

# Homework 3: Individual assignment

[Start Assignment](#)

**Due** Friday by 5:29pm      **Points** 100      **Submitting** a file upload (Turnitin enabled)

**File Types** doc, pdf, txt, py, r, ipynb, csv, and zip

**Available** Oct 13 at 9pm - Nov 10 at 11:59pm

CLO 3 - Employ tools (such as Hadoop and Spark) and techniques for big data systems, technologies, and applications as part of the homework and project. [PLO 2, 5, 6]

CLO 7 - Effectively present and communicate knowledge about big data systems, technologies, and applications acquired in the course. [PLO 3, 6]

CLO 9 - Leverage emerging big data systems, technologies, and applications to transform data into knowledge through capturing, managing, analyzing, and understanding large data at volumes and rates. [PLO 2]


CLO 10 – Design analytical solutions in data warehouse, big data databases, data streaming. [PLO 7]

**\*\*Strictly for Class use. Do NOT share outside the class\*\***


**Submit individually - one submission per student**

**For the following assignments, please provide as much evidence of the results as possible, including the code, screenshots (only figures / plots – not text or code or other material that can be copy-pasted) and documentation. Submit only one pdf file and .sql (if applicable) / .ipynb / .py files containing the code with documentation. Along with .ipynb file(s), please also upload the corresponding .py file after converting the .ipynb to .py file. Please reduce the file size. It should ideally not exceed 1 MB. Not reducing the file size before submission may result in a penalty being assessed particularly if Canvas runs out of space causing other students being blocked from submitting their assignments.**

**Please remember the honesty pledge that you signed and give credit to the limited sources from which you quoted material in your homework. Make sure to follow ethical data usage practices, respect data privacy, and cite your sources appropriately. In the interest of time, be brief, to the point and precise, conveying the most important and pertinent information in as few words as possible. Submit your solutions only via Canvas. Make sure you are within the free tier limits – I do not want you to pay to the providers for any of this course HW.**

1. [25 points] Implement a PySpark program to list out the Wikipedia pages with the top ten PageRanks on a sample Wikipedia XML dataset. This exercise will help you understand how to process XML data, extract relevant information, and perform PageRank calculations using PySpark. Check: [https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download)   
([https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download))
- 

2. Here's a naive implementation of Logistic Regression:

[https://github.com/apache/spark/blob/master/examples/src/main/python/logistic\\_regression.py](https://github.com/apache/spark/blob/master/examples/src/main/python/logistic_regression.py)  
 ([https://github.com/apache/spark/blob/master/examples/src/main/python/logistic\\_regression.py](https://github.com/apache/spark/blob/master/examples/src/main/python/logistic_regression.py))

- 2.a. [5 points] Explain the code to the extent you can.
  - 2.b. [10 points] Make any changes to suit the task and run the code on one of your favorite classification datasets. You can use spark installed on your laptop. Determine the train and test accuracy.
  - 2.c. [10 points] Use LogisticRegression from pyspark.ml.classification to determine the train and test accuracy on the same dataset. Compare the two approaches.
- 

3. a. [5 points] Provide an algorithm and

- b. [10 points] write a program in Python to handle the case where the stream to a DGIM algorithm is not bits, but integers, and we want the sum of the last  $k$ .
  - c. [10 points] Use Apache Spark and Kafka to implement it.
- 




4. [25 points]

Analyzing Big Data with PySpark and BigQuery

**Objective:** Learn to use PySpark to analyze large datasets stored in BigQuery; perform data manipulation and analysis tasks using PySpark, demonstrating how to work with distributed data in a cloud-based environment.

Please do not exceed the free tier!

Setup:

1. Create a GCP Project: If you don't already have a GCP account, create one and set up a project. Detailed instructions can be found in the Google Cloud documentation <https://cloud.google.com/resource-manager/docs/creating-managing-projects>  (<https://cloud.google.com/resource-manager/docs/creating-managing-projects>).
2. Enable the BigQuery API: In the GCP Console, enable the BigQuery API for your project. Instructions are available here <https://cloud.google.com/bigquery/docs/quickstarts>  (<https://cloud.google.com/bigquery/docs/quickstarts>).
3. Create a BigQuery Dataset: In the BigQuery Console, create a new dataset and import a large dataset (e.g., public datasets like Google Analytics or the BigQuery public datasets). Instructions for creating a dataset are available here: <https://cloud.google.com/bigquery/docs/datasets>  (<https://cloud.google.com/bigquery/docs/datasets>).
4. Install PySpark on your local machine / laptop.

```
```bash
```

```
pip install pyspark
```

```
```
```

1. Initialize a PySpark Session, connecting it to the Google Cloud project.

```
```python
```

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder \
```

```
    .appName("BigQuery with PySpark") \
```

```
    .config("spark.jars", "/path/to/google-cloud-bigquery-<version>.jar") \
```

```
    .getOrCreate()
```

```
```
```

2. Read Data from BigQuery into a PySpark DataFrame using the `read` method.

```
```python
df = spark.read \
    .format("bigquery") \
    .option("table", "your_project_id.dataset_name.table_name") \
    .load()
```
```



3. Perform Data Analysis: Use meaningful PySpark operations (e.g., `select`, `filter`, `groupBy`, `agg`) to perform data analysis tasks on the DataFrame.




4. Write Results to BigQuery:

```
```python
df.write \
    .format("bigquery") \
    .option("table", "your_project_id.dataset_name.new_table_name") \
    .save()
```
```

- Submit Jupyter Notebook and Python script with comments explaining the analysis and results.

### **References:**

- Google Cloud BigQuery Documentation (<https://cloud.google.com/bigquery/docs> )  
(<https://cloud.google.com/bigquery/docs>)
- PySpark Documentation (<https://spark.apache.org/docs/latest/api/python/index.html> )  
(<https://spark.apache.org/docs/latest/api/python/index.html>)

- PySpark Quick Start ([https://spark.apache.org/docs/latest/api/python/getting\\_started/index.html](https://spark.apache.org/docs/latest/api/python/getting_started/index.html)  
 ([https://spark.apache.org/docs/latest/api/python/getting\\_started/index.html](https://spark.apache.org/docs/latest/api/python/getting_started/index.html))
- Google Cloud Python Client Libraries (<https://cloud.google.com/python/docs/reference>   
<https://cloud.google.com/python/docs/reference>.)
- Google Cloud Education Grant (<https://edu.google.com/programs/credits/cloud/>   
<https://edu.google.com/programs/credits/cloud/>.)

| Some Rubric (1) |                      |                   |                   |
|-----------------|----------------------|-------------------|-------------------|
| Criteria        | Ratings              |                   | Pts               |
| Q1              | 25 pts<br>Full Marks | 0 pts<br>No Marks | 25 pts            |
| Q2.a            | 5 pts<br>Full Marks  | 0 pts<br>No Marks | 5 pts             |
| Q2.b            | 10 pts<br>Full Marks | 0 pts<br>No Marks | 10 pts            |
| Q2.c            | 10 pts<br>Full Marks | 0 pts<br>No Marks | 10 pts            |
| Q3.a            | 5 pts<br>Full Marks  | 0 pts<br>No Marks | 5 pts             |
| Q3.b            | 10 pts<br>Full Marks | 0 pts<br>No Marks | 10 pts            |
| Q3.c            | 10 pts<br>Full Marks | 0 pts<br>No Marks | 10 pts            |
| Q4              | 25 pts<br>Full Marks | 0 pts<br>No Marks | 25 pts            |
|                 |                      |                   | Total Points: 100 |