Univerzitet u Kragujevcu

Domaći zadatak

iz predmeta Sistem za podršku odlučivanju

Tema:

Teorijski izveštaj o klasifikacionim algoritmima sa nadgledanim učenjem

mentori: Ognjen Pavić,
prof. dr. Tijana Geroski,
prof. dr. Nenad Filipović
student: Željko Simić 3vi/2023

Kragujevac 2024.

1 Uvod

1.1 Opis procesa učenja[1]

Problem obuke obuhvata skupinu n uzoraka podataka i time se pokušava predvideti svojstvo nepoznatih podataka. Ako svaki uzorak ponaosob je više nego 1 unos, npr., multidimenzionalni unos, može se reći da ima više atributa ili features-a.

Problemi obuke potpadaju u nekoliko kategorija:

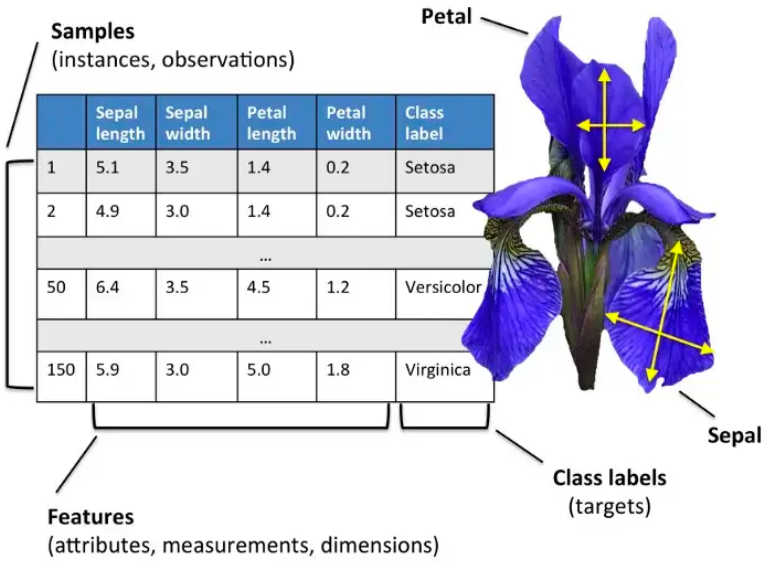
- *Nenadgledajuće učenje (nije tema ovog rada)* - obuka podacima sastoji se u unosu više vektora x bez ikakvih uparenih ciljnih vrednosti. Cilj je otkriti pripadnost grupi po sličnosti naspram drugih vektora (klasterizacija), ili po raspodeli podataka unutar prostora ulaznih podataka (procena gustine), ili po projekciji podataka višedimenzionog prostora na manje dimenzioni shodnom za vizuelizaciju.
- *Delimično nadgledano učenje (nije tema ovog rada)[2]* - situacija gde skup podataka namenjen za obuku ima poneki uzorak koji nije uparen sa ciljanom vrednošću, gde se zarad adekvatnijeg previđanja boljom generalizacijom i distribucijom koriste ciljanom vrednošću nenaznačeni uzorci.
- *Podsticajuće učenje (nije tema ovog rada)[3]* - opšti okvir rada gde se vrši obuka *agenata* (tj. algoritma obuke) težeći da se obuče da vrše akcije u zadatom *okruženju* (tj. problema za rešavanje) zarad najboljeg mogućeg slučaja nagrađivanja.
- ***Nadgledano učenje*** - gde se skup podataka sa prikazanim primerom na slici 1. prilaže sa dodatnim atributima po kojim će se predviđanje usmeravati (klasa za svaki od uzoraka, labela). Sastoji se od 2 načina rada (gde samo jednog obrađujemo):
  - *Regresija (delimično nije tema rada)* - gde željeni izlaz u priloženom skupu podataka se izražava kao jedna ili više kontinualnih promenljivih, a zadatak predviđanja da nepoznatim ulazima uzorka se naziva regresijom.
  - ***Klasifikacija*** - uzorci mogu da pripadaju dvema (binarna klasifikacija) ili više klasa (višeklasna klasifikacija) i želimo ustanoviti kako bi se predviđanje klasa za nenaznačene podatke vršilo uz pomoć već klasama naznačenih podataka. Način da se sagleda klasifikacija je kao diskretni (suprotno kontinualnom) oblik nadgledane obuke gde od ograničenog broja kategorija za svaki od n uzoraka se namerava dodeliti tačna klasa ili kategorija.

1.1.1 Trening skup i test skup

Mašinsko učenje se sačinjava od poduhvata gde se vrši obuka uz svojstva jednog skupa podataka, a vrši testiranje svojstvima uz pomoć nekih drugih skupova podataka. Uobičajena praksa u mašinskom učenju je da se vrši evaluacija algoritma uz podelu skupa podataka na 2 dela. Gde je jedan deo namenjen za obuku (trening skup), a drugi za testiranje (test skup).

1.1.2 Multiclass vs. multilabel obuka

Kada se koriste multiklasni klasifikatori, zadaci obuka i predviđanja se izvršavaju tako da zavise od formata podataka ciljanih vrednosti. 1D niz višeklasna labela skupljenih svih uzoraka



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Slika 1: Primer višeklasnog skupa podataka - Iris flower, sa atributima Sepal length, Sepal width, Petal length, Petal width i atributom klase *Class label* u posebnih 150 uzoraka.

je moguće navesti da radi *multiklasna predviđanja*. Ciljane vrednosti je moguće mapirati tako da se konvertuju u binarni zapis, tj. niz sastojan od binarnih cifara tako da skup uzoraka ima ciljane vrednosti kao 2d niz i nakon obuke, za obavljana predviđanja smatra se da su *multilabel predviđanja*. Takođe je moguće da ciljana vrednost ima niz više labela (skupljeno po svim uzorcima ciljana vrednost biva 2d niz) za svaki uzorak pri obuci i kasnije naspram toga vršiti predviđanja.

1.1.3 Popularni algoritmi za klasifikaciju[4]

- **Logistička regresija** - je algoritam nalik linearnoj regresiji samo što umesto ciljana vrednost da bude neki broj, ona biva neka binarna vrednost ('Da'/'Ne', 0/1). Iako se naziva regresijom ona vrši klasifikaciju na osnovu regresije.
- **K-Najbližih suseda (kNN)** - najjednostavniji klasifikatorni algoritam korišćen da uoči uzorke (npr. Euklidskoj distanci) podeljene po raznovrsnim klasama da bi predvideli klasifikaciju novonastalog uzorka.
- **Mašina potpornih vektora (SVM)** - korišćen u oba tipa nadgledanog učenja i zasnovan na konceptu ravni odlučivanja (hiperravni) koje ističu ograničenja odlučivanja, tj. dele skup uzoraka na pripadnost u različitim klasama.
- **Naivni Bajes** - zasnovan na Bajesovoj teoremi uz više nezavisnih pretpostavki među predviđačima.

- **Stablo odlučivanja** - korišćen u oba tipa nadgledanog učenja u obliku strukture stabla. Razbija svakom iteracijom skup podataka u sve manje podskupove, dok pritom je stablo odlučivanja inkrementalno razvijeno.
- *Metodi ansambla* - sadrže nekolicinu nadzirajućih modela obuke koji su svaki za sebe posebno obučeni i rezultati su objedinjeni na raznolike načine da bi se dobilo kranje predviđanje. Neki od algoritama ovog tipa su:
  - **Klasifikacija slučajnih šuma** - algoritam zasnovan na bagging/bootstrap agregaciji koja eliminiše problem overfitting-a, poboljšava tačnost i smanjuje bias (pristrasnost).
  - *Klasifikacija boost-ovanog gradijenta (neće biti obrađivan)* - objedinjava slabe obučivače, primarno da bi izbegao bias pri predviđanju. Umesto da se radi bagging - gomilanje predviđaća, radi se boosting gde se radi "slivanje" iz jednog obučivača u drugi.

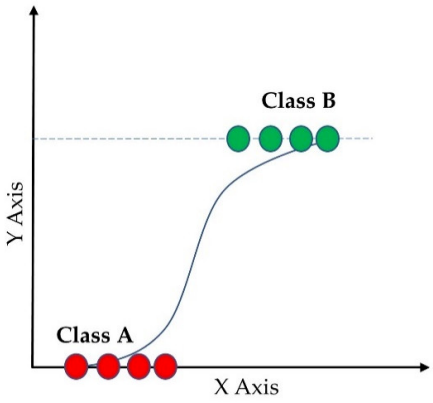
U sledećim sekcijama biće obrađivani pomenuti algoritmi.

2 Teorijske osnove i metodologija

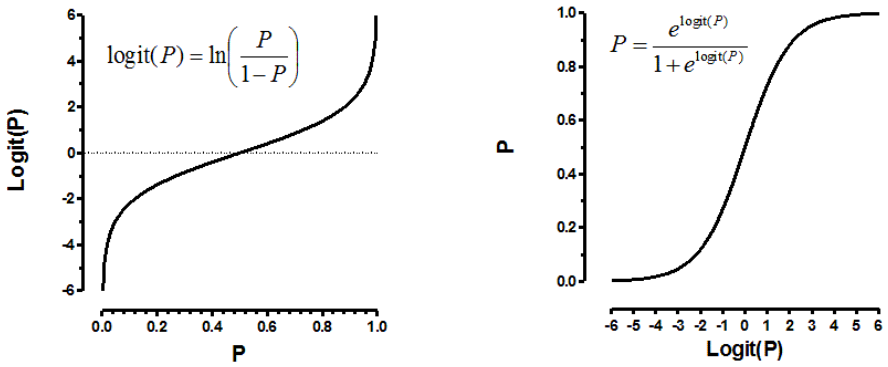
2.1 Logistička regresija

Predstavlja se kao linearni model pre za klasifikaciju, nego za regresiju. Poznata je u nekim literaturama kao:

- logit regresija - predstavljena (i vizuelizovana na slici 2.) kao inverz sigmoidne-logističke funkcije, gde po verovatnoći p, važi $logit(p) = log(\frac{p}{1-p})$[5], tzv. logaritam neobičnosti - odnos verovatnoća da će se događaj desiti i da se neće desiti; moguće je kasnije svesti na oblik linearne kombinacije $logit(p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$[6]; prikazana na slici 3.;

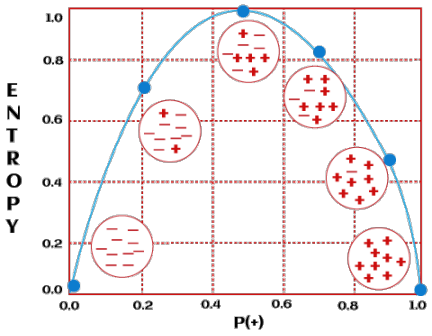


Slika 2: Rad logističke regresije za klasifikaciju uzoraka A i B.



Slika 3: Grafici logit i sigmoidne funkcije.

- klasifikacija maksimuma-entropije - što veće neizvesnosti/raznolikosti u skupu podataka[7] po verovatnoći da se desi događaj P_i slučajnog odabira uzorka klase i, gde važi $E = -\sum_{i=1}^{N} P_i * log_2(P_i)$, prikaz na slici 4.;



Slika 4: Ilustracija putem grafika funkcije entropije.

- iliti log-linearni klasifikator - koji za razliku od linearnog modela, maločas pomenutog, se zasniva na množenju, umesto sabiranja: $Y_i = X_{1i}^{\beta_1} X_{2i}^{\beta_2} \dots X_{ki}^{\beta_k} e^{\epsilon_i}$, gde se lako preoblikuje u: $log(Y_i) = \beta_1 log(X_{1i}) + \beta_2 log(X_{2i}) + \cdots + \beta_k log(X_{ki}) + \epsilon_i$.[8]

U modelu logističke regresije, verovatnoće predstavljaju moguće ishode nekog događaja uobičajeno koristeći se logističkom-sigmoidalnom funkcijom. Primenjuje se kao (liči na priču multiklasifikatornih vs. multilabel ciljanih vrednosti)[9]:

- binarna,
- 'sam-protiv-svih' (OVR)

ili multinomijalna logistička regresija. sa opcionim l_1, l_2 argumentima, ili regularizacija elastične mreže; regulazizacija je implicitno postavljena i stvar je mašinskog učenja, ali ne i statistike - poboljšava numeričku stabilnost.

2.1.1 Binarna logistička regresija

U slučaju binarne logističke regresije uzima se u obzir pretpostavljena je ciljana vrednost $y_i \in 0, 1$, za uzorak i. Nakon obuke, obavljanjem predviđanja npr. verovatnoće klase pozitivnog ishoda $P(y_i = 1|X_i) = \hat{p}(X_i)$ i predstavlja se kao

$$\hat{p}(X_i) = \text{expit}(X_i w + w_0) = \frac{1}{1 + \exp(-X_i w - w_0)}$$

.

Pritom, pošto je ovo optimizacioni problem, binarna klasa logističke regresije sa koeficijenom regularizacije $r(w)$ minimizuje prateću funkciju procene:

$$\min_w C \sum_{i=1}^{n} s_i \left(-y_i \log(\hat{p}(X_i)) - (1 - y_i) \log(1 - \hat{p}(X_i))\right) + r(w),$$

gde s_i naznačava korisnički dodeljene težine po određenom uzorku obuke (vektor s je sveden na množenje težinske klase y_i sa težinom uzoraka svih elemenata zajedno X_i).

Da ne bi bilo ikakve regularizacije C je moguće podeiti na vrlo visoku vrednost. Uočavajući da je moguće pomnožiti nekom konstantom $b > 0$ težine uzoraka je jednako delotvorno množenju (inverzne) jačine regularizacija C po b.

kazna	$r(w)$
None	0
l_1	$\|w\|_1$
l_2	$\frac{1}{2}\|w\|_2^2 = \frac{1}{2}w^T w$
regularizacija elastične mreže	$\frac{1-\rho}{2}w^T w + \rho\|w\|_1$

Tabela 1: Koeficijent regularizacija $r(w)$ po argumentu *kazne* u binarnoj logističkoj regresiji

Postoje 4 načina da se uspostavi rad uz koeficijent regularizacija $r(w)$ po argumentu *kazne*, prikazano na tabeli 1.

Za regularizaciju elastične mreže sa ρ (koji deluje u skladu sa l_1 koeficijentom) kontroliše se snaga l_1 regularizacije protiv l_2 regularizacije. Elastična mreža je ekvivalentna l_1 regularizaciji ako je $\rho = 1$, inače l_2 regularizaciji ako je $\rho = 0$.

2.1.2 Multinomijalna logistička regresija

Vrši se proširenje binarnog slučaja na K klasa i svođenjem njega na multinomijalnu logističku regresiju i udeljuje se značaj na pojam *log-linearnih modela*. Moguće je parametrizovati model K-klasne klasifikacije koristeći se samo sa $K-1$ težinskim vektroima, ostavljajući da samo verovatnoća svih klasa mora biti objedinjena. Namerno se prekomerno parametrizuje model korišćenjem K težinskih vektora u čast olakšavajuće implementacije zarad uspostavljanja simetričnog intuktivnog bias-a (sa pristrasno zastupljenim svim klasama pri obuci[10]) u odnosu na raspored klasa. Ovaj efekat postaje izrazito važan pri regularizaciji. Čin prekomerne parametrizacije dovodi do nepovoljnosti za nekažnjivačke modele pošto rešenje ne bi bilo posebno.[11]

Neka je $y_i \in 1, \ldots, K$ ordinalna (uređena kategorička) labela enkodirana kao ciljana promenljiva za posmatranje uzorka i. Umesto jednog vektora koeficijenata, imamo matricu koeficijenata W i svaki red vektor W_k je ustupljen za klasu k. Stremi se da se predvide verovatnoće $P(y_i = k|X_i)$ kao:

$$\hat{p}_k(X_i) = \frac{\exp(X_i W_k + W_{0,k})}{\sum_{l=0}^{K-1} \exp(X_i W_l + W_{0,l})},$$

gde se time ustanovljava oblik optimizacije kao:

$$\min_W -C \sum_{i=1}^{n} \sum_{k=0}^{K-1} [y_i = k] \log(\hat{p}_k(X_i)) + r(W).$$

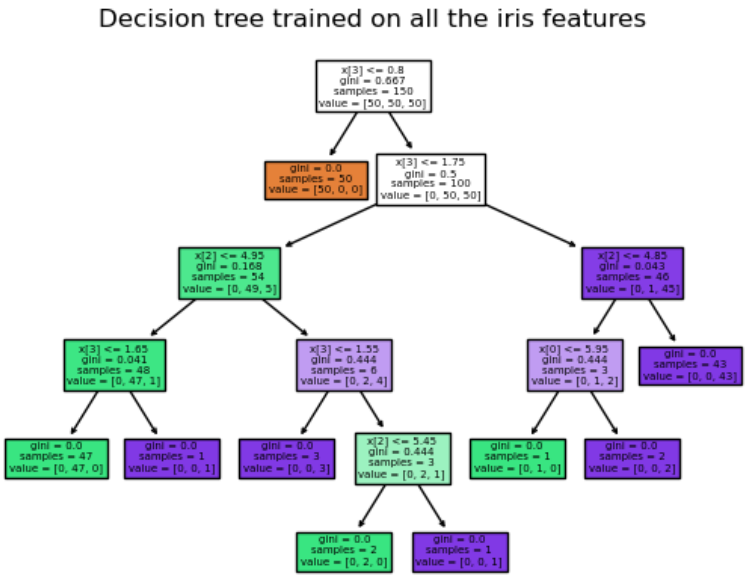
Po $[y_i = k]$ su korišćene Iverson zagrade, koje daju vrednost 0 ako $y_i \neq k$, inače daju 1. Kao i ranije, postoje 4 načina da se uspostavi rad uz koeficijent regularizacija $r(w)$ po argumentu *kazne*, po tabeli 2.:

kazna	$r(w)$
None	0
l_1	$\|\|W\|\|_{1,1} = \sum_{i=1}^{m}\sum_{j=1}^{K}\|W_{i,j}\|$
l_2	$\frac{1}{2}\|\|W\|\|_F^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{K}W_{i,j}^2$
regularizacija elastične mreže	$\frac{1-\rho}{2}\|\|W\|\|_F^2 + \rho\|\|W\|\|_{1,1}$

Tabela 2: Koeficijent regularizacija $r(w)$ po argumentu *kazne* u multinomijalnoj logističkoj regresiji

2.2 Stabla odlučivanja[18]

Metod nadgledanog učenja neparametrizovanog korišćenog. Cilj je napraviti model koji predviđa vrednosti ciljanih promenljivih po uzoricima namenjenjih za obuku putem pravila odluka naspram atributa skupa podataka. Drvo može biti posmatrano kao isparčana konstantna aproksimacija. Stabla odlučivanja uče naspram podataka da klasifikuju kreirajući skupinu if-then-else pravila odlučivanja. Što dublje drvo ide to su kompleksnija pravila odlučivanja, tako i tačniji model. Jednostavna su za razumevanje i rastumačavanje. Stabla moguće vizuelizovati kao na slici 5.



Slika 5: Primer stabla odlučivanja obučena naspram Iris skupa podataka.

Takođe je uzeti u obzir smanjenje dimenzionalnosti (Principal component analysis - korišćen da vrši rastavljanje skupova podataka u skup rastućih ortogonalnih komponenti zarad pojašnjenja maksimalne količine disperzije[24], Independent Component Analysis - rastavljanje višestrukih signala u sabirajuće podkomponente koje su maksimalno nezavisne[25], Feature selection - radi se zarad pojačavanja tačnosti procenjivača ili ubrzavanja performansi[26]) zara

boljeg nalaženja features-a koji su diskiriminativni.

2.2.1 Multi-output problemi

Problem nadgledanog učenja sa nekolicinom izlaza za predviđanje, gde je ciljana vrednost Y u skupu podataka oblika 2D niza gde mu je broj redova jednak broju uzoraka, a broj kolona jedna broju izlaza za uzorak. Ako nema međusobnih korelacija među izlazima, jednostavan način da se reši taj problem je pravljenje n nezavisnih modela, tj. jedan model za svaki izlaz, i svaki nezavisno iskoristiti za predviđanje n izlaza. Uočavajući da za iste ulaze izlazne vrednosti su međusobno srodne (u korelaciji), najbolje je sve n izlaze istovremeno predvideti. Pritom, tačnost generalizacije rezultujućeg procenjivača će obično se uvećati.

Zahtevaju se prateće promene naspram običnog binarnog stabla odlučivanja:

- n izlaznih vrednosti u listovima, umesto 1,
- Pri pravilu podele računati središnje umanjenje preko svih čitavih n izlaza,

Klasifikaciju moguće demonstrirati kroz primer upotpunjivanja ljudskog lica sa multi-output estimators. Ulazi predstavljeni kroz vektor X mogu biti pikseli gornjeg dela lica, a izlaz predstavljen vektorom Y mogu biti pikseli donjeg dela lica, kao na slici 6.

Slika 6: Korišćen multi-output decision tree za prepoznavanje lica u drugoj koloni.

2.2.2 Složenost izvršavanja

Vremenska složenost da se izradi balansirano binarno stablo $O(n_{uzoraka} n_{features} \log(n_{uzoraka}))$, a pristupa je $O(\log(n_{uzoraka}))$. Iako je cilj da stablo bude balansirano, ona neće biti nužno takva.

Neka nagovestimo da podstabla su aprokisamtivno balansirajuća, cena svakog čvora uzima u obzir pretragu $O(n_{features})$ da bi se došlo do feature-a koji nudi najveću redukciju po *kriterijumu nečistosti* gde mu je primer logaritamski gubitak (tj. information gain-u kojim dajemo značaj sagledanim obrascu u skupu podataka, kao i redukciji u entropiji: $InformationGain = entropija_{čvora\ roditelja} - entropija_{čvora\ deteta}$).[7] Cena $O(n_{features} n_{uzoraka} \log(n_{uzoraka}))$ za svaki čvor, vodeći se do ukupne cene naspram svih stabala (sumiranjem cena svakog od čvora) koja je $O(n_{features} n^2_{uzoraka} \log(n_{uzoraka}))$.

2.2.3 Matematička formulacija

Za čvor m, vektor obuke $x_i \in \mathbf{R}^n$, $i \in \{1, \ldots, l\}$, vektor labele $y \in \mathbf{R}^l$. Podaci čvora m su Q_m, sa n_m uzorcima. Podela kandidat $\theta = (j, t_m)$, gde j je feature, a t_m prag. Podele skupa podataka su $Q_m^{left}(\theta)$ i $Q_m^{right}(\theta)$.

$$Q_m^{left}(\theta) = \{(x, y) | x_j \leqslant t_m\}$$
$$Q_m^{right}(\theta) = Q_m \backslash Q_m^{left}(\theta)$$

$H()$ je funkcija gubitka ili nečistosti i zavisi na osnovu oba tipa nadgledane obuke.

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta))$$

Vrši se odabir parametara zarad minimalizacije nečistosti:

$$\theta^* = \text{argmin}_\theta \, G(Q_m, \theta)$$

Slučaj rekurentne baze za podskupove $Q_m^{left}(\theta^*)$, $Q_m^{right}(\theta^*)$ dostizanje maksimalne dubine, $n_m < \min_{samples}$ ili $n_m = 1$.

2.2.4 Kriterijum klasifikacije

Se sagleda za čvor m, gde izlaz raspolaže vrednostima $0, \ldots, K-1$.

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k)$$

Obavlja se računanje mera nečistosti:

- Gini : $H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$
- Log Loss ili entropija : $H(Q_m) = -\sum_k p_{mk} \log(p_{mk})$

Kriterijum entropije sračunat je uz pomoć Shannon entropije za dosupne klase. Sagleda verovatnoću predstavljenu kao učestalost zastupljenosti klasa uzoraka namenjenih za obuku koji su dosegli do nivoa lista stabla, lista m. Kriterijum podele čvora stabla sprovodi se isto kao *minimalizacija logaritmičkog gubitka*, tzv. *cross-etropija*, *multinomialna devijacija* mežu pravim labelama y_i i probabilističkih predviđanja $T_k(x_i)$, modela stabla T za klasu k. Računa se prvi odziv logaritmičkog gubitka za skup podataka D:

$$\text{LL}(D, T) = -\frac{1}{n} \sum_{(x_i, y_i) \in D} \sum_k I(y_i = k) \log(T_k(x_i))$$

U klasifikatornom stablu, verovatnoće klasa predviđanja unutar čvorova listova su konstantni, gde $\forall(x_i, y_i) \in Q_m$, a sada je $T_k(x_i) = p_{mk}$, za svaku klasu k. Drugačiji zapis Shannon entropija za svaki list-čvor modela T, uzetim u obzir težinama naspram broja podataka obuke je:

$$\mathrm{LL}(D, T) = \sum_{m \in T} \frac{n_m}{n} H(Q_m)$$

2.2.5 Podrška po postojanosti nedostajućih vrednosti

Za svaki potencijalni prag na neizgubljenim podacima, podela će se vršiti uz evaluaciju sa svim izgubljenim vrednostima uvrstavajući ih u levi ili desni čvor.

Odluke će biti donošene na sledeći način:

- Implicitno kada se vrši predviđanje uzorci sa izgubljenim vrednostima će biti klasifikovani sa klasama korišćenim u nailazećim podelama tokom treninga.
- Pri situaciji da kriterim evaluacije je isti za oba čvora, tada se u neizvesnosti eksplicitno uvrštava izgubljena vrednost na desnu stranu prilikom vršenja predviđanja. Podela je ustanovljena gde sve izgubljene vrednosti idu na stranu jedno čvora potomka, a ostale (neizgubljene) na drugu.
- Ako prilikom obuke ne bude izgubljenih vrednosti za dati feature, onda tokom predviđanja nedostajućih vrednosti se mapira u čvoru potomka sa više zastupljenih uzoraka.

2.2.6 Odsecanje minimalnom cenom složenosti

Algoritam koji je korišćen za odsecanje stabla zarad izbegavanje situacije overfitting-a. a ima parametre:

- $\alpha \geqslant 0$ - parametar složenosti,
- T - stablo,
- $R_\alpha(T)$ - cena složenosti.
- $|\widetilde{T}|$ - broj termalnih čvorova (listaova) u T
- $R(T)$ - tradicionalno ustanovljen kao ukupna stopa loše klasifikovanih termalnih čvorova (npr. ukupna nečistoća težina uzoraka termalnih čvorova - kao kriterijum)

$$R_\alpha(T) = R(T) + \alpha|\widetilde{T}|$$

Minimalna cena složenosti odecanja ustanovljena je u podstablu T koji minimalizuje $R_\alpha(T)$. Cena složenosti kao mera jedinog čvora je $R_\alpha(t) = R(t) + \alpha$, za granu T_t sagledanu kao stablo čvora korena t. Obično je nečistoća čvora veća od sume nečistoća termalnih čvorova $R(T_t) < R(t)$.

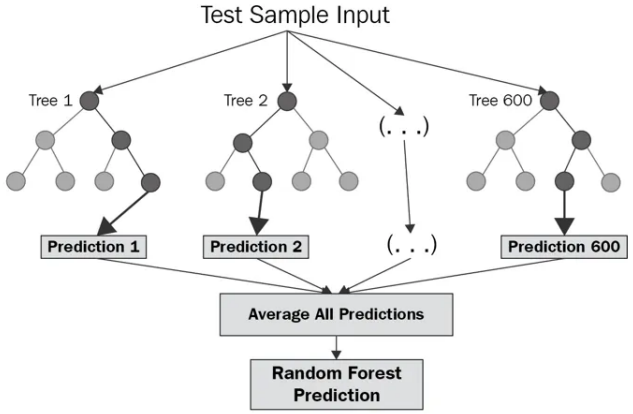
Određuje se efektivna vrednost α gde $R_\alpha(T_t) = R_\alpha(t)$ ili $\alpha_{eff}(t) = \frac{R(t)-R(T_t)}{|T|-1}$.

Netermalni čvorovi sa najmanjom vrednošću α_{eff} su veza koja treba biti odsečena.

Postupak se terminira kada je minimalna vrednost α_{eff} stabla koja je odsečena veća od unapred podešene ccp_{alpha} vrednosti[27].

2.3 Slučajne šume[31]

Meta-procenjivač (metod ansambla) nad više klasifikatora slučajnih stabala nad raznolikim poduzorcima skupa podataka koji se koriste pri ustanovljavanju proseka tačnosti predviđanja (prikaz na slici 7.) i upravljanje overfitting-om.[28] Ističe se kao tehnika pomešati i kombinovati (preturb and combine; kreira više verzija datog grafa, primenjuje funkciju cena za čvorve pojedinačno za svaki graf, ukombinuje rezultate[29]) namenjenog za stabla i sa time je uveden koncept nasumičnosti u konstrukcije klasifikacija.



Slika 7: Plan predviđanja slučajnih šuma.

Podržavaju proširenje na probleme više izlaza kao što je pomenuto u sekciji 2.2.1.

Svako stablo u ansamblu građeno je iz uzorka sa zamenom (bootstrap uzorak) iz trening skupa. Pri gradnji stabla podeli po svakom čvoru, najbolja podela je se pravi pri iscrpnoj pretrazi vrednosti features-a ili po nasumičnom skupu veličine naspram broja najviše mogućih features-a koja je unapred podešena. Cilj je izbeći visoku disperziju i overfitting procenjivača slučajnih šuma. Uvođenjem efekta nasumičnosti razložilo je greške pri predviđanju, a pri uzimanju proseka - neke greške su eliminisane. Zauzvrat, moguće je naići na blag porast cene u bias-u.

Neki od kombinacije klasifikatora služe računanjem proseka, neki se koriste "glasanjem" za posebnu klasu.

2.3.1 Unapred postavljeni parametri

Glavni parametri zaslužni za prilagođavanje koji su korišćeni su *broj procenjivača* i *broj najviše features-a*, a može i da bude nekad *broj stabala*. Preporuka je da budu veće vrednosti parametara, ističe se zahtevnost računa kao ishod. Nakon nekog kritičnog broja količine stabala ishoduje stagnaciji boljitka performansi. Sporedna veličina nasumičnih podskupova features-a se uzima u obzir kada se po čvoru vrši podela - što je niža to je veće umanjenje u disperziji, a veće pojačanje bias-a, sa manjom vrednošću je dosegnuta veća nasumičnost. Moguće je podesiti i *najveću dubinu* i *minimalan broj podela uzoraka*. Preporuka je da se izvrši cross-validation proces (ima moć da poredi i vrši odabir, manje se vezuje za određene koncepte nego druge tehnike predviđanja[30]) pri najbolje moguće podešenim parametrima. Moguće je podesiti zamenu uzorka (bootstrap mogućnost) gde greška generalizacije biva procenjena sa običnim uzorcima, uz bootstrap-bagging agregacije.

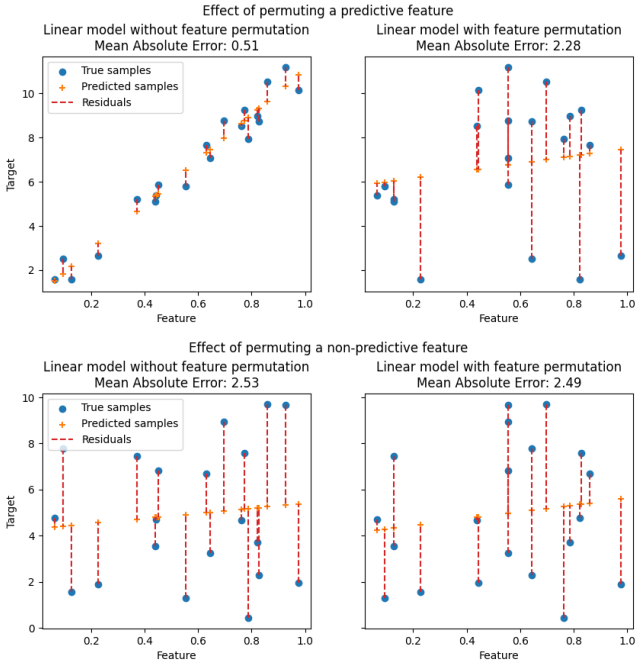
Paralelizacija je moguća i ustanovljava paralelno izračunavanje predviđanja. Moguće je ekspilicitno podesiti broj zadataka koji će se izvršiti nad tom količinom jezgara u procesoru istovremeno. Vidljivo će biti uvećan performans pri radu sa većim brojem stabala ili pri radu sa većom količinom podataka na samom stablu.

Rangiranje srodnosti (npr. dubina) feature-a korišćeno je kada čvor odluke u stablu ustanovljava *važnost feature-a* sve qdo predvidljivosti ciljane promenljive. Features-i korišćeni na vrhu stabla dprinose konačnom predviđanju krupnije frakcije ulaznih uzoraka. Očekivana frakcija uzoraka doprinosi pri proceni srodne važnosti features-a. U nekim sistemima frakcije uzoraka, kako feature doprinosi se uz kombinaciju umanjenja nečistosti po podeli pravi normalizovane procene za snagu predvđanja po feature-u. Po vršenju uprosečavanja procene moći predviđanja nad nekolicinom nasumičnih stabala se smanjuje disperzija takvih procena i njihovih korišćenja uz odabire feature-a - MDI (središnje opadanje u nečistosti). Tim sračunavanjima feature važnosti zasnovanih na nečistosti sleduju 2 mane koje vode u zablude:

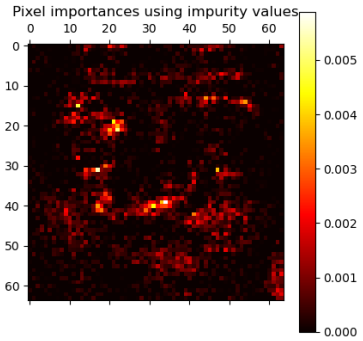
- Sračunatim statistikama (funkcijom uzoraka; slučajnim promenljivama) svedenim iz trening skupa i sa time nas ne informiše o tome koji features-i su najvažniji da bi se napravile dobra predviđanja po priloženim skupovima podataka.

- Daje se prednost features-ima visokog kardinaliteta, pa i više jednoznačnih vrednosti. Permutacija važnosti feature-a (tehnika za obuku modela po statističkim performansama, korisna za nelinearne, neprovidne procenjivače koji uključuju nasumično pretumbavanje vrednosti posebnog feature-a i posmatranje degradacije rezultata ocene modela; ustanovljava se koliko se model osanja na određeni feature)[32] je alternativa sračunavanjima feature važnosti zasnovanih na nečistosti, i eliminacija problema oslanjanja na feature, prikaz na slici 8. Dovodi se pojam *važnosti permutacija* naspram MDI-a.

Na slici 9. je dat primer korišćenja ekstremnih slučajnih šuma koji demonstriraju primenu feature važnosti po primeru individualnih piksela za svrhe prepoznavanje lica, što je svetlija tačka piksela to je važnost feature-a poklopljena pri funkcije procene predviđanja.



Slika 8: Grafici sračunavanjima feature važnosti zasnovanih na nečistosti i permutacija važnosti feature-a (po kolonama) u okolnostima svojstva (ne)predvidljivog feature-a (po redovima).



Slika 9: Važnost piksela sa paralelnim slučajnim šumama, pri prepoznavanju lica.

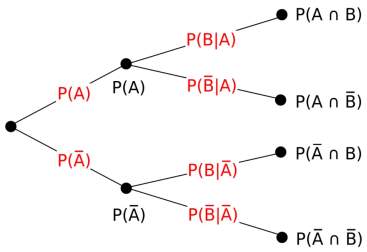
2.4 Klasifikacija naivnim Bajesom[34]

Metode naivnog Bajesa se zasnimavju na primeni Bajesove teoreme sa "naivnim"pretpostavkama naspram nezavisnosti uslova među parovima features-a datih vrednosti klasa promenljivih.

Bajesova teorema ustanovljava odnos promenljive y (namenjena za vrednosti klasa uzoraka) i zavisnih x_i-eva (namenjena za feature vrednosti uzoraka; zavisnost delimično predstavljena slikom 10.), za $i = \overline{1, n}$:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Gde je:



Slika 10: Dijagram stabla verovatnoća po nivoima događaja za formulu uslovne verovatnoće $P(A \cap B) = P(B|A)P(A)$, tj. $P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$.

- $P(y)$ - *prior* verovatnoća ishoda da se događaj nailaženja vrednosti klase y desi pri predviđanju,
- $P(x_1, \ldots, x_n)$ - verovatnoća da se desi nailaženje nove posebne torke vrednosti features-a uzorka pri predviđanju, tj. ovde predstavljenih u vidu niza promenljivih, tzv. *pokrića (evidence)*,
- $P(x_1, \ldots, x_n \mid y)$ - verovatnoća *verodostojnosti (likelihood)*,
- $P(y \mid x_1, \ldots, x_n)$ - *posterior verovatnoća*. Po kombinovanju sagledane informacije, ažurira se a priori informacija po sagledanju klase y.[35]

$$posterior = \frac{prior \times likelihood}{evidence}$$

Verovatnoća *nezavisnosti događaja*[**?**] je $P(A|B) = P(A)$ (tj. $P(A \cap B) = P(A)P(B)$; npr. računanje verovatnoće tokom bacanja 2 novčića - koji su međusobno **zavisna 2 događaja**, i nakon toga, biranje ta dva).

I definiciji da je *uslovna nezavisnost* ustanovljena događajem C, gde $P(C) > 0$ ako

$$P(A \cap B|C) = P(A|C)P(B|C).$$

Pritom, koristeći se $P(A|B) = \frac{P(A \cap B)}{P(B)}$, ako $P(B) > 0$. I ako je $P(B|C), P(C) \neq 0$ onda svedemo na:

$$P(A|B \cap C) = \frac{P(A \cap B|C)}{P(B|C)}.$$

To uradimo postupkom: $P(A|B \cap C) = \frac{P(A \cap B \cap C)}{P(B \cap C)} = \frac{P(A \cap B|C)\cancel{P(C)}}{P(B|C)\cancel{P(C)}} = \frac{P(A \cap B|C)}{P(B|C)}$

Nakon toga, ako su A i B uslovno nezavisne od C.

$$P(A|B \cap C) = \frac{P(A \cap B|C)}{P(B|C)} = \frac{P(A|C)P(B|C)}{P(B|C)} = P(A|C).$$

Ako A i B su uslovno nezavisni ustanovljenim C, tada važi da:

$$P(A|B, C) = P(A|C).$$

Koristeći se ovime kao naivnom uslovnim nezavisnim predviđanjem da:

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y)$$

Potom, $\forall i$ odnos je pojednostavljen (uz svojstvo nezavisnosti događaja formulom $P(A \cap B) = P(A)P(B)$) na:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Uzimajući u obzir da je $P(x_1, \ldots, x_n)$ konstantno pošto važi za ulaz (features-i), radi se klasifikaciono pravilo za neko k:

$$P(y \mid x_1, \ldots, x_n) = k * P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y)\prod_{i=1}^{n} P(x_i \mid y),$$

gde se koristi procena $\hat{y}$ - *maksimalne a posteriori* - MAP za izvođenje $P(y)$ i $P(x_i|z)$. Prethodni iskaz pre dobijenog MAP-a je srodan po učestalosti klase y koja se nalazi u trening skupu.

Naivni Bajes klasifikatori se razlikuju najglavnije po predviđanjima koje vrše za distribuciju $P(x_i \mid y)$.

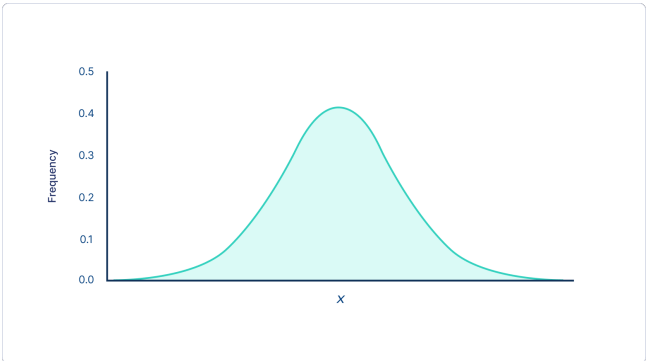
Uprkos preuprošćenim predviđanjima, naivni Bajes klasifikatori rade vrlo pouzdano u realnom svetu, u slučaju klasifikacije dokumenata i filtriranja spamova. Ne zahtevaju krupne skupove podataka zarad obuke da bi procenjivali po neophodnim parametrima. Obučavači naivnog Bajesa i klasifikatori mogu biti ekstremno brzi u poređenju sa više sofisticiranijim metodama. Raskopčavanjem (rasparivanje, decoupling) klasa uslovljenog feature-a se raspodeljuje (distribuira) ponaosob vršeći procenu nezavisno po dimenzionoj distribuciji. Ovim se olakšava pitanje dimenzionalnosti. Zauzgred, iako je poznat kao pouzdan klasifikator, poznat je i kao loš procenjivač, pa verovatnoće na izlaza ne uzimati "zdravo za gotovo".

Neke od vrsta klasifikatora naivnog Bajesa su:

- *Gausov naivni Bajes* - likelihood features-a ustanovljeni kao Gausijanski (distribucija verovatnoća koja je simetrična naspram očekivane vrednosti - središta, prikazivanje da neki od podataka su frekventniji po zastupljenosti od ostalih, više od središnje vrednosti)[36], gde su $\sigma_y = \sqrt{\frac{\sum_{i=1}^{n}(x-\mu_y)^2}{n-1}}$ (standardna devijacija za promenljivu klase) i $\mu_y = \frac{1}{n}\sum_{i=1}^{n} x_i$

(očekivanje/središnja vrednost za promenljivu klase) procenjeni putem maksimalnog likelihood-a.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}$$



Slika 11: Gausova Bell kriva distribucije, učestalost zastupljenosti po vrednosti.

- *Multinomijalni naivni Bajes* - namenjen za multinomijalno distribuirane (generalizacija binomne distribucije koja diskretno obrađuje do ishoda 2 ili više vrednosti fiksnih verovatnoća za svaku vrednost nezavisno generisanih[37]) podatke, jedan od uobičajenih Bajesovih vrsta namenjenih za tekstualnu klasifikaciju (podaci su tipično reprezentacija brojčani vektori reči, vizuelizovan rezultat na slici 12., iako tf-idf vektori kao alternativa su poznati kao pouzdani u praksi koji mogu biti razgranati na 2 slučaja učestalost pojma - TF i inverzne učestalosti dokumenta - IDF).

Slika 12: Multinomijalni naivni Bajes model daje ovakav rezultat naknadno vizuelizovan, ključna reč veća po svojoj učestalosti.

Distribucija je parametrizovana po vektorima $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ $\forall y$ gde je n broj features-a (u tekst klasifikaciji, veličina rečnika) i $\theta_{yi} = P(x_i \mid y)$ kao verovatnoća feature-a i koji se pojavljuje u uzorcima udeljenim klasi y.

Parametri θ_y su procenjeni po *smooth-ovanoj (uglađenoj)* veziji maksimalne verodostojnosti (likelihood-a), tj. brojke relativne učestalosti:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

gde $N_{yi} = \sum_{x \in T} x_i$ je broj puta gde feature i nailazi pri klasi y u trening skupu T, a $N_y = \sum_{i=1}^{n} N_{yi}$ je ukupno prebrajanje svih features-a za klasu y.

Smoothing zastupljen uz $\alpha \geqslant 0$ odgovoran za features-e neuvrštene među uzorke obuke i eliminiše nepogodnosti u vezi verovatnoća jednakih 0 u daljim sračunavanjima. Podešavanjem $\alpha = 1$ dobija se *Laplace smoothing*, dok uz $\alpha < 1$ dobija se *Lidstone smoothing*.

- *Komplementarni naivni Bajes (CNB)* - proširenje standardnog algoritma multinomialnog naivnog Bajesa - MNB koji je posebno namenjen za nebalansirane skupove podataka. CNB koristi statistiku komplementa svake posebno klase zarad izračunavanja težina modela. Od konstruktora ovog algoritma je izjavljeno da su proračuni parametara empirički više pouzdanija nego kod MNB-a. CNB često prevazilazi MNB (do uočljivih razmera) pri zadacima klasfikacija teksta. Postupak za sračunavanje težina sledi:

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$
$$w_{ci} = \log \hat{\theta}_{ci}$$
$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

  gde sumiranja svih dokumenata j nisu u klasi c, d_{ij} je ili-ili brojka tf-idf vrednosti pojmova i u dokumentu j, a α_i je smoothing hiperparametar kao kod MNB-a i $\alpha = \sum_i \alpha_i$. Druga normalizacija adresira težnju da duži dokumenti imaju zastupljeniji proračun parametra u MNB. Pravilo klasifikacije je:

$$\hat{c} = \arg\min_c \sum_i t_i w_{ci}$$

  tj. dokument dodeljen klasi je onaj sa *najsiromašnijim* poklapljanjem komplementa.

- *Bernulijev naivni Bajes (BNB)* - rade nad podacima distribuiranih na odgovornost višestruke Bernulijeve distribucije (slična geometrijskoj raspodeli, ali nisu iste)[38]; tj. mogu biti više features-a, ali svaki ponaosob je predviđan da bude binarno-vrednovana (Bernilijeva, Bulova) promenljiva. Klasa zahteva uzorke da budu predstavljeni kao binarno-vrednovani feature vektori. Pravilo odluka za Bernilijev naivni Bajes zasnovan je na:

$$P(x_i \mid y) = P(x_i = 1 \mid y)x_i + (1 - P(x_i = 1 \mid y))(1 - x_i)$$

  gde se razlikuje od MNB-ovog pravila po tome što ova eksplicitno kažnjava neispoljavanje feature-a i koji jej indikator klase y, gde multinomijalna varijanta tome ne daje značaj prosto.

  U slučaju klasifikacije teksta, vektori pojavljivanja reči (pre nego vektori brojnosti reči) moguse koristiti pri obuci i korišćenju klasifikatora. Pogodniji za korišćenje kod kraćih dokumenata.

- *Kategorički naivni Bajes* - zasnovan na podacima koji su kategorički distribuirani. Predviđaju da svaki feature posebno, opisan indeksom i ima sopstvenu kategoričku distribuciju. Svaki feature i je u trening skupu X pretpostavljen klasi y. Skup indeksa uzoraka je definisan $J = \{1, \ldots, m\}$, gde m označava broj uzoraka.

  Verovatnoća kategorije t u feature-u i date klase c je procenjena kao:

$$P(x_i = t \mid y = c\,;\, \alpha) = \frac{N_{tic} + \alpha}{N_c + \alpha n_i},$$

gde $N_{tic} = |\{j \in J \mid x_{ij} = t, y_j = c\}|$ je broj puta kategorija t se ispoljava u uzorcima x_i, koja pripada klasi c, dok $N_c = |\{j \in J \mid y_j = c\}|$ je broj uzoraka sa klasama c, α je smoothing parametar i n_i je broj dostupnih kategorija u feature-u i.

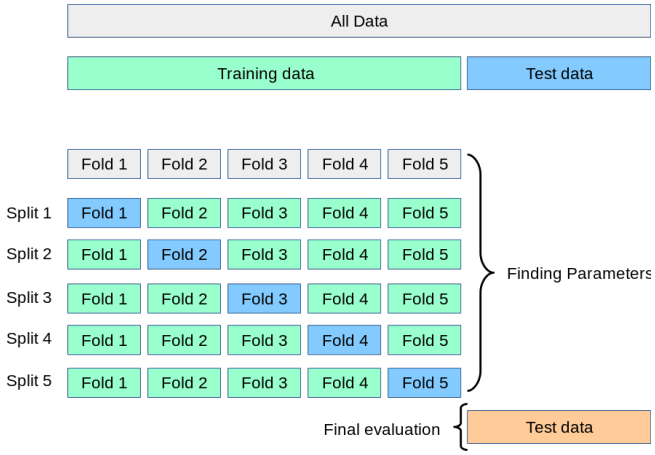
Obično se X enkodira po ordinalnom enkoderu[39] u praksi kategroije reprezentovane za svaki feature i brojevima $\overline{0, n_i - 1}$ gde je n_i broj dostupnih kategorija feature-a i.

Naivni Bajes modeli mogu da se susretnu sa problemima klasifikacija visokih razmera u slučaju pun trening skup ne može se smestiti u memoriju. Neki od sistema koriste metode koji klasifikatori inkrementalno vrše klasifikacije, tj. *out-of-core (izvan srži)* klasifikacije. Može da vrši iscrpne proračune sa time je opet preporučljiv isparčan rad (da bi se izbeglo preopterećenje primarne memorije) i pri ovom rešenju.

2.5 Mašina potpornih vektora (SVM)[40]

Korišćeni za klasifikacije, regresije i detekciju outliers-a (diskrimnatornih uzoraka u odnosu na druge). Kasnije će detaljnije osnovna metodologija biti predstavljena, a sada idu pojašnjenja za osobine SVM-a.

Ne pružaju direktno procene verovatnoća, sračunati su iscrpnim 5-fold cross-validacijama (uzorke izdeli u 5 grupa, tzv. folds-e, koristeći se "ostavi 1 van" strategijom, obuka zarad predviđanja se radi nad 5-1=4 folds-a, a onaj 1 fold je ostavljen za testiranje, prikazani na slici 14.)[30].



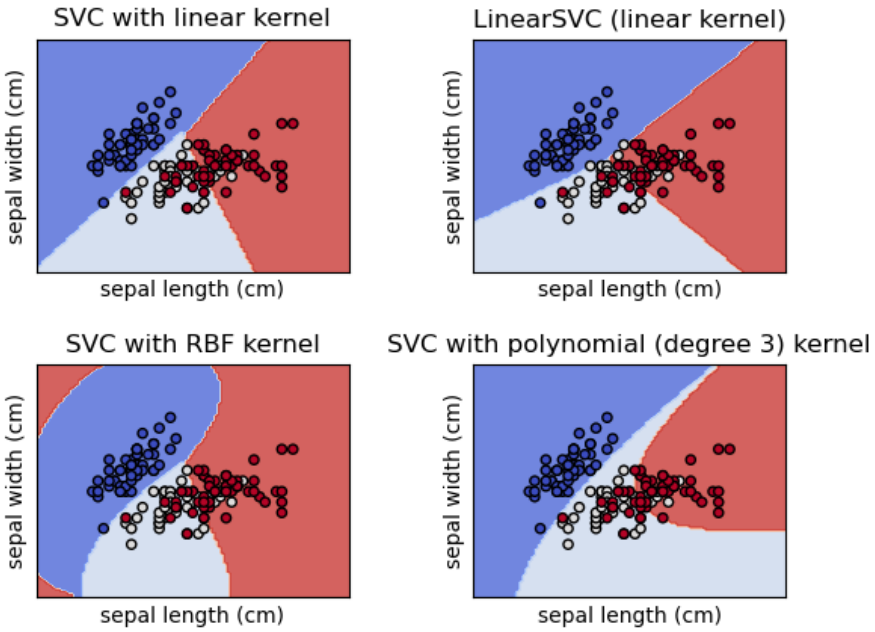
Slika 13: primer 5-fold cross-validacije i naknadnog testiranja posle predviđanja crossvalidaion-om

Neki razvijeni sistemi SVM-a podržavaju *guste* i *raspršene* uzoračke vektore kao ulaze. Klasifikacije koje se obavljaju mogu biti binarne i multiklasifikatorne za skup podataka.

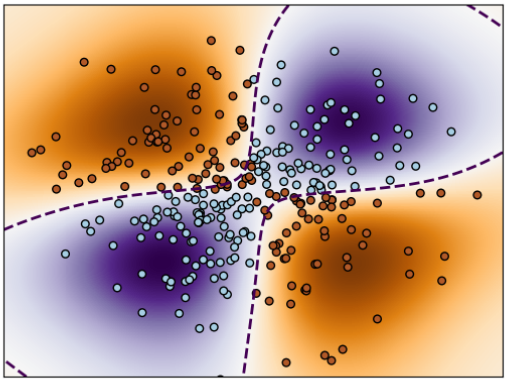
Nu-SVM klasifikatori su malo specifičniji i imaju različite matematičke formulacije. *Linearni-SVM klasifikatori* su brža verzija SVM klasifikatora u slučaju linearnih kernela, koristi gubitke *kvadratnih hindževa* (alternativa cross-entropiji - ustanovljava greške kada su znakovi prisutni među tačnim i predviđenim vrednostima i u ovom slučaju taj sračunat gubitak se kvadrira)[42], isto reguliše preseke pri radu sa LIBLINEAR rešavačima[13], ako su postavljeni kao potrebni. Moguće je vršiti postupak *skaliranja presecanja* (sintetički feature - veštački izgenerisan / nije na osnovu događaja iz realnog sveta - sa konstantnom vrednošću jednaka je ovoj meri, pa je primenjena na vektor uzorka i sagledana je kao multiplikacija težine sintetičkog feature-a)[43][44] kao vid *finetunning-a* (finog podešavanja) zarad eliminisanja efekta gubitka, ovo omogućava po principu presecanja da se ispoljavaju različiti ishodi regularizacija naspram drugih features-a. Klasifikacije linearnih SVM klasifikatora rezultuju drugačijim ocenama od ostala dva SVM klasifikatora.

Tu su, takođe, dostupni primeri *nelinearnih SVM-ova* (koji koriste RBF kernele - kasnije će se pominjati zarad rešavanja problema linearne neseparabilnosti, prikaz na slici 16.)[45], *SVM-Anova*(rad sa šumovima od uzoraka pri klasifikaciji)[46].

Pri nebalansiranim problemima, gde je poželjno dati više važnosti određenim klasama ili određenim posebnim uzorcima, tu se dodeljuje težinski faktor. SVM klasifikator sa sobom može da nosi težinski faktor klase pri metodi obuke uz multiplikaciju parametra kažnjavanja, slika 17. predstavlja granicu odlučivanja za nebalansirajući problem sa i bez uticaja težinskog faktora.



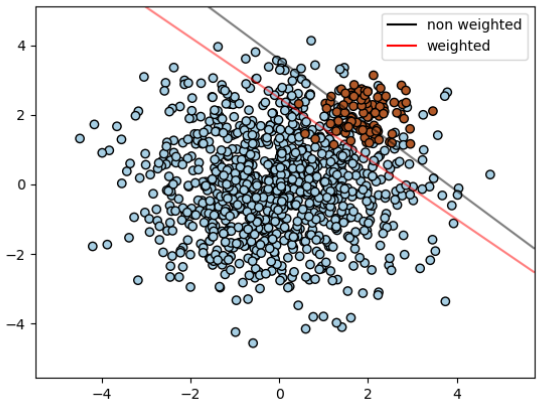
Slika 14: 2 feature-a iris skupa podataka upoređena po različitim režimima



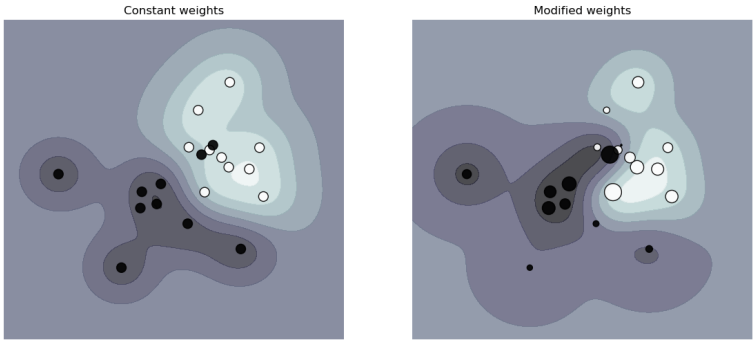
Slika 15: Primer funkcije odlučivanja obučene nelinearnim SVM-om

SVM, nuSVM, linearni SVM klasifikatori (prikazani na na slici 15.), jednoklasni SVM metoda (zaslužna za outlier detekcije) koriste pri obuci težinski faktor uzoraka. Ohrabriće se klasifikator težinskim faktorom uz multipliciranje nekim parametrom pozitivne decimalne vrednosti kažnjavanja da vrši prave klasifikacije nad uzorcima. Prikaz na slici 18. je efekat težinskog faktora uzoraka po granici odlučivanja.

SVM su snažni alati, ali zahtevnosti sračunavanja i skladišta rapidno se uvećavaju po broju vektora obuke. Srž SVM-a je *kvadratni programerski problem (QP - namenjen za rad od bilinearnih do (ne)jednačinama polinoma 2. stepena, korišćen za obradu slika i signala)*[49],



Slika 16: Granice odlučivanja pri nebalansiranom problemu uz/bez težinskog faktora klase



Slika 17: Granice odlučivanja pri nebalansiranom problemu bez/uz težinski faktor uzoraka

izdvajajući potporne vektore od ostalih trening podataka. QP rešavač može se skalirati u granicama $O(n_{features} \times {n_{uzoraka}}^2)$ prema $O(n_{features} \times n_{uzoraka}^3)$, ako su podaci vrlo raspršeni $n_{uzoraka}$ bi trebalo biti zamenjeni sa *prosečnim brojem nenula features-a u vektoru uzoraka.*

Za linearne slučajeve korišćen je linearni SVM klasifikator uz LIBLINEAR rešavač koji je pogodan da se skalira skoro linearno na milione uzoraka i/ili features-a.

U praktičnom smislu preproučuje se da se izbegavaju kopije podataka pri radu sa linearnim klasifikatorima, oprez sa veličinom cache memorije pri odabiru kernela, sagledanje parametra za rad sa težinskim faktorima u slučaju da su šumoviti uzorci u skupu podataka (vremenska složenost može biti uvećana u suprotnom), da se skaliraju prodaci korišćenjem pipeline-a[50] - predviđenog za sekvencijalno transformisanje i obradu podataka kroz sve do konačnog predviđača, sagledanje parametra sužavanaja za umanjenje vremena udeljenog obuci, parametar aproksimiranja grešaka pri obuci, usaglašavanja parametra kažnjavanja sa ne(balansiranosti) podataka, podešavanja nasumičnosti, podešavanja L1 kažnjavanja linearnom SVM klasfikatoru.

Kernel funkcije koje su dostupne su:

- linearni : $\langle x, x' \rangle$ - skalarni proizvod vektora između tačaka uzoraka,

- polinomijalan : $(\gamma\langle x, x'\rangle + r)^d$, stepena d i posebnog koeficijenta r,
- RBF : $\exp(-\gamma\|x - x'\|^2)$ - predstavljen parametrom $\gamma > 0$,
- sigmoidna : $\tanh(\gamma\langle x, x'\rangle + r)$.

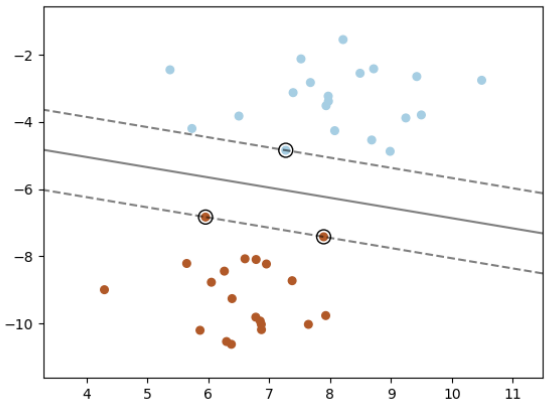
Radial Basis-a funkcije - RBF kernel ima 2 parametra koji su sagledani[51]:

- C - raspolaže izbegavanjem obukom uzorcima loše klasifikacije u odnosu na površ odlučivanja za sve SVM kernele; uspostavlja glatkost površi odlučivanja; visoka vrednost je namenjena za sve uzorke da obuka bude neprikosnovena) i
- γ - ustanovljava koliko ima uticaja jedan uzorak pri obuci. Višom vrednošću uzorci moraju biti bliži međusobno da bi bili prikladni.

Preporučuje se rad sa hiperparametrima za nalaženje pogodnih vrednosti parametara. Neki sistemi omogućuju definiciju prerađenih kernela. Moguće je za kernele koristiti Gram matricu G (za skup V od m vektora elemenata iz $\mathbb{R}^n$ ona sadrži skalarne proizvode među vektorima iz V, tj. $g_{i,j} = v_i^T v_j$; očuvava dužine izometrijom nakon transformacija)[52][53].

2.6 Metodologija

SVM grade hiper-ravan ili skup hiper-ravni u visoko ili beskonačno dimenzionalnim prostorima. Intuitivno, dobra separacija je postignuta kada za hiper-ravan po najvećoj distanci bliskoj uzorcima bilo koje klase, tzv. *funkcionalnoj margini*. Uopšteno, većom marginom dobija se niža generalizacija greškom klasifikatora. Slika 19. daje prikaz linearno separabilnog probleima 3 tačke predstavljaju uzorke na granicama margine, tzv. *potpornim vektorima*.



Slika 18: Granice margine, potrporni vektori nad vizuelizacijom skupa podataka

Kada problem je linearno **neseparabilan** onda se uzorci nalaze u granici margina kao potporni vektori.

Datim vektorima obuke $x_i \in \mathbf{R}^p, i = \overline{1,n}$, u 2 klase vrednosti vektora $y \in \{-1, 1\}^n$, cilj je naći $w \in \mathbf{R}^p$ i naći $b \in \mathbf{R}^n$ tako da predviđanje je dato po $\text{sign}(w^T\phi(x) + b)$ je tačan za većinu uzoraka.

SVM klasifikacija rešava **glavni problem**:

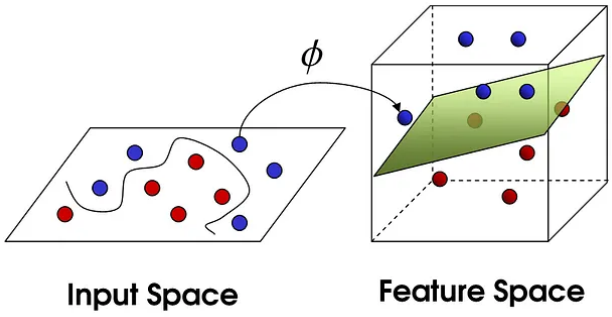
$$\begin{aligned} &\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \\ \iff &y_i(w^T \phi(x_i) + b) \geqslant 1 - \zeta_i, \\ &\zeta_i \geqslant 0, i = 1, ..., n \end{aligned}$$

Intuitivno, poželjno je maksimizovati margine (minimizacijom $||w||^2 = w^T w$), pritom nezgodno kažnjavanje se dešava pri netačnoj klasfikaciji uzorka ili pri upadanju granicu margina. Idealan slučaj je da važi $y_i(w^T \phi(x_i) + b) \geqslant 1$ za sve uzorke, ukazuje na perfektnu predikciju. Problemi nisu uvek savršeno separabilni sa hiper-ravnima, pa dopuštamo nekim uzorcima da budu na distanci ζ_i naspram njihove tačne granice margine. Kazneni parametar C upravlja jačinom kazne i kao rezultat, deluje kao **parametar** inverzne **regularizacije**.

Sporedni problem naspram glavnog je:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \iff &y^T \alpha = 0 \\ &0 \leqslant \alpha_i \leqslant C, i = 1, ..., n \end{aligned}$$

gde $e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$, i Q je polu-definitna matrica (ako su joj sve jedinične vrednosti nenegativne, vršenim normiranjem)[54]. $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, gde $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ je **kernel**. *Dualni koeficijenati* su α_i, gde $\alpha \leqslant sup((-\infty, C))$, i ona ispoljava činjenicu da vektori obuke su implicitno mapirani na (možda beskonačne) dimenzionalne prostore po funkciji ϕ, tj. **kernel trik** (prikazan na slici 20.).



Slika 19: Kernel trik

Čim je problem optimizacije razrešen, izlaz **funkcije odlučivanja** za dati uzorak biće:

$$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b$$

i predviđena klasa predodređena je njenim znakom. Moramo samo sumirati potporne vektore (tj. uzorke koji leže na margini) pošto **dualni koeficijenti su za ostale uzorke** $\alpha_i = 0$.

Pojedini optimizatori C parametar kažnjivanja označavaju sa *alpha*. Tačna jednakost količine regularizacije 2 modela zavisi od tačne funkcije cilja optimizacirane, gde za model u nekim primerima je $C = \frac{1}{alpha}$.

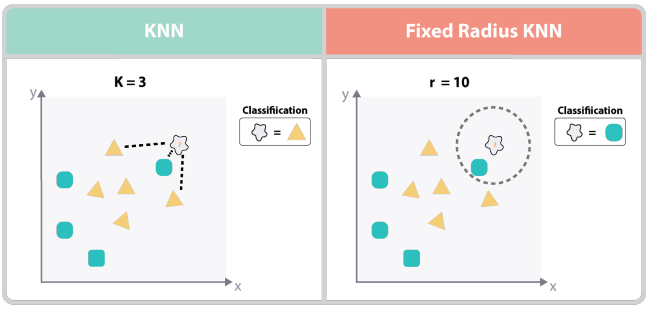
Linearni SVM klasifikator ima za svoj oblik svoj glavni problem za rešavanje:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \max(0, 1 - y_i(w^T \phi(x_i) + b)),$$

važno je da se postavi primena hindža[42]. Forma je direktno optimizovana po linearnoj SVM klasfikaciji, ali za razliku od dualne forme, ova ne uključuje skalarne proizvode među uzorcima, pritom kernel trik nije doveden u obzir. Ovo je bio razlog zašto linearni kernel je jedino podržan od ovog klasfiikatora (ϕ je funkcija idenititeta).

2.7 K-najbližih suseda (kNN)[41]

Ustupaju funkcionalnost za metode (ne)nadgledane obuke zasnovane na susedima. Nadgledano obučavanje zasnovano na susedima ima dve vrste, od kojih je klasifikaciju za podatke diskretnih labela, a drugi regresija. Princip iza metoda najbližih suseda (NN) je naći predefinisan broj najbližih trening uzoraka u odstojanju od tačke novog uzorka, pa tako predvide labelu iz trening uzoraka. Broj uzoraka može biti ili korisnički definisana konstanta k (kNN), ili samo da varira po gustini lokalnih tačaka (*NN zasnovane po radiusu korisnički definisanog* r), prikazane na slici 21.



Slika 20: k-najbliži sused i radius zasnovan najbliži sused klasifikacija za neki skup podataka

Distanca može imati bilo koju metričku meru: *standardna Euklidska distanca* je najobičniji izbor. Metode zasnovane na susedima su poznate kao *negeneralizujuće* metode mašinskog učenja, pošto prosto "se sete" svih podataka iz trening skupa (moguće transformisanih u brze sturkture indeksiranja kao što su *KD Tree* ili *Ball Tree* o kojima će biti reči kasnije).

Naspram njene jednostavnosti, NN su se pokazali kao uspešni pri velikom broju klasifikacionih problema, uključujući rukopise cifara i scene satelitskih snimaka. Kao neparametarski metod, uspešno radi klasifikacije i pri neregularnim granicama odlučivanja.

Za matrice kao ulazne podatke se ustupa:

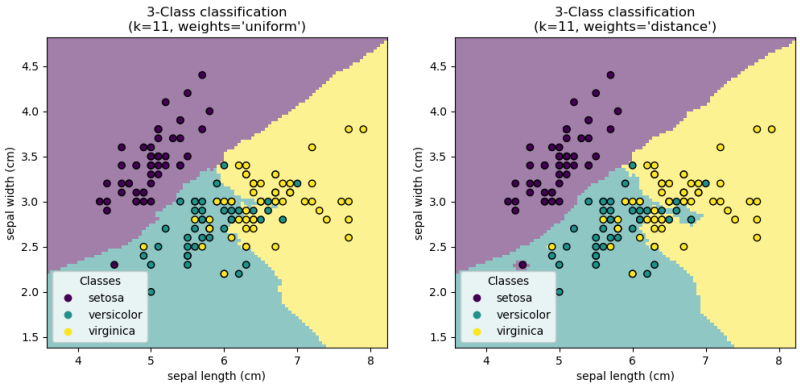
- Za guste matrice, veliki broj je podržanih metrika distance.
- Za raspršene arbitrarne metrike Mikovskog su podržane : npr. $d_{\mathbf{p}} : (x, y) \mapsto \|x - y\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$.[56][57]

Tu su većina rutina obuke koje se oslanjaju na NN u njihovoj srži, kao što je *procena gustina kernelom.*

Klasifikacije zasnovane na susedu su *tipa obuke zasnovane na uzorcima/instancama kao negeneralizujuće*: ne pokušava da gradi opšte unutrašnje modele, ali jednostavno skladišti uzorke trening skupa. Klasifikacija sračunava po prostom "glasanju" većine najbližih suseda: tačke upita dodeljene klase podataka koje imaju najviše zastupljenika od najbližih suseda tačke. KNN je najčešće korišćena tehnika, po kojoj od optimalnog izbora vrednosti k postoji jaka zavisnost podataka - veće k potiskuje efekat šuma, ali pravi ograničavanja klasifikacija manje nezastupljenim. Za uniformno uzorkovane podatke predlaže se NN zasnovan na radiusu, td. je lakše u situaciji kada su tačke uzoraka raspršene. Za visoku dimenzionalnost parametarskih prostora, ovaj metod postaje manje efikasan zbog *"prokletstva dimenzionalnosti"*.

U nekim slučajevima je bolje *udeliti težine susedima* tako da što bliži susedi doprinose više pri poklapanju obuke, a one mogu biti *uniformne*, *na osnovu distance* (proporcionalno

inverzna distanci tačke upita) i *korisnički definisana funkcija*. Priložen je primer na slici 22. koja prikazuje ponašanje težinskih faktora.



Slika 21: kNN klasifikacija skupa podataka iris od 3 klase, gde je $k = 11$ u 2 slučaja težinskih faktora uniformnih i na osnovu distance

3 Diskusija o algoritmima

Svaki algoriritam ima sopostvene osobine koji mogu da se iskoriste posebne probleme ili ne moraju biti baš dobar izbor za neki drugi slučaj korišćenja. Tako se uvek oslanjamo na isprobavanje drugačijih algoritama da bi se srela što bolja tačnost i pouzdan model za određen projekat, kao što je prikazano na slici 31.[66]

3.1 Prednosti i mane algoritama, primena[67][68][69][70][71]

3.1.1 Logistička regresija

Prednosti:

- Lak, brz, jednostavan klasifikacioni metod;
- β parametar objašnjava smer i intenzitet važnosti nezavisnih promenljivih naspram zavisnih promenljivih;
- Može biti korišćen za višeklasne klasifikacije isto;
- Funkcija gubitka (uglavnom cross-entropija) je uvek konveksna;
- Vrši efikasna predviđanja na linearno separabilnim skupovima podataka;
- Može se izboriti sa (ne)linearnim srodnostima između features-a i cilja (možda ne?);
- Dobar za binarne klasifikacije;
- Moguća je regulacija zarad izbegavanja overfitting-a;
- Moguća je brza ažuriranja novim podacima.

Mane:

- Ne može biti primenjen na nelinearnim klasifikacionim problemima;
- Adekvatna selekcija features-a je obavezna;
- Očekuje se dobar SNR (signal-šum proporcija);
- Kolinearnost i diskiriminativnost (outliers-i) štete tačnosti modela;
- Moguća je samo primena diskretnih promenljivih pri predviđanju;
- Podleže overfitting ako je skup podataka mali;
- Nagađa da su features-i i ciljevi linearno srodni, što ne mora uvek da znači.

Pogodni slučajevi primene:

- Kada odnos između features-a i cilja je jednostavan i linearan;
- Kada efikasnost složenosti izračunavanja je bitna.

Sveukupno viđenje: (upitna tačnost)

- Izlaz je verovatnoća;
- Granica odluke - linearna;
- Osetljiva na outliers-e;
- Osetljiva na overfitting i underfitting (zanemarivanje objektivno značajnog broja uzoraka klase)[72];
- Nagađanja da podaci su linearni, normalizovani;
- Složenost modela - niska;
- Interpretabilnost (ne zahteva fino podešavanje od strane korisnika)[73] - visoka;
- Lakoća implementacije - visoka.

3.1.2 Stabla odlučivanja

Prednosti:

- Predobrada podataka nije potrebna (pri manju parametrizacija, nedostajućim vrednostima);
- Nema nagađanja u vezi distribucije podataka;
- Obrađuje kolinearnost (uzorke duplikate)[74] efikasno;
- Ustupaju dobro pojašnjenje predviđanja (mimikuju ljudsku logiku zarad toga);
- Efikasni u dohvatanju međusobno nelinearnih features-a podataka;
- Mogu raditi sa kategoričkim i numeričkim podacima;
- Efikasnost uspostavljena po algoritmu pretraga stabla;

Mane:

- Mogućnost overfitting-a modela ako se iznova gradi stablo da bi se dosegla visoka čistost, tj. može doseći preterano veliku dubinu - za koje je uz odsecanje stabala ili korišćenje slučajnih šuma su moguća rešenja ovog problema;
- Štete mu outliers-i - svakim drugačijim rukovanjem njima moguće je dobiti vrlo različite oblike stabala;
- Može se izgraditi u veoma složen model pri radu sa komplikovanim skupovima podataka (višedimenzionalnim naspram broja features-a);
- Gubi vredne informacije dok radi sa kontinualnim vrednostima;
- Može biti previše oslonjen (da ima bias) na features-e sa vrednostima više kategorija.
- Znaju davati lokalno optimalne rezultate, umesto idealno globalne.

Pogodni slučajevi primene:

- Kada srodnost između features-a i ciljanih vrednosti je složena i nelinearna;
- Interpretabilnost je važna;

Sveukupno viđenje:

- Izlaz je labela klase;
- Granica odluke je nelinearna;
- Osetljiva na outliers-e;
- Osetljiva na overfitting i underfitting;
- Složenost modela - srednja;
- Nagađanja o svojstvima podatka nema;
- Interpretabilnost - visoka;
- Lakoća implementacije - visoka;

3.1.3 Slučajne šume

Prednosti:

- Tačan i moćan model;
- Rukuje overfitting-om efikasno;
- Podržava implicitan odabir feature-a i izvodi važnost feature-a;
- Kombinuje više stabala, pa je pouzdaniji od stabala odlučivanja;
- Radi sa kategoričkim i numeričkim podacima;
- Efikasni u dohvatanju međusobno nelinearnih features-a podataka.

Mane:

- Složeniji i sporiji pri izvršavanju kada šuma postane veća;
- Nisu dobar opisni model naspram predviđanja;

Pogodni slučajevi primene:

- Kada srodnost između features-a i ciljanih vrednosti je složena i nelinearna;
- Kada je overfitting problematika kod drugih klasifikatora.

Sveukupno viđenje:

- Izlaz je labela klase;

- Granica odluke je nelinearna;
- Nije osetljiva na outliers-e (Možda da?);
- Nije osetljiva na overfitting i underfitting;
- Nagađanja o svojstvima podatka nema;
- Složenost modela - visok;
- Interpretabilnost - niska;
- Lakoća implementacije - visoka;

3.1.4 Naivni Bajes

Prednosti:

- Radi dobro sa manjim skupom podataka;
- Ako je korišćen model nezavisan od uslova, onda konvergira brže od drugih modela;
- Rukuje redudantnim features-ima (nepoznatim vrednostima);
- Podržava binarne i multiklasifikatorne probleme klasfikacija;
- Lak pri implementaciji;
- Može rukovati podacima visokedimenzionalnosti;
- Dobar za probleme klasifikacije teksta;
- Radi sa ili kategoričkim ili numeričkim podacima;
- Pravi dobre stohastičke preporuke.

Mane:

- Očekuje da features-i budu strogo međusobno nezavisni - što nije baš primenljivo u stavrnom životu;
- Uzorci trening skupa velike brojnosti i $P(X = feature|Y) = 0$ za feature, posterior može postati 0. Što nahodi da taj uzorak može biti loš zastupnik u populaciji - tj. osetljiv na redudantne vrednosti;
- Kontinualne promenljive su sumirane segmentima (binning) da ukažu na diskretne vrednosti features-a. Ovaj zadatak je neophodno raditi pažljivo da ne bi došlo do gubitka podataka i netačnosti.
- Previše su jednostavni modeli;
- Nije pogodan u višedimenzionalnim slučajevima;
- Vreme izvršavanja je složenije pri predviđanju naspram SVM i logističke regresije.

Pogodni slučajevi primene:

- Kada su features-i međusobno nezavisni i imaju ograničenja trening skupa podataka;
- Pri klasifikaciji teksta, spam filtriranja, sistema preporuka, itd.
- Kada su visokodimenzionalni? - Ne baš.

Sveukupno viđenje:

- Izlaz je labela klase;
- Granica odluke je linearna;
- Nije osetljiva na outliers-e;
- Nije osetljiva na overfitting i underfitting;
- Nagađanja o svojstvima podatka je u vezi nezavisnosti;
- Složenost modela - niska;
- Interpretabilnost - visoka;
- Lakoća implementacije - visoka;

3.1.5 Metode potpornih vektora (SVM)

Prednosti:

- Koristi kernel trik da ustupi kompleksna rešenja;
- Koristi funkciju konveksne oprimizacije kojom se uvek dostiže globalni minimum;
- Hindž gubici su visoke tačkosti;
- Outliers-i mogu biti obrađeni korišćenjem mekših margina konstantom C;
- Rade sa visokodimenzionalnim podacima;
- Efikasni u dohvatanju međusobno nelinearnih features-a podataka.
- Rade dobro sa malim skupovima podataka.

Mane:

- Hindž gubitak vodi u raspršenost;
- Hiperparametrima i kernelima je važno pažljivo raditi podešavanja za podnošljivu tačnost;
- Većim skupovima podataka duže je vreme obuke;
- Interpretabilnost složenija;
- Može biti osetljiva na odabir parametara regularizacije C (što je teško za utvrditi), i kernela;

- Sračunavanja verovatnoća nisu ustupljene ovim modelom.

Pogodni slučajevi primene:

- Kada srodnost između features-a i ciljanih vrednosti je složena i nelinearna;
- Kada se radi sa manjim skupom podataka.

Sveukupno viđenje:

- Izlaz je labela klase;
- Granica odluke je nelinearna;
- Nije osetljiva na outliers-e;
- Osetljiva na overfitting i underfitting;
- Nagađanja o svojstvima podatka nema;
- Složenost modela - visoka;
- Interpretabilnost - niska;
- Lakoća implementacije - niska;

3.1.6 k-Najbližih suseda

Prednosti:

- Lak za izgradnju i jednostavan za razumevanje model mašinskog učenja;
- Malo hiperparametara za podešavanje;
- Laka implementacija;
- Dobar za male skupove podataka;
- Koristan pri klasifikaciji višelabelnim vrednostima klasa;
- 2 puta bolje ustanovljavanje netačnosti naspram naivnih Bajesa;
- Npr. pri utvrđivanju svojstava proteina, profili istaknuti su bolje na osnovu funkcija, time prednjači naspram SVM pristupa.

Mane:

- k je neophodno pažljivo odabrati;
- Viša složenost izračunavanja ako je ogroman broj uzoraka;
- Neophodno je skaliranje predobradom features-a, da bi ulaz bio podnošljiv;
- Ostetljiv na redudantne features-e, klasifikacija nepoznatih vrednosti je skuplja;
- Zahteva sračunavanja distanci k suseda;

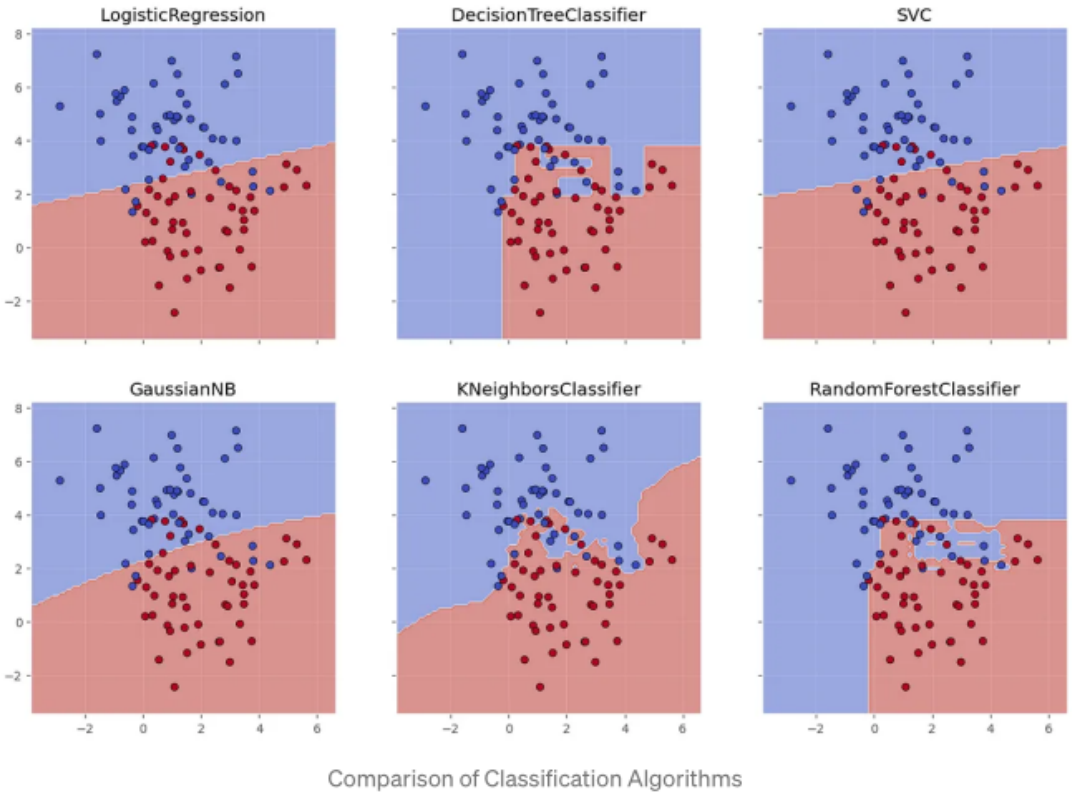
- Tačnost ispašta šumom i redudantnim vrednostima ;

Pogodni slučajevi primene:

- Kada je skup podataka manji;
- Kada je interoperabilnost važna.

Sveukupno viđenje:

- Izlaz je labela klase;
- Granica odluke je nelinearna;
- Osetljiva na outliers-e;
- Nije osetljiva na overfitting i underfitting;
- Nagađanja o svojstvima podatka nema;
- Složenost modela - niska;
- Interpretabilnost - visoka;
- Lakoća implementacije - visoka;



Slika 22: Poređenje klasifikacija određenih modela

3.2 Poređenje među algoritmima[67][?]

Radiće se kombinacije po redosledu : LR, Stabla odlučivanja, kNN, SVM, a pritom NB će već biti obrađen.

3.2.1 Logistička regresija

Logistička regresija vs SVM:

- Sveukupno viđenje:SVM rukuje nelinearnim rešenjima, dok logistička regresija linearnim;
- Sveukupno viđenje:Linearni SVM rukuje bolje outliers-ima, kako izvodi rešenje maksimalne margine;
- Sveukupno viđenje:Hidž sračunat gubitak kod SVM-a prevazilazi log loss kod logističke regresije.

Logistička regresija vs Stablo odlučivanja

- Sveukupno viđenje:Stabla odlučivanja rukuju kolinearnošću bolje nego logistička regresija.
- (bolji) Stabla odlučivanja ne mogu izvesti važnosti features-a, ali logistička regresija može;
- Sveukupno viđenje:Stabla odlučivanja se pokazuju bolje pri radu nad kategoričkim vrednostima od logističke regresije.

Logistička regresija vs Naivni Bajes

- NB je generativni model gde LR je diskiriminativni;
- NB radi dobro sam malim skupovima podataka, gde LR uz regularizaciju možde dostići slične performanse;
- LR radi bolje od NB pri slučaju kolinearnosti, dok NB očekuje da svi features-i budu nezavisni.

Logistička regresija vs kNN

- kNN je neparametarski model, dok LR je parametarski model;
- kNN je uporedivo sporiji od LR;
- kNN podržava nelinarna rešenja dok LR podržava samo linearna;
- LR može generisati nivo pouzdanosti predviđanja, dok kNN samo

3.3 Stabla odlučivanja

Sličan ishod upoređivanja Stabla odlučivanja je i za upoređivanje sa Slučajnim šumama, ali evo poređenja ova dva.

Stabla odlučivanja vs Slučajne šume

- Slučajna šuma je skupina stabala odlučivanja i prosečno/većinsko glasanje šume je uzeto kao ishodujuće predviđanje;

- Slučajna šuma je manje oštećen overfitting-om od stabla odlučivanja, da je više generalizovanije rešenje;
- Slučajna šuma je pouzdanija i tačnija od stabla odlučivanja.

Stabla odlučivanja vs Naivni Bajes

- Stablo odlučivanja je diskiriminativni model, gde Naivni Bajes je generativni;
- Stablo odlučivanja je fleksibilniji i lakši;
- Stablo odlučivanja uz odsecanje mogu zanemariti ključne vrednosti pri obuci podataka, što tačnost dovodi u rizik.

Stabla odlučivanja vs SVM

- SVM koristi kernel trik da reši probleme nelinearnosti dok stablo odlučivanja koristi hiperpravougaonike da bi rešio problem ulaznog prostora;
- Stbla odlučivanja su bolja za kategoričke podatke i rukuju sa kolinearnostima bolje od SVM-a.

Stablo odlučivanja vs kNN

- Oba su neparametarski metodi;
- Stablo odlučivanja ima podršku za automatsku interakciju feature-a, dok kNN nema.
- Stablo odlučivanja je brže od kNN-ovog skupog vremenski složenog izvršavanja.

3.3.1 k-Najbližih suseda(kNN)

kNN vs NB

- NB je mnogo brži od kNN po vremenskoj složenosti;
- NB je parametarski, dok kNN je neparametarski.

kNN vs SVM

- SVM brine više o outliers-ima nego kNN;
- Ako trening skup $n_{uzoraka} >> n_{features}$, kNN je bolji od SVM, inače SVM prevazilazi kNN.

3.3.2 Metode potpornih vektora (SVM)

SVM vs Slučajna šuma

- Slučajna šuma podržava višeklasne klasifikacije, gde SVM mora da obezbedi više modela za to;
- Slučajna šuma daje verovatnoću naspram predviđanja, dok SVM to ne radi;

- Slučajna šuma rukuje bolje kategoričkim podacima od SVM.

SVM vs NB

- Oba rade bolje sa manjim trening skupom i većim brojem features-a;
- Ako su međusobno zavisni features-i, SVM prevazilazi NB;
- SVM je diskiriminativni model, dok NB je generativni model.

Dodatna sagledanja[66]

Za **brzinu pri obuci** oslanja se na LR, NB klasifikatore (slučajne šume su spore za to). kNN je uporedivo **generalno sporiji** od LR, NB i sporiji od stabala odlučivanja.

kNN je skup što se tiče **memorije pri obuci** pošto mora da vodi računa o svim trening podacima i o tome da treba da nađe čvorove suseda.

NB radi dobro sa malim skupovima podataka.

Fleksibilnost nije prisutna kod LR toliko da može da uhvati složenije srodnosti features-a. Stablo odlučivanja podržava nelinearnost. SVM podržava oba (ne)linearna rešenja. kNN je bolji od LR kada podaci imaju visok SNR. Slučajne šume su pouzdanije i tačnije od stabala odlučivanja.

4 Zaključak

Na početku su pokušana izlaganja objašnjenja za stukturu skupa podataka uz razjašćanjavanje (šta je učenje, šta se najčešće koristi, šta su zahtevi rada pri nadgledanom učenju) i postepeno razgraničenje pojmova sadržanih (uzorak, atribut/feature, labela po brojnosti, klasa po brojnosti, trening i test skup), načina obuke sve do klasifikacija.

U teorijskim osnovama i metodologijama redom su istaknute teme logičke regresije, stabla odlučivanja, slučajne šume, naivnog Bajesa, SVM i kNN-a.

Za logičku regresiju pomenuti su koncepti logit funkcije (logaritam neobičnosti) uz linearne kombinacije kojom može se predstaviti, klasifikacije maksimuma entropije zasnovanom na verovatnoći i na slučajnom odabiru uzorka, log-linearnom klasifikatorom koji je nalik logit funkciji samo preoblikovanom, pa onda podela na verovatnoće ishoda tipa binarna (koeficijenta regulacije, korisnički dodeljenih težina uzoraka/klase, funkcija procene) i OVR, a tu su multiunomijalne logističke regresije (po log-linearnim modelima, prekomerne parametrizacije i simetričnog induktivnog bias-a, optimizacije), kao l_1, l_2 i regularizacija elastične mreže.

Za stabla odlučivanja koji je uvelo pojmove neparametarskog obučavanja, isparčane konstantne aproksimacije, pravila odlučivanja (dato if-then-else), pomenuto smanjenje dimenzionalnosti uz PCA (uz njega rastuće ortogonalne komponente, ICA, feature selection procesi), dat je uvid u slučaju multilabel problema i predstavljen je koncept istovremenog predviđanja, dat je primer njegove upotrebe pri upotpunjavanju dela slike lica na osnovu predviđanja. Data je analiza složenosti i pojma balansiranog stabla, podstabla (aproksimativno) balansirajućeg, kriterijum nečistosti, logaritamski gubitak, information gain. Data matematička formulacija kao predstavljanje metodologije, uz principe podele praga, funkcije gubitka(gini, entropija/log-loss), kriterijuma klasifikacije, Shannon entropije, cross-entropije. Predstavljen je koncept obrade nepostojanih vrednosti (pri nailazećim podelama, neizvesnosti, itd.). Rešenje dato za problem situacije overfitting-a kao što je odsecanje minimalnom cenom složenosti i parametra za postavljanje ccp_alpha.

Za slučajne šume kao metod ansambla ukazano je na to da se oslanjaju na ustanovljavanje proseka tačnosti i upravljanja overfitting-a. Uključeni su koncepti "pomešati i kombinovati", koncept nasumičnosti, bootstrap uzorka. Ukazuje se blag na porast cene u bias-u. Ukazivanje na parametre brojnosti komponenti stabla i ishoda, ukazivanje na rešenje cross-validation-om. Daje se uvid u bootstrap bagging agregacije za procenjivanje grešaka generalizacija. Ističe se pojam važnosti feature-a (pominje se za feature selection mera središnjeg opadanja u nečistosti MDI i ukazuje se na 2 zablude i među njima se ukazuje na koncept permutacije važnosti features-a).

Za klasifikaciju naivnim Bajesom ukazuje se na pojmove Bajesove teoreme i naivnim pretpostavkama. Bitni koncepti su prior, evidence, likelihood, posterior verovatnoće, nezavisnosti događaja, uslovne nezavisnosti, verovatnoća maksimalnog a posteriori - MAP, Gausova verzija - koja koristi koncepte Gausove distribucije, i verovatnoće maksimalne verodostojnosti - MLE, multinomijalna (MNB) verzija za 2 ili više vrednosti nezavisnih features-a uz parametar smooth-ovanja MLE-a (slučaja Laplace i Lidstone), komplementarni naivni bajes (CNB) kao proširenjem MNB-a, Bernulijeva verzija (BNB) koji koristi višestruku Bernilijevu distribuciju

koji eksplicitno kažnjava neispoljavanje feature-a, i kategorički NB je zasnovan na kategoričkoj distribuciji. Dat je uvid u problem klasifikacije velikih razmera i koncepta out-of-core klasifikacije.

Za metode potpornih vektora (SVM) dat je uvid u njegove karakteristike, verzije nu-SVM, linearnog SVM-a kom je udeljen linearni kernel, gubici kvadratnih hindževa, kombinacija LIBLINEAR rešavačima, skaliranja presecanja, finetunning slujave, verziju SVM-a koji je nelinearan RBF kernelom i koji rešava problem linearne nesaparabilnosti. Dalje se pominju težinskih faktora klase i uzoraka, kvadratnog programerskog problema kao srži SVM-a, itd. Date su kernel funkcije linearnba, polinomijalne, Radial-basis funkcija - RBF (koja sa sobom zahteva 2 parametra: C za regularizaciju, tj. izbegavanje uzoraka koji loše se klasifikujum; γ za ustanovljavanja uticaja jednog uzorka), sigmoidna, mogućnost prepravljanja funkcija kernela uz korišćenje Gram matica. Potom se ide na glavnu stvar gde se pominju bitni koncepti hiperravni, funkcionalna margina separacija uzoraka, granica margine, potpornih vektora, glavnog problema maksimizovanja margina, kažnjavanja, parametra izverzne regularizacije, sporednog problema, kernela, dualnih koeficijenata, kernel trika ϕ, funkcije odlučivanja kao problema optimizacije, itd. Za linearni SVM klasifikator se pominje njegov glavni problem za rešavanje, primena hindž funkcije gubitka, nekorišćenje kernel trika. Priče o verziji nu-SVM-a uspostavljanja gornje granice frakcija gešaka i donje granice frakcija potpornih vektora.

Za kNN je ustanovljen princip nalaženja broja najbližih suseda trening uzoraka u odstojanju od tačke novog uzorka, gde je k unapred korisnički postavljena brojka ili samo zavisi od radijusa (poluprečnika) korisnički postavljenog r. Obraća se pažnja na indeksiranja pri transoformisanja negeralizujuće metode, primenu kod rukopisa cifara, scene satelitskih snimaka. Naglašava se da je neparametarski i radi klasifikacije u neregularnim granicama odlučivanja gusta, raspršena; da se vrše procena gustina kernelom kao srž većina NN rutina. slučaja negeralizujuće obuke i skladištenja, tačke upita, koncepta glasanja, prokletstva dimenzionalnosti, težina udeljenih susedima.

U 3. sekciji obavljena je diskusija algoritama šta su povoljnosti, šta nepogodnosti (težina implementacije, brzina izvršavanja, jednostavnost interpretacije, inteoperabilnost, povoljnost parametara koji se podešavaju unapred, adekvatnost funkcija gubitaka, kakav je odnos na linearnu separabilnost, itd.), situacija kada je poželjna primena, sveukupno sagledanje (vrsta vrednosti izlaza labele, linearnost granica odluke, osetljivost na outliers-e, osetljivost na underfitting i overfitting, složenost modela, nagađanja o svojstvima podataka, interoperabilnost, lakoću implementacije) za svaki opšte pomenut algoritam.

Dalje su data neka usoprotstavljena poređenja algoritama po nekim osobinama (gori, nerešeno, bolji).

Literatura

[1] An introduction to machine learning with scikit-learn, https://scikit-learn.org/stable/tutorial/basic/tutorial.html, Datum poslednjeg pristupa: 20. mart 2024.

[2] Semi-supervised learning, https://scikit-learn.org/stable/modules/semi_supervised.html, Datum poslednjeg pristupa: 20. mart 2024.

[3] Introduction to RL and Deep Q Networks, https://www.tensorflow.org/agents/tutorials/0_intro_rl, Datum poslednjeg pristupa: 20. mart 2024.

[4] 5 Classification Algorithms for Machine Learning, https://builtin.com/data-science/supervised-machine-learning-classification, Datum poslednjeg pristupa: 20. mart 2024.

[5] R Documentation, The logit and inverse-logit functions, https://search.r-project.org/CRAN/refmans/LaplacesDemon/html/logit.html, Datum poslednjeg pristupa: 20. mart 2024.

[6] Understanding Logit: The Link Function in Logistic Regression, https://deepai.org/machine-learning-glossary-and-terms/logit, Datum poslednjeg pristupa: 20. mart 2024.

[7] JavaTpoint, Entropy in Machine Learning https://www.javatpoint.com/entropy-in-machine-learning, Datum poslednjeg pristupa: 20. mart 2024.

[8] Soczewica R., 2021., When should we use the log-linear model?, https://towardsdatascience.com/when-should-we-use-the-log-linear-model-db76c405b97e, Datum poslednjeg pristupa: 20. mart 2024.

[9] Brownlee J., 2021., One-vs-Rest and One-vs-One for Multi-Class Classification, https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/, Datum poslednjeg pristupa: 20. mart 2024.

[10] Raman N., Magazzeni D., Shah S. 2023., Bayesian Hierarchical Models for Counterfactual Estimation, Figure 4, https://www.researchgate.net/figure/Inductive-bias-in-categorical-values-of-generated-counterfactuals-left-Posterior_fig2_367359800, Datum poslednjeg pristupa: 20. mart 2024.

[11] Schmidt M., Le Roux N., Bach F., Minimizing Finite Sums with the Stochastic Average Gradient. Mathematical Programming, 2017, 162 (1-2), pp.83-112. 10.1007/s10107-016-1030-6. hal- 00860051v2, https://inria.hal.science/hal-00860051/document, Datum poslednjeg pristupa: 20. mart 2024.

[12] sklearn linear model LogisticRegression, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression, Datum poslednjeg pristupa: 20. mart 2024.

[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, Chih-Jen Lin, 2022., LIBLINEAR: A Library for Large Linear classification

[14] P.J.T. Notsawo, 2023., Stochastic Average Gradient : A Simple Empirical Investigation, description https://arxiv.org/abs/2310.12771, Datum poslednjeg pristupa: 20. mart 2024.

[15] Defazio A., Bach F. 2014., SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives, description https://arxiv.org/abs/1407.0202v3, Datum poslednjeg pristupa: 20. mart 2024.

[16] Gilles-Philippe Paillé,2019., SciPy optimisation: Newton-CG vs BFGS vs L-BFGS, https://stackoverflow.com/questions/42424444/scipy-optimisation-newton-cg-vs-bfgs-vs-l-bfgs, Datum poslednjeg pristupa: 20. mart 2024.

[17] J. Bräuninger, 1980., A quasi-Newton method with Cholesky factorization, description https://link.springer.com/article/10.1007/BF02259641, Datum poslednjeg pristupa: 20. mart 2024.

[18] 1.10. Decision Trees, https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart, Datum poslednjeg pristupa: 20. mart 2024.

[19] Difference between Black Box Vs White Vs Grey Box Testing, https://www.geeksforgeeks.org/difference-between-black-box-vs-white-vs-grey-box-testing/, Datum poslednjeg pristupa: 20. mart 2024.

[20] Awati R., 2022., extrapolation and interpolation, https://www.techtarget.com/whatis/definition/extrapolation-and-interpolation, Datum poslednjeg pristupa: 20. mart 2024.

[21] baeldung, 2022., P, NP, NP-Complete and NP-Hard Problems in Computer Science, https://www.baeldung.com/cs/p-np-np-complete-np-hard, Datum poslednjeg pristupa: 20. mart 2024.

[22] Is there a proof to explain why XOR cannot be linearly separable?, https://ai.stackexchange.com/questions/25228/is-there-a-proof-to-explain-why-xor-cannot-be-linearly-separable, Datum poslednjeg pristupa: 20. mart 2024.

[23] Open question: The parity problem in sieve theory, https://terrytao.wordpress.com/2007/06/05/open-question-the-parity-problem-in-sieve-theory/, Datum poslednjeg pristupa: 20. mart 2024.

[24] 2.5.1. Principal component analysis (PCA), https://scikit-learn.org/stable/modules/decomposition.html#pca, Datum poslednjeg pristupa: 20. mart 2024.

[25] 2.5.6. Independent component analysis (ICA), https://scikit-learn.org/stable/modules/decomposition.html#independent-component-analysis-ica, Datum poslednjeg pristupa: 20. mart 2024.

[26] 1.13. Feature selection, https://scikit-learn.org/stable/modules/feature_selection.html#feature-selection, Datum poslednjeg pristupa: 20. mart 2024.

[27] Post pruning decision trees with cost complexity pruning, https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py, Datum poslednjeg pristupa: 20. mart 2024.

[28] sklearn.ensemble.RandomForestClassifier, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, Datum poslednjeg pristupa: 20. mart 2024.

[29] Tixier A.J.P., 2018., Perturb and Combine to Identify Influential Spreaders in Real-World Networks, https://arxiv.org/pdf/1807.09586.pdf, Datum poslednjeg pristupa: 20. mart 2024.

[30] Brownlee J., 2023. A Gentle Introduction to k-fold Cross-Validation, https://machinelearningmastery.com/k-fold-cross-validation/, Datum poslednjeg pristupa: 20. mart 2024.

[31] 1.11.2. Random forests and other randomized tree ensembles, https://scikit-learn.org/stable/modules/ensemble.html#forest, Datum poslednjeg pristupa: 20. mart 2024.

[32] 4.2. Permutation feature importance, https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance, Datum poslednjeg pristupa: 20. mart 2024.

[33] 6.2. Feature extraction, https://scikit-learn.org/stable/modules/feature_extraction.html, Datum poslednjeg pristupa: 20. mart 2024.

[34] 1.9. Naive Bayes, https://scikit-learn.org/stable/modules/naive_bayes.html, Datum poslednjeg pristupa: 20. mart 2024.

[35] Naïve Bayes Classifier, UC Business Analytics R Programming Guide, https://uc-r.github.io/naive_bayes, Datum poslednjeg pristupa: 20. mart 2024.1.4.4 Conditional Independence, https://www.probabilitycourse.com/chapter1/1_4_4_conditional_independence.php, Datum poslednjeg pristupa: 20. mart 2024.

[36] Pathak M., Normal Distribution, https://www.kaggle.com/code/themrityunjaypathak/normal-distribution?scriptVersionId=156424946, Datum poslednjeg pristupa: 20. mart 2024.

[37] Hosch L.W., 2024., multinomial distribution, Brittanica, https://www.britannica.com/science/multinomial-distribution, Datum poslednjeg pristupa: 20. mart 2024.

[38] Notes: Bernoulli, Binomial, and Geometric Distributions, https://my.eng.utah.edu/~cs3130/lectures/L07-BernoulliBinomialGeometric.pdf, Datum poslednjeg pristupa: 20. mart 2024.

[39] sklearn.preprocessing.OrdinalEncoder, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html#sklearn.preprocessing.OrdinalEncoder, Datum poslednjeg pristupa: 20. mart 2024.

[40] 1.4. Support Vector Machines, https://scikit-learn.org/stable/modules/svm.html#, Datum poslednjeg pristupa: 20. mart 2024.

[41] 1.6. Nearest Neighbors, https://scikit-learn.org/stable/modules/neighbors.html, Datum poslednjeg pristupa: 20. mart 2024.

[42] Kumawat N., 2020., Hinge Loss and Square Hinge loss, InsideAIML, https://insideaiml.com/blog/Hinge-Loss-and-Square-Hinge-loss-1068, Datum poslednjeg pristupa: 20. mart 2024.

[43] sklearn linear model LogisticRegression - intercept scaling argument, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression, Datum poslednjeg pristupa: 20. mart 2024.

[44] synthetic data, https://www.techtarget.com/searchcio/definition/synthetic-data, Datum poslednjeg pristupa: 20. mart 2024.

[45] Non-linear SVM, https://scikit-learn.org/stable/auto_examples/svm/plot_svm_nonlinear.html#sphx-glr-auto-examples-svm-plot-svm-nonlinear-py, Datum poslednjeg pristupa: 20. mart 2024.

[46] SVM-Anova: SVM with univariate feature selection, https://scikit-learn.org/stable/auto_examples/svm/plot_svm_anova.html#sphx-glr-auto-examples-svm-plot-svm-anova-py, Datum poslednjeg pristupa: 20. mart 2024.

[47] Chandra Prakash Bathula, 2023., Machine Learning Concept 68: Platt's Scaling, https://medium.com/@chandu.bathula16/machine-learning-concept-68-platts-scaling-b8245421739e, Datum poslednjeg pristupa: 20. mart 2024.

[48] Wu, Lin and Weng, 2004., Probability estimates for multi-class classification by pairwise coupling, https://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf, Datum poslednjeg pristupa: 20. mart 2024.

[49] Heider J., 2015, Quadratic programming, https://optimization.cbe.cornell.edu/index.php?title=Quadratic_programming, Datum poslednjeg pristupa: 20. mart 2024.

[50] sklearn.pipeline.Pipeline, https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline, Datum poslednjeg pristupa: 20. mart 2024.

[51] RBF SVM parameters, https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html#sphx-glr-auto-examples-svm-plot-rbf-parameters-py, Datum poslednjeg pristupa: 20. mart 2024.

[52] Gram Matrix, https://mathworld.wolfram.com/GramMatrix.html, Datum poslednjeg pristupa: 20. mart 2024.

[53] Isometry, https://mathworld.wolfram.com/Isometry.html, Datum poslednjeg pristupa: 20. mart 2024.

[54] Semidefinite Matrices, https://www.sfu.ca/~mdevos/notes/semidef/semidef_mat.pdf, Datum poslednjeg pristupa: 20. mart 2024.

[55] 1.6. Nearest Neighbors, https://scikit-learn.org/stable/modules/neighbors.html

[56] Distance Metrics, https://numerics.mathdotnet.com/Distance, Datum poslednjeg pristupa: 20. mart 2024.

[57] sklearn.metrics.DistanceMetric, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html#sklearn.metrics.DistanceMetric, Datum poslednjeg pristupa: 20. mart 2024.

[58] Advantages and Pitfalls of Pattern Recognition, 2020., Hyperspheres, https://www.sciencedirect.com/topics/earth-and-planetary-sciences/hyperspheres, Datum poslednjeg pristupa: 20. mart 2024.

[59] Probabilistic Pocket Druggability Prediction via One-Class Learning, https://www.frontiersin.org/journals/pharmacology/articles/10.3389/fphar.2022.870479/full, Datum poslednjeg pristupa: 20. mart 2024.Omohundro M.S., 1989.,

[60] https://citeseerx.ist.psu.edu/pdf/17ac002939f8e950ffb32ec4dc8e86bdd8cb5ff1, Datum poslednjeg pristupa: 20. mart 2024.

[61] 1.2.2. Mathematical formulation of the LDA and QDA classifiers, https://scikit-learn.org/stable/modules/lda_qda.html#mathematical-formulation-of-the-lda-and-qda-classifiers, Datum poslednjeg pristupa: 20. mart 2024.

[62] Soltan Mohammadi, Mahdi Cheshmi, Kazem Gopalakrishnan Ganesh, Hall Mary, Dehnavi MM Venkat Anand, Yuki Tomofumi, Strout Michelle. 2018., Sparse Matrix Code Dependence Analysis Simplification at Compile Time. https://www.researchgate.net/figure/Compressed-Sparse-Row-CSR-sparse-matrix-format-The-val-array-stores-the-nonzeros-by_fig1_326696933, Datum poslednjeg pristupa: 20. mart 2024.

[63] Linear Transformations in Machine Learning: A Fundamental Guide, https://techntales.medium.com/linear-transformations-in-machine-learning-a-fundamental-guide Datum poslednjeg pristupa: 20. mart 2024.

[64] What is the difference between latent and embedding spaces?, https://ai.stackexchange.com/questions/11285/what-is-the-difference-between-latent-and-embedding-spaces, Datum poslednjeg pristupa: 20. mart 2024.

[65] Radečić D., 2020., Softmax Activation Function Explained, Towards Data Science https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60, Datum poslednjeg pristupa: 20. mart 2024.

[66] Comparison of Classification Algorithms (LR, DT, RF, SVM, knn), 2020., https://jouneidraza.medium.com/comparison-of-classification-algorithms-lr-dt-rf-svm-knn-6631493e300f, Datum poslednjeg pristupa: 20. mart 2024.

[67] Comparative Study on Classic Machine learning Algorithms, https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9 Datum poslednjeg pristupa: 20. mart 2024.

[68] Comparative Study on Classic Machine learning Algorithms Part-2, https://medium.com/@dannymvarghese/comparative-study-on-classic-machine-learning-algorithms-part-2-5ab58b683ec0, Datum poslednjeg pristupa: 20. mart 2024.

[69] Prateek Grewal1, Prateek Sharma2, Dr Anu Rathee3, Dr Shikha Gupta, 2022., COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS, https://eprajournals.com/IJSR/article/7089/download, Datum poslednjeg pristupa: 20. mart 2024.

[70] Comparing Popular Classification Algorithms: A Guide to Choosing the Right One for Your Project, https://behesht.medium.com/classification-determining-the-appropriate-model-support-vector-machine-or-logistic-
Datum poslednjeg pristupa: 20. mart 2024.

[71] Vraj Sheth, Urvashi Tripathi, Ankit Sharma, 2022., Data Communication Technology and Application A Comparative Analysis of Machine Learning Algorithms forClassification Purpose https://www.sciencedirect.com/science/article/pii/S1877050922021159?ref=pdf_download&fr=RR-2&rr=8668735f4d21b335, Datum poslednjeg pristupa: 20. mart 2024.

[72] The Complete Guide on Overfitting and Underfitting in Machine Learning, https://www.simplilearn.com/tutorials/machine-learning-tutorial/overfitting-and-underfitting, Datum poslednjeg pristupa: 20. mart 2024.

[73] What is Interoperability? - AWS, https://aws.amazon.com/what-is/interoperability/, Datum poslednjeg pristupa: 20. mart 2024.

[74] Collinearity Diagnostics, Model Fit & Variable Contribution, https://cran.r-project.org/web/packages/olsrr/vignettes/regression_diagnostics.html, Datum poslednjeg pristupa: 20. mart 2024.