Database Systems

HW 7: *k-Nearest Neighbors for Similarity*

Our company is doing well in the greater Los Angeles area, and wants to open some new locations.

Our most successful location is in the Los Angeles zip code 93591.

We have a list of 319 zip codes in the area that we can consider for new locations.

Your job is to recommend the 10 best candidate zip codes for us to consider, based on their *demographic* similarity to our most successful location (zip 93591).

To find similarity, we will define a *distance function* that measures demographic similarity, and apply that function to find the demographic distance between our most successful zip code and every other zip code.

For this case, we will treat three properties like physical dimensions, and use them to compute the *Euclidian Distance* between zip codes.

The three properties that we will use are the following[1] :

(**Median Age, Male Percentage of the Population, Average Household Size)** [2]

The *Euclidian distance, d(point **p**, point **q**)*, is defined as follows:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

This is just the good old-fashioned geometric distance formula, a.k.a. the Pythagorean Theorem.

You can now think of each zip code as being a point in 3D space with coordinates (x, y, z), where

x = median age,
y = male percentage of the population, and
z = average household size.

Once we compute the distance between each candidate zip code and our target zip code (93591), all we need to do is order the zip codes by their demographic "distance" from 93591, and take the 10 smallest distances.

These are the 10 "nearest neighbors" in 3D feature space - where our features are *demographic properties*.

Your Assignment:

Download the .csv data file.
Create a table with the proper data types for the census data.
Load the data from the .csv file into your table.
Compute the 3-Dimensional demographic distance
between each zip code and zip code 93591.
Sort the results and report the Top 10 closest zip codes by demographic distance.
These will be your recommended candidate locations.

Hints:

Watch your integer division vs. float division - you may have to cast a few things to float.
Watch for division by zero - you may need to impose some conditions WHERE columns != 0.

Begin by selecting only the values for zip code 93591 into a table (the 'target" table).

Make sure you have the male (or female) population percentage in your 93591 table
AND also in the main "census" table - which contains all the rest of the zip codes.

SELECT the *distance computation* from your "target" table *CROSS JOIN*ed with
the "census" table. This will give you the distance between the zip code 93591
and every other zip code.

From there, you've done the hard part - now just order by distance, and take the top 10.

Extra credit:

Statistically n*ormalize*[3] this data before computing the similarity between zip codes, and compare it to your non-normalized results. If you choose to do this, consider using the following normalization method:

$$x_{normalized} = \frac{x - min(x)}{max(x) - min(x)}$$

Which can be accomplished in SQL as follows:

```
SELECT
   (column - MIN(column) OVER()) / (MAX(column) OVER () - MIN(column) OVER ()) AS col_norm,
   ...
FROM  ...  ;
```

Submit:

One .sql file with the queries you used to get your answers, and
one separate .txt file with your final zip code recommendations.

1. Really, these should be statistically *normalized* first - so that all measure are of equal magnitude.
See "Extra Credit" section for more information.

2. Note that if we have percent males, we also have percent females - they are not independent.
You can use either one.

3. Note that this is a completely different definition of *normalization* than the one used in *first-normal-form,*
(1NF, 2NF, 3NF), etc. This means roughly: "making all column values the same size for comparison".