Heaven's Light is Our Guide

RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

Computer Science & Engineering

Course No: CSE 4204

Sessional based on CSE 4203

Name of the Experiment

Implementation of Nearest Neighbor classification algorithms with and without distorted pattern

Submitted by

Tushar Das

Roll: 1803108

Dept. of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Submitted to

Rizoan Toufiq

Assistant Professor

Dept. of Computer Science & Engineering

Rajshahi University of Engineering & Technology

1 K-nearest Neighbor Classification algorithm

- Begin by defining the value of k, which represents the number of nearest neighbors to consider.
- Next, gather and organize the data that will be used for the analysis. This data should include a set of labeled training examples and a set of unlabeled test examples.
- For each test example, calculate the distance between the test example and each training example using a distance metric, such as Euclidean distance.
- Sort the training examples by their distance to the test example, with the closest training examples at the top of the list.
- Select the k training examples that are closest to the test example.
- Determine the majority label among the k training examples and assign that label to the test example.
- Repeat steps 3-6 for each test example, then evaluate the accuracy of the model by comparing the predicted labels to the true labels.
- If necessary, adjust the value of k or other parameters to improve the accuracy of the model.
- Once the algorithm is deemed accurate, it can be used to classify new examples.

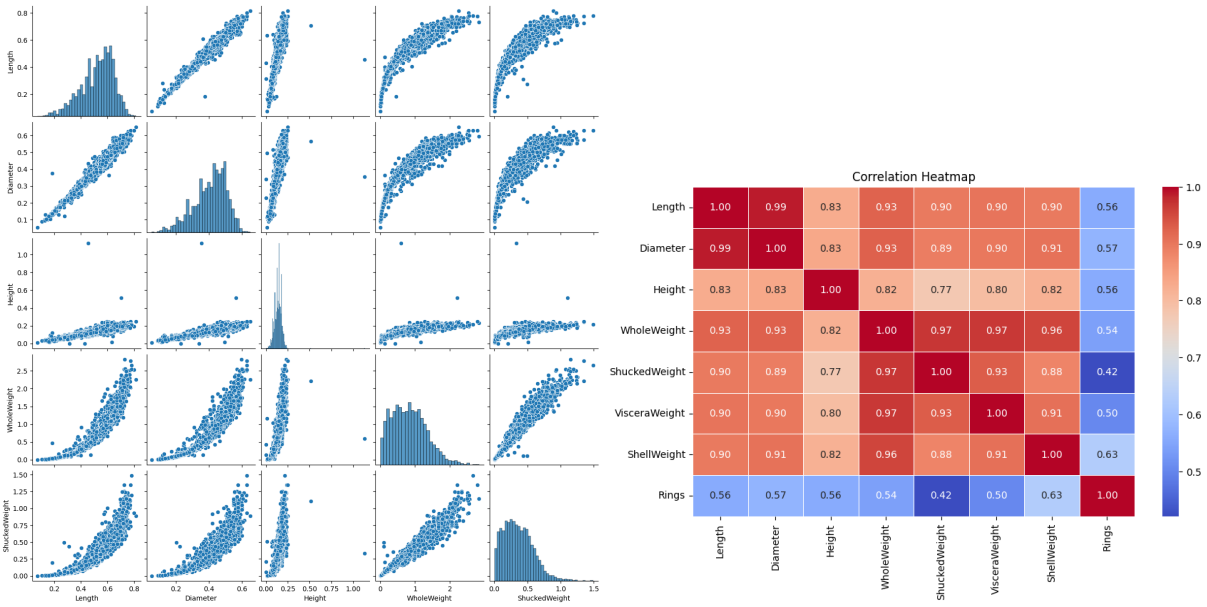
2 The Abalone Dataset

The Abalone dataset [3] is a widely used dataset in machine learning and statistics. It contains measurements and other attributes of abalone, a type of marine mollusk. The dataset has the following characteristics:

- **Features:** The dataset includes 8 physical and geometrical measurements such as Length, Diameter, Height, Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight. Additionally, there is a Sex attribute representing the gender of the abalones.
- **Target Variable:** The target variable is 'Rings,' which is a numerical value indicating the age of the abalones. This variable is crucial for age prediction.
- **Size:** The dataset consists of a total of 4,177 instances.
- **Data Types:** The dataset contains a mixture of numerical and categorical data types, making it suitable for various data preprocessing techniques.

2.1 Exploratory Data Analysis

- **Pair Plot:** A pair plot was created to visualize pairwise relationships among the continuous attributes. All the plots followed a straight line. The pair plots were not scattered. This is a major drawback for a KNN [2] classification. There were a few outliers also, but they are tiny in number and these need not to be removed.



(a) Pair Plot of Continuous Attributes

(b) Correlation Heatmap

Figure 1: Relationship Analysis

- **Correlation Heatmap:** A Heatmap was generated to visualize the correlations between continuous attributes. There were high correlations among the features. The Diameter and Length, Whole Weight and Length, Shucked Weight and Lenght are highly correlative. From these, one can be removed from the dataset.

- **Class Distribution Analysis:** The 'Rings' attribute includes a wide range of classes, with ages from 1 to 29. The class distribution reveals that some age classes have significantly more instances than others. Classes [9, 10, 8, 11, 7, 12] are the most abundant, while the other classes have fewer examples. For simplicity, the less-numbered classes were excluded and in the KNN Model, the top 4 classes were classified.

2.2 Feature Engineering

- **Target Variable Transformation & Class Aggregation** In the Abalone dataset, the target variable 'Rings' represents the age of abalones and has a wide range of values. To simplify the classification task and improve model performance, we applied feature engineering to transform 'Rings' into a more manageable target variable. The less-numbered classes were excluded and in the KNN Model, the top 4 classes were classified.

- **Dataset Transformation:** The dataset was transformed to retain only the selected classes and exclude the 'Other' category. Columns related to the original 'Sex' and 'Rings' were removed.

- **Scaling:** Feature scaling was applied to the numeric attributes to ensure that all features have the same scale. The StandardScaler was used to standardize the data, making it suitable for machine learning algorithms.

- **Label Encoding:** The label encoder was used to encode the target labels into numerical values. This step is crucial for models that require numeric inputs.

2.3 Training and Test Dataset Ratio

For model development and evaluation, it is common to split the dataset into training and test sets. The recommended ratio is typically 80/20 or 70/30. In this analysis, we have divided the dataset into a training set and a test set with an 80/20 split. This means that 80% of the data is used for training the K-Nearest Neighbors (KNN) classifier, while the remaining 20% is reserved for testing and evaluating the classifier's performance.

3 K-Nearest Neighbor (KNN) Classifier

In the machine learning pipeline, the K-Nearest Neighbor (KNN) algorithm is a versatile and intuitive classification method. It is used for both classification and regression tasks. K-NN is an instance-based learning method that classifies a data point based on the majority class of its nearest neighbors in the feature space.

3.1 Custom K-NN Class Implementation

In this analysis, a custom K-NN class was developed to perform classification tasks on the Abalone dataset. The custom KNN class includes the following key functionalities:

- **Fit Function:** The 'fit' function is responsible for training the KNN classifier. It takes the training data (X_train) and the corresponding target labels (y_train) as input.
- **Predict Function:** The 'predict' function predicts the class label for a given test data point. To do this, it calculates the Euclidean distance between the test data point and all data points in the training set.
- **Euclidean Distance Calculation:** In the 'predict' function, the key calculation is the Euclidean distance between the test data point and each training data point. The Euclidean distance is used as a similarity metric, and it is computed as the square root of the sum of squared differences between feature values.
- **Classification:** After calculating the Euclidean distances for all training data points, the 'k' nearest neighbors with the smallest distances are selected. The majority class among the 'k' nearest neighbors is considered the predicted class label for the test data point. In the case of a classification task, this class label is returned as the prediction.

3.2 Choosing the Optimal 'k' Value

- In my analysis, a 5-fold cross-validation was used, meaning the dataset was divided into five parts.
- The KNN classifier was trained and evaluated multiple times, with different 'k' values, using this cross-validation approach.
- For each 'k' value tested, the accuracy scores were computed for each fold of the cross-validation.
- The cross-validation scores provide a measure of the model's performance on different subsets of the data, helping to identify the 'k' value that yields the best generalization.

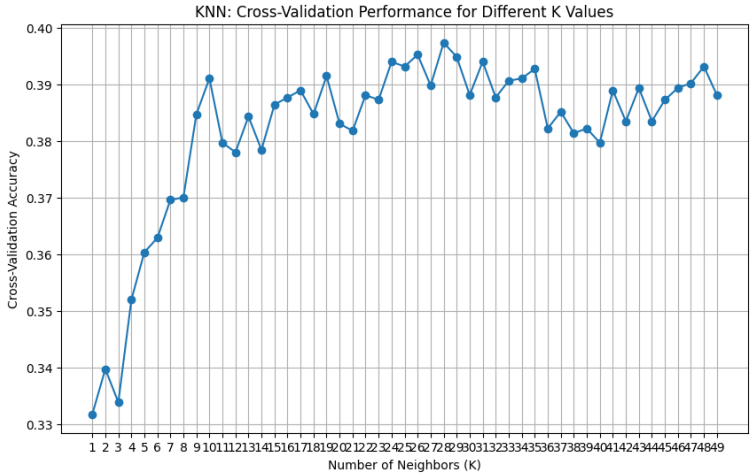


Figure 2: Choosing Optimal K Value

- The optimal ‘k’ value was determined by selecting the ‘k’ that resulted in the highest average cross-validation accuracy score across all folds.

3.3 Model Training and Evaluation

- The custom KNN class was instantiated and trained using the training data, and predictions were made for the test data.
- The accuracy of the KNN classifier was evaluated by comparing the predicted class labels to the actual class labels.
- The Accuracy was 36.34%

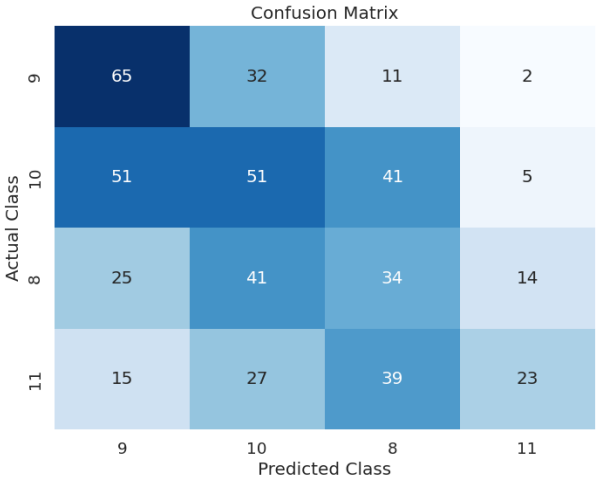


Figure 3: Confusion Matrix

4 Classifier Limitations

- One of the primary limitations of the KNN classifier, as applied to the Abalone dataset, is the observed low accuracy. The Dataset is not suitable for KNN Classifier. There may have other reasons also for the low accuracy.
- The Abalone dataset contains multiple features, and K-NN can struggle when working with high-dimensional feature spaces. High dimensionality often leads to the curse of dimensionality, where data points become sparse, making it challenging to find meaningful neighbors.
- K-NN is sensitive to noisy data points and outliers. If the dataset contains noisy or outlier data, it can significantly affect classification accuracy. In the above experiment, outliers were not removed. This can be a reason for low accuracy.
- The dataset exhibits class imbalance, with certain classes having a significantly larger number of instances than others. KNN may struggle to correctly classify the minority classes.

References

[1] Create a K-Nearest Neighbors Algorithm from Scratch in Python
Published in Towards Data Science
”https://rb.gy/6xto4”

[2] Source Code ”https://rb.gy/jwk4r”

[3] Abalone Dataset ”https://rb.gy/2ov6cr”

[4] Neural Computing: An Introduction - R Beale and T Jackson