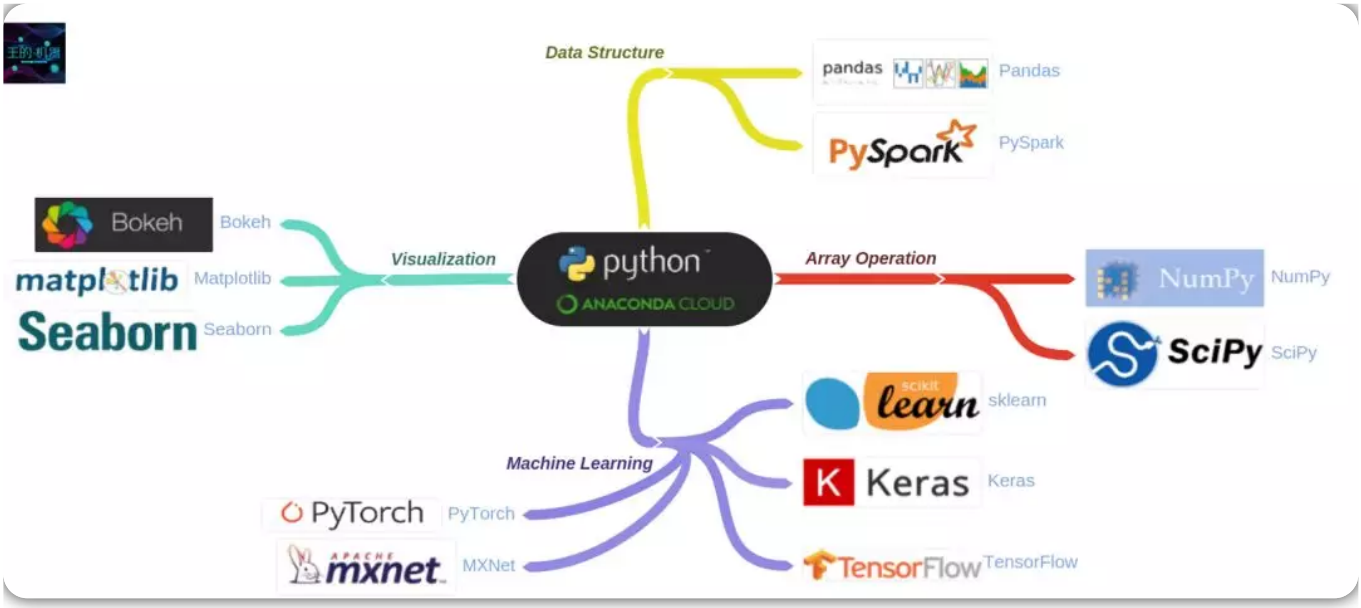
盘一盘 Python 特别篇 16 - Cross Table

原创 王圣元 王的机器 6月17日

来自专辑

Python



本文含 **2573** 字，**16** 图表截屏
建议阅读 **14** 分钟

我的 Python 进阶版课程马上要出了
星期五发推文，敬请关注

本文是 Python 系列的特别篇的第十六篇

- **特别篇 15 - 透视表 Pivot Table**
- **特别篇 16 - 交叉表 Cross Table**

交叉表 (cross table) 是**透视表**的特例，其默认的整合函数是计算**个数**或**频率**。

初探数据

我们拿一个贷款数据举例，首先载入数据，打印出首三行尾两行。

```
loan = pd.read_csv('Loan Data.csv')
loan.head(3).append(loan.tail(2))
```

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1
32579	56	150000	MORTGAGE	5.0	PERSONAL	B	15000	11.48	0
32580	66	42000	RENT	2.0	MEDICAL	B	6475	9.99	0

用 `info()` 函数查阅数据信息，有 32,581 条数据，11 条特征和 1 个标签 (loan_status 那列，0 代表未违约，1 代表违约。)

```
loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32581 entries, 0 to 32580
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   person_age                  32581 non-null  int64
 1   person_income               32581 non-null  int64
 2   person_home_ownership       32581 non-null  object
 3   person_emp_length           31686 non-null  float64
 4   loan_intent                 32581 non-null  object
 5   loan_grade                  32581 non-null  object
 6   loan_amnt                   32581 non-null  int64
 7   loan_int_rate               29465 non-null  float64
 8   loan_status                 32581 non-null  int64
 9   loan_percent_income         32581 non-null  float64
 10  cb_person_default_on_file   32581 non-null  object
 11  cb_person_cred_hist_length  32581 non-null  int64
dtypes: float64(3), int64(5), object(4)
memory usage: 3.0+ MB
```

在机器学习中，我们通常用其他 11 个特征 (或特征转换) 建立模型来预测贷款的良莠。在选择特征前，用交叉表可以做单变量分析，即看看每个特征下的不同特征值对应的“违约”和“不违约”的贷款个数或比例。

按贷款种类统计个数

用交叉表来统计 person_home_ownership 列每个类别 (MORTGAGE, OTHER, OWN, RENT) 下面贷款状态的个数，0 代表未违约，1 代表违约。

```
1  pd.crosstab( index=loan['person_home_ownership'],
2               columns=loan['loan_status'] )
```

loan_status	0	1
person_home_ownership		
MORTGAGE	11754	1690
OTHER	74	33
OWN	2391	193
RENT	11254	5192

从上表可以一下看出 RENT 下面的违约贷款比例很高。

用 `pivot_table()` 函数可以等价实现上面用 `crosstab()` 的产出结果。由于是统计个数，那么整合函数用的是 `len` 。

```
pd.pivot_table( loan, index='person_home_ownership',
                      columns='loan_status',
                      aggfunc={'loan_status':len},
                      fill_value=0 )
```

	loan_status	
loan_status	0	1
person_home_ownership		
MORTGAGE	11754	1690
OTHER	74	33
OWN	2391	193
RENT	11254	5192

按贷款评级统计个数

用交叉表来统计 loan_grade 列每个类别 (从 A 到 G) 下面贷款状态的个数，显示总数 (设置 `margins=True`) 并起名为 Total (设置 `margins_name='Total'`)。

```
pd.crosstab( index=loan['loan_grade'],
             columns=loan['loan_status'],
             margins=True,
             margins_name='Total' )
```

loan_status	0	1	Total
loan_grade			
A	9704	1073	10777
B	8750	1701	10451
C	5119	1339	6458
D	1485	2141	3626
E	343	621	964
F	71	170	241
G	1	63	64
Total	25473	7108	32581

评级越高，违约贷款比例越低，这不正是评级的含义么。

按贷款种类计算利率均值

除了统计个数，交叉表也能做透视表做的事情。下列是在不同的 person_home_ownership 和 loan_status 下计算贷款利率的均值。

```
pd.crosstab( index=loan['person_home_ownership'],
             columns=loan['loan_status'],
             values=loan['loan_int_rate'],
             aggfunc='mean').round(2)
```

loan_status	0	1
person_home_ownership		
MORTGAGE	10.06	13.43
OTHER	11.41	13.56
OWN	10.75	12.24
RENT	10.75	12.97

可以看出，违约贷款的利率都比没有违约贷款的利率高。

没有 fill_value 参数

在 `crosstab()` 函数中没有 `fill_value` 参数，如果结果有 NaN 值，只能紧接一个 `.fillna()` 函数。

```
pd.crosstab( index=loan['person_home_ownership'],
             columns=loan['loan_grade'],
             values=loan['loan_int_rate'],
             aggfunc='mean')
```

loan_grade person_home_ownership	A	B	C	D	E	F	G
MORTGAGE	7.215221	10.967141	13.461798	15.350553	17.077922	18.418043	20.085357
OTHER	8.696250	11.454333	12.921250	15.068750	15.961667	17.465000	NaN
OWN	7.288959	11.073681	13.564615	15.426308	17.251600	18.268000	20.650000
RENT	7.467690	11.001820	13.453823	15.361370	16.957408	18.847143	20.353846

在 OTHER 类下没有评级为 G 的贷款，因此显示 NaN。由于 `crosstab()` 函数返回对象就是一个数据帧 (DataFrame)，那么可以用其下的 `fillna()` 方法将 NaN 用其他值代替，比如下例用 0 值代替 NaN。

```
pd.crosstab( index=loan['person_home_ownership'],
             columns=loan['loan_grade'],
             values=loan['loan_int_rate'],
             aggfunc='mean').fillna(0)
```

loan_grade person_home_ownership	A	B	C	D	E	F	G
MORTGAGE	7.215221	10.967141	13.461798	15.350553	17.077922	18.418043	20.085357
OTHER	8.696250	11.454333	12.921250	15.068750	15.961667	17.465000	0.000000
OWN	7.288959	11.073681	13.564615	15.426308	17.251600	18.268000	20.650000
RENT	7.467690	11.001820	13.453823	15.361370	16.957408	18.847143	20.353846

按贷款目的统计百分比

上面已经展示交叉表的计数功能，如果最终结果想用频率展示的话，可以设置 `normalize` 参数，其中

- `normalized = True` 或者 `all`，在所有元素上做标准化
- `normalized = columns`，在列上做标准化

- `normalized = index`，在行上做标准化

下面在不同的 loan_intent 和 loan_status 下统计贷款状态的百分比。

设置 `normalize= True` **按元素**计算百分比，即所有元素下的百分比加起来等于 100%。

```
pd.crosstab( index=loan['loan_intent'],
             columns=loan['loan_status'],
             normalize=True ).style.format("{:.2%}")
```

loan_status	0	1
loan_intent		
DEBTCONSOLIDATION	11.42%	4.57%
EDUCATION	16.40%	3.41%
HOMEIMPROVEMENT	8.18%	2.89%
MEDICAL	13.66%	4.98%
PERSONAL	13.58%	3.37%
VENTURE	14.95%	2.60%

设置 `normalize=columns` **按列**计算百分比，即在每列的百分比加起来等于 100%。

```
pd.crosstab( index=loan['loan_intent'],
             columns=loan['loan_status'],
             normalize='columns' ).style.format("{:.2%}")
```

loan_status	0	1
loan_intent		
DEBTCONSOLIDATION	14.61%	20.96%
EDUCATION	20.97%	15.63%
HOMEIMPROVEMENT	10.46%	13.24%
MEDICAL	17.47%	22.81%
PERSONAL	17.36%	15.45%
VENTURE	19.13%	11.92%

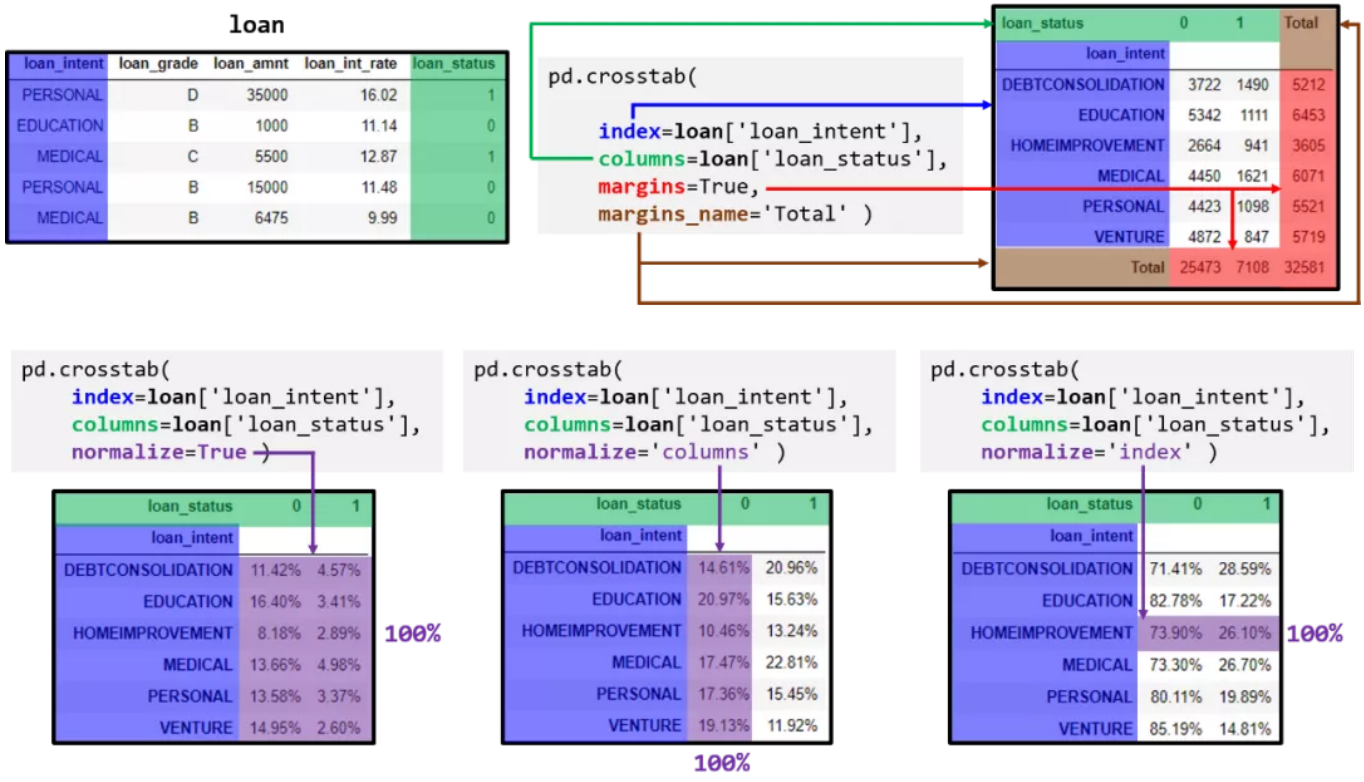
设置 `normalize=index` **按行**计算百分比，即在每行的百分比加起来等于 100%。

```
pd.crosstab( index=loan['loan_intent'],
             columns=loan['loan_status'],
```

```
3                normalize='index' ).style.format("{:.2%}")
```

loan_status	0	1
loan_intent		
DEBTCONSOLIDATION	71.41%	28.59%
EDUCATION	82.78%	17.22%
HOMEIMPROVEMENT	73.90%	26.10%
MEDICAL	73.30%	26.70%
PERSONAL	80.11%	19.89%
VENTURE	85.19%	14.81%

总结，一图胜千言！下图可视化 crosstab() 函数的用法。



Stay Tuned!

END