Worksheet 3

Student Name:Sarvesh yadav UID:25IMC13002
Branch: UIC Section/Group:25MAM-3(B)
Semester: 2 Date of Performance:17/02/26
Subject Name: Generative AI Subject Code: 25CAP-674

1. Aim –

To implement a sentiment analysis system using a pretrained BERT Transformer model in PyTorch that classifies input text as Positive or Negative based on predicted probabilities.

2. Apparatus:

Supported Operating Systems:

- Windows 10 64-bit Pro, Enterprise, or Education

Programming Language:

- Python

Development Environment:

- Google cloab

3.Coding :

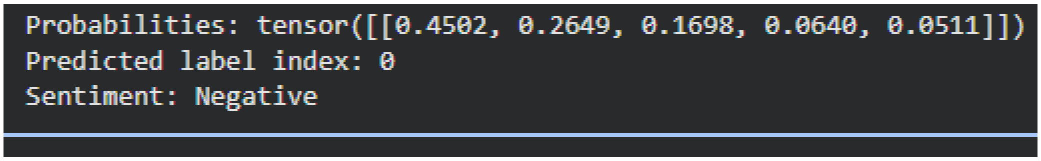
```
from transformers import AutoModelForSequenceClassification, AutoTokenizer

import torch

# Load model and tokenizer
```

```
model_name = "nlptown/bert-base-multilingual-uncased-sentiment"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)
# Sample text
text = "We should not encourage corruption"
# Tokenization
inputs = tokenizer(
    text,
    return_tensors="pt",
    padding=True,
    truncation=True
)
# Inference (no gradients needed)
with torch.no_grad():
    logits = model(**inputs).logits
# Convert logits to probabilities
probabilities = torch.nn.functional.softmax(logits, dim=-1)
# Predicted class index (0–4)
prediction = torch.argmax(probabilities, dim=1).item()
# Interpret result
sentiment = "Positive" if prediction > 2 else "Negative"
print("Probabilities:", probabilities)
print("Predicted label index:", prediction)
print("Sentiment:", sentiment)
```

4)output-



```
Probabilities: tensor([[0.4502, 0.2649, 0.1698, 0.0640, 0.0511]])
Predicted label index: 0
Sentiment: Negative
```

5. Learning outcomes (What I have learnt):

- Used a pretrained BERT model to perform sentiment analysis on text data.
- Understood tokenization and how text is converted into numerical input for the model.
- Performed inference using PyTorch without computing gradients.
- Applied Softmax to convert logits into probability values.
- Classified text into Positive or Negative based on predicted output.