Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 4, April 2020, pg.27 – 35

Multiclass Classification with Iris Dataset using Gaussian Naive Bayes

Zainab Iqbal[1]; Manoj Yadav[2]
[1]Department of Computer Science and Engineering, Al-Falah University, India
[2]Department of Computer Science and Engineering, Al-Falah University, India
zainaishna1234@gmail.com [1]; manoj200.yadav@gmail.com [2]

Abstract— A prominent subset of artificial intelligence is machine learning which in today's modern era, is all around us. A model is created in machine learning based on training data and it is predicted that whether the inferences made were correct .Thus the essence of machine learning lies in data extraction and then predictions. It assists a computer to be programmed by self-learning and thereby improve its performance at a specific task. Supervised machine learning tasks primarily include classification for which various algorithms have been applied so far. In this paper, we apply a supervised learning algorithm such as Gaussian Naïve Bayes to classify the species of an Iris flower based on the length and width of their sepals and petals. The performance of the classifier is then tested in terms of its accuracy and classification metrics.

Keywords—Effectiveness Measures, Gaussian Naïve Bayes, Iris Dataset, Supervised learning, Multiclass classification

I. INTRODUCTION

Machine learning approach has a vital role in classification. Classification algorithms come under the category of supervised machine learning concept which fundamentally categorizes a set of data into classes. We present a multiclass classification for the Iris dataset through implementation of a supervised machine learning algorithm Gaussian Naive Bayes which determines the accuracy and performance for prediction of the class of an Iris flower. The dataset that we have used for our research is based on the version present in the UCI machine learning repository as mentioned in [1]. This data set consists of 3 classes of 50 instances each, where each class refers to a type of an Iris plant. The classes into which it is classified are Iris setosa, Iris versicolor and Iris virginica. Python is used along with machine learning on the Iris dataset to facilitate the classification.

II. METHODOLOGY

The method and steps in implementation are discussed in this section along with the dataset used and algorithm applied**.**

A. The Data Set

The data set chosen for this work is collected from the UCI Machine Learning repository which contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Based on Fisher's linear discriminant model[2], this data set has became a significant dataset for many statistical classification techniques in machine learning . Our work focusses to predict which class the Iris flower belongs to by extraction of data from this dataset. The task is to model the probabilities of class membership, based on the flower features. This dataset is also included in the machine learning package Scikit-learn. The scikit-learn [3]library of python comes with the inbuilt dataset for Iris dataset stored in a 150x4 numpy.ndarray. The rows being the samples and the columns being: Sepal Length, Sepal Width, Petal Length and Petal Width.

The Iris dataset consists of :

1) *Samples*

   There are 150 samples in total with 50 instances belonging to each class.

2) *Three labels(class)*:These are the three species of Iris which are Iris setosa, Iris versicolor and Iris virginica. The different species of an Iris flower are illustrated in Figure 1, Figure 2 and Figure3.

Fig 1 Iris setosa

Fig 2 Iris versicolor

Fig 3 Iris virginica

3) Four features:

Sepal length in cm, Sepal width in cm, Petal length in cm, Petal width in cm are the attributes or features which are mentioned in the dataset.

A scatter plot in computer statistics is a two-dimensional data visualization that uses dots to represent the values obtained for two different features .One variable is plotted along the x-axis and the other plotted along the y-axis. The following scatter plot in Figure 4 gives the two-dimensional representation of the Iris dataset for features sepal length and sepal width, both measured in centimeters.

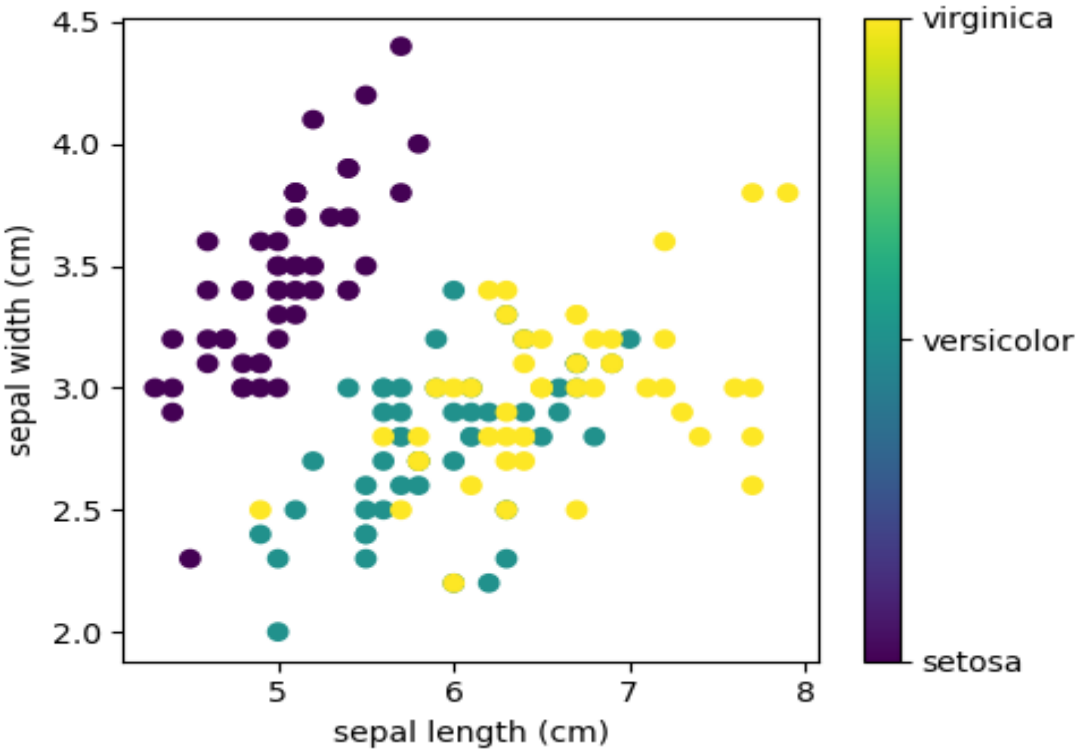

Fig 4 Two dimensional view of the Iris dataset

Through the analysis of flower attributes, a scatterplot matrix can also help to distinguish between different classes of Iris. A scatter plot matrix is a grid or matrix of scatter plots where each scatter plot resembles the relationship between a pair of variables, which enables many relationships to be represented in one chart. Pandas library of python has DataFrame method to frame the data and takes data rows in numpy format and column names to frame the data as a table. The scatter graph of species are plotted and all features are compared to each other in Figure 5 which is a scalar matrix. This type of representation helps to clearly model the relationship between variables and predict the classes.

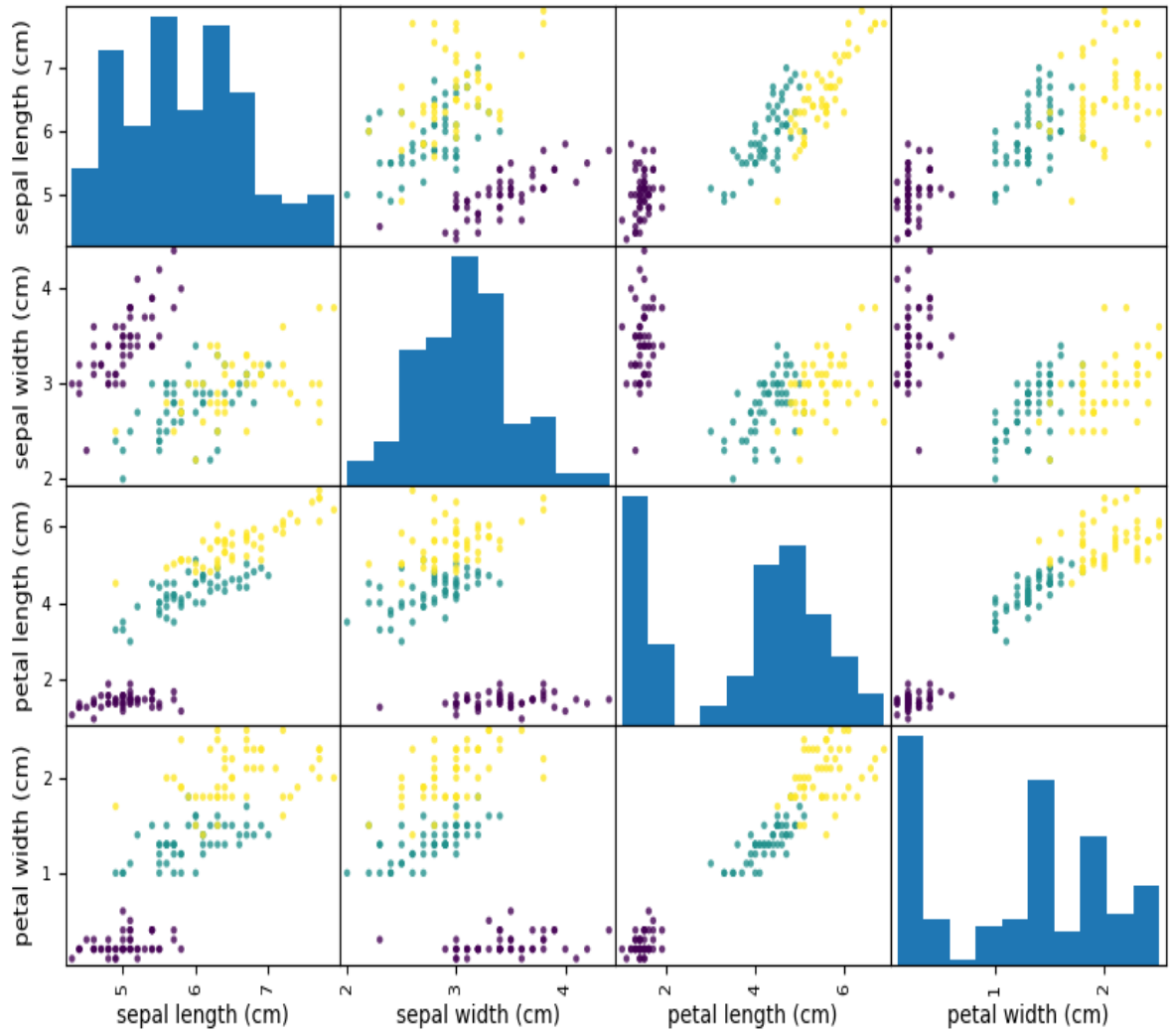

Fig 5 Scalar Matrix for the Iris dataset

B. Naive Bayes Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the assumption of conditional independence between every pair of features given the value of the class variable. This model assumes that all the features are independent[4].

The terminology used in the Bayesian method of probability is as follows:

A is called the proposition

B is called the evidence

P(A) is called the prior probability of proposition

P(B) is called the prior probability of evidence.

P(A|B) is called the posterior probability

P(B|A) is the likelihood.

A posterior probability is calculated in which the probability of an event A is dependent on the probability of occurrence of event B

$$\text{Posterior} = \frac{(\text{Likelihood}).(\text{Propositionpriorprobability})}{\text{Evidence prior probability}}$$

Naïve Bayes classifier uses bayes theorem to predict the probability that a given set of features is a part of particular label.

$$P(\text{label/features}) = \frac{P(\text{label}) * P(\text{features/label})}{P(\text{features})}$$

Where P(label) = prior probability of label

P(features/label) = prior probability that feature set is classified as label

P(features) = prior probability that feature set will occur.

Classification algorithm is one of the most significant techniques in data mining [5].In classification, the goal of a learning algorithm is to construct a classifier given a set of training examples with class labels[6].Naive bayes is a simple but efficient algorithm for predictive modeling. It is a classification algorithm for binary (two-class) as well as multiclass classification problems Also, a high degree of accuracy can be achieved using Naïve Bayes for text classification [7].

C. Gaussian Naive Bayes

The classifier that we have used for our model is Gaussian Naive Bayes, which a variation of the traditional Naïve Bayes algorithm discussed in this paper. In Naive Bayes, the probabilities for input values for each class using a frequency is calculated. With real-valued inputs in a Gaussian distribution, the mean and standard deviation of input values for each class can be calculated to summarize the distribution. It infers that in addition to the probabilities for each class, we must also store the mean and standard deviations of each input variable for each class

D. Effectiveness Measures

Four effective measures that have been used in this study are based on confusion matrix output, which are True Positive(TP), False Positive(FP), True Negative (TN), and FalseNegative(FN).

Positive (P) : Observation is positive

Negative (N) : Observation is not positive

True Positive (TP) : Observation is positive and is predicted to be positive.

False Negative (FN) : Observation is positive, but is predicted negative.

True Negative (TN) : Observation is negative and is predicted to be negative.

False Positive (FP) : Observation is negative, but is predicted to be positive.

E. Confusion matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. The correctness of a classification can be assessed by calculating the number of correctly identified class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that were not recognized as class examples (false negatives). These four counts are applied to construct a confusion matrix [8] as shown in Table 1.This table is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. Most performance measures are computed from the confusion matrix.

TABLE I
CONFUSION MATRIX FOR A CLASSIFIER

	Actual Positive	Actual Negative
Predicted Positive	True Positive(TP)	False positive(FP)
Predicted Negative	False negative(FN)	True Negative(TN)

F. Performance Indicators:

The following evaluation indexes are used to measure the performance of our model which are calculated using the effectiveness measures as discussed above in this paper.

1) Accuracy: The portion of all true predicted instances against all predicted instances is known as accuracy of a classifier.

$$\text{Accuracy(A)} = \frac{(TP+TN)}{(TP + TN + FP + FN)}$$

2) Precision: Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

$$\text{Precision(P)} = \frac{TP}{(TP+FP)}$$

3) Recall: Recall is the ability of a classifier to find all positive instances For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

$$\text{Recall(R)} = \frac{TP}{(TP+FN)}$$

4) F1-score: A harmonic average of precision and recall is known as F1-score.

$$\text{F-Measure(Micro-averaging)} = \frac{2.(P.R)}{(P+R)}$$

III.RESULTS AND DISCUSSION

The implementation of a supervised learning algorithm such as Naive Bayes, more particularly Gaussian Naive Bayes gives an accuracy of 95% which depicts its efficiency for classification. The performance indicators for the classification task are represented in Table II through a classification report. Here 0, 1, 2 represents the target classes Iris setosa, Iris Versicolor and Iris Virginica respectively. The main classification metrics such as precision, recall, accuracy and F1 score are shown with respect to each class. These metrics are calculated using the elements of the confusion matrix as represented in Table I. For the dataset which we are working on, positive and negative corresponds to the predicted classes.

TABLE III
CLASSIFICATION REPORT FOR IRIS DATASET WITH GAUSSIAN NAIVE BAYES

	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	50
1	0.96	0.90	0.93	50
2	0.91	0.96	0.93	50
Macro avg	0.95	0.95	0.95	150
Weighted avg	0.95	0.95	0.95	150

The confusion matrix in Table III evaluates the performance of the Gaussian Naive Bayes classifier on the Iris dataset which shows that accurate predictions are made for the species of Iris flower.

TABLE IIIII
CONFUSION MATRIX FOR IRIS DATASET WITH GAUSSIAN NAIVE BAYES

	Iris Setosa	IrisVersicolor	Iris Virginica
Iris Setosa	50	0	0
Iris Versicolor	0	45	5
Iris Virginica	0	2	48

CONCLUSION

The work in this paper includes loading the Iris dataset, installation of the required libraries such as sklearn, numpy and matplotlib. A scatter plot and scatter matrix is created which gives a bivariate analysis of the Iris dataset. Gaussian Naive Bayes algorithm has been used in this paper along with python to classify the species of an Iris flower. An accuracy has been achieved of about 95% which shows that Gaussian Naïve Bayes algorithm is efficient for supervised learning based classification. Future scope includes classification with other datasets and evaluation of performance with other supervised learning algorithms so as to explore more concepts of machine learning.

REFERENCES

[1]. UCI Machine Learning Repository. Iris Data Set.

[2]. Fisher, R.A. (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7, 179-188.

[3]. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[4]. Kang Hanhoon, Yoo Seong Joon, Han Dongil., "Senti-lexicon and improved Naı¨ve Bayes algorithms for sentiment analysis of restaurant reviews", Expert Syst Appl ,39:6000–10, 2012.

[5]. Wu, Yuanyuan & He, Jing & Ji, Yimu & Huang, Guangli & Haichang, Yao & Zhang, Peng & Xu, Wen & Guo, Mengjiao & Li, Youtao. (2019). Enhanced Classification Models for Iris Dataset. Procedia Computer Science. 162. 946-954. 10.1016/j.procs.2019.12.072.

[6]. Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. 2.

[7]. M. N. M. Ibrahim and M. Z. M. Yusoff, "Twitter sentiment classification using Naive Bayes based on trainer perception," *2015 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, Melaka, 2015, pp. 187-189. doi: 10.1109/IC3e.2015.7403510.

[8]. Sokolova, Marina & Lapalme, Guy. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management. 45. 427-437. 10.1016/j.ipm.2009.03.002.

