CSC 369: Distributed Computing
Spring 2020
Alex Dekhtyar
(Optional!) Final Mini-Project

Due: Sunday, June 14, 12:00pm (noon)

Submission instructions. Submit your code and report (in PDF format) using the following command:

```
$ handin dekhtyar project369 <files>
```

on `unixN.csc.calpoly.edu` machines.

Make sure your data is uploaded to HDFS or MongoDBand your code accesses it properly.

Data submission instructions. For **hadoop/spark** projects I created an hdfs directory `/projectdata` and opened it for reading and writing to everyone. If you are working on a mini project, place your datasets in the `/projectdata/<loginId>` directory, where `<loginId>` is your ambari-head account. For example, my data would be located in `/projectdata/dekhtyar`.

This ensures that I can run your hadoop and spark programs.

For **MongoDB** submissions, create all necessary collections in your own database (i.e., your `<loginId>` database). I will be able to access these collections when I run your code.

Assignment Specification

Rules and Impact. This is an optional assignment that takes place in lieu of the in-person final exam. This assignment is worth 10% of the course grade. Please note, that you have the option of NOT COMPLETING this project, in which case your grade on the remaining 90% of the course will be scaled (this is known as ***"keep my current grade"*** option). So, for the project to affect your grade positively, you need to score a higher score on it than your percent achievement in the remainder of the course. Therefore, the impact of this project is marginal, but not invisible.

Example. If your score is 60% on the 90% of the remaining course, and you earn 100/100 on the mini project (i.e., full 10% credit), then, your final score will be:

100*(0.6*0.9 + 1.0*0.1) = 64%

an elevation of 4 percentage points that may make a difference between a C and a D letter grade.

Specification.

Demonstrate mastery of one of the technology covered in the class: MongoDB, Hadoop MapReduce, or Spark/PySpark, by telling an interesting and relevant story with data using one of the abovementioned technologies (you can use more than one, but you must use at least one to do your main data processing).

This is a purposefully open-ended assignment, however, it does have certain conditions and expectations.

Data. Your dataset must be societally relevant and must be something that was not used in class. It also must be legally obtained - either data that is available publicly, or data that data the collection of which did not violate any use terms or policies. For datasets that fall in grey zones (publicly released *leaked,* or something of that sort), consult me first. In some cases, if the dataset has important societal impact, it will be deemed acceptable. In other cases (stolen credit card numbers), it may be deemed unacceptable.

The **relevance** aspect of the dataset is judged by the insight the dataset provides into today's society (or the society over time). It is largely a judgement call - if you have questions or concerns - please consult with me.

There is a variety of places where you can find data - starting with government sites (all US states have data portals, US Census bureau is a well known location for a lot of feature rich data, etc..), Kaggle, and a few other dataset aggregator sites.

A list of datasets can be found in this file (I will also make this link available on the web site and on Slack). Pretty much all the datasets references there easily pass the "social relevance" criterion.

The dataset must also have non-trivial size. That is, there should be some reason for you to want to use distributed computing to process the data. Again - if in doubt, contact me, I will examine the dataset you are planning to use and will let you know.

You can use ***multiple datasets.*** Some of the more interesting data-driven stories take place when different data is connected to each other.

Questions. You must ask one or more clear questions that your data can answer. The question/questions can be broad (e.g., *"What is the impact of education on the number Amazon.com purchases?")* or very detailed (e.g., *"Which zip codes in the USA have the highest per capita consumption of strawberries?"*). There may be followup questions (e.g., "*Can we explain why these specific zip codes consume so many strawberries?").*

You do not have to ask questions that require statistical analysis or data mining algorithms that you do not know how to implement. But the question/questions you ask should attempt to tell a story, and or, provide a meaningful and insightful view of the data.

Distributed Computing. Write data cleaning, data transformation, and data analysis code using one of the three platforms we studied in this course: MongoDB, Hadoop, or PySpark. Your code should do **all its processing** in the platform/platforms of your choice. E.g., extracting a data collection from MongoDB into a Python NumPy array and then finding the averages of the columns is **wrong.** You must write MongoDB code that finds the averages. The exception is data transfer - if you need to convert your data from one format to another (e.g., create a dataset in MongoDB, place it into HDFS for further processing by Spark) - you perform some data transfer activities in the host programming language, or with linux scripts.

Each step of your processes has to be reflected in a separate piece of code that is clearly delineated. (Multiple pieces of code can be placed in the same program, but they must be clearly identified with comments, and with the program structure).

Reporting. Your processing will yield output. The output should be visualized (this does not necessarily mean creating a graph or a map of the data, sometimes, a contingency table, or a simple table of results is sufficient) in a clear and concise way. Visualization/output can be handled by code working outside of your distributed computing frameworks. Your output, as a rule should be small, and therefore, can be processed using regular sequential programs with no distributed computing features.

Final report. Create a report of your project. The report shall contain the following information (sections):

1. **Introduction**: describe why you chose the specific project, introduce data you used, preview results.
2. **Dataset:** provide an overview of the provenance and structure of all the data that you are using for the project
3. **Questions.** Articulate your research questions.

4. **Problem decomposition.** Describe how you approached answering the questions. What steps did you take? How did you decompose your problems into appropriate code in the framework of your choice?
5. **Results.** Present the results of your analysis, tell your story, discuss the effects.
6. **Conclusion.** Make any parting remarks.
7. **appendices (if needed).** Include any relevant information that for one reason or another was inconvenient to place in the body of the report (e.g., your report can contain 10 zip codes with the highest strawberry consumption, but you can place the top 100 zipcodes in a table in the appendix).

Some of the instructions are purposefully vague. This is intended to be something to the tune of:

- 2 hours prep time (thinking about the problems, searching for data, revising what you can do)
- 1-2 hours setup time (data munging, data cleaning, and so on)
- 3-4 hours coding (writing code for your actual problem)
- 1-2 hours visualization
- 1-2 hours report writing

This might sound like a lot for one week that has other exams in it, but with the deadline on Sunday, you can spread this into multiple days… I would select the dataset and ask questions (and perhaps even write Dataset and Questions portions of your report) early. Everything else can be done in one-two settings. (Recall, this is replacing the final exam, and you always can opt to take the final grade. This project is an outlet for people who want to raise their score a little bit).

Final Comments. (a) No plagiarism, please. Datasets are available everywhere, so is code to use them. While Hadoop and Spark code is somewhat harder to find than regular Python/Java code, it is still out there. It should take you less time to solve the problem than to search for the solutions elsewhere. **(b)** If you have questions, consult me. Aspects of this assignment were made purposefully broad to give you an opportunity to think about what you want to do. But I am happy to discuss any proposed plan of action.

GOOD LUCK!