technical transactions

No. 2020/030

computer sciences

sciendo

An evaluation of machine learning and latent semantic analysis in text sentiment classification

Justyna Miazga

justyna.miazga@up.krakow.pl | http://orcid.org/0000-0003-1094-2854

Tomasz Hachaj

tomekhachaj@o2.pl | http://orcid.org/0000-0003-1390-9021

Institute of Computer Science, Pedagogical University of Cracow

Scientific Editor: Włodzimierz Wójcik, Cracow University of Technology Press
Technical Editor: Aleksandra Urzędowska, Cracow University of Technology Press
Language Editor: Tim Churcher, Big Picture
Typesetting: Anna Basista, Cracow University of Technology Press

Received: June 20, 2020
Accepted: September 22, 2020

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Competing interests: The authors have declared that no competing interests exist.

Citation: © 2020 Miazga, Hachaj. An evaluation of machine learning and latent semantic analysis in text sentiment classification. *Technical Transactions*, e2020030. https://doi.org/10.37705/TechTrans/e2020030

Abstract

In this paper, we compare the following machine learning methods as classifiers for sentiment analysis: k – nearest neighbours (kNN), artificial neural network (ANN), support vector machine (SVM), random forest. We used a dataset containing 5,000 movie reviews in which 2,500 were marked as positive and 2,500 as negative. We chose 5,189 words which have an influence on sentence sentiment. The dataset was prepared using a term document matrix (TDM) and classical multidimensional scaling (MDS). This is the first time that TDM and MDS have been used to choose the characteristics of text in sentiment analysis. In this case, we decided to examine different indicators of the specific classifier, such as kernel type for SVM and neighbour count in kNN. All calculations were performed in the R language, in the program R Studio v 3.5.2. Our work can be reproduced because all of our data sets and source code are public.

Keywords: sentiment analysis, machine learning, sentiment classification, K-nearest neighbours, neural network, support vector machine

1. Introduction

Nowadays, access to the Internet is common and a lot of people decide to communicate and to state opinions using electronic media. In the present day, social media has an especially important role in providing information about various areas of life (Tripathy, Agrawal, & Rath, 2016). Sentiment analysis is a source of much important information. It can be great tool for preparing material for advertising purposes in various industries. Information can be collected about the dependence of the (positive or negative) reaction of the recipients to the hours when the ads are presented (Mattila & Salman, 2018). Another example is election-result prediction based on Twitter sentiment analysis (Ramteke, Shah, Godhia, & Shaikh, 2016). Sentiment analysis is a valuable tool for both psychology and sociology. Datasets originating from social networks can be used as support in detecting signs of anomalous or disturbing human behaviour, such as signs of depression which can be quickly diagnosed (Wang, et al., 2013). With the help of natural language processing (NLP) algorithms, one can collect information about products, movies, books and political grouping or analyse human reactions to various kinds of written text. Among most useful NLP methods is sentiment analysis, which can be used to estimate whether a certain sentence has a positive or negative sentiment. This task is especially difficult when data contains only written text without any paralinguistic features of nonverbal communication. One of the methods used to recognise the sentiment of sentences is deep convolutional neural networks including character to sentence convolutional neural network (CharSCNN). It uses two convolutional layers and extracts relevant features of sentences and words (Dos Santos & Gatti, 2014). Another possible way of identifying sentence context is with machine learning algorithms. The example can be the bag-of-words method which allows the selection of only important features by eliminating the irrelevant elements (Agarwal & Mittal, 2016). Classification using machine learning is also one of the most popular tools in spam filtering, advertising, search engines and loan qualification (Burrell, 2016).

Machine learning is one of the most powerful sources of knowledge in sentiment analysis. One popular machine learning method is SVM, which was used in this study. The created vector can easily be combined with other solutions like TFIDF. It obtains a high level of accuracy in sentiment prediction. This can be seen in our work and other research (Trsteniak, Mikac, & Donko, 2014). Another method for obtaining details about sentence sentiment are lexicon-based methods.

Our research examines the kNN, ANN and SVM machine learning methods. The original dataset is 50,000 movie reviews which were prepared and shared by scientists from Stanford University. The same data sets were tested and described in their publication 'Learning Word Vectors for Sentiment Analysis' (Andrew L. Maas, 2011). This research compares the effectiveness of machine learning methods such as SVM, TFIDF, the bag-of-words method and their own semantic model. They obtained 85% - 90% of accuracy in prediction. One of the methods which obtained a low result is latent dirichlet allocation (LDA), which assumes information about document topics. It obtained results about 66% of accuracy. They have used IMDB movie reviews which they collected and make free available. We used the same dataset, but we limited it to 5,000 opinions. Another difference is that we used four methods to classify the sentiment of sentences. We did not examine the bag-of-words classifier but we tested the kNN method which proved to be the least effective compared to random forest, SVM and ANN. Our research is of a similar subject, which is the comparison of efficiency methods in sentiment prediction modelling; however, we focused on the comparison of four chosen methods. Other research that is based on the IMDB dataset is sentiment analysis posts from the social media platform, Twitter. In that paper (D. Tang, 2014), sentiment classification was performed by various methods, but the most similar was SVM. In this case, a support vector

https://doi.org/10.37705/TechTrans/e2020030

machine was connected with uni-, bi- and trigrams. As can be seen, sentiment analysis needs classification. There can be a lot of methods to do this, but we decided to compare just four of them.

2. Materials and methods

This study focuses on the following machine learning methods for sentiment analysis: SVM, kNN, ANN, random forest. We used datasets with movie reviews marked either positive or negative. This is the set we used to investigate the accuracy of the mentioned methods. The best results were obtained by SVM. Detailed information is described in the following paragraphs.

2.1. The features generation

In this research, we use a freely available dataset with 5,000 positive and negative movie reviews. All of the collected words were sorted decreasing by their frequency and their rank was labelled. The ranking is based on the place obtained after sorting and is equal to the position in the table. Next, we calculated the term frequency (TF) and inverse term frequency (IDF) according to the following formulas (Soucy, 2005):

$$TF = \left(\frac{term\ frequency\ in\ document}{total\ word\ count\ in\ document} \right) \tag{1}$$

$$IDF = \ln\left(\frac{document\ numbers}{document\ numbers\ containing\ term} \right) \tag{2}$$

IDF determines the validity of individual words. Finally, we multiplied TF with IDF (TFIDF) which allows us to mark the weight of the word to all text.

Latent semantic analysis (LSA) is a statistical model that provides details about word similarity. LSA requires a matrix with the occurrences of words in sentences, paragraphs or documents. LSA uses singular value decomposition (SVD) where each of the words in the document is represented as a set of orthogonal factors. As a result, words are represented not as characters but as continuous values of factors.

We filtered words by ranking the 5,188 words that were most used in the analysed texts. We removed the first 10 words, (movie, the, film, one, like, story, time, good, just, see) because it is determined by dataset content and can give distorted results. This was achieved through the calculation of the following formula (Kruskal J. B., 1978):

$$rank\_subset = linear\ model(log_{10}\left(term\ frequency\right) + log_{10}\left(term\ rank\right) \tag{3}$$

Figure 1 presents the weight of the word for the whole text and compares it to its number of occurrences. The more often a given expression is found, the higher the ranking is. As a result, we have set of words which formed our lexicon for the purposes of the study. This serves as an indicator in preparing the model for predicting the sentiment of statements.

In the next step, we took 5,000 movie reviews, 2,500 positive and 2,500 negative. All of the words contained in the dataset were prepared by making all of the letters lower case and removing punctuation signs. After that, we prepared a corpus (collection of text documents) and made a vector sourced from our sentences. These data preparation, allowed us to develop a term document matrix containing 17,535,000 elements; next, we calculated the distance matrix (12,497,500 elements).

The distance matrix makes it possible to find dissimilarities between words by calculating distances between any points *i* and *j* in the matrix **X** with the coordinate points '*a*×*b*' according to formula (Borg & Groenen, 2005):

$$d_{ij}(X) = \sqrt{\sum_{m=1}^{b} \left(x_{im} - x_{jm}\right)^2} \tag{4}$$

Finally, we scaled the matrix using all possible spaces, which is 190. We have tried the SVM classifier for different scaling for 130, 150 and 190 spaces with various kernels – radial, linear, sigmoid and polynominal. The best effect was obtained for 190 dimensions. The details can be found in Table 1.

We did classical multidimensional scaling (MDS) of the data matrix. This is similar to principal components analysis (PCA) but in this case, there is need for a dissimilarity matrix which shows the distance between all pairs of objects. The formula below (5) shows how similarity can be found. The main goal of using this method is dimensionality reduction. It has to find all of the coordinates of **X** which, as was said before, are in the set *a*×*b*. The calculations are covered by the function called Kruskal Stress (Kruskal J. B., 1964):

$$Stress(X) = \sqrt{\frac{\sum_{i<j}\left(d_{i,j} - f(x_{i,j})\right)^2}{\sum_{i<j} d_{i,j}^{\ 2}}} \tag{5}$$

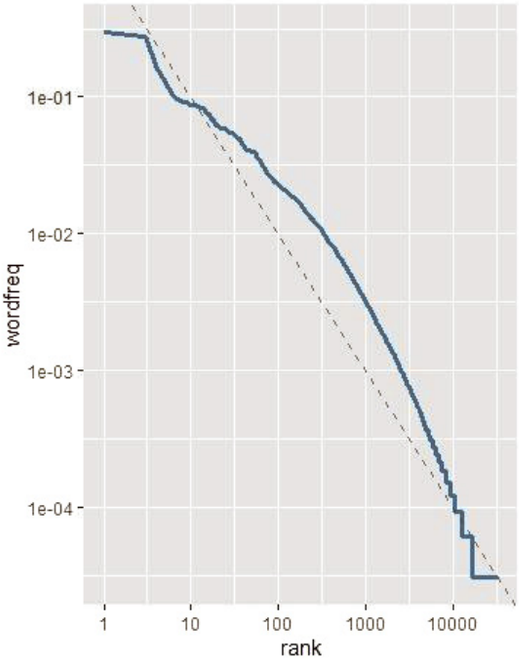

Fig. 1. The line chart shows word frequency and its ranking

Where d_{ij} equals (4) and $f(x,j)$ equals $(x_i - x_j)$ which allows minimisation of the stress function. Insufficient dimensionality can be a reason of non-zero stress. When the number of dimensions is too small, it may be impossible to obtain a valuable representation of the input data. The dataset can be represented by using *n* dimensions, where n is the number of scaled items and it has to be in range of 1 to $n - 1$. The increase of the dimensions number causes the stress function to either stay at the same level or to go down. In general, MDS allows us to visualise distances between samples, which gives information about similarities or dissimilarities between samples.

To summarise, analyses of the acquired sentences progress as follows:

1. In each sentence, all the letters are changed to lower cased and the punctuation is removed.
2. We filter out words which are not preserved by LSA analysers.
3. We obtain a vector of words that in our case has 5,188 dimensions in projected to 190-dimension space with MDS.
4. The feature vector generated for a given sentence is classified to either the positive or negative sentiment class using the appropriate classifier.

2.2. The classification

2.2.1. Support Vector Machine

At the start of our modelling prediction about the sentiment, we used the support vector machine (SVM) (Sebastiani, 2002). This is a very popular algorithm used during research focused on sentiment classification.

We have randomly chosen 75% from the dataset. We repeated the calculations 10 times to make our results comparable with others that can be found in the other publications. We tested it for the various kernels (Shimodaira, Noma, Nakai, & Sagayama, 2002):

- The 'radial' (radial basis function is also known as the Gaussian kernel) defined as:

https://doi.org/10.37705/TechTrans/e2020030

$$K(x_1, x_2) = \text{exponent}\left(-\gamma \left|x_1 - x_2\right|^2\right) \tag{6}$$

$|x_1 - x_2|$ is the Euclidean distance between these vectors, γ is the parameter that determines the "spread" of the kernel – in this research, it is 1/(data dimension).

- The 'polynomial' according to the formula:

$$K\left(x_1, x_2\right) = \left(a + x_1 x_2^{\ T}\right)^{degree} \tag{7}$$

'a' is default 0 and kernel degree, for us it is equal to 3.

- The 'sigmoid' calculated by:

$$K\left(x_1, x_2\right) = \tanh\left(\gamma x_1 x_2^{\ T} + a\right) \tag{8}$$

'γ' and 'a' as above.

- The 'linear', calculated by

$$K\left(x_1, x_2\right) = \left(x_1^{\ T} x_2\right) \tag{9}$$

As might be expected, the lowest average results for ten attempts was obtained for the experiment with 'polynomial' kernels. This is the result of leaving the degree feature as default, which is 3. This does not work well in natural language processing. More information about the results of attempt can be found in Table 1. The effect of predicting sentiment with kernels radial and linear are similar and are between 78 and 80% of accuracy. The details can be found in the results section.

Table 1. The average accuracy of 10 attempts for different kernel types and data dimensionality in the matrix for sentiment classification with SVM with default parameter values

Data dimensionality	Radial	Sigmoid	Polynomial	Linear
130	0.77	0.74	0.54	0.77
150	0.78	0.75	0.53	0.78
190	0.79	0.77	0.53	0.79

For the SVM classifier, the following parameters values were tested:

- kernel type – radial, sigmoid, polynomial, linear;
- gamma coefficient – for the linear kernel, the default value is 1, in other cases, it equals 1/data dimension (which in this case is 190);
- polynomial degree – available only in case of SVM with polynomial kernel, the default is 3.

Table 2. The accuracy of sentiment prediction with SVM for various gamma coefficient values with radial and sigmoid kernels

Gamma coefficient	Kernel type	
	radial	sigmoid
1/190	0.79	0.78
10/190	0.74	0.73
20/190	0.71	0.70
30/190	0.68	0.69
40/190	0.65	0.68
50/190	0.62	0.66
60/190	0.61	0.66
70/190	0.59	0.66
80/190	0.58	0.66
90/190	0.56	0.66
100/190	0.54	0.66

Table 2 presents results for different SVM coefficients with radial and sigmoid kernels. As can be observed, in both cases, the rising of the gamma coefficient causes a reduction in sentiment prediction accuracy. For both types of kernels, the highest accuracy was achieved in the case of gamma coefficient equal to 1/190. All other attempts show lower results.

In the case of polynomial kernels, the additional argument which needs to be considered is the polynomial degree. Details can be found in Table 3.

Table 3. Accuracy of SVM sentiment prediction with polynomial kernel

Gamma coefficient	Polynomial degree	Results
1/190	1	0.78
	2	0.55
	3	0.53
	4	0.53
	5	0.52
	6	0.52
	7	0.52
	8	0.51
	9	0.51
	10	0.51
	11	0.51
10/190	1	0.79
20/190		0.79
30/190		0.80
40/190		0.79
50/190		0.80
60/190		0.80
70/190		0.80
80/190		0.78
90/190		0.79
100/190		0.80
110/190		0.80

The highest result was achieved for the polynomial degree equal to 1. For this degree, we evaluated the gamma value in the range of 1/190 to 100/190. The accuracy of results were similar. For 10 attempts, the difference is not larger than 1%. The highest accuracy was achieved for the polynomial degree equal to 1 and a gamma coefficient of 100.

The results of the SVM prediction accuracy with linear kernels is the same for the gamma coefficient in the range of 1 to 100 and this equals .

2.2.2. Artificial Neural Network

ANN is another one of the most popular methods used in machine learning. It shows very good results in researching many different scenarios, for example, in social media analysis (Yan, 2017), image processing (Jifara, Jiang, Rho, Cheng, & Liu, 2019) or marketing efficiency (Salminen, Yoganathan, Corporan, Jansen, & Jung, 2019). The indisputable advantage of this method is its effectiveness in comparison with other methods that we have investigated. In our research, it achieved very stable results – for 10 attempts, the difference in the results was around 2%. A similar result was obtained for each of the ten attempts. The efficiency for determining positive and negative sentiment is evenly distributed.

https://doi.org/10.37705/TechTrans/e2020030

This is not so unequivocal with the next method, where the determination of negative words is less accurate and errors sometimes appear. The highest accuracy was obtained for 1 hidden neuron.

Table 4. ANN accuracy results with various neuron counts in the hidden layer and two types of activation function

Number of neurons in hidden layers	Types of activation functions	
	logistic	tanh
1	0.78	0.78
5	0.73	0.74
10	0.75	0.66

Table 4 presents the results of ANN prediction with various parameters. The numbers of neurons in the hidden layers are from 1 to 10 with logistic and tanh activation functions. The highest accuracy is 0.78 and has been obtained for 1 neuron in the hidden layer and logistic and tanh activation functions.

2.2.3. K-Nearest Neighbours

kNN is another way to classify the sentiment of sentences (Krouska, Troussas, & Virvou, 2016). The accuracy of results depends on parameter *k* and the distance between the neighbours which in this case is calculated by using the Euclidean distance formula. We tested *k* values in the range of 2 to 100. This method achieves a high level of accuracy with the labelling of positive words. It is more challenging for the negative class. This situation is known to scientists involved in sentiment analysis. This method obtained the lowest level of accuracy. This may be due to a lack of significant differences in the analysed data set. This means that this method is not accurate enough. Another disadvantage of the method is the time consumption of the data processing (Cox & Cox, 2008).

Table 5 contains information about the accuracy of the prediction made with the kNN classifier. One attempt was made for each value of k. In case of this research, *k* equals to 61 allows us to get the highest accuracy of 10 attempts.

Table 5. The accuracy of sentiment prediction using the kNN method for different values of *k*

Value of k	Accuracy
2	0.61
5	0.64
10	0.66
20	0.69
30	0.69
40	0.68
50	0.67
60	0.66
75	0.66
100	0.66

2.2.4. Random Forest

Another method for classifying sentiment is random forest. We used Breiman's random forest algorithm (Kruskal J. B., 1964). We examined this method with different numbers of trees and variables attempted in each split. Table 6 presents the results of attempts from 100 to 500 trees and 13, 20 and 100 variables tried at each split.

Table 6. The accuracy of random forest attempts with different numbers of trees and variables tried in each split

Number of trees	Number of variables tried at each split	Error rate (%)
100	13	30.19
	20	29.28
	100	27.55
200	13	27.84
	20	27.73
	100	27.92
300	13	26.91
	20	26.64
	100	27.09
400	13	26.29
	20	25.92
	100	26.61
500	13	26.51
	20	25.52
	100	26.64

As can be seen in Table 6, the error rate reduces with higher numbers of trees.

2.3. The implementation

To prepare this article, we used R language in 'R Studio' v 3.5.2. We decided to use this programming language because we used it in earlier research and it works well for sentiment analysis. The subsequent parts of the data evaluation required a variety of packages. For text processing, we used *'tm'*, *'ggplot2' 'devtools'*,*'tidytext'*, *'tokenizers'*. The specific methods require other packages, which are:

- Term Document Matrix, LSA – '*slam*'
- SVM – *'e1071'*
- ANN – '*neuralnet'*
- kNN – '*DMwR'*, *'class'*
- Random Forest – *'randomForest'*
- MDS – *'stats'*

3. Results

We used SVM, ANN, random forest and kNN to see which of them would be the most accurate. This means that the outcome for all attempts does not get a difference in the results above 3%. To gather this information we decided to use a confusion matrix which is exactly as presented in Table 7. We also examined the SVM, ANN and kNN methods and checked the prediction accuracy (see Table 8).

Table 7. Confusion Matrix used in this research.

Class	Negatives	Positives
Negatives	True Negatives	False Negatives
Positives	False Positives	True Positives

https://doi.org/10.37705/TechTrans/e2020030

For all of the tested methods, we collected the following parameters: true positive rate (TPR), true negative rate (TNR), accuracy (ACC), positive predictive value (PPV). The results for the individual parameters have been calculated in accordance with the following formulas (Santra, 2012):

$$TPR = \frac{TP}{P} \tag{10}$$

where P is all of the positive terms in the dataset.

$$TNR = \frac{TN}{N} \tag{11}$$

N is equal to all of the negative words in the dataset.

$$\text{ACC} = \frac{TP + TN}{P + N} \tag{12}$$

$$\text{PPV} = \frac{TP}{FP + TP} \tag{13}$$

Using these patterns allows us to assess the actual accuracy of the chosen solution. As mentioned, we repeated the attempts 10 times. Detailed information for all methods is available in Table 8. All of results have been rounded to two decimal places.

Table 8. Comparison of the average percentage accuracy for all methods

Method	TNR	TPR	PPV	ACC
SVM Radial	0.80	0.78	0.81	0.79
SVM Linear	0.80	0.78	0.80	0.79
ANN	0.77	0.78	0.77	0.78
kNN	0.78	0.60	0.89	0.65
Random Forest	0.75	0.73	0.76	0.74
SVM linear, ANN, kNN	0.83	0.74	0.85	0.78

Table 8 shows the average accuracy for all 10 tests of all methods. Accuracy is the highest for SVM with 'radial' kernels. Slightly lower results were obtained for the SVM classifier using linear kernels. Also worth noting are the results from ANN. As can be seen, the outcomes for all parameters (TNR, TPR, PPV and ACC) are similar. Their difference is around 0.5%. This means that the neural network successfully sets the label for positive and negative words. Random forest obtained an accuracy of around 75%. True Positive and True Negative values are similar for SVM, ANN and random forest. By contrast in the case of kNN, the true positive rate is much higher than the true negative rate. As mentioned before, kNN does not collect clear results. The positive label is marked randomly. This means that this method is not valuable for sentiment classification.

We decided to examine four classifiers – SVM linear, ANN, random forests and kNN – to check the accuracy of prediction in case of all of methods. The accuracy of all methods attempt is similar to ANN result which is 0.78. True negative rate (TNR) is equal to 83% which is the best value for all examined methods. The rest of the results are similar to other classifiers outcome. Because of this, more and more researchers try to use different algorithms to find information. This kind of examination allows us to get more information, that would otherwise be expensive and time consuming to obtain.

4. Discussion and conclusionss

The comparison of machine learning methods – SVM, ANN, kNN and random forest – allows us to indicate the one which has the highest accuracy in predicting sentiment of sentences. The best results we have obtained for SVM with 190 dimensions. It was almost 80% for three kinds of kernels. For other kernels, the difference between the results was at a level of ~2%.

The second method (ANN) obtained results equal to 78%, which is around 2% lower than in the case of SVM. The next method is random forest with an accuracy of 75%. The last classification method (kNN) turned out to be not effective enough. As can be seen in Table 8, the accuracy of this method is 65%, which is definitely insufficient. Another interesting point is that kNN is better at predicting negative sentences (TNR) (78%) whereas for positive words, it is only 60%. In the case of other methods, the assessment of positive and negative words is similar. All of the parameters (TNR, TPR, PPV, ACC) are almost equal.

Due to the matrix spaces being larger than 3 (in our research it is equal to 190) it is not possible to present the results in the form of a graph. The results contained in the tables show that the classification of sentiment using machine learning is quite accurate. The space vector machine, random forest and artificial neural network are reasonable solutions and give stable outcomes. The assessment of sentiment based on the closest neighbours does not work. The results obtained are random; therefore, this method should not be used in sentiment analysis.

To summarise, machine learning is a great source of knowledge in the field of obtaining information on the sentiment of statements. Before starting the sentiment analysis, you should carefully prepare the text for research. In this case, the solutions available at PCA (e.g. MDS) are very helpful in matrix preparation. Reliable results can be achieved using SVM, random forest and ANN. The kNN method should be rejected because it brings random results and predicting sentiment is not enough accurate for all labels (in this case, 'positive' and 'negative').

References

Agarwal, B., & Mittal, N. (2016). Machine Learning Approach for Sentiment Analysis. In *Prominent Feature Extraction for Sentiment Analysis* (pp. 21–45). Springer, Cham.

Andrew L. Maas, R. E. (2011). Learning Word Vectors for Sentiment Analysis. *49th annual meeting of the association for computational linguistics: Human language technologies*, 142–150.

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications.* Springer Science & Business Media.

Burrell, J. (2016). How the machine 'thinks': Understanding. *Big Data & Society*, 1–12. http://doi.org/10.1177/2053951715622512

Cox, M. A., & Cox, T. F. (2008). Multidimensional scaling. In Handbook of data visualization. In *Handbook of Data Visualization* (pp. 315–347). Berlin, Heidelberg: Springer.

D. Tang, F. W. (2014). Learning Sentiment-Specific Word Embedding. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, *1*, 1555–1565.

Dos Santos, C. N., & Gatti, M. (2014). Deep Convolutional Neural Networks for. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 69–78.

Jifara, W., Jiang, F., Rho, S., Cheng, M., & Liu, S. (2019). Medical image denoising using convolutional neural network: a residual learning approach. *The Journal of Supercomputing*, 704–718.

https://doi.org/10.37705/TechTrans/e2020030

Krouska, A., Troussas, C., & Virvou, M. (2016). The effect of preprocessing techniques on Twitter sentiment analysis. *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, 1–6.

Kruskal, J. B. (1964). *Nonmetric multidimensional scaling: A numerical approach.* Psychometrika.

Kruskal, J. B. (1978). *Multidimensional scaling.* Sage.

Mattila, M., & Salman, H. (2018). *Analysing Social Media Marketing on Twitter using Sentiment Analysis.* Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-229787 (access: 20/06/2020).

Miazga, J., & Hachaj, T. (n.d.). *Datasets and source code used in this article*. Retrieved from https://github.com/JusMia/sentimentanalysis_ML (August 20, 2020).

Ramteke, J., Shah, S., Godhia, D., & Shaikh, A. (2016). Election result prediction using Twitter sentiment analysis. *2016 international conference on inventive computation technologies (ICICT)*, Vol. 1, IEEE, 1–5.

Salminen, J., Yoganathan, V., Corporan, J., Jansen, B. J., & Jung, S.-G. (2019). Machine learning approach to auto-tagging online content for content marketing efficiency: A comparative analysis between methods and content type. *Journal of Business Research*, 203–217.

Santra, A. K. (2012). Genetic Algorithm and Confusion Matrix for Document Clustering. *International Journal of Computer Science Issues (IJCSI)*, *9*(1), 322–328.

Sebastiani, F. (2002). Consiglio Nazionale Delle Ricerche. *Machine learning in automated text categorization. ACM Computing Surveys*, *34*, 1–47.

Shimodaira, H., Noma, K.-I., Nakai, M., & Sagayama, S. (2002). Dynamic Time-Alignment Kernel in Support Vector Machine. *Advances in neural information processing systems*, 21–928.

Soucy, P. &. (2005, July). Beyond TFIDF weighting for text categorization in the vector space model. *IJCAI*, *5*, 1130–1135.

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126.

Trsteniak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based Framework for Text Categorization. *Procedia Engineering*, *69*, 1356–1364.

Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., & Bao, Z. (2013). A depression detection model based on sentiment analysis in micro-blog social network. *Pacific--Asia Conference on Knowledge Discovery and Data Mining*, 201–213.

Yan, B. Y. (2017). Microblog sentiment classification using parallel SVM in apache spark. *2017 IEEE International Congress on Big Data (BigData Congress)*, IEEE, 282–288.