spark_setup

April 23, 2023

```
[ ]: !apt-get update
```

```
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/ InRelease
Hit:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64
InRelease
Get:6 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:7 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal InRelease [18.1
kB]
Hit:8 http://ppa.launchpad.net/cran/libgit2/ubuntu focal InRelease
Hit:9 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Hit:10 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu focal InRelease
Hit:11 http://ppa.launchpad.net/ubuntugis/ppa/ubuntu focal InRelease
Get:12 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal/main Sources
[2,445 kB]
Get:13 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu focal/main amd64
Packages [1,156 kB]
Fetched 3,955 kB in 3s (1,346 kB/s)
Reading package lists… Done
```

```
[ ]: !apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

```
[ ]: !wget -q https://downloads.apache.org/spark/spark-3.4.0/spark-3.4.0-bin-hadoop3.
      ↪tgz
```

```
[ ]: !tar -xvf ./spark-3.4.0-bin-hadoop3.tgz
```

```
[ ]: !pip install -q findspark
     import os
     os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
     os.environ["SPARK_HOME"] = "/content/spark-3.4.0-bin-hadoop3"
```

```
[ ]: import findspark
     findspark.init()
```

[ ]:

```python
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

[ ]:

```python
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

[ ]:

```python
df = spark.read.csv('ratings.csv', header=True, inferSchema=True)
cols = df.columns
df.printSchema()
```

```
root
 |-- userID: integer (nullable = true)
 |-- trackID: integer (nullable = true)
 |-- recommendation: string (nullable = true)
 |-- album: integer (nullable = true)
 |-- artist: integer (nullable = true)
 |-- num_genre_ratings: integer (nullable = true)
 |-- max: integer (nullable = true)
 |-- min: integer (nullable = true)
 |-- mean: double (nullable = true)
 |-- variance: double (nullable = true)
 |-- median: integer (nullable = true)
```

[ ]:

```python
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving testItem.data.zip to testItem.data.zip
Saving trainItem.data.zip to trainItem.data.zip
```

[ ]:

```
!unzip ./testItem.data.zip
!unzip ./trainItem.data.zip
```

```
Archive:  ./testItem.data.zip
  inflating: testItem.data
Archive:  ./trainItem.data.zip
  inflating: trainItem.data
```

[ ]:

```python
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
```

[ ]:

```python
training = spark.read.csv("trainItem.data", header = False)
training.show(5)
```

```
+------+------+---+
|   _c0|   _c1|_c2|
+------+------+---+
|199808|248969| 90|
|199808|  2663| 90|
|199808| 28341| 90|
|199808| 42563| 90|
|199808| 59092| 90|
+------+------+---+
only showing top 5 rows
```

[ ]:
```
training = training.withColumnRenamed("_c0", "userID").withColumnRenamed("_c1",␣
 ↪"itemID").withColumnRenamed("_c2", "rating")
training.show(5)
```

```
+------+------+------+
|userID|itemID|rating|
+------+------+------+
|199808|248969|    90|
|199808|  2663|    90|
|199808| 28341|    90|
|199808| 42563|    90|
|199808| 59092|    90|
+------+------+------+
only showing top 5 rows
```

[ ]:
```
from pyspark.sql.types import IntegerType
training = training.withColumn("userID", training["userID"].cast(IntegerType()))
training = training.withColumn("itemID", training["itemID"].cast(IntegerType()))
training = training.withColumn("rating", training["rating"].cast('float'))
training.show(3)
```

```
+------+------+------+
|userID|itemID|rating|
+------+------+------+
|199808|248969|  90.0|
|199808|  2663|  90.0|
|199808| 28341|  90.0|
+------+------+------+
only showing top 3 rows
```

[ ]:
```
als = ALS(
    maxIter=5,
    rank = 5,
    regParam=0.01,
    userCol="userID",
    itemCol="itemID",
    ratingCol="rating",
    nonnegative = True,
    implicitPrefs = False,
    coldStartStrategy="drop"
)
```

[ ]:

```
model = als.fit(training)
```

[ ]:

```
testing = spark.read.csv("testItem.data", header = False)
```

[ ]:

```
testing = testing.withColumnRenamed("_c0", "userID").withColumnRenamed("_c1",␣
 ↪"itemID").withColumnRenamed("_c2", "rating")
testing.show(5)
```

```
+------+------+------+
|userID|itemID|rating|
+------+------+------+
|199810|208019|     0|
|199810| 74139|     0|
|199810|  9903|     0|
|199810|242681|     0|
|199810| 18515|     0|
+------+------+------+
only showing top 5 rows
```

[ ]:

```
testing = testing.withColumn("userID", testing["userID"].cast(IntegerType()))
testing = testing.withColumn("itemID", testing["itemID"].cast(IntegerType()))
testing = testing.withColumn("rating", testing["rating"].cast('float'))
testing.show(3)
```

```
+------+------+------+
|userID|itemID|rating|
+------+------+------+
|199810|208019|   0.0|
|199810| 74139|   0.0|
|199810|  9903|   0.0|
+------+------+------+
only showing top 3 rows
```

[ ]:

```
predictions = model.transform(testing)
predictions.show(5)
```

```
+------+------+------+----------+
|userID|itemID|rating|prediction|
+------+------+------+----------+
|233686|     1|   0.0| 23.677156|
|215400|     3|   0.0|  56.50849|
|224379|     5|   0.0| 34.220764|
|200179|    13|   0.0| 52.283817|
|199859|    17|   0.0| 31.217907|
+------+------+------+----------+
only showing top 5 rows
```

[ ]:

```
predictions.coalesce(1).write.csv("predictions")
```

[ ]:

```
predictions.toPandas().to_csv('myprediction.csv')
```

[ ]:

```

```