Hugging Face Model Exploration

Objective:

To explore an open-source LLM from the Hugging Face Hub, load it locally, and use it for a simple NLP task such as text classification. This assignment helps understand the environment setup, model loading, and basic inference using the Transformers library.

1. Environment Setup Commands

 # Step 1: Create a Python virtual environment
python -m venv hf_env

 # Step 2: Activate the environment
On Windows:
hf_env\Scripts\activate
On macOS/Linux:
source hf_env/bin/activate

 # Step 3: Install required libraries
pip install transformers torch

Explanation:

python -m venv hf_env → Creates an isolated environment for dependencies.

activate → Enables the virtual environment.

pip install transformers torch → Installs Hugging Face Transformers library and PyTorch backend.

2. Model Selection

Chosen Model: distilbert-base-uncased-finetuned-sst-2-english

DistilBERT is a smaller, faster version of BERT.

Uncased → Ignores letter case (treats "Hello" and "hello" the same).

Finetuned SST-2 → Trained on the Stanford Sentiment Treebank for sentiment classification.

3. Python Script

# Import the required library
from transformers import pipeline

# Load sentiment analysis pipeline with the chosen model
classifier = pipeline(

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

# Input text for testing
test_text = "Hugging Face makes working with transformers very easy!"

# Perform sentiment classification
result = classifier(test_text)

# Print the results
print("Input Text:", test_text)
print("Model Output:", result)
```

4. Sample Output

```
Input Text: Hugging Face makes working with transformers very easy!
Model Output: [{'label': 'POSITIVE', 'score': 0.9998}]
```

Explanation:

Label: Sentiment category (POSITIVE or NEGATIVE).

Score: Confidence level of the prediction (between 0 and 1).

5. Observations

i. The model correctly detected the text sentiment as Positive with a high confidence score.
ii. Using pipeline() makes model loading and inference extremely simple — no manual tokenization or preprocessing required.
iii. Open-source models from Hugging Face can be swapped for different tasks such as summarization, translation, and question answering by just changing the pipeline type and model.

Final Note

This exercise demonstrated:

1. Creating and activating a Python virtual environment.
2. Installing Transformers & PyTorch.
3. Selecting and loading a Hugging Face model.
4. Running inference and interpreting results.