Objectives: The objectives of the code are to extract and analyze key features from a PPG signal for physiological analysis:

1. **Pulse Amplitude**: Measures the peak-to-baseline difference of each pulse, indicating the strength of the heartbeat.
2. **Pulse Duration**: Calculates the time between successive peaks, representing the duration of a pulse.
3. **Inter-Beat Interval (IBI)**: The time between consecutive heartbeats, useful for assessing heart rate variability.
4. **Peak-to-Peak Interval (PPI)**: The time between consecutive peaks, used to measure the pulse interval.
5. **Signal Entropy**: Measures the unpredictability or randomness of the PPG signal, providing insights into the complexity of the signal.
6. **Root Mean Square (RMS)**: Represents the signal's power, showing the overall signal intensity.
7. **Dominant Frequency**: Identifies the dominant frequency in the signal, giving information about the heart rate.
8. **Total Power**: The total energy present in the PPG signal across frequencies, indicating the signal's strength.

The Code is here:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import find_peaks, butter, filtfilt
from scipy.stats import entropy
from scipy.signal import welch

#Dataset
file_path = 'emd_1_imfs.csv'
ppg_data = pd.read_csv(file_path)
```

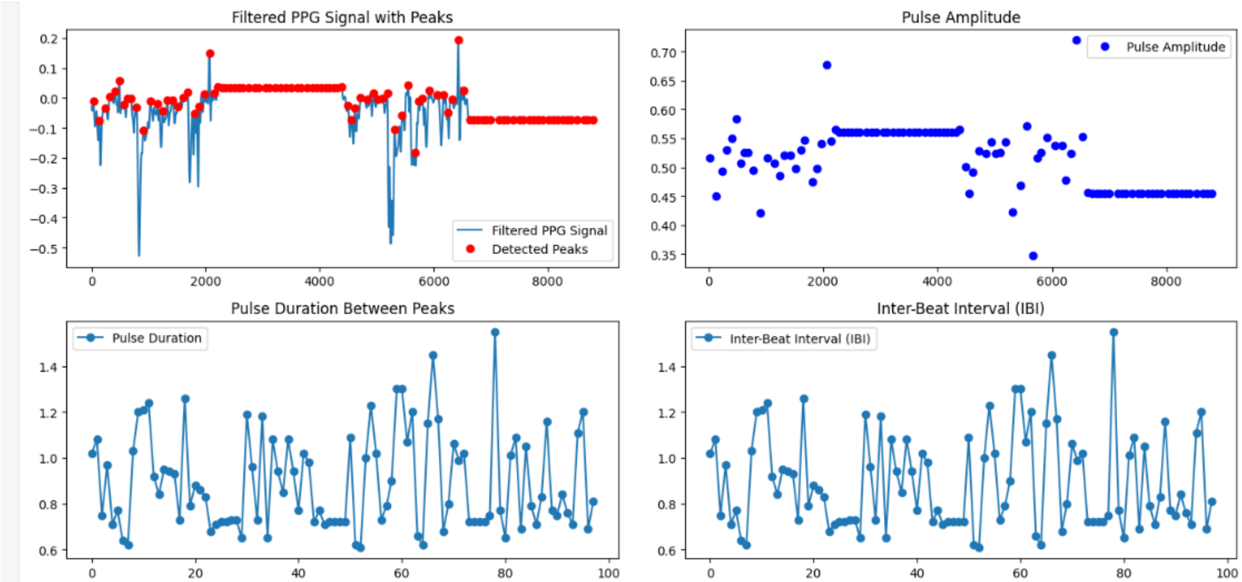
```python
ppg_signal = ppg_data['Imf_1_MEAN'].values
def butter_lowpass(cutoff, fs, order=4):
    nyquist = 0.5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = butter(order, normal_cutoff, btype='low',
analog=False)
    return b, a

def butter_lowpass_filter(data, cutoff, fs, order=4):
    b, a = butter_lowpass(cutoff, fs, order)
    y = filtfilt(b, a, data)
    return y

# Filter PPG signal
fs = 100
cutoff = 3
filtered_ppg = butter_lowpass_filter(ppg_signal, cutoff,
fs)

#Feature Extraction

# Peak detection using scipy find_peaks
peaks, _ = find_peaks(filtered_ppg, distance=fs*0.6

# Pulse Amplitude: Peak-to-baseline difference
pulse_amplitude = filtered_ppg[peaks] -
np.min(filtered_ppg)

pulse_duration = np.diff(peaks) / fs

# Inter-Beat Interval (IBI
```

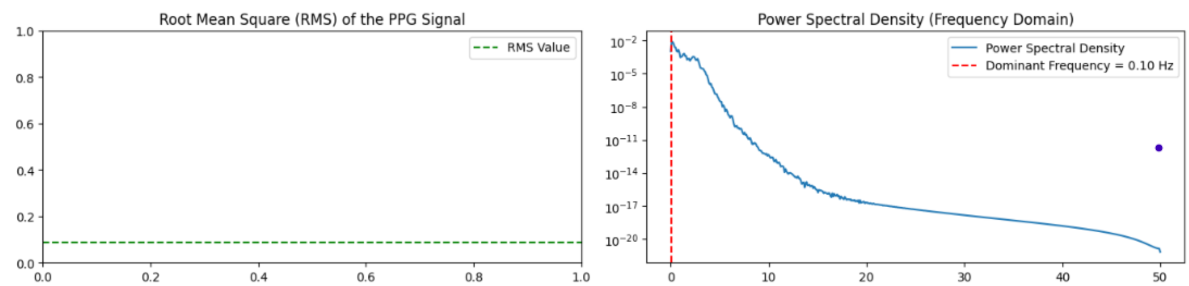
```
ibi = np.diff(peaks) /

# Signal Variability (IBI Standard Deviation)
ibi_variability = np.std(ibi)

# Peak-to-Peak Interval (PPI
ppi = np.diff(peaks) / fs

# Entropy of the Signal (measuring unpredictability)
ppg_entropy = entropy(filtered_ppg)

# Root Mean Square (RMS) of the signal
rms_value = np.sqrt(np.mean(filtered_ppg**2))

# Frequency Domain Analysis
f, psd = welch(filtered_ppg, fs, nperseg=1024)
dominant_frequency = f[np.argmax(psd

total_power = np.sum(psd)

#Visualize Extracted Features
plt.figure(figsize=(14, 10))

# Plot filtered signal with peaks
plt.subplot(3, 2, 1)
plt.plot(filtered_ppg, label='Filtered PPG Signal')
plt.plot(peaks, filtered_ppg[peaks], 'ro',
label='Detected Peaks')
plt.title("Filtered PPG Signal with Peaks")
plt.legend()

# Pulse Amplitude
```

```
plt.subplot(3, 2, 2)
plt.plot(peaks, pulse_amplitude, 'bo', label='Pulse
Amplitude')
plt.title("Pulse Amplitude")
plt.legend()

# Pulse Duration
plt.subplot(3, 2, 3)
plt.plot(np.arange(len(pulse_duration)), pulse_duration,
label="Pulse Duration", marker='o')
plt.title("Pulse Duration Between Peaks")
plt.legend()

# Inter-Beat Interval (IBI)
plt.subplot(3, 2, 4)
plt.plot(np.arange(len(ibi)), ibi, label="Inter-Beat
Interval (IBI)", marker='o')
plt.title("Inter-Beat Interval (IBI)")
plt.legend()

# RMS of the signal
plt.subplot(3, 2, 5)
plt.axhline(rms_value, color='g', linestyle='--',
label="RMS Value")
plt.title("Root Mean Square (RMS) of the PPG Signal")
plt.legend()

# Frequency Domain (Power Spectral Density)
plt.subplot(3, 2, 6)
plt.semilogy(f, psd, label="Power Spectral Density")
```

```
plt.axvline(dominant_frequency, color='r', linestyle='--
', label=f"Dominant Frequency = {dominant_frequency:.2f}
Hz")
plt.title("Power Spectral Density (Frequency Domain)")
plt.legend()

plt.tight_layout()
plt.show()

# Step 5: Print Extracted Features
print(f"Pulse Amplitude (mean):
{np.mean(pulse_amplitude):.2f} V")
print(f"Average Pulse Duration:
{np.mean(pulse_duration):.2f} seconds")
print(f"Average Inter-Beat Interval (IBI):
{np.mean(ibi):.2f} seconds")
print(f"IBI Variability (Standard Deviation):
{ibi_variability:.2f}")
print(f"Peak-to-Peak Interval (mean): {np.mean(ppi):.2f}
seconds")
print(f"Signal Entropy: {ppg_entropy:.2f}")
print(f"RMS of the Signal: {rms_value:.2f}")
print(f"Dominant Frequency: {dominant_frequency:.2f}
Hz")
print(f"Total Power in Signal: {total_power:.2f}")
```

The output is here:






```
Pulse Amplitude (mean): 0.51 V
Average Pulse Duration: 0.89 seconds
Average Inter-Beat Interval (IBI): 0.89 seconds
IBI Variability (Standard Deviation): 0.21
Peak-to-Peak Interval (mean): 0.89 seconds
Signal Entropy: -inf
RMS of the Signal: 0.09
Dominant Frequency: 0.10 Hz
Total Power in Signal: 0.04
```

[ ]:

Pulse Amplitude (mean): 0.51 V

Average Pulse Duration: 0.89 seconds

Average Inter-Beat Interval (IBI): 0.89 seconds

IBI Variability (Standard Deviation): 0.21

Peak-to-Peak Interval (mean): 0.89 seconds

Signal Entropy: -inf

RMS of the Signal: 0.09

Dominant Frequency: 0.10 Hz

Total Power in Signal: 0.04