IRIS SPECIES RECOGNITION: AN ANALYSIS USING PYTHON AND MACHINE LEARNING ALGORITHMS

Yanamadala Ujjwala[1], Mohana Priya[*1]
[1,*1]School of Computing
SASTRA Deemed University, Thanjavur, India

ABSTRACT

The objective of this research work is to classify various iris species and to cluster them for easy identification from different groups. In this research work, machine learning algorithms are implemented both in Weka and Python to compare their performance analysis in both tasks. Iris species recognition is done based on the classification algorithms such as K-Nearest Neighbor, Naïve Bayes Algorithm, Support Vector Machine, Logistic Regression, Decision tree and Random Forest algorithms. K-Means and Hierarchical Clustering algorithms are used to create clusters of same iris species. The main focus of this research work is to apply classification and clustering algorithms on Iris dataset and to have visual representations of clusters.

KEYWORDS

Machine learning, Data mining, Python, Classification, Clustering

1.INTRODUCTION

Classification is identifying a new category of observations based on training data. Classification algorithms classify datasets into various classes based on many characteristics. Image recognition is playing a crucial role even in medical fields by identifying the problem through attained X-ray pictures, radios and scans and also in facial recognition. Due to wide usage, having high accuracy is crucial.Data mining is also known for extracting rules while using big data[1] and determining outcome. Data mining has been around since 1930 and Machine learning was introduced in 1950. This research work concentrates on comparing the classification accuracies obtained by making use of Machine Learning and Data Mining tool weka . Learning algorithms are classified into supervised and unsupervised. Supervised learning consists of classification and regression algorithms. Clustering algorithms are classified as unsupervised.For this experiment Iris flower dataset has been used . It consists of 50 samples for each species:setosa,versicolor,virginica.

Classification algorithms works based on training and testing the labeled data whereas clustering works on predicting using unlabeled data.In this experiments classification algorithms such as KNN,NAIVE BAYES,SVM,LOGISTIC REGRESSION,RANDOM FOREST,DECISION FOREST and clustering algorithms such as KMeans and HIERARCHICAL clustering are implemented. They have been compared by taking the standard metrics into consideration such as accuracy,precision,recall and f1-score. Prediction is also done after training and testing the model.This research contributed in analysis of iris species recognition by making use of classification and clustering algorithms.The highlights of research are the results of the ML algorithms are compared when implemented in python and in a data mining tool named weka, sort out the most reliable procedure out of implementing in python or in data mining tool,WEKA and proposing the better method.

The article is organized in such a way that Section 2 discusses literature survey, Section 3 displays the block diagram of implementation, Section 4 explains the experimental setup that has been used, Section 5 showcases and compares the results obtained, Section 6 discusses conclusion and future scope.

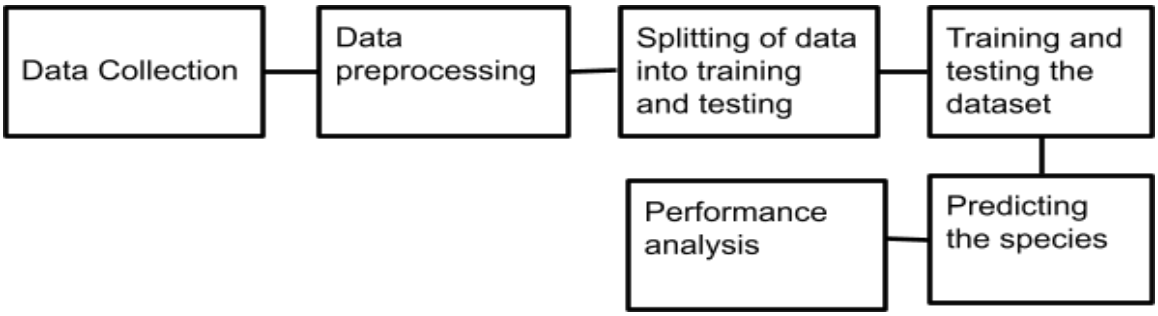
2.LITERATURE REVIEW

Joylin Priya Pinto et. al.[2] focused on implementing the SVM, KNN and Logistic Regression methods on the IRIS dataset. They used cross validation techniques to get accuracy and compared them. They proposed that the SVM classification method is more effective than other methods. Poojitha V et. al.[3] implemented optimal clustering of data items of IRIS dataset using a neural network by specifying the number of clusters,instead of random selection of centroids. Ayşe ELDEM et.al.[4] proposed a novel deep neural network model, two hidden layers are used between input and output layer which have 50 and 20 neurons and implemented on IRIS dataset. The classification success rate is 96%. YuanyuanWu et.al.[5] proposed a new decision tree-based ensemble method for classification named GraftedTrees algorithm . Implemented Random Forests, Boosting Tree model,KNN,SMO and Simple Cart to do classification on IRIS dataset and compare results.

Vaishali Arya et.al.[6] proposed a new method for designing fuzzy rule based classifier in a neuro-fuzzy framework. Their proposed method classifies Iris datasets into four classes. Wafaa Mohammed Ali et.al.[7] implemented Convolutional neural networks[8] to process the flowers' images and implemente nine algorithms collectively Logistic regression, decision tree, random forest, KNN, kernel SVM-poly, kernel SVM-rbf, linear SVM, naïve Bayes, and ANN and proposed SVM gives high accuracy. Anshuman Singh et.al.[9] developed a web application for flower species identification using supervised algorithms KNN and SVM. Hedyeh A. Kholerdi et.at.[10] proposed a multiple classifier system which selects a classifier out of NN,SVM and Naive Bayes based on confidence metric.

Yellamma Pachipala et.at.[11] has proposed a IRIS dataset classifier by making use of Random Forest classifier and deployed it to AWS cloud. Lokesh Pawar et.at.[12] has proposed an optimized ensemble model to recognise and categorize the pattern. Decision Tree , OneR, Adaboost, Random Forest and bayesnet models are applied to improve performance. They tried to improve the efficiency of traditional algorithms by applying features of individual algorithms on state of art parameters. R.A.Abdulkadir et.at.[13] has simulated a backpropagation neural network for the classification of Iris dataset and got 100% accuracy.

3.BLOCK DIAGRAM OF IMPLEMENTATION



The first step of implementation is data collection which is collected from UCI ML Repository. Data has been preprocessed by dividing the attributes into dependent and independent variables. independent variables are sepal length,sepal width,petal length,petal width and dependent variable is species. Out of 150 instances train and test has been split with 80:20 ratio. ML algorithms have been trained using a train set and then obtained a predicted test set. Species prediction has been done by providing all the independent variables randomly. Finally algorithms performance is analyzed by comparing predicted test sets and original test sets.

4.DATASET

Iris dataset is collected from UCI Machine Learning Repository. It consists of a total of 150 instances and 4 attributes. It is a multivariate dataset which is specifically for classification. Iris species are labeled as iris-setosa, iris-versicolor and iris-virginica. The attributes include Sepal Length,Sepal Length,Petal Length,Petal Width in cm. Each species consists of 50 instances.

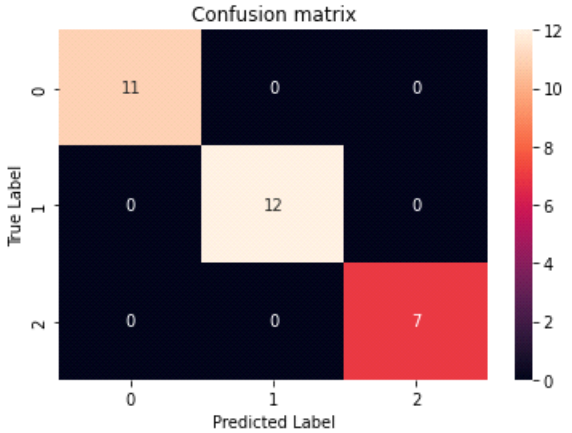
5.EXPERIMENTAL SETUP

5.1. GOOGLE COLAB

In this research work machine learning algorithms are executed in GOOGLE COLAB[17] where PYTHON VERSION 3 is used and for comparative analysis all those machine learning algorithms are also implemented in a data mining tool named WEKA[18]version 3.8.3 where we implemented inbuilt machine learning algorithms on the dataset.The experiment is performed on 11th Gen, Intel Core i5-1135G7 @ 2.40GHz , 2.42 GHz, 8GB RAM personal computer. The research work is experimented using IRIS dataset[19] which is collected from ML UCI REPOSITORY[20].For implementing various machine learning algorithms all SCIKIT LEARN[21] libraries and its packages are imported by making use of 'import' statement . The 'numpy' package is used to perform functions on arrays for concatenating test and predicted values of iris species.For data manipulation and creation of dataframes 'pandas' is used. 'matplotlib.pyplot' package is used to plot graphs to plot confusion matrices,'seaborn' to visualize complex statistical graphs and also provide an aesthetic view. From sklearn.preprocessing 'labelencoder' package is imported which is used to transform categorical data to numerical data. The 'train_test_split' package is imported from sklearn.model_selection used for splitting train and test sets which follow an 80:20 ratio. The performance of all ML algorithms are tested using imported packages from sklearn.metrics such as 'classification_report','confusion_matrix' and 'accuracy_score'. Made use of 'datasets' from sklearn to import datasets . KNN[22] algorithm is implemented by importing 'KNeighborsClassifier' from sklearn.neighbors. Naive Bayes[23] algorithm is implemented by importing 'MultinomialNB' from sklearn.naive_bayes.Logistic Regression[24] algorithm is implemented by importing 'LogisticRegression'from sklearn.linear_model.Random Forest[25] algorithm is implemented by importing 'RandomForestClassifier' from sklearn.ensemble.Decision Tree[26] was implemented by importing 'DecisionTreeClassifier' from sklearn.tree. To give file access for strings 'StringIO' is implemented by importing six library,'Image' was imported from IPython.display to display image.Imported 'export_graphviz' from sklearn.tree is used to plot a decision tree.SVM[27] algorithm is implemented using 'LinearSVC' from sklearn.svm library. The 2-D graph is plotted by data reduction using Principal Component Analysis[28] 'PCA package is collected from sklearn.decomposition. The imported 'plot_decision_regions' from mlxtend.plotting for making predictions for grid values.KMeans[29] algorithm is implemented by importing 'KMeans' from sklearn.cluster, plt.scatter for plotting a scatter graph plt.scatter library is used. For plotting dendrograms[30] and Hierarchical clustering [30] scipy.cluster.hierarchy is imported using SCIPY[31] and 'AgglomerativeClustering' is implemented by importing sklearn.cluster library.

6.RESULTS AND DISCUSSIONS

In this section,the outputs of the machine learning algorithms implemented in python and the outputs of the inbuilt machine learning algorithms of the WEKA tool are analyzed and compared.We considered confusion matrix,ROC curve,precision,recall,F1 score,accuracy as the metrics for comparison. K-Nearest Neighbors algorithm is implemented in python on iris dataset gives the following outputs:



3

Fig 6.1 CONFUSION MATRIX

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	12
2	1.00	1.00	1.00	7
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Fig6.2 CLASSIFICATION REPORT

K-Nearest Neighbors inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 12  0 |  b = Iris-versicolor
  0  0  7 |  c = Iris-virginica
```

Fig 6.3 CONFUSION MATRIX

NAIVE BAYES algorithm is based on the bayes theorem and is given as follows

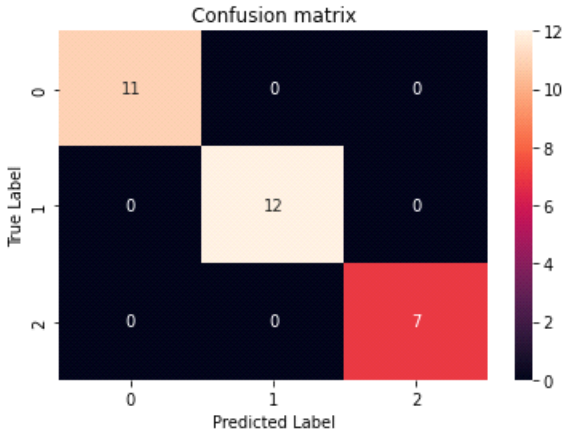


Fig 6.4 CONFUSION MATRIX

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	12
2	1.00	1.00	1.00	7
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Fig 6.5 CLASSIFICATION REPORT

Naive Bayes inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 12  0 |  b = Iris-versicolor
  0  0  7 |  c = Iris-virginica
```

Fig 6.6 CONFUSION MATRIX

	TP Rate	FP Rate	Precision	Recall	F-Measure
	1.000	0.000	1.000	1.000	1.000
	1.000	0.000	1.000	1.000	1.000
	1.000	0.000	1.000	1.000	1.000
Weighted Avg.	1.000	0.000	1.000	1.000	1.000

Fig 6.7 CLASSIFICATION REPORT

SVM is implemented in python on iris dataset gives the following outputs:

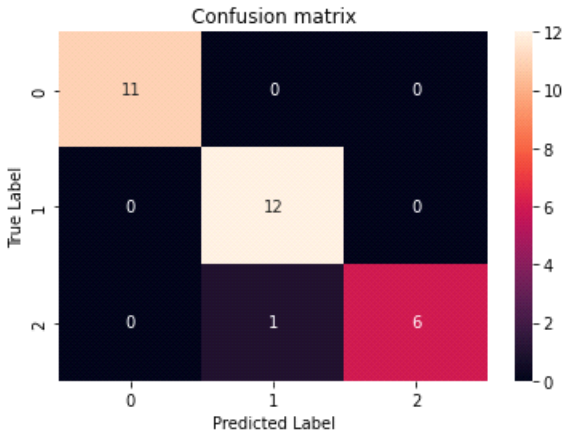


Fig 6.8 CONFUSION MATRIX

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	0.92	1.00	0.96	12
2	1.00	0.86	0.92	7
accuracy			0.97	30
macro avg	0.97	0.95	0.96	30
weighted avg	0.97	0.97	0.97	30

Fig 6.9 CLASSIFICATION REPORT

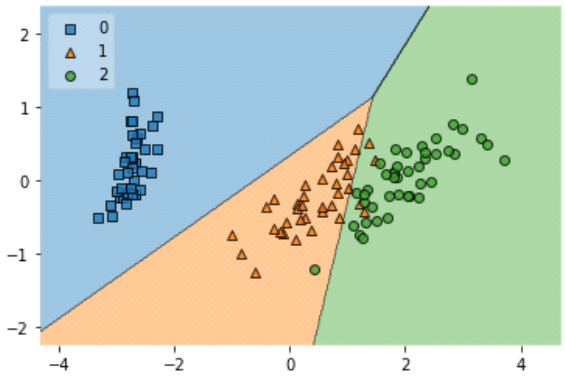


Fig 6.10 LINEAR CLASSIFICATION OF TRAINED SET

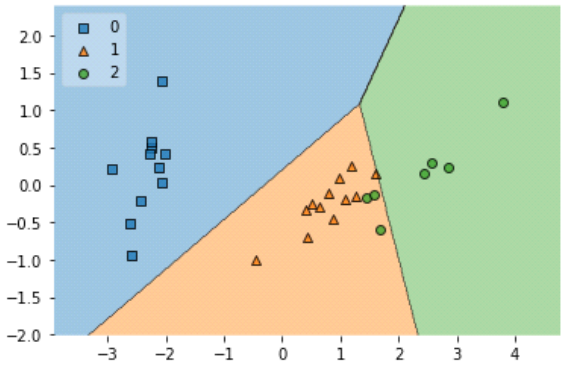


Fig 6.11 LINEAR CLASSIFICATION OF TEST SET

SVM inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 12  0 |  b = Iris-versicolor
  0  1  6 |  c = Iris-virginica
```

Fig 6.12 CONFUSION MATRIX

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure
               1.000    0.000    1.000      1.000    1.000
               1.000    0.050    0.909      1.000    0.952
               0.900    0.000    1.000      0.900    0.947
Weighted Avg.  0.967    0.017    0.970      0.967    0.967
```

Fig 6.13 CLASSIFICATION REPORT

Decision Tree algorithm is implemented in python on iris dataset gives the following outputs:

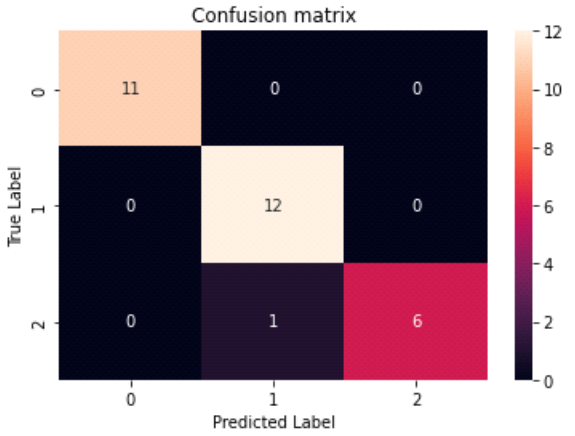


Fig 6.14 CONFUSION MATRIX

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	0.92	1.00	0.96	12
2	1.00	0.86	0.92	7
accuracy			0.97	30
macro avg	0.97	0.95	0.96	30
weighted avg	0.97	0.97	0.97	30

Fig 6.15 CLASSIFICATION REPORT

Decision Tree inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 12  0 |  b = Iris-versicolor
  0  1  6 |  c = Iris-virginica
```

Fig 6.16 CONFUSION MATRIX

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure
               1.000    0.000    1.000      1.000    1.000
               1.000    0.050    0.909      1.000    0.952
               0.900    0.000    1.000      0.900    0.947
Weighted Avg.  0.967    0.017    0.970      0.967    0.967
```

Fig 6.17 CLASSIFICATION REPORT

Random Forest algorithm is implemented in python on iris dataset gives the following outputs:

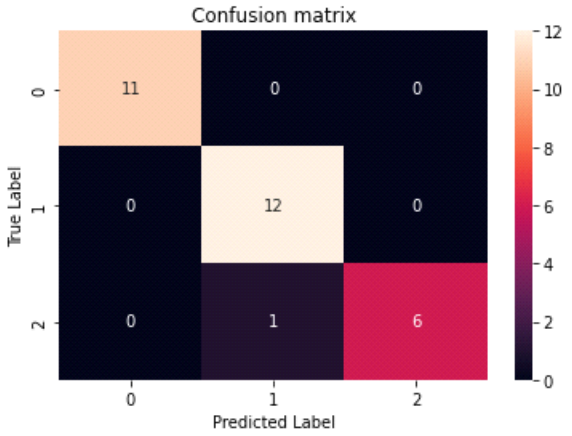


Fig 6.18 CONFUSION MATRIX

```
[25] from sklearn.metrics import accuracy_score
     accuracy_score(y_test, y_pred)

     0.9666666666666667
```

Fig 6.19 ACCURACY SCORE

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       0.92      1.00      0.96        12
           2       1.00      0.86      0.92         7

    accuracy                           0.97        30
   macro avg       0.97      0.95      0.96        30
weighted avg       0.97      0.97      0.97        30
```

Fig 6.20 CLASSIFICATION REPORT

Random Forest inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 12  0 |  b = Iris-versicolor
  0  0  7 |  c = Iris-virginica
```

Fig 6.21 CONFUSION MATRIX

	TP Rate	FP Rate	Precision	Recall	F-Measure
	1.000	0.000	1.000	1.000	1.000
	1.000	0.000	1.000	1.000	1.000
	1.000	0.000	1.000	1.000	1.000
Weighted Avg.	1.000	0.000	1.000	1.000	1.000

Fig 6.22 CLASSIFICATION REPORT

KMeans Clustering algorithm is implemented in python on iris dataset gives the following outputs:

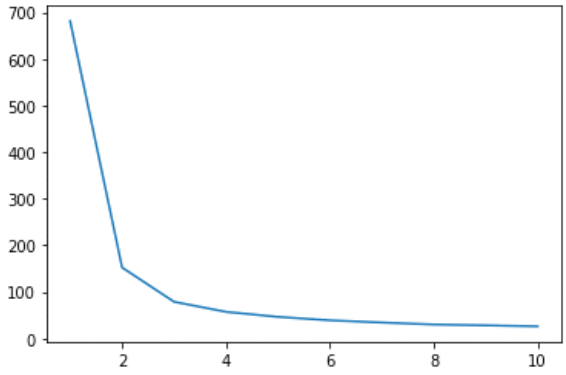


Fig 6.23 ELBOW CURVE

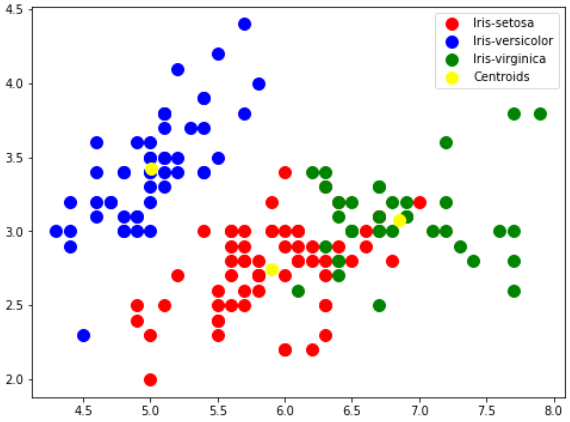


Fig 6.24 CENTROIDS OF THE CLUSTERS

The graph is drawn taking sepal length on x-axis and sepal width on y-axis.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	50
1	0.00	0.00	0.00	50
2	0.95	0.72	0.82	50
accuracy			0.24	150
macro avg	0.32	0.24	0.27	150
weighted avg	0.32	0.24	0.27	150

Fig 6.25 CLASSIFICATION REPORT

KMeans Clustering inbuilt algorithm in weka is executed on iris dataset gives the following outputs:

```
Final cluster centroids:
                                        Cluster#
Attribute               Full Data             0                1
                          (150.0)         (100.0)          (50.0)
=================================================================
sepallength               5.8433           6.262            5.006
sepalwidth                 3.054           2.872            3.418
petallength               3.7587           4.906            1.464
petalwidth                1.1987           1.676            0.244
class                 Iris-setosa Iris-versicolor     Iris-setosa



Time taken to build model (full training data) : 0 seconds

=== Evaluation on test set ===
Clustered Instances

0      19 ( 63%)
1      11 ( 37%)
```

Fig 6.26 CLUSTERING IN WEKA

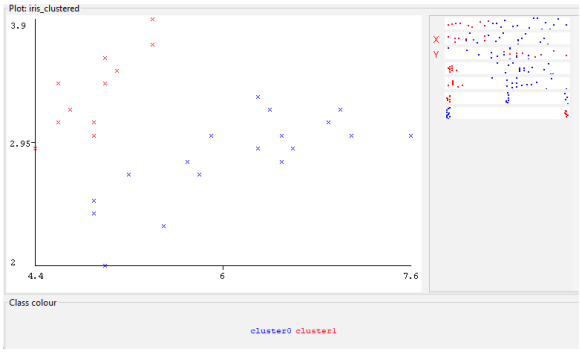


Fig 6.27 CLUSTERED GRAPH

The graph is drawn taking sepal length on x-axis and sepal width on y-axis.

According to fig. 6.28 and fig. 6.29 ,the classification accuracies of KK,NAIVE BAYES,DECISION TREE,SVM,RANDOM FOREST,LOGISTIC REGRESSION CLASSIFICATION when implemented in python are 100%,100%,96.6%,96.6%,96.6%,100% whereas the classification accuracies when implemented in weka are 100%,100%,96.6%,96.6%,100%,100% .From the above results,we observe that except Random Forest all the classification algorithms accuracy,confusion matrix and all performance metrics are same both when implemented using python and weka tool.

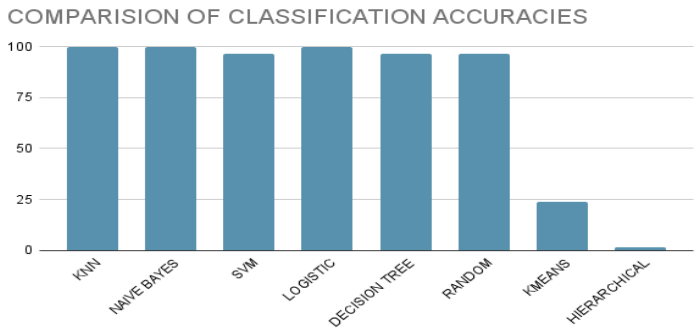


Fig 6.28 ACCURACIES OF ALGORITHMS IMPLEMENTED IN PYTHON

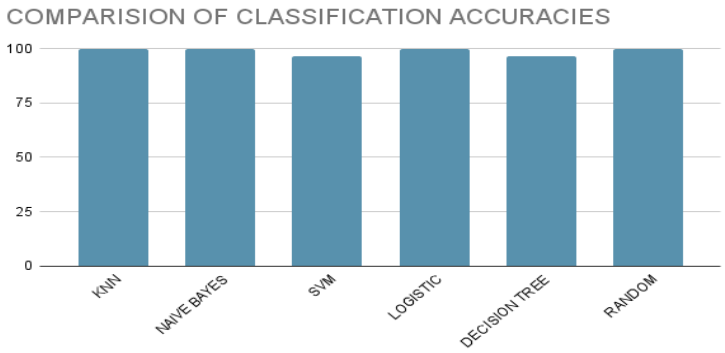


Fig. 6.29 ACCURACIES OF ALGORITHMS IMPLEMENTED IN WEKA

7.CONCLUSION AND FUTURE WORK:

This paper describes the implementation and comparison of algorithms used in the analysis of Iris dataset. KNN , NAIVE BAYES ,SVM,Logistic Regression,Decision Tree,Random Forest,K-Means clustering and Hierarchical clustering are implemented on the dataset. The accuracies of python implemented machine learning algorithms and data mining tool,WEKA algorithms are compared. WEKA gave relatively high accuracy only for the Random Forest algorithm. Clustering is performed well when implemented in python. Further classification algorithms need to be modified for getting higher accuracies.

REFERENCES

[1] What Is Big Data? | Oracle India https://www.oracle.com/in/big-data/what-is-big-data/
[2] J. P. Pinto, S. Kelur and J. Shetty, "Iris Flower Species Identification Using Machine Learning Approach," 2018 4th International Conference for Convergence in Technology (I2CT), 2018, pp. 1-4, doi: 10.1109/I2CT42659.2018.9057891.
[3] Poojitha V, M. Bhadauria, S. Jain and A. Garg, "A collocation of IRIS flower using neural network clustering tool in MATLAB," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 53-58, doi: 10.1109/CONFLUENCE.2016.7508047.
[4] A. ELDEM, H. ELDEM and D. ÜSTÜN, "A Model of Deep Neural Network for Iris Classification With Different Activation Functions," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620866.
[5] Yuanyuan Wu, Jing He, Yimu Ji, Guangli Huang, Haichang Yao, Peng Zhang, Wen Xu, Mengjiao Guo, Youtao Li, Enhanced Classification Models for Iris Dataset,

Procedia Computer Science,Volume 162,2019,Pages 946-954,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2019.12.072.
[6] V. Arya and R. K. Rathy, "An efficient Neuro-Fuzzy Approach for classification of Iris Dataset," 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), 2014, pp. 161-165, doi: 10.1109/ICROIT.2014.6798304.
[7]W. M. Ali, A. A. Abdulredah and A. F. Dakhil, "Web-based AI-IoT Multi Classifiers Model of IRIS Images in Real Live Farm Field," 2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE), 2021, pp. 1-6, doi: 10.1109/ITSS-IoE53029.2021.9615315.
[8] Introduction to Convolution Neural Network - GeeksforGeeks 2022 https://www.geeksforgeeks.org/introduction-convolution-neural-network/
[9] A. Singh, R. Akash and G. R. V, "Flower Classifier Web App Using Ml & Flask Web Framework," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022, pp. 974-977, doi: 10.1109/ICACITE53722.2022.9823577.
[10] H. A. Kholerdi, N. TaheriNejad and A. Jantsch, "Enhancement of Classification of Small Data Sets Using Self-awareness — An Iris Flower Case-Study," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8350992.
[11] Y. Pachipala, H. C. Maddipati, P. S. Udimudi, V. Thota, V. N. Sree and L. R. Burra, "Iris Flower Classification by using Random Forest in AWS," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 2022, pp. 1772-1777, doi: 10.1109/ICICCS53718.2022.9788415.
[12] L. Pawar, J. Singh, R. Bajaj, G. Singh and S. Rana, "Optimized Ensembled Machine Learning Model for IRIS Plant Classification," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), 2022, pp. 1442-1446, doi: 10.1109/ICOEI53556.2022.9776724.
[13]Abdulkadir, Rabiu Aliyu et al. “Simulation of Back Propagation Neural Network for Iris Flower Classification.” (2017).
[14] Conditional probability - Wikipedia 2022 https://en.wikipedia.org/wiki/Conditional_probability
[15] Separating Hyperplanes in SVM - GeeksforGeeks 2021 https://www.geeksforgeeks.org/separating-perplanes-in-svm/
[16] ML | Gini Impurity and Entropy in Decision Tree - GeeksforGeeks 2021 https://www.geeksforgeeks.org/gini-impurity-and-entropy-in-decision-tree-ml/
[17] Google Colab https://colab.research.google.com/
[18] WEKA 3 Witten IH, Frank E (2005). Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition. Morgan Kaufmann, San Francisco.
[19] IRIS dataset https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
[20] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
[21] Scikit-learn Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[22] K-Nearest Neighbours (KNN) Algorithm 2022 https://www.enjoyalgorithms.com/blog/k-nearest-neighbours-in-ml
[23] Naive Bayes Classifier in Machine Learning 2022 https://www.enjoyalgorithms.com/blog/naive-bayes-in-ml
[24] Logistic Regression in Machine Learning 2022 https://www.enjoyalgorithms.com/blog/logistic-regression-in-ml
[25] Random Forest Algorithm 2022 https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm
[26] Decision Tree: A Tree-Based Alorithm in Machine Learning 2022 https://www.enjoyalgorithms.com/blog/decision-tree-algorithm-in-ml
[27] Support Vector Machines: An "out of the box" classifier 2022 https://www.enjoyalgorithms.com/blog/support-vector-machine-in-ml
[28] PCA in Machine Learning: Assumptions, Steps to Apply & Applications | upGrad blog Pavan Vadapalli 2020.
[29] K-means Clustering Algorithm: Applications, Types, and Demos [Updated] | Simplilearn https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm

[30] SciPy - Cluster Hierarchy Dendrogram - GeeksforGeeks 2021 https://www.geeksforgeeks.org/scipy-cluster-hierarchy-dendrogram/
[31] Numpy and Scipy Documentation Author: Eric Jones, Travis Oliphant, Pearu Peterson and others. Title: SciPy: Open Source Scientific Tools for Python. Year: 2001 -