Suggesting a Specific Factor-driven Career Choice using KNN and Soft Set Algorithms

Joanna Bodora[1], Jadwiga Cader[1] and Nikola Gębka[1]

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

Abstract

Choosing perfect work path is not an easy task especially in IT sector. Lately we can notice that data science and jobs connected to this field are getting more and more popular. To reduce time consumed on finding perfect work position in data science, authors have presented solution, which selects best job based on factors introduced by user. Final job title is a result of combining soft set algorithm with analyzed accuracies of k-nearest neighbours algorithms classified with different k parameters and on various collections.

Keywords

Soft set, k-nearest neighbours, Classification

1. Introduction

Nowadays, IT systems [1, 2] very often use artificial intelligence methods, which allow not only to download and process data [3], but also to infer and support the decision-making process based on them. One of the important branches of artificial intelligence systems are fuzzy sets [4, 5, 6], which are used in numerous applications, among others, in the detection of pavement damage [7] or in smart home management [8, 9]. The second important direction of applications are the optimization algorithms [10, 11, 12, 13], which are used in optimization processes, where the aim is to minimize or maximize the objective function [14, 15, 2]. An interesting application of the heuristic algorithm concerns the reduction of energy consumption [16, 17, 18**?**]. An important part of optimization algorithms are algorithms modeled on the behavior of animals cooperating in large groups [19, 20]. These algorithms, imitating the behavior of the community, e.g. ants and bees, allow you to quickly and effectively achieve the goal. The third direction of the development of artificial intelligence are all kinds of methods based on artificial neural networks [21, 22]. They are widely used in medicine, in the care of the elderly [23, 24, 25], in detection [26, 27] as well as in machine learning [28, 29, 30, 31].

We created a program that allows you to choose a career path based on specific factors. The program will make it possible to select the optimal result using the k nearest neighbors algorithm and using soft sets. We create a table for soft sets with the accuracy of various types of distance calculation methods in the KNN algorithm. The soft set table consists of columns that are a specific factor on which we focus, and the rows are the next algorithms from KNN, while the content of the table is the accuracy that we obtained using a specific KNN algorithm.

The program is written in Python, has no graphical interface and is executed in the IDE. The data is in the form of a database in the *.csv* file, while the user enters the weights and values for each column in the form of a list directly in the program.

2. K-nearest neighbors Algorithm

The K Nearest Neighbors algorithm is a ranking algorithm, it evaluates to which group the point belongs to from the current iteration of the algorithm in the surface. The classification works on the basis of counting the number of the nearest neighbors points in a given group, the score is returned based on the vote of the majority.

Data analysis is based on clustering. The program classifies data based on different variants of the KNN (k-nearest neighbors) algorithm. It consists in finding the k elements already classified (neighbors) closest to the new element and assigning this element to the group to which most of its neighbors belong. Several metrics are used to determine the similarity, this program uses two: Manhattan (Taxi Cab) and Minkowski.

Manhattan metric

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{1}$$

Where:
d – distance,

SYSYEM 2022: 8th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Brunek, July 23, 2022

bodorajoanna@polsl.pl (J. Bodora); jadwcad575@polsl.pl (J. Cader); nikogeb061@polsl.pl (N. Gębka)

x – value of a sample,
y – value of a classified element,
n – amount of elements in the sample

Minkowski metric – a modified Euclidean metric

$$L_m(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{1/m}. \tag{2}$$

Where:
d – distance,
x – value of a sample,
y – value of a classified element,
n – amount of elements in the sample
m – any small integer,

After calculating the distance, the data is clustered: first sorted in ascending order, then voting is done on the basis of a 1:1 matching of the sample attribute to the test set attribute - the same elements are added to the common set. Then, the percentage share of the searched elements in relation to the entire data set is calculated:

$$Accuracy = \frac{sizeofthesetofmatchedelements}{sizeofthewholedataset} \times 100\%$$

The variable *k* largely determines the behavior of the classifier. Determines the number of the closest neighbors that decide on the classification of the element. It is a natural number. This parameter is arbitrary, but if we want our classifier to work efficiently, we must make a few assumptions:

- *K* must be greater than the square root of the number of all classified elements

  $$k \geq \sqrt{n},$$

  n - number of classified elements
- If the number of groups is even, k must be odd. Otherwise, k must be even.

  $$k = \begin{cases} 2a + 1, & c|2 \\ 2a, & \text{otherwise} \end{cases} \tag{3}$$

  c – number of groups, $a \in N$
- *K* must be greater than the number of groups

  $$k > c$$

3. Soft Set

Let U be the initial infinite set and E the set of parameters or attributes relative to U. Let $P(U)$ denote the power set U i $A \subseteq E$. The (F, A) pair is called the soft set above U, where F is the mapping given by $F : A \to P(U)$. Others in words, the soft set (F, A) over U is a parameterized family of the subset U. For $e \in A, F(e)$ can be considered a set of e-elements or e-approximate elements of soft sets (F, A). Thus, (F, A) is defined as:

$$(F, A) = \{F(e) \in P(U) : e \in E, F(e) = \varnothing \text{ , if } e \notin A\}$$

$$\sum_{i=1}^{n} s_i \cdot w_i \tag{4}$$

- s_i – element of the sample
- w_i – weight
- n – length of the sample

4. Other methods used

Cross validation– a statistical method involving division statistical sample for subsets, and then conducting analyzes of the training set, while the test set is used to confirm the plausibility of its results.

Rule extraction – rejection of variables not useful in the study.

Data normalization is scaling data into a range

Min-max normalization using a linear function, it reduces the data to the interval specified by the user (`newmin, newmax`). At the same time, we should know the range that the data can achieve. If we do not know it, we can use the highest and the smallest value in the analyzed set.

$$x' = \frac{x - min}{max - min} \cdot new_{max} - new_{min} + new_{min}$$

This algorithm is used for both regression and classification. Useful when dependencies between objects of the same classes are difficult to interpret.

5. Database

The project was created with the use of a database taken from the website https://www.kaggle.com. The database deals with salaries in individual professions in work related to the field of data analysis.

Database link:
https://www.kaggle.com/datasets/saurabhshahane/data-science-jobs-salaries

6. Implementation of KNN algorithm

The final program was developed to return best KNN algorithms based on accuracy which we get from analyzing different options. We implemented two types of KNN algorithms, one based on distances between values of sample and dataset tried to give best job position sorting by distances and summing appearance of various job titles. Second algorithm also calculated distances but firstly it focused on getting a specific category of work and then from this limited collection of data it returned nearest neighbours for job positions. Both of these algorithms were closely analyzed and results showed that classic KNN algorithm without any categorization gives best accuracy.

```
Data: Input $sample$, $dataTab$, $k$
Result: $jobTitle$
$dist := []$;
$classes := []$;
while $i < len(dataTab)$ do
    Calculate distance between sample and record
      in $dataTab$, save it to $dist$;
end
Add $dist$ as new column to $dataTab$;
Sort $dataTab$ by column $dist$;
for $i$ in range(0,k) do
    Save number of different job title's
      occurrences for $k$ first records in $dataTab$ to
      $classes$;
end
return $jobTitle$ that appeared most frequently
  in $classes$;
```

Algorithm 1: Algorithm of our implementation of KNN

7. Analyzing dataset

The histogram Fig. 1 and plot Fig. 3 show the performance of the earnings in the field of *datascience.* It informs us that there are over 160 people earning between 0 to 10000 USD per year. We note that earnings cumulate in the range of approximately 50000 to 200000 USD. The remaining values are sporadic and we look at them as outliers.

From the chart Fig. 2, we obtain information about the earnings for a specific position. We also note the number of records that will define a given job. Positions such as **Data Scientist** or **Data Engineer** have more records than, for example **Data Specialist**, which appears only once in the database. Not having the same number of records for different positions will affect the accuracy of the classified data. Also salaries in different jobs positions overlap in ranges, which may make it difficult to distinguish positions based on the amount of salary.

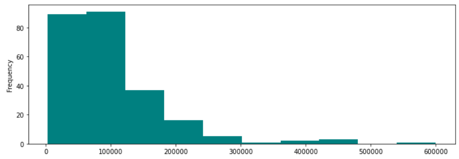


Figure 1: Histogram presenting values of annual salary

Plots Fig. 4 and Fig. 5 presenting the connection between the location or the nationality of employee and the amount of salary shows that the research was conducted mainly on the American market, also the scope of salaries of employees of different nationalities and companies from other countries rather coincides, i.e. the amount of the salary does not depend on the citizenship of the employee or the country in which he works. Therefore it can be concluded that there are certain salary scales that are offered in IT positions in data analysis regardless of location or nationality.

Fig. 6 shows the connection between the employee's level of experience and his salary. The highest rate was offered to the person with the greatest responsibility, i.e. working in an executive position, for example the position of director, leader or project manager. Then the seniors have the highest stake. The lowest stake is accumulated in the junior experience group. There are also single outliers in each group.

Fig. 7 checks if there is any connection between the amount of the salary and company's size. We may notice the lack of huge differences in stakes for employees from various companies.

Pie charts Fig. 8 and Fig. 9 were generated to verify the percentage of various work modes and the types of employments. It shows that remote or semi-remote work is provided in almost 85 percent of positions, while full-time employment predominates in the type of employment.

Summing up, the data available does not stand out for a specific group of job positions or, for example, for a certain location of the company, which may result in the difficulty of their classification and lower accuracy. The lack of visible boundaries in the rates due to the size of the company shown in Fig. 7 or the small number of records for certain positions Fig. 2 will be factors that make classification difficult. Also, the predominance of the location of companies and the citizenship of employees from the United States makes the data reflect the reality rather for developed countries.

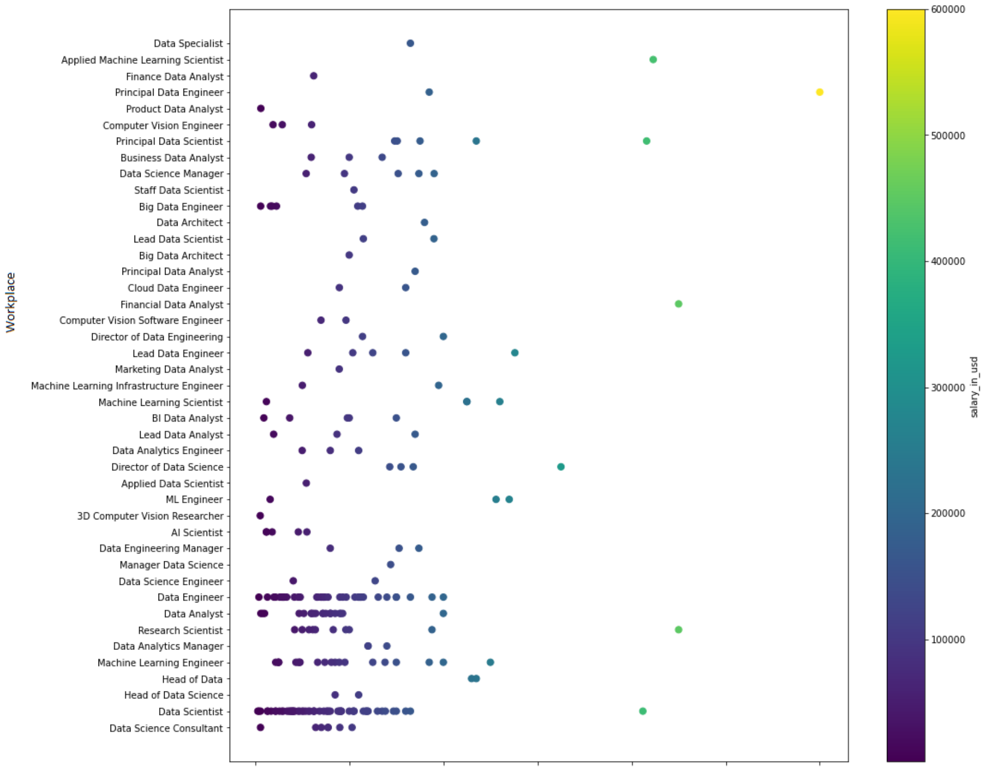


Figure 2: Plot presenting values of annual salary according to the job title

8. Analyzing KNN performance

Presented KNN algorithms have achieved an accuracy between 37 to 89% for classification based on job title. Data were divided in proportions adequately 30% testing and 70% training part. Results were analyzed to determine perfect combination of dataset, k parameter and variety of distance metrics used in KNN algorithm. We focused on two types of distance metrics Minkowski and Manhattan. Comparison test consist on checking performance of KNN algorithm on normalized dataset, not normalized dataset and normalized data but only in salary column.

Graphs presented in Fig. 10 show the influence of k on the accuracy of the algorithm for k nearest neighbors using an additional column of job categories. We can see that for k equal to 8 there is a sudden decrease in accuracy for both the Minkowski method of distance calculation and the taxi method. Then the values from k equal to 9 decrease. Better accuracy is obtained by using the Manhattan distance metrics.

The impact of k on accuracy shown in Fig. 11, informs us that normalizing all columns with little variation in data does not allow algorithm to classify properly. As a result, we get low accuracy of the algorithm's operation. Therefore, in further action, despite the re-verification of the operation on normalized values, we gave up using this normalized data due to the very low accuracy.

We may notice on Fig. 12 that normalizing only salary column itself, which initially takes values in thousandths, allows to increase the accuracy of the classification with the use of job type categorization. One more time, the taxicab metric is a better method of calculating distances.

Graph and table on Fig. 13 show the accuracies for different k using the classic KNN algorithm without additional categorization. The accuracy values are practically the same with minimal variation depending on the distance metric used.

Working on completely normalized data in each of the columns turns out to be pointless due to the very low accuracy that we obtain regardless of the parameter k Fig. 14. Therefore, in the created table for the soft set algorithm, we do not take into account the accuracy obtained when working on this type of data sets.

In the presented graphs Fig. 15, we may notice that the parameter k affects the determination of the accuracy.

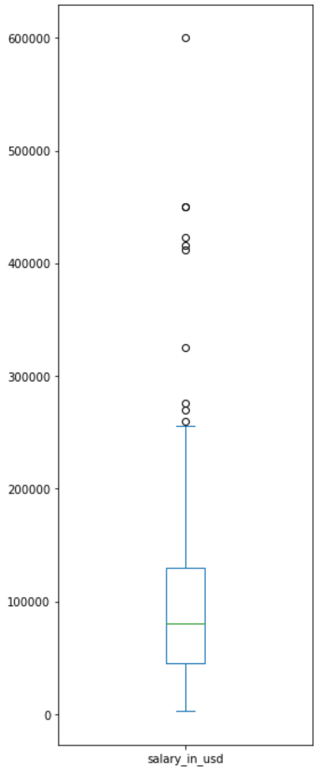


Figure 3: Plot presenting values of annual salary

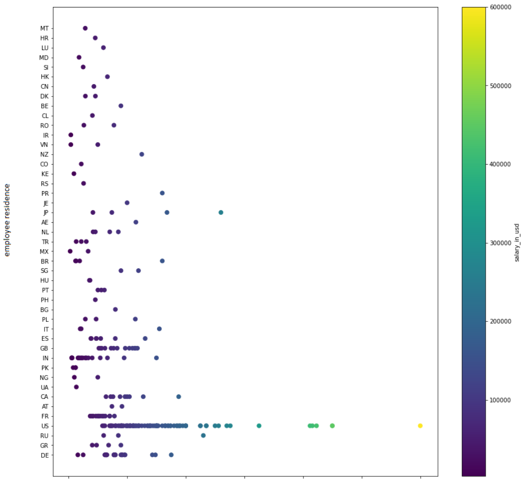


Figure 4: Plot presenting values of annual salary according to the nationality of an employee

In the graphs on the left, which uses the Minkowski metric to calculate the distance, we see that the accuracy remains high for the initial 4 k values and then gradually decreases. On the other hand, when using the Manhattan metric, values decrease from the intial k.

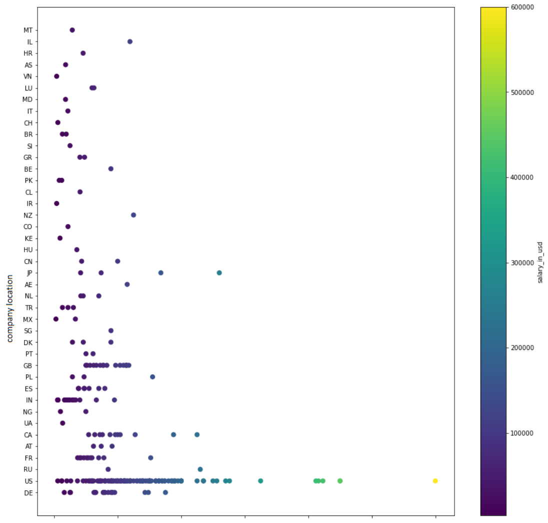


Figure 5: Plot presenting values of annual salary according to the company location

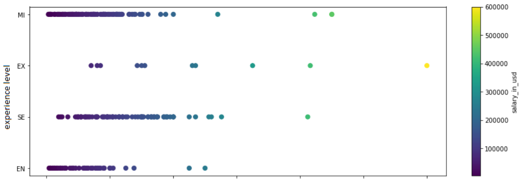


Figure 6: Plot presenting values of annual salary according to the employee's experience level

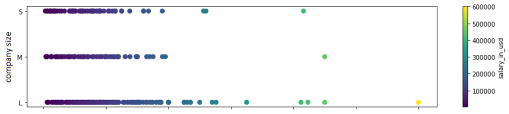


Figure 7: Plot presenting values of annual salary according to the company size

9. Experiments

Table Fig. 16 presents the obtained table for the operation of the soft set algorithm. This table contains accuracies for the following KNN algorithms from the lines, using the given parameter k as well as a specific data set. We obtain this soft set table after analyzing for which parameters k gives the best accuracy.

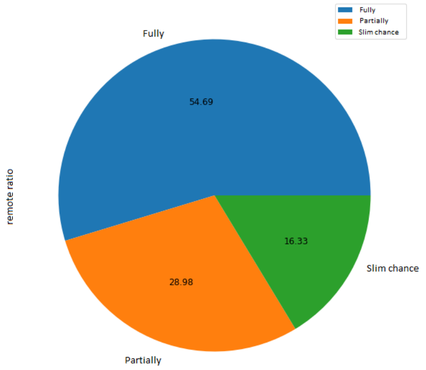


Figure 8: Pie chart presenting the percentage of different types of work

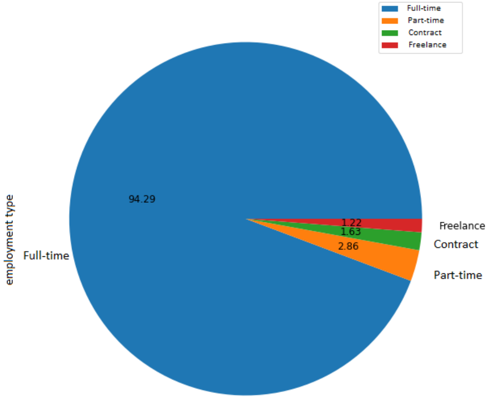


Figure 9: Pie chart presenting the percentage of different form of employments

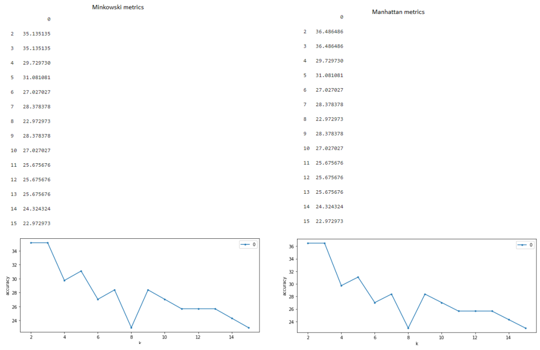
10. Conclusion

As we can see presented solution allows the user to find perfect job position based on factors, which he or she focuses on. Because of in-depth reporting of data set we could distinguish best combinations of KNN algorithm in terms of k parameter, distance metric and data set itself. Thanks to creating soft set table of accuracies of different KNN solutions we get best algorithm, which also gives factors we focus on the most the utmost importance.

A. Online Resources

The sources for the solution are available via

- GitHub

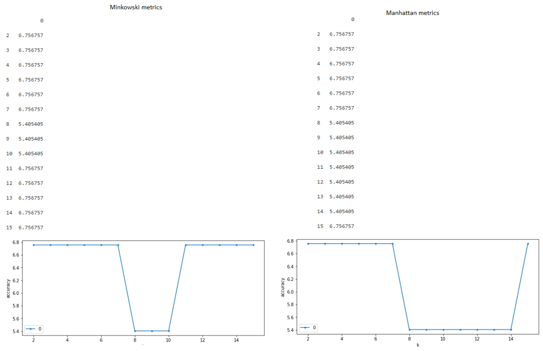

Minkowski metrics

```
            0
2   35.135135
3   35.135135
4   29.729730
5   31.081081
6   27.027027
7   28.378378
8   22.972973
9   28.378378
10  27.027027
11  25.675676
12  25.675676
13  25.675676
14  24.324324
15  22.972973
```

Manhattan metrics

```
            0
2   36.486486
3   36.486486
4   29.729730
5   31.081081
6   27.027027
7   28.378378
8   22.972973
9   28.378378
10  27.027027
11  25.675676
12  25.675676
13  25.675676
14  24.324324
15  22.972973
```

Figure 10: Results and plots presenting impact of K parameter on accuracy of KNN classification with category of not normalized values using Minkowski and Manhattan distance metrics

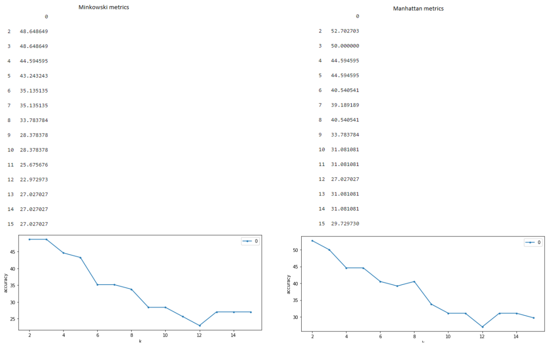

Minkowski metrics

```
           0
2   6.756757
3   6.756757
4   6.756757
5   6.756757
6   6.756757
7   6.756757
8   5.405405
9   5.405405
10  5.405405
11  6.756757
12  6.756757
13  6.756757
14  6.756757
15  6.756757
```

Manhattan metrics

```
           0
2   6.756757
3   6.756757
4   6.756757
5   6.756757
6   6.756757
7   6.756757
8   5.405405
9   5.405405
10  5.405405
11  5.405405
12  5.405405
13  5.405405
14  5.405405
15  6.756757
```

Figure 11: Results and plots presenting impact of K parameter on accuracy of KNN classification with category of normalized values using Minkowski and Manhattan distance metrics

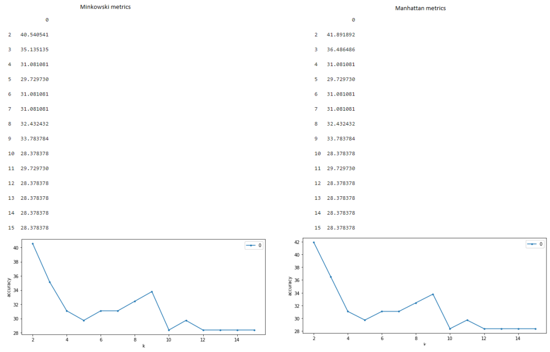

Minkowski metrics

```
            0
2   48.648649
3   48.648649
4   44.594595
5   43.243243
6   35.135135
7   35.135135
8   33.783784
9   28.378378
10  28.378378
11  25.675676
12  22.972973
13  27.027027
14  27.027027
15  27.027027
```

Manhattan metrics

```
            0
2   52.702703
3   50.000000
4   44.594595
5   44.594595
6   40.540541
7   39.189189
8   40.540541
9   33.783784
10  31.081081
11  31.081081
12  27.027027
13  31.081081
14  31.081081
15  29.729730
```

Figure 12: Results and plots presenting impact of K parameter on accuracy of KNN classification with category of normalized values only in salary column using Minkowski and Manhattan distance metrics



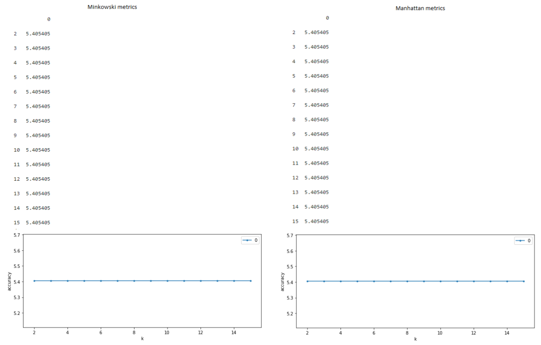
Minkowski metrics

```
            0
2   40.540541
3   35.135135
4   31.081081
5   29.729730
6   31.081081
7   31.081081
8   32.432432
9   33.783784
10  28.378378
11  29.729730
12  28.378378
13  28.378378
14  28.378378
15  28.378378
```

Manhattan metrics

```
            0
2   41.891892
3   36.486486
4   31.081081
5   29.729730
6   31.081081
7   31.081081
8   32.432432
9   33.783784
10  28.378378
11  29.729730
12  28.378378
13  28.378378
14  28.378378
15  28.378378
```

Figure 13: Results and plots presenting impact of K parameter on accuracy of KNN classification of not normalized values using Minkowski and Manhattan distance metrics



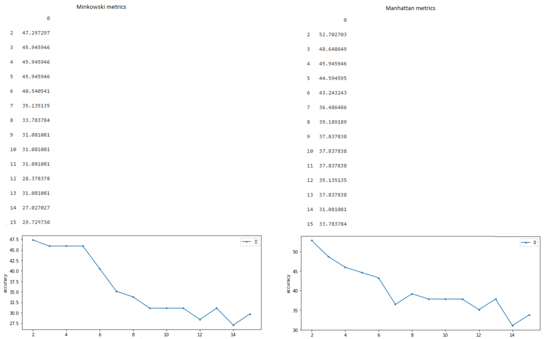
Minkowski metrics

```
           0
2   5.405405
3   5.405405
4   5.405405
5   5.405405
6   5.405405
7   5.405405
8   5.405405
9   5.405405
10  5.405405
11  5.405405
12  5.405405
13  5.405405
14  5.405405
15  5.405405
```

Manhattan metrics

```
           0
2   5.405405
3   5.405405
4   5.405405
5   5.405405
6   5.405405
7   5.405405
8   5.405405
9   5.405405
10  5.405405
11  5.405405
12  5.405405
13  5.405405
14  5.405405
15  5.405405
```

Figure 14: Results and plots presenting impact of K parameter on accuracy of KNN classification of normalized values using Minkowski and Manhattan distance metrics



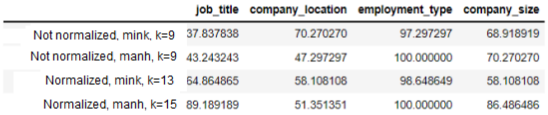
Minkowski metrics

```
            0
2   47.297297
3   45.945946
4   45.945946
5   45.945946
6   40.540541
7   35.135135
8   33.783784
9   31.081081
10  31.081081
11  31.081081
12  28.378378
13  31.081081
14  27.027027
15  29.729730
```

Manhattan metrics

```
            0
2   52.702703
3   48.648649
4   45.945946
5   44.594595
6   43.243243
7   36.486486
8   39.189189
9   37.837838
10  37.837838
11  37.837838
12  35.135135
13  37.837838
14  31.081081
15  33.783784
```

Figure 15: Results and plots presenting impact of K parameter on accuracy of KNN classification of normalized values only in salary column using Minkowski and Manhattan distance metrics



	job_title	company_location	employment_type	company_size
Not normalized, mink, k=9	37.837838	70.270270	97.297297	68.918919
Not normalized, manh, k=9	43.243243	47.297297	100.000000	70.270270
Normalized, mink, k=13	64.864865	58.108108	98.648649	58.108108
Normalized, manh, k=15	89.189189	51.351351	100.000000	86.486486

Figure 16: Table showing the final accuracies for selected algorithms on specific data

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

[3] J. W. W. L. Z. B. Wei Dong, Marcin Woźniak, Denoising aggregation of graph neural networks by using principal component analysis, IEEE Transactions on Industrial Informatics (2022).

[4] 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.

[5] 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.

[6] A. Carpenzano, R. Caponetto, L. Lo Bello, O. Mirabella, Fuzzy traffic smoothing: An approach for real-time communication over ethernet networks, in: 4th IEEE International Workshop on Factory Communication Systems, IEEE, 2002, pp. 241–248.

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

[8] 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.

[9] C. Napoli, G. Pappalardo, E. Tramontana, Improving files availability for bittorrent using a diffusion model, in: Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, IEEE Computer Society, 2014, pp. 191–196. doi:10.1109/WETICE.2014.65.

[10] T. Qiu, B. Li, X. Zhou, H. Song, I. Lee, J. Lloret, A novel shortcut addition algorithm with particle swarm for multisink internet of things, IEEE Transactions on Industrial Informatics 16 (2019) 3566–3577.

[11] D. Yu, C. P. Chen, Smooth transition in communication for swarm control with formation change, IEEE Transactions on Industrial Informatics 16 (2020) 6962–6971.

[12] G. Capizzi, G. Lo Sciuto, C. Napoli, R. Shikler, M. Wozniak, Optimizing the organic solar cell manufacturing process by means of afm measurements and neural networks, Energies 11 (2018).

[13] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, An advanced neural network based solution to enforce dispatch continuity in smart grids, Applied Soft Computing Journal 62 (2018) 768 – 775.

[14] J. Yi, J. Bai, W. Zhou, H. He, L. Yao, Operating parameters optimization for the aluminum electrolysis process using an improved quantum-behaved particle swarm algorithm, IEEE Transactions on Industrial Informatics 14 (2017) 3405–3415.

[15] C. Napoli, G. Pappalardo, E. Tramontana, Using modularity metrics to assist move method refactoring of large systems, in: Proceedings - 2013 7th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2013, 2013, pp. 529–534. doi:10.1109/CISIS.2013.96.

[16] F. Bonanno, G. Capizzi, C. Napoli, Some remarks on the application of rnn and prnn for the charge-discharge simulation of advanced lithium-ions battery energy storage, in: SPEEDAM 2012 - 21st International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2012, pp. 941–945. doi:10.1109/SPEEDAM.2012.6264500.

[17] 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.

[18] F. Bonanno, G. Capizzi, A. Gagliano, C. Napoli, Optimal management of various renewable energy sources by a new forecasting method, 2012, pp. 934–940. doi:10.1109/SPEEDAM.2012.6264603.

[19] 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.

[20] 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.

[21] 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.

[22] C. Napoli, F. Bonanno, G. Capizzi, An hybrid neuro-wavelet approach for long-term prediction of solar wind, Proceedings of the International Astronomical Union 6 (2010) 153 – 155.

[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] S. Illari, S. Russo, R. Avanzato, C. Napoli, A cloud-oriented architecture for the remote assessment and follow-up of hospitalized patients, in: CEUR Workshop Proceedings, volume 2694, CEUR-WS, 2020, pp. 29–35.

[25] N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, in: CEUR Workshop Proceedings, volume 3118, CEUR-WS, 2021, pp. 51–63.

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

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

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

[29] N. Brandizzi, V. Bianco, G. Castro, S. Russo, A. Wajda, Automatic rgb inference based on facial emotion recognition, in: CEUR Workshop Proceedings, volume 3092, CEUR-WS, 2021, pp. 66–74.

[30] R. Brociek, G. Magistris, F. Cardia, F. Coppa, S. Russo, Contagion prevention of covid-19 by means of touch detection for retail stores, in: CEUR Workshop Proceedings, volume 3092, CEUR-WS, 2021, pp. 89–94.

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