Recommender System

method load_movie_names()

    read movies.csv file and store the movie names into a global dictionary

    where movie_id is key and movie_name is the value

method map(rating_pairs)

    actual <- []

    prediction <- []

    for all rating_pair in rating_pairs **do**

        actual_rating, predicted_rating <- split(rating_pair)

        actual.append(actual_rating)

        prediction.append(predicted_rating)

    actual_ranking <- generate rankings of the list of actual ratings

    predicted_ranking <- generate rankings of the list of predicted ratings

    #return only first 10 entries in both lists

    return actual_ranking[:10], predicted_ranking[:10]

method main()

    read text file as RDD

    RDD.map(split row into user_id, movie_id, actualRating)

    Shuffle RDD

    #Performing 5-fold Cross Validation

    Total_RMSE = 0

    Total_MSE = 0

    Total_MAP = 0

    Create 5 folds of RDD

    repeat 5 times for different sets of folds

train a Model with certain set of parameters on 4 folds of RDD(trainRDD)

predictedRDD <- predict for remaining fold (testRDD) using the trained model

ratesAndPreds <- testRDD.join(predictedRDD)

metrics <- generate RegressionMetrics of ratesAndPreds

MeanSquareError <- metrics.meanSquaredError

Total_MSE <- Total_MSE + MeanSquareError

RootMeanSquareError <- metrics.rootMeanSquaredError

Total_RMSE <- Total_RMSE + RootMeanSquareError

ratingPairs <- ratesAndPreds.reduceByKey(merge all rating pairs into a list)

rankAndPreds <- ratingPairs.map(map)

convert this PipeLinedRDD into RDD

ranking_metrics <- generate RankingMetrics of rankAndPreds

MeanAveragePrecison <- ranking_metrics.meanAveragePrecision

Average_RMSE <- Total_RMSE/5

Average_MSE <- Total_MSE/5

Average_MAP <- Total_MAP/5

Print(Average_RMSE)

Print(Average_MSE)

Print(Average_MAP)

#Adding new user to database and giving recommendations to new user

new_user_ID <- 0 #since all userID's start from 1

newUserRDD <- get new_user_ratings from user

combinedRDD <- RDD.union(newUserRDD)

Train a new model with the parameters found from Cross Validation performed above

newUserTestRDD <- RDD.filter(all movies except the ones given by new user)

newUserTestRDD.map(new_user_ID, movieID)

newUserRecommendRDD <- Use new model to predict ratings for movies in newUserTestRDD

movie_name_dictionary = load_movie_names()

newUserRecommendRDD.filter(rating > 4)

newUserRecommendRDD.map(get_movie_name)

Print(newUserRecommendRDD.take(5)) #display any 5 movies with good rating