FLEXIBLE QUERY ANSWERING SYSTEM WITH HEART DISEASE DATASET

(TERM PROJECT SUMMER-II 2016)

Project Team:

1. Jai Kiran Duvvu
2. Bala Aditya Devineni
3. Pranathi Reddy

Introduction:

Flexible Query answering system tries to suffice the problem of failing query problem using Muslea’s algorithm. In the project we used the numerical data set from UCI archive on Heart diseases which has 14 attributes and we chose a query with 5 attributes which does not return any data from the given dataset.

Link for the data set: <http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/>

WEKA (Brief Description):

The Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. The original non-Java version of Weka was a TCL/TK front-end to (mostly third-party) modeling algorithms implemented in other programming languages, plus data preprocessing utilities in C, and a Makefile-based system for running machine learning experiments. This original version was primarily designed as a tool for analyzing data from agricultural domains but the more recent fully Java-based version, is now used in many different application areas, in particular for educational purposes and research.

Advantages of Weka include:

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using Weka. Another important area that is currently not covered by the algorithms included in the Weka distribution is sequence modeling.

Implementation of the Muslea’s algorithm (Project implementation):

Initially we are reading the query given by the user and then splitting them in to different parts (attributes specified, operator symbol, and the values used i.e. the threshold specified for each attribute in the query) then putting them in to the corresponding arrays.

Extracting the domain knowledge:

In this step we look at the constraints given in the query and create duplicate data set for each constraint in which the constraint attribute value is given a binary value depending on whether the constraint is satisfied or not in the original dataset. Then generate rules by using J48 class of the WEKA library and generate a set of decision rules.

Finding the most useful rule:

From the decision rules learnt from the step 1 the most useful rules are derived using the nearest neighbor technique. Here we used the distance function for continuous data set, maximum and minimum vales of each attribute to find the refiner statement. In this we are generating the arff files corresponding to each attribute which are specified in the query along with ‘Yes / NO’ for each attribute by comparing the rest of the query attributes.

Relaxing the failing query:

The implementation creates a relaxed query that contains only constraints on attributes that appear both in the user input query and the duplicate datasets generated in the previous steps, for each of these constraints the relaxed query uses the less constraining value of those in the user input query and the data set generated. If the attributes doesn’t have any values in common (or does not have any intersecting values) then those attributes are ignored while finding the relaxed query.

Steps to execute the project:

* Download the zip file.
* Import the project files into Eclipse.
* Make sure to add WEKA .jar files located in the lib folder. (For the current project we have already added the required jar files so need to add them again.)
* Copy the files dataset.csv, Keywords.txt, Minvalues.txt, Maxvalues.txt which are present in the Files folder to the path specified in the Class ‘Path.java’

(by default it is set to T:\\summer2\\kbs\\Project\\FQAS\\Files\\).

* As we have developed a web application using servlets we need to add servlet – api.jar to this project. This can be done by
* Right click the project
* Select Properties and in that select Java Build Path and select Libraries tab
* Click on the Add External JARs button and select the ‘servlet-api.jar’ file present in the lib folder of the tomcat server. For convenience this jar file is attached along with the project documents.
* Configure Apache tomcat server and add the project to the server and run on server.

Project input and output

Input Query: age >= 58 ^ sex >0 ^ cp > 2 ^ trestbps >120 ^ chol > 350

Output:

* Relaxed Query: age > 58 ^ sex > 0 ^ trestbps > 120
* Rules:

1. trestbps <= 138 AND cp <= 3 AND cp > 1 ^ age > 58
2. sex > 0 AND trestbps <= 155 ^ age > 58
3. trestbps > 138 ^ age > 58
4. chol <= 300 AND cp > 3 ^ sex > 0
5. chol > 335 ^ sex > 0
6. cp <= 1 AND trestbps <= 138 ^ sex > 0
7. age > 47 AND sex <= 0 AND age <= 66 AND age > 60 ^ trestbps > 120
8. age > 47 AND sex > 0 ^ trestbps > 120
9. chol > 172 AND sex <= 0 AND cp <= 3 AND cp > 2 ^ trestbps > 120
10. chol > 172 AND sex <= 0 ^ trestbps > 120
11. sex > 0 AND chol <= 305 AND cp > 3 AND chol <= 221 ^ trestbps > 120
12. sex > 0 AND cp > 2 AND cp <= 3 ^ trestbps > 120

FilesGenerated (In the same folder where we have placed the Input files):

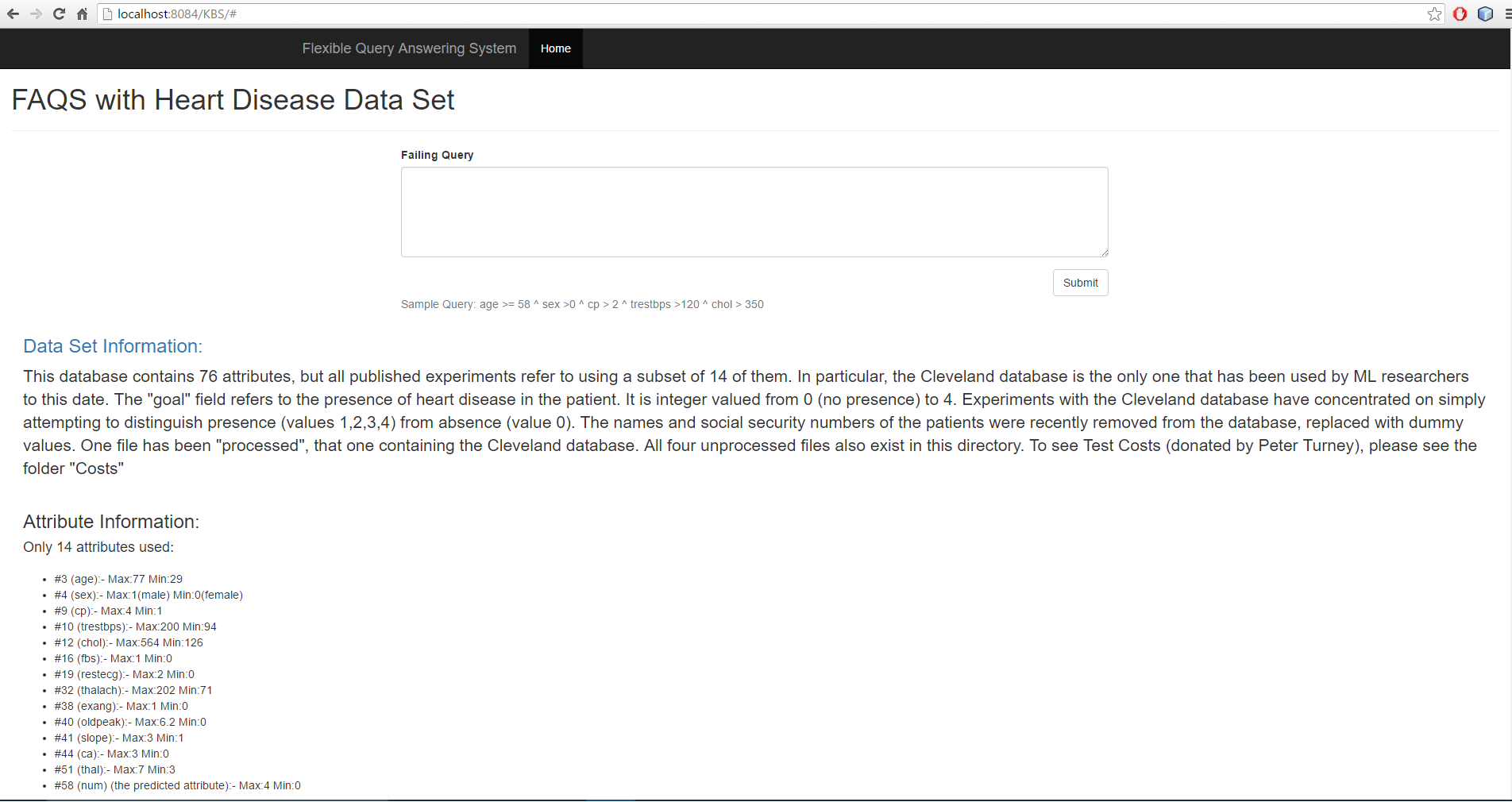
* age.arff and age excel file
* sex.arff and sex excel file.
* cp.arff and cp excel file.
* trestbps.arff and trestbps excel file
* chol.arff and chol excel file

Sample data from age excel file

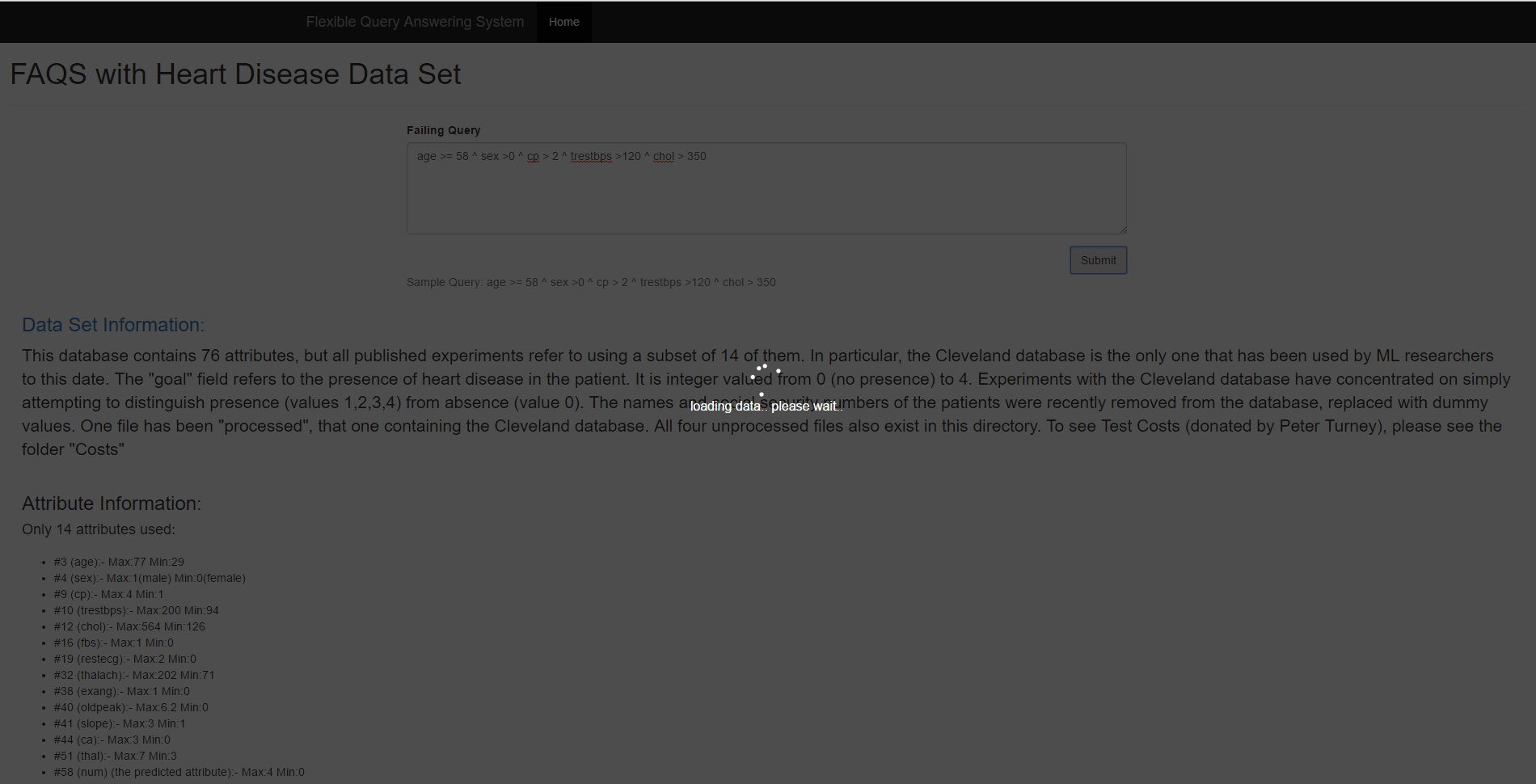
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| age | sex | cp | trestbps | chol |
| YES | 1 | 1 | 145 | 233 |
| YES | 1 | 4 | 160 | 286 |
| YES | 1 | 4 | 120 | 229 |

Screen shots:

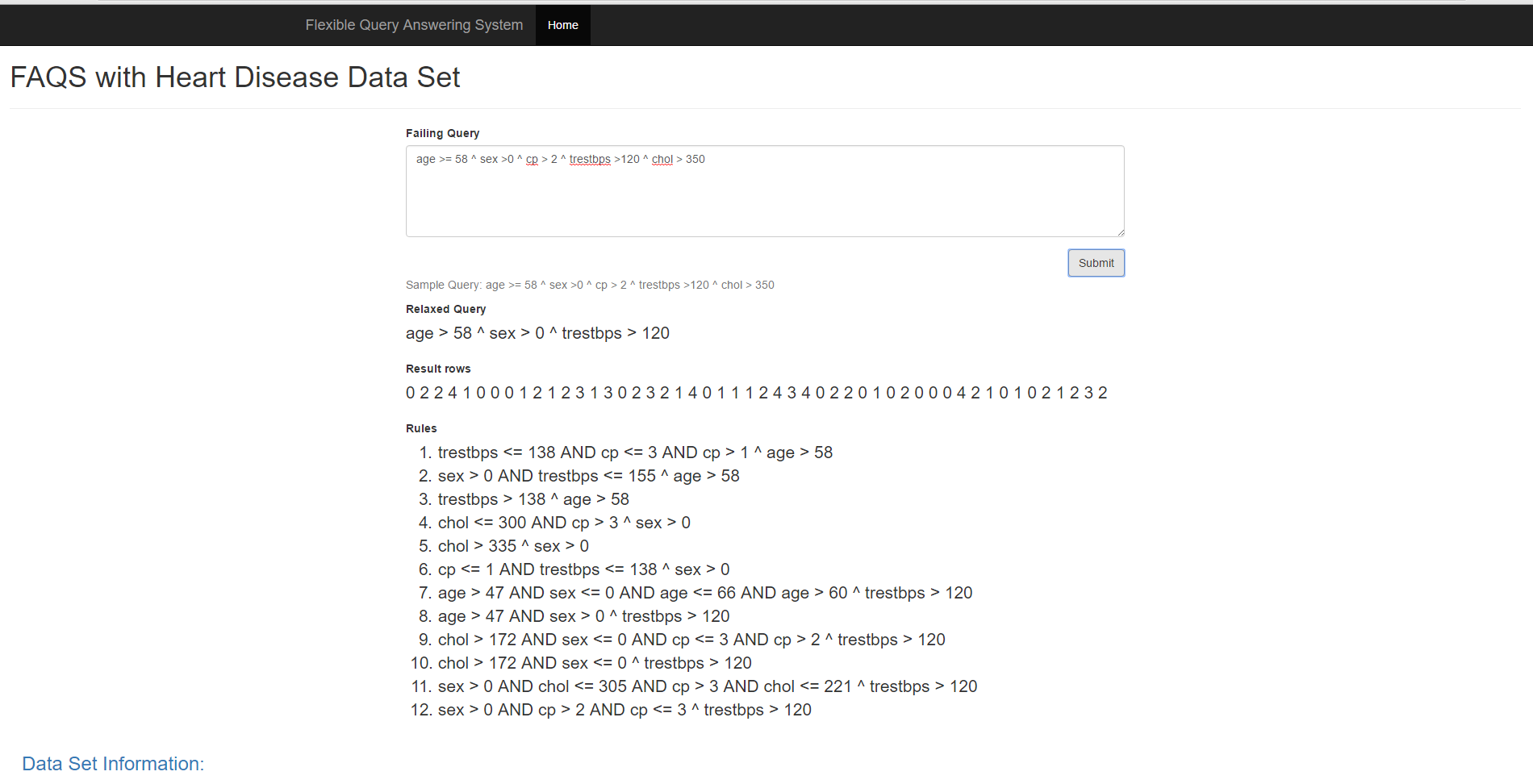
Home Screen: Enter a failing query in the text box provided and click on Submit button.



Loading: Please wait while the results are being extracted



Results Screen:



**References:**

Machine Learning for Online Query Relaxation by Ion Muslea (SRI International) .

Action Rules Mining, A. Dardzinska

<http://en.wikipedia.org/wiki/Weka_%28machine_learning%29>