Classification of IRIS Dataset using Classification Based KNN Algorithm in Supervised Learning

Thirunavukkarasu K.+
School of Computer Science and Engineering
Galgotias University
Greater Noida, India
+prof.thiru@gmail.com

Ajay S. Singh*
School of Computer Science and Engineering
Galgotias University
Greater Noida, India
*drajay.cse@gmail.com

Prakhar Rai&
School of Computer Science and Engineering
Galgotias University
Greater Noida, India
&raiprakhar5@gmail.com

Sachin Gupta#
School of Computer Science and Engineering
Galgotias University
Greater Noida, India
#sachingupta123211@gmail.com

***Abstract*— Machine learning is about prediction on unseen data or testing data and a set of algorithms are required to perform task on machine learning. There are three types of machine learning are called as Supervised, Unsupervised and Reinforcement learning. In this paper we have worked on supervised learning. We have taken the iris dataset and used K-Nearest Neighbors (KNN) classification Algorithm. Our purpose is build the model that is able to automatically recognize the iris species. Tools used for this in paper are Numpy, Pandas, Matplotlib and machine learning library Scikit-learn.**

Index Terms:— Supervised learning, Classification technique, KNN, Machine learning, Numpy, Pandas, Matplotlib, Scikit-learn, Jupyter, Anaconda.

I. INTRODUCTION

Machine learning is about prediction on unseen data or testing data. In machine learning a computer first learn to perform a task by training dataset. Then the computer perform the same task with the testing data [1]. In Supervised learning we pass both input and output data and the result is already known. Supervised learning is of two types Classification based and Regression based. In this paper we are using classification based supervised learning [2]. KNN is a simple algorithm that stores all available cases and classifies based on a similarity measures (e.g distance function) [3].

The implementation of the model includes six basic steps of machine learning that are:

1. Collect data/prepare data
2. Choose algorithm
3. Creating object of the model
4. Train the model by training dataset
5 .Making prediction on unseen data or testing data
6. Evaluation of the model.

The dataset contain 150 Samples of data that has 3 classes, each contain 50 samples. To train the machine we split the dataset into two Parts training and testing dataset, then the machine will train by training dataset and then it will test on testing dataset. Now we will evaluate the model weather it recognize the iris species accurately or not.

II. PRELIMINARY

As our intention is to design a model that is able to automatically recognize the iris species accurately. So for that we collected/prepared data which involve data preprocessing and splitting of data. Data preprocessing involve handling of missing data, handle of categorical data and handling of feature scaling. Categorical data involves nominal data and ordinal data which can be handle by pandas as well as machine learning and for handling missing data and feature scaling we use pandas and machine learning respectively.[8] Splitting of dataset involves training data and testing data. We shuffle the data so that there is no any particular sequence in training as well as testing dataset. K-Nearest Neighbors is the simplest supervised machine learning algorithm that classifies a data point based on how its neighbors are classified [4].

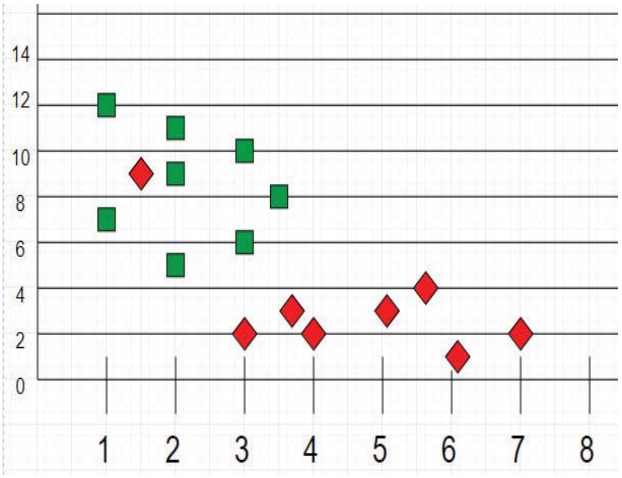

Fig 1.

In this example the following table has two type of dataset.

Initially we have some data (training data), which classifies coordinates into groups identified by an attribute, and another set of data points (testing data) that is allocated by analyzing the training data set and the unclassified points are marked as white.

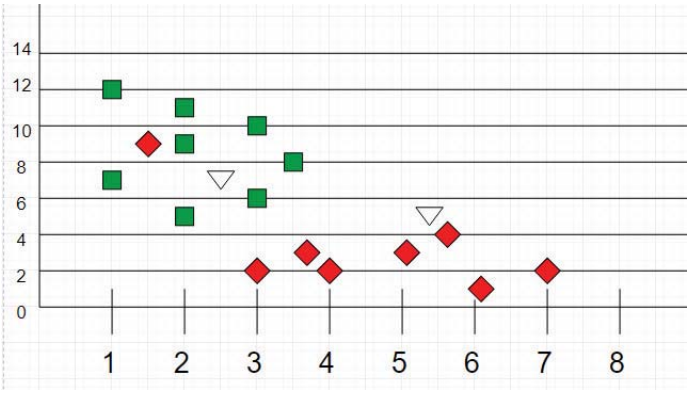

Fig 2.

III. DATASET

In this paper we have taken the iris dataset from Scikit learn (machine learning library) in which iris dataset is already inbuilt.

Dataset Information: The dataset contain 150 sample data in it. The dataset has three classes of data that are Setosa, Versicolor and Virginica each having 50 sample data.

Number of attributes in the datasets are:

4 numeric attributes, predictive attribute (class of iris plant) and the class attribute information.

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm

A. Iris Setosa

Fig. 3. Iris Setosa

B. Iris Versicolor

Fig. 4. Iris Versicolo

C. Iris Virginica

Fig. 5. Iris Virginica

IV. IMPLEMENTATION

We used Anaconda software (Jupyter Notebook) to build the model. Initially we load the iris dataset from Scikit learn library.

TABLE I. ATTRIBUTES OF THE DATASET THAT IS SEPAL LENGTH, SEPAL WIDTH, PETAL LENGTH AND PETAL WIDTH

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4

Above table shows the attributes of the dataset that is Sepal length, Sepal width, Petal length and Petal width. A dataset contain value of all attribute. As the dataset is already preprocessed so we don't need to do data preprocessing. Now we decide target variable that is 0,1,2

TABLE II.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0

Now we shuffle the input and output data and after shuffling the data we split the data into training and testing data. Where training data must be greater than testing data and training data must include all three class of data training data contains 120 rows and testing data contains 30 rows. We need four variable training input, training output testing input and testing output. Now we import model/algorithm K-Neighbors Classifier from SK learn Library and create an object for the KNN classifier. Now with the help of training data we train the model using fit method, after training the model we make prediction on unseen data that is testing data.

```
1 pred=Knn.predict(x_test)
```

pred

```
1 pred
```

```
array([1, 2, 0, 1, 0, 1, 2, 1, 0, 1, 1, 2, 1, 0, 0, 2, 1, 0, 0, 0, 2, 2,
       2, 0, 1, 0, 1, 1, 1, 2])
```

```
1 pred[2]
```

```
0
```

```
1 y_test[2]
```

```
0
```

Fig 6.

V. EVALUATION

We evaluate the model to check wheather the model is properly working or not. And the other purpose to evaluate the model is to modify the model and to get the better result. The metrics that you choose to evaluate your machine learning algorithms are very important.

Choice of metrics influences how the performance of machine learning algorithms is measured and compared. They influence how you weight the importance of different characteristics in the results and your ultimate choice of which algorithm to choose.

Classification problems are perhaps the most common type of machine learning problem and as such there are a myriad of metrics that can be used to evaluate predictions for these problems. In this paper we have used the classification Accuracy metrics.

Classification accuracy is the number of correct predictions made as a ratio of all predictions made.

This is the most common evaluation metric for classification problems, it is also the most misused. It is really only suitable when there are an equal number of observations in each class (which is rarely the case) and that all predictions and prediction errors are equally important, which is often not the case.

```
1 from sklearn.metrics import accuracy_score
```

```
1 accuracy_score(y_test,pred)
```

```
0.9666666666666667
```

```
1 Knn.score(x_test,y_test)
```

```
0.9666666666666667
```

```
1 print('training accuracy :',Knn.score(x_train,y_train))
```

```
training accuracy : 0.975
```

```
1 print('testing accuracy:',Knn.score(x_test,y_test))
```

```
testing accuracy: 0.9666666666666667
```

Fig 7.

VI. CONCLUSION

In this paper we tried to build a model that is able to recognize the iris species accurately on the basis of 3 classes, but some sample provide the misclassified result. Prediction for class0 and calss2 is 100% correct but prediction for class1 is 4% wrong.

REFERENCES

[1] 1.Louridas, P., & Ebert, C. (2016). Machine Learning. IEEE Software, 33(5), 110–115. doi:10.1109/ms.2016.114

[2] https://www.geeksforgeeks.org/supervised-unsupervised-learning/

[3] .https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

[4] .https://www.geeksforgeeks.org/k-nearest-neighbours/

[5] .Kanu Patel1, Jay Vala2, Jaymit Pandya3 1Assist. Prof, I.T Department, BVM Engineering College, V.V.Nagar kanu.patel@bvmengineering.ac.in 2Assist. Prof., I.T Department, GCET Engineering College, V.V.Nagar, jayvala@gcet.ac.in 3Assist. Prof, I.T Department, GC ETEngineeringCollege, V.V.Nagar, jaymitpandya@gcet.ac.in

[6] machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/

[7] Madhusmita Swain1 Sanjit Kumar Dash2,Sweta Dash3 ajjjhhhtnd Ayeskanta Mohapatra41,2Departmentof Information Technology,College ofEngineering and Technology,Bhubaneswar,Odisha,India3Department of Computer Science and Engineering, Synergy Institute of Engineering andTechnology, Dhenkanal, Odisha, India

[8] Been-Chian Chien1, Cheng-Feng Lu2,and Steen J. Hsu31 Department of Computer Science and Information Engineering,National University of Tainan,33, Sec. 2, Su-Lin St., Tainan 70005, Taiwan, R.O.C.bcchien@mail.nutn.edu.tw 2 Department of Information Engineering, I-Shou University,Kaohsiung 840, Taiwan, R.O.C. phon@seed.net.tw 3 Department of Information Management, Ming Hsin University of Science and Technology,

1 Hsin-Hsing Road, Hsin-Fong, Hsin-Chu, Taiwan 304, R.O.C