Meeting 8

10/28/2021

Chisom Sopuruchi

Deliverables

- Edit Epoch Detection Function
- Heart rate Function

Methodology and Learnings

How I did it:
Implement the pseudocode

Results

```python
# import libraries
import pandas as pd
import math
# import class
from sklearn.tree import DecisionTreeRegressor

def decisionTree_epochDetection(num_bins,Xy_array):
    # exception handling
    # max_leaf_nodes must either be None or larger than 1
    # therefore num_bins must be at least 2
    if(num_bins < 2):
        print("num_bins must be greater than one")
        print("Changing value of num_bins to minimum possible value")
        num_bins = 2
    # fitting the regression tree X as features/predictor and y as label/target
    clf = DecisionTreeRegressor(max_leaf_nodes = num_bins).fit(Xy_array[:,0].reshape(-1, 1), Xy_array[:,1])

    # variables creation
    num_nodes = clf.tree_.node_count
    left_child = clf.tree_.children_left
    right_child = clf.tree_.children_right
    threshold = clf.tree_.threshold
    # list to store the bin edges
    bin_edges = [0,146884]

    # loop through all the nodes
    for i in range(num_nodes):
        # If the left and right child of a node is not the same(-1) we have an internal node
        # which we will append to bin_node list
        if left_child[i]!=right_child[i]:
            bin_edges.append(math.ceil(threshold[i]))
    # sort the nodes in increasing order
    bin_edges.sort()
    # create dictionary to store epoch bin edges
    epoch_dict = {}
    # put in each dictionary index 2 consecutive bin edges
    for i in range(num_bins):
        epoch_dict[str(i+1)] = [bin_edges[i], bin_edges[i+1]]
    return epoch_dict
```

Results

```
#read eeg from 'log.csv'
biometric_dataframe = pd.read_csv('log.csv')
biometric_var = biometric_dataframe.Temp
time_index = biometric_dataframe.index

# convert to numpy 2D array containing time index (column 0) and biometric vairable
(column 1)
Xy_array = np.column_stack((time_index, biometric_var))
num_bins = 10
epoch_dict = decisionTree_epochDetection(num_bins,Xy_array)
print(epoch_dict)
# HR_per_epoch(epoch_dict, biometric_dataframe)
```

```
uche@Boo-VirtualBox:~/Documents/SeniorDesign/functions$ python3 epoch_function.py
{'1': [0, 53029], '2': [53029, 53528], '3': [53528, 121033], '4': [121033, 130001], '5': [130001, 132990], '6': [132990, 133986], '7': [133986, 134484]
, '8': [134484, 138719], '9': [138719, 145195], '10': [145195, 146884]}
```

Results

```
def HR_per_epoch(epoch_dict, eeg_data):
    #import library
    import pandas as pd

    # define output dictionary
    epochbm_dict={}

    for key, value in epoch_dict.items():
        list_HR = [eeg_data.HR[i] for i in range(value[0], value[1])]
        # populate the output dictionary
        epochbm_dict[key]=[epoch_dict[key],
pd.DataFrame(data=pd.Series(list_HR).rolling(window=100).mean(),
index=range(value[0], value[1]), columns=['HR'])]

    return epochbm_dict
```

```
uche@Boo-VirtualBox:~/Documents/SeniorDesign/functions$ python3 epoch_function.py
{'1': [[0, 53029],          HR
0       NaN
1       NaN
2       NaN
3       NaN
4       NaN
...     ...
53024  54.0
53025  54.0
53026  54.0
53027  54.0
53028  54.0

[53029 rows x 1 columns]], '2': [[53029, 53528],        HR
53029 NaN
53030 NaN
53031 NaN
53032 NaN
53033 NaN
...    ..
53523 NaN
53524 NaN
53525 NaN
53526 NaN
53527 NaN

[499 rows x 1 columns]], '3': [[53528, 121033],          HR
53528   59.0
53529   59.0
53530   59.0
```