K-Nearest Neighbors (K-NN) Algorithm: Implementation and Analysis

Table of Contents

Implementation

K-Nearest Neighbors (K-NN) is a simple method for classification. It works by comparing new data points with existing ones and classifying them based on similarity. The main steps include:

- **Distance Calculation:** The algorithm measures how close data points are using Euclidean or Manhattan distance.
- **Fit Method:** This step stores the training data for later use.
- **Predict Method:** It finds the `k`-nearest points to the new data and assigns the most common class.
- **Score Method:** This calculates how well the model performs by checking its accuracy.

Data Preparation and Preprocessing

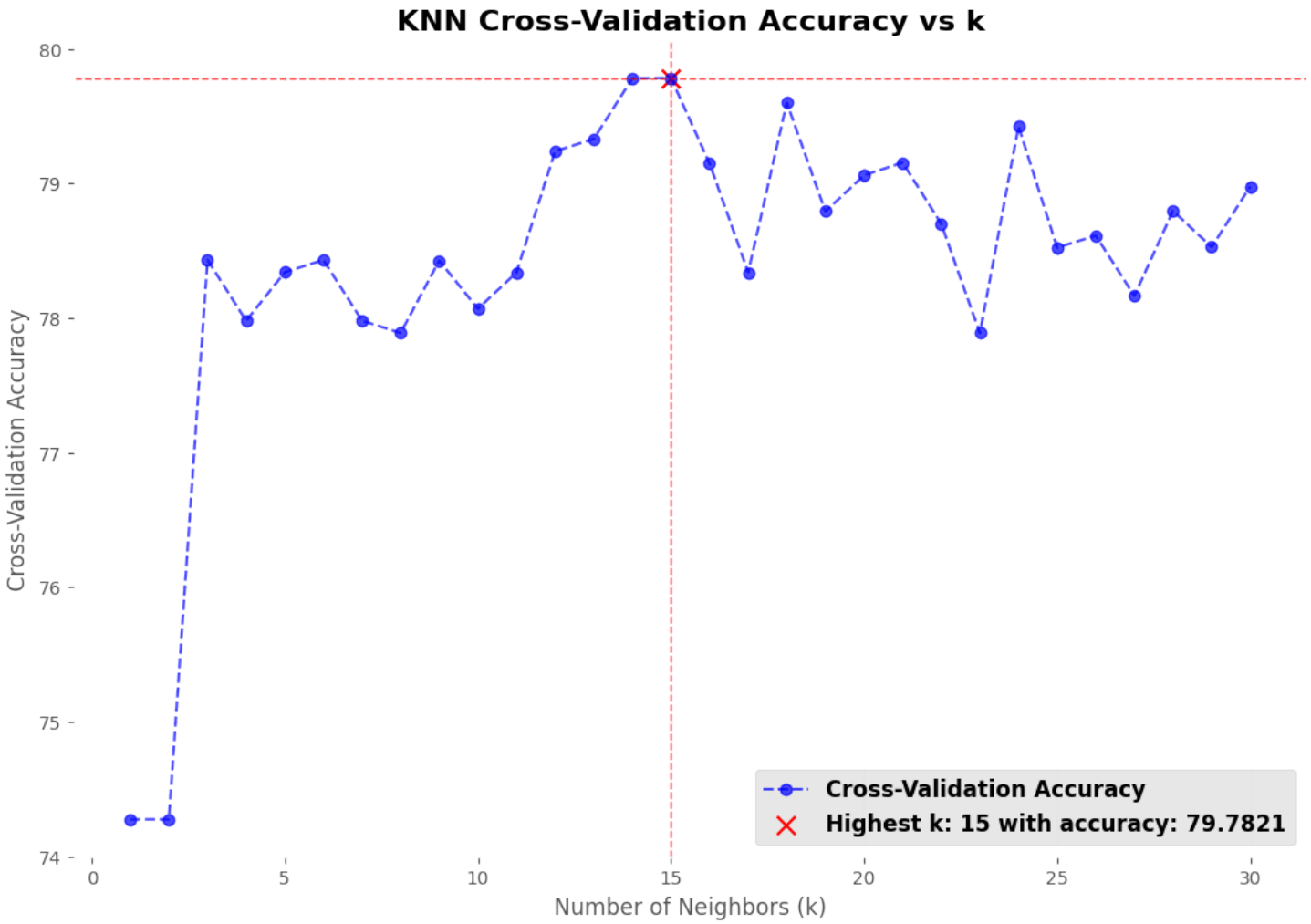
Before using K-NN, the data needs to be prepared. The dataset includes:

- **Training Inputs:** `X_train` (features of training samples)
- **Training Labels:** `y_train` (correct class for training samples)
- **Test Inputs:** `X_test` (features of test samples)
- **Test Labels:** `y_test` (correct class for test samples)

To make sure all features contribute equally, the `StandardScaler` from Scikit-learn is used to scale them.

Performance Analysis

The results below were obtained by testing the K-Nearest Neighbors (KNN) algorithm on a modified version of the MNIST dataset. This dataset includes only images of the digits 5 and 6. The test aimed to evaluate how well KNN can classify these two digits



The accuracy of K-NN depends on the choice of `k`:

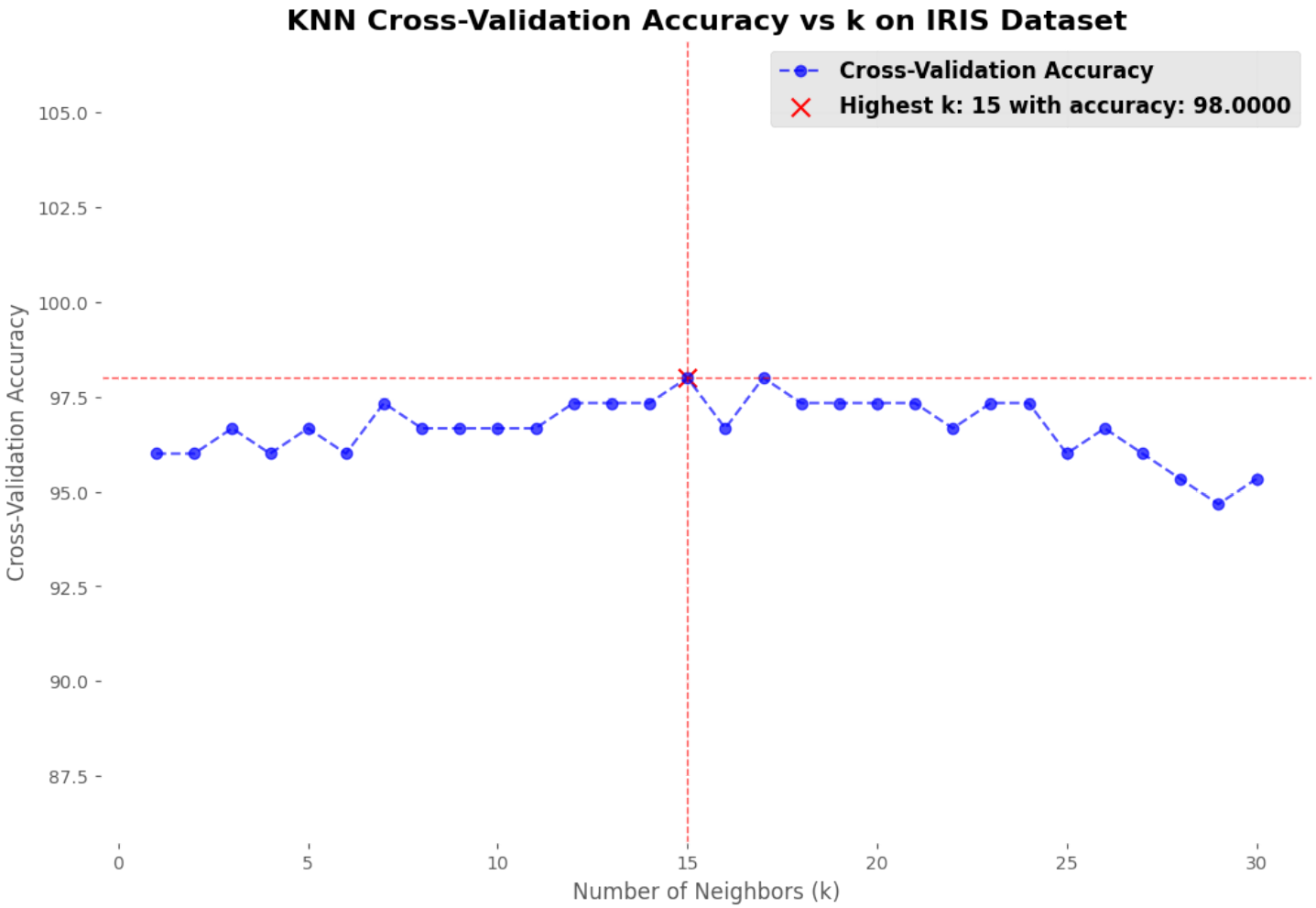
- **Small k Values:** The model performs poorly at `k = 1` and `k = 2` with **74.25%** accuracy. Small `k` makes the model sensitive to noise.
- **Moderate k Values:** When `k` is `3` or `4`, accuracy improves to **78.5%**, as the effect of noise reduces.
- **Large k Values:** The accuracy peaks at `k = 15` with **79.51%**. However, increasing `k` too much causes loss of detail and higher computation time.

Additional Tests

To further evaluate the performance of K-NN, additional tests were conducted using various datasets from Scikit-learn:

- **Iris Dataset:** A well-known dataset for classification.
- **Digits Dataset:** Handwritten digit recognition.
- **Wine Dataset:** Classification of different wine types.
- **Breast Cancer Dataset:** Medical diagnosis of breast cancer.
- **Diabetes Dataset:** Predicting diabetes progression.
- **California Housing Dataset:** Regression task for housing prices.

Iris Dataset



Diabetes Dataset

Conclusion

The K-NN algorithm is effective for classification. Choosing the right `k` is key to its performance.