A personalized classification of employment offers using data mining methods

Cristina Ioana Muntean, Darie Moldovan and Ovidiu Veres

***Abstract*—** In this paper we describe a method designed for filtering the information available on job portals, according to users' preferences. We started by collecting the information needed crawling the specialized web sites, in order to build the necessary dataset. Testing two classification algorithms (Naïve Bayes and J48) we found that the second obtained higher performances, thus we concluded to be a good candidate for this type of classification and suggested possible applications for users of such web sites.

***Keywords*—** Data mining, Information retrieval, J48, Labor market, Naïve Bayes

I. Introduction

THE labor markets worldwide became more and more competitive in the last years due to increasing unemployment. The technological advances and the increased access of the population to the internet determined the birth of new search methods. The traditional search behavior of the workers changed accordingly, migrating from the print media and employment agencies to the specialized job search websites [2].

In Romania, the unemployment rate in January 2010 was 8.1%, according to the National Agency for Workforce Employment [3]. The rate rose 0.3% from the previous month and 3.2% from January 2009. An increased traffic on the most job search websites was observed, workers and employers finding the virtual space very appropriate for their interests.

Even if the Internet search turns out to be a very useful tool for job searching, could the simple *look-and-chose* system be improved? How could the user benefit from the large amount of job ads available, but instead of having to process and search all, he would only have to look at the ones that better suit his interests? If the allocated time for the search is an important factor, in which way can it be reduced to a minimum, but without missing opportunities?

Some studies regarding the Romanian labor market [16] motivated us in developing our research, with the intention to find an efficient solution that brings the labor market actors closer, regardless of the fast changing pace in this field.

In the paper we intend to propose a model for automatically search the web for job ads, considering some user preferred criteria. The system built collects the data from a specialized job search website, using Data mining methods it rates employment ads and extracts the relevant ones, in order to offer them to the final user, in accordance with his/her preferences.

To perform the research we employ web data retrieval and extraction mechanisms, and a powerful tool (Weka) containing machine learning algorithms for data mining, developed at the University of Waikato, New Zealand [4]. The method succeeds in refining a large amount of data and extracting relevant and accurate information. The rating of the ads is made considering the user preferences and changes dynamically by incorporating the new data added on the website.

The paper develops as follows. In section 2 we brief the related work in the field of job search analysis. Section 3 presents the methodology adopted for the data mining process. Section 4 presents the experiments made and the results obtained, while section 5 concludes the paper.

II. Related work and research

In our research, we took into consideration the social and economical aspects that appear on the labor market in the past few years, in order to be able to integrate them into the more technical features of the paper like the expert function for scoring. Even if in [5] the authors make an analysis on individual search versus public employment services usually provided by the state, the dynamics for the Romanian market can be noticed empirically, the use of Internet and job search sites is one of the most popular methods for finding employment ads. The research resulted in the conclusion that public employment agencies are preferred due to cost and time considerations from a job seeker's point of view. At the same time, they are proven less effective, but the rate of conversion of a job application into an actual job is higher through this system.

This work was supported by *CNCSIS IDEI 1598 Grant "* Higher education and the labor market. IT-based researches on the correlation between the qualifications required by the labor market and real knowledge of the students" and *POSDRU/88/1.5/S/60185* project "Innovative Doctoral Studies in a Knowledge Based Society", co-financed by the European Social Fund and Babeş-Bolyai University.

C. Muntean is a PhD Candidate at the Faculty of Economics and Business Administration, Babes-Bolyai University, Cluj-Napoca, Romania (e-mail: cristina.muntean@econ.ubbcluj.ro).

D. Moldovan is a PhD Candidate at the Faculty of Economics and Business Administration, Babes-Bolyai University, Cluj-Napoca, Romania (e-mail: darie.moldovan@econ.ubbcluj.ro).

O. Veres is a PhD Candidate at the Faculty of Economics and Business Administration, Babes-Bolyai University, Cluj-Napoca, Romania (e-mail: ovidiu.veres@econ.ubbcluj.ro).

In a recent study [6] results show that job search assistance programs are an effective means to reduce unemployment and thus a valuable asset to the labor and job market, as well as the fact that are well correlated with counseling and courses for job seekers. Another study [7] on more than 2000 MBA graduates shows that candidates tend to apply for positions for which they qualify better and in which they are better prepared. From this, we could deduce the fact that a student or a job seeker is more prone to applying to jobs in the fields he previously studied or worked in. Graduates start searching for job before graduation but with a low intensity [8], usually with the fine intention of labor market prospecting. As they are about to graduate the intensity is higher and they are more pragmatic, have more realistic wage expectations. Our proposed model helps them be slowly acquainted with the targeted domain for job search, as our model learns which ad might be of interest to them and delivers only the jobs ads that meet the established score level. A high score level means the ad fits the applicant choices, whereas a small one means the job is of lesser interest to him. The above-mentioned study [8] states that the employment rate at graduation can be increased from 40 to 65 % if the student would start looking for jobs 6 months prior to graduation. The study was conducted in the Netherlands so the figures do not apply exactly to the Romanian market but the raising tendency can be observed, making our system appropriate in this context.

Regarding the methods used for the ad classification, we could choose from several classifiers usually employed in data processing. In [9] the authors ran a comparison test between three classification methods: Naïve Bayes, Decision Trees and Neuronal Networks in order to classify "training" web pages. The conclusion was that the best result on their data set was achieved with the help of the Naïve Bayes classifier. In our research, we have done a similar test for our data, using both a Naïve Bayes Classifier and a J48 Decision Tree, but the result of the comparison indicated that our training set offered more accurate results for the J48 classifier.

The information extraction methodology proposed for the web data retrieval is similar to the one in [9] because it employs both a crawler and a function that eliminates the unrelated URLs, but the models are different regarding the data extraction. We parse the web page in order to extract the general data from an employment ad corpus (like date, type of job, department and so on), while they wish to extract the title, URL and Meta-description from every page, thus the types of data used for analysis and classification are different. This kind of information from web pages, like meta-data (description, keywords) and titles are used in information retrieval for specific search, ranking and indexing tasks and are at the basis of modern commercial search engines. In our research we focus in extracting more accurate information from a page and focus on its actual content, rather than just identifying what that certain page is generally about [17].

Their classification separated web pages into relevant and non-relevant, whereas ours wishes to offer a score on a scale from 1 to 5 for a certain job ad, according to its accuracy from the subject's point of view.

Relevant methods for retrieving data from the web we found also in [10]. The solution found for retrieving data records from a webpage was creating a HTML tag tree, whose leaves are represented usually by a data record. The data records can be identified because they generally have the same structure and size. This approach is proven efficient in the case of extracting non-contiguous data records from web pages, compared to other methods, but for our research in cannot be applied because each ad is fount on a separate web page. General IR methods have a low efficiency rate and for this reason, we have opted for a more customized approach, but as a general setting for this kind of study we have tried to customize the consecrated models like for data retrieval, meaning the use of a transversal crawler which creates a list of candidate links from a general web graph structure but which in our case is restricted to the pages within the web site domain, and a parser to extract the information [17].

III. Methodology

The first step in our research was to study the content of several employment offer websites for the Romanian labor market in order to observe the structure and formats of employment ads, to determine the similarities and differences between them and to locate the problems that might arise during the data extraction process.

A similarity in the structure of most of the websites analyzed is the weak structuring of the data. The content of the ads is not fully standardized into an easy to extract format and because of this, we had to build a few exceptions and special cases. Most of the websites have a similar structure and many employment offers can be found on several of them. The first criteria for choosing the website was the traffic ranking, then the availability of an ad archive from where to collect the test data. From several website options, we decided on one of the most visited website on the Romanian market [15]. According to SATI [11] the website has over 1.300.000 viewed pages, 120.000 visits, from which more than 100.000 unique visitors per day. We considered it relevant for our study due to its quality indicators.

One of the biggest challenges when extracting data from the web is the heterogeneity of data and the unstructured manner it is presented. For our research, we have succeeded to a big extent to indentify the data we want to extract and present it in a more structured manner.

The methodology adopted has several steps and includes: the data extraction from the website, the preprocessing of the data, the evaluation and filtering of the attributes considered, the training of the algorithm and the tests conducted for the validation of the model. The final step is the results dissemination.

Fig. 1 describes the above steps in a schematic manner, while a more comprehensive description will follow.

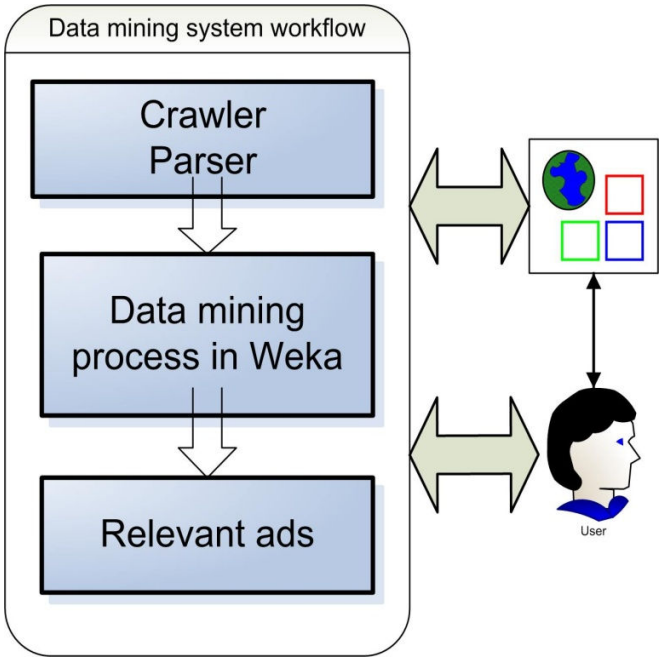

Figure 1. System functionality

A. Data extraction

In order to extract data from the website we employed a couple of techniques we have considered to be both cost and time effective, meaning a crawler and a parser for data extraction.

Information retrieval is the generic term for searching documents, information within documents, data from relational databases or the World Wide Web [17]. Concepts like data, document, information and text retrieval may overlap, but in our context we can delimit our space of interest to mining semi-structured data records, with the help of observed data patterns. While there are several IE methods [12], for our experiment we used a custom parser that could efficiently extract the data in the proper format that we were interested in. After extracting the data, we inserted it into a Weka database, a step suggested in [13].

With the help of the crawler, we were able to visit the links of the website that were of interest to us and store them in a list. A general crawler, as the one used in our case, has the purpose of crawling all the pages of a website. The principle behind the crawler is as follows: starting from an URL, the contents of the page are analyzed and the hyperlink parser detects the URL within the page, adds them to a queue with pages to be visited, which are later accessed by the transversal crawler. With the use of the crawler we also ensured that fact that the URLs are unique and do not repeat themselves, thus the risk of having duplicated data that could interfere with the results was reduced. Since not all the pages are of interest, in this case we had to adapt to this situation by crawling mostly ads from the archive. We have positioned ourselves to the beginning of the archive for 1 January 2010 and decided that crawling to a depth of 2 into the main URL tree would be enough to cover the archive for January, meaning around 4000 URLs containing employment ads. For a depth of 3 for the crawler the number of URL retrieved would be of more than 100.000 URLs, from which less than half containing actual employment ads.

As expected, the results after the crawl were unrefined and the retrieved URLs were not all of interest to our study, they included pages that were not withholding ads but intermediary pages and general pages of the website, like company pages or archive pages close to the root.

After a brief observation we could easily notice the fact that the employment ads had a similar structure and path (e.g. http://www.bestjobs.ro/locuri-de-munca-{job_title}). This helped us eliminate some of the URLs, which did not contain a job advertisement. After this processing step, the URL data was clean and for the month of January, we retrieved 3670 general ads.

For extracting data from HTML pages we did not employ a wrapper technology [14], a programmed designed to extract structured data from the web [18], instead we used a parser that was able to select the valuable information. The parser is based on a string matching algorithm [10], meaning we look for a certain string and retrieve the corresponding value for the data fields we are interested in, while eliminating the rest of the HTML code containing irrelevant information.

The steps followed by our algorithms were:

- Retrieve an URL
- Open and search the relevant content as following:
  - Indentifying the repeating element that helps us separate the fields within the so called data record, meaning the entire content of an employment advertisement
  - Identifying the specific lines containing the useful information
  - Retrieving the values for the selected attributes considering the repeating element field
- Write the output content in the specific format file

The parser extracted the text variables corresponding to the fields identified on the page and put them in an *.arff* file, the specific file format for Weka.

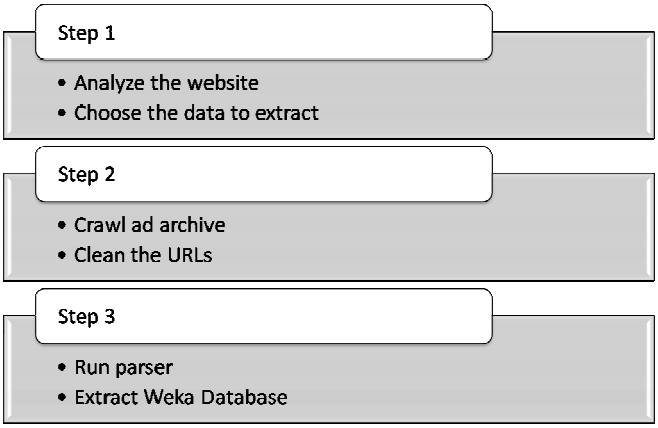

Figure 2. Retrieving data from the web

B. Data preprocessing

The preprocessing step in our research is a component of the data mining process and it consists in the preparations made before testing the classification algorithms.

After the data was collected from the website, we made a pre-analysis of each record, rating it according to the user preference.

The typical user for our research was considered a university graduating student, specialized in Finance, which will soon need to be employed.

For the preferences to be considered more relevant, we used the data from a survey conducted with a representative sample of 540 students, expressing their expectations, difficulties and experiences in searching for a job.

Following their preferences, we built an expert function that rates every record. The rating was introduced as the final attribute of each record. One of the limitations found in the ads rating process is the fact that one of the most important factor when applying for a job, the salary level is missing from the records, the employers not making public their precise offer.

In order to begin the data processing there was one more step to be covered: the evaluation of the attributes. Weka has several tools for attribute evaluation, from which we chose Info Gain Attribute Evaluator, Chi-Squared Attribute Evaluator.

The Info Gain Attribute Evaluator discretizes the numerical attributes then evaluates the attributes by measuring their information gain, considering the class. The Chi-Squared Attribute Evaluator calculates the intensity of the relation between attributes, using the Chi-Square test.

C. The classification algorithms

In the data mining literature there are various algorithms we can use in order to obtain knowledge and we must select the most appropriate ones for our case. Multiple classifiers are run in order to derive better classification results [19]. This was also our intention when deciding to run different algorithms in order to study the results and find an accurate method for our dataset.

The J48 algorithm represents the implementation of the C4.5 algorithm, created by J. Ross Quinlan, using an inductive top-down method for decision trees. They are built by testing every node of the tree starting with the first node, for every record. Each node represents the name of an attribute. The algorithm tries to introduce the record tested into an existing class, considering the similar aspects, evaluating the attribute that corresponds to the current node. Depending on its value, the instance will follow a branch of the tree. If a class does not seem enough different from another due to similarities of their records, they will be united. The process has the name of *prunning*. Decision trees are a common method to visualize the results of a machine learning algorithm. The tuning of the algorithm becomes a very important task. The tendency to apply the pruning on the results, as a method to obtain fewer and easily interpreted data is not always appropriate, as significant information could be lost. The unpruned decision tree is very efficient on the training data, but it has a significant chance to perform poor on a test dataset. The pruning will reduce however the accuracy for the training set because of its aim to create more flexible rules in the decision tree. The J 48 algorithm has two methods for pruning. The first attempts to replace the nodes in a certain branch with a leaf (backwards, from the leaves to the root of the tree), reducing the rules on a certain path. The other option is to move a node and all its sub nodes and leaves upwards to the root. There is a need of balance when applying the pruning in order to obtain a flexible and robust system [22].

The Naïve Bayes classification can be a very accurate and simple classification method for data in general but also for web data [9]. It starts from Bayes' theorem (1) and the naïve assumption that the data is independent and that variables do not influence each other when making a classification. The Bayesian classifier assumes that the variables are independent and they are analyzed from a probabilistic point of view. Nevertheless, this kind of classification is efficient in the process of supervised learning.

$$p(C|F_1,\dots,F_n) = \frac{p(C)p(F_1,\dots,F_n|c)}{p(F_1,\dots,F_n)} \qquad (1)$$

The Naïve Bayes classifier is based on the Naïve Bayes probability model (1), but generally also implies a MAP decision rule, namely *maximum a posteriori.*

IV. Results

Following the above-described methodology, we built a database containing 3670 instances as a training set used for learning the algorithms and a 917 instances test set.

To validate the attributes we used the attribute evaluation functions available in Weka. The attribute evaluators tested the variables against the rating and all three found five attributes relevant (in order of their importance): the experience, the city, the domain, the offers and the job status. Info Gain Evaluator ranked the *experience* attribute as being the most important, followed by the *domain* and the *city*. Chi Squared Evaluator found the *city* as the most meaningful attribute, while the *domain* and the *experience* came in the next places. Appendix 1 shows in detail the results of the evaluations.

In the next step of the experiments, we applied the J48 and Naïve Bayes algorithms on the entire data set for the training. We extracted a different and smaller data set for testing. The data set consisted of 3670 instances that were classified by the algorithm with a success rate of 79.18%. The tests were conducted on a 917 instances data set, the results showing that the system learned very well on the training set, providing a rate of correctly classified instances of 77.42%. Appendix 2 presents the results of the tests.

Next, we tried to verify if the results could be improved by using another algorithm, and we chose J48 for testing. The training set provided was the same with the one used for the Naïve Bayes algorithm, but the results were considerably different. The J48 algorithm performances were much better,

obtaining an accuracy of 88.71%, which means 3256 correctly classified instances out of the total of 3670. We supplied after that the test set, in order to validate the learning. The system correctly classified 771 instances, representing 84.07% out of the total data set for test.

When analyzing the results we took into considerations the main indicators of performance, specific data mining evaluators, that allow us to draw more complex conclusions regarding our data: precision, recall, F-measure, mean average and others [17], all also available for the Weka implementation of the selected algorithms.

The improvement of 9% in accuracy leads to the conclusion that J48 fits much better the data (as reflected in Figure 3) and is more appropriate for our study. Appendix 3 shows the detailed results of the J48 classification.

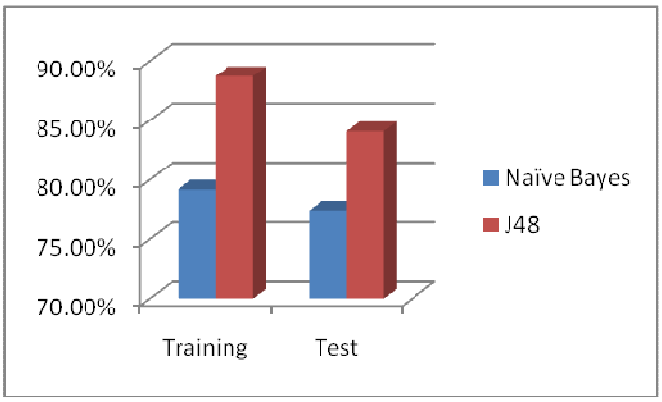

Figure 3. Classification accuracy

As anticipated, the decision tree is very large, because of the multitude of values in some variables, such as for *city* and *domain*. The system however identifies the jobs that score a high rating, and they can easily be provided to the end user.

An example of interpretation of the decision tree is the following: "If the required experience level is "no experience", the city selected is "Cluj-Napoca" and the domain is "Banks" the offer can be rated 5."

We can observe that out of the 917 instances tested only 20 obtained a rating of 5, being of real interest for the user.

V. Conclusion

In this paper, we applied the Naïve Bayes and J48 algorithms on a large data set of employment offers with the aim to extract relevant information according to the specific user needs.

Web content mining aims to extract useful information or knowledge from the content of web pages. They can be clustered and classified based on their topic [18]. With the help of a crawler we can collect information in order to be analyzed using the data mining methods for obtaining new and useful knowledge.

The tests conducted showed a better performance for the J48 algorithm, obtaining a performance of classification of 84%, compared to 77% in the case of Naïve Bayes. We concluded that J48 is more appropriate for our study and propose it as valid for future applications.

We identified that the specialized job search websites in Romania provide the offers semi-structured manner, part of the content being organized, while the rest not at all, and needing a consistent work to obtain structured information in order to analyze it. Another problem discovered is that the employers do not offer the level of the salary for the offers placed, the workers having to choose between other characteristics in order to apply for an interview.

From an economical view, the applications of the proposed work can be highly efficient for young graduates looking for employment. Its purpose can be extended not only to graduates but also other categories, and not only to a certain field of interest. The model can be extended according to a user profile and preference to any subscriber of the website that fill in her CV with experience and education, making the model even more accurate and personalized.

The applications of the model can be in a different search function than the simple filtering that is available on the site. The implementation of such a model can lead to a personalization of the results and of course a better match between the ad and the applicant's preferences.

The other application is creating personalized RSS feeds for subscribers. This model does not imply for the user to search for job. The entire content is delivered to him as a new job opening and offers in his field of expertise arises are added to the website, this application being extremely useful for those who prospect the market, future graduates or already employed job seekers.

For further improvements in our results, a method like the one proposed by Tsatsoulis [21] could be used, where signals from different classifiers are gathered and weighted in order to obtain increased performances.

Our goal was to bring the employment advertisements a step closer to the user by using a more personalized approach. Personalized information offers various advantages: it saves the user's time by eliminating repetitive or time consuming tasks, money, offers better information and tends to actual needs and seizes the best opportunities [20]. Our proposed system is useful due to better content management for users, offering a personalized and prioritized insight over a large amount of heterogeneous data.

Appendix

APPENDIX 1

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 6 rating):
Information Gain Ranking Filter

Ranked attributes:
0.30151 2 experience
0.24563 3 city
0.21441 4 domain
0.02539 1 job_status
0.0067 5 offers

Selected attributes: 2,3,4,1,5 : 5

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 6 rating):
Chi-squared Ranking Filter

Ranked attributes:
1791.892 3 city
1619.129 4 domain
1404.708 2 experience
362.762 1 job_status
36.796 5 offers

Selected attributes: 3,4,2,1,5 : 5

APPENDIX 2

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes
Relation: rating_jobs
Instances: 3670
Attributes: 6
job_status
experience
city
domain
offers
rating
Test mode: evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

Time taken to build model: 0.03 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	2906	79.1826 %
Incorrectly Classified Instances	764	20.8174 %
Kappa statistic	0.6409	
Mean absolute error	0.1405	
Root mean squared error	0.2495	
Relative absolute error	57.9434 %	
Root relative squared error	71.6598 %	
Total Number of Instances	3670	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.2	0	0.8	0.2	0.32	0.997	1
0.263	0.017	0.584	0.263	0.363	0.964	2
0.821	0.151	0.827	0.821	0.824	0.9	3
0.938	0.178	0.785	0.938	0.855	0.95	4
0.047	0.007	0.194	0.047	0.075	0.969	5
Weighted Avg.	0.792	0.145	0.767	0.792	0.769	0.929

=== Confusion Matrix ===

a	b	c	d	e		
4	1	15	0	0	|	a = 1
0	80	213	11	0	|	b = 2
1	48	1409	253	6	|	c = 3
0	8	66	1407	19	|	d = 4
0	0	1	122	6	|	e = 5

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	710	77.4264 %
Incorrectly Classified Instances	207	22.5736 %
Kappa statistic	0.612	
Mean absolute error	0.1494	
Root mean squared error	0.2621	
Relative absolute error	61.4917 %	
Root relative squared error	75.1028 %	
Total Number of Instances	917	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.333	0	1	0.333	0.5	0.999	1
0.198	0.016	0.552	0.198	0.291	0.952	2
0.805	0.169	0.797	0.805	0.801	0.868	3
0.925	0.177	0.792	0.925	0.853	0.942	4
0.032	0.017	0.063	0.032	0.043	0.917	5
Weighted Avg.	0.774	0.153	0.749	0.774	0.751	0.909

=== Confusion Matrix ===

a	b	c	d	e		
1	0	2	0	0	|	a = 1
0	16	62	3	0	|	b = 2
0	12	334	62	7	|	c = 3
0	1	20	358	8	|	d = 4
0	0	1	29	1	|	e = 5

APPENDIX 3

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: rating_jobs
Instances: 3670
Attributes: 6
job_status
experience
city
domain
offers
rating
Test mode: evaluate on training data

=== Classifier model (full training set) ===

Time taken to build model: 0.2 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	3256	88.7193 %
Incorrectly Classified Instances	414	11.2807 %
Kappa statistic	0.8109	
Mean absolute error	0.0645	
Root mean squared error	0.1729	
Relative absolute error	26.5817 %	
Root relative squared error	49.6701 %	
Total Number of Instances	3670	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.25	0.001	0.714	0.25	0.37	0.99	1
0.75	0.01	0.877	0.75	0.809	0.992	2
0.904	0.105	0.883	0.904	0.894	0.976	3
0.917	0.08	0.888	0.917	0.902	0.98	4
0.736	0.001	0.979	0.736	0.841	0.99	5
Weighted Avg.						
0.887	0.083	0.887	0.887	0.885	0.98	

=== Confusion Matrix ===

a	b	c	d	e		<-- classified as
5	0	15	0	0	|	a = 1
2	228	67	7	0	|	b = 2
0	32	1553	132	0	|	c = 3
0	0	123	1375	2	|	d = 4
0	0	0	34	95	|	e = 5

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	771	84.0785 %
Incorrectly Classified Instances	146	15.9215 %
Kappa statistic	0.7325	
Mean absolute error	0.0837	
Root mean squared error	0.2067	
Relative absolute error	34.4531 %	
Root relative squared error	59.2215 %	
Total Number of Instances	917	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.333	0	1	0.333	0.5	0.998	1
0.667	0.006	0.915	0.667	0.771	0.952	2
0.824	0.106	0.866	0.824	0.844	0.948	3
0.915	0.166	0.801	0.915	0.854	0.956	4
0.645	0	1	0.645	0.784	0.977	5
Weighted Avg.	0.841	0.118	0.848	0.841	0.839	0.953

=== Confusion Matrix ===

a	b	c	d	e		<-- classified as
1	0	2	0	0	|	a = 1
0	54	18	9	0	|	b = 2
0	5	342	68	0	|	c = 3
0	0	33	354	0	|	d = 4
0	0	0	11	20	|	e = 5

REFERENCES

[1] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, 2005, pp 198.

[2] D.Van Rooy, A. Alonso, and Z. Fairchild, "In With the New, Out With the Old: Has the Technological Revolution Eliminated the Traditional Job Search Process?" in *International Journal of Selection and Assessment*, Vol. 11, pp. 170-174, June 2003.

[3] C. Neagu, *Situaţia statistică operativă a şomajului înregistrat la 30 septembrie 2010*. Retrieved from National Agency for Workforce Employment. Available: http://www.anofm.ro, 2010, October 22.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, "The WEKA Data Mining Software: An Update" in *SIGKDD Explorations*, Volume 11, Issue 1, 2009, pp. 10-18.

[5] Denis Fougère & Jacqueline Pradel & Muriel Roger, "Does Job-Search Assistance Affect Search Effort and Outcomes? A Microeconometric Analysis of Public versus Private Search Methods," *IZA Discussion Papers*, 1825, Institute for the Study of Labor (IZA). Available: http://ideas.repec.org/p/iza/izadps/dp1825.html, Oct 2005.

[6] Stephan Thomsen, "Job Search Assistance Programs in Europe: Evaluation Methods and Recent Empirical Findings," *FEMM Working Papers* 09018, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management. Available: http://ideas.repec.org/p/mag/wpaper/09018.html, May 2009.

[7] Kuhnen, Camelia M., "Searching for Jobs: Evidence from MBA Graduates," *MPRA Paper* 21975, University Library of Munich, Germany. Available: http://ideas.repec.org/p/pra/mprapa/21975.html, Jan 2010.

[8] van der Klaauw, Bas & van Vuuren, Aico & Berkhout, Peter, "Labor market prospects search intensity and the transition from college to work," *Working Paper Series* 2005:9, IFAU - Institute for Labour Market Policy Evaluation. Available: http://ideas.repec.org/p/hhs/ifauwp/2005_009.html, Apr 2005.

[9] D. Xhemali, C. J. Hinde and Roger G. Stone, "Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages", *International Journal of Computer Science Issues*, IJCSI, Volume 4, Issue 1, pp16-23, September 2009.

[10] Liu, B., Grossman, R., and Zhai, Y., "Mining data records in Web pages" in *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining* (Washington, D.C., August 24 - 27, 2003). KDD '03. ACM, New York, NY, pp. 601-606.

[11] The Romanian Internet Traffic and Audience Studio. Available: http://www.sati.ro/

[12] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, Khaled F. Shaalan, "A Survey of Web Information Extraction Systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1411-1428, October 2006.

[13] McCallum, A., "Information Extraction: Distilling Structured Data from Unstructured Text", in *Queue* 3, 9 (Nov. 2005), pp. 48-57.

[14] Chun-Nan Hsu, Ming-Tzung Dung, "Generating finite-state transducers for semi-structured data extraction from the Web", *Information Systems*, Volume 23, Issue 8, Semistructured Data, pp. 521-538, December 1998.

[15] Job search website. Available: www.bestjobs.ro.

[16] M. E. Andreica, M. D. Antonie, A. Cristescu, and N. Cătăniciu, „A Panel Data Analysis of the Romanian Labour Market", in *Proc of the 5th WSEAS International Conference on Economy and Management Transformation*, Timisoara, Romania, 2010, pp. 406-411

[17] Hazem M. El-Bakry, Alaa M. Riad, Ahmed Atwan, Sameh Abd El-Ghany, and Nikos Mastorakis. 2010. A new automated information retrieval system by using intelligent mobile agent. In *Proceedings of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases* (AIKED'10), Lotfi A. Zadeh, Janusz Kacprzyk, Nikos Mastorakis, Angel Kuri-Morales, Pierre Borne, and Leonid Kazovsky (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 339-351.

[18] I. Dzi¸tac, I. Moisil, Advanced AI Techniques for Web Mining, *Proc. of 10th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems* (MAMECTIS '08, Corfu, Greece, 343-346, 2008.

[19] T. Lahiri, S. Samal, A novel technique for making multiple classifier based decision, Proc. WSEAS International Conference on Mathematical Biology and Ecology, Corfu, Greece, 2004, 488-341.

[20] Nicoleta David, Claudia-Georgeta Carstea, Lucian Patrascu, Ioan-Gheorghe Ratiu, and Lidia Mandru. 2010. Building solutions for web personalization. In *Proceedings of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases* (AIKED'10), Lotfi A. Zadeh, Janusz Kacprzyk, Nikos Mastorakis, Angel Kuri-Morales, Pierre Borne, and Leonid Kazovsky (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 98-101.

[21] Tsatsoulis, C., Lee, D., Using an ensemble classifier for machine learning applications, in *Proceedings of the 9th WSEAS International Conference on Computers* (ICCOMP'05), Stevens Point, Wisconsin, USA, Article 58, 2005.

[22] Han J, Kamber M., Data Mining: Concepts and Techniques. Morgan Kaufmann. Second edition, 2006, pp.304-306.