Classification of Iris Species Using K-Nearest Neighbors

Perlin Precious S. Sasil
College of Computer Science and Engineering
Bachelor of Science in Information Technology
Mandaluyong, Philippines
perlinprecious.sasil@my.jru.edu

Abstract—This paper presents a machine learning study using the Iris dataset to implement a K-Nearest Neighbors (KNN) model. The methodology includes data acquisition, partitioning, model training, and performance validation using metrics such as accuracy and confusion matrices. Results indicate that KNN provides a high-fidelity performance summary for biological classification. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.*

Keywords—K-Nearest Neighbors, Iris Dataset, Scikit-learn, Classification, Performance Summary.

I. Introduction (Heading 1)

This study serves as a technical validation of the K-Nearest Neighbors (kNN) algorithm. kNN is a "lazy learner" that predicts labels based on the similarity (Euclidean distance) between new data points and existing training samples. The Iris dataset, introduced by Ronald Fisher, is utilized to classify 150 flowers into three species: Setosa, Versicolor, and Virginica.

II. Program Implementation (Phase 2)

The implementation is divided into four logical modules to ensure professional code organization and readability.

A. Module 1: Data Acquisition & Preprocessing

Why: To load the biological measurements into the environment and prepare them for mathematical modeling.

How: load_iris() imports the dataset. train_test_split partitions the data, using test_size=0.2 to reserve 20% for validation.[1]

B. Module 2: Model Configuration

Why: To define the logic the model uses to "vote" on a classification.

How: KNeighborsClassifier(n_neighbors=5) sets the algorithm to look at the 5 closest points. knn.fit maps the relationship between features and labels.[2]

C. Module 3: Evaluation Metrics

Why: To quantify the model's predictive success.

How: accuracy_score provides a snapshot of total correct guesses. classification_report details precision/recall per species.[3]

D. Module 4: Visual Diagnostics

Why: To identify patterns in misclassifications.

How: plt.imshow(cm) creates a Confusion Matrix heatmap to visualize where species are being mistaken for one another.

III. Output Analysis (Phase 3)

1. **Purpose:** The program predicts the species of an iris flower based on four numerical measurements (sepal/petal length and width).
2. **n_neighbors Influence:** Higher k values smooth decision boundaries, reducing the impact of outliers but potentially causing "underfitting." Experimentation shows that $k=5$ often yields peak accuracy (~96-100%) for this dataset.
3. **Confusion Matrix:** It reveals that errors typically occur between *Versicolor* and *Virginica* because their measurements overlap. It tells us not just that the model failed, but *specifically* which species it finds confusing. [4]

IV. Research & Government Application (Phase 4)

A. Operational Investigation

Research identifies kNN use in **Healthcare Fraud Detection** (Medicare), **Urban Land-Use Zoning**, and **Disaster Response (Resource Allocation)**. In zoning, kNN predicts categorical land-use (Residential vs. Industrial) based on proximity to geographical markers.

B. Performance Factors

kNN effectiveness varies with **Sample Size**. In disaster response, low-quality, noisy data can significantly decrease the model’s success, as kNN is highly sensitive to outliers.

C. Technical Specification Audit

- **Distance Metric:** Euclidean distance is standard; it is appropriate for continuous, normalized physical measurements.
- **Preprocessing:** Min-Max scaling is often used in literature to ensure that features like "Total Area" don't outweigh "Age."
- **k-Value Justification:** Many studies use the **Elbow Method** to find the optimal k that minimizes error without increasing complexity.

D. Ethical Decision-Making

As a government official, relying on kNN for high-stakes decisions (e.g., fraud) requires caution. While the Confusion Matrix shows high accuracy, kNN lacks **algorithmic transparency** (it doesn't explain *why* a choice was made). For high-stakes decisions, a "Human-in-the-loop" approach is necessary to audit the model's "Black Box" predictions.

V. Conclusion

The implementation confirms that kNN is a robust tool for classification when data clusters are well-defined. However, technical success must be balanced with ethical oversight when applied to government operations.

References

[1] Scikit-learn, "sklearn.datasets.load_iris," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html.

[2] Scikit-learn, "sklearn.model_selection.train_test_split," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[3] Scikit-learn, "sklearn.metrics.accuracy_score," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html.

[4] StackOverflow, "Fixing plt.scatter errors," [Online]. Available: https://stackoverflow.com/questions/73132314/.