import time

start = time.time()

from pyspark.mllib.recommendation import ALS, MatrixFactorizationModel, Rating

# Load and parse the data

data = sc.textFile("../Desktop/Dump/sample\_movielens\_ratings.txt")

ratings = data.map(lambda l: l.split(','))\

.map(lambda l: Rating(int(l[0]), int(l[1]), float(l[2])))

# Build the recommendation model using Alternating Least Squares

rank = 10

numIterations = 10

model = ALS.train(ratings, rank, numIterations)

# Evaluate the model on training data

testdata = ratings.map(lambda p: (p[0], p[1]))

predictions = model.predictAll(testdata).map(lambda r: ((r[0], r[1]), r[2]))

ratesAndPreds = ratings.map(lambda r: ((r[0], r[1]), r[2])).join(predictions)

MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])\*\*2).mean()

print("Mean Squared Error = " + str(MSE))

# Save and load model

model.save(sc, "target/tmp/myCollaborativeFilter")

sameModel = MatrixFactorizationModel.load(sc, "target/tmp/myCollaborativeFilter")

print(time.time() - start)