

ECG Analysis and Stress Classification

Federico Cesare Cattò

Gennaio 2026

Contents

1	Dataset Loading Exploratory Analysis	2
1.1	ECG Analysis and Stress Classification	2
2	Signal preprocessing and segmentation	4
2.1	ECG Signal Preprocessing	4
2.2	ECG Segmentation and Window Labeling	4
3	Feature Extraction	6
4	Dataset construction and scaling	7
5	Model Training and Evaluation	8
6	Conclusions	10

Chapter 1

Dataset Loading Exploratory Analysis

1.1 ECG Analysis and Stress Classification

The objective of the first task is to analyze electrocardiogram (ECG) signals and perform stress classification using classical machine learning techniques. The ECG data were obtained from a public PhysioNet dataset, which provides multimodal physiological recordings collected from multiple subjects under different stress conditions.

In this project, ECG signals from fifteen different subjects were considered. Each subject's data were stored in individual files and processed independently, allowing the analysis to capture inter-subject variability and to avoid subject-specific bias. The ECG signal was acquired from a chest-mounted sensor and sampled at a frequency of 700 Hz, ensuring high temporal resolution and preserving detailed cardiac dynamics.

Each recording is associated with a sequence of labels indicating the subject's physiological state over time. To simplify the classification task and focus on stress detection, the original labels were converted into a binary representation. In particular, samples corresponding to stress conditions were assigned to the positive class, while all remaining samples were grouped into the non-stress class. This formulation transforms the problem into a binary classification task, which is well suited for classical machine learning models and facilitates the interpretation of the results.

After loading the data, an initial exploratory analysis was performed on the raw ECG signals. The ECG waveform was first visualized over the entire recording duration to assess signal quality, amplitude variability, and the presence of potential artifacts. Subsequently, stress-related samples were highlighted directly on the ECG signal, allowing a clear inspection of how stress periods are distributed over time and how they relate to changes in the cardiac signal. Finally, the ECG signal was plotted using a time axis expressed in seconds, improving interpretability and providing a more intuitive understanding of the temporal structure of the recordings.

This preliminary analysis plays a crucial role in validating the correctness of the data loading procedure, verifying the alignment between ECG samples and stress labels, and motivating the subsequent steps of signal preprocessing and feature extraction.

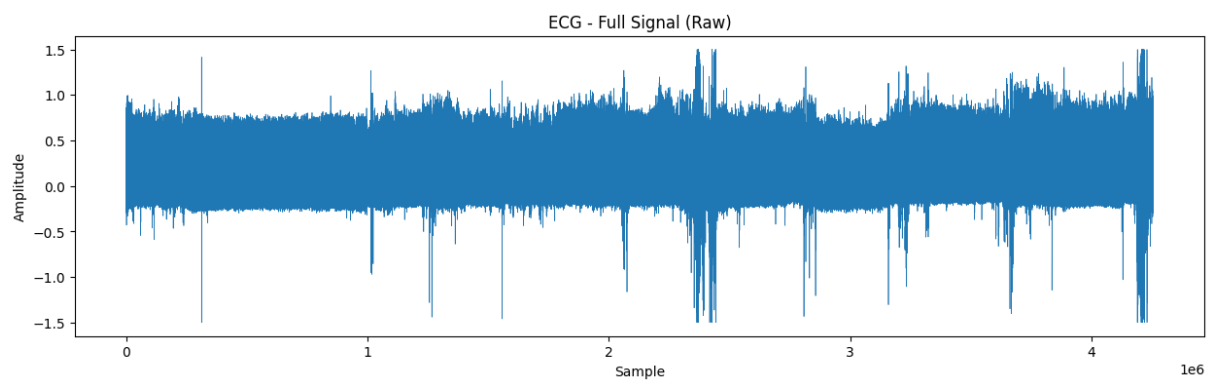


Figure 1.1: Full raw ECG signal for a single subject.

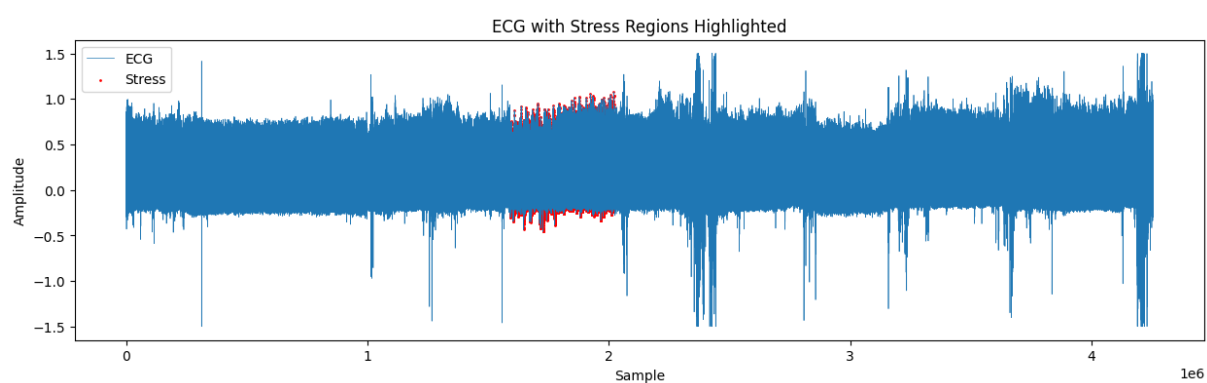


Figure 1.2: ECG signal with stress regions highlighted in red.

Chapter 2

Signal preprocessing and segmentation

2.1 ECG Signal Preprocessing

Before extracting features and training machine learning models, the raw ECG signals were subjected to a dedicated preprocessing pipeline aimed at improving signal quality and reducing inter-subject variability.

First, each ECG recording was standardized using z-score normalization. This step ensures that the signal has zero mean and unit variance, mitigating amplitude differences across subjects and recording sessions. Such normalization is particularly important when combining data from multiple individuals and applying classical machine learning algorithms.

Subsequently, a band-pass filter was applied to the standardized ECG signal. The filter was designed to retain frequencies between 0.5 Hz and 40 Hz, which correspond to the physiologically relevant components of the ECG waveform. Lower frequencies, typically associated with baseline wander, and higher frequencies, often related to noise and muscle artifacts, were effectively attenuated. To avoid phase distortion and preserve the temporal structure of the signal, zero-phase filtering was performed using a forward-backward filtering approach.

The effectiveness of the preprocessing pipeline was qualitatively assessed by visually comparing the standardized and filtered ECG signals. This comparison highlights the removal of noise while preserving the characteristic morphology of the cardiac waveform.

2.2 ECG Segmentation and Window Labeling

After preprocessing, the continuous ECG signals were segmented into fixed-length windows to enable feature extraction and supervised learning. A sliding window approach was adopted, using windows of 10 seconds (7000 samples) with a 50% overlap between consecutive segments. This configuration provides a good trade-off between temporal resolution and the ability to capture meaningful cardiac patterns related to stress.

Each window was assigned a binary label based on the stress annotations of the samples it contained. Specifically, a majority voting strategy was used: a window was labeled as “stress” if more than half of its samples corresponded to stress conditions; otherwise, it was labeled as “no stress”. This approach increases robustness to short-term fluctuations and potential label noise.

The segmentation process was performed independently for each subject, resulting in a set of labeled ECG windows per patient. This design choice helps prevent data leakage

and ensures that the subsequent machine learning models are trained on well-defined, physiologically meaningful segments.

Chapter 3

Feature Extraction

For each segmented ECG window, a comprehensive set of features was extracted to characterize the signal from multiple complementary perspectives. The feature extraction strategy combines time-domain, frequency-domain, and heart rate variability (HRV) descriptors, enabling the machine learning models to capture both morphological and physiological aspects of cardiac activity.

Time-domain features were computed directly from the ECG waveform to summarize its statistical properties and temporal dynamics. These include basic statistical measures such as mean, standard deviation, root mean square, and peak-to-peak amplitude, which describe signal amplitude and variability. Additional descriptors, such as zero-crossing rate and signal energy, were included to capture waveform fluctuations. Furthermore, Hjorth parameters (activity, mobility, and complexity) were computed to quantify signal variance, frequency content, and structural complexity in a compact and computationally efficient manner.

Frequency-domain features were extracted using Welch’s method to estimate the power spectral density of each ECG window. From the estimated spectrum, power was integrated over the low-frequency (LF: 0.04–0.15 Hz) and high-frequency (HF: 0.15–0.40 Hz) bands. These components are commonly associated with autonomic nervous system activity. In addition, the LF/HF ratio was computed as an indicator of the balance between sympathetic and parasympathetic modulation.

To further enhance physiological relevance, heart rate variability features were extracted by detecting R-peaks within each ECG window. Based on the resulting RR intervals, standard HRV metrics were computed, including the standard deviation of NN intervals (SDNN), the root mean square of successive differences (RMSSD), the proportion of interval differences greater than 50 ms (pNN50), and the mean heart rate. These features provide insight into cardiac rhythm regulation and are particularly informative for stress-related analysis.

All extracted features were concatenated into a single feature vector for each ECG window, resulting in a compact yet expressive representation suitable for classical machine learning models. Feature extraction was performed independently for each subject, preserving the structure of the dataset and preparing it for subsequent classification and evaluation.

Chapter 4

Dataset construction and scaling

After feature extraction, the dataset was organized to ensure a reliable and realistic evaluation of the stress classification models. In particular, the data were split into training and test sets at the subject level. ECG windows from twelve subjects were used for training, while the remaining subjects were reserved exclusively for testing. This subject-wise split prevents data leakage and allows the evaluation of model generalization across unseen individuals, which is a critical requirement in physiological signal analysis.

Feature vectors extracted from all windows belonging to the training subjects were concatenated to form the training set, while windows from the test subjects were similarly aggregated into the test set. The resulting dataset consists of approximately 14,000 training samples and 3,300 test samples, with each sample represented by a 16-dimensional feature vector.

An inspection of the class distribution revealed an imbalance between stress and non-stress samples, with the non-stress class being more prevalent. This reflects the natural distribution of physiological states in real-world recordings and was taken into account during model evaluation.

Before training the classifiers, feature scaling was applied using standardization. A standard scaler was fitted on the training data to transform features to zero mean and unit variance, and the same transformation was applied to the test data. This step ensures that all features contribute equally to the learning process and improves the numerical stability and performance of classical machine learning models.

Dataset	Samples	Stress	No Stress	Features
Train	13,995	1,576	12,419	16
Test	3,357	418	2,939	16

Table 4.1: Summary of the dataset split and feature dimensions.

Chapter 5

Model Training and Evaluation

Several strategies were employed to train support vector classifiers (SVC) for stress detection from ECG-derived features, with particular attention to handling class imbalance. The training set exhibits a strong imbalance between non-stress and stress samples, reflecting the natural distribution of physiological states.

1. SMOTE oversampling

Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the training set by generating synthetic samples of the minority (stress) class. A support vector classifier was trained using grid search with cross-validation to optimize the kernel type, regularization parameter C , and kernel coefficient γ . The best-performing model utilized an RBF kernel with $C = 10$ and $\gamma = \text{scale}$, achieving a cross-validated macro F1 score of 0.82. On the test set, this approach provided a good compromise between precision and recall for both classes, particularly improving recall for the stress class.

2. Undersampling strategies

Several undersampling techniques were tested, including random undersampling and variants of the NearMiss algorithm. Each method reduced the majority class to balance the dataset, and SVC hyperparameters were optimized via grid search. While undersampling often improved recall for the stress class, it typically resulted in lower precision and overall F1 score, highlighting the sensitivity of these methods to the removal of majority-class samples.

3. Class-weighted SVC

An alternative approach used the original training set with the SVC `class_weight='balanced'` option to account for the imbalance. This method achieved a macro F1 of 0.68, with high recall for the stress class (0.88), demonstrating that reweighting the loss function is a simple and effective way to address imbalance without generating synthetic data.

4. Bagging SVC

Finally, ensemble learning was explored using a bagging classifier with SVC as the base estimator. This approach improves model stability and reduces variance. However, while bagging slightly increased robustness, the macro F1 score (~ 0.615) did not surpass the performance of SMOTE-based oversampling.

Overall, SMOTE oversampling combined with RBF SVC was chosen as the preferred strategy for this dataset. While NearMiss_v3 achieved higher stress-class recall and

precision on this specific test set, SMOTE provides a better balance between minority-class detection, overall F1, and generalization across unseen patients. These results highlight the importance of carefully handling class imbalance while maintaining robustness in physiological signal classification tasks. SMOTE was preferred over Bagging or class-weighted SVC because it creates synthetic minority samples, enriching the diversity of stress windows and improving the SVC’s ability to generalize across unseen patients. Bagging and class weighting improve stability and recall, respectively, but do not expose the model to new minority patterns, which is crucial in physiological datasets with high inter-subject variability.

Method	Macro F1	Accuracy	Stress Precision	Stress Recall
SMOTE	0.67	0.81	0.35	0.66
RandomUnderSampler	0.64	0.73	0.31	0.97
NearMiss_v3	0.73	0.84	0.43	0.81
Class-weighted	0.68	0.78	0.35	0.88
Bagging	0.69	0.79	0.36	0.88

Table 5.1: Comparison of different training strategies for stress classification using SVC.

Chapter 6

Conclusions

In this chapter, we presented a comprehensive workflow for stress classification from ECG signals using classical machine learning techniques. Starting from the raw physiological recordings of multiple subjects, we implemented a robust preprocessing pipeline including z-score normalization and band-pass filtering, followed by segmentation into fixed-length windows and feature extraction encompassing time-domain, frequency-domain, and HRV descriptors.

The dataset was carefully constructed with a subject-wise split to prevent data leakage and ensure realistic evaluation of model generalization. Class imbalance, a natural characteristic of physiological stress data, was addressed through different strategies, including SMOTE oversampling, undersampling, class-weighting, and bagging.

Among the tested approaches, SMOTE combined with an RBF SVC provided the best compromise between stress-class detection and overall classification performance, achieving a macro F1 score of 0.67 and an accuracy of 0.81 on the test set. While undersampling and class-weighted strategies improved recall for the stress class, they often sacrificed overall F1, highlighting the trade-offs inherent in handling imbalanced physiological datasets.

Overall, the results demonstrate that classical machine learning models, when coupled with carefully engineered features and appropriate imbalance handling, can effectively detect stress from ECG signals. However, the performance is limited by inter-subject variability and the complexity of stress patterns, suggesting that future work could explore advanced approaches such as deep learning models capable of learning directly from raw ECG signals, or multimodal fusion with additional physiological signals to further enhance stress detection accuracy.