# Project 1: *k*-Nearest Neighbor Search

Nearest neighbor search is an important task which arises in different areas - pattern recognition, recommendation systems, DNA sequencing and even game development.

Usually, this task is formulated as follows. We have *N* points in some space (*S* dataset). We have to work with queries, which have dataset *S* and some point *X* as their parameters (*X* does not have to belong to *S*). Typical queries are "find *k* nearest neighbors of *X*" or "find all points in *S* at given distance *R* from *X* or closer". Depending on problem, we may have: a) different number of dimensions - from one to thousands, b) different metric type (Euclidean, 1-norm, ...), c) different dataset size. Hence, for different problems different algorithms are feasible.

The key point of the problem formulation is that dataset *S* is considered fixed. *X* may vary from request to request, but *S* remains unchanged. It allows to preprocess dataset and build data structure which accelerates processing. All strategies which promise better than O(*N*) processing time rely on some kind of preprocessing. Different preprocessing strategies have different features (see <http://www.alglib.net/other/nearestneighbors.php>).

**Problem:** Solve the *k*-nearest neighbors search by means of kd-trees.

**Specifications:**

* Use the kd tree implementation from SciPy (*KDTree* class in the *scipy.spatial.kdtree* package). This class can perform *k*-nearest neighbor queries on multi-dimensional datasets, including finding all dataset points within some distance from the point of interest.
* Present experimental results and discuss the asymptotic efficiency of the algorithm.