predictionio由两部分组成： eventserver和predictionserver，eventserver负责数据源的收集，predictionserver为推荐引擎。三类数据需要保存：数据源eventdata、模型（modeldata）、metadata。三类数据默认保存在PostgreSQL

**eventserver部署完成并已启动：pio-start-all，pio status**

**predictionserver如下：**

一、新建一个appid (pio app new MyApp1)

[INFO] [HBLEvents] The table pio\_event:events\_3 doesn't exist yet. Creating now...

[INFO] [App$] Initialized Event Store for this app ID: 3.

[INFO] [Pio$] Created a new app:

[INFO] [Pio$] Name: MyApp1

[INFO] [Pio$] ID: 3

[INFO] [Pio$] Access

Key: 5dyFmh3NGpCai7EoJzQ6YBMdzSimF5jBUlseWHPArdtuCjXqGAsJiR2jv99o5jz7

二、查看所有的appid(pio app list)

三、导入数据

[root@localhostbin]#ACCESS\_KEY=5dyFmh3NGpCai7EoJzQ6YBMdzSimF5jBUlseWHPArdtuCjXqGAsJiR2jv99o5jz7

[root@localhostbin]# curl -i -X POST http://localhost:7070/events.json?accessKey=5dyFmh3NGpCai7EoJzQ6YBMdzSimF5jBUlseWHPArdtuCjXqGAsJiR2jv99o5jz7 \

-H "Content-Type: application/json" \

-d '{

"event" : "rate",

"entityType" : "user",

"entityId" : "u0",

"targetEntityType" : "item",

"targetEntityId" : "i0",

"properties" : {

"rating" : 5

}

"eventTime" : "2014-11-02T09:39:45.618-08:00"

}'

HTTP/1.1 201 Created

Server: spray-can/1.3.3

Date: Fri, 13 Oct 2017 17:34:41 GMT

Content-Type: application/json; charset=UTF-8

Content-Length: 57

{"eventId":"illrLcpg1dDE2bvZZ1NpggAAAUlxl11Skvne7F9IGMI"}[root@localhost bin]#

[root@localhost bin]# curl -i -X POST http://localhost:7070/events.json?accessKey=$ACCESS\_KEY \

> -H "Content-Type: application/json" \

> -d '{

> "event" : "buy",

> "entityType" : "user",

> "entityId" : "u1",

> "targetEntityType" : "item",

> "targetEntityId" : "i2",

> "eventTime" : "2014-11-10T12:34:56.123-08:00"

> }'

HTTP/1.1 201 Created

Server: spray-can/1.3.3

Date: Fri, 13 Oct 2017 17:35:26 GMT

Content-Type: application/json; charset=UTF-8

Content-Length: 57

{"eventId":"Z0813DMQIKz7N4VGxZhmngAAAUmbap37oWglLlIYxsU"}[root@localhost bin]#

[root@localhost bin]# curl -i -X GET "http://localhost:7070/events.json?accessKey=$ACCESS\_KEY"

HTTP/1.1 200 OK

Server: spray-can/1.3.3

Date: Fri, 13 Oct 2017 17:36:03 GMT

Content-Type: application/json; charset=UTF-8

Content-Length: 528

[{"eventId":"Z0813DMQIKz7N4VGxZhmngAAAUmbap37oWglLlIYxsU","event":"buy","entityType":"user","entityId":"u1","targetEntityType":"item","targetEntityId":"i2","properties":{},"eventTime":"2014-11-10T12:34:56.123-08:00","creationTime":"2017-10-13T17:35:26.664Z"},{"eventId":"illrLcpg1dDE2bvZZ1NpggAAAUlxl11Skvne7F9IGMI","event":"rate","entityType":"user","entityId":"u0","targetEntityType":"item","targetEntityId":"i0","properties":{"rating":5},"eventTime":"2014-11-02T09:39:45.618-08:00","creationTime":"2017-10-13T17:34:40.998Z"}][root@localhost bin]#

[root@localhost bin]#

**将推荐引擎部署为服务：pio build，pio train，pio deploy**

**获取推荐结果：curl -H "Content-Type: application/json" -d '{ "user": "u1", "num": 4 }' http://localhost:8000/queries.json**

使用python脚本导入数据

1. 安装python的SDK

pip install predictionio或者 easy\_install predictionio

1. 下载一些模板数据并导入

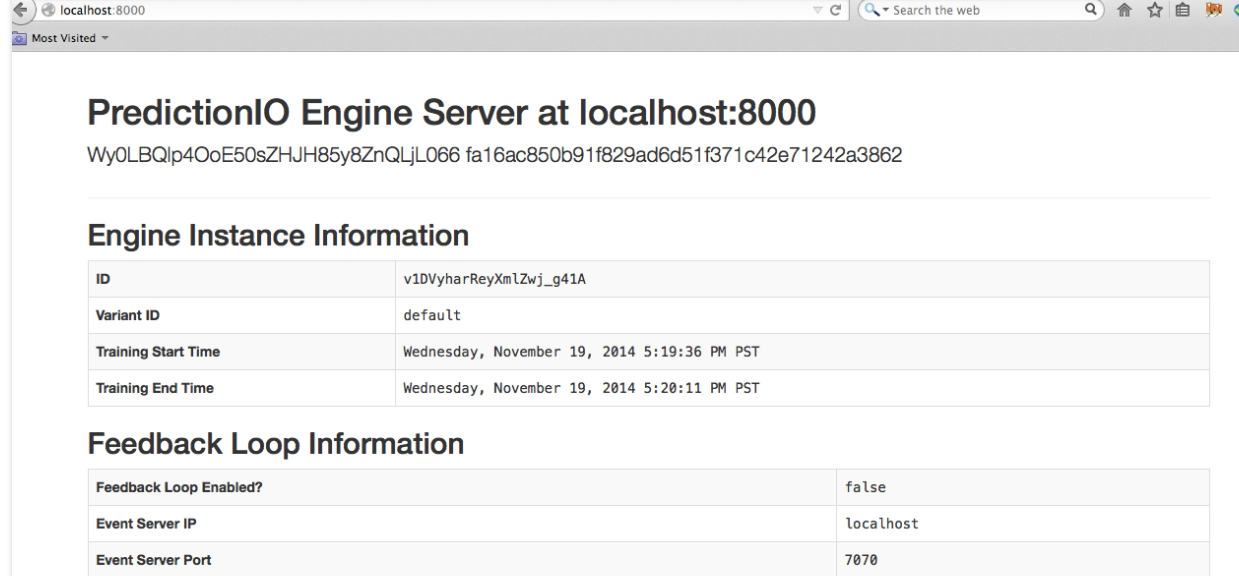
$curl https://raw.githubusercontent.com/apache/spark/master/data/mllib/sample\_movielens\_data.txt --create-dirs -o data/sample\_movielens\_data.txt

$ python data/import\_eventserver.py --access\_key $ACCESS\_KEY

1. 修改engine.json里面的appname,和数据文本名字（我把上面的两个文件拷贝的/engines/ur/data下）
2. pio build –verbose看到[INFO] [Console$] Your engine is ready for training.说明成功
3. pio train看到 [INFO] [CoreWorkflow$] Training completed successfully. 说明成功
4. pio deploy看到 [INFO] [HttpListener] Bound to /0.0.0.0:8000

[INFO] [MasterActor] Bind successful. Ready to serve. 说明成功

1. 在游览器上<http://localhost:8000>看到有



1. 使用引擎

查询$ curl -H "Content-Type: application/json" \

-d '{ "user": "1", "num": 4 }' <http://localhost:8000/queries.json>

