# Motivation

CS7720 Data Mining requires a project worth 30% of the final grade (Saunders, Data Mining Syllabus, 2015). I had completed designing a database in a previous course (CS7700 Advanced Database Systems) so it only feels natural to mine that database.

# General Description of the Chosen Problem

Data mining is a highly involved process. In *blah* the data mining, or knowledge discovery, process is broken down into seven steps (Han, Kamber, & Pei, 2011). I will attempt to complete these steps. These steps are:

## Data cleaning

This step involves the fixing or removal of bad data. It can either be a manual or automated process. Bad data can be the result of malfunctioning equipment, user error, or other problems. I have already begun this process.

|  |  |
| --- | --- |
| Name | Type |
| 31 Oct 1889 | state |
| Ireland | state |

In the above table, *31 Oct 1889* is determined to be a state. This is most likely because a date was entered in as a place name. Also, *Ireland* is determined to be a state. This is because the algorithm reading in the data assumes the last value in a place string (e.g. City, State) is a state, which isn’t the case, for example, Dublin, Ireland.

These mistakes, and others, I have been fixing manually. Some data I will try to clean automatically. For instance, some data, such as births or deaths, may be missing. I need both to determine the age someone lived to. Age will be part of my hypothesis questions. Therefor I could assume everyone born before 1930 and that I do not have either a birthdate or death date for lived so many (perhaps fifty?) years. Furthermore, if I have neither birth nor death date, and I have either the birth of a child or a marriage date, I could assume they were a certain age (perhaps 25?) at their marriage or first child’s birth, which would not only give me the birth year (using 25, then 25 years prior), using the same assumed age (fifty in my previous example) I would have a death date (in this case, 25 years later). These methods could possibly be compounded to estimate even more birth and death dates.

## Data integration

Here multiple sources are combined. While some data mining application have a huge multitude of sources, I have few. I have my family tree (hendrixfamily.fte.GED) and a major source for my family tree (thurmer.ged), which I found on the Internet a while back, but is no longer available.

## Date selection

Depending on what my hypothesis question is, I may require different data. If I only care about the age of women in the 1800s, I will only select women who were born and died between 1800 and 1900.

## Data transformation

I haven’t looked at yet, but I don’t think there is a strong standard for GEDCOM (which have the GED extension) files.

## Data mining

## Pattern evaluation

## Knowledge presentation

On the home / news page for the course in Pilot, several project requirements and thoughts were listed (Saunders, News - Project Requirements and Thoughts, 2015). The following is a subset of them, with responses:

## Which answers am I trying to mine or solve?

Probably the easiest question to answer is how long people live based on various criteria.

## What kind of data is being mined? Where does the data originate?

The data is about people. While currently it exists in a single file, the data has originated from interviews with relatives, obituaries, family bibles, Facebook, and various online resources.

## Is you project identifying patterns, associations, and/or correlations?

Most definitely - probably involving age. For instance, who lives longer, men or women? Is there a correlation between age and year born? What about age and location born?

## Does your dataset contain outliers, if so, please identify them.

Yes. There is one case where an individual was married at ten years old. There are also cases where events (such as the birth of a child or marriage) happened before or after the person was alive. In the first case, the data was valid - there is even a note on the source stating as such. In the second case, the data is invalid - no one has a child before they are even born, or years after they die.

## Does the data contain missing values or noisy data?

Yes - there is a lot of missing births and deaths. Also, assuming everyone in the world is related, I don’t have every human who has ever existed in my database.

## Does the dataset contain redundant data? If so, how did you remove duplicated values?

With places, definitely. The source file does not aggregate places, so I have to take care of that as I insert the data. If I integrate hendrixfamily.fte.GED with thurmer.ged I will most definitely have duplicate individuals. If those people have the same name, birth, and death then they will be easy to detect and eliminate or merge. However, there may be cases where duplicate values will have similar names, or birth and death dates that are close but not the same.

## Did you determine the size of your data set?

I have over 1500 individuals in my tree with anywhere from 100 to over 200 places in hendrixfamily.fte.GED alone.

# Technologies to be Used

Most of the software I plan on using mirrors the software I am using on the contract I am working on at work. The main difference between the software I am using at work and the software I will be using on this project is that I may use new versions of the software on this project. Also, I may or may not upgrade the software over the course of working on this project. It depends on if a new version if available, if the newer version has a bug fix or new feature I find necessary or desirable, and if I feel it is worth the time and effort to actually upgrade.

## Java Tools and Libraries

### Java Development Kit (JDK) 1.8

### Java Server Faces (JSF) 2.2

### PrimeFaces 5.2

PrimeFaces is simply an extension of JSF.

### ojdbc7.jar

This is Oracle’s JDBC driver to connect to their various Oracle databases (“**O**racle **J**ava **D**ata**b**ase **C**onnectivity version **7**”).

### Maven 3.3.3

Maven is a popular build tool written in Java. It is mostly used for Java projects. It automatically collects and downloads dependencies, and can generate documentation through various plugins (which are also automatically downloaded).

### JUnit 4.12

JUnit is a popular testing framework and can be used for integration tests rather than just unit tests. Its greatest usefulness is the prevention of random “main” entry methods throughout production code.

## Server and Database

### GlassFish Server 4.1

This is Oracle’s implementation of a Java EE server.

### Oracle Database 11g Express Edition

## Development Tools

### IntelliJ IDEA 14.1.5 Ultimate Edition Student License

IntelliJ IDEA is an advanced Java IDE by JetBrains. Typically only the Community Edition is free and the Ultimate Edition requires a rather expensive license. However I was able to obtain a student license with my wright.edu email.

### NetBeans IDE 8.0.2

NetBeans is mostly used for JavaDoc generation. Unfortunately, IntelliJ IDEA does not have similar functionality, despite the much higher price tag.

### Oracle SQL Developer 4.1.0.19

### Notepad++ v6.7.8.2

An advanced text editor with a very powerful search and replace feature.

### TortoiseGit 1.8.15.0

A Windows GUI Git client, allowing me to sync my code on my computer.

## Android Tools

Finally, I am using a few development tools on my Android phone for on-the-go developing:

### ForkHub for GitHub

I mostly use this to use GitHub’s issue tracking system (if I get an idea in class, for instance, I’ll create an issue to later look at).

###  SGit

SGit is a free Git client for Android, allowing me to sync my code on my phone.

### Quoda

A text editor with code highlighting. I use the free version.

## The Cloud (GitHub)

To facilitate syncing and provide issue tracking, this project is hosted at <https://github.com/hendrixjoseph/FamilyTree>.

# Deliverables

All deliverables will be submitted before the end of the semester at a time of the instructor’s choosing. Most likely deliverables that are files will be submitted via Pilot.

## Source code

### Java files

### SQL files

### XHTML files

### Cascading Style Sheet (CSS) files

### JavaScript (JS) files (if applicable)

## Maven generated documentation

### Standard Maven documentation

### JavaDoc for the Java code

### VDL Documentation

## Diagrams

### Schema diagrams

### ER Diagrams

## Results / knowledge presentation

## Demonstration of the application

# References

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Waltham, Massachusetts: Elsevier.

Saunders, E. (2015, Fall). Data Mining Syllabus. Dayton, OH: Department of Computer Science and Engineering, Wright State University.

Saunders, E. (2015, Fall). *News - Project Requirements and Thoughts*. Retrieved from Pilot CS-7720-01 - Data Mining: http://pilot.wright.edu