Spark Preparation

We check if we are in Google Colab. If this is the case, install all necessary packages.

To run spark in Colab, we need to first install all the dependencies in Colab environment i.e. Apache Spark 3.3.2 with hadoop 3.3, Java 8 and Findspark to locate the spark in the system. The tools installation can be carried out inside the Jupyter Notebook of the Colab. Learn more from [A Must-Read Guide on How to Work with PySpark on Google Colab for Data Scientists!]

```
try:
  import google.colab
  IN_COLAB = True
except:
  IN_COLAB = False
```

+ Code + Text

```
if IN_COLAB:
    !apt-get install openjdk-8-jdk-headless -qq > /dev/null
    !wget https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hado
    !tar xf spark-3.2.0-bin-hadoop3.2.tgz
    !mv spark-3.2.0-bin-hadoop3.2 spark
    !pip install -q findspark
    import os
    os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
    os.environ["SPARK_HOME"] = "/content/spark"
```

--NORMAL--

Start a Local Cluster

```
import findspark
findspark.init()
from pyspark.sql import SparkSession

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

# Verify Spark session
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
`v3.2.0`
Master
`local[*]`
AppName
`ColabSparkApp`

Spark Assignment

Based on the movie review dataset in 'netflix-rotten-tomatoes-metacritic-imdb.csv', answer the below questions.

Note: do not clean or remove missing data

```
# df = spark.read.csv("/content/my_file.csv", header=True, inferSchema=True)
df = spark.read.csv("netflix-rotten-tomatoes-metacritic-imdb.csv", header=True,
df.show()
df.printSchema()
df.describe().show()
print(df.count())
print(df.columns)
```

```
|            Instynkt|              Crime|TV Dramas,Crime T...|
|       Only a Mother|              Drama|Social Issue Dram...|
|          Snowroller|             Comedy|Sports Movies,Spo...|Swedish, F
|        Sunes Summer|Comedy, Family, F...|Children & Family...|
|       The Invisible|Crime, Drama, Fan...|Thriller Movies,M...|
|The Simple Minded...|              Drama|Social Issue Dram...|    Scania
|The Stig-Helmer S...|      Comedy, Drama|Romantic Dramas,R...|    Swedis
|     To Kill a Child|       Short, Drama|Dramas,Swedish Mo...|
```

```
|               Joker|Crime, Drama, Thr...|Dark Comedies,Cri...|
|                  I|Action, Adventure...|Dramas,Swedish Mo...|   Englis
|    Harrys Daughters|Adventure, Drama,...|Dramas,Swedish Mo...|
|       Gyllene Tider|               Music|Music & Musicals,...|
|      False As Water|     Drama, Thriller|Psychological Thr...|
+--------------------+--------------------+--------------------+---------
only showing top 20 rows

root
 |-- Title: string (nullable = true)
 |-- Genre: string (nullable = true)
 |-- Tags: string (nullable = true)
 |-- Languages: string (nullable = true)
 |-- Series or Movie: string (nullable = true)
 |-- Hidden Gem Score: double (nullable = true)
 |-- Country Availability: string (nullable = true)
 |-- Runtime: string (nullable = true)
 |-- Director: string (nullable = true)
 |-- Writer: string (nullable = true)
 |-- Actors: string (nullable = true)
 |-- View Rating: string (nullable = true)
 |-- IMDb Score: string (nullable = true)
 |-- Rotten Tomatoes Score: string (nullable = true)
 |-- Metacritic Score: string (nullable = true)
 |-- Awards Received: double (nullable = true)
 |-- Awards Nominated For: double (nullable = true)
 |-- Boxoffice: string (nullable = true)
 |-- Release Date: string (nullable = true)
 |-- Netflix Release Date: string (nullable = true)
 |-- Production House: string (nullable = true)
 |-- Netflix Link: string (nullable = true)
 |-- IMDb Link: string (nullable = true)
 |-- Summary: string (nullable = true)
 |-- IMDb Votes: string (nullable = true)
 |-- Image: string (nullable = true)
 |-- Poster: string (nullable = true)
 |-- TMDb Trailer: string (nullable = true)
 |-- Trailer Site: string (nullable = true)

+-------+--------------------+------------------+-----------------------
|summary|               Title|             Genre|
+-------+--------------------+------------------+-----------------------
|  count|               15480|             13770|
|   mean|              1131.0|              null|
| stddev|   990.1592318146744|              null|
|    min|"TV series ""Keep...|            Action|                 20th Cen
|    max|                침묵|Western, Adventure|海外コメディ,ロマンチックコメテ
+-------+--------------------+------------------+-----------------------

15480
['Title', 'Genre', 'Tags', 'Languages', 'Series or Movie', 'Hidden Gem Sc
```

What is the maximum and average of the overall hidden gem score?

```
from pyspark.sql import functions as F
HIDDEN_GEM_SCORE=("Hidden Gem Score")
average_hidden_gem_score = df.select(F.avg(HIDDEN_GEM_SCORE)).collect()[0][0]
print(f'average_hidden_gem_score: {average_hidden_gem_score}')
max_hidden_gem_score = df.select(F.max(HIDDEN_GEM_SCORE)).collect()[0][0]
print(f'max_hidden_gem_score: {max_hidden_gem_score}')
```

```
average_hidden_gem_score: 5.937551386501234
max_hidden_gem_score: 9.8
```

How many movies that are available in Korea?

```
number_of_movies_available_in_korea = df.filter(df["Languages"].rlike('Korea'))
print(number_of_movies_available_in_korea)
```

```
735
```

Which director has the highest average hidden gem score?

```
DIRECTOR='Director'
director,score=df.groupBy(DIRECTOR).agg(
    F.avg(HIDDEN_GEM_SCORE).alias("Average_Hidden_Gem_Score")
).orderBy(F.col("Average_Hidden_Gem_Score").desc()).first()
print(f'director: {director}, score: {score}')
```

```
director: Dorin Marcu, score: 9.8
```

How many genres are there in the dataset?

```
GENRE = 'Genre'
genres = [genre[0] for genre in (
    df.withColumn(GENRE, F.split(df[GENRE],","))
    .withColumn(GENRE, F.explode(F.col(GENRE)))
    .withColumn(GENRE, F.trim(F.col(GENRE)))
    ).select(GENRE).collect()]

print(f'number of distinct genres: {len(set(genres))}')
```

```
number of distinct genres: 28
```