Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Aron Birsa

Iskanje in razvrščanje spletnih trgovin

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: izr. prof. dr. Marko Robnik Šikonja

Ljubljana, 2017

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Za preiskovanje spleta je marsikdaj smiselno definirati specializirane iskalnike, ki za ozko področje ponudijo bolj kakovostno informacijo. Tak primer so spletne trgovine, saj jih obstaja ogromno, prodajajo pa zelo različne vrste izdelkov. Ponudniki novih proizvodov težko najdejo vse trgovine, ki bi bile primerne za prodajo njihovega produkta. Sestavite prototip specializiranega spletnega iskalnika, ki bo najprej uporabil spletnega pajka za iskanje trgovin, nato pa bo najdene spletne trgovine klasificiral v nekaj vnaprej definiranih kategorij. K problemu pristopite z analizo besedil in strojnim učenjem. Razvito rešitev ovrednotite s pomočjo že razvrščenih trgovin v spletnih imenikih.

Zahvaljujem se mentorju izr. prof. dr. Marku Robniku Šikonji za strokovno pomoč in za vse nasvete pri izdelavi diplomske naloge.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
SVM	support vector machine	metoda podpornih vektorjev
HTTP	hypertext transfer protocol	protokol za izmenjavo hiperteksta
TLD	top level domain	vrhnji imenski strežnik
HTML	hypertext markup language	označevalni jezik za oblikovanje večpredstavnostnih dokumentov
TF-IDF	term frequency–inverse document frequency	uteževanje s frekvenco besed in inverzno dokumentno frekvenco
kNN	k-nearest neighbors	k-najbližjih sosedov
BFS	breath first search	iskanje v širino
IP	internet protocol	internetni protocol
IPv4	internet protocol version 4	internetni protokol verzije 4
IPv6	internet protocol version 6	internetni protokol verzije 6

Povzetek

Naslov: Iskanje in razvrščanje spletnih trgovin

Cilj diplomske naloge je razvoj orodja, ki omogoča avtomatsko zaznavanje spletnih trgovin glede na tip izdelkov, ki jih ponuja. Spletne strani smo klasificirali v sedem vnaprej določenih kategorij: starine in zbirke, oblačila, zabavna elektronika, pohištvo, dom in vrt, nakit in pisarniški izdelke. Glavni problem je bil pridobivanje ustreznih podatkov za izgradnjo učne in testne množice ter klasificiranje spletnih strani. Uporabili smo naslednje metode strojnega učenja: naivni Bayesov klasifikator, k-najbližjih sosedov, metodo naključnih gozdov, nevronsko mrežo in metodo podpornih vektorjev. Najbolj obetavne rezulate smo dobili z metodo podpornih vektorjev.

Ključne besede: specializirani iskalnik, podatkovno rudarjenje, strojno učenje, spletne trgovine, analiza besedil, naivni Bayesov klasifikator, k-najbližjih sosedov, metoda naključnih gozdov, nevronska mreža, metoda podpornih vektorjev.

Abstract

Title: Search and classification of web shops

The aim of the thesis was to develop a tool for automatic classification of online stores depending on the type of products they offer. Websites are classified into seven predefined categories: antiques and collectibles, clothing, consumer electronics, furniture, home and garden, jewelry and office products. The main problem was getting relevant data to build a learning and test data set and classifying web sites. The following machine learning methods were used: naive Bayesian classifier, k-nearest neighbors algorithm, random forests, neural networks and support vector machine. The most promising result were obtained using the support vector machine classifier.

Keywords: specialized search engine, data mining, machine learning, e-commerce, text analysis, naive Bayesian classifier, k-nearest neighbors algorithm, random forests, neural networks, support vector machine.

Poglavje 1

Uvod

Zaradi obsežnosti podatkov, ki jih ponuja svetovni splet, postaja iskanje specifičnih spletnih vsebin čedalje zahtevnejše. S pomočjo splošnih iskalnikov, kot so npr. Google, Bing in Yahoo, je v kratkem času praktično nemogoče priti do ciljnega nabora informacij in ustreznih spletnih mest, ki služijo pri opravljanju neke specifične dejavnosti, kot je spletna prodaja ali spletno nakupovanje. Zato smo se v diplomski nalogi ukvarjali z razvojem specializiranega iskalnika, ki omogoča pridobivanje želenih vsebin za določeno področje, dejavnost ali izdelek. Pri tem smo se ukvarjali z rešitvijo klasifikacijskega problema, s katerim smo prišli do želenih ciljnih informacij in razvili orodje za učinkovito iskanje spletnih trgovin.

Večina uporabnikov interneta se ne zaveda, da v ozadju poteka neprestano zbiranje informacij s pomočjo spletnih pajkov. Prvotno so bili namenjeni preiskovanju spleta za iskalnike, ki potrebujejo čim več informacij, da lahko nudijo točne rezultate. Nadaljni razvoj spletnih pajkov je šel v smeri vse večje specializacije in danes se uporabljajo tudi oz. predvsem za pridobivanje specifičnih informacij. Ker s splošnimi iskalniki ne dosegamo dovolj natančnih iskalnih rezultatov, menimo, da bo v prihodnosti še večji poudarek na razvoju specializiranih programov za specifično raziskovanje in indeksiranje svetovnega spleta.

Struktura diplomskega dela je naslednja. Poglavje 2 opisuje pristop k iskanju in razvrščanju spletnih trgovin, opis metod pridobivanja podatkov in učinkovitega iskanja spletnih trgovin. Poglavje 3 obravnava podrobnosti strojnega učenja, strukturo rešitve in uporabljene algoritme. Poglavje 4 prikazuje rezultate strojnega učenja, ki smo jih predstavili s pomočjo tabel in vizualizacij. Poglavje 5 predstavi sklepne ugotovitve diplomskega dela, ter opisuje možne izboljšave pridobivanja podatkov, spletnega pajka in izboljšave klasifikatorja.

Poglavje 2

Kategorije spletnih strani

Svetovni splet postaja glavni vir dostopa do podatkov in informacij, na osnovi katerih izvajamo naše vsakdanje aktivnosti. Po eni strani nam nudi obsežen nabor vsebin, ki so razmeroma lahko dostopne, po drugi strani pa se je skupaj z relevantnimi vsebinami povečala tudi redundantnost podatkov, tako da za iskanje informacije porabimo več časa, kot bi ga z uporabo prilagojenih spletnih iskalnikov.

Vse večjemu pretoku informacij po spletu sledi tudi vse večji pretok blaga in storitev. Spletno nakupovanje se poleg udobnosti, hitrosti in anonimnosti od klasičnega razlikuje po bistveno večji izbiri izdelkov. Z večanjem izbire izdelkov je za kupca ključno, da v čim krajšem času pride do izdelka, ki ga išče. Splošni internetni brskalniki, kot so Firefox, Internet Explorer in Chrome, ter splošni internetni iskalniki, kot je npr. Google, so za povprečnega uporabnika sicer uporabniško prijazni, vendar pri podajanju rezultatov velikokrat zgrešijo cilj, bodisi ker ponujajo presplošne zadetke bodisi ker pri iskanju ne upoštevajo vseh uporabnikovih kriterijev.

Navedimo primer, ko uporabnik ne ve natančno, kako se izdelek, ki ga išče, imenuje oz. kakšen opis ima na spletu. Polega tega mogoče ni seznanjen z novostmi na danem področju, torej z novimi spletnimi mesti in izdelki, ki ga zanimajo. Kot zadetke splošnih iskalnikov dobi veliko število spletnih strani, ki morda niti niso spletne trgovine ali pa ne ponujajo iskanega izdelka.

Iskanje večje količine specifičnih podatkov s splošnimi internetnimi iskalniki je torej lahko časovno potratno in neučinkovito, zato smo se v diplomski nalogi osredotočili na razvoj primerne rešitve iskanja spletnih trgovin.

Uporbniku smo želeli ponuditi programsko rešitev, ki bi mu poenostavila iskalno oz. nakupovalno izkušnjo. Ker na podobno rešitev še nismo naleteli, smo rešitev zastavili sami. Odločiti smo se morali, kakšne parametre iskanja bomo uporabili in po katerih kriterij bomo razvrščali trgovine. Po pridobitvi podatkov smo ugotovili, da so za razvrščanje spletnih trgovin najprimernejše metode strojnega učenja. Le-te omogočajo, da se računalnik s pomočjo ustreznega algoritma uči in prilagaja novim podatkom. Program ne predvidi vseh prihodnih situacij, vendar omogoča sprotno učenje in prepoznavanje novih vzorcev na podalgi predhodno naučenega. V našem primeru smo sestavili program, ki smo ga na učni množici podatkov naučili ustreznega razvrščanja trgovin glede na izbrane ciljne kategorije. Po učenju je program sam razvrščal nove primere.

2.1 Pridobivanje podatkov

V splošnih iskalnikih je težko zožiti izbor podatkov, izločiti odvečne in nerelevantne podatke pa tudi pridobljeni podatki večkrat niso popolni, pri čemer zlahka zgrešimo cilj iskanja.

Ponudniki novih proizvodov tako na primer težko najdejo spletna mesta, ki bi bila primerna za prodajo njihovih izdelkov. Pri iskanju ustreznih spletnih trgovin porabijo bodisi preveč časa bodisi je zaradi različne stopnje relevantnosti podatkov glede na vpisano ključno besedo težko izostriti poizvedbo. Zato smo sestavili prototip spletnega iskalnika, ki spletne trgovine klasificira v nekaj vnaprej določenih kategorij. Zanimalo nas je nekaj ciljnih kategorij razredov:

- starine in zbirke (*antiques and collectibles*),
- oblačila (*clothing*),

- zabavna elektronika (*consumer electronics*),
- pohištvo (*furniture*),
- dom in vrt (*home and garden*),
- nakit (*jewelry*) in
- pisarniški izdelki (*office products*).

Najprej smo raziskali možnosti učinkovitega iskanja spletnih trgovin in izdelkov. Želeli smo izboljšati izkušnjo spletnega iskanja tako, da bi naš prototip iskalnika ponudil le spletne trgovine s točno določeno kategorijo izdelkov. Tak iskalnik bi skrajšal čas iskanja, omogočil bi primerjavo spletnih trgovin z istimi kategorijami izdelkov, ponujal pregled nad spletnimi trgovinami in iz nabora zadetkov samodejno izločil napačne spletne trgovine ali izdelke.

Spletnega pajka in analitično orodje za kategoriziranje spletne strani smo zgradili s pomočjo programskega jezika *python* in knjižnic *scikit-learn*, *numpy*, *requests*, *tornado* in *html2text*. *Python* je prost, odprtokoden in objekten programski jezik, ki lahko predstavlja podlago za različne programerske projekte. Danes se uporablja na številnih platformah npr. Linux, macOS in Windows. Uporablja se pri strojnem učenju, podatkovnem rudarjenju, pisanju spletnih pajkov, izgradnji spletnih strani itd.

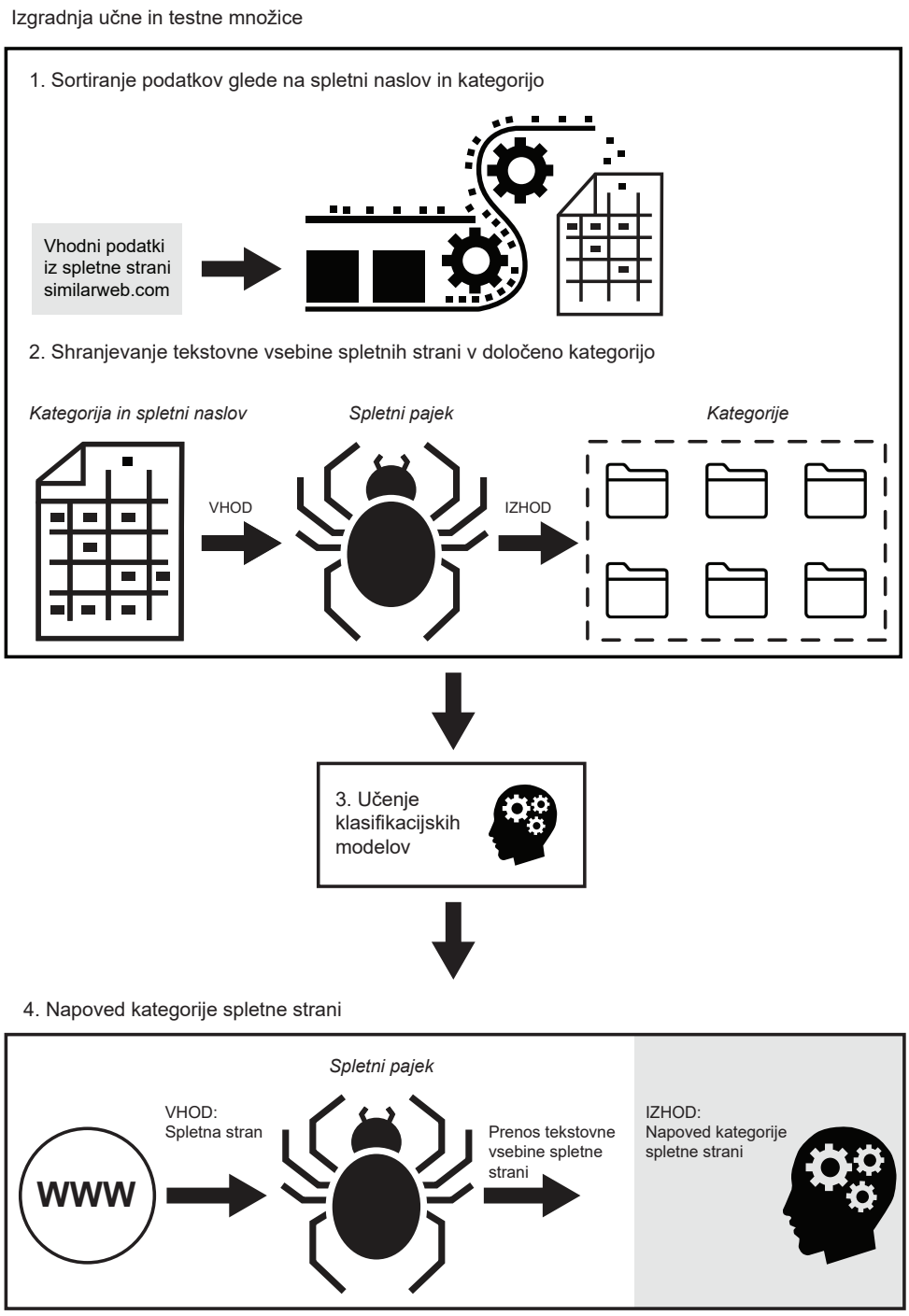
Poglavje 3

Strojno učenje

Problem, s katerim se ukvarjamo v diplomski nalogi, zajema področje strojnega učenja. V osnovi je za strojno učenje potrebno pripraviti učno množico. Na ta način pridobivamo znanje na podlagi predhodnih primerov, s čimer lahko napovemo najverjetnejši izid v novih situacijah.

Strojno učenje je "vsaka sprememba sistema, ki mu omogoča, da opravlja isto nalogo bolje. Rezultat učenja je znanje, ki ga sistem uporabi za reševanje novih nalog [1]". Sistem za strojno učenje vsebuje učni in izvajalni algoritem ter model oz. hipotezo. Za slednjo je pomembno, da čim bolj ustreza vhodnim podatkom. Zato strojno učenje opredelimo kot *modeliranje* podatkov po kriterijih optimalnosti in učinkovitosti [1]. Strojno učenje se uporablja denimo v zdravstvu, meteorologiji ali igralništvu. Rezultati strojnega učenja so funkcije, pravila ali relacije, ki jih predstavimo z različnimi podatkovnimi shemami ali modeli.

Metode in algoritme strojnega učenja izberemo glede na podatke, ki jih modeliramo. V našem primeru smo analizirali besedilo spletnih trgovin iz sedmih kategorij, v katere smo želeli klasificirati spletne trgovine na podlagi izdelkov, ki jih ponujajo. Klasifikacija ali uvrščanje je poleg segmentacije, vizualizacije ter ocenjevanja, ena najpogosteje uporabljenih metod. Uporabili smo več modelov, ki jih opisujemo v nadaljevanju. Za lažjo predstavo sistema smo na sliki 3.1 prikazali osnovne korake.



Slika 3.1: Shema arhitekture sistema.

3.1 Zbiranje podatkov

Ključno za izvedbo metod strojnega učenja so dobro pripravljeni podatki s pravilno kategorizacijo. Za izvedbo spletnega iskalnika smo najprej pridobili veliko število spletnih strani, ki smo jih kasneje uporabili za strojno učenje. Najprej smo želeli imeti pregled nad ponudbo svetovnega spleta. Za izbor spletnih trgovin bi bilo potrebno pregledati praktično celoten splet in iz njega izločiti tiste spletne strani, ki ustrezajo kriterijem. Izkazalo se je, da obstaja spletna stran (`https://www.similarweb.com/`), ki ponuja dovolj podatkov o tipih obstoječih spletnih trgovin.

Izbrali smo 1000 naslovov spletnih trgovin skupaj z njihovo kategorizacijo na sedem razredov. Potrebno je bilo izločiti nerelevantne in napačne podatke. Strani smo ročno pregledali in izločili nezadovoljive zadetke. Ostalo je 883 spletnih mest. Nad ročno pridobljenem naboru trgovin smo zgradili učni model in ga testirali s prečnim preverjanjem.

Spletne trgovine smo uvrstili glede na tip izdelkov v sedem razredov: starine in zbirke, oblačila, zabavna elektronika, pohištvo, dom in vrt, nakit, pisarniški izdelki.

Prečiščen nabor podatkov je vseboval 883 naslovov spletnih trgovin. Vsakemu naslovu je pripadala ustrezna kategorija spletne trgovine, npr. www.spletnatrgovina.si - nakit.

Naslednji korak je bil razvoj spletnega pajka, ki je na podlagi obiskov spletnih trgovin pridobil njihov opis v obliki besedila. Besedilo vsake spletne strani je bilo shranjeno v samostojno tekstovno datoteko z ustrezno kategorijo.

Za izvedbo opisane naloge smo izvedli naslednji postopek:

1. Postavitev strukture map, ki bodo služile za uvrščanje pridobljenih podatkov.
2. Implementacija uporabniškega posrednika (*user agent*), ki je simuliral obiskovalca spletne strani in omogoča pridobitev podatkov tudi z morebiti zaščitenih spletnih strani (*anti-bot protection*).
3. Preverba dosegljivosti posameznega spletnega mesta je potrebna zaradi morebitnih sprememb statusa spletnih strani. Na ta način smo izločili spletne strani, ki niso bile dosegljive.
4. V izogib morebitnim blokadam in jezikovnim omejitvam smo uporablili posredniški strežnik, ki je spletnega pajka prikazoval kot uporabnika iz tujine.

Za izvedbo naštetih nalog smo uporabili nabor *python* knjižnic, ki se uporabljajo pri tekstovnem rudarjenju in analizi pridobljenih podatkov (*scikit-learn*), izvajanju *HTTP* poizvedb na spletnih strežnikih (*requests*), branju zelo velikih datotek (*fileinput*), pridobivanju vrhnjih internetnih domen (*tld*), enostavnemu kodiranju in dekodiranju zahtev (*simplejson*), neblokiranih omrežnih operacij (*tornado*) in ugotavljanju tehnologij, ki jih uporablja določena spletna stran (*wappalyzer*).

Za luščenje besedila smo uporabili knjižnico, ki je iz spletne strani odstranila *HTML* elemente (*html2text*). Za učenje in testiranje smo uporabili stratificirano prečno preverjanje. Za strojno učenje smo besedilo vektorizirali z metodo vreče besed (*bag of words*) [1], ter izločili nerelevantne in nepomembne besede, kot so npr. vezniki.

Uporabili smo naslednje klasifikatorje:

1. Naivni Bayesov klasifikator (*Naive Bayesian classifier*) [1] ocenjuje verjetnosti iz učne množice. Ocenjujemo verjetnost za vsak razred pri danem opisu novega primera. Zvezne atribute atribute je potrebno diskretizirati z uporabo mehke diskretizacije. Slabost algoritma je, da pri močnih odvistnostih med atributi odpove, v tem primeru konstruiramo nove, bolj informativne atribute. Lepa lastnost algoritma je inkrementalno učenje: Če dobimo nove učne primere, lahko le dopolnimo znanje, ki ga že imamo. Saj je potrebno samo popraviti frekvence [1].

2. Metoda naključnih gozdov (*Random forests*) [1] temelji na ideji generiranja množice odločitvenih dreves, pri čemer pri vsakokratni izbiri najboljšega atributa v vozlišču drevesa naključno izbere majhno število atributov, izmed katerih izberemo najboljšega. Na strukturo drevesa vplivajo tudi učni primeri, ki so za vsako drevo vzorčeni z vračanjem. Dobra lastnost metode je, da zmanjšuje varianco posameznih dreves, slabost pa je otežena razlaga odločitev [1].

3. Pri nevronskih mrežah [1] je učenje pogojeno z napako, glede na katero se uravnavajo uteži. *Perceptron* je najbolj razširjen tip nevronov. Uporablja se za nadzorovano učenje binarnih klasifikacijskih problemov. Vhod je predstavljen kot vektor številk, vse povezave med nevroni pa so pri *Perceptronu* usmerjene naprej. Vhodne in izhodne vrednosti so lahko poljubne zvezne spremenljivke. Učenje poteka postopoma in traja dokler ni napaka dovolj majhna [9]. Dobra lastnost nevronskih mrež je, da podpira večsmerno izvajanje, isti podatek je lahko vhod ali

izhod, slabost pa razlaga rešitve. Pri nevronskih mrežah se uporablja princip gradientnega učenja, to pomeni, da nevronski mreži spreminjamo uteži, odvod pa nam pove, kam se moramo premaknit, da bo napaka manjša [1].

4. Metoda podpornih vektorjev SVM (*Support Vector Machine*) [1] se uspešno uporablja pri večdimenzionalnih podatkih in velja za enega najboljših klasifikatorjev tekstovnih podatkov. Je dvorazredni klasifikator (za pozitivne in negativne primere). Zaradi uporabe le dveh razredov zahteva dodatne strategije pri večrazrednih problemih npr. eden proti enemu [3].

5. K-najbližjih sosedov (*K-nearest neighbors*) [1] za dani nov primer najde najbližje oz. najbolj podobne primere in oceni verjetnostno porazdelitev razreda iz porazdelitve razredov teh primerov. Nov primer klasificiramo v večinski razred, ki ga določa k najbližjih primerov. Algoritem je občutljiv na izbrano metriko razdalje med učnimi primeri. Slabost metode je njena relativno počasna klasifikacija [1][3]. Z zniževanjem vrednosti parametra k povečujemo vpliv šuma v učni množici. Parameter k ponavadi nastavimo na liho število, s tem se izognemo morebitnemu neodločenmu rezultatu (npr. pri dveh razredih). Vrednost k=1 uporabimo takrat, ko v podatkih ni napak, če imamo šum v podatkih, povečujemo vrednost k. Optimalen parameter k večinoma iščemo na validacijski množici. Pomembno pri algoritmu je, kako definiramo atributni prostor, da ga ne deformiramo z nepotrebnimi atributi.

Za oceno točnosti klasifikacije smo izvedli desetkratno stratificirano prečno preverjanje na naboru podatkov, kot je prikazano na sliki 3.2. Učna množica je bila razdeljena na deset enakih delov, za vsak del je bila zgrajena hipoteza iz devetih delov, testirali smo jo na preostalem delu. Končna ocena točnosti predstavlja povprečje dobljenih klasifikacijskih točnosti. Za vsakega od zgoraj navedenih algoritmov smo uporabili stratificirano prečno preverjanje.

Slika 3.2: Potek 10-kratnega prečnega preverjanja.

Poglavje 4

Rezultati

Tabela 4.1 prikazuje povprečno klasifikacijsko točnost posameznih algoritmov, pridobljeno z desetkrtanim stratificiranim prečnim preverjanjem.

	Povprečje z standardno deviacijo
Metoda podpronih vektorjev	0.874 (0.014)
Multinomialni naivni Bayes	0.854 (0.012)
Nevronska mreža	0.847 (0.015)
Metoda naključnih gozdov	0.829 (0.015)
K-najbližjih sosedov	0.824 (0.013)
Binarni naivni Bayes	0.757 (0.017)

Tabela 4.1: Povprečje klasifikacijske točnosti z standardno deviacijo za desetkratno stratificirano prečno preverjanje pri napovedovanju vrste trgovine.

1. Metoda podpornih vektorjev je v našem primeru najuspešnejša, pri čemer je bila klasifikacijska točnost 87%. Uporabljene so bile privzete nastavitve algoritma iz knjižnice *scikit-learn*, razen parameter *random_state* (z katerim nastavljamo generator naključnih števil izbiranja podatkov) smo nastavili na 42.

2. Klasifikacijska točnost multinomialnega naivnega Bayesovega klasifika-

torja je bila ocenjena na 85%. Od nastavljivih parametrov smo spremenili dovzetnost do učenja α na vrednost 0.01.

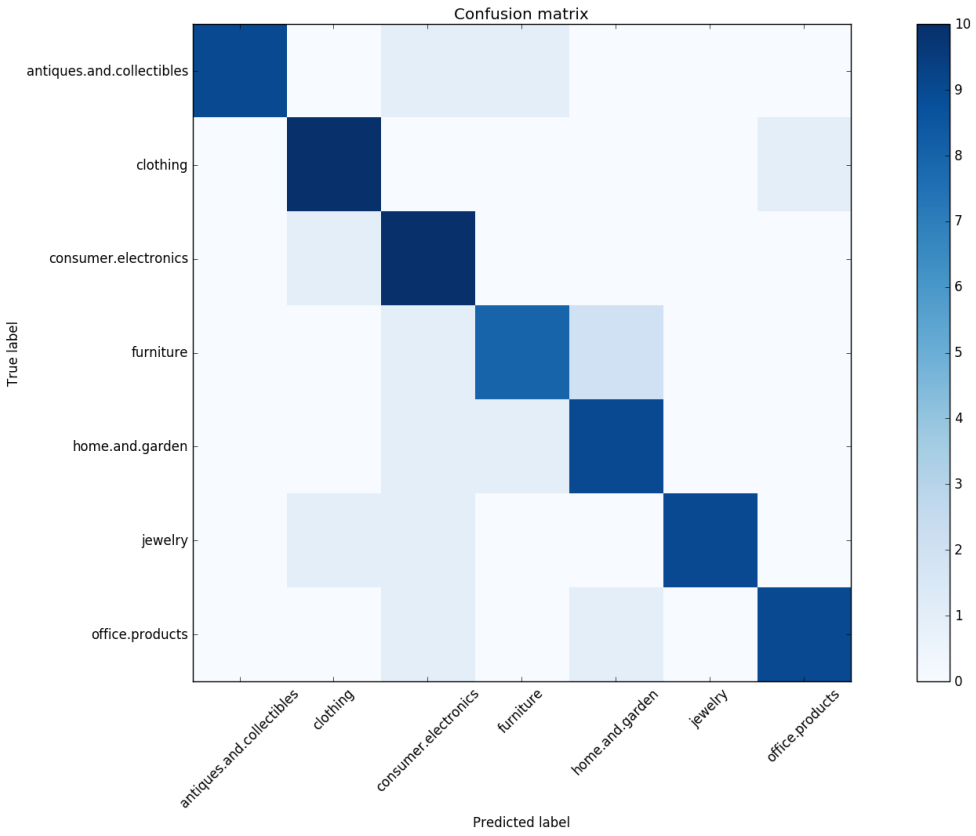
3. Tudi z nevronsko mrežo, smo dobili klasifikacijsko točnost 85%. Uporabljene so bile privzete nastavitve algoritma iz knjižnice *scikit-learn*, spremenili smo le parameter *n_iter*, ki je zadolžen za število prehodov skozi učno množico. Nastavili smo ga iz 5 na 50.

4. Z metodo naključnih gozdov smo doesgli klasifikacijsko točnost 83%. Uporabljene so bile privzete nastavitve algoritma iz knjižnice *scikit-learn*, spremenili smo samo vrednost za število dreves iz 10 na 100 (parameter *n_estimators*).

5. Za algoritem k-najbližjih sosedov smo dobili 82% klasifikacijsko točnost. Število sosedov k smo nastavili na vrednost 100 (parameter *n_neighbors*), ker z večjim številom sosedov smo dobili boljše rezultate. Za vse ostale parametre smo pustili privzete vrednosti.

6. Binarni naivni Bayesov klasifikator je dosegel klasifikacijsko točnost 76%. Uporabljene so bile privzete nastavitve algoritma iz knjižnice *scikit-learn*, spremenili smo samo dovzetnost do učenja α na vrednost 0.01.

Slika 4.1 predstavlja matriko zmot za metodo podpornih vektorjev, ki se je izkazala za najboljšo metodo razvrščanja spletnih strani. Vsota vsake vrstice podaja delež pravilnih razredov, vsote stolpcev pa nam povejo število ali delež problemov, ki so uvrščeni v posamezni razred. Iz diagonale matrike zmot, kjer so podana števila pravilnih klasifikacij je razvidno, da je metoda podpornih vektorjev najuspešnejša pri klasificiranju oblačil in zabavne elektronike, najslabša pa pri pohištvu.



Slika 4.1: Matrika zmot za metodo podpornih vektorjev.

Ocenjujemo, da so pridobljeni rezultati dovolj natančni za praktično rabo. Glede na rezultate lahko pričakujemo, da bomo s 87% točnostjo razporejali spletne trgovine v pravilne kategorije.

Poglavje 5

Sklepne ugotovitve

V diplomskem delu smo razvili orodje za pridobivanje spletnih trgovin s spleta ter kategoriziranje spletnih strani namenjenih prodaji. Naš spletni pajek deluje na osnovi asinhronih poizvedb. Na seznamu domen izvaja poizvedbe in shrani njihovo besedilno spletno vsebino. Za kategorizacijo trgovin uporabljmo metode strojnega učenja. Rezultati prototipa specializiranega spletnega iskalnika so spodbudni, z nekaj izboljšavami bi iskanje in kategoriziranje lahko nadgradili do mere, da bi postalo avtomatsko. Pri praktični rabi smo zaznali več priložnosti za izboljšanje, ki jih v nadaljevanju opišemo.

Izboljšave pridobivanja podatkov:

1. Za izgradnjo nabora podatkov je bilo potrebno ročno obiskati vse spletne strani, ter preveriti ali so pravilno kategorizirane. Izločiti je bilo potrebno napake, ki so bile prisotne v prvotnem naboru podatkov. Za hitrejše pregledovanje bi lahko razvili prilagojen spletni brskalnik, ki bi na vhod dobil seznam domen in preko dveh gumbov shranjeval relevantne spletne strani. Tako orodje bi nam omogočilo lažje pregledovanje velike količine spletnih strani in bi bilo uporabno za izločanje napačnih spletnih strani.

2. Pridobivanje novih spletnih strani: S pomočjo *TLD zone* datotek bi lahko vsak dan pridobili sezname novo registriranih domen, ki bi jih

lahko uporabili v spletnem pajku.

Možne izboljšave spletnega pajka:

1. Uporaba knjižnice *Scrapy framework* (`https://scrapy.org/`), ki je robustna knjižnica, namenjena pridobivanju podatkov iz različnih virov.

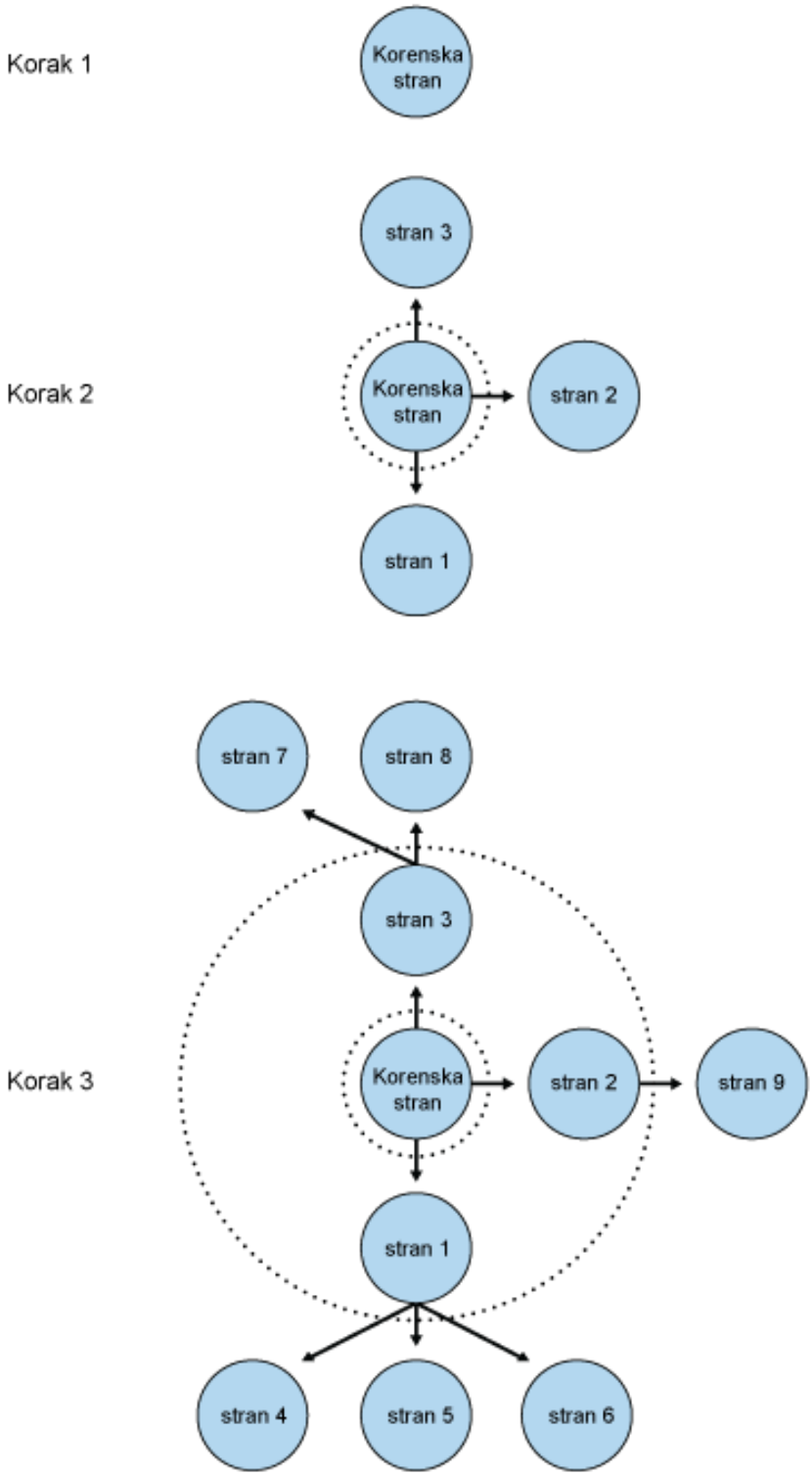
2. Nadgradnja spletnega pajka z uporabo omejenega iskanja v širino (*bound BFS*). Iskanje v širino je osnovni algoritem za spletno preiskovanje. Preiskuje graf, določen z izhodiščnim vozliščem, kar je v našem primeru korenska spletna stran, kakor je razvidno na sliki 5.1. Omejeno preiskovanje v širino preiskuje v širino do določene globine. To bi nam omogočilo izgradnjo obsežnejšega nabora podatkov.

3. Nadgradnja spletnega pajka z uporabo različnih uporabniških posrednikov(*user agent*) ob vsaki poizvedbi. S tem se lahko izognemo blokadam spletnega pajka, ker simuliramo vsakič drug brskalnik in operacijski sistem.

4. Uporaba posredniškega strežnika: Z pomočjo spletne strani `http://proxymesh.com`, ki ponuja spreminjajoče posredniške strežnike, bi lahko nadgradili naš spletni pajek tako, da bi uporabili izmenjujoče se posredniške strežnike in se izognili blokadam določenih *IP* naslovov, kar pride v upoštev pri spletnih straneh, ki blokirajo uporabnike določenih držav.

5. S pomočjo knjižnice *Langdetect* je mogoče nadgraditi spletnega pajka tako, da določi jezik spletne strani. Če npr. spletna stran ni v angleškem jeziku, jo je mogoče prevesti in šele nato kategorizirati.

6. Ker se naslovni prostor *IPv4*, ki je trenutno najbolj razširjen, zapolnjuje, je smiselno prilagoditi spletnega pajka tako, da obiskuje tudi naslovni prostor *IPv6*, na katerem že delujejo nekatere spletne trgovine.



Slika 5.1: Vizualizacija delovanja algoritma z omejenim iskanjem v širino.

Možne izboljšave klasifikatorja:

1. Klasifikator bi lahko napovedal ciljni spol določene strani trgovine. Želimo napovedati ali spletna trgovina ponuja samo ženske, samo moške izdelke ali oboje. Prav tako bi lahko uvedli večnivojsko klasifikacijo, ki bi kategorije razvrstila še na podkategorije.

Literatura

[1] I. Kononenko in M. Robnik Šikonja. *Inteligentni sistemi*. Ljubljana: Fakulteta za računalništvo in informatiko, 2010.

[2] I. Kononenko. *Strojno učenje*. Ljubljana: Fakulteta za računalništvo in informatiko, 2005.

[3] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. New York: Springer, 2011.

[4] M. Zorman in drugi. *Inteligentni sistemi in profesionalni vsakdan*. Maribor: Center za interdisciplinarne in multidisciplinarne raziskave in študije Univerze v Mariboru, 2003.

[5] Napovedovanje vrednosti z algoritmom K najbližjih sosedov. [Online]. Dosegljivo: `https://dk.um.si/Dokument.php?id=10770`. [Dostopano 27. 11. 2016].

[6] Python. [Online]. Dosegljivo: `https://www.python.org/`. [Dostopano 27. 11. 2016].

[7] Python requests. [Online]. Dosegljivo: `http://docs.python-requests.org/en/master/`. [Dostopano 27. 11. 2016].

[8] Python html2text. [Online]. Dosegljivo: `https://pypi.python.org/pypi/html2text/2016.9.19`. [Dostopano 27. 11. 2016].

[9] Perceptron. [Online]. Dosegljivo: `http://lcn.epfl.ch/tutorial/english/perceptron/html/learning.html`. [Dostopano 27. 11. 2016].