ECE 656 Winter 2017: Project

There is an extremely large quantity of data stored in relational databases. While we know how to do basic transactions on such data (*e.g.*, debit account to make a bill payment; book a flight; *etc*.), there is an immense amount of information implied by that data which is much less obvious. For example, when a person books a flight, we know that person, or someone close to that person, will be in a different location for some period of time. We also know that person has the resources necessary to book such a flight. Depending on where the person is going, it might be a business or personal trip. If we correlate that flight with other data about the person, we may deduce more facts. For example, if the person has booked many such flights to the same general location, but it is not particularly a business location, it may imply a sick relative. Other data may be available allowing confirmation or refutation of such a hypothesis (*e.g.*, there may be data about a parking lot payment near a hospital, which would support the hypothesis). By analyzing the data in a database, we may infer information and knowledge beyond the raw data.

In this project, you are required to analyze the Yelp dataset to see what knowledge can be gleaned from it. However, two things must be done prior to even starting an analysis:

1. Data cleaning
2. Data indexing

An analysis is only as good as the underlying data. If the data is garbage, the knowledge inferred will be nonsense. In the case of the Yelp data, this means that we need to know several things:

1. Does the data satisfy basis sanity checks?
2. Is the sample we are analyzing representative?
3. Are reviewers representative of the general population?

The third of these is not something that we can do much about at this data-cleaning stage, but will be a relevant question when it comes to analysis.

As regards basic sanity checking, data generally has to be viewed as correct (*e.g.*, if a user claims his name is “Fred Jones” we will not assume otherwise) but should be checked for self-consistency (*i.e.*, it is consistent with other data within the database) and consistency with reality (*e.g.*, a review cannot be from the future). Some examples of basic sanity checking include:

1. the identified Years Elite must be both consistent with the time the user started reviewing for Yelp (*e.g.*, the user cannot be Elite in 2005 if s/he only started reviewing in 2007) and with our own knowledge of current data (*e.g.*, the user cannot be Elite in 1855, since there were no computers back then; the user cannot be Elite in 2001 since Yelp did not start until 2004; Years Elite cannot include 2018, since that is the future).
2. The number of reviews written by a user as recorded in the User table cannot be smaller than the number of reviews in the sample set.

The second issue is about the sample selected, and we have looked at this in a very limited way in Assignment 1, Part 3(f) where there were two ways to determine the average rating written by users: the stored value in the Users table and the value determined computing an average of the ratings issued by a user for businesses reviewed. If the sample is representative, then these two numbers should be approximately equal; conversely, the sample is skewed if these numbers are not approximately equal (*e.g.*, if the average-stars value is stored is 4.5 but computed as 2.6, then we have a sample that substantially skews toward that user's negative reviews). Similarly, the review rating for a business can be computed two ways and an unbiased sample will have approximately similar values for these two.

For the data-cleaning portion of the project, you are required to identify as many forms of consistency checking and sanity checking as you can, and implement queries to determine if there are problems with portions of the data. You are further required to recommend solutions for such situations. Solutions can include ignoring some portion of the data for analysis or adjusting the analysis in order to compensate for the data skew.

The second portion, data indexing, requires you to determine what indexes are necessary to enable efficient querying in both the cleaning and analysis portions of your work. You will necessarily need to leave this until after you have determined what analysis queries you will be doing. Once you know your cleaning and analysis queries, you will need to determine what indexes are appropriate. You are required to measure the performance of your cleaning and analysis queries both with and without the indexes (and for the "without" portion, you should also remove all primary and foreign keys from your database). If the time to execute queries exceeds some reasonable thresholds (this is likely on any sizable joins when indexes are not present), you may terminate the query early and report that fact. All timing results will need to be included in your submission report.

Having ensured that the sample is suitably clean, you then need to analyze the data. For this, you are seeking knowledge about what the data can tell you. One approach would be to implement a variant of the *a priori* algorithm which will mine association rules (*i.e*., determine what is associated with what, and to what support level). This can be implemented as a stored procedure and run against the database. Other clustering algorithms may be used and/or you may wish to use the R statistical language.

You are not expected to determine all possible knowledge that can be inferred from the data. To the contrary, it is much better to do a detailed and careful study of some particular things than a shallow study of a large number things. Things to study include the following:

1. Based on the data, determine if a business is declining or improving in its ratings. This is particularly relevant if a business has a very large number of ratings, in which case a small number or recent ratings that are substantially different than the long-term average would not necessarily affect that long-term average, but would reflect a change in the current state of the business.
2. Different users have different ways of evaluating businesses. For example, some will readily rate businesses well, and must be very upset to rate a business badly. Predict what rating a user will give a business based on how s/he has rated other businesses and how others have rated that business? This is particularly valuable in enabling customized recommendations for Yelp users.
3. Do operation hours affect the rating of a business?
4. Does review length affect how other users perceive a review?

Finally, it is, of course, not sufficient simply to analyze all of the data and state the correlations. Validation is necessary. A typical approach to validation would be to divide the data in two, at random. Half of the data is used for analysis, to make predictions about what matters. The other half can then be used to validate or refute your hypothesis. You will want to ensure that your split is representative, per the comments on data cleaning above.

This validation approach would not work for suggested study (2) above. Why not? How could you validate your methodology there?

Write a 20-page (approx..) report describing the work you performed for this project, all timing results (yes, running complex analysis takes time, and so timing your queries is important), and analysis. Submit your report and all code to Learn (a Dropbox will be created for the submissions) by the last day or lectures.