



Research Article

The effects of the feature reduction by machine learning algorithms and feature decomposition by fuzzification on the mathematical model success criterion

Filiz KANBAY^{1,*}, Melike GÖKSU¹

¹Department of Mathematics, Yıldız Technical University, Istanbul, 34220, Türkiye

ARTICLE INFO

Article history

Received: 01 July 2025

Revised: 24 October 2025

Accepted: 27 January 2026

Keywords:

Feature Construction; Feature Reduction; Fuzzy Logic; Machine Learning; Membership Function

ABSTRACT

In this study, the effects of reducing the number of features or classes and creating new features on the model success criteria were examined. For this purpose, the parameters that will maximize the model success of the models built with k-nearest neighbor algorithm, naive bayes algorithm, support vector machine, random forest algorithm, boosting, decision trees, artificial neural networks algorithms for the selected dataset were investigated and the model with the best performance was determined. In order to determine the best model, the Wilcoxon test was applied to all algorithm results evaluated on the same layers. To provide a more balanced evaluation, weighted classification report was generated and additionally model accuracy values were obtained by applying SMOTE. All the results obtained were evaluated together and the best performed model was determined. Visualization of the data was plotted based on the two features with the highest impact on the model. Firstly the effects of supervised and unsupervised dimensionality reduction technique, such as principal component analysis, linear discriminant analysis, neighbourhood components analysis, non-negative matrix factorization, on the model performance were examined; then a method was proposed that calculates fuzzy set number and membership degree of each feature, and adds them to the dataset, and effects of using it together with dimensionality reduction methods on model success were examined. As a result of the analysis, it was observed that applying the proposed technique to the dataset before reducing it to two dimensions using the principal component analysis enhanced performance of all algorithms.

Cite this article as: Kanbay F, Göksu M. The effects of the feature reduction by machine learning algorithms and feature decomposition by fuzzification on the mathematical model success criterion. Sigma J Eng Nat Sci 2026;44(2):1029–1040.

INTRODUCTION

The importance of designing and analyzing artificial learning experiments has become even more evident as

data mining methods have become increasingly widespread in many different fields such as robotics, medical diagnosis, biometrics and finance, thanks to the developments in the

*Corresponding author.

*E-mail address: fkanbay@yildiz.edu.tr

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic



field of artificial learning in recent years [1]. In data mining, to make classification using a small number of features (a subset of features or combining features) allows us to better understand the process that produces this data, saves the cost required for measuring and obtaining an unnecessary input, and makes training the model easier by reducing the complexity of the model [1]. For this purpose, feature extraction and feature decomposition algorithms can be used. However, feature construction method which builds new features from the existing features in the dataset can be useful to maximize the model performance [2]. Vouk et al. [3] in their work give a summary of feature construction approaches and divide them into tree categories such as operator based feature construction, methods embedded into automat M.L. and feature interactions. Hummel in his work mentioned the different methods whose a common component is the use of additional values for involving uncertainty concepts in evidential reasoning. Moreover, fuzzy set theory presented by Zadeh [4] has been applied in practice in various disciplines such as data processing, decision support and others [5-17]. For example, the fuzzy set theory applied to multi attribute analysis by Bellman and Zadeh [18]. Bezdek developed fuzzy cluster method [19]. Fuzzy Multiple Attribute Decision Making Methods can be examined in [20]. Fuzzy soft set theoretic approach to decision making problems were studied [15,21]. Zadeh presented a mathematical basis the concept of linguistic variable whose values are words or sentences in natural or artificial language [22]. In [23], methods for measuring amount of information provided by a linguistic summary were discussed; the use of linguistic summaries as a tool for representing knowledge in data was suggested. After Yager many researcher studied on linguistic summary. Herrera and Martinez presented a model composed by a pair of values called 2 tuple (a linguistic term and a term) [14, 24]. Ranking fuzzy numbers is often a necessary step in many mathematical models [16]. There are several studies on the fuzzy versions of machine learning algorithms and classification methods: [9-10, 12, 25-34]. Der-Chiang Li and et al. mention that having too many features makes data analysis inefficient, while too few reduces model stability, and define a fuzzy logic-based transformation on the features using class labels [35]. Fuzzy sets are used in feature selection with the ability to handle uncertain and in complete data [36]. Unsupervised feature reduction algorithm is proposed by developing the fuzzy improved approximation set method for mixed data [37]. By defining a fuzzy learning ensemble system, unnecessary information is reduced and computational efficiency is increased [38]. Fuzzy rough approximation is used to extract the most important features from the low resolution image [39]. A new distance definition is made to measure the difference between nominal features and two feature algorithms based on this distance are proposed [40]. Dimensionality reduction techniques are used in many different areas [41-43]. The rest of the paper is structured into three parts. Firstly, the data

description and the accuracy scores of some machine learning algorithms are presented. Secondly the number of features are reduced by using the feature reduction and feature decomposition methods. So the classification can be illustrated in the plane. Moreover the effects of reducing of the number of the classes are examined. In last part, in order to enhance accuracy of the model, as a novel approach, fuzzification method is utilized.

MATERIALS AND METHODS

Model Accuracy According to Machine Learning Algorithms

In this study, Python tools, including scikit-learn, Numpy, Pandas, Matplotlib, Seaborn, XGBoost were used within Jupyter Notebook to analyze a dataset consisting of 5 features with 403 samples and no missing data [44,45]. Dataset named 'User Knowledge Modeling' obtained from Ph.D. Thesis uploaded from the UCI Machine Learning Repository used for the empirical analysis of machine learning algorithms (<https://archive.ics.uci.edu/>) [46]. Kahraman et al. designed a model. The reason for designing this model is to disseminate adaptive smart instructional technology applications, as well as to enable knowledge space models to be developed without the support of software developers and to prevent students from getting lost in the hypermedia environment. This designed system has been tested. When the experimental analysis tool, which can make fourteen different pedagogical evaluations at the same time, was applied. The accuracy of following the pages related to an instructional goal of the students using the intelligent learning system was much higher than the users of the traditional teaching application [47]. It is the real dataset about the students' knowledge status about the subject of Electrical DC Machines. The inputs of the dataset (five features): Students degree of study time related to learning objects (STG), number of repetitions grade (SCG), user information performance (PEG) and degree of study time related to the student's prerequisite subjects (STR), the learning percentage (LPR) are classified as the knowledge levels of students (UNS) defined four different classes: high, medium, low and very low [48]. Mathematical models were built and the results were visualized using following machine learning algorithms with Python tools [44,45,49]. The model hyperparameters were determined using a grid search on dataset split into %80 train %20 test. Using these optimized hyperparameters the model's performance was then evaluated on the entire dataset via 5-fold cross validation. By using k nearest neighbour algorithm (KNN) that classifies a query point with the majority vote of its nearest neighbors in training dataset, the results obtained with different metrics (Minkowski distance, Euclidean distance and Manhattan distance) are visualized in Figure 1. So Figure 1 shows that the highest accuracy score of KNN is 88%.

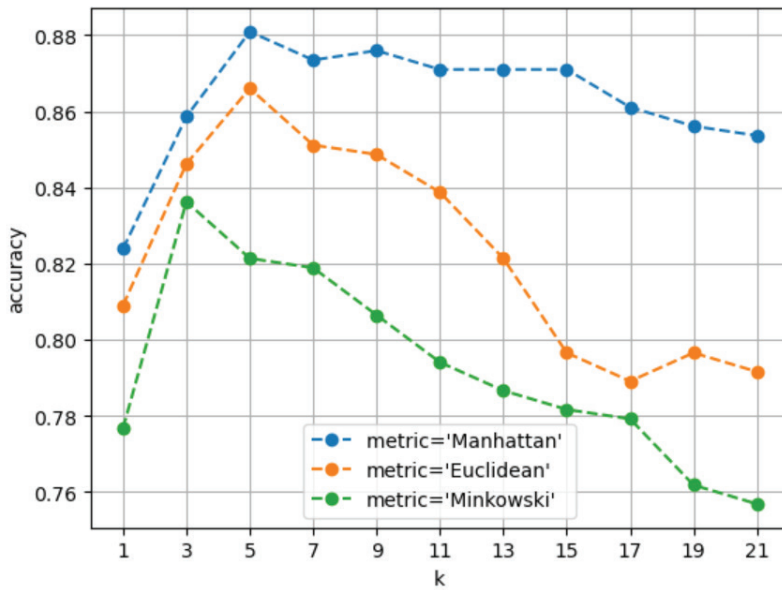


Figure 1. The accuracy score of KNN Model according to the number of neighborhoods.

In addition, the accuracy score of the Naive Bayes algorithm, which enables the classification of unlabeled data by using the conditional probability function for labeled data is 87.59%; the accuracy score of the Decision Trees algorithm is 93% with the best parameters {‘criterion’: ‘gini’, ‘max_depth’: 4, ‘max_features’: None, ‘splitter’: ‘best’}. The accuracy score of the Random Forest algorithm is 93.3% with the best parameters {‘criterion’: ‘entropy’, ‘max_depth’: 4, ‘max_features’: None, ‘n_estimators’: 200}. The accuracy score of Adaboost with the best parameters {‘learning_rate’: 3, ‘n_estimators’: 400} is 82.71%, the accuracy score of XGboost with the best parameters {‘learning_rate’: 1, ‘max_depth’: 2, ‘n_estimators’: 300} is 93.82%. The accuracy score of Support Vector Machines, which classifies

the data points marked on the plane by leaving them at the maximum distance for both classes with a line to be drawn, with the best parameters {‘C’: 100, kernel=‘linear’} is 96.02%. The accuracy score of the Artificial Neural Networks (ANN) classifier is 93.7% with the best parameters {‘activation’: ‘tanh’, ‘hidden_layer_sizes’: (4,), ‘learning_rate’: ‘constant’, ‘solver’: ‘lbfgs’}. The accuracy score of the Logistic Regression is 95.78% with the best parameters {‘C’: 100, ‘class_weight’: None, ‘max_iter’: 150, ‘penalty’: ‘l1’, ‘solver’: ‘saga’}. The results obtained can be summarized in Figure 2.

Model success was measured using various algorithms on the dataset, with ‘accuracy’ selected as the model performance criterion. Among all models, the highest performance was achieved with Support Vector Machines (SVM),

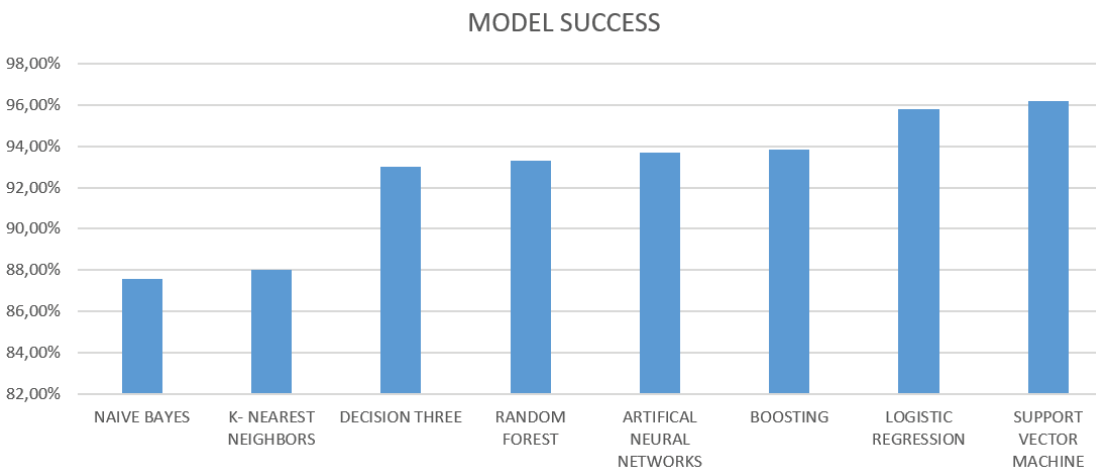


Figure 2. Model accuracy score according to classifiers.

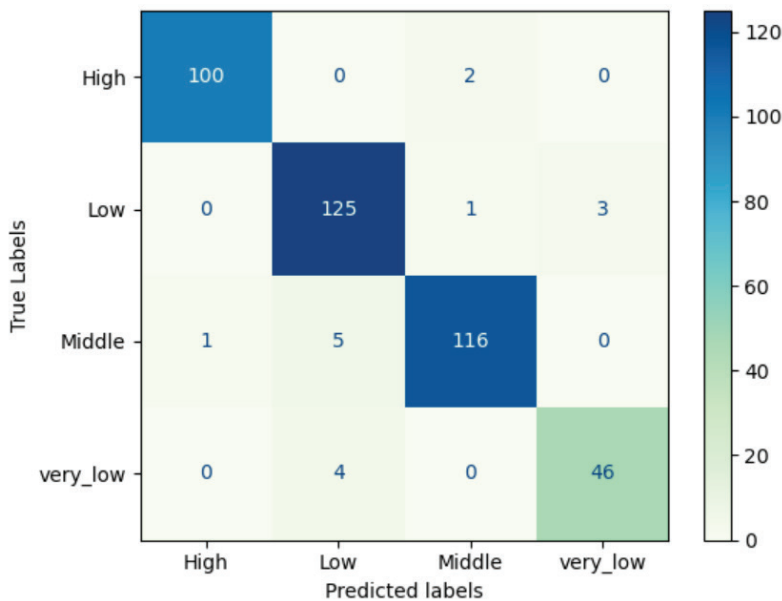


Figure 3. Support vector machine model confusion matrix.

as shown in Figure 2. Correct detection was made except for all but 11 out of a total of 403. The confusion matrix is shown in Figure 3.

Additionally, after stratified 30-fold cross validation (to increase the statistical power of the Wilcoxon test by obtaining by observations) the Wilcoxon test results comparing the sequential performance of the algorithms trained at each layer are presented as a Wilcoxon heat map in Figure 4. To determine the statistical significance of differences between two independent paired samples, p-values less than 0.05 and 0.01 were marked with * and ** respectively in Figure 4. These markings indicate the extent of difference between the performances of the two algorithms [50,51]. Since Naive Bayes and KNN perform worse than others, and while logistic regression and SVM perform better than others, they have more markings than others.

In order to provide a more balanced evaluation, the weighted classification metrics-computed using cross validation results from the previous stage and considering the class distribution- are given in Table 1. The last column of the table represents the accuracy obtained after applying the SMOTE technique [52].

In this regard, the algorithm with the highest model success was Support Vector Machines.

THE EFFECTS OF THE FEATURE SELECTION AND FEATURE EXTRACTION ON MODEL ACCURACY REDUCING THE NUMBER OF FEATURES

The random forest algorithm was applied identify the features that have little effect on the model. The importance ranking of the features were obtained using

the random forest algorithm as the following form in Figure 5. Working with a dataset that includes variables of different importance allows for elaboration during the analyses.

In feature selection process, the number of features was reduced based on importance ranking obtained from the random forest algorithm, Using the most important three features the Support vector machines (SVM) model with the optimized parameters $\{C:17, \text{kernel}='linear'\}$ achieved 95.2% accuracy, when two most important features were used the SVM model with the optimized parameters $\{C:5, \text{kernel}='rbf'\}$ achieved 95%. So the model is visualized on the plane in Figure 6.

In the feature extraction process, when the number of features was reduced to three by using LDA, the accuracy score of the SVM model with the parameters $\{C:17, \text{kernel}='poly'\}$ was 96.0%, When the same operations were repeated using SVM with three and two features after applying NCA, the model accuracy scores with $\{C:3, \text{kernel}='rbf'\}$ were found to be 97.7%. After applying Principal Component Analysis (PCA), SVM model with three features and the parameters $\{C:1000, \text{kernel}='linear'\}$ achieved 78.1% accuracy; SVM model with two features and the parameters $\{C:1000, \text{kernel}='rbf'\}$ achieved 74.9% accuracy. After applying NMF, when the processes were repeated using SVM with three features and the parameters $\{C:4, \text{kernel}='poly'\}$, the accuracy was 72.7% , and while the model for two features with the parameters $\{C:3, \text{kernel}='rbf'\}$ achieved 71.4% accuracy.

Reducing the Number of Classes

When the number of classes was reduced to two classes such as lower middle and upper middle, the SVM

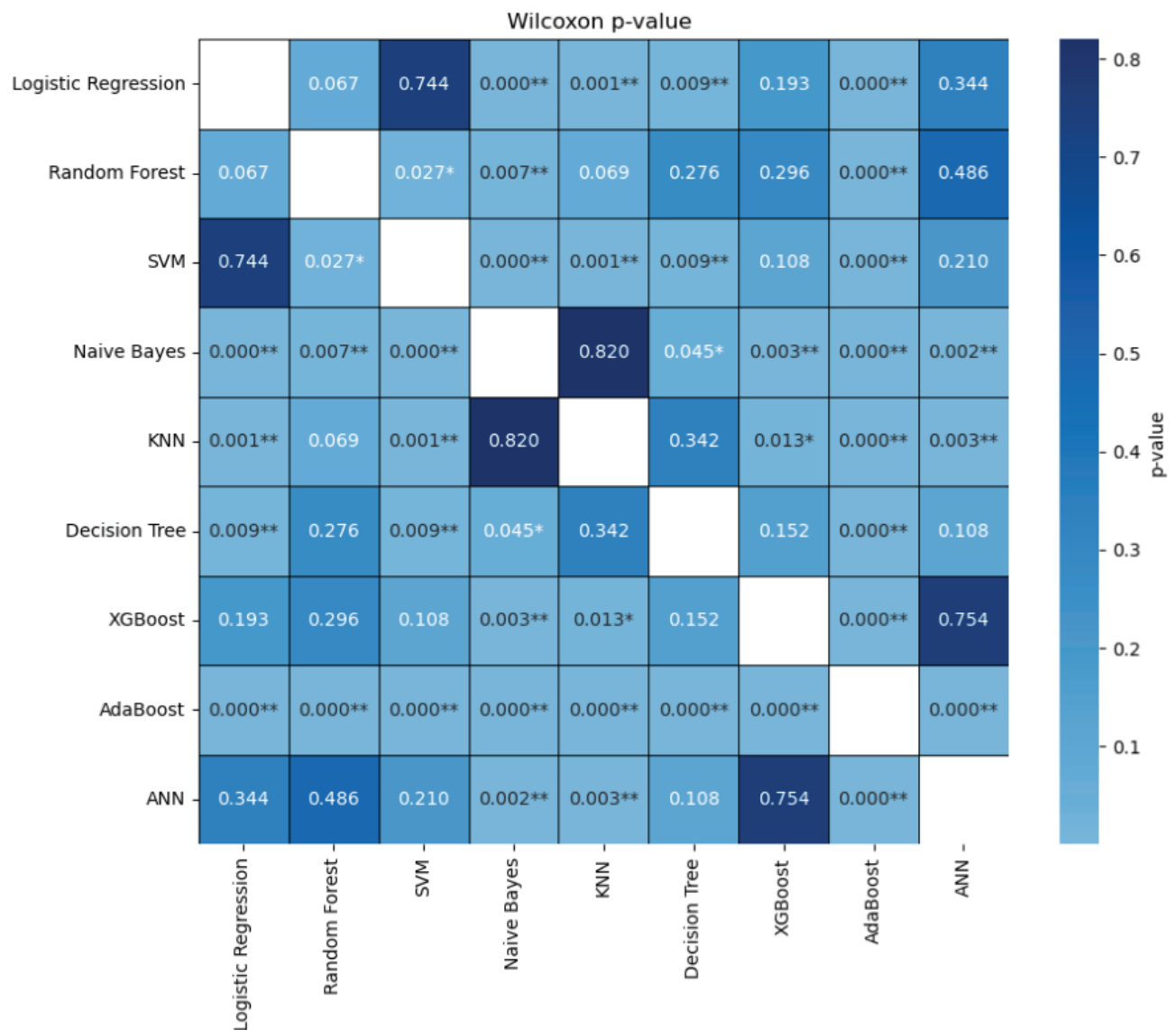


Figure 4. Wilcoxon p-value heatmap.

Table 1. Weighted classification metrics and model accuracy after applying SMOTE

Model	Weighted precision	Weighted recall	Weighted f1_score	Weighted accuracy	SMOTE accuracy
Logistic regression	0.9700	0.9630	0.9612	0.9584	0.9655
Random forest	0.9428	0.9377	0.9322	0.9213	0.9402
SVM	0.9726	0.9657	0.9646	0.9668	0.9582
Naive bayes	0.9176	0.8985	0.8943	0.9029	0.8886
KNN	0.9131	0.9076	0.9000	0.885	0.8952
Decision tree	0.9339	0.9258	0.9207	0.9118	0.9133
XGBoost	0.9498	0.9476	0.9424	0.9376	0.9479
AdaBoost	0.2796	0.4739	0.3391	0.4952	0.5406
ANN	0.9629	0.9527	0.9518	0.9559	0.9529

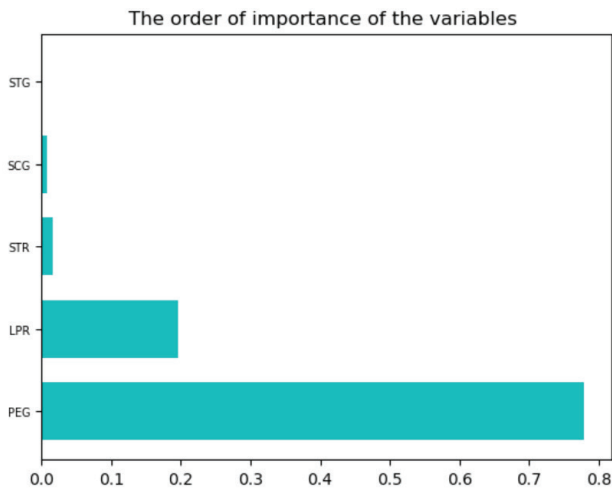


Figure 5. The order of importance of the variables determined by the random forest algorithm.

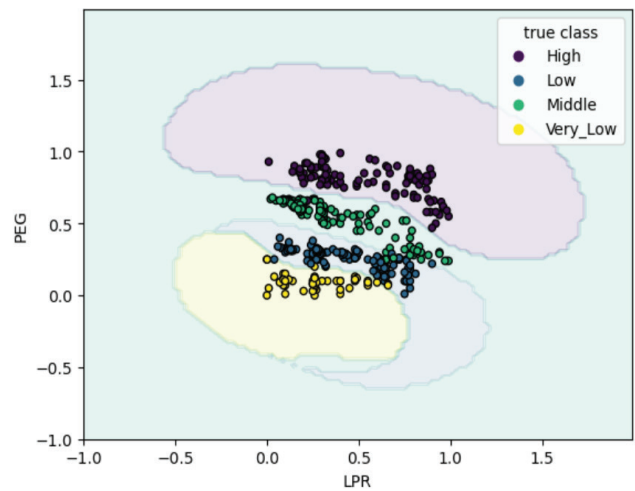


Figure 6. Visualization of the four-classes model on the data plane using two features.

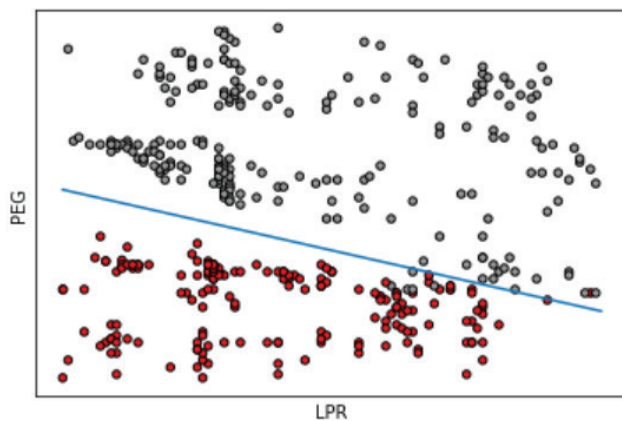


Figure 7. Visualization of the two-classes model on the data plane using SVM with a linear kernel.

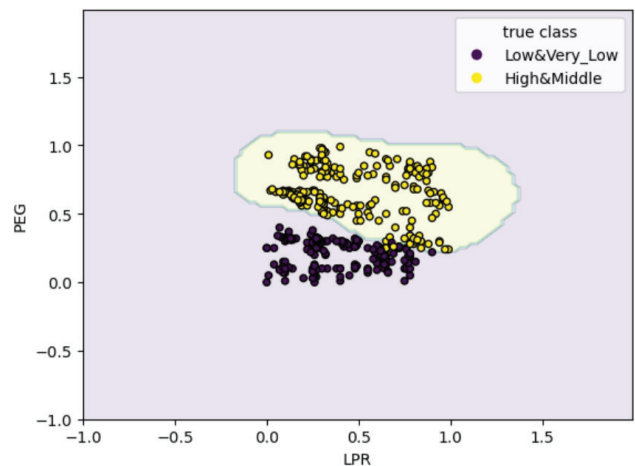


Figure 8. Visualization of the two-classes model on the data plane using SVM with an RBF kernel.

model with five features and the parameters {‘C’: 1000, kernel=‘linear’} yielded 98.7% accuracy. Moreover the number of features reduced by considering the order of importance in the random forest algorithm, the model with three features and the parameters {‘C’: 17, kernel=‘linear’} yielded an accuracy of 98.5%; while the model with two features and the parameters {‘C’: 4, kernel=‘linear’} achieved an accuracy of 98.5%. When the number of classes was reduced to two classes such as lower middle and upper middle and the number of features was reduced, model accuracy with {‘C’: 0.1, kernel=‘rbf’} for two classes and three features was 98.5%; for two classes and two features with {‘C’: 1000, kernel=‘rbf’} was 91.3%. Reducing the number of features to two means dividing the data into two regions on the plane, both linearly as seen in Figure 7 and curvilinearly as seen in Figure 8.

Feature Construction on Fuzzification

This section aims to develop a flexible modeling approach that uses statistical quantities of feature distributions, as opposed to classical fuzzy approaches. Similar data-driven approaches have been discussed in the literature under the term adaptive fuzzy partitioning, and it has been observed that they improve performance by better representing uncertainty in the data [53].

$X_{m \times n}$ denotes the dataset consisting of m observation with n features (here $m = 403, n = 5$). $Y_{m \times 1}$ is a matrix consisting of labels. We denote the labels as a matrix Y with one column. So the i_{th} row of $X_{m \times n}$ represents the i_{th} unlabeled observation of the dataset as $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$. When the fuzzification process is applied on each p_{th} columns of the dataset: $x_p = (x_{1p}, x_{2p}, \dots, x_{mp})$ which represents p_{th} feature, we get two new features such as membership degree

X_p Fuzzy and fuzzy class number X_p FClass for $p = 1, 2, \dots, n$. Here, the fuzzification process given by the following steps

For each p_{th} column of the dataset $X_p = (x_{1p}, x_{2p}, \dots, x_{mp})$ for $p = 1, 2, \dots, n$.

Step 1. decide the boundaries of the fuzzy sets such as $[\beta]_{1p}, [\beta]_{2p}, [\beta]_{3p}$ according to the statistical quantities of the feature. ($[\beta]_{1p}$ is standart deviation of the variable X_p ; $[\beta]_{2p}$ and $[\beta]_{3p}$ denote the mean value of the variable X_p)

Step 2. generate the fuzzy sets using following formulas for $i = 1, 2, \dots, m$

$$\mu_{1p} = \begin{cases} \frac{x_{ip}}{-[\beta]_{1p}} + 1, & 0 < x_{ip} \leq [\beta]_{1p} \\ 0, & [\beta]_{1p} < x_{ip} \leq 1 \end{cases} \quad (1)$$

$$\mu_{2p} = \begin{cases} \frac{x_{ip}}{[\beta]_{2p}}, & 0 < x_{ip} \leq [\beta]_{2p} \\ \frac{x_{ip}-1}{[\beta]_{2p}-1}, & [\beta]_{2p} < x_{ip} \leq 1 \end{cases} \quad (2)$$

$$\mu_{3p} = \begin{cases} 0, & 0 < x_{ip} \leq [\beta]_{3p} \\ \frac{x_{ip}-[\beta]_{3p}}{1-[\beta]_{3p}}, & [\beta]_{3p} < x_{ip} \leq 1 \end{cases} \quad (3)$$

Step 3. Calculate

$$X_p Fuzzy = \max_{t \in \{1,2,3\}}(\mu_{tp}) \quad (4)$$

and

$$X_p FClass = \arg_{i \in \{1,2,3\}}(\mu_{ip}) \quad (5)$$

Step 4. add two columns to the dataset $X_{m \times n}$ matrix (X_p Fuzzy to the dataset matrix as an entry $x_{i(n+p)}$; X_p FClass to the dataset matrix as an entry $x_{i(n+p+1)}$ in the i_{th} row, $(n+p)_{th}$; $(n+p+1)_{th}$ columns of the data matrix, $i = 1, 2, \dots, m$)

Step 5. apply steps 1 to 4 for every p value and obtain the new data matrix $X_{m \times (3n)}$ (note: if it is necessary, the steps may be applied for only one feature column and obtained the new data matrix $X_{m \times (n+2)}$ or the steps may be applied for only k feature columns so the new data matrix $X_{m \times (n+2k)}$ will be obtained)

Step 6. build the machine learning model with the new data matrix and label matrix Y

From Step 2, when membership functions of the five features are calculated separately, for each feature Fuzzy sets are obtained as Figure 9-13. Here, the lines named low, medium and height were obtained with $\mu_{1p}, \mu_{2p}, \mu_{3p}$ formulas defined in step 2.

The steps are applied on each feature, we obtain following results: When the fuzzification is applied on STG, the model accuracy score with {'C': 100, kernel='linear'} is 94.8%. If STG is dropped from the new dataset, we get the accuracy score as 95.0% with {'C': 100, kernel='linear'}. When the fuzzification is applied on SCG, the model accuracy score with {'C': 17, kernel='linear'} is 95.55%. If SCG is dropped from the new dataset, we get the accuracy score as 93.81% with {'C': 150, kernel='linear'}. When the fuzzification is applied on STR, the model accuracy score with {'C': 100, kernel='linear'} is 94.79%. If STR is dropped from the

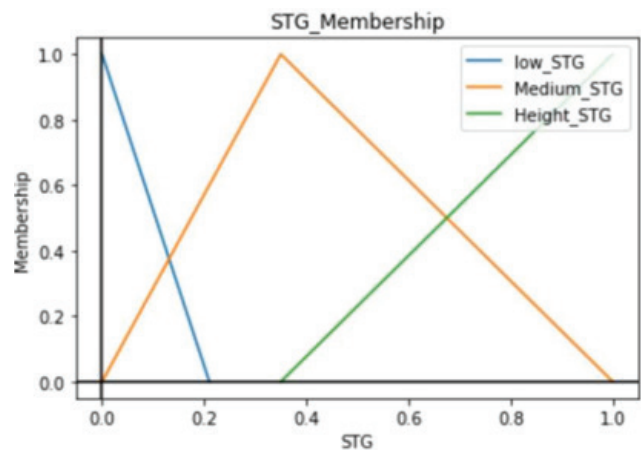


Figure 9. Fuzzy sets and Membership degrees of the feature STG.

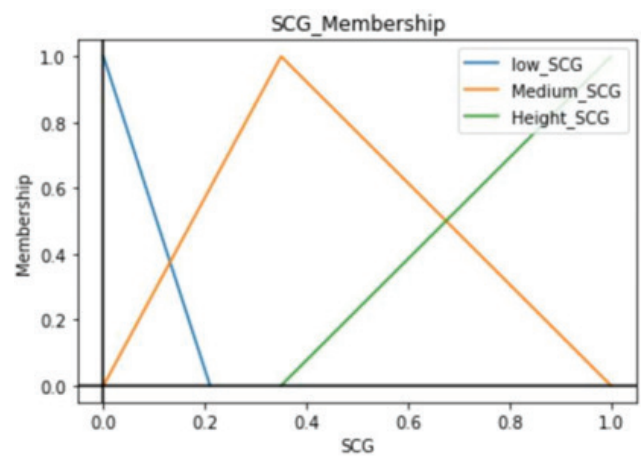


Figure 10. Fuzzy sets and Membership degrees of the feature SCG.

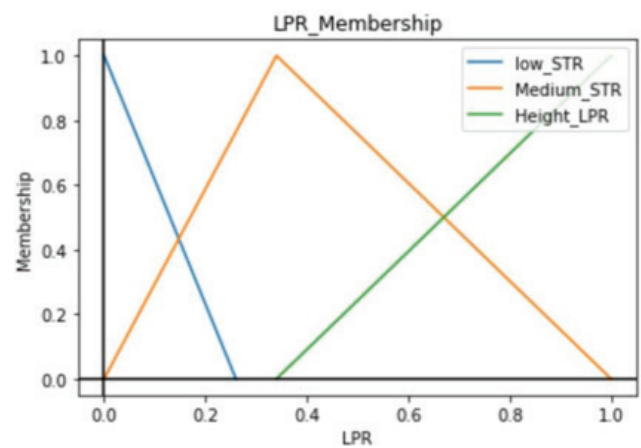


Figure 11. Fuzzy sets and Membership degrees of the feature LPR.

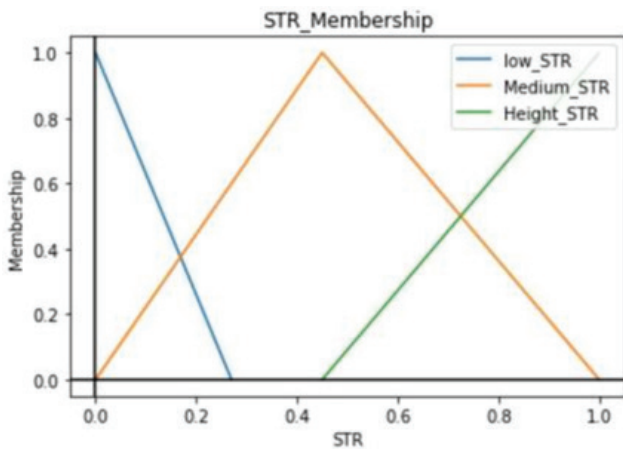


Figure 12. Fuzzy sets and Membership degrees of the feature STR.

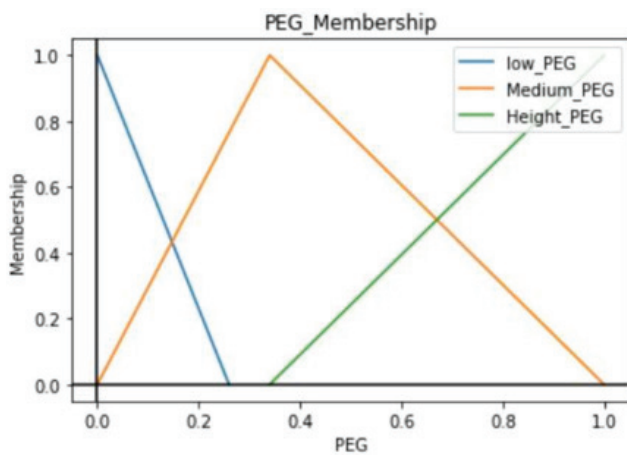


Figure 13. Fuzzy sets and Membership degrees of the feature PEG.

new dataset, we get the accuracy score as 95.05% with $\{C: 1000, \text{kernel}=\text{'linear'}\}$. When the fuzzification is applied on LPR, the model accuracy score with $\{C: 4, \text{kernel}=\text{'linear'}\}$ is 95.05%. If LPR is dropped from the new dataset, we get the accuracy score as 95.05% with $\{C: 5, \text{kernel}=\text{'linear'}\}$. When the fuzzification is applied on PEG, the model accuracy score with $\{C: 17, \text{kernel}=\text{'rbf'}\}$ is 96.78%. If PEG is dropped from the new dataset, we get the accuracy score as 85.60% with $\{C: 17, \text{kernel}=\text{'rbf'}\}$. When the fuzzification is applied on all features, adding fuzzy set labels and membership degrees to the dataset at the same time, The model accuracy score with $\{C: 3, \text{kernel}=\text{'linear'}\}$ is 95.05%.

It was taken into consideration that the proposed approach, which aims to reduce uncertainty in the original feature space, could operate compatibly with unsupervised dimensionality reduction methods such as PCA, as PCA reduces inter-feature correlations and projects the data into a lower-dimensional space. By applying this approach, uncertainty in feature distributions is reduced, thereby enabling PCA to extract more informative components. Accordingly, the fuzzification process was applied prior to PCA. While the data is reduced to two dimensions using PCA, Table 2 shows the performance of all algorithms observed when the technique is applied or not applied before PCA.

To investigate the effects of applying fuzzy transformation before PCA on model performance, a data leakage-preventing processing algorithm, as shown in Figure 14, was implemented. All preprocessing steps were performed strictly within the inner loop of a stratified nested cross-validation framework to avoid any optimistic bias. The impact of the fuzzification module on classification performance was evaluated under two different conditions, with the module enabled and disabled, and visualized using a box plot. The results of evaluating the obtained F1 scores using a paired t-test, Hodge-Lehmann estimation, and Cohen’s d test are given in Table 3.

Table 2. The effect of applying to the proposed technique to the data before reducing it to two dimensions applying PCA on the model’s performance

Method	Accuracy 4D → 2D via PCA	Accuracy 4D → 2D via FUZY+PCA	Accuracy 3D → 2D via PCA	Accuracy 3D → 2D via FUZY+PCA
Logistic Regression	0.6604	0.7117	0.5974	0.7558
Random Forest	0.6476	0.7217	0.6580	0.8357
SVM	0.6604	0.7190	0.6075	0.7686
Naive Bayes	0.6622	0.6879	0.5902	0.7340
KNN	0.6503	0.7441	0.6113	0.8653
Decision Tree	0.6069	0.6666	0.6053	0.7641
XGBoost	0.5844	0.6582	0.5639	0.8276
AdaBoost	0.3849	0.3761	0.3895	0.3917
ANN	0.7049	0.7413	0.6282	0.7941

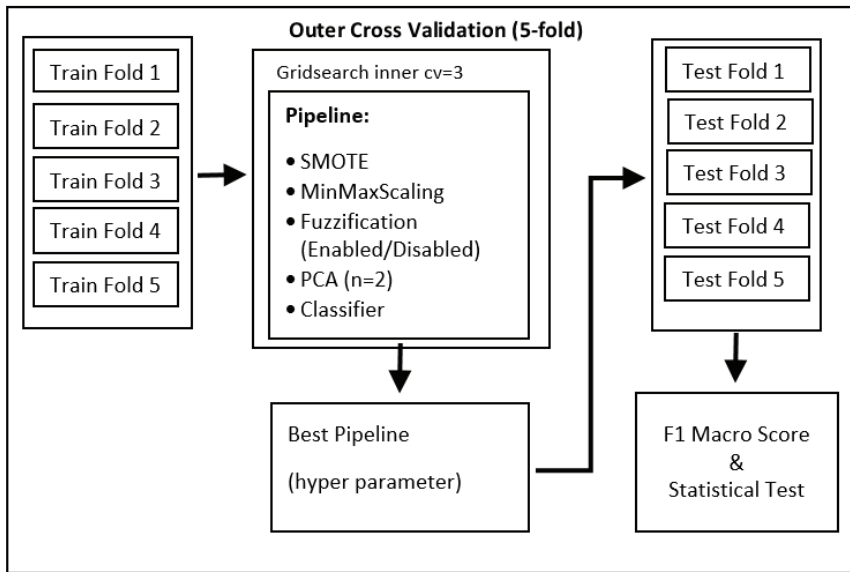


Figure 14. Proposed nested pipeline, showing how fuzzy preprocessing and PCA ordering are applied within inner and outer loops.

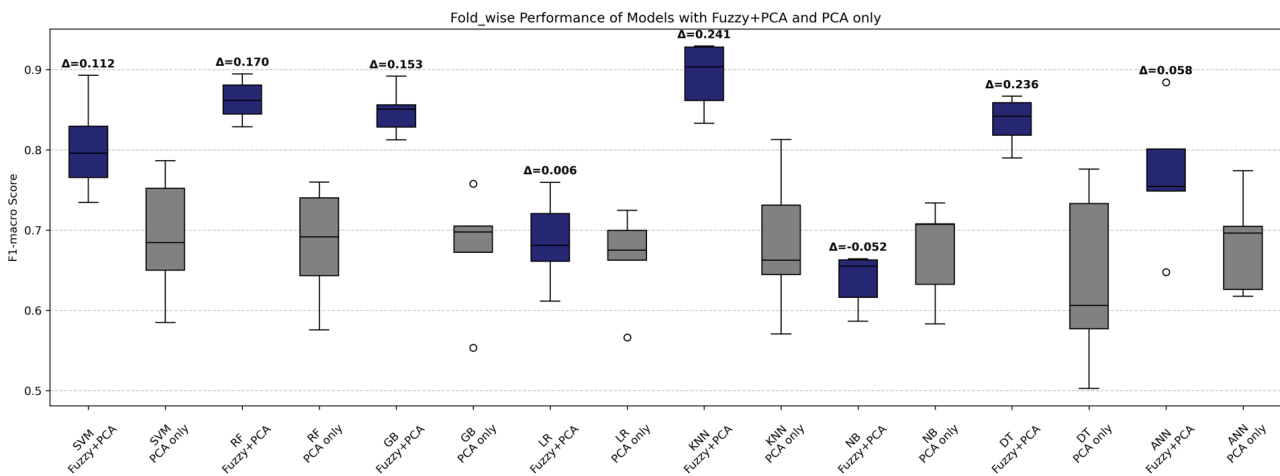


Figure 15. Effect of the fuzzification module on F1-score performance, illustrated using box plot.

Table 3. Statistical comparison of F1 scores with the fuzzification module enabled and disabled using a paired t-test, Hodges-Lehmann estimation, and Cohen’s d

Model	Paired t _p	Hodges-Lehmann	Cohens d
SVM	0.021	0.112	1.658
Random forest	0.002	0.170	3.423
XGBoost	0.005	0.153	2.525
Logistic Regression	0.643	0.006	0.224
KNN	0.003	0.241	2.972
Naive bayes	0.423	-0.052	-0.398
DT	0.037	0.236	1.374
ANN	0.054	0.058	1.205

As shown in Figure 15, in our experiments, when the fuzzification module was applied prior to PCA, an increase in F1-scores was observed in all classifiers except NB. The consistency of these results was further confirmed by the paired t-test, Hodges-Lehmann estimation, and Cohen's analysis [54,55]. The outcomes are summarized in Table 3.

Table 3 shows that the Hodges-Lehmann median differences are positive, and Cohen's d values for the SVM, Random Forest, XGBoost, KNN and Decision Tree models, indicating that the fuzzification module significantly impacts the F1 scores. In contrast, no statistically significant effect is observed in the Logistic Regression and Naïve Bayes models.

RESULTS AND DISCUSSION

In this study, which introduces a novel fuzzification-based feature construction approach for ML models, the effects of feature reduction on the model accuracy score were examined using supervised and unsupervised methods. When the results are compared, it can be seen that reducing the number of features using supervised feature extraction methods has a greater positive impact on the model's accuracy score than reductions based on the importance ranking determined by random forest algorithm, and unsupervised feature extraction methods have negative effects on the model accuracy score, as they do not use class labels. However the proposed unsupervised method has positive effects on it. When the membership degrees and the set numbers are added as two new features to the dataset by creating the fuzzy sets based on the most important feature according to the random forest algorithm, the model yields a better accuracy score. It has been observed that unsupervised new feature construction approach based on fuzzification has a positive effect on the accuracy score even though unsupervised feature extraction methods have negative effects on the model accuracy score. Moreover, when the effect of applying the proposed technique to the data before dimensionality reduction was investigated, among all dimensionality reduction techniques a noticeable improvement in the performance of all algorithms was observed specifically when the data was reduced to two dimensions applying PCA. Applying fuzzification did not have a significant impact on training time or memory usage. Using a dataset of 403 samples, proposed method has shown its effectiveness, though the small dataset limits generalization, it can be generalized to other dataset. This results suggest that the applied technique can be recommended as a pre-processing technique and can be applied in areas such as image processing.

CONCLUSION

This study investigates the effects of reducing the number of features or classes and generating new features on model performance. Parameters that maximize model success were investigated for models created using various

machine learning algorithms, including k-nearest neighbors, Naïve Bayes, support vector machines, random forest, boosting methods, decision trees, and artificial neural networks. The Wilcoxon test was applied to compare algorithm performances within the same layers and identify statistically significant differences. In addition, a weighted classification report was used to ensure a balanced evaluation, and model accuracy values were obtained using SMOTE.

The impact of both supervised and unsupervised dimensionality reduction techniques on model performance was analyzed. Furthermore, a novel fuzzification-based feature generation approach was proposed, which computes the number of fuzzy clusters, and membership degrees for each feature and incorporates them into the dataset. Experimental results demonstrate that applying the proposed method prior to dimensionality reduction, particularly before reducing data to two dimensions using PCA, generally leads to improvements in the performance of most evaluated algorithms.

REFERENCES

- [1] Alpaydin E. Yapay öğrenme. 2. baskı. İstanbul: Boğaziçi Üniversitesi Yayınları; 2011.
- [2] Vouk B, Guid M, Robnik Sikonja M. Feature construction using explanations of individual predictions. *Eng Appl Artif Intell* 2023;120:105823:1–27. [\[CrossRef\]](#)
- [3] Hummel R, Manevitz LM. Combination calculi for uncertainty reasoning: representing uncertainty using distribution. *Ann Math Artif Intell* 1997;20:69–109. [\[CrossRef\]](#)
- [4] Zadeh LA. Fuzzy sets. *Inf Control* 1965;8:338–353. [\[CrossRef\]](#)
- [5] Zimmermann HJ. Fuzzy set theory and its application. Berlin/Heidelberg, Germany: Springer Science & Business Media; 2011.
- [6] Zimmermann HJ. Fuzzy set theory. *Wiley Interdiscip Rev Comput Stat* 2010;2:317–332. [\[CrossRef\]](#)
- [7] Shokori B, Shureshjoni RA, et al. A parametric method for ranking intuitionistic fuzzy numbers and its application to solve intuitionistic fuzzy network data envelopment analysis models. *Complexity* 2020;2020:6408613. [\[CrossRef\]](#)
- [8] Celikyilmaz A, Turksen IB. Modelling uncertainty with fuzzy logic. In Kacprzyk J, (editor). *Studies in Fuzziness and Soft Computing*. Newyork: Springer Nature; 2009. p. 1–10. [\[CrossRef\]](#)
- [9] Han J, Liu G. A novel approach based on fuzzy rough sets for web query system. In: Cao BY, et al., editors. *Fuzzy Information and Engineering*. Newyork: Springer Nature; 2010. p. 667–673. [\[CrossRef\]](#)
- [10] Biswas N, Chakraborty C, Mullick SS, Das S. A parameter independent fuzzy weighted k nearest neighbor classifier. *Pattern Recogn Lett* 2018;101:80–87. [\[CrossRef\]](#)

- [11] Nowicki RK, Starczewski JT. A new method for classification of imprecise data using fuzzy rough fuzzification. *Inf Sci* 2017;414:33–52. [\[CrossRef\]](#)
- [12] Martino FD, Sessa S. A classification algorithm based on multidimensional fuzzy transforms. *J Ambient Intell Humanized Comput* 2022;13:2873–2885. [\[CrossRef\]](#)
- [13] Hsu HM, Chen CT. Aggregation of fuzzy opinions under group decision makers. *Fuzzy Sets Syst* 1996;79:279–285. [\[CrossRef\]](#)
- [14] Herrera F, Martinez LA. 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Trans Fuzzy Syst* 2000;8:746–752. [\[CrossRef\]](#)
- [15] Anil PN, Patil PG. Multi-parameter fuzzy soft set approach to decision making problems. *Afr J Math Comput Sci Res* 2018;11:61–71. [\[CrossRef\]](#)
- [16] Brunelli M, Mezei J. How different are ranking methods for fuzzy numbers: A numerical study. *Int J Approx Reason* 2013;54:627–639. [\[CrossRef\]](#)
- [17] Kanbay F, Vardar N. On a fuzzy application of the particulate matters estimation. *Iran J Fuzzy Syst* 2018;15:133–143.
- [18] Belman RE, Zadeh LA. Decision making in a fuzzy environment. *Manag Sci* 1970;17:141–164. [\[CrossRef\]](#)
- [19] Bezdek JC. *Pattern recognition with fuzzy function algorithms*. New York: Plenum Press; 1981. [\[CrossRef\]](#)
- [20] Chen SJ, Hwang CL. *Fuzzy Multiple Attribute Decision Making Methods*. In: Beckmann M, Künzi HP, Fandel G, Trockel W, Dawid H, et al. (editors). *Fuzzy Multiple Attribute Decision Making. Lecture Notes in Economics and Mathematical Systems*. Heidelberg, Baden-Württemberg, Germany: Springer-Verlag; 1992. p. 289–486. [\[CrossRef\]](#)
- [21] Roy AR, Maji PK. A fuzzy soft set theoretic approach to decision making problems. *J Comput Appl Math* 2007;203:412–418. [\[CrossRef\]](#)
- [22] Zadeh LA. The concept of linguistic variable and its application to approximate reasoning I. *Inf Sci* 1975;8:199–249. [\[CrossRef\]](#)
- [23] Yager RR. Database discovery using fuzzy sets. *Int J Intell Syst* 1996;11:691–712. [\[CrossRef\]](#)
- [24] Martinez L, Rodriguez RM, Herrera F. The 2-tuple linguistic model, computing with words in decision making. In *The 2-tuple Linguistic Model*. New York: Springer Nature; 2015. p. 1–10. [\[CrossRef\]](#)
- [25] Ezghari S, Zahi A, Zenkouar Z. A new nearest neighbor classification method based on fuzzy set theory and aggregation operators. *Expert Syst Appl* 2017;80:58–74. [\[CrossRef\]](#)
- [26] Shang W, Huang H, Zhu H, Lin Y, Wang Z, Qu Y. An improved kNN algorithm fuzzy kNN. In: *CIS Part I, LNAI 3801*. Heidelberg, Baden-Württemberg, Germany: Springer-Verlag; 2005. p. 741–746. [\[CrossRef\]](#)
- [27] Yuan Y, Shaw MJ. Introduction of fuzzy decision trees. *Fuzzy Sets Syst* 1995;69:125–139. [\[CrossRef\]](#)
- [28] Bonissone P, Cadenas JM, et al. A fuzzy random forest. *Int J Approx Reason* 2010;51:729–747. [\[CrossRef\]](#)
- [29] Lin CF, De Wang S. Fuzzy support vector machines. *IEEE Trans Neural Netw* 2002;13:464–471. [\[CrossRef\]](#)
- [30] Wang XZ, Xing HJ, Li Y, Hua Q, Dong CR, Pedrycz W. A study on relationship between generalization abilities and fuzziness base classifiers in ensemble learning. *IEEE Trans Fuzzy Syst* 2014;23:1638–1654. [\[CrossRef\]](#)
- [31] Kuncheva LI. *Fuzzy classifier design*. Vol. 49. Berlin/Heidelberg, Germany: Springer Science & Business Media; 2000. [\[CrossRef\]](#)
- [32] Yang YL, Bai XY. A research on classification performance of fuzzy classifiers based on fuzzy set theory. *Iran J Fuzzy Syst* 2019;16:15–27.
- [33] Yang MS, Chen CT. On strong consistency of the fuzzy generalized nearest neighbor rule. *Fuzzy Sets Syst* 1993;60:273–281. [\[CrossRef\]](#)
- [34] Chen HL, Liu DY, Yang B, Wang SJ. An adaptive fuzzy k-nearest neighbor method based on parallel particle swarm optimization for bankruptcy prediction. In: *Lecture Notes in Computer Science 6634 LNAI (Part 1)*. Berlin/Heidelberg, Germany: Springer-Verlag; 2011. p. 249–264. [\[CrossRef\]](#)
- [35] Li DC, Liu CW, Hu SC. A fuzzy-based data transformation for feature extraction to increase classification performance with small medical datasets. *Artif Intell Med* 2011;52:45–52. [\[CrossRef\]](#)
- [36] Theng D, Bhojar KK. Feature selection techniques for machine learning: A survey of more than two decades of research. *Knowl Inf Syst* 2024;66:1575–1637. [\[CrossRef\]](#)
- [37] Wen H, Zhao S, Liang M. Unsupervised attribute reduction algorithm for mixed data based on fuzzy optimal approximation set. *Mathematics* 2023;11:3452. [\[CrossRef\]](#)
- [38] Gu X. Self-adaptive fuzzy learning ensemble systems with dimensionality compression from data streams. *Inf Sci* 2023;634:382–399. [\[CrossRef\]](#)
- [39] Ismail M, Shang C, Yang J, et al. Supporting ANFIS interpolation for image super resolution with fuzzy rough feature selection. *Appl Intell* 2024;54:5373–5388. [\[CrossRef\]](#)
- [40] Ma X, Hu H, Zhang Q, Xu Y. Feature selection for hybrid information systems based on β fuzzy covering and fuzzy evidence theory. *Complex Intell Syst* 2024;10:7531–7552. [\[CrossRef\]](#)
- [41] Mohammed SJ, Abbas AK, Ahmad AA, Mohammed S, Sarı M, Uslu TH. Determining the parameters of data mining techniques and their effects on iron deficiency based prediction. *Sigma J Eng Nat Sci* 2025;43:655–664. [\[CrossRef\]](#)
- [42] Akkiraz C, Türkmen Hİ, Güvensan MA, Can B. Battery-friendly tiny models for activity recognition on energy-constrained devices. *Sigma J Eng Nat Sci* 2023;41:716–728. [\[CrossRef\]](#)

- [43] Yiğit ÖE, Öz E. Feature extraction for DNA capillary electrophoresis signals based on discrete wavelet transform combined with multi-scale permutation entropy. *Sigma J Eng Nat Sci* 2022;40:475–490. [\[CrossRef\]](#)
- [44] Jake V. Python data science handbook: Essential tools for working with data. Sebastopol, California, USA: O'Reilly Media; 2017.
- [45] Albon C. Machine learning with Python cookbook: Practical solutions from preprocessing to deep learning. 1st ed. Sebastopol, California, USA: O'Reilly Media; 2018.
- [46] Kahraman H, Colak I, Sagiroglu S. User knowledge modeling [dataset]. Uttar Pradesh, India: UCI Machine Learning Repository; 2009.
- [47] Kahraman HT. Designing and application of web-based adaptive intelligent education system (Unpublished doctoral thesis). Ankara: Gazi University; 2009.
- [48] Kahraman HT, Sagiroglu S, Colak I. The development of intuitive knowledge classifier and the modeling of domain dependent data. *Knowl-Based Syst Man Cybern* 2013;37:283–295. [\[CrossRef\]](#)
- [49] Sarker IH. Machine learning: Algorithms, real-world applications, and research directions. *Sn Comput Sci* 2021;2:160. [\[CrossRef\]](#)
- [50] Dao PB. On Wilcoxon rank sum test for condition monitoring and fault detection of wind turbines. *Appl Energy* 2022;318:1–15. [\[CrossRef\]](#)
- [51] Bagkavos D, Patil PN. Improving the Wilcoxon signed rank test by a kernel smooth probability integral transformation. *Stat Probab Lett* 2021;171:1–10. [\[CrossRef\]](#)
- [52] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. *J Artif Intell Res* 2002;16:321–357. [\[CrossRef\]](#)
- [53] Ruiz E, Casillas J. Adaptive fuzzy partitions for evolving association rules in big data stream. *Int J Approx Reason* 2018;93:463–486. [\[CrossRef\]](#)
- [54] Hodges Jr. L, Lehmann EL. Estimates of location based on rank tests. *Ann Math Stat* 1963;34:598–611. [\[CrossRef\]](#)
- [55] Cohen J. Statistical power analysis for the behavioral sciences. New York: Lawrence Erlbaum Associates; 1988.