

PRIMENA METODA MAŠINSKOG UČENJA U PROCESU RAZVOJA INFORMACIONIH SISTEMA

Vladislav Miškovic

Univerzitet Singidunum, Departman za informatiku i računarstvo

Danijelova 32, Beograd, Srbija

Telefon: +381 11 3121 351

Fax: +381 11 3093 294

E – mail: vmiskovic@singidunum.ac.rs

Sažetak

Razvoj informacionih sistema je složen proces, koji uključuje definisanje, analizu i specifikaciju zahteva, projektovanje, implementaciju, testiranje i održavanje softvera informacionog sistema. Mnogi zadaci u procesu razvoja mogu se formulisati kao problemi učenja, za čije se (automatizovano) rešavanje mogu upotrebiti različiti metodi mašinskog učenja. U radu se razmatra primena metoda mašinskog učenja u različitim fazama razvoja informacionog sistema i daje pregled karakterističnih praktičnih problema i metoda za njihovo rešavanje. Demonstriraju se rezultati primene u nekoliko poznatih problema razvoja informacionog sistema: proceni mogućnosti višestrukog korišćenja u budućem razvoju (*reusability*) i proceni cene razvoja (*software cost*) prema COCOMO/COCOMO II modelu.

Ključne reči: razvoj informacionih sistema, mašinsko učenje, algoritmi učenja, klasifikacija, regresija

Abstract

Information systems development is a complex process, which includes defining, analysing and specifying requirements, designing, implementing, testing and maintaining software and information systems. Many tasks in process of development can be formulated as learning problems, for whose (automated) solving different methods of machine learning can be used. The paper discusses application of machine learning methods in different stages of information systems development and provides an overview of typical problems and practical methods for their solving. The results of applying several well-known problems of information system development are demonstrated: evaluation of reusability in future development and development cost assessment (software cost) according to the COCOMO/COCOMO II model.

Keywords: information systems development, machine learning, learning algorithms, classification, regression

1. UVOD

Metodi mašinskog učenja predstavljaju snažan analitički alat za otkrivanje i primenu skrivenog znanja u rešavanje praktičnih problema u mnogim empirijskim disciplinama [1], [2]. U složenom procesu razvoja informacionih sistema prikupljaju se brojni podaci, vrše različite analize i donose odluke na osnovu empirijskih činjenica. Mnogi zadaci u procesu razvoja mogu se formulisati kao problemi učenja, za čije se (automatizovano) rešavanje mogu upotrebiti različiti metodi mašinskog učenja, koji su posebno korisni kada nisu poznate egzaktna procedure rešavanja, kada postoji veliki broj implicitnih međuzavisnosti dokumentovanih u bazama podataka, kao i za izgradnju sistema koji moraju da funkcionišu u

okruženju koje se neprekidno menja. Mogu se koristiti samostalno ili u sastavu alata za podršku razvoja informacionih sistema.

1.1 Proces razvoja informacionih sistema

Proces razvoja informacionog sistema (*system development process, system development lifecycle*) je skup aktivnosti, metoda, praktičnih iskustava, preporuka i automatizovanih alata, koje učesnici u razvoju koriste za razvoj i stalno usavršavanje informacionog sistema i softvera [3]. Tipične faze ili discipline projekta razvoja su: definisanje zahteva (*requirements*), analiza (*analysis*), specifikacija (*specification*), projektovanje (*design*), implementacija (*implementation*), testiranje (*testing*), isporuka/uvođenje (*deployment*) i održavanje (*maintenance*).

Mnogi zadaci procesa razvoja ili održavanja oslanjaju se na određene funkcije (preslikavanja, modele), koje se koriste za predviđanje, procenu, klasifikaciju, otkrivanje, usvajanje, razumevanje, stvaranje ili transformaciju određenih kvalitativnih ili kvantitativnih aspekata informacionog sistema ili proizvoda procesa razvoja. Primena mašinskog učenja u procesu razvoja sastoji se u upotrebi metoda automatizovanog učenja [2] za pronalaženje takvih funkcija, koje se kasnije mogu koristiti za realizaciju određenih zadataka razvoja.

1.2 Pojam induktivnog učenja

Metodi mašinskog učenja obuhvataju učenje memorisanjem, učenje na osnovu rečenog, učenje po analogiji i induktivno učenje, u koje spada učenje na osnovu primera i učenje samostalnim otkrivanjem. Induktivno učenje se može posmatrati i kao proces u kome programski sistem poboljšava svoje performanse na nekom zadatku bez dodatnog programiranja, ali i kao deo jedne opšte eksperimentalne procedure, koja se koristi u različitim područjima nauke i tehnike [2]. Opšti eksperimentalni metod otkrivanja zakonitosti u podacima predviđa: (1) iskazivanje problema, (2) formulaciju hipoteza (3) dizajniranje eksperimenta i generisanje podataka (4) prikupljanje i pretprocesiranje podataka (5) estimaciju modela i (6) interpretaciju modela i izvođenje zaključaka.

Svi elementi eksperimentalnog metoda su važni i utiču na konačne zaključke. Formalni metodi i algoritmi estimacije (korak 5) ne mogu da nadoknade nedovoljno informativne podatke ili neadekvatan model problema (koraci 1-3), koji može nepovoljno da utiče na zaključivanje na osnovu podataka ili ga čak onemogućiti.

Učenje na osnovu podataka ima istovremeno dva različita cilja: tačnost naučenog znanja (generalizaciju) i interpretaciju (objašnjenje). Rezultat učenja je model visoke tačnosti (korak 5), ali i *objašnjenje* ustanovljenih zavisnosti (korak 6), koje je neophodno za dublje razumevanje pojave, ograničenja u primeni metoda, veze ulaznih i izlaznih promenljivih i donošenje odluka.

Sistem induktivnog mašinskog učenja se može nalaziti u dva stanja: (1) učenje ili estimacija na osnovu obučavajućih primera i (2) upotreba ili predviđanje budućih ili testnih primera.

1.3 Metodi (algoritmi) i sistemi mašinskog učenja

Induktivno učenje se formalno definiše kao proces estimacije nepoznate ulazno-izlazne zavisnosti ili strukture posmatranog sistema S na osnovu konačnog broja opservacija x [2]. Skup funkcija koje se mogu naučiti i način ocene njihove najbolje aproksimacije definiše se unapred, samom konstrukcijom algoritma A i na osnovu nekog (pred)znanja o sistemu S .

Razlikuje se *učenje* koncepata (ili učenje s učiteljem), gde spadaju problemi klasifikacije i regresije i *formiranje* koncepata (ili učenje bez učitelja, grupisanje ili klastering).

Formiranje modela problema vrši se na osnovu predznanja o problemu i često se svodi na izbor atributa, odnosno promenljivih x_i , $i=1..n$. Uključivanje irelevantnih i redundantnih atributa u model problema negativno utiče na performanse većine metoda induktivnog učenja, pa se takvi atributi često uklanjaju iz razmatranja postupkom prethodne ili ugrađene selekcije atributa (*feature selection*) [2], [4], [5].

Najčešći konkretni problemi učenja su klasifikacija i regresija, odnosno učenja funkcije predviđanja neke diskretne (klasifikacija) ili kontinualne veličine ili atributa (regresija).

Predviđanje vrednosti diskretne veličine se vrši metodima učenja funkcija klasifikacije, koje mogu biti u obliku matematičkih ili logičkih iskaza, stabala, pravila i tabela odlučivanja, grafova, mreža, hiperpovršina i drugih pogodnih formalizama.

Predviđanje vrednosti neke kontinualne veličine na osnovu empirijskih podataka može se izvršiti metodima linearne i nelinearne regresije [2], [6]. Mnogo opštije i razumljivije funkcije predviđanja mogu se dobiti učenjem prediktivnih modela, kao što su npr. stabla i pravila regresije [7], [8], [9]. Stabla regresije se koriste za predviđanje vrednosti numeričkog atributa i mogu biti po delovima konstantne funkcije (*regression trees*) ili linearne funkcije (*model trees*).

2. PRIMENA INDUKTIVNOG UČENJA NA PROBLEME RAZVOJA INFORMACIONIH SISTEMA

U praksi su metodi mašinskog učenja uspešno korišćeni u rešavanju brojnih zadataka razvoja informacionih sistema i to u prikupljanju korisničkih zahteva (izvlačenje znanja, izrada prototipova), estimaciji mogućnosti ponovnog korišćenja softverskih modula (aplikacije i generatori), testiranju i validaciji sistema, održavanju (razumevanje softvera) i upravljanju projektima razvoja (procena cene i trajanja projekta razvoja, obima sistema i predviđanje ili procena greški u sistemu). Osnovne kategorije problema razvoja prikazani su u Tabeli 1 [1].

Tabela 1: Pregled opštih problema procesa razvoja i primenjenih metoda učenja

	NN	IBL, CBR	DT	GA-GP	ILP	EBL	CL	BL	AL	SVM
Predviđanje i estimacija	•	•	•	•	•		•	•		
Otkrivanje svojstava i modela	•				•	•		•		•
Transformacija	•	•		•						
Generisanje i sinteza		•	•	•	•	•	•		•	
Ponovno korišćenje softvera	•	•	•	•		•				
Prikupljanje			•		•	•	•			
Upravljanje		•								

Metodi: NN - neuralne mreže, IBL-CBR - učenje na osnovu instanci, DT - učenje stabala odlučivanja, GA-GP - genetički algoritmi, ILP - učenje logičkih pravila, EBL - učenje na osnovu objašnjenja (reformulacija), CL - učenje koncepata-pravila, BL - Bayesovski metodi, AL - analitički (deduktivni) metodi, SVM - noseći vektori

Pregled predstavlja samo jedan presek stanja i ne postavlja ograničenja na mogućnost primene brojnih srodnih metoda učenja na istom zadatku [2], [6]. Npr. u ovom radu će se primena metoda mašinskog učenja u predviđanju mogućnosti ponovne ili višestruke upotrebe softvera (*software reuse*) i proceni troškova razvoja informacionog sistema (*software cost/effort*) demonstrirati metodima učenja stabala odlučivanja (*decision trees*, DT) i produkcionih pravila (*rule learning*). Koriste se samo tzv. razumljivi (*comprehensible, understandable*) metodi induktivnog učenja na osnovu primera [10], [5], jer su rezultati učenja direktno razumljivi čoveku, pa ih je moguće direktno interpretirati i koristiti i bez pomoći računara.

Razumljivost rezultata učenja veoma je važna u procesu razvoja informacionih sistema, jer omogućava kontrolu automatizovanih procedura, stvara poverenje u njihove rezultate i poboljšava komunikaciju unutar razvojnih timova.

U razumljive metode učenja spadaju učenje stabala i pravila klasifikacije i regresije, koji istovremeno vrše i selekciju relevantnih atributa. Tipični predstavnici su klasični algoritmi za induktivno učenje stabala odlučivanja (C4.5 [11], CART [7]), produkcionih pravila (C45Rules [6], RIPPER [6], Empiric.Rules [12]) i metodi učenja regresionih stabala CART [7], M5 [8] i M5P [9] i regresionih pravila M5Rules [13].

3. PRIMER PROCENE MOGUĆNOSTI VIŠESTRUKOG KORIŠĆENJA SOFTVERA

Važan zadatak višestruke upotrebe softvera je izbor i procena mogućnosti ponovne upotrebe objekata (modula, baza podataka, testnih procedure ili podataka), koja se može izvršiti ako se znaju faktori koji utiču na uspešnost višestruke upotreba softvera.

Skup podataka **reuse** [14] formiran je na osnovu intervjua o rezultatima industrijskih projekata u kojima je korišćena praksa višestruke upotrebe softvera [15]. Izdvojena su 23 primera ozbiljnih pokušaja uvođenja ove prakse od ukupno 288 realizovanih u periodu 1994-1997, opisana sa 27 atributa. Projekti su razvrstani u dve grupe, prema tome da li je višestruka upotreba softvera bila uspešna ili neuspešna. Na Sl. 1. prikazana su naučena stabla i pravila za izabrane metode učenja iz sistema WEKA, kao i tačnost predviđanja za metode mašinskog učenja iz sistema WEKA (C45, C45Rules, JRipper) [6].

stabla	
WEKA.C45 95,8%	Human Factors = yes: success (16.0/1.0) Human Factors = no: failure (8.0)
WEKA.CART 95,8%	CART Decision Tree : success (15.0/9.0)
pravila	
WEKA.C45Rules 95,8%	Human Factors = yes: success (16.0/1.0) : failure (8.0)
WEKA.Ripper 95,8%	(Human Factors = no) => Success or Failure=failure (8.0/0.0) => Success or Failure=success (16.0/1.0)

Slika 1: Rezultati metoda učenja iz sistema WEKA za problem reuse

Na Sl. 2. prikazana su naučena pravila i tačnost predviđanja za metode mašinskog učenja iz sistema Empiric [12].

pravila	
Empiric.Gini (3) 95,0% ±15.0	[ReuseApproach=tight]->[Success or Failure=success] 11/0 [DevelopmentApproach=OO,proc]->[Success or Failure=success] 6/0 [HumanFactors=no]->[Success or Failure=failure] 7/0
Empiric.Q (2) 90,0%±20.00	[HumanFactors=yes][TypeofSoftwareProduction=product-family]-> [Success or Failure=success] 15/0 [HumanFactors=no]-> [Success_or_Failure=failure] 8/0 [TypeofSoftwareProduction=isolated]-> [Success_or_Failure=failure] 1/0
Empiric.ls (3) 90,0%±20.00	[ReuseApproach=tight]->[Success or Failure=success] 11/0 [ReuseProcessesIntroduced=yes] [TopManagementCommitment=yes]-> [Success or Failure=success] 4/0 [HumanFactors=no]->[Success or Failure=failure] 8/0 [TypeofSoftwareProduction=isolated]-> [Success or Failure=failure] 1/0

Slika 2: Rezultati metoda učenja iz sistema Empiric za problem reuse

U ovom primeru, metodi mašinskog učenja su upotrebljeni za otkrivanje faktora koji utiču na uspešnost primene višestruke upotreba softvera. Različiti metodi su ustanovili i različite

faktore, ali su svi sistemi izdvojili atribut *Human Factors* kao najvažniji ili jedan od najvažnijih faktora uspešnosti, uz procenu buduće tačnosti predviđanja na oko 95%.

Sistem *CART* [7] je kao najbolje rešenje našao apriornu procenu da će projekt biti uspešan. Sistemi *C45*, *Ripper* i *C45Rules* odluku zasnivaju samo na ljudskom faktoru (*HumanFactors*), a sistem *Empiric*, uz ljudski faktor predlaže još nekoliko atributa prikazanih na Sl. 2. Slični rezultati još nekih metoda učenja za isti problem detaljno se diskutuju u [15], [16] i [17].

4. PRIMERI PREDVIĐANJA CENE I TRAJANJA PROJEKTA RAZVOJA

Predviđanje troškova projekta razvoja je problem učenja pravila regresije. U ovom radu se koriste tri modela problema, dve verzije COCOMO (*Constructive Cost Model*) modela i jedan poseban model [14]. Prva verzija COCOMO modela je namenjena softveru razvijenom u starijim tehnologijama, a estimacije se vrši na osnovu veličine, odnosno broja linija izvornog koda. Novija verzija modela COCOMO-II troškove razvoja procenjuje na osnovu uloženog rada izraženog u kalendarskim mesecima [18], [19].

1. Problem **cocomo81** [14] sadrži primere 63 realizovana projekta razvoja softvera, opisanih pomoću 17 numeričkih atributa, na osnovu prve verzije COCOMO modela estimacije.
2. Problem **cocomonasa_2** [14] sadrži primere 93 projekta razvoja softvera realizovanih u NASA u periodu 1971-1987. Projekti su opisani sa 24 atributa: 7 opštih (naziv, kategorija i sl.), 15 atributa iz COCOMO-I modela u diskretnom obliku, jedan atribut za veličinu informacionog sistema (u broju linija izvornog koda) i jedan *ciljni* atribut - potreban obim rada na projektu u čovek-mesecima.
3. Problem **desharnais** [14] sadrži 81 primer projekata razvoja, opisanih prema posebnom modelu sa 12 numeričkih atributa.

Upotrebljena su tri metoda: učenje regresionih stabala metodima *M5* [8] i *M5P* [9] i učenje regresionih pravila pomoću algoritma *M5Rules* [13] implementiranih u sistemu *WEKA* [6].

U Tabeli 3 prikazani su koeficijent korelacije, srednja kvadratna greška (*root mean squared error*, *RMSE*) i relativna kvadratna greška (*root relative squared error*, *RRSE*) [6].

Tabela 3: Koeficijent korelacije i greške predviđanja za razmatrane probleme

Problem	Obučavajući skup		Algoritam učenja								
	Broj primera	Broj attrib.	M5			M5P			M5PRules		
			<i>k.kor.</i>	<i>RMSE</i>	<i>RRSE</i>	<i>k.kor.</i>	<i>RMSE</i>	<i>RRSE</i>	<i>k.kor.</i>	<i>RMSE</i>	<i>RRSE</i>
cocomo81	270	13	0.537	1547.5	84.3%	0.684	1334.7	72.7%	0.766	1290.0	70.3%
cocomonasa_2	93	23	0.701	809.1	70.8%	0.709	798.7	69.9%	0.711	801.1	70.1%
desharnais	81	12	0.694	3202.5	71.8%	0.706	3145.7	70.5%	0.699	3178.6	71.2%

Za dva problema prema COCOMO modelu, regresiona stabla i pravila su istovremeno i uspešnija od modela linearne regresije, a za model **desharnais**, bolje rezultate daje model estimacije pomoću linearne regresije (*k.kor*=0.735, *RMSE*=2988.9 i *RRSE*=67.0%).

5. ZAKLJUČAK

U radu su navedeni neki problemi procesa razvoja informacionih sistema, koji se mogu uspešno rešavati različitim metodima mašinskog učenja. Istaknuta je uloga razumljivih metoda induktivnog učenja, koji na osnovu empirijskih podataka o ispravnim ili rešenim slučajevima generišu znanje, koje se može upotrebljavati za automatizovano rešavanje novih problema, ali i za bolje razumevanje razmatranog problema i upotrebu od strane čoveka.

Demonstrirana je primena ovih metoda u rešavanju dva praktična problema. Procena mogućnosti višestruke upotrebe softverskog modula je problem projektovanja, dok je procena troškova razvoja problem upravljanja projektom. U oba slučaja su, metodima mašinskog učenja, dobijena razumljiva rešenja prihvatljive tačnosti predviđanja.

Osim ovih primera, navedeni su i drugi zadaci u procesu razvoja koji se mogu uspešno rešavati ovim metodima, koristeći opšte metode i alate mašinskog učenja, kako besplatne [6], tako i komercijalne, koji se ugrađuju u razvojne sisteme, npr. podsisteme poslovne inteligencije softvera za razvoj i upravljanje bazama podataka [20], [21].

LITERATURA

- [1] Zhang Du, Tsai J. J.P. (eds), *Machine Learning Applications in Software Engineering*, World Scientific Publishing Co. Pte. Ltd., 2005.
- [2] Cherkassky V., Mulier F. M., *Learning from Data: Concepts, Theory, and Methods*, 2nd edition, John Wiley - IEEE Press, 2007.
- [3] Whitten J. L., Bentley L. D., Dittman K. C., *Systems Analysis and Design Methods*, 6th Ed, Irwin McGraw Hill, 2004.
- [4] Milosavljević M., Buturović L.J., "Prepoznavanje i ekstrakcija obeležja genskih ekspresija", *Zbornik radova 50. Konferencije za ETRAN*, tom III, pp. 177-179, Beograd, 6-8. juna 2006.
- [5] Miškovic V., *Induktivno učenje razumljivog znanja na osnovu oskudnih obučavajućih skupova*, Doktorska disertacija, Univerzitet Singidunum, Beograd, april 2008.
- [6] Witten I. H., Frank E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, pp. 265-320, Morgan Kaufmann Publishers, 2005.
- [7] Breiman L., Friedman J.H., Olshen R.A., Stone C.J., *Classification and Regression Trees*, Wadsworth, Belmont, 1984.
- [8] Quinlan R. J., "Learning with Continuous Classes", In *5th Australian Joint Conference on Artificial Intelligence*, pp. 343-348, Singapore, 1992.
- [9] Wang Y., Witten I. H., "Induction of model trees for predicting continuous classes", In *Poster papers of the 9th European Conference on Machine Learning*, 1997.
- [10] Michalski, R., Carbonell, J., Mitchell T. (Eds.), *Machine learning: An artificial intelligence approach (Vol. I)*, San Francisco, CA: Morgan Kaufmann, 1983.
- [11] Quinlan J. R., "Bagging, Boosting and C4.5", in *Proc. of AAAI-96 Fourteenth national Conference on Artificial Intelligence*, Portland, OR, AAAI Press, Menlo Park, CA, 1996.
- [12] Miškovic V., *Jedna klasa algoritama za induktivno učenje*, Magistarski rad, Elektrotehnički fakultet, Univerzitet u Beogradu, Beograd, novembar 2002.
- [13] Holmes G., Hall M., Frank E., "Generating Rule Sets from Model Trees", *Proc. Twelfth Australian Joint Conference on Artificial Intelligence*, pp. 1-12, 1999.
- [14] Sayyad Shirabad J., Menzies T.J., *The PROMISE Repository of Software Engineering Databases*, School of Information Technology and Engineering, University of Ottawa, Canada, 2005 . Available: <http://promise.site.uottawa.ca/SERepository>
- [15] Morisio M., Ezran M., Tully C., "Success and failure factors in software reuse", *IEEE Transactions on Software Engineering*, volume 28, 4, pp. 340-357, 2002.
- [16] Di Stefano J.S., Menzies T., "Machine Learning for Software Engineering: Case Studies in Software Reuse", *Proceedings, 14th IEEE International Conference on Tools with AI (ICTAI 2002)*, pp. 246-251, 2002.

- [17] Menzies T., Di Stefano J.S., "More Success and Failure Factors in Software Reuse", *IEEE Transactions on Software Engineering*, 2003.
- [18] Boetticher G.D., "Using Machine Learning to Predict Project Effort: Empirical Case Studies in Data-Starved Domains", *Model Based Requirements Workshop*, pp. 17-24, San Diego, 2001.
- [19] Menzies T., Port D., Chen Z., Hihn J., Stukes S., "Validation Methods for Calibrating Software Effort Models", *Proceedings ICSE*, 2005.
- [20] Taft M., Krishnan R., Hornik M., Muhkin D., Tang G., Shiby T., Stengard P., *Oracle Data Mining Concepts*, 10g Release 2, June 2005.
- [21] Tang ZhaoHui, MacLennan J., *Data Mining with SQL Server 2005*, Wiley Publishing, Inc., 2005.