2023 Fall Session 5: Introduction to Python Coding in Databricks

1. **Q - How can I connect Centralized Data Repository's (CDR's) data (e.g.: schema = beneficiary_data) in Databricks by using Python? Do you have a coding example?**
   A - One way would be to use PySpark functionality with SQL statements. You should be able to run the following: ben_df = spark.sql("SELECT * from beneficiary_data.beneficiary limit 10").

2. **Q - If we are new would you recommend PySpark Pandas or just PySpark?**
   A - It depends on your use case and experience. For a bit DataFrames (DFS) PySpark Pandas (aka Pandas application programming interface (API) on Spark) shines compared to just PySpark and the Pandas-like syntax and viewing are huge pluses. For larger DFS, PySpark is preferred.

3. **Q - What is the sort algorithm working behind sort function? Can we pick sort algorithms of our choice here i.e., quicksort etc.?**
   A - This function takes up the sorting algorithm to sort the data based on input columns provided. It takes up the column value and sorts the data based on the conditions provided. The sort condition can be ascending or descending depending on the condition value provided. The columns values are checked accordingly with the corresponding values and the data is sorted up. It can take up a single column value as well as the multiple column values sorting the data accordingly over. It is a multiphase process so shuffling of data is done while sorting the data. It does not repartition the data and keeps the current partitions only. PySpark has the concept of Eliminate Sort function that is an optimization technique that is used, it eliminates the sort function that has no effect over the final operation that makes the operation less expensive while sorting the data in the PySpark model.

4. **Q - Does NumPy work with Spark Pandas? For example, I use np.where() with regular Pandas. Can I do this with the Spark Pandas DataFrames?**
   A - NumPy has some availability with PySpark Pandas, but with the PySpark Pandas package still in development and being a new library, it does not have all the features of Pandas and other libraries that Pandas utilizes effectively. There may be times when just using PySpark DataFrames will be more effective than trying to use PySpark Pandas due to some limitations, especially when working with large DataFrames.

5. **Q - Are window joins allowed and what are the functions to use?**
   A - We would need to know exactly which joins and how they would be used to provide an answer. For questions on a specific join, submit a ServiceNow ticket.

6. **Q - Can we share the underlying reasons for Pandas Spark DataFrame being less performant than Spark DataFrame?**
   A - A Spark DataFrame is always faster than a Panda Spark DataFrame. That is because

even though it is utilizing the Spark distribution on the compute, it still utilizes some of the Pandas functionality. Pandas was derived from the NumPy package and how it takes information and looks at it from a multiple dimension array, a matrix. There are certain functionalities and features that Pandas rely on the data being distributed in a certain way. Spark Pandas does distribute the data, but it does not distribute it as efficiently as a true Spark DataFrame distributes it.