Final Project

DS730

In the final project you will be working with all of the technologies you have learned in this course. Problem 1 must be solved using one of the high performance tools learned in this class[[1]](#footnote-1): Hadoop MapReduce, Pig, Hive or Spark. You do not need to choose 1 tool for all problems nor do you need to use all tools. You can mix and match them for problem 1 in any way that you want. Problem 2 must be solved using Java or Python threads. **Note that a solution written without using a tool learned in this course will not earn any points[[2]](#footnote-2).** If you are unsure if the way you solved them meets the requirements of this project, ask first before submitting your solution.

If there is a tie for any of the questions, you should print out all of the correct answers.

1. (63 pts) For the first problem, you will be reading in from two weather almanac files. You are encouraged to look at the files to see what the data looks like. Once you have a good grasp on what is contained in the files, you are to answer the following questions.  
     
   One of the objectives for the final project is assessing your ability to select and use a tool for a specific problem. With that said, we won't provide a lot of specific "code-related" help for this question. When you were learning the tools in previous activities and projects, we would provide as much help as we could so you could learn the tool. However, we want to assess your ability to select and use the tools on the final project with minimal assistance. We will clarify problems but we won't help out much with the actual code. If we have to help out with writing code on the final project, then that takes away from the objective of assessing your ability to use the correct tool.  
     
   Some of these problems may have a subjective meaning. If I ask which day was the hottest, that could be interpreted in different ways. Do I want to know which day had the highest temperature or do I want to know which day was the hottest with respect to the average? For example, assume one day was 94 the entire day and another day was in the 70’s for most of the day and then jumped to 95 for a short amount of time before falling back to the 70’s. Which day would be considered *hotter*? I will try and explain exactly what I am looking for. If a question feels ambiguous, ask me before attempting it.   
     
   You should be aware of bogus values in the input. This is almost always the case in real datasets and you will almost always have to clean your data before you analyze it. All cleaning must be done using the tools we learned in this course. For instance, some of the values are set to -9999 if no value was recorded (or the cell is just empty). Only the bogus/missing cell should be removed from your calculations. However, the rest of the row should be included if a cell has a bogus/missing value. For example, in the Oshkosh weather file, on January 26th, 2008, the temperature was recorded as -9999 but there is a valid wind speed. The temperature should be ignored but the wind speed should be included in your answer. Make sure to look at the data before beginning so you can figure out what values are invalid. Ask me if you are not sure if a value is valid or not. Note that a Calm wind speed is equivalent to 0mph. A Calm direction is equivalent to N/A.  
     
   You must assume the data is stored in the HDFS folder of: **/user/maria\_dev/final/Oshkosh/** for the Oshkosh csv file and /**user/maria\_dev/final/IowaCity/** for the Iowa City csv file.  
     
   You are to solve each of these problems and provide the answers to all of the questions in a file called **prob1output.txt**. For each of these problems, make sure that you create a separate script or program for each problem. Each script/program should work on it’s own independent of any other script. Also be sure to create an appropriately named file containing your solution. Your solution should be stored in *prob1W.XYZ* where W is equal to the actual problem letter and XYZ is the appropriate extension for the tool you are using. For example, if you solve 1a with hive, your program will be stored in prob1a.q. If you solve 1b with pig, your program will be stored in prob1b.pig.  
     
   Your code does not need to output the exact answer for each of the questions (a-f) in this problem. It is fine to do a *little bit* of extra manual work to find the answer. For example, for part (a), you do not need to output “cold is more common” or “hot is more common.” Rather, you could output the number of “cold” days along with the number of “hot” days and manually look at the output to determine the answer. However, the amount of manual work you do ought to be limited. An incorrect solution for number 1 would be to simply print out the entire weather file and then claim that you manually checked every row and counted the number of cold days and counted the number of hot days. If you are unsure if your solution does too much manual work, just ask.

**CLEANUP:**

* **Remove any rows where time = “No hourly…”**
* **Wind Speed ‘Calm’ = 0.0**
* **Gust Speed “-“ = 0.0**
* **Humidity “N/A” = -1**
* **Precip “N/A” = -1**
* **VisibilityMPH**
  1. In Oshkosh, which is more common: days where the temperature was really cold (-10 or lower) at any point during the day or days where the temperature was hot (95 or higher) at any point during the day? –
     1. **Hive or scala –**
        1. **Build Date with year – Month – Day.**
        2. **Group by date – find max temp and min temp**
        3. **Filter by temp > -10 or < 95**
        4. **Categorize by temp**
        5. **Group by category – count each.**
  2. When I moved from Wisconsin to Iowa for school, the summers and winters seemed similar but the spring and autumn seemed much more tolerable. For this problem, we will be using meteorological seasons:  
     Winter - Dec, Jan, Feb  
     Spring - Mar, Apr, May  
     Summer - Jun, Jul, Aug  
     Fall - Sep, Oct, Nov  
     Compute the average temperature (sum all temperatures in the time period and divide by the number of readings) for each season for Oshkosh and Iowa City. What is the difference in average temperatures for each season for Oshkosh vs Iowa City?
     1. **Scala** 
        1. **Categorize months by season**
        2. **Aggregate by season – average temp**
        3. **Calculate diff by season of average temp**
        4. **Output diff by season**
  3. For Oshkosh, what 7 day period was the hottest? By hottest I mean, the average temperature of all readings from 12:00:00am on day K to 11:59:59pm on day K+6. For example, April 30th, 2006 to May 6th, 2006 is a 7 day period. December 29, 2005 to January 4, 2006 is a 7 day period. Look at all 7 day periods and determine which one is the hottest.
     1. **Scala** 
        1. **Reduce variables**
        2. **Create date from dates and times – convert to unix timestamp**
        3. **Determine the size of each 7 day period**
  4. Solve this problem for Oshkosh only. For each day in the input file (e.g. February 1, 2004, May 11, 2010, January 29, 2007), determine the coldest time for that day. The coldest time for any given day is defined as the hour(s) that has/have the coldest average. For example, a day may have had two readings during the 4am hour, one at 4:15am and one at 4:45am. The temperatures may have been 10.5 and 15.3. The average for 4am is 12.9. The 5am hour for that day may have had two readings at 5:14am and 5:35am and those readings were 11.3 and 11.5. The average for 5am is 11.4. 5am is thus considered colder. If multiple hours have the same coldest average temperature on any given day, then those hours that have the coldest average are all considered the coldest for that day. Once you have determined the coldest hour for each day, return the hour that has the highest frequency. This is not a windowing problem. You only need to consider the 24 “hours” of the day, i.e. 12am, 1am, 2am, etc.
     1. **Hive**
        1. **Set View with constructed date, only hour + AM/PM, temperature**
        2. **Aggregate by date + hour – find average**
        3. **Rank averages by date + hour**
        4. **Aggregate by date – find highest ranks.**
  5. Which city had a time period of 24 hours or less that saw the largest temperature difference? Report the city, the temperature difference and the minimum amount of time it took to obtain that difference. Do not only consider whole days for this problem. The largest temperature difference may have been from 3pm on a Tuesday to 3pm on a Wednesday. The largest temperature difference could have been from 11:07am on a Tuesday to 4:03am on a Wednesday. Or the largest difference could have been from 3:06pm on a Wednesday to 7:56pm on that same Wednesday. For a concrete example, consider Iowa City on January 1, 2000 at 2:53pm through January 2, 2000 at 2:53pm. The maximum temperature in that 24 hour span was 54 and the minimum temperature in that 24 hour span was 36. Therefore, in that 24 hour span, the largest temperature difference was 18 degrees. If this were the final answer, you would output “Iowa City”, “18 degrees” and January 2, 2000 3:53am to January 2, 2000 10:53am.
     1. **Scala**
        1. **Load IowaCity and Oshkosh Data**
        2. **Reduce variables and filter as needed. (date, time, temp)**
        3. **Create full date + time stamp. + temp**
        4. **.withcolumn(“diff\_temp,max(ds(“col”).over(windowSpec) – min(ds(“col”).over(windowSpec**
  6. As a runner, I want to know when is the best time and place to run. For each month, provide the hour (e.g. 7am, 5pm, etc) and city that is the best time to run. The best time and place to run will be defined as the time where the temperature is as close to 50 as possible. For each month, you are averaging all temperatures with the same city and same hour and checking how far that average is from 50 degrees. If there is a tie, a tiebreaker will be the least windy hour (i.e. the windspeed column) on average. If there is still a tie, both hours and cities are reported.

1. (42 pts) – See Java Thread
2. (45 pts) The goal of the last problem is to answer questions about a large dataset that you are interested in. Amazon has many large datasets available at <https://aws.amazon.com/public-datasets/>. Here is another link to many large datasets:   
     
   <http://www.datasciencecentral.com/profiles/blogs/big-data-sets-available-for-free>.   
     
   You do not need to find all of your data from one place. You can combine data from multiple sources if need be. Find something that interests you. Once you have found a dataset that interests you, create at least 4 interesting questions about that dataset and answer them. The questions themselves are entirely up to you but they should be somewhat involved. Since the data might not be in a perfect format, you will likely have to clean and prepare your data before you can analyze it.

For example, determining the most common wind direction in the weather file is quite trivial to do and would not be an acceptable question. A good example from the baseball document: one could try and come up with a query that would accurately predict the most valuable player for a given season. One could combine the weather and baseball datasets to determine the likelihood a player will hit a homerun given certain weather conditions (wind, temperature, humidity, location, etc). Because I do not know if everyone has taken the statistics course, I do not know your statistics background. Therefore, your use of statistics in your questions and answers is completely up to you. I also can’t assume you have had the visualization course so feel free to submit your answer to this question in any reasonable format.  
  
If you can find a dataset that is many gigabytes, you’ll be able to see the power of cloud computing on AWS if you use S3 and an EMR cluster. Assuming you found a dataset that is many gigabytes in size, transferring it from your local computer to S3 might take a long time. However, using wget on an EC2 server and then transferring it to S3 from your EC2 instance will be much faster. As a reminder from a previous activity, the following command uploads all contents of the current folder to some bucket in S3 (see task 7 of activity 2 for how to install awscli to run this command):  
  
**aws s3 sync . s3://name-of-your-bucket-here/folderName**  
To give you an idea of cost when running mapreduce on a large file, I’ve tested a file on the order of 10GB. I created 17 instances (1 master and 16 core nodes). With 17 instances, I ran a rather simple Pig script and it cost roughly $6. I ran a similar program in the past with 10 instances and noticed a considerable speedup. That particular running cost about $4. Choose your instance size wisely. This is not something you want to run more than a couple of times, especially if you are using large instances. Be aware that our AWS Educate accounts are limited in the number of instances that can be created at any one time. The sum of the EC2 instances and the EMR instances must be below 20.

The following are FAQ about this question:

1. **How big is big enough?** This problem is a pseudo mini capstone project. You have been forced into learning different areas without any choice in what the datasets have been. For this problem, you are encouraged to use data that you really care about and answer questions that are of interest to you. In an ideal world, the data you choose would be very large (e.g. at least a few GBs). If you choose data that is 10GB or higher, you will see the power of the tools that we have learned in this course. Instead of taking hours for your local machine to process some dataset, you can leverage the power of the tools learned in this course to process data in minutes.

Therefore, I strongly encourage using a dataset that is at least several GBs in size. However, if you absolutely love some dataset that is smaller, you can use it. If you do, you will miss out on seeing the true power of high performance computing… but you won’t lose points for it as long as your code uses the tools we learned in this course.

1. **Do I have to use AWS to solve these problems?** The answer is really the same as the previous question. You could solve these using some virtual machine or using a setup on your local machine but that kind of defeats the purpose of this problem. Since many of you don’t have access to a large cluster of machines locally, using a cloud solution is the next best option. Seeing how to spin up a large cluster to solve some problems on a huge dataset is a great skill to have. If you work at a business that cannot afford a huge cluster of machines locally, you can show them that you can spin up clusters in the cloud for a fraction of the cost. You can utilize the power of high performance computing without having to set it up or administer a cluster locally. The answer is the same as the last one. I strongly encourage using AWS but it is not a requirement.
2. **How do I know if my questions are difficult enough?** This is fairly subjective and we don’t really want to be pre-approving a bunch of questions. Use your own judgment here. Did it take you less than a couple of minutes to come up with the solution to the problem? If so, it was probably too easy. Has it been two hours and you still don’t have a solution; then it was likely too difficult. Most people have created very nice questions/answers for this problem and only a handful have pushed the limit of what is acceptable[[3]](#footnote-3). If you are really concerned about it, post your questions to piazza and we’ll let you know whether or not they are acceptable.
3. **Does the code/naming/answers/questions/output need to be in any specific format?** No. You can use any reasonable format for any of these things.
4. **Can I use my answer to problem 3 as part of a portfolio?** Absolutely. Not only that, but I encourage you to do this. This is especially true if you are using a really big dataset and can show your results were obtained quickly using the tools learned in this course.

1. The only exception is that threading cannot be used for problem 1. [↑](#footnote-ref-1)
2. You can double check your work using Excel or whatever other tools you want. But your solution has to use the tools we learned in this course. [↑](#footnote-ref-2)
3. Some examples of things that are generally too easy: any kind of counting that doesn’t involve doing much else. For example, in our weather file, how many rows were warmer than 90 degrees would be too easy. Any kind of aggregate function that stands alone without doing much else. For example, what was the average temperature in the weather file. However, a combination of these starts becoming acceptable. For example, of the days that reached 90 degrees, what was the average temperature for those days? [↑](#footnote-ref-3)