

函数说明

	Description	Input	Expected output
1. add_UserSongList(List, datadir)	Add new songList	add_UserSongList (List, "D://DD/song_example")	{'success': True, 'error': ""}
2. update_SystemList(file, datadir)	Update SystemSongList	update_SystemList(1.xml,"C://")	{'success': True, 'error': ""}
3. Get_Song(ListName)	Get the Songs of one SongList	Get_Song(music)	{'success': True, "Songs are""", }
4. Add_newSong(Song)	Add new songs	Add_newSongs(Song)	{'success': True, 'error': ""}

1.readRatingDate(path='')
读取评分数据并存储为 csv 文件
param path:文件路径
return: DataFrame

2.calcuteSimilar(series1,series2):
计算余弦相似度
param data1: 数据集 1 Series
param data2: 数据集 2 Series
return: 相似度

3.calcuteUser(csvpath, targetID=1, TopN=10):
计算 targetID 的用户与其他用户的相似度
return:相似度 TopN Series

4.calcuteInterest(frame, similarSeries, targetItemID):

，，，

计算目标用户对目标物品的感兴趣程度

:param frame: 数据

:param similarSeries: 目标用户最相似的 K 个用户

:param targetItemID: 目标物品

:return: 感兴趣程度

(u 代表用户, i 代表商品, $S(u, k)$ 包含和用户兴趣最接近的 K 个用户, $N(i)$ 是对物品 i 有过行为的用户集合, w 代表用户 u 和用户 v 的相似程度, r 代表用户 v 对物品 i 的兴趣。)