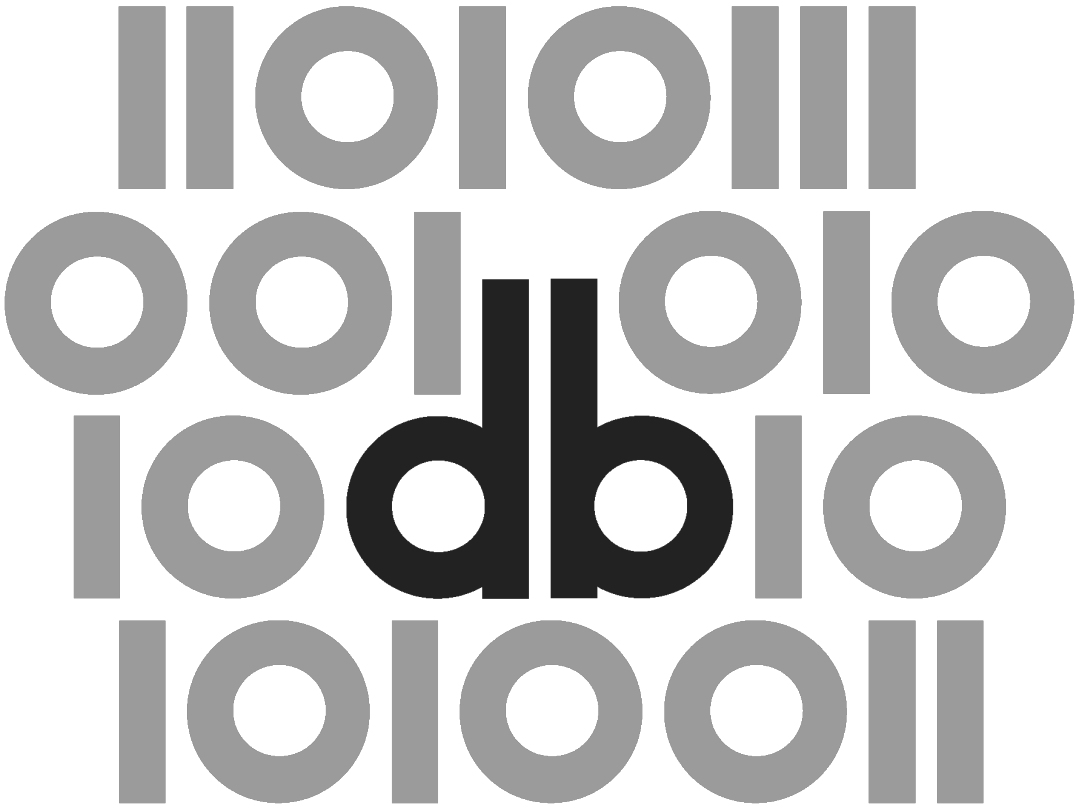
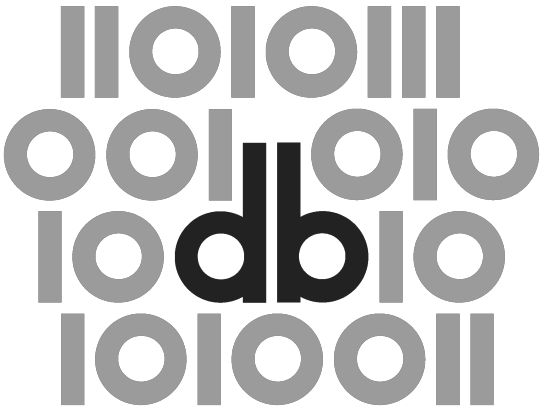


Digitális Bölcsészet
2022., hatodik szám

<DIGITÁLIS BÖLCSÉSZET>



6 (2022)

ISSN 2630-9696
DOI 10.31400/dh-hun.2022.6

Kiadja a Bakonyi Géza Alapítvány és az ELTE BTK Régi Magyar Irodalom Tanszéke (1088 Budapest, Múzeum krt. 4/A).
Felelős kiadó az ELTE BTK Régi Magyar Irodalom Tanszék vezetője.

Megjelenik az Open Journal Systems (OJS) v. 3. platformon, melynek működtetését az ELTE Egyetemi Könyvtár- és Levéltár biztosítja.

Honlap: `http://ojs.elte.hu/digitalisbolcseszet`
Email cím: dbfolyoirat@gmail.com

Olvasószerkesztő: Bucsics Katalin
Tördelés: Hegedüs Béla
Grafika: Hegyi Gábor

<KRITIKA>

Király Péter 0000–0002–8749–459

Göttingen eResearch Alliance Gesellschaft für wissenschaftliche Datenverarbeitung mbH

peter.kiraly@gwdg.de

Folgert Karsdorp, Mike Kestemont, and Allen Riddel. *Humanities Data Analysis: Case studies with Python*. Princeton and Oxford: Princeton University Press, 2021. 360 oldal. ISBN 9780691172361

A Folgert Karsdorp, Mike Kestemont és Allen Riddel alkotta szerzőhármas a szó szoros értelmében nehéz könyvet tett le az asztalra. A keménytáblás borító és a magasfényű papír miatt a kötet súlya 1,2 kg. Az oldalak fényvisszaverődése miatt napsütésben, tükröződő lámpafényben nehéz olvasni. Aki azonban eme nehézségeken átküzdi magát, igen színvonalas tartalommal töltekezhet. Mindemellett immár nyílt eléréssel, online könyvként is olvasható a `https://www.humanitiesdataanalysis.org/`-on, a felhasznált adatok pedig letölthetőek a *Zenodo* adatrepozitóriumból.[1]

A könyv célja, hogy tipikus esettanulmányokon keresztül bevezetést nyújtson a Python-alapú digitális bölcsészeti kutatás módszertanába, ehhez a Python programozási nyelvet választotta útitársként. A Python választása logikus, hiszen talán a legnépszerűbb nyelv ezen a területen, és viszonylag könnyű eljutni odáig, hogy a tanuló megírja és lefuttassa az első saját kódsorait. Bár a könyv a szerzők szerint igényel valamennyi Python-ismeretet, az olvasó nemcsak a digitális bölcsészetbe, de a nyelvbe is bevezetést nyer az alapoktól kezdve, még ha csak vázlatosan is (a kezdő olvasó számára megfelelő bevezető könyvet ajánlva). Az esettanulmányok kiválasztása is szilárd talajon áll, többnyire a digitális bölcsészeti szakirodalom egy-egy tanulmányát választották kiindulási pontnak, így az olvasó könnyen utánanézhet a könyv által nem tárgyalt részleteknek.

A könyv két nagy részből áll: az első négy fejezet az adatelemzés alapjait (programozási alapok, a Python által használt főbb adatszerkezetek, a legfontosabb fájltípusok és kezelésük); a többi öt fejezet pedig a haladó adatelemzés egy-egy kiemelt tárgykörét ismerteti (a statisztika alapjai, valószínűségszámítás, térképek, témamodellek [Topic Model] és stilometria). A fejezeteknek kötött dramaturgiájuk van. Bevezetéssel kezdődnek, melyek ismertetik az adott helyen érvényes kutatási kérdéseket, a feldolgozandó forrásokat, sőt néhány esetben a kérdés kutatástörténetére is kitérnek. A fejezetek végén további olvasmányokra vagy tudnivalókra hívják fel a figyelmet, illetve kezdő, mérsékelten nehéz és kihívásszámba menő gyakorlópéldák találhatók. Bizonyos, a témához tartozó, de a bevezetési szinten túli, haladóknak szánt fogalmat vagy módszert csak a példák tartalmaznak. Aki a könyvből igazán profitálni szeretne, annak azt tanácsolom, hogy oldja meg a példákat. A tanároknak pedig, akik a könyvet választják egy digitális bölcsészeti kurzus tananyagául (hiszen ez erre a legteljesebb

[1] `https://doi.org/10.5281/zenodo.891264`.

mértékben alkalmas), a feladatok lehetőséget adnak arra, hogy rávilágítsanak a Pythonban megvalósítható különböző módszerek előnyeire és hátrányaira.

Az első fejezet *(Bevezetés)* nagyon röviden áttekinti a bölcsészetben alkalmazott kvantitatív adatelemzés történetét, majd átveszi a Python legfontosabb eszközeit (változók, sorozatok indexelése, iteráció, listák, halmazok és szótárak, feltételes utasítások, külső modulok importálása, függvények definiálása, fájlműveletek). A könyv tucatnyi kódkönyvtárra támaszkodik, és sajnos elkerülhetetlen, hogy némely esetben ezek újabb változatainak funkciói nem kompatibilisek már a könyv írása idején aktuálissal, ilyen esetekben szükséges a könyvtár eredeti dokumentációját tanulmányozni, hogy a példaprogramot működésre bírjuk (pl. a PyPDF esetében). A rövid Python-bevezető után rögtön egy esettanulmányon lehet tanulmányozni a nyelv alkalmazását: Mit evett az Egyesült Államok lakossága? A cél a „feltáró adatelemzés" *(Exploratory Data Analysis)* módszerének ismertetése, amellyel behatóbban ismerhetjük meg adatainkat, jelen esetben a 18. század végétől a 20. század elejéig tartó időszakban kiadott szakácskönyveket tartalmazó gyűjteményből vett mintákat. Az adatokat egy CSV-fájlból a Pandas nevű kódkönyvtár által definiált speciális adatszerkezetbe, úgynevezett „adatkeretbe" *(Data Frame)*[2] másoljuk, ami az adatok tárolásán túl számos metódust is biztosít a bennfoglalt adatok elemzésére, például megjeleníthetjük az adatsor elejét vagy végét, egy adott oszlopban szereplő egyedi értékek listáját akár előfordulásuk mennyiségével együtt, de akár grafikonokat is létrehozhatunk. A feltáró adatelemzés során ezen módszerek segítségével meg tudjuk becsülni, hogy két változó között van-e összefüggés, és – ha van – milyen. Például hogyan változott a paradicsom konyhai szerepe 1810 és 1930 között, vagy melyek azok az alapanyagok, amelyek kezdetben főként egy adott népcsoport receptjeiben fordultak elő, majd később általános népszerűségre tettek szert.

A második fejezet címe *Strukturált adatok feldolgozása és kezelése.* Az adatok forrása valamilyen fájl – a könyv csupán említés szintjén foglalkozik más adatforrásokkal, például adatbázisokkal, példákat ezekre nem ad. Sorra veszik a bölcsészet által leginkább használt fájltípusokat: az egyszerű szövegállományt, a CSV-t (ebben az oszlopokban elrendezett adatsor mezőértékeit valamilyen határolójel, legtöbbször vessző választja el), a PDF-et,[3] a JSON-t (JavaScript-objektumjelölés), az XML-t általában és külön a TEI-t és a HTML-t. A szigorúan vett fájlfeldolgozás mellett megtudhatjuk, hogyan szűrjük az információkat, hogyan csomagoljunk ki tömörített állományokat, illetve hogyan lehet fájlt az internetről letölteni Pythonban. A fejezet bölcsészeti kérdése a drámaszövegek szereplői között fennálló interakciós hálózat kinyerése és az ebből adódó tanulságok levonása. A szakirodalomban tájékozott magyar olvasóknak Barabási Albert-László munkásságának köszönhetően ismerősek lesznek az itt olvasható

[2] Az adatkeret neve és fogalma megtalálható más adatelemző eszközökben is, például az R nyelvben, vagy az Apache Sparkban. Hogy melyikben bukkant fel először arra nem találtam adatot. Az utóbbi években ezek nagyon sokban hatottak egymásra, s ennek az egyik előnye, hogy a Pandas ismeretében legalább olvasni lehet a másik két eszközön írt adatelemző programokat.

[3] Érdemes itt is megjegyezni, hogy a PyPDF kurrens változatának (2.x) metódusnevei megváltoztak a könyvben található 1.x változathoz képes, így PDF.PdfFileReader() helyett PDF.PdfReader()-t, pdf.getPage(1) helyett pdf.pages[1]-t stb. kell alkalmazni. A kódkönyvtár alkotói szerencsére készítettek egy részletes migrációs útmutatót: PyPDF2. Migration Guide: 1.x to 2.x., hozzáférés: 2023.02.19., `https://pypdf2.readthedocs.io/en/latest/user/migration-1-to-2.html`

 DOI 10.31400/dh-hun.2022.6.5580

hálózatelméleti metrikák, és világos lesz, hogy a fejezet (miként a többi is) épphogy csak elindítja az olvasót a téma felé.

A harmadik fejezet témája a szövegjellemzők megismerése a vektortérmodell használatával. A vektortérmodell a keresőgépek révén vált széles körűen ismertté, de teret nyert más szövegfeldolgozási feladatokban is.[4] A lényege, hogy egy szövegkorpuszt egy nagy táblázatként tárolunk. Ennek a sorai az egyes dokumentumokat reprezentálják, oszlopai a korpuszban található szavakat, az egyes mezőértékek pedig adott kifejezés adott dokumentumban található előfordulási gyakoriságát tartalmazzák. A szótár előállítása általában többlépcsős normalizálási folyamaton keresztül történik: a szöveget – miután kikerültek belőle az írásjelek – szavakra bontják (tokenizálás), a szavakat kisbetűsítik, de akár szótövesíthetik *(stemming),* vagy lemmatizálhatják *(lemmatization)* is, ekkor a ragozott alak helyett valamilyen egységes szótővel vagy a szótári alakkal számolnak. Fontos tudni, hogy a modellben a szavak sorrendjét nem reprezentálja semmi, ez az információ eltűnik. Mivel a modell egy nagy, számokból álló táblázat, kiválóan lehet rajta mátrixműveleteket végezni. Ennek eredményeként ki tudjuk számolni a két vagy több dokumentum közötti távolságot, vagyis azt, hogy ezek, szókészletüket tekintve mennyire hasonlítanak egymásra. A könyv több lehetséges módszert is bemutat mind a szöveg feldolgozására, mind a távolságok kiszámítására. A módszert az automatikus műfajazonosítás feladatán keresztül ismertetik, ehhez a Paul Fièvre által gondozott, klasszikus 17. századi francia drámákat tartalmazó *Théâtre Classique* TEI-gyűjteményt használják fel.[5] A korpuszban található darabok háromféle korabeli műfajbesorolást tartalmaznak: tragédia, komédia és tragikomédia. A dokumentumtávolság elemzésével kiszűrhetjük a műfaj tipikus reprezentánsaitól nagyban eltérő darabokat. Kiviláglik továbbá, hogy összességében a tragikomédiák a két másik műfajhoz képest nem középen állnak, hanem sokkal inkább közelebb a tragédiákhoz, vagyis ezek lényegében olyan tragédiák, amelyekhez – a drámaiság élét tompítandó – némi humoros csavart adott a szerző. A fejezet továbbá tartalmaz egy kiegészítést a NumPy kódkönyvtár vektor- és mátrixműveleteiről.

Az első rész utolsó, negyedik fejezete a táblázatos adatok feldolgozásáról szól. Ennek a fő eszköze a Pandas kódkönyvtár, ezen belül is a már említett adatkeret nevű adatszerkezet. A Python kiváló lehetőségeket nyújt az adatok szűrésére, szelektálására, csoportosítására, módosítására. Az esettanulmány témája az Egyesült Államok-beli névadási szokások változásainak elemzése. Alapötlete, hogy évről évre vizsgáljuk meg, melyek voltak a legnépszerűbb keresztnevek, és vessük össze az egymásra következő időszakokat. Ennek alapján nemcsak azt állapíthatjuk meg, hogy melyek azok a nevek, amelyek újonnan lettek népszerűek, de azt is, hogy milyen gyakran váltak népszerűvé új nevek. A szerzők megvizsgálták azt is, vajon igaz-e az a feltételezés, hogy az utóbbi időben egyre többször találkozni *n*-re végződő nevekkel, illetve azt is, hogy

[4] Mártonfi Attila hívta fel a figyelmemet Jékel Pálnak és Papp Ferencnek a vektortérmodell felfutását több mint húsz évvel megelőző alkalmazását felvonultató kutatására, ami (eddig) meglehetősen visszhangtalan maradt a szakirodalomban (Jékel Pál és Papp Ferenc, *Ady Endre összes költői műveinek fonémastatisztikája* (Budapest: Akadémiai Kiadó, 1974). Kétségtelen – értékel Mártonfi –, itt a vektorteret nem a lexémák, hanem a fonémák feszítik ki (a korabeli gépi kapacitás nem is tett volna mást lehetővé); egyéb tekintetben azonban nagyon erős a hasonlóság.

[5] hozzáférés: 2023.03.17., `https://www.theatre-classique.fr/`

mi a helyzet a lányoknak és fiúknak egyaránt adott, uniszex nevekkel. A szerzők felhívják a figyelmet a forráskritika fontosságára: az alapforrás megbízhatóságát, reprezentativitását ugyanis számos ok befolyásolta az idők során (vagyis az adatok nem mindig tükrözik a teljes populáció névadási szokásait), ennek következtében bizonyos időszakokra a következtetések is szükségszerűen kevésbé szignifikánsak. Mindezeken felül e fejezetben ismerkedhetünk meg a vonaldiagram kiugró csúcsait lelapító, és így a trendeket tisztábban ábrázoló mozgóátlag-számítással, valamint néhány vizualizációs trükkel.

A könyv második része az elsőben megtanult fogalmakra épít, és míg az elsőben igyekeztek a kódokban előforduló újdonságokat a szövegben megmagyarázni, a második részben ez már ritkábban fordul elő, ott is inkább csak a főbb pontoknál (pl. leírva, hogy egy metódus mire jó, de nem kitérve egyes paramétereire). Ezért a második rész egyrészt több figyelmet igényel az olvasótól, másrészt – különösen az Allen Riddell által írt fejezetek – kicsit több odafigyelést kaphattak volna a szerzőktől is, lévén ezekben olyan kisebb-nagyobb gondolati ugrások, elmaradt magyarázatok akadnak, amelyek megnehezíthetik a tanultak alkalmazását másféle forrásokon és másféle kutatási kérdések esetében.

Az ötödik fejezet – címéhez *(A statisztika alapjai: ki olvas regényeket?)* híven – a statisztikai mérésekkel foglalkozik, s ehhez illusztrációképp az egyesült államokbeli általános társadalmi felmérést használja. A statisztikai mérőszám *(statistic)* megfigyelésekből álló adatgyűjtemény függvénye, ilyen például az összeg, az átlag, a minimum érték vagy a szórás. A fejezet sorra veszi a leíró vagy összegző statisztika fontosabb fogalmait és mérőszámait, valamint bemutatja a mennyiségi és a kategorikus változókra vonatkozó műveleteket. Itt találkozunk először matematikai modellel, amelynek a segítségével létre tudunk hozni a megfigyelt adatsorra hasonlító mesterségesen generált adatsort, a modell paramétereinek változtatásával pedig szimulálni tudunk eltérő kimeneteket is. A könyvben a háztartások jövedelemeloszlásának tanulmányozására alkalmazzák, a gammaeloszlás modelljével. Ezek a matematikai modellek jól ismert tulajdonságokkal bírnak, és a segítségükkel jobban le lehet írni az általános trendeket, vagy megkülönböztetni azokat az eseteket, amelyeknek az általános trendtől való eltérését magyarázhatja a véletlen, azoktól, amelyeknél az eltérés szignifikáns, vagyis a véletlennel nem magyarázható. A könyvben említett modellek általában valamilyen eloszlást is leírnak. A könyv kétféle eljárást ismertet annak megállapítására, hogy melyik modell illeszkedik az adatokra. Az elsőben az adatokhoz illeszkedő modellből különféle paraméterbeállításokkal ábrákat generálunk, és a szemre leginkább illeszkedőt választjuk, a másodikban pedig a Python (főként a scikit-learn kódkönyvtár által biztosított) gépi tanulási eszköztárát használjuk a modell paramétereinek kiválasztására. A fejezet utolsó része a mennyiségi és kategorikus változók közötti kapcsolatokkal foglalkozik, vagyis azzal, hogy két változó között van-e valamilyen korreláció. Végül megtudjuk, hogy a regényolvasási szokások és az USA régiói között mért 0,0069-es kapcsolat nem jelez olyan erős viszonyt, amelyet ne lehetne betudni a puszta véletlennek.

A hatodik fejezet a valószínűségszámításba, méghozzá annak is a Bayes-féle következtetésekről szóló ágába enged betekintést. A Bayes-szabály egy esemény bekövetkezésének valószínűségét számolja ki abban az esetben, ha egy ezzel összefüggő másik esemény már bekövetkezett. Például tudjuk, hogy ha véletlenszerűen kiválasztunk egy

 DOI 10.31400/dh-hun.2022.6.5580

1960 és 2010 között megjelent irodalmi művet, akkor 0,001% annak az esélye, hogy a mű szerzője Thomas Pynchon. Tegyük fel, hogy létezik olyan stilometriai alkalmazás, amely az általa írt műveket 90%-os valószínűséggel tulajdonítja helyesen Pynchonnek, és ez egy általunk vizsgált művet az írónak tulajdonít. Mennyi a valószínűsége, hogy a művet valóban ő írta? Itt a két esemény a következő: a művet Pynchon írta, és az alkalmazás neki tulajdonítja. A Bayes-szabály értelmében a válasz – elsőre talán meglepő módon – valamivel kisebb, mint 0,1%. A fejezet egy ehhez hasonló kérdést vizsgál: Ki írta *A föderalista (The Federalist Papers)*[6] vitatott cikkeit, Alexander Hamilton vagy James Madison? A szerzőség megállapításához a nem vitatott szerzőségű cikkekben szereplő kötőszavak (pl. *upon, by*) eloszlását vizsgálták (a stilometria egyik felismerése, hogy az ilyen, szinte öntudatlanul használt funkciószavak *[function words]* inkább jellemzők egy szerzőre, mint a főnevek, a melléknevek vagy az igék), majd a megfigyelt eloszláshoz egy arra illeszkedő matematikai modellt illesztettek (a negatív binomiális eloszlást). A modell előnye, hogy számszerűen össze lehet vetni az egyes szerzők szóeloszlási valószínűségeit, majd az így kapott értéket behelyettesíteni a Bayes-szabályba.

A hetedik fejezet címe: *Elbeszélés térképekkel.* Az amerikai harcmezővédelmi program nyilvántartást vezet az amerikai polgárháború jelentősebb csatáiról. Az egyes csatákról rögzítik, többek között, az időpontot, a helyszínt, a veszteségek számát, a győztes oldal nevét. A településnevek földrajzi koordinátáit geokódoló szolgáltatás segítségével nyerik ki.[7] A koordináták térképre vetítéséhez azonban kell még két összetevő: egy alaptérkép, amelynek a legelterjedtebb fájlformátuma az úgynevezett *shapefile,* valamint az adott terület sajátosságait leginkább tükröző vetület kiválasztása. A Python-térképek, amiként a Pandas grafikonjai is a Matplotlib kódkönyvtárral működnek együtt, ennek az előnye, hogy a grafikai ábrázolások kezelése itt is hasonló. A fejezet példája esetében ezt úgy aknázzák ki, hogy a polgárháború minden egyes hónapjáról készül egy – színkódokkal a csata eredményét, a pont nagyságával pedig a veszteséget jelölő – térkép, és ezek egyetlen nagyobb képbe vannak rendezve, összességében világosan megrajzolva a polgárháború hadi eseményeinek főbb tendenciáit.

A nyolcadik fejezet visszatér a stilometria és a szerzőazonosítás témájához, de egy másik módszert, a klaszterálást állítva fókuszba, és amerikaiak helyett középkori szerzőkkel: Bingeni Szent Hildegárddal, utolsó titkárával, Guibert de Gembloux-val és Clairvaux-i Szent Bernáttal. A fejezet ismerteti a „Burrows-féle Delta" nevű eljárást, amely egy gépi tanuló algoritmus, s az első, tanulási fázisban ismert szerzők műveit elemezve nyeri ki a szerzőre jellemző ismérveket, majd a második, előrejelző fázisban azt nézi meg, hogy a nem ismert szerzőjű művek ismérvei melyik szerző ismérveire hasonlítanak leginkább. Az ismérveket egy normalizált szám jelöli, a statisztikából vett *z* szám, amely azt mutatja meg, hogy az adott érték hány szórásnyira van az átlagértéktől. A szerzőazonosítás a tapasztalatok szerint akkor működik jól, ha a vizsgált szövegek hossza nagyobb egy bizonyos mennyiségnél (a könyvben idézett

[6] Magyarul: Alexander Hamilton, James Madison, és John Jay, *A föderalista: Értekezések az amerikai alkotmányról,* ford. Balabán Péter, jegyz. Magyarics Tamás (Budapest: Európa Könyvkiadó, 1998).

[7] A 232. oldal 6. lábjegyzete szerint az a Python-szkript, amely kiolvassa a forrásból a településnevet és lekérdezi a geokódoló szolgáltatást a könyv weboldalán elérhető, ez azonban sajnos nem így van. Sem ott, sem a könyv kódrepozitóriumaiban (GitLab és Zenodo) nem érhető el.

kutatás szerint az alsó határ 6500 szó), valamint a vizsgált szövegek nagyjából egyforma hosszúak. Jelen esetben a szövegeket 10000 lemmás darabokra bontották, részben azért is, hogy megnézzék, az egyes szerzők különböző szövegei valóban ugyanabba a csoportba kerülnek-e. A vizsgálandó szavak körének kijelölésére lehet gyakoriságon alapuló automatikus módszert választani a Python gépi tanulási módszereket tartalmazó scikit-learn kódkönyvtára eszköztárából, de akár saját szótárral is lehet dolgozni. A vektortérmodellt ezúttal nem a korábban ismertetett kézi módon, hanem a scikittel valósították meg.

A Burrows-féle Delta az úgynevezett felügyelt tanítás családjába tartozik, tipikusan címkét rendel a vizsgált egyedekhez attól függően, hogy a tanulás során az ismert egyedeknek milyen címkéjük volt (jelen esetben a címke a szerző neve). Létezik azonban egy másik algoritmuscsalád, a felügyelet nélküli tanulás, amely nem használ efféle címkéket. A könyv két ilyen eljárást ismertet, a hierarchikus egyesítő klaszterezést *(Hierarchical Agglomerative Clustering)* és a főkomponens-elemzést *(Principal Component Analysis, PCA)* – közös tulajdonságuk, hogy egyik sem kínál osztálynevet, és mindkettő alkalmas grafikai megjelenítésre. Az első eredménye bináris faszerkezet, ahol az ág egyszerre legfeljebb kétfelé ágazhat el, és ahol a kutató az ágrajz vizsgálata után döntheti el, hogy hány csoportot képez. Szemben a címkézéssel, amely pusztán szerzők szerint csoportosítja a szövegeket, itt az azonos szerzőhöz tartozó művek alcsoportjai is detektálhatók. A PCA úgynevezett dimenziócsökkentési eljárás. Jelen esetben a szövegek dimenziószáma a szótár nagyságával, 65-tel egyenlő (a szótár csak a funkciószavakat tartalmazza). Az egyes dimenziók azonban sokszor mutatnak valamilyen negatív vagy pozitív korrelációt egymással, vagyis az egyik ismeretében a másik értéke előrejelezhető. A főkomponensek alkotásához az algoritmus elemzi ezeket a korrelációkat, és sokváltozós egyenletet állít fel, amelyben az egyes dimenziók súlyként szerepelnek. Ezek a főkoponensek, számuk megegyezik a dimenziók számával, de magyarázó képességük erősen eltér. A gyakorlatban csak néhány főkomponens képes reprezentálni a sokaság jelentős csoportjait, a többi inkább csak finomítja a képet. A könyvben két ilyen főkomponenssel számolnak, mindkettőben külön-külön súllyal szerepelnek az egyes dimenziók; együttes magyarázó erejük közel 60%. A két dimenziót már meg lehet jeleníteni grafikonon, ezen kiválóan látszik, hogy a három szerző mennyire tér el egymástól.

Az utolsó, kilencedik fejezet az Egyesült Államok Legfelsőbb Bíróságán 100 éves időtávban született döntésekhez csatolt különvélemények témamodelljét vizsgálja. Az alkalmazott matematikai modell a Dirichlet-féle rejtett elhelyezkedés *(latent Dirichlet allocation)*. A modell alapja az a feltételezés, hogy a trendszerűen egymás szomszédságában vagy egyazon dokumentumban előforduló szavak csoportja egyúttal egy témát is kirajzol. A metódushoz, amely a scikit-learn kódkönyvtár része, meg kell adnunk, hány témával szeretnénk számolni (láttuk, hogy a hierarchikus klaszterezésnél erről utólag dönthettünk). A számítás eredményeképpen megkapjuk az egyes témákhoz tartozó szavak listáját a befoglaló dokumentumok számával, illetve az egyes dokumentumokhoz tartozó témákat és azok súlyát, vagyis hogy azok mennyire relevánsak az adott dokumentumra nézve. Az egyes témáknak nincs nevük, csak azonosítójuk, azokat a kutatónak kell elneveznie vagy legalább a jelentését értelmeznie, és ez nem minden esetben egyszerű. Fontos, hogy a szavak eloszlása a témák között

 DOI 10.31400/dh-hun.2022.6.5580

nem kizárólagos, egy-egy szó több különböző témának is része lehet, ugyanakkor érdekes módon a többjelentésű szavak különféle jelentései általában más-más témába kerülnek. Részletesen csupán a leginkább használatos témamodell leírását olvashatjuk, de a szerzők utalnak arra, hogy vannak egyéb közelítések, például olyan, amely a kronológiát is figyelembe veszi. A magyar és más agglutináló nyelvek szempontjából érdekes az a kutatás, amely a szótövezés és a stopszavak alkalmazásának hatását vizsgálja.

A könyv végén bibliográfia és jó tanácsok sorakoznak a tudományos célú informatika „elég jó gyakorlatainak" tárgyában (az adatkezelésről, a szoftverről, az együttműködésről, a munkaszervezésről, a változáskövetésről és a kéziratokról).

A könyvben a Python olyan segédeszköz, amelyet kutatási (adatelemzési) céllal használnak, ennek megfelelően a nyelvnek csak azokat a tulajdonságait érintik, amelyek ehhez a feladatkörhöz tartoznak, de még ezek közül is csak a legfontosabb összetevőket. Nincs szó például osztályokról, függvények és változók névadásáról, tesztelésről vagy akárcsak arról sem, hogy egy Python-szkriptnek mit kell tartalmaznia ahhoz, hogy le tudjuk futtatni. Ezekről a könyvben ajánlott forrásokat kell tanulmányozni. Hasznos lett volna a kutatásiszoftver-fejlesztés sajátosságainak tárgyalása is, ami a kutatásiadat-kezeléssel párhuzamoson fejlődő újabb szakterület, de erre csak az epilógusban van utalás.[8] A könyvre épülő kurzusban véleményem szerint ezeknek a témáknak a könyvnél kifejtettebb módon kellene szerepelnie, mivel e két további összetevő garantálja, hogy az általunk írott kód néhány év múlva is futtatható és érthető maradjon.

Nagy erénye a könyvnek, hogy nem akar mindentudó lenni, és hogy a műfaj legjobb hagyományai szerint a vizsgált témákról további irodalmat ajánl. Bár a bölcsészeti adatelemzés számos témáját átfogja, de – és ezt láthatjuk akár a szakterület érettségének a bizonyítékaként is – legalább ennyi minden nem fért bele e vállalkozásba. Hogy csak néhányat említsünk: adattisztítás és adatminőség; nevek, fogalmak kinyerése és a hozzájuk tartozó entitások azonosítása; nyelvészeti (pl. szófaj-, mondat-) elemzés; hipotézistesztek vagy a nem szöveges adatok (zene, kép, mozgókép, térbeli objektumok) elemzése. Nagyon remélem, hogy ezen és más hiányzó témák bemutatása céljából vagy a szerzők rugaszkodnak neki egy második kötetnek, vagy a könyvtől ihletett olvasók készítenek hasonlókat.

[8] Wilson et al., „Good Enough Practices in Scientific Computing," *Plos Computational Biology,* 2017. június 22., `https://doi.org/10.1371/journal.pcbi.1005510`. Emellett érdemes elolvasni az Európai Kutatási Infrastruktúra-szoftver Mérnökök Hálozata (EURISE) műszaki referenciadokumentumát, hozzáférés: 2023.03.17, `https://technical-reference.readthedocs.io/en/latest/`, illetve a brit Software Sustainability Institute útmutatóit, hozzáférés: 2023.03.17, `https://www.software.ac.uk/resources/get-speed`. Tananyagként pedig a következő mű használható: Damien Irving et al., *Research Software Engineering with Python: Building Software that Makes Research Possible,* hozzáférés: 2023.03.07, `https://merely-useful.tech/py-rse/index.html`.