International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

A Model Identifying Iris Species using Machine Learning

Pala Prathima[1], Ranjith Kumar T[2]

[1]*Assistant Professor, Dept. of Computer Science, Chaitanya Deemed to be University, Warangal Urban, Telangana, India*

[2]*Assistant Professor, Dept. of Computer Science, Chaitanya Deemed to be University, Warangal Urban, Telangana, India*

Abstract - The aim of this paper is to get awareness about creation of a model in Machine Learning. We take an iris dataset which is existed in Scikit tools. This dataset is identified as a classification problem in supervised learning. Then this data is processed using scikit tool by splitting dataset into training data and test data. Using K-Nearest Neighbors algorithm a model is build and tested with new samples.

Key Words: Machine Learning, K-Nearest Neighbors, Scikit

1.INTRODUCTION

In the recent technologies, Machine Learning is an application of artificial Intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly Programmed. Mainly it focuses on the development of computer programs that can access data and use it learn for themselves. There are two main categories [1] in Machine learning one is Supervised Learning and Unsupervised Learning. Supervised learning as the name suggests getting supervised by someone. It is a learning in which the machine uses data which is already tagged with the correct answer. After that, the machine is provided with a new set of data. While in supervised learning the problems are grouped into two ways and solved with different algorithms. One way is regression and other is classification. With regression [3] various algorithms are used like Linear Regression, logistical regression, and polynomial regression are popular. With classification various algorithms are used like Decision Tree, Bayes classifier, K-Nearest Neighbors, Support Vector Machine and many more.

In this paper, a machine learning model is built from an iris dataset which contains the measurements of some irises that have been previously identified by an expert botanist [2] as belonging to the species *setosa*, *versicolor*, or *virginica.* From these, measurement we can predict iris species belongs to. There were three possible species, which made the task a three class classification problem which comes under supervised learning task. In classification, the possible species are called *classes* and single iris is called its label. To predict an iris species, belong to different tools, packages and libraries are used like Scikit tools, Numpy, pandas, Matplotlib, mglearn packages, Juypter Notebook editior with python programming.

2. LITERATURE REVIEW

Many methods are implemented on iris dataset using different strategies. Some of the authors ideas and implementations in papers are share here.

Fisher's Iris dataset [4] is introduced by Ronald Fisher with multivariate characteristics in his 1936 paper. He developed a linear discriminant model to recognize the species from each other.

Asmita et.al [3] implemented their method can automatically recognize the class of flowers with three approaches are segmentation, feature extraction and classification. Using Neural network, Logistic Regression, Support Vector Machine and K-Nearest Neighbors.

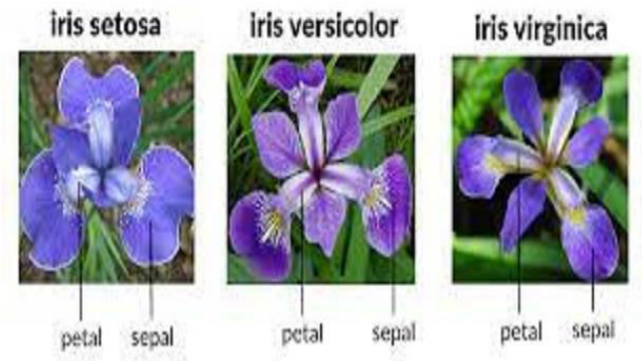
K.Thirunavukkarasu et.al [4] author discussed various methods and used different tools like Scikit and libraries like Numpy, Pandas etc. using all this tools they tested iris dataset flowers.

3. METHODOLOGIES

To implement a machine learning model for recognize the iris specie belongs to. We can use mainly three machine algorithms are Support Vector Machine, Logistic Regression and K-Nearest Neighbor classifier. But we use K-Nearest Neighbor algorithm, this algorithm is used with scikit-learn tool kit based on python.

A. Dataset

In this paper, we take iris dataset from scikit open source project which is already inbuilt. This dataset contains 150 samples in that 50 samples of Setosa, 50 samples of versicolor and 50 samples of virginica



Each sample has four properties; we call *features* in machine learning. The properties are sepal length, sepal width, petal length and petal width. In the below Fig. 1 show each properties measurements. Each row represents the measurement of a flower. Our goal is to build a model that can learn from these measurements and predict species for a new iris. This looks like an example of classification problem in supervised learning.

Sepal_len	sepal_width	petal_len	petal_width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5.0	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1

Fig.1 Samples of Iris dataset

From the above table we see that first five flowers petal width is 0.2 cm and the first flower has longest sepal at 5.1 cm. Now each flower that were measured belongs to which iris species is placed in target array . The target array is a NumPy array type it is one dimensional array. In this the species are encoded as integers 0 to 2. The meaning of the numbers 0 means *setosa*, 1 means *versicolor* and 2 means *virginica*.

B. Data Processing

From this data, we can build a model to predict the species of iris for a new set of measurements. But before we apply this model for new measurements we check whether it works or not. The available data can only predict the correct target for existed measurements. It means it cannot perform well for new data. To build a model and estimate performance, we split the data into two parts. One part is *training set* or *training data* and the other is *test set* or *test data*

.

Scikit-learn contains a function *train_test_split()*. This function extracts 75% of data as *training data* and 25% as *test data*. The below Fig. 2 show that splitting dataset into two parts

```
In [2]: from sklearn.model_selection import train_test_split
        X_train, x_test, y_train, y_test= train_test_split(iris_dataset['data'], iris_dataset['target'], random_state=0)
```

Fig.2 splitting dataset into train set and test set

From above the X_train contains 75%of rows from iris dataset and x_test contains remaining 25%

```
In [3]: print('x_train :{}'.format(X_train.shape))
        print('x_test :{}'.format(x_test.shape))

        x_train :(112, 4)
        x_test :(38, 4)
```

Before building model once best idea is to inspect data in visual. The below Fig.3 is a pair plot of the feature in the training set. In the figure the data points are shown with colors according to the species that iris belong it.

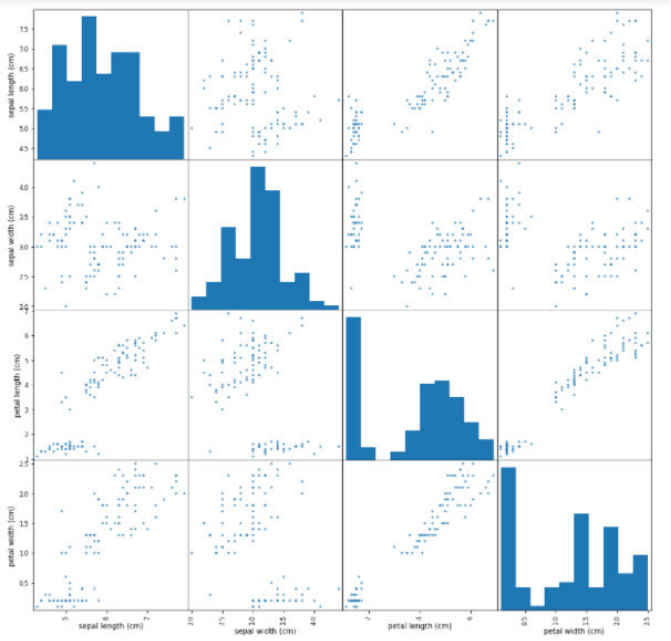


Fig.3 Pair Plot of Iris Dataset

From the above figure it is observed that each classes are well separately shown using sepal and petal measurements.

C. Build Model

We start building a model with K-Nearest Neighbor classifier as show in Fig. 4, which is easy to understand, in this model we consist only training set. To predict any new data point, then it finds out the closest points from the training set. After that it assigns a name or label to new data point.

```
In [6]: from sklearn.neighbors import KNeighborsClassifier
        knn = KNeighborsClassifier(n_neighbors=1)
        knn.fit(X_train,y_train)

Out[6]: KNeighborsClassifier(n_neighbors=1)
```

Fig. 4 K-Nearest Neighbors Model

D. Testing

Now we make predications using this model by taking new data. Example 5cm as sepal length, 2cm as sepal width, 1cm as petal length and 0.2cm as petal width. In the below Fig. 5, shows the new data point belongs to. In our model, Prediction 0 (zero) means its species is *setosa*

Previously, we assumed that species are encoded in integer 0 to 2. 0 means *setosa*, 1 means *versicolor* and 2 means *virginica*

```
In [7]: x_new=np.array([[5,2,1,0.2]])
        print(x_new.shape)

        (1, 4)

In [8]: prediction=knn.predict(x_new)
        print(iris_dataset.keys())
        print('Prediction : {}'.format(prediction))
        print('Target name : {}'.format(iris_dataset['target'][prediction]))

        dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])
        Prediction : [0]
        Target name : [0]
```

Fig. 5 Testing new data example

E. Result

We measured how well the model works by computing accuracy.

```
In [10]: y_pred=knn.predict(x_test)
         print('Test Set Prediction: \n{}'.format(y_pred))
         print('Test set Score: {:}'.format(np.mean(y_pred==y_test)))

         Test Set Prediction:
         [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
          2]
         Test set Score: 0.9736842105263158
```

4. CONCLUSIONS

In this paper, we tried to build a model using K-Nearest Neighbors for this model, the test accuracy is shown as 0.97368421. it means that we made a right prediction for 97% of the irises in the dataset.

5. REFERENCES

[1] S. T. Halakatti and S. T. Halakatti, “Identification Of Iris Flower Species Using Machine Learning,” vol. 5, no. 8, pp. 59–69, 2017.

[2] J. Cutler and M. Dickenson, *Introduction to Machine Learning with Python*. 2020.

[3] Asmita Shukla, Ankita Agarwal, Hemlata Pant, and Priyanka Mishra, “Flower Classification using Supervised Learning,” *Int. J. Eng. Res.*, vol. V9, no. 05, pp. 757–762, 2020.

[4] K. Thirunavukkarasu, A. S. Singh, P. Rai, and S. Gupta, “Classification of IRIS dataset using classification based KNN Algorithm in supervised learning,” *2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018*, pp. 1–4, 2018.

[5] https://www.ibm.com/cloud/learn/supervised-learning

[6] https://en.wikipedia.org/wiki/Iris_flower_data_set

[7] https://www.marsja.se/pandas-scatter-matrix-pair-plot/