DOI: http://dx.doi.org/10.26483/ijarcs.v14i3.6989

ISSN No. 0976-5697

Volume 14, No. 3, May-June 2023

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

INCORPORATING DENSITY IN K-NEAREST NEIGHBORS REGRESSION

Mohamed A.Mahfouz
Faculty of Computer Science,
MSA University, Egypt

Abstract: The application of the traditional *k*-nearest neighbours in regression analysis suffers from several difficulties when only a limited number of samples are available. In this paper, two decision models based on density are proposed. In order to reduce testing time, a k-nearest neighbours table (*k*NN-Table) is maintained to keep the neighbours of each object *x* along with their weighted Manhattan distance to *x* and a binary vector representing the increase or the decrease in each dimension compared to *x*'s values. In the first decision model, if the unseen sample having a distance to one of its neighbours *x* less than the farthest neighbour of *x*'s neighbour then its label is estimated using linear interpolation otherwise linear extrapolation is used. In the second decision model, for each neighbour *x* of the unseen sample, the distance of the unseen sample to *x* and the binary vector are computed. Also, the set S of nearest neighbours of *x* are identified from the *k*NN-Table. For each sample in S, a normalized distance to the unseen sample is computed using the information stored in the *k*NN-Table and it is used to compute the weight of each neighbor of the neighbors of the unseen object. In the two models, a weighted average of the computed label for each neighbour is assigned to the unseen object. The diversity between the two proposed decision models and the traditional *k*NN regressor motivates us to develop an ensemble of the two proposed models along with traditional *k*NN regressor. The ensemble is evaluated and the results showed that the ensemble achieves significant increase in the performance compared to its base regressors and several related algorithms.

Keywords: Small Data, Ensemble Algorithms, Nearest Neighbors Regression, Neighborhood Component Analysis

1. INTRODUCTION

An estimation of a real-valued continuous response (output) based on the values of one or more input variables is referred to as a regression issue in machine learning. A regression approach find the relationships between output and input factors to predict a target value numerically. Various regression techniques have been proposed in the literature. Due to its ease of use and robustness, its ability to learn non-linear decision boundaries, its ability to evolve with new data since no explicit training phase, has only a single hyper parameter to be tuned and can be applied using several distance metrics, k-nearest neighbour regression (*k*NNR) [1] and [2] has emerged as one of the most popular regression approaches [3]. This approach is a modified version of the k-nearest neighbour (*k*NN) model, which is first proposed as a solution to classification issues in [4]. For finding the *k* nearest neighbors, the distance between the unseen sample and all training samples should be calculated. Thus, when *k*NN applied to very large dataset it suffers high computational complexity. In {Mahfouz, 2018 #46}, rough and fuzzy sets concepts are applied to distinguish between core and border objects. The author partitions data into several clusters, and then, for each unseen sample, the nearest neighbors would be searched in one core cluster and some border clusters according to its membership in the clusters. In {Saadatfar, 2020 #47}, other factors are considered such as different cluster shapes and densities which may have influences on choosing the proper cluster. On the other side, when kNN is applied to small dataset, it may overfit, combining several diverse decision models may be a solution for this problem {Mahfouz, 2021 #48}.

In *k*NNR, the output value for a specific test sample is computed by averaging the results of the samples closest to the test sample [5]. Even though the KNN approach has several noteworthy benefits above, it has some inherent flaws, such as the fact that it treats all nearest neighbors equally in the classification process (even though some of them are extremely far from the test sample). To enhance the model and resolve such problems, [6] proposed the concept of using the *k*NN method's degree of membership to suggest a fuzzy version of the algorithm known as the fuzzy k-nearest neighbors (FKNN) classifier. The FKNN model has shown promise for classification challenges due to its ability to address data uncertainty issues [7, 8] compared to the classical *k*NN method. Although the FKNN classifier has garnered a lot of interest in the classification context, regression has received less attention. This motivates the authors of [9] to propose the fuzzy k-nearest neighbor regression (FKNNreg) model by modifying the original FKNN rule.

One of the key elements of distance-based classifiers, such the *k*NN and FKNN techniques, is often the distance metric [10]. Although the Euclidean distance is the most popular distance metric employed in such methods to determine how similar two data samples are, it is sometimes not the best option for all problem domains [11]. With a more diverse selection of distance metrics, better outcomes have been reported in several studies [12] and [13]. The Euclidean distance also has several drawbacks. For instance, in the case of missing data, two data sample pairs may have a shorter distance than other sample pairs with the same feature values if they share no feature values [14]. Using Minkowski distance in the FKNN rule in the regression setting for low- and high-dimensional datasets showed a better results than using Euclidian [9].

1- In this paper parameters K and *k* play different roles (see Table 1).

The *k*NNR model has the capacity to effectively address both linear and non-linear issues [15] It functions admirably in a high-dimensional space in particular. As a result, the *k*NNR approach is becoming more and more common in a variety of industries, including renewable energy [16], physics research [17], biological studies [18], transportation [15], robotics [19], and telecommunication [20]. In several instances, the *k*NNR model has also been combined with other methodologies to create powerful hybrid models for particular applications. For instance, [21] proposed an integrated framework for stock market prediction using support vector machines (SVM) and *k*NNR[22] introduced a brand-new hybrid solution for classification issues that combines a genetic algorithm (GA), the *k*NNR technique, and an artificial neural network (ANN). [23] utilized the same idea as the *k*NNR to introduce a novel approach for missing value imputations. Furthermore, the simplicity and strength of the *k*NNR algorithm have encouraged researchers to develop different enhanced variants, for examples, [3]; [24]; [11]; and to construct mathematical estimations [25]. In order for a distance measure to be ideal, it must be able to accurately identify similarities between two samples while also enabling researchers to compare, categorize, or cluster those samples. As a result, these indicators have a great chance of influencing the results of the models being employed [26]. Thus, some recent studies concentrated only on which similarity metric best suited the specific circumstance [27] and [28]. Makowski distances have an exception for Euclidean distances. The fuzzy theories idea was initially presented by [29], can function under uncertainty and has developed greatly across a wide range of applications [30]. The FKNN classifier [6] was created using fuzzy theory, and it has shown to be one of the best methods for supervised machine learning tasks. [31] applied the FKNN classifier to a regression application without modifying its original algorithm explicitly (i.e., it as operated as a classification task). Also, [9] attempts to utilize the FKNN model in the regression setting. Thus, the effectiveness of *k*NNR for machine learning applications requires further investigation.

The main goal of this study is to introduce density based regressors related to the traditional *k*NN regressor. This led us to develop the Minkowski density-based fuzzy *k*-nearest neighbor regression (MDFI-*k*NNR) algorithm based on interpolation and another density based regressor MDFNN-*k*NNR as a weighted average of the label of the neighbors of neighbors using a normalized distance that is a combination of Minkowski distance metric and hamming distance. A diversity between the proposed models and the traditional *k*NNR is expected, thus, this motivates us to create an ensemble of the two proposed method along with the traditional *k*NNR termed EMDF-*k*NNR. Also, the use of fuzzy weights increases the robustness of the proposed algorithms and the utilization of the Minkowski distance along with hamming distance allows additional freedom to find nearby, more pertinent samples that are close to the unseen sample.

The majority of existing regression models, such as least absolute shrinkage and selection operator (LASSO) regression and multiple linear regression (MLR), intuitively rely on presumptions about the distribution of the data. However, it is rarely proven that these presumptions apply to issues in the real world. In light of this, it's interesting to note that the kNNR and its related methods make no explicit assumptions about the underlying data [18] or model's elements and just utilize training data to generate forecasts. Another benefit is that they can potentially be used for non-linear situations because to their generally simple implementation and interpretation [5]. One of the most popular methods for non-linear regression challenges is support vector regression (SVR). However, its usage is restricted in some areas due to the difficulty in selecting acceptable model parameters. [32].

In summary, few studies have been conducted to capture the increase in density when an unseen sample is added. Sometimes the unseen sample is far from one of its neighbors however it is closer to it than several of its neighbors. Also, how much the difference between the distance between unseen sample and an object *x* which is one of its neighbor compared to the distance of an object y that is also a neighbor of *x*. When this difference is very small it is better to consider the label of y than the label of *x*. This motivates the work described in this paper. To the best of our knowledge, we are the first to propose the relative normalized distance and use it to capture the increase in density by the unseen object to be tested.

For non-linear regression problems, the suggested ensemble EMDF-*k*NNR consists of the two proposed density-based regressors and the conventional *k*NNR is found to be significant, and outperform its base models. We conducted several experiments to evaluate the performance of the suggested models using real-world data from various applications. When compared to multiple linear regression, KNNreg, Lasso, and SVR models, the proposed variant's performance in terms of regression was examined. Additionally, the outcomes of the methods of Manhattan distance-based fuzzy k-nearest neighbor regression (Man-FKNNreg) and Euclidean distance-based fuzzy k-nearest neighbor regression (Euc-FKNNreg) were compared. The effectiveness of the regression was evaluated using the coefficient of determination (R2) and root mean square error (RMSE) measures.

The main contributions of this research study:

1) Two new regression models based on density are proposed.
2) A normalized distance that combines both Minkowski and hamming distance is introduced and is used in the second model to compute the weights for each neighbor of the neighbors of the unseen object.
3) The effectiveness of the suggested regression models is explained on real data from many fields that is both low-dimensional and high-dimensional.
4) The results of the proposed ensemble are compared to the results of several well-known, cutting-edge regression methods.

The rest of this paper is structured as follows. In section 2, the notation, data, similarity and performance metrics along with some background materials are introduced. In section 3, the proposed EMDF-*k*NNR along with its base models is explained. The experimental setup and the empirical findings achieved using the suggested methods are shown and discussed in Section 4. The main conclusions are outlined in Section 5 along with some last thoughts.

2. PRELIMINARIES

2.1. Notation

This section provides supporting material to help the reader better understand the remaining sections. Commonly used abbreviations and symbols are listed in Table 1.

Table 1 Abbreviations and symbols used in the text

Abbreviation	Description
*k*NNR	traditional k-nearest neighbors regressor
EMDF-*k*NNR	The proposed ensemble
MDFNN-*k*NNR	The proposed decision model based on normalized distance to neighbors of neighbors
MDFI-*k*NNR	The proposed decision model based on interpolation
NCA	Neighborhood Component Analysis
X	dataset comprising *n samples* × *d features*
n	total number of samples
x_{ij}	*j*thfeature value of the *i*thsample in the dataset X
KNN-table	A table of size *n*×*k*cells, contains K nearest neighbors and their distances for each sample
k	number of nearest neighbors that are kept in the KNN-table

2.2. Datasets

In our experiment, We used the same eight real-world datasetsthat are used in [9] and freely available at the UCI Machine Learningrepository [33] and at theKnowledge Extraction based on Evolutionary Learning(KEEL) repository [34]. Table 2 shows the number of instances, features and the domain of these datasets.

Table 2 Summary of the datasets used in the experiment

Data set	Repository	Instances	Features	Domain
Stock	KEEL	950	9	Business
Airfoil	UCI	1503	5	Physics
AutoMPG	KEEL	392	6	Engineering
Baseball	KEEL	337	16	Sociology
Servo	UCI	167	4	IT
Laser	KEEL	993	4	Physics
Qsar Fish	UCI	908	6	Biology
Parkinson	UCI	5875	26	Medicine

2.3. Performance Evaluation Metrics

The effectiveness of EMDF-*k*NNR and its base models have been evaluated using R^2. R^2 is a statistical measure that shows how closely the data points in the response variable fit to the values of the regression model. It is measured as the proportion of the variation in the response variable, which is "explained'' by the regression model compared to the mean [35].

$$R^2 = \left(1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}\right) \times 100\% \qquad (1)$$

where *n* is the number of samples in the test data, $\hat{y}_i$ and $y_i$are the predicted value and the true value of the i^{th} test sample, respectively, and $\bar{y}_i$ is the average of the true values. As shown in Eq. (1), the percentage values of R^2 are considered. The higher the value of R the better the regression model is.

2.4. Feature Selection for Regression using Neighborhood Component Analysis(NCA)

NCA is a non-parametric method for selecting and weighting features with the goal of minimizing regression loss and classification accuracy over the training data [38], [39], also, NCA used in unsupervised learning [40]. NCA implementation is available in both Python and MATLAB. For Regression [41], the algorithm computes feature weights such that the average leave-one-out regression accuracy over the whole training data is maximized by minimizing an objective function with regularization term. The objective function of NCA is derived for regression as is derived in for classification using kNN as follows:-

Given a dataset S=(X,y) where X is the feature matrix of size n samples × d features and *y* is the label vector and its elements are real numbers, the aim is to predict the response of unseen sample x given the training set *(X, y).*

The distance between two samples x_i, and x_j using Manhattan distance is computed as follows:-

$$d_w = \sum_{r=1}^{d} |x_{ir} - x_{ir}| \qquad (2)$$

The distance between two samples x_i, and x_j is computed as a weighted Manhattan distance as follows:-

$$d_w = \sum_{r=1}^{d} w_r^2 \ |x_{ir} - x_{jr}| \qquad (3)$$

w_r are the feature weights

Consider a randomized regression model that Randomly picks a point (Ref(*x*)) from *S* as the 'reference point' for *x such that P*(Ref(*x*)= x_j|*S*) ∝$k(d_W(x,x_j))$, and sets the response value at *x* equal to the response value of the reference point Ref(*x*). The probability *P*(Ref(*x*)= x_j|*S*) that point x_j is picked from *S* as the reference point for *x*

$$P(\mathrm{Ref}(x\ \ ) = x_j \mid S) = \frac{k(d_W(x\ \ ,x_j))}{\sum_{j=1}^{n} k(d_W(x\ \ ,x_j))} \qquad (4)$$

Now consider the leave-one-out application of this randomized regression model, that is, predicting the response for x_i using the data in S^{-i}, the training set *S* excluding the point (x_i,y_i). The probability that point x_j is picked as the reference point for x_i is

$$p_{ij} = P(\mathrm{Ref}(x_i) = x_j \mid S) = \frac{k(d_W(x_i,x_j))}{\sum_{j=1}^{n} k(d_W(x_i,x_j))} \qquad (5)$$

Where *k* is some kernel function that results large values when $d_W(x,x_j)$ is small. as recommended in [38]:-

$$k(z) = e^{\frac{-z}{\sigma}}$$

Let $\hat{y}_i$be the response value the randomized regression model predicts and y_i be the actual response for x_i. Using Mean absolute deviation as the loss function that measures the disagreement between $\hat{y}_i$ and y_i, the average loss l_i in predicting y_i is

$$l_i = E(l(y_i, \hat{y}_i)|S - \{s_i\}) = \sum_{j=1,i\neq j}^{n} p_{ij}\ l(y_i, y_j) = \sum_{j=1,i\neq j}^{n} p_{ij}\ |y_i - y_j| \qquad (7)$$

The objective function to be minimized is

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} l_i + \lambda \sum_{r=1}^{d} w_r^2 \qquad (8)$$

3. METHODOLOGY

3.1. First Model: Density-based k-nearest Neighbors Regressor using interpolation (MDFI-kNNR)

In the first model, as shown in fig. 1, a weighted Manhattan distance between the unseen sample$x_u$and all the samples are computed, then, the *k*-nearest neighbors of the unseen sample are identified as 1-NN(x_u)…… *k*-NN(x_u)then the *k*NN-Table is used to identify the neighbours of each i-NN(u) for i=1,2..*k*. Let i-NN(u) be x_i and d(x_u,x_i) between $d(N_j(x_i), x_i)$ and $d(N_{j-1}(x_i), x_i)$.

The label of x_u is computed from the labels of $N_j(x_i)$ and $N_{j-1}(x_i)$ as follows:-

$$y_i = l(N_j(x_i)) + \frac{d(x_u, x_i) - d(N_j(x_i), x_i)}{d(N_j(x_i), x_i) - d(N_{j-1}(x_i), x_i)} (l(N_j(x_i)) - l(N_{j-1}(x_i)))$$

To be noted that for a certain $d(x_u, x_i)$ the triangular inequality implies an upper bound on $d(N_j(x_i), x_u)$ that depends on $d(N_j(x_i), x_i)$,so, from Eq. (4), the probability P(Ref(x_u)= $N_j(x_i)$ |S) that point $N_j(x_i)$ is picked from S as the reference point for x_u in randomized regression model is inversely proportional to $d(N_j(x_i), x_i)$.

If feature weighting using NCA is used the distance between two samples is computed using Eq. (3), otherwise, The distance between two samples x_i, and x_j is computed as a Manhattan distance as in Eq. (2).

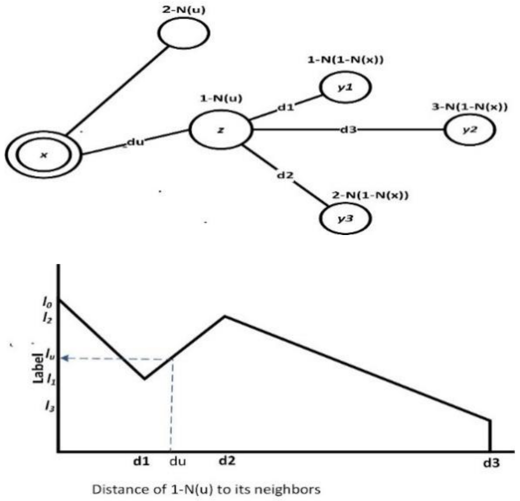

Fig. 1 The distance $d_u$is used in MDFI-kNNR to predict a label for the unseen object x using linear interpolatio

Algorithm 1: MDFI-*k*NNR

```
Let N_j(x_i) is the j^th neighbour of  x_i , j = 1,2,…k
   for each x_i ϵ N(x_u) do
      if (d(x_i , N_k(x_i)) > d(x_u , x_i))
j = the smallest h such that d(x_u , x_i) ≤ d(x_i , N_h(x_i))
   else
      j = k
$y_i = l(N_j(x_i)) + \frac{d(x_u, x_i) - d(N_j(x_i), x_i)}{d(N_j(x_i), x_i) - d(N_{j-1}(x_i), x_i)} (l(N_j(x_i)) - l(N_{j-1}(x_i)))$
$w_i = \frac{1}{(1/d(X_u, N_i(X_u))^{2/(q-1)}}$        // fuzzifier q >1
 end
//estimate $\hat{y}_u$ by taking the weighted average as follows:
$\hat{y}_u = \frac{\sum_{i=1}^{k} w_i\, y_i}{\sum_{i=1}^{k} w_i}$
```

n

3.2. Fuzzy density based kNN regressor using normalized distance to neighbors of neighbors MDFNN-kNNR

In the second model, as shown in fig. 2, if z is a neighbor of unseen object *x* and y is a neighbor of *z*, the distance between *x* and y is measured as a weighted average of the normalized Minkowskidistance between *x* and y and the hamming distance between two binary vectors representing the increase or the decrease in each dimension compared to *z* as follows: -

$$\mathrm{d}(x, y) = \propto \left(\sum_{i=1}^{d} \left(\frac{|x_i - y_i|}{r_i}\right)^q\right)^{1/q} + (1 - \propto) \sum_{i=1}^{d} (((x_i \geq z_i) \text{ and } (y_i < z_i)) \text{ or } ((x_i < z_i) \text{ and } (y_i \geq z_i))) \qquad (9)$$

Where $r_i$is the range of the *i*th feature and ∝ is a number between 0 and 1.

Algorithm 2: MDFNN-*k*NNR

```
for each z ϵ N(x) do
 for each y ϵ N(z)
        compute the normalized distance d(x,y)
between x and y as in Eq. (1)
$w_y = \frac{1}{(1+d(x,y))^{2/(q-1)}}$        // fuzzifier q >1
        $l_z(x) = \frac{\sum_{y \epsilon N(z)} w_y\, l(y)}{\sum_{y \epsilon N(z)} w_y}$
$w_z = \frac{1}{(1+d(x,z))^{2/(q-1)}}$        // fuzzifier q >1
end
//estimate $\hat{y}_u$ by taking the weighted average as follows:
        $l(x) = \frac{\sum_{z \epsilon N(x)} w_z\, l_z(x)}{\sum_{z \epsilon N(x)} w_z}$
```

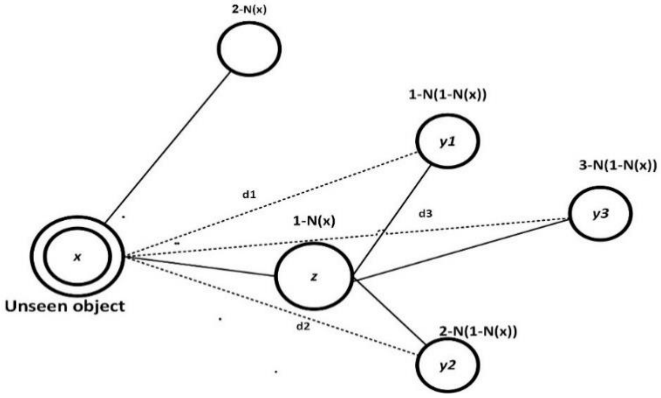

Fig. 2 The normalized distance $d_{1,}d_2$and $d_3$is used in MDFNN-kNNRto predict a label for the unseen object x

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Performance of the proposed algorithm on standard datasets

Table 2 summarizes the classification results of proposed ensemble and compares its accuracy with each of its base models. As shown in Table 2, EMDF-kNNR outperforms all its base classifiers on all datasets. The proposed ensemble outperform all related works on two datasets namely AutoMPG and Servo.

Table 2 comparison with related works on several standard datasets

Data set	MDFI-*k*NNR	MDFNN-kNNR	EMDF-*k*NNR	Md-FKNNreg	Man-FKNNreg	Euc-FKNNreg	*k*NNR [25]	SVR [42]	LASSO [43]	MLR [44]
Stock	0.0286	0.0291	0.0334	0:0294	0:0294	0.0302	0.0311	0.0406	0.0762	0.0407
Airfoil	0.1007	0.1076	0:1042	0:0963	0.0966	0.1002	0.1036	0.0986	0.1342	0.1182
AutoMPG	0.0811	0.0779	**0.0831**	0:0687	0:0687	0.0719	0.0728	0.0707	0.0824	0.0725
Baseball	0.1006	0.1101	0.1421	0:1184	0:1184	0.1239	0.1316	0.1448	0.1329	0.1350
Servo	0.1314	0.1323	**0.1617**	0:1120	0:1120	0.1231	0.1167	0.1577	0.1602	0.1197
Qsar Fish	0.0903	0.0915	0.1002	0:0902	0.0905	0.0917	0.0942	0.0943	0.0976	0.0973
Parkinson	0.0578	0.0712	0.0798	0:0566	0:0566	0.0608	0.0666	0.0786	0.1915	0.1844
Overall	0.0677	0.0849	0.0913	0.0769	0.0770	0.0807	0.0837	0.0917	0.1217	0.1023

4.2 Runtime of the proposed models on Laser data

It is clear from table 3 that the testing time of the proposed models are much higher than the other related work. The testing time kan be further improved using KDD tree to reduce both training and testing time of the proposed algorithm.

Table 3 Runtime comparison on Laser data

Algorithm	Fit Time	Score Time	R^2
SVR	0.036749	2.10E-05	0.752904
RFR	0.197916	4.72E-05	0.721418
DTR	0.005209	9.33E-06	0.94019
MLR	2.017367	9.33E-06	0.968873
Lasso	0.004151	9.33E-06	0.758507
*k*NNR	0	9.33E-06	0.944207
MDFI-*k*NNR	0.004205	9.33E-06	0.924207
MDFNN-kNNR	0.006166	0.000929	0.939836
EMDF-*k*NNR	0.011403	0.000527	0.961322

5. CONCLUSIONS

This research study proposed two novel density based decision models for regression using *k*NN. Unlike the traditional *k*NNR, the proposed algorithm incorporates the increase in density in predicting a label for unseen sample. From our experimental study the following can be concluded:

1) An ensemble of two density-based regressors along with the traditional *k*NNR is proposed.
2) The proposed ensemble is more effective than traditional *k*NNR and outperforms it on all the datasets that are considered in this study.
3) The results of the proposed ensemble are compared to the results of several regression methods on real data from two fields.

The analysis of the scheme proposed in this paper suggests several directions for future work:

1) Applying feature weighting using neighbourhood component analysis.
2) Learning distance Functions that are appropriate for the proposed regressors.
3) Using interpolation methods other than the linear interpolation which is used in the paper.

CONFLICTS OF INTEREST

The author declare no conflict of interest.

REFERENCES

1. Benedetti, J.K., On the nonparametric estimation of regression functions. Journal of the Royal Statistical Society: Series B (Methodological), 1977. **39**(2): p. 248-253.
2. Stone, C.J., Consistent nonparametric regression. The annals of statistics, 1977: p. 595-620.

3. Buza, K., A. Nanopoulos, and G. Nagy, Nearest neighbor regression in the presence of bad hubs. Knowledge-Based Systems, 2015. **86**: p. 250-260.
4. Cover, T. and P. Hart, Nearest neighbor pattern classification. IEEE transactions on information theory, 1967. **13**(1): p. 21-27.
5. Hu, C., et al., Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. Applied Energy, 2014. **129**: p. 49-55.
6. Keller, J.M., M.R. Gray, and J.A. Givens, A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, 1985(4): p. 580-585.
7. Chen, H.-L., et al., An efficient diagnosis system for detection of Parkinson’s disease using fuzzy k-nearest neighbor approach. Expert systems with applications, 2013. **40**(1): p. 263-271.
8. Yu, S., S. De Backer, and P. Scheunders, Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery. Pattern Recognition Letters, 2002. **23**(1-3): p. 183-190.
9. Mailagaha Kumbure, M. and P. Luukka, A generalized fuzzy k-nearest neighbor regression model based on Minkowski distance. Granular Computing, 2022. **7**(3): p. 657-671.
10. Rastin, N., M.Z. Jahromi, and M. Taheri, A generalized weighted distance k-nearest neighbor for multi-label problems. Pattern Recognition, 2021. **114**: p. 107526.
11. Nguyen, B., C. Morell, and B. De Baets, Large-scale distance metric learning for k-nearest neighbors regression. Neurocomputing, 2016. **214**: p. 805-814.
12. Koloseni, D., J. Lampinen, and P. Luukka, Optimized distance metrics for differential evolution based nearest prototype classifier. Expert Systems With Applications, 2012. **39**(12): p. 10564-10570.
13. Koloseni, D., J. Lampinen, and P. Luukka, Differential evolution based nearest prototype classifier with optimized distance measures for the features in the data sets. Expert Systems with Applications, 2013. **40**(10): p. 4075-4082.
14. Shirkhorshidi, A.S., S. Aghabozorgi, and T.Y. Wah, A comparison study on similarity and dissimilarity measures in clustering continuous data. PloS one, 2015. **10**(12): p. e0144059.
15. Cai, L., et al., A sample-rebalanced outlier-rejected $ k $-nearest neighbor regression model for short-term traffic flow forecasting. IEEE access, 2020. **8**: p. 22686-22696.
16. Zhou, Y., M. Huang, and M. Pecht, Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization. Journal of Cleaner Production, 2020. **249**: p. 119409.
17. Durbin, M., et al., K-nearest neighbors regression for the discrimination of gamma rays and neutrons in organic scintillators. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2021. **987**: p. 164826.
18. Yao, Z. and W.L. Ruzzo. A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. in BMC bioinformatics. 2006. BioMed Central.
19. Chen, J. and H.Y. Lau. Learning the inverse kinematics of tendon-driven soft manipulators with K-nearest Neighbors Regression and Gaussian Mixture Regression. in 2016 2nd International Conference on Control, Automation and Robotics (ICCAR). 2016. IEEE.
20. Adege, A.B., et al. Indoor localization using K-nearest neighbor and artificial neural network back propagation algorithms. in 2018 27th Wireless and Optical Communication Conference (WOCC). 2018. IEEE.
21. Chen, Y. and Y. Hao, A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. Expert Systems with Applications, 2017. **80**: p. 340-355.
22. Dell'Acqua, P., et al., Time-aware multivariate nearest neighbor regression methods for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 2015. **16**(6): p. 3393-3402.
23. Cheng, C.-H., C.-P. Chan, and Y.-J. Sheu, A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. Engineering Applications of Artificial Intelligence, 2019. **81**: p. 283-299.
24. Guillén, A., et al., New method for instance or prototype selection using mutual information in time series prediction. Neurocomputing, 2010. **73**(10-12): p. 2030-2038.
25. Biau, G., et al., An Affine Invariant k-Nearest Neighbor Regression Estimate.
26. Bergamasco, L.C.C. and F.L. Nunes, Intelligent retrieval and classification in three-dimensional biomedical images—a systematic mapping. Computer Science Review, 2019. **31**: p. 19-38.
27. Rodrigues, É.O., Combining Minkowski and Chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier. Pattern Recognition Letters, 2018. **110**: p. 66-71.
28. Huo, J., et al., Mahalanobis distance based similarity regression learning of NIRS for quality assurance of tobacco product with different variable selection methods. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 2021. **251**: p. 119364.
29. Goguen, J., LA Zadeh. Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353.-LA Zadeh. Similarity relations and fuzzy orderings. Information sciences, vol. 3 (1971), pp. 177–200. The Journal of Symbolic Logic, 1973. **38**(4): p. 656-657.
30. Zeng, S., S.-M. Chen, and M.O. Teng, Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm. Information Sciences, 2019. **484**: p. 350-366.
31. Nikoo, M.R., R. Kerachian, and M.R. Alizadeh, A fuzzy KNN-based model for significant wave height prediction in large lakes. Oceanologia, 2018. **60**(2): p. 153-168.
32. Liu, X., et al., Effects of temperature on life history traits of Eodiaptomus japonicus (Copepoda: Calanoida) from Lake Biwa (Japan). Limnology, 2014. **15**(1): p. 85-97.
33. Dheeru, D. and E.K. Taniskidou, UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. 2017.
34. Derrac, J., et al., Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult. Valued Logic Soft Comput, 2015. **17**.
35. Kurz-Kim, J.-R. and M. Loretan, On the properties of the coefficient of determination in regression models with infinite variance variables. Journal of econometrics, 2014. **181**(1): p. 15-24.
36. Karahoca, A., Advances in data mining knowledge discovery and applications. 2012: BoD–Books on Demand.
37. Chen, H.-L., et al. An adaptive fuzzy k-nearest neighbor method based on parallel particle swarm optimization for bankruptcy prediction. in Pacific-asia conference on knowledge discovery and data mining. 2011. Springer.
38. Yang, W., K. Wang, and W. Zuo, Neighborhood component feature selection for high-dimensional data. J. Comput., 2012. **7**(1): p. 161-168.
39. Yang, W., K. Wang, and W. Zuo, Fast neighborhood component analysis. Neurocomputing, 2012. **83**: p. 31-37.
40. Qin, C., et al., Unsupervised neighborhood component analysis for clustering. Neurocomputing, 2015. **168**: p. 609-617.
41. Xiong, R., et al., A data-driven method for extracting aging features to accurately predict the battery health. Energy Storage Materials, 2023. **57**: p. 460-470.
42. Drucker, H., et al., Support vector regression machines. Advances in neural information processing systems, 1996. **9**.
43. Tibshirani, R., Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 1996. **58**(1): p. 267-288.
44. Jammalamadaka, S.R., Introduction to linear regression analysis. 2003, Taylor & Francis.