## **K-Nearest Neighbor (KNN) Implementation**

### **1. Introduction.**

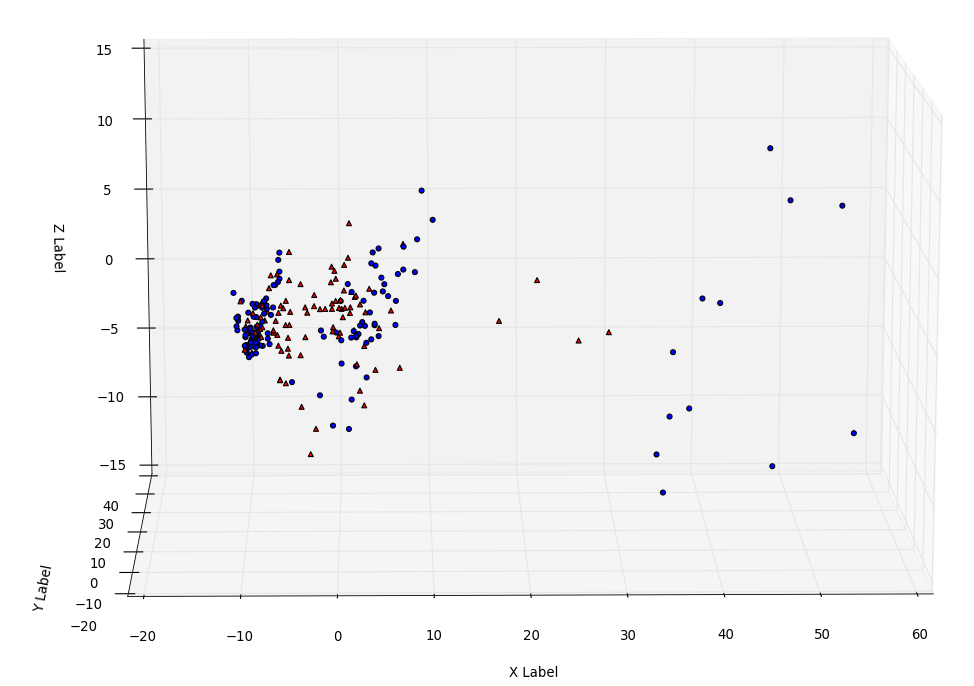
We implement the KNN algorithm in C++ and obtained predictions of revenues for the test data set. We also developed *unittests* for our .*cpp* code.

### **2. Method.**

### The KNN algorithm is one of the most basic and handy algorithm in machine learning. The idea is straight-forward. Consider each training datum as a pair **(x, r(x))**, where **x** is the vector that describes the restaurant, for instance location, type etc., and **r(x)** is the revenue of the restaurant. For a datum in the testing set, denoted **y**, KNN computes its predicted revenue as **r(y) = avg({r(x): x *is one of the* K *nearest neighbors of* y})**. Here **K** is a fixed number of nearest neighbors that are taken into account.

### **3. Training data and testing data**

In our data set, we have 100,000 data points in the testing set and only 137 data points in the training set. Below is a figure of the 3d positions of these data points. In this figure, the blue points are the first 100 testing data points, and the red points are the 137 training data points, presented in the reduced 3-dimensional space. Due to the large size of the testing set, we choose not to render all the data points in this figure.



### **4. Implementation in C++.**

We have the following source and header files associated to KNN implementation: kNN.cpp, kNN.h, kNN\_main.cpp, kNN\_main.h, KNN\_main\_unittest.cpp.

kNN.cpp, kNN.h define the class kNN, which computes the kNN prediction of testing data given training data.

kNN\_main.cpp, kNN\_main.h have the main function and its header.

KNN\_main\_unittest.cpp has the unittests for each of the functions in kNN\_main.cpp and member functions of classkNN.

Output of the algorithm is in file kNN\_out.txt, with predicted reveneus in original order of the testing data set.