```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

#Loading the dataset

```
df=pd.read_csv('/content/Dataset-Mental-Disorders.csv')

print(df.head())

  Patient Number    Sadness    Euphoric  Exhausted Sleep dissorder
Mood Swing  \
0     Patiant-01    Usually      Seldom  Sometimes       Sometimes
YES
1     Patiant-02    Usually      Seldom    Usually       Sometimes
NO
2     Patiant-03  Sometimes  Most-Often  Sometimes       Sometimes
YES
3     Patiant-04    Usually      Seldom    Usually      Most-Often
YES
4     Patiant-05    Usually     Usually  Sometimes       Sometimes
NO

  Suicidal thoughts Anorxia Authority Respect Try-Explanation  \
0              YES      NO                NO             YES
1               YES      NO                NO              NO
2                NO      NO                NO             YES
3               YES     YES                NO             YES
4                NO      NO                NO              NO

  Aggressive Response Ignore & Move-On Nervous Break-down Admit
Mistakes  \
0                  NO               NO                YES
YES
1                  NO               NO                 NO
NO
2                 YES               NO                YES
YES
3                  NO               NO                 NO
NO
4                  NO               NO                YES
YES
```

```
  Overthinking Sexual Activity Concentration  Optimisim Expert
Diagnose
0          YES       3 From 10     3 From 10  4 From 10  Bipolar Type-
2
1           NO       4 From 10     2 From 10  5 From 10
Depression
2           NO       6 From 10     5 From 10  7 From 10  Bipolar Type-
1
3           NO       3 From 10     2 From 10  2 From 10  Bipolar Type-
2
4          YES       5 From 10     5 From 10  6 From 10
Normal
```

```
df.info
```

```
<bound method DataFrame.info of      Sadness  Euphoric  Exhausted
Sleep Disorder  Mood Swing  \
0        2.0       1.0        1.0             1.0          1
1        2.0       1.0        2.0             1.0          0
2        1.0       3.0        1.0             1.0          1
3        2.0       1.0        2.0             3.0          1
4        2.0       2.0        1.0             1.0          0
..       ...       ...        ...             ...        ...
115      3.0       1.0        2.0             1.0          0
116      1.0       1.0        1.0             1.0          1
117      2.0       1.0        2.0             1.0          1
118      2.0       1.0        1.0             1.0          0
119      1.0       2.0        1.0             2.0          0

     Suicidal thoughts  Anorxia  Authority Respect  Try-Explanation  \
0                  0.0        0                  0                1
1                  1.0        0                  0                0
2                  0.0        0                  0                1
3                  1.0        1                  0                1
4                  0.0        0                  0                0
..                 ...      ...                ...              ...
115                1.0        0                  0                1
116                0.0        0                  0                0
117                0.0        1                  1                0
118                1.0        1                  0                1
119                0.0        0                  1                1

     Aggressive Response  Ignore & Move-On  Nervous Break-down  \
0                      0                 0                   1
1                      0                 0                   0
2                      1                 0                   1
3                      0                 0                   0
4                      0                 0                   1
..                   ...               ...                 ...
115                    0                 1                   0
```

```
116                    1                 0                   0
117                    0                 0                   1
118                    1                 1                   0
119                    0                 1                   1

     Admit Mistakes  Overthinking  Sexual Activity  Concentration
Optimisim  \
0                  1             1              3.0            3.0
4.0
1                  0             0              4.0            2.0
5.0
2                  1             0              6.0            5.0
7.0
3                  0             0              3.0            2.0
2.0
4                  1             1              5.0            5.0
6.0
..               ...           ...              ...            ...
...
115                0             1              2.0            5.0
3.0
116                0             1              6.0            7.0
8.0
117                0             1              1.0            5.0
3.0
118                1             1              7.0            7.0
7.0
119                0             0              7.0            3.0
8.0

     Expert Diagnose
0                  1
1                  2
2                  0
3                  1
4                  3
..               ...
115                2
116                0
117                1
118                2
119                3

[120 rows x 18 columns]>
```

```
df.dtypes
```

```
Sadness                float64
Euphoric               float64
Exhausted              float64
```

```
Sleep Disorder         float64
Mood Swing               int64
Suicidal thoughts      float64
Anorxia                  int64
Authority Respect        int64
Try-Explanation          int64
Aggressive Response      int64
Ignore & Move-On         int64
Nervous Break-down       int64
Admit Mistakes           int64
Overthinking             int64
Sexual Activity        float64
Concentration          float64
Optimisim              float64
Expert Diagnose          int64
dtype: object
```

```
df.isnull().sum()
```

```
Sadness                0
Euphoric               0
Exhausted              0
Sleep Disorder         0
Mood Swing             0
Suicidal thoughts      0
Anorxia                0
Authority Respect      0
Try-Explanation        0
Aggressive Response    0
Ignore & Move-On       0
Nervous Break-down     0
Admit Mistakes         0
Overthinking           0
Sexual Activity        0
Concentration          0
Optimisim              0
Expert Diagnose        0
dtype: int64
```

Drop useless columns (Cause we don’t need Patient Number for predictions)

```
df.drop(columns=["Patient Number"], inplace=True)
```

Renaming columns

```
df.rename(columns={"Sleep dissorder": "Sleep Disorder"}, inplace=True)
```

```python
from sklearn.preprocessing import LabelEncoder

df = df.dropna(thresh=len(df) * 0.5, axis=1)

for col in df.columns:
    if df[col].dtype == "object":
        df[col].fillna(df[col].mode()[0], inplace=True)
    else:
        df[col].fillna(df[col].median(), inplace=True)
```

```
<ipython-input-31-8300b7822a28>:5: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  df[col].fillna(df[col].median(), inplace=True)
```

Convert YES/NO and other categorical values into numbers because ML models don’t speak human

```python
le = LabelEncoder()
for col in df.select_dtypes(include=["object"]).columns:
    df[col] = le.fit_transform(df[col])

df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 120,\n  \"fields\": [\
n    {\n      \"column\": \"Sadness\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.7445412928069721,\n
\"min\": 1.0,\n        \"max\": 3.0,\n        \"num_unique_values\":
3,\n        \"samples\": [\n          2.0,\n          1.0,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Euphoric\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.6078386193019043,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Exhausted\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8217968970834723,\n        \"min\": 1.0,\n        \"max\": 3.0,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1.0,\n
2.0,\n          3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sleep Disorder\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.766234197839261,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Mood Swing\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Suicidal thoughts\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.500979432868119,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1.0,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Anorxia\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Authority Respect\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Try-Explanation\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Aggressive Response\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Ignore & Move-On\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Nervous Break-down\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

```
\"column\": \"Admit Mistakes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Overthinking\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Sexual Activity\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.0104593452023556,\n
\"min\": 1.0,\n        \"max\": 9.0,\n        \"num_unique_values\":
9,\n        \"samples\": [\n          2.0,\n          4.0\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Concentration\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7975239739638496,\n        \"min\": 1.0,\n        \"max\": 8.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          2.0,\n
6.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Optimisim\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.991297593641536,\n        \"min\":
1.0,\n        \"max\": 9.0,\n        \"num_unique_values\": 9,\n
\"samples\": [\n          8.0,\n          5.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Expert Diagnose\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 120,\n  \"fields\": [\
n    {\n      \"column\": \"Sadness\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.7445412928069721,\n
\"min\": 1.0,\n        \"max\": 3.0,\n        \"num_unique_values\":
3,\n        \"samples\": [\n          2.0,\n          1.0,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Euphoric\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.6078386193019043,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Exhausted\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
0.8217968970834723,\n        \"min\": 1.0,\n        \"max\": 3.0,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1.0,\n
2.0,\n          3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sleep Disorder\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.766234197839261,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Mood Swing\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Suicidal thoughts\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.500979432868119,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1.0,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Anorxia\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Authority Respect\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Try-Explanation\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Aggressive Response\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Ignore & Move-On\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Nervous Break-down\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
```

```
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Admit Mistakes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Overthinking\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Sexual Activity\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.0104593452023556,\n
\"min\": 1.0,\n        \"max\": 9.0,\n        \"num_unique_values\":
9,\n        \"samples\": [\n          2.0,\n          4.0\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Concentration\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7975239739638496,\n        \"min\": 1.0,\n        \"max\": 8.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          2.0,\n
6.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Optimisim\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.991297593641536,\n        \"min\":
1.0,\n        \"max\": 9.0,\n        \"num_unique_values\": 9,\n
\"samples\": [\n          8.0,\n          5.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Expert Diagnose\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f30a6956ed0>
```

```
from matplotlib import pyplot as plt
_df_0['Sadness'].plot(kind='hist', bins=20, title='Sadness')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_1['Euphoric'].plot(kind='hist', bins=20, title='Euphoric')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_2['Exhausted'].plot(kind='hist', bins=20, title='Exhausted')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_3['Sleep Disorder'].plot(kind='hist', bins=20, title='Sleep
Disorder')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f30a3bfb050>
```

```
from matplotlib import pyplot as plt
_df_4.plot(kind='scatter', x='Sadness', y='Euphoric', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_5.plot(kind='scatter', x='Euphoric', y='Exhausted', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_6.plot(kind='scatter', x='Exhausted', y='Sleep Disorder', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_7.plot(kind='scatter', x='Sleep Disorder', y='Mood Swing', s=32,
alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f30a3bfaf10>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Authority Respect']
  ys = series['Sadness']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_8.sort_values('Authority Respect', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Authority Respect')
_ = plt.ylabel('Sadness')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Authority Respect']
  ys = series['Euphoric']
  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_9.sort_values('Authority Respect', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Authority Respect')
_ = plt.ylabel('Euphoric')

from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Authority Respect']
  ys = series['Exhausted']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_10.sort_values('Authority Respect', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Authority Respect')
_ = plt.ylabel('Exhausted')

from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Authority Respect']
  ys = series['Sleep Disorder']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = _df_11.sort_values('Authority Respect', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Authority Respect')
_ = plt.ylabel('Sleep Disorder')

<google.colab._quickchart_helpers.SectionTitle at 0x7f30a407d410>

from matplotlib import pyplot as plt
_df_12['Sadness'].plot(kind='line', figsize=(8, 4), title='Sadness')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_13['Euphoric'].plot(kind='line', figsize=(8, 4), title='Euphoric')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_14['Exhausted'].plot(kind='line', figsize=(8, 4),
title='Exhausted')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_15['Sleep Disorder'].plot(kind='line', figsize=(8, 4),
title='Sleep Disorder')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from sklearn.model_selection import train_test_split
```

```
df.columns
```

```
Index(['Sadness', 'Euphoric', 'Exhausted', 'Sleep Disorder', 'Mood
Swing',
       'Suicidal thoughts', 'Anorxia', 'Authority Respect', 'Try-
Explanation',
       'Aggressive Response', 'Ignore & Move-On', 'Nervous Break-
down',
       'Admit Mistakes', 'Overthinking', 'Sexual Activity',
'Concentration',
       'Optimisim', 'Expert Diagnose'],
      dtype='object')
```

```
X = df.drop(columns=["Expert Diagnose"])
y = df["Expert Diagnose"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```
print(f"Training set: {X_train.shape}")
print(f"Testing set: {X_test.shape}")
```

```
Training set: (96, 17)
Testing set: (24, 17)
```

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)

def convert_categorical(df, cols):
    for col in cols:
        df[col] = df[col].map({"YES": 1, "NO": 0, "Usually": 2, "Most-
Often": 3, "Sometimes": 1})
```

List of columns that need conversion

```
binary_columns = ["Sadness", "Euphoric", "Exhausted", "Sleep
Disorder", "Mood Swing", "Suicidal thoughts",
                  "Anorxia", "Authority Respect", "Try-Explanation",
"Aggressive Response", "Ignore & Move-On",
                  "Nervous Break-down", "Admit Mistakes",
"Overthinking"]

convert_categorical(df, binary_columns)
```

Convert ratings like 'X From 10' into just numbers

```
numeric_columns = ["Sexual Activity", "Concentration", "Optimisim"]
for col in numeric_columns:
    df[col] = df[col].str.extract("(\\d+)").astype(float)
```

Fill missing values with the most common value in each column

```
df.fillna(df.mode().iloc[0], inplace=True)
```

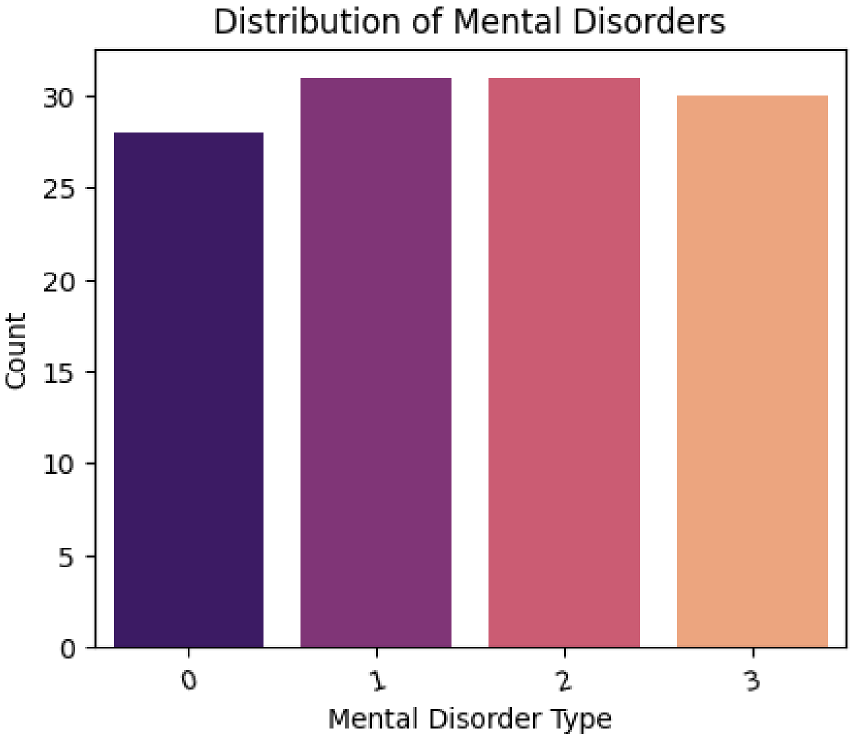
Encode target variable so the model understands it

```
le = LabelEncoder()
df["Expert Diagnose"] = le.fit_transform(df["Expert Diagnose"])
```

Visualizing class distribution (to see how balanced or unbalanced our dataset is)

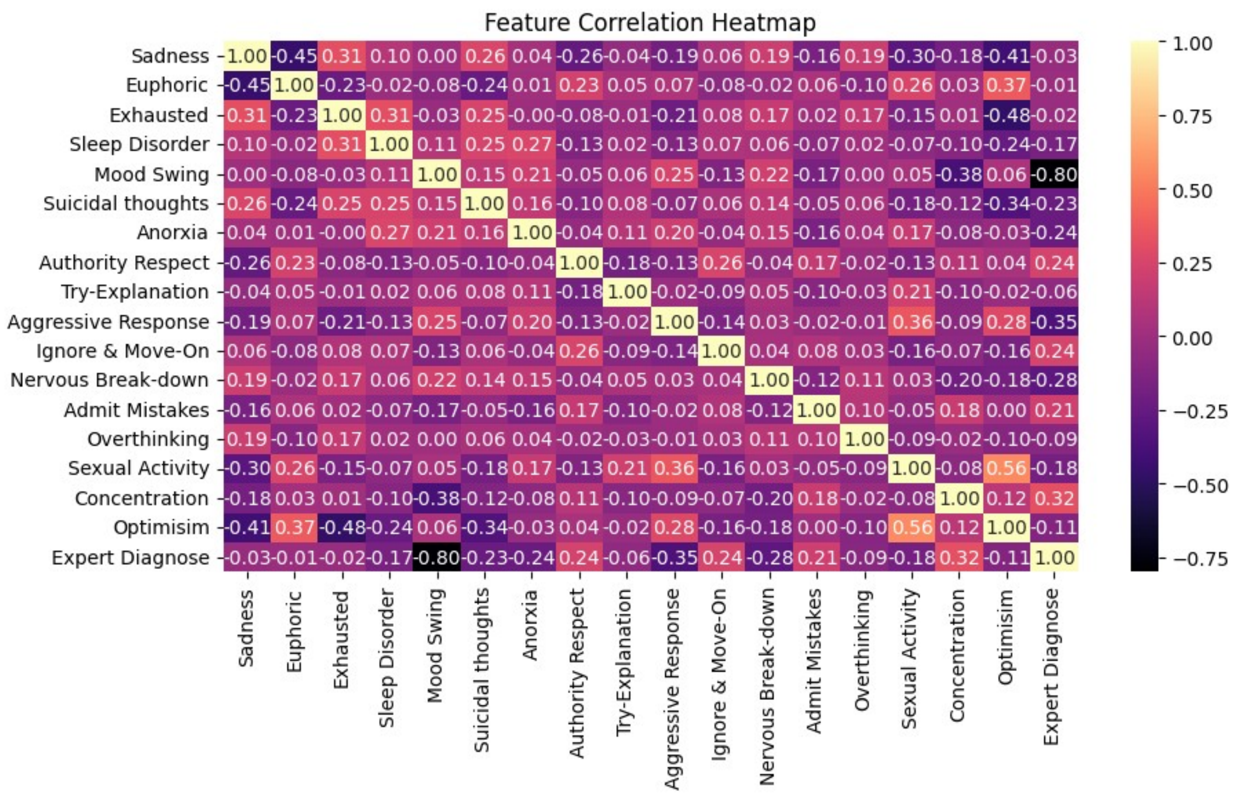
```
plt.figure(figsize=(5, 4))
sns.countplot(x=df["Expert Diagnose"], palette="magma")  # Sunset
tones
plt.xticks(rotation=15)
plt.title("Distribution of Mental Disorders")
plt.xlabel("Mental Disorder Type")
plt.ylabel("Count")
plt.show()
```

```
<ipython-input-16-fae4bd001845>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(x=df["Expert Diagnose"], palette="magma")  # Sunset
tones
```



Heatmap to check feature correlation (to see if some features are too similar)

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(), annot=True, cmap="magma", fmt=".2f")
plt.title("Feature Correlation Heatmap")
plt.show()
```



	Sadness	Euphoric	Exhausted	Sleep Disorder	Mood Swing	Suicidal thoughts	Anorxia	Authority Respect	Try-Explanation	Aggressive Response	Ignore & Move-On	Nervous Break-down	Admit Mistakes	Overthinking	Sexual Activity	Concentration	Optimisim	Expert Diagnose
Sadness	1.00	-0.45	0.31	0.10	0.00	0.26	0.04	-0.26	-0.04	-0.19	0.06	0.19	-0.16	0.19	-0.30	-0.18	-0.41	-0.03
Euphoric	-0.45	1.00	-0.23	-0.02	-0.08	-0.24	0.01	0.23	0.05	0.07	-0.08	-0.02	0.06	-0.10	0.26	0.03	0.37	-0.01
Exhausted	0.31	-0.23	1.00	0.31	-0.03	0.25	-0.00	-0.08	-0.01	-0.21	0.08	0.17	0.02	0.17	-0.15	0.01	-0.48	-0.02
Sleep Disorder	0.10	-0.02	0.31	1.00	0.11	0.25	0.27	-0.13	0.02	-0.13	0.07	0.06	-0.07	0.02	-0.07	-0.10	-0.24	-0.17
Mood Swing	0.00	-0.08	-0.03	0.11	1.00	0.15	0.21	-0.05	0.06	0.25	-0.13	0.22	-0.17	0.00	0.05	-0.38	0.06	-0.80
Suicidal thoughts	0.26	-0.24	0.25	0.25	0.15	1.00	0.16	-0.10	0.08	-0.07	0.06	0.14	-0.05	0.06	-0.18	-0.12	-0.34	-0.23
Anorxia	0.04	0.01	-0.00	0.27	0.21	0.16	1.00	-0.04	0.11	0.20	-0.04	0.15	-0.16	0.04	0.17	-0.08	-0.03	-0.24
Authority Respect	-0.26	0.23	-0.08	-0.13	-0.05	-0.10	-0.04	1.00	-0.18	-0.13	0.26	-0.04	0.17	-0.02	-0.13	0.11	0.04	0.24
Try-Explanation	-0.04	0.05	-0.01	0.02	0.06	0.08	0.11	-0.18	1.00	-0.02	-0.09	0.05	-0.10	-0.03	0.21	-0.10	-0.02	-0.06
Aggressive Response	-0.19	0.07	-0.21	-0.13	0.25	-0.07	0.20	-0.13	-0.02	1.00	-0.14	0.03	-0.02	-0.01	0.36	-0.09	0.28	-0.35
Ignore & Move-On	0.06	-0.08	0.08	0.07	-0.13	0.06	-0.04	0.26	-0.09	-0.14	1.00	0.04	0.08	0.03	-0.16	-0.07	-0.16	0.24
Nervous Break-down	0.19	-0.02	0.17	0.06	0.22	0.14	0.15	-0.04	0.05	0.03	0.04	1.00	-0.12	0.11	0.03	-0.20	-0.18	-0.28
Admit Mistakes	-0.16	0.06	0.02	-0.07	-0.17	-0.05	-0.16	0.17	-0.10	-0.02	0.08	-0.12	1.00	0.10	-0.05	0.18	0.00	0.21
Overthinking	0.19	-0.10	0.17	0.02	0.00	0.06	0.04	-0.02	-0.03	-0.01	0.03	0.11	0.10	1.00	-0.09	-0.02	-0.10	-0.09
Sexual Activity	-0.30	0.26	-0.15	-0.07	0.05	-0.18	0.17	-0.13	0.21	0.36	-0.16	0.03	-0.05	-0.09	1.00	-0.08	0.56	-0.18
Concentration	-0.18	0.03	0.01	-0.10	-0.38	-0.12	-0.08	0.11	-0.10	-0.09	-0.07	-0.20	0.18	-0.02	-0.08	1.00	0.12	0.32
Optimisim	-0.41	0.37	-0.48	-0.24	0.06	-0.34	-0.03	0.04	-0.02	0.28	-0.16	-0.18	0.00	-0.10	0.56	0.12	1.00	-0.11
Expert Diagnose	-0.03	-0.01	-0.02	-0.17	-0.80	-0.23	-0.24	0.24	-0.06	-0.35	0.24	-0.28	0.21	-0.09	-0.18	0.32	-0.11	1.00

Splitting the dataset into training and testing sets (80% train, 20% test)

```
X = df.drop(columns=["Expert Diagnose"])
y = df["Expert Diagnose"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Training the Random Forest model (because why not? It's fast and works well)

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)
```

Making predictions on the test set

```
y_pred = model.predict(X_test)
```

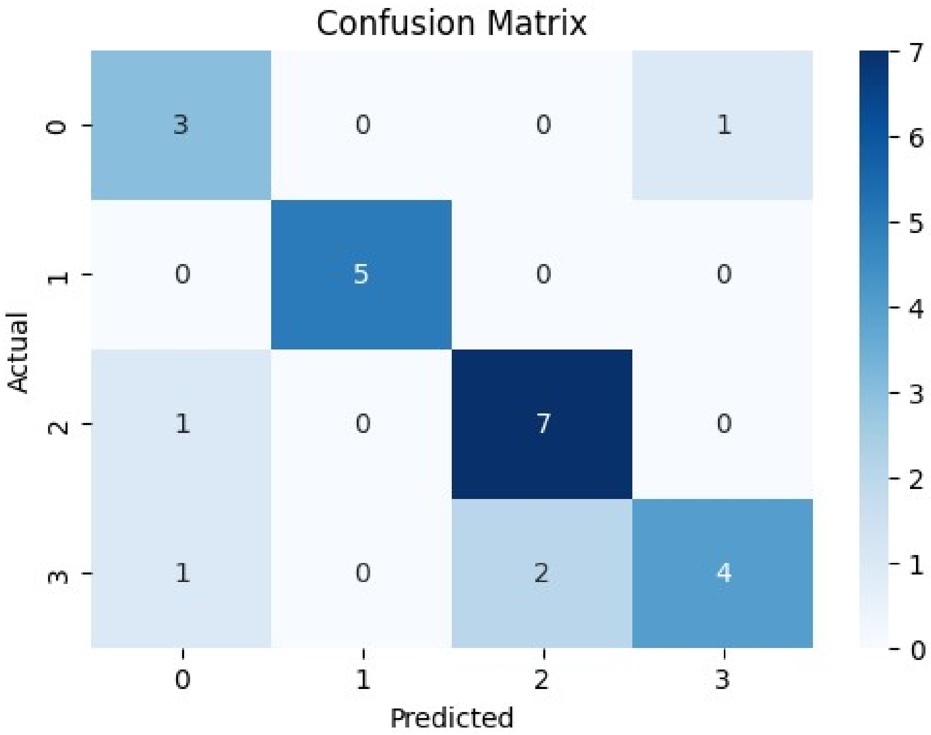
Evaluating how good (or bad) our model is

```python
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test,
y_pred))
```

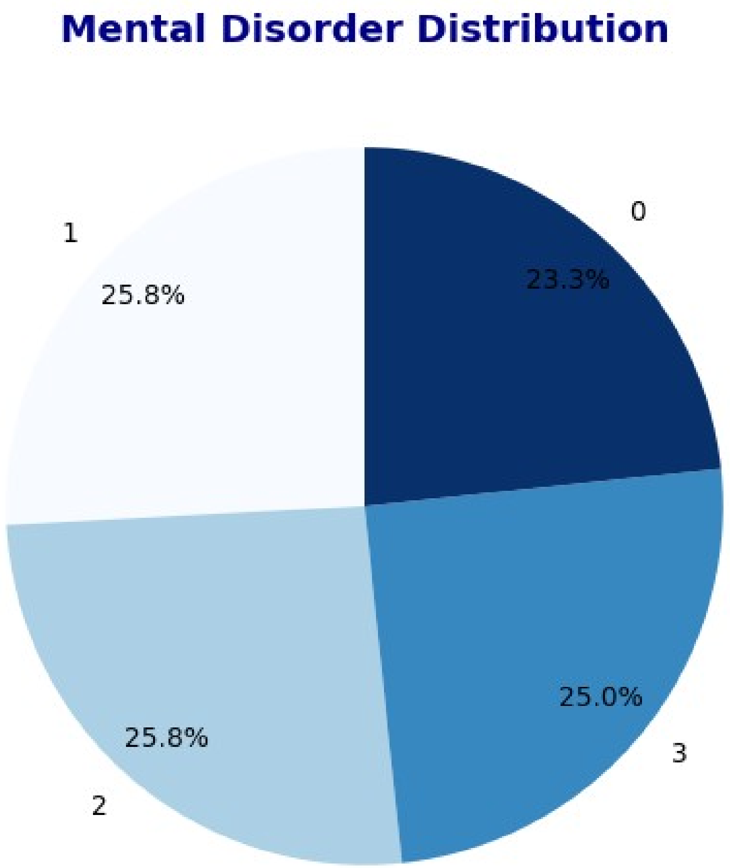
```
Accuracy: 0.7916666666666666
Classification Report:
               precision    recall  f1-score   support

           0       0.60      0.75      0.67         4
           1       1.00      1.00      1.00         5
           2       0.78      0.88      0.82         8
           3       0.80      0.57      0.67         7

    accuracy                           0.79        24
   macro avg       0.79      0.80      0.79        24
weighted avg       0.80      0.79      0.79        24
```

Confusion matrix to see where the model messed up

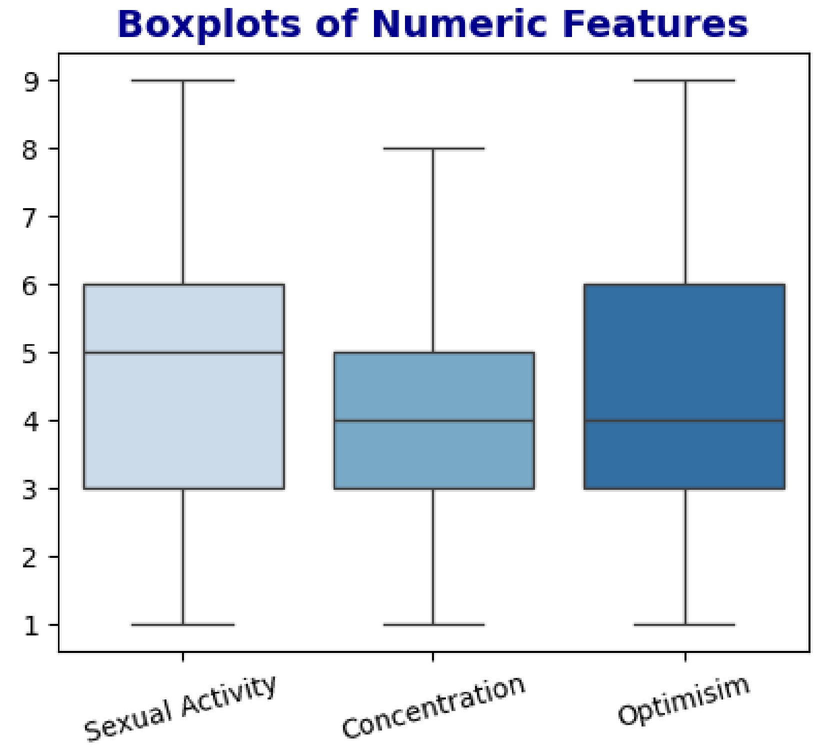
```python
plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d',
cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



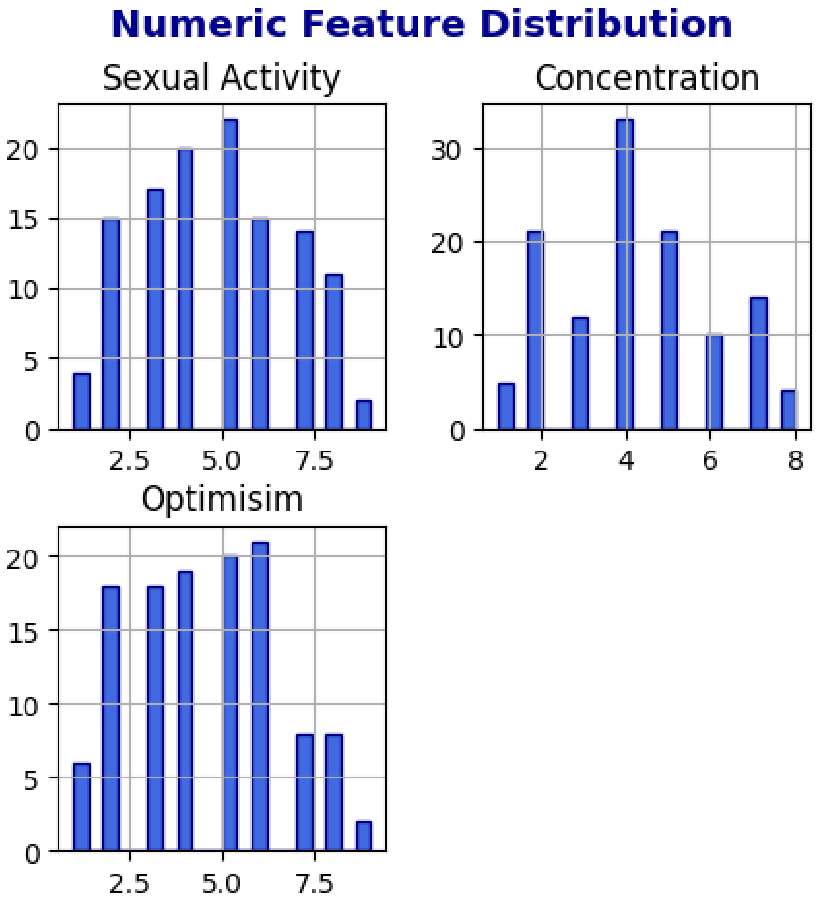
```
plt.figure(figsize=(6, 6))
df["Expert Diagnose"].value_counts().plot.pie(
    autopct="%1.1f%%", cmap="Blues", startangle=90, pctdistance=0.85)
plt.title("Mental Disorder Distribution", fontsize=14,
fontweight="bold", color="darkblue")
plt.ylabel("")
plt.show()
```



```
plt.figure(figsize=(5,4))
sns.boxplot(data=df[numeric_columns], palette="Blues")
plt.title("Boxplots of Numeric Features", fontsize=14,
fontweight="bold", color="darkblue")
plt.xticks(rotation=15)
plt.show()
```



```
df[numeric_columns].hist(figsize=(5,5), bins=20, color="royalblue",
edgecolor="navy")
plt.suptitle("Numeric Feature Distribution", fontsize=14,
fontweight="bold", color="darkblue")
plt.show()
```

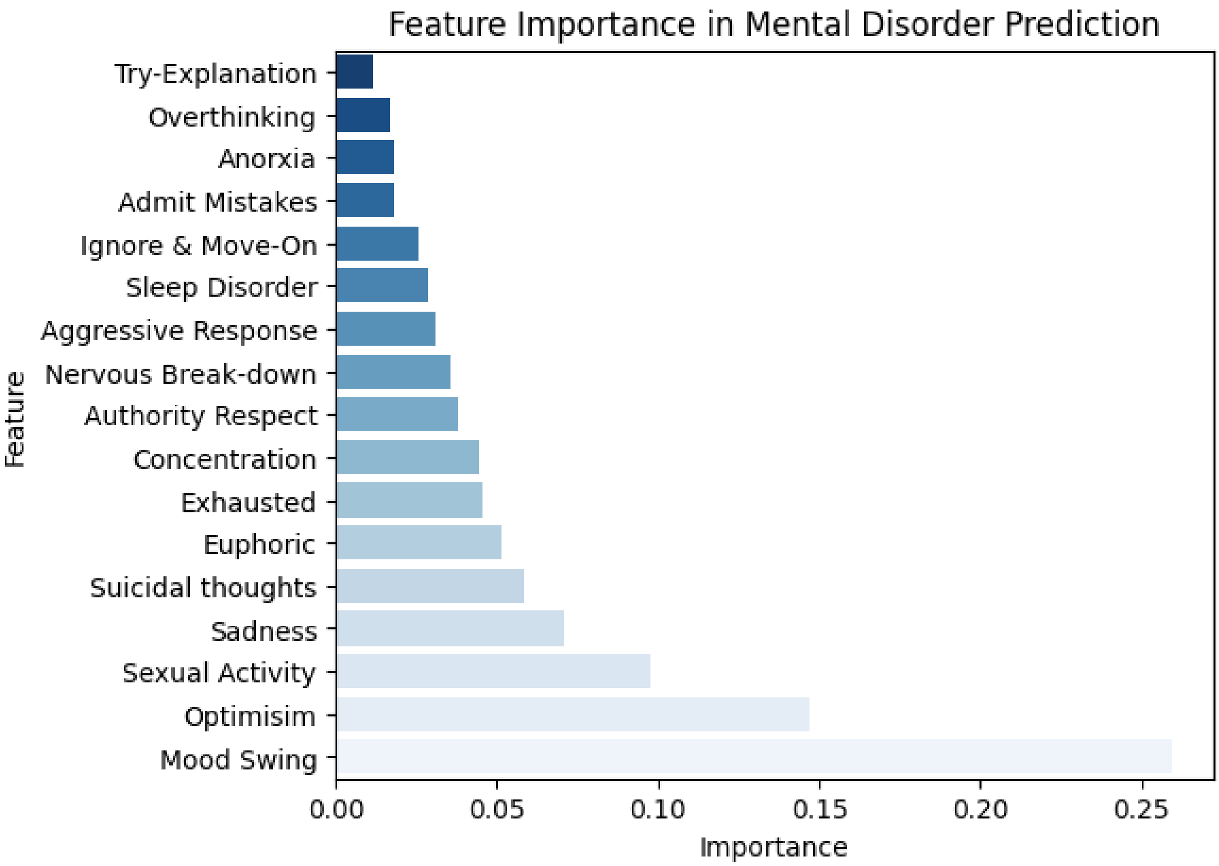


#Feature Importance (Which Symptoms Matter the Most?)

```
feature_importance = pd.DataFrame({"Feature": X.columns, "Importance": 
model.feature_importances_})
feature_importance = feature_importance.sort_values(by="Importance", 
ascending=True)  # Sorted for better view

plt.figure(figsize=(6, 5))
sns.barplot(x=feature_importance["Importance"], 
y=feature_importance["Feature"], palette="Blues_r", orient="h")
plt.title("Feature Importance in Mental Disorder Prediction")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.show()
```

```
<ipython-input-56-e7466d14d0b7>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be 
removed in v0.14.0. Assign the `y` variable to `hue` and set 
`legend=False` for the same effect.
```

```
  sns.barplot(x=feature_importance["Importance"],
y=feature_importance["Feature"], palette="Blues_r", orient="h")
```

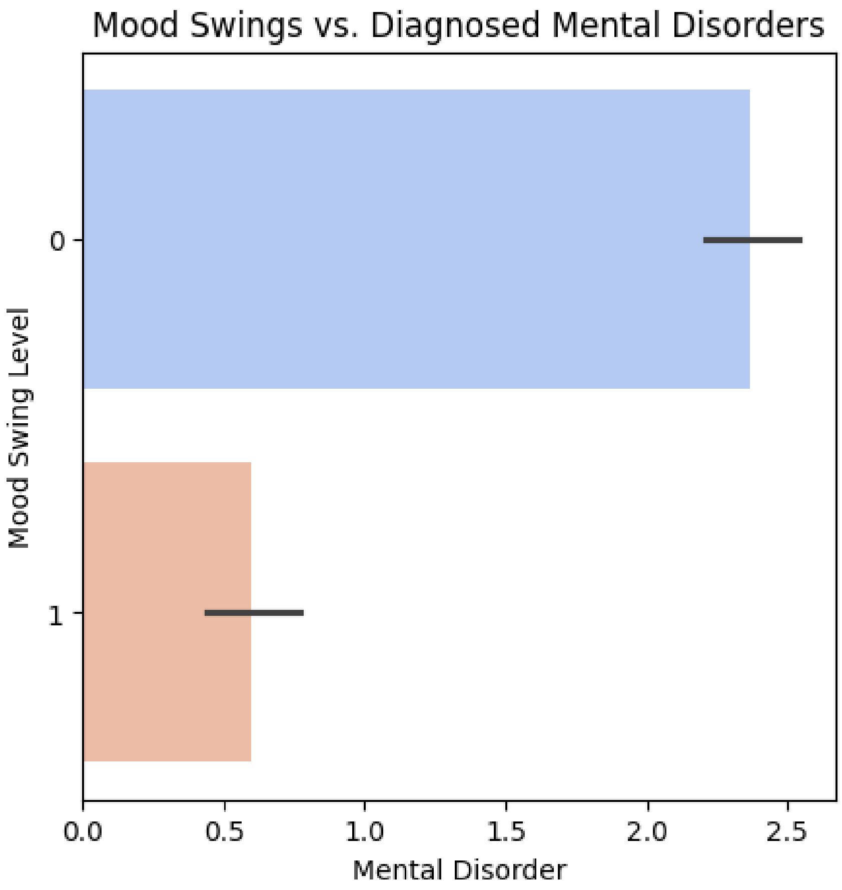


Mood Swings Across Diagnosed Disorders (Horizontal)

```
plt.figure(figsize=(5, 5))
sns.barplot(y=df["Mood Swing"], x=df["Expert Diagnose"],
palette="coolwarm", orient="h")
plt.title("Mood Swings vs. Diagnosed Mental Disorders")
plt.ylabel("Mood Swing Level")
plt.xlabel("Mental Disorder")
plt.show()
```

```
<ipython-input-58-2e9b52ea34ba>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.
```

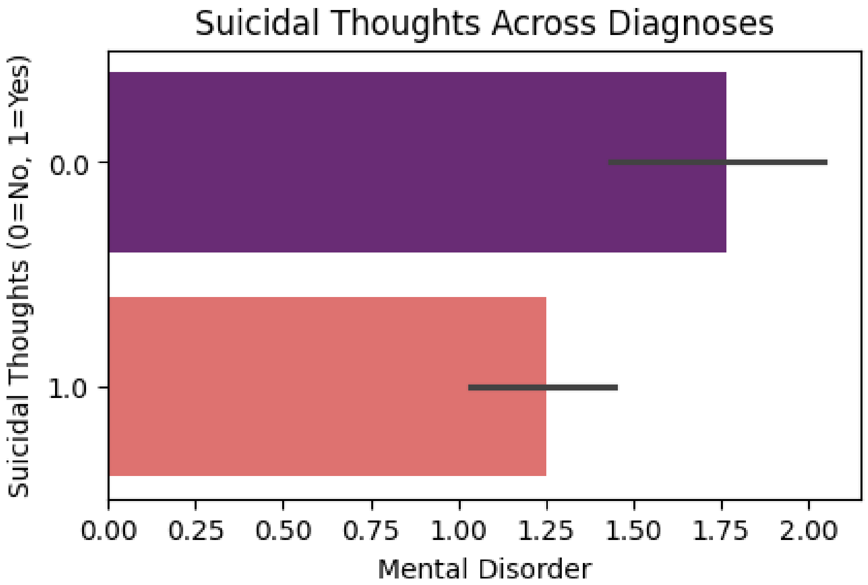
```
  sns.barplot(y=df["Mood Swing"], x=df["Expert Diagnose"],
palette="coolwarm", orient="h")
```



#Suicidal Thoughts Distribution (Horizontal)

```
plt.figure(figsize=(5, 3))
sns.barplot(y=df["Suicidal thoughts"], x=df["Expert Diagnose"],
palette="magma", orient="h")
plt.title("Suicidal Thoughts Across Diagnoses")
plt.ylabel("Suicidal Thoughts (0=No, 1=Yes)")
plt.xlabel("Mental Disorder")
plt.show()
```

```
<ipython-input-61-4ae07375278e>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(y=df["Suicidal thoughts"], x=df["Expert Diagnose"],
palette="magma", orient="h")
```



```
df
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 120,\n  \"fields\": [\
n    {\n      \"column\": \"Sadness\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.7445412928069721,\n
\"min\": 1.0,\n        \"max\": 3.0,\n        \"num_unique_values\":
3,\n        \"samples\": [\n          2.0,\n          1.0,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Euphoric\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.6078386193019043,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Exhausted\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8217968970834723,\n        \"min\": 1.0,\n        \"max\": 3.0,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1.0,\n
2.0,\n          3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Sleep Disorder\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.766234197839261,\n        \"min\":
1.0,\n        \"max\": 3.0,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1.0,\n          3.0,\n          2.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Mood Swing\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"Suicidal thoughts\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.500979432868119,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1.0,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Anorxia\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Authority Respect\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Try-Explanation\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Aggressive Response\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Ignore & Move-On\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Nervous Break-down\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Admit Mistakes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Overthinking\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Sexual Activity\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.0104593452023556,\n
\"min\": 1.0,\n        \"max\": 9.0,\n        \"num_unique_values\":
```

```
9,\n        \"samples\": [\n          2.0,\n          4.0\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Concentration\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7975239739638496,\n        \"min\": 1.0,\n        \"max\": 8.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          2.0,\n
6.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Optimisim\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.991297593641536,\n        \"min\":
1.0,\n        \"max\": 9.0,\n        \"num_unique_values\": 9,\n
\"samples\": [\n          8.0,\n          5.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Expert Diagnose\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```