

# Predictive Modelling

## Final Project: Exploring Mental Health Data

Presented By **Alamelu Ramanathan**

# Problem Statement

Depression is a complex mental health condition influenced by a variety of factors, including demographic, lifestyle, and psychosocial variables. This study aims to identify and analyze key factors that contribute to the experience of depression in individuals, with the goal of uncovering patterns and potential risk factors. Understanding these relationships will help inform targeted interventions and support strategies to improve mental health outcomes.

# Key Objectives

- **Prediction of Depression onset:** Develop a machine learning model that can accurately classify individuals onset of depression.
- **Understand Key Drivers:** Explore factors that may cause individuals to experience depression.
- **Actionable Insights:** Use the findings to generate insights for healthcare system, policy makers, organizations, educational institutions and individuals to mitigate depression risk.

# Exploratory Data Analysis

Analyse the information in the mental health dataset , find the missing / null values if any and find ways to handle them.

Dataset has 140700 records with 20 columns that could be grouped under the following categories.

- Individual demographics,
- Lifestyle and
- psychosocial information

On analysis it is found that there are more categorical features in the dataset and lot of noise is there.

```
#2. Exploratory data analysis
dsTrain.info()
TARGET = 'Depression'

#Find the features with missing values
print(dsTrain.isnull().sum())

#Analyse the values in the demographics features
print (dsTrain['Age'].value_counts())
print(dsTrain['City'].value_counts())
print(dsTrain['Gender'].value_counts())
print(dsTrain['Profession'].value_counts())
print(dsTrain['Working Professional or
Student'].value_counts())

#Aanlayse the values in psychosocial features
print(dsTrain['Financial Stress'].value_counts())
print(dsTrain['Job Satisfaction'].value_counts())
print(dsTrain['Study Satisfaction'].value_counts())
print(dsTrain['Academic Pressure'].value_counts())
print(dsTrain['Work Pressure'].value_counts())
print(dsTrain['Family History of Mental
Illness'].value_counts())
print(dsTrain['Have you ever had suicidal thoughts
?'].value_counts())
```

# Exploratory Data Analysis contd...

These are findings on analysis,

- Id , name column seems irrelevant for our analysis
- There are around 98 cities 64 professions with high cardinality features
- pressure & satisfaction cols has scale from 1 to 5
- Sleep duration, Dietary habits, Degree has noise
- Age group with higher risk , 18 - 32!!!
- job satisfaction and work satisfaction could be merged
- Academic pressure and work pressure could be merged
- There is no class imbalance as the the ratio is 1:4.5 for the target feature , Depression

```
#Analyse the lifestyle features
print(dsTrain['Sleep Duration'].value_counts())
print(dsTrain['Dietary Habits'].value_counts())
print(dsTrain['Degree'].value_counts())
print(dsTrain['CGPA'].value_counts())
print(dsTrain['Work/Study Hours'].value_counts())

#Analyse age group in order with suicidal thoughts and depression
res = dsTrain[dsTrain['Have you ever had suicidal thoughts?']=='Yes'].groupby(by = ['Age'])
print(res['Age'].value_counts().sort_values(ascending=False))

#Analyse the class imbalance
classCount= dsTrain['Depression'].value_counts()
print(f'class imbalance ratio , majority over minority class : {round(classCount[0]/classCount[1],2)} : 1')
classCount.to_frame().T

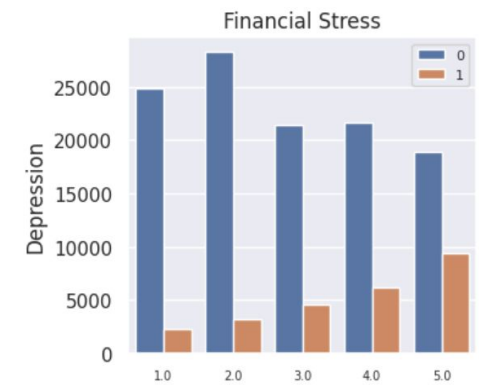
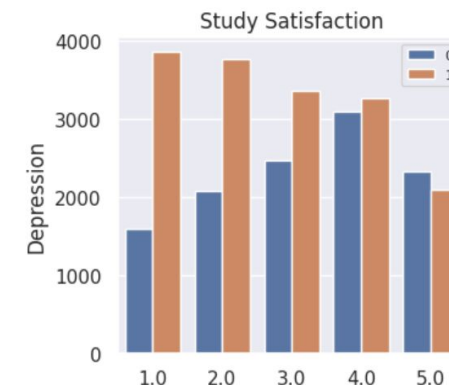
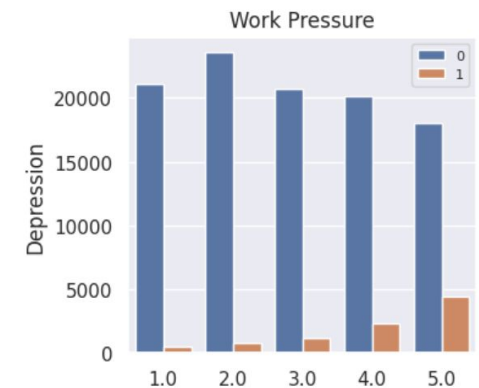
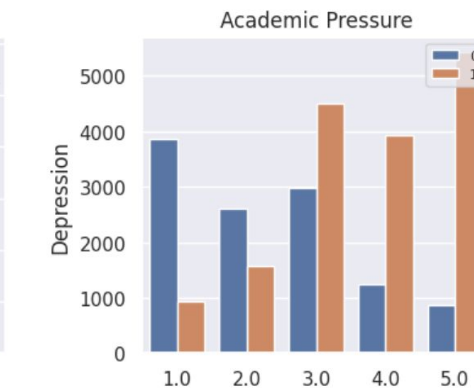
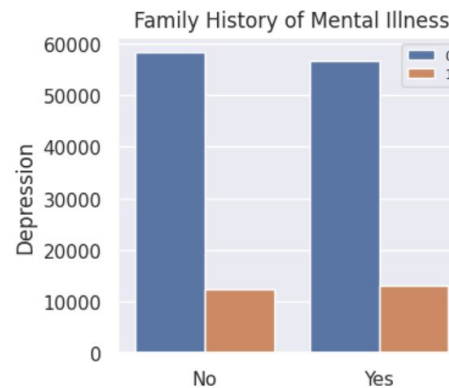
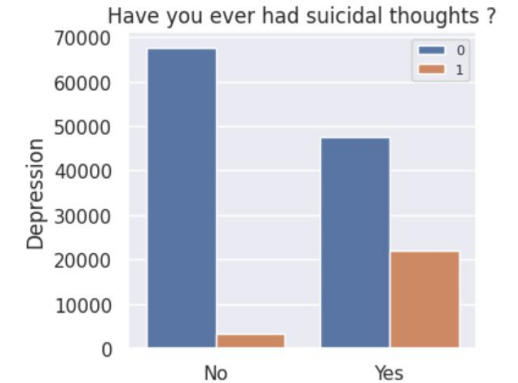
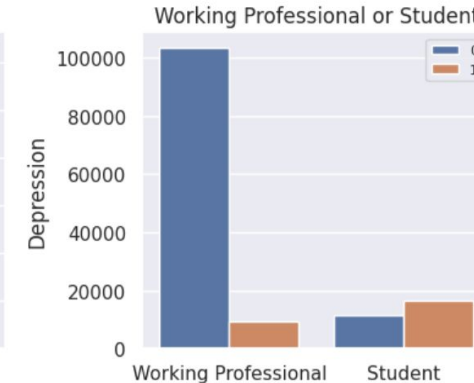
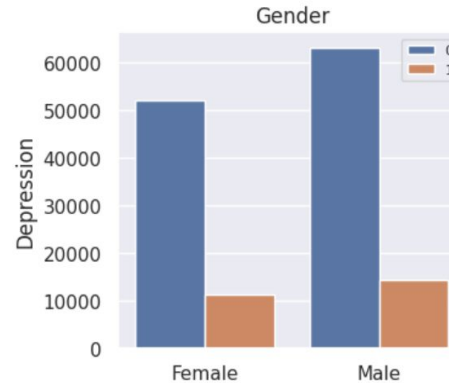
print(dsTrain[dsTrain['Job Satisfaction'].notnull() &
dsTrain['Study Satisfaction'].notnull()])

# F14 No merging conflicts with the pressure columns
dsTrain[dsTrain['Academic Pressure'].notnull() &
dsTrain['Work Pressure'].notnull()]
```

# Visual Insights

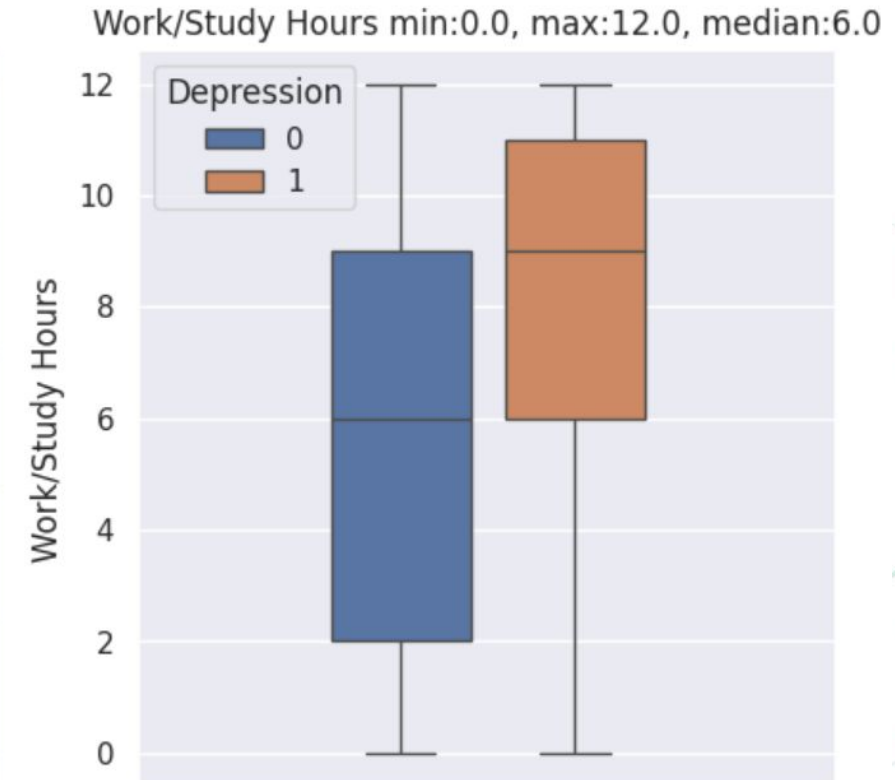
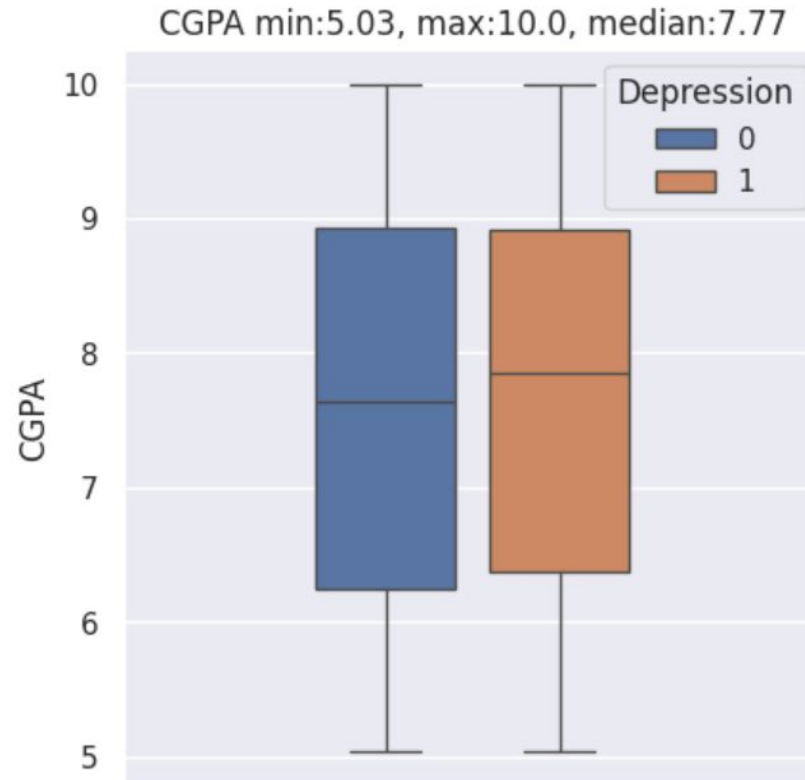
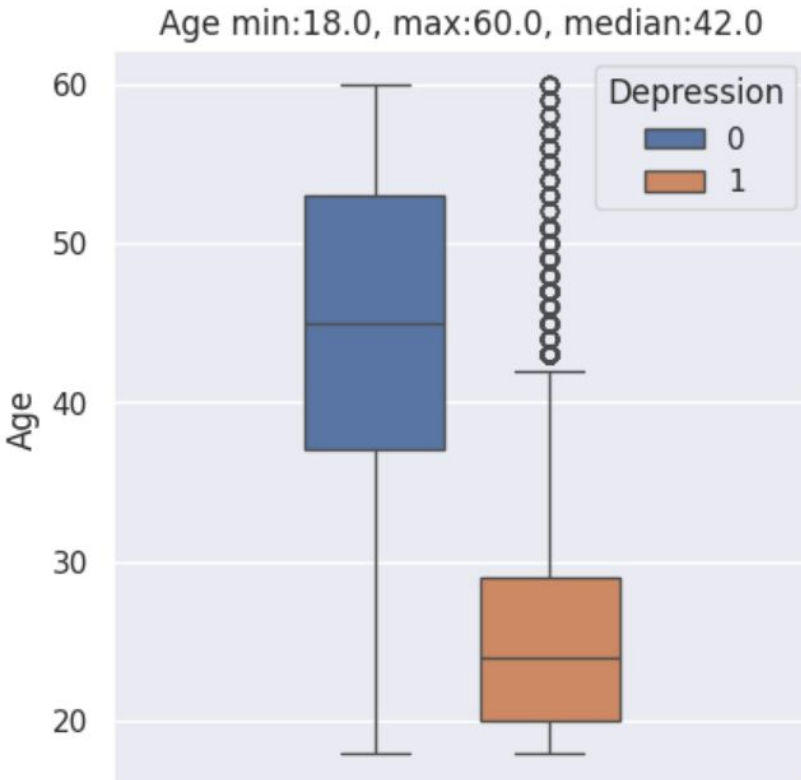
Analysing the features against the target feature 'Depression' gives insights like,

- Financial stress,
  - Work pressure
  - Academic pressure,
  - Suicidal thoughts,
  - Financial stress
- contribute more to Depression.
- Family mental history or no history have the same impact on the target.



# Visual Insights contd...

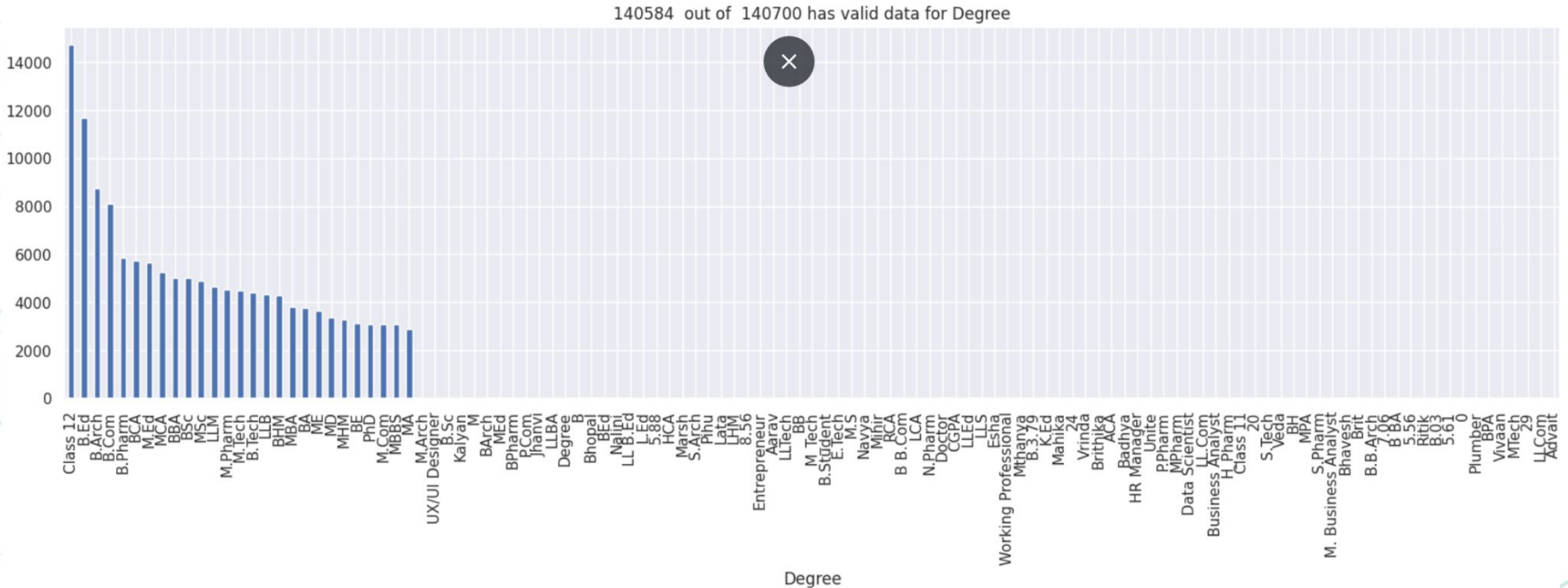
Box plot to visualize distribution of numerical variables in relation to the target class





# Visual Insights contd...

Bar plot to visualize high cardinality data with noises





## Model Selection

CATBoost is the well suited algorithm for this classification problem with high cardinality features because dataset has,

- high cardinality categorical data
- using encoders will lead to sparse matrix

### Training and test data

- Used Train test split from sklearn library to prepare training data and validation data.

```
#Split train and val data
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE

y=processed_Train[TARGET]
X=processed_Train.drop(TARGET, axis = 1)
X_train, X_val, y_train, y_val = train_test_split(X, y,
test_size=0.3, random_state=40)
```

# Training and testing the model

```

from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from matplotlib import pyplot as plt
from catboost import CatBoostClassifier, Pool
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

trainingPool = Pool(X_train, label=y_train, cat_features=catFeatureIndices)
valPool = Pool(X_val, label=y_val, cat_features=catFeatureIndices)

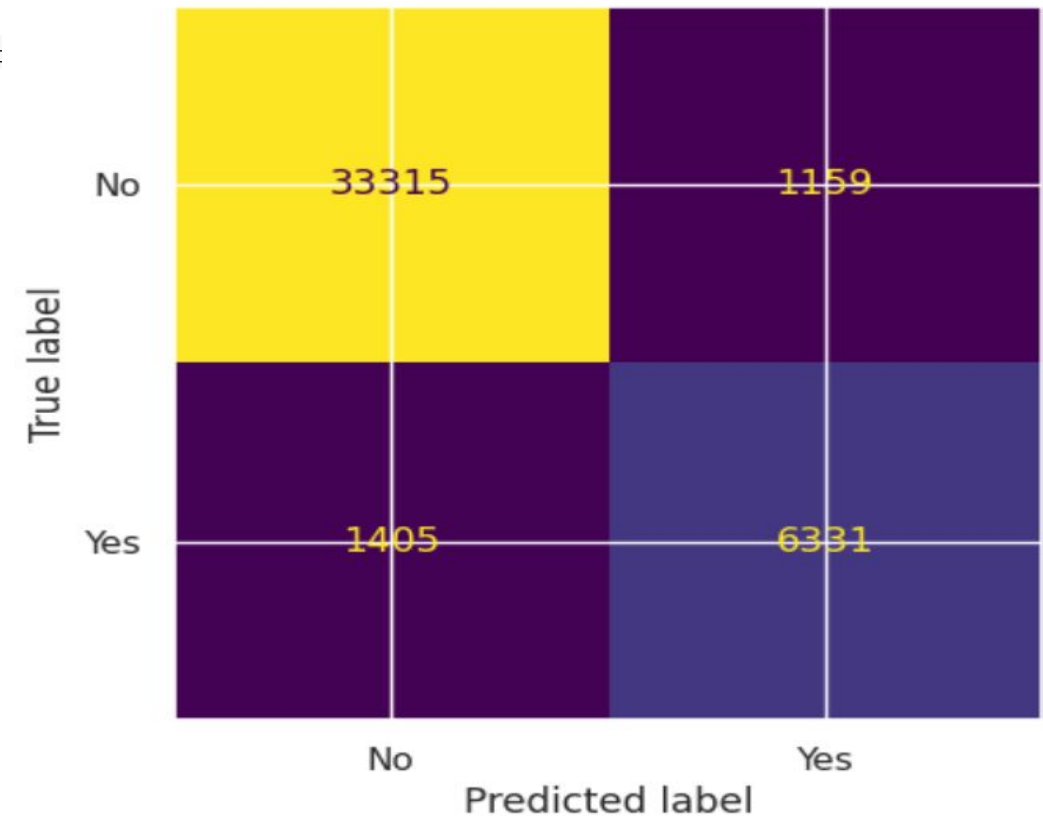
model = CatBoostClassifier(silent=True, iterations=1000, learning_rate=
0.06935994732017255,
                           depth=6, colsample_bylevel=0.2388830216292116,
                           min_data_in_leaf=7)

model.fit(trainingPool)

# predicting for validation data
y_prob=model.predict_proba(valPool)
pred=model.predict(valPool)
report= classification_report(y_val, pred, output_dict=True)
print(report)
fpr, tpr, threshold =roc_curve(y_val, y_prob[:,1])
roc_auc=auc(fpr,tpr)
plt.plot(fpr, tpr, label =f'{model} with roc {roc_auc:.2f}')

```

CatBoostClassifier



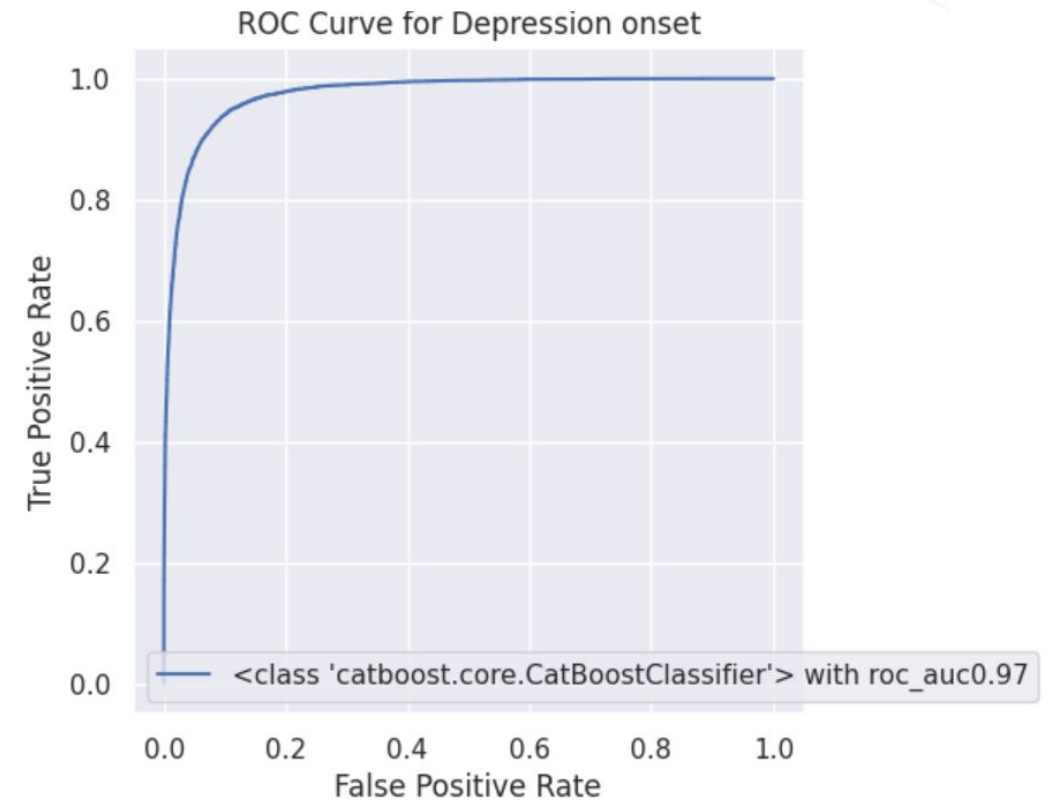
# Classification metrics

Report for **testing data**

	precision	recall	f1-score	support
0	0.96	0.97	0.96	34474
1	0.85	0.82	0.83	7736
<b>accuracy</b>			<b>0.94</b>	42210
macro avg	0.90	0.89	0.90	42210
weighted avg	0.94	0.94	0.94	42210

Report for **training data**

	precision	recall	f1-score	support
0	0.96	0.97	0.97	80659
1	0.86	0.83	0.85	17831
<b>accuracy</b>			<b>0.95</b>	98490
macro avg	0.91	0.90	0.91	98490
weighted avg	0.94	0.95	0.94	98490



# Hyperparameter Tuning

- Optuna is an open source library that automates the process of finding optimal hyperparameters for machine learning and deep learning models.
- Defined an objective function and the number of trials involved as param to optimize on the model parameters.
- A study with the objective is created which focuses on optimizing the objective functions return value, which is accuracy in this case.

```
import optuna
from sklearn.metrics import classification_report
def Objective(test):
    params = {
        "iterations": 1000,
        "learning_rate":
test.suggest_float("learning_rate", 1e-3, 0.1, log=True),
        "depth": test.suggest_int("depth", 1, 10),
        "colsample_bylevel":
test.suggest_float("colsample_bylevel", 0.05, 1.0),
        "min_data_in_leaf":
test.suggest_int("min_data_in_leaf", 1, 100),
    }
    trainingPool = Pool(X_train, label=y_train,
cat_features=catFeatureIndices)
    valPool = Pool(X_val,
label=y_val,cat_features=catFeatureIndices)

    catModel= CatBoostClassifier(**params, silent=True)
    catModel.fit(trainingPool)
    pred=catModel.predict(valPool)
    report= classification_report(y_val, pred,
output_dict=True)
    print(report)
    f1score=report['0']['f1-score'] #Focus on false negative
    accuracy = accuracy_score(y_val,pred)
```

# Feature Importance

<Axes: >



# Conclusion


- Machine learning algorithm, CATBoost is well suited for this problem with 94% accuracy.
- Age, Suicidal thoughts, financial, work and academic stress are the primary factors that contribute to the onset of depression.
- With the insights from mental health predictions,
  - **Government Policymakers:** Allocate resources, develop policies, and prioritize mental health in public agendas.
  - **Public Healthcare Systems:** Set up early intervention programs and enhance crisis intervention services.
  - **Organizations:** Establish mental health support programs and wellness initiatives to reduce stress.
  - **Educational Institutions:** Set up mental health clubs, offer stress reduction programs, and promote mental health literacy.

# Kaggle and further enhancements due...

These are the further enhancements inline to improve the accuracy of the model,

- Feature engineering, merging, the satisfaction columns together and pressure columns together
- Scaling the numerical features like Age, CGPA and study hours.
- Binning the feature variables like Age
- Converting all the noise data to one 'unknown' category.
- Finding the optimal threshold value for the classification.

This leaderboard is calculated with approximately 20% of the test data. The final results will be based on the other 80%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Join
263	Alamelu Ramanathan		0.94253	6	1d	



**Alamelu Ramanathan**  
**alamurm@gmail.com**

Thank You!

Jayanth Rasamsetti

340S, Lemon Avenue, Walnut, California, USA 91789  
4th Floor, T-Hub, IIIT-Hyderabad, Gachibowli, India



Microsoft  
for Startups

[jay@pixeltests.com](mailto:jay@pixeltests.com)

