IJEEC

International Journal of Electrical Engineering and Computing
Vol. 4, No. 2 (2020)

Original research paper
UDC 005.721/.722:[004.62:519.246.8
DOI 10.7251/IJEEC2002093C

Application of machine learning in the process of classification of advertised jobs

Branislava Cvijetic[1] and Zaharije Radivojevic[2]

[1]Agency for Statistics of Bosnia and Herzegovina, Sarajevo, Bosnia and Herzegovina
[2]University of Belgrade School of Electrical Engineering, Belgrade, Serbia

branislava.cvijetic@bhas.gov.ba, zaki@etf.bg.ac.rs

Abstract — **Institutions that provide official statistics tend to use external data sources such as administrative data sources besides regular statistical surveys. In addition to the mentioned data sources, Big Data became recognized as a new data source for the provider of official statistics. Classification of textual data is one of the elementary tasks for the provider of official statistics, regardless of data sources. In this paper, application of traditional machine learning algorithms, Multinomial Naive Bayes and Support Vector Machine, for the classification of advertised jobs according to ISCO-08, has been presented. The paper presents the methods of collecting data on advertised jobs from four websites and procedures for creating a multilingual dataset. There are different types of text preprocessing, such as converting uppercase letters into lowercase letters, stopword removal, punctuation mark removal, lemmatization, correction of commonly misspelled words, and reduction of replicated characters. We hypothesized that the application of different combinations of preprocessing methods influenced the text classification results. Two experiments had conducted to test the hypothesis. Both experiments results showed that using the Support Vector Machine algorithm on a created dataset gives better results than Multinomial Naive Bayes. Performed experiments showed that the proposed algorithms gave a good performance with an overall accuracy of up to 90% but with different accuracy for individual classes due to an imbalanced dataset.**

Keywords- Big Data; text classification; supervised machine learning

I. INTRODUCTION

Big Data can be described as the technology used to collect, process, and analyze large amounts of data. These data can be structured, semi-structured, and unstructured, and they can be generated and can come at high speed at different intervals (sometimes in real-time). All of the above makes these data very complicated for analysis. However, collecting and storing large amounts of data is not just what characterizes Big Data technology. The ability to process and analyze the collected data for further use, as well as the extraction of information from the given data, is something that makes this technology significant. Without the ability to analyze and the necessary tools to extract information from them, it would be just a mess of collected data. The possibility of applying Big Data covers various areas, such as analyzing the content of posts on social networks and even in politics for analyzing publicly available data on voter's opinions on the current state of society to create effective election campaigns [1].

The European Statistical System (ESS) and its strategic partners, in 2010 [2], recognized the importance of using Big Data in providing official statistics. In the last ten years, a lot of work has been done to create strategies, launch projects, and organize seminars to prepare statistical institutions for an adequate response to the challenge called Big Data. All of the above efforts can help statistical institutions to prepare themselves to integrate Big Data as a new source in providing official statistics. One component of the ESSnet Big Data project [3] refers to statistical estimates of advertised jobs on the Internet. Representatives of the official statistical institutions in Bosnia and Herzegovina did not participate in this project.

The automated classification (or categorization) of texts into classes has a long history, dating back to the early 1960s. However, the considerable increase of available documents in the last two decades, developments in machine learning algorithms, developments in deep learning algorithms inspired by the work of the human brain renewed interest in automated text classification. Initially, text classification focused on problem-solving by applying a set of rules based on the expertise of the domain experts [6], and today the focus is on methods of automatic text classification using machine learning algorithms [7], [8], [9], as well as deep learning algorithms [10].

One of the challenging tasks for providers of official statistics, regardless of data sources (e.g., regular statistical survey, administrative data, Big Data), is the automatic classification of textual data [4], [5]. The literature related to job classification is rooted in American tradition, i.e., predominantly for classifying job titles using the Occupational Information Network (O*NET) classification system. The U.S. Department of Labor, Employment and Training Administration introduced

This paper is a revised and expanded version of the paper, presented at the XIX International Symposium INFOTEH-JAHORINA 2020 [17]

the Occupational Information Network (O*NET) to the public in 1998. On the other hand, countries of the European Union, since 2009, have used the International Standard Classification of Occupations ISCO-08 [11]. Many job search websites, such as www.monster.com and www.careerbuilder.com, originate from the United States. On the other hand, job search websites like www.stepstone.com originate from Europe. According to the above, papers [12], [13] describe the job classification system used in the www.careerbuilder.com website. In the academic papers relevant to European countries [14], [15] the job description is mainly classified according to ISCO-08 job classification.

This paper describes the application of the machine learning algorithm for text classification on an unstructured set of data collected from the Internet, i.e., on advertised jobs data in the field of information and communication technologies. The collected datasets contain the job title in English, German, Serbian, or Greek. A multilingual dataset with the job title and the corresponding ISCO-08 code has been developed using the collected primary data. Many machine learning algorithms are used for text classification. In this paper, two machine learning algorithms, i.e., Multinomial Naive Bayes and Support Vector Machine, will be used. The algorithms are selected based on results presented in [8], [9], and [10]. This paper is the beginning of the research, with the purpose to investigate the use of machine learning algorithms for the classification of multilingual textual data. Obtained results can be used during the production of Experimental Statistics or as an extra tool for regular statistical surveys.

The paper is structured as follows. The second section presents a brief theoretical review of the text classification and machine learning algorithms used in the experimental part. The third section describes the process of collecting unstructured data from the Internet and the process of creating a training dataset based on publicly available data. The fourth section describes the experimental part of the paper. Finally, Section five offers the conclusion of this paper.

II. TEXT CLASSIFICATION AND MACHINE LEARNING ALGORITHMS

A. Text classification

The goal of text classification is to assign text string to one or more predefined classes or categories. Text classification has various applications, such as separating film reviews into positive or negative ones, separating e-mails into spam or legitimate e-mails, separating newspaper articles into politics, sports or culture. To each instance (movie review, e-mail, and newspaper article) from the above examples of application, it is possible to assign a class or a target value to which that instance belongs. If a set of target values contains exactly two-class, it is a binary classification of the text. An example of binary text classification is the separation of movie reviews into positive or negative ones. Multi-class text classification is a problem of classifying instances into more than two classes. An example of multi-class text classification is the division of film genres by mood into action, adventure, comedy, drama, fantasy, horror, thriller, etc. Multi-class text classification assumes that each instance has assigned one and only one class. For example, the film genre can be drama or horror but not, at the same time, drama and horror. In case the text belongs to several classes at the same time, then this is a multi-label classification. For this paper, text classification is defined as a process classifying text strings of different lengths (job titles) into twelve different categories (occupational classes according to the International Standard Classification of Occupation, ISCO-08) depending on the string content, using machine learning algorithms.

B. Algorithms for text classification

The most significant step in solving a text classification problem is to select an appropriate algorithm. There is a wide range of machine learning algorithms, and one should keep in mind that the type of data and the type of problem to be solved dictates the selection of machine learning algorithms. In practice, before selecting an algorithm for machine learning, it is advised to compare the results of different machine learning algorithms on a particular dataset and to consider the prediction performance and computational efficiency.

The following two algorithms have applied in the experimental part of this paper:

- Multinomial Naive Bayes MNB belongs to the set of Naive Bayes NB classification algorithms. Naive Bayes algorithms are machine learning algorithms based on applying the theorem of the English statistician and philosopher Thomas Bayes. The Bayes theorem describes the probability of an event based on the prior knowledge of the conditions that might be related to the event. Multinomial Naive Bayes is a specialized version of Naive Bayes. The term multinomial means that features follow a multinomial distribution. It means that there are more features/words for text classification. The algorithm counts the number of occurrences or calculates the relative frequency of each feature/word. Based on the counter or on the relative frequency with which a feature appears in a text string, multinomial NB classification algorithms classify a text string in class. One of the advantages of this algorithm is that it gives good results when there is not enough data. Also, it is possible to use it in a case when small computational resources exist [8].
- Support Vector Machine SVM is another popular algorithm used for text classification. Like Naive Bayes, SVM does not require much training data to give good results [8]. SVM is an algorithm that determines the best decision boundary between vectors that belong to a given category and vectors that do not belong to it. It can be applied to each type of vector which encodes any data type. Thus, texts need to be transformed into vectors to leverage the power of SVM text classification. Vectors are lists of numbers which represent a set of coordinates in space. So, when SVM determines the decision boundary, SVM decides where to draw the best "line" (or the best hyperplane) that divides the space into two subspaces: one for the vectors which belong to the given category and one for the vectors that do not belong to it.

III. CREATING A DATASET FOR MACHINE LEARNING

A labeled dataset is a group of samples tagged with one or more labels. That dataset is required to train the system using the supervised learning method. Created labeled dataset contains the name of the advertised job and occupation code according to the International Standard Classification of Occupations, ISCO-08. The text below describes the process of creating the primary dataset later used in Section 4.

A. Data collection

For creating a labeled dataset, it was necessary to collect primary data or obtain information directly from the source. Our data sources were four websites (www.stepstone.at, www.stepstone.de, www.jobfind.gr, poslovi.infostud.com) that advertise jobs in the field of information and communication technology. For collecting the title of the advertised job position, job position description, location, company name, and URL from the listed websites, the ParseHub web scraping tool was used (Fig. 1). Data were collected five times in the period from 14 December 2019 to 12 January 2020.

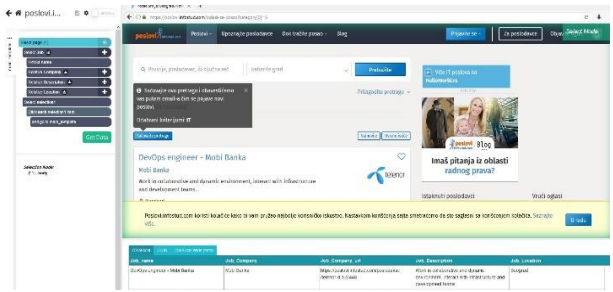

Figure 1. Example of the project in the ParseHub tool

B. Creating an initial dataset

Twenty collected primary datasets of advertised IT jobs were merged into one dataset. This dataset contains the following information: job title, city, job description, country, and data collection date. When creating the merged dataset, only records with a unique name, city, and job description were considered.

Table I shows the summary statistics for the number of advertised IT job titles from four websites by data collection dates, based on a merged dataset.

TABLE I. NUMBER OF ADVERTISED JOBS TITLE BY DATA COLLECTION DATES

	Data collection dates					**Total**
	14.12. 2019.	**22.12. 2019.**	**29.12. 2019.**	**05.01. 2020.**	**12.01. 2020.**	
www.stepstone.at	868	884	822	764	734	**4072**
www.stepstone.de	837	817	768	777	801	**4000**
www.jobfind.gr	149	140	120	110	128	**647**
poslovi.infostud.com	305	302	271	248	238	**1364**
	2159	**2143**	**1981**	**1899**	**1901**	**10083**

Based on the initial dataset, a dataset with 2984 unique job titles/descriptions was created. Job titles are in English, German, Greek, or Serbian. There are lots of possible reasons for the duplication of job titles. For example, the same job title had been advertised for five weeks (883 records in the initial dataset), the job titles with the same name exist on different websites, etc.

Combining the automatically labeled initial dataset and the second dataset marked manually by domain experts, the training data set was created. A labeled dataset is a dataset that consists of advertised job titles with assigned appropriate occupation codes according to ISCO-08.

As a starting point for automatic labeling advertised job titles, another dataset was used. This dataset contains 363 records with marked occupation codes manually assigned by domain experts. This small dataset has been analyzed first, after which the procedures for the automatic assignment of the ISCO-08 code were developed. In the last stage, the ISCO-08 code was automatically assigned to job titles in the initial dataset using the created procedures.

The training dataset was created in the following way:

1. A table of the abbreviations and synonyms has been created. The table contains a total of 98 records. (Fig. 2)

	abr_in	abr_out
1	Datenbank	database
2	Daten-bank	database
3	Netzwerk	network
4	sap basis	database
5	dba	database
6	baza podataka	database
7	hardware	network

Figure 2. Part of the table of abbreviations and synonyms

2. A rule control table was created. Rules are in the form of combinations of one or two keywords. Every rule has two levels that define the order in which the rules are executed. This table contains 73 records. (Fig. 3)

	Level1	Level2	Code	Word1	Word2
1	1	1	2521	administrator	database
2	1	2	2521	analyst	database
3	1	3	2521	architect	database
4	1	4	2514	developer	database
5	1	4	2521	specialist	database
6	6	1	2522	administrator	network
7	10	5	2522	engineer	network
8	10	10	2522	developer	network

Figure 3. Part of the rule control table

3. SQL procedures were applied to the dataset with manually assigned occupation codes. First, the abbreviations and synonyms table was used and the value “abr_in” was replaced with the created value “abr_out” in the job description column. After that, the occupation code for the job description was automatically assigned using the rule control table.
4. Occupation codes assigned automatically were compared with occupation codes assigned manually. The process of updating the two tables (from step 1 and step 2) stopped when the match ratio close-reached 90%. Also, running the SQL procedures from the third step on the second dataset stopped when the match ratio close-reached 90%.
5. SQL procedures use two tables from step 1 and step 2 (with the same number of records). After performing SQL procedures on the initial dataset, for 2054 records, occupation codes are assigned automatically. In other words, 68.83% of the total of 2984 records had automatically assigned occupation codes.
6. The datasets from step 4 (325 records) and step 5 (2054 records) were merged into the training dataset. Since there were overlaps in the records, unique records were singled out. The final dataset that will be used further in the experimental part consists of 2323 records. (Fig. 4)

	JobDescription	isco08
1	(Junior) Datenbankadministrator (m/w)	2521
2	Administrator baza podataka i aplikativnih platfor...	2521
3	Administrator baza podataka i aplikativnih platfor...	2521
4	Database Administrator	2521
5	Database Administrator (DBA)	2521
6	Database Administrator (m/f/d)	2521
7	Database Administrator "	2521
8	Database Analyst	2521
9	Database Architect / Developer (f/m/d)	2521
10	Database Architect/Engineer	2521
11	Database Engineer/Architect PostgreSQL (m/f/d)	2521

Figure 4. Part of the final training dataset

IV. EXPERIMENT

The Python 3.7.1 programming language through the Anaconda Navigator 1.9.7 was used for the experimental part. Anaconda is a collection of scientific Python packages, tools, resources, and IDEas. Individual' edition Anaconda is free and open-source. This package includes many tools that a data scientist can use for a machine learning project on a single machine. Python scripts, inside the Jupyter Notebook 6.0.3, were created. Jupiter Notebook is a powerful data analysis tool because it allows for writing lines of code one by one and running them one at a time rather than writing and rewriting an entire program. Changing the code and re-running the program is done in the same window. For manipulation or understanding of the textual string, the Natural Language Toolkit – NLTK 3.4. was used. This toolkit is one of the most vigorous libraries, as it contains packages that enable machines to understand human language and reply to it with an appropriate response. Scikit-learn 0.22.1 was used for machine learning. Scikit-learn is a Python module that integrates a wide range of state-of-the-art machine learning algorithms for supervised and unsupervised problems.

A. Dataset analysis

Firstly it is fundamental to explore the training dataset and learn about it when encountering a text classification problem. The process of analyzing a dataset includes identifying and removing inaccurate and irrelevant data, dealing with the missing data, removing duplicate data, etc. Thus, text preprocessing is eliminating large inconsistencies, enabling more efficient work with data.

The Python software library pandas 1.0.5. was used for the training dataset analysis, while the dataset has visualized using the Orange software 3.23.1.

The analysis revealed the following:

- The dataset has two columns. The first column contains the advertised IT job titles, while the second column contains the code according to the ISCO-08. During the creation of the dataset, all duplicate records and records without assigned occupation codes had removed.
- The dataset contains 12 classes according to ISCO-08.
- The dataset is imbalanced, e.g., the dataset does not represent all classes of data equally. (Fig. 5)
- The dataset consists of 2323 records with 11772 unique words. Fig. 6 shows the Word Cloud, i.e., visual representations of Words that give greater prominence to words that appear more frequently for the training dataset.

The difficulty with imbalanced datasets is that standard classification learning algorithms are often biased towards the majority class, so there is a higher misclassification rate for the minority class instances. In the worst case, minority classes are treated as exceptions and neglected. In some cases (for example, in health care), it is necessary to balance datasets artificially to predict rare diseases using classifiers. There are several techniques to do this, but this will not be discussed in this paper. In this experiment, classifiers work with a potential statistical data source. The risk that algorithms ignore minority classes, such as classes 2523 or 3341, and give better results for majority classes, can be accepted.

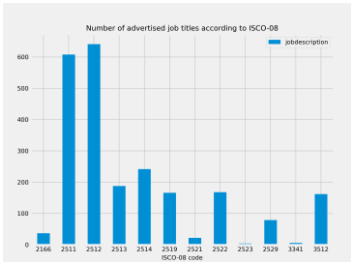

Figure 5. Distribution of ISCO-08 codes in the training dataset

Figure 6. Word Cloud for the training dataset produced in the Orange software

B. Text preprocessing

Text preprocessing transforms text into a form that facilitates and improves the operation of the machine learning algorithm. Generally, there are three main components of texts processing:

- Text tokenization represents the splitting of text strings into smaller parts or *tokens*. For example, documents could split into paragraphs, paragraphs into sentences, and sentences into words. It is also called text segmentation or lexical analysis.
- Text normalization is the process of converting raw text into a convenient, consistent, and standard form, depending on the type of data and application. Converting all words to lowercase is an example of text normalization. Stemming and lemmatization are text normalization techniques. Stemming is a technique used to extract the base form of the words by removing affixes from them. Lemmatization takes into consideration the morphological analysis of words. The output from lemmatization is called a *lemma*. In stemming root word is called a stem. In general, the stemming technique only looks at the word form, while the lemmatization technique looks at the word meaning.
- Filtering aims to obtain a cleaner text. It is achieved, for example, by removing empty strings, special characters, numbers, etc. Filtering removes certain words, the so-called stop words, from the text, i.e., words with little or no meaning in the dataset. Stop words can be, for example, conjunctions or some other specific textual string that exists in the text.

Preprocessing the collected IT job titles involves removing certain special characters, numbers, whitespace characters, and stop words for English, Greek, and German.

Some specific stop words that are not relevant to the occupation classification were removed from the dataset. Stop words list was created for the experiments, it involves position-related words (senior, junior, head, lead, expert, assistant, student), words related to job requirement (experience, knowledge, professional, German), name of locations (Athens, Stockholm, Greece, Germany, Wien, Berlin, Thessaloniki, Belgium), words for working hours (vollzeit, teilzeit, part, gender, etc.), etc.

Job titles in the datasets are unstructured text sentences, so tokenization was done on the word level using NLTK libraries. The technique which converting all words to lowercase had applied as normalization techniques.

The word count was reduced from 11772 to 8064 words after text preprocessing. Fig. 7 shows the Word Cloud for the processed text. Comparing the two results from Fig. 6 and Fig. 7 can be seen the effects of the applied text processing.

Figure 7. Word Cloud for the training dataset after text processing produced in the Orange software

C. Feature extraction

Machine learning algorithms operate in a numeric feature space, expecting input as a two-dimensional array where rows are instances and columns are feature. For performing machine learning on the text, it is necessary to transform the text string into vector representations so that it possible to apply machine learning algorithms. This process is called feature extraction or vectorization. Some of the most popular feature extraction methods are bag-of-words and term frequency-inverse document frequency (tf-idf). Combining linguistic techniques from NLTK and machine learning techniques in Scikit-learn, 335 features were extracted using the frequency-inverse document frequency method. In the experiment, the parameter min_df, for the feature extraction, was set on value 5. A word that appears at least five times in the entire dataset is considered a feature. The default value for ngram_range has changed on (1, 2). An n-gram is just a string of n words in a row. Analyzing job names in the preprocessed dataset was realized that many job names are two-word long (for example, data scientist, data engineer, data analyst, etc.). Unigrams and bigrams have been using through the process of feature extraction.

D. Creating a text classification model

Implementing text classification with Python could be a difficult task, especially when creating a classifier from scratch. Luckily, there are open-source resources that could help to carry out this without a lot of problems. In this experiment, the Scikit-Learn open-source library has used. This library contains many machine learning algorithms, including the Multinomial Naive Bayes and Support Vector Machine. In the following, Multinomial Naive Bayes and Support Vector Machine are called MultinomialNB and LinearSVC. Usually, the implementation of a multi-class classification machine learning algorithm involves the following steps: importing libraries, fetching the dataset into a panda data frame (Fig. 8), dataset preprocessing, extracting and scaling features, train-test dataset splitting, training the model, calculating the model score using the appropriate metric, and saving the model for future use.

	jobdescription	isco08
0	Software QA Engineer	2519
1	(Agile) Fullstack .NET/C# Web Developer (m/f/d)	2513
2	(Junior) Application Manager (w/m/x) für Finanzapplikationen	2511
3	(Junior) Cloud Engineer for Digital Platform Operations (f/m/d)	2512
4	(Junior) Consultant Agile Methoden/Scrum Master (m/w/d)	2511
...	...	...
2318	Υπάλληλος Υποστήριξης Εφαρμογών (Applications Support)	3512
2319	ΥΠΑΛΛΗΛΟΥΣ ΓΡΑΦΕΙΟΥ (full time) Scanning & Data entry	3341
2320	Υπεύθυνη / Υπεύθυνος E-shop & Social manager	2511
2321	Υπεύθυνος Support - Τεχνικός Η/Υ	3512
2322	Φωτογράφος Eshop - Graphic Designer	2166

2323 rows × 2 columns

Figure 8. Dataframe before text preprocessing

Since only one dataset was available, it was divided into train and test datasets using the Scikit-Learn train_test_split function. Setting parameter test_size on 0.3, samples (in this experiment, column "jobdescription" (Fig. 8)) were divided into 30% for the test dataset and 70% for the training dataset. The training dataset was used to build the model. A test dataset was used to validate the trained model.

LinearSVC and MultinomialNB models were tested throughout two experiments. In the first experiment, while features were extracted from the dataset, the stop words list was taken into consideration. In the second experiment, the stop words list was neglected. In the following text, the results are presented respectively as Case I and Case II.

E. Model evaluation

After doing the usual feature engineering, selection, implementing the model, and getting some output in the form of a probability or a class, the next step is to investigate how effective the model is, based on some metric, using testing datasets. The metric explains the performance of the model.

For multiclass classification problems, classifier performance typically was defined according to the confusion matrix associated with the classifier. According to [16], a confusion matrix is a table where each cell [i,j] indicates how often label j was predicted when the correct label was i. Thus, the diagonal entries (i.e., cells [i,j]) indicate labels that were correctly predicted, and the off-diagonal entries indicate errors.

Entities of the confusion matrix are:

- True Positive (TP) refers to the number of predictions where the classifier correctly predicts the positive class as positive.
- True Negative (TN) refers to the number of predictions where the classifier correctly predicts the negative class as negative.
- False Positive (FP) refers to the number of predictions where the classifier incorrectly predicts the negative class as positive.

- False Negative (FN) refers to the number of predictions where the classifier incorrectly predicts the positive class as negative.

For a 3x3 confusion matrix (Fig. 9), entities of the confusion matrix are: TP=c_{11}, FN=c_{12}+c_{13}, FP=c_{21}+c_{31} and TN=c_{22}+c_{23}+c_{32}+c_{33}. The created dataset contains 12 classes. All parameters for them were calculated in Python, using Scikit-Learn metrics and a Python Seaborn 0.10.1 visualization library.

		Predicted		
		Class 1	Class 2	Class 3
Actual	Class 1	c_{11}	c_{12}	c_{13}
	Class 2	c_{21}	c_{22}	c_{23}
	Class 3	c_{31}	c_{32}	c_{33}

Figure 9. A 3x3 confusion matrix

Based on the entries of the confusion matrix, it is possible to compute the most common performance measures such as:

- Accuracy gives the overall accuracy of the model, meaning the fraction of the total samples that were correctly classified by the classifier.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- The misclassification rate tells what fractions of predictions were incorrect. It is also known as Classification Error.

$$Error = (1 - Accuracy) = \frac{FP+FN}{TP+TN+FP+FN} \quad (2)$$

- Precision tells what fraction of predictions of positive classes were positive.

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

- Recall tells what fraction of all positive samples were correctly predicted as positive by the classifier.

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

- F1-score combines precision and recall into a single measure. Mathematically it's the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for classification problems. It is a probability curve that plots the True Positive Rate/Recall against False Positive Rate.

- False Positive Rate was calculated as the ratio between the number of negative fractions wrongly categorized as positive (false positives) and the total number of the actual negative fractions.

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

AUC stands for Area under the ROC Curve. It measures the entire two-dimensional area underneath the entire ROC curve from (0, 0) to (1, 1).

F. Model evaluation Case I

Fig.10.a shows a confusion matrix for the LinearSVC classifier, while Fig.10.b shows a confusion matrix for the MultinomialNB classifier.

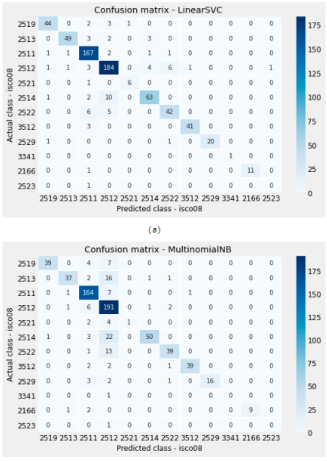

Figure 10. A 12x12 confusion matrix: (a) LinearSVC model, (b) MultinomialNB model

Fig.11.a shows a classification report for the LinearSVC classifier, while Fig.11.b shows a classification report for the MultinomialNB classifier.

	precision	recall	f1-score	support
2519	0.92	0.88	0.90	50
2513	0.96	0.86	0.91	57
2511	0.88	0.97	0.92	173
2512	0.89	0.92	0.90	201
2521	0.86	0.86	0.86	7
2514	0.89	0.83	0.86	76
2522	0.84	0.79	0.82	53
3512	0.98	0.93	0.95	44
2529	1.00	0.91	0.95	22
3341	1.00	1.00	1.00	1
2166	1.00	0.92	0.96	12
2523	0.00	0.00	0.00	1
accuracy			0.90	697
macro avg	0.85	0.82	0.84	697
weighted avg	0.90	0.90	0.90	697

(a)

	precision	recall	f1-score	support
2519	0.97	0.78	0.87	50
2513	0.93	0.65	0.76	57
2511	0.87	0.95	0.91	173
2512	0.72	0.95	0.82	201
2521	1.00	0.14	0.25	7
2514	0.96	0.66	0.78	76
2522	0.89	0.74	0.80	53
3512	0.97	0.89	0.93	44
2529	1.00	0.73	0.84	22
3341	0.00	0.00	0.00	1
2166	1.00	0.75	0.86	12
2523	0.00	0.00	0.00	1
accuracy			0.84	697
macro avg	0.78	0.60	0.65	697
weighted avg	0.86	0.84	0.83	697

(b)

Figure 11. Classification report: (a) LinearSVC model, (b) MultinomialNB model

Accuracy alone is not enough for an evaluation of the classifier. If a dataset has an unequal number of observations for each class, or if a dataset has more than two-classes, just observing the accuracy can lead to a wrong estimation classifier. For both classification models, accuracy is greater than 80%. From this parameter value, it is unknown if the reason for the high accuracy value is that all classes were equally well predicted, or the model neglected one or two-classes. The classification report shows accuracy and recall as additional assessment parameters, and therefore gives a better idea of which classification model is more accurate. In the case of high values for precision and recall, that is an indicator that the classifier returns correct results (Precision) and returns most of all positive results (Recall). The ideal system of high precision and high recall returns many correctly marked results.

As shown in Fig.11.a, the LinearSVC algorithm produces better similarity between precision and recall than the MultinomialNB algorithm (Fig.11.b). The MultinomialNB algorithm has large deviations for the precision and recall

parameters for most classes. Therefore it is possible to say that LinearSVC gives better results than the MultinomialNB algorithm.

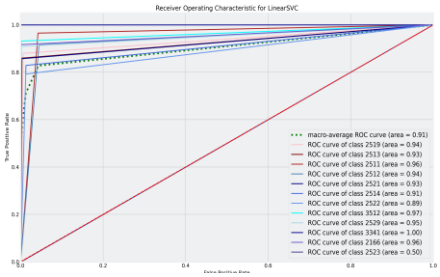

Figure 12. A ROC curve for the LinearSVC model

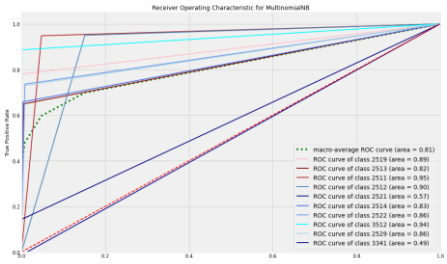

Figure 13. A ROC curve for MultinomialNB model

ROC curves and the area value from Fig. 12 and Fig. 13 shows that the LinearSVC algorithm gives better results for most classes against the MultinomialNB algorithm. The LinearSVC, for each class code, generates an AUC value higher than 0.9, which means there is a 90% chance that the model will be able to distinguish between positive class values and negative class values. For the MultinomialNB model, we can create a ROC curve for ten classes. The reason for this is the fact that just ten class codes exist in the test dataset. After splitting the small imbalanced training dataset using a train_test_split function, only ten classes exist in the test dataset.

G. Model evaluation Case II

In order to study the impact of stop words on the classification results another experiment was done. During extraction features, stop words did not filter. The number of extracted features increased from 335 (Case I) to 459 (Case II).

As can be seen from data in Fig. 11 and Fig. 15 that, in the case of using stop words, the LinearSVM has improvements in results, but the MultinomialNB has small degradation results for accuracy. ROC curves and the area value from Fig. 16 and Fig. 17 shows that the LinearSVC algorithm gives better results for most classes against the MultinomialNB algorithm.

In both experiments (Case I and Case II), the LinearSVM model gives better results than the MultinominalNB model. Also, the LinearSVM algorithm improves overall results by using a generated list of stop words on the dataset used in the experiment.

The algorithms tested using a small dataset. Due to the short time frame for data collection, it is significant to check the validity of the results in case of a longer time frame for data collection or data collection from other websites.

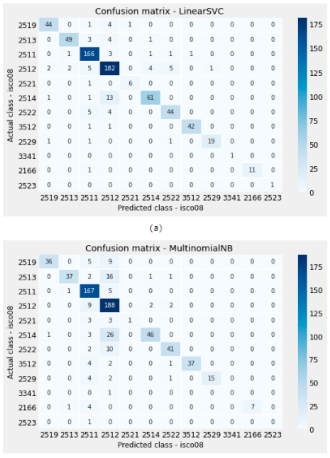

Figure 14. A 12x12 confusion matrix: (a) LinearSVC model – no stop words, MultinomialNB model –no stop words

	precision	recall	f1-score	support
2519	0.92	0.88	0.90	50
2513	0.94	0.86	0.90	57
2511	0.90	0.96	0.93	173
2512	0.86	0.91	0.88	201
2521	0.86	0.86	0.86	7
2514	0.91	0.80	0.85	76
2522	0.86	0.83	0.85	53
3512	0.98	0.95	0.97	44
2529	0.95	0.86	0.90	22
3341	1.00	1.00	1.00	1
2166	1.00	0.92	0.96	12
2523	1.00	1.00	1.00	1
accuracy			0.90	697
macro avg	0.93	0.90	0.92	697
weighted avg	0.90	0.90	0.90	697

(a)

	precision	recall	f1-score	support
2519	0.97	0.72	0.83	50
2513	0.95	0.65	0.77	57
2511	0.82	0.97	0.89	173
2512	0.72	0.94	0.81	201
2521	1.00	0.14	0.25	7
2514	0.94	0.61	0.74	76
2522	0.89	0.77	0.83	53
3512	1.00	0.84	0.91	44
2529	1.00	0.68	0.81	22
3341	0.00	0.00	0.00	1
2166	1.00	0.58	0.74	12
2523	0.00	0.00	0.00	1
accuracy			0.82	697
macro avg	0.77	0.57	0.63	697
weighted avg	0.85	0.82	0.82	697

(b)

Figure 15. Classification report for LinearSVC model – no stop words

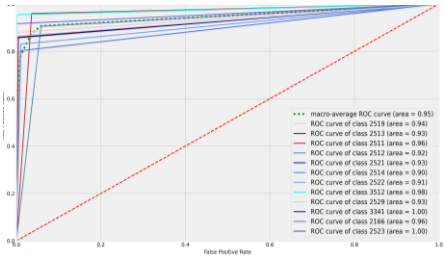

Figure 16. A ROC curve for the LinearSVC model – no stop words

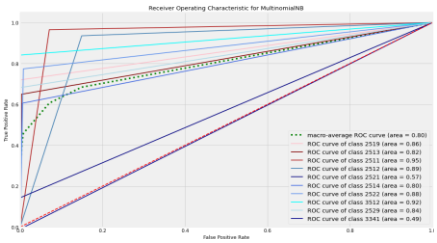

Figure 17. A ROC curve for MultinomialNB model – no stop words

V. CONCLUSION

This paper presented the use of MultinomialNB and LinearSVC machine learning algorithms for text classification on a labeled dataset. Training dataset was created for this paper. This dataset contains the name of the advertised job and the occupation code assigned according to ISCO-08. The created dataset is multilingual, holds the most job titles in English, but there are also job titles in Serbian, Greek, or German. Due to the specificity of the dataset, a specific stop word list was created.

Experimental results showed the successful use of machine learning algorithms for classifying advertised jobs. Both algorithms give good results on the analyzed dataset. However, according to the evaluation parameters, the LinearSVC algorithm provided better results. The results showed that these two algorithms could ignore classes in an imbalanced dataset.

Future work will attempt to apply the acquired knowledge of text classification to a larger dataset, i.e., on a dataset with occupation codes which are not included in the dataset used in this paper. To achieve this goal, it is necessary to create a new training dataset. In future work, depth learning algorithms will be used for text classification on the existing dataset, as well as deep learning algorithms will be used on an extended version of this dataset. One of the further investigations machine learning algorithms will be applied for text classification in the production of Experimental Statistics. It is significant to explore another possible use of machine learning algorithms to classify textual data collected by regular statistical surveys.

ACKNOWLEDGMENT

The Ministry of Education, Science and Technological Development of the Republic of Serbia partially supported this paper (III44009). The authors are grateful for the provided financial support.

REFERENCES

[1] S. Sudhahar, G. Veltri, and N. Cristianini, "Automated analysis of the US presidential elections using Big Data and network analysis," Big Data & Society, vol. 2, 2015.

[2] Collaboration in Research and Methodology for Official Statistics, "Big Data," https://ec.europa.eu/eurostat/cros/content/big-data_en, [accessed 15 January 2020]

[3] European Statistical System, "Essnet Bif Data," https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/ESSnet_Big_Data [accessed 15 January 2020]

[4] H. Gweon, M. Schonlau, L. Kaczmirek, M. Blohm, and S. Steiner, "Three Methods for Occupation Coding Based on Statistical Learning," Journal of Official Statistics, vol. 33, 2017.

[5] A. Bethmann, M. Schierholz, K.Wenzig, and M. Zielonka, "Automatic Coding of Occupations. Using Machine Learning Algorithms for Occupation Coding in Several German Panel Surveys," WAPOR 67th Annual Conference, 2014.

[6] F. Thabtah, P. Cowling, and Y. Peng, "MCAR: Multi-class Classification based on Association Rule," International Conference on Computer Systems and Applications, AICCSA, IEEE, 2005.

[7] S. Fatima, and B. Srinivasu, "Text Document categorization using support vector machine," International Research Journal of Engineering and Technology, IRJET, vol. 4, 2017.

[8] K. Kowsari, K. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," Information, 2019.

[9] V. K. Vijayan, K. R. Bindu, and L. Parameswaran, "A comprehensive study of text classification algorithms," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1109-1113, DOI: 10.1109/ICACCI.2017.8125990.

[10] A. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," Springer, 2019.

[11] European Commission, "Commission Recommendation of 29 October 2009 on the use of the International Standard Classification of Occupations (ISCO-08)," https ://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1396369855234&uri=CELEX:32009H0824 [accessed 1 March 2020]

[12] F. Javed, Q. Luo, M. McNair, F. Jacob, M. Zhao, and T.S. Kang, "Carotene: A Job Title Classification System for the Online Recruitment Domain," IEEE First International Conference on Big Data Computing Service and Applications, 2015.

[13] F. Javed, M. McNair, F. Jacob, and M. Zhao, "Towards a Job Title Classification System," ArXiv, 2016.

[14] R. Boselli, M. Cesarini, F. Mercorio, and M. Mezzanzanica, "Classifying online Job Advertisements through Machine Learning," Future Generation Computer Systems, 2018.

[15] K. Tijdens, and C. Kaandorp, "Classifying job titles from job vacancies into ISCO-08 and related job features - the Netherlands," Technical Report, 2019.

[16] S. Bird, E. Klein, and E. Loper, "Natural Language Processing with Python," O'Reilly Media, 2009, p. 240

[17] B. Cvijetić, and Z. Radivojević, "Primjena mašinskog učenja u procesu klasifikacije oglašenih radnih mjesta," 19th International Symposium INFOTEH-JAHORINA, Jahorina, Bosna i Hercegovina, pp.146-151, 2020

Branislava Cvijetic was born in Sarajevo, Bosnia and Herzegovina, in 1978. She received B.Sc. and M.Sc. degrees from the Faculty of Electrical Engineering, University of East Sarajevo, in 2002 and 2013, respectively. Currently working at the Agency for Statistics of Bosnia and Herzegovina. The main topics of interest are data analysis and machine learning.

Zaharije Radivojevic received his BSc (2002), MSc (2006), and Ph.D. (2012) degrees in Computer Engineering from the University of Belgrade, School of Electrical Engineering. He is currently an associate professor at the University of Belgrade, School of Electrical Engineering at the Department of Computer Engineering and Information Theory, teaching several courses on computer architecture and organization, e-business infrastructure, and mobile device programming. His research interests include computer architecture and organization, concurrent and distributed programming, data analysis, simulations, and reverse engineering.