

Add All Imports

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
```

Declare Training and Testing Data

```
In [2]: train = pd.read_csv("data/train.csv.zip")
test = pd.read_csv("data/test.csv.zip")
```

```
In [3]: test.head()
```

```
Out[3]:
```

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	HR	pNN25	f
0	721.901897	727.267280	74.722315	12.361264	12.361069	6.044877	84.121868	4.933333	0.0
1	843.538633	844.407930	58.499429	19.298880	19.298795	3.031234	71.478642	21.000000	0.2
2	958.523868	966.671125	132.849110	21.342715	21.342653	6.224565	63.874293	24.133333	1.8
3	824.838669	842.485905	117.822094	11.771814	11.771248	10.008830	74.330531	4.733333	0.9
4	756.707933	747.941620	143.968457	13.357748	13.356388	10.777899	82.092049	5.933333	0.6

5 rows × 36 columns

```
In [4]: train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 369289 entries, 0 to 369288
Data columns (total 36 columns):
#   Column                Non-Null Count  Dtype
---  -
0   MEAN_RR                369289 non-null float64
1   MEDIAN_RR              369289 non-null float64
2   SDRR                   369289 non-null float64
3   RMSSD                  369289 non-null float64
4   SDSD                   369289 non-null float64
5   SDRR_RMSSD             369289 non-null float64
6   HR                     369289 non-null float64
7   pNN25                  369289 non-null float64
8   pNN50                  369289 non-null float64
9   SD1                    369289 non-null float64
10  SD2                     369289 non-null float64
11  KURT                    369289 non-null float64
12  SKEW                    369289 non-null float64
13  MEAN_REL_RR            369289 non-null float64
14  MEDIAN_REL_RR          369289 non-null float64
15  SDRR_REL_RR            369289 non-null float64
16  RMSSD_REL_RR           369289 non-null float64
17  SDSD_REL_RR            369289 non-null float64
18  SDRR_RMSSD_REL_RR      369289 non-null float64
19  KURT_REL_RR            369289 non-null float64
20  SKEW_REL_RR            369289 non-null float64
21  VLF                     369289 non-null float64
22  VLF_PCT                369289 non-null float64
23  LF                      369289 non-null float64
24  LF_PCT                 369289 non-null float64
25  LF_NU                   369289 non-null float64
26  HF                      369289 non-null float64
27  HF_PCT                 369289 non-null float64
28  HF_NU                   369289 non-null float64
29  TP                      369289 non-null float64
30  LF_HF                   369289 non-null float64
31  HF_LF                   369289 non-null float64
32  sampen                  369289 non-null float64
33  higuci                  369289 non-null float64
34  datasetId               369289 non-null int64
35  condition                369289 non-null object
dtypes: float64(34), int64(1), object(1)
memory usage: 101.4+ MB
```

```
In [5]: test.columns
```

```
Out[5]: Index(['MEAN_RR', 'MEDIAN_RR', 'SDRR', 'RMSSD', 'SDSD', 'SDRR_RMSSD', 'HR',
              'pNN25', 'pNN50', 'SD1', 'SD2', 'KURT', 'SKEW', 'MEAN_REL_RR',
              'MEDIAN_REL_RR', 'SDRR_REL_RR', 'RMSSD_REL_RR', 'SDSD_REL_RR',
              'SDRR_RMSSD_REL_RR', 'KURT_REL_RR', 'SKEW_REL_RR', 'VLF', 'VLF_PCT',
              'LF', 'LF_PCT', 'LF_NU', 'HF', 'HF_PCT', 'HF_NU', 'TP', 'LF_HF',
              'HF_LF', 'sampen', 'higuci', 'datasetId', 'condition'],
              dtype='object')
```

Preprocess

```
In [6]: print(train.isnull().sum().sum())
        print(test.isnull().sum().sum())
```

0
0

```
In [7]: train['condition'][train['condition']=='interruption'] = "time pressure"
        test['condition'][test['condition']=='interruption'] = "time pressure"
```

C:\Users\Administrator\AppData\Local\Temp\ipykernel_9784\3502460124.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
train['condition'][train['condition']=='interruption'] = "time pressure"
```

C:\Users\Administrator\AppData\Local\Temp\ipykernel_9784\3502460124.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
test['condition'][test['condition']=='interruption'] = "time pressure"
```

Transform Y-value to binary (0 and 1) value

```
In [8]: le = preprocessing.LabelEncoder()
        le.fit(train['condition'])
        train['condition'] = le.transform(train['condition'])
        test['condition'] = le.transform(test['condition'])
        train['condition'].unique()
```

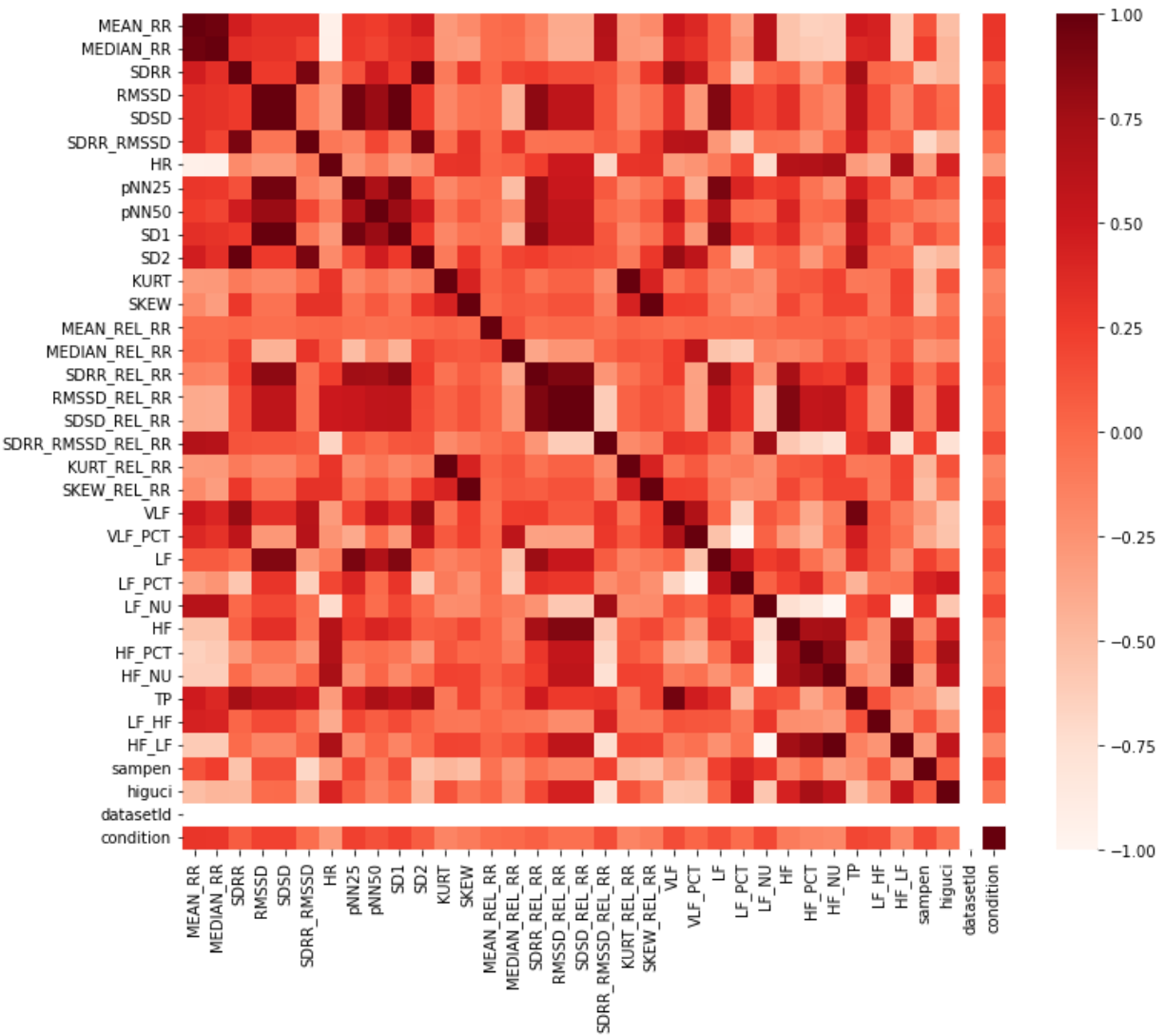
Out[8]: array([0, 1])

Stressed = 1 and Not Stressed = 0

Feature Extraction

Finding Correlations from heatmap

```
In [9]: plt.figure(figsize=(12,10))
        corr = train.corr()
        sns.heatmap(corr, annot=False, cmap=plt.cm.Reds)
        plt.show()
        corr
```



Out[9]:

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	H
MEAN_RR	1.000000	0.960949	0.462882	0.333046	0.332950	0.332924	-0.944552
MEDIAN_RR	0.960949	1.000000	0.333753	0.309061	0.309021	0.203815	-0.929640
SDRR	0.462882	0.333753	1.000000	0.262933	0.262610	0.914952	-0.202327
RMSSD	0.333046	0.309061	0.262933	1.000000	1.000000	-0.067463	-0.284871
SDSD	0.332950	0.309021	0.262610	1.000000	1.000000	-0.067835	-0.284863
SDRR_RMSSD	0.332924	0.203815	0.914952	-0.067463	-0.067835	1.000000	-0.084332
HR	-0.944552	-0.929640	-0.202327	-0.284871	-0.284863	-0.084332	1.000000
pNN25	0.286793	0.270630	0.136393	0.951750	0.951771	-0.145871	-0.262900
pNN50	0.245215	0.196992	0.473307	0.794846	0.794774	0.187629	-0.114890
SD1	0.332950	0.309021	0.262610	1.000000	1.000000	-0.067835	-0.284863
SD2	0.462577	0.333450	0.999997	0.260933	0.260609	0.915475	-0.202000
KURT	-0.292933	-0.284508	-0.096440	-0.178586	-0.178621	-0.029829	0.303670
SKEW	-0.197770	-0.317130	0.276610	-0.053997	-0.054125	0.313448	0.306400
MEAN_REL_RR	-0.016858	-0.018019	0.000217	-0.024209	-0.024228	0.014103	0.017770
MEDIAN_REL_RR	0.009506	-0.006349	0.196399	-0.437706	-0.437774	0.300669	0.054640
SDRR_REL_RR	-0.142334	-0.157426	0.236845	0.836688	0.836678	-0.048227	0.234550
RMSSD_REL_RR	-0.396832	-0.401170	0.157551	0.583895	0.583878	-0.041412	0.499860
SDSD_REL_RR	-0.396833	-0.401170	0.157551	0.583894	0.583877	-0.041412	0.499860
SDRR_RMSSD_REL_RR	0.654578	0.635819	0.119951	0.102232	0.102240	0.078020	-0.675550
KURT_REL_RR	-0.292933	-0.284508	-0.096440	-0.178586	-0.178621	-0.029829	0.303670
SKEW_REL_RR	-0.197770	-0.317130	0.276610	-0.053997	-0.054125	0.313448	0.306400
VLF	0.499427	0.399023	0.798173	0.341457	0.341293	0.617610	-0.304560
VLF_PCT	0.388775	0.313607	0.579652	-0.278834	-0.278996	0.637485	-0.254540
LF	0.079001	0.084147	-0.018551	0.886955	0.887012	-0.276739	-0.097220
LF_PCT	-0.335932	-0.262426	-0.575309	0.299521	0.299684	-0.637903	0.194130
LF_NU	0.631864	0.627497	0.003269	0.185242	0.185263	-0.041196	-0.724260
HF	-0.550557	-0.550244	0.049095	0.332323	0.332324	-0.062388	0.647420
HF_PCT	-0.643472	-0.596973	-0.281334	-0.072118	-0.072056	-0.262097	0.656420
HF_NU	-0.631864	-0.627497	-0.003269	-0.185242	-0.185263	0.041196	0.724260
TP	0.482112	0.388862	0.749198	0.595021	0.594883	0.499011	-0.301290
LF_HF	0.435019	0.421338	0.016527	0.170218	0.170234	-0.040053	-0.404740
HF_LF	-0.606072	-0.602190	-0.000024	-0.163887	-0.163904	0.036832	0.701860
sampen	0.122610	0.235890	-0.549505	0.134697	0.134970	-0.680969	-0.307580

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	H
higuci	-0.516444	-0.461297	-0.473624	-0.007870	-0.007754	-0.455149	0.42095
datasetId	NaN	NaN	NaN	NaN	NaN	NaN	NaN
condition	0.294276	0.285178	0.076084	0.211658	0.211673	-0.020103	-0.29529

36 rows × 36 columns

Return all features that are above threshold of 0.1

```
In [10]: def feature_selection(correlation, threshold):
         selected_features = []
         for i in range(corr.shape[0]):
             if corr.iloc[i,35] > threshold:
                 selected_features.append(train.iloc[:,i])
         return pd.DataFrame(selected_features).T
```

```
In [11]: print(feature_selection(corr,0.1).columns)

Index(['MEAN_RR', 'MEDIAN_RR', 'RMSSD', 'SDSD', 'pNN25', 'pNN50', 'SD1',
       'SDRR_RMSSD_REL_RR', 'VLF', 'LF', 'LF_NU', 'TP', 'LF_HF', 'sampen',
       'condition'],
      dtype='object')
```

Feature Selection

```
In [12]: reduced_train = train[["MEAN_RR", "pNN50", "RMSSD", "HR", "condition"]]
```

```
In [13]: reduced_train.head()
```

```
Out[13]:
```

	MEAN_RR	pNN50	RMSSD	HR	condition
0	885.157845	0.533333	15.554505	69.499952	0
1	939.425371	0.000000	12.964439	64.363150	1
2	898.186047	0.200000	16.305279	67.450066	1
3	881.757865	0.133333	15.720468	68.809562	0
4	809.625331	0.200000	19.213819	74.565728	0

Model Training (Random Forest)

```
In [14]: X_train = reduced_train.iloc[:, :-1]
         y_train = reduced_train.iloc[:, -1]
```

```
In [15]: X_test = test[X_train.columns]
         y_test = test['condition']
```

Defining the RandomForest Classifier

```
In [16]: model = RandomForestClassifier()
         model.fit(X_train,y_train)
```

```
Out[16]: RandomForestClassifier()
```

```
In [17]: y_pred = model.predict(X_test)
```

```
In [18]: y_pred
```

```
Out[18]: array([0, 1, 0, ..., 0, 0, 1])
```

```
In [19]: def stressed(i):
         if model.predict([X_test.iloc[i]]) == [0]:
             return "Not Stressed"
         else:
             return "Stressed"
         print(stressed(46))
```

```
Not Stressed
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

Accuracy Results

```
In [20]: accuracy_score(y_test,y_pred)
```

```
Out[20]: 0.9995125874296298
```

```
In [21]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	22158
1	1.00	1.00	1.00	18875
accuracy			1.00	41033
macro avg	1.00	1.00	1.00	41033
weighted avg	1.00	1.00	1.00	41033

```
In [ ]:
```