Primjena algoritama strojnog učenja u svrhu klasifikacije tekstilnih plošnih proizvoda u programskom paketu Weka

The application of machine learning algorithms for textile fabrics classification in the Weka software

Znanstveni rad / Scientific paper

Selma Imamagić [1*], Tomislav Rolich [2], Željko Penava [3]

[1] Sveučilište u Zagrebu Tekstilno-tehnološki fakultet, Zavod za odjevnu tehnologiju, Prilaz baruna Filipovića 28a, 10 000 Zagreb, Republika Hrvatska

[2] Sveučilište u Zagrebu Tekstilno-tehnološki fakultet, Zavod za temeljne prirodne i tehničke znanosti, Prilaz baruna Filipovića 28a, 10 000 Zagreb, Republika Hrvatska

[3] Sveučilište u Zagrebu Tekstilno-tehnološki fakultet, Zavod za projektiranje i menadžment tekstila, Prilaz baruna Filipovića 28a, 10 000 Zagreb, Republika Hrvatska

*Korespondencija: selma.imamagic@ttf.unizg.hr

Sažetak

Strojno učenje predstavlja jednu od grana umjetne inteligencije koja simulira procese ljudskog učenja. Algoritmi razvijeni u ovom području nastoje 'naučiti' zaključivati i predviđati na temelju empirijskih podataka. Jedna od primjene ovih algoritama je u klasifikaciji, tj. razvrstavanju podataka u različite kategorije prema određenim karakteristikama ili svojstvima istih. Cilj ovog rada bilo je proučavanje performansi algoritama strojnog učenja namijenjenih klasifikaciji koristeći podatake iz tekstilne industrije. Testiranje se provelo na skeniranim prikazima različitih tekstilnih plošnih proizvoda (tkanina, pletiva, netkanog tekstila, čipki i tepiha) u programskom paketu Weka. Ovi proizvodi su trebali biti klasificirati u kategorije prema vrsti tekstilije kojoj pripadaju. U tu svrhu, eksperimentiralo se s brojem potrebnih kategorija (klasa), vrstom klasifikatora te filterima dostupnim u programskom paketu. Provedena su tri različita pokusa, a rad modela vrednovan je kroz parametar točnosti. Najbolji rezultati u sva tri pokusa postignuti su primjenom FCTH-filtera te IBk klasifikatora.

Ključne riječi: strojno učenje; klasifikacija; tekstilna industrija; Weka

Abstract

Machine learning is a branch of artificial intelligence (AI) that simulates human learning. The algorithms developed in this field attempt to 'learn' how to draw conclusions and make predictions based on empirical data. One of the applications of these algorithms is data classification, i.e. the categorisation of data into different categories based on certain characteristics or properties. The aim of this work was to observe the performance of machine learning algorithms for classification using data from the textile industry. The tests were performed on scanned images of different textiles (woven, knitted, non-woven, lace and carpets) in the Weka software. These textiles were to be categorised according to the type of textile fabric they belong to. For this purpose, the number of categories (classes), the type of classifier used and the filters available in Weka were experimented with. Three different tests were performed and the performance of the model was evaluated based on the classification accuracy. The best results in all three tests were obtained with the FCTH-filter and the IBk classifier.

Keywords: machine learning; classification; textile industry; Weka

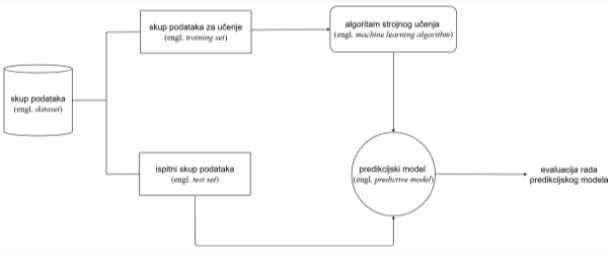
1. Uvod

Strojno učenje je tehnika izgradnje modela, odnosno obrasca ponašanja određene skupine podataka na temelju „učenja“ istih i odnosa među njima. Cilj strojnog učenja jest pronaći uzroke i poveznice među podacima te time steći uvid, a zatim i mogućnost predviđanja izlaznih podataka uz određenu predikcijsku točnost [1].

Algoritmi strojnog učenja se, zahvaljujući svojoj sposobnosti učenja iz iskustva, mogu dobro prilagođavati podacima, odnosno promjenjivoj okolini pa se tako parametri modela ponašanja nakon svakog viđenog primjera ažuriraju, a uspješnost predviđanja povećava [2]. S tim u vezi ističe se i nužno obilježje koje ovi algoritmi moraju posjedovati - generalizacijsku sposobnost, odnosno mogućnost izgradnje modela ponašanja koji će ispravno predvidjeti izlazne vrijednosti podataka koje algoritam nije vidio tijekom učenja [2].

Jedna od tehnika strojnog učenja jest nadzirano učenje (engl. Supervised Learning). Radi se o učenju temeljenom na prethodno označenom skupu podataka u kojem je svakom skupu vrijednosti ulaznih varijabli (podataka) pridružena vrijednost izlaznih varijabli (podataka) [3]. Ulazni podaci pritom su podijeljeni na dva skupa podataka. Prvi skup namijenjen je “učenju” algoritma (engl. training set). Algoritam „uči“ odnose među podacima unutar prvog skupa i stvara prikladni predikcijski model (engl. predictive model). Drugi skup naziva se ispitnim skupom (engl. test set) s obzirom da se na njemu provjerava točnost rada predikcijskog modela (engl. accuracy). Ukoliko se točnost rada stvorenog modela smatra prihvatljivom, predikcijski model može se primijeniti za previđanje na novim, nepoznatim primjerima, sl. 1.

DOI: 10.34187/ko.73.3.2



Slika 1. Prikaz rada algoritma za nadzirano učenje [4]

Dva glavna područja primjene ovog učenja su regresija (engl. regression) i klasifikacija (engl. classification). Regresija se temelji na predviđanju kontinuiranih izlaznih varijabli, tj. pridruživanju ulaznim podacima ciljane numeričke vrijednosti dok klasifikacija (razvrstavanje) na predviđanju diskretnih izlaznih varijabli, odnosno pridruživanju oznake klase (razreda) zadanim primjerima.

2. Definiranje problema klasifikacije

Svaki algoritam predstavlja niz naredbi koje se provode kako bi se ulaz pretvorio u izlaz [2]. Tako i algoritmi strojnog učenja na svom ulazu imaju skup primjera (engl. example, instance) za koje se želi da algoritam napravi predviđanje [5].

Primjeri su okarakterizirani nizom značajki (engl. atributte) od kojih se kao ulazni podaci uzimaju samo one koje se smatraju relevantnima za rješavanje određenog problema [6]. Takav primjer prikazuje se kao vektor značajki:

$$x=(x_1, x_2 \ldots x_n) \tag{2.1}$$

pri čemu su xi pojedine značajke, a n njihov ukupan broj.

Skup primjera nalazi se u ulaznom prostoru ili prostoru primjera (engl. instance space, input space). U prostoru primjera svaki primjer predstavlja jedan vektor, odnosno točku u vektorskom prostoru. Označivši vektorski prostor s X slijedi

$$x \in X \tag{2.2}$$

gdje je x primjer, a X prostor primjera [5]. Broj dimenzija ulaznog prostora X pritom odgovara broju značajki primjera x [5].

S obzirom da kod nadziranog učenja svaki primjer ima svoju unaprijed poznatu oznaku (engl. label), ukoliko se oznaku primjera označi s y, a skup svih mogućih oznaka u skupu podataka Y, slijedi kako je svaki ulazni primjer uređeni par (x, y) pri čemu je x oznaka za primjer, a y oznaka za oznaku primjera [6]. Tako se skup označenih primjera (engl. labeled dataset) koji se sastoji od N primjera može iskazati kao skup uređenih parova

$$D= \{(x^{(t)}, y^{(t)}) \}_{t=1}^{N}. \tag{2.3}$$

Pritom t predstavlja oznaku primjera x, odnosno oznaku oznake primjera y [5, 6].

Cilj nadziranog strojnog učenja jest naučiti algoritam da primjerima iz skupa X dodjeljuje oznake iz skupa Y, odnosno naučiti preslikavanje iz jednog u drugi skup. Takva funkcija naziva se hipotezom i definira se kao

$$h: X \rightarrow Y \tag{2.4}$$

gdje je h oznaka za hipotezu, X skup primjera, Y skup oznaka primjera [5].

Ukoliko je željeni problem kojeg određeni algoritam strojnog učenja treba riješiti klasifikacijski problem oznaka primjera predstavlja klasu (razred), a h funkciju koja svakom primjeru iz prostora primjera pridružuje oznaku klase. Cilj klasifikacijskog algoritma jest naučiti hipotezu h, odnosno odrediti poprima li neki primjer x oznaku klase y.

3. Klasifikatori

Prilikom rješavanja klasifikacijskih problema izgradnja predikcijskog modela tzv. klasifikatora odvija se u dva koraka (sl. 1). Prvi korak obuhvaća „učenje algoritma“ (engl. learning step) na skupu podataka za učenje (engl. training set) gdje se algoritam “uči” pridruživanju primjera iz skupa X njihovih klasama iz skupa Y [7]. Pritom se gradi spomenuti model – klasifikator [7]. Drugi korak obuhvaća klasikifaciju (engl. classficiation step) gdje se izgrađeni klasifikator koristi za klasifikaciju na ispitnom skupu podataka (engl. test set) koji nije korišten prilikom izgradnje klasifikatora [7]. Tada se na temelju dobivenih rezultata klasifikacije vrednuje ispravnost rada klasifikatora odnosno njegova generalizacijska sposobnost (engl. accuracy), iskazana kao postotak ispravno klasificiranih primjera.

U nastavku su opisani najčešći oblici pojavljivanja klasifikatora: Bayseovi klasifikatori, klasifikatori temeljeni na stablima odlučivanja, klasifikatori na temelju klasifikacijskih pravila i „lijeni“ klasifikatori.

3.1. Bayesovi klasifikatori

Bayesovi klasifikatori temelje se predviđanju vjerojatnosti da neki primjer pripada određenoj klasi primjenom Bayesovog pravila.

Pod pretpostavkama da je primjer x vektor definiran pomoću n značajki pod izrazom (2.1), a h hipoteza da primjer x pripada određenoj (specifičnoj) klasi C, želi se odrediti kolika je vjerojatnost da je hipoteza h istinita za zadani ulazni primjer x, odnosno:

$$P(h|x)= \frac{P(x|h)\ P\ (h)}{P\ (x)} \tag{2.5}$$

gdje su:

P(h|x) – aposteriorna vjerojatnost istinitosti hipoteze h, odnosno da primjer x pripada određenoj klasi C (engl. posterior probability)

P(x|h) – aposteriorna vjerojatnost da je određena klasa C povezana s primjerom x (engl. class likelihood)

P(h) – apriorna vjerojatnost hipoteze h (engl. prior probability)

P(x) – apriorna vjerojatnost primjera (engl. evidence) [6,7].

Bayesovi klasifikatori imaju široku primjenu kod klasifikacije teksta i uzoraka zbog svoje brzine i točnosti klasificiranja na velikim bazama podataka te, u odnosu na druge klasifikatore, u teoriji najmanjeg stupnja pogreške klasificiranja [7, 8]. Ipak zbog svojih pretpostavki o jednakoj važnosti značajki primjera prilikom pridavanja oznake klase primjeru te njihovoj međusobnoj neovisnosti, potrebno je naglasiti da ovi klasifikatori predstavljaju idealan slučaj klasifikacije što u praksi može dovesti do smanjenja točnosti broja klasificiranih primjera [7].

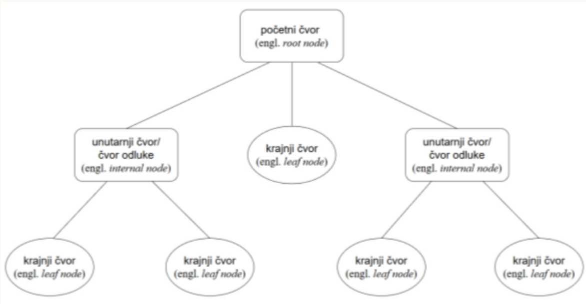
Neki od primjera Bayesovih klasifikatora su: NaiveBayes, BayesNet, AODE (engl. Averaged one-dependence estimators), NaiveBayesSimple i drugi [9].

3.2. Stabla odlučivanja

Stablo odlučivanja (engl. decision tree) je grafički prikaz strukture odlučivanja izgrađen od:

- čvorova odluke (engl. internal node) u kojima se testiraju značajke primjera
- grana (engl. branch) koje svojim vrijednostima zadovoljavaju uvjete postavljene na značajke primjera i time predstavljaju moguća rješenja zadanog uvjeta
- krajnjih čvorova (engl. leaf node) kojim određena grana stabla završava, odnosno koji predstavljaju klasu primjera [7], sl. 2.

DOI: 10.34187/ko.73.3.2



Slika 2. Shematski prikaz stabla odlučivanja

Klasifikacija stabla odlučivanja izvodi se na način da se značajke nepoznatog primjera vrijednosno uspoređuju po pojedinim čvorovima, od početnog čvora (engl. root node) pa do krajnjeg čvora koji daje klasu kojoj nepoznati primjer pripada.

Klasifikatori koji koriste stabla odlučivanja odlikuju fleksibilnost i laka interpretacija zahvaljujući grafičkom prikazu postupka odlučivanja. S druge strane, kompleksna stabla odlučivanja dovode do prenaučenosti modela na podacima za učenje (engl. overfitting), tj. neispravne klasifikacije nepoznate skupine podataka. Ovaj problem rješava se eliminacijom nepotrebnih grana stabla (engl. pruning) na dva načina: zaustavljanjem daljnjeg grananja u određenom čvoru odluke (engl. prepruning) ili naknadnim uklanjanjem grana (engl. postpruning) [7, 10].

Najpoznatiji klasifikatori koji koriste stabla odlučivanja su: J48, ID3 (engl. Induction of Decision Trees), NBTree, slučajna šuma (engl. RandomForest) i drugi [9].

3.3. Klasifikacijska pravila

Klasifikacijska pravila (engl. classification rules) popularna su alternativa stablima odlučivanja temeljena na „ako-onda“ pravilu. Kao i kod stabla odlučivanja u kojem se testiranje provodilo u čvorovima, ovdje „ako“ prethodi testiranju koje se provodi nad značajkama primjera, dok krajnji čvor u pravilima je definiran s „onda“ iza kojeg slijedi posljedica, tj. klasa kojoj određeni primjer pripada [9]. Uvjeti su međusobno povezani konjukcijom testova (logičkim operatorom „i“) dok su pojedina pravila međusobno povezana disjunkcijom (logički operator „ili“). Iz navedenog slijedi da, ukoliko bilo koji primjer odgovara određenom pravilu, on se tada svrsta u klasu koja slijedi ispunjenjem tog pravila [9].

Popularnost primjene klasifikacijskih pravila leži u činjenici da svako pravilo predstavlja zaseban „dio“ znanja koji se može primijeniti prilikom klasifikacije individualno ili kao set pravila [9].

Negativna strana klasifikacijskih pravila jest da, u usporedbi s klasifikatorima zasnovanim na stablima odlučivanja, prilikom klasifikacije primjera se temelje na jednoj klasi cijelo vrijeme, zanemarujući pritom ostale klase. S druge strane stabla odlučivanja uzimaju u obzir sve klase prilikom postavljanja uvjeta na određenu značajku primjera u čvoru odluke u svrhu veće točnosti klasifikacije [9].

Klasifikatori koji se temelje na primjeni spomenutih pravila su: ZeroR, OneR, M5Rules, Part i drugi [9].

3.4. „Lijeni“ klasifikatori

Prethodne tri skupine načina prikazivanja klasifikatora nastale su opisanom tehnikom klasifikacije u dva koraka (slika 1; poglavlje 3). Za takve modele može se reći da su „spremni“ i „željni“ klasifikacije novih, neviđenih podataka [7]. Kod „lijenih“ klasifikatora pak nema njihova učenja na skupu podataka za učenje, nego algoritam strojnog učenja pohranjuje skup podataka za učenje i razvija predikcijski model, tj. klasifikator na osnovi usporedbe ispitnog skupa podataka s pohranjenim skupom za učenje [7]. Iz tog razloga nazvani su „lijenim“ klasifikatorima jer manje posla obavljaju prilikom učenja, a više u drugom koraku, tj. klasifikaciji.

Neki od najpoznatijih „lijenih“ klasifikatora su: IB1 (engl. Nearest-neighbour classifier), KStar, LBR (engl. Lazy Bayesian Rules) te IBk (engl. K-nearest neighbour's classifier) [9].

4. Eksperimentalni dio

4.1. Ulazni podaci

Ulazni skup podataka čini skenirani prikazi pet različitih vrsta tekstilija, sl. 3:

- tkanina (engl. woven fabrics)
- pletiva (engl. knitts)
- netkanog tekstila (engl. non-woven fabrics)
- čipki (engl. lace)
- tepiha (engl. carpets).

a) b) c) d) e)

Slika 3. Primjeri skeniranih prikaza različitih vrsta tekstilija: a) tkanine (engl. woven fabric), b) pletiva (engl. knitted fabric), c) netkanog tekstila (engl. non-woven fabric), d) čipki (engl. lace) i e) tepiha (engl. carpet)

Baza ulaznog skupa podataka sastoji se od sveukupno 804 skenirana prikaza tekstilija, od čega skoro pola prikaza čine prikazi tkanina (njih 400), 130 pletiva, 41 prikaz netkanog tekstila, 214 čipki te preostalih 2,36 % baze podataka čine prikazi tepiha (njih 19).

4.2. Metodologija rada

Prikupljene skenirane prikaze bilo je potrebno prvo predobraditi kako bi bili primjereni za testiranje rada klasifikacijskog modela. Predobrada slikovnih podataka osigurava izdvajanje dovoljno informativnih značajki skeniranih prikaza kojima će se osigurati njihovo međusobno razlikovanje i pripadnost različitim klasama. Nakon predobrade podataka slijedi njihovo klasificiranje primjenom različitih vrsta klasifikatora. Klasifikacija je provedena tri puta primjenom pet različitih filtera te četiri vrste klasifikatora što sveukupno daje broj od dvadeset mjerenja po pokusu.

4.2.1. Predobrada podataka

Predobrada podataka provedena je u dva koraka. Prvo, skenirane primjere bilo je potrebno staviti u jednu mapu i prikazati kao .arff (engl. Atributte-Relation File Format) datoteku, vrstu zapisa prikladnu za rad u Weka programskom paketu [11, 12]. Svaki skenirani zapis u toj datoteci morao je biti opisan dvama značajkama: nazivom datoteke (engl. file name) i informacijom o klasama (engl. class information) [12]. Drugi korak obuhvaćao je primjenu filtera dostupnih u programskom paketu Weka (paket ImageFilters) kojima se skenirani prikazi opisuju i prikazuju preko mjerljivih značajki (engl. features). Značajka slike definira se kao dio slike od posebnog interesa te s tim u vezi pravilna detekcija značajki slike utječe na ispravnost klasifikacije [11]. Pritom, broj izdvojenih značajki skeniranih prikaza razlikuje se s obzirom na korišteni filter, sl. 4. U nastavku je opisano pet filtera iz programskog paketa Weka odabranih za predobradu skeniranih prikaza u radu.

CLF	[33,16,15,16,16,16,16,16,16,16,16,15,16,16,16,15,15,16,15,15,15,33,16,16,16,16,16,31,16,16,16,16,16,knitwear]
EHF	[1,4,4,5,7,1,4,5,6,7,2,4,5,6,6,2,5,2,4,7,1,5,5,5,6,1,6,4,6,6,2,5,4,6,5,2,5,5,5,6,1,5,5,6,6,1,5,5,3,7,2,5,4,6,6,1,5,5,3,6,2,5,5,4,6,2,5,3,5,6,1,5,5,5,6,1,5,5,5,6,knitwear]
FCTHF	[0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,knitwear]
GF	[10.420364,0.039542,10.423867,0.039555,10.423867,0.039555,10.420364,0.039542,10.423867,0.039555,10.423867,0.039555,10.768928,0.040865,10.772548,0.040878,10.772548,0.040878,10.768928,0.040865,10.772548,0.040878,10.772548,0.040878,10.758971,0.040827,10.762587,0.04084,10.762587,0.04084,10.758971,0.040827,10.762587,0.04084,10.762587,0.04084,10.730758,0.04072,10.734365,0.040733,10.734365,0.040733,10.730758,0.04072,10.734365,0.040733,10.734365,0.040733,10.650555,0.040415,10.654135,0.040429,10.654135,0.040429,10.650555,0.040415,10.654135,0.040429,10.654135,0.040429,knitwear]
SCHF	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,153,16,0,0,0,8,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,knitwear]

Slika 4. Prikaz skeniranog prikaza pletiva preko mjerljivih značajki u programskom paketu Weka

DOI: 10.34187/ko.73.3.2

ColorLayoutFilter (CLF) je filter koji izdvaja značajke rasporeda boja na slici [13]. Određivanje rasporeda boja na slici provodi se podjelom slike na 64 bloka te utvrđivanjem dominantne boje u pojedinom bloku [13]. Dominantna boja definira se na temelju prosječne boje svih piksela unutar bloka [13].

EdgeHistogramFilter (EHF) je filter koji izdvaja značajke položaja rubova, linija i diskontinuiteta slike [12]. Temelji se na određivanju položaja spomenutih elemenata slike i izradi histograma koji prikazuje distribuciju njihovih smjerova na slici (npr. okomiti, horizontalni, dijagonalni elementi, bez jasnog usmjerenja i sl.) [13].

FuzzyColorAndTextureHistogramFilter (FCTHF) izdvaja značajke kojima se opisuju boja i tekstura slike [14]. Postupak izdvajanja značajki provodi se u 3 koraka. Prvo, slika se podijeli u određeni broj blokova na koje se primjenjuje skup pravila temeljenih na neizrazitoj logici (engl. fuzzy logic) kojima se kreira histogram [15]. Svaki dio histograma pritom predstavlja učestalost pojavljivanja određene boje na slici u HSV prostoru boja (Hue, Saturation, Value - ton, zasićenost, svjetlina) [15]. U drugom koraku veličina histograma se mijenja, odnosno definiraju se nove informacije vezane za ton boje dok se u trećem koraku obrađuje i tekstura slike te stvara još veći histogram koji predstavlja svojevrsnu kombinaciju boje i teksture slike, a njegova vrijednost učestalost pojavljivanja te kombinacije na slici.

GaborFilter (GF) koristi se u svrhu izdvajanja teksturnih značajki slike [13]. Svoj rad ovi filteri temelje na grupi Gaussovih filtera koji pokrivaju frekvencijsku domenu s različitim frekvencijama i orijentacijama [16]. Teksture slike određene su na temelju odgovora na spomenute različite frekvencije u frekvencijskoj domeni, odnosno na temelju magnituda odziva pojedinih dijelova slike [16]. Prostorno, Gabor filter prikazuje se Gaussovom funkcijom moduliranom sinusoidalnom krivuljom [16].

SimpleColorHistogramFilter (SCHF) je filter kojim se izdvajaju značajke distribucije boje na skeniranom prikazu [12]. Učestalost pojavljivanja određene boje na slici temelji se na razdvajanju slike u „kanale“ boja, primjerice tri kanala – R (engl. red), G (engl. green), B (engl. blue) u RGB prostoru boja. S tim u vezi se za svaki „kanal“ boje određuje učestalost njezinog pojavljivanja na slici i prikazuje pomoću histograma (za RGB prostor boja: tri histograma distribucije boje – za crvenu, zelenu i plavu boju) [12, 13].

4.2.2. Postupak klasifikacije

Za potrebe klasifikacije skeniranih prikaza odabrana su četiri klasifikatora dostupna u programskom paketu Weka.

ZeroR je najjednostavniji klasifikator koji spada u klasifikatore temeljene na klasifikacijskim pravilima. Zanemaruje značajke skeniranog prikaza (primjera) te uzima u obzir samo oznaku klase. Radi na pravilu određivanja učestalosti pojavljivanja određene oznake klase te predviđanju klase kojoj pripada najveći broj primjera [9]. S tim u vezi svakom novom primjeru dodijelit će oznaku upravo spomenute klase kojoj pripada većina primjera iz skupa podataka za učenje. Zbog svog načina rada koristi se kao osnovni klasifikator za provjeru načina rada algoritma za strojno učenje, tj. razvijenog predikcijskog modela (klasifikatora). Ukoliko rezultati točnosti klasifikacije budu manjeg postotka u odnosu na one dobivene ZeroR klasifikatorom, razvijeni predikcijski model može se smatrati beskorisnim.

NaiveByes je probabilistički klasifikator koji pripada skupini Bayesovih klasifikatora. Temelji se na pretpostavkama o jednakosti utjecaja značajki primjera na pridruživanje oznake klase primjeru te međusobnoj neovisnosti značajki primjera [7]. Ove pretpostavke definiraju idealan slučaj klasifikacije zbog čega točnost klasifikacije u praktičnoj primjeni često može biti smanjena. Dodatno, zbog spomenute druge pretpostavke ovaj klasifikator je i dobio naziv „naivan“ Bayesov klasifikator.

Naivni Bayesov klasifikator daje jednostavan pristup klasificiranju i pokazuje dobre rezultate ukoliko su značajke međusobno neovisne jedna o drugoj. Međutim, u praksi takva „naivna“ pretpostavka često nije slučaj što dovodi do smanjenja točnosti klasifikacije ovog predikcijskog modela [9].

IBk jedan je od „lijenih“ klasifikatora koji klasificira primjere primjenom k-NN metode (engl. k-nearest-neighbor method), odnosno metode k-najbližih susjeda. Skup primjera za učenje definiran je pomoću n značajki primjera za učenje, koji grade određeni prostor primjera. Svaki primjer za učenje predstavlja jednu točku u n-dimenzionalnom prostoru. Dobivši ispitni skup primjera, k-NN klasifikator traži za svaki primjer iz ispitnog skupa primjere iz skupa za učenje koji su mu najbliži u n-dimenzionalnom prostoru. Taj broj k-primjera iz skupa za učenje čini k-najbližih susjeda nepoznatom primjeru iz ispitnog skupa. Nepoznatom primjeru k-NN klasifikator zatim pridjeljuje klasu koja je najčešća između njemu k-najbližih susjeda [7].

k-NN klasifikatori su lako prilagodljivi dodavanju novih podataka skupu za učenje s obzirom da ih pohranjuju. Zahtijevaju poznavanje k-vrijednosti, odnosno broja najbližih susjeda te izračun udaljenosti što smanjuje njihovu složenost u odnosu na druge klasifikatore koji imaju više hiperparametara. S druge strane, ovi klasifikatori svakoj značajki pridaju jednaku važnost zbog čega, u slučaju irelevantnih značajki primjera, mogu imati malu ispravnost klasifikacije [7]. Problem ispravnosti klasifikacije raste i s povećanjem broja značajki ulaznih podataka, posebice kod manjeg ulaznog skupa podataka. Skloni su prenaučenosti modela (engl. overfitting) ukoliko je broj najbližih susjeda malen, odnosno podnaučenosti (engl. underfitting) za veliku vrijednost hiperparametra k [7, 10].

J48 je klasifikator temeljen na metodi stabla odlučivanja. Radi se zapravo o C4.5 klasifikatoru koji je, prilikom njegova preuzimanja u Weka program pisan u Javi, dobio naziv „J48“ [17]. Rad modela temelji se izgradnji stabla „odozgo prema dolje“ (engl. top-down) [17].

Klasifikator na svakom čvoru odluke stabla odabire značajku primjera koju će prvo testirati. Odabir algoritma izvodi se na temelju kriterija za odabir značajke, u ovom slučaju informacijske dobiti (engl. information gain). Kriteriji za odabir značajke su heurističke metode kojima se označeni skup podataka D, definiran izrazom (2.3) najbolje dijeli na pojedine klase [7]. Iz tog razloga nazivaju se još i pravilima podijele (engl. splitting rules) jer određuju kako podijeliti skup primjera u određenom čvoru odluke [7].

Kod informacijske dobiti prvo se odabire značajka s najvećom informacijskom dobiti kojom će se izvesti podjela primjera u određenom čvoru odluke. Informacijska dobit značajke ukazuje na „dobit“ ukoliko se izvede podjela primjera po toj značajki, odnosno na smanjenje potrebne količine informacija za podjelu primjera u klase. Time se minimizira potrebna informacija za klasifikaciju primjera u klase i dovodi do najmanje pogreške klasifikacije; smanjuje potrebu za dodatnim testiranjima značajki i time u konačnici i veličine stabla odlučivanja [7].

Glavne prednosti ovog klasifikatora su: jednostavnost, lakoća razumijevanja, brzina rada te mogućnost rada s numeričkim i kategorijskim vrijednostima [17]. S druge strane, ovaj klasifikator uzima u obzir sve značajke primjera, bilo da se radi o značajkama važnim za klasifikaciju ili ne. Navedeno dovodi do nestabilnosti stabla odlučivanja, ali i prenaučenosti modela na podatke za treniranje [18]. Slična situacija događa se kod numeričkih značajki gdje, ukoliko iznos značajke iznosi nula ili blizu nuli, on svojom vrijednošću ne doprinosi pridruživanju oznake klase primjeru, nego može dovesti do nepotrebnog povećanja grana stabla i njegove kompliciranije strukture [18].

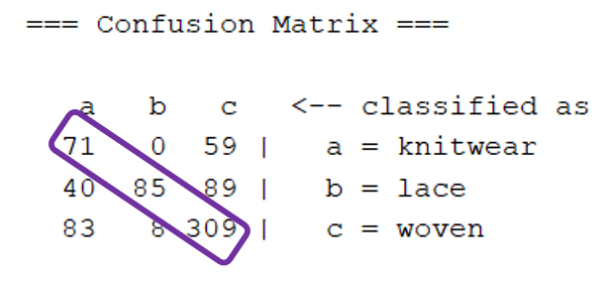
5. Rezultati i rasprava

U nastavku rada prikazani su rezultati triju pokusa klasifikacije skeniranih prikaza tekstilija u programskom paketu Weka. Rezultati su dani tablično (tablice 1-3). Dobiveni rezultati promatrani su kroz parametar točnosti klasifikatora (engl. accuracy) koji se određuje iz matrice zabune (engl. confusion matrix) pomoću sljedećeg izraza [19]:

$$\text{Acc} = \frac{\text{broj ispravno klasificiranih primjera}}{\text{broj svih klasificiranih primjera}} . \quad (5.1)$$

Ispravno klasificirani primjeri predstavljaju „dijagonalu“ u matrici zabune dok se ukupan broj svih primjera može dobiti zbrojem svih brojeva u matrici zabune, sl. 5.

DOI: 10.34187/ko.73.3.2



Slika 5. Matrica zabune u programskom paketu Weka

Na temelju danog primjera sa slike 5 točnost klasifikatora, odnosno broj ispravno klasificiranih primjera iznosio bi 62,5 %.

$$Acc = \frac{71+85+309}{71+0+59+40+85+89+83+8+309}$$

$$= \frac{465}{744} = 0,625 = 62,5\ \%$$

Točnost klasifikacije pratila se u odnosu na točnost osnovnog klasifikatora ZeroR; vrijednosti točnosti koje su bile manjeg iznosa u odnosu na točnost osnovnog klasifikatora u tablicama 1-3 označene su crvenim, dok one koje su pokazale veću vrijednost zelenom bojom.

U prvom pokusu („v1“) korišteno je 400 skeniranih prikaza tkanina te 130 skeniranih prikaza pletiva. Označeni skup podataka sastojao se dakle od 530 primjera te 2 oznake klase, tablica 1.

Tablica 1. Rezultati klasificiranja skeniranih prikaza tekstilnih plošnih proizvoda u prvom pokusu („v1“)

Filter	Broj značajki	ZeroR	NaiveBayes	IBk	J48
CLF	33	75,47	63,81	70,57	72,86
EHF	80	75,47	70,65	73,69	70,33
FCTHF	192	75,47	66,01	85,70	79,09
GF	60	75,47	62,91	72,48	74,99
SCHF	64	75,47	45,64	83,52	78,05

U drugom pokusu („v2“) uzeto je svih 5 vrsta skeniranih tekstilija, što prema ulaznoj bazi podataka (podglavlje 4.1.) znači da je označeni skup podataka sadržavao 804 primjera s 5 oznaka klasa, tablica 2.

Tablica 2. Rezultati klasificiranja skeniranih prikaza tekstilnih plošnih proizvoda u drugom pokusu („v2“)

	Broj značajki	ZeroR	NaiveBayes	IBk	J48
CLF	33	49,75	43,84	57,17	64,88
EHF	80	49,75	58,50	61,84	57,24
FCTHF	192	49,75	69,12	86,29	80,47
GF	60	49,75	47,67	67,87	73,00
SCHF	64	49,75	46,47	81,57	78,02

Treći pokus („v3“) proveden je s 3 različite skupine tekstilnih plošnih proizvoda – pletivima (130 skeniranih prikaza), čipkama (214 skeniranih prikaza) te netkanim tekstilijama (400 skeniranih prikaza). Označeni skup podataka u ovom slučaju sačinjavalo je 744 primjera kojima je potrebno pridružiti 3 oznake klase, tablica 3.

Tablica 3. Rezultati klasificiranja skeniranih prikaza tekstilnih plošnih proizvoda u trećem pokusu („v3“)

	Broj značajki	ZeroR	NaiveBayes	IBk	J48
CLF	33	53,77	66,61	67,00	72,47
EHF	80	53,77	59,20	62,87	59,99
FCTHF	192	53,77	71,24	88,96	83,08
GF	60	53,77	70,44	77,22	79,49
SCHF	64	53,77	53,76	87,38	82,25

Iz tablica 1-3 slijedi kako svi skupovi podataka („v1“, „v2“ i „v3“) imaju između 33 i 192 značajki, odnosno isti prosječan broj značajki od 85,8. Ovo ukazuje da se složenost značajki nije mijenjala u pojedinim pokusima. Također, uspoređujući rezultate dobivene trima pokusima, u sva tri pokusa najslabiji rezultati klasifikacije postignuti su primjenom filtera ColorLayoutFilter (CLF) temeljenom isključivo na značajkama rasporeda boja skeniranih prikaza te EdgeHistogramFilter (EHF) koji izdvaja značajke položaja rubova, linija i diskontinuiteta slike. Slabi rezultati klasifikacije nakon predobrade slika CL-filterom povezan je sa sličnosti obojenja skeniranih tekstilija različite vrste gdje sama boja slike ne čini dominantnu značajku pri klasifikaciji ovih skeniranih prikaza. Također, kod skeniranih prikaza tekstilnih plošnih proizvoda nema izražajnih rubova, linija ili diskontinuiteta na prikazima kao i njihova jasna usmjerenja (primjerice dijagonalne, okomite ili vertikalna linije) po kojima bi se određena vrsta tekstilije jasno razlikovala od drugih što je dovelo do slabih rezultata i EH-filtera.

S druge strane, najbolji rezultati dobiveni su primjerom FCTH-filtera. Ovaj filter iz skeniranih prikaza uz spomenutu značajku rasporeda boje slike izdvaja i teksturu skeniranih prikaza koja čini važnu ulogu u raspoznavanju tekstilnih plošnih proizvoda (primjerice reljefne nejednolikosti tekstilne površine u vidu vlasaste površine, površine nemirnog izgleda, hrapave, glatke površine, površine dobivene kombinacijom raznih vezova ili prepleta i druge). Kombinacijom ovih dviju značajki slike postignuti su najbolji rezultati klasifikacije različitih vrsta tekstilija u svim pokusima. Dodatno, s obzirom da se izdvajaju dvije vrste značajki skeniranih prikaza, ovim filterom dobiven je i najveći broj značajki primjera što je pridonijelo ispravnosti klasifikacije.

Uz odabir filtera, na ispravnost klasifikacije utječe i odabir klasifikatora. Tako je u prvom pokusu primjena osnovnog klasifikatora ZeroR dovela do 75,47 % ispravno klasificiranih primjera. Ovako relativno visok stupanj ispravne klasifikacije za jednostavan klasifikator koji zanemaruje značajke primjera pri svom radu može se zahvaliti nejednolikosti broja označenih primjera, odnosno primjera po klasama. Naime ZeroR predviđa klasu kojoj pripada najveći broj primjera i sve nove primjere dodaje upravo toj klasi koja „dominira“ tako da je u ovom pokusu, gdje dominira broj prikaza tkanina (400 skeniranih prikaza), ZeroR postigao relativno velik rezultati uspješnosti klasifikacije. U druga dva pokusa točnost ovog klasifikatora opada s obzirom na izbalansiraniju distribuciju pripadanja primjera po klasama, odnosno ne postoji jasno odstupanje u broju pripadanja primjera jednoj klasi kao što je to slučaj u prvom pokusu s tkaninama.

Klasifikator NaiveBayes u sva tri pokusa daje najslabije rezultate (u usporedbi s klasifikatorima IBk i J48). U prvom pokusu, njegova točnost manja je i od osnovnog klasifikatora ZeroR. Ovako nizak rezultat ispravno klasificiranih primjera može se povezati s njegovom „naivnom“ pretpostavkom o međusobnoj neovisnosti značajki (podpoglavlje 4.2.2). Iako u teoriji ovaj klasifikator daje mali stupanj pogreške, u praksi to često nije slučaj. Takav primjer moguće je uočiti i ovdje, gdje točnost ovog klasifikatora u prvom pokusu varira od 45,64% do 70,65%, u drugom od 43,84% do 69,12% dok u trećem pokusu od 53,76% do 71,24%.

IBk klasifikator daje najbolje rezultate klasifikacije u sva tri pokusa. U prvom pokusu postotak ispravno klasificiranih primjera kreće se od 70,57% do 85,70%, u drugom od 57,17% do 86,29% dok se u trećem pokusu kreće od 62,87% do 88,96%. S obzirom da se ovaj klasifikator

DOI: 10.34187/ko.73.3.2

temelji na dodjeljivanju klase primjeru na temelju klase koja se najčešće pojavljuje između njegovih k-najbližih susjeda, dobiveni rezultati ukazuju da prostorna blizina pruža dobru osnovu za klasifikaciju unutar ovih skupova. Dodatno, najbolji rezultat klasifikacije ovog klasifikatora može se objasniti i njihovom manjom složenošću u odnosu na druge klasifikatore, odnosno potrebom za poznavanjem manjeg broja hiperparametara algoritma (poznavanje k-vrijednosti te izračun euklidske udaljenosti).

Četvrtim klasifikatorom J48 postignute su sljedeće točnosti:

- od 70,33% do 79,09% (prvi pokus)
- između 57,24% i 80,47% (drugi pokus)
- u rasponu od 59,99% i 83,08% (treći pokus).

Većinom dobri rezultati klasificiranja uspoređujući s osnovnim klasifikatorom ZeroR koji su postignuti ovim klasifikatorom mogu se objasniti metodom rada ovog klasifikatora. Prilikom izgradnje stabla „odozgo prema dolje“, ovaj klasifikator uzima u obzir sve značajke primjera te odabire jednu značajku po kojoj će izvesti podjelu označenog skupa podataka u klase na temelju njezine najveće informacijske dobiti. On ne zanemaruje klase niti značajke te odnose među njima što je, proučavajući rezultate dobivene klasifikatorima ZeroR i NaiveBayes koji u sebi imaju neke od ovih pretpostavki, dalo puno veći postotak uspješno klasificiranih primjera.

6. Zaključak

Strojno učenje je postupak „učenja“ algoritama informacijama i njihovim odnosa iz skupa podataka za učenje. Svrha postupka “učenja” jest stvaranje predikcijskog modela kojim će algoritmi strojnog učenja moći naučene odnose između informacija primijeniti na nove, neviđene skupove podataka.

Ukoliko se spomenuti skup podataka za učenje algoritma sastoji od poznatog ulaznog i izlaznog skupa podataka, takvo učenje naziva se nadziranim učenjem. Dodatno, ukoliko su izlazi klase u koje se trebaju svrstati ulazni podaci, takav problem kojeg algoritam treba naučiti i moći riješiti naziva se problemom klasifikacije.

Primjer jednog takvog problema prikazan je u eksperimentalnom dijelu ovoga rada gdje su korišteni skenirani prikazi pet vrsta tekstilnih plošnih proizvoda – tkanina, pletiva, netkanih tekstilija, čipki i tepiha – koji su trebali biti klasificirani u klase upravo sukladno vrsti tekstilije kojoj pripadaju.

Ovim radom ukazalo se kako na problem klasifikacije utječu karakteristike ulaznog skupa podataka (veličina, jednolikost primjera po klasama, broj klasa) primijenjeni filter i značajke koje isti izdvaja kao i njihov broj te vrsta klasifikatora. Za postizanje što većeg postotka točnosti klasifikatora, odnosno broja ispravno klasificiranih primjera, potrebno je dobro poznavanje vrste ulaznog skupa podataka i njihovih značajki – koje značajke donose veću razliku u njihovom razlikovanju i time doprinose bržoj, lakšoj i ispravnijoj klasifikaciji te postoji li njihova međuovisnost. Imajući ta saznanja potrebno je samo odabrati filter, a kasnije i klasifikator iz programskom paketa koji se za danu situaciju smatraju najprikladnijima.

U slučaju klasificiranja skeniranih tekstilnih plošnih proizvoda, najbolji rezultati postignuti su primjeno FCTH-filtera te IBk klasifikatora u sva tri pokusa. FuzzyColorAndTextureHistogramFilter (FCTHF) izdvaja dvije vrste značajki skeniranih prikaza (boju i teksturu) čime je dobiven najveći broj značajki primjera što je pridonijelo boljem raspoznavanju tekstilnih plošnih proizvoda i time većoj ispravnosti klasifikacije. IBk klasifikator temelji se na k-NN metodi. Ovom metodom se ulaznom primjeru pridjeljuja klasa koja je najčešća između njemu k-najbližih susjeda. S tim u vezi, prostorna blizina primjera pružila je dobru osnovu za klasifikaciju unutar ovih skupova podataka.

Literatura

[1] Bolf N.: Osvježimo znanje, Kemija u Industriji 70 (2021.) 9-10, 591-593

[2] Alpaydin E.: Strojno učenje, MATE d.o.o. 2021, 1-28; 29-54

[3] Juričić V.: Strojno učenje u uvjetima manje raspoloživosti podataka, Politehnika: Časopis za tehnički odgoj i obrazovanje 7 (2023.) 2, 26-32

[4] Karasu Asnaz S. M. et al.: The role of machine learning on metal hydride for hydrogen storage, Book of Abstracts: TUBA World Conference on Energy Science and Technology, Turkish Academy of Sciences, Ankara 2021, 319-320

[5] Šnajder J.: Osnovni koncepti – Strojno učenje 1, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2021./2022., 1-7

[6] Alpaydin E.: Introduction to Machine Learning 3rd edition, The MIT Press 2014, 21-27

[7] Han J., Kamber M. & Pei, J.: Data Mining: concepts and techniques 3rd edition, Morgan Kaufmann Publishers 2012, 327-373; 422-425

[8] Pak I. & Lee Teh P.: Machine Learning Classifiers: Evaluation of the Performance in Online Reviews, Indian Journal of Science and Technology 9 (2016.) 45, 1-9

[9] H Witten I. & Frank E.: Data Mining: Practical Machine Learning Tools and Techniques 2nd edition, Morgan Kaufmann Publishers 2005, 65-69, 403-414

[10] https://www.ibm.com/topics/knn, Pristupljeno: 2024-04-12

[11] Arganda-Carreras I. et al.: Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification, Bioinformatics 33 (2017.) 15, 2424-2426

[12] Abidin D.: The Effect of Derived Features on Art Genre Classification with Machine Learning, Sakarya University Journal of Science 25 (2021.) 6, 1275-1286

[13] Weka 3.8.6 program

[14] Musleh D. et al.: Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT, Journal of Sensor and Actuator Networks 12 (2023.) 29, 1-19

[15] Chatzichristofis A. S. & Boutalis S. Y.: FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. Proceedings: Ninth International Workshop on Image Analysis for Multimedia Interactive Services, Kawada S. (ur.), Institute of Electrical and Electronics Engineers (IEEE), Klagenfurt 2008., 191-196

[16] Recio A. J., Ruiz Fernández A. L. & Fernández-Sarriá A.: Use of Gabor filters for texture classification of digital images, Física de la Tierra (2005.), 17, 47-59

[17] Cvijetić B. & Radivojević Z.: Primjena Weka softvera i konvolucijskih neuronskih mreža u procesu klasifikovanja digitalnih fotografija, Proceedings: 18th International Symposium INFOTEH-JAHORINA, Sarajevo, Institute of Electrical and Electronics Engineers (IEEE), Sarajevo 2019., 224-229

[18] Saravanan N. & Gayathri V.: Performance and Classification Evaluation of J48 Algorithm and Kendall's Based J48 Algorithm (KNJ48), International Journal of Computational Intelligence and Informatics 7 (2018.) 4, 188-198

[19] Alhaidari F. et al.: ZeVigilante: Detecting Zero-Day Malware Using MachineLearning and Sandboxing Analysis Techniques, Computational Intelligence and Neuroscience, (2022.) 1, 1-15

DOI: 10.34187/ko.73.3.2