5. Rosin P.L., West G.A. Nonparametric Segmentation of Curves Into Various Representations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1995, no.12, pp. 1140-1153. doi: 10.1109/34.476507
6. Sampson P.D. Fitting Conic Sections to Very Scattered Data: An Iterative Refinement of the Bookstein Algorithm. *Computer Graphics and Image Processing*, 1982, no.18, pp. 97-108. doi:10.1016/0146-664X(82)90101-0
7. Taubin G. Estimation of Planar Curves, Surfaces and NonPlanar Space Curves Defined by Implicit Equations, With Applications to Edge and Range Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991, no.11, pp. 1115-1138. doi: 10.1109/34.103273.
8. Diyazitdinov R.R. Otcenivanie parametrov pologeniya kontura krivoy v profilnoy sisteme [Rate position's parameters of curve for profile sensor]. *Infokommunikacionnye tehnologii*, 2014 no.2, pp. 70-73.
9. Vasin N.N., Kurinskiy V.Y. Rasshirenie funktcionalinih vozmognostey sistem videonablyudeniya [Improving features video surveillance]. *Infokommunikacionnye tehnologii*, 2013, no.6, pp. 63-66.
10. GOST R 50864-96 *Rezba konicheskaya zamkovaya dlya elementov burilnih kolonn* [State Standard R 50864-96. Conus bolted joint for elememts drill pipes]. Moscow, Standartinform Publ., 1996. 13 p.

Received 23.10.2015

УДК 004.8

ПРИМЕНЕНИЕ МЕТОДА ДЕРЕВА РЕШЕНИЙ В ЗАДАЧАХ КЛАССИФИКАЦИИ И ПРОГНОЗИРОВАНИЯ

Мифтахова А.А.
Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ
E-mail: miftaxovaa@mail.ru

В статье рассматривается и описывается один из алгоритмов Data Mining, предназначенных для решения задач классификации и прогнозирования, – метод деревьев решений (decision trees). Представлен процесс построения дерева решений для решения задачи классификации сотрудников магазина в виде ручного построения, а также с помощью языка объектно-ориентированного программирования Python. Приведен пример построения дерева решений для решения задачи классификации стандартного набора данных ирисов Фишера. Для этого примера приведены не только построение дерева вручную и с помощью Python, а также показана реализация деревьев решений в разных программных системах.

Ключевые слова: дерево решений, атрибут, энтропия, прирост информации, Phyton, Deductor, Orange Canvas.

Введение

Искусственный интеллект является обширной областью науки, алгоритмы которой используются при решении задач, для которых часто сложно и невозможно создать явный алгоритм решения. В настоящее время известно достаточно много алгоритмов, предназначенных для решения задач классификации или прогнозирования: метод опорных векторов, метод *k* ближайших соседей, нейронные сети и деревья решений [1].

Деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение [2]. На ребрах дерева записываются атрибуты, от которых зависит целевая функция. Данная статья посвящена одному из классических методов интеллектуального анализа данных– построению деревьев решений. Наиболее общее определение дерева решений – это средство поддержки принятия решений при прогнозировании, которое широко применяется в статистике и анализе данных.

Цель процесса построения дерева принятия решений – создать модель, по которой можно было бы классифицировать случаи и решать, какие значения может принимать целевая функция, имея на входе несколько переменных.

Применения метода дерева решений для решения задач классификации и прогнозирования

Приведем пример построения дерева решений, решив задачу классификации сотрудников магазина. В качестве исходных данных выберем небольшой набор данных – сотрудники магазина, представленный в таблице 1.

В качестве целевой переменной возьмем переменную status; 5 записей из 11 целевая переменная имеет значение senior, а оставшиеся 6 записей – junior.

Таблица 1. Сотрудники магазина

department	status	age, years	salary, thousand rubles
sales	senior	31	46
sales	junior	26	28
sales	junior	31	33
marketing	senior	36	46
marketing	junior	31	41
systems	senior	31	66
systems	junior	26	46
systems	senior	41	66
secretary	senior	46	36
secretary	junior	26	28

Энтропия исходного множества до разбиения составит [3]:

$$Info\ (N) = -\sum_j p_j \log_2 p_j = -(5/11)\log_2(5/11) -$$
$$-(6/11)\log_2(6/11) = 1{,}023 \text{ бит.}$$

Произведем разбиение по атрибуту department. Три записи данного атрибута имеют значение sales, четыре – systems, два – marketing, два – secretary. Соответствующие вероятности будут равны:

$$P_{sales} = 3/11;\ P_{systems} = 4/11;$$
$$P_{marketing} = 2/11;\ P_{secretary} = 2/11.$$

Одна из трех записей, содержащих значение sales, указывает на senior, а две – на junior. Тогда при случайном выборе из этих трех записей вероятность появления senior составит 1/3, а junior – 2/3. Вычислим энтропию для значения sales:

$$Info_{sales}(N) = -\sum_j p_j \log_2 p_j = -(1/3)\log_2(1/3) -$$
$$-(2/3)\log_2(2/3) = 1{,}35 \text{ бит.}$$

Две записи из четырех, содержащих значение systems, указывают на senior, а две – на junior. Вычислим энтропию:

$$Info_{systems}(N) = -\sum_j p_j \log_2 p_j =$$
$$= -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1 \text{ бит.}$$

Одна из двух записей, содержащих значение marketing, указывает на senior, а другая – на junior. Вычислим энтропию:

$$Info_{marketing}(N) = -\sum_j p_j \log_2 p_j =$$
$$= -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1 \text{ бит.}$$

Одна из двух записей, содержащих значение secretary, указывает на senior, а другая – на junior. Вычислим энтропию:

$$Info_{secretary:}(N) = -\sum_j p_j \log_2 p_j =$$
$$= -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1 \text{ бит.}$$

На основе полученных данных можно рассчитать полную энтропию разбиения:

$$Info_{department}(N) = (3/11)\times 1{,}35 + (4/11)\times 1 +$$
$$+ (2/11)\times 1 + (2/11)\times 1 = 1{,}095 \text{ бит.}$$

Прирост информации, полученный в результате разбиения по атрибуту department, будет равен:

$$Gain_{department}(N) = Info(N) -$$
$$- Info_{department}(N) = 1{,}023 - 1{,}095 = -0{,}072 \text{ бит.}$$

Аналогичным образом находим и получаем прирост информации по остальным атрибутам:

$$Gain_{age}(N) = Info(N) - Info_{age}(N) = 0{,}659 \text{ бит};$$
$$Gain_{salary}(N) = Info(N) - Info_{salary}(N) =$$
$$= 0{,}658 \text{ бит.}$$

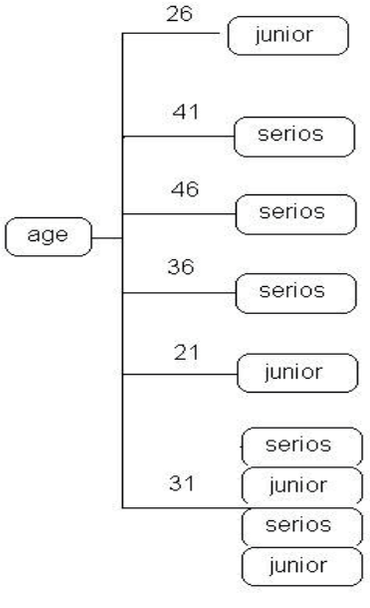

Рис. 1. Результат первого шага построения дерева

Таким образом, прирост информации в результате разбиения по атрибуту age больше по сравнению с другими атрибутами, поэтому выбирается в качестве начального разбиения в корневом узле дерева. Схема начального разбиения представлена на рис. 1.

Для значения 31 год получен узел, содержащий две записи со значением целевой переменно senior и две со значением junior. Далее производим поиск оптимального разбиения данного подмножества (подмножество $N1$) данного узла.

Таблица 2. Множество $N1$

department	status	age, years	salary, thousand rubles
sales	senior	31	46
sales	junior	31	33
systems	senior	31	66
marketing	junior	31	41

В качестве целевой переменной возьмем переменную status. В двух записях из четырех целевая переменная принимает значение senior и в двух – junior. Поэтому энтропия исходного множества до разбиения составит:

$$Info(N1) = -\sum_j p_j \log_2 p_j =$$

$$= -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1 \text{ бит}.$$

Находим прирост информации по каждому нецелевому атрибуту:

$$Gain_{department}(N1) = Info(N1) - Info_{department}(N1) =$$

$$= 0{,}5 \text{ бит};$$

$$Gain_{salary}(N1) = Info(N1) - Info_{salary}(N1) =$$

$$= 1 - 0 = 1 \text{ бит}.$$

Разбиение по атрибуту salary обеспечивает наибольший прирост информации, поэтому выбирается в качестве дальнейшего разбиения дерева. Полное дерево, полученное в результате разбиения, представлена на рис. 2 [4].

Распространенный способ реализации деревьев решений – это построение дерева на языке программирования Python. Чтобы оценить, насколько хорош выбранный атрибут, алгоритм сначала вычисляет энтропию всей группы. Затем он пытается разбить группу по возможным значениям каждого атрибута и вычисляет энтропию двух новых групп.

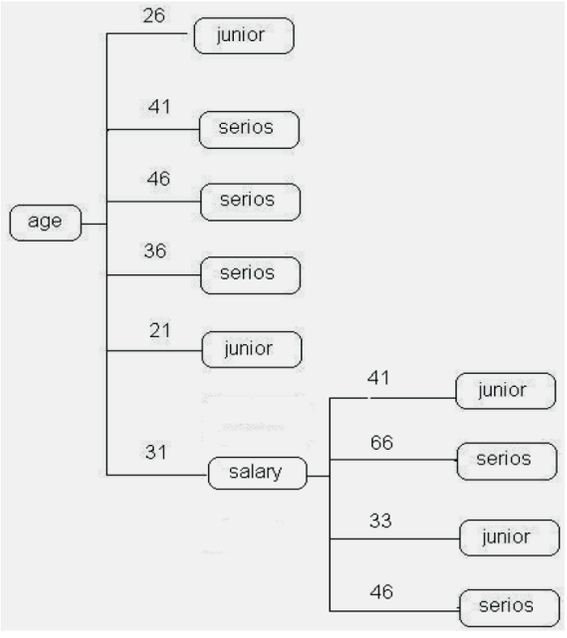

Рис. 2. Полное дерево решений

Для определения того, какой атрибут дает наилучшее разбиение, вычисляется информационный выигрыш, то есть разность между текущей энтропией и средневзвешенной энтропией двух новых групп. Он вычисляется для каждого атрибута, после чего выбирается тот, для которого информационный выигрыш максимален. Вычисляя для каждого узла наилучший атрибут и расщепляя ветви, алгоритм создает дерево [5].

На рис. 3 представлен результат построения дерева решений с помощью интерпретатора языка программирования Python 2.7.6 [6].

```
>>> ================================ RESTART
>>>
['department', 'status', 'age', 'salary']
age
salary
-------------------------
--    Decision tree    --
-------------------------

age
        26
                ->      junior
        21
                ->      junior
        46
                ->      senior
        31
        salary
                46
                        ->      senior
                33
                        ->      junior
                66
                        ->      senior
                41
                        ->      junior
        36
                ->      senior
        41
                ->      senior
>>>
```

Рис. 3. Результат построения дерева решений с помощью Python 2.7

Рассмотрим следующий пример построения дерева решений. В качестве исходных данных выберем классический набор данных – ирисы Фишера. В отличие от предыдущего примера данный набор содержит 150 экземпляров ирисов, принадлежащих к трем видам (setosa, versicolor, virginica). Для каждого экземпляра ириса известны четыре величины: длина и ширина чашелистика, длина и ширина лепестка [7]. Наша задача выработать критерии, по которым можно различить три вида. В качестве целевой переменной возьмем переменную Class; 50 записей из 150 принадлежат атрибуту Iris-setosa, 50 записей – Iris-versicolor, 50 записей – Iris-verginica. Энтропия исходного множества до разбиения составит:

$$Info(N) = -\sum_j p_j \log_2 p_j =$$

$$= -(50/150)\log_2(50/150) - (50/150)\log_2(50/150) -$$

$$-(50/150)\log_2(50/150) = 0{,}477 \text{ бит.}$$

Прирост информации по атрибутам:

$$Gain_{s\text{-}length}(N) = Info(N) - Info_{s\text{-}length}(N) =$$

$$= 0{,}477 - 1{,}383 = -0{,}906 \text{ бит};$$

$$Gain_{s\text{-}width}(N) = Info(N) - Info_{s\text{-}width}(N) =$$

$$= 0{,}477 - 1{,}506 = -1{,}029 \text{ бит};$$

$$Gain_{p\text{-}length}(N) = Info(N) - Info_{p\text{-}length}(N) =$$

$$= 0{,}477 - 0{,}039 = 0{,}438 \text{ бит};$$

$$Gain_{p\text{-}width}(N) = Info(N) - Info_{p\text{-}width}(N) =$$

$$= 0{,}477 - 0{,}284 = 0{,}193 \text{ бит.}$$

Прирост информации в результате разбиения по атрибуту petal-length больше по сравнению с другими атрибутами, поэтому выбирается в качестве начального разбиения в корневом узле дерева.

После этого производим дальнейшее разбиение, пока не получим оптимальное разбиение полного множества. Полное дерево, полученное в результате разбиения, представлено на рис. 4.

Чем больше набор данных содержит записей, тем более сложным и трудоемким становится процесс построения дерева решений вручную. Он занимает огромное количество времени, и вероятность ошибок возрастает. Для решения данной проблемы существуют программные продукты, предназначенные для анализа данных и содержащие алгоритм построения деревьев решений.

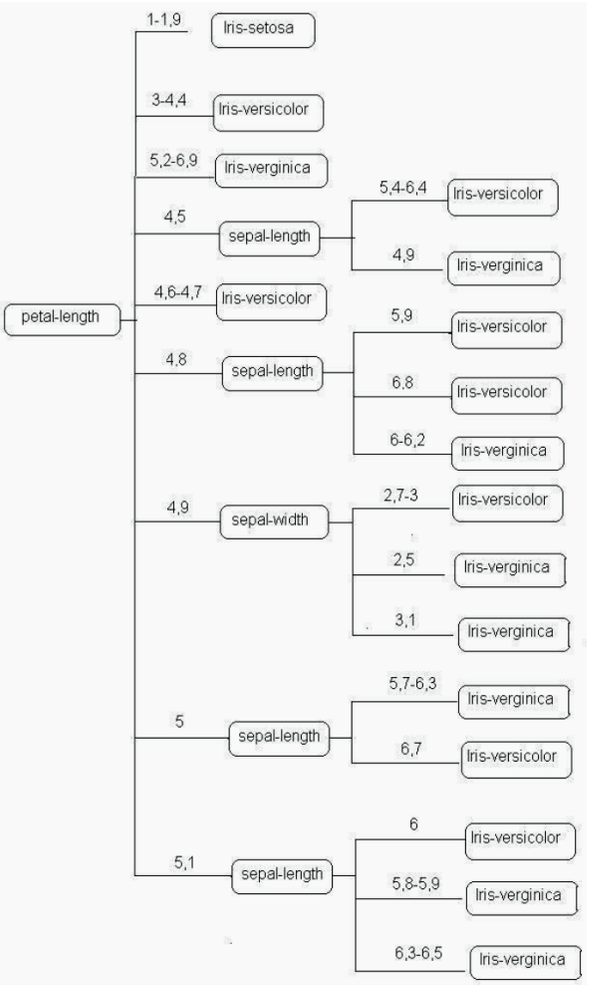

Рис. 4. Полное дерево решений

Одними из самых распространенных программ на сегодняшний день являются Deductor[8] и Orange Canvas [9]. На рис. 6 представлен результат построения дерева решений в Deductor. На рис. 7 представлен результат построения дерева решений в Orange Canvas [10].

Как видно из рис. 6-7, результаты, полученные с помощью программ Deductor и Orange Canvas, соответствуют результатам формульных вычислений. Поэтому можно сделать вывод о том, что задачи (наборы данных) с относительно небольшим числом атрибутов могут решаться вручную, в то время как задачи (наборы данных) с большим числом атрибутов легче и целесообразнее решать с помощью программных систем.

Чем больше набор данных, тем дольше и сложнее становится расчет по формулам. Могут затрачиваться часы, дни и даже месяцы, а программы производит расчеты в течение нескольких секунд. Программа сама отсекает несущественные факторы, выявляет степень влияния тех или иных факторов на результат, а также выдает информацию о достоверности правил дерева

Условие	Следствие	Поддержка	Достоверность
ЕСЛИ		142	49
petal_length < 2,45	Iris-setosa	45	45
petal_length >= 2,45		97	49
petal_width < 1,75		53	48
petal_length < 4,95	Iris-versicolor	47	46
petal_length >= 4,95		6	4
petal_width < 1,55	Iris-virginica	3	3
petal_width >= 1,55	Iris-versicolor	3	2
petal_width >= 1,75	Iris-virginica	44	43

Рис. 6. Результат построения дерева решений в Deductor

решений. Кроме того, полученные результаты, представленные в программах, являются более простыми для восприятия и понимания.

Результаты, полученные с помощью Orange Canvas и Deductor, обладают примерно равными возможностями. Однако работа с деревьями решений в Deductor реализована заметно удобнее. Программа имеет несколько визуализаторов дерева решений.

Пользователь может выбрать наиболее удобный для понимания. Один из визуализаторов Deductor – правила «если – то» – удобное представление построенного дерева в виде правил.

Заключение

В статье рассмотрены несколько вариантов реализации алгоритма деревьев решений на конкретных примерах решения задач.

Качество работы метода деревьев решений зависит как от выбора метода, так и от набора исследуемых данных.

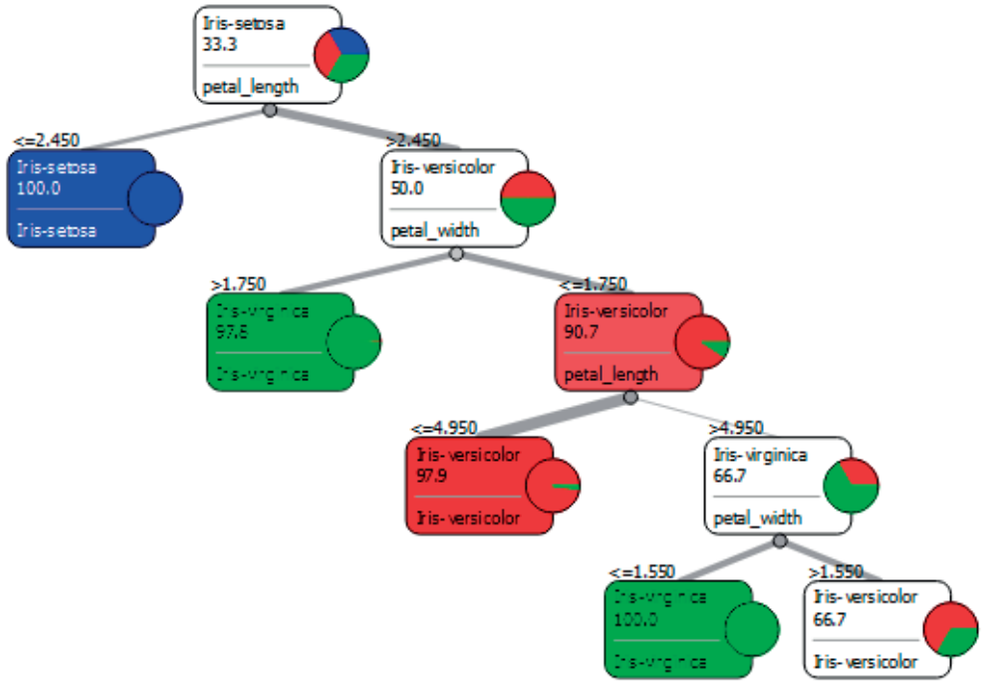

Рис. 7. Результат построения дерева решений в Orange Canvas

Литература

1. Васильев В.И., Шарабыров И.В. Обнаружение атак в локальных беспроводных сетях на основе интеллектуального анализа данных // Известия ЮФУ. Технические науки. №2(151), 2014. – С.57-67.
2. Миронов С. Современные методы анализа данных // URL: http://old.ci.ru/inform05_02/p_04.htm (дата обращения 10.08.2015).
3. Методы и средства анализа данных // URL: http://bourabai.ru/tpoi/analysis.htm#.D0.90.D0.BB.D0.B3.D0.BE.D1.80.D0.B8.D1.82.D0.BC_C4.5 (дата обращения 10.08.2015).

Таблица 3. Достоверность работы для разных методов

№	Условие	Следствие	Достоверность, %		
			Deductor	Orange Canvas	Ручной расчет
1	petal_length < 2,45	Iris-setosa	100	100	100
2	petal_length >= 2,45 petal_width < 1,75 petal_length < 4,95	Iris-versicolor	97,9	97,9	97
3	petal_length >= 2,45 petal_width < 1,75 petal_length >= 4,95 petal_width < 1,55	Iris-virginica	100	100	98
4	petal_length >= 2,45 petal_width < 1,75 petal_length >= 4,95 petal_width >= 1,55	Iris-versicolor	66,6	66,6	66
5	petal_length >= 2,45 petal_width >= 1,55 petal_width < 1,75	Iris-virginica	97,7	97,8	97,5

4. Мифтахова А.А. Реализация алгоритма с 4.5 интеллектуального анализа данных, основанного на деревьях решений // Труды 12 МНПК «Проблемы теории и практики современной науки». Нефтекамск, 2015. – С. 113-120.
5. Сегаран Т. Программируем коллективный разум. Пер. с англ. СПб: Символ-Плюс, 2008. – 368 с.
6. Python 2.7.6 Release // URL: https://www.python.org/download/releases/2.7.6/ (дата обращения 11.08.2015).
7. Бэстенс Д.Э., Ван Ден Берг В.М., Вуд Д. Нейронные сети и финансовые рынки: принятие решений в торговых операциях. – М.: ТВП, 1997. – 236 с.
8. Deductor. Продвинутая аналитика без программирования // URL: http:// basegroup.ru/deductor/description (дата обращения 11.08.2015).
9. Orange.Data Mining - Fruitful and Fun // URL: http://orange.biolab.si/ (дата обращения 11.08.2015).
10. Пальмов С.В., Мифтахова А.А. Реализация деревьев решений в различных аналитических системах // Перспективы науки. №1(64), 2015. – С. 81-87.

Поступило 20.08.2015

Мифтахова Альфия Асхатовна, аспирант Кафедры информационных систем и технологий Поволжского государственного университета телекоммуникаций и информатики. Тел. 8-937-795-01-95.э E-mail: miftaxovaa@mail.ru

APPLICATION OF DECISION TREE METHOD TO CLASSIFICATION AND PREDICTION PROBLEMS

Miftakhova A.A.
Povolzhskiy State University of Telecommunication and Informatics, 23 L.Tolstoy str., Samara 443010, Russian Federation
E-mail: miftaxovaa@mail.ru

Nowadays intellectualization of methods for data processing and data analysis is modern rapidly developing application known as Data Mining. This work is concerned with description of one of the Data Mining algorithm designed for solution of classification and prediction problems based on decision tree method. This method is also known as decision rule tree method or classification and regression tree method. The main feature of Data Mining is a combination of extended mathematical tools and novel achievements in the information technologies together with new hardware and software opportunities. The most methods were developed

within to artificial intelligence theory. This work describes decision tree for solution classification problem of store employees under hand-building and by object-oriented programming language Python. We considered an example of decision tree for solution of Iris-Fisher data set classification problem, described hand-build tree and tree build by Python, and concern with implementation of decision trees over different software systems.

Keywords: decision tree, attribute, entropy, information gain, Phyton, Deductor, Orange Canvas.

DOI: 10.18469/ikt.2016.14.1.10

Miftakhova Alfiya Ashatovna, Povolzhsky State University of Telecommunications and Informatics, 23 Lev Tolstoy str., Samara 443010, Russian Federation; PhD Student of the Department of Information Systems and Technologies. Tel.: +79377950195. E-mail: miftaxovaaa@mail.ru.

References

1. Vasil'ev V.I., SHarabyrov I.V. Obnaruzhenie atak v lokal'nykh besprovodnykh setyakh na osnove intellektual'nogo analiza dannykh [Detection of attacks on local wireless networks based on data mining]. *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki,* 2014, vol. 151, no. 2, pp. 57-67.
2. Mironov S. *Sovremennye metody analiza dannykh* [Modern methods of data analysis]. Available at: http://old.ci.ru/inform05_02/p_04.htm (accessed 10.08.2015)
3. *Metody i sredstva analiza dannykh* [Methods and tools for data mining] Available at: http://bourabai.ru/tpoi/analysis.htm#.D0.90.D0.BB.D0.B3.D0.BE.D1.80.D0.B8.D1.82.D0.BC_C4 (accessed 10.08.2015)
4. Miftakhova A.A. Realizatsiya algoritma s 4.5 intellektual'nogo analiza dannykh, osnovannogo na derev'yakh reshenij [Implementation of algorithm 4.5 data mining based on decision trees]. *Trudy 12 mezhdunarodnoj nauchno-prakticheskoj konferentsii «Problemy teorii i praktiki sovremennoj nauki»* [Proc. 12th Int scientific and practical conference «Problems of the theory and practice of modern science»], Neftekamsk, 2015, pp. 113-120.
5. Segaran T. *Programmiruem kollektivnij razum* [Programmable collective intelligence]. SPb, Simvol-Plus Publ., 2008. 368 p.
6. Python 2.7.6 Release. Available at: https://www.python.org/download/releases/2.7.6/ (accessed 11.08.2015)
7. Behstens D.EH., Van Den Berg V.M., Vud D. *Nejronnye seti i finansovye rynki: prinyatie reshenij v torgovykh operatsiyakh* [Neural networks and financial markets: decision-making in trade]. Moskov, TBP Publ., 1997. 236 p.
8. Deductor. Prodvinutaya analitika bez programmirovaniya. Available at: http://basegroup.ru/deductor/description (accessed 11.08.2015). (In Russ.)
9. Orange.Data Mining - Fruitful and Fun. Available at: http://orange.biolab.si/ (accessed 11.08.2015).
10. Pal'mov S.V., Miftakhova A.A. Realizatsiya derev'ev reshenij v razlichnykh analiticheskikh sistemakh [Implementation of decision trees in a variety of analytical systems]. Perspektivy nauki, 2015, vol. 64, no. 1, pp. 81-87.

Received 20.08.2015

ТЕХНОЛОГИИ РАДИОСВЯЗИ, РАДИОВЕЩАНИЯ И ТЕЛЕВИДЕНИЯ

УДК 621.397.2.037.372

ПРОБЛЕМЫ ОРГАНИЗАЦИИ ВЕЩАНИЯ В СТАНДАРТЕ DVB-T2 СО ВСТАВКОЙ РЕГИОНАЛЬНОГО КОНТЕНТА

Карякин В.Л., Карякин Д.В., Морозова Л.А.
Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ.
E-mail: vl@karyakin.ru

Представлен анализ методов организации вещания первого мультиплекса в стандарте DVB-T2 со вставкой регионального контента в различных вариантах построения одночастотных сетей SFN цифрового наземного вещания РФ. Отмечаются проблемы импортозамещения технологии распределенной модификации программ с использованием реплейсера, поскольку компания Enensys Technologies владеет Российским патентом на способ вещания DVB-T2 со вставкой регионального контента и устройство, используемое в этом способе. Недостатком применяемых в РФ технических решений по реализации задачи доставки региональной версии первого мультиплекса является необходимость вещания совмещенных потоков T2-MI в различных регионах с едиными параметрами, устанавливаемыми в федеральном центре мультиплексирования (ФЦФМ). Отсутствие возможностей выбирать