ANN (APPROXIMATE NEAREST NEIGHBOR)

Sarina Heshmati

Yasamin Tafakor

sarinaheshmatii@gmail.com

yasamintafakor@gmail.com

INTRODUCTION

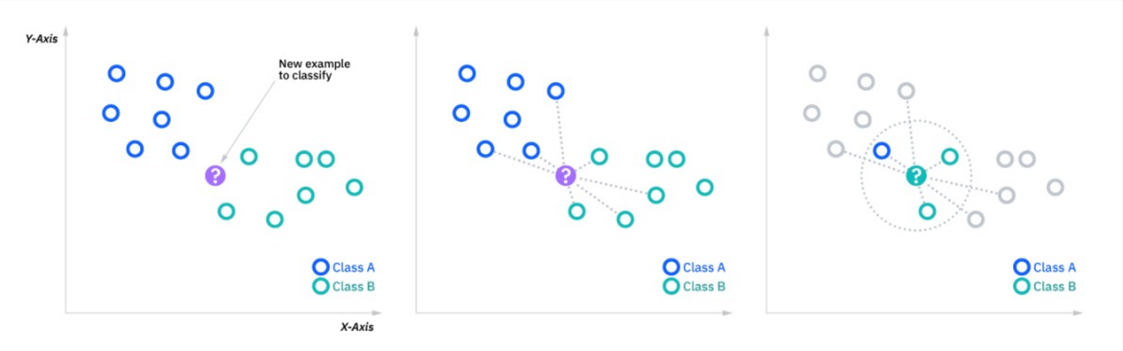
INTRODUCTION

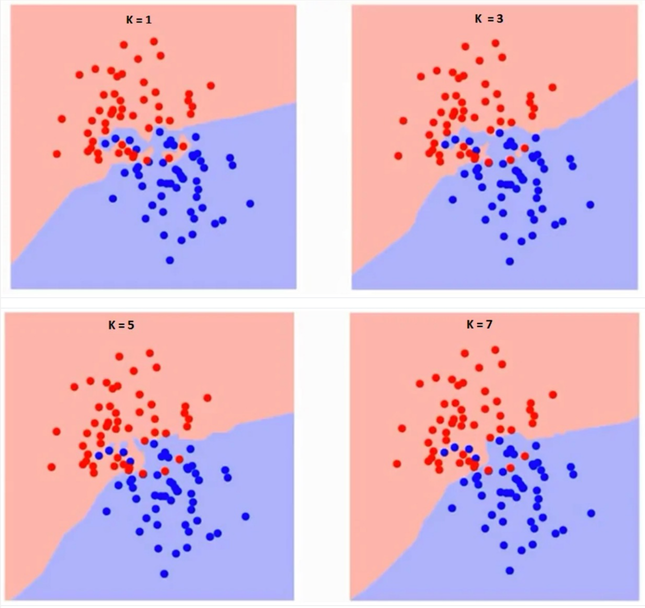
- Nearest Neighbors Motivation
- What is KNN?
- What is ANN?
- KNN vs. ANN

KNN

ALGORITHM

- Non-parametric
- Supervised classifier
- How does it work? (lazy learning)
- Choosing the right K





PROS & CONS

PROS

- Easy to implement
- Adapts easily
- Few hyperparameters

CONS

- Highly time consuming (single query = O(N))
- Does not scale well (specific DS :'ball-tree')
- Curse of dimensionality
- Prone to overfit

ANN

ANN

- Basic intuition
- Much faster – each query = O(log N)
- Trade off between accuracy & time
- KNN is not really an option to consider
- Different implementations such as:

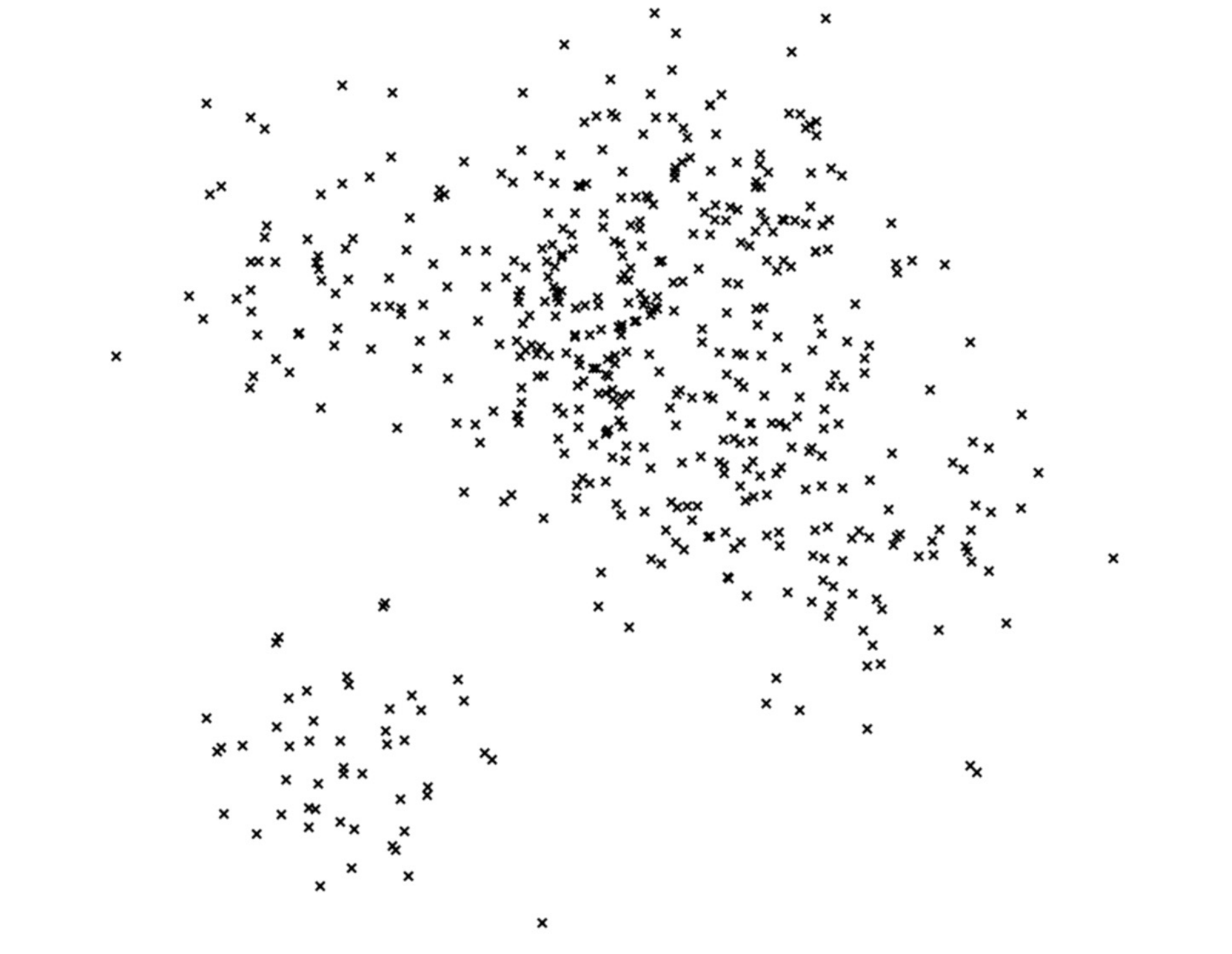
1. Facebook's PySparseNN (for sparse interaction matrices)
2. Spotify's annoy (Approximate nearest neighbor, Oh YEAH)

HOW DOES IT WORK?

HOW DOES IT WORK?

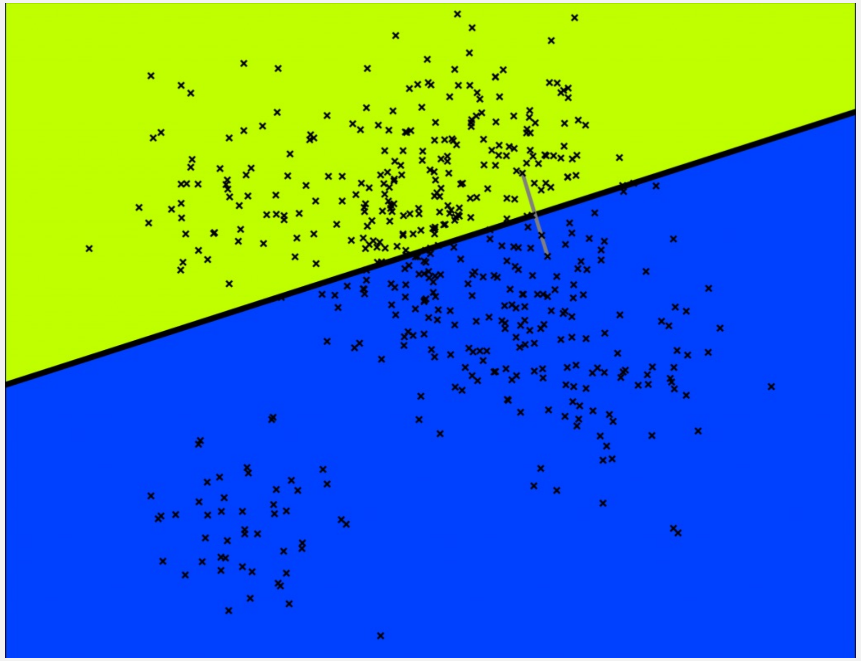
- Annoy
- The explanation will be in 2D
- It will apply to much higher dimension
- Goal : to build a data structure that let us find the nearest points to any query points in sublinear time
- Its gonna be a Tree!
- A binary tree that each node is a random split

How does it work?
06 of 20



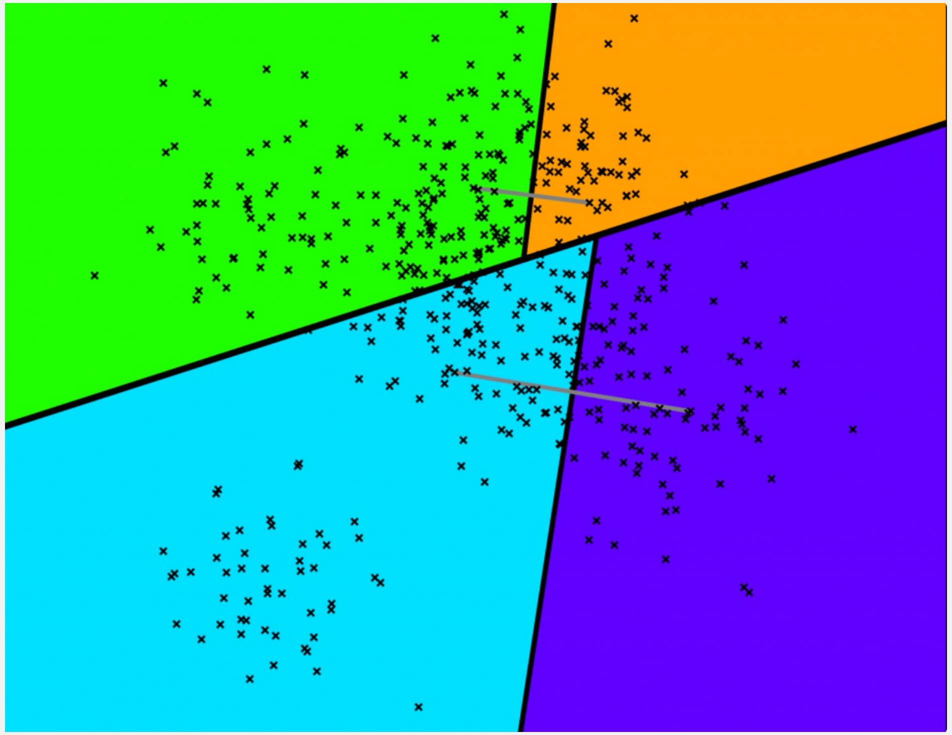
PREPROCESS

- pick two points randomly and then split by the hyperplane equidistant from those two points



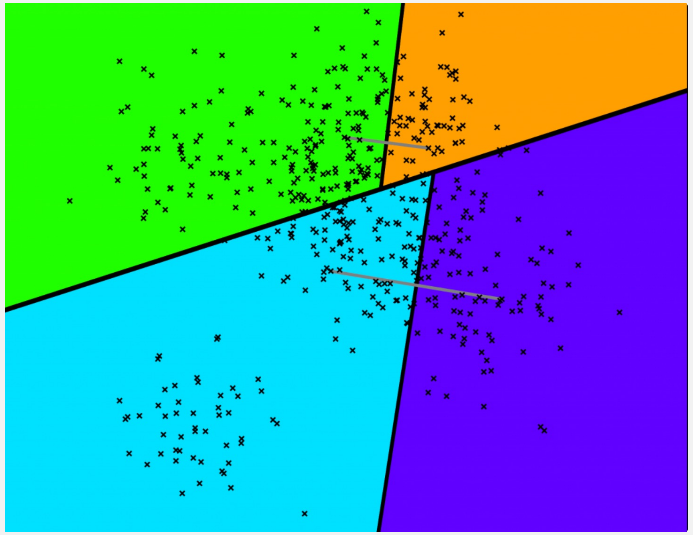
PREPROCESS

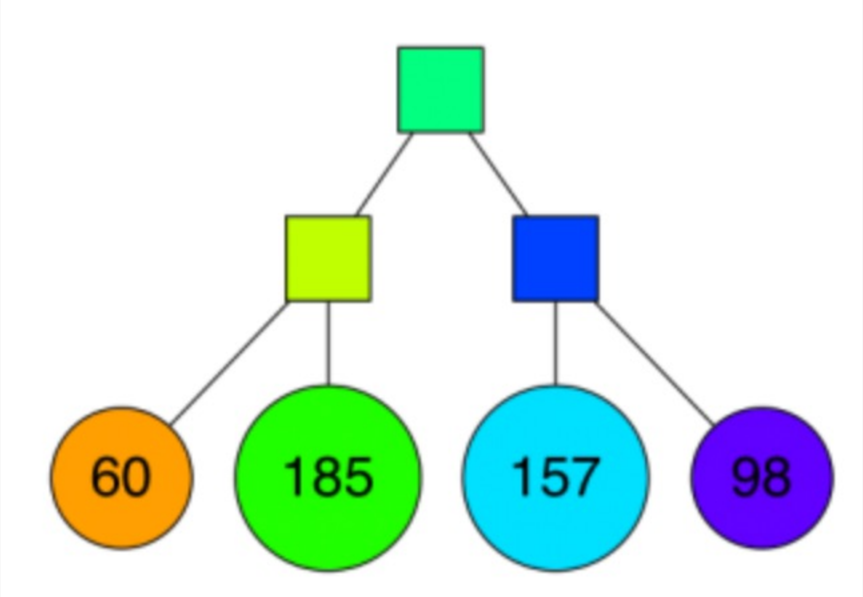
- keep splitting each subspace recursively



PREPROCESS

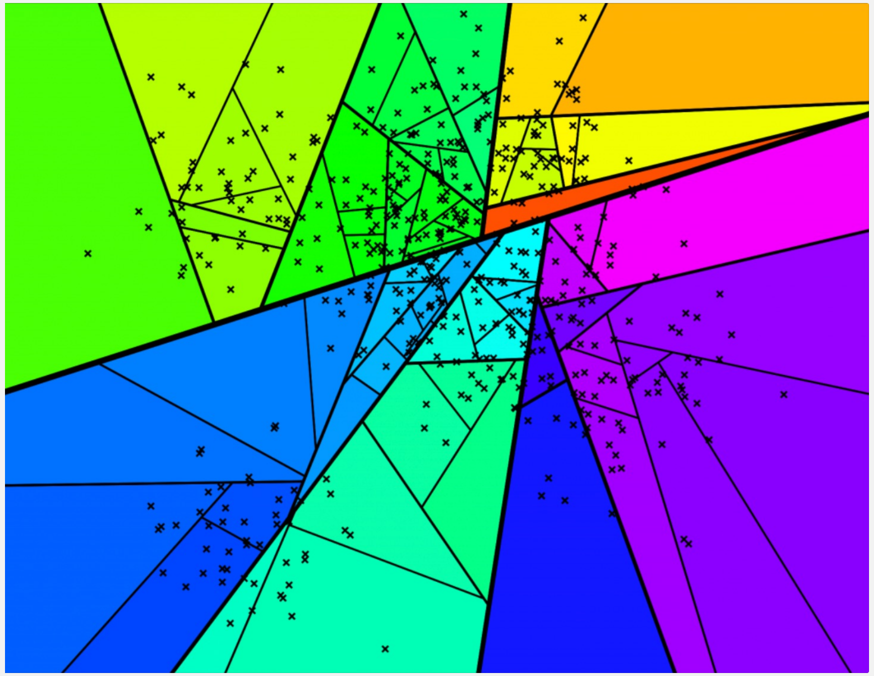
- keep splitting each subspace recursively
- This is our tree so far!





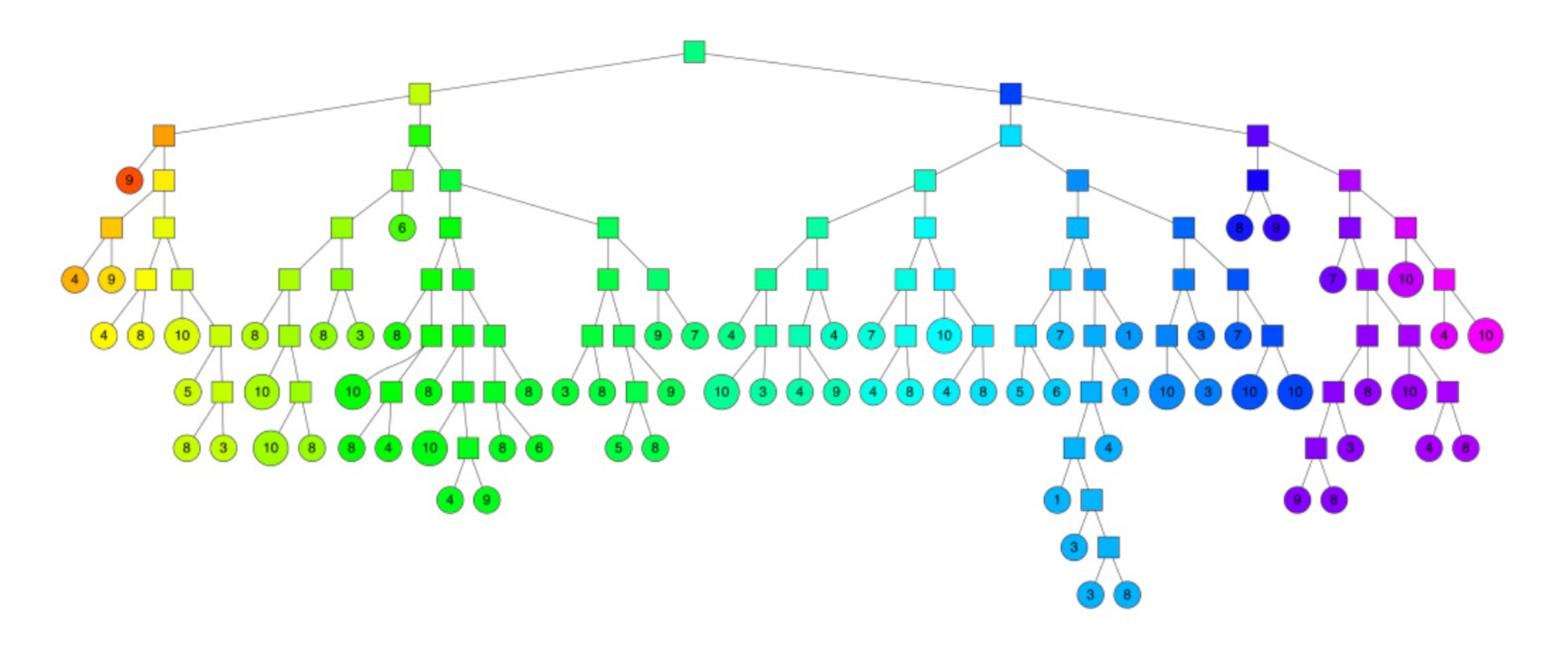
PREPROCESS

- We keep doing this until there's at most K items left in each node (K=10)



PREPROCESS

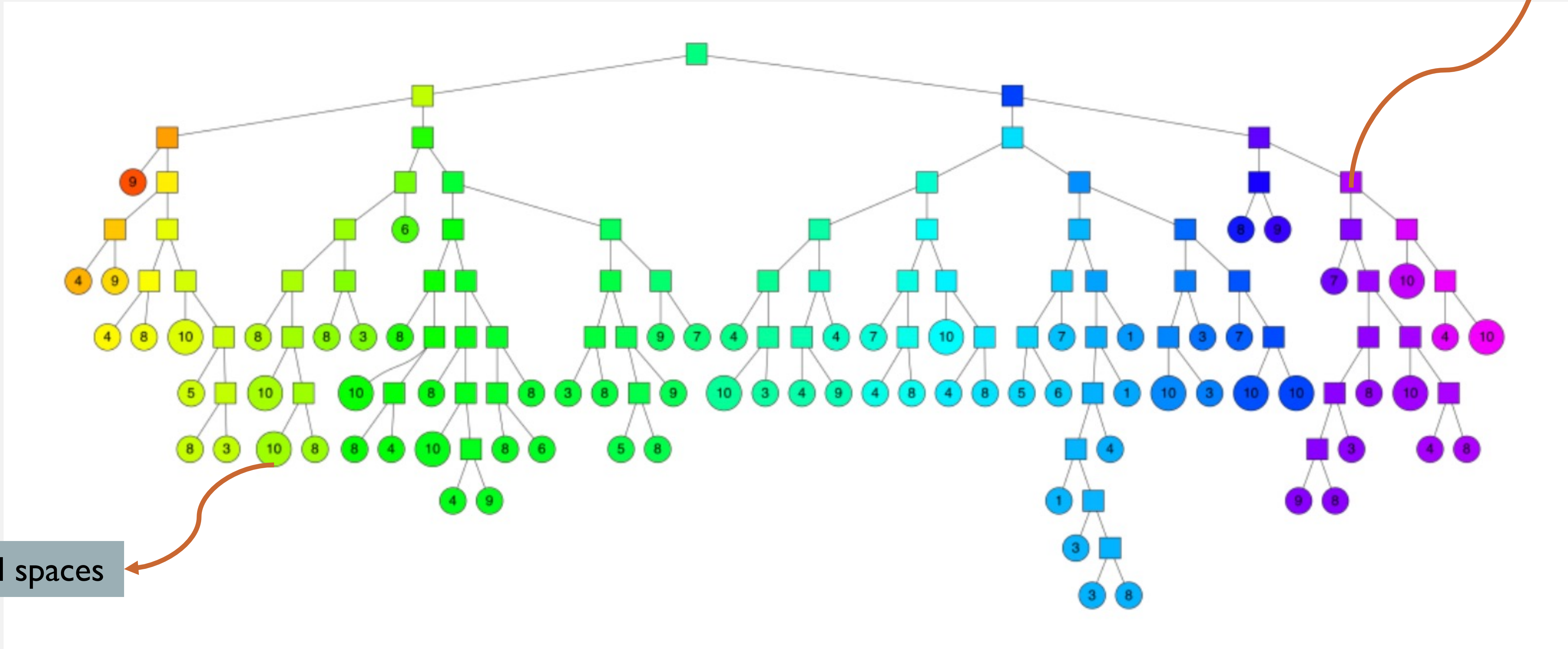
- Final tree



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

PREPROCESS

Hyperplane; to figure out which side of the hyperplane to go

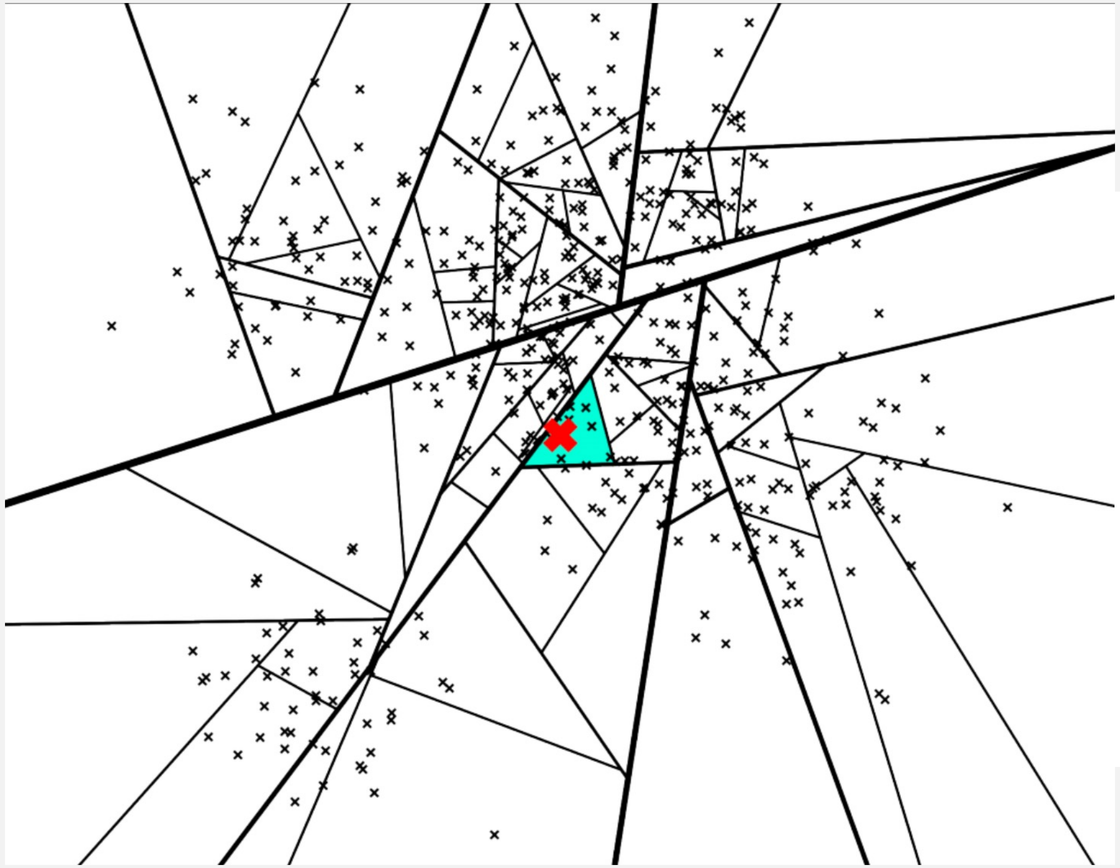


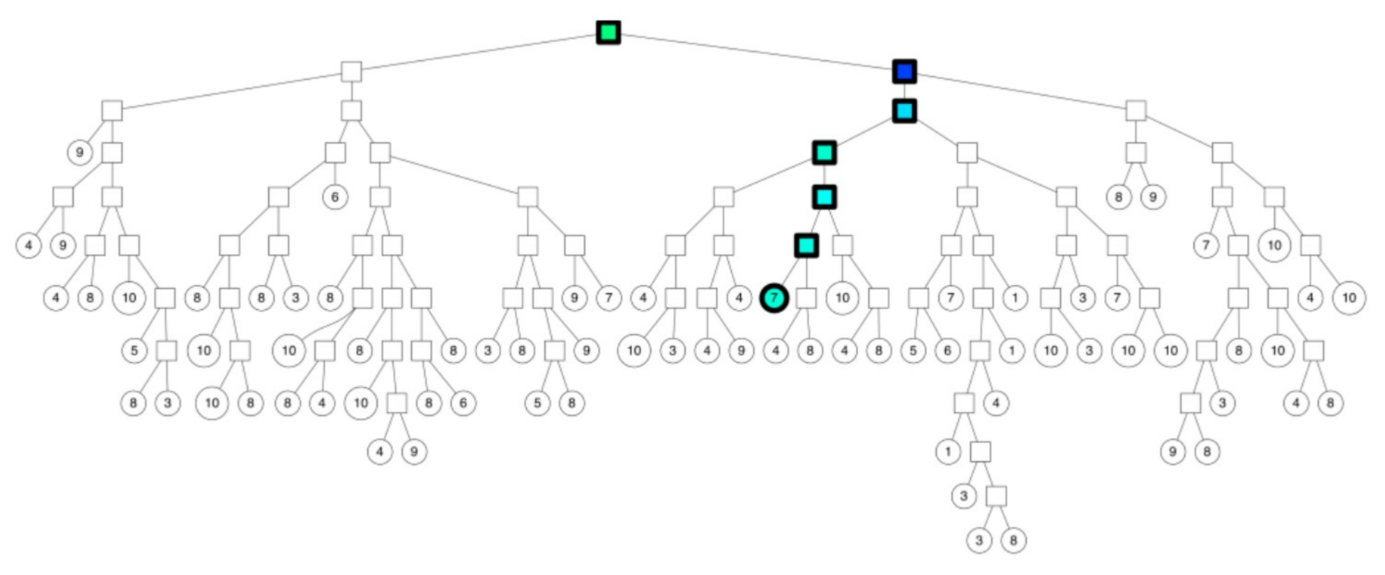
The final spaces

PREPROCESS

- The resulting binary tree partitions the space
- points that are close to each other in the space are more likely to be close to each other in the tree
- In other words if two points are close to each other in the space, it's unlikely that any hyperplane will cut them apart
- Searching for a point can be done in logarithmic time since that is the height of the tree

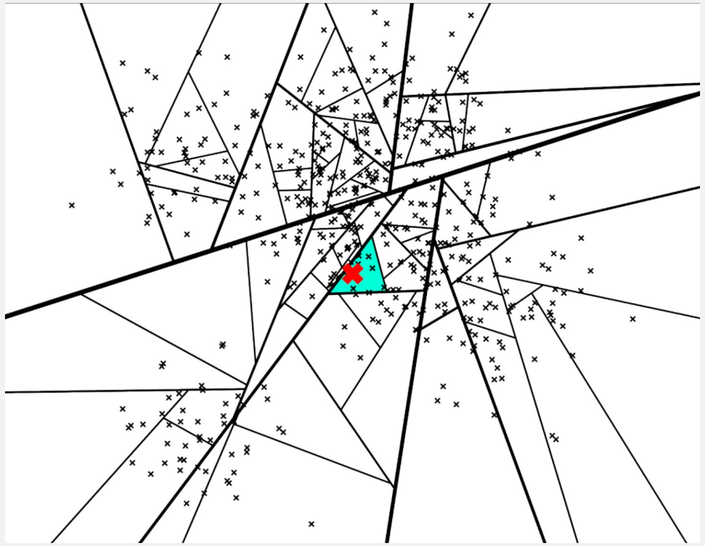
QUERY

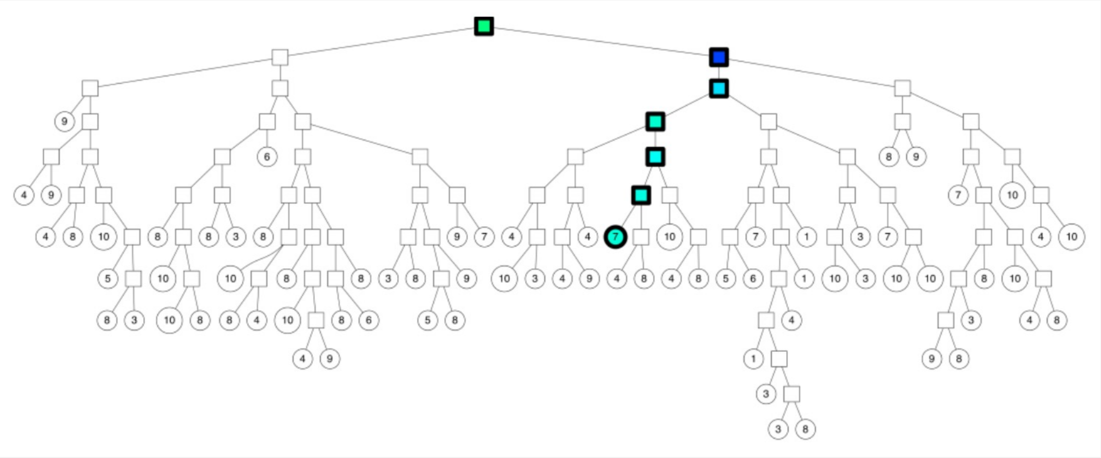



9
6
8
9
4
9
7
10
4
8
10
8
8
3
8
9
7
4
4
7
10
7
1
3
7
4
10
5
10
10
8
8
3
8
9
10
3
4
9
4
8
4
8
5
6
1
10
3
10
10
8
10
8
3
10
8
8
4
10
8
6
5
8
4
3
4
8
4
9
1
9
8
3
3
8

NOT GREAT

1. What if we want more than 7 neighbors?
2. Some of the nearest neighbors are actually outside of this leaf polygon



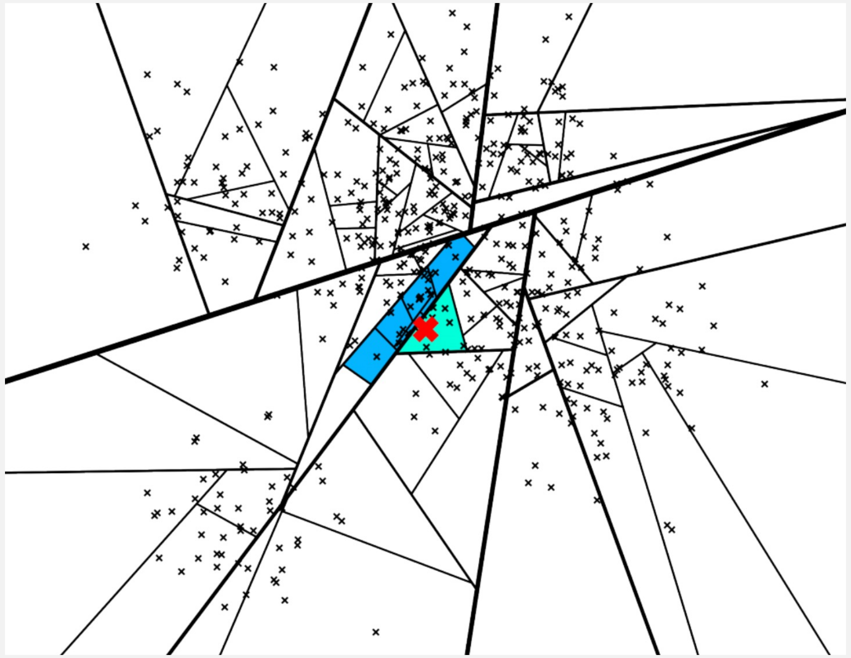


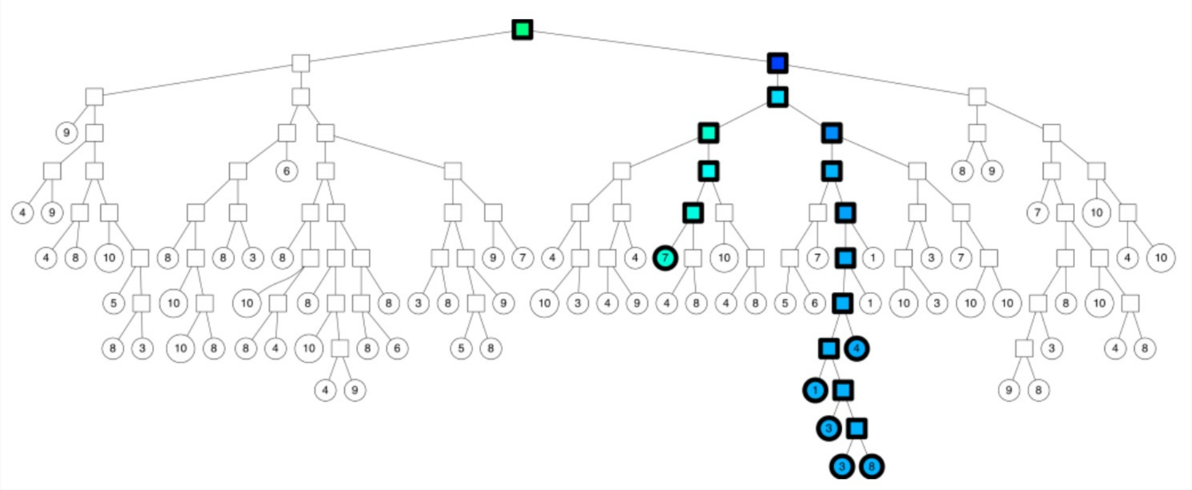
IMPROVEMENTS

TRICK 1

- We go down on both sides of a split if we are "close enough"
- We acknowledge a threshold of how far we are willing to go into the "wrong" side of the split. If the threshold is 0, then we will always go on the "correct" side of the split.
- So instead of just going down one path of the binary tree, we will go down a few more

TRICK I

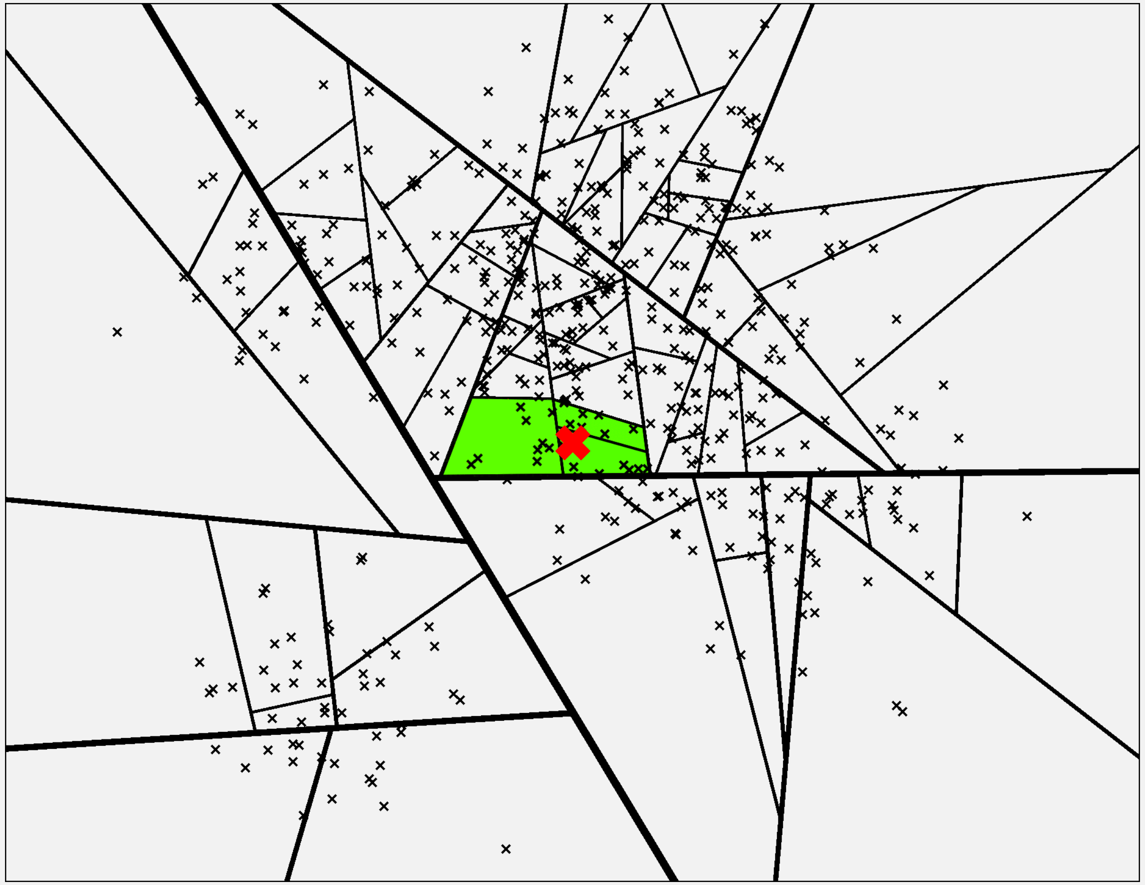



9
6
8
9
4
9
7
10
4
8
10
8
8
3
8
9
7
4
4
7
10
7
1
3
7
4
10
5
10
10
8
8
3
8
9
10
3
4
9
4
8
4
8
5
6
1
10
3
10
10
8
10
8
3
10
8
8
4
10
8
6
5
8
4
3
4
8
4
9
1
9
8
3
3
8

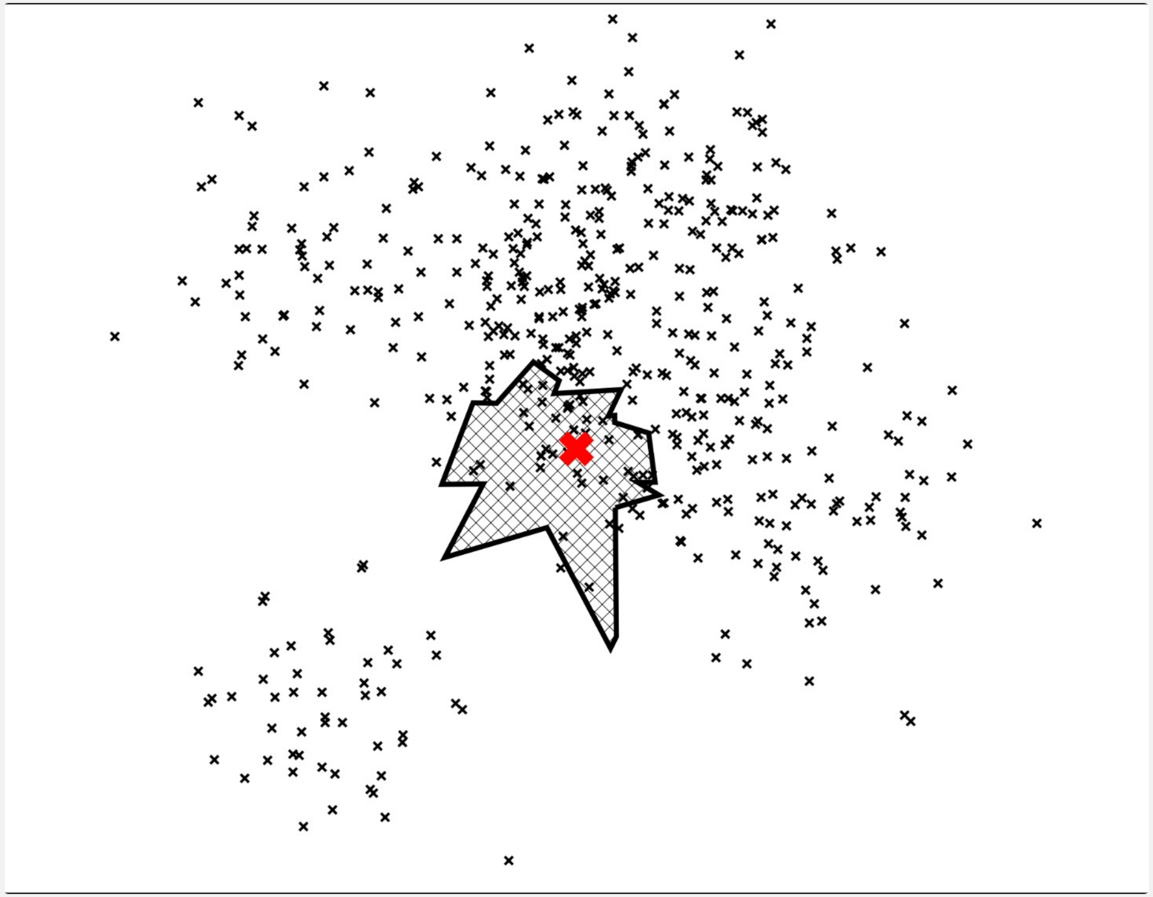
TRICK II

- Build a forest of trees
- Search down all these trees at the same time
- If we look at the union of the leaf nodes we get a pretty good neighborhood

TRICK II – THE FOREST



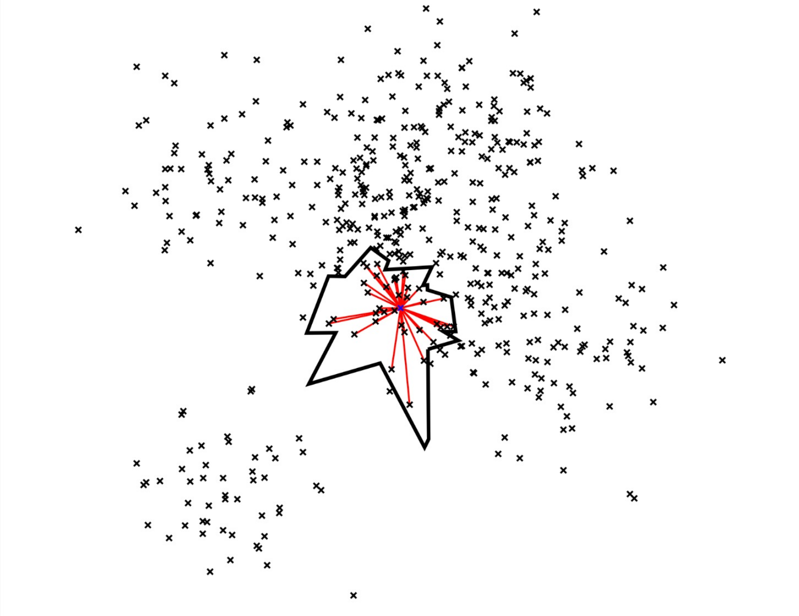
TRICK II – UNION OF THE LEAVES



FINAL STEP

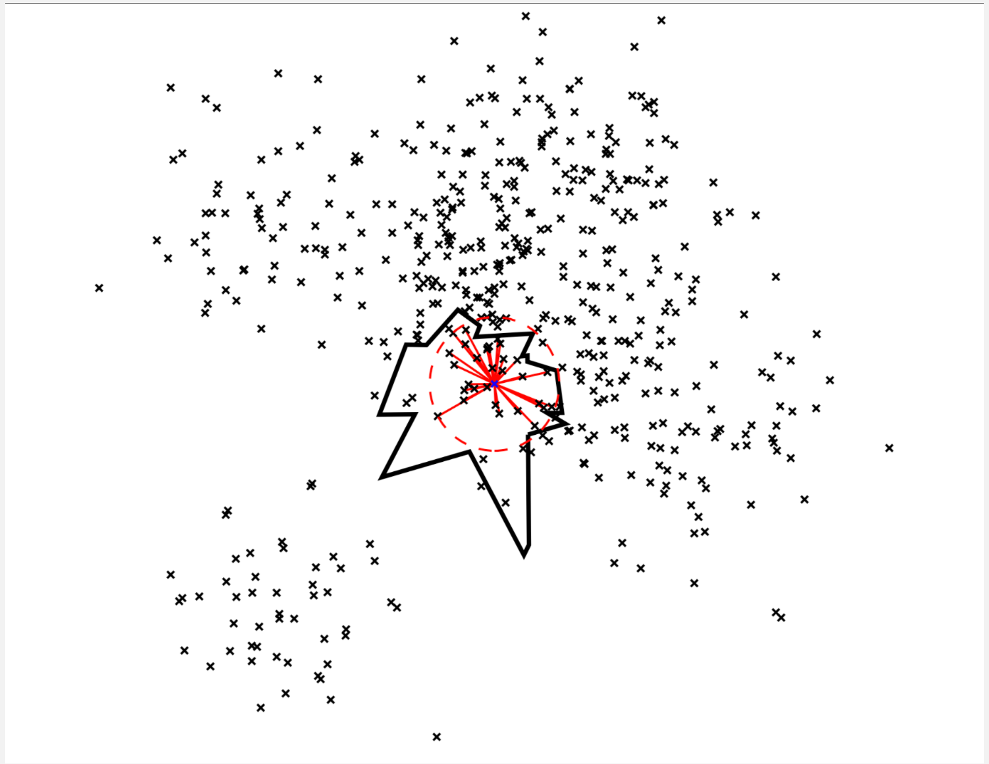
FINAL STEP

- Now we have a small set of points that are almost the closest ones
- We only compute the distance of these points (instead of the whole data set)
- And just like KNN we rank the distances and so on!



FINAL STEP

- The trade off between time and accuracy



FINAL WORDS

REFRENCES

1. https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN,used%20in%20machine%20learning%20today.
2. https://blog.faradars.org/introduction-k-neighbours-algorithm-clustering/
3. https://www.geeksforgeeks.org/k-nearest-neighbours/
4. https://www.linkedin.com/pulse/approximate-nearest-neighbors-ann-mina-ashraf-gamil/
5. https://erikbern.com/2015/10/01/nearest-neighbors-and-vector-models-part-2-how-to-search-in-high-dimensional-spaces.html

THANKS FOR YOUR ATTENTION.
CONTACT INFO:

sarinaheshmatii@gmail.com

yasamintafakor@gmail.com