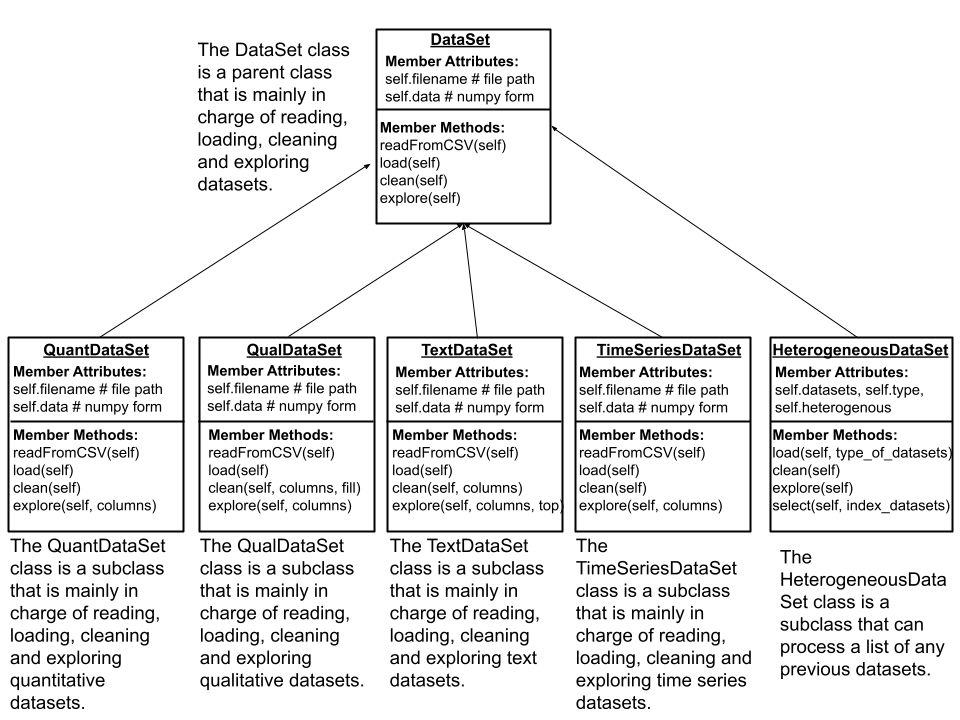
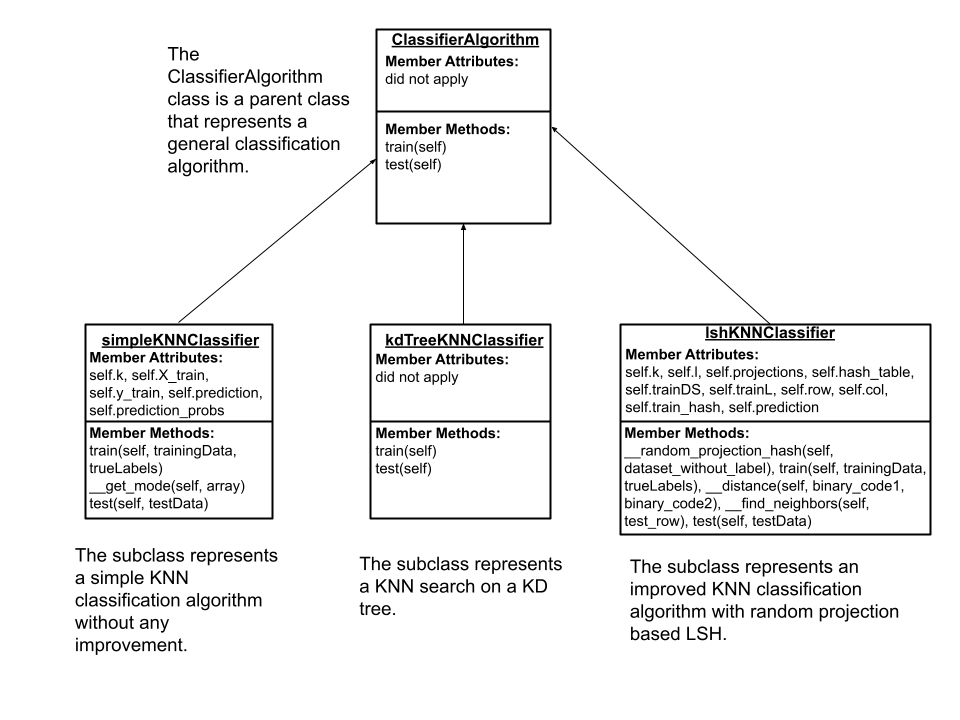
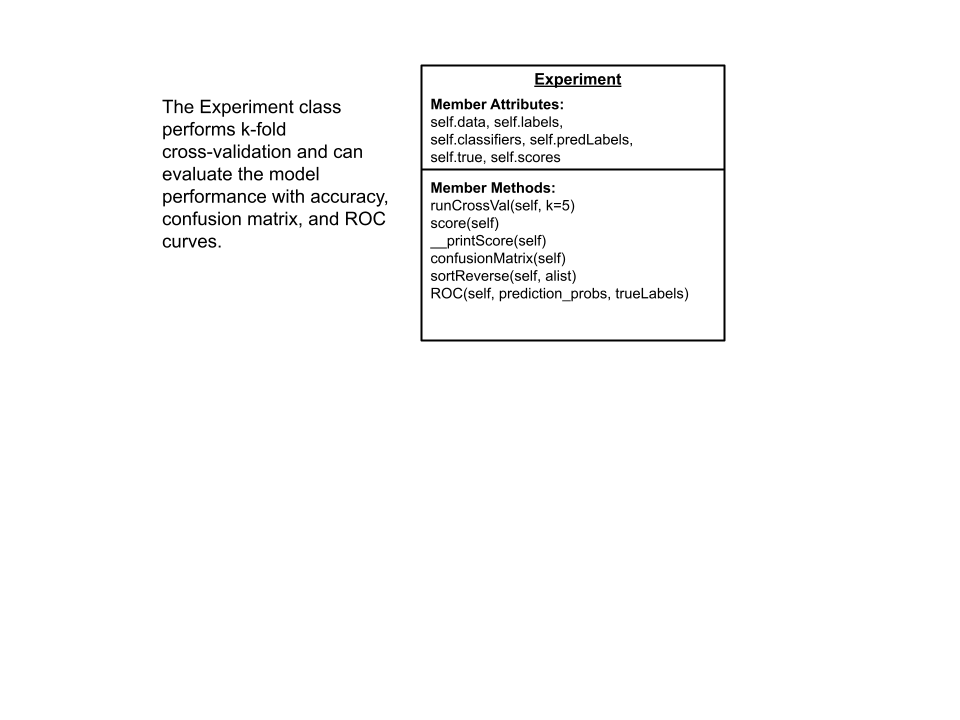
**Summary of Object-oriented Toolbox Design**





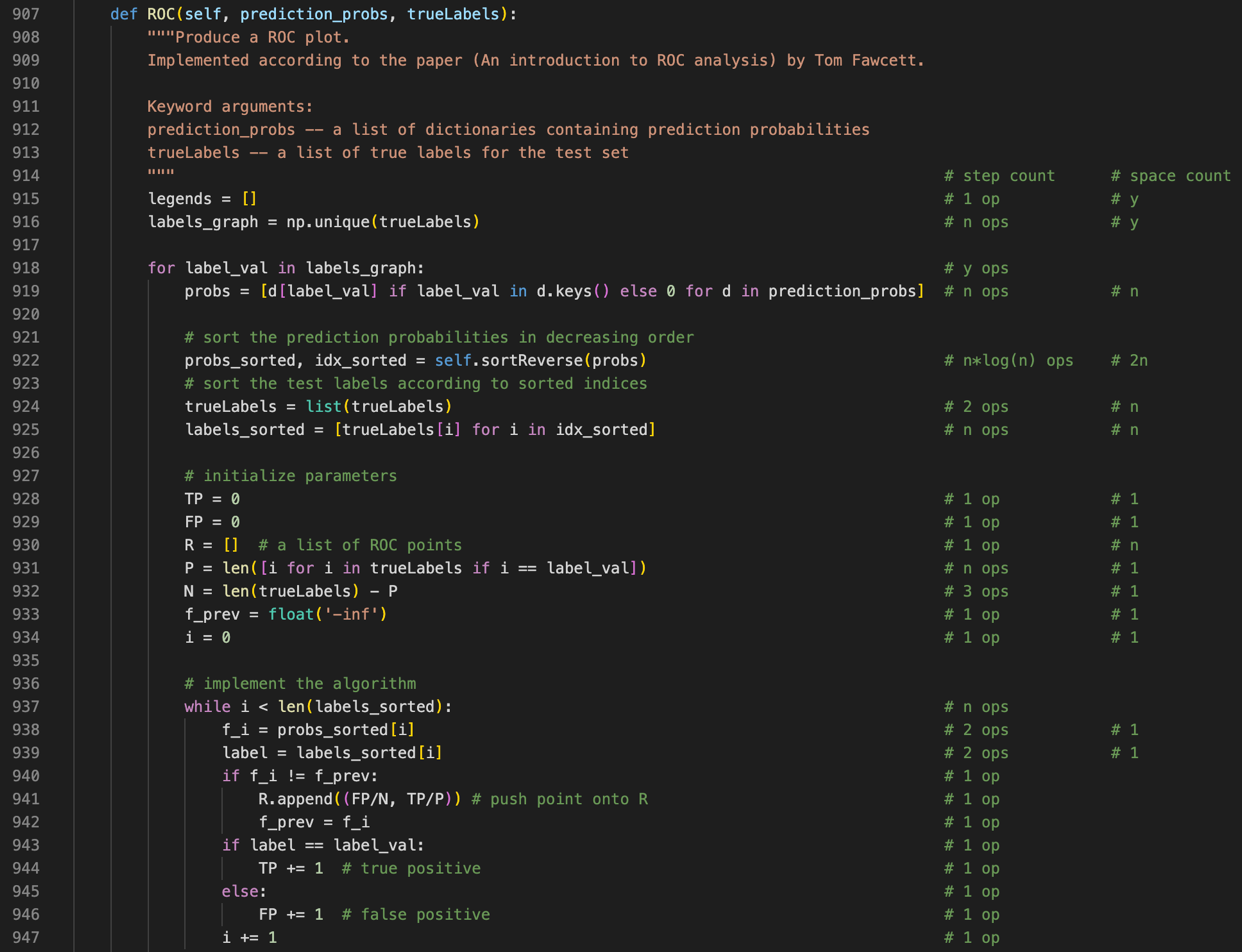


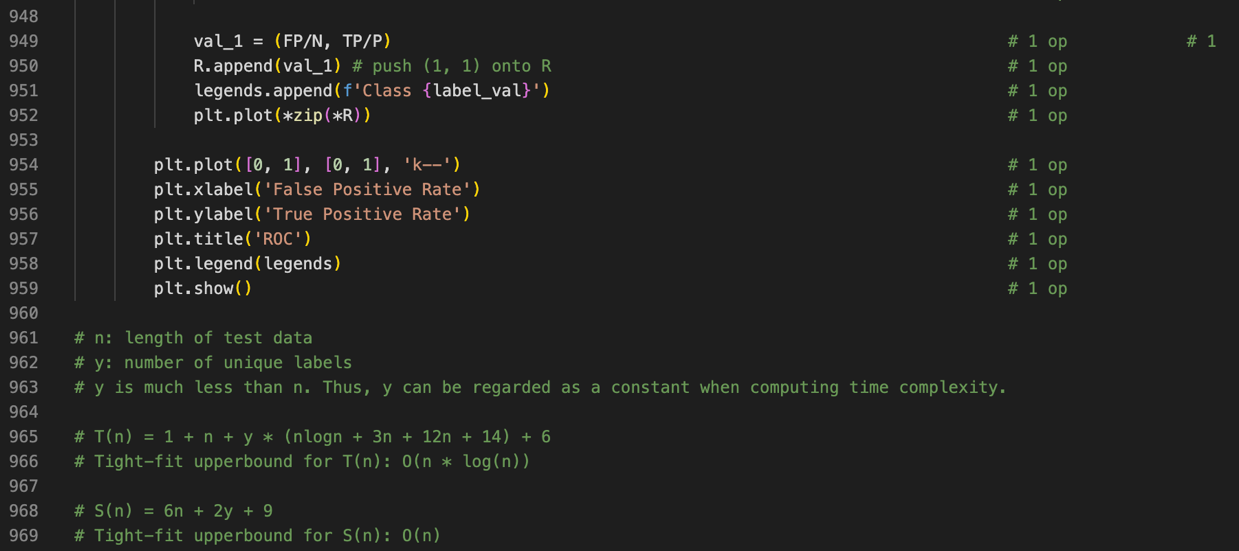
**ROC Algorithm**

**1. Summary**

The ROC algorithm is implemented according to the paper given. We can simply sort the probabilities reversely and update TP and FP for each predicted label. For each positive instance, we increase TP and for each negative instance, we increment FP. We also save each new ROC point to stack R. Finally, we plot the curve with the stack.

**2. pseudo code or actual code**





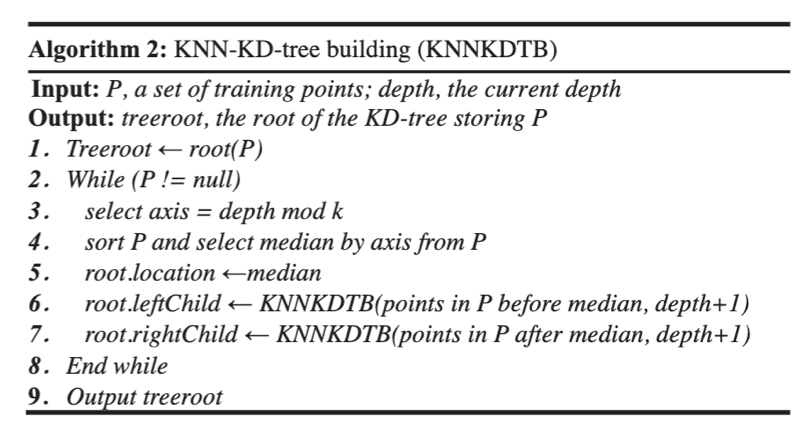
**3. time complexity analysis:** shown above.

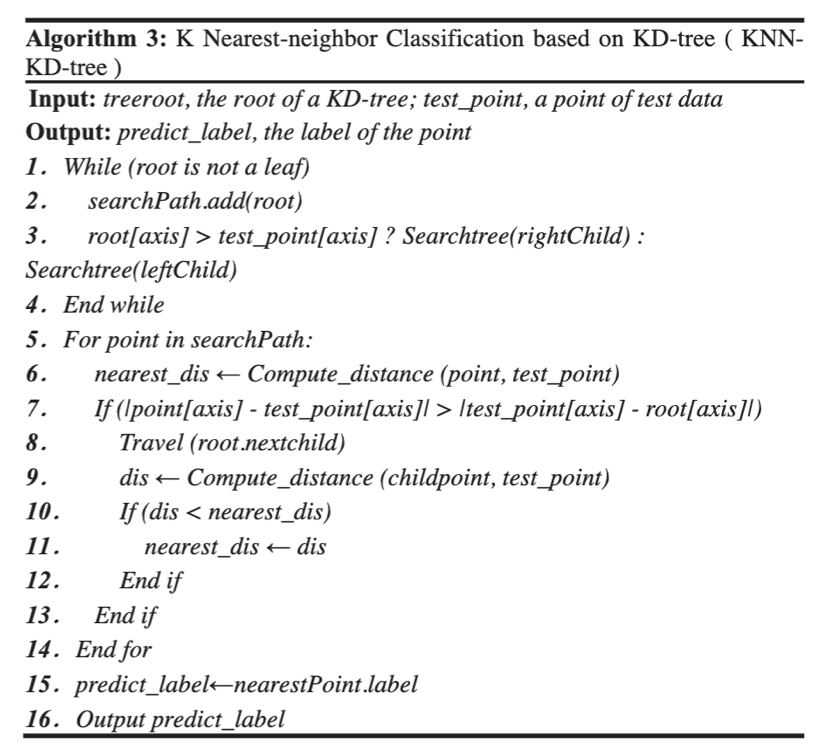
**kd-trees Algorithm**

**1. Summary**

To build a KNN-KD tree, first, select its root node and select an axis. Then sort the first dimensional data and found the median. Then divide the data into root nodes, the left subtrees, and the right subtrees. The above process goes recursively until all data is divided. The search process includes a forward search and a retrospective search. If the data to be searched is smaller than the axis data, it will go to the left subtree. Otherwise, it will take the right subtree until the last point is the leaf node. Then update the points repeatedly in the search path until we find the closest point.

**2. pseudo code or actual code**





**3. time complexity analysis**

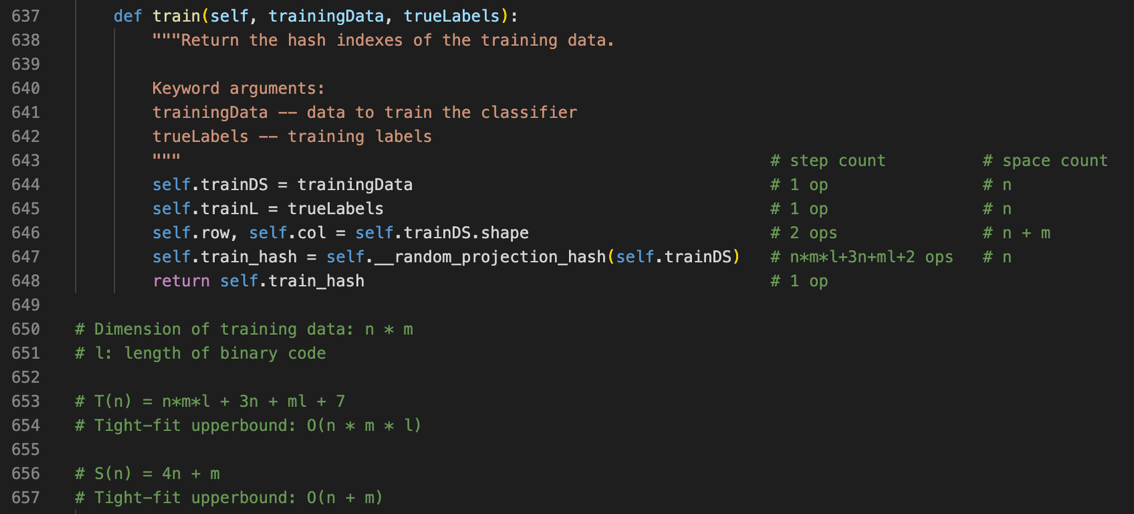
The time complexity of a KNN search on a KD-tree is O(n^(1-(1/k)) + m), where m is the number of nodes and k is the dimension of the KD-tree. In high-dimensional data, it is not as efficient as the simple KNN search.

**lsh-knn Algorithm**

**1. Summary**

First, we create an m\*l random matrix A from the standard normal distribution. Then we project the training data onto l dimensional space with this random matrix A. Then convert the n\*l dimensional matrix to n l-bits binary codes. For each point in the test set, we can compute the Hamming distance between it and each training point. Finally, we sort the distances and get the k neighbors.

**2. pseudo code or actual code (helper functions not include)**





**3. time complexity analysis:** shown above.