SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 660

SUSTAV ZA OZNAČAVANJE TEKSTUALNOG SADRŽAJA NAVOĐENO AKTIVNIM UČENJEM

Franjo Mindek

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 660

SUSTAV ZA OZNAČAVANJE TEKSTUALNOG SADRŽAJA NAVOĐENO AKTIVNIM UČENJEM

Franjo Mindek

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

ZAVRŠNI ZADATAK br. 660

Pristupnik:	**Franjo Mindek (0036523095)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Mario Brčić

Zadatak: **Sustav za označavanje tekstualnog sadržaja navođeno aktivnim učenjem**

Opis zadatka:

Strojno učenje je veoma uspješna, štoviše dominantna, paradigma za obradu teksta. No, efektivnost tih metoda znatno ovisi o kvaliteti ulaznih podataka. Za postojeća je rješenja uobičajeno da se utreniravaju nad podatcima koji su indiskriminativno skupljeni s interneta i labelirani za neku općenitu ili specifičnu svrhu. To rezultira potrebom za velikim skupovima da bi se ostvarila željena performansa. Za uspjeh samog model u određenoj zadaći je bitno imati kvalitetan skup podataka koji učinkovito pokriva prostor mogućnosti tako da model može pouzdano i precizno vršiti svoju namjenu. Aktivno učenje povećava uzorkovnu učinkovitost postupka utreniravanja modela jer se pronalaze čim informativniji uzorci. Time se smanjuju veličina, trošak i trajanje postupka označavanja skupa podataka. U ovom radu treba, koristeći dostupne alate, implementirati sustav za označavanje tekstualnog sadržaja koje je navođeno aktivnim učenjem. Označivačima podataka treba biti izloženo web sučelje koje ih vodi kroz postupak te im zadaje uzorke koji su visoko informativni za odabrani model.

Rok za predaju rada: 10. lipnja 2022.

SADRŽAJ

1. Uvod

Zbog potrebe rješavanja UI[1]-potpunih problema sve se više susrećemo s raznim primjenama strojnog učenja. Ipak, zbog ulazno-izlazne povezanosti podataka i rezultata modela strojnog učenja nužno je imati kvalitetan skup podataka.

Većina problema umjetne inteligencije rješava se treniranjem modela nad indiskriminativno skupljenim podatcima koji su često označeni za preopćenitu ili nama krivo specifičnu svrhu, nisu potpuni i sadrže netočne informacije. Rezultat toga je potreba za ogromnim skupovima podataka varirajuće kvalitete kako bi se približili željenoj efikasnošću modela [5]. Uspjeh samog modela zasniva se na kvalitetnom skupu podataka koji efikasno razapinje prostor problema, što modelu omogućuje pouzdane i precizne odgovore [16].

Cilj aktivnog učenja je povećati uzorkovanu učinkovitost postupka utreniravanja modela minimizirajući broj podataka i maksimizirajući dobit traženog informacijskog svojstva [28]. Tim postupkom smanjujemo trošak i vrijeme stvaranja [4] te veličinu skupa podataka dok istovremeno povećavamo i preciznost modela.

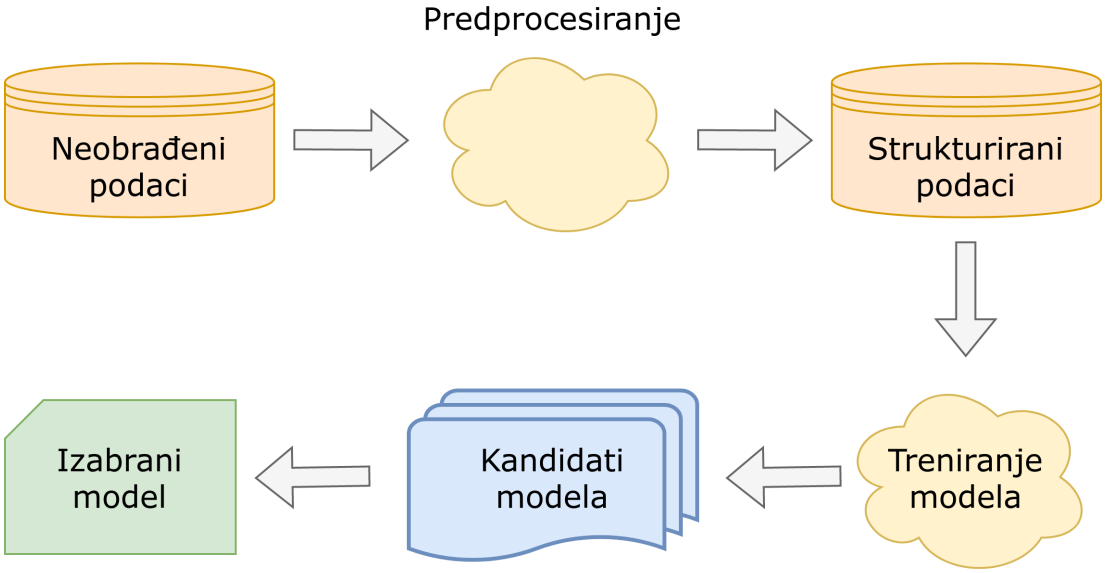
U ovom završnom radu koristeći javno dostupne alate, implementirat ćemo sustav za označavanje tekstualnog sadržaja navođeno aktivnim učenjem. Označivačima podataka bit će ponuđeno web sučelje koje će ih uvesti u postupak označavanje te će im zadavati uzorke visoke informacijske dobiti za odabrani model.

[1] UI - umjetna inteligencija

2. Označavanje podataka

Označavanje podataka je jedan od početnih koraka u procesu stvaranju modela strojnog učenja te ga svrstavamo u razinu pretprocesiranja. Definiramo ga kao proces prepoznavanja neobrađenih podataka (slika, tekst, video...) koji se onda pomoću jedne ili više oznaka opisuju u kontekstu modela, što dozvoljava modelu strojnog učenja da može stvarati točna predviđanja.

Označavanje podataka često je najsporiji i najvažniji dio modeliranja modela strojnog učenja. U prosjeku više od 80% ukupnog vremena modeliranja umjetne inteligencije potroši se na rad s podacima [26]. Vremenska neefikasnost označavanja podataka dovela je do potrebe razvoja modela umjetne inteligencije koji pomažu u procesu označavanja podataka. Jedan primjer takve umjetne inteligencije je aktivno učenje čiji ćemo utjecaj na označavanje podataka obraditi u Poglavlje 3.



Slika 2.1: Pojednostavljeni dijagram stvaranja modela strojnog učenja.

2.1. Strojno učenje i podatci

Problem s kojim se danas u svijetu često susrećemo jest to da imamo pristup obilju podataka, danas skupovi podataka često se broje u milijunima redaka. U susretu s takvom količinom podataka čovjeku je prezahtjevan posao da iz njih izvući sve međusobne skrivene povezanosti. Upravo iz potrebe iskorištavanja tog informacijskog potencijala razvijali su se sustavi koji na temelju podatak mogu učiti, a po završetku učenja nuditi korisna ponašanja. Uspješan primjer takvih sustava je umjetna inteligencija, gdje nam je posebno zanimljiva grana strojnog učenja zbog njene povezanost s podacima.

Strojno učenje je programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva. Raspolažemo modelom koji je definiran do na neke parametre, a učenje se svodi na izvođenje algoritma koji optimizira parametre modela na temelju podataka ili prethodnog iskustva [3].

Iz prethodnih definicija možemo uočiti povezanost između strojnog učenja i skupa podataka. Odgovor ili rješenje problema kojeg zahtijevamo od strojnog učenja je izlaz, ali taj izlaz možemo dobiti samo temeljem ulaza. Ako smo odabrali dobar skup podataka vrlo vjerojatno doći ćemo do traženog rješenja problema (osim ako su podaci prekomplicirani za izabrani model), no ako je naš skup podataka na bilo koji način iskrivljen ta iskrivljenost će se propagirati i na izlaze, što dovodi do krivih spoznaja.

Općenito se u strojnom učenjem susrećemo s dva tipa podataka, to su numerički ili kategorički podatci. Numeričke podatke predstavljaju brojevi nad kojima možemo vršiti računske operacije. To može biti visina čovjeka, cijena dionica, količina prodanih artikala i slično. Kategorički podatci su podatci podijeljeni u kategorije. Dijelimo ih na 2 vrste kategoričkih varijabli, nominalne i ordinalne. Razlikuju se po tome što nominalne varijable nemaju definiran poredak unutar kategorija. Primjer nominalne kategorije jest kada trebamo kategorizirati životinje sa slike, a dane su nam opcije "pas", "mačka" i "krava". Nemoguće je uspostaviti značenje usporedbe "krava" > "mačka". S druge strane ordinalne kategorije često možemo susresti u anketama zadovoljstva korisnika. Ako su korisniku ponuđene opcije "potpuno zadovoljan", "skoro potpuno zadovoljan", "jako zadovoljan"... podatke možemo lako staviti u poredak. Ono što je zajedničko nad oba kategorička primjera jest da nad njima ne možemo raditi aritmetiku.

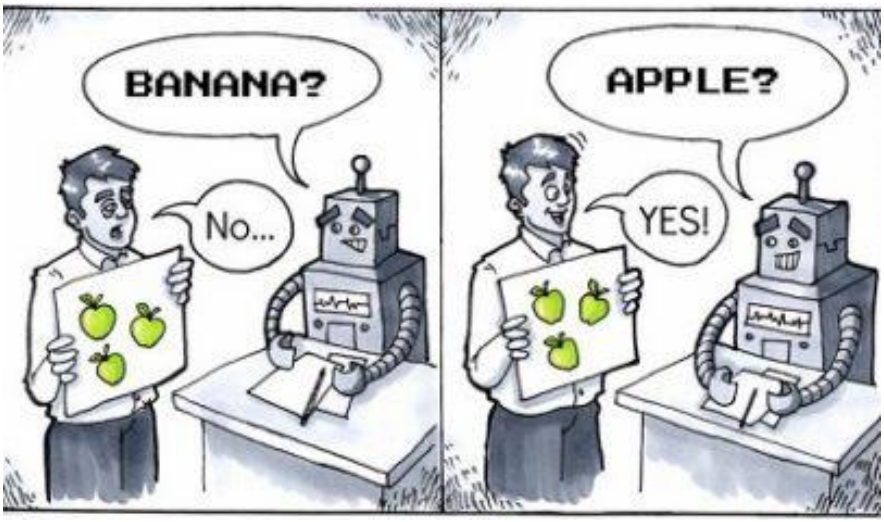
2.1.1. Paradigme strojnog učenja i strukturiranost skupa podataka

U pravilu strojno učenje možemo podijeliti u tri paradigme koje se između ostaloga razlikuju po izvoru informacija pomoću kojih se treniraju. Ovisno o izabranoj paradigmi i samoj namjeni modela drukčije ćemo pristupati procesu strukturiranja podataka.

Nadzirano učenje

Cilj nadziranog učenja je naučiti model kako ulaz preslikavati na izlaz. Definirano je korištenjem označenih skupova podataka koji sadrže informacije o ulazu i značajkama tog ulaza, to jest izlazu svakog uzorka. Ti skupovi podataka treniraju i „nadziru“ algoritam u procesu klasifikacije ili regresije. To jest, na temelju poznatih rješenja predviđamo rješenje problema, ili u smislu podataka, predviđamo izlaz za nove, neoznačene podatke na temelju označenih podatka.

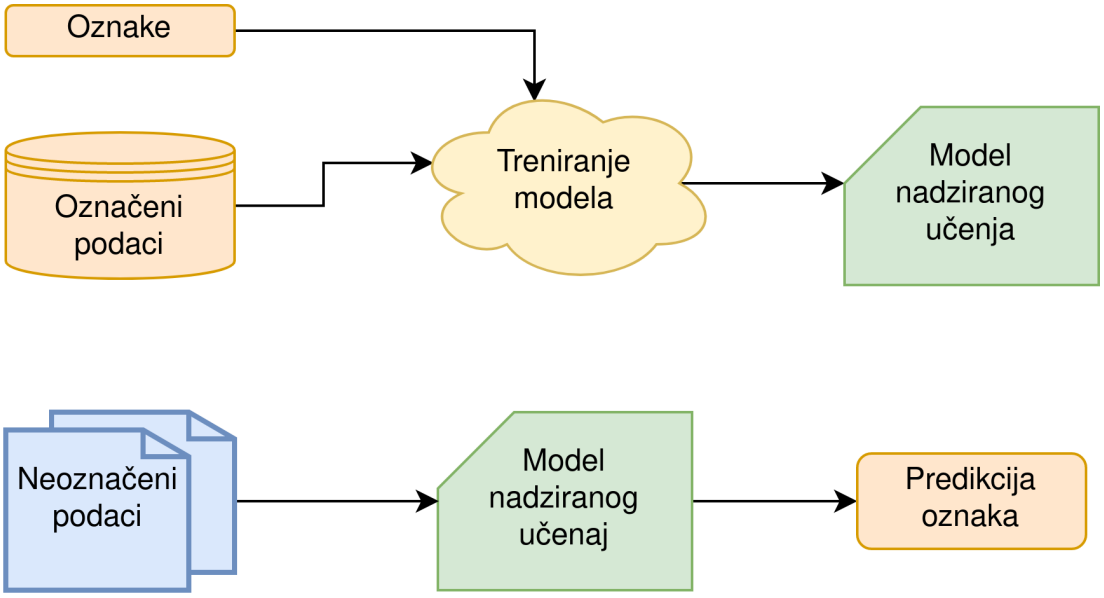
Naziv nadziranog učenja dolazi iz profesor-student odnosa tijekom procesa treniranja modela. Profesor usmjerava studenta iz kojih materijala učiti, a po završetku učenja student se ocjenjuje. Ako je student točan prolazi bez ispravljanja, ako je pak netočan, profesor ispravlja studenta i navodi ga da uči na temelju svojih pogrešaka.



Slika 2.2: Ilustracija odnosa tijekom modeliranja modela nadziranog učenja. [20]

Klasifikacija označava da radimo s izlazom koji je kategorička varijabla, što znači da naš izlaz treba poprimiti vrijednost jedne od preddefiniranih kategorija. Na primjeru poštanskog sandučića, dolaznu poštu možemo kategorizirati na *spam* i normalnu poštu.

Regresija ipak označava da radimo s izlazom brojčane vrijednosti. Cilj regresije je riješiti problem otkrivanja veza zavisnih i nezavisnih varijabli. Primjer toga može biti da imamo tvrtku koja svake godine ulaže u razne vrste marketinga. Sada na primjeru povijesnih podataka želi predvidjeti prodaju nove marketinške kampanje.



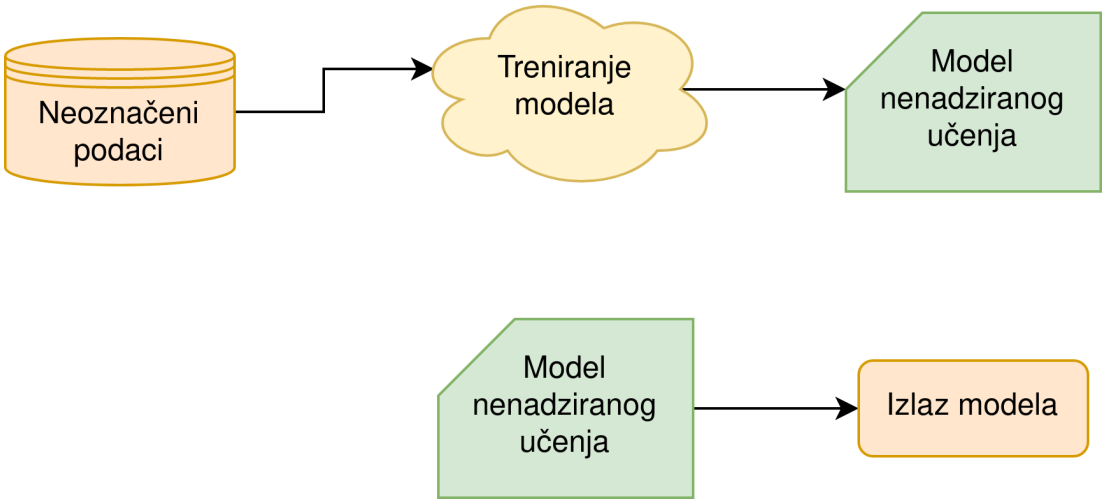
Slika 2.3: Pojednostavljeni dijagram faze učenja i iskorištavanja modela nadziranog učenja

Nenadzirano učenje

Nenadzirano učenje se koristi kada skup podataka nije označen, to jest kada nisu definirani izlazi ulaza. Koristi se kada je takve podatke potrebno analizirati i grupirati. Cilj je da algoritam otkrije skrivene uzorke ponavljanja u podacima bez potrebe ljudskog nadzora. U postupke nadziranog učenja spadaju grupiranje, otkrivanje stršećih/novih vrijednosti, otkrivanju povezanosti i smanjenje dimenzionalnosti.

Cilj grupiranja jest da podatke na temelju sličnost razdijeli u određeni broj kategorija. Taj broj kategorija možemo biti unaprijed određen, ili dinamički izračunat od strane algoritma.

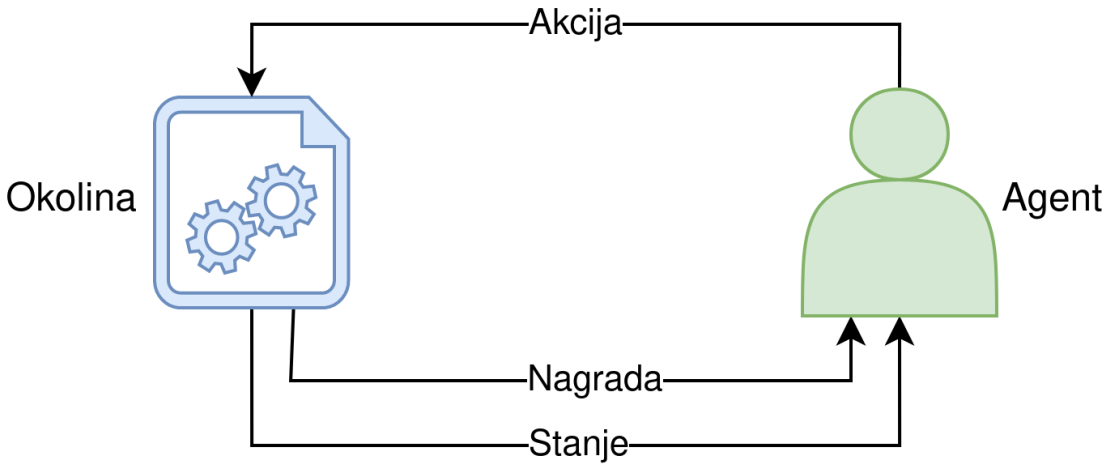
Postupak otkrivanja stršećih/novih vrijednosti pokušava naći podatke unutar skupa podataka koji su se po nekom određenom kriteriju dovoljno razlikuju. Ti podatci često predstavljaju pogreške u podacima ili neke nove spoznaje, zbog čega ih i želimo otkriti.



Slika 2.4: Pojednostavljeni dijagram faze učenja i iskorištavanja modela nenadziranog učenja

Podržano učenje

Podržano/ojačano učenje predstavlja dio strojnog učenja koje se bavi optimizacijom ponašanja gdje je cilj učenje optimalne strategije na temelju pokušaja s odgođenom nagradom. Definirano je agentom, okolinom i akcijama koje agent može raditi. Agent na temelju informacija iz okoline obavlja akcije za koje je onda nagrađen ili kažnjen, a njegova je zadaća otkriti strategiju tako da maksimizira nagrade koje dobiva *na duge staze*.



Slika 2.5: Pojednostavljeni dijagram podržanog učenja

2.2. Karakteristike dobrog skupa podataka

Karakteristike modela strojnog učenja, a time i skupova podataka nad kojima se trenira uvelike ovise o namjeni umjetne inteligencije. Svejedno možemo opisati određena svojstva koja su uvijek poželjna unutar skupa podataka neovisno o njihovoj primjeni.

1. Potpunost - pretpostavka je da u skupu podataka ne postoje prazne „ćelije“. Svaki uzorak unutar skupa podataka mora sadržavati informacije o svim traženim oznakama. Neispravne uzorke treba nadopuniti informacijama ili izbrisati iz skupa podataka. [5]

2. Objedinjenost - potrebno je pokriti cijelo područje problema koje se rješava. Primjer, model koji predviđa emocije preko slike lica osobe mora u svom skupu podataka sadržavati slike lica ljudi iz raznih kultura i različitih fizičkih karakteristika. Kako različite kulture drukčije iskazuju emocije, a i postoje fizičke razlike u strukturama lica, bez dovoljne objedinjenosti model ne bi zadovoljavao svojstvo generalizacije. [17] [2]

3. Relevantnost - podatci ne bi trebali pokrivati područje izvan problema. Pokrivanjem područja koji nisu dio našeg problema smanjuje se preciznost unutar područja problema. Kod skupova podataka sakupljenih s vanjskih izvora potrebno je filtrirati samo nama korisne informacije. Često je posljedica nerazumijevanja svojstva objedinjenosti podataka.

4. Ortogonalnost - prostor problema koji opisujemo trebamo opisati sa što manje ponavljanja između uzoraka. [1] [17] [2]

5. Konzistentnost - kriterij opisivanja uzoraka skupa podataka mora ostati konzistentan. S potrebom ogromnih skupova podataka od više desetaka tisuća do nekoliko milijuna uzoraka rijetko će sve uzorke opisati jedna osoba. U takvim slučajevima s više označivača podatci svejedno moraju ostati konzistentno opisani.

6. Ispravnost - po principu „*garbage in, garbage out*“, neispravni podatci skupa podataka dovode do neispravnih rezultata modela. Problem je više izražen kada radimo s podacima subjektivne prirode. [5]

7. Suvremenost - podatci moraju opisivati trenutno stanje problema koji se opisuje. Primjer, model koji predviđa rezultat sportskog natjecanja treba biti

upoznat s trenutnim stanjem suparničkih timova. U protivnom, on ne ispunjava svoju svrhu. [18]

Upravo zbog potreba zadovoljavanja što više korisnih svojstava skupova podataka dovelo je do razvoja mnogih internih i javno dostupnih sustava označavanja podataka te razvoja UI sa svrhom ubrzanja i povećanja preciznosti tog procesa. [26]

2.3. Alati za označavanje podataka

Zbog porasta veličine skupova podataka sustavi za označavanje podataka postali su nužni dio tehnologije strojnog učenja. Njihova korist može biti od bržeg označavanja do veće kvalitete i konzistentnosti označavanja podataka.

Alati omogućuju označavanje podataka pomoću lako upotrebljivih grafičkih sučelja koji proces označavanja rastavljaju na manje podzadatke. Prednost rastavljanja zadataka na podjedinice je to što određene podzadatke koje modeli strojnog učenja obavljaju velikom preciznošću možemo automatizirati te ljudima prepustiti rad nad podzadacima koje modeli još nišu savladali.

Također rastava posla na podzadatke omogućuje lakše praćenje, mjerenje i analizu posla koja dovodi do veće kvalitete označavanja podataka. Segmentacijom posla provjeru podataka možemo ostaviti osobama nadležnim za osiguravanje kvalitete pojedinog segmenta procesa.

U ovisnosti o podacima koje koristimo bitno je uzeti prikladan alat za označavanje podataka. Alati se razlikuju po funkcionalnostima administrativnog sučelja, pregledu informacija pomoću statističke analize, podržanim vrstama podataka (tekst, slika, zvuk, video...) i podržanim vrstama označavanja podataka (klasifikacija, označavanje sekvence...).

Zbog specifičnih potreba problema modeliranja strojnog učenja većina velikih poduzeća okreće se prema internom rješenju problema alata ili rjeđe profesionalnim vanjskim alatima [26]. Uslijed potrebe stvaranja sustava za označavanje tekstualnog sadržaja opisat ćemo dva javno dostupa alata koji pružaju web sučelje i zadovoljavaju specifičnosti zadanog problema.

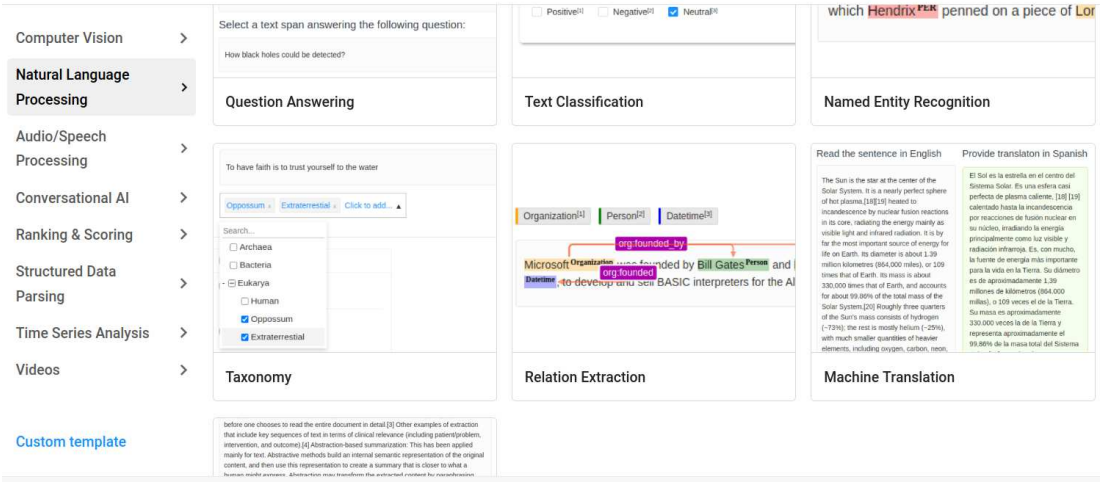
2.3.1. Label Studio

Label Studio je sustav za označavanje otvorenog izvora koje integrira grafičko sučelje kroz cijeli proces označavanja. Omogućuje označavanje tekstualnog, slikovnog,

audio, video i vremenskog sadržaja (te hibridnih oblika), podržavajući pri tome velik izbor predefiniranih predložaka pravila i sučelja označavanja podataka. U slučaju da ne sadrži predložak željene vrste označavanja podataka nudi mogućnost lakog stvaranja vlastitih pravila i sučelja. Isto tako nudi i bogate funkcionalnosti filtriranja i sortiranja već učitanih podataka, što olakšava snalaženje na velikim projektima. Podupire organizaciju i rad na više projekata odjednom te podržava rad više korisnika istovremeno što ga činim pogodnim za rad u timovima.

Također dolazi uz gotove Docker slike[1] te predloške potrebne za pokretanje programa na popularnijim servisima u oblaku što omogućava lagano pokretanje timsko orijentiranog projekta označavanja podataka.

Ipak, problem Label Studia je to što većinu administrativnih mogućnosti skriva iza verzije za poduzetnike koja nije otvorenog izvora i zahtijeva kupovanje programa. Najveći propust je to što su svi sudionici projekta ujedno i administratori projekta, što znači da bilo koji označivač podataka može izbrisati sve podatke projekata koje označuje. U slučaju da želimo imati raznolike uloge ovisno o zadatku i ovlastima korisnika to nije moguće ostvariti bez vlastite implementacije. [14] [15]



Slika 2.6: Label Studio pokušava biti „all-in-one“ rješenje sustava označavanja podataka.

2.3.2. Doccano

Doccano je također sustav za označavanje podataka otvorenog izvora koji integrira grafičko sučelje u proces označavanja podataka. Za razliku od Label Studia, Doccano se usredotočio na označavanje tekstualnog sadržaja gdje podržava klasifikaciju,

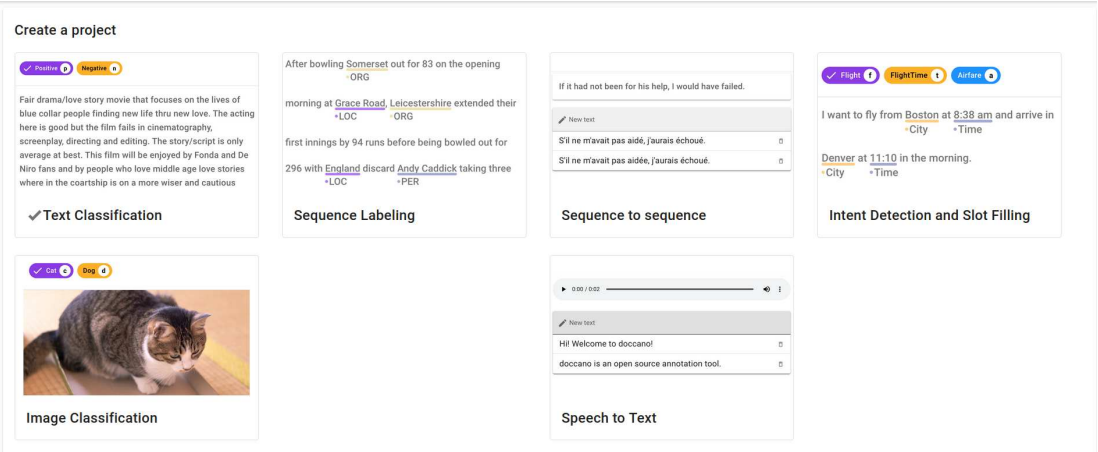
[1]Docker slika je *read-only* predložak koji sadržava instrukcije potrebne za pokretanje Docker kontejnera, tj. virtualnog okruženja potrebnog za rad programa

označavanje sekvenci i označavanje sekvencijskih odnosa. Za razliku od Label Studia nema integrirano definiranje novih pravila i sučelja označavanja podataka što znači u slučaju potrebe drukčijeg načina označavanja podataka potrebno ga je implementirati u samom kodu Doccana. Nadalje nudi puno slabiju mogućnost filtriranja i sortiranja podataka, što znači da je strukturiranost podataka koji se učitavaju u Doccano bitna.

Također omogućava rad na više projekata odjednom i istodoban rad korisnika u timskim projektima. Za razliku od Label Studia sadrži potpuno administrativno sučelje i sposobnost dodjeljivanja prava po ulogama, što omogućava puno veću sigurnost projekta nego kod Label Studia.

Isto dolazi uz vlastite Docker slike i predloške potrebne za pokretanje na servisima u oblaku, čime se omogućuje lagano započinjanje procesa označavanja podataka tekstualnog sadržaja. [22] [23]

Kako je Doccano projekt započet samostalno od strane Hiroki Nakayama, dok iza Label Studia stoji grupa Heartex, Doccano je projekt manje dozrelosti nego Label Studio.



Slika 2.7: Doccano se primarno usredotočio na označavanje podataka tekstualnog sadržaja.

2.3.3. Izbor alata

Implementaciju alata za označavanje tekstualnog sadržaja navođeno aktivnim učenjem moguće je napraviti u oba alata, ipak ovisno o namjeni projekta jedan ili drugi alat bit će pogodniji za korištenje.

Glavna prednost Label Studia je njegova puno veća dozrelost i drastično veće funkcionalnosti vezane uz personalizaciju formata skupova podataka i oznaka. S druge strane Doccano je specijaliziran kao alat za označavanje skupova podataka obrade prirodnog jezika što je ujedno i područje problema završnog rada. S te strane

odabir Doccana značio bi da imamo puno manje redundantnih funkcionalnosti. Još jedna prednost Doccana jest što zapravo podržava siguran timski rad preko kontrole pristupa temeljene na ulogama, dok Label Studio nema nikakve administrativne i sigurnosne funkcionalnosti u besplatnoj inačici.

Ipak kada sagledamo problem koji rješavamo općenita krutost Doccano zahtijeva puno veće nepotrebno pretprocesiranje podataka. S druge strane fleksibilnost Label Studia omogućava jako laganu implementaciju kontrole toka podatkovnog upita koja je nužna za ovaj rad.

Iako besplatna verzija Label Studia ima velike nedostatke vezane uz upravljanje timskih projekata, projekt napravljen u sklopu završnog rada nije zamišljen s timskim rad u vidu te nisu potrebne dodatne administrative mogućnosti koje nedostaju u Label Studiu.

Dodatna prednost Label Studia je njegova dokumentacija koja je puno opširnija, i time omogućava lakši razvoj personaliziranih projekata označavanja podataka.

Zbog ovih razloga u sklopu završnog rada koristiti ćemo Label Studio.

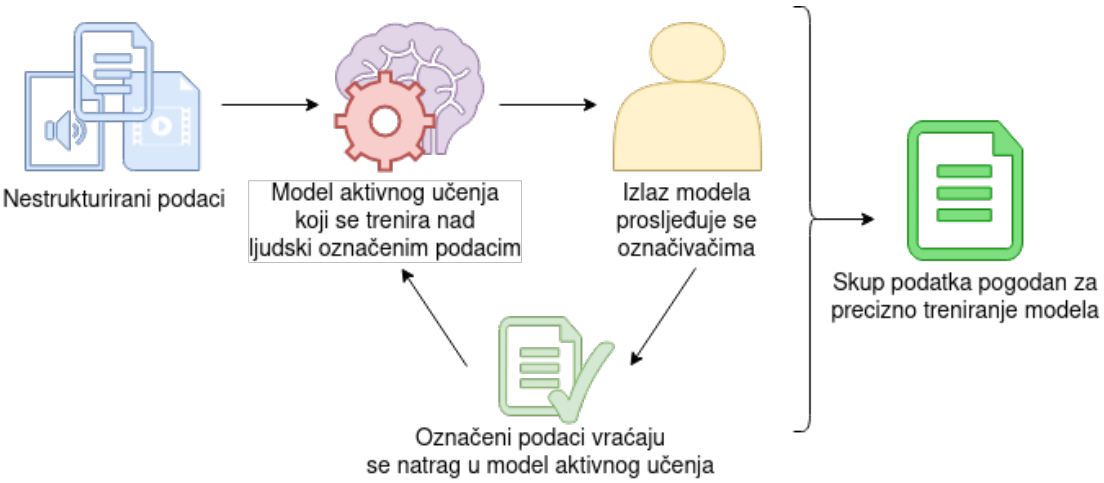
Ukoliko se podrazumijeva timski rad na alatu za označavanje podataka, pogotovo gdje su označivači osobe nevezane uz sam projekt modeliranja umjetne inteligencije, izbor bi bio između dodatne modifikacije koda Label Studia ili korištenje Doccana.

3. Aktivno učenje

Aktivno učenje je podskup polunadziranog učenja gdje sam model koji se trenira korisniku interaktivno šalje uzorke podataka na označavanje. Uzorci skupa podataka odabrani su optimalno prema određenom informacijskom svojstvu, a sami uzorak može biti označen, poslan na označavanje ili odbačen ovisno o algoritmu aktivnog učenja.

Koristi se u situacijama gdje postoji previše neoznačenih podataka, no ručno označivanje je prespor ili preskup proces. U takvim situacijama algoritam koji se trenira sam odabire koje će uzorke podataka slati označivaču. Osnovno vjerovanje je da će algoritam aktivnog učenja potencijalno dovesti do veće preciznosti modela drastično smanjujući pri tome potrebnu količinu označenih podataka. To je rezultat toga što u usporedbi s tradicionalnom pristupu modeliranja strojnog učenja, modelu sam bira koje podatke želi označiti.

Stoga, modelu aktivnog učenja dozvoljeno je interaktivno mijenjati uzorke za označavanje tijekom treniranja modela. Većinski to se odnosi na neoznačene podatke koje će kasnije ljudski označivač sam morati označiti. To čini aktivno učenje jedan od najuspješnijih primjeraka paradigme „čovjeka u petlji“ (*man-in-the-loop*).

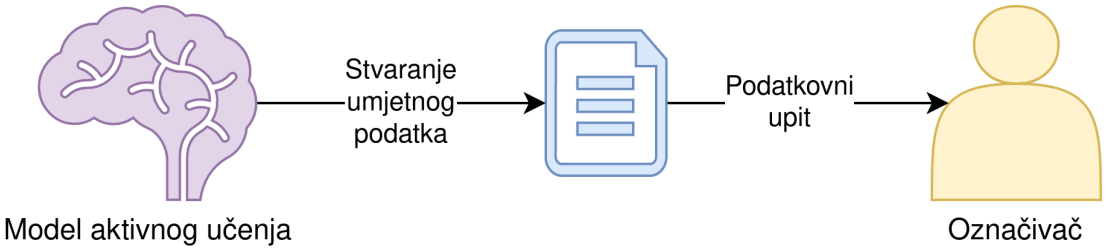


Slika 3.1: Pojednostavljeni dijagram proces aktivnog učenja.

U ovisnost o svojstvima rješavanog problema, budžetu i odluke o tome treba li označiti svaki mogući uzorak iz skupa podataka ili da li je dobit iz označavanja uzoraka veća nego trošak dobiti te informacije drukčije pristupamo problemu aktivnog učenja.

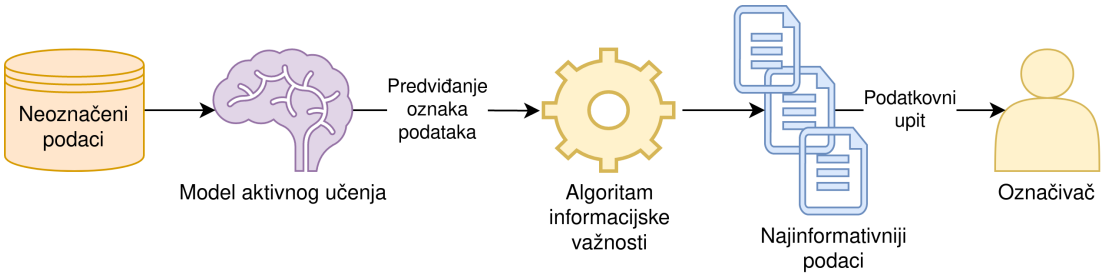
Postoje tri glavna pristupa aktivnom učenju:

1. Sinteza članskog podatkovnog upita - model na temelju postojanih podataka stvara umjetne podatke koje šalje označivaču na označavanje. Koristan samo u problemima gdje je lagano stvarati nove podatke iz postojećih. Primjer u slučaju klasifikacije brojeva to može biti broj koji je rotiran i/ili nedostaje dio tog broja.



Slika 3.2: Vizualna reprezentacija pristupa sinteze članskog podatkovnog upita. [11]

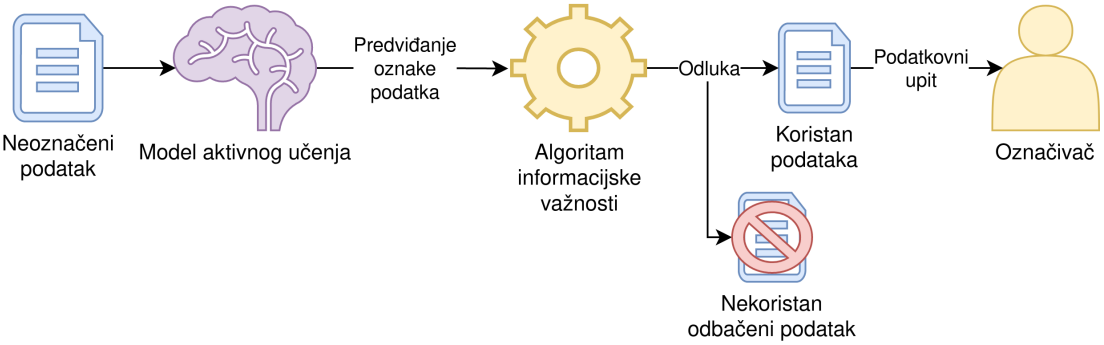
2. Uzorkovanje bazirano na bazenu - model ocjenjuje sve neoznačene podatke prema nekom kriteriju informacijske važnosti. U ovisnosti o tom kriteriju model šalje informacijski optimalne uzorke označivaču na označivanje. Model aktivnog učenja je prvo već treniran na manjem skupu označenih podataka, pomoću kojih se onda započinje proces odluke najkorisnijih podataka za sljedeću iteraciju i ponovnog treniranja modela. Iako je nedostatak ovog pristupa visoka memorijska složenost, i dalje je najpopularniji pristup aktivnom učenju.



Slika 3.3: Vizualna reprezentacija uzorkovanja baziranog na bazenu. [11]

3. Selektivno uzorkovanje bazirano na protoku - model pojedinačno prolazi kroz

neoznačene podatke te ovisno o njihovom informacijskom kriteriju i nepouzdanosti odlučuje podatak poslati na označavanje ili ignorirati.



Slika 3.4: Vizualna reprezentacija selektivnog uzorkovanja baziranog na protoku. [11]

3.1. Strategija podatkovnog upita

Općenito algoritme koji odlučuju sadržaj podatkovnog upita možemo podijeliti kategorije s obzirom na njihovu svrhu. Neke od kategorija su sljedeće:

1. Uravnotežavanje istraživanja i iskorištavanja[1] - na izbor podatkovnog upita gleda se kao na dilemu između istraživanja i iskorištavanja podatkovnog prostora. To postižemo modeliranjem aktivnog učenja kao *contextual bandit* problem.

2. Očekivana promjena modela - odabiremo podatke koji bi najviše promijenili trenutni model. Promjenu računamo kao razliku između parametara trenutnog modela i parametara modela nakon treniranja s povećanim skupom podataka. [8]

3. Očekivano smanjenje pogreške - odabiremo podatke koji bi najviše smanjili pogrešku generalizacije modela[2]. Jedna mogućnost je da pomoću Monte Carlo pristupa predviđamo očekivano smanjenje pogreške kao posljedicu označavanja upita. [27]

4. Potencirani gradijent istraživanja aktivnog učenja - sekvencijski algoritam koji može poboljšati rezultate drugih algoritama aktivnog učenja pomoću optimalnog nasumičnog istraživanja. [7]

[1] Također korištena u podržanom učenju
[2] Mjera koliko precizno algoritam može predvidjeti izlaznu vrijednost za dosad neviđeni podatak

5. Neizvjesnost uzorkovanja - odabiremo podatke za koje je trenutni model najviše nesiguran. Uobičajeno za mjeru nesigurnost se koristi entropija ili slične mjere probabilističke prirode.

6. Odbor upita - nad trenutno označenim podacima trenira se više različitih modela koji glasanjem odlučuju izlaz neoznačenih podataka. Kao podatkovni upit šaljemo podatke oko kojih se odbor najmanje složio.

7. Stvaranje upita iz raznolikih potprostora - kada koristimo model slučajnih šuma listovi mogu predstavljati potprostore (koje se preklapaju) originalnog prostora problema. U tom slučaju odabiremo podatke iz nepreklapajućih ili minimalno preklapajućih potprostora. [10]

8. Smanjenje varijance - odabiremo podatke koji bi minimalizirali varijancu izlaza modela

9. Konformni predviđači - koristi algoritam konformnog predviđanja koji mjeri nesigurnost previđanja. Algoritam se povodi tako da na temelju starih podataka predviđa da će novi podatci imati slične oznake uzimajući u obzir udaljenosti najbliže jednake i različite oznake. Dobivena vrijednost postupka se onda koristi u izračunu sigurnosti previđanja novih podataka. Cilj algoritma je minimizirati broj potrebnih podataka. [12]

10. Nepodudaranje prvo, najveća udaljenost - koristi algoritam temeljen na grupiranju i odborskom izboru upita koji se koristi u klasifikaciji zvučnih uzoraka . Algoritam započinje grupiranjem podataka nad kojima se onda koristi metoda nepodudaranje prvo, najveća udaljenost. Cilj ovog pristupa je optimizirati raznolikost izabranih podataka. [30]

11. Korisnički usmjerene strategije označavanja - implicitno se koristi u mnogim interaktivnim vizualizacijama i vizualno analitičkim pristupima kako bi se korisniku dala aktivna uloga u procesu učenja. Vizualna sučelja najčešće prikazuju podatke (ili prostor problema) i trenutno stanje modela koji se trenira koristeći smanjenje dimenzionalnosti u kombinaciji s raspršnim grafovima ili sličnim 2D vizualizacijama. Jednom vizualizirano, od korisnika se zahtijeva da izabere individualne uzorke i označi ih. [6]

Uspjeh aktivnog učenja ovisit će o izboru ispravne strategije podatkovnog upita za zadani problem [21]. Kao odgovor na problem ispravnog odabira strategije aktivnog

učenja razvijaju se algoritmi „meta-učenja“. Jedan primjer je traženje optimalne strategija podržanim učenjem tako što proces označavanja formaliziramo Markovljevim procesom odlučivanja [19].

3.2. Izbor pristupa i strategije aktivnog učenja

Pristup aktivnom učenju koji ćemo koristiti u rješavanju problema jest uzorkovanje bazirano na bazenu. Ovaj pristup rješava problem ogromnih skupova neoznačenih podataka. Postupak kreće od pretpostavke da postoji mali skup označenih podataka i veliki skup neoznačenih podataka. Upiti se vrše iz bazena koji se često pretpostavlja da je zatvoren (statičan, nepromjenjiv), iako nije nužno. Najčešće će podatci unutar upita odabiru pohlepnim algoritmom tako da se računa informacijska vrijednost svakog podatka unutar bazena. [28]

Uobičajeni postupak ovog pristupa je da koristimo bazen (neoznačeni skup podataka) i skup podataka za testiranje. Bazen se dodatno dijeli na skup za treniranje i skup za validaciju. Kako neoznačeni skup podataka zna biti ogromnih veličina često ga se razdjeljuje u više manjih bazena.

Postupak započinje odabirom k uzoraka iz bazena koje ćemo koristiti u skupu za treniranje dok ostatak ide u skup za validaciju. Model treniramo nad unijom označenog skupa podataka i skupa za treniranje, validiramo, a preciznost izračunavamo na skupu za testiranje.

U stanju validacije algoritam pokušava predvidjeti oznake skupa za validaciju te na temelju procjene oznaka izračuna informacijsku vrijednost tog uzorka. Informacijsku vrijednost računa na temelju jednog od algoritama strategija podatkovnog upita.

Zatim uspoređujemo informacijske vrijednosti unutar skupa za validaciju i izabiremo k novih uzoraka koje premještamo u skup za treniranje. Jednom kada su u skupu za treniranje od označivača tražimo da ih označe. Konačno, osvježimo naš skup za treniranje s novim informacijama.

Ovaj postupak ponavljamo sve dok nismo sretni s preciznošću ili ne ispunjavamo neki drugi kriterij.

Strategiju koju ćemo koristiti za podatkovni upit je neizvjesnost uzorkovanja koju ćemo računati pomoću Shannonovog modela entropije. Razlog zašto koristimo Shannovu entropiju je zbog poželjnih svojstava modela [24], neka od njih su:

1. Uniformne distribucije imaju najveću neizvjesnost
2. Neizvjesnost je aditivna za nezavisne događaje

3. Događaji s vjerojatnošću nula ne utječu na entropiju

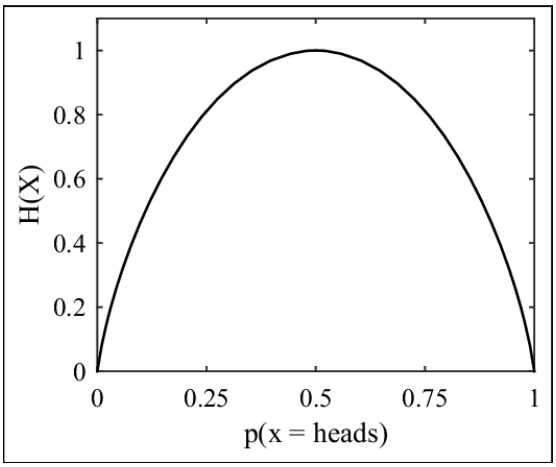
U teoriji informacija entropija na govori koliko je informacije sadržano u događaju. Što je događaj više vjerojatan ili deterministčan to će manje informacije sadržavati. Točnije, informacija je porast neizvjesnosti ili entropije.

Entropiju možemo izračunati preko sljedeće formule, gdje X označava skup vjerojatnosti čija je vrijednost između nula i jedan:

$$H(X) := -\sum_{x \in X} p(x) \log p(x) \tag{3.1}$$

S tim da je bitno da je suma ulaznih vjerojatnosti jednaka jedan, jer to omogućava preslikavanje entropije na vrijednosti između nula i jedan.

Veća entropija sugerira veću nesigurnost. Što znači da ćemo u svakom koraku učenja za svaki podatak tijekom validacije izračunati entropiju predviđenih vjerojatnosti oznaka. Iz skupa validacije odabrat ćemo podatke s najvećom entropijom, jer upravo nam to govori da su to podatci oko kojih je model najviše nesiguran.



Slika 3.5: Poznati primjer entropije bacanja novčića, izračunate u bitovima (baza logaritma je 2). Donja os prikazuje vjerojatnost kojom novčić pada na glavu, dok lijeva os prikazuje vrijednost entropije za određenu vjerojatnost glave. Entropija svoj maksimum od 1 postiže kada je vjerojatnost glave 50%, to jest kada je jednako vjerojatno da će novčić biti glava ili pismo. U slučaju kada je vjerojatnost glave 0% ili 100% već unaprijed znamo rezultat bacanja novčića te je vrijednost entropije 0. [31]

4. Problem

Cilj ovoga rada je prikazati implementaciju sustava za označavanje tekstualnog sadržaja navođeno aktivnim učenjem koja će označivačima teksta omogućiti lakši i brži rad na označavaju skupa podataka.

Obradu problema započet ćemo instaliranjem i podešavanjem alata za označavanje podataka. Opisat ćemo kako postići razna poželjna svojstva unutar alata i kako prilagoditi rad alata prema potrebama skupa podataka i prostora problema. Zatim ćemo prikazati rad tog alata u odabranim prilagođenim uvjetima nad nekim skupom podataka. Kao što je navedeno u Pododjeljak 2.3.3 alat koji ćemo koristiti u našoj implementaciji je Label Studio.

Potom ćemo odrediti nama najprikladniji oblik aktivnog učenja prema nekom kriteriju čiju ćemo implementaciju prikazati. Opisat ćemo postupak integracije takvog algoritma u sustav označavanja podataka i prikazati rezultate primjene algoritma nad skupom podataka. Metodu aktivnog učenja koju ćemo koristiti naveli smo i opisali u Odjeljak 3.2.

Konačno povezat ćemo prilagođeni alat iz prvog dijela i algoritam iz drugog dijela da bismo ostvarili sustav za označavanje tekstualnog sadržaja navođeno aktivnim učenjem. Komentirat ćemo moguće izmjene trenutnog stanja sustava i mogućnosti budućih nadogradnji.

Konkretni problem zbog kojeg označavamo skup podataka navođeno aktivnim učenjem je otkrivanje emocija u tekstualnom sadržaju. Problematika i oznake koje će se koristiti inspirirane su skupom podataka GoEmotions. Prije pojave GoEmotions većina razvoja NLP[1] u području emocija bile je usredotočena na uske domene, a same emocije su bile suviše općenite obuhvaćajući većinom samo 6 osnovnih emocija (ljutnja, iznenađenje, gađenje, sreća, strah i tuga) ili manje. Usporedno s prijašnjim iteracijama GoEmotions prepoznaje 27 različitih emocija plus dodatnu oznaku za

[1]NLP - Natural Language Processing ili obrada prirodnog jezika je grana umjetne inteligencije čija je svrha shvatiti ljudski jezik. Cilj je stvoriti modele koji mogu shvatiti, prevesti, provjeriti gramatiku teksta i slično.

neutralno stanje (nedostatak emocija). Obuhvaća 12 pozitivnih, 11 negativnih i 4 emocionalno višeznačnih emocija što omogućava otkrivanje i suptilnih promjena u emocijama. [9]

4.1. Implementacija alata za označavanje podataka

Nakon što sto odabrali koji alat koristiti prvo što je potrebno jest instalirati sam alat. U slučaju Label Studio to je moguće napraviti na više načina.

Label Studio podržava instalaciju preko pip-a[2], Dockera (Docker slika ili Docker Compose[3]), izvornog koda ili Anaconde[4]. No prije nego što počnemo instalirati Label Studio bitno je instalirati preduvjetnu bazu podataka koja može biti PostgreSQL ili SQLite.

Nakon instalacije i prije pokretanja alata bitno se je odlučiti koju bazu podataka koristiti. Ako namjeravamo koristiti ogromne skupove podataka (stotine tisuća podataka i više) ili očekivano puno korisnika u isto vrijeme predlaže se korištenje PostgreSQL baze podataka.

Dodatno u slučaju da se Label Studio instalirao preko Dockera, on po zatvaranju Label Studio neće spremiti učitane i označene podatke. Taj problem možemo riješiti preko Docker volumes, mehanizma Docker containera za perzistenciju podataka.

U slučaju da koristimo Docker sliku perzistenciju možemo omogućiti izmjenom naredbe CLI[5] kojom pokrećemo Label Studio:

```
docker run -it -p 8080:8080 -v <yourvolume>:/label-studio/data
    heartexlabs/label-studio:latest
```

Gdje <*yourvolume*> predstavlja Docker volume koji smo prethodno stvorili.

U slučaju da smo koristili Docker Compose potrebno je podesiti docker-compose.yml datoteku na sljedeći način:

[2]pip ili Package Installer for Python je preporučeni i najpopularniji upravitelj paketa za Python

[3]Docker Compose je alat za definiranje i pokretanje Docker aplikacija koje sadrže više kontejnera

[4]Anaconda je distribucije Pythona i R programskog jezika namijenjena za znanstvene svrhe koja pokušava olakšati upravljanje i razvijanje paketa

[5]CLI - Command Line Interface ili sučelje naredbenog retka

```yaml
version: "3.3"
services:
  label_studio:
    image: heartexlabs/label-studio:latest
    container_name: label_studio
    ports:
      - 8080:8080
    volumes:
      - ./mydata:/label-studio/data

volumes:
  mydata:
```

Gdje je opet potrebno odabrati Docker volume u koji smo odlučili spremati podatke.

U slučaju da se želi dodatno prilagoditi Label Studio, sve integrirane mogućnosti prilagodbe alata dostupne su na stranici dokumentacije [14].

Skup podataka koji je odabran za rad na projektu je skup citata na engleskom. Citati su prikupljeni sa Goodreads Quotes, gdje sami podatci sadrže citat na engleskom i oznake pojmova kojima je citat blizak. Skup podataka spremljen je u JSON[6] formatu. Skup podataka je napravila i objavila Abir Eltaief. [13]

Kako želimo imati neoznačeni skup podataka koji ćemo na temelju GoEmtions modela označavati koristeći aktivno učenje, potrebno je prilagoditi skup podataka tako da zadržimo samo tekst podataka. To možemo napraviti preko sljedeće Python skripte.

```python
import pandas as pd


data = pd.read_csv('quotes.csv', usecols=['quote'])
mask = (data['quote'].str.len() < 512)
data = data.loc[mask]
data.to_csv('quotes_text_only_less_than_512.csv', index=False,
    header=True, encoding='utf8')
# index i header ovisno po potrebama
```

Također smo se ograničili samo na citate čija je duljina manja od 512 znakova kako ne bismo prešli maksimalno veličinu ulaza modela.

Nakon što imamo pogodni skup podataka možemo započeti sa stvaranjem projekta. To radimo tako što se prijavimo u podignuti sustav Label Studia te odaberemo opciju stvaranja novog projekta. Nakon što učitamo novouređeni skup podataka moramo

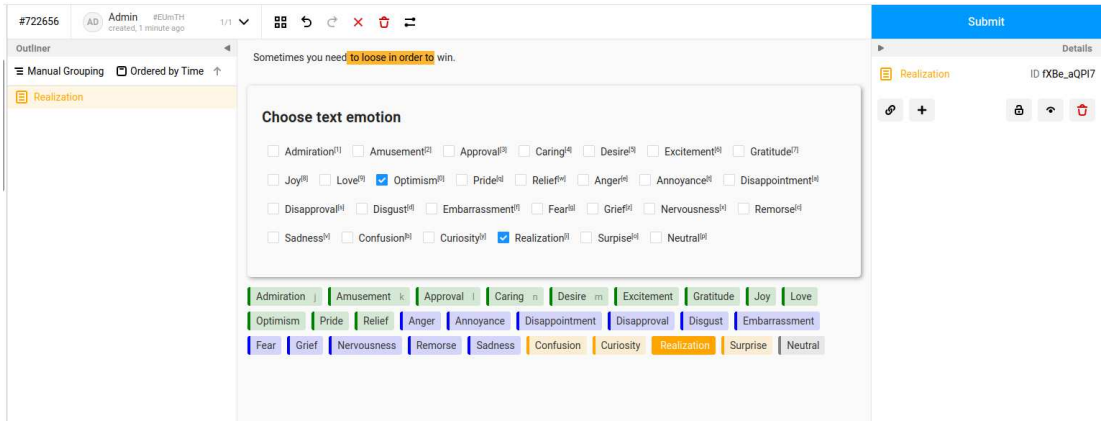
[6]JSON - JavaScript Object Notation je otvoreni format zapisa i razmjene informacija

izabrati pravila označavanja podataka.

Kako želimo postići mogućnost označavanja 28 emocija preko više izborne klasifikacije i sekvenci moramo stvoriti vlastito sučelje označavanja podataka. Stvaranje vlastitog sučelja olakšano je time što nam Label Studio nudi izbor gotovih često korištenih komponenti, a njihovo korištenje opisano je u dokumentaciji Label Studia. Sučelje potrebno za naš problem možemo postići sljedećim kodnim isječkom:

```
<View>
  <Text name="text" value="$quote"/>
  <View style="box-shadow: 2px 2px 5px #999; padding: 20px; margin-
   top: 2em; border-radius: 5px;">
    <Header value="Choose text emotion"/>
    <Choices name="emotion" toName="text" choice="multiple"
   showInLine="true">
      <Choice value="Admiration"/>
      <Choice value="Amusement"/>
      <Choice value="Approval"/>
      ...
      <Choice value="Realization"/>
      <Choice value="Surpise"/>
      <Choice value="Neutral"/>
    </Choices>
  </View>
  <View>
    <Labels name="emotion_span" toName="text">
      <Label value="Admiration" background="green"/>
      <Label value="Amusement" background="green"/>
      <Label value="Approval" background="green"/>
      ...
      <Label value="Realization" background="orange"/>
      <Label value="Surprise" background="orange"/>
      <Label value="Neutral" background="gray"/>
    </Labels>
  </View>
</View>
```

Potpuni kod sučelja za označavanje podataka dostupan je na Kodni isječak A.1 Ukoliko se sve ispravno izvršilo odabirom označavanja određenog podataka unutar skupa podataka trebali bi imati sljedeći prikaz.



Slika 4.1: Primjer jednog označenog citata. Vidimo da je podatke moguće klasificirati preko potvrdnih gumbova, ali i naglasiti dio teksta označivanjem sekvence.

Sada odabirom određenog potvrdnog gumba kategorije možemo klasificirati podatak po emocijama, a odabirom gumba za označivanje sekvenci možemo omogućiti označavanje dijela teksta kao područje u kojem postoji veći intenzitet određene emocije. Ova pravila moguće je naknadno izmijeniti izmjenjivanjem pravila označavanja koja smo unijeli tijekom stvaranja projekta.

Ako u slučaju kompleksnih podataka želimo dodatno navođenje tako što označivačima prije početka označavanja prikazujemo upute kako pravilno podatke označavati, možemo ih dodati preko postavki projekta.

Ovim postupkom prikazali smo kako koristiti Label Studio za ručno označavanje podataka. Problem ručnog označavanja je sporost procesa, ukoliko ovaj proces želimo ubrzati to možemo napraviti pomoću aktivnog učenja.

4.2. Implementacija aktivnog učenja nad skupom podataka

Nakon što je uspostavljen rad alata za označavanje podataka potrebno je optimizirati označavanje integracijom aktivnog učenja u proces označavanja. Strategija i pristup aktivnom učenju opisani su već u Odjeljak 3.2.

Za uzorkovanje bazirano na bazenu potrebni su nam neoznačeni i označeni skup podataka. Skup neoznačenih podataka već smo odabrali u poglavlju Odjeljak 4.1, tako da ostajem problem označenog skupa podataka. Kako već koristimo iste oznake kao skup podataka GoEmotions, za skup označenih podataka koristit ćemo GoEmotions.

Kako zadatak problema nije samostalno stvoriti model strojnog učenja, nego

prikazati proces označavanja podataka podržan aktivnim učenjem, da bi dodatno ubrzali proces umjesto treniranja vlastitog modela koristit ćemo model već treniran nad GoEmotions javno dostupan preko Hugging Face[7] platforme. [25]

Sljedeće je potrebno ostvariti proces određivanja podatkovnog upita. Pomoću treniranog modela izračunat ćemo emocije neoznačenih podataka, a onda na temelju izračunatih emocija, izračunati ćemo entropiju podataka. To je potrebno napraviti za svaki podatak unutar bazena. Nakon izračuna entropije odabrat ćemo k uzoraka najveće entropije koje ćemo poslati označivačima na označavanje.

Proces određivanja entropije možemo provesti pomoću sljedeće Python skripte. Za pokretanje skripte potrebni su paketi pandas[8], PyTorch[9] i transformers[10]. Također treba imati u umu da ćemo pokretanjem ove skripte cijeli model preuzeti lokalno na računalo.

```
import math
import pandas as pd
from transformers import pipeline


# odabrati model s Hugging Face pomocu sljedece sintakse:
# "profil/ime_modela"
MODEL_NAME = "monologg/bert-base-cased-goemotions-original"

pipe = pipeline(model = MODEL_NAME, return_all_scores = True)

def entropy(row):
    values = pipe(row['quote'])[0]
    values = [item['score'] for item in values]
    # prilagoditi izlaz modela za izracun entropije
    return = -sum([x*math.log(x,2) for x in values])
    # entropija

df = pd.read_csv('quotes_text_only_less_than_512.csv')
df['entropy'] = df.apply(lambda row: entropy(row), axis=1)
df.to_csv('quotes_entropy.csv', index=False, header=True, encoding='
    utf8')
```

U navedenom kodnom isječku pomoću *pipeline* objekta dohvaćamo parove

[7]Hugging Face je platforma koja omogućava korisnicima da razvijaju, treniranju i dijele modele i skupove podataka po principu otvorenog koda.

[8]pandas je paket koji pruža brzu i fleksibilnu analizu i manipulaciju podataka

[9]PyTorch je radni okvir otvorenog izvora za strojno učenje baziran na paketu Torch i Pythonu.

[10]Transformers je paket koji pruža API za jednostavno korištenje Hugging Face modela

emocije-vjerojatnost za pojedini citat, iz kojih onda izvlačimo samo vjerojatnosti. Na temelju vjerojatnosti emocija izračunat ćemo entropiju podatka. Ukratko, pročitat ćemo CSV datoteku kao tablicu gdje ćemo nad svakim redom (citatom) pozvati funkciju koja izračuna i vraća entropiju teksta koju spremamo u novi stupac tablice. Zatim ćemo novu tablicu spremiti u novu CSV datoteku.

Ako je proces izračuna entropije uspješno proveden izlazna CSV datoteka bi trebala uz svaki citati imati i vrijednost entropije tog citata. Primjer ispravnog sadržaja je:

	A	B
1	quote	entropy
2	These people you used to see every day, friends or acquaintances, after a while they	0.41243054
3	In a Pyongyang restaurant, don't ever ask for a doggie bag.	0.913048999
4	The best thing to do is stare it in the face and move on. We have to face our fears and	2.458672816
5	Magic has its own weight, and that weight, the gravity of it, is pulling the fabric of reality	0.409545504
6	I've been a sports fan all my life, and like most other actors, I'm convinced I could hav	2.699480524
7	The love of nature is the love for the Creator.	1.034730606
8	At daybreak on the first day, thousands of Cambodians are already calmly waiting out	2.980903321
9	To act upon the outer world and change it, you must be first be fine to act upon yourse	0.888638196
10	Jiu Jitsu is meant to serve us, not the other way around. It is meant to make you more	3.651791365
11	When we feel used and abused, the Bible says we are loved. When we feel abandon	2.463898049
12	Might and wrong combined, like iron magnetized, are endowed with irresistible attracti	0.670189399
13	How do you wish to live in lifetime existence; envying, hating and loving? It is better to	2.73491287
14	Our being is continually undergoing and entering upon changes. ... We must strictly s	2.631208942
15	History is filled with tragic examples of wars that result from diplomatic impasse. Whetl	3.624916367
16	Sometimes you need to loose in order to win.	0.36579064
17	There are a number of paths that lead to this place. I have been avoiding them for son	1.92103917
18	We think there are limits to the dimensions of fear. Until we encounter the unknown. T	1.844418114
19	Although I am basically self taught, I consider Debussy my teacher - the most importa	2.071062855
20	When God blesses you, the highest form of gratitude is open-handedness.	1.939215788
21	All endeavors, heroic or monstrous, scientific or philosophical, rise from the electroche	0.963478751
22	I also like men who like dogs. I couldn't date a man who doesn't like my dog.	2.337295677

Slika 4.2: Izgled nekoliko redaka CSV datoteke koja uz citate sadrži i vrijednosti entropije.

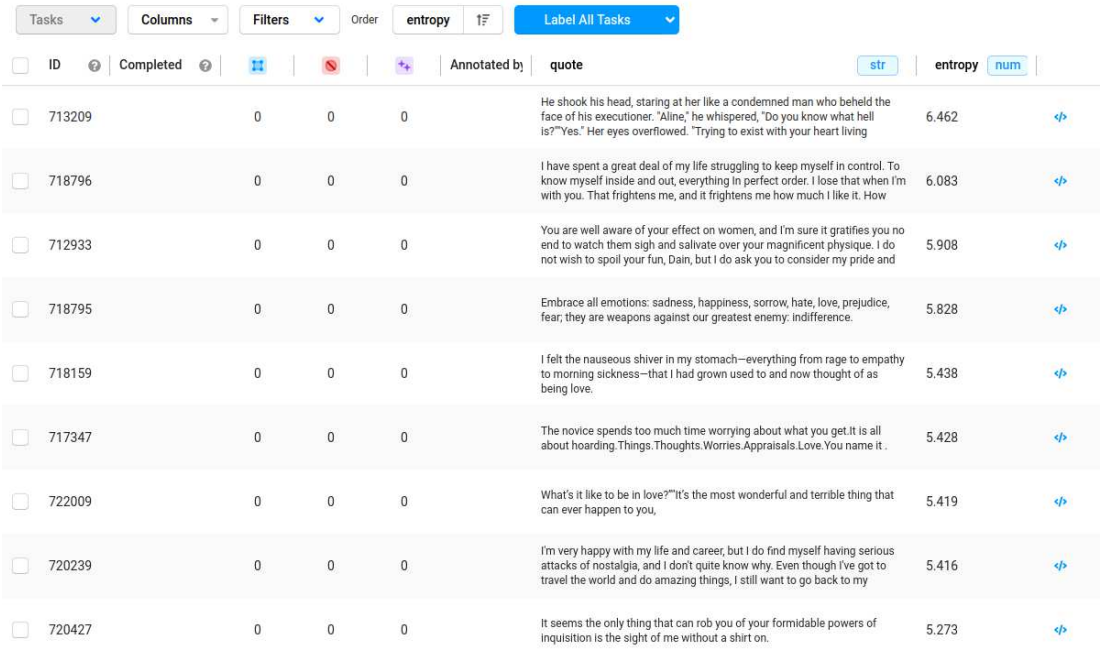
4.3. Označavanje podataka navođeno aktivnim učenjem

Nakon što smo implementirali strategiju podatkovnog upita, moguće je započeti proces označavanja podataka navođenog aktivnim učenjem. U ovom ćemo odjeljku ručno ćemo pokazati jednu iteraciju procesa aktivnog učenja.

Prateći upute iz Odjeljak 4.1 opet ćemo stvoriti podataka s identičnim pravilima i sučeljem označavanja podataka, samo ćemo umjesto neoznačenih podataka učitati CSV koja osim citata sadrži i njihovu entropiju.

Nakon što smo učitali datoteku promijenit ćemo tip stupca „entropy“ u „num“ te

odabrati filtriranje po entropiji od najveće vrijednosti prema manjima. Iz toga bi trebali dobiti sljedeći prikaz:



Slika 4.3: Na slici vidimo silazno soritrane podatke prema entropiji. Ovi podaci predstavljaju podatke oko kojih je naš model bio najviše nesiguran.

Sada kada imamo skup podataka silazno sortiran prema entropiji, ukoliko bismo redom išli označavati uzorke postigli bismo jako jednostavan postupak označavanja podataka navođen aktivnim učenjem.

Sljedeći korak u iteraciji aktivnog učenja bio bi označavanje k podataka koje ćemo iz skupa validacije prebaciti u skup za treniranje. Nakon što smo označili potreban broj podataka i prebacili ih u skup za treniranje, započinjemo opet s procesom treniranja modela čime cijeli proces počinje ispočetka. Što znači ponovno računanje entropije skupa validacije preko koje odabiremo novi podatkovni upit.

Iako vrlo jednostavno u implementaciji, dohvaćanje uzoraka prema najvećoj entropiji umjesto nasumičnog izbora u prosjeku poboljšava kvalitetu učenog modela [29]. Nedostatak prikazanog procesa je jako sporo računanje entropije validacijskog skupa podataka i manjak automatiziranosti procesa, to jest potreba da čovjek samostalno prebacuje najinformativnije podatke u skup za treniranje, pokrene proces treniranja i ponovno pokrene proces računanja entropije validacijskog skupa za svaku iteraciju aktivnog učenja.

No ipak, upravo tim spajanjem strojnog učenja s odabirom uzoraka sljedećeg podatkovnog upita nastaje proces aktivnog učenja koji smo i implementirali.

4.4. Moguće izmjene i nadogradnje

Ukoliko se želi zadržati ista strategija i pristup aktivnom učenju moguće je riješiti probleme navedene u prošlom odjeljku. Jedna od glavnih mana prikazane implementacije je nedostatak automatizacije procesa. Automatizaciju možemo vrlo lako implementirati, a moguće je automatizirati svaki dio procesa osim samog označavanja podataka.

Dodatna nadogradnja bila bi omogućiti paralelni rad unutar procesa izračunavanja entropije. Naime trenutni pristup kod velikih skupova podataka zahtijeva nekoliko sati za izračun entropije, a ako bi radili sa skupovima podataka od nekoliko milijuna uzoraka onda bi potrebno vrijeme bilo mjereno u danima. Brži izračun omogućio bi brže iteriranje procesa aktivnog učenja. Brže iteracije uz manji k ili broj podataka koje označujemo unutar iteracije dovele bi do veće ažurnosti izračunate entropije što bi opet dovelo do boljeg treniranja modela.

5. Zaključak

Cilj ovog rada bio je ostvariti sustav za označavanje podataka tekstualnog sadržaja podržano aktivnim učenjem koje bi označivačima podataka bilo dostupno kroz web sučelje.

U svrhu ostvarenja zadatka istražili smo važnost označavanja podataka u cjelokupnom procesu stvaranja modela strojnog učenja. Analizirali smo povezanost podataka i strojnog učenja te prikazali razlike u pristup podacima paradigmi strojnog učenja. Iz tih zaključaka i cilja rada odabrali smo nama relevantnu paradigmu te smo u kontekstu te paradigme istražili poželjna svojstva skupova podataka koja želimo ostvariti.

Za potrebe rada istražili smo i analizirali javno dostupne alate za označavanje podataka koji zadovoljavaju potrebe problema. Analizom i argumentacijom njihovih prednosti i mana odabrali smo alat koji ćemo koristiti u radu te smo predložili okruženja koje bi dovele do korištenja drugih alata.

Zatim smo istražili područje aktivnog učenja gdje smo opisali postojeće pristupe aktivnom učenju i često korištene strategije podatkovnog upita. U kontekstu problema odabrali smo nama povoljan pristup i strategiju gdje smo dodatno pojasnili njihova svojstva i opisali postupke njihovog provođenja.

Potom smo krenuli s implementacijom samog web sučelja za označavanje podataka pomoću odabranog alata gdje smo prikazali kako ostvariti neka dodatna poželjna svojstva alata i kako prilagoditi samo sučelje alata za potrebe određenog problema. Nakon toga smo odabrali i opisali resurse potrebne za aktivno učenje (model, skup označenih i neoznačenih podataka) i prikazali kako pomoću njih ostvariti strategiju podatkovnog upita za odabrani pristup aktivnog učenja. Konačno smo povezali prethodno implementirane komponente procesa aktivnog učenja te na primjeru pokazali kako izgleda jedna iteracija u procesu aktivnog učenja. Na samom kraju raspravljali smo o nedostacima trenutnog rješenja te o mogućim izmjenama i nadogradnjama trenutnog sustava.

LITERATURA

[1] Salem Alelyani, Huan Liu, i Lei Wang. The effect of the characteristics of the dataset on the selection stability. U *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, stranice 970–977. IEEE, 2011.

[2] Aida Ali, Siti Mariyam Shamsuddin, i Anca L Ralescu. Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3), 2013.

[3] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[4] Jason Baldridge i Alexis Palmer. How well does active learning actually work? time-based evaluation of cost-reduction strategies for language documentation. U *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[5] Ganesh Baliga, Sanjay Jain, i Arun Sharma. Learning from multiple sources of inaccurate data. *SIAM Journal on Computing*, 1997.

[6] Jürgen Bernard, Matthias Zeppelzauer, Markus Lehmann, Martin Müller, i Michael Sedlmair. Towards user-centered active learning algorithms. U *Computer Graphics Forum*. Wiley Online Library, 2018.

[7] Djallel Bouneffouf. Exponentiated gradient exploration for active learning. *Computers*, 5(1):1, 2016.

[8] Wenbin Cai, Ya Zhang, i Jun Zhou. Maximizing expected model change for active learning in regression. U *2013 IEEE 13th international conference on data mining*. IEEE, 2013.

[9] Jeongwoo Ko Dana Alon. Goemotions: A dataset for fine-grained emotion classification, 2021. URL `https://ai.googleblog.com/2021/10/goemotions-dataset-for-fine-grained.html`.

[10] Shubhomoy Das. Active anomaly discovery. https://github.com/shubhomoydas/ad_examples, 2018. [Online; accessed 19-Sep-2018].

[11] Datacamp. Active learning: Curious ai algorithms. URL https://www.datacamp.com/tutorial/active-learning.

[12] Leo Dreyfus-Schmidt. Measuring models' uncertainty: Conformal prediction, 2020. URL https://blog.dataiku.com/measuring-models-uncertainty-conformal-prediction.

[13] Abir Eltaief. Dataset card for english quotes, 2021. URL https://huggingface.co/datasets/Abirate/english_quotes.

[14] Heartex. Get started with label studio, 2019. URL https://labelstud.io/guide/index.html#Quick-start.

[15] Heartex. What is label studio?, 2019. URL https://github.com/heartexlabs/label-studio.

[16] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, i Vitobha Munigala. Overview and importance of data quality for machine learning tasks. U *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[17] Justin M Johnson i Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 2019.

[18] Gozde Karatas, Onder Demir, i Ozgur Koray Sahingoz. Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset. *IEEE Access*, 2020.

[19] Ksenia Konyushkova, Raphael Sznitman, i Pascal Fua. Discovering general-purpose active learning strategies. *arXiv preprint arXiv:1810.04114*, 2018.

[20] Akash Kumar. What is supervised learning and its different types?, 2020. URL https://www.edureka.co/blog/supervised-learning/.

[21] Punit Kumar i Atul Gupta. Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology*, 2020.

[22] Hiroki Nakayama. Get started with doccano, 2018. URL https://doccano.github.io/doccano/.

[23] Hiroki Nakayama. Doccano github, 2018. URL https://github.com/doccano/doccano.

[24] Alireza Namdari i Zhaojun Li. A review of entropy measures for uncertainty quantification of stochastic processes. *Advances in Mechanical Engineering*, 11 (6):1687814019857350, 2019.

[25] Jangwon Park. monologg/bert-base-cased-goemotions-original, 2021. URL https://huggingface.co/monologg/bert-base-cased-goemotions-original.

[26] Cognilytica Research. Data Engineering, Preparation, and Labeling for AI 2019. Technical report, Cognilytica Research, 01 2019.

[27] Nicholas Roy i Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2001.

[28] Burr Settles. Active learning literature survey. 2009.

[29] H Sebastian Seung, Manfred Opper, i Haim Sompolinsky. Query by committee. U *Proceedings of the fifth annual workshop on Computational learning theory*, 1992.

[30] Zhao Shuyang, Toni Heittola, i Tuomas Virtanen. An active learning method using clustering and committee-based sample selection for sound event classification. U *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018.

[31] Huy Tran, Jean Domercant, i Dimitri Mavris. Evaluating the agility of adaptive command and control networks from a cyber complex adaptive systems perspective. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 12, 07 2015. doi: 10.1177/1548512915592517.

Dodatak A
Potpuni isječci korištenog koda

A.1. Kod za stvaranje sučelja za označavanje podataka unutar alata Label Studio

```
<View>
  <Text name="text" value="$quote"/>
  <View style="box-shadow: 2px 2px 5px #999; padding: 20px; margin-top: 2em; border-radius: 5px;">
    <Header value="Choose text emotion"/>
    <Choices name="emotion" toName="text" choice="multiple" showInLine="true">
      <Choice value="Admiration"/>
      <Choice value="Amusement"/>
      <Choice value="Approval"/>
      <Choice value="Caring"/>
      <Choice value="Desire"/>
      <Choice value="Excitement"/>
      <Choice value="Gratitude"/>
      <Choice value="Joy"/>
      <Choice value="Love"/>
      <Choice value="Optimism"/>
      <Choice value="Pride"/>
      <Choice value="Relief"/>
      <Choice value="Anger"/>
      <Choice value="Annoyance"/>
      <Choice value="Disappointment"/>
      <Choice value="Disapproval"/>
      <Choice value="Disgust"/>
      <Choice value="Embarrassment"/>
      <Choice value="Fear"/>
```

```
      <Choice value="Grief"/>
      <Choice value="Nervousness"/>
      <Choice value="Remorse"/>
      <Choice value="Sadness"/>
      <Choice value="Confusion"/>
      <Choice value="Curiosity"/>
      <Choice value="Realization"/>
      <Choice value="Surpise"/>
      <Choice value="Neutral"/>
    </Choices>
  </View>
  <View>
    <Labels name="emotion_span" toName="text">
      <Label value="Admiration" background="green"/>
      <Label value="Amusement" background="green"/>
      <Label value="Approval" background="green"/>
      <Label value="Caring" background="green"/>
      <Label value="Desire" background="green"/>
      <Label value="Excitement" background="green"/>
      <Label value="Gratitude" background="green"/>
      <Label value="Joy" background="green"/>
      <Label value="Love" background="green"/>
      <Label value="Optimism" background="green"/>
      <Label value="Pride" background="green"/>
      <Label value="Relief" background="green"/>
      <Label value="Anger" background="blue"/>
      <Label value="Annoyance" background="blue"/>
      <Label value="Disappointment" background="blue"/>
      <Label value="Disapproval" background="blue"/>
      <Label value="Disgust" background="blue"/>
      <Label value="Embarrassment" background="blue"/>
      <Label value="Fear" background="blue"/>
      <Label value="Grief" background="blue"/>
      <Label value="Nervousness" background="blue"/>
      <Label value="Remorse" background="blue"/>
      <Label value="Sadness" background="blue"/>
      <Label value="Confusion" background="orange"/>
      <Label value="Curiosity" background="orange"/>
      <Label value="Realization" background="orange"/>
      <Label value="Surprise" background="orange"/>
      <Label value="Neutral" background="gray"/>
    </Labels>
  </View>
```

```
68 </View>
```

Sustav za označavanje tekstualnog sadržaja navođeno aktivnim učenjem

Sažetak

Strojno učenje je grana umjetne inteligencije čiji se uspjeh uveliko zasniva na obrađenim podacima. Upravo zato izbor kvalitetnih podataka ima važnu ulogu u samom procesu stvaranja modela strojnog učenja. Jedan od načina kontrole toka obrađenih podataka je aktivno učenje - potkategorija polunadziranog učenja koje se koristi kako bi se povećala uzorkovanu učinkovitost tijekom procesa treniranja modela. U ovom radu istražene su ovisnosti između skupa podataka i modela strojnog učenja. Zatim su istraženi pristupi aktivnom učenju i strategije podatkovnog upita. Na temelju istraženih spoznaja prikazuje se postupak implementacije sustava za označavanje tekstualnog sadržaja navođeno aktivnim učenjem koje svoje usluga pruža preko web sučelja.

Ključne riječi: aktivno učenje, strategije podatkovnog upita, skupovi podataka, polunadzirano učenje, označavanje podataka.

System for textual data labeling guided by active learning

Abstract

Machine learning is a subdiscipline in the field of artificial intelligence whose success heavily relies on previously processed data. The quality of afore-mentioned data therefore plays a significant role in the process of model creation. This discipline can be further dissected into categories, one of which is active learning - a special case of semi-supervised learning used for improving sampling efficiency. This science paper explores important relations connecting data sets to machine learning models and vice versa. It further examines various approaches to active learning, as well as query strategies. Finally, the paper showcases the implementation process of a text labeling system that relies on active learning and provides a web service through a simple interface.

Keywords: active learning, query strategies, datasets, semi-supervised learning, data labeling.