UNIVERZA NA PRIMORSKEM

FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

PROJEKTNI SEMINAR I – POROČILO

Detekcija vegetacije iz satelitskih podatkov

Pripravil: Duško Topić

MAJ 2017

Kazalo vsebine

1. UVOD

V tem dokumentu bo podrobneje predstavljena problematika ter pristop k reševanju problematike zaznavanja vegetacije na določenem geografskem področju. Prikazane bodo uporabljene metode v aplikaciji, podprte z več diagrami. Poleg rezultatov natančnosti aplikacije bo predstavljen tudi način uporabe na realnem primeru, od vhodnih podatkov, do samega rezultata končnega produkta.

1.1. Razumevanje problema

Na področju agronomije je ena izmed problematik, kako spremljati, ali se na določenem področju nahaja katera od poljščin ter spremljanje njenega razvoja in širitve oziroma manjšanja površine. Cilj celotnega projekta je pripraviti model, ki bo na podlagi satelitskih posnetkov znal ustrezno klasificirati, ali se na izbranem področju nahaja poljščina ali ne. Namen projekta je pridobiti podatke za pripravo zemljevida vegetacije na želenem področju.

2. PODATKOVNA BAZA

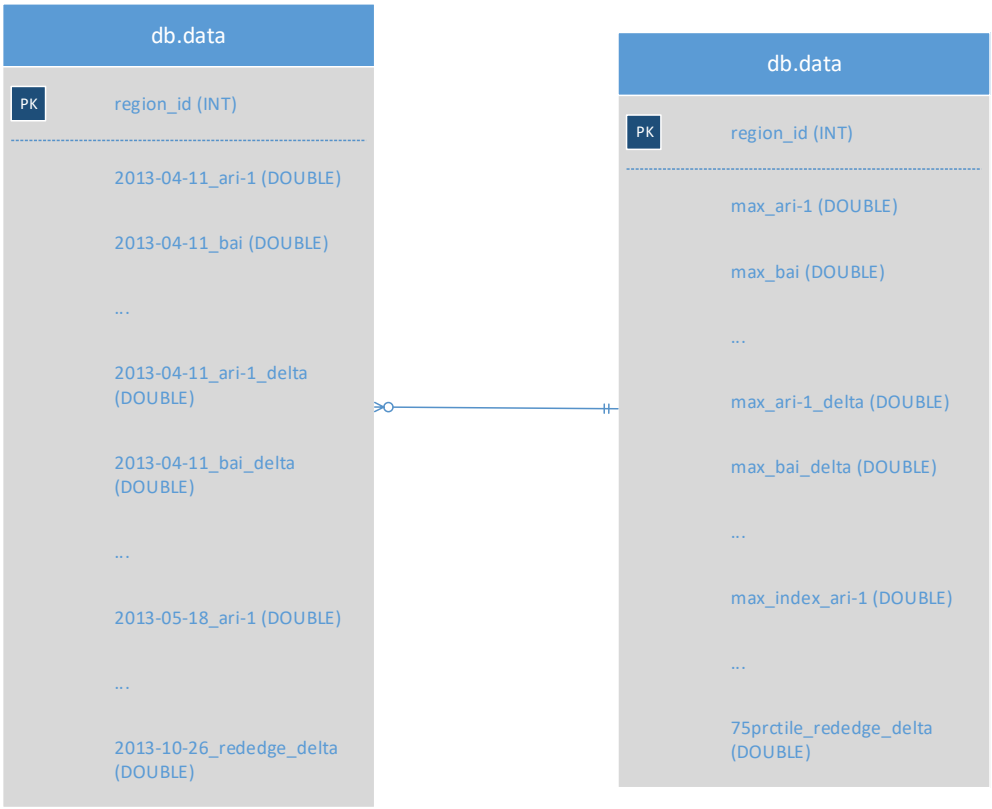
Podatkovna baza je namenjena detekciji vegetacije iz satelitskih podatkov, zgrajena je iz 619644 poligonov, kjer vsak poligon predstavlja točno določeno geografsko območje in hkrati je za vsak poligon pridobljenih 11 satelitskih meritev v letih 2012 in 2013. Definiranih je 9 različnih tipov poljščin:

- ječmen
- pšenica
- koruza
- oljna buča
- oljna ogrščica
- tritikala
- neznana poljščina
- druga poljščina
- ne-poljščina

Prvih 8 tipov hkrati spada pod kategorijo poljščina, medtem ko pod zadnji tip spadajo vsi preostali poligoni, ki niso klasificirani kot poljščina. V našem primeru bomo uporabili zgolj meritve iz sezone 2013 – teh je 9, in sicer med aprilom in oktobrom, meritve so zajete na približno 3-4 tedne.

2.1. Vhodna podatkovna baza

Da bi pripravili model, ki bo izpolnil cilj, ga je potrebno najprej naučiti na primerljivih podatkih, zato bomo najprej napisali referenčno podatkovno bazo, nad katero smo model gradili. Vhodna podatkovna baza se nahaja na levi tabeli v Sliki 1, ki predstavlja diagram vhodne podatkovne baze. Desna tabela predstavlja diagram preurejene podatkovne baze – kar bomo razložili v nadaljevanju. Zaenkrat si oglejmo levo tabelo.



Slika 1: ER diagram vhodne in izpeljane podatkovne baze.

Podatkovna baza je sestavljena iz satelitskih meritev, ki so za posamezno področje beležile naslednjih 11 parametrov:

- ARI-1 (ang. *Anthocyanin Reflectance Index 1*) beleži vrednost antocianina – gre za vodotopen pigment, veliko ga je v novih in odmrlih listih. Višje vrednosti so praviloma v pomladnih in jesenskih mesecih
- BAI (ang. *Burn Area Index*) beleži indeks pogorelosti zemlje. Višje vrednosti je opaziti v poletnih mesecih, kar je za pripisati večjim sušnim obdobjem
- Blue: modra valovna dolžina
- Red: rdeča valovna dolžina
- Green: zelena valovna dolžina
- NIR (ang. *Near Infra Red*) beleži vrednost valovne dolžine 700-1200nm
- RedEdge: valovna dolžina 690-730nm
- PSRI-NIR (ang. *Plant Senescence Reflectance Index*) beleži indeks odbojnosti senescence rastlin v ti. NIR območju
- ChlRedEdge beleži razmerje med vrednostjo klorofila v dveh valovnih dolžinah (rdeča ter RedEdge)
- NDVI (ang. *Normalized Difference Vegetation Index*) je vegetacijski indeks, ki beleži stopnjo vegetacije. Je najbolj znan in tudi najbolj razširjen vegetacijski indeks. Razpon vrednosti indeksa je med -1 in 1, kjer negativne vrednosti predstavljajo vodnato področje (npr. morje, jezero), vrednosti okrog ničle predstavljajo gola, nerodovitna področja (npr. kamen, puščava, sneg), medtem ko pozitivne vrednosti predstavljajo

znake rastja, od travnatih površin pri nizkih vrednostih, pa vse do tropskega pragozda (bogato rastje) pri vrednosti 1.
- NDVI-Green: produkt indeksa NDVI ter zelene valovne dolžine

Vsi zgoraj navedeni parametri so bili izmerjeni v 9-ih različnih časovnih obdobjih:

- 2013-04-11
- 2013-05-18
- 2013-06-15
- 2013-07-02
- 2013-07-29
- 2013-08-18
- 2013-09-14
- 2013-10-08
- 2013-10-26

Da bi dobili še večjo relacijo med meritvami v dveh časovnih obdobjih, je dodatno vpeljan parameter delta, ki beleži spremembo vrednosti parametra med dvema zaporednima poligonoma v določenem času. V bistvu gre za funkcijo odvoda:

$$\Delta_t = \frac{\sum_{\mathrm{T}=1}^{\Theta} \mathrm{T}(c_{t+\mathrm{T}} - c_{t-\mathrm{T}})}{2 \cdot \sum_{\mathrm{T}=1}^{\Theta} \mathrm{T}^2}$$

Zato imamo pri vhodni podatkovni množici skupno (11*2)*9=198 atributov ter atribut ID z oznako poljščine. Kot omenjeno, je zapis atributov naveden v levi tabeli na Sliki 1.

2.2. Izpeljava nove podatkovne baze

Problem take podatkovne baze je v tem, da je preveč statična, kar pomeni, da ni možno pripraviti modela, ki bi znal učinkovito delovati nad novo podatkovno bazo, ki bi imela različno število časovnih meritev. Rešitev je v izpeljavi atributov na način, da bo neodvisen od števila časovnih meritev. Postopek se imenuje izpeljava časovnih značilk[2]. V praksi to pomeni, da bo v fazi predpriprave podatkov iz izvorne podatkovne baze potrebno izpeljati ustrezne značilke, ki bodo predstavljale novo podatkovno bazo za modeliranje. Cilj postopka je imeti nespremenjeno število atributov, ne glede na količino časovnih meritev. S tem dosežemo, da ni potrebno prilagajati modela v primeru dodajanja meritev novega časovnega obdobja. Izpeljane značilke bodo v našem primeru statistični funkcionali, navedeni v Tabeli 1.

Prvi štirje statistični funkcionali predstavljajo najvišjo ter najnižjo vrednost znotraj vektorja, ter pripadajoči indeks, kjer se vrednost nahaja. Naslednji statistični funkcional predstavlja razpon (ang. *range*) v meritvah, torej razliko med najvišjo in najnižjo vrednostjo vektorja. Nato sledita povprečna vrednost vektorja, ter povprečna absolutna sprememba – to je razlika med izbrano vrednostjo v primerjavi z vrednostjo meritve v naslednjem časovnem obdobju. Sledi statistična metoda, ki meri razpršenost podatkov, in sicer standardni odklon. Asimetrija (ang. *skewness*) in sploščenost (ang. *kurtosis*) ugotavljata, koliko so podatki razpršeni, oz. koliko se razlikujejo od povprečja. Visoka vrednost asimetrije pomeni, da podatki niso simetrični, visoka vrednost sploščenosti pa pomeni, da je prisotno veliko število ti. osamelcev (ang. *outlier*). V

primeru nizkih vrednosti velja ravno nasprotno, torej pri asimetriji so podatki simetrični, pri sploščenosti pa nimamo osamelcev.

Opis	**Izračun**	**Ime**
Najvišja vrednost s pripadajočim indeksom	$\max x_i$	max; max_index
Najnižja vrednost s pripadajočim indeksom	$\min x_i$	min; min_index
Razpon	$\max x_i - \min x_i$	range
Povprečje	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$	mean
Povprečna absolutna sprememba	$\frac{1}{N}\sum_{i=1}^{N-1} \lvert x_i - x_{i+1} \rvert$	aac
Standardni odklon	$\sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$	stdev
Asimetrija	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\right)^3}$	skewness
Sploščenost	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^2}$	kurtosis
Križno povprečje	$št. križanj\ od\ povprečja$	crossmean
25-percentil	$n = \left\lceil \frac{25}{100} \cdot N \right\rceil$	25prctile
50-percentil	$n = \left\lceil \frac{50}{100} \cdot N \right\rceil$	50prctile
75-percentil	$n = \left\lceil \frac{75}{100} \cdot N \right\rceil$	75prctile

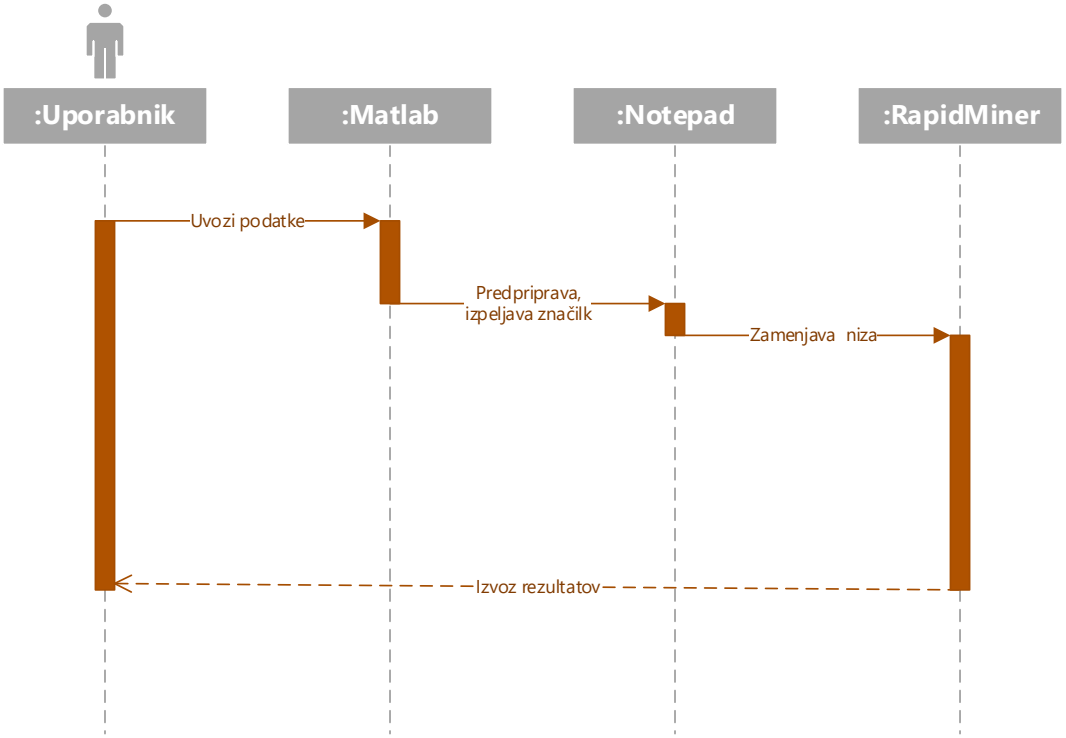
Tabela 1: Statistični funkcionali, izpeljani nad vhodnim vektorjem $x = [x_1, x_2, \ldots, x_n]$, kjer n predstavlja število časovnih obdobij, x_i pa meritev v i-tem časovnem obdobju.

Naslednji statistični funkcional je križno povprečje, ki beleži, kolikokrat signal preseka povprečno vrednost. Zadnji trije funkcionali predstavljajo 25-percentil, mediano (oz. 50-percentil) ter 75-percentil. Skupno imamo izpeljanih 14 statističnih funkcionalov. Ker smo vseh 14 statističnih funkcionalov aplicirali nad vsakim parametrom, vključno z deltami – skupno torej nad 22-imi parametri, imamo v tem primeru skupno 308 atributov v podatkovni zbirki. Za lažje razumevanje si lahko ogledamo desno tabelo na Sliki 1, kjer so zapisani izpeljani atributi nove podatkovne baze. Najprej je zapisanih 11 atributov v zaporedju »funkcional_parameter«, kjer se poleg določenega statističnega funkcionala zapišejo vsi izmerjeni parametri (z izjemo delte). Temu sledi nadaljnjih 11 atributov v zaporedju »funkcional_parameter_delta«, kjer so za posamezen statistični funkcional in parameter zapisane še vrednosti parametra delta. Tako imamo za posamezen statistični funkcional skupno 22 atributov. Nato se postopek ponovi s preostalimi statističnimi funkcionali.

3. MODEL

Za izvedbo projekta in pripravo modela smo uporabili tri orodja, in sicer proces predpriprave podatkov ter izračun statističnih funkcionalov smo izvedli z orodjem Matlab[4] s pomočjo razširitvenega modula »Statistics and Machine Learning Toolbox«. Pripravljeno podatkovno množico smo nato uvozili v urejevalnik datotek (npr. Notepad), kjer smo spremenili zapis črke

»e« v »E« zaradi različnega načina branja datotek pri obeh orodjih. Nato smo podatke uvozili v RapidMiner[5], s katerim smo izpeljali različne tehnike modeliranja in tudi analizirali pridobljene rezultate, kot bo podrobneje predstavljeno v nadaljevanju. Celoten proces je tudi grafično prikazan na diagramu aktivnosti na Sliki 2.



Slika 2: Diagram aktivnosti za celoten proces pridobivanja želenih rezultatov.

3.1. Predpriprava podatkov

Kot je razvidno iz diagrama na Sliki 2, je prvi korak celotnega procesa, da uporabnik uvozi izvorno podatkovno bazo v Matlab, kjer se jo ustrezno procesira, da bo primerna za izgradnjo modela. V Matlabu izpeljemo proces izpeljave statističnih funkcionalov, navedenih v Tabeli 1. Za nekatere statistične funkcionale Matlab že podpira vgrajene funkcije, nekatere pa je bilo ročno programirati (npr. križno povprečje). Poleg statističnih funkcionalov izračunamo še pripadajoče delte.

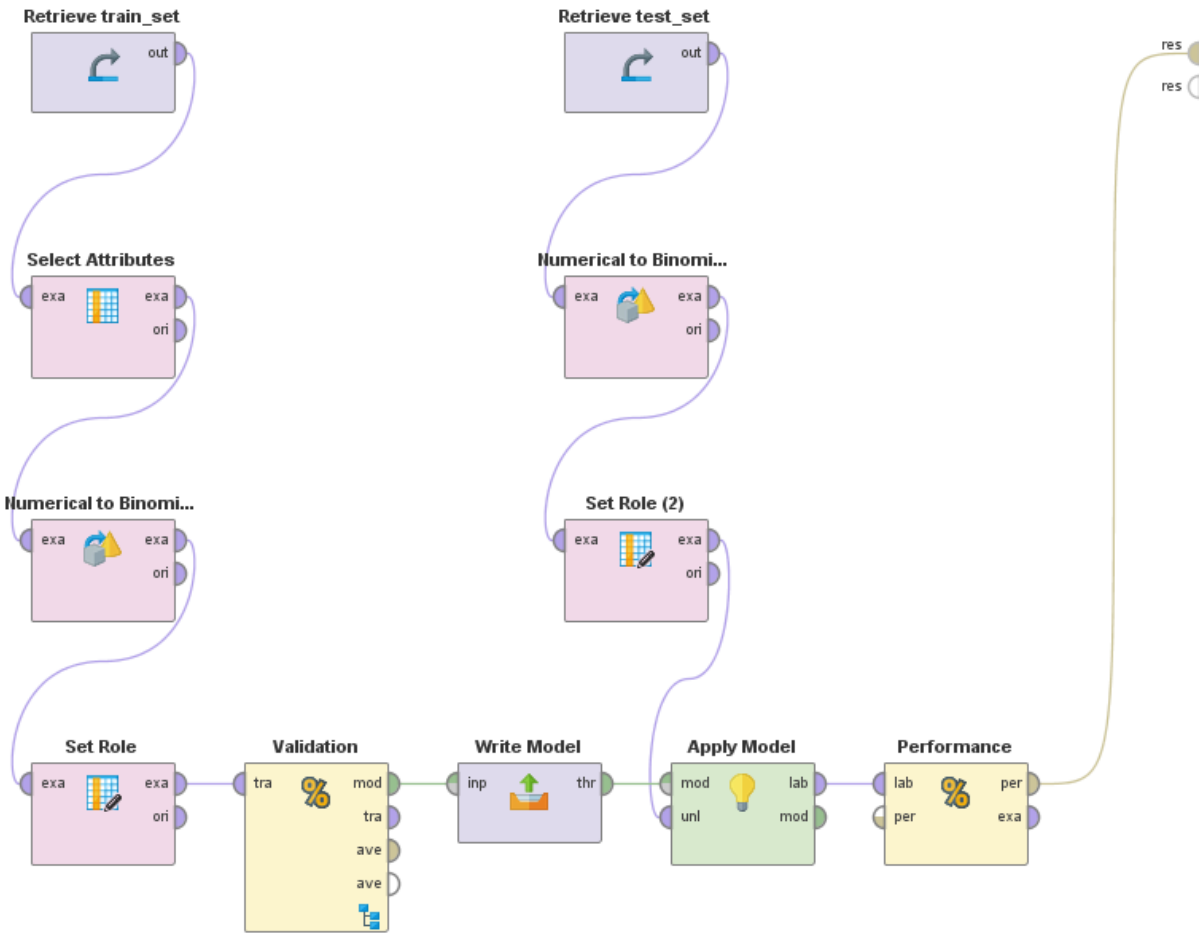
Za potrebe priprave ter učenja modela je ob vsakem vzorcu v bazi dodanih še 9 binarnih atributov, ki definirajo tip poljščine. Torej, če je vzorec tritikala, vsebuje atribut »Tritikala« vrednost 1, vseh ostalih 8 atributov pa vrednost 0.

Po zaključenih izračunih generiramo imena atributov v zaporedju »funkcional_parameter« oziroma »funkcional_parameter_delta« ter jih v ustreznem vrstnem redu zapišemo v prvo vrstico nove datoteke formata CSV. Vse nadaljnje vrstice seveda predstavljajo vrednosti omenjenih atributov, kjer vsaka vrstica predstavlja svoje geografsko področje. Za namene testiranja smo izvorno podatkovno bazo razdelili na dva dela – učno in testno bazo v razmerju 2/3 – 1/3 v korist učne baze. Vsaka podatkovna baza se nahaja v svoji datoteki formata CSV.

Izhodno datoteko smo nato uvozili v urejevalnik datotek (npr. Notepad), kjer smo zamenjali vse nize »e« z veliko začetnico »E«, zaradi različne interpretacije decimalnih številk med orodjema Matlab in RapidMiner. Če bi ta proces preskočili, bi RapidMiner take primere zaznal kot manjkajoče vrednosti.

3.2. Modeliranje

Celoten proces modeliranja v RapidMiner se nahaja na Sliki 3. Posebnost orodja je v tem, da so vsi procesi definirani s ti. operatorji – to so procesi, ki predstavljajo posamezno dejanje (npr. filtriranje podatkov, klasifikacija z izbranim algoritmom). Posamezen operator je lahko sestavljen iz več podprocesov. S povezovanjem operatorjev je možno vizualno predstaviti celoten postopek podatkovnega rudarjenja, z vsemi parametri, vhodnimi ter izhodnimi podatki.



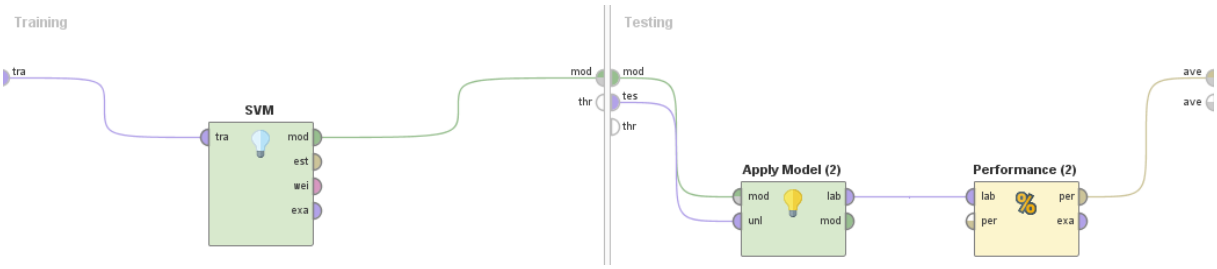
Slika 3: Celoten proces modeliranja v RapidMiner.

Proces se prične z branjem vhodne CSV datoteke (operator »Retreive«). Na levi strani najprej uvozi učno bazo. Sledi operator »Select Attributes«, kjer izberemo le tiste atribute, s katerimi želimo nadaljevati proces. V tem primeru smo izločili 8 atributov, ki določajo tip rastja – ohranili smo torej le atribut »Ne-poljščina«, saj nam je cilj zaznati, ali je vzorec poljščina ali ne. Na tip poljščine se v tem projektu nismo osredotočali.

Zatem uporabimo operator »Numerical to Binomial«, kjer atribut »ne-poljščina« pretvorimo v binomski atribut, kar v praksi pomeni, da vrednost »0« pretvorimo v »false« ter vrednost

»1« pretvorimo v »true«. Kot bomo videli v nadaljevanju, je ta pretvorba nujna, saj algoritem za klasifikacijo deluje le nad binarnim razrednim atributom. Z operatorjem »Set Role« eksplicitno definiramo razredni atribut, to je »ne-poljščina«.

Naslednji v izvedbi je operator »Validation« - gre za metodo navzkrižnega preverjanja (ang. *Cross Validation*). V našem primeru smo uporabili standardno 10-kratno navzkrižno preverjanje. Ta operator vsebuje podproces, ki se 10-krat izvede periodično na enak način, vsakič z drugo desetino testne množice. Podproces je definiran na Sliki 4. Leva polovica se izvede v fazi učenja – izbrali smo algoritem SVM. Algoritem bo podrobneje predstavljen v posebni sekciji. V fazi procesa testiranja izpeljemo ravno to – testiramo naučen model (operator »Apply Model«) ter preverimo natančnost modela (operator »Performance«).



Slika 4: Podproces »Validation«, kjer se izvaja proces 10-kratnega prečnega preverjanja.

Po izpeljanih 10-ih iteracijah navzkrižnega preverjana, pridobljen model izvozimo v datoteko za kasnejšo uporabo modela nad novimi podatki. To naredimo z operatorjem »Write Model« na Sliki 3. V tej fazi imamo pripravljen model, ki je naučen nad učno množico. Da bi se prepričali, ali je model sprejemljiv oziroma dovolj natančen, ga bomo testirali na način, da poskusimo klasificirati testno podatkovno bazo – torej tisto bazo, katere v fazi učenja nismo obravnavali. Testno podatkovno bazo uvozimo v RapidMiner na enak način, nato pa z operatorjem »Apply Model« klasificiramo nove primere. Za konec uporabimo operator »Performance«, da preverimo še natančnost modela.

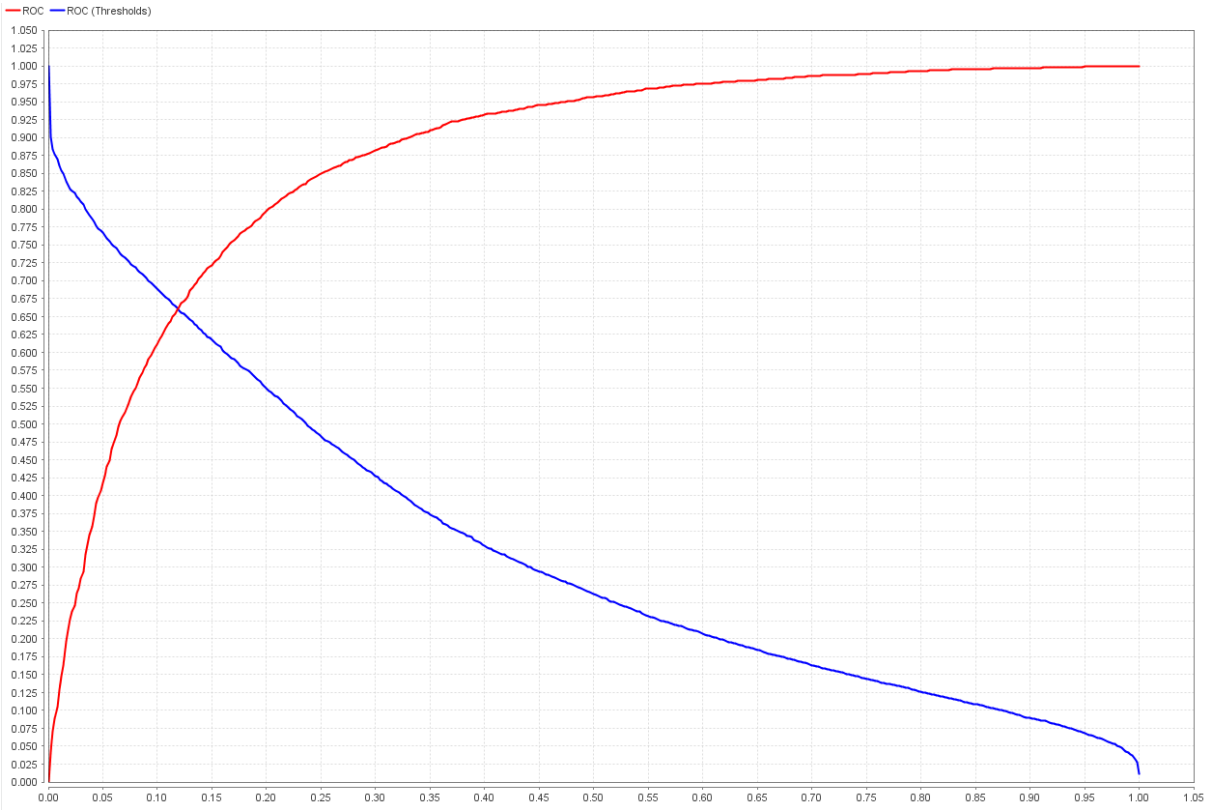
3.2.1. SVM

Modeliranje smo izvedli z algoritmom SVM oziroma metodo podpornih vektorjev (ang. *Support Vector Machine*). Gre za algoritem, ki določi tolikšno mejo med dvema razredoma, da dosežemo maksimalni razmik med vzorci[3]. Izhodišče za metodo SVM je učna množica meritev, za katere vemo, kateremu razredu (v našem primeru poljščini) pripadajo. Vsako meritev predstavimo z vektorjem v vektorskem prostoru, z metodo SVM nato poiščemo v tem večdimenzionalnem prostoru hiperravnino, ki ločuje meritve iz različnih razredov. Razdaljo vektorjev, ki se nahajajo najbližje hiperravnini, poskušamo maksimirati, namreč večja kot je razdalja praznega območja med razredi, toliko bolj natančno deluje klasifikacija novih meritev. Algoritem hkrati za posamezen atribut določi še utež *w*, tako da skupek uteži v bistvu definira vektor na hiperravnini. Višja kot je vrednost uteži, bolj je atribut pomemben za model, na ta način pridobimo še dodatno informacijo o pomembnosti ter vplivu posameznih atributov.

3.3. Rezultati

Kakovost modela smo ocenjevali z ROC krivuljo (ang. *Receiver Operating Characteristics*). Gre za dvodimenzionalen graf[1], kjer *x* os predstavlja delež lažnih pozitivnih primerov (ang. *False Positive Rate*), *y* os pa delež resničnih pozitivnih primerov (ang. *True Positive Rate*). Oglejmo si Sliko 5, ki prikazuje ROC krivuljo za pridobljen model. Z rdečo barvo je označena krivulja, ki predstavlja ROC, z modro barvo pa krivulja, ki določa prag oz. mejno vrednost napovedi verjetnosti pozitivnega razreda. Idealen scenarij je, ko se rdeča krivulja nahaja višje proti levem zgornjem kotu. Krivulja se v konkretnem primeru nahaja dokaj visoko, v praksi to pomeni, da bomo razpoznali veliko pravilno napovedanih pozitivnih primerov, hkrati pa bo stopnja napačno ocenjenih pozitivnih primerov nižja.

Iz omenjenega grafa lahko opazujemo natančnost modela le vizualno. Da bi lahko pridobili neko objektivno številsko natančnost, je dodatno vpeljan parameter AUC (ang. *Area Under the ROC Curve*), ki izračuna ploščino prostora na spodnji desni strani pod ROC krivuljo. Vrednost AUC bo vedno v razponu med 0 in 1 (v nadaljevanju bomo označevali razpon med 0-100% površino), saj računa ploščino na enotskem kvadratu. V praksi pomeni, da mora vrednost AUC stremeti k 100% pokritosti, vsekakor pa mora biti vsaj strogo večja od 50%, saj gre v nasprotnem primeru za neustrezen model, ker bi v takem primeru dosegli enako natančnost, kot če bi naključno določali pripadnost razredu.



Slika 5: Primerjava ROC krivulj obeh pristopov za detekcijo ne-poljščine.

Vrednost AUC smo pridobili ravno z operatorjem »Performance«. AUC ploščina je pri dobljenem modelu znašala 82.3%, kar predstavlja sprejemljiv rezultat, saj ni potrebe po natančni določitvi popolnoma vseh geografskih področij.

Da bi bolje razumeli pridobljene rezultate, si oglejmo še uteži atributov, ki jih je določil SVM algoritem glede na razredni atribut. Tabela 2 prikazuje uteži prvih 5 najbolj informativnih atributov. Kot je razvidno, sta najbolj informativna parametra PSRI-NIR ter NDVI. Iz celotnega seznama uteži pri modelu gre razbrati, da ima parameter PSRI-NIR občuten vpliv na detekcijo ne-poljščine. Naslednji parametri se med seboj izmenjujejo po pomembnosti, nekoliko izstopata parametra NDVI ter zelena valovna dolžina v različnih mesecih. Poleg PSRI-NIR ima prav tako pomembno vlogo parameter NDVI, kar je tudi razumljivo, saj parameter predstavlja stopnjo vegetacije.

Atribut	Utež
75prctile_psri-nir	1.498
mean_psri-nir	1.381
25prctile_ndvi_delta	1.377
stdev_ndvi_delta	1.295
stdev_ndvi	1.272

Tabela 2: Pregled uteži atributov algoritma SVM.

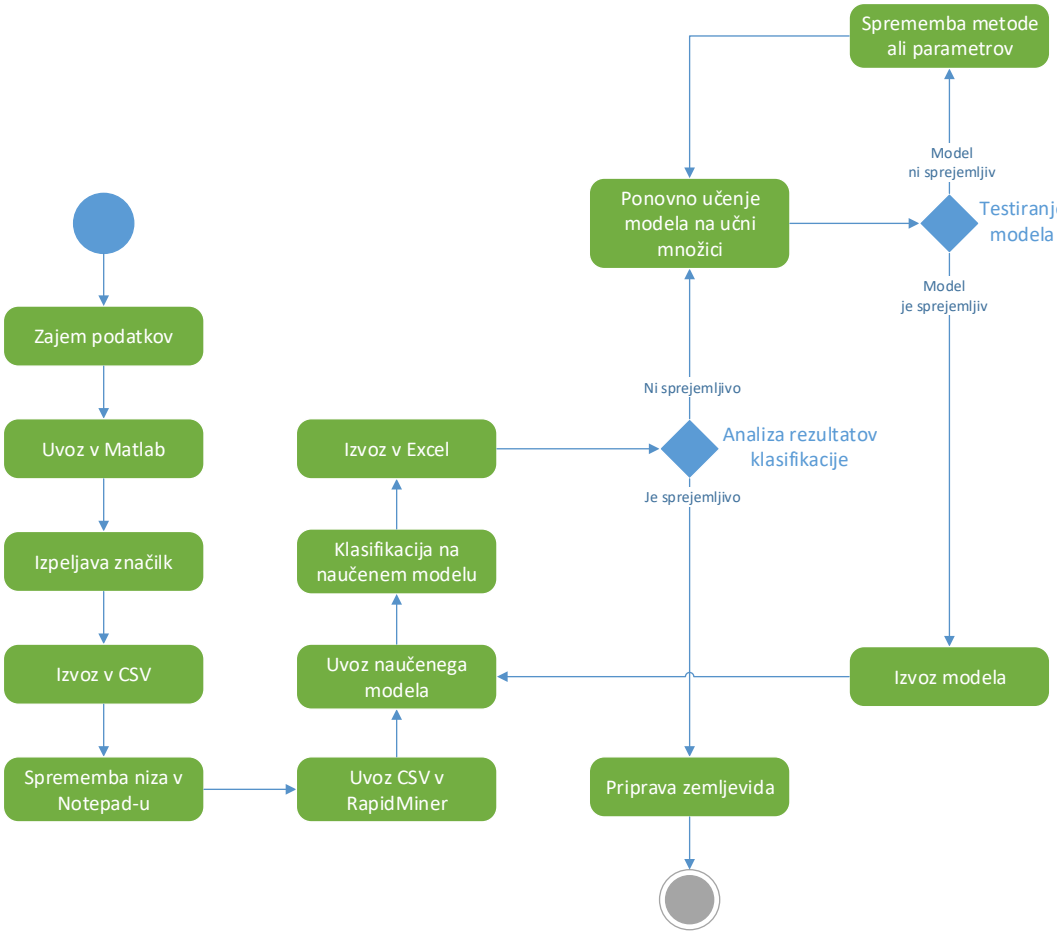
4. UPORABA

Do sedaj so bile podrobneje definirane funkcijske ter tehnične podrobnosti projekta. V tem poglavju bo zapisan proces uporabe aplikacije z že pripravljenim modelom. Slika 6 prikazuje diagram aktivnosti pri uporabi modela v praksi. Proces se prične z zajemom satelitskih podatkov želenega področja. Nato se izpeljejo postopki, kot so opisani v prejšnjem poglavju – od uvoza ter obdelave v Matlabu, do klasifikacije novih podatkov v RapidMiner-ju. Klasificirane podatke nato izvozimo v Excelov format datoteke, kjer ga uporabnik analizira na način, da preveri določeno manjše področje, ali je ustrezno klasificirano. V kolikor je rezultat za uporabnika sprejemljiv, lahko nadaljuje s pripravo zemljevida vegetacije področja – ta del je v domeni uporabnika in ni predmet tega projekta.

V kolikor uporabnik zazna prevelika odstopanja v napovedani klasifikaciji, pomeni da je potrebno model ustrezno prilagoditi. V tem primeru je potrebno pripraviti novo učno ter testno množico, kjer bomo ponovno izpeljali proces modeliranja. Morda bo zadostovala le sprememba nekaj parametrov, v skrajnem primeru pa bo potrebno spremeniti metode, algoritme itd. Model se testira na testni množici, vse dokler ne dosežemo sprejemljive natančnosti. Ko dosežemo sprejemljivo natančnost novega modela, ga izvozimo, da ga lahko uporabnik kadarkoli ponovno uvozi v RapidMiner za klasifikacijo novih podatkov.

Kot je iz diagrama aktivnosti razvidno, v teoriji proces zahteva kar nekaj dinamike, kajti model najbolje deluje na takih tipih podatkov, kot je naučen. V praksi to pomeni, da bo model najbolje deloval v podobni okolici podatkov, nad katerimi se je izvajal proces učenja. Za omenjen projekt je tak pogoj zadosten, saj je cilj klasificirati določeno področje, ki ima podobne karakteristike.

V nadaljevanju bosta podrobneje predstavljena primera uporabe aplikacije, prvi primer bo prikazal, kako se izvede proces učenja, drugi primer pa bo izpeljal klasifikacijo na že naučenem modelu na konkretnem primeru novih podatkov.



Slika 6: Diagram aktivnosti pri uporabi načrtovanega modela.

4.1. Primer uporabe priprave modela

Ko so satelitski podatki zajeti, razdelimo podatke na dva dela, najbolje 2/3 meritev za učno množico (datoteka »X_train_develop10_noatt.csv«) ter 1/3 za testno množico (datoteka »X_test_develop10_noatt.csv«). Zatem znotraj aplikacije Matlab zaženemo naslednjo funkcijo:

```
>> feature_extraction_delta_ucenje
```

Omenjena funkcija bo uvozila obe datoteki ter nad podatki izpeljala statistične funkcionale, navedene v Tabeli 1. Rezultat bo nova podatkovna množica, zapisana v datotekah »X_features_delta_train_develop10.csv« ter »X_features_delta_test_develop10.csv«. Poleg tega funkcija v prvi vrstici obeh datotek zapiše imena vseh atributov, da bodo le-ti ob uvažanju v RapidMiner ustrezno poimenovani.

V naslednjem koraku obe izvoženi datoteki odpremo s poljubnim urejevalnikom besedil, v našem primeru smo uporabili Notepad. Zaženemo funkcijo »Replace«, kjer vse pojavitve niza »e« zamenjamo z nizom »E«. To je potrebno narediti, da bo RapidMiner ustrezno interpretiral vrednosti decimalnih števil. Primer: Število »7.4258e-05« se zapiše kot »7.4258E-05«. Po zaključeni zamenjavi shranimo spremembe kar v enako datoteko.

Sledi uporaba orodja RapidMiner za proces modeliranja. Obe CSV datoteki uvozimo v RapidMiner, zatem izpeljemo proces, kot je naveden na Sliki 3 ter zaženemo model. Če z dobljenim rezultatom natančnosti nismo zadovoljni, ustrezno spremenimo parametre v modelu ter poskusimo znova.

Ko smo z rezultatom zadovoljni, smo s procesom priprave modela zaključili in lahko nadaljujemo s procesom klasifikacije novih podatkov s pomočjo naučenega modela.

4.2. Primer uporabe klasifikacije novih podatkov

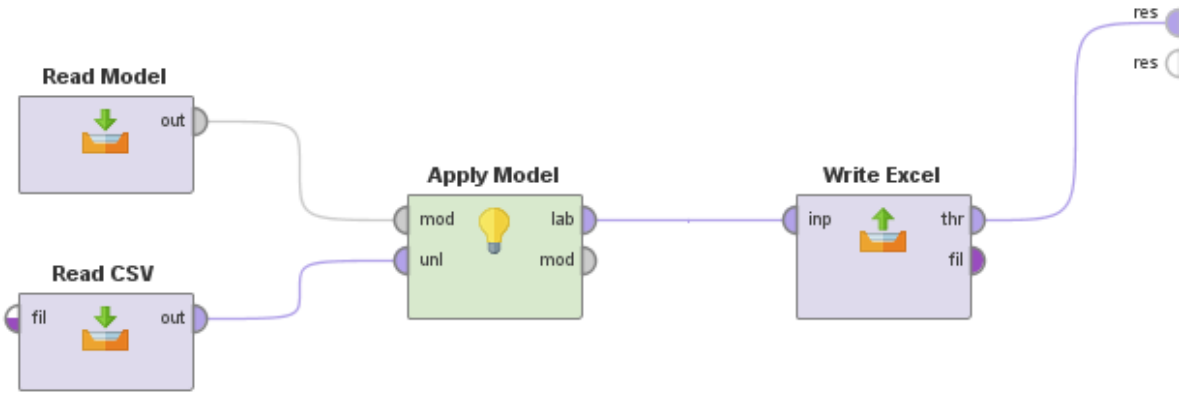
Ko so novi satelitski podatki zajeti, vse parametre shranimo v datoteko »input.csv«. Zatem znotraj aplikacije Matlab zaženemo naslednjo funkcijo:

```
>> feature_extraction_delta_uporaba
```

Omenjena funkcija bo uvozila datoteko »input.csv« ter izpeljala statistične funkcionale, navedene v Tabeli 1. Kot rezultat bo funkcija ustvarila novo datoteko »outputRM.csv«. Poleg tega funkcija v prvi vrstici obeh izhodnih datotek zapiše imena vseh atributov, da bodo le-ti ob uvažanju v RapidMiner ustrezno poimenovani.

V naslednjem koraku izvoženo datoteko odpremo s poljubnim urejevalnikom besedil, v našem primeru smo uporabili Notepad. Zaženemo funkcijo »Replace«, kjer vse pojavitve niza »e« zamenjamo z nizom »E«. To je potrebno narediti, da bo RapidMiner ustrezno interpretiral vrednosti decimalnih števil. Primer: Število »7.4258e-05« se zapiše kot »7.4258E-05«. Po zaključeni zamenjavi shranimo spremembe kar v enako datoteko.

V zadnjem koraku zaženemo RapidMiner ter pripravimo proces, kot je naveden na Sliki 7. Z operatorjem »Read Model« preberemo model, ki smo ga pripravili po navodilih v prejšnji sekciji. Z operatorjem »Read CSV« preberemo pripravljeno datoteko »outputRM.csv«. Z operatorjem »Apply Model« apliciramo naučen model na novih podatkih, na koncu še z operatorjem »Write Excel« zapišemo rezultate klasifikacije v XLS format, recimo »output.xls«. Ko model zaženemo, bo rezultat procesa datoteka »output.xls«.



Slika 7: Proces klasificiranja novih podatkov v RapidMiner.

Analiziramo poljubno število primerov ter preverimo, ali je model pravilno klasificiral primere. Če smo z rezultati zadovoljni, lahko nadaljujemo z uporabo pridobljenih rezultatov (npr. priprava zemljevida vegetacije).

5. ZAKLJUČEK

S pripravo kakovostnega modela lahko trdimo, da smo dosegli zastavljen cilj, torej pripraviti model, ki bo na podlagi satelitskih posnetkov znal ustrezno klasificirati, ali se na izbranem področju nahaja poljščina ali ne.

Možnih izboljšav ter pogledov v nadaljnji razvoj je kar nekaj. Obstoječ model je možno na enostaven način razširiti v detekcijo preostalih tipov poljščin, torej ali se na območju nahaja specifična poljščina (npr. pšenica). Poleg tega je model možno nadgraditi na način, da bo znal pravilno klasificirati konkreten tip poljščine iz vnaprej določenega nabora.

6. LITERATURA

[1] T. Fawcett: An introduction to ROC analysis, *Pattern Recognition Letters – Special issue: ROC analysis in pattern recognition*, št. 27, str. 861-874, 2006

[2] J. Wiens, J. V. Guttag, E. Horvitz: Patient Risk Stratification for Hospital-Associated C.diff as a Time-Series Classification Task*, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems,* str. 467-475, december 2013

[3] I. H. Witten, E. Frank Data Mining: Practical machine learning tools and techniques, Third Edition, Morgan Kaufmann, 2011.

[4] MATLAB – MathWorks. https://www.mathworks.com/products/matlab.html

[5] RapidMiner – Open Source Data Science Platform. https://rapidminer.com