Introduction:

Stress and mental health monitoring is a highly emerging application area for machine learning interfacing with wearable devices. Wearables that are enhanced with ML models like this one can be more effective for continuous monitoring of daily stress, detecting burnout symptoms early, or catching anxiety or PTSD symptoms. Stress-focused wearables are especially critical for students in today's day and age, as a robust tool like this one can help promote mental well-being and take steps towards preventive mental health care.

My project proposal is to develop a binary classifier model that detects psychological stress using photoplethysmography (PPG) sensor data, a sensor commonly used in wearables. A PPG uses light to measure blood volume changes in tissues to track physiological parameters like heart rate and blood pressure. By leveraging machine learning, I intend to build a model that can distinguish between stressed versus non-stressed physiological states.

Data Sources:

https://www.kaggle.com/datasets/chtalhaanwar/mental-stress-ppg

I have found a dataset on Kaggle that is best suited for my project. This dataset is from a study including 15 males and 12 females with a mean age of 21. They are all bachelor's students, which is best for my focus on college students for my problem. The data is collected from a PPG sensor that was placed behind the ear for all of these students. The method that they were evaluated between stressed and not stressed is by evaluating them using the Stroop test, which can help measure stress because it places the participants under high pressure. The dataset came labeled as stress and nonstress episodes, which allows me to better visualize that information.

Methodology

My methodology will follow a structured multi step process from raw data to a trained predictive model:

1. Data Handling
    a. I will handle the data using Pandas and NumPy to efficiently conduct analysis, followed by visualization with Matplotlib and Seaborn to identify redundant features and evaluate correlations before preprocessing.
2. Data Exploration & Visualization
    a. Basic features of the data, like reading in the shape and showing it on a time-series plot, can be useful before processing the data. Data will be visualized using Matplotlib and Seaborn before conducting preprocessing. This will allow me to identify redundant features and visualize correlation matrices to evaluate what features are most valuable to the model.
3. Data Preprocessing

    a. First, there will need to be some feature engineering to get HRV data. Time-domain features include minimum, maximum, median, mean, standard deviation, variance, skew, and kurtosis. For frequency domain features, this will require a Fast-Fourier Transform to be applied to the time-domain data values to extract values like power spectral density.
    b. For this dataset, all of the signals are sampled at the same frequency and are already windowed, so I don't have to worry about that.
    c. Next, I will treat missing data and remove fields if they are null, and then scale my data using StandardScaler. Once all of that is done, it is ready for a final train test split, and I will utilize an 80-20 split to validate my data.
4. Model Algorithm & Fine Tuning
    a. I want to experiment with different model algorithms to see what yields the best results. As per my research, SVM and RF have been used in research papers before to accurately classify PPG data. Classical machine learning techniques will be effective as long as proper feature engineering is done.
5. Data Results & Cross Validation
    a. Cross-validation of the models using testing data is necessary to fine-tune the hyperparameters of the model. By finding the right parameters, I will be able to yield a more accurate model. I will visualize the accuracy also through confusion matrices. This will allow me to see whether the model needs to improve in terms of precision or recall. In this scenario, it is more important to me to minimize false negatives, because not identifying stress in a stressful situation is more detrimental to the user.

Products to be Delivered

The result that I am going to deliver is a binary classification model that will identify stressed and non-stressed activity levels. I will aim to use classical machine learning models, mostly SVM and RF. I will check for over and underfitting by doing an appropriate train-test split and validating the model on the testing data. I will analyze the training error and testing error and make changes to the model accordingly. This might be through adjusting hyperparameters or using different models entirely.

Eventually, I want this model to be applicable to other datasets and ensure that it is generalizable.