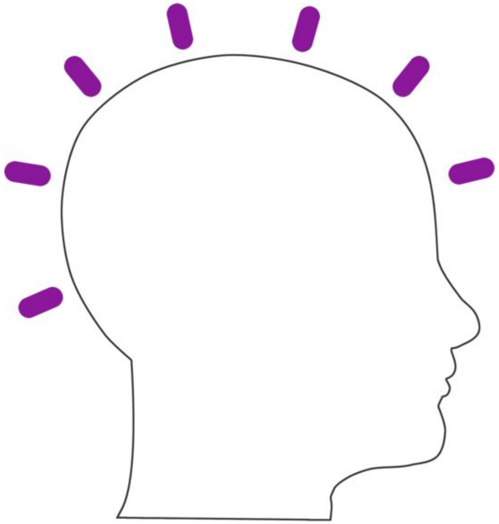
Biosignal Analytics

Group 4 - Sleep stage classification using electroencephalography



Andrei Ivanov
Faiza Anan Noor

Presentation contents

- **The problem** - Sleep stage classification
- **EEG Basics**
  - Time-domain features
  - Frequency-domain features
- **Project workflow**
  - Provided data: import and query
  - Functions for feature extraction
  - Dataset building
  - ML algorithms
    - Random Forest Classification
    - Support Vector Classification
- **Results, findings and conclusions**
  - What could have been done differently?
- **Resources and references**

The problem

Sleep stage detection

This project's goal is to classify sleep stages into six classes using machine learning algorithms: **awake, REM, N1, N2, N3, and N4** by using features extracted from EEG signals.

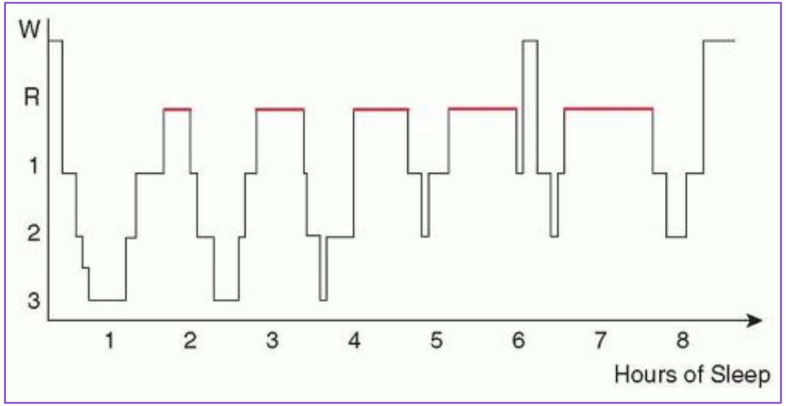
Understanding Sleep Categories

- **REM Sleep vs. NREM Sleep:**
  - REM: Rapid eye movements, irregular breathing, increased heart rate.
  - Vital for healthy emotion regulation, majority of dreams take place during REM sleep
  - NREM: Comprises stages N1-N4, categorized into light (N1-N2) and deep (N3-N4) sleep.

The problem

Breakdown of NREM Sleep Stages

- **Light Sleep (N1-N2):**
  - Preliminary stages of sleep.
- **Deep Sleep (N3-N4):**
  - Crucial for various bodily processes.
  - Production of delta waves identifiable in N3



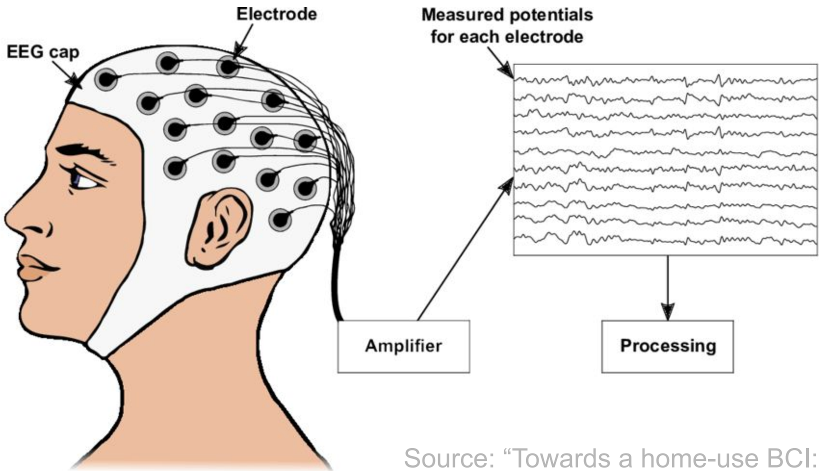
Hypnogram

Source: https://neupsykey.com/sleep-disorders-13/

AI in predicting sleep stages:

- **Early detection:** Potential to identify sleep disorders
- **Treatment integration:** Help doctors' recommendations based on individual patterns observed
- **Research support:** Broadening the understanding of the role of sleep in general health (automating hypnogram annotation)

EEG Signal Basics



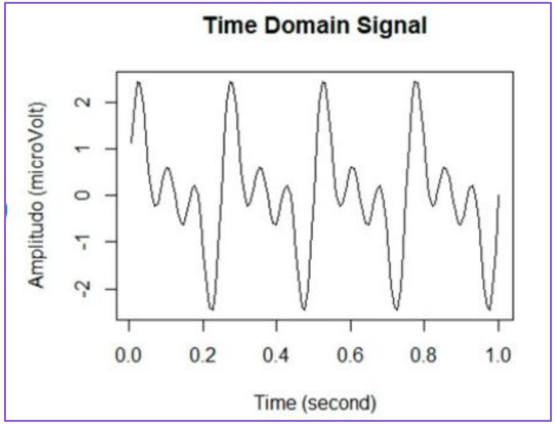
Source: "Towards a home-use BCI: fast asynchronous control and robust non-control state detection" (ResearchGate)

- **Physiological significance:** reflects neural processes and cognitive functions through recording brain activity patterns and associating certain states
- **What it measures:** records brain's electrical potentials.
- **How it is measured:** electroencephalography electrodes placed on scalp; signals amplified and recorded.
- **Signal acquisition:** standard 10-20 EEG setup consists of 19 channels
  - Our channels: Fpz-Cz (frontal to central), Pz-Oz (parietal to occipital)
- **General characteristics of signal:** non-stationary (varies over time), dynamic patterns reflect brain function

EEG Basics

Time domain features

Source: "Classification of Brainwaves for Sleep Stages by High-Dimensional FFT Features from EEG Signals" (ResearchGate)



- Root-Mean-Square
- Zero-Crossing-Rate
- Mean, Standard Deviation, Variance, Skewness, Kurtosis

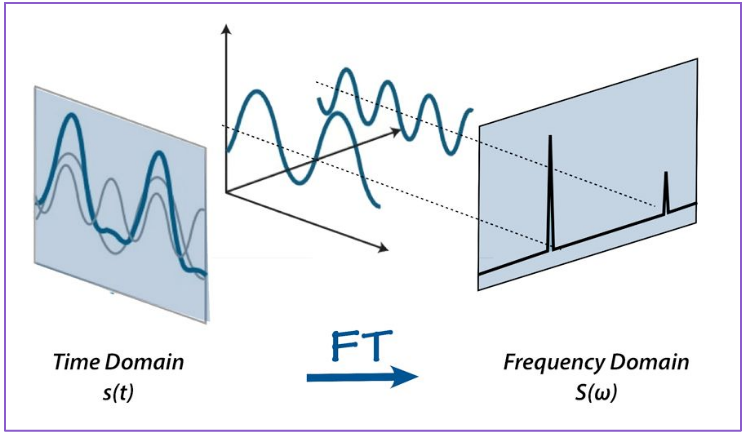
Reason for usage:

- Characterize the *overall* intensity and variability of neural activity and temporal correlations
- Higher RMS values may indicate higher signal intensity or increased neuronal activity), "Zero-crossing patterns reveal subtle epileptiform discharges in the scalp EEG" (https://www.nature.com/articles/s41598-021-83337-3)
- Useful for AI models: even though patterns are not always obvious to humans, algorithms could find useful relationships between time-domain features and certain classes

EEG Basics

Frequency domain features

- Low delta (0.5-1 Hz), delta (1-4 Hz) , theta (4-8 Hz), alpha (8-12 Hz), sigma (12-15 Hz), and beta (15-30 Hz) bands relative power
- Spectral Entropy, Katz Fractal Dimension
- Spectral Edge Frequency
- Inter-channel Itakura Distance



Source: Fundamental Terms of Signal Processing (Medium.com)

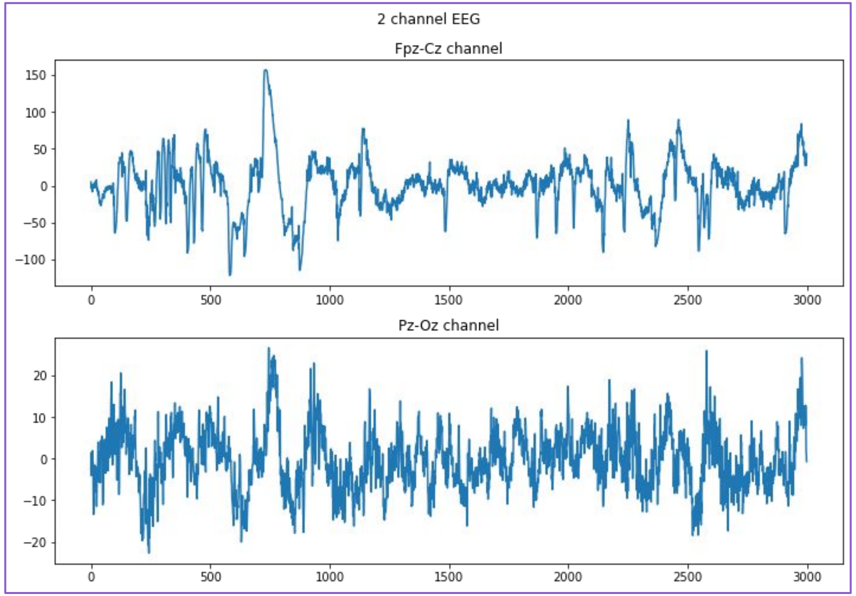
Reason for usage:

- Different brain processes and cognitive states are associated with distinct frequency patterns in EEG signals (characteristic frequency patterns associated with different sleep stages)
- Entropy measures capture the complexity and unpredictability of EEG signals
- The Itakura-Saito distance is a non-symmetric measure of the difference between two probability distributions, can be adapted for frequency spectra

Project workflow

Importing and querying provided data

Our data is extracted from the Physionets Sleep EDF database.



Fpz-Cz channel
(index 1 => Pz-Oz channel)

train_x[0][:,0]

1st signals matrix

- The code looks through the data folders and reads each .csv file provided, each having the EEG recording for the Fpz-Cz and Pz-Oz channels
- Classes in the *y* arrays as well as data arrays are read in the order found in folders (ex: train_x[5] has its corresponding class train_y[5])

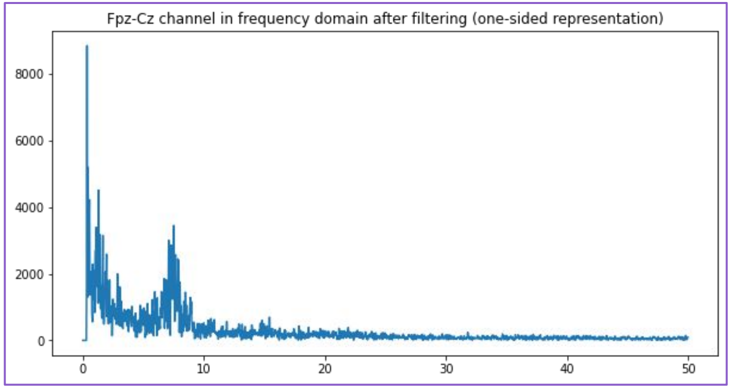
Project workflow

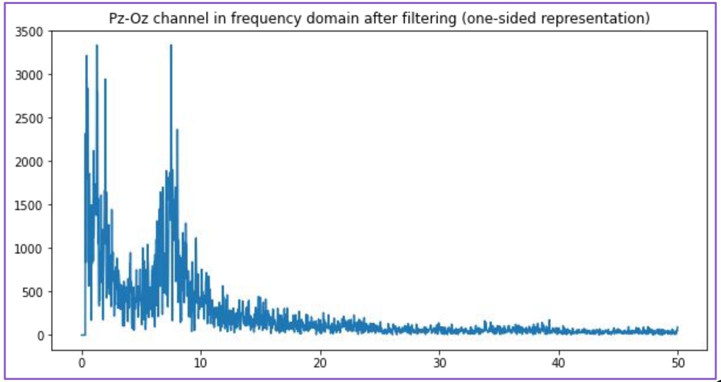
Functions for feature extraction

```python
def time_dom_features(signal):
    rms = np.sqrt(np.mean(np.square(signal)))
    zcr = np.mean(np.abs(np.diff(np.sign(signal))))
    # summary statistics
    mean = np.mean(signal)
    std = np.std(signal)
    var = np.var(signal)
    skewness = sp.stats.skew(signal)
    kurtosis2 = sp.stats.kurtosis(signal)
    return rms, zcr, mean, std, var, skewness, kurtosis2
```

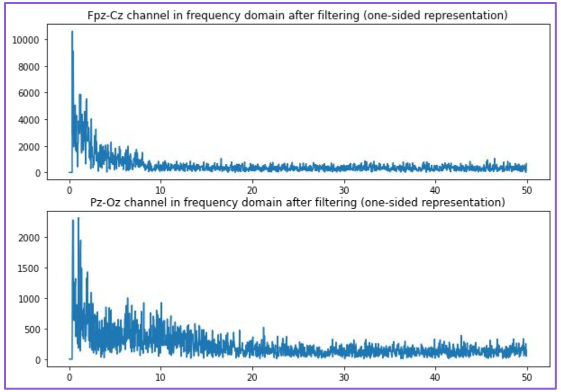
Frequency Domain Features:

FFT_prepare	Extract_bands
Relative_power	Itakura_dist_channels
Extract_entropies	Spectral_edge_frequency

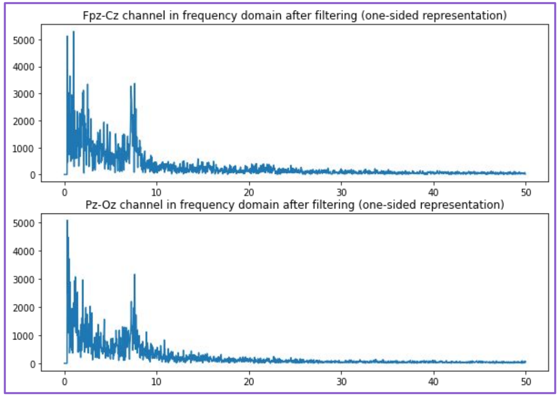




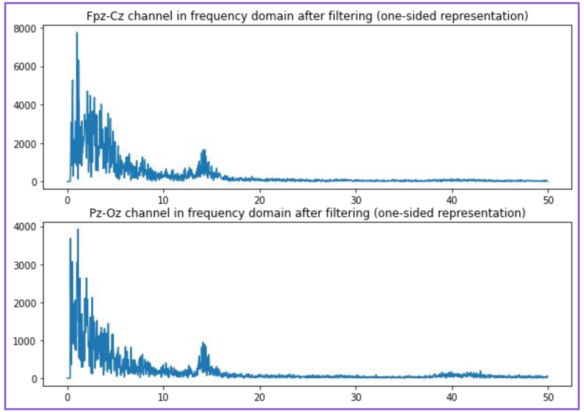
train_x[1356] signal matrix used for plots above



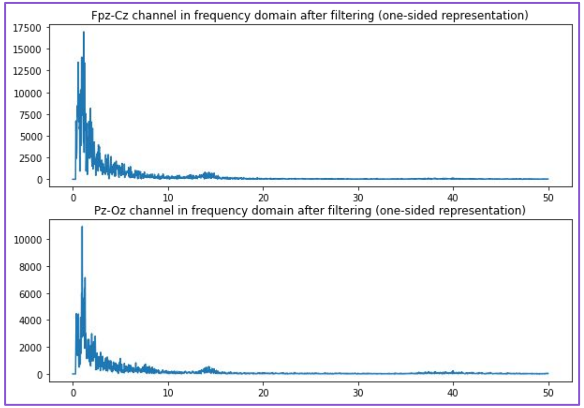
Awake



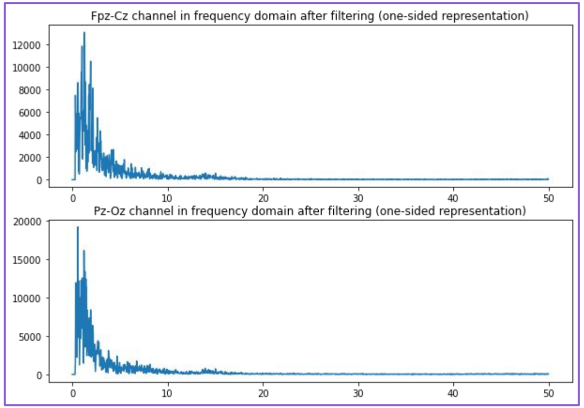
N1



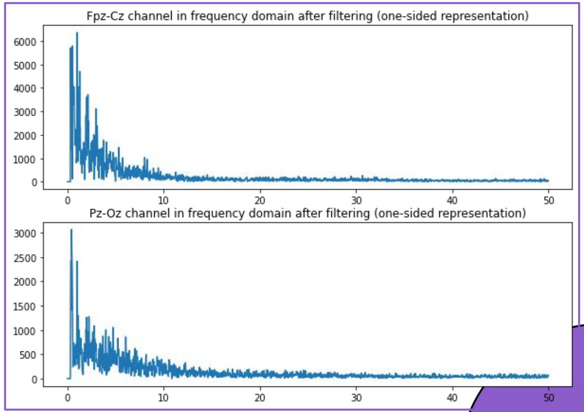
N2



N3



N4



REM

```
[CV 1/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.779 total time=   0.6s
[CV 2/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.740 total time=   0.7s
[CV 3/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.712 total time=   1.2s
[CV 4/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.749 total time=   1.1s
[CV 5/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.767 total time=   1.0s
[CV 1/5] END C=1000, gamma=0.001, kernel=linear;, score=0.748 total time= 1.9min
[CV 2/5] END C=1000, gamma=0.001, kernel=linear;, score=0.738 total time= 1.9min
[CV 3/5] END C=1000, gamma=0.001, kernel=linear;, score=0.737 total time= 2.2min
[CV 4/5] END C=1000, gamma=0.001, kernel=linear;, score=0.771 total time= 1.7min
[CV 5/5] END C=1000, gamma=0.001, kernel=linear;, score=0.789 total time= 2.3min
[CV 1/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.743 total time=   0.5s
[CV 2/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.746 total time=   0.5s
[CV 3/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.737 total time=   0.6s
[CV 4/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.761 total time=   0.5s
[CV 5/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.790 total time=   0.6s
[CV 1/5] END C=1000, gamma=0.0001, kernel=linear;, score=0.748 total time= 1.9min
[CV 2/5] END C=1000, gamma=0.0001, kernel=linear;, score=0.738 total time= 1.9min
[CV 3/5] END C=1000, gamma=0.0001, kernel=linear;, score=0.737 total time= 2.2min
[CV 4/5] END C=1000, gamma=0.0001, kernel=linear;, score=0.771 total time= 1.7min
[CV 5/5] END C=1000, gamma=0.0001, kernel=linear;, score=0.789 total time= 2.3min
{'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
```

Excerpt from the hyperparameter grid search

```
Accuracy: 0.8074501573976915 for p= 0.1
Accuracy: 0.8053515215110179 for p= 0.15
Accuracy: 0.8111227701993704 for p= 0.2
Accuracy: 0.8111227701993704 for p= 0.25
Accuracy: 0.8111227701993704 for p= 0.3
Accuracy: 0.8064008394543547 for p= 0.35
Accuracy: 0.8048268625393494 for p= 0.4
Accuracy: 0.8058761804826863 for p= 0.45
Accuracy: 0.8074501573976915 for p= 0.5
Accuracy: 0.810598111227702 for p= 0.55
Accuracy: 0.8116474291710388 for p= 0.6
Accuracy: 0.8069254984260231 for p= 0.65
Accuracy: 0.8037775445960126 for p= 0.7
Accuracy: 0.8001049317943337 for p= 0.75
Accuracy: 0.8022035676810073 for p= 0.8
Accuracy: 0.8090241343126967 for p= 0.85
Accuracy: 0.8090241343126967 for p= 0.9
Accuracy: 0.8121720881427072 for p= 0.95
```

Spectral Edge Frequency Percentage choice

Project workflow

Dataset building

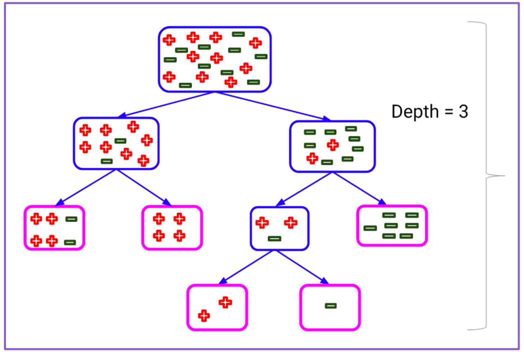
- Features can thus be extracted from each signal matrix, so a set of time-domain features and frequency-domain features result from both the Fpz-Cz channel and Pz-Oz channel, along with the only one inter-channel feature
- All this data should be stored in a comprehensible way to be “fed” to the ML algorithms

	Fpz-Cz Channel				Pz-Oz Channel				inter-channel
	RMS	ZCR	...	SEF	RMS	ZCR	...	SEF	Itakura dist
sample 1									
sample 2									
sample 3									
...									

Training dataset length: 5963
Testing dataset length: 1906

Final number of features: 33

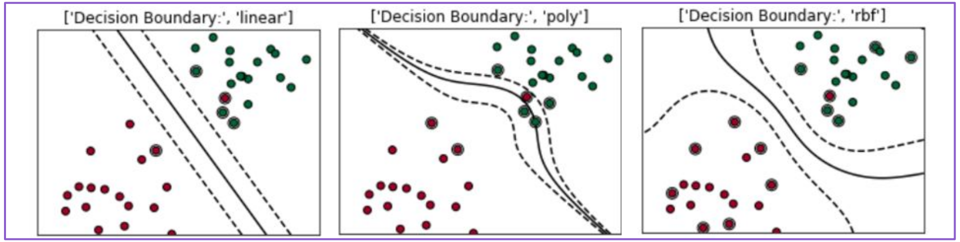
Source: https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/



Machine Learning Algorithms

- **Random Forests (RF):**
  - Combines the output of multiple decision trees to reach a single result for high-dimensional data.
  - Requires feature engineering for optimal performance.

- **Support Vector Machines (SVMs):**
  - Effective for binary classification in feature-rich spaces utilizing a kernel trick to find optimal hyperplanes.
  - Also requires feature extraction.



Source: "Support Vector Machine — Simply Explained" (Medium.com)

Potential use of Neural Networks

process raw time series data directly, extract spatial features directly from EEG signals without prior feature extraction.

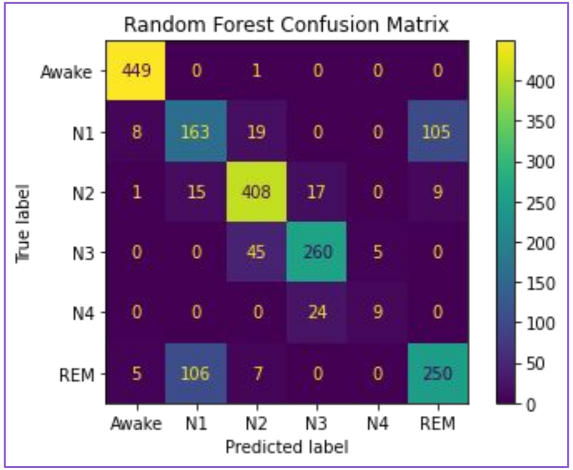
Implementation of Random Forest classifier

- On different runs, accuracy might be slightly different (0.80 - 0.82) for the Random Forest Classifier because of the "randomness" that creating each decision tree involves
- We haven't performed hyperparameter tuning, but we could have altered the depth of each tree in the model, number of decision trees (estimators) or maximum number of features that the random forest model is allowed to try at each split.

Random Forest Classifier Performance Metrics

Overall accuracy: 0.807

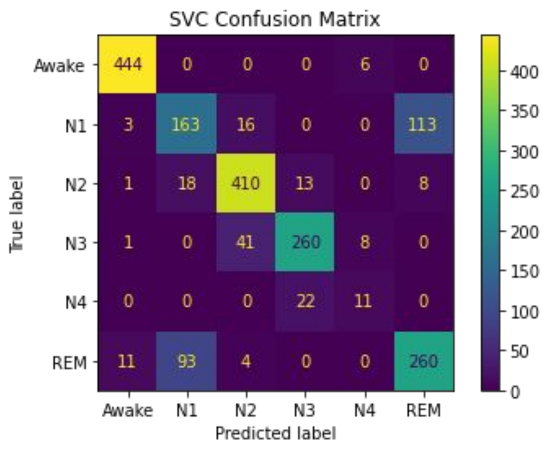
	precision	recall	f1-score	support
Awake	0.97	1.00	0.98	450
N1	0.57	0.55	0.56	295
N2	0.85	0.91	0.88	450
N3	0.86	0.84	0.85	310
N4	0.64	0.27	0.38	33
REM	0.69	0.68	0.68	368
accuracy			0.81	1906
macro avg	0.76	0.71	0.72	1906
weighted avg	0.80	0.81	0.80	1906



Implementation of Support Vector classifier

- **Implementation Details:**
  - RBF kernel with hyperparameters:
    - C = 100
    - Gamma = 0.001
- We utilized GridSearchCV for automatic hyperparameter optimization, the algorithm explored a range of hyperparameter values to find the best combination.
- SVC accuracy value will remain constant at 0.812.

	precision	recall	f1-score	support
Awake	0.97	0.99	0.98	450
N1	0.59	0.55	0.57	295
N2	0.87	0.91	0.89	450
N3	0.88	0.84	0.86	310
N4	0.44	0.33	0.38	33
REM	0.68	0.71	0.69	368
accuracy			0.81	1906
macro avg	0.74	0.72	0.73	1906
weighted avg	0.81	0.81	0.81	1906



Results, findings and conclusions

Evaluation of used algorithms

- It can be concluded that the models are able to distinguish between "Awake" and "People in any sleep state" but have troubles with distinguishing between different sleep stages.
  - Good predictions for the “Awake”, “N2” and “N3” classes
  - Not so good predictions for “N1”, “N4” and “REM” classes
- One conclusion can be that both classifiers perform similarly on this classification problem.
- Hyperparameter tuning could (should) be also done for the Random Forest implementation.
- Presented accuracy is definitely biased due to dataset imbalance and model performance on specific classes, a more realistic evaluation of the models would require cross-validation techniques

Results, findings and conclusions

Observations for results

- The reason why the models are not good with distinguishing between the sleep stages can be because the support data that is used to train and test the models has disproportionate representation for different stages (N4 class a really small support).
- Importances were computed and it was noticed that each model focuses more on different features to "decide" on how they predict the sleep stage, but some important repetitions can be observed: feature index 3 (**Fpz-Cz channel standard deviation**), feature index 0 (**Fpz-Cz channel Root Mean Square**), feature index 28 (**Pz-Oz channel Beta band power**), feature indices 11 and 12 (**Fpz-Cz channel Sigma and Beta band power**), feature 17 (**Pz-Oz channel Zero-Crossing-Rate**), feature index 26 (**Pz-Oz channel Alpha band power**), feature index 29 (**Pz-Oz channel spectral entropy)**

Results, findings and conclusions

Potential for improvement

- Extracting More Relevant and Useful Features:
  - Use supervised feature selection to filter out irrelevant features.
  - Feed neural network with signal chunks for nuanced patterns.
  - Advanced feature extraction captures detailed data characteristics.
- Address Data Imbalance:
  - Some classes (like N4) had limited training/testing samples.
  - Implement techniques like gathering more data, oversampling or class weights adjusting.
- Initial Awake State Classification:
  - Separate awake state classification before sleep stage prediction.
  - Minimize bias and improve model accuracy.
- Focus more on different frequency band operations
  - What if we reconstruct time-domain signals for each frequency band and compute time-domain features for each separate signal?