Modeli klasifikacije u alatu Scikit-Learn Python

Blažević, Tamara

Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski studij informatike – jednopredmetni

Tamara Blažević

Modeli klasifikacije u alatu Scikit-Learn Python

Završni rad

Mentor: Dr. sc. Lucia Načinović Prskalo

Rijeka, rujan 2019.

Zadatak završnog rada

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 15.3.2018.

Zadatak za završni rad

Pristupnik: Tamara Blažević

Naziv završnog rada: Modeli klasifikacije u alatu Scikit-Learn Python

Naziv završnog rada na eng. jeziku: Classification models in Scikit-Learn Python tool

Sadržaj zadatka: Zadatak završnog rada je pojasniti postupak klasifikacije u strojnom učenju, pojasniti osnovne pojmove koji se pojavljuju prilikom primjene postupka klasifikacije, zatim opisati i isprobati neke od najznačajnijih algoritama za klasifikaciju u alatu Scikit-Python kao što su primjerice Naive Bayes, stabla odlučivanja, neuronske mreže, SVM i slično te pokazati njihov rad na primjerima.

Mentor

Dr. sc. Lucia Načinović Prskalo

Voditelj za završne radove

Dr. sc. Miran Pobar

Zadatak preuzet: 22.3.2018.

Tamara Blažević

(potpis pristupnika)

Sažetak

U ovom završnom radu glavni je cilj opisati i pojasniti postupak klasifikacije u strojnom učenju. Opisan je pojam klasifikacije te dana predodžba o tome kako funkcionira i gdje se koristi. Navedeni su i detaljno objašnjeni neki od značajnijih algoritama za klasifikaciju od kojih je jedan kasnije korišten na problemu razrješavanja višeznačnosti u hrvatskom jeziku. Također, ukratko je opisan naglasni sustav u hrvatskom jeziku i problem višeznačnosti riječi. Prikazana je primjena jednog od algoritama klasifikacije na zadanom problemu te je opisan način na koji je primijenjen algoritam i kako se došlo do dobivenog rezultata. Alat u kojem je prikazan rad algoritma je Scikit-Learn Python. Na posljetku je prokomentiran dobiveni rezultat za zadani problem te su dane zaključne misli vezane uz pojam klasifikacije.

Ključne riječi: strojno učenje, klasifikacija, modeli klasifikacije, Scikit-Learn, algoritmi za klasifikaciju, Naive Bayes, stabla odlučivanja (Decision Trees), Support Vector Machines, problem višeznačnosti, homonimi.

Sadržaj

1. Uvod

Strojno učenje jedno je od danas najaktivnijih i najuzbudljivijih područja istraživanja u računalnoj znanosti, umjetnoj inteligenciji i statistici, ponajviše zbog brojnih mogućnosti primjene [1]. Kod strojnog učenja, fokus je osposobiti algoritme za učenje obrazaca i predviđanje podataka kako bi nam omogućili korištenje računala za automatizaciju procesa donošenja odluka, čime uvelike olakšavaju rad u određenim slučajevima. Strojno učenje se danas vrlo aktivno koristi u mnogim područjima kao što su primjerice računalna obrada jezika, računalni vid, raspoznavanje i sintezu govora i slično.

Jedna od vrsti strojnog učenja je i nadzirano učenje u koje spada klasifikacija. Nadzirano učenje znači da se model uči na temelju poznatih podataka. Za klasifikaciju možemo reći da je to podjela ili razvrstavanje različitih stvari, pojmova i sl. prema određenim kriterijima koje sami biramo. Klasifikacija je kao takva sveprisutna i susrećemo se sa njom u raznim područjima života. Možemo čak reći da je sve što nas okružuje moguće klasificirati po nekom obilježju.

Kod klasifikacije u strojnom učenju koriste se razni modeli klasifikacije, ovisno o tome koji je najprikladniji za zadani problem. Neki od algoritama za klasifikaciju su Naive Bayes, stabla odlučivanja (Decision Trees), Support Vector Machines, Neural networks, Nearest Neighbors, Gaussian Processes…

U sljedećem poglavlju opisan je općenit pojam klasifikacije, princip kako funkcionira, za što se koristi te su navedeni neki od primjera gdje se ona koristi.

U trećem poglavlju navedeni su neki od značajnijih algoritama za klasifikaciju te su isti detaljno opisani. Jedan od njih kasnije je korišten u primjeru razrješavanja problema višeznačnosti.

U četvrtom poglavlju ukratko je opisan naglasni sustav u hrvatskom jeziku i problem višeznačnosti, a naročito je naglasak na tip višeznačnosti kada višeznačne riječi imaju isti naglasak i oblik, a različito značenje.

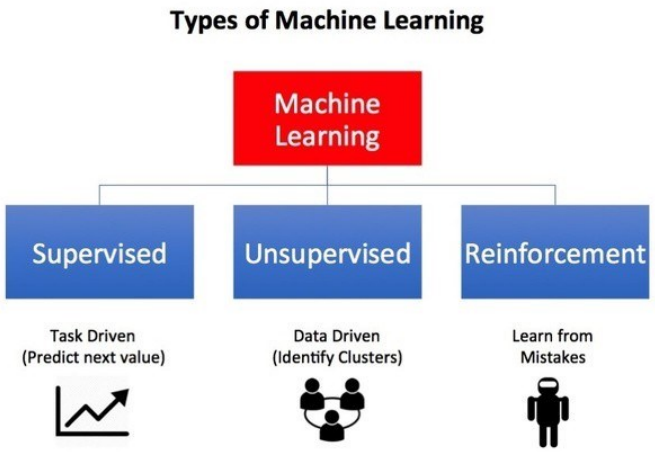
U petom poglavlju prikazana je primjena algoritma klasifikacije na problemu razrješavanja višeznačnosti, opisan je način na koji je algoritam primijenjen te što je bilo potrebno napraviti i prilagoditi kako bi isti funkcionirao na zadanim rečenicama.

U šestom poglavlju prikazan je rezultat korištenog algoritma te su isti ukratko prokomentirani.

Na posljetku, kao zadnje poglavlje ovog rada iznesen je zaključak vezan uz primjer i rezultat iz petog poglavlja. Dane su zaključne misli za sam pojam klasifikacije i njeno korištenje.

2. Klasifikacija kao metoda strojnog učenja

Strojno učenje (Machine Learning) je znanstveno proučavanje algoritama i statističkih modela koje računalni sustavi koriste za izvođenje određenog zadatka bez izričitih uputa, oslanjajući se na različite obrasce i zaključke. Na njega se gleda kao na podskup umjetne inteligencije. Algoritmi strojnog učenja grade matematički model koji se temelji na uzorcima podataka, poznatijima pod nazivom „skup za učenje“ („training data“), s ciljem predviđanja ili donošenja odluka bez izričitog programiranja za obavljanje zadatka [2]. Algoritmi strojnog učenja koriste se u mnogim aplikacijama, poput filtriranje e-pošte i računalnog vida gdje je teško i neizvedivo razviti konvencionalni algoritam koji bi učinkovito izvršavao zadatak. Strojno je učenje usko povezano sa računalnom statistikom koja se usredotočuje na predviđanje pomoću kompjutora. Strojno učenje se dijeli na nadzirano, nenadzirano i nagrađivano.

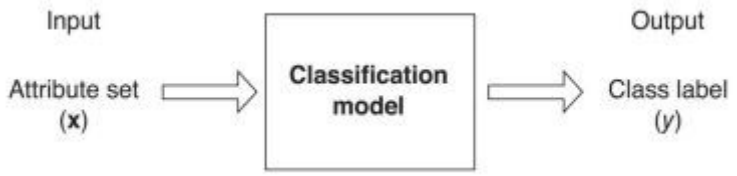


Slika 1: Vrste strojnog učenja[1]

U ovom radu bavit ćemo se nadziranim učenjem u koje spada klasifikacija. Glavni cilj nadziranog učenja je naučiti model iz označenih podataka koji nam omogućuju da predvidimo neviđene ili buduće podatke [3]. Izraz „nadzirani“ odnosi se na skup uzoraka gdje su željene izlazne oznake već poznate [3].

Klasifikacija je potkategorija nadziranog učenja čiji je cilj predvidjeti kategoričke oznake novih istanci na temelju prošlih opažanja. Klasifikacija je zadatak učenja ciljne funkcije f koja mapira svaki skup atributa x na jednu od unaprijed definiranih oznaka klase y. [4]

[1] Izvor: https://miro.medium.com/max/602/0*-068ud_-o3ajwq_z.jpg



Slika 2: Klasifikacija kao zadatak mapiranja ulaznog skupa atributa x u klasnu oznaku y[2]

Ulazni podaci za klasifikaciju su skup zapisa, odnosno podataka. Svaki zapis (podatak) je oblika (x, y), gdje je x ulaz, a y izlaz označen kao klasna oznaka. Ciljna funkcija je također neformalno poznata kao model klasifikacije koji se isto tako može koristiti za predviđanje oznaka klase nepoznatih zapisa.

Klasifikacijska tehnika ili klasifikator je sustavni pristup klasifikacijskim modelima gradnje iz ulaznog seta podataka. Klasifikacijske tehnike najprikladnije su za predviđanje ili opisivanje skupova podataka sa binarnim ili nominalnim kategorijama. Binarna kategorija podrazumijevaju podatke čije su vrijednosti 0 ili 1. Takve vrste klasifikatora se još nazivaju i binarni klasifikatori. Klasifikacijske tehnike su manje učinkovite za redne kategorije jer ne uzimaju u obzir implicitni poredak među kategorijama. Svaka tehnika koristi algoritam učenja za prepoznavanje modela koji najbolje odgovara odnosu između postavljenog atributa i klasnih oznaka ulaznih podataka. Prvo se mora osigurati skup za učenje koji se sastoji od zapisa čije su klasne oznake poznate. Skup za učenje koristi se za izgradnju klasifikacijskog modela koji se kasnije primjenjuje na skupu za testiranje koji se sastoji od zapisa s nepoznatim klasnim oznakama. Evaluacija performansi klasifikacijskom modela temelji se na brojanju testnih zapisa koje je model ispravno i netočno predvidio. Ta su brojanja zapisana u tabeli koja se naziva „confusion matrix“. Većina algoritama za klasifikaciju traži modele koji postižu najveću točnost ili najnižu stopu pogreške kod primjene skupa za učenje.

Klasifikatori su se pokazali dobri u mnogim stvarima. Najčešće su korišteni u klasifikaciji dokumenata i filtriranju neželjene pošte na e-mailu. Koriste se i u oglašavanju; mnogi oglasi koje vidimo dok pregledavamo Internet postavljeni su tamo jer je algoritam učenja rekao da su razumne popularnosti. Koriste ih primjerice Netflix i Amazon kod preporuka za neki proizvod, banke za otkrivanje prevara kod kartičnih transakcija, kod dokumenata za prepoznavanje autora (prepoznaje se po stilu pisanja). Koriste se kod recenzija za filmove – jesu li pozitivne ili negativne. Facebook ih koristi za prepoznavanje lica – sustav koji slika, pronalazi lica i otkriva tko je na fotografiji (sugerirajući oznaku). Također, zdravstvene tvrtke

[2] Izvor: https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf

počinju sve više koristiti strojno učenje za praćenje, procjenu i dijagnosticiranje stanja pacijenata.

Kao što vidimo, klasifikacija se primjenjuje u različitim područjima, od velike je pomoći te vrlo dobro funkcionira.

3. Modeli klasifikacije

Vrste algoritama strojnog učenja razlikuju se u pristupu, vrsti podataka koji unose i iznose te vrsti zadatka ili problema koji namjeravaju riješiti. Trenutno je na raspolaganju mnogo algoritama za klasifikaciju, međutim nemoguće je zaključiti koji je bolji od drugoga. To ovisi o prirodi i primjeni dostupnih skupova podataka. Ovisno o problemu, odabire se algoritam koji je najprikladniji za njegovo rješavanje.

U ovom poglavlju navedeni su i opisani neki od značajnijih modela klasifikacije od kojih je jedan kasnije korišten pri rješavanju problema višeznačnosti u hrvatskom jeziku.

3.1. Naive Bayes

Naive Bayes metode čine skup algoritma nadziranog učenja koji se temelji na primjeni Bayesovog teorema s „naivnom" pretpostavkom neovisnosti između svakog para značajki [5]. Oslanja se na vrlo jednostavno predstavljanje dokumenata. Ovaj pristup klasificira tekstualne dokumente koristeći dva parametra: uvjet vjerojatnosti svakog značenja (Si) riječi (w) i značajke (fj) u kontekstu [6]. Maksimalna vrijednost procjene iz formule predstavlja najprikladnije značenje u kontekstu. Ovdje je broj značajki predstavljen sa m. Vjerojatnost P izračunava se iz učestalosti zajedničke pojave skupu za učenje, a P(Si | fj) se izračunava iz obilježja u prisutnosti.

$$\hat{S} = \underset{S_i \in Sense_D(w)}{\operatorname{argmax}} P(S_i \mid f_1, \dots, f_m) = \underset{S_i \in Senses_D(w)}{\operatorname{argmax}} \frac{P(f_1, \dots, f_m | S_i) P(S_i)}{P(f_1, \dots, f_m)}$$

$$= \underset{S_i \in Senses_D(w)}{argmax} P(S_i) \prod_{j=1}^{m} P(f_j | S_i)$$

Slika 3: Naive Bayes-ova formula

Cilj bilo kojeg probabilsitičkog klasifikatora je, sa značajkama f_0, …, f_n i klasama S_0, …, S_k odrediti vjerojatnost pojavljivanja značajki u svakoj klasi i vratiti najvjerojatniju klasu. Stoga za svaku klasu želimo izračunati P(Si | f_0, … , f_n). Izračunavanje pojedinih P(Si | fj) uvjeta ovisit će o distribuciji koje naše značajke slijede. U kontekstu klasifikacije teksta, gdje značajke mogu

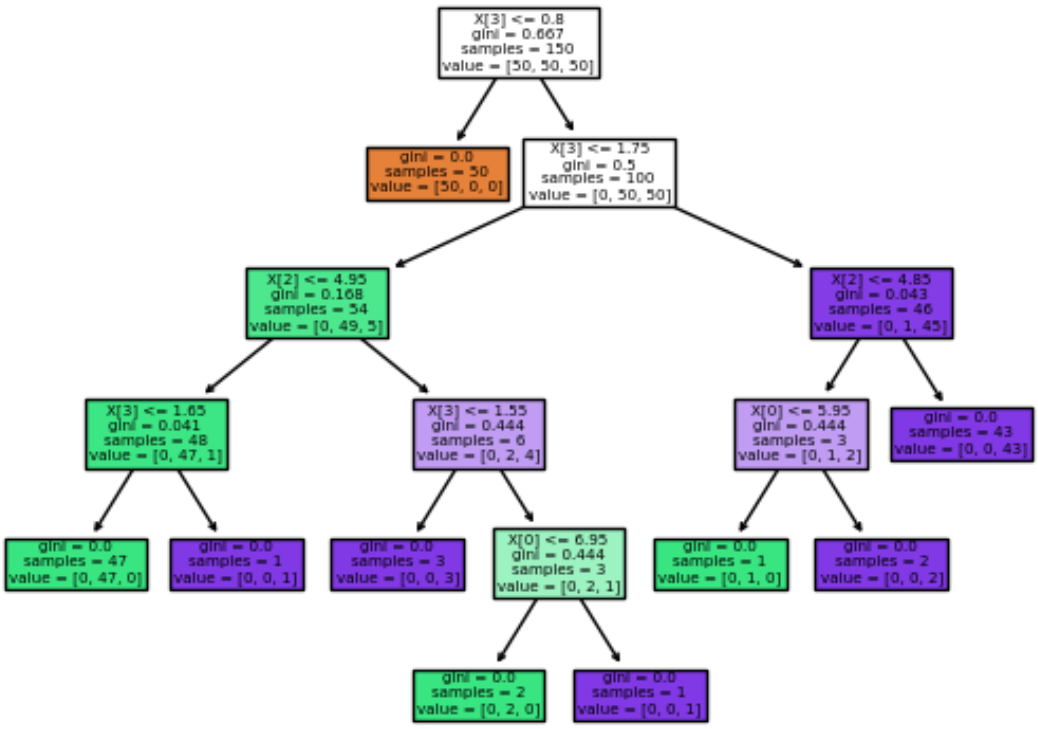
biti npr. brojanje riječi, značajke mogu slijediti multinomnu distribuciju, dok se u drugim slučajevima, u kojima su značajke neprekinute, one mogu slijediti Gaussovu distribuciju. Jedini posao koji je potrebno napraviti prije predviđanja je pronalaženje parametara za značajke pojedinačne distribucije vjerojatnosti. To znači da Naive Bayes-ovi klasifikatori mogu biti uspješni čak i sa velikim brojem podataka.

Naive Bayes klasifikatori pokazali su vrlo dobrima u stvarnim situacijama, kao što je klasifikacija dokumenata i filtriranje neželjene pošte. Zahtijevaju malu količinu podataka u skupu za učenje za procjenu potrebnih parametara. Vrlo je jednostavan algoritam za implementaciju i dobri rezultati su postignuti u većini slučajeva. Klasifikatori Naive Bayesa mogu biti iznimno brzi u usporedbi sa sofisticiranijim metodama. Odvajanje raspodjele klasnih uvjetnih značajki znači da se svaka distribucija može samostalno procijeniti kao jednodimenzionalna distribucija.

Različiti Naive Bayes klasifikatori razlikuju se uglavnom zbog pretpostavki koje oni čine u vezi distribucije P(Si | fj). Neke od vrsta klasifikatora su Gaussov Naive Bayes, multinomni Naive Bayes, Bernoullijev Naive Bayes.

3.2. Stabla odlučivanja (Decision Trees)

Stabla odlučivanja predstavljaju neparametarski nadzirni način učenja koji se koristi za klasifikaciju i regresiju [7]. Cilj je stvoriti model koji predviđa vrijednost ciljane varijable učenjem jednostavnih pravila odlučivanja koja su izvedena iz podatkovnih značajki. Koriste se za označavanje pravila klasifikacije u strukturi stabla koja rekurzivno dijeli skup podataka za učenje [6]. Unutarnji čvor stabla označava test koji će biti primijenjen na trenutnu vrijednost, a svaka grana označava izlaz testa. Kada se dostigne čvor lista, predstavljen je smisao riječi.



Slika 4: Primjer stabla odlučivanja[3]

Stabla odlučivanja koriste skup pravila if-then koji je međusobno isključiv i iscrpan za klasifikaciju. Pravila se uče uzastopno koristeći podatke za učenje, jedno po jedno. Svaki put kada se neko pravilo nauči, uklanjaju se n-torke koje su obuhvaćene pravilima. Taj se postupak nastavlja na skupu za učenje sve dok se ne ispuni uvjet prekida. Stablo je izgrađeno rekurzivno na način dijeljenja i osvajanja odozgo prema dolje. Svi atributi trebaju biti kategorični. Atributi u vrhu stabla imaju više utjecaja na klasifikaciju i oni se identificiraju korištenjem koncepata dobivanja informacija. Podjela se vrši kako bi se smanjila stopa pogreške u klasifikaciji. Stopa pogreške u klasifikaciji je udio promatranja treninga u regiji koja ne pripada najčešćem razredu. Međutim, u praksi se najčešće koriste druge dvije metode a to su Gini indeks i cross-entropy.

Kao i kod ostalih klasifikatora kao ulaz se uzimaju dva polja: niz X, veličine *[n_samples, n_features]* koje sadrže uzorke za učenje i polje Y od cjelobrojnih vrijednosti, veličine *[n_samples]* koje sadrže oznake za klase učenja. Nakon postavljanja tih polja, model se može koristiti za predviđanje klase uzoraka. Klasifikator stabla odlučivanja sposoban je i za binarnu klasifikaciju i za klasifikaciju multiklasa.

Stablo se lako može prekomjerno natrpati, stvaranjem previše grana i može odražavati anomalije uslijed buke i odljeva. Prekomjerno natrpan model ima vrlo slabe performanse na neviđenim podacima iako daje impresivne performanse na podacima za učenje. To se može

[3] Izvor: https://scikit-learn.org/stable/_images/sphx_glr_plot_iris_dtc_0021.png

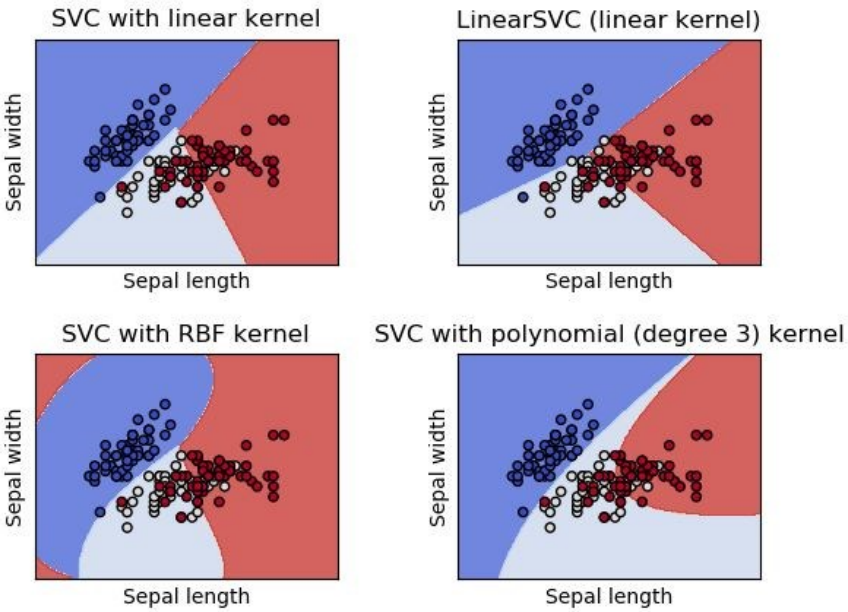
izbjeći prethodnim obrezivanjem, što zaustavlja ranu izgradnju stabla ili naknadnim obrezivanjem, što uklanja grane potpuno izgrađenog stabla.

3.3. Support Vector Machines (SVM)

Support Vector Machines je skup nadziranih metoda učenja koje se koriste za klasifkaciju, regresiju i vanjsko otkrivanje (outlier detection) [8]. Cilj ove metode je odvajanje pozitivnih primjera od negativnih primjera s maksimalnom granicom, a granica je udaljenost od hiperravnine do najbližeg pozitivnog ili negativnog primjera, tj. cilj je pronaći hiperravninu u n-dimenzionalnom prostoru koji jasno razdvaja podatkovne točke. Ti primjeri nazivaju se potporni vektori (support vectors) [6]. Ovi algoritmi koriste teoriju minimalizacije strukturnog rizika.

Algoritmi bazirani na SVM upotrebljavaju se za klasifikaciju nekoliko primjera u dvije različite klase [9]. Ovaj algoritam nalazi hiperravninu između te dvije klase tako da odvajanje granica između te dvije klase postaje maksimalno. Klasifikacija testnog primjera ovisi o strani hiperravnine, odnosno strani gdje se testni primjer nalazi. Ulazne značajke mogu se preslikati i u prostor visokih dimenzija, ali u tom slučaju, za smanjenje računskih troškova obuke i postupka ispitivanja u prostoru visokih dimenzija, koriste se neke funkcije kernela. Parametar regularizacije koristi se u slučaju neodvojivih primjera treninga. Zadana vrijednost ovog parametra smatra se 1.

SVM ima nekoliko metoda pomoću kojih se može izvesti klasifikacija: To su SVC, NuSVC i Linear SVC. Sve metode vrlo slične funkcioniraju, ali prihvaćaju malo drugačije skupove parametara i imaju različite matematičke formulacije. Imaju dva ulaza: niz X veličine *[n_samples, n_features]* koji sadrži skup za učenje i niza y sa klasnim oznakama veličine *[n_samples]*. Nakon postavljanja tih polja, model se može koristiti za predviđanje novih vrijednosti. SVM-ova funkcija odluke ovisi o nekom podskupu skupa za učenje koji se naziva vektor podrške. Koristeći ove vektore maksimiziramo marginu klasifikatora.



Slika 5: Primjeri SVM-a u različitim klasama[4]

Metoda SVM daje značajnu preciznost sa manje računarske snage. Također, učinkovita je u prostorima visokih dimenzija, čak i kada je broj dimenzija veći od broja uzoraka.

[4] Izvor: https://scikit-learn.org/stable/_images/sphx_glr_plot_iris_svc_001.png

4. Homonimi u hrvatskom jeziku

Hrvatski jezik morfološki je izrazito bogat i složen jezik [12]. U njemu postoje riječi koje su različitog podrijetla i značenja ali se jednako pišu ili izgovaraju (imaju isti grafemski i prozodemski slijed). Takve riječi nazivamo homonimima. Kod nas ljudi to ne predstavlja nikakav problem zato što iz konteksta možemo zaključiti značenje riječi. Međutim, veliki je izazov u informatičkoj znanosti stvoriti algoritam koji bi imao sposobnost razrješavanja tog problema.

U doktorskom radu „*Automatsko predviđanje i modeliranje hrvatskih prozodijskih obilježja na temelju teksta*“ [10] izrađen je hrvatski naglasni rječnik u kojem se svaka natuknica sastoji od naglašenog oblika, morfosintaktičke oznake i nenaglašenog oblika. Pomoću tog rječnika mogu se riješiti slučajevi kada imenice imaj različitu morfosintaktičku oznaku i/ili različiti naglasak te slučajevi imenica koje se jednako pišu, a različito izgovaraju. Problem ostaje sa riječima koje imaju istu morfositaktičku oznaku, isti naglasak, a dva ili više različitih značenja. Takav problem možemo riješiti isključivo preko konteksta. Jedan od načina rješavanja tog problema je pomoću konkordanci i kolokacija koje je pobliže opisano u završnom radu „*Razrješavanje višeznačnosti riječi pomoću konkordanci*“ [11].

U mnogim svjetskim jezicima postoje riječi čije značenje ovisi o kontekstu u kojem se nalaze. Razrješenje problema višeznačnosti (Word Sense Disambiguation, WSD) obuhvaća pronalaženje točnog značenja dvosmislene ili višeznačne riječi u određenom kontekstu [13]. Glavno područje primjene WSD-a je strojno prevođenje, međutim ono se koristi u gotovo svim vrstama lingvističkih istraživanja [14]. U istraživanjima provedenim u [15] i [16] pokazalo se da prethodno razrješavanje višeznačnosti riječi daje puno bolje rezultate strojnog prevođenja u hrvatskom jeziku. U hrvatskom jeziku ne pišu se naglasci pa u pisanim riječima postoje riječi i oblici koji imaju isti grafemski slijed, a različito značenje. Takve riječi nazivamo homogramima (*mol* i *mol*). Homografi ili istopisnice se podudaraju potpuno po grafemskom i po prozodemskom slijedu.

Morfološki bogatim jezicima se uz vrstu riječi dodaju i dodatne morfološke oznake: Takve se slučajeve za hrvatski jezik možemo riješiti pomoću hrvatskog naglasnog rječnika [10]. Međutim, kada imamo riječi koje imaju istu morfološku oznaku, isti naglasak a različito značenje (*bor – vrsta stabla, bor – kemijski element*), taj problem možemo riješiti isključivo

pomoću konteksta. No, kod automatskih postupaka razrješavanja višeznačnosti, nije lako odrediti pravo značenje.

Postoje dva glavna pristupa WSD-a – dubinski i plitki pristup. Plitki pristupi ne pokušavaju razumjeti tekst već samo razmatraju okolne riječi koristeći informacije poput „ako se uz riječ bor pojavljuju riječi „*stablo*“ i/ili „*šuma*“ tada je riječ o vrsti drva, a ako se pojavljuju riječi „*element*“ i/ili „*kemija*“, tada je riječ o kemijskom elementu“. Ova pravila mogu biti automatski izvedena od strane računala koristeći korpus riječi s oznakom značenja [17]. Postoje četiri konvencionalna pristupa u WSD-u, a to su pristupi utemeljeni na rječniku i znanju, polu-nadzirni ili minimalno nadzirani pristup, nadzirani pristup i nenadzirani pristup. Ovdje ćemo se baviti nadziranim pristupom. On se zasniva na pretpostavci da kontekst može dati dovoljno informacija za razrješavanje modela višeznačnosti. Nadzirani pristup koristi tehniku strojnog učenja iz ručno označenih značenja čime se ostvaruju bolji rezultati nego kod ostalih pristupa. Glavni postupak kod razrješavanja višeznačnost riječi je odlučivanje o tome koje je njihovo značenje. Problem kod tog postupka može se usporediti sa problemom dodjeljivanja oznaka riječima – oba uključuju razrješavanje ili označavanje riječi. Morfološki bogatim jezicima uz vrstu riječi dodaju se i dodatne morfološke oznake – takve slučajeve se u hrvatskom jeziku može riješiti pomoću hrvatskog naglasnog rječnika.

U završnom radu „*Razrješavanje višeznačnosti riječi pomoću konkordanci*“ [11] odabrane su pojedine riječi iz hrvatskog jezika koje imaju isti oblik, iste su vrste te imaju isti naglasak, ali dva ili više različita značenja. Odabrane riječi pronađene su u hrvatskim korpusima odakle je izvučen njihov kontekst, zatim je njihovo značenje ručno klasificirano na temelju konteksta. Za potrebe ovog rada iskoristi će se izvučene riječi s istim naglaskom i oblikom, a različitim značenjem i lista riječi korištenih za analizu grupirane po korištenim korpusima.

5. Primjena algoritma klasifikacije na problem razrješavanja višeznačnosti

Scikit-Learn je knjižnica koja pruža razne tehnike nadziranog i nenadziranog učenja te dolazi s nekoliko standardnih skupova podataka [18]. Osim za nadzirano strojno učenje (klasifikaciju i regresiju), može se koristiti i za klasteriranje, smanjenje dimenzionalnosti, izdvajanje značajki i inženjering, te obradu podataka. Sučelje je dosljedno svim ovim metodama, tako da ne samo da se lako koristi, već je i lako sastaviti veliki skup modela klasifikatora/regresije i osposobiti ih istim naredbama.

Slika 6: Logo Scikit - Learn[5]

5.1. Naive Bayes algoritam

Prvi korak je priprema dokumenta za klasifikaciju. Kako se ovdje radi o riječima s istim naglaskom i oblikom, a različitim značenjem, svaka riječ, odnosno rečenice u kojima se ta riječ spominje, nalaze se u zasebnoj tekstualnoj datoteci – rečenice sa riječju jednog značenja u jednoj datoteci, drugog značenja u drugoj datoteci i tako dalje. Te tekstualne dokumente potrebno je spojiti u jedan Excel dokument. Pri tome potrebno je svakoj rečenici dodijeliti značenje odnosno oznaku (na primjer rečenicama iz prvog dokumenta značenje 1, iz drugog dokumenta značenje 2 i tako dalje, ovisno koliko značenja pojedina riječ ima). Kada se to napravi, potrebno je kreirati podatkovni okvir sa zadanim nazivima stupaca – u ovom slučaju

[5] Izvor: https://commons.wikimedia.org/wiki/File:Scikit_learn_logo_small.svg

će ti nazivi biti „rečenica“ i „značenje“ (u kodu „sentence“ i „meaning“). Postupak kreiranja podatkovnog okvira u Python-u prikazan je u dijelu koda koji slijedi.

```
import pandas as pd

data = pd.read_excel("faks.xls", header=[0])
df = pd.DataFrame(data)

print (df)
```

Rezultat ovog koda je tablica odnosno podatkovni okvir sljedećeg izgleda (prikazani su početak i kraj):

```
                                                SENTENCE  MEANING
0    uvrijezenih metoda naobrazbe i upoznati svijet...         1
1    izgledima da ga stvarno nadje o plesnoj ponudi...         1
2    te poteskoce razmisljala je o tome sto zapravo...         1
3    bilo tako bitno da poduzme nesto znacajno tko ...         1
4    navrata odlazili tamo na radionice pocela je n...         1
5    koja ce se odrzati od  do  studenog  godine u ...         1
...
934  u sustavu provedbe veterinarske djelatnosti du...         2
935  takodjer americki novinari izvjestavaju da je ...         2
936  on ponovno nije pristupio saslusanju a sudac i...         2
937  s gradjanima upiti predstavke i zalbe gradjana...         2
938  u mom timu kad sam se nakon dva sata vratio mo...         2

[939 rows x 2 columns]
```

Za pokretanje Naive Bayesovog klasifikatora kategorije moraju biti numeričke. U ovom primjeru oznaka 0 dodijeljena je prvom značenju riječi, a oznaka 1 drugom značenju riječi.

```
df['label'] = df['MEANING'].apply(lambda x: 0 if x==1 else 1)
```

Ukoliko ima više značenja, dodati će se još „if“ izraza.

Slijedeći korak je podjela podataka na skup za učenje i skup za testiranje. Za to se može koristiti Scikit-Learn-ov train_test_split. Dio koda u kojem je implementiran taj postupak prikazan je u nastavku.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df['SENTENCE'],
                                   df['label'], random_state=1)
```

Zatim je potrebno rečenice pretvoriti u vektore brojanja riječi. Naive Bayes-ov klasifikator mora biti u mogućnosti izračunati koliko se puta svaka riječ pojavljuje u svakom dokumentu i koliko se puta pojavljuje u svakoj kategoriji, odnosno u ovom slučaju značenju. Da bi to bilo moguće podaci moraju izgledati ovako:

[0, 1, 0, …]

[1, 1, 1, ….]

[0, 2, 0, ….]

Svaki redak predstavlja dokument, a svaki stupac predstavlja riječ. Da bismo dobili rečenice u ovom formatu, koristimo CountVectorizer. On stvara vektor brojača riječi za svaku rečenicu kako bi tvorio matricu. Svaki indeks odgovara riječi i svaka riječ koja se pojavljuje u rečenici je zastupljena. Također, možemo koristiti razne argumente koji mogu biti korisni za predviđanje naših kategorija odnosno značenja. Ukoliko želimo vidjeti pregled podataka i istražiti brojanje riječi, to možemo učiniti uz pomoć podatkovnog okvira za brojanje riječi. Navedeni postupak prikazan je u dijelu koda koji slijedi.

```
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True,)
X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

word_freq_df = pd.DataFrame(X_train_cv.toarray(), columns=cv.get_feature_names())
top_words_df = pd.DataFrame(word_freq_df.sum()).sort_values(0, ascending=False)

print (word_freq_df)
print (top_words_df)
```

Nakon toga možemo ubaciti multinomni Naive Bayes klasifikator u naše podatke za učenje i upotrijebiti ga za predviđanje oznaka testnih podataka. Kod za navedeni klasifikator prikazan je niže.

```
from sklearn.naive_bayes import MultinomialNB

naive_bayes = MultinomialNB()
naive_bayes.fit(X_train_cv, y_train)
predictions = naive_bayes.predict(X_test_cv)

print(predictions)
```

Posljednji korak je prikaz rezultata, odnosno ispis rezultata točnosti, rezultata preciznosti i rezultat opoziva. Iz tih rezultata vidimo u koliko će posto vremena klasifikator ispravno predvidjeti o kojem se značenju riječi radi. Implementacija sljedećeg dijela koda nalazi se u nastavku.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

print('Accuracy score: ', accuracy_score(y_test, predictions))
print('Precision score: ', precision_score(y_test, predictions))
print('Recall score: ', recall_score(y_test, predictions))
```

Kako bi bolje razumjeli rezultate, možemo dodati dio koda koji slikovno prikazuje predviđene oznake, te koliko je model točno predvidio a koliko netočno. Navedeni dio koda prikazan je niže.

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['1', '2'], yticklabels=['1', '2'])
plt.xlabel('true label')
plt.ylabel('predicted label')
```

Također, kako bi istražili netočne oznake, možemo staviti stvarne oznake i predviđene oznake u podatkovni okvir. To možemo učiniti uz pomoć sljedećeg koda:

```
testing_predictions = []
for i in range(len(X_test)):
    if predictions[i] == 1:
        testing_predictions.append('1')
    else:
        testing_predictions.append('2')
check_df = pd.DataFrame({'actual_label': list(y_test), 'prediction': testing_predictions, 'sentence':list(X_test)})
check_df.replace(to_replace=0, value='1', inplace=True)
check_df.replace(to_replace=1, value='2', inplace=True)
print(testing_predictions)
```

6. Rezultati

U ovom poglavlju detaljno je prikazan rezultat algoritma za riječ „faks“. Prvo značenje riječi je fakultet, a drugo značenje je telefaks.

Izgled podatkovnog okvira:

	SENTENCE	MEANING
0	uvrijezenih metoda naobrazbe i upoznati svijet...	1
1	izgledima da ga stvarno nadje o plesnoj ponudi...	1
2	te poteskoce razmisljala je o tome sto zapravo...	1
3	bilo tako bitno da poduzme nesto znacajno tko ...	1
4	navrata odlazili tamo na radionice pocela je n...	1
5	koja ce se odrzati od do studenog godine u ...	1
6	tjedna na nastavi a onda kraj nastave za maut...	1
7	za precednicke izbore bil je i rektor sveucili...	1
..	...	...
909	pristup internetu j omogucava pristup webmjest...	2
910	uredjaju slanje i primanje faksa s drugog komp...	2
911	vukovara zagreb ili na email adresu j ...	2
912	nove i stare brojeve rheme mozete naruciti sva...	2
913	prema lyonu koji je bio spreman ponuditi cetir...	2
914	gradski ured za zdravstvo i socijalnu skrb bra...	2
915	obnasa tu duznost ponavljam pitanje gospodin k...	2

Dodjeljivanje oznaka za svako značenje riječi:

	SENTENCE	MEANING	label
0	uvrijezenih metoda naobrazbe i upoznati svijet...	1	0
1	izgledima da ga stvarno nadje o plesnoj ponudi...	1	0
2	te poteskoce razmisljala je o tome sto zapravo...	1	0
3	bilo tako bitno da poduzme nesto znacajno tko ...	1	0
4	navrata odlazili tamo na radionice pocela je n...	1	0
5	koja ce se odrzati od do studenog godine u ...	1	0
6	tjedna na nastavi a onda kraj nastave za maut...	1	0
..	...	...	...
909	pristup internetu j omogucava pristup webmjest...	2	1
910	uredjaju slanje i primanje faksa s drugog komp...	2	1
911	vukovara zagreb ili na email adresu j ...	2	1
912	nove i stare brojeve rheme mozete naruciti sva...	2	1
913	prema lyonu koji je bio spreman ponuditi cetir...	2	1
914	gradski ured za zdravstvo i socijalnu skrb bra...	2	1

Pretvorba rečenica u vektore brojanja riječi:

	a	abbyy	acidu	acovjeka	administracije	administratorhzhmhr	adresa	\
0	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	

...

Prikaz podatkovnog okvira za brojanje riječi:

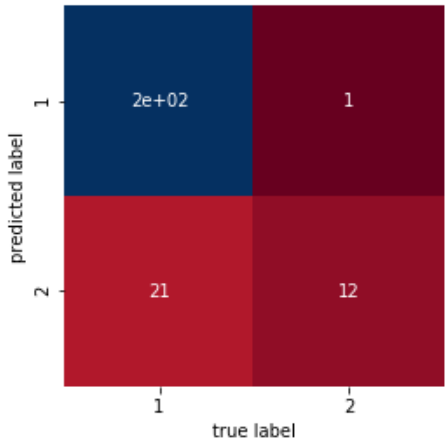
	0
i	886
faks	753
je	660
na	562
da	541
se	496
u	486
sam	327
za	271
ne	247

Predviđanje oznaka testnih podataka :

```
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 0 0]
```

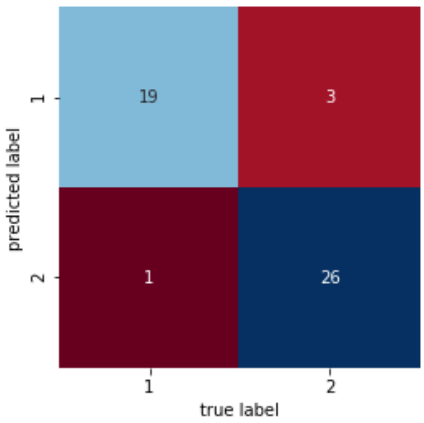
Prikaz rezultata:

```
Accuracy score:  0.9063829787234042
Precision score:  0.9230769230769231
Recall score:  0.36363636363636365
```



Iz navedenog vidimo da je točnost za riječ „faks“ u svim rečenicama preko 90%. Rezultat preciznosti je također vrlo visok i iznosi 92%. On označava koliko je puta prvo značenje označeno kao prvo, odnosno točno. Do problema dolazi kod rezultata opoziva koji je svega 36%. Taj rezultata označava da drugo značenje nije dobro predviđeno. Pretpostavka je da se to dogodilo iz razloga što u ovom primjeru u korpusu za učenje i testiranje ima puno više rečenica za prvo značenje riječi nego za drugo značenje. Smanjenjem broja rečenica za prvo značenje, dobili su se puno bolji rezultati opoziva. U nastavku su prikazani rezultati kad se je u korpusu ujednačio broj rečenica za jedno i drugo značenje.

```
Accuracy score:  0.9183673469387755
Precision score:  0.896551724137931
Recall score:  0.9629629629629629
```



Iz primjera se vidi da je riječima iz testnog skupa u 19 slučajeva pravilno predviđeno prvo značenje riječi „faks“ te da je prvo značenje 3 puta krivo predviđeno. Drugo je značenje pravilno predviđeno 26 puta i jedanput krivo.

U nastavku slijedi još nekoliko prikaza dobivenih rezultata za pojedine riječi.

Riječ „žal“ – prvo značenje riječi je obala, a drugo značenje je žalost:

```
Accuracy score:  1.0
Precision score:  1.0
Recall score:  1.0
```

Riječ „gore“ – prvo značenje riječi je suprotno od dole, a drugo značenje je jako loše:

```
Accuracy score:  0.8071428571428572
Precision score:  0.7777777777777778
Recall score:  0.9891304347826086
```

Riječ „predavati“ – prvo značenje riječi je predavati nekome nešto, a drugo je predati nekome nešto:

```
Accuracy score:  0.8275862068965517
Precision score:  1.0
Recall score:  0.5652173913043478
```

Riječ „reket“ – prvo značenje riječi je reketiranje, a drugo je tenis:

```
Accuracy score:  0.7894736842105263
Precision score:  0.7941176470588235
Recall score:  0.9642857142857143
```

Prosječna točnost za navedenih 5 riječi je 0,87 što je prilično dobar rezultat. U budućem radu, rezultat će se pokušati poboljšati izbacivanjem stop riječi (en. stopwords) za hrvatski jezik – riječi poput veznika, prijedloga i slično koje nemaju semantičkog značenja te mogu pridonijeti lošijem rezultatu modela klasifikacije.

7. Zaključak

Klasifikatori su se pokazali dobrima u mnogočemu – od klasifikacije dokumenata i e-pošte do korištenja u oglašavanju i zdravstvenim tvrtkama. Od velike su pomoći te vrlo dobro funkcioniraju gdje god bili korišteni. Ovisno o njihovoj primjeni, koristi se algoritam koji je najprikladniji za zadani problem.

Kod rješavanja problema višeznačnosti u hrvatskom jeziku korišten je multinomni Naive Bayes algoritam. Iz rezultata možemo vidjeti da je točnost predviđanja za zadane riječi vrlo dobra. Međutim, pokazalo se da ukoliko korpus za učenje i testiranje modela sadrži puno više rečenica jednog značenja od drugog značenja, rezultati neće biti toliko dobri. Iz tog razloga se isprobalo uravnotežiti korpus tako da je otprilike podjednaki broj rečenica na kojima se model uči i za jedno i za drugo značenje čime su se dobili bolji rezultati. U budućem radu, rezultati će se pokušati još poboljšati izbacivanjem stop riječi za hrvatski jezik – riječi poput veznika, prijedloga i slično koje nemaju semantičkog značenja te mogu pridonijeti lošijem rezultatu modela klasifikacije. Također će se isprobati i drugi modeli klasifikacije te će se izgraditi modeli za sve riječi hrvatskog jezika koje imaju isti oblik, iste su vrste i imaju isti naglasak, ali dva ili više različita značenja.

8. Popis izvora

[1] https://www.fer.unizg.hr/predmet/su

[2] Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer

[3] Rascha, Sebastian, (2016.), „Python Machine Learning“, Chapter 3: A Tour of Machine Learning Classifiers Using Scikit-learn, Packt Publishing, Birmingham

[4] https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf

[5] Scikit-learn: Naive Bayes https://scikit-learn.org/stable/modules/naive_bayes.html (2017 - 2019)

[6] Pal, A. R., Saha D., (2015.), „Word Sene Disambiguation: A Suvery“, International Jurnal of Control Theory and Computer Modeling (IJCTCM), Vol.5, No.3, July 2015

[7] Scikit-learn: Decision Trees https://scikit-learn.org/stable/modules/tree.html (2017 - 2019)

[8] Scikit-learn: Support Vector Machines https://scikit-learn.org/stable/modules/svm.html (2017 - 2019)

[9] Vapnik, V., Cortes C. (1995.), „Support Vector Networks“

[10] Načinović, Prskalo, Lucia: Automatsko predviđanje i modeliranje hrvatskih prozodijskih obilježja na temelju teksta, Zagreb 2016. (doktorski rad)

[11] Smoković, Matea: Razrješavanje višeznačnosti riječi pomoću konkordanci, Rijeka 2018. (završni rad)

[12] Hržica G., Ordulj A. (2013.), „Dvočlane glagolske konstrukcije u usvajanju hrvatskog jezika“, Zagreb (znanstveni rad)

[13] Ide N., Véronis J., (1998.) „Word Sense Disambiguation: The State of the Art“, Computational Linguistics, Vol. 24, No. 1

[14] Sanderson, M., (1994.) „Word Sense Disambiguation and Information Retrieval“, Proceedings of the 17th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR‟94, Dublin, Ireland, Springer, New York, pp 142-151

[15] Nacinovic Prskalo, Lucia; Brkic Bakaric, Marija, Disambiguation of Homograms in a Pitch Accent Language // Proceedings of 2017 International Conference on Computer Science and Artificial Intelligence CSAI 2017 New York: ACM, 2017. str. 32-37 doi:10.1145/3168390.3168409

[16] Nacinovic Prskalo, Lucia; Brkic Bakaric, Marija: The Role of Homograms in Machine Translation // International journal of machine learning and computing (IJMLC), 8 (2018.), 2; 90-97 doi:10.18178/ijmlc.2018.8.2.669

[17] Xiaohua Zhou, Hyoil Han (2005.),“Survey of Word Sense Disambiguation Approaches”, College of InformationScience & Technology, Drexel University 3401Chestnut Street, Philadelphia, PA 19104, Appeared inThe 18th FLAIRS Conference, Clearwater Beach,Florida

[18] Data Robot: Classification with scikit-learn https://blog.datarobot.com/classification-with-scikit-learn (3.3.2014.)

9. Popis literature

1. Ethem, Alpaydin, (2014.), „Introduction to Machine Learning, Third edition“, The MIT Press Cambridge, Massachusetts Institute of Technology, London
2. Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer
3. [MREŽNO] https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf (2019.)
4. Bonaccorso, Giuseppe, (2017.), „Machine Learning Algorithms“, Chapters 3, 6, 7, 8, Packt Publishing, Birmingham
5. [MREŽNO] Data Science Central: An Introduction to Classification Models https://www.datasciencecentral.com/profiles/blogs/data-science-simplified-part-10-an-introduction-to-classification (19.9.2017.)
6. [MREŽNO] Towards Data Science: Machine Learning Classifiers https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623 (11.6.2018.)
7. [MREŽNO] Scikit-learn: Naive Bayes https://scikit-learn.org/stable/modules/naive_bayes.html (2017 - 2019)
8. [MREŽNO] https://web.stanford.edu/class/cs124/lec/naivebayes.pdf
9. [MREŽNO] Towards Data Science: Introduction to Naive Bayes Classification https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54 (16.5.2018)
10. Pal, A. R., Saha D., (2015.), „Word Sene Disambiguation: A Suvery“, International Jurnal of Control Theory and Computer Modeling (IJCTCM), Vol.5, No.3, July 2015
11. [MREŽNO] Scikit-learn: Decision Trees https://scikit-learn.org/stable/modules/tree.html (2017 - 2019)
12. [MREŽNO] Towards Data Science: Everithing You Need to Know About Decision Trees https://towardsdatascience.com/everything-you-need-to-know-about-decision-trees-8fcd68ecaa71 (16.1.2019.)

13. [MREŽNO] Scikit-learn: Support Vector Machines https://scikit-learn.org/stable/modules/svm.html (2017 - 2019)

14. [MREŽNO] Towards Data Science: Support Vector Machine – Introduction to Machine Learning Algoriths https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (7.6.2018.)

15. Vapnik, V., Cortes C. (1995.), „Support Vector Networks“

16. Načinović Prskalo, Lucia: Automatsko predviđanje i modeliranje hrvatskih prozodijskih obilježja na temelju teksta, Zagreb 2016. (doktorski rad)

17. Nacinovic Prskalo, Lucia; Brkic Bakaric, Marija: The Role of Homograms in Machine Translation // International journal of machine learning and computing (IJMLC), 8 (2018.), 2; 90-97 doi:10.18178/ijmlc.2018.8.2.669 (međunarodna recenzija, članak, znanstveni)

18. Nacinovic Prskalo, Lucia; Brkic Bakaric, Marija, Disambiguation of Homograms in a Pitch Accent Language // Proceedings of 2017 International Conference on Computer Science and Artificial Intelligence CSAI 2017 New York: ACM, 2017. str. 32-37 doi:10.1145/3168390.3168409

19. Smoković, Matea: Razrješavanje višeznačnosti riječi pomoću konkordanci, Rijeka 2018. (završni rad)

20. Hržica G., Ordulj A. (2013.), „Dvočlane glagolske konstrukcije u usvajanju hrvatskog jezika“, Zagreb (znanstveni rad)

21. Ide N., Véronis J., (1998.) „Word Sense Disambiguation: The State of the Art“, Computational Linguistics, Vol. 24, No. 1

22. Sanderson, M., (1994.) „Word Sense Disambiguation and Information Retrieval“, Proceedings of the 17th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR‟94, Dublin, Ireland, Springer, New York, pp 142-151

23. Xiaohua Zhou, Hyoil Han (2005.),“Survey of Word Sense Disambiguation Approaches”, College of InformationScience & Technology, Drexel University 3401Chestnut Street, Philadelphia, PA 19104, Appeared inThe 18th FLAIRS Conference, Clearwater Beach,Florida

24. Rascha, Sebastian, (2016.), „Python Machine Learning“, Chapter 3: A Tour of Machine Learning Classifiers Using Scikit-learn, Packt Publishing, Birmingham

25. [MREŽNO] Ahmet Taspinar: Classification with Scikit-Learn http://ataspinar.com/2017/05/26/classification-with-scikit-learn/ (26.5.2017.)

26. [MREŽNO] Digital Ocean: How to Build a Machine Learning Classifier in Python with Scikit-Learn https://www.digitalocean.com/community/tutorials/how-to-build-a-machine-learning-classifier-in-python-with-scikit-learn (24.3.2019.)

27. [MREŽNO] Data Robot: Classification with scikit-learn https://blog.datarobot.com/classification-with-scikit-learn (3.3.2014.)

28. [MREŽNO] Codementor Community: Introduction to Machine Learning with Python's Scikit-learn https://www.codementor.io/garethdwyer/introduction-to-machine-learning-with-python-s-scikit-learn-czha398p1 (18.10.2017.)

29. https://www.nltk.org/book/ch06.html

10.Popis priloga

1. Tekstualne datoteke sa izvučenim riječima s istim naglaskom i oblikom, a različitim značenjem
2. Tekstualne datoteke sa listom riječi korištenih za analizu grupirane po korištenim korpusima
3. Excel datoteke sa riječima istog naglaska i oblika, a različitog značenja
4. Python datoteka sa multinomnim Naive Bayes algoritmom za klasifikaciju korištenim u primjeru razrješavanja problema višeznačnosti u hrvatskom jeziku