outputRate

March 29, 2020

1 Determining the required output rate for datasets in Group B

Using data from the highest C-rate during cycling

1.0.1 Import necessary libraries

[1]:
```
import numpy as np
import pandas as pd
import copy
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# import codebase
import thermalModel_main as tmm
import thermalModel_groupC as tm_gc

import importlib
importlib.reload(tmm)
importlib.reload(tm_gc)
```

```
Using TensorFlow backend.
```

[1]:
```
<module 'thermalModel_groupC' from
'C:\\Users\\user\\Anaconda3\\lib\\thermalModel_groupC.py'>
```

[2]:
```
b3c_1 = tm_gc.load_preprocess_csv(filename = 'battery_1_3C.csv', to_plot =␣
 ↪False)

b3c_2 = tm_gc.load_preprocess_csv(filename = 'battery_1_3C_test2.csv', to_plot␣
 ↪= False)
```

```
Data loaded!
Data loaded!
```

[3]:

```
print("Shape: {}".format(b3c_1.shape))
print("Shape: {}".format(b3c_2.shape))
```

```
Shape: (3277, 16)
Shape: (3139, 16)
```

[4]:

```
b3c_1.describe()
```

[4]:

```
          Charging  Seconds count      Current      Voltage        AhCha  \
count  3277.000000    3277.000000  3277.000000  3277.000000  3277.000000
mean      0.777846    8195.000000    -0.069490     3.720561     4.684011
std       0.415758    4730.663713     4.867019     0.159376     2.231633
min       0.000000       5.000000    -8.430000     3.400625     0.000069
25%       1.000000    4100.000000    -0.420000     3.590750     3.271708
50%       1.000000    8195.000000    -0.320000     3.664750     3.828208
75%       1.000000   12290.000000    -0.130000     3.873000     7.165681
max       1.000000   16385.000000     8.640000     4.093250     7.658750

             AhDch       kWhCha       kWhDch         Tamb           T1  \
count  3277.000000  3277.000000  3277.000000  3277.000000  3277.000000
mean      4.000759    17.381164    14.782946    26.025081    40.535200
std       2.258859     8.276002     8.290770     0.348535     5.685099
min       0.000000     0.000237     0.000000    25.440000    27.870000
25%       2.296222    12.573116     8.861122    25.870000    36.060000
50%       3.586653    15.106861    13.069763    25.940000    41.630000
75%       5.992375    26.854964    23.262400    26.060000    45.440000
max       7.218403    30.681804    27.242252    27.620000    50.060000

                T2           T3           T4           T5           T6  \
count  3277.000000  3277.000000  3277.000000  3277.000000  3277.000000
mean     40.455041    40.513500    37.459628    40.723500    39.841358
std       6.076221     5.700552     4.513559     5.901069     5.501779
min      27.870000    27.870000    27.810000    27.870000    27.870000
25%      35.500000    36.060000    33.880000    36.130000    35.440000
50%      42.440000    41.750000    38.130000    42.380000    40.690000
75%      45.310000    45.380000    41.440000    45.560000    44.690000
max      49.500000    49.880000    45.130000    49.940000    49.310000

                T7
count  3277.000000
mean     38.661636
std       5.713062
min      27.190000
25%      33.750000
50%      40.810000
75%      43.130000
max      46.750000
```

[5]:

```
b3c_2.describe()
```

[5]:

```
          Charging  Seconds count      Current      Voltage       AhCha  \
count  3139.000000    3139.000000  3139.000000  3139.000000  3139.000000
mean      0.769035    7850.000000    -0.095155     3.723073     4.608977
std       0.421517    4531.477868     5.033231     0.156760     2.163969
min       0.000000       5.000000    -8.190000     3.375750     0.000097
25%       1.000000    3927.500000    -0.570000     3.594625     3.131549
50%       1.000000    7850.000000    -0.320000     3.657000     3.831542
75%       1.000000   11772.500000    -0.130000     3.878250     6.347521
max       1.000000   15695.000000     8.350000     4.105750     7.609236

             AhDch       kWhCha       kWhDch         Tamb           T1  \
count  3139.000000  3139.000000  3139.000000  3139.000000  3139.000000
mean      4.009164    17.097340    14.805198    25.590385    41.021803
std       2.229546     7.975852     8.121912     0.269293     5.686401
min       0.000000     0.000343     0.000000    24.940000    25.940000
25%       2.429069    12.563985     9.415377    25.370000    36.845000
50%       3.659375    14.792624    13.357634    25.620000    42.440000
75%       6.107764    25.184980    22.630029    25.810000    45.750000
max       7.361417    30.571709    27.136022    26.190000    49.250000

                T2           T3           T4           T5           T6  \
count  3139.000000  3139.000000  3139.000000  3139.000000  3139.000000
mean     40.688789    40.930503    37.865785    41.048684    40.440841
std       6.064306     5.689323     4.593925     5.892505     5.554298
min      25.190000    25.810000    26.000000    25.560000    26.000000
25%      36.130000    36.810000    34.380000    36.750000    36.250000
50%      43.000000    42.560000    38.750000    43.000000    41.690000
75%      45.500000    45.630000    41.630000    45.880000    45.000000
max      48.630000    48.880000    44.750000    49.000000    48.750000

                T7
count  3139.000000
mean     38.704737
std       5.640044
min      24.870000
25%      34.250000
50%      40.750000
75%      43.190000
max      46.250000
```

1.0.2 Check original rates

[6]:
```
df = copy.deepcopy(b3c_1)

df['temp_difference'] = df['T2'].diff(periods=1)
df['temp_difference'] = df['temp_difference'].abs()

row_indices=df[(df['temp_difference'] == 0.0)].index
df.drop(row_indices, inplace=True)
df.dropna(axis=0, inplace=True)
df.reset_index(drop=True, inplace=True)

df.describe()
print("Shape: {}".format(df.shape))
```

```
Shape: (2235, 17)
```

[7]:
```
df1 = copy.deepcopy(b3c_2)

df1['temp_difference'] = df1['T2'].diff(periods=1)
df1['temp_difference'] = df1['temp_difference'].abs()

row_indices=df1[(df1['temp_difference'] == 0.0)].index
df1.drop(row_indices, inplace=True)
df1.dropna(axis=0, inplace=True)
df1.reset_index(drop=True, inplace=True)

df1.describe()
print("Shape: {}".format(df1.shape))
```

```
Shape: (2147, 17)
```

[8]:
```
df.head(20)
```

[8]:
```
    Charging  Seconds count  Current   Voltage     AhCha  AhDch    kWhCha  \
0          1             40     7.29  3.473125  0.025125    0.0  0.087262
1          1             45     7.56  3.524444  0.035903    0.0  0.126537
2          1             50     7.53  3.562750  0.046611    0.0  0.166064
3          1             55     7.41  3.579500  0.057153    0.0  0.204578
4          1             60     7.32  3.643500  0.067625    0.0  0.246392
5          1             65     7.39  3.669500  0.078167    0.0  0.286833
6          1             70     7.27  3.698000  0.088542    0.0  0.327427
7          1             75     7.32  3.745750  0.099056    0.0  0.371037
8          1             80     7.19  3.791250  0.109389    0.0  0.414721
9          1             85     7.29  3.832750  0.119861    0.0  0.459398
10         1             90     7.29  3.862250  0.130333    0.0  0.503380
11         1             95     7.19  3.831000  0.140569    0.0  0.538522
12         1            100     7.37  3.846500  0.151083    0.0  0.581142
```

```
13         1            105     7.19  3.896750  0.161319    0.0  0.628622
14         1            110     7.34  3.854250  0.171722    0.0  0.661860
15         1            115     7.44  3.904250  0.182306    0.0  0.711766
16         1            120     7.46  3.939750  0.192917    0.0  0.760043
17         1            125     7.25  3.894500  0.203292    0.0  0.791719
18         1            130     7.41  3.893000  0.213833    0.0  0.832453
19         1            135     7.20  3.859250  0.224111    0.0  0.864901

    kWhDch   Tamb     T1     T2     T3     T4     T5     T6     T7  \
0      0.0  27.56  27.94  27.94  27.94  27.87  27.87  27.94  27.94
1      0.0  27.56  27.94  28.06  27.94  27.87  27.87  27.94  28.06
2      0.0  27.62  27.94  28.12  27.94  27.87  27.87  27.94  28.19
3      0.0  27.62  27.94  28.19  27.94  27.87  27.94  27.94  28.31
4      0.0  27.62  28.00  28.37  27.94  27.87  28.00  28.00  28.50
5      0.0  27.62  28.06  28.44  28.00  27.94  28.06  28.06  28.69
6      0.0  27.62  28.06  28.56  28.06  27.94  28.12  28.06  28.81
7      0.0  27.56  28.12  28.75  28.06  28.00  28.19  28.12  29.00
8      0.0  27.56  28.19  28.87  28.12  28.00  28.25  28.19  29.19
9      0.0  27.56  28.25  29.00  28.19  28.00  28.31  28.25  29.31
10     0.0  27.56  28.37  29.19  28.31  28.06  28.44  28.31  29.56
11     0.0  27.50  28.44  29.31  28.37  28.12  28.50  28.37  29.69
12     0.0  27.50  28.50  29.44  28.44  28.12  28.62  28.50  29.94
13     0.0  27.44  28.62  29.62  28.50  28.19  28.69  28.56  30.06
14     0.0  27.44  28.69  29.69  28.56  28.19  28.75  28.62  30.25
15     0.0  27.50  28.75  29.87  28.69  28.25  28.87  28.69  30.44
16     0.0  27.50  28.87  30.06  28.81  28.31  29.00  28.81  30.62
17     0.0  27.44  28.94  30.19  28.87  28.37  29.12  28.87  30.87
18     0.0  27.50  29.06  30.31  29.00  28.44  29.25  28.94  31.00
19     0.0  27.50  29.12  30.50  29.06  28.50  29.37  29.00  31.12

    temp_difference
0              0.07
1              0.12
2              0.06
3              0.07
4              0.18
5              0.07
6              0.12
7              0.19
8              0.12
9              0.13
10             0.19
11             0.12
12             0.13
13             0.18
14             0.07
15             0.18
```

```
16            0.19
17            0.13
18            0.12
19            0.19
```

[9]:
```
df1.head(20)
```

[9]:
```
    Charging  Seconds count  Current   Voltage     AhCha  AhDch    kWhCha  \
0          1             15    -0.22  3.591667  0.000125    0.0  0.000449
1          1             20     7.83  3.591875  0.011347    0.0  0.040758
2          1             35     7.83  3.663571  0.045042    0.0  0.165013
3          1             40     7.81  3.708438  0.056194    0.0  0.208394
4          1             45     7.54  3.697222  0.067042    0.0  0.247868
5          1             50     8.05  3.727500  0.078500    0.0  0.292609
6          1             55     7.80  3.760500  0.089681    0.0  0.337244
7          1             60     7.83  3.783500  0.100806    0.0  0.381398
8          1             65     7.69  3.787000  0.111792    0.0  0.423355
9          1             70     7.71  3.798000  0.122806    0.0  0.466416
10         1             75     8.12  3.816750  0.134028    0.0  0.511551
11         1             80     7.90  3.836750  0.145278    0.0  0.557395
12         1             85     7.61  3.867500  0.156222    0.0  0.604189
13         1             90     7.61  3.853750  0.167097    0.0  0.643951
14         1             95     7.68  3.899750  0.178014    0.0  0.694210
15         1            100     7.78  3.880750  0.189069    0.0  0.733731
16         1            105     7.88  3.881750  0.200292    0.0  0.777482
17         1            110     7.83  3.912500  0.211514    0.0  0.827548
18         1            115     7.93  3.920250  0.222806    0.0  0.873453
19         1            120     7.68  3.959750  0.233819    0.0  0.925867

    kWhDch   Tamb     T1     T2     T3     T4     T5     T6     T7  \
0      0.0  25.31  26.00  25.19  25.81  26.00  25.56  26.00  24.87
1      0.0  25.31  25.94  25.25  25.81  26.00  25.56  26.00  24.87
2      0.0  25.37  26.00  25.31  25.81  26.00  25.56  26.12  25.19
3      0.0  25.37  26.06  25.37  25.87  26.00  25.62  26.12  25.25
4      0.0  25.37  26.06  25.44  25.87  26.00  25.62  26.19  25.31
5      0.0  25.37  26.12  25.50  25.94  26.06  25.69  26.25  25.44
6      0.0  25.37  26.19  25.62  26.00  26.06  25.75  26.31  25.56
7      0.0  25.31  26.31  25.69  26.12  26.12  25.81  26.37  25.75
8      0.0  25.31  26.31  25.87  26.19  26.12  25.94  26.44  25.87
9      0.0  25.31  26.44  25.94  26.25  26.19  26.00  26.56  26.00
10     0.0  25.37  26.50  26.06  26.37  26.19  26.12  26.62  26.12
11     0.0  25.37  26.62  26.19  26.44  26.25  26.19  26.69  26.25
12     0.0  25.37  26.75  26.31  26.56  26.31  26.37  26.75  26.37
13     0.0  25.37  26.81  26.44  26.62  26.31  26.44  26.87  26.50
14     0.0  25.44  26.87  26.56  26.75  26.44  26.56  26.94  26.69
15     0.0  25.37  27.00  26.69  26.81  26.44  26.69  27.06  26.87
16     0.0  25.37  27.06  26.81  26.94  26.50  26.75  27.12  26.94
17     0.0  25.44  27.19  27.00  27.00  26.56  26.94  27.19  27.12
```

```
18     0.0  25.44  27.31  27.12  27.12  26.62  27.06  27.31  27.31
19     0.0  25.44  27.44  27.31  27.25  26.69  27.19  27.44  27.50

    temp_difference
0              0.06
1              0.06
2              0.06
3              0.06
4              0.07
5              0.06
6              0.12
7              0.07
8              0.18
9              0.07
10             0.12
11             0.13
12             0.12
13             0.13
14             0.12
15             0.13
16             0.12
17             0.19
18             0.12
19             0.19
```

[10]:

```
def cumsum_breach(x, target):
    total = 0
    for i, y in enumerate(x):
        total += y
        if total >= target:
            yield i
            total = 0

# list_for_cumsum = df['temp_difference'].values.tolist()
list_for_cumsum1 = df['temp_difference'].to_numpy(dtype=None, copy=True)
list_1 = list(np.around(list_for_cumsum1,2))
list_toKeep1 = list(cumsum_breach(list_1, 0.3)) # change this to change the␣
 ↪magnitude of cummulative change
list_2 = [x for x in range(0, len(df))]
list_toDrop1 = [x for x in list_2 if x not in list_toKeep1]
print("Number of elements to keep: {}".format(len(list_toKeep1)))
print("Number of elements to drop: {}".format(len(list_toDrop1)))
df_reduced = df.drop(list_toDrop1)
df_reduced = df_reduced.drop(columns = ['temp_difference'])
df_reduced.describe()
df_reduced.to_csv('groupC_reduced_dataset.csv')
```

```
# list_for_cumsum = df['temp_difference'].values.tolist()
list_for_cumsum3 = df1['temp_difference'].to_numpy(dtype=None, copy=True)
list_3 = list(np.around(list_for_cumsum3,2))
list_toKeep3 = list(cumsum_breach(list_3, 0.3)) # change this to change the␣
 ↪magnitude of cummulative change
list_4 = [x for x in range(0, len(df1))]
list_toDrop3 = [x for x in list_4 if x not in list_toKeep3]
print("Number of elements to keep: {}".format(len(list_toKeep3)))
print("Number of elements to drop: {}".format(len(list_toDrop3)))
df_reduced1 = df1.drop(list_toDrop3)
df_reduced1 = df_reduced1.drop(columns = ['temp_difference'])
df_reduced1.describe()
df_reduced1.to_csv('groupC1_reduced_dataset.csv')
```

```
Number of elements to keep: 598
Number of elements to drop: 1637
Number of elements to keep: 542
Number of elements to drop: 1605
```

[11]:
```
df_reduced.head(5)
```

[11]:
```
    Charging  Seconds count  Current  Voltage     AhCha  AhDch    kWhCha  \
3          1             55     7.41  3.57950  0.057153    0.0  0.204578
6          1             70     7.27  3.69800  0.088542    0.0  0.327427
8          1             80     7.19  3.79125  0.109389    0.0  0.414721
10         1             90     7.29  3.86225  0.130333    0.0  0.503380
13         1            105     7.19  3.89675  0.161319    0.0  0.628622

    kWhDch   Tamb     T1     T2     T3     T4     T5     T6     T7
3      0.0  27.62  27.94  28.19  27.94  27.87  27.94  27.94  28.31
6      0.0  27.62  28.06  28.56  28.06  27.94  28.12  28.06  28.81
8      0.0  27.56  28.19  28.87  28.12  28.00  28.25  28.19  29.19
10     0.0  27.56  28.37  29.19  28.31  28.06  28.44  28.31  29.56
13     0.0  27.44  28.62  29.62  28.50  28.19  28.69  28.56  30.06
```

[12]:
```
df_reduced1.head(5)
```

[12]:
```
    Charging  Seconds count  Current   Voltage     AhCha  AhDch    kWhCha  \
4          1             45     7.54  3.697222  0.067042    0.0  0.247868
8          1             65     7.69  3.787000  0.111792    0.0  0.423355
11         1             80     7.90  3.836750  0.145278    0.0  0.557395
14         1             95     7.68  3.899750  0.178014    0.0  0.694210
17         1            110     7.83  3.912500  0.211514    0.0  0.827548

    kWhDch   Tamb     T1     T2     T3     T4     T5     T6     T7
4      0.0  25.37  26.06  25.44  25.87  26.00  25.62  26.19  25.31
8      0.0  25.31  26.31  25.87  26.19  26.12  25.94  26.44  25.87
11     0.0  25.37  26.62  26.19  26.44  26.25  26.19  26.69  26.25
14     0.0  25.44  26.87  26.56  26.75  26.44  26.56  26.94  26.69
```

```
17     0.0  25.44  27.19  27.00  27.00  26.56  26.94  27.19  27.12
```

[13]:

```
print('Temperature changes with an cummulative magnitude of 0.3 degrees every:')
df_reduced['Seconds count'].diff(periods=1).describe()
```

```
Temperature changes with an cummulative magnitude of 0.3 degrees every:
```

[13]:

```
count    597.000000
mean      27.294807
std       16.854396
min        5.000000
25%       15.000000
50%       25.000000
75%       30.000000
max      120.000000
Name: Seconds count, dtype: float64
```

[14]:

```
print('Temperature changes with an cummulative magnitude of 0.3 degrees every:')
df_reduced1['Seconds count'].diff(periods=1).abs().describe()
```

```
Temperature changes with an cummulative magnitude of 0.3 degrees every:
```

[14]:

```
count    541.000000
mean      28.807763
std       15.767759
min       10.000000
25%       20.000000
50%       25.000000
75%       35.000000
max      125.000000
Name: Seconds count, dtype: float64
```