

main_nb

January 12, 2021

```
[2]: import pandas as pd
import matplotlib.pyplot as plt
import time

from helpers.MetHelper import recognize_met_value, \
    ↳did_user_executed_moderate_activity, get_amount_of_executed_moderate_activity
from helpers.PandasHelper import read_csv_activpal20
from factories.ActivPalUserFactory import ActivPalUserFactory
from helpers.ActivityHelper import recognize_activity
from helpers.PlotHelper import plot_day, activites_distribution_plot

start_time = time.time()
```

1 Phase 1: loading in data

```
[3]: user_code = 'BMR002'
user_factory = ActivPalUserFactory()

activPal_user = user_factory.create_from_respondent_code(user_code)
ap_df = read_csv_activpal20(user_code)
```

2 Phase 2 Recognizing activities

```
[4]: dataset = pd.DataFrame(index=pd.to_datetime([]))
unique_days = list(set(ap_df.index.to_period('D').strftime('%Y-%m-%d')))

for day in unique_days:
    day_dataset = ap_df[day].copy()

    dataset = pd.concat([dataset, recognize_activity(day_dataset, \
    ↳activPal_user)])

    break
```

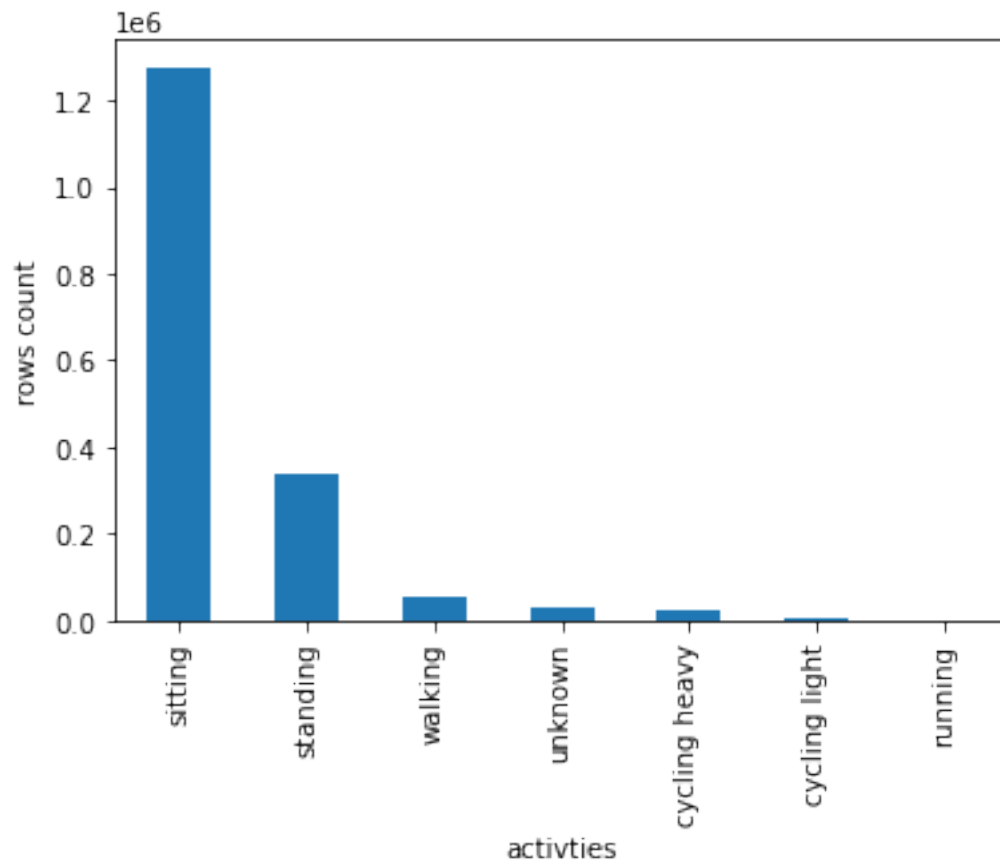
```
/opt/jupyterhub/anaconda/lib/python3.6/site-packages/dask/dataframe/utils.py:15:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
```

```
import pandas.util.testing as tm
```

```
[01:30:32] WARNING: ../src/objective/regression_obj.cu:174: reg:linear is now
deprecated in favor of reg:squarederror.
```

```
[01:30:32] WARNING: ../src/objective/regression_obj.cu:174: reg:linear is now
deprecated in favor of reg:squarederror.
```

```
[5]: activites_distribution_plot(dataset)
```



3 phase 3 Recognizing MET-value

```
[6]: dataset = recognize_met_value(dataset, activPal_user)
```

```
[01:32:27] WARNING: ../src/objective/regression_obj.cu:174: reg:linear is now
deprecated in favor of reg:squarederror.
```

```
[01:32:27] WARNING: ../src/objective/regression_obj.cu:174: reg:linear is now
```

deprecated in favor of `reg:squarederror`.

PHASE 3.1 PREPARING DATA

<code>sum_mag_acc</code>	<code>float64</code>
<code>mean_speed</code>	<code>float64</code>
<code>activity</code>	<code>object</code>
<code>gender</code>	<code>int64</code>
<code>estimated_level</code>	<code>int64</code>
<code>is_sporter</code>	<code>int64</code>
<code>length_cm</code>	<code>float64</code>
<code>weight_kg</code>	<code>float64</code>
<code>bmi</code>	<code>float64</code>
<code>age_category</code>	<code>int64</code>
<code>meets_balance_guidelines</code>	<code>int64</code>
<code>meets_activity_guidelines</code>	<code>int64</code>
<code>dtype:</code>	<code>object</code>

PHASE 3.2 MODEL PREDICTING

PHASE 3.3 DONE

4 Phase 4: calculating if userr has been active

```
[7]: has_user_been_active = did_user_executed_moderate_activity(dataset)
```

```
[8]: "User has been active" if has_user_been_active else "User did not move enough_
    ↪for the week"
```

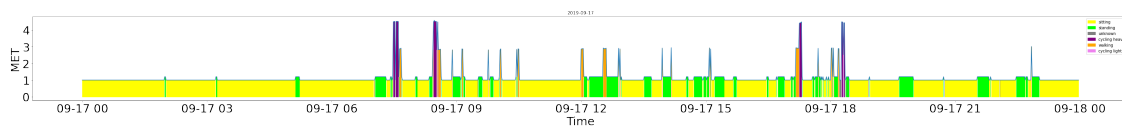
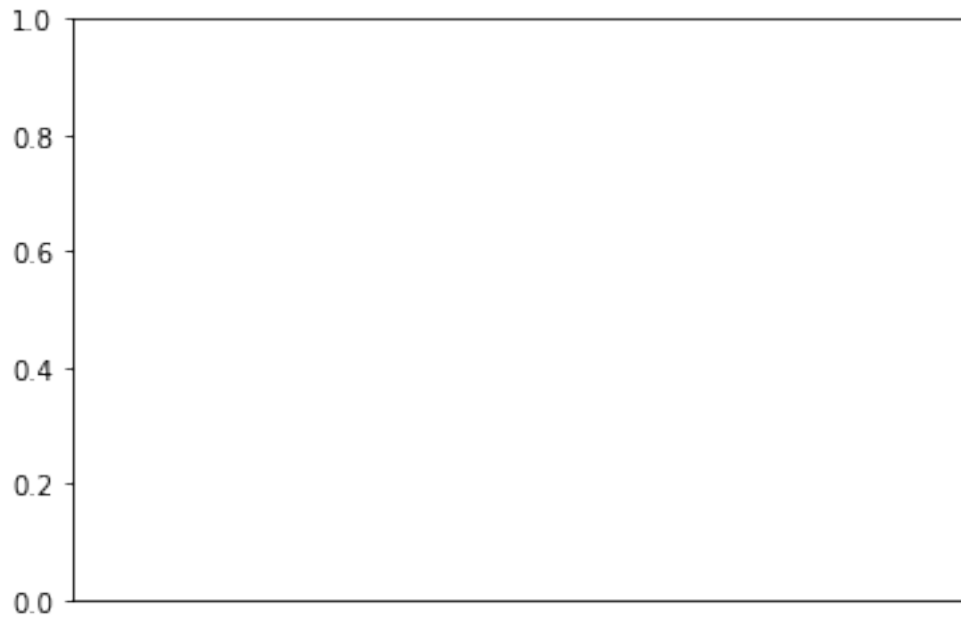
```
[8]: 'User did not move enough for the week'
```

```
[9]: str(get_amount_of_executed_moderate_activity(dataset)) + "minutes"
```

```
[9]: '22minutes'
```

5 Phase 5: plot daily activities

```
[13]: for day in unique_days:
        if day in dataset.index:
            day_dataset = dataset[day]
            plot_day(day_dataset, day)
```

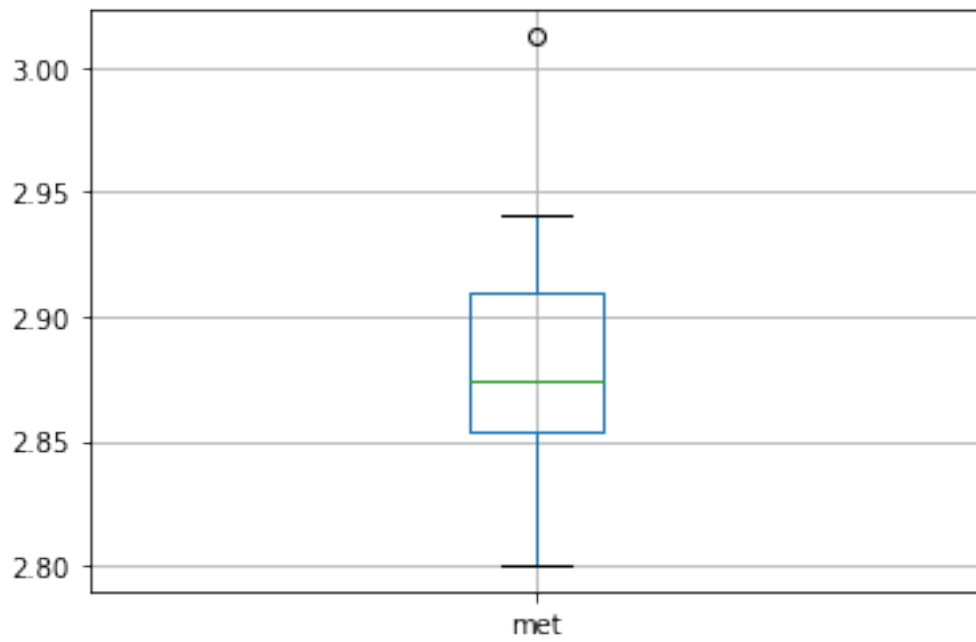


```
[11]: print("--- %s seconds ---" % (time.time() - start_time))
```

```
--- 193.5533390045166 seconds ---
```

```
[12]: dataset[dataset['activity']=='walking'].boxplot('met')
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0abc4a6cc0>
```



[]:

[]: