**Project Title: Predicting Mental Health Risk in Students**

**Project Aim:**

The aim of this project is to develop a predictive model that assesses the risk of mental health issues among students based on various demographic and academic-related factors. Mental health problems, such as depression, anxiety, and panic attacks, are significant concerns in the academic community, affecting students' well-being and academic performance. By leveraging deep learning techniques and a dataset collected from student surveys, this project seeks to provide a tool for early detection and intervention in potential cases of mental health issues.

**Procedure:**

1. Data Preprocessing:
2. Encode categorical variables and standardize numerical features.
3. Data Splitting: Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).
4. Model Creation: Develop a deep learning model for binary classification.
5. Model Compilation: Specify optimizer, loss function, and evaluation metrics (e.g., accuracy).
6. Model Training: Train the model using the training data with batch processing and epochs.
7. Model Evaluation: Assess the model's performance with metrics like accuracy, precision, recall, and F1-score.
8. Risk Prediction: Use the trained model to predict the risk of mental health issues for each student in the testing dataset.
9. Results Interpretation:
10. Results Presentation: Present results, including percentage-based risk assessments and actionable recommendations.

**Data Collection:**

A STATISTICAL RESEARCH ON THE EFFECTS OF MENTAL HEALTH ON STUDENTS’ CGPA dataset

This Data set was collected by a survey conducted by Google forms from University student in order to examine their current academic situation and mental health.

Dataset available here : <https://www.kaggle.com/datasets/shariful07/student-mental-health>

**Model Development:**

The model is at the core of our project's predictive capabilities. It's a neural network built to extract patterns from student data and forecast the risk of mental health issues. This architecture includes an input layer, hidden layers with ReLU activation for learning complex relationships, and an output layer with sigmoid activation for binary classification. We implement this architecture using TensorFlow/Keras, allowing for a sequential layer-wise construction, resulting in a powerful tool for risk assessment.

**Model Training:**

In this critical phase, the model is exposed to the training data, where it learns to recognize patterns and associations between features and mental health risk. Training involves multiple iterations (epochs) with batch processing to update the model's weights. The outcome is a model fine-tuned to make accurate risk predictions, setting the stage for its evaluation and real-world application.

**Model Evaluation:**

This phase assesses the model's effectiveness in predicting mental health risk. Key metrics, including accuracy, precision, recall, and F1-score, are calculated using the testing data. These metrics provide insights into the model's performance and its ability to make accurate classifications, ensuring its suitability for practical use and actionable results.

**Prediction on New Data:**

Once the model is trained and validated, it's ready for real-world application. Using this trained model, we can predict the risk of mental health issues in new data, such as incoming student surveys. The model's predictions are based on the patterns it has learned from the training data, providing valuable insights for early intervention and support.

**Tools and Libraries Used:**

1. **Python:** The primary programming language for implementation.
2. **TensorFlow or PyTorch:** Deep learning frameworks for building and training segmentation models.
3. **scikit-learn:** A versatile machine learning library that offers tools for data preprocessing, model evaluation, and splitting data into training and testing sets.
4. **StandardScaler:** Part of scikit-learn, this tool standardizes numerical features to maintain uniformity in data scaling during model training.
5. **NumPy:** For numerical operations and array manipulation.
6. **Matplotlib**:Used for creating visualizations and result analysis**.**

**Program:**

Importing

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from collections import Counter

from imblearn.over\_sampling import RandomOverSampler

from sklearn.metrics import classification\_report,confusion\_matrix,precision\_score,recall\_score,accuracy\_score,f1\_score

from sklearn import preprocessing

import pandas as pd

import numpy as np

import tensorflow as tf

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report

from pydrive.auth import GoogleAuth

from pydrive.drive import GoogleDrive

from google.colab import drive

drive.mount('/content/drive/')

Data Processing and Analysis

df = pd.read\_csv('/content/drive/MyDrive/StudentMentalHealth.csv') # Reading data from the Student Mental health.csv file

df.head(2)

df.isnull().sum() # age has a null

df['Age'] = df['Age'].fillna(df['Age'].mean())

df['Age'].isnull().sum()

df.columns = ['Date\_Time', 'Gender', 'Age', 'Course', 'Year', 'CGPA', 'Marital\_Status', 'Depression', 'Anxiety', 'Panic\_Attack', 'Treatment']

df.columns

df.head(1) # Renamed columns

df['Year'].unique() # year 1 and Year 1 are the same.We will only keep the year

df['Year'] = df['Year'].apply(lambda x : int(x.split(' ')[-1]))

df['Year'].unique()

df['CGPA'] = df['CGPA'].apply(lambda x : x.strip())

df['CGPA'].unique()

# To see the no of courses students are enrolled in

Course\_List = df['Course'].unique().tolist()

print(len(Course\_List))

df['Course'].unique() # There are multiple courses with the same name.

course\_dic = {'engin': 'Engineering' , 'Engine':'Engineering' , 'Islamic education':'Islamic Education' ,

              'Pendidikan islam':'Pendidikan Islam' , 'BIT':'IT', 'psychology':'Psychology', 'koe': 'Koe',

              'Kirkhs': 'Irkhs', 'KIRKHS': 'Irkhs', 'Benl': 'BENL', 'Fiqh fatwa ': 'Fiqh', 'Laws': 'Law'}

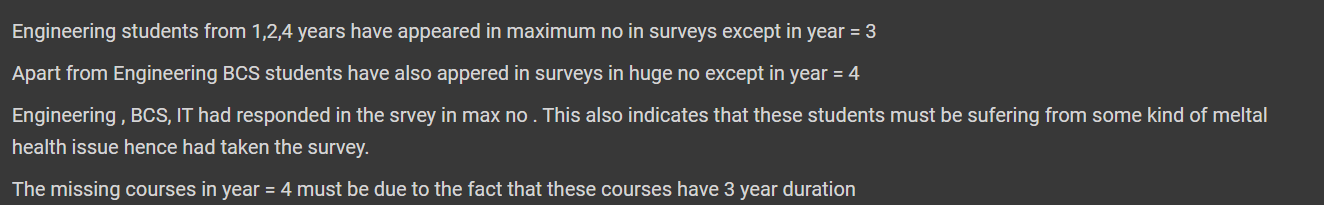
df['Course'].replace(course\_dic,inplace=True)

df["Age"].value\_counts() # 20.53 age has 1 record which is to be removed

df.iloc[row\_idx] # This row have 20.53 age

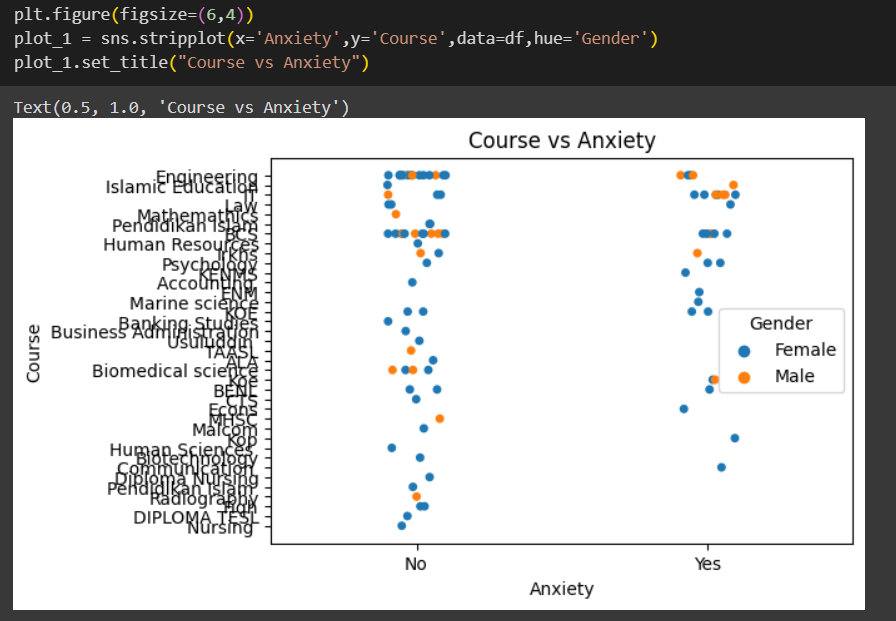
df\_1 = df.drop(row\_idx)

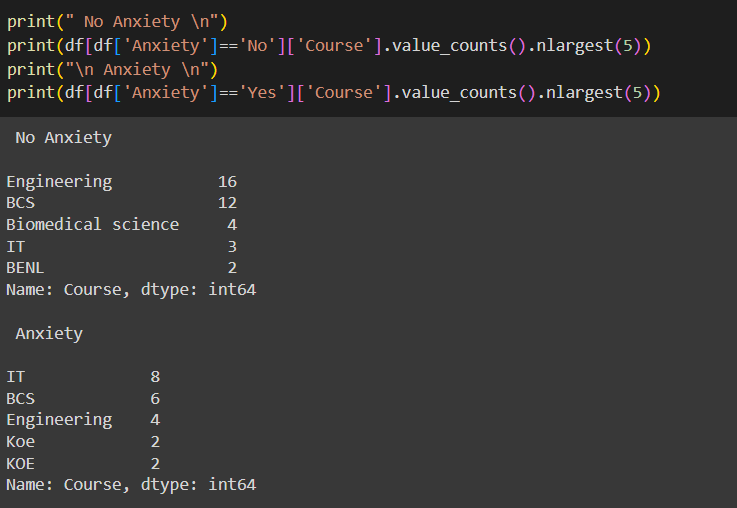
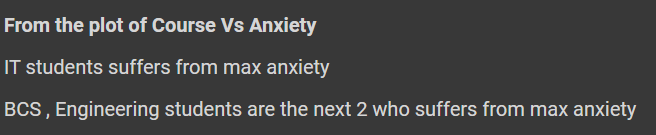
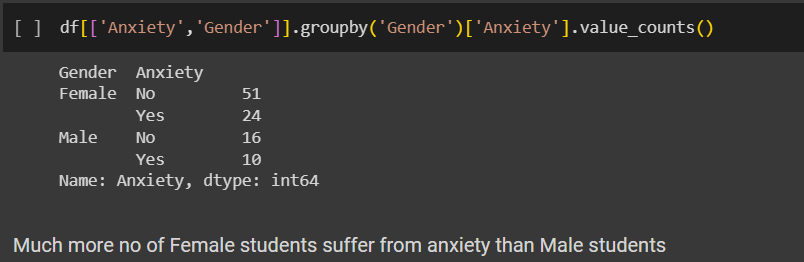
print(df\_1.shape)  # Removed the fractional age



Anxiety

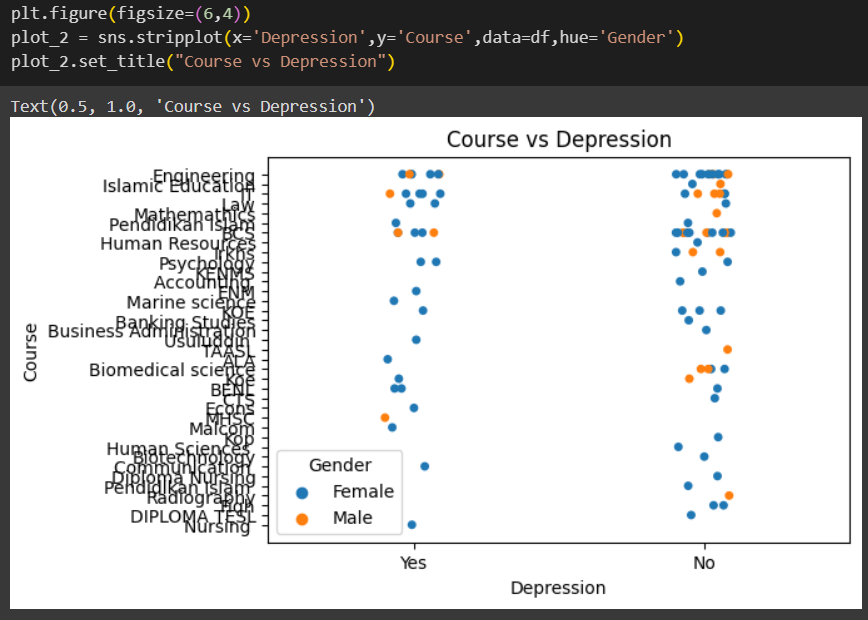
df['Anxiety'].value\_counts() # Out of 101 students 34 are suffering from anxiety



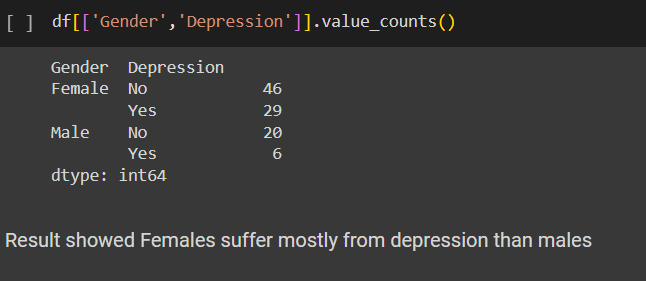
**  **

Depression

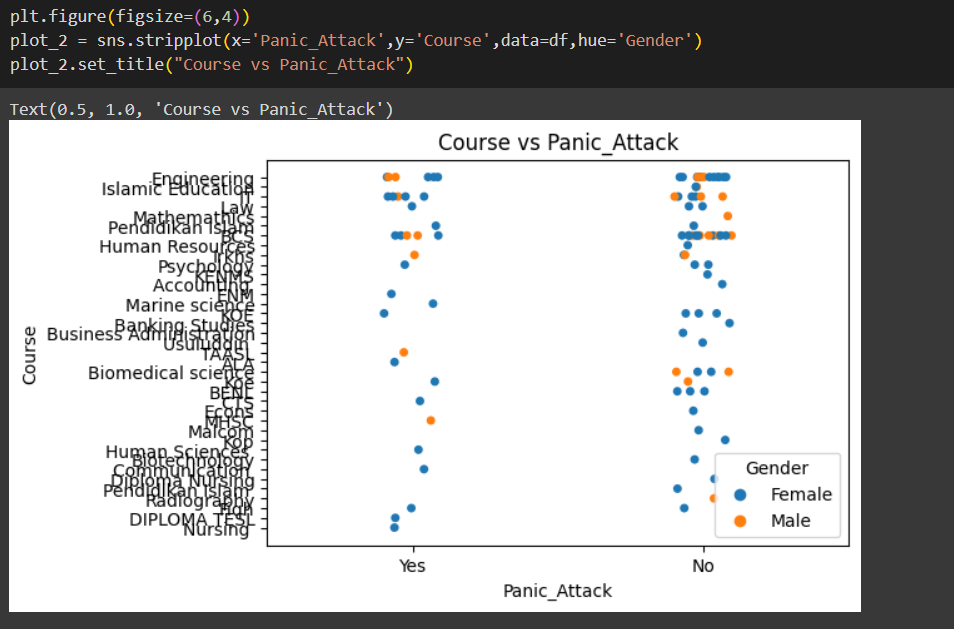
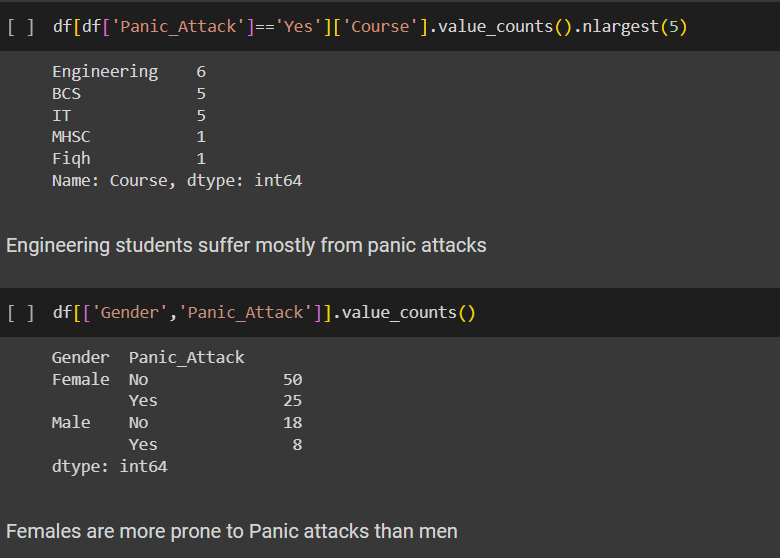
df['Depression'].value\_counts()



print(df[df['Depression']=='Yes']['Course'].value\_counts().nlargest(5))

Panick Attacks

** **

Mental Health in general

def problem(dataframe):

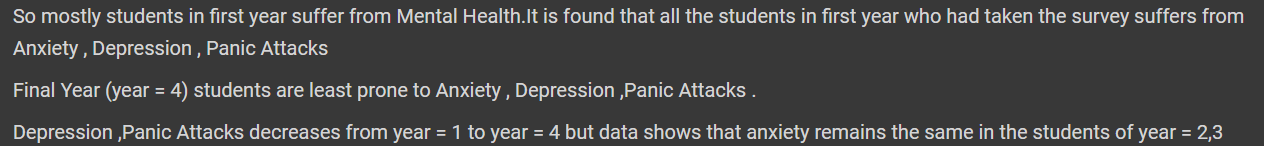
    list = ['Depression','Anxiety','Panic\_Attack']

    for i in list:

        print(df[['Year',i]].groupby('Year')[i].value\_counts())

        print('\n')

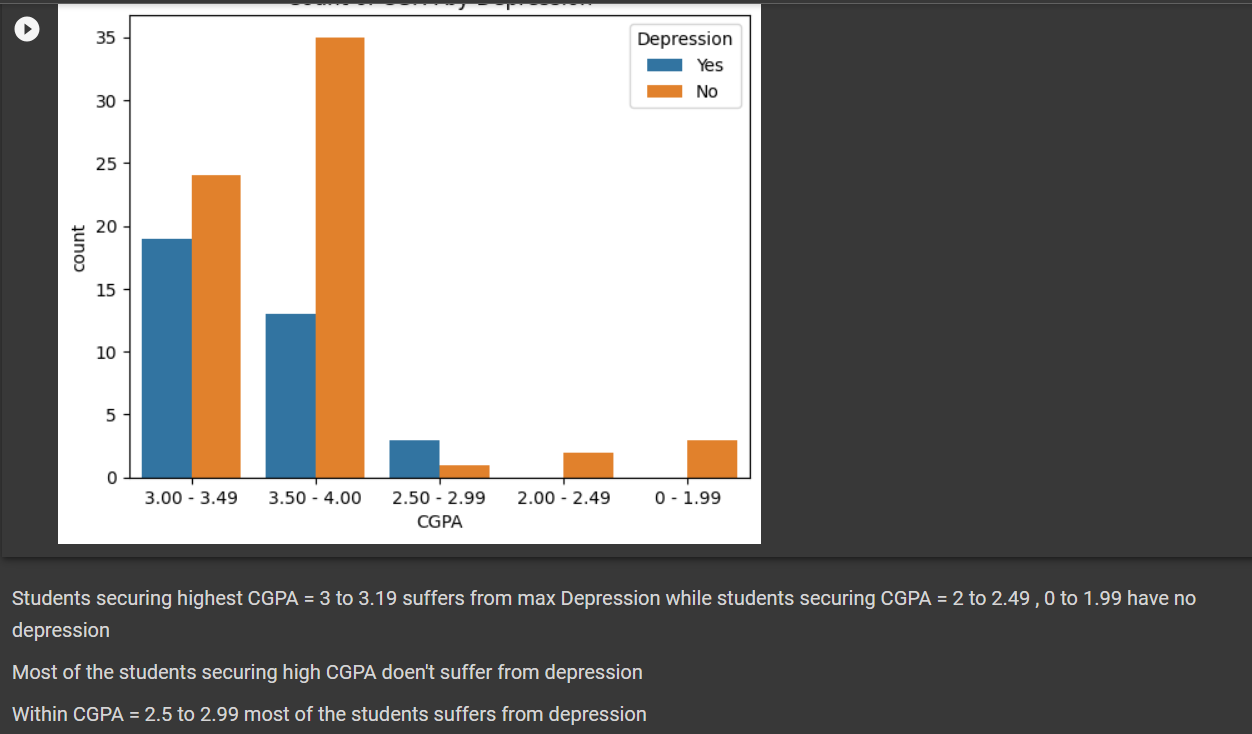
problem(df)



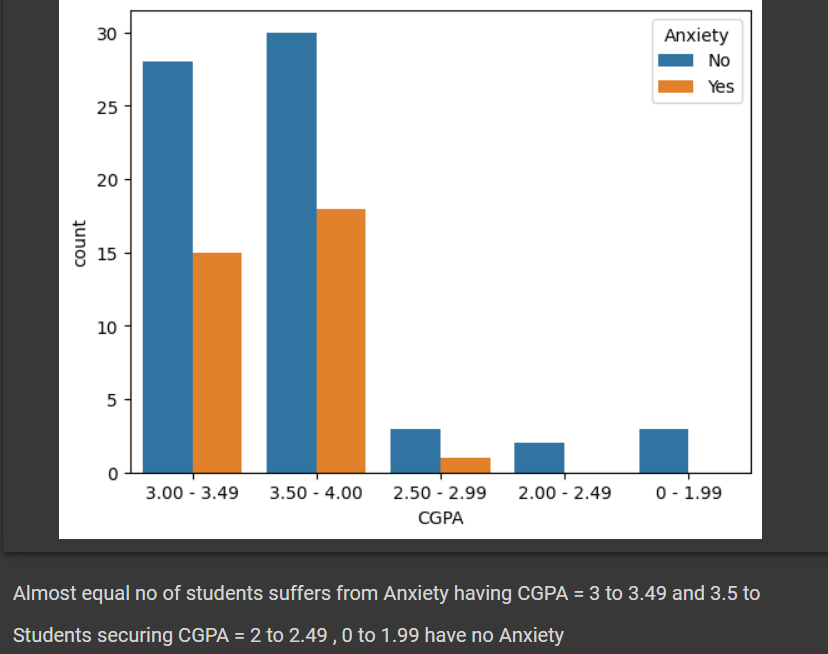
Effect of CGPA on Mental Health

plot\_2 = sns.countplot(x='CGPA',data=df\_1,hue='Depression')

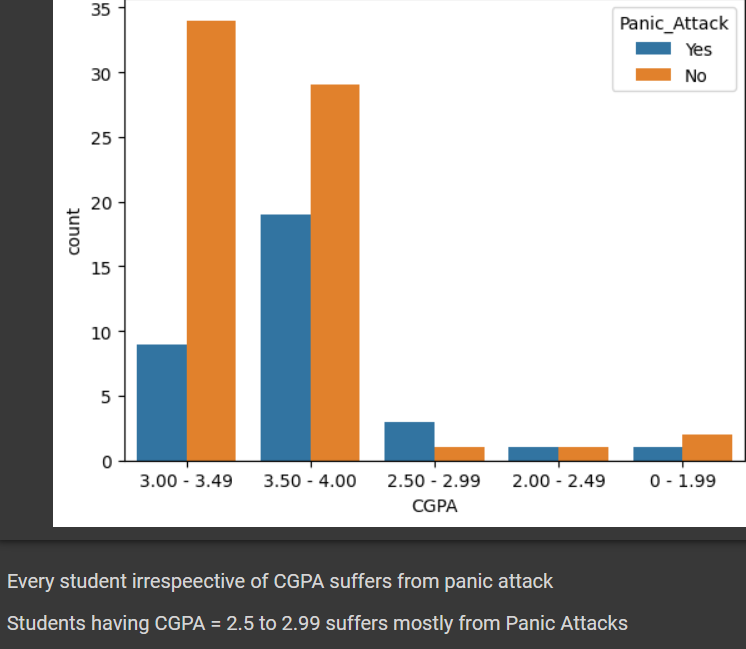
plot\_2.set\_title('Count of CGPA by Depression')



sns.countplot(x='CGPA',data=df\_1,hue='Anxiety')



sns.countplot(x='CGPA',data=df\_1,hue='Panic\_Attack')



Encoding the Categorical Features

df\_2 = df\_1.drop('Date\_Time',axis=1) # Date\_Time only shows the time at which the survey was being taken

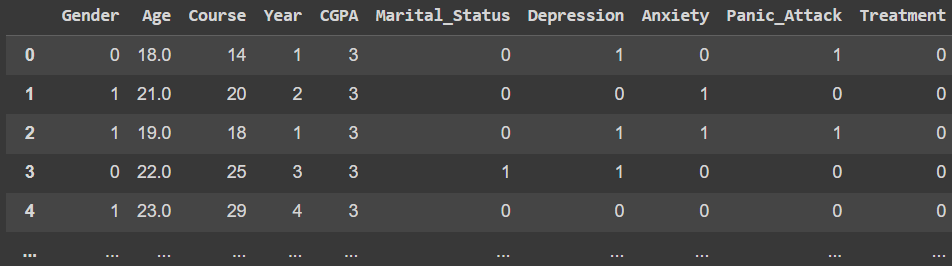
enc = preprocessing.LabelEncoder()

cat\_features = df\_2.select\_dtypes(include=['object']).columns.tolist()

for feature in cat\_features:

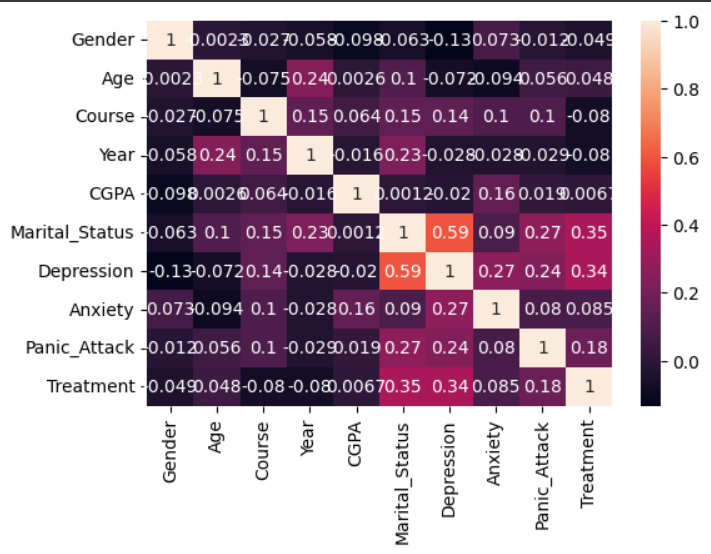
    df\_2[feature] = enc.fit\_transform(df\_2[feature])

df\_2 # Encoded the categorial feature labels

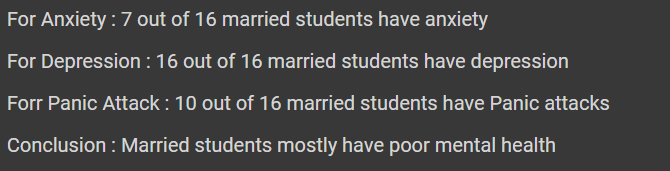
 plt.figure(figsize=(6,4))

cor\_mat = df\_2.corr(method='pearson')

sns.heatmap(cor\_mat,annot=True)







Trainning Model

import pandas as pd

import numpy as np

import tensorflow as tf

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

# Assuming you have already renamed your DataFrame to df\_2

# Preprocessing the data

df\_2['Gender'] = df\_2['Gender'].astype('category').cat.codes

df\_2['Course'] = df\_2['Course'].astype('category').cat.codes

df\_2['Marital\_Status'] = df\_2['Marital\_Status'].astype('category').cat.codes

# Define features and target

X = df\_2.drop(columns=['Treatment'])  # Features

y = df\_2['Treatment']  # Target variable

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize the numerical features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Create a deep learning model

model = tf.keras.Sequential([

    tf.keras.layers.Input(shape=(X\_train.shape[1],)),

    tf.keras.layers.Dense(64, activation='relu'),

    tf.keras.layers.Dense(32, activation='relu'),

    tf.keras.layers.Dense(1, activation='sigmoid')

])

# Compile the model

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_data=(X\_test, y\_test))

# Evaluate the model

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f"Test loss: {loss:.4f}, Test accuracy: {accuracy \* 100:.2f}%")

# Make predictions

y\_pred\_probabilities = model.predict(X\_test)

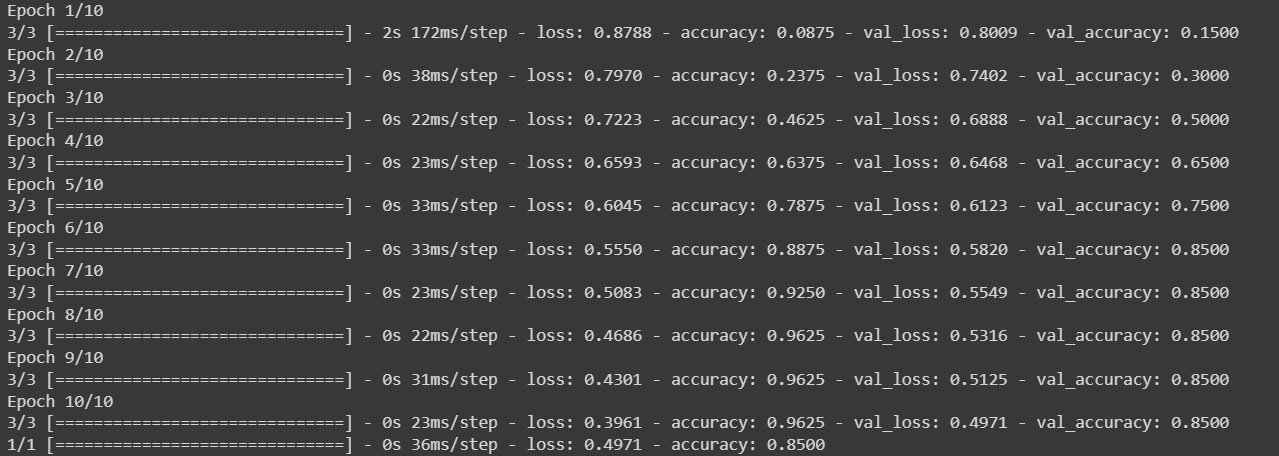
# Calculate risk percentages

risk\_percentages = y\_pred\_probabilities \* 100

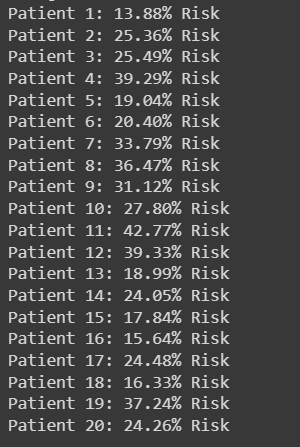
# Display the risk percentages for each patient in the testing dataset

for i, risk\_percentage in enumerate(risk\_percentages):

    print(f"Patient {i+1}: {risk\_percentage[0]:.2f}% Risk")



**Output:**

**Conclusion**

In this project, we have successfully developed a deep learning model to predict the risk of mental health issues in students. Leveraging a range of data preprocessing techniques and powerful libraries, we created an effective model that can analyze a student's demographic and academic features to provide actionable insights regarding mental health risk.

The model's architecture, developed using TensorFlow/Keras, is equipped to capture intricate patterns and relationships within the data. After thorough training and evaluation, it demonstrates its capability to make accurate risk predictions.

The project's significance lies in its potential for early intervention and support in the context of mental health concerns among students. By providing percentage-based risk assessments, it offers a valuable tool for educators and healthcare providers to prioritize their resources and assistance.

**Github link:**

**https://github.com/Jeri20/Codes/tree/main/HealthCare%20Analysis%20Assignment/MiniProject**