UNIVERZA V LJUBLJANI

EKONOMSKA FAKULTETA

ZAKLJUČNA STROKOVNA NALOGA VISOKE POSLOVNE ŠOLE

PRIMERJAVA ORODIJ ZA PRIPRAVO PODATKOV V PROCESU PODATKOVNE ANALITIKE

Ljubljana, maj 2020 MARCEL LAH

IZJAVA O AVTORSTVU

Podpisani Marcel Lah, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Primerjava orodij za pripravo podatkov v procesu podatkovne analitike, pripravljenega v sodelovanju s svetovalcem dr. Jurijem Jakličem.

IZJAVLJAM

1. da sem predloženo delo pripravil samostojno;

2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;

3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;

4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;

5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;

6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;

7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;

8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;

9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;

10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne _____ Podpis študenta:_____

KAZALO

UVOD **1**
1 POSLOVNOINTELIGENČNI SISTEM IN PROCES PODATKOVNE ANALITIKE **2**
1.1 Struktura poslovnointeligenčnega sistema **2**
1.2 Tehnologije znotraj poslovnointeligenčnega sistema **3**
1.3 Proces podatkovno analitičnega projekta **3**
1.3.1 Razumevanje poslovnega problema 4
1.3.2 Poznavanje zbranih podatkov 4
1.3.3 Priprava podatkov 5
1.3.4 Analiza podatkov in modeliranje 5
1.3.5 Ovrednotenje uporabljenih modelov 5
1.3.6 Uvajanje modelov ali poročanje rezultatov 5
2 PRIPRAVA PODATKOV V PROCESU PODATKOVNE ANALITIKE **6**
2.1 Proces pridobivanja in raziskovanja podatkov **6**
2.2 Aktivnosti čiščenje podatkov **7**
2.1.2 Obravnavanje manjkajočih vrednosti 7
2.1.3 Obdelava in iskanje osamelcev 8
2.3 Aktivnosti transformacije podatkov **9**
2.3.1 Normalizacija 10
2.3.2 Sestavljanje in konstrukcija novih dimenzij 10
2.3.3 Agregacija in generalizacija 10
2.3.4 Metode glajenja 11
2.2 Integracija podatkov **12**
2.4 Aktivnost redukcije podatkov **12**
3 ORODJA ZA PRIPRAVO IN OBDELAVO PODATKOV **12**
3.1 Samopostrežna orodja in uporaba programskih jezikov **12**
3.2 Obravnavani kriteriji **13**
3.3 Izbrana orodja **14**
3.4 Predstavitev primera **15**
3.5 Programski jezik Python **16**
3.6 Knime analitična platforma **18**
3.7 Microsoft Power Query **19**
3.8 Opis izvedbe zgornjega primera s predstavljenimi orodji **21**

3.9 Diskusija **23**
SKLEP **25**
LITERATURA IN VIRI **26**

KAZALO SLIK

Slika 1: Struktura poslovno inteligenčnega sistema 2
Slika 2: CRISP-DM metodologija podatkovno analitičnega projekta 4
Slika 3: Grafični prikaz škatle z brki 8
Slika 4: Grafični prikaz delovanja algoritma DBSCAN 9
Slika 5: Primer agregacija podatkov po izbranem atributu 11
Slika 6: Uporabniški vmesnik razvojnega okolja Spyder 17
Slika 7: Uporabniški vmesnik Knime analitične platforme 19
Slika 8: Uporabniški vmesnik Microsoft Power Query 21

SEZNAM KRATIC

angl. - angleško
CRISP-DM – (ang. Cross industry standard process for data mining); Medpanožni standardni proces podatkovnega rudarjenja
DBSCAN – (ang. Density based spatial clustering of applications with noise); Algoritem prostorskega gručenja na podlagi gostote
ELT – (ang. Extract, load, transform); Pridobi, naloži, transformiraj
ETL – (ang. Extract, transform, load); Pridobi, transformiraj, naloži
IQR – (ang. Interquartile range); medkvartilni interval
OPSI – Odprti podatki Slovenije
PIS – Poslovnointeligenčni sistem

UVOD

Koristi podatkovno podprtih poslovnih odločitev so bile v preteklosti že večkrat dokazane. Profesorji z univerze Messachusetts Institute of Technology so izvedli raziskavo, kjer so hoteli ugotoviti, kako odločitve, ki so podprte z zbranimi podatki, vplivajo na uspešnost poslovnega subjekta. Rezultati so pokazali, da bolj kot se podjetje osredotoča na sprejemanje podatkovno podprtih odločitev, bolj produktivno je, beleži večje donosnosti sredstev in kapitala, bolje izkorišča svoja sredstva in povečuje lastno tržno vrednost (Provost & Fawcett, 2013).

Stodder (2016, str. 4) poudarja, da je ključ za uspešno analitiko kvaliteta in relevantnost zbranih podatkov. To dosežemo z ustrezno pripravo podatkov, ki mnogokrat velja za počasen in zahteven proces. Mnogo uporabnikov se tako naslanja na različna samopostrežna orodja, ki tovrstno delo poskušajo olajšati. Hkrati omogočajo dovoljšno mero fleksibilnosti, kar zagotavlja samostojnost pri izvajanju procesa priprave, brez potreb po posredovanju specializiranega kadra.

Orodij za pripravo podatkov je na trgu ogromno. Procesa se lahko lotimo z uporabo samopostrežnih orodij ali pisanjem lastnih programskih rešitev, z uporabo primernih programskih jezikov. Izbira orodja za dano situacijo je lahko problematična. Nekatera ne ponujajo uporabniškega vmesnika, zahtevajo visoko predhodno znanje ali pa niso dovolj fleksibilna. V ta namen bom v nalogi natančneje predstavil tri popolnoma različna orodja, s katerimi se lahko lotimo procesa priprave. Hkrati jim bom pripisal tudi ustrezno mesto uporabe. Namen je torej predstaviti glavne aktivnosti znotraj procesa priprave podatkov in izvesti primerjalno analizo orodij za opravljanje le-teh. Vedeti je potrebno, da na voljo ni orodja, ki bi zanesljivo zadovoljil vse potrebe, zato je potrebno smiselno opredeliti in izbrati najprimernejše v dani situaciji. Vsebina naloge je lahko torej v pomoč vsakomur, ki se srečuje s procesom priprave podatkov in želi izbrati primerno orodje in hkrati dobiti vpogled v način dela, ki ga posamezno orodje zahteva.

Pri pregledu posameznega orodja bom uporabil srednje velik podatkovni set, ki beleži vse podatke, ki so bili pridobljeni na tehničnih pregledih motornih vozil v Sloveniji v letu 2019, pridobljen s spletnega mesta OPSI (Odprti podatki Slovenije). Sama primerjava orodij bo temeljila na predhodno določenih kriterijih, ki jih bom upošteval pri pregledu.

Struktura naloge sestoji s teoretičnega in praktičnega dela. V prvem poglavju teoretičnega dela predstavim področje poslovne inteligence in tehnologije, ki se pri tem uporabljajo. V nadaljevanju okvirno opredelim splošen proces podatkovno analitičnega projekta in predstavim aktivnosti, ki sodijo zraven. Drugo poglavje teoretičnega pregleda pa je namenjen predvsem natančnejši opredelitvi procesa priprave podatkov za zahteve kvalitetnih podatkovnih analiz. Namen je, da se osredotočimo na teoretično podlago procesa, iz katere bom lahko kasneje črpal znanje za izvedbo kakovostnega testa primerjave treh orodij. Po celovitem teoretičnem pregledu sledi praktična izvedba testov primerjave na

izbranem primeru. V začetnem delu bom opredelil kriterije, ki jih bom upošteval pri izvajanju testov. Nato se bom lotil natančnejšega opisa orodij glede na posamezne kriterije. Predstavil bom vsa uporabljena orodja in pri vsakem od kriterijev izpostavil nekaj ključnih lastnosti. Zapisal bom tudi proces, kako sem se z vsakim od orodij lotil zastavljenega primera. Na koncu bom podal tudi končno mnenje in zapisal nekatere pozitivne oz. negativne opazke, ki sem jih zaznal tekom testiranja.

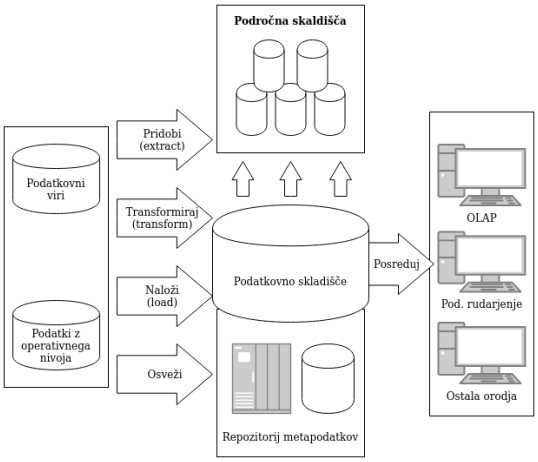
1 POSLOVNOINTELIGENČNI SISTEM IN PROCES PODATKOVNE ANALITIKE

Poslovnointeligenčni sistem (v nadaljevanju PIS) je namenjen poslovnemu odločanju, ki sloni na procesu zbiranja in shranjevanja podatkov, ter hkrati omogoča upravljanje znanja pridobljenega iz zbranih podatkov. To izvajamo s pomočjo uporabe metod podatkovne analitike, ki lahko zajemajo izdelavo enostavnih poizvedb, ad-hoc poročil, večdimenzionalne analitike, kompleksnejše metode napovedne analitike in metode podatkovnega rudarjenja (Negash & Gray, 2008, str. 175-176).

1.1 Struktura poslovnointeligenčnega sistema

Osrednji in ključni del PIS predstavlja podatkovno skladišče, ki opravlja nalogo posredovanja podatkov analitikom za namene obdelave in analize. Podatkovna skladišča hranijo združene podatke iz operativnega nivoja različnih poslovnih enot in morebitnih zunanjih virov. Negash in Gray (2008, str. 177-178) podatkovno skladišče opišeta kot vmesni člen med ti. okoljem podatkovnega skladiščenja in analitičnim okoljem. Okolje podatkovnega skladiščenja zajema proces pridobivanja surovih podatkov iz operativnega nivoja podjetja, kateri po zajemu potujejo skozi proces integracije, ki mu pravimo ETL (angl. extract, transform, load). ETL zajema pridobitev, transformacijo in vnos podatkov v podatkovno skladišče. Analitično okolje ima nato možnost črpanja teh podatkov za izvedbo podatkovnih analiz.

Slika 1: Struktura poslovno inteligenčnega sistema



Vir: Prirejeno po Chen (2001).

Slika 1 natančno prikaže tradicionalno strukturo PIS in proces podatkovnega skladiščenja, od pridobivanja podatkov do končnih analiz. Potrebno je omeniti tudi področna podatkovna skladišča (angl. data marts), ki hranijo podatke iz posameznih oddelkov oz. področij v podjetju in morebitne podatke iz zunanjih virov. Te služijo kot vir podatkov za izvedbo področnih analiz. Na sliki 1 opazimo tudi repozitorij metapodatkov. Metapodatki so v splošnem pomenu podatki o podatkih. V zgornjem prikazu služijo opisovanju vseh zbranih podatkov, ki so shranjeni v podatkovnem skladišču.

Marín-Ortega, Dmitriyev, Abilov in Gómez (2014, str. 669-670) poudarjajo, da se je v zadnjih letih poleg klasičnega ETL procesa začel uporabljati tudi proces ELT (angl. extract, load, transform). Podatke se v tem primeru iz najrazličnejših podatkovnih virov naloži neposredno v podatkovno skladišče oz. v tem primeru podatkovno jezero v svoji prvotni obliki in šele nato transformira, počisti in pripravi za potrebe analitike. Razlog za uporabo ELT procesa je predvsem v napredku pri hitrosti in zmogljivosti tehnologij podatkovnih baz, ki se uporabljajo za potrebe podatkovnega skladiščenja, nižjih stroškov shranjevanja podatkov ter raznolikosti in količine zbranih podatkov. Proces ELT hkrati zagotavlja večjo mero fleksibilnosti, kajti v podatkovnem jezeru hrani prvotne in neobdelane podatke. To prepreči izgubo morebitno pomembnih informacij, kar se lahko zgodi tekom transformacijskega koraka v klasičnem ETL procesu. Ta način zagotavlja analitikom in ostalim uporabnikom nenehen dostop do prvotnih podatkov, katere lahko transformiramo in obdelamo v skladu s potrebami analitike.

1.2 Tehnologije znotraj poslovnointeligenčnega sistema

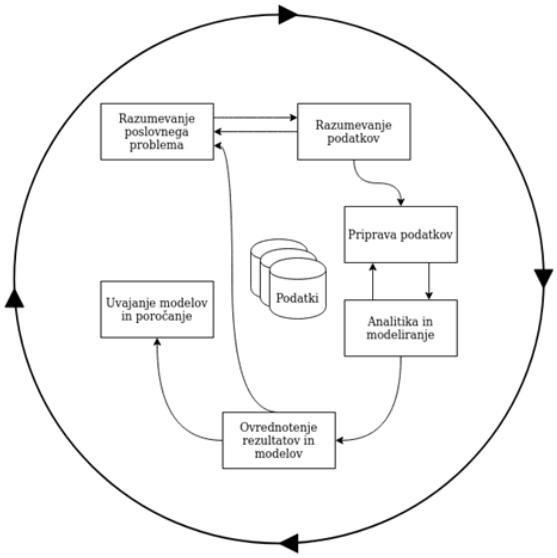
Glavne tehnologije PIS lahko razvrstimo glede na njihovo zahtevnost izvedbe in dodano vrednost, ki jo prinašajo poslovnemu subjektu. Eckerson (2007, str. 5) tehnologije razdeli na štiri različne skupine, ki rešujejo svojevrstne probleme. To so poročanje, ki nam pomaga odgovoriti, kaj se je zgodilo, analiza, ki pomaga odgovoriti, zakaj se je zgodilo, nadzor, ki odgovori, kaj se trenutno dogaja, in napovedna analitika, ki nam pomaga ugotoviti, kaj se bo zgodilo. V poročanje uvrščamo predvsem izvajanje poizvedb in poročil, v analizo večdimenzionalno analitiko in različne vizualizacije podatkov. V nadzor spadajo interaktivne plošče (angl. dashboard), ki omogočajo hitro interakcijo uporabnikov z različnimi grafičnimi prikazi in tabelami. V napovedno analitiko pa lahko uvrščamo uporabo različnih statističnih metod v okviru podatkovnega rudarjenja in strojnega učenja.

1.3 Proces podatkovno analitičnega projekta

V splošnem lahko proces podatkovne analitike razdelimo na šest ključnih aktivnosti. To so poznavanje poslovnega problema, poznavanje zbranih podatkov, priprava podatkov, analiza podatkov in modeliranje, ovrednotenje uporabljenih modelov in uvajanje rezultatov v poslovni proces ali poročanje rezultatov.

Slika 2 prikazuje model CRISP-DM (angl. Cross-industry standard process for data mining), ki opredeljuje standardni model cikla analitičnega projekta, bolj natančno projekta podatkovnega rudarjenja. Podatkovno rudarjenje spada pod okrilje podatkovne analitike. Je izraz s katerim opišemo proces odkrivanja vzorcev v podatkih. Uporablja se predvsem pri izvedbah kompleksnejše vrste analitike, kot je napovedna analitika in klasifikacija. Omeniti je potrebno še, da Wirth in Hipp (2000) poudarjata, da zunanji krog v CRISP-DM modelu prikazuje cikličnost samega projekta, češ da ob zaključku projekta pridobimo nova spoznanja in odkritja, ki ponovno pripeljejo do novih vprašanj, na katera želimo odgovoriti.

Slika 2: CRISP-DM metodologija podatkovno analitičnega projekta



Vir: Prirejeno po Wirth & Hipp (2000).

1.3.1 Razumevanje poslovnega problema

Začetna faza projekta je natančno razumevanje poslovnega problema, ki ga želimo rešiti. Potrebno je vedeti, kako se problema lotiti in kako ga rešimo s pomočjo zbranih podatkov in ostalimi resursi, ki so nam na voljo. Glavne naloge, ki jih je potrebno pri tej aktivnosti opraviti so natančna opredelitev poslovnih ciljev, ki jih s projektom želimo uresničiti. Sledi naloga opredelitve trenutne situacije, znotraj katere moramo upoštevati resurse, ki so nam na voljo, morebitna tveganja, zastaviti moramo potrebe in omejitve in izvesti analizo stroškov in koristi. Prav tako je potrebno začrtati plan celotnega projekta in opredeliti orodja in metode, ki bi jih lahko uporabili za uspešno izvedbo. Na podlagi omenjenih nalog lahko nato gradimo z naslednjimi aktivnostmi (Wirth & Hipp, 2000).

1.3.2 Poznavanje zbranih podatkov

Druga faza v procesu je faza poznavanja zbranih podatkov. Naloge, ki spadajo v ta del procesa so zbiranje podatkov, opis podatkov, raziskovanje zbranih podatkov in ovrednotenje

njihove kvalitete. Tukaj je ključnega pomena izvedba poročil za vsako izmed omenjenih nalog. Faza je ključna, saj se tako seznanimo s podatki in zaznamo morebitne zanimivosti ter tvorimo nove hipoteze za skrite informacije. Faza je močno povezana s fazo razumevanja poslovnega problema, kajti le z dobrim poznavanjem zbranih podatkov lahko začrtamo plan in metode dela, ki jih bomo uporabili pri nadaljnjih analizah (Wirth & Hipp, 2000).

1.3.3 Priprava podatkov

Tretja aktivnost po vrsti sledi priprava podatkov. Predstavlja začetni del konkretne analize, kajti od same priprave je odvisna kvaliteta in zmožnost nadaljnjih analiz. Podatkovni seti, ki jih zberemo v začetni fazi niso primerni za takojšnjo uporabo v procesu modeliranja in analitike, zato je glavna naloga te faze pripraviti in oblikovati podatke v tako strukturo, da bo primerna za vnos v analitične modele. Sivakumar in Gunasundari (2017, str. 788) kot glavne naloge znotraj aktivnosti priprave podatkov opredeljujeta čiščenje, integracijo, transformacijo in redukcijo podatkov. Hkrati fazo priprave opisujeta kot ključen proces z vidika celotnega analitičnega projekta, kajti kvalitetne priprave izboljšajo učinkovitost in kvaliteto samih analiz.

1.3.4 Analiza podatkov in modeliranje

Eckerson (2007, str. 14) opisuje izvedbo analitičnih modelov v okviru napovedne analitike. Ti zahtevajo uporabo različnih algoritmov, to so lahko regresijski modeli, nevronske mreže in ostale metode podatkovne analitike in rudarjenja. Vhodni podatki uporabljenega modela imajo znano odvisno spremenljivko, ki jo želimo napovedati. Nato podatkovni set razdelimo na učeči del, kjer bo odvisna spremenljivka znana, in testni del, kjer odvisna spremenljivka ne bo znana, in jo bo izbran algoritem napovedal. Napovedane rezultate testnega dela nato primerjamo z resničnimi vrednostmi in ovrednotimo uspešnost napovedi.

Eckerson (2007, str. 14) poleg tega omenja, da je proces modeliranja ponavljajoč proces, ki sestoji iz nenehnega učenja, testiranja in ovrednotenja posameznega modela. Prav tako je potrebno vedno znova preverjati različne kombinacije spremenljivk, ki jih bomo vnašali v model, ter poiskati odnose med njimi.

1.3.5 Ovrednotenje uporabljenih modelov

Faza ovrednotenja modelov zahteva, da rezultate, ki jih podajo analize natančno pregledamo in ovrednotimo na podlagi tega, če zadovoljujejo poslovne potrebe in zagotavljajo poslovni uspeh. Poleg tega je potrebno pregledati tudi celoten proces in ugotoviti, če se dosegajo kriteriji in omejitve, ki so bili določeni v prvi fazi (Wirth & Hipp, 2000).

1.3.6 Uvajanje modelov ali poročanje rezultatov

Zadnja faza lahko zajema predstavitev modela, implementacijo modela v poslovnointeligenčna poročila ali implementacijo v aplikacijo. Skratka, potrebno je uvajanje

vseh spoznanj in analiz v sam poslovni proces. To so lahko npr. različni procesi vrednotenja tržnih kampanj, strateških in številnih poslovnih odločitev. V to fazo sodi tudi proces izvedbe poročil o rezultatih in poteku celotnega projekta. Prav tako je potrebno natančno poročati o tem, katera orodja in podatke smo uporabili, ter metode, ki smo jih pri delu izvajali. Pomembno je tudi natančno prikazati delovanje končne rešitve (Chen, 2001, str. 48).

2 PRIPRAVA PODATKOV V PROCESU PODATKOVNE ANALITIKE

V poglavju 1.3.3 sem zapisal pomembnost kakovostne priprave podatkov za uspešno izvedbo analitičnega projekta. Dobra izvedba faze priprave podatkov je ključna, če želimo kvalitetne in zanesljive rezultate nadaljnjih analiz. Sivakumar in Gunasundari (2017, str. 785) tudi navajata, da faza priprave podatkov v kompleksnejših projektih, predvsem, ko imamo opravka z napovedno analitiko in podatkovnim rudarjenjem, lahko zajema do 80% časa, ki ga namenimo izvedbi celotnega projekta.

Proces obdelave podatkov za namene nadaljnjih analiz lahko razdelimo na štiri glavne aktivnosti. To so čiščenje, integracija, transformacija in redukcija podatkov (Sivakumar & Gunasundari, 2017, str. 786). Enako delitev omenjata tudi Han in Kamber (2001, str. 105-108). Nekateri v sam proces priprave vključujejo tudi proces predhodnega raziskovanja podatkov.

Poudaril bi, da se pri opisu in predstavitvi nalog priprave podatkov ne bom spuščal v vse podrobnosti, kajti področje je izredno široko, prav tako je proces priprave močno odvisen od posameznega projekta in vrste analiz in modelov, ki jih bomo uporabljali. Npr. napovedna analitika in različni modeli podatkovnega rudarjenja zahtevajo več oz. drugačno pripravo, kot npr. enostavna opisna in pojasnjevalna analitika. Predstaviti želim torej okvirno, katere so glavne in splošne aktivnosti priprave podatkov, ki jih lahko uporabljamo za namene najrazličnejših vrst podatkovne analitike, od opisne in pojasnjevalne do kompleksnejših napovednih analitik in podatkovnega rudarjenja.

2.1 Proces pridobivanja in raziskovanja podatkov

Stodder (2016, str. 8) pod aktivnostjo pridobivanja ustreznih podatkov vključuje iskanje, zbiranje in odkrivanje podatkov iz najrazličnejših virov in platform. Hkrati poudarja, da je potrebno avtomatizirati proces pridobitve redno uporabljenih podatkovnih setov.

Hellerstein, Heer in Kandel (2018, str. 25) navajajo, da je pred kakršnimkoli začetkom priprave podatkov potrebno tudi raziskati s kakšnimi podatki imamo pravzaprav opravka. Torej raziskati strukturo podatkov, distribucijo numeričnih spremenljivk, razmerje pojavov kategoričnih spremenljivk, podatkovne tipe atributov in formate zapisa vrednosti ipd. Poudarjajo, da je pri tej aktivnosti lahko izredno koristna vizualizacija podatkov. Izredno pomembno je, da spoznamo podatke, kajti le tako lahko zagotovimo učinkovito pripravo in nadaljnjo analizo. Stodder (2016, str. 8) ob tej aktivnosti poudarja tudi ostale aktivnosti, kot

so ugotavljanje kvalitete podatkov (tj. delež manjkajočih vrednosti, napake pri beleženju, podvajanje podatkov itd.), izvedba opisno statistične analize podatkov, pregled in odkrivanje odnosov med podatkovnimi seti iz različnih virov, sodelovanje s poslovnimi uporabniki za določanje skupnih pravil in iskanje anomalij v podatkih s pomočjo vizualizacije podatkov.

Mawer (2017) navaja, da je raziskovalna analiza podatkov (angl. exploratory data analysis) v širšem pomenu proces uporabe različnih vizualizacijskih in kvantitativnih metod za namene spoznavanja in opisa podatkovnega seta, brez predpostavk o njegovi vsebini. Analitikom omogoča, da podatke pravilno interpretirajo in, da so uporabni za analize in reševanje poslovnih problemov. Nekatere najpogosteje uporabljene metode znotraj procesa so lahko pregled distribucij vrednosti posameznih atributov, pregled opisne statistike atributov, vizualizacija odvisnosti med dvema ali tremi atributi s pomočjo grafov raztrosa in uporaba metod gručenja za kategorizacijo podatkovnih zapisov.

2.2 Aktivnosti čiščenje podatkov

Podatki, ki pridejo v proces obdelave so v veliko primerih neurejeni. Torej so nekonsistentni, vsebujejo prazna polja in osamelce (angl. outliers). Vse to lahko močno oteži nadaljnje delo, zato se je tovrstnih problemov treba lotiti sistematično. Han in Kamber (2001, str. 109) kot glavne naloge čiščenja podatkov vključujeta iskanje in pravilno obravnavo manjkajočih vrednosti, odpravo šuma v podatkih, iskanje osamelcev in odpravo neskladnosti in ostalih nepravilnosti.

2.1.2 Obravnavanje manjkajočih vrednosti

Manjkajoče vrednosti atributov je potrebno pravilno obravnavati oz. nadomestiti. Obstajajo različni načini, kako lahko rešimo problem. Han in Kamber (2001, str. 109) kot ene izmed rešitev za obravnavo manjkajočih podatkov opredeljujeta sledeče:

- Uporaba povprečne vrednosti ali mediane vseh vrednosti atributa.
- Uporaba vrednosti z največjo verjetnostjo pojava v atributu.
- Uporaba povprečne vrednosti ali mediane tistih vrstic, ki spadajo v enak razred (npr. ne vnesemo samo aritmetične sredine vseh vrednosti atributa, ampak izberemo le tiste vrednosti, ki jih lahko razvrstimo v isti razred. V primeru atributa "višina osebe", pogledamo najprej spol osebe in nato v atribut višine vnesemo aritmetično sredino populacije enakega spola v podatkovnem setu.
- Izpust vrstice, kjer imamo manjkajoči podatek.
- Uporaba primernih napovednih modelov (npr. linearna regresija) za napovedovanje manjkajoče vrednosti.

Metode, ki so na koncu izbrane so močno odvisne od deleža manjkajočih podatkov, količine podatkov in vsebine, ki jo podatki predstavljajo. Poleg tega je izbira ustrezne metode odvisna tudi od vrste analiz, ki jih bomo izvajali.

Kotsiantis, Kanellopoulos in Pintelas (2006, str. 112) poudarjajo, da je pri obravnavanju manjkajočih vrednosti potrebno upoštevati tudi razloge za nastanek manjkajoče vrednosti. Vrednost lahko manjka, ker je bila izgubljena ali pozabljena. Lahko se atribut ne navezuje na določen zapis, torej vrednosti ne moremo pripisati. Lahko pa je vrednost le nepomembna in zato ni zapisana.

2.1.3 Obdelava in iskanje osamelcev

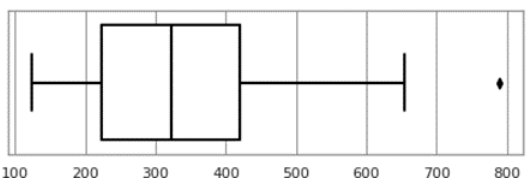
Osamelec je zapis v podatkovnem setu, ki se drastično razlikuje od ostalih zabeleženih vrednosti. Mnogokrat predstavlja napako pri merjenju ali beleženju podatkov. V primeru, da osamelce pustimo v setu nespremenjene, lahko vplivajo na pravilnost rezultatov končnih analiz, zato jih je potrebno optimalno obravnavati. Kwak in Kim (2017, str. 407-411) poudarjata tri glavne postopke obravnavanja zaznanih osamelcev. To so odstranjevanje, nadomestitev vrednosti (s tem zmanjšamo vpliv osamelcev) in ocenitev vrednosti z uporabo statističnih modelov.

Metod zaznavanja osamelcev je ogromno, zato bom predstavil le dve, ki ju lahko uporabimo v procesu zaznavanja osamelcev. Prva je uporaba metode medkvartilnega intervala, ki jo v svojem delu poudarjata Kwak in Kim (2017, str. 407-411). Druga predstavljena metoda pa je DBSCAN (angl. Density based spatial clustering of applications with noise), ki jo natančneje predstavita Han in Kamber (2001, str. 363-365). Widmann, Heine in Silipo (2018) ti dve metodi med drugimi ovrednotijo kot najbolj uporabljene in tradicionalne tehnike zaznavanja osamelcev.

Uporaba metode medkvartilnega intervala

Metoda je v nekaterih delih lahko zapisana s kratico IQR (angl. interquartile range). Gre za preprosto metodo s katero je možno zaznati osamelce na podlagi uvrstitve podatkov v ustrezne kvartile. Metodo se lahko uporablja na enodimenzionalnih podatkih. Vrednost IQR predstavlja razliko med tretjim (Q3) in prvim (Q1) kvartilom, IQR = Q3 - Q1. Ekstremne vrednosti, ki so lahko osamelci so posledično vrednosti, ki so manjše od $Q1 - 1.5 \times IQR$ ali večje od $Q3 + 1.5 \times IQR$. Tovrstne osamelce se lahko enostavno prikaže z grafom ti. škatle z brki (angl. boxplot), ki je prikazan na spodnji sliki 3. Osamelci so posamično označeni s točko, navpična črta znotraj škatle predstavlja mediano vseh vrednosti atributa, levi rob škatle označuje prvi kvartil, desni rob označuje tretji kvartil, konca ti. brk pa predstavljata vrednosti $Q1 - 1.5 \times IQR$ in $Q3 + 1.5 \times IQR$.

Slika 3: Grafični prikaz škatle z brki



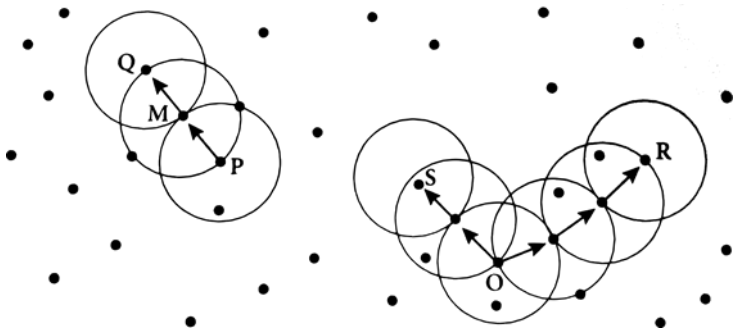
Vir: Lastno delo.

Uporaba metode gručenja DBSCAN

Gručenje je postopek razdelitve podatkov v skupine glede na njihovo podobnost. Podatki znotraj posamezne skupine morajo biti podobni, medtem ko se morajo posamezne skupine razlikovati med seboj (Provost & Fawcett, 2013, str. 164).

Han in Kamber (2001, str. 111, 363-365) navajata, da so različni algoritmi gručenja učinkoviti za proces odkrivanja osamelcev. Eden izmed teh je DBSCAN oz. gručenje na podlagi porazdelitvene gostote podatkovnih točk. Za delovanje algoritma sta ključna dva parametra, maksimalna dovoljena razdalja med sosednjimi točkami (oznaka ε) in minimalno dovoljeno število sosednjih točk (oznaka MinPts). Algoritem poišče posamezne gruče tako, da pri vsaki podatkovni točki pregleda število sosednjih točk, ki ležijo znotraj območja ε. V primeru, da je sosednjih podatkovnih točk vsaj MinPts, potem jo obravnavamo kot osrednjo točko gruče (angl. core object). Če pa je sosednjih manj kot MinPts točk, vendar je vsaj ena izmed njih označena kot osrednja, potem jo obravnavamo kot robno točko gruče. Gruče torej sestavlja maksimalno število med sabo povezanih točk. Vse podatkovne točke, ki niso zaznane kot del ene izmed gruč, so ovrednotene kot šum ali osamelec. Primer opisane metode gručenja je prikazan na spodnji sliki 4, kjer ima parameter MinPts vrednost 3, parameter ε pa je predstavljen z radijem kroga. Med označenimi točkami, M, P, O in R predstavljajo osrednje točke. Točke O, R in S so med sabo povezane in so del skupne gruče.

Slika 4: Grafični prikaz delovanja algoritma DBSCAN



Vir: Han & Kamber (2001).

2.3 Aktivnosti transformacije podatkov

Transformacija podatkov predstavlja proces preoblikovanja podatkov v obliko primerno za izvajanje analitičnega procesa. Han in Kamber (2001, str. 114) v kontekstu naprednejše analitike in podatkovnega rudarjenja predstavita pet glavnih aktivnosti, ki jih izvajamo znotraj procesa transformacije. To so glajenje, agregacija, generalizacija, normalizacija in izpeljevanje oz. konstrukcija novih dimenzij.

2.3.1 Normalizacija

Normalizacija je metoda, ki vrednosti atributa razporedi na skupni skali, kar igra pomembno vlogo pri pravilnosti analitičnih modelov. Han in Kamber (2001, str. 114-115) poudarjata močan vpliv normalizacije na računske hitrosti analitičnih modelov in preprečevanje, da bi uporabljeni modeli pripisali večji pomen atributom z večjimi intervali med svojimi vrednostmi, kot tistimi, ki imajo manjše intervale med vrednostmi. Dve najpogosteje uporabljeni metodi normalizacije sta metoda min-max in metoda z-vrednosti.

Metoda min-max je prikazana s splošno enačbo (1). X' predstavlja normalizirano vrednost, ki ima lahko vrednost med 0 in 1. Najmanjša vrednost atributa, označena z X_{min}, dobi vrednost 0, največja, označena z X_{max}, pa dobi vrednost 1. Oznaka X pa predstavlja vrednost, ki jo želimo normalizirati.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Splošen izračun z-vrednosti je prikazan z enačbo (2), ki normalizira vrednosti glede na število standardnih odklonov od povprečne vrednosti. Oznaka μ predstavlja aritmetično sredino vrednosti atributa, σ pa označuje standardni odklon vrednosti atributa. X' označuje normalizirano vrednost, X pa vrednost, ki jo želimo normalizirati.

$$X' = \frac{X - \mu}{\sigma} \tag{2}$$

2.3.2 Sestavljanje in konstrukcija novih dimenzij

Aktivnost sestavljanja in dodajanja novih dimenzij oz. atributov opisuje postopek, ko že obstoječemu podatkovnemu setu dodamo nov stolpec z novimi vrednostmi, ki smiselno dopolnijo naše podatke. Han in Kamber (2001, str. 116) pišeta, da združevanje dimenzij in sestavljanje novih lahko pripelje do odkrivanja novih spoznanj in odnosov med posameznimi atributi.

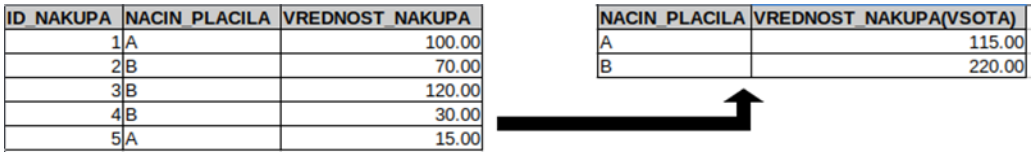
2.3.3 Agregacija in generalizacija

Agregacija podatkov je proces, kjer združimo kategorične atribute in jim pripišemo agregirane vrednosti opisne statistike. To so lahko aritmetična sredina, vsota, minimum, maksimum ipd. Han in Kamber (2001, str. 116) omenjata, da sta tako agregacija in generalizacija metodi, ki ju lahko uporabimo tudi pri redukciji podatkov.

Slika 5 prikazuje primer agregacije. Vrednosti v desni tabeli so združene po atributu "NAČIN_PLAČILA". Atributu "VREDNOST_NAKUPA (VSOTA)" smo dodali vsoto vseh vrednosti atributa "VREDNOST_NAKUPA" iz desne tabele. Agregacija podatkov tako

omogoča izpeljavo novih vrednosti, katere lahko v nadaljevanju obdelujemo in uporabljamo pri analizah.

Slika 5: Primer agregacija podatkov po izbranem atributu



ID_NAKUPA	NACIN_PLACILA	VREDNOST_NAKUPA
1	A	100.00
2	B	70.00
3	B	120.00
4	B	30.00
5	A	15.00

NACIN_PLACILA	VREDNOST_NAKUPA(VSOTA)
A	115.00
B	220.00

Vir: Lastno delo.

Generalizacija pa je proces s katerim zamenjamo vrednosti atributov na nižjih ravneh z vrednostmi na višjih ravneh oz. pomaknemo se po hierarhiji navzgor. Enostaven primer metode generalizacije je, ko vrednost prodaje po dnevih generaliziramo na prodajo v tednih, mesecih ali letih. Leto v tem primeru predstavlja najvišji nivo in tako zajema vsoto vseh podatkov, ki so vključeni v nižjem nivoju.

2.3.4 Metode glajenja

V primerih, ko imamo opravka s podatki časovnih vrst, Janert (2011, str. 84-90), kot najenostavnejšo metodo opredeljuje metodo drseče sredine, ki določeno število predhodnih vrednosti nadomesti s povprečno vrednostjo le-teh. Naslednja je metoda tehtane drseče sredine, kjer posamezno predhodno vrednost pomnožimo s pripadajočo utežjo (vsota vseh uporabljenih uteži mora biti 1) in s tem določimo moč vpliva posamezne vrednosti pri končnemu izračunu. Kot učinkovitejše metode poudarja tudi različne variacije eksponentnega glajenja, ki jih zaradi obsežnosti ne bom dodatno opredeljeval

Han in Kamber (2001, str. 110-112) za glajenje poudarjata tudi metode, kot so gručenje (ang. clustering), vnašanje vrednosti v regresijsko funkcijo in združevanje v zaboje (angl. binning). Metoda združevanja v zaboje zahteva, da številske vrednosti atributa najprej razvrstimo po velikosti. Nato lahko vse vrednosti razdelimo na poljubno število skupin ti. zabojev, ki vsebujejo enako število elementov. Na koncu vsako vrednost v zaboju spremenimo v aritmetično sredino vseh vrednosti v zaboju. Uporabimo lahko tudi robni vrednosti zaboja in vmesnim vrednostim pripišemo tisto robno vrednost, ki je bližje.

Naslednja možnost glajenja pa je, da podatke, ki jih želimo zgladiti, vstavimo v regresijsko funkcijo. Najosnovnejša metoda regresije je enostavna linearna regresija, z enačbo $y = \alpha + \beta x$, ki ji nastavimo parametra tako, da se najbolje prilega podatkovnim točkam dveh spremenljivk. Za določanje najboljšega prileganja funkcije največkrat uporabljamo metodo vsote najmanjših kvadratov, kar pomeni, da morajo biti parametri linearne funkcije taki, da minimiziramo vsoto vseh kvadratov razlik med podatkovnimi točkami in vrednosti linearne funkcije.

2.2 Integracija podatkov

Integracija podatkov je aktivnost, ki zajema združevanje podatkov iz različnih virov v skupno celoto. V procesu lahko naletimo na težave, kot so problem identifikacije entitet. Problem identifikacije entitet nastopi takrat, ko združujemo entitete, kjer so enaki atributi različno poimenovani. Z namenom preprečevanja nastanka tovrstnih težav, moramo skrbno pregledati in uporabiti metapodatke podatkovne baze oz. podatkovnega skladišča. Naslednji je problem pojava redundance podatkov, kjer so redundantni podatki tisti, ki jih lahko izpeljemo iz drugega podatkovnega seta. Tretji primer pa je neskladje z istimi entitetami, ki prihajajo iz različnih virov. Vrednosti atributov so lahko zapisani v različnih formatih, kar posledično predstavlja problem pri združevanju. Primer tovrstnega neskladja je lahko zapis dnevne prodaje v različnih valutah (Han & Kamber, 2001, str. 112-114).

2.4 Aktivnost redukcije podatkov

Han in Kamber (2001, str. 115-116) poudarjata, da je izvajanje naprednejše podatkovne analitike in podatkovnega rudarjenja na prevelikih podatkovnih setih lahko nepraktično in počasno, zato je potrebno podatkovni set zmanjšati na način, da ohranimo njegovo integriteto. Tako zagotovimo učinkovito izvajanje procesa in enake analitične rezultate. Nekatere glavne metode redukcije so agregacija podatkov, zmanjševanje dimenzionalnosti oz. števila atributov podatkovnega seta, zmanjševanje številčnosti oz. števila vrstic, metode stiskanje podatkov in diskretizacija podatkov. Natančneje predstavita tudi druge metode in pristope, ki pa jih v nalogi zaradi same obsežnosti ne bom dodatno predstavil.

3 ORODJA ZA PRIPRAVO IN OBDELAVO PODATKOV

Orodij za pripravo podatkov in podatkovno analitiko je ogromno. Širok nabor možnosti nam velikokrat oteži odločitev, katero orodje izbrati za določen problem. V primeru, ko potrebujemo specializirane in nestandardne rešitve, moramo poseči po pisanju lastne programske kode. Potrebno je poudariti, da ne potrebujemo vedno visoko specializiranih rešitev, zato so na voljo tudi različna ti. samopostrežna orodja s prilagojenim grafičnim vmesnikom, ki poskušajo proces priprave podatkov olajšati in ga približati tudi tistim uporabnikom, ki nimajo znanja programiranja ali pa le hočejo hitre rezultate brez dodatnega in včasih tudi nepotrebnega truda.

3.1 Samopostrežna orodja in uporaba programskih jezikov

Stodder (2016, str. 25-27) navaja, da je bistvo samopostrežnih specializiranih orodij za pripravo podatkov predvsem omogočanje poslovnim uporabnikom, da samostojno opravijo potrebne aktivnosti priprave podatkov za potrebe poslovne inteligence in podatkovne analitike. Samopostrežna orodja poskušajo avtomatizirati in poenostaviti izvedbo aktivnosti pridobivanja podatkov iz različnih virov, čiščenja in transformacije. S tem posledično tudi razbremenijo posredovanje specializiranega kadra na področju dela s podatki.

Hellerstein, Heer in Kandel (2018, str. 23-24) prav tako poudarjajo, da je tradicionalno obstajal razkol med uporabniki, ki najbolje poznajo podatke in njihovo uporabno vrednost, ter ljudmi, ki imajo znanje za pripravo podatkov po klasičnem pristopu programiranja lastnih rešitev. To dejstvo je posledično povzročalo številne nesporazume in zapravljanje časa pri usklajevanju potreb in dela uporabnikov z različnih oddelkov.

Kimball in Caserta (2004, str. 10-13) primerjata uporabo specializiranih ETL orodij z uporabo programskih jezikov oz. ročnega razvoja procesov. Nekatere prednosti specializiranih orodij, ki jih navajata so enostavnejši in hitrejši razvoj procesa, enostavnejša povezava s številnimi podprtimi podatkovnimi viri in možnost dela z orodjem brez znanja programiranja. Prednosti pri uporabi programskih jezikov pa poudarjata predvsem visoko stopnjo fleksibilnosti in neomejenost pri funkcionalnosti.

3.2 Obravnavani kriteriji

Glavne kriterije, ki jih bom upošteval pri primerjavi orodij za pripravo podatkov, bodo določeni na način, da bodo temeljili na nekaterih glavnih ciljih procesa priprave podatkov, ki jih opisuje Stodder (2016, str. 8-9). To so iskanje in pridobivanje ustreznih podatkov, spoznavanje podatkov in vzpostavljanje osnovnega znanja o njih, integracija in izboljšanje pridobljenih podatkov s čiščenjem in transformacijo, ponovno uporaba podatkov in deljenje pridobljenega znanja o njih v različnih okoljih.

Stodder (2016, str. 8-9) poleg tega navaja, da je proces priprave podatkov težko natančno definirati, kajti vsebuje lahko številne korake in različne cilje, ki so odvisni od potreb posameznih uporabnikov. Kot glavne kriterije pri primerjavi bom upošteval in predstavil sledeče:

1. Funkcionalnost pri opravljanju nalog priprave podatkov in fleksibilnost orodja
    a. Zmožnost izvajanja različnih aktivnosti priprave podatkov opisanih v zgornjih poglavjih.
    b. Zmožnost dodajanja funkcionalnosti z vtičniki oz. nadgradljivost orodja.
    c. Integracija z ostalimi analitičnimi orodji.
    d. Hitrost in učinkovitost orodja pri delu priprave podatkov in zmožnost optimalnega dela z večjimi količinami podatkov.
2. Povezljivost s podatkovnimi viri
    a. Podprti podatkovni formati.
    b. Možnosti vzpostavljanja povezave s podatkovnimi bazami in oblačnimi platformami.
    c. Možnosti nadgraditev za branje privzeto nepodprtih podatkovnih formatov.
3. Uporabniška izkušnja
    a. Kakovost grafičnega vmesnika.
    b. Krivulja učenja.
    c. Kvaliteta in količina virov uporabniške podpore.

3.3 Izbrana orodja

V nalogi se bom lotil natančnejšega opisa in predstavitve treh različnih orodij, Python (Python, 2019), Knime analitična platforma (Knime, 2019) in Microsoft Power Query znotraj Microsoft Power BI Desktop (Microsoft Power BI, 2020), s katerimi se lahko lotimo procesa priprave podatkov. Vsako orodje zahteva drugačen začeten pristop in drugačen nivo predhodnega znanja.

Prvo orodje, ki ga bom obravnaval, je programski jezik Python z nekaterimi glavnimi knjižnicami za delo s podatki. Python je interpretiran, dinamičen, objektno usmerjen in visoko nivojski programski jezik. Ima izredno berljivo in relativno enostavno sintakso. Poleg uporabnosti na področjih spletnega razvoja in razvoja splošne programske opreme, je Python v zadnjih letih postal izredno priljubljeno orodje na številnih področjih podatkovne znanosti. V svojem ekosistemu ima ogromno visokokakovostnih in zrelih knjižnic, ki orodju dodajajo veliko funkcionalnosti za učinkovito delo s podatki. Prednost uporabe orodja, kot je Python je predvsem večja stopnja fleksibilnosti, hitrost procesiranja podatkov in hkrati skoraj neomejen nabor funkcij, ki jih lahko izvajamo. Podobno orodje, ki je prav tako zrelo in uporabljeno na področju podatkovnih znanosti je programski jezik R, ki je bil ustvarjen z namenom izvajanja statističnih operacij in dela s podatki. S Python-om si delita ogromno podobnosti, oba jezika oz. orodja imata izvrsten nabor kvalitetnih knjižnic, veliko skupnost uporabnikov in dobro uporabniško podporo. Tako kot Python je tudi R odprtokoden. V letni anketi (Piatetsky, 2019) najbolj uporabljenih orodij na področju podatkovne znanosti in analitike, ki jo izvaja priznana spletna publikacija Kdnuggets, so ugotovili, da je 65,8% vseh anketirancev v letu 2019 za potrebe podatkovne analitike in podatkovne znanosti vsaj na enem projektu uporabilo programski jezik Python, R pa 46,6%.

Drugo orodje, ki ga bom predstavil je Knime analitična platforma. To je samopostrežno odprtokodno programsko orodje, napisano v Javi, ki omogoča kvalitetno pripravo podatkov, kot tudi izvajanje zahtevnejših analitičnih procesov. Knime platforma poleg tega, da je opremljena z velikim spektrom privzetih funkcij za delo s podatki, omogoča tudi visoko stopnjo nadgradljivosti z uvažanjem različnih vtičnikov in tako zagotavlja dokaj visoko mero funkcionalnosti. Vtičnike razvijajo razvijalci ekipe Knime in njihova skupnost odprtokodnih razvijalcev. Anketa Kdnuggets (Piatetsky, 2019) je pokazala, da je Knime analitično platformo v letu 2019 za delo na področju podatkovne znanosti in analitike uporabilo 10,7% anketirancev. Podobno samopostrežno orodje za izvajanje analitike in priprave podatkov je tudi RapidMiner platforma, ki pa napram Knime ni v celoti brezplačen in popolnoma odprtokoden, namreč brezplačna različici orodja ima precej veliko omejitev, ki pa jih lahko odpravimo z nakupom programa.

Tretje orodje, ki bo vključen v primerjavo, je Microsoft Power Query. Orodje je sicer brezplačno, vendar ni samostojno, kajti deluje v okolju Microsoft PowerBI platforme, Microsoft Excel in nekaterih drugih. Sam ga bom testiral znotraj PowerBI Desktop, ki je brezplačen. Priprava podatkov deluje interaktivno in je izredno podobna delu v klasičnih

orodjih za delo z razpredelnicami. Orodij, ki zahtevajo podoben pristop in način dela kot Power Query, je na trgu veliko. Med odprtokodnimi je to OpenRefine, ki pa je samostojno in popolnoma brezplačno.

3.4 Predstavitev primera

Pri testiranju orodij bom primarno delal s podatki, pridobljenih s portala OPSI. Bolj natančno, uporabil bom srednje velik podatkovni set zabeleženih meritev vseh tehničnih pregledov registriranih motornih vozil v Sloveniji v letu 2019 (Ministrstvo za infrastrukturo, brez datuma). Set ima preko 1,3 milijona vrstic in 170 atributov. V prvotni obliki je zapisan v dveh tekstovnih datotekah, ki ju je najprej potrebno združiti. Nato bom izločil vse atribute, katerih delež manjkajočih vrednosti presega 20 odstotkov. Hkrati bom odstranil tudi nekatere druge nepotrebne atribute. Odstranil bom tudi vse zapise, kjer motorno vozilo ni kategorizirano kot osebni avtomobil. Podatkovni set na ta način zmanjšamo na približno milijon vrstic in 16 različnih atributov z majhnim deležem manjkajočih vrednosti. Pri atributu, ki beleži znamko vozila, je več tekstovnih napak, kar pomeni, da je potrebno tovrstne napake odstraniti in nadomestiti s pravilno vrednostjo (npr. znamka Volkswagen je v nekaterih vrsticah zapisana kot Volkswagwn). To lahko storimo z uporabo algoritma, ki bo na podlagi razmerja Levenshteinove razdalje (ta meri različnost dveh besed na podlagi števila korakov vstavljanja, izbrisa in zamenjave posamezne črke) in skupne dolžine primerjanih tekstovnih zapisov izračunal stopnjo podobnosti. Tistim vrednostim znamk, ki presegajo določeno stopnjo podobnosti, bom pripisal ustrezno vrednost. Algoritma Lavenshteinove razdalje ne bom sam implementiral, ampak bom za ta namen uporabil ustrezno knjižnico. Zadnja aktivnost, ki jo bom izvedel je pregled osamelcev. Vse vrednosti numeričnih atributov bom vizualiziral s prikazom škatle z brki in na podlagi metode medkvartilnega intervala pridobil zaznane ekstremne vrednosti. Kot osamelce bom označil oz. ovrednotil le nemogoče vrednosti, ki so plod napake pri beleženju.

Spodaj je natančneje opisana izvedba algoritma, ki ga bom uporabil za popravljanje tekstovnih napak.

1. Vnos: Niz vseh različnih vrednosti atributa sortiranih po številu pojavov v celotnem setu (od največjega do najmanjšega).
2. Vsak element X1 znotraj vnosa primerjamo z vsakim od preostalih elementov X2. Potrebno je preveriti, da elementa nista že uporabljena.
3. X1 in X2 vnesemo v funkcijo, ki na podlagi razmerja Levenshteinove razdalje in skupne dolžine obeh besed izračuna stopnjo podobnosti. Za boljšo primerjavo, pred izvajanjem funkcije lahko obdelamo obe besedi na način, da vsebujeta le črkovni zapis, brez nepotrebnih znakov.
4. Vrednost X2 je potrebno označiti kot že uporabljeno, da preprečimo ponovno primerjavo.
5. V iteracijah, kjer stopnja podobnosti presega določeno mejo, vrednost X2 zamenjamo z vrednostjo X1.

3.5 Programski jezik Python

Funkcionalnost pri opravljanju nalog priprave podatkov in fleksibilnost

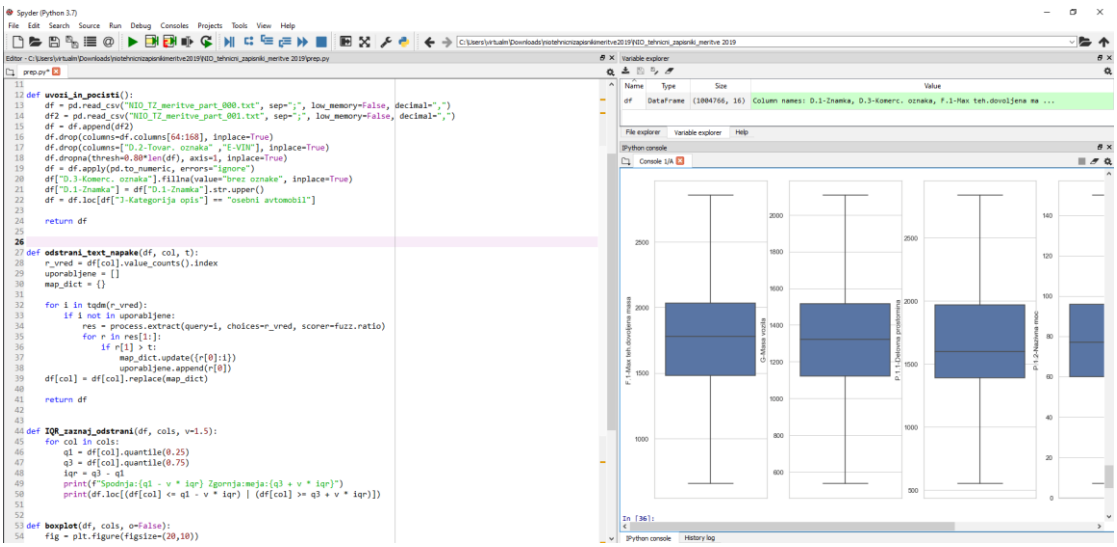
Python predstavlja izredno dobro izbiro v primerih, ko potrebujemo visoko stopnjo funkcionalnosti in fleksibilnosti. Eden izmed glavnih razlogov za porast uporabe Pythona na področju podatkovne znanosti in analitike je, da zaradi enostavne sintakse služi kot izredno dober vmesnik številnim knjižnicam in ogrodji za izvajanje napredne analitike, podatkovnega rudarjenja in dela z masivnimi podatki. Veliko analitičnih orodij (tudi Knime in Power Query) omogoča integracijo s Pythonom in tako uporabnikom ponudijo dostop do glavnih analitičnih knjižnic znotraj Python ekosistema. Na ta način zagotovijo izvajanje kompleksnejših in bolj specifičnih procesov priprave podatkov. Python služi kot vsestranski jezik in napram programskemu jeziku R ni specializiran za delo s podatki. Zaradi tega se za optimalno uporabo naslanjamo na nekaj standardnih knjižnic, kot sta Pandas (Pandas, 2019) in NumPy (Numpy, 2019), ki zagotavljata hitro in učinkovito izvajanje operacij obdelave podatkov. Pandas, poleg številnih funkcij za delo s podatki, v Python vključi podatkovno strukturo, poimenovano dataframe. To je objekt implementiran v sami knjižnici, ki omogoča shranjevanje podatkov v obliki klasičnih tabel in ponuja številne funkcionalnosti za učinkovito in hitro manipuliranje vsebujočih podatkov. Pandas vse prebrane podatke drži v fizičnem pomnilniku, kar pomeni, da ni učinkovito za obdelavo podatkovnih setov, ki presegajo količino pomnilnika, ki ga imamo na voljo. V tovrstnih primerih lahko podatke preberemo in obdelamo po več manjših delih, ali pa uporabimo druge knjižnice. Z uporabo ustreznih knjižnic lahko torej izvajamo najrazličnejše aktivnosti procesa priprave podatkov. Uporaba Pythona je s primernimi knjižnicami učinkovita tudi za namene raziskave in vizualizcije podatkov. Pomembno je poudariti še dejstvo, da imamo ob izvajanju vseh aktivnosti na voljo celotno moč programskega jezika, zato pomanjkanje fleksibilnosti in funkcionalnosti ne predstavlja nikakršnega problema. Funkcij priprave podatkov se lahko lotimo na več načinov, ter pri tem nimamo veliko omejitev, kako se zadeve lotiti. Slabost tega je, da včasih za izvedbo enostavnih in rutiniranih aktivnosti potrebujemo več napora in časa, kot ga porabimo za isti postopek v ostalih samopostrežnih orodjih. Za učinkovito pripravo podatkov je potrebno izbrati tudi primerno razvojno okolje. V primeru, da želimo interaktivno obdelovati zbrane podatke in ne želimo zagnati celotnega programa naenkrat, lahko uporabimo razvojno okolje kot sta JupyterLab in Spyder, ki omogočata poganjanje Python programske kode po posameznih celicah, ki jih lahko samostojno poženemo in sproti spremljamo rezultate izvedenih funkcij. Ta način se imenuje interaktivno programiranje in je priročen predvsem v postopku raziskovanja podatkov.

Uporabniška izkušnja

Uporabniška izkušnja napram samopostrežnim orodjem ni najboljša. Za optimalno pripravo podatkov mora uporabnik poznati koncepte programiranja in imeti znanje sintakse Pythona, kar je lahko za marsikatere enostavnejše in rutinirane oz. manj specifične probleme povsem nepotrebno. Razvojno okolje primerno za proces priprave podatkov je Spyder (Spyder,

2019), ki je prikazan na spodnji sliki 6. Na levi strani imamo tekstovno polje za pisanje kode. Pod zavihkom variable explorer lahko pogledamo vrednosti posameznih spremenljivk, kar je priročno za pregled tabel oz. dataframe objektov. Pod zavihkom help lahko pogledamo dokumentacijo knjižnic brez potrebe, da zapustimo programsko okolje. Okno v desnem spodnjem kotu pa je interaktivna konzola, ki omogoča pisanje kode v celicah, ki jih lahko samostojno poganjamo. Krivulja učenja je izredno strma, kar pomeni, da je potrebno veliko začetnega znanja, če želimo rešiti tudi najenostavnejše probleme znotraj procesa priprave podatkov. Kakovost grafičnega vmesnika je težko opredeliti, kajti močno je odvisna od posameznega razvijalskega okolja v katerem delamo. Prav tako mislim, da v primeru uporabe programskega jezika ne moremo ravno govoriti o grafičnem vmesniku, kajti vso funkcionalnost je še vedno potrebno pisati ročno. Uporabniška podpora pa je točka kateri zlahka dodelim velik plus. Omenil sem že, da ima Python izjemno skupnost uporabnikov, kar se kaže v številnih izvrstno napisanih knjižnicah in dokumentacijah, aktivnih portalih in forumih, namenjenim reševanju vseh vrst problemov, s katerimi se uporabnik sreča.

Slika 6: Uporabniški vmesnik razvojnega okolja Spyder



Vir: Lastno delo.

Povezljivost z viri podatkov

Povezljivost s podatkovnimi viri je s Pythonom izredno dobra, praktično brez omejitev. Na voljo je ogromno odprtokodnih knjižnic za branje različnih podatkovnih formatov, vzpostavljanje povezav s podatkovnimi bazami in programskih vmesnikov za pridobivanje podatkov iz oblačnih platform. Že omenjena knjižnica Pandas omogoča branje formatov kot so csv, xls/xlsx, json, xml in nekateri drugi. Za vzpostavljanje povezave z SQL podatkovnimi bazami lahko uporabimo knjižnico SQLAlchemy (SQLAlchemy, 2019), ki ponuja povezovanje s podatkovnimi bazami Oracle, PostgreSQL, MySQL, SQLite in MS SQL server. Glavna slabost Pythona na tem področju je predvsem ta, da je veliko funkcionalnosti

za povezovanje z različnimi viri razčlenjenih po različnih knjižnicah, ki jih je potrebno poiskati, pregledati njeno dokumentacijo in se seznaniti s programskim vmesnikom.

3.6 Knime analitična platforma

Funkcionalnost pri opravljanju nalog priprave podatkov in fleksibilnost

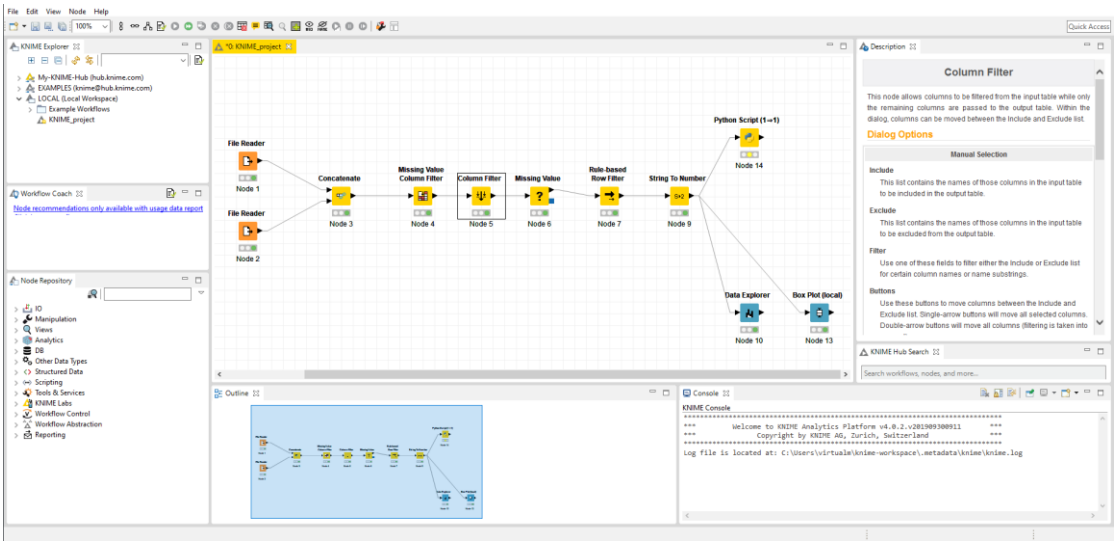
Izvajanje procesov v Knime temelji na povezovanju ti. vozlišč, nekateri tovrsten delovni tok poimenujejo tudi vizualno programiranje. Vsako vozlišče predstavlja določeno funkcijo, ki potrebuje nek vnos, nato ga na podlagi določenih parametrov obdela in vrne nov objekt. Parametre funkcije lahko nastavljamo v nastavitvah posameznega vozlišča. Osrednji del orodja predstavlja delovno platno, kamor postavljamo vozlišča in jih med sabo smiselno povezujemo. Celoten proces je torej ponazorjen z mrežo različnih vozlišč (ali funkcij). Rezultat tovrstnega delovnega toka je izredna preglednost, visoka stopnja fleksibilnosti, brez potrebe po pisanju lastne programske kode, in hkrati dokaj prijazna uporabniška izkušnja. Orodje ima širok nabor funkcij za pripravo in obdelavo podatkov, kot tudi izvajanje enostavne in napredne podatkovne analitike. Knime omogoča tudi učinkovito raziskovanje podatkov s funkcijami za vizualizacijo in statističen pregled uvoženih podatkovnih setov. Orodje v svoji privzeti različici zajema večino osnovnih funkcij za optimalno pripravo podatkov, ki sem jih tekom testiranja potreboval. V primerih, ko potrebujemo izvesti bolj specifične aktivnosti ali uvoziti podatke iz nepodprtih virov, pa Knime omogoča enostavno namestitev vtičnikov za dodajanje nove funkcionalnosti. Poleg tega omogoča tudi poganjanje lastnih skript v programskem jeziku Java in hkrati zagotavlja tudi dobro integracijo z orodjem Python (z namestitvijo Knime Python Integration vtičnika) in R (z namestitvijo Knime Interactive R Statistics Integration). Uporabnik lahko torej uporablja privzete funkcije, ki so v mnogih primerih priprave podatkov zadovoljive. Za bolj specifične naloge pa se lahko poseže po uporabi orodja Python ali R, brez potrebe, da zapustimo Knime. Uporaba programskih jezikov Python in R znotraj Knime je manj fleksibilna, izvajanje operacij pa počasnejše, kot če to delamo izključno s Pythonom ali R v ustreznem razvojnem okolju.

Uporabniška izkušnja

Uporabniški vmesnik je dober in novemu uporabniku omogoča učinkovito delo brez strme krivulje učenja. Spodnja slika 7 prikazuje delovno okolje Knime platforme. Osrednji del predstavlja delovno platno, kamor postavljamo vozlišča in tako sestavljamo proces oz. podatkovni tok. Vse funkcije, ki jih lahko uporabljamo so smiselno razporejene v oknu node repository v levem spodnjem kotu. Ob izbiri funkcije se nam v desnem oknu help izpiše njena natančna dokumentacija. Parametre posamezne funkcije določamo preko uporabniškega vmesnika v nastavitvah posameznega vozlišča. Krivulja učenja je položna, kajti celotno orodje je zgrajeno na način, da omogoča vsem potencialnim uporabnikom hitro seznanitev za delo z orodjem. Uporabniški vmesnik je izredno pregleden, delo z vozlišči pa je prav tako dovolj fleksibilno in zmogljivo za nove uporabnike. Za namene uporabniške

podpore je na voljo uradni Knime forum (Knime AG, brez datuma a), ki je aktiven in predstavlja dober vir za iskanje rešitev ob morebitnih težavah. Knime poleg tega ponuja tudi spletno mesto Knime Hub (Knime AG, brez datuma b), kjer uporabniki delijo lastne procese in primere uporabe vozlišč, s katerimi si lahko ostali uporabniki pomagajo in jih prenesejo.

Slika 7: Uporabniški vmesnik Knime analitične platforme



Vir: Lastno delo.

Povezljivost z viri podatkov

Privzete možnosti za uvoz podatkov so obsežne. Za branje datotek so na voljo funkcije, ki podpirajo podatkovne formate za shranjevanje tabelaričnih podatkovnih struktur csv in xls/xlsx, formate za shranjevanje slik in strukturirana podatkovna formata xml in json. Orodje omogoča tudi enostavno vzpostavljanje povezave z glavnimi vrstami relacijskih podatkovnih baz, to so Oracle, PostgreSQL, MySQL, SQLite, MS SQL server, MS Access in H2. Poleg tega Knime ponuja namestitev vtičnikov, ki omogočajo vnos nekaterih podatkovnih formatov, ki niso podprti v privzeti različici.

3.7 Microsoft Power Query

Funkcionalnost pri opravljanju nalog priprave podatkov in fleksibilnost

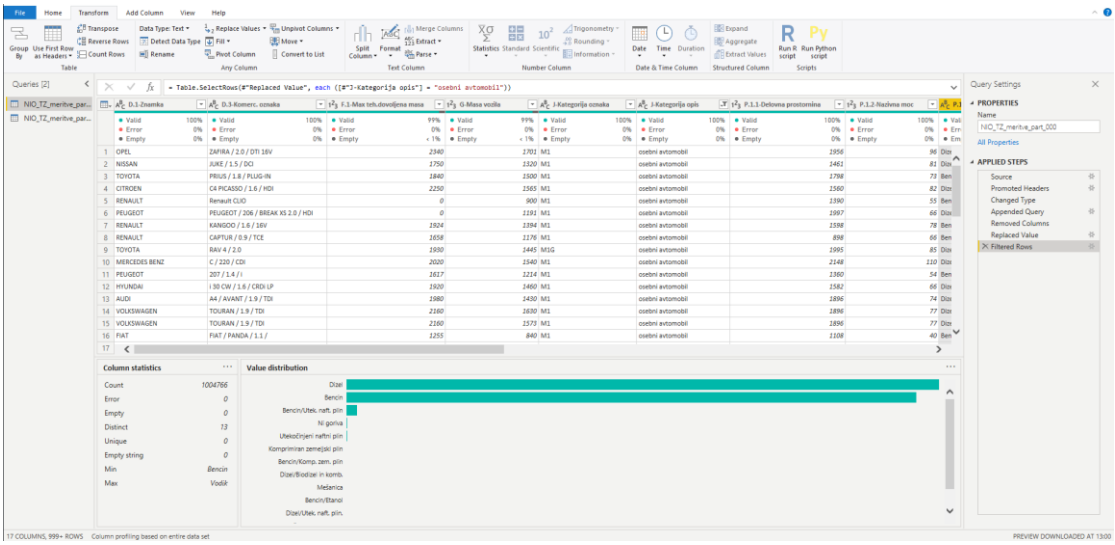
Delo v Power Query poteka interaktivno in uporabniku nenehno omogoča pregled nad obdelovanimi podatki. Potek dela nekoliko spominja na klasičen proces dela v razpredelniških orodjih, npr. Microsoft Excel, Libreoffice Calc in Google Spreadsheets. Glavna razlika je, da v Power Query izvajamo operacije po stolpcih in ne individualno po celicah. Orodje operacije izvaja na deležu podatkov, ne nad celotnim podatkovnim setom, kar omogoča dobro odzivnost med delom tudi pri večjih podatkovnih setih. Vse uporabljene operacije se šele po zaključku izvedejo nad celotnim setom. Funkcionalnost orodja za

namene priprave podatkov je dobra. Iskanje in nadomestitev manjkajočih vrednosti, aktivnosti transformacije podatkov kot so normalizacija in agregacija podatkov, izpeljevanje novih dimenzij so le nekatere izmed ključnih aktivnosti, ki jih lahko izvajamo. Vsi procesi, ki jih kot uporabniki izvajamo preko uporabniškega vmesnika, se v ozadju izvajajo preko programskega jezika Power Query M formula. Tekom dela se lahko poljubno vrnemo na katerokoli predhodno izvedeno funkcijo, jo modificiramo in nadaljujemo z delom. Spisek vseh izvedenih aktivnosti lahko uporabimo tudi v ostalih projektih, ki jih izvajamo znotraj Power Query orodja. Ta proces je sicer precej neintuitiven, kajti potrebno je odpreti napredni urejevalnik, kjer se v programskem jeziku Power Query M izpisujejo vse operacije, ki smo jih do sedaj izvedli. Te lahko nato kopiramo in uporabimo pri naslednjem procesu. Orodje omogoča tudi vklop možnosti prikaza opisne statistike posameznega atributa, distribucije njegovih vrednosti in hkrati ponuja vizualen pregled nad kvaliteto atributov, kar je priročno predvsem pri pregledu deleža manjkajočih vrednosti. Potrebno je poudariti, da so opisna statistika, distribucija in kvaliteta atributov izračunane na deležu podatkov in ne na celotnem podatkovnem setu. Power Query sicer ponuja možnost obravnave celotnega seta, vendar lahko to v primeru večje količine podatkov postane izredno počasno. Zato bi izpostavil, da orodje ni najbolj primerno za namene raziskovanja podatkov. Možnosti za vizualizacijo podatkov Power Query nima, kajti orodje je specializirano le za obdelavo podatkov. Fleksibilnost orodja je kljub nekaterim že omenjenih pomanjkljivosti dobra in uporabniku omogoča visoko stopnjo funkcionalnosti, predvsem za namene čiščenja, transformacije in integracije podatkov. Nekaj zaslug temu lahko pripišemo tudi integraciji s programskima jezikoma R in Python, ki omogočata dostop do ustreznih knjižnic znotraj njunega ekosistema.

Uporabniška izkušnja

Slika 8 prikazuje grafični uporabniški vmesnik orodja Power Query znotraj Power BI Desktop platforme. V osrednjem delu je prikazana razpredelnica podatkovnega seta, ki je trenutno aktiviran. V zgornjem delu se nahaja meni funkcij, ki jih lahko izvajamo. Na desni strani pa se izpisuje zgodovina vseh aktivnosti, ki smo jih izvedli. V spodnjem delu je prikazan graf distribucije posameznih vrednosti trenutno izbranega atributa in opisna statistika njegovih vrednosti. Krivulja učenja uporabe orodja je položna, zato je izvedba večine enostavnih in rutinskih aktivnosti izredno enostavna in hitra. Na spletu je dostopna tudi izvrstna uradna dokumentacija (Microsoft, brez datuma a), kjer so zapisana navodila in prikazani postopki, ki so pomembni za učinkovito delo z orodjem. V primeru potrebe po dodatni pomoči ali morebitnih vprašanj, je na voljo tudi aktiven uradni forum (Microsoft, brez datuma b).

Slika 8: Uporabniški vmesnik Microsoft Power Query



Vir: Lastno delo.

Povezljivost z viri podatkov

Nabor podatkovnih virov, s katerimi Power Query lahko razpolaga, je ogromen. Napram ostalima dvema orodjema ima najenostavnejši sistem branja podatkov iz najrazličnejših virov. Vnašamo lahko datoteke različnih formatov, povezujemo se lahko s številnimi oblačnimi platformami in vzpostavljamo povezave s podatkovnimi bazami. S poganjanjem R in Python skript lahko preberemo in uvozimo tudi podatkovne zapise, ki niso podprti s privzetimi funkcijami. Glavna slabost orodja je, da ne omogoča možnosti izvoza obdelanih podatkov v željene podatkovne formate in podatkovne baze. To seveda ne predstavlja problema, če želimo podatke uporabljati le znotraj Power BI platforme oz. ostalih orodij kompatibilnih s Power Query. V primeru, da obdelane podatke želimo uvoziti v drugo analitično orodje, potrebujemo hiter način za izvoz le-teh. Težavo je sicer moč odpraviti z uporabo Python ali R skript. Zgoraj sem poudaril dobro integracijo z omenjenima jezikoma, zato lahko znotraj orodja poženemo skripto, ki podatke izvozi v željen format. Med preizkusom sem na tovrsten način, s pomočjo Python skripte, obdelane podatke izvozil v csv datoteko.

3.8 Opis izvedbe zgornjega primera s predstavljenimi orodji

Izvedbo sem začel z uporabo orodja Python, kajti bil sem že predhodno seznanjen z njegovo uporabo. Hkrati sem vedel, kakšne so zmožnosti in poznal nekatere pristope, ki so primerni za izvedbo zastavljenega problema. Delal sem znotraj Spyder programskega okolja, kar mi je omogočalo interaktivno delo s podatki. To pomeni, da sem lahko dele kode poganjal posamično in sproti pregledoval izpise in rezultate posamezne funkcije. Večino dela sem

opravil s pomočjo knjižnice Pandas, ki mi je omogočila branje podatkov zapisanih v tekstovnih datotekah, združevanje prebranih podatkovnih setov, odstranjevanje atributov, filtriranje vrednosti atributa, zamenjavo vrednosti in označevanje osamelcev. S pomočjo knjižnice Fuzzywuzzy (Fuzzywuzzy, 2018) pa sem izračunal oceno podobnosti dveh tekstovnih vrednosti atributa avtomobilske znamke. Tiste, ki so presegle določeno mejo podobnosti, sem zamenjal z ustrezno vrednostjo. Nazadnje je bilo potrebno odpraviti še osamelce numeričnih atributov, ki sem jih zaznal s pomočjo metode medkvartilnega intervala. Vizualiziral sem jih z grafom škatle z brki, ki sem ga izrisal s pomočjo knjižnice Seaborn (Seaborn, 2019). S to metodo je bilo možno zaznati nekaj močno odstopajočih vrednosti, za katere je bilo jasno, da so plod napak beleženja, zato sem jih odstranil.

Delo na primeru z orodjem Knime je prav tako potekalo brez večjih zapletov. Orodje je izredno enostavno in intuitivno. Poudaril bi, da nam napram Pythonu omogoča, da se hitro posvetimo jedru problema in ne rabimo izgubljati časa z nepotrebnimi opravili, kot je popravljanje sintakse in branje dokumentacije posameznih knjižnic, ki jih želimo uporabiti. Sem pa zaznal, da orodje počasneje procesira podatke, kot če uporabljamo Python znotraj primernega razvojnega okolja. Uvoz podatkov zapisanih v tekstovnih datotekah, združevanje podatkovnih setov, odstranjevanje atributov, filtriranje vrednosti atributa in vizualizacijo podatkov sem brez težav izvedel z uporabo privzetih funkcij, ki jih Knime ponuja. Obravnavanje osamelcev numeričnih atributov prav tako ni predstavljalo problemov. Knime ponuja privzeto funkcijo, ki označi vse osamelce zaznane po metodi medkvartilnega intervala. Osamelce sem brez težav tudi vizualiziral s pomočjo funkcije za izris škatle z brki. Pri postopku iskanja in odpravljanja tekstovnih napak sem moral poseči po pisanju lastne skripte, s katero sem lahko dosegel željen rezultat. Za ta namen sem izkoristil dobro integracijo s Pythonom in posledično tudi Pandas knjižnico. Na ta način sem uspešno poiskal tekstovne napake pri vrednostih atributa avtomobilskih znamk in jih nato nadomestil s pravilno vrednostjo. Izvajanje skripte je bilo počasnejše (izmerjen čas je bil 285 sekund), kot v primeru, ko sem delal izključno s Pythonom (izmerjen čas je bil 3 sekunde).

Tekom dela z orodjem Microsoft Power Query sem bil manj uspešen, kot z uporabo ostalih dveh orodij. Oba podatkovna seta zapisana v tekstovnih datotekah sem brez problemov uvozil. Poleg tega sem brez težav združil podatkovna seta, odstranil nepotrebne atribute in filtriral vrednosti. Tovrstne aktivnosti se lahko zaradi interaktivnega dela izvede izredno hitro in enostavno, hitreje kot z uporabo ostalih dveh orodij. Več težav sem imel pri odpravi tekstovnih napak pri imenih avtomobilskih znamk. Tako kot pri orodju Knime sem bil primoran izkoristiti integracijo s Pythonom in pognati skripto. Izvajanje skripte je bilo prav tako dokaj počasno (izmerjen čas je bil približno 190s). Obravnavanje osamelcev prav tako ni bilo enostavno. Vizualiziral jih sicer nisem znotraj Power Query, kajti ta ne ponuja vizualizacijskih funkcij, ampak v Power BI, ki služi kot izredno dobro orodje za izdelovanje grafičnih prikazov. Problem je nastal, ko sem hotel znotraj Power Query filtrirati vse vrednosti, ki so po metodi medkvartilnega intervala ovrednoteni kot osamelci. Opazil sem,

da Power Query ne ponuja možnosti za izračun kvantilov izbranega atributa, zato sem tudi v tem primeru moral poseči po pisanju lastne skripte.

3.9 Diskusija

V spodnjem delu sem zapisal nekatere pozitivne (označene z znakom +) in negativne lastnosti (označene z znakom -) preizkušenih orodij, ki sem jih zaznal tekom testiranja. Namen je prikazati prednosti in slabosti, ter s tem zagotoviti lažjo izbiro orodja za izvedbo določene naloge.

Programski jezik Python

- (+) Izvrstne knjižnice za delo s podatki (tj. podatkovno rudarjenje, statistične analize, vizualizacija podatkov, strojno učenje, upravljanje z masovnimi podatki itd.).
- (+) Objektno usmerjen programski jezik z relativno enostavno sintakso.
- (+) Celoten ekosistem je odprtokoden in brezplačen.
- (+) Izjemno učinkovito za opravljanje nalog napredne analitike.
- (+) Ogromna skupnost uporabnikov.
- (+) Možnost interaktivnega programiranja, kar je lahko priročno pri raziskovanju podatkov, pripravi, kot tudi sami analitiki.
- (+) Možnost avtomatizacije ponavljajočih aktivnosti priprave podatkov.
- (-) Nekatere enostavnejše procese priprave podatkov je hitreje in lažje izvesti v samopostrežnem orodju.
- (-) Strma krivulja učenja napram ostalim samopostrežnim orodjem.
- (-) Za optimalno uporabo je potrebno znanje programiranja.
- (-) Odvisnost od številnih odprtokodnih knjižnic, ki jih je potrebno poiskati/naložiti/pregledati/se jih naučiti.

Knime analitična platforma

- (+) Delo z vozlišči omogoča visoko mero fleksibilnosti brez potrebe po pisanju lastne kode.
- (+) Enostavno nadgradljivo s številnimi vtičniki. Možnost pisanja lastnih vozlišč (funkcij).
- (+) Možnost integracije z R in Python programskima jezikoma. Fleksibilnost ni enaka, kajti skripte morajo biti napisane po določeni predlogi.
- (+) Dobra dokumentacija in uporabniški vmesnik.
- (+) Odprtokoden in popolnoma brezplačen.
- (+) Brezplačna različica nima nobenih omejitev.
- (+) Na voljo je tudi plačljiva različica Knime Server, ki med drugimi omogoča tudi poganjanje procesov na zunanjih strežnikih, kolaboracijo z več uporabniki in načrtovanje avtomatskega izvajanja procesov.
- (-) Branje in procesiranje večjih količin podatkov postane počasno.

- (-) Za bolj specifične aktivnosti se je potrebno naslanjati na Python, R ali Java skripte.

Microsoft PowerQuery znotraj Power BI platforme

- (+) Konstanten pregled nad podatki s katerimi operiramo. Pregled imamo tudi nad statističnimi vrednostmi, distribucijo vrednosti in kvaliteto posameznega atributa. Ta funkcionalnost lahko v primeru večji podatkovnih setov postane zaradi počasnosti manj uporabna.
- (+) Hiter med pripravljanjem tudi večjih podatkovnih setov, kajti operiramo le na deležu podatkov. Vse operacije se po končanju aplicirajo na celoten podatkovni set, kar pa lahko vzame več časa, odvisno od količine podatkov.
- (+) Izvajanje manj specifičnih aktivnosti priprave podatkov je zaradi interaktivnosti izredno enostavno.
- (+) Ogromno podprtih podatkovnih virov in enostaven uvoz podatkov.
- (+) Integracija s Python in R programskim jezikom za namene branja in priprave podatkov. Fleksibilnost ni enaka, kajti skripte morajo biti napisane po določeni predlogi.
- (+) Dober in pregleden uporabniški vmesnik.
- (-) Brez enostavne možnosti izvoza obdelanih podatkov.
- (-) Med branjem večjih podatkovnih setov in končnem apliciranju vseh uporabljenih funkcij je lahko precej počasen.
- (-) Ni samostojno orodje, kajti deluje le znotraj Microsoft programskega okolja.
- (-) Poganjanje skript je na večjih podatkih počasno
- (-) Omejena funkcionalnost. Primerno predvsem za izvajanje standardnih procesov priprave (odstranjevanje atributov, obravnavanje manjkajočih vrednosti, agregacija podatkov, integracija podatkov iz različnih virov, filtriranje, izdelava oz. izpeljevanje novih atributov ipd.).
- (-) Za bolj specifične aktivnosti se je potrebno naslanjati na Python ali R skripte.

Uporaba Pythona v kombinaciji z ustreznim razvojnim okoljem in primernimi knjižnicami je po mojem mnenju najprimernejša v primerih, ko imamo poleg standardnih procesov priprave podatkov opravka tudi z zahtevnejšimi procesi. V zgoraj opisanem primeru je tak zahtevnejši proces predstavljal proces zaznavanja in odpravljanja tekstovnih napak vrednosti atributa znamke vozila. Takšne in podobne primere lahko elegantno rešimo s pisanjem lastne kode, zato je uporaba Pythona v tovrstnih primerih povsem primerna. Poleg tega bi uporabo Pythona priporočil tudi za potrebe raziskovanja podatkov in pripravo v procesu napredne analitike (npr. podatkovno rudarjenje). Namreč, kot je bilo že poudarjeno, je Python izredno priljubljeno na tem področju, zato je v takih primerih smiselno celotno pripravo podatkov izvesti z njim. Tako ne rabimo prehajati med različnimi orodji po nepotrebnem.

Knime lahko služi kot povsem samostojno analitično orodje, ki omogoča izvajanje analitike in priprave podatkov brez potrebe po pisanju lastne kode. Priprava podatkov po načinu povezovanja vozlišč je enostavnejša kot v primeru pisanja lastne kode in hkrati dovolj fleksibilna za številna standardna opravila. Priporočil bi ga predvsem uporabnikom, ki želijo doseči visoko zmogljivost in fleksibilnost brez pisanja programske kode. Orodje prav tako omogoča enostaven izvoz podatkov, tako da prehajanje med ostalimi analitičnimi orodji ne predstavlja nikakršnih težav.

Power Query znotraj Power BI platforme predstavlja najenostavnejše orodje med testiranimi za opravljanje standardnih procesov priprave podatkov. Ni najprimernejše za raziskovanje podatkov, obdelavo osamelcev in poganjanjem skript na večjih podatkovnih setih. Hkrati je močno omejeno pri izvozu podatkov za uporabo v drugih orodjih. Ob testiranju sem ugotovil, da je sicer možno izvoziti podatke, vendar je postopek izredno nepraktičen in nepriporočljiv. Prav to je tudi glavni razlog, zakaj ga ne bi priporočal za samostojno uporabo, temveč le v primerih, ko želimo podatke kasneje uporabljati znotraj orodij, ki podpirajo integracijo s Power Query (npr. Microsoft Power BI in Microsoft Excel). V tovrstnih primerih se izkaže za izredno hitro in učinkovito.

SKLEP

Področje poslovne inteligence in podatkovne znanosti se v današnjih časih izredno hitro spreminja in nadgrajuje. Z razvojem zmogljivejše strojne opreme, večanja količine zbranih podatkov in števila različnih podatkovnih kanalov ipd., se konstantno razvija tudi programska oprema za delo s podatki. Nimamo konkretnega zagotovila, da bodo vsa tri uporabljena orodja še vedno tako priljubljena in uporabljena tudi v naslednjih letih. V nalogi sem tako, tudi zaradi tega, poskušal izbrati taka orodja, ki so že dodobra ustaljena na tem področju, imajo dovolj veliko skupnost uporabnikov in se konstantno posodabljajo.

Celoten postopek primerjave orodij je v večji meri potekal po načrtih, ki sem si jih zastavil. Ponovno bi poudaril dejstvo, da je sam proces priprave podatkov težko posplošiti in natančno razdeliti na posamezne aktivnosti. Samih aktivnosti znotraj procesa priprave je veliko in so močno odvisne od pridobljenih podatkov, ki jih imamo na voljo, in potreb analitike. Prav to je bila ena izmed večjih omejitev te primerjave, namreč nemogoče je prikazati uporabo posameznega orodja na vseh vrstah različnih primerov, s katerimi se lahko uporabnik sreča. Vsa uporabljena orodja sem tako poskušal na čim boljši možen način predstaviti in jih primerno ovrednotiti, ne glede na to, s kakšnim primerom priprave podatkov imamo opravka. Prav zagotovo mi ni uspelo izpostaviti vsake lastnosti orodja in jih popolno in povsem objektivno ovrednotiti. Uspelo pa mi je zapisati nekaj ključnih opazk, ki sem jih zaznal tekom uporabe in dela z njimi. Hkrati bi poudaril tudi, da se lahko posameznikova izkušnja nekoliko razlikuje od moje. To lahko pripišemo predvsem drugačnimi pričakovanji posameznika, njegovo predhodno seznanjenostjo s podobnimi orodji in hkrati tudi s seznanjenostjo s samim procesom priprave podatkov. Vseeno menim,

da lahko končne ugotovitve služijo kot dobra pomoč vsakemu, ki želi izbrati ustrezno orodje za opravljanje najrazličnejših nalog priprave podatkov v procesu podatkovne analitike.

LITERATURA IN VIRI

1. Chen, Z. (2001). *Data mining and uncertain reasoning*. New York: John Wiley & Sons, Inc.
2. Eckerson, W. W. (2007). Predictive Analytics: Extending the Value of Your Data Warehousing Investment. *TDWI Best Practices Report*, 1, 1-36.
3. Han, J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.
4. Hellerstein, J. M., Heer, J. & Kandel, S. (2018). Self-Service Data Preparation: Reasearch to Practice. *IEEE Data Engineering Bulletin*, *41*(2), 23-34.
5. Janert, K. P. (2011*). Data Analysis with Open Source Tools*. Sebastopol: O'Reilly Media, Inc.
6. Kimball, R. & Caserta, J. (2004*). The Data Warehouse ETL Toolkit*. Indianapolis: Wiley Publishing, Inc.
7. Knime Analytics Platform [programska oprema]. (2019). Pridobljeno 15. novembra iz https://www.knime.com/
8. Knime AG. (brez datuma a). *Knime forum*. Pridobljeno 8. aprila iz https://forum.knime.com/
9. Knime AG. (brez datuma b). *Knime hub*. Pridobljeno 8. aprila iz https://hub.knime.com/
10. Kotsiantis S. B., Kanellopoulos, D. & Pintelas P. E. (2006). Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, *1*(2), 111-117.
11. Kwak, K. S. & Kim, H. J. (2017). Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, *70*(4), 407-411.
12. Marín-Ortega, P. M., Dmitriyev, V., Abilov, M. & Gómez, J. M. (2014). ELTA: New Approach in Designing Business Intelligence Solutions in Era of Big Data. *Procedia Technology*, *16*, 667-674. Pridobljeno 3. februarja 2020 iz https://www.sciencedirect.com/science/article/pii/S2212017314002424
13. Mawer, C. (2017, 9 marec). *The Value of Exploratory Data Analysis* [objava na blogu]. Pridobljeno iz https://www.svds.com/value-exploratory-data-analysis/
14. Microsoft Power BI Desktop [programska oprema]. (2020). Pridobljeno 18. marca na spletnem mestu https://powerbi.microsoft.com/en-us/
15. Microsoft. (brez datuma a). *Microsoft Power Query Documentation*. Pridobljeno 8. aprila iz https://docs.microsoft.com/en-us/power-query/
16. Microsoft. (brez datuma b). *Microsoft Power BI Community*. Pridobljeno 8. aprila iz https://community.powerbi.com/
17. Ministrstvo za infrastrukturo. (brez datuma). *Rezultati tehničnih pregledov motornih vozil*. Pridobljeno 22. februarja 2020 iz https://podatki.gov.si/dataset/rezultati-tehnicnih-pregledov-motornih-vozil

18. Negash, S. & Gray, P. (2008). Business Intelligence. V *Handbook on decision support systems 2* (str. 175-193). Berlin, Heidelberg: Springer.
19. NumPy [programska oprema]. (2019). Pridobljeno 8. januarja iz https://numpy.org/
20. Pandas [programska oprema]. (2019). Pridobljeno 8. januarja iz https://pandas.pydata.org/
21. Piatetsky, G. (2019). *Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis* [objava na blogu]. Pridobljeno 14. januarja 2019 iz https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html
22. Provost, F. & Fawcett, T. (2013). *Data Science for Business*. Sebastopol: O'Reilly Media, Inc.
23. Python [programska oprema]. (2019). Pridobljeno 8. januarja iz https://www.python.org/
24. Seaborn [programska oprema]. (2019). Pridobljeno 8. januarja iz https://seaborn.pydata.org/
25. Fuzzywuzzy [programska oprema]. (2018). Pridobljeno 22. februarja iz https://github.com/seatgeek/fuzzywuzzy
26. Sivakumar, A. & Gunasundari, R. (2017). A Survey on Data Preprocessing Techniques for Bioinformatics and Web Usage Mining. *International Journal of Pure and Applied Mathematics*, *117* (20), 785-794.
27. Spyder [programska oprema]. (2019). Pridobljeno 8. januarja iz https://www.spyder-ide.org/
28. SQLAlchemy [programska oprema]. (2019). Pridobljeno 8. januarja iz https://www.sqlalchemy.org/
29. Stodder, D. (2016). Improving Data Preparation for Business Analytics. *TDWI Best Practices Report*. Pridobljeno 7. februarja 2020 iz https://www.redpointglobal.com/wp-content/uploads/2016/10/TDWI_BPReport_Q316_RedPoint_F_rev2_code_Final.pdf
30. Widmann, M., Heine M. & Silipo, R. (2018). *Four Techniques for Outlier Detection* [objava na blogu]. Pridobljeno 3. decembra 2019 iz https://www.kdnuggets.com/2018/12/four-techniques-outlier-detection.html
31. Wirth, R. & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 29-39. London, UK: Springer-Verlag.