# 1、系统概述

**1.1、系统简介**

简要概述系统的基本情况和背景。

基本情况：电影推荐系统是一种基于用户兴趣和行为数据的应用，旨在为用户提供个性化的电影推荐。它利用用户的历史行为、偏好和其他相关信息来预测用户可能感兴趣的电影，并向其推荐相关内容，从而提高用户的满意度和体验。在本项目中，我们基于网上已有的电影数据集进行推荐算法的设计与开发，通过划分训练集、测试集和验证集，训练出合适的推荐模型并提供反馈指标以监测其性能，再加上前端的页面展示使得这个系统更有应用性，更加完整。

背景：电影推荐系统的发展得益于互联网的普及和电影产业的快速发展。随着在线视频平台的兴起，用户面临着大量的电影选择，而如何从众多的电影中找到自己感兴趣的内容成为了一个挑战。传统的电影推荐方式，如排行榜、分类标签等，无法满足用户个性化的需求。因此，电影推荐系统的出现填补了这一空白，为用户提供了个性化的电影推荐服务。

**1.2、术语表**

定义系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息。

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | ItemCF | 基于物品的协同过滤算法 |
| 2 | Movielens | MoveieLens数据集是推荐系统领域最为经典的数据集之一，里面包含了多个电影评分数据集 |
| 3 | topN | 给用户推荐前N个其最有可能喜欢的商品 |

**1.3、系统运行环境**

包括对硬件平台、操作系统、数据库系统、编程平台、网络协议等的描述。

数据库：MySQL

编程平台：Spyder、Jupyter Notebook（方便以代码块进行调试）、PyCharm

网络协议：HTTPS、URL

**1.4、开发环境**

列举进行系统分析、程序设计和程序开发时要使用的工程工具和开发语言。应描述每一工具软件的名称、版本等。

数据库：MySQL 5.5 Command Line Client

编程平台：Spyder 4.0.1（python 3.7）、Jupyter Notebook 6.0.3、PyCharm

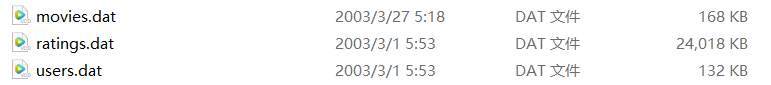
# 2、数据结构说明

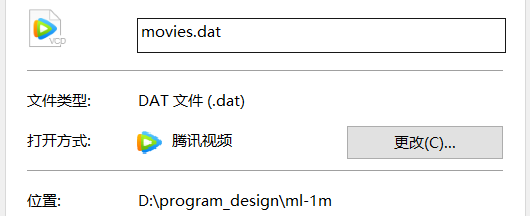
本章说明本程序系统中使用的全局数据常量、变量和数据结构。

2.1、常量

包括数据文件名称及其所在目录，功能说明，具体常量说明等。

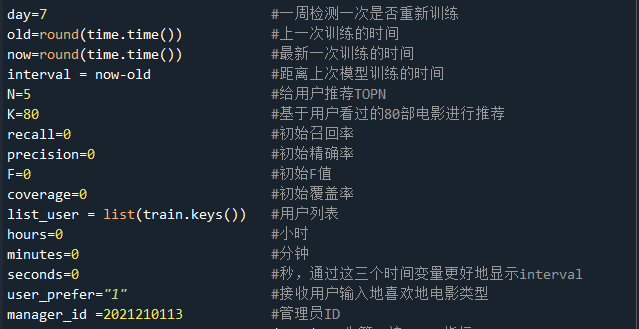
数据文件来自MovieLens官方数据集





2.2、变量

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

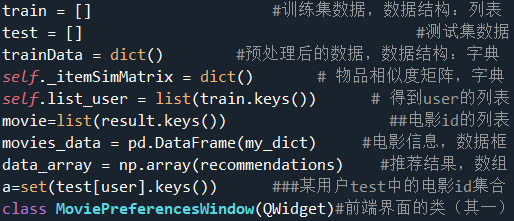


2.3、数据结构

包括数据结构名称，功能说明，具体数据结构说明（定义、注释、取值）等。

本程序中使用了多种数据结构，有python原生数据结构列表、字典、集合

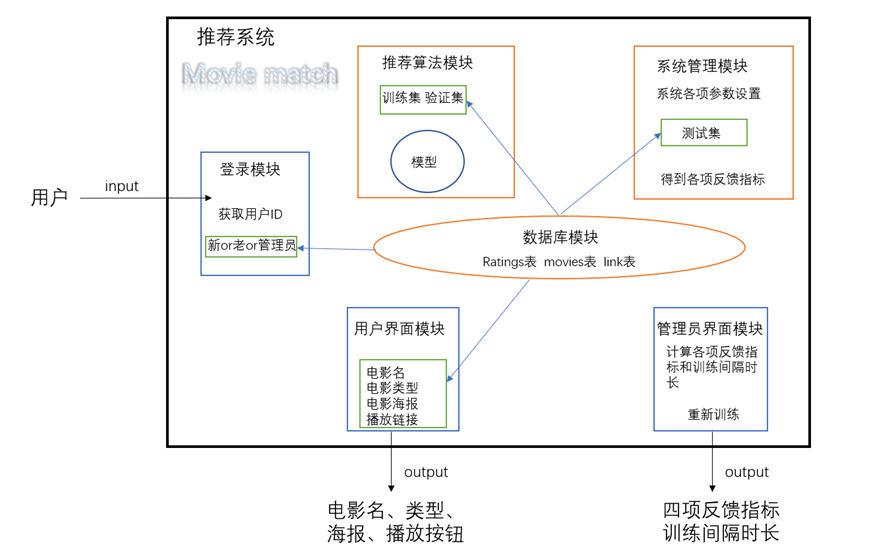
也有Numpy包中的数据结构数组和Pandas中的数据结构数据框，具体如下：



# 3、模块设计

**3.1、软件结构**

以图形方式给出软件系统的子系统（或软件包）划分，模块划分，子系统间、模块间关系等，并用接口来描述各模块之间的调用关系，给出各模块之间的松散耦合关系。



**3.2、功能设计说明**

结合上图阐述软件的基本设计思想和理念。

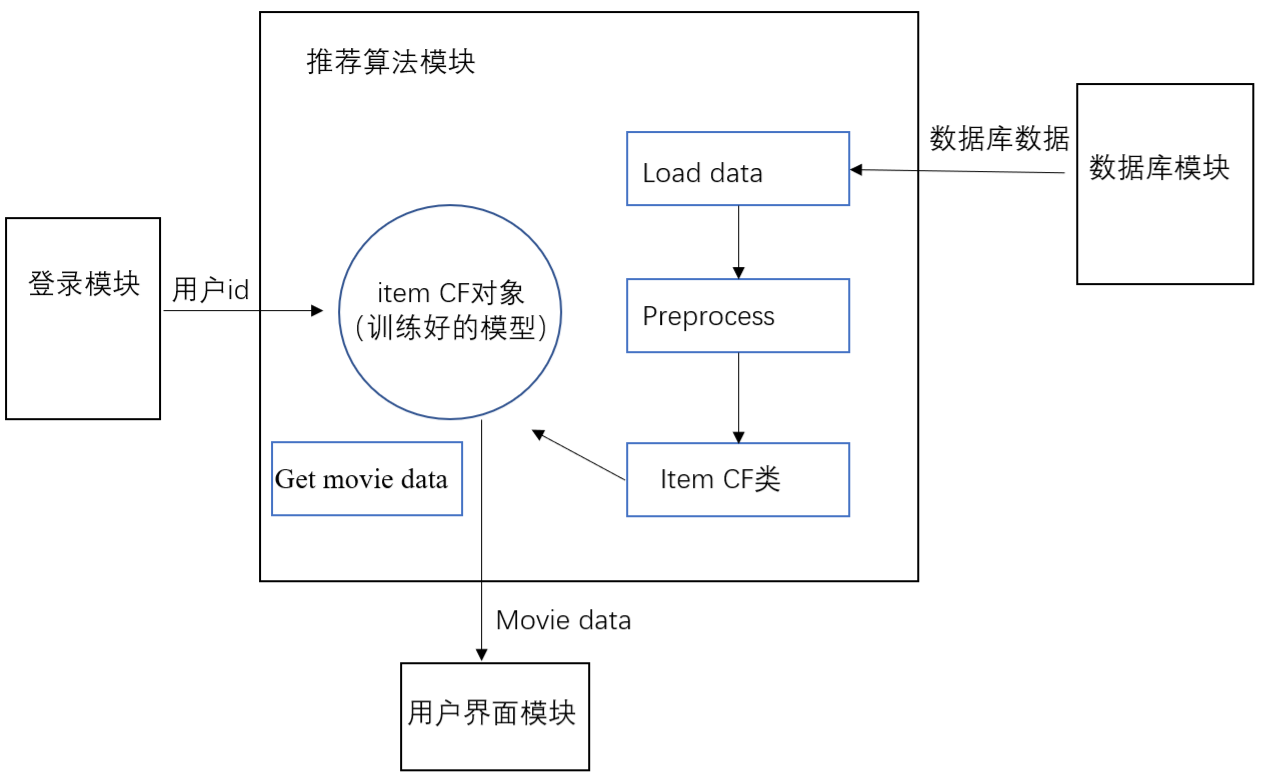
设计思想：作为一个完整的推荐系统，应该有前端、后端、数据库模块等，我们基于此进行设计，基于数据库中的数据，我们训练出了一个推荐模型，可以根据用户输入推荐出合适的电影，考虑到反馈的问题，我们设计了系统管理模块，根据用户的反馈，实时检测推荐系统的各项指标，进而判断何时需要重新训练、调整模型。从用户的角度出发，我们基于上述设计进行前端页面的开发，具体由登录模块、用户界面模块和管理员界面模块组成。

设计理念：尽可能简洁、方便，希望通过我们的推荐系统，可以更有效地连接用户与电影内容，从而节约时间和成本。

**3.3、推荐算法模块**

详细描述各功能模块的功能、数据结构、具体算法和流程等。

3.3.1、设计图



3.3.2、功能描述

简要描述模块1的业务功能。

该模块是推荐系统的核心模块，当传入用户id后，该模块能够将涵盖电影名、类型、海报、播放链接的moviedata传入用户界面模块。

3.3.3、输入数据

详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

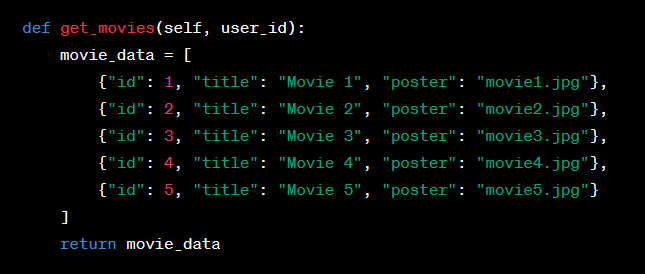
一是从登录模块获取用户id,用户id的有效性在登录模块中已经进行检验，确保传入推荐算法的id是有效的。

二是从数据库模块ratings表中获取系统中的记录，存入数据库时已经将数据处理成理想形式，确保输入数据的有效性。

3.3.4、输出数据

详细描述模块1所产生的数据以及这些数据的表现形式。

输出moviedata列表给用户界面模块,数据类型大致如下：



3.3.5、数据设计

给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）

字典类型：train和test 其关键字是userid,值是movieid和point构成的子字典

列表类型：moviedata

类和对象：ItemCF类和itemCF对象

数据库表：ratings表，存储所有用户记录

3.3.6、算法和流程

详细描述根据输入数据产生输出数据的算法和流程。

Loadmoviedata函数调用数据库中ratings表中数据，并将其划分问训练集和验证集，经过preprocessdata函数预处理后得到可供训练的数据，使用基于物品的协同过滤方法训练出itemCF模型。当传入用户id后，能够返回推荐电影id,再经过get\_movie\_data函数，将电影id变成涵盖电影名、类型、海报、播放链接的moviedata传入用户界面模块。

3.3.7、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

LoadMovieLensData函数：

文件：main.py

功能：读取数据库表中所有数据，并按照一定比例划分训练，测试集并返回

格式与参数：LoadMovieLensData( train\_rate)

变量：全局：old(上一次系统训练的时间)；局部：train和test列表

返回值：return PreProcessData(train), PreProcessData(test)，返回预处理后的训练集和测试集

PreProcessData函数：

文件：main.py

功能：将传入的[(‘userid’,’movieid’,’point’,’timestamp’)……]列表数据转换成字典

{‘userid’:{‘movieid’:’point’}……}便于处理

格式与参数：PreProcessData(originData)

变量：局部：trainData字典

返回值：return trainData返回预处理后的数据

Get\_user\_data函数：

文件：main.py

功能：输入用户id,返回该用户个性化的movie\_data

格式与参数：get\_user\_data(userid)

变量：全局：itemCF(训练好的模型)；局部：result（推荐结果）movie\_data（信息列表）

返回值：return movie\_data

Popularity函数:

文件：main.py

功能：计算每部电影的流行度

格式与参数：popularity(test)

变量：全局：pop 、all\_items

返回值：return pop

ItemCF类：

文件：main.py

功能：封装了init、similarity和recommend函数，实现基于物品协同过滤功能

格式与参数：ItemCF(object)

def \_\_init\_\_(self, trainData, similarity="cosine", norm=True)

变量：全局：N（topN），K(参考多少物品进行推荐)

返回值：返回训练好的对象

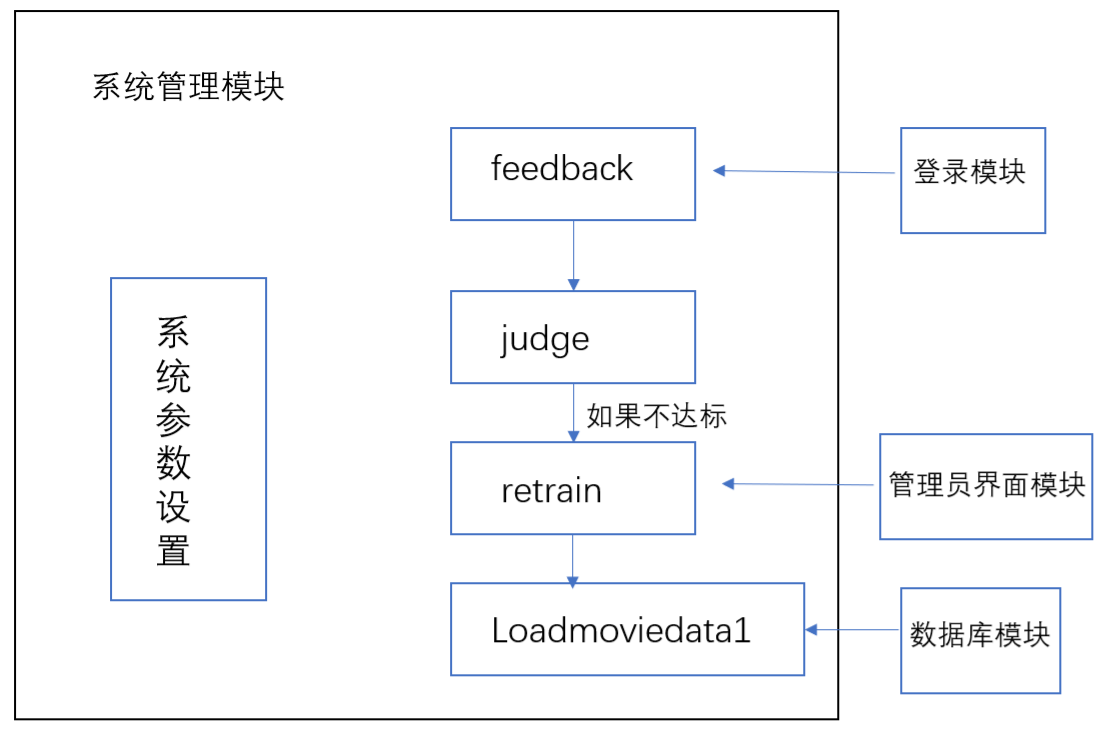
3.3.8 全局数据结构与该模块的关系

访问了old和itemCF,能够使用itemCF模型进行预测，并且更新上一次训练时间

**3.4、系统管理模块**

详细描述各功能模块的功能、数据结构、具体算法和流程等。

3.4.1、设计图



3.4.2、功能描述

简要描述模块1的业务功能。

一方面该模块实现系统初始化参数的设置，包括训练需要的N和K,显示需要的四项指标及其门限值

另一方面该模块能够不断检测距离上一次训练时长，若系统达不到一定要求，进行重新训练。

3.4.3、输入数据

详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

一是登录模块获取用户id后触发一次feedback检测距离上一次训练间隔时长

二是从数据库模块ratings表中获取系统中上一次训练之后新加入的记录，作为测试集传入judge函数

三是管理员界面重新训练按钮触发retrain函数

3.3.4、输出数据

详细描述模块1所产生的数据以及这些数据的表现形式。

该模块没有输出，但会调用推荐算法模块，进行重新训练，并且会更改部分全局参数

3.3.5、数据设计

给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）

字典类型：pop,将每部电影的流行度以字典类型存储

数值类型：N和K设置，

列表类型：recall\_score等各项指标，设计用列表存储，最后平均求得系统的该指标

数据库表：ratings表，存储所有用户记录

3.3.6、算法和流程

详细描述根据输入数据产生输出数据的算法和流程。

Feedback函数能够计算interval(距离上一次训练时长)，达到一定时长后（目前设定一周）调用judge函数检测系统各项指标，放入judge函数中的测试集是上一次训练后得到的真实数据，也是训练的模型未见过的数据。若不达标调用retrain函数进行重新训练，若达标则跳过，但间隔一个月时，就算指标达标也会自动重新训练。

3.3.7、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

feedback函数：

文件：main.py

功能：检测训练间隔时长，能够调用judge和retrain完成系统反馈监督和动态管理

格式与参数：feedback():

变量：全局：old,interval,hours, minutes, seconds,train,test,recall,F,precision,coverage

返回值：无返回值

judge函数：

文件：main.py

功能：对传入的测试集进行四大指标检测并返回

格式与参数：judge(test)

变量：全局：list\_user（训练集中的用户列表）

返回值：return recall,precision,F,coverage

retrain函数：

文件：main.py

功能：重新训练模型，并计算此时模型对于验证集的四大指标

格式与参数：retrain()

变量：全局：train,test,pop,recall, precision, F, coverage

返回值：无返回值

LoadMovieLensData1函数：

文件：main.py

功能：读取数据库表中上一次训练后加入的新数据，当做测试集返回

格式与参数：LoadMovieLensData1()

变量：全局：old(上一次系统训练的时间)；局部：test列表

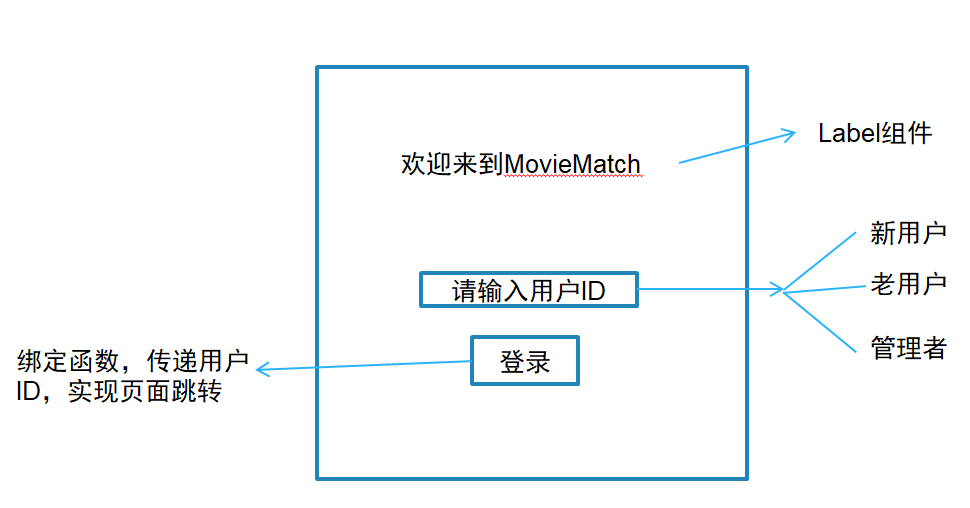
返回值：return PreProcessData(test)，返回预处理后的测试集

3.3.8 全局数据结构与该模块的关系

访问了old,interval,hours, minutes, seconds,train,test,recall,F,precision,coverage等全局变量，完成系统初始化参数设定和四大指标和流行度字典更新

**3.6、登录模块**

3.6.1、设计图



3.6.2、功能描述

简要描述模块1的业务功能。

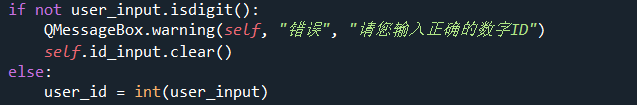
登录模块是系统运行后首先呈现在用户面前的模块，需要实现用户可以输入ID，并点击登录的功能，登陆后，会根据用户的ID跳转相应的页面

3.6.3、输入数据

详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

用户的ID，只能是数字ID，具体有效性检验规则如下:



3.6.4、输出数据

详细描述模块1所产生的数据以及这些数据的表现形式。

传递用户ID给推荐算法模块，页面跳转

3.6.5、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

函数名称：login

功能：实现用户点击后能根据用户输入的ID进行页面跳转

格式与参数：def login(self,stacked\_widget)

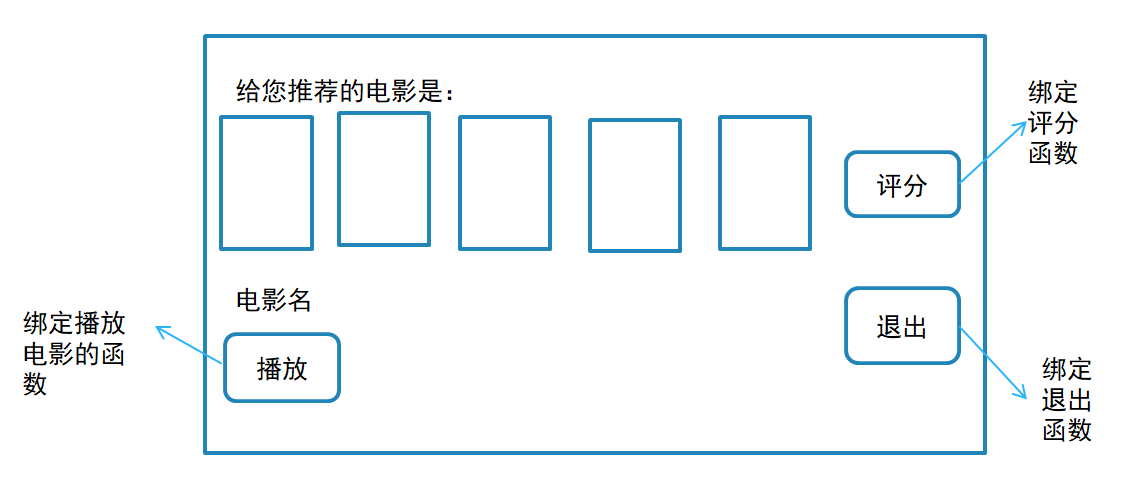
全局变量：无

局部变量：user\_input，user\_id

无返回值

**3.7、用户界面模块**

3.7.1、设计图



3.7.2、功能描述

简要描述模块1的业务功能。

用户界面模块是用户登陆后呈现的页面，能够实现将推荐的电影的id、名称、播放链接呈现给用户，用户可以在此页面上对推荐的电影进行评分，或者退出登录

3.7.3、输入数据

详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

用户对按钮的点击、评分

3.7.4、输出数据

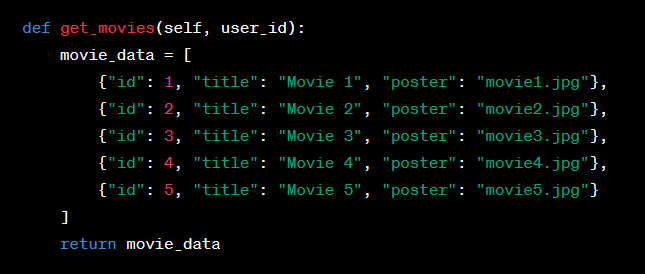
详细描述模块1所产生的数据以及这些数据的表现形式。

跳转到电影播放页面、跳转到初始登录页面、将评分插入数据库

3.7.5、数据设计

给出本程序中的局部数据结构说明，包括数据结构名称，功能说明，具体数据结构说明（定义、注释设计、取值）等。相关数据库表，数据存储设计（具体说明需要以文件方式保存的数据文件名、数据存储格式、数据项及属性等。）

Movie\_data:需要有要呈现的电影的id、名称、图片链接等信息，如下图：



3.7.6、算法和流程

详细描述根据输入数据产生输出数据的算法和流程。

流程：此页面接收到了登录模块传来的user\_id这个信息，通过get\_movie\_data函数获得movie\_data，然后呈现到页面中

3.7.7、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

(1)函数名称：get\_movie\_data

功能：获得电影的id、名称、图片地址信息

格式与参数：def get\_movie\_data(user\_id)

全局变量：itemCF，协同过滤的类实例化，需要由此获得推荐电影相关信息

局部变量：result接收电影信息

返回值：movie\_data

(2)函数名称：get\_movie\_link

功能：从数据库中获得电影的播放链接

格式与参数：def get\_movie\_link(self,movie\_id)

局部变量：result用于接收获取的数据

(3)函数名称：confirm\_submit

功能：确认提交评分

格式与参数：def confirm\_submit(self, stacked\_widget)

局部变量：reply用于接收提示框的返回值

(4)函数名称：submit\_ratings

功能：提交评分并将评分记录插入数据库

格式与参数：def submit\_ratings(self, stacked\_widget)

局部变量：selected\_movie，rating，current\_time分别接收选中的电影、评分和时间戳

cursor打开数据库

(5)函数名称：disable\_rating\_input

功能：实现用户评分一次后不可以重复评分

格式与参数：def disable\_rating\_inputs(self)

(6)函数名称：logout

功能：实现退出登录

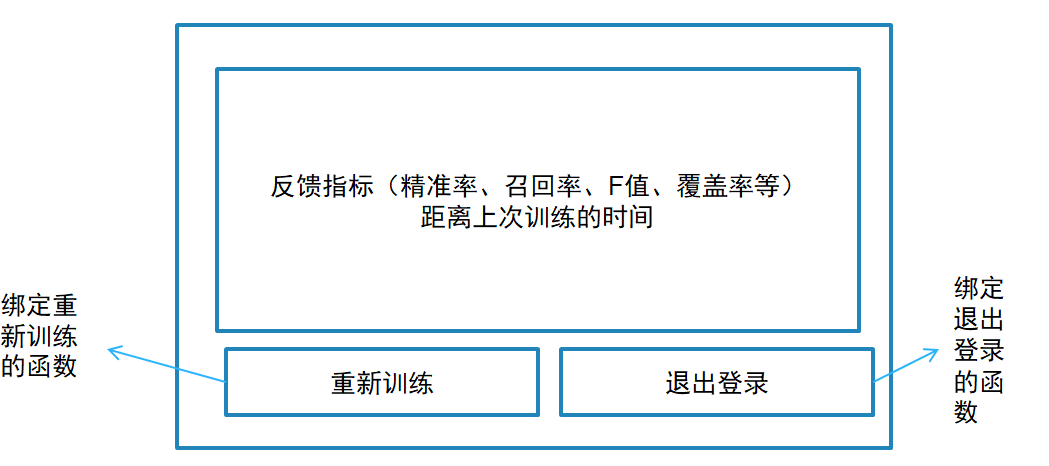
3.7.8 全局数据结构与该模块的关系

说明该模块访问了哪些全局数据结构。

此模块访问了itemCF这个全局数据结构，从而获得协同过滤后的结果

**3.8、管理员界面模块**

3.8.1、设计图



3.8.2、功能描述

简要描述模块1的业务功能。

管理者界面模块实现了对系统当前推荐效果的呈现，通过管理员ID登录后，在界面中，管理员可以看到反馈指标、距离上次训练的时间等，并且可以对系统进行重新训练，也可以退出登录。

3.8.3、输入数据

详细描述用户输入的数据(包括任何输入设备)以及这些数据的有效性检验规则。

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

管理员的ID，管理员的对按钮的点击。

3.8.4、输出数据

详细描述模块1所产生的数据以及这些数据的表现形式。

系统进行重新训练，页面跳转。

3.8.5、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

(1)函数名称：retrain

功能：实现管理员点击后，系统进行重新训练

格式与参数：def retrain()

全局变量：四项反馈指标recall,precision,F,coverage，以及训练集、测试集

(2)函数名称：logout

功能：实现退出登录

格式与参数：def logout(self, stacked\_widget)

3.8.6 全局数据结构与该模块的关系

说明该模块访问了哪些全局数据结构。

此模块访问了itemCF，recall，precision，F，coverage、train、test等全局数据结构。

……

# 4、 接口设计

**4.1、 用户接口**

说明将向用户提供的接口。

图形化的接口，如登录、提交评分、退出登录等按钮

**4.2、 外部接口**

描述本软件同外界的所有接口，包括软件、硬件、本系统与各支持系统之间的接口关系、控制方式。

播放按钮与外界浏览器的相连接

**4.3、 内部接口**

4.3.1、 接口说明

例如：xx子模块通过xx从xx子模块取得xx等，相关标准，调用示例，可根据需要增加章节描述接口。

a.登录模块从按钮绑定函数中获得用户输入ID

b.用户界面模块从登录模块获得用户输入的ID，从数据库模块与推荐算法模块获得推荐出来的电影名、电影类型、电影海报、电影播放链接

c.管理员界面模块从登录模块获得用户输入的ID，从系统管理模块获得系统的四项反馈指标以及训练间隔时长

d.推荐算法模块通过传参从登录模块获取用户id

e.推荐算法模块通过pymysql从数据库模块获取记录

4.3.2、 调用方式

例如：

/\*\*

\*通过用户服务号码取得该客户认证密码等信息，如果该客户存在返回为0，其他情况参考错误编码

\*/

public RUserInfo getUserInfo (String userNo);

接口a：通过函数调用

def login(self, stacked\_widget):

user\_input = self.id\_input.text()

接口b：通过类的定义，函数的参数调用

movie\_ratings\_window = MovieRatingsWindow(stacked\_widget, user\_id)

self.movie\_data = get\_user\_data(user\_id)

接口c：通过类的定义，函数的参数调用

if user\_id == manager\_id:

movie\_manager\_window = MovieManagerWindow(stacked\_widget)

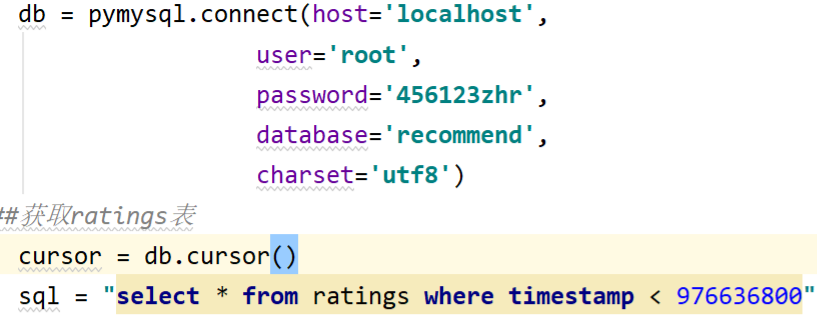
recall,precision,F,coverage=judge(test) 四项反馈指标

seconds=convert\_seconds(interval) 训练间隔时长

接口d：通过函数调用

itemCF.recommend(userid, 5,80)

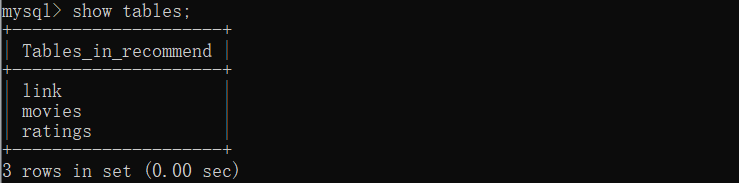
接口e：通过pymysql调用



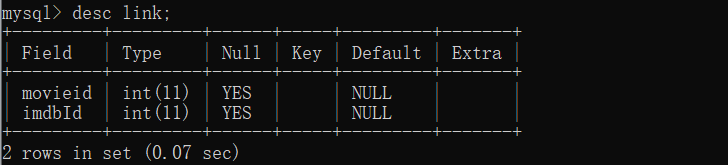
# 5、数据库设计

描述所使用的数据库系统,及数据库和数据表设计。如果系统不以数据库方式存储数据则可省略。

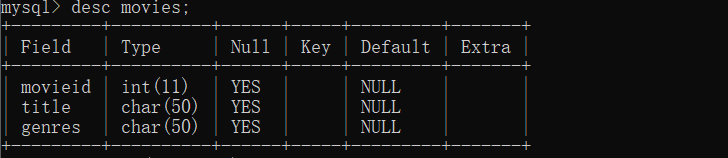
本系统使用了MySQL数据库系统，建立了一个名为recommend的数据库，数据库下有三个数据表，具体描述如下图：



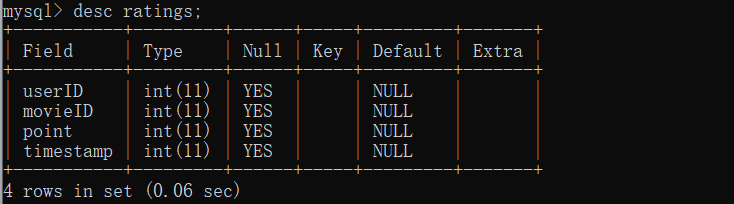
Link表存放电影id以及电影的播放链接



Movies表存放电影的id、名称以及电影类型



Ratings表存放用户id、看过的电影id、用户的评分以及时间戳



# 6、系统出错处理

**6.1、 出错信息**

用一览表的方式说明每种可能的错误和故障，以及系统输出信息的形式、含义和处理方式。

|  |  |
| --- | --- |
| 可能出现的错误 | 系统输出 |
| 用户输入id非数字 | print("输入无效，请输入一个合法id！") |
| 用户对喜欢电影id输入非法 | print("输入无效，请输入一个整数！") |
| 用户对喜欢电影的评分非法 | print("输入不全，请再输入您的评分") |
| 用户选择喜欢电影类型非法 | print("输入无效，请输入喜欢类型") |

**6.2、 补救措施**

说明故障出现后可能采取的补救措施，如恢复、再启动技术等。

对于上述输入错误，通过前端知识，生成一个提示框，提醒用户输入错误，用户点击确定后可以重新输入.

# 7、其他设计

如系统安全设计、性能设计等。