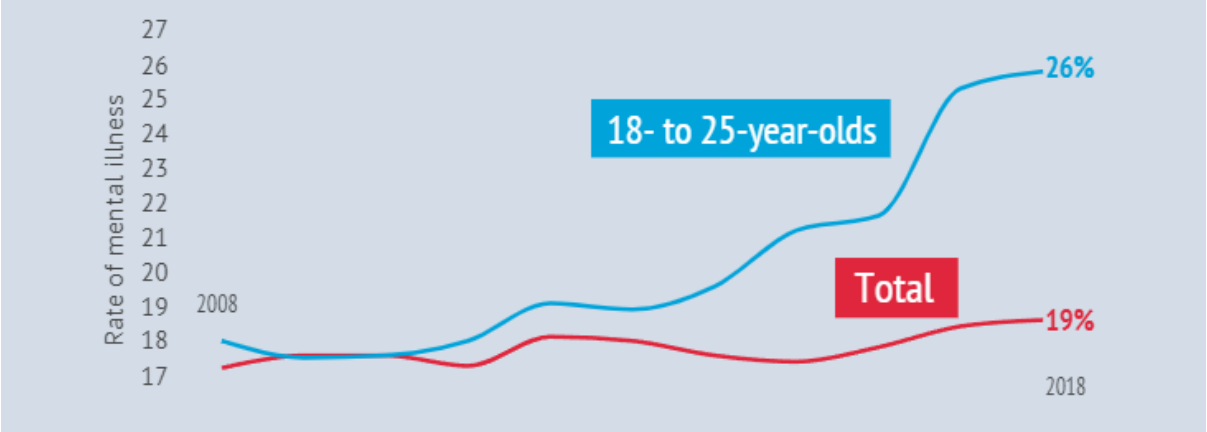
MENTAL HEALTH DETECTION SYSTEM

Submitted by Gunjana dhakad



Increasing mental health issues among students

ABSTRACT:

The increase of mental health problems among students and the need for effective medical health care have led to an investigation of machine learning that can be applied in mental health problems.
This abstract summarises the application of machine learning techniques for the detection of mental health conditions. Mental health disorders are a global public health concern, affecting millions of people worldwide. Early detection and intervention play a crucial role in improving outcomes for individuals with mental health conditions.
Machine learning algorithms have shown great promise in identifying patterns and extracting insights from complex datasets, including those related to mental health. This abstract explores the development of a mental health detection system utilising machine learning models.

PROBLEM STATEMENT:

The problem aims to develop an App or design a system that accurately predicts the mental health of students using machine learning algorithms and details entered by the user.By using data set from other students ,our model should predict correctly.System should be easy to operate and accurate.

MARKET/ CUSTOMER/ BUSINESS ANALYSIS:

- This market-based analysis focuses on the potential market for a mental health detection system utilising machine learning specifically designed for students. The prevalence of mental health issues among students is a growing concern, and early detection and intervention are crucial for their well-being and academic success.

- The target market for this system primarily includes educational institutions such as universities, colleges,high schools and coaching centres. These institutions have a vested interest in the well-being of their students and can benefit from proactive mental health support.

- Market analysis suggests a strong demand for mental health solutions in the education sector. The increasing awareness and stigmatisation of mental health, coupled with the rise in reported cases among students, create a favourable market environment for innovative solutions.

- Competitive analysis reveals a limited number of existing products in the market that specifically cater to the mental health needs of students using machine learning. This presents an opportunity for a novel and specialised solution to gain a competitive advantage.

- This market-based analysis emphasises the potential demand and market opportunities for a mental health detector using machine learning specifically designed for students. By understanding the target market, competitive landscape, economic feasibility, and marketing strategies, the development and implementation of such a system can

address the pressing need for mental health support among students and create a viable commercial product.

TARGET SPECIFICATIONS AND CHARACTERIZATION:

Target specifications include educational institutions such as universities, colleges,high schools and coaching centres.

- System will generate responses with respect to details (age ,gender ,performance,interaction,lifestyle,genetic history) will determine output.Hence large training data with these parameters will be required for accurate predictions.

- User friendly interface to ease the use application and privacy of data by students must be ensured.

- It can be customised to a particular institute as well to include some customised features such as course enrolled in,no of hours engaged in academic or non academic activities etc.For this kind of application various kinds of data set will be required.

- Continuous research and feedback will improve algorithms as well as accuracy of the system.

APPLICABLE PATENTS:

The potential patents that could be apply to mental health detection system:

- Patents related to Machine Learning Algorithms
- Patents related to data privacy collected from students and security
- Patents related to designing of App.

Applicable Regulations:

The application may be subject to several different regulations, depending on the country in which it is being developed and marketed, they are as follows:

- Data privacy of user
- Protection/ownership regulation
- Design responsibility
- Ensuring open source and research community for an audit of algorithms.
- Review of existing work authority regulations

APPLICABLE CONSTRAINTS:

By considering these constraints during the development and implementation phases, the mental health detector can effectively address the unique challenges and considerations of student mental health while ensuring privacy, ethics, accuracy, and user acceptance.

- **Data set:**Requires huge training data and for customised versions it will change according to required features.The availability and quality of labelled data can impact the system's performance. Limited or biassed data can lead to inaccurate predictions and hinder the effectiveness of the mental health detector.

- **Expertise in machine learning:** Developing and training a machine learning model requires specialised expertise in data science and machine learning algorithms.For App development another team of expertise would be required.
- **Ethical Considerations:**Ensuring ethical practices in data collection, usage, and decision-making is paramount. The system should prioritise

informed consent, transparent data handling, and responsible use of the generated insights.

- **Privacy and security of students' data:**Compliance with privacy regulations, such as GDPR or FERPA, is crucial when dealing with sensitive student data. The system must ensure the proper anonymization, encryption, and secure storage of data to protect student privacy.

BUSINESS MODEL:

- **Key Partnership:**Partnership with educational institutes to detect mental well being of students as part of services provided, as well as with mental health experts by giving some discount offers and collecting commission, will generate revenue.

- **Subscription Model:**subscription-based pricing model, with different tiers based on the size of the educational institution or the number of students using the system.

- **Distribution Channel:**Engage in direct sales efforts by reaching out to educational institutions and mental health professionals through targeted marketing campaigns, conferences, and industry events.Establishing strategic partnerships with educational technology companies or mental health service providers to expand the reach and distribution of the mental health detector system.

- **Key Activities:**Continuously improve and refine the mental health detector system using machine learning techniques, incorporating feedback and emerging research .Ensure proper data collection, storage, and analysis while adhering to privacy regulations and maintaining data security.

MODEL DEVELOPMENT:

Classification approaches will be used in the model to predict whether a student seeks medical help or not.Accuracy of training set and test set will be calculated and compared to avoid overfitting or underfitting of data.

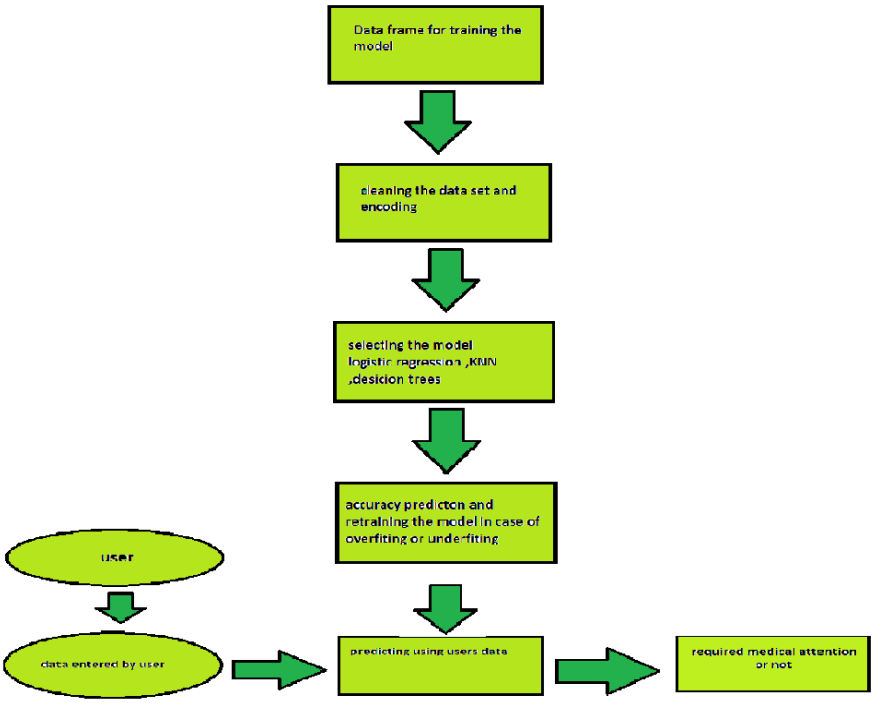
- Details of user:Here details will be entered by user such as age,gender, performance (cgpa,percentile,percentage),lifestyle based data(Exercise or interaction with society) and depression status. Customised data can be added according to customers requirements(here training data should be chosen accordingly).

- Encoding and cleaning the data:Classification data(gender,questions with answer as Yes or No) should be encoded and null values and duplicates will be handled in training data.

- Model building and predictions:After data cleaning and encoding the model will be selected by analysing the data (analysing the relation of features with other features with the help of graphs).In the code logistic regression is used but random forest,decision trees,naive bayes ,KNN algorithms can be used and cumulative
  results can be obtained to ensure higher accuracy of the system.

- Model is ready to predict according to new data of the user.

- The product is designed to be continuously improved based on real-world usage, feedback from users, and emerging research. Regular updates and refinements enhance its performance, reliability, and ability to detect a wide range of mental health conditions.



PRODUCT DETAILS:

- The mental health detector using machine learning for students is a sophisticated software application designed to analyse data and provide insights into students' mental health conditions.

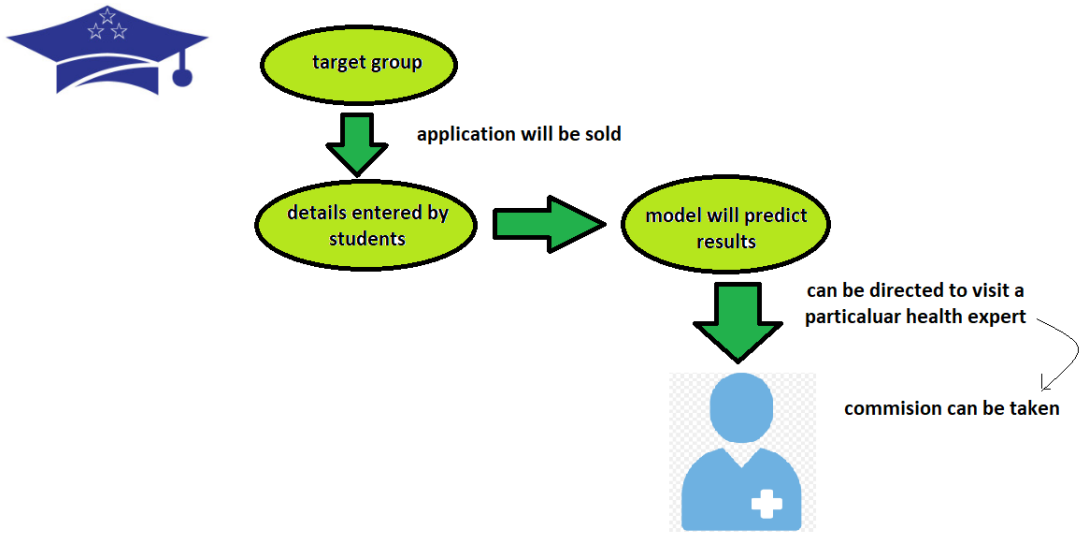
- The product features a user-friendly interface that allows educational institutions, student support services, and mental health professionals to access and interact with the system.Here users will provide details to the application.

- Data sets collected from various resources linked with machine learning algorithms (logistic regression,KNN,decision trees) will provide analysis based on the student data; the system generates predictions about potential mental health conditions or risks.

- The product is designed to be continuously improved based on real-world usage, feedback from users, and emerging research. Regular

updates and refinements enhance its performance, reliability, and ability to detect a wide range of mental health conditions.

- The product is scalable to accommodate various educational institutions, ranging from small colleges to large universities. It can handle diverse student populations and adapt to the specific needs and requirements of different institutions by including specific features.



Team required to develop the model includes:

- Data scientists :These professionals have expertise in machine learning algorithms, data preprocessing, feature extraction, model training, and evaluation. They develop and fine-tune the machine learning models that power the mental health detection system.

- Software developers: This team is responsible for designing and implementing the software infrastructure that supports the mental health detector. They develop the backend systems, APIs, and integration with data sources, as well as ensure the scalability, performance, and security of the application.

- UX/UI Designers: User experience (UX) and user interface (UI) designers are crucial for creating an intuitive and engaging user interface for the mental health detector. They focus on designing

user-friendly interfaces, visualisations, and interactions that enhance the user experience and ease of use.

- Data Privacy and Security Experts: These experts ensure that the system adheres to data privacy regulations and implements robust security measures.

CODE IMPLEMENTATION:

- ## Importing libraries and visualising data set:

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [2]:
```
data_set=pd.read_csv("../input/student-mental-health/Student Mental health.csv")
```

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

Out[3]:

	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

In [4]:
```
data_set.dtypes
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 11 columns):
 #   Column                                        Non-Null Count  Dtype  
---  ------                                        --------------  -----  
 0   Timestamp                                     101 non-null    object 
 1   Choose your gender                            101 non-null    object 
 2   Age                                           100 non-null    float64
 3   What is your course?                          101 non-null    object 
 4   Your current year of Study                    101 non-null    object 
 5   What is your CGPA?                            101 non-null    object 
 6   Marital status                                101 non-null    object 
 7   Do you have Depression?                       101 non-null    object 
 8   Do you have Anxiety?                          101 non-null    object 
 9   Do you have Panic attack?                     101 non-null    object 
 10  Did you seek any specialist for a treatment?  101 non-null    object 
dtypes: float64(1), object(10)
memory usage: 8.8+ KB
```

- **Handling missing values:** Here missing value is replaced by median of that data(age).

In [5]:
```
#taking care of null values and replacing null with median
data_set.isnull().sum()
```

Out[5]:
```
Timestamp                                  0
Choose your gender                         0
Age                                        1
What is your course?                       0
Your current year of Study                 0
What is your CGPA?                         0
Marital status                             0
Do you have Depression?                    0
Do you have Anxiety?                       0
Do you have Panic attack?                  0
Did you seek any specialist for a treatment?    0
dtype: int64
```

In [6]:
```
data_set['Age'].fillna(data_set['Age'].mean(), inplace = True)
```

In [7]:
```
data_set.isnull().sum()
```

Out[7]:
```
Timestamp                                  0
Choose your gender                         0
Age                                        0
What is your course?                       0
Your current year of Study                 0
What is your CGPA?                         0
Marital status                             0
Do you have Depression?                    0
Do you have Anxiety?                       0
Do you have Panic attack?                  0
Did you seek any specialist for a treatment?    0
dtype: int64
```

- Handling Duplicate data:

In [8]:
```
#fixing duplicated data
print("CGPA column values:", data_set['What is your CGPA?'].unique())
print("Year column values:", data_set['Your current year of Study'].unique())
```

```
CGPA column values: ['3.00 - 3.49' '3.50 - 4.00' '3.50 - 4.00 ' '2.50 - 2.99' '2.00 - 2.49'
 '0 - 1.99']
Year column values: ['year 1' 'year 2' 'Year 1' 'year 3' 'year 4' 'Year 2' 'Year 3']
```

In [9]:
```
data_set['What is your CGPA?']=np.where(data_set['What is your CGPA?'] =='3.50 - 4.00 ', '3.50 - 4.00', data_set['What is your CGPA?'])
print(" CGPA column values:", data_set['What is your CGPA?'].unique())
```

```
 CGPA column values: ['3.00 - 3.49' '3.50 - 4.00' '2.50 - 2.99' '2.00 - 2.49' '0 - 1.99']
```

In [10]:
```
data_set['Your current year of Study']=np.where(data_set['Your current year of Study'] =='Year 1', 'year 1', data_set['Your curr
ent year of Study'])
data_set['Your current year of Study']=np.where(data_set['Your current year of Study'] =='Year 2', 'year 2', data_set['Your curr
ent year of Study'])
data_set['Your current year of Study']=np.where(data_set['Your current year of Study'] =='Year 3', 'year 3', data_set['Your curr
ent year of Study'])

print("column values:", data_set['Your current year of Study'].unique())
```

```
column values: ['year 1' 'year 2' 'year 3' 'year 4']
```

In [11]:
```
train_data= data_set.copy()
```

- **Encoding Data:**Classification data types(gender[male,female],family history of mental illness[yes,no] etc) are encoded to 0 and 1 while cgpa and year is encoded using pd.dummies.Less important columns are removed (married status etc) from data set.

In [12]:
```
#transforming data to the required needs
train_data.rename(columns = {'Choose your gender': 'gender',
                    'Do you have Depression?': 'depression_status',
                    'Do you have Panic attack?': 'Family_history_of_mental_disorders',
                    'Do you have Anxiety?': 'Exercise',
                    'Timestamp': 'timestamp',
                    'What is your CGPA?': 'cgpa',
                    'Age': 'age',
                    'What is your course?': 'course',
                    'Your current year of Study': 'year',
                    'Marital status': 'marital_status',
                    'Did you seek any specialist for a treatment?': 'Medical_attention_required'}, inplace = True)
train_data.head()
```

Out[12]:

	timestamp	gender	age	course	year	cgpa	marital_status	depression_status	Exercise	Family_history_of_mental_disorders	Medical_attention_required
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

In [13]:
```
#dropping unused columns and encoding classification data
training=pd.get_dummies(train_data, columns=["cgpa","year"])
training.Exercise.replace(('Yes', 'No'), (0, 1), inplace=True)
training.Family_history_of_mental_disorders.replace(('Yes', 'No'), (1,0), inplace=True)
training.Medical_attention_required.replace(('Yes', 'No'), (1, 0), inplace=True)
training.gender.replace(('Female', 'Male'), (1, 0), inplace=True)
training.depression_status.replace(('Yes', 'No'), (1, 0), inplace=True)
training.drop('timestamp', axis=1, inplace=True)
training.drop('course', axis=1, inplace=True)
training.drop('marital_status', axis=1, inplace=True)
training.head()
```

Out[13]:

	gender	age	depression_status	Exercise	Family_history_of_mental_disorders	Medical_attention_required	cgpa_0 - 1.99	cgpa_2.00 - 2.49	cgpa_2.50 - 2.99	cgpa_3.00 - 3.49	cgpa_3.50 - 4.00	ye 1
0	1	18.0	1	1	1	0	0	0	0	1	0	1
1	0	21.0	0	0	0	0	0	0	0	1	0	0
2	0	19.0	1	0	1	0	0	0	0	1	0	1
3	1	22.0	1	1	0	0	0	0	0	1	0	0
4	0	23.0	0	1	0	0	0	0	0	1	0	0

In [14]:
```
training.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 15 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   gender                              101 non-null    int64
 1   age                                 101 non-null    float64
 2   depression_status                   101 non-null    int64
 3   Exercise                            101 non-null    int64
 4   Family_history_of_mental_disorders  101 non-null    int64
 5   Medical_attention_required          101 non-null    int64
 6   cgpa_0 - 1.99                       101 non-null    uint8
 7   cgpa_2.00 - 2.49                    101 non-null    uint8
 8   cgpa_2.50 - 2.99                    101 non-null    uint8
 9   cgpa_3.00 - 3.49                    101 non-null    uint8
 10  cgpa_3.50 - 4.00                    101 non-null    uint8
 11  year_year 1                         101 non-null    uint8
 12  year_year 2                         101 non-null    uint8
 13  year_year 3                         101 non-null    uint8
 14  year_year 4                         101 non-null    uint8
```

- Splitting of data set:data set is splitted into training and test data with ratio 0.2 by using sklearn.

In [15]:
```
#spliting data set into training and test set
from sklearn.model_selection import train_test_split
train_set,test_set=train_test_split(training,test_size=0.2,random_state=42)
print(f"Row in train set={len(train_set)}\nRow in test set={len(test_set)}")
train_set.head()
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required
for this version of SciPy (detected version 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"

Row in train set=80
Row in test set=21
```

Out[15]:

	gender	age	depression_status	Exercise	Family_history_of_mental_disorders	Medical_attention_required	cgpa_0 - 1.99	cgpa_2.00 - 2.49	cgpa_2.50 - 2.99	cgpa_3.00 - 3.49	cgpa_3.50 - 4.00
89	0	24.0	0	1	1	0	0	1	0	0	0
26	1	19.0	0	1	0	0	0	0	0	0	1
42	1	20.0	1	1	0	0	0	0	0	1	0
70	1	24.0	0	1	0	0	0	0	0	1	0
15	0	24.0	0	1	0	0	0	0	0	0	1

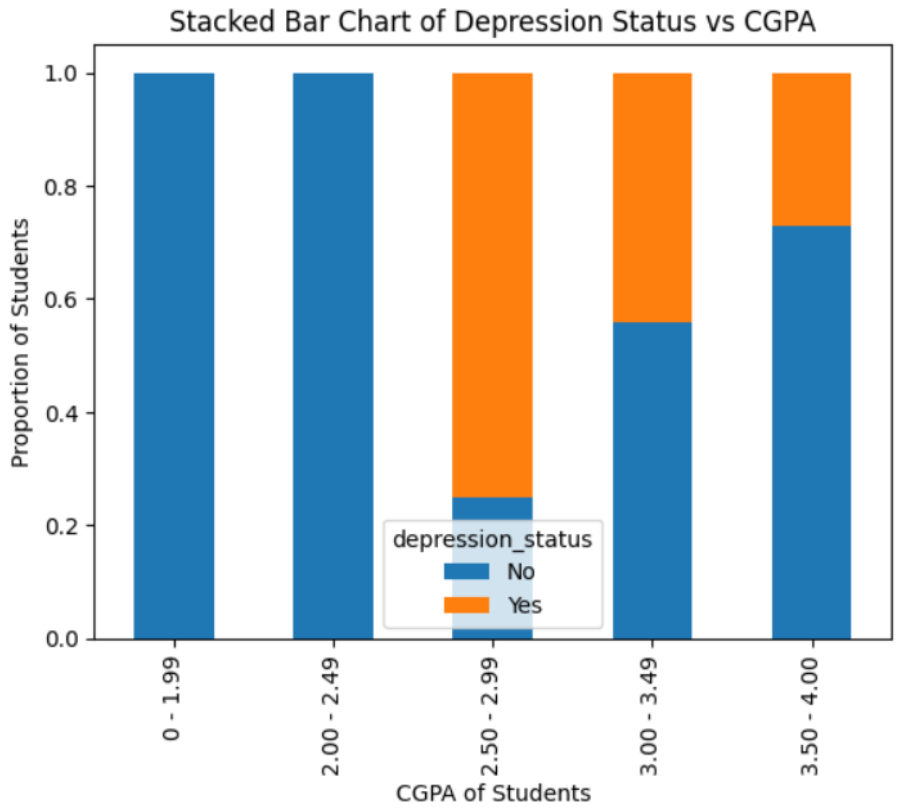
- Data analysis:

In [16]:

```
#analysing given data set
import matplotlib.pyplot as plt
table=pd.crosstab(train_data['cgpa'],train_data['depression_status'])
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Depression Status vs CGPA')
plt.xlabel('CGPA of Students')
plt.ylabel('Proportion of Students')
```

Out[16]:

```
Text(0, 0.5, 'Proportion of Students')
```

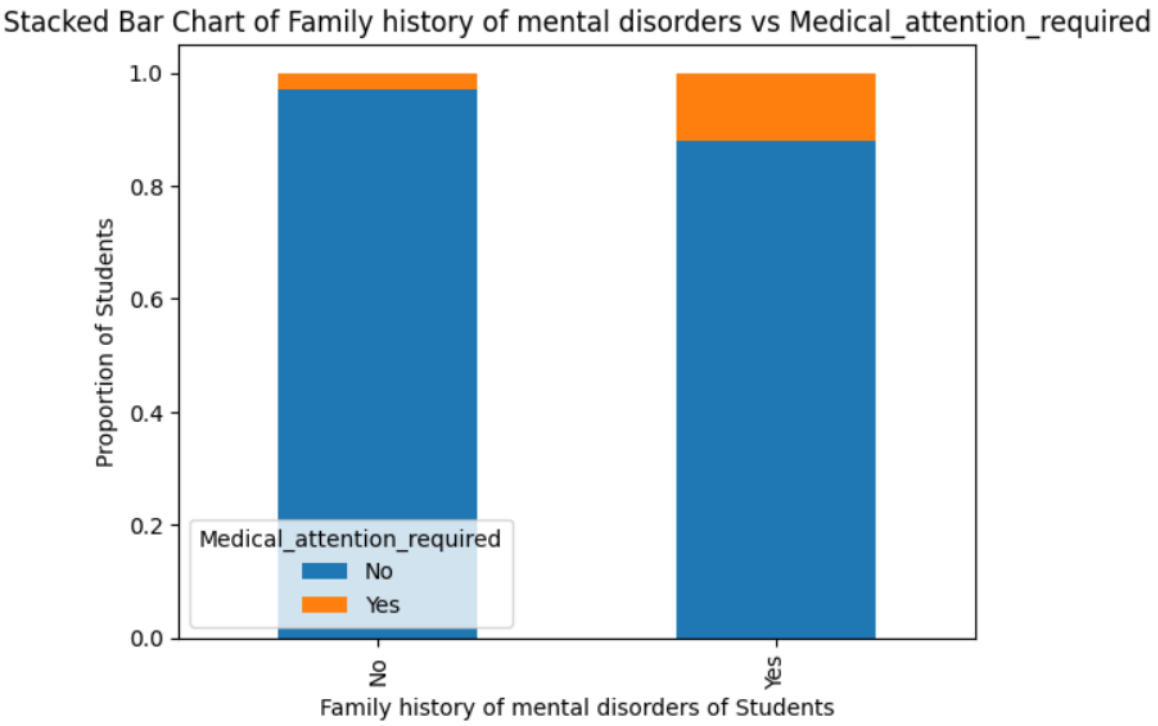


In [17]:

```
table=pd.crosstab(train_data['Family_history_of_mental_disorders'],train_data['Medical_attention_required'])
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Family history of mental disorders vs Medical_attention_required')
plt.xlabel('Family history of mental disorders of Students')
plt.ylabel('Proportion of Students')
```

Out[17]:

```
Text(0, 0.5, 'Proportion of Students')
```

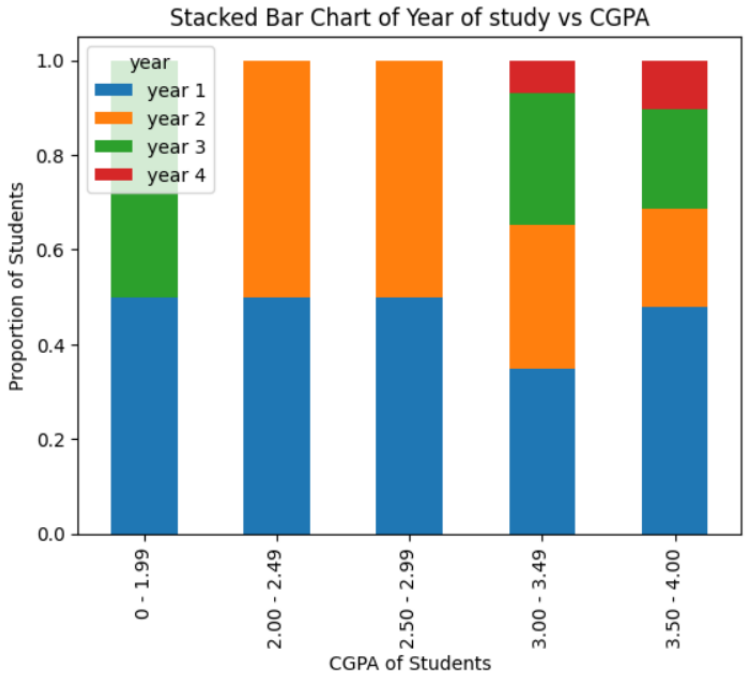


In [18]:

```
table=pd.crosstab(train_data['cgpa'],train_data['year'])
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Year of study vs CGPA')
plt.xlabel('CGPA of Students')
plt.ylabel('Proportion of Students')
```

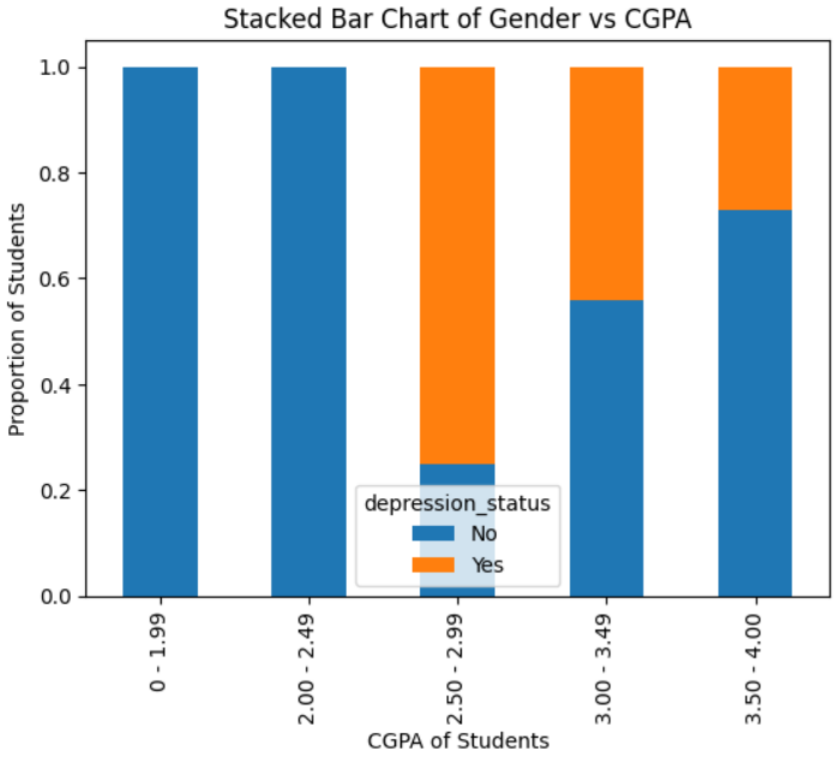
Out[18]:

```
Text(0, 0.5, 'Proportion of Students')
```



In [19]:
```
table=pd.crosstab(train_data['cgpa'],train_data['depression_status'])
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Gender vs CGPA')
plt.xlabel('CGPA of Students')
plt.ylabel('Proportion of Students')
```

Out[19]:
```
Text(0, 0.5, 'Proportion of Students')
```



- **Training the model**:Logistic regression model is used while algorithms such as KNN naive bayes,Decision trees and random forest can be used for more complicated data sets.

In [20]:
```
#creatin data sets for predications
train_setX=train_set.drop("Medical_attention_required",axis=1)
train_sety=train_set["Medical_attention_required"].copy()
test_setX=test_set.drop("Medical_attention_required",axis=1)
test_sety=test_set["Medical_attention_required"].copy()
```

In [21]:
```
#choosing the model
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [22]:
```
#training the model overtrain set
model.fit(train_setX,train_sety)
```

Out[22]:
```
LogisticRegression
LogisticRegression()
```

- Accuracy measurement: accuracy on both training data and test data is compared.accuracy of 90% is obtained over test set which will be slightly lower if the large data set was used.

In [23]:
```
#predicting accuracy over training data
from sklearn.metrics import accuracy_score
pred=model.predict(train_setX)
accuracy=accuracy_score(pred,train_sety)
print(accuracy)
```

```
0.95
```

In [24]:
```
#predicting accuracy over testing data
pred=model.predict(test_setX)
accuracy=accuracy_score(pred,test_sety)
print(accuracy)
```

```
0.9047619047619048
```

Github link: https://github.com/Gunjanadhakad/mental-health-detector/

Data set:

- https://www.kaggle.com/datasets/shariful07/student-mental-health?rvi=1 (link for data set)
- Here some of the features are changed to obtain desired features.New features obtained by replacing old ones are:Family history of mental illness,Exercise,Need medical help.
- Instead of this data set,if a big data set was used then the model will be more accurate.

EXTERNAL RESEARCHES:

- Information about increasing number of cases of depression among youth and role of machine learning:

https://www.hindawi.com/journals/acisc/2022/9970363/
https://www.analyticsvidhya.com/blog/2022/06/mental-health-prediction-using-machine-learning/
https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012021/pdf

- The code is implemented in kaggle notebook by and linked to github,Data set is taken from kaggle.

https://www.kaggle.com/

CONCLUSION:

In conclusion, a mental health detector using machine learning for students holds significant potential in addressing the pressing need for early detection and intervention of mental health conditions among students. By leveraging advanced algorithms and data-driven insights, such a system can provide personalised support, improve mental health outcomes, and enhance the overall well-being and academic success of students.