pegah

November 8, 2024

[7]:
```
# Import necessary libraries
import findspark
findspark.init()

from pyspark.sql import SparkSession, functions as F
from pyspark.sql.types import StructType, StructField, StringType, IntegerType

# Step 1: Initialize Spark session
spark = SparkSession.builder \
    .appName("CleaningData") \
    .master("local[*]") \
    .config("spark.driver.memory", "4g") \
    .config("spark.executor.memory", "5g") \
    .config("spark.shuffle.memoryFraction", "0.5") \
    .config("spark.storage.memoryFraction", "0.5") \
    .config("spark.driver.maxResultSize", "0") \
    .getOrCreate()

# Step 2: Define schema for the Netflix dataset

# Netflix schema
netflix_schema = StructType([
    StructField("show_id", StringType(), True),
    StructField("type", StringType(), True),
    StructField("title", StringType(), True),
    StructField("director", StringType(), True),
    StructField("cast", StringType(), True),
    StructField("country", StringType(), True),
    StructField("date_added", StringType(), True),
    StructField("release_year", StringType(), True),
    StructField("rating", StringType(), True),
    StructField("duration", StringType(), True),
    StructField("listed_in", StringType(), True),
    StructField("description", StringType(), True)
])

# Step 3: Load the Netflix dataset from HDFS
```

```
netflix_df = spark.read.csv("hdfs://localhost:9000/user/reviews/netflix_titles.
 ↪csv",
                            schema=netflix_schema, header=True)

# Step 4: Clean and preprocess the Netflix dataset
netflix_cleaned = netflix_df.dropna(how='any') \
    .withColumn("country", F.trim(F.col("country"))) \
    .withColumn("release_year", F.col("release_year").cast(IntegerType())) \
    .filter(F.col("release_year") >= 2000)

# Step 5: Save the cleaned Netflix dataset back to HDFS
netflix_cleaned.write.csv("hdfs://localhost:9000/user/reviews/
 ↪cleaned_netflix_titles",
                          header=True, mode='overwrite')

# Step 6: Display the cleaned data
netflix_cleaned.show(5)
```

```
24/11/08 21:32:33 WARN Utils: Your hostname, dsbda-vm resolves to a loopback
address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
24/11/08 21:32:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
24/11/08 21:32:47 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable
+-------+-------+-------------------+-------------------+--------------------+-
------------------+-----------------+------------+------+---------+-----------
---------+--------------------+
|show_id|   type|              title|           director|                cast|
country|        date_added|release_year|rating| duration|           listed_in|
description|
+-------+-------+-------------------+-------------------+--------------------+-
------------------+-----------------+------------+------+---------+-----------
---------+--------------------+
|     s9|TV Show|The Great British…|    Andy Devonshire|Mel Giedroyc, Sue…|
United Kingdom|September 24, 2021|        2021| TV-14|9 Seasons|British TV
Shows,…|A talented batch …|
|    s10|  Movie|        The Starling|     Theodore Melfi|Melissa McCarthy,…|
United States|September 24, 2021|        2021| PG-13|  104 min|    Comedies,
Dramas|A woman adjusting…|
|    s13|  Movie|        Je Suis Karl|Christian Schwochow|Luna Wedler,
Jann…|Germany, Czech Re…|September 23, 2021|        2021| TV-MA|  127
min|Dramas, Internati…|After most of her…|
|    s28|  Movie|           Grown Ups|       Dennis Dugan|Adam Sandler, Kev…|
```

```
United States|September 20, 2021|        2010| PG-13|  103 min|
Comedies|Mourning the loss…|
|    s29|  Movie|          Dark Skies|      Scott Stewart|Keri Russell, Jos…|
United States|September 19, 2021|        2013| PG-13|   97 min|Horror Movies,
Sc…|A family's idylli…|
+-------+-------+--------------------+-------------------+--------------------+-
------------------+-----------------+------------+------+---------+----------
---------+--------------------+
only showing top 5 rows
```

[2]:
```
!pip install hdfs
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting hdfs
  Downloading hdfs-2.7.3.tar.gz (43 kB)

43.5/43.5 kB 235.8 kB/s eta 0:00:001m595.0 kB/s
eta 0:00:01
  Preparing metadata (setup.py) … done
Collecting docopt (from hdfs)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: requests>=2.7.0 in
./.local/lib/python3.8/site-packages (from hdfs) (2.32.3)
Requirement already satisfied: six>=1.9.0 in /usr/lib/python3/dist-packages
(from hdfs) (1.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
./.local/lib/python3.8/site-packages (from requests>=2.7.0->hdfs) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests>=2.7.0->hdfs) (2.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests>=2.7.0->hdfs) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests>=2.7.0->hdfs) (2019.11.28)
Building wheels for collected packages: hdfs, docopt
  Building wheel for hdfs (setup.py) … done
  Created wheel for hdfs: filename=hdfs-2.7.3-py3-none-any.whl size=34321
sha256=f1b061639deb5bef7499a5b381c5d18a2aabfb4dcc2cc6a096b6bd702bdf1f5e
  Stored in directory: /home/ubuntu/.cache/pip/wheels/68/dd/29/c1a590238f9ebbe4f
7ee9b3583f5185d0b9577e23f05c990eb
  Building wheel for docopt (setup.py) … done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl
size=13704
sha256=ce10b2794fc44b87e4627fb34317eb2f42d1f3af2258f6506e356e0e56188e73
  Stored in directory: /home/ubuntu/.cache/pip/wheels/56/ea/58/ead137b087d9e3268
52a851351d1debf4ada529b6ac0ec4e8c
Successfully built hdfs docopt
```

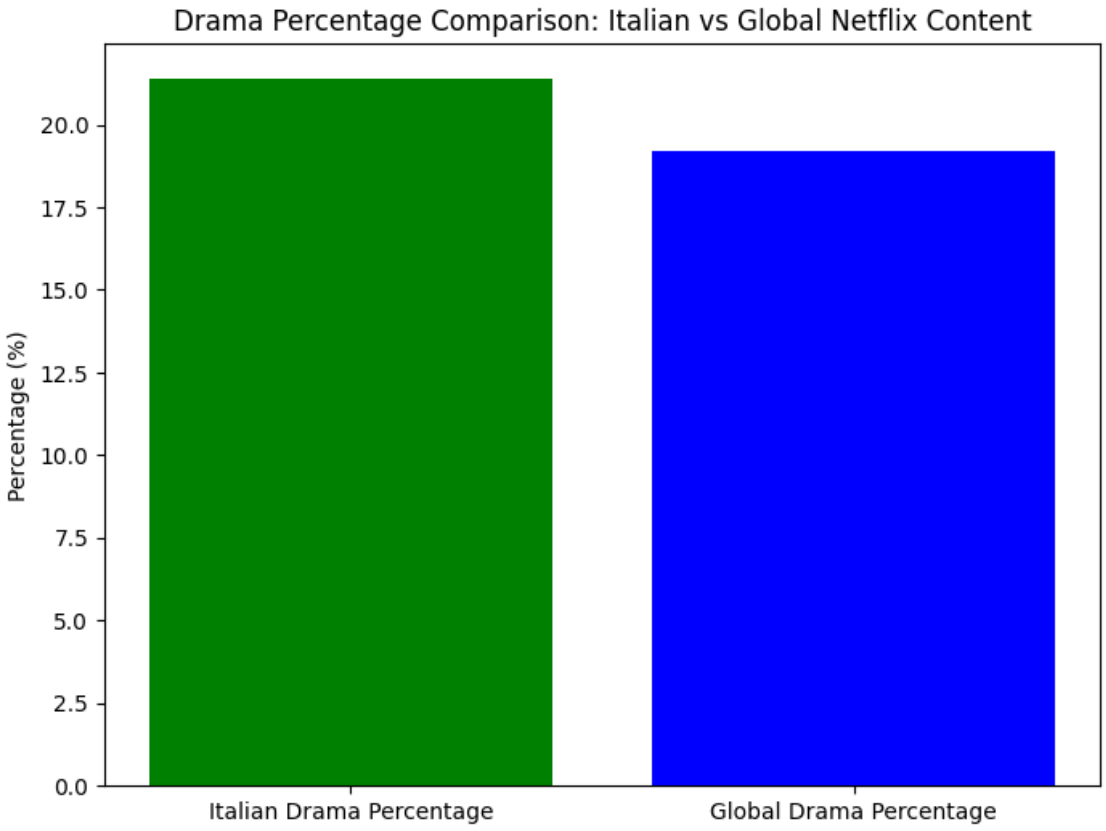
```
DEPRECATION: distro-info 0.23ubuntu1 has a non-standard version number. pip
24.1 will enforce this behaviour change. A possible replacement is to upgrade to
a newer version of distro-info or contact the author to suggest that they
release a version with a conforming version number. Discussion can be found at
https://github.com/pypa/pip/issues/12063
DEPRECATION: python-debian 0.1.36ubuntu1 has a non-standard version
number. pip 24.1 will enforce this behaviour change. A possible replacement is
to upgrade to a newer version of python-debian or contact the author to suggest
that they release a version with a conforming version number. Discussion can be
found at https://github.com/pypa/pip/issues/12063
Installing collected packages: docopt, hdfs
Successfully installed docopt-0.6.2 hdfs-2.7.3

[notice] A new release of pip is
available: 24.0 -> 24.3.1
[notice] To update, run:
python3 -m pip install --upgrade pip
```

[4]:
```python
from hdfs import InsecureClient
import pandas as pd
import io
import matplotlib.pyplot as plt

# Connect to HDFS - using `localhost` and `9870
hdfs_client = InsecureClient('http://localhost:9870', user='ubuntu')

# Path to the output file on HDFS
output_file_path = '/user/ubuntu/output/part-00000'

# Read the file directly from HDFS
with hdfs_client.read(output_file_path, encoding='utf-8') as reader:
    # Load into a DataFrame
    data = pd.read_csv(reader, sep='\t', header=None, names=['Genre',␣
 ↪'Country', 'Count'])

# Display the first few rows
print("Data preview:")
print(data.head(50))

# Step 1: Pivot the Data
pivot_data = data.pivot(index='Genre', columns='Country', values='Count').
 ↪fillna(0).reset_index()
```

```python
# Step 2: Calculate Total Counts
total_italy_count = pivot_data['Italy'].sum()
total_global_count = pivot_data['Global'].sum()

# Step 3: Calculate Drama Percentages
drama_row = pivot_data[pivot_data['Genre'] == 'Dramas']
italy_drama_count = drama_row['Italy'].values[0] if not drama_row.empty else 0
global_drama_count = drama_row['Global'].values[0] if not drama_row.empty else 0

italian_drama_percentage = (italy_drama_count / total_italy_count) * 100 if␣
 ↪total_italy_count > 0 else 0
global_drama_percentage = (global_drama_count / total_global_count) * 100 if␣
 ↪total_global_count > 0 else 0

print(f"Italian Drama Percentage: {italian_drama_percentage:.2f}%")
print(f"Global Drama Percentage: {global_drama_percentage:.2f}%")

# Step 4: Visualization
percentages = [italian_drama_percentage, global_drama_percentage]
labels = ['Italian Drama Percentage', 'Global Drama Percentage']

plt.figure(figsize=(8,6))
plt.bar(labels, percentages, color=['green', 'blue'])
plt.title('Drama Percentage Comparison: Italian vs Global Netflix Content')
plt.ylabel('Percentage (%)')
plt.show()
```

```
Data preview:
                       Genre Country  Count
0                       2016   Italy      0
1                       2016  Global      1
2                     98 min   Italy      0
3                     98 min  Global      1
4          Action & Adventure   Italy      4
5          Action & Adventure  Global    672
6              Anime Features   Italy      0
7              Anime Features  Global     58
8                Anime Series   Italy      0
9                Anime Series  Global     10
10           British TV Shows   Italy      0
11           British TV Shows  Global     21
12  Children & Family Movies   Italy      3
13  Children & Family Movies  Global    464
14         Classic & Cult TV   Italy      0
15         Classic & Cult TV  Global      3
16            Classic Movies   Italy      0
```

```
17            Classic Movies  Global      1
18                  Comedies   Italy     10
19                  Comedies  Global   1401
20            Crime TV Shows   Italy      2
21            Crime TV Shows  Global     35
22               Cult Movies   Italy      0
23               Cult Movies  Global     26
24               December 12   Italy      0
25               December 12  Global      1
26             Documentaries   Italy      5
27             Documentaries  Global    372
28                Docuseries   Italy      1
29                Docuseries  Global     12
30                    Dramas   Italy     28
31                    Dramas  Global   2051
32      Faith & Spirituality   Italy      1
33      Faith & Spirituality  Global     57
34             Horror Movies   Italy      2
35             Horror Movies  Global    315
36        Independent Movies   Italy      4
37        Independent Movies  Global    695
38      International Movies   Italy     40
39      International Movies  Global   2177
40    International TV Shows   Italy      1
41    International TV Shows  Global     85
42                 Kids' TV   Italy      3
43                 Kids' TV  Global     10
44           Korean TV Shows   Italy      0
45           Korean TV Shows  Global     10
46              LGBTQ Movies   Italy      1
47              LGBTQ Movies  Global     74
48                    Movies   Italy      0
49                    Movies  Global     19
Italian Drama Percentage: 21.37%
Global Drama Percentage: 19.22%
```



[6]:
```
from scipy.stats import chi2_contingency

# Create a contingency table
# Assuming `italy_drama_count`, `total_italy_count`, `global_drama_count`, and␣
 ↪`total_global_count` are already defined
# Non-drama counts are the total count minus the drama count for each group
italy_non_drama_count = total_italy_count - italy_drama_count
global_non_drama_count = total_global_count - global_drama_count

# Contingency table format:
# [[Italy Drama, Italy Non-Drama], [Global Drama, Global Non-Drama]]
contingency_table = [
    [italy_drama_count, italy_non_drama_count],
    [global_drama_count, global_non_drama_count]
]

# Perform Chi-square test
chi2, p, _, _ = chi2_contingency(contingency_table)

# Display the test result with 4 decimal places
```

```
print(f"Chi-square value: {chi2:.4f}")
print(f"P-value: {p:.4f}")

# Interpret the p-value
alpha = 0.05
if p < alpha:
    print("The difference in drama proportions is statistically significant.")
else:
    print("The difference in drama proportions is not statistically significant.
 ↪")
```

```
Chi-square value: 0.2601
P-value: 0.6101
The difference in drama proportions is not statistically significant.
```

[ ]: