Data Mining Techniques Assignment 1

Annemijn Dijkhuis 2686937, Sanne Donker 2585734 & Sam Verhezen 2686974

Vrije Universiteit Amsterdam, group 104

Task 1

At the start of the course "Data Mining Techniques" students were asked to fill in a questionnaire with different questions about their life. In this task, the dataset that arose from this questionnaire is analysed.

A: Exploration

The dataset contains 280 responses of students in this course, who answered to 16 questions. Of these attributes, 10 are categorical and 6 are numerical.

The data was preprocessed following a number of steps. First, the dates and times were cleaned and formatted to readable values. From the birthdate only the birthyears were kept, since these were provided by most participants. If the birthyear was not given, it was replaced with an NaN. The programme attribute was subdivided into classes: Computational Science (n = 30), Artificial Intelligence (n = 70), Econometrics (n = 26), Computer Science (n = 34), Business Analytics (n = 37), Bioinformatics (n = 23), Quantative Risk Management (n = 15), and others (n = 45). All attributes with binary values (no/yes, male/female, etc.) were replaced with 0s and 1s respectively. Next, all attributes containing integers and floats were processed. Neighbors were considered integers, whereas random numbers, stress levels and money were considered floats. First, all strings and other incorrect datatypes were replaced with NaNs. If the question belonging to the feature stated a range in which the answer needed to be, values outside this range were also replaced with NaNs.

When analysing the responses to the question "What makes a good day for you?", two main reasons were found: social interactions and productivity. Therefore, two extra features were added to the dataset: "Social answer" and"Productive answer". When someone gave an answer with the word "friend", "family" or "social", they were given a 1 for social answer, otherwise they were given a 0. The same system was used for the productivity answers, using the words: "productive", "study", "work", "working", "getting done", "productivity", "school", "research", "papers", "assignment", "coding", "goals", "achieve", "competing", "progress", "accomplish", excluding words such as "work-out" and "working out". After constructing these new features, the responses to the good day-questions were removed from the dataset.

An extra feature was added to the dataset: the lateness of a bedtime. When trying to analyse bedtimes, a difficulty is that 01:00h in the morning is a later bedtime than 23:00h, even though numerically 23 is a greater number than 1.

Therefore, this new feature is constructed in a way where 19:00h is the earliest bedtime and 7:00h the latest. These times were transformed to numerical values. When specific features of the dataset were analysed, all instances with NaN values for any of these features were removed.

Of all responses to the dataset, 56% were male and 42% were female (the other 2% were unknown). The mean bedtime among responders was 00:30h and the median year of birth was 1996. Furthermore, most of the respondents did a course in statistics before (86%) whereas only 28% previously took a course in information retrieval.

Taking a deeper look into the data, interesting details were found. A correlation between stresslevel and bedtime might come to mind when looking for relationships in this dataset. However, when calculating Pearson's correlation coefficient (after removing outliers from the data), $R = 0.06$ with $p > 0.05$, meaning that no correlation could be found. When comparing the answers to the question "What makes a good day for you?", it was found that 35% of women gave a social answer, versus 12% of men (see Fig. 1). Furthermore, when comparing women to men in their stresslevel, no extreme differences were found (Fig. 2). Looking at the boxplot, we can see that the median of the stresslevel for men is lower, but the men's fourth quartile spans a greater range with a higher maximum than women's. In one of the questions of the questionnaire, repsondents were asked to provide a random number between 0 and 10. Figure 3 shows that these numbers were not evenly distributed, but that the graph is more similar to a left skewed distribution.

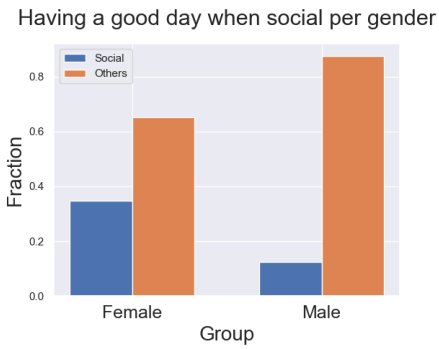

Fig. 1: Having a good day when social per gender. $N_{female} = 118$, $N_{male} = 162$.

Fig. 2: A boxplot of stress levels of students per gender. $N_{female} = 116$, $N_{male} = 155$.

B: Basic classification/regression

For this task, two classifiers were built to predict gender based on different features in the dataset. The first classifier that was used is the K-nearest neighbour algorithm, which is a supervised classifier that places all instances of a dataset in

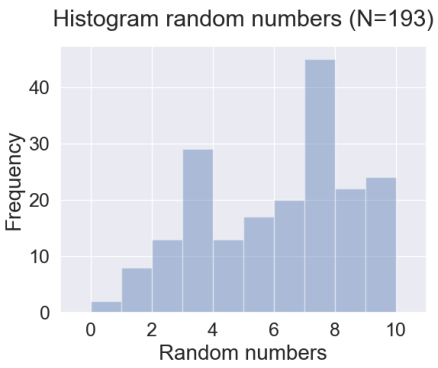

Fig. 3: A histogram of the numbers students provided when asked for a random number.

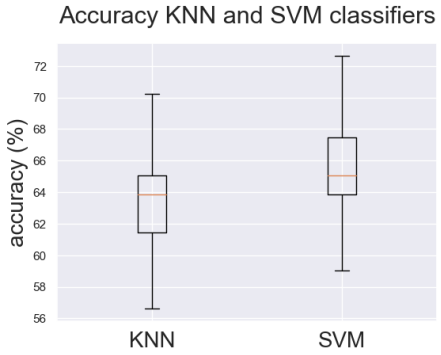

Fig. 4: The accuracy of a K-nearest neighbour algorithm ($k = 7$) and an SVM algorithm calculated 100 iterations of 10-fold cross validation.

a space. When a prediction is made, it analyses the position of the new instance in the space and assigns the class that is most frequently present in the k-nearest neighbours [1]. The second classifier is a support vector machine (SVM), which uses a kernel function to transform a dataset to a higher dimension [2]. This transformation makes it able to deal with non-linear classification problems. A radial basis function kernel was used. To use the dataset for a classification algorithm, one-hot encoding was used for the attributes considering chocolate and programme. Information gain feature selection was used to obtain five features: $chocolate_{neither}$, $chocolate_{slim}$, lateness of bedtime, good day when social and good day when productive. The number of features was decided on by exploratory data analysis, which revealed that this was the optimal number of features for both classifiers. The most optimal value for k was found to be 7. Instances with NaN values in any of the chosen features were dropped, leaving the dataset with 256 instances. The accuracy of the classifiers was determined by performing 100 iterations of 10-fold cross validation (see Fig. 4). Most of the predictions of the SVM were more accurate than those of the KNN classifier, but the differences are not big. The average accuracy of the KNN classifier was 63% and was 65% for the SVM. To calculate the confidence interval (CI), the formula $ACC_s \pm z\sqrt{\frac{1}{n}ACC_s(1-ACC_s)}$ is used with $z = 1.96$ and $n = 1000$ for a 95% CI, giving a CI of [60%, 66%] and [62%, 68%] for the KNN and SVM, respectively.

Task 2

A: Preparation

The Titanic dataset contains numerical and categorical explanatory variables and a binary response variable which represents the survival of an individual.

An overview of the variables is provided by Table 1 where count equals the amount of datapoints. The outcome variable Survived is only available for two third of the data (the training dataset) and shows a mean of 0.38. Age shows a mean of 29.7. SibSp is the amount of siblings or spouse and Parch is the amount of parents. Fare is the money which was paid for the ticket. Next to the name and sex of passengers, there is the cabin number (Cabin), the ticket class (Pclass) and location of embarkation (Embarked). Regarding the categorical data, "unique" is the amount of different cell values that are present in the column, "top" is the most common cell value per column and "freq" is the most common value's frequency. In addition, there is a variable for the passenger ID, which was not included in the summary.

Numerical	Age	Fare	Parch	SibSp	Survived
count	1046.00	1308.00	1309.00	1309.00	891.00
mean	29.88	33.30	0.39	0.50	0.38
std	14.41	51.76	0.87	1.04	0.49
min	0.17	0.00	0.00	0.00	0.00
max	80.00	512.33	9.00	8.00	1.00
Categorical	Cabin	Embarked	Name	Pclass	Sex
count	295	1307	1309	1309	1309
unique	186	3	1307	3	2
top	C27	S	K, Mr. J.	3	male
freq	6	914	2	709	843

Table 1: Description of numerical data.

Title	**Including**
Mr	Mr
Mrs	Mme
Master	Master
Miss	Ms, Mlle
Noble	Don, Sir, Jonkheer, the Countess, Lady, Dona
Other	Capt, Col, Dr, Major, Rev

Table 2: Title categories.

Transformations. From visual inspection of the data it was noted that each individual has an informative title in their name field showing their status. Titles were extracted and 17 different titles were distinguished. They were categorised into six groups that are depicted in Table 2. The title category was included in the dataset under variable Title. As variables Parch and SibSp are both indications about the passenger's family size, they were combined into a single variable FamSize. A distinction is made between "alone", 1, 2, 3 and "4 or more" companions.

Exclusions. After extraction of the titles, the Name variable was dropped. Likewise, variables SibSp and Parch were excluded. Variable Cabin was removed due to many missing values. In addition, variable Ticket and PassengerID wre dropped as they were considered irrelevant for survival.

Correlations. Survival for each group in the different categorical variables was plotted (Figures 5-7). The mean survival per group is used per group and error bars show the standard deviation from the mean.

Most notable is that the different factors affect the chances of survival. In Figure 5, it can be noted that women have a much higher survival rate than men. Figure 6 shows that sole travelers and people accompanied by 4 or more family members have the lowest survival rate. People in groups of three showed the highest survival. Survival of each title group is shown in Figure 7. A high

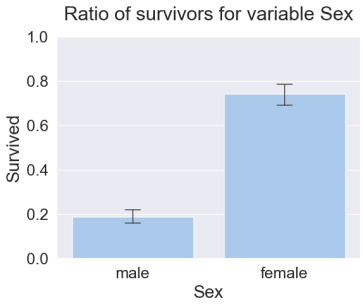

Fig. 5: Survival rate per sex.

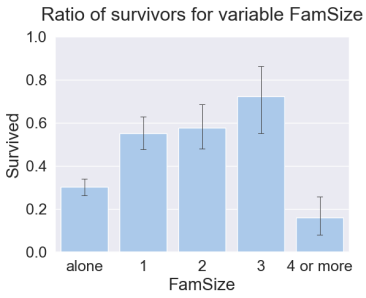

Fig. 6: Survival per family size.

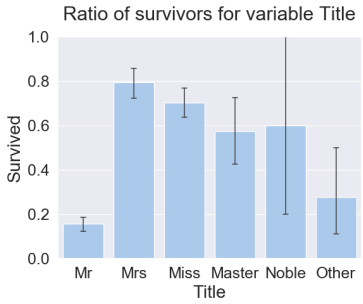

Fig. 7: Survival rate per title.

variance in survival can be noted for the different groups. Most likely to survive are women (Mrs), young women (Miss), young men (Master) and noble people.

In addition, importance of the parameters were tested using sklearn's *SelectKBest* function. Parameters of primary interest are Title, Sex, Pclass, Fare FamSize. Less conclusive, but also included, are Age and Embarked.

Data preparation. The classification algorithms require the scaling of numerical variables and the transformation of categorical variables. Numerical variables Age and Fare were scaled to float values between -1 and 1 using the scaler function *StandardScaler* from the Sklearn preprocessing package. Categorical variables Embarked, Title, FamSize and Pclass were transformed into binary variables. Therefore, each column was multiplied by the amount of groups in the variable and the group name was appended to the categorical variable's name. E.g. for the variable Embarked, the transformation results in Embarked_C, Embarked_Q and Embarked_S. The cell values were adjusted accordingly, i.e. if a passenger has cell value "S" for Embarked, then Embarked_C = 0, Embarked_Q = 0 and Embarked_S = 1.

B: Classification and evaluation

Evaluation set up After preprocessing, the data was split without replacement into training and test sets. The training set contains all explanatory variable values (x-values) of which survival is known (y-values). Both have a total of 891 rows. For training of the classifier, a fraction of the training set is used for validation (the validation set). The testing dataset contains 418 rows, which are all the explanatory variable values of which the survival has to be determined using prediction by the model.

Classifier 1: Neural Network. The neural network (NN) was created with the Keras package in python, using a *Sequential* model with *Dense* layers [3]. The input layer has the same dimension as the amount of rows in the training dataset (891). The output layer has a dimension of 1 (the survival prediction) and uses a sigmoid activation function to obtain output values between 0 and 1. The survival prediction is rounded to the nearest integer to predict whether the passenger survived (1) or not (0). Between the input and output layer, a number of hidden layers are included.

Parameter tuning. To find the optimal amount of hidden layers, a total of seven layers is tested. All additional layers had the same activation function as the input layer and a comparison was made between the linear and Rectified Linear Unit (ReLU) function [3]. An average was taken over 10 runs for each model. To avoid overfitting, a number of nodes was dropped between the last hidden layer and the output layer with a probability of 20%. The initial tests were performed with 100 epochs, a default batch size of 32, and a 80/20 cross-validation (CV) split (see Training). One hidden layer resulted in the best performance. Including a dropout after each hidden layer did not lead to a better result for multiple layers..

Based on these findings, a model with 1 hidden layer and a ReLU activation function was chosen. Next, parameters batch size, dropout probability and epochs were tuned using the sklearn *GridSearchCV* function [8]. Batch sizes of 10, 30 and 50 were investigated. Epochs of 25, 50 and 100 were chosen and dropout probabilities were varied between 0, 0.2 and 0.4. A K-fold CV of 5 was used in accordance with the 80/20 CV split. The optimal parameters were determined for one hidden layer with 4 neurons, a batch size of 10, 100 epochs and dropout of 0.2 (20%).

Training. The parameters determined during parameter tuning are used for training, which are epochs=100, batch size=10, and a dropout of 0.2 (20%). The dropout occurred between the last hidden layer and the output layer. Including a dropout between each hidden layer did not result in higher accuracy. A validation split of 0.2 is used, so 20% of the data is used for validation. The same validation set is used for all epochs in the same fit. After training the model using the fit function, a mean accuracy of 82.7% (N=100) was obtained to predict the survival for the training set. The confidence interval can be approximated through a normal distribution. Given the formula for the 95% CI (see 1A), we find a CI of [75.2, 90.1].

Testing. Applying this model to predict the survival for the testing set resulted in an accuracy of 79.4% (public leaderboard score of 0.79425). Using the same model with linear activation, an accuracy of 81.6% during training and 78.9% for testing was obtained (score of 0.78947).

Classifier 2: Random Forest. The random forest model is implemented using sklearn's *RandomForestClassifier* function [8]. As a random forest is a collection of decision trees, they share some hyperparameters such as maximum depth (i.e. maximum path length between the root and leaf node), minimum samples split (i.e. minimum requirement of splitting a node) and minimum samples leaf (i.e. the minimum samples in a terminal node). The maximum terminal nodes can be determined as to put a condition on the node splitting. By restricting the tree growth, overfitting is avoided. In addition, the number of estimators (trees) can be chosen [8].

Parameter tuning. Parameter evaluation was performed using *GridSearchCV*. Values are tested for 10, 30, 50, 70, 100 and 500 estimators; 1, 5, 10 and 15 maximum depths, .1, .2, .5, 2 and 5 minimum samples leaf. Small values are chosen for the maximum depth to avoid overfitting. Likewise, a minimum samples split

of 6 is chosen to prevent overfitting the training set. Optimal values that were found are a max depth of 10, minimum samples leaf of 2, minimum samples split of 5 and 50 estimators.

Training. Using hyperparameter of maximum samples, only a fraction of the training dataset is given to each individual. By using only fractions of bootstrapped data, the training time is decreased significantly without decreasing the accuracy. The optimal values are applied to the model and training is done using the fit function. A mean accuracy score of 89.1% (N=100) was obtained for the training set with 95% CI of [83.0, 95.2].

Testing. The model resulted in a prediction of survival for the test set which is accurate for 78.5% (public leaderboard score of 0.78468).

Evaluation classification The neural network and random forest classifier showed similar performance in predicting the survival of individuals by using the variables Title, Sex, Pclass, Fare, FamSize, Age and Embarked. The neural network performed slightly better with a 78.9% accuracy, compared to the 78.5% accuracy of the random forest model. During training, the NN performed with an average accuracy of $82.6 \pm 1.4\%$ and the RF averagely showed $89.1 \pm 0.4\%$ over 100 runs. The 95% CIs are [75.2, 90.1] and [83.0, 95.2], for NN and RF respectively. There is overlap between the intervals. The normality of the data was confirmed with the Shapiro Wilk test ($W = 0.97$; $p = 0.052 > 0.05$ for both NN and RF). The performance was compared with the t-test, which showed a significant difference between groups ($t = -44.4$; $p < 0.001$).

When training both models, the NN was significantly less accurate on the training data than the RF. Therefore, it was expected that the RF would perform better on the test set. However, high accuracy of the RF on the training set may imply an overfitted model, which can explain the reduction in performance.

Task 3

A: Research - State of the art solutions

In the summer of 2019 a Kaggle competition was held about detecting blindness which is caused from diabetic retinopathy in rural parts of India.

For this competition a dataset of retina images is provided. Every image is rated by a clinician with a rate of severity of the disease ranging form 0 to 4, with 0 being healthy and 4 having proliferative diabetic retinopathy. For the evaluation every image is given a rating made by a human and the predicted rating. The agreement between these ratings is calculated based on the quadratic weighted kappa. An interesting aspect to this competition is that in 2015 a similar competition was held. A dataset in the same format and the same evaluation measure is used, which means that more data was available and using that data is exactly what the winners did.

The submission that won first place is made by Guanshuo Xu. He combined the whole dataset of 2015, so both train and test set, and the train set of 2019

and used this as training set. This means that solely the test set of 2019 is used for validation. Besides resizing the images Xu did not preprocess the data. First he made multiple models based on neural networks. To acquire more accurate results he trained the best models in pairs for more stable results, for this he used different seeds. After reducing the degree of freedom on hyper parameters, to avoid overfitting, a first few optimal solutions were found. However, he did not stop here: he added more data to the train set. This new data consisted of the public test set and of two external sources of data: the Idrid and the Messidor dataset. For both external datasets the labels were smoothed: for Idrid to avoid bias and for Messidor to acquire a fifth label since the Messidor dataset only contains four labels. His final submission was the average of the predictions of 8 of his best models.

Even though many models were used, due to a time shortage Xu did not implement an EfficientNet model. This is something his runner-ups did implement. Other differences are in the preprocessing and in the training sets. Both the second place submission, by the team 'Eye of Private LB', and the fourth place submission, by Qishen Ha, cropped the background out of the images. Ha additionally to the cropping, converted every image to the same type of brightness. This was only done for the training set. Similar to Xu, both runner ups averaged the predictions of their best models for final prediction. Both runner ups also used the 2015 dataset but they did this in a different way. Ha pretrained on the full dataset of 2015 and later trained with a 5 fold cross validation on the data of 2019. Meaning Ha used less training data than Xu did. The team also pretrained on the 2015 data and later continued training on the 2019 data. To acquire even more data they pseudo-labeled the 2019 test data. They used this pseudo-labeled data in combination with their already existing training set to fine tune their model. A fun side note is that one of the solutions the team came up with was actually better than the winning submission.

B: Theory - MSE vs MAE

Both the mean squared error (MSE) and the mean absolute error (MAE) are used for error measurement between observed and predicted values. Both formulae are shown below, where X represents the observed value and Y the predicted value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \quad and \quad MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - Y_i|$$

Since in the MSE the difference between the observed and predicted value, the error, is squared, the MSE is very sensitive to outliers. So if we have skewed data, we would prefer using the MAE over the MSE. When however, specifically the large errors are undesirable the MSE is more useful.

When we let $r_i = |X_i - Y_i|$ and $r = [r_1, \ldots, r_n]^T$ we can rewrite the formulas for the MSE and MAE:

$$MSE = \frac{1}{n}\vec{1}^T r \quad and \quad MAE = \frac{1}{n}r^T r,$$

where $\vec{1}$ is a $n \times 1$ vector of ones. Theoretically it could happen that the MSE and MAE are the same, then:

$$(\vec{1}^T - r^T)r = 0. \tag{1}$$

This holds for the vectors $r = \vec{0}$ and $r = \pm\vec{1}$, the zero and one vector. Since the zero dot product implies orthogonality it holds that MSE and MAE are equal when the vector consisting of $r_i - 1$ is orthogonal to r, the original absolute values. Moreover, we can rewrite this formula by completing the square into:

$$\left(r - \frac{\vec{1}}{2}\right)^T \left(r - \frac{\vec{1}}{2}\right) = \frac{n}{4}, \tag{2}$$

where $\frac{\vec{1}}{2}$ is a $n \times 1$ vector of halves. From here it follows that there is a n dimensional hypersphere at $\frac{\vec{1}}{2}^T$ with a radius of $\frac{1}{2}\sqrt{n}$. When the absolute differences between the observed and predicted values are on the surface of this hypersphere the values of MSE and MAE are equal.

When for example all data from a data set comes from the formula $y = ax+b$, the data is perfectly fitted by said formula. In such a case where data is perfectly fitted both the MSE and MAE are equal to zero and thus equal to each other.

To investigate the difference between MSE and MAE two regression methods are run on a data set which contains gender, weight in lb and height in inches of 10000 people. The data set can be found on Kaggle: `https://www.kaggle.com/mustafaali96/weight-height`. This data set is chosen because of the known correlation between weight and height of humans. In Figure 8 the weight and the height are plotted against each other, there the correlation between the two is visible.

The two methods that are used are linear regression and tree regression. The tree regression is performed twice: once with the MSE as criterion and once with the MAE as criterion. First the linear regression is performed on the data, the red line in Figure 8. This resulted in a MSE of 122.7 and a MAE of 10.5, results are shown in Table 3. The difference between these two can be explained by the broad range of values around the fitted line. Which means that the bigger values weigh more heavily to the MSE than to the MAE resulting in a higher MSE than MAE. Due to the tree based structure a tree regression gives us a discrete line when plotted, the yellow and green lines in the figure. This means that per 'group' the data is fitted optimally, where a group is a chunk of data which has values close to each other. In Table 3 it can be seen that the errors of the tree regression are much lower: approximately 42.6 for the MSE and 5.7 for the MAE. Interesting to note is that when the MAE criterion is used the errors of both MSE and MAE are a bit lower than when the MSE criterion is used. It is assumed that this difference derives from the fact that the resulting MAE is lower than the MSE, hence a criterion based on the MAE gives a more accurate result. The discrete nature of the tree regression explains why the MSE and MAE of the tree regressions are lower than those of the linear regression. Additionally,

in the tree regression the data gets split up based on the chosen criterion, which means that in every layer the criterion is minimised. In comparison to linear regression where the least square error is minimised.

Fig. 8: The three regression methods plotted together with the data of height and data.

Model	MSE	MAE
Linear regression	122.7	10.535
Tree regression MSE	42.772	5.7295
Tree regression MAE	42.533	5.7161

Table 3: The three regression methods with their corresponding values of MSE and MAE.

C: Theory - Analyse a less obvious dataset

To build a model that can distinguish spam texts from bonafide texts, a dataset of 5574 text messages from the UK was analysed. Different classification models can be used: KNNs, SVMs, Neural networks and Naïve Bayesian classification are popular supervised spam filtering methods [4]. Data transformations can consist of creating new features such as the number of capital and lowercase letters, the number of punctuation marks and the length of the text. Unimportant words and characters could be removed from the texts. Next, word lemmatisation or word stemming techniques could be used, which can help bring down the complexity of a sentence and make it possible to compare sentences that use different words that have the same meaning. Word stemming strips down words to their stem (i.e.: walking to walk) [5], whereas word lemmatisation analyses words and replaces them with their corresponding lemmas (i.e. could to can) [6]. To create more features based on the content of the text, word counters or word embedding could be used. Word embedding is a technique that transforms words to vectors, which can then be compared by calculating the distance in the word space [7]. Words with similar meanings will be closer to each other than words with differing meanings. For the SMS dataset, the number of punctuation marks and lower- and uppercase letters were collected from the dataset and added as new features. Furthermore, the length of each text was added as well, making a total of 48 features of which all were used. Next, a SVM with a radial basis function kernel was used to build a model. 10 iterations of 5-fold cross validation were performed, with an average accuracy of 96% and 95% confidence interval of [90%, 100%]. The average precision was 0.95 and the average recall was 0.89. It might be possible that a higher level of accuracy could by obtained by using the previously mentioned techniques.

References

1. B. Surya Prasath, B.S., Alfeilat H.A.A., Lasassmeh, O., A.B.A.: Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier - A Review. CoRR (2017)
2. Chorowski J., Wang J., Zurada., J.M.: Review and performance comparison of SVM- and ELM-based classifiers.Neurocomputing /textbf128, 507-516 (2014)
3. Chollet, F. (2015) keras, GitHub. https://github.com/fchollet/keras
4. Dada, E.G., Bassi, J.S., Chiroma, H., Abdulhamid, S.M., Adetunmbi, A.O., Ajibuwa O.E: "Machine learning for email spam filtering: review, approaches and open research problems". Heliyon **5**(6), 2405-8440 (2019)
5. Jivani, A.: A Comparative Study of Stemming Algorithms. Int. J. Comp. Tech. Appl.. 2. 1930-1938 (2011)
6. Korenius, T., Laurikkala, J., Järvelin, K., Juhola, M.: Stemming and lemmatization in the clustering of Finnish text documents. Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM'04. 625-633.(2004). 10.1145/1031171.1031285.
7. Hamed Z., Croft, W.B.: Relevance-based Word Embedding. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17). Association for Computing Machinery, New York, NY, USA, 505–514 (2017) DOI:https://doi.org/10.1145/3077136.3080831
8. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.