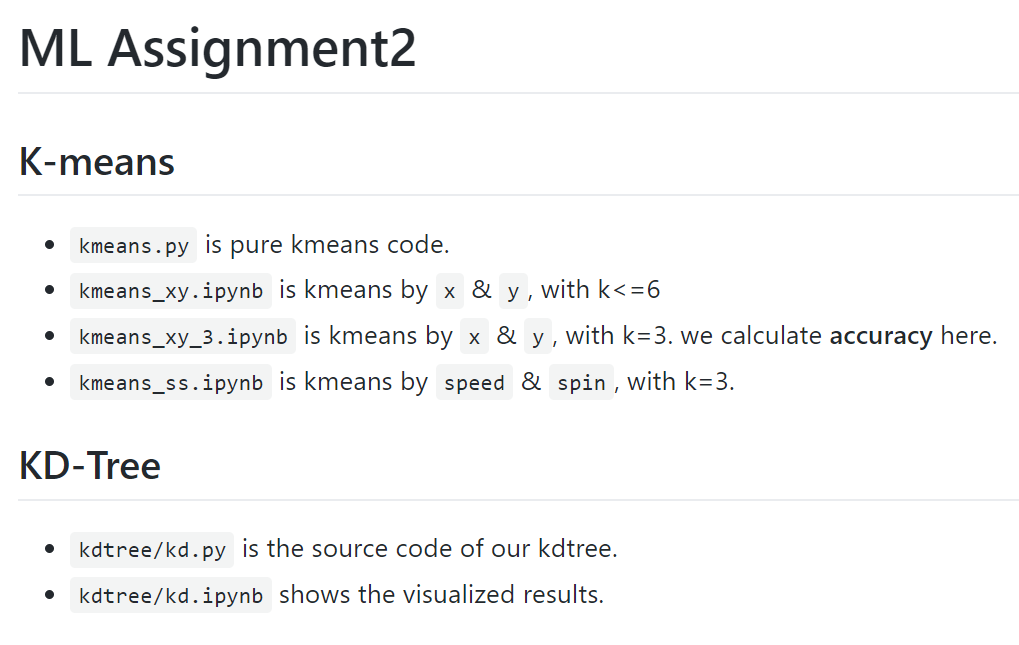
**Machine Learning Assignment2 (group24)**source: <https://github.com/WarClans612/machine_learning/tree/master/hw2>

* Working environment

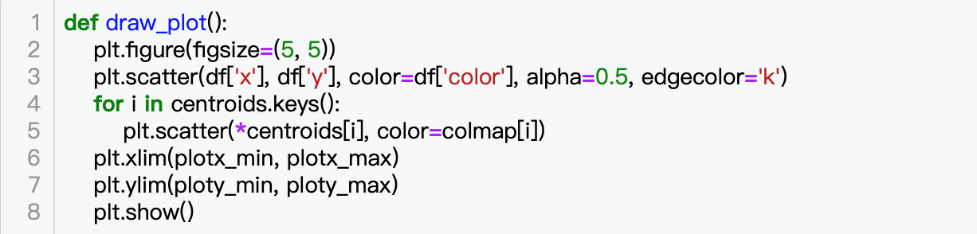
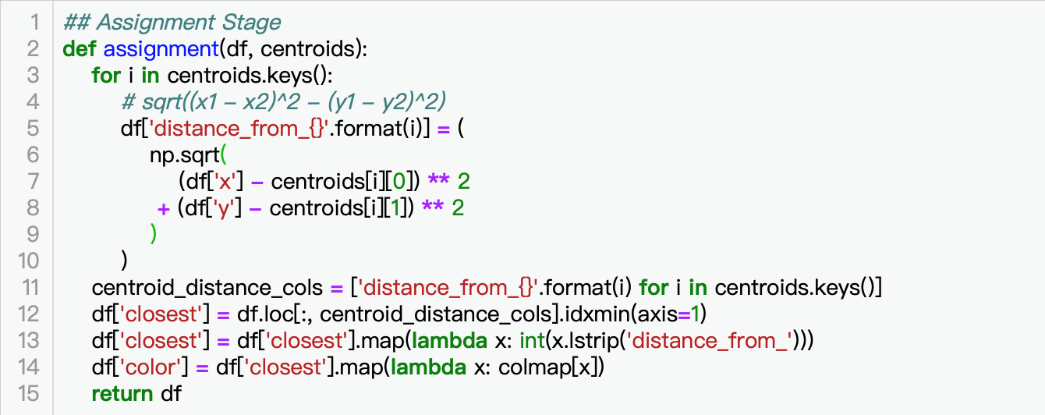
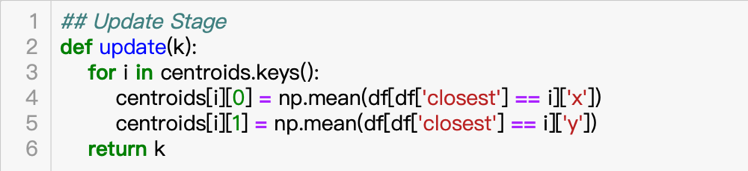
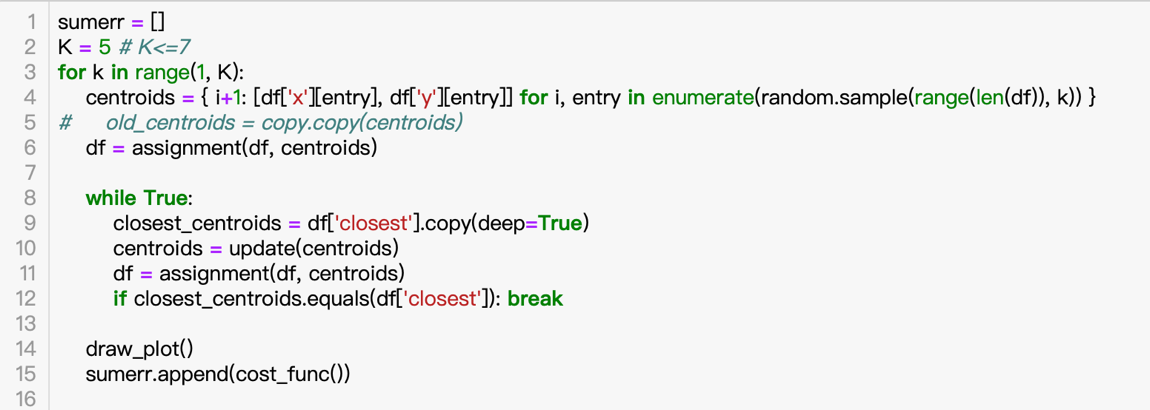
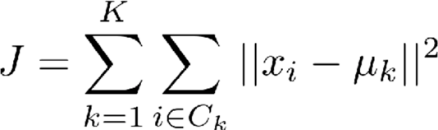
|  |  |  |  |
| --- | --- | --- | --- |
| 張翔中 | 劉昱劭 | 彭敬樺 | 周才錢 |
| macOS | macOS/Ubuntu 16.04 | Ubuntu 16.04 | Windows |

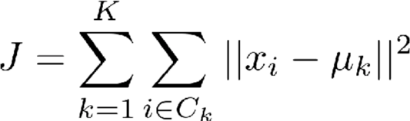
|  |  |
| --- | --- |
| *IDE* | Jupyter Notebook |

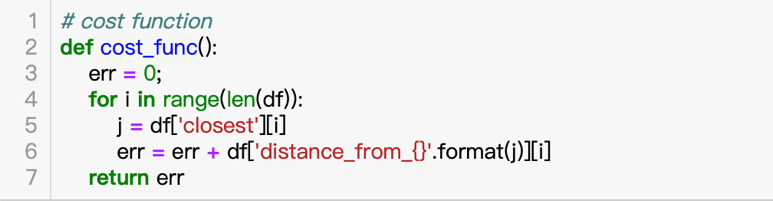
* [Our files](https://github.com/AilurusUmbra/machine_learning/tree/readme/hw2) (click to see readme.md)

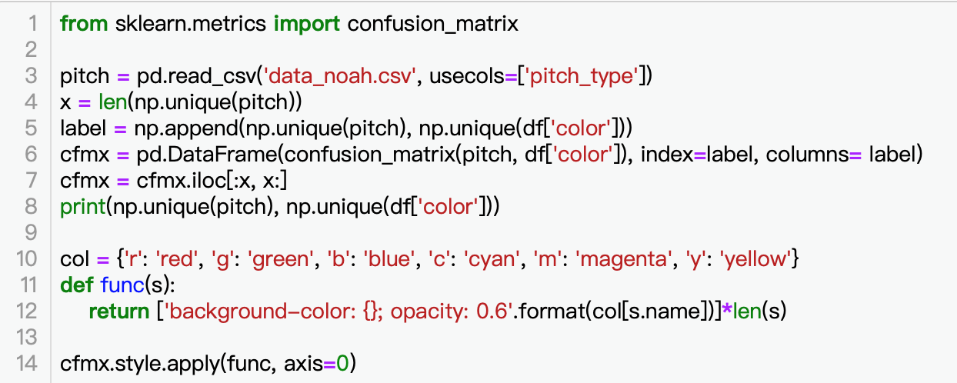
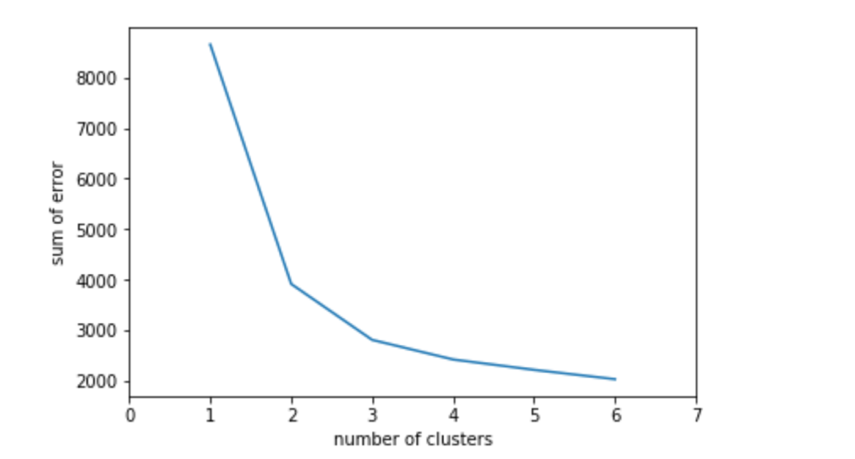
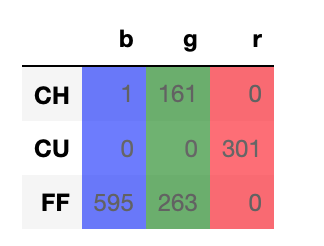


1. [**K-means**](https://github.com/WarClans612/machine_learning/blob/master/hw2/kmeans_xy.ipynb) **(click to see .ipynb file)**

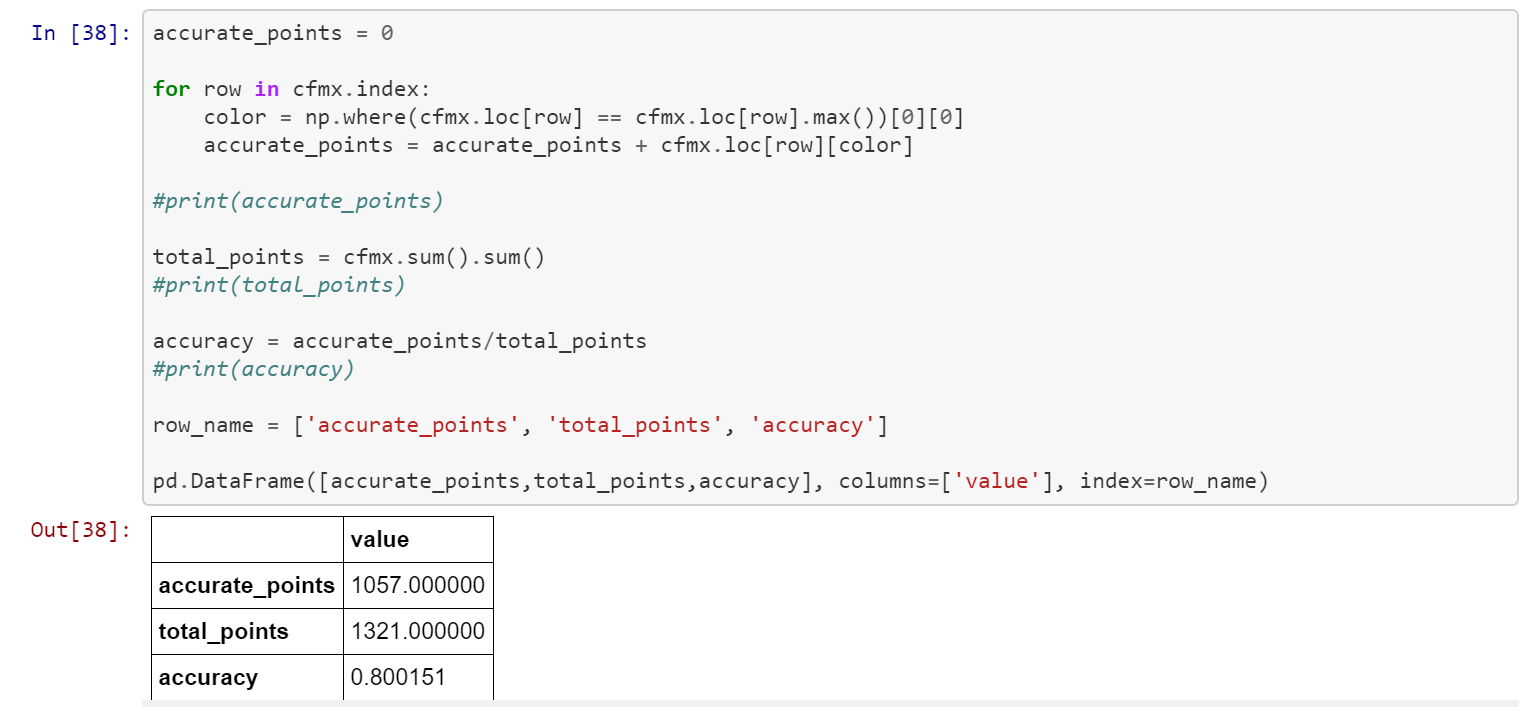
* Code (split in few partitions)
  + Plot the result of k means clustering
  + Assignment each data point to a cluster
  + Update the center of each cluster
  + Main loop (randomly pick centroids at first)
  + Cost function (sum of error)

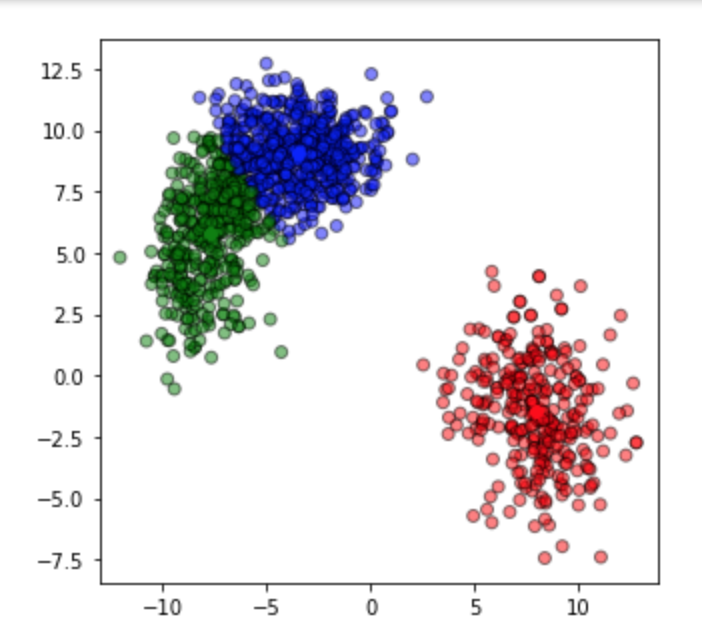
We use instead of

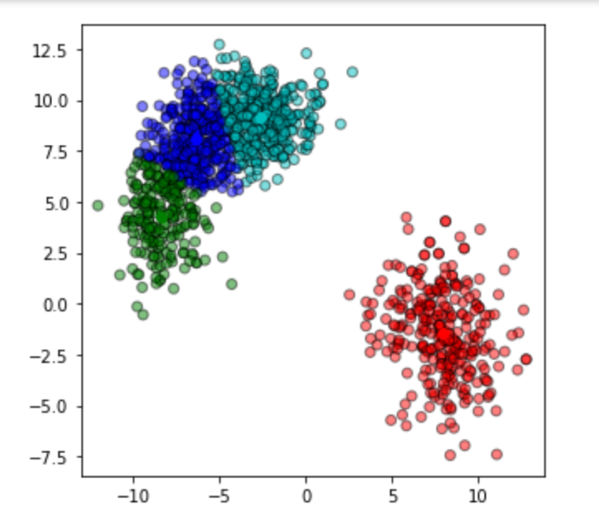
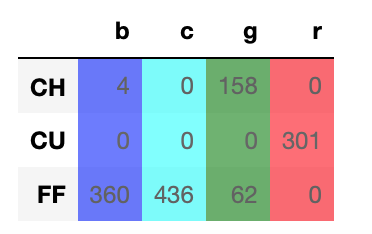
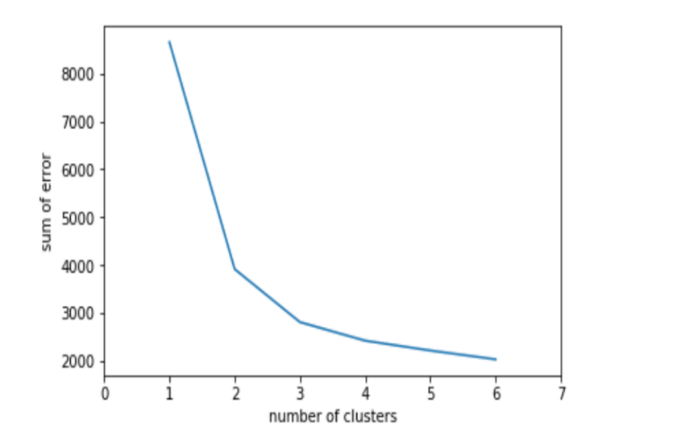


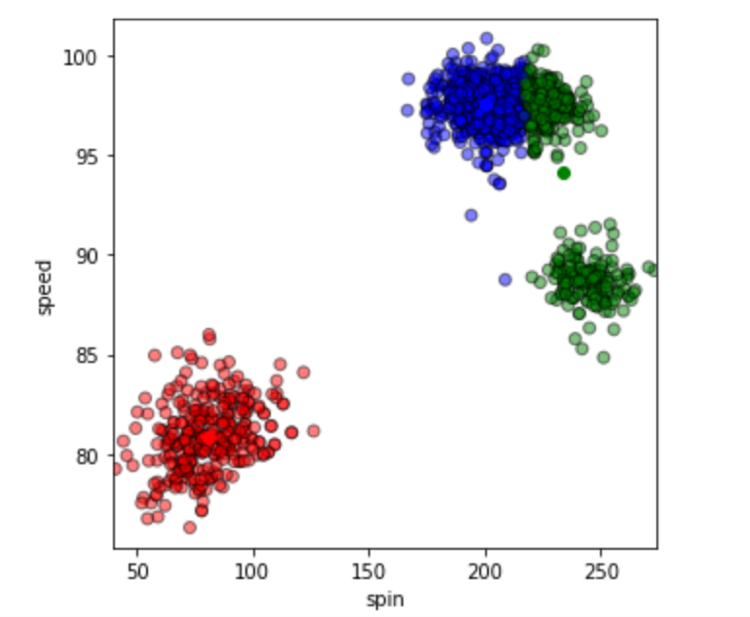
* + Confusion matrix (accuracy)
* Cost function & accuracy
  + Sum of error among different numbers of K means clustering
  + Confusion matrix (k = 3)
  + accuracy   
    source: [click here](https://github.com/WarClans612/machine_learning/blob/master/hw2/kmeans_xy_3.ipynb)   
    we count

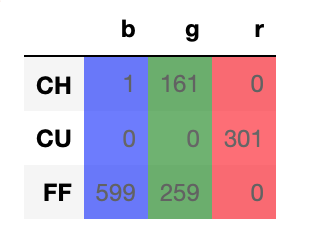
accuracy of k=3 is about 0.8.



* The result of K-Means clustering
* Is k=3 the best clustering?

If we use the elbow method, we can say that k=3 is the best k, since adding more clusters didn’t get significant decrease of sum of error. However, there could be room for discussion that k=3 may not be the best k. As we can see in the confusion matrix at k=3, nearly one third of four-seam fastballs are mixed with changeups. From the picture and confusion matrix of k=4 below, it is quite obvious that the four-seam fastballs are well split from the changeups, we can therefore combine blue and cyan clusters to make it k=3. It is k=3 at final but through an indirect way, which is not the case.

* Use another two or more attributes to partition
  + source: [click here](https://github.com/WarClans612/machine_learning/blob/master/hw2/kmeans_xy_3.ipynb)
  + Ex. Use speed and spin attributes
  + This result looks not good enough.



**II.** [**KD-Tree**](https://github.com/WarClans612/machine_learning/blob/master/hw2/kdtree/kd.ipynb) **(click to see .ipynb file)**

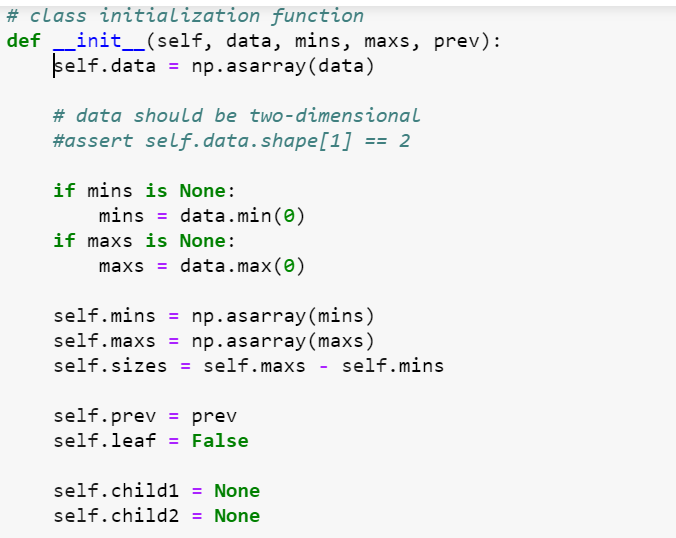
* KD-Tree code  
  We modifed the sample code from astroML, but we didn't use astroML's library

source: <http://www.astroml.org/book_figures/chapter2/fig_kdtree_example.html>

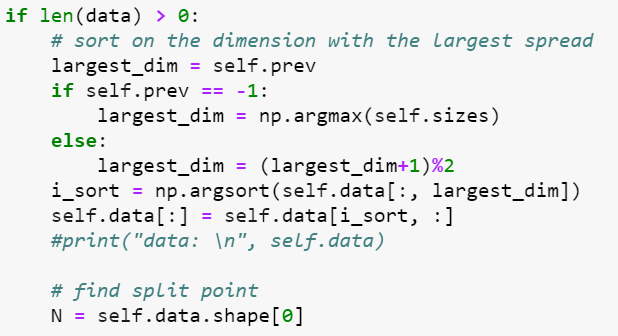
* + Build a KDTree class

C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\kdtree.png

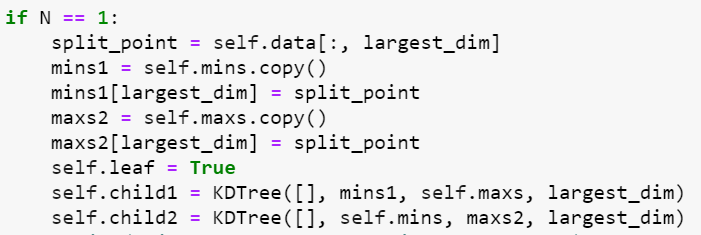
* + init function & variable initialization



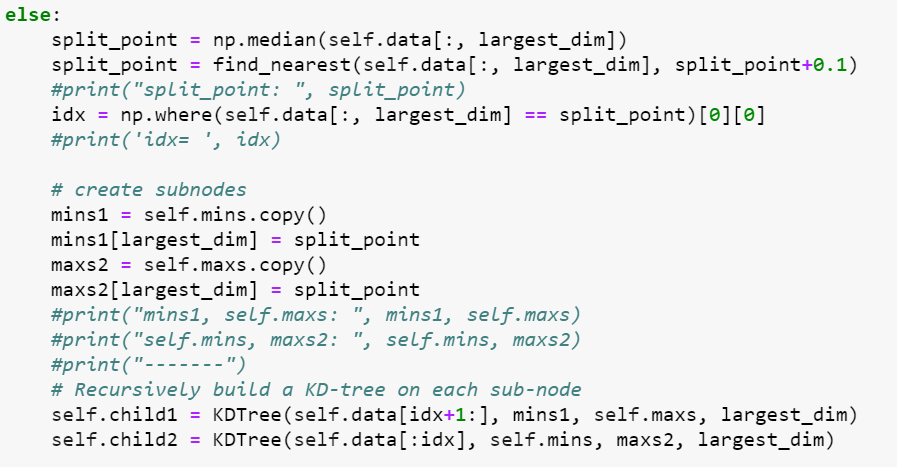
* + Find the more spread dimension

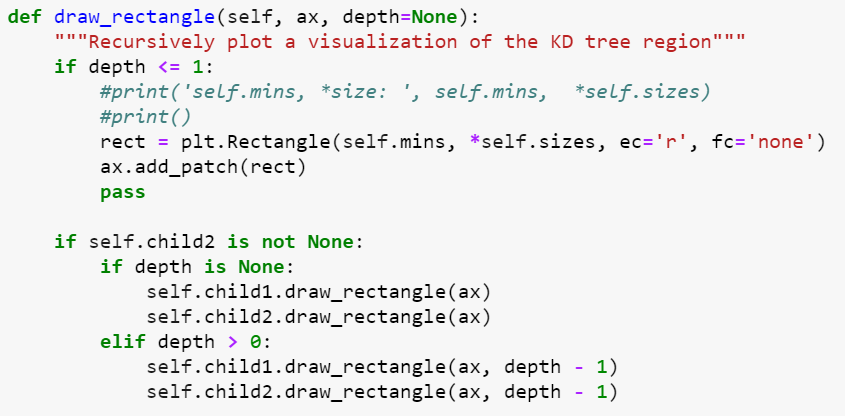
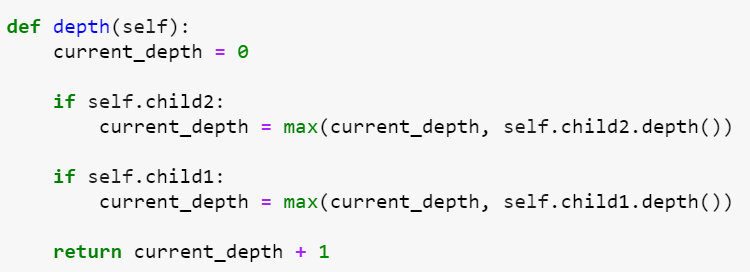


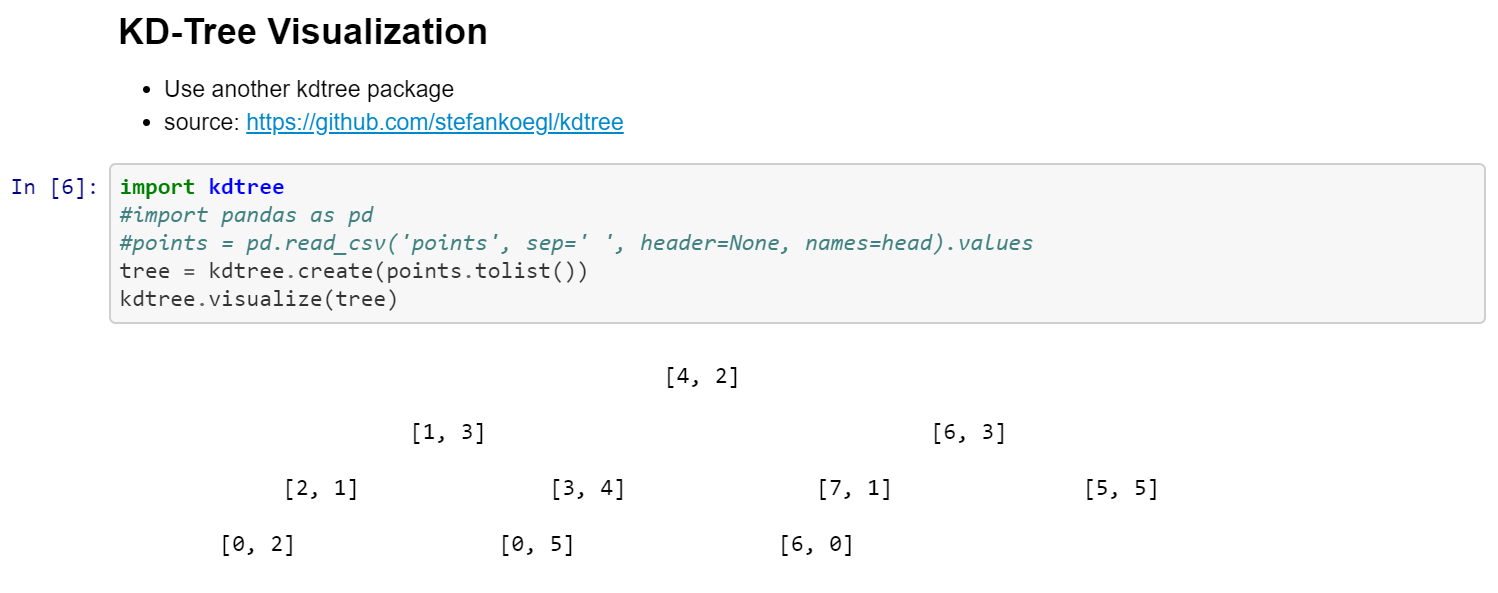
* + For leaf node



* + For non-leaf node, recursively create sub-trees.



* + draw\_rectangle is used to plot the divided region of KD-Tree.  
    
  + depth is used to get the depth of KD-Tree.  
    
* Result of KD-Tree
  + Visualize the KD-Tree   
    Use kdtree package to visualize the order of KD-Tree



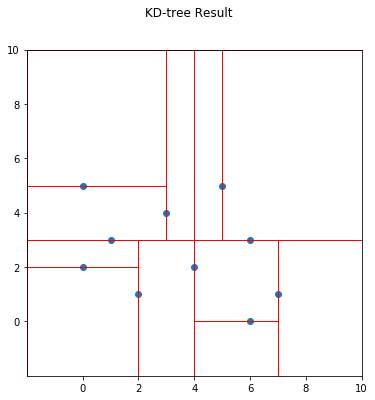
level 1 n

level 2 n

level 3 n

level 4 n

* + Draw the 2-dim divided square of kd-tree  
    Use our draw\_rectangle function to draw the divided rectangle.



* + Draw the 2-dim divided square for each level of kd-tree   
    We can see the each step of draw\_rectangle.

