

Spark – Flight Dataset

Keto rezultate janë marrë nga ekzekutimi i

<https://colab.research.google.com/drive/1hA40AloYgKOjsGDNITsMCPL-EPJDFrIE?usp=sharing>

1) Instalimi i spark

```
# Instalimi i spark. Behet vetem nje here.

# Step 1: Install Java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Step 2: Download Spark 3.4.1 (latest confirmed working version)
!wget -q https://archive.apache.org/dist/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

# Step 3: Extract Spark
!tar -xzf spark-3.4.1-bin-hadoop3.tgz

# Step 4: Install findspark
!pip install -q findspark

# Step 5: Set environment variables
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.4.1-bin-hadoop3"

import findspark
findspark.init()
```

2) Ngarkimi i dataset nga google Drive

```
#Na lejon me marre dataset qe eshte i ngarkuar ne google drive

!pip install -q gdown

# Download using the file ID
!gdown --id 1DB44cymNuJxmQTZ1SikWCpGJZnRKn_Ka --output flight_dataset.csv
```


3) Krijimi i Spark Session

```
# Krijon nje spark Session.
# Gjendja pass mbylljes se session nuk ruhet

from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("BigDataProject") \
    .getOrCreate()

spark
```

 **SparkSession - in-memory**

SparkContext

[Spark UI](#)

Version

v3.4.1

Master



local[*]

AppName

BigDataProject

4) E vendos datasetin si tabelë të përkohshme në memorie. Zgjidhni se cilat nga sessionet e spark

```
flight_dataset = spark.read.csv("flight_dataset.csv", header=True, inferSchema=True)
flight_dataset.createOrReplaceTempView("flight_dataset")
flight_dataset.show(100, truncate=False)
```

from_airport_code	from_country	dest_airport_code	dest_country	aircraft_type	airline_n
ALG	Algeria	AEP	Argentina	Airbus A318 Canadair RJ 1000 Airbus A330 Airbus A320	multi
ALG	Algeria	AEP	Argentina	Airbus A318 Canadair RJ 1000 Boeing 787 Airbus A320	multi
ALG	Algeria	AEP	Argentina	Airbus A320 Airbus A321 Boeing 787 Airbus A320	multi
ALG	Algeria	AEP	Argentina	Airbus A318 Airbus A320 Boeing 787 Airbus A320	multi
ALG	Algeria	AEP	Argentina	Airbus A321neo Boeing 777 Airbus A320	multi

5) Query 1: Fluturimet më të Lira kundrejt më Eco-Friendly sipas Rrugës dhe Numrit të Ndalimeve

```
# Query 1: Fluturimet më të Lira kundrejt më Eco-Friendly sipas Rrugës dhe Numrit të Ndalimeve

# -----
# 1) Krijo view me statistikat çmim/CO2 për çdo rrugë + numrin e ndalimeve
spark.sql("""
SELECT
    from_country,
    dest_country,
    stops,
    COUNT(*)                AS flight_count,
    MIN(price)              AS min_price,
    MIN(co2_emissions)      AS min_co2,
    AVG(price)              AS avg_price,
    AVG(co2_emissions)      AS avg_co2,
    ROUND(AVG(price) / NULLIF(AVG(co2_emissions), 0), 6) AS price_per_co2
FROM flight_dataset
WHERE price IS NOT NULL
    AND co2_emissions IS NOT NULL
GROUP BY from_country, dest_country, stops
HAVING flight_count > 5      -- vetëm rute me ≥5 fluturime
ORDER BY price_per_co2 ASC
""").createOrReplaceTempView("eco_price_stats")

# -----
# 2) Shfaq 50 rreshtat e parë (instruktori sheh rezultatet e plota)
spark.sql("""
SELECT *
FROM   eco_price_stats
ORDER BY price_per_co2 ASC
LIMIT 50
""").show(truncate=False)
```

```
# 3) Grafiku: Bar-chart me 20 rruget më efikase (çmim/CO2), ngjyra = nr.ndalimesh
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

eco_df = spark.sql("SELECT * FROM eco_price_stats").toPandas()

top20 = (
    eco_df.sort_values("price_per_co2")
    .head(20)
    .drop_duplicates(subset=["from_country", "dest_country"]) # një rresht për rrugë
)

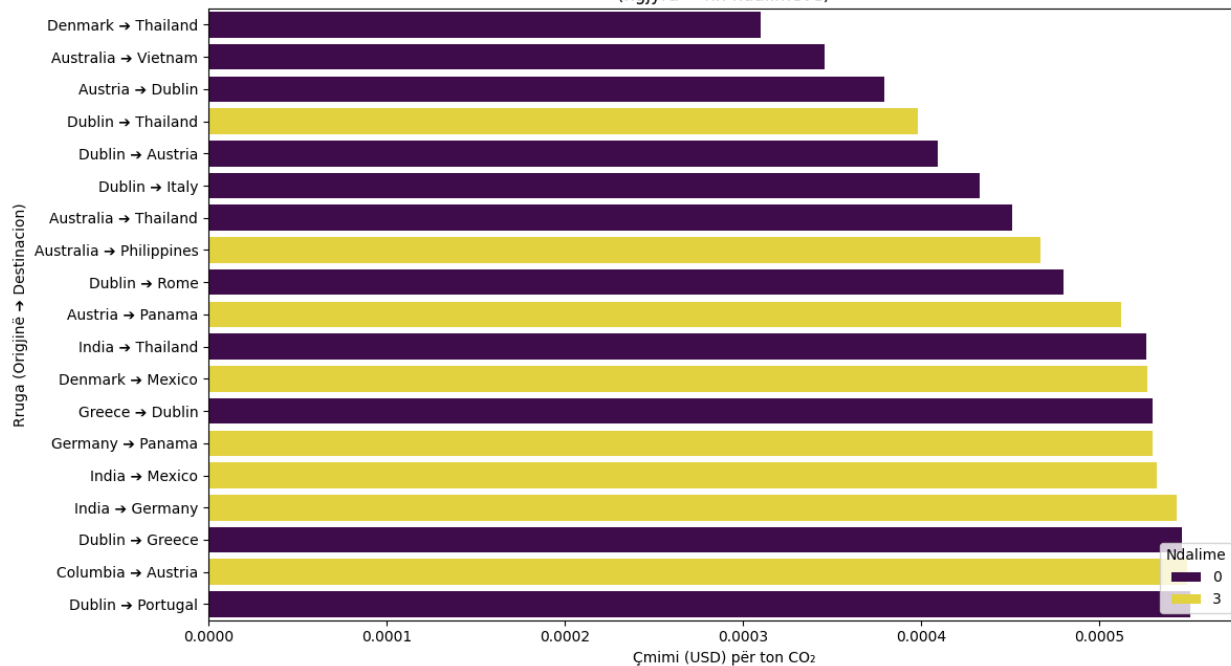
# Etiketë më të shkurtër për boshtin Y
top20["route"] = (
    top20["from_country"].str[:15] +
    " → " +
    top20["dest_country"].str[:15]
)

plt.figure(figsize=(12, 7))
sns.barplot(
    data=top20,
    y="route",
    x="price_per_co2",
    hue="stops",
    palette="viridis",
    dodge=False
)

plt.xlabel("Çmimi (USD) për ton CO2")
plt.ylabel("Rruga (Origjinë → Destinacion)")
plt.title("Top-20 Rrutat më të Lira & më Efiçente ndaj CO2-s\n(ngjyra = nr. ndalimeve)")
plt.legend(title="Ndalime", loc="lower right")
plt.tight_layout()
plt.show()
```

from_country	dest_country	stops	flight_count	min_price	min_co2	avg_price	avg_co2	price_per_co2
Denmark	Thailand	0	6	237.0	1022000	316.3333333333333	1022000.0	3.1E-4
Australia	Vietnam	0	9	180.0	606000	228.1111111111111	659777.777777778	3.46E-4
Austria	Dublin	0	11	25.0	186000	73.8181818181818	194909.090909090	3.79E-4
Dublin	Thailand	3	10	367.0	1237000	548.9	1380100.0	3.98E-4
Dublin	Austria	0	12	28.0	185000	78.9166666666667	192916.666666667	4.09E-4
Denmark	Thailand	3	9	451.0	1161000	598.555555555555	1404000.0	4.26E-4
Dublin	Italy	0	12	14.0	158000	68.5833333333333	158416.666666667	4.33E-4
Australia	Thailand	0	19	166.0	602000	332.4210526315789	737684.210526315	4.51E-4
Australia	Philippines	3	7	292.0	964000	498.0	1066428.571428571	4.67E-4
Dublin	Rome	0	14	69.0	202000	96.9285714285714	202000.0	4.8E-4
Austria	Panama	3	10	339.0	897000	606.8	1184500.0	5.12E-4
India	Thailand	0	42	97.0	263000	173.4047619047619	329952.380952380	5.26E-4
Denmark	Mexico	3	12	574.0	946000	668.416666666667	1268083.33333333	5.27E-4
Greece	Dublin	0	8	88.0	254000	152.75	288000.0	5.3E-4
Germany	Panama	3	8	476.0	918000	637.75	1204125.0	5.3E-4
India	Mexico	3	14	914.0	1587000	1452.71428571428	2728571.42857142	5.32E-4
India	Germany	3	20	354.0	756000	447.8	824000.0	5.43E-4
Dublin	Greece	0	8	94.0	254000	157.125	288000.0	5.46E-4
Columbia	Austria	3	7	622.0	1136000	709.428571428571	1293142.85714285	5.49E-4
Dublin	Portugal	0	44	30.0	157000	107.045454545455	194431.818181818	5.51E-4

Top-20 Rutat më të Lira & më Efiçente ndaj CO₂-s
(ngjyra = nr. ndalimeve)



6) Query 2: Sa % MË LIRË apo MË SHUMË kushton të shtosh ndalesa?

```
# QUERY 2: Sa % MË LIRË apo MË SHUMË kushton të shtosh ndalesa?

spark.sql("""
SELECT
    from_country,
    dest_country,
    COUNT(*) AS total_flights,
    SUM(CASE WHEN stops = 0 THEN 1 ELSE 0 END) AS direct_flights,
    SUM(CASE WHEN stops > 0 THEN 1 ELSE 0 END) AS multi_flights,
    ROUND(AVG(CASE WHEN stops = 0 THEN price END), 2) AS avg_price_direct,
    ROUND(AVG(CASE WHEN stops > 0 THEN price END), 2) AS avg_price_multi,
    ROUND(
        (AVG(CASE WHEN stops = 0 THEN price END) -
         AVG(CASE WHEN stops > 0 THEN price END))
        / AVG(CASE WHEN stops = 0 THEN price END) * 100
        , 2) AS pct_savings_vs_direct -- pozitive = më lirë me ndalesa
FROM flight_dataset
WHERE price IS NOT NULL AND stops IS NOT NULL
GROUP BY from_country, dest_country
HAVING direct_flights >= 5 AND multi_flights >= 5
ORDER BY pct_savings_vs_direct DESC
""").createOrReplaceTempView("stop_price_compare_fixed")

# -----
# 1) PRINT top 20 rreshtat
spark.sql("""
SELECT *
FROM stop_price_compare_fixed
ORDER BY pct_savings_vs_direct DESC
LIMIT 20
""").show(truncate=False)
```

```
# 2) Grafik kolonash: Çmimi mesatar direct vs multi (Top 10 kursimet më të mëdha)
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plot_df = spark.sql("""
SELECT from_country,
       dest_country,
       avg_price_direct,
       avg_price_multi,
       pct_savings_vs_direct
FROM stop_price_compare_fixed
ORDER BY pct_savings_vs_direct DESC
LIMIT 10
""").toPandas()

# Label rruge
plot_df["route"] = plot_df["from_country"].str[:12] + " → " + plot_df["dest_country"].str[:12]

plt.figure(figsize=(12,6))
bar_width = 0.4
positions = range(len(plot_df))

# Kolona: Çmimi direct
plt.bar(
    [p - bar_width/2 for p in positions],
    plot_df["avg_price_direct"],
    width=bar_width,
    label="Direct ($)",
    color="steelblue"
)

# Kolona: Çmimi multi-stop
plt.bar(
    [p + bar_width/2 for p in positions],
    plot_df["avg_price_multi"],
    width=bar_width,
    label="Multi-stop ($)",
    color="steelblue"
)
```

```
# Linjë: % kursimi mbi boshtin dytësor
plt.twinx()
plt.plot(
    positions,
    plot_df["pct_savings_vs_direct"],
    color="green",
    marker="o",
    linewidth=2,
    label="% Kursim (multi vs direct)"
)

plt.xticks(positions, plot_df["route"], rotation=45, ha="right")
plt.ylabel("Çmimi i Mesatar ($)")
plt.title("Direct vs Multi-stop - Top 10 Rrutat me Kursimin më të Madh")
plt.legend(loc="upper left")
plt.grid(axis="y", linestyle="--", alpha=0.4)
plt.tight_layout()
plt.show()
```

from_country	dest_country	total_flights	direct_flights	multi_flights	avg_price_direct	avg_price_multi	pct_savings_vs_direct
Chile	Mexico	921	13	908	2370.69	845.13	64.35
Brazil	Mexico	1924	10	1914	3233.5	2257.0	30.2
Chile	Spain	696	13	683	2422.54	1807.86	25.37
Germany	Singapore	1478	30	1448	2811.1	2143.93	23.73
France	Netherlands	1157	67	1090	446.91	350.51	21.57
Denmark	United States	10021	25	9996	1483.28	1181.63	20.34
Canada	Rome	1064	11	1053	1185.45	992.27	16.3
Austria	Zurich	867	48	819	500.9	444.7	11.22
France	United States	9508	190	9318	1972.87	1791.21	9.21
France	Canada	961	19	942	1581.79	1439.84	8.97
Greece	Canada	962	5	957	1614.0	1522.85	5.65
France	Mexico	804	22	782	2016.18	1924.26	4.56
Dublin	United Arab Emirates	654	7	647	721.0	694.7	3.65
Germany	Zurich	2113	101	2012	438.81	425.82	2.96
Brazil	Peru	2140	13	2127	920.92	895.03	2.81
Brazil	Zurich	1444	5	1439	2831.6	2761.48	2.48
Brazil	Netherlands	1008	7	1001	2204.0	2165.33	1.75
Germany	Sweden	2002	72	1930	445.67	443.85	0.41
Dublin	Sweden	876	6	870	367.67	368.05	-0.1
Germany	Canada	1884	34	1850	1434.29	1444.65	-0.72

