EXPLICATIONS DE LA DONNEE

**Aperçu de la santé mentale :** L’ensemble de données fournit des informations précieuses sur la santé mentale en analysant les modèles linguistiques, les sentiments et les indicateurs psychologiques dans les données textuelles. Les chercheurs et les scientifiques des données peuvent mieux comprendre comment les problèmes de santé mentale se manifestent dans la communication en ligne.

C’est pour l’analyse des sentiments et aux indicateurs psychologiques, probablement dérivées de publications ou de données textuelles.

Cette donnée est copiée suivant le lien : kaggle datasets download -d bhavikjikadara/mental-health-dataset

Nous avons comme variables :

* Sexe
* Pays du répondant
* Occupations
* Être travailleur autonome ou pas
* Avoir des antécédents familiaux de maladie mentale
* Avoir cherché un traitement pour un problème de santé mentale
* Et d’autres variables

QUELQUES PISTES D’ANALYSE DE LA DONNÉE

Grâce à un large éventail de fonctionnalités, notamment des scores d’analyse des sentiments et des indicateurs psychologiques, l’ensemble de données offre la possibilité de développer des modèles prédictifs pour identifier ou prédire les résultats en matière de santé mentale à partir de données textuelles. Cela peut être utile pour une intervention et un soutien précoce. En analysant l’ensemble de données, nous pouvons découvrir des corrélations et des modèles entre les caractéristiques linguistiques, le sentiment et les indicateurs de santé mentale. Ces connaissances peuvent éclairer les interventions, les politiques et les systèmes de soutien visant à promouvoir le bien-être mental.

Nous allons essayer ici de décrire quelques variables qui sont:

* Le sexe, pays du répondant, travailleur autonome ou pas, antécédant familiaux, avoir chercher un traitement.

Nous allons aussi essayer de voir le lien entre le sexe et les autres variables

LES EXPLICATIONS ET LES COMMANDES UTILISÉES

Mettons en place un ETL en python qui permettre de charger les données dans une base de données MySQL. (Commentons et expliquons bien le code)

import pandas as pd

import mysql.connector

# Fonction pour extraire les données

def extract\_data(file\_path):

"""

Cette fonction prend en entrée le chemin vers un fichier CSV

et renvoie un DataFrame pandas contenant les données extraites.

"""

data = pd.read\_csv(file\_path)

return data

# Fonction pour transformer les données

def transform\_data(data):

"""

Cette fonction effectue toute transformation nécessaire sur les données

avant de les charger dans la base de données MySQL.

Dans cet exemple, aucune transformation n'est nécessaire.

"""

return data

# Fonction pour charger les données dans la base de données MySQL

def load\_data\_to\_mysql(data, table\_name, host, user, password, database):

"""

Cette fonction charge les données du DataFrame pandas dans une table MySQL.

"""

try:

# Connexion à la base de données MySQL

connection = mysql.connector.connect(

host=host,

user=user,

password=password,

database=database

)

# Création du curseur pour exécuter les requêtes SQL

cursor = connection.cursor()

# Insertion des données dans la table MySQL

data.to\_sql(name=table\_name, con=connection, if\_exists='replace', index=False)

# Validation de la transaction et fermeture de la connexion

connection.commit()

print("Les données ont été chargées avec succès dans la base de données MySQL.")

except Exception as e:

print("Erreur lors du chargement des données dans MySQL :", str(e))

finally:

if connection.is\_connected():

cursor.close()

connection.close()

# Chemin vers le fichier CSV contenant les données à charger

file\_path = "donnees.csv"

# Nom de la table MySQL où les données seront chargées

table\_name = "ma\_table"

# Informations de connexion à la base de données MySQL

host = "localhost"

user = "root"

password = "

database = "Mental\_Health"

# Extraction des données

data = extract\_data(file\_path)

# Transformation des données si nécessaire

data = transform\_data(data)

# Chargement des données dans MySQL

load\_data\_to\_mysql(data, table\_name, host, user, password, database)

Bien sûr, voici un exemple de code Python pour mettre en place un ETL (Extract, Transform, Load) qui permet de charger des données dans une base de données MySQL. Ce code utilise la bibliothèque pandas pour l'extraction et la transformation des données, ainsi que la bibliothèque mysql-connector-python pour la connexion à la base de données MySQL. Assurez-vous d'avoir installé ces bibliothèques au préalable via pip (`pip install pandas mysql-connector-python`).

Nous avons essentiellement trois fonctions qui sont principales :

1. `extract\_data`: Cette fonction prend en entrée le chemin vers un fichier CSV et renvoie un DataFrame pandas contenant les données extraites.

2. `transform\_data`: Cette fonction peut être utilisée pour effectuer des transformations sur les données si nécessaire. Dans cet exemple, aucune transformation n'est effectuée.

3. `load\_data\_to\_mysql`: Cette fonction charge les données du DataFrame pandas dans une table MySQL en utilisant la bibliothèque mysql-connector-python. Les paramètres de connexion (hôte, utilisateur, mot de passe, base de données) ainsi que le nom de la table sont fournis en entrée.