Data exploration and analysis was completed utilizing big data tools such as HDFS, NiFi, Spark, and Solr.

Workout Activity

Week 11 - 12

Julie Campbell

The dataset that was chosen for the project consists of records of a person's physical activity recorded by a Redmi GPS Watch (Sharma, 2022). It includes data such as the date the activity was performed, workout type, duration, calories burned, and additional details about the result of the workout. Utilizing the tools discussed in the course, the data was explored and evaluated to determine how to maximize workout times to achieve fitness goals. For this project the workout duration, calorie burn, and heart rate will be the primary way to evaluate how to achieve the best workout pattern.

After downloading the data, I placed the file on the google virtual machine. NiFI was chosen as my first step to monitor and organize the flow of data between tools. Exploration was done on the 'InvokeHTTP' processor, but I found that it was more practical to replace the original file. I also did some exploration on 'GetHDFS' and 'FetchHDFS' based on the template that was provided through teams; however, I was unable to successfully connect to the HDFS instance even when inserting a temporary file. After all options were researched, I chose the 'GetFile' processor to modify and store the dataset. Once the file entered the flow, the 'QueryRecord' processor was used to remove the first column with the dummy ID. The final file replaces the original and the dataset was also transferred over to an existing Solr collection as shown in Figure 1.

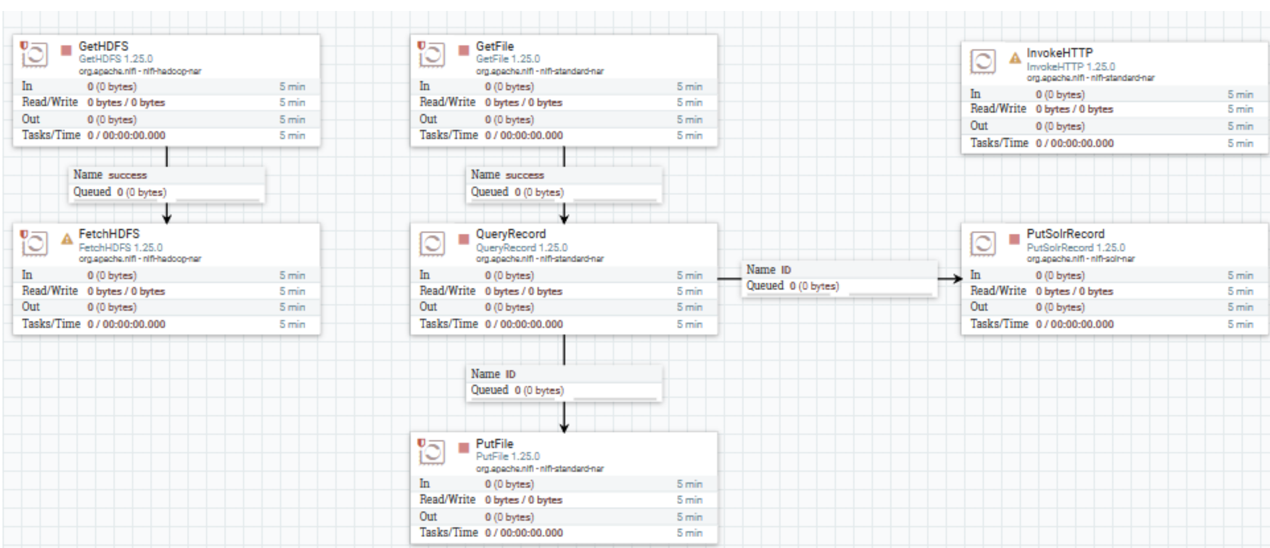

Figure 1: NiFI Flow

With the modified file in place, it was then loaded into HDFS to connect the dataset within the PySpark environment. During the initial load, I had noticed that the numeric data that I was utilizing was uploaded as strings and wouldn't allow me to perform the mathematical operations I needed. A schema was created utilizing the pyspark.sql.types library and struct types to define the format I wanted to load my data in as. The result of the schema change is shown in Figure 2.

```
>>> df.printSchema()
root
 |-- id: integer (nullable = true)
 |-- activity_day: date (nullable = true)
 |-- workout_type: string (nullable = true)
 |-- distance: double (nullable = true)
 |-- time: integer (nullable = true)
 |-- calories: integer (nullable = true)
 |-- total_steps: double (nullable = true)
 |-- avg_speed: double (nullable = true)
 |-- avg_cadence: double (nullable = true)
 |-- max_cadence: double (nullable = true)
 |-- avg_pace: string (nullable = true)
 |-- max_pace: string (nullable = true)
 |-- min_pace: string (nullable = true)
 |-- avg_heart_rate: double (nullable = true)
 |-- max_heart_rate: double (nullable = true)
 |-- min_heart_rate: integer (nullable = true)
 |-- vo2_max: integer (nullable = true)
 |-- aerobic: integer (nullable = true)
 |-- anaerobic: integer (nullable = true)
 |-- intensive: integer (nullable = true)
 |-- light: integer (nullable = true)
```

Figure 2 Pyspark Schema

I also imported the pyspark.sql functions library to perform operations such as mean, minimum, maximum, and count. Utilizing group by and aggregate, I was able to determine basic information about the workout, the heart rate range during workout, and the intensity category of the workout type. The results of the various queries performed are documented in Figure 3-5 with their respective lines of code. One of the biggest obstacles was that the results showed a mix of integers and floats. I had tried to lookup away to make the table reporting cleaner, but when I tried to change the data type in the schema it just removed the data since it didn't fit the pattern described (ex: float to int).

```
>>> df.groupBy("workout_type").agg(F.count("workout_type"), F.mean("calories")
,F.sum("distance"),F.sum("total_steps")).show()
+--------------+-------------------+-----------------+------------------+----------------+
|  workout_type|count(workout_type)|    avg(calories)|     sum(distance)|sum(total_steps)|
+--------------+-------------------+-----------------+------------------+----------------+
|    Open Water|                 91|296.7472527472527|3|471.44000000000017|            null|
|       Walking|                 98| 276.0408163265306| 550.0400000000002|        153958.0|
|     Trail Run|                 90|267.96666666666664|            463.32|        251100.0|
|     Freestyle|                 96| 278.5520833333333| 484.2999999999999|            null|
|      Trekking|                 94| 283.1276595744681|            496.19|        472632.0|
| Indoor Cycling|                 80|           280.45|420.72000000000014|            null|
|Outdoor Running|                 81| 301.4691358024691|408.88999999999993|        617058.0|
|       Cricket|                 93| 307.5483870967742|491.39000000000016|        320292.0|
|     Treadmill|                 98|278.14285714285717|489.98999999999995|        445900.0|
|  Pool Swimming|                 94| 283.4148936170213|             504.6|            null|
|Outdoor Cycling|                 85| 299.1294117647059| 462.7399999999999|            null|
+--------------+-------------------+-----------------+------------------+----------------+
```

Figure 3 Workout Summary

The results of the query in Figure 3 show that walking and treadmill were the most utilized workout types, but the top calorie burning workouts were cricket and outdoor running.

```
>>> df.groupBy("workout_type").agg(F.min("min_heart_rate"),F.mean("avg_heart_r
ate"),F.max("max_heart_rate")).show()
+--------------+-------------------+-------------------+-------------------+
|  workout_type|min(min_heart_rate)|avg(avg_heart_rate)|max(max_heart_rate)|
+--------------+-------------------+-------------------+-------------------+
|    Open Water|                 80|  130.7032967032967|              166.0|
|       Walking|                 80|   90.31122448979592|               85.0|
|     Trail Run|                 81| 141.98333333333332|              188.0|
|     Freestyle|                 80| 109.44791666666667|              122.0|
|      Trekking|                 80| 110.35106382978724|              124.0|
| Indoor Cycling|                 80|               92.4|               90.0|
|Outdoor Running|                 80| 143.57407407407408|              190.0|
|       Cricket|                 80|  93.51075268817205|               92.0|
|     Treadmill|                 80| 100.29591836734694|              105.0|
|  Pool Swimming|                 80| 135.41489361702128|              175.0|
|Outdoor Cycling|                 80| 136.50588235294117|              177.0|
+--------------+-------------------+-------------------+-------------------+
```

Figure 4 Workout Heart Rate

In Figure 4, I compared the heart rate range between the various workout types. The results showed anomalies with walking, indoor cycling, and cricket. After further analysis, I discovered that the data captured for max heart rate was entered as one number across all records in the anomalies. For the purposes of analysis, this will carry less weight than other features.

```
>>> df.groupBy("workout_type").agg(F.mean("aerobic"),F.mean("anaerobic"),F.mea
n("intensive"),F.mean("light")).show()
+---------------+------------+--------------+------------------+------------------+
|   workout_type|avg(aerobic)|avg(anaerobic)|    avg(intensive)|        avg(light)|
+---------------+------------+--------------+------------------+------------------+
|     Open Water|         0.0|           5.0| 38.15384615384615|50.252747252747255|
|        Walking|        22.0|           0.0|34.785714285714285| 45.11224489795919|
|      Trail Run|         0.0|          80.0| 38.08888888888889| 50.98888888888889|
|      Freestyle|        28.0|           2.0|37.635416666666664|           49.5625|
|       Trekking|        50.0|           0.0| 36.02127659574468|53.329787234042556|
| Indoor Cycling|        32.0|           0.0|              39.6|              51.1|
|Outdoor Running|         0.0|          71.0|32.888888888888886|55.160493827160494|
|        Cricket|        40.0|          10.0|36.924731182795696|  58.8494623655914|
|      Treadmill|        26.0|           7.0|37.826530612244895| 45.87755102040816|
|  Pool Swimming|        20.0|          15.0| 33.08510638297872| 51.93617021276596|
|Outdoor Cycling|        45.0|          15.0|33.470588235294116|49.247058823529414|
+---------------+------------+--------------+------------------+------------------+
```

Figure 5 Workout Intensity

The results from Figure 5 show an average of the category percentage that each workout type had assigned per session. Instead of utilizing average, I wanted to apply a ranking system to see which category best fits the workout type to get a baseline for the intensity. I tried out a method using window and rank as suggested by a stack overflow post, but I was unsuccessful (Group By, Rank and aggregate spark data frame using pyspark, n.d.) as shown in Figure 6.

```
df.select("workout_type","light").withColumn("dense_rank",dense_rank().over(windowSpec)).show()
```

Figure 6 Workout Intensity Rank

The final tool that was used during analysis was Solr. According to heart.org, a healthy 35-year-old should target a heart rate of 93-157 bpm with a maximum of 185bpm (Target Heart Rates Chart, 2021). This reference will be used as a baseline to evaluate the person's workout routine. The average and max heart rates were filtered to meet the criteria. The top workout descriptions that met the criteria were cycling, freestyle, pool, swimming, and treadmill as shown in Figure 7.

```
"facet_counts":{
  "facet_queries":{ },
  "facet_fields":{
    "workout_type":["cycling",122,"freestyle",96,"pool",94,"swimming",94,"treadmill",94,"trekking",94,"open",91,"water",91,"outdoor",85]
  },
  "facet_ranges":{ },
  "facet_intervals":{ },
  "facet_heatmaps":{ }
}
```

Figure 7 Workout Facet Counts

The top 10 calorie burn entries were also recorded with the filters for target and max heart rate range. The top workout types were open water, trekking, indoor cycling, freestyle, outdoor cycling, and treadmill. This is shown in Figure 8. Compared to the frequent workout types, there appears to be diversity in the types that will achieve the results wanted.

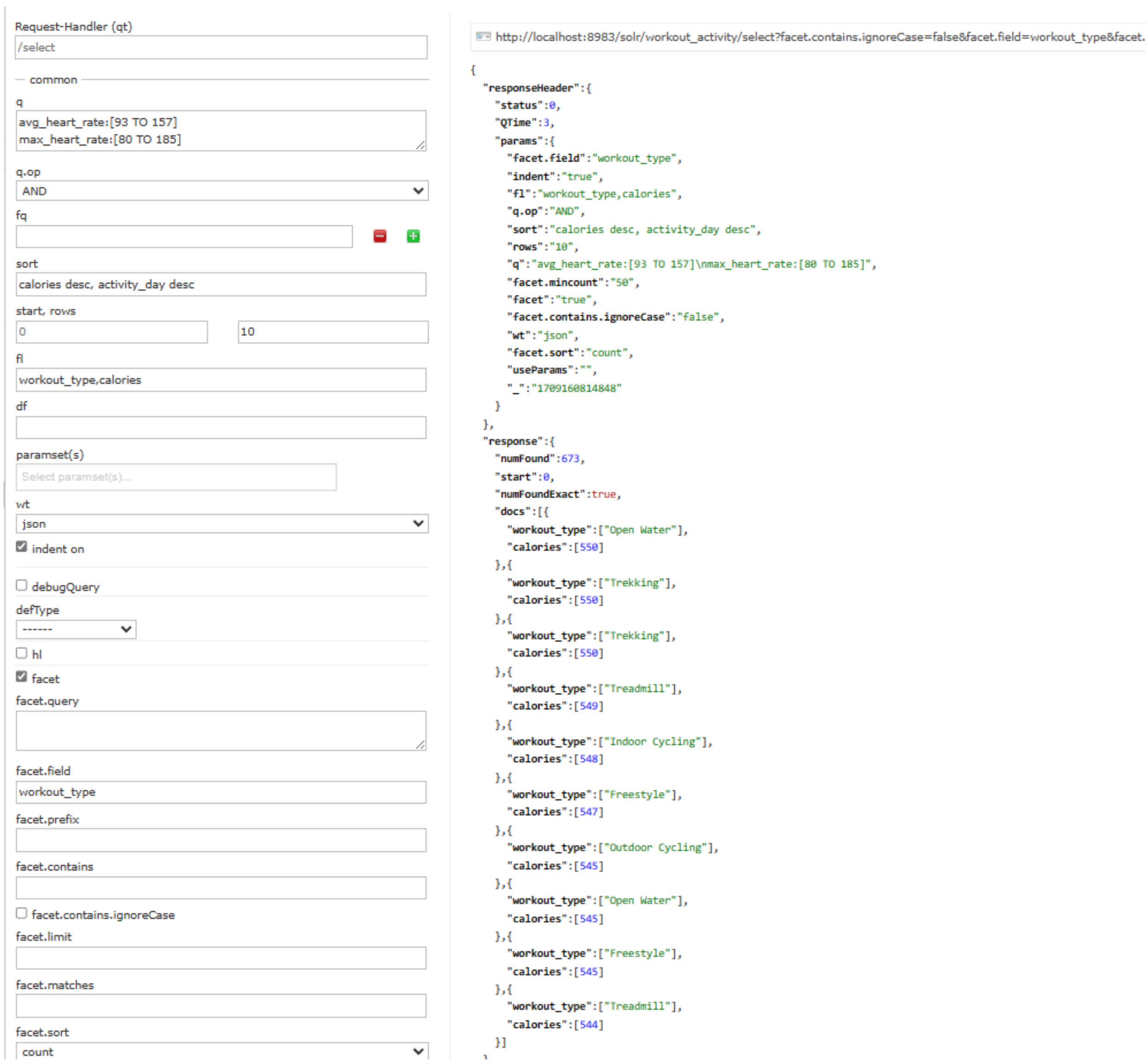

Figure 8 Workout Solr Results

In conclusion, it appears that the person's workout routines should primarily focus on cycling, freestyle, pool swimming, treadmill, and trekking. The big data tools (HDFS, NiFi, Spark, and Solr) provided the framework to explore the data and organize the flow. Next steps would be to build a machine learning model to provide a personalized workout recommendation for multiple users.

References

Group By, Rank and aggregate spark data frame using pyspark. (n.d.). Retrieved from Stack Overflow: https://stackoverflow.com/questions/41661068/group-by-rank-and-aggregate-spark-data-frame-using-pyspark

Sharma, T. (2022). *Redmi Fuel Band Record Tracker (Fitbit Dataset)*. Retrieved from Kaggle: https://www.kaggle.com/datasets/tanisha1416/my-redmi-fuel-band-record-tracker-fitbit-dataset?rvi=1

Target Heart Rates Chart. (2021). Retrieved from Heart: https://www.heart.org/en/healthy-living/fitness/fitness-basics/target-heart-rates