Sentiment Analysis for Social Media Monitoring

It uses the Hugging Face `transformers` library to perform **sentiment analysis** on a set of example posts.

1. Import the Sentiment Analysis Pipeline

```
from transformers import pipeline
```

This line imports the `pipeline` function from Hugging Face's `transformers` library. Pipelines are pre-built models for various tasks (like sentiment analysis, translation, etc.), making it easy to apply machine learning without extensive configuration.

2. Initialize the Sentiment Analysis Model

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

Here, a sentiment analysis pipeline is created using the `pipeline` function. The model `distilbert-base-uncased-finetuned-sst-2-english` is explicitly specified, which is a lightweight BERT model pre-trained for sentiment analysis. This model categorizes text as either positive or negative.

3. Define Sample Posts

```
posts = [
    "The recent changes in the app have made it much easier to use!",
    "I didn’t find the update helpful at all; it made things more complicated.",
    "The support team was so responsive and helped solve my issues quickly!",
    "This product is overpriced and doesn’t meet my expectations."
]
```

This list contains sample social media posts, with each post representing different types of user feedback. Some posts express positive feedback, while others convey negative opinions, providing a range for the model to classify.

4. Perform Sentiment Analysis

```
for post in posts:
    result = sentiment_model(post)[0]
    print(f"Text: {post}")
    print(f"Sentiment: {result['label']} with score {result['score']:.2f}\n")
```

In this loop:

- Each `post` in the `posts` list is processed by `sentiment_model(post)`.

- `result = sentiment_model(post)[0]`: The pipeline returns a list of results; `[0]` extracts the first result for each post.

- `result['label’] ` and `result['score’] ` provide the sentiment (e.g., "POSITIVE" or "NEGATIVE") and confidence score for that sentiment (e.g., 0.98).

- `print(f"Text: {post}")` and `print(f"Sentiment: {result['label']} with score {result['score']:.2f}\n")`: Print each post and its sentiment with a confidence score formatted to two decimal places.

This loop processes each post, displaying whether it's positive or negative and how confident the model is in its classification.

Required Installations

To run this code, you need the following packages:

1. **Transformers** (by Hugging Face):

pip install transformers

2. **PyTorch** (required to run the Hugging Face models):

pip install torch