Mental Health Prediction AI

Group 10: Sonika Madhu, Richard Roman, Kara Loomis

Background

Mental Health Prediction AI is an innovative technology designed to address the growing need for early detection and intervention in mental health disorders. With the increasing prevalence of mental health issues globally, there is a pressing demand for tools that can identify at-risk individuals and provide timely support.

The Mental Health Prediction AI will utilize data analysis and data imaging from research and clinical results from Mental Health studies to accurately diagnose those in need of assistance. By implementing this new technology into society, the mental health prediction AI can help lessen the strain on healthcare systems, and enhance millions of people's quality of life anywhere around the world.

Technical Component

One way to use approach this concept is to train an AI model to detect and classify anatomical structures or lesions associated with psychiatric disorders. To build the model one can collect data such as age, gender, ICD coded diagnoses, and information acquired from hospital archives and clinical records. The information from these records (imaging, tests, family history) can classify certain individuals with pre-existing conditions that could be more susceptible to mental illnesses.

Using Supervised Learning

Supervised learning is a subcategory of machine learning which uses labeled datasets to train models to classify data/predict outcomes.

Step 1: Import datasets (from Kaggle) into python using pandas.

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Import any other libraries you may need
import matplotlib.pyplot as plt
import numpy as np
#load your .csv dataset into a dataframe
df=pd.read_csv('CustomDataset/scores1.csv')
```

Using Supervised Learning

Step 2: Examine the data by checking the number of rows and columns in our dataset and look at a summary to check for null values.

```
# Use the shape member variable to observe the shape of our dataset.
df.shape
```

```
(23, 12)
```

```
# See a summary of the data to check for null values.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23 entries, 0 to 22
Data columns (total 12 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   number     23 non-null     object
 1   days       23 non-null     int64 
 2   gender     23 non-null     int64 
 3   age        23 non-null     object
 4   afftype    23 non-null     int64 
 5   melanch    23 non-null     int64 
 6   inpatient  23 non-null     int64 
 7   edu        23 non-null     object
 8   marriage   23 non-null     int64 
 9   work       23 non-null     int64 
 10  madrs1     23 non-null     int64 
```

Using Supervised Learning

Step 3: Clean the data by dropping any redundant columns or columns with data that you don't want to be used. For our purposes we removed number, days, age, and gender.

We can also see the statistical information using the describe function.

```
# Inspect statistical information about the data set
df.describe()
```

	afftype	melanch	inpatient	marriage	work	madrs1	madrs2
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000
mean	1.739130	1.695652	1.782609	1.521739	1.869565	22.739130	20.000000
std	0.540824	0.702902	0.421741	0.510754	0.344350	4.797892	4.729021
min	1.000000	0.000000	1.000000	1.000000	1.000000	13.000000	11.000000
25%	1.000000	2.000000	2.000000	1.000000	2.000000	18.500000	16.000000
50%	2.000000	2.000000	2.000000	2.000000	2.000000	24.000000	21.000000
75%	2.000000	2.000000	2.000000	2.000000	2.000000	26.000000	24.500000
max	3.000000	2.000000	2.000000	2.000000	2.000000	29.000000	28.000000

```
# Drop any redundant columns from your data.
df.drop(['number', 'days', 'gender', 'age'], axis=1, inplace=True)
```

Using Supervised Learning

Step 2: Filter outliers and replace the outlier data with the average of the data within the range

Also drop columns that are unnecessary or missing too many values

```
# Calculate the 25th and 75th percentiles of the 'age' column
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)

# Calculate the interquartile range (IQR)
IQR = Q3 - Q1

# Define the upper and lower bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter outliers using boolean indexing
df_filtered = df[(df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)]
avg_age = df_filtered['Age'].mean()
outliers_index = (df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)

df.loc[outliers_index, 'Age'] = avg_age
#drops unneeded columns
df = df.drop('comments', axis=1)
df = df.drop('state', axis=1)
df = df.drop('work_interfere', axis=1)
df = df.drop('self_employed', axis=1)
```

Using Supervised Learning

Step 4: import DecisionTreeClassifier to create a decision tree.

Using the data provided and the help of sentiment analysis the polarity of text was analyzed along with the other numerical data and a training and testing set was made by separating the initial data set the target

```
#split data set into testing and training set
#
train_df, testa_df = train_test_split(df, test_size = 0.2, random_state=87)
train_df_copy = train_df.copy()

train_df_copy['mental_health_issues'] = ((train_df_copy['treatment_nums'] == 1) & (train_df_copy['mental_health_consequence_nums'] >= 1)).astype(int)
testa_df['mental_health_issues'] = ((testa_df['treatment_nums'] == 1) & (testa_df['mental_health_consequence_nums'] >= 1)).astype(int)
```

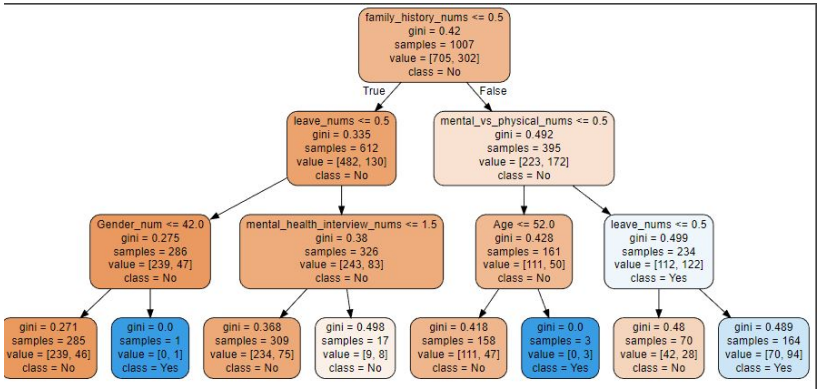
Code for Decision tree

```
#split data set into testing and training set
#
train_df, testa_df = train_test_split(df, test_size = 0.2, random_state=87)
train_df_copy = train_df.copy()

train_df_copy['mental_health_issues'] = ((train_df_copy['treatment_nums'] == 1) & (train_df_copy['mental_health_consequence_nums'] >= 1)).astype(int)
testa_df['mental_health_issues'] = ((testa_df['treatment_nums'] == 1) & (testa_df['mental_health_consequence_nums'] >= 1)).astype(int)



#tech tree portion after data filtering
x_train = train_df_copy[['Age','family_history_nums','mental_vs_physical_nums','mental_health_interview_nums','Gender_num','leave_nums']].values
y_train = train_df_copy[['mental_health_issues']].values

y_test = testa_df[['mental_health_issues']].values
x_test = testa_df[['Age','family_history_nums','mental_vs_physical_nums','mental_health_interview_nums','Gender_num','leave_nums']].values

clf = tree.DecisionTreeClassifier()

clf = clf.fit(x_train,y_train)

y_pred = clf.predict(x_test)
tree.plot_tree(clf)

print(accuracy_score(y_test, y_pred))
```

Using Supervised Learning

Step 4: import DecisionTreeClassifier to create a decision tree.

The model was then used with a Decision Tree classifier and the accuracy was then tested which resulted in an accuracy of 70.63%



Ethical Impact of using AI in Mental Health Prediction

With the use of any technology is it important to consider the ethical impacts. We believe it is crucial to be transparent and responsible when sourcing and using patient data to build/train ML models. We can do this by clearly asking patients for permission to use their personal data and telling them what it will be used for. We would also ensure that there is no bias or discrimination while building/training the models that could impact a certain group or sacrifice the quality of the AI-based research.

Another big part would be ensuring the security of the collected data. Legally, there are very strict laws in place to protect patient data such as HIPAA and this should be taken into account when collecting and using the data for research purposes.

Sources

https://www.kaggle.com/datasets/arashnic/the-depression-dataset

https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey

https://www.frontiersin.org/articles/10.3389/fsurg.2022.862322/full