HW4-520

1.

(1) Just use ‘’pandas.read_csv()’’ function.

(2) Use “DataFrame.drop()” function, set “axis=1” to drop column;

Use “replace()” function to replace dictionary{'dlttq':0} with mean;

Use “interpolate()” to interpolate missing value of the column ‘intanq’

```
28    8090.0
29    8105.0
30    8015.0
31    8038.0
32    8280.5
33    8523.0
Name: intanq, dtype: float64
```

(3) data1['niq'], data1['citotalq'] are both Series. Use series[series>15000] = 15000

(4) When use “apply(f)” for def f(x), it separates DataFrame x into some Series. So I preprocess data1, creating temp=data1.iloc[:,[13,14,15,17,18]], which only contains the exact columns we need. Then temp.apply(f)

	acoq	actq	apq	citotalq	cogsq
max	18681	143810	62985	15000	51636
min	3914	32336	5666	3088	7658
secondLarget	15085	130053	49049	14316	45188

(5) Just use “DataFrame.corr()” function

	acoq	actq	apq	citotalq	cogsq
acoq	1.000000	0.604590	0.664430	0.643810	0.706019
actq	0.604590	1.000000	0.906624	0.578143	0.751786
apq	0.664430	0.906624	1.000000	0.693744	0.848097
citotalq	0.643810	0.578143	0.693744	1.000000	0.907289
cogsq	0.706019	0.751786	0.848097	0.907289	1.000000

(6) Use “pd.merge()” function, set on='datadate', how='inner', which means only merge the elements with same datadate value.

(7) First create a dictionary converting “A/B” to integer, then use “DataFrame.map()” function to add column ‘Rate’

(8)(9) sampler = np.random.randint(0, len(Matched), size = 2*len(Matched))

final = Matched.take(sampler)

final.to_csv('HW4.csv')

2.

(1) Scan the whole array line by line, replace “np.inf” with 2**8

```
[[1.00e+00 5.00e-01]
 [2.50e-01 2.56e+02]]
```

(2) The principle is as above

```
[[ 1  1]
 [-1  0]]
```

(3) First initialize a 1-d array with all zeros. Then scan column by column to get the mean value for the column

```
[-3.   0.4  3. ]
```

(4) First reshape() the two array to 4*25, then use “concatenate()” with axis=0

Leetcode 561

To get the max sum of the smaller one in pairs, the best condition is that every pair has two close numbers. Thus, sort the line, every two adjacent numbers form a pair, in which the first one should be min. Finally sum them up.

```
nums = [1,4,3,2]
print(arrayPairSum(nums), '\n')
```

```
4
```

Appendix:

```
import csv

import pandas as pd

import numpy as np

from pandas import Series, DataFrame


def f(x):
    Max = x.max()
    secondLarget = -1000000   # just a very small number
    for ele in x:
        if ele>secondLarget and ele!=Max:
            secondLarget = ele
    return Series([x.max(), x.min(), secondLarget] ,index=['max', 'min',
'secondLarget'])


def replaceInf(arr, num):
    row = arr.shape[0]
```

```
    column = arr.shape[1]
    for i in range(0, row):
        for j in range(0, column):
            if arr[i][j] == np.inf:
                arr[i][j] = num

def replace2(arr):
    row = arr.shape[0]
    column = arr.shape[1]
    for i in range(0, row):
        for j in range(0, column):
            if arr[i][j] > 0:
                arr[i][j] = 1
            if arr[i][j] < 0:
                arr[i][j] = -1

def columnMean(arr, weight):
    row = arr.shape[0]
    column = arr.shape[1]
    ret = np.zeros(column)
    for j in range(0,column):
        for i in range(0,row):
            ret[j] += weight[i]*arr[i][j]
    return ret
```

```
def fuseTwoArray(x1, x2):
   x1r = x1.reshape(4,25)
   x2r = x2.reshape(4,25)
   return np.concatenate((x1r,x2r), axis=0)


#1.1
data1 = pd.read_csv('/home/jiang/Downloads/AAPL BS.csv')
data2 = pd.read_csv('/home/jiang/Downloads/AAPL Ratings.csv')
#1.2
data1 = data1.drop(['aqepsq', 'gdwlamq'], axis = 1)  #axis=1 means column
mean = data1['dlttq'].mean()
data1 = data1.replace({'dlttq':0}, mean)
data1['intanq'] = data1['intanq'].interpolate()
print('#1.2\n', data1['intanq'], '\n')
#1.3 ````````````````````````````````````````````
data1['niq'][data1['niq'] > 15000] = 15000
data1['citotalq'][data1['citotalq'] > 15000] = 15000
#1.4
temp = data1.iloc[:,[13,14,15,17,18]]  #[’acoq’, ’actq’, ’apq’, ’chq’,’citotalq’]
print('#1.4\n',temp.apply(f), '\n')
#1.5
print('#1.5\n',temp.corr(), '\n')
#1.6
```

```
Matched = pd.merge(data1, data2, on='datadate', how='inner')  # only merge corresponding element
print('#1.6\n',Matched)
#1.7
splticrmToRate = {
   'AAA': 0, 'AA+': 1, 'AA': 2, 'AA-': 3, 'A+': 4, 'A': 5, 'A-': 6,
   'BBB+': 7, 'BBB': 8, 'BBB-': 9, 'BB+': 10, 'BB ': 11
   }
Matched['Rate'] = Matched['splticrm'].map(splticrmToRate)
print('#1.7\n',Matched)
#1.8 1.9
sampler = np.random.randint(0, len(Matched), size = 2*len(Matched))
final = Matched.take(sampler)
final.to_csv('HW4.csv')


#2.1
a = np.array([[1, 2],[4,0]])
b = 1/a
replaceInf(b, 2**8)
print('#2.1\n', b, '\n')
#2.2
c = np.array([[1, 2],[-4,0]])
replace2(c)
print('#2.2\n',c, '\n')
```

```
#2.3
data_matrix = np.array([[1,2,3],[-4,0,3]])
weight = np.array([0.2,0.8])
print('#2.3\n',columnMean(data_matrix, weight), '\n')
#2.4
x1 = np.random.rand(100,)
x2 = np.random.rand(100,)
print('#2.4\n',fuseTwoArray(x1,x2), '\n')


#leetcode 561
def arrayPairSum(nums):
    """
    :type nums: List[int]
    :rtype: int
    """
    length = len(nums)
    nums.sort()
    Sum = 0
    i = 0
    while i < length:
       Sum += nums[i]
       i += 2
    return Sum
```

```
nums = [1,4,3,2]
print('leetcode\n',arrayPairSum(nums), '\n')
```