





INSIDE THE “Movierecommendation.class” file…

*//  
// Source code recreated from a .class file by IntelliJ IDEA  
// (powered by Fernflower decompiler)  
//***package** com.sparkaws;  
  
**import** com.sparkaws.Movierecommendation.Rating;  
**import** com.sparkaws.Movierecommendation..anonfun.2;  
**import** com.sparkaws.Movierecommendation..anonfun.3;  
**import** com.sparkaws.Movierecommendation..anonfun.4;  
**import** com.sparkaws.Movierecommendation..anonfun.loadMovieNames.1;  
**import** com.sparkaws.Movierecommendation.Rating.;  
**import** java.nio.charset.CodingErrorAction;  
**import** org.apache.spark.ml.recommendation.ALS;  
**import** org.apache.spark.ml.recommendation.ALSModel;  
**import** org.apache.spark.rdd.RDD;  
**import** org.apache.spark.sql.Column;  
**import** org.apache.spark.sql.Dataset;  
**import** org.apache.spark.sql.SparkSession;  
**import** scala.collection.Iterator;  
**import** scala.collection.immutable.Map;  
**import** scala.collection.immutable.StringOps;  
**import** scala.io.Codec;  
**import** scala.reflect.ScalaSignature;  
**import** scala.reflect.api.JavaUniverse;  
**import** scala.reflect.api.Mirror;  
**import** scala.reflect.api.TypeTags;  
**import** scala.reflect.api.JavaUniverse.JavaMirror;  
**import** scala.runtime.BoxedUnit;  
**import** scala.runtime.BoxesRunTime;  
**import** scala.runtime.ObjectRef;  
  
@ScalaSignature(  
 bytes = **"\u0006\u0001\u0005\u0005e\u0001B\u0001\u0003\u0001\u001d\u00111#T8wS\u0016\u0014XmY8n[\u0016tG-\u0019;j\_:T!a\u0001\u0003\u0002\u0011M\u0004\u0018M]6boNT\u0011!B\u0001\u0004G>l7\u0001A\n\u0003\u0001!\u0001\"!\u0003\u0007\u000e\u0003)Q\u0011aC\u0001\u0006g\u000e\fG.Y\u0005\u0003\u001b)\u0011a!\u00118z%\u00164\u0007\"B\b\u0001\t\u0003\u0001\u0012A\u0002\u001fj]&$h\bF\u0001\u0012!\t\u0011\u0002!D\u0001\u0003\r\u0011!\u0002\u0001Q\u000b\u0003\rI\u000bG/\u001b8h\'\u0011\u0019\u0002BF\r\u0011\u0005%9\u0012B\u0001\r\u000b\u0005\u001d\u0001&o\u001c3vGR\u0004\"!\u0003\u000e\n\u0005mQ!\u0001D\*fe&\fG.\u001b>bE2,\u0007\u0002C\u000f\u0014\u0005+\u0007I\u0011\u0001\u0010\u0002\rU\u001cXM]%e+\u0005y\u0002CA\u0005!\u0013\t\t#BA\u0002J]RD\u0001bI\n\u0003\u0012\u0003\u0006IaH\u0001\bkN,\'/\u00133!\u0011!)3C!f\u0001\n\u0003q\u0012aB7pm&,\u0017\n\u001a\u0005\tOM\u0011\t\u0012)A\u0005?\u0005AQn\u001c<jK&#\u0007\u0005\u0003\u0005\*\'\tU\r\u0011\"\u0001+\u0003\u0019\u0011\u0018\r^5oOV\t1\u0006\u0005\u0002\nY%\u0011QF\u0003\u0002\u0006\r2|\u0017\r\u001e\u0005\t\_M\u0011\t\u0012)A\u0005W\u00059!/\u0019;j]\u001e\u0004\u0003\"B\b\u0014\t\u0003\tD\u0003\u0002\u001a5kY\u0002\"aM\n\u000e\u0003\u0001AQ!\b\u0019A\u0002}AQ!\n\u0019A\u0002}AQ!\u000b\u0019A\u0002-Bq\u0001O\n\u0002\u0002\u0013\u0005\u0011(\u0001\u0003d\_BLH\u0003\u0002\u001a;wqBq!H\u001c\u0011\u0002\u0003\u0007q\u0004C\u0004&oA\u0005\t\u0019A\u0010\t\u000f%:\u0004\u0013!a\u0001W!9ahEI\u0001\n\u0003y\u0014AD2paf$C-\u001a4bk2$H%M\u000b\u0002\u0001\*\u0012q$Q\u0016\u0002\u0005B\u00111\tS\u0007\u0002\t\*\u0011QIR\u0001\nk:\u001c\u0007.Z2lK\u0012T!a\u0012\u0006\u0002\u0015\u0005tgn\u001c;bi&|g.\u0003\u0002J\t\n\tRO\\2iK\u000e\\W\r\u001a,be&\fgnY3\t\u000f-\u001b\u0012\u0013!C\u0001\u007f\u0005q1m\u001c9zI\u0011,g-Y;mi\u0012\u0012\u0004bB\'\u0014#\u0003%\tAT\u0001\u000fG>\u0004\u0018\u0010\n3fM\u0006,H\u000e\u001e\u00134+\u0005y%FA\u0016B\u0011\u001d\t6#!A\u0005BI\u000bQ\u0002\u001d:pIV\u001cG\u000f\u0015:fM&DX#A\*\u0011\u0005QKV\"A+\u000b\u0005Y;\u0016\u0001\u00027b]\u001eT\u0011\u0001W\u0001\u0005U\u00064\u0018-\u0003\u0002[+\n11\u000b\u001e:j]\u001eDq\u0001X\n\u0002\u0002\u0013\u0005a$\u0001\u0007qe>$Wo\u0019;Be&$\u0018\u0010C\u0004\_\'\u0005\u0005I\u0011A0\u0002\u001dA\u0014x\u000eZ;di\u0016cW-\\3oiR\u0011\u0001m\u0019\t\u0003\u0013\u0005L!A\u0019\u0006\u0003\u0007\u0005s\u0017\u0010C\u0004e;\u0006\u0005\t\u0019A\u0010\u0002\u0007a$\u0013\u0007C\u0004g\'\u0005\u0005I\u0011I4\u0002\u001fA\u0014x\u000eZ;di&#XM]1u\_J,\u0012\u0001\u001b\t\u0004S2\u0004W\"\u00016\u000b\u0005-T\u0011AC2pY2,7\r^5p]&\u0011QN\u001b\u0002\t\u0013R,\'/\u0019;pe\"9qnEA\u0001\n\u0003\u0001\u0018\u0001C2b]\u0016\u000bX/\u00197\u0015\u0005E$\bCA\u0005s\u0013\t\u0019(BA\u0004C\_>dW-\u00198\t\u000f\u0011t\u0017\u0011!a\u0001A\"9aoEA\u0001\n\u0003:\u0018\u0001\u00035bg\"\u001cu\u000eZ3\u0015\u0003}Aq!\_\n\u0002\u0002\u0013\u0005#0\u0001\u0005u\_N#(/\u001b8h)\u0005\u0019\u0006b\u0002?\u0014\u0003\u0003%\t%`\u0001\u0007KF,\u0018\r\\:\u0015\u0005Et\bb\u00023|\u0003\u0003\u0005\r\u0001Y\u0004\n\u0003\u0003\u0001\u0011\u0011!E\u0001\u0003\u0007\taAU1uS:<\u0007cA\u001a\u0002\u0006\u0019AA\u0003AA\u0001\u0012\u0003\t9aE\u0003\u0002\u0006\u0005%\u0011\u0004\u0005\u0005\u0002\f\u0005EqdH\u00163\u001b\t\tiAC\u0002\u0002\u0010)\tqA];oi&lW-\u0003\u0003\u0002\u0014\u00055!!E!cgR\u0014\u0018m\u0019;Gk:\u001cG/[8og!9q\"!\u0002\u0005\u0002\u0005]ACAA\u0002\u0011!I\u0018QAA\u0001\n\u000bR\bBCA\u000f\u0003\u000b\t\t\u0011\"!\u0002 \u0005)\u0011\r\u001d9msR9!\'!\t\u0002$\u0005\u0015\u0002BB\u000f\u0002\u001c\u0001\u0007q\u0004\u0003\u0004&\u00037\u0001\ra\b\u0005\u0007S\u0005m\u0001\u0019A\u0016\t\u0015\u0005%\u0012QAA\u0001\n\u0003\u000bY#A\u0004v]\u0006\u0004\b\u000f\\=\u0015\t\u00055\u0012\u0011\b\t\u0006\u0013\u0005=\u00121G\u0005\u0004\u0003cQ!AB(qi&|g\u000e\u0005\u0004\n\u0003kyrdK\u0005\u0004\u0003oQ!A\u0002+va2,7\u0007C\u0005\u0002<\u0005\u001d\u0012\u0011!a\u0001e\u0005\u0019\u0001\u0010\n\u0019\t\u0015\u0005}\u0012QAA\u0001\n\u0013\t\t%A\u0006sK\u0006$\'+Z:pYZ,GCAA\"!\r!\u0016QI\u0005\u0004\u0003\u000f\*&AB(cU\u0016\u001cG\u000fC\u0004\u0002L\u0001!\t!!\u0014\u0002\u0017A\f\'o]3SCRLgn\u001a\u000b\u0004e\u0005=\u0003\u0002CA)\u0003\u0013\u0002\r!a\u0015\u0002\u0007M$(\u000f\u0005\u0003\u0002V\u0005mcbA\u0005\u0002X%\u0019\u0011\u0011\f\u0006\u0002\rA\u0013X\rZ3g\u0013\rQ\u0016Q\f\u0006\u0004\u00033R\u0001bBA1\u0001\u0011\u0005\u00111M\u0001\u000fY>\fG-T8wS\u0016t\u0015-\\3t)\t\t)\u0007E\u0004\u0002V\u0005\u001dt$a\u0015\n\t\u0005%\u0014Q\f\u0002\u0004\u001b\u0006\u0004\bbBA7\u0001\u0011\u0005\u0011qN\u0001\u0005[\u0006Lg\u000e\u0006\u0003\u0002r\u0005]\u0004cA\u0005\u0002t%\u0019\u0011Q\u000f\u0006\u0003\tUs\u0017\u000e\u001e\u0005\t\u0003s\nY\u00071\u0001\u0002|\u0005!\u0011M]4t!\u0015I\u0011QPA\*\u0013\r\tyH\u0003\u0002\u0006\u0003J\u0014\u0018-\u001f"**)  
**public class** Movierecommendation {  
 **private volatile** Rating$module;  
  
 **private** Rating$lzycompute() {  
 BoxedUnit var10000;  
 **synchronized**(**this**) {  
 **if**(**this**.Rating$module == **null**) {  
 **this**.Rating$module = **new** (**this**);  
 }  
  
 var10000 = BoxedUnit.UNIT;  
 }  
  
 var10000 = **null**;  
 **return this**.Rating$module;  
 }  
  
 **public** Rating() {  
 **return this**.Rating$module == **null**?**this**.Rating$lzycompute():**this**.Rating$module;  
 }  
  
 **public** Rating parseRating(String str) {  
 String[] fields = str.split(**"::"**);  
 scala.Predef..MODULE$.**assert**(scala.Predef..MODULE$.refArrayOps((Object[])fields).size() == 4);  
 **return new** Rating(**this**, (**new** StringOps(scala.Predef..MODULE$.augmentString(fields[0]))).toInt(), (**new** StringOps(scala.Predef..MODULE$.augmentString(fields[1]))).toInt(), (**new** StringOps(scala.Predef..MODULE$.augmentString(fields[2]))).toFloat());  
 }  
  
 **public** Map<Object, String> loadMovieNames() {  
 Codec codec = scala.io.Codec..MODULE$.apply(**"UTF-8"**);  
 codec.onMalformedInput(CodingErrorAction.REPLACE);  
 codec.onUnmappableCharacter(CodingErrorAction.REPLACE);  
 ObjectRef movieNames = ObjectRef.create((Map)scala.Predef..MODULE$.Map().apply(scala.collection.immutable.Nil..MODULE$));  
 Iterator lines = scala.io.Source..MODULE$.fromFile(**"/home/hadoop/movies.dat"**, codec).getLines();  
 lines.foreach(**new** 1(**this**, movieNames));  
 **return** (Map)movieNames.elem;  
 }  
  
 **public void** main(String[] args) {  
 SparkSession spark = org.apache.spark.sql.SparkSession..MODULE$.builder().appName(**"ALSExample"**).getOrCreate();  
 spark.sparkContext().setLogLevel(**"ERROR"**);  
 scala.Predef..MODULE$.println(**"Loading movie names..."**);  
 Map nameDict = **this**.loadMovieNames();  
 Dataset var10000 = spark.read().textFile(**"s3://movierecommendation/ml-1m/ratings.dat"**);  
 com.sparkaws.Movierecommendation..anonfun.1 var10001 = **new** com.sparkaws.Movierecommendation..anonfun.1(**this**);  
 org.apache.spark.sql.SparkSession.implicits. var10002 = spark.implicits();  
 JavaUniverse $u = scala.reflect.runtime.**package**..MODULE$.universe();  
 JavaMirror $m = scala.reflect.runtime.**package**..MODULE$.universe().runtimeMirror(Movierecommendation.**class**.getClassLoader());  
 Dataset ratings = var10000.map(var10001, var10002.newProductEncoder(((TypeTags)$u).TypeTag().apply((Mirror)$m, **new** com.sparkaws.Movierecommendation..typecreator1.1(**this**))));  
 Dataset ratingCounts = ratings.groupBy(**"movieId"**, scala.Predef..MODULE$.wrapRefArray((Object[])(**new** String[0]))).count();  
 ALS als = (**new** ALS()).setRank(8).setMaxIter(10).setRegParam(0.1D).setSeed(1234L).setUserCol(**"userId"**).setItemCol(**"movieId"**).setRatingCol(**"rating"**);  
 Rating[] newUserRatings = (Rating[])((Object[])(**new** Rating[]{**new** Rating(**this**, 0, 260, 5.0F), **new** Rating(**this**, 0, 329, 5.0F), **new** Rating(**this**, 0, 1356, 4.0F), **new** Rating(**this**, 0, 904, 5.0F), **new** Rating(**this**, 0, 908, 4.0F), **new** Rating(**this**, 0, 2657, 1.0F)}));  
 org.apache.spark.sql.SparkSession.implicits. var23 = spark.implicits();  
 RDD var24 = spark.sparkContext().parallelize(scala.Predef..MODULE$.wrapRefArray((Object[])newUserRatings), spark.sparkContext().parallelize$default$2(), scala.reflect.ClassTag..MODULE$.apply(Rating.**class**));  
 var10002 = spark.implicits();  
 JavaUniverse $u1 = scala.reflect.runtime.**package**..MODULE$.universe();  
 JavaMirror $m1 = scala.reflect.runtime.**package**..MODULE$.universe().runtimeMirror(Movierecommendation.**class**.getClassLoader());  
 Dataset newUserRatingsDS = var23.rddToDatasetHolder(var24, var10002.newProductEncoder(((TypeTags)$u1).TypeTag().apply((Mirror)$m1, **new** com.sparkaws.Movierecommendation..typecreator3.1(**this**)))).toDS();  
 Dataset allRatings = ratings.union(newUserRatingsDS);  
 ALSModel model = als.fit(allRatings);  
 **int**[] moviesIveSeen = (**int**[])scala.Predef..MODULE$.refArrayOps((Object[])newUserRatings).map(**new** 2(**this**), scala.Array..MODULE$.canBuildFrom(scala.reflect.ClassTag..MODULE$.Int()));  
 Dataset unratedMovies = ratings.filter(**new** 3(**this**, moviesIveSeen));  
 4 var25 = **new** 4(**this**);  
 var10002 = spark.implicits();  
 JavaUniverse $u2 = scala.reflect.runtime.**package**..MODULE$.universe();  
 JavaMirror $m2 = scala.reflect.runtime.**package**..MODULE$.universe().runtimeMirror(Movierecommendation.**class**.getClassLoader());  
 Dataset myUnratedMovies = unratedMovies.map(var25, var10002.newProductEncoder(((TypeTags)$u2).TypeTag().apply((Mirror)$m2, **new** com.sparkaws.Movierecommendation..typecreator4.1(**this**)))).distinct();  
 Dataset myUnratedMoviesWithCounts = myUnratedMovies.join(ratingCounts, **"movieId"**);  
 Dataset myPopularUnratedMovies = myUnratedMoviesWithCounts.filter(myUnratedMoviesWithCounts.apply(**"count"**).$greater(BoxesRunTime.boxToInteger(25)));  
 Dataset predictions = model.transform(myPopularUnratedMovies);  
 scala.Predef..MODULE$.println(**"\nRatings for fictitious user or user ID 0:"**);  
 scala.Predef..MODULE$.refArrayOps((Object[])newUserRatings).foreach(**new** com.sparkaws.Movierecommendation..anonfun.main.1(**this**, nameDict));  
 scala.Predef..MODULE$.println(**"\nTop 10 recommended movies:"**);  
 scala.Predef..MODULE$.refArrayOps((Object[])predictions.orderBy(scala.Predef..MODULE$.wrapRefArray((Object[])(**new** Column[]{org.apache.spark.sql.functions..MODULE$.desc(**"prediction"**)}))).take(10)).foreach(**new** com.sparkaws.Movierecommendation..anonfun.main.2(**this**, nameDict));  
 spark.stop();  
 }  
  
 **public** Movierecommendation() {  
 }  
}