DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY
(AFFILIATED TO JNTUH, GHANPUR (V), GHATKESAR(M), MEDCHAL(D)-501301)

A MINOR PROJECT REVIEW ON

CLASSIFICATION OF IRIS FLOWERS : A MACHINE LEARNING APPROACH BASED ON PETAL AND SEPAL MEASUREMENTS

PRESENTED BY:

N . Sai Vignyan (21RA1A05B0)

N . Nithin Kumar (21RA1A0573)

S . Ravi Teja (21RA1A0597)

INTERNAL GUIDE:

Mr. Ritesh Kumar

INDEX

- ABSTRACT
- INTRODUCTION
- PROBLEM STATEMENT
- LITERATURE SURRVEY
- EXISTING METHODOLOGY
- DRAWBACKS
- PROPOSED SYSTEM
- REQUIREMENTS
- WORKING MODEL
- PSEUDOCODE
- RESULT AND OUTPUTS
- CONCLUSION
- FUTURE SCOPE
- REFERENCES

ABSTRACT

- This study uses machine learning to classify iris flowers based on petal and sepal measurements, improving accuracy over traditional manual methods. It aids species identification in botany, supports breeding in horticulture, and enhances ecosystem monitoring in environmental science. Unlike traditional approaches, which are time-consuming and subjective, this system employs supervised learning to automatically detect patterns, ensuring scalability and robustness. Techniques like cross-validation and hyperparameter tuning further optimize performance. The methodology extends beyond botany to fields such as healthcare, finance, and marketing, demonstrating its versatility in classification tasks.

INTRODUCTION

- In the realm of botanical research, the classification of iris flowers based on their petal and sepal measurements stands as a foundational challenge. This task holds significance not only in the context of botany but also extends its implications into horticulture, agriculture, and environmental science. By utilizing machine learning techniques, this research endeavors to automate and enhance the process of iris flower classification. The proposed system harnesses the power of supervised learning algorithms to discern discriminative patterns from input features, paving the way for efficient species identification. Through meticulous feature extraction and model training on a labeled dataset, the system learns to differentiate between iris species based on their unique petal and sepal characteristics. Moreover, the incorporation of cross-validation and hyperparameter tuning techniques ensures the robustness and reliability of the classification model.

PROBLEM STATEMENT

- Traditional methods for classifying iris flowers heavily rely on manual measurements and expert knowledge, resulting in time-consuming and subjective processes prone to inconsistencies and errors. Moreover, these methods often struggle to handle the complexities of high-dimensional feature spaces and subtle differences between species. The need for a more efficient and accurate classification approach is evident, especially considering the importance of iris species identification in various domains such as botany, horticulture, and environmental science. Thus, the primary problem addressed by this research is to develop a machine learning-based system capable of automating iris flower classification based on petal and sepal measurements while overcoming the limitations of traditional methods

LITERATURE SURRVEY

This review explores recent machine learning techniques for iris flower classification. Corne and Ursani proposed an evolutionary algorithm, while Nauck and Kruse used fuzzy neural networks. Transfer Learning and Adam Deep Learning were applied to improve classification efficiency. Huang et al. achieved a 95% recognition rate using Difference Image Entropy (DIE). Iqbal employed Gaussian Naïve Bayes, achieving 95% accuracy. Mijwil and Abttan used C4.5 decision trees to address overfitting. Other studies compared J48 and Random Forest, with J48 reaching 95.83% accuracy. These approaches highlight advancements in classification methods, improving accuracy, efficiency, and scalability in machine learning applications.

EXISTING METHODOLOGY

- **KNN ALGORITHM**
- K-Nearest Neighbour (KNN) Algorithm for Machine Learning
  - K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
  - K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
  - K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

DRAWBACKS

- **Selecting the Optimal K Value is Complex** – There is no fixed rule to determine the best K value. A small K (e.g., K=1 or 2) can be too sensitive to noise, while a large K may cause misclassification.
- **High Computation Cost** – KNN requires calculating the distance between the new data point and all training samples, making it computationally expensive, especially for large datasets.
- **Sensitive to Noisy Data** – If the training data contains noise or irrelevant features, KNN may struggle to classify new data points accurately.
- **Storage Requirement** – Unlike other algorithms that learn a model from training data, KNN stores all instances, increasing memory usage.
- **Slow Prediction Time** – Since KNN does not have a training phase and makes classifications in real time, the prediction process can be slow, especially with a large dataset.

PROPOSED SYSTEM

- **Logistic Regression Overview**
- **Purpose**: Used for binary classification (e.g., yes/no, 0/1, true/false).
- **Output**: Produces a probability value between 0 and 1 using the sigmoid function.
- **Threshold**: Values above 0.5 are classified as Class 1, and values below 0.5 are classified as Class 0.
- **Curve**: Fits an "S" shaped logistic function instead of a linear regression line.

- **Key Points**
- **Categorical Dependent Variable**: Predicts the outcome of a categorical dependent variable.
- **Probabilistic Output**: Provides probabilistic values between 0 and 1.
- **Logistic Function (Sigmoid Function)**: Maps any real value to a range between 0 and 1.

Advantages:

1. **Interpretability**: Coefficients indicate the strength and direction of relationships, enhancing trust and understanding.
2. **Probabilistic Output**: Estimates probabilities, aiding in risk assessment and strategic planning.
3. **Computational Efficiency**: Suitable for large datasets, real-time, or high-throughput applications due to its simplicity and linear nature.
4. **Robust with Small Datasets**: Performs well even with limited data.
5. **Resistance to Multicollinearity**: Accurate estimates even with correlated independent variables.
6. **Ease of Implementation**: Simple mathematical formulation and intuitive graphical representation, accessible to users without advanced statistical expertise.

REQUIREMENTS

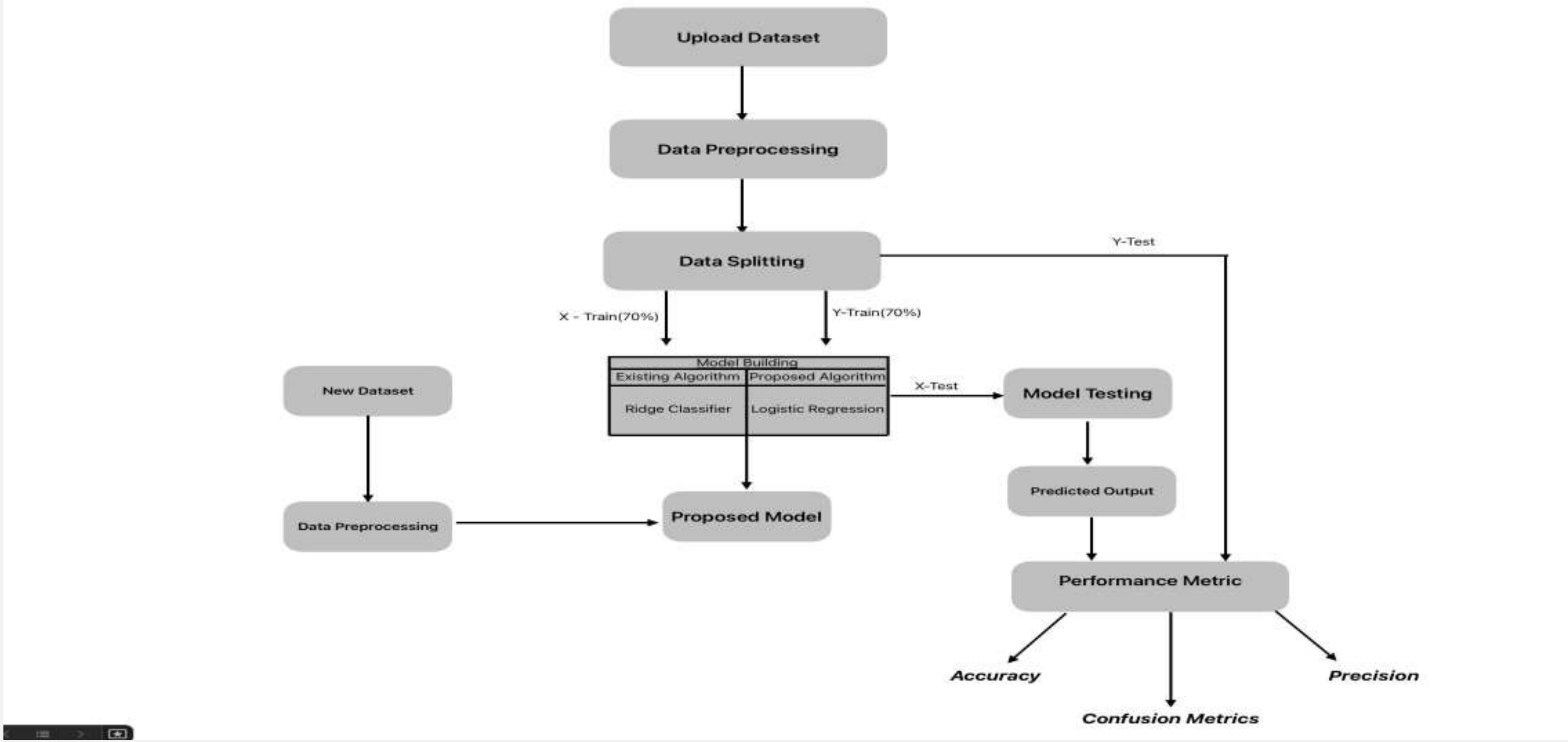
- **SOFTWARE REQUIREMENTS**
- The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.
- Python IDLE 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google colab

HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

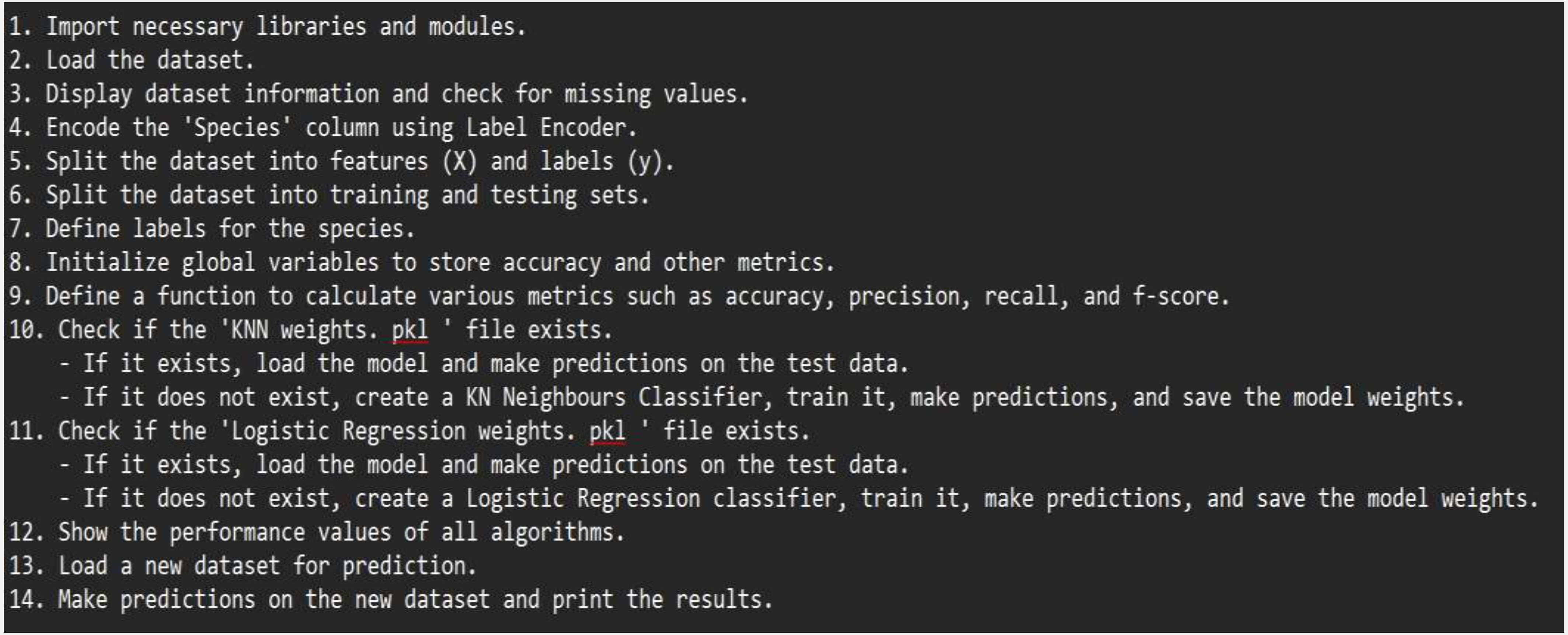
- Operating system : Windows, Linux
- Processor : minimum intel i3
- Ram : minimum 4 GB
- Hard disk : minimum 250GB

WORKING MODEL



Model Building	
Existing Algorithm	Proposed Algorithm
Ridge Classifier	Logistic Regression

PSEUDOCODE

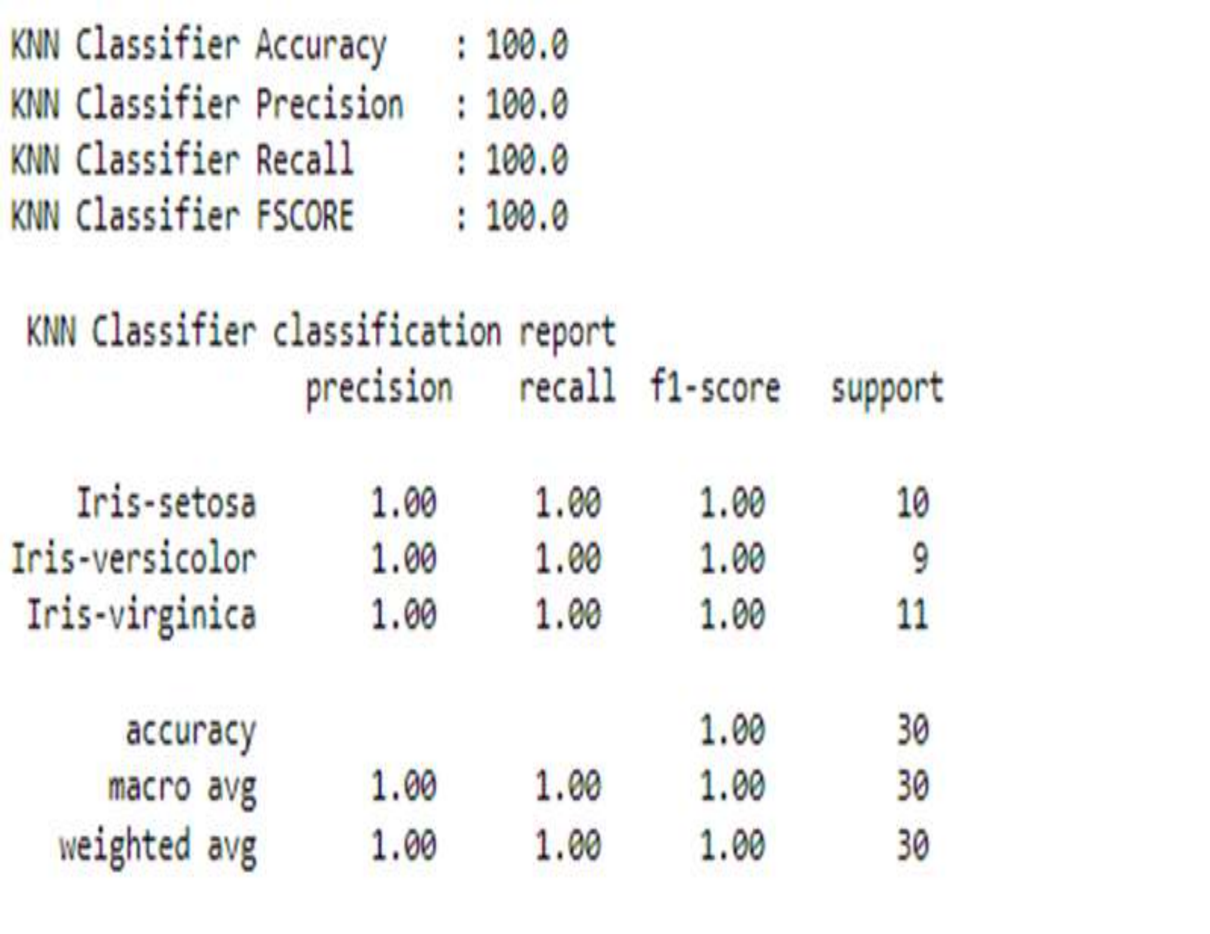


```
1. Import necessary libraries and modules.
2. Load the dataset.
3. Display dataset information and check for missing values.
4. Encode the 'Species' column using Label Encoder.
5. Split the dataset into features (X) and labels (y).
6. Split the dataset into training and testing sets.
7. Define labels for the species.
8. Initialize global variables to store accuracy and other metrics.
9. Define a function to calculate various metrics such as accuracy, precision, recall, and f-score.
10. Check if the 'KNN weights. pkl ' file exists.
    - If it exists, load the model and make predictions on the test data.
    - If it does not exist, create a KN Neighbours Classifier, train it, make predictions, and save the model weights.
11. Check if the 'Logistic Regression weights. pkl ' file exists.
    - If it exists, load the model and make predictions on the test data.
    - If it does not exist, create a Logistic Regression classifier, train it, make predictions, and save the model weights.
12. Show the performance values of all algorithms.
13. Load a new dataset for prediction.
14. Make predictions on the new dataset and print the results.
```

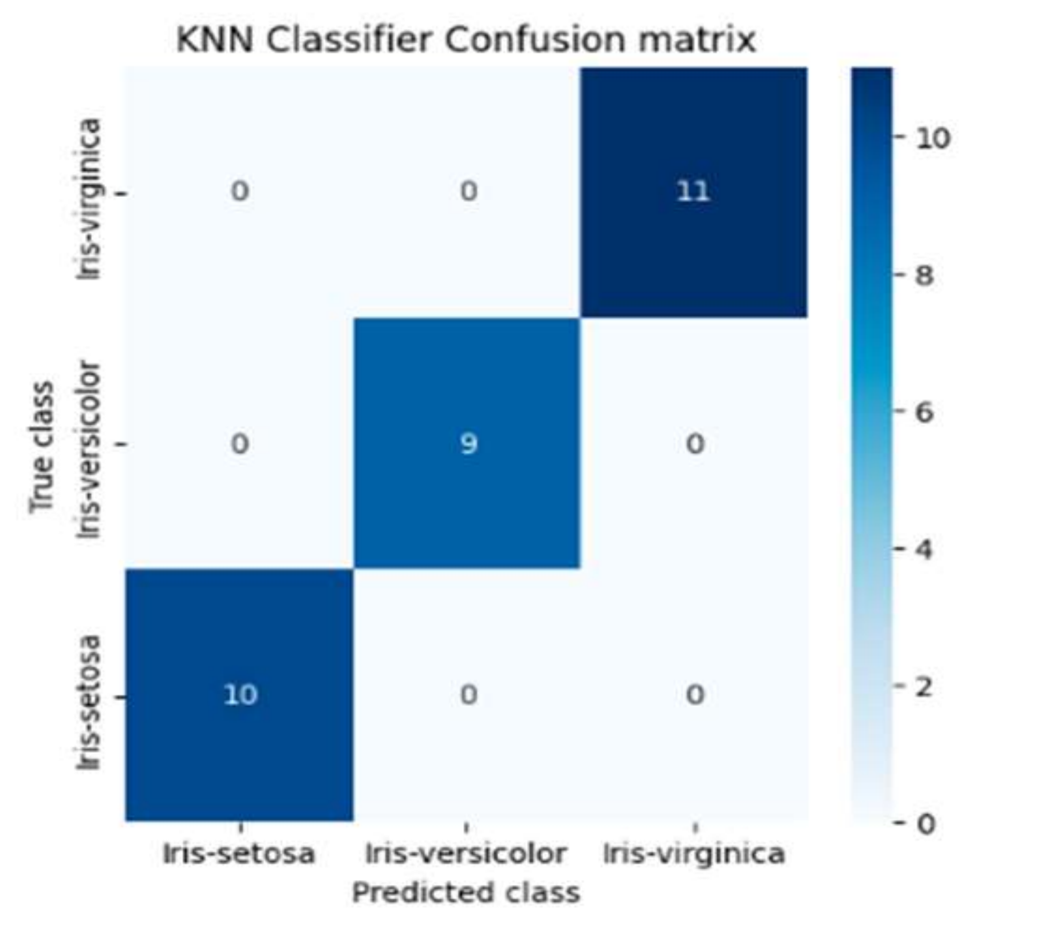
RESULT AND OUTPUTS

- The KNN classifier is a simple algorithm that makes predictions based on the labels of the data points that are closest to the new data point. In the case of the Iris flower dataset, the KNN classifier would likely classify a new data point as Iris-setosa if it was closest to other data points that were labeled as Iris-setosa.
- Logistic Regression Precision: 100.0 - Precision refers to the ratio of true positives to the total number of positive predictions. A value of 1 here means the model identified only Iris flowers (positive cases) and none of the other plants (negative cases) as Iris flowers.
- Both Logistic Regression and KNeighbors Classifier algorithms achieved 100% accuracy on all metrics (precision, recall, F1-score, and accuracy). This suggests that both algorithms performed equally well on this dataset. But we choose logistics Regression as proposed algorithm because KNN have some limitations in this dataset

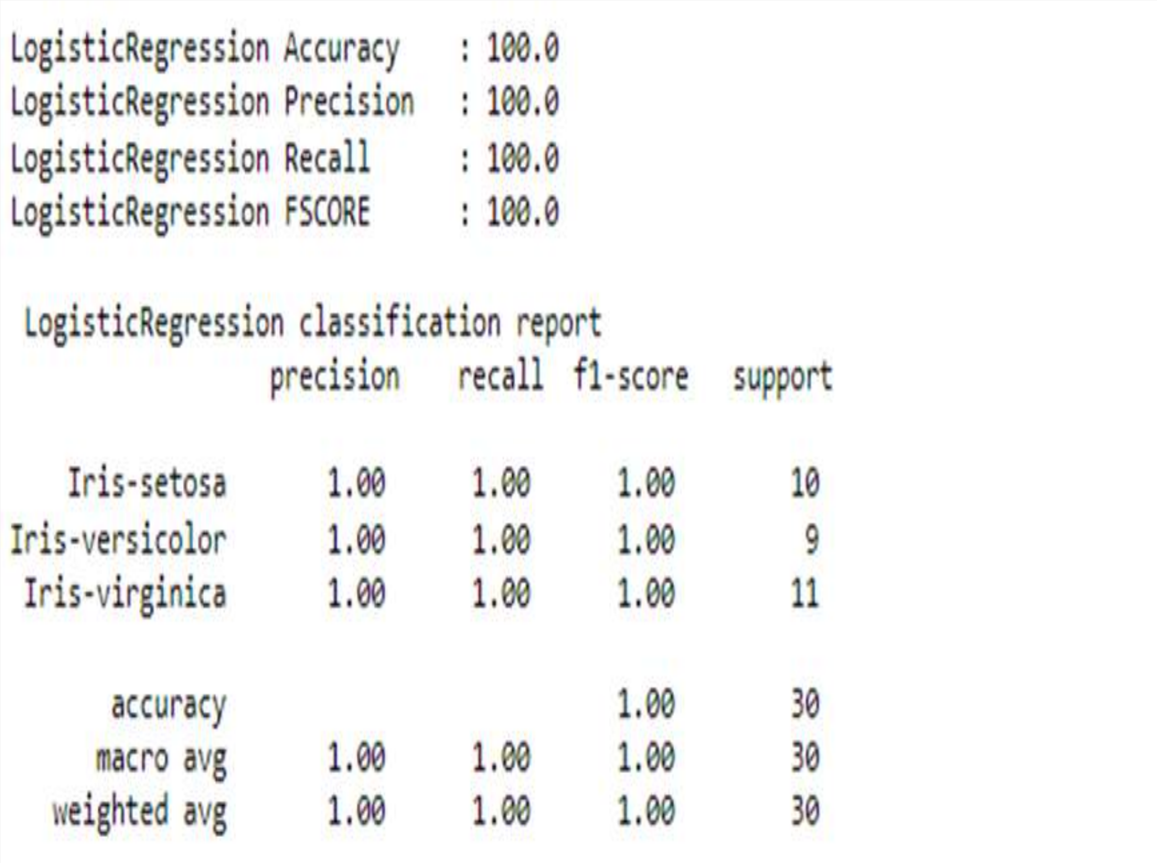
KNN CLASSIFIER



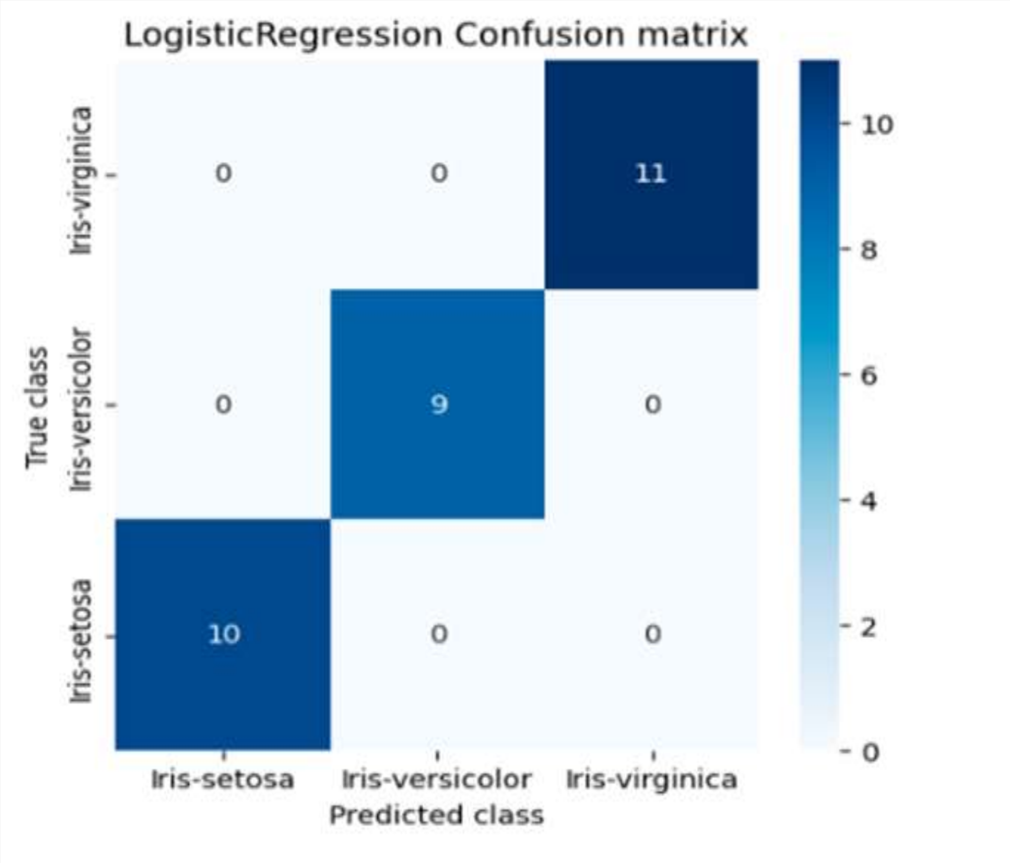
```
KNN Classifier Accuracy    : 100.0
KNN Classifier Precision   : 100.0
KNN Classifier Recall      : 100.0
KNN Classifier FSCORE      : 100.0

 KNN Classifier classification report
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```



LOGISTIC REGRESSION



```
LogisticRegression Accuracy    : 100.0
LogisticRegression Precision   : 100.0
LogisticRegression Recall      : 100.0
LogisticRegression FSCORE      : 100.0

 LogisticRegression classification report
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```



CONCLUSION

- Machine learning techniques for classifying iris flowers based on petal and sepal measurements streamline species identification and extend to fields like horticulture, agriculture, and environmental science. This approach reduces the need for manual measurements and expert knowledge, speeds up the process, and improves scalability. Optimized models ensure reliable classification, which is crucial for ecosystem monitoring and conservation. The adaptability of these methods shows potential for broader applications, such as in healthcare, finance, and marketing. Overall, this study marks significant advancement in botanical research and opens doors to interdisciplinary applications and collaborations.

FUTURE SCOPE

- **Enhanced Model Performance:** Integrate advanced techniques like deep learning to improve accuracy and robustness.
- **Expanding to Other Plant Species:** Apply the methodology to classify other plants with similar traits.
- **Automated Plant Identification Systems:** Develop mobile-based systems for real-time classification and data collection.
- **Integration with Genomic Data:** Combine morphological features with genomic data for better species identification.
- **Applications Beyond Botany:** Adapt techniques for other domains, such as healthcare diagnostics.
- **Real-Time Monitoring and Conservation:** Use models for environmental monitoring and conservation efforts.
- **User-Friendly Tools for Non-Experts:** Create accessible software tools for hobbyists and educators.
- **Collaborative Research Platforms:** Foster innovation through data-sharing platforms for researchers.

REFERENCES

- [1]. Ziauddin Ursani and David W. Corne , “A Novel Nonlinear Discriminant Classifier Trained by an Evolutionary Algorithm” DOI: 10.1145/3195106.3195132

- [2]. Detlef Nauck and Rudolf Kruse, “NEFCLASS-A Neuro-Fuzzzy approach for the classification of data” DOI: 10.1145/315891.316068

- [3] Jing FENG, Zhiwen WANG, Min ZHA and Xinliang CAO, “Flower Recognition Based on Transfer Learning and Adam Deep Learning Optimization Algorithm”. DOI: 10.1145/3366194.3366301

- [4] Roung– Guo Huang, Sang-Hyeon Jin, Jung –Hyun Kim and KwangSeck Hong, “Flower Image Recognition Using Difference Image Entropy”. DOI: 10.1145/1821748.1821868

- [5] Shilpi Jain, V Poojitha, “By Using Neural Network Clustering tool in MATLAB Collecting the IRIS Flower”, Proc. IEEE , vol. 109, 2020.

- [6] M. M. Mijwil and R. A. Abttan, “Utilizing the Genetic Algorithm to Pruning the C4. 5 Decision Tree Algorithm,” Asian J. Appl. Sci. ISSN 2321– 0893, vol. 9, no. 1, 2021.

- [7] Roung– Guo Huang, Sang-Hyeon Jin, Jung –Hyun Kim and Kwang- Seck Hong, “Flower Image Recognition Using Difference Image Entropy”. DOI: 10.1145/1821748.1821868 **Academic Journal of Nawroz University (AJNU), Vol.11, No.4, 2022**

- 475

[8] K R Rathy, Arya Vaishali, “Classification of Dataset using Efficient Neural Fuzzy Approach”, vol. 099, August 2019.

[9] D. Decoste, E. Mjolsness. 2001. “State of the art and future prospects by using Machine Learning”, vol. 320, 2013.

[10] Y. Lakhdoura and R. Elayachi, “Comparative Analysis of Random Forest and J48 Classifiers for ‘IRIS’ Variety Prediction,” Glob. J. Comput. Sci. Technol., 2020

[11] Zebari, D. A., Abrahim, A. R., Ibrahim, D. A., Othman, G. M., & Ahmed, F. Y. (2021). Analysis of Dense Descriptors in 3D Face Recognition. In *2021 IEEE 11th International Conference on System Engineering and Technology (ICSET)* (pp. 171-176). IEEE.

[12] Abdulqadir, H. R., Abdulazeez, A. M., & Zebari, D. A. (2021). Data mining classification techniques for diabetes prediction. Qubahan Academic Journal, 1(2), 125-133.

[13] Ibrahim, D. A., Zebari, D. A., Ahmed, F. Y., & Zeebaree, D. Q. (2021, November). Facial Expression Recognition Using Aggregated Handcrafted Descriptors based Appearance Method. In 2021 IEEE 11th International Conference on System Engineering and Technology (ICSET) (pp. 177-182). IEEE.

Thank you