DMQL Project Proposal

Project Name:

Automated system to identify mental disorder diagnosed people via tweet patterns.

Team Name: Dino Squad

Team Members:

Akash Ponduru (akashpon), Abhishek Kumar Singh (singh33), Kalali Bhargav Reddy (bkalali)

Problem Statement:

In a world where people are connected via social media, the internet, and other channels more than ever, the rate of mental disorders like depression, PTSD, anxiety, etc. is the highest among the population. As per the news and research, people who have been diagnosed are not able to live normal lives and eventually take drastic measures to end their lives or are involved in drugs or other forms of substances. fortunately, the chance of identifying such individuals at the earlier stage is not very difficult, the easiest way would be to identify patterns that display symptoms of mental illness.

We are proposing an automated system that keeps track of the tweeting patterns of users identify them into 7 different classes:

- control: People who are living are free from any mental disorders.
- disorder: People who are suffering from mental disorders. This class is further subdivided into depression, anxiety, PTSD, bipolar, borderline, panic.

The original dataset that we got from Kaggle are excel files, the main reason we want to use database instead of excel files are following:

- Size of data: the sheer amount of data generated by social networking websites is enormous and handling them using excel sheets leads to system lags, memory crashes.
- File handling: Although we assume that the size of data is not an issue, but the risk of file corruption is too much.
- Privacy: Handling sensitive tasks like we mentioned and leaving the information of users in open excel sheets which anyone can open is a huge risk of privacy.
- Datatype of attributes: Most of our data is different attributes are text, we are trying to build a system which makes use of it. We intend to make use of vector databases to store this text information in encoded format for better processing and evaluation.

Dataset: Twitter mental disorder tweets and music dataset

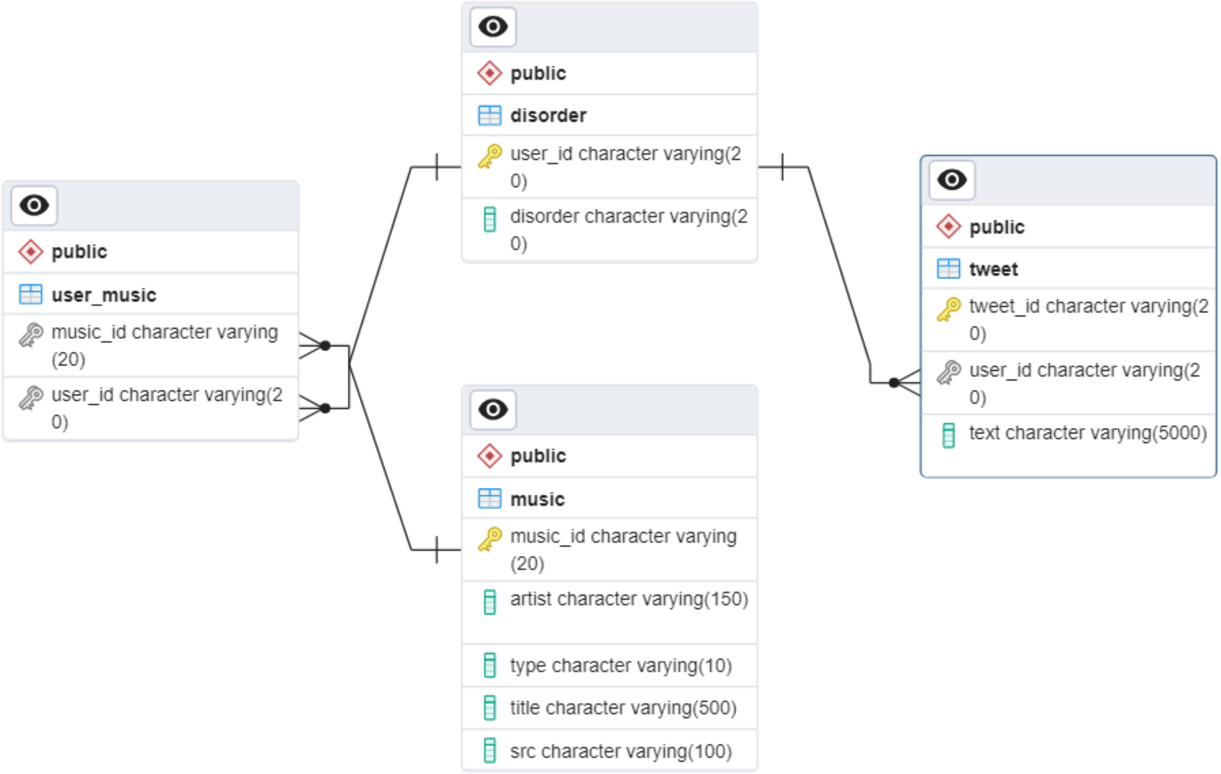
Link: https://www.kaggle.com/datasets/rrmartin/twitter-mental-disorder-tweets-and-musics/data

Target User:

Currently, we do not have a set of users in mind, but anyone from research institutions and social media companies can use the database to work on sensitive topics and build solutions which can help people suffering from mental disorders.

People who are technically sound and are in respectable positions should only administer the database for example: Managers handling privacy, people setting up infrastructures, should only administer the database under supervision and should log all changes or improvements performed.

Entity Relationship Diagram:


public
disorder
user_id character varying(20)
disorder character varying(20)
public
user_music
music_id character varying (20)
user_id character varying(20)
public
tweet
tweet_id character varying(20)
user_id character varying(20)
text character varying(5000)
public
music
music_id character varying (20)
artist character varying(150)
type character varying(10)
title character varying(500)
src character varying(100)

The source for this E/R diagram is two excel files: tweets.csv and music.csv incorporating different attributes mentioned in the above diagram. We removed partial and transitive dependency to make sure that the data is 3NF and BCNF compliant.

Disorder Table: The table contains the unique user ID and the associated disorder they are suffering from; each row indicates a unique user.

Tweets Table: The table contains all the tweets posted by people who are present in the disorder table. Each tweet has been assigned a unique id to suffice the normalization constraint. Apart from that we can use user_id to retrieve all the tweets from a specific user if needed.

Music Table: The table contains records of unique entries of music listened by the users present in disorder table, each row is a unique track.

User_Music Table: The table contains mapping of users and the music they listen to. The table has been created to eliminate transitive dependencies in Music table.

Database Design:

The four table that were obtained after decomposition, all follow BCNF.

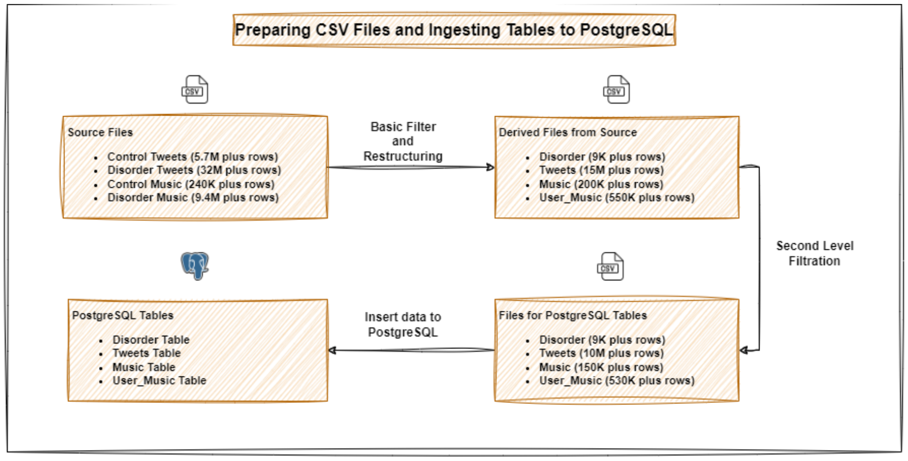
1. **User_Music Table:**
   Here the table is in BCNF as any table that has two attributes, and they are always in BCNF.

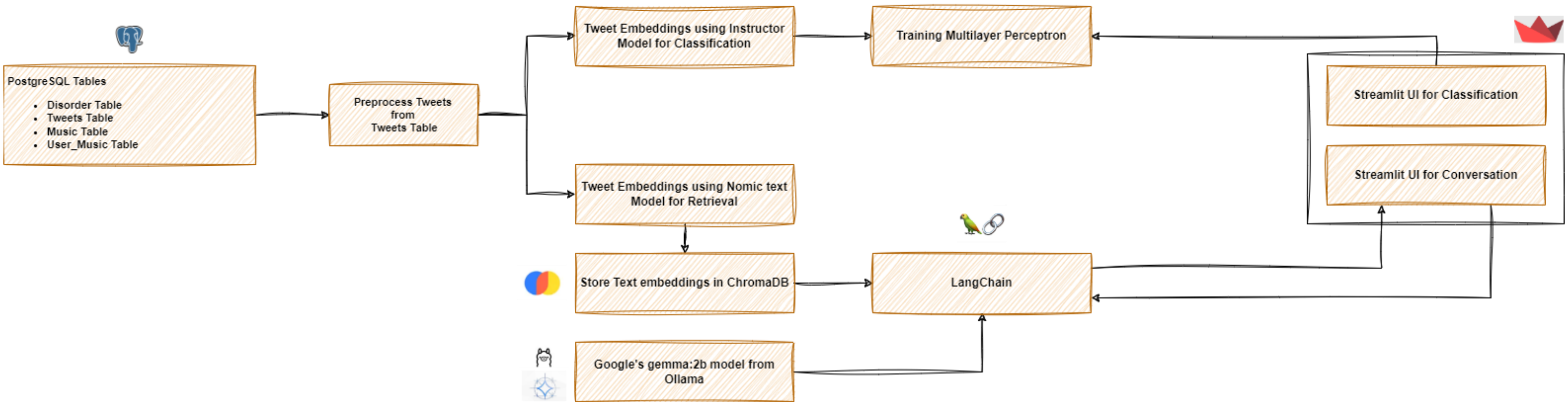
2. **Disorder Table:**
   Here too, the table is in BCNF as this table too has two attributes, and they are always in BCNF.

3. **Music Table:**
   The music_id acts as a unique identifier for each music record. Non-key attributes depend only on the music_id, and no other dependencies exist, fulfilling the BCNF criteria.

4. **Tweet Table:**
   The tweet_id is the primary key. The text of the tweet is dependent on tweet_id, and user_id provides a link to the user who tweeted. There are no dependencies between non-key attributes, thereby satisfying BCNF.

The Data Flow:


Preparing CSV Files and Ingesting Tables to PostgreSQL
Source Files
Control Tweets (5.7M plus rows)
Disorder Tweets (32M plus rows)
Control Music (240K plus rows)
Disorder Music (9.4M plus rows)
Basic Filter and Restructuring
Derived Files from Source
Disorder (9K plus rows)
Tweets (15M plus rows)
Music (200K plus rows)
User_Music (550K plus rows)
Second Level Filtration
Files for PostgreSQL Tables
Disorder (9K plus rows)
Tweets (10M plus rows)
Music (150K plus rows)
User_Music (530K plus rows)
Insert data to PostgreSQL
PostgreSQL Tables
Disorder Table
Tweets Table
Music Table
User_Music Table


PostgreSQL Tables
Disorder Table
Tweets Table
Music Table
User_Music Table
Preprocess Tweets from Tweets Table
Tweet Embeddings using Instructor Model for Classification
Training Multilayer Perceptron
Tweet Embeddings using Nomic text Model for Retrieval
Store Text embeddings in ChromaDB
LangChain
Google's gemma:2b model from Ollama
Streamlit UI for Classification
Streamlit UI for Conversation

1. In the presented data flow, the initial corpus consisted of a voluminous dataset arranged into two primary CSV files: one related to tweets and the other to music preferences, both targeting general individuals as well as those with mental disorders.
2. An initial assessment of the dataset was conducted, following which a filtration process was executed. This involved removing records that exclusively contained non-textual data such as images or links. Adding to them, any entries exhibiting NULL values or duplicates were excluded to ensure the integrity of the dataset.
3. With the cleaned dataset, we proceeded to assign Super Keys for the Tweets and Music tables, identified by 'tweet_id' and 'music_id' respectively. This served to enhance the data's organization and retrieval efficiency.
4. A second round of cleaning was performed, which specifically targeted tweets containing inappropriate language or those that were excessively brief (under five words), further refining the dataset's quality.
5. Following the data cleansing, we progressed to the storage phase, inputting the purified data into a PostgreSQL database with a custom script designed to prevent any data leakage. The script facilitated precise table creation and data insertion.

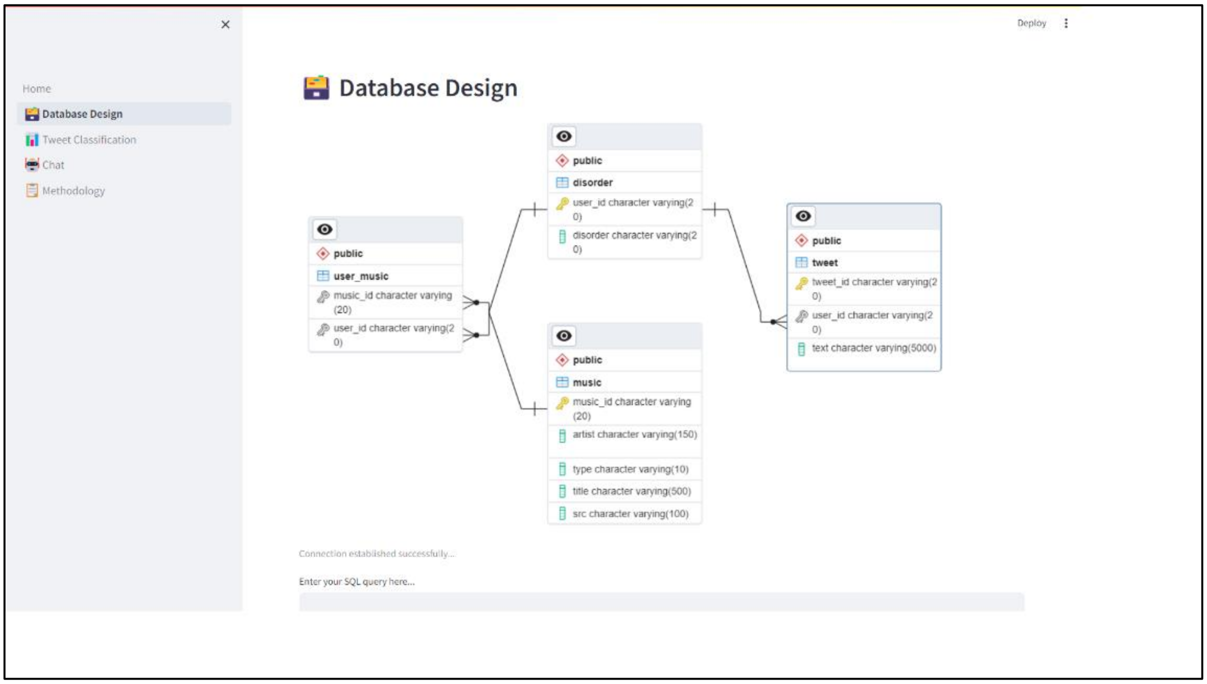
6. Subsequently, a subset of tweets characterized by a word count ranging between 40 to 50 was extracted from the database to generate text embeddings. These embeddings were generated by Instructor Text Embedding base model.
7. In parallel, a distinct embedding for retrieval tasks was generated using the NOMIC Text Embed model from the Ollama suite.
8. The next stage entailed training a Multilayer Perceptron (MLP) model on the derived text embeddings, facilitating the classification of the data into seven distinct categories. This model was then integrated into a Streamlit user interface, allowing users to input their data and receive instant probability assessments for each class.
9. For efficient retrieval, we stored the alternative embeddings within ChromaDB. Inference tasks were handled by leveraging Google's gemma:2b model from Ollama, selected for its proven efficacy.
10. Finally, we used Langchain framework as the cohesive element linking our retrieval system, the language model, and the user input interface. This setup ensured that users could interact with our system in a straightforward and effective manner, receiving accurate outputs in response to their inputs.

Welcome to Our Tweet Classification Portal!

Our application is designed to classify tweets for research on mental health. Here's how you can navigate through our platform and use it effectively:

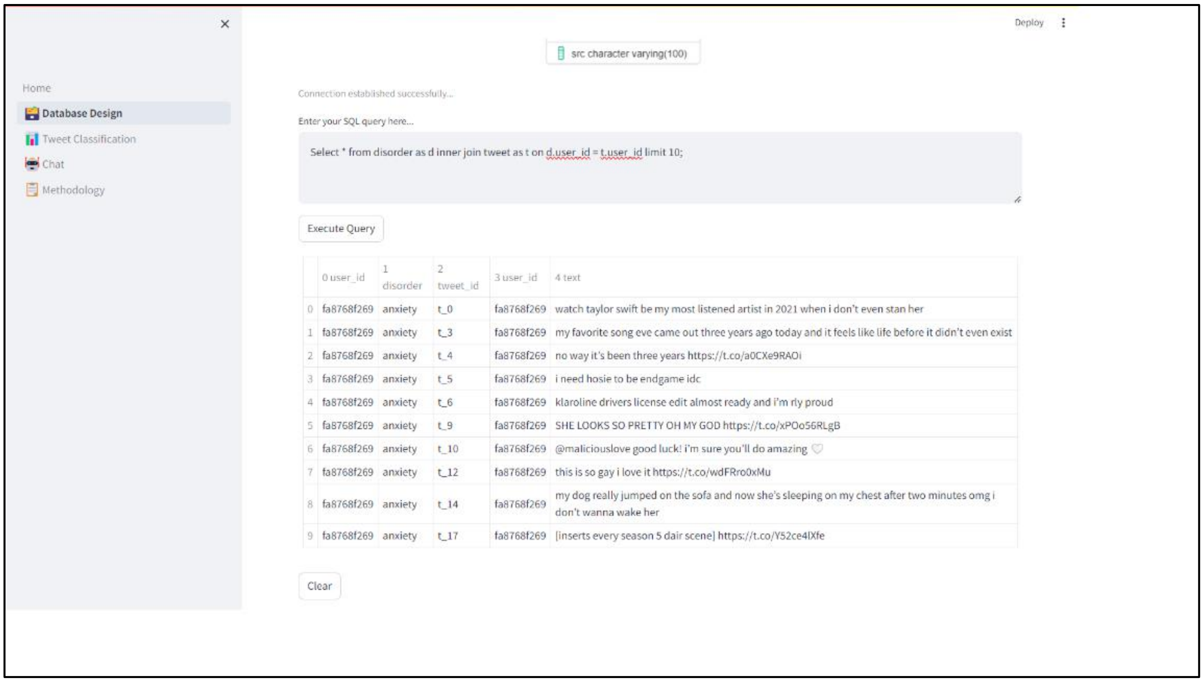
1. Database Design:

Here you can run a SQL query, that can be used either to retrieve data or insert data in the database. It also displays the sample data retrieved. On the top you can get an idea of the database schema and how the tables are connected.


Deploy
Home
Database Design
Tweet Classification
Chat
Methodology
Database Design
public
disorder
user_id character varying(20)
disorder character varying(20)
public
user_music
music_id character varying (20)
user_id character varying(20)
public
tweet
tweet_id character varying(20)
user_id character varying(20)
text character varying(5000)
public
music
music_id character varying (20)
artist character varying(150)
type character varying(10)
title character varying(500)
src character varying(100)

Connection established successfully...

Enter your SQL query here...



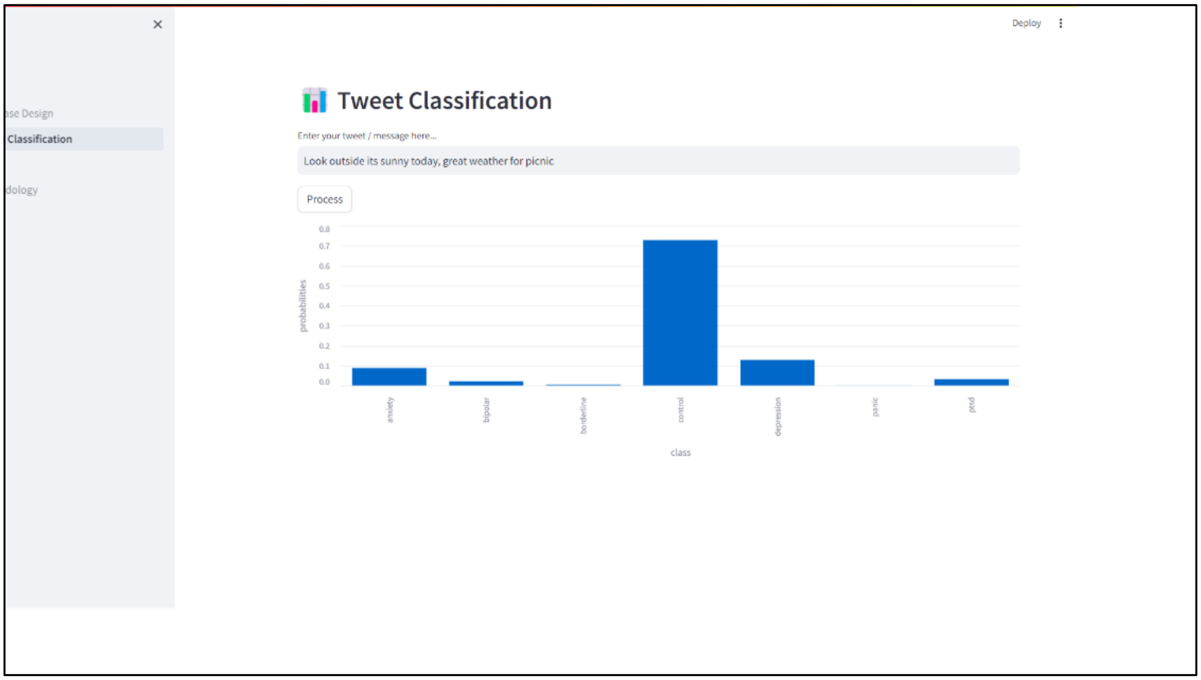
Connection established successfully...

Enter your SQL query here...

```
Select * from disorder as d inner join tweet as t on d.user_id = t.user_id limit 10;
```

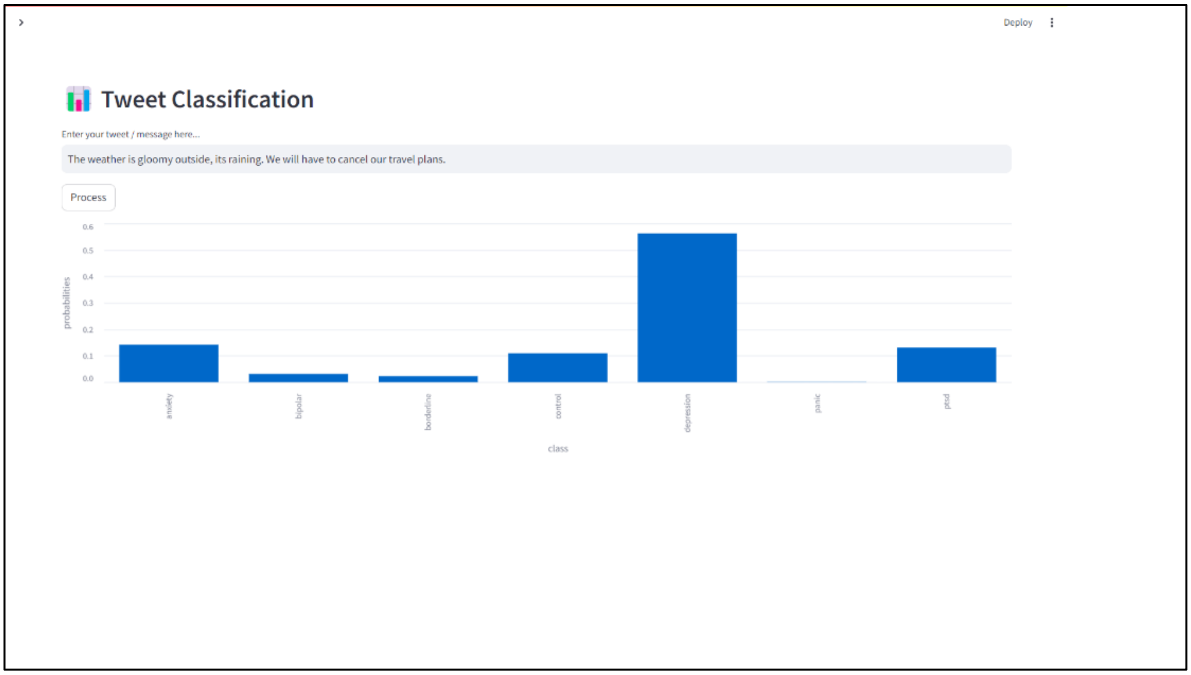
	0 user_id	1 disorder	2 tweet_id	3 user_id	4 text
0	fa8768f269	anxiety	t_0	fa8768f269	watch taylor swift be my most listened artist in 2021 when i don't even stan her
1	fa8768f269	anxiety	t_3	fa8768f269	my favorite song eve came out three years ago today and it feels like life before it didn't even exist
2	fa8768f269	anxiety	t_4	fa8768f269	no way it's been three years https://t.co/a0CXe9RAOi
3	fa8768f269	anxiety	t_5	fa8768f269	i need hosie to be endgame idc
4	fa8768f269	anxiety	t_6	fa8768f269	klaroline drivers license edit almost ready and i'm rly proud
5	fa8768f269	anxiety	t_9	fa8768f269	SHE LOOKS SO PRETTY OH MY GOD https://t.co/xPOo56RLgB
6	fa8768f269	anxiety	t_10	fa8768f269	@maliciouslove good luck! i'm sure you'll do amazing ♡
7	fa8768f269	anxiety	t_12	fa8768f269	this is so gay i love it https://t.co/wdFRro0xMu
8	fa8768f269	anxiety	t_14	fa8768f269	my dog really jumped on the sofa and now she's sleeping on my chest after two minutes omg i don't wanna wake her
9	fa8768f269	anxiety	t_17	fa8768f269	[inserts every season 5 dair scene] https://t.co/Y52ce4lXfe

2. Tweet Classification:



Enter your tweet / message here...

Look outside its sunny today, great weather for picnic



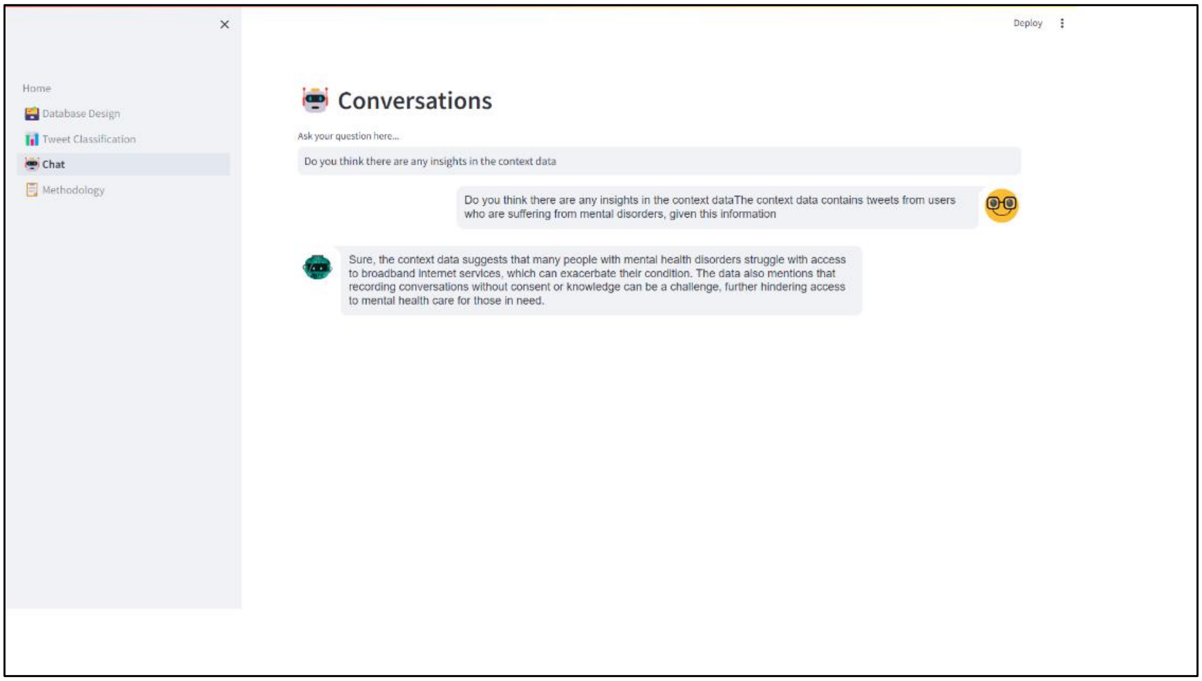
Tweet Classification

Enter your tweet / message here...

The weather is gloomy outside, its raining. We will have to cancel our travel plans.

Here a user can enter a tweet and the system will analyze the sentiment of your text using the underlying machine learning model. The results of the classification will be displayed in a bar graph format. Each bar represents a class of sentiment such as 'anxiety', 'bipolar', 'borderline', 'control', 'depression', 'panic', and 'ptsd'. Here ‘control’ refers to not having any identified mental illness. The height of the bar indicates the probability that your text belongs to that class.

3. Chat:



Conversations

Ask your question here...

Do you think there are any insights in the context data

Do you think there are any insights in the context dataThe context data contains tweets from users who are suffering from mental disorders, given this information

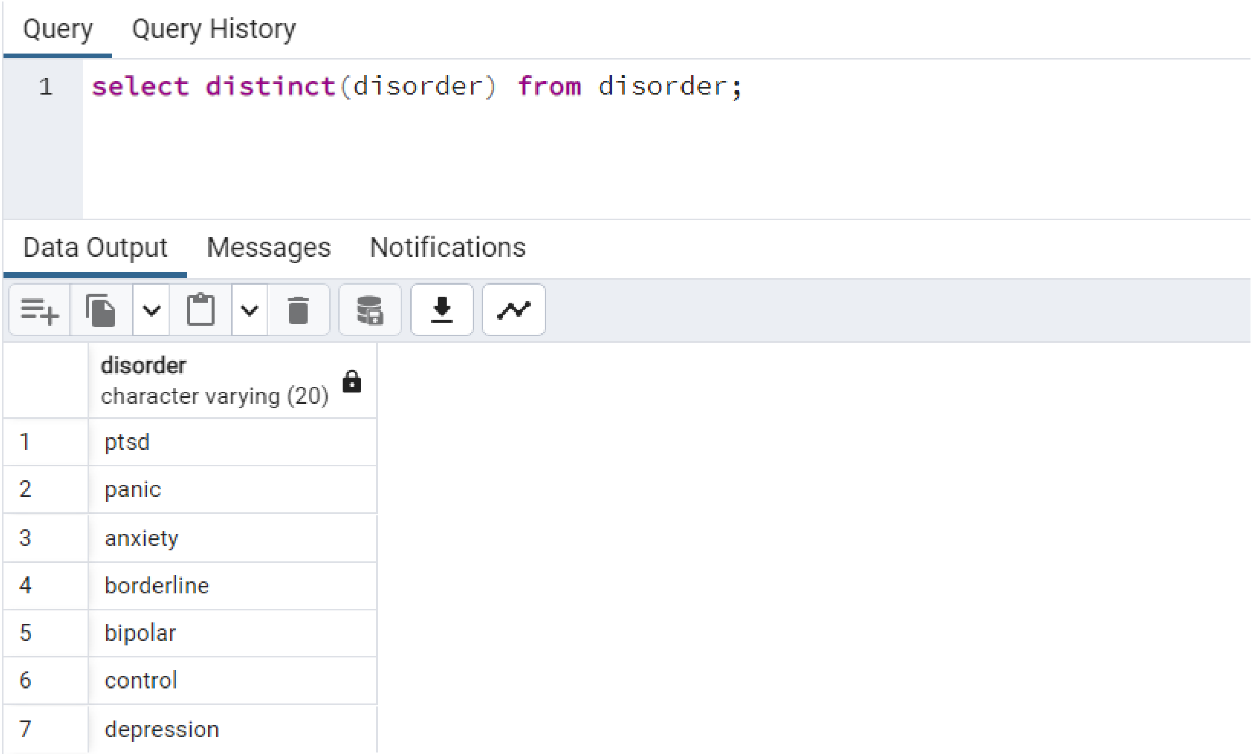
Sure, the context data suggests that many people with mental health disorders struggle with access to broadband internet services, which can exacerbate their condition. The data also mentions that recording conversations without consent or knowledge can be a challenge, further hindering access to mental health care for those in need.

Here a user can interact and seek answers to their questions related to our data and its implications. Users can enter their query about our data and the bot will provide and answer to them. The bot analyzes the data and provides insights such as challenges faced by individuals with mental disorders or any patterns that can be found.

SQL Queries:

# Query to get the list of the distinct disorders in the dataset

select distinct(disorder) from disorder;

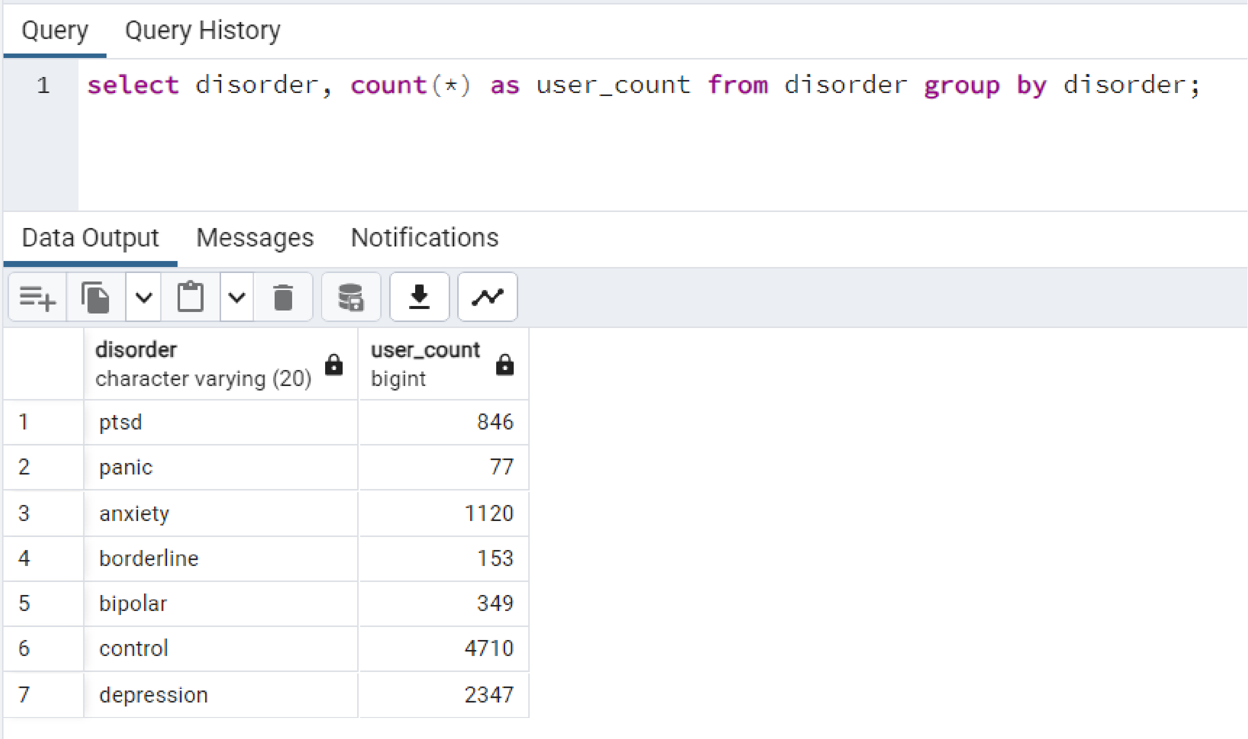


```
select distinct(disorder) from disorder;
```

	disorder character varying (20)
1	ptsd
2	panic
3	anxiety
4	borderline
5	bipolar
6	control
7	depression

# Query to calculate the count of users in the dataset based on the unique disorder

select disorder, count(*) as user_count from disorder group by disorder;

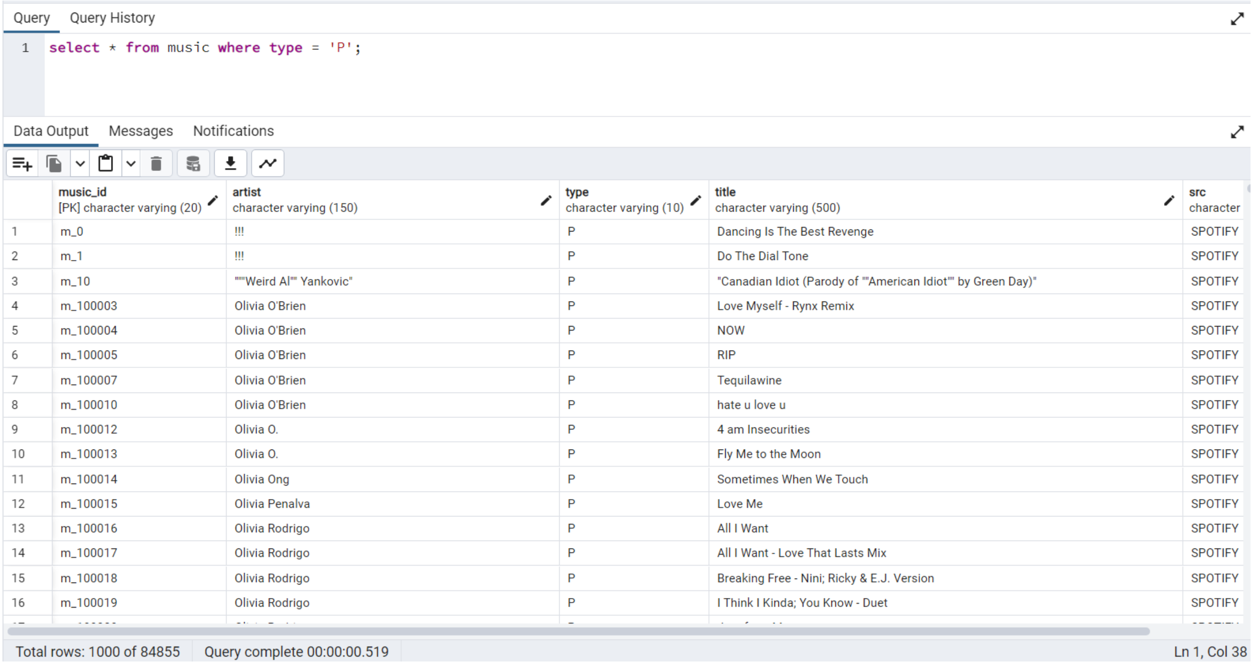


```sql
select disorder, count(*) as user_count from disorder group by disorder;
```

	disorder character varying (20)	user_count bigint
1	ptsd	846
2	panic	77
3	anxiety	1120
4	borderline	153
5	bipolar	349
6	control	4710
7	depression	2347

# Query to select all the rows from music relation where type is P

select * from music where type = 'P';



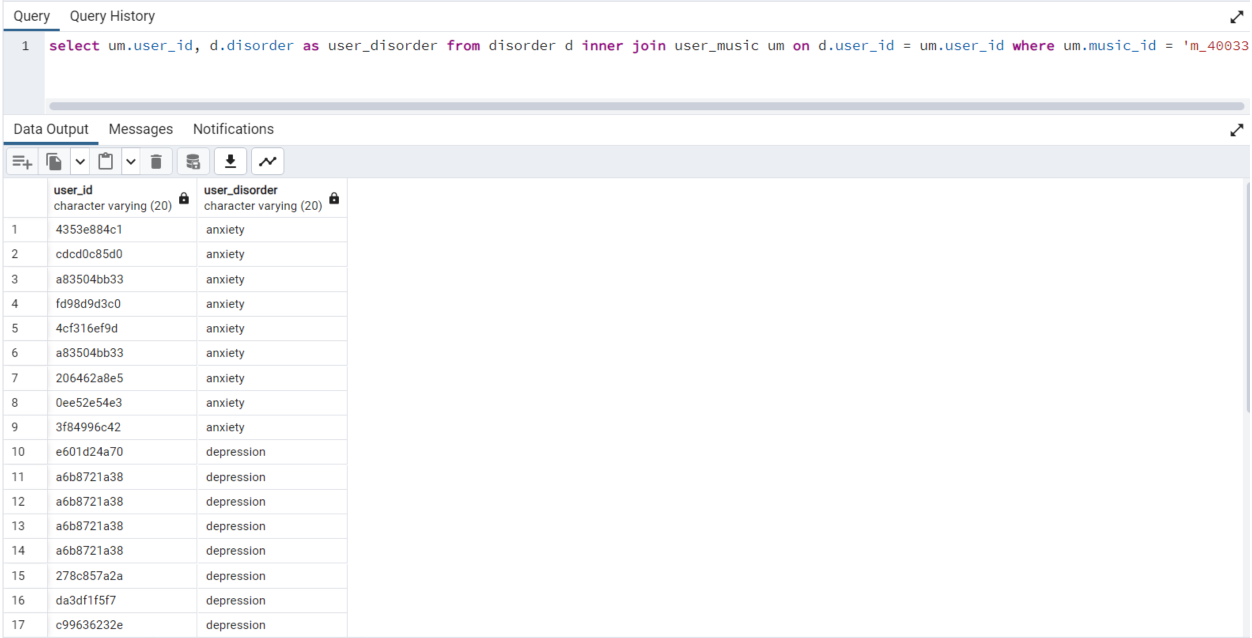
```sql
select * from music where type = 'P';
```

	music_id [PK] character varying (20)	artist character varying (150)	type character varying (10)	title character varying (500)	src character
1	m_0	!!!	P	Dancing Is The Best Revenge	SPOTIFY
2	m_1	!!!	P	Do The Dial Tone	SPOTIFY
3	m_10	"""Weird Al""" Yankovic"	P	"Canadian Idiot (Parody of ""American Idiot"" by Green Day)"	SPOTIFY
4	m_100003	Olivia O'Brien	P	Love Myself - Rynx Remix	SPOTIFY
5	m_100004	Olivia O'Brien	P	NOW	SPOTIFY
6	m_100005	Olivia O'Brien	P	RIP	SPOTIFY
7	m_100007	Olivia O'Brien	P	Tequilawine	SPOTIFY
8	m_100010	Olivia O'Brien	P	hate u love u	SPOTIFY
9	m_100012	Olivia O.	P	4 am Insecurities	SPOTIFY
10	m_100013	Olivia O.	P	Fly Me to the Moon	SPOTIFY
11	m_100014	Olivia Ong	P	Sometimes When We Touch	SPOTIFY
12	m_100015	Olivia Penalva	P	Love Me	SPOTIFY
13	m_100016	Olivia Rodrigo	P	All I Want	SPOTIFY
14	m_100017	Olivia Rodrigo	P	All I Want - Love That Lasts Mix	SPOTIFY
15	m_100018	Olivia Rodrigo	P	Breaking Free - Nini; Ricky & E.J. Version	SPOTIFY
16	m_100019	Olivia Rodrigo	P	I Think I Kinda; You Know - Duet	SPOTIFY

Total rows: 1000 of 84855 Query complete 00:00:00.519 Ln 1, Col 38

# Query to select all users' disorders which listen to music with music id m_40033

select um.user_id, d.disorder as user_disorder from disorder d inner join user_music um on d.user_id = um.user_id where um.music_id = 'm_40033';



```
select um.user_id, d.disorder as user_disorder from disorder d inner join user_music um on d.user_id = um.user_id where um.music_id = 'm_40033
```

	user_id character varying (20)	user_disorder character varying (20)
1	4353e884c1	anxiety
2	cdcd0c85d0	anxiety
3	a83504bb33	anxiety
4	fd98d9d3c0	anxiety
5	4cf316ef9d	anxiety
6	a83504bb33	anxiety
7	206462a8e5	anxiety
8	0ee52e54e3	anxiety
9	3f84996c42	anxiety
10	e601d24a70	depression
11	a6b8721a38	depression
12	a6b8721a38	depression
13	a6b8721a38	depression
14	a6b8721a38	depression
15	278c857a2a	depression
16	da3df1f5f7	depression
17	c99636232e	depression

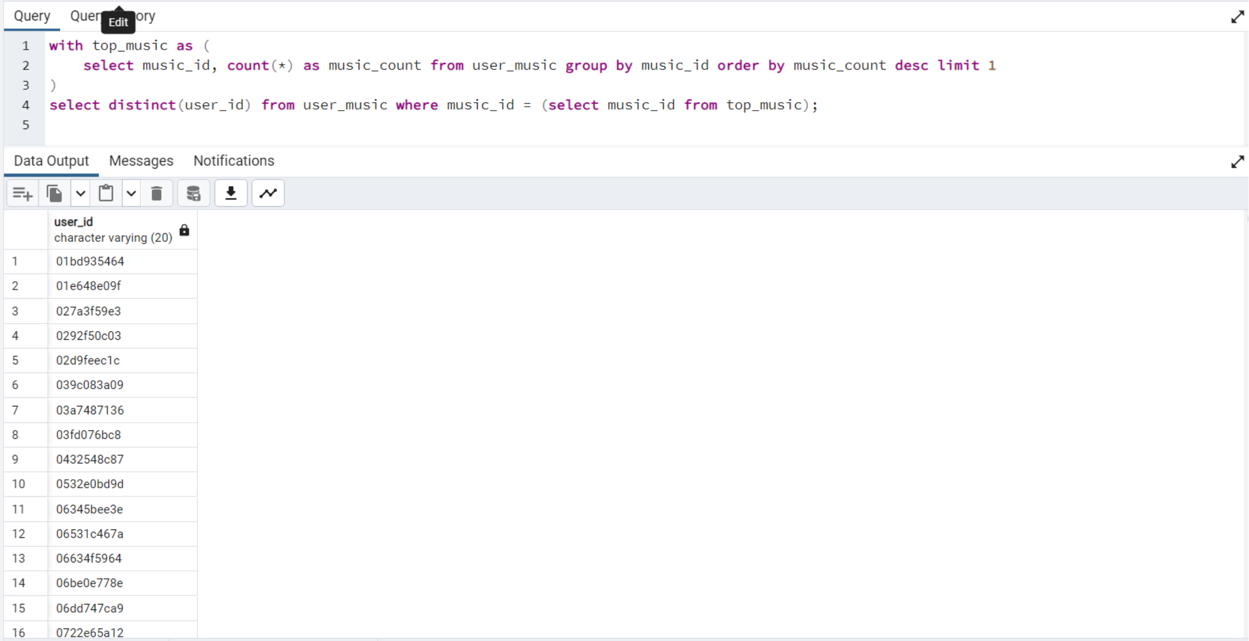
# Query to select all the users which listen to the most listened song in the dataset

with top_music as (

select music_id, count(*) as music_count from user_music group by music_id order by music_count desc limit 1

)

select distinct(user_id) from user_music where music_id = (select music_id from top_music);



```
with top_music as (
    select music_id, count(*) as music_count from user_music group by music_id order by music_count desc limit 1
)
select distinct(user_id) from user_music where music_id = (select music_id from top_music);

```

	user_id character varying (20)
1	01bd935464
2	01e648e09f
3	027a3f59e3
4	0292f50c03
5	02d9feec1c
6	039c083a09
7	03a7487136
8	03fd076bc8
9	0432548c87
10	0532e0bd9d
11	06345bee3e
12	06531c467a
13	06634f5964
14	06be0e778e
15	06dd747ca9
16	0722e65a12