Proceedings of the Eighth International Conference on Research in Intelligent Computing in Engineering pp. 7–10
DOI: 10.15439/2023R63
ISSN 2300-5963 ACSIS, Vol. 38

Classification of Plant Species with Iris Dataset Using ANN, KNN and K-Means Algorithms

M. Hanefi Calp
Department of Management Information Systems
Faculty of Economics & Administrative
Sciences, Ankara Hacı Bayram Veli University
Ankara, Turkey
hanefi.calp@hbv.edu.tr
0000-0001-7991-438X

Vijender Kumar Solanki
CMR Institute of Technology
Hyderabad, TS, India
spesinfo@yahoo.com
0000-0001-5784-1052

***Abstract*—In this study, plant species were classified on the Iris dataset using Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), and K-Means algorithms. In this process, models were developed for each method, success rates were obtained, and a model with a minimum error rate was introduced. The dataset of the study was obtained from the Kaggle website. The classification process was applied repeatedly on the iris dataset, and the classification or prediction with the minimum error rate was aimed at the established models. In the study process, first of all, the dataset was obtained, prepared, and visualized. Models were created using the Jupiter Notebook editor via the Anaconda desktop GUI. Then, the models were analyzed and the most successful algorithm was selected. As a result, according to the prediction/classification models, it was seen that the most successful model was obtained with the KNN algorithm, and the most unsuccessful model was obtained with the ANN algorithm.**

***Index Terms*—iris, plant species, ann, knn, k-means, classification.**

I. Introduction

Artificial intelligence is defined as the ability of a computer or computer-assisted machine to perform tasks such as human characteristics, thinking like a human, solving problems, finding solutions, understanding, making sense, generalizing, and learning from past experiences [1].

In general, Artificial Intelligence, inspired by humans and nature and successfully solving real-world problems with mathematical and logical approaches, has increased its preference level as it can be used effectively in many areas. Problems that cannot be solved with traditional methods and techniques or that cannot be achieved at the desired level can be successfully solved by using models created with Artificial Intelligence. At this point, very successful results can be obtained for real-world problems such as prediction, diagnosis, classification, pattern recognition, optimization, and interpretation with AI. Artificial Intelligence is used in many fields such as engineering-based fields, education, economy, military, and health [2-7]. In summary, AI is the general name of technology created with completely artificial tools, which can exhibit human-like behaviors and movements. In other words, AI can fully perform human actions such as feeling, predicting behavior, and making decisions [8].

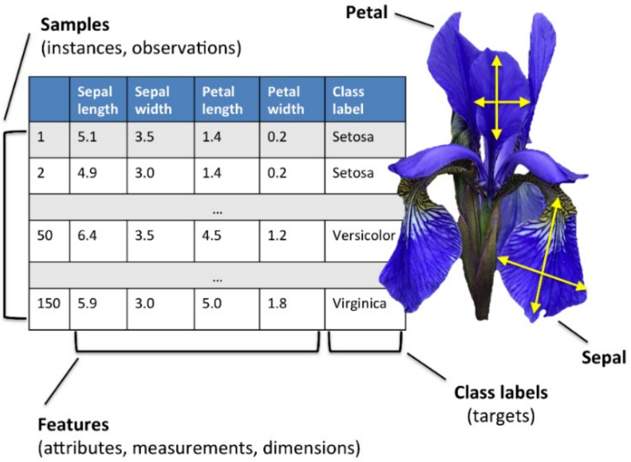
In this context, this study aims to classify plant species on the iris dataset by using ANN, KNN, and K-Means algorithms. In the second part of the study, all the details of the material and method are explained. Then, in the third part, the results and recommendations obtained from the study are given.

II. Material and Method

In the study, ANN, KNN, and K-Means algorithms were used and models were created. Python programming language was used to create the models. Matplotlib and Numpy, Sckit Learn libraries were used for visualization and calculations. All the details of the models are given in this section.

A. Dataset

The dataset used in the creation of the models belongs to "Iris". The dataset was obtained from the Kaggle website. The Iris dataset contains information on petal length and width, sepal length and width of Iris versicolor, Iris virginica, and Iris setosa plant species. In addition, the dataset includes a total of 150 samples, 50 from each plant species. All values (minimum, maximum, and average values) of the petal and sepal of each species are given in Figure 1.


Samples
(instances, observations)
Petal
Sepal
Class labels
(targets)
Features
(attributes, measurements, dimensions)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Fig. 1. Dataset [9]

In Table 1, the example 5 rows of input data (input, output) were given. Output variables indicated the plant species. Species were coded 0 (0: Setosa), 1 (1: Versicolor), and 2 (2: Virginica).

TABLE 1. The sample data for input and output

	Sepal	Sepal	Petal	Petal		
No	**lenght**	**width**	**lenght**	**width**	**Iris Type**	
	(cm)	**(cm)**	**(cm)**	**(cm)**		
0	5.1	3.5	1.4	0.2	0	2
1	4.9	3.0	1.4	0.2	1	1
2	4.7	3.2	1.3	0.2	2	0
3	4.6	3.1	1.5	0.2	3	1
4	5.0	3.6	1.4	0.2	4	2

B. *ANN*

ANN is a technology that analyzes data by copying the working logic of the human brain and generates new information from this data with different learning algorithms. ANN provides the opportunity to make some decisions or actions using data. Technically, the task of ANN is to produce an output using the dataset entered into the model. In order to achieve this, the network is trained with sample datasets. Then, the network becomes able to make a comment or make a decision, and thus the outputs are determined [10-13]. ANN is widely used in many fields such as prediction, classification, system diagnostics, pattern recognition, robotics, and signal processing [14].

As seen in Figure 2, an artificial neural network was established with plant species, namely Setosa, Versicolor, and Virginia, in the input layer, sepal width, sepal length, petal width, and petal length in the output layer in the iris dataset in which artificial neural networks were applied.

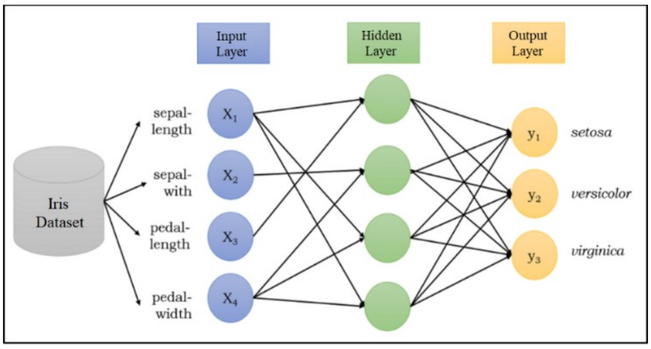


Fig. 2. Classification of the dataset

1) *Data Preprocessing Phase*

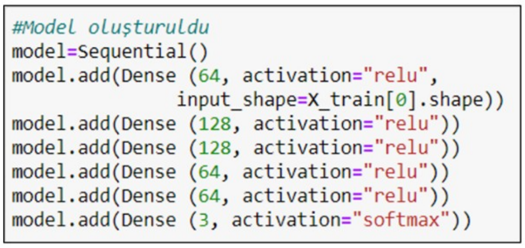
The Data Preprocessing stage is very important before the model setup. At this stage, missing data is corrected and data is normalized. The info() method in the Pandas library is used to find the missing data. In case of missing data, the average value was entered in that row or completely removed. The dataset was divided into two a 10% (15) test and a 90% (135) training set. The "Train Test Split" function is used to split the dataset (input and output values). This function is imported from the "sklearn library". The model was established with the training data, and the model was tested with the test data that the model had never seen.

2) *Normalizing of the Data*

The variance of the data was tested and the data with high variance was normalized. In the output obtained, X_train and Y_test have low variance. Therefore, there is no need for normalization.

3) *Creating of the Model*

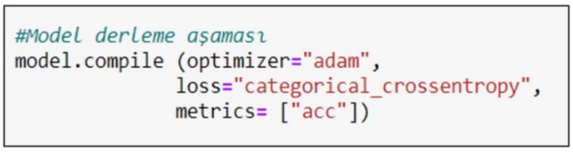
At this stage, layers were added to the model with the model.add() function, and how many neurons were printed into these layers was included in the model.add() function (Figure 3).



```
#Model oluşturuldu
model=Sequential()
model.add(Dense (64, activation="relu",
               input_shape=X_train[0].shape))
model.add(Dense (128, activation="relu"))
model.add(Dense (128, activation="relu"))
model.add(Dense (64, activation="relu"))
model.add(Dense (64, activation="relu"))
model.add(Dense (3, activation="softmax"))
```

Fig. 3. Pseudocode for the model creating

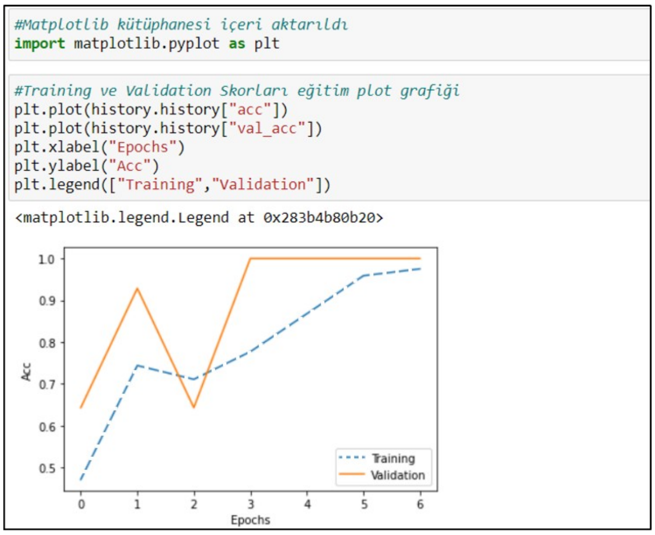
The number of epochs indicates how many times the model will be trained. As the number of epochs increases, the training time of the model also increases. 7 epoch values were entered during the model compilation phase. In the output of the code, acc value: 0.9752, val_acc: 1.000.



```
#Model derleme aşaması
model.compile (optimizer="adam",
               loss="categorical_crossentropy",
               metrics= ["acc"])
```

Fig. 4. Compilation phase of the model

In Figure 5, there is a graph showing the data on the training history. In the graph, the blue lines show the Training scores and the orange lines show the Validation score. With this table, it was concluded that the model learns more with the increase in the epoch value.



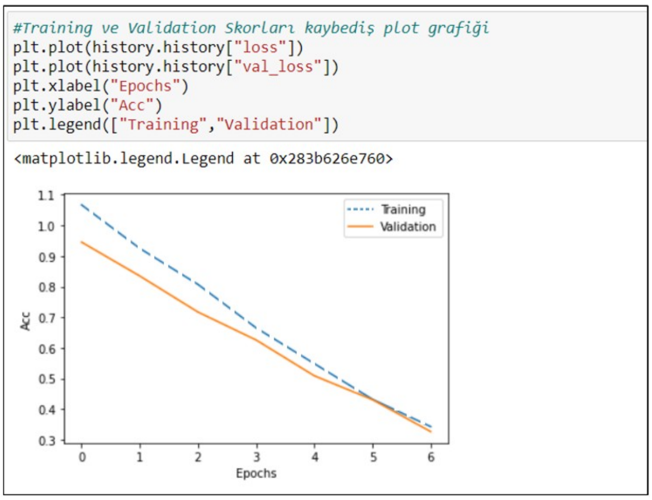
```
#Matplotlib kütüphanesi içeri aktarıldı
import matplotlib.pyplot as plt
```

```
#Training ve Validation Skorları eğitim plot grafiği
plt.plot(history.history["acc"])
plt.plot(history.history["val_acc"])
plt.xlabel("Epochs")
plt.ylabel("Acc")
plt.legend(["Training","Validation"])
```

```
<matplotlib.legend.Legend at 0x283b4b80b20>
```

Fig. 5. Training Graph of the Training and Validation Scores

In Figure 6, the graph of the loss value was drawn. After a certain value, it was seen that both values, namely Training and Validation scores, decreased. In other words, as the Epochs value increased, the model made better predictions and the error decreased.



```
#Training ve Validation Skorları kaybediş plot grafiği
plt.plot(history.history["loss"])
plt.plot(history.history["val_loss"])
plt.xlabel("Epochs")
plt.ylabel("Acc")
plt.legend(["Training","Validation"])
```

```
<matplotlib.legend.Legend at 0x283b626e760>
```

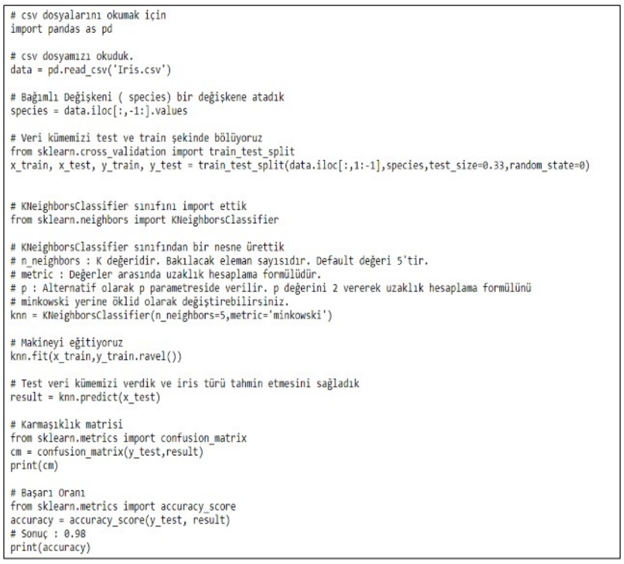
Fig. 6. Loss Graph of the Training and Validation Scores

Accuracy estimation shows the percentage value of how the model will predict test data it has never seen before. As a result, the created ANN model showed 86% success.

C. KNN

KNN is one of the machine learning algorithms that is easy to implement because it produces successful results against noisy training data. KNN is used in classification and regression problems in supervised learning and there is no training phase. KNN is considered the simplest machine learning algorithm. In the KNN algorithm, the K number determined to determine which class the object will be included in shows how many neighbors are closest to it. The distance to these neighbors is calculated with the Euclidean function. The unknown object is included in that class if it is closest to which class in the number K [15-17].

The closeness of the accuracy rate to 1 is extremely important for the accuracy of the dataset. In Figure 7, the codes developed for the creation of the KNN model and their explanations are indicated.



```
# csv dosyalarını okumak için
import pandas as pd

# csv dosyamızı okuduk.
data = pd.read_csv('Iris.csv')

# Bağımlı Değişkeni ( species) bir değişkene atadık
species = data.iloc[:,-1:].values

# Veri kümemizi test ve train şekinde bölüyoruz
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(data.iloc[:,1:-1],species,test_size=0.33,random_state=0)


# KNeighborsClassifier sınıfını import ettik
from sklearn.neighbors import KNeighborsClassifier

# KNeighborsClassifier sınıfından bir nesne ürettik
# n_neighbors : K değeridir. Bakılacak eleman sayısıdır. Default değeri 5'tir.
# metric : Değerler arasında uzaklık hesaplama formülüdür.
# p : Alternatif olarak p parametreside verilir. p değerini 2 vererek uzaklık hesaplama formülünü
# minkowski yerine öklid olarak değiştirebilirsiniz.
knn = KNeighborsClassifier(n_neighbors=5,metric='minkowski')

# Makineyi eğitiyoruz
knn.fit(x_train,y_train.ravel())

# Test veri kümemizi verdik ve iris türü tahmin etmesini sağladık
result = knn.predict(x_test)

# Karmaşıklık matrisi
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,result)
print(cm)

# Başarı Oranı
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, result)
# Sonuç : 0.98
print(accuracy)
```

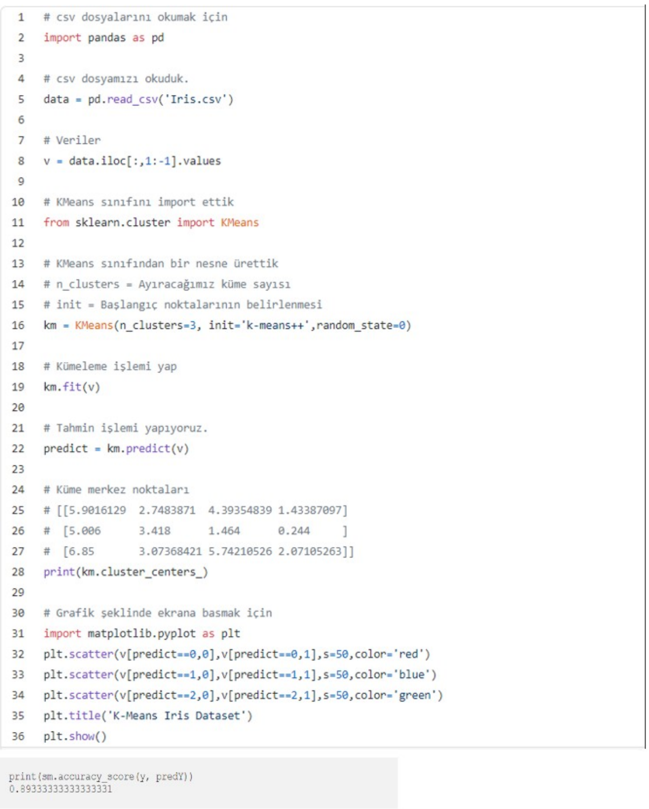
Fig. 7. KNN codes and explanations

At this stage, the Complexity matrix was obtained. According to this result, when the dataset was considered, as a result of the complexity matrix of the application, it was concluded that 48 of the 50 datasets were estimated correctly, while 1 of them was incorrectly estimated. The success rate is 48/50 = 0.96. The Accuracy Score is very close to 1 with 0.96. The closer the obtained value was to 1, the higher the accuracy or performance of the model.

D. K-Means Clustering

K-means clustering is one of the widely used and easy-to-implement clustering algorithms and uses the exploratory data analysis technique [18]. The aim is to ensure that the clusters obtained at the end of the partitioning process have maximum similarities within clusters and minimum similarities between clusters. It can cluster large-scale data quickly and effectively. “K” refers to the fixed number of clusters needed before starting the algorithm. With its iterative partitioner structure, the K-means algorithm reduces the sum of the distances of each data to the cluster it belongs to. The K-means algorithm tries to detect K clusters that will make the square error minimum [19].

The closeness of the accuracy rate to 1 is extremely important for the accuracy of the dataset. In Figure 8, the codes of the algorithm developed for the creation of the K-Means model are given.



```
# csv dosyalarını okumak için
import pandas as pd

# csv dosyamızı okuduk.
data = pd.read_csv('Iris.csv')

# Veriler
v = data.iloc[:,1:-1].values

# KMeans sınıfını import ettik
from sklearn.cluster import KMeans

# KMeans sınıfından bir nesne ürettik
# n_clusters = Ayıracağımız küme sayısı
# init = Başlangıç noktalarının belirlenmesi
km = KMeans(n_clusters=3, init='k-means++',random_state=0)

# Kümeleme işlemi yap
km.fit(v)

# Tahmin işlemi yapıyoruz.
predict = km.predict(v)

# Küme merkez noktaları
# [[5.9016129  2.7483871  4.39354839 1.43387097]
#  [5.006      3.418      1.464      0.244     ]
#  [6.85       3.07368421 5.74210526 2.07105263]]
print(km.cluster_centers_)

# Grafik şeklinde ekrana basmak için
import matplotlib.pyplot as plt
plt.scatter(v[predict==0,0],v[predict==0,1],s=50,color='red')
plt.scatter(v[predict==1,0],v[predict==1,1],s=50,color='blue')
plt.scatter(v[predict==2,0],v[predict==2,1],s=50,color='green')
plt.title('K-Means Iris Dataset')
plt.show()
```

```
print(sm.accuracy_score(y, predY))
0.8933333333333331
```

Fig. 8. Codes of the K-Means Algorithm

In Figure 9, the clustering result of the K-Means algorithm was given. In our model, which was made using K-Means and Hierarchical techniques, the k parameter, that was, the number of clusters was determined as 3. Based on the k parameter, that was, the number of clusters, out of 150 data in the dataset, it was continued as 3 clusters as Setosa, Virginica, and Versicolor. Setosa, Virginica, and Versicolor clusters each contain 33.33% of the Iris dataset. Looking at Figure 9, it was seen that the success rate/accuracy rate of the data in the dataset is 89%. The closer the obtained value was to 1, the higher the performance of the model.

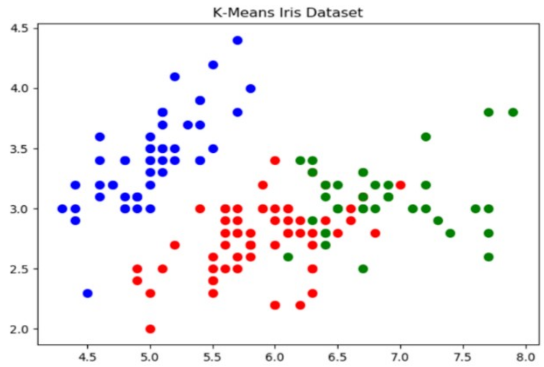


Fig. 9. Clustering result of the K-Means algorithm

In general, when the results obtained from all models were compared, the accuracy rates of 86% in the ANN method, 96% in the KNN method, and 89% in the K-Means method were reached. This situation revealed that the highest performance was obtained with the KNN method.

Finally, different tests were carried out using real data on the proposed models, and the experimental results obtained are given in Table 2.

TABLE 2. The experimental results

No	KNN	Loss (KNN)	K-Means	Loss (K-Means)	ANN	Loss (ANN)
1	97,4	2,6	89,14	10,86	87,4	12,6
2	97,3	2,7	88,25	11,75	85,6	14,4
3	96,25	3,75	89,54	10,46	87,3	12,7
4	95,35	4,65	86,2	13,8	86,5	13,5
5	97,27	2,73	92,5	7,5	90,8	9,2
6	96,14	3,86	90,7	9,3	83,2	16,8
7	96,7	3,3	91,5	8,5	88,4	11,6
8	94,8	5,2	90,8	9,2	87,2	12,8
9	97,6	2,4	89,6	10,4	81,5	18,5
10	97,35	2,65	88,2	11,8	87,6	12,4
Average	96,616	3,384	89,643	10,357	86,55	13,45

When Table 2 is examined, it confirms the success rates obtained from the models. In other words, it can be seen that the most successful results are obtained with the KNN model.

III. Conclusion and Recommendations

There is a significant increase in the amount of data day by day with the development of technology and the internet. This situation has brought about the discovery, analysis, use, and application of the available data in every field. This makes it possible to easily identify some relationships that cannot be detected by classical methods using data, and to make predictions and predictions for the future by performing big data analysis. Big data is used effectively in many current algorithms from artificial intelligence methods to image processing techniques and provides many advantages.

Therefore, in this study, big data from the iris plant was used and the flower type was estimated according to the sepal length, sepal width, petal length, and petal width values of the iris flower using artificial intelligence methods. For this purpose, ANN, KNN (K-Nearest Neighbors Algorithm), and K-Means Clustering methods were used. As a result of the methods used, the lowest success rate was obtained in the ANN method, and the highest success rate was obtained in the KNN method. As a result of the codes written, the method with the highest accuracy rate was KNN. The study is expected to be a sample application for other estimation methods. In future studies, it is planned to make comparisons using different algorithms.

References

[1] V. V., Nabiyev, Yapay Zekâ: İnsan-Bilgisayar Etkileşimi. Baski Yeri: Seçkin Yayıncılık, 2012.

[2] S.J., Russell, P., Norvig, J.F., Canny, J.M. Malik, and D.D., Edwards, “Artificial Intelligence: A Modern Approach”, 2(9): Upper Saddle River: Prentice Hall, 2003.

[3] V.V., Nabiyev, “Yapay Zeka: Problemler-Yöntemler-Algoritmalar”, Seçkin Yayıncılık, 2012.

[4] N., Allahverdi, “Uzman Sistemler: Bir Yapay Zeka Uygulaması”, Atlas Yayın Dağıtım, 2002.

[5] A., Strong, “Applications of artificial intelligence and associated technologies”, Science (ETEBMS-2016), 5(6), 2016.

[6] R., Butuner, I., Cinar, Y. S., Taspinar, R., Kursun, M. H., Calp, & M. Koklu, (2023). Classification of deep image features of lentil varieties with machine learning techniques. European Food Research and Technology, 249(5), 1303-1316.

[7] T., Savaş & S. Savaş, (2021). Tekdüzen kaynak bulucu yoluyla kimlik avı tespiti için makine öğrenmesi algoritmalarının özellik tabanlı performans karşılaştırması. Politeknik Dergisi, 1-1.

[8] S., Ipek, Yapay zekanın toplum üzerindeki etkisi ve yapay zekâ (AI) filmi bağlamında yapay zekaya bakış. Uluslararası Ders Kitapları ve Eğitim Materyalleri Dergisi, 2(2), 203-215, 2019.

[9] S. Raschka: https://sebastianraschka.com/images/blog/2015/principal_component_analysis_files/iris.png, Access Date: 02.02.23.

[10] B. Marr, What is an Artificial Neural Networks?, https://www.bernardmarr.com/default.asp?contentID=2126, Access Date: 10.02.23.

[11] F. Fahrettin, https://medium.com/@fahrettinf/4-1-1-artificial-neural-networks-6257a7a54bb3, Access Date: 21.02.23.

[12] M. F., Keskenler, & E. F., Keskenler, Geçmişten günümüze yapay sinir ağları ve tarihçesi. Takvim-i Vekayi, 5(2), 8-18, 2017.

[13] M. H., Calp, & U., Kose, Estimation of burned areas in forest fires using artificial neural networks. Ingeniería Solidaria, 16(3), 1-22, 2020.

[14] M. Ö., Efe, & O. Kaynak, (2004). Yapay sinir ağları ve uygulamaları. Boğaziçi Üniversitesi.

[15] E. Hatipoğlu, https://medium.com/@ekrem.hatipoglu/machine-learning-classification-k-nn-k-en-yak%C4%B1n-kom%C5%9Fu-part-9-6f18cd6185d, Access Date: 23.02.23.

[16] J., Han, J. Pei & M., Kamber (2011). Data Mining: Concepts and Techniques. Elsevier.

[17] D., Kilinc, E., Borandag, F., Yucalar, V., Tunali, M., Simsek, & A. Ozcift, (2016), Classification of Scientific Articles Using Text Mining with KNN Algorithm and R Language. Marmara Journal of Pure and Applied Sciences, 3, 89-94. https://doi.org/10.7240/mufbed.69674.

[18] https://medium.com/deep-learning-turkiye/k-means-algoritmas%C4%B1-b460620dd02a, Access Date: 15.02.23.

[19] Vikipedi, https://tr.wikipedia.org/wiki/K-means_k%C3%BCmeleme, Access Date: 18.02.23.