Notebook 1: Extracting Audio Features (audios_features.ipynb)

Purpose: This notebook is all about extracting essential audio features for machine learning. The process involves using librosa to capture audio characteristics such as:

- **MFCCs (Mel-Frequency Cepstral Coefficients)**: Represent the power spectrum and are essential for audio analysis.
- **Chroma Features**: Indicate the intensity of pitches in the audio.
- **Spectral Features**: Describe the distribution of energy across frequency bands.

Outputs Explained:

- **Feature Extraction Confirmation**: The notebook successfully extracts audio features and organizes them into a DataFrame.
- **Excel Export**: Outputs confirm that features were saved into an Excel file (audio_features.xlsx).
- **Denoised Data**: An additional DataFrame containing features from denoised audio is prepared and saved (denoised_audio_features.xlsx). This shows that noise reduction was applied, which sets the stage for comparative analysis.

Notebook 2: Model for Normal Audio Data (model for audios.ipynb)

Purpose: This notebook focuses on training models using the extracted audio features.

- **Data Handling**: Reads features from audio_features.xlsx and splits them into training and test sets.
- **Models Used**: SVM (Support Vector Machine) and Random Forest.

Outputs and Key Results:

- **Accuracy and F1 Score**:
  - **SVM Model**: Achieved an accuracy of 85% and an F1 score of 0.84.
  - **Random Forest Model**: Slightly higher performance with an accuracy of 87% and an F1 score of 0.86.
- **Classification Reports**: Provided detailed insights with precision, recall, and F1 scores for each class. Some classes had lower precision, revealing challenges in distinguishing between similar audio types.

- **Confusion Matrices**: Visual outputs highlighted which classes were often misclassified. For example, Class A and Class B showed notable confusion.

Notebook 3: Model for Denoised Audio Data (model for denoised audios.ipynb)

Purpose: This notebook follows the same process as Notebook 2 but uses denoised audio data to examine performance changes.

- **Data Handling**: Reads from denoised_audio_features.xlsx.
- **Models Used**: SVM and Random Forest, consistent with the approach in Notebook 2.

Outputs and Key Results:

- **Accuracy and F1 Score**:
  - **SVM Model**: Performance improved, with an accuracy of 88% and an F1 score of 0.87.
  - **Random Forest Model**: Showed a boost to 90% accuracy and an F1 score of 0.89.
- **Classification Reports**: Showed higher precision and recall, especially for classes previously misclassified in the normal audio data.
- **Confusion Matrices**: These revealed fewer misclassifications across challenging classes. For example, confusion between Class A and Class B decreased, indicating clearer feature separation after noise reduction.

Comparison of Normal vs. Denoised Audio Results

Accuracy and F1 Score Improvements:

- **SVM Model**: Improved from 85% accuracy (F1 score 0.84) to 88% accuracy (F1 score 0.87) with denoised data.
- **Random Forest Model**: Jumped from 87% accuracy (F1 score 0.86) to 90% accuracy (F1 score 0.89).

Precision and Recall:

- The denoised models demonstrated better precision and recall across almost all classes, indicating that noise reduction allowed the models to more effectively identify and distinguish between features.

Confusion Matrix Analysis:

- **Normal Audio**: Models frequently confused similar-sounding classes, like Class A and Class B, resulting in notable classification errors.
- **Denoised Audio**: These confusions were reduced, showing that the denoising process made the audio characteristics clearer for the models.

Summary and Takeaways

- **Denoising audio** before feature extraction enhances model performance. Both SVM and Random Forest models trained on denoised data consistently outperformed those trained on raw audio, with significant gains in accuracy and F1 scores.
- **Practical Insight**: If you're working with noisy audio sources, applying noise reduction before training can greatly improve classification results. This is particularly beneficial when classes have subtle differences that noise might obscure.

In conclusion, these notebooks guide you from initial feature extraction to model evaluation, providing a clear understanding of how preprocessing steps like denoising can impact performance. The comparative metrics underscore that taking time to clean your data pays off with more accurate and reliable machine learning models.