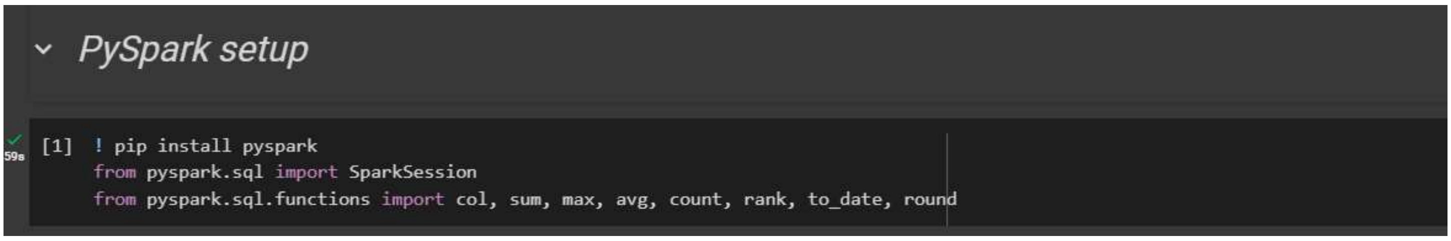
SaiPrabath Chowdary S

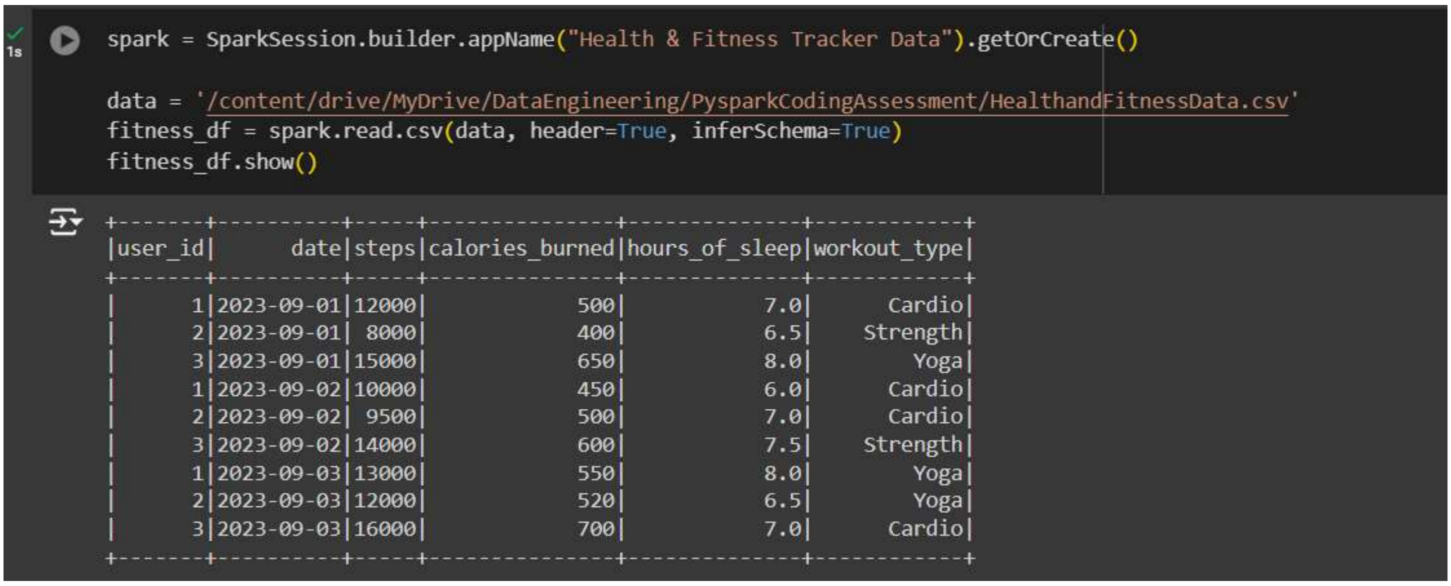
PySpark Coding Challenge 10th Sep 2024

# Health & Fitness Tracker Data



## PySpark setup

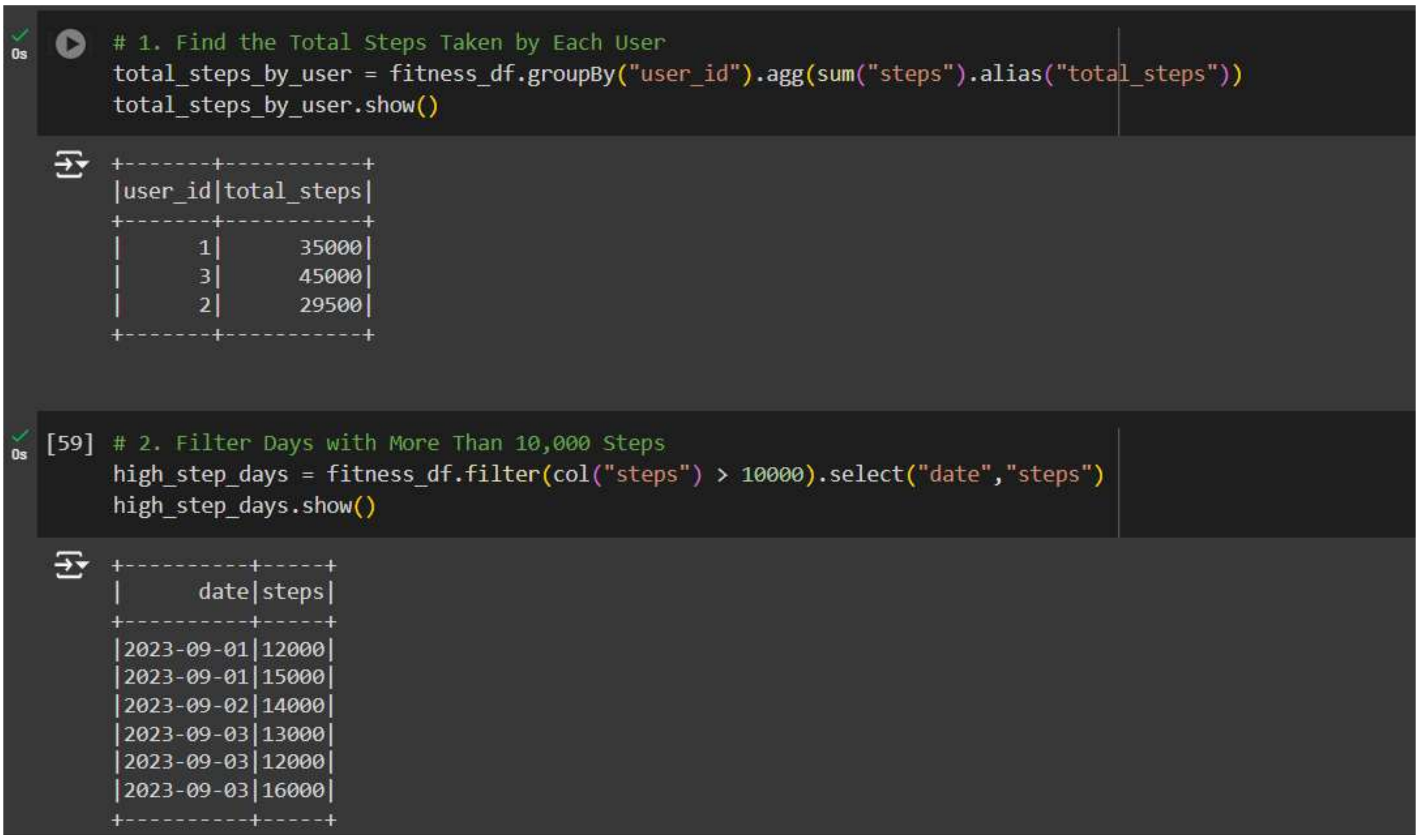
```
[1] ! pip install pyspark
    from pyspark.sql import SparkSession
    from pyspark.sql.functions import col, sum, max, avg, count, rank, to_date, round
```



```
spark = SparkSession.builder.appName("Health & Fitness Tracker Data").getOrCreate()

data = '/content/drive/MyDrive/DataEngineering/PysparkCodingAssessment/HealthandFitnessData.csv'
fitness_df = spark.read.csv(data, header=True, inferSchema=True)
fitness_df.show()
```

```
+-------+----------+-----+---------------+--------------+------------+
|user_id|      date|steps|calories_burned|hours_of_sleep|workout_type|
+-------+----------+-----+---------------+--------------+------------+
|      1|2023-09-01|12000|            500|           7.0|      Cardio|
|      2|2023-09-01| 8000|            400|           6.5|    Strength|
|      3|2023-09-01|15000|            650|           8.0|        Yoga|
|      1|2023-09-02|10000|            450|           6.0|      Cardio|
|      2|2023-09-02| 9500|            500|           7.0|      Cardio|
|      3|2023-09-02|14000|            600|           7.5|    Strength|
|      1|2023-09-03|13000|            550|           8.0|        Yoga|
|      2|2023-09-03|12000|            520|           6.5|        Yoga|
|      3|2023-09-03|16000|            700|           7.0|      Cardio|
+-------+----------+-----+---------------+--------------+------------+
```

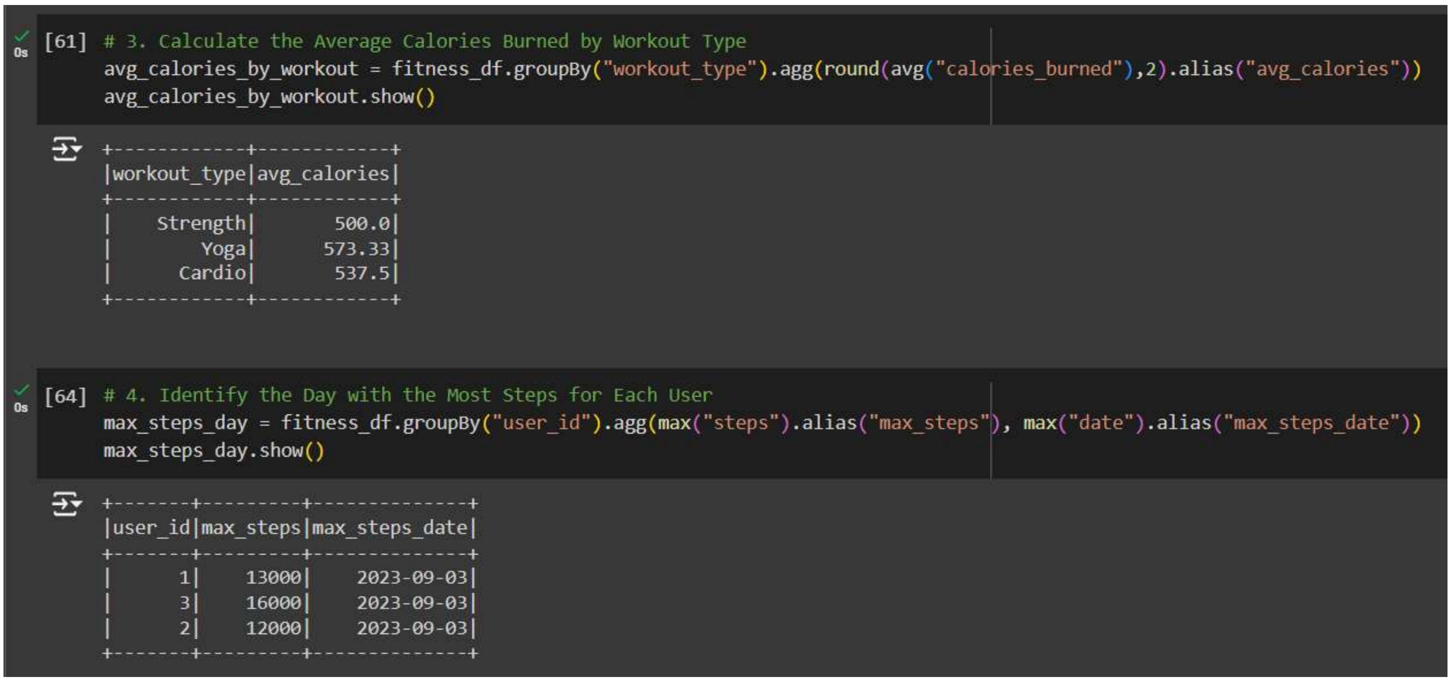


```
# 1. Find the Total Steps Taken by Each User
total_steps_by_user = fitness_df.groupBy("user_id").agg(sum("steps").alias("total_steps"))
total_steps_by_user.show()
```

```
+-------+-----------+
|user_id|total_steps|
+-------+-----------+
|      1|      35000|
|      3|      45000|
|      2|      29500|
+-------+-----------+
```

```
[59] # 2. Filter Days with More Than 10,000 Steps
     high_step_days = fitness_df.filter(col("steps") > 10000).select("date","steps")
     high_step_days.show()
```

```
+----------+-----+
|      date|steps|
+----------+-----+
|2023-09-01|12000|
|2023-09-01|15000|
|2023-09-02|14000|
|2023-09-03|13000|
|2023-09-03|12000|
|2023-09-03|16000|
+----------+-----+
```

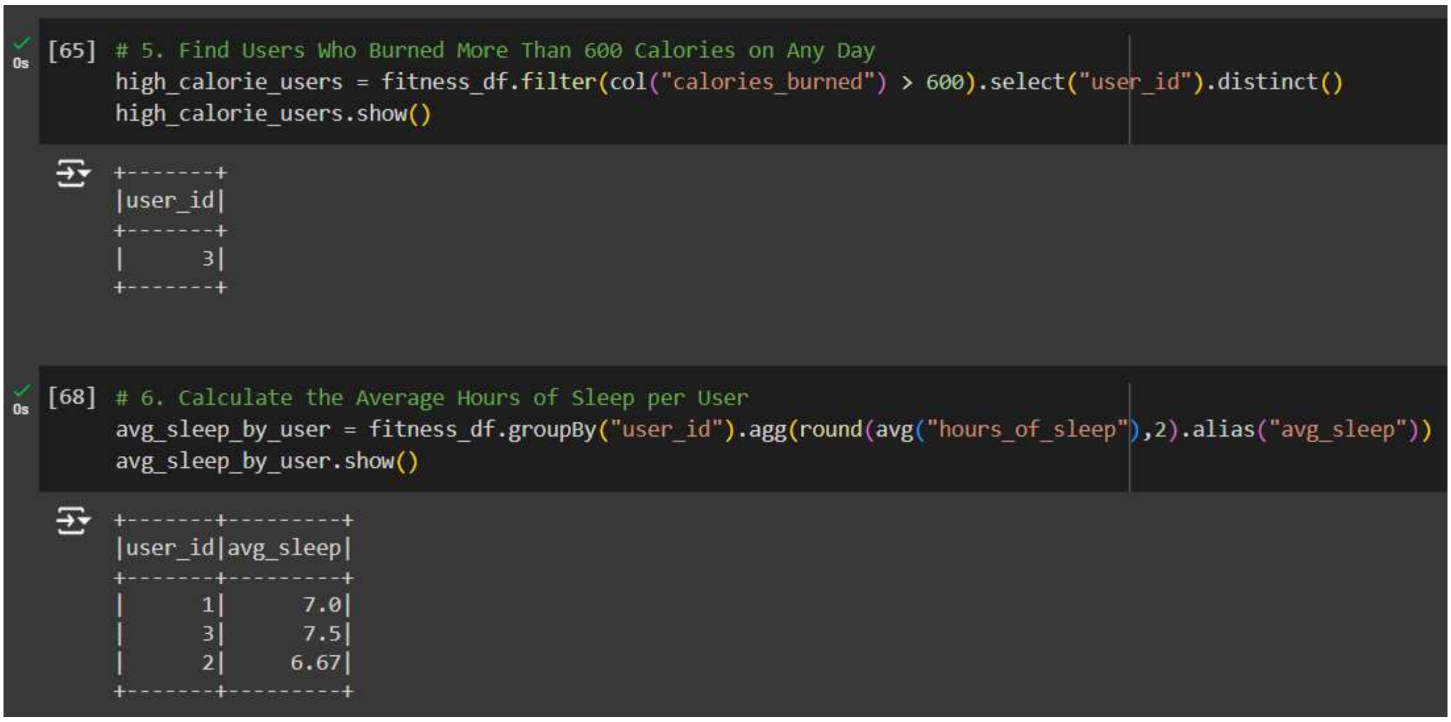


```
[61] # 3. Calculate the Average Calories Burned by Workout Type
     avg_calories_by_workout = fitness_df.groupBy("workout_type").agg(round(avg("calories_burned"),2).alias("avg_calories"))
     avg_calories_by_workout.show()
```

```
+------------+------------+
|workout_type|avg_calories|
+------------+------------+
|    Strength|       500.0|
|        Yoga|      573.33|
|      Cardio|       537.5|
+------------+------------+
```

```
[64] # 4. Identify the Day with the Most Steps for Each User
     max_steps_day = fitness_df.groupBy("user_id").agg(max("steps").alias("max_steps"), max("date").alias("max_steps_date"))
     max_steps_day.show()
```

```
+-------+---------+--------------+
|user_id|max_steps|max_steps_date|
+-------+---------+--------------+
|      1|    13000|    2023-09-03|
|      3|    16000|    2023-09-03|
|      2|    12000|    2023-09-03|
+-------+---------+--------------+
```

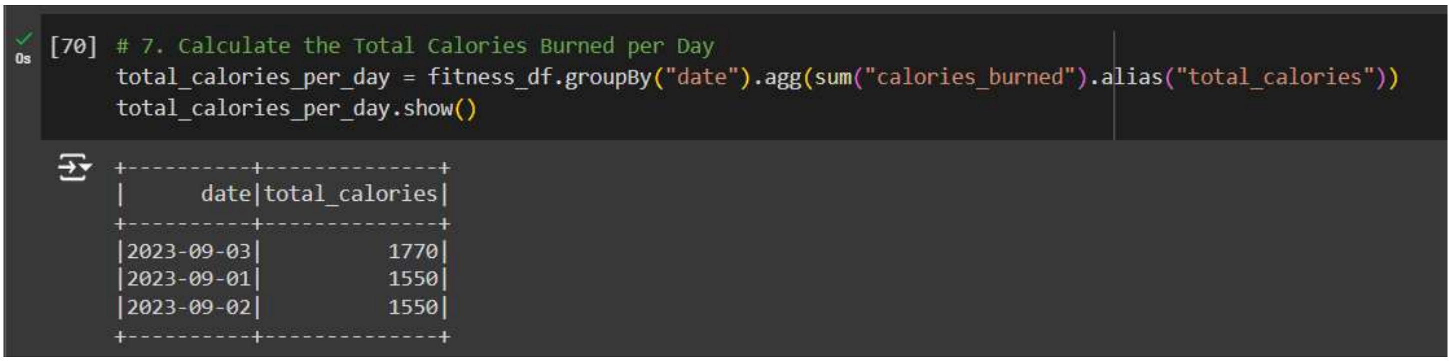


```
[65] # 5. Find Users Who Burned More Than 600 Calories on Any Day
     high_calorie_users = fitness_df.filter(col("calories_burned") > 600).select("user_id").distinct()
     high_calorie_users.show()
```

```
+-------+
|user_id|
+-------+
|      3|
+-------+
```

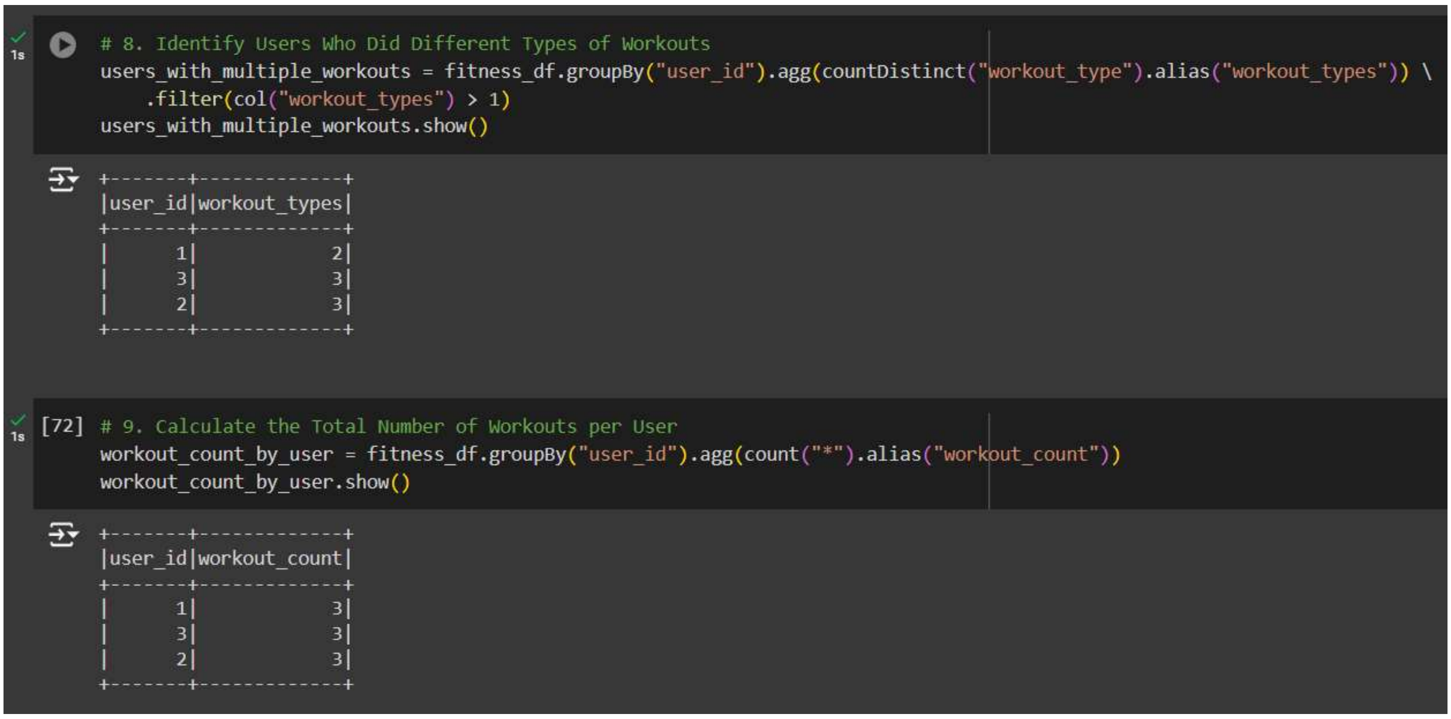
```
[68] # 6. Calculate the Average Hours of Sleep per User
     avg_sleep_by_user = fitness_df.groupBy("user_id").agg(round(avg("hours_of_sleep"),2).alias("avg_sleep"))
     avg_sleep_by_user.show()
```

```
+-------+---------+
|user_id|avg_sleep|
+-------+---------+
|      1|      7.0|
|      3|      7.5|
|      2|     6.67|
+-------+---------+
```



```
[70] # 7. Calculate the Total Calories Burned per Day
     total_calories_per_day = fitness_df.groupBy("date").agg(sum("calories_burned").alias("total_calories"))
     total_calories_per_day.show()
```

```
+----------+--------------+
|      date|total_calories|
+----------+--------------+
|2023-09-03|          1770|
|2023-09-01|          1550|
|2023-09-02|          1550|
+----------+--------------+
```



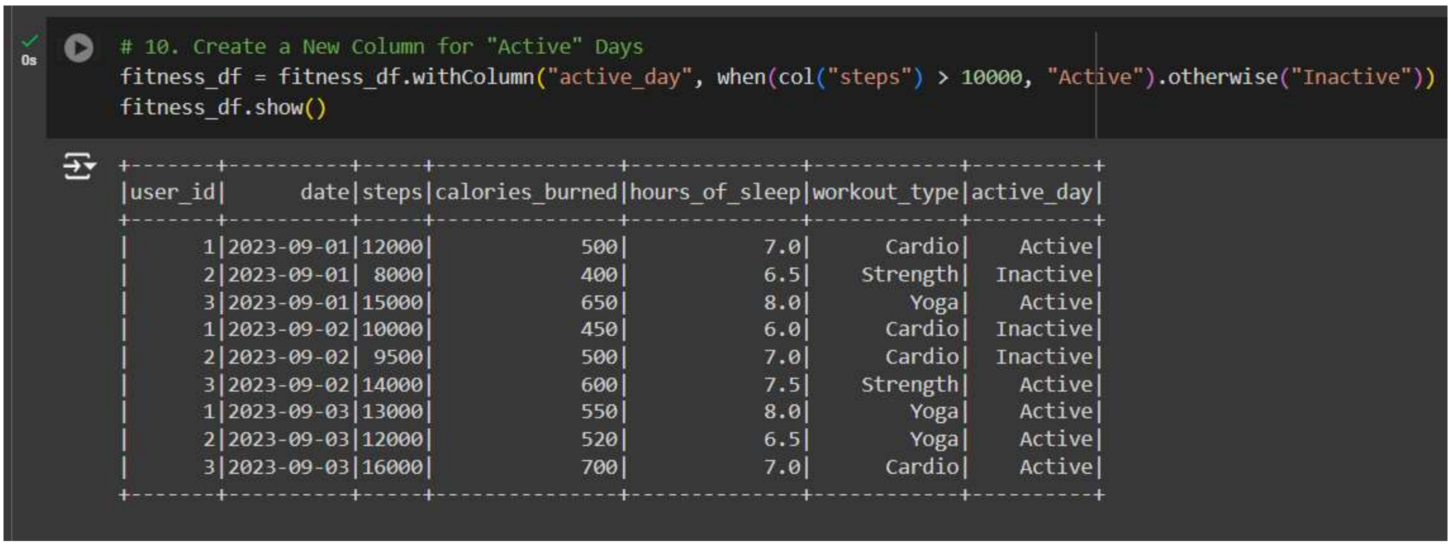
```
# 8. Identify Users Who Did Different Types of Workouts
users_with_multiple_workouts = fitness_df.groupBy("user_id").agg(countDistinct("workout_type").alias("workout_types")) \
    .filter(col("workout_types") > 1)
users_with_multiple_workouts.show()
```

```
+-------+-------------+
|user_id|workout_types|
+-------+-------------+
|      1|            2|
|      3|            3|
|      2|            3|
+-------+-------------+
```

[72]

```
# 9. Calculate the Total Number of Workouts per User
workout_count_by_user = fitness_df.groupBy("user_id").agg(count("*").alias("workout_count"))
workout_count_by_user.show()
```

```
+-------+-------------+
|user_id|workout_count|
+-------+-------------+
|      1|            3|
|      3|            3|
|      2|            3|
+-------+-------------+
```



```
# 10. Create a New Column for "Active" Days
fitness_df = fitness_df.withColumn("active_day", when(col("steps") > 10000, "Active").otherwise("Inactive"))
fitness_df.show()
```

```
+-------+----------+-----+---------------+--------------+------------+----------+
|user_id|      date|steps|calories_burned|hours_of_sleep|workout_type|active_day|
+-------+----------+-----+---------------+--------------+------------+----------+
|      1|2023-09-01|12000|            500|           7.0|      Cardio|    Active|
|      2|2023-09-01| 8000|            400|           6.5|    Strength|  Inactive|
|      3|2023-09-01|15000|            650|           8.0|        Yoga|    Active|
|      1|2023-09-02|10000|            450|           6.0|      Cardio|  Inactive|
|      2|2023-09-02| 9500|            500|           7.0|      Cardio|  Inactive|
|      3|2023-09-02|14000|            600|           7.5|    Strength|    Active|
|      1|2023-09-03|13000|            550|           8.0|        Yoga|    Active|
|      2|2023-09-03|12000|            520|           6.5|        Yoga|    Active|
|      3|2023-09-03|16000|            700|           7.0|      Cardio|    Active|
+-------+----------+-----+---------------+--------------+------------+----------+
```