CS-GY-6513 Big Data Spring 2025

Assignment 2 - Spark (115 points + 25 points extra credit)

Due Date: March 22nd, 11:55PM Eastern

Instructions

Refer to the jupyter notebook **HW2.ipynb**, and the ***data*** folder in the resources section of the course website.

*** ALL DATASETS ARE IN THE JUPYTER HUB SHARED FOLDER ***

- **Give attribution to any code you use that is not your original code. This includes any use of AI**
- SUBMIT YOUR SOLUTION AS A JUPYTER NOTEBOOK.
  Use your netid: e.g. jcr365-hw2.ipynb
- If I cannot run your notebook, you will not get full credit.

1. 25 points

Datafile: Bakery.csv

Solve: Show the top 10, by count bought, **per day of the week, for the time period between 6AM and 10AM**

For example (these are made up numbers….)

```
 Item     day           qty
  Bread, Monday, 102
  :
```

2. 25 points

Datafile: Bakery.csv

Solve: Show the top **pair** of items (by qty bought) bought **by Daypart**

**** Must be in the format shown**

For example (not necessarily the right numbers….)

```
  Morning, (bread, coffee), Weekend
```

Where: Weekend = (Saturday, Sunday)
Weekday = (Monday, Tuesday, Wednesday, Thursday, Friday)

3. 40 Points

Dataset: populationbycountry19802010millions.csv
Solve:

For each year after 1980,compute **the country** with the ***biggest percentage increase*** in population year over year. Ignore regions (World, North America, Africa, etc.).

Then find the years with the top rate increase in population and the least increase(or decrease) in population.

For example, (these are made up numbers)
for the year over year for USA in 1981 is:
1981, USA, (324.44694 - 320.27638) / 320.2763 = 1.30%
1982, Aruba, 3.4%

Th. Extren

1981, USA (biggest rate increase)

1987, Kenya (biggest decrease/lowest increase)

4. 25 Points

Dataset: hw1dir
Solve: WordCount

Do ***word count*** exercise using pyspark.
Normalize to ***lower case***.
Replace characters in NOT in this set: **[0-9a-z]** with **space.**

Show the top 20 words

5. Extra Credit, 25 points

Dataset: internet_archive_scifi_v3.txt

Find the 10 most common trigrams
Hint: look in the pyspark.ml library