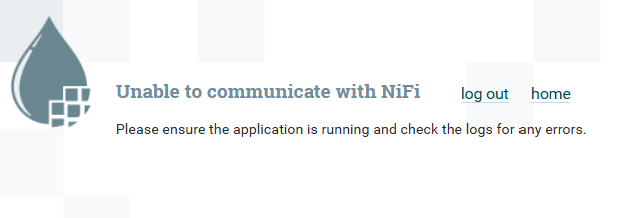
one of the issues encountered from the beginning is recalling all the resources used in perviouse assignments

Byreopening and starting all docker to get Nifi to load.

For some reason Kafka stopped and made Nifi shut down as well

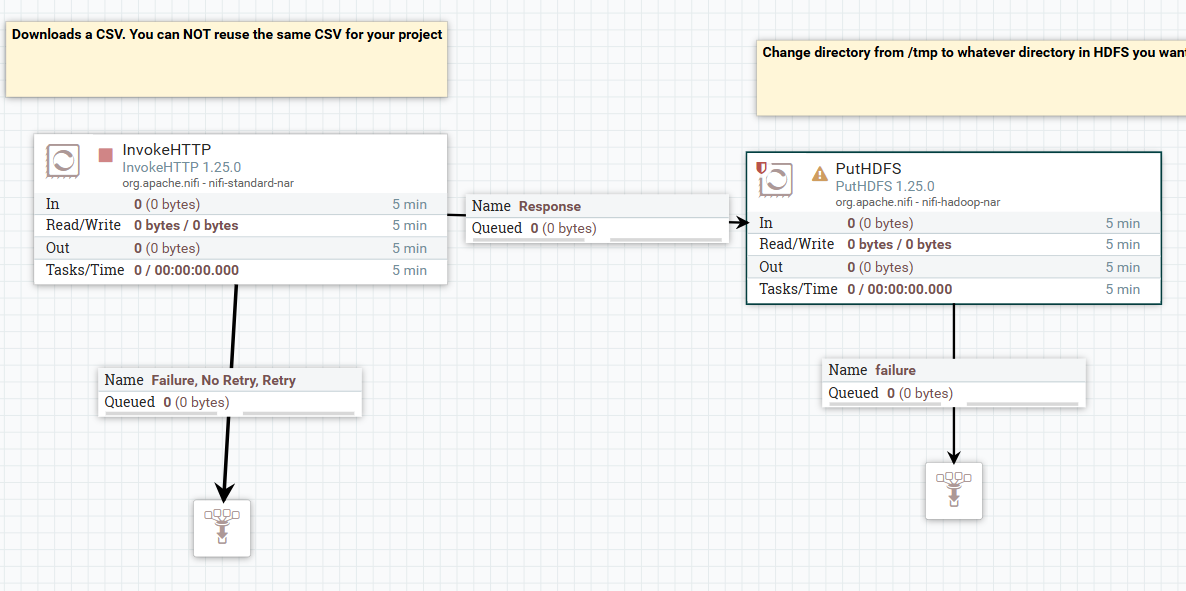


A computer screen shot of a program

Description automatically generated

I followed the same approach and used a straightforward dataset with just 100 rows. The aim is to ensure that all components are functioning properly and to note any challenges that arise.

First challenge



Create a new directory to in hdfs and change its permission to write to it from NiFi

A screen shot of a computer

Description automatically generated

Keeping track of resources used in previous assignments can indeed be challenging. Here are some strategies and tools to help you better organize and recall resources:

Once we got everything up and running we had a vew of our data running

A close up of a screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black background with many small letters

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

In Apache NiFi, the "InvokeHTTP" processor is used to make HTTP requests to an external web service or API. It allows NiFi to interact with external HTTP/HTTPS endpoints, which is useful for integrating with REST APIs, web services, or other HTTP-based resources as part of a data flow.

Redid everything with new dat set caleed mobile.csv

A screenshot of a computer

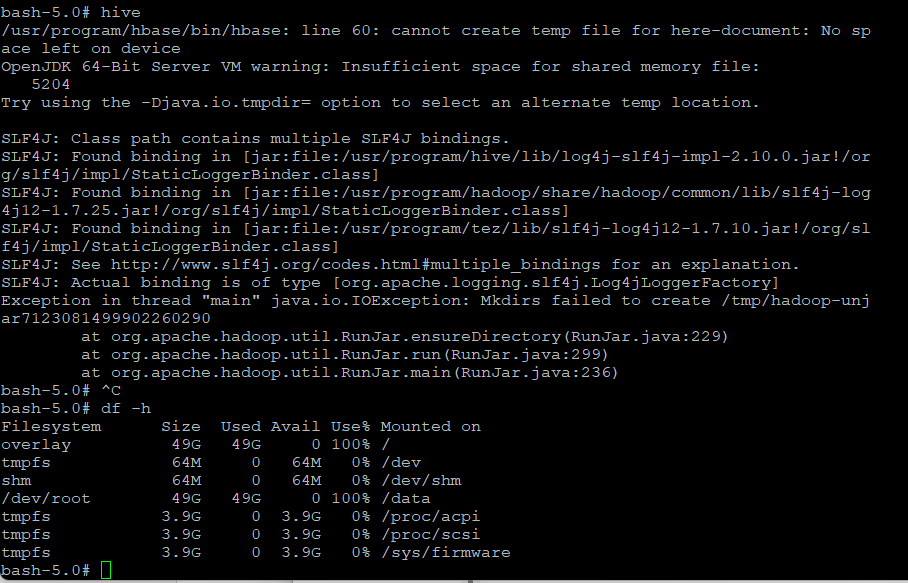
Description automatically generated

Chowing our data in hdfs under project directory

A screen shot of a computer code

Description automatically generated

When starting the hive sessin I encountered a memoryerror, where I was running ouf of space in the tem directory

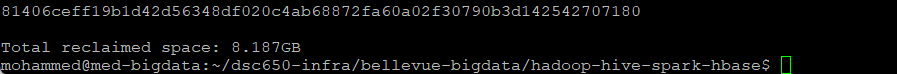


Now I cant even load the containers A screenshot of a computer screen

Description automatically generated

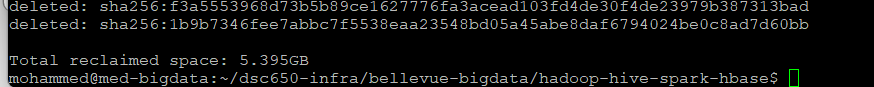
After some research I had to remove unused Volumes to free up space, this permitted to regain 8gb od space

docker volume prune -f



I’ve also removed unused networks

1. docker system prune -a --volumes -f



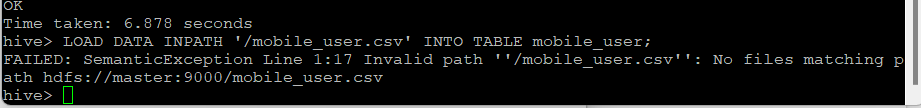
I<ve used the command line sudo find / -type f -size +100M 2>/dev/null to find the large log files and delet them all using

Creating Hive table

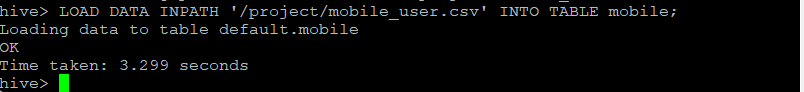
A screen shot of a computer program

Description automatically generated

Can not find path



Loading the data to the table in hive



LOAD DATA INPATH '/project/mobile\_user.csv ' INTO TABLE mobile;

A number on a black background

Description automatically generated

Table with 700 rows

A black screen with white text

Description automatically generated

Moving to pyspark

A computer screen with white text

Description automatically generated

Changing names

A computer code on a black background

Description automatically generated

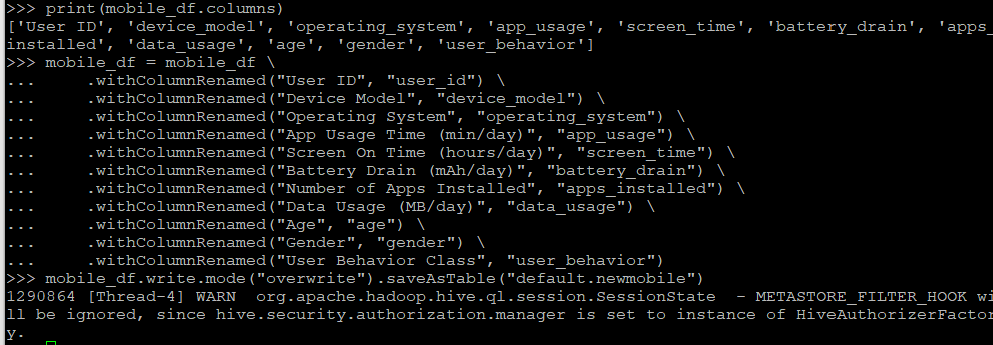
Writing the Dataframe to a new Have table

Running into issues

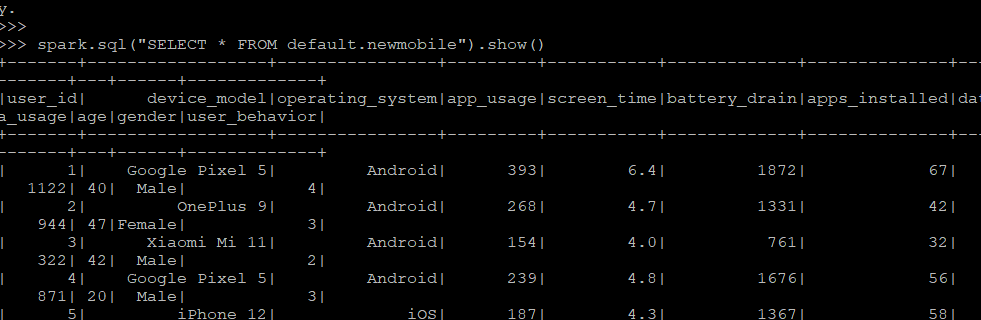
A screen shot of a computer code

Description automatically generated

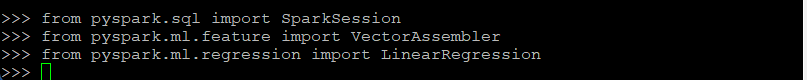
Successfully renaming the data into new hive table



Showing the data with new header



After installing NumPy importing for machine learning in Pyspark



I did run into datatype issues while applying machine learning algorithms in spark.

I had this error: IllegalArgumentException: Data type string of column ... is not supported, this typically occurs when PySpark's machine learning (ML) transformations encounter columns with unsupported data types

I had to change datatypes using the following code

from pyspark.sql import functions as F

mobile\_df = mobile\_df \ .withColumn("screen\_time", F.col("screen\_time").cast("double")) \ .withColumn("app\_usage", F.col("app\_usage").cast("double")) \ .withColumn("battery\_drain", F.col("battery\_drain").cast("double")) \ .withColumn("apps\_installed", F.col("apps\_installed").cast("double"))

making sure we have the right data types

A computer screen with white text

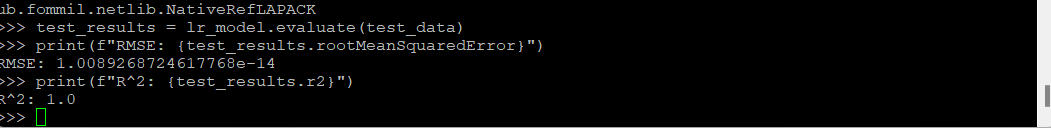
Description automatically generated

Starting ML by dividing data randomly

A computer screen with text on it

Description automatically generated

the model performance metrics



Explanation :

n this case, the RMSE is approximately 1.0089268724617768e-14, which is a very small number (almost zero). This implies that the predictions are extremely close to the actual values in the test data. A near-zero RMSE often indicates either:

* The model is highly accurate in predicting this particular test dataset.
* There might be data leakage or a trivial relationship between the features and the target variable in the test set.

Here, R^2 is 1.0, which indicates a perfect fit. This suggests that the model explains 100% of the variance in the test data.

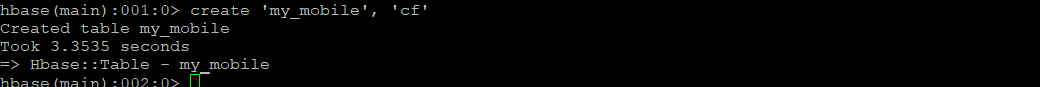
he combination of an extremely low RMSE and an R² of 1.0 suggests that the model is fitting the test data almost perfectly. This might be ideal in some contexts, but it could also indicate potential issues such as:

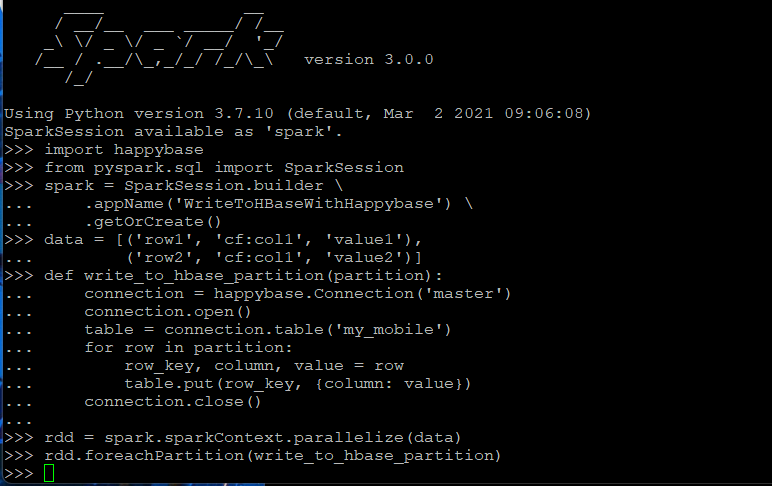
* **Overfitting**: If the model performs much worse on other datasets, it may be overfitted to the specific data used for training.
* **Data Leakage**: If test data was unintentionally used in training, the model might be "cheating" by memorizing rather than generalizing.

To validate the model’s generalizability, test it on a completely new dataset to confirm that it performs consistently well.

Saving data in HBase:

Creating a table in HBase



to write data to HBase inside each partition

HBase tableexit

A computer screen with white text

Description automatically generated

A screen shot of a computer program

Description automatically generated

Showing table in Hbase

A screen shot of a computer

Description automatically generated

**1. Documentation and Notes**

* **Create a Personal Wiki**: Use tools like [Notion](https://www.notion.so), [Obsidian](https://obsidian.md/), or even a simple text editor to maintain a personal wiki or knowledge base where you log all resources and notes.
* **Markdown Files**: Maintain a README.md file in each project directory, listing all resources, commands, and steps used.

**2. Version Control (Git)**

* **Commit Messages**: Use meaningful commit messages to document changes and resources.