HACKATHON LEVEL-2

GEN-AI

Githublink : https://github.com/Sasiprakash-maker/GenAI-Hackathon.git

Name : SASI PRAKASH.L

Reg No : 628122114052

College Name: AVS ENGINEERING COLLEGE

Department : Mechanical engineering

Step 1: Import Libraries

Start by importing essential libraries for data handling, preprocessing, modeling, and evaluation. Commonly used libraries include:

Python Code

```
import pandas as pd and manipulation


# for data loading

import numpy as np operations

# for numerical

import matplotlib.pyplot as plt # for

visualizations

import seaborn as sns

# for enhanced

visualizations

from sklearn .model_selection import

train_test_split # for data splitting from sklearn.metrics import accuracy_score, f1_score # for
evaluation metrics


# Importing model-related libraries, e.g.. transformers for NLP tasks


from transformers import AutoTokenizer,
AutoNodelForSequenceClassification, Trainer TrainingArguments
```

Step 2:Load data

Load your dataset, whether in CSV, JSON, or text format,

using pandas.

```
# Example for loading a CSV file data = pd.read_ csv(path_ to_your_file. csv')
```

Inspect the data to understand its structure.

```
data.head() data.info()
```

Step3:Data processing

Perform necessary preprocessing steps:

• For text data: tokenization, removing special characters,etc.

For numerical data: normalization or standardization if needed.

Split the data into training and test sets:

Python Code

```
# Splitting data (80-20 split, can be adjusted) train_data, test_data = train_test_split(data,
test_size-0.2, random_state-42)
# Example: Tokenize text data if using a text -based model
# tokenizer = AutoTokenizer.from_pretrained ('model_name')
# train_data_encodings = tokenizer(1ist
(train_data['text_column']), truncation =True, padding=True)
# test_data_encodings = tokenizer (list (test_data['text_column'), truncation=True
.padding-True)
```

Step 4: Choose a model

Either load a pre-trained model (for NLP or similar tasks) or define a customn model if working with a neural network approach.

Python Code:

```
# Load a pre-trained model (for example, a sentiment analysis model) model = AutoModelFor
SequenceClassification.from_pretrained('model_name)
```

Step5: Train the model

Set up training arguments, and train the model with the training dataset.

Python Code

```
# Example: Define training arguments and start training training_args = TrainingArguments(
output_dir=/results

num_train_epochs=3,

per_device_train_batch_size=8

per_device_eval_batch_size-8,

evaluation_strategy='epoch")

trainer = Trainer( model=model, args=training_args, train_dataset=train_data, eval_
dataset=test_data

trainer. train()
```

Step6: Evaulate the model

Use the test dataset to evaluate your model. Calculate metrics such as accuracy and F1-score, then display the results

Python Code

```
# Make predictions and evaluate predictions = trainer.predict(test_data) accuracy =
accuracy_score (test_data['labels'), predictions)

f1 =f1_score (test_datal"labels'l, predictions, average-'weighted')

print(f'Accuracy: laccuracyl. F1-score: If1}
```