ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/10.55185/researcher.1102901

Learning Capabilities of AI Methodologies on Multi-Class Datasets

Ender SEVINÇ[1*]
[1]Ankara Science University, Faculty of Engineering and Architecture, Computer Engineering Department, Ankara, Türkiye; ORCID: 0000-0001-7670-722X
* Corresponding Author: ender.sevinc@ankarabilim.edu.tr

Received: 13 April 2022; Accepted: 16 June 2022

Reference/Atıf: E. Sevinç, "Learning Capabilities of AI Methodologies on Multi-Class Datasets", Researcher, vol. 02, no. 01, pp. 19-28, Jul. 2022, doi:10.55185/researcher.1102901

Check for **updates**

Abstract

Machine Learning (ML) methods have numerous kinds of application areas up to now. Since they generally have remarkable success in learning, study areas and research field have diversified drastically. Neural networks seem to be appropriate for such a learning capability. The study discusses and examines several ML methodologies to decide the output. Since binary classification is another interesting area, the study focuses on multi-class classification problems. Datasets are chosen from a commonly known and accepted repository to avoid fakeness. Totally four different classifiers have been used to understand and know the different output classes in four different datasets. The classifiers use various arguments to work with and these will be shown and explained in detail. Two of the datasets are newly added and medium-sized, this is preferred to show that there is almost no time of execution difference among all. The system developed gives remarkable success rates and eliminates the differences among the classes using a neural networks system. It is believed that ML methods will have a wide range of application fields as researchers widen their point of view for academic studies.

Keywords: machine learning, multi-class classification, classifiers

1. Introduction

Machine learning means a neural network implementation that does mapping the input-output relationship from the known input-output pairs. The aim is to deduce the system response for unknown conditions (unknown input data). This constitutes one of the humankind's obsessions, i.e., to know the unknown. This emotion makes the human work on such algorithms or methods, machine learning is extremely promising in that sense. Supervised learning is the first step to be taken, easy to implement, and overcome the problem. The study focuses on such supervised methods.

In fact, the goal of machine learning is to choose an input-output mapping model. For example, when a model with too much capacity is overtrained, it means is ss overfitted, while a model with very less capacity is undertrained, then it is under fitted. These are the difficulties of the models used in ML algorithms.

However, ML has numerous and big advantages in making all manual processes automated. Nowadays, the world is in a big race for such automation processes. Such a power not only gives incredible capability in terms of knowledge but also provides incredible speeds and functionalities to manage heavy works previously.

Though there have been made several categorizations of ML fields, the study focuses on the most commonly used Machine Learning (ML) Algorithms. These algorithms can be listed as Linear regression, Logistic regression, Decision trees, Support Vector Machines (SVM) algorithm, Naive Bayes (NB) algorithm, KNN algorithm, K-means, Random Forest algorithm, Dimensionality reduction algorithms, Gradient boosting algorithm, and AdaBoosting algorithms, etc.

Here in the study, four among these algorithms are examined, and presented the results to show the success of ML algorithms using known datasets. These algorithms are; Decision tree, SVM, k-Nearest Neighbors algorithm (k-NN), and Naive Bayes. Decision Trees (DT) are one of the most common, advanced easy-to-use tools in decision-making. SVM, on the other hand, reaches promising results when multi-class categorization researches. K-Nearest Neighbor is one of the oldest known approaches so as the Naïve Bayes algorithm.

There are also prominent works using the Random Forest algorithm, Gradient boosting algorithm, and AdaBoosting algorithms in the literature. The AdaBoost algorithm is a boosting technique, and one example is presented in [1]. As the algorithm runs, all the weights are re-assigned to each instance, higher weights are for incorrectly classified instances while low weights are for correct ones. The methodology works on the principle of learners growing sequentially. As a result, each subsequent learner is grown from previously grown learners. Briefly, weak learners become strong ones as the error rate increases and this goes on up to the final decision is made.

Section 2 discusses the related studies issued up to now, then in Section 3, used algorithms and classifiers are explained in detail. Their capabilities and arguments are discussed Section 4 shows the success and accuracy rates obtained by the classifiers. Finally, conclusions and future works are discussed in the last section.

2. Related Work

It is very probable to find numerous studies in the literature, artificial intelligence and even focusing on deep learning nowadays, are extremely hot and attractive. You can easily find many researchers and many studies have been issued in the field.

ML methods have a wide range of improvement and application fields. Initially, genetic algorithms (GA) have been used for taking some sort of action in a reasonable time. This attitude has been lastly explained in [2]. A good implementation of GA is presented in [3] and a data exchange through the web in [4]. Then as the needs are changes smart methodologies come to the scene, and methodologies such as Extreme Learning Machines (ELM) are used. ELM is a learning methodology for the SLFNs and its speed is remarkable. This is shown in [5] and the purposed algorithm combines GA with ELM for feature selection to reach better success rates. This study focuses on a wrapper feature selection algorithm that predicts and forms a network to map the input nodes to their output counterparts. The proposed algorithm works uniquely to reach the best solutions.

A similar approach can be seen in studies [6] and [7]. These studies are mainly about graph coloring and solving the maximum vertex weight clique problem in huge graphs. The Graph Coloring Problem (GCP) can be defined as separating and grouping the vertices of a graph into various sets to minimize the colors used. In [6], the Teaching-Learning-Based Optimization (TLBO) metaheuristic is used which is a different version of GA integrated with tabu search algorithm. Study in [7] is mainly a local tabu search algorithm that is implemented parallelly by using MPI capabilities.

However, ensemble methods have an increasing effect on ML among learning methodologies. Such methods construct a set of classifiers and then classify new data points due to their predictions. The point is to combine the predictions of several base estimators of a given learning algorithm to reach a more general and robust model for prediction.

A recent and final study is presented in [1]. Here, an Adaptive Boost Algorithm using a Decision Tree estimator is proposed. The algorithm is run on a Covid-19 dataset and a tuning process is done to a classifier, and a binary classification problem is solved with higher success rates against state-of-the-art algorithms. There are similar studies feature selection methods using ML methods. One prominent one is presented in [8], i.e., a novel Hyper Learning Binary Dragonfly Algorithm which proposes a method for making feature selection in classification algorithms.

3. Proposed Work

Four commonly known machine learning algorithms have been used in the study. These are explained below respectively.

3.1 Decision tree

A decision tree classifier is a commonly known, interesting, and used approach in multiclass classification problems. It presents a set of questions and choices. The decision tree classification algorithm is very similar to that of a binary tree. There are a root and internal nodes each of which a decision is posed, and that node can be further split into separate records which have different ongoing characteristics. Finally, the leaves refer to the classes in which the dataset is split. The Decision Tree Algorithm is a building decision tree called ID3 in [9] which employs a top-down, greedy search through the space of possible branches with no backtracking.

A decision tree has a top-down structure that starts from a root node and involves partitioning the data into subsets. Intermediate nodes simply constitute the ongoing decision processes as the process proceeds. The homogeneity of a numerical sample is calculated to evaluate and use standard deviation. For example, in the case of completely homogeneous data, the deviation is zero.

In a decision tree, all the calculations are done due to the features and ongoing processes. Figure 1 presents a graphical representation for dealing with the possible solutions to a problem. The process starts with simply asking a question that is based on the answer (Yes/No), then it further split the tree into subtrees. Figure 1 visualizes the general structure of a decision tree.

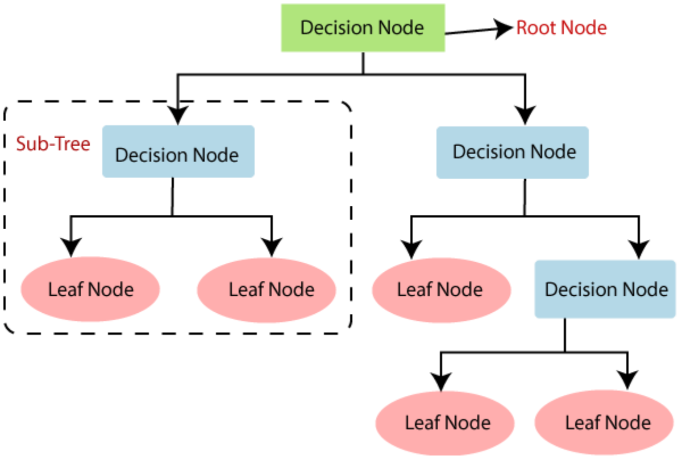


Figure 1: Sample Decision Tree

3.2 Support Vector Machines (SVM)

Support Vector Machines (SVMs) are one of the most used and promising methods in ML. It consists mainly of a set statistical method for supervised learning, and it can be applied to both classification and regression problems. The goal of SVM is to find a hyperplane among the input space of features, which has k-dimensions (k-the number of features) that distinctly classifies the data points.

One reason for the success of the SVM algorithm is that it chooses the best line to classify the data points. It decides the line that separates the data, and this line must be the furthest away from the closest data points as possible. If you minimize the bounds, the expected probability of error will be low, i.e., good generalization, and less error prone as stated in [10].

For the two-class classification problem as commonly known as separation is presented in Figure 2.

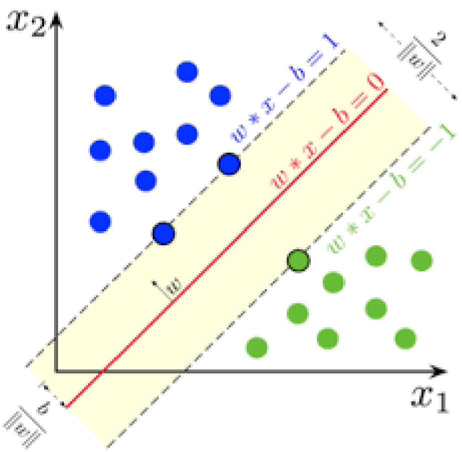


Figure 2: Support Vector Machine

3.3 k-Nearest Neighbor Algorithm

The k-Nearest Neighbors (KNN) algorithm is another commonly known and used supervised machine learning. The logic behind the KNN algorithm is simply that similar things exist in close proximity. In other words, similar things are near to each other. KNN depends on similarity which is evaluated by distance, proximity, or closeness. This calculation is done simply by evaluating the distance between points on a graph. On the other hand, the direct-line distance is a popular and known option which is also called as the Euclidean distance. This algorithm is used to solve both classification and regression problems in the literature

When it comes to classifying the data, KNN mostly and mainly uses the concept of "Majority Voting". This means that within the given range of K values, the class is chosen with the most votes. This can be barely seen in Figure 3.

However, the impact of selecting a smaller or larger K value on the model has the following

- Larger K value: The case of underfitting occurs when the value of k is increased.
- Smaller k value: The condition of overfitting occurs when the value of k is smaller.

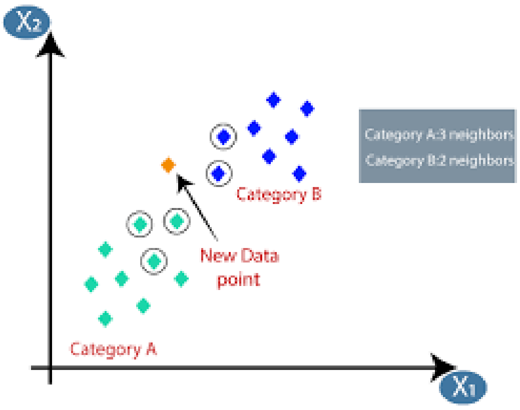


Figure 3: K-Nearest Neighbor Graph

3.4 Naive Bayes (NB)

Naïve Bayes uses Bayes Theorem as it can be understood from the name. It fundamentally uses a probabilistic approach, and this enables it to be used in a wide range of ML fields.

The Naïve Bayes algorithm is a good example among all ML algorithms for being simple. Naive Bayes' Theorem first starts with the assumption of independence predictors. This means that a particular feature in a class is irrelevant and unrelated to any other feature. Besides that, it is known to be easy to build and particularly useful for huge data sets. Along with simplicity, Naive Bayes is known to perform better than most state-of-the-art and sophisticated classification methods. as stated in [11]

The theorem works with posterior probabilities like P(c|x) from P(c), P(x), and P(x|c). Likewise, conditional probability is evaluated according to which is the probability of some sort of events. A general equation for NB is shown in Formula 1 below;

$$\mathrm{P(A|B)} = \frac{P(B|A).P(A)}{P(B)} \tag{1}$$

However, the study is a multi-class classification problem that implements the Gaussian Naive Bayes algorithm for classification. The similarity of the features is naive and calculations will be done accordingly.

4. Experiments and Implementations

The experiments are carried out on a normal laptop Windows-10 machine. It has an i7 CPU (i7-5700HQ CPU @ 2.70GHz) and 16 GB of memory. The code is implemented in Jupyter-notebook version 3.2.1. All coding is Python 3.9 compliant and in *.ipynb format.

The dataset for experiments is selected from the UCI Machine Learning Repository in [12]. Four known multi-class datasets are used. The definitions of the datasets are shown in Table 1.

Table 1: Features of Used Datasets

Dataset Name	ID	# of Instances	# of Attributes	# of Output Classes
Iris	IRI	150	4	3
Wine	WIN	178	13	3
Maternal Health Risk Data Set	MAT	1014	7	3
Nonverbal tourists	TOU	73	22	6

The important point about the datasets is that 2 of them are old while the final two are recently added to the repository. All of them are classification problems consisting of real, integer, or string values. "Null" values are accepted as 0, and all strings have been converted to their numeric value by a label encoder library.

Each dataset has been divided into training and test datasets with the code stated in Algorithm 1. For any kind of coding process, python has been used, and the Scikit-learn framework in python is an open-source machine learning library that supports supervised and unsupervised learning. It provides even basic and even sophisticated tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities. In other words, scikit-learn provides most of the main features which are needed for a basic working knowledge of machine learning practices (model fitting, predicting, cross-validation, etc.).

Algorithm 1: Train-Test Split

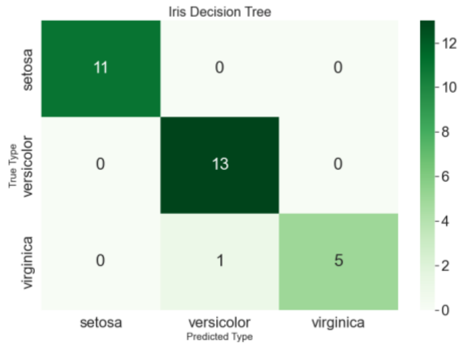
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
                     X, y, test_size=0.2, random_state=42)
```

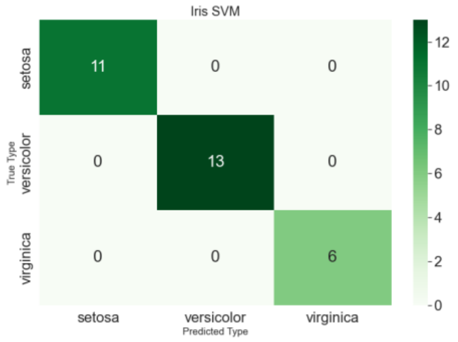
Then with the help of a related scikit-learn library, all datasets have been split into random train and test subsets. Train dataset contains the random 80% and the test data set contains the rest random 20% of the whole. “Random state” is a seed control value while the shuffling is applied to the data before applying the split.

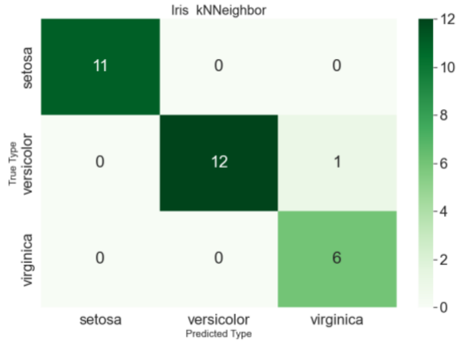
The related classifiers (Decision Tree, SVM, kNN, and Gaussian NB) use the following parameters because they all have a positive or negative effect on the result. This tuning process is left as a future work for the study.

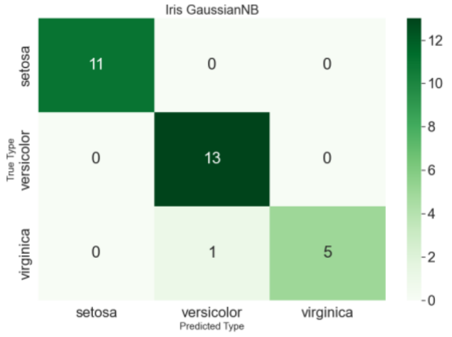
- DecisionTreeClassifier(max_depth = 4)
- SVC(kernel = 'linear', C = 4)
- KNeighborsClassifier(n_neighbors = 5)
- GaussianNB()

Table 2: Confusion Matrix of Classifiers for IRIS Dataset



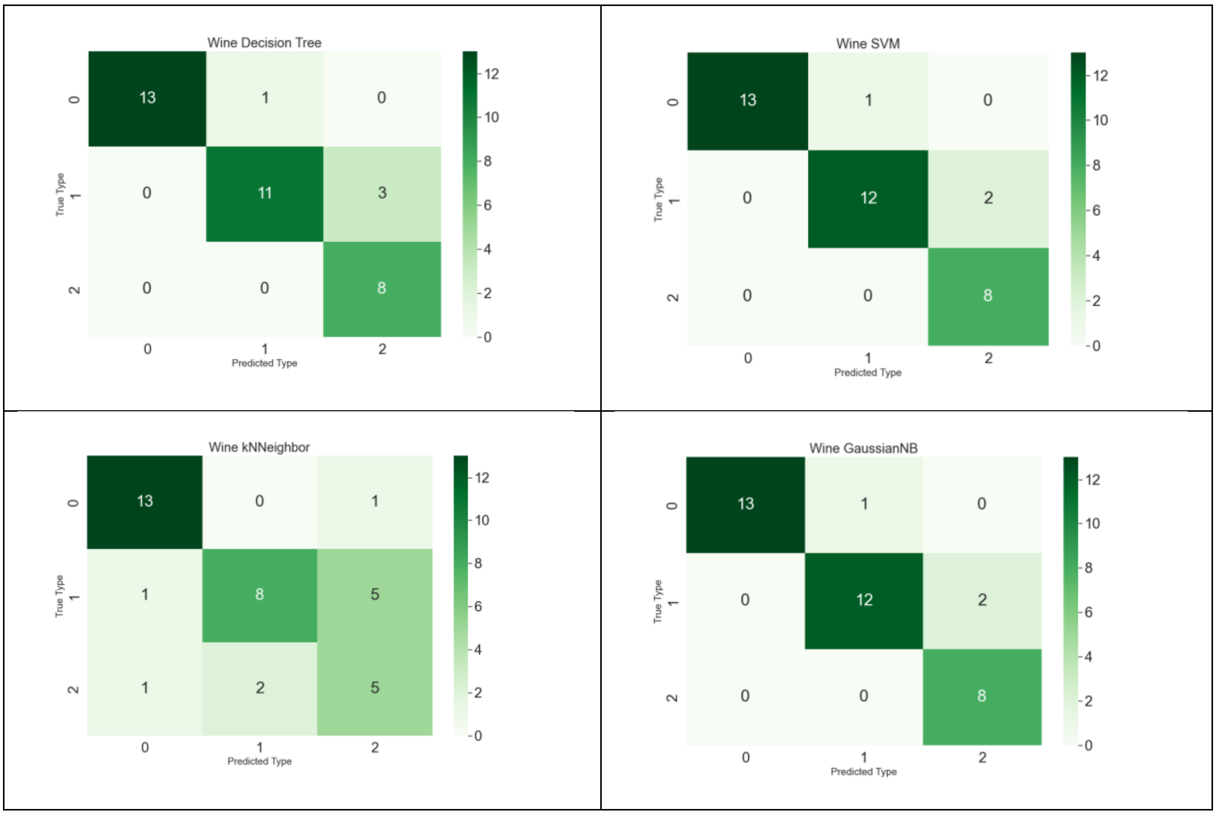






If to examine the results of each dataset, in Table 2, Iris dataset results are presented using the 4 classifiers stated above

Table 3: Confusion Matrix of Classifiers for WINE Dataset



Especially the results of the final dataset TOU, with 6 output classes worth examining. SVM seems to be the best classifier among others since it has the highest accuracy values in 3 out of 4 datasets. However, the Gaussian NB algorithm has also promised results. If the results are examined due to the size, then it can be inferred that Gaussian NB can be a good alternative.

All results are presented as a Confusion Matrices in Tables 2-5. A confusion matrix can be defined as a summary of prediction results against real values. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the main goal and a brief representation of a confusion matrix.

Besides that, the confusion matrix shows how your classification model is confused when it makes predictions. It provides the insight not only into the errors being made by the selected classifier but more importantly the types of errors that are being made.

Table 4: Confusion Matrix of Classifiers for MAT Dataset

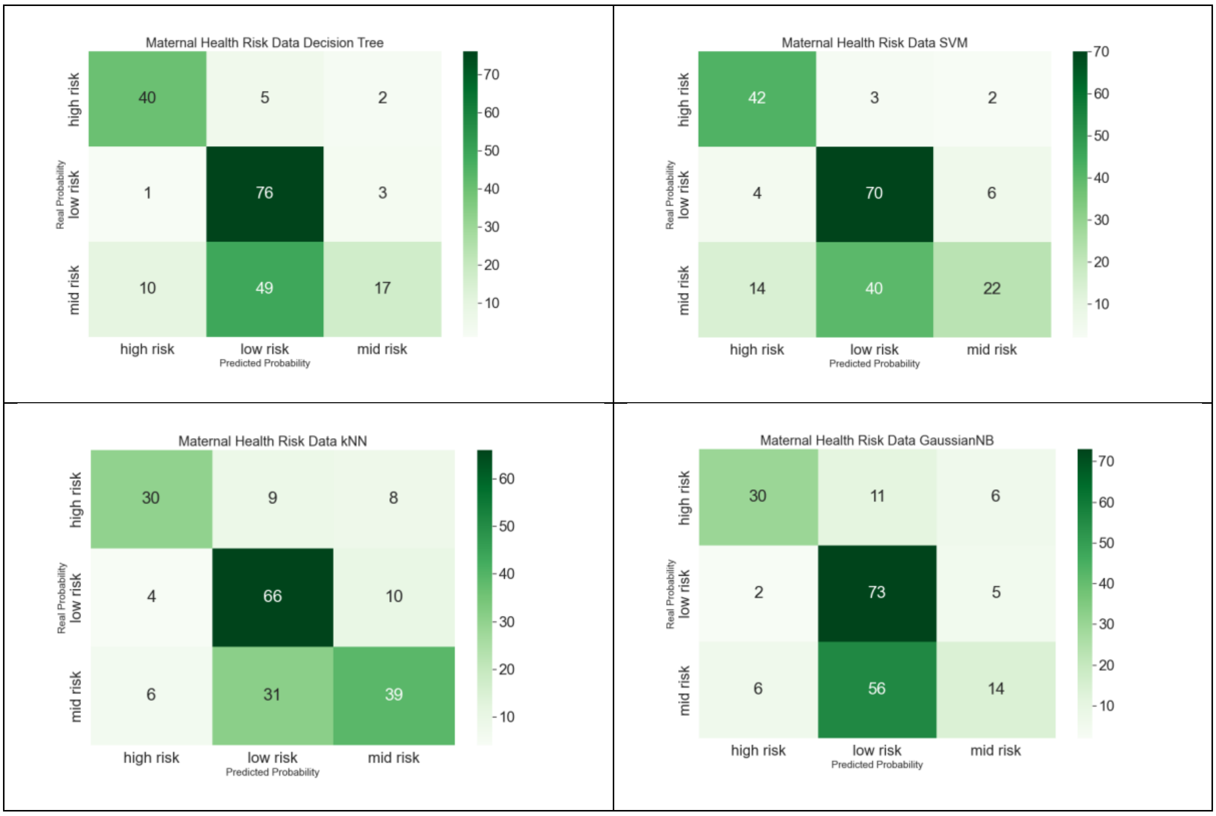
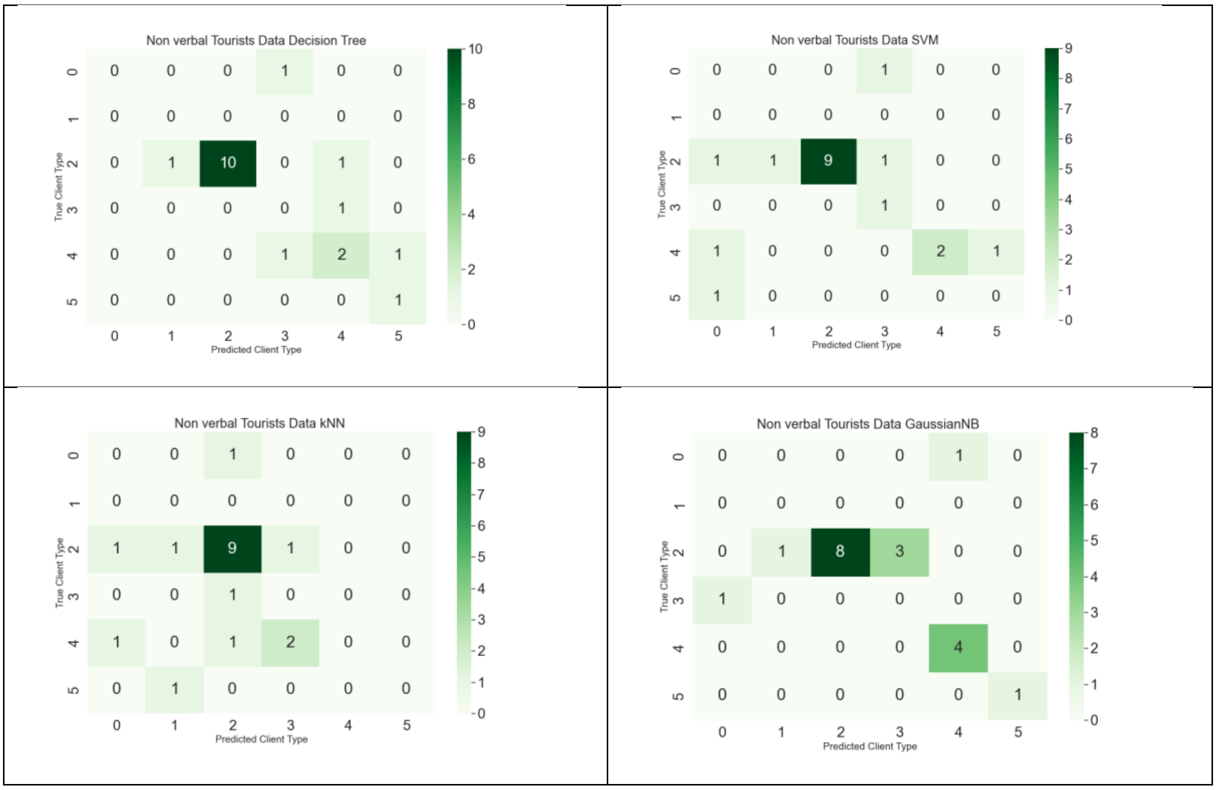


Table 5: Confusion Matrix of Classifiers for TOU Dataset



For example, if we check Table 2 for the Iris dataset solved by Decision Tree, the y-axis shows the true values while the x-axis shows the predicted values. The first row says ("11,0,0") that 11 *setosa* type iris flower is predicted, which is true, then the second row ("0,13,0") says that 11 *versicolor* type iris flower is predicted, which is true, then finally third row ("0,1,5") says that 5 out of 6 virginica iris flower is predicted, which is true, however, 1 out 6 is predicted as *versicolor*, but it must be *virginica*. The result is we have only one false prediction for this dataset.

When we check the whole accuracy figures, all the results show that SVM is more probable for reaching better results which can be seen in Table 6. The highest values among the classifiers are bolded.

SVM knows the best 3 out of 4, Gaussian NB the best 2 out of 4, and the rest algorithms know the best for only one dataset. This result briefly shows the power of SVM algorithm, esp. for such multiclass datasets.

Table 6: Results for Classifiers

ID	Decision Tree	SVM	kNN Alg.	GaussianNB
IRI	0,9666	**1,0000**	0,9666	0,9666
WIN	0,8888	**0,9167**	0,7222	**0,9167**
MAT	0,6552	**0,6601**	**0,6601**	0,5763
TOU	**0,6842**	0,6316	0,4736	**0,6842**

4. Conclusions and Future Work

In the study, 4 known and accepted algorithms have been executed and solved by using the known and accepted repositories. The results showed that ML algorithms are both easy to use and good at reaching better accuracy in a reasonable time amount. This success can be developed by tuning parameters. Such a capability makes them the best used and preferred methods indisputably among the learning methodologies.

As stated in the previous section, all the experiments must be diversified with different and various types of classifiers. Here, only 4 of them have been used and they will be beneficial for a real-life application. Because as more datasets and characteristics are used, the success rate may degrade or underperform. Different data characteristics, wrong models/classifiers, or lack of time may cause such results.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] Sevinç, Ender. "An empowered AdaBoost algorithm implementation: A COVID-19 dataset study." Computers & Industrial Engineering (2022): 107912.

[2] Mirjalili, Seyedali. "Genetic algorithm." Evolutionary algorithms and neural networks. Springer, Cham, 2019. 43-55.

[3] Karakaya, Murat, and Ender SEVİNÇ. "An efficient genetic algorithm for routing multiple uavs under flight range and service time window constraints." Bilişim Teknolojileri Dergisi 10.1 (2017): 113.

[4] Cingil, Ibrahim, et al. "Dynamic modification of XML documents: External application invocation from XML." ACM SIGecom exchanges 1.1 (2000): 1-6.

[5] Sevinc, Ender. "A novel evolutionary algorithm for data classification problem with extreme learning machines." IEEE Access 7 (2019): 122419-122427.

[6] Dokeroglu, Tansel, and Ender Sevinc. "Memetic Teaching–Learning-Based Optimization algorithms for large graph coloring problems." Engineering Applications of Artificial Intelligence 102 (2021): 104282.

[7] Sevinc, Ender, and Tansel Dokeroglu. "A novel parallel local search algorithm for the maximum vertex weight clique problem in large graphs." Soft Computing 24.5 (2020): 3551-3567.

[8] Too, Jingwei, and Seyedali Mirjalili. "A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study." Knowledge-Based Systems 212 (2021): 106553.

[9] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[10] Pisner, Derek A., and David M. Schnyer. "Support vector machine." Machine learning. Academic Press, 2020. 101-121.

[11] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.

[12] UC Irvine Machine Learning Repository [Online]. Available: https://archive.ics.uci.edu/ml/datasets.php [Accessed: 10-Apr-2022].