BiBDR | 博雅大数据学院 BOYA BIG DATA COLLEGE | 数据酷客 COOKDATA

北京市大数据人才培训示范基地

数据科学“云实训”项目训练营

第二课：精准营销数据预处理

讲师：高扬

数据酷客

数据科学人工智能

时间安排

7/21 第一课 数据科学精准营销项目介绍
7/21 晚8点 时长：1.5h

7/23 第二课 精准营销数据预处理
7/23 晚8点 时长：1.5h

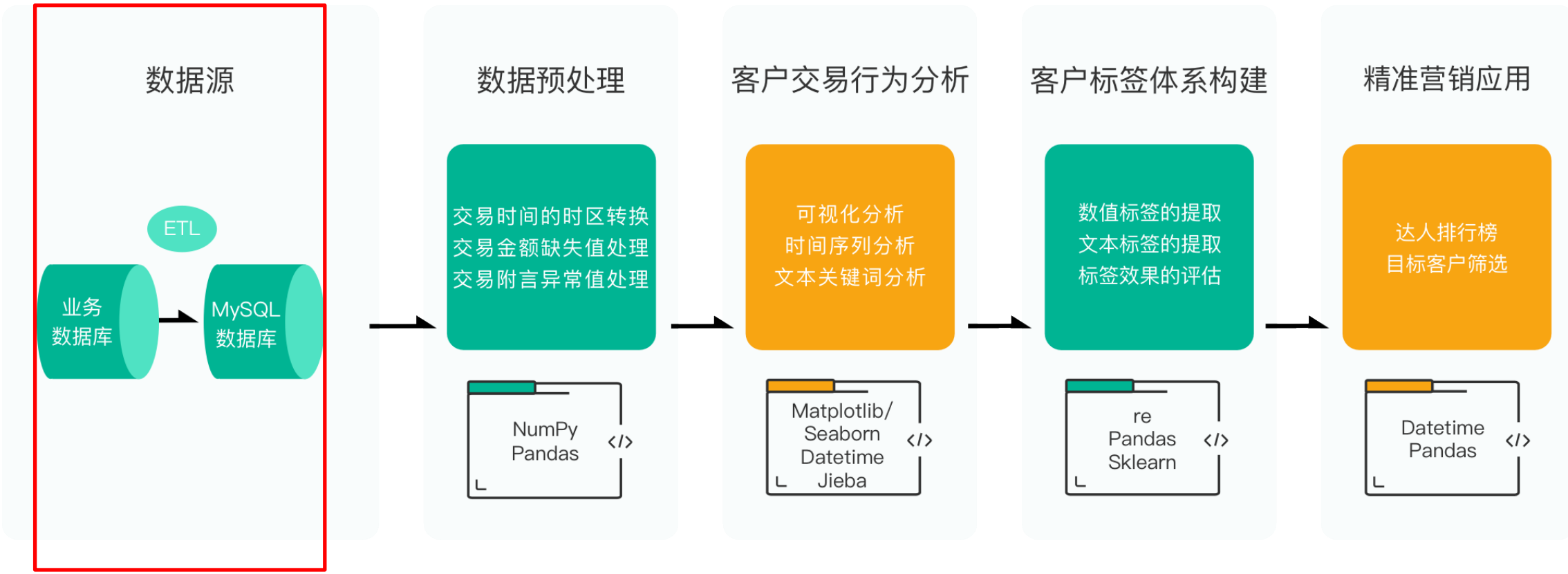
7/28 第三课 客户交易行为分析
7/28 晚8点 时长：1.5h

7/30 第四课 事实类标签与规则类标签构建
7/30 晚8点 时长：1.5h

8/4 第五课 预测类标签与文本类标签构建
8/4 晚8点 时长：1.5h

8/6 第六课 典型客户分析与精准营销应用
8/6 晚8点 时长：1.5h

项目流程


数据源
ETL
业务
数据库
MySQL
数据库
数据预处理
交易时间的时区转换
交易金额缺失值处理
交易附言异常值处理
NumPy
Pandas
客户交易行为分析
可视化分析
时间序列分析
文本关键词分析
Matplotlib/
Seaborn
Datetime
Jieba
客户标签体系构建
数值标签的提取
文本标签的提取
标签效果的评估
re
Pandas
Sklearn
精准营销应用
达人排行榜
目标客户筛选
Datetime
Pandas

目录

本部分主要包括以下三个步骤：

- 数据描述：字段中英文、取值范围和备注信息
- 数据调用：MySQL数据库的连接和数据提取
- 数据解析：将数据格式转换为DataFrame

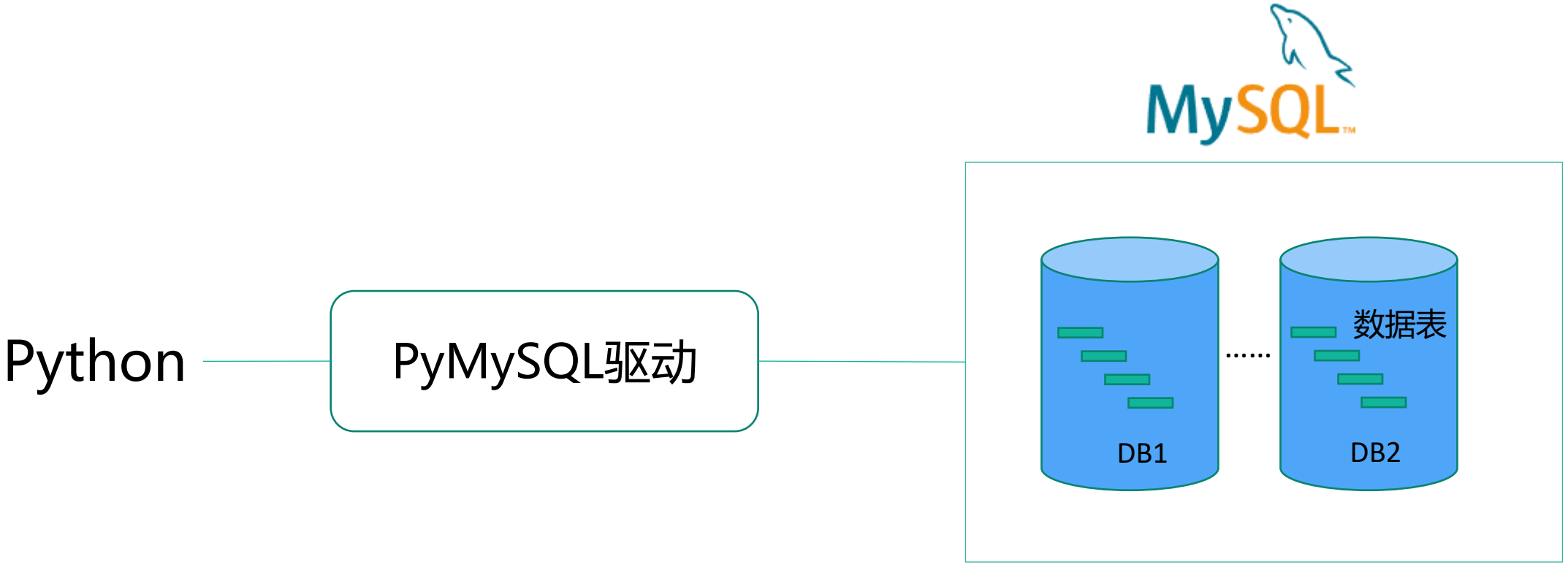
主要涉及PyMySQL、Pandas等模块的基本使用

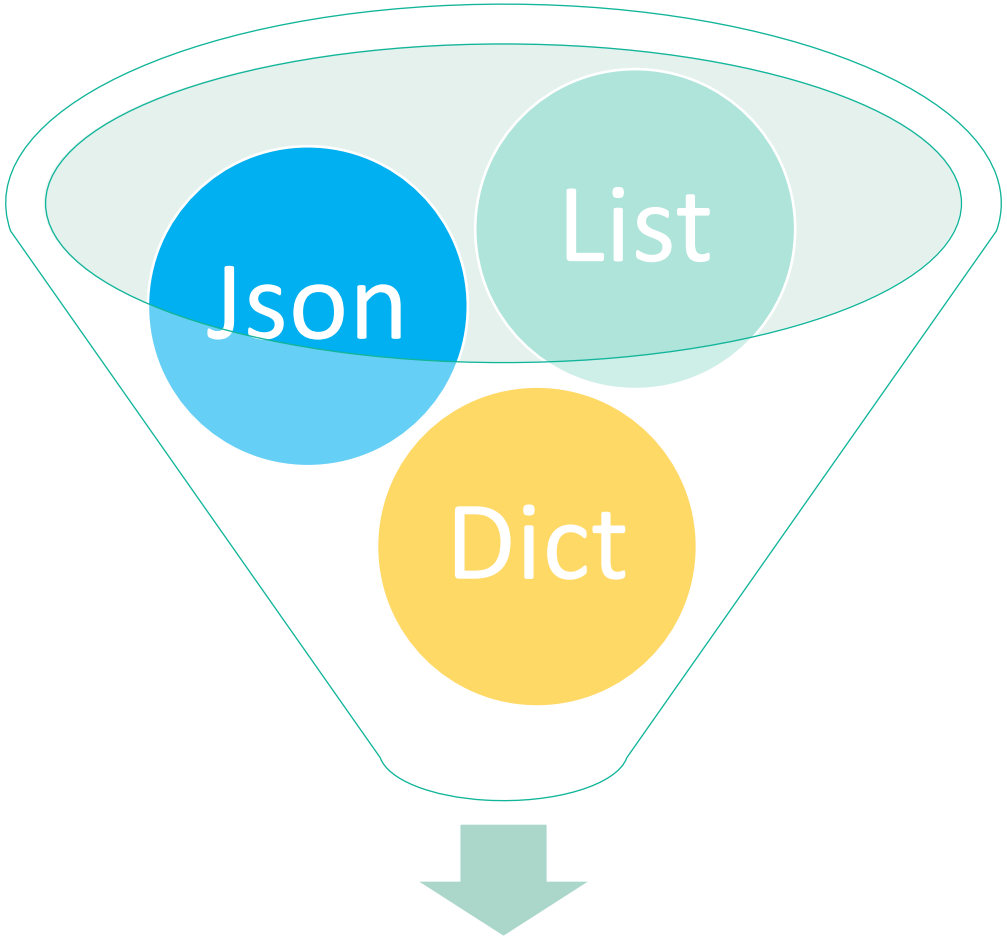
数据调用

MySQL是开源的关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司

- 性能强劲，支持大型数据库，单表可容纳5000万条记录
- 使用标准SQL语言
- 分为社区版与企业版

Python连接MySQL数据库





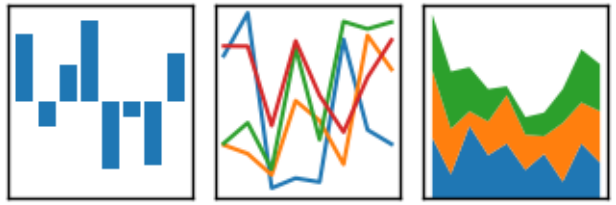
Pandas：DataFrame

Pandas

- Pandas是一个开源的、专注于数据分析的Python第三方库
- Pandas提供了高效、简单易用的数据结构和数据分析工具

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



A Fiscally Sponsored Project of NUMFOCUS

OPEN CODE = BETTER SCIENCE

为什么使用Pandas做数据预处理?

- Python对于数据预处理和数据再加工非常的友好，但是缺少了一些数据预处理和清洗部分
- Pandas使得可以只使用Python完成完整的数据清洗流程，并且不用依靠其他的特定领域的语言
- 使用非常广泛，功能强大，得到很多公司和个人的认可

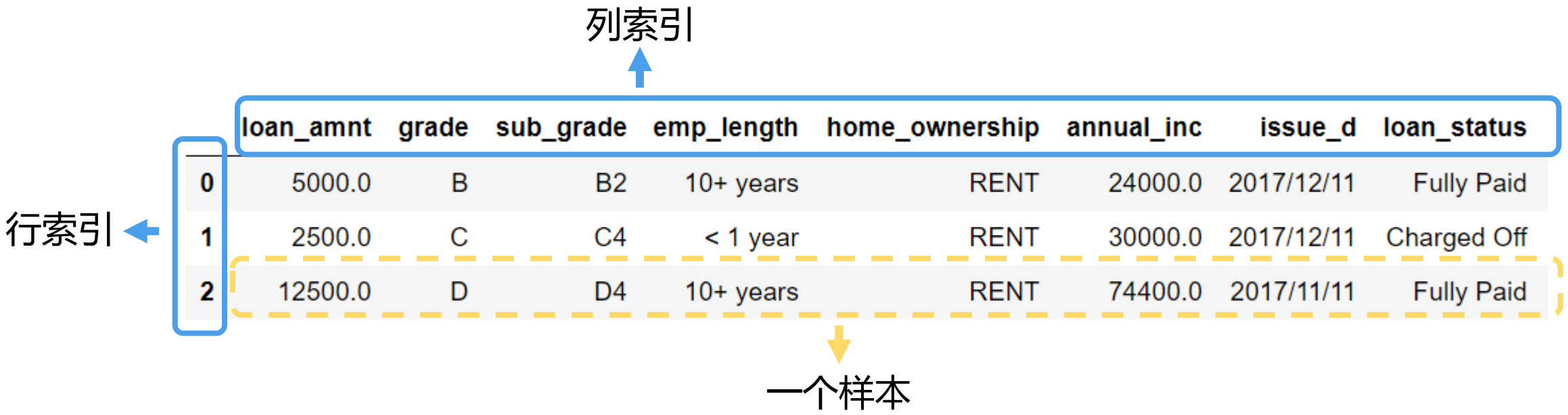
AQR CAPITAL MANAGEMENT

DATADOG

appnexus

DataFrame结构

- DataFrame数据结构是一种二维的结构，可以将它想象成一个Excel工作表
- 如一个DataFrame对象中的前三行数据



	loan_amnt	grade	sub_grade	emp_length	home_ownership	annual_inc	issue_d	loan_status
0	5000.0	B	B2	10+ years	RENT	24000.0	2017/12/11	Fully Paid
1	2500.0	C	C4	< 1 year	RENT	30000.0	2017/12/11	Charged Off
2	12500.0	D	D4	10+ years	RENT	74400.0	2017/11/11	Fully Paid

DataFrame对象创建

pd.DataFrame(data, index, columns)

data允许输入以下一些格式：

- 包含列表，字典或Series的字典
- 二维数组
- 一个Series对象
- 另一个DataFrame对象

Series结构

- Series是带索引的一维数组
- Series对象的两个重要的属性：index（索引）和values(数据值)
- DataFrame的任意一行或者一列就是一个Series对象

DaraFrame数据切片

- 与Numpy中的数组相比，Series和DataFrame由于存在行索引和列索引，并且索引的方式不再只有0，1，…的整数索引的方式，所以它们索引数据更高效和便利。

- 通过索引进行数据切片：

操作	语法	结果
选取某一列	df[label]	Series
通过标签选取某一行/列	df.loc[label]	Series
通过位置（整数表示）获取某一行/列	df.iloc[loc]	Series
通过切片方式获取多行/列	df[5:10]	DataFrame
通过布尔向量获取多行/列	df[bool_vec]	DataFrame

使用布尔向量获取多行/列

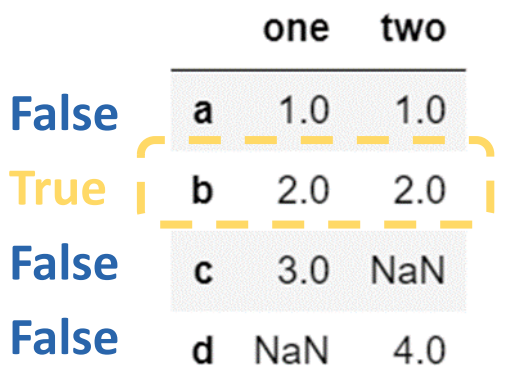
- 在实际应用中，经常使用布尔向量进行数据切片，即筛选满足一定条件的数据。

```
df['one'] == 2.0
```

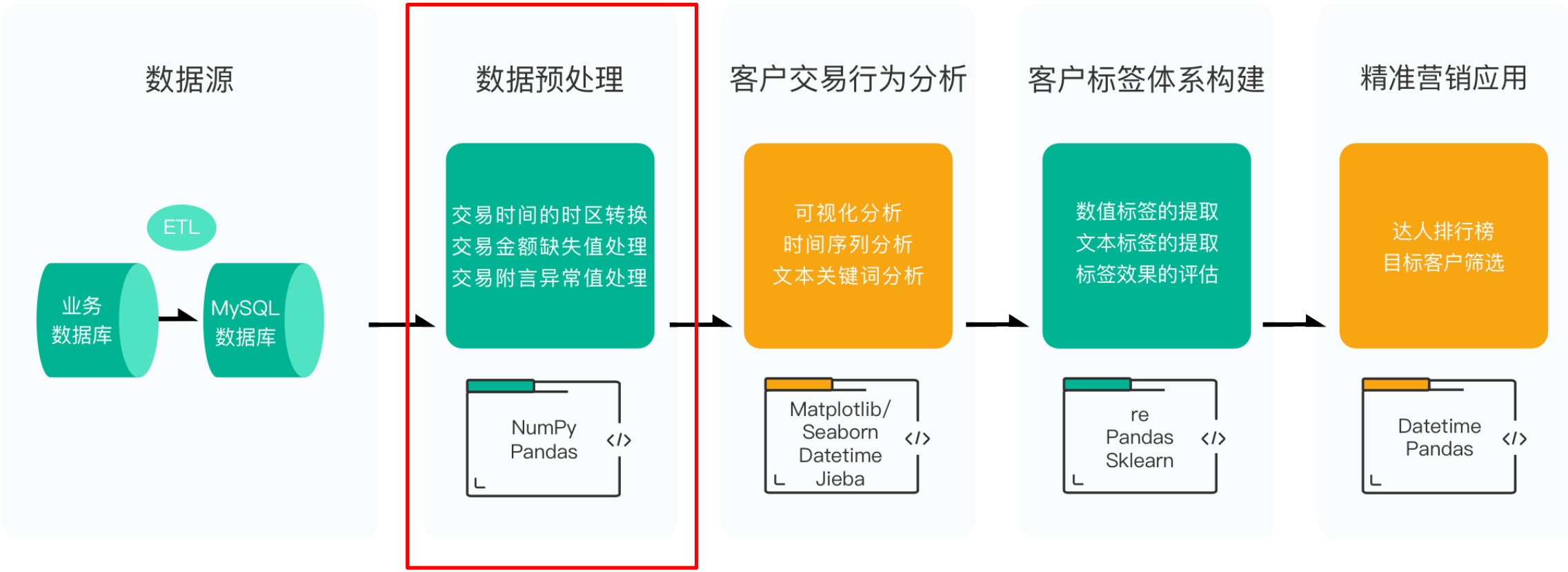
```
a    False
b     True
c    False
d    False
Name: one, dtype: bool
```

```
df[df['one'] == 2.0]
```

	one	two
b	2.0	2.0



项目流程


数据源
ETL
业务
数据库
MySQL
数据库
数据预处理
交易时间的时区转换
交易金额缺失值处理
交易附言异常值处理
NumPy
Pandas
客户交易行为分析
可视化分析
时间序列分析
文本关键词分析
Matplotlib/
Seaborn
Datetime
Jieba
客户标签体系构建
数值标签的提取
文本标签的提取
标签效果的评估
re
Pandas
Sklearn
精准营销应用
达人排行榜
目标客户筛选
Datetime
Pandas

本部分主要包括以下五个步骤：

- 统计分析：对数据进行统计分析，初步了解数据特点。
- 异常值处理：对交易时间等字段中出现的异常数据进行诊断，并确定异常值处理方法。
- 缺失值处理：对于存在缺失值的交易金额和交易附言字段，诊断缺失值产生的原因，确定缺失值处理方法。
- 数据格式转换：为了便于后续分析，对于金额字段的量纲、交易时间字段的时间格式进行转换。
- 重复数据过滤：检测交易数据中存在的重复交易记录，并删除重复的记录。

主要涉及NumPy、Pandas等模块的基本使用

目录

常用函数

DataFrame.shape：

- 查看DataFrame的行列数

DataFrame.describe()：

- 查看DataFrame各列的统计指标（如均值、最大值、最小值等）

DataFrame.info()：

- 查看DataFrame各列的基本信息（如数据类型、是否有空值等）

DataFrame.head()/tail()/sample()：

- 查看DataFrame的前五行、后五行和随机一行
- 括号中可填入数字代表取出多少行

统计分析

user_id	payment	describe	unix_time
2099234	200000	支付宝（95188）	1487088000
22677084	-27500	北京京东三佰陆拾度电子商务公司	1510329600
17775403	5000000	银联POS消费	1503849600
8960118	6600	贷记卡转账还款	1496332800
14042200	-400000	转帐存入　厦门建行会计部清算	1473342861

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。

```
RangeIndex: 3672588 entries, 0 to 3672587
Data columns (total 4 columns):
user_id          3672588 non-null int64
payment          3672588 non-null object
describe         3640295 non-null object
unix_time        3672588 non-null object
dtypes: int64(1), object(3)
memory usage: 112.1+ MB
```

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。

目录

正则表达式

正则表达式(Regular Expression)检索、替换符合某种模式(规则)的字符串

- 正则表达式是由普通字符(例如字符 a 到 z)以及特殊字符(称为元字符)组成的文字模式
- 正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配

正则表达式

在正则表达式中，如果直接给出字符，就是精确匹配

用\d可以匹配一个数字，\w可以匹配一个字母或数字，所以：

- '00\d'可以匹配'007'，但无法匹配'00A'
- '\d\d\d'可以匹配'010'
- '\w\w\d'可以匹配'010'

.可以匹配任意字符，所以：

- 'py.'可以匹配'pyc'、'pyo'、'py!'等等

正则表达式

要匹配变长的字符，用*表示0个或者多个字符，用+表示至少一个字符，
用?表示0个或1个字符，用{n}表示n个字符，用{n,m}表示n-m个字符

- 正则表达式语法示例：\d{3}\s+\d{3,8}.
- ^表示行的开头，^\d表示必须以数字开头
- $表示行的结束，\d$表示必须以数字结束

- \d{3}表示匹配3个数字，例如'010'
- \s可以匹配一个空格（也包括Tab等空白符），所以\s+表示至少有一个空格
- \d{3,8}表示3-8个数字，例如'1234567’

综合起来，上面的正则表达式可以匹配以任意个空格隔开的带区号的电话号码

交易时间异常值处理

- Unix时间戳是指格林尼治时间1970年01月01日00时00分00秒起至现在的总秒数。
- Unix时间戳通常为10位数字(如果精确到毫秒为13位)，我们使用正则表达式对数据进行匹配，发现有一些异常值。

user_id	payment	describe	unix_time
6729440	-4540	NaN	0
27164939	-2580	支付宝网络还款	14 3264000

- 对于数字缺失的情况，我们根据上下文的时间对其进行了填补；
- 对于时间戳的值是0的情况，我们进一步观察这些交易记录，发现这些记录交易附言也为空。

交易金额异常值处理

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

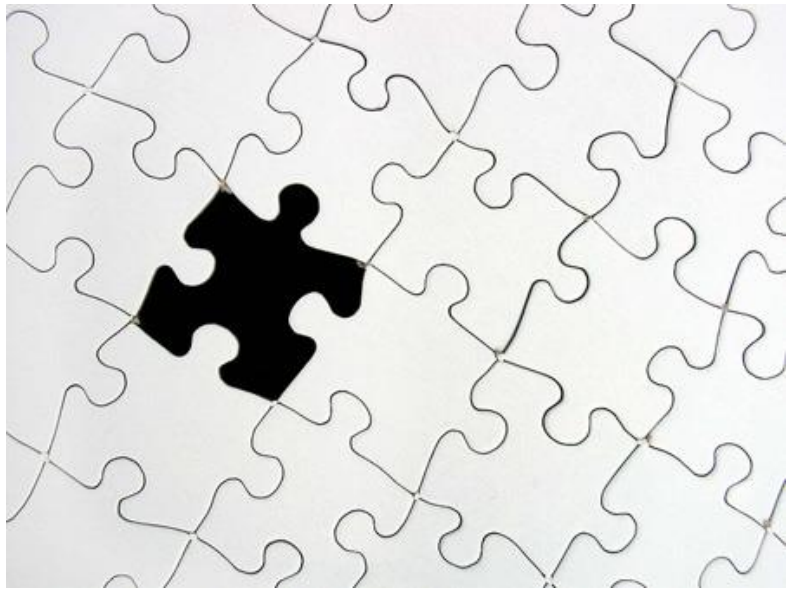
- 我们发现存在一些行的交易金额为\N，这是数据中的一个特殊字符，可以把它视为空
- 对于这些交易金额异常的行，我们无法进行后续的分析，对这些数据我们直接将其删除

目录

什么是数据缺失?

指某一数据集由于信息缺失等主客观因素
导致的字段记录的缺失，遗漏等现象

本项目中的交易附言字段存在数据缺失的情况



交易附言数据缺失

user_id	payment	describe	unix_time
17621220	-500	NaN	1483702743
17621220	-2900	NaN	1489588488
17621220	-200	NaN	1508509279
17621220	-12	NaN	1508825428

- 可以看到交易记录中存在交易附言为空的行
- 对于这些行，其交易金额与交易时间都是完整的，这部分数据可以描绘一些客户的消费习惯，

我们不进行删除

目录

时间格式转换与时区转换

- 对时间格式进行转换，将时间戳转换成“年-月-日 时:分:秒”的格式
- 对时区进行转换，将格林威治时间转换为北京时间

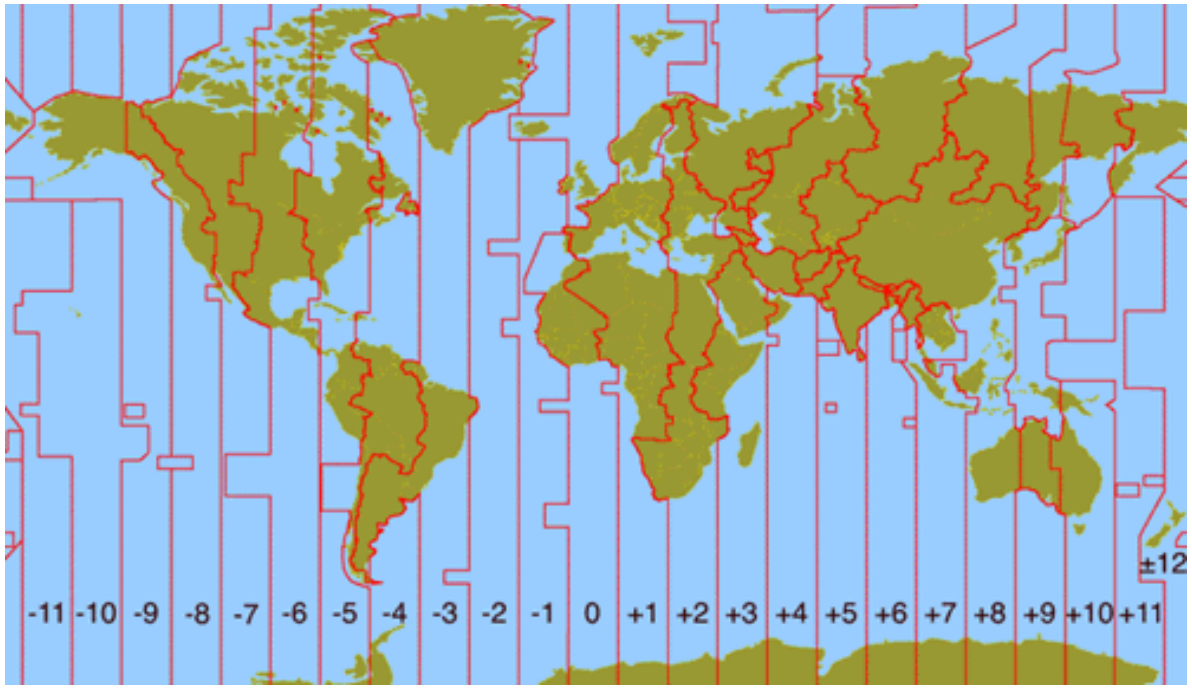
Pandas的to_datetime()函数可以将unix时间戳转换为标准时间格式，语法如下：

pandas.to_datetime(arg,unit = None)

- arg：需要转换的时间
- unit：时间的最小表示单位(D,s,ms,us,ns)，例如unit ='ms'代表时间以毫秒表示

时间格式转换与时区转换

- 由于世界各国家与地区经度不同，会划分为不同的时区。 每隔经度15°划分一个时区，一共24个时区，相邻时区的时间相差1小时
- 格林威治时间为0时区，北京为东八区，转换为北京时间需要加8个小时



交易金额量纲转换

user_id	payment	describe	unix_time	pay_time
22171955	6500	湖州天虹百货有限公司	1509379200	2017/10/31 0:00
22171955	-2096	支付宝 - 中国铁路总公司资金清算中心	1509120000	2017/10/28 0:00
22171955	22450	湖州市星火服装有限公司	1509379200	2017/10/31 0:00
22171955	20900	湖州市星火服装有限公司	1509379200	2017/10/31 0:00
22171955	2340	吴兴晓华化妆品商行	1509379200	2017/10/31 0:00

交易金额(payment)字段以分为单位，为了符合我们的观察习惯，我们将其量纲改为元

目录

重复数据检测

DataFrame.duplicated(subset=None, keep='first')

- subset：默认判定整行是否重复。可以选择固定列，即判定某些列是否重复。
- keep：可以为first和last，表示选择最前一项保留还是最后一项保留，默认为first。

其中无重复值的行标记为False，有重复值的行标记为True。

user_id	payment	describe	unix_time	pay_time
1575826	1000	易宝支付	1509984000	2017/11/7 0:00
1575826	1000	易宝支付	1509984000	2017/11/7 0:00
1575826	1000	易宝支付	1509984000	2017/11/7 0:00
1575826	1000	易宝支付	1509984000	2017/11/7 0:00

重复数据过滤

DataFrame的drop_duplicates()函数可以删除重复值，

DataFrame.drop_duplicates(subset=None, keep='first', inplace=False)

- subset：默认判定整行是否重复。可以选择固定列，即判定某些列是否重复
- keep：可以为first和last，表示选择最前一项保留还是最后一项保留，默认为first
- inplace：是否在原数据上进行更改，默认为False

数据酷客

数据科学人工智能

加入数据酷客交流群