Classification and analysis of the MNIST dataset using PCA and SVM algorithms

Mokhaled N. A. Al-Hamadani

University of Debrecen, Doctoral School of Informatics,
Department of Data Science and Visualization, Debrecen, Hungary;
Northern Technical University, Technical Institute/Alhawija,
Department of Electronic Techniques, Adan, Kirkuk, Republic of Iraq,
e-mail: mokhaled_hwj@ntu.edu.iq,
ORCID iD: https://orcid.org/0000-0002-7042-3178

DOI: 10.5937/vojtehg71-42689; https://doi.org/10.5937/vojtehg71-42689

FIELD: mathematics, computer sciences, IT
ARTICLE TYPE: original scientific paper

Abstract

Introduction/purpose: The utilization of machine learning methods has become indispensable in analyzing large-scale, complex data in contemporary data-driven environments, with a diverse range of applications from optimizing business operations to advancing scientific research. Despite the potential for insight and innovation presented by these voluminous datasets, they pose significant challenges in areas such as data quality and structure, necessitating the implementation of effective management strategies. Machine learning techniques have emerged as essential tools in identifying and mitigating these challenges and developing viable solutions to address them. The MNIST dataset represents a prominent example of a widely-used dataset in this field, renowned for its expansive collection of handwritten numerical digits, and frequently employed in tasks such as classification and analysis, as demonstrated in the present study.

Methods: This study employed the MNIST dataset to investigate various statistical techniques, including the Principal Components Analysis (PCA) algorithm implemented using the Python programming language. Additionally, Support Vector Machine (SVM) models were applied to both linear and non-linear classification problems to assess the accuracy of the model.

Results: The results of the present study indicate that while the PCA technique is effective for dimensionality reduction, it may not be as effective for visualization purposes. Moreover, the findings demonstrate that both *linear and non-linear SVM models were capable of effectively classifying the dataset.*

Conclusion: The findings of the study demonstrate that SVM can serve as an efficacious technique for addressing classification problems.

Keywords: statistical analysis, machine learning, SVM, PCA, classification.

Introduction

Classification and data analysis are central to the discipline of machine learning, as they enable the effective comprehension of data (Suthaharan, 2014; Wang et al, 2019; Subasi, 2020; Ahmed et al, 2022). In this study, statistical methods were utilized to gain a deeper understanding of the data. Two machine learning techniques, namely the Support Vector Machine (SVM) and the Principal Components Analysis (PCA), were employed on the MNIST dataset, the largest collection of handwritten digit images used for classification problems (LeCun, 2023). The MNIST dataset consists of 70,000 handwritten digits, divided into 60,000 training images and 10,000 testing images (Al-Hamadani, 2015). Simple statistical techniques were applied to the MNIST dataset to improve the knowledge about the data. The PCA, an unsupervised machine learning technique, was utilized for visualization purposes, utilizing two principal components for plotting (Abdi & Williams, 2010). The SVM, a supervised machine learning technique, was utilized in linear and nonlinear models to classify the MNIST dataset into classes (Suthaharan, 2016).

The paper is organized as follows: Section 2 provides an overview of the dataset, programming language, and statistical techniques employed. In Section 3, the PCA algorithm utilized for visualization is explained, as well as the determination of the number of principal components for dimensionality reduction. The SVM linear and nonlinear models are discussed in Section 4. Finally, conclusions and future work are presented in Section 5.

Research method

Dataset

The MNIST (Modified National Institute of Standards and Technology) dataset is the largest and most well-known handwritten digits dataset for recognition and classification problems in machine learning (LeCun et al, 1995). This dataset has been taken from Yann LeCun website (LeCun, 2023). The MNIST dataset consists of 70,000 images of handwritten digits, each labeled with a number from 0 to 9. These images were collected by the National Institute of Standards and Technology (NIST) and split into two parts (Nielsen, 2019). The first part includes 60,000 images which were provided by 250 individuals, including employees of the US Census Bureau and high school students. These images were normalized and centered in (28 *28) = 784 gray scale pixels per image. Each pixel is represented by a value between 0 and 255, with 0 being black, 255 being white, and any value in between representing a shade of gray. The second part of the MNIST dataset includes 10,000 images for testing, which were also provided by 250 different individuals to properly evaluate the performance of the algorithms trained on the dataset.

Program

Python has gained widespread popularity among data scientists due to its simplicity and ease of use in coding (Hao & Ho, 2019). Python is a widely-used programming language that supports various programming styles such as object-oriented, functional, and imperative programming (Raschka et al, 2020). Nowadays, most of machine learning complex problems, or AI in general, are being solved by using Python. Python can be written and developed using various Integrated Development Environments (IDEs), including PyCharm and Google Colab. For this paper, we have used both IDEs for building models over the MNIST dataset.

Visualizing and analyzing the MNIST dataset

To better understand the MNIST dataset, we need to analyze and visualize it. We started by examining the training and testing parts of the dataset to see if the data is balanced in terms of understanding the number of classes and their attributes. Also, we checked the dimensionality of each part of the dataset. Additionally, we plotted some samples from the training dataset to visualize the digits included in the MNIST dataset. Furthermore, we used the describe () method to calculate some statistical data such as mean, standard deviation, and percentile of the numerical values in the data frame. Table 1 shows the dimensionality of the training and testing dataset.

In Table 2, we have computed various statistical measures. One of them is the standard deviation which indicates how spread out the data in a dataset is from the mean.

The mean is the average of a set of numbers and can be found by adding all the numbers together and dividing by the number of values. In the table, the mean of the 10 classes (ranging from class 0 to class 9) is 4.45.

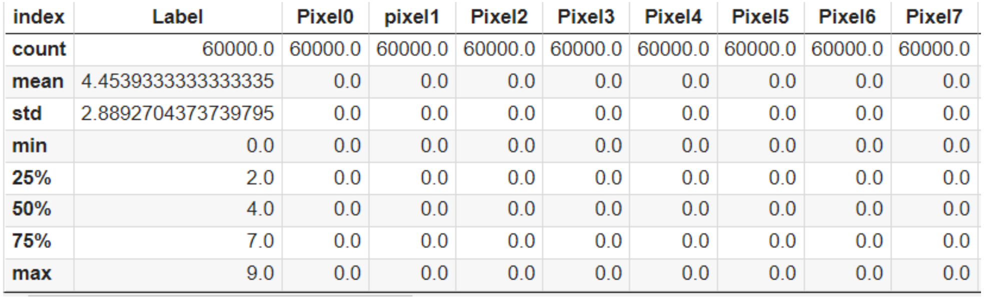
Additionally, the minimum and maximum values in each column are shown in Table 2.

Lastly, we have calculated the quantiles which divide the dataset into groups containing an equal number of data points.

Table 1 – Dimensionality of the MNIST training and testing dataset
Таблица 1 – Размерность набора данных для обучения и тестирования MNIST
Табела 1 – Димензионалност скупа података за увежбавање и тестирање МНИСТ

No.	Dataset	Dimensions (Shape)
1	Training Data	(60000, 784)
2	Training Labels	(60000, 1)
3	Testing Data	(10000, 784)
4	Testing Labels	(10000, 1)

Table 2 – Statistical distribution of the MNIST training dataset
Таблица 2 – Статистическое распределение набора данных для обучения MNIST
Табела 2 – Статистичка расподела скупа података за увежбавање МНИСТ



index	Label	Pixel0	pixel1	Pixel2	Pixel3	Pixel4	Pixel5	Pixel6	Pixel7
count	60000.0	60000.0	60000.0	60000.0	60000.0	60000.0	60000.0	60000.0	60000.0
mean	4.4539333333333335	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	2.8892704373739795	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

To continue with the visualization part, we have checked the number of observations for each class in the MNIST dataset by plotting randomly some samples from the dataset, as the following:

MNIST training dataset

It contains 60,000 images of handwritten digits (ranging from 0 to 9). Figure 1 shows the ratio of the training set. At the same time, Table 3 shows the number of the observations with the corresponding digits.

We observed that the training set is balanced because each class has 6,000 images approximately.

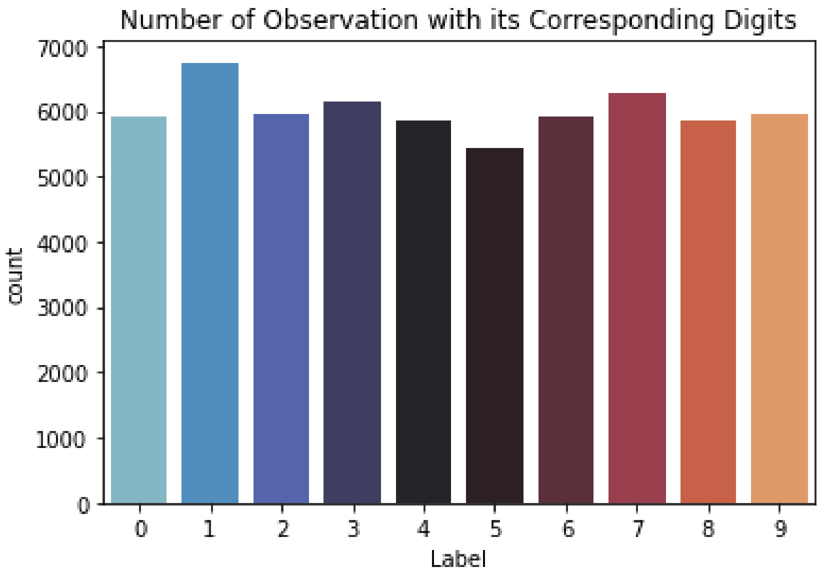


Figure 1 – Number of observations for each digit-training set
Рис. 1 – Количество распознаваний по каждой цифре в учебном наборе
Слика 1 – Број опсервација за сваку цифру у сету за увежбавање

Table 3 – Number of observations in the MNIST training set
Таблица 3 – Количество распознований в обучающей выборке MNIST
Табела 3 – Број опсервација у сету за увежбавање МНИСТ

MNIST Training Dataset Classes	Number of Observations
0	5923
1	6742
2	5958
3	6131
4	5842
5	5420
6	5918
7	6265
8	5851
9	5949

MNIST testing dataset

It contains 10,000 images of handwritten digits (ranging from 0 to 9). Figure 2 shows the ratio of the training set. At the same time, Table 4 shows the number of the observations with the corresponding digits. We observed that the training set is balanced because each class has 1,000 images approximately.

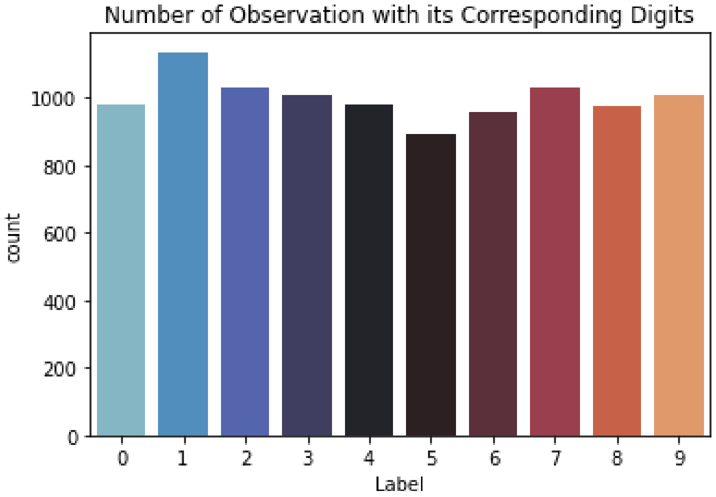


Figure 2 – Number of observations for each digit-testing set
Рис. 2 – Количество распознаваний по каждой цифре в тестовом наборе
Слика 2 – Број опсервација за сваку цифру у сету за тестирање

Table 4 – Number of observations in the MNIST testing set
Таблица 4 – Количество распознаваний в тестовом наборе MNIST
Табела 4 – Број опсервација у сету за тестирање МНИСТ

MNIST Training Dataset Classes	**Number of Observations**
0	980
1	1135
2	1032
3	1010
4	982
5	892
6	958
7	1028
8	974
9	1009

Samples plotting from the MNIST dataset

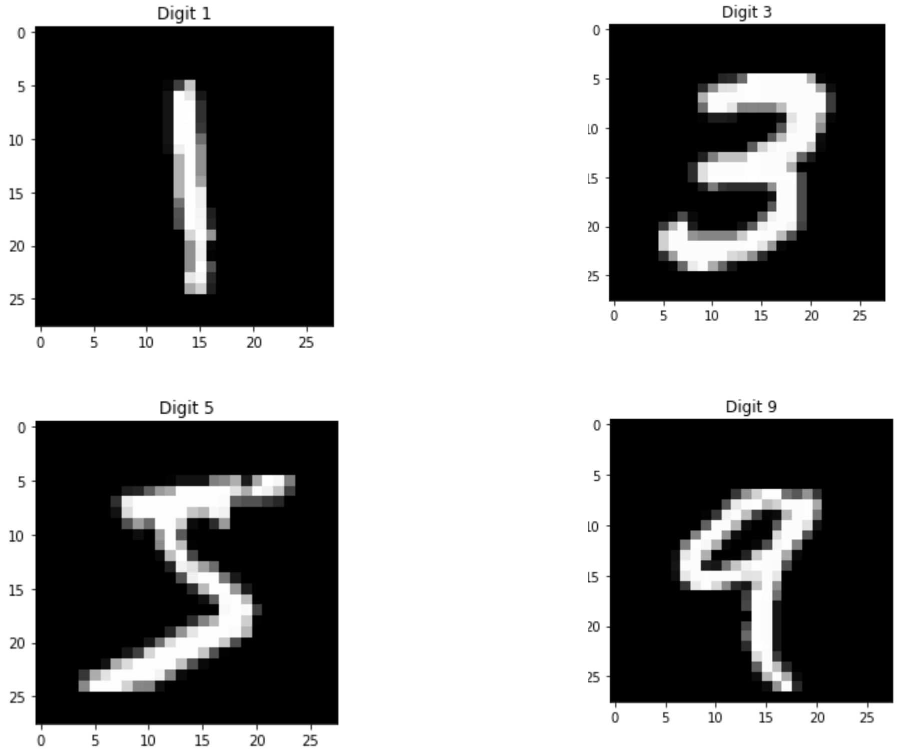


Figure 3 – Plotting samples from the MNIST dataset
Рис. 3 – Формирование выборок из набора данных MNIST
Слика 3 – Узорци добијени плотовањем за скуп података МНИСТ

Principal Components Analysis (PCA)

The PCA is unsupervised machine learning and statistical technique that is used to examine relationships between multiple variables in a dataset (Abdi & Williams, 2010). It aims to extract important information from the data by finding a new set of variables, known as principal components. These variables can describe the data more efficiently. These principal components are orthogonal (uncorrelated) and are ranked by their ability to explain the variance in the data. The PCA relies on the decomposition of positive semi-definite matrices using eigenvalues and eigenvectors, as well as the decomposition of rectangular matrices using singular values and singular vectors (Mishra et al, 2017). Therefore, the PCA is a statistical technique used for dimensionality reduction and visualization of high-dimensional datasets such as, in our case, the MNIST dataset. Since each digit in the MNIST dataset consists of 784 features (dimensions) or pixels, the PCA reduces this number into a smaller number of dimensions while keeping as much of the variance in the data. This method is useful in terms of reducing the computational cost of training machine learning techniques, reducing the problem of overfitting in the data, and making the data easy for plotting in 2D or 3D plots.

To determine the principal components of a dataset, the PCA performs the following steps as shown in Figure 4:

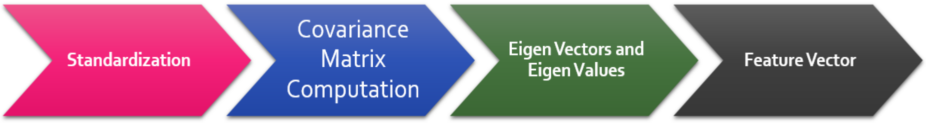


Figure 4 – PCA processing flowchart
Рис. 4 – Блок-схема обработки PCA
Слика 4 –Ток процесирања ПЦА

Standardization

This step is known as standardization and is often referred to as a preprocessing step. Standardization involves transforming the data so that each variable has a mean of zero and a standard deviation of one. This helps to ensure that all of the variables are on a similar scale and have similar variances, which can improve the accuracy of the PCA analysis. Standardization can be done as follows:

$$S = \frac{Variable\ values - mean}{Standard\ deviation}$$

We have used the whole MNIST training dataset which contains 60,000 handwritten digits. For this purpose, we used this method (**standardized_data = StandardScaler().fit_transform(data)).**

Covariance matrix computation

A covariance matrix is a square matrix that shows the correlation between any two or more elements (attributes) in a multidimensional dataset.

$$\begin{bmatrix} \mathrm{Var}(x_1) & \cdots\cdots & \mathrm{Cov}(x_n,x_1) \\ \vdots & . & \vdots \\ \vdots & \quad . & \vdots \\ \mathrm{Cov}(x_n,x_1) & \cdots\cdots & \mathrm{Var}(x_n) \end{bmatrix}$$

The correlation between the attributes can be either positive covariance or negative covariance. Positive covariance means that increasing the value of one variable is associated with an increase in the value of some other variable and vice versa. Negative covariance means that increasing the value of one variable is associated with a decrease in the value of some other variable. Therefore, we calculate the covariance matrix which is equal to (A^T * A) using this method **(covar_matrix = np.matmul(sample_data.T , sample_data)).**

Eigenvectors and eigenvalues

Eigenvectors and eigenvalues are mathematical concepts that are used in linear algebra, and they are closely related to the diagonalization of matrices. Given a square matrix A, an eigenvector of A is a non-zero vector v such that $Av = \lambda v$, where λ is a scalar value known as the eigenvalue associated with the eigenvector v. The eigenvectors of a matrix are often used to represent the "directions" of the maximum variance in a dataset and are used as the principal components. Therefore, finding the eigenvector with the highest eigenvalue is the principal component of the dataset. To project the data onto a 2D space, we can find the top two eigenvalues and the corresponding eigenvectors, and use these to transform the data.

Feature vector

Feature vectors are constructed by selecting the top eigenvectors from the covariance matrix of the original dataset (Manshor et al, 2011). The eigenvectors are determined by calculating the eigenvalues and eigenvectors of the covariance matrix, which capture the most important features or variations of the data. These eigenvectors are then sorted by eigenvalues in a descending order to determine which ones represent the most important features. The final feature vector can be in this form:

$$v = (eig_1, eig_2, eig_3, \ldots\ldots, eig_n)$$

Plotting

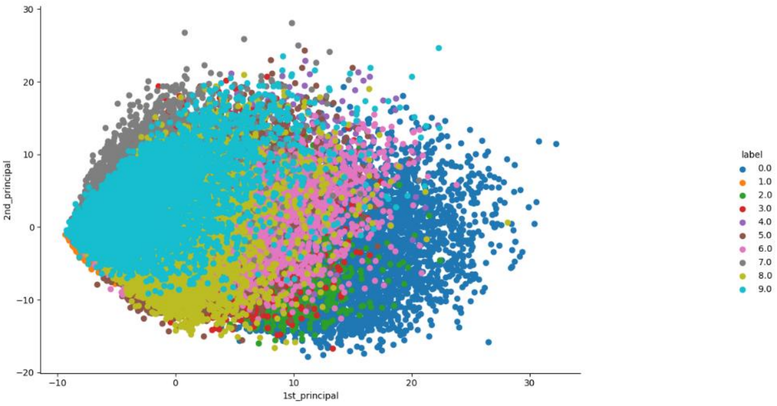


Figure 5 – 2D data points visualization
Рис. 5 –Визуализация точек 2D данных
Слика 5 – Визуализација тачака 2Д података

The overlapping of classes in Figure 5 suggests that the PCA is not the most effective method for visualizing high-dimensional data, as it is unable to clearly distinguish between different classes. In such cases, it may be more effective to use other visualization techniques such as t-SNE, UMAP or non-linear dimensionality reduction techniques, which are more effective in visualizing high-dimensional datasets.

To determine the number of principal components that we keep for the purpose of visualization in our work, we take into consideration what we really want from the PCA in order to deduce the right value of the number of components. We applied the following methods:

1. Since we use the PCA for the purpose of visualization, we need to select 2 or 3 principal components in order to plot them as 2D or 3D. Therefore, we convert high dimensional data into 2-dimensional data to visualize it in a 2D plot. This is shown in Figure 5.

2. We can create a scree plot which is the visual representation of eigenvalues that define the magnitude of eigenvectors (principal components). Figure 6-A shows the explained variance ratio for each component. Figure 6-B shows the eigenvalues for each component on the y-axis and the number of components on the x-axis. By these two plots, we can determine the number of principal components we have to keep.

Note: We used the PCA for visualization only in this paper.

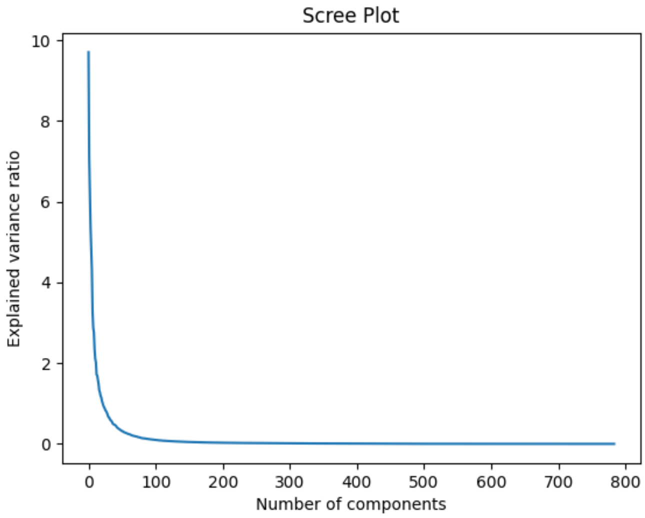


Figure 6a – Scree plot of the explained variance ratio for each component
Рис. 6a – График осыпи объясненной дисперсии ро каждому компоненту
Слика 6a – Scree plot објашњене варијансе за сваку компоненту

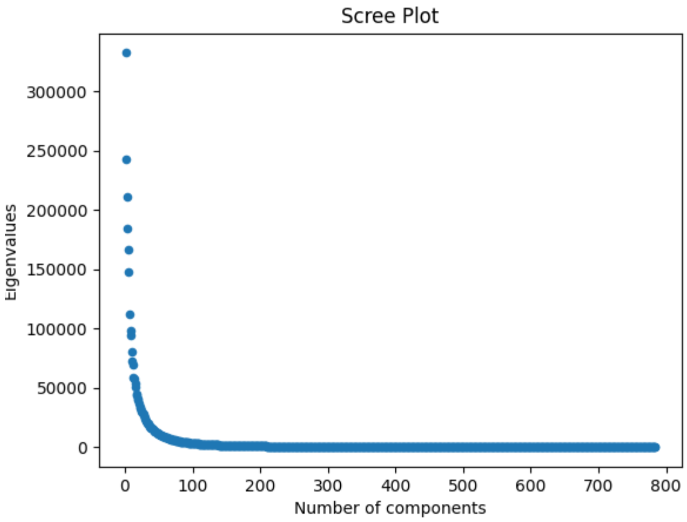


Figure 6b – Scree plot of the eigenvalues for each component
Рис. 6б – График осыпи собственных значений каждого компонента
Слика 6б – Scree plot сопствених вредности за сваку компоненту

Dimensionality reduction

We applied the Principal Component Analysis (PCA) for dimensionality reduction. As shown in Figure 7, the first 300 components explain almost 90% of the variance in the data. Based on this, we can reduce the dimensions of the dataset by selecting only the most important components, which will preserve a high level of variance while also significantly reducing the complexity of the data. This can be useful for tasks such as classification or clustering, where a high-dimensional dataset may be more difficult to work with.

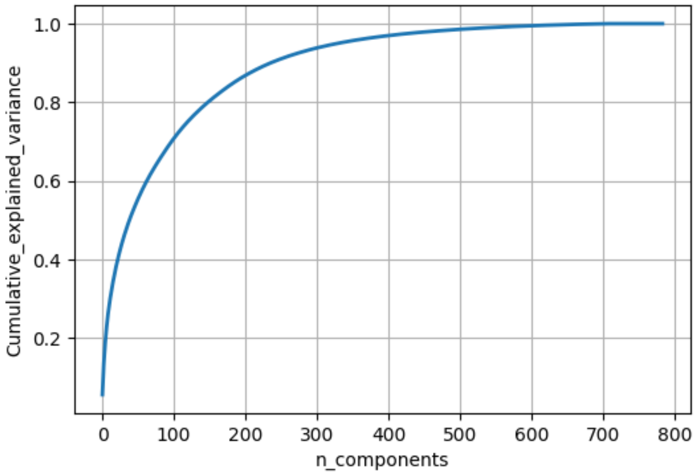


Figure 7 – Cumulative sum of variance with the components
Рис. 7 – Совокупная сумма дисперсии с компонентами
Слика 7 – Кумулативни збир варијансе са компонентама

Classifier

Support Vector Machine (SVM)

The Support Vector Machine is a well-known supervised machine learning technique that can be applied to solve big data classification and regression problems (Suthaharan, 2016). The SVM is relatively robust to overfitting that occurs in other machine learning algorithms, especially when using a nonlinear kernel (Guenther & Schonlau, 2016). There are two main types of SVM models which are linear and nonlinear (Saputra et al, 2022). The SVM is called linear when data classes can be separated by a linear line or a hyperplane using the simple mathematical model $y = wx + b$ where y is the response (dependent) variable, x is the predictor (independent) variable, w is the estimated slope, and b is the estimated intercept. The optimal hyperplane can be selected by finding the hyperplane with the maximum margin which is the distance between the nearest data points of different classes. This is in case the problem can be solved by a linear hyperplane. If a problem cannot be solved linearly, then the data can be separated into different classes using a linear boundary in a transformed space called the feature space. This case is called the nonlinear SVM which uses the mathematical model $y = w\emptyset(x) + b$ where the $\emptyset$ is the kernel function (Suthaharan, 2016).

Since the SVM is a binary classification method,it is used to support classification for two classes. In our case, we used the SVM algorithm for multiclass classification. In multiclass classification, the SVM approach mainly involves splitting the dataset into several binary classification sub-datasets and then fitting a separate binary classifier for each one. There are two different types of multiclass classification for this SVM approach which are one versus one (OvO) and one versus the rest or all (OvR).

The OvO method is splitting the Multiclass dataset the data into Multiple binary classification problems. This approach splits the dataset into one dataset for each class versus every other class. The formula for calculating the number of binary datasets is: (Num_Classes * (Num_Classes – 1)) / 2. For the MNIST dataset is (10 *(10-1)/2)=45 binary classification problems.

The OvR method is also splitting the multiclass dataset into multiple binary classification problems. This approach trains a binary SVM classifier for each class, where the class is treated as a positive class and all other classes are combined into a single negative class. For example, class No (0) versus all the other classes combined as one class.

In this paper, we used the SVC (Support Vector Classification) method which implements the “one-versus-one” approach for multi-class classification.

For our MNIST dataset, we build linear and nonlinear SVM models to check the accuracy of these models. For the linear SVM model, we build it with its default hyperparameters to check the accuracy using the confusion matrix. The confusion matrix provides a decision that has been collected in training and testing which contains the actual labels with the predicted ones (Saputra et al, 2022). Accuracy can be calculated based on this equation:

$$Accuracy = \frac{TP + TN}{(Tp + TN + FP + FN)}$$

where true positive (TP) refers to the data that has been correctly classified as positive, true negative (TN) refers to the data that has been correctly classified as negative while false positive (FP) refers to the data that has been incorrectly classified as positive and false negative (FN) refers to the data that has been incorrectly classified as negative.

We build the nonlinear SVM with its randomly chosen hyperparameters using the RBF kernel (Specifying the kernel type since it has several types such as “linear”, RBF, which is the radial basis function, “poly”, which is polynomial kernel function, “Sigmoid”) and it is used to take the data as an input and transform it into a required form of processing data. We used the RBF kernel because it is a commonly used kernel function due to its ability to perform well in classification tasks, as well as due to its flexibility since it does not require prior information about the dataset to be used effectively. When C=1, it is the regularization parameter which controls the trade-off between maximizing the margin and minimizing the training error in the model. When the value of the regularization parameter C is large, the model prioritizes correctly classifying all training points over having a larger margin. On the other hand, when C is small, the model will prioritize having a larger margin, even if it leads to misclassifying more training points. Both models have been built using sklearn.svm.SVC from scikit-learn (Scikit-learn, 2023). Table 5 shows the accuracy of linear and nonlinear SVM models.

Table 5 – Accuracy of linear and nonlinear SVM models
Таблица 5 –Точность линейных и нелинейных SVM-моделей
Табела 5 – Прецизност линеарних и нелинеарних СВМ модела

SVM Model	Accuracy
Linear SVM	91 %
Nonlinear SVM	94 %

Conclusion

In recent years, understanding and analyzing datasets have been a major challenge in the field of machine learning. To address this issue, we used the MNIST dataset which is the largest collection of handwritten digits. We applied various statistical techniques, including the PCA, to analyze the dataset. While the PCA was effective for dimensionality reduction, it was not as effective for visualization. In addition, we used both linear and nonlinear SVM models to classify the dataset. These models achieved accuracies of 91% and 94%, respectively. For future endeavors, utilizing the PCA and the SVM together can enhance the efficiency and accuracy of the classifier. The PCA will perform dimensionality reduction by transforming features into a set of principal components, thus reducing the number of features in the dataset. Subsequently, training the SVM on this reduced dataset will result in quicker training times and improved performance.

References

Abdi, H. & Williams, L.J. 2010. Principal component analysis. *WIREs (Wiley Interdisciplinary Reviews*), 2(4), pp.433-459. Available at: https://doi.org/10.1002/wics.101.

Ahmed, A.H., Al-Hamadani, M.N.A. & Abdulrahman Satam, I. 2022. Prediction of COVID-19 disease severity using machine learning techniques. *Bulletin of Electrical Engineering and Informatics*, 11(2), pp.1069-1074. Available at: https://doi.org/10.11591/eei.v11i2.3272.

Al-Hamadani, M.N.A. 2015. *Evaluation of the Performance of Deep Learning Techniques Over Tampered Dataset*. Master thesis. Greensboro, North Carolina, USA: The University of North Carolina, Faculty of The Graduate School [online]. Available at: https://www.proquest.com/openview/769d2aa550c12fcf40655405e8df7689/1?pq-origsite=gscholar&cbl=18750 [Accessed: 05 February 2023].

Guenther, N. & Schonlau, M. 2016. Support Vector Machines. *The Stata Journal*, 16(4), pp.917-937. Available at: https://doi.org/10.1177/1536867X1601600407.

Hao, J. & Ho, T.K. 2019. Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, 44(3), pp.348-361. Available at: https://doi.org/10.3102/1076998619832248.

LeCun, Y. 2023. *MNIST dataset* [online]. Available: https://yann.lecun.com/exdb/mnist/.

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P. & Vapnik, V. 1995. Comparison of learning algorithms for handwritten digit recognition. In: *Fogelman, F. & Gallinari, P. (Eds.) International Conference on Artificial Neural Networks (ICANN'95),* Paris, pp. 53-60, October 9-13.

Manshor, N., Halin, A.A., Rajeswari, M. & Ramachandram, D. 2011. Feature selection via dimensionality reduction for object class recognition. In: *2011 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, Bandung, Indonesia, pp.223-227, November 08-09. Available at: https://doi.org/10.1109/ICICI-BME.2011.6108645.

Mishra, S.P., Sarkar, U., Taraphder, S., Datta, S., Swain, D.P., Saikhom, R., Panda, S. & Laishram, M. 2017. Multivariate Statistical Data Analysis-Principal Component Analysis (PCA). *International Journal of Livestock Research*, 7(5), pp.60-78.

Nielsen, M. 2019. *Neural Networks and Deep Learning* [online]. Available at: http://neuralnetworksanddeeplearning.com/ [Accessed: 05 February 2023].

Raschka, S., Patterson, J. & Nolet, C. 2020. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, 11(4), art.number:193. Availiable at: https://doi.org/10.3390/info11040193.

Saputra, D., Dharmawan, W.S. & Irmayani, W. 2022. Performance Comparison of the SVM and SVM-PSO Algorithms for Heart Disease Prediction. *International Journal of Advances in Data and Information Systems*, 3(2), pp.74-86. Available at: https://doi.org/10.25008/ijadis.v3i2.1243.

-Scikit-learn. 2023. *sklearn.svm.SVC* [online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html [Accessed: 05 February 2023].

Subasi, A. 2020. *Practical Machine Learning for Data Analysis Using Python.* London, United Kingdom: Elsevier, Academic Press. ISBN: 978-0-12-821379-7.

Suthaharan, S. 2014. Big data classification: problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review,* 41(4), pp.70-73. Available at: https://doi.org/10.1145/2627534.2627557.

Suthaharan, S. 2016. Support Vector Machine. In: *Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems*, 36. Boston, MA: Springer. Available at: https://doi.org/10.1007/978-1-4899-7641-3_9.

Wang, P., Li, Y. & Reddy, C.K. 2019. Machine Learning for Survival Analysis: A Survey. *ACM Computing Surveys*, 51(6), art.number:110, pp.1-36. Available at: https://doi.org/10.1145/3214306.

Классификация и анализ набора данных MNIST с использованием алгоритмов PCA и SVM

Мохалед Н.А. Аль-Хамадани

Дебреценский университет, Докторская школа информатики,
факультет науки о данных и визуализации, г. Дебрецен, Венгрия;
Северный технический университет, Технический институт/Аль-Хавиджа,
департамент электронных технологий, Адан, Киркук, Республика Ирак

РУБРИКА ГРНТИ: 27.41.23 Машинные, графические и другие методы вычислительной математики,
50.35.37 Программирование гибридных ЭВМ и ВК
ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: В современном мире использование методов машинного обучения стало незаменимым при анализе крупномасштабных и сложных данных. Данные методы широко применяются в различных областях: от оптимизации бизнес-операций до продвижения научных исследований. Несмотря на то что такие большие наборы данных открывают возможности для *глубокого понимания и инноваций, они также представляют серьезную проблему в таких областях, как качество и структура данных, которые требуют применения эффективных стратегий управления. Методы машинного обучения стали ключевыми инструментами в выявлении и смягчении вышеописанных проблем, а также в разработке соответствующих решений. Набор данных MNIST представляет собой яркий пример широко используемого набора данных в этой области, отличающегося огромной коллекцией рукописных цифр. Данный набор часто используется в таких задачах, как классификация и анализ, о чем свидетельствует настоящее исследование.*

Методы: В данном исследовании использовался набор данных MNIST для изучения различных статистических методов, включая алгоритм анализа основных компонентов (PCA), с помощью скриптового языка программирования Python. Также применялись модели опорных векторов (SVM) для оценки точности модели в задачах линейной и нелинейной классификации.

Результаты: Результаты настоящего исследования показывают, что хотя метод PCA эффективен для уменьшения размерности, он может оказаться не столь эффективным для целей визуализации. Более того, полученные результаты показали, что линейные, так же как и нелинейные SVM-модели эффективно классифицируют набор данных.

Выводы: Результаты исследования показали, что SVM явдяется эффективным методом для решения проблем классификации.

Ключевые слова: статистический анализ, машинное обучение, SVM, PCA, классификация.

Класифиkција и анализа скупа података МНИСТ помоћу алгоритама ПЦА и СВМ

Мокхалед Н.А. Ал-Хамадани

Универзитет у Дебрецину, Докторске студије информатике,
Одсек за науку и визуализацију података, Дебрецин, Мађарска;
Северни технички универзитет, Технички институт/Алхавија,
Одељење за електронске технике, Адан, Киркук, Република Ирак

ОБЛАСТ: математика, рачунарске науке, информационе технологије
КАТЕГОРИЈА (ТИП) ЧЛАНКА: оригинални научни рад

Сажетак:

Увод/циљ: Методи машинског учења постали су незаменљиви у анализи сложених података великог обима у савременим окружењима заснованим на подацима. Примењују се у најразличитијим областима, од оптимизације пословних процеса до сложених научних истраживања. Упркос томе што овакви обимни *скупови података нуде могућности дубинског сагледавања, као и иновација, они представљају и велики изазов у областима као што су квалитет и структура података, што захтева примену ефикасних стратегија управљања. Технике машинског учења су се показале као суштински важни алати за идентификацију и смањивање тих изазова, као и за развијање могућих решења. Скуп података МНИСТ представља изразит пример широко коришћених сетова података у овој области, познат по својој великој колекцији руком писаних цифара, и често је употребљаван за класификације и анализе, као што је то показано у овој студији.*

Методе: Скуп података МНИСТ коришћен је за испитивање различитих статистичких поступака, укључујући алгоритам анализе главних компоненти (Principal Components Analysis (PCA – ПЦА)) уз помоћ програмског језика Пајтон. Такође, примењени су модели метода потпорних вектора (Support Vector Machine (SVM – СВМ)) за процењивање тачности модела у линеарним и нелинеарним класификационим проблемима.

Резултати: Показано је да, иако техника ПЦА јесте ефикасна у редуковању димензионалности, она није толико ефикасна за визуализацију. Штавише, налази показују да су и линерани и нелинеарни модели СВМ успели да ефикасно класификују скуп података.

Закључак: Резултати студије показују да СВМ може да буде ефикасна техника за решавање проблема класификације.

Кључне речи: статистичка анализа, машинско учење, СВМ, ПЦА, класификација.

Paper received on / Дата получения работы / Датум пријема чланка: 08.02.2023.
Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 26.03.2023.
Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 28.03.2023.