*Nlp project *

```
!ls
```

```
sample_data
```

STEP 1: Install dependencies

```
!pip install transformers datasets torch
```

```
                              664.8/664.8 MB 2.7 MB/s eta 0
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl
                              211.5/211.5 MB 6.4 MB/s eta 0
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.wl
                              56.3/56.3 MB 12.6 MB/s eta 0:
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.wl
                              127.9/127.9 MB 8.7 MB/s eta 0
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64
                              207.5/207.5 MB 5.7 MB/s eta 0
Downloading nvidia_nccl_cu12-2.21.5-py3-none-manylinux2014_x86_64.whl (18
                              188.7/188.7 MB 6.7 MB/s eta 0
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.
                              21.1/21.1 MB 65.6 MB/s eta 0:
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-nccl-cu12, n
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-nccl-cu12
    Found existing installation: nvidia-nccl-cu12 2.23.4
    Uninstalling nvidia-nccl-cu12-2.23.4:
      Successfully uninstalled nvidia-nccl-cu12-2.23.4
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
```

```
      Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-cupti-cu12
      Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
      Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
    Attempting uninstall: nvidia-cublas-cu12
      Found existing installation: nvidia-cublas-cu12 12.5.3.2
      Uninstalling nvidia-cublas-cu12-12.5.3.2:
        Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
    Attempting uninstall: nvidia-cusparse-cu12
      Found existing installation: nvidia-cusparse-cu12 12.5.1.3
      Uninstalling nvidia-cusparse-cu12-12.5.1.3:
        Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
    Attempting uninstall: nvidia-cudnn-cu12
      Found existing installation: nvidia-cudnn-cu12 9.3.0.75
      Uninstalling nvidia-cudnn-cu12-9.3.0.75:
        Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
    Attempting uninstall: nvidia-cusolver-cu12
      Found existing installation: nvidia-cusolver-cu12 11.6.3.83
      Uninstalling nvidia-cusolver-cu12-11.6.3.83:
        Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
  Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-
```

STEP 2: Import libraries

```
import pandas as pd
import re
from datasets import Dataset
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trai
```

Importing and reading dataset

```
df = pd.read_csv("/content/labeled_data.csv")

# Show dataset structure
print(df.head())
print(df['class'].value_counts())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
class
1    19190
2     4163
0     1430
Name: count, dtype: int64
```

STEP 4: Clean text

```
def clean_text(text):
    text = text.lower()
    text = re.sub(r"http\S+", "", text)       # remove urls
    text = re.sub(r"@\w+", "", text)          # remove mentions
    text = re.sub(r"#\w+", "", text)          # remove hashtags
    text = re.sub(r"[^a-z\s]", "", text)      # remove punctuation/numbers
    return text

df['clean_text'] = df['tweet'].apply(clean_text)
```

LABEL

```
from datasets import ClassLabel

# Create ClassLabel feature (3 classes)
class_label = ClassLabel(num_classes=3, names=["Hate Speech", "Offensive", "Nei

# Convert labels to ClassLabel
df["labels"] = df["class"].astype(int)

dataset = Dataset.from_pandas(df)
dataset = dataset.cast_column("labels", class_label)

# Now stratified split works
dataset = dataset.train_test_split(test_size=0.2, stratify_by_column="labels")
```

```
Casting the dataset: 100% 24783/24783 [00:00<00:00, 230880.40 examples/
```

STEP 6: Tokenization (RoBERTa)

```
model_name = "roberta-base"
tokenizer = AutoTokenizer.from_pretrained(model_name)

def tokenize(batch):
    return tokenizer(batch["clean_text"], padding="max_length", truncation=True

dataset = dataset.map(tokenize, batched=True)

# Format for PyTorch
dataset.set_format("torch", columns=["input_ids", "attention_mask", "labels"])
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
  warnings.warn(
tokenizer_config.json: 100% 25.0/25.0 [00:00<00:00, 2.73kB/s]
config.json: 100% 481/481 [00:00<00:00, 38.1kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 2.12MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 2.15MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 1.55MB/s]
Map: 100% 19826/19826 [00:03<00:00, 5583.93 examples/s]
Map: 100% 4957/4957 [00:00<00:00, 5937.00 examples/
```

STEP 7: Load RoBERTa model

```
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labe
```

```
model.safetensors: 100% 499M/499M [00:08<00:00, 85.7MB/s]
Some weights of RobertaForSequenceClassification were not initialized from
You should probably TRAIN this model on a down-stream task to be able to us
```

```
!pip install -U transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/di
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/l
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dis
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/pyt
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.1
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/pytho
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/p
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/di
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3
```

STEP 8: Training setup

```
from transformers import TrainingArguments

training_args = TrainingArguments(
    output_dir="./results",
    do_eval=True,                # enables evaluation
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    learning_rate=2e-5,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=50,
    save_total_limit=2
)
```

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be r
```

STEP 9: Define Trainer

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=dataset["train"],
    eval_dataset=dataset["test"],
    tokenizer=tokenizer
)
```

```
/tmp/ipython-input-1688317445.py:1: FutureWarning: `tokenizer` is deprecate
  trainer = Trainer(
```

```
import os
os.environ["WANDB_DISABLED"] = "true"
```

```
trainer.train()
```

[3720/3720 22:57, Epoch 3/3]

Step	Training Loss
50	0.342100
100	0.386900
150	0.364500
200	0.392200
250	0.351700
300	0.349300
350	0.261900
400	0.266100
450	0.325700
500	0.255400
550	0.318300
600	0.280800
650	0.274600

700	0.228400
750	0.321600
800	0.266900
850	0.259700
900	0.269200
950	0.270100
1000	0.222800
1050	0.295200
1100	0.286600
1150	0.303300
1200	0.261000
1250	0.303900
1300	0.222000
1350	0.277000
1400	0.258700
1450	0.198100
1500	0.191500
1550	0.251900
1600	0.250000
1650	0.231300
1700	0.274600
1750	0.233200
1800	0.200300
1850	0.236800
1900	0.271000
1950	0.228200
2000	0.270700
2050	0.282600
2100	0.185300

2150	0.263500
2200	0.248200
2250	0.315700
2300	0.232200
2350	0.226700
2400	0.241100
2450	0.243100
2500	0.232400
2550	0.179600
2600	0.203700
2650	0.200600
2700	0.213500
2750	0.208200
2800	0.174300
2850	0.205900
2900	0.263900
2950	0.233100
3000	0.224900
3050	0.247700
3100	0.204500
3150	0.216400
3200	0.179300
3250	0.208100
3300	0.166100
3350	0.207300
3400	0.200200
3450	0.191100
3500	0.223000
3550	0.197700
3600	0.200600

```
3650        0.199300
3700        0.179900
```

```
TrainOutput(global_step=3720, training_loss=0.24942343247834073, metrics=
{'train_runtime': 1377.5828, 'train_samples_per_second': 43.176,
'train_steps_per_second': 2.7, 'total_flos': 3912364964906496.0
```

```python
from transformers import RobertaForSequenceClassification, RobertaTokenizer
import shutil
from google.colab import files

save_folder = "roberta_base"
model.save_pretrained(save_folder)
tokenizer.save_pretrained(save_folder)

# Zip and download to your computer
shutil.make_archive(save_folder, 'zip', save_folder)
files.download(save_folder + ".zip")
```

```python
from transformers import RobertaForSequenceClassification, RobertaTokenizer
import torch

# Load your trained model
model_name = "roberta_base"  # or "yash_roberta_model"
model = RobertaForSequenceClassification.from_pretrained(model_name)
tokenizer = RobertaTokenizer.from_pretrained(model_name)

# Set model to evaluation mode
model.eval()
```

```
RobertaForSequenceClassification(
  (roberta): RobertaModel(
    (embeddings): RobertaEmbeddings(
      (word_embeddings): Embedding(50265, 768, padding_idx=1)
      (position_embeddings): Embedding(514, 768, padding_idx=1)
      (token_type_embeddings): Embedding(1, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): RobertaEncoder(
      (layer): ModuleList(
        (0-11): 12 x RobertaLayer(
          (attention): RobertaAttention(
            (self): RobertaSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768,
bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768,
bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): RobertaSelfOutput(
              (dense): Linear(in_features=768, out_features=768,
bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-05,
elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): RobertaIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): RobertaOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-05,
elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
  )
  (classifier): RobertaClassificationHead(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (dropout): Dropout(p=0.1, inplace=False)
    (out_proj): Linear(in_features=768, out_features=3, bias=True)
  )
)
```

```
texts = [
    "fuck me !",
    "This is amazing!",
    "Just an ordinary day."
]
```

```
inputs = tokenizer(texts, padding=True, truncation=True, return_tensors="pt")
```

```
with torch.no_grad():
    outputs = model(**inputs)
    logits = outputs.logits
    predictions = torch.argmax(logits, dim=-1)
```

```
label_names = ["Hate Speech", "Offensive", "Neither"]
decoded_preds = [label_names[p.item()] for p in predictions]
print(decoded_preds)
```

```
['Offensive', 'Neither', 'Neither']
```

```
!pip install gradio

import gradio as gr
from transformers import RobertaForSequenceClassification, RobertaTokenizer
import torch

# Load your model
model = RobertaForSequenceClassification.from_pretrained("roberta_base")
tokenizer = RobertaTokenizer.from_pretrained("roberta_base")
model.eval()

label_names = ["Hate Speech", "Offensive", "Neither"]

# Define prediction function
def predict(text):
    inputs = tokenizer(text, padding=True, truncation=True, return_tensors="pt'
    with torch.no_grad():
        logits = model(**inputs).logits
        pred = torch.argmax(logits, dim=-1).item()
    return label_names[pred]

# Create Gradio interface
```

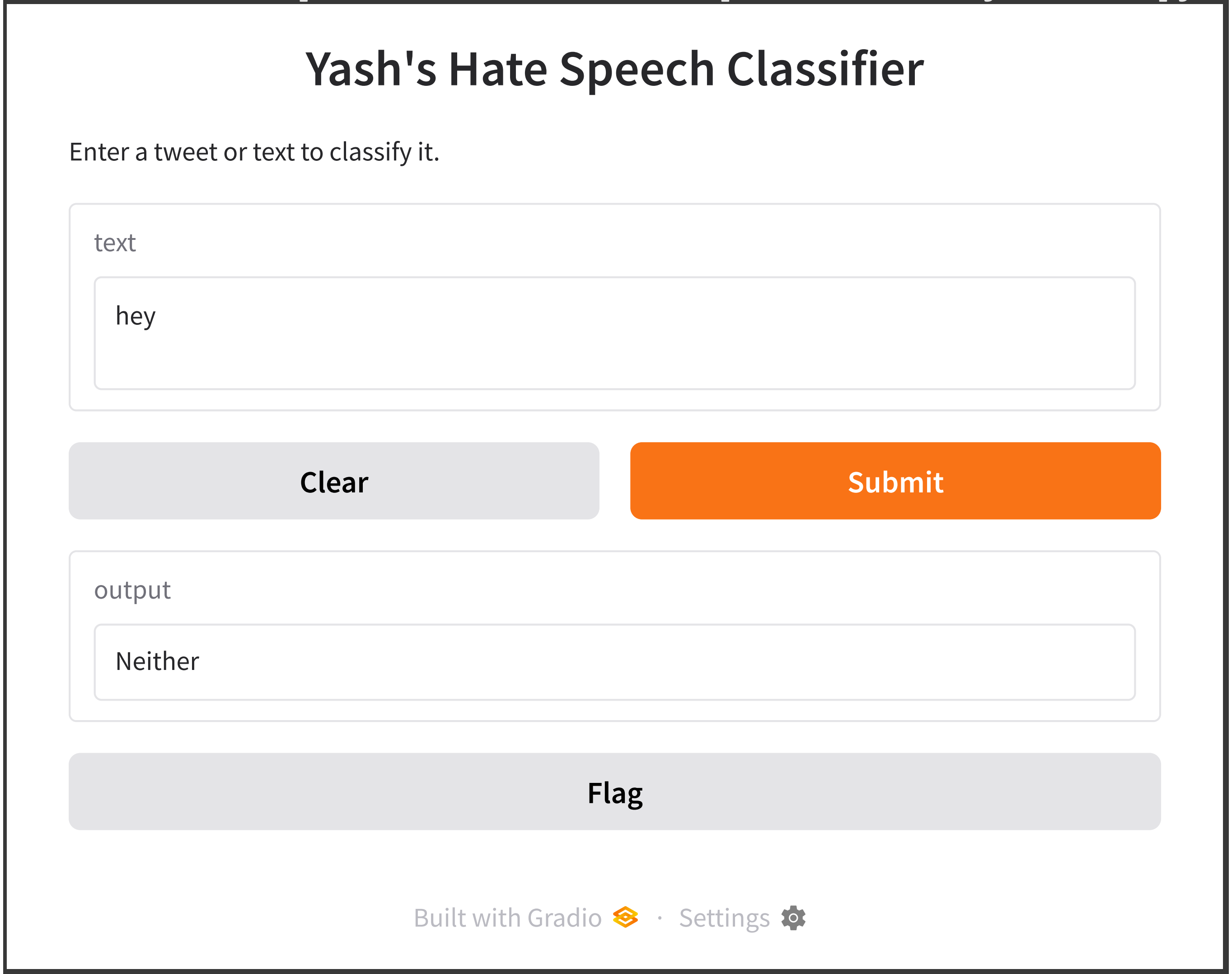
```
iface = gr.Interface(fn=predict,
                     inputs=gr.Textbox(lines=2, placeholder="Type your text her
                     outputs="text",
                     title="Yash's Hate Speech Classifier",
                     description="Enter a tweet or text to classify it.")

# Launch the GUI
iface.launch()
```

```
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-pacl
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/pythor
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/
Requirement already satisfied: brotli>=1.1.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/pytho
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: gradio-client==1.11.1 in /usr/local/lib/pytho
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: httpx<1.0,>=0.24.1 in /usr/local/lib/python3
Requirement already satisfied: huggingface-hub<1.0,>=0.33.5 in /usr/local/li
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/pythor
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.1
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/py
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/pyt
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/pytho
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/pyth
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/pyt
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/pyth
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-pacl
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/pyth
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/di
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/pythor
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyth
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/c
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/pyt
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/pyth
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/p
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/pyth
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/py
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/p
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist
It looks like you are running Gradio on a hosted Jupyter notebook, which re

Colab notebook detected. To show errors in colab notebook, set debug=True i
* Running on public URL: https://958afe1b95e267584f.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgra
```



Yash's Hate Speech Classifier

Enter a tweet or text to classify it.

text

hey

output

Neither

Start coding or generate with AI.