VINEET SRIVASTAVA (AI-Intern DFCI)

Observations while creating delta-live table pipeline for Large Language Models Inference:

1. Comparison between Azure Data Factory (ADF) and Delta-Live Tables (DLT) for Orchestration of data pipelines

DLT Advantages

Simpler to use. DLT pipelines are declarative, meaning that you define the desired state of your data, and DLT figures out how to get there. This makes DLT pipelines easier to understand and maintain than ADF pipelines, which are more procedural.

More efficient. DLT pipelines use Delta Lake's incremental processing capabilities to only update the parts of your data that have changed. This can result in significant performance improvements over ADF pipelines, which typically process all your data every time they run.

Better data lineage. DLT pipelines track the lineage of your data, so you can see how each piece of data has been transformed over time. This can be helpful for debugging problems and understanding how your data is used.

ADF Advantages

More flexibility. ADF pipelines are more flexible than DLT pipelines, as they can be used to orchestrate a wider variety of data processing tasks.

Better integration with other Azure services. ADF is tightly integrated with other Azure services, such as Azure Storage and Azure Synapse Analytics. This can make it easier to build end-to-end data pipelines that use these services.

2. **Ease of use.** Pretty easy to use almost like plug and play since due to the available UI settings and one can integrate the delta-lake layers (bronze, silver, gold) notebooks easily.

3. **Visual Representation** of pipelines or delta-lake layers available with DLT pipelines

4. **Data Quality checks/controls** using built-in exceptions like ‘drop records’ or ‘fail validation’.

5. Can create libraries conflicts and that can be hard to debug. Python Libraries conflicts like Tensor-flow and NumPy or Transformers can be hard to debug. This can be faced by one user while the other user may not face it which makes it more challenging. Current versions of libraries which resolved conflicts:

```
! pip install tensorflow=='2.11.0'

! pip install numpy=='1.21.5'
! pip install transformers=='4.29.2'
```

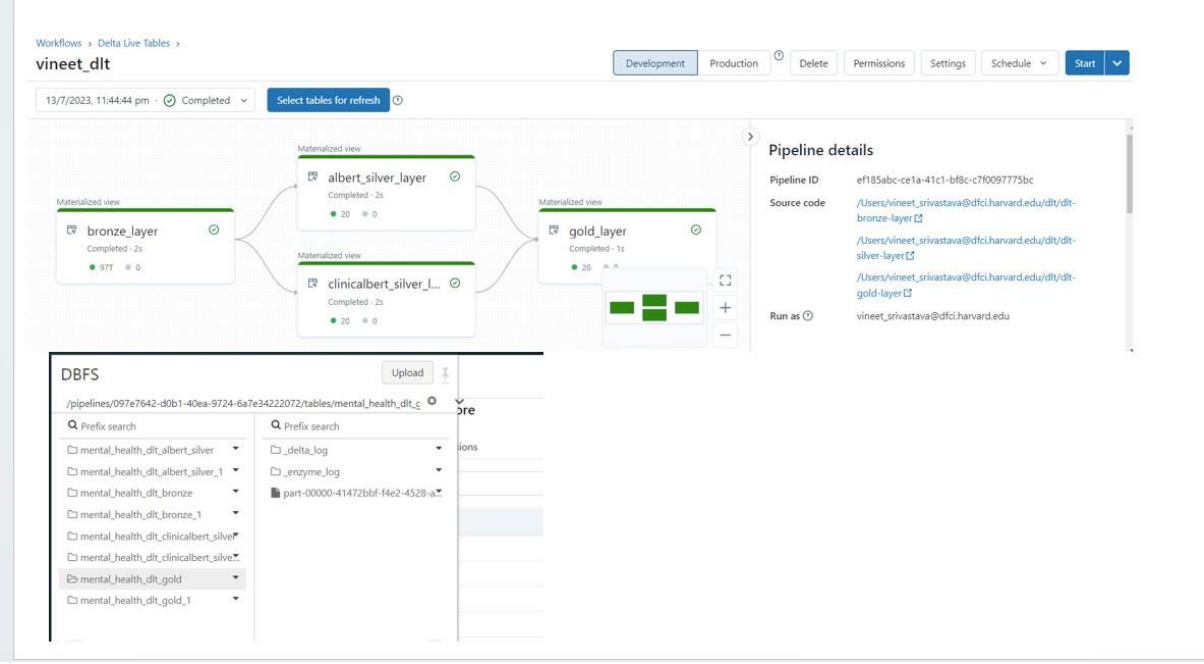
6. Spark based Data Transformation issues especially when dealing with Large Language based models or deep learning-based models which requires inference pre-trained or fine-tuned models. Delta-live tables are spark based and even though reading a delta-live table (dlt.read()) returns a spark dataframe, it doesn't perform or support the same operations that a normal spark dataframe does. This can be hard to debug when dealing with complex models.
7. May be hard to debug. It can be hard to debug in terms of adding loggers to check the messages or understanding DLT tables content. {However, need to explore this more before making final verdict.}
8. Based on point number 6, as above, to solve spark-based transformation issues especially for complex deep learning or large language model inference, following points could be helpful.
    a. Firstly, to infer/make predictions on hidden dataset using pre-trained models, firstly load the data into databricks ecosystem as tables. Keep a dedicated notebook for this.
    b. Secondly, ingest the above ingested databricks table into delt-live bronze layer by reading it as spark dataframe.

        df=spark.read.format("parquet").option("header","true").load("dbfs:/user/hive/warehouse/<table>")
        now do all necessary spark/pandas-based transformations and model inference in this layer itself such as tokenization on texts, encoding, spark-to-pandas and then pandas-to-spark conversions, etc. This is because once you save the dataframe as DLT table and then read the table in next layer like silver-layer, you may face issues with above operations.

    c. Only keep normal spark-based operations while reading DLT saved tables (df = dlt.read(<path of DLT Table>)) from previous layers like while reading bronze dlt layer in silver layer. Some basic operations like join, drop null values, drop columns and other basic EDA operations. Do not do complex spark operations here because DLT read based dataframe, even though returns spark, doesn't support the same functionalities especially while making complex inferences.

Also, DLT table to pandas conversions is not possible currently unlike spark-to-pandas conversions.

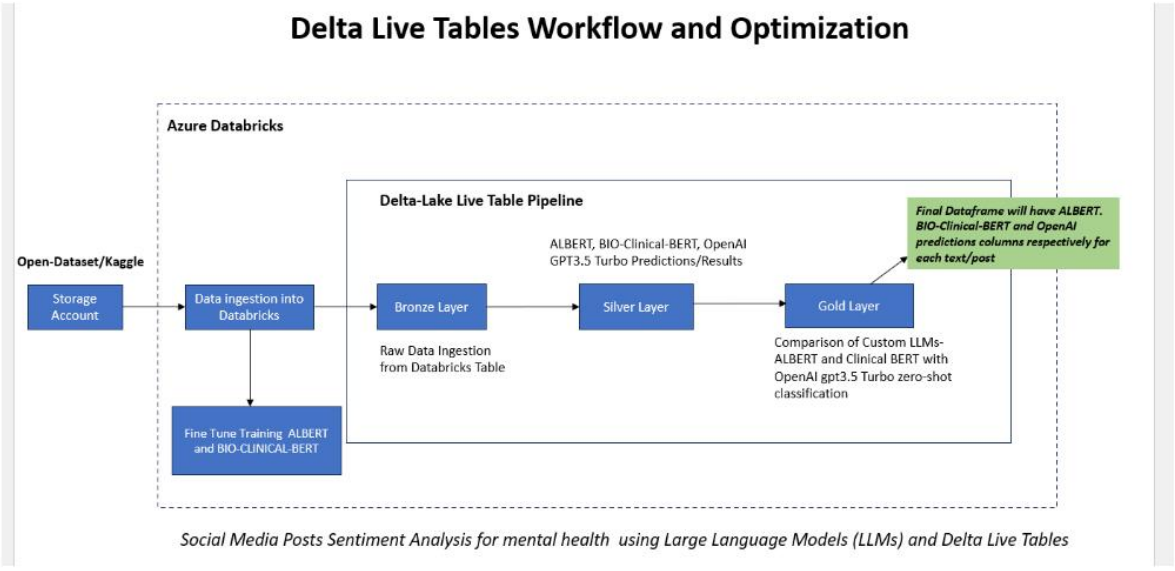
Results:



Architecture:

DLT Tables Utilization- pipeline creation

Architecture



Social Media Posts Sentiment Analysis for mental health using Large Language Models (LLMs) and Delta Live Tables