# Coursework for ECS765P - Big Data Processing

**References:**


1. Spark: The Definitive Guide: Big Data Processing Made Simple, 1st Edition by Bill Chambers and Matei Zaharia, published by O'Reilly Media, 2018. Available at `https://www.oreilly.com/library/view/spark-the-definitive/9781491912201/`.

2. Week 3, 4 and 5 Lectures, Labs and related resources.


## Task 1: Analysis of Twitter Data (25 points)

**Dataset Description:** This dataset contains geospatial and timestamp data for one week worth of Tweets in the contiguous United States. Tweets were created between January 12, 2013 (Saturday) and January 18, 2013 (Friday). The dataset includes fields such as longitude, latitude, timestamp, and timezone. The dataset is available as twitter.csv in the //data-repository-bkt/ECS765/Twitter/ directory.

**Purpose:** The purpose of this task is to analyze Twitter data to gain insights into user behavior, tweet patterns, and geographical distribution over a full week.

## Sample Records

Here are some sample records from the dataset to explain the fields:

| longitude | latitude | timestamp | timezone |
|---|---|---|---|
| -79.9700362 | 39.64467173 | 2.01301E+13 | 1 |
| -81.39673152 | 28.30565653 | 2.01301E+13 | 1 |
| -84.28054332 | 37.75283217 | 2.01301E+13 | 1 |
| -90.69813819 | 38.79061857 | 2.01301E+13 | 2 |
| -85.93 | 39.58 | 2.01301E+13 | 1 |
| -119.156169 | 34.16541731 | 2.01301E+13 | 4 |
| -82.34229422 | 29.65016196 | 2.01301E+13 | 1 |
| -96.7986904 | 32.7927607 | 2.01301E+13 | 2 |
| -75.311087 | 41.0380725 | 2.01301E+13 | 1 |
| -82.34037695 | 29.6488562 | 2.01301E+13 | 1 |

## Explanation of Fields

- **longitude**: The longitude coordinate of the tweet's location.
- **latitude**: The latitude coordinate of the tweet's location.
- **timestamp**: The timestamp when the tweet was created, in scientific notation.
- **timezone**: The timezone offset from UTC where the tweet was created.

## Questions:

1. **Load Data (2 points)**
    - Load the Twitter dataset into a dataframe.
    - Print the total number of entries in the dataset.
    - Include a screenshot of your results in your report. For example:

```
2025-02-07 10:34:55,865 INFO scheduler.TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
2025-02-07 10:34:55,866 INFO scheduler.DAGScheduler: Job 2 finished: count at NativeMethodAccessorImpl.java:0, took 4.027691 s
Total number of entries in the Twitter Dataset: [redacted]
2025-02-07 10:34:55,885 INFO server.AbstractConnector: Stopped Spark@63def715{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2025-02-07 10:34:55,887 INFO ui.SparkUI: Stopped Spark web UI at http://task1-1-b2933094dff96a10-driver-svc.data-science-eex654.svc:4040
2025-02-07 10:34:55,891 INFO k8s.KubernetesClusterSchedulerBackend: Shutting down all executors
2025-02-07 10:34:55,892 INFO k8s.KubernetesClusterSchedulerBackend$KubernetesDriverEndpoint: Asking each executor to shut down
2025-02-07 10:34:55,901 WARN k8s.ExecutorPodsWatchSnapshotSource: Kubernetes client has been closed (this is expected if the application
is shutting down.)
```

2. **Filter Data (2 points)**
    - Filter the dataset to include only tweets from Monday through Friday.
    - Convert the timestamp field to "YYYY-MM-DD" format.
    - Sort your results by date.
    - Include a screenshot of your results in your report. For example:

```
2025-02-07 10:45:57,822 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
2025-02-07 10:45:57,824 INFO scheduler.DAGScheduler: ResultStage 2 (showString at NativeMethodAccessorImpl.java:0) finished in 31.867 s
2025-02-07 10:45:57,824 INFO scheduler.DAGScheduler: Job 2 is finished. Cancelling potential speculative or zombie tasks for this job
2025-02-07 10:45:57,824 INFO scheduler.TaskSchedulerImpl: Killing all running tasks in stage 2: Stage finished
2025-02-07 10:45:57,825 INFO scheduler.DAGScheduler: Job 2 finished: showString at NativeMethodAccessorImpl.java:0, took 31.876406 s
2025-02-07 10:45:57,879 INFO codegen.CodeGenerator: Code generated in 32.294436 ms
2025-02-07 10:45:57,908 INFO codegen.CodeGenerator: Code generated in 14.884427 ms
+------------+----------+-------------------+--------+----------+
|   longitude|  latitude|          timestamp|timezone|      date|
+------------+----------+-------------------+--------+----------+
| ********|********|yyyy-mm-dd hh:mm:ss|    *| yyyy-mm-dd |
| ********|********|yyyy-mm-dd hh:mm:ss|    *| yyyy-mm-dd |
| ********|********|yyyy-mm-dd hh:mm:ss|    *| yyyy-mm-dd |
| ********|********|yyyy-mm-dd hh:mm:ss|    *| yyyy-mm-dd |
| ********|********|yyyy-mm-dd hh:mm:ss|    *| yyyy-mm-dd |
```

3. **Geographical Distribution (6 points)**

   - Create a new column that combines longitude and latitude into a single location field.
   - Group the dataset by location and count the number of tweets per location.
   - Visualize the geographical distribution of tweets using a scatter plot. For an example, see the Scatter Plot Example at `https://matplotlib.org/stable/gallery/shapes_and_collections/scatter.html`.
   - Include a screenshot of your visualization in your report.

4. **Add Columns (5 points)**

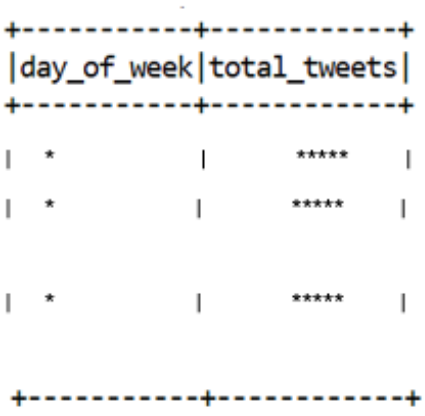
   - Add new columns by extracting the hour and the day of the week from the timestamp.
   - Create a new column that categorizes the time of day into "Morning" (5 AM - 11 AM), "Afternoon" (12 PM - 4 PM), "Evening" (5 PM - 9 PM), and "Night" (10 PM - 4 AM).
   - Show 10 rows of the results in your report.
   - Plot a bar chart for number of tweets for each time of day (Morning, Afternoon, Evening, Night). For an example, see Bar Chart Example at `https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html`.

5. **Group and Aggregate (4 points)**

   - Group the dataset by day of the week and aggregate total number of tweets per day.
   - Create new columns based on the aggregated metrics.
   - Show 10 samples of the results in your report.
   - Visualize the aggregated metrics using a bar chart. For an example, see Bar Chart Example at `https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html`.
   - Include a screenshot of your visualization in your report.

6. **Filter Aggregates (3 points)**

   - Apply a filter to the aggregated results to identify days with an unusually high number of tweets. For this task, consider a day to have an unusually high number of tweets if the total number of tweets on that day is greater than the mean number of tweets per day.
   - Include the filtered results showing the days with an unusually high number of tweets. For example:




7. **Top Entries (3 points)**

   - Find the top 10 locations with the highest number of tweets.
   - Include a screenshot of your results. For example: