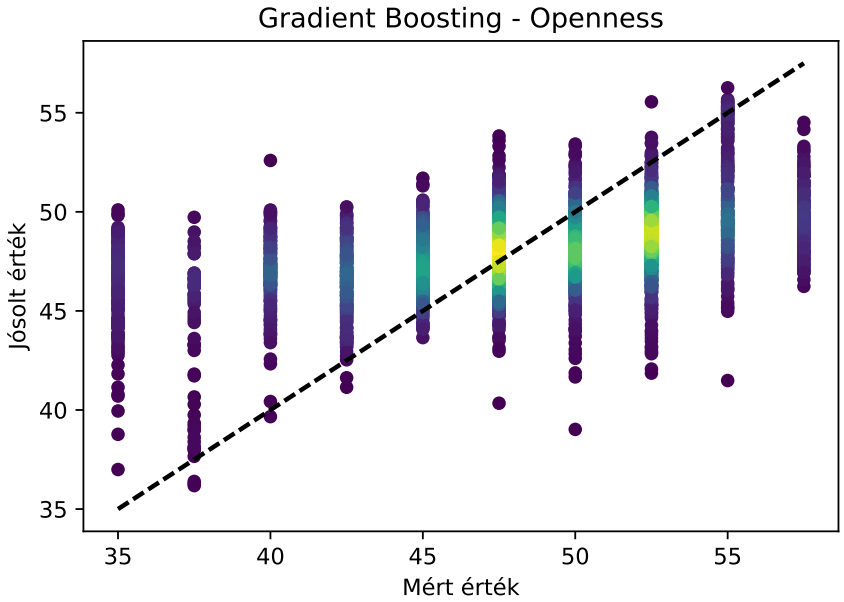
A módosítható paraméterek közül ezen kívül még a gyenge regresszorok arányát módosítottam, illetve maximalizáltam 90%-ban, továbbá a boostingok számát - hogy mennyi fát építsen - kerestem Line Search módszerrel, a legjobb teljesítményt 59-60-ra kaptam.

Beállítható még, hogy melyik hibára minimalizáljon a tanítás során az algoritmus, mivel a kutatásom során végig a négyzetes eltérés átlagát szeretném csökken-

teni, most is ennek megfelelően a legkisebb négyzetekre állítottam be.

Utolsóként a levelek maximális darabszámát és az ezekben lévő elemek számát vizsgáltam, hasonló eredményeket kaptam, mint a véletlen erdőknél, a levelek száma legfeljebb 19 lehet, ha jó eredményt szeretnénk, illetve egy levélben legalább 12 elemnek kell lennie.

A eddigi legjobb modellben a jósolt értékek - az 13. ábra alapján látszik, hogy - a középegyenes mentén helyezkednek el. A hiba értéke 10,2, ami a teljes skála 13%-a, 0,5%-al jobb, mint a véletlen erdő esetében.



14. ábra. Gradient Boosting technikával meghatározott célértékek ábrázolása hőtérképpel

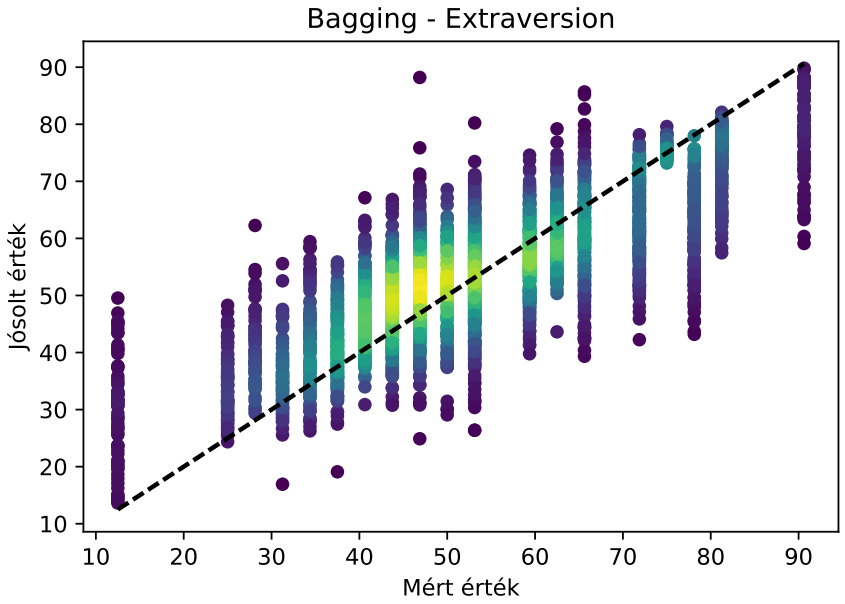
3.5. Bagging

Ahogy a Gradient Boostingnál, itt is egy erős hibrid regresszort épít az algoritmus több, gyenge modellből, a teljes halmazból mintavételezéssel veszi az adatot, visszatevés nélküli részhalmazokra építi a gyenge módszereket.

Elméletben a Bagging [17] jól kezeli, csökkenti a varianciát, míg a Boosting algoritmusok nemcsak a varianciát, hanem a torzítást is csökkentik. A gyakorlatban ez nem mindig teljesül, például az AdaBoost hajlamos túltanulni a tanító adathal-

mazon.

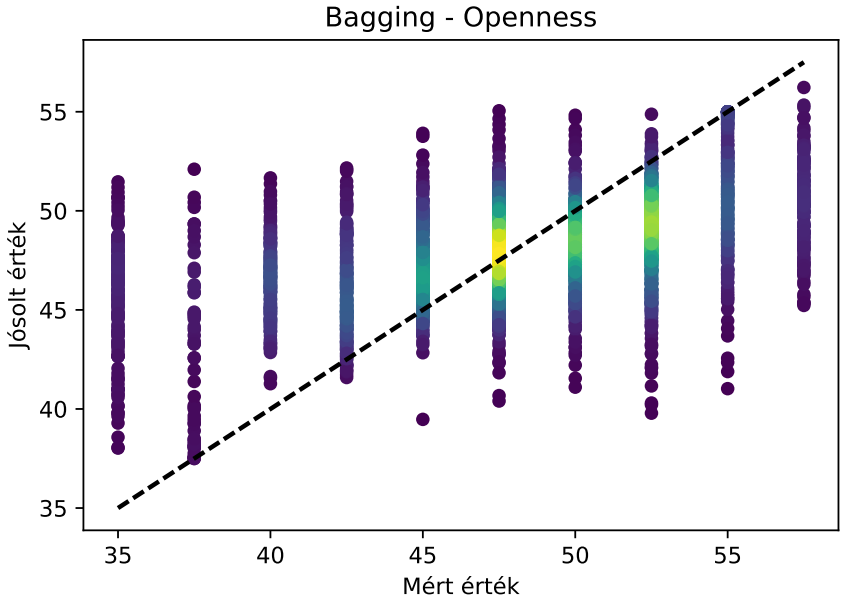
A paraméterek hangolása nem vezetett eredményre, több, különböző variációra sem kaptam szignifikánsan jobb eredményt, így az alapbeállításokat hagytam meg. Szükség volt egy gyenge prediktorra itt is, mely az általam már megépített regressziós fa lett. Ennek a regressziós fának az egyetlen hangolt paramétere a korábban meghatározott levél egyedszám, mely 55 esetén a legjobb.



15. ábra. Bagging technikával meghatározott célértékek ábrázolása hőtérképpel

A legfontosabb tulajdonsága, amivel javítani lehet a pontosságon, hogy hány fát tartalmaz az összetett modell. Ezt addig érdemes emelni, amíg már nem csökken tovább a hiba, beáll egy szintre, amit csak jelentős energiabefektetéssel lehetne tovább javítani. A próbák során 50, 100, 200, 500, 1000 és 2000 fára próbáltam ki a módszert, ez az öt célváltozóra körülbelül egy óra alatt futott le, viszont sajnos már 500 fánál sem kaptam lényegesen jobb eredményeket, ezért a továbbiakban 200 fával dolgoztam.

A Gradient Boosting és a Bagging közötti eltérés kevésbé látható, mint korábban, átlagosan mindössze 0.1%-al teljesített jobban az utóbbi. Habár ez önmagában nem sokkal jobb eredmény, de a futási idő itt jelentősen javult. Míg a Gradient Boosting esetében volt, hogy 10 percig futott a program, addig a Bagging a futások során mindig 6-7 perc alatt végzett.



16. ábra. Bagging technikával meghatározott célértékek ábrázolása hőtérképpel

3.6. AdaBoost

Az AdaBoost [18] az előzőektől abban tér el, hogy a tanítás során változnak a tanulóhalmaz elemeinek súlyai. Első körben minden egyes adat azonos súllyal szerepel, majd attól függően, hogy adott rekordhoz tartozó célváltozót pontosan találta e el, csökken a súly, ha pontos, növekszik, ha pontatlan. A következő körben már az új súlyozással építjük fel a prediktort, és ezt ismételjük, amíg el nem érjük az előre megadott iterációk számát, vagy minden adatra teljesen pontos nem lesz a modell. Ezzel azt szeretnénk elérni, hogy a pontatlan adatokra jobban koncentráljon a módszer. A kiugró értékekre és a zajos adatra érzékenyen reagál a módszer, így ennél az adatnál ezért fordulhatott elő, hogy nem teljesített olyan jól.

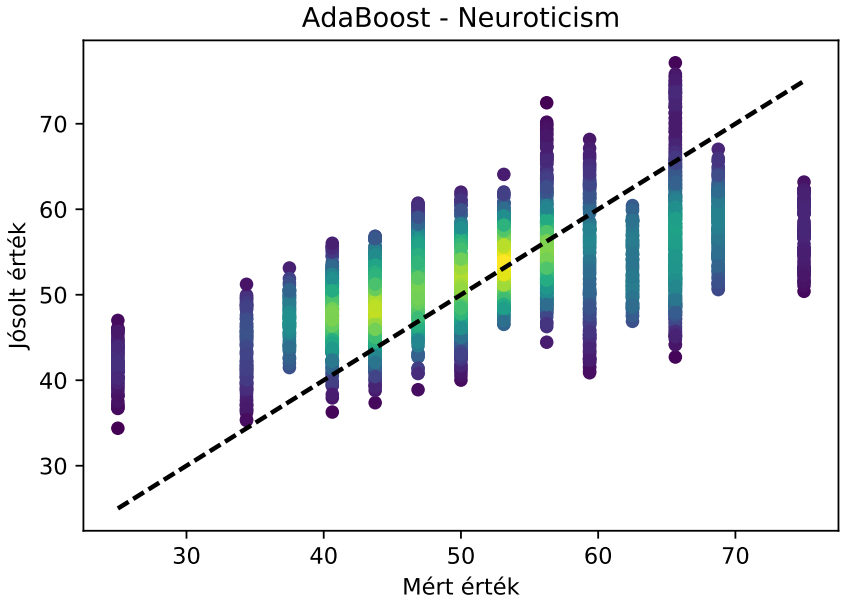
Az AdaBoost esetében szintén kevés tulajdonságot lehet hangolni, csak a gyenge prediktort és az építendő darabszámot tudjuk beállítani. Ahogy korábban is, a gyenge módszernek a regressziós döntési fát választottam, és ebből építettem 50, 100, 200, 500 darabot, és most sem tudott az algoritmus 500 fára annyival jobbat jósolni, hogy megérje a plusz időt.

A 6. táblázatból látható, hogy átlagos 47%-al jobb eredményt tudunk elérni az összetett módszerekkel, mintha konstans számmal próbálkoznánk. A regressziós fa

Modell	Átlagos négyzetes eltérés				
	Extrav.	Agree.	Conscient.	Neurot.	Openn.
Azonos érték	274.18	136.40	60.94	116.23	28.74
Lineáris regresszió	139.22	87.25	50.84	71.86	27.49
Döntési Fa	129.56	81.51	49.18	68.42	27.22
Véletlen Erdő	110.91	66.38	40.42	57.08	22.23
AdaBoost	135.65	78.76	40.35	57.24	21.94
Gradient Boosting	104.42	65.40	38.98	53.50	21.84
Bagging	101.20	62.39	37.38	52.28	21.38

4. táblázat. Az átlagos négyzetes eltérések összehasonlítása

és a véletlen erdő esetében még nem ilyen eredményes a modell, ott az átlag 29%, viszont a Gradient Boosting és a Bagging nagyon jól teljesít, az extraverziót több,



17. ábra. AdaBoost technikával meghatározott célértékek ábrázolása hőtérképpel

mint 63%-al jobban jósolja, mintha egy konstans értéket választanánk, a legnehezebben prediktálható változót, a kultúra, intellektust pedig több, mint 25%-al. Ez rendkívül bíztató eredmény, a mérések alapján a Bagging algoritmussal kapott modellt használnám új adatok esetén az öt faktor meghatározására, mivel nem csak az igaz rá, hogy ennek a legkisebb a hibája, hanem időben is ez a leggyorsabb a

Modell	Teljesítmény				
	Extrav.	Agree.	Conscient.	Neurot.	Openn.
Lineáris regresszió	49,2%	36,0%	16,6%	38,2%	4,3%
Döntési Fa	52,7%	40,2%	19,3%	41,1%	5,3%
Véletlen Erdő	59,5%	51,3%	33,7%	50,9%	22,7%
AdaBoost	50,5%	42,3%	33,8%	50,8%	23,7%
Gradient Boosting	61,9%	52,1%	36,0%	54,0%	24,0%
Bagging	63,1%	54,3%	38,7%	55,0%	25,6%

5. táblázat. A teljesítmény javulása a konstans értékhez képest

hibrid módszerek között.

4. Következtetések

Összességében azt gondolom, hogy célomat sikerült elérni, egy bizonyos pontosságig sikerüt az eszközhasználat alapján megjósolnunk az egyének személyiségjegyeit. További mérésekkel, és a modellek továbbfejlesztésével valószínűleg a pontosságuk növelhető, azonban ezek az eredmények is jelentősnek tekinthetők.

A Big Five modell felállításakor gyakran nincs szükség a százalékos pontosságra, amit én vizsgáltam a dolgozatom során, elegendő lehet egy ötfokú skála használata. Amennyiben áttérnénk erre, az adatainkat már más eszközzel, nevezetesen klasszifikációval vizsgálhatnánk. A jósolt adatokat is átskálázhatjuk az ötfokú skálára, ekkor a 7. táblázatban látható eredményeket kapjuk. A kultúra, intellektus látható, hogy kis sávban mozog, ezért nem kerül több skatulyába, így pedig nagy arányban eltalálja a módszer. Ezzel ellentétben, az extraverziót már nehezebben tudja prediktálni, erre igaz, hogy széles sávban vesz fel értékeket, emiatt pedig nagyobb a hiba is. A dolgozat során használt adatok pontos értékeket tartalmaztak, így ki tudtam használni az adatok folytonosságát, melyet a regresszió figyelembe vesz, azonban, ha címkézési feladatként foglalkoztam volna ezzel, akkor a folytonosság elveszik, és klasszifikációs feladat lett volna. Ez esetben lehet, hogy jobb eredményeket kapok az ötfokú skálán, viszont kevésbé lett volna pontos, mint regresszióval.

Érdekesnek tartanám további kísérletekkel annak vizsgálatát, hogy az életritmus felállítható e a személyiség faktorizálásval, és meghatározható e a mobiltelefon használatából. Ha sikerülne meghatározni valamilyen szabályosságot ezek között, akkor a mobiltelefonok sokkal személyre szabhatóbbak lehetnének, bizonyos körülmények között, ténylegesen a személyi asszisztensünk is lehetne a telefonunk.

Más szempontból a telefonok és kiegészítők által gyűjtött adatok felhasználásával egy nagy halmazra tanítva egyes modelleket, meghatározható és megelőzhető lenne adott betegségek kialakulása. Az okosórák elterjedésével - melyek a pulzust,

Modell	Helyesen klasszifikált arány				
	Extrav.	Agree.	Conscient.	Neurot.	Openn.
Azonos érték	50.98%	54.93%	62.07%	66.26%	93.72%
Lineáris regresszió	58.78%	62.96%	65.75%	70.54%	93.72%
Döntési fa	60.51%	62.86%	69.14%	71.60%	93.72%
Véletlen erdő	63.46%	67.51%	72.99%	75.19%	94.08%
AdaBoost	58.62%	60.27%	72.79%	75.02%	94.12%
Gradient Boosting	64.99%	67.91%	72.86%	75.42%	94.25%
Bagging	65.99%	69.51%	75.29%	76.98%	94.45%

6. táblázat. A regresszióval prediktált adatok átskálázásával nyert skatulyák illeszkedése az eredeti adatokra, százalékos pontosságban

vérnyomást, lépésszámot és egyéb tulajdonságokat is képesek mérni - rengeteg lehetőség nyílik ilyen témájú kutatásokat végezni.

Irodalomjegyzék

[1] Deborah A Cobb-Clark. Stefanie Schurer. "The stability of big-five personality traits". In: *Economics Letters* 115.1 (2012), pp. 11–15.

[2] Kendra Cherry. *The Big Five Personality Traits.* May 08, 2017. url: https://www.verywell.com/the-big-five-personality-dimensions-2795422.

[3] Xu R. Frey R. M. Fleisch E. Ilic A. "Understanding the impact of personality traits on mobile app adoption-Insights from a large-scale field study". In: *Computers in Human Behavior* 62 (2016), pp. 244–256.

[4] Peter H. Langford Cameron B. Dougall Louise P. Parkes. "Measuring leader behaviour: evidence for a "big five" model of leadership". In: *Leadership & Organization Development Journal* 38.1 (2017), pp. 126–144. doi: 10.1108/LODJ-05-2015-0103. url: http://dx.doi.org/10.1108/LODJ-05-2015-0103.

[5] Fanglin Chen Zhenyu Chen Tianxing Li Gabriella Harari Stefanie Tignor Xia Zhou Dror Ben-Zeev Wang Rui and Andrew T. Campbell. *StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones.* 2014. url: http://studentlife.cs.dartmouth.edu.

[6] Python Software Foundation. *Python Language Reference, version 3.6.1.* url: http://www.python.org.

[7] Computer software. Vers. 2-2.4.0. Continuum Analytics. *Anaconda Software Distribution.* Nov. 2016. url: https://continuum.io.

[8] Fernando Pérez and Brian E. "IPython: a System for Interactive Scientific Computing" Granger. In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 21–29. issn: 1521-9615. doi: 10.1109/MCSE.2007.53. url: http://ipython.org.

[9] S. Chris Colbert Stéfan van der Walt and Gaël Varoquaux. *The NumPy Array: A Structure for E cient Numerical Computation, Computing in Science and Engineering.* 13, 22-30 (2011). url: http://www.numpy.org/.

[10] Wes McKinney. *Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference.* 51-56 (2010). url: http://pandas.pydata.org/.

[11] John D. Hunter. *Matplotlib: A 2D Graphics Environment, Computing in Science and Engineering.* 9, 90-95 (2007). url: https://matplotlib.org/.

[12] Fabian Pedregosa et al. *Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research*. 12, 2825-2830 (2011). url: http://scikit-learn.org/.

[13] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. url: http://www.scipy.org/.

[14] Leo Breiman. Nong Shang. *Distribution based trees are more accurate*. url: https://www.stat.berkeley.edu/~breiman/DB-CART.pdf.

[15] Leo Breiman. *Random Forests*. January 2001. url: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf.

[16] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine". In: *The Annals of Statistics* Vol. 29.No. 5 (2001), 1189–1232. url: http://projecteuclid.org/euclid.aos/1013203451.

[17] Leo Breiman. "Bagging Predictors". In: *Technical Report* No. 421 (September 1994). url: https://www.stat.berkeley.edu/~breiman/bagging.pdf.

[18] Robert E Schapire. "Explaining AdaBoost". In: *Empirical inference*. Springer, 2013, pp. 37–52.