Bishop’s University

CS 450 – Elements of Big Data

CS 550 – Big Data Management and Analytics

Final Project: Similarity Search

1. Introduction

The objective of this final project 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[1].

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:

trial_COVID-19_Hospital_Impact.csv
test_COVID-19_Hospital_Impact.csv

An example of implementation of Minhash algorithm and finding similar pairs using LSH algorithm based on Spark is provided in Python. Thus, you will not need to implement the final project from scratch. The provided code will serve you for guidance. Therefore, you can adapt the code the way you think it is efficient for you. If you use any external library, you must mention it in your report. The data should be read in with *`hospitals_rdd = spark.sparkContext.textfile.textfile("input.csv", 32)`* and the rest is up

[1] https://healthdata.gov/Hospital/COVID-19-Reported-Patient-Impact-and-Hospital-Capa/anag-cw7u

to you as per fulfilling the instructions below. You should write your code in a Python notebook file that can be run on Google Colab.

2. Tasks

A. Extract binary features (i.e. a sparse representation of the characteristic matrix) per hospital

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 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.

Hints:

- 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.

B. Minhash

Create a "signature" for each hospital: Use the efficient Minhashing approach 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 B) At the end of this step, print the signature vector for the following hospital_pks: 150034, 050739, 330231,241326, 070008.

Hints:

- 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)

C. **Find similar pairs using LSH**

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).

Checkpoint 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.

Hints:

- 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.

3. Submission

You must submit the *pdf* file of your report. In fact, it must contain the responses in form of outputs for the three tasks. In addition, you must submit all your implementations in a`.ipynb` file that can be run on Google Colab. ***Please, do not your final project in .zip or .rar files.***

4. Useful external references

- NumPy documentation
- SciPy documentation
- Spark Apache
- MinHash and LSH
- Extracting, transforming and selecting features
- MinHashLSH example on Spark
- Python programming language