工作记录17之pandas

2016-11-08 19:05:04

常用的pandas方法： 1.pandas基础之numpy 打开train.csv，脚本如下： import csv import numpy as np csv_file_object = csv.reader(open('./train.csv', 'rb')) header = csv_file_object.next() data = [] for row in csv_file_object: data.append(row)data = np.array(data) print data 显示结果： [['1' '0' '3' ..., '7.25' '' 'S'] ['2' '1' '1' ..., '71.2833' 'C85' 'C'] ['3' '1' '3' ..., '7.925' '' 'S'] ..., ['889' '0' '3' ..., '23.45' '' 'S'] ['890' '1' '1' ..., '30' 'C148' 'C'] ['891' '0' '3' ..., '7.75' '' 'Q']] print data[0:15, 5] # 打印第5列的前15行 显示结果： array(['22', '38', '26', '35', '35', '', '54', '2', '27', '14', '4', '58', '20', '39', '14'], dtype='|S82') 输出结果的数据类型为numpy.nparray，也就是说，numpy数组的切片仍为一个numpy数组。此时对若想对这组数据求均值，那么使用如下命令： ages_onboard = data[0::,5].astype(np.float) 结果报错ValueError: could not convert string to float: 那么就需要对数据中的缺失值进行处理。 2.pandas的dataframe 使用pandas读取数据的脚本如下： import pandas as pd import numpy as np df = pd.read_csv('/home/train.csv', header = 0) # header means the key of the column,0 means the row 0 df.head(3) 如果需要查看df的每个列的数据类型，可以使用 df.dtypes pandas可以自动识别数据类型，需要更为精确地数据类型可以使用命令 df.info()和df.describe()，其中df.describe()可以列出所有数值列，并给出均值标准差，最大值最小值。 接下来pandas将对数据进行过滤/填充/替换等操作。 3.pandas的数据清洗 数据统计 在pandas中查看数据，命令如下： df['Age'][0:10] 0 22 1 38 2 26 3 35 4 35 5 NaN 6 54 7 2 8 27 9 14 Name: Age, dtype: float64 也可使用这种方法：df.Age[0:10] 它的数据类型为pandas特有的数据序列（Series）,求均值可用：df['Age'].mean() 如果需要观察特定子集，可以这样表达：df[['Sex','Pclass','Age']], df[df['Age']>60]，df[df['Age']>60][['Sex','Pclass','Age','Survived']]等。 计数统计可以使用df.value_counts() 过滤缺失值可以用如下命令： df[df['Age'].isnull()][['Sex', 'Pclass', 'Age']] 这时，所有行中存在缺失值的数据都会略去不显示。 还有一些其他语法也可使用： for i in range(1, 4): print i, len(df[(df['Sex'] == 'male') & (df['Pclass'] == i)]) pandas画图函数使用方法如下： import pylab as p df['Age'].hist() # 直方图 P.show() 在hist中可以设置参数如： df['Age'].dropna().hist(bins=16, range=(0,80), alpha = .5) # 去掉缺失值 P.show() 对数据的基本情况有了解后，就需要将dataframe中的值转化为机器学习可以实用的数据形式。 数据清洗 首先，对于某些列，字符串类型的数据就需要做转换，这里提供几种方法。 在转换过程中，不改动原始列，为数据集加入新列。 pandas中增加列的方法： df['Gender'] = 4 改变Gender列的值，把它的值变成Sex列的首字母大写形式： df['Gender'] = df['Sex'].map(lambda x: x[0].upper()) lambda x是python中内建的匿名函数，x[0]返回任何字符串的首个字符。 这时，再将Gender映射到0,1，可以暂定F = 0， M = 1. df['Gender'] = df['Sex'].map({'female': 0, 'male': 1}).astype(int) 在处理缺失数据时，常用方法是将已知数据的均值填充进缺失值中，根据直方图的倾斜程度也可以将中值作为替代值。结合数据，也可以根据数据类别来确定填入的值。 具体方法如下： 建立一个参照表： median_ages = np.zeros((2,3)) #生成一个array([[0., 0., 0.], [0., 0., 0.]]) 计算数组： for i in range(0, 2): for j in range(0, 3): median_ages[i,j] = df[(df['Gender'] == i) & \ (df['Pclass'] == j+1)] ['Age'].dropna().median() 得到的结果为： array([[35. , 28. , 21.5], [40. , 30. , 25.]]) 得到的值可以填充到数据中，为了区分哪些地方有缺失值，可以新建一列Fill： df['AgeFill'] = df['Age'] 查看Age列值为空的数据： df[df['Age'].isnull()] [['Gender','Pclass','Age','Fill']].head(10) 用medial_ages中的值填充到Fill中： for i in range(0, 2): for j in range(0, 3): df.loc[(df.Age.isnull()) & (df.Gender == i) & (df.Pclass == j+1),\ 'AgeFill'] = median_ages[i,j] 查看一下数据确认填充完毕： df[df['Age'].isnull()][['Gender','Pclass','Age','AgeFill']].head(10) 可以再创建一个记录原始Age缺失的特征： df['AgeIsNull'] = pd.isnull(df.Age).astype(int) 数据特征工程 通过简单数学运算创建特征： df['FamilySize'] = df['SibSp'] + df['Parch'] 还可以通过人为构造特征： df['Age*Class'] = df.AgeFill * df.Pclass # 扩大某些特征的值 通过直方图，可以为构造特征提供思路。 df.dtypes[df.dtypes.map(lambda x: x=='object')] # 显示数据类型为object类型的列，需要转化或者删除 df = df.drop(['name', 'Sex', 'Cabin']) df = df.dropna() # 去掉有缺失值的任意行 pandas.dataframe不能直接使用sklearn包，因此还需要将df转化为numpy数组： train_data = df.values 原文链接 https://www.kaggle.com/c/titanic/details/getting-started-with-python-ii pandasAPI文档 http://pandas.pydata.org/pandas-docs/stable/api.html pandas教程http://pda.readthedocs.io/en/latest/chp5.html