**KNN Classification:** Determining which class a data point belongs to based on euclidean distance

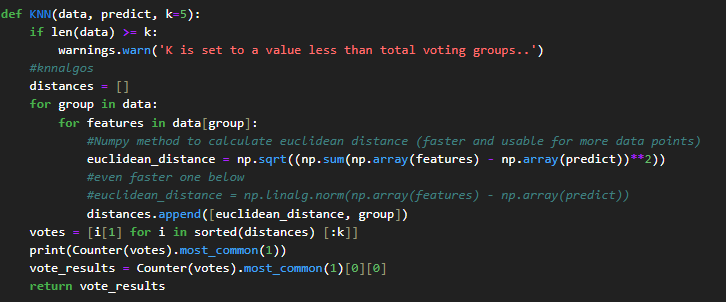
**euclidean distance** = sqrt((x1-x2)^2+(y1-y2)^2)

**confidence:** a value that tells us how accurately a data point has been classified.

**Modules used: ‘neigbors’, ‘preprocessing’, ‘model\_selection’, ‘neighbors.KNeighborsClassifier()’**

If dataset is not initially encoded to numeric and has values like ‘high’, ‘low’, ‘true’, ‘false’, ‘v. high’, we have to use the module ‘label\_encoder’ from sklearn.preprocessing to encode it to numeric before applying KNN model.

**KNN Creation Function**



**Plotting KNN Graph**

A computer screen shot of text

Description automatically generated

A graph with red green and black dots

Description automatically generatedThe point of KNN is to determine whether the black point on the graph belongs to the red class or to the green class. This is done using KNN classification which uses Euclidean distance and compares all the distances and sorts in ascending order to give the shorted distances first. Then votes are counted.  
Voting Classes = Total Number of Classes

K = an (odd) value that is greater than number of voting classes . The majority in K value is used to predict the class of a data point.