MSIN0166: Data Engineering Group Assignment

Group 5

Due: 12 April 2022 10:00am

1.0 Introduction

Data engineering, as the practice of designing and building systems for collecting, storing, and analyzing data at scale (Coursera, 2021), is a field that values the ability of data sourcing, processing and infrastructure construction. In the past few decades, the information explosion brought by the tide of digitization enabled us to access data with higher volume and a larger variety in a more efficient and organized method. In this report, we aimed to develop the aforementioned data engineering skills by gathering new datasets from various online sources and applying different data processing methods to them.

Earlier this year, with the quadrennial Winter Olympics taking place in Beijing, winter sports have again drawn great attention around the world. Among all the events, short track speed skating (STK) has won the favour of many spectators with its exciting atmosphere. As a racing sport, time data is naturally vital in STK. But meanwhile, due to the frequent occurrence of accidents in this game, penalties and advances by referees also play an important role in the course of the games. Therefore, we believe that insights from previous Olympics STK games can be very valuable for national teams in their training strategy.

In order to achieve our research objectives, we gathered both the game results and athlete data of the STK games in the 2022 Beijing Winter Olympics from relevant authorities. Through data processing and pipeline creating, we were able to construct a basic querying system for STK teams to refer to in their future training, which allows the users to gain insights and experience from historical data. Moreover, apart from the quantitative data, we also built a model that can collect recent Twitter posts on STK and perform sentiment analysis models as a supplement from a different perspective.

In [1]:
```
# import the packages
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
import re
import os
import requests
from datetime import datetime
from IPython.display import Image

from dvc.api import make_checkpoint
```

2.0 Structure of the project

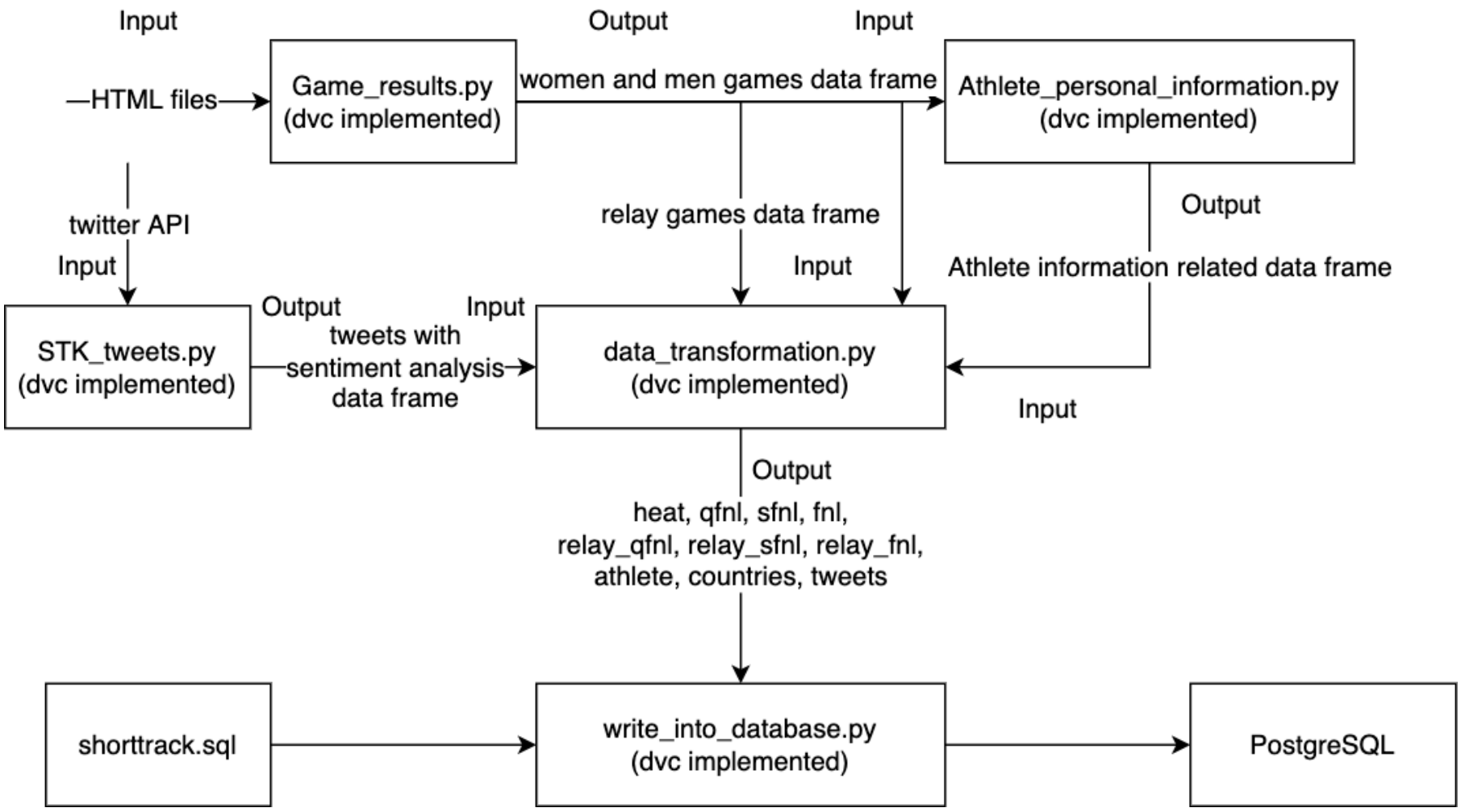
The graph shown below is the flow chart of how each file is connected and their input and output information. As this process needs to be reusable, we tried our best to reduce the coupling in the code. Only athlete information is dependent on the other files output (names of the athletes) in the stages before data transformation.

In [2]:
```

```

```
Image("/project/DataEngineering/graphs/structure.png", width = 900)
```

Out[2]:



2.1 Data Storage

To make data more tractable and manageable, we have stored our data in a uniform format using parquet. The datasets gathered from the three different sources above are converted and stored in parquet format after collection and some basic data cleaning. Parquet is an open-source file format and a column-based storage type that is useful for dealing with big datasets since it allows us to query columns quickly. Due to the nature of our large volume of data, this data storage format improves efficiency as well as reduces storage requirements, including scan and deserialization time and financial cost. The free of charge nature of the parquet storage format would be an essential consideration for both individuals and organizations when working with large volume datasets.

As data will continue flowing in for our data analysis project, we designed a structured schema for our database aiming for efficient operation and maintenance. Our designed schema contains ten tables, with 7 different race tables, athlete information, country information and a Twitter data table. It allows us to extract information effectively. For example, if we want to quickly see the performance of a female athlete in terms of her game performance and her demographic information, we can simply look through our “athlete” table and see what games are suitable for her and what kind of games she tends to lose so that coaches are able to analyze her skill gaps. In addition, the age and the clubs are also important for club owners to find the next sports star and for coaches to make better training plans. We have also visually presented the schema using a DB diagram, showing the architecture of our designed database for better data management.

2.2 Source Version Control

While we are collaborating on the group project, source control allows us to keep track of the changes team members are making to the codes. Meanwhile, source control management systems record the history of the evolution of code which also benefits resolving merging conflicts. As Git is an open-source tool for the SCM systems, Git and Github will be used during the process of our group coursework. Each team member worked independently on our own branch, then we pushed the commits, made a pull request and finally merged it to the main branch. If conflicts arise at the merging stage, we would resolve them together and one of us would push a new commit.

Link of the Github repository: https://github.com/Haiyun-Zou/DataEngineering

2.3 Automate Terraform with Github Actions

In order to perform best practices and improve better collaborations, Terraform has been set up with Github actions and the Terraform CLI has been configured to automate the workflow. In the terraform.yml file, Terraform Format check is used to prevent anyone in the team from merging improperly formatted configuration to the main branch. Terraform Plan generates a plan when there is a pull request. When the main branch is updated, the workflow will apply the configuration.

In [3]:
```
Image("/project/DataEngineering/graphs/tf.png", width = 400)
```

Out[3]:

```yaml
name: "Terraform"

on:
  push:
    branches:
      - main
  pull_request:

jobs:
  terraform:
    name: "Terraform"
    runs-on: ubuntu-latest
    environment: myenvironment
    steps:
      - name: Checkout
        uses: actions/checkout@v2

      - name: Setup Terraform
        uses: hashicorp/setup-terraform@v1
        with:
          # terraform_version: 0.13.0:
          cli_config_credentials_token: ${{ secrets.TERRAFORM_API_TOKEN }}

      - name: Terraform Format
        id: fmt
        run: terraform fmt -check

      - name: Terraform Init
        id: init
        run: terraform init

      - name: Terraform Validate
        id: validate
        run: terraform validate -no-color

      - name: Terraform Plan
        id: plan
        if: github.event_name == 'pull_request'
        run: terraform plan -no-color
        continue-on-error: true

      - uses: actions/github-script@0.9.0
        if: github.event_name == 'pull_request'
        env:
          PLAN: "terraform\n${{ steps.plan.outputs.stdout }}"
        with:
          github-token: ${{ secrets.GITHUB_TOKEN }}
          script: |
            const output = `#### Terraform Format and Style 🖌\`${{ steps.fmt.outcome }}\`
            #### Terraform Initialization ⚙\`${{ steps.init.outcome }}\`
            #### Terraform Validation 🤖\`${{ steps.validate.outcome }}\`
            #### Terraform Plan 📖\`${{ steps.plan.outcome }}\`
            <details><summary>Show Plan</summary>
            \`\`\`\n
            ${process.env.PLAN}
            \`\`\`
            </details>
            *Pusher: @${{ github.actor }}, Action: \`${{ github.event_name }}\`*;
            github.issues.createComment({
              issue_number: context.issue.number,
              owner: context.repo.owner,
              repo: context.repo.repo,
              body: output
            })
      - name: Terraform Plan Status
        if: steps.plan.outcome == 'failure'
        run: exit 1

      - name: Terraform Apply
        if: github.ref == 'refs/heads/main' && github.event_name == 'push'
        run: terraform apply -auto-approve
```

2.4 Data Lineage

Data Lineage shows the complete flow of the data from the sources to the transformation and the consumption stage. Data Lineage is a tool ensuring the accuracy and consistency of the data, from origins to the endpoint. It allows us to confirm whether it comes from a trustworthy source, whether correct data transformation has been applied and whether it has been loaded to the right place. It is particularly useful when machine learning has been deployed. When the data lineage of a dataset is known, reproducibility would be achieved an easier way.

Data Version Control(DVC) is used in this group assignment. DVC normally is for tracking machine learning experiments data and versions, in this assignment, it has been used to track the version of data frames in each stage. The auto script for running the complete process is also implemented based on the DVC actions. (shown in the two cells below)

In [4]:
```
Image("/project/DataEngineering/graphs/dvc.png", width = 500)
```

Out[4]:

```
stages:
  game_results_checks:
    cmd: python /project/DataEngineering/python_files/Game_results.py
    outs:
      - game_results.txt:
          checkpoint: true

  Athlete_personal_information_checks:
    cmd: python /project/DataEngineering/python_files/Athlete_personal_information.py
    outs:
      - Athlete_personal_information.txt:
          checkpoint: true

  STK_tweets_checks:
    cmd: python /project/DataEngineering/python_files/STK_tweets.py
    outs:
      - STK_tweets.txt:
          checkpoint: true

  data_transformation_checks:
    cmd: python /project/DataEngineering/python_files/data_transformation.py
    outs:
      - data_transformation.txt:
          checkpoint: true

  write_into_database_checks:
    cmd: python /project/DataEngineering/python_files/write_into_database.py
    outs:
      - write_into_database.txt:
          checkpoint: true
```

In [5]:
```
Image("/project/DataEngineering/graphs/auto.png", width = 400)
```

Out[5]:
```
bash install_spark.sh
pip install vaderSentiment
pip install dvc


dvc exp run -f game_results_checks
dvc exp run -f Athlete_personal_information_checks
dvc exp run -f STK_tweets_checks
dvc exp run -f data_transformation_checks
dvc exp run -f write_into_database_checks

dvc exp show
```

In terms of data processing, Apache Spark has been installed at the very beginning to utilize in-memory caching, and optimize query execution.

In [6]:
```
!bash install_spark.sh
```

```
  .::::.      .::.
 ....yy:     .yy.
 :.  .yy.     y.
      :y:    .:
      .yy   .:
       yy...:
       :y:..
       .y.
        .
           ...
         .....
         ::::.


• Project files and data should be stored in /project. This is shared among everyone
  in the project.
• Personal files and configuration should be stored in /home/faculty.
• Files outside /project and /home/faculty will be lost when this server is terminated.
• Create custom environments to setup your servers reproducibly.

Hit:1 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://eu-west-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:5 https://packages.cloud.google.com/apt cloud-sdk InRelease
Fetched 252 kB in 0s (654 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  python3-crcmod
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk-headless is already the newest version (8u312-b07-0ubuntu1~18.04).
```

```
The following package was automatically installed and is no longer required:
  python3-crcmod
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
/project /project/DataEngineering
spark-3.2.1-bin-hadoop3.2/
spark-3.2.1-bin-hadoop3.2/LICENSE
spark-3.2.1-bin-hadoop3.2/NOTICE
spark-3.2.1-bin-hadoop3.2/R/
spark-3.2.1-bin-hadoop3.2/R/lib/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/DESCRIPTION
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/INDEX
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/Rd.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/features.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/hsearch.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/links.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/nsInfo.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/package.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/Meta/vignette.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/NAMESPACE
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/R/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/R/SparkR
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/R/SparkR.rdb
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/R/SparkR.rdx
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/doc/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/doc/index.html
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.R
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.Rmd
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.html
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/AnIndex
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/SparkR.rdb
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/SparkR.rdx
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/aliases.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/help/paths.rds
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/html/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/html/00Index.html
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/html/R.css
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/profile/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/profile/general.R
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/profile/shell.R
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/tests/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/tests/testthat/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/tests/testthat/test_basic.R
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/worker/
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/worker/daemon.R
spark-3.2.1-bin-hadoop3.2/R/lib/SparkR/worker/worker.R
spark-3.2.1-bin-hadoop3.2/R/lib/sparkr.zip
spark-3.2.1-bin-hadoop3.2/README.md
spark-3.2.1-bin-hadoop3.2/RELEASE
spark-3.2.1-bin-hadoop3.2/bin/
spark-3.2.1-bin-hadoop3.2/bin/beeline
spark-3.2.1-bin-hadoop3.2/bin/beeline.cmd
spark-3.2.1-bin-hadoop3.2/bin/docker-image-tool.sh
spark-3.2.1-bin-hadoop3.2/bin/find-spark-home
spark-3.2.1-bin-hadoop3.2/bin/find-spark-home.cmd
spark-3.2.1-bin-hadoop3.2/bin/load-spark-env.cmd
spark-3.2.1-bin-hadoop3.2/bin/load-spark-env.sh
spark-3.2.1-bin-hadoop3.2/bin/pyspark
spark-3.2.1-bin-hadoop3.2/bin/pyspark.cmd
spark-3.2.1-bin-hadoop3.2/bin/pyspark2.cmd
spark-3.2.1-bin-hadoop3.2/bin/run-example
spark-3.2.1-bin-hadoop3.2/bin/run-example.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-class
spark-3.2.1-bin-hadoop3.2/bin/spark-class.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-class2.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-shell
spark-3.2.1-bin-hadoop3.2/bin/spark-shell.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-shell2.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-sql
spark-3.2.1-bin-hadoop3.2/bin/spark-sql.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-sql2.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-submit
spark-3.2.1-bin-hadoop3.2/bin/spark-submit.cmd
spark-3.2.1-bin-hadoop3.2/bin/spark-submit2.cmd
spark-3.2.1-bin-hadoop3.2/bin/sparkR
spark-3.2.1-bin-hadoop3.2/bin/sparkR.cmd
spark-3.2.1-bin-hadoop3.2/bin/sparkR2.cmd
spark-3.2.1-bin-hadoop3.2/conf/
spark-3.2.1-bin-hadoop3.2/conf/fairscheduler.xml.template
spark-3.2.1-bin-hadoop3.2/conf/log4j.properties.template
spark-3.2.1-bin-hadoop3.2/conf/metrics.properties.template
spark-3.2.1-bin-hadoop3.2/conf/spark-defaults.conf.template
spark-3.2.1-bin-hadoop3.2/conf/spark-env.sh.template
spark-3.2.1-bin-hadoop3.2/conf/workers.template
spark-3.2.1-bin-hadoop3.2/data/
spark-3.2.1-bin-hadoop3.2/data/graphx/
```

spark-3.2.1-bin-hadoop3.2/data/graphx/followers.txt
spark-3.2.1-bin-hadoop3.2/data/graphx/users.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/
spark-3.2.1-bin-hadoop3.2/data/mllib/als/
spark-3.2.1-bin-hadoop3.2/data/mllib/als/sample_movielens_ratings.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/als/test.data
spark-3.2.1-bin-hadoop3.2/data/mllib/gmm_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/images/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/license.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/29.5.a_b_EGDP022204.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/54893.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/DP153539.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/DP802813.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/kittens/not-image.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/license.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/multi-channel/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/multi-channel/BGRA.png
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/multi-channel/BGRA_alpha_60.png
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/multi-channel/chr30.4.184.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/origin/multi-channel/grayscale.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-01/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-01/29.5.a_b_EGDP022204.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-01/not-image.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-02/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-02/54893.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-02/DP153539.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=kittens/date=2018-02/DP802813.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-01/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-01/BGRA.png
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-01/BGRA_alpha_60.png
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-02/
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-02/chr30.4.184.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/images/partitioned/cls=multichannel/date=2018-02/grayscale.jpg
spark-3.2.1-bin-hadoop3.2/data/mllib/iris_libsvm.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/kmeans_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/pagerank_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/pic_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/ridge-data/
spark-3.2.1-bin-hadoop3.2/data/mllib/ridge-data/lpsa.data
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_binary_classification_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_fpgrowth.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_isotonic_regression_libsvm_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_kmeans_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_lda_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_lda_libsvm_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_libsvm_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_linear_regression_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_movielens_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_multiclass_classification_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/sample_svm_data.txt
spark-3.2.1-bin-hadoop3.2/data/mllib/streaming_kmeans_data_test.txt
spark-3.2.1-bin-hadoop3.2/data/streaming/
spark-3.2.1-bin-hadoop3.2/data/streaming/AFINN-111.txt
spark-3.2.1-bin-hadoop3.2/examples/
spark-3.2.1-bin-hadoop3.2/examples/jars/
spark-3.2.1-bin-hadoop3.2/examples/jars/scopt_2.12-3.7.1.jar
spark-3.2.1-bin-hadoop3.2/examples/jars/spark-examples_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/examples/src/
spark-3.2.1-bin-hadoop3.2/examples/src/main/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaHdfsLR.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaLogQuery.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaPageRank.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaSparkPi.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaStatusTrackerDemo.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaTC.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/JavaWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaAFTSurvivalRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaALSExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaBinarizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaBisectingKMeansExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaBucketedRandomProjectionLSHExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaBucketizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaChiSqSelectorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaChiSquareTestExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaCorrelationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaCountVectorizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaDCTExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaDecisionTreeClassificationExample.java

spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaDecisionTreeRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaDocument.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaElementwiseProductExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaEstimatorTransformerParamExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaFMClassifierExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaFMRegressorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaFPGrowthExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaFeatureHasherExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaGaussianMixtureExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaGeneralizedLinearRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaGradientBoostedTreeClassifierExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaGradientBoostedTreeRegressorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaImputerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaIndexToStringExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaInteractionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaIsotonicRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaKMeansExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLDAExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLabeledDocument.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLinearRegressionWithElasticNetExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLinearSVCExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLogisticRegressionSummaryExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaLogisticRegressionWithElasticNetExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaMaxAbsScalerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaMinHashLSHExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaMinMaxScalerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaModelSelectionViaCrossValidationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaModelSelectionViaTrainValidationSplitExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaMulticlassLogisticRegressionWithElasticNetExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaMultilayerPerceptronClassifierExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaNGramExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaNaiveBayesExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaNormalizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaOneHotEncoderExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaOneVsRestExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaPCAExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaPipelineExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaPolynomialExpansionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaPowerIterationClusteringExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaPrefixSpanExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaQuantileDiscretizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaRFormulaExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaRandomForestClassifierExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaRandomForestRegressorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaRobustScalerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaSQLTransformerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaStandardScalerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaStopWordsRemoverExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaStringIndexerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaSummarizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaTfIdfExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaTokenizerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaUnivariateFeatureSelectorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaVarianceThresholdSelectorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaVectorAssemblerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaVectorIndexerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaVectorSizeHintExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaVectorSlicerExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/ml/JavaWord2VecExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaALS.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaAssociationRulesExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaBinaryClassificationMetricsExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaBisectingKMeansExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaChiSqSelectorExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaCorrelationsExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaDecisionTreeClassificationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaDecisionTreeRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaElementwiseProductExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaGaussianMixtureExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaGradientBoostingClassificationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaGradientBoostingRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaHypothesisTestingExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaHypothesisTestingKolmogorovSmirnovTestExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaIsotonicRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaKMeansExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaKernelDensityEstimationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaLBFGSExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaLatentDirichletAllocationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaLogisticRegressionWithLBFGSExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaMultiLabelClassificationMetricsExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaMulticlassClassificationMetricsExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaNaiveBayesExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaPCAExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaPowerIterationClusteringExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaPrefixSpanExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaRandomForestClassificationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaRandomForestRegressionExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaRankingMetricsExample.java

spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaRecommendationExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaSVDExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaSVMWithSGDExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaSimpleFPGrowth.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaStratifiedSamplingExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaStreamingTestExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/mllib/JavaSummaryStatisticsExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/JavaSQLDataSourceExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/JavaSparkSQLExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/JavaUserDefinedScalar.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/JavaUserDefinedTypedAggregation.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/JavaUserDefinedUntypedAggregation.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/hive/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/hive/JavaSparkHiveExample.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredComplexSessionization.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredKafkaWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredKerberizedKafkaWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredNetworkWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredNetworkWordCountWindowed.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/sql/streaming/JavaStructuredSessionization.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaCustomReceiver.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaDirectKafkaWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaDirectKerberizedKafkaWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaNetworkWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaQueueStream.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaRecord.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaRecoverableNetworkWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaSqlNetworkWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/java/org/apache/spark/examples/streaming/JavaStatefulNetworkWordCount.java
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/als.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/avro_inputformat.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/kmeans.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/logistic_regression.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/aft_survival_regression.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/als_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/binarizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/bisecting_k_means_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/bucketed_random_projection_lsh_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/bucketizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/chi_square_test_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/chisq_selector_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/correlation_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/count_vectorizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/cross_validator.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/dataframe_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/dct_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/decision_tree_classification_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/decision_tree_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/elementwise_product_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/estimator_transformer_param_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/feature_hasher_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/fm_classifier_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/fm_regressor_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/fpgrowth_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/gaussian_mixture_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/generalized_linear_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/gradient_boosted_tree_classifier_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/gradient_boosted_tree_regressor_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/imputer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/index_to_string_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/interaction_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/isotonic_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/kmeans_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/lda_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/linear_regression_with_elastic_net.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/linearsvc.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/logistic_regression_summary_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/logistic_regression_with_elastic_net.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/max_abs_scaler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/min_hash_lsh_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/min_max_scaler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/multiclass_logistic_regression_with_elastic_net.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/multilayer_perceptron_classification.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/n_gram_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/naive_bayes_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/normalizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/one_vs_rest_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/onehot_encoder_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/pca_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/pipeline_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/polynomial_expansion_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/power_iteration_clustering_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/prefixspan_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/quantile_discretizer_example.py

spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/random_forest_classifier_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/random_forest_regressor_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/rformula_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/robust_scaler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/sql_transformer.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/standard_scaler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/stopwords_remover_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/string_indexer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/summarizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/tf_idf_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/tokenizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/train_validation_split.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/univariate_feature_selector_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/variance_threshold_selector_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/vector_assembler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/vector_indexer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/vector_size_hint_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/vector_slicer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/ml/word2vec_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/binary_classification_metrics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/bisecting_k_means_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/correlations.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/correlations_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/decision_tree_classification_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/decision_tree_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/elementwise_product_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/fpgrowth_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/gaussian_mixture_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/gaussian_mixture_model.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/gradient_boosting_classification_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/gradient_boosting_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/hypothesis_testing_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/hypothesis_testing_kolmogorov_smirnov_test_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/isotonic_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/k_means_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/kernel_density_estimation_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/kmeans.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/latent_dirichlet_allocation_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/linear_regression_with_sgd_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/logistic_regression.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/logistic_regression_with_lbfgs_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/multi_class_metrics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/multi_label_metrics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/naive_bayes_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/normalizer_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/pca_rowmatrix_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/power_iteration_clustering_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/random_forest_classification_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/random_forest_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/random_rdd_generation.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/ranking_metrics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/recommendation_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/regression_metrics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/sampled_rdds.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/standard_scaler_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/stratified_sampling_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/streaming_k_means_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/streaming_linear_regression_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/summary_statistics_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/svd_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/svm_with_sgd_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/tf_idf_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/word2vec.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/mllib/word2vec_example.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/pagerank.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/parquet_inputformat.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/pi.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sort.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/arrow.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/basic.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/datasource.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/hive.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/streaming/structured_kafka_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/streaming/structured_network_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/streaming/structured_network_wordcount_windowed.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/sql/streaming/structured_sessionization.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/status_api_demo.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/hdfs_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/network_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/network_wordjoinsentiments.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/queue_stream.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/recoverable_network_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/sql_network_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/streaming/stateful_network_wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/python/transitive_closure.py

spark-3.2.1-bin-hadoop3.2/examples/src/main/python/wordcount.py
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/RSparkSQLExample.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/data-manipulation.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/dataframe.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/als.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/bisectingKmeans.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/decisionTree.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/fmClassifier.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/fmRegressor.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/fpm.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/gaussianMixture.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/gbt.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/glm.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/isoreg.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/kmeans.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/kstest.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/lda.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/lm_with_elastic_net.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/logit.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/ml.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/mlp.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/naiveBayes.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/powerIterationClustering.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/prefixSpan.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/randomForest.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/survreg.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/ml/svmLinear.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/r/streaming/structured_network_wordcount.R
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/META-INF/
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/META-INF/services/
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/META-INF/services/org.apache.spark.sql.SparkSessionExtensionsProvider
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/META-INF/services/org.apache.spark.sql.jdbc.JdbcConnectionProvider
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/dir1/
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/dir1/dir2/
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/dir1/dir2/file2.parquet
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/dir1/file1.parquet
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/dir1/file3.json
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/employees.json
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/full_user.avsc
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/kv1.txt
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/people.csv
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/people.json
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/people.txt
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/user.avsc
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/users.avro
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/users.orc
spark-3.2.1-bin-hadoop3.2/examples/src/main/resources/users.parquet
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/AccumulatorMetricsTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/BroadcastTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/DFSReadWriteTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/DriverSubmissionTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ExceptionHandlingTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/GroupByTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/HdfsTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LocalALS.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LocalFileLR.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LocalKMeans.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LocalLR.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LocalPi.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/LogQuery.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/MultiBroadcastTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SimpleSkewedGroupByTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SkewedGroupByTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkALS.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkHdfsLR.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkKMeans.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkLR.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkPageRank.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkPi.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkRemoteFileTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/SparkTC.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/extensions/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/extensions/AgeExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/extensions/SessionExtensionsWithLoader.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/extensions/SessionExtensionsWithoutLoader.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/extensions/SparkSessionExtensionsTest.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/AggregateMessagesExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/Analytics.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/ComprehensiveExample.scala

spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/ConnectedComponentsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/LiveJournalPageRank.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/PageRankExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/SSSPExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/SynthBenchmark.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/graphx/TriangleCountingExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/AFTSurvivalRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ALSExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/BinarizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/BisectingKMeansExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/BucketedRandomProjectionLSHExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/BucketizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ChiSqSelectorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ChiSquareTestExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/CorrelationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/CountVectorizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DCTExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DataFrameExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DecisionTreeClassificationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DecisionTreeExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DecisionTreeRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/DeveloperApiExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ElementwiseProductExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/EstimatorTransformerParamExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/FMClassifierExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/FMRegressorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/FPGrowthExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/FeatureHasherExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/GBTExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/GaussianMixtureExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/GeneralizedLinearRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/GradientBoostedTreeClassifierExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/GradientBoostedTreeRegressorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ImputerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/IndexToStringExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/InteractionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/IsotonicRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/KMeansExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LDAExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LinearRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LinearRegressionWithElasticNetExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LinearSVCExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LogisticRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LogisticRegressionSummaryExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/LogisticRegressionWithElasticNetExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/MaxAbsScalerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/MinHashLSHExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/MinMaxScalerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ModelSelectionViaCrossValidationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/ModelSelectionViaTrainValidationSplitExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/MulticlassLogisticRegressionWithElasticNetExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/MultilayerPerceptronClassifierExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/NGramExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/NaiveBayesExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/NormalizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/OneHotEncoderExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/OneVsRestExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/PCAExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/PipelineExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/PolynomialExpansionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/PowerIterationClusteringExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/PrefixSpanExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/QuantileDiscretizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/RFormulaExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/RandomForestClassifierExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/RandomForestExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/RandomForestRegressorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/RobustScalerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/SQLTransformerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/StandardScalerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/StopWordsRemoverExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/StringIndexerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/SummarizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/TfIdfExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/TokenizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/UnaryTransformerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/UnivariateFeatureSelectorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/VarianceThresholdSelectorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/VectorAssemblerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/VectorIndexerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/VectorSizeHintExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/VectorSlicerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/ml/Word2VecExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/AbstractParams.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/AssociationRulesExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/BinaryClassification.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/BinaryClassificationMetricsExample.scala

spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/BisectingKMeansExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/ChiSqSelectorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/Correlations.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/CorrelationsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/CosineSimilarity.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/DecisionTreeClassificationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/DecisionTreeRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/DecisionTreeRunner.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/DenseKMeans.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/ElementwiseProductExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/FPGrowthExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/GaussianMixtureExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/GradientBoostedTreesRunner.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/GradientBoostingClassificationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/GradientBoostingRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/HypothesisTestingExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/HypothesisTestingKolmogorovSmirnovTestExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/IsotonicRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/KMeansExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/KernelDensityEstimationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/LBFGSExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/LDAExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/LatentDirichletAllocationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/LogisticRegressionWithLBFGSExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/MovieLensALS.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/MultiLabelMetricsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/MulticlassMetricsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/MultivariateSummarizer.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/NaiveBayesExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/NormalizerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/PCAOnRowMatrixExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/PCAOnSourceVectorExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/PMMLModelExportExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/PowerIterationClusteringExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/PrefixSpanExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/RandomForestClassificationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/RandomForestRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/RandomRDDGeneration.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/RankingMetricsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/RecommendationExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SVDExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SVMWithSGDExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SampledRDDs.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SimpleFPGrowth.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SparseNaiveBayes.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StandardScalerExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StratifiedSamplingExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StreamingKMeansExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StreamingLinearRegressionExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StreamingLogisticRegression.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/StreamingTestExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/SummaryStatisticsExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/TFIDFExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/TallSkinnyPCA.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/TallSkinnySVD.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/mllib/Word2VecExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/pythonconverters/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/pythonconverters/AvroConverters.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/RDDRelation.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/SQLDataSourceExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/SimpleTypedAggregator.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/SparkSQLExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/UserDefinedScalar.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/UserDefinedTypedAggregation.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/UserDefinedUntypedAggregation.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/hive/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/hive/SparkHiveExample.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/jdbc/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/jdbc/ExampleJdbcConnectionProvider.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredComplexSessionization.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredKafkaWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredKerberizedKafkaWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredNetworkWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredNetworkWordCountWindowed.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredSessionization.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/CustomReceiver.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/DirectKafkaWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/DirectKerberizedKafkaWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/HdfsWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/NetworkWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/QueueStream.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/RawNetworkGrep.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/RecoverableNetworkWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/SqlNetworkWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/StatefulNetworkWordCount.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/StreamingExamples.scala

spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/clickstream/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/clickstream/PageViewGenerator.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scala/org/apache/spark/examples/streaming/clickstream/PageViewStream.scala
spark-3.2.1-bin-hadoop3.2/examples/src/main/scripts/
spark-3.2.1-bin-hadoop3.2/examples/src/main/scripts/getGpusResources.sh
spark-3.2.1-bin-hadoop3.2/jars/
spark-3.2.1-bin-hadoop3.2/jars/HikariCP-2.5.1.jar
spark-3.2.1-bin-hadoop3.2/jars/JLargeArrays-1.5.jar
spark-3.2.1-bin-hadoop3.2/jars/JTransforms-3.1.jar
spark-3.2.1-bin-hadoop3.2/jars/RoaringBitmap-0.9.0.jar
spark-3.2.1-bin-hadoop3.2/jars/ST4-4.0.4.jar
spark-3.2.1-bin-hadoop3.2/jars/activation-1.1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/aircompressor-0.21.jar
spark-3.2.1-bin-hadoop3.2/jars/algebra_2.12-2.0.1.jar
spark-3.2.1-bin-hadoop3.2/jars/annotations-17.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/antlr-runtime-3.5.2.jar
spark-3.2.1-bin-hadoop3.2/jars/antlr4-runtime-4.8.jar
spark-3.2.1-bin-hadoop3.2/jars/aopalliance-repackaged-2.6.1.jar
spark-3.2.1-bin-hadoop3.2/jars/arpack-2.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/arpack_combined_all-0.1.jar
spark-3.2.1-bin-hadoop3.2/jars/arrow-format-2.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/arrow-memory-core-2.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/arrow-memory-netty-2.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/arrow-vector-2.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/audience-annotations-0.5.0.jar
spark-3.2.1-bin-hadoop3.2/jars/automaton-1.11-8.jar
spark-3.2.1-bin-hadoop3.2/jars/avro-1.10.2.jar
spark-3.2.1-bin-hadoop3.2/jars/avro-ipc-1.10.2.jar
spark-3.2.1-bin-hadoop3.2/jars/avro-mapred-1.10.2.jar
spark-3.2.1-bin-hadoop3.2/jars/blas-2.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/bonecp-0.8.0.RELEASE.jar
spark-3.2.1-bin-hadoop3.2/jars/breeze-macros_2.12-1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/breeze_2.12-1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/cats-kernel_2.12-2.1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/chill-java-0.10.0.jar
spark-3.2.1-bin-hadoop3.2/jars/chill_2.12-0.10.0.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-cli-1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-codec-1.15.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-collections-3.2.2.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-compiler-3.0.16.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-compress-1.21.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-crypto-1.1.0.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-dbcp-1.4.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-io-2.8.0.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-lang-2.6.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-lang3-3.12.0.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-logging-1.1.3.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-math3-3.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-net-3.1.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-pool-1.5.4.jar
spark-3.2.1-bin-hadoop3.2/jars/commons-text-1.6.jar
spark-3.2.1-bin-hadoop3.2/jars/compress-lzf-1.0.3.jar
spark-3.2.1-bin-hadoop3.2/jars/core-1.1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/curator-client-2.13.0.jar
spark-3.2.1-bin-hadoop3.2/jars/curator-framework-2.13.0.jar
spark-3.2.1-bin-hadoop3.2/jars/curator-recipes-2.13.0.jar
spark-3.2.1-bin-hadoop3.2/jars/datanucleus-api-jdo-4.2.4.jar
spark-3.2.1-bin-hadoop3.2/jars/datanucleus-core-4.1.17.jar
spark-3.2.1-bin-hadoop3.2/jars/datanucleus-rdbms-4.1.19.jar
spark-3.2.1-bin-hadoop3.2/jars/derby-10.14.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/dropwizard-metrics-hadoop-metrics2-reporter-0.1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/flatbuffers-java-1.9.0.jar
spark-3.2.1-bin-hadoop3.2/jars/generex-1.0.2.jar
spark-3.2.1-bin-hadoop3.2/jars/gson-2.2.4.jar
spark-3.2.1-bin-hadoop3.2/jars/guava-14.0.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hadoop-client-api-3.3.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hadoop-client-runtime-3.3.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hadoop-shaded-guava-1.1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hadoop-yarn-server-web-proxy-3.3.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-beeline-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-cli-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-common-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-exec-2.3.9-core.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-jdbc-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-llap-common-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-metastore-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-serde-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-service-rpc-3.1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-shims-0.23-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-shims-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-shims-common-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-shims-scheduler-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-storage-api-2.7.2.jar
spark-3.2.1-bin-hadoop3.2/jars/hive-vector-code-gen-2.3.9.jar
spark-3.2.1-bin-hadoop3.2/jars/hk2-api-2.6.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hk2-locator-2.6.1.jar
spark-3.2.1-bin-hadoop3.2/jars/hk2-utils-2.6.1.jar
spark-3.2.1-bin-hadoop3.2/jars/htrace-core4-4.1.0-incubating.jar
spark-3.2.1-bin-hadoop3.2/jars/httpclient-4.5.13.jar

spark-3.2.1-bin-hadoop3.2/jars/httpcore-4.4.14.jar
spark-3.2.1-bin-hadoop3.2/jars/istack-commons-runtime-3.0.8.jar
spark-3.2.1-bin-hadoop3.2/jars/ivy-2.5.0.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-annotations-2.12.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-core-2.12.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-core-asl-1.9.13.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-databind-2.12.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-dataformat-yaml-2.12.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-datatype-jsr310-2.11.2.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-mapper-asl-1.9.13.jar
spark-3.2.1-bin-hadoop3.2/jars/jackson-module-scala_2.12-2.12.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.annotation-api-1.3.5.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.inject-2.6.1.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.servlet-api-4.0.3.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.validation-api-2.0.2.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.ws.rs-api-2.1.6.jar
spark-3.2.1-bin-hadoop3.2/jars/jakarta.xml.bind-api-2.3.2.jar
spark-3.2.1-bin-hadoop3.2/jars/janino-3.0.16.jar
spark-3.2.1-bin-hadoop3.2/jars/javassist-3.25.0-GA.jar
spark-3.2.1-bin-hadoop3.2/jars/javax.jdo-3.2.0-m3.jar
spark-3.2.1-bin-hadoop3.2/jars/javolution-5.5.1.jar
spark-3.2.1-bin-hadoop3.2/jars/jaxb-api-2.2.11.jar
spark-3.2.1-bin-hadoop3.2/jars/jaxb-runtime-2.3.2.jar
spark-3.2.1-bin-hadoop3.2/jars/jcl-over-slf4j-1.7.30.jar
spark-3.2.1-bin-hadoop3.2/jars/jdo-api-3.0.1.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-client-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-common-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-container-servlet-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-container-servlet-core-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-hk2-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jersey-server-2.34.jar
spark-3.2.1-bin-hadoop3.2/jars/jline-2.14.6.jar
spark-3.2.1-bin-hadoop3.2/jars/joda-time-2.10.10.jar
spark-3.2.1-bin-hadoop3.2/jars/jodd-core-3.5.2.jar
spark-3.2.1-bin-hadoop3.2/jars/jpam-1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/json-1.8.jar
spark-3.2.1-bin-hadoop3.2/jars/json4s-ast_2.12-3.7.0-M11.jar
spark-3.2.1-bin-hadoop3.2/jars/json4s-core_2.12-3.7.0-M11.jar
spark-3.2.1-bin-hadoop3.2/jars/json4s-jackson_2.12-3.7.0-M11.jar
spark-3.2.1-bin-hadoop3.2/jars/json4s-scalap_2.12-3.7.0-M11.jar
spark-3.2.1-bin-hadoop3.2/jars/jsr305-3.0.0.jar
spark-3.2.1-bin-hadoop3.2/jars/jta-1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/jul-to-slf4j-1.7.30.jar
spark-3.2.1-bin-hadoop3.2/jars/kryo-shaded-4.0.2.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-client-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-admissionregistration-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-apiextensions-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-apps-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-autoscaling-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-batch-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-certificates-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-common-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-coordination-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-core-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-discovery-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-events-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-extensions-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-flowcontrol-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-metrics-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-networking-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-node-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-policy-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-rbac-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-scheduling-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/kubernetes-model-storageclass-5.4.1.jar
spark-3.2.1-bin-hadoop3.2/jars/lapack-2.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/leveldbjni-all-1.8.jar
spark-3.2.1-bin-hadoop3.2/jars/libfb303-0.9.3.jar
spark-3.2.1-bin-hadoop3.2/jars/libthrift-0.12.0.jar
spark-3.2.1-bin-hadoop3.2/jars/log4j-1.2.17.jar
spark-3.2.1-bin-hadoop3.2/jars/logging-interceptor-3.12.12.jar
spark-3.2.1-bin-hadoop3.2/jars/lz4-java-1.7.1.jar
spark-3.2.1-bin-hadoop3.2/jars/macro-compat_2.12-1.1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/mesos-1.4.0-shaded-protobuf.jar
spark-3.2.1-bin-hadoop3.2/jars/metrics-core-4.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/metrics-graphite-4.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/metrics-jmx-4.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/metrics-json-4.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/metrics-jvm-4.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/minlog-1.3.0.jar
spark-3.2.1-bin-hadoop3.2/jars/netty-all-4.1.68.Final.jar
spark-3.2.1-bin-hadoop3.2/jars/objenesis-2.6.jar
spark-3.2.1-bin-hadoop3.2/jars/okhttp-3.12.12.jar
spark-3.2.1-bin-hadoop3.2/jars/okio-1.14.0.jar
spark-3.2.1-bin-hadoop3.2/jars/opencsv-2.3.jar
spark-3.2.1-bin-hadoop3.2/jars/orc-core-1.6.12.jar
spark-3.2.1-bin-hadoop3.2/jars/orc-mapreduce-1.6.12.jar
spark-3.2.1-bin-hadoop3.2/jars/orc-shims-1.6.12.jar
spark-3.2.1-bin-hadoop3.2/jars/oro-2.0.8.jar

spark-3.2.1-bin-hadoop3.2/jars/osgi-resource-locator-1.0.3.jar
spark-3.2.1-bin-hadoop3.2/jars/paranamer-2.8.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-column-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-common-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-encoding-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-format-structures-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-hadoop-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/parquet-jackson-1.12.2.jar
spark-3.2.1-bin-hadoop3.2/jars/protobuf-java-2.5.0.jar
spark-3.2.1-bin-hadoop3.2/jars/py4j-0.10.9.3.jar
spark-3.2.1-bin-hadoop3.2/jars/pyrolite-4.30.jar
spark-3.2.1-bin-hadoop3.2/jars/rocksdbjni-6.20.3.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-collection-compat_2.12-2.1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-compiler-2.12.15.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-library-2.12.15.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-parser-combinators_2.12-1.1.2.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-reflect-2.12.15.jar
spark-3.2.1-bin-hadoop3.2/jars/scala-xml_2.12-1.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/shapeless_2.12-2.3.3.jar
spark-3.2.1-bin-hadoop3.2/jars/shims-0.9.0.jar
spark-3.2.1-bin-hadoop3.2/jars/slf4j-api-1.7.30.jar
spark-3.2.1-bin-hadoop3.2/jars/slf4j-log4j12-1.7.30.jar
spark-3.2.1-bin-hadoop3.2/jars/snakeyaml-1.27.jar
spark-3.2.1-bin-hadoop3.2/jars/snappy-java-1.1.8.4.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-catalyst_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-core_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-graphx_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-hive-thriftserver_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-hive_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-kubernetes_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-kvstore_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-launcher_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-mesos_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-mllib-local_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-mllib_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-network-common_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-network-shuffle_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-repl_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-sketch_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-sql_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-streaming_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-tags_2.12-3.2.1-tests.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-tags_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-unsafe_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spark-yarn_2.12-3.2.1.jar
spark-3.2.1-bin-hadoop3.2/jars/spire-macros_2.12-0.17.0.jar
spark-3.2.1-bin-hadoop3.2/jars/spire-platform_2.12-0.17.0.jar
spark-3.2.1-bin-hadoop3.2/jars/spire-util_2.12-0.17.0.jar
spark-3.2.1-bin-hadoop3.2/jars/spire_2.12-0.17.0.jar
spark-3.2.1-bin-hadoop3.2/jars/stax-api-1.0.1.jar
spark-3.2.1-bin-hadoop3.2/jars/stream-2.9.6.jar
spark-3.2.1-bin-hadoop3.2/jars/super-csv-2.2.0.jar
spark-3.2.1-bin-hadoop3.2/jars/threeten-extra-1.5.0.jar
spark-3.2.1-bin-hadoop3.2/jars/tink-1.6.0.jar
spark-3.2.1-bin-hadoop3.2/jars/transaction-api-1.1.jar
spark-3.2.1-bin-hadoop3.2/jars/univocity-parsers-2.9.1.jar
spark-3.2.1-bin-hadoop3.2/jars/velocity-1.5.jar
spark-3.2.1-bin-hadoop3.2/jars/xbean-asm9-shaded-4.20.jar
spark-3.2.1-bin-hadoop3.2/jars/xz-1.8.jar
spark-3.2.1-bin-hadoop3.2/jars/zjsonpatch-0.3.0.jar
spark-3.2.1-bin-hadoop3.2/jars/zookeeper-3.6.2.jar
spark-3.2.1-bin-hadoop3.2/jars/zookeeper-jute-3.6.2.jar
spark-3.2.1-bin-hadoop3.2/jars/zstd-jni-1.5.0-4.jar
spark-3.2.1-bin-hadoop3.2/kubernetes/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/Dockerfile
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/bindings/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/bindings/R/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/bindings/R/Dockerfile
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/bindings/python/
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/bindings/python/Dockerfile
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/decom.sh
spark-3.2.1-bin-hadoop3.2/kubernetes/dockerfiles/spark/entrypoint.sh
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/autoscale.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/decommissioning.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/decommissioning_cleanup.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/py_container_checks.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/pyfiles.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/python_executable_check.py
spark-3.2.1-bin-hadoop3.2/kubernetes/tests/worker_memory_check.py
spark-3.2.1-bin-hadoop3.2/licenses/
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-AnchorJS.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-CC0.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-JLargeArrays.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-JTransforms.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-antlr.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-arpack.txt

spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-automaton.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-blas.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-bootstrap.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-cloudpickle.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-d3.min.js.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-dagre-d3.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-datatables.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-dnsjava.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-f2j.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-graphlib-dot.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-istack-commons-runtime.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jakarta-annotation-api
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jakarta-ws-rs-api
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jakarta.activation-api.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jakarta.xml.bind-api.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-janino.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-javassist.html
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-javax-transaction-transaction-api.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-javolution.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jaxb-runtime.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jline.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jodd.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-join.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jquery.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-json-formatter.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-jsp-api.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-kryo.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-leveldbjni.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-machinist.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-matchMedia-polyfill.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-minlog.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-modernizr.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-mustache.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-netlib.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-paranamer.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-pmml-model.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-protobuf.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-py4j.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-pyrolite.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-re2j.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-reflectasm.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-respond.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-sbt-launch-lib.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-scala.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-scopt.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-slf4j.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-sorttable.js.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-spire.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-vis-timeline.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-xmlenc.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-zstd-jni.txt
spark-3.2.1-bin-hadoop3.2/licenses/LICENSE-zstd.txt
spark-3.2.1-bin-hadoop3.2/python/
spark-3.2.1-bin-hadoop3.2/python/.coveragerc
spark-3.2.1-bin-hadoop3.2/python/.gitignore
spark-3.2.1-bin-hadoop3.2/python/MANIFEST.in
spark-3.2.1-bin-hadoop3.2/python/README.md
spark-3.2.1-bin-hadoop3.2/python/dist/
spark-3.2.1-bin-hadoop3.2/python/docs/
spark-3.2.1-bin-hadoop3.2/python/docs/Makefile
spark-3.2.1-bin-hadoop3.2/python/docs/make.bat
spark-3.2.1-bin-hadoop3.2/python/docs/make2.bat
spark-3.2.1-bin-hadoop3.2/python/docs/source/
spark-3.2.1-bin-hadoop3.2/python/docs/source/_static/
spark-3.2.1-bin-hadoop3.2/python/docs/source/_static/copybutton.js
spark-3.2.1-bin-hadoop3.2/python/docs/source/_static/css/
spark-3.2.1-bin-hadoop3.2/python/docs/source/_static/css/pyspark.css
spark-3.2.1-bin-hadoop3.2/python/docs/source/_templates/
spark-3.2.1-bin-hadoop3.2/python/docs/source/_templates/autosummary/
spark-3.2.1-bin-hadoop3.2/python/docs/source/_templates/autosummary/class.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/_templates/autosummary/class_with_docs.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/conf.py
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/contributing.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/debugging.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/setting_ide.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/development/testing.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/getting_started/
spark-3.2.1-bin-hadoop3.2/python/docs/source/getting_started/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/getting_started/install.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/getting_started/quickstart_df.ipynb
spark-3.2.1-bin-hadoop3.2/python/docs/source/getting_started/quickstart_ps.ipynb
spark-3.2.1-bin-hadoop3.2/python/docs/source/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/koalas_to_pyspark.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_1.0_1.2_to_1.3.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_1.4_to_1.5.rst

spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_2.2_to_2.3.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_2.3.0_to_2.3.1_above.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_2.3_to_2.4.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_2.4_to_3.0.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/migration_guide/pyspark_3.1_to_3.2.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.ml.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.mllib.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/extensions.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/frame.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/general_functions.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/groupby.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/indexing.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/io.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/ml.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/series.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.pandas/window.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.resource.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.sql.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.ss.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/reference/pyspark.streaming.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/arrow_pandas.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/best_practices.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/faq.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/from_to_dbms.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/index.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/options.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/pandas_pyspark.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/transform_apply.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/typehints.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/pandas_on_spark/types.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/python_packaging.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/sql/
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/sql/arrow_pandas.rst
spark-3.2.1-bin-hadoop3.2/python/docs/source/user_guide/sql/index.rst
spark-3.2.1-bin-hadoop3.2/python/lib/
spark-3.2.1-bin-hadoop3.2/python/lib/PY4J_LICENSE.txt
spark-3.2.1-bin-hadoop3.2/python/lib/py4j-0.10.9.3-src.zip
spark-3.2.1-bin-hadoop3.2/python/lib/pyspark.zip
spark-3.2.1-bin-hadoop3.2/python/mypy.ini
spark-3.2.1-bin-hadoop3.2/python/pylintrc
spark-3.2.1-bin-hadoop3.2/python/pyspark/
spark-3.2.1-bin-hadoop3.2/python/pyspark/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/__pycache__/
spark-3.2.1-bin-hadoop3.2/python/pyspark/__pycache__/install.cpython-38.pyc
spark-3.2.1-bin-hadoop3.2/python/pyspark/_globals.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/_typing.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/accumulators.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/accumulators.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/broadcast.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/broadcast.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/cloudpickle/
spark-3.2.1-bin-hadoop3.2/python/pyspark/cloudpickle/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/cloudpickle/cloudpickle.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/cloudpickle/cloudpickle_fast.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/cloudpickle/compat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/conf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/conf.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/context.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/daemon.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/files.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/files.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/find_spark_home.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/install.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/java_gateway.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/join.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/_typing.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/base.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/classification.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/classification.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/clustering.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/clustering.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/common.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/common.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/evaluation.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/evaluation.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/feature.py

spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/feature.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/fpm.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/fpm.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/functions.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/image.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/image.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/linalg/
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/linalg/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/linalg/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/_shared_params_code_gen.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/_shared_params_code_gen.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/shared.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/param/shared.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/pipeline.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/pipeline.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/recommendation.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/recommendation.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/regression.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/regression.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/stat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/stat.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_algorithms.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_evaluation.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_feature.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_image.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_linalg.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_param.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_persistence.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_pipeline.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_stat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_training_summary.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_tuning.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tests/test_wrapper.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tree.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tree.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tuning.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/tuning.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/util.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/wrapper.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/ml/wrapper.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/_typing.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/classification.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/classification.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/clustering.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/clustering.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/common.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/common.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/evaluation.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/evaluation.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/feature.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/feature.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/fpm.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/fpm.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/linalg/
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/linalg/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/linalg/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/linalg/distributed.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/linalg/distributed.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/random.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/random.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/recommendation.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/recommendation.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/regression.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/regression.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/KernelDensity.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/KernelDensity.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/_statistics.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/_statistics.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/distribution.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/distribution.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/test.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/stat/test.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_algorithms.py

spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_feature.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_linalg.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_stat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_streaming_algorithms.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tests/test_util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tree.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/tree.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/mllib/util.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/_typing.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/accessors.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/categorical.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/config.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/binary_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/boolean_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/categorical_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/complex_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/date_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/datetime_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/null_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/num_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/string_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/data_type_ops/udt_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/datetimes.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/exceptions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/extensions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/frame.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/generic.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/groupby.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/category.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/datetimes.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/multi.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexes/numeric.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/indexing.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/internal.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/common.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/frame.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/groupby.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/indexes.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/series.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/missing/window.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/ml.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/mlflow.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/namespace.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/numpy_compat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/plot/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/plot/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/plot/core.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/plot/matplotlib.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/plot/plotly.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/series.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/spark/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/spark/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/spark/accessors.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/spark/functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/spark/utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/sql_processor.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/strings.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_binary_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_boolean_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_categorical_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_complex_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_date_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_datetime_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_null_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_num_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_string_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/test_udt_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/data_type_ops/testing_utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/indexes/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/indexes/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/indexes/test_base.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/indexes/test_category.py

spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/indexes/test_datetime.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_frame_plot.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_frame_plot_matplotlib.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_frame_plot_plotly.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_series_plot.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_series_plot_matplotlib.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/plot/test_series_plot_plotly.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_categorical.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_config.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_csv.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_dataframe.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_dataframe_conversion.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_dataframe_spark_io.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_default_index.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_expanding.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_extension.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_frame_spark.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_groupby.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_indexing.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_indexops_spark.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_internal.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_namespace.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_numpy_compat.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_ops_on_diff_frames.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_ops_on_diff_frames_groupby.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_ops_on_diff_frames_groupby_expanding.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_ops_on_diff_frames_groupby_rolling.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_repr.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_reshape.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_rolling.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_series.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_series_conversion.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_series_datetime.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_series_string.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_spark_functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_sql.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_stats.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_typedef.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/tests/test_window.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/typedef/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/typedef/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/typedef/string_typehints.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/typedef/typehints.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/usage_logging/
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/usage_logging/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/usage_logging/usage_logger.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/pandas/window.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/profiler.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/profiler.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/py.typed
spark-3.2.1-bin-hadoop3.2/python/pyspark/python/
spark-3.2.1-bin-hadoop3.2/python/pyspark/python/pyspark/
spark-3.2.1-bin-hadoop3.2/python/pyspark/python/pyspark/shell.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/rdd.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/rdd.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/rddsampler.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/information.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/information.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/profile.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/profile.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/requests.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/requests.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resource/tests/test_resources.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resultiterable.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/resultiterable.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/serializers.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/shell.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/shuffle.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/_typing.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/avro/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/avro/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/avro/functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/avro/functions.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/catalog.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/catalog.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/column.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/column.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/conf.py

spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/conf.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/context.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/dataframe.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/dataframe.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/functions.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/group.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/group.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/protocols/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/protocols/__init__.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/protocols/frame.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/_typing/protocols/series.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/conversion.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/conversion.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/functions.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/group_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/group_ops.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/map_ops.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/map_ops.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/serializers.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/typehints.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/types.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/pandas/utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/readwriter.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/readwriter.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/session.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/session.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/streaming.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/streaming.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_arrow.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_catalog.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_column.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_conf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_dataframe.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_datasources.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_functions.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_group.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_cogrouped_map.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_grouped_map.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_map.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_udf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_udf_grouped_agg.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_udf_scalar.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_udf_typehints.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_pandas_udf_window.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_readwriter.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_serde.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_session.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_streaming.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_types.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_udf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/tests/test_utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/types.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/types.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/udf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/udf.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/window.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/sql/window.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/statcounter.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/statcounter.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/status.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/status.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/storagelevel.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/storagelevel.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/context.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/dstream.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/dstream.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/kinesis.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/kinesis.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/listener.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/listener.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/test_context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/test_dstream.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/test_kinesis.py

spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/tests/test_listener.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/streaming/util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/taskcontext.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/taskcontext.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/mllibutils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/mlutils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/pandasutils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/sqlutils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/streamingutils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/testing/utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/__init__.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_appsubmit.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_broadcast.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_conf.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_context.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_daemon.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_install_spark.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_join.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_pin_thread.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_profiler.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_rdd.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_rddbarrier.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_readwrite.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_serializers.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_shuffle.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_taskcontext.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/tests/test_worker.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/traceback_utils.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/util.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/util.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/version.py
spark-3.2.1-bin-hadoop3.2/python/pyspark/version.pyi
spark-3.2.1-bin-hadoop3.2/python/pyspark/worker.py
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/PKG-INFO
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/SOURCES.txt
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/dependency_links.txt
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/requires.txt
spark-3.2.1-bin-hadoop3.2/python/pyspark.egg-info/top_level.txt
spark-3.2.1-bin-hadoop3.2/python/run-tests
spark-3.2.1-bin-hadoop3.2/python/run-tests-with-coverage
spark-3.2.1-bin-hadoop3.2/python/run-tests.py
spark-3.2.1-bin-hadoop3.2/python/setup.cfg
spark-3.2.1-bin-hadoop3.2/python/setup.py
spark-3.2.1-bin-hadoop3.2/python/test_coverage/
spark-3.2.1-bin-hadoop3.2/python/test_coverage/conf/
spark-3.2.1-bin-hadoop3.2/python/test_coverage/conf/spark-defaults.conf
spark-3.2.1-bin-hadoop3.2/python/test_coverage/coverage_daemon.py
spark-3.2.1-bin-hadoop3.2/python/test_coverage/sitecustomize.py
spark-3.2.1-bin-hadoop3.2/python/test_support/
spark-3.2.1-bin-hadoop3.2/python/test_support/SimpleHTTPServer.py
spark-3.2.1-bin-hadoop3.2/python/test_support/hello/
spark-3.2.1-bin-hadoop3.2/python/test_support/hello/hello.txt
spark-3.2.1-bin-hadoop3.2/python/test_support/hello/sub_hello/
spark-3.2.1-bin-hadoop3.2/python/test_support/hello/sub_hello/sub_hello.txt
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/ages.csv
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/ages_newlines.csv
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/_SUCCESS
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=0/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=0/c=0/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=0/c=0/.part-r-00000-829af031-b970-49d6-ad39-30460a0be2c8.orc.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=0/c=0/part-r-00000-829af031-b970-49d6-ad39-30460a0be2c8.orc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=1/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=1/c=1/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=1/c=1/.part-r-00000-829af031-b970-49d6-ad39-30460a0be2c8.orc.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/orc_partitioned/b=1/c=1/part-r-00000-829af031-b970-49d6-ad39-30460a0be2c8.orc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/_SUCCESS
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/_common_metadata
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/_metadata
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2014/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2014/month=9/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2014/month=9/day=1/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2014/month=9/day=1/.part-r-00008.gz.parquet.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2014/month=9/day=1/part-r-00008.gz.parquet
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=25/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=25/.part-r-00002.gz.parquet.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=25/.part-r-00004.gz.parquet.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=25/part-r-00002.gz.parquet
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=25/part-r-00004.gz.parquet
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=26/

spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=26/.part-r-00005.gz.parquet.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=10/day=26/part-r-00005.gz.parquet
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=9/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=9/day=1/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=9/day=1/.part-r-00007.gz.parquet.crc
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/parquet_partitioned/year=2015/month=9/day=1/part-r-00007.gz.parquet
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/people.json
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/people1.json
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/people_array.json
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/people_array_utf16le.json
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/streaming/
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/streaming/text-test.txt
spark-3.2.1-bin-hadoop3.2/python/test_support/sql/text-test.txt
spark-3.2.1-bin-hadoop3.2/python/test_support/userlib-0.1.zip
spark-3.2.1-bin-hadoop3.2/python/test_support/userlibrary.py
spark-3.2.1-bin-hadoop3.2/sbin/
spark-3.2.1-bin-hadoop3.2/sbin/decommission-slave.sh
spark-3.2.1-bin-hadoop3.2/sbin/decommission-worker.sh
spark-3.2.1-bin-hadoop3.2/sbin/slaves.sh
spark-3.2.1-bin-hadoop3.2/sbin/spark-config.sh
spark-3.2.1-bin-hadoop3.2/sbin/spark-daemon.sh
spark-3.2.1-bin-hadoop3.2/sbin/spark-daemons.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-all.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-history-server.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-master.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-mesos-dispatcher.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-mesos-shuffle-service.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-slave.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-slaves.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-thriftserver.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-worker.sh
spark-3.2.1-bin-hadoop3.2/sbin/start-workers.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-all.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-history-server.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-master.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-mesos-dispatcher.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-mesos-shuffle-service.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-slave.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-slaves.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-thriftserver.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-worker.sh
spark-3.2.1-bin-hadoop3.2/sbin/stop-workers.sh
spark-3.2.1-bin-hadoop3.2/sbin/workers.sh
spark-3.2.1-bin-hadoop3.2/yarn/
spark-3.2.1-bin-hadoop3.2/yarn/spark-3.2.1-yarn-shuffle.jar
/project/DataEngineering
Requirement already satisfied: pyspark in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (3.2.1)
Requirement already satisfied: py4j==0.10.9.3 in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from pyspark) (0.10.9.3)

In [7]:
```
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/project/spark-3.2.1-bin-hadoop3.2"
```

In [8]:
```
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .appName("PySpark App") \
    .config("spark.jars", "postgresql-42.3.2.jar") \
    .getOrCreate()
```

3.0 Data Mining

As mentioned in the introduction section, the datasets we collected for this report contain three main parts, the game results data from the 2022 Winter Olympics, STK athlete data, and recent Twitter posts on STK related topics.

3.1 Game results

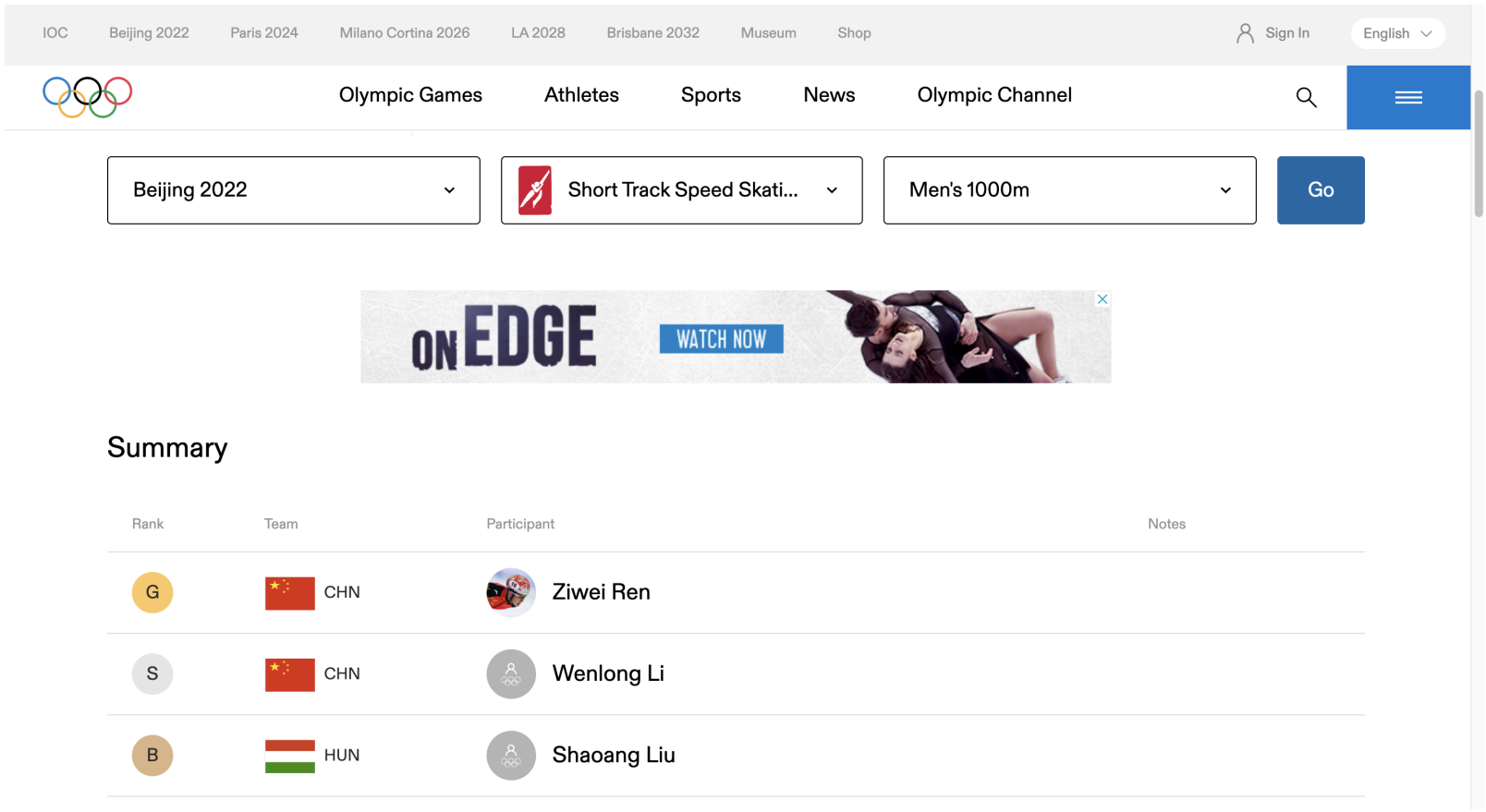
Since we were initially interested in analyzing the general performance of STK games, we first gathered the full game results of STK races of the 2022 Winter Olympics by web scraping from the International Olympic Committee (IOC) official website (International Olympic Committee, 2022), using Beautiful Soup, a widely used Python package for parsing HTML and XML documents, We first defined a 'get_file' function as a helper function to save the information pages from IOC as HTML files for further data scraping. Then we defined the function to get the needed information such as time, country and athlete from the saved HTML files using regular expressions. The data scraping was done by defined functions, which not only saves the repeated work but also provides usability of this approach for similar problems in the future.

In [9]:
```
Image("/project/DataEngineering/graphs/screenshot_current.png", width = 800)
```

Out[9]:

In [10]:
```
Image("/project/DataEngineering/graphs/html_screenshot.png", width = 800)
```

Out[10]:



Men's 1000m - Finals Results - Olympic Short Track Speed Skating

Capital Indoor Stadium 7 Feb - 13:00

My time (Change to Beijing time)

Current Records

Record	Time	Name	Date	Location
WR	1:20.875	KOR HWANG DHWANG Daeheon	12 Nov 2016	Salt Lake City, UT (USA)
OR	1:23.042	KOR HWANG DHWANG Daeheon	5 Feb 2022	Beijing (CHN)

OR:
Olympic Record
WR:
World Record

See all records

- Results
- Official Reports

1-1

Final A
Official

Rank	Helmet Number	NOC	Name	Time
1	54	CHN	REN ZREN Ziwei	1:26.768
2	94	CHN	LI WLI Wenlong	1:29.917
3	1	HUN	LIU SLIU Shaoang	1:35.693
4	48	CHN	WU DWU Dajing	1:42.937
	2	HUN	LIU SSLIU Shaolin Sandor	YC

Lap	Rank Lap Time	Time

It is worth noticing that, when we collected the game results data in early March, as the 2022 Paralympic Winter Games were still taking place, all detailed game results of each sport in the 2022 Winter Olympics including the exact time and penalty record for each athlete starting from the group qualifying stage were still available on IOC website. While entering April, with the Paralympic also closed, the IOC downsized its data for the 2022 Games, keeping only final rankings, as it did for all previous Games. This is a routine data management step for IOC, taking into account practical situations, but it also demonstrates the importance of timely data collection, storage and analysis. (shown as the two cells above)

In [11]:
```
def get_file(path):
    '''
    function to get the file and convert it into html format with BeautifulSoup
    '''
    with open(path, 'r') as f:
        contents = f.read()
    soup = BeautifulSoup(contents, 'lxml')
    soup_body = str(soup.body)
    return soup_body
```

In addition, when browsing the original data from the IOC website, we noticed that due to the variance in the competition format, the data for men's/women's singles and relay games are slightly different. In relay games, instead of athletes' names and helmet numbers, only the country names are recorded. Therefore, to avoid differentials in column numbers, the results of single games and relay games are collected by two similar but separate functions and recorded as two different datasets.

In [12]:
```
def women_men_games(event, paths, df):
```

```
    '''
    function using regular expression to get each columns in the html file
    '''
    soup_body = get_file(paths)
    pattern_country = r'<div class="playerTag" country="(.*)" register='
    pattern_name = r'<td data-sort="(.*)">'
    pattern_time = r'</span></a></div></div></td>\n<td class="text-right">\n(.*)'
    pattern_helmet = r'<td class="text-right d-none d-md-table-cell">\n(.*)</td>'
    pattern_group = r'splitContentResult-STK(.*)" role="row"'
    pattern_rank = r'<td class="text-right sorting_1" data-sort="(.*)">'
    pattern_Q_q = r'<td class="text-right">\n([\s\S]*?)\n</td>\n<td class="text-center d-none d-md-table-cell">'

    country = re.findall(pattern_country,soup_body)
    name = re.findall(pattern_name,soup_body)
    time = re.findall(pattern_time,soup_body)
    helmet = re.findall(pattern_helmet,soup_body)
    group_info =  re.findall(pattern_group,soup_body)
    rank =  re.findall(pattern_rank,soup_body)
    qualified = re.findall(pattern_Q_q,soup_body)

    time = [i.replace('</td>', '').strip() if 'td' in i else 'No Time' if 'No Time' in i else i for i in time]

    helmet = [int(i.strip()) for i in helmet]

    country = country[2:]

    group = [i[i.index(event)+len(event):i.index(event)+len(event)+7] for i in group_info]

    game_name = [i[:i.index(event)+len(event)].replace('-', ' ') for i in group_info]

    qualified = [re.findall(r'">(.*)</abr></strong>',i)  for i in qualified]


    df['country'] = country
    df['helmet_number'] = helmet
    df['name'] = name
    df['group'] = group
    df['game'] = game_name
    df['rank'] = rank
    df['time'] = time
    df['qualified'] = qualified
    make_checkpoint()
    return df
```

In [13]:

```
# get all the path for all html files
list_files = list(os.listdir('/project/DataEngineering/Html_files'))
```

In [14]:

```
# convert the file path into relative path
list_files = ['/project/DataEngineering/Html_files/' + i for i in list_files]
```

In [15]:

```
# create all the data frames
df_W500_sfnl = pd.DataFrame()
df_M1000_heat = pd.DataFrame()
df_M500_qfnl = pd.DataFrame()
df_M1000_fnl = pd.DataFrame()
df_M500_heat = pd.DataFrame()
df_M1000_qfnl = pd.DataFrame()
df_W1000_sfnl = pd.DataFrame()
df_M500_fnl = pd.DataFrame()
df_W1000_qfnl = pd.DataFrame()
df_W500_fnl = pd.DataFrame()
df_M500_sfnl = pd.DataFrame()
df_W500_heat = pd.DataFrame()
df_W500_qfnl = pd.DataFrame()
df_M1000_sfnl = pd.DataFrame()
df_W1000_heat = pd.DataFrame()
df_W1000_fnl = pd.DataFrame()
df_M1500_qfnl = pd.DataFrame()
df_W1500_sfnl = pd.DataFrame()
df_M5000R_fnl = pd.DataFrame()
df_MixR_sfnl = pd.DataFrame()
df_W3000R_fnl = pd.DataFrame()
df_M1500_fnl = pd.DataFrame()
df_W1500_fnl = pd.DataFrame()
df_MixR_fnl = pd.DataFrame()
df_W3000R_sfnl = pd.DataFrame()
df_M1500_sfnl = pd.DataFrame()
df_W1500_qfnl = pd.DataFrame()
df_M5000R_sfnl = pd.DataFrame()
df_MixR_qfnl = pd.DataFrame()
```

In [16]:

```
# form them into two lists 1) the normal men and women games; 2) the relay games
list_df = [
    df_W500_sfnl,
    df_M1000_heat,
    df_M500_qfnl,
    df_M1000_fnl,
```

```
    df_M500_heat,
    df_M1000_qfnl,
    df_W1000_sfnl,
    df_M500_fnl,
    df_W1000_qfnl,
    df_W500_fnl,
    df_M500_sfnl,
    df_W500_heat,
    df_W500_qfnl,
    df_M1000_sfnl,
    df_W1000_heat,
    df_W1000_fnl,
    df_M1500_qfnl,
    df_W1500_sfnl,
    df_M1500_fnl,
    df_W1500_fnl,
    df_M1500_sfnl,
    df_W1500_qfnl,
]

list_relay_df = [
    df_M5000R_fnl,
    df_MixR_sfnl,
    df_W3000R_fnl,
    df_MixR_fnl,
    df_W3000R_sfnl,
    df_M5000R_sfnl,
    df_MixR_qfnl
]
```

In [17]:

```
# event information for two list respectively
list_event = [
'SFNL',
'HEAT',
'QFNL',
'FNL',
'HEAT',
'QFNL',
'SFNL',
'FNL',
'QFNL',
'FNL',
'SFNL',
'HEAT',
'QFNL',
'SFNL',
'HEAT',
'FNL',
'QFNL',
'SFNL',
'FNL',
'FNL',
'SFNL',
'QFNL'
]

list_relay_event = [
    'FNL',
    'SFNL',
    'FNL',
    'FNL',
    'SFNL',
    'SFNL',
    'QFNL']
```

In [18]:

```
# remove the relay files from the list of paths and add them into the list of paths for relay only
list_files.remove("/project/DataEngineering/Html_files/Men's 5000m Relay - Finals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Mixed Team Relay - Semifinals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Women's 3000m Relay - Finals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Mixed Team Relay - Finals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Women's 3000m Relay - Semifinals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Men's 5000m Relay - Semifinals Results - Olympic Short Track Speed Skating.html")
list_files.remove("/project/DataEngineering/Html_files/Mixed Team Relay - Quarterfinals Results - Olympic Short Track Speed Skating.html")

list_relay_files = ["/project/DataEngineering/Html_files/Men's 5000m Relay - Finals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Mixed Team Relay - Semifinals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Women's 3000m Relay - Finals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Mixed Team Relay - Finals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Women's 3000m Relay - Semifinals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Men's 5000m Relay - Semifinals Results - Olympic Short Track Speed Skating.html",
          "/project/DataEngineering/Html_files/Mixed Team Relay - Quarterfinals Results - Olympic Short Track Speed Skating.html"]
```

In [19]:

```
# call the function to update the data frames
for i,df in enumerate(list_df):
    df = women_men_games(list_event[i], list_files[i],df)
```

In [20]:

```
def relay(event, paths, df):
```

```
    '''
    function using regular expression to get the information for relay games
    '''
    soup_body = get_file(paths)
    pattern_country = r'<td class="text-right" data-sort="(.*)">\n<div'
    pattern_name = r'<td data-sort="(.*)">'
    pattern_time = r'</a></div></div></td>\n<td class="text-right">\n(.*)'
    pattern_group = r'splitContentResult-STK(.*)" role="row"'
    pattern_rank = r'<td class="text-right sorting_1" data-sort="(.*)">'
    pattern_Q_q = r'<td class="text-right">\n([\s\S]*?)\n</td>\n<td class="text-center d-none d-md-table-cell">'

    country = re.findall(pattern_country,soup_body)
    name = re.findall(pattern_name,soup_body)
    time = re.findall(pattern_time,soup_body)
    group_info =  re.findall(pattern_group,soup_body)
    rank =  re.findall(pattern_rank,soup_body)
    qualified = re.findall(pattern_Q_q,soup_body)

    time = [i.replace('</td>', '').strip() if 'td' in i else 'No Time' if 'No Time' in i else i for i in time]

    group = [i[i.index(event)+len(event):i.index(event)+len(event)+6] for i in group_info]

    game_name = [i[:i.index(event)+len(event)].replace('-', ' ') for i in group_info]


    qualified = [re.findall(r'">(.*)</abbr></strong>',i)  for i in qualified]


    df['country'] = country
    df['name'] = name
    df['group'] = group
    df['game'] = game_name
    df['rank'] = rank
    df['time'] = time
    df['qualified'] = qualified
    make_checkpoint()
    return df
```

In [21]:

```
# call the relay function
for i,df in enumerate(list_relay_df):
    df = relay(list_relay_event[i], list_relay_files[i],df)
```

In [22]:

```
def concat(df,list_):
    '''
    function to concat all the information and trim the time column
    '''
    for i in list_:
        df = df.append(i)
    list_time = []
    for count, time in enumerate(df['time']):
        if '>' in time:
            list_time.append(re.findall(r'>(.*)</abbr>', time)[0])
        else:
            list_time.append(time)
    df['time'] = list_time
    make_checkpoint()
    return df
```

In [23]:

```
# create a new data frame and call the function to store all the women and men game information
df_w_m_game = pd.DataFrame()
df_w_m_game = concat(df_w_m_game, list_df)
```

In [24]:

```
# create a new data frame and call the function to store all the relay game information
df_relay_game = pd.DataFrame()
df_relay_game = concat(df_relay_game, list_relay_df)
```

Description of columns in 'df_w_m_game':

Column Name	Description
country	Three letter ISO country code of which the athlete represents
helmet_number	Number on the athlete’s helmet
name	Name of the athlete
group	Group number the athlete competed in
game	Type of game the athlete competed in first letter (M/W) indicate gender, number indicate length of race, last four letters represent level of game (FNL= final, SFNL = semi-final, QFNL = quarter-final, HEAT = group)
rank	Ranking of the athlete’s grade within the group
time	Time the athlete used to complete the race

Column Name	Description
qualified	Whether and how (including advanced by referee) the athlete was qualified for next stage race

In [25]:
```
df_w_m_game
```

Out[25]:

	country	helmet_number	name	group	game	rank	time	qualified
0	ITA	3	FONTANA Arianna	000100-	W500M SFNL	1	42.387	[QA]
1	CAN	35	BOUTIN Kim	000100-	W500M SFNL	2	42.664	[QA]
2	ROC	141	SEREGINA Elena	000100-	W500M SFNL	3	42.685	[QB]
3	CAN	50	CHARLES Alyson	000100-	W500M SFNL	4	42.829	[QB]
4	ITA	97	VALCEPINA Arianna	000100-	W500M SFNL	5	44.044	[]
...	...	...	...	...	...	...	...	...
31	BEL	5	DESMET Hanne	000600-	W1500M QFNL	2	2:18.931	[Q]
32	ROC	7	PROSVIRNOVA Sofia	000600-	W1500M QFNL	3	2:19.432	[Q]
33	CHN	42	ZHANG Chutong	000600-	W1500M QFNL	4	2:19.839	[q]
34	POL	16	MALISZEWSKA Natalia	000600-	W1500M QFNL	5	2:25.850	[]
35	ROC	9	EFREMENKOVA Ekaterina	000600-	W1500M QFNL	6	No Time	[]

429 rows × 8 columns

Description of columns in 'df_relay_game':

Column Name	Description
country	Three letter ISO country code of the team
name	Name of the country
group	Group number the team competed in
game	the first letter indicates gender (Men/Women/Mixed), the number indicates the length of the race, last four letters represent the level of the game (FNL= final, SFNL = semi-final, QFNL = quarter-final, HEAT = group)
rank	Ranking of the team's grade within the group
time	Time the team used to complete the race
qualified	Whether and how (including advanced by the referee) the team was qualified for next stage race

In [26]:
```
df_relay_game
```

Out[26]:

	country	name	group	game	rank	time	qualified
0	CAN	Canada	-A0010	M5000MRY4 FNL	1	6:41.257	[]
1	KOR	Republic of Korea	-A0010	M5000MRY4 FNL	2	6:41.679	[]
2	ITA	Italy	-A0010	M5000MRY4 FNL	3	6:43.431	[]
3	ROC	ROC	-A0010	M5000MRY4 FNL	4	6:43.440	[]
4	CHN	People's Republic of China	-A0010	M5000MRY4 FNL	5	6:51.654	[]
5	HUN	Hungary	-B0010	M5000MRY4 FNL	1	6:39.713	[]
6	NED	Netherlands	-B0010	M5000MRY4 FNL	2	6:39.780	[]
7	JPN	Japan	-B0010	M5000MRY4 FNL	3	6:40.545	[]
0	CAN	Canada	000100	XRELAY4 SFNL	1	2:36.808	[QA]
1	ITA	Italy	000100	XRELAY4 SFNL	2	2:36.895	[QA]
2	KAZ	Kazakhstan	000100	XRELAY4 SFNL	3	2:42.575	[QB]
3	NED	Netherlands	000100	XRELAY4 SFNL	4	2:51.919	[QB]
4	HUN	Hungary	000200	XRELAY4 SFNL	1	2:38.052	[QA]
5	CHN	People's Republic of China	000200	XRELAY4 SFNL	2	2:38.783	[QA]
6	ROC	ROC	000200	XRELAY4 SFNL	3	PEN	[]
7	USA	United States of America	000200	XRELAY4 SFNL	4	PEN	[]
0	NED	Netherlands	-A0010	W3000MRY4 FNL	1	4:03.409	[OR]
1	KOR	Republic of Korea	-A0010	W3000MRY4 FNL	2	4:03.627	[]
2	CHN	People's Republic of China	-A0010	W3000MRY4 FNL	3	4:03.863	[]
3	CAN	Canada	-A0010	W3000MRY4 FNL	4	4:04.329	[]
4	ITA	Italy	-B0010	W3000MRY4 FNL	1	4:09.688	[]
5	POL	Poland	-B0010	W3000MRY4 FNL	2	4:10.210	[]

	country	name	group	game	rank	time	qualified
6	ROC	ROC	-B0010	W3000MRY4 FNL	3	PEN	[]
7	USA	United States of America	-B0010	W3000MRY4 FNL	4	PEN	[]
0	CHN	People's Republic of China	-A0010	XRELAY4 FNL	1	2:37.348	[]
1	ITA	Italy	-A0010	XRELAY4 FNL	2	2:37.364	[]
2	HUN	Hungary	-A0010	XRELAY4 FNL	3	2:40.900	[]
3	CAN	Canada	-A0010	XRELAY4 FNL	4	PEN	[]
4	NED	Netherlands	-B0010	XRELAY4 FNL	1	2:36.966	[]
5	KAZ	Kazakhstan	-B0010	XRELAY4 FNL	2	2:44.148	[]
0	NED	Netherlands	000100	W3000MRY4 SFNL	1	4:04.133	[QA]
1	CHN	People's Republic of China	000100	W3000MRY4 SFNL	2	4:04.383	[QA]
2	POL	Poland	000100	W3000MRY4 SFNL	3	4:10.074	[QB]
3	ITA	Italy	000100	W3000MRY4 SFNL	4	4:17.438	[QB]
4	CAN	Canada	000200	W3000MRY4 SFNL	1	4:05.893	[QA]
5	KOR	Republic of Korea	000200	W3000MRY4 SFNL	2	4:05.904	[QA]
6	ROC	ROC	000200	W3000MRY4 SFNL	3	4:06.064	[QB]
7	USA	United States of America	000200	W3000MRY4 SFNL	4	4:06.098	[QB]
0	CAN	Canada	000100	M5000MRY4 SFNL	1	6:38.752	[QA]
1	ITA	Italy	000100	M5000MRY4 SFNL	2	6:38.899	[QA]
2	JPN	Japan	000100	M5000MRY4 SFNL	3	6:40.446	[QB]
3	CHN	People's Republic of China	000100	M5000MRY4 SFNL	4	6:51.040	[ADVA]
4	KOR	Republic of Korea	000200	M5000MRY4 SFNL	1	6:37.879	[QA]
5	ROC	ROC	000200	M5000MRY4 SFNL	2	6:37.925	[QA]
6	NED	Netherlands	000200	M5000MRY4 SFNL	3	6:37.927	[QB]
7	HUN	Hungary	000200	M5000MRY4 SFNL	4	6:45.172	[QB]
0	CHN	People's Republic of China	000100	XRELAY4 QFNL	1	2:37.535	[Q]
1	ITA	Italy	000100	XRELAY4 QFNL	2	2:38.308	[Q]
2	KOR	Republic of Korea	000100	XRELAY4 QFNL	3	2:48.308	[]
3	POL	Poland	000100	XRELAY4 QFNL	4	2:50.513	[]
4	NED	Netherlands	000200	XRELAY4 QFNL	1	2:36.437	[OR, Q]
5	CAN	Canada	000200	XRELAY4 QFNL	2	2:36.747	[Q]
6	KAZ	Kazakhstan	000200	XRELAY4 QFNL	3	2:43.004	[q]
7	FRA	France	000200	XRELAY4 QFNL	4	2:51.221	[]
8	HUN	Hungary	000300	XRELAY4 QFNL	1	2:38.396	[Q]
9	ROC	ROC	000300	XRELAY4 QFNL	2	2:38.445	[Q]
10	USA	United States of America	000300	XRELAY4 QFNL	3	2:39.043	[q]
11	JPN	Japan	000300	XRELAY4 QFNL	4	2:39.112	[]

In [27]:
```
# convert the data frame into spark data frame
w_m_game_spark_df = spark.createDataFrame(df_w_m_game)
relay_spark_df = spark.createDataFrame(df_relay_game)
```

In [28]:
```
w_m_game_spark_df.printSchema()
```

```
root
 |-- country: string (nullable = true)
 |-- helmet_number: long (nullable = true)
 |-- name: string (nullable = true)
 |-- group: string (nullable = true)
 |-- game: string (nullable = true)
 |-- rank: string (nullable = true)
 |-- time: string (nullable = true)
 |-- qualified: array (nullable = true)
 |    |-- element: string (containsNull = true)
```

In [29]:
```
relay_spark_df.printSchema()
```

```
root
 |-- country: string (nullable = true)
 |-- name: string (nullable = true)
 |-- group: string (nullable = true)
 |-- game: string (nullable = true)
 |-- rank: string (nullable = true)
```

```
|-- time: string (nullable = true)
|-- qualified: array (nullable = true)
|    |-- element: string (containsNull = true)
```

In [30]:
```
# convert the data frame into parquet format
w_m_game_spark_df.write.parquet("/project/DataEngineering/parquet_files/w_m_game.parquet", mode = 'overwrite')
relay_spark_df.write.parquet("/project/DataEngineering/parquet_files/relay.parquet", mode = 'overwrite')
make_checkpoint()
```

3.2 Athlete Information Dataset

After gathering the game data, we decided to also collect more detailed information of all athletes who participated in the STK games of Beijing 2022 Winter Olympic Games. From a data analysis point of view, this further enriches our data set from a very different standpoint compared to the game results data. Meanwhile, from the perspective of practical utility, we believe that for professional sports competitions, athletes' individual status information in a certain season is also an important factor to determine their performance. Therefore, collecting data on both the game results and the athlete information can help us enrich the data sources and the dataset, and thus maximize the practical value.

In [31]:
```
# read the game results from the parquet file
w_m_game_df = spark.read.parquet("/project/DataEngineering/parquet_files/w_m_game.parquet").toPandas()
```

In [32]:
```
w_m_game_df
```

Out[32]:

	country	helmet_number	name	group	game	rank	time	qualified
0	CAN	50	CHARLES Alyson	000100-	W500M QFNL	4	1:07.206	[ADV]
1	CAN	14	BRUNELLE Florence	000100-	W500M QFNL	5	PEN	[]
2	HUN	10	JASZAPATI Petra	000200-	W500M QFNL	1	43.476	[Q]
3	ROC	141	SEREGINA Elena	000200-	W500M QFNL	2	43.712	[Q]
4	USA	52	BINEY Maame	000200-	W500M QFNL	3	46.099	[]
...	...	...	...	...	...	...	...	...
424	USA	19	HEO Andrew	000100-	M1000M QFNL	1	1:24.603	[Q]
425	CHN	48	WU Dajing	000100-	M1000M QFNL	2	1:33.302	[Q]
426	KOR	195	PARK Janghyuk	000100-	M1000M QFNL	3	No Time	[ADV]
427	ITA	7	SIGHEL Pietro	000100-	M1000M QFNL	4	PEN	[]
428	CAN	67	PIERRE-GILLES Jordan	000100-	M1000M QFNL	5	PEN	[]

429 rows × 8 columns

Since we aim to only collect the information on athletes who participated in the individual STK games in the 2022 Beijing Winter Olympics, we first sorted and filtered all the unique athlete names from the dataset generated above ('w_m_game_df'). The filtered names and their country code formed a new data frame with 100 rows in total. Starting from only the names and country codes, we tried to gain more personal information about each athlete.

In [33]:
```
# get all the unique athlete information
athlete_info = w_m_game_df[['country', 'name','helmet_number']].groupby(['country','name']).nunique().reset_index().drop('helmet_number', axis = 1)
```

In [34]:
```
athlete_info
```

Out[34]:

	country	name
0	AUS	COREY Brendan
1	BEL	DESMET Hanne
2	BEL	DESMET Stijn
3	CAN	BLAIS Danae
4	CAN	BOUTIN Kim
...	...	...
95	USA	HEO Andrew
96	USA	LETAI Julie
97	USA	PIVIROTTO Ryan
98	USA	SANTOS Kristen
99	USA	STODDARD Corinne

100 rows × 2 columns

In [35]:
```
# set all the athlete name to lower case
athlete_info.name = [i.lower() for i in athlete_info.name]
```

In [36]:
```
# replace country code of ROC to RUS
athlete_info.country = athlete_info.country.replace(to_replace = 'ROC', value = 'RUS')
```

At first, we planned to gather the needed information from each athlete's Wikipedia page. However, after some trials, we found the idea not that feasible. Firstly, as the editing of Wikipedia pages is almost completely open, the information display is not very structured, which makes it more difficult for us to scrape specific data. Moreover, the amount of information each athlete's Wikipedia page contains varies greatly, depending on their nationality, popularity and other factors. Some athletes don't even have their own Wikipedia pages. Not to mention the possibility of getting directed to other celebrities with the same name. Therefore, we gave up on the idea of collecting athletes' personal information from their Wikipedia pages.

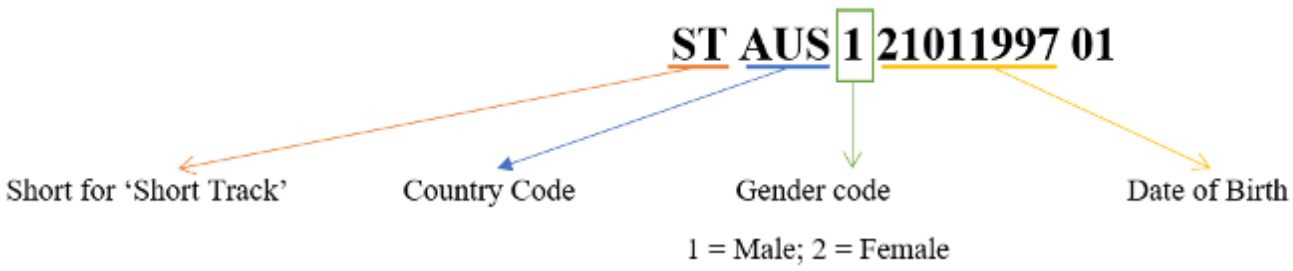
After some searching, we found Short Track Online (STO), a website that systematically stores a large volume of STK related data including the personal information of the Olympic athletes we were looking for. On the STO website, we were able to quickly identify the unique ID number of each athlete in our data frame by their country code and name using the defined function 'get_id'. The function first navigates to the list of all STK athletes serving for the country by country code, and then finds the athlete by name and gets their ID number. The package used here is still Beautiful Soup.

Among all 100 athletes, there were 6 whose unique IDs were not identified through the 'get_id' function. Taking a closer look, we found that this is caused by the non-English letters or different spacing in their names. Therefore, we manually replaced the name and successfully got their ID.

With their unique ID number, we were able to navigate to each athlete's information page and again use defined regular expressions functions to collect the data we need. But through observation, we also noticed the generating rule of their ID number, which contains information about their country code, gender and date of birth (shown below). Therefore, we were able to directly get the age and gender data for each athlete from their ID.

In [37]:
```
Image("/project/DataEngineering/graphs/id_description.png", width = 900)
```

Out[37]:



In [38]:
```
def get_id (df, cols):
    '''
    function to get each athlete id
    '''
    dict_id = {}
    for c, n in zip(df[cols[0]], df[cols[1]]):
        URL =f"http://www.shorttrackonline.info/athletes.php?country={c}"
        page = requests.get(URL)

        soup = BeautifulSoup(page.content, "html.parser")
        soup_body = str(soup.body)
        temp_dict_id_name = {''.join(k.lower().split(',')):v for k,v in zip (re.findall(r'">(.*)</a></td>\n', soup_body), re.findall(r'<a href="skaterbio.php\?id=(
        if n in temp_dict_id_name.keys():
            dict_id[n] = temp_dict_id_name[n]
    return dict_id
```

In [39]:
```
dict_id = get_id(athlete_info, ['country', 'name'])
```

In [40]:
```
# merge the id information with the athlete data frame
athlete_info = athlete_info.merge(pd.DataFrame({'name':dict_id.keys(), 'id': dict_id.values()}), how = 'left', on = 'name', copy = False)
make_checkpoint()
```

In [41]:
```
athlete_info
```

Out[41]:

	country	name	id
0	AUS	corey brendan	STAUS12101199701
1	BEL	desmet hanne	STBEL22610199601
2	BEL	desmet stijn	STBEL11004199801
3	CAN	blais danae	NaN
4	CAN	boutin kim	STCAN21612199401
...	...	...	...
95	USA	heo andrew	STUSA10705200101
96	USA	letai julie	STUSA22306200001

	country	name	id
97	USA	pivirotto ryan	STUSA11405199501
98	USA	santos kristen	STUSA20211199401
99	USA	stoddard corinne	STUSA21508200101

100 rows × 3 columns

In [42]:
```
# check the NaN rows
athlete_info[athlete_info.id.isnull()]
```

Out[42]:

	country	name	id
3	CAN	blais danae	NaN
14	CHN	han yutong	NaN
28	FRA	lepape sebastien	NaN
63	KOR	lee juneseo	NaN
65	KOR	park janghyuk	NaN
82	RUS	airapetian denis	NaN

In [43]:
```
# manually replace the names
athlete_info.name = athlete_info.name.replace('blais danae', 'blais danaé')
athlete_info.name = athlete_info.name.replace('han yutong', 'han yu tong')
athlete_info.name = athlete_info.name.replace('lepape sebastien', 'lepape sébastien')
athlete_info.name = athlete_info.name.replace('lee juneseo', 'lee june seo')
athlete_info.name = athlete_info.name.replace('park janghyuk', 'park jang hyuk')
athlete_info.name = athlete_info.name.replace('airapetian denis', 'ayrapetyan denis')
```

In [44]:
```
athlete_info[athlete_info.id.isnull()]
```

Out[44]:

	country	name	id
3	CAN	blais danaé	NaN
14	CHN	han yu tong	NaN
28	FRA	lepape sébastien	NaN
63	KOR	lee june seo	NaN
65	KOR	park jang hyuk	NaN
82	RUS	ayrapetyan denis	NaN

In [45]:
```
# call the function again to get the rest of the athlete id
dict_id_second = get_id(athlete_info[athlete_info.id.isnull()], ['country', 'name'])
```

In [46]:
```
# get the new data frame for the NaN athlete
temp_df = pd.DataFrame({'name':dict_id_second.keys(), 'id': dict_id_second.values()})
```

In [47]:
```
# map the id and name into the athlete_info data frame
athlete_info.id.fillna(athlete_info['name'].map(dict_id_second), inplace=True)
make_checkpoint()
```

In [48]:
```
athlete_info
```

Out[48]:

	country	name	id
0	AUS	corey brendan	STAUS12101199701
1	BEL	desmet hanne	STBEL22610199601
2	BEL	desmet stijn	STBEL11004199801
3	CAN	blais danaé	STCAN21005199901
4	CAN	boutin kim	STCAN21612199401
...	...	...	...
95	USA	heo andrew	STUSA10705200101
96	USA	letai julie	STUSA22306200001
97	USA	pivirotto ryan	STUSA11405199501
98	USA	santos kristen	STUSA20211199401
99	USA	stoddard corinne	STUSA21508200101

100 rows × 3 columns

In [49]:
```
# get the birth year from each id
athlete_info['birth_year'] = athlete_info['id'].apply(lambda x: x[-6:-2])
```

In [50]:
```
athlete_info['birth_year'] = athlete_info['birth_year'].astype(int)
make_checkpoint()
```

In [51]:
```
# get the age by 2022-birth_year
athlete_info['age'] = 2022 - athlete_info['birth_year']
make_checkpoint()
```

In [52]:
```
# get the gender information from the id
athlete_info['gender'] = athlete_info['id'].apply(lambda x: x[-11:-10])
```

In [53]:
```
# replace 1 and 2 with male and female
athlete_info.gender = athlete_info.gender.replace({'1': 'Male', '2': 'Female'})
make_checkpoint()
```

Lastly, we used the Beautiful Soup package again to collect athletes' other information including age category and the club they work for. The final athlete information dataset is demonstrated below.

In [54]:
```
# get the age category and club information from each athlete
dict_info = {}
for i in athlete_info['id']:
    URL =f"https://www.shorttrackonline.info/skaterbio.php?id={i}"
    page = requests.get(URL)

    soup = BeautifulSoup(page.content, "html.parser")
    soup_body = str(soup.body)
    age_cate = re.findall(r'Age Category:</td>\n<td class="bio">(.*)</td>', soup_body)
    club = re.findall(r'Club:</td>\n<td class="bio">(.*)</td>', soup_body)
    temp = []
    if len(age_cate) != 0:
        temp.append(age_cate[0])
    else:
        temp.append('')
    if len(club) != 0:
        temp.append(club[0])
    else:
        temp.append('')

    dict_info[i] = temp
```

```
Some characters could not be decoded, and were replaced with REPLACEMENT CHARACTER.
Some characters could not be decoded, and were replaced with REPLACEMENT CHARACTER.
Some characters could not be decoded, and were replaced with REPLACEMENT CHARACTER.
Some characters could not be decoded, and were replaced with REPLACEMENT CHARACTER.
```

In [55]:
```
# merge the information of age_category and club into the athlete_info data frame
athlete_info = athlete_info.merge(pd.DataFrame({'id': dict_info.keys(), 'age_category': [i[0] for i in dict_info.values()], 'club': [i[1] for i in dict_info.values()
make_checkpoint()
```

In [56]:
```
athlete_info
```

Out[56]:

	country	name	id	birth_year	age	gender	age_category	club
0	AUS	corey brendan	STAUS12101199701	1997	25	Male	Senior (2021/2022)	
1	BEL	desmet hanne	STBEL22610199601	1996	26	Female	Senior (2021/2022)	Ice Diamonds Antwerp , Deurne
2	BEL	desmet stijn	STBEL11004199801	1998	24	Male	Senior (2021/2022)	Ice Diamonds Antwerp , Deurne
3	CAN	blais danaé	STCAN21005199901	1999	23	Female	Senior (2021/2022)	Speed Skating Canada,
4	CAN	boutin kim	STCAN21612199401	1994	28	Female	Senior (2021/2022)	
...	...	...	...	...	...	...	...	...
95	USA	heo andrew	STUSA10705200101	2001	21	Male	Senior (2021/2022)	
96	USA	letai julie	STUSA22306200001	2000	22	Female	Senior (2021/2022)	
97	USA	pivirotto ryan	STUSA11405199501	1995	27	Male	Senior (2021/2022)	
98	USA	santos kristen	STUSA20211199401	1994	28	Female	Senior (2021/2022)	
99	USA	stoddard corinne	STUSA21508200101	2001	21	Female	Senior (2021/2022)	

100 rows × 8 columns

In [57]:
```
# convert the athlete_info into spark data frame
athlete_info_df = spark.createDataFrame(athlete_info)
```

In [58]:

```
athlete_info_df.printSchema()
```

```
root
 |-- country: string (nullable = true)
 |-- name: string (nullable = true)
 |-- id: string (nullable = true)
 |-- birth_year: long (nullable = true)
 |-- age: long (nullable = true)
 |-- gender: string (nullable = true)
 |-- age_category: string (nullable = true)
 |-- club: string (nullable = true)
```

In [59]:

```
# convert the data frame into parquet format
athlete_info_df.write.parquet("/project/DataEngineering/parquet_files/athlete_info.parquet", mode = 'overwrite')
make_checkpoint()
```

3.3 Twitter Dataset

The two datasets gathered above are both highly quantitative and are both from a more professional perspective. In order to enrich our datasets and make our analysis more comprehensive and reliable, we decided to add another dataset that is from a more qualitative dimension. Therefore, we developed a tweet collection model using JSON which allows us to gather the most recent Twitter posts containing given words or phrases.

For this report, we decided to query it on the phrase 'Short Track Skating', which found the 10 most recent tweets on this topic. Combined with natural language processing skills, this dataset can be used to keep track of the latest STK relevant news and gain insights into the public attributes towards STK. Moreover, simply by changing the query word, this model can be easily applied to collect tweets on any other topic.

All three datasets mentioned above are converted and stored in parquet format after collection and some basic data cleaning. This format ensures the safety and integrity of the datasets and largely improves the data storage efficiency.

In [60]:

```
# get the tweets using the twitter api
import pandas as pd
import requests
import json

api_key = 'wkxIkajRgLTVkbL2N9zz0G7RN'
api_secret_key = 'nXuL6pk0OaluzSzxIjIbSBhbsz5EfBrsULX6IatiPJ3D00Auk5'
bearer_token = 'AAAAAAAAAAAAAAAAAAAAADx2aQEAAAAAfdKUH18l6uqR9DAwzr4fER9CS1U%3DnHcc0nZYYe8JfMdYMSnlkDiE1Qiupp6zVqRxzUQp


query = "Short Track Skating"

# Prepare the headers to pass the authentication to Twitter's api
headers = {
    'Authorization': 'Bearer {}'.format(bearer_token),
}

params = (
    ('query', query),
)

# Does the request to get the most recent tweets
response = requests.get('https://api.twitter.com/2/tweets/search/recent', headers=headers, params=params)

# Validates that the query was successful
if response.status_code == 200:
    print("URL of query:", response.url)

    # Let's convert the query result to a dictionary that we can iterate over
    tweets =  json.loads(response.text)

    for tweet in tweets['data']:
        print("tweet_id: ", tweet['id'], "tweet_text: ", tweet['text'])
```

```
URL of query: https://api.twitter.com/2/tweets/search/recent?query=Short+Track+Skating
tweet_id:  1513626887200952320 tweet_text:  RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in a
n emotional farewell to competitive short…
tweet_id:  1513623598321258496 tweet_text:  RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in a
n emotional farewell to competitive short…
tweet_id:  1513623524539314182 tweet_text:  RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in a
n emotional farewell to competitive short…
tweet_id:  1513622926204964866 tweet_text:  A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in an emotion
al farewell to competitive short track speed skating. | @CBCSports
https://t.co/WveNo57Fvg
tweet_id:  1513619149221601282 tweet_text:  Charles Hamelin ended his career in short track speed skating https://t.co/OCamPwdh3U
tweet_id:  1513593051373002755 tweet_text:  RT @Fradi_HU: Congratulations to Liu Shaoang for his spectacular performance at the World Short Track
Speed Skating Championships! 👏💚🤍

#F…
tweet_id:  1513575736606658563 tweet_text:  RT @Fradi_HU: Congratulations to Liu Shaoang for his spectacular performance at the World Short Track
Speed Skating Championships! 👏💚🤍

#F…
tweet_id:  1513575411883687937 tweet_text:  Congratulations to Liu Shaoang for his spectacular performance at the World Short Track Speed Skating
```

Championships! 👏💚🤍

#Fradi #ftc #ferencvaros https://t.co/ZKeVsQLvPJ
tweet_id: 1513574899788533780 tweet_text: Kim Boutin wins four silver medals at the 2022 World Short Track Speed Skating Championships https://t.co/0yoUsy68Iz
tweet_id: 1513564029851090944 tweet_text: RT @seastar_sa: 湖南卫视你好星期六 weibo updated

EP13《你好，星期六》Immersive experience of short track speed skating, different re-enactments of Miao lan...

In [61]:
```
# convert the tweets information into data frame
stk_tweet = pd.DataFrame({'id':[tweet['id'] for tweet in tweets['data']], 'text':[tweet['text'] for tweet in tweets['data']]})
make_checkpoint()
```

In [62]:
```
stk_tweet
```

Out[62]:

	id	text
0	1513626887200952320	RT @CBC: A day to remember in Montreal Olympic...
1	1513623598321258496	RT @CBC: A day to remember in Montreal Olympic...
2	1513623524539314182	RT @CBC: A day to remember in Montreal Olympic...
3	1513622926204964866	A day to remember in Montreal Olympic legend C...
4	1513619149221601282	Charles Hamelin ended his career in short trac...
5	1513593051373002755	RT @Fradi_HU: Congratulations to Liu Shaoang f...
6	1513575736606658563	RT @Fradi_HU: Congratulations to Liu Shaoang f...
7	1513575411883687937	Congratulations to Liu Shaoang for his spectac...
8	1513574899788533780	Kim Boutin wins four silver medals at the 2022...
9	1513564029851090944	RT @seastar_sa: 湖南卫视你好星期六 weibo updated\n\nEP1...

3.3.1 NLP

Natural Language Processing (NLP) has also been used as a tool for us to analyze our data. Extracting data from Twitter could help us know about the response of the public to our athletes and STK games. Hence, vaderSentimental analysis has been introduced to our analysis. However, due to the complexity of the literature, we could only collect 10 tweets in total, which is not really enough to run a complete sentimental analysis. In order to reduce bias and maximize our test results, we decided to analyze our whole tweets instead of splitting them into training sets and test sets.

In [63]:
```
!pip install vaderSentiment
```

```
      .::::.      .::.
    ....yy:     .yy.
    :.  .yy.     y.
         :y:    .:
         .yy   .:
          yy...:
          :y:.
          .y.
         .:.
    .....:.
    ::::.
```

• Project files and data should be stored in /project. This is shared among everyone in the project.
• Personal files and configuration should be stored in /home/faculty.
• Files outside /project and /home/faculty will be lost when this server is terminated.
• Create custom environments to setup your servers reproducibly.

Requirement already satisfied: vaderSentiment in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (3.3.2)
Requirement already satisfied: requests in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from vaderSentiment) (2.27.1)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from requests->vaderSentiment) (2021.5.30)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from requests->vaderSentiment) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from requests->vaderSentiment) (1.26.4)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/anaconda/envs/Python3/lib/python3.8/site-packages (from requests->vaderSentiment) (2.0.12)

In [64]:
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

In [65]:
```
# doing the sentiment analysis
analyzer = SentimentIntensityAnalyzer()
list_all_score = []
for t in stk_tweet['text']:
    vs = analyzer.polarity_scores(t)
    print(t)
    print(vs)
    list_all_score.append(vs)
```

```
RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in an emotional farewell to competitive short…
{'neg': 0.0, 'neu': 0.758, 'pos': 0.242, 'compound': 0.6597}
RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in an emotional farewell to competitive short…
{'neg': 0.0, 'neu': 0.758, 'pos': 0.242, 'compound': 0.6597}
RT @CBC: A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in an emotional farewell to competitive short…
{'neg': 0.0, 'neu': 0.758, 'pos': 0.242, 'compound': 0.6597}
A day to remember in Montreal Olympic legend Charles Hamelin captured one last medal in an emotional farewell to competitive short track speed sk
ating. | @CBCSports
https://t.co/WveNo57Fvg
{'neg': 0.0, 'neu': 0.789, 'pos': 0.211, 'compound': 0.6597}
Charles Hamelin ended his career in short track speed skating https://t.co/OCamPwdh3U
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
RT @Fradi_HU: Congratulations to Liu Shaoang for his spectacular performance at the World Short Track Speed Skating Championships! 👏💚🤍

#F…
{'neg': 0.0, 'neu': 0.74, 'pos': 0.26, 'compound': 0.8122}
RT @Fradi_HU: Congratulations to Liu Shaoang for his spectacular performance at the World Short Track Speed Skating Championships! 👏💚🤍

#F…
{'neg': 0.0, 'neu': 0.74, 'pos': 0.26, 'compound': 0.8122}
Congratulations to Liu Shaoang for his spectacular performance at the World Short Track Speed Skating Championships! 👏💚🤍

#Fradi #ftc #ferencvaros https://t.co/ZKeVsQLvPJ
{'neg': 0.0, 'neu': 0.749, 'pos': 0.251, 'compound': 0.8122}
Kim Boutin wins four silver medals at the 2022 World Short Track Speed Skating Championships https://t.co/0yoUsy68Iz
{'neg': 0.0, 'neu': 0.67, 'pos': 0.33, 'compound': 0.7845}
RT @seastar_sa: 湖南卫视你好星期六 weibo updated
EP13《你好，星期六》Immersive experience of short track speed skating, different re-enactments of Miao lan…
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
```

In [66]:
```
# merge the sentiment data for each row
stk_tweet = pd.concat([stk_tweet, pd.DataFrame({'neg':[i['neg'] for i in list_all_score],
          'neu':[i['neu'] for i in list_all_score],
          'pos':[i['pos'] for i in list_all_score],
          'compound':[i['compound'] for i in list_all_score]})], axis = 1)
make_checkpoint()
```

In [67]:
```
stk_tweet
```

Out[67]:

	id	text	neg	neu	pos	compound
0	1513626887200952320	RT @CBC: A day to remember in Montreal Olympic…	0.0	0.758	0.242	0.6597
1	1513623598321258496	RT @CBC: A day to remember in Montreal Olympic…	0.0	0.758	0.242	0.6597
2	1513623524539314182	RT @CBC: A day to remember in Montreal Olympic…	0.0	0.758	0.242	0.6597
3	1513622926204964866	A day to remember in Montreal Olympic legend C…	0.0	0.789	0.211	0.6597
4	1513619149221601282	Charles Hamelin ended his career in short trac…	0.0	1.000	0.000	0.0000
5	1513593051373002755	RT @Fradi_HU: Congratulations to Liu Shaoang f…	0.0	0.740	0.260	0.8122
6	1513575736606658563	RT @Fradi_HU: Congratulations to Liu Shaoang f…	0.0	0.740	0.260	0.8122
7	1513575411883687937	Congratulations to Liu Shaoang for his spectac…	0.0	0.749	0.251	0.8122
8	1513574899788533780	Kim Boutin wins four silver medals at the 2022…	0.0	0.670	0.330	0.7845
9	1513564029851090944	RT @seastar_sa: 湖南卫视你好星期六 weibo updated\n\nEP1…	0.0	1.000	0.000	0.0000

In [68]:
```
# convert the data frame into spark data frame
stk_tweet_df = spark.createDataFrame(stk_tweet)
```

In [69]:
```
stk_tweet_df.printSchema()
```

```
root
 |-- id: string (nullable = true)
 |-- text: string (nullable = true)
 |-- neg: double (nullable = true)
 |-- neu: double (nullable = true)
 |-- pos: double (nullable = true)
 |-- compound: double (nullable = true)
```

In [70]:
```
# convert the data frame into parquet format
stk_tweet_df.write.parquet("/project/DataEngineering/parquet_files/stk_tweet.parquet", mode = 'overwrite')
make_checkpoint()
```

4.0 Data Transformation

Although the datasets collected above are mostly well organized and structured, and we have also done some basic data cleaning, they are still not ideal for further analysis. Besides, instead of analyzing each dataset on its own, our objective is to architect and deploy a systematic infrastructure necessary to deliver data for analysis. Therefore, more data processing and transformations are required.

We mostly focused the data transformations on the game results dataset as it is the base data that we are initially and eventually interested in. Therefore, for both the men/women singles and the relay game results data frame, we applied the following transformations.

The data processing and transformation steps above are applied similarly to both the individual game results dataset and the relay dataset.

In [71]:
```
# read the parquet files
w_m_game_df = spark.read.parquet("/project/DataEngineering/parquet_files/w_m_game.parquet").toPandas()
relay_df =  spark.read.parquet("/project/DataEngineering/parquet_files/relay.parquet").toPandas()
athlete_info_df = spark.read.parquet("/project/DataEngineering/parquet_files/athlete_info.parquet").toPandas()
```

In [72]:
```
# replace all the ROC with RUS
w_m_game_df.country = w_m_game_df.country.replace(to_replace = 'ROC', value = 'RUS')
relay_df.country = relay_df.country.replace(to_replace = 'ROC', value = 'RUS')
relay_df.name = relay_df.name.replace(to_replace = 'ROC', value = 'RUS')
make_checkpoint()
```

In [73]:
```
# set the athlete name into lower cases
w_m_game_df ['name'] = w_m_game_df['name'].apply(lambda x: x.lower())
make_checkpoint()
```

In [74]:
```
# replace the special cases
w_m_game_df.name = w_m_game_df.name.replace('blais danae', 'blais danaé')
w_m_game_df.name = w_m_game_df.name.replace('han yutong', 'han yu tong')
w_m_game_df.name = w_m_game_df.name.replace('lepape sebastien', 'lepape sébastien')
w_m_game_df.name = w_m_game_df.name.replace('lee juneseo', 'lee june seo')
w_m_game_df.name = w_m_game_df.name.replace('park janghyuk', 'park jang hyuk')
w_m_game_df.name = w_m_game_df.name.replace('airapetian denis', 'ayrapetyan denis')
make_checkpoint()
```

Qualified

Firstly, we looked at the 'qualified' column. As mentioned in previous discussions, the referee's decisions largely influence the STK games. It is not unusual to have some athletes who had bad time results but got 'advanced' into the next level by referees considering the accidents that occurred. We found that, apart from the qualification data, record-breaking results are also marked in this column. As we do not intend to analyze the record-breaking, they are thus removed.

In [75]:
```
# check the special cases in qualified column
w_m_game_df[w_m_game_df['qualified'].apply(lambda x: len(x)>1)]['qualified']
```

Out[75]:
```
32      [OR, Q]
70      [OR, Q]
141     [WR, Q]
191     [OR, Q]
227    [OR, QA]
348     [OR, Q]
Name: qualified, dtype: object
```

In [76]:
```
# merge the athlete information with the women and men game information
w_m_game_df = w_m_game_df.merge(athlete_info_df[['name', 'id']], how = 'left', on = 'name')
make_checkpoint()
```

In [77]:
```
def get_qualified(x):
    '''
    only retrun the qualified details
    '''
    if len(x) == 1 and x[0] != 'OR':
        return x[0]
    elif len(x) == 2:
        return x[1]
    else:
        return ''
```

In [78]:
```
# apply the function on both women & men game information and relay game information
w_m_game_df.qualified = w_m_game_df['qualified'].apply(get_qualified)
make_checkpoint()
relay_df.qualified = relay_df['qualified'].apply(get_qualified)
make_checkpoint()
```

In [79]:
```
# check the information returned
set(w_m_game_df.qualified)
```

Out[79]:
```
{'', 'ADV', 'ADVA', 'ADVB', 'Q', 'QA', 'QB', 'q'}
```

In [80]:
```
# check the information returned
set(relay_df.qualified)
```

Out[80]:
```
{'', 'ADVA', 'Q', 'QA', 'QB', 'q'}
```

In [81]:
```
def get_special_cases(df):
    '''
    convert the special cases in time column into the qualified column
    '''
    index = df[df['time'].isin([ 'PEN', 'No Time', 'DNS', 'YC','DNF']) & (df['qualified']=='')].index
    df['qualified'].loc[index] = df['time'].loc[index]
    return df
```

In [82]:
```
# apply to both data frame
w_m_game_df = get_special_cases(w_m_game_df)
make_checkpoint()
relay_df = get_special_cases(relay_df)
make_checkpoint()
```

```
/opt/anaconda/envs/Python3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
```

In [83]:
```
w_m_game_df
```

Out[83]:

	country	helmet_number	name	group	game	rank	time	qualified	id
0	CAN	50	charles alyson	000100-	W500M QFNL	4	1:07.206	ADV	STCAN23010199801
1	CAN	14	brunelle florence	000100-	W500M QFNL	5	PEN	PEN	STCAN22012200301
2	HUN	10	jaszapati petra	000200-	W500M QFNL	1	43.476	Q	STHUN23112199801
3	RUS	141	seregina elena	000200-	W500M QFNL	2	43.712	Q	STRUS23012200101
4	USA	52	biney maame	000200-	W500M QFNL	3	46.099		STUSA22801200001
...	...	...	...	...	...	...	...	...	...
424	USA	19	heo andrew	000100-	M1000M QFNL	1	1:24.603	Q	STUSA10705200101
425	CHN	48	wu dajing	000100-	M1000M QFNL	2	1:33.302	Q	STCHN12407199401
426	KOR	195	park jang hyuk	000100-	M1000M QFNL	3	No Time	ADV	STKOR13110199801
427	ITA	7	sighel pietro	000100-	M1000M QFNL	4	PEN	PEN	STITA11507199901
428	CAN	67	pierre-gilles jordan	000100-	M1000M QFNL	5	PEN	PEN	STCAN12405199801

429 rows × 9 columns

In [84]:
```
relay_df
```

Out[84]:

	country	name	group	game	rank	time	qualified
0	KOR	Republic of Korea	000200	M5000MRY4 SFNL	1	6:37.879	QA
1	RUS	RUS	000200	M5000MRY4 SFNL	2	6:37.925	QA
2	NED	Netherlands	000200	M5000MRY4 SFNL	3	6:37.927	QB
3	HUN	Hungary	000200	M5000MRY4 SFNL	4	6:45.172	QB
4	CHN	People's Republic of China	000100	XRELAY4 QFNL	1	2:37.535	Q
5	ITA	Italy	000100	XRELAY4 QFNL	2	2:38.308	Q
6	KOR	Republic of Korea	000100	XRELAY4 QFNL	3	2:48.308	
7	POL	Poland	000100	XRELAY4 QFNL	4	2:50.513	
8	NED	Netherlands	000200	XRELAY4 QFNL	1	2:36.437	Q
9	CAN	Canada	000200	XRELAY4 QFNL	2	2:36.747	Q
10	KAZ	Kazakhstan	000200	XRELAY4 QFNL	3	2:43.004	q
11	FRA	France	000200	XRELAY4 QFNL	4	2:51.221	
12	HUN	Hungary	000300	XRELAY4 QFNL	1	2:38.396	Q
13	RUS	RUS	000300	XRELAY4 QFNL	2	2:38.445	Q
14	USA	United States of America	000300	XRELAY4 QFNL	3	2:39.043	q
15	JPN	Japan	000300	XRELAY4 QFNL	4	2:39.112	
16	NED	Netherlands	-B0010	XRELAY4 FNL	1	2:36.966	
17	KAZ	Kazakhstan	-B0010	XRELAY4 FNL	2	2:44.148	
18	NED	Netherlands	000100	W3000MRY4 SFNL	1	4:04.133	QA
19	CHN	People's Republic of China	000100	W3000MRY4 SFNL	2	4:04.383	QA
20	POL	Poland	000100	W3000MRY4 SFNL	3	4:10.074	QB
21	ITA	Italy	000100	W3000MRY4 SFNL	4	4:17.438	QB
22	CAN	Canada	000200	W3000MRY4 SFNL	1	4:05.893	QA

	country	name	group	game	rank	time	qualified
23	KOR	Republic of Korea	000200	W3000MRY4 SFNL	2	4:05.904	QA
24	RUS	RUS	000200	W3000MRY4 SFNL	3	4:06.064	QB
25	USA	United States of America	000200	W3000MRY4 SFNL	4	4:06.098	QB
26	CAN	Canada	000100	M5000MRY4 SFNL	1	6:38.752	QA
27	ITA	Italy	000100	M5000MRY4 SFNL	2	6:38.899	QA
28	JPN	Japan	000100	M5000MRY4 SFNL	3	6:40.446	QB
29	CHN	People's Republic of China	000100	M5000MRY4 SFNL	4	6:51.040	ADVA
30	CAN	Canada	-A0010	M5000MRY4 FNL	1	6:41.257	
31	KOR	Republic of Korea	-A0010	M5000MRY4 FNL	2	6:41.679	
32	ITA	Italy	-A0010	M5000MRY4 FNL	3	6:43.431	
33	RUS	RUS	-A0010	M5000MRY4 FNL	4	6:43.440	
34	CHN	People's Republic of China	-A0010	M5000MRY4 FNL	5	6:51.654	
35	HUN	Hungary	-B0010	M5000MRY4 FNL	1	6:39.713	
36	NED	Netherlands	-B0010	M5000MRY4 FNL	2	6:39.780	
37	JPN	Japan	-B0010	M5000MRY4 FNL	3	6:40.545	
38	CAN	Canada	000100	XRELAY4 SFNL	1	2:36.808	QA
39	ITA	Italy	000100	XRELAY4 SFNL	2	2:36.895	QA
40	KAZ	Kazakhstan	000100	XRELAY4 SFNL	3	2:42.575	QB
41	NED	Netherlands	000100	XRELAY4 SFNL	4	2:51.919	QB
42	HUN	Hungary	000200	XRELAY4 SFNL	1	2:38.052	QA
43	CHN	People's Republic of China	000200	XRELAY4 SFNL	2	2:38.783	QA
44	RUS	RUS	000200	XRELAY4 SFNL	3	PEN	PEN
45	USA	United States of America	000200	XRELAY4 SFNL	4	PEN	PEN
46	NED	Netherlands	-A0010	W3000MRY4 FNL	1	4:03.409	
47	KOR	Republic of Korea	-A0010	W3000MRY4 FNL	2	4:03.627	
48	CHN	People's Republic of China	-A0010	W3000MRY4 FNL	3	4:03.863	
49	CAN	Canada	-A0010	W3000MRY4 FNL	4	4:04.329	
50	ITA	Italy	-B0010	W3000MRY4 FNL	1	4:09.688	
51	POL	Poland	-B0010	W3000MRY4 FNL	2	4:10.210	
52	RUS	RUS	-B0010	W3000MRY4 FNL	3	PEN	PEN
53	USA	United States of America	-B0010	W3000MRY4 FNL	4	PEN	PEN
54	CHN	People's Republic of China	-A0010	XRELAY4 FNL	1	2:37.348	
55	ITA	Italy	-A0010	XRELAY4 FNL	2	2:37.364	
56	HUN	Hungary	-A0010	XRELAY4 FNL	3	2:40.900	
57	CAN	Canada	-A0010	XRELAY4 FNL	4	PEN	PEN

Time

Having cleaned the qualified column, the next is 'time'. In such kinds of racing games, time is undoubtedly the most important statistic. It is the principle of ranking. To enable us to run more ranking data, and also to get rid of the penalty notes such as 'PEN' and 'No Time', we generated a new column from the original 'time' column, namely the 'timestamp'. Each row will then be ranked on the timestamp data.

In [85]:
```
# check there are only two types of time format in time column
w_m_game_df[~w_m_game_df['time'].apply(lambda x: ':' in x or '.' in x)]
```

Out[85]:

	country	helmet_number	name	group	game	rank	time	qualified	id
1	CAN	14	brunelle florence	000100-	W500M QFNL	5	PEN	PEN	STCAN22012200301
6	NED	6	velzeboer xandra	000200-	W500M QFNL	5	PEN	PEN	STNED20709200101
11	RUS	7	prosvirnova sofia	000300-	W500M QFNL	5	PEN	PEN	STRUS22012199701
15	USA	8	santos kristen	000400-	W500M QFNL	4	PEN	PEN	STUSA20211199401
16	ITA	13	valcepina martina	000400-	W500M QFNL	5	PEN	PEN	STITA20406199201
20	KOR	52	hwang daeheon	000100-	M1000M SFNL	4	PEN	PEN	STKOR10507199901
21	KOR	195	park jang hyuk	000100-	M1000M SFNL	5	DNS	DNS	STKOR13110199801
27	KOR	46	lee june seo	000200-	M1000M SFNL	6	PEN	PEN	STKOR10306200001
39	HUN	27	konya zsofia	000300-	W1000M HEAT	4	No Time	No Time	STHUN20602199501
43	RUS	7	prosvirnova sofia	000400-	W1000M HEAT	4	PEN	PEN	STRUS22012199701
47	CAN	35	boutin kim	000500-	W1000M HEAT	4	No Time	No Time	STCAN21612199401

	country	helmet_number	name	group	game	rank	time	qualified	id
55	FRA	20	huot marchand tifany	000700-	W1000M HEAT	4	PEN	PEN	STFRA21005199401
64	ITA	3	fontana arianna	-A00100	W1000M FNL	5	PEN	PEN	STITA21404199001
81	ITA	7	sighel pietro	000200-	M1500M QFNL	6	PEN	PEN	STITA11507199901
87	USA	23	pivirotto ryan	000300-	M1500M QFNL	6	PEN	PEN	STUSA11405199501
93	KAZ	17	nikisha denis	000400-	M1500M QFNL	6	PEN	PEN	STKAZ10708199501
99	CHN	206	zhang tianyi	000500-	M1500M QFNL	6	No Time	No Time	STCHN12508200401
110	CAN	72	dion pascal	000200-	M1000M QFNL	4	No Time	No Time	STCAN10808199401
111	JPN	59	yoshinaga kazuki	000200-	M1000M QFNL	5	PEN	PEN	STJPN13107199901
115	HUN	66	krueger john-henry	000300-	M1000M QFNL	4	PEN	PEN	STHUN12703199501
116	AUS	180	corey brendan	000300-	M1000M QFNL	5	PEN	PEN	STAUS12101199701
121	NED	14	knegt sjinkie	000400-	M1000M QFNL	5	YC	YC	STNED10507198901
136	ITA	7	sighel pietro	-A00100	M500M FNL	5	DNF	DNF	STITA11507199901
178	KOR	52	hwang daeheon	000200-	M500M SFNL	5	PEN	PEN	STKOR10507199901
198	USA	32	stoddard corinne	000500-	W500M HEAT	4	No Time	No Time	STUSA21508200101
225	JPN	65	kikuchi sumire	000200-	W1500M SFNL	6	No Time	ADVB	STJPN21501199601
226	HUN	10	jaszapati petra	000200-	W1500M SFNL	7	PEN	PEN	STHUN23112199801
278	NED	14	knegt sjinkie	000200-	M1500M SFNL	7	PEN	PEN	STNED10507198901
283	ITA	20	confortola yuri	000300-	M1500M SFNL	5	No Time	ADVA	STITA12404198601
284	CAN	6	hamelin charles	000300-	M1500M SFNL	6	PEN	PEN	STCAN11404198401
285	CHN	54	ren ziwei	000300-	M1500M SFNL	7	PEN	PEN	STCHN10306199701
297	GER	67	seidel anna	000200-	W1500M QFNL	6	PEN	PEN	STGER23103199801
301	CAN	46	blais danaé	000300-	W1500M QFNL	4	No Time	No Time	STCAN21005199901
302	HUN	27	konya zsofia	000300-	W1500M QFNL	5	PEN	PEN	STHUN20602199501
303	KAZ	72	tikhonova olga	000300-	W1500M QFNL	6	PEN	PEN	STKAZ22310199701
321	RUS	9	efremenkova ekaterina	000600-	W1500M QFNL	6	No Time	No Time	STRUS23112199701
331	CHN	41	qu chunyu	000200-	W500M SFNL	5	PEN	PEN	STCHN22007199601
355	RUS	57	ayrapetyan denis	000600-	M1000M HEAT	4	PEN	PEN	STRUS11701199701
359	BEL	9	desmet stijn	000700-	M1000M HEAT	4	PEN	PEN	STBEL11004199801
363	ITA	10	spechenhauser luca	000800-	M1000M HEAT	4	PEN	PEN	STITA11412200001
383	CAN	67	pierre-gilles jordan	000400-	M500M QFNL	5	No Time	No Time	STCAN12405199801
388	HUN	2	liu shaolin sandor	-A00100	M1000M FNL	5	YC	YC	STHUN12011199501
395	KOR	46	lee june seo	000100-	M500M HEAT	4	PEN	PEN	STKOR10306200001
399	CAN	11	laoun maxime	000200-	M500M HEAT	4	No Time	No Time	STCAN11208199601
415	NED	5	de laat itzhak	000600-	M500M HEAT	4	No Time	No Time	STNED11306199401
419	NED	122	van 't wout jens	000700-	M500M HEAT	4	PEN	PEN	STNED10610200101
423	NED	8	hoogerwerf dylan	000800-	M500M HEAT	4	No Time	No Time	STNED10908199502
426	KOR	195	park jang hyuk	000100-	M1000M QFNL	3	No Time	ADV	STKOR13110199801
427	ITA	7	sighel pietro	000100-	M1000M QFNL	4	PEN	PEN	STITA11507199901
428	CAN	67	pierre-gilles jordan	000100-	M1000M QFNL	5	PEN	PEN	STCAN12405199801

In [86]:
```
# check there are only two types of time format in time column
relay_df[~relay_df['time'].apply(lambda x: ':' in x or '.' in x)]
```

Out[86]:

	country	name	group	game	rank	time	qualified
44	RUS	RUS	000200	XRELAY4 SFNL	3	PEN	PEN
45	USA	United States of America	000200	XRELAY4 SFNL	4	PEN	PEN
52	RUS	RUS	-B0010	W3000MRY4 FNL	3	PEN	PEN
53	USA	United States of America	-B0010	W3000MRY4 FNL	4	PEN	PEN
57	CAN	Canada	-A0010	XRELAY4 FNL	4	PEN	PEN

In [87]:
```
def timestamp(x):
    '''
    convert the string of time into timestamp format
    '''
    if ':' in x:
        return datetime.strptime(x,'%M:%S.%f').timestamp()
    elif  '.' in x:
        return datetime.strptime(x, '%S.%f').timestamp()
```

```
    else:
        return 0
```

In [88]:
```
# apply the function
w_m_game_df['timestamp'] = w_m_game_df['time'].apply(timestamp)
make_checkpoint()
relay_df['timestamp'] = relay_df['time'].apply(timestamp)
make_checkpoint()
```

Game and Rank

The last step before we could generate a more comprehensive ranking is to separate and specify game level from game types. In our original dataset, both data are contained in the 'game' column. For example, 'W500M SFNL' means the semi-final race of the women's 500 meters game. In this step, we separated 'W500M' as 'game_type' and 'SFNL' as a game level. This enables us to group data by either game type or level and generates rankings respectively for analysis.

In [89]:
```
# split the game column into two columns game_type and level
w_m_game_df[['game_type', 'level']] = w_m_game_df.game.str.split(expand = True)
make_checkpoint()
relay_df[['game_type', 'level']] = relay_df.game.str.split(expand = True)
make_checkpoint()
```

In [90]:
```
# create a new column and filled with 0
w_m_game_df['rank_by_game'] = 0
relay_df['rank_by_game'] = 0
```

Based on all the aforementioned processing, we were able to generate a 'rank_by_game' column from the timestamp and group by game type and levels. Different from the 'rank' column in the original dataset which indicates only the ranking within the small group, this new ranking data would allow us to evaluate each athlete's performance from a full-scaled point of view.

In [91]:
```
def rank_game(df):
    '''
    function to get the overall rank for each game and event
    '''
    temp_other = df[df['level']!='FNL']
    temp_other['rank_by_game'] = temp_other.groupby(['level', 'game_type'])['timestamp'].rank(ascending = True).astype(np.int64)

    temp_fnl = df[df['level'] == 'FNL']
    list_game = set(temp_fnl['game_type'])
    for g in list_game:
        index_A = temp_fnl[(temp_fnl['game_type'] == g) & (temp_fnl['group'].apply(lambda x: 'A' in x))].index
        index_B = temp_fnl[(temp_fnl['game_type'] == g) & (temp_fnl['group'].apply(lambda x: 'B' in x))].sort_values('rank').index
        max_rank_groupA = int(temp_fnl[(temp_fnl['game_type'] == g) & (temp_fnl['group'].apply(lambda x: 'A' in x))]['rank'].max())
        rank_B = [i+max_rank_groupA for i in range(1,len(index_B)+1)]

        temp_fnl['rank_by_game'].loc[index_A] = temp_fnl['rank'].loc[index_A]
        temp_fnl['rank_by_game'].loc[index_B] = rank_B

    return temp_other.append(temp_fnl)
```

In [92]:
```
# apply the function
w_m_game_df = rank_game(w_m_game_df)
make_checkpoint()
relay_df = rank_game(relay_df)
make_checkpoint()
```

```
<ipython-input-91-1aa9741a83af>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp_other['rank_by_game'] = temp_other.groupby(['level', 'game_type'])['timestamp'].rank(ascending = True).astype(np.int64)
/opt/anaconda/envs/Python3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
/opt/anaconda/envs/Python3/lib/python3.8/site-packages/pandas/core/indexing.py:692: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  iloc._setitem_with_indexer(indexer, value, self.name)
```

In [93]:
```
# convert the type into integer
w_m_game_df.rank_by_game = w_m_game_df.rank_by_game.astype(np.int64)
make_checkpoint()
relay_df.rank_by_game = relay_df.rank_by_game.astype(np.int64)
make_checkpoint()
```

In [94]:
```
w_m_game_df
```

Out[94]:

	country	helmet_number	name	group	game	rank	time	qualified	id	timestamp	game_type	level	ran
0	CAN	50	charles alyson	000100-	W500M QFNL	4	1:07.206	ADV	STCAN23010199801	-2.208989e+09	W500M	QFNL	
1	CAN	14	brunelle florence	000100-	W500M QFNL	5	PEN	PEN	STCAN22012200301	0.000000e+00	W500M	QFNL	
2	HUN	10	jaszapati petra	000200-	W500M QFNL	1	43.476	Q	STHUN23112199801	-2.208989e+09	W500M	QFNL	
3	RUS	141	seregina elena	000200-	W500M QFNL	2	43.712	Q	STRUS23012200101	-2.208989e+09	W500M	QFNL	
4	USA	52	biney maame	000200-	W500M QFNL	3	46.099		STUSA22801200001	-2.208989e+09	W500M	QFNL	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
387	CHN	48	wu dajing	-A00100	M1000M FNL	4	1:42.937		STCHN12407199401	-2.208989e+09	M1000M	FNL	
388	HUN	2	liu shaolin sandor	-A00100	M1000M FNL	5	YC	YC	STHUN12011199501	0.000000e+00	M1000M	FNL	
389	NED	5	de laat itzhak	-B00100	M1000M FNL	1	1:35.925		STNED11306199401	-2.208989e+09	M1000M	FNL	
390	TUR	32	akar furkan	-B00100	M1000M FNL	2	1:36.052		STTUR10603200201	-2.208989e+09	M1000M	FNL	
391	USA	19	heo andrew	-B00100	M1000M FNL	3	1:36.140		STUSA10705200101	-2.208989e+09	M1000M	FNL	

429 rows × 13 columns

In [95]:
```
relay_df
```

Out[95]:

	country	name	group	game	rank	time	qualified	timestamp	game_type	level	rank_by_game
0	KOR	Republic of Korea	000200	M5000MRY4 SFNL	1	6:37.879	QA	-2.208988e+09	M5000MRY4	SFNL	1
1	RUS	RUS	000200	M5000MRY4 SFNL	2	6:37.925	QA	-2.208988e+09	M5000MRY4	SFNL	2
2	NED	Netherlands	000200	M5000MRY4 SFNL	3	6:37.927	QB	-2.208988e+09	M5000MRY4	SFNL	3
3	HUN	Hungary	000200	M5000MRY4 SFNL	4	6:45.172	QB	-2.208988e+09	M5000MRY4	SFNL	7
4	CHN	People's Republic of China	000100	XRELAY4 QFNL	1	2:37.535	Q	-2.208989e+09	XRELAY4	QFNL	3
5	ITA	Italy	000100	XRELAY4 QFNL	2	2:38.308	Q	-2.208989e+09	XRELAY4	QFNL	4
6	KOR	Republic of Korea	000100	XRELAY4 QFNL	3	2:48.308		-2.208989e+09	XRELAY4	QFNL	10
7	POL	Poland	000100	XRELAY4 QFNL	4	2:50.513		-2.208989e+09	XRELAY4	QFNL	11
8	NED	Netherlands	000200	XRELAY4 QFNL	1	2:36.437	Q	-2.208989e+09	XRELAY4	QFNL	1
9	CAN	Canada	000200	XRELAY4 QFNL	2	2:36.747	Q	-2.208989e+09	XRELAY4	QFNL	2
10	KAZ	Kazakhstan	000200	XRELAY4 QFNL	3	2:43.004	q	-2.208989e+09	XRELAY4	QFNL	9
11	FRA	France	000200	XRELAY4 QFNL	4	2:51.221		-2.208989e+09	XRELAY4	QFNL	12
12	HUN	Hungary	000300	XRELAY4 QFNL	1	2:38.396	Q	-2.208989e+09	XRELAY4	QFNL	5
13	RUS	RUS	000300	XRELAY4 QFNL	2	2:38.445	Q	-2.208989e+09	XRELAY4	QFNL	6
14	USA	United States of America	000300	XRELAY4 QFNL	3	2:39.043	q	-2.208989e+09	XRELAY4	QFNL	7
15	JPN	Japan	000300	XRELAY4 QFNL	4	2:39.112		-2.208989e+09	XRELAY4	QFNL	8
18	NED	Netherlands	000100	W3000MRY4 SFNL	1	4:04.133	QA	-2.208989e+09	W3000MRY4	SFNL	1
19	CHN	People's Republic of China	000100	W3000MRY4 SFNL	2	4:04.383	QA	-2.208989e+09	W3000MRY4	SFNL	2
20	POL	Poland	000100	W3000MRY4 SFNL	3	4:10.074	QB	-2.208989e+09	W3000MRY4	SFNL	7
21	ITA	Italy	000100	W3000MRY4 SFNL	4	4:17.438	QB	-2.208989e+09	W3000MRY4	SFNL	8
22	CAN	Canada	000200	W3000MRY4 SFNL	1	4:05.893	QA	-2.208989e+09	W3000MRY4	SFNL	3
23	KOR	Republic of Korea	000200	W3000MRY4 SFNL	2	4:05.904	QA	-2.208989e+09	W3000MRY4	SFNL	4
24	RUS	RUS	000200	W3000MRY4 SFNL	3	4:06.064	QB	-2.208989e+09	W3000MRY4	SFNL	5
25	USA	United States of America	000200	W3000MRY4 SFNL	4	4:06.098	QB	-2.208989e+09	W3000MRY4	SFNL	6

	country	name	group	game	rank	time	qualified	timestamp	game_type	level	rank_by_game
26	CAN	Canada	000100	M5000MRY4 SFNL	1	6:38.752	QA	-2.208988e+09	M5000MRY4	SFNL	4
27	ITA	Italy	000100	M5000MRY4 SFNL	2	6:38.899	QA	-2.208988e+09	M5000MRY4	SFNL	5
28	JPN	Japan	000100	M5000MRY4 SFNL	3	6:40.446	QB	-2.208988e+09	M5000MRY4	SFNL	6
29	CHN	People's Republic of China	000100	M5000MRY4 SFNL	4	6:51.040	ADVA	-2.208988e+09	M5000MRY4	SFNL	8
38	CAN	Canada	000100	XRELAY4 SFNL	1	2:36.808	QA	-2.208989e+09	XRELAY4	SFNL	1
39	ITA	Italy	000100	XRELAY4 SFNL	2	2:36.895	QA	-2.208989e+09	XRELAY4	SFNL	2
40	KAZ	Kazakhstan	000100	XRELAY4 SFNL	3	2:42.575	QB	-2.208989e+09	XRELAY4	SFNL	5
41	NED	Netherlands	000100	XRELAY4 SFNL	4	2:51.919	QB	-2.208989e+09	XRELAY4	SFNL	6
42	HUN	Hungary	000200	XRELAY4 SFNL	1	2:38.052	QA	-2.208989e+09	XRELAY4	SFNL	3
43	CHN	People's Republic of China	000200	XRELAY4 SFNL	2	2:38.783	QA	-2.208989e+09	XRELAY4	SFNL	4
44	RUS	RUS	000200	XRELAY4 SFNL	3	PEN	PEN	0.000000e+00	XRELAY4	SFNL	7
45	USA	United States of America	000200	XRELAY4 SFNL	4	PEN	PEN	0.000000e+00	XRELAY4	SFNL	7
16	NED	Netherlands	-B0010	XRELAY4 FNL	1	2:36.966		-2.208989e+09	XRELAY4	FNL	5
17	KAZ	Kazakhstan	-B0010	XRELAY4 FNL	2	2:44.148		-2.208989e+09	XRELAY4	FNL	6
30	CAN	Canada	-A0010	M5000MRY4 FNL	1	6:41.257		-2.208988e+09	M5000MRY4	FNL	1
31	KOR	Republic of Korea	-A0010	M5000MRY4 FNL	2	6:41.679		-2.208988e+09	M5000MRY4	FNL	2
32	ITA	Italy	-A0010	M5000MRY4 FNL	3	6:43.431		-2.208988e+09	M5000MRY4	FNL	3
33	RUS	RUS	-A0010	M5000MRY4 FNL	4	6:43.440		-2.208988e+09	M5000MRY4	FNL	4
34	CHN	People's Republic of China	-A0010	M5000MRY4 FNL	5	6:51.654		-2.208988e+09	M5000MRY4	FNL	5
35	HUN	Hungary	-B0010	M5000MRY4 FNL	1	6:39.713		-2.208988e+09	M5000MRY4	FNL	6
36	NED	Netherlands	-B0010	M5000MRY4 FNL	2	6:39.780		-2.208988e+09	M5000MRY4	FNL	7
37	JPN	Japan	-B0010	M5000MRY4 FNL	3	6:40.545		-2.208988e+09	M5000MRY4	FNL	8
46	NED	Netherlands	-A0010	W3000MRY4 FNL	1	4:03.409		-2.208989e+09	W3000MRY4	FNL	1
47	KOR	Republic of Korea	-A0010	W3000MRY4 FNL	2	4:03.627		-2.208989e+09	W3000MRY4	FNL	2
48	CHN	People's Republic of China	-A0010	W3000MRY4 FNL	3	4:03.863		-2.208989e+09	W3000MRY4	FNL	3
49	CAN	Canada	-A0010	W3000MRY4 FNL	4	4:04.329		-2.208989e+09	W3000MRY4	FNL	4
50	ITA	Italy	-B0010	W3000MRY4 FNL	1	4:09.688		-2.208989e+09	W3000MRY4	FNL	5
51	POL	Poland	-B0010	W3000MRY4 FNL	2	4:10.210		-2.208989e+09	W3000MRY4	FNL	6
52	RUS	RUS	-B0010	W3000MRY4 FNL	3	PEN	PEN	0.000000e+00	W3000MRY4	FNL	7
53	USA	United States of America	-B0010	W3000MRY4 FNL	4	PEN	PEN	0.000000e+00	W3000MRY4	FNL	8
54	CHN	People's Republic of China	-A0010	XRELAY4 FNL	1	2:37.348		-2.208989e+09	XRELAY4	FNL	1
55	ITA	Italy	-A0010	XRELAY4 FNL	2	2:37.364		-2.208989e+09	XRELAY4	FNL	2
56	HUN	Hungary	-A0010	XRELAY4 FNL	3	2:40.900		-2.208989e+09	XRELAY4	FNL	3
57	CAN	Canada	-A0010	XRELAY4 FNL	4	PEN	PEN	0.000000e+00	XRELAY4	FNL	4

The example of a ranked game is shown below.

In [96]:
```
# example of one of the game
w_m_game_df[(w_m_game_df['game_type'] == 'M1000M') & (w_m_game_df['level'] == 'QFNL')].sort_values('rank_by_game')
```

Out[96]:

	country	helmet_number	name	group	game	rank	time	qualified	id	timestamp	game_type	level	ra
107	KOR	46	lee june seo	000200-	M1000M QFNL	1	1:23.682	Q	STKOR10306200001	-2.208989e+09	M1000M	QFNL	
108	HUN	1	liu shaoang	000200-	M1000M QFNL	2	1:23.940	Q	STHUN11303199801	-2.208989e+09	M1000M	QFNL	
109	FRA	73	fercoq quentin	000200-	M1000M QFNL	3	1:24.411		STFRA10503199901	-2.208989e+09	M1000M	QFNL	
424	USA	19	heo andrew	000100-	M1000M QFNL	1	1:24.603	Q	STUSA10705200101	-2.208989e+09	M1000M	QFNL	
117	KOR	52	hwang daeheon	000400-	M1000M QFNL	1	1:24.693	Q	STKOR10507199901	-2.208989e+09	M1000M	QFNL	
112	TUR	32	akar furkan	000300-	M1000M QFNL	1	1:25.490	Q	STTUR10603200201	-2.208989e+09	M1000M	QFNL	
118	CHN	94	li wenlong	000400-	M1000M QFNL	2	1:30.550	Q	STCHN10402200101	-2.208989e+09	M1000M	QFNL	

	country	helmet_number	name	group	game	rank	time	qualified	id	timestamp	game_type	level	ra
425	CHN	48	wu dajing	000100-	M1000M QFNL	2	1:33.302	Q	STCHN12407199401	-2.208989e+09	M1000M	QFNL	
113	CHN	54	ren ziwei	000300-	M1000M QFNL	2	1:34.211	Q	STCHN10306199701	-2.208989e+09	M1000M	QFNL	
114	NED	5	de laat itzhak	000300-	M1000M QFNL	3	1:42.490	ADV	STNED11306199401	-2.208989e+09	M1000M	QFNL	
119	HUN	2	liu shaolin sandor	000400-	M1000M QFNL	3	1:55.248	ADV	STHUN12011199501	-2.208989e+09	M1000M	QFNL	
120	USA	23	pivirotto ryan	000400-	M1000M QFNL	4	2:08.364		STUSA11405199501	-2.208989e+09	M1000M	QFNL	
426	KOR	195	park jang hyuk	000100-	M1000M QFNL	3	No Time	ADV	STKOR13110199801	0.000000e+00	M1000M	QFNL	
121	NED	14	knegt sjinkie	000400-	M1000M QFNL	5	YC	YC	STNED10507198901	0.000000e+00	M1000M	QFNL	
116	AUS	180	corey brendan	000300-	M1000M QFNL	5	PEN	PEN	STAUS12101199701	0.000000e+00	M1000M	QFNL	
115	HUN	66	krueger john-henry	000300-	M1000M QFNL	4	PEN	PEN	STHUN12703199501	0.000000e+00	M1000M	QFNL	
111	JPN	59	yoshinaga kazuki	000200-	M1000M QFNL	5	PEN	PEN	STJPN13107199901	0.000000e+00	M1000M	QFNL	
110	CAN	72	dion pascal	000200-	M1000M QFNL	4	No Time	No Time	STCAN10808199401	0.000000e+00	M1000M	QFNL	
427	ITA	7	sighel pietro	000100-	M1000M QFNL	4	PEN	PEN	STITA11507199901	0.000000e+00	M1000M	QFNL	
428	CAN	67	pierre-gilles jordan	000100-	M1000M QFNL	5	PEN	PEN	STCAN12405199801	0.000000e+00	M1000M	QFNL	

In [97]:
```
# convert the 0s into timestamp format
w_m_game_df.timestamp = w_m_game_df.timestamp.replace(to_replace = 0 , value =datetime.strptime('0.0', '%S.%f').timestamp())
```

Countries

Beautiful Soup is used again for gathering country code and name data.

In [98]:
```
# get all the country information
URL =f"http://www.shorttrackonline.info/athletes.php?"
page = requests.get(URL)

soup = BeautifulSoup(page.content, "html.parser")
soup_body = str(soup.body)
dict_country = {i.split('">')[0]:i.split('">')[1] for i in re.findall(r'country=(.*)</a', soup_body)}
```

In [99]:
```
# convert the dictionary into data frame
countries = pd.DataFrame({'country_code':dict_country.keys(), 'name': dict_country.values()})
make_checkpoint()
```

In [100]:
```
countries
```

Out[100]:

	country_code	name
0	ARG	Argentina
1	KAZ	Kazakhstan
2	AUS	Australia
3	LAT	Latvia
4	AUT	Austria
5	LTU	Lithuania
6	BLR	Belarus
7	LUX	Luxembourg
8	BEL	Belgium
9	MAS	Malaysia
10	BIH	Bosnia and Herzegovina
11	MGL	Mongolia
12	BRA	Brazil
13	NED	Netherlands

	country_code	name
14	BUL	Bulgaria
15	NZL	New Zealand
16	CAN	Canada
17	NOR	Norway
18	CHN	China
19	PHI	Philippines
20	TPE	Chinese Taipei
21	POL	Poland
22	COL	Colombia
23	QAT	Qatar
24	CRO	Croatia
25	KOR	Republic of Korea
26	CZE	Czech Republic
27	ROU	Romania
28	PRK	D.P.R. Korea
29	RUS	Russia
30	DEN	Denmark
31	SRB	Serbia
32	EST	Estonia
33	SGP	Singapore
34	FIN	Finland
35	SVK	Slovak Republic
36	FRA	France
37	SLO	Slovenia
38	GER	Germany
39	RSA	South Africa
40	GBR	Great Britain
41	ESP	Spain
42	HKG	Hong Kong
43	SWE	Sweden
44	HUN	Hungary
45	SUI	Switzerland
46	IND	India
47	THA	Thailand
48	INA	Indonesia
49	TUR	Turkey
50	IRL	Ireland
51	USA	U.S.A.
52	ISR	Israel
53	UKR	Ukraine
54	ITA	Italy
55	UZB	Uzbekistan
56	JPN	Japan

4.1 Subsets the data frame

After all the process of data transformation, they are separated into each table in the schema. Spark is used to show the schema of each data frame and make changes to their data types.

In [101]:
```
Image("/project/DataEngineering/graphs/shorttrack.png", width = 900)
```

Out[101]:

In [102]:
```
# based on the schema get all the tables
heat = pd.DataFrame(w_m_game_df[w_m_game_df['level'] == 'HEAT'][['id', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game', 'qualified']].reset_i
qfnl = pd.DataFrame(w_m_game_df[w_m_game_df['level'] == 'QFNL'][['id', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game', 'qualified']].reset_i
sfnl = pd.DataFrame(w_m_game_df[w_m_game_df['level'] == 'SFNL'][['id', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game', 'qualified']].reset_in
fnl = pd.DataFrame(w_m_game_df[w_m_game_df['level'] == 'FNL'][['id', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game']].reset_index(drop = T
make_checkpoint()
```

In [103]:
```
# change the columns name
heat.columns = ['id', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank', 'qualified']
qfnl.columns = ['id', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank', 'qualified']
sfnl.columns = ['id', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank', 'qualified']
fnl.columns = ['id', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank']
make_checkpoint()
```

In [104]:
```
heat
```

Out[104]:

	id	game_type	group	time	inGroup_rank	game_rank	qualified
0	STKOR20909199801	W1000M	000100-	-2.208989e+09	1	9	Q
1	STNED21405199901	W1000M	000100-	-2.208989e+09	2	10	Q
2	STRUS22408200201	W1000M	000100-	-2.208989e+09	3	12	q
3	STGBR22601199601	W1000M	000100-	-2.208989e+09	4	22	
4	STNED22509199701	W1000M	000200-	-2.208989e+09	1	1	Q
...	...	...	...	...	...	...	...
123	STNED10610200101	M500M	000700-	-2.208989e+09	4	30	PEN
124	STCHN12407199401	M500M	000800-	-2.208989e+09	1	1	Q
125	STITA11507199901	M500M	000800-	-2.208989e+09	2	3	Q

	id	game_type	group	time	inGroup_rank	game_rank	qualified
126	STHKG12207199901	M500M	000800-	-2.208989e+09	3	27	
127	STNED10908199502	M500M	000800-	-2.208989e+09	4	30	No Time

128 rows × 7 columns

In [105]:
```
qfnl
```

Out[105]:

	id	game_type	group	time	inGroup_rank	game_rank	qualified
0	STCAN23010199801	W500M	000100-	-2.208989e+09	4	14	ADV
1	STCAN22012200301	W500M	000100-	-2.208989e+09	5	18	PEN
2	STHUN23112199801	W500M	000200-	-2.208989e+09	1	7	Q
3	STRUS23012200101	W500M	000200-	-2.208989e+09	2	8	Q
4	STUSA22801200001	W500M	000200-	-2.208989e+09	3	9	
...	...	...	...	...	...	...	...
147	STUSA10705200101	M1000M	000100-	-2.208989e+09	1	4	Q
148	STCHN12407199401	M1000M	000100-	-2.208989e+09	2	8	Q
149	STKOR13110199801	M1000M	000100-	-2.208989e+09	3	16	ADV
150	STITA11507199901	M1000M	000100-	-2.208989e+09	4	16	PEN
151	STCAN12405199801	M1000M	000100-	-2.208989e+09	5	16	PEN

152 rows × 7 columns

In [106]:
```
sfnl
```

Out[106]:

	id	game_type	group	time	inGroup_rank	game_rank	qualified
0	STCHN10306199701	M1000M	000100-	-2.208989e+09	1	5	QA
1	STCHN10402200101	M1000M	000100-	-2.208989e+09	2	6	QA
2	STTUR10603200201	M1000M	000100-	-2.208989e+09	3	7	QB
3	STKOR10507199901	M1000M	000100-	-2.208989e+09	4	10	PEN
4	STKOR13110199801	M1000M	000100-	-2.208989e+09	5	10	DNS
...	...	...	...	...	...	...	...
79	STNED22509199701	W500M	000200-	-2.208989e+09	1	2	QA
80	STCHN20408199901	W500M	000200-	-2.208989e+09	2	6	QA
81	STHUN23112199801	W500M	000200-	-2.208989e+09	3	7	QB
82	STBEL22610199601	W500M	000200-	-2.208989e+09	4	9	ADVA
83	STCHN22007199601	W500M	000200-	-2.208989e+09	5	10	PEN

84 rows × 7 columns

In [107]:
```
fnl
```

Out[107]:

	id	game_type	group	time	inGroup_rank	game_rank
0	STNED22509199701	W1000M	-A00100	-2.208989e+09	1	1
1	STKOR20909199801	W1000M	-A00100	-2.208989e+09	2	2
2	STBEL22610199601	W1000M	-A00100	-2.208989e+09	3	3
3	STUSA20211199401	W1000M	-A00100	-2.208989e+09	4	4
4	STITA21404199001	W1000M	-A00100	-2.208989e+09	5	5
...	...	...	...	...	...	...
60	STCHN12407199401	M1000M	-A00100	-2.208989e+09	4	4
61	STHUN12011199501	M1000M	-A00100	-2.208989e+09	5	5
62	STNED11306199401	M1000M	-B00100	-2.208989e+09	1	6
63	STTUR10603200201	M1000M	-B00100	-2.208989e+09	2	7
64	STUSA10705200101	M1000M	-B00100	-2.208989e+09	3	8

65 rows × 6 columns

In [108]:
```
# based on the schema and get all the tables
relay_qfnl = pd.DataFrame(relay_df[relay_df['level'] == 'QFNL'][['country', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game', 'qualified']].reset_in
relay_sfnl = pd.DataFrame(relay_df[relay_df['level'] == 'SFNL'][['country', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game', 'qualified']].reset_inc
```

```
relay_fnl = pd.DataFrame(relay_df[relay_df['level'] == 'FNL'][['country', 'game_type', 'group', 'timestamp', 'rank', 'rank_by_game']].reset_index(drop = Tr
make_checkpoint()
```

In [109]:

```
# change the columns name
relay_qfnl.columns = ['country_code', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank', 'qualified']
relay_sfnl.columns = ['country_code', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank', 'qualified']
relay_fnl.columns = ['country_code', 'game_type', 'group', 'time', 'inGroup_rank', 'game_rank']
make_checkpoint()
```

In [110]:

```
relay_qfnl
```

Out[110]:

	country_code	game_type	group	time	inGroup_rank	game_rank	qualified
0	CHN	XRELAY4	000100	-2.208989e+09	1	3	Q
1	ITA	XRELAY4	000100	-2.208989e+09	2	4	Q
2	KOR	XRELAY4	000100	-2.208989e+09	3	10	
3	POL	XRELAY4	000100	-2.208989e+09	4	11	
4	NED	XRELAY4	000200	-2.208989e+09	1	1	Q
5	CAN	XRELAY4	000200	-2.208989e+09	2	2	Q
6	KAZ	XRELAY4	000200	-2.208989e+09	3	9	q
7	FRA	XRELAY4	000200	-2.208989e+09	4	12	
8	HUN	XRELAY4	000300	-2.208989e+09	1	5	Q
9	RUS	XRELAY4	000300	-2.208989e+09	2	6	Q
10	USA	XRELAY4	000300	-2.208989e+09	3	7	q
11	JPN	XRELAY4	000300	-2.208989e+09	4	8	

In [111]:

```
relay_sfnl
```

Out[111]:

	country_code	game_type	group	time	inGroup_rank	game_rank	qualified
0	KOR	M5000MRY4	000200	-2.208988e+09	1	1	QA
1	RUS	M5000MRY4	000200	-2.208988e+09	2	2	QA
2	NED	M5000MRY4	000200	-2.208988e+09	3	3	QB
3	HUN	M5000MRY4	000200	-2.208988e+09	4	7	QB
4	NED	W3000MRY4	000100	-2.208989e+09	1	1	QA
5	CHN	W3000MRY4	000100	-2.208989e+09	2	2	QA
6	POL	W3000MRY4	000100	-2.208989e+09	3	7	QB
7	ITA	W3000MRY4	000100	-2.208989e+09	4	8	QB
8	CAN	W3000MRY4	000200	-2.208989e+09	1	3	QA
9	KOR	W3000MRY4	000200	-2.208989e+09	2	4	QA
10	RUS	W3000MRY4	000200	-2.208989e+09	3	5	QB
11	USA	W3000MRY4	000200	-2.208989e+09	4	6	QB
12	CAN	M5000MRY4	000100	-2.208988e+09	1	4	QA
13	ITA	M5000MRY4	000100	-2.208988e+09	2	5	QA
14	JPN	M5000MRY4	000100	-2.208988e+09	3	6	QB
15	CHN	M5000MRY4	000100	-2.208988e+09	4	8	ADVA
16	CAN	XRELAY4	000100	-2.208989e+09	1	1	QA
17	ITA	XRELAY4	000100	-2.208989e+09	2	2	QA
18	KAZ	XRELAY4	000100	-2.208989e+09	3	5	QB
19	NED	XRELAY4	000100	-2.208989e+09	4	6	QB
20	HUN	XRELAY4	000200	-2.208989e+09	1	3	QA
21	CHN	XRELAY4	000200	-2.208989e+09	2	4	QA
22	RUS	XRELAY4	000200	0.000000e+00	3	7	PEN
23	USA	XRELAY4	000200	0.000000e+00	4	7	PEN

In [112]:

```
relay_fnl
```

Out[112]:

	country_code	game_type	group	time	inGroup_rank	game_rank
0	NED	XRELAY4	-B0010	-2.208989e+09	1	5
1	KAZ	XRELAY4	-B0010	-2.208989e+09	2	6
2	CAN	M5000MRY4	-A0010	-2.208988e+09	1	1

	country_code	game_type	group	time	inGroup_rank	game_rank
3	KOR	M5000MRY4	-A0010	-2.208988e+09	2	2
4	ITA	M5000MRY4	-A0010	-2.208988e+09	3	3
5	RUS	M5000MRY4	-A0010	-2.208988e+09	4	4
6	CHN	M5000MRY4	-A0010	-2.208988e+09	5	5
7	HUN	M5000MRY4	-B0010	-2.208988e+09	1	6
8	NED	M5000MRY4	-B0010	-2.208988e+09	2	7
9	JPN	M5000MRY4	-B0010	-2.208988e+09	3	8
10	NED	W3000MRY4	-A0010	-2.208989e+09	1	1
11	KOR	W3000MRY4	-A0010	-2.208989e+09	2	2
12	CHN	W3000MRY4	-A0010	-2.208989e+09	3	3
13	CAN	W3000MRY4	-A0010	-2.208989e+09	4	4
14	ITA	W3000MRY4	-B0010	-2.208989e+09	1	5
15	POL	W3000MRY4	-B0010	-2.208989e+09	2	6
16	RUS	W3000MRY4	-B0010	0.000000e+00	3	7
17	USA	W3000MRY4	-B0010	0.000000e+00	4	8
18	CHN	XRELAY4	-A0010	-2.208989e+09	1	1
19	ITA	XRELAY4	-A0010	-2.208989e+09	2	2
20	HUN	XRELAY4	-A0010	-2.208989e+09	3	3
21	CAN	XRELAY4	-A0010	0.000000e+00	4	4

In [113]:
```
# get the information needed for athlete information
athlete = pd.DataFrame(athlete_info_df[['id','name', 'country', 'birth_year', 'age_category', 'club']])
make_checkpoint()
```

In [114]:
```
# change the columns
athlete.columns = ['id', 'name', 'country_code', 'birth_year', 'age_category', 'club']
make_checkpoint()
```

In [115]:
```
# change the data type into int type
athlete.birth_year = athlete.birth_year.astype(np.int64)
```

In [116]:
```
# convert all the data frame into spark data frame
heat_df = spark.createDataFrame(heat)
qfnl_df = spark.createDataFrame(qfnl)
sfnl_df = spark.createDataFrame(sfnl)
fnl_df = spark.createDataFrame(fnl)
relay_qfnl_df = spark.createDataFrame(relay_qfnl)
relay_sfnl_df = spark.createDataFrame(relay_sfnl)
relay_fnl_df = spark.createDataFrame(relay_fnl)
countries_df = spark.createDataFrame(countries)
athlete_df = spark.createDataFrame(athlete)
make_checkpoint()
```

In [117]:
```
heat_df.printSchema()
qfnl_df.printSchema()
sfnl_df.printSchema()
fnl_df.printSchema()
relay_qfnl_df.printSchema()
relay_sfnl_df.printSchema()
relay_fnl_df.printSchema()
countries_df.printSchema()
athlete_df.printSchema()
```

```
root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)
 |-- qualified: string (nullable = true)
```

```
root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: double (nullable = true)
 |-- inGroup_rank: string (nullable = true)
 |-- game_rank: long (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- name: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- country_code: string (nullable = true)
 |-- birth_year: long (nullable = true)
 |-- age_category: string (nullable = true)
 |-- club: string (nullable = true)
```

In [118]:

```
def change_data_type(df, col, type_):
    '''
    function to change the data type for spark data frame
    '''
    return df.withColumn(col,  df[col].cast(type_))
```

In [119]:

```
athlete_df = change_data_type(athlete_df, 'birth_year', 'int')
heat_df = change_data_type(heat_df, 'time', 'timestamp')
heat_df = change_data_type(heat_df, 'inGroup_rank', 'int')
heat_df = change_data_type(heat_df, 'game_rank', 'int')
make_checkpoint()

qfnl_df = change_data_type(qfnl_df, 'time', 'timestamp')
qfnl_df = change_data_type(qfnl_df, 'inGroup_rank', 'int')
qfnl_df = change_data_type(qfnl_df, 'game_rank', 'int')
make_checkpoint()

sfnl_df = change_data_type(sfnl_df, 'time', 'timestamp')
sfnl_df = change_data_type(sfnl_df, 'inGroup_rank', 'int')
sfnl_df = change_data_type(sfnl_df, 'game_rank', 'int')
make_checkpoint()

fnl_df = change_data_type(fnl_df, 'time', 'timestamp')
fnl_df = change_data_type(fnl_df, 'inGroup_rank', 'int')
fnl_df = change_data_type(fnl_df, 'game_rank', 'int')
make_checkpoint()

relay_qfnl_df = change_data_type(relay_qfnl_df, 'time', 'timestamp')
relay_qfnl_df = change_data_type(relay_qfnl_df, 'inGroup_rank', 'int')
relay_qfnl_df = change_data_type(relay_qfnl_df, 'game_rank', 'int')
make_checkpoint()
```

```
relay_sfnl_df = change_data_type(relay_sfnl_df, 'time', 'timestamp')
relay_sfnl_df = change_data_type(relay_sfnl_df, 'inGroup_rank', 'int')
relay_sfnl_df = change_data_type(relay_sfnl_df, 'game_rank', 'int')
make_checkpoint()

relay_fnl_df = change_data_type(relay_fnl_df, 'time', 'timestamp')
relay_fnl_df = change_data_type(relay_fnl_df, 'inGroup_rank', 'int')
relay_fnl_df = change_data_type(relay_fnl_df, 'game_rank', 'int')
make_checkpoint()
```

In [120]:

```
heat_df.printSchema()
qfnl_df.printSchema()
sfnl_df.printSchema()
fnl_df.printSchema()
relay_qfnl_df.printSchema()
relay_sfnl_df.printSchema()
relay_fnl_df.printSchema()
countries_df.printSchema()
athlete_df.printSchema()
```

```
root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)
 |-- qualified: string (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- group: string (nullable = true)
 |-- time: timestamp (nullable = true)
 |-- inGroup_rank: integer (nullable = true)
 |-- game_rank: integer (nullable = true)

root
 |-- country_code: string (nullable = true)
 |-- name: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
```

```
|-- country_code: string (nullable = true)
|-- birth_year: integer (nullable = true)
|-- age_category: string (nullable = true)
|-- club: string (nullable = true)
```

In [121]:

```
# convert all data frames into parquet files
heat_df.write.parquet("/project/DataEngineering/parquet_files/heat.parquet", mode = 'overwrite')
qfnl_df.write.parquet("/project/DataEngineering/parquet_files/qfnl.parquet", mode = 'overwrite')
sfnl_df.write.parquet("/project/DataEngineering/parquet_files/sfnl.parquet", mode = 'overwrite')
fnl_df.write.parquet("/project/DataEngineering/parquet_files/fnl.parquet", mode = 'overwrite')
relay_qfnl_df.write.parquet("/project/DataEngineering/parquet_files/relay_qfnl.parquet", mode = 'overwrite')
relay_sfnl_df.write.parquet("/project/DataEngineering/parquet_files/relay_sfnl.parquet", mode = 'overwrite')
relay_fnl_df.write.parquet("/project/DataEngineering/parquet_files/relay_fnl.parquet", mode = 'overwrite')
countries_df.write.parquet("/project/DataEngineering/parquet_files/countries.parquet", mode = 'overwrite')
athlete_df.write.parquet("/project/DataEngineering/parquet_files/athlete.parquet", mode = 'overwrite')
make_checkpoint()
```

5.0 Write into PostgreSQL database

The schema is written into the database by a line of shell script and the selected data are all written into the database using spark.write.JDBC.

In [122]:

```
# create the schema
!PGPASSWORD=qwerty123 psql -h depgdb.crhso94tou3n.eu-west-2.rds.amazonaws.com -d haiyunzou21 -U haiyunzou21 -c '\i shorttrack.sql'
```

```
        .::::.     .::.
      ....yy:    .yy.
      :.  .yy.    y.
          :y:   .:
          .yy  .:
           yy..:
           :y:.
           .y.
          .:.
      .....:.
      ::::.

• Project files and data should be stored in /project. This is shared among everyone
  in the project.
• Personal files and configuration should be stored in /home/faculty.
• Files outside /project and /home/faculty will be lost when this server is terminated.
• Create custom environments to setup your servers reproducibly.

psql:shorttrack.sql:1: NOTICE:  drop cascades to 10 other objects
DETAIL:  drop cascades to table shorttrack.countries
drop cascades to table shorttrack.athlete
drop cascades to table shorttrack.relay_qfnl
drop cascades to table shorttrack.relay_sfnl
drop cascades to table shorttrack.relay_fnl
drop cascades to table shorttrack.heat
drop cascades to table shorttrack.qfnl
drop cascades to table shorttrack.sfnl
drop cascades to table shorttrack.fnl
drop cascades to table shorttrack.tweets
DROP SCHEMA
CREATE SCHEMA
psql:shorttrack.sql:5: NOTICE:  table "athlete" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:6: NOTICE:  table "countries" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:7: NOTICE:  table "relay_qfnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:8: NOTICE:  table "relay_sfnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:9: NOTICE:  table "relay_fnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:10: NOTICE:  table "heat" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:11: NOTICE:  table "qfnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:12: NOTICE:  table "sfnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:13: NOTICE:  table "fnl" does not exist, skipping
DROP TABLE
psql:shorttrack.sql:14: NOTICE:  table "tweets" does not exist, skipping
DROP TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

In [123]:
```
# read all the tables
heat = spark.read.parquet("/project/DataEngineering/parquet_files/heat.parquet")
qfnl = spark.read.parquet("/project/DataEngineering/parquet_files/qfnl.parquet")
sfnl = spark.read.parquet("/project/DataEngineering/parquet_files/sfnl.parquet")
fnl = spark.read.parquet("/project/DataEngineering/parquet_files/fnl.parquet")
relay_qfnl = spark.read.parquet("/project/DataEngineering/parquet_files/relay_qfnl.parquet")
relay_sfnl = spark.read.parquet("/project/DataEngineering/parquet_files/relay_sfnl.parquet")
relay_fnl = spark.read.parquet("/project/DataEngineering/parquet_files/relay_fnl.parquet")
countries = spark.read.parquet("/project/DataEngineering/parquet_files/countries.parquet")
athlete = spark.read.parquet("/project/DataEngineering/parquet_files/athlete.parquet")
make_checkpoint()
```

In [124]:
```
# information for log into postgresql
postgres_uri = "jdbc:postgresql://depgdb.crhso94tou3n.eu-west-2.rds.amazonaws.com:5432/haiyunzou21"
user = "haiyunzou21"
password = "qwerty123"
```

In [125]:
```
# write each table into the database
countries.write.jdbc(url=postgres_uri, table="shorttrack.countries", mode="append", properties={"user":user, "password": password, "driver": "org.pos
make_checkpoint()
athlete.write.jdbc(url=postgres_uri, table="shorttrack.athlete", mode="append", properties={"user":user, "password": password, "driver": "org.postgres
make_checkpoint()
heat.write.jdbc(url=postgres_uri, table="shorttrack.heat", mode="append", properties={"user":user, "password": password, "driver": "org.postgresql.Dr
make_checkpoint()
qfnl.write.jdbc(url=postgres_uri, table="shorttrack.qfnl", mode="append", properties={"user":user, "password": password, "driver": "org.postgresql.Dri
make_checkpoint()
sfnl.write.jdbc(url=postgres_uri, table="shorttrack.sfnl", mode="append", properties={"user":user, "password": password, "driver": "org.postgresql.Driv
make_checkpoint()
fnl.write.jdbc(url=postgres_uri, table="shorttrack.fnl", mode="append", properties={"user":user, "password": password, "driver": "org.postgresql.Driver
make_checkpoint()
relay_qfnl.write.jdbc(url=postgres_uri, table="shorttrack.relay_qfnl", mode="append", properties={"user":user, "password": password, "driver": "org.po
make_checkpoint()
relay_sfnl.write.jdbc(url=postgres_uri, table="shorttrack.relay_sfnl", mode="append", properties={"user":user, "password": password, "driver": "org.pos
make_checkpoint()
relay_fnl.write.jdbc(url=postgres_uri, table="shorttrack.relay_fnl", mode="append", properties={"user":user, "password": password, "driver": "org.postg
make_checkpoint()
```

5.1 SQL query

As identified at the beginning, we intended to build a data querying system which can help with the decision making for each national STK team in their training and gaming strategy. In this section, two SQL query examples are given to illustrate such utilities.

The first SQL query is used to see the time and ranking range for each country in each game type. This query can be used to check if the strength of each athlete in the country is evenly distributed. For example, if the time and rank range are both very small, this country has a relatively even team where the no athletes perform extremely outstanding or especially poor. On the other hand, if the time and rank range are both very large, this may suggest that the team might have one or more very talented athlete on this particular game where he or she can largely outperform their colleges. This data can be a very useful guidance for the daily training within each team as it allows the coaches to see the team's strength and weaknesses and adjust training strategy accordingly.

In [126]:
```
sql1 = '''
SELECT country_code, game_type, EXTRACT(EPOCH FROM (max(time) - min(time))) AS time_difference, (max(game_rank) - min(game_rank)) AS rank_d
FROM shorttrack.athlete as a
JOIN (select id, time, game_type, game_rank from shorttrack.heat
Union
select id, time, game_type, game_rank from shorttrack.qfnl
Union
select id, time, game_type, game_rank from shorttrack.sfnl
Union
select id, time, game_type, game_rank from shorttrack.fnl) as t

ON a.id = t.id

Group by game_type, country_code
Order by game_type, time_difference DESC
'''
```

In [127]:
```
sql1_df = spark.read \
    .format("jdbc") \
    .option("url", postgres_uri) \
    .option("query", sql1) \
    .option("user", user) \
    .option("password", password) \
    .option("driver", "org.postgresql.Driver") \
    .load()

sql1_df.printSchema()
```

```
root
 |-- country_code: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- time_difference: double (nullable = true)
 |-- rank_difference: integer (nullable = true)
```

```
 |-- number_of_athletes: long (nullable = true)
```

In [128]:
```
sql1_df = sql1_df.toPandas()
```

In [129]:
```
sql1_df
```

Out[129]:

	country_code	game_type	time_difference	rank_difference	number_of_athletes
0	ITA	M1000M	130.039000	15	2
1	HUN	M1000M	115.248001	22	3
2	NED	M1000M	102.490000	14	3
3	JPN	M1000M	85.574000	9	2
4	CAN	M1000M	84.771000	7	2
...	...	...	...	...	...
91	HUN	W500M	1.057001	18	2
92	POL	W500M	0.398000	3	3
93	GBR	W500M	0.000000	0	1
94	CRO	W500M	0.000000	0	1
95	CZE	W500M	0.000000	0	1

96 rows × 5 columns

The second SQL query is used to detect how each athlete performed in each type of game they have attended by calculating their overall average ranking on each game level. Each athlete has their strengths, and weaknesses, this table would enable the coaches to know the athletes better and train them with more targeted methods. For example, if they are good at long-distance games, does it mean they lack explosive force? Or, if they are good at short distances do they lack endurance? More important, this data can also help the athletes to achieve better overall grades as a team. With the knowledge of each athlete's strong and weak game, the coaches can now make a smarter task allocation in future competitions.

In [130]:
```
sql2 = '''
SELECT name, game_type, ROUND(CAST(avg(game_rank) AS FLOAT)) as average_rank
FROM shorttrack.athlete as a
JOIN (select id, time, game_type, game_rank from shorttrack.heat
Union
select id, time, game_type, game_rank from shorttrack.qfnl
Union
select id, time, game_type, game_rank from shorttrack.sfnl
Union
select id, time, game_type, game_rank  from shorttrack.fnl) as t

ON a.id = t.id

Group by name, game_type

Order by name, game_type, average_rank
'''
```

In [131]:
```
sql2_df = spark.read \
    .format("jdbc") \
    .option("url", postgres_uri) \
    .option("query", sql2) \
    .option("user", user) \
    .option("password", password) \
    .option("driver", "org.postgresql.Driver") \
    .load()

sql2_df.printSchema()
```

```
root
 |-- name: string (nullable = true)
 |-- game_type: string (nullable = true)
 |-- average_rank: double (nullable = true)
```

In [132]:
```
sql2_df = sql2_df.toPandas()
```

In [133]:
```
sql2_df
```

Out[133]:

	name	game_type	average_rank
0	akar furkan	M1000M	11.0
1	ascic valentina	W1500M	15.0
2	ascic valentina	W500M	27.0

	name	game_type	average_rank
3	ayrapetyan denis	M1000M	31.0
4	ayrapetyan denis	M1500M	6.0
...	...	...	...
195	zhang chutong	W1000M	12.0
196	zhang chutong	W1500M	12.0
197	zhang tianyi	M1500M	34.0
198	zhang yuting	W1500M	12.0
199	zhang yuting	W500M	8.0

200 rows × 3 columns

In [134]:
```
sql3 = '''SELECT country_code,game_type, average(time)
FROM shorttrack.athlete as a
JOIN (select id, time, game_type from shorttrack.heat
Union
select id, time, game_type from shorttrack.qfnl
Union
select id, time, game_type from shorttrack.sfnl
Union
select id, time, game_type from shorttrack.fnl) as t

ON a.id = t.id

Group by game_type, country_code
Order by difference DESC
'''
```

6.0 Limitations and Further steps

Admittedly, due to time and space constraints, there are some limitations to our research. Firstly, all of our data were based on STK games from only the 2022 Beijing Winter Olympics. While it's true that this is the most recent and official statistic, in practice we need to consider the contingency of sporting events. Therefore, we believe that adding relevant historical data such as recent word STK championships may further improve the practical application effect of the model. Moreover, the Twitter post data we gathered are not large enough to carry out a more proper sentiment analysis due to the limitations we set when mining tweets. Besides, due to lack of text information, we left out the preprocessing part for the sentiment analysis to avoid a further loss of text. Given more text data, we could develop a tweet specified NLP data preprocessing method for better sentiment analysis results.

For future steps, to actually apply our data engineering system in practical utility, we suggest that more comprehensive data should be gathered and the schema should therefore be expanded accordingly. Finally， we believe this report can be generalised to more sports beyond STK especially racing ones such as athletics and swimming.

7.0 Conclusion

In conclusion, for our group coursework focusing on the winter olmpic short track speed skating, we have gathered related data from three sources. We have collected full game results of STK races from the International Olympic Committee (IOC) official website through web scraping and regular expression. In order to enrich the dataset, personal information of the Olympic STK athletes have been gathered from the STO website. Moreover, we have used twitter API for collecting additional recent tweets in the related topic so as to have a more comprehensive view on the STK races. Although we are having limited amount of tweets in our database, we will expand on it and NLP sentiment analysis will be carried out at the later stage. After the intial collection, transformation has been applied on the data, followed by storing the data in parquet format in the postgres databased according to the designed schema.

During the whole process, data version control has been implemented for better data lineage practice and source version control has been applied through Git for improving process transparency as there is collaboration on this group coursework. Terraform has also been set up with Github actions and the Terraform CLI has been configured to automate the workflow.

Reference

Coursera, (2021). What Is a Data Engineer?: A Guide to This In-Demand Career Available from: https://www.coursera.org/articles/what-does-a-data-engineer-do-and-how-do-i-become-one. (Accessed: 11 April 2022)

(2022). BEIJING 2022 SHORT TRACK SPEED SKATING RESULTS Avaiable from: https://olympics.com/en/olympic-games/beijing-2022/results/short-track-speed-skating (Accessed: 11 April 2022)

Appendix

We mainly carried out the project management by meetings. Below are the minutes of our group meetings including attendance, discussion, task allocation recordings.

In [135]:
```
Image("/project/DataEngineering/graphs/meeting_minute_1.png", width = 800)
```

Out[135]:

Date:	12 March 2022 8 pm - 9 pm	Attendance:	Huizi Hu, Zihui Liang, Haiyun Zou (On time) Yihan Sun (Late 20 minutes) Jingyi Chang (Late more than 20 minutes)

Meeting Summary						
Total Number of Participants	5					
Meeting Title	Data Engineering meeting					
Meeting Start Time	3/12/2022, 7:58:31 PM					
Meeting End Time	3/12/2022, 9:06:48 PM					
Meeting Id	d9696ab3-addd-41d7-9f59-54aad1f13ae4					
Full Name	Join Time	Leave Time	Duration	Email	Role	Participant ID (UPN)
Liang, Zoe	3/12/2022, 7:58:31 PM	3/12/2022, 9:06:48 PM	1h 8m	zihui.liang.21@ucl.ac.uk	Presenter	uceiz23@ucl.ac.uk
Hu, Huizi	3/12/2022, 7:59:17 PM	3/12/2022, 9:06:41 PM	1h 7m	huizi.hu.21@ucl.ac.uk	Presenter	uceih04@ucl.ac.uk
Zou, Haiyun	3/12/2022, 8:05:05 PM	3/12/2022, 9:06:45 PM	1h 1m	haiyun.zou.21@ucl.ac.uk	Organiser	uceihzo@ucl.ac.uk
Sun, Hermione	3/12/2022, 8:21:51 PM	3/12/2022, 9:06:42 PM	44m 51s	yihan.sun.21@ucl.ac.uk	Presenter	ucei006@ucl.ac.uk
Chang, Jingyi	3/12/2022, 8:28:42 PM	3/12/2022, 9:06:39 PM	37m 57s	jingyi.chang.21@ucl.ac.uk	Presenter	uceij47@ucl.ac.uk

Discussion:	Contribution: - Huizi Hu: large venue accidents, fashion shoes/clothes daily or weekly price - Zihui Liang: Ticketmaster/Viagogo or other tickets related API - Haiyun Zou: K-pop Twitter accounts information, house price (could include data from different agents), Movie information (e.g. rating, review, genre, etc.), Olympic game results - Yihan Sun: Contributed to the discussion of the movie idea and find the Olympic game results related websites and information. - Jingyi Chang: (She sporadically participated in the discussion) Agreement: - Focusing on the Short Track Speed Skating result in this Beijing 2022 Olympic Winter Game - Haiyun will try to gather all the data needed for the next meeting to build the schema

In [136]:
```
Image("/project/DataEngineering/graphs/meeting_minute_2.png", width = 800)
```

Out[136]:

Date:	29 March 2022 9 pm - 11 pm	Attendance:	Huizi Hu, Zihui Liang, Haiyun Zou, Yihan Sun (On time) Jingyi Chang (Absents)

Meeting Summary						
Total Number of Participants	4					
Meeting Title	Data Engineering meeting					
Meeting Start Time	3/29/2022, 8:55:43 PM					
Meeting End Time	3/29/2022, 11:00:41 PM					
Meeting Id	02f32480-678b-4fa6-977a-944df6ff6403					
Full Name	Join Time	Leave Time	Duration	Email	Role	Participant ID (UPN)
Zou, Haiyun	3/29/2022, 8:55:43 PM	3/29/2022, 11:00:38 PM	2h 4m	haiyun.zou.21@ucl.ac.uk	Organiser	uceihzo@ucl.ac.uk
Hu, Huizi	3/29/2022, 8:58:23 PM	3/29/2022, 11:00:37 PM	2h 2m	huizi.hu.21@ucl.ac.uk	Presenter	uceih04@ucl.ac.uk
Sun, Hermione	3/29/2022, 8:58:24 PM	3/29/2022, 11:00:41 PM	2h 2m	yihan.sun.21@ucl.ac.uk	Presenter	ucei006@ucl.ac.uk
Liang, Zoe	3/29/2022, 9:08:15 PM	3/29/2022, 10:51:57 PM	1h 43m	zihui.liang.21@ucl.ac.uk	Presenter	uceiz23@ucl.ac.uk
Liang, Zoe	3/29/2022, 10:54:31 PM	3/29/2022, 11:00:36 PM	6m 5s	zihui.liang.21@ucl.ac.uk	Presenter	uceiz23@ucl.ac.uk

Discussion:	Contribution: - Huizi Hu: helping to fix the errors - Zihui Liang: Web scraping Twitter posts for “Short track skating” - Haiyun Zou: demonstrate the current progress about the data-gathering stage - Yihan Sun: helping to fix the errors - Jingyi Chang: (Absents) Agreement: - Huizi and Yihan will build the schema - Yihan will do some sentiment analysis score for the tweets - Haiyun will gather more related information if needed and support other team members - Zihui will set up the terraform

In [137]:
```
Image("/project/DataEngineering/graphs/meeting_minute_3.png", width = 800)
```

Out[137]:

Date:	4 April 2022 9 pm - 10 pm	Attendance:	Huizi Hu, Zihui Liang, Haiyun Zou, Yihan Sun (On time) Jingyi Chang (Absents)

Meeting Summary						
Total Number of Participants	4					
Meeting Title	Data Engineering meeting					
Meeting Start Time	4/4/2022, 8:59:06 PM					
Meeting End Time	4/4/2022, 10:29:30 PM					
Meeting Id	3bee5b2e-c111-48d9-a14b-845faee06622					
Full Name	Join Time	Leave Time	Duration	Email	Role	Participant ID (UPN)
Zou, Haiyun	4/4/2022, 8:59:06 PM	4/4/2022, 10:29:29 PM	1h 30m	haiyun.zou.21@ucl.ac.uk	Organiser	uceihzo@ucl.ac.uk
Liang, Zoe	4/4/2022, 8:59:14 PM	4/4/2022, 10:03:29 PM	1h 4m	zihui.liang.21@ucl.ac.uk	Presenter	uceiz23@ucl.ac.uk
Liang, Zoe	4/4/2022, 10:05:43 PM	4/4/2022, 10:05:46 PM	3s	zihui.liang.21@ucl.ac.uk	Presenter	uceiz23@ucl.ac.uk
Sun, Hermione	4/4/2022, 8:59:25 PM	4/4/2022, 10:29:30 PM	1h 30m	yihan.sun.21@ucl.ac.uk	Presenter	ucei006@ucl.ac.uk
Hu, Huizi	4/4/2022, 9:05:08 PM	4/4/2022, 10:29:30 PM	1h 24m	huizi.hu.21@ucl.ac.uk	Presenter	uceih04@ucl.ac.uk

Discussion:	Main discussion: - Haiyun went through the complete process with other team members and distributed the report write up parts to other team members.