In [5]:
```
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
from tpot import TPOTClassifier
%matplotlib inline
import matplotlib.pyplot as plt
from scipy import signal
import pickle


import sklearn.metrics
from sklearn.model_selection import cross_val_score
from sklearn import svm
import numpy as np
import pandas as pd
from sklearn.metrics import precision_recall_fscore_support

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', -1)
```

```
/Users/aryamihirs/Library/Python/3.9/lib/python/site-packages/tpot/builtins/__
init__.py:36: UserWarning: Warning: optional dependency `torch` is not availab
le. - skipping import of NN models.
  warnings.warn("Warning: optional dependency `torch` is not available. - skip
ping import of NN models.")
/var/folders/js/9ql558xx43752p8dhdcbmdhm0000gn/T/ipykernel_2289/3351167182.py:
21: FutureWarning: Passing a negative integer is deprecated in version 1.0 and
will not be supported in future version. Instead, use None to not limit the co
lumn width.
  pd.set_option('display.max_colwidth', -1)
```

In [6]:
```
dataframe_hrv = pd.read_csv("dataset/dataframe_hrv.csv")
dataframe_hrv = dataframe_hrv.reset_index(drop=True)
```

In [7]:
```
display(dataframe_hrv.head(5))
```

	ECG	EMG	HR	RESP	Seconds	footGSR	handGSR	interval in seconds	m
0	-0.001974	-0.004737	77.815789	10.801842	12.529684	2.417132	10.889447	0.614632	
1	0.002935	-0.004457	101.978261	10.750609	30.503500	2.417109	11.251065	0.647826	
2	0.006745	-0.003426	104.957447	10.557234	52.523021	2.226872	11.379638	0.646383	
3	-0.004043	-0.002532	87.702128	10.640128	74.402170	2.173021	11.470830	0.645000	
4	0.012745	-0.004426	88.829787	10.699319	96.219617	2.017106	11.135255	0.645000	

In [8]:
```
display(dataframe_hrv.describe())
print(dataframe_hrv.columns)
```

	ECG	EMG	HR	RESP	Seconds	footGSR	h
count	4129.000000	4023.000000	4129.000000	4129.000000	4129.000000	4129.000000	4056
mean	0.170927	0.604475	83.136251	29.846928	2278.789710	6.487689	11
std	0.137222	0.807892	17.633319	14.024560	1313.496347	4.307487	6
min	-0.699585	-0.697800	11.800000	-12.606244	12.529684	0.971111	-28
25%	0.084500	0.108651	72.861111	28.565064	1114.941256	2.800841	6
50%	0.144171	0.302947	79.926829	32.916500	2295.794860	5.770098	9
75%	0.207457	0.822690	89.400000	38.627444	3425.276711	8.837477	1(
max	0.681714	9.491700	372.000000	52.089590	5005.373902	22.582463	3(

```
Index(['ECG', 'EMG', 'HR', 'RESP', 'Seconds', 'footGSR', 'handGSR',
       'interval in seconds', 'marker', 'newtime', 'stress', 'time', 'NNRR',
       'AVNN', 'SDNN', 'RMSSD', 'pNN50', 'TP', 'ULF', 'VLF', 'LF', 'HF',
       'LF_HF'],
      dtype='object')
```

In [9]:
```
def fix_stress_labels(df='',label_column='stress'):
    df['stress'] = np.where(df['stress']>=0.5, 1, 0)
    display(df["stress"].unique())
    return df
dataframe_hrv = fix_stress_labels(df=dataframe_hrv)
```

```
array([0, 1])
```

In [10]:
```
def missing_values(df):
    df = df.reset_index()
    df = df.replace([np.inf, -np.inf], np.nan)
    df[~np.isfinite(df)] = np.nan
#     df.plot( y=["HR"])
    df['HR'].fillna((df['HR'].mean()), inplace=True)
    df['HR'] = signal.medfilt(df['HR'],13)
#     df.plot( y=["HR"])

    df=df.fillna(df.mean(),inplace=True)
    return df

dataframe_hrv = missing_values(dataframe_hrv)
```

In [11]:
```
selected_x_columns = ['HR','interval in seconds','AVNN', 'RMSSD', 'pNN50', 'TP'

X = dataframe_hrv[selected_x_columns]
y = dataframe_hrv['stress']

display(X.columns)
display(X.describe())
display(X.shape)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [11], line 3
      1 selected_x_columns = ['HR','interval in seconds','AVNN', 'RMSSD', 'pNN
50', 'TP', 'ULF', 'VLF', 'LF', 'HF','LF_HF']
----> 3 X = dataframe_hrv[selected_x_columns]
      4 y = dataframe_hrv['stress']
      6 display(X.columns)

TypeError: 'NoneType' object is not subscriptable
```

In [14]:
```
def do_tpot(generations=5, population_size=10,X='',y=''):

    X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                        train_size=0.80, test_s

    tpot = TPOTClassifier(generations=generations, population_size=population_s
    tpot.fit(X_train, y_train)
    print(tpot.score(X_test, y_test))
    tpot.export('tpot_pipeline.py')
    return tpot

tpot_classifer = do_tpot(generations=10, population_size=20,X=X,y=y)
```

```
Optimization Progress:  17%|██         | 38/220 [00:17<01:14,  2.46pipeline/s]
Generation 1 - Current best internal CV score: 0.7359913799971757
Optimization Progress:  26%|███        | 57/220 [00:26<01:18,  2.09pipeline/s]
Generation 2 - Current best internal CV score: 0.76929638292857
Optimization Progress:  34%|████       | 74/220 [00:43<01:56,  1.25pipeline/s]
Generation 3 - Current best internal CV score: 0.76929638292857
Optimization Progress:  41%|████       | 91/220 [00:59<01:43,  1.25pipeline/s]
Generation 4 - Current best internal CV score: 0.76929638292857
Optimization Progress:  50%|█████      | 110/220 [01:39<04:11,  2.29s/pipeline]
Generation 5 - Current best internal CV score: 0.7774748988273074
Optimization Progress:  58%|██████     | 127/220 [02:01<01:59,  1.29s/pipeline]
Generation 6 - Current best internal CV score: 0.7838352359049431
Optimization Progress:  67%|███████    | 147/220 [02:19<00:55,  1.32pipeline/s]
Generation 7 - Current best internal CV score: 0.7838352359049431
Optimization Progress:  75%|████████   | 166/220 [02:53<00:56,  1.05s/pipeline]
Generation 8 - Current best internal CV score: 0.7838352359049431
Optimization Progress:  85%|█████████  | 186/220 [03:07<00:24,  1.41pipeline/s]
Generation 9 - Current best internal CV score: 0.7914052141106741

Generation 10 - Current best internal CV score: 0.7920090778568057

Best pipeline: KNeighborsClassifier(input_matrix, KNeighborsClassifier__n_neig
hbors=4, KNeighborsClassifier__p=DEFAULT, KNeighborsClassifier__weights=distan
ce)
0.805084745763
```

In [15]:
```
def plotFitBitReading(dfnewHRV='', predictor = "none",selected_x_columns=''):
    dfnewHRV = missing_values(dfnewHRV)
    dfnewPol = dfnewHRV[selected_x_columns].fillna(0)
```

```
    print(dfnewPol.columns)
    print(dfnewPol.shape)
    pred = predictor.predict_proba(dfnewPol)

    dfpred = pd.DataFrame(pred)

    dfpred.columns = [["FALSE","TRUE"]]
    dfpred['stress'] = np.where(dfpred["TRUE"] > 0.5, 1, np.nan)


    dfnewHRV["stress"] = dfpred["stress"]
    dfnewHRV.loc[dfnewHRV["steps"] > 0, 'stress'] = np.nan
    #mark is to mark the RR peaks as stress
    dfnewHRV.loc[dfnewHRV["stress"] == 1, 'stress'] = dfnewHRV['interval in sec
    dfnewHRV.loc[dfnewHRV["steps"] > 0, 'moving'] = dfnewHRV['interval in secor
    dfnewHRV["minutes"] = (dfnewHRV['newtime']/60)/1000

    from itertools import cycle, islice
    my_colors = list(islice(cycle(['b', 'r', 'y', 'k']), None, len(dfnewHRV)))
    plot = dfnewHRV.plot(x="minutes", y=['interval in seconds',"stress", "movir

    fig = plot.get_figure()
```

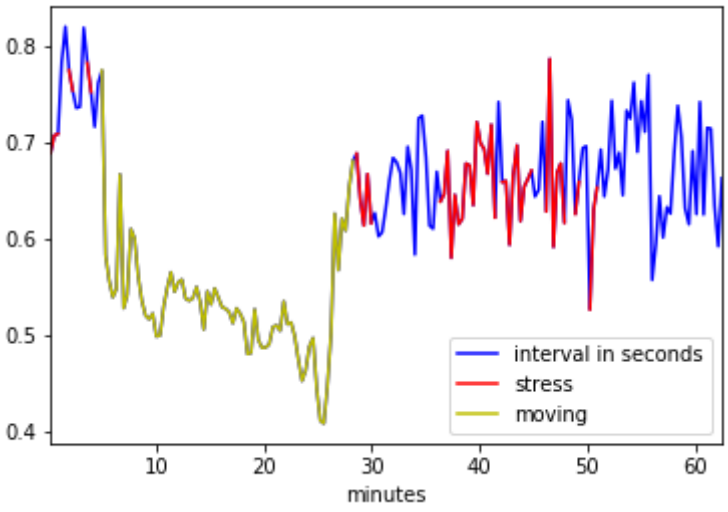
In [16]:
```
import glob

for filename in glob.iglob('dataset/**/*.csv', recursive=True):
    if 'dfnew' in filename:
        print(filename)
```

```
dataset/Vikings/Female_33_years_old/dfnewHRV_Female_33_years_old_vikings.csv
dataset/Vikings/male_22years/dfnewHRVVIKINGSFULLSERIES.csv
dataset/Vikings/Male_23_years_old/dfnewHRVMale_23_years_old.csv
dataset/Vikings/Male_21_years/dfnewHRV.csv
dataset/Vikings/Female_24_years_old/dfnewHRV.csv
dataset/American Horror Story/male_22years_american_horror_story/dfnewHRVmale_
22yearsAmericanHorrorStory.csv
dataset/American Horror Story/Male_25_years_american_horror_story/dfnewHRV_Mal
e_25_years_american_horror_story.csv
dataset/American Horror Story/female_24_years_american_horror_story/dfnewHRV.c
sv
```

In [17]:
```
input_df = pd.read_csv('dataset/Vikings/Female_24_years_old/dfnewHRV.csv')
plotFitBitReading(input_df,tpot_classifer,selected_x_columns)
```

```
Index(['HR', 'interval in seconds', 'AVNN', 'RMSSD', 'pNN50', 'TP', 'ULF',
       'VLF', 'LF', 'HF', 'LF_HF'],
      dtype='object')
(185, 11)
```


0.8
0.7
0.6
0.5
0.4
10
20
30
40
50
60
minutes
interval in seconds
stress
moving