Weeks 11 & 12: Final Project Assignment

# Objective:

The goal of this project is to build a complete machine learning pipeline with a focus on data engineering. You will use NiFi to ingest data, manage it in HDFS and Hive, perform machine learning with Spark MLlib, store the results in HBase via Spark, and query the results through Hive. This project will demonstrate your ability to integrate multiple components in an enterprise data architecture while ensuring the uniqueness of your dataset and downstream schemas.

# Requirements:

1. Data Ingestion (NiFi):

* Use NiFi to ingest your dataset and load it into HDFS.

2. Data Storage (Hive):

* Move the data from HDFS into a Hive table.
* You can either:
  + Manually load the data into a Hive table, or
  + Use an external table with the LOCATION clause to put a schema on top of your HDFS directory, or
  + Use Spark to read the data from HDFS and load into a Hive table.

3. Machine Learning (Spark MLlib):

* Use Spark MLlib to build and train a machine learning model (either supervised or unsupervised) on your dataset.
* The Spark code must read the data from the Hive table.
* Evaluate your model performance using metrics like accuracy, F-score, or other relevant measures within the Spark code.

4. Store Results in HBase (Spark):

* Load the results of your machine learning model (e.g., performance metrics) into an HBase table using Spark.

5. Hive-HBase Integration:

* Create a Hive table on top of the HBase data.
* Run SQL queries on the HBase data through this Hive table.

# Important Notes:

You will be provided with examples that demonstrate how to connect NiFi, HDFS, Hive, Spark, and HBase. However, you must use a unique dataset and schema. Reusing data or schemas from the examples will result in a grade of zero.

Ensure the dataset and all downstream schemas are entirely unique.

# Submission Guidelines:

* Submit your work in a comprehensive report.
* Your report must:  
  + Provide a clear, step-by-step explanation of the entire pipeline, from data ingestion using NiFi to querying the results from HBase.
  + Include detailed screenshots for each component (NiFi, HDFS, Hive, Spark MLlib, HBase), showing successful execution and integration.
  + Provide screenshots of all commands executed in Hive and HBase, along with the output to verify that the data was successfully processed.
  + Include the full Spark code used for the machine learning task, along with output metrics (e.g., accuracy, F-score) to verify the model's performance.
  + Include the output of your Spark code execution proving it ran with no errors.
  + Include a clear explanation of any issues encountered and how they were resolved.
  + Include the link to your chosen dataset, ensuring it is different from any provided examples.

# Evaluation Criteria:

* Completeness and correctness of your end-to-end pipeline.
* Effective use of the technologies taught in the course.
* Uniqueness of the dataset and schemas.
* Clarity and organization of your report.

## Examples:

You can refer to examples from the following repository: [DSC650 Week 11-12 Assignment Examples](https://github.com/bellevue-university/dsc650-updated/tree/main/week11-12/assignment/examples)  
  
You are encouraged to use these resources as guides, **but you cannot use the same datasets or schemas provided in these examples:**

**Please refer to the interactions in the examples below, as the Docker networking has already been configured and tested for seamless connectivity between the containers.**

* **Example\_Project.json:** This NiFi template downloads **customers-100.csv** from the assignment folder and puts it into HDFS in the /tmp directory. You can use your own data, load it to HDFS, and then put it in a Hive table. You may also explore Hive external tables, which allow Hive to read data from a directory without moving the data.
* **SparkAddLibraries.docx:** This guide walks through adding a Python library and running it with PySpark (e.g., using pip to install the requests library). You may explore other libraries in your project.
* **HdfsToHivePySpark.docx**: This document guides you through reading a CSV file from HDFS, creating a Hive table using PySpark, loading the data into the Hive table with PySpark, and then querying the table using PySpark.
* **QueryHivePySpark.docx:** This document guides you through creating a Hive table with sample data using the Hive CLI and querying it using PySpark.
* **WriteHBasePyspark.docx:** This document walks you through installing the happybase Python library, creating an HBase table using the HBase shell, and writing data to it using PySpark.
* **SparkMachineLearning.docx:** This document walks you through creating a Linear Regression Machine Learning model with PySpark, using data in Hive and Spark MLib library.
* **HiveQueryHbase.docx**: This document walks you through creating an HBase table and loading data into it using the HBase CLI, creating a Hive table ontop of the HBase table, and using Hive to query the data stored in the HBase table.