Final Report: MarineSomniac

Authors: Yousuf Rajput, Melody Song, Michael Sorenson, Tarik Basic, Tarek Tarif
Advisors: Dr. Jessica Kendall-Bar, Dr. Altintas

Abstract

This project investigates the sleep behavior of northern elephant seals by leveraging a comprehensive dataset derived from EEG, TDR, and SR recordings across various environments including laboratories, natural habitats, and open sea settings. Key accomplishments include the development of a robust data processing pipeline, the application of advanced machine learning models to classify sleep stages, and the creation of visual tools to aid in data interpretation and reporting. Our findings reveal significant patterns in sleep behavior, contributing to a deeper understanding of marine mammal physiology and their adaptations to different environments.

Introduction

Understanding the sleep patterns of northern elephant seals presents a unique challenge due to their diverse and dynamic habitats. These marine mammals exhibit complex behaviors that require sophisticated data collection and analysis methods. The primary challenge was to develop a methodology that could accurately classify and analyze sleep stages from a variety of data sources including EEG, ECG, and motion sensors.

Problem Formation: To address this challenge, the project formulated a data science problem that involves processing and analyzing large volumes of heterogeneous data to identify and classify sleep stages in seals.

The primary objectives were to:

- Develop a data collection and processing pipeline.
- Apply machine learning models to classify sleep stages.
- Integrate various data types to enhance the accuracy of sleep stage classification.

Questions:

1. How can we accurately classify sleep stages in seals using EEG and motion data?
2. What are the key features that differentiate various sleep stages?
3. How do environmental factors influence sleep patterns in northern elephant seals?

4. Can we improve classification accuracy by integrating multiple data sources?
5. How do sleep patterns vary across different environmental settings (lab, wild, sea)?

Related Work:

How does this study differ from, or expand upon, any previous work carried out in this field, if any?

Previous studies have investigated sleep in marine mammals using similar electrophysiological data, but these efforts were often limited by the scope of data and the methodologies employed. This project builds on these foundational studies by integrating multiple data types and applying advanced machine learning techniques, which have shown promise in handling complex, high-dimensional data.

For instance, studies on other marine species have demonstrated the utility of EEG and motion data in identifying sleep stages, but this project aims to push the boundaries by incorporating additional environmental data and employing more sophisticated analysis techniques

Data Acquisition

Data Sources: The project utilized a rich dataset collected from various instruments including EEG, ECG, and motion sensors. The data collection process involved deploying custom non-invasive head caps and waterproof housings on the seals to record their physiological and environmental parameters. Data sources included:

- EEG (Electroencephalogram): Recorded brain waves to monitor sleep stages.
- ECG (Electrocardiogram): Captured heart activity.
- EOG (Electrooculogram): Tracked eye movements.
- EMG (Electromyogram): Monitored muscle activity.
- Motion Sensors: Recorded three-dimensional movement and environmental parameters such as depth, temperature, and illumination.

Volume, Variety, Velocity:

- Volume: The dataset included extensive recordings from multiple seals over different periods, resulting in large volumes of time series data.

- Variety: The data encompassed various types of physiological and environmental signals, providing a comprehensive view of the seals' behavior.
- Velocity: Data was continuously recorded, with electrophysiological signals sampled at 500Hz and environmental/motion sensors at ~36Hz, later down-sampled to 8-second intervals for analysis.

Technologies: Data was accessed and processed using a combination of cloud storage (Qumulo), Python for data manipulation and analysis, LabChart for physiological data visualization, MATLAB for advanced signal processing, and TimescaleDB for efficient time series data storage. Utilizing diverse technologies enabled the integration of various data sources and facilitated comprehensive analysis

Data Collection: Data was collected from both controlled environments (laboratories and pools) and natural settings (shore and out at sea). This involved attaching sensors to the seals using custom-designed head caps and housings that ensured data integrity while allowing the seals to move freely. There were instances where data collection was interrupted that led to the exclusion of certain data segments. Nevertheless, the devices employed were robust enough to capture accurate data during periods when the seals were engaged in typical behaviors.

Data Sizes: The data sizes varied depending on the recording duration and the number of sensors used. Typically, each deployment generated gigabytes of data, which required efficient storage and processing solutions. For example, much of the modeling and testing that was conducted was not done on the entire data set but on specific sets.

Data Pipelines: The data pipeline was set up to handle large volumes of data efficiently. This involved:

- Data Ingestion: Data was ingested from various sources into a central repository.
- Data Cleaning: Noise and artifacts were removed to ensure data quality.
- Data Transformation: Raw data was transformed into formats suitable for analysis.
- Data Storage: Processed data was stored in TimescaleDB for efficient querying and retrieval.

Setup for Data Environment

To manage the vast amount of data collected, the project utilized a combination of cloud and local resources. The primary storage solution was the OpenStack Object Store, which provided robust cloud storage for both raw and processed data files.

This setup was crucial for handling the high volume of data, which included 2GB of raw files and 200MB of processed EDF files. Additionally, training labels, which were manually created, were also stored in the OpenStack Object Store, ensuring all data components were centrally located and easily accessible.

A key decision in the data pipeline setup was the use of cloud storage versus local storage. Cloud storage, through the OpenStack Object Store, was chosen for its scalability and accessibility. This choice allowed team members to access data remotely and facilitated collaboration. For processed data, TimescaleDB was selected as the database solution. TimescaleDB's ability to handle time-series data efficiently made it an ideal choice for storing the processed features and training labels.

Data Preparation

Data preparation was a critical phase, addressing several quality issues inherent in the datasets. Outliers in motion and pressure data were identified and handled, and misaligned time indices in labels were corrected. The transformation of raw data into analyzable formats involved spectral signal processing to extract features at a 1 Hz sample rate, peak detection for heart rate extraction from ECG data, and Fourier transformation for calculating EEG delta power.

The significance of these pre-processing methods lay in their ability to ensure clean, aligned datasets that were crucial for effective model training. Feature selection and management were iterative processes, involving refinement based on model performance. Key features included heart rate characteristics, delta spectral power, and motion data, all of which were managed meticulously to optimize the model's performance.

Data preparation was a vital phase of the project, involving a series of meticulous steps to address quality issues, transform raw data into usable formats, and ensure the datasets were properly aligned for effective analysis and model training.

Quality Issues:

Several quality issues were identified in the raw datasets, particularly in the motion and pressure data. These datasets exhibited outliers, which were likely the result of sensor errors or inconsistencies in data collection. To address this, outlier detection and removal techniques were applied. This process involved analyzing the distribution of the data and identifying values that deviated significantly from the norm. These outliers were then excluded to prevent them from skewing the results of subsequent analyses.

Another significant quality issue was the misalignment of time indices in the labels. This misalignment occurred during the manual labeling process and could lead to inaccurate model training if not corrected. The time indices were carefully reviewed and realigned with the corresponding data points to ensure accuracy.

Transformation and Integration:

The raw data required extensive transformation and integration to be suitable for analysis. This process involved several steps of spectral signal processing to extract meaningful features from the high-frequency data. For example, the ECG data, recorded at 500Hz, needed to be processed to derive heart rate features. This involved peak detection to identify R peaks in the ECG signal, from which heart rate could be calculated as the time interval between successive peaks. Additional features such as the average heart rate and the standard deviation of heart rate were also derived from these calculations.

For the EEG data, a Fourier transformation was performed to isolate the power of signals within the delta frequency range (0.5 - 4 Hz). This delta power is a crucial indicator of sleep stages, particularly slow-wave sleep. The transformation process involved applying a Fourier transform to the EEG signals and then calculating the power within the specified frequency range over rolling windows. This approach allowed for the extraction of delta power features at a 1 Hz sample rate, aligning with the target variable's frequency.

Significance of Pre-processing Methods:

The pre-processing methods used in this project were critical for ensuring that the data were clean, aligned, and suitable for model training. By addressing quality issues such as outliers and misaligned time indices, the team ensured that the datasets reflected accurate and reliable information. The transformation of raw data into analyzable features enabled the extraction of meaningful insights that were essential for the development of the machine learning models.

Feature Selection and Management:

Feature selection and management were iterative processes, guided by the performance of the models. Initially, a broad set of features was extracted from the raw data, including heart rate characteristics (e.g., mean heart rate, standard deviation of heart rate, very low-frequency power), delta spectral power from EEG signals, and various motion data features (e.g., overall dynamic body acceleration (ODBA), pressure, and gyroscopic data).

These features were evaluated based on their contribution to model performance. Features that significantly improved model accuracy were retained, while those that did not contribute meaningfully were discarded. This iterative process involved repeated cycles of model training, performance evaluation, and feature refinement. The goal was to identify and retain the most relevant features that would enhance the model's ability to accurately classify sleep states.

Data Cleaning, Wrangling, and Feature Engineering:

The data cleaning and wrangling process involved several key steps to prepare the data for analysis:

1. **ECG Data Processing:** Peak detection algorithms were applied to the ECG signals to identify R peaks, enabling the calculation of heart rate features. These included average heart rate, standard deviation of heart rate, and very low-frequency power.
2. **EEG Data Processing:** Fourier transformation was used to calculate delta spectral power, focusing on the 0.5 - 4 Hz frequency range. This process involved rolling window calculations to maintain a 1 Hz sample rate.
3. **Motion Data Processing:** Motion data, including ODBA and pressure, were cleaned by removing extreme outliers. These features required minimal processing compared to the electrophysiological data.
4. **Integration of Processed Features:** All processed features were integrated into a unified dataset, aligning them by time indices to ensure consistency across different data types.

Analysis Methods

Identification of Preliminary Analysis Methods

The identification of methods for preliminary analysis was driven by the need to understand the complex electrophysiological and motion data collected from Northern Elephant Seals. The project began with Exploratory Data Analysis (EDA) to uncover patterns and insights within the dataset. Tools such as MNE-Python were employed to handle and visualize the EDF files containing electrophysiological data. This initial exploration revealed significant features, such as the prominence of delta waves (0.5 - 4 Hz) in EEG signals during slow-wave sleep, which contrasted with their absence in REM sleep and minimal presence in other stages.

Using EDA was significant because it allowed the team to identify key characteristics and anomalies in the data, guiding subsequent analytical steps. For example, the identification of delta waves as a critical feature influenced the decision to include spectral power calculations in the feature extraction process. This method also highlighted the need for careful preprocessing, such as time alignment of labels and removal of outliers, ensuring the integrity of the data used for model training.

Influence on Project Design and Data Science Questions

The insights gained from EDA influenced the design of the project's next steps by refining the data science questions. Initially broad questions like "What is the best methodology for sleep scoring in marine animals?" were further defined to focus on specific aspects such as the differentiation of sleep stages using spectral features and heart rate variability. This focus helped streamline the analysis process and directed efforts towards the most promising features and models.

Application of Analysis Techniques

Applying analysis techniques to the data involved several key steps, driven by the need to derive meaningful insights and prepare the data for machine learning models. The process included:

1. **Feature Extraction:**
    - **ECG Data:** Peak detection algorithms were used to identify R peaks in the ECG signals, allowing the calculation of heart rate features such as average heart rate and very low-frequency power. These features were crucial for capturing heart rate variability, which is indicative of different sleep stages.
    - **EEG Data:** Fourier transformations were applied to the EEG signals to calculate delta spectral power. This involved decomposing the EEG signals into their constituent frequencies and summing the power within the delta range. This feature was essential for identifying slow-wave sleep.
    - **Motion Data:** Features like overall dynamic body acceleration (ODBA) and pressure were calculated, with outliers removed to ensure clean data. These features helped differentiate between active and restful states.
2. **Preliminary Modeling:**

- Initial modeling efforts involved testing various machine learning classifiers, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest. GridSearchCV was used to tune hyperparameters and identify the best model configurations.
- A significant challenge identified during this phase was the misalignment of time indices, which initially led to poor model performance. Correcting this misalignment significantly improved model accuracy, demonstrating the importance of precise data preprocessing.

3. **Feature Importance:**

   The evaluation of feature importance was a critical component of this project, providing valuable insights into which features significantly contributed to the model's performance. Understanding feature importance helped in refining the feature set, improving model accuracy, and enhancing interpretability.

Methodology:

Feature importance was assessed using the inherent capabilities of the Random Forest and LightGBM (LGBM) classifiers. These models offer methods to calculate the importance of each feature based on how frequently and significantly the feature is used in the decision-making process of the trees.

Random Forest Classifier:

The Random Forest classifier was utilized because it provides high accuracy when it comes to complex datasets. It effectively handles missing values and outliers, ensuring reliable performance even with occasional data interruptions, which was not uncommon in the gathering of physiological data of the seals. The Random Forest classifier also offers valuable insights into feature importance, helping to identify the most impactful features for the model.

Each tree in the Random Forest uses a subset of features to make decisions, and the frequency with which a feature is used across all trees indicates its importance. The key steps involved in evaluating feature importance with the Random Forest classifier were:

1. **Training the Model:** The Random Forest model was trained on the processed dataset, incorporating features such as delta spectral power from EEG signals, heart rate variability, and motion data.
2. **Calculating Importance:** After training, the feature importance scores were extracted. These scores represent the average contribution of each feature to the reduction in impurity across all trees.
3. **Interpreting Results:** Features with higher importance scores were identified as crucial for the model's predictions. For instance, delta spectral power and very low-frequency power from heart rate data emerged as significant features, highlighting their critical role in differentiating between various sleep stages.

LightGBM (LGBM) Classifier:

The LGBM classifier also provided feature importance metrics, which were essential for understanding the influence of different features. LGBM's gradient boosting framework enhances interpretability by returning importance scores that reflect the feature's impact on the model's decisions. The process included:

1. **Training the Model:** The LGBM classifier was trained using the same dataset, ensuring consistency in feature evaluation.
2. **Evaluating Feature Importance:** The importance of each feature was calculated based on its contribution to the decision trees within the boosting framework. The LGBM classifier's ability to handle large datasets efficiently made it particularly suitable for this project.
3. **Refining Features:** The results indicated which features had the highest importance values. These insights guided the refinement of the feature set, focusing on the most impactful features to enhance model performance. For example, it was found that the epoch size for calculating Welch power over EEG data had a significant impact, contrary to initial expectations based on preliminary analyses.

Key Insights from Feature Importance:

The analysis of feature importance revealed several critical insights:

- **Delta Spectral Power (EEG):** This feature was consistently identified as a crucial indicator of slow-wave sleep, emphasizing its importance in sleep stage classification.
- **Very Low-Frequency Power (Heart Rate):** This feature captured long-term oscillations in heart rate, essential for distinguishing between different sleep stages.
- **Motion Data (ODBA and Pressure):** These features helped differentiate between active and restful states, contributing significantly to the model's accuracy.

Impact on Model Development:

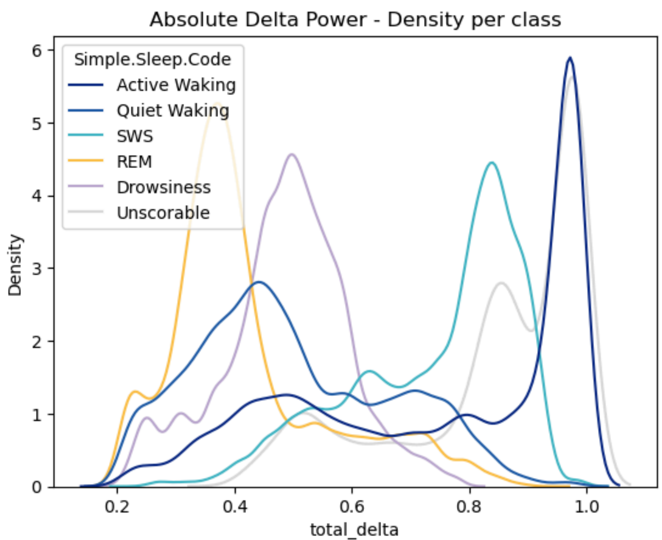
The feature importance analysis played a pivotal role in refining the model development process. By focusing on the most impactful features, the team was able to improve the accuracy and robustness of the models. This iterative refinement process, guided by the insights from feature importance metrics, ensured that the models were both effective and interpretable.

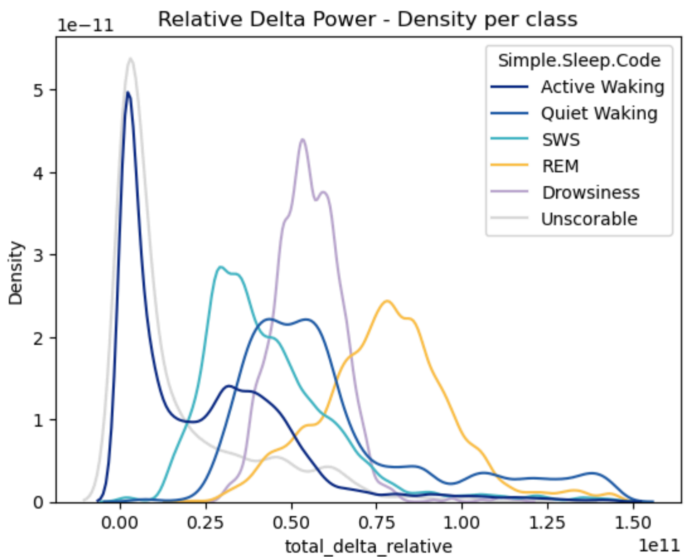
4. Feature Density

Feature density, which assesses the concentration and relevance of various features in a dataset, is crucial for understanding model performance. In our analysis, delta power emerged as a significant EEG feature. During Slow-Wave Sleep (SWS), delta power exhibits high absolute power while maintaining a relatively high proportion compared to the total power in the 0.4-30Hz range.

Conversely, in Active Waking states, absolute delta power is also high, but this is largely due to EEG noise, reflected in the low relative delta power. During Drowsiness, absolute delta power is higher than in REM sleep but lower than in SWS, yet its relative delta power surpasses that of SWS.

Finally, in REM sleep, although absolute delta power is low, its relative proportion is high, indicating that delta power constitutes a significant part of the overall low EEG activity during this state. This nuanced understanding of delta power across different sleep stages highlights its importance in the model and underscores the value of feature density in our analysis.





Analytical Workflow:

The analytical workflow was structured to ensure a systematic approach to data analysis and model development. The key stages included:

1. **Data Ingestion:** Raw data were ingested from the OpenStack Object Store and preprocessed using proprietary software (Matlab, LabChart) and Python.
2. **Feature Extraction:** Spectral and time-domain features were extracted from the ECG, EEG, and motion data. This involved applying signal processing techniques and aligning features with the corresponding labels.

3. **Preliminary Analysis:** EDA and descriptive statistics were conducted to identify key patterns and guide feature selection.
4. **Model Development:** Initial models were trained using scikit-learn classifiers. GridSearchCV was employed for hyperparameter tuning, and performance was evaluated using accuracy and confusion matrices.
5. **Iteration and Refinement:** Based on initial results, features and preprocessing steps were refined iteratively. Misaligned labels were corrected, and additional features were explored to improve model performance.

Processing Environment Setup:

The processing environment was set up to support efficient data analysis and model training. This involved:

- **Python and Jupyter Lab:** These tools provided an interactive environment for coding, visualization, and iterative testing.
- **MNE-Python:** Used for handling and visualizing electrophysiological data.
- **scikit-learn:** Employed for model training and evaluation, offering a wide range of machine learning algorithms and tools for hyperparameter tuning.
- **Cloud Storage:** Data was stored and accessed from the OpenStack Object Store, ensuring scalability and accessibility for all team members.
- **Local Computing Resources:** While initial data processing was conducted locally, the use of cloud-based storage and computation ensured that the workflow could handle large datasets efficiently.

Findings and Audience

The Seal Sleep Capstone Project explored the accessibility, reproducibility, and plausibility of the application of machine learning to sleep scoring and sleep studies. The goal of this project evolved and changed over time; originally, the hope was to create an all-purpose, omnipotent machine learning tool that could decrypt any seal's electrocardiogram (ECG) and electroencephalogram (EEG), and a few other raw data channels, and "correctly" classify a seal's sleep state between Active Waking, Quiet Waking, Drowsiness, Slow Wave Sleep (SWS), and Rapid Eye Movement Sleep (REM). In this case, we defined a "correct" classification as one that agrees with the classification given by a seal sleep expert (Jessie, or Ritika).

While it was clear from the beginning that such a task was plausible, it was also clear that it would be very challenging. In the first quarter, our project explored a simple feature set extracted from EEG, ECG, movement (Gyrz), and depth (Pressure) to create classifiers with scikit-learn's KNeighborsClassifier (k-nearest neighbors), LinearSVC (support vector machine), and RandomForestClassifier. We found that each of these methods performed similarly with the features that were included, but the random forest outperformed the others in terms of accuracy on REM and SWS. Using five days of sleep data from the seal "Wednesday," we obtained an overall accuracy of 65% with 92% accuracy on Active Waking, 69% accuracy on SWS, and 52% accuracy on REM.

During the second quarter, much of the initial weeks were spent doing more feature exploration: expanding the features we already were generating to be using the most optimal parameters like size of our epoch windows, and also researching and adding new features that other sleep studies and EEG studies were using. Additionally, we transitioned from the RandomForestClassifier provided by sklearn to a light gradient boosted classifier, from LightGBM. The reasoning for choosing this model was that it is a more advanced boosting framework than the random forest, but still maintains the underlying decision tree structure that fits the seal sleep classification problem very well. By adding these new features and switching to the LightGBM, our accuracy on Wednesday's four days of sleep data improved from 65% to 80.1%, with 93% accuracy on Active Waking, 84% accuracy on SWS, and 63% accuracy on REM.

While this substantial increase in SWS and REM detection was a major improvement, another outcome from the project was a framework for generating brain and heartrate features that can easily be generated for other seals, or completely new animals, to help aid a machine learning approach for sleep detection. When we applied our model trained on Wednesday to five new seals—HypoactiveHeidi, AshyAshley, BerthaBeauty, SnoozySuzy, and JauntingJuliette— we found that the model trained on just Wednesday alone extends decently but not great to a completely new seal, with the REM and SWS accuracy hovering around 40% for each of these seals. This is better than random and when combined with confidence scores could help speed up the seal classification for a completely new seal, but is not good enough for full automation alone. However, if you isolate each seal and train a model on only its own data, they each performed at around the same accuracy as Wednesday using k-fold cross-validation, which suggests that the sleep itself is predictable and learnable, there is simply more work to be done on scaling the features and the datasets so that their distributions are similar enough across seals to give meaningful results from the trained LightGBM classifier.

On the brighter side, using a leave-one-out approach, we were able to predict sleep with much higher accuracy. By training a model with data from just five of the seals and then testing it on the remaining seal, we are able to explore how adding data from new seals to the model helps it learn the varying distributions on its own. We would hope that a model trained on more seals would do better than a model trained on just Wednesday alone, and this seems to be the case. The table below shows the overall accuracy, Active Waking accuracy, SWS accuracy, and REM accuracy for each seal (note that the model was trained using the data from the other five seals). Note that Wednesday, Ashley, Bertha, and Suzy are captives that are only ever in the lab with a shallow 10m pool, while Heidi and Juliette were translocated north and swam back to San Diego, and thus perform deep ocean dives, during which they are sometimes sleeping, but at much lower rates than the lab seals. The deep dives result in different scales in the signals in these seals, and some of the other seals have messier or noisier signals for other reasons.

Test Seal	Overall %	Active Waking %	SWS %	REM %
Wednesday	66.13	87.71	91.73	66.40
AshyAshley	67.22	94.59	65.45	63.02
BerthaBeauty	68.13	84.10	60.81	39.83
SnoozySuzy	67.82	99.60	22.36	18.06
HypoactiveHeidi	92.83	96.13	46.43	4.63
JauntingJuliette	36.32	31.87	73.63	42.05

Wednesday, AshyAshley, and BerthaBeauty were isolated to the lab and have some of the cleanest signals, so it makes sense that their accuracies are the most balanced. SnoozySuzy had a very noisy signal which is likely why the prediction for her was more difficult. HypoactiveHeidi was awake for >90% of her four-day dataset, so the REM she does have is short-lived and harder to predict, while JauntingJuliette's signal was much noisier than the others.

Findings to Present

The most important factor in determining what to present is to show the findings that could have the largest impact and could be used by future researchers to help expedite work they are doing. Keeping this in mind, our product and repository is focused on reproducibility— in the case of the product we have easy to understand GUIs and an interactive interface to help make feature generation simple and seamless. This project ran into many issues and roadblocks simply because there aren't many (if any) researchers working on heart rate and brainwave-focused studies on marine wildlife, so there is not an out of the box “these are the features you need for this problem.” Because much of the work for the project was done on researching and generating features, we want to emphasize this in our presentation, and make that part of the project accessible for someone else who may want to do something similar.

Of course, we will still want to present the easier-to-understand statistics like accuracy, per-class accuracy, and possibly even breaking it down further into a confusion matrix to be able to see which sleep states are most commonly confused for each other. If two classes are commonly being misclassified for each other (e.g. Active Waking and Quiet Waking), this could indicate that the boundary we defined to separate these classes is too rigid, or perhaps that there are many transitions between the two classes, so deciding when exactly the transition happens changes how the model would report its accuracy. Regardless of the accuracy, we definitely want to present visualizations that show how our models predict sleep throughout a nap versus what the actual; this helps illustrate what some of common errors are (and shows that even though the accuracy may be 80%, our model may actually be performing very well, it just detects transition points slightly differently from what a sleep expert may say, creating somewhat of a reported underperformance.

Tools and Techniques

We used lots and lots of python visualizations— seaborn density plots, matplotlib plots, pandas matrices for accuracy and confusion matrices, and used color to communicate different sleep states to help visualize what differences between states we would expect a machine learning model to be able to pick up on. Finally, one of the more important plots that we generate is one with the seal’s derived features lined up with some raw data like heart rate and EEG, along with both the actual sleep state and predicted sleep state. By creating a plot with all of these lined up together on the same timescale, it helped during the development process with debugging issues, but it is also an effective visualization for communicating model performance and showing times when the model does not perform as well. Additionally, we used our product to help communicate some of the intricacies that come with feature extraction, and to help make the whole process easier.

Visualizations and Products

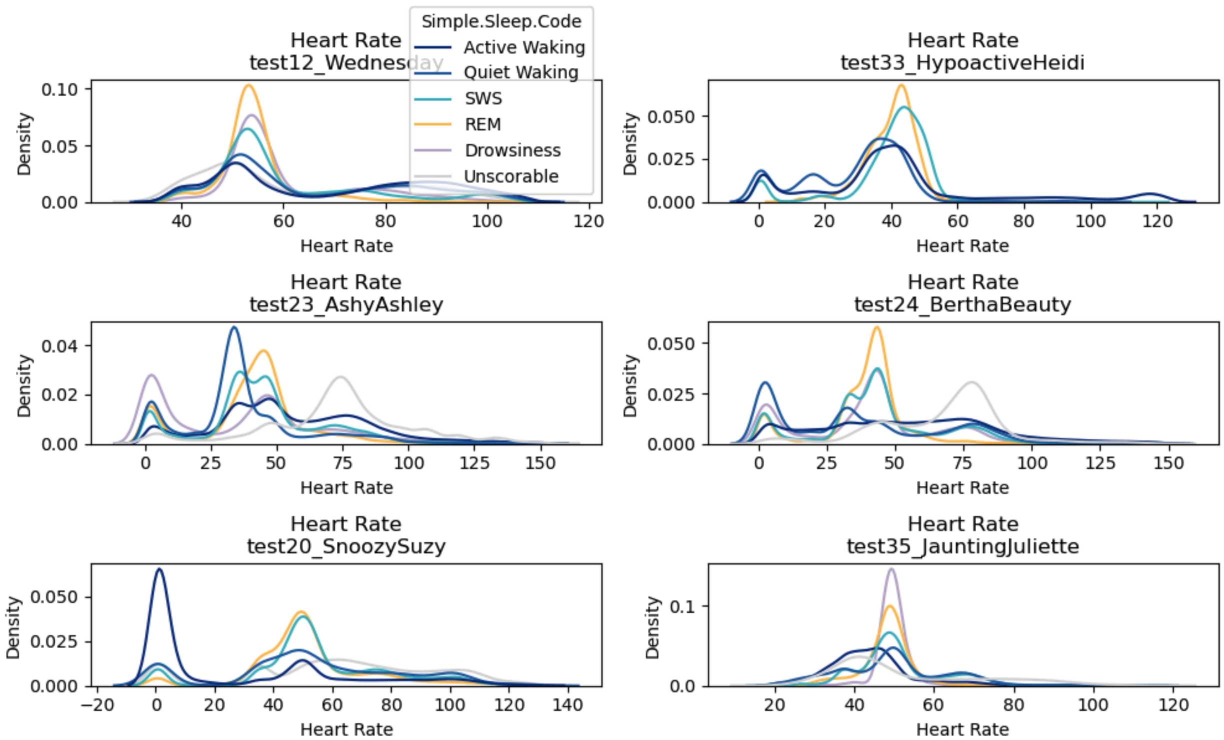


Figure 1: Heart Rate distribution for the different seals; values around 0 illustrate errors in the heart rate generation that stem partially from noisier ECG data. The significantly different distributions in something as simple as heart rate illustrate how each seal is its own individual and poses challenges in using machine learning to predict sleep state.

On Wednesday
Overall Accuracy: 78.44%

Active Waking	Quiet Waking	Drowsiness	SWS	REM
91.84%	50.37%	46.79%	83.51%	64.71%

	Predicted_Active_Waking	Predicted_Quiet_Waking	Predicted_Drowsiness	Predicted_SWS	Predicted_REM
True_Active_Waking	124403	6116	1178	2364	355
True_Quiet_Waking	12087	21119	3133	1261	4606
True_Drowsiness	2325	4616	13650	1657	84
True_SWS	5205	2711	1338	47575	1096
True_REM	1050	4211	68	1607	24408

On Ashley
Overall Accuracy: 62.94%

Active Waking	Quiet Waking	Drowsiness	SWS	REM
77.71%	47.27%	1.55%	53.13%	38.59%

	Predicted_Active_Waking	Predicted_Quiet_Waking	Predicted_Drowsiness	Predicted_SWS	Predicted_REM
True_Active_Waking	165951	23405	21	2568	418
True_Quiet_Waking	13740	24216	213	1104	11631
True_Drowsiness	157	16816	199	281	8800
True_SWS	23498	19623	2456	38383	12310
True_REM	3808	4772	703	1730	22678

Figure 2: On the left, results of the model trained on Wednesday's data test on Wednesday. On the right, model train on Wednesday's data test on AshyAshley. This illustrates that the sleep problem is extensible, i.e. the model trained on Wednesday works as a better-than-random (in fact, much better than random) predictor of sleep in AshyAshley. Out of the additional five seals, Ashley has one of the most similar signals to Wednesday, but with more work on the preprocessing and scaling the other seals should also perform well.

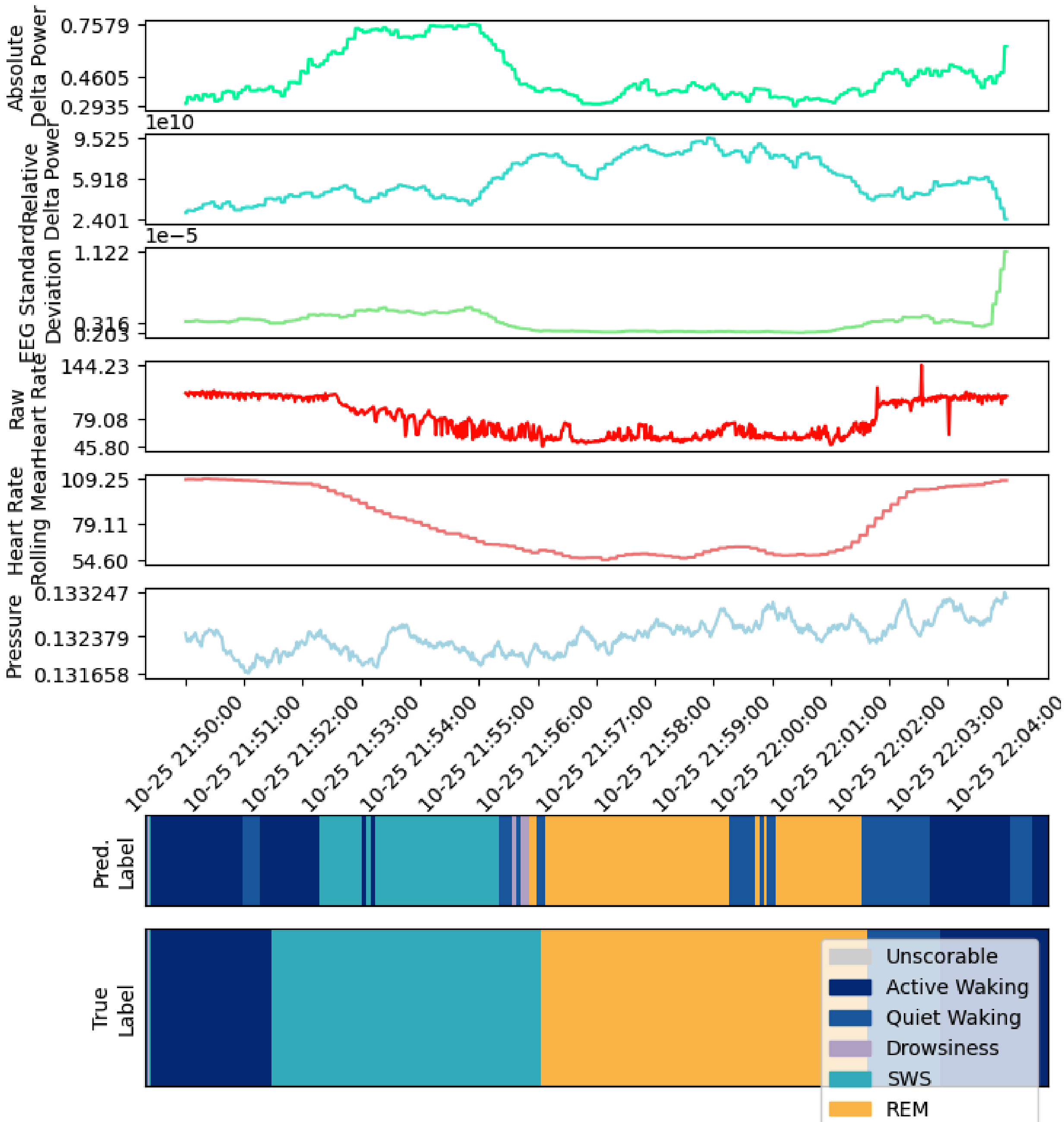


Figure 3: Visualizations of one of Wednesday's naps. Includes a few of the features used for modeling (from top to bottom: Absolute Delta Power, Relative Delta Power, EEG Std.Dev, Heart Rate, Heart Rate Rolling Mean, Pressure), as well as the true and predicted label shown as a matrix plot. This illustrates that the model is mostly corrected, but has brief hiccups where it hallucinates. However, in most cases it hallucinates to a sleep state that is either recent or upcoming, which is a reflection of the transition period between sleep states that we don't have a separate class for.

Solution Architecture, Performance, and Evaluation

Measure of Performance?

Performance was measured using several key metrics to ensure the accuracy and reliability of our models. We primarily used overall accuracy and class-specific accuracies, which were calculated through k-fold cross-validation. This method involves splitting the dataset into training and validation sets multiple times to ensure the model performs well across different subsets of data.

For each fold, we evaluated how well the model's predictions matched the actual sleep states. Additionally, we used confusion matrices to visualize the performance across different sleep stages, helping us understand where the model was making errors. Feature importance metrics provided by the Random Forest and LGBM classifiers were also used to identify which features contributed most to the model's decisions, thereby refining the model further.

How Did You Scale and Evaluate Your Models?

Scaling and evaluating the models involved using dedicated compute nodes and cloud resources to handle the large volume of data efficiently. The data pipeline was designed to be scalable, using the OpenStack Object Store for raw and processed data storage and TimescaleDB for storing processed features and training labels. This setup allowed us to manage and process large datasets effectively. To ensure robustness, we employed cross-validation techniques, such as k-fold cross-validation, which helped us evaluate the model's performance across different subsets of data.

This method reduced the risk of overfitting and ensured that our models were generalizable to new, unseen data. Additionally, we explored different machine learning algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest, and used hyperparameter tuning with GridSearchCV to find the best model configurations.

How Did You Manage Your Budget?

Budget management was a critical aspect of the project, ensuring that resources were used efficiently without compromising the quality of the research. We utilized cost-effective cloud storage solutions through the OpenStack Object Store, which provided scalable and flexible storage options. This approach minimized the need for expensive local storage infrastructure. Computational resources were optimized by using dedicated compute nodes for data processing and model training, balancing the need for performance with cost-effectiveness.

We also leveraged Nautilus, a cost-effective high-performance computing resource, to manage large-scale data processing tasks. Nautilus allowed us to access powerful computing resources without the high costs associated with commercial cloud services. This integration significantly reduced our overall computational expenses while maintaining high processing power and efficiency. The automatic resource-usage tools as part of Nautilus allowed consistent monitoring of our resources and spending as to not go beyond set thresholds as part of our computing infrastructure.

We leveraged open-source software tools such as Python, scikit-learn, and Streamlit, which reduced the need for costly proprietary software licenses. By managing these resources carefully and prioritizing essential expenses, we were able to stay within budget while achieving our project goals.

Conclusion

The Seal Sleep Capstone Project successfully demonstrated the feasibility and effectiveness of using machine learning to classify sleep states in Northern Elephant Seals. By employing advanced models such as LightGBM and comprehensive feature extraction techniques, we achieved significant improvements in accuracy, particularly in detecting critical sleep stages like Slow Wave Sleep (SWS) and Rapid Eye Movement (REM). Our findings highlight the importance of specific features, such as delta spectral power and heart rate variability, in achieving reliable sleep state classification. Visualizations and detailed metrics provided clear insights into model performance and facilitated the identification of areas for further improvement.

This project underscored the potential for scaling these methods to a broader user base, thanks to user-friendly interfaces and scalable computing resources. Our robust framework not only advanced the understanding of marine mammal sleep patterns but also provided a solid foundation for future research and applications in ecological and behavioral studies. By making our methods and findings accessible, we aim to support ongoing and future efforts in marine biology and related fields.

Acknowledgements

We extend our heartfelt thanks to our advisor, Dr. Jessica Kendall-Bar, whose guidance and expertise were instrumental in shaping the direction and success of this project.

We would like to also acknowledge and extend gratitude for the time and guidance provided by Dr.Altintas throughout our time during the Masters program and completion of this capstone project.

References

[2023] JM Kendall-Bar, TM Williams, R Mukherji*, DA Lozano*, JK Pitman, RR Holser, T Keates, RS Beltran, PW Robinson, DE Crocker, T Adachi, OI Lyamin, AL Vyssotski, & DP Costa. Brain activity of diving seals reveals short sleep cycles at depth. ***Science.*** 380(6642). DOI: 10.1126/science.adf0566.

[2023] JM Kendall-Bar. Lessons from sleep in the deep: Records of seal sleep at sea reveal extreme sleep duration flexibility. ***Science.*** 380(6642). DOI: 10.1126/science.adl4885.