Fall 2023 **CSC 466:** Knowledge Discovery from Data Alexander Dekhtyar

Lab 1: Getting Insight From Data

Due date: Monday, October 2, 2:00pm.

Preface

The core objective of Knowledge Discovery in Data/Data Mining/Machine Learning methods is to provide efficient algorithms for gaining insight from data. CSC 466 primarily studies the methods and the algorithms that enable such insight, and that specifically take this insight above and beyond traditional statistical analysis of data (more about this — later in the course).

However, the true power of KDD/DM/ML methods that we will study in this course is witnessed only when these methods are applied to actually gain insight from the data. As such, in this course, the deliverables for your laboratory assignments will be partitioned into two categories:

1. **KDD Method implementation.** In most labs you will be asked to implement *from scratch* one or more KDD methods for producing a special type of insight from data. This part of the labs is similar to your other CS coursework - you will submit your code, and, sometimes, your tests and/or output.
2. **Insight, a.k.a., data analysis.** For each lab assignment we will provide one or more datasets for your perusal, and will ask you to perform the analysis of these datasets using the methods you implemented. The results of this analysis, i.e., *the insight*, are as important for successful completion of your assignments, as your implementations. Most of the time, you will be asked to submit a *lab report* detailing your analysis, and containing the answers to the questions you are asked to study.

The *insight* portion of your deliverables is something that you may be seeing for the first time in your CS coursework (unless you took DATA 301, where some instructors also do this). It is **not an afterthought** in your lab assignments. Your grade will, in no small part, depend on the results of your analysis, and the writing quality on your report. This lab assignment, and further assignments will include detailed insturctions on how to prepare reports, and we will discuss report writing several times as the course progresses.

Lab Assignment

This introductory lab is designed to both give you some idea of the structure of the labs in the course, and to test your creativity. As part of the completing the lab,

- You will examine a (fairly large) dataset
- You will prepare multiple study questions
- You will write some code to extract and analyze a subset of data from the given dataset
- You will examine the results of your analysis and, where necessary, prepare visualizations
- You will prepare a lab report detailing your questions, your analysis, and the insight you obtained.

In most of the follow-up labs, you will be given questions to study, but for this lab assignment we want **you** to practice asking questions of the data.

This is a *pair programming assignment.* You pick your partner during the September 25 lab session. I **strongly discourage** individual work for this (and other team/pair programming) lab(s), even if you think you can do it all by yourself. Also, this is a **pair programming** assignment, not a "work in teams of two" assignment. **Pair programming** requires joint work on all aspects of the project *without delegating portions of the work to individual* *team members*. For this lab, I want all your work — discussion, software development, analysis of the results, report writing — to be products of joint work.

Students enrolled in the class can pair with other students enrolled in the class. Students on the waitlist can pair with other students on the waitlists. In the cases of ”odd person out” situations, a team of three people can be formed, but that team must (a) ask and answer one additional question, and (b) work as a pair would, without delegation of any work off-line.

Programming language. In the past this was a language-agnostic assignment, but in advance of our efforts to switch to data-driven Python programming, you shall not only use Python for this assignment, but you shall also use specific packages and techniques, when dealing with the data.

- **IDE/Environment.** While you have access to Jupyter Labs server on our system, for this lab, you are allowed to choose whether you want to use Jupyter notebooks for development, or whether you want to develop outside of Jupyter. The specific deliverables will be spelled out for both cases. In future labs, we will be more strict about dictating a single environment (as works for a specific lab), but for this lab, your have a choice.

- **Data management.** You are required to use `pandas` to read and manage your data in memory. You can convert `pandas` data frames into lists or `NumPy` arrays in some cases, but for the most part, you shall use `pandas`.

  In particular, **all data filtering operations** (such as ”find all records about the name 'Alex'”) shall be performed using `pandas` functionality.

- **Data aggregation, data analysis.** Use `pandas` as much as you can for data aggregation (”What's the total number of male children named 'Alex' across Southern States?”). Use `NumPy` for help with numerical/statistical analysis of your data.

- **Data Access.** Avoid looping over elements of a `pandas` data frame or a `NumPy` array. Instead, use the appropriate operation/function/method broadcasting. (If you need to learn what ”broadcasting” is, *now is the best time to learn”.*

- **Data Visualization**. Use `pandas` visualization tools or `matplotlib`. `Pandas` visualization methods are essentially a wrapper around `matplotlib` functionality, but sometimes it is a bit less cumbersome. For the future, I recommend `matplotlib` - it is more versatile. You can add `seaborn` to allow for convenient styling of your visualizations. If you have used other visualization packages (e.g., `PlotLy`, feel free to use them. Some are more advanced than `matplotlib`, so no reason not to use them if you know how to.

  One thing you **should not do** is export data as CSV, load it into a spreadsheet (Google sheet, MS Excel, or any other similar tool), and visualize using the spreadsheet functionality. *Please, construct all visualizations programmatically.*

The Dataset

Dataset Description

For this assignment, we are using a version of the Kaggle Baby Names dataset[1]. There are two data files available to you.

`NationalNames.csv`. This file contains information about the total number of babies with a given name born each year from 1880 to 2014 in the US. The file has five columns separated by commas (the column names can be found in the first line of the file:

[1] https://www.kaggle.com/kaggle/us-baby-names

Column Name	Type	Meaning
Id	Integer	Unique id of a row/line
Name	String	Baby name
Year	Integer	Year of birth
Gender	Enumerated: { "F","M"}	Gender of babies with given name (see note below)
Count	Integer	Number of babies of the given gender who were given the name in the given year

Note. The dataset contains historic data pertaining to names given to US-born children at birth. Throughout most of the period under consideration, the biological sex of the baby at birth was equated with the baby's gender for the purpose of reporting the frequency of the given names (which makes sense, the report is specifically for names of newborn babies). Because of this, in the Baby Names dataset Gender is a binary feature with only two values, "M" (male) and "F" (female). Understand that it really signifies *biological sex at birth.*

For example, the following line from the file:

```
12874,Beatrice,1886,F,192
```

states that in 1886 there were 192 female babies named Beatrice.

StateNames.csv . This file contains state-by-state information about the frequency of baby names in the years 1910 – 2014. The dataset contains six features (columns) – their names are specified in the first line of the file.

Column Name	Type	Meaning
Id	Integer	Unique id of a row/line
Name	String	Baby name
Year	Integer	Year of birth
Gender	Enumerated: { "F","M"}	Gender of babies with given name
State	Enumerated	Two-letter state code
Count	Integer	Number of babies of the given gender born in a given state who were given the name in the given year

For example, the following line from the file:

```
3597745,Patricia,1921,F,NY,392
```

states that in 1921 there were 392 female babies named Patricia.

Dataset Access

The dataset is available in two locations:

Course Web Page. If you are working on your own laptops, you can download a copy of the dataset from the course web page. The URL is

http://www.csc.calpoly.edu/ dekhtyar/466-Fall2023/labs/lab01.html

The page contains the links to the dataset files.

CSL Machines. Because of the size of the dataset, I do not want those of you working on the CSL machines to have to manage your own local copy of the data[2]. As a result, on the CSL machines, a read-only version of the dataset files is available at the following locations:

[2] You are welcome to create your own local copies of any subsets of the data you extracted for your analysis – in fact, I would like to encourage you to do it. It's the main dataset files that I do not want to see 70 copies of on the CSL machines - it creates an unnecessary waste of space, and consumes resources on your CSL accounts.

```
 dekhtyar/www/466-Fall2023/data/NationalNames.csv
dekhtyar/www/466-Fall2023/data/StateNames.csv
```

(note: these are actually the same files as the ones you would be accessing through the web site, as evidenced by the path.) ‘

The Task

Despite having only a few features, the Baby Names dataset contains rich and fascinating data that with some ingenuity can shed light on history and culture of the US, and on the changes the country as a whole, and its individual states have undergone over time.

This lab assignment offers you an opportunity to perform some data exploration[3] and visualization. The specific tasks are described below.

Ask Questions. Each team will formulate three ”research questions”, or areas of investigation. At least one question of the three must use the `NationalNames.csv` data, and at least one question must use the `StateNames.csv` data[4].

A **successful research question** shall have the following components to it:

- **Identification of data of interest.** You have to explicitly identify a subset of the entire dataset that is of interest to you.

  For example, your question may limit the data to a specific set of years (e.g., 1990 - 1999), a specific set of names (e.g., `Alexander` and `Alexandra`, or top 10 male and female names from each year), and, for state data - a specific subset of states (e.g., `Alabama` and `New Jersey`).

  Some questions can be asked of the full dataset, but your analysis (especially in the case of looking at state-level data) may become more complicated.

- **Object of study.** Your research question must specifically address an object of study.

  For example, one may be interested in the growth of the popularity of the name `Pauline` vs. the name `Paulette`. One may also be interested in the change in the gender breakdown of babies named `Dana` over the course of the 20th century. Or one may be interested in the growth of different spelling variants of popular names, such as `Zachary` or `Brittany`.

Note. Please make sure that you are not asking questions that can be answered (in prinicple) with a single simple SQL query. For example a question *”How many children were given the name ”Boris” in 1953?”* is not a valid research question for this assignment. Neither is the question *”What is the most popular boys’ name in the US in the 1990s?”* by itself a valid research question (although finding this information might be part of the analysis you would perform to answer a ”true” research question)

Extract Data. For each research question you come up with, write code that extracts the appropriate data from the dataset you are using. You are welcome to dump the extracted data into a local file[5].

Answer the question. For each research question, determine the computations that need to be performed in order to answer it, and write code performing these computations. Make sure to capture properly all relevant output of your computations.

[3] In statistics there is a term ”exploratory data analysis” (EDA) that may be somewhat appropriate for the intent of this assignment. However, the exact details of what you are are asked to do in this lab are somewhat less strict and more general than what is typically meant by this term in statistics. Hence, we will use more informal terminlogy to reference your activities in this lab.

[4] Any team of three people will work on four questions, out of which at least one must address national data, and at least one — state data.

[5] The assumption is that most of your questions will involve a relatively small subset of data from the datasets you are given.

Analyze and Visualize. For each research question, determine *what insight you gained* from perofrming your analysis, and *what is a good way to visualize your inisght.* Create (either programmatically, or by hand - this is left up to you, although many programming languages have appropriate visualization packages) the visualizations you want, and *write the portion of your report describing the insights you gained.*

Hints

There is a large number of ways to approach this assignment, and I **really-really** want each team to try to perform some analysis that the team members are *genuinely interested in.*

One piece of advice is this: baby naming patterns exist in their historic and cultural context. Therefore, in order to formulate proper questions I encourage you to consult available sources on the World Wide Web[6] to place your questions in a proper historic context.

Here are some subtle (and not so subtle) hints to get you started. Feel free to **NOT READ THEM** if you feel like you have already been inspired.

Names are a fascinating window into the culture, attitudes, and traditions of a nation. Through changes in the absolute and relative frequency of baby names, you can track a lot of interesting things:

- **Demographic shifts.** Can you trace the influxes of immigrants in different parts of the country by looking at shifts in frequency of certain names/types of names?
- **Major events.** Are there any changes in naming patterns that take place around major events (e.g., wars, Great Depression, major changes in law, etc...)?
- **Fads.** Temporary changes in popularity of certain names often can be attributed to a specific pop-culture phenomenon (e.g., is there a correlation between the incidences of the name "Barbie" and the growth in popularity of the famous doll), or to a specific person (are there more boys named "Clark" following the release of "Gone with the Wind").
- **Long-term trends.** Can you quantify the changes in the variability/diversity of baby naming over time? A good example of this is the trends related to the frequency "western" baby names not based on the religious texts (think "John" and "Michael" vs. "Braden" and "Holton")
- **Geographic differences.** Over time, naming trends in certain geographic areas can diverge. Can you detect such divergence? Can you hypothesize what specific societal trends may be responsible for such divergence?

Some of what you may ask may be profound and a reflection of true changes in the society (e.g., what happened to German-sounding names in 1940s and 1950s? Why?). Some questions may be completely frivoulous (e.g., can we rank the states in terms of their propensity to invent spellings of "Brittany"?). Some questions may lie somewhere in between.

Report

Report-writing is as important in this course as programming. The quality of your report - both in terms of content, *and in terms of writing* will be a determining factor in the lab grades you receive.

In this section we outline some general report-writing requirements, as well as the specific requirements for *this lab's report.*

Typesetting. All reports **must be** created using proper word-processing software. If you know LaTeX - great (I recommend using Overleaf - this allows for on-line collaboration between teammates)! If not — MS Word, Google Docs word processor, `oowriter` and other similar word processing solutions can be used.

Submission format. All reports shall be always be submitted in PDF format.

[6] Yes, this **can** including Wikipedia.

Structure. All reports must contain the following information.

1. **Title.** Each report must have an appropriate title. Typically, for you, this title would be something like ”CSC 466 Lab 1 Report”, but you may also show some creativity, and add a subtitle that actually properly describes the contents of your report, e.g., ”CSC 466 Lab 1 Report: Effects of World War II on Trends in Baby Names in the US”).
2. **Authors and Affiliations.** Each report **must list all its authors** and the contact information (Cal Poly email address) of each author.
3. **Abstract.** In most reports, I will ask you to include a short abstract describing the contents of the report. Because Lab 1 is a creative assignment and I expect different teams to ask different questions, *please include an abstract with a short description of your work and results* in Lab 1 report.
4. **Introduction.** A short introduction section discussing your work will be required for most reports, including `Lab 1` report.

In addition to these parts, your `Lab 1` report shall contain the following sections/information:

- **Dataset Description.** A short section describing the datasets you are working with. This section will allow readers not familiar with CSC 466 to understand the contents of your report.
- **Research Questions.** A short section listing all research questions you are asking - supply a brief explanation for each question.
- **Methods.** A short description of the methods you are using to answer each question. Discuss what data you are extracting from the datasets provided to you, and what computations you are performing on this data.
- **Results.** For each research question, your results section shall contain the visualizations and the explanations of the observed results, concentrating on the actual insight from the data that you have obtained.
- **Discussion and Conclusion.** A short summary of your work, and a discussion of any limitations, bugs/errors experiences, problems/challenges encountered and so on.
- **Bibliography.** As appropriate. If your report needs to cite any outside sources, make sure you do it properly and include the sources in the bibliography (and yes, Wikipedia counts as a proper source here.)

Submission Instructions.

Submit the following.

Readme File. The `README` file shall be a plain text file containing, at the very least, the following information:

- Names and email addresses of all students in the group.
- Description of what you are submitted: direct Python code or Jupyter notebook/notebooks.
- Instructions for how to run your code.
- List of submitted programs/files, with a short description of the intent of each program.
- Mentions of any errors that might be experienced (e.g., if your error-checking is not up to the standards).

Code. Submit all code you developed for this lab. Each Python file you submit must have a header comment with the identification of the course, (CSC 466), quarter (Fall 2023), assignment (Lab 1) and the names and email addresses of all students on the team (regardless of who specifically developed that file). The header comment shall also contain a short description of the purpose of the file[7].

Please note, if you are using Jupyter notebooks (or any other notebooks for that matter), you can submit the notebooks as your code. Each notebook shall contain a Markdown section at the top with the same content as the header comment described above.

Report. Submit your report as a PDF file named `Lab1-Report.pdf`.

Submission instructions. You can submit all code in a single archive (zip or tar.gz), but please submit your `README` file and your report outside of the main archive. Use `handin` for submission. The `handin` command - when you are logged on one of the `unixN` CSL servers depends on your section.

Log into any of our unix servers (`unix1.csc.calpoly.edu` for example), transfer the deliverable archive there (or assemble it, if you worked on the servers), and execute the following command:

```
$ handin dekhtyar lab01-466 <files>
```

GOOD LUCK!

[7] In future labs, there will be specific file naming conventions given to you, so I will not need as much to rely on your self-reporting in order to properly grade your work, but for this lab, I am not setting any limitations on how you do your work, so self-reporting is important!