Optimal Data Analysis Methods for Email Spam Detector Using KNNs and Bayes Classifiers

Aleksandra Schoepe[1], Magdalena Góras[1] and Szymon Kiełkowski[1]

[1]*Faculty of Applied Mathemathics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland*

Abstract

The article contains a detailed data analysis based on the Email Spam Classification database. The results of using various methods of column elimination as well as classification and regression algorithms will be presented.

Keywords

LaTeX class, Data Analysis, Spam, Algorithms, Python

1. Introduction

Almost every area of life has been computerized. At every step we are dealing with more and more complex IT systems, which thanks to equipping with artificial intelligence systems [1, 2, 3] are becoming more and more useful and useful. The current situation in the world forces people to reduce energy consumption. A very interesting application of heuristic algorithms to reduce energy losses during its transmission is presented by [4, 5]. Heuristic algorithms are often based on the observation of the biological world [6, 7] so that they can be used in many problems of everyday life that lead to optimization tasks in which the objective function is so complex that it cannot be adapted to classical methods. Many real world problems can also be solved by using neural networks and fuzzy logic [8, 9, 10, 11]. These applications concern solutions related to smart home management [12, 13] as well as in expert systems used to determine the quality of roads [14]. Artificial intelligence is also built on the basis of artificial neural networks [15]. This algorithm is a very effective tool in identifying certain features [16, 17, 18, 19, 20], they are also used in the machine learner [21, 22]. Neural sieve applications also apply to the protection of health [23].

Did you ever wonder how often people check their mailboxes? According to the USPS, 98% of people visit their mailbox every day to get their mail and 77% of people sort through their mail immediately after they wake up. Nowadays, the mailbox is a big part of our daily lifestyle. There are a lot of programs for mailbox from where we can choose the one that suits us best, for example Apple Mail, Mail App for Gmail, Outlook, Spark, and BlueMail. There is one major function that all of those software includes – all of them can detect the infuriating messages from shops, organizations, or websites where you had logged in only once to get the coupon for 20% off the electric toothbrush. We all perfectly know how annoying are those emails called spam. But have you ever taken into consideration how the email spam detector in your daily mailbox works? We were interested in investigating how does the tool we use and love everyday works this much that we took our research to prepare and to comapre the best data analysis methods for email spam detector using KNN and Naïve Bayes Classifier.

2. How does the program work?

The main task in our project is to detect spam among the emails. At first program in purpose to divide spam and non-spam messages are based on the analysis of the frequency of occurrence of given word in their content. For the purposes of the project, the program first reduces the size of the database. Then it eliminates the columns, leaving the significant ones, with a few different algorithms. Finally with the Naive Bayes algorithm and KNN we are detecting witch of the emails are called a spam.

3. DataBase - Email Spam Classification

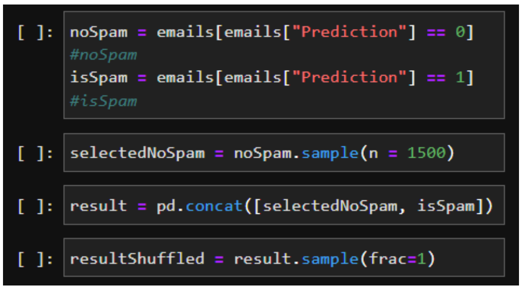
The database on which we conduct our tests is the email spam classification database. It contains 3002 columns and 5172 lines, where each line represents a separate mail. The columns show the most common words in all e-mails. The last column contains the predictive labels: 1 for spam and 0 for no spam. The name was set with numbers, not the names of the recipients, to protect privacy. Information is stored in a compact data frame. The base was taken from the website available in this link [24] where you can also learn more about it.

ICYRIME 2022: International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Catania, August 26-29, 2022

aleksch@polsl.pl (A. Schoepe); magdgor@polsl.pl (M. Góras); szymkie@polsl.pl (S. Kiełkowski)

4. Preparation of the base

Initially, the complete database is divided into spam and non-spam. Then selected are randomly 1500 records from two those separate groups. The data is then combined successively and mixed, and then the newly created database will be subject to further analysis.



```
[ ]: noSpam = emails[emails["Prediction"] == 0]
     #noSpam
     isSpam = emails[emails["Prediction"] == 1]
     #isSpam

[ ]: selectedNoSpam = noSpam.sample(n = 1500)

[ ]: result = pd.concat([selectedNoSpam, isSpam])

[ ]: resultShuffled = result.sample(frac=1)
```

Figure 1: Piece of code with database preparation.

5. Selecting useful features

There are many ways to select only useful features in database, but those methods for feature selection can be divided into three main groups:

- Filter based: we define some data and metric and based on these filter functions. An example of such a metric may be a chi-square.
- Wrapper-based: Those methods consider the selection of a set of features as a search problem. An example of such methods may be a Recursive Feature Elimination.
- Embedded : Embedded methods use algorithms which have built-in feature selection methods. For instance, Lasso.

6. Data Preprocessing Methods

Due to the size of the base and the number of columns, there was a need to eliminate them. Therefore, 6 methods will be used for this purpose. However, first it is worth presenting the types of these methods.

6.1. Pearson Correlation

This is one of the filtering methods. First, we check the absolute value of the Pearson correlation between the target and the numerical features in the dataset being checked. Based on this criterion, we keep the n best functions. The Pearson correlation coefficient r is used to test whether two quantitative variables are related to each other by a linear relationship. This factor ranges from -1 to 1.

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2\sum(y-\bar{y})^2}} \tag{1}$$

6.2. Chi-Squared

This is another filtering method. The chi-square test is used to determine if there is a significant difference between the expected and observed frequencies in one or more categories. Then it considers if there is a sampling error.

In this method, we calculate the chi-square metric between the target variable and the numeric variable. The algorithm then selects only the variables with the maximum chi-square values. To correctly compute chi-square, we first find the values that we would expect in each part if there were indeed independence between the two categorical variables.

Finally, follow the formula below; we multiply the sum of the rows and the sum of the columns for each cell and divide it by the sum of the observations.

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2}$$

6.3. Recursive Feature Elimination

This is a wrapper method. Wrapping methods treat the selection of a feature set as a search problem.

When using RFE, there are two important configuration options: choosing the number of functions to select and selecting the algorithm used to select the functions. Both of these hyperparameters can be tested, although the method's performance is not strongly dependent on the correct configuration of these hyperparameters.

From the sklearn documentation:

"The goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on an initial feature set and the importance of each feature is obtained either through the coef attribute or through the feature-importances attribute. Then, the least important features are removed from the current feature set. This procedure is repeated recursively on the truncated set until the desired number of features to be selected is finally reached."

6.4. Lasso

This is an embedded method. As mentioned earlier, embedded methods use algorithms that have built-in function selection methods. For example, Lasso and RF have function selection methods. Lasso sets parameters to zero. The higher the alpha value, the fewer features have non-zero values.

$$\alpha \sum_{i=1}^{k} |w_i| \tag{3}$$

6.5. Tree-based

It is also an embedded method. Team machine learning for classification, regression, and other tasks, which involves building multiple decision trees while training, and generating a class that dominates the classes or the predicted mean of each tree. We can also use RandomForest to select functions based on their importance.

We compute the importance of the features using the nodal impurities in each decision tree. In a random forest, the final importance of a feature is the average importance of all the features of the decision tree.

6.6. Principal Component Analysis (PCA)

PCA is a linear dimension reduction technique that can be used to extract information from a high dimension space by projecting it onto a lower dimension subspace. The algorithm tries to keep the most important parts while removing the less important ones.

One important thing to note with PCA is that it is an unsupervised dimension reduction technique where you can group similar data points based on the correlation of features between them without any supervision. PCA is a statistical procedure that uses an orthogonal transformation to transform a set of observations of potentially correlated variables into a set of values for linearly uncorrelated variables, called principal components. The way it works is that if the correlation among a subset of features is definitely high, the algorithm tries to combine similar features.

Thanks to these components, it is possible to recreate the original characteristics sufficiently accurately. The PCA algorithm actively tries to minimize the reconstruction error when searching for optimal components.

6.7. Comparison of column elimination algorithms

Comparison of column elimination algorithms for different number of columns using the Bayes algorithm.

When analyzing the table above, it can be seen that good results of 80% can be obtained in some algorithms with as little as 10 columns.

The 20 columns were adopted for further consideration to obtain better results and maintain efficiency.

Sample correlation matrix for Pearson algorithm and 20 columns.

7. Algorithms

7.1. Naive Bayes

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4}$$

Naive Bayesian classifiers are based on the assumption that predictors (independent variables) are independent of each other. They often have nothing to do with reality because of their contractual assumptions and are therefore called naive.

These algorithms are most often used in sentiment analysis, spam filtering, and recommendation systems. They are quick and easy to implement, but unfortunately have the downside of requiring the independence of the predictors.

7.2. K Nearest Neighbors (KNN)

The k-nearest neighbor algorithm, also known as KNN, is a non-parametric supervised learning classifier that uses proximity to create a classification or group prediction for a single data point. It is used for regression and classification, but more often for the second application, which is based on the assumption of proximity of points.

The K Nearest Neighbors method belongs to the group of lazy algorithms. This is because it does not create an internal representation of the training data, but only looks for a solution when a test pattern appears. It stores all the training patterns on the basis of which it determines the distance of the test pattern.

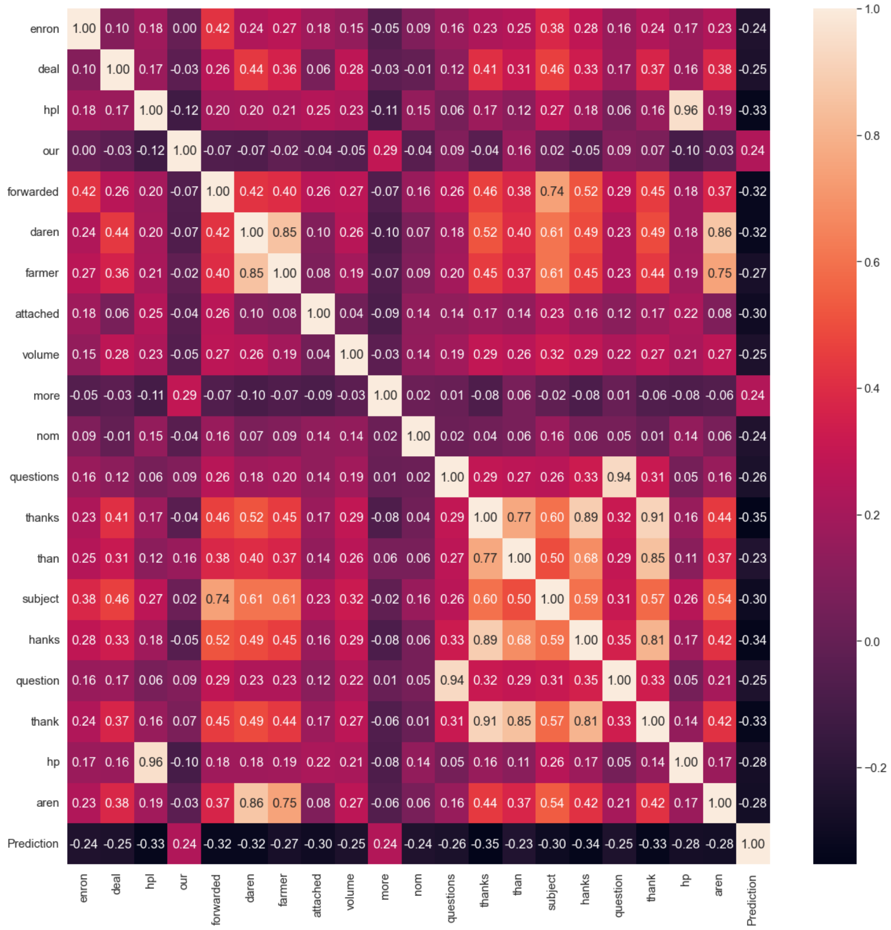
Learning an algorithm can be represented in three ways. The first is instance-based, which uses whole training instances to predict the result. Another is Lazy Learning, which is distinguished by the fact that the model training process is postponed until the program requests a forecast in a new instance. The last one is non-parametric, meaning there is no predefined form of the mapping function.

8. Tests

The data set was divided into a training and validation set on a scale of 70:30.

Table 1
Naive Bayes accuracy with different features selection algorithms and number of chosen features.

Features	Pearson	Chi-Square	RFE	Lasso	Tree-based	Combined
2	0.66	0.58	0.67	0.71	0.69	0.00
5	0.84	0.78	0.78	0.73	0.77	0.76
10	0.78	0.85	0.83	0.79	0.78	0.74
20	0.84	0.86	0.84	0.78	0.84	0.81
50	0.85	0.87	0.88	0.83	0.87	0.84
100	0.90	0.85	0.90	0.89	0.88	0.87



	enron	deal	hpl	our	forwarded	daren	farmer	attached	volume	more	nom	questions	thanks	than	subject	hanks	question	thank	hp	aren	Prediction
enron	1.00	0.10	0.18	0.00	0.42	0.24	0.27	0.18	0.15	-0.05	0.09	0.16	0.23	0.25	0.38	0.28	0.16	0.24	0.17	0.23	-0.24
deal	0.10	1.00	0.17	-0.03	0.26	0.44	0.36	0.06	0.28	-0.03	-0.01	0.12	0.41	0.31	0.46	0.33	0.17	0.37	0.16	0.38	-0.25
hpl	0.18	0.17	1.00	-0.12	0.20	0.20	0.21	0.25	0.23	-0.11	0.15	0.06	0.17	0.12	0.27	0.18	0.06	0.16	0.96	0.19	-0.33
our	0.00	-0.03	-0.12	1.00	-0.07	-0.07	-0.02	-0.04	-0.05	0.29	-0.04	0.09	-0.04	0.16	0.02	-0.05	0.09	0.07	-0.10	-0.03	0.24
forwarded	0.42	0.26	0.20	-0.07	1.00	0.42	0.40	0.26	0.27	-0.07	0.16	0.26	0.46	0.38	0.74	0.52	0.29	0.45	0.18	0.37	-0.32
daren	0.24	0.44	0.20	-0.07	0.42	1.00	0.85	0.10	0.26	-0.10	0.07	0.18	0.52	0.40	0.61	0.49	0.23	0.49	0.18	0.86	-0.32
farmer	0.27	0.36	0.21	-0.02	0.40	0.85	1.00	0.08	0.19	-0.07	0.09	0.20	0.45	0.37	0.61	0.45	0.23	0.44	0.19	0.75	-0.27
attached	0.18	0.06	0.25	-0.04	0.26	0.10	0.08	1.00	0.04	-0.09	0.14	0.14	0.17	0.14	0.23	0.16	0.12	0.17	0.22	0.08	-0.30
volume	0.15	0.28	0.23	-0.05	0.27	0.26	0.19	0.04	1.00	-0.03	0.14	0.19	0.29	0.26	0.32	0.29	0.22	0.27	0.21	0.27	-0.25
more	-0.05	-0.03	-0.11	0.29	-0.07	-0.10	-0.07	-0.09	-0.03	1.00	0.02	0.01	-0.08	0.06	-0.02	-0.08	0.01	-0.06	-0.08	-0.06	0.24
nom	0.09	-0.01	0.15	-0.04	0.16	0.07	0.09	0.14	0.14	0.02	1.00	0.02	0.04	0.06	0.16	0.06	0.05	0.01	0.14	0.06	-0.24
questions	0.16	0.12	0.06	0.09	0.26	0.18	0.20	0.14	0.19	0.01	0.02	1.00	0.29	0.27	0.26	0.33	0.94	0.31	0.05	0.16	-0.26
thanks	0.23	0.41	0.17	-0.04	0.46	0.52	0.45	0.17	0.29	-0.08	0.04	0.29	1.00	0.77	0.60	0.89	0.32	0.91	0.16	0.44	-0.35
than	0.25	0.31	0.12	0.16	0.38	0.40	0.37	0.14	0.26	0.06	0.06	0.27	0.77	1.00	0.50	0.68	0.29	0.85	0.11	0.37	-0.23
subject	0.38	0.46	0.27	0.02	0.74	0.61	0.61	0.23	0.32	-0.02	0.16	0.26	0.60	0.50	1.00	0.59	0.31	0.57	0.26	0.54	-0.30
hanks	0.28	0.33	0.18	-0.05	0.52	0.49	0.45	0.16	0.29	-0.08	0.06	0.33	0.89	0.68	0.59	1.00	0.35	0.81	0.17	0.42	-0.34
question	0.16	0.17	0.06	0.09	0.29	0.23	0.23	0.12	0.22	0.01	0.05	0.94	0.32	0.29	0.31	0.35	1.00	0.33	0.05	0.21	-0.25
thank	0.24	0.37	0.16	0.07	0.45	0.49	0.44	0.17	0.27	-0.06	0.01	0.31	0.91	0.85	0.57	0.81	0.33	1.00	0.14	0.42	-0.33
hp	0.17	0.16	0.96	-0.10	0.18	0.18	0.19	0.22	0.21	-0.08	0.14	0.05	0.16	0.11	0.26	0.17	0.05	0.14	1.00	0.17	-0.28
aren	0.23	0.38	0.19	-0.03	0.37	0.86	0.75	0.08	0.27	-0.06	0.06	0.16	0.44	0.37	0.54	0.42	0.21	0.42	0.17	1.00	-0.28
Prediction	-0.24	-0.25	-0.33	0.24	-0.32	-0.32	-0.27	-0.30	-0.25	0.24	-0.24	-0.26	-0.35	-0.23	-0.30	-0.34	-0.25	-0.33	-0.28	-0.28	1.00

Figure 2: Correlation matrix with features selected by Pearson Correlation algorithm.

8.1. Unnormalized data

Test results for all column elimination algorithms and Bayes' algorithm for non-normalized data.
The best result was obtained using the Ch-Square algorithm and the worst for the Lasso algorithm. It is worth adding that all the results do not differ too much from each other.

Additionally, tests were carried out for the combined above five algorithms. Below you can see the statistics and results of word matches during the test. As it turns out, we managed to get the correctness at the level of exactly 80%, which is less than in each algorithm separately.

Table 2
Statistics of the results of the combined 5 elimination algorithms.

	Feature	Pearson	Chi-Square	RFE	Logistics	Random-Forest	Total
1	thanks	True	True	True	True	True	5
2	questions	True	True	True	True	True	5
3	hpl	True	True	True	True	True	5
4	hp	True	True	True	True	True	5
5	deal	True	True	True	True	True	5
6	daren	True	True	True	True	True	5
7	attached	True	True	True	True	True	5
8	thank	True	True	False	True	True	4
9	nom	True	False	True	True	True	4
10	hanks	True	True	False	True	True	4
11	forwarded	True	True	True	False	True	4
12	question	True	True	False	True	False	3
13	our	True	False	True	False	True	3

8.2. Normalized data

The results of the tests performed for all column elimination algorithms along with the combined version for the Bayes algorithm for normalized data.
As can be seen, normalization does not have a significant impact on the results as the data is very comparable. However, in this case, the highest accuracy was calculated for the Pearson algorithm and the worst for Recursive Feature Elimination and Lasso.

9. Experiments

9.1. K Nearest Neighbors

As part of the experiment, the KNN algorithm was tested for a different number of columns, as shown in the first table, and for different elimination algorithms, as shown in the second table, for k equal to 3, 5, 7, 9, respectively.

The first table shows that the best results are obtained for 20 columns, and with more and fewer of them, the effectiveness decrease. On the other hand, from the second table, we learn that it is best to use the Pearson algorithm. Based on these conclusions, further tests were performed for different k for non-normalized and normalized data.

Table 3
KNN ACCURACY - 20 FEATURES, ALL SELECTION ALGORITHMS.

Algorithm	k=3	k=5	k=7	k=9
Pearson	91	91	90	90
Chi-Square	89	90	90	89
Recursive	86	85	83	84
Lasso	88	87	87	87
Tree	87	87	86	87

Table 4
KNN ACCURACY - FEATURES FROM PEARSON CORRELATION.

Features	k=3	k=5	k=7	k=9
5	75	81	81	81
10	84	84	84	87
20	91	91	90	90
50	88	88	87	86
100	80	80	81	80

9.2. Principal Component Analysis

The PCA algorithm for two and three components was also tested. As can be seen, the results for three are definitely better than for two.
It is also worth comparing this algorithm with those tested earlier. It can be concluded that the best result of all is obtained by using the PCA algorithm for the three components.

Table 5
PCA with 2 components created these columns.

	component 1	component 2	Prediction
0	-7.95	-3.35	1
1	-7.72	-2.87	1
2	-6.79	-2.93	1
...	...	...	...
2997	11.75	4.46	0
2998	14.17	7.78	0
2999	-6.34	-2.35	1

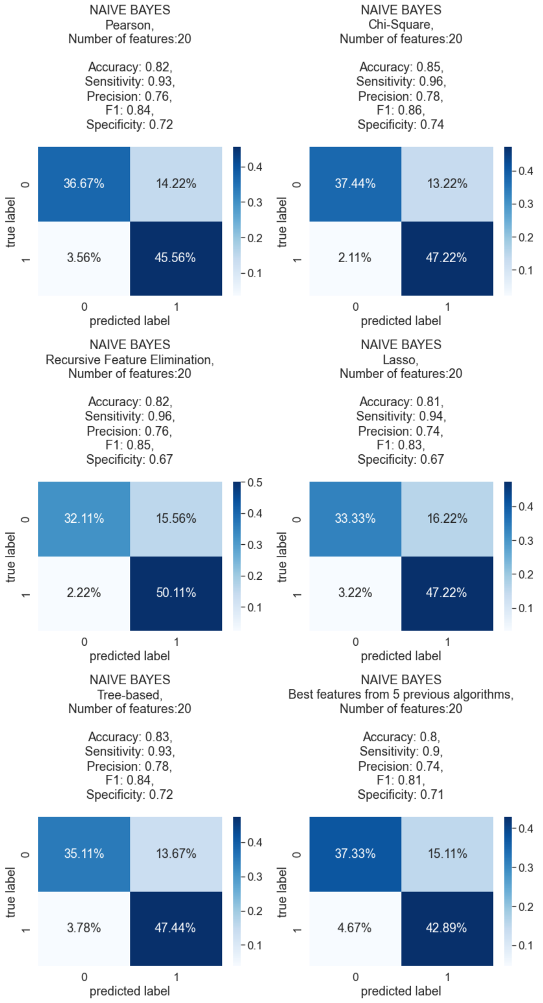


Figure 3: Naive Bayes results with 20 features and different features selection algorithms. Data unnormalized.

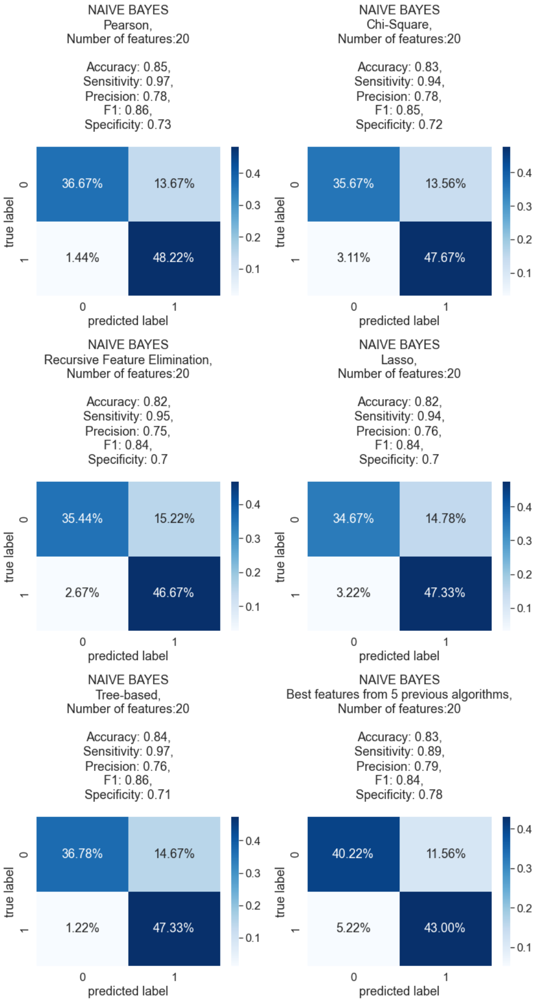


Figure 4: Naive Bayes results with 20 features and different features selection algorithms. Data normalized.

Table 6
PCA with 3 components created these columns.

	comp. 1	comp. 2	comp. 3	Prediction
0	-7.95	-3.35	-1.53	1
1	-7.72	-2.87	-0.99	1
2	-6.79	-2.93	-0.52	1
3	27.24	18.96	-0.25	0
4	-3.03	0.47	3.45	0
...	...	...	...	...
2995	24.81	10.35	-2.04	1
2996	-6.66	-2.25	-0.99	1
2997	11.75	4.47	5.85	0
2998	14.17	7.79	4.49	0
2999	-6.36	-2.35	-0.29	1

Table 7
PCA accuracy results with different number of components.

Numbers of features	Accuracy
2	58
3	90

References

[1] M. A. Sanchez, O. Castillo, J. R. Castro, Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems, Expert Systems with Applications 42 (2015) 5904–5914.

[2] Q.-b. Zhang, P. Wang, Z.-h. Chen, An improved

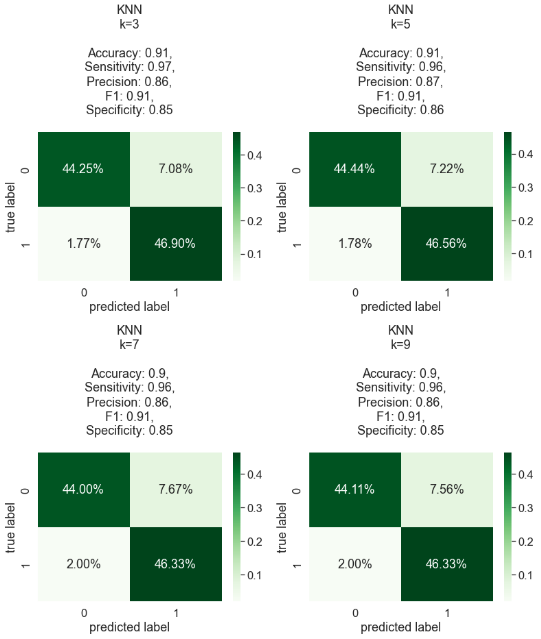


Figure 5: KNN results with 20 features and different k values. Data unnormalized.

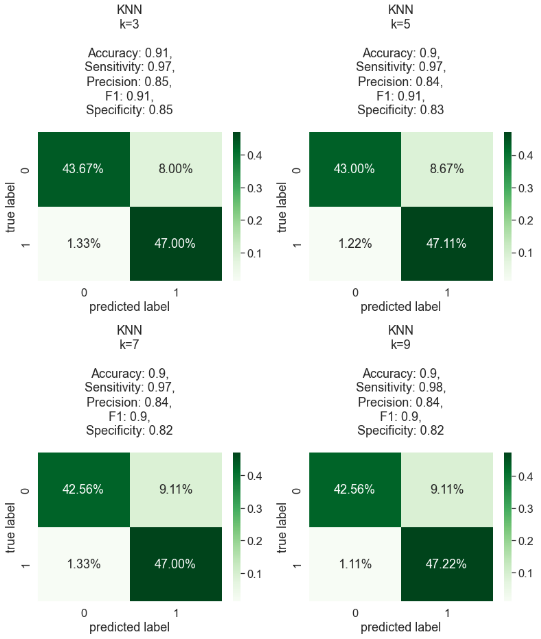


Figure 6: KNN results with 20 features and different k values. Data normalized.

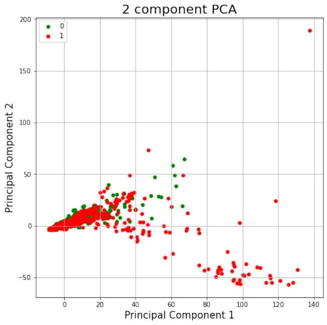


Figure 7: PCA with 2 components and their values with both prediction results.

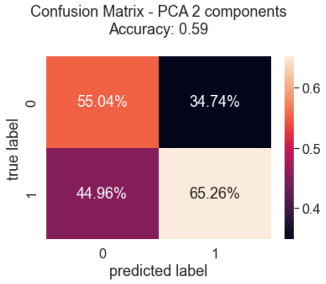


Figure 8: PCA with 2 components prediction results.

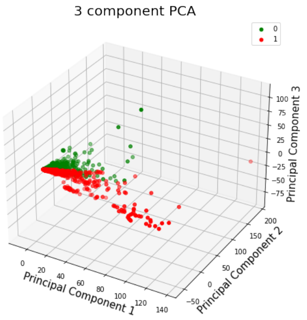


Figure 9: PCA with 3 components and their values with both prediction results.

particle filter for mobile robot localization based on particle swarm optimization, Expert Systems with Applications 135 (2019) 181–193.

[3] R. Aureli, N. Brandizzi, G. Magistris, R. Brociek, A customized approach to anomalies detection by using autoencoders, in: CEUR Workshop Proceedings, volume 3092, 2021, pp. 53–59.

[4] M. Woźniak, A. Sikora, A. Zielonka, K. Kaur, M. S. Hossain, M. Shorfuzzaman, Heuristic optimization of multipulse rectifier for reduced energy consumption, IEEE Transactions on Industrial Informatics 18 (2021) 5515–5526.

[5] V. Ponzi, S. Russo, A. Wajda, R. Brociek, C. Napoli,

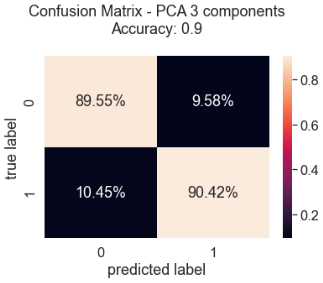


Figure 10: PCA with 3 components prediction results.

Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models, in: CEUR Workshop Proceedings, volume 3360, 2022, pp. 55–63.

[6] Y. Zhang, S. Cheng, Y. Shi, D.-w. Gong, X. Zhao, Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm, Expert Systems with Applications 137 (2019) 46–58.

[7] M. Ren, Y. Song, W. Chu, An improved locally weighted pls based on particle swarm optimization for industrial soft sensor modeling, Sensors 19 (2019) 4099.

[8] G. Lo Sciuto, G. Susi, G. Cammarata, G. Capizzi, A spiking neural network-based model for anaerobic digestion process, in: 2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), IEEE, 2016, pp. 996–1003.

[9] C. Napoli, F. Bonanno, G. Capizzi, Exploiting solar wind time series correlation with magnetospheric response by using an hybrid neuro-wavelet approach, Proceedings of the International astronomical union 6 (2010) 156–158.

[10] Y. Li, W. Dong, Q. Yang, S. Jiang, X. Ni, J. Liu, Automatic impedance matching method with adaptive network based fuzzy inference system for wpt, IEEE Transactions on Industrial Informatics 16 (2019) 1076–1085.

[11] F. Qu, J. Liu, H. Zhu, D. Zang, Wind turbine condition monitoring based on assembled multidimensional membership functions using fuzzy inference system, IEEE Transactions on Industrial Informatics 16 (2019) 4028–4037.

[12] M. Woźniak, A. Zielonka, A. Sikora, M. J. Piran, A. Alamri, 6g-enabled iot home environment control using fuzzy rules, IEEE Internet of Things Journal 8 (2020) 5442–5452.

[13] S. Russo, S. Illari, R. Avanzato, C. Napoli, Reducing the psychological burden of isolated oncological patients by means of decision trees, in: CEUR Workshop Proceedings, volume 2768, 2020, pp. 46–53.

[14] M. Woźniak, A. Zielonka, A. Sikora, Driving support by type-2 fuzzy logic control model, Expert Systems with Applications 207 (2022) 117798.

[15] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, M. F. Ijaz, M. Woźniak, A survey of deep convolutional neural networks applied for prediction of plant leaf diseases, Sensors 21 (2021) 4749.

[16] O. Dehzangi, M. Taherisadr, R. ChangalVala, Imu-based gait recognition using convolutional neural networks and multi-sensor fusion, Sensors 17 (2017) 2735.

[17] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[18] H. G. Hong, M. B. Lee, K. R. Park, Convolutional neural network-based finger-vein recognition using nir image sensors, Sensors 17 (2017) 1297.

[19] G. Capizzi, G. Lo Sciuto, M. Woźniak, R. Damaševicius, A clustering based system for automated oil spill detection by satellite remote sensing, in: Artificial Intelligence and Soft Computing: 15th International Conference, ICAISC 2016, Zakopane, Poland, June 12-16, 2016, Proceedings, Part II 15, Springer, 2016, pp. 613–623.

[20] N. Brandizzi, S. Russo, R. Brociek, A. Wajda, First studies to apply the theory of mind theory to green and smart mobility by using gaussian area clustering, in: CEUR Workshop Proceedings, volume 3118, 2021, pp. 71–76.

[21] A. T. Özdemir, B. Barshan, Detecting falls with wearable sensors using machine learning techniques, Sensors 14 (2014) 10691–10708.

[22] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, D. Bochtis, Machine learning in agriculture: A review, Sensors 18 (2018) 2674.

[23] M. Woźniak, M. Wieczorek, J. Siłka, D. Połap, Body pose prediction based on motion sensor data and recurrent neural network, IEEE Transactions on Industrial Informatics 17 (2020) 2101–2111.

[24] B. Biswas, Email spam classification dataset csv, 2020. URL: https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv.