Lab4-Task2: Guided solution for detect 500% Carrier Delay Outliers

Objective: Identify flights for each airline (carrier) where the arrival delay of a specific flight is more than 500% of the average delay of all flights for that carrier.

Overview

The given Spark application performs the following tasks:

1. Reads the flight data into a DataFrame.
2. Identifies unique combinations of carrier, origin, and destination airports.
3. Utilizes Window functions to look for sequences of consecutive days with the same flight.
4. Isolates the first and last days of such sequences.
5. Calculates the length of each sequence and sorts them in descending order.

Guided Solution:

Spark Environment Setup: Import Libraries:

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql import Window
```

Initialize Spark Session: Establish a local Spark session utilizing all available cores with 4GB memory allocation.

```
spark = SparkSession \
    .builder \
    .master("local") \
    .config("spark.driver.memory", "4g") \
    .appName('ex4_anomalies_detection') \
    .getOrCreate()
```

Define Window Specification: This window groups data by Carrier and considers all rows for each carrier, which means it takes into account all records when calculating the average.

```
unbounded_window =
Window.partitionBy(F.col('Carrier')).rowsBetween(Window.unboundedPreceding,
Window.unboundedFollowing)
```

Load Data: Fetch the flight data from the specified S3 path and cache it for enhanced performance during subsequent operations.

```
flights_df = spark.read.parquet('s3a://spark/data/transformed/flights/')

flights_df.cache()
```

Calculate All-Time Average Delay: For each flight of a specific carrier, compute the average delay of all flights for that carrier. The result is stored in a new column named avg_all_time.

```
avg_delay_df = flights_df \
   .withColumn('avg_all_time', F.avg(F.col('arr_delay')).over(unbounded_window))
```

Determine Delay Deviation: For each flight, calculate the percentage difference between its arrival delay and the all-time average delay for its carrier.

```
deviation_df = avg_delay_df \
   .withColumn('avg_diff_percent', F.abs(F.col('arr_delay') / F.col('avg_all_time')))
```

Filter Outliers: Retain only the flights where the delay deviation is more than 500% (or 5.0 as a decimal fraction).

```
outliers_df = deviation_df.where(F.col('avg_diff_percent') > F.lit(5.0))
```

Display Results:
Showcase the records that exhibit a substantial deviation from their historical average delay. Release the cached data and shut down the Spark session.

```
outliers_df.show()
flights_df.unpersist()
spark.stop()
```