



**دانشگاه صنعتی امیرکبیر**  
( پلی تکنیک تهران )

دانشکده مهندسی کامپیوتر

تمرین امتیازی

درس رایانش ابری

هستی السادات جلالی چیمه-۹۸۳۱۰۱۸

هلیاسادات هاشمی پور-۹۸۳۱۱۰۶

استاد

دکتر جوادی

بهمن ۰۱

این کد یک سیستم فیلتر مشترک را با استفاده از رویکرد کاربر-کاربر و الگوریتم حداقل مربعات متناوب (ALS) راه اندازی می کند.

یک SparkSession ایجاد می کند که نقطه ورود به استفاده از عملکرد Spark است و نام برنامه را به عنوان "CollaborativeFiltering-v2" تنظیم میکنیم.

```
# Create a spark session
spark = SparkSession.builder.appName("CollaborativeFiltering-v2").getOrCreate()
```

در مجموعه داده رتبه بندی ها از یک فایل CSV با استفاده از روش spark.read.csv خوانده ، هدر را به عنوان ردیف اول تنظیم می کنیم و طرح را از داده ها استنتاج می کنیم.

```
# Read in the ratings dataset
ratings = spark.read.csv("./Dataset/ratings.csv", header=True, inferSchema=True)
```

این مجموعه داده را با استفاده از روش randomSplit به یک مجموعه آموزشی (۸۰٪) و یک مجموعه آزمایشی (۲۰٪) تقسیم میکند.

```
# Split the dataset into training and testing sets
(training, test) = ratings.randomSplit([0.8, 0.2])
```

این یک نمونه از الگوریتم ALS با پارامترهای مشخص شده برای حداکثر تعداد تکرار، پارامتر تنظیم، ستون کاربر، ستون آیتم و ستون رتبه بندی ایجاد می کند.

```
# Build the ALS model
als = ALS(maxIter=5, regParam=0.01, userCol="user_id", itemCol="game_id",
ratingCol="rating")
```

با استفاده از روش برازش، مدل را با داده های آموزشی برازش می دهد. با استفاده از روش تبدیل، پیش بینی هایی را روی داده های آزمایش ایجاد می کند.

```
# Fit the model to the training data
model = als.fit(training)
```

```
# Generate predictions on the test data
predictions = model.transform(test)
```

با استفاده از روش recommendForAllUsers به همه کاربران توصیه هایی میکند.

```
# Make recommendations for all users
userRecs = model.recommendForAllUsers(5)
```

توصیه هایی را برای یک کاربر خاص نشان می دهد و چارچوب داده userRecs را توسط user\_id فیلتر میکند.

```
# Show the recommendations for a specific user
userRecs.filter(col("user_id") == 1).show()
```

در مجموعه داده بازی ها خوانده شده و دو تابع تعریف میکنیم که نام بازی را بر اساس game\_id دریافت میکند و با فراخوانی تابع بالا، نام بازی را برای کاربر خاص ۵ توصیه می کند.

```
# get game name based on game id
def getGameName(game_id):
    return games.filter(col("game_id") == game_id).collect()[0].name
# get 5 recommendations game name for specific user
def getRecsForUser(user_id):
    recommendations = userRecs.filter(col("user_id") ==
user_id).collect()[0].recommendations
    for recommendation in recommendations:
        print(getGameName(recommendation.game_id))
```

این جلسه spark را متوقف میکنیم.

```
# Stop the spark session
spark.stop()
```

خروجی هم به شکل زیر شد:

```
1  getRecsForUser(10)
```

```
Crash 'N' Burn
The Franz Kafka Videogame
Super Meat Boy Forever
Zendoku
CALENDULA
```