Ben Zamboldi
William & Mary
B.S., CS & CAMS ‘22
3 April 2022

KNN README

The Python packages used for this part of the problem set include Pandas and Math. The Pandas package was used to store the data in a Pandas dataframe for data wrangling. The Math package was used for the square root method in the Euclidean distance calculation.

To implement the KNN I created a class KNN. The init method initializes global variables to hold the training data, testing data, and value for K.

The method Euclidean is used to calculate the Euclidean distance between two points was implemented. The function takes in two lists of points of the same dimension. The function loops through each dimension of the points and adds the squared difference to the total sum and finally returns the square root of the sum. The Math package was used to calculate the square root. Because the dimensions of the points are constant, this method runs in constant O(1) time.

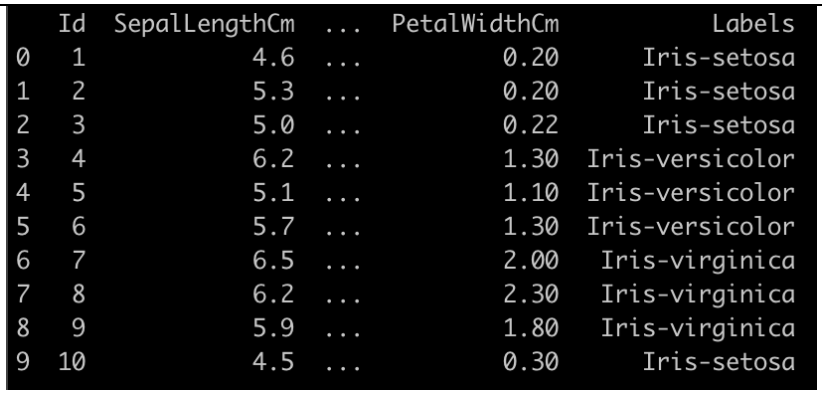
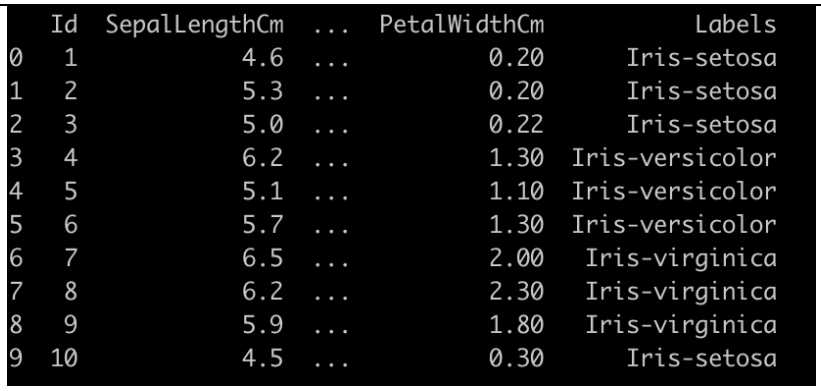
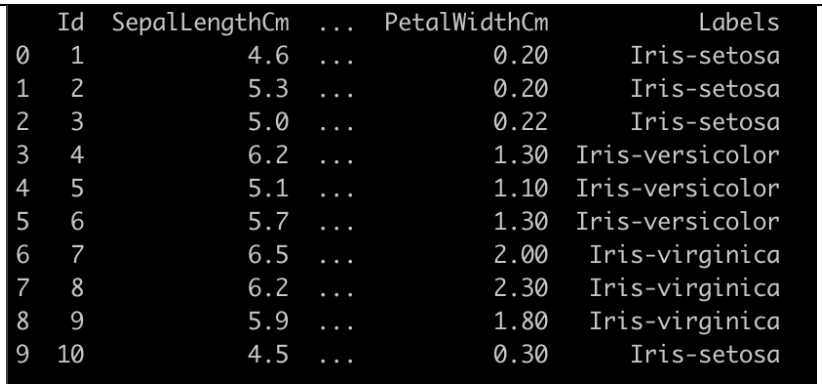
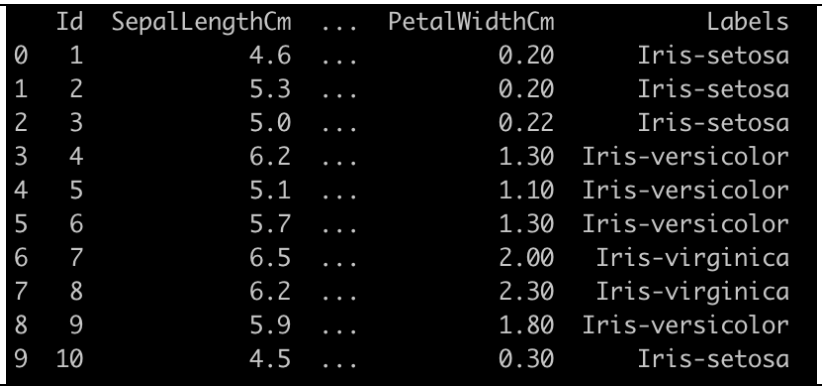
The method calculate_neighbors is used to calculate the K nearest neighbors for a given point. The method takes on parameter, a vector representing the coordinates of the point, and calculates the Euclidean distance between that point and all the points in the training set. The distances to all points in the training set is stored in a list and sorted. Then the K closest points are saved to a list representing the K nearest neighbors of that point. Because this method iterates across all points in the training dataset, the runtime is linear O(n) time where n is the number of items in the training dataset.

The classify method is used to classify a point to a label. The method takes on parameter, the point of interest stored as a vector of coordinates. The method then calls the calculate_neighbors method to get the K nearest neighbors for that point. Based on majority voting, the most occurring label in the K nearest neighbors is assigned as the label for the point of interest. This method is of linear runtime O(n) where n is the number of items in the training set, but only because of the calculate_neighbors method call.

The run method implements the KNN overall algorithm. For each point in the testing dataset, the classify method is called to classify the point based off of the training dataset. When the method is finished, a list containing the predicted label for each point in the testing set is returned. This algorithm takes linear time O(n) to run.

With the implementation of the KNN class, I created four instances of the class for K=20, K=30, K=40, and K=50 to test my validation results. For each value of K, the instance was trained on the training set and used to predict the labels in the validation and testing sets. The differing values of K did not affect the predictions produced by my algorithm in testing, although K=50 produced 1/10 results different than K=20, K=30, and K=40 for the testing set. The results of the predictions on the testing set can be seen below.

K=20

K	Prediction Result
20	
30	
40	
50	

	Id	SepalLengthCm	...	PetalWidthCm	Labels
0	1	4.6	...	0.20	Iris-setosa
1	2	5.3	...	0.20	Iris-setosa
2	3	5.0	...	0.22	Iris-setosa
3	4	6.2	...	1.30	Iris-versicolor
4	5	5.1	...	1.10	Iris-versicolor
5	6	5.7	...	1.30	Iris-versicolor
6	7	6.5	...	2.00	Iris-virginica
7	8	6.2	...	2.30	Iris-virginica
8	9	5.9	...	1.80	Iris-virginica
9	10	4.5	...	0.30	Iris-setosa

	Id	SepalLengthCm	...	PetalWidthCm	Labels
0	1	4.6	...	0.20	Iris-setosa
1	2	5.3	...	0.20	Iris-setosa
2	3	5.0	...	0.22	Iris-setosa
3	4	6.2	...	1.30	Iris-versicolor
4	5	5.1	...	1.10	Iris-versicolor
5	6	5.7	...	1.30	Iris-versicolor
6	7	6.5	...	2.00	Iris-virginica
7	8	6.2	...	2.30	Iris-virginica
8	9	5.9	...	1.80	Iris-virginica
9	10	4.5	...	0.30	Iris-setosa

	Id	SepalLengthCm	...	PetalWidthCm	Labels
0	1	4.6	...	0.20	Iris-setosa
1	2	5.3	...	0.20	Iris-setosa
2	3	5.0	...	0.22	Iris-setosa
3	4	6.2	...	1.30	Iris-versicolor
4	5	5.1	...	1.10	Iris-versicolor
5	6	5.7	...	1.30	Iris-versicolor
6	7	6.5	...	2.00	Iris-virginica
7	8	6.2	...	2.30	Iris-virginica
8	9	5.9	...	1.80	Iris-versicolor
9	10	4.5	...	0.30	Iris-setosa