Movies Reviews Sentiment Analysis and Classification

Mais Yasen, Sara Tedmori
Department of Computer Science
Princess Sumaya University for Technology
Amman, Jordan
mai20130045@std.psut.edu.jo, s.tedmori@psut.edu.jo

Abstract- **As humans' opinions help enhance products efficiency, and since the success or the failure of a movie depends on its reviews, there is an increase in the demand and need to build a good sentiment analysis model that classifies movies reviews. In this research, tokenization is employed to transfer the input string into a word vector, stemming is utilized to extract the root of the words, feature selection is conducted to extract the essential words, and finally classification is performed to label reviews as being either positive or negative. A model that makes use of all of the previously mentioned methods is presented. The model is evaluated and compared on eight different classifiers. The model is evaluated on a real-world dataset. In order to compare the eight different classifiers, five different evaluation metrics are utilized. The results show that Random Forest outperforms the other classifiers. Furthermore, Ripper Rule Learning performed the worst on the dataset according to the results attained from the evaluation metrics.**

Keywords- Sentiment Analysis; IMDB Reviews; Tokenization; Stemming; Feature Selection; Classification; Random Forest.

I. Introduction

Humans are subjective creatures and their opinions are important because they reflect their satisfaction with products, services and available technologies. Being able to interact with people on that level has many advantages for information systems; such as enhancing products quality, adjusting marketing and business strategies, improving customer services, managing crisis, and monitoring performances [1].

A movie review is an article reflecting its writers' opinion about a certain movie and criticizing it positively or negatively, which enables everyone to understand the overall idea of that movie and make the decision whether to watch it or not. A movie review can affect the whole crew who worked on that movie. A study illustrates that in some cases, the success or the failure of a movie depends on its reviews [2]. Therefore, a vital challenge is to be able to classify movies reviews to capture, retrieve, quantify and analyze watchers more effectively [3].

Movie reviews classification into positive or negative reviews is connected with words occurrences from the reviews text, and weather those words have been used before in a positive or a negative context. These factors help enhance the review understanding process using Sentiment Analysis (SA), where SA has become the gateway to understanding consumer needs [4].

SA, also referred to as opinion mining, is concerned with identifying and categorizing opinions -which are subjective impressions not facts- expressed in a text and determining whether the writer's feelings, attitudes or emotions towards a particular topic are positive or negative [4]. SA is also defined as the process of transferring concrete data to subjective data, and it can be performed on different levels (document, sentence, or aspect).

The process of SA includes tokenization, word filtering, negation handling, stemming, and classification. Tokenization is the identification of the basic units by the process of segmenting text into sentences and words. Tokenization is considered a pre-processing step. Text needs to be segmented into linguistic units; such as words, numbers and punctuations, before performing any processing [5]. Words in English are usually separated by white spaces, and sentences are separated by full stops. Errors in tokenization are very dangerous because they will result in more errors in subsequent steps [6].

Stemming is the process of removing prefixes and affixes to convert the word into its stem or root form [7].

One vital data mining function is classification, which builds a model for labeling testing data based on previous training data. Different measures can be used to evaluate this model, such as accuracy, Area Under the Curve (AUC), F-measure, recall, and precision. Assigning classes (negative or positive) to reviews can be done by such model which predicts the label of new data. Some of the classification algorithms that has proven their efficiency in previous works are Naïve Bayes (NB), K-nearest Neighbors (KNN), Bayes Net (BN), Ripper Rule Learning (RRL), Support Vector Classifier (SVM), Random Forest (RF), Stochastic Gradient Descent (SGD), and Decision Tree (DT).

This research addresses SA of movies reviews as a classification task. Different classification algorithms are considered and compared to assess their performances for the task at hand. The reason why NB, BN, DT, KNN, RRL, SVM, RF, and SGD classifiers were chosen to be compared with each other is that these algorithms are supervised classifiers that have proven their efficiency and reliability in SA based

978-1-5386-7942-5/19/$31.00 ©2019 IEEE

on the previous works studied. Furthermore, these classifiers are the most popular to be used tackling SA.

The contribution of this research can be summarized by:

1. Using a real reviews dataset from IMDB which contained almost 43 thousand instances for training and testing.
2. Using eight different well-known classifiers; NB, BN, DT, KNN, RRL, SVM, RF, and SGD for evaluation in SA for the first time.
3. Comparing the results using different evaluation metrics.

The paper is structured as follows: Section II describes the related work in the area of SA. Section III includes the methods used in the development. Section IV shows the proposed methodology. Section V presents the experiments and the results of the proposed approach, and Section VI concludes the research and discusses future work.

II. Related Work

The authors of [8] addressed SA at the document-level and proposed the use of a combination of supervised and unsupervised algorithms and rich sentiment content for learning word vectors (also known as words description) capturing techniques, which includes continuous, multi-dimensional sentiments and non-sentiment annotations. For evaluation, the authors used the IMDB movie reviews dataset and their model got better performance in comparison to other learn vectors capturing techniques.

In [9], the authors presented a model to classify movie reviews as “thumbs up” or “thumbs down”. They proposed a text-categorization method using machine learning, and found the difference between subjectivity detection and polarity classification. The results showed that using subjectivity detection lead to shorter reviews, in Naive Bayes classifier the subjectivity detection was more effective in comparison to the original document without any subjectivity detection. Also, the minimum-cut classification enhanced the accuracy.

According to [10], references to the movie in weblog posts and the movie financial success are important factors. The results showed that positive sentiment is more efficient for movies domain with small number of existing reviews, which was not a good indication for building a model based on sentiment only, where sentiment could perform better in conjunction with other factors such as movie genre and season.

The authors of [11] tackled SA of text in a multilingual system. They used the lexical resources in the English SentiWordNet. With the aid of a translation software, the authors first translated different languages to English. Then, they classified them into “positive” or “negative” by searching for sentiment holding words such as adjectives. The authors used Amazon’s German movie reviews and compared their work to a statistical polarity classifier based on n-grams. The results reflected that their approach was good for multilingual SA.

In [12], the authors proposed a framework for classifying Web forum reviews in multiple languages (English and Arabic). The authors used entropy weighted genetic algorithm (EWGA) to improve the performance. For evaluation a movie review dataset was used and results showed that using EWGA with SVM gave higher performance in comparison to other feature selection methods, with accuracy of over 91%.

In [13], the authors presented a feature-based heuristic for SA of IMDB movie reviews using an aspect-oriented scheme. The authors then combined all aspects scores and generated a net sentiment profile. Using a SentiWordNet with two feature selection methods, the sentiment analysis of documents belonging to each movie was found and compared to Alchemy API. The results showed that their approach gave higher accuracy in comparison to simple document-level SA.

In [14] the authors used four classifiers; Maximum Entropy (ME), Naive Bayes (NB), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) for SA on the IMDB dataset. They used precision, recall, f-measure, and accuracy for evaluation. The results showed that using a combination of unigram, bigram, trigram gave better accuracy in all the classifiers used.

The authors of [15] proposed using Naive Bayes (NB) and Support Vector Machines (SVM) classifiers, and the use of a modified SVM using NB log-count ratios. The results showed that NB was better than SVM in short documents. In longer documents however, SVM was better. Their modified algorithm gave good results. Also, the use of logistic regression instead of SVM gave the same results, and the use of word bigram gave a consistent gain on SA.

The authors of [16] proposed using a multi knowledge-based approach to produce a feature class-based automatically in a movie review analysis system. Their approach combined statistical analysis, WordNet, and knowledge of movies. The results showed that their approach was effective.

As mentioned in [17], the authors worked on enhancing the Naive Bayes accuracy for SA. The proposed approach can be specified to a certain number of string categorizations to enhance accuracy. The results showed that the combination of negation handling, feature selection, and word n-grams improved the accuracy, using a Naive Bayes classifier which had a linearly increasing time of training and testing. On the IMDB dataset the accuracy was 88.80%.

As discussed in [18] two SA methods were proposed to label reviews as positive, negative, or neutral. The first method is to label reviews using the number of negative and positive words, and to extend the term-counting using external resources. The second method is to use Support Vector Machine (SVM). The authors applied three valence shifters; which are negations that invert the label polarity of a text, intensifiers to increase and decrease positivity or negativity of a term, and diminishers. The results showed that the term-counting method got higher accuracy, and the accuracy of SVM was very high.

From the related work studied it can be deduced that using feature selection in SA is an open field, with NB and SVM as the most commonly used classifiers.

This research proposes a model for SA of movie reviews. Different evaluation measures were considered such as recall, accuracy, AUC, precision, and f-measures. The model will be evaluated using 8 different well-known classifiers in an inclusive study which enables us to judge on which classifier results in a better performance more reliably and accurately. As far as the authors are aware, this work is the first effort that aims to compare the use of NB, BN, DT, KNN, RRL, SVM, RF, and SGD classifiers in SA, using different evaluation metrics.

III. Methods Description

The methods descriptions of this research are included in this section.

A. Text Tokenization

Text tokenization is segmenting text into sentences and words by specifying the basic linguistic units; words, numbers and punctuations [5]. In English language words are usually separated by white spaces.

Sentence tokenization is dividing a string into sentences. In English, punctuations especially the full stop character are indications of a sentence ending [19]. However, the full stop character can be used for abbreviations, which does not always terminate the sentence. To prevent such problem, a table of abbreviations is used. Sentence tokenization was applied using NLTK [20], which is trained on many languages including English. The training includes the identification of punctuation and characters that appears in the end of a sentence and the beginning of a new sentence.

In NLTK [20], word tokenization is a wrapper function that uses the Treebank Word Tokenizer, and splits out punctuations other than periods.

B. Word Filtering

After tokenization, unexpected words that will not affect the process of classification were removed. Firstly, regular expressions which are sequence of characters that illustrates a search pattern to find and replace unwanted words were used [21]. Secondly, unwanted words were removed manually.

C. Stemming

Stemming is the process of removing prefixes and affixes to convert the word into its stem or root form [7]. Porter stemming algorithm is used, which is a rule-based algorithm introduced by Martin Porter [22]. It defines a word consonant as any letter other than vowels. Form 1 shows how to calculate the conditions in this algorithm, where the square brackets denote optional content, and (VC)m denote a Vowel (V) followed by a Consonant (C) m times.

$$[\mathrm{C}](\mathrm{VC})\mathrm{m}[\mathrm{V}] \tag{1}$$

This algorithm follows a list of rules that contain patterns with their conditions, the rules follow the following form:

$$S1 \rightarrow S2 \tag{2}$$

if the pattern matches, and the word ends with the suffix S1, the suffix is transformed from S1 to S2 and the algorithm restarts from the beginning of the list to find the next matching pattern. If no pattern matches, then the algorithm outputs the result.

D. Attribute Selection

Gain ratio was used as the attribute selection algorithm, which can be defined as the rate of information gain to the essential information. The attributes with a high count of values are the most important in gain ratio. Gain ratio takes the attribute with the highest gain value and uses it to split attributes, which reduces the number of features.

Equation (3) [23] is used to calculate the expected information.

$$I(T) = -\sum_{i=1}^{n} \frac{freq(r_i.T)}{|T|} \times log_2\left(\frac{freq(r_i.T)}{|T|}\right) \tag{3}$$

Where T is the training data and |T| is the total number of records, r_i represents a specific value for each feature and the freq is all the possible values in that feature, where i goes from 1 to n (all possible values of the candidate feature).

Equation (4) [23] calculates the essential information for a specific value of split (S).

$$IS(T) = \sum_{j=1}^{m} \frac{|T_j|}{|T|} \times I(T_j) \tag{4}$$

Where $|T_j|$ represents all the possible values of attribute number j, and m is the number of possible attributes.

Equation (5) [23] gives the value of information gain of split (S).

$$G(S) = I(T) - IS(T) \qquad (5)$$

Equation (6) [23] calculates the information gain ratio between the information gain and the essential information.

$$GR(S) = \frac{G(S)}{IS(S)} \qquad (6)$$

E. Classification

To evaluate the proposed model, eight different well-known classifiers were run on the same training and testing datasets. The classifiers could be summarized as mentioned below:

- **Naïve Bayes (NB):** This classifier has two probabilities: P(class) which is the probability an input will produce a certain class, and P(input_condition|class) is the probability an input feature has a certain value, given the class. Otherwise, default probability is 0.
- **Decision Tree (DT):** A classifier model that gives labels to tokens based on a tree structure, where tree branches represent conditions on features, and tree leaves represent the label.
- **Support Vector Classifier (SVM):** A classifier that deals with missing values, normalizes nominal features to binary features. It formalizes all features by default. The output coefficients are found using the normalized form of features.
- **Bayes Network (BN):** In this classifier learning is done using search algorithms and quality measures. BN provides conditional probability distributions.
- **K-nearest Neighbors (KNN):** This classifier does distance weighting and is capable choosing the K value using cross-validation.
- **Ripper Rule Learning (RRL):** A classifier that uses RIPPER to gradually prune its propositional rule learner, to decrease error.
- **Random Forest (RF):** The underlying data structure of the forest classifier is a decision tree, but with random selection of features to split on.
- **Stochastic Gradient Descent (SGD):** A classifier used with many linear models (SVM, logistic regression, squared, Huber, and epsilon-insensitive losses). It changes missing instances and changes nominal attributes. Furthermore, it normalizes data features. A high rate of learning is needed by both Epsilon-insensitive and Huber loss.

IV. Proposed Approach

In this research, the researchers present SA model and classification algorithms for the classification of IMDB movies reviews.

The execution steps that are shown in Fig. 1 could be summarized as the following:

1. Retrieving IMDB reviews datasets.
2. Labeling datasets into POS/NEG classes.
3. Sentence tokenization.
4. Word tokenization: String to Word Vector.
5. Remove unwanted words using regular expressions.
6. Remove unwanted words manually.
7. Stemming.
8. Attribute Selection using Gain Ratio.
9. Split data into training and testing.
10. Classifying the data using 8 different classifiers and comparing their results using different metrics.

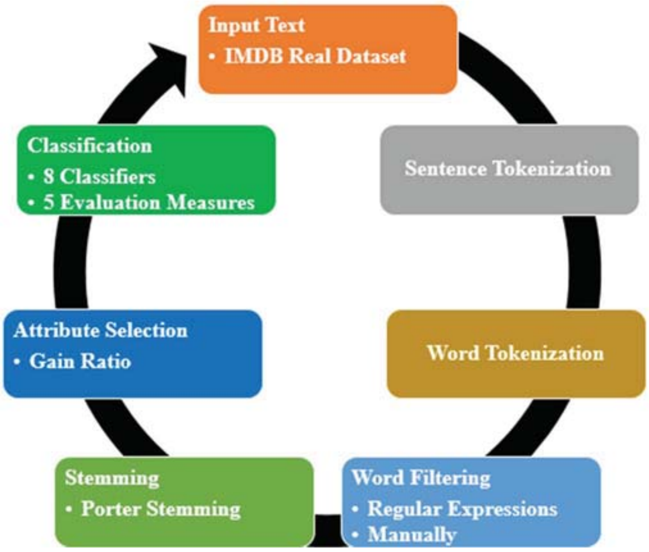

Fig. 1 Proposed Approach

V. Experiments and Results

Experiments conducted to evaluate the performance of the proposed approach are demonstrated in this section.

A. Data

For purposes of testing the performance of the proposed approach, the IMDB reviews dataset was used [24]. This dataset represents a group of movie reviews, and it contains 42926 review instances along with their binary classification (positive or negative). Where in each data file the first line represents the headers of which the description of the attributes is in. A “\n” is assigned for missing field.

B. Experiments settings

First, attribute selection using Gain Ratio was performed on the dataset to consider only the attributes that are most relevant to the label feature to train and test our proposed approach. The total count of attributes before and after attribute selection with the label feature is shown in Table 1.

Next, the data was divided into two sections, training included 66% of the total instances in the dataset, and testing included 34% of the total instances in the dataset; the reason behind this split percentage is that it is the most commonly used split in research. Dataset instances distribution after dividing it is also shown in Table 1.

Table 1 Total Number of Instances and Attributes in the dataset

Training dataset	28331 instances	Positive instances = 14230
		Negative instances = 14101
Testing dataset	14595 instances	Positive instances = 7252
		Negative instances = 7343
Features before selection	1135	
Features after selection	896	

The classifiers used are provided by NLTK [20], and WEKA wrapper library for Python [25]. The settings of the classifiers could be summarized as the following:

- **NB**: The size of batch equals 100, and default probability was set to 0.
- **DT**: For pruning 1 fold was set, for tree growing 2 folds were set, and leaf instances number equals 2 (minimum value).
- **SVM**: The complexity feature equals 1, logistic regression was applied as the calibration.
- **BN**: alpha value equals 0.5 which is used for calculating the conditional probability, and hill climbing was applied as the search method.
- **KNN**: neighbors count equals one, search of nearest neighbor was done applying brute force, and window size equals 0.
- **RRL**: pruning was done using 1 fold, to grow the tree 2 folds were used, optimization executions number equals 2, and the rule instances weight equals 2.
- **RF**: seed was set to 1, number of execution slots was set to 1, bag size was set to 100, batch size was also set to 100, maximum depth was set to 0.
- **SGD**: seed was set to 1, epochs is set to 500, lambda is set to 0.0001, batch size was 100, loss function was Hinge loss, learning rate was set to 0.01.

For evaluation five measures were used: Precision, Recall, Accuracy, AUC and F-measure. These measures can be calculated applying the Equations 7 to 11, where TP stands for True Positives, TN stands for True Negatives, FP stands for False Positives, FN stands for False Negatives.

$$Accuracy = (\mathrm{TP} + \mathrm{TN}) / (TN + FN + TP + FP) \tag{7}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{8}$$

$$Recall = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{9}$$

$$F = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{10}$$

$$AUC = \frac{(1 - FPR) \times (1 + TPR)}{2} + \frac{FPR \times TPR}{2} \tag{11}$$

Where TPR stands for True Positive Rate (TPR = TP/(TP+FN)) and FPR stands for False Positive Rate (FPR = FP/(FP+TN)).

C. Results

As conducted from Table 2, RF got the best accuracy (96.01%) in comparison with all of the other classifiers. Moreover, it got the highest precision (0.93), f-measure (0.96) and AUC (0.96). Also, RF and KNN got the best recall (1.00) in comparison to all of the classifiers in the table where they achieved false negatives count of 0. Furthermore, DT got a very competitive recall (0.97), KNN also got a very competitive f-measure (0.93) and AUC (0.93).

Table 2 Results

Classifier	TP	FN	FP	TN	Accuracy %	Precision	Recall	Fmeasure	AUC
NB	5789	1507	1145	6153	81.83	0.84	0.79	0.81	0.82
DT	7088	208	1064	6234	91.28	0.87	0.97	0.92	0.91
SVM	6412	884	947	6351	87.45	0.87	0.88	0.88	0.88
BN	5697	1599	1106	6192	81.47	0.84	0.78	0.81	0.82
KNN	7296	0	1033	6265	92.92	0.88	1.00	0.93	0.93
RRL	5574	1722	1269	6029	79.51	0.82	0.76	0.79	0.80
RF	7296	0	583	6715	96.01	0.93	1.00	0.96	0.96
SGD	6399	897	1028	6270	86.81	0.86	0.88	0.87	0.87

To summarize the results, RF has proved its efficiency over 7 other classifiers where it got the best result in all of the evaluation measures taken into account, the accuracy of the 8 classifiers ranged from 79.51% to 96.01%. RRL performed the worst.

VI. Conclusion and Future Work

The research goal of this work is to address SA by constructing an approach that can classify movie reviews and then compare the results in an inclusive study of eight well-known classifiers. To evaluate the proposed model, IMDB reviews real dataset was utilized. Tokenization was applied on the dataset to transfer strings into word vector, then stemming was used to extract the root of the words, afterwards gain ratio was applied on the dataset as an attribute selection algorithm. Then, the data was split into training and testing datasets using the percentages 66%, 34% respectively. To evaluate the results accuracy, precision, f-measure, recall, and AUC were used.

The results showed that RF has proved its efficiency over 7 other classifiers where it got the best result in all of the evaluation metrics taken into consideration, KNN also was able to get a recall similar to RF and a very competitive f-measure and AUC. Furthermore, DT got a very competitive recall value. Finally, RRL got the worst result.

The authors wish to conduct a similar study on different languages specifically on Arabic. In addition, the authors wish to experiment with different SA methods in order to increase the accuracy of the results.

References

[1] Sampriti Sarkar, “Benefits of Sentiment Analysis for Businesses”, retrieved on: December 22, 2018, from: www.analyticsinsight.net.

[2] ACME, “The Significance of a Film Review”, retrieved on: December 22, 2018, from: www.revue-acme.com.

[3] Mshne, Gilad and Natalie Glance, (2006), “Predicting Movie Sales from Blogger Sentiment”, AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, PP 1-4.

[4] Bo Pang and Lillian Lee, (2008), “Opinion Mining and Sentiment Analysis”, Foundations and Trends in Information Retrieval, Volume 2, PP 1–135.

[5] Vinodhini, Chandrasekaran (2012), “Sentiment Analysis and Opinion Mining: A Survey”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, PP 1-11.

[6] CraigTrim, “The Art of Tokenization”, retrieved on: December 22, 2018, from: www.ibm.com.

[7] Lovins, Julie Beth, (1968), “Development of a Stemming Algorithm”, Mechanical Translation and Computational Linguistics, Vol. 11, PP 22–31.

[8] Andrew Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng, Christopher Potts, (2011), “Learning word vectors for sentiment analysis”, the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, PP 142-150.

[9] Bo Pang, Lillian Lee, (2004), “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”, the 42nd Annual Meeting on Association for Computational Linguistics.

[10] Mishne, Gilad and Natalie Glance, (2006), “Predicting Movie Sales from Blogger Sentiment”, AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs.

[11] Kerstin Denecke, (2008), "Using SentiWordNet for multilingual sentiment analysis", IEEE 24th International Conference on Data Engineering Workshop, Cancun, PP 507-512.

[12] Ahmed Abbasi, Hsinchun Chen, and Arab Salem, (2008), “Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums”, ACM Transactions on Information Systems (TOIS), Vol. 26, PP 1-34.

[13] Vivek Singh, R Piryani, Ashraf Uddin and Pranav Waila, (2013), "Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification," International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), PP 712-717.

[14] Abinash Tripathy, Ankit Agrawal, Santanu Kumar Rath, (2016), “Classification of sentiment reviews using n-gram machine learning approach”, Expert Systems with Applications, Vol. 57, PP. 117-126.

[15] Sida Wang and Christopher Manning, (2012), “Baselines and bigrams: simple, good sentiment and topic classification”, the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2, PP 90-94.

[16] Li Zhuang, Feng Jing, Xiao-Yan Zhu, (2006), “Movie Review Mining and Summarization”, the 15th ACM international conference on Information and knowledge management, PP. 43-50.

[17] Vivek Narayanan, Ishan Arora, Arjun Bhatia, (2013), “Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model”, Intelligent Data Engineering and Automated Learning – IDEAL, Springer, Vol. 8206.

[18] Alistair Kennedy and Diana Inkpen, (2006), “Sentiment Classification of Movie Reviews Using Contextual Valence Shifters”, Computational Intelligence, Vol. 22, PP 110-125.

[19] Jeffrey Reynar, (1998), “Topic Segmentation: Algorithms and Applications”, IRCS Technical Reports Series, Vol. 66, PP 1-189.

[20] NLTK, version: 3.3, retrieved on: October 20, 2018, from: www.nltk.org.

[21] Mark Lawson, (2003), “Finite Automata”, CRC Press, PP 98-100.

[22] M.F. Porter, (1980), “An algorithm for suffix stripping”, Program, Vol. 14, PP 130-137.

[23] SAS, (2015), “Visual Analytics 7.2”, SAS Institute Inc., Ch. 37, PP 281.

[24] Arunava, “IMDB Movie Reviews Dataset”, retrieved on: November 1, 2018, from: www.kaggle.com.

[25] python-weka-wrapper, version: 3.14, retrieved on: November 30, 2018, from: pypi.org.