|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 推荐引擎- recommendation\_engine | | |
| 接口实现 | 基于用户的协同过滤 – user\_CF | | |
| 请求参数: userID | | | |
| 数据集合（MovieLens） | | 备注 | 是否必须 |
| userID | | 用户ID | 是 |
| itemID | | 物品ID | 是 |
| Score | | 用户给物品的评分 | 是 |
| 返回值 | | | |
| Success： dict() 集合 | | | |
| Error： None | | | |
| 代码实现（中间有部分实现省略） | | | |
| **class** **user\_CF(recommendation\_engine):**  **def** \_\_init**(**self**,** train**,** test**){**  self**.**train **=** train  self**.**test **=** test  **}**  **def** user\_similarity**(**self**):**  """  计算用户之间相似度  return： W集合  """  U **=** dict**()** #用户之间有交集的集合  Num **=** dict**()** #和用户有关系的物品number  # 计算U 和 Num 的代码省略  # 计算用户之间相似度  self**.**W **=** dict**()**  **for** user**,**friend **in** U**.**items**():**  self**.**W**[**user**]** **=** **{}**  **for** v**,**count **in** friend**.**items**():**  self**.**W**[**user**][**v**]** **=** count **/** math**.**sqrt**(**Num**[**user**]** **\*** Num**[**v**])**  **return** self**.**W  #为用户推荐两个相关物品  **def** recommendation**(**self**,**user**):**  rank **=** dict**()**  relvent\_item **=** self**.**train**[**user**].**keys**()**  **for** v**,**similar\_value **in** self**.**W**[**user**].**items**():**  **for** i**,**rvi **in** self**.**train**[**v**].**items**():**  **if** i **in** relvent\_item**:**  **continue**  rank**.**setdefault**(**i**,**0**)**  rank**[**i**]** **+=** similar\_value **\*** rvi  **return** dict**(**rank**.**items**())** | | | |