

计算欧氏距离分数

- `import json`
- `import numpy as np`
- `# 计算user1和user2的欧氏距离分数``def euclidean_score(dataset, user1, user2):`
- `if user1 not in dataset:`
- `raise TypeError('User ' + user1 + ' not present in the dataset')`
- `if user2 not in dataset:`
- `raise TypeError('User ' + user2 + ' not present in the dataset')`
- `# 提取两个用户均评过分的电影 # 如果没有两个用户共同评过分的电影, 则说明这两个用户之间没有相似度 (至少根据数据库中的评分信息无法计算出来 rated_by_both = {}`
- `for item in dataset[user1]:`
- `if item in dataset[user2]:`
- `rated_by_both[item] = 1 # 如果两个用户都没评分过, 得分为0 if len(rated_by_both) == 0:`
- `return 0 #对于每个共同评分, 只计算平方和的平方根, 并将该值归一化, 使得评分取值在0到1之间 ''' 如果评分相似, 那么平方和的差别就会很小, 因此评分就会变得很高, 这也是我们希望指标达到的效果。 ''' squared_differences = []`
- `for item in dataset[user1]:`
- `if item in dataset[user2]:`
- `squared_differences.append(np.square(dataset[user1][item] - dataset[user2][item]))`
- `return 1 / (1 + np.sqrt(np.sum(squared_differences)))`
- `if __name__ == '__main__':`
- `#加载数据文件中的movie_ratings.json文件 data_file = 'movie_ratings.json' with open(data_file, 'r') as f:`
- `data = json.loads(f.read())`
- `# 假定两个随机用户, 计算其欧氏距离分数 user1 = 'John Carson' user2 = 'Michelle Peterson' print ("\nEuclidean score:")`
- `print (euclidean_score(data, user1, user2))`