```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
!pip install --upgrade transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.34.
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transform
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (
```

```
!pip install transformers datasets scikit-learn
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.1)
Requirement already satisfied: datasets in /usr/local/lib/python3.12/dist-packages (4.0.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (1.6.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
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: pandas in /usr/local/lib/python3.12/dist-packages (from datasets) (2.2.2)
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]<=20
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.16.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=202
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transform
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1-
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->fs
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1-
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
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas->dat
```

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

import torch
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_recall_fscore_support, classification_report
from transformers import DistilBertTokenizerFast, DistilBertForSequenceClassification, Trainer, TrainingArguments
```

```
import pandas as pd
```

```
# Adjust paths if your files are named differently or in another folder
fake_df = pd.read_csv("/content/Fake (1).csv")
true_df = pd.read_csv("/content/True.csv")

fake_df["label"] = 0  # Fake
true_df["label"] = 1  # True

df = pd.concat([fake_df, true_df], ignore_index=True)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
df.head()  # just to preview
```

	title	text	subject	date	label
0	Ben Stein Calls Out 9th Circuit Court: Committ...	21st Century Wire says Ben Stein, reputable pr...	US_News	February 13, 2017	0
1	Trump drops Steve Bannon from National Securit...	WASHINGTON (Reuters) - U.S. President Donald T...	politicsNews	April 5, 2017	1
2	Puerto Rico expects U.S. to lift Jones Act shi...	(Reuters) - Puerto Rico Governor Ricardo Rosse...	politicsNews	September 27, 2017	1

Next steps: Generate code with df | New interactive sheet

```
train_texts, test_texts, train_labels, test_labels = train_test_split(
    df["text"].tolist(),
    df["label"].tolist(),
    test_size=0.2,
    random_state=42,
    stratify=df["label"]
)

len(train_texts), len(test_texts)
```

```
(35918, 8980)
```

```
tokenizer = DistilBertTokenizerFast.from_pretrained("distilbert-base-uncased")

def encode_data(texts, labels, max_len=256):
    enc = tokenizer(texts, truncation=True, padding=True, max_length=max_len)
    dataset = torch.utils.data.TensorDataset(
        torch.tensor(enc["input_ids"]),
        torch.tensor(enc["attention_mask"]),
        torch.tensor(labels)
    )
    return dataset

train_dataset = encode_data(train_texts, train_labels)
test_dataset  = encode_data(test_texts,  test_labels)
```

```
/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
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, 4.02kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 16.2MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 29.1MB/s]
config.json: 100%  483/483 [00:00<00:00, 48.8kB/s]
```

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = DistilBertForSequenceClassification.from_pretrained(
    "distilbert-base-uncased",
    num_labels=2
).to(device)
```

```
model.safetensors: 100%  268M/268M [00:02<00:00, 165MB/s]
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncase
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
import transformers
print(transformers.__version__)
```

```
4.57.1
```

```
import torch
from torch.utils.data import Dataset

class NewsDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __len__(self):
        return len(self.labels)

    def __getitem__(self, idx):
        item = {k: torch.tensor(v[idx]) for k, v in self.encodings.items()}
        item["labels"] = torch.tensor(int(self.labels[idx]))
        return item

# (re-)tokenize in case you only have texts/labels
train_encodings = tokenizer(train_texts, truncation=True, padding=True, max_length=256)
test_encodings  = tokenizer(test_texts,  truncation=True, padding=True, max_length=256)

train_dataset = NewsDataset(train_encodings, train_labels)
test_dataset  = NewsDataset(test_encodings,  test_labels)
```

```
import transformers
print(transformers.__version__)
```

```
4.57.1
```

```
from transformers import TrainingArguments

training_args = TrainingArguments(
    output_dir="./results",              # save model here
    eval_strategy="steps",         # evaluates every few steps
    save_total_limit=2,                  # keep only last 2 checkpoints
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=1,
    weight_decay=0.01,
    logging_dir="./logs",
    logging_steps=100,
    save_strategy="steps",               # when to save checkpoints
    save_steps=500,
    eval_steps=500,
)
```

```
Using the `WANDB_DISABLED` environment variable is deprecated and will be removed in v5. Use the --report_to flag to control
```

```
from transformers import TrainingArguments
help(TrainingArguments)
```

Show hidden output

```
from transformers import Trainer
import numpy as np
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

def compute_metrics(pred):
    labels = pred.label_ids
    preds = np.argmax(pred.predictions, axis=1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binary')
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }

trainer = Trainer(
    model=model,
```

```
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    compute_metrics=compute_metrics,
)
```

```
import transformers
print(transformers.__version__)
```

```
4.57.1
```

```
#Now train model
trainer.train()
```

[2245/2245 20:21, Epoch 1/1]

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
500	0.002300	0.003947	0.999220	0.999183	0.999300	0.999066
1000	0.000300	0.001361	0.999666	0.999650	1.000000	0.999300
1500	0.000100	0.000092	1.000000	1.000000	1.000000	1.000000
2000	0.000100	0.000150	0.999889	0.999883	0.999767	1.000000

```
TrainOutput(global_step=2245, training_loss=0.010022231180818216, metrics={'train_runtime': 1223.71,
'train_samples_per_second': 29.352, 'train_steps_per_second': 1.835, 'total_flos': 2378982012463104.0, 'train_loss':
```

```
metrics = trainer.evaluate()
print(metrics)
# Detailed classification report
preds = trainer.predict(test_dataset).predictions.argmax(-1)
print(classification_report(test_labels, preds, target_names=["Fake (0)", "True (1)"]))
```

```
{'eval_loss': 0.0008041490218602121, 'eval_accuracy': 0.9996659242761693, 'eval_f1': 0.999649982499125, 'eval_precision': 0.
              precision    recall  f1-score   support

    Fake (0)       1.00      1.00      1.00      4696
    True (1)       1.00      1.00      1.00      4284

    accuracy                           1.00      8980
   macro avg       1.00      1.00      1.00      8980
weighted avg       1.00      1.00      1.00      8980
```

```
def predict_one(text):
    model.eval()
    enc = tokenizer(text, return_tensors="pt", truncation=True, padding=True, max_length=256)
    enc = {k: v.to(device) for k, v in enc.items()}
    with torch.no_grad():
        logits = model(**enc).logits
        pred = int(torch.argmax(logits, dim=-1).cpu().item())
    return "Fake News ❌" if pred == 0 else "True News ✅"
```

```
while True:
    text = input("Enter a news text (or 'exit'): ").strip()
    if text.lower() == "exit":
        break
    print(predict_one(text))
```

```
Enter a news text (or 'exit'): dog has 4 legs
Fake News ❌
Enter a news text (or 'exit'): exit
```

```
{'eval_loss': ..., 'eval_accuracy': ..., 'eval_precision': ..., 'eval_recall': ..., 'eval_f1': ...}
```

```
{'eval_loss': Ellipsis,
 'eval_accuracy': Ellipsis,
 'eval_precision': Ellipsis,
 'eval_recall': Ellipsis,
 'eval_f1': Ellipsis}
```