BDA-9

November 25, 2018

1 Answers

After running our Utility function (Based on Profits of machine, calculated as a mean of profit/loss from the products of those machine) which is written in the next section, we summarize the utilities as below:

M1	M2	M3	M4	M5	M6
-30.6	68.3	14.7	74.3	21.6	6.85

And now sorting them in the order of **worst to best**: Where the Worst is M1 (Loosing Money here) and the best is M4

M1	M6	M3	M5	M2	M4
-30.6	6.85	14.7	21.6	68.3	74.3

From the above Table it is clear that M1 is loosing a lot of money on average, as there is a loss of -30.6 which signifies the Quality of products from this machine are not upto the mark.
On every other machine there is a profit, signifying overall there is a profit, especially on the M4 with a profit of 74.3, meaning the Quality of products from this machine are upto the mark, hence the high profit.

1.1 7th Machine

The expected Utility of the 7th machine is: **23.75**
Decision Analysis: Based on our Expected Utility the Company should go ahead and buy a new 7th machine as they may make a good profit on it even though not the best. One more way to look at it could be, buying this new machine may offset some of the loss(if not all) they are incurring from the first machine as all they care about is money

1.2 Hierarchical Model,Utility Function Code

```
In [1]: import numpy as np
        import pandas as pd
        import pystan
        import matplotlib.pyplot as plt

In [2]: factory_data = pd.read_csv('factory.txt',delimiter="\s+",header=None)

In [3]: factory_data = factory_data.values

In [4]: factory_model_hierachical ="""
        data {
          int<lower=0> N; // number of data points
          int<lower=0> K; // number of groups
          int<lower=1,upper=K> x[N]; // group indicator
          vector[N] y; //
        }
        parameters {
          real mu0;        // group means
          real<lower=0> sigma0; // common std
          vector[K] mu;
          real<lower=0> sigma;

        }
        model {
          mu ~ normal(mu0,sigma0);
          y ~ normal(mu[x], sigma);
        }
        generated quantities {
          real y_pred;
          real mu_pred;
          real y7_pred;
          vector[K] y_preds;
          for (i in 1:K){
              y_preds[i] = normal_rng(mu[i],sigma);
          }
          mu_pred = normal_rng(mu0,sigma0);
          y7_pred = normal_rng(mu_pred,sigma);
        }
        """
        hier_model = pystan.StanModel(model_code=factory_model_hierachical)

INFO:pystan:COMPILING THE C++ CODE FOR MODEL anon_model_4a08e4a25a8acd14e51bed036d90bca4 NOW.


In [5]: hierarchical_data ={
            'N':factory_data.size,
            'K':6,
```

```
            'x':list(range(1,7,1))*5,
            'y':factory_data.flatten()
        }

        fit = hier_model.sampling(data=hierarchical_data,iter=2000,chains=4)
        y7_pred = fit.extract()['y7_pred']
        mu_pred = fit.extract()['mu_pred']
```

```
In [6]: y_preds = fit.extract()['y_preds']
        y_preds.shape
```

```
Out[6]: (4000, 6)
```

1.3 Utility function

```
In [7]: Profits = np.zeros((y_preds.shape))
        for i, measurements in enumerate(y_preds):
            for j, utility_value in enumerate(measurements):
                if utility_value < 85:
                    #LOSS on the product
                    Profits[i,j] = -106
                else:
                    Profits[i,j] = 94
```

```
In [8]: print(np.mean(Profits,axis=0))
```

```
[-30.6   68.3   14.7   74.3   21.6    6.85]
```

```
In [9]: y7_pred = fit.extract()['y7_pred']
```

1.4 Utility of 7th machine

```
In [10]: y7_Profits = [(lambda x: 200-106 if x >= 85 else -106)(x) for x in y7_pred]
```

```
In [11]: print(np.mean(np.array(y7_Profits)))
```

```
23.75
```