Technical University of Crete
School of Electrical and Computer Engineering

Classification and Dimensionality Reduction Methods in Machine Learning Problems

Chivintar Amenty

For the MLDS postgraduate course: Machine Learning
Instructors: Vasileios Digalakis, Eleutherios Dermitzakis

June 5, 2025

Contents

1 Introduction

This project explores some fundamental Machine Learning (ML) classification methods with the Python programming language. The study is done on sleep data, namely drawn from the Sleep EDF Database Expanded [1] as accessed on April 16th, 2025. This document is complemented by the Python code which can be found on the public GitHub repository [2]. The structure includes the necessary theoretical background followed by the specifics of the experimental setup, implementation, and results. The assignment is concluded with a discussion section and conclusions.

2 Background

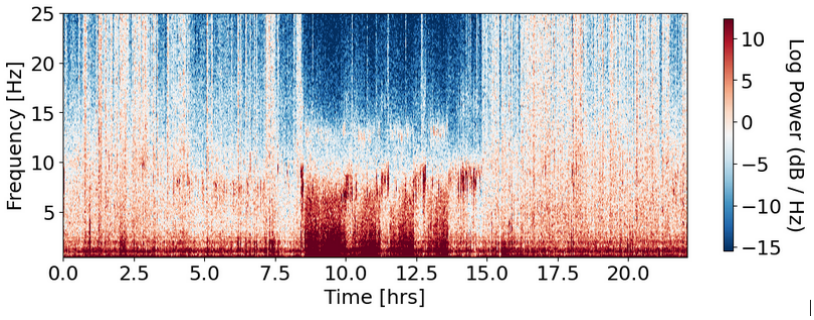

Figure 1: A 24-hour spectrogram of brain activity. The sleeping patterns are observed from T=8 to T=15Hrs.

2.1 Sleep stages

While we sleep, we undergo an orchestrated sequence of states. In healthy individuals, this can be mapped into a series of distinct stages throughout the sleeping session. We know (Table 1) that the first stage of sleep, referred to as **N1**, is a transitional phase characterized by light sleep, a gradual reduction in muscle activity and eye movement. This is followed by **N2**, during which brain activity further slows down and exceptional patterns appear in the electrograph known as spindles. N3, often combined with N4 in some classification schemes which this study follows, form **N3/4**, which represents deep sleep; this stage is dominated by slow frequency waves and is considered the most "restorative". Finally, **REM** (Rapid Eye Movement) sleep, as the name suggests, is characterized by rapid eye movements, vivid dreaming, low-amplitude and mixed-frequency EEG activity, something that resembles the desynchronized patterns of wakefulness. These stages cycle multiple times per night in a predictable pattern (Fig. 1), and their identification plays a crucial role in both clinical diagnostics (e.g. epileptic diagnoses) and sleep research [3].

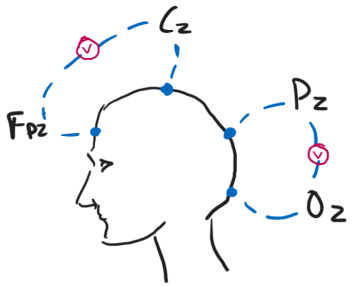

Figure 2: Conceptual sketch of EEG Fpz-Cz and Pz-Oz channels. For acronym elaboration: **Fpz**: Frontal pole, positioned at the very front of the head. **Cz**: Central midline.**Pz**: Parietal midline. **Oz**: Occipital midline.

Sleep Stage	EEG Characteristics
Wake	**More than 50% of alpha rhythm (8–13 Hz)** **Beta rhythm (13–30 Hz)**
N1	**Vertex sharp waves (5–14 Hz)** **Low amplitude, mixed frequency activity (4–7 Hz)** Less than 50% of alpha rhythm (8–13 Hz)
N2	**K complex (8–16 Hz)** **Sleep spindle (12–14 Hz)** Low amplitude, mixed frequency activity (4–7 Hz) Less than 50% of alpha rhythm (8–13 Hz)
N3/4	**More than 20% of slow wave activity (0.5–2 Hz)** Sleep spindle (12–14 Hz)
REM	**Sawtooth waves (2–6 Hz)** **Low amplitude, mixed frequency activity (4–7 Hz)** Alpha rhythm (8–13 Hz) K complex (8–16 Hz) Sleep spindle (12–14 Hz)

Table 1: EEG characteristics of each sleep stage according to the American Academy of Sleep Medicine (AASM) rule [5]. Bold indicates the fundamental rationale that scores the corresponding sleep stage.

2.2 Electroencephalography (EEG) data

EEG is a high-temporal-resolution technique used to study neurocognitive processes [4]. The method is conducted by placing electrodes on the scalp of a living being and recording the electrical activity of its brain. The recording can span multiple "channels" each of which represent relative measurements, i.e. the change in the measured electrical potential between that electrode and a reference electrode placed somewhere else on the head. Typical scale of measurements are between 1-200 microvolts.

EEG is widely used in sleep research to study brain dynamics across different sleep stages, as they exhibit characteristic frequencies (or rhythms) in EEG signals. These distinct stages —Awake, N1, N2, N3/N4, and REM— can be identified based on the spectral composition and temporal dynamics of the measurements [5].

Brain rhythms are grouped into bands that are defined by increasing center frequencies and frequency widths. Brain rhythm frequency bands include **delta** (0.5–4 Hz), **theta** (4–8 Hz), **alpha** (8–13 Hz), **beta** (13–30 Hz), lower gamma (30–80 Hz), and upper gamma (80–150 Hz). This assignment, as sleep science in general, focuses on the first four bands. In Figure 1 the emerging patterns can be observed.

Sleep data is high-dimensional, noisy, and temporally correlated, thus requiring preprocessing steps like segmentation into time windows (called epochs).

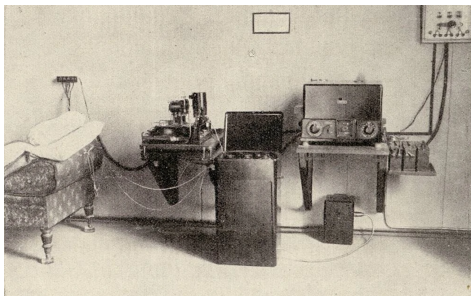

Figure 3: First EEG machine, invented by Hans Berger in 1926 [6].

2.3 Dimensionality Reduction

Principal Component Analysis (PCA)

PCA is one of the most fundamental unsupervised techniques used to reduce the dimensionality of a problem and is essentially a linear transformation of the feature vector to a (reduced in dimension) version of itself. By (i) computing the eigenvectors of the covariance matrix of the data and (ii) ranking them according to their eigenvalues, it projects the data in a lower-dimensional space, the size of which is defined by (iii) the number of eigenvectors we choose to keep.

Linear Discriminant Analysis (LDA)

Similarly to PCA, LDA also looks for linear combinations of variables which best explain the data. However, LDA works when the measurements made on independent variables for each observation are continuous quantities and is used when groups (classes) are known a priori. As such, is a supervised technique which maximises the between-class variance while minimising the within-class variance. LDA assumes normally distributed classes with equal covariance, and, although EEG data might not satisfy these assumptions, this method will prove to give one of the highest accuracies in our case study.

2.4 Classification

Naive Bayes

Bayes classifiers are probabilistic models, which pick their classifying decision so as to minimize the probability of misclassificaton. The Naive Bayes (NB) classifier comes with the simplifying assumption that the features are conditionally independent within a class. Despite this strong independence assumption, this model often performs surprisingly well in high-dimensional settings, as will our case study show too. Some popular variants of Naive Bayes classifiers include the Gaussian, the Multinomial (best for categorical-like data) and the Bernoulli (best for discretized data). In this project, only the Gaussian variant (GNB) was employed due to the continuous nature of our EEG-derived features. GNB assumes that the continuous values associated with each class are distributed according to a Gaussian distribution.

Not So Naive Bayes [7].

While the standard Gaussian Naive Bayes assumes feature independence, a more interesting approach would be the multivariate Gaussian model, which accounts for potential feature correlations by estimating a full covariance matrix. For this reason, we shall explore the generalized form of this probabilistic model, and think of GMMs: Gaussian Mixture Models.

Gaussian Mixture Models

GMMs are probabilistic models that assume data is generated from a mixture of several Gaussian distributions. In supervised GMM classification, a separate Gaussian Model (of one or of a Mixture of Gaussians) is fitted to each class using the labeled training data. For classification, the likelihood of a sample under each class's GMM is computed and the class with the highest likelihood is selected. We apply this method to multiple variants of our dataset (original, PCA-reduced, LDA-reduced) and compare its performance with that of Gaussian Naive Bayes classifiers, noting that, when using one Gaussian component per class with a diagonal covariance matrix, we essentialy are in the case of GNB.

3 Implementation

In sleep data acquisition, well trained technicians score (annotate) one sleeping stage on the electrogram every 30 seconds.

The famous Sleep-EDF Database Expanded [1] contains two types of files: (A) 197 Sleep Cassettes (sleep recordings, SC), and (B) corresponding Hypnograms (sleep stage annotations), manually scored by trained sleep scientists, for a total of 78 subjects (individuals). For each individual both files are used for analysis.

While each SC contains numerous channels, in this project only two EEG are chosen, namely, (1) Fpz-Cz and (2) Oz-Pz (Fig. 2). In the following, the first subject (00) on their first night (1) is thoroughly examined (subject id: SC4 00 1).

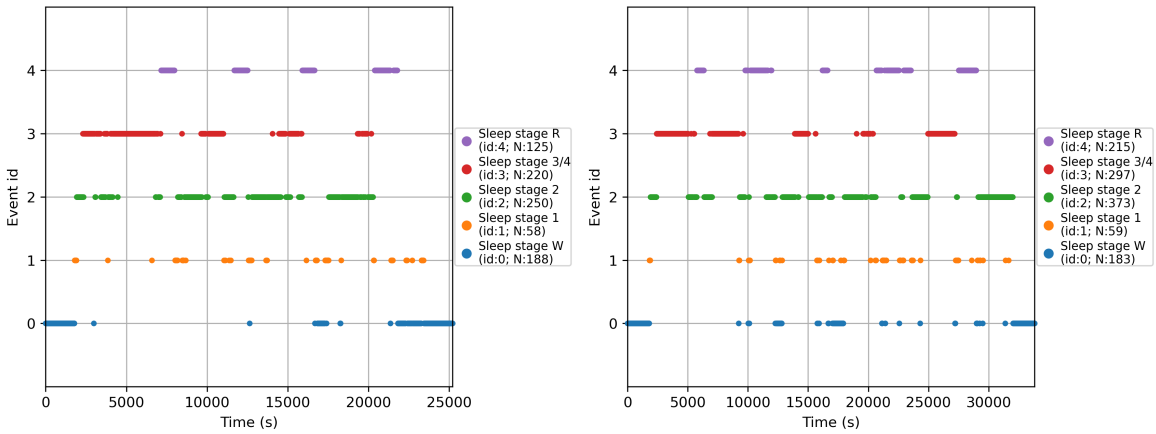

Figure 4: Two different nights' annotated events from the same individual. First night (left) was used in our experiments. Each dot represents a 30-second-epoch annotation of a sleep stage scoring. 30 minutes of wake time before sleep and after sleep were kept to account for class imbalance.

Preprocessing

For the preprocessing of data the domain-specific Python packages `MNE` and `YASA` were used. First step of the preprocessing includes aligning procedures between the two files that correspond to each subject. That is, aligning the *hypnogram* labels (annotations made from experts) (Hypnogram file) with the Sleep Cassette (SC) Polysomnography data (PSG file). Choosing the channels of interest can be done at any point of preprocessing after the file is now loaded as a `MNE` object. Second step of the preprocessing involves cropping the file (`MNE` object) to 30-second windows, that each of them will correspond to an epoch.

Continuing to the third step, we want our data to include only the sleeping period (omitting the many hours of 'Awake' stages that originally exist). Therefore, the `MNE` object is cropped from 30 minutes before sleep onset (first detection of sleeping stage) to 30 minutes after sleep offset (last detection of a sleeping stage). Note: this also ensures less class imbalance.

As such, after this step, our data consists of a number of 841 epochs in the case of the first subject, each with its own sleep stage label (annotation). The final number of the epochs for each individual depends on their total sleeping hours, 900 epochs being for instance 7.5 hours. These epochs will now consist the rows of our dataset.

In Figure 4 each data point corresponds to an epoch. The gradual shift from each stage to the next can also be observed and the numbers of events per class are shown.

Feature extraction

For this examination numerical and statistical quantities are chosen as features. In the time domain, the first four statistical moments as well as rms, peak-to-peak, and the number of zero crossings are selected. In the frequency domain, the bandpower corresponding to the aforementioned four major bands is selected. Basic Python libraries were used for these procedures, including `numpy`, `sklearn` and `scipy`. A particular example being the application of the Welch method with a tapered window for the extraction of bandpower, specifically, using a *Hann* window of 4 seconds (typical duration for sleep stage analysis [1]).

Feature	Description
For each EEG channel (Fpz-Cz and Pz-Oz):	
mean	Mean amplitude of the signal
std	Standard deviation
skewness	Skewness (asymmetry of the signal distribution)
kurtosis	Kurtosis (tailedness of the distribution)
rms	Root mean square
ptp	Peak-to-peak amplitude
zero_crossings	Number of zero crossings
delta_power	Bandpower in delta range (0.5–4 Hz)
theta_power	Bandpower in theta range (4–8 Hz)
alpha_power	Bandpower in alpha range (8–13 Hz)
beta_power	Bandpower in beta range (13–30 Hz)

Table 2: Features extracted per EEG channel (Fpz-Cz and Pz-Oz). Total features: 2 channels $\times$ 11 = 22.

The final feature vector consists of 22 features, 11 for each channel and presented in Table 2. Finally, our first feature matrix consists now of 841 rows (each row is a 30s epoch) and 22 columns.

Dimensionality reduction

As our feature space has a size of 22, we will examine how much can we reduce it while maintaining optimal classification performance. In Figure 5 the explained variance for PCA is shown. In some sense, all of our features (for each channel) seem to have embody important information. For this reason, I chose to examine PCA(n=2) and PCA(n=10) for completeness.
Of course, the use of LDA is highly appropriate to exploit the supervised advantage of this case study. The idea is, if we leverage all classes, that is we take maximum possible component in LDA, will the separation performance be even better? Since our classes are 5, LDA (n=4) was also put on the queue, together with LDA(n=2) for completeness.

4 Experiments

The specific experiments included classification tests of the following six datasets with the GNB and GMM models:

1. Original (normalized) dataset

[1] In order to define the optimal window duration, a commonly used approach is to take a window sufficiently long that encompasses at least two full cycles of the lowest frequency of interest. In our case, the lowest frequency of interest is 0.5Hz, so we choose a window of 2/0.5=4 seconds.

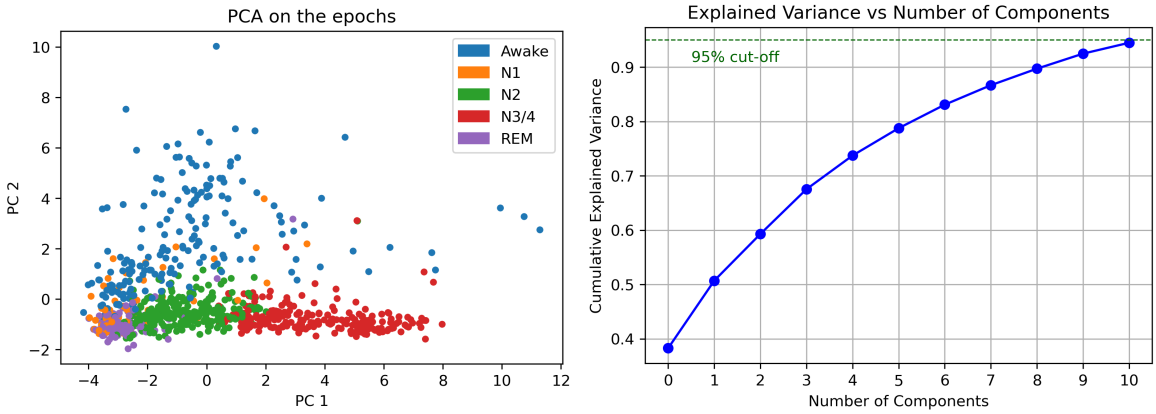

Figure 5: Left: 2D projection of the data using PCA (n=2). Right: Explained variance ratio per component.

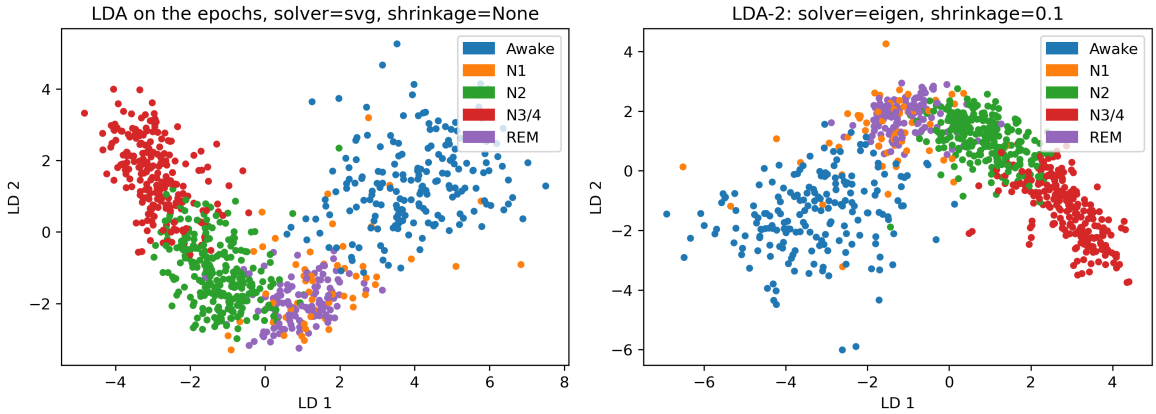

Figure 6: Left: 2D LDA embedding. Right: LDA embedding using eigen solver and shrinkage 0.1.

2. PCA (n=2)
3. PCA (n=10)
4. LDA (n=2) with SVD solver
5. LDA (n=4) with SVD solver
6. LDA (n=4-S) with EVD solver and `shrinkage=0.1`

In Figure 5, with only 2 components, PCA showed good separation of the sleep stages N2, N3/4 and REM. It can be observed that REM is the closest to the wake state, and N3/4 (deep sleep) is the farthest from it. In Figure 6 LDA shows a much more obvious separation in the 2D projection.

The datasets were split to 70-30 train-test ratio, and stratified sampling was used for the testing. Experiments were run on a `random_seed=42` for reproducibility and performance scores used macro average were applicable (F1, Recall).

For the GNB classifier, uniform (uninformative) priors were used to the training of the model, so that the cross-subject testing is as unbiased as possible. For the GMM classifier, one Gaussian was fitted per class.

Tuning

The classifiers used are simple and small, as such only test sets were used to evaluate performance. While no validation sets were used, those could be used for the GMM models in order to tune the hyperparameters (e.g. number of Gaussians per class) and some validation criterion such as Akaike or Bayesian Information Criterion would be taken into account (ideas for future work).

5 Results and Evaluation

Summary tables are depicted below for selected models from a random run, while in Table 3 performance scores were calculated over 50 random runs. All summary tables can be found in the `.ipynb` file [2].

```
------------------------------------------------------------
GNB LDA-4
              precision    recall  f1-score   support

           W       0.96      0.95      0.96        57
          N1       0.50      0.41      0.45        17
          N2       0.87      0.89      0.88        75
        N3/4       0.97      0.92      0.95        66
         REM       0.72      0.82      0.77        38

    accuracy                           0.87       253
   macro avg       0.80      0.80      0.80       253
weighted avg       0.87      0.87      0.87       253

------------------------------------------------------------
```

```
Original: Accuracy = 0.79, Recall = 0.72, F1 = 0.71
PCA-2: Accuracy = 0.75, Recall = 0.67, F1 = 0.64
PCA-10: Accuracy = 0.78, Recall = 0.69, F1 = 0.68
LDA-2: Accuracy = 0.86, Recall = 0.75, F1 = 0.74
LDA-4: Accuracy = 0.87, Recall = 0.80, F1 = 0.80
LDA-4s: Accuracy = 0.85, Recall = 0.74, F1 = 0.73
```

Figure 7: Performance summary for Gaussian Naive Bayes classifier, with uniform priors. Left: Per class performance metrics on LDA(n=4) dataset. Right: Summary on all datasets.

```
------------------------------------------------------------
GMM LDA-4
              precision    recall  f1-score   support

           W       0.93      0.91      0.92        57
          N1       0.30      0.35      0.32        17
          N2       0.91      0.84      0.88        75
        N3/4       0.95      0.95      0.95        66
         REM       0.76      0.84      0.80        38

    accuracy                           0.85       253
   macro avg       0.77      0.78      0.77       253
weighted avg       0.86      0.85      0.86       253

------------------------------------------------------------
```

```
Original: Accuracy = 0.83, Recall = 0.73, F1 = 0.73
PCA-2: Accuracy = 0.72, Recall = 0.67, F1 = 0.66
PCA-10: Accuracy = 0.82, Recall = 0.74, F1 = 0.74
LDA-2: Accuracy = 0.83, Recall = 0.74, F1 = 0.74
LDA-4: Accuracy = 0.85, Recall = 0.78, F1 = 0.77
LDA-4s: Accuracy = 0.84, Recall = 0.75, F1 = 0.75
```

Figure 8: Performance summary for Gaussian Mixture Model classifier, with 1 Gaussian fitted per class (and full covariance matrix). Left: Per class performance metrics on LDA(n=4) dataset. Right: Summary on all datasets.

For the best-performance models, we see the weak spot being the class corresponding to sleep stage N1, while the detection of the deep sleep N3/4 being the most accurate, followed by accurate detection of the Wake state. This phenomenon is predicted when we see the projection on lower dimensions was depicted in 2D. In addition, these results stand with the long known real problem of detecting N1 stages, tackled in numerous scientific studies.

In Figure 9, we can also see the contours of the Gaussian distributions fitted for each class in two standard deviations. Fitting two Gaussian components for each class was also examined, but the results were mostly worse for this instance. And finally, in Figure 10 we can visually see the performance of each model to each dataset.

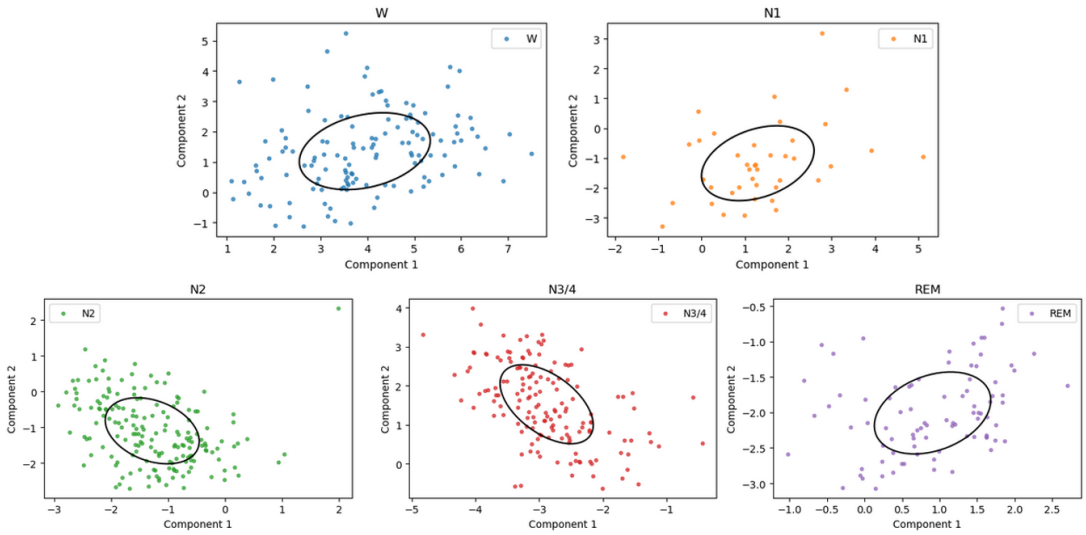

Figure 9: GMM contours on LDA(4) dataset. The ellipses correspond to two standard deviations of the Gaussian of each class.

Dataset	Dim.	Classifier	Accuracy ± 0.02	Recall	F1
Original (scaled)	22	GNB	0.79	0.72	0.71
PCA (n=2)	2	GNB	0.77	0.67	0.65
PCA (n=10)	10	GNB	0.76	0.66	0.64
LDA (n=2)	2	GNB	0.84	0.75	0.74
LDA (n=4)	4	GNB	**0.88**	**0.81**±0.02	**0.81**±0.02
LDA (n=4-S)	4	GNB	0.85	0.75	0.75
Original (scaled)	22	GMM	0.85	0.77	0.76
PCA (n=2)	2	GMM	0.72	0.66	0.65
PCA (n=10)	10	GMM	0.81	0.73	0.72
LDA (n=2)	2	GMM	0.83	0.75	0.75
LDA (n=4)	4	GMM	**0.86**	**0.80**±0.03	**0.79**±0.03
LDA (n=4-S)	4	GMM	0.83	0.75	0.75

Table 3: Comparison over 50 runs of classifiers across dimensionality-reduced datasets, measured in percent (%). GMM was trained with supervision. Class-specific scores can be found in the `.ipynb` file. Errors were suppressed for visibility and ranged from 0.02 to 0.03.

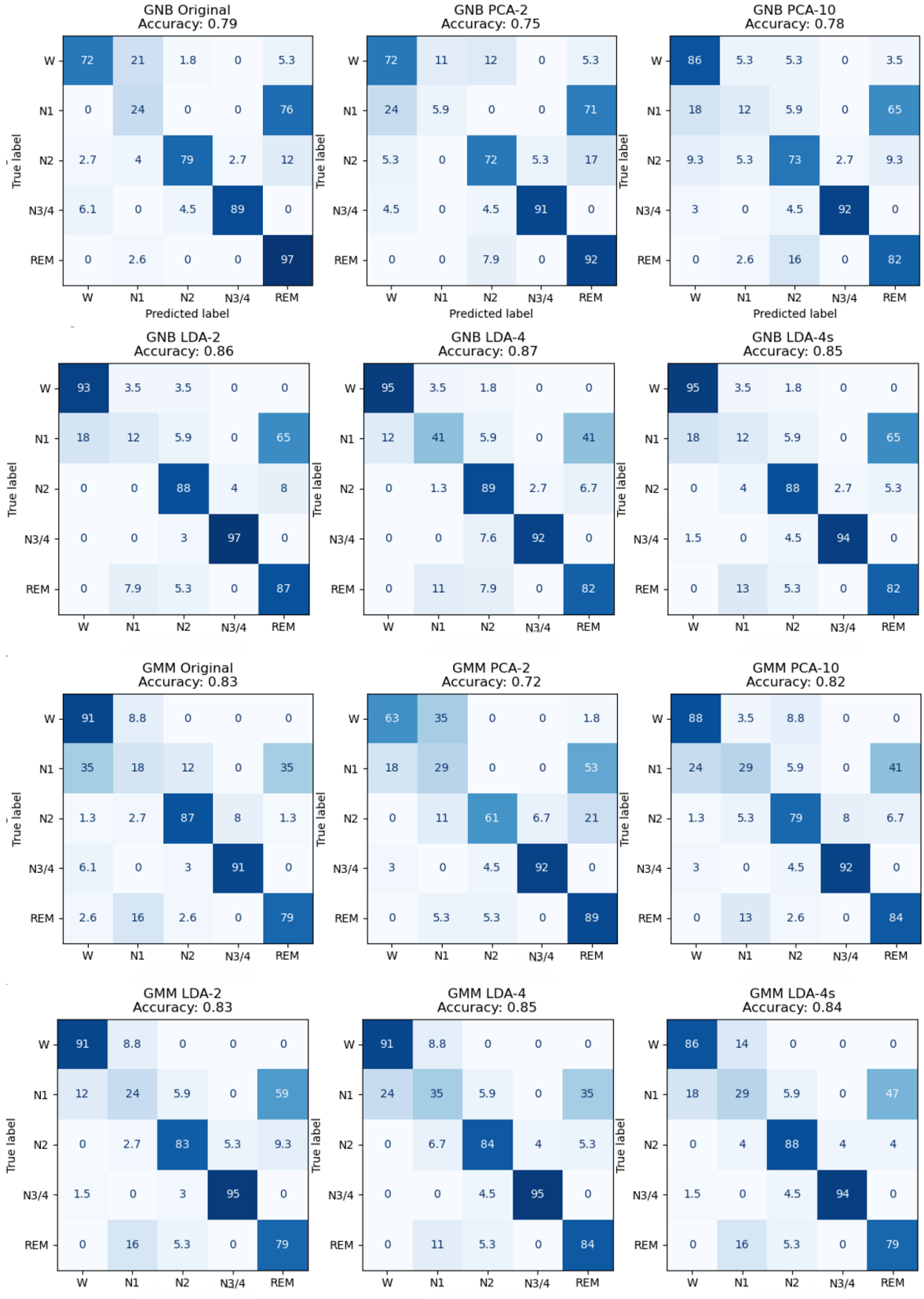

Figure 10: Confusion matrices for of a random run for the Gaussian Mixture Model (one component per class) and the Gaussian Naive Bayes classifiers. Shown in percentage (%). LDA-4 shows an outstanding performance.

6 Discussion

The results show LDA's advantage in supervision, achieving the highest classification accuracy for this subject. PCA, although unsupervised, also provided good compression and visualization, especially on the second subject (not shown here). Gaussian Mixture Models with one component/class showed success depending on the dataset. Mixtures of components did not yield better results on the tested datasets. The winner obviously stands with LDA (n=4) when trained with Gaussian Naive Bayes.

While the elephant in the room wearing a hat that reads "*What about cross subject model evaluation?*" has patiently waited to the very end of the document, it is time to address their logical concern. A less rigorous examination done on another subject (subject id SC4 01 2) showed the challenge that exists in the generalisation of these models. For instance, better performing models (trained on subject 1) came to be PCA-2 (71%) and LDA-2 (72%) on the GNB classifier. It makes some sense, if we think about the natural high variance on these features across individuals; probabilistic model trained on these statistical features would lead to a kind of "subject specific overfitting".

Thus, to improve the performance of the developed classifiers, introducing more features is a natural afterthought. Many sleep classification studies do use more mathematical quantities and information metrics. Entropy measures as well as fractal dimensions very often are used, as there is evidence that sleep stages hold information in these metrics. Additionally, more rigorous pipelines for frequency information extraction would elevate the performance, an example being combining multiple sliding windows of different duration for the extraction of bandpower (multi-resolution feature extraction) [8]. For reference, current state-of-the-art benchmarks are around a level of accuracy $\approx$86–87% (cross subject evaluation) with some reports of 90% for some Deep Learning models and complex architectures.

On the other hand, our small feature vector, featuring only two EEG channels, with our naive selection of statistical measures, without any injection of domain-specific knowledge, managed to show a satisfying average accuracy for a first try. Following the suggestion of a 2023 relevant paper, discussing a more in-depth traditional ML approach, [We shall] *"not sleep on traditional Machine Learning"* [8].

7 Conclusion

In this assignment we examined dimensionality reduction and classification methods on Electroencephalography data obtained from sleeping individuals. We applied different dimensionality reductions and tested each compared to the full-dimensional dataset on two fundamental classifiers, Gaussian Naive Bayes and Gaussian Mixture Models.

For the evaluation of the classifiers, data from the same individual was used, both for the same night and another's night dataset. Cross subject evaluation showed weaker performance. Numerical results as well as the full Python code can be found on GitHub public repository, currently holding a first version of this code (`v1.ipynb`) [2].

While the overall performance did not exceed 88$\pm$2% accuracy, and while current benchmarks on subject-specific evaluations can exceed 90% accuracy, overall GNB on a 4-dimensional embedding could stand between the low state-of-the-art performers. Hence, besides feature exploration as discussed in Discussion section, future work could include thorough examination of the multivariate Bayes classifier, as well as an ensemble of new approaches, i.e. selective classifiers best suited for different sleep stages or more sophisticated pipelines.

This assignment was completed as part of the postgraduate course *Machine Learning*, in the School of Electrical and Computer Engineering at the Technical University of Crete.

References

[1] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, rRID:SCR_007345. [Online]. Available: https://physionet.org/about/citation/

[2] C. Amenty, "sleep-classification," https://github.com/chivintar/sleep-classification, 2025, gitHub repository.

[3] M. Walker, *Why We Sleep*. Harlow, England: Penguin Books, 2018.

[4] M. X. Cohen, *Analyzing Neural Time Series Data: Theory and Practice*. The MIT Press, 2014. [Online]. Available: https://www.amazon.com/Analyzing-Neural-Time-Data-Practice/dp/0262019876

[5] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, and B. V. Vaughn, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. Darien, Illinois: American Academy of Sleep Medicine, 2012, vol. 176.

[6] J. R. Lemke, G. Kluger, and G. Krämer, "Hans berger and 100 years of the electroencephalogram," *Clinical Epileptology*, vol. 37, no. Suppl 3, pp. 112–119, 2024. [Online]. Available: https://doi.org/10.1007/s10309-024-00704-6

[7] A. B. Downey, *Think Bayes: Bayesian Statistics in Python*, 2nd ed. O'Reilly Media, 2021. [Online]. Available: https://learning.oreilly.com/library/view/think-bayes-2nd/9781492089452/

[8] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, N. Vandenbussche, M. Rademaker, G. Vandewiele, and S. Van Hoecke, "Do not sleep on traditional machine learning," *Biomedical Signal Processing and Control*, vol. 81, p. 104429, Mar. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.bspc.2022.104429