Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Igor Murgić

Analiza podatkov pacientov z Alzheimerjevo boleznijo z metodami strojnega učenja

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Matjaž Kukar

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Z izbranimi metodami strojnega učenja analizirajte podatke bolnikov z Alzheimerjevo boleznijo iz več različnih podatkovnih zbirk. Raziščite morebitne povezave med atributi in jih ustrezno vizualizirajte. Podatke analizirajte s pomočjo nenadzorovanih in nadzorovanih metod strojnega učenja. Atribute na podlagi predznanja združite v gruče in z njihovo ustrezno razvrstitvijo skušajte minimizirati število opravljenih preiskav, potrebno za zanesljivo diagnozo. Skušajte tudi ovrednotiti praktično uporabnost dobljenih rezultatov.

Za strokovno vodenje in pomoč pri izdelavi diplomskega dela se zahvaljujem mentorju izr. prof. dr. Matjažu Kukarju. Zahvaljujem se tudi Brankici Bratić, prof. dr. Mirjani Ivanović in prof. dr. Vojislavi Bugarski za pomoč pri raziskovanju.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
ADNIDS	ADNI dataset	ADNI - podatkovna zbirka bolnikov z Alzheimerjevo boleznijo
NMDS	Nature medicine dataset	Nature medicine - podatkovna zbirka vzorcev krvne plazme
ADDS	Alzheimer Disease dataset	Podatkovna zbirka bolnikov klinike za nevrologijo in psihiatrijo, Novi Sad

Povzetek

Naslov: Analiza podatkov pacientov z Alzheimerjevo boleznijo z metodami strojnega učenja

Avtor: Igor Murgić

Cilj diplomske naloge je analiza podatkov bolnikov z Alzheimerjevo boleznijo in uporaba napovednih modelov, zgrajenih z metodami strojnega učenja. Zbrane podatke smo analizirali in poiskali zakonitosti med atributi. Atribute podatkov smo predstavili v obliki neusmerjenega grafa. Z uporabo zgrajenih modelov smo med atributi poiskali najpomembnejše in zavrgli tiste, ki so povzročali prekomerno prileganje. Tako dobljene modele smo testirali s pomočjo prečnega preverjanja in dobili rezultate točnosti modela. Zgrajeni modeli in primerjave med njimi so pokazale izstopanja nekaterih atributov, ki bi nam z manj preiskavami omogočili enostavnejšo in hitrejšo postavitev diagnoze same bolezni. Izločanje preiskav za zdravnike ni smiselno, saj jim povedo marsikaj o stanju bolnika. Lahko pa bi prilagodili vrstni red preiskav in s tem hitreje postavili diagnozo.

Ključne besede: strojno učenje, Alzheimerjeva bolezen, analiza podatkov, gručenje, klasifikacija, prečno preverjanje, odločitvena drevesa, neusmerjeni grafi.

Abstract

Title: Analyzing Alzheimer's patients data with machine learning methods

Author: Igor Murgić

The aim of the diploma thesis is to analyze the data concerning patients with Alzheimer's disease and to use the predictive models constructed through machine learning methods. The collected data was analyzed and the laws between attributes were defined. The data attributes were presented in the form of an undirected graph. The most relevant attributes were determined using the constructed models, the attributes that caused overfitting were eliminated. The models thus obtained were tested through cross-validation and the accuracy of each model was calculated. The constructed models and the comparisons between them showed that certain attributes were more distinctive than others. These attributes would enable us to simplify and expedite the establishment of the diagnosis of the disease, conducting fewer tests. Doctors deem the elimination of certain tests unreasonable, though, since a lot of information on the patient's condition can be deduced from them. We could, however, modify the sequence of the tests, which would lead to more rapid establishment of the diagnosis.

Keywords: machine learning, Alzheimer's disease, data analysis, clustering, classfication, cross-validation, decision trees, undirected graphs.

Poglavje 1

Uvod

Alzheimerjeva bolezen je nevrodegenerativna bolezen, katere znaka sta demenca in izguba kognitivnih funkcij [27]. Poimenovana je po nemškem psihiatru in patologu Aloisu Alzheimerju [20]. Obstaja več oblik Alzheimerjeve bolezni, ki jih ločimo glede na napredek bolezni [23]. Bolezen se najpogosteje razvije po 65. letu starosti in predstavlja 70 % vseh oblik demence. Leta 2016 je z Alzheimerjevo boleznijo živelo 47 milijonov ljudi po vsem svetu [4]. Poročilo organizacije Alzheimer's Disease International (ADI) navaja, da bo do leta 2050 ta številka narasla na več kot 130 milijonov [4]. Stroške zdravljenja bolezni po vsem svetu ocenjujejo na več kot 800 milijard ameriških dolarjev [4]. V Sloveniji je v letu 2012 z Alzheimerjevo boleznijo živelo približno 32000 ljudi, kar predstavlja več kot 1,5 % prebivalstva države [1].

Za odkrivanje bolezni se uporablja slikanje možganov z magnentno resonanco in študije bioloških kazalcev [12]. Vzroke bolezni pripisujejo genetskim faktorjem in boleznim srca in ožilja. Zdravila za Alzheimerjevo bolezen trenutno ni, obstajajo pa zdravila za lajšanje bolezni, ki zavirajo propad živčnih celic in s tem preprečujejo napredovanje bolezni za 6 do 12 mesecev [28]. Za zaviranje delovanja encima acetilholinesteraze, ki razgrajuje acetilholin, ki ga živčne celice uporabljajo za prenašanje signalov, se uporabljajo zdravila donepezil, rivastigmine in galantamine [26]. Alzheimerjeva bolezen poškoduje možganske celice, ki posledično prekomerno izločajo aminokislino, imeno-

vano glutamat [11]. Zdravilo memantin zavira izločanje glutamata in tako preprečuje nadaljnje poškodbe možganskih celic [11].

Namen diplomske naloge je zbrati in analizirati zbrane podatke o Alzheimerjevi bolezni iz več virov, poiskati zakonitosti v podatkih in na podlagi ugotovitev zgraditi model, ki bi nam z določeno verjetnostjo povedal ali gre za primer bolnika z Alzheimerjevo boleznijo. Z zmanjšanjem števila testov potrebnih za diagnozo bolezni ali prilagajanjem njihovega vrstnega reda, bi omogočili hitrejše diagnosticiranje bolnikov. Zgodnja diagnoza bolezni bi omogočila bolnikom uspešnejše zdravljenje in preprečevala napredek bolezni.

Priprava podatkov vključuje obravnavo manjkajočih podatkov in transformacijo nekaterih atributov. Z izračunom korelacij smo poiskali povezave med atributi. Za gradnjo in preverjanje modelov smo uporabili algoritme nadzorovanega in nenadzorovanega strojnega učenja. Uporabili smo metode redukcije dimenzij podatkov za zmanjšanje števila atributov in poiskali najpomembnejše atribute. Rezultate posameznih metod smo predstavili v obliki grafov, slik in tabel.

Diplomsko delo je strukturirano na naslednji način:

- kratek uvod v tematiko naloge
- opis podatkov (poglavje 2)
- opis metod uporabljenih pri raziskovanju (poglavje 3)
- podroben opis rezultatov (poglavje 4)
- zaključek z glavnimi ugotovitvami (poglavje 5).

Vsaka od uporabljenih metod vsebuje teoretični opis in način uporabe pri raziskovanju.

Poglavje 2

Opis podatkov

Pridobljeni podatki za analizo izvirajo iz treh virov. Prva zbirka podatkov (ADNIDS) je del podatkovne baze bolnikov Alzheimer's Disease Neuroimaging Initiative cohort (ADNI), natančneje ADNIMERGE tabele, ki je del ADNIMERGE R podatkovne zbirke [12].

Drugi vir podatkov (NMDS) je pridobljen iz analize arhiviranih vzorcev krvne plazme, raziskava katerih je opisana v članku z naslovom *Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins* [27].

Zadnja podatkovna zbirka (ADDS) vsebuje splošne podatke bolnikov in rezultate kognitivnih testov Klinike za nevrologijo in psihiatrijo Novi Sad, ki so opisani v naslednjih odstavkih [18]. Med atributi podatkovne zbirke smo na osnovi sklopov preiskav določili 13 skupin.

Tabela 2.1 prikazuje število stolpcev in primerov posamezne podatkovne zbirke. Atributi, ki niso vsebovali vrednosti ali le te niso ustrezale, so bili pri raziskovanju izpuščeni.

	ADNIDS	**NMDS**	**ADDS**
Število vrstic (primerov)	919	176	85
Število stolpcev (atributov)	50	121	146

Tabela 2.1: Velikosti podatkovnih zbirk

2.1 ADNIDS

ADNIDS vsebuje 919 primerov in 50 atributov v treh večjih skupinah [12].

Prva skupina vsebuje osebne podatke pacienta, podatke o kraju in datumu meritev in diagnozo bolezni. Diagnoza same bolezni je opisana s petimi kategorijami:

- kognitivno normalen (CN) (188 primerov)
- skrb vzbujajoča sprememba spomina (SMC) (106 primerov)
- zgodnje lažje kognitivno poslabšanje (EMCI) (310 primerov)
- pozno lažje kognitivno poslabšanje (LMCI) (165 primerov)
- Alzheimerjeva bolezen (AD) (150 primerov).

Podatki vsebujejo 481 primerov moških in 438 žensk, katerih povprečna starost je približno 73 let.

Druga skupina vsebuje podatke bioloških meritev ali potencialnih bioloških označevalcev, ki so genetske variacije gena APOE4, rezultati pozitronsko emisijske tomografije (PET) in podatki magnetne resonance [12, 13]. APOE4 je protein cerebrospinalne tekočine in predstavlja do sedaj največjo znano potencialno genetsko možnost obolenja za Alzheimerjevo boleznijo pri raznih etničnih skupinah[12, 7].

Tretja skupina vsebuje z vprašalniki in testi pacientov pridobljene klinične meritve in teste, ki opisujejo kognitivno in psiho-nevrološko stanje bolnikov [12, 13].

Testi in meritve so:

- lestvica ocene Alzheimerjeve bolezni (ADAS13)

  ADAS13 se uporablja pri kliničnem preizkušanju zdravil za Alzheimerjevo bolezen [9].

- Mini Mental State Examination (MMSE)

  MMSE vsebuje 11 vprašanj oziroma meritev, ki sistematsko ocenjujejo mentalno stanje bolnika [19, 13]. Ta vprašanja pokrivajo področja kognitivnih funkcij, ki so orientacija, prepoznavnost, pozornost,

preračunljivost, spomin in jezik [19]. Vprašalnik je kratek, zato je zelo praktičen za ocenjevanje bolezni.

- Rey Auditory Verbal Learning Test (RAVLT)

  Test temelji na ocenjevanju verbalnih sposobnosti in spomina. Pacient mora ponavljati slišane besede v intervalih. Med naštevanjem pacienta prekinejo in po časovnem zamiku ponovno povprašajo po besedah prejšnjega intervala [5].

- Functional Assessment Questionnaire (FAQ)

  Vprašalnik vsebuje 10 kategorij vsakdanjih opravil in aktivnosti, katere se ovrednoti z od 0 do 3 točkami glede na zmožnost samostojnega opravljanja aktivnosti [17].

- Montreal Cognitive Assessment (MoCA)

  MOCA je 10 minutni vprašalnik s 30 točkami, pri katerem rezultat nad 26 točk nakazuje normalne kognitivne sposobnosti pacienta [22, 29]. Ocenjuje in poudarja naloge prednjega dela možganov [22, 29].

- Everyday Cognition (ECog)

  ECog je vprašalnik o kognitivnih funkcijah (spomin, jezik, vizualno-prostorske sposobnosti, planiranje, organiziranost), izpolnjen s strani pacienta in sodelujočega partnerja [13].

ADNIDS je vseboval malo manj kot 4% manjkajočih vrednosti, natančneje 1802. Atribut „PIB_bl“, pri katerem manjkajo vrednosti za vse primere, smo izločili. Dodatno so bili izločeni tudi atributi, ki so vsebovali časovne vrednosti tj. datume, atribut za identifikacijo primerov zbirke, identifikator lokacije opravljanja testov in atributa, ki označujeta protokol zbiranja podatkov. Za analizo smo uporabili 38 atributov.

2.2 NMDS

NMDS zbirka podatkov vsebuje 176 primerov in 121 atributov [27]. Diagnoza bolezni je v dani zbirki podatkov opisana z dvema kategorijama:

- Alzheimerjeva bolezen (AD) (64 primerov)
- nedementno stanje (NDC) (112 primerov)

Izbor atributov v NMDS temelji na predpostavki, da proces ki vodi do Alzheimerjeve bolezni, povzroča karakteristične spremembe proteinov v krvi, ki povzročajo molekularne spremembe, specifične za bolezen [27]. Za razliko od prve podatkovne zbirke, ta ni vsebovala manjkajočih vrednosti.

2.3 ADDS

Podatkovna zbirka vsebuje podatke kliničnih meritev in kognitivnih testov bolnikov iz Republike Srbije [18]. ADDS vsebuje 85 primerov zbranih podatkov s 145 atributi, izmed katerih smo na podlagi samega imena atributa in začetnih korakov analize podatkovne zbirke, ki so pokazali, da atributa nakazujeta na razred primera, za ustreznost analize odstranili atributa „BrojMeseciSaBolesti“ in „BrojMeseciSaPoremecajemUPamcenju“. Atribute smo uvrstili v 13 skupin, ki pripadajo naslednjim skupinam testov:

- ZnaciFokalnogOstecenja (14 atributov)
- Brojevi (4 atributi)
- QoLIspitanik (2 atributa)
- WCST (9 atributov)
- VerbalnaFluentnost (3 atributi)
- TMT (4 atributi)
- BNT (3 atributi)
- NPI (11 atributov)

- MMSE (12 atributov)
- ACEIII (6 atributov)
- RAVLT (11 atributov)
- ROCF (3 atributi)
- ostalo (61 atributov).

12 skupin so določala imena atributov, ki so vsebovala enako predpono. V zadnjo skupino smo uvrstili preostale atribute, ki niso bili razvrščeni. Diagnoza bolezni je v ADDS opisana s tremi razredi:

- Alzheimerjeva bolezen (A) (29 primerov)
- kontrolna skupina (C) (29 primerov)
- kognitivno poslabšanje (MC) (27 primerov).

ADDS je vseboval malo manj kot 5% (615) manjkajočih vrednosti, označenih z znakom „?“, ki so bile nadomeščene z mediano atributa pri določeni manjkajoči vrednosti. Med primeri je bilo 25 moških in 60 žensk katerih povprečna starost je bila nad 71 let. Skupini MMSE in RAVLT sta vsebovali podatke testov, ki jih lahko najdemo tudi v ADNIDS. Pri skupini MMSE je podobnost z ADNIDS pri skupni oceni testa, medtem ko za skupino RAVLT nismo uspeli določiti ujemanja atributov.

Poglavje 3

Metode

Za analizo podatkov in gradnjo modelov s strojnim učenjem smo uporabili odprtokodni programski jezik Python, ki nam s pomočjo knjižnic, namenjenih analizi podatkov, strojnemu učenju in vizualizaciji podatkov, omogoča enostavno kodiranje in delo.

Scikit-learn je prostodostopna knjižnica, namenjena strojnemu učenju v Python-u [25]. Vsebuje številne algoritme za nadzorovano in nenadzorovano strojno učenje. Scikit-learn uporablja visokonivojske ukaze in s tem omogoča, da se osredotočimo na vsebino problema in ne zgolj na programiranje [25].

Za delo v okolju Windows smo uporabili paket ANACONDA (v.4.3.1), ki že vsebuje Python (v.3.5.2) in številne knjižnice, potrebne za tovrstne analize. Nekatere izmed knjižnic, uporabljenih predvsem za vizualizacijo podatkov, je bilo potrebno nastaviti dodatno. Pri raziskovanju smo uporabili naslednje Python knjižnice:

- SciPy (izračun korelacij) [10]
- Matplotlib (vizualizacije podatkov) [16]
- Scikit-learn (algoritmi strojnega učenja) [25]
- Pandas (delo s podatki)
- Neo4j Python Driver (povezava z Neo4j bazo)
- Networkx (vizualizacija grafov)

- Seaborn (vizualizacije podatkov).

Za delo s Python-om smo v povezavi z Bitbucket storitvijo za verzioniranje uporabili razvojno okolje PyCharm.

Med vsemi pari atributov posamezne podatkovne zbirke smo izračunali korelacije, jih uvozili v Neo4j grafno bazo in predstavili v obliki polnega grafa, za vsako izmed podatkovnih zbirk posebej. Uporabili smo metodo hierarhičnega gručenja in na podlagi podobnosti primerov ocenili pripadnost primera ciljnemu razredu. Za nadzorovano strojno učenje smo izbrali ustrezen klasifikator in preverili točnost dobljenega modela z uporabo prečnega preverjanja. Število atributov smo zmanjšali na podlagi pomembnosti atributov in uporabe algoritma za reduciranje problemskega prostora.

Za predstavitev atributov in korelacij med njimi smo uporabili Neo4j grafno bazo, ki nam preko spletnega vmesnika omogoča pregled in nadzor nad podatkovno bazo.

3.1 Imputacija manjkajočih vrednosti in izračun korelacij

Z imputacijo manjkajočih vrednosti podatkov se izognemo odstranjevanju atributov ali primerov, katerih vrednosti niso podane. Nekateri algoritmi strojnega učenja zahtevajo popolnost podatkov. Da bi določili povezave atributov, smo med njimi izračunali korelacije. Korelacije atributov nam poleg moči povezanosti atributov razkrijejo tudi razmerja med njimi.

3.1.1 Imputacija

Začetni pregled podatkov je zahteval imputacijo manjkajočih vrednosti v ADNIDS in ADDS, ki so bile med podatki zapisane z znakom „?“. Kjer so se pojavile manjkajoče vrednosti atributa, smo jih nadomestili z mediano. Pri branju vhodne datoteke s knjižnico Pandas smo kot parameter za manjkajoče vrednosti podali znak „?“. Tako dobljen DataFrame je vseboval

vrednosti NaN, ki smo jih z metodo *fillna()* enostavno nadomestili z uporabo želene metode tj. izračun mediane [2]. Da bi v analizo lahko vključili tudi atribute z diskretnimi vrednostmi, smo jih transformirali. Pri tem smo uporabili metodo *get_dummies()*, ki v podanem DataFrame objektu transformira diskretne atribute tako, da generira dodatne indeksne atribute [2]. Število indeksnih atributov je odvisno od števila razredov diskretnega atributa.

Tabela 3.1 prikazuje primer transformacije atributa PTMARRY podatkovne zbirke ADNIDS. Generiranim atributom se dodelijo binarne vrednosti glede na vrednost originalnega atributa.

PTMARRY	**PTMARRY_Divorced**	**PTMARRY_Married**	**PTMARRY_Unknown**	**PTMARRY_Widowed**	**PTMARRY_Never married**
Divorced	1	0	0	0	0
Married	0	1	0	0	0
Unknown	0	0	1	0	0
Widowed	0	0	0	1	0
Never married	0	0	0	0	1

Tabela 3.1: Transformacija diskretnega atributa PTMARRY podatkovne zbirke ADNIDS

Atribut PTGENDER podatkovne zbirke ADNIDS, ki vsebuje podatek o spolu bolnika, smo transformirali z zamenjavo diskretnih vrednosti v številske. Tabela 3.2 prikazuje transformacijo atributa PTGENDER, kjer so bile vrednosti atributa zamenjane brez generiranja dodatnih atributov.

PTGENDER	PTGENDER
Female	0
Male	1

Tabela 3.2: Transformacija diskretnega atributa PTGENDER podatkovne zbirke ADNIDS

3.1.2 Izračun korelacij

Korelacija oziroma koeficient korelacije v statistiki predstavlja ali pozitivno ali negativno povezanost dveh spremenljivk. Za statistično analizo izbranih podatkovnih zbirk smo izračunali Pearsonov in Spearmanov korelacijski koeficient med atributi. Spremenljivke v izračunu koeficientov predstavljajo atributi.

Pearsonov koeficient

Pearsonov koeficient korelacije nam pove, ali med spremenljivkama obstaja linearna povezanost in kako močna je le ta [6]. Koeficient lahko zavzame vrednost na intervalu od -1 do 1, kjer 1 predstavlja popolno (funkcijsko) pozitivno linearno povezanost, -1 negativno linearno povezanost in 0 da med spremenljivkama ne obstaja nikakršna linearna povezanost. Pozitivna povezanost pomeni naraščanje druge spremenljivke ob naraščanju prve in obratno, negativna povezanost pomeni zmanjševanje druge spremenljivke ob naraščanju prve. Enačba 3.1 prikazuje izračun Pearsonovega koeficienta korelacije, ki ga izračunamo kot količnik med kovarianco spremenljivk in produktov standardnih odklonov posamezne spremenljivke.

$$\rho_{x,y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

Spearmanov koeficient

Medtem, ko Pearsonov koeficient korelacije predpostavlja linearno povezanost in normalno porazdelitev spremenljivk, Spearmanov koeficient korelacije ne vsebuje predpostavk o porazdelitvi spremenljivk [6]. Povezanost ali razmerje dveh spremenljivk opisuje z monotono funkcijo in je neparametrična metoda. Za izračun uporabimo range spremenljivk oziroma kvadrat razlike med rangoma dveh spremenljivk.

Za izračun Pearsonovega in Spearmanovega koeficienta smo uporabili funkcijo *pearsonr()* in *spearmanr()* knjižnice Scipy, katerima kot parametra podamo spremenljivki oziroma vektorja za izračun [10]. Funkciji smo uporabili pri računanju korelacij dveh posameznih atributov ob uvozu podatkov v Neo4j grafno bazo. Izračun korelacijskih koeficientov nam omogoča tudi funkcija *corr()* knjižnice Pandas, kateri kot parameter podamo želeno metodo za izračun. Funkcija izračuna korelacijske koeficiente nad celotnim DataFrame objektom za vse pare atributov, zato je bolj primerna pri računanju korelacij med vsemi pari atributov podatkovne zbirke.

Za podatkovne zbirke, ki so vsebovale veliko število atributov (ADDS in NMDS), smo prikaz korelacij na toplotnem grafu zaradi preglednosti prikaza omejili z določeno vrednostjo moči korelacije. Moč korelacije smo primerjali glede na absolutno vrednost.

3.1.3 Uvoz v Neo4j bazo grafov

Neo4j je grafom namenjena nerelacijska podatkovna baza, ki poudarja povezave med vozlišči. Baza uporablja specifičen jezik za poizvedbe imenovan Cypher. Za povezavo z bazo smo uporabili knjižnico Neo4j Python Driver. Do Neo4j strežnika dostopamo preko lokalnega internetnega naslova, kamor podamo uporabniško ime in geslo. Atributi so v grafu predstavljeni kot vozlišča, korelacije med atributi pa kot neusmerjene povezave. Ob vsaki izračunani korelaciji smo v bazo vnesli vozlišči z imenoma atributov, za katera smo izračunali korelacijo med njima. Kreirani povezavi smo kot atribut

podali Pearsonov in Spearmanov koeficient korelacije. Vozliščem smo za atribut dodali ime podatkovne zbirke, iz katere izhaja. V bazo smo vnesli 301 vozlišče in 17996 povezav med njimi. Vsaka izmed podatkovnih zbirk je tako predstavljena v obliki polnega grafa.

3.2 Vizualizacija atributov v obliki grafa

Spletni vmesnik Neo4j baze za prikaz grafa uporablja *force-directed* algoritem knjižnice D3.js [24], zato ni primeren za prikaz zaporedja grafov ob spreminjajoči se meji Spearmanove korelacije. Algoritem poskuša pozicionirati vozlišča tako, da so povezave med vozlišči kar se da kratke in se ne prekrivajo med seboj. Posledično ob dodajanju vozlišč in povezav med njimi ta spreminjajo pozicijo.

Neo4j podpira samo usmerjene povezave med vozlišči, zato smo za prikaz zaporedja grafov uporabili knjižnico NetworkX 1.11 [15]. Knjižnica je prostodostopna in nam omogoča preučevanje grafov in omrežij.

Podatke za izris grafa z uporabo NetworkX knjižnice smo uvozili iz Neo4j baze. Tako dobljen graf nam omogoča veliko več svobode pri vizualizaciji, med drugim tudi nastavljanje pozicij vozlišč, zato je tak način glede na spreminjajočo se stopnjo Spearmanove korelacije veliko bolj primeren za prikaz grafa. Barve vozlišč smo določili glede na podatkovni vir. Modra vozlišča predstavljajo atribute ADNIDS, rdeča vozlišča atribute NMDS in zelena vozlišča atribute ADDS. Povezavam smo dodali atribut s stopnjo Spearmanove korelacije in jih izrisali kot oznake na povezavi.

3.3 Hierarhično gručenje

Hierarhično gručenje smo uporabili z namenom, da bi na podlagi podobnosti primerov kreirali gruče in glede na večinski razred in distribucijo primerov gruče ocenili uspešnost gručenja. Dobljeni rezultati bi pokazali, kako dobro lahko ločimo primere po podobnosti glede na vrednost ciljnega razreda.

Iskanje gruč je postopek, ki nam omogoči vpogled v hierarhijo podatkov, ki jo zgradimo z združevanjem gruč. Pri raziskovanju smo uporabili princip od spodaj navzgor, kjer na začetku vsak primer predstavlja gručo, ki jo zatem združujemo v večje gruče po določenem pravilu oziroma kriteriju. Za združevanje gruč smo uporabili Ward-ovo metodo oziroma kriterij minimalne variance. Pri vsakem koraku združevanja iščemo gruči, katerih združevanje najmanj vpliva na varianco znotraj gruče [21]. Hierarhijo dobljenih gruč smo predstavili z dendrogramom. Za izris dendrograma smo uporabili funkcijo *dendrogram()* knjižnice SciPy [10].

3.3.1 Prečno preverjanje gručenja

Preverjanje točnosti dobljenega modela hierarhičnega gručenja smo realizirali z deljenjem primerov na učno in testno množico, kjer smo za klasifikator določili distribucijo primerov po razredih. Uporabili smo metodo *AgglomerativeClustering()* knjižnice Scikit-learn za pridobivanje gruč [25]. Primere smo razdelili v učno in testno množico v razmerju $9:1$, kar pomeni 10-kratno gradnjo modela in testiranje oziroma 10-kratno prečno preverjanje. S primeri učne množice smo zgradili model in za vsako gručo izračunali centroid. Vsaki izmed gruč smo glede na distribucijo primerov gruče določili večinski razred. Zatem smo za vsakega izmed testnih primerov pridobili razdalje do centroidov gruč učne množice in tako določili, v katero spada.

Ob tem smo prešteli primere, ki so bili pravilno napovedani, in tako pridobili točnost modela. Končni rezultat smo izračunali kot srednjo vrednost posameznega prehoda deljenja na učne in testne primere.

3.4 Izbira klasifikatorja nadzorovanega učenja

Pri izbiri klasifikatorja smo se odločali med:

- klasifikatorjem metode podpornih vektorjev (SVM - RBF jedro)

Metoda z iskanjem prostora med vektorji primerov išče hiperravnino in tako klasificira primere.

- klasifikatorjem odločitvenega drevesa

  Klasifikacijski model gradi rekurzivno z deljenjem učnih podatkov glede na vrednosti atributa izbranega s statističnim testom.

- klasifikatorjem naključnega gozda

  Zgradi podano število klasifikatorjev odločitvenih dreves in uporablja povprečja za povečanje točnosti modela.

- naivni Bayesov klasifikator

  Uporablja algoritem naivnega Bayesa za klasifikacijo primerov, ki na podlagi verjetnosti pripadanja primera razredu klasificira primere.

- klasifikator k-najbližjih sosedov

  Klasifikator z uporabo metode najbližjih sosedov klasificira primere na podlagi razdalje do ostalih primerov.

Zaradi enostavne interpretacije zgrajenega modela in izračunane najvišje točnosti za izbrane podatkovne zbirke smo kot klasifikatorja uporabili odločitvena drevesa in naključni gozd. Za izbrana klasifikatorja smo uporabili 10-kratno prečno preverjanje in izračunali srednjo vrednost točnosti.

3.5 Odločitvena drevesa

Odločitvena drevesa so grafi, podobni drevesom, kjer vsako vozlišče predstavlja test, vsaka veja izzid testa in vsak list oznako razreda [8]. Model, ki ga dobimo z izgradnjo odločitvenega drevesa, naj bi z uporabo pravil določil ciljni razred primera. Odločitveno drevo je primer nadzorovanega učenja in ne zahteva priprave podatkov, saj deluje nad numeričnimi in diskretnimi vrednostmi atributov. Zaradi lažje interpretacije in vizualizacije odločitvenega drevesa, ga je potrebno omejiti in določiti parametre za izgradnjo.

Za gradnjo odločitvenga drevesa in iskanje parametrov drevesa smo uporabili knjižnico Scikit-learn [25]. Za ponovljivost rezultatov smo nastavili generator naključnih števil. Scikit-learn za gradnjo odločitvenega drevesa uporablja optimizirano različico CART algoritma [25].

3.5.1 Optimizacija parametrov s prečnim preverjanjem

Za iskanje parametrov smo uporabili metodo naključnega iskanja s prečnim preverjanjem klasifikatorja. Za implementacijo smo uporabili metodo *RandomizedSearchCV()* knjižnice Scikit-learn [25]. Pri naključnem iskanju podamo interval iskanja za določen parameter. Parametri iskanja so:

- minimalno število primerov za razcep
- maksimalna globina drevesa
- minimalno število primerov v listu.

Za preverjanje smo uporabili interno 10-kratno prečno preverjanje na učni množici. Izmed vseh kombinacij parametrov smo izbrali najboljše tri in z njimi zgradili odločitveno drevo, ki je bilo uporabljeno za klasifikacijo testne množice. Najboljše kombinacije parametrov smo določili glede na rezultate točnosti modela.

3.5.2 Vizualizacija odločitvenega drevesa

Knjižnica Scikit-learn nam omogoča, da zgrajeno drevo izvozimo v format DOT. DOT je tekstovni opisni jezik, ki grafe izrisuje v hierarhiji. Strukturo drevesa smo izvozili s funkcijo *export_graphviz()* in drevo izrisali z uporabo orodja Graphviz [14]. Listi drevesa vsebujejo število primerov in distribucijo primerov po razredih. Vsako vozlišče drevesa poleg vsebine lista vsebuje še logični pogoj, ki usmerja pot po drevesu.

3.6 Naključni gozd

Naključni gozd zgradi podano število odločitvenih dreves in uporabi povprečje za izboljšanje natančnosti predikcije. Za gradnjo modela naključnega gozda smo pri vseh primerih uporabili 1000 odločitvenih dreves in nastavili parameter generatorja naključnih števil za ponovljivost rezultatov. Z 10-kratnim prečnim preverjanjem smo preverili točnost, preciznost, priklic, izračunali srednjo vrednost in standardni odklon ter pridobili pomembnost atributov. Za gradnjo naključnega gozda smo uporabili knjižnico Scikit-learn [25].

3.6.1 Prečno preverjanje z naključnim gozdom

Prečno preverjanje smo realizirali s pomočjo vgrajene funkcije knjižnice Scikit-learn *cross_val_score()*, kateri kot parametre podamo klasifikator za preverjanje, ciljni razred oziroma vrednosti le tega in število ponovitev prečnega preverjanja [25]. Za deljenje podatkov na učno in testno množico z uporabo funkcije *cross_val_score()* nam ni potrebno skrbeti, saj za delitev poskrbi funkcija sama [25]. Funkcija vrne ocene vsakega prehoda prečnega preverjanja v obliki matrike. Končni rezultat točnosti smo pridobili z izračunom srednje vrednosti in standardnega odklona matrike ocen.

3.6.2 Pomembnost atributov

Da bi zgradili dober in hiter klasifikacijski model, moramo vedeti, kako pomembni so atributi in katere izmed njih lahko odstranimo. Za pridobitev pomembnosti atributov smo poleg pomembnosti atributov, ki nam jo vrne naključni gozd, uporabili tudi metodo *SelectKBest* knjižnice Scikit-learn [25].

Pomembnost atributov naključnega gozda

Klasifikator naključnih dreves knjižnice Scikit-learn nam za atribute podanih učnih podatkov vrne pomembnosti le teh v obliki seznama. V seznamu

se vedno nahajajo pomembnosti vseh atributov. Večja kot je številka za posamezen atribut, bolj pomemben je.

Metoda SelectKBest

Metoda *SelectKBest* nam s pomočjo ocenjevalnih funkcij vrne zahtevano število najbolje ocenjenih atributov. Za pridobivanje pomembnosti atributov smo uporabili implementacijo funkcije *SelectKBest()* knjižnice Scikit-learn, kateri kot parameter podamo ocenjevalno funkcijo in želeno število najbolje ocenjenih atributov [25].

Ocenjevalne funkcije metode *SelectKBest()*, ki jih lahko uporabimo pri klasifikaciji so:

- chi2

  Izračun hi-kvadrat testa med atributom in ciljnim razredom.

- f_classif

  Analiza variance.

- mutual_info_classif

  Izračun informacijskega prispevka med atributom in ciljnim razredom z entropijo.

Za ocenjevanje smo uporabili funkcijo *mutual_info_classif* [25].

3.6.3 Krivulja učenja

Uspešnost klasifikatorja oziroma našega modela, ki nam napove razred primera, merimo z razmerji med pravilno in napačno napovedanimi primeri. Ločimo pravilno pozitivne primere (TP - true positive), pravilno negativne primere (TN - true negative), napačno pozitivne primere (FP - false positive) in napačno negativne primere (FN - false negative). Krivulja učenja je graf, na katerem prikažemo točnost (angl. *accuracy*) napovedi v razmerju z velikostjo problema oziroma števila atributov v učnih podatkih. Točnost

3.2 predstavlja razmerje med vsemi pravilno napovedanimi primeri in vsemi primeri klasifikacije.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Za učenje smo uporabili naključni gozd s 1000 drevesi in 10-kratno prečno preverjanje za preverjanje modela. Pri inicializaciji klasifikatorja smo nastavili tudi parameter naključnega generatorja števil za ponovljivost rezultatov. Število atributov smo zmanjševali glede na naraščajoči vrstni red pomembnosti atributov, pridobljen z metodo *SelectKBest*. Postopek smo ponovili pri obeh opisanih metodah za pridobivanje pomembnosti atributov. V vsaki iteraciji smo iz učne množice odstranili posamezen atribut z najmanjšo oceno pomembnosti in izračunali srednjo vrednost rezultatov prečnega preverjanja funkcije *cross_val_score()*. Učne podatke smo predstavili v obliki Pandas DataFrame objekta, ki omogoča preprosto odstranjevanje atributov in manipulacijo podatkov.

3.6.4 Preciznost in priklic

Preciznost (angl. *precision*) predstavlja razmerje med številom pravilno pozitivno napovedanimi primeri in vsemi pozitivnimi primeri napovedi za ciljni razred. Enačba 3.3 prikazuje izračun preciznosti.

$$precision = \frac{TP}{TP + FP} \tag{3.3}$$

Priklic (angl. *recall*) predstavlja razmerje med pravilno pozitivnimi napovedmi in vsemi pravilnimi napovedmi. Enačba 3.4 prikazuje izračun priklica.

$$recall = \frac{TP}{TP + FN} \tag{3.4}$$

Za pridobivanje napovedi primerov smo uporabili funkcijo *cross_val_predict()* knjižnice Scikit-learn, kateri smo podali klasifikator, učne podatke, ciljni atribut in število prehodov prečnega preverjanja [25]. Tako pridobljene napovedi smo primerjali s pravilnimi napovedmi in izračunali preciznost in priklic. Kot pri krivulji učenja smo za klasifikator izbrali naključni gozd s 1000 drevesi v gozdu, uporabili 10-kratno prečno preverjanje in nastavili generator naključnih števil za ponovljivost rezultatov. Za izračun preciznosti in priklica smo uporabili funkciji *precision_score()* in *recall_score()* knjižnice Scikit-learn [25]. Preciznost in priklic smo izračunali po iteracijah odstranjevanja atributov glede na njihovo pomembnost. Pridobili smo jih z naključnim gozdom in metodo *SelectKBest*. Ocene smo predstavili v obliki grafa s pari krivulj preciznosti in priklica za vsak ciljni razred posebej.

Pri izračunu preciznosti in priklica za ADDS smo upoštevali tudi zaporedje atributov po skupinah. Zaporedje skupin atributov smo določili glede na najpomembnejši atribut v skupini. Pomembnost atributov smo pridobili z naključnim gozdom. V vsaki skupini smo označili najpomembnejši atribut in tako dobili zaporedje odstranjevanja skupin. Glede na pozicioniranje skupine nerazvrščenih atributov oziroma skupine ostalo smo ločili tri primere:

- brez ostalih atributov
- ostali atributi kot skupina na koncu
- ostali atributi kot skupina.

Skupina nerazvrščenih atributov je vsebovala 61 atributov.

3.6.5 Matrika napak

Matrika napak (angl. *confusion matrix*) nam na grafičen način predstavi število pravilnih napovedi klasifikatorja. Velikost matrike je odvisna od možnih vrednosti ciljnega razreda. Matrika napak na osi x prikazuje napovedano vrednost ciljnega razreda, na osi y pa pravilno vrednost razreda.

Za učenje nad podatki smo uporabili klasifikator naključnega gozda s 1000 drevesi in nastavili generator naključnih števil za ponovljivost rezultatov.

Napovedi za posamezen primer smo pridobili s funkcijo *cross_val_predict()* in uporabili 10-kratno prečno preverjanje. Za izračun matrike napak smo uporabili funkcijo *confusion_matrix()* knjižnice Scikit-learn in jo izrisali s knjižnico Matplotlib [25, 16]. Funkciji podamo pravilne vrednosti primerov za ciljni razred, napovedane vrednosti in možne vrednosti ciljnega razreda.

3.7 Redukcija problemskega prostora

Problemski prostor ali število atributov, ki jih vsebujejo analizirani podatki, lahko pri visokem številu pri strojnem učenju predstavljajo problem. Z zmanjšanjem problemskega prostora podatke lažje predstavimo in zmanjšamo čas izvajanja algoritmov za pridobivanje modela. Pri vsem tem pazimo, da izgubimo čim manj informacij. Problem redukcije problemskega prostora lahko razdelimo na področje izbora atributov in ekstrakcije atributov. Pri izboru atributov poiščemo podmnožico atributov in ostale zavržemo, medtem ko pri ekstrakciji atributov le te transformiramo v prostor z nižjo dimenzijo.

Za transformacijo podatkov smo uporabili metodo glavnih komponent in s tem zmanjšali problemski prostor. Pri izboru atributov smo upoštevali rezultate pomembnosti atributov. Analiza ni pokazala performančnih problemov izvajanja algoritmov, ker učni podatki niso vsebovali veliko primerov.

3.7.1 Algoritem PCA

Namen metode glavnih komponent je iz podatkov izvleči pomembne informacije in jih predstaviti z novimi ortogonalnimi spremenljivkami, imenovanimi glavne komponente [3]. Za implementacijo smo uporabili algoritem PCA knjižnice Scikit-learn v povezavi s cevovodi [25]. S cevovodom enostavno povežemo operacije transformacij atributov in klasifikator. Uporabili smo 10-kratno prečno preverjanje ob uporabi izčrpnega iskanja, kjer je bil podan parameter število glavnih komponent algoritma PCA. Kot klasifikator smo uporabili naključni gozd s 1000 iteracijami in nastavili generator naključnih števil za ponovljivost rezultatov. Postopek smo ponovili za vsa možna števila

glavnih komponent od 1 do števila atributov podatkovne zbirke in ob vsaki iteraciji preverili točnost.

Poglavje 4

Rezultati

Rezultate opisanih metod, ki smo jih uporabili pri raziskovanju, smo predstavili v obliki tabel, slik in grafov. Opis vsebuje primerjave rezultatov med posameznimi podatkovnimi zbirkami. Pri vsakem izmed poglavij smo dobljene rezultate poskušali pojasniti in ocenili kakovost dobljenih rezultatov.

4.1 Korelacije med atributi

Rezultate izračuna Spearmanovih korelacij posamezne podatkovne zbirke smo predstavili v obliki tabele z desetimi najmočneje koreliranimi atributi, ki so razvrščeni v padajočem vrstnem redu glede na absolutno vrednost korelacije. Moči korelacij, pridobljene s knjižnico Pandas, smo primerjali z vrednostmi iz Neo4j grafne baze in jih predstavili s toplotnim grafom. Na slikah toplotnih grafov so visoko pozitivno korelirani atributi označeni s temno rdečo barvo, visoko negativno korelirani atributi pa s temno modro barvo.

Tabela 4.1 prikazuje deset najmočneje koreliranih atributov ADNIDS. Opazimo lahko močno povezanost ocene ADAS in povezanost atributov testa Everday Cognition (ECog), kjer so atributi posameznega področja kognitivnih funkcij močno povezani z rezultatom celotnega testa.

Atribut	**Atribut**	**Moč korelacije**
ADAS11_bl	ADAS13_bl	0.963
EcogSPMem_bl	EcogSPTotal_bl	0.939
EcogSPLang_bl	EcogSPTotal_bl	0.876
EcogPtMem_bl	EcogPtTotal_bl	0.873
EcogSPDivatt_bl	EcogSPTotal_bl	0.863
EcogPtLang_bl	EcogPtTotal_bl	0.863
EcogSPPlan_bl	EcogSPTotal_bl	0.861
EcogSPOrgan_bl	EcogSPTotal_bl	0.835
WholeBrain_bl	ICV_bl	0.803
EcogPtDivatt_bl	EcogPtTotal_bl	0.798

Tabela 4.1: Korelacije atributov ADNIDS

Tabela 4.2 prikazuje deset najmočneje koreliranih atributov NMDS. Moč korelacij pri NMDS je v primerjavi z ostalima podatkovnima zbirkama opazno manjša.

Atribut	**Atribut**	**Moč korelacije**
BMP-4_1	BMP-6_1	0.716
IL-13_1	IL-15_1	0.674
NT-4_1	OSM_1	0.654
Fit-3 Ligand_1	Fractalkine_1	0.647
BMP-6_1	CNTF_1	0.643
IGF-1 SR	IL-1R4 /ST2_1	0.640
AXL_1	bFGF	0.639
Fit-3 Ligand_1	GCP-2_1	0.630
BMP-4_1	CNTF_1	0.615
GCP-2_1	IL-13_1	0.613

Tabela 4.2: Korelacije atributov NMDS

Tabela 4.3 prikazuje deset najmočneje koreliranih atributov ADDS. Razlog

popolne funkcijske povezanosti med atributoma „DezinhibicijaPonasanja“ in „FluktuacijaKognicije“ je ta, da je vrednost obeh atributov le pri enem primeru različna od nič. Opazimo lahko močno koreliranost med atributi testa „RAVLT“, kjer so posamezni atributi testa močno korelirani z skupno oceno testa. Med atributoma „WCST.NetacniOdgovori“ in „WCST.Procenat OdrzavanjaKonceptualnogNivoa“ opazimo negativno povezanost. Ob upadu konceptualnega nivoja se število napačnih odgovorov poveča, kar pojasnjuje negativno povezanost med atributoma.

Atribut	**Atribut**	**Moč korelacije**
DezinhibicijaPonasanja	FluktuacijaKognicije	1
WCST.NetacniOdgovori	WCST.ProcenatOdrzavanja KonceptualnogNivoa	- 0.971
ROCF.3Minuta	ROCF.45Minuta	0.948
RAVLT.A4	RAVLT.RAVLTUkupno	0.947
RAVLT.A3	RAVLT.RAVLTUkupno	0.931
RAVLT.A5	RAVLT.RAVLTUkupno	0.930
ACEIII.Fluentnost	KategorijalnaFluentnost UkupniSkor	0.922
RAVLT.A4	RAVLT.A5	0.902
ACEIII.UkupniSkor	ACEIII.Pamcenje	0.899
RAVLT.A2	RAVLT.RAVLTUkupno	0.893

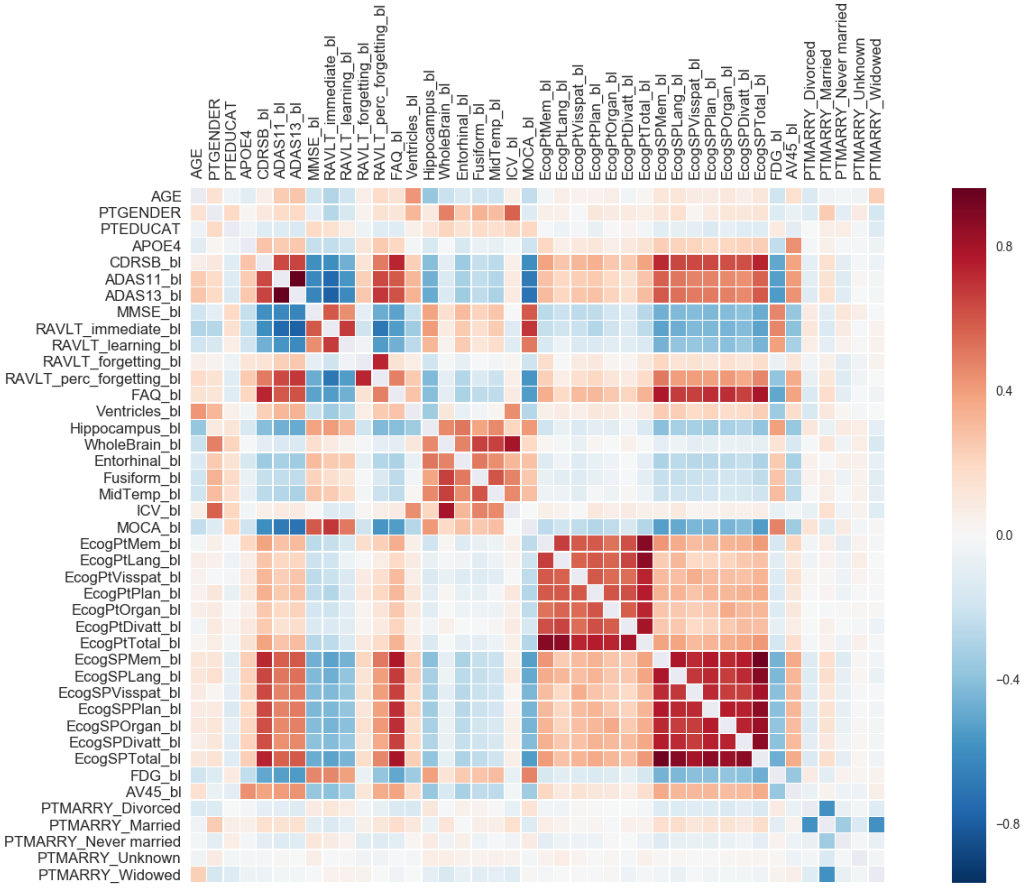
Tabela 4.3: Korelacije atributov ADDS

Tabela 4.4 prikazuje število atributov posamezne podatkovne zbirke, glede na interval moči povezanosti. Atribut je bil uvrščen v določen interval glede na absolutno vrednost maksimalne moči korelacije z ostalimi atributi. Pri ADDS opazimo, da 17 atributov nima nikakršne povezanosti z ostalimi. Večina atributov NMDS pripada srednjemu intervalu povezanosti, kar pojasni opazno manjše moči korelacij med desetimi najmočneje koreliranimi atributi v tabeli 4.2.

Interval moči povezanosti	ADNIDS	NMDS	ADDS
0,00 ni povezanosti	2	0	17
[0.01, 0.2) neznatna povezanost	0	0	0
[0.20, 0.4) nizka/šibka povezanost	1	26	24
[0.40, 0.7) srednja/zmerna povezanost	13	92	57
[0.70, 0.9) visoka/močna povezanost	18	2	33
[0.90, 1) zelo visoka/močna povezanost	4	0	10
1 popolna (funkcijska) povezanost	0	0	2

Tabela 4.4: Število atributov z maksimalno močjo korelacije po intervalih moči povezanosti posamezne podatkovne zbirke

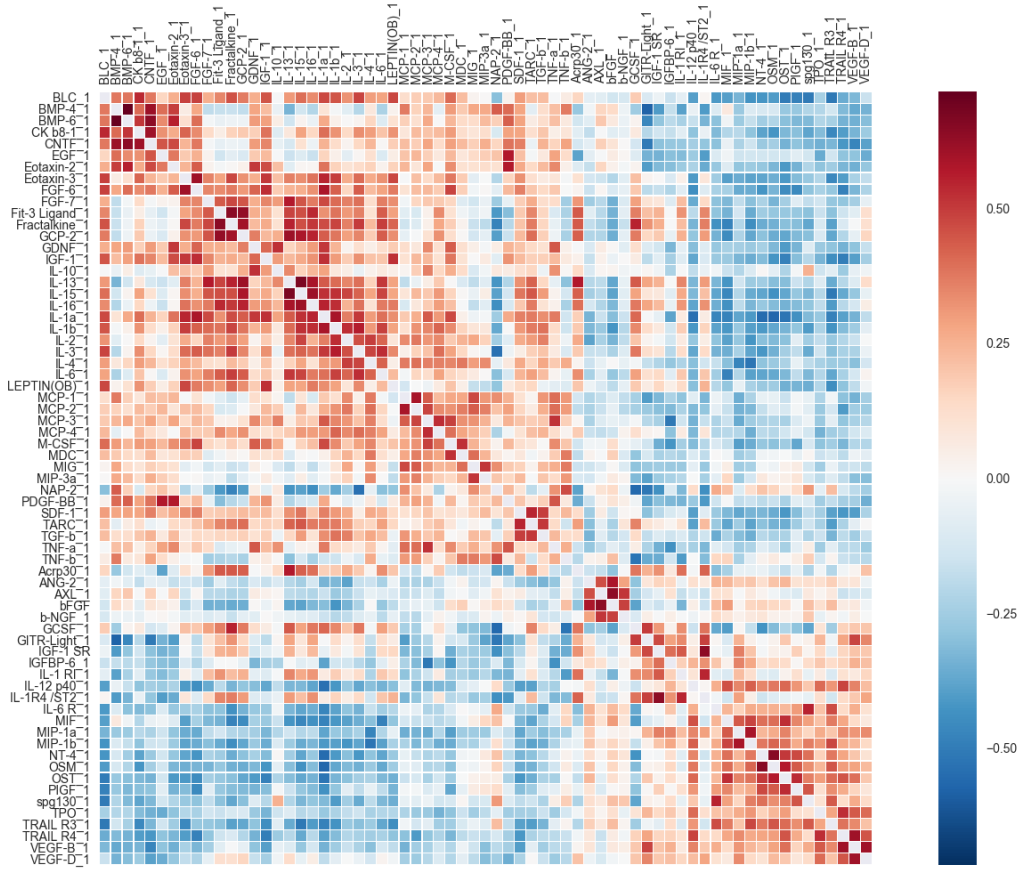
Slika 4.1 prikazuje primer vizualizacije korelacij atributov ADNIDS, za katero smo uporabili toplotni graf knjižnice Seaborn. Na toplotnem grafu so predstavljeni vsi atributi podatkovne zbirke ADNIDS, ki so bili upoštevani pri analizi. Z modro barvo so označene negativne korelacije, z rdečo barvo pa pozitivne. Opazimo lahko večja temno rdeča in temno modra območja, kjer so korelacije med atributi visoke. Pričakovano se kvadratna območja atributov, ki pripadajo posameznem testu, obarvajo intenzivnejše. Izstopajo atributi testa Everyday Cognition (ECog), ki ocenjuje kognitivne funkcije. Med atributi testa opazimo dve večji skupini, ki ju lahko pojasnimo z načinom ocenjevanja, ki vključuje ocenjevanje pacienta in sodelujočega partnerja. Pri atributih testa Everyday Cognition za sodelujočega partnerja opazimo močno koreliranost z lestvico ADAS in močno negativno koreliranost s testi RAVLT, medtem ko so pri atributih pacienta korelacije šibkejše.



Slika 4.1: Toplotni graf korelacij atributov ADNIDS

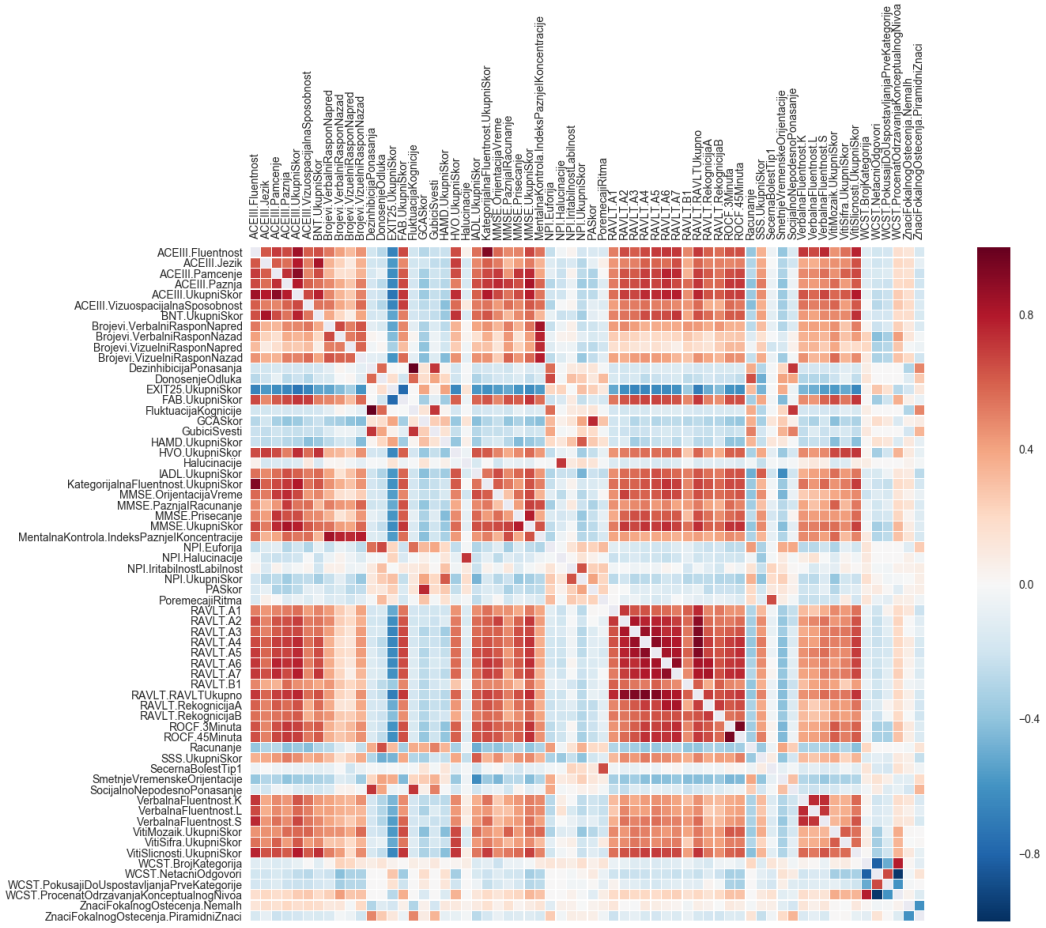
Toplotni graf 4.2 prikazuje korelacije NMDS za atribute, pri katerih je maksimalna vrednost korelacija moči večja od 0,5. Pri tem smo upoštevali moč korelacije po absolutni vrednosti. Prikaz na toplotnem grafu smo zaradi prevelikega števila atributov podatkovne zbirke omejili.

Na sliki 4.2 lahko opazimo dve večji skupini atributov glede na pozitivne moči korelacij (področja rdeče barve). Prva skupina se nahaja v levem zgornjem kotu, kar predstavlja približno $\frac{2}{3}$ atributov na sliki. Druga skupina se nahaja v spodnjem desnem kotu. Večina atributov iz prve in druge skupine ima negativne moči korelacij, kar lahko opazimo kot večje modro območje v spodnjem levem kotu.



Slika 4.2: Toplotni graf korelacij atributov NMDS

Toplotni graf 4.3 prikazuje korelacije ADDS za atribute, pri katerih je maksimalna vrednost korelacija moči večja od 0,6. Pri tem smo upoštevali moč korelacije po absolutni vrednosti. Število atributov smo zaradi jasnosti prikaza na toplotnem grafu omejili. Pri ADDS toplotnem grafu smo imena atributov zaradi določenih skupin uredili po abecednem vrstnem redu glede na ime atributa. Tako lahko opazimo skupine atributov pripadajočih testov, ki se ob diagonali grafa na sliki odražajo kot večja območja rdeče barve. Območja rdeče barve, ki so oddaljena od diagonale, predstavljajo korelacije med skupinami testov.



Slika 4.3: Toplotni graf korelacij atributov ADDS

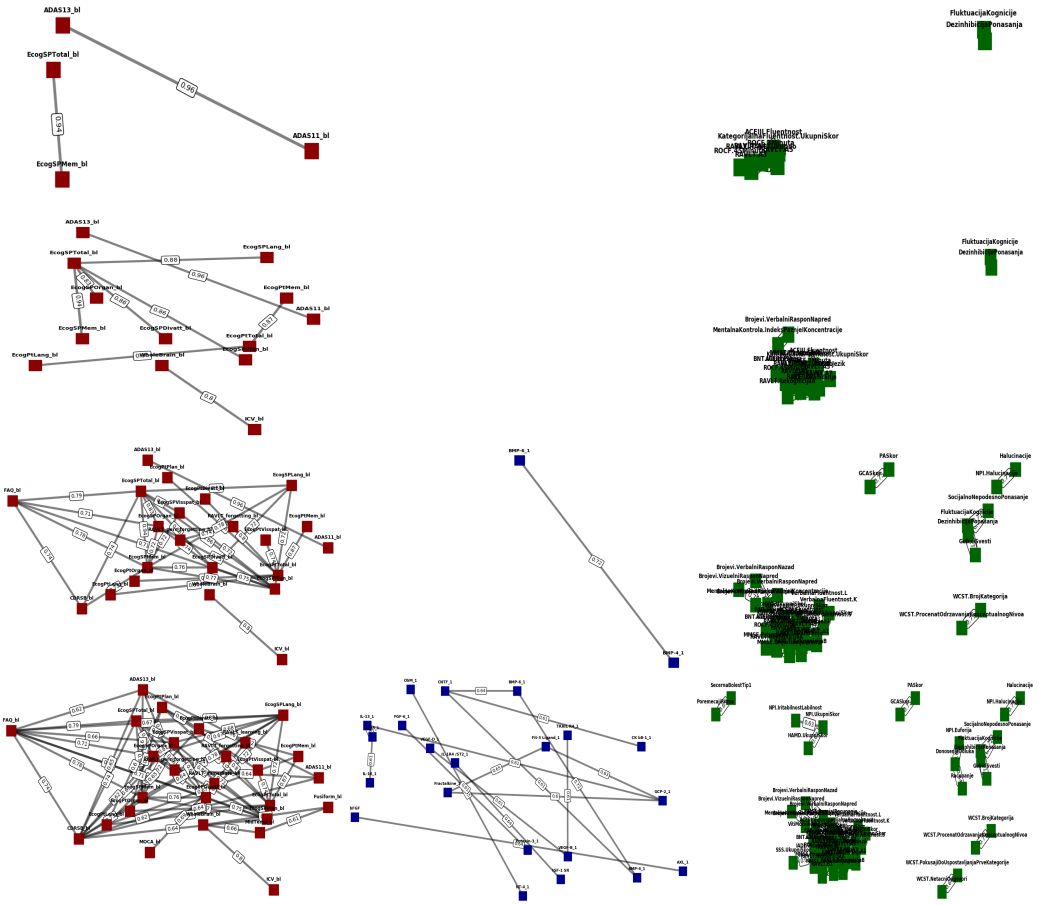
4.1.1 Grafi korelacij

Grafe atributov in korelacij med njimi smo izrisali v vmesniku grafne baze Neo4j in z uporabo knjižnice Matplotlib [16].

Slika 4.4 prikazuje primer grafa v spletnem vmesniku baze Neo4j, kjer je stopnja korelacije višja od 0,7 za ADNIDS. Opazimo, da so povezave med vozlišči usmerjene, zato tak način prikaza grafa ne ustreza.

Slika 4.4: Spletni vmesnik grafne baze Neo4j za atribute ADNIDS

Slika 4.5 prikazuje primere grafov izrisanih s knjižnice matplotlib, kjer stopnje korelacije sledijo intervalu od 0,9 do 0,6 s korakom 0,1 za posamezen graf od zgoraj navzdol. Rdeča vozlišča predstavljajo atribute ADNIDS, modra vozlišča NMDS in zelena vozlišča ADDS. Pri zaporedju grafov za NMDS se zaradi nizko koreliranih atributov vozlišča izrišejo šele na grafu, kjer so stopnje korelacije višje od 0,7. Opazimo, da so povezave med vozlišči neusmerjene, zato je tak način prikaza grafov ustreznejši, od prikaza v vmesniku Neo4j grafne baze.



ADNIDS NMDS ADDS

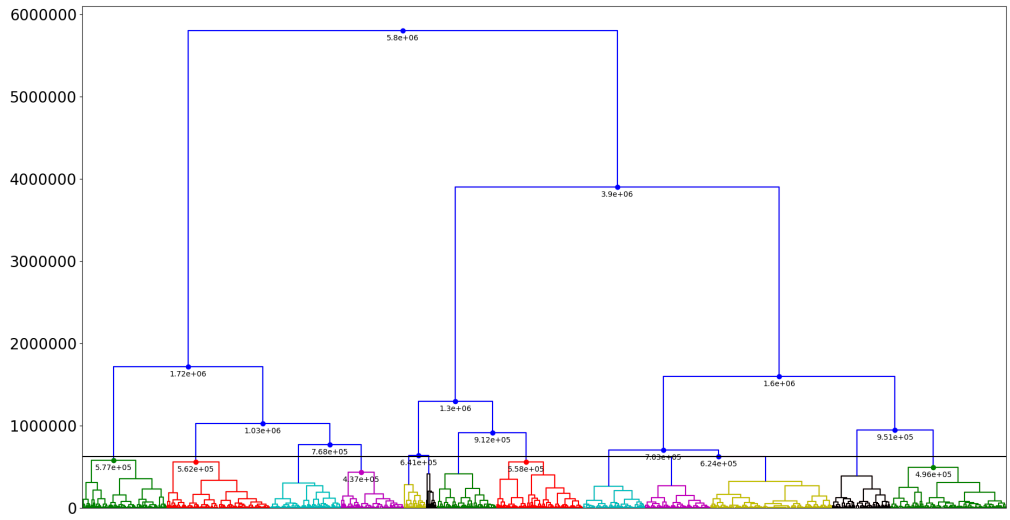
Slika 4.5: Grafi atributov in korelacij posamezne podatkovne zbirke, izrisani s knjižnico Matplotlib

4.2 Rezultati hierarhičnega gručenja

Pridobljeno hierarhijo gručenja smo predstavili z dendrogramom. Pri določanju števila gruč smo upoštevali smiselno število gruč in velikosti razdalj med gručami v hierarhiji.

4.2.1 Hierarhija ADNIDS

Dendrogram na sliki 4.6 prikazuje dobljeno hierarhijo gruč ADNIDS z uporabo Ward-ove metode. Ker ADNIDS vsebuje preveliko število primerov za prikaz, smo oznake primerov odstranili. Na sliki opazimo, da se gruče hierarhije združujejo na zelo veliki razdalji. Hierarhijo gruč smo prerezali na razdalji 623000 in tako dobil 13 gruč obarvanih vsaka s svojo barvo. Prerez označuje črna horizontalna črta na sliki. Oznake razdalj, na kateri se združita gruči, smo zaradi jasnosti prikaza dendrograma prikazali nad določeno mejo.



Slika 4.6: Dendrogram za ADNIDS z uporabo Ward-ove metode združevanja gruč

Tabela 4.5 prikazuje distribucijo primerov gruč, večinski razred gruče in čistost gruče, izračunano glede na večinski razred gruče. Oznake gruč sledijo vrstnemu redu gruč dendrograma 4.6 od leve proti desni. Opazimo da zgrajena hierarhija vsebuje gruče z nizko čistostjo glede na večinski razred, zato lahko zaključimo, da postopek hierarhičnega gručenja na osnovi podobnosti med primeri ne najde pravih povezav med podobnostjo in razredom.

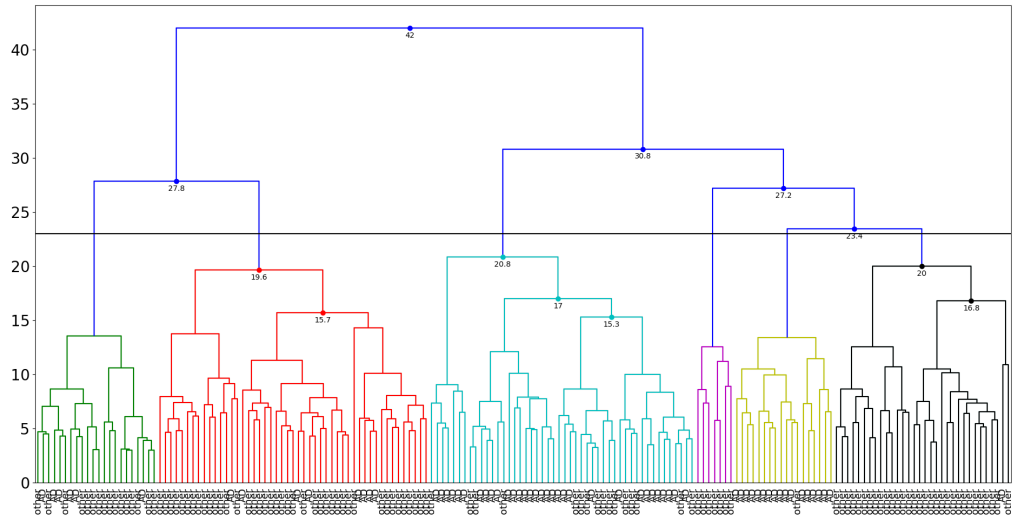
Gruča	Distribucija primerov [AD/CN/EMCI/LMCI/SMC]	Večinski razred	Čistost gruče
1	20/21/23/14/6	EMCI	≈ 27 %
2	10/27/34/15/17	EMCI	≈ 33 %
3	21/14/19/12/4	AD	≈ 30 %
4	20/8/21/10/4	EMCI	≈ 33.3 %
5	6/3/4/7/2	LMCI	≈ 31.8 %
6	3/1/2/3/2		
7	17/11/14/13/4	AD	≈ 28.8 %
8	8/18/41/10/9	EMCI	≈ 47.7 %
9	5/10/23/10/14	EMCI	≈ 37 %
10	2/14/25/15/9	EMCI	≈ 38.5 %
11	16/28/40/20/17	EMCI	≈ 33 %
12	14/6/17/16/5	EMCI	≈ 29.3 %
13	8/27/47/20/13	EMCI	≈ 40.9 %

Tabela 4.5: Distribucija števila primerov ciljnega razreda ADNIDS po hierarhiji gručenja

S prečnim preverjanjem hierarhičnega gručenja smo dobili ≈ 18.5 % točnost, kjer smo prav tako upoštevali hierarhijo 13 gruč.

4.2.2 Hierarhija NMDS

Slika 4.7 prikazuje izris dendrograma za NMDS. Dendrogram smo razrezali na razdalji 23 in tako dobili šest gruč. Vsaka gruča je predstavljena s svojo barvo. Razrez označuje črna horizontalna črta na sliki 4.7.



Slika 4.7: Dendrogram za NMDS z uporabo Ward-ove metode združevanja gruč

Tabela 4.6 prikazuje število primerov posamezne gruče glede na ciljni razred. Številke gruč označujejo zaporedje gruč na dendrogramu 4.7 od leve proti desni. V štirih gručah kot večinski razred nastopajo primeri, ki imajo vrednost ciljnega razreda „other“ in dveh gručah z vrednostjo „AD“.

Gruča	**Distribucija primerov [AD / other]**	**Večinski razred**	**Čistost gruče**
1	6 / 16	other	$\approx$ 73 %
2	8 / 41	other	$\approx$ 84 %
3	32 / 16	AD	$\approx$ 67 %
4	0 / 7	other	100 %
5	17 / 1	AD	$\approx$ 94 %
6	1 / 31	other	$\approx$ 97 %

Tabela 4.6: Distribucija števila primerov ciljnega razreda NMDS po hierarhiji gručenja

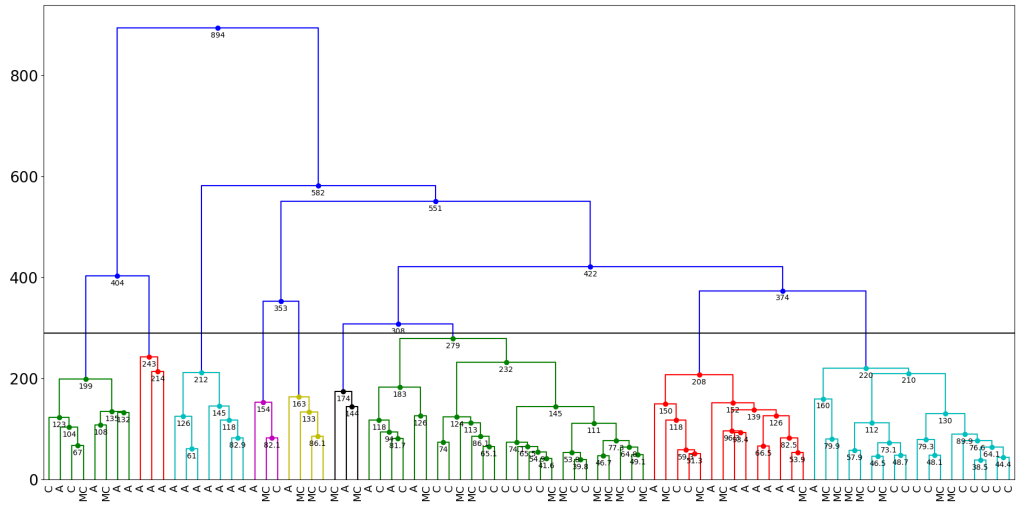
Čistost posamezne gruče v tabeli 4.6 smo izračunali glede na večinski ra-

zred gruče. Ob predpostavki, da kot klasifikator uporabimo distribucijo primerov po gručah, smo izračunali ≈ 85.5 % točnost, ki smo jo izračunali kot povprečje čistosti gruč.

S prečnim preverjanjem hierarhičnega gručenja smo dobili ≈ 88.9 % točnost, kjer smo prav tako upoštevali hierarhijo šestih gruč.

4.2.3 Hierarhija ADDS

Slika 4.8 prikazuje hierarhijo gruč za ADDS, dobljeno z uporabo Ward-ove metode. Prerez, ki ga označuje črna horizontalna črta na sliki dendrograma, smo določili na razdalji 290 in dobili devet gruč. Na sliki lahko opazimo, da se dve izmed dobljenih gruč glede na združevanje ostalih gruč združita na dokaj veliki razdalji.



Slika 4.8: Dendrogram za ADDS z uporabo Ward-ove metode združevanja gruč

Tabela 4.7 prikazuje distribucijo primerov posamezne gruče. Oznake gruč sledijo zaporedju, dobljenem pri hierarhičnem gručenju na sliki 4.8, od leve proti desni. Za vsako gruče smo določili večinski razred gruče in izračunali čistost gruče. Popolna čistost dveh gruč primerov Alzheimerjeve bolezni

nakazuje, da lahko na podlagi podobnosti zelo dobro ločimo primere Alzheimerjeve bolezni od ostalih.

Gruča	**Distribucija primerov [A/C/MC]**	**Večinski razred**	**Čistost gruče**
1	4/2/2	A	50 %
2	3/0/0	A	100 %
3	7/0/0	A	100 %
4	1/1/1		
5	1/1/2	MC	50 %
6	1/0/2	MC	$\approx$ 33.3 %
7	3/13/9	C	52 %
8	8/2/4	A	$\approx$ 57 %
9	1/10/7	C	$\approx$ 55.6 %

Tabela 4.7: Distribucija števila primerov ciljnega razreda ADDS po hierarhiji gručenja

S prečnim preverjanjem hierarhičnega gručenja smo dobili $\approx$ 55.6 % točnost, kjer smo prav tako upoštevali hierarhijo devetih gruč.

4.2.4 Povzetek rezultatov gručenja

Tabela 4.8 prikazuje rezultate hierahičnega gručenja s prečnim preverjanjem, kjer smo upoštevali število gruč hierarhije posamezne podatkovne zbirke. Opazimo, da točnost upada z naraščanjem števila gruč. Najvišjo točnost smo dobili pri gručenju NMDS s šestimi gručami in najnižjo pri ADNIDS s trinajstimi gručami.

Podatkovna zbirka	Število gruč	Točnost
ADNIDS	13	$\approx 18.5\ \%$
NMDS	6	$\approx 88.9\ \%$
ADDS	9	$\approx 55.6\ \%$

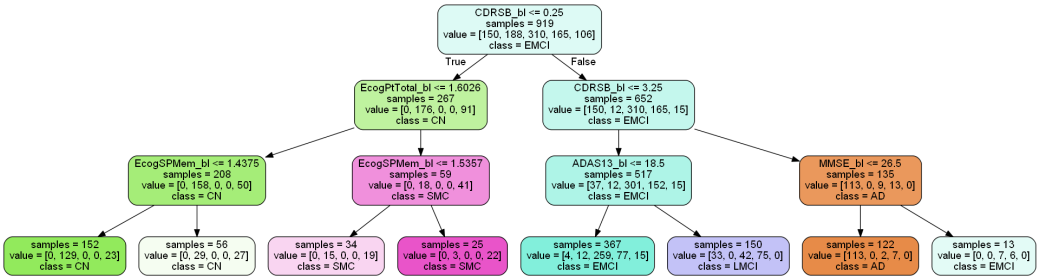
Tabela 4.8: Rezultati točnosti hierarhičnega gručenja s prečnim preverjanjem posamezne podatkovne zbirke

4.3 Prikaz odločitvenega drevesa

Vsako vozlišče odločitvenega drevesa vsebuje pravilo odločanja, število primerov v vozlišču (angl. *samples*), distribucijo primerov po razredih (angl. *value*) in večinski razred vozlišča (angl. *class*). Vozlišča so obarvana glede na večinski razred vozlišča, kar nam omogoča hiter pregled drevesa po ciljnem razredu. Intenzivnost obarvanosti nakazuje čistost večinskega razreda v vozlišču.

4.3.1 Odločitveno drevo ADNIDS

Slika 4.9 prikazuje primer izrisa drevesa za ADNIDS z uporabo orodja Graphviz. Globino drevesa smo omejili na tri, omejili število primerov v listu na štiri in nastavili minimalno število primerov za razcep, tj. osem.



Slika 4.9: Odločitveno drevo za ADNIDS z distribucijo primerov ciljnega razreda v vozliščih, obarvanih glede na večinski razred in čistost vozlišča

Tabela 4.9 prikazuje distribucijo primerov listov odločitvenega drevesa na sliki 4.9, kjer številke listov sledijo zaporedju listov na sliki od leve proti desni. Povprečje čistosti vseh listov drevesa je $\approx$ 67 %. V tabeli 4.9 lahko opazimo zelo uspešno klasifikacijo primerov Alzheimerjeve bolezni v listu 7.

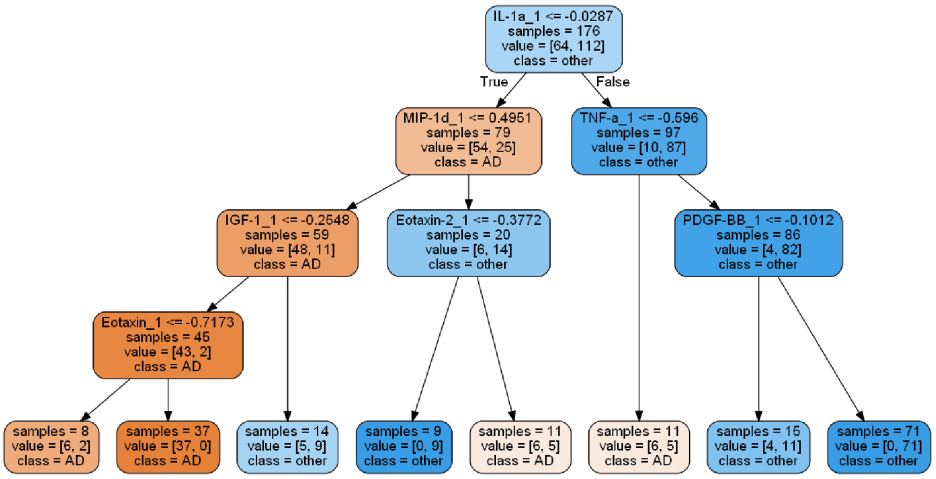
List	**Distribucija primerov [AD/CN/EMCI/LMCI/SMC]**	**Večinski razred**	**Čistost lista**
1	0/129/0/0/23	CN	$\approx$ 85 %
2	0/29/0/0/27	CN	$\approx$ 52 %
3	0/15/0/0/19	SMC	$\approx$ 56 %
4	0/3/0/0/22	SMC	88 %
5	4/12/259/77/15	EMCI	$\approx$ 71 %
6	33/0/42/75/0	LMCI	50 %
7	113/0/2/7/0	AD	$\approx$ 93 %
8	0/0/7/6/0	EMCI	$\approx$ 54 %

Tabela 4.9: Distribucija števila primerov v listih odločitvenega drevesa ADNIDS

4.3.2 Odločitveno drevo NMDS

Slika 4.10 prikazuje primer izrisa drevesa za NMDS z uporabo orodja Graphviz. Globino drevesa smo omejili na štiri, omejili število primerov v listu na

osem in nastavili minimalno število primerov za razcep, tj. tri. Prva vrednost vektorja distribucije primerov predstavlja število primerov z Alzheimerjevo boleznijo, druga pa število primerov nedementnega stanja.



Slika 4.10: Odločitveno drevo za NMDS z distribucijo primerov ciljnega razreda v vozliščih, obarvanih glede na večinski razred in čistost vozlišča

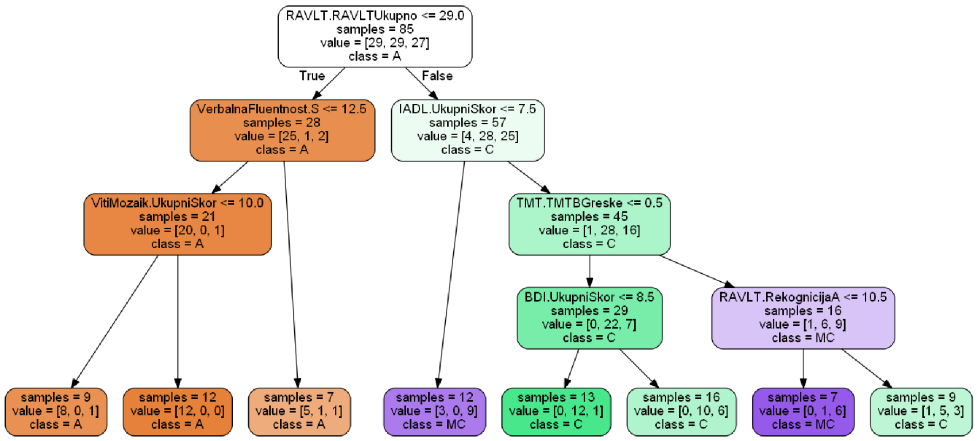
Na podlagi popolne čistosti treh listov, ki vsebujejo približno $\frac{2}{3}$ primerov drevesa lahko sklepamo, da je točnost klasifikatorja odločitvenega drevesa visoka. Tabela 4.10 prikazuje distribucijo primerov listov odločitvenega drevesa na sliki 4.10, kjer številke listov sledijo zaporedju listov na sliki od leve proti desni. Točnost dobljenega klasifikatorja je ≈ 77.8 %. Ob uporabi 10-kratnega prečnega preverjanja smo z enakimi parametri klasifikatorja, zabeležili 73 % točnost.

List	Distribucija primerov	Večinski razred	Čistost lista
1	6 / 2	AD	75 %
2	37 / 0	AD	100 %
3	5 / 9	other	≈ 64 %
4	0 / 9	other	100 %
5	6 / 5	AD	≈ 55 %
6	6 / 5	AD	≈ 55 %
7	4 / 11	other	≈ 73 %
8	0 / 71	other	100 %

Tabela 4.10: Distribucija števila primerov v listih odločitvenega drevesa NMDS

4.3.3 Odločitveno drevo ADDS

Slika 4.11 prikazuje primer izrisa drevesa za ADDS z uporabo orodja Graphviz. Globino drevesa smo omejili na štiri, omejili število primerov v listu na sedem in nastavili minimalno število primerov za razcep, tj. dva. Na sliki lahko opazimo, da skoraj celotno levo stran drevesa določajo primeri z Alzheimerjevo boleznijo, iz česar lahko sklepamo, da klasifikator te primere dobro loči od ostalih. Primere kontrolne skupine in primere s kognitivnim poslabšanjem pogosto napačno klasificiramo.

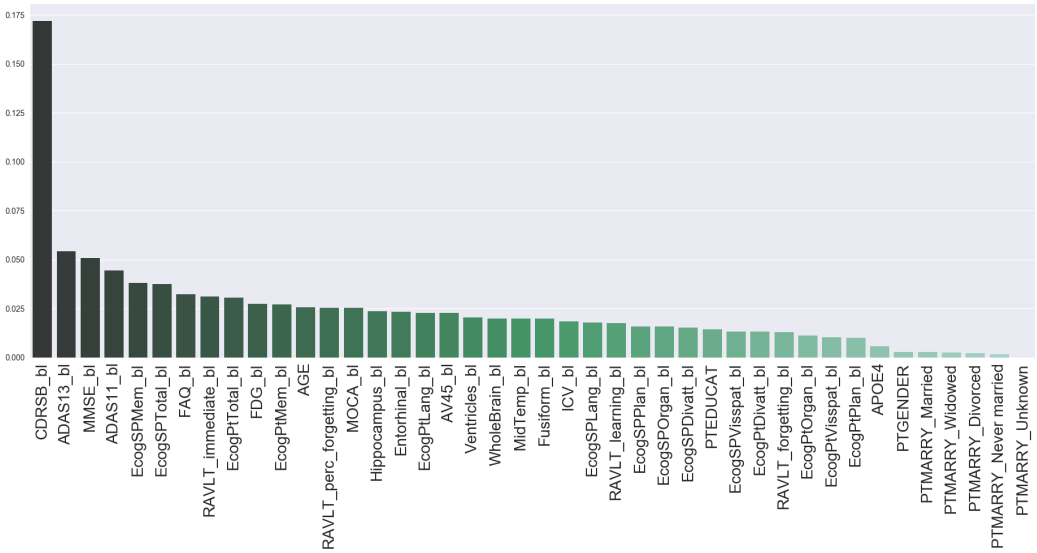


Slika 4.11: Odločitveno drevo za ADDS z distribucijo primerov ciljnega razreda v vozliščih, obarvanih glede na večinski razred in čistost vozlišča

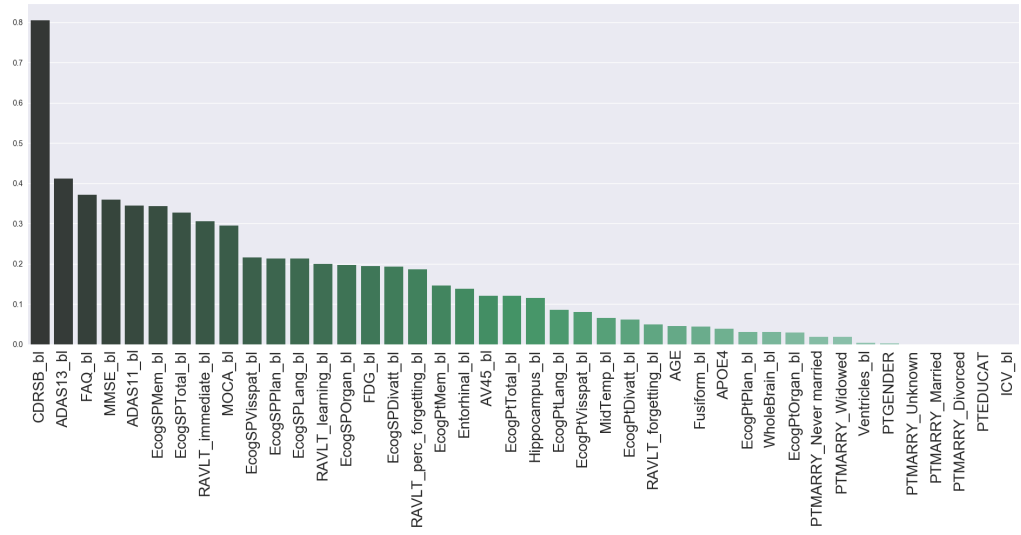
4.4 Primerjava pomembnosti atributov

Pomembnost atributov smo pridobili z naključnim gozdom in metodo *SelectKBest* knjižnice Scikit-learn z oceno informacijskega prispevka [25]. Pridobljena zaporedja atributov smo primerjali med sabo glede na mesto pomembnosti posameznega atributa.

Slika 4.12 prikazuje pomembnosti atributov, pridobljene z naključnim gozdom, v padajočem vrstnem redu za ADNIDS. Za izris smo uporabili tip grafa barplot knjižnice Seaborn.



Slika 4.12: Pomembnost atributov, dobljenih z naključnim gozdom za ADNIDS



Slika 4.13: Pomembnost atributov metode *SelectKBest*, dobljenih z ocenjevalno funkcijo *mutual_info_classif()* za ADNIDS

Na sliki 4.13 vidimo vrstni red pomembnosti atributov, pridobljen z me-

todo *SelectKBest* za ADNIDS, ki se nekoliko razlikuje od pomebnosti atributov naključnega gozda. Za pridobivanje zaporedja pomembnosti atributov je bila uporabljena ocenjevalna funkcija *mutual_info_classif()*. Pri pomembnosti atributov metode *SelectKBest* lahko opazimo večje razlike v razmerju pomembnosti med atributi kot pa pri pomembnosti atributov, dobljeni z naključnim gozdom.

4.5 Točnost, preciznost in priklic

Rezultate točnosti smo predstavil v obliki tabele in grafa krivulje učenja. Preciznost in priklic smo predstavili z grafom za posamezno vrednost ciljnega razreda podatkovne zbirke. Za vsako izmed vrednosti ciljnega razreda smo izrisali par krivulj preciznosti in priklica. Točnost, preciznost in priklic smo računali ob zmanjševanju števila atributov podatkovne zbirke, katerih zaporedje je ustrezalo pomembnosti atributov, pridobljeni z metodo *SelectKBest*.

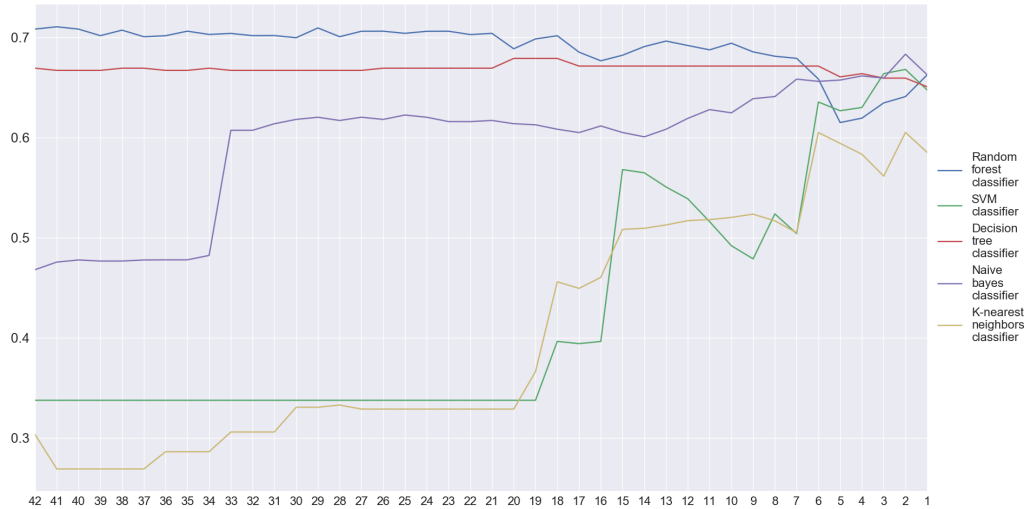
4.5.1 Rezultati točnosti

Tabela 4.11 prikazuje izmerjene točnosti klasifikatorjev posamezne podatkovne zbirke. Pri vsakem izračunu je uporabljeno 10-kratno prečno preverjanje. Najvišje točnosti klasifikatorjev smo zabeležili pri NMDS. Visoke razlike točnosti med klasifikatorji lahko opazimo pri ADNIDS in ADDS. Pri klasifikatorju k-najbližjih sosedov smo uporabili privzeto vrednost parametra števila sosedov, tj. pet. Za klasifikator odločitvenega drevesa smo upoštevali izbrane parametre, tako kot pri gradnji odločitvenega drevesa.

Klasifikator	**Točnost ADNIDS**	**Točnost NMDS**	**Točnost ADDS**
Metoda podpornih vektorjev	33.7 %	76.3 %	38.6 %
Odločitveno drevo	66.9 %	73 %	63.3 %
Naključni gozd	70.8 %	80.3 %	73.1 %
Naivni Bayes	46.8 %	78.2 %	66.1 %
K-najbližjih sosedov	30.3 %	77 %	60 %

Tabela 4.11: Točnosti klasifikatorjev posamezne podatkovne zbirke

Slika 4.14 prikazuje krivulje učenja za izbrane klasifikatorje ADNIDS. Na osi x vidimo število atributov podatkovne zbirke, na osi y pa izmerjeno točnost. Opazimo lahko, da klasifikatorja metode podpornih vektorjev in klasifikator k-najbližjih sosedov v primerjavi s klasifikatorjem odločitvenega drevesa ali naključnega gozda dosežeta zelo nizko točnost. Ob zmanjševanju števila atributov metodi klasifikacije dosežeta približno enake rezultate točnosti kot ostale.

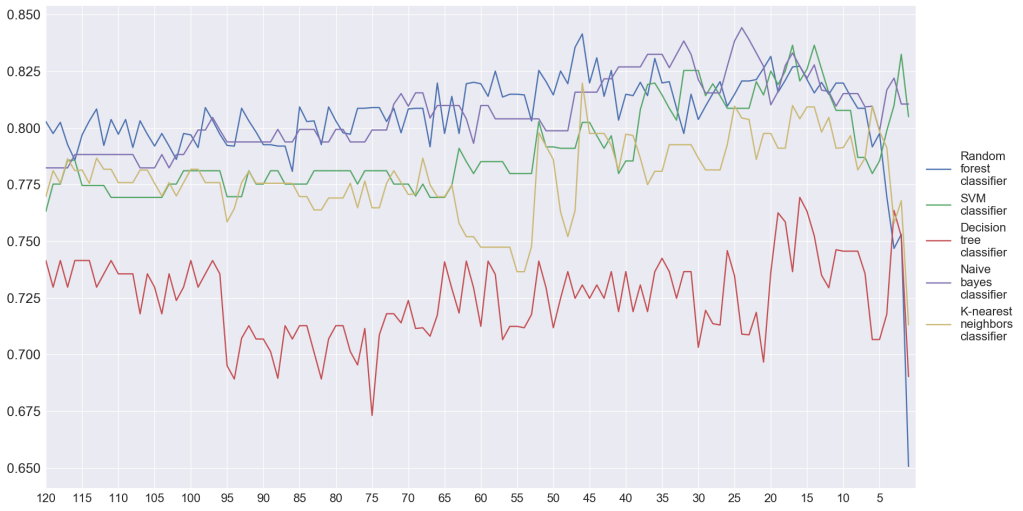


Slika 4.14: Krivulja učenja ADNIDS z uporabljenimi klasifikatorji

Maksimalno točnost (71.1 %) smo dosegli pri 41 atributih s klasifikatorjem

naključnega gozda. Število atributov smo zmanjšali tako, da smo odstranjevali atribute enega za drugim, od najmanj do najbolj pomembnega, po oceni informacijskega prispevka z metodo *SelectKBest*.

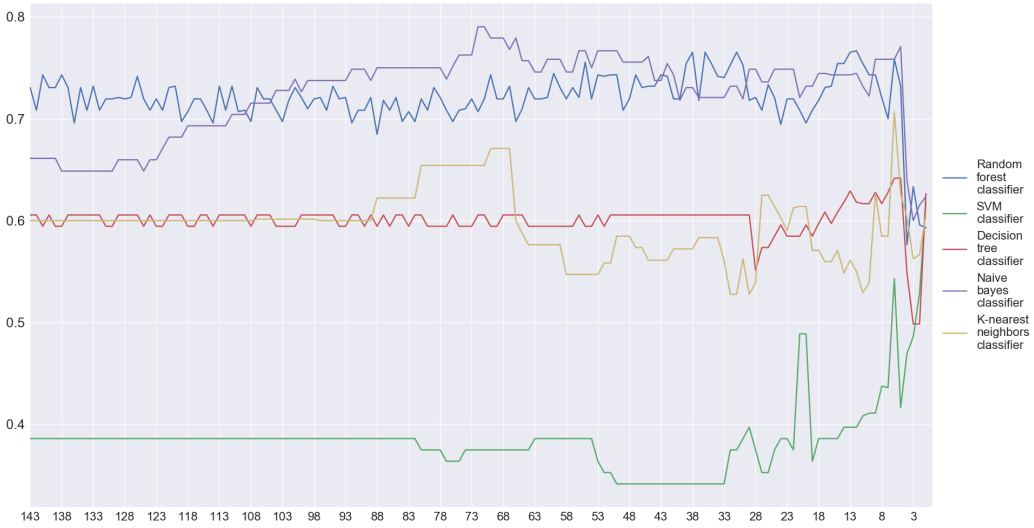
Na sliki 4.15, ki prikazuje krivuljo učenja za NMDS opazimo, da se klasifikator odločitvenega drevesa izkaže za najslabšega. Na osi x vidimo število atributov učne množice, na osi y pa izmerjeno točnost. Odstranjevanje atributov sledi zaporedju, pridobljenemu z metodo *SelectKBest* in uporabljeno oceno informacijskega prispevka s funkcijo *mutual_info_classif()*.



Slika 4.15: Krivulja učenja NMDS z uporabljenimi klasifikatorji

Maksimalno točnost (84.1 %) smo s klasifikatorjem naključnega gozda dosegli pri 46 atributih.

Slika 4.16 prikazuje primer krivulje učenja za ADDS, kjer je na osi x število atributov, na osi y pa izmerjena točnost posameznega klasifikatorja. Odstranjevanje atributov sledi zaporedju pomembnosti atributov, pridobljenih z metodo *SelectKBest*, pri kateri je bila kot ocenjevalna funkcija uporabljena *mutual_info_classif()*. Pri ADDS smo dobili najslabše rezultate točnosti za klasifikator metode podpornih vektorjev.



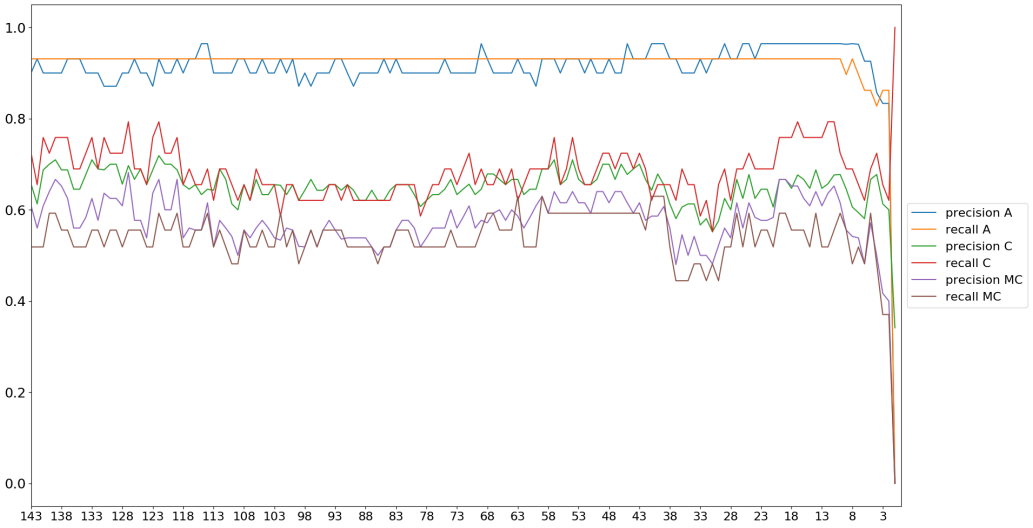
Slika 4.16: Krivulja učenja ADDS z uporabljenimi klasifikatorji

Maksimalno točnost (76.6 %) smo s klasifikatorjem naključnega gozda dosegli pri 12 atributih.

Pri večini meritev točnosti klasifikatorjev smo opazili, da točnost narašča ob odstranjevanju atributov z nizko pomembnostjo. Pri vseh treh podatkovnih zbirkah bi lahko odstranili približno $\frac{2}{3}$ atributov in dosegli višjo ali enako klasifikacijsko točnost našega modela.

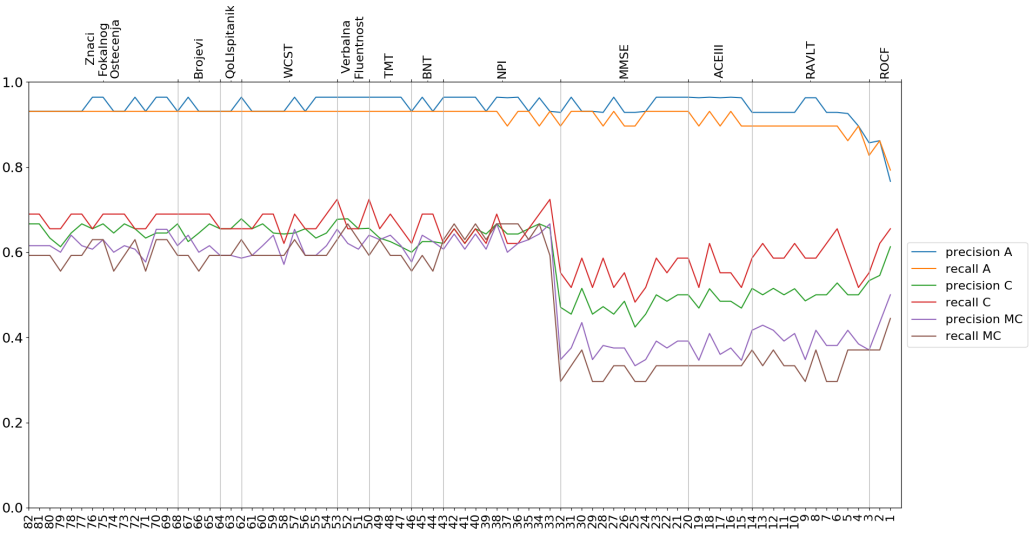
4.5.2 Prikaz preciznosti in priklica po razredih

Slika 4.17 prikazuje primer izračuna preciznosti in priklica za vse tri razrede podatkovne zbirke ADDS ob zmanjševanju števila atributov glede na pomembnost atributov, pridobljeno z metodo *SelectKBest* in oceno informacijskega prispevka. Na sliki je razvidna očitna razlika med ocenami preciznosti in priklica za primere bolnikov z Alzheimerjevo boleznijo.



Slika 4.17: Preciznost in priklic za ADDS

Na sliki 4.18 vidimo primer izračuna preciznosti in priklica za vse tri razrede podatkovne zbirke ADDS po skupinah brez skupine ostalih atributov. Vertikalne črte predstavljajo meje med določenimi skupinami atributov.

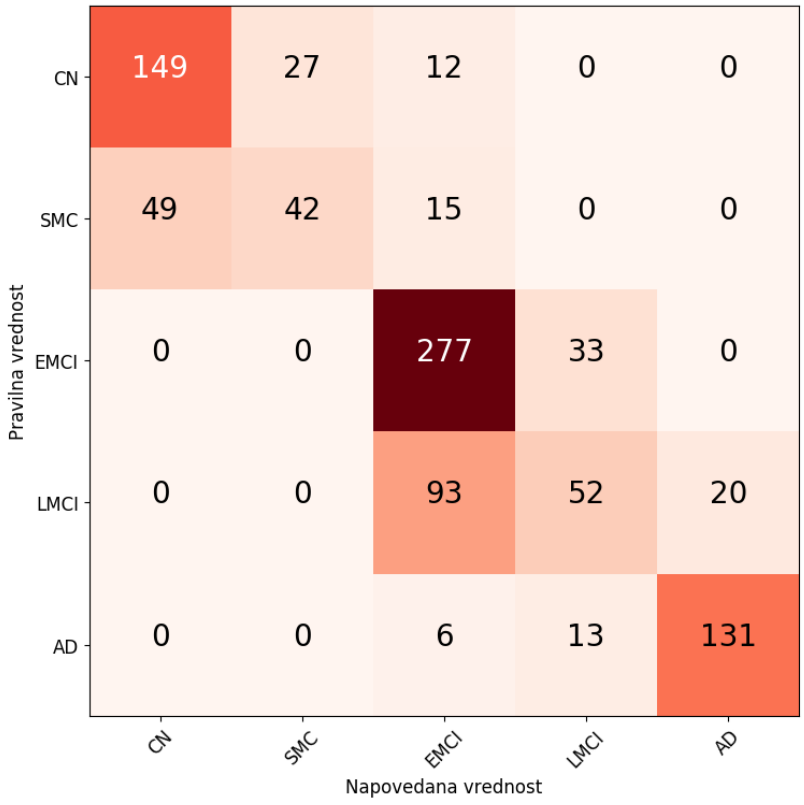


Slika 4.18: Preciznost in priklic po skupinah atributov za ADDS

4.6 Matrika napak

Matrika napak nam omogoča pregled kakovosti klasifikatorja. Diagonalne vrednosti posameznega ciljnega razreda prikazujejo število pravilnih napovedi primerov, medtem ko vrednosti izven diagonale prikazujejo število napačnih napovedi posamezne vrednosti ciljnega razreda. Intenzivnost obarvanosti posameznega elementa matrike nakazuje na število primerov.

Slika 4.19 prikazuje matriko napak za ADNIDS, kjer so na levi strani označene pravilne vrednosti primerov, na spodnji strani pa napovedane vrednosti primerov.

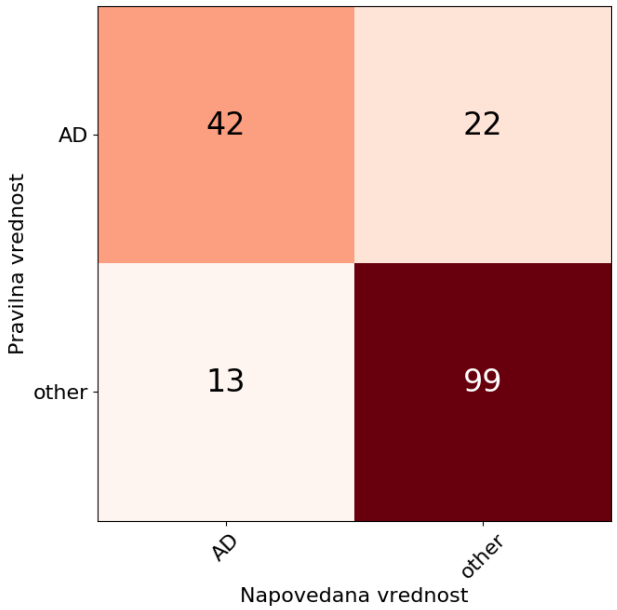


Slika 4.19: Matrika napak za ADNIDS

Vrednosti ciljnega razreda so na sliki razporejene glede na resnost diagnoze - od kognitivno normalen (CN) do diagnoze Alzheimerjeve bolezni

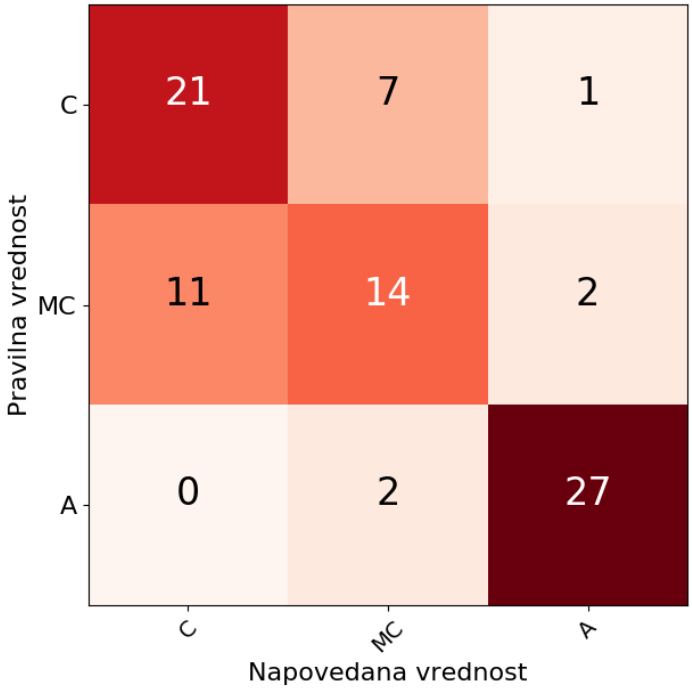
(AD). Manjše odstopanje pri številu primerov napačnih napovedi opazimo za vrednosti ciljnega razreda CN in AD, medtem ko pri vrednosti SMC in LMCI večje število primerov klasificiramo v napačen razred. Število napačno klasificiranih primerov je za vrednosti ciljnega razreda SMC in LMCI večje kot število pravilno klasificiranih primerov.

Matrika napak NMDS na sliki 4.20 nam pove, da približno $\frac{2}{3}$ primerov z Alzheimerjevo boleznijo napovemo pravilno.



Slika 4.20: Matrika napak za NMDS

Slika 4.21 prikazuje matriko napak ADDS kjer opazimo, da pogosto napačno določimo napovedano vrednost za kontrolno skupino in skupino primerov s kognitivnim poslabšanjem. Primere z Alzheimerjevo boleznijo napovemo z 93 % priklicem.



Slika 4.21: Matrika napak za ADDS

4.7 Rezultati PCA

Rezultati metode glavnih komponent so pokazali, da bi lahko z uporabo metode reducirali problemski prostor in tako hitreje pridobili napovedni model.

Tabela 4.12 prikazuje rezultate metode glavnih komponent z uporabo klasifikatorja naključnega gozda. Za vsako posamezno zbirko smo zabeležili najboljši rezultat točnosti metode. Opazimo lahko, da točnost napovednega modela upade le pri ADNIDS (za približno 10 %). Pri NMDS in ADDS točnost ostane skoraj enaka, kljub veliki redukciji problemskega prostora.

Podatkovna zbirka	**Število glavnih komponent**	**Točnost**	**Redukcija problemskega prostora [%]**
ADNIDS	25 od 42	60 %	$\approx$ 40 %
NMDS	37 od 120	80.1 %	$\approx$ 69 %
ADDS	14 od 143	71.7 %	$\approx$ 90 %

Tabela 4.12: Rezultati metode glavnih komponent posamezne podatkovne zbirke

Tako pri ADDS ohranimo približno enako točnost klasifikatorja pri zmanjšanju problemskega prostora za $\approx$ 90 % in pri NMDS za $\approx$ 69 %. Rezultati nam povedo, da za uspešno klasifikacijo z naključnim gozdom pri zadnjih dveh podatkovnih zbirkah potrebujemo zelo malo atributov.

Poglavje 5

Zaključek

Za podane podatkovne zbirke smo opravili analizo korelacij atributov in odkrili povezave med njimi. Povezave med atributi smo predstavili v obliki neusmerjenega grafa. Z uporabo metod strojnega učenja smo zgradili napovedne modele in jih ocenili. Uporabili smo metode nenadzorovanega in nadzorovanega strojnega učenja. Pri gradnji napovednih modelov smo upoštevali rezultate začetne analize podatkovnih zbirk. Izbor atributov in kategorije diagnoze so določale gradnjo napovednega modela. Z metodami redukcije problemskega prostora smo poskušali zmanjšati število potrebnih testov za diagnozo. Rezultate napovednih modelov smo predstavili s slikami, grafi in tabelami.

Analiza podatkovnih zbirk nam je razkrila, kateri izmed testov so najpomembnejši pri postavljanju diagnoze. Točnost zgrajenih napovednih modelov je pokazala, da lahko z več kot 70 % verjetnostjo pri vseh podatkovnih zbirkah napovemo, ali gre za primer Alzheimerjeve bolezni. Uporabljene metode za redukcijo problemskega prostora so pokazale, da za uspešno napoved ne potrebujemo vseh meritev testov.

Dobljeni rezultati bi lahko pripomogli k hitrejšemu postavljanju diagnoze Alzheimerjeve bolezni. Vrstni red preiskav bi lahko spremenili in s tem olajšali postopek diagnosticiranja bolezni za bolnike. Zgodnje odkrivanje bolezni bi omogočilo učinkovitejše zdravljenje in preprečevanje napredka bo-

lezni. Zdravniki iz Novega Sada so izrazili veliko zanimanje za optimizacijo vrstnega reda.

Literatura

[1] The prevalence of dementia in Europe, 2013. [Online; accessed 12-August-2017].

[2] Pandas: powerful Python data analysis toolkit, 2016. [Online; accessed 16-April-2017].

[3] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[4] Alzheimer's Disease International. World alzheimer report 2016, 2016. [Online; accessed 8-April-2017].

[5] Jessica Bean. *Rey Auditory Verbal Learning Test, Rey AVLT*, pages 2174–2175. Springer New York, New York, NY, 2011.

[6] Bolboaca, Sorana-Daniela and Jäntschi, Lorentz. Pearson versus Spearman, Kendall's tau correlation analysis on structure-activity relationships of biologic active compounds. *Leonardo Journal of Sciences*, 5(9):179–200, 2006.

[7] María Jesus Bullido, María Jesús Artiga, María Recuero, Isabel Sastre, Miguel Angel García, Jesús Aldudo, Corinne Lendon, Sang Woo Han, John C Morris, Anna Frank, et al. A polymorphism in the regulatory region of APOE associated with risk for Alzheimer's dementia. *Nature genetics*, 18(1):69–71, 1998.

[8] Alin Dobra. Decision Tree Classification. In *Encyclopedia of Database Systems*, pages 765–769. Springer, 2009.

[9] PM Doraiswamy, F Bieber, L Kaiser, KR Krishnan, J Reuning-Scherer, and B Gulanski. The Alzheimer's disease assessment scale Patterns and predictors of baseline cognitive performance in multicenter Alzheimer's disease trials. *Neurology*, 48(6):1511–1517, 1997.

[10] Eric Jones and Travis Oliphant and Pearu Peterson and others. SciPy: Open source scientific tools for Python, 2001. [Online; accessed 17-April-2017].

[11] ALZHEIMER EUROPE. Treatment of Alzheimer's disease, 2014. [Online; accessed 12-August-2017].

[12] Tal Galili, Alexis Mitelpunkt, Netta Shachar, Mira Marcus-Kalish, and Yoav Benjamini. Categorize, cluster, and classify: a 3-c strategy for scientific discovery in the medical informatics platform of the human brain project. In *International Conference on Discovery Science*, pages 73–86. Springer, 2014.

[13] Dragan Gamberger, Bernard Ženko, Alexis Mitelpunkt, Netta Shachar, and Nada Lavrač. Clusters of male and female Alzheimer's disease patients in the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. *Brain Informatics*, 3(3):169–179, 2016.

[14] Graphviz. Graph visualization software, 2017. [Online; accessed 26-August-2017].

[15] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.

[16] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[17] K Ito, MM Hutmacher, and BW Corrigan. Modeling of Functional Assessment Questionnaire (FAQ) as continuous bounded data from the ADNI database. *Journal of pharmacokinetics and pharmacodynamics*, 39(6):601–618, 2012.

[18] Novi Sad Klinika za nevrologijo in psihiatrijo, 2017. [Online; accessed 26-August-2017].

[19] Lenore Kurlowicz, Meredith Wallace, et al. The mini-mental state examination (MMSE). *Journal of gerontological nursing*, 25(5):8–9, 1999.

[20] H-J Möller and MB Graeber. The case described by Alois Alzheimer in 1911. *EuropeanAarchives of Psychiatry and Clinical Neuroscience*, 248(3):111–122, 1998.

[21] Fionn Murtagh and Pierre Legendre. Ward's Hierarchical Agglomerative Clustering Method: Which algorithms implement Ward's criterion? *Journal of Classification*, 31(3):274–295, 2014.

[22] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow. The Montreal Cognitive Assessment, MoCA: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

[23] National Institute on Aging. Alzheimer's Disease Fact Sheet, 2017. [Online; accessed 11-September-2017].

[24] Neo4j. Graph Visualization for Neo4j, 2017. [Online; accessed 13-August-2017].

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] GenePlanet personalizirana genetika. Alzheimerjeva bolezen, 2017. [Online; accessed 12-August-2017].

[27] Sandip Ray, Markus Britschgi, Charles Herbert, Yoshiko Takeda-Uchimura, Adam Boxer, Kaj Blennow, Leah F Friedman, Douglas R Galasko, Marek Jutel, Anna Karydas, et al. Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins. *Nature medicine*, 13(11):1359–1362, 2007.

[28] Spominčica Alzheimer Slovenija. Alzheimerjeva bolezen, 2017. [Online; accessed 12-August-2017].

[29] Tasha Smith, Nadia Gildeh, and Clive Holmes. The Montreal Cognitive Assessment: validity and utility in a memory clinic setting. *The Canadian Journal of Psychiatry*, 52(5):329–332, 2007.