Received: 22 July 2024 | Revised: 23 September 2024 | Accepted: 2 January 2025 | Published online: 10 February 2025

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2025, Vol. 00(00) 1–12
DOI: 10.47852/bonviewJDSIS52023906

BON VIEW PUBLISHING

Integrating Data Analysis Methods with Machine Learning Algorithms for Mixed Data Types: Does This Combination Improve Predictive Models' Accuracy?

Nikolaos Papafilippou[1,*], Zacharenia Kyrana[1], Emmanouil Pratsinakis[1], Efstratios Kiranas[2], Alexandra-Maria Michailidou[3], Angelos Markos[4] and George Menexes[1]

[1]*Laboratory of Agronomy, Aristotle University of Thessaloniki, Greece*

[2]*Department of Nutritional Sciences and Dietetics, International Hellenic University, Greece*

[3]*Department of Food Science and Technology, Aristotle University of Thessaloniki, Greece*

[4]*Department of Primary Education, Democritus University of Thrace, Greece*

Abstract: In this study, we examined the potential of integrating multivariate data analysis methods as a preliminary stage for machine learning techniques to augment their predictive power. These methods encompass principal component analysis, multiple correspondence analysis, and non-linear categorical principal component analysis with optimal scaling. The machine learning approaches evaluated include Support Vector Machines, Stochastic Gradient Descent, Naïve Bayes, K-Nearest Neighbor, Decision Trees, Random Forests, Adaptive Boosting, and Multinomial Logistic Regression. We conducted experiments using data from a nationwide survey, comprising a total sample of 42,593 adolescents who answered more than 155 questions related to their eating habits. The dependent variable, body mass index (BMI), was measured and employed in the analysis as both a quantitative and qualitative variable. The index values were initially classified based on the World Health Organization's recommendations. The results indicated that predictions are more reliable when utilizing the BMI as a qualitative variable within a four-class structure. Implementing a multivariate data analysis strategy before applying machine learning algorithms not only conserves time but also facilitates the selection of the most effective predictive model. Although dimensionality reduction may not consistently enhance the models' predictive abilities, it contributes to the "interpretability" of the results.

Keywords: multivariate data, principal component analysis, categorical principal component analysis, machine learning algorithms, SVC, random forest classifier, multinomial logistic regression

1. Introduction

Data analysis (Analyse des Données in French) provides an alternative methodological and philosophical approach to statistical inference. It includes three main families of methods [1, 2]: (a) the correspondence analysis (CA) or Analyse Factoriel des Correspondances—AFC (bivariate and multivariate), (b) principal component analysis (PCA), and (c) hierarchical cluster analysis. A distinctive characteristic of these methods is the symmetrical treatment of the variables, with no distinction made between dependent and independent. The aim of these methods is to reveal and describe latent structures potentially present in multi-dimensional data tables; accomplished through dimension reduction and transformation of the initial mathematical space that represents the phenomenon under study. The new dimensions, which are usually structured by complex relationships between variables, are ultimately interpreted as new composite variables or "factors", "dimensions", "components", or "factorial axis". A crucial aspect of these methods is that they do not necessitate an a priori assumption of the existence of some theoretical distribution or assumption regarding the parameters of the population under consideration.

Machine learning falls within the realm of computer science and focuses on the development of algorithms that "learn" from collected data utilizing prior knowledge and experience without being programmed with explicit rules. The objective is to discover patterns and relationships to make predictions or decisions. There are three fundamental forms of machine learning [3]: (a) supervised learning, in which the algorithm constructs a function mapping given inputs (training set) to known desired outputs, with the aim of generalizing the function to inputs with unknown outputs, (b) unsupervised learning, where the algorithm constructs

*Corresponding author: Nikolaos Papafilippou, Laboratory of Agronomy, Aristotle University of Thessaloniki, Greece. Email: npapafil@agro.auth.gr

a model for a set of inputs in the form of observations without knowledge of the desired outputs, and (c) reinforcement learning in which the algorithm learns an action strategy through direct interaction with the environment. The first form is employed in classification, prediction, and interpretation problems; the second form in association analysis, clustering, and dimensionality reduction problems; and the third form in planning problems, such as controlling a robot's movements.

When applying the algorithms, the data set is divided into a subset for training (train_set), a subset for testing (test_set), and sometimes also a subset for validation (cross_validation). The model is trained on the training subset, and its predictive ability is assessed using the testing subset [4–7]. We also applied classes such as Standarscaler, GridSearch, Ada-boosting, and pipeline to improve the algorithms' prediction accuracy.

Predicting health-related outcomes, such as body mass index (BMI), is crucial for understanding and managing risks associated with adolescent health [8, 9]. One approach to enhancing prediction accuracy is through the integration of traditional data analysis methods and machine learning techniques. This study aims to explore how data analysis methods, typically employed in the preparatory stage of machine learning (commonly known as "data preprocessing"), can improve the performance of predictive models. Despite the growing interest in applying machine learning to health-related data, few studies have examined the synergy between data analysis methods, such as CA or PCA, and machine learning algorithms. This paper aims to fill that gap by focusing on a specific health metric (BMI) and its relationship with dietary habits, using a dataset of 140 food consumption frequencies collected from adolescent students in Greece. The research problem lies in predicting BMI with higher accuracy by leveraging latent structures within the dietary data that could be missed by conventional machine learning techniques alone. This study proposes a hybrid methodology that combines dimension reduction techniques from data analysis with supervised learning algorithms.

The purpose of this study is to examine the potential of incorporating data analysis methods during the data preparation stage before applying machine learning algorithms, in order to enhance their predictive capabilities. The expected results are that data analysis, prior to the application of algorithms, can offer significant advantages in prediction accuracy. For instance, if a particular data analysis or statistical method provides a satisfactory level of explanation for the variability of BMI, the analysis can be concluded at that point without proceeding further with machine learning algorithms.

This article is structured as follows: Section 2 provides a detailed literature review, covering both the multivariate data analysis methods and the machine learning algorithms evaluated. The emphasis is placed on how data analysis methods can enhance the preprocessing stage of machine learning. In Section 3, we describe the research methodology, outlining the research design, participant information, and the process of integrating data analysis methods before machine learning application, and present the results of the multivariate analysis and the predictive accuracy of the machine learning algorithms. The discussion in Section 4 focuses on the importance of combining data analysis techniques in the preparatory stage, particularly when a method successfully explains the variability of BMI. Finally, in Section 5 we conclude with insights on the potential impact of this approach.

2. Literature Review

2.1. The multivariate methods of analysis

The data analysis methods examined in this study include PCA [1, 10, 11], multiple correspondence analysis (MCA) [1, 2, 12], categorical principal component analysis with optimal scaling (CATPCA) [2, 13], categorical regression with optimal scaling (CATREG) [14], and chi-squared automatic interaction detection (CHAID) [15].

PCA [11] is a statistical technique employed to identify patterns and relationships in high-dimensional datasets. It involves transforming the original data into a new coordinate system such that the first coordinate axis (first principal component) accounts for the maximum possible variance in the data, with subsequent axes accounting for the maximum remaining variance. The primary objective of PCA is to reduce data dimensionality, while preserving as much of initial variation as possible. This is achieved by identifying the principal components, which are linear combinations of the initial variables, and projecting the data onto these components. Typical steps for performing PCA include the following:

1) Standardize the data by subtracting the mean and dividing by the standard deviation
2) Compute the covariance matrix of the standardized data [that is equivalent to pair-wise correlation matrix (with Pearson's r correlation coefficients)]
3) Compute the eigenvectors and eigenvalues of the covariance matrix
4) Sort the eigenvectors according to their corresponding eigenvalues (in descending order) to obtain the principal components
5) Project the data onto the principal components to obtain a new, lower-dimensional representation.

MCA [1, 2] is a statistical technique used for analyzing categorical data, serving as an extension of PCA for handling categorical variables. MCA identifies patterns and relationships among categorical variables by transforming the original data into a new coordinate system. MCA involves creating a special contingency table of categorical variables (the Burt Table) and computing the chi-square distance between all pairs of categories. The chi-square distance measures the similarity between two categories, and this information is used to identify groups of similar categories. Like PCA, MCA also identifies principal components (or factorial axes) that explain the maximum possible variance (inertia) in the data. The first principal component accounts for the maximum inertia, the second for the maximum remaining, and so on. MCA is useful for exploring the data structure and identifying patterns and relationships among categorical variables not evident from simple frequency tables. Visualization of data using graphical representations (French plots and biplots) can also aid in this process.

CATPCA [11] is another statistical technique for analyzing categorical data, similar to MCA. CATPCA is an extension of PCA that can simultaneously handle categorical variables (nominal, ordinal) and scale variables. CATPCA is valuable when dealing with large datasets containing numerous categorical and scale variables, as it can help identify the most important features and reduce noise in the data. It is also useful for visualization, data compression, and feature extraction [11].

CATREG [14] is a statistical method used to analyze the relationship between a categorical or scale-dependent variable and

one or more continuous or categorical variables. CATREG uses optimal scaling to transform categorical variables into scale ones that can be used in regression analysis. This transformation is achieved by assigning numerical values to each category of the categorical variables using the alternating least squares algorithm. CATREG is useful in situations where the relationship between the dependent and independent variables is not linear or where there are non-linear interactions between the independent variables. The output of CATREG typically includes coefficients for the independent variables, standard errors, and significance levels. These coefficients can be used to interpret the relationship between the dependent and independent variables.

CHAID [15] is a decision tree-based statistical technique used for conducting exploratory data analysis and predictive models. It is particularly suited for analyzing datasets with categorical (nominal or ordinal) variables as both predictors and the target variable. CHAID uses a recursive partitioning algorithm that iteratively splits the data into subgroups based on the categories of predictor variables, aiming to maximize the homogeneity (i.e., similarity) of the target variable within each subgroup. The splitting is performed based on the chi-squared statistic, which measures the independence between the predictor and the target variable. The CHAID algorithm continues to split the data into subgroups until certain stopping criteria, such as a minimum sample size or a maximum number of levels, are met.

2.2. The machine learning algorithms

The machine learning algorithms applied in this study include the support vector classifier (SVC), Stochastic Gradient Descent (SGDClassifier), Naïve Bayes (GaussianNB), K-Nearest Neighbor (KNN), Decision Tree Classifier, Random Forest Classifier, and Logistic Regression Multinomial [4–6, 16].

2.2.1. Support vector machines (SVM)

SVM [17] is a machine learning algorithm that is commonly used for classification and regression analysis. SVM is based on the idea of finding the maximum margin hyperplane, which is the line or plane that separates data points into different classes with the maximum possible margin or distance between classes. In the case of two-class classification problem, the SVM finds the hyperplane that best separates the data into two classes. The hyperplane is selected in a way that it maximizes the margin between the two classes, which is the distance between the hyperplane and the closest data points from each class. The data points that are closest to the hyperplane are called support vectors. In multi-class classification, multiple binary classifiers are trained, one for each pair of classes. SVM can efficiently handle high-dimensional data and data with high "noise", as the maximum margin approach helps to reduce the influence of noisy points. Additionally, SVM can handle non-linearly separable data by projecting it into higher dimensional space, where a linear boundary can be found. This is achieved by using kernel functions, which map the data to a higher dimensional space.

SVM can handle both linearly separable and non-linearly separable data by using different types of kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid. The choice of kernel function depends on the nature of the data and the classification problem. The function maps input values to output values based on Euclidean distances from a central point or multiple central points in multi-dimensional space and is defined as:

$$f(x) = \sum_i^m \varphi(\|x - c_i\|), \varphi(\|x - c_i\|) = \exp(-\boldsymbol{\gamma} \|x - c_i\|^2),$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{c_i}$ is the central vector, $\|\cdot\|$ is the Euclidean distance between the input vector and the central vector, and $\boldsymbol{\gamma}$ is a parameter that controls the shape of the decision boundary. A small value of $\boldsymbol{\gamma}$ means a larger radius for the RBF kernel, resulting in a smoother decision boundary and a model more tolerant of the "noise" and outliers. In contrast, a large value leads to a smaller radius for the kernel, resulting in a more complex decision boundary that best fits the training data, but is more prone to overfitting. Another kernel function is the sigmoid, which maps each real value to a value between 0 and 1, given by the relation:

$$\sigma(x) = 1/(1 + \exp(-x)),$$

where x is the input vector probabilities and binary decisions.

Other parameters of the SVM algorithm [17], include class weights, which accounts for class imbalances in the data by assigning different weights to each class, the tolerance for stopping criterion, which determines the minimum improvement in the objective function required to continue the iterations and the regularization parameter C, which determines the trade-off between maximizing the margin and minimizing classification error. A smaller C value results in a wider margin, which can lead to more misclassifications, while a larger C value results in a narrower margin, which can lead to overfitting. These parameters can be adjusted using techniques such as Grid Search, Random Search, or Bayesian optimization to find the combination that yields the best performance on the dataset.

2.2.2. Decision tree

A decision tree is a supervised machine learning algorithm used for classification and regression tasks. It constructs a tree-like model of decisions and their possible consequences based on features of the input data. Each node of the tree represents a decision based on a feature or a combination of features, and each branch represents a possible outcome or further decision. The leaves of the tree represent the final decision or prediction. For classification trees, the prediction is the class label, while for regression trees, it is a continuous value. The tree structure provides a visual representation of the decisions and relationships between the features and the target variable. Creating a decision tree involves choosing the feature to be split at each node and determining the optimal split point. One of the main advantages of decision trees is their ease of understanding and interpretation, as they can handle both linear and non-linear relationships between attributes and the target variable.

There are several algorithms for creating decision trees [18], such as ID3 (Iterative Dichotomizer 3), C4.5 an improved version of ID3, and CART (Classification and Regression Trees). The choice of algorithm depends on the specific problem and the type of data used. ID3 uses information gain as a criterion to choose the best feature to split the data. The information gain is calculated as the difference between the entropy of the parent node and the weighted sum of the entropies of the child nodes. C4.5 uses the gain ratio to separate the data, while CART constructs binary trees by recursively dividing the data into two subsets based on the value of a single feature and uses the Gini index as a criterion for separating the data choosing the attribute that minimizes its value.

The Gini index [19] is calculated as the probability that one case of a data set is misclassified if it is assigned a class label based on the class distribution of cases in the data set and is defined as:

$$\text{Gini} = 1 - {p_1}^2 - {p_2}^2 - \ldots - {p_k}^2 = 1 - \sum_1^k p_i^2$$

where p_i is the probability that the case belongs to the i class out of the k of the data set. The Gini index ranges from 0 to 1, with a value of 0 indicating a perfectly clean data set (all cases belong to the same class) and a value of 1 indicating a perfectly impure data set (cases are evenly distributed across all classes). For example, consider a dataset with two classes A and B, and let's assume 70% ($p_A = 0.7$) of examples belong to class A and 30% ($p_B = 0.3$) belongs to class B., So, Gini is calculated:

$$\text{Gini} = 1 - (0.7^2 + 0.3^2) = 0.42$$

Entropy [19] is a measure of the purity or uncertainty of a set of examples in a decision tree or any other machine learning algorithm. The entropy value ranges from 0 to 1, where 0 indicates that the set is completely pure (all examples have the same category) and 1 indicates that the set is equally balanced (half positive and half negative). A high entropy value indicates high uncertainty or impurity in the ensemble, while a low entropy value indicates low uncertainty or purity in the ensemble. The entropy of a set S is defined as:

$$\text{Entropy(S)} = -\sum_{i=1}^{k} \left(pi * log_2(pi)\right),$$

where pi is the proportion of the samples in the node that belong to class i and k is the number of classes. For the same example, entropy is calculated as:

$$\text{Entropy(S)} = -(\text{p}_\text{A} \cdot \log_2(\text{p}_\text{A}) + \text{p}_\text{B} \cdot \log_2(\text{p}_\text{B})) =$$
$$Entropy(S) = -(0.7 \cdot \log_2(0.7) + 0.3 \cdot \log_2(0.3)) = 0.8813$$

Information gain [19] is a measure used in decision tree algorithms to determine the best feature to split a dataset. The feature that provides the highest information gain is selected as the splitting criterion. Information gain is based on the concept of entropy, which is a measure of the impurity or randomness of a set of examples. In decision trees, entropy is calculated for each attribute, and the attribute with the highest information gain is chosen as the next split. The formula for information gain is:

$$\text{IG(S, A)} = \text{Entropy(S)} - \sum(|\text{Sv}|/|\text{S}|) * \text{Entropy(Sv)},$$

where S is the current dataset being split, A is an attribute being considered for the split, Sv is the subset of S where the value of attribute A is v, $|Sv|$ is the number of examples in subset Sv, $|S|$ is the total number of examples in S, $Entropy(S)$ is the entropy of S, $Entropy(Sv)$ is the entropy of subset Sv.

Let's assume we have a dataset S with 10 samples, 6 samples belong to class A, and 4 samples belong to class B. The entropy of the full dataset S is:

$$\text{Entropy(S)} = -(\text{p}_\text{A} \cdot \log_2(\text{p}_\text{A}) + \text{p}_\text{B} \cdot \log_2(\text{p}_\text{B}))$$
$$= -(0.6 \cdot \log_2(0.6) + 0.4 \cdot \log_2(0.4)) = 0.971$$

Now suppose we split the data on an attribute A that results in two subsets: S_1 (5 samples), 4 class A and 1 class B, S_2 (5 samples), 2 class A and 3 class B. Entropy for each subset and IG are calculated as:

$$Entropy(S_1) = -(0.8 \cdot log_2(0.8) + 0.2 \cdot log_2(0.2)) = 0.722$$
$$Entropy(S_2) = -(0.4 \cdot log_2(0.4) + 0.6 \cdot log_2(0.6)) = 0.971$$
$$\sum(|Sv|/|S|) * Entropy(Sv) = 0.5 * 0.722 + 0.5 * 0.971 = 0.8465$$
$$IG(S, A) = 0.971 - 0.8465 = 0.1245$$

All three algorithms follow a top-down approach to growing the decision tree, starting at the root node and recursively splitting the data until a cut-off criterion is met. The stopping criterion can be based on the depth of the tree (max_depth), the number of instances in a leaf node, or the amount of purity in a node. However, decision trees may also have some disadvantages, such as the tendency to overfit the data and the instability of the tree structure due to small changes in the data. To address these issues, various techniques, such as pruning and random forests, have been developed to improve their performance.

2.2.3. Random forest

Random forest is a machine learning algorithm that is used for classification, regression, and other tasks. It is an ensemble learning method that combines multiple decision trees and aggregates their predictions to make more accurate predictions than any individual tree. The idea behind random forest [20, 21] is to generate a large number of decision trees, each trained on a randomly selected subset of data. The randomness introduced by training each tree on a different subset of the data helps reduce overfitting and improve the generalizability of the model. It is more robust to overfitting, has less variance, and can handle missing data and noisy data more efficiently. It also provides a measure of feature importance, which can be useful for feature selection. However, random forest is a more complex algorithm and can be computationally expensive, especially when the number of trees in the forest is large. Additionally, the prediction time can be slower than that of a decision tree, since each tree in the forest must make a prediction.

2.2.4. Logistic regression

Logistic regression is a machine learning algorithm [22], used for binary classification problems, where the goal is to predict a binary output variable (e.g., true/false or positive/negative) based on one or more input variables (also known as features or predictors). It works by modeling the probability of the output variable as a function of the input variables using a logistic function, which maps any real-valued input to a value between 0 and 1. However, in some cases, we may have more than two classes, so we use a variant of logistic regression called multinomial logistic regression or softmax regression. The logistic function is the sigmoid

$$\sigma(\text{z}) = 1/(1 + \text{e}^{-\text{z}}),$$

where z is the linear combination of the input variables and their relative weights.

The logistic regression model is trained using a set of cases where the output variable is known and the weights of the input variables are adjusted to maximize the probability of the observed outputs given the inputs. This is typically done using an optimization algorithm, such as the Gradient Descent algorithm, which iteratively adjusts the weights to minimize a cost function that measures the difference between the predicted probabilities and the actual probabilities. The cost function is typically the cross-entropy loss, which is defined as:

$$J(w) = -1/m * \sum [y(i) * \log(h(x(i))) + (1 - y(i)) * \log(1 - h(x(i)))],$$

where w is the vector of weights, m is the number of cases, $x(i)$ and $y(i)$ are the input and output variables for the i-case, and $h(x(i))$ is the predicted probability that the output variable is positive for the i-case. The goal of model training is to find the set of weights w that minimizes the cross-entropy loss function. In the case of polynomial logistic regression, the probability of each class is modeled via the softmax function:

$$P(Y = j|X = x) = e^{(b_{-j} + w_{-j'} * x))} / \sum_{j=1}^{k} e^{(b_{-j} + w_{-j'} * x))},$$

where $P(Y = j|X = x)$ is the probability that the outcome variable is j with input variables x, b_j is the bias term for class j, w_j is the weight vector for class j, and k is the index for all classes. The cost function is similar to the cost function used for binary logistic regression.

Let's assume we have a classification problem where the outcome variable Y have three classes ($k = 3$) and the input variable X have two features x_1 and x_2 with two classes each of them. The input features $x_1 = [1,0]$ and, $x_2 = [0,1]$, indicates that the data points belongs to class 1 for x_1 and to class 2 for x_2. The initial values of the weights w_j and biases b_j for each class of Y are as follows: for class 1: $w_1 = [0.2, 0.4]$, $b_1 = 0.1$; for class 2: $w_2 = [0.3, 0.3]$, $b_2 = 0.2$; and for class 3: $w_3 = [0.1, 0.5]$, $b_3 = 0.1$. The Softmax function for each observation is computed as follows:

For $x_1 = [1, 0]$

$$z_1 = b_1 + w_1\prime * x_1 = 0.1 + (0.2 * 1 + 0.4 * 0) = 0.3$$
$$z_2 = b_2 + w_2\prime * x_1 = 0.2 + (0.3 * 1 + 0.3 * 0) = 0.5$$
$$z_3 = b_3 + w_3\prime * x_1 = 0.1 + (0.1 * 1 + 0.5 * 0) = 0.2$$
$$P(Y = j|X = x_1) = \frac{e^{(z_j)}}{(e^{(z_1)} + e^{(z_2)} + e^{(z_3)})}, j = 1, 2, 3$$

$$P(Y = 1|X = x_1) = 0.3197$$
$$P(Y = 2|X = x_1) = 0.3906$$
$$P(Y = 3|X = x_1) = 0.2897$$

For $x_2 = [0, 1]$

$$z_1 = b_1 + w_1\prime * x_1 = 0.1 + (0.2 * 0 + 0.4 * 1) = 0.5$$
$$z_2 = b_2 + w_2\prime * x_1 = 0.2 + (0.3 * 0 + 0.3 * 1) = 0.5$$
$$z_3 = b_3 + w_3\prime * x_1 = 0.1 + (0.1 * 0 + 0.5 * 1) = 0.6$$
$$P(Y = j|X = x_2) = \frac{e^{z_j}}{e^{z_1} + e^{z_2} + e^{z_3}}, j = 1, 2, 3$$
$$P(Y = 1|X = x_2) = 0.3220$$
$$P(Y = 2|X = x_2) = 0.3220$$
$$P(Y = 3|X = x_2) = 0.3559$$

For x_1 (true class y_1), the contribution to cost function is:

$$J_1 = -\log(P(Y = 1|X = x_1)) = -\log(0.3197) = 1.1140$$

For x_2 (true class y_2), the contribution to cost function is:

$$J_2 = -\log(P(Y = 2|X = x_2)) = -\log(0.3220) = 1.134$$

The total cost function is $J(w) = \frac{1}{2}(J_1 + J_2) = 1.137$

During the training process, through gradient descent, the model adjusts the weights w to minimize this cost function and improve the prediction accuracy.

2.2.5. Gradient descend

Gradient descent [23] is an iterative optimization algorithm used to minimize a function by iteratively adjusting the values of the parameters of the function in the direction of the negative gradient of the function. It is a widely used optimization technique in machine learning, especially in training neural networks. The basic idea behind gradient descent is to start with some initial values for the parameters of the function and then compute the gradient of the function with respect to each parameter. The gradient tells us the direction of steepest increase of the function at the current point, so taking the negative gradient gives us the direction of steepest decrease. We repeat this process iteratively until we reach a minimum of the function, or until we reach a stopping criterion such as a maximum number of iterations.

2.2.6. Gaussian Naive Bayes

Gaussian Naive Bayes [4] is a probabilistic algorithm used for classification in machine learning. It is based on Bayes' theorem and assumes that the features are independent and normally distributed. Given a set of training data, the algorithm calculates the prior probability of each class based on the frequency of that class in the training data. For each feature in the training data, the algorithm calculates the mean and variance of that feature for each class. This is done separately for each class. To classify a new data point, the algorithm calculates the conditional probability of that data point belonging to each class, based on the mean and variance of each feature for that class. The conditional probability is calculated using the Gaussian distribution. The algorithm then selects the class with the highest conditional probability as the predicted class for the new data point.

2.2.7. KNNs

KNN [4] is a simple, non-parametric algorithm used for classification and regression tasks in machine learning. It works by finding the K closest data points (nearest neighbors) to a given input data point and predicting the class (in classification) or the value (in regression) based on the classes or values of those nearest neighbors. The distance metric used to find *KNNs* can be Euclidean distance, Manhattan distance, or other distance metrics. Once the KNNs are identified, the algorithm uses a majority voting scheme to determine the class of the new data point for classification problems or a weighted average to determine the value for regression problems.

One advantage of KNN is its simplicity and ease of implementation. It can work well with both small and large datasets and can be used for both binary and multi-class classification problems. However, its performance may be affected by the choice of K and the distance metric used. Additionally, KNN may not work with datasets that have many irrelevant features or features with high dimensionality.

2.2.8. Adaptive boosting

Adaptive boosting [24] is a boosting algorithm that combines weak classifiers to form a strong one. It works by iteratively training weak classifiers on the same dataset, with a different weight assigned to each sample in the dataset at each iteration. The weights of correctly classified samples from the previous iteration are increased, and the weights of correctly classified

samples are decreased. In this way, the subsequent weak classifiers focus on the samples that the previous weak classifiers tried to classify. Once all weak classifiers are trained, their predictions are combined using weighted majority or weighted average, depending on whether the task is classification or regression, respectively. The weights of each weak classifier in the final prediction are determined by its performance on the training data.

2.2.9. Pipeline

In machine learning, a pipeline [25] is a sequence of data processing steps that transform raw data into a final prediction model. A pipeline typically includes several stages, such as data preprocessing, feature extraction, model training, hyperparameter tuning, and model evaluation. Each stage of the pipeline takes input from the previous stage and produces output that is used as input for the next stage. The main advantages of using a pipeline in machine learning are as follows:

1) Consistency: A pipeline ensures that the same data preprocessing and feature extraction steps are applied to both the training data and the test data, ensuring consistency between them.
2) Reproducibility: A pipeline allows for easy replication of experiments, as it ensures that the same sequence of processing steps is used every time.
3) Automation: A pipeline automates many of the routine tasks involved in machine learning, such as data preprocessing, feature extraction, and hyperparameter tuning, saving time, and reducing the risk of errors.
4) Modularity: A pipeline allows different stages to be easily swapped out or modified, allowing for rapid experimentation with different approaches.

2.3. Models evaluation and strategy steps

To evaluate the models, we divided the data set into a training subset (train) and a test subset (test), where the size of the test subset was set to 25% (test_size = 0.25). The accuracy metric and the confusion matrix were used for evaluation. Accuracy measures the percentage of correct predictions and is defined as Accuracy = (Correct Positive Predictions + Correct Negative Predictions)/ Sample Size, while the confusion matrix indicates the true values versus the predicted values in a table format, the main diagonal of which has the true predictions, negative and positive [3, 26].

To improve accuracy, the data were transformed (Standarscaler), while the best parameters of the algorithms were searched, such as the maximum depth (max_depth) for the Decision Tree algorithm or the number of KNNs for the KNN algorithm, through the GridSearchCV class. In addition, the cross-validation and bootstrapping methods were applied through the AdaBoostingClassifier class, and all the previous ones in the series were also implemented through the "pipeline" class [3, 25]. The above algorithms were implemented in the Python programming environment.

The strategy followed in this work consisted of the following steps (Figure 1):

1) Collection of a "representative" sample.
2) "Cleaning" of the data (data cleaning/cleansing).
3) Apply transformations to data.
4) Application of bivariate and multivariate correlation analyses.
5) Reduction of mathematical dimensions (data reduction).
6) Prediction with and without Machine Learning methods

3. Research Methodology

3.1. Research design

This study has employed to explore the feasibility of using data analysis methods during the preparatory stage of applying machine learning techniques ("data preprocessing" in machine learning), to enhance their predictive power. Specifically, we examined the prediction of BMI, based on the consumption frequencies of 140 foods by adolescent students in Greece [27]. The statistical software used was Python 3.10 [28] via the Anaconda platform, Jupyter notebook 6.4.5, and IBM SPSS Statistics v26.0.

The methodological approach involves a combination of exploratory data analysis, dimensionality reduction, and model fitting. This includes techniques such as PCA and other multivariate methods to preprocess the data before applying machine learning algorithms. The rationale for these methods is to reduce dimensionality and improve the interpretability of the data, thereby enhancing the predictive accuracy of the machine learning models used in subsequent analyses.

The following research questions will guide this study:

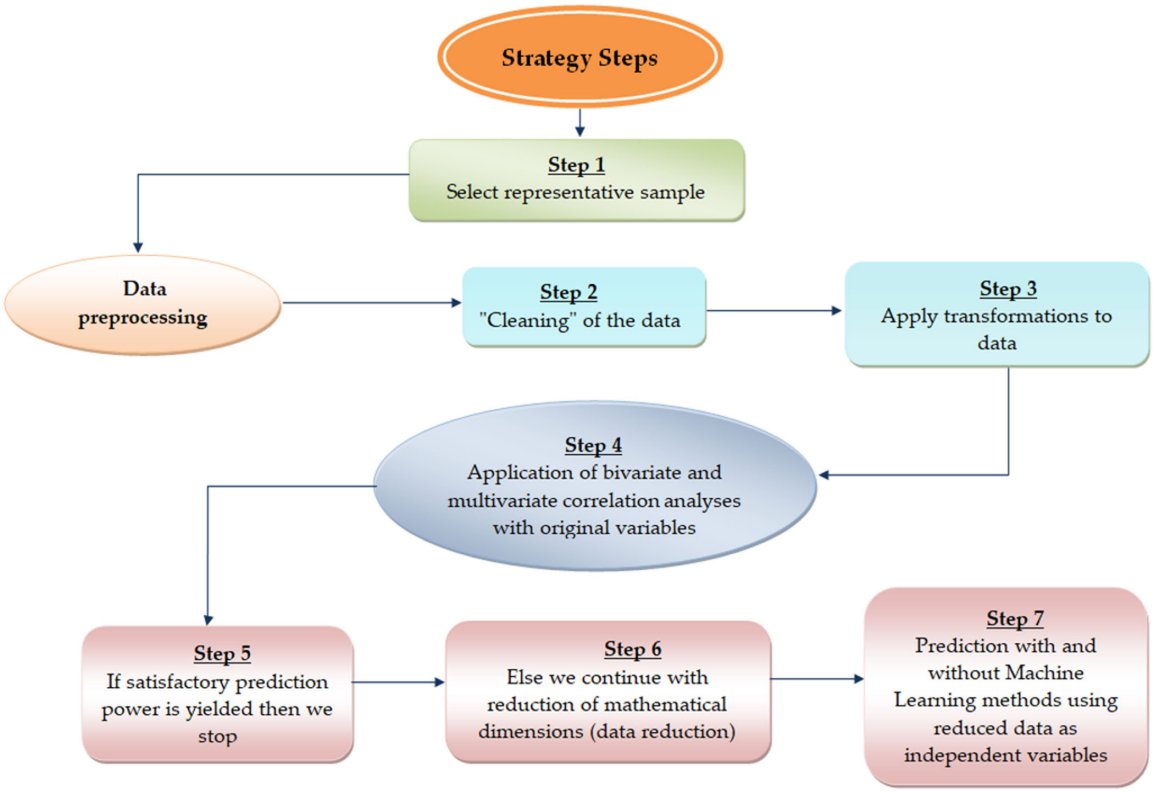
1) Which data analysis method provides the best support for BMI predictions?
2) How does the categorization of BMI affect the performance of machine learning algorithms?

3.2. Participants

The dataset employed in this study consisted of real data obtained through proportional stratified random sampling from high schools in all Greek prefectures. This dataset was gathered during a nationwide epidemiological study on adolescent nutritional habits carried out from 2010 to 2012 by the Department of Nutrition and Dietetics at the Alexandria Technological Educational Institute of Thessaloniki. The study received approval from the Pedagogical Institute and the Ministry of Education, Research, and Religious Affairs. According to the 2011 census, the sample was representative of the entire adolescent population in Greece [27].

The dataset contained information on 42,593 adolescents, aged 12 to 19 years (50.4% boys and 49.6% girls) along with 155 mixed-type variables (characteristics). The BMI was defined as a dependent variable and was employed in analyses as both a quantitative (scale) variable (minimum value = 12.17, maximum value = 55.23) and an ordinal variable with four classes, as per the World Health Organization's recommendations (Underweight: <18.50, Normal weight: 18.50–24.99, Over-weight: 25.00–29.99 and Obese: ≥30.00) [8, 29]. Independent variables included the following: 140 choices of Greek cuisine food items and dishes (frequency of consumption/week) as quantitative (scale) variables; daily hours of sleep, the daily number of glasses of water consumed; weekly consumption of fast food; daily number of meals; weekly frequency of breakfast consumption and weekly delivery frequency (these variables pertain to individual eating habits of adolescents), which were also quantitative variables. Additional independent variables were the weekly family meal frequency, an ordinal variable with four categories (Never = 0, 1–2 times = 1, 3–4 times = 2, Daily = 3), and qualitative (nominal) variables, such as gender (2 categories), prefecture (37 categories), geographical area (3 categories: Urban = 1, Suburban = 2, Rural = 3), family form structure (5 categories: Without parents = 0, With both parents = 1, With one parent due to divorce = 2, With one parent

Figure 1
A general guide of strategy steps for analyzing the specific dataset



due to death = 3, Single parent = 4), fasting 3 categories: No = 0, Sometimes = 1, Yes = 2), delivery (2 categories: No = 0, Yes = 1), (these variables concern the demographic characteristics and habits of teenagers).

3.3. Statistical analysis

A detailed statistical analysis was performed to assess the relationships between the independent variables and the BMI. Initially, exploratory data analysis was conducted to understand the distribution and correlations among variables

The methods chosen in this study were driven by the desire to balance simplicity and complexity at different stages of the analysis. Initially, we employed traditional statistical methods and regression analyses to explore relationships between individual and social characteristics and BMI. Despite the low predictive power, this provided a foundation for understanding the data and served as a baseline for comparison against more advanced models.

We then applied dimensionality reduction techniques to handle the large number of food consumption variables, allowing us to distill this high-dimensional data into a manageable number of factors. Even though these factors did not yield high predictive power in linear regression models, they provided a more structured representation of the data.

Finally, by moving to classification trees and utilizing the CHAID algorithm, we accounted for the non-linear and interaction effects that were likely influencing BMI classification. This method significantly improved predictive accuracy, highlighting the importance of using more sophisticated machine learning approaches when simpler models fail to capture the underlying complexity of the data.

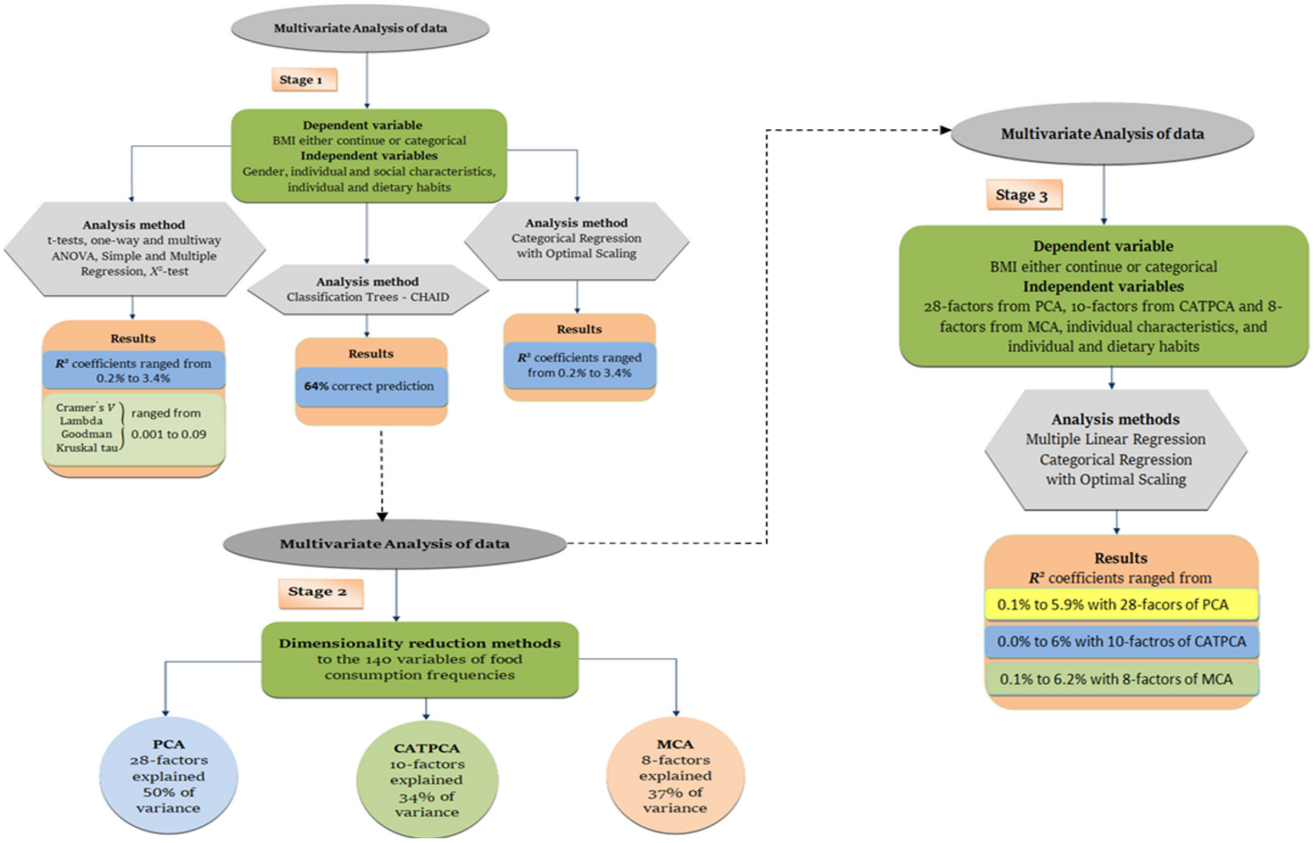
Given the nature of the problem—predicting BMI categories, which is a multi-class classification task—the use of supervised machine learning algorithms is necessary. The specific algorithms chosen provide a balance between interpretability (Decision Trees, Logistic Regression), complexity handling (SVM, Random Forest), and efficiency (GaussianNB, SGD, KNN). Together, these algorithms provide a comprehensive evaluation of the BMI prediction problem, ensuring that both complex and simple models are explored.

3.4. Results

3.4.1. Multivariate analysis of data

In the first stage, we investigated the relationship of the BMI both as a quantitative and as a qualitative (ordinal) dependent variable, with gender, individual and social characteristics, as well as with individual and dietary habits. Various statistical tests were performed including a t-test, one-way and multiway ANOVA, simple and multiple regression with and without selection of variables, and X^2-test. The predictive power of the examined models was found to be too low, with R^2 coefficient values of the general linear models ranging from 0.2% to 3.4% and the Cramer's V or Lambda or Goodman and Kruskal tau coefficients ranging from 0.001 to 0.09. We then applied CATREG with optimal scaling, considering BMI either as a quantitative (scale) or as a qualitative (ordinal categorical) dependent variable, and individual and social characteristics, as well as the individual and eating habits of the adolescents, as independent variables. The predictive power of the models remained low, with R^2 coefficient values ranging from 2.4% to 3.9%. This approach aimed to examine the predictive power of reasonable independent variables

Figure 2
Multivariate analysis methods and summary results



(e.g., gender and other individual and social characteristics of the adolescents) using relatively simple models and statistical methods, hoping that these analyses might eliminate the need for more complex statistical analyses utilizing the data of the 140 choices of food and "dishes" of the Greek cuisine (Figure 2).

In a second stage, we applied dimensionality reduction methods to the 140 variables related to food consumption frequencies. Specifically, we used PCA, CATPCA, and MCA, transforming the quantitative variables into 3 classes (cut-off points 33.3% and 66.6%). From PCA, 28 factors explained 50% of the total inertia, while 68 factors explained 73% of the total inertia. CATPCA identified 10 significant factors explaining 34% of the total inertia, while MCA yielded 8 significant factors, accounting for 37% of the total inertia (Figure 2).

Next, we considered the factor scores of the 28 factors resulting from PCA as independent variables. We applied multiple linear regression with BMI as the quantitative dependent variable, CATREG with BMI as an ordinal dependent variable and factor scores as independent variables, as well as CATREG with BMI as a quantitative or categorical dependent variable and factor scores, individual characteristics, and individual and dietary habits as independent variables. The predictive power of the models remained low with R^2 coefficient values ranging from 0.1% to 5.9% (Figure 2).

Applying similar models to the factor scores of the 10-factor CATPCA and the 8-factor MCA also resulted in low predictive power, with R^2 coefficient values ranging from 0.0% to 6% and from 0.1% to 6.2% respectively (Figure 2).

Despite the low predictive power of the models and the low values of the corresponding effect sizes, it is noteworthy that the application of the Classification Trees—CHAID method, considering BMI in categories as dependent, individual characteristics and habits as independent, resulted in 64% correct prediction (Figure 2).

3.4.2. Accuracy of machine learning algorithms

Lastly, we applied machine learning techniques through classification algorithms [5] in order to predict BMI as a categorical dependent variable (with 4 BMI classes according to FAO), using both the raw original data and the factor scores, resulting from the application of PCA, CATPCA, and MCA, respectively. We used the accuracy measure to test the predictive ability and compare the different methods. Using BMI as categorical variable makes more sense for nutritional assessment. Standardized BMI categories not only provide a world-level framework for evaluating weight status, enabling more explicit comparisons and descriptions of weight distribution within populations, but also facilitate the identification of weight trends and patterns over time [30]. Furthermore, there is considerable evidence that each BMI category is associated with different levels of risk for various chronic conditions, including cardiovascular disease, type 2 diabetes, and certain types of cancer [31]. Thus, the BMI classification system serves as a simplified and practical tool for identifying priority groups for targeted interventions or nutritional counseling [9, 32].

We first applied the SVC algorithm to the raw original data, which resulted in an accuracy of $\alpha = 0.66$. This accuracy was not improved by normalizing the data or by applying the optimal parameters (kernel functions, C regularization, and γ parameter) identified through the Grid Search class.

Next, we applied the SDG algorithm for classification (SGDClassifier). On the raw original data, the evaluation yielded an accuracy of $\alpha = 0.60$, while data normalization improved the accuracy to $\alpha = 0.65$. Cross-validation ($cv = 5$) resulted in an average accuracy of 0.63 with a standard deviation of $s = 0.016$. The application of the GaussianNB resulted in an accuracy of $\alpha = 0.38$, while the KNN algorithm produced an accuracy of 0.60, with an optimal value of $K = 6$, from the Grid Search class.

We then applied the "Decision Tree Classifier" algorithm, where the initial evaluation yielded an accuracy of $\alpha = 0.51$, which was improved by applying the Adaboost algorithm to $\alpha = 0.66$. The assessment was similar ($\alpha = 0.66$) when applying the random forest algorithm for classification. We also searched for the optimal parameters (Gini, Entropy, max_depth), through the GridSearchCV class, identifying entropy and maximum depth max_depth = 5 as the best criteria. Figure 3 displays a decision tree for classifying the data with a depth of 3, highlighting the characteristics used for separation.

The application of the multinomial logistic regression algorithm (Logistic Regression), considering that we had four classes, yielded an accuracy of $\alpha = 0.66$. We then applied all the aforementioned procedures to the raw original data using the automated pipeline class, resulting in the second column (Accuracy) of the table above (Table 1). The remaining three columns of the table (Accuracy PCA_28, Accuracy CATPCA_10, Accuracy MCA_8) were obtained from the application of machine learning algorithms to the factor scores, which were derived from PCA, CATPCA, and MCA, respectively. The results shown in Table 1 were obtained by applying the pipeline code to each case (Figure 4).

The results from the various machine learning algorithms, as presented in Table 1, provide valuable insights into the effectiveness of different data preprocessing methods on predictive performance.

1) **Multinomial Logistic Regression:** The accuracy of **0.66** achieved using the multinomial logistic regression algorithm indicates a solid performance given the complexity of predicting BMI categories across four classes. This accuracy serves as a benchmark for comparison against the results obtained from dimensionality reduction techniques.
2) **SVC:** The SVC maintained an accuracy of **0.66** when applied to both the raw data and the PCA-derived factors. This consistency suggests that SVC is robust to dimensionality reduction and can effectively leverage the reduced feature set without a loss in predictive power.
3) **KNN:** The KNN algorithm exhibited a slightly lower accuracy of **0.60** across all data variations. This could imply that KNN may require more discriminative features or that its performance is

Figure 3
Decision tree with depth = 3 and BMI as the categorical dependent variable, using Gini index as a criterion at each node to split samples into classes

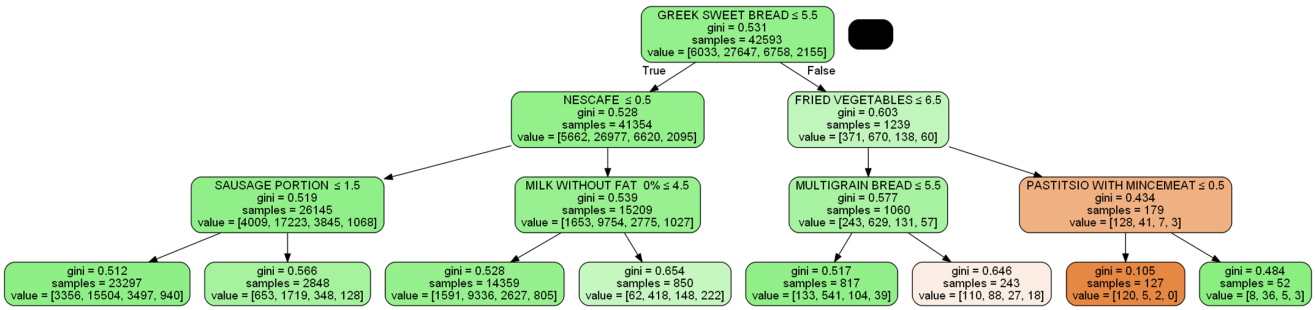


Table 1
Prediction accuracy of BMI as a categorical dependent variable using machine learning algorithms and methods

Algorithm	Accuracy using raw original data[1]	Accuracy PCA_28[2]	Accuracy CATPCA_10[3]	Accuracy MCA_8[4]
Logistic Regression Multinomial	0.66	0.65	0.65	0.65
SVC	0.66	0.66	0.65	0.66
KNN	0.60	0.60	0.61	0.60
Decision Tree	0.52	0.52	0.51	0.53
Random Forest	0.66	0.66	0.67	0.66
SGD	0.64	0.65	0.65	0.65
Naive Bayes	0.38	0.61	0.63	0.63

[1]Raw original data: Application of an algorithm with initial (140 variables related to food consumption frequencies) after preprocessing data
[2]PCA_28: Application of an algorithm with characteristics of the 28 factors resulting from the principal component analysis
[3]CATPCA_10: Application of an algorithm with characteristics of the 10 factors resulting from the categorical principal component analysis
[4]MCA_8: Application of an algorithm with characteristics of the 8 factors obtained from the correspondence analysis

Figure 4
Pipeline code

```
from sklearn.pipeline import make_pipeline
model_pipeline=[]
model_pipeline.append(LogisticRegression(multi_class='multinomial', penalty='none',solver='newton-cg',
max_iter=1000))
model_pipeline.append(SVC())
model_pipeline.append(KNeighborsClassifier())
model_pipeline.append(DecisionTreeClassifier())
model_pipeline.append(RandomForestClassifier())
model_pipeline.append(SGDClassifier())
model_pipeline.append(GaussianNB())

model_list=['LogisticRegression','SVC','KNN','Decision Tree','Random Forest','SGD','Naive Bayes']
acc_list=[]
for model in model_pipeline:
  model.fit(X_train,y_train)
  y_pred=model.predict(X_test)
  acc_list.append(metrics.accuracy_score(y_test,y_pred))

result_df=pd.DataFrame({'Model':model_list,'Accuracy':acc_list})
result_df
```

sensitive to the choice of input features. The negligible difference between the raw data and the PCA/CATPCA/MCA results indicates that while KNN is affected by feature reduction, it does not benefit significantly from it.

4) **Decision Trees:** The performance of the decision tree algorithm was the weakest among the tested models, achieving an accuracy of **0.52** on raw data and remaining relatively unchanged across the factor scores. This suggests that decision trees may struggle with the complexity of the data, possibly due to overfitting or underfitting issues.
5) **Random Forest:** The Random Forest Classifier yielded the highest accuracy of **0.67** when using the factor scores from CATPCA. This result underscores the effectiveness of ensemble methods in capturing complex interactions among variables, indicating that the reduced set of factors from CATPCA enhances the model's predictive ability. The fact that the random forest performs better with fewer features suggests that it may have successfully identified the most informative aspects of the data while mitigating noise.
6) **SGD:** The accuracy achieved with SGD was **0.64**, with minimal changes across different data sets. This consistency suggests that SGD can generalize well across various input configurations, reaffirming its utility in high-dimensional spaces.
7) **Naïve Bayes:** The Naïve Bayes algorithm showed a significant increase in accuracy from **0.38** with raw data to **0.63** when using factor scores from PCA, CATPCA, and MCA. This highlights the advantages of feature reduction for Naïve Bayes, suggesting that the algorithm benefits from a more compact feature set that aligns better with its underlying assumptions of independence among features.

Overall, the results reveal that dimensionality reduction techniques, particularly CATPCA, can enhance the predictive performance of certain algorithms, such as Random Forest and Naïve Bayes. The minor performance variations among the algorithms applied to the original and reduced datasets indicate that many models can handle reduced dimensionality without significant loss in accuracy, although some, like Naïve Bayes, notably improve. These findings suggest that implementing preprocessing steps like PCA, CATPCA, and MCA could lead to more efficient models while maintaining or enhancing prediction accuracy.

Additionally, the lack of significant performance deterioration among most algorithms implies that reducing the feature space may simplify the model training process and potentially improve interpretability without sacrificing predictive power.

4. Discussion

The study results indicate that the initial multivariate analyses, conducted to determine if some individual characteristics of adolescents (e.g., demographic characteristics, habits) could have higher predictive power than dietary characteristics, demonstrated very low predictive power in each case. Similar results were obtained when the analyses were conducted with the PCA, MCA, and CATPCA factor scores as independent variables derived from the 140 variables related to food consumption frequencies. Interestingly, the CHAID method emerged as a notable exception, offering relatively good prediction accuracy. CHAID's ability to capture interactions between variables may explain its superior performance in this context, suggesting that non-linear relationships in the data could provide valuable insights that traditional linear methods overlook. This observation highlights the need for incorporating a wider variety of analytical techniques when exploring complex health-related data.

Consequently, we proceeded to apply the machine learning methods to both the raw original data (140 variables related to food consumption frequencies as prediction variables) and the factor scores obtained from PCA, MCA, and CATPCA as prediction variables, with the target variable in each case being the BMI (categorical variable with four classes).

From the analysis, it was determined that for the specific data set, the application of the algorithms (SVC, KNN, SGD, Naive Bayes, Decision Tree Classifier, Random Forest Classifier, Logistic Regression Multinomial) to data with reduced dimensions yielded similar results and in some cases better than when applied to raw original data. Additionally, the prediction is more reliable when using BMI as the dependent variable as a qualitative variable with four (4) classes.

In general, designing with a data analysis strategy helps save time and chooses the best forecasting model. Dimensionality

reduction, if it does not improve the predictive ability of the models, at least contributes to the "interpretability" of the results. This is because the factors, which obtained from the PCA, the MCA, and the non-linear CATPCA (28, 8, and 10, respectively), all had a natural interpretation within the theoretical framework of the study. As a result, the 140 variables can be represented by a smaller number of components or by a smaller number of new complex and most importantly, "interpretable" variables. This approach illustrates how complex datasets can be reduced into fewer, more interpretable variables, enhancing both the clarity of the findings and their practical applications in public health.

Therefore, it is suggested to attempt dimensionality reduction using various methods before applying machine learning methods. Also, the small values of the R^2 coefficients, obtained from the samples examined in the preparatory stage (correlation analyses, *t*-tests, one-way ANOVA, Simple and multiple regression, X^2-tests), indicate the necessity to check both the data quality and to "clean" the data (data cleaning/cleansing), before applying any method. This is provided that the variables used in both the preparatory stage and the prediction models and algorithms are representative and describe as completely as possible the phenomenon-system under consideration.

5. Conclusion

The strategy proposed in this study involves first collecting a representative sample and checking the quality of the data, cleaning them (missing values, outliers), and coding and transforming them appropriately for subsequent analysis. Next, it is recommended to apply ordinary bivariate and multivariate analyses, aiming to find potential characteristics of the sample that largely explain the phenomenon under consideration. This approach was followed in order to examine the predictive power of reasonable independent variables using relatively simple models and statistical methods, with the hope that these analyses will eliminate the need for more complicated statistical analyses utilizing the data of the 140 choices of food and "dishes" of the Greek cuisine.

If no single characteristics are found that largely explain the phenomenon under consideration, it is then proposed to reduce the dimensions with various multivariate methods and apply prediction methods to the new resulting variables (factorial scores), using both statistical methods and machine learning algorithms, in order to find the best prediction method or algorithm.

Our findings underscore that a well-structured data analysis strategy not only saves time but also aids in selecting the most suitable prediction model. Moreover, while dimensionality reduction may not always enhance predictive accuracy, it significantly contributes to the interpretability of the results, facilitating a clearer understanding of the underlying relationships within the data. The importance of employing a diverse range of analysis methods and machine learning algorithms is evident, as this variety is crucial for either discovering a satisfactory dimensionality reduction solution or enhancing the robustness of the results.

Ultimately, this study suggests that employing dimensionality reduction techniques prior to the application of machine learning methods can lead to improved model performance and interpretability. It also highlights the critical importance of data quality and the need for rigorous data-cleaning processes to enhance the validity of predictive analytics in adolescent health research.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Nikolaos Papafilippou: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Zacharenia Kyrana:** Resources. **Emmanouil Pratsinakis:** Conceptualization. **Efstratios Kiranas:** Conceptualization, Resources. **Alexandra-Maria Michailidou:** Resources. **Angelos Markos:** Conceptualization, Methodology, Software, Formal analysis, **George Menexes:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – review & editing, Supervision, Project administration.

References

[1] Le Roux, B., & Rouanet, H. (2004). *Geometric data analysis: From correspondence analysis to structured data analysis*. Germany: Springer.

[2] Menexes, G. (2006). *Experimental designs in data analysis.* Doctoral Dissertation, University of Macedonia.

[3] Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8), 832.

[4] Mahesh, B. (2020). Machine learning algorithms: A review. *International Journal of Science and Research*, *9*, 381–386.

[5] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160.

[6] Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018,* 99–111.

[7] Paul, R., & Das, K. N. (2022). Trends of optimization algorithms from supervised learning perspective. *Journal of Computational and Cognitive Engineering*, *3*(4), 447–461.

[8] World Health Organisation. (2021). *Obesity related diseases*. Retrieved from: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

[9] Musa, F., Basaky, F., & Osaghae, E. O. (2022). Obesity prediction using machine learning techniques. *Journal of Applied Artificial Intelligence*, *3*(1), 24–33.

[10] Anderson, T. W. (1984). *An introduction to multivariate statistical analysis*. USA: John Wiley & Sons.

[11] Greenacre, M., Groenen, P. J. F., Hastie, T., d'Enza, A. I., Markos, A., & Tuzhilina, E. (2022). Principal component analysis. *Nature Reviews Methods Primers*, *2*(1), 100.

[12] Michailidis, G., & De Leeuw, J. (1998). The Gifi system of descriptive multivariate analysis. *Statistical Science*, *13*(4), 307–336. https://doi.org/10.1214/ss/1028905828

[13] Bond, J., & Michailidis, G. (1996). Homogeneity analysis in Xlisp-Stat. *Journal of Statistical Software*, *1*(i02), 12–32. https://doi.org/10.18637/jss.v001.i02

[14] Agresti, A., & Kateri, M. (2021). *Foundations of statistics for data scientists: With R and Python*. UK: Chapman and Hall/CRC.

[15] Amalita, N., Fahira, A. A., Arnellis, A., & Fitria, D. (2023). Analysis graduates in getting a job using the CHAID methods. In *AIP Conference Proceedings*, *61*, 020004. https://doi.org/10.1063/5.0122664

[16] Mpia, H. N., Syasimwa, L. M., & Muyisa, D. M. (2024). Comparative machine learning models for predicting loan fructification in a semi-urban area. *Archives of Advanced Engineering Science*, 1–11. https://doi.org/10.47852/bonviewaaes42022418

[17] Papafilippou, N., Kyrana, Z., Pratsinakis, E., Markos, A., & Menexes, G. (2024). Using data analytics methods before using machine learning algorithms: Prediction on mixed data. *Data Analysis Bulletin*, *20*(1), 32–44. https://doi.org/10.1007/s12652-020-02741-3

[18] Wang, H., & Liu, B. (2021). Customer loan risk prediction based on decision tree. In *2nd International Conference on Machine Learning and Computer Application*, 1–5.

[19] Tangirala, S. (2020). Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, *11*(2), 612–619.

[20] Parmar, A., Katariya, R., & Patel, V. (2019). A review on random forest: An ensemble classifier. In *International Conference on Intelligent Data Communication Technologies and Internet of Things 2018*, 758–763.

[21] Ayua, S. I. (2024). Random forest ensemble machine learning model for early detection and prediction of weight category. *Journal of Data Science and Intelligent Systems*, *2*(4), 233–240.

[22] Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. USA: Apress.

[23] Ighalo, J. O., Adeniyi, A. G., & Marques, G. (2020). Application of linear regression algorithm and stochastic gradient descent in a machine-learning environment for predicting biomass higher heating value. *Biofuels, Bioproducts and Biorefining*, *14*(6), 1286–1295.

[24] Wang, W., & Sun, D. (2021). The improved AdaBoost algorithms for imbalanced data classification. *Information Sciences*, *563*, 358–374.

[25] Mohr, F., Wever, M., Tornede, A., & Hüllermeier, E. (2021). Predicting machine learning pipeline runtimes in the context of automated machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 3055–3066.

[26] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: An overview. *arXiv Preprint: 2008.05756*.

[27] Grammatikopoulou, M. G., Poulimeneas, D., Gounitsioti, I. S., Gerothanasi, K., Tsigga, M., Kiranas, E., & ADONUT Study Group. (2014). Prevalence of simple and abdominal obesity in Greek adolescents: The ADONUT study. *Clinical Obesity*, *4*(6), 303–308.

[28] Eidelman, A. (2020). Python data science handbook by jake VANDERPLAS (2016). *Statistique et Société*, *8*(2), 45–47.

[29] Sommer, I., Teufer, B., Szelag, M., Nussbaumer-Streit, B., Titscher, V., Klerings, I., & Gartlehner, G. (2020). The performance of anthropometric tools to determine obesity: A systematic review and meta-analysis. *Scientific Reports*, *10*(1), 1–12. https://doi.org/10.1038/s41598-020-69498-7

[30] Estivaleti, J. M., Guzman-Habinger, J., Lobos, J., Azeredo, C. M., Claro, R., Ferrari, G., …, & Rezende, L. F. (2022). Time trends and projected obesity epidemic in Brazilian adults between 2006 and 2030. *Scientific Reports*, *12*(1), 12699. https://doi.org/10.1038/s41598-022-16934-5

[31] Grier, T., Canham-Chervak, M., Sharp, M., & Jones, B. H. (2015). Does body mass index misclassify physically active young men? *Preventive Medicine Reports*, *2*, 483–487. https://doi.org/10.1016/j.pmedr.2015.06.003

[32] Finkelstein, E. A., Brown, D. S., Wrage, L. A., Allaire, B. T., & Hoerger, T. J. (2010). Individual and aggregate years-of-life-lost associated with overweight and obesity. *Obesity*, *18*(2), 333–339. https://doi.org/10.1038/oby.2009.253

How to Cite: Papafilippou, N., Kyrana, Z., Pratsinakis, E., Kiranas, E., Michailidou, A.-M., Markos, A., & Menexes, G. (2025). Integrating Data Analysis Methods with Machine Learning Algorithms for Mixed Data Types: Does This Combination Improve Predictive Models' Accuracy?. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS52023906