BDA 2024-2025 Assignment:

Datasets:

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

Description:

This project aims to simulate a real-time Big Data solution for detecting fake news using assertive textual statements. We begin by training a machine learning model in PySpark using labeled datasets containing real and fake assertives. The model is designed to classify new statements based on their textual features through a pipeline of tokenization, stopword removal, and TF-IDF vectorization, followed by logistic regression.

Once the model is trained and saved, we simulate a streaming environment in Databricks Community Edition by feeding new assertive statements incrementally from a monitored directory. Each new input is processed in real time using Spark Structured Streaming, passed through the pre-trained model, and classified as fake or real. The predictions are persistently stored in a Delta Table, allowing for scalable storage and fast querying.

This approach demonstrates not only the machine learning capabilities of Spark but also how a full Big Data pipeline can be simulated in a constrained environment, with components for ingestion, processing, storage, and real-time analysis — reflecting the core principles of Big Data systems: volume, velocity, variety, and veracity.

1 – Offline: Model Training

This is where we prepare the model to classify assertives as *real* or *fake*.

Tasks:

1.1 Load and merge datasets (real.csv, fake.csv)

1.2 Preprocess text: Tokenizer, StopWordsRemover, TF-IDF

1.3 Train classification model (e.g., Logistic Regression) using PySpark MLlib

1.4 Evaluate the model: accuracy, confusion matrix

1.5 Save the trained model to /mnt/models/fake_news_model

---

2 – Online: Streaming Simulation

This wiil simulate assertives arriving in real-time and being classified by the model.

Tasks:

2.1 Prepare small CSV files with new assertives (simulate batches arriving)

2.2 Set up a folder to receive those files: /mnt/streaming/fake_news_input

2.3 Use readStream to watch that folder for new data

2.4 Load the saved model and apply it to each incoming assertive

---

3 – Persistence: Storing Predictions

We don't just predict — we store the results for analysis and further action.

Tasks;

3.1 Write streaming results to a **Delta Table** using .writeStream

3.2 Use /mnt/checkpoints/... for tracking state

3.3 Table created: streamed_fake_news_predictions

---

4 – Exploration: Querying & Visualizing

Once stored, we can treat predictions as regular structured data.

Tasks:

4.1 Use %sql to query the Delta Table (e.g., count fake vs real)

4.2 Use charts inside Databricks: pie, bar, time-series

4.3 Optional: Simulate dashboarding (Power BI, Grafana in real-world cases)

---

5 – Report and Presentation

Our final document and oral discussion should explain all of this clearly.

Section	**Content**
Introduction	Fake news context, why it's a Big Data problem
Dataset	Description of data sources, labels
Methodology	Preprocessing, model training, evaluation
Streaming Architecture	How streaming was simulated, what tools were used
Data Persistence	Why you store predictions, Delta Table structure

Section	Content
Results	Model accuracy, streaming predictions, SQL queries
Final Thoughts	How it scales, real-world application possibilities