JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

COMPARISON ON IRIS DATASET USING CLASSIFICATION TECHNIQUES

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

[1]Assistant Professor, [2]Assistant Professor

[1,2] Department of Computer Science

[1]Chaitanya Deemed to Be University, Hanamakonda, Telangana, India

Abstract : Many classification techniques are implemented on different datasets in Machine Learning. This paper gives an idea on different classification techniques implementation and comparison with example an Iris Species Dataset. First dataset is preprocessed and categorized into two parts training set and test set then techniques like K-Nearest Neighbors, Decision Tree, Support Vector Machine(SVM) and Random Forest are used. Finally, accuracy of different techniques is compared.

keywords - Machine Learning, K-Nearest Neighbors, Decision Tree, SVM, Random Forest

I. INTRODUCTION

In current technologies, Machine Learning is a subset 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 i.e. Supervised Learning and Unsupervised Learning. Supervised learning as the name itself says supervised by someone. It is a study 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, different machine learning models are built on 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 with different modeling algorithms. 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.

II. LITERATURE REVIEW

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

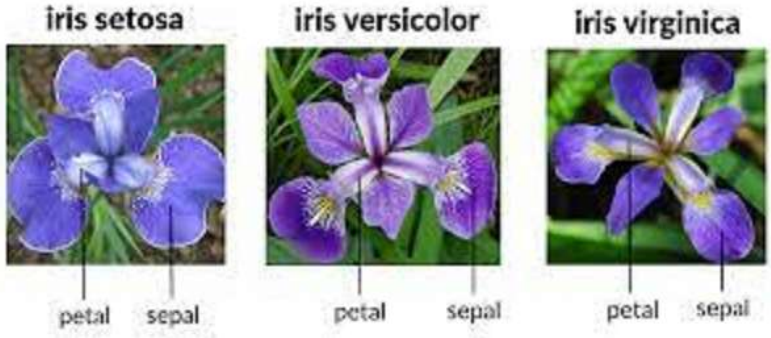
Fisher's Iris dataset [6] is introduced by Ronald Fisher with multivariate characteristics in his 1936 paper. He developed a linear discriminant model to recognize the species from each other. Asmita et.al [3] implemented their method can automatically recognize the class of flowers with three approaches 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.

III. METHODOLOGIES

To implement different machine learning models for recognizing which model is giving accuracy result in identifying iris specie belongs to. We use mainly four machine algorithms like Decision Trees, K-Nearest Neighbor, Support Vector Machine(SVM), and Random Forest classifier. But we use all these four supervised learning algorithms 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 length	sepal width	petal length	petal width	species
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
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

[150 rows x 5 columns]

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

```
from sklearn.model_selection import train_test_split
X_train, x_test, y_train, y_test= train_test_split(iris['data'], iris['target'], test_size=0.25)
```

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.

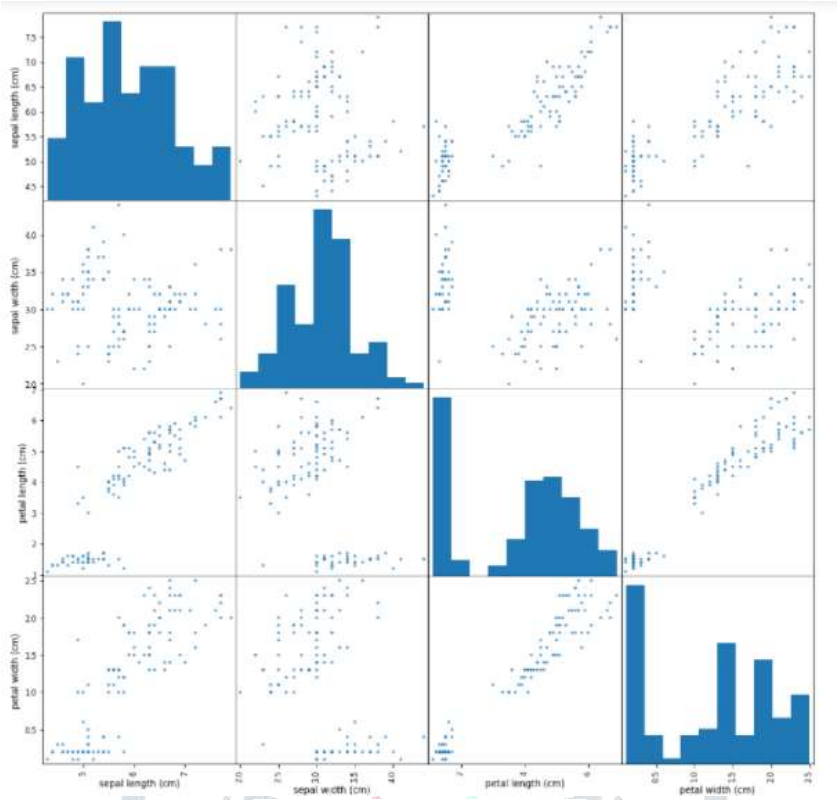


Fig.3 Pair Plot of Iris Dataset

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

C. Comparison

Now we start building a model with different machine learning algorithms after training set and perform prediction on test set. After training, we check the accuracy using actual and predicted value.

First we see the accuracy of a Decision Tree algorithm in the given below Fig.4. it shows the classification, prediction and testing accuracy.

```
dtclf=DecisionTreeClassifier()
dtclf.fit(X_train,y_train)
y_pred=dtclf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9736842105263158
```

Fig. 4 Decision Tree Algorithm

Second, we use K-Nearest Neighbors algorithm in the given below Fig.5 it shows the classification, prediction and testing accuracy.

```
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train,y_train)
y_pred=knn.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9473684210526315
```

Fig. 5 K-Nearest Neighbors Algorithm

Third, we use Support Vector Machine(SVM) algorithm in the given below Fig.6 it shows the accuracy of this model

```
clf=svm.SVC(kernel='linear')
clf.fit(X_train,y_train)
y_pred=clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 1.0
```

Fig. 6 Support Vector Machine Algorithm

Fourth, similarly like other algorithms we use Random Forest algorithm to test the accuracy on the trained set. It is shown in the Fig.7

```
rdclf=RandomForestClassifier(n_estimators=50)
rdclf.fit(X_train,y_train)
y_pred=rdclf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9473684210526315
```

Fig. 7 Random Forest Algorithm

D. Testing

From the above four different algorithms, we have tested the accuracy of model with 75% training set and 25% test set. The observed accuracy of each model is shown below in table format Fig.8

Sno	Algorithms	Accuracy
1	Decision Tree	0.97
2	K-Nearest Neighbors	0.94
3	Support Vector Machine	1.0
4	Random Forest	0.94

Fig. 8 Different Algorithms Accuracy

As per the above table it is tested on 75% training set and 25% test set. But we consider 70% training set and 30% test set in Test I. after we take 80% training set and 20% test set in Test II. At last we observer that percentage of training set and test set is to be take more or less based on the accuracy we get.

Test I

Training set 70% and Test set 30%

Sno	Algorithms	Accuracy
1	Decision Tree	0.88
2	K-Nearest Neighbors	0.93
3	Support Vector Machine	0.95
4	Random Forest	0.93

Test II

Training set 80% and Test set 20%

Sno	Algorithms	Accuracy
1	Decision Tree	0.96
2	K-Nearest Neighbors	1.0
3	Support Vector Machine	1.0
4	Random Forest	1.0

E. Result

After performing classification, training and testing on iris species dataset with various models with different percentages we understand the Support Vector Machine is the best model in giving accuracy result.

In the table training set is represented as A and test set is represented as B

Sno	Algorithms	A=75%, B=25%	A=70%, B=30%	A=80%, B=20%
1	Decision Tree	0.97	0.88	0.96
2	K-Nearest Neighbors	0.94	0.93	1.0
3	Support Vector Machine	1.0	0.95	1.0
4	Random Forest	0.94	0.93	1.0

IV CONCLUSIONS

In this paper, we tried different machine learning models with different percentages of training set and test set. Overall it is observed that from the above Test I, Test II and Default Test results in the table shows that the Support Vector Machine algorithm is giving the best accuracy then other algorithm

.

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/

[8] https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[9] https://www.datacamp.com/community/tutorials/random-forests-classifier-python