s299165_lab5

January 25, 2024

Laboratory 5 - Graph analytics with SparkGraphFrames

1 Input Data

[1]:
```
# Input data files Addresses
airportsPath = "/data/students/bigdata_internet/lab5/airports.csv"
airlinesPath = "/data/students/bigdata_internet/lab5/airlines.csv"
routesPath = "/data/students/bigdata_internet/lab5/routes.csv"
```

[2]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

[3]:
```
# Read Files
dfAirport_data = spark.read.load(airportsPath, format="csv" ,sep = ",", header␣
↪= True, inferSchema=True)
dfAirline_data = spark.read.load(airlinesPath, format="csv" ,sep = ",", header␣
↪= True, inferSchema=True)
dfRoute_data = spark.read.load(routesPath, format="csv" ,sep = ",", header =␣
↪True, inferSchema=True)
```

2 Top airports and airlines

- Question 2.1) Which are the countries in the world with more than 200 airports? Reports these countries and their number of airports (in descending order).
- Question 2.2) Which are the Top-5 airlines by total number of flights? For each airline in theTop-10, provide airline name, airline icao code and number of flights (in ascending order).
- Question 2.3) Which are the Top-5 routes in the world? Routes are identified by airport source and airport destination,independently on the airline. For each route in the Top-5, provide the source airport name and city, the destination airport name and city, and thenumber of routes (in descending order).

Answer

[4]:
```
AP_Countries = dfAirport_data.groupBy("country").count().
↪sort("count",ascending=False)
```

```
AP_Countries_filter = AP_Countries.filter(AP_Countries["count"] > 200)
```

[5]:
```
# Answer Question 2.1:
AP_Countries_filter.withColumnRenamed("count", "Num_Airports").show()
```

```
[Stage 7:=======================================================>(199 + 1) / 200]

+-------------+------------+
|      country|Num_Airports|
+-------------+------------+
|United States|        1512|
|       Canada|         430|
|    Australia|         334|
|       Russia|         264|
|       Brazil|         264|
|      Germany|         249|
|        China|         241|
|       France|         217|
+-------------+------------+
```

[6]:
```
from pyspark.sql.functions import col, concat
# Group by airline and count the number of flights
dfTopAirlines = dfRoute_data.groupBy("airline_id").count().sort("count",␣
 ↪ascending= False)\
.withColumnRenamed("count", "Num_Flights").limit(10)
# Join with the airline data to get the airline name and ICAO code
dfTopAirlinesInfoAll = dfTopAirlines.join(dfAirline_data, on = "airline_id",␣
 ↪how = "inner")
# Select relevant columns
dfTopAirlinesInfo = dfTopAirlinesInfoAll.select("name", "icao", "num_flights")
```

[7]:
```
# Answer Question 2.2:
dfTopAirlinesInfo.orderBy("Num_Flights").show(truncate=False)
```

```
+----------------------+----+-----------+
|name                  |icao|num_flights|
+----------------------+----+-----------+
|easyJet               |EZY |1130       |
|Southwest Airlines    |SWA |1146       |
|Air China             |CCA |1260       |
|China Eastern Airlines|CES |1263       |
|China Southern Airlines|CSN |1454       |
|US Airways            |USA |1960       |
|Delta Air Lines       |DAL |1981       |
```

```
|United Airlines         |UAL |2180       |
|American Airlines       |AAL |2354       |
|Ryanair                 |RYR |2484       |
+------------------------+----+-----------+
```

[8]:
```
# Number of routes
dfTopRouteInfo_count = dfRoute_data.
 ↪groupBy("airport_source_id","airport_destination_id")\
.count().sort("count", ascending= False).withColumnRenamed("count", "Num_Rout")
```

[9]:
```
# Find Departure Airports and cities
TopRoute_Source = dfTopRouteInfo_count.join(dfAirport_data,␣
 ↪dfTopRouteInfo_count["airport_source_id"] \
                == dfAirport_data["id"]).
 ↪select("airport_source_id","name","city","airport_destination_id","Num_Rout")\
                .withColumnRenamed("name","Departure_Airport").
 ↪withColumnRenamed("city","Departure_city")
```

[10]:
```
# Find Arrival Airports and cities
TopRoute_Dest = TopRoute_Source.join(dfAirport_data,␣
 ↪TopRoute_Source["airport_destination_id"]== dfAirport_data["id"])\
    .withColumnRenamed("name", "Arrival_Airport")\
    .withColumnRenamed("city", "Arrival_city")\
    .select("airport_source_id","Departure_Airport",␣
 ↪"Departure_city","airport_destination_id", "Arrival_Airport",␣
 ↪"Arrival_city", "Num_Rout")
```

[11]:
```
# Answer Question 2.3:
TopRoute_Dest.show(5)
```

```
[Stage 16:=================================================>   (187 + 1) / 200]

+----------------+--------------------+-------------+----------------------+--
-----------------+------------+--------+
|airport_source_id|   Departure_Airport|Departure_city|airport_destination_id|
Arrival_Airport|Arrival_city|Num_Rout|
+----------------+--------------------+-------------+----------------------+--
-----------------+------------+--------+
|             3830|Chicago O'Hare In…|       Chicago|
3682|Hartsfield Jackso…|     Atlanta|      20|
|             3682|Hartsfield Jackso…|       Atlanta|
3830|Chicago O'Hare In…|     Chicago|      19|
|             3830|Chicago O'Hare In…|       Chicago|
3861|Louis Armstrong N…| New Orleans|      13|
|             3179|Phuket Internatio…|        Phuket|
3885|Suvarnabhumi Airport|     Bangkok|      13|
|             2179|Abu Dhabi Interna…|     Abu Dhabi|
```

```
2194|Muscat Internatio…|      Muscat|      12|
+-----------------+--------------------+-------------+----------------------+--
-----------------+------------+--------+
only showing top 5 rows
```

3 Create the graph of flight connections

- Question 3.1) How many rows did you filter out?

Answer

[12]:
```
# Apply filter
dfRoute_data_filtered = dfRoute_data.filter((dfRoute_data["airport_source_id"] !
 ↪= "\\N")\
                                            &␣
 ↪(dfRoute_data["airport_destination_id"] != "\\N"))
```

[13]:
```
# Answer Question 3.1
print("Number of routes after filtering is {}, and {} rows are filtered out."\
      .format(dfRoute_data_filtered.count(),(dfRoute_data.
 ↪count()-dfRoute_data_filtered.count())))
```

```
Number of routes after filtering is 67240, and 423 rows are filtered out.
```

[14]:
```
# Features for Vertices and Edges
vertices = dfAirport_data.select("id", "name", "city", "country", "icao")\
                         .withColumnRenamed("id", "id").
 ↪withColumnRenamed("name", "label")
edges = dfRoute_data_filtered.select("airport_source_id",␣
 ↪"airport_destination_id", "airline_id")\
                .withColumnRenamed("airport_source_id", "src")\
                .withColumnRenamed("airport_destination_id", "dst")\
                .withColumnRenamed("airline_id", "label")
```

[15]:
```
#converting columns to string
string_v = vertices.withColumn("id", vertices.id.cast("string"))
string_e = edges.withColumn("src", edges.src.cast("string"))\
                .withColumn("dst", edges.dst.cast("string"))\
                .withColumn("label", edges.label.cast("string"))
```

[16]:
```
# Create the GraphFrame
from graphframes import GraphFrame
g = GraphFrame(string_v, edges)
```

4 Analyze and process the graph

- Question 4.1) Which are the airports with the highest ratio of outgoing edges overincoming ones)? Report the TOP-5 airports' names, ids, edgeIN, edgeOUT,and ratios (in descending order).
- Question 4.2.1) From how many airports can you reach the city of "Torino" taking exactly1 flight?
- Question 4.2.2) How many airports can you reach from the city of "Torino" taking exactly 1flight?
- Question 4.2.3) Any comments in the numbers? Can you justify the result (consider the node's degrees…)?
- Question 4.3) How many airports can you reach from the city of "Torino" taking exactly 2flight?
- Question 4.4.1) From how many airports in the world you can reach "Los Angeles International Airport" using less hops than to reach the city of Torino?
- Question 4.4.2) From how many airports in the world you can reach the city of Torino using less hops than to reach "Los Angeles International Airport"?
- Question 4.4.3) From how many airports in the world you can reach with the same numberof hops Torino and "Los Angeles International Airport"?
- Question 4.5) How many connected components of at least two airports are there in the graph? Report the number of connected components and their sizes.
- Question 4.6) Consider only the subgraph of the flights that are performed by two different airlines (identified by the ICAO), each involving at least 5 cities. Choose two airlines and report the names and ICAO of these airlines.
- Question 4.7) Plot the subgraph of these flights. Report the name of the cities (of theairports) in the graph.

Answer

[17]:
```
edgeIn = g.inDegrees   #Find edgeIN (inDegree)
edgeOUT = g.outDegrees #Find edgeOUT (outDegree)
```

[18]:
```
# Join edgeIn and edgeOUT
airports_degree = vertices.join(edgeIn, on='id').join(edgeOUT, on='id')
```

[19]:
```
# Calculate edgeOUT/edgeIN ratio
airport_OI_ratio = airports_degree.withColumn("ratio_OUT/IN", col("outDegree") /
 ↪ col("inDegree"))
```

[20]:
```
top5_airports = airport_OI_ratio.orderBy("ratio_OUT/IN", ascending=False).
 ↪limit(5)
```

[21]:
```
# Answer Question 4.1
top5_airports.show()
```

```
+----+--------------------+----------+---------------+----+--------+---------+
------------+
```

```
|  id|               label|       city|
country|icao|inDegree|outDegree|ratio_OUT/IN|
+----+--------------------+-----------+---------------+----+--------+---------+
------------+
|1033|       Bunia Airport|      Bunia|Congo (Kinshasa)|FZKA|       1|        3|
3.0|
|1662|Transilvania Târg…|Tirgu Mures|         Romania|LRTM|       2|        5|
2.5|
|5521|  Pikangikum Airport| Pikangikum|          Canada|CYPM|       2|        5|
2.5|
| 211|Djanet Inedbirene…|     Djanet|         Algeria|DAAJ|       1|        2|
2.0|
| 428|       Ivalo Airport|      Ivalo|         Finland|EFIV|       1|        2|
2.0|
+----+--------------------+-----------+---------------+----+--------+---------+
------------+
```

[22]:
```
# Answer Question 4.2.2, Using motif
Torino_inMotif = g.find("(a)-[e]->(b)").filter("b.city = 'Torino'").filter("b.
  ↪id = 1526")
print("Direct flights to Torino:", Torino_inMotif.count())
```

```
[Stage 32:>                                                         (0 + 1) / 1]
```

```
Direct flights to Torino: 42
```

[23]:
```
# Answer Question 4.2.3, Using motif
Torino_outMotif = g.find("(a)-[e]->(b)").filter("a.city = 'Torino'").filter("a.
  ↪id = 1526")
print("Direct flights from Torino:", Torino_outMotif.count())
```

```
Direct flights from Torino: 44
```

Answer Question 4.2.3: Torino has more departing flights, and flights might not back to this city; as mentioned above, the INdegree and OUTdegree of Torino can show the number of cities that can directly connect with exactly one flight.

[24]:
```
# Extract data for Turin & LA Airports for next steps
airport_OI_ratio.filter((airport_OI_ratio["city"] == "Torino") |\
                        (airport_OI_ratio["city"] == "Los Angeles")).show()
```

```
+----+--------------------+-----------+-------------+----+--------+---------+---
---------------+
|  id|               label|       city|      country|icao|inDegree|outDegree|
ratio_OUT/IN|
```

```
+----+--------------------+----------+------------+----+--------+---------+---
--------------+
|1526|       Turin Airport|     Torino|        Italy|LIMF|      42|
44|1.0476190476190477|
|3484|Los Angeles Inter…|Los Angeles|United States|KLAX|     498|
492|0.9879518072289156|
+----+--------------------+----------+------------+----+--------+---------+---
--------------+
```

[25]:
```
# Answer Question 4.3, Using motif
Torino_outMotif = g.find("(a)-[e1]->(b); (b)-[e2]->(c)")\
                        .filter("a.city = 'Torino'").filter("a.id = 1526")
```

[26]:
```
# Select distinct destination airports
distinct_airports = Torino_outMotif.selectExpr("b.id as airport_id")\
.union(Torino_outMotif.selectExpr("c.id as airport_id")).distinct()
# Answer Question 4.3,
num_distinct_airports = distinct_airports.count()
print(f"Number of cities reachable from Torino with 2 flights:␣
 ↪{num_distinct_airports}")
```

```
[Stage 48:=====================================================>(198 + 2) / 200]
```

```
Number of cities reachable from Torino with 2 flights: 590
```

[27]:
```
# Torino Airport ID = 1526 - LA Airport ID = 3484
Destinations = ['1526' ,'3484']
# Find the paths to LA & TO
pathsto_dest = g.shortestPaths(landmarks=Destinations)
```

[28]:
```
from pyspark.sql.functions import expr
# Answer Question 4.4.1
num_airports_lessLA_hops = pathsto_dest.filter(expr("distances['1526'] >\
                                                    distances['3484']")).count()
print("You can reach to LA with less hops than TO from {} airports"\
                                            .format(num_airports_lessLA_hops))
```

```
[Stage 279:=========================================>          (159 + 3) / 200]
```

```
You can reach to LA with less hops than TO from 1831 airports
```

[29]:
```
# Answer Question 4.4.2
num_airports_lessTO_hops = pathsto_dest.filter(expr("distances['1526'] <\
```

```
                                                    distances['3484']")).count()
print("You can reach to TO with less hops than LA from {} airports"\
                                                .format(num_airports_lessTO_hops))
```

```
[Stage 313:================================================>    (183 + 1) / 200]
```

```
You can reach to TO with less hops than LA from 94 airports
```

[30]:
```
# Answer Question 4.4.3
num_airports_eq_hops = pathsto_dest.filter(expr("distances['1526'] =\
                                                    distances['3484']")).count()
print("You can reach to TO with equal hops to LA from {} airport"\
                                                .format(num_airports_eq_hops))
```

```
[Stage 347:================================================>    (181 + 3) / 200]
```

```
You can reach to TO with equal hops to LA from 1244 airport
```

[33]:
```
# Find connected components & filter & set the checkpoint
sc.setCheckpointDir("user/s299165/lab")
# Proceed with connected components computation
connected = g.connectedComponents()
filtered_connected = connected.groupBy("component").count().filter("count >= 2")
```

```
24/01/25 21:43:57 WARN spark.SparkContext: Spark is not running in local mode,
therefore the checkpoint directory must not be on the local filesystem.
Directory 'user/s299165/lab' appears to be on the local filesystem.
```

[34]:
```
# Answer Question 4.5
filtered_connected.sort("count").show()
```

```
[Stage 510:>                                                        (0 + 1) / 1]
```

```
+------------+-----+
|   component|count|
+------------+-----+
|1460288880657|    2|
| 721554505752|    2|
| 266287972371|    4|
| 352187318292|    4|
|           23|    4|
| 300647710730|   10|
|           11| 3188|
+------------+-----+
```

[35]:
```
# Filter routes with 2 different airlines involving 5 cities
subgraph_routes = dfRoute_data_filtered.groupBy("airline_id").\
            agg(expr("count(distinct airport_source_id) as␣
 ↪num_source_airports"))
two_airlines = subgraph_routes.filter("num_source_airports >= 5").limit(2)
# Join with airline data to get the names and ICAO codes
two_airlines_info = two_airlines.join(dfAirline_data, on="airline_id",\
                                      how="inner").select("name", "icao").
 ↪distinct()
```

[36]:
```
# Answer Question 4.6
two_airlines_info.show()
```

```
+---------------+----+
|           name|icao|
+---------------+----+
|Binter Canarias| IBB|
|      First Air| FAB|
+---------------+----+
```

[37]:
```
#  Airports involved in routes with airlines
selected_airlines = two_airlines_info.select("icao").rdd.flatMap(lambda x: x).
 ↪collect()
selected_airports = dfRoute_data_filtered.filter(col("airline_id").
 ↪isin(selected_airlines)) \
    .select("airport_source_id", "airport_destination_id") \
    .rdd.flatMap(lambda x: x).distinct().collect()
# Extract the subgraph using selected airports
subgraph_vertices = g.vertices.filter(col("id").isin(selected_airports))
subgraph_edges = g.edges.filter((col("src").isin(selected_airports)) &␣
 ↪(col("dst").isin(selected_airports)))
```

[38]:
```
# Answer Question 4.7: Show the subgraph vertices and edges
subgraph = GraphFrame(subgraph_vertices, subgraph_edges)
#print("subgraph vertices :")
#subgraph.vertices.show()
#print("subgraph edges :")
#subgraph.edges.show()
```

5 Bonus Task

[39]:
```
# Find the neighbors of the starting airport (airport id = 2537)
first_hop = g.find("(a)-[e]->(b)").filter(f"a.id == '2537'").select("b.id").
 ↪distinct()
```

[40]:
```
# Find the second hop destinations excluding those in Belo Horizonte
second_hop_destinations = g.find("(a)-[e]->(b)").filter(col("e.src").
 ↪isin(first_hop.rdd.flatMap(lambda x: x).collect())
                                                        & ~col("e.dst").
 ↪isin(["2528", "2537"]))
```

[41]:
```
from pyspark.sql.functions import min as spark_min
# Find the destination airport at a minimum distance
min_distance_destination = (second_hop_destinations.groupBy("e.dst")
    .agg(spark_min("e.label").alias("min_distance")))
result_df = min_distance_destination.join(dfAirport_data,\
    on=min_distance_destination["dst"] == dfAirport_data["id"], how='inner')
```

[42]:
```
from pyspark.sql.functions import col, concat
from pyspark.sql.types import IntegerType
# Find the details of the destination airport
result_df_shortPath = (result_df.withColumn("name", concat(col("id"),␣
 ↪col("iata"))).withColumn("min_distance",\
            col("min_distance").cast(IntegerType())).orderBy("min_distance"))
```

[43]:
```
# Answer Question 5.1
final_result = result_df_shortPath.select("name", "city", "country", "icao").
 ↪show(1,truncate=False)
```

```
[Stage 542:================================================>    (184 + 10) / 200]
+-------+-----+-------+----+
|name   |city |country|icao|
+-------+-----+-------+----+
|1386ORY|Paris|France |LFPO|
+-------+-----+-------+----+
only showing top 1 row
```

[ ]:
```
```