r2knowle_a1q5

January 21, 2023

1 A1

[1]:
```python
# Standard imports
import numpy as np
np.seterr(all='ignore');  # allows floating-point exceptions
import matplotlib.pyplot as plt
```

1.1 Q5: Bank Fraud

1.1.1 Supplied Code (do not edit)

[2]:
```python
def ReceiveTransactions():
    '''
     credit, debit = ReceiveTransactions()

     Returns two 10000x1 single-precision arrays of transactions.
       credit has all positive values
       debit has all negative values
    '''
    N = 10000

    credit = np.float32(abs(np.random.randn(N)))*20000.
    debit = -np.float32(abs(np.random.randn(N)))*20000.

    credit = np.round(credit, decimals=2)
    debit = np.round(debit, decimals=2)

    return [credit, debit]
```

[5]:
```python
def CalculateNet(credit, debit):
    '''
     net = CalculateNet(credit, debit)

     Add together the credits and debits to get the net income.

     Input
       credit:  An array of positive values
       debit:   An array of negative values

     Output
       net:     The result from adding up all the credits and debits
    '''

    N = len(credit)

    scredit = np.sort(credit)
    sdebit = -np.sort(-debit)

    # Method A
    netA = np.sum(credit) + np.sum(debit)


    # Method B
    netB = np.sum(credit)
    for d in sdebit:
        netB += d

    # Method C
    netC = 0.
    for c,d in zip(scredit, sdebit):
        netC += c + d

    net = np.min([netA, netB, netC])

    return net
```

1.1.2 Process a set of transactions

[6]:

```
c, d = ReceiveTransactions()

net_income = CalculateNet(c, d)

print('Net income: '+str(net_income))
```

```
Net income: 8439.76009273529
```

1.1.3 (a)

The most accurate method is method C.

1.1.4 (b)

To prove that method C, lets first discuss method A and B.

In method A we first sum all the credit scores and debit scores. Both the debit and credit scores will be very large numbers which must be rounded. We will then find the sum of the two rounded values to get an inaccurate result

In method B we first find the sum of the credit scores and then add each debit score. The sum of credit values will be a very large number, so while we are subtracting each debit score at a time (which allows us to be very specific and not round) the result will be inaccurate because we rounded the first value.

In method C we add one credit value and one debit value. Since these values are roughly equal in size, the result should be very small which means that we can keep track of it with only minimal rounding. For these reasons method C should be the most accurate.

1.1.5 (c)

CalculateNet gets the smallest estimation out of the following methods, in other words its getting when credit is minimized and when debit is maximized. This is fradulent as you are knowingly picking an inacurate value which benifits the bank the most and should be considered a crime.

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: