内容简介

数据检视主要通过对数据描述性统计和数据分布类型进行判断，进一步提出数据改造方案。

1. 主要内容

- 生成描述性统计文档及绘制所有数值型字段的分布形状并进行判断；
- 提出数据改造方案。

2. 学习目标

学完本节，能解决以下问题：

- 本案例的与前两个案例的区别之处有哪些？

北京课工场教育科技有限公司

提出数据改造方案

由于“新三板”金融数据分析项目的数据改造方案基本一致，故本次案例只针对不同之处进行分析并提出改造方案，相同部分不做重复叙述。

1. 明显可以放弃的字段（放弃字段均标灰）

部分字段放弃依据表（7 个字段）

英文名称	特征名称	放弃依据
company_code	公司代码	公司标识，不参与数据建模
company_name	公司名称	
trading_days	交易日期	交易反生后字段，不适合用于预测分析
trading_price	交易平均价格	
total_transaction_amount	交易总金额	
financing_amount_wan	融资金额（万）	
financing_cnt	融资次数	

2. 朴素贝叶斯模型对数据的要求

本案例计划使用朴素贝叶斯模型对公司的最终去向进行分类。朴素贝叶斯模型往往用于较小数据集、数据特征少、都是同类型数据的情况。

根据输入数据的类型不同，朴素贝叶斯模型在实现上主要有两种方法：

在特征全部是离散变量且服从多项式分布的情况下，采用**多项式朴素贝叶斯**（MultinomialNB）方法；在特征全部是连续变量且服从正态分布的情况下，采用**高斯朴素贝叶斯**（GaussianNB）方法。

剩余 42 个字段的字段类型如下表所示：

文本型变量（9 个）	离散型变量（7 个）	连续型变量（26 个）		
transfer_mode	maker_num	roa	holder_rt_1st	supplier_rt_5all
layer	employee_num	asset	net_profit_rt	ave_executive_age
high_executive_edu	board_num	asset_gr	holder_rt_10all	earning_per_share
industry	supervisor_num	income_gr	gross_profit_rt	asset_liablity_rt
province	executive_num	liability	business_income	ave_executive_edu
accounting_firm	contain_org_holder	current_rt	customer_rt_1st	registered_capital
local_accounting_firm	outway	listing_days	supplier_rt_1st	total_stock_equity
broker		net_profit_gr	gross_cash_flow	net_asset_per_share
local_broker		register_days	customer_rt_5all	

北京课工场教育科技有限公司

在本次案例中，因为大部分变量是连续变量，所以选择使用 **GaussianNB 方法**，并保留 26 个连续变量作为输入变量，1 个离散变量（new_outway）作为输出变量。

同时，在理想情况下，朴素贝叶斯模型也希望所有的输入变量组都服从标准正态分布。

26 个连续型字段分布情况统计表

分布类型	英文名称	特征名称	分布类型	英文名称	特征名称
近似正态分布（7 个）	customer_rt_5all	五大客户占比		business_income	营业收入
	supplier_rt_5all	五大供应商占比		asset	资产总计
	net_asset_per_share	净资产收益率		liability	负债总计
	earning_per_share	基本每股收益		gross_cash_flow	现金流量总额
	asset_liablity_rt	资产负债率	特殊分布	holder_rt_10all	前十大股东占比
	ave_executive_age	高管平均年龄	其他分布（8 个）	gross_profit_rt	毛利率
	holder_rt_1st	第一大股东占比		net_profit_rt	净利润
近似对数正态分布（10 个）	listing_days	挂牌时长		roa	净资产收益率
	registered_capital	注册资本		current_rt	流动比率
	register_days	注册时长		asset_gr	总资产增长率
	customer_rt_1st	第一大客户占比		income_gr	营业收入增长率
	supplier_rt_1st	第一大供应商占比		net_profit_gr	净利润增长率
	total_stock_equity	总股本		ave_executive_edu	高管平均学历

3. 数据分割

朴素贝叶斯模型也属于有监督学习，因此需要对数据集进行分割，采用直接下定义和交叉验证相结合的方法。

跟前一个案例不同的是，这里我们在搜索超参数的时候只需计算平衡准确率 balanced accuracy score，而这个分数是可以直接通过交叉验证评估得分的函数诸如 cross_val_score()得到。既然有现成的函数可以帮助我们在训练集上实现交叉验证并评估得分，我们就没有必要从训练集中手动切分出验证集了。因此，这里不需要切分验证集，只需要进行训练集和测试集的切分。

本案例除了使用 cross_val_score()进行交叉验证并得到最终准确度外，新增 GridSearchCV 直接实现网格搜索，并直接得到交叉验证的最高分和模型的最优参数。

4. 数据重构

输出变量组为 new_outway 字段，通过调整，取值及样本个数如下所示：

取值	样本个数
留存	4246
摘牌	682
转板	24

取值	样本个数	备注
0	706	非留存
1	4246	留存

摘牌：与挂牌相对应，即股票终止上市；
转板：指标公司自主选择或被强制性地将其股票从一个市场板转到另一个市场板块进行交易。

数据分割后，训练集依然存在非平衡数据的问题，因此需要进行数据重构。imblearn 库提供了 RandomOverSampler()方法，分别对训练集和测试集进行重采样。同时后续模型效果评估时，也可以对比模型在平衡数据集和非平衡数据集上不同表现进行分析。

将数据改造方案中的所有需要完成的操作，以字段为依据在 EXCEL 中进行整理，得到“数据字段处理记录.xlsx”，放置压缩包中，请下载并认真查看思考，完成课程相关作业。

北京课工场教育科技有限公司

数据检视总结

1、本案例的区别之处

（1）主业务问题的区别

输出变量为公司去向（outway）。

（2）选用模型不同导致数据集的要求不同

本案例使用朴素贝叶斯模型对公司去向进行分类预测。

根据输入数据的类型不同，朴素贝叶斯模型及数据分布要求：

多项式朴素贝叶斯（MultinomialNB）：离散型多项式分布；

高斯朴素贝叶斯（GaussianNB）：连续型正态分布。

因此初步选取了案例相关字段 26 个连续型字段

（3）数据分割与数据重构知识扩充

新增 GridSearchCV 交叉验证方法直接实现网格搜索，并直接得到交叉验证的最高分和模型的最优参数；

imblearn 库提供了 RandomOverSampler()方法，分别对训练集和测试集进行重采样。

北京课工场教育科技有限公司

内容简介

本节主要依据数据改造方案完成数据预处理工作。

1. 主要内容

- 提取新特征：类别值；
- 分类特征处理。

2. 学习目标

学完本节，能解决以下问题：

- 如何对 DataFrameMapper 数据转换后的数组绘制数据分布图？

北京课工场教育科技有限公司

数据预处理总结

1、如何对 DataFrameMapper 数据转换后的数组绘制数据分布图？

sklearn 对 pandas 的 DataFrame 类型不支持，必须先用 DataFrame 自带的.value、.as_matrix 之类的方法，将 DataFrame 类型转换成 numpy 的 ndarray 类型，再输入到 sklearn 模块中。DataFrameMapper 是 sklearn_pandas 提供的一个方便的转换接口，省去自己转换数据的过程。

《Python 数据分析中级案例一》课程“数据预处理总结”中详细的介绍了如何使用 DataFrameMapper 将 sklearn 的数据转换方法用在 DataFrame 中。

其中设置了参数 *df_out* 取值为 *False* 时，直接执行转换 *fit_transform(features.copy())* 即可抽取使用于 numpy 数组形式，如图所示：

```
    ], df_out=False)

# 执行数据转换
X = x_mapper.fit_transform(features.copy())
```

```
print(X)
```

```
[[ 2.45928688  2.47064641  1.65801025 ... -0.23515297  0.01385905
   0.64547091]
 [ 2.45516587  0.67350315  1.13910545 ... -0.90553098 -1.13679501
   0.77764447]
 [ 2.43867667  1.24644371  2.08297926 ... -0.25626518 -0.12945283
   0.84843031]
 ...
 [-1.61476613 -1.37911813 -0.53829371 ... -0.94471355 -0.10624496
   0.99980004]
 [-1.6805358   0.87341975  0.4770941  ... -0.50589835 -0.00651502
   0.92941412]
 [-2.11056961  1.7943778   1.45969176 ... -0.50137421  0.08022545
   0.86862627]]
```

但是得到数组格式在绘制 hist()图出现了报错：

```
# 查看处理后的分类特征数据分布
X.hist(figsize=(18, 18), bins=50)
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-28-491d5a01fb67> in <module>()
      1 # 查看处理后的分类特征数据分布
----> 2 X.hist(figsize=(18, 18), bins=50)
      3 plt.show()

AttributeError: 'numpy.ndarray' object has no attribute 'hist'
```

前面介绍过当 df_out 取值 True 时，输出为 DataFrame 格式，这种格式对于绘制

图形是友好的。但是后续数据依然需要的是 ndarray 类型，因此使用 DataFrame 自带的.value 方法抽取具体的值，如案例中的代码：

```
X = X.values
```

注：

df_out 取值为 *True* 时，直接执行转换得到的数据如图所示：

```
    ], df_out=True)

# 执行数据转换
X = x_mapper.fit_transform(features.copy())
```

```
print(X)
```

company_code	listing_days	registered_capital	register_days
430002	2.459287	2.470646	1.658010
430003	2.455166	0.673503	1.139105
430009	2.438677	1.246444	2.082979
430010	2.437936	1.382005	0.926460
430011	2.434462	2.388941	0.959898
430016	2.419386	-0.023720	1.552178
430019	2.413656	-0.195284	0.538696
430021	2.413656	1.832113	1.353971
430022	2.411809	0.452237	1.030812
430024	2.405124	0.771232	1.055074
430028	2.392265	0.648021	0.540207
430029	2.386649	0.117519	0.702857

DataFrameMapper()中参数 df_out 默认取值为 False。

北京课工场教育科技有限公司

内容简介

在处理数据预处理工作后，本节主要进行数据分割和数据重构操作。

1. 主要内容

- 分割数据集并查看数据平衡情况；
- 使用 RandomOverSampler()方法对数据集进行重采样。

2. 学习目标

学完本节，能解决以下问题：

- 如何使用 RandomOverSampler()方法对训练集和测试集进行重采样？

北京课工场教育科技有限公司

数据分割与重构总结

1、使用 RandomOverSampler()方法对训练集和测试集进行重采样

关于重采样和亚采样的原理可参照《Python 数据分析中级案例二》课程中的“数据分割与重构总结”，这里就不做介绍。

RandomOverSampler()是 imblearn 库关于重采样的一个函数，可以通过设置 random_state 参数生成采样器，然后分别对训练集和测试集进行重采样。主要分为以下几步：

首先导入该函数：

```
from imblearn.over_sampling import RandomOverSample
```

其次设置参数：

```
ros = RandomOverSampler(random_state=0)
```

最后分别对训练集和测试集进行重采样:

```
X_train_bal, y_train_bal = ros.fit_resample(X_train, y_train)
X_test_bal, y_test_bal = ros.fit_resample(X_test, y_test)
```

注：

（1）RandomOverSampler

官方文档：http://imbalanced-learn.org/en/stable/generated/imblearn.over_sampling.RandomOverSampler.html#imblearn.over_sampling.RandomOverSampler

参数及默认取值如下：

sampling_strategy:float, str, dict or callable, (default=’auto’)

random_state:int, RandomState instance or None, optional (default=None)

北京课工场教育科技有限公司

return_indices:bool, optional (default=False)

ratio:str, dict, or callable

（2）同时对测试集进行非平衡数据采样，是为了对比最终模型在平衡数据和非平衡数据上的不同表现。

北京课工场教育科技有限公司