UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
Katedra za računarstvo

IZBOR INSTANCI PODATAKA (INSTANCE SELECTION)

- PRIKUPLJANJE I PREDOBRADA PODATAKA -

Profesor: Prof. Dr Aleksandar Stanimirović **Student:** Aleksa Milić 1610

Niš, 2024.

Sadržaj

Uvod

Prikupljanje i priprema podataka predstavljaju ključne korake u mašinskom učenju, od kojih zavisi uspeh svakog projekta. Neophodno je da podaci budu što čistiji, bez prisustva šuma, nedostajućih vrednosti ili duplikata. Ovakva predobrada poboljšava kvalitet rezultata, optimizuje performanse algoritama i smanjuje opterećenje hardverskih resursa potrebnih za obradu.

Jedan od izazova u predobradi podataka je velika količina informacija koje treba obraditi. Podaci se često nalaze u tabelarnom obliku, gde svaka kolona predstavlja neki atribut (eng. Feature), dok svaki red (eng. Instance) sadrži konkretne vrednosti tih atributa. Algoritmi mašinskog učenja koriste ovakve tabele za treniranje modela, koji zatim na osnovu poznatih vrednosti atributa pokušavaju da predvide nepoznate vrednosti za nove instance.

Brzina izvođenja ovih algoritama direktno zavisi od količine podataka koja se obrađuje. Sa konstantnim rastom količine podataka na internetu, raste i broj instanci u dataset-ovima, što postavlja zahtev za efikasnim rešenjima koja mogu obraditi velike količine podataka i istovremeno pružiti tačne rezultate. Jedan od načina za rešavanje ovog problema je korišćenje Big Data tehnologija, gde se ogromni skupovi podataka (reda veličine GB, TB ili više) smeštaju na udaljene servere sa praktično neograničenim hardverskim resursima.

Alternativno, može se primeniti pristup smanjenja količine podataka (eng. Data reduction) unutar samog skupa podataka. To podrazumeva uklanjanje atributa i instanci koji nisu značajni za analizu (eng. Knowledge discovery), uz očuvanje ili čak poboljšanje performansi modela u poređenju sa modelom treniranim na kompletnom skupu podataka. Za ovu svrhu razvijeno je nekoliko metoda [1]:

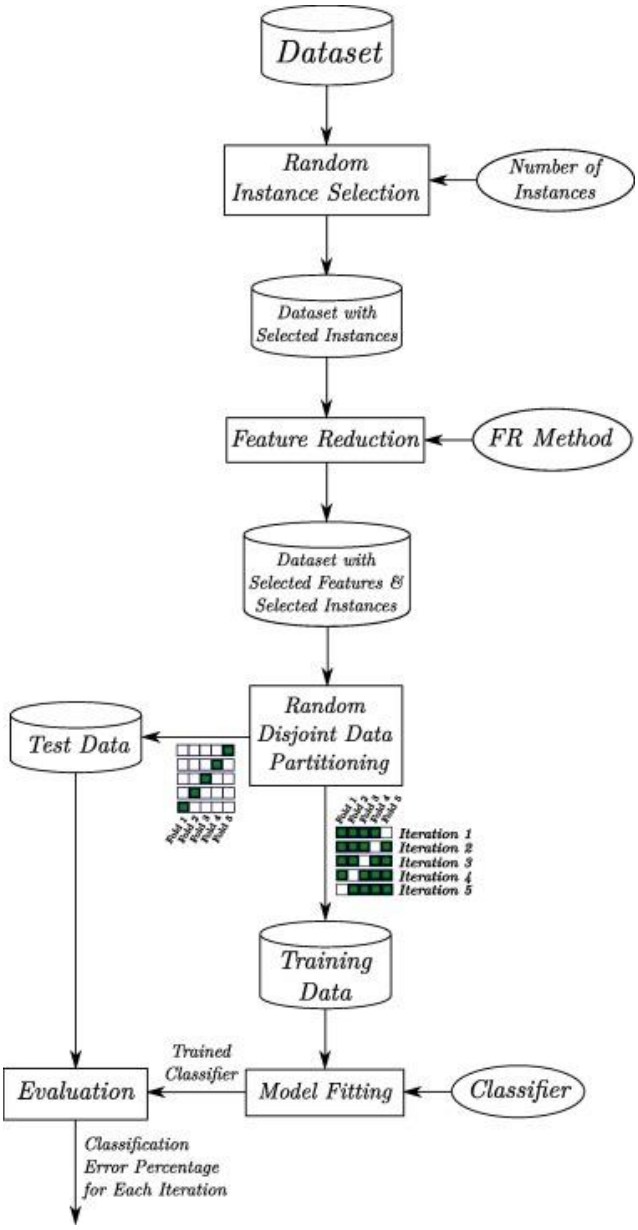
- **Ekstrakcija karakteristika (eng. Feature extraction)** - obuhvata različite manipulacije atributima, kao što su uklanjanje jednog ili više atributa, kombinovanje više atributa u jedan, ili stvaranje novih, sintetičkih atributa.

- **Izbor instanci (eng. Instance selection)** - ovaj metod se fokusira na identifikaciju najreprezentativnijeg podskupa originalnog skupa podataka, bez ugrožavanja prediktivnih sposobnosti modela. Cilj je da algoritmi trenirani na originalnom skupu podataka i na selektovanom podskupu daju slične rezultate. Izbor instanci može se smatrati posebnom vrstom generisanja instanci, ali sa ograničenjem da nove instance moraju biti deo originalnog skupa.

- **Diskretizacija (eng. Discretization)** - predstavlja tehniku transformacije numeričkih podataka u diskretne kategorije. Na primer, starost osobe može se klasifikovati u grupe poput "mlad", "srednjih godina" i "stariji". Ključni izazov u ovom procesu je pronalaženje optimalnih granica ili intervala za ove kategorije.

- **Generisanje instanci (eng. Instance generation)** - u ovom procesu se kreiraju nove, veštačke instance podataka koje služe kao sažetak originalnih podataka. Cilj je povećati reprezentativnost skupa podataka, a istovremeno smanjiti njegovu veličinu.

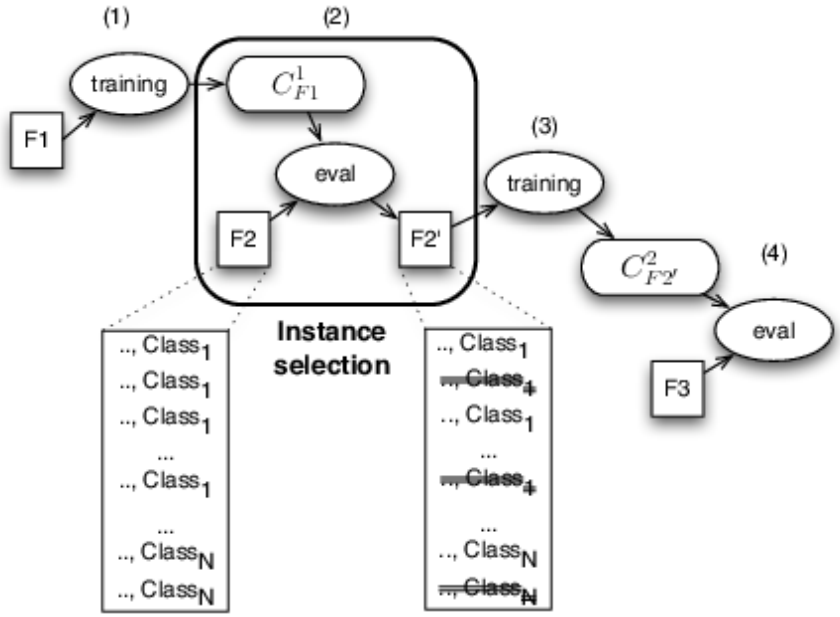
- **Izbor atributa (eng. Feature selection)** - odnosi se na proces selekcije relevantnih atributa iz većeg skupa za korišćenje u treniranju modela. Ovaj pristup pomaže u smanjenju dimenzionalnosti podataka i uklanjanju irelevantnih informacija, čime se poboljšava performans modela.



Slika 1. Proces selekcije instanci i redukcije karakteristika u masinskom ucenju

Izbor instanci podataka (Instance selection)

Kao što je prethodno objašnjeno, algoritmi za odabir instance imaju za cilj da smanje složenost algoritama za učenje smanjenjem broja primera skupova. Svrha ovih algoritama je da izdvoje najznačajniji podskup instanci odbacivanjem onih koji ne pružaju vredne informacije. Slika 2. ilustruje proces odabira instance. Smanjenje skupa podataka ima tri glavne prednosti: smanjuje i prostorne zahteve sistema i vreme obrade zadataka učenja, ali i uklanjanje šuma i suvišnih instanci [2].



Slika 2. Proces selekcije instanci

Odabrani skup instanci se može koristiti za obuku bilo koje vrste algoritama, ali, tradicionalno, mnogi algoritmi za odabir instanci su razvijeni za klasifikator k najbližih suseda, ili skraćeno kNN. Iz tog razloga, termin koji se koristi za proces selekcije je takođe odabir prototipa. U ovom radu, termin selekcija instance se koristi za označavanje zadatka koji uključuje izbor podskupa instanci iz originalnog skupa podataka, bez razmatranja naknadnog algoritma koji treba da se obuči.

Kada se ispitaju skupovi podataka iz stvarnog sveta, imperativna potreba za algoritmima za izbor primera postaje sve jasnija. S jedne strane, prosečna veličina skupa podataka postaje sve veća i veća. S druge strane, stvarni skupovi podataka obično sadrže bučne instance (eng. Noisy data), izuzetke (eng. Outliers) i anomalije (eng. Anomalies). Pokušaji da se obuči klasifikator, na primer, na osnovu miliona primera može biti težak, pa čak i nerešiv zadatak. Izbor odgovarajućeg podskupa slučajeva je stoga dobra opcija za smanjenje veličine uzorka, omogućavajući njegov naknadni tretman

Izbor instanci podataka (eng. Instance selection) ima ključnu ulogu u zadatku smanjenja podataka zbog činjenice da obavlja suprotan proces u odnosu na izbor atributa(eng. Feature selection - FS). Iako je nezavistan od FS-a, u većini slučajeva, oba procesa se zajednički primenjuju. Suočavanje sa ogromnim količinama podataka može se postići smanjenjem količine podataka kao alternativa za poboljšanje skaliranja algoritama za upravljanje podacima. FS već postiže ovaj cilj, kroz uklanjanje nerelevantnih i nepotrebnih atributa. Na ortogonalan način, uklanjanje instanci može se smatrati jednakim ili čak još efikasnijim načinom sa stanovišta smanjenja podataka u određenim aplikacijama.

Instance selection se odnosi na odabir podskupa podataka kako bi se postigao originalni cilj aplikacije za predikciju podataka kao da se koriste svi podaci. Međutim, odabir samog podskupa podataka ne predstavlja uvek izbor instanci, ovaj pristup se smatra ozbiljnom, inteligentnom operacijom kategorizacije instanci, u odnosu na relevantnost ili zavisnost od šuma u podacima u okviru zadatka koji je potrebno rešiti. S tim u vezi, popularni „Data sampling“ se ne smatra izborom instanci, jer on predstavlja metod selekcije instanci po slučajnom principu (može doći do gubitka bitnih informacija za izračunavanje modela) kako bi se dobilo na brzini izračunavanja, ne uzimajući u obzir logiku koja je potrebna za očuvanje kvaliteta kasnijeg izračunavanja.

Optimalan rezultat IS-a je minimum podskupa podataka, nezavistan od modela, koji može obavljati isti zadatak bez gubitka performansi. $P(A_s) = P(A_t)$

Gde ***P*** predstavlja performanse, ***A*** predstavlja neki od algoritama masinskog učenja, $_s$ je podskup podataka koji je izdvojen i $_t$ je kompletan ili trening skup instanci podataka.
Zadaci izbora instanci [3]:

- Omogućavanje – Kada je skup podataka preveliki, možda izvršenje algoritma ne bude moguće ili algoritam ne može da bude primenjen adekvatno. Selekcija instanci omogućava algoritmu da radi sa velikim količinama podataka, smanjujući njihov obim na manji, reprezentativan uzorak koji zadržava ključne karakteristike originalnog skupa, čime se poboljšava efikasnost i izvodljivost algoritma.

- Fokusiranje – Podaci su formirani od mnogo informacija iz gotovo svih oblasti, ali konkretan zadatak prediktovanja se usredsređuje samo na jedan od aspekata interesa. Selekcija instanci se fokusira na podatke relevantne za traženi zadatak, uklanjajući one instance koje ne doprinose značajno rešavanju problema ili su čak štetne po performanse algoritma, čime se poboljšava kvalitet predikcije i efikasnost algoritma.

- Čišćenje – Odabirom relevantnih instanci uklanjaju se suvišni podaci i šum u podacima, čime se poboljšava kvalitet ulaznih podataka algoritma i time dobija poboljšanje rezultata dobijenih primenom algoritama mašinskog učenja. Selekcija instanci može da identifikuje i ukloni izuzetke, redundancije i nejasnoće u podacima, što dovodi do čistijeg i efikasnijeg skupa podataka za treniranje i testiranje algoritama, što na kraju rezultira boljim performansama modela.

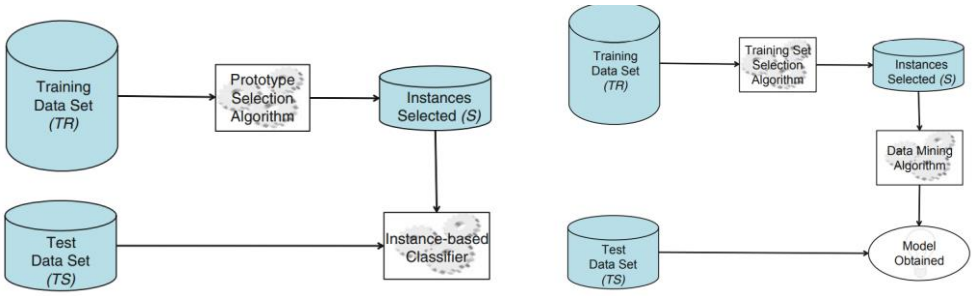
Razlika između *Training set selection* i *Prototype selection*

U početku je nekoliko predloga za odabir najrelevantnijih podataka iz skupa za obuku predloženo razmišljajući uglavnom u KNN algoritmu. Kasnije, kada je termin učenje zasnovano na instanci (eng. Instance-based learning), takođe poznat kao lenjo učenje (eng. Lazy learning), smišljen za okupljanje svih onih metoda koje ne izvode fazu obuke tokom učenja, pojavljuje se termin za odabir prototipa (eng. Prototype selection). Danas porodica IS metoda takođe uključuje predlog za koji se smatralo da radi sa drugim metodama učenja, kao što su stabla odlučivanja, ANN ili SVM. Međutim, nije postojao način da se odredi konkretan slučaj u kojem je IS metoda validna i može se primeniti na bilo koju vrstu DM algoritma (unutar iste paradigme učenja, naravno). Iz tog razloga razlikujemo dve vrste procesa: odabir prototipa (Prototype selection - PS) i izbor skupa za obuku (Training set selection - TSS). Ove metode imaju za cilj da prilagode skupove podataka za različite algoritme, kako bi se poboljšala njihova efikasnost i tačnost, optimizujući resurse i poboljšavajući performanse.

PS metode su IS metode koje očekuju da pronađu skupove za treniranje koji pružaju najbolju tačnost klasifikacije i stopu smanjenja korišćenjem klasifikatora zasnovanih na instanci koji uzimaju u obzir određenu sličnost ili meru udaljenosti. Nedavno su PS metode postale popularnije u oblasti smanjenja podataka. Ova metoda se koristi za smanjivanje veličine skupa podataka u cilju ubrzavanja klasifikacije. Primeri koji se odabiraju kao prototipovi su oni koji se smatraju najreprezentativnijim za celokupni skup podataka. Bavi se odabirom određenog broja primera koji će predstavljati ostale primere u skupu podataka. Pored smanjenja vremena potrebnog za klasifikaciju, PS metode doprinose i boljem razumevanju podataka, olakšavajući analizu i interpretaciju rezultata, što je posebno korisno u oblastima poput medicinske dijagnostike, finansija i marketinga.

TSS metode su definisane na sličan način kao i PS metode. One su poznate kao primena IS metoda nad skupom za trening koji se koristi za izgradnju bilo kog prediktivnog modela. Dakle, TSS se može koristiti kao način da se poboljša ponašanje prediktivnih modela, preciznost, i interpretabilnost. Ova metoda se koristi kada postoji veliki broj primera u skupu podataka i želimo da smanjimo veličinu skupa kako bismo smanjili vreme potrebno za treniranje modela i smanjili mogućnost prenaučenja. TSS metode omogućavaju bolje iskorišćavanje resursa, pružajući efikasnije rešenje za primene mašinskog učenja u različitim industrijskim i istraživačkim domenima, kao što su prepoznavanje obrazaca, analiza slike, i detekcija anomalija. - *Izbor seta za obuku bavi se odabirom skupa primera za treniranje modela koji će biti korišćen u klasifikaciji*.

Činjenica je da se danas mnogo više istražuju PS metode. Okvirna procena predloga prijavljenih u specijalizovanoj literaturi koja se posebno razmatra za TSS može biti oko 10% od ukupnog broja tehnika. Iako PS monopolizuje skoro sve napore u IS, TSS trenutno pokazuje uzlazni trend.



Slika 3. PS proces vs TSS proces

Prototype selection

Postojeći efikasni algoritmi klasifikacije uveliko troše vreme i prostor prilikom obrade velikih skupova podataka, a naročito tokova podataka.
Kako bi se smanjila veličina skupa podataka i vreme izvršavanja klasifikacije, a istovremeno zadržali visoko referentni obrasci efikasnih klasifikacionih doprinosa, postalo je istraživačko žarište u oblasti klasifikacije obrazaca. Stoga je predložena efikasna strategija obrade, koja se naziva selekcija prototipa, a koja predstavlja neophodno smanjenje originalnog skupa podataka na osnovu određenih tehnika [4].

Korišćenjem selekcije prototipa moguće je dobiti skup referentnih prototipova koji mogu odražavati distribucione i klasifikacione karakteristike originalnog skupa podataka. Strategijom selekcije prototipa moguće je postići brzo vreme izvršavanja klasifikacije smanjujući osetljivost na veličinu skupa podataka i šum, bez gubitka tačnosti klasifikacije. Na taj način, problem neprihvatljive potrošnje vremena i prostora delimično je rešen. Međutim, algoritmi selekcije prototipa sami po sebi imaju mnoge nedostatke, kao što su osetljivost na redosled čitanja obrazaca, outlier-e i šum. Stoga, kako bi se prevazišla osetljivost na redosled čitanja obrazaca u algoritmu Kondenzovanog Najbližeg Suseda (CNN) i postigli referentni prototipovi koji su bliže granici klasifikacije na efikasniji način.

Smer pretrage (Direction of search)

Selekcija instanci može se shvatiti kao problem traženja, dat je određeni set podataka, cilj je da se pronađe najreprezentativniji podskup primera za taj set. Može se definisati pet smerova u nastojanju za dobijanje referentnog skupa podataka:

- Dodavanje (eng. **Incremental/Forward selection**) - Inkrementalna pretraga počinje sa praznim podskupom S, i dodaje svaki primer iz seta podataka (TR) u S ako ispunjava neke kriterijume. U ovom slučaju, algoritam zavisi od redosleda prikazivanja i ovaj faktor može biti veoma važan. Trebalo bi da bude slučajan, ali postoje specijalni slučajevi. Jedna prednost inkrementalnog načina rada je da ako su podaci kasnije dostupni, nakon što je obuka završena, oni se mogu i dalje dodavati u S prema istim kriterijumima. Ova funkcionalnost može biti veoma korisna kod rada sa tokovima podataka ili onlajn učenjem. Još jedna prednost je

što su brzi i koriste manje skladišnog prostora tokom faze učenja nego neinkrementalni algoritmi. Glavni mana je da inkrementalni algoritmi moraju da donesu odluku na osnovu malo informacija i stoga su skloni greškama dok nije dostupno više informacija.

- Uklanjanje (eng. **Decremental/Backward selection**) - počinje sa S = TR i zatim traži primere za uklanjanje iz S. Opet, redosled prikazivanja je važan, ali suprotno od inkrementalnog procesa, svi trening primeri su dostupni za pregled bilo kada.
  Jedna mana uklanjanja jeste da vremenski skuplju operaciju od inkrementalnih algoritama. Osim toga, faza učenja mora da se izvede van mreže, jer uklanjanje zahteva sve moguće podatke. Međutim, ako primena dekrementalnog algoritma može da dovede do većeg smanjenja memorije, onda dodatno izračunavanje tokom učenja (koje se radi samo jednom) može pomoći u smanjenju vremena izračunavanja.

- **Batch** - Još jedan način primene PS procesa je u batch modu. Ovo podrazumeva odlučivanje da li svaki primer zadovoljava kriterijum za uklanjanje pre nego što se bilo koji od njih ukloni. Zatim se svi oni koji zadovoljavaju kriterijum istovremeno uklanjaju. Kao i kod dekrementalnih algoritama, batch obrada pati od povećane vremenske složenosti u odnosu na inkrementalni algoritam.

- Mešoviti (eng. **Mixed**) - početak mešane pretrage se obavlja sa unapred odabranim skupom S (slučajno ili odabranim inkrementnim ili dekrementnim procesom) i iterativno može dodati ili ukloniti bilo koju instancu koja zadovoljava specifičan kriterijum. Ovaj tip pretraga omogućava ispravke već izvršenih operacija i njegova glavna prednost je lakše dobijanje skupa instanci sa dobrom preciznošću. Obično pati od istih nedostataka koji su prijavljeni kod dekrementnih algoritama, ali ovo u velikoj meri zavisi od specifičnih predloga.

- **Fixed -** je podgrupa mešane pretrage u kojoj broj dodavanja i uklanjanja ostaje isti. Dakle, broj konačnih prototipova određuje se na početku faze učenja i nikada se ne menja.

Tipovi selekcije

Ovaj faktor je uglavnom uslovljen vrstom pretrage koju sprovode PS algoritmi, bilo da nastoje da zadrže granične tačke, centralne tačke ili neki drugi skup tačaka:

- Kondenzacija (eng. **Condensation**) - ovaj skup uključuje tehnike koje imaju za cilj da zadrže tačke koje su bliže granicama odluke, koje se takođe nazivaju granične tačke. Intuicija koja stoji iza zadržavanja graničnih tačaka je da unutrašnje tačke ne utiču toliko na granice odlučivanja kao granične tačke, i stoga se mogu ukloniti sa relativno malim efektom na klasifikaciju. Ideja je da se sačuva tačnost nad skupom za obuku, ali na tačnost generalizacije nad skupom za testiranje može negativno uticati. Ipak, sposobnost redukcije metoda kondenzacije je obično visoka zbog činjenice da u većini podataka ima manje graničnih tačaka nego unutrašnjih tačaka.

- Izdanje (eng. **Edition**) – ove vrste algoritama umesto toga nastoje da uklone granične tačke uklanjaju tačke koje su „bučne“ ili se ne slažu sa susedima. Ovo uklanja granične tačke, ostavljajući glatke granice odluke iza sebe. Međutim, takvi algoritmi ne uklanjaju unutrašnje tačke koje ne doprinose nužno granicama odluke. Dobijeni efekat se odnosi na poboljšanje tačnosti generalizacije u podacima testa, iako je dobijena stopa redukcije niska.

- Hibrid (eng. **Hybrid**) – ove metode pokušavaju da pronađu najmanji podskup S koji održava ili čak povećava tačnost generalizacije u podacima testa. Da bi se ovo postiglo, omogućava uklanjanje unutrašnjih i graničnih tačaka na osnovu kriterijuma koje slede dve prethodne strategije. KNN klasifikator je veoma prilagodljiv ovim metodama, postižući velika poboljšanja čak i sa veoma malim podskupom odabranih instanci.

Evaluacija pretrage

KNN je jednostavna tehnika i može se koristiti za usmeravanje pretrage PS algoritma. Cilj kojem se teži je da se napravi predviđanje o nedefinitivnoj selekciji i da se uporede izbori. Ova karakteristika utiče na kriterijum kvaliteta i može se podeliti na:

- Filter (eng. **Filter**) - Kada se pravilo kNN koristi za parcijalne podatke kako bi se odredili kriterijumi dodavanja ili uklanjanja, a nije korišćena *leave-one-out* validaciona šema kako bi se dobila dobra procena generalizacijske tačnosti. Činjenica da se koriste podskupovi trening podataka pri svakoj odluci povecava efikasnost ovih metoda, ali tačnost možda neće biti poboljšana. - *Odluka se donosi korišćenjem neke heuristike ili pravila i nije zasnovana na klasifikatoru.*
- Omotač (eng. **Wrapper**) - Kada se pravilo kNN koristi za celokupni skup podataka za obuku uz primenu *leave-one-out* validacione procedure. Kombinacija ova dva faktora nam omogućava da dobijemo odličnu procenu opšte tačnosti, što pomaže da dobijemo bolju tačnost na test podacima. Međutim, svaka odluka uključuje kompletnu računsku obradu pravila kNN nad skupom za obuku, a faza učenja može biti računski skupa.. - *Odluku o odabiru ili brisanju instance donosi klasifikator*

Kriterijumi za poređenje

U ovom delu će biti opisani osnovni kriterijumi po kojima se porede algoritmi selekcije instanci. Svaki od kriterijuma je bitan u konačnom zbiru i svaki set podataka ima neki kriterijum koji mu je najznačajniji kako bi rezultati obrade bili što bolji. Osnovni krijetijumi se mogu podeliti u četiri grupe:

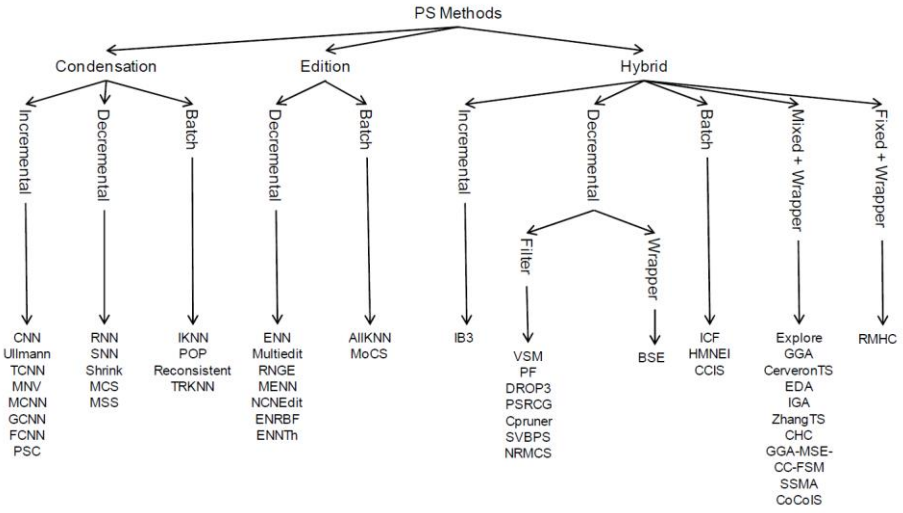
- Smanjenje zauzeća memorijskog prostora (eng. **Storage reduction**) - jedan od glavnih ciljeva PS metoda je smanjenje zahteva za memorijom. Štaviše, još jedan cilj koji je usko povezan sa ovim je da se ubrza klasifikacija. Smanjenje broja sačuvanih instanci obično će dovesti do odgovarajućeg smanjenja vremena potrebnog za pretragu ovih primera i klasifikaciju novog ulaznog vektora.

- Tolerancija buke (eng. **Noise tolerance**) - dva glavna problema mogu se pojaviti u prisustvu buke. Prvi je da će vrlo malo instanci biti uklonjeno jer je potrebno mnogo instanci da bi se održale bučne granice odlučivanja. Drugo, tačnost generalizacije može da trpi, posebno ako se zadrže bučne instance umesto dobrih instanci.

- Tačnost generalizacije (eng. **Generalization accuracy**) – uspešan algoritam često može značajno smanjiti veličinu skupa za obučavanje bez značajnog smanjenja opste tačnosti generalizacije.

- Vremenska zahtevnost (eng. **Time requirements**) - obično se proces učenja obavlja samo jednom na setu za obuku, tako da izgleda da to nije veoma važan metod evaluacije. Međutim, ako faza učenja traje predugo, može postati nepraktična za stvarne primene.

Prototype selection algoritmi



Slika 4. Prototype selection algoritmi

Kao sto se može primetiti na slici 4., postoji veliki broj algoritama koji sluze za odabir manjeg skupa podataka koji će biti reprezentativan za veći skup koji predstavlja. Algoritmi su podeljeni u grupe po svojim osobinama i načinu rada, svaki od njih ima svoje prednosti i mane gledano u odnosu na ostale. Algoritmi su podeljeni u tri osnovne grupe: Condensation, Edition i Hybrid. U nastvaku će gradacijski biti opisani najbitniji predstavnici grupa i objašnjen sam rad algoritama, njihove prednosti i mane, kao i specifičnosti za pojedine predstavnike.

Condensed nearest neighbor(CNN)

Istorijski gledano, od nastanka same potrebe za redukcijom dimenzionalnosti podataka, prvo rešenje koje se pojavilo kada je u pitanju selekcija instanci jeste Condensed Nearest Neighbor(CNN), davne 1968. godine. Danas, iako je prošlo vise od pedeset godina od nastanka ovog algoritma, isti i dalje predstavlja polaznu osnovu za rešenje problema i mnogi savremeni i unapređeni algoritmi su nastali kao nadogradnja na postojeću ideju. U skladu sa tim, za početak priče o algoritmima za redukciju dimenzionalnosti će biti opisan rad CNN algoritma [5].

U osnovi, CNN je undersampling tehnika koja bira podskup podataka iz većeg skupa podataka koji ne dovodi do smanjenja performansi modela, a istovremeno bira minimalan skup instanci podataka. Ovo se postiže dodavanjem instanci podataka u skup ako i samo ako se ne mogu pravilno klasifikovati već postojećim sadržajem skupa podataka, sam algoritam je nastao kao rešenje problema zahteva za memorijom algoritma k-najbližih suseda(KNN).
Sam algoritam je implementiran u *scikit* biblioteci *imbalanced-learn* i dostupan je na korišćenje nakon instaliranja ove biblioteke. Tokom izvršavanja algoritma koristi se KNN algoritam za klasifikaciju tačaka kako bi se odredilo da li će instanca biti dodata u podskup ili ne. Algoritam se izvršava veoma sporo i preporučeno je korišćenje manjih skupova podataka [6].

Algoritam je iterativan, uzevši u obzir da radi sa velikim količinama podataka, jasno je da je tvrdnja da je veoma spor ispravna.

Ako pažljivije pogledamo metod, možemo zaključiti da nemamo osiguranje da izlazni podskup neće biti u potpunosti isti kao i ulazni skup podataka, to se može zavesti kao jedan od nedostataka ovog algoritma, međutim u praksi je drugačije i retki su slučajevi u kojima se ovo može desiti, dok se sanse za ovakav ishod direktno proporcionalno smanjuju sa rastom broja instanci u datom ulaznom skup podataka, uzimajući u obzir da će algoritam biti korišćen za bas takve skupove podataka zanemarljiva je ova činjenica, ali treba imati na umu tu mogućnost.

Na samom početku algoritma može se uočiti prazan početni skup *samples*, kroz iteracije ovaj skup se proširuje, sto nam govori o osnovnoj osobini i opredeljenosti ovog algoritma, a to je da pripada grupi inkrementalnih algoritama za redukciju dimenzionalnosti.

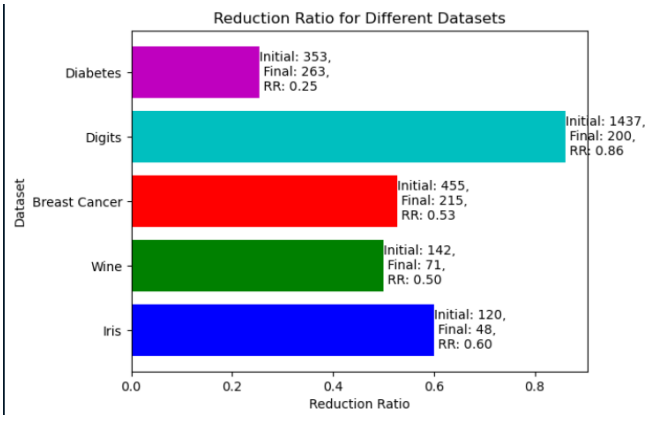
Ovaj metod takođe dokazuje i svoju osobinu *condensed* na taj način sto bira samo granične tačke, tj. instance trening skupa podataka koje određuju njegovu specifičnost, međutim ovo može biti okidač za smanjenje tačnosti predviđanja novih test podataka, zato sto se mogu pojaviti nove granične tačke koje pripadaju nekoj klasi, međutim neće biti približne odabranim graničnim tačkama.

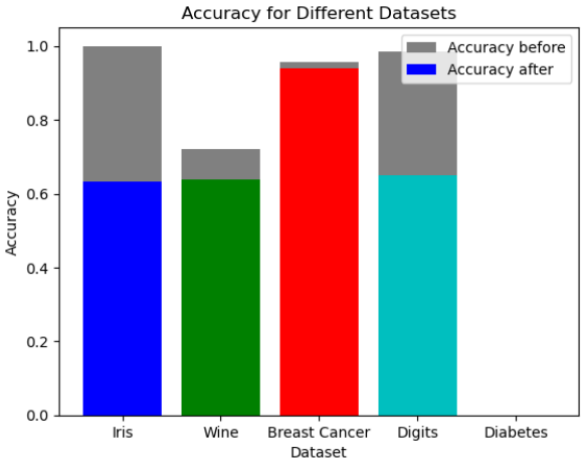
Algoritam je takođe zavistan od raspoređenosti instanci u datasetu, neće za sve rasporede instanci davati isti rezultujući podskup podataka, sto nas navodi na razmišljanje, da li je izlazni set podataka ustvari i najreprezentativniji podskup ili postoji neki podskup, sto je vrlo verovatno, koji bolje reprezentuje ulazni skup.

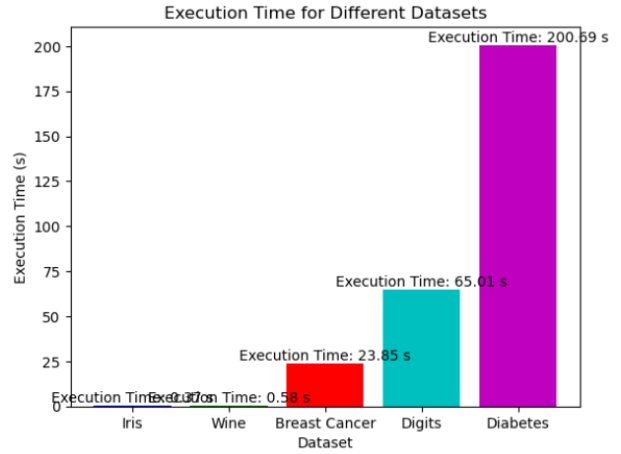
U slučajevima kada se koristi za balansiranje setova podataka, ovaj konačni podskup podataka se sastoji od svih instanci koje pripadaju manjinskoj klasi i samo primeri iz dominantne klase koji se ne mogu pravilno klasifikovati se postepeno dodaju u podskup.

Kompleksnost Condensed Nearest Neighbor algoritma zavisi od nekoliko faktora, kao što su broj instanci u trening skupu podataka, broj karakteristika (atributa) i broj klasa. U najgorem slučaju, kompleksnost algoritma je O(n^3), gde je n broj instanci u trening skupu podataka.
Ovaj slučaj se javlja kada algoritam mora da prođe kroz sve instance više puta kako bi izvršio redukciju. Međutim, u praksi, CNN algoritam često može da postigne zadovoljavajuće rezultate sa manje iteracija i time smanji vreme izvršenja. U stvarnim scenarijima, kompleksnost može biti znatno manja od gornje granice.

Važno je napomenuti da je ovaj algoritam pogodan za inkrementalno učenje i može se poboljšati kroz razne optimizacije, kao što su korišćenje efikasnijih struktura podataka ili ubrzavanje procesa računanja udaljenosti.







Slika 6. Rezultati primene CNN algoritma na različite setove podataka

Na slici 6. prikazani su rezultati izvršavanja CNN algoritma, prvo je prikazana efikasnost algoritma kada je u pitanju redukcija, zatim tačnost klasifikacije pre redukcije i nakon redukcije i na kraju vreme koje je bilo potrebno za izvršenje redukcije.
Zapažanja su sledeća:

- Najveće smanjenje podataka se dogodilo kada je u pitanju data set „digits“, međutim, kada uzmemo u obzir tačnost klasifikacije i vreme izvršenja, sa smanjenjem broja instanci od 80% smanjena je i tačnost izračunavanja za otprilike 25% sto je ogroman gubitak u konačnom skoru i može se zaključiti da ovaj algoritam nije bio bas efikasan kada je u pitanju ovaj skup podataka, iako je smanjenje bilo ogromno, slično se desilo i sa skupom podataka „Iris“, uzevši u obzir količinu podataka.

- Set podataka koji se odnosi na dijabetes apsolutno nije dobro odreagovao na smanjenje količine podataka, međutim može se reci da je u ovom slučaju sam algoritam najbližih suseda praktično neprimenjiv za ovakav skup podataka, tako da bi se rezultati u svakom slučaju trebali zanemariti u konačnom zapažanju.

- Skupovi podataka koji su dali najbolje rezultate jesu „Wine“ i „Breast Cancer Data set“, smanjenje podataka za oko 50%, sto je odličan rezultat i tačnost klasifikacije koja je skoro pa zanemarljivo smanjena su pokazatelji da je u ovom slučaju CNN algoritam u velikoj meri doprineo redukciji dimenzionalnosti, a da je postignut cilj održavanja tačnosti klasifikacije.

U daljem radu će sva ispitivanja biti izvršena nad datim setovima podataka kako bi se stvorila realna slika i poređenje različitih pristupa nad istim podacima.

Opis rada algoritma:

Ulazni parametar metode jeste trening skup(*train*) podataka koji treba da se redukuje. *samples* na kraju treba da predstavlja reprezentativni, smanjen podskup trening skupa podataka.
Početnom skupu se na početku dodaje slučajni element iz trening skupa podataka. Promenljiva *additions* sluzi za izlazak iz beskonačne *while* petlje nakon sto se prekine dodavanje elemenata u podskup.

Nakon dodavanja početnog elementa u traženi podskup, počinje beskonačna petlja koja je aktivna sve do trenutka kada se prestaje sa dodavanjem novih elemenata u podskup. Za svaku instancu iliti element ulaznog, već redukovanog skupa podataka, se bira slučajan element ulaznog skupa, zatim se vrši proračun i trazi najmanja udaljenost između slučajnog elementa i elemenata već dodatih u podskup samples, kada se pronađe najbliži element, proverava se da li je pronađeni element u istoj klasi sa slučajnim elementom, ako jeste to znaci da nema potrebe dodati ga u podskup, ako nije dodaje se u podskup jer predstavlja novitet i graničnu vrednost.

Nedostaci ovog algoritma:

1. Izabrani skup podataka je u velikoj meri nasumičan
2. Vremenski zahtevan
3. Zahtevan u pogledu memorijskih resursa
4. Postoji mogućnost preterane adaptacije podskupa na trening skup
5. Ne radi dobro za podatke koji sadrže *noise* ili greške
6. Može smanjiti interpretabilnost podataka
7. Ne garantuje smanjenje broja instanci
8. Ne izbacuje duplikate iz trening podataka

Prednosti CNN algoritma:

1. Smanjenje količine podataka
2. Sprečava overfitting

Edited Nearest Neighbour (ENN)

Algoritam uređivanja najbližih suseda (eng. Edited Nearest Neighbour - ENN) predstavljen je 1972. godine od strane D. L. Wilsona. Wilsonov rad se bavio ograničenjima algoritama Najbližeg suseda (eng. Nearest Neighbor - NN) i k-Najbližih suseda (eng. k - Nearest Neighbour k-NN) u vezi sa klasifikacijom *noisy* podataka.

Wilson je predložio metodu za uređivanje trening skupa podataka, uklanjanjem instanci koje su verovatno pogrešno klasifikovane. ENN je poboljšao performanse klasifikacije, smanjujući uticaj šuma, autsajdera ili pogrešno označenih instanci, čime su NN i k-NN klasifikatori postali robustniji i precizniji.

Od tada je ENN proučavan, proširen i prilagođen za razne primene u mašinskom učenju i „rudarenju“ podataka. Istraživači su predložili modifikacije kao što su varijacije metrika

udaljenosti, različite strategije za odabir k i alternativne tehnike uređivanja. Razvoj ENN-a doveo je do istraživanja drugih tehnika za smanjenje šuma i metoda izbora instanci za poboljšanje performansi klasifikatora zasnovanih na NN[5].

Navedeni algoritam po svojim preferencama u potpunosti predstavlja suprotnost ranije opisanom CNN algoritmu. Pre svega radi se o algoritmu iz *Edition* grupe algoritama koji je dekrementalan, dakle, ovaj algoritam nastoji da od početnog trening skupa podataka stvori novi, redukovan skup podataka, izbacivanjem instanci koje mogu dovesti do greške u klasifikovanju test podataka.

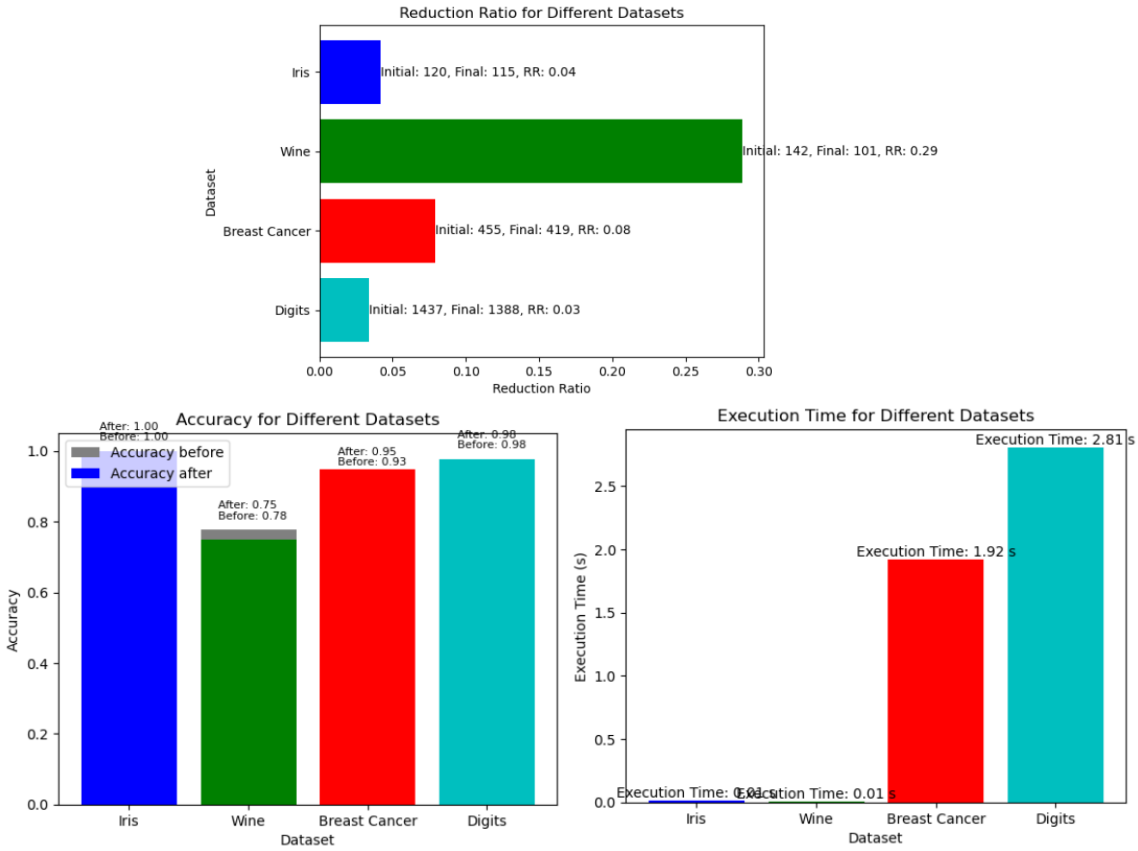
Ovaj algoritam se u programskom jeziku *Python* može implementirati kroz biblioteku *imbalanced-learn* kao *EditedNearestNeighbours.*

ENN metod radi tako što prvo pronalazi K-najbližih suseda za svaku instancu, a zatim proverava da li je većinska klasa iz K-najbližih suseda instance ista kao i klasa instance ili ne. Ako se većinska klasa K-najbližih suseda instance i klasa instance razlikuju, tada se instanca i njenih K-najbližih suseda brišu iz skupa podataka. Podrazumevano, broj najbližih suseda koji se koristi u ENN metodi je K=3.

Koraci u ENN algoritmu su sledeći:

1. Dat je ulazni set trening podataka za N instanci, proverava se postavljena vrednost za K(broj najbližih suseda). Ako nije postavljeno K, uzima se difoltna vrednost 3.
2. Pronalazi se K najbližih suseda instance u setu podataka u odnosu na ostale instance.
3. Ako je instanca u klasi koja je različita od dominantne klase svoje okoline, instanca se izbacuje iz trening skupa podataka.
4. Ponavljati korake 2 i 3 dok se ne postigne željena posednutost klasa

Kompleksnost ovog algoritma zavisi od dva faktora, a to su pronalaženje najbližih suseda i uklanjanje instanci koje ne zadovoljavaju uslov. Prvi faktor ima složenost $O(n^2)$ dok drugi faktor ima složenost O(n*k) sto nas dovodi do zaključka da je složenost ovog algoritma O(n*(n+k)).



Slika 10. Rezultati primene ENN algoritma na različite setove podataka

Na slici 10. su prikazani rezultati primene ENN algoritma na ugrađene setove podataka. Za broj suseda u ENN algoritmu je odabrano 3 i vršena je klasifikacija sa jednom najbližim susedom k-NN algoritma.

Zapažanja su sledeća:

1. Set podataka koji sadrži podatke o vinu je najbolje odreagovao na ENN algoritam, procenat podataka koji su izbačeni iz trening skupa podataka je 30, sto govori o tome da je u trening skupu podataka bilo dosta „lutajućih“ instanci, međutim, smanjen je accuracy za 3%, može se zaključiti da je primena ovog algoritma bila uspešna kada su podaci o vinu bili u pitanju.

2. Specifičnost je i to da se Breast Cancer set podataka odlično adaptirao na ENN algoritam, sto je rezultovalo smanjenjem broja instanci za skoro 10% i povećanjem tačnosti klasifikacije za cak 3% sto je ogroman rast uzevši u obzir da se radi o tačnosti koja je blizu savršenoj.

3. Ostali skupovi podataka su pokazali raspoređenost i minimalnu *noise* u podacima, pa u skladu sa tim nije mnogo instanci podataka izbačeno iz početnog skupa, ali ovim setovima podataka tačnost obrade nije smanjena, tako da se i

ovde može reci da je algoritam bio uspešan u određenoj meri, ali njegova primena nije bila potrebna ako se faktor vremena uzme u obzir.

4. Takođe se može primetiti da kod ovog algoritma veličina skupa podataka igra glavnu ulogu po pitanju vremena izvršenja, sto je i logično, jer je za svaku instancu potrebno pronaći najbliže susede. Ovo može biti zabrinjavajuće ako uzmemo u obzir da se lako može pojaviti set podataka sa brojem instanci reda stotina hiljada ili nekoliko miliona.

Ovaj algoritam je veoma moćno oružje kada se radi o „zalutalim“ podacima u zadacima klasifikacije. Jasno je da efekti nisu zadovoljavajući kada je u pitanju redukcija podataka, međutim, primenom na primer Breast Cancer seta podataka je dokazano da se ovaj algoritam može izuzetno dobro odraziti na same rezultate klasifikacije.

Prednosti:

1. Početni elementi nisu slučajno odabrani
2. Uklanja *noisy* podatke
3. Pozitivno utiče na tačnost klasifikacije

Nedostaci:

1. Neznatno smanjuje broj instanci
2. Sa povećanjem količine podataka, povecava se vreme potrebno za izvršenje algoritma

Poređenje algoritama

	CNN	ENN	DROP	Tomek-Links
Smer pretrage	Incremental	Decremental	Decremental	Decremental
Tip selekcije	Condensed	Edition	Hybrid	Condensed
Evaluacija pretrage	/	/	Filter	/
Redukcija dimenzionalnosti	Velika	Mala	Srednja	Mala
Uklanja *noisy* instance	Ne	Da	Da	Ne
Povecava/smanjuje/ne menja accuracy	Ne menja/smanjuje	Povecava/smanjuje/ne menja	Povecava/Ne menja	Povecava/Ne menja
Vremenski zahtevan	Da	Ne	Ne	Ne
Kompleksnost	$O(n^3)$	O(n*(k+n))	O(n*k)	O(n)
Sprečava overfitting	Da	Ne	Ne	Ne
Garantuje redukciju	Ne	Ne	Ne	Ne
Izbacuje duplikate	Ne	Ne	Ne	Ne
Nasumičan rad	Da	Ne	Ne	Ne

Tabela 1. Osnovne karakteristike Prototype Selection algoritama

Aktivno učenje (Active Learning)

Aktivno učenje je posebna paradigma u mašinskom učenju gde algoritam učenja ima mogućnost da interaktivno postavlja upite o određenim instancama podataka kako bi dobio njihove željene izlaze. Ovaj pristup je posebno koristan u situacijama gde je dobijanje označenih trening podataka skupo ili vremenski zahtevno.
Osnovna ideja aktivnog učenja je da algoritam mašinskog učenja može postići veću tačnost sa manje trening podataka ako može da bira podatke iz kojih uči. U mnogim scenarijima mašinskog učenja, neoznačeni podaci mogu biti obilni i lako dostupni, ali označavanje istih može biti skupo. U takvim situacijama, aktivno učenje pokušava da prevaziđe problem označavanja podataka tako što pametno bira podskup primera za označavanje.

Proces aktivnog učenja

1. Inicijalizacija:
    - Počinje se sa malim setom označenih podataka i velikim setom neoznačenih podataka.
    - Inicijalni model se trenira na dostupnim označenim podacima.

2. Iterativni proces: a) Predviđanje: Model daje predviđanja za neoznačene primere. b) Strategija upita: Algoritam bira najinformativnije primere za označavanje. c) Označavanje: Odabrani primeri se označavaju (obično od strane ljudskih eksperata). d) Ažuriranje modela: Model se ponovo trenira uključujući novooznačene primere. e) Evaluacija: Procenjuju se performanse ažuriranog modela. f) Ponavljanje: Proces se ponavlja od koraka a) dok se ne zadovolji određeni kriterijum zaustavljanja.

3. Finalizacija:
    - Konačni model se evaluira na posebnom test setu.

Strategije upita
Ključni aspekt aktivnog učenja je strategija upita - metod kojim algoritam bira koje primere da označi. Neke popularne strategije uključuju:

1. Uncertainty Sampling: Bira primere o kojima model nije siguran. Ovo može biti:
    - Najmanje pouzdano predviđanje
    - Margina najmanje pouzdanosti: razlika između dve najverovatnije klase
    - Entropija: mera nesigurnosti preko svih mogućih oznaka
2. Query-By-Committee: Koristi ansambl modela i bira primere o kojima se modeli najviše ne slažu.
3. Expected Model Change: Bira primere koji bi najviše promenili trenutni model ako bi bili označeni.
4. Expected Error Reduction: Bira primere koji bi najviše smanjili očekivanu grešku modela.
5. Density-Weighted Methods: Kombinuje nesigurnost sa reprezentativnošću primera u prostoru podataka.

Primena u praksi

U praksi, implementacija aktivnog učenja često izgleda ovako:

1. Priprema podataka:
    - Podela dostupnih podataka na označeni set, neoznačeni set i test set.
    - Pretprocesiranje i priprema značajki (features).
2. Inicijalni trening:
    - Trening početnog modela na malom setu označenih podataka.
3. Glavni ciklus aktivnog učenja:
    - Predviđanje na neoznačenom setu.
    - Primena strategije upita za odabir primera za označavanje.
    - Simulacija procesa označavanja (u eksperimentalnom okruženju) ili stvarno označavanje od strane eksperata.
    - Ažuriranje trening seta sa novim označenim primerima.
    - Ponovno treniranje modela.
    - Evaluacija na validacionom setu.
4. Evaluacija:
    - Finalna evaluacija na test setu.
    - Analiza krive učenja (kako se performanse poboljšavaju sa povećanjem broja označenih primera).

Prednosti i izazovi

Prednosti aktivnog učenja:

- Smanjenje troškova označavanja podataka.
- Poboljšanje efikasnosti učenja.
- Mogućnost rada sa ograničenim resursima za označavanje.

Izazovi:

- Odabir odgovarajuće strategije upita za specifičan problem.
- Balansiranje istraživanja (exploration) i iskorišćavanja (exploitation) u procesu selekcije.
- Potencijalno uvođenje pristrasnosti u trening set kroz proces selekcije.
- Računska složenost nekih sofisticiranijih strategija upita.

Primene

Aktivno učenje se uspešno primenjuje u mnogim domenima, uključujući:

- Obrada prirodnog jezika (NLP): klasifikacija teksta, analiza sentimenta
- Računarski vid: prepoznavanje objekata, segmentacija slika
- Biomedicina: analiza genomskih podataka, otkrivanje lekova
- Prepoznavanje govora
- Daljinska detekcija i geografski informacioni sistemi

Active learning instance selection je posebno koristan u scenarima kao što je klasifikacija toksičnih komentara, gde označavanje velikog broja primera može biti vremenski zahtevno i subjektivno. Ovaj pristup omogućava efikasno korišćenje resursa za označavanje fokusirajući se na najinformativnije primere.

Zaključak

Ovaj rad se bavi istraživanjem Instance Selection metoda, posebno fokusirajući se na redukciju dimenzionalnosti kao ključni način za smanjenje količine podataka bez gubitaka u performansama modela. Prototype Selection (PS) se ističe kao osnovni pristup koji omogućava kreiranje podskupa podataka, zadržavajući tačnost testiranja. U radu su opisane i razmatrane različite strategije selekcije, kao što su Condensation i Edition, kao i hibridne metode koje kombinuju prednosti ovih pristupa. Takođe, istaknuti su ključni algoritmi za redukciju podataka, kao što su CNN, ENN, Tomek Links, DROP koji imaju svoje prednosti i mane.

Aktivno učenje je prikazano kao komplementaran pristup, koji kroz pametnu selekciju primera za označavanje može smanjiti potrebnu količinu označenih podataka, čime efikasno koristi resurse. Kombinovanjem redukcije dimenzionalnosti i aktivnog učenja, moguće je postići optimalne performanse modela uz minimalne podatke i resurse, uz pažljiv odabir strategija upita i prilagođavanje specifičnostima problema koji se rešava.

Literatura

[1] J. L. Salvador García, in *Data Preprocessing in Data Mining*.

[2] J. A. O.-L. ·. J. A. C.-O. ·, in *A review of instance selection methods*, https://sci2s.ugr.es/sites/default/files/files/TematicWebSites/pr/2010-Olvera-AIR.pdf.

[3] S. Herbst, "An Introduction to Instance Selection in Data Mining," https://medium.com/@sabrinaherbst/an-introduction-to-instance-selection-in-data-mining-7b2ac94fb07.

[4] J. B. a. R. Tibshirani, in *PROTOTYPE SELECTION FOR INTERPRETABLE*, https://arxiv.org/pdf/1202.5933.pdf.

[5] J. Brownlee, in *Undersampling Algorithms for Imbalanced Classification*, https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/.

[6] E. Alpaydın, in *Voting over Multiple Condensed Nearest Neighbors*, https://www.researchgate.net/publication/2644628_Voting_over_Multiple_Condensed_Nearest_Neighbors#pf4.

[7] I. TOMEK, in *Two Modifications of CNN*, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4309452.

[8] N. Z. R. V. Christoph F. Eick, in *Using Representative-Based Clustering for Nearest Neighbor Dataset Editing*, https://www.researchgate.net/publication/4133603_Using_Representative-Based_Clustering_for_Nearest_Neighbor_Dataset_Editing.

[9] R. A. A. Viadinugroho, in *Imbalanced Classification in Python: SMOTE-ENN Method*, https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50.