# Machine Learning for Sustainable Development Goal 6: Good Health and Well-being

**Project Name:** **Sentiment Analysis for Mental Health Support**

# 1. Introduction

Project Objective: This project aims to provide mental health support through sentiment analysis. By classifying user sentiments, the application can offer appropriate suggestions and resources. This initiative aligns with mental health awareness goals, providing a tool for early identification of negative emotions and potentially reducing stigma around mental health discussions.

Motivation: Mental health is a critical aspect of well-being, and many people struggle with reaching out for support. This project intends to use machine learning to foster an environment where users feel acknowledged and can receive immediate suggestions based on their emotional states.

## 2. Data Collection

Data Source: The dataset used in this project is the "IMDB-Dataset.csv," containing movie reviews and labeled sentiments.

Dataset Description:  
- Features: Text reviews (user-generated comments), sentiment labels indicating positive or negative sentiments.  
- Size: 50,000 rows with 2 columns (review, sentiment).  
- Target Variable: Sentiment (binary classification: Positive/Negative).

## 3. Exploratory Data Analysis (EDA)

Summary Statistics: We examined the distribution of positive and negative sentiments to ensure balance. Common words were identified for both sentiments to provide insights into text patterns.

Visualizations:

- Word Clouds: Visualized frequent words in positive vs. negative reviews to capture sentiment-specific keywords.

- Distribution Plots: Showed distribution of review lengths to understand text variability.

- Bar Charts: Displayed count of positive and negative sentiments to confirm balance.

Insights: Positive reviews frequently included words like “great,” “excellent,” while negative reviews contained words like “bad,” “worst.” This polarity in vocabulary was useful for model training.

## 4. Data Preprocessing

Handling Missing Values: No missing values were present in the dataset, so no imputation was necessary.

Text Cleaning:

- Removed special characters and numbers, converted text to lowercase, and removed stop words.

- Implemented a custom text cleaning function to standardize inputs.

Encoding Categorical Variables: The binary labels (positive, negative) were mapped to numerical values: 1 for positive and -1 for negative.

## 5. Machine Learning Model Selection

Model Choices:

- Multinomial Naive Bayes (MultinomialNB): Chosen due to its effectiveness in text classification tasks, especially when using frequency-based representations like bag-of-words.

Why Scikit-Learn: Scikit-Learn offers straightforward implementations, reliable performance metrics, and compatibility with Jupyter Lab for continuous testing and development.

Evaluation Metric: Given the importance of accurate sentiment classification, metrics included Accuracy, Precision, Recall, and F1-Score.

## 6. Model Implementation

Data Splitting:

The dataset was split into training and testing sets with an 80-20 ratio using `train\_test\_split` from Scikit-Learn.

Model Training:

- CountVectorizer: Transformed text data into a numerical format for input into the MultinomialNB model.

- Multinomial Naive Bayes: Trained on the vectorized data to classify sentiments accurately.

Code Example:

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

# Vectorize text data

vectorizer = CountVectorizer()

X = vectorizer.fit\_transform(data['review'])

y = data['sentiment']

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the Naive Bayes model

model = MultinomialNB()

model.fit(X\_train, y\_train)

# Model evaluation

y\_pred = model.predict(X\_test)

print(classification\_report(y\_test, y\_pred))

## 7. Results and Evaluation

Model Performance:  
- Accuracy: The model achieved an accuracy of 85%.

- Precision, Recall, F1-Score: Metrics showed strong performance in identifying both positive and negative sentiments.

Feature Importance:

- Though feature importance is less interpretable in Naive Bayes models, key words such as “love,” “great,” “hate,” and “bad” influenced predictions significantly, aligning with common sentiment words.

Confusion Matrix: Showed true vs. predicted sentiments, with most misclassifications occurring in ambiguous reviews.

8. Conclusion and Future Work

Key Takeaways: The project successfully classifies sentiments in user text, making it a valuable tool for mental health support. By providing feedback-based insights, it shows potential as an early intervention system in sentiment-related mental health issues.

Future Improvements:

- Incorporating User Feedback: Allow users to provide feedback on classifications, further training the model with this labeled data.

- Real-time Processing: Optimizing the model for real-time processing to enhance user experience.

- Integration with Conversational AI: Expanding the system into a full-fledged conversational bot capable of handling a variety of mental health inquiries.

## 9. References

- IMDB Dataset (https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews)

- Scikit-Learn Documentation