

Coding Assignment #1

Assignment:

Blipper's AI-based Speech Analytics product is designed to derive key insights from voice-based conversations. Your assignment involves creating a rudimentary prototype, following these directions:

Approach:

This code is designed to perform sentiment analysis on audio files. It uses the SpeechRecognition library to transcribe audio files into text and the NLTK library's VADER sentiment analyzer to assess sentiment. The code then creates a Pandas DataFrame to store the results, prints the transcriptions and sentiment analysis, and generates a line chart to visualize sentiment trends in the provided audio files.

Features Chosen:

1. **Speech Recognition:** Utilizes the Google Web Speech API to transcribe audio files.
2. **Sentiment Analysis:** Employs the VADER sentiment analyzer from NLTK to evaluate sentiment.
3. **Data Storage:** Stores the results in a Pandas DataFrame.
4. **Data Visualization:** Produces a line chart to display sentiment analysis results graphically.

Preprocessing Steps:

1. Downloaded the VADER lexicon using ``nltk.download("vader_lexicon")``.
2. Defined a custom function, ``map_sentiment()``, to categorize compound sentiment scores as "happy," "sad," or "neutral."

Libraries and Tools:

- ``os``: For file handling operations.
- ``speech_recognition``: To recognize and transcribe audio.
- ``nltk``: For sentiment analysis using VADER.
- ``pandas``: To create a DataFrame for storing results.
- ``matplotlib``: To generate line charts for sentiment visualization.

Challenges Faced:

1. **Audio Format Limitation:** The code is designed to work with audio files in WAV format. If you have audio files in other formats (e.g., MP3), you'll need to convert them to WAV or modify the code to handle different formats.
2. **Internet Connection:** SpeechRecognition relies on an internet connection to access the Google Web Speech API. Ensure you have a working internet connection for transcription.

3. Accurate Transcriptions: The accuracy of transcriptions may vary depending on the audio quality and the clarity of the speaker's voice. If you encounter issues with transcription accuracy, you may need to preprocess the audio or explore other transcription options.

4. Sentiment Analysis Accuracy: The sentiment analysis provided by VADER is based on a pre-trained model and may not be perfect for all types of content. It's essential to understand the limitations and potential biases of the sentiment analysis tool being used.

User Manual:

1. Input Audio Files:

- You need to provide the path to the directory containing your audio files. You'll be prompted to enter the directory path when running the script.
- The code assumes that all audio files in the specified directory are in WAV format. If you have audio files in other formats, convert them to WAV or modify the code to handle the desired format.

2. Accepted Audio File Formats:

- The code is set up to work with audio files in WAV format. Ensure that your audio files have a ".wav" extension.

3. Print the Results:

- The code will transcribe the audio files, perform sentiment analysis, and print the results to the console.

- You can view the transcriptions and their corresponding sentiment labels for each audio file.

4. Data Table:

- The code will generate a Pandas DataFrame to store the results, and it will print this table to the console. The table includes columns for the file name, transcription, and sentiment label.

5. Sentiment Chart:

- The code will also generate a line chart to visualize sentiment analysis results. This chart will display the sentiment labels for each audio file.

- The chart will be displayed on your screen for visual inspection.

****Ensure you have the necessary libraries (speech_recognition, nltk, pandas, matplotlib) installed before running the code, and be aware of the limitations and requirements mentioned in the "Challenges Faced" section.****