Emotion. ipynb

Dataset & Metadata

You started with metadata.csv that contains detailed info for each EEG recording:

- **EEG recording info:** Filename, sample rate, number of channels, duration.
- **Subject info:** Age, sex, health status, experience/treatment group.
- **EEG context:** Sleep stage, artifacts, other signals (EOG, EMG, ECG).

This metadata helps **contextualize the EEG data**—for instance, high artifact proportion may require preprocessing, or sleep stage affects frequency bands.

---

2. Preprocessing

Before analysis, EEG data is cleaned:

- **Filtering**: Removes noise (e.g., 0.5–45 Hz keeps relevant brain rhythms).
- **Artifact removal**: Blinks, muscle activity, or ECG interference are removed using ICA or artifact proportion info from metadata.
- **Channel selection**: Only EEG channels are used; other signals like EOG/EMG are ignored for emotion analysis.

This ensures that **subsequent feature extraction reflects true brain activity**.

---

3. Feature Extraction (Frequency Bands)

EEG is decomposed into five main bands:

Band	Frequency	Associated State
Delta	0.5–4 Hz	Deep sleep, rest
Theta	4–8 Hz	Drowsy, relaxation
Alpha	8–13 Hz	Calm, relaxed, awake
Beta	13–30 Hz	Alert, active, anxious

Band	Frequency	Associated State
Gamma	30–45 Hz	Cognitive processing, attention

- **Method**: Power spectral density (PSD) using Welch's method.
- **Output**: Numerical values representing "strength" of each band for each time segment.

These values are **interpretable features** that correlate with mental/emotional states.

---

4. Emotion Mapping

You mapped EEG features to emotional states:

- **Fear:** High beta relative to alpha/theta (alertness, stress).
- **Happy:** Higher alpha + beta (relaxed but engaged).
- **Neutral:** Balanced theta, alpha, beta (no extreme emotional activation).

This mapping is **biologically motivated**, based on well-known EEG-emotion correlations.

---

5. Modeling Approaches

(a) EEGClassifier

- A supervised classifier trained on [delta, theta, alpha, beta, gamma].
- Learns **patterns in band powers** corresponding to labeled emotions.
- Fast, interpretable, baseline for comparison.

(b) KMeans Clustering

- Unsupervised approach to group similar EEG patterns.
- Can **discover hidden clusters** of emotional responses without explicit labels.
- Helps validate whether EEG features naturally separate emotions.

(c) LSTM (Long Short-Term Memory)

- Treats EEG as **temporal sequences** rather than independent segments.
- Captures **dynamic changes in brain activity** over time.
- Ideal for EEG because emotions evolve gradually, and patterns over time are meaningful.

- More robust than static classifiers for sequential signals.

---

6. Pipeline Overview

1. **Load & preprocess EEG** → clean data.
2. **Extract band powers** → delta, theta, alpha, beta, gamma.
3. **Assign emotion labels** → fear, happy, neutral (based on band ratios).
4. **Apply models**:
    - EEGClassifier → supervised learning.
    - KMeans → unsupervised exploration.
    - LSTM → temporal pattern recognition.
5. **Analyze output** → predict emotional state per time segment or per subject.

---

7. Scientific Justification

- Frequency bands are **biomarkers of cognitive and emotional states**.
- Supervised classifiers validate known patterns, while clustering explores unknown patterns.
- LSTM respects the **time-series nature of EEG**, making predictions more accurate and biologically meaningful.
- This combination of methods gives a **complete picture**: statistical patterns, clusters, and temporal dynamics.

SleepingStage.ipynb

Dataset & Metadata

You already have metadata.csv with details like:

- **EEG sample rate**: 256 Hz
- **Number of EEG channels**: 32

- **Sleep stage info**: N1, N2, N3, REM, Wake
- **Other signals**: EOG, EMG, ECG (useful for sleep stage detection)
- **Artifact proportion**: Helps decide if data cleaning is needed

This metadata allows you to **preprocess EEG appropriately** for sleep stage classification.

---

2. Preprocessing

Steps to ensure clean input:

- **Filtering:** Bandpass 0.5–45 Hz (EEG frequencies relevant for sleep stages).
- **Artifact removal:** ICA or threshold-based removal (muscle, eye movements).
- **Segmentation:** EEG is divided into epochs (e.g., 30-second windows), the standard in sleep analysis.

This gives **clean, uniform epochs** to extract features from.

---

3. Feature Extraction

For sleep stage prediction, you extract **features that correlate with brain states**:

- **EEG frequency bands:** Delta, Theta, Alpha, Beta, Gamma (PSD via Welch).
- **Derived features:**
  - Delta/Theta ratio
  - Alpha/Beta ratio
  - Band power variance
  - Energy, entropy, Hjorth parameters
- **Optional:** Include EOG and EMG signals (eye movements and muscle activity are crucial for REM vs NREM).

These features form a **numerical vector per epoch**, which is the input to classifiers.

---

4. Labeling

- Labels come from **ground truth sleep staging** (metadata or manual scoring):

- **Wake, N1, N2, N3, REM**

- Each EEG epoch is assigned a sleep stage label.

This is **supervised learning**.

---

5. Models Applied

(a) Random Forest (RF)

- Ensemble of decision trees.
- Captures **nonlinear relationships** between EEG features and sleep stages.
- Works well with mixed features (PSD, ratios, entropy).
- Provides **feature importance**, e.g., delta power might dominate N3 detection.

(b) XGBoost

- Gradient boosting algorithm (tree-based).
- Strong **accuracy and generalization**.
- Focuses on **hard-to-classify epochs**, refining predictions iteratively.
- Often outperforms RF in structured data tasks.

(c) SVM (Support Vector Machine)

- Maps feature vectors to a high-dimensional space to separate classes.
- Works well for **small to medium-sized datasets**.
- Uses kernels (linear, RBF) to capture nonlinear EEG patterns.

All three are **classic supervised classifiers** suitable for EEG feature-based sleep stage prediction.

---

6. Pipeline Overview

1. **Load EEG & metadata** → check sampling rate, channels, artifacts.
2. **Preprocess signals** → filter, remove artifacts, segment into epochs.
3. **Feature extraction per epoch** → band powers, ratios, energy, entropy, etc.
4. **Assign sleep stage labels** → Wake, N1, N2, N3, REM.

5. **Train classifiers**:
    - Random Forest → fast, interpretable
    - XGBoost → high accuracy, handles complex patterns
    - SVM → effective with smaller datasets or clean feature space
6. **Predict & evaluate** → compare accuracy, F1-score, confusion matrix.

---

7. Scientific Justification

- **Delta power ↑** → deep sleep (N3)
- **Theta & alpha patterns** → light sleep (N1, N2)
- **Eye movements & EMG** → REM detection
- **Machine learning models** learn **patterns across multiple channels and features** simultaneously, improving automated sleep staging.
- **RF & XGBoost** → capture nonlinear combinations of features, robust to noise.
- **SVM** → good baseline for EEG feature classification, especially with smaller datasets.

Dream_text-generation.ipynb

1. EEG Feature Extraction

- Reads the **EDF file** using pyedflib.
- Takes the **first EEG channel** and calculates **Power Spectral Density (PSD)** with Welch's method.
- Computes **band powers** for delta, theta, alpha, beta, and gamma.
- Combines the band powers to generate a **seed value** that will control the GPT-2 generation reproducibly.

Purpose: Translating brain activity into numerical features and a seed for narrative generation.

---

2. GPT-2 Dream Story Generation

- Uses transformers pipeline with **GPT-2** (PyTorch backend).
- Applies the **EEG-derived seed** to make story generation deterministic.
- Creates a **dream narrative prompt** that includes EEG band powers.
- Generates a **short story** starting with *"Once upon a dream…"* that reflects the sleeper's brain activity.

Purpose: Convert numerical EEG patterns into a **human-readable dream summary**.

---

3. Scene-Based Video Preparation

- Splits the dream story into **scenes** (by splitting sentences).
- Fetches **images for each scene** from Pexels using their API.
- Fallback: black background if no image found.
- Adds **text overlay** of the scene description using Pillow.

Purpose: Visual representation of the dream, scene by scene.

---

4. Text-to-Speech (TTS)

- Converts each scene's text into **audio** using gTTS.
- Generates separate MP3 files for each scene.

Purpose: Narration of the dream, synchronized with visuals.

---

5. Video Compilation

- Creates a **video file** using OpenCV:
  - Resolution: 1280x720
  - FPS: 24
  - Each scene lasts **duration_per_scene** (5 sec × 24 frames = 120 frames per scene)
- Writes frames with **image + text overlay**.
- Can later combine scene audio for final dream video.

Purpose: Final multimedia output: a **dream video** with visuals + narration reflecting EEG-derived brain activity.

Final.py

1. **Defining EEG frequency bands**

Different brainwave frequency bands are defined:

- **Delta**: 0.5–4 Hz (deep sleep)
- **Theta**: 4–8 Hz (drowsiness, light sleep)
- **Alpha**: 8–12 Hz (relaxation, calmness)
- **Beta**: 12–30 Hz (active thinking)
- **Gamma**: 30–45 Hz (high-level cognitive activity)s

These bands are later used to extract features from EEG signals.

---

2. **Loading and defining the emotion model**

- An LSTM-based neural network is used for **emotion prediction** from EEG.
- The model has **two LSTM layers** and a final fully connected layer to output probabilities for three emotions: Happy, Fear, Neutral.
- The model weights are loaded from a file, ready for inference.

---

3. **Loading classical ML models**

- Another model, a **Random Forest**, is loaded along with a **scaler** and **label encoder**.
- This model is used for **sleep stage classification** based on EEG features.

---

4. **Setting up MongoDB and encryption**

- A connection to **MongoDB** is created to save user data and EEG results.

- **AES encryption** is applied to user information to securely store sensitive data like name, age, and gender.

---

5. **EEG data loading and preprocessing**

- Uploaded EEG files (EDF format) are read and converted into arrays.
- The raw EEG signals are filtered to remove noise and only the EEG channels are selected.
- Epoching is done by splitting the data into segments (e.g., 30 seconds each) for feature extraction.

---

6. **Feature extraction**

- For each EEG epoch, two kinds of features are computed:
    1. **Frequency-domain features**: Relative power in delta, theta, alpha, beta, and gamma bands.
    2. **Time-domain features**: Measures like variance, mobility, complexity, and entropy.

These features are used by the Random Forest for sleep stage classification.

---

7. **Emotion prediction**

- A small segment of the EEG is scaled and passed to the LSTM model.
- The model outputs **probabilities for each emotion**.
- The predicted emotion is selected as the one with the highest probability.
- A text description explains what the emotion means in terms of brain activity.

---

8. **Dream feature extraction**

- EEG data is analyzed to compute overall statistics: mean, standard deviation, and band power in 1–30 Hz.
- These features are then converted into a **short dream description** using simple text formatting.

---

9. Dream text generation

- Brainwave power values are used to **seed a text-generation model** (GPT-2) so that the generated dream story is reproducible and related to EEG patterns.
- A short story is created describing the dream.

---

10. Tab layout in the Streamlit app

The app shows **six tabs**:

1. **Overview**: EEG summary, number of channels, duration, heatmap visualization.
2. **Emotion Analysis**: Predicted emotion, probabilities, and description.
3. **Sleep Stages**: Sleep stage prediction per epoch with line plot and table.
4. **Dream Decoding**: Numerical dream features and textual interpretation.
5. **Music & AR/VR**: Saves EEG-derived metadata for AR/VR applications.
6. **Dream & Video**: Generates a video of the dream with:
    - Scene images fetched online.
    - Overlay of scene text.
    - Text-to-speech audio for each scene.

---

11. Dream video generation

- Each sentence of the dream is treated as a “scene”.
- Images are fetched from the web to match the scene description.
- Scene text is drawn on top of the image.
- Multiple frames are written per scene to create a short video.
- Each scene also gets a text-to-speech audio file.
- The video and audio are combined and displayed in the app.

---

12. Saving results

- Emotion, probabilities, dream features, and generated text are saved in MongoDB.
- User information is encrypted before saving.