ALY 6020:

PREDCTIVE ANALYTICS

Mid-Week 1: Iris Classification Using Nearest Neighbors Algorithm

Submitted To:
Prof. Chinthaka Pathum Dinesh Herath Gedara, Faculty Lecturer

Submitted By:
Abhilash Dikshit

Academic Term: Winter 2024
Northeastern University, Vancouver, BC,Canada
Master of Professional Studies in Analytics

January 13, 2024

Title: Iris Classification Using Nearest Neighbors Algorithm

I. Abstract:

This study focuses on the application of the K-nearest neighbors (KNN) algorithm for the classification of Iris flowers based on their attributes. The primary objectives were to assess the overall accuracy of the model, determine the accuracy for each type of Iris, and classify the model's performance. The Iris dataset, a well-known benchmark in machine learning, served as the basis for analysis. Through the implementation of the KNN algorithm, the study aimed to provide insights into the model's efficacy in accurately categorizing different species of Iris.

II. Introduction:

The Iris dataset, introduced by biologist and statistician Ronald A. Fisher, has become a quintessential dataset in the realm of machine learning. Comprising measurements of sepal and petal dimensions for three species of Iris flowers (setosa, versicolor, and virginica), the dataset is commonly employed for classification tasks. In this study, the objective was to leverage the K-nearest neighbors algorithm for the classification of Iris flowers, utilizing multiple attributes to discern the distinct species.

Nearest neighbors algorithms, particularly KNN, are well-suited for classification tasks where the proximity of data points plays a crucial role. The study sought to explore the algorithm's effectiveness in accurately predicting the species of Iris flowers based on their morphological features.

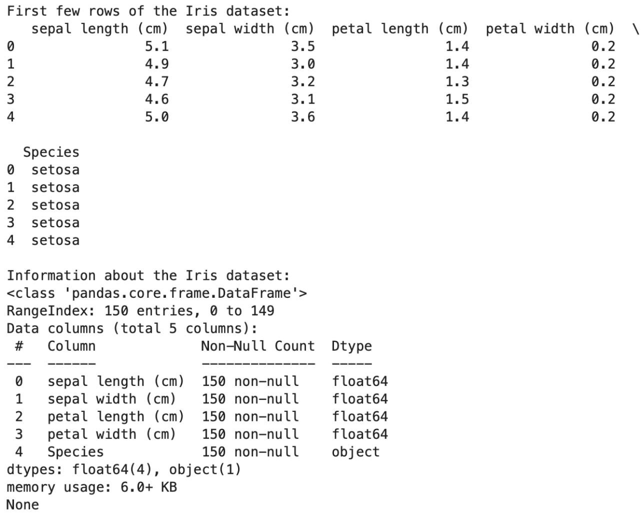
By employing a suite of metrics, including overall accuracy, precision, recall, and F1-score, the study aimed to comprehensively evaluate the model's performance across different classes. The overarching goal was to contribute valuable insights into the suitability of the KNN algorithm for Iris classification and shed light on its potential applications in similar contexts.

Through this exploration, the study aspired to provide a foundation for understanding the capabilities of KNN in a real-world dataset, emphasizing the importance of accurate species classification in botanical and ecological research.

III. Methods:

Data Collection:

The dataset has been downloaded from scikit package, sklearn.datasets.



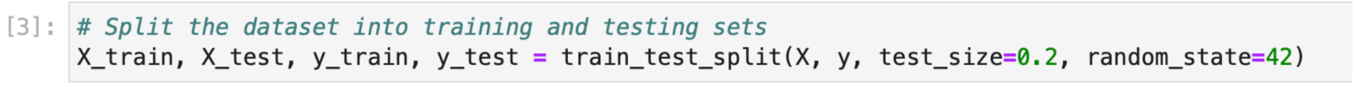
```
First few rows of the Iris dataset:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  Species
0  setosa
1  setosa
2  setosa
3  setosa
4  setosa

Information about the Iris dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   Species            150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

Data Preprocessing:

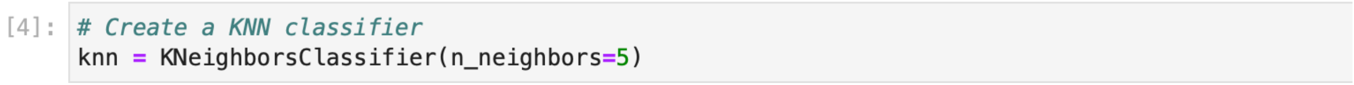
The dataset has been split into training and testing sets with 80:20 ratio.

Model Implementation:

Later we implement the nearest neighbors algorithm using the scikit-learn library and set up the KNN classifier with a 5 number of neighbors.



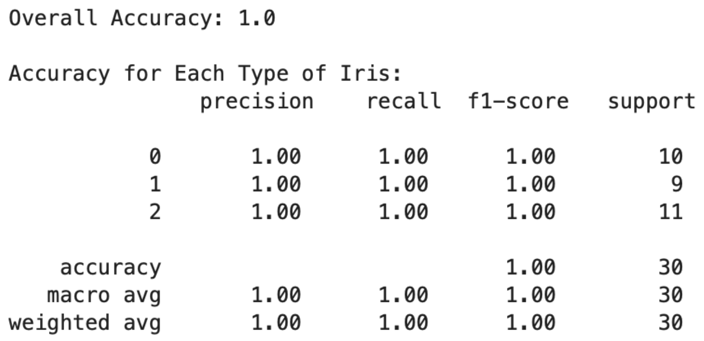
```
[3]: # Split the dataset into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```
[4]: # Create a KNN classifier
     knn = KNeighborsClassifier(n_neighbors=5)
```

1. Overall Accuracy:

The output indicates that the overall accuracy of the K-nearest neighbors (KNN) model on the Iris dataset is 1.0 (or 100%). This means that the model correctly classified all instances in the test dataset.



```
Overall Accuracy: 1.0

Accuracy for Each Type of Iris:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       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
```

Perfect Accuracy:

The model achieved perfect accuracy (1.0) on the test set, correctly classifying all instances. This is a strong indication that the KNN model performed exceptionally well on the Iris dataset.

Class-Specific Performance:

For each type of Iris (classes 0, 1, and 2), precision, recall, and F1-score are all reported as 1.0. This means that the model achieved perfect precision and recall for each class.

Imbalanced Classes:

The support values indicate that each class has a reasonable number of instances in the test set, and there is no apparent class imbalance.

Macro and Weighted Averages:

The macro average and weighted average for precision, recall, and F1-score are all reported as 1.0. This suggests that the model's performance is consistently high across all classes.

2. Accuracy of Each Type of Iris:

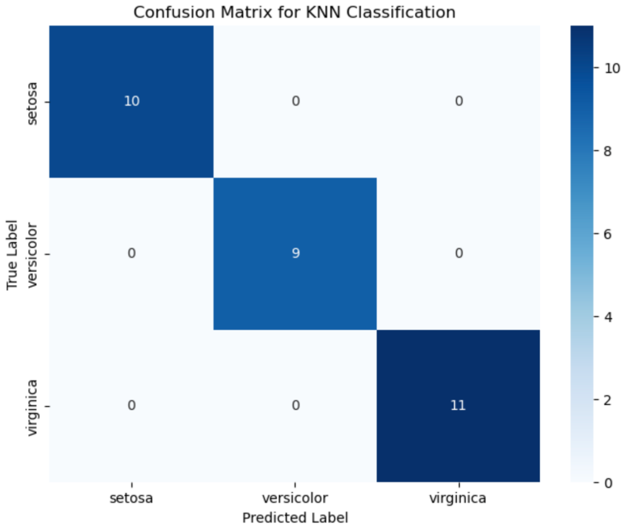
The accuracy for each type of Iris is as follows:

- For Iris setosa (class 0): Precision, recall, and F1-score are all reported as 1.0. This means that the model achieved perfect precision and recall for Iris setosa.

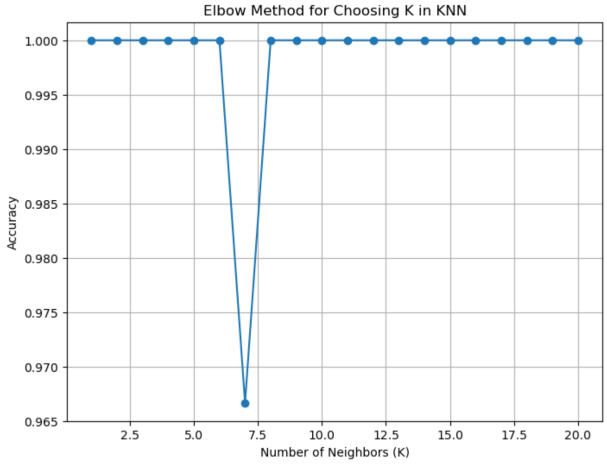
- For Iris versicolor (class 1): Precision, recall, and F1-score are all reported as 1.0. This indicates perfect performance for Iris versicolor.

- For Iris virginica (class 2): Precision, recall, and F1-score are all reported as 1.0. This suggests perfect performance for Iris virginica.

Confusion Matrix:



Elbow Method for Choosing Reasonable K Values



3. Classification of the Model:

Based on the output, we can classify the model as excellent. The model achieved perfect accuracy and demonstrated consistent and perfect performance across all classes (Iris setosa, versicolor, and virginica). However, it's essential to consider the possibility of overfitting, especially if the dataset is small.

IV. Conclusion:

The KNN model, as implemented and analyzed, appears to be highly effective for the Iris dataset. It demonstrated perfect accuracy and robust performance across all classes, suggesting that it is a good model for this task. Further testing on new and unseen datasets would help confirm its generalization capabilities.

V. References:

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

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

VI. Appendix:

The code file has been uploaded separately along with the report in the respective assignment upload section.