Technical Sciences, 2023, 26, 125–140

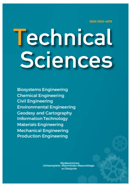
UNIWERSYTET
WARMIŃSKO-MAZURSKI
W OLSZTYNIE

Yearbook peer-reviewed scientific journal

ISSN 1505-4675
e-ISSN 2083-4527

TECHNICAL SCIENCES

Homepage: https://czasopisma.uwm.edu.pl/index.php/ts/



ISSN 1505-4675

Technical Sciences

Biosystems Engineering
Chemical Engineering
Civil Engineering
Environmental Engineering
Geodesy and Cartography
Information Technology
Materials Engineering
Mechanical Engineering
Production Engineering

Wydawnictwo
Uniwersytetu Warmińsko-Mazurskiego
w Olsztynie

DOI: https://doi.org/10.31648/ts.9319

IMPROVING THE CREDIBILITY OF THE EXTRACTED POSITION FROM A VAST COLLECTION OF JOB OFFERS WITH MACHINE LEARNING ENSEMBLE METHODS

***Paweł Drozda*[1], *Krzysztof Ropiak*[2], *Bartosz A. Nowak*[3], *Arkadiusz Talun*[4], *Maciej Osowski*[5]**

[1]ORCID: 0000-0003-3163-9408
Faculty of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn
Emplocity SA, Warszawa

[2]ORCID: 0000-0001-8314-0276
Faculty of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn

[3]ORCID: 0000-0002-2577-4470
Faculty of Mathematics and Computer Science
University of Warmia and Mazury in Olsztyn
Emplocity SA, Warszawa

[4]Emplocity SA, Warszawa

[5]ORCID: 0000-0003-0277-3798
Emplocity SA, Warszawa

Received 27 August 2023, accepted 14 September 2023, available online 19 September 2023.

Keywords: machine learning, web scraping, granularity methods, classification.

Abstract

The main aim of this paper is to evaluate crawlers collecting the job offers from websites. In particular the research is focused on checking the effectiveness of ensemble machine learning methods for the validity of extracted position from the job ads. Moreover, in order to significantly reduce the training time of the algorithms (Random Forests and XGBoost), granularity methods were also tested to significantly reduce the input training dataset. Both methods achieved satisfactory

Correspondence: Paweł Drozda, Katedra Metod Matematycznych Informatyki, Wydział Matematyki i Informatyki, ul. Słoneczna 54, 10-710 Olsztyn, e-mail: pdrozda@matman.uwm.edu.pl.

results in accuracy and F1 measures, which exceeded 96%. In addition, granulation reduced the input dataset by more than 99%, and the results obtained were only slightly worse (accuracy between 1% and 5%, F1 between 3% and 8%). Thus, it can be concluded that the considered methods can be used in the evaluation of job web crawlers.

Introduction

The process of finding suitable candidates for jobs is an increasing problem in developed countries and requires a lot of resources for its successful completion. Large companies often employ their own staff in HR departments to conduct employee recruitment processes. The entire process is divided into the phase of filtering candidates in terms of their competencies, and then inviting them to interviews and presenting the job offer. A significant part of these activities is still a manual process, which increases the overall cost of the recruitment process and extends its time.

It is common for the same job offer to be placed in many advertisement systems, which makes it difficult to both manage them and filter offers for potential candidates to apply for. The project called Emplobot is a solution that automates the recruitment process, which minimizes the amount of work needed to match the right candidates with the available job offers. The development of NLP gave the opportunity to use ML techniques and conduct research on its usefulness in many areas, such as the job market.

Emplobot is a solution that conducts a conversation with the candidate and prepares a virtual CV for him, which is then matched to the database of job offers collected by the crawlers. The data that is collected is sometimes redundant, i.e. it contains the same requirements for candidates as other offers, so processing it in its entirety is not always optimal. In such situations, knowledge granulation methods can be useful when we want to reduce the number of observations when preparing data that will feed into ML algorithms in order to train a classification model.

The main purpose of the paper is to present mechanisms that automate certain stages of the recruitment process using computer techniques, from web scraping through knowledge granulation and the use of machine learning methods. The novelty of the techniques described is both the area of the labor market and the way of combining many techniques (scraping, granulation, classification, NLP bot) in order to support the work of people in recruitment departments, but also employees looking for job offers that are best suited to them. The experimental part focuses on the use of granulation of knowledge based on the theory of rough sets as a data preprocessing phase before training classification models, which is to reduce the volume of data entering the models during the training process. The good results achieved in the experiments provide a basis for further testing and development of proposed solution.

This paper has the following structure. In the next section related works have been indicated. The subsequent section presents the methodology of realized experiments. There can be found information about data collected by Emplobot crawler and logic behind data granulation used as well. Experimental results are described in section 4 along with conclusions drawn after they have been carried out in the last section.

Related work

With the development of the Web, more and more information is posted there. The pages containing the content are designed to be conveniently read by humans, and the way data is presented is not so heavily standardized that the content can be easily processed by computers. Sometimes page authors even want the information on their pages to be hard for machines to process.

This paper addresses the topic of extracting information about a proposed position in a job advertisement from a web page. Thus, the process addresses many subjects, from collecting useful text from the page to verifying that the predicted position is accurate.

The subject of extracting data from web pages is often called web-scraping (PARVEZ et al. 2018, LOTFI 2001). This process involves extracting valuable text from a web page and managing mass processing of multiple pages. During this task, it is necessary to remove unnecessary things like tags, scripts, styles and other content concerning the appearance of the page. The solution presented here, however, did not directly process the site only in this way. As a result, it was possible to preserve and use additional information about the structure of the webpage. There are similar solutions known and appreciated (FINN et al. 2001).

Emplobot uses general information about the site at the beginning of the information gathering process. This time it is used to investigate whether the site contains a job ad. First, information is collected in a heavily heuristic manner, then the computer system uses them by an appreciated XGBoost (CHEN, GUESTRIN 2016) classifier (NOWICKI, STARCZEWKI 2017).

Similar solutions (HASHEMI 2020, SHETE et al. 2021) are also used for a wide variety of applications. A very common use of similar techniques, for example, is to check whether a site is malicious and should be either censored or provided with a warning (ZOU et al. 2019, CHANG et al. 2023).

The next thing, after determining whether the site contains a job advertisement, is the whole process involved in extracting information about the proposed position. Unfortunately, ads may contain this information in various places, or it may be embedded inside the text describing the entire ad. Techniques known from Natural Language Processing (NLP) can be used for this purpose. This field, in a nutshell, mainly deals with the extraction of information by a machine from text written in “human” language (KAO, POTEET 2006, TREVISO et al. 2023).

There are very few documents dedicated to the classification of job advertisements (KIM, LEE 2016). This document describes methods related to the further development of the Emplobot information system. This time it is to verify that the prediction of a position is reliable.

Granular computing is a fairly common data processing technique, and the homogeneous granulation that was presented in (ROPIAK, ARTIEMJEW 2018) has been applied to a much larger data set. The effectiveness of data granulation in the process of reducing the number of observations and maintaining the effectiveness of classification in ensemble models has been confirmed, among others, in ARTIEMJEW and ROPIAK (2021).

Used methods

According to data obtained from the LinkedIn website (RABBI 2021), it takes an average of 6-7 months to find a new job. This time, of course, depends strongly on the type of job sought, the location, qualifications and commitment to the candidate's search. It turns out that the automated system for finding work has great applications.

The task of automatically finding a suitable job offer for a person can be divided into parts:

– Gathering data on existing job offers;

– Acquiring information from a real person regarding the preferred job's characteristics and that person's existing skills and even characteristics;

– Matching the most suitable offers to the person.

This article focuses on a part of the work required by the first phase. Below will be the process starting from collecting the information from the web page to getting information about what position is proposed on this ad and whether the system is confident of this answer. This entire process can be depicted in Figure 3.

In previous papers (DROZDA et al. 2019, TALUN et al. 2020), the authors focused on the initial elements of this task, while in this paper the final elements are mainly discussed.

The whole process starts with gathering information about the web page being processed. The main purpose of this is to check whether the specified site is a job advertisement.

The authors used a tool widely called Web Crawler to acquire the following information about a single page:

– Quantities of HTML elements of type A, BUTTON, DIV, FORM, IMG, INPUT, LI, counted separately;

– Text length counted in various ways: on the whole page; on the whole page after removing unnecessary things like scripts, styles, etc.; in the header only.

In addition, simple calculations have been added to these data to help the classifier:

– the ratio of the length of the text after removing unnecessary elements to the length of the text on the whole page;

– the ratio of the number of DIV elements to the length of the text after removing unneeded webpage elements.

Based on presented data, the XGBoost classifier (CHEN, GUESTRIN 2016) was used to predict whether the site was a job advertisement. The details of this process were explained in a previous paper (TALUN et al. 2020).

The next part of the system is responsible for gathering data about all job positions mentioned in the web page. This part can be described by the following process:

1. Convert all words into their basic forms. This process is particularly useful for processing text data in Polish, but should also be used for other languages. For example, in Polish, words: "kierownik", "kierownika", "kierownikiem", "kierowniku" can be translated into English words: "manager" or "manager's".
2. Remove irrelevant text content such as dots etc.
3. Find all words and groups of words that are directly associated with a job position. For this step the system uses a manually prepared list of words related to job positions, such as: "manager of production", "IT technician" etc.
4. For every job position found in the previous step, count the number of its occurrence.
5. Extract additional information about every job position (details are explained in the next paragraph).

After successful extraction the table with candidates of the main job position is created. The table consists of rows and columns, the row describes a single job's position. The tables has following columns:

– count – number of occurences of this word / group of words on the website;

– tag – name of HTML tag of the element with the word;

– child – name of HTML tag of the first child's tab (if exists), or "x";

– parent – name of HTML tag of element's parent (if exists), or "x";

– xpath – length of full path to the current HTML element (XPath);

– raw_text – length of raw text in current HTML element;

– elements – number of child HTML elements for current element;

– position_name – current word / group of words;

– link_pos – checks if the job position also exists inside the URL of the website.

It is important to note that column "link_pos" is a surprisingly important feature.

Previously created table describes all potential job's positions mentioned in the job offer. Inside a single job offer there could be a few job offers, also websites often contain words that misleads in the process of recognition of a true job's position. For example some job offer contains following sentence “Our company is world-famous for production of ..., we offer a job as a CAD-designer”, in that example use of the word “production” may mislead the system that the job offer is about job position ”production” instead of “CAD-designer”.

The system uses Random Forest (Ho 1995, Qi 2012) in order to determine the main job position mentioned in the job’s offer. For every row, the random forest after training returns the probability that the appropriate “position_name” is the main job position of the page. Random forest was created using the following parameters: number of trees equal 100, entropy as function to measure the quality of a split in branching.

The last part of the system deals with resolving whether the predictions about the main position are certain. For this purpose, the position_name column is removed from the previously prepared table. For the preparation of the learning dataset, several thousand samples were almost manually prepared, each containing the content of the page and information regarding the main position in the job posting.

Granulation method

The granulation used in the experimental part is a knowledge granulation method based on the logic defined in the rough set theory (Pawlak 1982), which in turn derives from the theory of mereology described by Leśniewski (Leśniewski 1916).

The granulation method that has been used in this paper is homogeneous granulation, which is a variant of the concept dependent granulation described in more detail in (Polkowski 2007). The difference is that the granulation process is not performed for every possible granulation radius (the number of which is equal to $|A|$ + 1), but it is selected automatically for each central object until the granule (i.e. similar objects) they only hit an object of the same decision class.

The granulation process is several-stage and consists of successive steps.

1. Loading the original decision system (**U** – universe of objects, **A** – non decision attributes, **d** – decision attribute).

2. Eliminating the problem of object ambiguity within different decision classes.

3. The granules are formed as follows:

$$g_{r_u}^{\text{homogeneus}} = \{v \in U\colon \left|g_{r_u}^{cd}\right| - \left|g_{r_u}\right| = 0, \text{for minimal } r_u \text{ fulfills the equation}\},$$

where

$$g_{r_u}^{cd} = \left\{v \in U\colon \frac{|\mathrm{IND}(u,v)|}{|A|} \le r_u \text{ AND } d(u) = d(v)\right\}$$

and

$$g_{r_u} = \left\{ v \in U : \frac{|\mathrm{IND}(u, v)|}{|A|} \leq r_u \right\},$$

$$r_u \in \left\{ \frac{0}{|A|}, \frac{1}{|A|}, \ldots, 1 \right\}$$

μ is an approximate inclusion, formally derived from Lukasiewicz's *t*-norm.

4. Granular coverage is created based on chosen strategy:
   – hierarchical coverage (granules are selected by sequence),
   – random selection of granules,
   – selecting granules with minimal, mean or maximal length,
   – selecting granules that convey the least, most or average number of new objects, respectively,
   – random selection of granules depending on concept size.

When given granule passes, at least one new object into the coverage is selected.

The original decision system is considered covered when the unique set of objects derived from the coverage granules overlaps with the entire original set of objects. It means that granules with center of **u** meet the condition:

$$\cup \left\{ g^{cd}_{r_{\mathrm{gran}}}(u) : g^{cd}_{r_{\mathrm{gran}}}(u) \in U_{\mathrm{cover}} \right\} = U,$$

All objects in each granule are voting through a **majority voting** function which is used to select a representative new object. All ties are resolved by random choice. After all granules are processed, a new granular decision system is formed.

The granular reflection of the original decision system $D = (U, A, d)$ is the new decision system (COV(U, μ, r), the set of objects formed from granules.

$$v \in g^{cd}_{r}(u) \text{ if and only if } \mu(v, u, e)\mu \wedge (d(u) = d(v))$$

for a given rough (weak) inclusion **μ**.

Majority voting is represented by the following notation:

$$\{MV(\{a(v) : u \in g\}) : a \in A \cup \{d\}\}.$$

This process allows a high degree of preserving the internal knowledge, representation, of the original decision system while reducing the number of observations that remain in the new reflective dataset. This amount depends on the degree of diversity of such a collection. Experiments conducted on the data described in this paper allowed to reduce the number of observations by over 99%, which proves the low diversity of observations and is an excellent example of the possible use of data granulation techniques.

Dataset description

Table 1 describes numbers for used datasets. Figures 1 and 2 show class balance in the original and granulated dataset.

Table 1

Datasets description

Dataset	Number of observations	Number of features
Original	197,846	61
Granulated	485	61

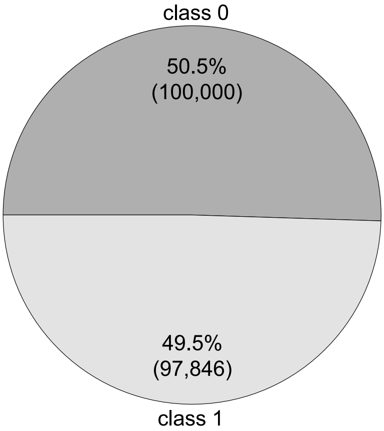


Fig. 1. Original dataset class balance structure

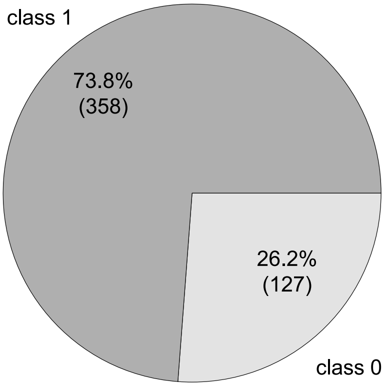


Fig. 2. Granulated dataset class balance structure

Experiments

The first objective of the experimental session was to evaluate the performance of two machine learning algorithms based on ensemble methods in determining the correctness of identifying the name of an occupation in job postings aggregated by the company's proprietary job parsers. In addition to determining effectiveness with the most commonly used measures: F1 and accuracy, the most relevant features considered in the study with the feature importance measurement were also determined.

The second goal of the experimental session was to measure the impact of significant data reduction through the granulation process, described in detail in the previous section. In particular, the research consisted of determining the decrease in effectiveness in F1 and accuracy measures after a very large reduction in the number of cases in the learning dataset. More specifically, the number of cases in the dataset under consideration was reduced from 197,846 to just 485 using granular algorithms while preserving the characteristics of the data provided. This means that in the reduced dataset, only 0.2% of the data was taken into account in the process of training the algorithms. This allowed for a significant reduction in training time, as is thoroughly indicated in the following subsections.

XGBoost and Random Forests Evaluation

The first part of the research conducted for this paper was to determine the effectiveness of the XGBoost and Random Forests algorithms in correctness of job name identification on job ads collected by Emplocity crawlers. The choice of the mentioned methods was motivated by the fact that they are very fast and relevant for many classification tasks. Specifically, models for various parameters were tr ained and evaluated through the most commonly used measures in evaluating model performance: precision, recall, accuracy and F1. The input dataset used for the study consisted of vectors described by 61 features, some of which were created by mapping categorical features using one hot encoding to binary vectors.

In addition to the main research objective of this part, the relevance of the individual attributes included in the input vector was also examined.

First, the Random Forests algorithm was taken into account. At the beginning, it should be mentioned that each evaluation of training models was carried out using a 5-fold cross-validation, which greatly reduced the possibility of badly dividing the input dataset into training and testing, and thus obtaining unreliable results. For all parameters, which determine the effectiveness of models, averaged measures were provided as final values. The evaluation consisted in tuning hyperparameters, where different values for n_estimators, max_depth and criterion were provided. In particular, *n*_estimators took the values of: 50, 100, 150, 200, max_depth: 4, 8, 12, 16, 20 and criterion: gini, log_loss, entropy. During the study, it turned out that the parameters *n*_estimators and criterion had a very negligible effect on the effectiveness of the Random Forests model in the training process, so in Table 2, the magnitudes of the measures were provided only for different max_depth. 200 and gini were chosen as default values for *n*_estimators and criterion, however it is worth mentioning that they had no effect on the final results.

Considering the results shown in the Table 2, it can be deduced that as the max_depth of random trees increases, the effectiveness of the model also increases. In the case of the accuracy measure, it reaches values from 91.52% to 96.02%, where the lowest value was achieved for max_depth = 4 and the highest for max_depth = 20. A similar trend can be observed for the F1 measure which also increases as max_depth increases from 91.52% to 96.02%.

Table 2

Results for Random Forests

Depth	Precision [%]	Recall [%]	F1 [%]	Accuracy [%]	Fit_time [s]
4	91.55	91.53	91.52	91.52	12.34
8	93.73	93.73	93.73	93.73	10.96
12	94.79	94.80	94.79	94.79	17.39
16	95.56	95.56	95.56	95.56	14.13
20	96.03	96.03	96.02	96.02	17.28

In addition to the results on the model's effectiveness, the study aimed to identify the most relevant features in terms of their impact on the algorithm's training. Figure 3 shows the distribution of the main features along with their importance for the best obtained results (max_depth = 20, criterion = gini, *n*_estimators = 200).

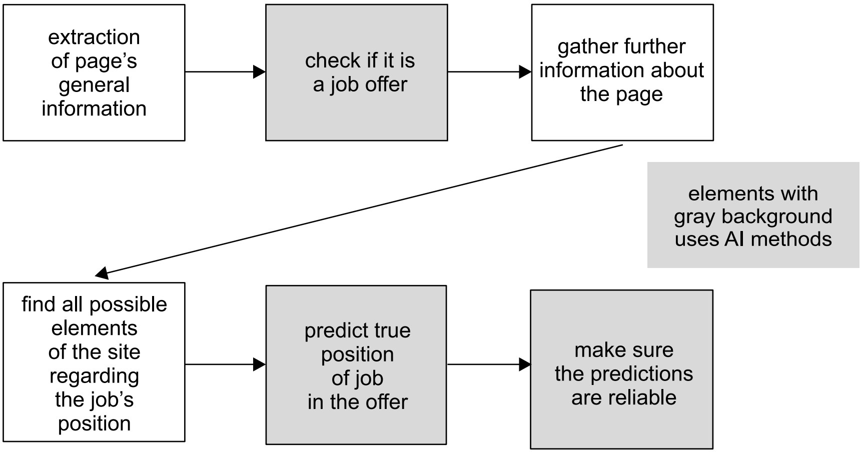

extraction of page's general information
check if it is a job offer
gather further information about the page
elements with gray background uses AI methods
find all possible elements of the site regarding the job's position
predict true position of job in the offer
make sure the predictions are reliable

Fig. 3. Diagram of the entire system

It can be seen that the count attribute containing information about the number of occurrences of the word or group of words gained the highest relevance for Random Forests algorithm. This may be due to the fact that the repeated appearance of a job title increases the probability of its correct classification. In addition to this parameter, elements related to tags and raw text length are also of great value.

The second algorithm considered in the study was XGBoost. As in the previous case, here, too, the evaluation consisted of appropriate selection of hyperparameters so that the final model achieves the highest possible scores in the previously defined measures. Again, only the parameter determining the maximum depth of the tree proved to be the most valuable in terms of changes in the model's efficiency level. For comparability of results when evaluating the XGBoost method, the values of max_depth was determined as: 4, 8, 12, 16 and 20 as well. Table 3 summaries obtained results.

Table 3

Results of XGBoost

Depth	Precision [%]	Recall [%]	F1 [%]	Accuracy [%]	Fit_time [s]
4	94.86	94.87	94.86	94.86	3.33
8	95.75	95.74	95.73	95.73	5.77
12	96.08	96.08	96.06	96.06	9.13
16	96.19	96.19	96.17	96.17	13.80
20	96.18	96.18	96.17	96.17	16.22

As in the case of Random Forest, it can also be observed for XGBoost that as the value of max_depth increases, the measures that determine the effectiveness of the model also increase. However, the differences between successive values are small, much smaller than in the case of Random Forest. In addition, it can be seen that the XGBoost algorithm runs relatively faster and achieves minimally better results, where for max_depth=20 Random Forests reported accuracy = 96.02%, while XGBoost 96.17%.

An interesting thing can be observed when considering the results achieved for the feature importance. The obtained values are summarized in Figure 4.

As can be observed, the most important features are related to categorical data, for which one hot encoding has been applied in data preprocessing. A feature that is related to the indication of the main document descendant (HTML child tag) was considered the most important. In addition, elements related to tags and xpath have also been identified as significant. Comparing the relevance of features in the two considered methods, it should be noted that a large part of them overlap (count, tag_a, tag_li, link_pos, child_h2, xpath), however, they have a significantly different impact on the training of the studied algorithms.

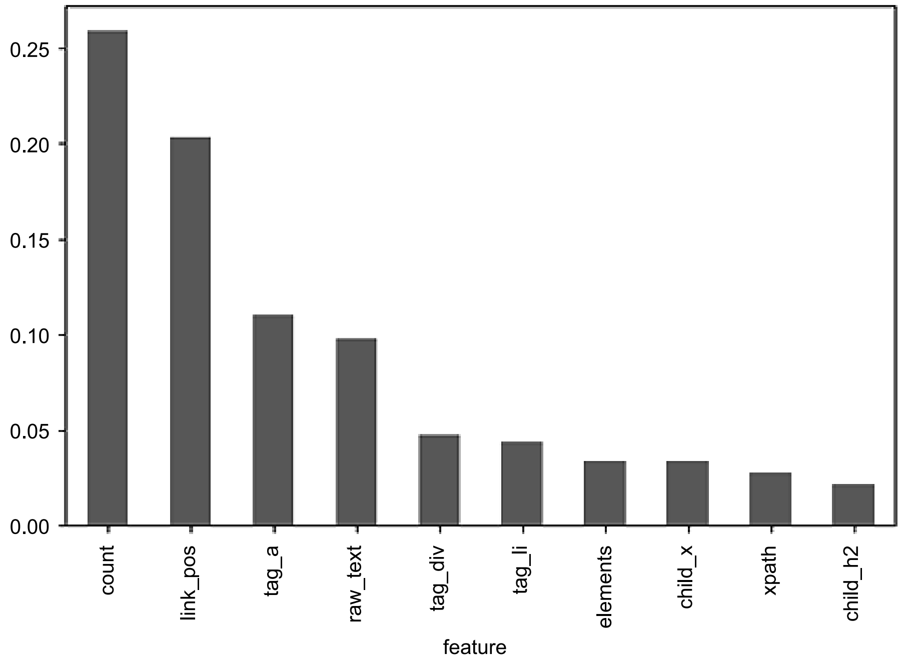


Fig. 4. Importance of the key features in Random Forests

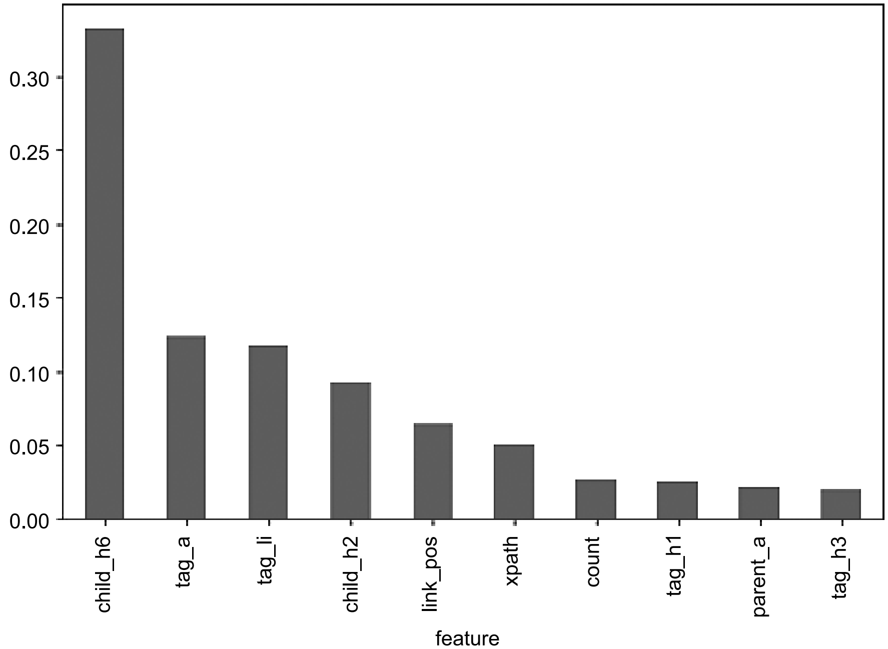


Fig. 5. Importance of key features in XGBoost

Impact of granularity on the effectiveness of models

The second part of the experiments involved examining the impact on model quality when the dataset size is significantly reduced using a granularity algorithm. The use of granulation reduced the dataset by about 500 times, that is, only 0.2% of the original dataset was considered when training the Random Forests and XGBoost algorithms. The results for each max_depth value are shown in Tables 4 and 5, respectively.

Table 4

Results for Random Forests after granulation

Depth	F1			Accuracy			Fit_time		
	full [%]	reduced [%]	diff [%]	full [%]	reduced [%]	diff [%]	full	reduced	times
4	91.52	71.47	20.05	91.52	83.50	8.02	12.34	0.15	83.40
8	93.73	90.68	3.05	93.73	92.78	0.95	10.96	0.16	70.69
12	94.79	90.91	3.88	94.79	92.78	2.01	17.39	0.23	76.20
16	95.56	89.54	6.02	95.56	92.78	2.78	14.13	0.23	60.42
20	96.02	89.54	6.48	96.02	92.78	3.24	17.28	0.21	84.03

Table 5

Results for XGBoost after granulation

Depth	F1			Accuracy			Fit_time		
	full [%]	reduced [%]	diff [%]	full [%]	reduced [%]	diff [%]	full	reduced	times
4	94.86	88.02	6.84	94.86	90.72	4.14	3.332	0.040	84.02
8	95.73	86.97	8.76	95.73	90.72	5.01	5.771	0.055	104.36
12	96.06	88.24	7.82	96.06	91.75	4.31	9.129	0.062	147.49
16	96.17	88.24	7.93	96.17	91.75	4.42	13.796	0.062	224.32
20	96.17	88.24	7.93	96.17	91.7	4.42	16.224	0.065	251.15

The tables contain the main measures of model quality (F1 and accuracy) for runs trained on the full dataset and on the reduced dataset. In addition, the training time has been reported. For each pair of F1 and accuracy values, the difference by how many percentage points the obtained results differed was calculated and for training time a value was determined – how many times faster the training ran on the reduced dataset.

For runs of the Random Forests algorithm, it can be seen that a huge drop in model quality can be observed for max_depth = 4, where the value of the F1 measure is lower by 20% and accuracy by 8%. On the other hand, considering all the other training runs, it should be noted that the decreases in model quality, especially for the accuracy measure, are small and range from 1% to 3%.

When taking into account the training execution times, it can be seen that reducing the dataset significantly speeds up the processing of the input data. The choice of hyperparameters does not play such an important role, where the acceleration varies between 60 and 84 times.

Other regularities can be observed when processing the XGBoost algorithm. Regardless of the hyperparameter settings, the results obtained for the accuracy measure on the reduced dataset are 4-5% weaker than for the full dataset, and for the F1 measure this difference is 7-8%. Taking into account the training time, a significant acceleration can be observed with an increase in the parameter max_depth, where for the value max_depth = 4 the reported acceleration is 84 times, and for the maximally considered value max_depth = 20 more than 250 times.

Conclusions

The main objective of this paper was to determine the feasibility of using machine learning algorithms to determine the correctness of job name recognition in job postings collected by crawlers implemented at Emplocity. In particular, specific HTML tags were taken into account, which, according to the authors, could affect the training quality of algorithms based on decision trees: Random Forests and XGBoost. The first part of the study determined the quality of the models and the most relevant features considered for training, where the best results for the accuracy and F1 measures reached 96%, which proves the correctness of models for the problem under consideration. The second part of the study concerned the possibility of reducing the dataset through granulation algorithms. After reducing the input dataset and using only 0.2% of the original data, the decrease in the accuracy measure ranged between 1 and 5% for most cases and 3-8% for the F1 measure. On the other hand, considering the training time, it decreased from 60 to as much as 250 times. Thus, it can be concluded that the granularity method used is suitable for the research conducted.

Acknowledgments

This paper is about the promotion of the results of the project „Development of autonomous artificial intelligence using the learning of deep neural networks with strengthening, automating recruitment processes" co-financed by the European Union.

References

Artiemjew P., Ropiak K. 2021. *A Novel Ensemble Model – The Random Granular Reflections*. Fundam. Informaticae, 179(2): 183-203.

Chang Y.J, Tsai K.L., Jiang W.C., Liu M.K. 2023. *Content-aware malicious webpage detection using convolutional neural network*. In *Multimedia Tools and Applications*, p. 1-19. https://doi.org/10.1007/s11042-023-15559-8

Chen T., Guestrin C.E. 2016. *XGBoost: A Scalable Tree Boosting System*. In: KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 785-794. https://doi.org/10.1145/2939672.2939785

Drozda P., Talun A., Bukowski L. 2019. *Emplobot – design of the system*. In Proceedings of the 28th International Workshop on Concurrency, Specification and Programming.

Finn A., Kushmerick N., Smyth B. 2001. *Fact or fiction: Content classification for digital libraries*. In Proc. Joint DELOS-NSF Workshop, Personalization Recommender Syst. Digit. Libraries.

Hashemi M. 2020. *Web page classification: a survey of perspectives, gaps, and future directions*. Multimed Tools Appl, 79: 11921-11945. https://doi.org/10.1007/s11042-019-08373-8

Ho T.K. 1995. *Random decision forests*. Proceedings of 3rd International Conference on Document Analysis and Recognition, 1: 278–282. https://doi.org/10.1109/ICDAR.1995.598994

Kao A., Poteet S. 2006. *Natural Language Processing and Text Mining*. Springer, Berlin.

Kim Y.S., Lee C.K. 2016. *An Empirical Evaluation of Job Classification Using Online Job Advertisements*. In AI 2016: Advances in Artificial Intelligence. LNCS, 9992. https://doi.org/10.1007/978-3-319-50127-7_65

Leśniewski S. 1916. *Podstawy ogólnej teoryi mnogości*. I. Prace Polskiego Koła Naukowego w Moskwie, Sekcya Matematyczno-Przyrodnicza, No. 2, Zakład Wyd. Popławski. Eng. tr. in S. Leśniewski. 1992. *Collected Works*. Kluwer, Dodrecht, p. 129-173.

Lotfi C., Srinivasan S., Ertz M., Latrous I. 2021. *Web Scraping Techniques and Applications: A Literature Review*. In R. Pal, P.K. Shukla (eds), *SCRS Conference Proceedings on Intelligent Systems*. SCRS, India, p. 381-394. https://doi.org/10.52458/978-93-91842-08-6-38

Nowicki R.K, Starczewski J.T. 2017. *A new method for classification of imprecise data using fuzzy rough fuzzification*. Information Sciences, 414. https://doi.org/10.1016/j.ins.2017.05.049.

Parvez M.S., Tasneem K.S.A., Rajendra S.S., Bodke K.R. 2018. *Analysis of Different Web Data Extraction Techniques*. International Conference on Smart City and Emerging Technology (ICSCET), p. 1-7. https://doi.org/10.1109/ICSCET.2018.8537333

Pawlak Z. 1982. *Rough sets*. International Journal of Computer & Information Sciences, 11: 341–356.

Polkowski L. 2007. *Granulation of knowledge in decision systems: The approach based on rough inclusions. the method and its applications*. LNAI, 4585, proceedings for RSEISP 2007: Rough Sets and Intelligent Systems Paradigms, p. 69-79.

Qi J. 2012. *Random Forest for Bioinformatics*. In: *Ensemble Machine Learning*. Springer, New York. https://doi.org/10.1007/978-1-4419-9326-7_1

Rabbi J. 2021. *How long does it take to land a new job and how to reduce this time*. Retrieved from https://www.linkedin.com/pulse/how-long-does-take-land-new-job-reduce-time-juliana (2.03.2021).

Ropiak K., Artiemjew P. 2018. *A Study in Granular Computing: Homogenous Granulation*. 24th International Conference, ICIST 2018, Vilnius, Lithuania, October 4-6, pp. 336-346. Proceedings. https://doi.org/10.1007/978-3-319-99972-2_27

Shete D., Bojewar S., Sanghvi A. 2021. *Survey Paper on Web Content Extraction & Classification*. 6th International Conference for Convergence in Technology (I2CT), pp. 1-6. https://doi.org/10.1109/I2CT51068.2021.9417947

TALUN A., DROZDA P., BUKOWSKI L., SCHERER R. 2020. *FastText and XGBoost ContentBased Classification for Employment Web Scraping*. In: *Artificial Intelligence and Soft Computing*, ICAISC 2020. https://doi.org/10.1007/978-3-030-61534-5_39

TREVISO M., LEE J.-U., JI T., VAN AKEN B., CAO Q., CIOSICI M.R., HASSID M., HEAFIELD K., HOOKER S., RAFFEL C., MARTINS P.H., MARTINS A.F.T., FORDE J.Z., MILDER P., SIMPSON E., SLONIM N., DODGE J., STRUBELL E., BALASUBRAMANIAN N., DERCZYNSKI L., GUREVYCH I., SCHWARTZ R. 2023. *Efficient Methods for Natural Language Processing: A Survey*. Transactions of the Association for Computational Linguistics, 11: 826-860. https://doi.org/10.1162/tacl_a_00577

ZOU X.-Q., ZHANG P., HUANG C.-Y., BAO X.-G. 2019. *Malicious Websites Identification Based on Active-Passive Method*. CNCERT 2018. Communications in Computer and Information Science, 970. https://doi.org/10.1007/978-981-13-6621-5_9