

博雅云实训项目训练营

实训平台操作说明

目录

1 引言	2
1.1 编写目的	2
1.2 运行环境	2
2 功能描述	2
2.1.1 账号信息管理	2
2.1.2 闯关	2
2.1.3 练习	2
2.1.4 报告	3
2.1.5 记录	3
3 操作指南	3
3.1 用户操作	3
3.1.1 账号管理	3
3.1.2 闯关	4
3.1.3 练习	12
3.1.4 报告	16
3.1.5 记录	19
4 联系方式	20

1 引言

1.1 编写目的

为了让学员更好的使用实训平台，故编写了使用说明。

1.2 运行环境

推荐使用 Chrome 浏览器。

实训环境：Python3 版本。

2 功能描述

2.1.1 账号信息管理

支持登录、退出登录，登录出现问题请联系相关负责人。

2.1.2 闯关

用户可以总览项目的各个环节，并依次进行项目闯关，只有通过上一环节才能解锁下一环节。支持用户边阅读边操作，并记录实训笔记、查看运行结果、提交结果检验对错、生成报告、填写报告结论。

2.1.3 练习

用户可以总览项目的各个环节，并自由进入任意环节，可以边阅读边操作，并查看运行结果、变量描述、提交结果检验对错。

2.1.4 报告

用户可以对自已的报告进行生成、删除、编辑、浏览、查看进度、保存、重命名、提交、导出操作。

2.1.5 记录

用户可以查看自己提交的报告记录，预览提交的报告内容，可以撤回5分钟之内的报告。

用户可以查看自己的笔记汇总，编辑、导出笔记。

3 操作指南

3.1 用户操作

3.1.1 账号管理

登录

进入登录页面，依次输入自己的账号、密码，输入无误即可登录。



退出

用户登录后，鼠标移入右上角姓名处，在弹出的下拉菜单中点击“退出”回到登录页面。



3.1.2 闯关

闯关模式

登录成功后默认进入闯关模式，也可点击左侧“闯关”导航进入闯关

模式。

闯关模式首页显示项目各环节概述，点击概述文字下“开始”按钮可进入项目闯关页进行实训，只有通过了前面的闯关环节才能解锁后面的闯关环节，未解锁的环节有锁形图标标注。

页面右侧菜单包含项目流程缩略图和目录，点击流程缩略图可以查看项目全流程；鼠标悬停到“？”图标上，可查看该环节涉及到的课程，点击课程名称可以查看数据酷客内相关课程详情页。

项目背景与目标

移动互联网时代的到来，消费者的需求呈现出精细化和多样化的特点，市场竞争日益加剧，产品的生命周期越来越短，传统的跑马圈地式营销难见成效。

随着大数据分析技术的发展和各种社交工具的出现，依托精准定向分析方法，企业可以摆脱过去大众营销的套路，准确找到其产品的目标消费群，最大限度的获得目标客户，实现精准营销。

数据源：业务数据、日志数据、第三方数据

数据预处理：数据清洗、数据集成、数据转换

探索性数据分析：数据分布分析、数据关联分析、数据异常检测

客户标签体系构建：数据标签的提取、数据标签的关联、数据标签的评估

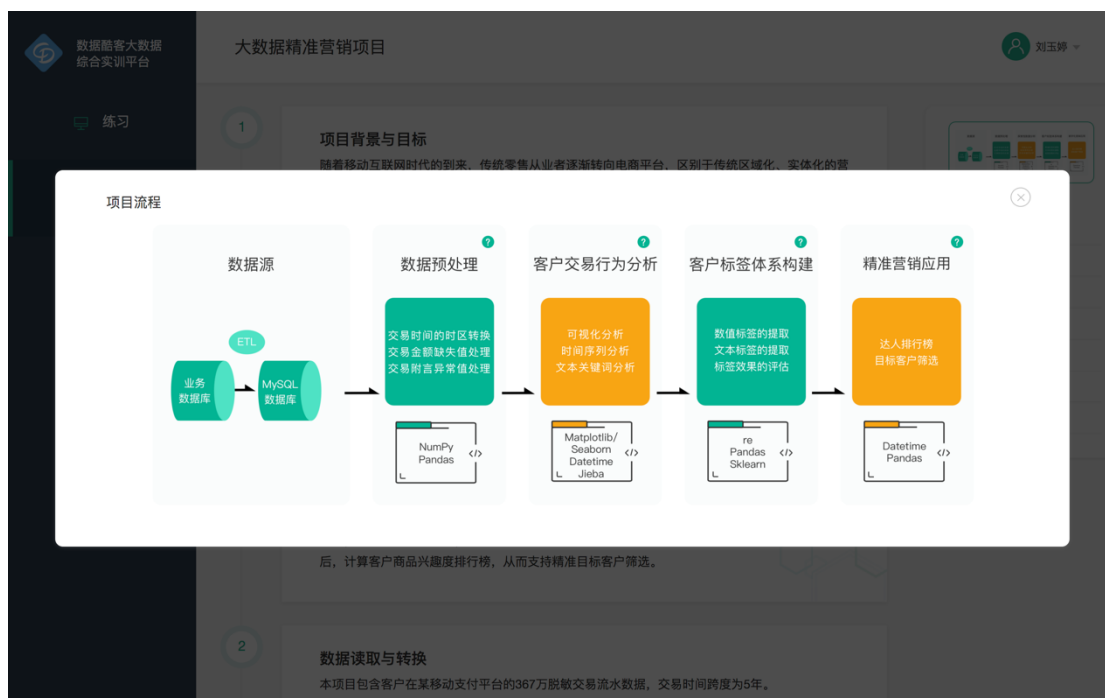
数字化营销应用：精准营销、个性化推荐、客户流失预警

首先通过对海量结构化数据和非结构化文本数据的深度分析和挖掘，构建全方位的客户标签体系。其次，基于客户标签体系，从基本信息、消费能力、行为习惯等多个维度对客户进行精准画像。最后，计算用户商品兴趣度排行榜，从而支持精准目标客户筛选。

数据读取与转换

本项目包含客户在某移动支付平台的367万条交易流水数据。每条流水信息记录了客户id、交易时间、交易金额、交易附言四个字段，每个客户有多条流水记录。

英文名称	中文名称	备注

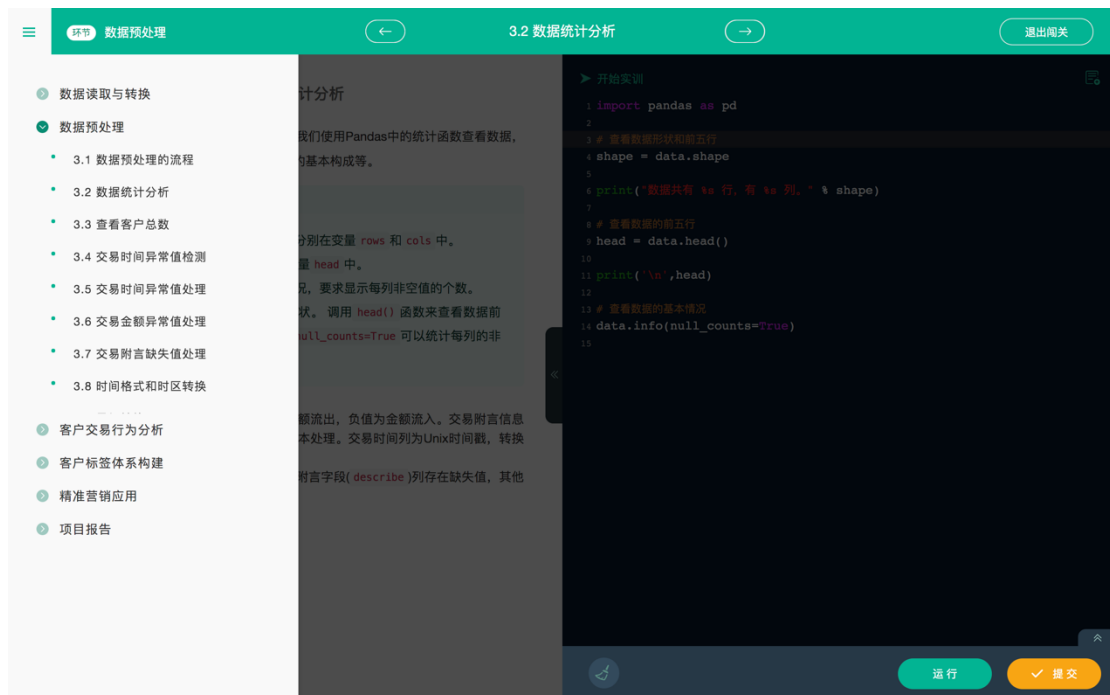


项目流程查看

闯关模式实训操作

在闯关模式实训页面，用户点击查看目录按钮可以展开/收起项目目录，点击目录中的名称，若该环节已解锁可以跳转到对应实训环节，

如未解锁则点击无效。未解锁环节前有锁型标记。已提交环节前有绿点标记。



展开目录效果

点击上一页可以查看上一个环节内容，点击下一页按钮，若下一环节已解锁则跳转到下一关卡，若下一环节未解锁则点击无效；点击代码框左下角图标清空已输入代码；点击“退出闯关”返回到闯关模式首页。

客户交易行为分析

4.6 客户交易次数的可视化分析

退出闯关

上一环节

4.6 客户交易次数的可视化分析

下一环节

目录

笔记功能

在对客户交易行为分析过程中，除了从时间维度上分析交易次数外，我们还可以从每个客户出发，可视化分析不同客户的交易分布，下面我们分析一下客户的交易次数分布。

Python绘图库Seaborn中的 `kdeplot()` 函数可以绘制“核密度图”，用来估计数据的密度函数，从而展现数据样本本身的分布特征。 `kdeplot()` 函数的语法为： `kdeplot(data, shade=False, legend=True)`，其中：

- `data`：输入数据。
- `shade`：添加区域着色，默认为 `False`，如果为 `True`，则在KDE曲线下方的区域中着色。
- `legend`：添加图例，默认为 `True`。

实训任务

- 请使用 `value_counts()` 函数统计 `data` 每个客户的交易次数，接着调用 `kdeplot()` 函数画出客户交易次数的核密度图，要求在KDE曲线下方的区域中着色，且不设置图例。
- 请设置 `x` 轴标签为 `交易次数`，`y` 轴标签为 `频率`。
- 请设置图的标题为 `客户交易次数分布`。

可以看到客户的交易次数从0-7000不等，分布主要集中在0-1000。在之后的指标构建中，我们要根据客户的交易记录构建指标，如果在此平台交易记录太少，我们认为此类为休眠客户。

开始实训

```

1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 fig = plt.figure(figsize=(16, 5))
5
6 # 绘制核密度图
7 sns.kdeplot(data.user_id.value_counts(), shade=True, legend=False)
8
9 # 设置x, y轴标签
10 plt.xlabel('交易次数')
11 plt.ylabel('频率')
12
13 # 设置图的标题
14 plt.title('客户交易次数分布')
15
16 plt.show()

```

清空代码

运行

提交

项目闯关页面左侧为项目内容说明部分；右侧为代码区，用户可在该区域运行和提交代码，支持代码补全。运行后可查看运行结果和相关变量，提交后不仅可以查看运行结果和相关变量，还可查看结果对错并将此结果自动更新报告草稿中。

客户交易行为分析

4.6 客户交易次数的可视化分析

退出闯关

4.6 客户交易次数的可视化分析

笔记功能

在对客户交易行为分析过程中，除了从时间维度上分析交易次数外，我们还可以从每个客户出发，可视化分析不同客户的交易分布，下面我们分析一下客户的交易次数分布。

Python绘图库Seaborn中的 `kdeplot()` 函数可以绘制“核密度图”，用来估计数据的密度函数，从而展现数据样本本身的分布特征。 `kdeplot()` 函数的语法为： `kdeplot(data, shade=False, legend=True)`，其中：

- `data`：输入数据。
- `shade`：添加区域着色，默认为 `False`，如果为 `True`，则在KDE曲线下方的区域中着色。
- `legend`：添加图例，默认为 `True`。

实训任务

- 请使用 `value_counts()` 函数统计 `data` 每个客户的交易次数，接着调用 `kdeplot()` 函数画出客户交易次数的核密度图，要求在KDE曲线下方的区域中着色，且不设置图例。
- 请设置 `x` 轴标签为 `交易次数`，`y` 轴标签为 `频率`。
- 请设置图的标题为 `客户交易次数分布`。

可以看到客户的交易次数从0-7000不等，分布主要集中在0-1000。在之后的指标构建中，我们要根据客户的交易记录构建指标，如果在此平台交易记录太少，我们认为此类为休眠客户。

开始实训

```

1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 fig = plt.figure(figsize=(16, 5))
5
6 # 绘制核密度图
7 sns.kdeplot(data.user_id.value_counts(), shade=True, legend=False)
8
9 # 设置x, y轴标签
10 plt.xlabel('交易次数')

```

结果输出

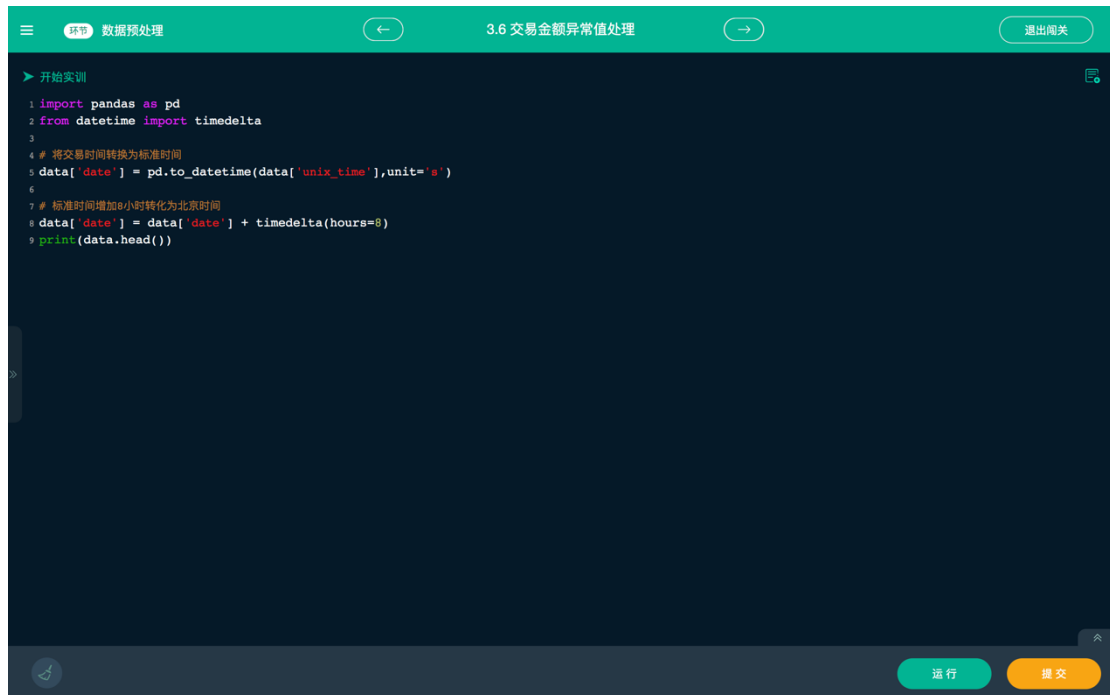
变量描述

- `data`
- `fig`

运行

提交

点击“提交”后效果



代码区展开效果

3.6 交易金额异常值处理

在处理过交易时间的异常值之后，接下来我们尝试分析一下交易金额这一列。

通过观察，发现存在一些行的交易金额为 `\N`，这是数据中的一个特殊字符，可以把它视为空。如下表所示：

user_id	payment	describe	unix_time
3916005	\N	6网上支付	1461168000
3916005	\N	6网上支付	1474473600
3916005	\N	6结息	1474214400
3916005	\N	6网上支付	1473609600
3916005	\N	6网上支付	1471536000

对于这些交易金额异常的行，我们无法进行后续的分析，对这些数据我们直接将其删除。

实训任务

- 请去除 `data` 中交易金额为 `\N` 的行，并将去除后的结果仍然保存在 `data` 中。

提示：`\` 为转义字符，对其进行匹配要使用 `\\` 形式。

打印结果为空则代表我们已经对交易金额为 `\N` 的行去除成功。

开始实训

```

1 import pandas as pd
2 from datetime import timedelta
3
4 # 将交易时间转换为标准时间
5 data['date'] = pd.to_datetime(data['unix_time'], unit='s')
6
7 # 标准时间增加8小时转化为北京时间
8 data['date'] = data['date'] + timedelta(hours=8)
9 print(data.head())
10
11 pass
12 pow
13 property
14 print
    
```

运行 提交

支持代码补全

3.6 交易金额异常值处理

退出/关闭

3.6 交易金额异常值处理

在处理过交易时间的异常值之后，接下来我们尝试分析一下交易金额这一列。

通过观察，发现存在一些行的交易金额为 \N，这是数据中的一个特殊字符，可以把它视为空。如下表所示：

user_id	payment	describe	unix_time
3916005	\N	6网上支付	1461168000
3916005	\N	6网上支付	1474473600
3916005	\N	6结息	1474214400
3916005	\N	6网上支付	1473609600
3916005	\N	6网上支付	1471536000

对于这些交易金额异常的行，我们无法进行后续的分析，对这些数据我们直接将其删除。

实训任务

请去除 data 中交易金额为 \N 的行，并将去除后的结果仍然保存在 data 中。

提示：\ 为转义字符，对其进行匹配要使用 \\ 形式。

打印结果为空则代表我们已经对交易金额为 \N 的行去除成功。

开始实训

```

1 import pandas as pd
2 from datetime import timedelta
3
4 # 将交易时间转换为标准时间
5 data['date'] = pd.to_datetime(data['unix_time'],unit='s')
6
7 # 标准时间增加8小时转化为北京时间
8 data['date'] = data['date'] + timedelta(hours=8)
9 print(data.head())
10

```

结果输出

	user_id	payment	describe	unix_time	date
0	22171955	6500	湖州天虹百货有限公司	1509379200	2017-10-31
1	22171955	-2096	支付宝 - 中国铁路总公司资金清算中心	1509120000	2017-10-28
2	22171955	22450	湖州市星火服装有限公司	1509379200	2017-10-31
3	22171955	20900	湖州市星火服装有限公司	1509379200	2017-10-31
4	22171955	2340	吴兴晓华化妆品商行	1509379200	2017-10-31

变量描述

data

reload

运行 提交

代码运行效果

4.6 客户交易行为的可视化分析

退出/关闭

4.6 客户交易次数的可视化分析

在对客户交易行为分析过程中，除了从时间维度上分析交易次数外，我们还可以从每个客户出发，可视化分析不同客户的交易分布，下面我们分析一下客户的交易次数分布。

Python绘图库Seaborn中的 `kdeplot()` 函数可以绘制“核密度图”，用来估计数据的密度函数，从而展现数据样本本身的分布特征。 `kdeplot()` 函数的语法为： `kdeplot(data, shade=False, legend=True)`，其中：

- `data`：输入数据。
- `shade`：添加区域着色，默认为 `False`，如果为 `True`，则在KDE曲线下方的区域中着色。
- `legend`：添加图例，默认为 `True`。

实训任务

请使用 `value_counts()` 函数统计 data 每个客户的交易次数，接着调用 `kdeplot()` 函数画出客户交易次数的核密度图，要求在KDE曲线下方的区域中着色，且不设置图例。

请设置 x 轴标签为 交易次数，y 轴标签为 频率。

请设置图的标题为 客户交易次数分布。

可以看到客户的交易次数从0-7000不等，分布主要集中在0-1000。在之后的指标构建中，我们要根据客户的交易记录构建指标，如果在此平台交易记录太少，我们认为此类为休眠客户。

开始实训

```

1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 fig = plt.figure(figsize=(16, 5))
5
6 # 绘制核密度图
7 sns.kdeplot(data.user_id.value_counts(),shade=True,legend=False)
8
9 # 设置x, y轴标签
10 plt.xlabel('交易次数')

```

结果输出

变量描述

data

fig

运行 提交

代码提交效果（答案正确）

1

www.cookdata.cn

环节

数据读取与转换

2.3 对客户数据进行格式转换

退出/闯关

2.3 对客户数据进行格式转换

在上一步中，我们已经成功地在数据库中进行了一些查询和读取操作，现在我们需要利用Python中的Pandas库进行数据格式的转换。

题目已预先从数据表中读入全部数据(`SELECT * FROM flow_data`), 并命名它为 `data` 。但目前的数据是以嵌套元组的形式存储的，所以需要将其转换为Pandas内置的DataFrame对象，再进行后续操作。

DataFrame对象是Pandas最常用的数据对象。DataFrame是一个二维的表结构，它可以存储多种不同的数据类型。

使用Pandas中的 `DataFrame()` 函数可以创建一个DataFrame对象。函数语法为: `DataFrame(data, index, columns, dtype)` , 其中:

- `data` : 输入的数据, 可以是各种形式, 比如ndarray、Series、map、list、dict、constant和另一个DataFrame。
- `index` : 设置行索引。
- `columns` : 设置列名。
- `dtype` : 设置每列的数据类型。

嵌套元组不能直接保存为DataFrame, 需要将外层的元组先转换成列表, 再使用 `DataFrame()` 函数将列表进行转换。但是直接转换时, 列名为默认值(即0、1、2、3), 所以需要在转化时重命名列名。

实训任务

开始实训

```

1 import pandas as pd
2
3 # 将嵌套元组转换为DataFrame
4 data = pd.DataFrame(list(data))
5 # 设置列名
6 data
7

```

结果输出

	0	1	2	3
0	22171955	6500		湖州天虹百货有限公司 150
9379200				
1	22171955	-2096		支付宝 - 中国铁路总公司资金清算中心
1509120000				
2	22171955	22450		湖州市星火服装有限公司 15
09379200				
3	22171955	20900		湖州市星火服装有限公司 15
09379200				
4	22171955	2340		吴兴晓华化妆品商行 1509
379200				
5	32394412	7600		财付通快捷支付 (客服 :0755-86013860) 150937
9200				
6	32394412	4100		北京弘泰基业 (大悦城) 15093
79200				

变量描述

data

f

运行

提交

代码提交效果 (答案错误)

点击笔记按钮可记录本环节实训笔记，支持自定义文字字号、颜色、背景颜色，支持插入代码、图片、公式。

环节

数据预处理

3.2 数据统计分析

退出/闯关

3.2 数据统计分析

上一步我们介绍了数据预处理的流程，接下来我们使用Pandas中的统计函数查看数据，了解数据的基本信息，如数据的行列数、数据的基本构成等。

实训任务

- 请查看数据 `data` 的总行数和总列数，分别在变量 `rows` 和 `cols` 中。
- 请查看数据的前五行，将结果保存在变量 `head` 中。
- 请使用 `info()` 函数查看数据的基本情况，要求显示每列非空值的个数。提示：调用 `shape` 属性可查看数据形状。调用 `head()` 函数来查看数据前五。调用 `info()` 函数时设置参数 `null_counts=True` 可以统计每列的非空值的个数。

- 可以看到，客户交易流水记录中正值为金额流出，负值为金额流入。交易附言信息为中文描述，该列数据之后可能要进行文本处理。交易时间列为Unix时间戳，转换为标准北京时间更易处理。
- 从数据的统计信息中可以看到，只有交易附言字段(`describe`)列存在缺失值，其他列均无缺失值。

开始实训

```

1 import pandas as pd
2
3 # 查看数据形状和前五行
4 shape = data.shape
5
6 print("数据共有 %s 行, 有 %s 列。" % shape)
7
8 # 查看数据的前五行
9 head = data.head()
10
11 print('\n',head)
12
13 # 查看数据的基本情况
14 data.info(null_counts=True)
15

```

运行

提交

记录本环节笔记

项目闯关页最后一个环节项目为报告任务，在闯关过程中会自动更新填写报告，用户需要根据报告任务填写完善报告。

页面左侧为项目报告导航栏，可查看任务总数及任务进度，点击任务环节可进入到相应任务中，点击图中图片可快速进入答题。

支持不限次修改内容。



3.1.3 练习

练习模式

点击左侧导航栏“练习”进入练习模式。

练习模式首页显示项目各环节概述，点击概述文字下的“开始”按钮可以进入实训练习。页面右侧菜单包含项目流程缩略图和目录，点击流程缩略图可以查看项目全流程；鼠标悬停到“？”图标上，可查看该环节涉及到的课程，点击课程名称弹出新页签可以查看数据酷客内课程详情页。

数据酷客大数据综合实训平台

练习

闯关

报告

记录

大数据精准营销项目

1

项目背景与目标

移动互联网时代的到来，消费者的需求呈现出了精细化和多样化的特点，市场竞争日益加剧，产品的生命周期越来越短，传统的跑马圈地式营销难见成效。

随着大数据分析技术的发展和各种社交工具的出现，依托精准定向分析方法，企业可以摆脱过去大众营销的套路，准确找到其产品的目标消费群，最大限度的获得目标客户，实现精准营销。

数据源

数据预处理

探索性数据分析

客户标签体系构建

数字化营销应用

首先通过对海量结构化数据和非结构化文本数据的深度分析和挖掘，构建全方面的客户标签体系。其次，基于客户标签体系，从基本信息、消费能力、行为习惯等多个维度对客户进行精准画像。最后，计算用户商品兴趣度排行榜，从而支持精准目标客户筛选。

2

数据读取与转换

本项目包含客户在某移动支付平台的367万条交易流水数据。每条流水信息记录了客户id、交易时间、交易金额、交易附言四个字段，每个客户有多条流水记录。

英文名称	中文名称	备注
------	------	----

项目流程 (缩略图)

01-项目背景与目标

02-数据读取与转换

03-数据预处理

04-探索性数据分析

05-客户标签体系构建

06-精准营销应用

07-项目总结

练习模式实训操作

在项目练习页面，点击查看目录按钮可以展开/收起项目目录，点击目录中的名称可以跳转到对应实训练习页；点击上一页/下一页按钮可以查看上一个/下一个练习内容；点击代码框左下角图标清空已输入代码；点击“退出练习”返回到练习模式首页。

数据读取与转换

2.2 从MySQL数据库中读取客户数据

退出练习

展开/收起目录

查看上一个练习

查看下一个练习

返回到练习详情页

2.2 从MySQL数据库中读取客户数据

在本项目中，数据存在MySQL中，首先从MySQL数据库导入数据。在这个步骤中我们需要用到 `pymysql` 模块。数据表名称为 `flow_data`，各字段及其格式如下图所示：

flow_data

user_id:int

payment:float

describe:text

unix_time:int

图2.1: flow_data数据表结构

数据库连接对象 `connection`

- 连接对象：建立Python客户端与数据库的连接
- 创建方法：`connection=pymysql.connect(参数)`

数据库游标对象 `cursor`

- 游标对象：用于执行查询和获取结果
- 创建方法：`cursor = conn.cursor()`
- 在MySQL中，所有sql语句的执行都要在游标对象的参与下完成

开始实训

```

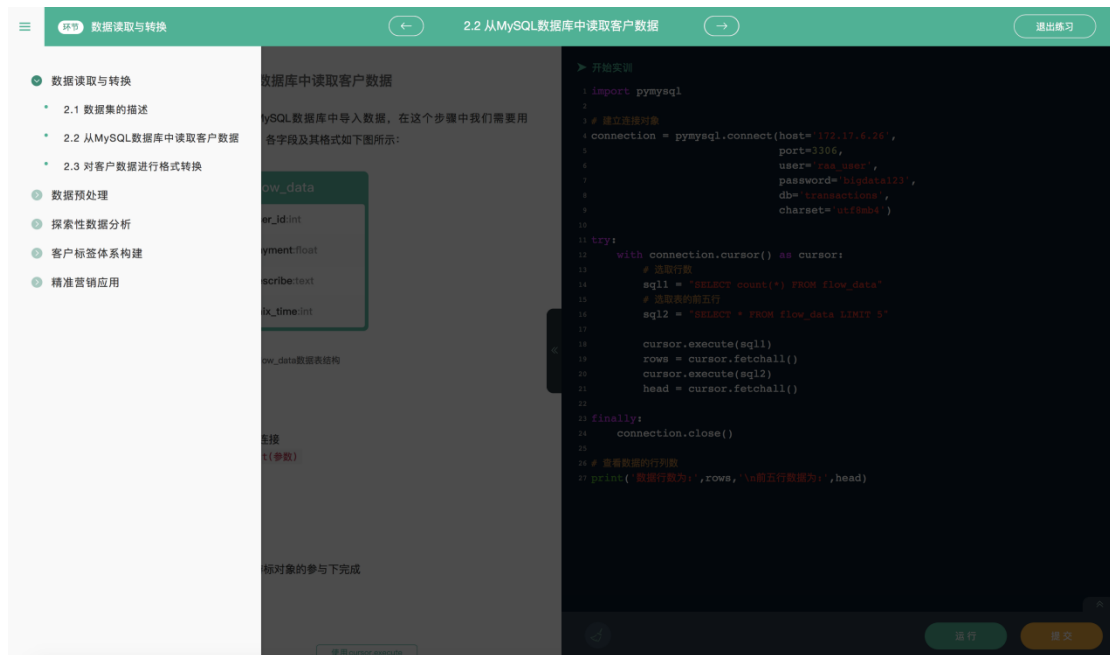
1 import pymysql
2
3 # 建立连接对象
4 connection = pymysql.connect(
5     host='localhost',
6     user='root',
7     password='123456',
8     database='cookdata'
9 )
10
11 # 创建游标对象
12 cursor = connection.cursor()
13
14 # 执行SQL语句
15 cursor.execute('select * from flow_data')
16
17 # 获取查询结果
18 rows = cursor.fetchall()
19
20 # 打印查询结果
21 print('查询到的数据为：', rows)
22
23 # 关闭游标对象
24 cursor.close()
25
26 # 关闭数据库连接
27 connection.close()

```

清空已输入代码

运行

提交



练习实训页面左侧为练习内容说明部分；右侧为代码区，代码区可展开至全屏，用户可在该区域运行和提交代码，支持代码补全。运行后可查看运行结果和相关变量，提交后不仅可以查看运行结果和相关变量，还可查看结果对错。



代码区展开模式

数据读取与转换

2.2 从MySQL数据库中读取客户数据

退出练习

2.2 从MySQL数据库中读取客户数据

在本项目中，数据存在MySQL中，首先从MySQL数据库导入数据，在这个步骤中我们需要用到 `pymysql` 模块。数据表名称为 `flow_data`，各字段及其格式如下图所示：

flow_data

user_id:int
payment:float
describe:text
unix_time:int

图2.1: flow_data数据表结构

数据库连接对象 `connection`

- 连接对象：建立Python客户端与数据库的连接
- 创建方法：`connection=pymysql.connect(参数)`

数据库游标对象 `cursor`

- 游标对象：用于执行查询和获取结果
- 创建方法：`cursor = conn.cursor()`
- 在MySQL中，所有sql语句的执行都要在游标对象的参与下完成

开始实训

```

1 import pymysql
2
3 # 建立连接对象
4 connection = pymysql.connect(host='172.17.6.26',
5                               port=3306,
6                               user='raa_user',
7                               password='bigdata123',
8                               db='transactions',
9                               charset='utf8mb4')
10
11 try:
12     with connection.cursor() as cursor:
13         # 选取行数
14         sql1 = "SELECT count(*) FROM flow_data"
15         # 选取表的前五行
16         sql2 = "SELECT * FROM flow_data LIMIT 5"
17
18         cursor.execute(sql1)
19         rows = cursor.fetchall()
20         cursor.execute(sql2)
21         head = cursor.fetchall()
22

```

结果输出

数据行数为: (3672588,1)
前五行数据为: ((2171955, '6500', '湖州天虹百货有限公司', '1509379200'), (2171955, '-2096', '支付宝 - 中国铁路总公司资金清算中心', '1509120000'), (2171955, '22450', '湖州市星火服装有限公司', '1509379200'), (2171955, '20900', '湖州市星火服装有限公司', '1509379200'), (2171955, '2340', '吴兴晓华化妆品商行', '1509379200'))

变量描述

- connection
- cursor

运行 提交

代码运行效果

数据读取与转换

2.2 从MySQL数据库中读取客户数据

退出练习

2.2 从MySQL数据库中读取客户数据

在本项目中，数据存在MySQL中，首先从MySQL数据库导入数据，在这个步骤中我们需要用到 `pymysql` 模块。数据表名称为 `flow_data`，各字段及其格式如下图所示：

flow_data

user_id:int
payment:float
describe:text
unix_time:int

图2.1: flow_data数据表结构

数据库连接对象 `connection`

- 连接对象：建立Python客户端与数据库的连接
- 创建方法：`connection=pymysql.connect(参数)`

数据库游标对象 `cursor`

- 游标对象：用于执行查询和获取结果
- 创建方法：`cursor = conn.cursor()`
- 在MySQL中，所有sql语句的执行都要在游标对象的参与下完成

开始实训

```

1 import pymysql
2
3 # 建立连接对象
4 connection = pymysql.connect(host='172.17.6.26',
5                               port=3306,
6                               user='raa_user',
7                               password='bigdata123',
8                               db='transactions',
9                               charset='utf8mb4')
10
11 try:
12     with connection.cursor() as cursor:
13         # 选取行数
14         sql1 = "SELECT count(*) FROM flow_data"
15         # 选取表的前五行
16         sql2 = "SELECT * FROM flow_data LIMIT 5"
17
18         cursor.execute(sql1)
19         rows = cursor.fetchall()
20         cursor.execute(sql2)
21         head = cursor.fetchall()
22

```

结果输出

数据行数为: (3672588,1)
前五行数据为: ((2171955, '6500', '湖州天虹百货有限公司', '1509379200'), (2171955, '-2096', '支付宝 - 中国铁路总公司资金清算中心', '1509120000'), (2171955, '22450', '湖州市星火服装有限公司', '1509379200'), (2171955, '20900', '湖州市星火服装有限公司', '1509379200'), (2171955, '2340', '吴兴晓华化妆品商行', '1509379200'))

变量描述

- connection
- cursor

运行 提交

代码提交效果（答案正确）

数据读取与转换

2.2 从MySQL数据库中读取客户数据

退出练习

2.2 从MySQL数据库中读取客户数据

在本项目中，数据存在MySQL中，首先从MySQL数据库导入数据，在这个步骤中我们需要用到 `pymysql` 模块。数据表名称为 `flow_data`，各字段及其格式如下图所示：

flow_data	
user_id	int
payment	float
describe	text
unix_time	int

图2.1: flow_data数据表结构

数据库连接对象 `connection`

- 连接对象：建立Python客户端与数据库的连接
- 创建方法：`connection=pymysql.connect(参数)`

数据库游标对象 `cursor`

- 游标对象：用于执行查询和获取结果
- 创建方法：`cursor = conn.cursor()`
- 在MySQL中，所有sql语句的执行都要在游标对象的参与下完成

开始实训

```

1 import pymysql
2
3 # 建立连接对象
4 connection = ____
5
6 try:
7     with connection.cursor() as cursor:
8         # 查询数据表行数
9         sql1 = "____"
10        # 选择表的前五行
11        sql2 = "____"
12
13        cursor.execute(sql1)
14        rows = cursor.fetchall()
15        cursor.execute(sql2)
16        head = cursor.fetchall()
17
18 finally:
19     connection.close()

```

结果输出

```

Traceback (most recent call last) in 0
2
3 # 建立连接对象
----> 4 connection = ____
5
6 try:
NameError: name '____' is not defined

```

变量描述

reload

运行 提交

代码提交效果（答案错误）

3.1.4 报告

生成报告页操作

用户点击左侧导航栏“报告”，默认进入“+生成报告”，用户可在该页面填写、生成报告。

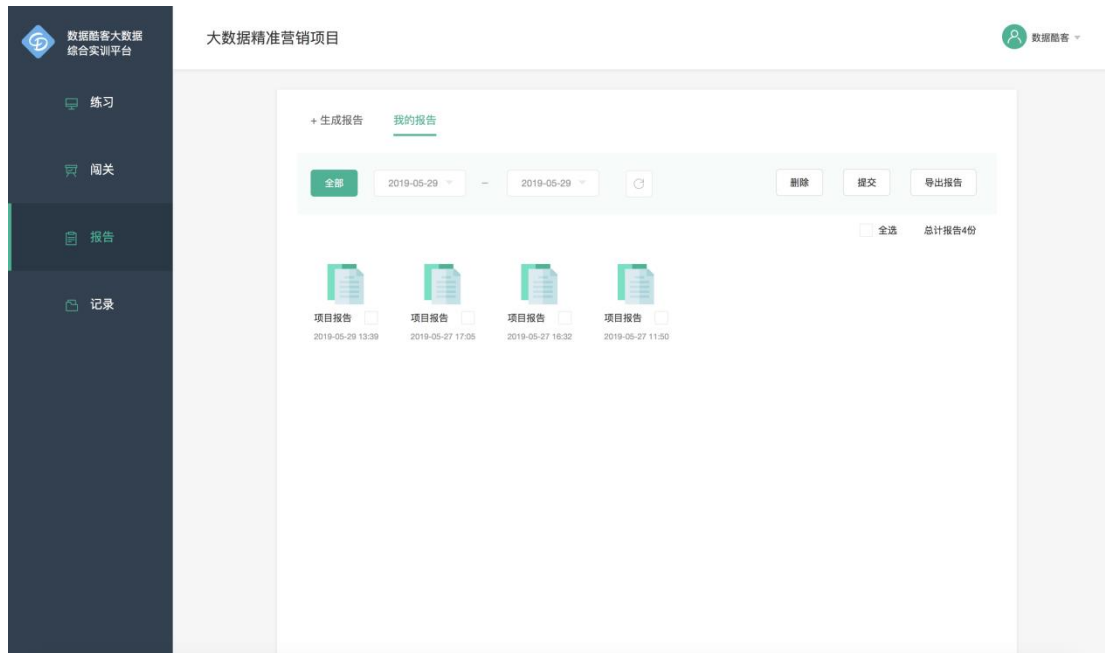
报告内容下方会显示报告完成进度；点击页面左下“预览”按钮预览报告；点击页面右下“保存草稿”可将报告存为草稿，被存为草稿的报告可在下次打开时继续编辑；点击页面右下方“完成”按钮报告会在“我的报告”标签下生成一份报告。



我的报告页操作

点击左侧导航内“报告”，再点击“我的报告”进入我的报告页面，用户可以在该页查看已经生成的报告列表，列表按时间降序排序。

点击“全部”按钮，可查看全部报告，也可以选择筛选某一时间段内生成的报告。勾选需要管理的报告（如需勾选所有报告，可以点击“全选”按钮选择），点击“导出报告”可以将被勾选的报告导出到本地；点击“删除”可以将被勾选的报告删除，删除需要二次确认；点击“提交”，再在弹窗中选择提交到的组（每次只可提交一份报告，支持多个组提交，可按教师名、组名、组 ID 搜索组），若选择了组内已存在提交过的报告，则会覆盖上一份报告；双击报告名称可对报告重命名。




点击报告，可预览该报告，在预览页点击“导出报告”可将报告导出至本地，点击“重新编辑”可进入我的报告编辑页，编辑后点击“完成”则覆盖我的报告中的原报告内容，点击“取消”则不保存二次编辑内容并退出，报告提交后编辑，成功提交的报告不做更新。




预览报告

3.1.5 记录

用户点击左侧“记录”导航，默认选择“提交记录”选项卡，选项卡下可以查看报告的提交记录，点击列表中的“查看”可查看对应报告的详情，已被删除的报告无法查看；点击列表中的“撤回”可撤回前5分钟内提交的报告，一旦超时不可被撤回，已被撤回的报告在列表中会变为“已撤回”状态。


 数据酷客大数据综合实训平台

大数据精准营销项目

 王小明

练习
闯关
报告
记录

提交记录

我的笔记

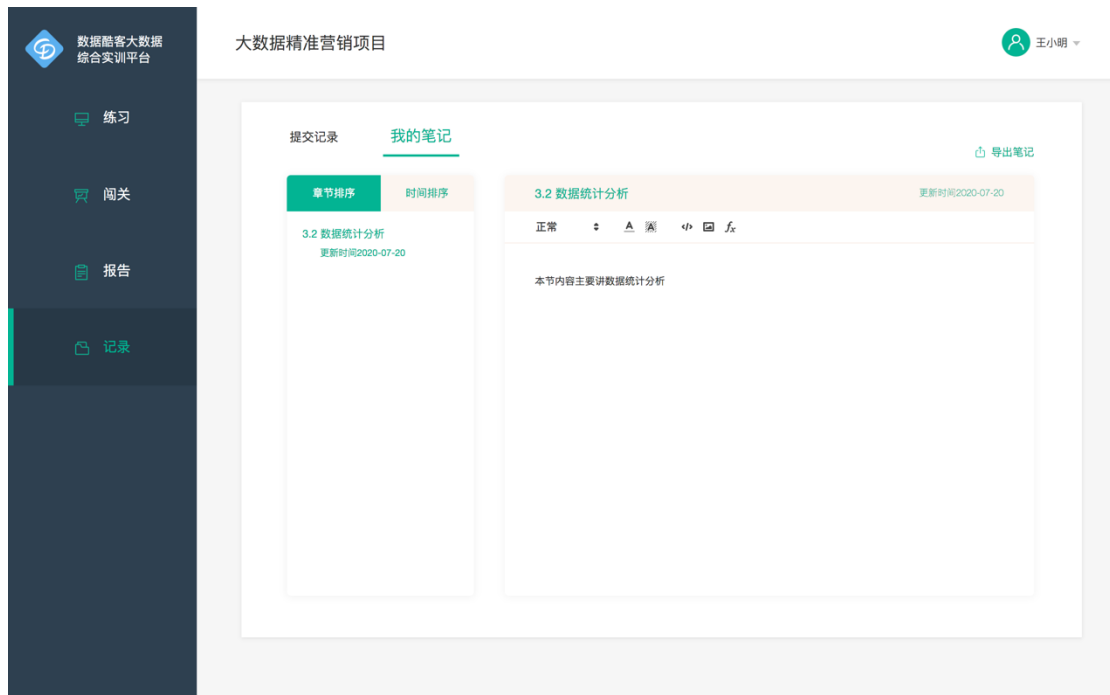
提交5分钟内可以撤回

ID	组名	老师	报告名称	提交时间	操作
29	特特的组	王小明	项目报告 2019-06-03 14:57	2019-06-03 15:00	查看 撤回
90	实验班-01	成成	项目报告 2019-06-03 14:57	2019-06-03 14:58	查看 撤回
88	实验班-01	成成	项目报告 2019-05-21 15:16	2019-05-31 14:29	已删除 撤回
66	8978	雨小风	项目报告 2019-05-06 14:07	2019-05-21 15:17	查看 撤回
60	1	cli	项目报告 2019-05-21 15:16	2019-05-21 15:17	已删除 撤回
64	222	cli	项目报告 2019-05-21 15:16	2019-05-21 15:17	已删除 撤回
59	9	cli	项目报告 2019-05-06 14:07	2019-05-20 11:44	查看 撤回
63	111	cli	项目报告 2019-05-06 14:07	2019-05-20 11:44	查看 撤回
27	111	cli	项目报告 2019-05-08 11:08	2019-05-17 15:13	查看 撤回
28	我行！我上！	何 玛	项目报告 2019-05-08 11:08	2019-05-17 15:13	查看 撤回
30	请输入组名称（长度不大于20个字符）	一只小可爱	项目报告 2019-05-08 11:08	2019-05-17 15:13	查看 撤回
31	组1	一只小可爱	项目报告 2019-05-08 11:08	2019-05-17 15:13	查看 撤回

[上一页](#)
1
2
3
[下一页](#)

在“记录”中选择“我的笔记”选项卡，可查看闯关时的笔记汇总。笔记汇总左侧为笔记列表，可按章节、时间排序；右侧为选定笔记的内容详情。用户可利用笔记编辑器编辑内容，自定义文字字号、颜色、背景颜色，插入代码、图片、公式。

点击“导出笔记”将笔记导出到本地。



4 联系方式

地址：北京市海淀区北四环西路 67 号中关村国际创新大厦 602-603

联系电话：010-82690160

联系邮箱：cooperation@boyabigdata.cn