**Project Idea: SDG 16: Peace, Justice, and Strong Institutions**

**Goal:** Hugging Face's pre-trained models will be used to summarize and analyze pertinent reports and papers in order to improve understanding and engagement with problems pertaining to peace, justice, and strong institutions.

**Project Details:**

**Goal:** Put NLP models to work summarizing and categorizing writings about justice, peace, and robust institutions. The project's goal is to improve information accessibility and usefulness for the general public and policymakers.

**Outcome:** To assist users in comprehending important information about SDG 16, create summaries of lengthy reports and group text into pertinent themes.

**Research and Setup**

**Overview:** The Hugging Face Transformers library provides a range of pre-trained models for various NLP tasks such as text classification, summarization, translation, and more.

**Capabilities:** These models can handle complex tasks like sentiment analysis, named entity recognition, text summarization, and more with state-of-the-art performance. Explanation:

**Model Selection:**

**Model Hub:** Browse the Hugging Face Model Hub to find pre-trained models suited for summarization and text classification.

**Selected Models:**

* **For Summarization:** facebook/bart-large-cnn or t5-base
* **For Classification:** bert-base-uncased or roberta-base

**Part 2: Implementation**

**I. Loading the Model:**

Load the chosen pre-trained models using the Hugging Face Transformers library.

from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Load the tokenizer and model for classification

tokenizer = AutoTokenizer.from\_pretrained('bert-base-uncased')

model = AutoModelForSequenceClassification.from\_pretrained('bert-base-uncased')

**Data Preparation:**

**1: Dataset Selection:** I selected the “Global Peace Index 2023” to work with in this assignment.

**2: Loading the dataset**

import pandas as pd

# Load your dataset

df = pd.read\_csv(‘GPI-2023.csv’)

**3: Perform EDA**

**Explore the Data:** Understand the structure and content of your dataset. Check for missing values, data distribution, and sample content.

print(df.head())

print(df.describe()

**4: Testing the model before fine-tuning**

# Example text for classification (select a sample)

sample\_text = df.loc[0, 'Country']

# Classify the sample text

result = classifier(sample\_text)

# Print the classification result

print("Sample Text:", sample\_text)

print("Classification Result:", result)

**5: Preprocess and Fine Tuning the model.**

from transformers import TrainingArguments

training\_args = TrainingArguments(

output\_dir='./results',

evaluation\_strategy="epoch",

per\_device\_train\_batch\_size=8,

per\_device\_eval\_batch\_size=8,

num\_train\_epochs=3,

weight\_decay=0.01,

logging\_dir='./logs', )

**Evaluation**

eval\_results = trainer.evaluate()

print(eval\_results)