UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Rok Močnik

Metode za oceno zanesljivosti napovedi učinkov malih molekul

DIPLOMSKO DELO
NA INTERDISCIPLINARNEM UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Blaž Zupan

Ljubljana, 2011

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00032/2011
Datum: 08.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Kandidat: **ROK MOČNIK**

Naslov: **METODE ZA OCENO ZANESLJIVOSTI NAPOVEDIH UČINKOV MALIH MOLEKUL**

RELIABILITY ESTIMATION FOR PREDICTION OF EFFECTS OF SMALL MOLECULES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Delovanje učinkovin, ki jih razvija farmacevtska industrija, je do določene mere moč napovedati iz strukturnih lastnosti molekul. Zanesljivost napovedi pa je seveda odvisna od modelirane učinkovine. V diplomski nalogi preučite, kako dobro je moč oceniti zanesljivosti napovedi iz atributnih opisov molekulske strukture. Pri tem predvsem uporabite že razvite tehnike ocenjevanja zanesljivosti s področja strojnega učenja. Pri vrednotenju vaših implementacij uporabite javno dostopne podatke iz baze PubChem in strukturne atribute, ki jih je moč pridobiti s prosto dostopnimi orodji.

Mentor:

prof. dr. Blaž Zupan

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan Fakultete za računalništvo in informatiko:

prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:

prof. dr. Andrej Likar

UNIVERZA V LJUBLJANI • Fakulteta za matematiko in fiziko • 6

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Rok Močnik,

z vpisno številko 63060206,

sem avtor/-ica diplomskega dela z naslovom:

Metode za oceno zanesljivosti napovedi učinkov malih molekul

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
  prof. dr. Blaža Zupana
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 14.9.2011 Podpis avtorja/-ice:

Zahvala

Zahvaljujem se mentorju prof. dr. Blažu Zupanu za vse predloge in pomoč pri izdelavi diplomske naloge. Za sprejem v Laboratorij za bioinformatiko, ter vse nasvete in znanje.

Prav tako bi se rad zahvalil vsem svojim sošolcem in sošolkam. Predvsem Anžetu, Nejcu, Slavku ter še posebej Gaji, ki je poskrbela v času študija za največ zapiskov ter nam vsem olajšala delo.

Zahvalila gre tudi mojim staršem za spodbudo, financiranje študija in prenašanja mojih muh.

Efficiency is intelligent laziness.

David Dunham

Kazalo

Seznam uporabljenih kratic in simbolov

BAGV	varianca modela bagging
BVCK	kombinacija metod BAGV in CNK
CNK	lokalno modeliranje napake
LCV	lokalno prečno preverjanje
RF	naključni gozd
RFV	varianca v naključnem gozdu
SAvar	varianca pri analizi občutljivosti
SAbias	pristranost pri analizi občutljivosti

Povzetek

Danes ljudje ustvarjamo ogromne količine podatkov, več kot kadarkoli prej. V teh podatkih se skrivajo pomembne informacije in znanja. Za človeške sposobnosti je količina teh podatkov postala preprosto prevelika, zato poskušamo naučiti računalnik, da nam pomaga pri teh nalogah. To področje se imenuje strojno učenje ter je ena izmed panog umetne inteligence.

Strojno učenje se ukvarja z čim natančnejšim napovedovanjem primerov za katere še ne poznamo željene vrednosti. Uspešnost neke ponavadi ocenjujemo na celotni testni množici. V tej diplomski nalogi pa nas zanima če lahko za posamezen primer natančno določimo oziroma je napoved za ta primer dobra ali ne. Za reševanja tega problema smo v programski paket Orange implementirali kopico že znanih metod za oceno zanesljivosti posameznih primerov. Te metode smo testirali na podatkih o kvantitativnih razmerjih med strukturo in delovanjem molekul.

Rezultati so pokazali, da se med preizkušenimi metodami najbolje obnese ta, ki tehniko ocenjevanja zanesljivosti izbere na podlagi notranjega prečnega preverjanja. Vendar pa ima ta metoda težavo ker na večjih podatkovnih množicah porabi veliko časa. Zato smo v iskanju učinkovitejše rešitve predlagali novo metodo za oceno zanesljivosti, ki pa deluje samo v navezi z naključnim gozdov. Metoda kot oceno uporabi varianco posamezne napovedi v naključnem gozdu. Metoda je hitra saj naključnemu gozdu ne doda ničesar razen izračuna variance, ter zelo stabilna saj na večini podatkovnih množic pokaže dobre rezultate.

Ključne besede:

strojno učenje, ocena zanesljivosti, kvantitativno razmerje med strukturo in delovanjem, naključni gozd

Abstract

Today in all the area of human activities, we gather enormous amounts of data, more than ever before. This data hides important information and knowledge. For human capabilities this amount of data is overwhelming, so we try to develop computer systems to help us with this. Much of essential research in this field is done within machine learning, a subfield of artificial intelligence.

Machine learning often deals with with predicting classes of attribute-value defined examples. Success of predictions is usualy estimated over whole test set. In this thesis we are interested whether we can - for a specific example estimate if the prediction for this example is accurate or not. To solve this problem, we implemented a set of already known methods for reliability estimation of specific examples. The methods were developed within Orange, a Python-based data mining suite. Method were tested on data about quantitative structure-activity relationships.

Among all the methods tested, the best-performing was the approach that selects the technique for reliability estimation based on internal cross-validation. But this method has a flaw. On bigger datasets this approach is computationally very demanding. In search of effective solution we proposed new method for reliability estimation, that only works in association with random forest. The method uses the variance inside specific prediction in random forest for reliability estimation. This approach is quick, because it does not add anything to random forest, except for calculating variance. It also shows good results on bigger datasets.

Key words:

machine learning, reliability estimation, quantitative structure-activity relationship, random forest

Poglavje 1

Uvod

Strojno učenje je ena izmed vej umetne inteligence in znanstvena disciplina, ki se ukvarja z uporabo in ustvarjanjem novih algoritmov, ki računalnikom pomagajo da v ogromnih količinah podatkov razpoznavajo vzorce ter se učijo novih znanj. V današnjem svetu človeštvo ustvarja ogromne količine podatkov. Tako si na primer: internetni trgovci si želijo čim več izvedeti o naših nakupovalnih navadah, biologi se spopadajo s skrivnostmi delovanja človeškega telesa, računalnike uporabljamo, da nam izločajo nezaželena elektronska sporočila, postavljajo medicinske diagnoze in še in še.

1.1 Motivacija in cilji

Ob uporabi metod nadzorovanega učenja za modeliranje podatkov poskušamo doseči čim boljšo napovedno točnost za še ne videne primere, ki niso bili vključeni v proces učenja našega modela. Pogosto opravilo v nadzorovanem učenju je grajenje regresijskih modelov. Ti najpogosteje za oceno, kako kvaliteten je naš regresijski model, uporabljajo koren srednje kvadratne napake (angl. *RMSE - root-mean-square error*). Čeprav takšne mere ocenijo kvaliteto modela z združevanjem napak vseh testnih primerov, ne povedo nič o zanesljivosti napovedi posameznih primerov. Ta informacija pa je lahko zelo pomembna v okoljih kjer se ukvarjamo z zelo veliki tveganji (npr. medicinske diagnoze, borza, navigacija). Tako na teh področjih potrebujemo ocene zanesljivosti za posamezne primere.

Diplomska naloga je nastala v sodelovanju z Laboratorijem za bioinformatiko ter veliko farmacevtsko družbo Astra Zeneca. Astra Zeneca se ukvarja z ustvarjanjem novih zdravil in zdravilnih učinkovin. V tem poslu je prav tako pomembno, da imamo metode, s katerimi lahko določimo, katere napovedi me-

tod strojnega učenja so bolj natančne in katere manj. Tako prihranimo veliko časa in denarja.

Cilj diplomske naloge je preučiti obstoječe metode za oceno zanesljivosti [2, 3, 4] ter jih implementirati v programskem okolju Orange [10] in ustvariti tudi možnosti za grafično uporabo teh orodij. Programsko okolje Orange želimo uporabiti za analizo delovanja teh metod na podatkih o kvantitativnih razmerjih med strukturo in delovanjem molekul, kar je še posebej zanimivo za Astra Zeneco.

1.2 Struktura diplomskega dela

Med uvodom in zaključkom v tem diplomskem delu najdemo še štiri osrednja poglavja. Drugo poglavja opisuje metode strojnega učenja, ki smo jih v tem diplomskem delu uporabili. V tretjem poglavju predstavimo že obstoječe metode za ocene zanesljivosti posameznih napovedi ter definiramo novo metodo za oceno zanesljivosti na naključnem gozdu. V četrtem opišemo postopke in metode, ki smo jih uporabili za testiranje različnih metod ocene zanesljivosti. V petem poglavju predstavimo in opišemo rezultate naših testov. V zaključku pa povzamemo ugotovitve, ter končamo z nekaj predlogi za nadaljnje delo.

Poglavje 2

Metode strojnega učenja za regresijske probleme

Metod strojnega učenja je veliko [12]. Najprej jih delimo po tem za kakšen tip problema so namenjene. Najpogostejša problema v strojnem učenju sta klasifikacija in regresija, obstajajo pa seveda tudi drugi tipi problemov. V tem diplomskem delo smo se omejili na regresijske probleme, saj imamo zanje primerne podatkovne množice ter že izdelane algoritme za ocene točnosti regresijske napovedi.

Učni algoritem dobi na vhodu predznanje in ter množico učnih primerov. Vsak izmed učnih primerov je sestavljen iz atributov ter razreda kateremu pripada. Atributi so neodvisne zvezne ali diskretne spremenljivke, ki opisujejo nek primer, razred pa je odvisna spremenljivka, ki je odvisna od vrednosti neodvisnih spremenljivk. Razred je v primeru klasifikacije diskretna spremenljivka, v primeru regresije pa je zvezna spremenljivka. Naloga učnega algoritma je tako, da na množici učnih primerov ter s pomočjo predznanja preišče prostor hipotez ter vrne zaključno hipotezo. To zaključno hipotezo uporabimo na novih primerih za katere še ne poznamo vrednosti razreda, ter tako napovemo to vrednost razreda.

V tej diplomski nalogi smo uporabili štiri različne algoritme za reševanje regresijskih problemov: regresijska drevesa, naključni gozdovi regresijskih dreves, metoda najbližjih sosedov in metoda podpornih vektorjev. Vsi ti algoritmi so že implementirani v orodju Orange in smo jih kot take uporabili. Algoritme smo uporabili take kot so, brez da bi jih natančno nastavili za čim večjo napovedno točnost.

2.1 Regresijska drevesa

Regresijsko drevo [6] je algoritem pri katerem se s pomočjo učne množice primerov zgradimo drevo. V tem drevesu notranja vozlišča ustrezajo atributom, vejitve ustrezajo podmnožicam vrednosti atributov, listi pa ustrezajo vrednosti razreda. Tako pot od korena drevesa do lista ustreza enemu izmed ßčitvenih pravil. Algoritmi za gradnjo regresijskih dreves glede na oceno informativnosti posameznih atributov izbirajo atribute in ustrezne podmnožice njihovi vrednosti za gradnjo regresijskega drevesa. Pri regresijskih drevesih za dani diskretni atribut A in njegove vrednosti $V, V_1, V_2, ..., V_n$ imamo konjuktivne pogoje oblike $A = V$ ali $A \in \{V_1, ..., V_n\}$. Zvezne atribute pa je potrebno bodisi vnaprej diskretizirati, ali pa so konjuktivni pogoji v pravilih za dani zvezni atribut A in vrednosti V, V_1, V_2 oblike $A > V$ ali $A \leq V$ oziroma $V_1 \geq A > V_2$. Regresijsko drevo tako gradimo dokler nimamo v listih minimalnega števila primerov, ki smo ga vnaprej izbrali. Po končani gradnji drevesa lahko uporabimo še kakšnega izmed algoritmov za obrezovanje drevesa, ki poenostavijo drevo v primeru, da to ne vpliva na njegovo napovedno točnost. Regresijsko drevo uporabimo za uvrščanje novega primera tako da pričnemo v korenu drevesa in opazujemo atribut v njem, glede na možne vejitve in vrednost atributa se sprehodimo do naslednjega vozlišča. Tako napredujemo dokler ne pridemo do lista. Pri klasifikacijskih problemih tako izberemo za napoved razreda tisto vrednost, ki se največkrat pojavi pri primerih v listu. Pri regresijskih problemih pa vzamemo povprečno vrednost primerov v listih.

Regresijska drevesa so zelo dobro implementirana v orodju Orange, najdemo jih v modulu *Orange.classification.tree*.

2.2 Naključni gozdovi regresijskih dreves

Algoritem za gradnjo naključnega gozda regresijskih dreves [7] deluje tako, da izdelamo poljubno število regresijskih dreves. Vendar pa zato, da drevesa niso enaka posamezna regresijska drevesa učimo na različnih učnih množicah. Naj bo N velikost osnovne učne množice, M pa število vseh atributov. Dani parameter m predstavlja število atributov, ki se bodo uporabili pri učenju posameznega drevesa. Učno množico za posamezno regresijsko drevo izdelamo tako da izbiramo iz originalne učne množice primere s ponavljanjem N-krat. Prav tako za vsako posamezno učno množico izberemo m naključnih spremenljivk iz originalne učne množice. Nato na vsaki izmed učnih množic naučimo eno regresijsko drevo. Ko želimo uvrstiti nov primer za katerega ne poznamo vrednosti razreda, ta primer poženemo skozi vsako izmed regresijskih dreves.

Za napoved v primeru regresijskega problema vzamemo povprečje teh vrednosti.

Naključni gozdovi regresijskih dreves so kot ena izmed ansambelskih metod implementirani v orodju Orange; najdemo jih lahko v modulu *Orange.ensemble.forest*.

2.3 Metoda najbližjih sosedov

Algoritem k-najbližjih sosedov (angl. *k-nearest neighbors*) uporablja celotno množico učnih primerov. Ob napovedi novega primera algoritem poišče k najbližjih primerov - sosedov glede na vnaprej definirano razdaljo. Pri klasifikacijskem problemu razred novega primera določimo tako, da izberemo večinski razred med temi k najbližjimi primeri v učni množici. Pri regresijskih problemih pa vrednost razreda določimo kot povprečje vrednosti razredov k najbližjih sosedov.

Uporabili smo implementacijo v modulu *Orange.classification.knn* v orodju Orange. Število najbližjih sosedov, ki jih algoritem uporabi za napoved vrednosti razreda je enako korenu vseh primerov v učni množici.

2.4 Metoda podpornih vektorjev

Metoda podpornih vektorjev [9] s pomočjo linearne algebre in optimizacije poskuša čim bolje napovedovati vrednost razreda. To stori tako da definira rob okoli regresijske hiperravnine, znotraj katerega je odvisna spremenljivka za vse primere eksaktno napovedana, zunaj njega pa so podporni vektorji, ki določajo potek hiperravnine. Ker metoda podpornih vektorjev uporablja implicitno transformacijo atributnega prostora s pomočjo jedrnih funkcij, ni potrebno uporabljati nelinearnih funkcij za predstavitev hiperravnine. Seveda želimo tukaj rob okoli hiperravnine minimizirati, hkrati pa želimo minimizirati tudi napako na učnih primerih. Tako je potrebno optimizirati kriterijsko funkcijo, ki upošteva napake napovedi regresijske spremenljivke na učnih primerih ter kompleksnost funkcije.

Implementacija metode podpornih vektorjev se v orodju Orange nahaja v modulu *Orange.classification.svm*. Pri testiranju smo uporabili vse vnaprej nastavljene parametre.

Poglavje 3

Metode ocene zanesljivosti

V tem poglavju predstavimo različne metode za oceno zanesljivosti napovedi, kot so te predstavljene v [3]. Metodo razdalje po Mahalanobisu smo povzeli kot primer referenčne metode, ki jo trenutno uporabljajo v farmacevtskem podjetju Astra Zeneca za razvoj novih molekul. Te metode so uporabne pri ocenjevanju napovedne točnosti kakršnega koli učnega algoritma. V nasprotju s temi metodami pa predlagamo metodo variance napovedi v naključnem gozdu, ki ocenjuje točnost napovedi samo v naključnem gozdu regresijskih dreves.

3.1 Analiza občutljivosti

Analiza občutljivosti (angl. *sensitivity analysis*) [2] nam omogoča analizo lokalnih posebnosti učnih algoritmov. Cilj te metode je ugotoviti koliko so rezultati algoritma povezani z majhnimi spremembami v učni množici. Ta metoda vsebuje dve med seboj podobni meri za oceno zanesljivosti napovedi SAvar (angl. *Sensitivity analysis - variance*) ter SAbias (angl. *Sensitivity analysis - bias*).

Za ocene zanesljivosti izbranega primera najprej ta primer uporabimo na algoritmu ter dobimo napoved razreda, ki jo označimo K. Da ocenimo zanesljivost, v učno množico dodamo primer, ki ima vse atribute enake kot izbrani primer, vendar pa mu vrednost razreda nastavimo na $K + \epsilon * (l_{max} - l_{min})$, kjer je ϵ parameter občutljivosti, l_{max} in l_{min} pa predstavljata zgornjo in spodnjo mejo vrednosti razreda na učni množici primerov. Na novi učni množici ponovno učimo isti algoritem z istimi parametri ter napovemo isti primer in dobimo novo napoved vrednosti razreda K_ϵ.

Po izračunih napovedi razreda z različnimi vrednostmi parametra ϵ (v tej diplomski nalogi smo vedno uporabili $\epsilon \in E, E = \{0.01, 0.1, 0.5, 1.0, 2.0\}$). Te

napovedi združimo v dve različni oceni. Z uporabo več vrednosti ϵ tako dobimo večji pregled na tem, kaj se dogaja z učnim algoritmov v okolici izbranega primera. Dve različni oceni za točnost napovedi sta definirani [2] kot:

$$SAvar = \frac{1}{|E|} \sum_{\epsilon \in E} (K_\epsilon - K_{-\epsilon}) \tag{3.1}$$

ter

$$SAbias = \frac{1}{2|E|} \sum_{\epsilon \in E} (K_\epsilon - K) + (K_{-\epsilon} - K) \tag{3.2}$$

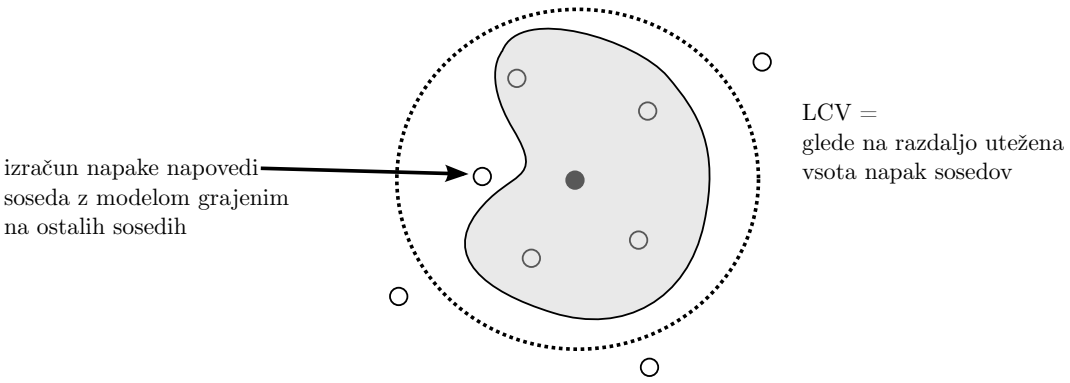
kjer so vsi parametri enaki kot je opisano v zgornjih odstavkih.

3.2 Lokalno prečno preverjanje

Ocena zanesljivosti LCV (angl. *local cross validation*) temelji na postopku prečnega preverjanja. Ilustracijo delovanja lahko vidimo na sliki 3.1. Za nov primer brez vrednosti razreda izračunamo zanesljivost po naslednji metodi:

1. poišči množico k najbližjih sosedov $N = \{(x_1, C_1), ..., (x_k, C_k)\}$ primera x
2. za vsak $(x_i, C_i) \in N$
   1. zgradimo model M na podatkih $N \backslash (x_i, C_i)$
   2. z modelom M za (x_i, C_i) izračunaj napoved K_i
   3. za (x_i, C_i) izračunaj napako $E_i = |C_i - K_i|$
3. $LCV(x) = \frac{\sum_{(x_i, C_i) \in N} d(x_i, x) * E_i}{\sum_{(x_i, C_i) \in N} d(x_i, x)}$

Tako na množici najbližjih sosedov novega primera s pomočjo postopka prečnega preverjanja izračunamo napako, ki jo na model napove s pomočjo teh sosedov. Za izračun ocene napake novega primera tako uporabimo vse napake najbližjih sosedov, ki pa jih utežimo z razdaljo do novega primera $d(x_i, x)$. Za to lahko uporabimo katerokoli od razdalj, v naših poskusih smo uporabili evklidsko razdaljo.



Slika 3.1: Ilustracija delovanja ocene zanesljivosti LCV.

3.3 Varianca modela bagging

Izraz bagging pride iz izraza "boostrap agreggating"[5]. Bootstrap je splošen princip razmnoževanja učnih primerov, kadar jih nimamo dovolj za učenje, pri baggingu pa ustvarimo serijo različnih učnih množic. Če ima učna množica n primerov, potem vsakič n krat naključno izberemo primer iz učne množice z ponavljanjem. Torej se lahko učni primer v tako ustvarjeni množici lahko ponovi večkrat, nekaterih primerov iz učne množice pa tako ustvarjena množica sploh ne vsebuje. Na tako ustvarjenih učnih množicah naučimo k naših osnovnih modelov.

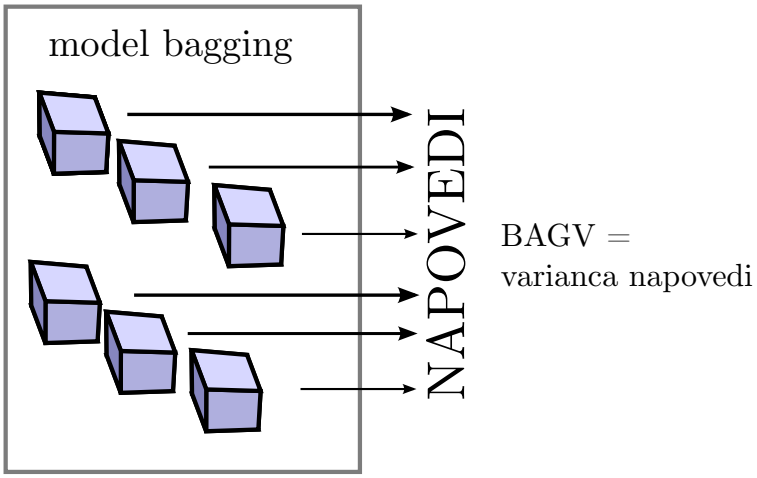
Za oceno napovedne točnosti uporabimo podoben koncept kot ga uporablja model bagging. Najprej izdelamo k podatkovnih množic po zgornji metodi, ter naučimo naše modele, tako dobimo k različnih modelov. Za primer katerega točnost nas zanima nato vsi modeli povedo svoje napovedi $K_i, i = 1, ..., k$. Točnost primera pa ocenimo tako da povprečimo te napovedi:

$$K = \frac{\sum_{i=1}^{k} K_i}{k} \tag{3.3}$$

ocena zanesljivosti BAGV pa je nato definirana kot varianca napovedi:

$$BAGV = \frac{1}{k} \sum_{i=1}^{k} (K_i - K)^2 \tag{3.4}$$

Približno ilustracijo delovanja te ocene prikazuje slika 3.2. V naših poskusih smo za ocene točnosti BAGV uporabili $k = 50$ modelov. Ta parameter je v



Slika 3.2: Ilustracija delovanja ocene zanesljivosti BAGV.

naši implementaciji za programsko okolje Orange seveda mogoče nastaviti tudi na druge vrednosti.

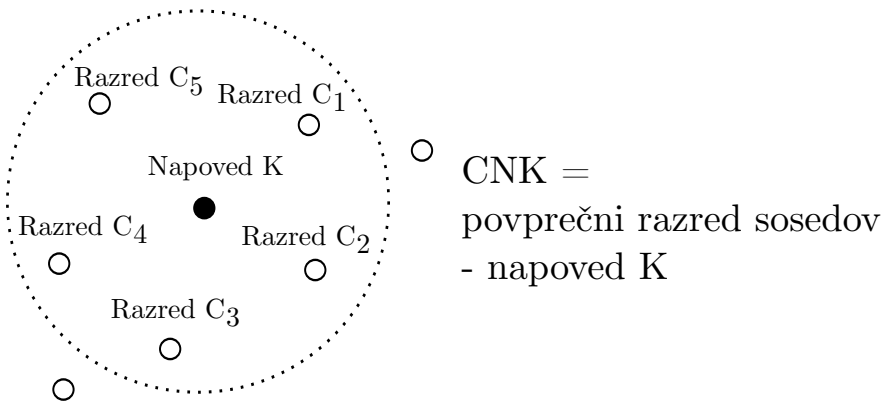
3.4 Lokalno modeliranje napovedne napake

Ocena zanesljivosti CNK (angl. *C neighbors minus K*) je še ena izmed ocen, ki deluje na okolici primera kateremu želimo oceniti zanesljivost. Na učni množici primerov najprej poiščemo množico k najbližjih sosedov $N = \{(x_1, C_1), ..., (x_k, C_k)\}$. Ocena zanesljivosti CNK je enaka razliki med povprečjem razrednih vrednosti najbližjih sosedov ter napovedjo K (z naučenim modelom na celotni učni množici):

$$CNK = \frac{\sum_{i=1}^{k} C_i}{k} - K \tag{3.5}$$

kjer je k število najbližjih sosedov, ki jih uporabimo, C_i pomeni vrednosti razreda teh sosedov in K pomeni napoved za izbrani primer.

Pomembna lastnost te ocene zanesljivosti je tudi, da je predznačena. Torej nam lahko pove ne samo absolutno velikost napake, ampak tudi ali je napovedana vrednost prevelika ali premajhna. Za ta namen, ter za primerjavo med drugimi ocenami zanesljivosti, sta tako definirani dve oceni zanesljivosti na podlagi CNK. CNK-a je samo absolutna vrednost ocene CNK, ter nam pove samo velikost napake. CNK-s pa je enaka CNK.



Slika 3.3: Ilustracija delovanja ocene zanesljivosti CNK.

3.5 Kombinacija BAGV in CNK

V [3] na podlagi eksperimentov definirajo še eno ocene, ki kaže boljše rezultate od zgoraj opisanih. To je linearna kombinacija ocen BAGV in CNK ter je natačneje definirana takole:

$$BVCK(x) = \frac{BAGV(x) + CNK(x)}{2} \tag{3.6}$$

kjer x označuje izbrani primer katerega ocene zanesljivosti nas zanima, $BAGV(x)$ ter $CNK(x)$ pa sta oceni zanesljivosti za ta primer.

3.6 Notranje prečno preverjanje

Notranje prečno preverjanje (angl. *internal cross-validation*) je prilagoditev metode prečnega preverjanja, pri kateri na določeni podatkovni množici izberemo oceno zanesljivosti, ki čim bolje ocenjuje napako.

Notranje prečno preverjanje deluje tako, da najprej razdeli učno množico na n enako velikih podmnožic, na enak način kot pri običajnem prečnem preverjanju. V n korakih je nato vsakič ena izmed podmnožic uporabljena kot testna množica, preostali primeri pa kot učna množica. Vsakič naš model naučimo na učni množici ter napovemo vrednosti razreda za testno množico, ter izračunamo vse ocene zanesljivosti, ki jih želimo uporabiti. Po n korakih imamo za vsak primer iz naše podatkovne množice izračunano napoved, ter vse ocene zanesljivosti. S pomočjo napovedi in originalnih vrednosti razreda

izračunamo napako, ki jo je model naredil. Za vsako izmed ocen zanesljivosti pa nato izračunamo korelacijski koeficient med izračunanimi napakami in vrednostmi te ocene. Nato izberemo oceno zanesljivosti, ki se je najbolje izkazala ter jo nato uporabimo.

Postopek opišemo s pseudokodo na naslednji način:

1. Razdelimo podatkovno množico na podmnožice $\{S_1, ..., S_n\}$
2. za ucna od 1 do n
    1. nauči model na $S \backslash S_{ucna}$
    2. za vsak $(x_i, C_i) \in S_{ucna}$
        1. izračunaj napoved za (x_i, C_i)
        2. izračunaj vse ocene zanesljivosti za (x_i, C_i)
3. za vse ocene zanesljivosti izračunaj korelacijski koeficient ter vrni najboljšega

Tako s pomočjo postopka prečnega preverjanja dobimo oceno zanesljivosti, ki se na tej podatkovni množici najboljše obnese. To oceno lahko sedaj uporabimo na novih primerih za katere ne vemo kakšno vrednost razreda imajo.

3.7 Mahalanobisova razdalja

Ocena zanesljivosti Mahalanobisova razdalja je nekakšen približek gostoti prostora okoli izbranega primera. Definirana je kot vsoto razdalj do n najbližjih sosedov, ko za razdaljo uporabimo Mahalanobisovo razdaljo.

Mahalanobisova razdalja med dvema primeroma je definirana kot:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \tag{3.7}$$

kjer sta $\vec{x}$ in $\vec{y}$ vektorska zapisa primerov, S pa je kovariančna matrika celotne učne množice zapisane v matrični obliki. Kadar je kovariančna matrika enaka identiteti takrat je Mahalanobisova razdalja enaka evklidski.

Ocena zanesljivosti Mahalanobisova razdalja tako napoveduje večjo zanesljivost kadar so primeri v okolici blizu izbranemu primeru. Manjšo zanesljivost pa napoveduje, kadar je primer bolj osamljen v prostoru primerov.

3.8 Varianca napovedi v naključnem gozdu

Do sedaj smo opisovali le ocene zanesljivosti, ki delujejo na vseh modelih. Te ocene zanesljivosti uporabijo modele kot črne škatle, ne zanima jih kaj se dogaja znotraj teh modelov. Nekatere ocene, recimo Mahalanobisova razdalja, celo ne uporabi modela, ampak uporabi samo podatkovno množico, da izračuna oceno zanesljivosti. Ocena zanesljivosti ki pa jo predlagamo v tem primeru pa ne deluje na vseh modelih ampak je posebna za naključni gozd.

V naključnem gozdu vsako izmed posameznih regresijskih dreves prispeva h končni napovedi. Zanimalo nas je, ali lahko razpršenost, varianca teh napovedi kaj pove o zanesljivosti napovedi v naključnem gozdu. Zato smo predlagali novo oceno za izračun zanesljivost. Najprej izračunamo povprečje napovedi posameznih regresijskih dreves:

$$K = \frac{\sum_{i=1}^{m} K_i}{m} \tag{3.8}$$

ocena zanesljivosti RFV pa nato definiramo kot varianco napovedi posameznih regresijskih dreves:

$$RFV = \frac{1}{m}\sum_{i=1}^{m}(K_i - K)^2 \tag{3.9}$$

kjer je m število regresijskih dreves v naključnem gozdu, K_i pa napoved i-tega regresijskega drevesa. Delovanje prikazuje tudi slika **??**.

V naključnem gozdu v splošnem vedno pričakujemo veliko varianco napovedi, da se lahko le-te dobro povprečijo, ter se tako izničijo napačne napovedi. Vendarle pa z uporabo te ocene zanesljivosti pričakujemo, da bodo primeri pri katerih bo varianca največja imeli tudi največjo absolutno napako.

3.9 Analiza časovnih zahtevnosti

Opravili smo tudi analizo časovnih zahtevnosti posameznih metod za oceno zanesljivosti. Časovne zahtevnosti so prikazane v tabeli 3.1.

Zaradi uporabe različnih metod strojnega učenja smo se odločili za skupno notacijo. Tako $\mathcal{L}$ pomeni časovno zahtevnost učenja, ter $\mathcal{P}$ časovno zahtevnost napovedi enega primera s pomočjo te metode strojnega učenja. Število primerov v podatkovni množici je označeno z n, število atributov pa z m. Pri analizi občutljivosti imamo še poseben parameter E, ki je množica možnih vrednosti ϵ. Pri metodi variance modela bagging parameter k pomeni število

Metoda	Časovna zahtevnost	
	učenje	ocena zanesljivosti
SAbias & SAvar	$O(n)$	$O(\|E\|(\mathcal{L}+\mathcal{P}))$
BAGV	$O(m(n+\mathcal{L}))$	$O(m\mathcal{P})$
LCV	$O(1)$	$O(n+k(\mathcal{L}+\mathcal{P}))$
CNK	$O(1)$	$O(n+k)$
BVCK	$O(m(n+\mathcal{L}))$	$O(m\mathcal{P}+n+k)$
Mahal	$O(nm^2+m^3)$	$O(m^2)$

Tabela 3.1: Časovne zahtevnosti metod za oceno zanesljivosti.

uporabljenih modelov, pri ostalih pa k pomeni število najbližjih sosedov, ki jih želimo uporabiti v naših izračunih.

Poglavje 4

Metodologija testiranja

4.1 Testne podatkovne množice

Za testiranje učinkovitosti delovanja ocen zanesljivosti smo uporabili različne podatkovne množice. Najprej smo izdelali dve podatkovni množici ki temeljita na matematičnem modeliranju nekega prostora primerov. Zatem pa smo uporabili tudi več različnih podatkovnih množic, ki vsebujejo podatke o kvantitativnih razmerjih med strukturo in delovanjem molekul.

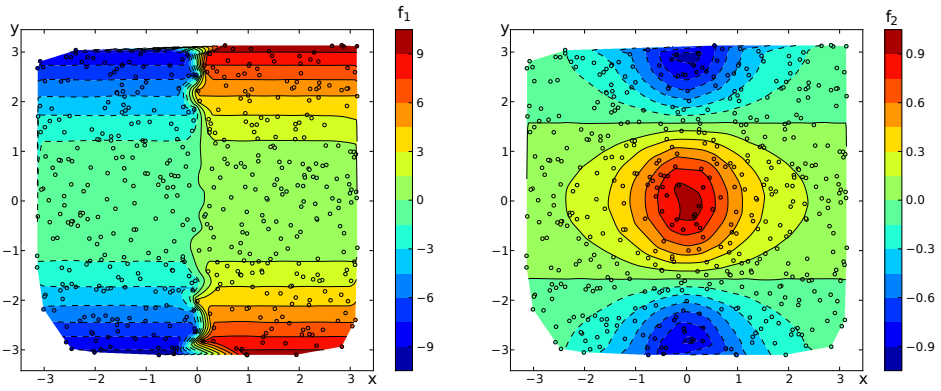
4.1.1 Umetno izdelane podatkovne množice

Vse naše metode za oceno zanesljivosti smo uporabili na dveh umetno izdelanih podatkovnih množicah. Obe podatkovni množici imata vnaprej definirano povezavo med atributi in razredno spremenljivko. Uporaba takšne metode se je že prej izkazala za uspešno [8]. Prostor atributov je v našem primeru dvodimenzionalen, ter enak za obe podatkovni množici. Razteza se od $-\pi$ do π v obeh dimenzijah. Ustvarjeni primeru so enakomerno naključno izbrani po tem prostoru. Za uporabo dvodimenzionalnega prostora smo se odločili, ker je potem takšen prostor enostavno prikazati. To lahko storimo z enostavnim prikazom višin na grafu.

Za funkciji ki definirata razmerje med atributoma in razredom smo uporabili naslednji dve:

$$f_1(x, y) = y^2 \frac{|x|}{x} \tag{4.1}$$

$$f_2(x, y) = \frac{\cos y}{1 + x^2} \tag{4.2}$$



Slika 4.1: Grafična predstavitev podatkovne množice ustvarjene s funkcijo f_1 in f_2.

kjer sta x in y spremenljivki v prostoru atributov. Za vsako izmed podatkovnih množic je povezava med atributi in razredov izračunana v vsaki izmed 500 naključno izbranih točk. Obe podatkovni množici tako vsebujeta 500 dvodimenzionalnih primerov. Vizualno predstavitev teh dveh podatkovnih množic predstavlja slika 4.1.

4.1.2 Podatkovne množice s podatki o kvantitativnih razmerjih med strukturo in delovanjem molekul

Za testiranje delovanja ocen zanesljivosti na regresijskih podatkih smo uporabili tudi resnične podatkovne množice. Te podatkovne množice temeljijo na podatkih o kvantitativnih razmerjih med strukturo in delovanjem molekul (angl. *Quantitative Structure-Activity Relationship*). Pri teh podatkih gre za iskanje povezav med strukturo kemijske substance in njenim delovanjem v organizmu ter kvantifikacijo teh povezav, kar v farmakologiji in sorodnih področjih omogoča napovedovanje učinkov neke spojine na osnovi njene kemijske zgradbe. Poznavanje teh povezav je zlasti pomembno pri načrtovanju novih zdravilnih učinkovin.

Podatkovne množice so izdelali v podjetju Astra Zeneca, ter nam jih posredovali. Vse podatkovne množice so izdelane na podlagi javnih podatkov v bazi PubChem, dostopni na http://pubchem.ncbi.nlm.nih.gov/. Izdelane so na podlagi desetih bioloških analiz, ki merijo različne vplive malih molekul na žive organizme. Kateri vplivi so izmerjeni v kateri izmed analiz je prikazano v

biološki test	merjeni vpliv
1239	povečanje genske ekspresije gena NF-kB
1479	preprečevanje interakcije med TR in SRC2
1511	varovanje gena hERG pred učinkovinami, ki povzročajo srčno aritmijo
2156	zmanjšanje genske ekspresije gena KCNQ2
2796	povečanje genske ekspresije gena AHR
631	povečanje vezave SRC-1 na PPARgamma
639	povečanje vezanja peptida SRC1 na vezno domeno estrogenskega receptorja
677	analiza modulatorjev na receptor M1
862	zmanjšanje genske ekspresije gena STAT3
932	povečanje ekspresije gena STAT1

Tabela 4.1: Opis merjenih vplivov malih molekul pri posameznih bioloških testih.

tabeli 4.1. Atributi so ustvarjeni s pomočjo orodja RDKit (http://rdkit.org/), ki izračuna različne lastnosti molekul, ter krožnega odtisa izbrane molekule [11]. Primer molekule in nekaj njenih atributov je predstavljen na sliki 4.6.

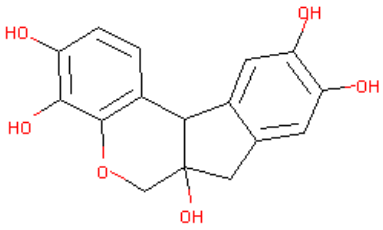
V našem naboru podatkovnih množic smo uporabili različno velike podatkovne množice: od majhnih z okoli 100 primeri, srednjih s 500 primeri do velikih, ki vsebujejo vse javno znane podatke, z številom primerov med 1400 in 16000. Podatkovne množice se razlikujejo tudi po številu atributov. Število atributov se giblje med 170 in 2000. Bolj natančne informacije o podatkovnih množicah so opisane v dodatku A.

4.2 Postopek testiranja

Različne metode ocen zanesljivosti je potrebno med seboj tudi primerjati ter ovrednotiti rezultate. Za ta postopek smo izbrali 10-kratno prečno preverjanje na podatkovni množici. Prečno preverjanje je ena izmed najbolj popularnih metod za analizo uspešnosti različnih modelov in njihovih parametrov, zato smo to metodo uporabili tudi v našem primeru.

Po tem, ko s prečnim preverjanjem dobimo napovedi vrednosti razreda za vse primere iz podatkovne množice, lahko za naš model izračunamo napovedno napako. Napako izračunamo za vsak primer posebej, tako da odštejemo napoved od resnične vrednosti razreda. S Pearsonovim koeficientom korelacije med

Hematoxylin - $C_{16}H_{14}O_6$



Molska masa	330.28
Število darovalcev vodikovih vezi	5
Število prejemnikov vodikovih vezi	6
Število dušikovih in kisikovih atomov	6
Število NH in OH	5
Število valenčnih elektronov	114
Število vrtljivih vezi	0

Slika 4.2: Molekula Hematoxylina in nekaj njenih strukturnih lastnosti.

napako in eno izmed ocen zanesljivosti dobimo mero, ki nam pove koliko je naša ocena dobra. Pove nam tudi koliko je neka ocena boljša od druge ocene. Nadalje pa lahko še izračunamo koliko statistično pomembna je ta ocena.

Pri uporabi Pearsonovega koeficienta korelacije pa moramo paziti, da ocene ki vsebujejo le absolutno vrednost primerjamo z absolutno vrednostjo napake. Prav tako pa predznačene ocene primerjamo z predznačeno. Tako zmeraj dobimo pravilne korelacije ter se izognemo napakam.

Za testiranje vseh metod in njihovo implementacijo smo uporabili programski paket Orange [10]. Meritve smo izvajali na gruči računalnikov v Laboratoriju za bioinformatiko s pomočjo okolja xgrid, ki razdeljuje posamezne naloge med prostimi računalniki.

4.2.1 Pearsonov koeficient korelacije

Pearsonov koeficient korelacije je matematična in statistična številska mera, ki predstavlja velikost linearne povezanosti dveh spremenljivk (X in Y), merjenih na istem predmetu preučevanja. Koeficient je definiran kot vsota vseh produktov standardnih odklonov obeh vrednosti v razmerju s stopnjami prostosti oziroma kot razmerje med kovarianco in produktom obeh standardnih odklonov:

$$r_{xy} = \frac{\sum z_x z_y}{N-1} \tag{4.3}$$

kjer je z_x z-vrednost spremenljivke X, z_y pa z-vrednost spremenljivke Y, N pa število statistični enot. Ali:

$$r_{xy} = \frac{C_{xy}}{\sigma_x \sigma_y} \tag{4.4}$$

kjer je C_{xy} kovarianca, σ_x standardni odklon spremenljivke X, σ_y standardni odklon spremenljivke Y.

Vrednost Pearsonovega koeficienta korelacije se vedno nahaja med -1 in 1. Tako vrednost 1 predstavlja popolno pozitivno povezanost spremenljivk, -1 pa predstavlja popolno negativno povezanost spremenljivk. Pearsonov koeficient 0 označuje ničelni vpliv ene spremenljivke na drugo.

Izračunali smo Pearsonov koeficient korelacije, sedaj pa nas še zanima kdaj se dovolj razlikuje od 0, da lahko statistično zagotovimo, da obstaja resnično razmerje med dvema spremenljivkama. To storimo s testom pomembnosti Pearsonovega koeficienta korelacije:

$$t = \frac{r_{xy}}{\sqrt{\frac{1-r_{xy}^2}{N-2}}} \tag{4.5}$$

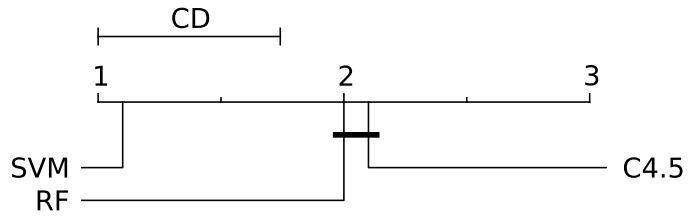
kjer je r_{xy} Pearsonov koeficint korelacije med spremenljivkama X in Y, N pa je število statističnih enot. Verjetnostna porazdelitev t je porazdeljena približno enako kot Studentova t porazdelitev s prostorsko stopnjo $N - 2$. Tako lahko preverimo nično hipotezo za izračunano vrednost r_{xy} in preverimo s kakšno verjetnostjo je koeficient korelacije r_{xy} statistično različen od koeficienta 0.

4.2.2 Graf rangov s kritično razdaljo

Graf rangov s kritično razdaljo je ena ena izmed možnosti vizualizacije kako primerjamo uspešnost različnih algoritmov. Omogoča nam da primerjamo več različnih metod na več podatkovnih množicah, ter to vizualiziramo na eni sliki.

Po izračunu rezultatov na več različnih podatkovnih množicah, za vsako množico razvrstimo metode po uvrstitvi, od prve do zanje. Nato za vsako izmed metod izračunamo povprečno uvrstitev. Te uvrstitve nato prikažemo na grafu kot je vidno na sliki 4.3. Za ugotavljanje značilnosti razlik uspešnosti uporabimo Nemenyijev test iz enačbe 4.6. Nemenyijev test nam pove kakšna mora biti razlika v povprečnem rangu dveh metod, da se metodi značilno razlikujeta.

Nemenyijev test potrdi značilnost razlik med dvema algoritmoma j_1 in j_2, če je razlika med povprečnima rangoma večja ali enaka kritični razdalji CD:



Slika 4.3: Primer grafa rangov s kritično razdaljo prikazuje uspešnost metod SVM, RF ter C4.5. Metoda SVM je značilno uspešnejša od drugih dveh metod.

$$|R_{j_1} - R_{j_2}| \geq CD = q_\alpha \sqrt{\frac{k(k+1)}{6D}} \tag{4.6}$$

kjer sta R_{j_1} in R_{j_2} povprečna ranga prvega in drugega algoritma, q_α pa je kritična vrednost testa Nemenyijev za stopnjo zaupanja α, k je število algoritmov, ki jih primerjamo, D pa število podatkovnih množic.

4.2.3 Testiranje notranjega prečnega preverjanja

Za splošno testiranje uporabljamo prečno preverjanje ter Pearsonov koeficient korelacije. Pri notranjem prečnem preverjanju v navezi z 10-kratnim zunanjim prečnim preverjanjem pa nastane problem, saj se lahko za različne dele podatkovne množice uporabijo različne ocene zanesljivosti. Različne ocene zanesljivosti pa med seboj niso primerljive. Uporabili smo v za ta namen predlagan postopek [4].

Postopek deluje tako, da podatkovno množico razdelimo na 10 enakih delov. Na vsakemu izmed teh desetih delov poženemo običajen postopek prečnega preverjanja ter izračunamo Pearsonov koeficient korelacije za vsako izmed uporabljenih metod ocene zanesljivosti. Tako imamo za vsako izmed metod ocene zanesljivosti deset koeficientov korelacije. Za vsako metodo nato izračunamo povprečje ter preverimo katera metoda ima najvišji povprečni koeficient korelacije. Za to metodo potem pogledamo v tabelo rezultatov prejšnjih testiranj kakšen Pearsonov koeficient korelacije je dosegla ter preverimo ali je statistično značilen. Ta rezultat potem vzamemo kot rezultat metode notranjega prečnega preverjanja.

Poglavje 5

Rezultati in diskusija

Vse splošne metode za oceno zanesljivosti smo pognali na izbranih metodah strojnega učenja. Uporabili smo le podatkovne množice srednje velikosti, to je 500 primerov v posamezni podatkovni množici. Že izračuni na množicah srednje velikosti so trajali več dni. Ocenjujemo da bi poganjanje metod za oceno zanesljivosti na velikih podatkovnih množicah trajalo tudi več mesecev, zato tega nismo naredili.

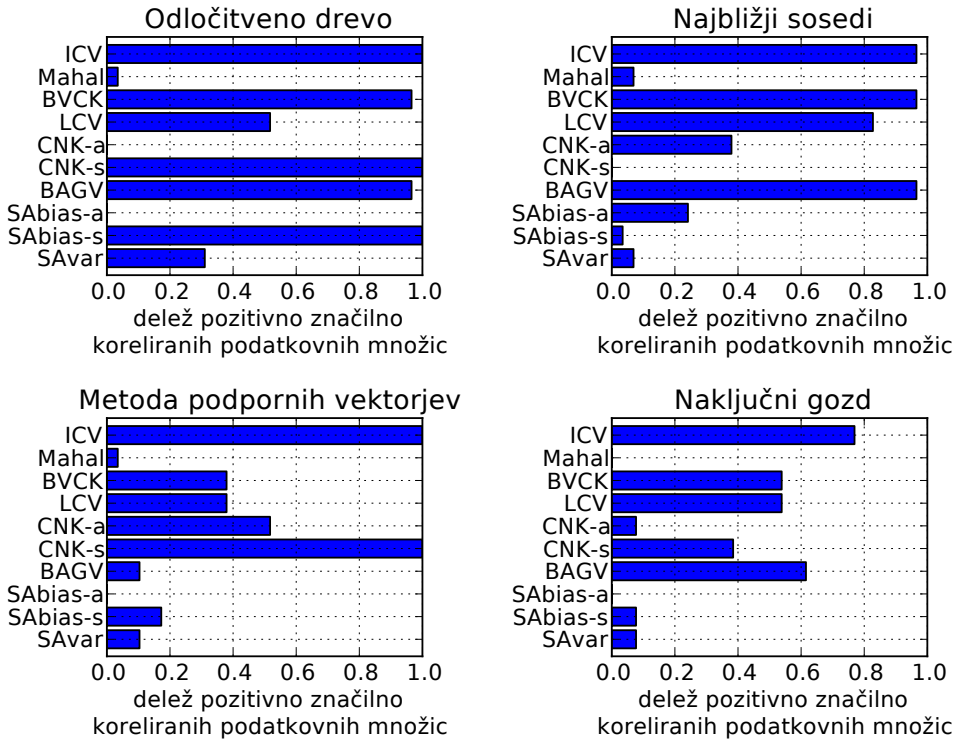
Rezultati so grafično prikazani na sliki 5.1. Najboljše se izkaže metoda ICV, saj za vsako izmed metod strojnega učenja zmaga med vsemi metodami. V povezavi z regresijskimi drevesi in metodo najbližjih sosedov se dobro obneseta metodi BVCK in BAGV. SAbias-s kaže dobre rezultate pri regresijskih drevesih, prav tako kot tudi CNK-s. Obe metodi ocenjujeta ne samo velikost napake, ampak tudi smer napake. S pomočjo teh dveh metod je najverjetneje mogoče tudi nekako popravljati napovedi ter tako doseči boljše rezultate.

5.1 Primerjava metod za oceno zanesljivosti

Graf rangov s kritično razdaljo smo uporabili, da bi primerjali metode za oceno zanesljivosti med seboj. Najprej si na sliki 5.2 oglejmo primerjavo med metodami, ko jih uporabimo skupaj z metodo najbližjih sosedov. Najbolje se po pričakovanjih izkaže metoda notranjega prečnega preverjanja. Za njo sta potem metodi LCV in BAGV.

Ob uporabi regresijskih dreves se poleg ICV izkažeta tudi metodi CNK-s in SAbias-S, kot vidimo na sliki 5.2. To nakazuje na to, da imajo regresijska drevesa še dovolj prostora za izboljšanje. V splošnem na sliki 5.1 vidimo, da pri regresijskih drevesih deluje največ metod za ocene zanesljivosti.

Za metodo podpornih vektorjev najbolje deluje metoda CNK-s. Na sliki 5.2

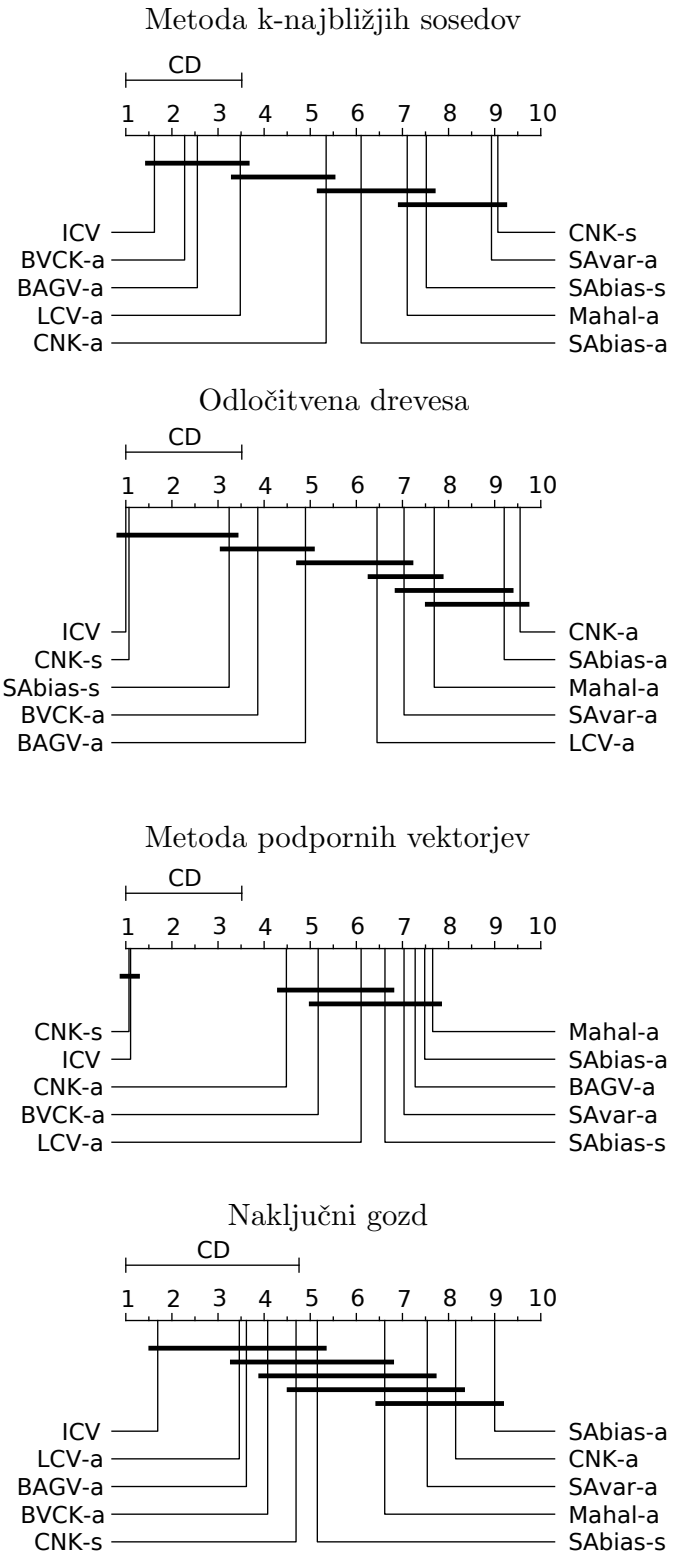


Slika 5.1: Deleži podatkovnih množic, kjer je posamezna metoda dosegla pozitivno značilno korelacijo s stopnjo zaupanja $\alpha \leq 0.05$.

vidimo, da je po uspešnosti zelo blizu ICV. Pomembno pa je tudi opaziti da sta obe metodi statistično značilno boljši od uporabe ostalih metod za oceno zanesljivosti na metodi podpornih vektorjev.

Metoda naključnega gozda je časovno zelo požrešna metoda. Na večji podatkovnih množicah porabi v kombinaciji z metodami za oceno zanesljivosti enostavno preveč časa. Zato smo jo testirali na manjšem naboru podatkov. Rezultati na sliki 5.2 kažejo na to, da je najbolj uspešna metoda spet notranje prečno preverjanje. Vendar ima v primeru naključnih gozdov nižji povprečni rang kot drugje. Dobro se izkažeta tudi metodi BAGV ter LCV.

V splošnem je najboljša metoda notranjega prečnega preverjanja. Problem te metode se pojavi pri večjih podatkovnih množicah, saj za izračune porabi ogromno časa. Najprej na celotni učni množici s prečnim preverjanjem izračuna vse ostale metode za oceno zanesljivosti, nato pa na celotni



Slika 5.2: Grafi rangov s kritično razdaljo prikazujejo za različne metode strojnega učenja kako se metode za oceno zanesljivosti primerjajo med seboj.

učni množici uporabi najboljšo metodo ter izračuna zanesljivosti za testno množico. Na podatkovnih množicah z več kot 1000 primeri postane to skoraj nemogoče. Zato je potrebno pohitriti te algoritme za oceno zanesljivosti ali pa v notranjem prečnem preverjanju ne poskusiti vseh metod ampak samo nekatere.

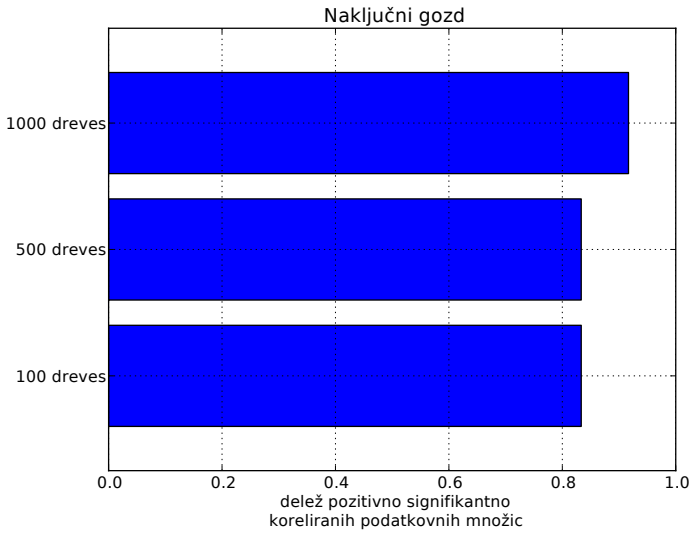
Rezultate uspešnosti različni metod smo primerjali z rezultati objavljenimi v [3, 4]. V teh dveh delih so bile uporabljene različne podatkovne množice, ki so dosegljive v UCI repozitoriju za strojno učenje [1] ter arhivu StatLib Datasets (http://lib.stat.cmu.edu/). Te podatkovne množice zajemajo podatke različnih področij, zato se razlikujejo od našega nabora podatkov, ki zajema točno določeno področje. Tako lahko vidimo če je delovanje metod za oceno zanesljivosti odvisno od domene podatkov.

Pri regresijskih drevesih se v obeh delih najbolje obnesejo metode notranjega prečnega preverjanja, SAbias-s in CNK-s, ostale metode pa pokažejo slabše rezultate. Metoda notranjega prečnega preverjanja se v obeh primerih najboljše obnese v navezi z metodo podpornih vektorjev. Zanimivo pa je, da v naših poskusih metoda CNK-s kaže dobre rezultate v [3, 4] pa se uvršča med najslabše metode. Metoda LCV, ki je v gornjih člankih takoj za metodo notranjega prečnega preverjanja v naših poskusih doseže slabše rezultate. Ob uporabi naključnega gozda se v gornjih člankih najbolje obnese metoda LCV, kmalu za njo pa metoda notranjega prečnega preverjanja, ter BAGV in BVCK. V naših poskusih se najbolje obnese metoda notranjega prečnega preverjanja, vendar z večjim deležem pozitivno značilno koreliranih podatkovnih množic. Sledijo pa jih metode BAGV, LCV ter BVCK, torej zelo podobno kot v [3, 4].

5.2 Varianca napovedi v naključnem gozdu

Farmacevtsko podjetje Astra Zeneca trenutno v njihovem raziskovalnem procesu uporablja za reševanje regresijskih problemov naključne gozdove. V kombinaciji z naključnimi gozdovi pa za oceno zanesljivosti Mahalanobisovo razdaljo. V tem delu smo se osredotočili na izboljšavo te ocene. Iz slik 5.2 in 5.1 je razvidno, da nekatere druge ocene v povezavi z naključnimi gozdovi delujejo dosti boljše kot Mahalanobisova razdalja. Vendar pa Astra Zeneca potrebuje poganjati te ocene na podatkovnih množicah velikosti večje kot 10000, zato te metode ne pridejo v poštev.

V ta namen smo uporabili varianco napovedi v naključnem gozdu. Ta metoda je tako hitra kot je hitra naša implementacija naključnega gozda, saj ne naredi nič dodatnega dela, razen da iz napovedi izračuna še njeno varianco.



Slika 5.3: Graf prikazuje na kolikšnem deležu podatkovnih množic je metoda variance v naključnem gozdu dosegla pozitivno značilno korelacijo za $\alpha \leq 0.05$

Na podatkovnih množicah srednje velikosti smo pognali to metodo ter dosegli rezultate v sliki 5.3. Dobri rezultati na manjšem številu regresijskih dreves v naključnem gozdu so pokazatelj, da metoda uspešno ocenjuje zanesljivost. Vendar nas je zanimalo kaj se zgodi ko število dreves povečamo. Lahko, da so rezultati pri manjšem številu regresijskih dreves dobri, ker algoritem naključnega gozda zaradi velikega števila primerov in atributov konvergira proti enemu samemu regresijskemu drevesu.

Tako smo poskusili s povečanjem števila dreves v naključnem gozdu. Na sliki 5.3 prikazani rezultati kažejo na to, da s povečanjem števila regresijskih dreves uspešnost naše ocene ne pada. Varianca napovedi v naključnem gozdu je torej dobra ocena zanesljivosti. To pokaže tudi primerjava Pearsonovih koeficientov korelacije na velikih podatkovnih množicah, posebej dobro se izkaže v primerjavi z Mahalanobisovo razdaljo, kar je vidno v tabeli 5.1. Varianca napovedi v naključnem gozdu na izbranih podatkovnih množicah vedno boljše oceni zanesljivost kot Mahalanobisova razdalja.

Iz kombinacije slike 5.1 in slike 5.3 je razvidno, da v primeru naključnih gozdov metoda variance v naključnem gozdu dosega najboljše rezultate med vsemi ocenami za zanesljivost. Boljša je tudi od notranjega prečnega preverjanja, ki se je izkazalo za najboljše med splošnimi metodami.

podatkovna množica	število primerov	RFV	Mahal
1239_RDK.tab	6202	**0.222**	0.015
1479_RDK.tab	1632	**0.234**	-0.031
1511_RDK.tab	3104	**0.365**	0.035
2156_RDK.tab	6814	**0.347**	0.017
2796_RDK.tab	15980	**0.290**	-0.009
631_RDK.tab	1622	**0.212**	0.012
639_RDK.tab	2302	**0.157**	**0.051**
677_RDK.tab	1446	**0.513**	0.026
862_RDK.tab	3448	**0.308**	**0.043**
932_RDK.tab	10268	**0.255**	0.002

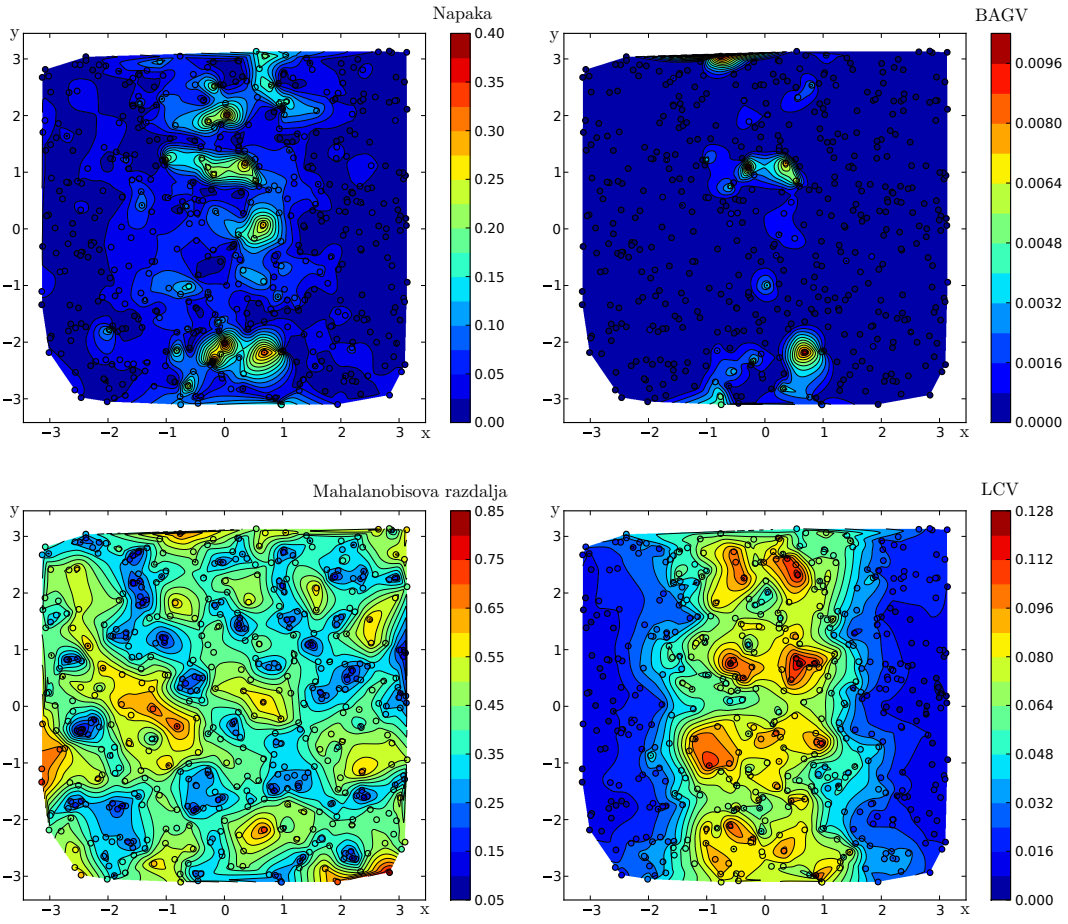
Tabela 5.1: Primerjava Pearsonovih koeficientov korelacije med metodo variance v naključnem gozdu ter Mahalanobisovo razdaljo. Odebeljene vrednosti so statistično značilne z $\alpha \leq 0.05$.

5.3 Rezultati na umetnih podatkovnih množicah

Metode za oceno zanesljivosti smo preizkusili tudi na dveh umetno ustvarjenih podatkovnih množicah. Podrobni rezultati so opisani v dveh tabelah v dodatku B. Iz teh rezultatov lahko zaključimo, da metode ocenijo zanesljivost na umetnih podatkovnih množicah bolje kot na resničnih podatkih o kvantitativnih razmerjih med strukturo in delovanjem molekul. Prav tako ti dve podatkovni množici opisujeta dokaj zvezni funkciji, kar olajša napoved.

Ti dve podatkovni množici sta pomembni saj imata le dva atributa in razred. Tako lahko enostavno vizualno prikažemo kako delujejo metode za oceno zanesljivosti na grafu. Na sliki 5.4 vidimo, da regresijska drevesa v primeru podatkovne množice, ki jo ustvarimo s pomočjo funkcije f_2, naredijo največjo napako v območju vrhov te funkcije. Zalo zanimiv je graf Mahalanobisove razdalje. Na njem vidimo razpršenost ocene, odlično prikaže delovanje Mahalanobisove razdalje, saj za osamljene točke napove večjo napako kot tiste z bližjimi sosedi.

Oceni BAGV in LCV se bolj približata napaki napovedi. Vsaka iz svoje strani, ocena BAGV napove napako na pravilnem območju, vendar za premalo primerov. Ocena LCV pa prav tako napove napako na pravilnem območju, vendar je ta napaka na večjem številu primerov prevelika.



Slika 5.4: Zgornji levi graf prikazuje napako regresijskih dreves na podatkovni množici ustvarjeni s funkcijo f_2. Ostali trije grafi pa prikazujejo tri ocene zanesljivosti (BAGV, Mahalanobisovo razdaljo in LCV.)

Poglavje 6

Zaključek

V diplomskem delu smo preučevali uporabnost metod za oceno zanesljivosti na regresijskih podatkih. V ta namen smo preizkusili že obstoječe metode, tej jih primerjali med seboj na podatkih o kvantitativnih razmerjih med strukturo in delovanjem molekul. Vse metode smo implementirali v programskem paketu Orange. Razvili smo še specializirano metodo variance v naključnem gozdu, ki je učinkovita in hitra metoda, ki oceni zanesljivost napovedi v naključnem gozdu.

6.1 Ugotovitve

Med obstoječimi splošnimi metodami se je za najbolj uspešno izkazala metoda notranjega prečnega preverjanja. V vseh primerih je boljša od ostalih metod, vendar ima eno pomanjkljivost. To je njena počasnost, na večjih podatkovnih množicah bi metoda enostavno porabila preveč časa.

Metoda variance napovedi v naključnem gozdu dosega odlične rezultate. Metoda je uporabna tudi na večjih podatkovnih množicah saj porabi samo toliko časa kot ga že porabi metoda naključnega gozda za izračun napovedi. Metoda je specializirana za uporabo na naključnem gozdu ter je ni mogoče uporabiti v povezavi z drugimi metodami strojnega učenja.

6.2 Nadaljnje delo

V okviru nadaljnjega dela bi bilo potrebno optimizirati hitrost delovanja različnih metod strojnega učenja v programskem paketu Orange. Prav tako pa preučiti kako bi bilo mogoče pohitriti metode za oceno zanesljivosti.

Specializirane metode se očitno dobro obnesejo, zato bi potrebno poskusiti razviti kakšno novo metodo za oceno zanesljivosti modele strojnega učenja, ki tega še nimajo. Novo razvite metode naj uporabljajo notranje informacije teh modelov za izboljšanje ocen zanesljivosti.

Potrebno bi bilo razviti tudi metodo kako iz numerične predstavitve ocene zanesljivosti preidemo na opisno predstavitev. To je predvsem uporabno ko bodo te metode uporabljali ljudje brez obširnega znanja na tem področju.

Dodatek A

Lastnosti podatkovnih množic

Število primerov ter atributov za podatkovne množice, ki smo jih uporabljali pri naših poskusih. Prva številka v imenu pomeni številko biološke analize, druga številka je število primerov. Kateri atributi so vsebovani označujeta FP (krožni odtis molekule) in RDK (opisni atributi v orodju RDKit).

podatkovna množica	število primerov	število atributov
1239_FPRDK.tab	6202	1713
1479_FPRDK.tab	1632	1395
1511_FPRDK.tab	3104	1551
2156_FPRDK.tab	6814	1914
2796_FPRDK.tab	15980	2164
631_FPRDK.tab	1622	1276
639_FPRDK.tab	2302	1252
677_FPRDK.tab	1446	1098
862_FPRDK.tab	3448	1493
932_FPRDK.tab	10268	1827
1239_RDK.tab	6202	176
1479_RDK.tab	1632	176
1511_RDK.tab	3104	176
2156_RDK.tab	6814	176
2796_RDK.tab	10000	176
631_RDK.tab	1622	176
639_RDK.tab	2302	176
677_RDK.tab	1446	176

podatkovna množica	število primerov	število atributov
862_RDK.tab	3448	176
932_RDK.tab	10000	176
1239_FP.tab	6202	1537
1479_FP.tab	1632	1219
1511_FP.tab	3104	1375
2156_FP.tab	6814	1738
2796_FP.tab	15980	1988
631_FP.tab	1622	1100
639_FP.tab	2302	1076
677_FP.tab	1446	922
862_FP.tab	3448	1317
932_FP.tab	10268	1651
1239_500FPRDK.tab	500	847
1479_500FPRDK.tab	500	974
1511_500FPRDK.tab	500	950
2156_500FPRDK.tab	500	949
2796_500FPRDK.tab	500	916
631_500FPRDK.tab	500	910
639_500FPRDK.tab	500	843
677_500FPRDK.tab	500	847
862_500FPRDK.tab	500	890
932_500FPRDK.tab	500	897
1239_500RDK.tab	500	176
1479_500RDK.tab	500	176
1511_500RDK.tab	500	176
2156_500RDK.tab	500	176
2796_500RDK.tab	500	176
631_500RDK.tab	500	176
639_500RDK.tab	500	176
677_500RDK.tab	500	176
862_500RDK.tab	500	176
932_500RDK.tab	500	176
1239_500FP.tab	500	671
1479_500FP.tab	500	798
1511_500FP.tab	500	774
2156_500FP.tab	500	773
2796_500FP.tab	500	740
631_500FP.tab	500	734
639_500FP.tab	500	667

podatkovna množica	število primerov	število atributov
677_500FP.tab	500	671
862_500FP.tab	500	714
932_500FP.tab	500	721
1239_100FPRDK.tab	100	527
1479_100FPRDK.tab	100	580
1511_100FPRDK.tab	100	571
2156_100FPRDK.tab	100	604
2796_100FPRDK.tab	100	573
631_100FPRDK.tab	100	565
639_100FPRDK.tab	100	552
677_100FPRDK.tab	100	574
862_100FPRDK.tab	100	601
932_100FPRDK.tab	100	565
1239_100RDK.tab	100	176
1479_100RDK.tab	100	176
1511_100RDK.tab	100	176
2156_100RDK.tab	100	176
2796_100RDK.tab	100	176
631_100RDK.tab	100	176
639_100RDK.tab	100	176
677_100RDK.tab	100	176
862_100RDK.tab	100	176
932_100RDK.tab	100	176
1239_100FP.tab	100	351
1479_100FP.tab	100	404
1511_100FP.tab	100	395
2156_100FP.tab	100	428
2796_100FP.tab	100	397
631_100FP.tab	100	389
639_100FP.tab	100	376
677_100FP.tab	100	398
862_100FP.tab	100	425
932_100FP.tab	100	389

Dodatek B

Rezultati na umetnih podatkovnih množicah

V tabeli so zapisani Pearsonovi koeficienti korelacije med oceno in napako izračunamo na podatkovni množici ustvarjeni s funkcijo f_1. Koeficienti korelacije, ki so statistično značilni, s stopnjo zaupanja $\alpha \leq 0.05$, so obarvani sivo.

metoda	regresijska drevesa	metoda podpornih vektorjev	naključni gozd	metoda najbližjih sosedov
BAGV	0.013	0.348	0.809	0.560
BVCK	0.144	0.902	0.472	0.525
CNK-a	0.145	0.902	0.450	0.520
CNK-s	0.121	0.941	0.207	0.239
LCV	0.264	0.545	0.310	0.590
Mahal	0.115	0.073	0.204	0.121
SAvar	0.076	0.417	0.028	-0.067
SAbias-a	0.112	0.236	0.067	0.589
SAbias-s	0.196	-0.021	-0.016	0.567

V naslednji tabeli so zapisani Pearsonovi koeficienti korelacije med oceno in napako izračunamo na podatkovni množici ustvarjeni s funkcijo f_2. Koeficienti korelacije, ki so statistično značilni, s stopnjo zaupanja $\alpha \leq 0.05$, so obarvani sivo.

metoda	regresijska drevesa	metoda podpornih vektorjev	naključni gozd	metoda najbližjih sosedov
BAGV	0.453	0.179	0.781	0.289
BVCK	0.809	0.964	0.820	0.291
CNK-a	0.807	0.964	0.818	0.291
CNK-s	0.830	0.980	0.855	0.002
LCV	0.428	0.801	0.535	0.596
Mahal	0.168	0.129	0.170	0.118
SAvar	0.279	0.095	-0.149	-0.029
SAbias-a	0.298	0.001	0.134	0.493
SAbias-s	0.375	0.009	0.019	0.307

Literatura

[1] A. Asuncion, D. J. Newman, "UCI machine learning repository,"2007.

[2] Z. Bosnić, I. Kononenko, "Estimation of individual prediction reliability using the local sensitivity analysis," *Applied Intelligence*, Št. 3, zv. 29, str. 187-203, 2008.

[3] Z. Bosnić, I. Kononenko, "Comparison of approaches for estimating reliability of individual regression predictions," *Data & Knowledge Engineering*, Št. 3, zv. 67, str. 504-516, 2008.

[4] Z. Bosnić, I. Kononenko, "Automatic selection of reliability estimates for individual regression predictions," *The Knowledge Engineering Review*, Št. 1, zv. 25, str. 27-47, 2010.

[5] L. Breiman, "Bagging predictors," *Machine Learning*, Št. 2, zv. 24, str. 123-140, 1996.

[6] L. Breiman, "Classidication and regression trees,Čhapman & Hall/CRC, Boca Raton (1984).

[7] L. Breiman, "Random forests," *Machine learning*, Št. 1, zv. 45, str. 5-32, 2001.

[8] L. Carlsson, E.A. Helgee, S. Boyer, "Interpretation of Nonlinear QSAR Models Applied to Ames Mutagenicity Data," *Journal of chemical information and modeling*, Št. 11, zv. 49, str. 2551-2558, 2009.

[9] C. Cortes, V. Vapnik, "Support-vector networks," *Machine learning*, Št. 3, zv. 20, str. 273-297, 1995.

[10] J. Demšar, B. Zupan, G. Leban, T. Curk, "Orange: From experimental machine learning to interactive data mining," *Knowledge discovery in databases: PKDD 2004*, Str. 537-539, 2004.

[11] R.C. Glem in drugi, "Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME," *IDrugs: the investigational drugs journal*, Št. 3, zv. 9, str. 199, 2006.

[12] I. Kononenko, "Strojno učenje," *Založba fakultete za elektrotehniko in fakultete za računalništvo in informatiko*, Ljubljana (2005)