CSE 4701, Spring 2017   Project 2

Part I: Due April 4, 2017 (Tue) 11:59 pm at HuskyCT

Objective: You learn how to build an embedded SQL application designed to access a shared database stored in Oracle DBMS. You build a local database in MySQL by accessing a third party database. You will also learn how to process "raw" data to create intermediate stage tables which are more amenable to data mining—a step called "data preprocessing".

Database Access: You have been given access privilege to an existing Oracle database, called CRCDB (Colorectal Cancer Database). You access it through JDBC connection (if you use Java) and try to answer the questions given below. The information to access CRCDB is given blow:

Login id: cse4701;   Password: datamine;   Port Number: 1521;   SID: BIBCI
Server: query.engr.uconn.edu

To learn how to access an Oracle database using JDBC, check URL:
http://www.oracle.com/technetwork/java/javase/jdbc/index.html

To learn how to embed SQL in Java, check URL:
http://tinman.cs.gsu.edu/~raj/books/Oracle9-chapter-6.pdf

To download IDE for Oracle browsing, check this URL:
http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html

An embedded SQL sample application using Java is provided in class WebCT.

Problem Description: CRCDB is shared by the entire class. It includes two tables, CLINICAL and MUTATION, which were produced by simplifing two original data sets downloaded from cBioPortal for Cancer Genomics (www.cbioportal.org) for this project. Note that these tables are normalized to be at least 1NF (domain should be atomic). CLINCAL has clinical information of 627 subjects (cancer patients). MUTATION include mutation information for these subjects. The overall goal of this project is to examine if there is any connectin between gene mutation and cancer patient's survival prognosis. The correct way of doing the analysis is using an estalised method, called Kaplan-Meier estimator, but in this course we are exmaining if use of Information Gain can also achieve a similar outcome. The first step is creating a table that helps you carry out our planned Information Gain analysis

(a)   By accessing the source tables in Oracle, create a table called IG_READY in your local MySQL database of the format given below. You are basically modifying CLINICAL by adding 10 additional columns labelled as APC, TP53, KRAS, PIK3CA, PTEN, ATM, MUC4, SMAD4, SYNE1, and FBXW7. Here each gene column value is set to 1 if the subject has a mutation in that particular gene or 0 if no such mutation; you will find that from the MUTATION table. When testing the presence or absence of the mutation, do not set to 1 if the mutation type is "silent" (as they are known having no impact). One question is what if a persone have same gene mutation more than once. It will set to 1 as long as the gene has at least one mutation. For Status, LIVING is set to 1 and DECEASED 0.

Patient Id	APC	TP53	KRAS	PIK3CA	PTEN	ATM	MUC4	SMAD4	SYNE1	FBXW7	Status
TCGA-4N-A93T											1
TCGA-4T-AA8H											1
TCGA-5M-AAT4											0
TCGA-5M-AAT6											0
TCGA-5M-AATE											1

(b)   To show that you created IG_READY successfully, you produce a report of the following format in which the value under each gene name is the count of 1's in IG_READY. That is, if 403 is entered for APC, that means 403 subjects out of 627 have APC mutation. Note that during Part II and any subsequent, you will be using this count feature again and again, and you could develop a small module that does count 1's for a given column.

APC	TP53	KRAS	PIK3CA	PTEN	ATM	MUC4	SMAD4	SYNE1	FBXW7	Status

Choice of Programming Language: You are free to choose any programming language (Java, C++, php, etc.) to do the exercise as long as you can embed SQL statements.

Report Format: 1. Show your source code. 2. Show your answer for (b).

The penalty for a late report is **20% off per day**. **Submission cut-off** is 4/6/2017.