**KNN Classifier Summary Report**

**On**

**Artificially Generated Make\_Blogs Dataset**

**Professor Submitted by**

**CJ Wu Pavan Chaitanya Bommadevara**

**Summary:**

KNN analysis to the artificial make\_blobs dataset and reporting the accuracy.

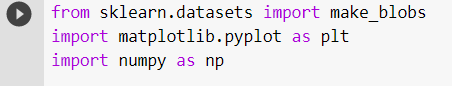
Procedure:

* There are 4 steps in solving the given problem.
* Step 1: Importing the Libraries.
* Step 2: Spliting the data.
* Step 3: KNN Analysis on Split data.
* Step 4: Accuracy Scores

**Importing the Libraries**

* My approach towards the problem is initially learned the importance of numpy ,matplotlib libraries.

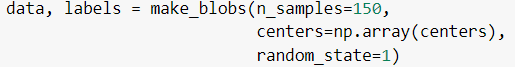
These are the required imports



We need to use the 3 classes where are a part of make\_blobs package.



Creating the data variables along with the data labels is shown below

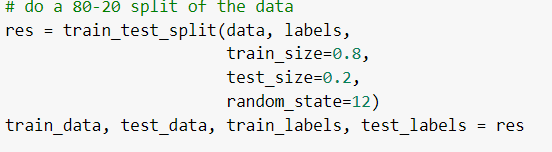


Here we are making the 150 samples with the random state of 1 with the centers as length of the centers.

**Spliting the data**

* Dividing the Artificial blobs dataset into 2 parts
  + 80 % Training Data
  + 20 % Test Data

So that we can see how the model is going to predict the output. And then assigning the splited data into different variables like train\_data,test\_data,tain\_labels,test\_labels.

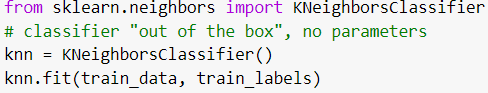


**KNN Analysis on Split data**

The k-nearest neighbors’ algorithm, often known as KNN, is a non-parametric, supervised learning classifier that utilizes proximity to create classifications or predictions about an individual data point's grouping.

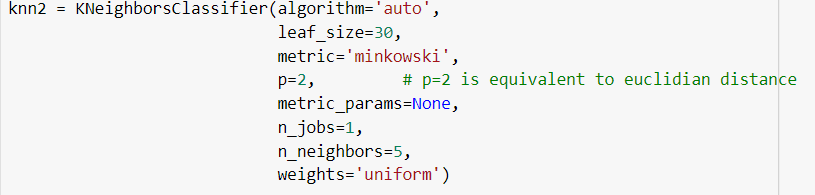
We now us the KNN classifier with different parameters and then we can analyses the accuracy scores that is generated with them.

**Knn classifier with no parameters:**

****

In this we doesn’t use any parameters like algorithm,leafsize,weights,metric etc.

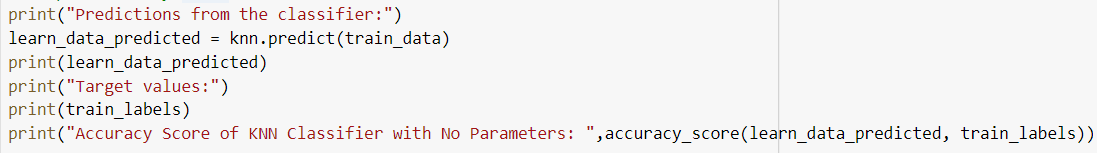
**Knn classifier with Specific parameters:**

****

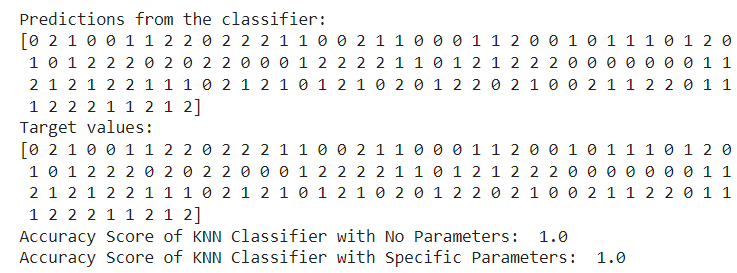
In this we use any parameters like algorithm,leafsize,weights,metric etc.

**Accuracy Scores**

We now identify the accuracy scores of different KNN classifier that we have used till now.

****

The Accuracy values that is generated for the above KNN classifier code are:

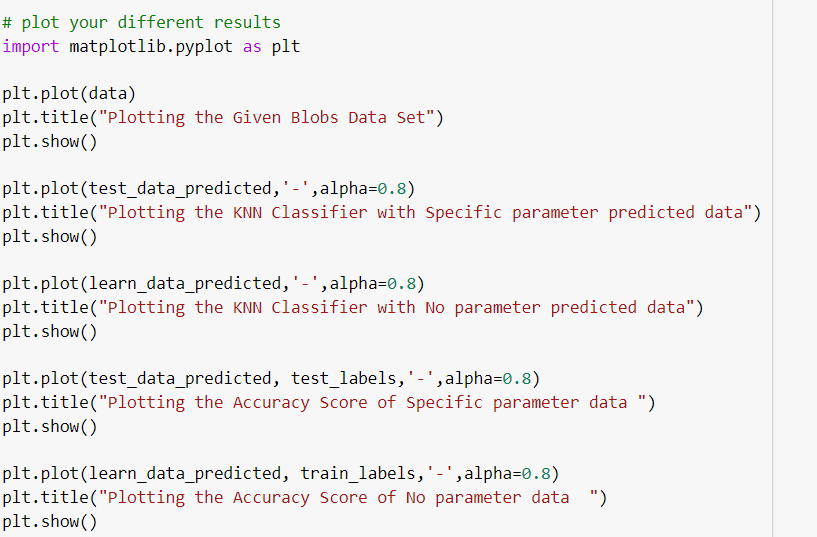
****

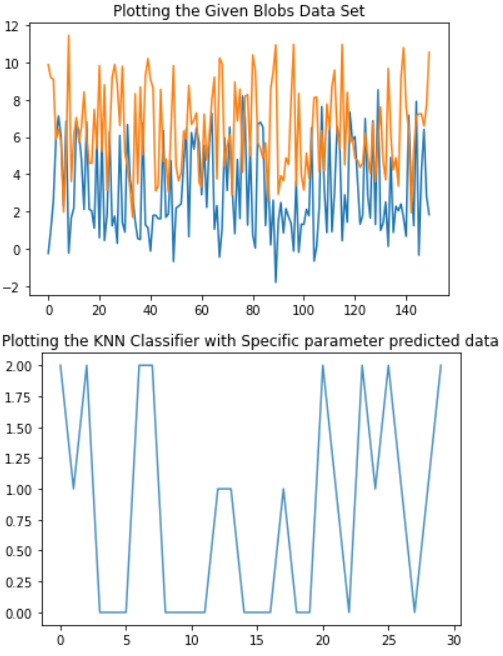
We have seen that the accuracy score with the KNN classifier is 1.0 which is quite good.

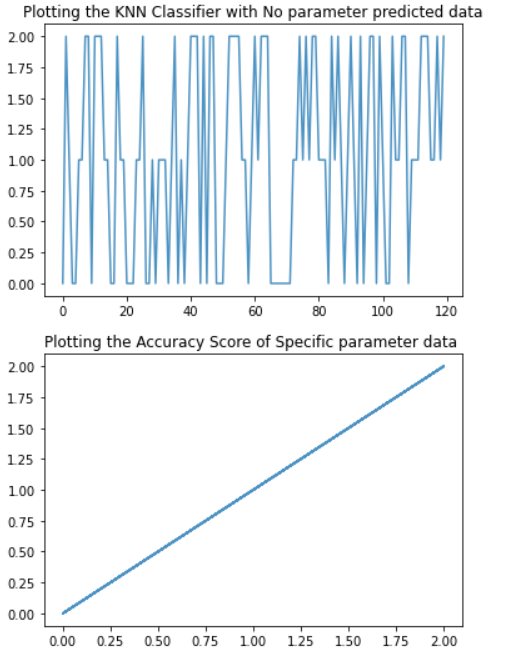
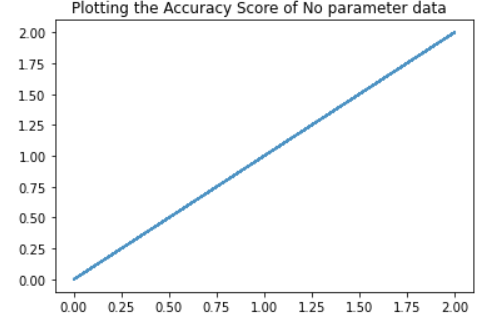
**Summary of results:**

|  |  |
| --- | --- |
| Knn Classifier | Accuracy Score |
| knn = KNeighborsClassifier() | 1.0 |
| Knn2 = KNeighborsClassifier(algorithm='auto',                              leaf\_size=30,                              metric='minkowski',                              p=2,                              metric\_params=None,                              n\_jobs=1,                              n\_neighbors=5,                              weights='uniform') | 1.0 |

**Plotting the Graphs**

We now see the graphs that are generated from written KNN classifier code are:****

Like this we can plot many data variables and the graphs looks like the below

**** ****

**References:**

[**https://www.tutorialspoint.com/google\_colab/google\_colab\_graphical\_outputs.htm**](https://www.tutorialspoint.com/google_colab/google_colab_graphical_outputs.htm)

[**https://colab.research.google.com/notebooks/charts.ipynb**](https://colab.research.google.com/notebooks/charts.ipynb)

<https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm>