

MovieLens 数据集的 ETL 与探索性分析

*ETL代表提取（Extract）、转换（Transform）和加载（Load）

实验目的

- 在所构建的集群上运行程序。
- 掌握在集群上做数据清洗、转换与聚合操作。
- 学习如何用 Spark 进行大规模的统计分析。

数据集

- **MovieLens-1M** 数据集是美国明尼苏达大学 **GroupLens** 研究小组发布的电影推荐系统研究用标准数据集之一。它包含约 **100 万条评分记录**，由 6,000 多名用户对近 4,000 部电影的评分组成。
- 评分范围：1-5 分（整数）
- 用户特征：性别、年龄、职业、邮编
- 电影特征：标题、类型（可多标签）

数据集的详细信息请参考：<https://grouplens.org/datasets/movielens/1m/>

数据集

HDFS (Hadoop)

Google Cloud Storage (GCS)

S3 (Amazon)

阿里 OSS



Spark

存储系统

计算框架

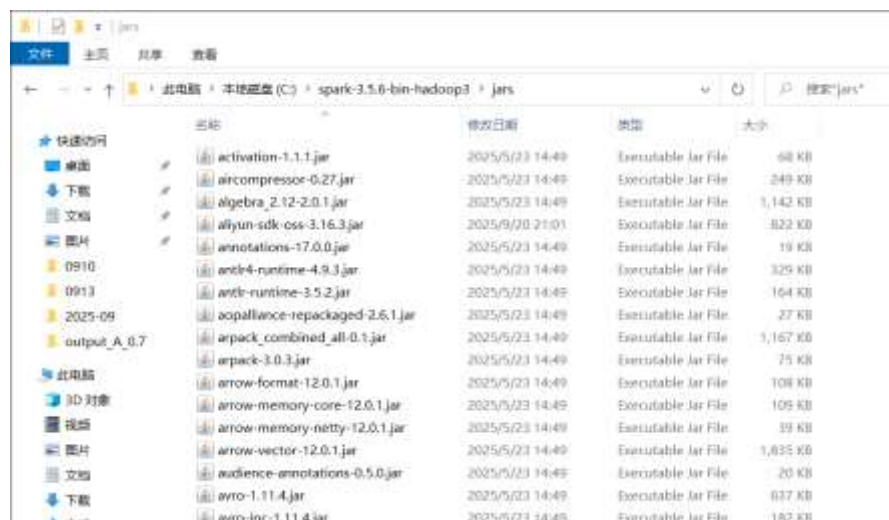
数据集

- 本次实验中所用数据已经预先放置到阿里云的OSS中
- 可以通过配套示例代码中给出accessKeyId和accessKeySecret来获取数据
- 但难点在于让Spark支持阿里云的OSS存储需要一点点操作。。。 （原生支持Amazon S3存储，其他兼容 S3 协议的对象存储，比如OSS则需要添加一些插件）

如何让Spark支持阿里云OSS存储?

- 需要下载三个插件到spark目录里的jars文件夹下

- hadoop-aliyun-3.3.6.jar (📎)
- aliyun-sdk-oss-3.16.3.jar (📎)
- jdome-2.0.6.jar (📎)



接下来运行示例程序

```
In [1]: import findspark
findspark.init() # notebook 里必须

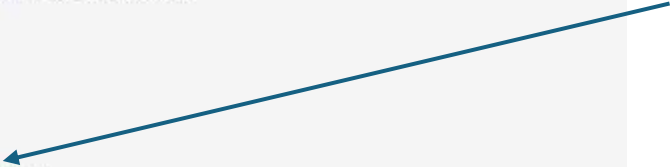
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, avg, count, desc
from pyspark.sql.types import IntegerType, FloatType, LongType
from pyspark.sql.functions import split, regexp_extract

# =====
# 启动 SparkSession, 配置 OSS
# =====
spark = SparkSession.builder \
    .appName("ML-ETL-EDA") \
    .master("spark://192.168.0.186:7077") \
    .config("fs.oss.impl", "org.apache.hadoop.fs.aliyun.oss.AliyunOSSFileSystem") \
    .config("fs.oss.accessKeyId", "LTAI5tKJCDxHBwyXrj5XVKQs") \
    .config("fs.oss.accessKeySecret", "nPwK7jBsRxxqq6xVWutU2DCI1RIZ55") \
    .config("fs.oss.endpoint", "oss-cn-chengdu.aliyuncs.com") \
    .getOrCreate()

In [2]: # =====
# OSS 上的 MovieLens 文件路径
# =====
movies_path = "oss://spark-experiments/MovieLens 1M Dataset/movies.dat"
ratings_path = "oss://spark-experiments/MovieLens 1M Dataset/ratings.dat"

In [3]: # =====
# 读取 movies.dat
# =====
raw_movies = spark.read.text(movies_path)
movies = raw_movies.withColumn("movieId", split("value", "::").getItem(0).cast(IntegerType()))
```

注意需要修改自己的
集群master地址



接下来运行示例程序

请完成以下两个题目

```
In [ ]: # =====  
# 最受欢迎（评分次数最多）的电影前 20;  
# =====
```

```
In [ ]: # =====  
# 平均评分最高且评分次数 $\geq 100$  的电影前 20;  
# =====
```