Project Implementation Report: AI-Generated Speech Detection Using Whisper Model

1. Implementation Process

Challenges Encountered:

- Audio Format Compatibility: Whisper expects audio in a specific format (16kHz, mono WAV). Initial uploads in MP3 caused conversion failures.
- Dataset Handling: Managing and organizing thousands of audio files from two large datasets (fake and real) required time and efficient I/O operations.

How Challenges Were Addressed:

- Audio Conversion: Used ffmpeg to convert audio files to the required format before feeding them to the model.
- Efficient Loading: Implemented batch processing and progress tracking to process large datasets effectively.

Assumptions Made:

- Whisper transcriptions of fake and real voices contain enough linguistic or phonetic signals to distinguish between the two.
- Transcription-based classification is viable for AI-generated speech detection.

2. Analysis Section

Model Selection Rationale:

- Whisper is a robust and open-source speech-to-text model from OpenAI, known for high accuracy and multilingual support.
- Its ability to transcribe noisy and diverse audio samples made it a suitable base for

downstream classification.

High-Level Technical Explanation:

- Whisper converts audio into text using a transformer-based encoder-decoder architecture.
- We used its transcriptions as input features (converted into TF-IDF vectors) for a logistic regression classifier to differentiate fake vs real audio.

Performance Results:

- Accuracy: 60% on a balanced test set of 40 audio files.
- Precision/Recall/F1: Averaged 0.60 across both classes, indicating room for improvement but a promising baseline.

Strengths and Weaknesses:

Strengths:

- Whisper reliably transcribes a wide variety of audio inputs.
- Easy integration with downstream ML pipelines for classification.

Weaknesses:

- Whisper does not distinguish between real/fake speech directly.
- Accuracy is limited without acoustic features (e.g., waveform analysis).

Suggestions for Improvement:

- Incorporate acoustic features (e.g., MFCC, pitch) along with text.
- Use deep learning classifiers like BiLSTM or transformer-based text classifiers.
- Expand dataset with more diverse real and AI voices.

3. Reflection Questions

1. Significant Challenges:

- Handling file conversions and Whisper’s audio constraints.

- Balancing data preprocessing speed with model training time.

2. Real-World vs Research Conditions:

- Real-world audio is often noisy, with accents, emotions, background noise—posing greater difficulty.

- Whisper performs well, but transcription inconsistencies could reduce accuracy in live scenarios.

3. Additional Resources Needed:

- Larger dataset covering multiple AI voice generators.

- Audio quality normalization tools.

- Use of pretrained audio embeddings (e.g., Wav2Vec).

4. Production Deployment Approach:

- Build a REST API that accepts audio uploads.

- Integrate Whisper transcription pipeline.

- Classify using an optimized model hosted via Flask/FastAPI or Streamlit for prototyping.

- Containerize with Docker for scalable deployment.

End of Report