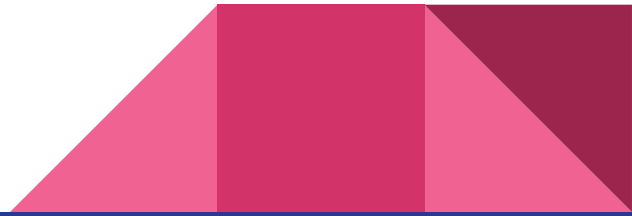
Movie Recommendation with MLlib - Collaborative Filtering (Implementation 3)

Shruti Kavishwar
San Francisco Bay University
Guided By: Prof. Henry Chang

Agenda

- Introduction
- Collaborative Filtering Overview
- What is ALS
- Dataset Setup
- Data Loading
- ALS model training
- Model Evaluation
- Hyperparameter Tunning
- Recommendations
- Steps
- Result



Introduction

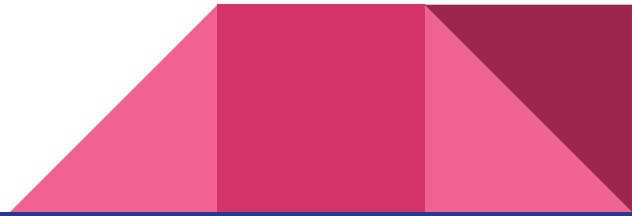
- Overview of Collaborative Filtering:
  - Technique used in recommendation systems.
- Importance of Recommendation Systems:
  - Enhances user experience by providing personalized suggestions.
- Introduction to ALS (Alternating Least Squares):
  - A matrix factorization technique to optimize collaborative filtering

Collaborative Filtering Overview

- Definition and Types:
  - User-User and Item-Item collaborative filtering.
- Focus on Matrix Factorization:
  - Decomposes the matrix of user-item interactions into lower-dimensional user and item matrices.
- Role of ALS in Collaborative Filtering:
  - Optimizes user and item latent factors iteratively to minimize the prediction error.

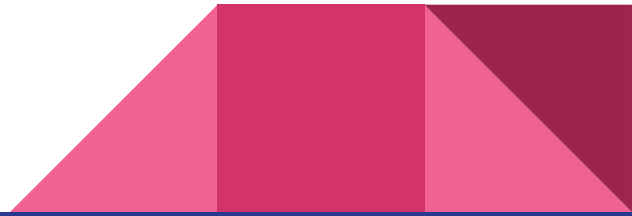
What is ALS?

- Matrix Factorization Technique:
  - Decomposes the user-item interaction matrix.
- Minimize Regularized Squared Error:
  - Reduces the prediction error while avoiding overfitting.
- Alternating Optimization:
  - Alternates between optimizing user and item matrices.



Dataset Details

- Description of the Dataset Used:
  - MovieLens dataset.
- Key Statistics:
  - Number of users, items, and ratings.
- Data Preprocessing Steps:
  - Cleaning and transforming data for analysis



PySpark Setup

- Import necessary Libraries and Initialize Spark session

```
import pandas as pd
from pyspark.sql.functions import col, explode
from pyspark import SparkContext
```

Initiate spark session

```
from pyspark.sql import SparkSession
sc = SparkContext
# sc.setCheckpointDir('checkpoint')
spark = SparkSession.builder.appName('Recommendations').getOrCreate()
```

Data Loading

```
movies = spark.read.csv("movies.csv",header=True)
ratings = spark.read.csv("ratings.csv",header=True)
```

```
ratings.show()
```

Build ALS Model Training

```
# Import the required functions
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
```

```
# Create test and train set
(train, test) = ratings.randomSplit([0.8, 0.2], seed = 1234)

# Create ALS model
als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating", nonnegative = True, implicitPrefs = Fa

# Confirm that a model called "als" was created
type(als)
```

Build ALS Model Training

```
# Import the requisite items
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

# Add hyperparameters and their respective values to param_grid
param_grid = ParamGridBuilder() \
            .addGrid(als.rank, [10, 50, 100, 150]) \
            .addGrid(als.regParam, [.01, .05, .1, .15]) \
            .build()
            # .addGrid(als.rank, [10, 50, 100, 150]) \
            # .addGrid(als.regParam, [.01, .05, .1, .15]) \
            #             .addGrid(als.maxIter, [5, 50, 100, 200]) \


# Define evaluator as RMSE and print length of evaluator
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")
print ("Num models to be tested: ", len(param_grid))
```

Model Evaluation and Hyperparameter Tuning

Tell Spark how to tune your ALS model

```
# Import the requisite items
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

# Add hyperparameters and their respective values to param_grid
param_grid = ParamGridBuilder() \
            .addGrid(als.rank, [10, 50, 100, 150]) \
            .addGrid(als.regParam, [.01, .05, .1, .15]) \
            .build()
            # .addGrid(als.rank, [10, 50, 100, 150]) \
            # .addGrid(als.regParam, [.01, .05, .1, .15]) \
            #             .addGrid(als.maxIter, [5, 50, 100, 200]) \


# Define evaluator as RMSE and print length of evaluator
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")
print ("Num models to be tested: ", len(param_grid))
```

```
Num models to be tested:  16
```

Build your cross validation pipeline

```
# Build cross validation using CrossValidator
cv = CrossValidator(estimator=als, estimatorParamMaps=param_grid, evaluator=evaluator, numFolds=5)

# Confirm cv was built
print(cv)
```

```
CrossValidator_d9751289a42d
```

Executing on GCP

- Create a Virtual Environment
- Install the pandas and Pyspark packages
- Upload all the .csv and the MLlib_Recommendation_system.py file downloaded from google collab.
- Execute The python code.

```
skavishw276@cloudshell:~/MLlibAssignment (mapreduce-week2-hw1-cs570)$ virtualenv venv
created virtual environment CPython3.10.12.final.0-64 in 648ms
  creator CPython3Posix(dest=/home/skavishw276/MLlibAssignment/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/skavishw276/.local/share/virtualenv)
    added seed packages: pip==24.1, setuptools==70.1.0, wheel==0.43.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
skavishw276@cloudshell:~/MLlibAssignment (mapreduce-week2-hw1-cs570)$ source venv/bin/activate
```

```
(venv) skavishw276@cloudshell:~/MLlibAssignment (mapreduce-week2-hw1-cs570)$ pip install pyspark pandas
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 3.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting pandas
  Downloading pandas-2.2.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Collecting py4j==0.10.9.7 (from pyspark)
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl.metadata (1.5 kB)
Collecting numpy>=1.22.4 (from pandas)
  Downloading numpy-2.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.9/60.9 kB 4.6 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 200.5/200.5 kB 16.1 MB/s eta 0:00:00
Downloading pandas-2.2.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.0/13.0 MB 72.5 MB/s eta 0:00:00
```

Results

```
(venv) skavishw276@cloudshell:~/MLlibAssignment (mapreduce-week2-hw1-cs570)$ python recommendation_engine_movielens.py
```

```
24/08/06 20:17:19 WARN NativeCodeLoa
+------+-------+------+----------+
|userId|movieId|rating|timestamp|
+------+-------+------+----------+
|     1|      1|   4.0|964982703|
|     1|      3|   4.0|964981247|
|     1|      6|   4.0|964982224|
|     1|     47|   5.0|964983815|
|     1|     50|   5.0|964982931|
|     1|     70|   3.0|964982400|
|     1|    101|   5.0|964980868|
|     1|    110|   4.0|964982176|
|     1|    151|   5.0|964984041|
|     1|    157|   5.0|964984100|
|     1|    163|   5.0|964983650|
|     1|    216|   5.0|964981208|
|     1|    223|   3.0|964980985|
|     1|    231|   5.0|964981179|
|     1|    235|   4.0|964980908|
|     1|    260|   5.0|964981680|
|     1|    296|   3.0|964982967|
|     1|    316|   3.0|964982310|
|     1|    333|   5.0|964981179|
|     1|    349|   4.0|964982563|
+------+-------+------+----------+
only showing top 20 rows
```

```
root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)

+------+-------+------+
|userId|movieId|rating|
+------+-------+------+
|     1|      1|   4.0|
|     1|      3|   4.0|
|     1|      6|   4.0|
|     1|     47|   5.0|
|     1|     50|   5.0|
|     1|     70|   3.0|
|     1|    101|   5.0|
|     1|    110|   4.0|
|     1|    151|   5.0|
|     1|    157|   5.0|
|     1|    163|   5.0|
|     1|    216|   5.0|
|     1|    223|   3.0|
|     1|    231|   5.0|
|     1|    235|   4.0|
|     1|    260|   5.0|
|     1|    296|   3.0|
|     1|    316|   3.0|
|     1|    333|   5.0|
|     1|    349|   4.0|
+------+-------+------+
only showing top 20 rows
```

```
The ratings dataframe is  98.30% empty.
+------+-----+
|userId|count|
+------+-----+
|   414| 2698|
|   599| 2478|
|   474| 2108|
|   448| 1864|
|   274| 1346|
|   610| 1302|
|    68| 1260|
|   380| 1218|
|   606| 1115|
|   288| 1055|
|   249| 1046|
|   387| 1027|
|   182|  977|
|   307|  975|
|   603|  943|
|   298|  939|
|   177|  904|
|   318|  879|
|   232|  862|
|   480|  836|
+------+-----+
only showing top 20 rows
```

```
+-------+-----+
|movieId|count|

+-------+-----+
|movieId|count|
+-------+-----+
|    356|  329|
|    318|  317|
|    296|  307|
|    593|  279|
|   2571|  278|
|    260|  251|
|    480|  238|
|    110|  237|
|    589|  224|
|    527|  220|
|   2959|  218|
|      1|  215|
|   1196|  211|
|     50|  204|
|   2858|  204|
|     47|  203|
|    780|  202|
|    150|  201|
|   1198|  200|
|   4993|  198|
+-------+-----+
only showing top 20 rows

Num models to be tested:  16
```

Result

```
**Best Model**
  Rank: 100
  MaxIter: 10
  RegParam: 0.15
0.8681593692522461
+------+-------+------+----------+
|userId|movieId|rating|prediction|
+------+-------+------+----------+
|   580|   1580|   4.0| 3.4899714|
|   580|  44022|   3.5| 3.2149775|
|   597|    471|   2.0|  4.191242|
|   108|   1959|   5.0| 3.8541899|
|   368|   2122|   2.0| 1.8182536|
|   436|    471|   3.0| 3.6141868|
|   587|   1580|   4.0| 3.8591955|
|    27|   1580|   3.0|  3.387604|
|   606|   1580|   2.5| 3.1743948|
|   606|  44022|   4.0| 2.8592076|
|    91|   2122|   4.0| 2.3004735|
|   157|   3175|   2.0| 3.4343777|
|   232|   1580|   3.5|  3.384147|
|   232|  44022|   3.0| 3.1172245|
|   246|   1645|   4.0| 3.8074667|
|   599|   2366|   3.0| 2.8804574|
|   111|   1088|   3.0|  3.280708|
|   111|   3175|   3.5| 3.1388845|
|    47|   1580|   1.5| 2.6884737|
|   140|   1580|   3.0| 3.3736506|
+------+-------+------+----------+
only showing top 20 rows
```

```
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|     1|[{3379, 5.7292676...|
|     2|[{131724, 4.80204...|
|     3|[{6835, 4.8498487...|
|     4|[{3851, 4.8558836...|
|     5|[{3379, 4.564525}...|
|     6|[{3925, 4.73028},...|
|     7|[{33649, 4.489432...|
|     8|[{3379, 4.648809}...|
|     9|[{3379, 4.803781}...|
|    10|[{71579, 4.539243...|
+------+--------------------+

+------+-------+---------+
|userId|movieId|   rating|
+------+-------+---------+
|     1|   3379|5.7292676|
|     1|  33649| 5.586225|
|     1|   5490| 5.482283|
|     1| 171495|5.3970823|
|     1|   5416|5.3507605|
|     1|   5328|5.3507605|
|     1|   3951|5.3507605|
|     1|  78836|5.3456903|
|     1|   5915|5.3334856|
|     1|   6460|5.2922673|
+------+-------+---------+
```

Result

movieId	userId	rating	title	genres
67618	100	5.073804	Strictly Sexual (...	Comedy|Drama|Romance
33649	100	5.013063	Saving Face (2004)	Comedy|Drama|Romance
3379	100	4.951482	On the Beach (1959)	Drama
42730	100	4.9439926	Glory Road (2006)	Drama
74282	100	4.9209943	Anne of Green Gab...	Children|Drama|Ro...
7121	100	4.8694086	Adam's Rib (1949)	Comedy|Romance
184245	100	4.8617606	De platte jungle ...	Documentary
179135	100	4.8617606	Blue Planet II (2...	Documentary
138966	100	4.8617606	Nasu: Summer in A...	Animation
117531	100	4.8617606	Watermark (2014)	Documentary

movieId	userId	rating	title	genres
1101	100	5.0	Top Gun (1986)	Action|Romance
1958	100	5.0	Terms of Endearme...	Comedy|Drama
2423	100	5.0	Christmas Vacatio...	Comedy
4041	100	5.0	Officer and a Gen...	Drama|Romance
5620	100	5.0	Sweet Home Alabam...	Comedy|Romance
368	100	4.5	Maverick (1994)	Adventure|Comedy|...
934	100	4.5	Father of the Bri...	Comedy
539	100	4.5	Sleepless in Seat...	Comedy|Drama|Romance
16	100	4.5	Casino (1995)	Crime|Drama
553	100	4.5	Tombstone (1993)	Action|Drama|Western

GitHub Link

https://github.com/ShrutiK02/Cloud-Computing/tree/61dadc22699dd6c734171e15849dfb116d436f75/Machine%20Learning/Movie%20Recommendation%20System