

Week 4 Practical Exercise of Lesson 2: Plug-and-Play Sentiment With BERT Pipelines

Your task

In one short notebook you will:

1. load a small Amazon-reviews subset,
2. run Hugging Face's ready-made pipeline("sentiment-analysis") with a BERT checkpoint,
3. compute simple accuracy and F1, and
4. reflect on latency, memory and model choice.

Step-by-step instructions

1. Open a Jupyter notebook in Colab
File name: bert sentiment pipeline.ipynb
2. Create an import cell

```
!pip -q install transformers datasets evaluate
from transformers import pipeline
from datasets import load_dataset
```

3. Create a cell to Load the data – 500 balanced reviews

```
ds = load_dataset("amazon_polarity", split="test[:500]")
texts = ds["content"]
labels = [1 if r == 2 else 0 for r in ds["label"]] # 1=positive, 0=negative
```

4. Instantiate the pipeline

Leave the default model

```
distilbert-base-uncased-finetuned-sst-2-english
```

or swap in another BERT family checkpoint via the **model_name** parameter.

```
clf = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english")
```

5. Run inference & time it
6. Compute metrics Using the evaluate library in python using the following Code

```
import evaluate
```

```
accuracy = evaluate.load("accuracy").compute(predictions=preds, references=labels)
f1 = evaluate.load("f1").compute(predictions=preds, references=labels)
```

```
print(accuracy, f1)
```

7. (Optional) Try a second checkpoint

Run the same loop with, for example, "bert-base-uncased" fine-tuned on SST-2 or a multilingual model and record speed / scores.

8. Write a 120-word reflection

In the final markdown cell answer:

- Which checkpoint did you keep and why?
- How do latency and F1 balance for your use-case?

Deliverables

- Completed notebook with all cells executed.
- Reflection paragraph inside the notebook (or separate file).