**give me a details for the part 01 also**

Here is a **detailed, mathematically rigorous, and assignment-compliant guide** for your **Part 01 (Audio Classification Using Feature Extraction)** report, based on your PDF and the high-accuracy code you used.

**Part 1: Audio Classification Using Feature Extraction**

**1. Introduction**

Briefly state the problem:

* The objective is to classify unknown audio signals as belonging to either **class\_1** or **class\_2** using digital signal processing and feature extraction techniques.
* The approach involves extracting robust features from each audio file, comparing these features between known and unknown samples, and assigning each unknown file to the most similar known class.

**2. Methodology**

**2.1. Feature Extraction**

**Process:**

* For every audio file, extract features that capture the unique characteristics of the sound.
* The two main methods suggested in the assignment are:
  + **MFCC (Mel-Frequency Cepstral Coefficients)**
  + **FFT (Fast Fourier Transform)**

**Mathematical Explanation:**

**MFCCs:**

* MFCCs are computed by:
  1. **Framing:** Divide the audio signal $ x[n] $ into short frames.
  2. **Windowing:** Multiply each frame by a window function (e.g., Hamming).
  3. **FFT:** Compute the magnitude spectrum for each frame:
  4. **Mel Filter Bank:** Pass the spectrum through a bank of triangular bandpass filters spaced on the Mel scale.
  5. **Logarithm:** Take the logarithm of the filter bank energies.
  6. **DCT:** Apply the Discrete Cosine Transform to decorrelate the coefficients and obtain the MFCCs:where $ E\_n $ is the energy in the $ n $-th Mel filter.
* **Why MFCCs?**  
  MFCCs are robust to noise and variation, and they mimic human auditory perception, making them ideal for distinguishing between classes of sounds.

**Alternative: FFT**

* The FFT provides the frequency spectrum, but MFCCs are generally more effective for classification due to their perceptual basis.

**2.2. Similarity/Distance Metrics**

**Process:**

* For each unknown file, compare its feature vector to those of all files in **class\_1** and **class\_2**.
* Use a distance or similarity metric to measure how "close" the unknown file is to each class.

**Mathematical Explanation:**

* **Euclidean Distance:**  
  For feature vectors $ \mathbf{f}\_u $ (unknown) and $ \mathbf{f}\_k $ (known):
* **Cosine Similarity:**
* **Manhattan Distance:**
* **Why Euclidean?**  
  Euclidean distance is simple, effective, and aligns with the assignment requirements.

**2.3. Classification Logic**

**Process:**

* For each unknown file:
  1. Extract its MFCC feature vector.
  2. Compute the distance to every file in **class\_1** and **class\_2**.
  3. Assign the file to the class whose *closest* training file (minimum distance) is nearest.

**Mathematical Rule:**

* Let $ D\_1 $ be the minimum distance to any file in **class\_1**, $ D\_2 $ for **class\_2**.
* Assign to:
* **Alternative:** Use K-Nearest Neighbors (KNN) to assign based on the majority class among the k closest training files.

**3. Implementation Details**

* **Code Structure:**
  1. Load all audio files from **class\_1**, **class\_2**, and **unknown**.
  2. Preprocess: Convert to mono, resample to a common sampling rate.
  3. Extract MFCC features for all files.
  4. For each unknown file, compute the Euclidean distance to all known files.
  5. Assign the class based on the minimum distance.
* **Tools:** MATLAB (using audioread, mfcc, mean, std, and KNN classifier).

**4. Results**

* **Present a table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File Name | Predicted Class | Confidence | Min Dist to Class 1 | Min Dist to Class 2 |
| unknown1.wav | class\_1 | 0.93 | 1.12 | 2.45 |
| unknown2.wav | class\_2 | 0.88 | 2.05 | 1.34 |
| ... | ... | ... | ... | ... |

* **Report overall accuracy:**

* **Confidence:**  
  Optionally, define confidence as the ratio of the distances:

**5. Conclusion**

* **Summary:**
  + MFCCs with Euclidean distance (or KNN) provided robust classification.
  + The method is fast, interpretable, and reliable for real-time and batch classification.
* **Challenges:**
  + Overlapping classes or noisy recordings may reduce accuracy.
* **Improvements/Alternatives:**
  + Combining MFCCs with other features (e.g., spectral centroid, zero-crossing rate) could further improve performance.
  + Trying different classifiers (SVM, Random Forest) or distance metrics.

**Mathematical Summary of the MFCC-Based Pipeline**

1. **Framing and Windowing:**  
   $ x[n] \to x\_w[n] = x[n] \cdot w[n] $
2. **FFT:**  
   $ X[k] = \sum\_{n=0}^{N-1} x\_w[n] e^{-j 2\pi k n / N} $
3. **Mel Filter Bank:**  
   Apply triangular filters to $ |X[k]| $ to get energies $ E\_n $.
4. **Logarithm and DCT:**  
   $ c\_m = \sum\_{n=1}^{K} \log(E\_n) \cos\left[\frac{\pi m}{K}(n-0.5)\right] $
5. **Distance Calculation:**  
   $ d(\mathbf{f}*u, \mathbf{f}k) = \sqrt{\sum{i=1}^{N} (f*{u,i} - f\_{k,i})^2} $
6. **Classification:**  
   Assign unknown to class with minimum distance.

**References**

* Your assignment PDF ([Simple-Audio-Classification-Using-Feature-Extraction-1-1.pdf](https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/65817725/d288811f-4b0c-4fe5-b4a9-c4e42367034f/Simple-Audio-Classification-Using-Feature-Extraction-1-1.pdf))
* [MFCC Theory and Applications](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum)
* [MATLAB MFCC Documentation](https://www.mathworks.com/help/audio/ref/mfcc.html)

**This structure will make your Part 1 section detailed, mathematically rigorous, and fully compliant with your assignment PDF. You can copy-paste and adapt these explanations, equations, and justifications directly into your report.**

⁂