Scenario 1: Health Sensing

Scenario 1 - Health Sensing [25 Marks]

You have recently joined a healthcare startup called DeepMedico™ as a data scientist. Your team is working on detecting breathing irregularities that occur during sleep. As part of a pilot study, DeepMedico™ has collected overnight sleep data (8 hours per participant) from 5 subjects. Firm has provided you with the below Dataset:

For each participant, the following physiological signals are available:

- Nasal Airflow (sampled at 32 Hz)
- Thoracic Movement (sampled at 32 Hz)
- SpO2 (Oxygen Saturation) (sampled at 4 Hz)

In addition to these signals, the dataset also includes:

- An event file that contains annotations for breathing irregularities (e.g., apneas, hypopneas)
- A sleep profile file that records sleep stages

Your task is to analyze and model this data to detect abnormal breathing patterns during sleep.

Understanding the Data and Visualization [3 Marks]

Your manager would like to explore how the recorded signals look across the 8-hour sleep sessions. Your task is to create clear visualizations of the data for each participant, making it easier to interpret patterns and irregularities.

Specifically, you are expected to:

- Plot the Nasal Airflow, Thoracic Movement, and SpO2 signals over the entire 8-hour duration.

- Overlay the annotated flow events (e.g., apneas, hypopneas) on top of the corresponding signal plots for visual reference.
- Export the visualizations in PDF format, as strictly requested by your manager.

You should write a Python script that can generate these visualizations for any participant.

Hints

- The respiration signals and SpO2 signals have different sampling rates (32 Hz vs 4 Hz). Fortunately, all signals include timestamps. You can treat them as time series and align them accordingly.
- Consider using Pandas for handling and aligning the time-indexed data. Making good use of timestamps will be essential to synchronize and plot the signals accurately.

Deliverable

Write a Python script named `vis.py` that accepts a folder path as input and generates a PDF visualization for that participant.

Example usage:

```
python vis.py -name "Data/AP28"
```

Here, AP28 is the folder containing the signal files for one participant. The script should generate a PDF visualization and store it in a directory called `Visualizations`.

Data Cleaning [4 Marks]

While reviewing the visualizations, one of your teammates notices that parts of the signal appear quite noisy. The team member responsible for data collection suggests that this could be due to participant movement during sleep, which may introduce high-frequency artifacts. Your task is to clean the raw signals by filtering out this high-frequency noise, making the data more suitable for further analysis and modeling.

Hints

- Human breathing typically occurs at a rate of 10 to 24 breaths per minute (BrPM), which corresponds to a frequency range of approximately 0.17 Hz to 0.4 Hz.
- Signals containing frequency components significantly higher than this range are likely noise.

- Explore digital filtering techniques such as filters to retain only the relevant breathing frequency range. Consider using libraries such as SciPy, NumPy, or PyWavelets to implement your filtering pipeline.

Dataset Creation [8 Marks]

You are now responsible for creating a dataset from the preprocessed signal data. The goal is to break down the continuous 8-hour-long recordings into smaller windows that can be used for training machine learning models. You are expected to perform the tasks below:

- Split the signals into 30-second windows with 50% overlap.
- Use the event file to determine the label for each window:
  - If a window overlaps by more than 50% with a labeled event (e.g., Hypopnea or Obstructive Apnea), assign that event's label to the window.
  - If there is no sufficient overlap with any event, label the window as Normal.
  - Only the following labels should be considered: Hypopnea, Obstructive Apnea, and Normal.

Deliverable

Write a Python script named `create_dataset.py` that:

- Reads the input signals and annotations from the `Data` directory.
- Processes the data into labeled windows.
- Saves the resulting dataset to the `Dataset` directory.

Think carefully about the file format you use for saving the dataset. Options include CSV (easy to inspect and share), Pickle (Python-native, good for complex objects), Parquet or TS (Time Series formats) (efficient for large time-indexed data), etc. You would be asked why you chose the used format. Choose the format that best suits your needs in terms of efficiency, usability, and compatibility with modeling tools.

Example usage:

```
python create_dataset.py -in_dir "Data" -out_dir "Dataset"
```

Modeling [10 Marks]

Due to unforeseen circumstances, your teammate who was in charge of modeling is unavailable. Your manager has now assigned this task to you, given your familiarity with the dataset you created. Your objective is to train models to classify breathing irregularities based on the labeled 30-second windows. You are required to experiment with the following architectures:

- 1D Convolutional Neural Network (1D CNN)
- 1D Conv-LSTM

Evaluation Strategy

You must evaluate your models using Leave-One-Participant-Out Cross-Validation:

- In each fold, train on the data from 4 participants and test on the remaining one.
- Repeat this process for all 5 participants, ensuring that every participant is used as the test subject exactly once.
- For each fold, report the performance metrics per class, and also compute the mean and standard deviation across folds.
- Why not use a random 80-20 train-test split? Look into the implications of data leakage and why subject-wise validation is preferred when dealing with physiological or personalized data.

Metrics to Report (per class)

- Accuracy
- Precision
- Recall
- Sensitivity
- Specificity
- Confusion Matrix

Clearly report these metrics for each model and fold, and provide aggregated results at the end.

Unique Opportunity for Raise! [Bonus 5 marks]

Your manager has offered a significant incentive if you are able to go beyond breathing irregularity detection and also classify sleep stages using the available

data. If you're up for the challenge, use the sleep profile file, which contains time-aligned annotations for different sleep stages (e.g., Wake, REM, N1, N2, N3), to create a new labeled dataset.

Your task will involve:

- Replacing the breathing event labels with sleep stage labels in your existing 30-second window framework.
- Ensuring each window is assigned the correct sleep stage label based on overlap with the annotations in the sleep profile file.
- Training and evaluating models to classify sleep stages using the same architectures as before (1D CNN, Conv-LSTM, Transformers).

Submission Instructions

Assignment Deliverables

You are expected to submit a GitHub repository containing your complete project for Scenario 1: Health Sensing. Below is an example of the recommended directory structure to organize your work:

```
Project Root/
├── Data/
│   ├── AP01/
│   │   ├── nasal_airflow.txt
│   │   ├── thoracic_movement.txt
│   │   ├── spo2.txt
│   │   ├── flow_events.csv
│   │   ├── sleep_profile.csv
│   ├── ...
├── Visualizations/
├── Dataset/
├── vis.py
├── create_dataset.py
```

Note: This structure is only an example. You may organize your repository differently if you prefer, as long as your code, data, and deliverables are logically arranged and easy to navigate.

Google Form

You are expected to fill the given Submission Form after completion of your assignment. You need to make a GitHub repository for Scenario-1: Health Sensing.

Frequently Asked Questions (FAQs)