math.e

Hrvatski matematički elektronički časopis

Predviđanje tipa osobnosti

1 Uvod

U današnje vrijeme, pogotovo zbog izazova u kojima se nalazimo, često je naglašeno pitanje unutarnjeg mira i pronalaska samog sebe. Pojam osobnosti usko je povezan s prethodnim pitanjem i problemom. Što je to osobnost i koja je njegova značajnost? Jedan od glavnih alata i pristupa u psihologiji je analiza osobnosti. Definicija osobnosti glasi: "ukupnost obilježja i ponašanja pojedinca kojima se razlikuje od ostalih pojedinaca". Poznavanje osobnosti može nam pomoći razumjeti bolje samog sebe i svoje postupke, također i psiholozima olakšava da nam bolje pristupe i da nas lakše usmjere. Međutim, psiholozi na razne načine interpretiraju osobnost i pokušavaju ljude klasificirati na temelju sličnosti u osobnostima.

Postoje razni pristupi klasifikaciji osobnosti i sastavljanju samog testa osobnosti. Jedan od trenutno najpoznatijih testova osobnosti je *Mayers - Briggs Type Indicator* koji osobnost dijeli u 16 kategorija — tipova, primjerice tip INTJ gdje svako slovo označava jednu od karakteristika te kategorije osobnosti, pogledati [2]. No, pristup osobnosti kojim ćemo se mi baviti bit će malo drugačiji, ali veoma često spominjan u literaturi. Detaljnije o njemu malo kasnije.

Već smo naveli značajnost otkrivanja karakteristika naše osobnosti, a to najbrže možemo rješavajući test osobnosti. Test koji ćemo mi koristiti kako bi ljude klasificirali u neku od kategorija iznijet ćemo u nastavku. Naš je zadatak u ovom radu predstaviti alat kojim na temelju odgovora na test možemo predvidjeti nečiju osobnost. Osim samog opisa testa i analize testnih podataka na kojima smo radili, prikazat ćemo pristup rješavanju zadatka klasifikacije i detaljno opisati metode koje smo pritom koristili. Također, demonstrirat ćemo predviđanje na već postojećim rezultatima testa.

2 Opis modela i testa

Jedan od često spominjanih modela ličnosti ili osobnosti je **Velikih pet tipova** (eng. *Big five*). Taj je model uveden 1980-tih godina i uvelike je olakšao psiholozima rad s ljudima. Prema tom modelu osobnost se svrstava u 5 kategorija:

(1) **Otvorenost** (eng.*Openness*)
(2) **Savjesnost** (eng. *Conscientiousness*)
(3) **Ekstrovertiranost** (eng. *Extraversion*)
(4) **Srdačnost** (eng. *Agreeableness*)
(5) **Neurotičnost** (eng. *Neuroticsm*)

Slika 1: Velikih pet tipova — OCEAN

Promatrajući prva slova engleskog nazivlja, ovaj je model često spominjan pod nazivom OCEAN model. Ukratko ćemo predstaviti svaku od kategorija (prema [7]) i navesti neka od njihovih glavnih obilježja jer će nam ona biti potrebna u samom testu. Osobe koje pripadaju kategoriji osobnosti **otvorenost** za sebe mogu reći da su, kao što i samo ime sugerira, otvoreni za nova iskustva, avanturistički orijentirani te imaju veliki raspon interesa. Upravo su zbog svoje otvorenosti prema novim iskustvima kreativniji, maštovitiji i znatiželjniji. Neke od najizraženijih karakteristika tipa osobnosti **savjesnost** su: samodisciplina, organiziranost i promišljenost. Osobe ove osobnosti obraćaju pažnju na detalje i vole svoje zadatke odraditi na vrijeme. Često ih se karakterizira i kao tvrdoglave i odlučne. Zbog svojih vještina planiranja i organizacije smatra ih se najboljim kandidatima pri odabiru za posao. Ekstroverti, ljudi tipa osobnosti **ekstrovertiranost**, veoma su društveni i vole provoditi vrijeme u društvu i upoznavati nove ljude. Njih veoma lako prepoznajemo u grupi ljudi jer su uvijek glasniji, energičniji i dominantniji. Njima nije problem snaći se u novom krugu ljudi, lako započinju razgovor i imaju velik krug ljudi koje poznaju. Predzadnji tip osobnosti, **srdačnost**, je rezerviran za ljude karakteristika poput ljubaznosti, povjerenja, empatije, pristojnosti, spremnosti za pomoć\ldots{} Ljudi tipa osobnosti srdačnost veoma se dobro slažu s ostalim pojedincima i brinu o tuđim potrebama. Često su članovi volonterskih društava ili su uključeni u razne akcije pomoći. Osobe koje pripadaju kategoriji osobnosti **neurotičnost** veoma loše podnose stres, često su tužni i lošeg raspoloženja. Razlog tome je emocionalna nestabilnost koja je njihova najznačajnija karakteristika. Česte promjene raspoloženja, nemogućnost kontroliranja emocija, anksiozni napadaji i razdražljivost posljedice su te nestabilnosti.

Analizirajući odgovore na testu osobnosti, psiholozi nastoje osobu svrstati u neku od kategorija osobnosti, uzimajući u obzir važnost svakog pojedinog pitanja i odgovor na njega. Mi ćemo analizu malo pojednostaviti budući da se ne bavimo stručnim psihološkim istraživačkim radom nego želimo znanja o umjetnoj inteligenciji demonstrirati na konkretnom primjeru u svakodnevnom životu. Željeli smo da test koji ćemo koristiti bude pristupačan za rješavanje, a da rezultati testa budu jednostavniji za analizu i korištenje. Odlučili smo se koristiti test baziran na obrascu pitanja sa Slike 2.

Test osobnosti

Koristeći kategorije modela osobnosti Big Five ocijenite sebe po kategorijama ocjenama od 1 do 8 na način
1 - ne primjećujem nikakve karakteristike tog tipa osobnosti na sebi,
8 - u potpunosti po navedenim karakteristikama pripadam tom tipu osobnosti

Spol:

- Ženski
- Muški

Godine:

Vaš odgovor

Otvorenost

1	2	3	4	5	6	7	8
○	○	○	○	○	○	○	○

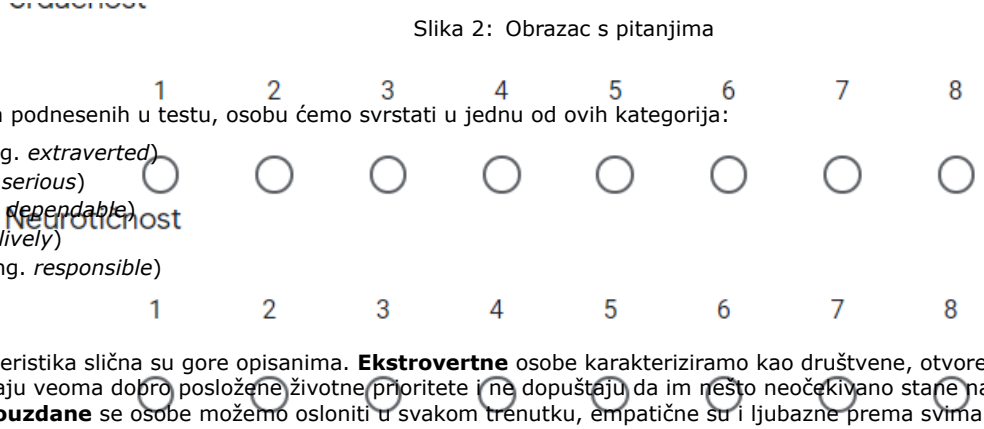
Savjesnost

1	2	3	4	5	6	7	8
○	○	○	○	○	○	○	○

Ekstrovertiranost

1	2	3	4	5	6	7	8
○	○	○	○	○	○	○	○

Srdačnost



Slika 2: Obrazac s pitanjima

Na temelju odgovora podnesenih u testu, osobu ćemo svrstati u jednu od ovih kategorija:

(1) **Ekstrovert** (eng. *extraverted*)
(2) **Ozbiljan** (eng. *serious*)
(3) **Pouzdan** (eng. *dependable*)
(4) **Živahan** (eng. *lively*)
(5) **Odgovoran** (eng. *responsible*)

Obilježja ovih karakteristika slična su gore opisanima. **Ekstrovertne** osobe karakteriziramo kao društvene, otvorene i primjetljive u društvu. Osobe koje su **ozbiljne** imaju veoma dobro posložene životne prioritete i ne dopuštaju da im nešto neočekivano stane na zacrtani put, čak i teže otkrivaju svoje emocije. Na **pouzdane** se osobe možemo osloniti u svakom trenutku, empatične su i ljubazne prema svima. **Živahne** osobe uvijek su spremne za avanturu, pune su energije i spremne za svaki izazov. Na kraju, **odgovorne** su osobe uvijek na vrijeme i svoje zadatke ne ostavljaju za zadnji tren. \newpage

3 Opis korištene baze podataka

Nakon što smo opisali model i test koji smo koristili pri implementaciji programa za predviđanje tipa osobnosti, preostalo nam je nešto reći i o korištenoj bazi podataka. Baza podataka koja odgovara svemu do sad opisanom je [6]. Konkretno, koristili smo datoteke naziva *train.csv* i *test.csv*, a njih smo spojili u jedan *DataFrame*. Razlog spajanja je taj da prilikom analize podataka vidimo svojstva svih podataka kojima raspolažemo. Podatke ćemo kasnije ponovno podijeliti za učenje i testiranje modela, ali o tome više u točki 5.

Skup podataka kojim raspolažemo se sastoji od 1024 retka i 8 stupaca, odnosno 1024 popunjenih testova osobnosti. U prvom stupcu nalazi se podatak o spolu (eng. *gender*) ispitanika s vrijednostima *Male* i *Female* za muškarce i žene, redom. U drugom stupcu se nalazi podatak o dobi, primijetili smo da se raspon godina kreće između 15 i 30 godina. Sljedeća četiri stupca sadrže podatke o odgovorima na pitanja o tipu osobnosti. Odgovor na pitanja je broj od 1 do 8, gdje 1 označava da ispitanik ne primjećuje nikakve karakteristike tog tipa osobnosti na sebi, a 8 da u potpunosti po navedenim karakteristikama pripada tom tipu osobnosti. Karakteristike u stupcima su redom: otvorenost (eng. *openness*), savjesnost (eng. *conscientiousness*), ekstrovertiranost (eng. *extraversion*), srdačnost (eng. *agreeableness*), neurotičnost (eng. *neuroticism*). U posljednjem stupcu je naziv jedne od pet osobnosti (s kraja točke 2) kojoj ispitanik pripada nakon rezultata testa. Prikazat ćemo nekoliko podataka kako bi pokazali izgled tablice.

	Gender	Age	openness	neuroticism	conscientiousness	agreeableness	\
0	Male	17	7	4	7	3	
1	Male	19	4	5	4	6	
2	Female	18	7	6	4	5	
3	Female	22	5	6	7	4	
4	Female	19	7	4	6	5	

	extraversion	Personality
0	2	extraverted
1	6	serious
2	5	dependable
3	3	extraverted
4	4	lively

4 Analiza podataka

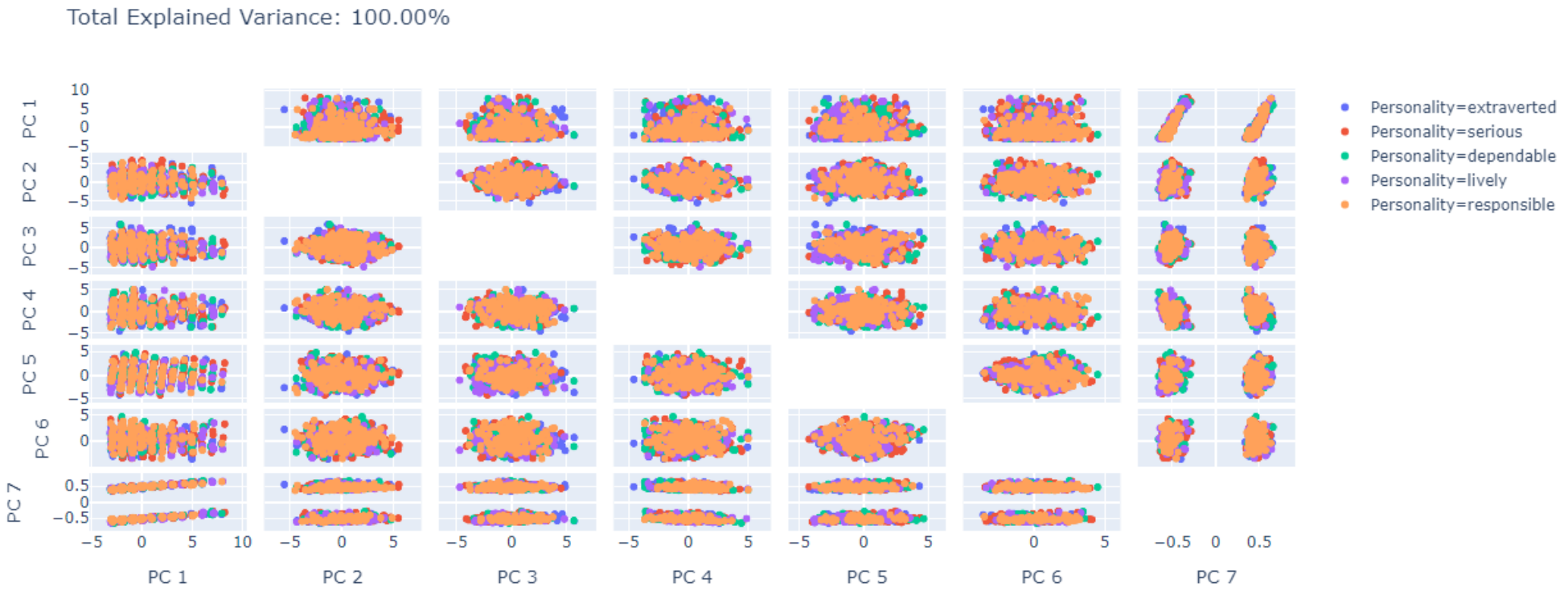
Sada kada znamo kojim podacima raspolažemo i kako izgledaju, pristupamo rješavanju zadatka. Prije samog pisanja koda i treniranja modela korisno je analizirati podatke kako bi stekli neki dojam o podacima s kojim radimo. Analizu podataka provodimo koristeći različite alate opisane u nastavku.

Prvo ćemo provjeriti jesu li svi podaci u tablici ispravni, odnosno ispravno zabilježeni/uneseni. Znamo da greške u podacima, poput *outliera*, mogu kasnije dosta utjecati na korektnost modela. Nepotpune ili pogrešne vrijednosti možemo, prema teoriji, tretirati na više načina. Redak s takvim podacima možemo samo odbaciti, možemo ih zamijeniti nulom ili nekom drugom konstantom (koja je prikladna s obzirom na ostale podatke) ili zamijeniti statističkom vrijednosti, poput srednje vrijednosti ili vrijednosti predviđene na temelju ostalih sličnih podataka.

Pronašli smo jedan test u kojem je oznaka spola označena nekorektno, taj smo dio popravili tako da smo ju zamijenili srednjom vrijednosti ostalih podataka u prvom stupcu. Također, dvije su ocjene greškom bile zapisane kao 9 pa smo ih prepravili na ocjenu 8 koja je maksimalna moguća ocjena. Za kraj, pronašli smo dvije godine upisane kao 5, što pretpostavljamo da je greška s obzirom na druge vrijednosti godina, njih smo zamijenili srednjom vrijednosti godina. Na ovaj smo način pokazali, u praksi, sve ranije navedene mogućnosti rješavanja problema nekorektnih podataka, bez odbacivanja podataka.

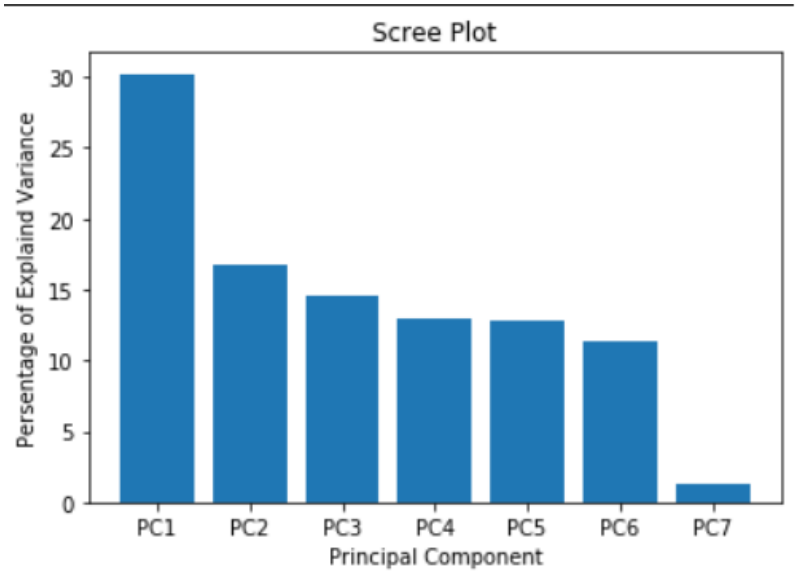
Analiza podataka često je povezana s analizom nekih statističkih elemenata. Koristeći statističke pojmove poput box-plotova, distribucije, koreliranosti ili srednje vrijednosti možemo doći do korisnih svojstava naših podataka. Povezano s time, korisno je takve elemente prikazati grafički, a mi smo se odlučili grafički prikazati analizu glavnih komponenti, primjer box-plotova i ispitati koreliranost pojedinih odgovora.

Prikupljene podatke želimo prikazati grafički, no kako imamo osam stupaca za prikaz bi nam trebao graf u osam dimenzija koji intuitivno nije jasan. Da bismo dobili predodžbu kako se naši podaci "ponašaju" koristimo tehniku zvanu **Analiza glavnih komponenti** (eng. *Principal component analysis*). Skraćeno PCA je tehnika smanjivanja dimenzije vektora kojeg proučavamo. Korištenjem PCA vektor dimenzije osam ćemo smanjiti na vektor u dvije dimenzije. Način, koji smo mi odabrali za jedan od prikaza, na koji smanjujemo dimenzionalnost podataka gubitkom minimalno informacija je uzimajući u obzir po dva stupca. Na taj način smanjujemo dimenzionalnost na neku koju možemo lakše grafički prikazati.



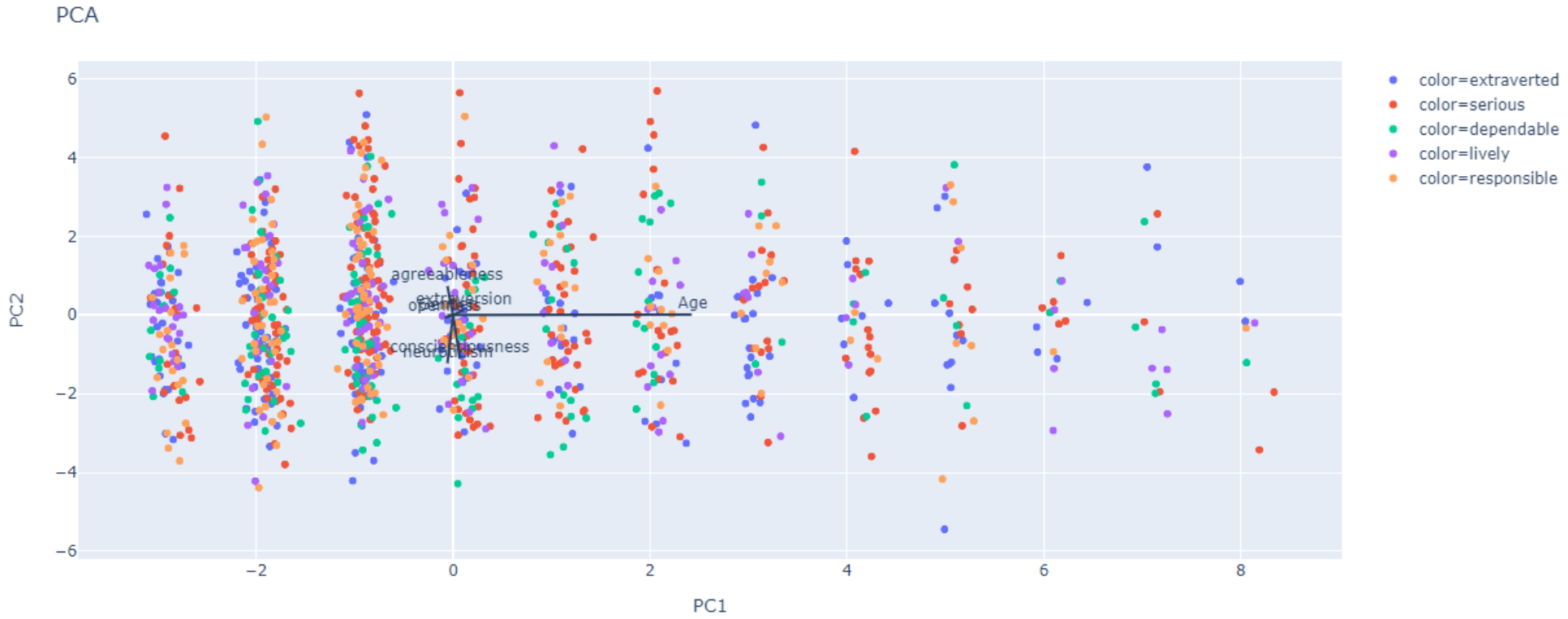
Slika 2: PCA s dva stupca

Na grafu 3 vidimo prikaz točaka u ravnini tako da gledamo svaku kombinaciju dvočlanog podskupa skupa stupaca, a legenda nam kaže koja boja točaka označava koji tip osobnosti. Analizom smo primijetili da nam niti jedan par pitanja nije dovoljan da grupiramo podatke. To vidimo iz činjenice da su točke raznih boja "nabacane" jedna preko druge i ne postoji krivulja koja ih odvaja. To nas je navelo da provjerimo koliko svako pitanje doprinosi u odabiru osobnosti.



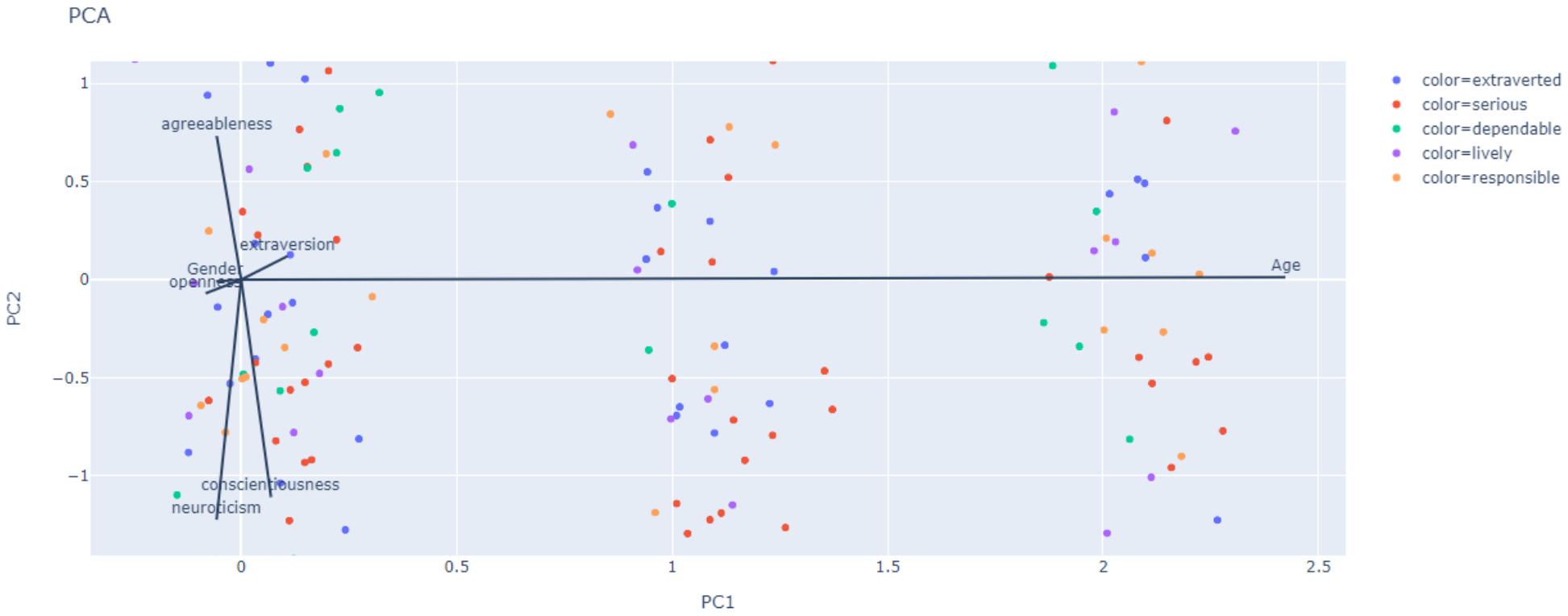
Slika 3: *Scree plot* i pripadne vrijednosti stupaca

Na *scree plotu* 3 vidimo da prvi stupac najviše određuje kojoj kategoriji podaci na kraju pripadaju, dok zadnji stupac najmanje. Razlog zašto u prvom grafu nismo imali "lijepo" grupiranje boja je taj što je postotak objašnjenja po komponentama malen, manji od 31%. Kad bismo imali situaciju u kojoj nam neka komponenta opisuje 80% ili više raznolikosti podataka tada bi grupiranje na prvom grafu bilo izražajnije. Sada možemo napraviti PCA na svih sedam stupaca istovremeno. To radimo tako da u osmerodimenzionalnom prostoru pronađemo dvodimenzionalnu ravninu od koje je udaljenost do svih točaka minimalna i napravimo projekciju na tu ravninu.



Slika 4: PCA na svih sedam stupaca istovremeno

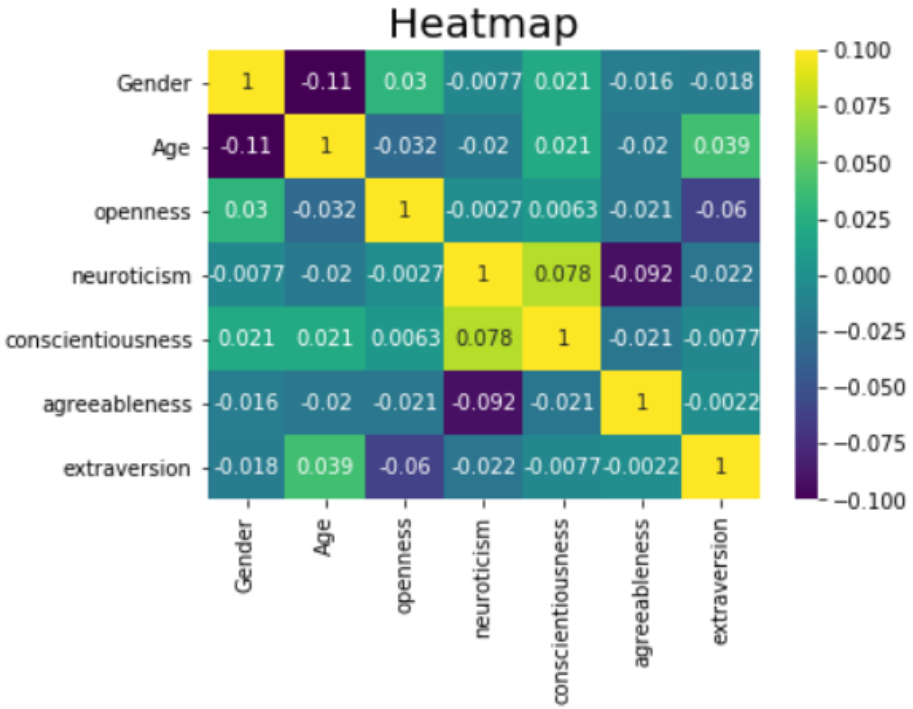
Kao što nam je prethodna analiza sugerirala nemamo vidljivo grupiranje boja. Dužine na grafu 4 prikazuju svojstvene vektore komponenti u PCA. Odnos između vektora nam govori o korelaciji između komponenti. Na primjer vidimo da su srdačnost (*agreeableness*) i neurotičnost (*neuroticism*) na suprotnim stranama pa je između njih negativna korelacija. Savjesnost (*conscientiousness*) i neurotičnost (*neuroticism*) su na istim stranama što nam govori da su pozitivno korelirane.



Slika 5: Zumirani prikaz grafa 4

Upravo je koreliranost važan statistički alat, koji bi u našem slučaju mogao biti od velike koristi. Iako već iz grafa 5 imamo uvid u koreliranost, htjeli smo testirati neke očite korelacije iz uvoda. Primjerice, ako detaljnije proučimo opise tipova osobnosti i krajnjih tipova osobnosti, zanimalo nas je koliko je ljudi koji su sebe ocijenili ocjenama 8 ili 7 za kategoriju *extraversion* u konačnici svrstani u klasu *extroverted*. Usporedbom broja ljudi koji su dali dvije najveće ocjene u toj kategoriji i broja ljudi koji su zapravo svrstani u kategoriju *extroverted* dobivamo jako mali broj, odnosno postotak. Napravili smo analognu provjeru za ostale osobnosti, ali niti jedna nije dala zanimljive rezultate, stoga ih ne navodimo.

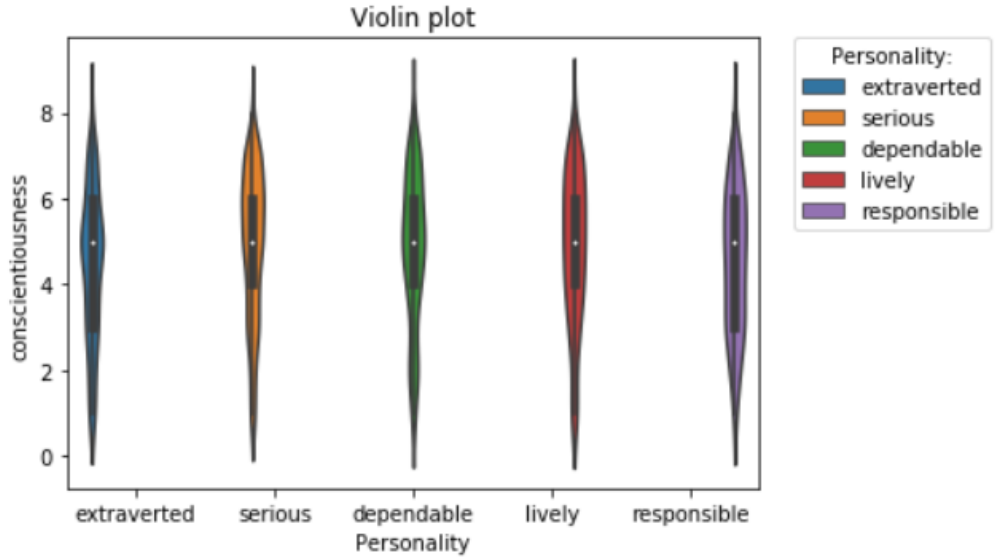
Za dodatnu analizu koreliranosti odlučili smo koristiti *heatmap* prikaz.



Slika 6: Heatmap

Prema grafu 6 zaključujemo da su korelacije kao što smo i najavili. No, svi koeficijenti koreliranosti su mali pa ne postoji linearna zavisnost među podacima. Do sada provedena analiza nam sugerira da jednostavniji modeli neće dobro opisivati naše podatke.

Jedno od svojstva podataka koje još nismo spomenuli je distribucija. **Distribucija** nam govori koja je vjerojatnost da će neka veličina poprimiti određenu vrijednost. Kako bi nam ta informacija mogla biti korisna prikazat ćemo distribuciju osobina za svaku kategoriju osobnosti. Za prikaz distribucija koristimo *violine plot*.



Slika 7: Distribucija savjesnosti (*conscientiousness*) za sve klase osobnosti

Na grafu 7 vidimo da sve osobnosti imaju sličnu distribuciju savjesnosti. Kod ekstrovertnih i odgovornih osoba češće se javljaju vrijednosti ispod sredine nego kod ostalih kategorija osobnosti. Distribucije ostalih osobine nećemo ovdje navoditi jer su razlike u distribuciji između osobnosti minimalne, stoga nam ne daju nikakve korisne informacije.

5 Predviđanje osobnosti

Slijedi glavni dio našeg rada, odnosno modeliranje i testiranje modela za predviđanje osobnosti. Veliki skup podataka koji smo do sada skupno analizirali potrebno je podijeliti i jedan dio koristiti za treniranje, dok drugi za testiranje. Omjer u kojem smo mi odlučili dijeliti podatke je 30% za testiranje i 70% za treniranje. Nakon podjele imat ćemo 717 redaka za treniranje modela i 307 za testiranje.

Prije modela i testiranja htjeli bismo ukratko skrenuti pažnju na još dva detalja. Prvi detalj vezan je uz distribuciji. Poželjno je da podaci za treniranje i testiranje imaju istu distribuciju. Modeli umjetne inteligencije bazirani su na pretpostavci da su podaci na kojima treniramo model i oni na kojima ćemo kasnije taj model testirati pa i koristiti jednako distribuirani, jer "vezu" koju pronađu na podacima za treniranje koriste i na onima koje predviđaju.

```
data.distribution()
```

```
Train distribution:
 serious         0.231520
extraverted      0.210600
dependable       0.193863
lively           0.186890
responsible      0.177127
```

Vidimo da distribucije nisu jednake. U podacima za testiranje broj osoba u klasi ozbiljnih (*serious*) je tri puta veća od broja osoba odgovornih (*responsible*), dok je u podacima za treniranje razlika mala. Ta činjenica je malo obeshrabrujući jer nam sugerira da niti jedan model kojeg treniramo na podacima za treniranje neće davati odlične rezultate na podacima za testiranje. Naime, ozbiljna narušenost pretpostavke rezultira slabom točnošću klasifikacije.

Drugi detalj tiče se karakteristika podataka, odnosno tipa kojem pripadaju. Kada ispišemo informacije o samoj bazi (iz točke 3) vidimo da se u njoj pojavljuju tipovi podataka *object* i *int64*.

```
data_train.info(verbose = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 709 entries, 0 to 708
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Gender            709 non-null    object
 1   Age               709 non-null    int64
 2   openness          709 non-null    int64
 3   neuroticism       709 non-null    int64
 4   conscientiousness 709 non-null    int64
 5   agreeableness     709 non-null    int64
 6   extraversion      709 non-null    int64
 7   Personality       709 non-null    object
dtypes: int64(6), object(2)
memory usage: 44.4+ KB
```

Kako neki modeli i funkcije na podacima rade preciznije ukoliko su podaci svi istog tipa ili samo numeričkog tipa odlučili smo sve podatke pretvoriti u tipove *int64*. Za tu smo potrebu vrijednosti *Female* i *Male* pretvorili redom u 0, 1, a kategorije osobnosti u brojeve od 1 do 5.

5.1 Pristup rješavanju

U ovoj bi točki kratko htjeli iznijeti teoriju koja stoji iza svih modela i koja prethodi odabiru svakog od njih. Priroda zadatka i podataka na kojima

radimo sugerira nam da ćemo naše modele bazirati na učenju s nadzorom. **Učenje s nadzorom** je vrsta strojnog učenja u kojem se svaki primjer sastoji od ulaznog podatka i željenog izlaza. Model na temelju ulaznih podataka donosi zaključke o izlaznim vrijednostima kako bi mogao dati odgovor na ulaz koji još nije vidio.

Naš zadatak predikcije osobnosti zapravo ima za cilj naučiti podatak svrstati u neku od klasa odnosno kategorija osobnosti. Bitna činjenica je da mi unaprijed znamo klase u koje moramo svrstati podatke.

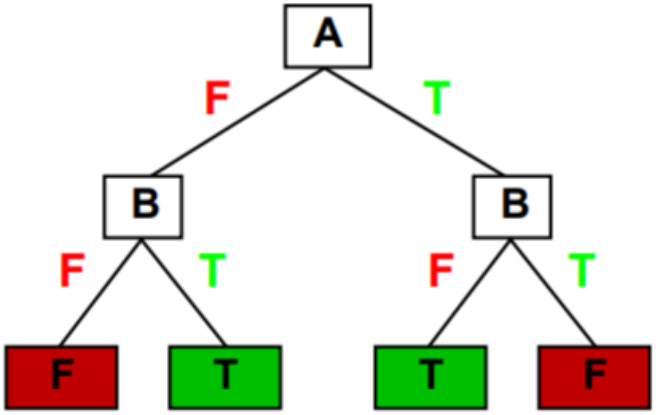
Kao što na početku poglavlja rekli, na podacima koji služe za treniranje implementirat ćemo model koji će ih *fitati* u najboljoj mjeri. Nakon toga ćemo taj model testirati na podacima za testiranje i odrediti točnost našeg modela. Ovo je samo jedan od načina validacije odnosno provjere točnosti konačnog modela. Za neke od modela demonstrirat ćemo **K-struku unakrsnu validaciju** (eng. *K-fold cross validation*), prema [9]. To znači da ćemo na slučajan način rasporediti podatke u K odvojenih skupova, u našem slučaju ćemo koristiti $K = 30$. Za svaki i od 1 do K, koristit ćemo i-ti od odvojenih skupova kao testni skup, a ostale ćemo spojiti u podatke za učenje i-tog modela. Za svaki model računat ćemo točnost predviđanja i na kraju prikazati prosječnu točnost i *box-plot* prikaz.

Isprobavali smo razne modele i mijenjajući njihove parametre pokušali smo dobiti što bolju točnost. Neke od njih ćemo prikazati u nastavku, a za detalje implementacije pogledajte [8]. Koristit ćemo tri pomoćne funkcije koje računaju točnosti, k-validaciju i traženje najboljih parametara. Svaki ćemo model, u zasebnoj točki, najprije teorijski opisati i iznijeti motivaciju za njegovo korištenje. Uz podatak točnosti predviđanja na podacima na kojima je trenirao, prikazat ćemo i predviđanje na podacima za testiranje i iznijeti točnosti tog predviđanja, koristeći usporedbu predviđenih vrijednosti s onim stvarnima.

5.2 Stablo odluke

S obzirom na prirodu zadatka i podatke na kojima radimo prvi model koji smo htjeli implementirati je stablo odluke. **Stablo odluke** je jedna od najčešće korištenih metoda induktivnog zaključivanja. To je skup više ako-onda pravila koje možemo prikazati grafički, u obliku stabla (na primjer Slika 8). U svakom listu stabla nalazi se klasa u koju treba svrstati ulazni podatak, dok su čvorovi uvjeti po kojima se ulazni podatak klasificira.

Važno je dobro odabrati atribute po kojima se stablo grana. Za grananje trenutnog stabla, među preostalim atributima, biramo onaj atribut koji tog trena daje najveći dobitak informacije, a to je onaj s manjom entropijom. **Entropija** je mjera nečistoće ili nereda u podacima. Pogledati [11] i [10] za više detalja o načinu računanja entropije i količine informacija. Stablu odluke možemo zadati i neka svojstva koje mora zadovoljavati, primjerice maksimalna dubina, i time utjecati na ponašanje modela.



Slika 8: Jednostavni grafički prikaz jednog stabla odluke, gdje su A i B atributi, T i F (true, false) njihove vrijednosti i konačne klase (iz [11])

U problemu kojim se bavimo, svaki list stabla je jedan tip osobnosti: *extraverted, serious, dependable, lively, responsible*. Stablo granamo u ovisnosti o vrijednostima stupaca: *gender, age, openness, neuroticism, conscientiousness, agreeableness, extraversion*. Naše stablo nema puno različitih atributa, no imat će dosta grananja za svaki atribut (osim spola) zbog raspona odgovora.

Za implementiranje stabla odluke koristili smo funkciju `sklearn.tree.DecisionTreeClassifier`(). Njome stvorimo model s unaprijed zadanim kriterijem *entropy*, čime označavamo da ćemo koristiti mjeru entropije u modelu. Ostale parametre spremimo u listu koju zatim prosljeđujemo funkciji `best_hyperparameters`() da nam vrati najbolje parametre za naše podatke. Nakon toga moramo *fitati* model na podacima za treniranje i radi predostrožnosti ispišemo točnost. Isto to napravimo i za test podatke. Mjera točnosti modela koju također koristimo je i funkcija `k_validation`() koja i grafički prikazuje točnost.

```
tree = DecisionTreeClassifier(criterion="entropy")
tree_param ={
   'splitter': ['best', 'random'],
   'max_features': [None, 'sqrt', 'log2'],
   'class_weight': [None, 'balanced'],
   #'min_samples_split' : np.arange(2,10,1),
   #'max_leaf_nodes' : np.arange(10000,1000000,1)
   #'min_samples_leaf' : np.arange(1,50,1)
}
tree = train.best_hyperparameters(tree, tree_param, train.X, train.y)
tree = tree.fit(train.X,train.y)
train.accuracy(tree,train.X,train.y)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight='balanced',
                       criterion='entropy', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, min_impurity_decrease=0.0,
                       min_impurity_split=None, min_samples_leaf=1,
                       min_samples_split=2, min_weight_fraction_leaf=0.0,
                       presort='deprecated', random_state=None,
                       splitter='random')
Accuracy: 99.58 %
```

Pomoću ispisa možemo vidjeti koliku je točnost model postigao na podacima za treniranje, kao i parametre koje je izabrao kao najbolje. Vidimo da je točnost na train podacima preko 99%, s čime smo bili veoma zadovoljni.

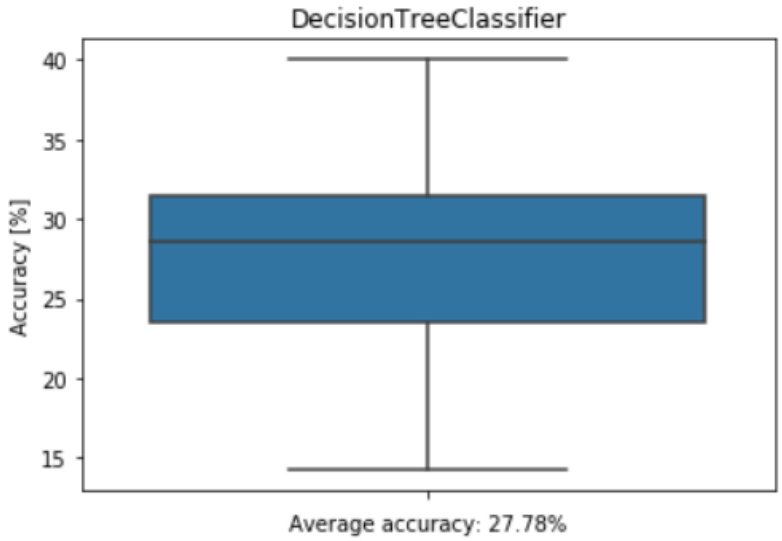
```
test.accuracy(tree,test.X_test,test.y_test)
```

```
Accuracy: 24.43 %
```

Nakon toga potrebno je koristiti model kako bi predvidjeli rezultate na test podacima. Na test podacima je točnost ispod 30% što nije puno, pogotovo ako uzmemo u obzir da bi nasumičnim pogađanjem točnost bila 20%. Razlog tako male točnosti je upravo razlika u distribuciji između podataka za treniranje i testiranje te slaba veza između podataka o kojoj smo govorili u točki 4. Kako bi se uvjerili da loša točnost nije ekstremni

slučaj napravit ćemo K-struku unakrsnu provjeru za $K = 30$.

```
test.k_validation(tree,test.data,30)
```



Slika 9: Prikaz rezultata K-unakrsne validacije za stablo odluke

Navest ćemo neke od tehnika kojima smo pokušali poboljšati rješenja. Prvi je promjena baze podataka na način da svaki stupac sadrži samo nule ili jedinice, čime povećavamo broj stupaca, a smanjujemo broj vrijednosti na 0 i 1. U nekim slučajevima ovaj način modificiranja podataka dovodi do poboljšanja rezultata, no u našem slučaju nije. Također, vizualizacijom stabla odluke možemo pažljivije odabrati parametre i time poboljšati točnost, iako je nismo prikazali u radu, obratili smo pažnju na njen izgled i time pokušali odabrati prikladne parametre za model.

Slika 10: Zapis vrijednosti u konfuzijskoj matrici iz [12]

Dodatna tehnika je proučavanje matrica zabune. **Matrica zabune** je matrica u kojoj su zapisane vrijednosti koje se računaju analizirajući stvarne i predviđene vrijednosti u modelu, a zapisane su redom kao na Slici 10. Matrica koju dobijemo u našem modelu je

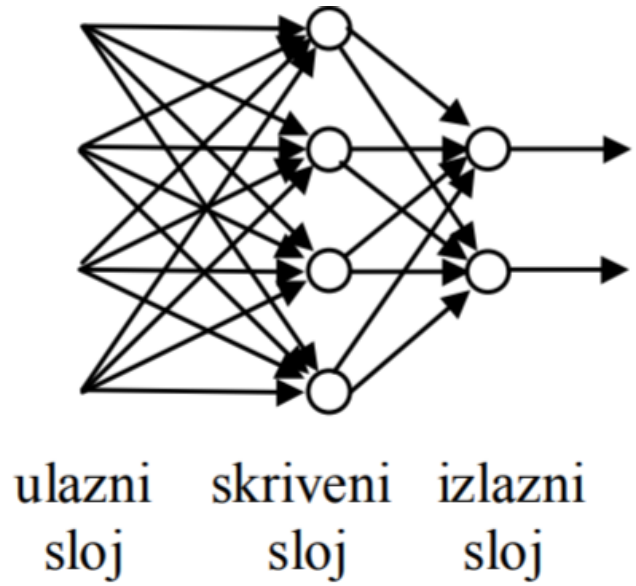
$$\begin{bmatrix} 23 & 12 & 14 & 14 & 14 \\ 34 & 32 & 33 & 29 & 25 \\ 4 & 7 & 5 & 2 & 3 \\ 3 & 8 & 4 & 4 & 5 \\ 5 & 8 & 7 & 8 & 12 \end{bmatrix},$$

a pomoću nje zaključujemo da su najčešće tipovi ekstrovert i ozbiljan bili zamjenjeni u predikciji, dok su najmanje zamjenjeni bili živahan i pouzdan.

5.3 Neuronske mreže

Sljedeći pristup koji smo odabrali za rješavanje problema bio je treniranje neuronske mreže. Razlog zbog kojeg smo se odlučili njih koristiti je njihova efikasnost pri rješavanju višedimenzionalnih problema, no one su zato i teoretski složenije i mogu dovesti do *overfittinga*. Također, važno svojstvo je i što podržavaju učenje s nadzorom i problem klasifikacije s kojim se mi susrećemo.

Neuronska mreža je računalni sustav koji oponaša živčani sustav ljudi i životinja. Izgrađena je od jednostavnih elemenata zvanih **neuroni**. Neuron je jednostavna funkcija f koja transformira određen broj ulaznih varijabli $x_1, x_2, \ldots$ s pripadnim težinama $w_1, w_2, \ldots$ u jednu izlaznu varijablu $y = f(x_1, x_2, \ldots)$, pojednostavljeno objašnjenje po [4]. Uz njih vežemo i nelinearnu funkciju aktiviranja, koja ograničava izlaz neurona na $[0, 1]$, za više detalja pogledati [5]. Važno svojstvo neuronske mreže je i njena arhitektura ili struktura. Osim jednoslojne arhitekture koja se sastoji od samo jednog sloja neurona — izlaznog sloja, mi ćemo koristiti višeslojnu arhitekturu kao sa Slike 11.



Slika 11: Višeslojna mreža s jednim skrivenim slojem sastavljenog od 4 neurona i izlaznim slojem s 2 neurona(iz [5])

Stvorili smo model koristeći funkciju `sklearn.neural_network.MLPClassifier()` koji ćemo *fitati* na podacima za treniranje. Dodatni parametri koje pozivamo u funkciji su povezani s karakteristikama neuronske mreže, pogledati u [5]. Parametre koje pozivamo u konačnom rješenju su oni koji su nam pri testiranju programa u konačnici pokazali najbolje rješenje. Funkcija aktivacije, koju smo spominjali kod neurona, u našem je slučaju logistička funkcija, oblika

$$f(x) = \frac{1}{1+e^{-x}}\,. \tag{1}$$

Struktura neuronske mreže koju mi koristimo opisana je kroz parametar `hidden_layer_sizes` i zaključujemo da se naša mreža sastoji od 3 skrivena sloja, svaki sadrži 600 neurona.

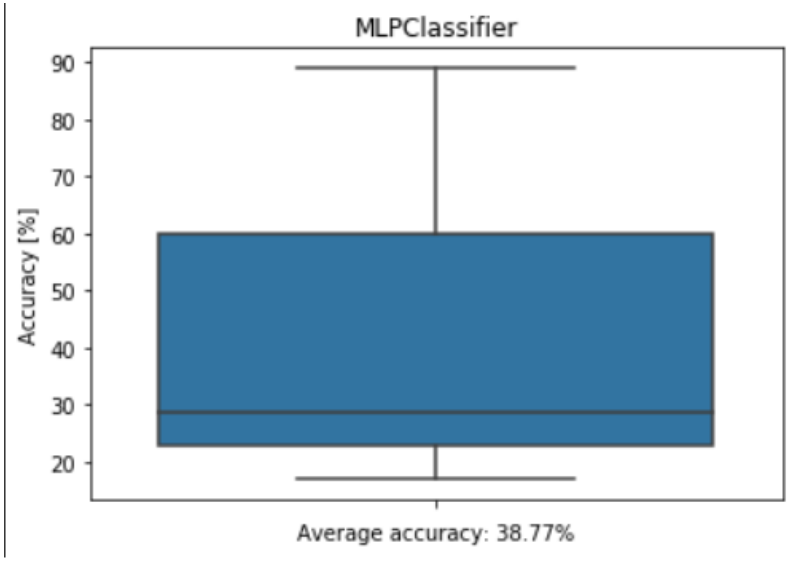
```
clf = MLPClassifier(activation='logistic', solver='lbfgs', max_iter=6000,
hidden_layer_sizes=(600,600,600), tol=1e-6 )
clf.fit(train.X, train.y)
train.accuracy(clf,train.X,train.y)
test.accuracy(clf,test.X_test,test.y_test)
test.k_validation(clf,test.data,30)
```

U ovome načinu pokretanja, koristeći najbolje parametre koje smo dobili pomoćnom funkcijom, dolazimo do točnosti na podacima za treniranje od 99%, dok je točnost na podacima za testiranje bila 26%.

Istu funkciju smo pokrenuli i sa standardnim (*defaultnim)* parametrima.

```
clf = MLPClassifier()
clf.fit(trsin.X, train.y)
train.accuracy(clf,train.X,train.y)
test.accuracy(clf,test.X_test,test.y_test)
test.k_validation(clf,test.data,30)
```

```
Accuracy: 31.1 %
Accuracy: 45.6 %
```



Slika 12: Prikaz rezultata K-unakrsne validacije za neuronsku mrežu

U ovoj situaciji je točnost na *train* podacima nešto niža, 32%, dok je točnost na *test* podacima viša od prethodne. Također, upotrijebili smo i `k_validation`() koji nam je pokazao da je ponašanje ovog modela prividno bolje od prošlog. Prvi poziv ovog modela nam daje lijepi primjer *overfittinga* na podacima. **Overfitting** je fenomen ili pojava kod kojeg model jako dobro opisuje *train* podatke, ali na *test* podacima daje loše rezultate.

5.4 Slučajne šume stabala odluka

Stablo odluke smo prethodno opisali u točki 5.2 i koristili u našem rješenju pa se slučajna šuma stabla odluke činila kao racionalan izbor za poboljšanje prethodnog rješenja. **Slučajna šuma stabala odluke**, kao što je intuitivno jasno, sastoji se od više stabala odluka nalik onima prethodno opisanim. Ideja je uzeti više stabala odluka koji uče iz slučajno odabranog podskupa skupa podataka i na kraju donijeti odluku na temelju svih stabala ([3]). Na taj način pronalazimo parametre koji su najbolji za naše podatke. Prednosti korištenja šume stabala su da češće daju bolje rezultate te da rjeđe vode do *overfitanja* podataka. Upravo iz navedenih razloga nadali smo se znatnijem poboljšanju rezultata korištenjem ove metode. Kako se već u samoj metodi podaci dijele na slučajan način i testiraju smatrali smo da provođenje K-unakrsne validacije u ovom slučaju nije potrebno.

```
data.random_forest()
```

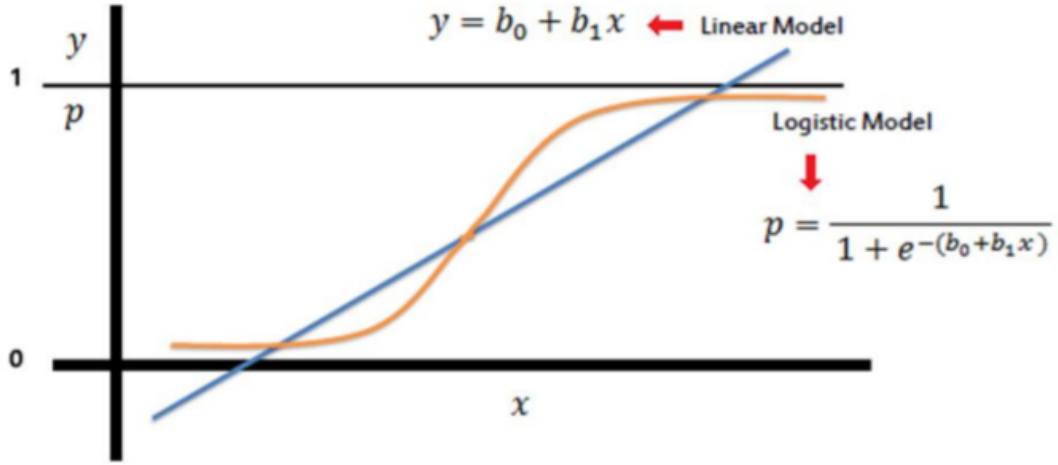
```
Accuracy: 57.0 %
```

U funkciji `random_forest()` koristili smo metodu `sklearn.ensemble.RandomForestClassifier`() koju smo pozvali s najboljim parametrima, čija su objašnjenja slična kao kod stabla odluke. Ispisana vrijednost je točnost na *test* podacima. Slučajne šume stabla odluke pokazale su znatno bolji rezultat od modela stabla odluke. Vrijeme izvršavanja `RandomForestClassifier`() znatno je dulje nego kod ostalih metoda jer se više puta radi stablo odluke s različitim parametrima i ulazim podacima.

5.5 Logistička regresija

Jedna od dodatnih metoda koju smo htjeli istražiti je i logistička regresija. Zbog loših rezultata predviđanja, razmišljali smo kako na neki drugačiji način poboljšati točnost. Razlog zbog kojeg nam se ova metoda činila zanimljivom je njezina povezanost s linearnom regresijom koja je svima dobro poznata.

Logistička regresija pripada familiji generaliziranih linearnih modela. Kako je teorija koja stoji iza generalizacije i samog modela zahtjevnija napomenut ćemo samo neka bitna svojstva (iz [1]). Jedno od prednosti koje ima ovaj pristup je što ne trebamo transformirati podatke kako bi varijable imale normalnu distribuciju. Također, parametri se drugačije procjenjuju, koristeći metodu maksimalne vjerodostojnosti umjesto klasične metode najmanjih kvadrata. Logistička regresija je model koji se koristi za predviđanje vjerojatnosti događaja pomoću prilagođavanja podataka logističkoj krivulji (1). Upravo je ta krivulja jedna od razlika u odnosu na pravac, na koji smo navikli kod linearne regresije, Slika 13. Dolazimo do važnog ograničenja, logistička regresija se najčešće koristi za probleme klasifikacija u dvije klase, odnosno zavisna varijabla je binarna.



Slika 13: Skica logističke i linearne krivulje

Međutim, postoje modifikacije modela u kojima je moguće koristiti logističku regresiju za klasificiranje u više klasa. Naravno, postoji više načina na koji se može napraviti modifikacija, no nećemo opisivati svaki od njih. Mi smo koristili multinomijalnu vjerojatnosnu distribuciju([13]). Model koji koristimo naziva se **Multinomijalna logistička regresija** koja u teoriji ima za zadatak učiti i predvidjeti multinomijalnu distribuciju.

Funkcija koju pozivamo je `sklearn.linear_model.LogisticRegression`() uz parametre koji su pronađeni kao najbolji i `multiclass='multinomial'`, koji je od velike važnosti jer imamo više klasa.

```
mul_lr = linear_model.LogisticRegression(multi_class='multinomial',max_iter
=10000)
lr_param ={
    #'C':np.arange(0.01, 1.01, 0.01),
    'solver' : ['newton-cg', 'lbfgs'],
    'tol' : np.arange(1e-6,1e-4,0.000001)
}
mul_lr.fit(train.X, train.y)
mul_lr = train.best_hyperparameters(mul_lr, lr_param, train.X, train.y)
train.accuracy(mul_lr, train.X, train.y)
test.accuracy(mul_lr, test.X_test, test.y_test)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=10000,
                   multi_class='multinomial', n_jobs=None, penalty='l2',
                   random_state=None, solver='newton-cg', tol=1e-06, verbose=0,
                   warm_start=False)
Accuracy: 24.83 %
Accuracy: 75.24 %
```

Iako je točnost na podacima za treniranje svega 25%, vidimo da je na test podacima točnost puno bolja i iznosi 75%. Ovaj je rezultat najbolji od svih modela koje smo koristili. Stoga, njime završavamo naše izlaganje o modelima umjetne inteligencije koje smo koristili u ovome radu.

6 Zaključak

Kao što smo rekli na početku, smatramo da je zadatak predikcije osobnosti dobar i koristan primjer korištenja umjetne inteligencije. Koristan je iz razloga što postoji potreba za analiziranjem rezultata testova osobnosti, ovog i ostalih koji postoje. Primjer je dobar jer smo na njemu mogli prikazati tijek jednog projekta čiji je zadatak predvidjeti konačni rezultat.

Na početku smo htjeli dati motivaciju i pregled u projekt kako bi bilo lakše razumjeti daljnji rad. Nadalje, smatramo da je analiziranje podataka i bavljenje s njim prije samog pisanja koda od velike važnosti, jer "priprema je pola posla". U tom dijelu rada mogu se otkriti zanimljive i korisne informacije koje kasnije mogu biti od velike koristi, kao i u našem slučaju. Analizom koreliranosti i prikazom distribucija, došli smo do zaključka da bi točnost modela koje ćemo raditi na našim podacima mogla biti loša. Tada je slijedilo pisanje koda i testiranje modela koje smatramo najvažnijima. Zbog lošijih rezultata predikcija koji su nas malo obeshrabrili, pokušali smo više pristupa kako bi probali doći do preciznijeg modela. Prikazali smo modele koje smatramo bitnima, uz detaljne rezultate testiranja. Kao što smo rekli najbolje nam se pokazao model logističke regresije.

Zaključno, smatramo da su podaci koje smo koristili zaslužni (odnosno "krivci") za loše rezultate predviđanja. S obzirom na ograničenu i malu količinu podataka u bazi, korišteni modeli nisu davali veliku točnost predviđanja na testnim podacima, ali demonstriraju njihovo korištenje na konkretnom primjeru i važnost poznavanja svojstva podataka prije modeliranja.

7Literatura

[1] I. Bistrovi´c. Logistička regresija u analizi smrtnosti. Zadnje pristupljeno: prosinac 2021. URL:

https://repozitorij.pmf.unizg.hr/islandora/object/pmf%3A5812/datastream/
PDF/view.
[2] D. H. Saklofske C. Coulacoglou. Psychometrics and Psychological Assessment. 2017.
[3] M. Cular. Modeli slučajnih šuma i primjene. Zadnje pristupljeno: prosinac 2021. URL: https://repozitorij.pmf.unizg.hr/islandora/object
/pmf%3A9099/datastream/....
[4] B. Gavranovic. Primjena modela dubokog uˇcenja na analizu sentimenata.
[5] S. Lončarič. Neuronske mreže: Uvod. Zadnje pristupljeno: prosinac 2021. URL: https://www.fer.unizg.hr/_download/repository/01-Uvod-1s.pdf.
[6] Kaggle baza podataka. Zadnje pristupljeno: prosinac 2021. URL: https://www.kaggle.com/
pavlorymarchuk/test3434.
[7] O. P. John R. R. McCrae. An Introduction to the Five-Factor Model and Its Applications. Zadnje pristupljeno: prosinac 2021. URL:
https://www.workplacebullying.org/multi/pdf/
5factor-theory.pdf.
[8] Git repozitorij. URL: https://github.com/Krcivoj/Personality-prediction.
[9] S. Singer. Provjera (validacija) modela. Zadnje pristupljeno: prosinac 2021. URL: http://degiorgi.math.hr/~singer/ui/ui_1415/ch_18c.pdf
[10] S. Singer. Učenje na primjerima. Zadnje pristupljeno: prosinac 2021. URL: http://degiorgi.math.hr/~singer/ui/ui_1415/ch_18b.pdf.
[11] S. Singer. Učenje na primjerima(promatranjem). Zadnje pristupljeno prosinac 2021. URL: http://degiorgi.math.hr/~singer/ui/ui_1415
/ch_18.pdf.
[12] J. Šnajder. Strojno učenje. Zadnje pristupljeno: prosinac 2021. URL: https://www.fer.unizg.hr/_download/repository/SU-2015-
Vrednovanje_modela....
[13] J. Šnajder. Strojno učenje: 7. Logistička regresija II. Zadnje pristupljeno: prosinac 2021.
URL: https://www.fer.unizg.hr/_download/repository/SU-2019-07-LogistickaRegre....

ISSN 1334-6083