Big Data Analytics

Stony Brook University  
CSE545 - Spring 2022

**Assignment 2**

**Assigned: 3/19/2022;   Due: 4/2/2022 11:59pm**

[**Overview**](https://docs.google.com/document/u/3/d/e/2PACX-1vTOFalQVlbBzQjH_NNcewvpfu3nBwoq4hT3ybvmbbqh2aqkXKqPFDAVbgDg_uWMNnGAwjWHSoTWe4pu/pub#h.k7js7on60mn7)

[**Part I. Sparkifying Income Descriptives (40 points)**](https://docs.google.com/document/u/3/d/e/2PACX-1vTOFalQVlbBzQjH_NNcewvpfu3nBwoq4hT3ybvmbbqh2aqkXKqPFDAVbgDg_uWMNnGAwjWHSoTWe4pu/pub#h.pvn219uxc1h4)

[**Part II. Similar Hospital Search (60 points)**](https://docs.google.com/document/u/3/d/e/2PACX-1vTOFalQVlbBzQjH_NNcewvpfu3nBwoq4hT3ybvmbbqh2aqkXKqPFDAVbgDg_uWMNnGAwjWHSoTWe4pu/pub#h.mfqja726bk8j)

[**Submission**](https://docs.google.com/document/u/3/d/e/2PACX-1vTOFalQVlbBzQjH_NNcewvpfu3nBwoq4hT3ybvmbbqh2aqkXKqPFDAVbgDg_uWMNnGAwjWHSoTWe4pu/pub#h.udpy7l5v81q)

**Overview**

|  |
| --- |
| **Objectives:**   * To gain experience programming in Spark * To understand differences in implementations using Spark versus MapReduce or standard streaming. * To better understand LSH through implementing portions of the algorithm * Gain further experience with Spark. * Gain further experience with data preprocessing. * Explore a different modality of data: structured data.   **Requirements.** You must use Python version 3.7 or later and PySpark 3.0 or later.  You must also keep the data in Spark RDDs – you may not use Spark DataFrames (mastering spark RDDs makes using dataframes efficiently easy; people who learn dataframes without first learning RDDs often implement very inefficient approaches).  **Python Libraries.**  Acceptable data science, machine learning, or statistics libraries are listed below. Other data science, machine learning, or statistics related libraries are prohibited unless listed below --  **ask if unsure.** The intention is for you to implement the algorithms we have gone over and problem solve in order to best understand the concepts of this course and their practical application.  Additional approved libraries that are not in the template will be listed here (if any):  import random  import numpy as np #for numeric algebra and arrays  import math  import hashlib  import csv |
| **Copying code from other students, online or other resources is prohibited** and will result in at least a zero on the assignment and report to graduate program director with possibility for more consequences. Please see syllabus for additional policies.  A word to the wise: As is tradition in CSE545 at SBU, parts of this assignment are completely novel, never given before. |

**Part I. Sparkifying Income Descriptives (40 points)**

|  |
| --- |
| Here, you will repeat objectives from assignment 1 but using Spark rather than Streaming/MapReduce.    **Data.**  You will use the same data as assignment 1: Two versions of the data are provided, (1) a small trial version with only 1000 integers to use while developing your method, and (2) a test that goes over 1 million integers to test your data on a larger dataset:  [trial\_incomes.csv](https://www.google.com/url?q=http://www3.cs.stonybrook.edu/~has/CSE545/a1/trial_incomes.csv&sa=D&source=editors&ust=1654368083555852&usg=AOvVaw3fmabvDeUlNR2IYI3c_Uu5)  [test\_incomes.csv.zip](https://www.google.com/url?q=http://www3.cs.stonybrook.edu/~has/CSE545/a1/test_incomes.zip&sa=D&source=editors&ust=1654368083556419&usg=AOvVaw0N17h6C5G2xfz5alCjBU11) (unzip for csv after downloading)  No **template code** is providedfor this. Place your name, sbu-id-number, and a brief description of the code at the top. The data should be read in with income\_rdd=sc.textfile("input.csv", 32) and the rest is up to you as per fulfilling the instructions below.  You should write your pyspark code in a .py file that can be run in pyspark with the following command:  spark-submit a2\_**p1**\_CSE545sp22\_lastname\_id.py trial\_incomes.csv  **Task I**) **Calculate distinct incomes, median of incomes, most frequent income, and count per 10power.** For this subtask, you must calculate the true value of all four data summaries you produced in assignment 1:   1. **count of distinct incomes**– The number of distinct incomes in the dataset 2. **median** – The median of all incomes in the dataset: the income at which there is an equal number of values greater than the income as there are values less than the income. 3. **mode**  – The mode of all incomes in the dataset: the most frequently seen income. 4. **count per 10power** –  counting the incomes by powers of 10. That is, for each integer round it down to its nearest power of 10 (for example 3 map to 1 = 100; 30 would map to 10 = 101. 87 would map to 10 = 101; 870 would map to 100 = 102, 100 would map to 100 = 102 etc….).  Your goal is to count the number of integers between each power of 10.   Do not use an approximate algorithm, but count every data point in the calculations.  **However, you are restricted to only doing so with at most four shufflable transformation – a transformation which may cause a shuffle:**  reduceByKey, groupByKey, combineByKey, sortByKey  join, leftOuterJoin, rightOuterJoin, intersection, cogroup, groupWith  distinct, repartition, coalesce  Note that the solution may use less than 4 such transformations and you will also be graded on using spark transformation efficiently. For example, using a groupByKey when reduceByKey would suffice will cause loss of points.  Your code will be tested on a dataset of the same size as test.csv but with a different distribution of positive integers as incomes. Do not assume any particular distribution.  **Checkpoint I)**At the end of this step, print the count of distinct incomes, median of incomes, most frequent income, and count per 10power. |

**Part II. Similar Hospital Search (60 points)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Here, your objective is to find hospitals with similar characteristics in the impact of covid. Being able to quickly find similar hospitals can be useful for connecting hospitals experiencing difficulties and finding the characteristics of hospitals that have dealt better with the pandemic.  **Data.**You will use the dataset "COVID-19 Reported Patient Impact and Hospital Capacity by Facility" provided by the US Health and Human Services, containing 420k rows and 109 columns as reported on March 14, 2022.  https://healthdata.gov/Hospital/COVID-19-Reported-Patient-Impact-and-Hospital-Capa/anag-cw7u  [trial\_COVID-19\_Hospital\_Impact.csv](https://www.google.com/url?q=http://www3.cs.stonybrook.edu/~has/CSE545/a2/trial_COVID-19_Hospital_Impact.csv&sa=D&source=editors&ust=1654368083562509&usg=AOvVaw0d1QMrt90qhs3e5Ien0bzW)  [test\_COVID-19\_Hospital\_Impact.csv](https://www.google.com/url?q=http://www3.cs.stonybrook.edu/~has/CSE545/a2/test_COVID-19_Hospital_Impact.csv&sa=D&source=editors&ust=1654368083563002&usg=AOvVaw1EwHTl7mcowEXiRB3pjZNY)  No **template code** is providedfor this. Place your name, sbu-id-number, and a brief description of the code at the top. The data should be read in with hospitals\_rdd=sc.textfile("input.csv", 32) and the rest is up to you as per fulfilling the instructions below.  You should write your pyspark code in a .py file then be run in pyspark with the following command:  spark-submit a2\_**p2**\_CSE545sp22\_lastname\_id.py trial\_COVID-19\_Hospital\_Impact.csv  Specifically, you must complete:   1. **Extract binary features (i.e. a sparse representation of the characteristic matrix) per hospital (20 points)** For each record, treat the first element as the unique record id. For each of the 108 other columns, treat them as a binary feature with an id as string of the form: "column\_name:value" and add that to a set that represents the values that are "1" in the characteristic matrix (every element of the set not present is assumed to be 0). For example, if the following was the data:  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | hospital\_pk | collection\_week | state | hospital\_name | fips\_code | is\_metro | beds\_used\_avg | | 131312 | 11/5/2021 | ID | ST LUKE'S MCCALL | 16085 | FALSE | 4.3 | | 50739 | 4/16/2021 | CA | CENTINELA HOSPITAL MEDICAL CENTER | 6037 | TRUE | 171.4 |   Then the first two records would be: **(131312,** set('collection\_week:11/5/2021', 'state:ID', 'hospital\_name:ST LUKE'S MCCALL', 'fips\_code:16085', 'is\_metro:FALSE', 'beds\_used\_avg: 4.3')**)**  **(50739,** set('collection\_week:4/16/2021', 'state:CA', 'hospital\_name:CENTINELA HOSPITAL MEDICAL CENTER', 'fips\_code:6037','is\_metro:TRUE', 'beds\_used\_avg:171.4' )**)**  **Checkpoint II.A)**At the end of this step, print the features for the following hospital\_pks (using the format above): 150034, **0**50739, 330231,241326, **0**70008.  **Tips:**   * Use csvreader with mapPartitions to properly convert each row to an array. * Store the header in a broadcast variable that maps row -> header name. * Use set union inside a reduceByKey to get the multiple rows of a single hospital\_pk into a single record of an RDD.  1. **Minhash (20 points)** Create a “signature” for each hospital: Use the efficient minhashing approach from within spark to convert the set representation of each hospital into 100 dimensions by using hashes on the set strings. Requirement: Do not store the hashed values of every potential set element in a broadcast variable – it will be too large.   **Checkpoint II.B)**At the end of this step, print the **signature** vector for the following hospital\_pks: 150034, **0**50739, 330231,241326, **0**70008.  **Tips:**   * Store the hash *functions* in a broadcast variable. * Consider setting things up such that a reduceByKey with key as (i, sid) can be used to find the minimum hashed value for a feat per sid. This would be instead of the line in the slides:                if hi(feat) < Sig[i][sid]: Sig[i][sid] = hi(feat)  1. **Find similar pairs using LSH  (20 points)** Run LSH to find approximately 20 candidates that are most similar to hospitals: 150034, 50739, 330231,241326, 70008. From the perspective of LSH, each hospital is a column with each row being a value of the signatures. Tweak bands and rows per band in order to get approximately 20 candidates (i.e. anything between 10 to 30 candidates per hospital is ok).  While pre-existing  hash code is fine, you must implement LSH otherwise.   **Checkpoint II.C)**At the end of this step, print the  10 to 30 hospitals your LSH returns for the following hospital\_pks: 150034, 50739, 330231,241326, 70008.  For each potential match, print:  (a) hospital\_pk, (b) the Jaccard similarity with the target hospital, and (c) the first 10 values of the signature matrix.  **Tips:**   * Note that there are 100 rows total, but you’re also welcome to divide into a number that doesn’t evenly fit, in which case just leave out the remainder (e.g. 3 bands of 5 rows, and ignore the last row). * Set things up such that each RDD record is a signature, and so from the perspective of LSH, each record is a column.   Your code should run in < 2 minutes across all tests. |

**Submission**

|  |
| --- |
| Submit a text file containing the output of your code as well as your code itself. Please use blackboard to submit two files each with your lastname and student id:   1. a2\_**p1**\_CSE545sp22\_lastname\_id.py 2. a2\_**p1**\_CSE545sp22\_lastname\_id\_OUTPUT.txt 3. a2\_**p2**\_CSE545sp22\_lastname\_id.py 4. a2\_**p2**\_CSE545sp22\_lastname\_id\_OUTPUT.txt   **Do not upload a zip file. Double-check that your files are there and correct after uploading and make sure to submit.** Uploading files that are zips or any other type than python code or txt files will result in the submission being considered invalid. Partially uploaded files or non-submitted files will count as unsubmitted.    **Questions**: Please post questions to the course piazza page.  See Piazza posts tagged "assignment2" ([https://piazza.com/stonybrook/spring2022/cse545](https://www.google.com/url?q=https://piazza.com/stonybrook/spring2022/cse545&sa=D&source=editors&ust=1654368083579293&usg=AOvVaw2K91WpExMvCVZ_EaQvyty4)). |