kata

March 3, 2023

1 Imported datasets

airport.csv https://ourairports.com/data/ airlines.csv https://raw.githubusercontent.com/jpatokal/openflights/ma
aircrafts.csv https://raw.githubusercontent.com/jpatokal/openflights/master/data/planes.dat

https://openflights.org/data.html#airline https://applications.icao.int/dataservices/default.aspx

[ ]:

```
!pip install FlightRadarAPI
!wget https://davidmegginson.github.io/ourairports-data/airports.csv
!wget https://raw.githubusercontent.com/jpatokal/openflights/master/data/
 ↪airlines.dat
!wget https://raw.githubusercontent.com/jpatokal/openflights/master/data/planes.
 ↪dat
```

```
Collecting FlightRadarAPI
  Downloading FlightRadarAPI-1.2.3.tar.gz (7.5 kB)
  Preparing metadata (setup.py) … done
Collecting Brotli
  Downloading Brotli-1.0.9-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (2.7 MB)
                           2.7/2.7 MB
3.6 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-
packages (from FlightRadarAPI) (2.28.2)
Collecting Deprecated
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting wrapt<2,>=1.10
  Downloading wrapt-1.15.0-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (78 kB)
                           78.4/78.4 kB
6.2 MB/s eta 0:00:00
Requirement already satisfied: idna<4,>=2.5 in
/opt/conda/lib/python3.10/site-packages (from requests->FlightRadarAPI) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from requests->FlightRadarAPI) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from requests->FlightRadarAPI)
```

```
(1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from requests->FlightRadarAPI)
(2022.12.7)
Building wheels for collected packages: FlightRadarAPI
  Building wheel for FlightRadarAPI (setup.py) … done
  Created wheel for FlightRadarAPI:
filename=FlightRadarAPI-1.2.3-py3-none-any.whl size=8741
sha256=dacc668c15c6e1474190f001f4e9ba960be3fa1a81504cb01e4c654a11d8dc17
  Stored in directory: /home/jovyan/.cache/pip/wheels/96/79/4a/2cb77e60b81d8cf00
355fd12ff2654a24e49e5d5a68f24517b
Successfully built FlightRadarAPI
Installing collected packages: Brotli, wrapt, Deprecated, FlightRadarAPI
Successfully installed Brotli-1.0.9 Deprecated-1.2.13 FlightRadarAPI-1.2.3
wrapt-1.15.0
--2023-03-03 17:30:42--  https://davidmegginson.github.io/ourairports-
data/airports.csv
Resolving davidmegginson.github.io (davidmegginson.github.io)…
185.199.110.153, 185.199.111.153, 185.199.109.153, …
Connecting to davidmegginson.github.io
(davidmegginson.github.io)|185.199.110.153|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 10736053 (10M) [text/csv]
Saving to: ‘airports.csv’

airports.csv        100%[===================>]  10.24M  2.85MB/s    in 3.5s

2023-03-03 17:30:46 (2.88 MB/s) - ‘airports.csv’ saved [10736053/10736053]

--2023-03-03 17:30:48--
https://raw.githubusercontent.com/jpatokal/openflights/master/data/airlines.dat
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.108.133, 185.199.111.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 396896 (388K) [text/plain]
Saving to: ‘airlines.dat’

airlines.dat        100%[===================>] 387.59K  2.16MB/s    in 0.2s

2023-03-03 17:30:48 (2.16 MB/s) - ‘airlines.dat’ saved [396896/396896]

--2023-03-03 17:30:50--
https://raw.githubusercontent.com/jpatokal/openflights/master/data/planes.dat
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.109.133, 185.199.108.133, 185.199.111.133, …
Connecting to raw.githubusercontent.com
```

```
(raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 8331 (8.1K) [text/plain]
Saving to: ‘planes.dat’

planes.dat          100%[===================>]   8.14K  --.-KB/s    in 0s

2023-03-03 17:30:50 (21.1 MB/s) - ‘planes.dat’ saved [8331/8331]
```

[ ]:

```python
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.sql.functions import broadcast, udf, expr
from pyspark.sql.types import FloatType, StructType, StructField, StringType
from pyspark.sql.dataframe import DataFrame
from FlightRadar24.api import FlightRadar24API


sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
```

[ ]:

```python
def get_and_write_flights() -> DataFrame:
    """Get flights from FlightRadar24 and write them to the file"""

    def get_file_name() -> str:
        """Generate file name for the current date and time"""

        from datetime import datetime
        now = datetime.now()
        year = now.year
        month = now.month
        day = now.day
        hour = now.hour
        minute = now.minute
        second = now.second
        milisecond = now.microsecond // 10_000
        return f"Flights/rawzone/tech_year={year}/tech_month={year}-{month}/
 ↪tech_day={year}-{month}-{day}/
 ↪flights{year}{month}{day}{hour}{minute}{second}{milisecond}.csv"


    fr_api = FlightRadar24API()
    flights = fr_api.get_flights()

    df = spark.createDataFrame(flights)
    df.coalesce(1).write.csv(get_file_name(), mode='overwrite', header=True,␣
 ↪sep=';')
```

```
    return df
```

[ ]:

```
def clean_dataframe(df: DataFrame) -> DataFrame:
    """Clean dataframe"""

    df = df.filter(~df.destination_airport_iata.isin(["NaN", "N/A"]))
    df = df.filter(~df.origin_airport_iata.isin(["NaN", "N/A"]))

    return df
```

[ ]:

```
def add_distance_dataframe(df: DataFrame) -> DataFrame:
    """Add details to dataframe"""
    from math import sin, cos, sqrt, atan2, radians
    def distance(lat1: float, lon1: float, lat2: float, lon2: float) -> float:
        """Calculate distance between two points"""
        if lat1 is None or lon1 is None or lat2 is None or lon2 is None:
            return -1
        # approximate radius of earth in km
        R = 6373.0

        lat1 = radians(lat1)
        lon1 = radians(lon1)
        lat2 = radians(lat2)
        lon2 = radians(lon2)

        dlon = lon2 - lon1
        dlat = lat2 - lat1

        a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
        c = 2 * atan2(sqrt(a), sqrt(1 - a))

        distance = R * c
        return distance

    distance_udf = udf(distance, FloatType())

    df_airport = spark.read.csv("airports.csv", header=True, sep=',')
    df_airport = df_airport.drop("id", "ident", "type", "name", "elevation_ft",␣
 ↪"iso_region", "municipality", "scheduled_service", "gps_code", "local_code",␣
 ↪"home_link", "wikipedia_link", "keywords", "iso_country")

    df_airport = df_airport.filter(df_airport.iata_code.isNotNull() &␣
 ↪df_airport.continent.isNotNull())

    df_airport = df_airport.withColumn("latitude_deg",␣
 ↪df_airport["latitude_deg"].cast("float"))
```

```
    df_airport = df_airport.withColumn("longitude_deg",␣
↪df_airport["longitude_deg"].cast("float"))

    df_airport_destination = df_airport.withColumnRenamed("iata_code",␣
↪"destination_airport_iata")\
                                       .withColumnRenamed("latitude_deg",␣
↪"destination_latitude_deg")\
                                       .withColumnRenamed("longitude_deg",␣
↪"destination_longitude_deg")\
                                       .withColumnRenamed("continent",␣
↪"destination_airport_continent")

    df_airport_origin = df_airport.withColumnRenamed("iata_code",␣
↪"origin_airport_iata")\
                                  .withColumnRenamed("latitude_deg",␣
↪"origin_latitude_deg")\
                                  .withColumnRenamed("longitude_deg",␣
↪"origin_longitude_deg")\
                                  .withColumnRenamed("continent",␣
↪"origin_airport_continent")

    df = df.join(broadcast(df_airport_destination),␣
↪["destination_airport_iata"], how='left')
    df = df.join(broadcast(df_airport_origin), ["origin_airport_iata"],␣
↪how='left')

    df = df.withColumn("distance", distance_udf(df.origin_latitude_deg, df.
↪origin_longitude_deg, df.destination_latitude_deg, df.
↪destination_longitude_deg))

    return df


def add_aircrafts_dataframe(df: DataFrame) -> DataFrame:
    """Add aircrafts to dataframe"""
    df_aircrafts = spark.read.csv("planes.dat", header=False, sep=',')
    df_aircrafts = df_aircrafts.drop("_c1")
    df_aircrafts = df_aircrafts.withColumnRenamed("_c0", "aircraft_name")\
                               .withColumnRenamed("_c2", "aircraft_code")
    df_aircrafts = df_aircrafts.filter(df_aircrafts.aircraft_code.isNotNull())

    df = df.join(broadcast(df_aircrafts), df.aircraft_code == df_aircrafts.
↪aircraft_code, how='left')
    return df

def add_airlines_dataframe(df: DataFrame) -> DataFrame:
    """Add airlines to dataframe"""
    df_airlines = spark.read.csv("airlines.dat", header=False, sep=',')
    df_airlines = df_airlines.select("_c1","_c3", "_c4")
    df_airlines = df_airlines.withColumnRenamed("_c1", "airline_name")\
                             .withColumnRenamed("_c3", "airline_iata")\
                             .withColumnRenamed("_c4", "airline_icao")

    df_airlines = df_airlines.filter(df_airlines.airline_iata.isNotNull() |␣
 ↪df_airlines.airline_icao.isNotNull())

    df = df.join(broadcast(df_airlines), [(df.airline_icao == df_airlines.
 ↪airline_icao) | (df.airline_iata == df_airlines.airline_iata)], how='left')
    df = df.drop("airline_icao").drop("airline_iata")

    return df
```

```
[ ]: def get_active_flights(df: DataFrame) -> DataFrame:
         """Get active flights"""

         df = df.filter(df.on_ground == 0)
         return df
```

```
[ ]: df = get_and_write_flights()

     df = clean_dataframe(df)

     df = add_distance_dataframe(df)
     df = add_aircrafts_dataframe(df)
     df = add_airlines_dataframe(df)

     df_active = get_active_flights(df)
```

```
[ ]: schema = StructType([
         StructField("continent_name", StringType(), True),
         StructField("continent", StringType(), True)
     ])

     data = [("North America", "NA"),
             ("South America", "SA"),
             ("Europe", "EU"),
             ("Asia", "AS"),
             ("Africa", "AF"),
             ("Australia", "OC")]
     df_continent = broadcast(spark.createDataFrame(data, schema))
```

[ ]:

```
# Q1

df_active.createOrReplaceTempView("df_active")

df_q1 = spark.sql("""SELECT airline_name, COUNT(airline_name) AS nb_flights
                                 FROM df_active
                                 GROUP BY airline_name
                                 ORDER BY nb_flights DESC
                                 LIMIT 1""")
```

[ ]:

```
# Q2

df_q2 = df_active.groupBy("origin_airport_continent",␣
 ↪"destination_airport_continent", "airline_name")\
          .agg({"airline_name": "count"})\
          .orderBy("count(airline_name)", ascending=False)\
          .filter(df_active.origin_airport_continent == df_active.
 ↪destination_airport_continent)

df_q2 = df_q2.withColumnRenamed("count(airline_name)", "number_of_flights")\
             .drop("origin_airport_continent")\
             .withColumnRenamed("destination_airport_continent", "continent")


df_q2.createOrReplaceTempView("df_q2")


df_q2 = spark.sql("""
    SELECT continent, airline_name, number_of_flights
    FROM (
        SELECT continent, airline_name, number_of_flights,
        ROW_NUMBER() OVER (PARTITION BY continent ORDER BY number_of_flights␣
 ↪DESC) AS row_number
        FROM df_q2
    )
    WHERE row_number = 1
""")

df_q2 = df_q2.join(df_continent, df_q2.continent == df_continent.continent,␣
 ↪how='left')
df_q2 = df_q2.drop("continent")
```

[ ]:

```
# Q3

df_q3 = spark.sql("""
    SELECT * FROM df_active
    WHERE distance = (SELECT MAX(distance) FROM df_active)
""")
```

[ ]:

```
# Q4

df.createOrReplaceTempView("df")

df_q4 = spark.sql("""
    SELECT origin_airport_continent, AVG(distance) AS distance_mean
    FROM df
    WHERE distance > 0
    GROUP BY origin_airport_continent
""")

df_q4 = df_q4.join(df_continent, df_q4.origin_airport_continent == df_continent.
 ↪continent, how='left')
df_q4 = df_q4.drop("origin_airport_continent", "continent")
```

[ ]:

```
# Q5

df_q5 = df_active.groupBy("aircraft_name")\
            .agg({"aircraft_name": "count"})\
            .orderBy("count(aircraft_name)", ascending=False)\
            .limit(1)

df_q5 = df_q5.withColumnRenamed("count(aircraft_name)", "number_of_flights")\
                .withColumnRenamed("aircraft_name", "aircraft")
```

[ ]:

```
# Q6

df_q6 = df.groupBy("airline_name", "aircraft_name")\
          .agg({"airline_name": "count"})\
          .orderBy("count(airline_name)", ascending=False)\
          .filter(df.airline_name.isNotNull())\
          .filter(df.aircraft_name.isNotNull())

df_q6 = df_q6.withColumnRenamed("count(airline_name)", "number_of_flights")

df_q6.createOrReplaceTempView("df_q6")

df_q6 = spark.sql("""
    SELECT airline_name, aircraft_name, number_of_flights
    FROM (
        SELECT airline_name, aircraft_name, number_of_flights,
        ROW_NUMBER() OVER (PARTITION BY airline_name ORDER BY number_of_flights␣
 ↪DESC) AS row_number
        FROM df_q6
    )
    WHERE row_number <= 3
""")
```

[ ]:

```
# Question Bonus

df_qb_1 = spark.sql("""
    SELECT origin_airport_iata, COUNT(id) AS nb_departures
    FROM df
    GROUP BY origin_airport_iata
""")

df_qb_2 = spark.sql("""
    SELECT destination_airport_iata, COUNT(id) AS nb_arrivals
    FROM df
    GROUP BY destination_airport_iata
""")

# Clean des valeurs nuls
df_qb_1 = df_qb_1.filter(df_qb_1.origin_airport_iata.isNotNull())
df_qb_2 = df_qb_2.filter(df_qb_2.destination_airport_iata.isNotNull())

df_qb = df_qb_1.join(df_qb_2, df_qb_1.origin_airport_iata == df_qb_2.
 ↪destination_airport_iata)


df_qb = df_qb.withColumn("difference", expr("abs(nb_departures - nb_arrivals)"))
df_qb = df_qb.drop("origin_airport_iata", "nb_departures", "nb_arrivals")\
             .withColumnRenamed("destination_airport_iata", "airport_iata")

df_qb.createOrReplaceTempView("df_qb")


df_qb = spark.sql("""
    SELECT * FROM df_qb
    WHERE difference = (SELECT MAX(difference) FROM df_qb)
""")
```

[ ]:

```
# Clean des vues

spark.catalog.dropTempView("df")
spark.catalog.dropTempView("df_active")
spark.catalog.dropTempView("df_q2")
spark.catalog.dropTempView("df_q6")
spark.catalog.dropTempView("df_qb")
```

[ ]:

```
True
```

[ ]:
```
# Affichage Q1
df_q1.show()
```

```
+---------------+----------+
|   airline_name|nb_flights|
+---------------+----------+
|United Airlines|        70|
+---------------+----------+
```

[ ]:
```
# Affichage Q2
df_q2.show()
```

```
+--------------------+-----------------+--------------+
|        airline_name|number_of_flights|continent_name|
+--------------------+-----------------+--------------+
|  Ethiopian Airlines|               10|        Africa|
|       Qatar Airways|               15|          Asia|
|Aeroflot Russian …|               3|        Europe|
|   American Airlines|               31| North America|
|              Qantas|                2|     Australia|
|Avianca - Aerovia…|                2| South America|
+--------------------+-----------------+--------------+
```

[ ]:
```
# Affichage Q3
df_q3.show()
```

```
+-------------------+------------------------+-------------+--------+--------+--
----------+-------+----------+--------+--------+---------+------+---------+-----
-------+------+----------+--------------+------------------------+--------------
-----------+-----------------------------+-------------------+------------------
--+------------------------+---------+---------------+-------------+------------
--------+
|origin_airport_iata|destination_airport_iata|aircraft_code|altitude|callsign|gr
ound_speed|heading|icao_24bit|
id|latitude|longitude|number|on_ground|registration|squawk|      time|vertical_s
peed|destination_latitude_deg|destination_longitude_deg|destination_airport_cont
inent|origin_latitude_deg|origin_longitude_deg|origin_airport_continent|
distance|  aircraft_name|aircraft_code|        airline_name|
+-------------------+------------------------+-------------+--------+--------+--
----------+-------+----------+--------+--------+---------+------+---------+-----
-------+------+----------+--------------+------------------------+--------------
-----------+-----------------------------+-------------------+------------------
--+------------------------+---------+---------------+-------------+------------
--------+
|                JFK|                     SIN|         A359|   41000|   SIA23|
538|    106|    76CCE1|2f634a69| 23.5177|  71.0424|  SQ23|        0|
9V-SGA|   N/A|1677864946|             0|                 1.35019|
103.994|                           AS|          40.639446|           -73.77932|
NA|15345.416|Airbus A350-900|         A359|  Singapore Airlines|
|                JFK|                     SIN|         A359|   41000|   SIA23|
538|    106|    76CCE1|2f634a69| 23.5177|  71.0424|  SQ23|        0|
9V-SGA|   N/A|1677864946|             0|                 1.35019|
103.994|                           AS|          40.639446|           -73.77932|
NA|15345.416|Airbus A350-900|         A359|Singapore Airline…|
|                SIN|                     JFK|         A359|   41000|   SIA24|
486|    118|    76CCE5|2f635f64| 61.6399|-118.5396|  SQ24|        0|
9V-SGE|   N/A|1677864947|             0|               40.639446|
-73.77932|                           NA|            1.35019|
103.994|                      AS|15345.416|Airbus A350-900|         A359|
Singapore Airlines|
|                SIN|                     JFK|         A359|   41000|   SIA24|
486|    118|    76CCE5|2f635f64| 61.6399|-118.5396|  SQ24|        0|
9V-SGE|   N/A|1677864947|             0|               40.639446|
-73.77932|                           NA|            1.35019|
103.994|                      AS|15345.416|Airbus A350-900|
A359|Singapore Airline…|
+--------------------+------------------------+-------------+--------+--------+--
----------+-------+----------+--------+--------+---------+------+---------+-----
-------+------+----------+--------------+------------------------+---------------
-----------+-----------------------------+-------------------+------------------
--+------------------------+---------+---------------+-------------+-------------
--------+
```

```
[ ]: # Affichage Q4
     df_q4.show()
```

```
+-----------------+--------------+
|    distance_mean|continent_name|
+-----------------+--------------+
| 5287.37030444502| North America|
|6145.360305373733| South America|
|6806.829909319196|          Asia|
|8872.555788352272|     Australia|
|7228.824730491639|        Europe|
|5012.622829127956|        Africa|
+-----------------+--------------+
```

```
[ ]: # Affichage Q5
     df_q5.show()
```

```
+----------------+-----------------+
|        aircraft|number_of_flights|
+----------------+-----------------+
|Boeing 777-300ER|              177|
+----------------+-----------------+
```

```
[ ]: # Affichage Q6
     df_q6.show()
```

```
+--------------------+------------------+-----------------+
|        airline_name|     aircraft_name|number_of_flights|
+--------------------+------------------+-----------------+
|  3 Valleys Airlines|    Boeing 747-400|                2|
|ABSA - Aerolinhas…|    Boeing 767-300|                1|
|             ABX Air|    Boeing 767-300|                1|
|AJT Air Internati…|   Airbus A330-300|                1|
|                ALAK|    Boeing 737-800|                1|
|          Aer Lingus|   Airbus A330-300|                6|
|          Aer Lingus|    Airbus A321neo|                5|
|Aero Asia Interna…|    Boeing 737-800|                1|
|          AeroMéxico|    Boeing 737-800|                2|
|          AeroMéxico|      Boeing 787-9|                1|
|          AeroMéxico|  Boeing 737 MAX 8|                1|
|Aeroflot Russian …|   Airbus A330-300|                3|
|Aeroflot Russian …|    Boeing 737-800|                2|
|Aeroflot Russian …|       Airbus A319|                1|
|    Aeroland Airways|  Boeing 777-200LR|                8|
|Aerolineas Argent…|   Airbus A330-200|                2|
|African Business …|   Airbus A330-200|                2|
|         Air Algerie|   Airbus A330-200|                2|
|         Air Algerie|Dassault Falcon 7X|                1|
|Air Antilles Express|  Boeing 777-200LR|                8|
+--------------------+------------------+-----------------+
only showing top 20 rows
```

```
[ ]: # Affichage Q Bonus
     df_qb.show()
```

```
+------------+----------+
|airport_iata|difference|
+------------+----------+
|         FRA|        83|
+------------+----------+
```