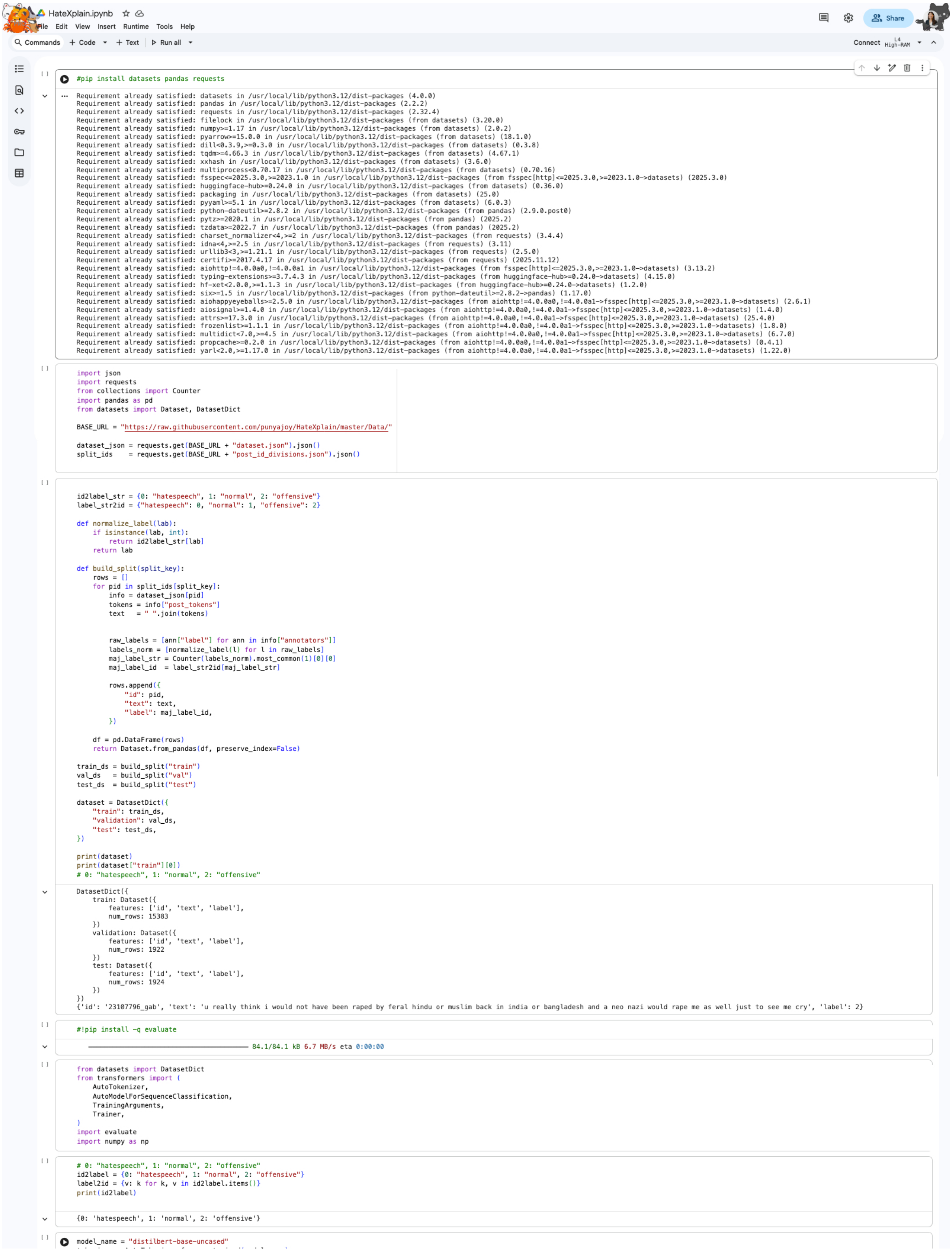


```
#pip install datasets pandas requests
```

```
Requirement already satisfied: datasets in /usr/local/lib/python3.12/dist-packages (4.0.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (2.32.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from datasets) (3.20.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from datasets) (2.0.2)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (18.1.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.12/dist-packages (from datasets) (4.67.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.12/dist-packages (from datasets) (3.6.0)
Requirement already satisfied: multiprocess<0.70.17 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2025.3.0,>=2023.1.0 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2025.3.0)
Requirement already satisfied: huggingface-hub>=0.24.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.36.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from datasets) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from datasets) (6.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests) (2025.11.12)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (3.13.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.24.0->datasets) (4.15.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.24.0->datasets) (1.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (25.4.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.8.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (6.7.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (0.4.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.22.0)
```

```
import json
import requests
from collections import Counter
import pandas as pd
from datasets import Dataset, DatasetDict

BASE_URL = "https://raw.githubusercontent.com/punyajoy/HateXplain/master/Data/"

dataset_json = requests.get(BASE_URL + "dataset.json").json()
split_ids    = requests.get(BASE_URL + "post_id_divisions.json").json()
```

```
id2label_str = {0: "hatespeech", 1: "normal", 2: "offensive"}
label_str2id = {"hatespeech": 0, "normal": 1, "offensive": 2}

def normalize_label(lab):
    if isinstance(lab, int):
        return id2label_str[lab]
    return lab

def build_split(split_key):
    rows = []
    for pid in split_ids[split_key]:
        info = dataset_json[pid]
        tokens = info["post_tokens"]
        text   = " ".join(tokens)


        raw_labels = [ann["label"] for ann in info["annotators"]]
        labels_norm = [normalize_label(l) for l in raw_labels]
        maj_label_str = Counter(labels_norm).most_common(1)[0][0]
        maj_label_id  = label_str2id[maj_label_str]

        rows.append({
            "id": pid,
            "text": text,
            "label": maj_label_id,
        })

    df = pd.DataFrame(rows)
    return Dataset.from_pandas(df, preserve_index=False)

train_ds = build_split("train")
val_ds   = build_split("val")
test_ds  = build_split("test")

dataset = DatasetDict({
    "train": train_ds,
    "validation": val_ds,
    "test": test_ds,
})

print(dataset)
print(dataset["train"][0])
# 0: "hatespeech", 1: "normal", 2: "offensive"
```

```
DatasetDict({
    train: Dataset({
        features: ['id', 'text', 'label'],
        num_rows: 15383
    })
    validation: Dataset({
        features: ['id', 'text', 'label'],
        num_rows: 1922
    })
    test: Dataset({
        features: ['id', 'text', 'label'],
        num_rows: 1924
    })
})
{'id': '23107796_gab', 'text': 'u really think i would not have been raped by feral hindu or muslim back in india or bangladesh and a neo nazi would rape me as well just to see me cry', 'label': 2}
```

```
#!pip install -q evaluate
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.1/84.1 kB 6.7 MB/s eta 0:00:00
```

```
from datasets import DatasetDict
from transformers import (
    AutoTokenizer,
    AutoModelForSequenceClassification,
    TrainingArguments,
    Trainer,
)
import evaluate
import numpy as np
```

```
# 0: "hatespeech", 1: "normal", 2: "offensive"
id2label = {0: "hatespeech", 1: "normal", 2: "offensive"}
label2id = {v: k for k, v in id2label.items()}
print(id2label)
```

```
{0: 'hatespeech', 1: 'normal', 2: 'offensive'}
```

```
model_name = "distilbert-base-uncased"
```

```
tokenizer = AutoTokenizer.from_pretrained(model_name)

def tokenize_batch(batch):
    return tokenizer(
        batch["text"],
        padding="max_length",
        truncation=True,
        max_length=256,
    )

tokenized_dataset = dataset.map(tokenize_batch, batched=True)

cols_to_remove = [c for c in tokenized_dataset["train"].column_names
                  if c not in ["input_ids", "attention_mask", "label"]]

tokenized_dataset = tokenized_dataset.remove_columns(cols_to_remove)

tokenized_dataset.set_format("torch")

tokenized_dataset
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 6.20kB/s]
config.json: 100%  483/483 [00:00<00:00, 60.6kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 526kB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 1.09MB/s]
Map: 100%  15383/15383 [00:01<00:00, 8689.33 examples/s]
Map: 100%  1922/1922 [00:00<00:00, 7872.31 examples/s]
Map: 100%  1924/1924 [00:00<00:00, 7374.84 examples/s]
DatasetDict({
    train: Dataset({
        features: ['label', 'input_ids', 'attention_mask'],
        num_rows: 15383
    })
    validation: Dataset({
        features: ['label', 'input_ids', 'attention_mask'],
        num_rows: 1922
    })
    test: Dataset({
        features: ['label', 'input_ids', 'attention_mask'],
        num_rows: 1924
    })
})
```

```
model = AutoModelForSequenceClassification.from_pretrained(
    model_name,
    num_labels=3,
    id2label=id2label,
    label2id=label2id,
)
```

```
model.safetensors: 100%  268M/268M [00:02<00:00, 89.5MB/s]
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncased and are newly initialized: ['classifier.bias', 'classifier.weight', 'pre_classifier.bias
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    preds = np.argmax(logits, axis=-1)

    acc  = accuracy_score(labels, preds)
    f1   = f1_score(labels, preds, average="macro")
    prec = precision_score(labels, preds, average="macro")
    rec  = recall_score(labels, preds, average="macro")

    return {
        "accuracy": acc,
        "f1_macro": f1,
        "precision_macro": prec,
        "recall_macro": rec,
    }
```

```
training_args = TrainingArguments(
    output_dir="./hatexplain_distilbert",
    num_train_epochs=3,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=32,
    learning_rate=2e-5,
    weight_decay=0.01,
    report_to="none",
    label_smoothing_factor=0.1
)
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["validation"],
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)
```

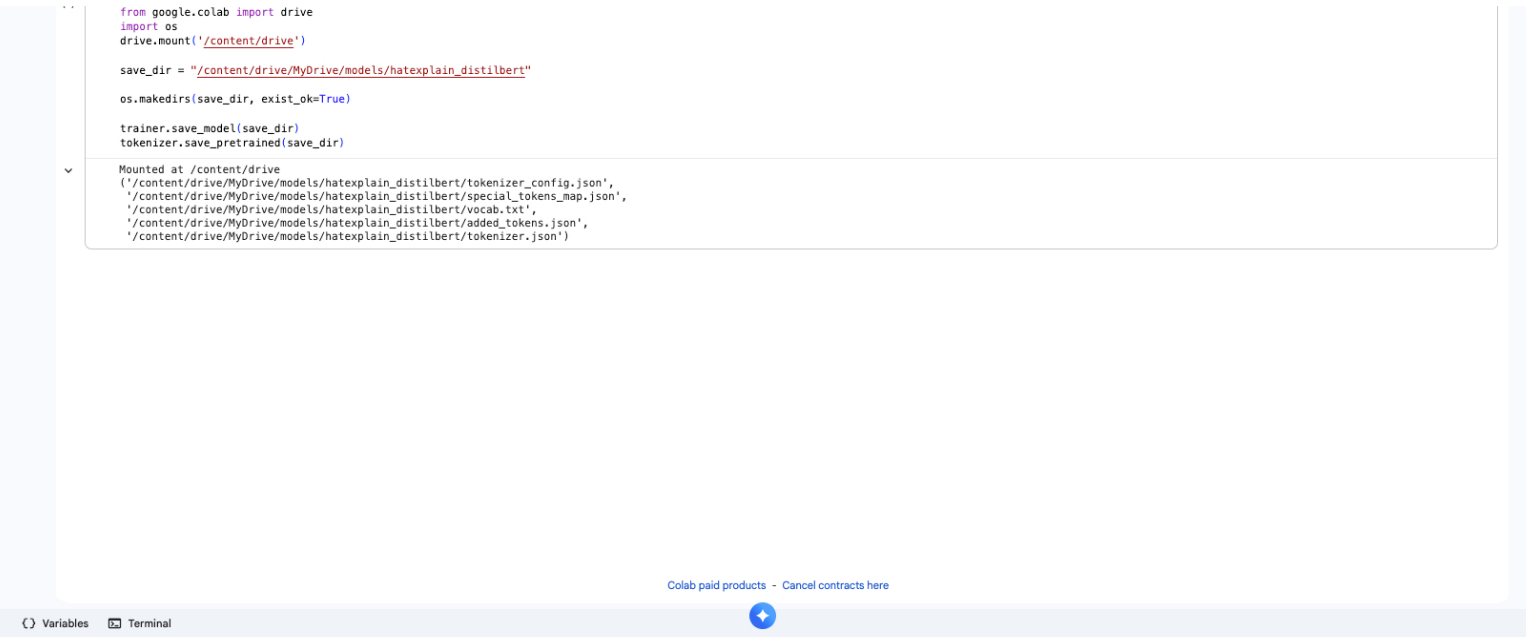
```
/tmp/ipython-input-991116914.py:1: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`. Use `processing_class` instead.
  trainer = Trainer(
```

```
trainer.train()

# validation
eval_results = trainer.evaluate()
print(eval_results)

# test
test_results = trainer.evaluate(tokenized_dataset["test"])
print(test_results)
```

[2886/2886 07:09, Epoch 3/3]

Step	Training Loss
500	0.891300
1000	0.807500
1500	0.738500
2000	0.722100
2500	0.649500

```
[61/61 00:13]
{'eval_loss': 0.8223803043365479, 'eval_accuracy': 0.6945889698231009, 'eval_f1_macro': 0.6818566126545894, 'eval_precision_macro': 0.6831365476743532, 'eval_recall_macro': 0.683054591797116, 'eval_runtime': 6.5496
{'eval_loss': 0.8156391978263855, 'eval_accuracy': 0.6923076923076923, 'eval_f1_macro': 0.6762298186866601, 'eval_precision_macro': 0.6757851449322296, 'eval_recall_macro': 0.6792344994720835, 'eval_runtime': 6.708
```



```
from google.colab import drive
import os
drive.mount('/content/drive')

save_dir = "/content/drive/MyDrive/models/hatexplain_distilbert"

os.makedirs(save_dir, exist_ok=True)

trainer.save_model(save_dir)
tokenizer.save_pretrained(save_dir)
```

```
Mounted at /content/drive
('/content/drive/MyDrive/models/hatexplain_distilbert/tokenizer_config.json',
 '/content/drive/MyDrive/models/hatexplain_distilbert/special_tokens_map.json',
 '/content/drive/MyDrive/models/hatexplain_distilbert/vocab.txt',
 '/content/drive/MyDrive/models/hatexplain_distilbert/added_tokens.json',
 '/content/drive/MyDrive/models/hatexplain_distilbert/tokenizer.json')
```