Baghdad Science Journal

Volume 23 | Issue 4 Article 15

4-23-2026

An Informative Analysis of Applying Feature Reduction Methods to Supervised Machine Learning Algorithms

Mustafa S. Abd
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq,
mustafa.abd@sc.uobaghdad.edu.iq

How to Cite this Article

Abd, Mustafa S. (2026) "An Informative Analysis of Applying Feature Reduction Methods to Supervised Machine Learning Algorithms," *Baghdad Science Journal*: Vol. 23: Iss. 4, Article 15.
DOI: https://doi.org/10.21123/2411-7986.5271

BAGHDAD SCIENCE JOURNAL 2026;23(4):1333–1351

Scan the QR to view the full-text article on the journal website

RESEARCH ARTICLE

An Informative Analysis of Applying Feature Reduction Methods to Supervised Machine Learning Algorithms

Mustafa S. Abd

Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

ABSTRACT

Feature reduction techniques are fundamental to enhancing machine learning (ML) algorithms by reducing the number of features in a dataset. The study here explores the impact of Principle Component Analysis (PCA) on ML algorithms within an unbalanced classification framework, in partnership with feature selection techniques like Cluster Variation Attribute Evaluator (CVAE) and Correlation Attribute Evaluator (CAE). In addition, the research introspects a comparison analysis evaluating the effectiveness of several ML methods, including Multilayer Perceptron (MLP), Decision Tree J48, k-Nearest Neighbor (k-NN) and Sequential Minimal Optimization (SMO). The informative analysis of results signifies that the MLP technique with PCA minimized the build time of the model about 50%, in the case of 5 folds' cross validation, whilst accuracy remained at the same levels. In contrast, the J48 technique clarified a weak response to feature reduction techniques, while CVAE had a negative impact on the performance of all models. Furthermore, applying PCA with SMO promoted diagnostic accuracy from 95.56% to 95.82%. The k-NN approach realized an accuracy increase to 92.42% with PCA, up from 91.12%, and CAE notably improved the model's accuracy. As a substantial point, this research employed the Weighted Average of Precision, Recall, and F-Measure to deliver a comprehensive assessment of model performance on an imbalanced dataset. The nominal-type thyroid dataset was utilized as the case study for this research.

Keywords: Correlation attribute evaluator, Clustering variation attribute evaluator, J48, k-NN, MLP, PCA, SMO, Nominal thyroid dataset, Weighted average

Introduction

The reduction of dataset's dimensions is a method for reducing the number of features. It can be applied not only to reduce dimensions for complex data, but it is also used for enhancing the performance of ML. When the algorithm learns a model based on labeled training data in the case of supervised ML, the reduction of dimensions may also have an effective and positive impact on model accuracy as well as model tactility. In supervised ML we have the most significant advantages of reducing dimensions is the potential for preventing overfitting. Overfitting is where the noise in data is learned by the model, not the basic pattern, and does poorly on new and unseen data. Dimensionality reduction can aid the model in concentrating on relevant information and prevent overfitting to noise with fewer features in the dataset.[1] Yet another benefit of reducing the dimension for supervised ML is to make the process of training data faster. The reduced number of features makes the algorithm handle the data in easy way; it fits faster when training the model. This may be very helpful especially when working with huge dataset or complex models that demands a lot of time to complete one iteration. Dimensionality reduction may also increase model interpretability. With fewer features, the model becomes less complex and we can easily understand it – so the results can be discussed the decisions can be taken easily based on the model.

Received 12 June 2025; revised 8 August 2025; accepted 15 September 2025.
Available online 23 April 2026

E-mail address: mustafa.abd@sc.uobaghdad.edu.iq (M. S. Abd).

https://doi.org/10.21123/2411-7986.5271

Therefore, it is potentially critical in such as healthcare field or finance field, where the decisions of ML models that informed after analyzing the results have real-world impact.[2] Furthermore, the generalization ability of the model can be enhanced using dimensionality reduction. Data simplifying makes the model sturdy and minimizes overfitting exposure, and enables better generalization for the obtained Data. This could result in improved forecasting and performance of the ML model as a whole. Yet, dimensionality reduction has a dark side as well. The most difficult part is deciding which features to keep and which to throw away. This procedure may be subjective and may involve some domain expertise or some trial-and-error empirical practice to determine which are the most relevant features to include as variables in the dataset.[3] A second possible problem with dimension reduction is the loss of information. When excluding the features, meaningful signals/features in the data may be discarded and result in the deterioration of the model performance. Trade-off considerations on excessively reducing dimensionality at the cost of losing useful information that model requires to learn effectively.[4] To conclude, dimensionality reduction could substantially benefit supervised learning, through model enhancement, frequency of overfitting reduction, accelerated training, and interpretability gain. In contrast to the limitations and overheads that are tangible, reducing dimensions is an enabling technique which improves the efficiency of ML models and their evaluation factors. Understanding swaps and determining the right approaches can enable researchers and executors to take advantage of reducing the dimension for betterment ML models.

Methodology

In our context and in this part, we will introduce a summary of the dataset which we used here and how we applied some methods to reduce feature space. At the first, the PCA which represents principal component analysis, will be applied as a method for extracting the features, also two other methods are specialized in selecting the features, the Correlation Attribute Evaluator (CAE) and the Clustering Variation Attribute Evaluator (CVAE). After extracting these features as above, we will feed them to four different ML algorithms—J48, MLP, SMO, and k-NN—to investigate and compare the effect of these techniques on the performance of each algorithm. Furthermore, we will use these four algorithms also without feature reduction to compare respective performance measures of the precision factor, factor of recall, f measure, the accuracy and the time of execution. The holistic technique will enable the researcher to determine the effects of reducing the features on ML results efficiently.

Thyroid cancer data set

The data set shown in Table 1, consists of 13 clinicopathological features for recurrence of well-

Table 1. Thyroid data set description.

Variable Name	Role	Type	Demographic	Description	Missing Values
Age	Feature	Integer	Age	15–82	No
Gender	Feature	Categorical	Gender	Female, Male	No
Smoking	Feature	Categorical		Yes, No	No
Hx Smoking	Feature	Categorical		Yes, No	No
Hx Radiothreapy	Feature	Categorical		Yes, No	No
Thyroid Function	Feature	Categorical		Euthyroid, Clinical Hyperthyroidism, Subclinical Hyperthyroidism, Clinical Hypothyroidism, Subclinical Hypothyroidism	No
Physical Examination	Feature	Categorical		Single nodular goiter-left, Single nodular goiter-right, Multinodular goiter, Normal, Diffuse goiter	No
Adenopathy	Feature	Categorical		No, Right, Extensive, Left, Bilateral, Posterior	No
Pathology	Feature	Categorical		Micropapillary, Papillary, Follicular, Hurthel cell	No
Focality	Feature	Categorical		Uni-Focal, Multi-Focal	No
Risk	Feature	Categorical		Low, Intermediate, High	No
T	Feature	Categorical		T1a, T1b, T2, T3a, T3b, T4a, T4b	No
N	Feature	Categorical		N0, N1a, N1b	No
M	Feature	Categorical		M0, M1	No
Stage	Feature	Categorical		I, II, III, IVA, IVB	No
Response	Feature	Categorical		Intermediate, Excellent, Structural Incomplete, Biochemical Incomplete	No
Class = Recurred	Target	Categorical		Yes, No	No

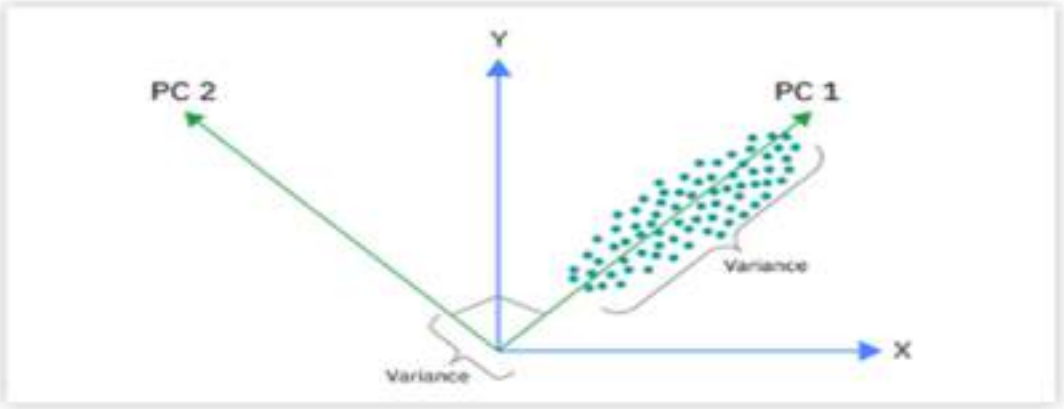

Fig. 1. Principle component analysis.

differentiated thyroid cancer. Amassed over 15 years, each patient in the study was observed for at least 10 years. The present dataset is an outcome of an AI (Artificial Intelligence)/medicine hybrid research project with no external funding and consists of single patient cases. It is composed of 16 features, one class label (Recurred: Yes/No) and has 383 instances.[5,6]

Extracting of features using PCA (Principal-component-analysis)

PCA is an advanced statistic method by which you take a set of variables with known correlations and measure how closely these variables track each other. It is among the most popular techniques used for exploring analyzed data and Machine Learning, especially in prediction model development.[7] PCA is considered as an un-supervised learning-algorithm that aims to find relations between a set of features or variables. Commonly it is known as a type of common factor analysis, PCA detects the best fit line through-regression.[8]. PCA is used mainly for dimensionality reduction of data and also to extract the most informative patterns and relationships between the input variables; however, a priori information of the target variables is unnecessary.[9]. By finding a smaller set of variables, it captures most of the information occurred in the dataset. This dimensionality reduction not only makes the data easier to handle, but also makes it more useful for further regression and classification.[8]. Two principal components are extracted in PCA: PC1, and PC2. (PC1) which is the first-principal-component that represents the orientation in the space of data that explains the points and the variances among them. It is the best fit line that explains the spread of the expected data. More information from the original data set is retained if a larger amount of the variation is absorbed by PC1. Note that no other PC may explain more variation than-PC1. (PC2) which represents the second-principal-component is also derived depending on the method for PC1. PC2 explains the next largest variation in the data and has to be un-correlated with PC1. This ortho-gonality, the ortho-gonality, guarantees that PC2 affords information by being separate from-PC1, with its own unique characteristic information. In terms of the mathematical formulation, this corresponds to zero-correlation between PC1 and PC2. So to illustrate the relationship between PC1-and-PC2, in general, when PCA is done to a dataset, scatter plot is used. In this depiction, the axes of PC1 and PC2 are perpendicular to each other, emphasizing these axes are independent and capture separate variance,[10] Fig. 1 shows the basic idea of PCA.

Techniques of feature selection

It is an imperative step in the preliminary-stage of identification and selection of a subset of important features from the basic original set. This procedure is designed to decrease the dimensionality of the features, by decreasing the feature space according to certain criteria, which improves the efficiency and effectiveness of the following analysis. It is famed that the activity of a model can be optimized, computing time can be eliminated, and overfitting problem can be eliminated through emphasis on relevant features.[11]

Clustering variation attributes evaluator (CVAE)

It will focus on an evaluation of Clustering Variation attributes using ML. It includes examining the importance of each attribute by measuring the extent to which it contributes to the entire Clustering

Variation value depending on the class. Clustering Variation looks for a good subset of attributes in order to improve the classification accuracy of supervised learning techniques in classification problems with a huge number of attributes involved. It first creates a ranking of attributes based on the Variation value, then divide into two groups, last using Verification method to select the best group. For this analysis, the powerful machine learning software Weka (Waikato Environment for Knowledge Analysis) is employed as the platform for the CVAE feature selection approach. For additional information, the cited website[12] provides and allows for the download of the complete package necessary for implementing this feature selection method, which is fully executed using WEKA within the attributes selection window.[12] This approach leverages the Ranker Search Method to efficiently pinpoint and prioritize the most influential attributes that significantly impact the class.[12,13]

Correlation-attribute evaluator (CAE)

One of the established techniques of discovering relevant features from a dataset is correlation analysis, also referred to as Pearson's correlation coefficient in statistics. This method works by calculating the correlation between each attribute and the target variable, instead of each attribute is measuring with the other one, this correlation is done individually for each of the attributes, then you can choose to select only those attributes which have some decent to good correlation with the output variable and discard those attributes that can be removed. On the other hand, we should discard attributes that have low (or near zero) correlation. There are several pieces of software that support this method, such as the powerful data mining Weka software[12,14] that implements a Correlation Attribute Evaluator that is used to select the most relevant features based on correlation between features using the Ranker search method. This not only reduces the complexity and the computational cost of the feature selection, but also improves the quality of the supervised model you built.[15]

Multi-layer-perceptron (MLP)

This is a type of feedforward NN in which each neuron in layer j k connects to each neuron in layer j k + 1 (they are dense). A multi-layer perception is a special class of Artificial-Neural-Network which follows the mammalian brain as model with interconnected neurons. This model consists of a number of interconnected layers of nodes - artificial neurons - that the information passes through easily. The incoming signal of each neuron is processed to provide an output that affects following neurons in the network.[15] The architecture of MLP is often one-input-layer, one or more hidden layers, and one output layer. The neurons in these layers use nonlinear activation functions which allow the network to model and learn more complex data patterns.[16] This ability to represent complex nonlinear relationships is the reason behind the popularity of MLPs in ML, where they can perform well in variety of classification, regression, and pattern recognition.[17] Fig. 2, shows MLP with inner layer, two hidden layer and outer layer.

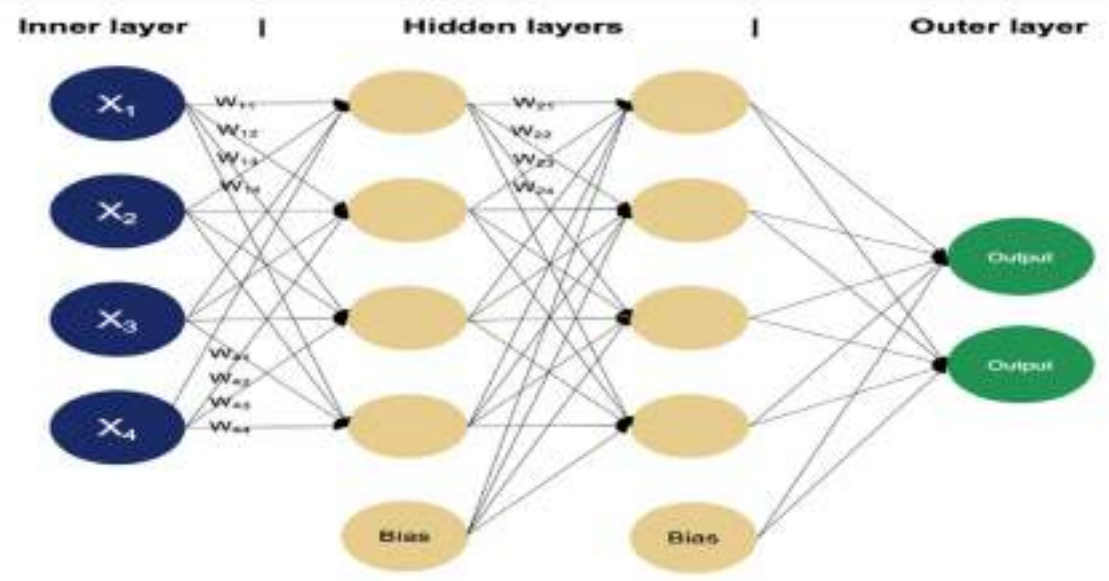

Fig. 2. MLP with two hidden layers.

Decision tree (J48)

The class for decision trees that are not pruned or pruned with the reduced-error pruning heuristic; can be converted to a pruned tree using class Pruneable.C45Wrapper. Decision-tree-J48-algorithm (root-node, branches, internal-nodes and leaf-nodes) is a famous machine learning algorithm for classifying data discretely and continuously. The C4.5 (J48) algorithm is widely applied by various domains for data classification, such as interpretation of clinical data for diagnosis of coronary heart disease, classification of E-governance data, and so on. The-C4.5 is a decision tree induction algorithm that is also a classification algorithm and generates decision trees based on information theory. It is a derivative of Ross Quinlan's earlier C4.5 algorithm, implemented in Weka as J48 (Java). The-C4.5 are subverted for classifying, so, C4.5 is commonly known as a statistical classifier. The-J48 implementation of the C4.5 is equipped with a lot of facilities and features, e.g., coping with missing values, pruning the decision trees, dealing with continuous attribute value scopes, generating rules, etc. In WEKA-machine learning software, J48 is an open-source Java clone of the C4.5. algorithm.[18] J48 enables classification using decision trees or rules produced from them.[17] The node in C4.5 tree selects the data attribute which best separates its set of samples into subsets that enriched in one or the other class. The divide condition is the normalized information gain, derived from the entropy difference. The attribute with the highest normalized information gain is selected to inform the decision. The C4.5. algorithm is than recused on the cut sub lists using a divide-and-conquer technique, and a greedy algorithm-based decision tree is constructed.[16]

Sequential-minimal-optimization (SVM/SMO)

The algorithm was introduced by John Platt in 1998 to solve the "primal" of the quadratic programming problem (though in this context it is also termed the "dual" problem), which is required for the training of Support Vector Machine (SVM) learning algorithms. Quadratic programs are mathematical optimization problems in which the objective function (also the constraints) is a quadratic function of the coordinates of the vector to be optimized or, decision variables.[19] The optimization problem refers to maximizing or minimizing a real function. Let us consider a function say f: Z→R, defined from a some-set Z to R (Real-numbers). Now we need to find an element y0 in Set-Z, so-that $f(y0) \leq f(y)$ for all y in Z. A natural name for this would be minimization. If $f(y0) \geq f(y)$, $\forall y \in Z$ we are in a maximization-problem. In addition, the Support Vector Machine is referred to as a supervised machine learning algorithm which is suitable for different data types, such nominal classes, incomplete class values, or binary classes. It works well with unary, nominal and numeric attributes, as well as dealing with instances having missing values. The flexibility of the SVM makes it an advantageous tool in machine learning.[20]

K-nearest-neighbors (k-NN)

The k-NN is one of the most popular classification algorithms, it uses distances between data points to perform the predictions. As a supervised learning method, k-NN assigns new cases to categories based on the majority class of their neighbors. The class label is assigned by selecting the class, which is repeated the most times in this group-of-neighbors.[21] The k-NN uses similarity of measures that are different to classify previously unseen objects into existing classes. This simple-algorithm is, however very accurate also easy to-execute, which is likely why it remains so popular among practitioners in the domain of data science and machine learning. Intuitive modelling results in an ease of interfacing to a wide variety of problems, ranging from recognition tasks to recommendation engines.[22]

Cross-validation

K-fold cross-validation was used in this study for training and validating classification algorithms. Cross-validation is an important concept in ML to evaluate how well a model performs on unseen data. This technique consists of dividing the dataset into distinct "folds". A fold is reserved as the validation set, and the remainder of the folds is used for training. This is repeated several times, and every fold gets a chance to be a validation-set. The outcomes of these cycles are averaged to produce a better estimate of model performance. Cross-validation is primarily used to prevent overfitting, which is a model that fits the data perfectly (a "good" model), but would not work or perform poorly on new, unseen data. By testing the model on several validation-sets, cross-validation provides a more accurate estimate of the ability of the model to generalize or to work well with data it hasn't seen before. In this project we consider 5 K folds for cross-validation procedure and the performance of our model will be showed as the Accuracy, Precision, Recall and F-measure.[23] Such statistics would complete the picture of the trustiness and the model efficacy.

Machine learning algorithm evaluation criteria

In this paper, it is defined as a bulky code that encompasses all the evaluation standards of ML algorithms. It discusses the fundamental measures such as accuracy, precision, recall and F-measure along-side with their significance and applicability. this information enables users to evaluate and compare different ML models efficiently and more confidently, allowing well-informed decisions in such a fast-growing area.[24]

Accuracy: Accuracy is a key standard that is utilized to measure the level of success in a machine learning model. It is the percentage of instances which were classified correctly. Accuracy is the number of correct predictions made as a proportion of the total number of predictions. However, the measure of accuracy is simply an ambiguous measure of performance that gives a false impression in situations where the dataset is asymmetric, that is, when some of the classes are represented by far additional cases than others. So, in like these problems, high accuracy can be deceiving when the model is dominated by the majority class,[25] as in Eq. (1)

$$\text{Accuracy} = (\text{TruePositive} + \text{TrueNegative}) / (\text{TruePositive} + \text{FalsePositive} + \text{TrueNegative} + \text{FalseNegative}) \quad (1)$$

Precision: is the number of true positive predictions to all positive predictions. It demonstrates how much a model is good in not classifying the negative as positives. In simpler words high precision indicates that the model will have extra number of true-positives and minimum number of false-positives. The precision is important in cases when false positives must be kept at a minimum, for example, medical diagnosis or fraud discovery,[25] as in Eq. (2).

$$\text{Precision} = \text{True Positives}/(\text{True Positives} + \text{False Positives}) \quad (2)$$

Recall: The recall (a.k.a. Sensitivity or True-Positive-Rate) quantifies the number of positive class observations that were predicted as such: How many of the all-positive predictions are true. It represents the model ability to label all the positive-samples. A good recall indicates that the model is better at distinguishing the true-positives and presumably unlikely to have missed any. This factor is particularly crucial in situation where there is a strong need to decrease false negatives like filtering of spam or screening of a disease,[25] as in Eq. (3).

$$\text{Recall} = \text{True Positives}/(\text{True Positives} + \text{False Negatives}) \quad (3)$$

F-Measure: The f-measure is the harmonic mean of precision and recall and represents an approximation of the degree of agreement of the model (Krause 1996). It provides a single value which incorporates both precision and recall and is more complete than the metrics separately. In practice we might want to balance precision and recall, and the F-measure offers exactly such a mechanism. For instance, if assuming true negatives less importunate than false positives and negatives, the f-measure can be employed to identify a model that is effective for both high and low FPs: FNs,[26] as in Eq. (4).

$$\text{F} - \text{Measure} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \quad (4)$$

Weighted Average (W. Avg.): The computation of weighted average strategy on "Precision, Recall, and F-Measure" of classification is quite significant, particularly for evaluating the performance of model when dealing with the imbalanced dataset.[27] A weighted average is calculated by first multiplying each value in a data-set (x) by a predetermined weight (w) before tracking the weights sum. These weights indicate the significance, or the frequency of occurrence of a specific figure within the observation sequence. The simple average considers all values as equal, while this particular measure reflects their significance. Consequently, it may be more representative than the others in specific instances.[27] where: wi represents the weight assigned to the value xi, which are the data values in the dataset. The symbol Σ denotes the summation of all values[27] as shown in Eqs. (5) to (7).

$$\text{Weighted Average} = (\text{Sum of weighted terms}) / (\text{Total number of terms}) \quad (5)$$

$$\bar{x} = (w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n)/(w_1 + w_2 + w_3 + \cdots + w_n) \quad (6)$$

$$\bar{x} = \left(\sum_{i=1}^{n} w_i x_i\right) \Big/ \left(\sum_{i=1}^{n} w_i\right) \quad (7)$$

To express this in simpler terms, the Weighted Average (W.Avg.) can be defined as follows in

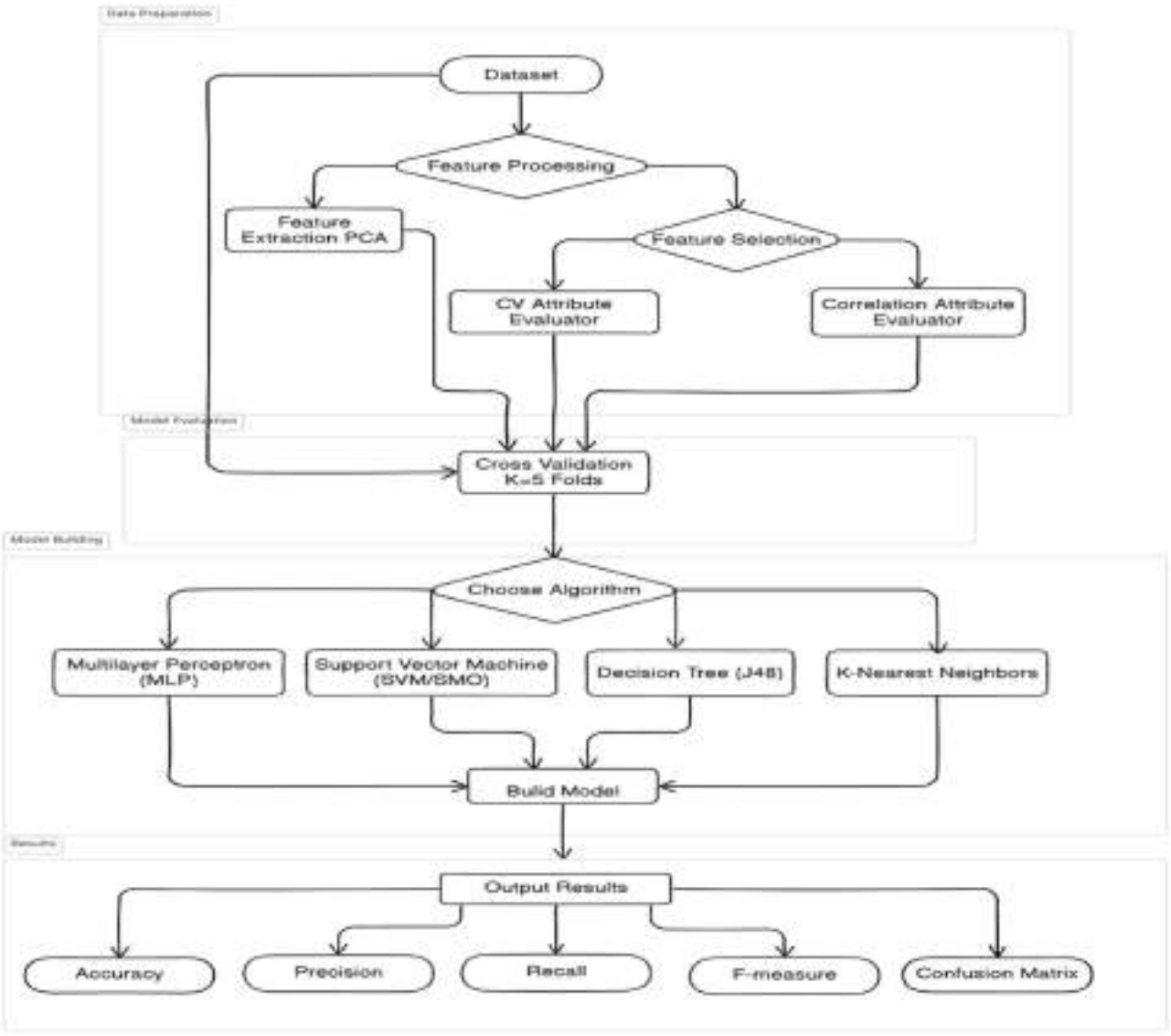

Fig. 3. ML building model and evaluation work flow.

Eqs. (8) to (10):

$$1.\ Q1 = (TP + FN) * Fact - 1 \tag{8}$$

$$2.\ Q2 = (FP + TN) * Fact - 2 \tag{9}$$

$$3.\ \text{“Weighted Average (W. Avg.)} = (Q1 + Q2) / (TP + FN + FP + TN)\text{”} \tag{10}$$

When Fact aligns with a metric like Precision, Recall, or F-measure, Fact-1 and Fact-2 represent the two characteristics of the main class, which are Positive (No) and Negative (Yes) class, respectively, according to the dataset used.

The flow diagram Fig. 3 represents the sequence of steps to implement a model based on our proposed procedure.

Result discussion

In this section, we present the experimental results of these approaches. I will briefly outline the research aims and anticipated findings. Let’s first use PCA feature extraction on the thyroid dataset. After this, the data would be supplied to the ML model. We are interested in analyzing the performance of the proposed selection strategies to down-sample the data. In addition, we want to analyze the effect of the extraction and selection on the time needed to

Table 2. MLP-activity.

Algorithms	Correct/Incorrect Classified Instances	Accuracy (TP + TN)/ (P + N)	Precision TP/(TP + FP)	Recall TP/(TP + FN)	F-Measure (2*Precision*Recall)/ (Precision + Recall)	Consuming Time to build Model
MLP-PCA	361	94.26%	0.953	0.967	0.960	≤2.10 s
Feature	22	5.74%	0.913	0.880	0.896	
Extraction			W.Avg=0.942	W.Avg=0.943	W.Avg=0.942	
MLP-CVAE	324	84.59%	0.906	0.876	0.891	≤2.45 s
Feature	59	15.40%	0.709	0.769	0.738	
Selection			W.Avg=0.851	W.Avg=0.846	W.Avg= 0.848	
MLP-CAE	365	95.30%	0.964	0.971	0.967	≤2.40 s
Feature	18	4.70%	0.925	0.907	0.916	
Selection			W.Avg=0.953	W.Avg=0.953	W.Avg=0.953	
MLP-without	362	94.51%	0.950	0.975	0.962	≤5.00 s
Feature	21	5.48%	0.931	0.870	0.900	
Reduction			W.Avg= 0.945	W.Avg= 0.945	W.Avg= 0.945	

Table 3. MLP confusion matrix.

Algorithms	Confusion matrix Class=No (TP) Class=Yes (FN) Class=No (FP) Class=Yes (TN)	
MLP-PCA Feature Extraction	266 Class=No (TP)	9 Class=Yes (FN)
	13 Class=No (FP)	95 Class=Yes (TN)
MLP-CVAE Feature Selection	241 Class=No (TP)	34 Class=Yes (FN)
	25 Class=No (FP)	83 Class=Yes (TN)
MLP-CAE Feature Selection	267 Class=No (TP)	8 Class=Yes (FN)
	10 Class=No (FP)	98 Class=Yes (TN)
MLP-without Feature Reduction	268 Class=No (TP)	7 Class=Yes (FN)
	14 Class=No (FP)	94 Class=Yes (TN)

build the model. In our analysis, we compared the accuracy, precision, recall and f measure, in order to figure out the best way to reduce the feature space of the thyroid dataset and obtain the best ML performances. These questions will all be answered in the Tables and accompanying article. The number of instances in the dataset is 383, and it contains 16 attributes and one recurred class with two features: No/Yes. The 275 samples belong to the No-Recurred class, while the 108 samples fall in the Yes-Recurred class, hence the proportion is about 72% (No) against 28% (Yes). Such an imbalance in the dataset can cause some classes to be dominated by others, which can make the overall accuracy misleading if only global statistics are used. In order to solve this problem, we take the weighted average to comprehensively evaluate the model.

The application of the PCA reduction technique resulted in a significant reduction in the runtime for model construction with the MLP, achieving speeds more than double (1.85–2.10 seconds with PCA feature extraction compared to 4–5 seconds for the MLP without PCA) method. However, this technique also brought about slight alterations in other parameters, rendering it less effective for feature reduction. In contrast, the Correlation Attribute Evaluator (CAE) as a feature selection demonstrated greater accuracy than both the PCA feature extraction method and the Clustering Variation Attribute Evaluator (CVAE) feature selection method when used alongside the MLP algorithm. In addition, the time to build a model was minimized about half in the case of utilizing all features directly as inputs to the MLP. The main point of PCA feature extraction is that it still mostly concentrates on time consumptions.

For more clarification, when applying the MLP-CAE model, it investigates a prominant accuracy of 95.30%, with an exact error rate of 4.70%. The value of true positives (TP) and true negatives (TN) of precision, which sequentially belong to the No and Yes classes was 0.964. Moreover, the value 0.925 is recorded for false negatives (FN) and false positives (FP). The precision of FN and FP when applying MLP without feature extraction rises to 0.931, which is higher than 0.925 because of the imbalanced dataset. To follow up this challenge, extracting the Weighted Average (W.Avg) was very important, resulting in a value of 0.953. This result outperformes those of alternative models, exploring a successful satisification of true positives and true negatives, while minimizing false positives effectively. The same considerations were utilized to the factor Recall, highlighting the advantages obtained from applying the Weighted Average computation. Tables 2 and 3 and Figs. 4 to 7

inrtoduce comperhensive information about the MLP models.

Upon analyzing the Decision Tree J48 machine learning model, it is clear that the CVAE feature selection approach presents some limitations when contrasted with PCA extraction, which also has minor drawbacks. However, the CAE selection method consistently yields similar results, regardless of whether it is combined with the J48 algorithm. All metrics including Accuracy, Precision, Recall, F measure, show consistent results, except for the time required, which is not influenced by the feature reduction methods. Additionally, using CAE for feature selection on the dataset produces the same outcomes for all metrics as when the J48 algorithm was executed without any feature reduction techniques. It is important to note

Fig. 4. PCA-MLP (Visualize classifier error).

Fig. 5. CVAE–MLP (Visualize classifier error).

Fig. 6. CAE–MLP (Visualize classifier error).

Fig. 7. MLP (Visualize classifier error).

that utilizing the previously discussed feature reduction methods in conjunction with the J48 algorithm does not provide any advantage in this study. Tables 4 and 5 and Figs. 8 to 11 present a comprehensive analysis of the J48 models.

It is obvious that applying the feature extraction method (PCA) with the Nearest Neighbors (kNN) algorithm noticeably develops the performance of the (k-NN) algorithm by producing an accuracy of 92.42%. This improvement is especially clear at the measurements of Precision, Recall, and F-measure. Therefore, a clear contrast appears in the Precision metric; exactly, the precision obtained by executing k-NN along with Correlation Attribute Evaluator (CAE) feature selection that figures the values of 0.945 for True Positives (TP) and True Negatives (TN), which surpasses the precision of 0.933 constituted from utilizing k-NN with PCA feature extraction. This difference can be related to the imbalanced characteristics of the dataset. To overcome these challenges, the computation of Weighted Average (W.Avg.) can be applied, generating a W.Avg. of

Table 4. D.T. J48 activity.

Algorithms	Correct/Incorrect Classified Instances	Accuracy (TP + TN)/ (P + N)	Precision TP/(TP + FP)	Recall TP/(TP + FN)	F-Measure (2*Precision*Recall)/ (Precision + Recall)	Consuming Time to build Model
D.T. J48-PCA	352	91.91%	0.933	0.949	0.944	<1s
Feature	31	8.10%	0.867	0.843	0.854	
Extraction			W.Avg=0.918	W.Avg= 0.919	W.Avg= 0.919	
D.T. J48-CVAE	335	87.46%	0.910	0.916	0.913	<1s
Feature	48	12.53%	0.783	0.769	0.776	
Selection			W.Avg=0.874	W.Avg=0.875	W.Avg=0.874	
D.T. J48-CAE	363	94.78%	0.944	0.985	0.964	<1s
Feature	20	5.22%	0.958	0.852	0.902	
Selection			W.Avg= 0.948	W.Avg= 0.948	W.Avg= 0.947	
D.T. J48-without	363	94.78%	0.944	0.985	0.964	<1s
Feature	20	5.22%	0.958	0.852	0.902	
Reduction			W.Avg= 0.948	W.Avg= 0.948	W.Avg= 0.947	

Table 5. D.T. J48 confusion matrix.

Algorithms	Confusion matrix Class=No (TP) Class=Yes (FN) Class=No (FP) Class=Yes (TN)	
D.T. J48- PCA Feature Extraction	261 Class=No (TP)	14 Class=Yes (FN)
	17 Class=No (FP)	91 Class=Yes (TN)
D.T. J48- CVAE Feature Selection	252 Class=No (TP)	23 Class=Yes (FN)
	22 Class=No (FP)	83 Class=Yes (TN)
D.T. J48- CAE Feature Selection	271 Class=No (TP)	4 Class=Yes (FN)
	16 Class=No (FP)	92 Class=Yes (TN)
D.T. J48- without Feature Reduction	271 Class=No (TP)	4 Class=Yes (FN)
	16 Class=No (FP)	92 Class=Yes (TN)

0.923 for k-NN with PCA, which prevails the values of all other models.

Concerning Recall, k-NN with CAE also appears higher rates of 0.861 for False Positives (FP) and False Negatives (FN) compared to k-NN with PCA, which registers a value of 0.824. Once again, the W.Avg. computation effectively processed this concern, yielding a W.Avg. score of 0.924 for k-NN with PCA that exceeds all other k-NN models. Likewise, in the F-measure metric, the similar pattern was noted, which also was reproduced by using the W.Avg. computation. In addition, remarkable results were obtained

Fig. 8. PCA-D.T. J48 (Visualize classifier error).

Fig. 9. CVAE-D.T. J48 ((Visualize classifier error).

Fig. 10. CAE-D.T. J48 ((Visualize classifier error).

Fig. 11. D.T. J48 ((Visualize classifier error).

through combining the CAE with the k-NN algorithm, whilst the CVAE feature selection method pretended to hinder its efficiency. For more explanation regarding kNN models, please review out Tables 6 and 7 and Figs. 12 to 15.

Moreover, superb precision and notable effectiveness have been observed when combining the PCA feature extraction and Sequential Minimal Optimization, leading to an accuracy of 95.82%. with regard to the Precision factor which related to the True Positive (TP) and True Negative (TN) cases, the SMO-PCA method returns a value of 0.951. On the other hand, the SMO algorithm without feature reduction shapes a brilliant score of 0.964. So, this discrepancy was reconciled through the Weighted Average (W.Avg.) computation, providing a value of 0.959 that exceeds all other SMO models, placing SMO-PCA at the forefront.

In concern to the Recall factor, a challenge was identified with the False Positive (FP) and False

Table 6. kNN-activity.

Algorithms	Correct/Incorrect Classified Instances	Accuracy (TP + TN)/ (P + N)	Precision TP/(TP + FP)	Recall TP/(TP + FN)	F-Measure (2*Precision*Recall)/ (Precision + Recall)	Consuming Time to build Model
kNN-PCA	354	92.42%	0.933	0.964	0.948	< 1s
Feature	29	7.57%	0.899	0.824	0.860	
Extraction			W.Avg= 0.923	W.Avg= 0.924	W.Avg= 0.923	
kNN -CVAE	332	86.86%	0.892	0.927	0.909	< 1s
Feature	51	13.31%	0.794	0.713	0.751	
Selection			W.Avg= 0.864	W.Avg= 0.867	W.Avg= 0.865	
kNN - CAE	353	92.16%	0.945	0.945	0.945	< 1s
Feature	30	7.83%	0.861	0.861	0.861	
Selection			W.Avg= 0.922	W.Avg= 0.922	W.Avg= 0.922	
kNN -without	349	91.12%	0.932	0.945	0.939	< 1s
Feature	34	8.87%	0.856	0.824	0.840	
Reduction			W.Avg= 0.910	W.Avg= 0.911	W.Avg= 0.911	

Table 7. kNN-confusion matrix.

Algorithms	Confusion matrix Class=No (TP) Class=Yes (FN) Class=No (FP) Class=Yes (TN)	
kNN- PCA Feature Extraction	265 Class=No (TP)	10 Class= Yes (FN)
	19 Class=No (FP)	89 Class= Yes (TN)
kNN- CVAE Feature Selection	155 Class=No (TP)	20 Class= Yes (FN)
	31 Class=No (FP)	77 Class= Yes (TN)
kNN- CAE Feature Selection	260 Class=No (TP)	15 Class=Yes (FN)
	15 Class=No (FP)	93 Class=Yes (TN)
kNN- without Feature Reduction	260 Class=No (TP)	15 Class= Yes (FN)
	19 Class=No (FP)	89 Class= Yes (TN)

Negative (FN) cases. without feature reduction the algorithm presented a value of 0.907, which exceeds the value of 0.870 indicated by the PCA-SMO model. once again, the W.Avg. labeled this issue, as a consequence of the imbalanced dataset, yielding a calculated value of 0.958 that overcame all other results across all models in the context of the SMO algorithm.

In addition, our results stipulate that the feature selection (CAE) method displays similar performance to PCA feature extraction, attaining a value of 95.30% accuracy, so it is inefficient when compared to the SMO algorithm without feature reduction. In contrast, the CVAE feature extraction method has shown restrictions when tested to nominal datasets that associated a nominal class. Tables 8 and 9 and

Fig. 12. PCA-kNN (Visualize classifier error).

Fig. 13. CVAE-kNN ((Visualize classifier error).

Fig. 14. CAE-kNN ((Visualize classifier error).

Fig. 15. kNN (Visualize classifier error).

Figs. 16 to 19 supply a precise analysis of all SMO models.

In view of the precise experiment, it is obvious that when coupling the SMO algorithm with PCA feature extraction, the most effective strategy will appear, for achieving maximal accuracy and accurate measures over various factors. This approach is extremely supportive for nominal datasets. Furthermore, this research has appeared that employing a Weighted Average for Precision, Recall and F-measure in classification tasks gives an inclusive estimation of model efficiency, which is critical when dealing with imbalanced datasets.

Table 10, along with Figs. 20 and 21, shows collected data regarding to the evaluation of all models over various algorithms and their individual measurement factors, while also appointing the time needed for configuring and constructing each model. It is worth noting that, when feature reduction techniques are employed before model execution, the MLP demonstrates a significant drop in configuration and construction time approximately halved,

Table 8. SMO-activity.

Algorithms	Correct/Incorrect Classified Instances	Accuracy (TP + TN)/ (P + N)	Precision TP/(TP + FP)	Recall TP/(TP + FN)	F-Measure (2*Precision*Recall)/ (Precision + Recall)	Consuming Time to build Model
SMO-PCA	367	95.82%	0.951	0.993	0.972	< 1s
Feature	16	4.17%	0.979	0.870	0.922	
Extraction			W.Avg= 0.959	W.Avg= 0.958	W.Avg= 0.957	
SMO-CVAE	331	86.42%	0.921	0.887	0.904	< 1s
Feature	52	13.57%	0.737	0.806	0.770	
Selection			W.Avg= 0.869	W.Avg= 0.864	W.Avg= 0.866	
SMO-CAE	365	95.30%	0.951	0.985	0.968	< 1s
Feature	18	4.69%	0.959	0.870	0.913	
Selection			W.Avg= 0.953	W.Avg= 0.953	W.Avg= 0.952	
SMO-without	366	95.56%	0.964	0.975	0.969	< 1s
Feature	17	4.43%	0.933	0.907	0.920	
Reduction			W.Avg=0.955	W.Avg=0.956	W.Avg=0.955	

Table 9. SMO-confusion matrix.

Algorithms	Confusion matrix Class=No (TP) Class=Yes (FN) Class=No (FP) Class=Yes (TN)	
SMO-PCA Feature Extraction	273 Class=No (TP)	2 Class=Yes (FN)
	14 Class=No (FP)	94 Class=Yes (TN)
SMO-CVAE Feature Selection	244 Class=No (TP)	31 Class=Yes (FN)
	21 Class=No (FP)	87 Class=Yes (TN)
SMO-CAE Feature Selection	271 Class=No (TP)	4 Class=Yes (FN)
	14 Class=No (FP)	94 Class=Yes (TN)
SMO-without Feature Reduction	268 Class=No (TP)	7 Class=Yes (FN)
	10 Class=No (FP)	98 Class=Yes (TN)

Fig. 16. PCA-SMO (Visualize classifier error).

Fig. 17. CVAE-SMO (Visualize classifier error).

in comparison to the cases without applying these techniques, despite it still involving more time than other models and their algorithms. Additionally, an improved efficiency of the k-NN model was shown in its results by utilizing PCA as a method for feature extraction, as previously indicated. On the other hand, the J48 model does not earn any benefits from feature reduction during its execution. Furthermore, the CVAE adversely influences all models across the various algorithms.

Fig. 18. CAE-SMO (Visualize classifier error).

Fig. 19. SMO (Visualize classifier error).

Table 10. Cumulative analysis of models.

Models	Accuracy	W.Avg. for Precision	W.Avg. for Recall	W.Avg. for F-measure	Investing time in model development
1- SMO-PCA Feature Extraction	95.82%	W. Avg= 0.959	W. Avg= 0.958	W. Avg= 0.957	<1 s
2- MLP-CAE Feature Selection	95.30%	W. Avg= 0.953	W. Avg= 0.953	W. Avg= 0.953	<2.40
3- D.T. J48-CAE Feature Selection	94.78%	W. Avg= 0.948	W. Avg= 0.948	W. Avg= 0.947	<1 s
4- D.T. J48-CAE without Feature Reduction	94.78%	W. Avg= 0.948	W. Avg= 0.948	W. Avg= 0.947	<1 s
5- kNN-PCA Feature Extraction	92.42%	W. Avg= 0.923	W. Avg= 0.924	W. Avg= 0.923	<1 s

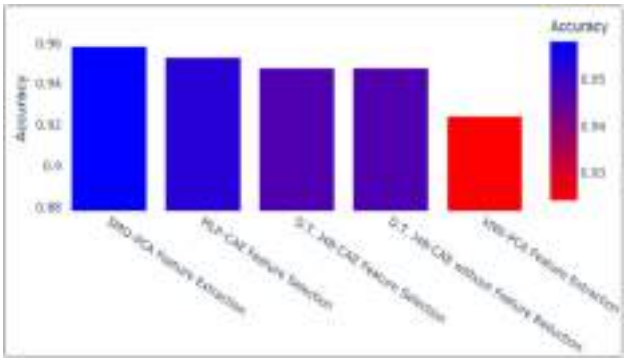

Fig. 20. Model accuracy comparison

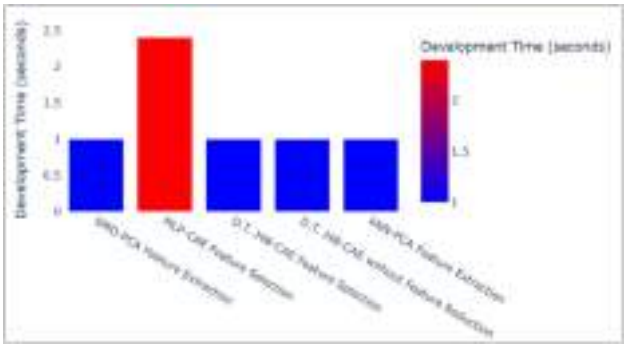

Fig. 21. Model development time Comparison

Conclusion

In any domain, be it medical, engineering, social, or otherwise, the strategies that are applied here highlight choosing the appropriate algorithms to simplify the data's feature space and to deal with nominal data in efficient way. Moreover, estimating the impact of feature reduction methods on machine learning processes. Research[5] stipulates that the SVM/SMO algorithm surpasses others in various phases. In this study, the conjunction of the SVM/SMO algorithm with PCA for feature reduction and using 5-k folds demonstrates exceptional effectiveness compared to substitute methods, as well when integrated with correlation. Therefore, we deduce that SVM/SMO is the impactful approach for dealing with nominal datasets. Whilst the J48 algorithm suffered an inconsiderable reduction in their efficacy when coupled with PCA, this contradicts the findings in paper,[28] which validated an enhancement in J48's rendition when PCA was utilized as a training set in lack of cross-validation. However, when the J48 algorithm was applied in conjunction with a correlation attribute evaluator, it produced results similar to those achieved without applying any of the feature reduction methods. The kNN algorithm executes efficiently when combined with PCA and when utilized in conjunction with a correlation attribute evaluator. Conversely, when the MLP algorithm is integrated

with the correlation attribute evaluator, it produces the optimal outcomes in respect of accuracy and the Weighted Average results. However, it should be noted that the PCA coupled with a MLP shows the greatest impact on reducing the time required to build the model.

Future work

I aspire to conduct tests on diverse data types, in the future, rather than limiting myself to nominal datasets. This exhaustive approach will supply researchers with the required insights to select the most qualified algorithms or models based on the specific datasets they are exploiting. This endeavor is crucial for improving model performance and attaining accurate results, which may assist in detecting issues such as the poor outcomes produced by the chosen model.

Acknowledgement

I would like to express my appreciation to the Machine Learning Repository website and its administrative team for their support in making datasets available to researchers, enabling them to conduct their studies effectively.

Author declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images that are not ours have been included with the necessary permission for republication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Baghdad.

References

1. Velliangiri, S, Alagumuthukrishnan, S, Thankumar joseph SI. A Review of Dimensionality Reduction Techniques for Efficient Computation. J Pro C S. 2019;165:104–11. https://doi.org/10.1016/j.procs.2020.01.079.
2. Jia W, Sun M, Lian J, Hou S. Feature dimensionality reduction: a review. J Complex Intell Syst., 2022;8:2663–93. https://doi.org/10.1007/s40747-021-00637-x.
3. Isomura T, Toyoizumi T. Dimensionality reduction to maximize prediction generalization capability. J Nat Mach Intell. 2021;3:434–46. https://doi.org/10.1038/s42256-021-00306-1.
4. Skaka-Čekić F, Husić JB, Odžak A, Hadžialić M, Huremović A, Šehić K. Dimensionality reduction of independent influence factors in the objective evaluation of quality of experience. Sci Rep. 2022;12:10320. https://doi.org/10.1038/s41598-022-13803-z.
5. Borzooei S, Briganti G, Golparian M, Lechien JR, Tarokhian A. Machine learning for risk stratification of thyroid cancer patients: a 15-year cohort study. J Eur Arch Oto-Rhino-Laryngol. 2024;281:2095–104. https://doi.org/10.1007/s00405-023-08299-w.
6. Borzooei S, Tarokhian A. Differentiated Thyroid Cancer Recurrence [Dataset]. 2023. UCI Machine Learning Repository. https://doi.org/10.24432/C5632J.
7. Greenacre M, Groenen PJF, Hastie T, D'Enza AI, Markos A, Tuzhilina E. Principal component analysis. Nat Rev Methods Primers. 2022 Dec;2:100. https://doi.org/10.1038/s43586-022-00184-w.
8. Patel K, Tiwary GJ, Pandey KK, Asrani DK. Unsupervised Machine Learning Algorithm: Pca (Principal Component Analysis) Comprehensive Review. IR J M E T S. 2024 Feb;06(02):1303–1314. https://www.doi.org/10.56726/IRJMETS49457.
9. Elst HV J. Tutorial on principal component analysis, with applications in R. arXiv preprint arXiv. Cornell University. 2021 Dec:1–37. https://doi.org/10.48550/arXiv.2112.12067.
10. Maćkiewicz A, Ratajczak W. Principal components analysis (PCA). J Comp Geosci., Mar 1993;19(3):303–42. https://doi.org/10.1016/0098-3004(93)90090-R.
11. Büyükkeçeci M, Okur MC, A Comprehensive Review of Feature Selection and Feature Selection Stability in Machine Learning. Gazi Uni J of Sci., 2023;36(4):1506–20. https://doi.org/10.35378/gujs.993763.
12. Frank E, Hall MA, Witten IH. The WEKA Workbench. Hamilton (NZ): The University of Waikato: Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Fourth Edition, 2016.
13. Simon J. The Complete Software Platform: Clustering Variation [Internet]. [San Diego] SourceForge [registered 2014 March 13; updated 2014 Dec 20].
14. Abdel-aziem AH, Soliman THM. A Multi-Layer Perceptron (MLP) Neural Networks for Stellar Classification: A Review of Methods and Results. International Journal of Advances in Applied Computational Intelligence, 2023;3(2):29–37. https://doi.org/10.54216/IJAACI.030203
15. Nasir IM, Khan MA, Yasmin M, Shah JH, Gabryel M, Scherer R, *et al.* Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training. Sensors. 2020;20(23),6793:1–18. https://doi.org/10.3390/s20236793
16. Behadili SF, Abd MS, Mohammed IK, Al-Sayyid MM. Breast cancer decisive parameters for Iraqi women via data mining techniques. J Cont Med Sci. 2019;5(2):71–6. https://doi.org/10.22317/jcms.v5i2.573.
17. Popescu M-C, Balas VE, Perescu-Popescu L, Mastorakis N. Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, July 2009;8(7):579–88. https://dl.acm.org/doi/abs/10.5555/1639537.1639542
18. Duran Z, Akargöl I, Doğan T. Data Mining, Weka Decision Trees. OPRD,2023;3(1):401–16. https://doi.org/10.56038/oprd.v3i1.376.
19. Abd AL-BND MS. Sentiment analysis and opinion mining via microblogging in social media like: twitter [master's thesis on the Internet]. master's thesis. Ankara: Çankaya University, Institute of Science. 2015. https://doi.org/10.13140/RG.2.2.22379.04649.

20. Nakanishi KM, Fujii K, Todo S. Sequential minimal optimization for quantum-classical hybrid algorithms. Phys Rev Res Oct. 2020;2:043158. https://doi.org/10.1103/PhysRevResearch.2.043158.
21. Danny M, Muhidin A, Jamal A. Application of the K-Nearest Neighbor Machine Learning Algorithm to Preduct Sales of Best-Selling Products. Brilliance. 2024 Jun;4(1):255–64. https://doi.org/10.47709/brilliance.v4i1.4063.
22. Sarang P. Thinking Data Science. Switzerland: Springer Cham; 2023 [cited 2023 Mar 02]. p.[XX–358]. https://doi.org/10.1007/978-3-031-02363-7.
23. Chamorro-Atalaya O, Arévalo-Tuesta J, Balarezo-Mares D, Gonzáles-Pacheco A, Mendoza-León O, Quipuscoa-Silvestre M, *et al.* K-Fold Cross-Validation through Identification of the Opinion Classification Algorithm for the Satisfaction of University Students. Int. J. Onl. Eng. 2023 Aug 16;19(11):140–58. https://doi.org/10.3991/ijoe.v19i11.39887.
24. Rainio O, Teuho J, Klén R. Evaluation metrics and statistical tests for machine learning. Sci Rep. 2024;14:6086. https://doi.org/10.1038/s41598-024-56706-x. .
25. Gaye B, Zhang D, Wulamu A. Sentiment classification for employees reviews using regression vector- stochastic gradient descent classifier (RV-SGDC). PeerJ Computer Science. 2021;7:e712. https://doi.org/10.7717/peerj-cs.712.
26. Hand DJ, Christen P, Kirielle N. F*: an interpretable transformation of the F-measure. Mach Learn. 2021;10:451–56. https://doi.org/10.1007/s10994-021-05964-1.
27. Lee DS, Son SY. Weighted Average Ensemble-Based PV Forecasting in a Limited Environment with Missing Data of PV Power. Sustainability. 2024;16(10):1–17. https://doi.org/10.3390/su16104069.
28. Oo AN. Comparative Study of Principal Component Analysis (PCA) based On Decision Tree Algorithms. IJASRE. 2018 Jun.; 4(6):122–6. https://doi.org/10.31695/IJASRE.2018.32767.

تحليل معلوماتي لتطبيق طرق تقليل الميزات على خوارزميات التعلم الآلي الخاضعة للإشراف

مصطفى س. عبد

قسم علوم الحاسب، كلية العلوم، جامعة بغداد، بغداد، العراق.

الخلاصة

تُعد طرق تقليل الميزات أساسية لتحسين خوارزميات التعلم الآلي ML من خلال تقليل عدد الميزات في مجموعة البيانات. تستكشف هذه الدراسة آثار تحليل المكونات الرئيسية PCA على خوارزميات التعلم الآلي في إطار التصنيف غير المتوازن، بالتزامن مع تقنيات اختيار الميزات مثل مُقيّم سمات التباين العنقودي CVAE ومُقيّم سمات الارتباط CAE قيّم البحث فعالية العديد من طرق التعلم الآلي، بما في ذلك المُدرك متعدد الطبقاتMLP ، وشجرة القرار J48، وأقرب جارk-NN، والتحسين الأدنى التسلسلي SMO تكشف النتائج أن نموذج MLP يستخدم تحليل المكونات الرئيسية (PCA) لما يقرب من 50% من وقت بناء النموذج أثناء التحقق المتبادل K--5 Foldsمع تحقيق دقة مماثلة. أظهر نموذج J48 استجابة محدودة لتقليل الميزات، في حين أن CVAE قد يؤثر سلبًا على الأداء في جميع النماذج. أدى تطبيق تحليل المكونات الرئيسية PCA مع SMO إلى رفع دقة التشخيص من 95.56% إلى 95.82%. وحقق نهج k-NN زيادة في الدقة من 91.12% إلى 92.42% مع تحليل المكونات الرئيسية، وحسّن CAE دقة النموذج بشكل ملحوظ. علاوة على ذلك، يستخدم هذا البحث المتوسط المرجح للدقة والتذكر ومقياس F لتقديم تقييم شامل لأداء النموذج على مجموعة بيانات غير متوازنة. وقد استُخدمت مجموعة بيانات الغدة الدرقية من النوع الاسمي كدراسة حالة لهذا البحث.

الكلمات المفتاحية: الارتباط، سمة CV، تقليل الميزة، MLP ،k-NN ،J-48، القيمة الاسمية، تحليل المكونات الرئيسية، SMO، بيانات الغدة الدرقية.