

Preprocessing

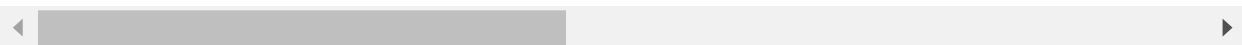
```
In [1]: import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import pandas as pd
```

```
In [11]: # Create dataframe for training dataset and print five first rows as preview
df = pd.read_csv('new_dataset.csv')
df.head()
```

Out[11]:

	pp	Condition	HR	RMSSD	SCL	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnam
0	PP1	0	61.0	0.061420	80.239727	0.944941	0.968862	0.023946	0.001300	0.016
1	PP1	0	61.0	0.061420	77.365127	0.930303	0.884570	0.076952	0.001144	0.017
2	PP1	0	64.0	0.049663	77.359559	0.933104	0.931965	0.031468	0.000371	0.023
3	PP1	0	60.0	0.052487	76.728772	0.904466	0.806947	0.105516	0.006459	0.009
4	PP1	0	61.0	0.051189	76.512877	0.929025	0.951412	0.028358	0.001095	0.018

5 rows × 27 columns



```
In [12]: #columns containg any null value  
df.isnull().any()
```

```
Out[12]: pp                False  
Condition                False  
HR                       True  
RMSSD                    True  
SCL                      True  
Unnamed: 5               False  
Unnamed: 6               False  
Unnamed: 7               False  
Unnamed: 8               False  
Unnamed: 9               False  
Unnamed: 10              False  
Unnamed: 11              False  
Unnamed: 12              False  
Unnamed: 13              False  
Unnamed: 14              True  
Unnamed: 15              True  
Unnamed: 16              True  
Unnamed: 17              True  
Unnamed: 18              True  
Unnamed: 19              True  
Unnamed: 20              True  
Unnamed: 21              True  
Unnamed: 22              True  
Unnamed: 23              True  
Unnamed: 24              True  
Unnamed: 25              True  
Unnamed: 26              True  
dtype: bool
```

```
In [13]: #total num of null value in columns data  
df.isnull().sum()
```

```
Out[13]: pp                0  
Condition                0  
HR                    561  
RMSSD                 563  
SCL                   106  
Unnamed: 5              0  
Unnamed: 6              0  
Unnamed: 7              0  
Unnamed: 8              0  
Unnamed: 9              0  
Unnamed: 10             0  
Unnamed: 11             0  
Unnamed: 12             0  
Unnamed: 13             0  
Unnamed: 14             45  
Unnamed: 15             45  
Unnamed: 16             45  
Unnamed: 17             45  
Unnamed: 18             45  
Unnamed: 19             45  
Unnamed: 20             45  
Unnamed: 21             45  
Unnamed: 22             45  
Unnamed: 23             45  
Unnamed: 24             45  
Unnamed: 25             45  
Unnamed: 26             45  
dtype: int64
```

```
In [14]: #not null value and datatype
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3138 entries, 0 to 3137
Data columns (total 27 columns):
#   Column          Non-Null Count  Dtype
---  -
0   pp              3138 non-null   object
1   Condition       3138 non-null   int64
2   HR              2577 non-null   float64
3   RMSSD           2575 non-null   float64
4   SCL             3032 non-null   float64
5   Unnamed: 5      3138 non-null   float64
6   Unnamed: 6      3138 non-null   float64
7   Unnamed: 7      3138 non-null   float64
8   Unnamed: 8      3138 non-null   float64
9   Unnamed: 9      3138 non-null   float64
10  Unnamed: 10     3138 non-null   float64
11  Unnamed: 11     3138 non-null   float64
12  Unnamed: 12     3138 non-null   float64
13  Unnamed: 13     3138 non-null   float64
14  Unnamed: 14     3093 non-null   float64
15  Unnamed: 15     3093 non-null   float64
16  Unnamed: 16     3093 non-null   float64
17  Unnamed: 17     3093 non-null   float64
18  Unnamed: 18     3093 non-null   float64
19  Unnamed: 19     3093 non-null   float64
20  Unnamed: 20     3093 non-null   float64
21  Unnamed: 21     3093 non-null   float64
22  Unnamed: 22     3093 non-null   float64
23  Unnamed: 23     3093 non-null   float64
24  Unnamed: 24     3093 non-null   float64
25  Unnamed: 25     3093 non-null   float64
26  Unnamed: 26     3093 non-null   float64
dtypes: float64(25), int64(1), object(1)
memory usage: 662.0+ KB
```

```
In [15]: #rows and columns
print(df.shape)
```

```
(3138, 27)
```

Removing null values

```
In [16]: #delete null values from dataset
df = df.dropna(axis = 0, how = 'any')
```

```
In [17]: #after removing null values length of our new dataset
#print("Old data frame length:", len(df), "\nNew data frame length:", len(df1), "\n")
```

```
In [18]: print(df.shape)
```

```
(2502, 27)
```

```
In [22]: df.columns=['PP', 'Condition', 'HR', 'RMSSD', 'SCL', 'Squality', 'Sneutral', 'Shappy',
                    'SxHeadOrientation', 'SzHeadOrientation', 'SmouthOpen', 'SleftEyeClosed',
                    'SrightEyebrowLowered', 'SrightEyebrowRaised', 'SgazeDirectionForward',
df['Condition'].unique()
```

```
Out[22]: array([0, 1], dtype=int64)
```

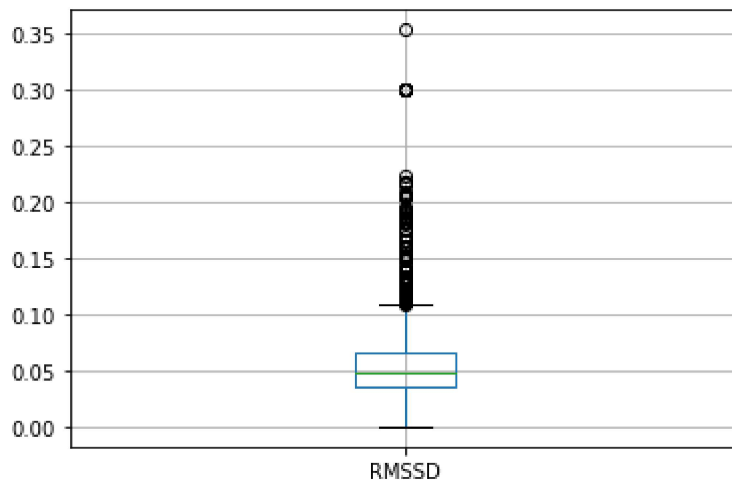
Type Conversion

```
In [24]: df['PP'] = pd.to_numeric(df['PP'], errors='coerce')
print(df.dtypes)
```

```
PP                                float64
Condition                        int64
HR                                float64
RMSSD                            float64
SCL                              float64
Squality                         float64
Sneutral                         float64
Shappy                           float64
Ssad                             float64
Sangry                           float64
Ssurprised                       float64
Sscared                          float64
Sdisgusted                       float64
Svalence                         float64
SyHeadOrientation                float64
SxHeadOrientation                float64
SzHeadOrientation                float64
SmouthOpen                       float64
SleftEyeClosed                  float64
SrightEyeClosed                  float64
SleftEyebrowLowered             float64
SleftEyebrowRaised              float64
SrightEyebrowLowered            float64
SrightEyebrowRaised              float64
SgazeDirectionForward            float64
SgazeDirectionLeft               float64
SgazeDirectionRight              float64
dtype: object
```

```
In [25]: df.boxplot(column=['RMSSD'])
```

```
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x2ee60e8c820>
```



extra

```
In [33]: grouped = df.groupby(['Condition'])  
grouped.agg(np.sum)  
grouped.size()
```

```
Out[33]: Condition  
0      1150  
1      1352  
dtype: int64
```

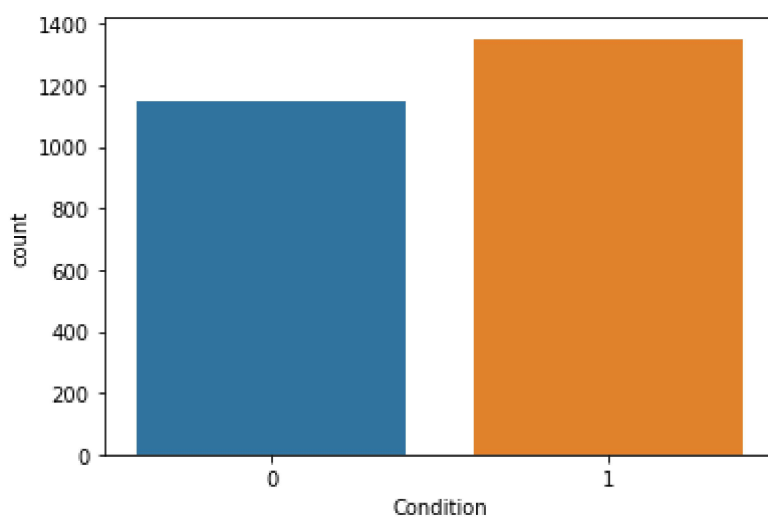
-----DATA VISUALIZATION ----

```
In [37]: sns.countplot(df["Condition"])  
print(df.Condition.value_counts())
```

```
1    1352
```

```
0    1150
```

```
Name: Condition, dtype: int64
```



HR RANGE

```

In [27]: p=df['HR'].max()
q=df['HR'].min()
print("Minimum::",q)
print("Maximum::",p)
r=p-q
print("Range::",r)
n=3
a=df['HR'].value_counts(bins=n)# all HR data
print("\n\n-----Stress or Non Stress HR Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a1=df_filtered_1['HR'].value_counts(bins=n)#stress
a0=df_filtered_0['HR'].value_counts(bins=n)#non stress
print("\n-----Stress Data HR-----")
print(a1)
a1.plot()

```

```

Minimum:: 57.0
Maximum:: 107.0
Range:: 50.0

```

```

-----Stress or Non Stress HR Data-----
(56.949000000000005, 73.667]    1367
(73.667, 90.333]              997
(90.333, 107.0]              138
Name: HR, dtype: int64

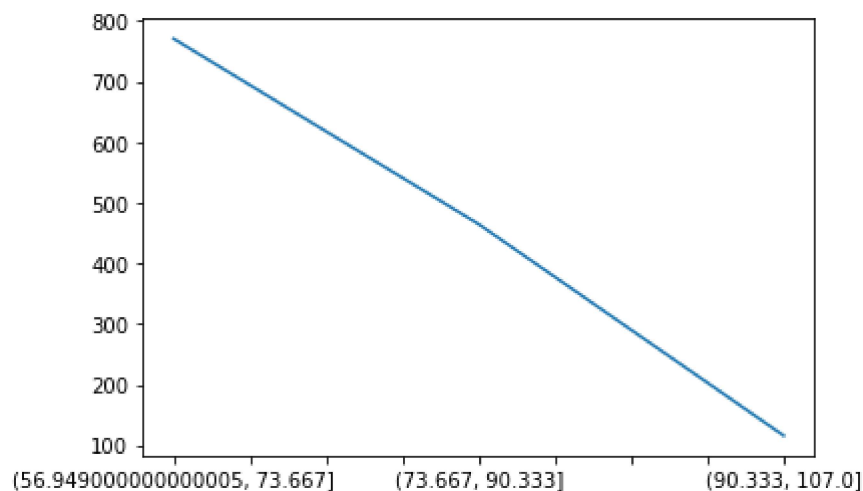
-----Stress Data HR-----
(56.949000000000005, 73.667]    771
(73.667, 90.333]              465
(90.333, 107.0]              116
Name: HR, dtype: int64

```

```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x2ee616881f0>

```



HRV RANGE


```

In [28]: p=df['RMSSD'].max()
q=df['RMSSD'].min()
print("Minimum::",q)
print("Maximum::",p)
r=p-q
print("Range::",r,"\n\n")
n=3
a=df['RMSSD'].value_counts(bins=n)#all HRV/RMSSD data
print("-----Stress or Non Stress HRV/RMSSD Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a1=df_filtered_1['RMSSD'].value_counts(bins=n)#stress data
a0=df_filtered_0['RMSSD'].value_counts(bins=n)#Non stress
print("\n-----Stress HRV/RMSSD Data-----")
print(a1)
a1.plot()

```

```

Minimum:: 0.0
Maximum:: 0.352978
Range:: 0.352978

```

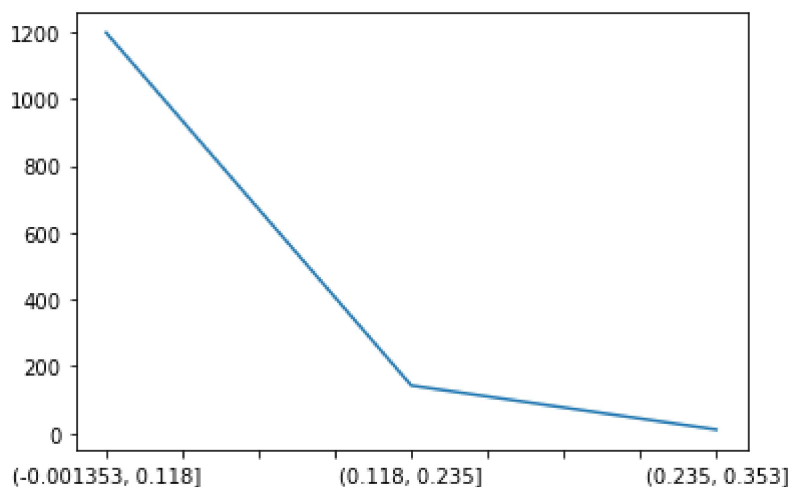
```

-----Stress or Non Stress HRV/RMSSD Data-----
(-0.001353, 0.118]      2319
(0.118, 0.235]          172
(0.235, 0.353]          11
Name: RMSSD, dtype: int64

-----Stress HRV/RMSSD Data-----
(-0.001353, 0.118]      1198
(0.118, 0.235]          143
(0.235, 0.353]          11
Name: RMSSD, dtype: int64

```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x2ee616fdc10>



GSR DATA RANGE

```
In [29]: p=df['SCL'].max()
q=df['SCL'].min()
print("Minimum::",q)
print("Maximum::",p)
r=p-q
print("Range::",r,"\n\n")
n=3
a=df['SCL'].value_counts(bins=n)#all gsr/scl data
print("-----Stress or Non Stress SCL Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SCL'].value_counts(bins=n)#non stress
a1=df_filtered_1['SCL'].value_counts(bins=n)#non stress
print("\n-----Stress SCL Data-----")
print(a1)
a1.plot()
```

Minimum:: 53.252680000000005

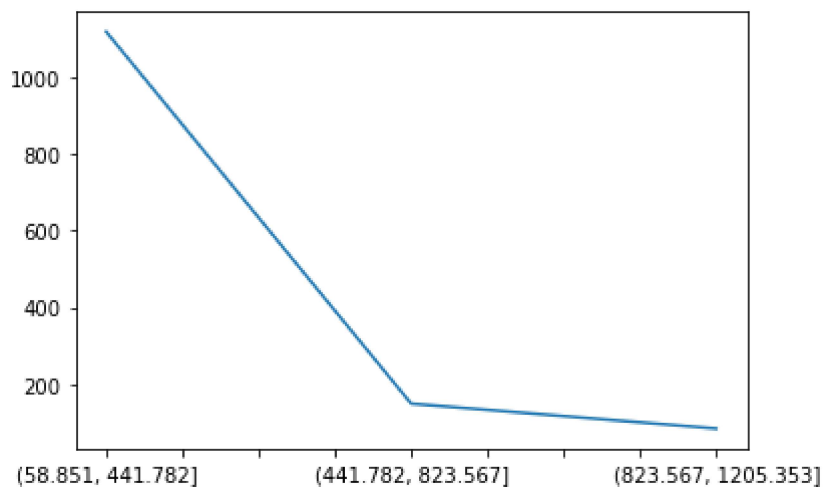
Maximum:: 1455.23043

Range:: 1401.97775

```
-----Stress or Non Stress SCL Data-----
(51.85, 520.579]      2264
(520.579, 987.905]    188
(987.905, 1455.23]    50
Name: SCL, dtype: int64
```

```
-----Stress SCL Data-----
(58.851, 441.782]      1119
(441.782, 823.567]     149
(823.567, 1205.353]     84
Name: SCL, dtype: int64
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x2ee617403a0>



```

In [53]: n=12
a=df['Ssad'].value_counts(bins=n)#all gsr/scl data
print("-----Sad Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['Ssad'].value_counts(bins=n)#non stress
a1=df_filtered_1['Ssad'].value_counts(bins=n)#non stress
print("\n::::::::::::::::Non Stress Data::::::::::::::::")
print(a0)

n=10
a=df['SleftEyeClosed'].value_counts(bins=n)#all gsr/scl data
print("-----Left Eyes Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SleftEyeClosed'].value_counts(bins=n)#non stress
a1=df_filtered_1['SleftEyeClosed'].value_counts(bins=n)#non stress
print("\n::::::::::::::::Non Stress Data::::::::::::::::")
print(a0)

n=4
a=df['SrighEyeClosed'].value_counts(bins=n)#all gsr/scl data
print("\n-----Right Eyes Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SrighEyeClosed'].value_counts(bins=n)#non stress
a1=df_filtered_1['SrighEyeClosed'].value_counts(bins=n)#non stress
print("\n::::::::::::::::Stress Data::::::::::::::::")
print(a1)

n=8
a=df['SleftEyebrowLowered'].value_counts(bins=n)#all gsr/scl data
print("-----Left Eyes Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SleftEyebrowLowered'].value_counts(bins=n)#non stress
a1=df_filtered_1['SleftEyebrowLowered'].value_counts(bins=n)#non stress
print("\n::::::::::::::::Non Stress Data::::::::::::::::")
print(a0)

a=df['SleftEyebrowRaised'].value_counts(bins=n)#all gsr/scl data
print("\n-----Right Eyes Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SleftEyebrowRaised'].value_counts(bins=n)#non stress
a1=df_filtered_1['SleftEyebrowRaised'].value_counts(bins=n)#non stress
print("\n::::::::::::::::Non Stress Data::::::::::::::::")
print(a0)

a=df['SrighEyebrowLowered'].value_counts(bins=n)#all gsr/scl data
print("-----Left Eyes Data-----")

```

```

print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SrightEyebrowLowered'].value_counts(bins=n)#non stress
a1=df_filtered_1['SrightEyebrowLowered'].value_counts(bins=n)#non stress
print("\n::::::::::::::::::::::::Non Stress Data::::::::::::::::::::::::")
print(a0)

a=df['SrightEyebrowRaised'].value_counts(bins=n)#all gsr/scl data
print("\n-----Right Eyes Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['SrightEyebrowRaised'].value_counts(bins=n)#non stress
a1=df_filtered_1['SrightEyebrowRaised'].value_counts(bins=n)#non stress
print("\n::::::::::::::::::::::::Non Stress Data::::::::::::::::::::::::")
print(a0)

n=15
a=df['Svalence'].value_counts(bins=n)#all gsr/scl data
print("\n-----Svalence-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['Svalence'].value_counts(bins=n)#non stress
a1=df_filtered_1['Svalence'].value_counts(bins=n)#non stress
print("\n::::::::::::::::::::::::Non Stress Data::::::::::::::::::::::::")
print(a0)

a=df['Shappy'].value_counts(bins=n)#all gsr/scl data
print("\n-----Happy Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['Shappy'].value_counts(bins=n)#non stress
a1=df_filtered_1['Shappy'].value_counts(bins=n)#non stress
print("\n::::::::::::::::::::::::Non Stress Data::::::::::::::::::::::::\n")
print(a0)

n=10
a=df['Sangry'].value_counts(bins=n)#all gsr/scl data
print("\n-----Sad Data-----")
print(a)
df_filtered_1 = df.query('Condition == 1')#stress
df_filtered_0 = df.query('Condition == 0')#non stress
a0=df_filtered_0['Sangry'].value_counts(bins=n)#non stress
a1=df_filtered_1['Sangry'].value_counts(bins=n)#non stress
print("\n::::::::::::::::::::::::Non Stress Data::::::::::::::::::::::::")
print(a0)

```

```

-----Sad Data-----
(-0.0016970000000000002, 0.0581]    1762
(0.0581, 0.116]                      428
(0.116, 0.174]                       174
(0.174, 0.232]                       73
(0.232, 0.29]                        36
(0.29, 0.348]                        20

```

```
(0.464, 0.522]      3
(0.348, 0.406]      3
(0.406, 0.464]      2
(0.639, 0.697]      1
(0.581, 0.639]      0
(0.522, 0.581]      0
Name: Ssad, dtype: int64
```

```
::::::::::::::::Non Stress Data::::::::::::::::
```

```
(-0.001697000000000002, 0.0581]    814
(0.0581, 0.116]                    192
(0.116, 0.174]                     77
(0.174, 0.232]                     36
(0.232, 0.29]                      15
(0.29, 0.348]                      10
(0.464, 0.522]                      2
(0.348, 0.406]                      2
(0.639, 0.697]                      1
(0.406, 0.464]                      1
(0.581, 0.639]                      0
(0.522, 0.581]                      0
```

```
Name: Ssad, dtype: int64
```

```
-----Left Eyes Data-----
```

```
(-0.00243, 0.143]    1773
(0.143, 0.287]       347
(0.287, 0.43]        164
(0.43, 0.573]        105
(0.573, 0.716]        45
(0.716, 0.86]         35
(0.86, 1.003]         31
(1.289, 1.433]         2
(1.146, 1.289]         0
(1.003, 1.146]         0
```

```
Name: SleftEyeClosed, dtype: int64
```

```
::::::::::::::::Non Stress Data::::::::::::::::
```

```
(-0.00243, 0.143]    844
(0.143, 0.287]       133
(0.287, 0.43]        69
(0.43, 0.573]        50
(0.573, 0.716]        26
(0.716, 0.86]        21
(0.86, 1.003]         5
(1.289, 1.433]         2
(1.146, 1.289]         0
(1.003, 1.146]         0
```

```
Name: SleftEyeClosed, dtype: int64
```

```
-----Right Eyes Data-----
```

```
(-0.001979, 0.245]    2117
(0.245, 0.489]        267
(0.489, 0.734]        82
(0.734, 0.979]        36
```

```
Name: SrightEyeClosed, dtype: int64
```

```
::::::::::::::::Stress Data::::::::::::::::
```

```
(-0.001979, 0.245]    1147
```

```
(0.245, 0.489]      140
(0.489, 0.734]      35
(0.734, 0.979]      30
Name: SrightEyeClosed, dtype: int64
-----Left Eyes Data-----
(-0.001983, 0.123]  1519
(0.123, 0.246]      262
(0.246, 0.369]      243
(0.369, 0.491]      173
(0.491, 0.614]      146
(0.614, 0.737]      81
(0.737, 0.86]       51
(0.86, 0.983]       27
Name: SleftEyebrowLowered, dtype: int64

::::::::::::::::::::Non Stress Data::::::::::::::::::::
(-0.00195, 0.119]   682
(0.237, 0.356]      111
(0.119, 0.237]      103
(0.356, 0.475]      81
(0.475, 0.593]      76
(0.593, 0.712]      42
(0.712, 0.831]      36
(0.831, 0.95]       19
Name: SleftEyebrowLowered, dtype: int64

-----Right Eyes Data-----
(-0.001996, 0.124]  1642
(0.124, 0.249]      247
(0.249, 0.373]      168
(0.373, 0.498]      126
(0.498, 0.622]      95
(0.747, 0.871]      81
(0.871, 0.996]      73
(0.622, 0.747]      70
Name: SleftEyebrowRaised, dtype: int64

::::::::::::::::::::Non Stress Data::::::::::::::::::::
(-0.001996, 0.124]  767
(0.124, 0.249]      124
(0.249, 0.373]      80
(0.373, 0.498]      52
(0.871, 0.996]      36
(0.747, 0.871]      33
(0.622, 0.747]      29
(0.498, 0.622]      29
Name: SleftEyebrowRaised, dtype: int64
-----Left Eyes Data-----
(-0.001996, 0.124]  1772
(0.124, 0.249]      243
(0.249, 0.373]      149
(0.373, 0.498]      109
(0.871, 0.996]      97
(0.498, 0.622]      59
(0.747, 0.871]      38
(0.622, 0.747]      35
Name: SrightEyebrowLowered, dtype: int64
```

::::::::::::::::::::Non Stress Data::::::::::::::::::::

(-0.001996, 0.124]	805
(0.124, 0.249]	111
(0.249, 0.373]	77
(0.373, 0.498]	51
(0.498, 0.622]	27
(0.622, 0.747]	15

Name: SrightEyebrowLowered, dtype: int64

-----Right Eyes Data-----

(-0.00214, 0.143]	2022
(0.143, 0.286]	140
(0.286, 0.429]	57
(0.429, 0.572]	64
(0.572, 0.715]	48
(0.715, 0.858]	48
(0.858, 1.001]	121
(1.001, 1.144]	2

Name: SrightEyebrowRaised, dtype: int64

::::::::::::::::::::Non Stress Data::::::::::::::::::::

(-0.00214, 0.143]	947
(0.143, 0.286]	78
(0.286, 0.429]	17
(0.429, 0.572]	23
(0.572, 0.715]	13
(0.715, 0.858]	12
(0.858, 1.001]	58
(1.001, 1.144]	2

Name: SrightEyebrowRaised, dtype: int64

-----Svalence-----

(-0.0294, 0.0542]	1117
(0.0542, 0.138]	99
(0.138, 0.221]	32
(0.221, 0.305]	17
(0.305, 0.389]	11
(0.389, 0.472]	2
(0.472, 0.556]	1
(0.556, 0.64]	1
(0.64, 0.724]	1
(0.724, 0.808]	1
(0.808, 0.892]	1
(0.892, 0.976]	1
(0.976, 1.06]	1
(1.06, 1.144]	1
(1.144, 1.228]	1
(1.228, 1.312]	1
(1.312, 1.396]	1
(1.396, 1.48]	1
(1.48, 1.564]	1
(1.564, 1.648]	1
(1.648, 1.732]	1
(1.732, 1.816]	1
(1.816, 1.9]	1
(1.9, 1.984]	1
(1.984, 2.068]	1
(2.068, 2.152]	1
(2.152, 2.236]	1
(2.236, 2.32]	1
(2.32, 2.404]	1
(2.404, 2.488]	1
(2.488, 2.572]	1
(2.572, 2.656]	1
(2.656, 2.74]	1
(2.74, 2.824]	1
(2.824, 2.908]	1
(2.908, 3.0]	1
(3.0, 3.084]	1
(3.084, 3.168]	1
(3.168, 3.252]	1
(3.252, 3.336]	1
(3.336, 3.42]	1
(3.42, 3.504]	1
(3.504, 3.588]	1
(3.588, 3.672]	1
(3.672, 3.756]	1
(3.756, 3.84]	1
(3.84, 3.924]	1
(3.924, 4.008]	1
(4.008, 4.092]	1
(4.092, 4.176]	1
(4.176, 4.26]	1
(4.26, 4.344]	1
(4.344, 4.428]	1
(4.428, 4.512]	1
(4.512, 4.596]	1
(4.596, 4.68]	1
(4.68, 4.764]	1
(4.764, 4.848]	1
(4.848, 4.932]	1
(4.932, 5.016]	1
(5.016, 5.1]	1
(5.1, 5.184]	1
(5.184, 5.268]	1
(5.268, 5.352]	1
(5.352, 5.436]	1
(5.436, 5.52]	1
(5.52, 5.604]	1
(5.604, 5.688]	1
(5.688, 5.772]	1
(5.772, 5.856]	1
(5.856, 5.94]	1
(5.94, 6.024]	1
(6.024, 6.108]	1
(6.108, 6.192]	1
(6.192, 6.276]	1
(6.276, 6.36]	1
(6.36, 6.444]	1
(6.444, 6.528]	1
(6.528, 6.612]	1
(6.612, 6.696]	1
(6.696, 6.78]	1
(6.78, 6.864]	1
(6.864, 6.948]	1
(6.948, 7.032]	1
(7.032, 7.116]	1
(7.116, 7.2]	1
(7.2, 7.284]	1
(7.284, 7.368]	1
(7.368, 7.452]	1
(7.452, 7.536]	1
(7.536, 7.62]	1
(7.62, 7.704]	1
(7.704, 7.788]	1
(7.788, 7.872]	1
(7.872, 7.956]	1
(7.956, 8.04]	1
(8.04, 8.124]	1
(8.124, 8.208]	1
(8.208, 8.292]	1
(8.292, 8.376]	1
(8.376, 8.46]	1
(8.46, 8.544]	1
(8.544, 8.628]	1
(8.628, 8.712]	1
(8.712, 8.796]	1
(8.796, 8.88]	1
(8.88, 8.964]	1
(8.964, 9.048]	1
(9.048, 9.132]	1
(9.132, 9.216]	1
(9.216, 9.3]	1
(9.3, 9.384]	1
(9.384, 9.468]	1
(9.468, 9.552]	1
(9.552, 9.636]	1
(9.636, 9.72]	1
(9.72, 9.804]	1
(9.804, 9.888]	1
(9.888, 9.972]	1
(9.972, 10.056]	1
(10.056, 10.14]	1
(10.14, 10.224]	1
(10.224, 10.308]	1
(10.308, 10.392]	1
(10.392, 10.476]	1
(10.476, 10.56]	1
(10.56, 10.644]	1
(10.644, 10.728]	1
(10.728, 10.812]	1
(10.812, 10.896]	1
(10.896, 10.98]	1
(10.98, 11.064]	1
(11.064, 11.148]	1
(11.148, 11.232]	1
(11.232, 11.316]	1
(11.316, 11.4]	1
(11.4, 11.484]	1
(11.484, 11.568]	1
(11.568, 11.652]	1
(11.652, 11.736]	1
(11.736, 11.82]	1
(11.82, 11.904]	1
(11.904, 11.988]	1
(11.988, 12.072]	1
(12.072, 12.156]	1
(12.156, 12.24]	1
(12.24, 12.324]	1
(12.324, 12.408]	1
(12.408, 12.492]	1
(12.492, 12.576]	1
(12.576, 12.66]	1
(12.66, 12.744]	1
(12.744, 12.828]	1
(12.828, 12.912]	1
(12.912, 13.0]	1
(13.0, 13.084]	1
(13.084, 13.168]	1
(13.168, 13.252]	1
(13.252, 13.336]	1
(13.336, 13.42]	1
(13.42, 13.504]	1
(13.504, 13.588]	1
(13.588, 13.672]	1
(13.672, 13.756]	1
(13.756, 13.84]	1
(13.84, 13.924]	1
(13.924, 14.008]	1
(14.008, 14.092]	1
(14.092, 14.176]	1
(14.176, 14.26]	1
(14.26, 14.344]	1
(14.344, 14.428]	1
(14.428, 14.512]	1
(14.512, 14.596]	1
(14.596, 14.68]	1
(14.68, 14.764]	1
(14.764, 14.848]	1
(14.848, 14.932]	1
(14.932, 15.016]	1
(15.016, 15.1]	1
(15.1, 15.184]	1
(15.184, 15.268]	1
(15.268, 15.352]	1
(15.352, 15.436]	1
(15.436, 15.52]	1
(15.52, 15.604]	1
(15.604, 15.688]	1
(15.688, 15.772]	1
(15.772, 15.856]	1
(15.856, 15.94]	1
(15.94, 16.024]	1
(16.024, 16.108]	1
(16.108, 16.192]	1
(16.192, 16.276]	1
(16.276, 16.36]	1
(16.36, 16.444]	1
(16.444, 16.528]	1
(16.528, 16.612]	1
(16.612, 16.696]	1
(16.696, 16.78]	1
(16.78, 16.864]	1
(16.864, 16.948]	1
(16.948, 17.032]	1
(17.032, 17.116]	1
(17.116, 17.2]	1
(17.2, 17.284]	1
(17.284, 17.368]	1
(17.368, 17.452]	1
(17.452, 17.536]	1
(17.536, 17.62]	1
(17.62, 17.704]	1
(17.704, 17.788]	1
(17.788, 17.872]	1
(17.872, 17.956]	1
(17.956, 18.04]	1
(18.04, 18.124]	1
(18.124, 18.208]	1
(18.208, 18.292]	1
(18.292, 18.376]	1
(18.376, 18.46]	1
(18.46, 18.544]	1
(18.544, 18.628]	1
(18.628, 18.712]	1
(18.712, 18.796]	1
(18.796, 18.88]	1
(18.88, 18.964]	1
(18.964, 19.048]	1
(19.048, 19.132]	1
(19.132, 19.216]	1
(19.216, 19.3]	1
(19.3, 19.384]	1
(19.384, 19.468]	1
(19.468, 19.552]	1
(19.552, 19.636]	1
(19.636, 19.72]	1
(19.72, 19.804]	1
(19.804, 19.888]	1
(19.888, 19.972]	1
(19.972, 20.056]	1
(20.056, 20.14]	1
(20.14, 20.224]	1
(20.224, 20.308]	1
(20.308, 20.392]	1
(20.392, 20.476]	1
(20.476, 20.56]	1
(20.56, 20.644]	1
(20.644, 20.728]	1
(20.728, 20.812]	1
(20.812, 20.896]	1
(20.896, 20.98]	1
(20.98, 21.064]	1
(21.064, 21.148]	1
(21.148, 21.232]	1
(21.232, 21.316]	1
(21.316, 21.4]	1
(21.4, 21.484]	1
(21.484, 21.568]	1
(21.568, 21.652]	1
(21.652, 21.736]	1
(21.736, 21.82]	1
(21.82, 21.904]	1
(21.904, 21.988]	1
(21.988, 22.072]	1
(22.072, 22.156]	1
(22.156, 22.24]	1
(22.24, 22.324]	1
(22.324, 22.408]	1
(22.408, 22.492]	1
(22.492, 22.576]	1
(22.576, 22.66]	1
(22.66, 22.744]	1
(22.744, 22.828]	1
(22.828, 22.912]	1
(22.912, 23.0]	1
(23.0, 23.084]	1
(23.084, 23.168]	1
(23.168, 23.252]	1
(23.252, 23.336]	1
(23.336, 23.42]	1
(23.42, 23.504]	1
(23.504, 23.588]	1
(23.588, 23.672]	1
(23.672, 23.756]	1
(23.756, 23.84]	1
(23.84, 23.924]	1
(23.924, 24.008]	1
(24.008, 24.092]	1
(24.092, 24.176]	1
(24.176, 24.26]	1
(24.26, 24.344]	1
(24.344, 24.428]	1
(24.428, 24.512]	1
(24.512, 24.596]	1
(24.596, 24.68]	1
(24.68, 24.764]	1
(24.764, 24.848]	1
(24.848, 24.932]	1
(24.932, 25.016]	1
(25.016, 25.1]	1
(25.1, 25.184]	1
(25.184, 25.268]	1
(25.268, 25.352]	1
(25.352, 25.436]	1
(25.436, 25.52]	1
(25.52, 25.604]	1
(25.604, 25.688]	1
(25.688, 25.772]	1
(25.772, 25.856]	1
(25.856, 25.94]	1
(25.94, 26.024]	1
(26.024, 26.108]	1
(26.108, 26.192]	1
(26.192, 26.276]	1
(26.276, 26.36]	1
(26.36, 26.444]	1
(26.444, 26.528]	1
(26.528, 26.612]	1
(26.612, 26.696]	1
(26.696, 26.78]	1
(26.78, 26.864]	1
(26.864, 26.948]	1
(26.948, 27.032]	1
(27.032, 27.116]	1
(27.116, 27.2]	1
(27.2, 27.284]	1
(27.284, 27.368]	1
(27.368, 27.452]	1
(27.452, 27.536]	1
(27.536, 27.62]	1
(27.62, 27.704]	1
(27.704, 27.788]	1
(27.788, 27.872]	1
(27.872, 27.956]	1
(27.956, 28.04]	1
(28.04, 28.124]	1
(28.124, 28.208]	1
(28.208, 28.292]	1
(28.292, 28.376]	1
(28.376, 28.46]	1
(28.46, 28.544]	1
(28.544, 28.628]	1
(28.628, 28.712]	1
(28.712, 28.796]	1
(28.796, 28.88]	1
(28.88, 28.964]	1
(28.964, 29.048]	1
(29.048, 29.132]	1
(29.132, 29.216]	1
(29.216, 29.3]	1
(29.3, 29.384]	1
(29.384, 29.468]	1
(29.468, 29.552]	1
(29.552, 29.636]	1
(29.636, 29.72]	1
(29.72, 29.804]	1
(29.804, 29.888]	1
(29.888, 29.972]	1
(29.972, 30.056]	1
(30.056, 30.14]	1
(30.14, 30.224]	1
(30.224, 30.308]	1
(30.308, 30.392]	1
(30.392, 30.476]	1
(30.476, 30.56]	1
(30.56, 30.644]	1
(30.644, 30.728]	1
(30.728, 30.812]	1
(30.812, 30.896]	1
(30.896, 30.98]	1
(30.98, 31.064]	1
(31.064, 31.148]	1
(31.148, 31.232]	1
(31.232, 31.316]	1
(31.316, 31.4]	1
(31.4, 31.484]	1
(31.484, 31.568]	1
(31.568, 31.652]	1
(31.652, 31.736]	1
(31.736, 31.82]	1
(31.82, 31.904]	1
(31.904, 31.988]	1
(31.988, 32.072]	1
(32.072, 32.156]	1
(32.156, 32.24]	1
(32.24, 32.324]	1
(32.324, 32.408]	1
(32.408, 32.492]	1
(32.492, 32.576]	1
(32.576, 32.66]	1
(32.66, 32.744]	1
(32.744, 32.828]	1
(32.828, 32.912]	1
(32.912, 33.0]	1
(33.0, 33.084]	1
(33.084, 33.168]	1
(33.168, 33.252]	1
(33.252, 33.336]	1
(33.336, 33.42]	1
(33.42, 33.504]	1
(33.504, 33.588]	1
(33.588, 33.672]	1
(33.672, 33.756]	1
(33.756	

```
(-0.28, -0.197]      61
(0.138, 0.221]      23
(-0.364, -0.28]     16
(0.221, 0.305]      13
(0.305, 0.389]      10
(-0.448, -0.364]      3
(0.389, 0.472]       2
(-0.531, -0.448]      2
(0.556, 0.64]        1
(0.472, 0.556]        1
(-0.617, -0.531]      1
Name: Svalence, dtype: int64
```

```
-----Happy Data-----
(-0.001923, 0.0615]   2163
(0.0615, 0.123]      204
(0.123, 0.185]       57
(0.185, 0.246]       32
(0.308, 0.369]       19
(0.246, 0.308]       19
(0.677, 0.738]        2
(0.369, 0.431]        2
(0.861, 0.923]        1
(0.615, 0.677]        1
(0.554, 0.615]        1
(0.431, 0.492]        1
(0.8, 0.861]          0
(0.738, 0.8]          0
(0.492, 0.554]        0
Name: Shappy, dtype: int64
```

```
.....Non Stress Data.....
(-0.001923, 0.0615]   919
(0.0615, 0.123]      119
(0.123, 0.185]       45
(0.185, 0.246]       26
(0.246, 0.308]       18
(0.308, 0.369]       15
(0.677, 0.738]        2
(0.369, 0.431]        2
(0.861, 0.923]        1
(0.615, 0.677]        1
(0.554, 0.615]        1
(0.431, 0.492]        1
(0.8, 0.861]          0
(0.738, 0.8]          0
(0.492, 0.554]        0
Name: Shappy, dtype: int64
```

```
-----Sad Data-----
(-0.0013210000000000001, 0.0321]   1951
(0.0321, 0.0642]                   276
(0.0642, 0.0963]                   116
(0.0963, 0.128]                     71
(0.128, 0.16]                       31
(0.16, 0.193]                       24
```



```
(0.193, 0.225] 18
(0.225, 0.257] 10
(0.289, 0.321] 3
(0.257, 0.289] 2
Name: Sangry, dtype: int64

::::::::::::::::::::Non Stress Data::::::::::::::::::::
(-0.001321000000000001, 0.0321] 924
(0.0321, 0.0642] 114
(0.0642, 0.0963] 34
(0.0963, 0.128] 31
(0.16, 0.193] 13
(0.128, 0.16] 13
(0.193, 0.225] 11
(0.225, 0.257] 7
(0.289, 0.321] 2
(0.257, 0.289] 1
Name: Sangry, dtype: int64
```

----- PRE MODELING TASK -----

SPLITTING DATASET

```
In [48]: from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier, NeighborhoodComponentsAnalysis
from sklearn.decomposition import PCA

X = df.drop(['Condition'], 1)
Y = df['Condition']
#to standardize the range
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state=42)
print("X_train", len(X_train))
print("X_test", len(X_test))
print("Y_train", len(Y_train))
print("Y_test", len(Y_test))
```

X_train 2001

X_test 501

Y_train 2001

Y_test 501

C:\Users\Drashti\anaconda3\lib\site-packages\sklearn\utils\extmath.py:770: RuntimeWarning: invalid value encountered in true_divide

updated_mean = (last_sum + new_sum) / updated_sample_count

C:\Users\Drashti\anaconda3\lib\site-packages\sklearn\utils\extmath.py:711: RuntimeWarning: Degrees of freedom <= 0 for slice.

result = op(x, *args, **kwargs)