**Student Mental Health Analysis**

# **Aim:**

To Perform EDA on Student Mental Health Data Set. Use a case study on a data set and apply the various EDA and visualization techniques and present an analysis report.

# **Algorithm:**

Step 1: load the student mental health dataset Start by loading the student mental health Data Set into your preferred data analysis environment,

such as Python or R.

Step 2: Explore the Dataset:

Take a look at the dataset's structure, features, and their data types. Check for missing

values, duplicates, or any other data quality issues.

Step 3: Summarize the Dataset:

Calculate basic statistics of the dataset, such as mean, median, standard deviation,

minimum, maximum, etc. This will provide an initial understanding of the data distribution.

Step 4: Visualize the Target Variable:

Since the dataset includes a quality rating for each wine sample, it's important to

understand its distribution. Create a histogram or bar plot to visualize the distribution of

wine quality ratings.

Step 5: Explore Correlations:

Calculate the correlation matrix to understand the relationships between different features.

Visualize the correlation matrix using a heatmap to identify potential correlations between

features.

Step 6: Analyze Feature Distributions:

Examine the distribution of each feature using histograms, box plots, or violin plots. This

will help identify outliers, skewed distributions, or any patterns within the data.

Step 7: Feature Interactions:

Investigate how different features interact with each other. Use scatter plots or pair plots to

visualize the relationships between pairs of features and observe any trends or patterns.

Step 8: Feature Engineering:

Based on the insights gained from the EDA, consider creating new features or transforming

existing ones to enhance the predictive power of your models.

Step 9: Additional Analysis:

Perform any additional analysis that may be relevant to your specific case study. This could

include comparing distributions across different wine types, analyzing the impact of

specific features on wine quality, or any other exploration based on your domain

knowledge.

Step 10: Draw Conclusions:

Summarize your findings and insights from the EDA. Identify any interesting patterns,

correlations, or outliers that could be further investigated or used in predictive modeling

# **Program:**

# **Imports**

import numpy as np

import pandas as pd

import matplotlib as mpl

import matplotlib.pyplot as plt

import random

import seaborn as sns

import statsmodels.stats as sm

from matplotlib\_venn import venn3

from scipy import stats

# **Preprocessing**

df = pd.read\_csv("Student Mental health.csv")

df.head()

| Timestamp | Choose your gender | Age | What is your course? | Your current year of Study | What is your CGPA? | Marital status | Do you have Depression? | Do you have Anxiety? | Do you have Panic attack? | Did you seek any specialist for a treatment? |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 8/7/2020 12:02 | Female | 18.0 | Engineering | year 1 | 3.00 - 3.49 | No | Yes | No | Yes | No |
| 1 | 8/7/2020 12:04 | Male | 21.0 | Islamic education | year 2 | 3.00 - 3.49 | No | No | Yes | No | No |
| 2 | 8/7/2020 12:05 | Male | 19.0 | BIT | Year 1 | 3.00 - 3.49 | No | Yes | Yes | Yes | No |
| 3 | 8/7/2020 12:06 | Female | 22.0 | Laws | year 3 | 3.00 - 3.49 | Yes | Yes | No | No | No |
| 4 | 8/7/2020 12:13 | Male | 23.0 | Mathemathics | year 4 | 3.00 - 3.49 | No | No | No | No | No |

newnames = ["Timestamp", "Gender", "Age", "Major", "Year", "CGPA", "Married", "Depression", "Anxiety", "Panic Attacks", "Treated"]

df.columns = newnames

df.head()

| Timestamp | Gender | Age | Major | Year | CGPA | Married | Depression | Anxiety | Panic Attacks | Treated |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 8/7/2020 12:02 | Female | 18.0 | Engineering | year 1 | 3.00 - 3.49 | No | Yes | No | Yes | No |
| 1 | 8/7/2020 12:04 | Male | 21.0 | Islamic education | year 2 | 3.00 - 3.49 | No | No | Yes | No | No |
| 2 | 8/7/2020 12:05 | Male | 19.0 | BIT | Year 1 | 3.00 - 3.49 | No | Yes | Yes | Yes | No |
| 3 | 8/7/2020 12:06 | Female | 22.0 | Laws | year 3 | 3.00 - 3.49 | Yes | Yes | No | No | No |
| 4 | 8/7/2020 12:13 | Male | 23.0 | Mathemathics | year 4 | 3.00 - 3.49 | No | No | No | No | No |

len(df.index)

101

df.isna().sum()

Timestamp 0

Gender 0

Age 1

Major 0

Year 0

CGPA 0

Married 0

Depression 0

Anxiety 0

Panic Attacks 0

Treated 0

dtype: int64

def to\_binary(d):

if d == "Yes" : return 1

if d == "No" : return 0

df["Married"] = df["Married"].apply(to\_binary)

df["Depression"] = df["Depression"].apply(to\_binary)

df["Anxiety"] = df["Anxiety"].apply(to\_binary)

df["Panic Attacks"] = df["Panic Attacks"].apply(to\_binary)

df["Treated"] = df["Treated"].apply(to\_binary)

df["Year"] = df["Year"].str[-1:]

df.head()

| Timestamp | Gender | Age | Major | Year | CGPA | Married | Depression | Anxiety | Panic Attacks | Treated |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 8/7/2020 12:02 | Female | 18.0 | Engineering | 1 | 3.00 - 3.49 | 0 | 1 | 0 | 1 | 0 |
| 1 | 8/7/2020 12:04 | Male | 21.0 | Islamic education | 2 | 3.00 - 3.49 | 0 | 0 | 1 | 0 | 0 |
| 2 | 8/7/2020 12:05 | Male | 19.0 | BIT | 1 | 3.00 - 3.49 | 0 | 1 | 1 | 1 | 0 |
| 3 | 8/7/2020 12:06 | Female | 22.0 | Laws | 3 | 3.00 - 3.49 | 1 | 1 | 0 | 0 | 0 |
| 4 | 8/7/2020 12:13 | Male | 23.0 | Mathemathics | 4 | 3.00 - 3.49 | 0 | 0 | 0 | 0 | 0 |

# **Data Visualization**

venn3(subsets = [set(depressed.index),

set(anxious.index),

set(panicking.index)],

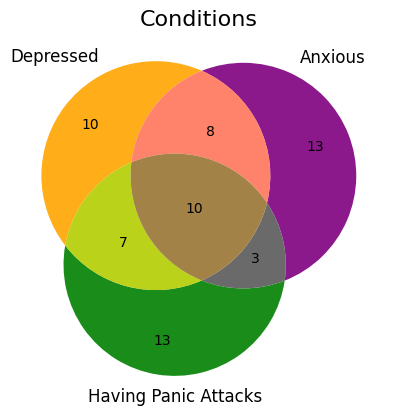
set\_labels = ("Depressed", "Anxious", "Having Panic Attacks"),

set\_colors = ("orange", "purple", "green"),

alpha = 0.9)

plt.title("Conditions", fontsize = 16)

plt.show()



labels = ['Depressed', 'Anxious', 'Having Panic **\n**Attacks',

'Depressed and **\n**Anxious', 'Depressed and Having **\n**Panic Attacks',

'Anxious and Having **\n**Panic Attacks', 'All Three']

treated\_counts = {

"Treated" : [(only\_depressed["Treated"] == 1).sum(),

(only\_anxious["Treated"] == 1).sum(),

(only\_panicking["Treated"] == 1).sum(),

(depressed\_anxious["Treated"] == 1).sum(),

(depressed\_panicking["Treated"] == 1).sum(),

(anxious\_panicking["Treated"] == 1).sum(),

(all\_three["Treated"] == 1).sum()],

"Untreated" : [(only\_depressed["Treated"] == 0).sum(),

(only\_anxious["Treated"] == 0).sum(),

(only\_panicking["Treated"] == 0).sum(),

(depressed\_anxious["Treated"] == 0).sum(),

(depressed\_panicking["Treated"] == 0).sum(),

(anxious\_panicking["Treated"] == 0).sum(),

(all\_three["Treated"] == 0).sum()]

}

fig, ax = plt.subplots(figsize = (10, 3))

bottom = np.zeros(7)

for treatment\_type, treatment\_count **in** treated\_counts.items():

p = ax.bar(labels,

treatment\_count,

width = 0.8,

label = treatment\_type,

bottom = bottom)

bottom += treatment\_count

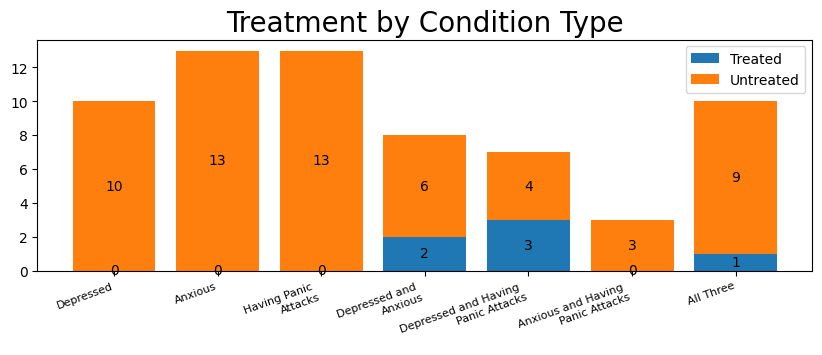
ax.bar\_label(container = p,

label\_type = 'center',

fontsize = 10)

ax.set\_title("Treatment by Condition Type", fontsize = 20)

plt.show()



## **CGPA v. Year v. Condition**

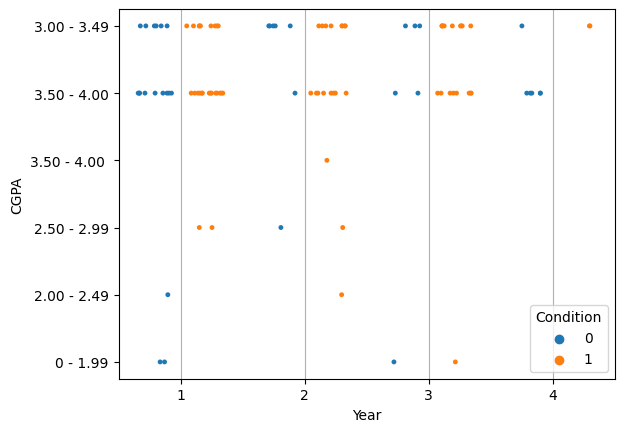
import warnings

warnings.filterwarnings('ignore')

grid = sns.stripplot(data = df, x = "Year", y = "CGPA", hue = "Condition", dodge = True, jitter = 0.3, size = 3.5)

plt.grid(axis = 'x')

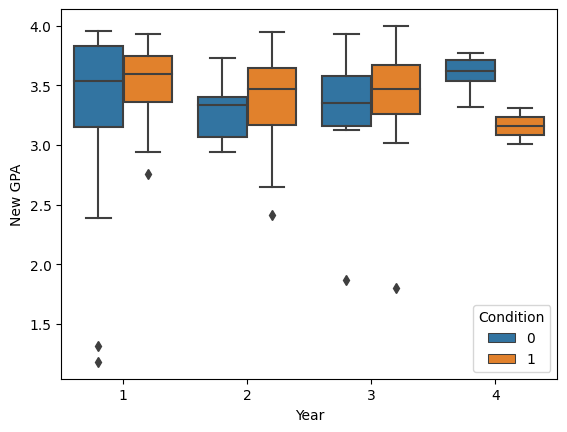
plt.show()



fig, ax = plt.subplots()

sns.boxplot(data = df, x = "Year", y = "New GPA", hue = "Condition")

plt.show()



# **Result:**

Thus the data exploration and visualization to Perform EDA on Student Mental Health Analysis Data Set. Use a case study on a data set and apply the various EDA and visualization techniques and present an analysis report was executed and verified successfully