

## 生成电影推荐

---

- `import json`
- `import numpy as np`
- `from Chapter05.euclidean_score import euclidean_score`
- `from Chapter05.pearson_score import pearson_score`
- `from Chapter05.find_similar_users import find_similar_users`
- `''' 功能作用：计算欧氏距离分数 构建推荐引擎中一个非常重要的任务是寻找相似的用户，也就是说，为某位用户生成的推荐信息可以同时推荐给与其相似的用户。接下来学习如何寻找相似用户。 '''`定义一个为给定用户生成电影推荐的函数。首先检查该用户是否存在于数据库中定义一个为给定用户生成电影推荐的函数。首先检查该用户是否存在于数据库中`'''# 为给定用户生成电影推荐def generate_recommendations(dataset, user):`
- `# 判断用户是否存在 if user not in dataset:`
- `raise TypeError('User ' + user + ' not present in the dataset')`
- `# 计算该用户与数据库中其他用户的皮尔逊相关系数 total_scores = {}`
- `similarity_sums = {}`
- `# # 计算所有用户的皮尔逊相关度 for u in [x for x in dataset if x != user]:`
- `similarity_score = pearson_score(dataset, user, u)`
- `if similarity_score <= 0:`
- `continue # 找到还未被该用户评分的电影 for item in [x for x in dataset[u] if x not in dataset[user] or dataset[user][x] == 0]:`
- `total_scores.update({item: dataset[u][item] * similarity_score})`
- `similarity_sums.update({item: similarity_score})`
- `# 如果该用户看过数据库中所有的电影，那就不能为用户推荐电影 if len(total_scores) == 0:`
- `return ['没有更多的电影推荐']`
- `# 生成一个电影评分标准化列表 movie_ranks = np.array([[total/similarity_sums[item], item]`
- `for item, total in total_scores.items()])`
- `# 根据第一列对皮尔逊相关系数进行降序排列 movie_ranks = movie_ranks[np.argsort(movie_ranks[:, 0])[:-1]]`
- `# 提取出推荐的电影 recommendations = [movie for _, movie in movie_ranks]`
- `return recommendations`
- `if __name__ == '__main__':`
- `data_file = 'movie_ratings.json' # 读取json文件 with open(data_file, 'r') as f:`
- `data = json.loads(f.read())`
- `# 为Michael Henry生成推荐: user = 'Michael Henry' print`
- `("\\nRecommendations for " + user + ":")`
- `movies = generate_recommendations(data, user)`
- `for i, movie in enumerate(movies):`
- `print (str(i+1) + '. ' + movie)`

- # 用户John Carson看过所有电影，因此在他推荐电影时，应该显示0推荐。     user = 'John Carson'     print ("\nRecommendations for " + user + ":")
- movies = generate\_recommendations(data, user)
- for i, movie in enumerate(movies):
- print (str(i+1) + '. ' + movie)