

rs-spark

December 15, 2023

1 Practice: Big Data Recommender Systems with Spark :

NAME : BOUABD CHAIMAA

1.1 Install pyspark

```
[2]: pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.0.tar.gz (316.9 MB)
    316.9/316.9
MB 4.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.0-py2.py3-none-any.whl
  size=317425345
  sha256=ac35a3bf076ff47b5bc608c9c2e33c9c31e7d353c2b62d22d0b1406f4fd30544
  Stored in directory: /root/.cache/pip/wheels/41/4e/10/c2cf2467f71c678cfc8a6b9ac9241e5e44a01940da8fbb17fc
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.0
```

1.2 Step 1: Setting Up Your Spark Session and Load data

```
[5]: from pyspark.sql import SparkSession
from pyspark.ml.recommendation import ALS

spark = SparkSession.builder.appName("Recommender system").getOrCreate()

input_file_path = '/content/drive/MyDrive/movielens20m/rating.csv'

data = spark.read.csv(input_file_path, header=True, inferSchema=True)
```

```
(training, test) = data.randomSplit([0.8, 0.2])
```

1.3 STEP 3: : Building the Collaborative Filtering Recommender System with ALS

```
[6]: training = training.drop('timestamp')

als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating",
          ↪nonnegative=True, coldStartStrategy="drop")

model = als.fit(training)
```

1.4 SETP 4: Generating Recommendations and Converting to Pandas DataFrames

```
[7]: userRecs = model.recommendForAllUsers(10)
movieRecs = model.recommendForAllItems(10)

userRecs_df = userRecs.toPandas()
movieRecs_df = movieRecs.toPandas()
```

1.5 STEP 5: Store Recommendations in Files

```
[8]: userRecs_df.to_csv('user_recommendations.csv', index=False)
movieRecs_df.to_csv('movie_recommendations.csv', index=False)
```