

Code with outputs

```
# Install required libraries (run once)

!pip install fastai datasets transformers sentence-transformers --quiet

# Core imports

from datasets import load_dataset

from fastai.text.all import *

import pandas as pd

# Load IMDb dataset from Hugging Face

ds = load_dataset("imdb")

print(ds)
```

```
... /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)
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(
README.md: 7.81k/? [00:00<00:00, 438kB/s]

plain_text/train-00000-of-21.0M/21.0M [00:01<00:00, 14.9MB/s]
00001.parquet: 100%

plain_text/test-00000-of-20.5M/20.5M [00:00<00:00, 26.9MB/s]
00001.parquet: 100%

plain_text/unsupervised-00000-of-42.0M/42.0M [00:00<00:00, 21.5MB/s]
00001.p(...): 100%

Generating train split: 100% 25000/25000 [00:00<00:00, 42406.65 examples/s]
Generating test split: 100% 25000/25000 [00:00<00:00, 45095.05 examples/s]
Generating unsupervised split: 100% 50000/50000 [00:01<00:00, 46122.58 examples/s]
DatasetDict({
  train: Dataset({
    features: ['text', 'label'],
    num_rows: 25000
  })
  test: Dataset({
    features: ['text', 'label'],
    num_rows: 25000
  })
  unsupervised: Dataset({
    features: ['text', 'label'],
    num_rows: 50000
  })
})
```

```
# Use smaller subsets to run faster
```

```

small_train = ds['train'].shuffle(seed=42).select(range(4000))

small_test  = ds['test'].shuffle(seed=42).select(range(2000))

# Convert to DataFrame

train_df = pd.DataFrame(small_train)

test_df  = pd.DataFrame(small_test)

print(train_df.head())

```

```

0  There is no relation at all between Fortier and Profiler but the fact that both are police ser
1  This movie is a great. The plot is very true to the book which is a classic written by Mark Tw
2  George P. Cosmatos' "Rambo: First Blood Part II" is pure wish-fulfillment. The United States c
3  In the process of trying to establish the audiences' empathy with Jake Roedel (Tobey Maguire)
4  Yeh, I know -- you're quivering with excitement. Well, *The Secret Lives of Dentists* will not

   label
0      1
1      1
2      0
3      1
4      0

```

```

# Create DataLoaders

dls = TextDataLoaders.from_df(

    train_df,

    text_col='text',

    label_col='label',

    valid_pct=0.1,

    bs=32

dls.show_batch(max_n=2)

)

```

	text	category
0	xxbos *!! - xxup spoilers - !! * \n\n xxmaj before i begin this , let me say that i have had both the advantages of seeing this movie on the big screen and of having seen the " xxunk xxmaj version " of this movie , remade by xxmaj stephen xxmaj king , himself , in 1997 . \n\n xxmaj both advantages made me appreciate this version of " the xxmaj shining , " all the more . \n\n xxmaj also , let me say that xxmaj i 've read xxmaj mr . xxmaj king 's book , " the xxmaj shining " on many occasions over the years , and while i love the book and am a huge fan of his work , xxmaj stanley xxmaj kubrick 's retelling of this story is far more compelling ... and xxup scary . \n\n xxmaj kubrick	1
1	xxbos xxrep 6 * xxup warning : xxup may xxup contain xxup spoilers xxrep 14 * \n\n xxmaj so who are these " mystery xxmaj men ? " xxmaj simply put , the xxmaj mystery xxmaj men are a group of sub - heroes desperately trying to live out their adolescent fantasy lives while xxunk both their real identities and their super identities . xxmaj the xxmaj xxunk (bill xxmaj macy) works construction during the day , and at night , leaves his wife and kids at home while he cruises the street looking for crimes to tackle with his extraordinary and unique xxmaj xxunk - fighting style . xxmaj the xxmaj blue xxmaj raja (hank xxmaj azaria) sells silverware to xxunk by day and xxunk xxunk at xxunk villians by night , if his mom is n't keeping him busy with the latest snooping .	1

```
# Create text classification learner
```

```
learn = text_classifier_learner(

    dls,

    AWD_LSTM,

    drop_mult=0.5,

    metrics=accuracy

)
```

```
... 100.00% [105070592/105067061 00:10<00:00]
```

```
# Disable problematic progress bar (Colab fix)
```

```
from fastai.callback.progress import ProgressCallback
```

```
learn.remove_cb(ProgressCallback)
```

```
# Train for 1 epoch (lab demo)
```

```
learn.fine_tune(1)
```

```
[0, 0.536102831363678, 0.4489690065383911, 0.7699999809265137, '00:24']
[0, 0.4285890460014343, 0.39358702301979065, 0.8374999761581421, '00:58']
```

```
# Validate accuracy
```

```
learn.validate()
```

```
[0.39358702301979065, 0.8374999761581421]
```

```
learn.export('export.pkl')
```

```
# Predict sentiment for a custom review
```

```
text = "This movie was absolutely fantastic with great acting."
```

```
pred, pred_idx, probs = learn.predict(text)
```

```
print("Review:", text)
```

```
print("Predicted Label:", pred)
```

```
print("Probabilities:", probs)
```

```
Review: This movie was absolutely fantastic with great acting.
```

```
Predicted Label: 1
```

```
Probabilities: tensor([0.0039, 0.9961])
```

```
!pip install gradio --quiet
```

```
import gradio as gr
```

```
from fastai.text.all import load_learner
```

```
# Load your trained FastAI model (exported earlier)
```

```
learn = load_learner('export.pkl')
```

```
# Define prediction function
```

```
def predict_sentiment(text):
```

```
    pred, pred_idx, probs = learn.predict(text)
```

```
    return f"**Sentiment:** {pred}\n**Probabilities:** {probs}"
```

```
# Sample reviews for radio buttons
```

```
sample_reviews = [
```

```
    "This movie was absolutely fantastic with great acting.",
    "I hated this movie. It was a complete waste of time.",
    "The plot was okay, but the acting was terrible.",
    "An amazing cinematic experience! Truly loved it.",
    "Not bad, but could have been much better."
]

# Create Gradio interface using radio buttons
iface = gr.Interface(
    fn=predict_sentiment,
    inputs=gr.Radio(choices=sample_reviews, label="Select a movie review"),
    outputs="text",
    title="IMDb Movie Review Sentiment Classifier",
    description="Select a review from the options and see the predicted sentiment."
)

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

```
... /usr/local/lib/python3.12/dist-packages/fastai/learner.py:455: UserWarning: load_learner` uses Python's insecure pickle module, which can execute malicious arbitrary code when loading. Only load files you trust.
If you only need to load model weights and optimizer state, use the safe `Learner.load` instead.
  warn("`load_learner` uses Python's insecure pickle module, which can execute malicious arbitrary code when loading. Only load files you trust.\nIf you only need to load model weights and optimizer state, use the safe `Lear
It looks like you are running Gradio on a hosted Jupyter notebook, which requires `share=True`. Automatically setting `share=True` (you can turn this off by setting `share=False` in `launch()`) explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://84c43d0ff00580d098.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

IMDb Movie Review Sentiment Classifier

Select a review from the options and see the predicted sentiment.

Select a movie review

☒ This movie was absolutely fantastic with great acting.

☐ I hated this movie. It was a complete waste of time.

☐ The plot was okay, but the acting was terrible.

☐ An amazing cinematic experience! Truly loved it.

☐ Not bad, but could have been much better.

Clear

Submit

output

Sentiment: 1

Probabilities: tensor([0.0039, 0.9961])

Flag