International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 14, No. 3, December 2025, pp. 1136~1145
ISSN: 2252-8776, DOI: 10.11591/ijict.v14i3.pp1136-1145

The bootstrap procedure for selecting the number of principal components in PCA

Borislava Toleva
Faculty of Economics and Business Administration, Sofia University, Sofia, Bulgaria

Article Info

Article history:

Received Nov 6, 2024
Revised Jun 20, 2025
Accepted Jul 1, 2025

Keywords:

ANOVA
Bootstrap
Decision trees
Principal component analysis
Support vector machines

ABSTRACT

The initial step in determining the number of principal components for both classification and regression involves evaluating how much each component contributes to the total variance in the data. Based on this analysis, a subset of components that explains the highest percentage of variance is typically selected. However, multiple valid combinations may exist, and the final choice is often made manually by the researcher. This study introduces a novel yet straightforward algorithm for the automatic selection of the number of principal components. By integrating ANOVA and bootstrapping with principal component analysis (PCA), the proposed method enables automatic component selection in classification tasks. The algorithm is evaluated using three publicly available datasets and applied with both decision tree and support vector machine (SVM) classifiers. Results indicate that this automated procedure not only eliminates researcher bias in selecting components but also improves classification accuracy. Unlike traditional methods, it selects a single optimal combination of principal components without manual intervention, offering a new and efficient approach to PCA-based model development.

Corresponding Author:

Borislava Toleva
Faculty of Economics and Business Administration, Sofia University
Tsrarigradsko Shosse, 125, Sofia, Bulgaria
Email: vrigazova@uni-sofia.bg

1. INTRODUCTION

Principal component analysis (PCA) is a widely used technique for dimensionality reduction and data exploration [1]. It transforms the original dataset into a set of linearly uncorrelated variables known as principal components. These components represent linear combinations of the original variables and are constructed to capture the maximum variance within the data. The core objective of PCA is to identify a subset of principal components that best captures the most informative structure of the dataset, thereby enabling more effective regression or classification modeling [1]. A central challenge in applying PCA lies in determining the optimal number of components to retain [1]. For decades, both academics and practitioners have relied on a standard approach to address this question [1], [2]. This traditional method involves transforming the data into principal components and evaluating how much variance is explained by various combinations—e.g., the first two, three, or four components. The subset that accounts for the highest cumulative variance is typically selected. However, in practice, it is often unclear whether to retain three, four, or five components, as the incremental gain in explained variance may be marginal and difficult to interpret [1].

Many researchers have tried to solve this issue by modifying the PCA equation. The focus of their research is how PCA can be used as a feature selection technique. For example, by modifying the equation of

the principal components analysis as in [3]. They propose unweighted version of the PCA combined with variable selection to avoid the issue of how many components to choose. Prieto-Moreno *et al.* [4] introduces "separability measure between multiple failures" to select the number of principal components. He uses discriminant information contained in the PCs to select the right combination. Sharifzadeh *et al.* [5] proposed a sparse PCA method known as SSPCA, designed for data pre-processing and dimensionality reduction. This variant of PCA introduces modifications to the computation of eigenvectors and eigenvalues, aiming to enhance performance in large datasets. The approach is particularly effective in scenarios where high-dimensional noise—arising from a large number of variables—needs to be minimized.

However, a central issue is that when PCA is used as a feature selection, the final set of principal components should be converted back to the original features [1]. Unlike other feature selection algorithms, the conversion is hard for interpretation as the direct link between the linear combination of principal components and the original features is not straightforward [1]. Therefore, PCA is used as a dimensionality reduction technique but rarely as a method to select concrete features. Regardless of this, the appropriate selection of principal components is a key for the success of the classification model. Therefore, researchers aim to find an unbiased and straightforward selection of principal components. For instance, Gajjar *et al.* [6] proposes a novel method to select non-zero loadings in sparse PCA instead of using eigenvalues and eigenvectors as it is in the standard PCA [1].

In 2021, Rahoma *et al.* [7] introduced a novel method for estimating loading factors in PCA. While their algorithm shares similarities with the approach proposed by Gajaar *et al.* [6] particularly in its focus on loading factors—it differs in the bootstrap techniques used to assess the distributional properties of the elements within the loading vectors. These elements are then leveraged to construct a sparse loading structure for PCA. Based on their findings, Rahoma *et al.* [7] proposed two new PCA variants: Bootstrap SPCA and Sparse IPCA, both of which rely on bootstrap-based resampling. Although these methods represent advancements in PCA, none of them provide an automated solution for selecting the number of principal components—a critical yet unresolved issue in many applications. This research addresses that gap by proposing a fully automatic algorithm for principal component selection. For instance, Pacheco *et al.* [8] outlines a multi-step variable selection process using PCA but explicitly avoids the core question of determining the optimal number of principal components.

An important yet underexplored research direction involves leveraging the textbook PCA approach for automatic selection of the number of principal components—without altering the core PCA equations. This paper focuses on advancing this line of inquiry and contributes in several key ways. First, we propose a novel algorithm that automatically selects a single optimal combination of principal components using the standard PCA framework [1]. Unlike other methods that modify PCA computations or rely on subjective judgment, our approach adheres strictly to the textbook method while automating the component selection process. Second, we expand on previous research [9] by demonstrating the effectiveness of the bootstrap procedure in PCA beyond its application in logistic regression. While earlier work showed that bootstrapping could guide component selection for logistic models, this study extends those findings to support vector machines (SVM) and decision tree classifiers, showing similar benefits in classification performance. Third, this work contributes to the broader exploration of bootstrap methods in machine learning, outside their traditional statistical applications. Our previous research established the bootstrap as a viable alternative to cross-validation in classification problems [10]. This study introduces a new application of bootstrapping: aiding the automatic selection of the number of principal components [11], [12]. The proposed methodology offers several advantages: it is simple to implement, computationally efficient, and easy to interpret, making it practical for real-world data analysis and machine learning tasks. Next section describes the algorithm proposed, while section 3 elaborates our findings.

2. RESEARCH METHOD

In this section, we present both the classical PCA algorithm, as described in standard textbooks [13], and our proposed algorithm. We use the classical method as a baseline to highlight its limitations and to compare its performance against our automated approach. The classical PCA procedure is implemented in Python 3.6 using built-in functions. Following the classical steps outlined in [10], [13], we apply PCA but adapt the classification stage by using decision tree classifiers and SVMs instead of logistic regression. All model parameters are kept at their default values in Python, with the SVM using C=1 and an RBF kernel. The classical PCA procedure includes the following steps [1], [13]:

- Data standardization: Standardize the input data [14] and transform them into principal components
- Variance analysis: Analyze eigenvalues [15] and eigenvectors [16] to determine the proportion of variance each principal component explains. Calculate the cumulative variance explained by the first n components.

- Component selection: Select the subset of components that explains the highest cumulative variance [17]. This method differs from other approaches that use criteria such as AIC or BIC for component selection [18].
- Model training and evaluation: Use the selected components to train and test classification models (decision trees and SVMs).

It is important to note that, in this classical approach, the selected components are always the first n components in order of their index. This is based solely on the decreasing proportion of variance explained by each successive component. The component selection process is independent of the classification model used.

Although the classical PCA algorithm [19] is simple and widely applied, it often presents a critical limitation: multiple valid combinations of components may explain similar amounts of variance [20]. The classical method does not provide automated means to resolve this ambiguity. Researchers are left to select a combination manually, relying on prior domain knowledge, literature guidelines, or ad hoc heuristics [1]. In many cases, however, such knowledge or rules are not available, and the lack of a clear criterion can introduce subjectivity or bias into the analysis [21], [22].

To overcome this issue, we propose a novel method called ANOVA-Bootstrapped-PCA [9], which automates the selection of the optimal number of principal components within a standard PCA framework. This method extends our previous work that applied bootstrap-based component selection in the context of logistic regression. In this study, we demonstrate its applicability to decision tree classifiers and SVM. We refer to the two implementations as:

– ANOVA-Bootstrapped-PCA-DT: using decision tree classifiers
– ANOVA-Bootstrapped-PCA-SVM: using SVM

he algorithm utilizes existing Python functions, including SVC(), DecisionTreeClassifier(), PCA() from sklearn.decomposition, and Pipeline from sklearn, to incorporate ANOVA feature selection. Additionally, we developed a custom script to implement the tenfold bootstrap procedure, originally introduced in our prior study [10]. While earlier bootstrap studies [11], [12] focused on its use as a resampling technique in statistical analysis, they did not explore its potential benefits in machine learning. Our previous work [10] addressed this gap by demonstrating how the bootstrap can be adapted to classification tasks. We now further extend this by integrating bootstrap into PCA for automated component selection. The ANOVA-Bootstrapped-PCA algorithm proceeds through the following steps:

a) Standardization: Standardize the input data (as in the classical approach).
b) PCA transformation: Apply PCA to the standardized data.
c) Normalization: Normalize the resulting principal components to the [0, 1] range to eliminate negative values.
d) ANOVA ranking: Perform ANOVA to rank the principal components by importance. Unlike the classical method, components are selected based on ANOVA ranking rather than index order, and the ranking remains independent of the classification model.
e) Percentile Grouping: Divide the components into percentiles (10%, 20%, ..., 100%), where each percentile contains the top n components based on ANOVA rankings.
f) Bootstrapped resampling: For each percentile, split the data into training and testing sets using a 70/30 ratio, repeated via the tenfold bootstrap [10].
g) Model training and evaluation: For each percentile group, train and evaluate ten models (both SVM and decision tree classifiers). Calculate the average accuracy and classification scores across bootstrap samples.
h) Component selection: Identify the percentile (i.e., component combination) that yields the highest classification performance. This defines the optimal number of components for each model.

3. RESULTS AND DISCUSSION

To conduct the experiments, we use three publicly available datasets [23]-[25]. We define X and Y variables, where Y is a target variable that represents categories. As PCA is conducted only on independent variables, the target variable Y is excluded from the experiments. All results presented in section 3 relate to the connection among the X variables as each principal component forms a linear combination of features that contains as much information about the data as possible. The aim is to find the most informative set of principal components by discovering the set of principal components with the highest variance [16]. Therefore, the classical approach produces a table, where the percentage of variance explained of each principal component is calculated (% of var explained). The most informative set of principal components consists of the first principal components, which contribute the most to the total variance explained. This criterion is referred to as ‘cumulative percentage explained’. However, when the total variance explained for

two or more sets of principal components is similar, selecting the correct number of principal components may not be straightforward. On the other hand, the proposed approach in this paper eliminates the involvement of the researcher as it provides an automatic selection of the number of principal components regardless of the total variance explained. The results from the two approaches are summarized by dataset in the next subsections.

3.1. The ED dataset [23]

Table 1 contains the output from the classical approach that calculates the contribution of each principal component to the variance explained and the cumulative percentage explained for the first n number of principal components. The percentage of variance explained is calculated by eigenvalues and eigenvectors [16].

Table 1. Principal components number according to the classical approach

	% of var explained		var explained, cumulatively %
PC 1	57.5%	PC1+PC2	97.4%
PC 2	39.9%	PC1+PC2+PC3	99.9%
PC 3	2.5%	PC1+PC2+PC3+PC4	100%
PC 4	0%	PC1+PC2+PC3+PC4+PC5	100%
PC 5	0%		

Source: author's research

Table 1 shows that the first principal component contributes the most to the data variance (57.9%), followed by the second (39.9%) and the third. The first two principal components together account for 97.4% of the variance in data, while the first three–99.9%. The contribution of the fourth and fifth principal component is too small to be considered. In this case, the book rule [16] advises to select the combination that results in the highest cumulative variance explained. This would be the first three principal components. When we use the first three principal components to run the SVMs with RBF kernel, the model achieves 96.9% accuracy. The decision tree classifier achieves 98.2%. However, the aim of the PCA is to perform dimensionality reduction [16]. Given that the first two principal components account for 97.4% of the variability in data and the very small contribution of the third principal component, another researcher may select the first two principal components. In this case, a smaller number of principal components would be selected, while the variance explained would be high enough. The example of the ed dataset demonstrates that in some cases more than one principal component combination is possible. In the case of the ed dataset selecting two or three principal components would not affect the outcome of the model significantly due to its small number of components. However, the issue of how many principal components to select and avoid the manual selection is very important in dataset with many principal components.

To achieve an automatic selection of the number of principal components, we propose the ANOVA-Bootstrapped-PCA classification. In this algorithm, the importance of the principal components is first calculated using ANOVA. Similarly, to the classical algorithm, their importance does not change with the classification model used. Table 2 summarizes the importance of the principal components in the ed dataset.

Table 2. Importance of the principal components according to the new proposed approach

PC	Importance
PC1	17.1677
PC2	8.70988
PC3	3550.92
PC4	41.9038
PC5	8.95789

Source: author's calculations

According to Table 2, the most important principal components are the third one, the fourth and the first one. An impotant highlight is that this outcome is different from the classical approach. The classical approach identifies the first n most importance principal components, where the first always contributes the most, and the second is second in order. However, the newly proposed approach observes the importance of each principal component separately and their importance does not depend on their place in the dataset. The importance of each principal component remains the same regardless of the classification model used. Table 3 shows how many principal components are selected using the ANOVA-Bootstrapped classification algorithm when the SVM and the decision tree classifier are fitted.

Table 3. Number of principal components selected using the ANOVA-bootstrapped classification in SVM and DT

Percentile	Number of PCs	Accuracy of DT	Accuracy SVM
10%	0.5	96.5%	97.4%
20%	1	96.5%	97.4%
30%	1.5	98.3%	98.2%
40%	2	98.3%	98.2%
50%	2.5	98.3%	98.2%
60%	3	99.2%	97.5%
70%	3.5	99.2%	97.5%
80%	4	99.2%	97.6%
90%	4.5	99.2%	97.6%
100%	5	99.3%	96.8%

Source: author's calculation

In the case of the decision tree classifier, the three most important principal components that are selected are the fourth, third and first (Table 3). Using this combination, the decision three classifier achieves the highest accuracy of 99.2%, while retaining the smallest possible combination of principal components (performing dimensionality reduction). Although the accuracy of 99.2% can also be achieved by adding the fifth principal component (as it is the fourth most important), this combination would use more principal components than optimal for dimensionality reduction. Therefore, selecting the fourth, third and first principal components are the best combination for achieving the highest accuracy in the decision tree classifier.

Using the proposed algorithm, the case for the SVM is different. Table 3 shows that the highest accuracy (98.2%) for the SVM model can be achieved using only 2 principal components, these being the third and the fourth (Table 2). The third and the fourth principal components are the best selection for the SVM for three reasons. First, they are the most important ones according to Table 2. Second, they produce the highest accuracy for the SVM. Third, the SVM accuracy using the two and three most important principal components is similar. Therefore, the third most important principal components does not add additional information to the model. This result differs from the classical approach. In the classical approach the number of principal components selected is the same regardless of the classification model used. However, our approach selects the number of principal components that would produce the best accuracy given the classification model used. Our algorithm can be used to select the combination of principal components that would improve the model's performance. For instance, the classical approach resulted in 96.9% accuracy from the SVM using the first 3 principal components. The ANOVA-bootstrapped-PCA SVM achieved 98.2% accuracy using only the first two most important principal components. The algorithm we propose improved the accuracy resulting from the classical PCA SVM by 1.3% and it performed dimensionality reduction better as it uses only 2 principal components. Therefore, the proposed algorithm can be used not only to automatically select the number of principal components, but it also improves the performance of the model and perform dimensionality reduction better. Similar case is observed with the decision tree classifier, where the classical PCA approach resulted in 98.2% accuracy using three principal components, while the proposed algorithm achieved 99.2% accuracy using 3 principal components (Table 3). In the example of the decision tree classifier, the proposed algorithm provides an automatic selection of three principal components, eliminating the choice between the first two and three principal components that is offered by the classical approach. Also, the necessary principal components are automatically selected using the ANOVA-bootstrapped-PCA algorithm.

3.2. The food dataset [24]

Similar results can be observed in the food dataset. Table 4 summarizes the contribution of each principal component to the total variability of data and the cumulative contribution according to the classical approach.

Table 4. Classical approach in the food dataset

PC	% of var explained		Var explained, cumulative %
PC 1	41%	PC1+PC2	63.0%
PC 2	22%	PC1+PC2+PC3	79%
PC 3	16%	PC1+PC2+PC3+PC4	93%
PC 4	14%	PC1+PC2+PC3+PC4+PC5	100%
PC5	7%		

Source: author's calculations

The output in Table 4 does not change with the classification model used. The number of principal components selected based on Table 4 is the one to be used in all classification models. Table 4 shows that the first four principal components have a significant contribution to the variability of data accounting for 93% of total variance in data. However, if the first three principal components are selected, then only 79% of the variability of data would be explained. In this case, the answer is straightforward, so the first four principal components should be selected. Selecting the first three would lead to a significant loss of important information. Table 4 demonstrates that in the case of the food dataset. The selection of principal components following the classical approach is obvious. However, the dimensionality reduction is not effective as only one principal component should be removed from the classification model according to the classical approach. Therefore, in complex models a better set of principal components might be the first three but that would come at the cost of some loss in data information. Therefore, the researcher should decide whether to use the first three or first four principal components depending on the purpose of their task.

Another disadvantage of the classical approach is that the researcher does not know whether a straightforward selection of principal components would be possible before running the classical algorithm. This makes the classical approach inefficient as it can lead to time consuming decisions and manual selection in big datasets. Also, bias can be introduced in the model in cases where the decision about the number and set of principal components should involve the researcher. The newly proposed approach, however, ranks the importance of principal components, while producing a table based on which identifying the combination that results in the highest accuracy is possible. This leads to an automatic and straightforward selection of principal components. Table 5 demonstrates the importance of the principal components resulting from the ANOVA step in our algorithm for the food dataset.

Table 5. Principal components importance according to the new approach for the food dataset

PC	Importance
PC1	708.444
PC2	677.15
PC3	1202.37
PC4	35.5811
PC5	1.17576

Source: author's calculation

According to Table 5, for the food dataset the first most important principal component is the third one, then the first, second, fourth and fifth. The combination of principal components that should be used for the SVM and DT according to the proposed approach is demosntrated in Table 6.

Table 6. Results from the ANOVA-Bootstrapped PCA classification

Percentile	Number of features	Accuracy of DT	Accuracy of SVM
10%	0.5	78.8%	86.2%
20%	1	78.8%	86.2%
30%	1.5	81.8%	86.2%
40%	2	81.8%	86.2%
50%	2.5	81.8%	86.2%
60%	3	83.4%	86.2%
70%	3.5	83.4%	86.2%
80%	4	83.9%	86.2%
90%	4.5	83.9%	86.2%
100%	5	83.5%	86.2%

Source: author's calculations

The proposed algorithm achieves accuracy of 83.9% from the DT using the four most important principal components in the food dataset, while the classical approach–accuracy of 83.2% using the first four principal components. The ANOVA-bootstrapped-PCA algorithm produces higher accuracy when used with the four most important variables based on the ANOVA ranking. As Tables 4 and 5 demonstrate the sets of four principal components for the decision tree classifier vary in the two approaches. The classical approach uses the first four principal components that have the highes total variance explained (Table 3), while the proposed algorithm uses the four most important principal components based on their ANOVA score in Table 4 the SVMs, on the other hand, results in 86.2% accuracy when the proposed algorithm is applied (Table 6). Table 6 shows that regardless of the number of principal components used, the accuracy achieved by the proposed SVM model is 86.2%. Therefore, fitting SVM withonly 1 principal component results in the

best accuracy according to the ANOVA-bootstrapped-PCA apporach while the classical approach achieves 90% using 4 principal components. The reason for this discrepancy is that the food dataset has imbalanced classes, so the proposed approach with SVM is not appropriate in this case. Class 1 cannot be correctly predicted as it has very few observations. This is not the case for the decision tree classifier, which accounts for the imbalanced classes and predicts both classes correctly. However, the prediction of class 1 in the proposed methodology is worse than in the classical approach. This can be seen in Table 7 that compares the classification metrics from the SVM and DT resulting from the classical approach and the proposed new approach. Table 7(a) shows that the classic PCA SVM predicts correctly about 90.5% of class 0 and 81.3% of class 1 despite the imbalanced classes. However, this is not the case in the ANOVA-Bootstrapped-PCA SVM as it predicts well the predominant class 0 but fails to predict the minority class 1. Table 7(b) shows that the classic PCA decision tree correctly predicts class 1 in about 39% of the cases, which is similar to the proposed approach. The decision tree classifier in both cases gives similar measures despite the class imbalance.

Table 7. Classification metrics of the (a) SVM and (b) DT resulting from the classical approach and the proposed algorithm

(a)

Classic PCA SVM	Precision	Recall	f1-score	Support
0	90.5%	98.6%	94.4%	20624
1	81.3%	36.5%	50.3%	3347
avg/total	89.2%	90.0%	88.3%	23971
ANOVA BOOTSTRAPPED PCA SVM	precision	recall	f1-score	support
0	86.2%	100.0%	92.6%	10256
1	0.0%	0.0%	0.0%	1645
avg/total	74.3%	86.2%	79.8%	11901

(b)

Classic PCA Class	Precision	Recall	f1-score	Support
0	91%	89%	90%	20624
1	39%	43%	41%	3347
avg/total	83%	83%	83%	23971
ANOVA BOOTSTRAPPED PCA DT				
0	89%	89%	89%	10226
1	34%	33%	34%	1669
avg/total	82%	82%	82%	11901

Source: author's calculations

In the case of imbalanced data, we do not recommend using the proposed approach with SVM. Further research should be conducted to explore the performance of the proposed algorithm on imbalanced data and other classification models that cannot compensate for imbalanced classes. The decision tree classifier, however, is appropriate to use with the proposed ANOVA-bootstrapped-PCA in imbalanced datasets.

3.3. The fraud dataset [25]

The Fraud dataset has 5 principal components. Its classes are relatively balanced as was in the Ed dataset. Table 8 shows the contribution of each principal component to the total variance of data according to the classical approach. The first three principal components account for 96.7% percent of the variability of data. They are used to fit the SVM and DT classifiers.

The classical PCA results in 96.7% accuracy when the decision tree classifier is fitted using the first three principal components (Table 8) and in 75.8% when the SVM classifier is fitted with the same principal components like in the decision tree classifier. As Table 9 shows the proposed ANOVA-Bootstrapped-PCA SVM and DT perform better than the classical approach.

As Table 9 shows the bootstrapped PCA decision tree achieved 98.1% accuracy using 3 principal components (the second, third and fourth as Table 10 shows), which is 1.4% p.p. higher than the classical

approach. While the SVM resulted in accuracy of 76.9% using 2 principal components, which is by 1.1 p.p. higher than the classic PCA SVM approach. The classification scores are similar for the classic and proposed approach as it is in the ed dataset.

When classes are balanced, the ANOVA-Bootstrapped PCA classification can select the number of principal components automatically and in many cases improve the accuracy of the model. As Tables 3, 6, and 9 demonstrate the proposed algorithm can also be used to compare the performance of different classification models using different numbers of principal components. A decision not only about the number of principal components but also about what model to use can be made. The proposed algorithm can also be used for model selection.

The proposed algorithm is a novel approach to selecting the number of principal components for classification. The ANOVA-Bootstrapped-PCA classification algorithm provides a fast and effective way to select the number of principal components and improve the accuracy of the model. It can also be used for model selection as the performance of several classification models can be compared. Based on the accuracy and number of principal components selected, one classification model can be selected over another one. However, the algorithm performs well only in datasets with balanced classes. In case of imbalanced data, the ANOVA-Bootstrapped-PCA algorithm works well with the decision tree classifier. The decision tree classifier handles the imbalance in classes, therefore allowing the ANOVA-Bootstrapped-PCA algorithm to be competitive to the classical PCA approach. The ANOVA-Bootstrapped-PCA decision tree classifier offers automatic selection of principal components, unlike the classical approach. Despite this advantage, the decision tree classifier is not appropriate in all cases, so the ANOVA-Bootstrapped-PCA decision tree classifier cannot be applied in all cases with imbalanced data. How the ANOVA-Bootstrapped-PCA Classification can handle class imbalance is a topic of further research.

Table 8. Principal components selected according to the classical approach

PC	% of var explained		var explained
PC 1	53%	PC1+PC2	83%
PC 2	30%	PC1+PC2+PC3	100%
PC 3	16%		
PC 4	0%		
PC 5	0%		

Source: author's calculations

Table 9. Results from the proposed approach

Percentile	Number of features	Accuracy of DT	Accuracy of SVM
10%	0.4	83.92%	76.78%
20%	0.8	83.92%	76.78%
30%	1.2	83.92%	76.78%
40%	1.6	97.95%	76.85%
50%	2	97.95%	76.85%
60%	2.4	97.95%	76.85%
70%	2.8	98.14%	75.98%
80%	3.2	98.14%	75.98%
90%	3.6	98.14%	75.98%
100%	4	98.45%	75.85%

Source: author's calculations

Table 10. Importance of principal components according to the proposed approach

PC	Importance
PC1	1.06356
PC2	560.478
PC3	126.382
PC4	3.58504

Source: author's calculations

4. CONCLUSION

This research develops a simple algorithm for automatic detection of the number of principal components in classification models. The advantages of the proposed algorithm include straightforward selection of principal components, model selection when necessary and improved model performance. Unlike the classical principal components analysis, the researcher can have a better overview of the model's performance given each combination of principal components, as well compare the model's performance

with the same principal components but different classification model. The ANOVA-Bootstrapped-PCA classification performs both principal components selection and model selection. Improvement of model's accuracy is also an advantage of the proposed model. In conclusion, we recommend the proposed algorithm in cases of balanced-class datasets and if possible, the ANOVA-Bootstrapped PCA decision tree classifier in case of imbalanced classes.

FUNDING INFORMATION

This research has not received any funding.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Borislava Toleva	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The author declares no conflict of interest.

DATA AVAILABILITY

All datasets used are freely available. Links are provided in the references.

REFERENCES

[1] G. James, D.Witten, T. Hastie., and T. R., "An introduction to statistical learning," *Springer*, 2013.
[2] S. Salata and C. Grillenzoni, "A spatial evaluation of multifunctional ecosystem service networks using principal component analysis: A case of study in Turin, Italy," *Ecological Indicators*, vol. 127, p. 107758, Aug. 2021, doi: 10.1016/j.ecolind.2021.107758.
[3] S. B. Kim and P. Rattakorn, "Unsupervised feature selection using weighted principal components," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5704–5710, May 2011, doi: 10.1016/j.eswa.2010.10.063.
[4] A. Prieto-Moreno, O. Llanes-Santiago, and E. García-Moreno, "Principal components selection for dimensionality reduction using discriminant information applied to fault diagnosis," *Journal of Process Control*, vol. 33, pp. 14–24, Sep. 2015, doi: 10.1016/j.jprocont.2015.06.003.
[5] S. Sharifzadeh, A. Ghodsi, L. H. Clemmensen, and B. K. Ersbøll, "Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 168–177, Oct. 2017, doi: 10.1016/j.engappai.2017.07.004.
[6] S. Gajjar, M. Kulahci, and A. Palazoglu, "Selection of non-zero loadings in sparse principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 162, pp. 160–171, Mar. 2017, doi: 10.1016/j.chemolab.2017.01.018.
[7] A. Rahoma, S. Imtiaz, and S. Ahmed, "Sparse principal component analysis using bootstrap method," *Chemical Engineering Science*, vol. 246, p. 116890, Dec. 2021, doi: 10.1016/j.ces.2021.116890.
[8] J. Pacheco, S. Casado, and S. Porras, "Exact methods for variable selection in principal component analysis: Guide functions and pre-selection," *Computational Statistics and Data Analysis*, vol. 57, no. 1, pp. 95–111, Jan. 2013, doi: 10.1016/j.csda.2012.06.014.
[9] B. Vrigazova, "Novel approach to choosing principal components number in logistic regression," *ENTRENOVA - ENTerprise REsearch InNOVAtion*, vol. 7, no. 1, pp. 1–12, Dec. 2021, doi: 10.54820/pucr5250.
[10] B. Vrigazova and I. Ivanov, "Tenfold bootstrap procedure for support vector machines," *Computer Science*, vol. 21, no. 2, pp. 241–257, Apr. 2020, doi: 10.7494/csci.2020.21.2.3634.
[11] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, no. 4, pp. 373–384, Nov. 1995, doi: 10.1080/00401706.1995.10484371.
[12] B. Efron, "Bootstrap methods: another look at the Jackknife," *The Annals of Statistics*, vol. 7, no. 1, Jan. 2007, doi: 10.1214/aos/1176344552.
[13] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An introduction to statistical learning second edition," *Springer Texts*, vol. 102. The R Foundation, p. 618, Jul. 2006, doi: 10.32614/cran.package.islr2.
[14] M. Gal and D. L. Rubinfeld, "Data standardization," *SSRN Electronic Journal*, 2018, doi: 10.2139/ssrn.3326377.
[15] H. Robert Frost, "Eigenvectors from eigenvalues sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 31, no. 2, pp. 486–501, Nov. 2022, doi: 10.1080/10618600.2021.1987254.

[16] S. Kalaivani, K. Sivakumar, and S. Balamuralitharan, "Higher order principal component analysis of eigen values with special structures covariance matrices," in *AIP Conference Proceedings*, 2020, vol. 2277, p. 150002, doi: 10.1063/5.0025241.
[17] E. Saccenti and J. Camacho, "Determining the number of components in principal components analysis: A comparison of statistical, crossvalidation and approximated methods," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 99–116, Dec. 2015, doi: 10.1016/j.chemolab.2015.10.006.
[18] S. L. Sclove, "Using model selection criteria to choose the number of principal components," *Journal of Statistical Theory and Applications*, vol. 20, no. 3, pp. 450–461, Sep. 2021, doi: 10.1007/s44199-021-00002-4.
[19] Q. Dong, X. Chen, and B. Huang, "Principal component analysis," *In Woodhead Publishing Series in Civil and Structural Engineering, Data Analysis in Pavement Engineering*, Elsevier, 2024, pp. 263-273, doi: 10.1016/B978-0-443-15928-2.00007-0.
[20] E. Henrique Novotny, "Principal component analysis, a more intuitive viewpoint," in *Chemometrics: Data Treatment and Applications*, Elsevier, 2024, pp. 511–523.
[21] J. Jiang, "Principal component analysis, applied multivariate statistical analysis in medicine," *Academic Press*, 2024, pp. 265-301, doi: 10.1016/B978-0-443-23587-0.00008-1.
[22] I. Si-Ahmed, L. Hamdad, C. J. Agonkoui, Y. Kande, and S. Dabo-Niang, "Principal component analysis of multivariate spatial functional data," *Big Data Research*, vol. 39, p. 100504, Feb. 2025, doi: 10.1016/j.bdr.2024.100504.
[23] "EPICA dome C ice core 800KYr Temperature estimates dataset." https://vincentarelbundock.github.io/Rdatasets/datasets.html.
[24] "BudgetFood." https://vincentarelbundock.github.io/Rdatasets/datasets.html.
[25] "Financial fraud dataset." https://www.kaggle.com/code/jeevankishore/synthetic-financial-datasets-for-fraud-detection/data.

BIOGRAPHIES OF AUTHORS

Borislava Toleva is a Ph.D. in Data science at Sofia University, Bulgaria. She obtained a master's degree in Statistics, financial econometrics and actuarial studies in 2015 after a bachelor's degree in Economics at the same university. Her research areas include practical applications of machine learning algorithms for prediction and how their performance can be boosted. Also, applications of big data techniques to small datasets in the field of economics as alternative to traditional econometrics theory. She challenges traditional econometric modelling techniques used to find connections among variables from institutional economics by combining feature selection methods and big data prediction models. As a result, new applications of machine learning techniques to economic data appear. She can be contacted at email: vrigazova@uni-sofia.bg.