Working with Geographical Data. Geohash Encoding

Many datasets that are relevant for Management Analytics are geographically based, that is each row of data refers to a specific geographical location. Traditionally, the geographical coordinate system based on longitude and latitude (see https://en.wikipedia.org/wiki/Geographic_coordinate_system) are used to identify a geographical location. While this system, dating back to 3rd century BC, is very useful, it creates issues in linking different data sets.

For example, suppose we are interested in how many traffic accidents occur near schools. We may have an ACCIDENT dataset listing every accident (say over last year), including its longitude and latitude. We may also have a SCHOOL dataset listing the coordinates of every school. How do we easily link the two? One option is for every accident to compute the distance to every school, find the closest one, and add it that school (as well as distance) to the ACCIDENT data set. This is quite a cumbersome operation (especially if the original datasets are large). Moreover, if a given accident is not within some reasonable distance (say 200 meters) of any school, whether the closest school is 2 or 5 kms away may be irrelevant. More generally, since no two geographical locations are likely to coincide exactly, linking two or more datasets organized at the geographical level can be challenging.

Geohashing https://en.wikipedia.org/wiki/Geohash is a technique to overcome this difficulty and make linking easier. The basic idea is to subdivide space into a rectangular grid, and associate each (longitude, latitude) point with the rectangle it belongs to. This is done by encoding the geographical coordinates into a long bit string, converting it to base 32 (to manage the length) and chopping of digits at the end to manage the grid size. For example, the coordinates of the Rotman School are (43°39'55.9"N, 79°23'54.1"W). Converting from degrees/minutes/seconds to decimals, we get (43.665527, -79.39833); the minus sign just designates that longitude as being "W" rather than "E". Converting this to a 9-character geohash (e.g., using the automatic converter from https://www.movable-type.co.uk/scripts/geohash.html) we get geohash code "dpz83jp7x", which identifies a 4.8m x 4.8m area at the precise coordinates above. Of course, the area of the Rotman school is much larger than this tiny area, so the 9-digit geohash location is a bit arbitrary (it could be anywhere in the school). But this is where geohasing really shows its power! All we need to get a larger (and less arbitrary) location is chop off digits at the end. Thus, the 8-digit geohash "dpz83jp7" identifies a 38m x 19.1m area containing the smaller 9-digit box, and the 7-digit geohash "dpz83jp" identifies a 153m x 153m which covers most of Rotman. The pictures on the following page illustrate this. Since the 7-digit area actually covers the school, it appears to be the most reasonable level to use in this case.

Another useful feature of geohasing is that finding areas that are near a given location is also not hard. For every geohash, there are 8 neighboring "boxes" (i.e. geohashes at the same level of precision); most software will allow you to find these - the pictures on the next page identify the neighboring geohashes. Thus finding all accidents near a school is relatively. The first step is to add (for example) 7-digit geohash to both ACCIDENT and SCHOOL files. The next steps depend on how precise you want to be:

- The simplest approach is to link the two files by 7-digit geohash and count how many accidents are near each school. This will work most of the time but not always – if an accident occurs in a neighbor geohash, it may still be close to a school but will be missed
- A more precise approach is to find all 8 "neighbor" geohashes for each school and count the total number of accidents in the 9 geohashes (the one containing the school and the eight

neighbors). The software links listed below allow you to compute both, the geohash for a given geographical location, and the 8 neighboring geohashes.

For the Datathon, we have added a 7-digit geohash to each of the data files. This should be sufficiently precise for the questions asked in the case. However, you feel you need more precision (many files you are given, e.g., the TPS KSI data set, contain specific coordinates), you can convert these to a 8-digit or 9-digit geohash codes using Geohash module on python (see https://pypi.org/project/Geohash/) or geohash package in R (https://cran.r-project.org/web/packages/geohash/vignettes/geohash.html) or using Proc GEOCODE in SAS (http://www.sascommunity.org/wiki/Geocoding). The same links will help you identify the neighboring geohashes.

The following pictures were obtained from https://www.movable-type.co.uk/scripts/geohash.html