**ANNOY**

**(Approximate Nearest Neighbors Oh Yeah)**

The algorithm which we will use in finding nearest data point. Like you want to search similar product of any given product. There are several methods of finding nearest neighbors such as kNN, kd-tree, Locality Sensitive Hashing (LSH) but ANNOY will perform extra-ordinarily.

1. **Locality Sensitivity Hashing**

Steps

1. In this algorithm, all data points along with testing data point projected on space.
2. After projection, random hyperplanes will be formed.
3. With respect to each hyperplane, calculate the classes of each data point.
4. Form key value pair dictionary for same labels (Here labels are calculated in step c).
5. Now take majority voting among the classes and then decide the class for that.
6. Display those nearest neighbors as a Result.

Advantages -

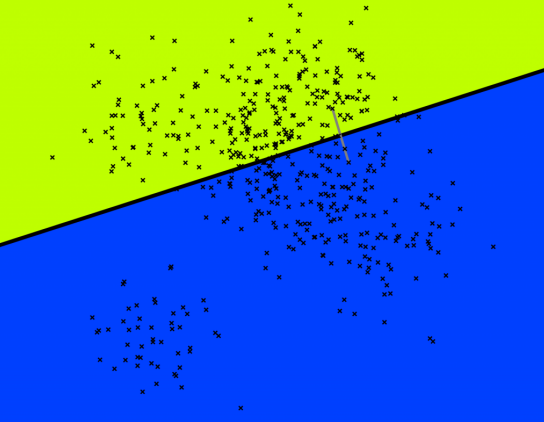
1. In Nearest Neighbors algorithm compare kNN, kdtree and LSH easy to implement.

Disadvantages -

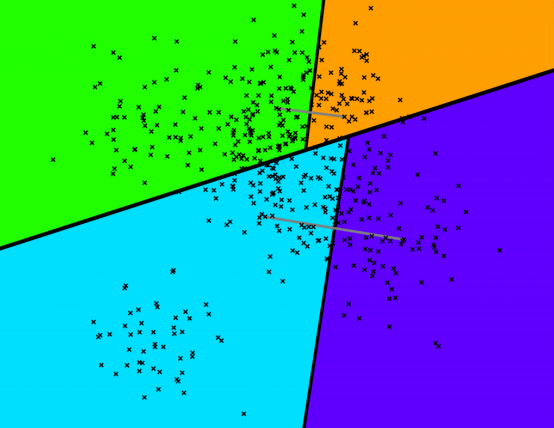
1. kNN uses O(nd) and O(nd) in space and time complexity.
2. If query point is near to boundary then also look 2d (d is dimension) adjoining regions.
3. For higher dimension kdtree is useless.
4. LSH Probabilistic in nature. So it will give output but not same always.
5. **ANNOY (Approximate Nearest Neighbors Oh Yeah)**
6. Below are the points which are projected on space.



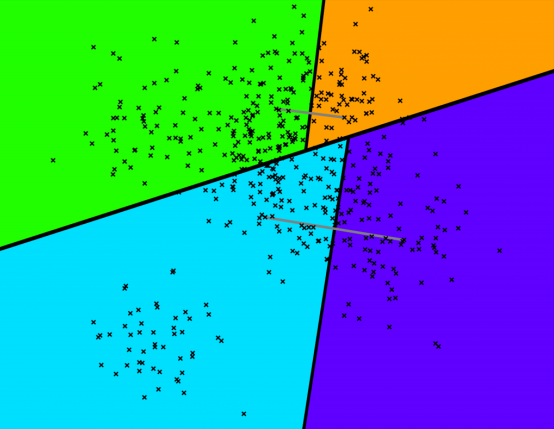
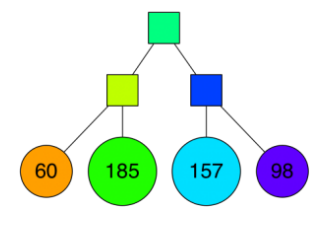
1. Pick two points randomly, split the hyper-space.



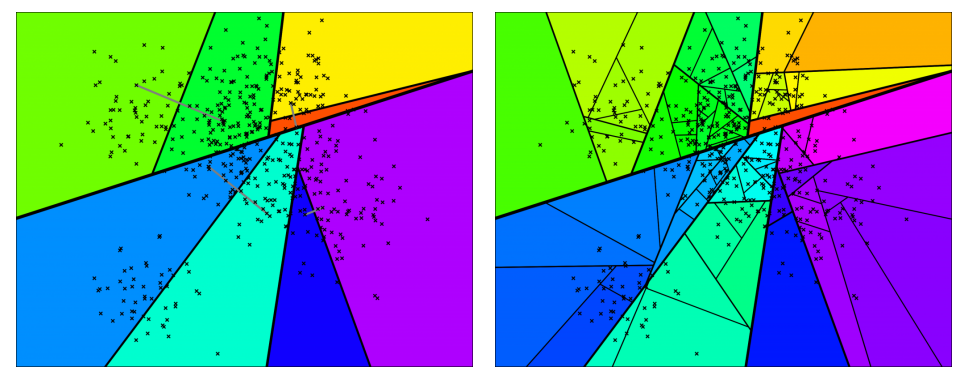
1. Split Recursively



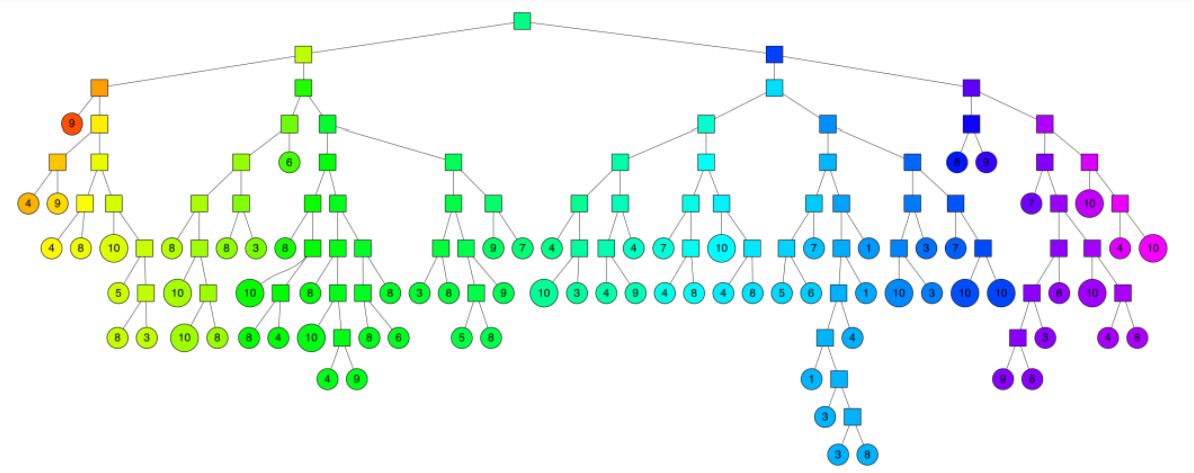
1. Split Recursively - Tiny Binary Tree appears.

1. Keep Splitting



1. End up with Binary Tree Partitioning the Space.
2. Nice thing : Points that are close to each other in the space are more likely to be close to each other in the tree

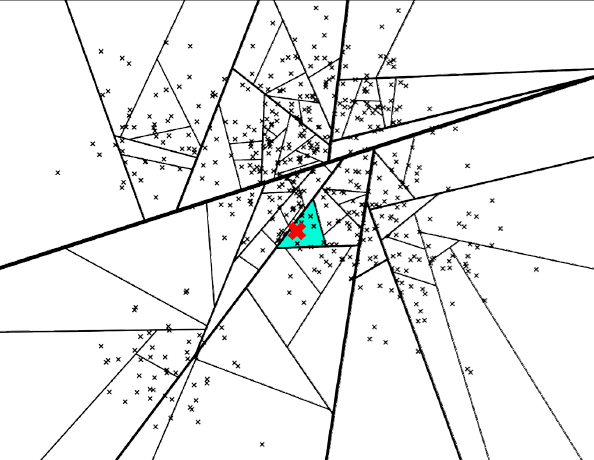


**Testing**

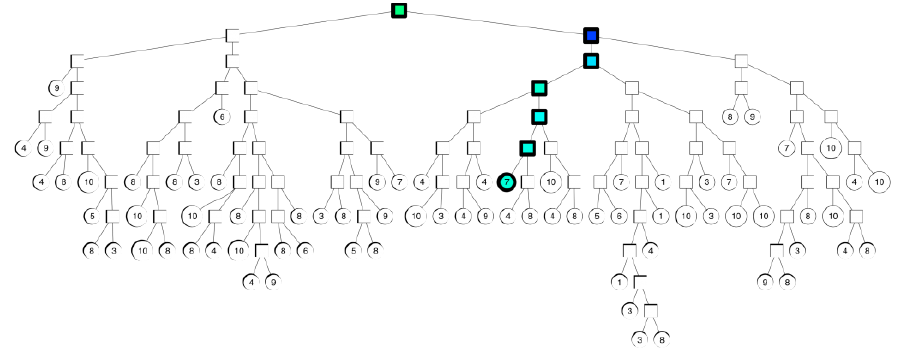
1. Extract the image features.

Using transfer learning extract all the feature from testing image. Pass all the features which are extracted from pre-trained model to ANNOY indexer and ANNOY will return all approximate nearby images.

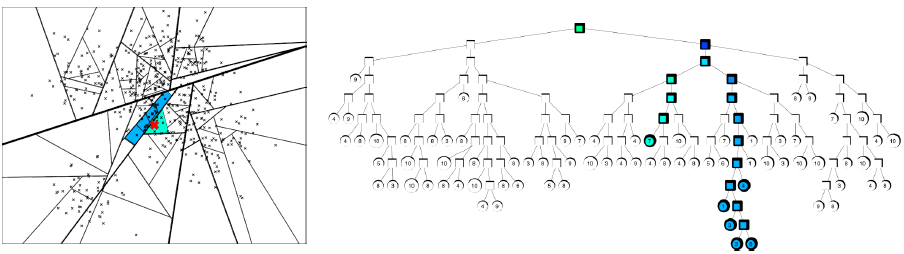
1. Searching for a point



1. Searching for a point: Path down the binary tree. We end up with: 7 neighbors



1. What if: We want more than 7 neighbors… Using Priority Queue we can traverse into both side of child trees.



Advantages –

1. ANNOY stores tree based data into mmap (memory mapped file).
2. This mmap file can be shared by multiple process.
3. This is too fast than LSH.
4. Accuracy too high than LSH.
5. For indexing ANNOY takes time but accuracy on testing data point too low.

**How to Run A Code**

**Step 1: Install all the required libraries**

All the required libraries stored in the requirements.txt

**Step 2: Run imagesearch.py**

Run imagesearch.py on python prompt.

**Step 3: Open web browser**

Visit <http://127.0.0.1:65000/>.

**Step 4: Train Images**

1. Select metric (distance type).
2. Number of trees – which will form forest of n\_trees.
3. Model file name – Give model filename without extension.
4. Upload all the Images on which you want to prepare a model.

**Step 5: Prediction**

1. Upload any image.
2. Give number of nearest images you want.
3. Select model which we prepared in last stage.
4. Select model which you prepared earlier so give meaningful model filename so that you can be able to get correct model as per image data.