ВЪЗМОЖНОСТИ ЗА ПРИЛОЖЕНИЕ НА WEKA ПРИ ИЗУЧАВАНЕТО НА АЛГОРИТМИ ЗА МАШИННО ОБУЧЕНИЕ*

ЕРТАН М. ГЕЛДИЕВ, НАЙДЕН В. НЕНКОВ

OPPORTUNITIES FOR APPLICATION OF WEKA IN LEARNING OF MACHINE LEARNING ALGORITHMS

ERTAN M. GELDIEV, NAYDEN V. NENKOV

ABSTRACT: *WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization (http://www.cs.waikato.ac.nz/ml/WEKA/)". The purpose of this article is to demonstrate to novice users of WEKA exemplary way to learn. The steps in this practice (2016 Jason Brownlee) could be distributed as: download and Install WEKA, load standard machine learning datasets, descriptive stats and visualization, rescale your data, perform feature selection on your data, machine learning algorithms in WEKA, estimate model performance, baseline performance on your data, classification algorithms, regression algorithms, ensemble algorithms, compare the performance of algorithms, tune algorithm parameters, save your model.*

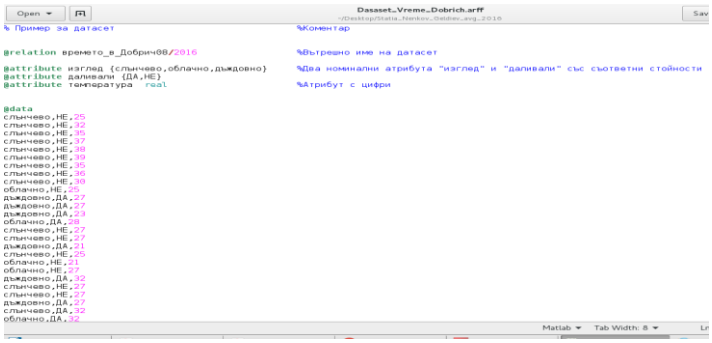
KEYWORDS: *WEKA, machine learning algorithms, estimate model performance*

ВЪВЕДЕНИЕ

Weka е софтуерна колекция от алгоритми за машинно обучение за извличане на данни задачи. За стартиране на WEKA е необходимо предварително да бъде инсталирана Java. WEKA може да се свали от сайта на университета Waikato в Нова Зеландия [8], където са достъпни версии за: MS Windows, Linux и Mac OS. Версиите за Linux, притежават система за управление на пакетите, която позволява доработка и надграждат функционалностите на WEKA. В статията е показан подход за изучаване на този инструмент, който позволява бързото му овладяване и успешното му приложение върху масиви от неструктурирани данни.

1. Използване на стандартни набори с данни (datasets) за машинно самообучение

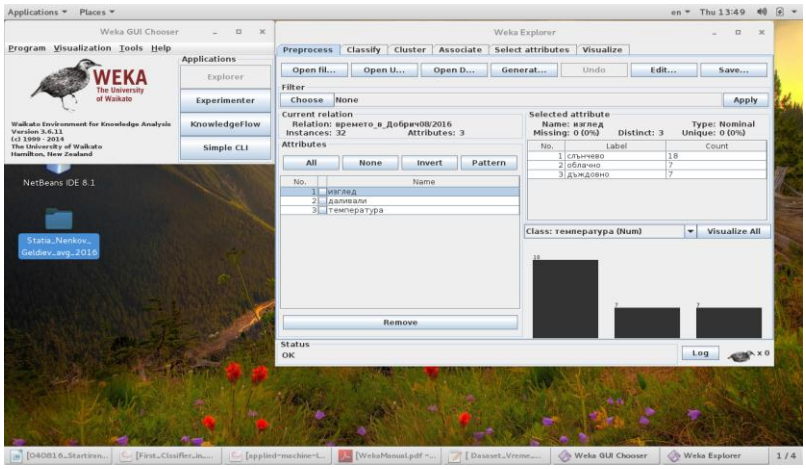
Стандартните набори с данни (datasets) са основна концепция при машинното самообучение.



Фиг. 1. Екран на WEKA със зареден метеорологичен dataset

* Настоящата статия е финансирана от Фонд „Научни изследвания" към Шуменския университет „Епископ Константин Преславски" по проект № РД-08-113/08.02.2016 г.

Те са приблизително еквивалент на двудименсионна електронна таблица или таблица от база данни. Datasets е колекция от примери, като всеки пример е инстанция на класа ***WEKA.core***. Instance. На фиг. 1 е показан такъв примерен dataset с метеорологични данни.



Фиг. 2. Начален екран на WEKA с основните функционалности

На фиг. 2 е показан екран с основните функционалности на WEKA.

2. Описателни статистики и визуализация

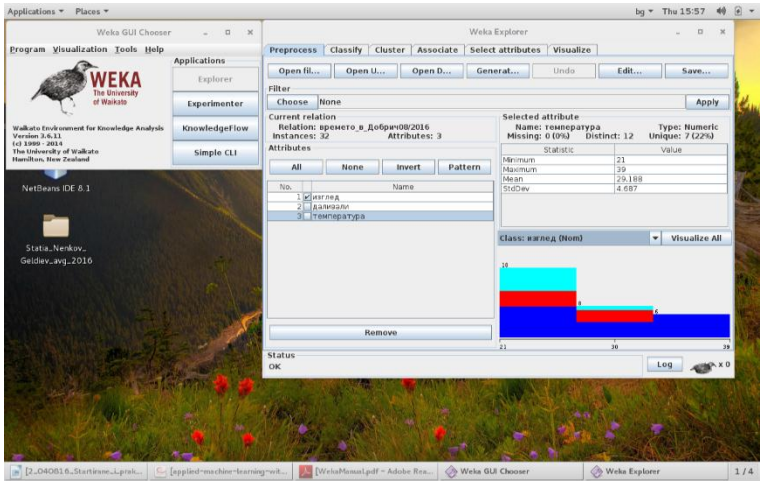
Weka показва описателни статистически данни, изчислени от вашите данни. Weka експлорера съдържа [6]:

- потребителски интерфейс
  - секционни табове
    - Предварителен процес за избор на данни и модифицирането им - препроцес
    - Класифициране – трениране и обучение с обучаващи схеми за класификация или позволяващи регресия
    - Групиране
    - Асоцииране
    - Избор на атрибути
    - Визуализация
  - Статус бокса ни съобщава за текущия процес който тече в момента
  - бутон на лог събитията
  - weka статус иконата показва броя работещи моменти едновременно
  - позволява запис към графичен файл
- Предварителна обработка (preprocessing)
  - зареждане на данни
  - текуща релация
    - Името на релацията дадена от заредената форма. Филтрите могат да модифицират името на релацията
    - Броя на инстанциите в данните (записите)
    - Броя на избраните атрибути в данните
- Класификация (classification)
- Клъстериране (clustering)
- Асоциации (associating)
- Избрани атрибути (selecting attributes)

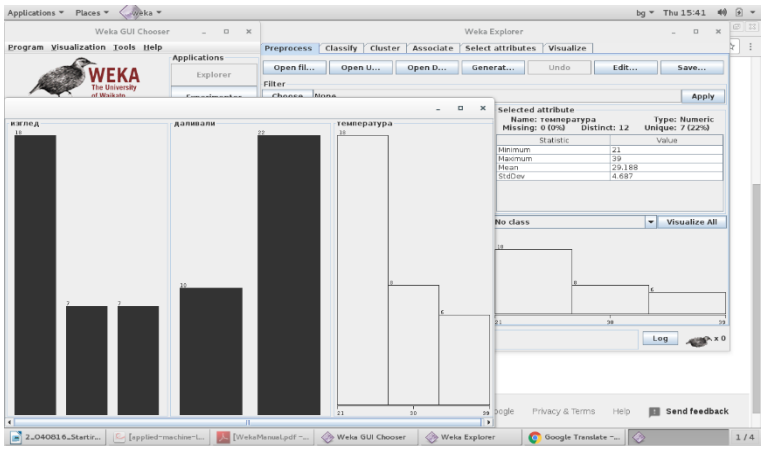
- Визуализация (visualizing)

Weka визуализира по различни начини данните. На фиг. 3 е показан графичния интерфейс с хистограма за избран атрибут като цветовете индицират всеки клас. При движение на мишката върху хистограмата се показва диапазона и броя на примерите.

На фиг. 4 по-долу е визуализирана статистика при избран атрибут “изглед”.



Фиг. 3. Графичен интерфейс на WEKA



Фиг. 4. Статистика по атрибут „изглед“

3. Нормализация и мащабиране на данните

Необработените данни понякога не са подходящи за моделиране. Можем да използваме филтри върху данните за ги преобразуваме в подходящ вид. Във WEKA филтрите се използват в предварителна обработка на данните. Те се намират в пакета: ***weka.filters***. Всеки филтър попада в следните две категории:

- с надзор (supervised) – филтърът изисква атрибута за клас да бъде избран;
- без надзор (unsupervised) – не се изисква да присъства атрибут за клас.

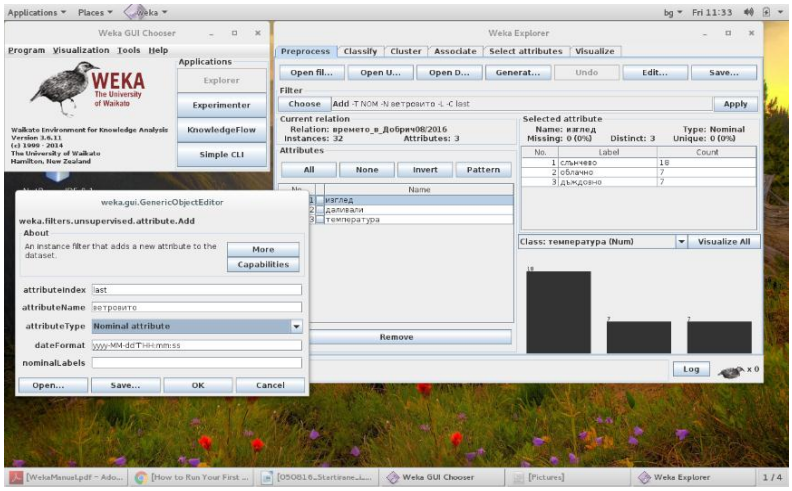
Освен това филтрите попадат и в две подкатегории:

- базирани на атрибути – колоните се обработват, добавят или премахват;
- базирани на инстанцията – редовете се обработват, добавят или премахват.

Тези категории изчистват разликата между дискретните филтри на WEKA. Филтрите с надзор поемат клас атрибута от dataset, с цел да се определи оптимален брой и размер на кошчетата, докато при филтрите без надзор броя им зависи от потребителя. Според тази

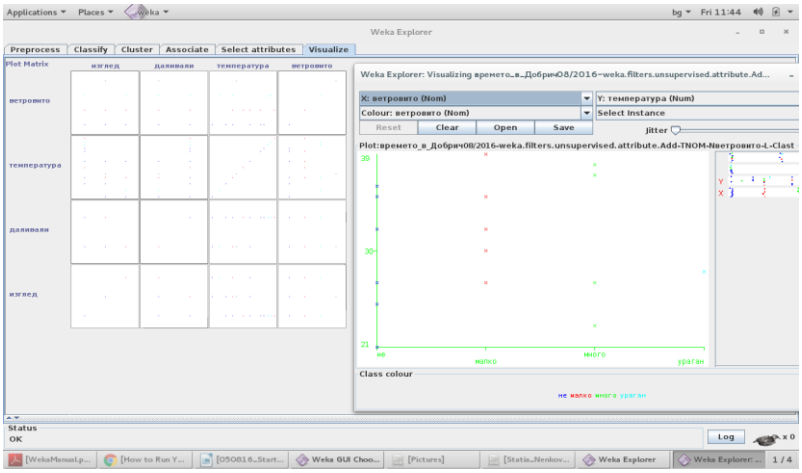
класификация филтрите са поточни или базирани (batch-based). Поточните филтри могат да обработват данните веднага. Базираните върху инстанция филтри са малко-по специални в начините по който се справят с данните. Те могат да обработват данните ред по ред след като първото количество данни премине.

Нека направим кратко упражнение като добавим филтър „add“ (фиг. 5), който добавя нов атрибут „ветровито“ без супервайзър върху атрибутите. След добавянето на филтъра ползваме бутона save в предпроцеса за да запишем новия ARFF файл.



Фиг. 5. Добавяне на нов атрибут

В новия файл с текстов редактор може да добавим номиналните стойности за атрибута „ветровито“ и отворим наново през експлорера този файл и през него се визуализазира матричния плот с атрибутите.



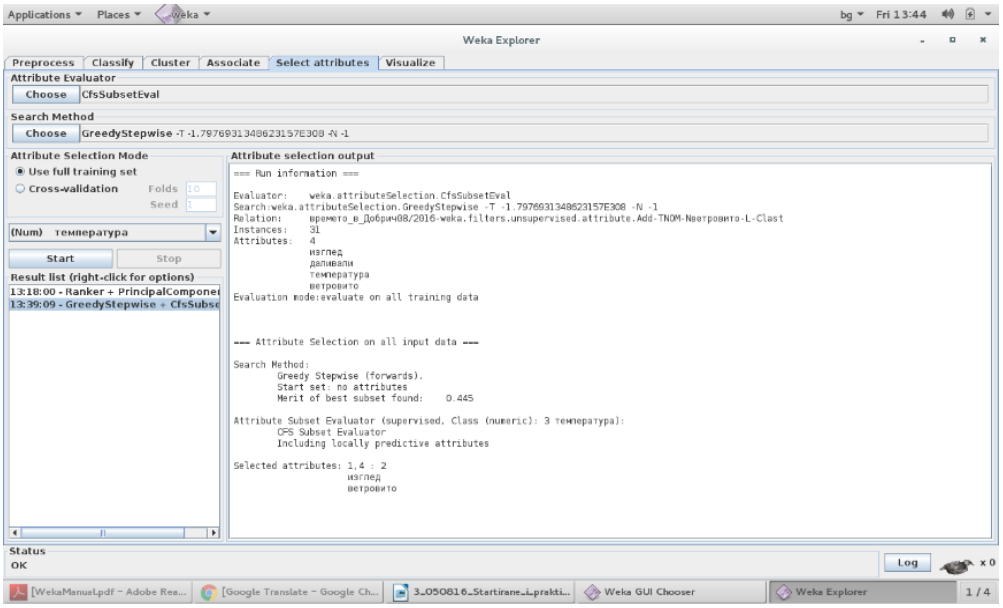
Фиг. 6. Начален екран на WEKA

При този пример не е удачно да се извърши нормализация на данни защото след изпълнението и цифровите показатели за температурата на въздуха ще придобият стойности между 0 до 1 включително.

4. Избирателна селекция на данните

Преди включването на модел трябва да се направи избирателна селекция, чрез която се избират релевантните атрибути и се премахват излишните нерелевантни атрибути. Основните причини за това са опростяване на модела, структуриране на познанието и други. При методи за селекция на данни се избира „какво се оценява“, „метод на оценка“,

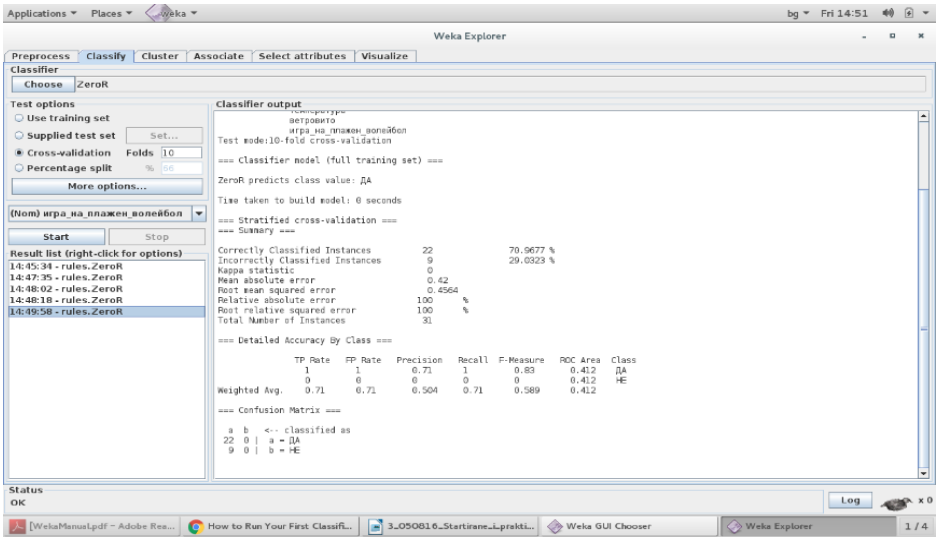
алгоритъм на обучение, филтри и т.н. В примерът се прилага филтър „CfsSubsetEval“ за избирателна селекция (фиг. 7) и се вижда, че WEKA селектира атрибутите „изглед“ и “ветровито“.



Фиг. 7. Избирателна селекция

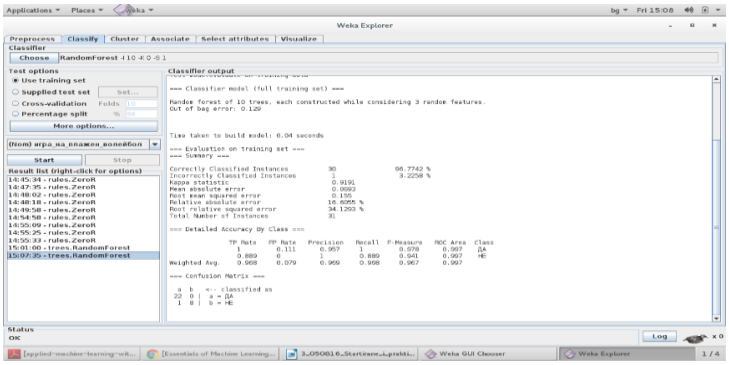
5. Алгоритми за машинно обучение

Алгоритмите за машинно обучение във WEKA са описани последователно в [8]. При отваряне на таб „Classify“ по подразбиране се отваря алгоритъма ZeroR (фиг. 8) който прилагаме върху нашите данни. ZeroR е опростен класификационен метод, който се основава на целта и игнорира всички предиктори [3, 4]. Нека за да го използваме като добавим атрибут в нашия dataset „игра_на_плажен_волейбол“ с номинали „да“ и „не“. Алгоритъмът не използва въздействието на предикторите в модела и отчита високи проценти с грешки.



Фиг. 8. Добавяне на нов атрибут

При използване на алгоритъма Random Forest (фиг. 9), който е алгоритъм за обучение свързан със предикторите, т.е. дава резултат въз основа на техните стойности В случая WEKA предсказва доста по-точно дали ще се играе волейбол.



Фиг. 9. Използване на алгоритъма Random Forest

6. Приблизителна оценка на производителност на модела

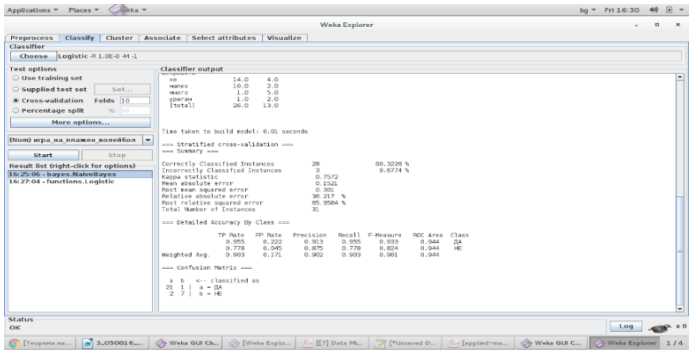
Често се налага да се сравнят две или повече обучителни схеми за един и същ проблем, т.е. да се прецени по-добрата. Оценяването на грешката става чрез валидиране на кръст (или друга подходяща оценяваща процедура) може да се повторя докато се избере схемата с най-ниска. Това е достатъчно за практически приложения. Ако една схема има по-малка оценка на грешката отколкото друга се ползва този алгоритъм. Понякога е по-важно да се определи коя схема е по-подходяща и независеща от грешките. В по-горните схеми, в които използвахме два алгоритъма се вижда, че вторият е по-подходящ заради по-малкото количество грешки.

7. Класификационни алгоритми

WEKA представя голям набор от класификационни алгоритми които се увеличават, като различни изследователски групи имплементират в нея такива [2, 6]. Тук ще се експериментира с „Логистична регресия“ и с „Naive Bayes“.

- ***Логистична регресия*** е класификационен а не регресионен алгоритъм. Използва се да оцени дискретни стойности (двоични стойности като 0/1, да/не, истина лъжа) . Базира се на независими променливи и предсказва възможността за срещане на събитие, след като постави данни в логистична функция. Тъй като се прогнозира вероятността, изходните стойности е между 0 и 1.
- ***Naive Bayes*** е класификационна техника базирана на теоремата на Бейс за настъпване на дадено събитие при известни няколко независими предиктора.

Оказа се че двата алгоритъма дават почти еднакви резултати с данните ни от предишните примери като при логистичната регресия грешките са около 23 процента докато при Naive Base около 36% (фиг. 10).



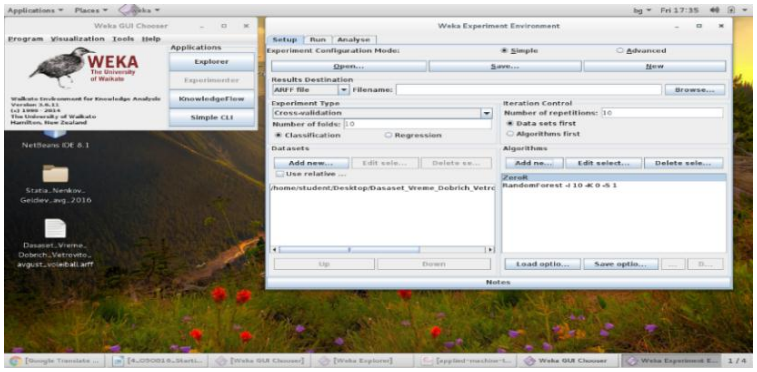
Фиг. 10. Оценка на грешката

8. Регресионни алгоритми

Регресията е както е известно, предсказване на количествени стойности за разлика от класификацията където се определят категории. В статията няма да се покажат примери с такива алгоритми поради това, че dataset не е подходящ за такъв анализ.

9. Смесени алгоритми

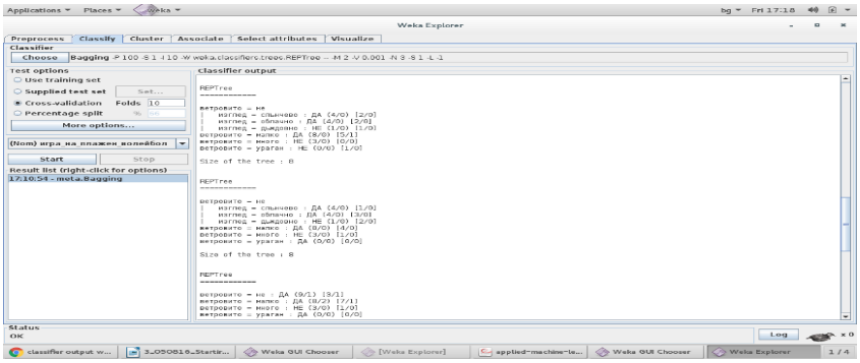
Bagging е алгоритъм за машинно обучение създаден да подобри стабилността и точността и използва статистически класификации и регресия. Той често намалява вариантите и помага да се избегне нагаждането на данни. Той често се добавя към методите за претърсване на дървета. ***Random Forests*** е смесен метод за машинно обучение (търсещ най-близкия съсед предиктор) за класификация и регресия и конструира дървета на решенията. Други смесени алгоритми (но не всички) са ***Ada Boost, metaAda Boost***, ***Voting, metaVotin***, ***Stacking meta.Stacking***. Като направихме анализ например чрез meta. Bagging (фиг. 11) WEKA със *dataset* в статията направи детайлна статистическа справка със сходни групи с данни.



Фиг. 11. Екран на WEKA за алгоритъма *Bagging*

10. Сравнение на производителност на алгоритми

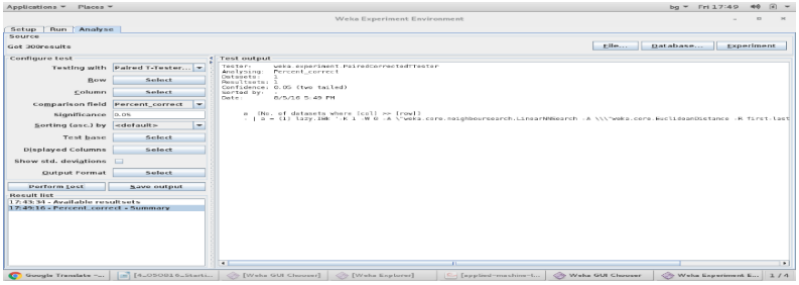
WEKA предлага различен инструментариум за сравнение на алгоритми с име WEKA експериментална среда (фиг. 12). Тази среда позволява да се проектират и изпълняват експерименти със алгоритми за машинно обучение и после да се анализират резултатите.



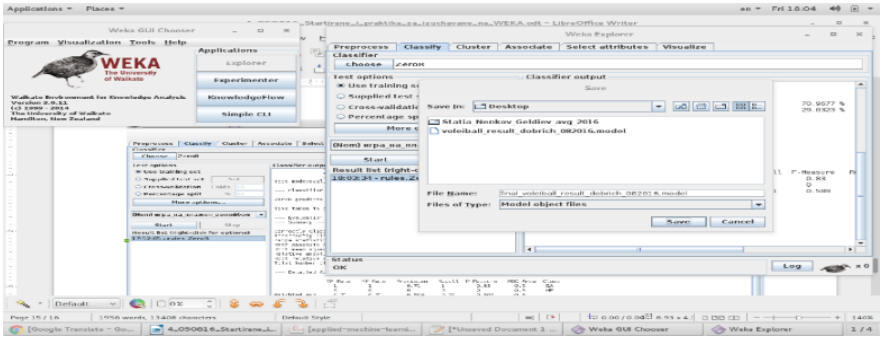
Фиг. 12. Инструменти за сравняване на WEKA

11. Настройки параметрите на алгоритмите и запазване на модела

WEKA като експериментална среда позволява да проектирате контролирани експерименти и сравните резултатите от различни алгоритми и дали различията са статистически значими. На фиг. 13 е показано примерно сравняване на параметрите при ***k-nearest.*** алгоритъма.



Фиг. 13. Пример за сравняване на параметри при алгоритъма k-nearest



Фиг. 14. Съхраняване на модел във WEKA

Съхраняването на модела става по начина, показан на фиг. 14. След това той може да се ползва за предсказване и с други datasets.

ЗАКЛЮЧЕНИЕ

От разгледаните примери за използване на WEKA се вижда, че зад привидно опростения интерфейс стоят мощни възможности за анализиране на големи datasets с данни. Инструмента разполага с множеството алгоритми за анализ и предсказване. Предоставят се и готови модели, които улесняват тези процеси. Възможностите на продукта могат да се разширяват, тъй като се допуска добавянето на скриптове на езиците: Java, Python и други, които се избират според конкретните специализирани задачи [1].

Средата е удобна за обучаване на както за начинаещи така и за напредвали специалисти по анализ на данни, включително и студенти в университет. Необходимостта от такива експерти все повече нараства с развитие на ИТ технологии, свързани с IoT (Inherent of Thinks), интелигентните системи, невронните мрежи [7], bioinformatics и други генериращи големи данни направление [5].

ЛИТЕРАТУРА

1. Ненков, Н. В, Илхан Истикбал Ибрям, Севгинар Мехмед Вели. Използване възможностите на софтуер с отворен код за анализ на данни от онлайн решени тестове. сп. Образование и технологии, бр.5 2014, ГОДИНА V, ISSN 1314–1791, стр.371-377
2. Ian H. Witten Eibe Frank Mark A. Hall: Data Mining Practical Machine Learning Tools and Techniques": Third Edition, 2011.
3. Machine Learning Mastery with Weka Analyze Data, Develop Models and Work Through Projects, 2014.
4. Mark Hall, Ian Witten и Eibe Frank, Data Mining:Practical Machine Learning Tools and Techniques, 2011, ISBN: 978-0-12-374856-0.
5. Nenkov, N. V., Elica Spasova, Implementation of a neural network using simulator and Petri net, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, No. 1, 2016, (DOI): 10.14569/IJACSA.2016.070155
6. Remco R. Bouckaert,Eibe Frank,Mark Hall,Richard Kirkby,Peter Reutemann,Alex Seewald,David Scuse WEKA Manual for Version 3-6-11 2014 - May 3, 2014
7. Zdravkova, Elitsa., Nenkov, N.V., Research of neural network simulators through two training data sets, (On-line: www.cmnt.lv), ISSN 1407-5806, ISSN 1407-5814, Transport and Telecommunication Institute, Riga, Latvia, 2016 20(1) 12-15.
8. Remco R. Bouckaert,,Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, David Scuse, WEKA Manual for Version 3-7-13, September 10, 2015.
URL: https://sourceforge.net/projects/weka/files/documentation/3.7.x/WekaManual-3-7-13.pdf/download