221501111_221501104_NLP PROJECT MODEL TRAINING

```
!pip install "nvidia-cublas-cu12==12.4.5.8" "nvidia-cuda-runtime-cu12==12.4.127" "nvidia-cudnn-cu12==9.1.0.70" --force-reinstall
```

```
Collecting nvidia-cublas-cu12==12.4.5.8
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl (363.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.1 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 23.2 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.3 MB/s eta 0:00:00
Installing collected packages: nvidia-cuda-runtime-cu12, nvidia-cublas-cu12, nvidia-cudnn-cu12
  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-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-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
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nv
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-runtime-cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70
```

```
!pip install -q datasets
!pip install lime
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.2/491.2 kB 13.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 5.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.9/183.9 kB 12.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 12.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 16.5 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system == "Linux" and platform_machine == "x86_64", but you have n
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nv
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is incompatible.
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 275.7/275.7 kB 7.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from lime) (3.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from lime) (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from lime) (1.14.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from lime) (4.67.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.11/dist-packages (from lime) (1.6.1)
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.11/dist-packages (from lime) (0.25.2)
Requirement already satisfied: networkx>=3.0 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (3.4.2)
Requirement already satisfied: pillow>=10.1 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (11.1.0)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (2.37.0
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (2025.3.30
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (24.2)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.11/dist-packages (from scikit-image>=0.12->lime) (0.4)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=0.18->lime) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=0.18->lime) (3.6.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (1.4.8)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib->lime) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib->lime) (1.17.0
Building wheels for collected packages: lime
  Building wheel for lime (setup.py) ... done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283834 sha256=263403eb25b10bc70bef09efbfb545d7e6484ad9d365fe472897
  Stored in directory: /root/.cache/pip/wheels/85/fa/a3/9c2d44c9f3cd77cf4e533b58900b2bf4487f2a17e8ec212a3d
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
```

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

```python
import pandas as pd
import numpy as np
import torch
from sklearn.model_selection import train_test_split
from transformers import BertTokenizer, BertForSequenceClassification, Trainer, TrainingArguments
from transformers import DataCollatorWithPadding
from datasets import Dataset
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
from lime.lime_text import LimeTextExplainer
import matplotlib.pyplot as plt
```

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

```
Mounted at /content/drive
```

```python
import os

model_dir = '/content/drive/MyDrive/fake_news_models'
os.makedirs(model_dir, exist_ok=True)
```

```python
# Upload your CSVs from Files -> Upload
from google.colab import files
uploaded = files.upload()

# Load datasets
fake_df = pd.read_csv('Fake.csv')
true_df = pd.read_csv('True.csv')

# Add labels: 0 - Fake, 1 - Real
fake_df['label'] = 0
true_df['label'] = 1

# Combine and shuffle
df = pd.concat([fake_df[['text', 'label']], true_df[['text', 'label']]], axis=0)
df = df.sample(frac=1).reset_index(drop=True)

# Train-test split
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```python
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

def tokenize_function(example):
    return tokenizer(example["text"], truncation=True)

train_dataset = Dataset.from_pandas(train_df)
```

```
test_dataset = Dataset.from_pandas(test_df)

train_tokenized = train_dataset.map(tokenize_function, batched=True)
test_tokenized = test_dataset.map(tokenize_function, batched=True)

data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

```
/usr/local/lib/python3.11/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 secre
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, 5.11kB/s]
vocab.txt: 100%                                   232k/232k [00:00<00:00, 1.63MB/s]
tokenizer.json: 100%                                   466k/466k [00:00<00:00, 3.15MB/s]
config.json: 100%                                   570/570 [00:00<00:00, 38.9kB/s]
Map: 100%                                   35918/35918 [04:29<00:00, 133.47 examples/s]
Map: 100%                                   8980/8980 [01:07<00:00, 132.88 examples/s]
```

```
model = BertForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=2)
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better perfo
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to r
model.safetensors: 100%                                   440M/440M [00:05<00:00, 114MB/s]
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are newly initiali
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 precision_recall_fscore_support, accuracy_score
import numpy as np
from transformers import TrainingArguments, Trainer, EarlyStoppingCallback

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

training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,  # you can go up to 10 epochs since early stopping is enabled
    weight_decay=0.01,
    load_best_model_at_end=True,        # ✅ loads best model (based on val metric)
    metric_for_best_model="f1",         # ✅ choose best by f1 score
    greater_is_better=True,             # ✅ maximize F1
    report_to="none",                   # ✅ disable W&B or other loggers
    save_strategy="epoch",             # ✅ save every epoch
    logging_strategy="epoch"
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_tokenized,
    eval_dataset=test_tokenized,
    tokenizer=tokenizer,
    data_collator=data_collator,
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=2)]  # ✅ stops if no improvement in 2 evals
)
```

```
/usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1611: FutureWarning: `evaluation_strategy` is deprecated and will
  warnings.warn(
<ipython-input-12-758431d95ecc>:33: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`
  trainer = Trainer(
```

```
trainer.train()
```

[6735/6735 3:11:56, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.009000	0.002041	0.999777	0.999764	1.000000	0.999529
2	0.001100	0.002291	0.999777	0.999764	0.999764	0.999764
3	0.000000	0.002276	0.999777	0.999764	0.999764	0.999764

```
TrainOutput(global_step=6735, training_loss=0.0033688701711368277, metrics={'train_runtime': 11519.6917, 'train_samples_per_second':
```

```
model_save_path = "/content/drive/MyDrive/fake_news_models"
trainer.save_model(model_save_path)
tokenizer.save_pretrained(model_save_path)
```

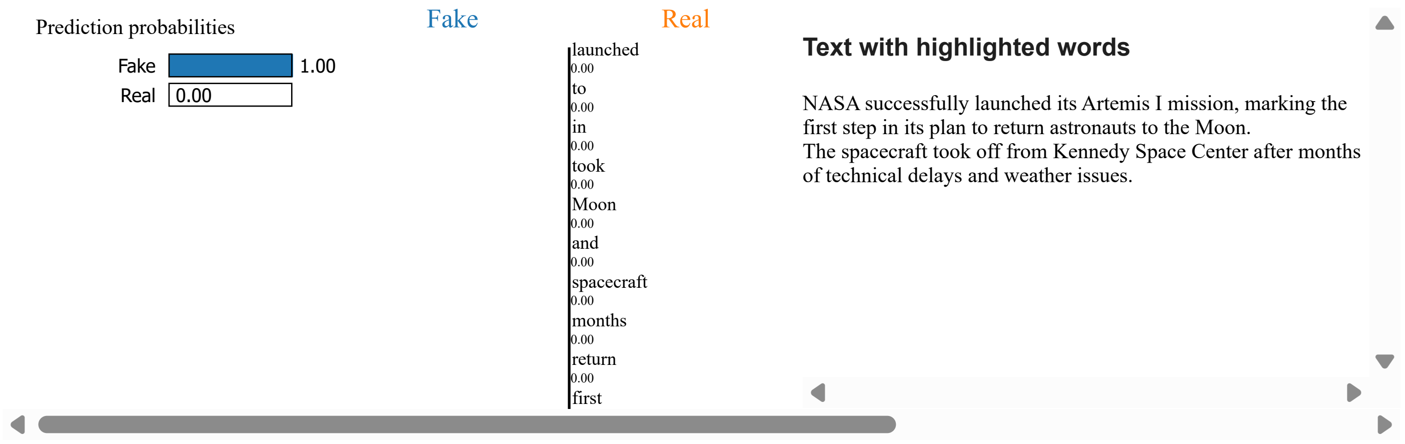
```
('/content/drive/MyDrive/fake_news_models/tokenizer_config.json',
 '/content/drive/MyDrive/fake_news_models/special_tokens_map.json',
 '/content/drive/MyDrive/fake_news_models/vocab.txt',
 '/content/drive/MyDrive/fake_news_models/added_tokens.json')
```

```
trainer.evaluate()
```

[562/562 04:50]

```
{'eval_loss': 0.002290518721565604,
 'eval_accuracy': 0.9997772828507795,
 'eval_f1': 0.9997644842204427,
 'eval_precision': 0.9997644842204427,
 'eval_recall': 0.9997644842204427,
 'eval_runtime': 291.3658,
 'eval_samples_per_second': 30.82,
 'eval_steps_per_second': 1.929,
 'epoch': 3.0}
```

```
from lime.lime_text import LimeTextExplainer

explainer = LimeTextExplainer(class_names=['Fake', 'Real'])

def predict_proba(texts):
    inputs = tokenizer(texts, return_tensors="pt", padding=True, truncation=True).to(model.device)
    with torch.no_grad():
        outputs = model(**inputs)
        probs = torch.nn.functional.softmax(outputs.logits, dim=1)
    return probs.cpu().numpy()

text =text = """
NASA successfully launched its Artemis I mission, marking the first step in its plan to return astronauts to the Moon.
The spacecraft took off from Kennedy Space Center after months of technical delays and weather issues.
"""

exp = explainer.explain_instance(text, predict_proba, num_features=10)
exp.show_in_notebook()
```



Text with highlighted words

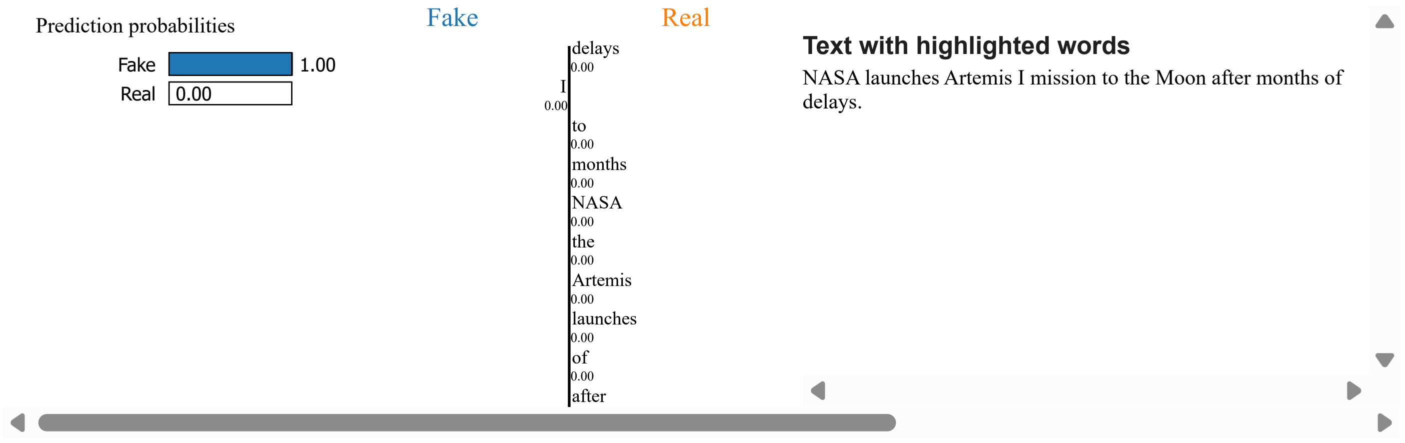
NASA successfully launched its Artemis I mission, marking the first step in its plan to return astronauts to the Moon.
The spacecraft took off from Kennedy Space Center after months of technical delays and weather issues.

```
text = "NASA announces the successful landing of the Perseverance rover on Mars, marking a historic achievement in space exploration."
inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True).to(model.device)
with torch.no_grad():
    outputs = model(**inputs)
    probs = torch.nn.functional.softmax(outputs.logits, dim=1)
    pred_label = torch.argmax(probs, dim=1).item()
    print("Predicted label:", pred_label)  # 0 = Fake, 1 = Real (check your label mapping)
```

```
Predicted label: 0
```

```
from lime.lime_text import LimeTextExplainer
import torch
import numpy as np

# ✅ Correct class names (assuming: 0 = Fake, 1 = Real)
explainer = LimeTextExplainer(class_names=['Fake', 'Real'])

# ✅ Updated probability function to work with LIME
def predict_proba(texts):
    # Tokenize input texts for the BERT model
    inputs = tokenizer(texts, return_tensors="pt", padding=True, truncation=True, max_length=512).to(model.device)
    with torch.no_grad():
        outputs = model(**inputs)
        # Apply softmax to get probabilities
        probs = torch.nn.functional.softmax(outputs.logits, dim=1)
    return probs.cpu().numpy()  # Shape: [num_samples, 2] → [Fake, Real]
```

```
# ✅ Sample real news article for testing
text = "NASA launches Artemis I mission to the Moon after months of delays."

# ✅ Run LIME explanation
exp = explainer.explain_instance(text, predict_proba, num_features=10)

# ✅ Visualize explanation in notebook
exp.show_in_notebook()
```



Text with highlighted words

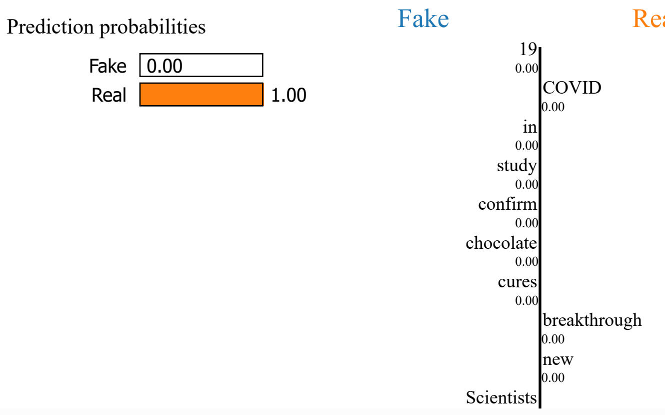
NASA launches Artemis I mission to the Moon after months of delays.

```
def predict_proba(texts):
    inputs = tokenizer(texts, return_tensors="pt", padding=True, truncation=True, max_length=512).to(model.device)
    with torch.no_grad():
        outputs = model(**inputs)
        probs = torch.nn.functional.softmax(outputs.logits, dim=1)
    return probs.cpu().numpy()[:, [1, 0]]  # Flip columns: [Real, Fake] → [Fake, Real]
```

```
probs = predict_proba([text])
predicted_label = np.argmax(probs[0])
print(f"Predicted Label: {'Fake' if predicted_label == 0 else 'Real'}")
print(f"Confidence: {probs[0][predicted_label]:.4f}")
```

```
Predicted Label: Real
Confidence: 1.0000
```

```
# ✅ Sample real news article for testing
text = "Scientists confirm chocolate cures COVID-19 in new breakthrough study."

# ✅ Run LIME explanation
exp = explainer.explain_instance(text, predict_proba, num_features=10)

# ✅ Visualize explanation in notebook
exp.show_in_notebook()
```



Text with highlighted words

Scientists confirm chocolate cures COVID-19 in new breakthrough study.

```
print(train_dataset[0])  # See the label and example
```

```
{'text': 'If you ve been living under a rock for the past few days, you re probably unaware that Donald Trump Jr. committed treason. Aft
```

```
label_map = {0: "Fake", 1: "Real"}
class_names = ['Fake', 'Real']
```

```
def predict_label(text):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True).to(model.device)
    with torch.no_grad():
        outputs = model(**inputs)
        probs = torch.nn.functional.softmax(outputs.logits, dim=1)
        predicted_class = torch.argmax(probs, dim=1).item()
        label = label_map[predicted_class]
        confidence = probs[0][predicted_class].item()
    return label, confidence
```

```
from lime.lime_text import LimeTextExplainer

class_names = ['Fake', 'Real']  # index 0 = Fake, 1 = Real
explainer = LimeTextExplainer(class_names=class_names)

def predict_proba(texts):
    inputs = tokenizer(texts, return_tensors="pt", padding=True, truncation=True).to(model.device)
    with torch.no_grad():
        outputs = model(**inputs)
        probs = torch.nn.functional.softmax(outputs.logits, dim=1)
    return probs.cpu().numpy()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-6-85b5e70124a5> in <cell line: 0>()
----> 1 from lime.lime_text import LimeTextExplainer
      2 
      3 class_names = ['Fake', 'Real']  # index 0 = Fake, 1 = Real
      4 explainer = LimeTextExplainer(class_names=class_names)
      5 

ModuleNotFoundError: No module named 'lime'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

OPEN EXAMPLES

```
text = "NASA launches Artemis I mission to the Moon after months of delays."
label, confidence = predict_label(text)
print(f"Prediction: {label} (Confidence: {confidence:.2f})")

# Run LIME
exp = explainer.explain_instance(text, predict_proba, num_features=10)
exp.show_in_notebook()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-22e62369a7cc> in <cell line: 0>()
      1 text = "NASA launches Artemis I mission to the Moon after months of delays."
----> 2 label, confidence = predict_label(text)
      3 print(f"Prediction: {label} (Confidence: {confidence:.2f})")
      4 
      5 # Run LIME

NameError: name 'predict_label' is not defined
```