BucovIA Project

Artificial Intelligence Tool to Detect Bullying ans Stress due to Covid-19

Project Delivery

I. Load Libraries

First, install and import the libraries, functions and classess we will use.

In [1]:
```
# NumPy for numerical computing
import numpy as np

# Pandas for DataFrames
import pandas as pd
pd.set_option('display.max_columns', 100)

# Pickle for reading model files
import pickle

# Scikit-Learn's train_test_split function
from sklearn.model_selection import train_test_split

# Area Under ROC Curve
from sklearn.metrics import roc_auc_score
```

II. Construct Custom Model Class

In [2]:
```
class BullyingDetectionModel:

    # Special function that is automatically run whenever an instance of the clas
    def __init__(self, model_location):
        with open(model_location, 'rb') as f:
            self.model = pickle.load(f)

    # predict_proba() function to apply our model to new data
    def predict_proba(self, X_new, clean=True, augment=True):
            if clean:
                X_new = self.clean_data(X_new)

            if augment:
                X_new = self.engineer_features(X_new)

            return X_new, self.model.predict_proba(X_new)

    # Add functions here
    def clean_data(self, df):
        # Convert the object type features (questions) into numeric values
        df.replace(['SÍ', 'Sí', 'ESTOY DE ACUERDO', 'DE ACUERDO', 'BAI', 'ADOS NA
        df.replace(['NO', 'ESTOY EN DESACUERDO', 'EN DESACUERDO', 'EZ', 'EZ NAGO

        # First drop the possible duplicates
        df = df.drop_duplicates()

        # Fill missing numerical (NaN) values as 0 as it has no influence at all.
        for column in df.select_dtypes(include=['number']):
            df[column].fillna(0, inplace=True)

        # Drop NaN categorical values (drop the entire row)
        df=df.dropna()

        # Femeni0 and Emakumea should be 'Femenino'
        df.Gender.replace(['Femeni0', 'Emakumea'], 'Femenino', inplace=True)

        # Masculi0 and Gizona should be 'Masculino'
        df.Gender.replace(['Masculi0', 'Gizona'], 'Masculino', inplace=True)

        # Beste bat should be 'Otro'
        df.Gender.replace('Beste bat', 'Otro', inplace=True)

        # LANBIDE HEZIKETA 1go maila should be '1º FORMACION PROFESIONAL'
        df.Classroom.replace(['LANBIDE HEZIKETA 1go maila','1º FORMACIÓN PROFESIC

        # The same with DBH
        df.Classroom.replace('DBH 1', '1º ESO',inplace=True)
        df.Classroom.replace('DBH 2', '2º ESO',inplace=True)
        df.Classroom.replace('DBH 3', '3º ESO',inplace=True)
        df.Classroom.replace('DBH 4', '4º ESO',inplace=True)

        # The same procedure for the "age" feature
        df.Age.replace('12 urte', '12 años', inplace=True)
        df.Age.replace('13 urte', '13 años', inplace=True)
        df.Age.replace('14 urte', '14 años', inplace=True)
        df.Age.replace('15 urte', '15 años', inplace=True)
```

```
        df.Age.replace('16 urte', '16 años', inplace=True)
        df.Age.replace('17 urte', '17 años', inplace=True)
        df.Age.replace('18 urte', '18 años', inplace=True)
        df.Age.replace('19 urte', '19 años', inplace=True)
        df.Age.replace('20 urte', '20 años', inplace=True)
        df.Age.replace('21 urte', '21 años', inplace=True)
        df.Age.replace('21 10 gehiago', 'más de 21', inplace=True)

        # Return cleaned dataframe
        return df

    def engineer_features(self, df):
        # Columns with similar content/meaning are gathered up creating a new col
        df['Pain'] = df['Pain1'] + df['Pain2 '] + df['Pain3 '] + df['Pain4 '] + d
        df['Intimidation'] = df['Intimidation1'] + df['Intimidation2']
        df['Humiliation'] = df['Humiliation1'] + df['Humiliation2'] + df['Humilia
        df['Ignore'] = df['Ignore1'] + df['Ignore2'] + df['Ignore3'] + df['Ignore
        df['Digital'] = df['Digital1'] + df['Digital2'] + df['Digital3'] + df['Di
        df['OcurrToMe'] = df['OcurrToMe1'] + df['OcurrToMe2'] + df['OcurrToMe3']
        df['Provoke'] = df['Provoke1'] + df['Provoke2'] + df['Provoke3'] + df['Pr
        df['PassiveObserve'] = df['PassiveObserve1'] + df['PassiveObserve2'] + df
        df['SituationGeneration'] = df['SituationGeneration1'] + df['SituationGer
        df['ActiveObserve'] = df['ActiveObserve1'] + df['ActiveObserve2'] + df['A
        df['Behaviour'] = df['Behaviour1'] + df['Behaviour2'] + df['Behaviour3']
        df['Feeling'] = df['Feeling1'] + df['Feeling2'] + df['Feeling3'] + df['Fe
        df['Thinking'] = df['Thinking1'] + df['Thinking2'] + df['Thinking3'] + df

        # Remove unnecessary columns
        df.drop(df.columns.difference(['Self-analaysis','Interactions', 'Intimida
                                       'PassiveObserve','SituationGeneration','Active
            1, inplace=True)

        # Return augmented DataFrame
        return df
```

III: Algorithm Trial

In [3]:
```
# Initialize an instance
victim_model = BullyingDetectionModel('final_model_observer.pkl')
```

In [4]:
```
# Load raw data
raw_data = pd.read_excel('Prueba_v2.xlsx')
```

In [5]:
```
# Predict raw data
_, pred = victim_model.predict_proba(raw_data, clean=True, augment=True)
```

In [6]:
```
# Get just the prediction for the positive class (0)-(NO)
pred = [p[0] for p in pred]

# Display first 10 predictions
print( np.round(pred,2) )
```

```
[0.4  0.41 0.75 0.55 0.47 0.75 0.96 0.36]
```

In [ ]: