

# Courier\_Kata

May 6, 2019

## 0.0.1 Courier Kata (by Indranil Dutta):

As the problem is multi-layered. I kept on adding few bits of code as I went along and saved it in this notebook.

In this notebook, I prepare a dataset in the code itself for fast testing.

The dummy dataset has information from 4 senders in a list as you see below. Each sender dictionary has the attributes:

- Sender name
- Dimensions (in cm) of each parcel
- Corresponding weights (in kg) of each parcel
- Speedy Shipping Request condition

```
In [251]: Sender_list = [{'Sender': 'Will',
                          'Dimensions': [8,8,8,30,8,30,30,30,100,10,100,100],
                          'Weights': [4,2,11,3,5,10,40,60,2,1,10,10],
                          'Speedy': 'Y'},

                          {'Sender': 'Rohan',
                          'Dimensions': [49,49,49],
                          'Weights': [2,10,3],
                          'Speedy': 'N'},

                          {'Sender': 'Toby',
                          'Dimensions': [25,20,3],
                          'Weights': [7,4,3],
                          'Speedy': 'Y'},

                          {'Sender': 'Lea',
                          'Dimensions': [9,80,5],
                          'Weights': [53,20,3],
                          'Speedy': 'N'}]
```

Below I created a loop iterating through each sender's detail. There is also a nested loop that iterates through each parcel's weight and dimensions of the sender.

```
In [260]: import numpy as np
          for Sender in Sender_list:
```

```

# Finding the number of parcels
parcels = len(list(Sender['Dimensions']))

# Each counter will count the type of parcel
Small =0; Medium = 0; Mixed=0;

# Setting an array of zeros for Sender cost
Sender['Cost'] = np.zeros(parcels)
Sender['Discount'] = np.ones(parcels)
# Here I set an array of '1's. If discount applies, then it will be '0'

for p in list(range(0,parcels)):
    Mixed+=1

    # Checking for Small parcels: Part 1
    if Sender['Dimensions'][p]<10:
        Small += 1

        # Every 4th Small parcel is free order: Part 5a
        if Small%4 ==0:
            Sender['Discount'][p] =0
            Mixed-=1 # -1 to avoid counting already discounted parcels: Part 5d

        Sender['Cost'][p] =3

        # Checking for overweight of Small parcels: Part 3
        if Sender['Weights'][p]>1:
            Sender['Cost'][p] += (Sender['Weights'][p]-1)*2

    # Checking for Medium parcels: Part 1
    elif Sender['Dimensions'][p]<50:
        Medium += 1

        # Every 3rd Medium parcel is free order: Part 5b
        if Medium%3 ==0:
            Sender['Discount'][p] =0
            Mixed-=1

        Sender['Cost'][p] =8

        # Checking for overweight of Medium parcels: Part 3
        if Sender['Weights'][p]>3:
            Sender['Cost'][p] += (Sender['Weights'][p]-3)*2

    # Checking for Large parcels: Part 1
    elif Sender['Dimensions'][p]<100:
        Sender['Cost'][p] =15

```

```

        # Checking for overweight of Large parcels: Part 3
        if Sender['Weights'][p]>6:
            Sender['Cost'][p] += (Sender['Weights'][p]-6)*2

    # Checking for XL parcels
    else:
        Sender['Cost'][p] =25
        # Checking for overweight of XL parcels: Part 1
        if Sender['Weights'][p]>10:
            Sender['Cost'][p] += (Sender['Weights'][p]-10)*2

    # Checking for overweight of Heavy parcels: Part 4
    if Sender['Weights'][p]>50:
        Sender['Cost'][p] = 50 +(Sender['Weights'][p]-50)

    # Every 5th parcel has no cost: Part 5c
    if Mixed%5==0:
        Sender['Discount'][p] =0

Sender['Total'] = Sender['Cost']*Sender['Discount']
Sender['Savings'] = np.sum(Sender['Cost'])-np.sum(Sender['Total'])

```

I now print the total discounted cost and the savings for each sender. If the sender has chosen Speedy Shipping, the updated cost will be printed also.

```

In [270]: for Sender in Sender_list:
            print('For {0}, Total delivery cost: €{1}; Total savings: €{2}'
                  .format(Sender['Sender'],np.sum(Sender['Total']),np.sum(Sender['Savings'])))

            # Checking for Speedy Shipping request: Part 2
            if Sender['Speedy'] =='Y':
                print('As Speedy Shipping has been chosen, Delivery cost: €{0}'
                      .format(np.sum(Sender['Total']*2)))
            print('\n')

```

```

For Will, Total delivery cost: €163.0; Total savings: €140.0
As Speedy Shipping has been chosen, Delivery cost: €326.0

```

```

For Rohan, Total delivery cost: €30.0; Total savings: €8.0

```

```

For Toby, Total delivery cost: €33.0; Total savings: €0.0
As Speedy Shipping has been chosen, Delivery cost: €66.0

```

```

For Lea, Total delivery cost: €103.0; Total savings: €0.0

```

**Comments:** I started with a smaller dataset in sender and also parcel number for all my code result testing.

I finished Part 4 in two hours time but went abit beyond as Part 5 was the most interesting bit. I did not finish the whole problem but got solve upto Part 5d.

The further changes would be: - To set the code up such that it takes inputs from the user, as a date file or a single entry inputs. - To finish the solution. - To reduce number of lines in the code. I introduced the comments for my ease and also for the reader's understanding. - Coming up with better decision stree steps to make multiple decisions in one code line