

pandas 笔记

徐世桐

1 import

```
import pandas
import pandas
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
```

2 使用 csv 数据

```
data = pandas.read_csv('.csv 文件路径')
```

data 为 *pandas.DataFrame* 类型

```
data.head() // 显示前 5 组数据
```

```
data.info() // 显示每一特征的信息 数据类型
```

```
data[' 特征名 ']
```

显示某一特征的所有数据, 输出 *pandas.Series*

```
data[' 特征名 '].value_counts() // 显示此特征所有取值, 对每一取值显示对应样本数, 输出 pandas.Series
```

```
data.describe() // 显示每一特征统计信息, 输出 pandas.DataFrame
```

```
data.hist(BINS, FIGSIZE)
```

```
matplotlib.pyplot.show()
```

将 *data* 中每一特征统计结果用直方图表示

BINS: *bins=N* 直方图将被分为 *N* 个值点, 有 *N - 1* 个区间

FIGSIZE: *figsize=(宽, 高)* 定义每一特征的直方图形状

```
data.iloc[index_array]
```

对 *index_array* 每一 *index* 得到 *data* 中对应位置的样本信息, 输出 *pandas.Series*

```
data.loc[row_array]
```

类似 *iloc*, 但根据行标签进行取样, 非行 *index* 号

```
data_copy = data.copy() // 复制数据
```

```
series.sort_values(ASCENDING)
```

对一个 *pandas.Series* 输出排序后的数据, 输出 *pandas.Series*

ASCENDING: 取 boolean 值, 是否按递增顺序输出

```
data.corr()
```

对所有特征两两求 correlation

输出 *pandas.DataFrame*, 通过 `data.corr()[' 特征名 ']` 得到一个特征关于其他特征的 corr 值

3 csv 绘图

`data.plot(KIND, X, Y, ALPHA*, S*, C*, CMAP*, FIGSIZE*)`

调用后使用 `plt.show()` 显示图像

KIND: 定义图表类型

`kind='scatter'` 描点图

X: `x=' 特征名 '`, Y: `y=' 特征名 '`

定义横纵坐标采用哪一特征下的值

ALPHA: `alpha=0.1` 点填充设为半透明, 使点浓度高处颜色深

S: `s=data[' 特征名 ']` 用点大小表示特征值高低

C: `c=' 特征名 '` 用点颜色表示特征值高低, 和 CMAP 同时使用

CMAP: `cmp=plt.get_cmap('jet')` 使用 plt 内定义的 jet 色谱。通过点颜色表示 C 中选择的特征值高低

FIGSIZE: `figsize=(宽, 高)`

`scatter_matrix(DATA, figsize=(宽, 高))`

同时显示多组散点图

DATA: 为 *pandas.DataFrame*

`=data` 对所有 data 中特征两两画图

`=data[' 特征 1', ' 特征 2', ...]` 选择某些特征两两画图

4 数据操作

`np.c_[a, b, ...]`

创建 numpy.ndarray 类型, 和 python array 不同

a, b, ... 类型相同, shape 相同

pandas.DataFrame 可调 `.shape` `.values` 转 *np.ndarray*

pandas.Series 可调 `.shape` `.values` 转 *np.ndarray*

np.ndarray 可调 `.reshape` `.shape`

5 MNIST 数据集

`mnist = fetch_openml('mnist_784', version=1)` //得到手写字母的训练集

`X = mnist['data'], y = mnist['target']` // 得到 *pandas.DataFrame* 特征集, label 集

`plt.imshow(X.iloc[0].values.reshape(28, 28))` // 绘制第一个图像, 使用 `plt.show()` 显示