УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ” - ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ISSN:1857-8691

ГОДИШЕН ЗБОРНИК
2016/2017
YEARBOOK
2016/2017

ГОДИНА 5 **VOLUME V**

GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ГОДИШЕН ЗБОРНИК
2016/2017
YEARBOOK
2016/2017

ГОДИНА 5 **MAJ, 2017** **VOLUME V**

GOCE DELCEV UNIVERSITY – STIP
FACULTY OF COMPUTER SCIENCE

ГОДИШЕН ЗБОРНИК
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
YEARBOOK
FACULTY OF COMPUTER SCIENCE

За издавачот:
Проф д-р Цвета Мартиновска Банде

Редакција и администрација
Универзитет „Гоце Делчев“-Штип
Факултет за информатика
ул. „Крсте Мисирков“ 10-А
п. фах 201, 2000 Штип
Р. Македонија

Address of the editorial office
Goce Delcev University – Stip
Faculty of Computer Science
Krste Misirkov 10-A
PO box 201, 2000 Štip,
R. of Macedonia

С О Д Р Ж И Н А

СПОРЕДБА НА АЛГОРИТМИ ЗА КЛАСИФИКАЦИЈА

Мирјана Коцалева [1], Цвета Мартиновска - Банде [1]

[1] Факултет за Информатика, Универзитет „Гоце Делчев“, Штип
mirjana.kocaleva@ugd.edu.mk
cveta.martinovska@ugd.edu.mk

Апстракт. Податочното рударење е една од најкористените технологии денес. За таа цел во овој труд прво ги разгледуваме податочното рударење и класификацијата, а потоа се задржуваме на дрвата за одлучување како еден од начините за имплементација на класификацијата. Исто така ги разгледуваме перформансите на алгоритамите за градење на дрва за одлучување, вклучувајќи ја и ентропијата, Gini коефициентот, грешката при класификација, прецизност и recall, ROC криви итн. Во трудот е направена споредба на едни од најпознатите алгоритми за класификација и се презентирани резултатите добиени со примена на ID3 и J48 алгоритмите (J48 и J48 graft) на наша база на податоци. Базата на податоци е Student Database содржи информации за бројот на студенти, број на положени и неположени испити и просекот на студентите. Врз база на извршените истражувања заклучуваме дека кога користиме податоци за тренирање најдобри предвидувања се добиваат со употреба на ID3 алгоритамот.
Клучни зборови: податочно рударење, класификација, дрва на одлучување, WEKA.

COMPARISON OF CLASSIFICATION ALGORITHMS

Mirjana Kocaleva [1], Cveta Martinovska - Bande [1]

[1] Faculty of Computer Science, Goce Delcev University, Stip, Macedonia
mirjana.kocaleva@ugd.edu.mk
cveta.martinovska@ugd.edu.mk

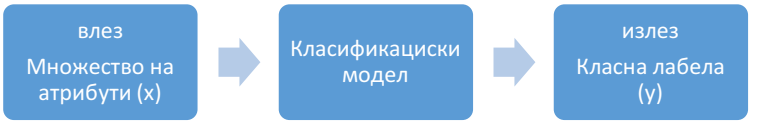
Abstract. Data mining is one of the most used technologies today. For this purpose, in this paper we first examine the data mining and classification, and then decision trees as a way of implementing the classification. We also look at the performance of algorithms for building decision trees, including entropy, Gini coefficient, the error in classification precision and recall, ROC curves etc. This paper is a comparison of some of the most famous algorithms for classification and also presents the results obtained by using ID3 and J48 (J48 and J48 graft) algorithms for our database. The database Student Database contains information for the number of students, number of passed and failed exams and average of students. Based on the research we conclude that when we use training set the best predictions are obtained using ID3 algorithm.
Keywords: data mining, classification, decision tree, WEKA.

1. Вовед

Основна задача на податочното рударење е откривање на корисни информации од големи бази на податоци и е еден од чекорите на KDD (knowedge discovery in databases) за откривање на нови, корисни и лесно разбирливи шеми на податоци. Секој KDD процес се состои од следниве чекори: селекција, предпроцесирање, трансформација, податочно рударење и интерпретација. Податочното рударење работи со алгоритами за решавање на проблеми од секаков вид. Алгоритмите ги испитуваат податоците и го определуваат моделот кој е најблизок до карактеристиките на тие податоци. Влез во алгоритмот е табела која се состои од атрибути (колони) и записи (редици), а излез е шаблон за тие податоци. Најчести шаблони кои се користат се: класификација, кластеризација и асоцијативни правила. Податочното рударење може да се примени во секоја научно истражувачка област, како на пример: Web mining, Bank loan decisions, Image screening, Load forecasting, Fault diagnosis, Marketing and sales итн.

2. Класификација

Класификацијата е задача за учење на одредена целна функција f која го означува секое множество на атрибути x во една од дефинираните класни лабели y. Целната функција е исто така позната и како модел за класификација (слика 1). Влезните податоци за класификација се колекција од записи. Секој запис се запишува како подреден пар (x, y) каде x е множество на атрибути и y е специјален атрибут или класна лабела. Атрибутите може да се дискретни, но и континуирани. Класната лабела мора да е дискретен атрибут. Ова е клучната карактеристика по која класификацијата се разликува од регресијата која е модел за предвидување кога y е континуиран атрибут (функција).



Слика 1 Класификација - влезното множество на атрибути x го поставува во класа y
Figure 1 The process of classification – an input attribute set x into class label y

Моделот за класификација се користи во следните цели:

Описно моделирање – овој модел служи како алатка за објаснување која прави разлика помеѓу објектите на различни класи. На пример, може да е корисна за биолозите да добијат описен модел кој сумира одредени податоци за животните и со нивна помош да објаснат кои карактеристики ги дефинираат птиците, рибите, влекачите, водоземците и цицачите.

Моделирање со предвидување – класифицирањето може да се користи за предвидување на класната лабела на непознат примерок.

Техниките за класификација се погодни за предвидување и опис на податоци со бинарни или номинални (мали) категории.

Податоците се делат на 2 дела, податоци за тренирање и податоци за валидација. Податоците кои се користат за тренирање го креираат моделот (имаат позната класна лабела). Потоа добиениот модел се тестира на податоците за валидација (кои се состојат од примероци со непозната класна лабела) и се мери колку е овој модел е добар. Тоа е стандардна процедура во Машинско учење.

		Предвидувачка класа	
		Класа=1	Класа=0
Актуелна класа	Класа=1	f_{11}	f_{10}
	Класа=0	F_{01}	F_{00}

Слика 2 Confusion matrix за дво класен проблем
Figure 2 Confusion matrix for two class problem

Евалуацијата на ефикасноста на класификациониот модел се базира на пребројување на податоците за тестирање кои се точно и неточно предвидени со моделот. Овие пребројувања се сместуваат во табела која се вика confusion matrix (слика 2). f_{ij} го означува бројот на примероци од класа i предвиден да биде од класа j. F_{01} е број на примероци од класа 0 кои се неточно предвидени како класа 1. Вкупниот број на точно предвидени примероци е ($f_{11} + f_{00}$) (формула 1), а на неточно предвидени е ($f_{10} + f_{01}$) (формула 2). За да се согледа колку е добар овој модел се пресметува точност и грешка:

$$\text{точност} = \frac{\text{број на точни предвидувања}}{\text{вкупен број на предвидувања}} = \frac{f_{11} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad (1)$$

$$\text{грешка} = \frac{\text{број на неточни предвидувања}}{\text{вкупен број на предвидувања}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad (2)$$

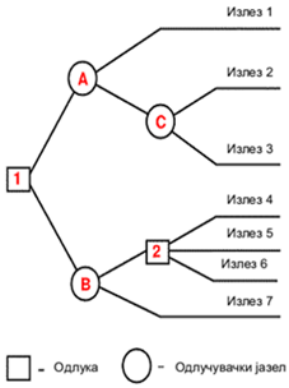
Обично се бараат модели со поголема точност и помала грешка.

3. Дрва за одлучување

Основна задача на класификацијата е да ја предвиди вредноста на непознатиот атрибут во зависност од вредноста на останатите атрибути. Во процесот на класификација влезот е множество на податоци, а на излез се генерира модел (шаблон или множество од шаблони). Добиениот модел се користи за предвидување на вредностите на класата за нови податоци. Од дадено множество податоци (табела) само еден дел се користат за да се генерира предиктивниот модел. Овој дел се нарекува множество за обучување. Останатиот дел се користи за евалуација на предвидувачките можности на научениот модел и се нарекува множество за тестирање. Множество за тестирање се користи за да се процена точноста на моделот на нови, дотогаш неупотребувани податоци, или со други зборови, да се процени валидноста на шаблонот на нови податоци. Препознавањето на шаблонот е вид на класификација каде влезниот шаблон е класифициран во една од неколкуте класи врз основа на сличност со предефинираните класи.

Дрвото за одлучување е метод за апроксимација на дискретни - вредносни целни функции, во кои функцијата е претставена со дрво на одлучување. Дрвата исто така, може да се претстават како множества на if - then правила за подобрување на човековата читливост. Дрвата за одлучување се состојат од одлучувачки јазли, каде секој одлучувачки јазел двојно или повеќекратно се разгранува, при

што секоја гранка репрезентира вредност на атрибутот кој се тестира. Листовите понатаму не се разгранууваат и истите се генератори на униформен (конечен) заклучок.



Слика 3 Дрво за одлучување
Figure 3 Decision tree

Секое дрво за одлучување започнува со јазел кој ја претставува така наречената првична одлука, од каде започнува разгранувањето на стеблото. На слика 4, јазелот кој ја репрезентира првичната одлука е претставена со квадрат. Листовите го генерираат конечниот резултат ако се следи конкретна патека долж стеблото.

Дрвото има три вида на основни јазли:

- Root – нема влезен јазол и или 0 или повеќе излезни јазли
- Internal – секој јазол кој има точно еден влезен јазол и 2 или повеќе излезни јазли
- Leaf/terminal (или класна лабела) – секој јазол кој има точно еден влезен јазол и нема излезни јазли

Постојат многу предности за употреба на дрва за одлучување за класификација. Тие се ефикасни и се лесни за употреба. Може да се генерираат правила кои се лесни за интерпретација и разбирање. Тие одговараат добро и за големи бази на податоци бидејќи големината на дрвото е независна од големината на базата. Дрва може да се конструираат за податоци со многу атрибути. Кај дрвата на одлучување постојат и негативности. Има можности за континуирани податоци, но не се справуваат лесно со нив, бидејќи се прави регресија во секој лист. Домените на ваквите атрибути мора да се поделат во категории кои понатаму треба да се опфатат во алгоритамите. Пристапот кој се користи во овој случај е да се подели доменскиот простор. Понатаму, тешко се решава проблемот на недостаток на одредени податоци бидејќи во тој случај не може да се одреди точната гранка по која би се движеле по дрвото. Бидејќи дрвото се конструира од податоци за обучување може да настане претерано нагодување. Ова може да се надмине со поткастрување на дрвото (tree pruning). Во процесот на индукција на дрвото за одлучување се игнорираат корелациите меѓу атрибутите во базата.

Најважните фактори што влијаат во перформансите на алгоритамите за градење на дрва за одлучување се големината на множеството за обучување и начинот како се определува најдобриот атрибут за поделба.

Кај повеќето алгоритами најчесто се разгледуваат следниве 7 прашања:

1.Избор на атрибут за поделба
2.Редослед на атрибутите за поделба
3.Поделби
4.Структура на дрвото
5.Критериум за запирање
6.Податоци за обучување
7.Поткастрување

Постојат и други мерки за оценка на перформансите на еден модел како што се: прецизност и recall, ROC криви и др. Поимите за прецизност (формула 3) и recall (формула 4) се дефинирани со следниве формули:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

каде TP е број на точни позитивни класификации, FP е број на неточни позитивни класификации, TN е број на точни негативни класификации и FN е број на неточни негативни класификации. Значи, recall ни кажува колкав дел од инстанците за тестирање се класифицираат како позитивни, додека прецизноста ни кажува колкав дел од случаите кои се класифицирани како позитивни се навистина позитивни.

ROC кривите ни го даваат односот на ратите на TP и FP, дефинирани со формулите 5 и 6:

$$TP\ rate = \frac{TP}{TP + FN} \quad (5)$$

$$FP\ rate = \frac{FP}{FP + TN} \quad (6)$$

Изгледот на дрвото на одлучување исклучително зависи од податоците за обучување и алгоритамот за индукција на дрвото. Пожелно е да се добие дрво кое работи добро врз податоците за обучување. Некои од алгоритамите креираат само бинарни дрва. Овие дрва се лесни за индукција, ама тие имаат тенденција да имаат поголема длабочина. Перформансите на ваквите дрва можат да бидат полоши бидејќи обично е потребен поголем број на споредби. Глобално гледано, овие споредби се сепак поедноставни од повеќенасочното гранење така да вкупните перформанси на бинарните дрва се споредливи со останатите.

Алгоритмите за индукција на дрвата за одлучување можат да го изградат дрвото и потоа да извршат негово поткастрување за класификацијата да биде поефикасна. Со техниките на поткастрување, делови од дрвото можат да се отстранат или да се комбинираат за да се редуцира вкупната големина на дрвото. Поткаструвањето може да се изведува додека се креира дрвото, со тоа да се избегне дрвото да стане премногу големо, или вториот пристап е да се поткастри дрвото по градењето. Временската и просторна комплексност на алгоритмите за индукција на дрва за одлучување зависи од големината на податоците за обучување q, бројот на атрибути h и обликот на креираното дрво. Во најлош случај, дрвото може да биде многу длабоко и да не е многу разгрането. Додека се гради дрвото, за секој од јазлите, секој атрибут ќе биде испитан за да се определи дали е најдобар. Тоа доведува временската комплексност на дрвото на дрвото да биде O(hq log q) . Времето што е потребно да се класифицира база со големина n се базира на висината на дрвото. Ако се претпостави висина на дрвото O(log q) , тогаш комплексноста е O(nlog q) .

Главниот проблем е од кој атрибут да се почне со класификацијата - целта е да се избере најкорисниот атрибут (сличен на бирање примерок кој ја преполовува верзијата на просторот на проблемот за формирање на концептот), и тоа го прави оценувањето на статистичките својства на информациите да добие (моќ и информациска добивка), која се дефинира со множество на ентропијата S (формула 7):

$$\mathrm{E}(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus} \quad (7)$$

каде $p_{\oplus}$ е пропорционално на односот на позитивните примероци во S , а $p_{\ominus}$ е односот на негативните во S (има вредност нула ако сите примероци припаѓаат на иста класа). Ако целниот атрибут во општиот случај има c дискретни вредности, тогаш $E(S) \equiv \sum_{i=1}^{c} -p_i log_2 p_i$.

Ако Values(A) е множество на вредности на атрибутот A и $S_v = \{s \in S : A(s) = v\}$ тогаш:

$$Gain(S, A) \equiv E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v) \quad (8)$$

Gini коефициентот (формула 8) ја мери нееднаквоста меѓу вредностите на распределбата на фреквенција (на пример, нивото на приход). Gini коефициентот нула изразува совршена еднаквост, каде што сите вредности се исти (на пример, каде што секој има еднакви примања). Gini коефициентот еден изразува максимална нееднаквост помеѓу вредностите (на пример, кога само една личност ги има сите приходи).

Максимум (1 - 1/nc) кога примероците се еднакво распределени помеѓу сите класи, што значи најмалку интересни информации - Минимум (0.0), кога сите примероци припаѓаат на една класа, што значи најинтересни информации. Овој коефициент се пресметува со формулата 9 :

$$Gini(t) = 1 - \sum_{i=0}^{c-1} |p(i|t)|^2 \quad (9)$$

Грешката при класификацијата на примероците се пресметува со формулата 10:

$$Classification\ error(t) = 1 - max_i[p(i|t)] \quad (10)$$

4. Видови на дрва за одлучување

CART е дрво на класификација за глобално оптимална анализа (GO-CTA) (исто така наречена хиерархиски оптимална дискриминантна анализа) и претставува генерализација на оптималната дискриминантна анализа што може да се користи за идентификување на статистичкиот модел кој има максимална точност за предвидување на вредноста на категорички зависната варијабла за база која се состои од категорични и континуирани променливи. Classification and regression trees (CART) (Дрвја за класификација и регресија) се не параметарски техники за учење на дрва за одлучување кои

произведуваат и класификациски и регресиски дрвја, во зависност од тоа дали зависната варијабла е категоричка или нумеричка, соодветно.

Дрвјата на одлука се формирани од сет на правила базирани на променливи во група на податоци за моделирање:

- Правила базирани на вредностите на променливите кои се избираат за да се добие најдоброто разделување на податоците врз основа на зависната променлива
- Откако правилото е избрано и јазолот е поделен на два, истиот процес се применува на секој "дете" јазол (т.е. тоа е рекурзивна постапка)
- Разделување запира кога CART не гледа дека понатамошно зголемување (gain) може да се направи, или се исполнети некои пред - сет правила на запирање.
- Секоја гранка на дрвото завршува во терминал јазол. Секој примерок припаѓа во еден и само еден терминал јазол, и секој терминал јазол е уникатно утврдени со сет на правила.

CHAID е еден вид на техника на дрва за одлучување. CHAID може да се користи за предвидување (на сличен начин како регресивната анализа), исто како и класификацијата, како и за откривање на интеракцијата помеѓу променливите.

Во пракса, CHAID често се користи во контекст на директен маркетинг за да се избере група на потрошувачи и да се предвиди како нивните реакции за некои променливи влијаат на другите променливи, иако почнува да се применува и во апликации од областа на медицинските и психијатриските истражувања.

Како и другите дрва за одлучување, предности кај CHAID техниката се дека нивниот излез е визуелен и лесно може да се протолкува. Поради тоа што користи повеќекратни поделби по дифолт, потребен му е прилично голем примерок за да работи ефикасно, бидејќи со мал примерок групите кои даваат одговор брзо може да станат премногу мали за сигурна анализа.

Една важна предност на CHAID покрај повеќето алтернативи како што е повеќекратната регресија е дека е не-параметарска.

ID3 (Iterative Dichotomiser 3) е алгоритам кој се користи за генерирање на дрва на одлучување од база. ID3 е претходник на алгоритамот C4.5, и обично се користи во процесот на машинското учење и при процесирање на природните јазици (Natural language processing). Алгоритамот ID3 започнува со оригиналниот сет S како root јазол. При секое повторување на алгоритмот, тоа повторување минува низ секој неискористен атрибут на сетот S и притоа се пресметува ентропијата H (S) (или информациска добивка (information gain) IG(S)) на тој атрибут. Потоа се избира атрибутот кој има најмала ентропија (или најголема информациска добивка) вредност. Множеството или сетот S потоа се дели според избраниот атрибут (на пример, возраста е помалку од 50, возраста е помеѓу 50 и 100, и возраста е поголема од 100) за креирање на подгрупи на податоци. Алгоритамот продолжува рекурзивно на секоја подгрупа, разгледувајќи ги само атрибутите што никогаш пред тоа не биле избрани.

Рекурзијата на подмножеството може да престане во еден од следниве случаи:

- секој елемент во подмножеството припаѓа на иста класа (+ или -), а потоа јазолот се претвора во лист и се означува со класата на примерокот
- нема повеќе атрибути кои може да бидат избрани, но примероците се уште не припаѓаат на иста класа (некои од нив се + и некои од нив се -), а потоа јазолот се претвора во лист и се означува со најчестата класа на примероците во подмножество
- не постојат примероци во подгрупата, ова се случува кога не се наоѓа примерок во родител сетот кој се совпаѓа со одредена вредност на одбраниот атрибут, на пример ако немал примерок со возраст > = 100. Потоа листот е создаден и означен со најчестата класа на примероци во родител сетот.

При извршување на алгоритамот, дрвото на одлучување е изградено со секој не-терминален јазол кој го претставува избраниот атрибут со кој податоците се поделени, и терминалните јазли ги претставуваат класните лабели на последната подгрупа на оваа гранка.

ID3 алгоритамот е дизајниран да помогне во случаите кога постојат многу атрибути и множеството за обучување содржи многу торки, а е потребно да се добие разумно добро дрво за одлучување без многу големо пресметување. Тој конструира едноставни дрва за одлучување и неговата индуктивна пристрасност е предност во однос на повеќето мали дрва, а во состојба е да ги класифицира и дисјунктивните концепти. Овој метод може да биде далеку поефикасен од другите системи за индуктивно учење, но исто така и не применлив во некои комплицирани домени. Учењето со дрва на одлука претставува вид на функција за апроксимација на (дискретни вредности) атрибути и нивните дискретни вредности. Дрвото на одлука ги класифицира примероците со прифаќање на атрибутите од коренот до листот на една гранка, а може да се гледа и како листа на if-then правила (секоја гранка е конјунктивен однос на атрибути, а целото дрво е дисјункција).

C4.5 е алгоритам за генерирање на дрва за одлучување и тој претставува проширување на ID3 алгоритамот. Дрвата на одлучување генерирани со овој алгоритам може да се користат за класификација, и за регресија. C4.5 често се нарекува и статистички класификатор. C4.5 гради дрво на одлучување од

множество на податоци на истиот начин како ID3, користејќи го концептот на информациона ентропија. Множеството на податоци е $S = s_1,s_2,..$ на веќе класифицирани примероци. Секој примерок s_i се состои од p-димензионален вектор $(x_{1,i}, x_{2,i},...,x_{p,i})$ каде x_j ги претставува вредностите на атрибутите или карактеристиките на примерокот, како и класата s_i во која припаѓа.

Во секој јазол на дрвото, C4.5 избира атрибут на податоците кои најефикасно го делат множеството на примероци во подгрупи збогатени од една класа или друга. Критериумот за разделување е нормализираната информациска добивка (information gain) или разликата во ентропијата. Атрибутот со највисока information gain е избран да ја донесе одлуката. C4.5 алгоритамот потоа се повторува на помали под-листи.

Алгоритамот има неколку случаеви:

- Сите примероци во листата припаѓаат на иста класа. Кога тоа се случува, едноставно се создава лист јазол за дрвото на одлука кое вели дека треба да се избере таа класа.
- Ниту една од карактеристиките не обезбедува information gain. Во овој случај, C4.5 создава одлука јазол повисоко на дрвото со помош на очекуваната вредност на класата.
- Се среќава примерок на претходно невидена класа. Повторно, C4.5 создава одлука јазол повисоко на дрвото со помош на очекуваната вредност.

Во псевдокодот, општиот алгоритам за изградба на дрва за одлучување е:

- Проверете ги сите базни случаи.
- За секој атрибут a, најдете го нормализираниот сооднос на information gain со разделувањето на a.
- Нека a_best биде атрибут со највисока нормализирана information gain.
- Креирај одлука јазол кој се дели со a_best.
- Се врши повторување на под - листите добиени со поделба на a_best, и додадете ги овие јазли како деца на јазлите.

C4.5 прави голем број на подобрувања на ID3. Некои од нив се:

Управува со континуирани и дискретни атрибути - Со цел да управува со континуирани атрибути, C4.5 создава праг, а потоа ја дели листата на оние чија вредност на атрибутот е над прагот и оние кои имаат помала или еднаква вредност од дефинираната.

Управува со податоци за обука на кои им недостасуваат атрибут вредностите - C4.5 овозможува атрибут вредностите да бидат означени како ? за непознати. Атрибут вредностите кои недостасуваат едноставно не се користат при пресметување на gain-от и ентропијата.

Управување со атрибути со различните трошоци.

Ги крати дрвјата по создавањето - C4.5 оди назад низ дрвото, кога веќе е создадено и се обидува да ги отстрани гранките кои не помагаат, со нивна замена со лист јазли.

J48 алгоритамот го имплементира алгоритамот за учење на C4.5. J48 претставува претходна верзија на ID3 алгоритамот. Таа генерира не бинарно дрво и ја користи мерката наречена gain (добивка) за изградба на дрво за одлучување, атрибутот со највисок нормализиран gain се зема како root јазол и базата е поделена врз основа на вредностите на root елементот. Повторно информационата добивка се пресметува за сите под - јазол поединечно и процесот се повторува се додека предвидувањето не се заврши. J48 може да се справи со двете и со континуирани и со дискретни атрибути, податоци за обука со непознати вредности на атрибутите и атрибути со различните трошоци и исто така обезбедува можност да поткастрување на дрвјата по нивното создавање. Постојат голем број на параметри поврзани со поткастрување на дрвата во алгоритам J48 и треба да се користат со внимание за да можат да направат значајна разлика во квалитетот на резултатите.

J48 graft генерира накалемени дрва на одлучување од J48 дрво. Техниката на калемење додава јазли на постојното дрво на одлучување со цел да се намали грешката на предвидување. Овие алгоритами идентификуваат просторни региони на примероци, кои не се зафатени од примероците за обука, или се составени само со неправилно класифицирани примероците за обука, и се разгледуваат алтернативни класификации за овие региони. Со други зборови, нов тест ќе се врши во листовите, генерирајќи нови гранки кои ќе доведат до нови класификации. Калемењето е алгоритам за додавање на јазли на дрво како пост - процес. Нејзината цел е да се зголеми веројатноста за правилно класифицирање на инстанциите кои се надвор од областите опфатени со податоците за обука. Калемењето е пост - процес кој може да се примени на дрвата за одлучување. Нејзината цел е да се намали грешката на предвидување со прекласификација на просторните региони, каде не постојат податоци за обука или каде што има само неправилно класифицирани примероци. Нејзината цел е да најдат најдобрите намалувања на постојните региони со листови и да се креираат гранки за создавање на нови листови со други класификации. Новото дрво кое се создава ги намалува грешките, наместо да воведе нови.

5. Добиени резултати со употреба на WEKA

WEKA е софтвер за машинско учење / софтвер за податочно рударење кој се користи за истражување, образование и апликации. Системот WEKA е напишан во JAVA, објектно – ориентиран програмски јазик што е широко достапен за сите компјутерски платформи како Windows, Linux и Macintosh. JAVA ни дозволува да обезбедиме униформен интерфејс за многу различни алгоритами за учење, методи за пред и пост процесирање и евалуација на резултатите на шемите за учење на кое било множество податоци. Всушност, основната цел на WEKA системот е развивање на алгоритами за класификација и филтрирање на податоци.

Негови главни карактеристики се:

- Сеопфатен сет на податоци за алатки за обработка, алгоритами за учење и методи на евалуација
- Графички кориснички интерфејси (вклучувајќи и визуелизација на податоци).
- Околина за споредување на алгоритами за учење.

Во WEKA се имплементирани следните методи: Bayes, Decision trees and rules, Neural networks, Functions, Meta, lazy classifiers итн. Со следните опции за испитување: Користете сет за обука, Крос валидација, Процентуална поделба.

Учењето на дрва за одлучување во системот WEKA е имплементирано преку множество од класи. Секоја класа посебно има определена улога во алгоритамот. Концептуално гледано, класите можат да се поделат во неколку групи според функцијата што ја имаат во алгоритамот и тоа: класи за манипулација со дрва за одлучување, класи за селекција на модел на дрво, класи за поделба на ниво на јазол, класи за критериуми за поделба, класа за пресметка на распределбата на веројатност на ниво на јазол и друго. WEKA работи со фајлови во ARFF формат (Атрибут - врска формат на датотеката или *Attribute-Relation File Format)* кој е ASCII текстуална датотека која опишува листа на примероци кои споделуваат заеднички сет од атрибути.

За тестирање на перформансите на алгоритамите што се опишани во овој труд користено е множеството на податоци Student Database. Student Database е наша, сопствена база на податоци што содржи записи за студентите на нашиот факултет и тоа записи за бројот на запишани студенти по година, бројот на положени испити од можни 10 во годината, бројот на неположени испити од можни 10 во годината и податоци за просекот на студентите по година. Во базата постојат 4 нумерички атрибути и нивните вредности се од 0 до 2.

Кај бројот на студенти

- “0” се однесува на број на студенти помал од 40,
- “1” за број на студенти помеѓу 40 и 70 и
- “2” за повеќе од 70 студенти.

Кај положени испити

- “0” се однесува на број на положени испити помал од 6,
- “1” за број на положени испити помеѓу 6 и 8 и
- “2” за повеќе од 8 положени испити.

Кај неположени испити

- “0” се однесува кога немаме неположени испити,
- “1” за број на неположени испити помеѓу 0 и 2 и
- “2” за повеќе од 2 неположени испити.

Целта на ова истражување е со помош на записите кои ги имаме за бројот на студенти запишани во одредена студиска година, бројот на положени и неположени испити да го предвидеме просекот на студентите кој исто така може да биде

- “0” кога е помал од 7.5,
- “1” кога е помеѓу 7.5 и 8.5 и
- “2” кога е поголем од 8.5.

На слика 4 е даден преглед на базата и атрибутите заедно со вредностите.

```
@relation prosek
@attribute broj_na_studenti {0,1,2}
@attribute polozeni_ispiti {0,1,2}
@attribute nepolozeni_ispiti {0,1,2}
@attribute prosek_po_studenti {0,1,2}
@data
2,1,1,1
1,0,2,0
```

```
2,1,1,2
0,2,1,1
2,1,1,0
1,0,2,2
2,1,0,2
0,2,2,1
1,1,1,0
```

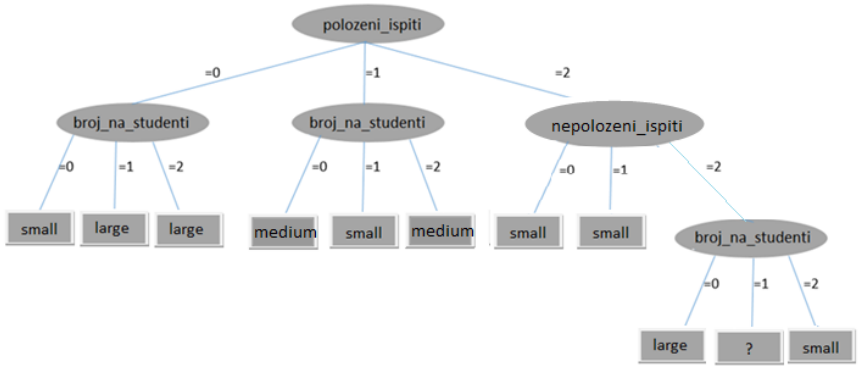
Слика 4 Атрибути на базата на податоци Student Database
Figure 4 Attributes of Student Database

Резултатите на примена на ID3, J48 и J48 graft алгоритамите врз ова множество податоци се дадени во Табела 1. Во сите случаи беше граден модел на непоткастрено дрво и резултатите се дадени со употреба на податоци за тренирање. Значи со ова множество правиме модел кој понатаму може да предвидува просек на студентите. Покрај тоа, табелата дава компаративни резултати за капа статистиката, средната апсолутна грешка, средната квадратна грешка, времето потребно да се изгради моделот, бројот на листовите, точноста и големината на дрво. Бројот на правилно и погрешно класифицирани примероци поврзани со секој од класификаторите, исто така може да се види од табелата.

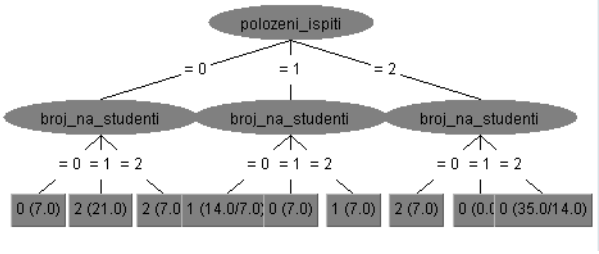
Табела 1 Обработка на податоците со ID3, J48 и J48 graft
Table 1 Processing data with ID3, J48 and J48 graft

Параметри	use training set		
	ID3	J48	J48 graft
Вкупен број на инстанци	105	105	105
Точно класифицирани инстанци	84	84	84
Неточно класифицирани инстанци	21	21	21
Kappa статистика	0,6918	0,6918	0,6918
Средна апсолутна грешка	0,1333	0,1689	0,1689
Средна квадратна грешка	0,2582	0,2906	0,2906
Време потребно за градење на моделот (во секунди)	0	0,01	0,01
Број на листови во дрвото	11	9	9
Големина на дрвото	16	13	13
Точност	80%	80%	80%

Дрвото на одлучување во WEKA го има следниов изглед (слика 5 и слика 6).



Слика 5 Дрво на одлучување (ID3)
Figure 5 Decision tree (ID3)



Слика 6 Дрво на одлучување (J48 и J48 graft)
Figure 6 Decision tree (J48 and J48 graft)

Од табелата можеме да забележиме дека сите три алгоритми вршат класификација со точност од 80 % со употреба на податоци за тренирање. J48 алгоритмите градат поминимално дрво на одлучување со 9 листови и вкупна големина 13 за разлика од ID3. Времето потребно за градење на моделот со ID3 изнесува 0 секунди. Од вкупно 105 инстанци (примероци) можеме да забележиме дека најмногу или 84 се точно класифицирани со користење на податоци за обука со сите три алгоритми.

6. Заклучок

Преку овој труд сеопфатна анализа преку различни класификатори (непоткастрени) со користење на WEKA софтверот беше спроведена на Student базата на податоци. Со помош на овие класификатори може да се предвиди некоја непозната вредност за просечна оценка, врз основа на некои вистински, познати вредности, со користење на прилично едноставни техники како што се ID3 и J48. Со користење на WEKA имаме добиено добри резултати за овој проблем. Капа статистиката со употреба на податоци за обука со сите три алгоритами изнесува 0,6918 (знаеме дека, со цел предвидувањето да се смета за важно, вредноста на Капа треба да биде поголема од 0,6). Кога ќе се земат во предвид сите параметри, доаѓаме до заклучок дека ID3 алгоритамот е најдобар за предвидување во споредба со други алгоритами, затоа што ID3 покрај другите карактеристики има и најмала средна апсолутна грешка и најмала средна релативна грешка. Значи, класификацијата на просекот со дрва на одлучување е техника која автоматски може да генерира соодветни и применливи модели од Student базата кои може да се применат во било која компјутерска средина.

Користена литература

[1] Anjan. K, Harish. R. (2013). Machine Learning with WEKA.
[2. Awadhesh, I. (2012). WEKA - IT For Business Intelligence. Term Paper , 17.
[3] Gupta, S. (2009). Term paper on Data mining - How to use Weka for data analysis.
[4] Pfahringer, B. (2013). Machine Learning with WEKA. New Zeland: University of Waikato.
[5] Zoran, P. (2007). Seminarski rad:Masinsko ucenje, inteligentni agenti. Univerzitet u Beogradu
[6] Overfitting in decision trees, available online at: http://www3.nd.edu/~rjohns15/cse40647.sp14/www/content/lectures/24%20-%20Decision%20Trees%203.pdf
[7] Reduced Error Pruning, available online at: https://www.cs.auckland.ac.nz/~pat/706_98/ln/node90.html
[8] Decision tree overfitting, available online at: http://www.saedsayad.com/decision_tree_overfitting.htm
[9] Şchiopu, D. (2007). Predicting Infractionality Rate by County Using ID3 Algorithm. BULETINUL Vol. LIX, No.1, pp. 43-50.
[10] Shweta. R, Amit. A. (2013). Designing Spam Model- Classification Analysis using Decision Trees. International Journal of Computer Applications (0975 – 8887). Volume 75– No.10.