Project 1: Supervised Learning Algorithms

Sarah Goodyear

Part 1: Summary Review of Supervised Learning Algorithms

Supervised Learning is the simplest type of machine learning where algorithms learn from labeled training data and map features to known output variables. The goal of these algorithms is to model a dataset to generalize well on unseen data. Both classification and regression tasks can be performed by supervised models. Seven of the most common supervised learning models include linear regression, logistic regression, k-nearest-neighbors, support vector machines, decision trees, random forests, and naïve bayes.

Linear regressions use the equation of a straight line to model the relationship between input and output variables to solve regression problems for quantitative datasets. The algorithm identifies coefficients for the line equation by minimizing the mean squared error. This simple, easily interpretable model works well for linear relationships. However, it is sensitive to outliers and assumes linearity, narrowing the range of potential use cases. Scenarios where a linear model may be a strong choice is for predicting sales and measuring stock market trends.

Logistic regressions work with quantitative and qualitative data to perform classification tasks by using the logistic function to model the probability of a binary result. This is computationally efficient and works well when classes are linearly separable. These models struggle with data that is not as easily separable due to their sensitivity of correlated features. This model is ideal for scenarios such as disease prediction and spam classification.

K-Nearest Neighbors (kNN) can perform both classification and regression tasks. Quantitative data is ideal for these models, but it is possible for them to handle categorical data as well. KNN models classify data points by measuring the Euclidean distance for k number of nearest neighbors. In regression settings, the averages of the k-nearest neighbors are used to determine the class. This model type is simple and intuitive, while being able to model complex decision boundaries. However, this leads to computationally expensive inference time, sensitivity to feature selection, and the decision of the k value to be highly impactful. For classification, kNN is ideal for recommendation systems. For regression, it is ideal for predicting house prices based on similar houses.

Support Vector Machines (SVM) works for both classification and regression problems by directly using quantitative data, though qualitative data can be applied with transformations. This model identifies the optimal hyperplane to maximize the margin between classes, using kernel functions to model non-linear relationships. SVM is effective for high-dimensional data and resistant to overfitting with tuning. However, it is computationally expensive for large datasets and requires tuning of kernel parameters. Scenarios where SVM is ideal include image classification and stock price prediction.

Decision Trees work with both quantitative and qualitative data on classification and regression tasks to split data into subsets based on features. For classification, either gini impurity or entropy is used, while regression uses mean squared error. These models are easily interpretable, don't require scaling or normalization, and handle missing data well. However, they are prone to overfitting and sensitive to small changes in data. Scenarios where decision trees work well is medical diagnosis and predicting house prices based on many factors.

Random Forest models are an extension on decision trees that work for both classification and regression with quantitative and qualitative data. This mode uses multiple decision trees trained on random subsets of data and features via bagging to aggregate predictions. This results in a majority dictating classification predictions and averages defining regression predictions. The use of multiple trees minimizes overfitting risk compared to one tree. This works well for large datasets and can handle missing data well. However, this is more computationally expensive than decision trees and harder to interpret. These models are ideal for fraud detection and customer churn prediction.

Naïve Bayes models are a family of classifiers based on Bayes' theorem that assumes features are independent. Overall, these models work well on high dimensional data, require little training, and are

computationally efficient. The three most common types are Gaussian Naïve Bayes (GNM), Multinomial Naïve Bayes (MNB), and Bernoulli Naïve Bayes (BNB). GNM works well with quantitative data that follows a Gaussian distribution and is ideal for scenarios such as weather classification. MNB works best with qualitative data, as it is designed for discrete count data such as word frequencies. For this reason, it is commonly used for natural language processing tasks such as spam filtering. BNB performs classification tasks using binary categorical data by modeling features as binary occurrences. This works well for scenarios such as binary text classification.

Each of these seven algorithms has their own unique strengths, weaknesses, and use cases making them valuable in different scenarios. All the discussed models are applicable for both classification and regression tasks, except linear regression for only regression tasks and logistic regression and naïve bayes for only classification tasks. Selecting the best algorithm for a given scenario depends on factors such as dataset size, feature distribution, computational efficiency requirements or limitations, and desired interpretability. Understanding these algorithms allows for the most appropriate model selection in application.

Part 2: Application of Supervised Learning Algorithms

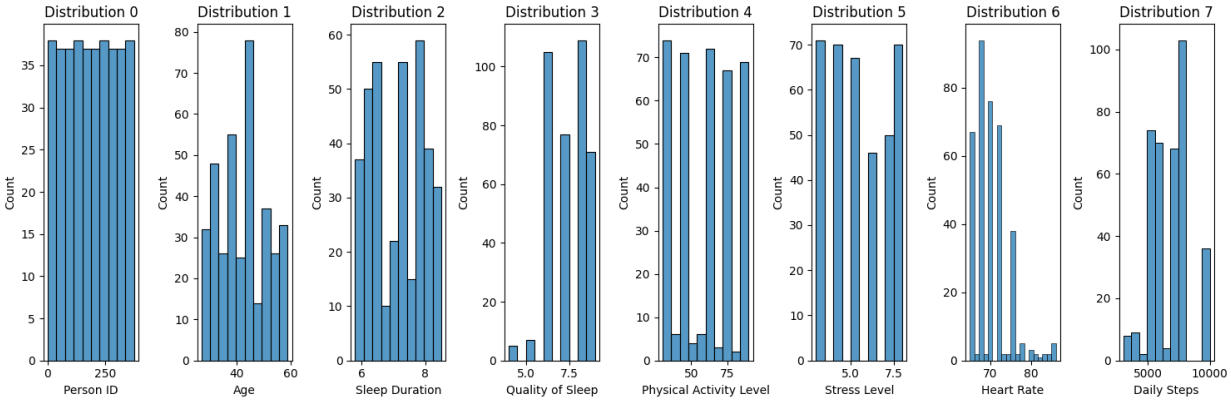
Step 1: Data Collection

The data used for this study is titled "Sleep Health and Lifestyle Dataset" and uploaded to Kaggle on February 25, 2024 by Laksika Tharmalingam. This dataset consists of synthetic data produced using algorithms to simulate results to allow for unconstrained public analysis, stored in CSV format. There are 374 samples with data for 13 features relating to sleep, lifestyle, and physiological characteristics.

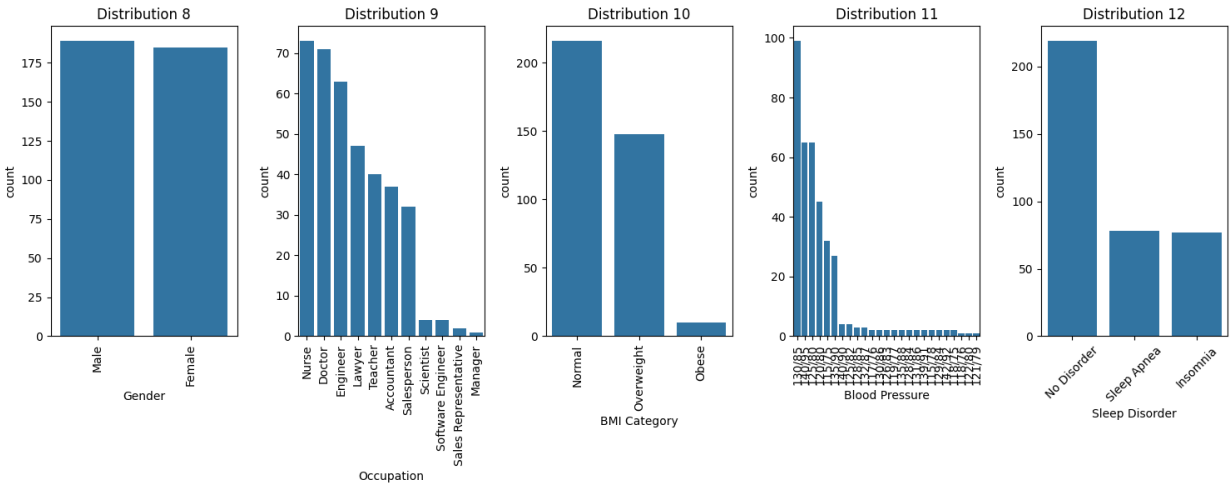
Step 2: Data Preprocessing and Visualization

Examining the contents of this dataset reveals all samples are unique and do not have any missing data. However, the python data frame interprets the "None" in "Sleep Disorder" column, representing no sleep disorder as clarified in the metadata, as a NaN value. Replacing these incorrectly labeled NaN values in the "Sleep Disorder" column with "No disorder" resolves this misinterpretation.

Histograms were used to visualize the distribution of individual numeric features. This, along with individual statistical summaries, reveals any possible skew or abnormalities. Age (Distribution 1) is spread in approximately a bell curve from the minimum age 27 to maximum age 59. Sleep duration (Distribution 2) is appropriately spread from 6.4 hours to 8.5 hours. Quality of sleep (Distribution 3) is left skewed from the minimum at a rating of 4 to maximum of 9, with the lower to upper quartiles falling from 6 to 8 and averaging 7. Physical activity level (Distribution 4) is about evenly spread from 30 to 90 minutes, averaging at 60 minutes. Stress level (Distribution 5) is relatively evenly spread from ratings 3 to 8 and averaging 5, on a scale of 10. Heart rate (Distribution 6) is right skewed with a minimum of 65 to maximum of 86, averaging at 70 beats per minute. Daily steps (Distribution 7) are slightly left skewed with a minimum of 3000, maximum of 10000, lower quartile of 5600, upper quartile of 8000, and average of 8000 steps per day. Person ID (Distribution 0) is an arbitrary label representing distinct subjects and is not relevant to any numeric analysis. Of the other seven features, none of the distributions have any notable anomalies that reasonably risk deviated results.

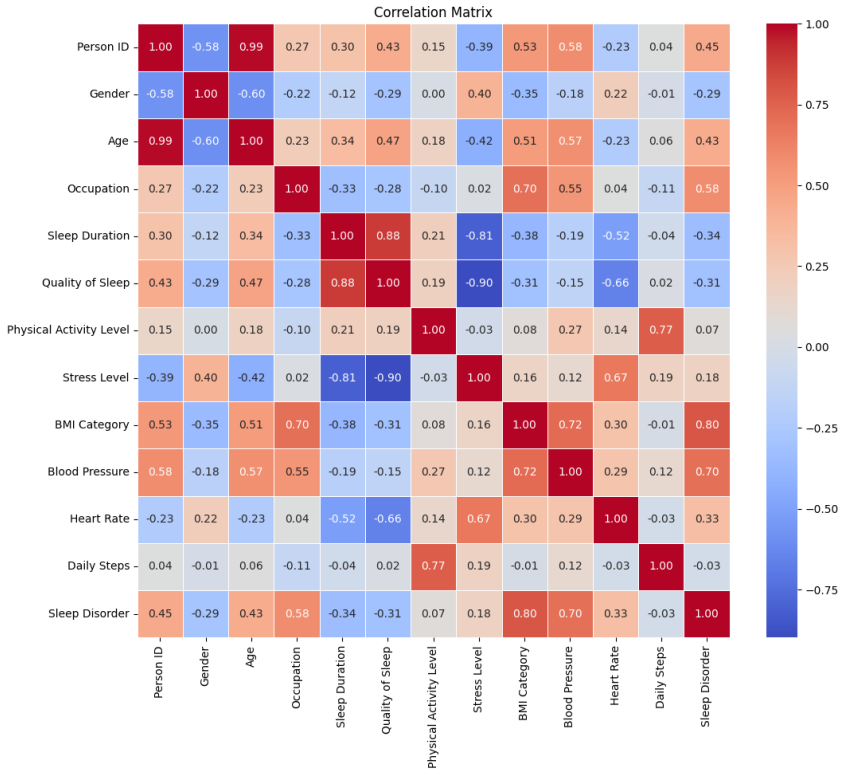


Similarly, boxplots were used to visualize the distribution of categorical features to examine possible anomalies. Gender (Distribution 8) is evenly distributed with only four more male samples than female samples. Occupation (Distribution 9) consists of nurse, doctor, engineer, lawyer, accountant, scientist, software engineer, sales representative, and manager, in that order of prevalence. BMI category (Distribution 9) has the most occurrences of normal weight, followed by overweight then obese, making this most representative of a healthy weighted population. However, normal weight is represented by both “Normal” and “Normal Weight,” which can be combined to “Normal.” Blood pressure (Distribution 11) has the most occurrences of 130/85, followed by 140/95, 125/80, and 120/80 which account for the majority of these observations. This means the data is most representative of a population with slightly elevated blood pressure. Sleep disorder (Distribution 12) measures whether the subject has no disorder, sleep apnea, or insomnia, with no disorder being about three times as common in the population as either disorder. Combining “Sleep Apnea” and “Insomnia” to represent “Sleep Disorder,” when a sleep disorder of any type is present, allows for prediction of the presence of a sleep disorder instead of focusing on specific types. Like the numeric features, no major anomalies are present.



After cleaning the data, dummy variables were created for the categorical features so both numeric and categorical features could be normalized. They were normalized on a scale of 0 to 1. Normalization was more ideal than standardization for this study to limit the impact of the skew and any outliers present in the features, by placing the values in a fixed range. Normalization also allows the data to be more easily interpreted by a wider audience, as well as having the potential to perform better on a wider range of machine learning models. Particularly, k-Nearest-Neighbors (kNN) and many unsupervised techniques are more applicable when data is fixed on a bounded scale. Normalizing the data allows for a more thorough analysis of supervised techniques and prepares for a follow-up analysis and comparison with unsupervised techniques.

Next, a correlation matrix was developed to aid in feature selection. Person ID is included in the matrix, but these metrics should be disregarded in analysis since this feature is an arbitrary value to distinguish uniqueness and holds no statistical significance. The warmer the tone, the more positive the correlation and the cooler the tone, the more negative the correlation. Age, occupation, physical activity level, stress level, BMI category, blood pressure, and heart rate are observed to have positive correlations with sleep disorder. Conversely, gender, sleep deprivation, quality of sleep and daily steps are observed to have negative correlations with sleep disorder.



Step 3: Research Questions and Feature Selection

With this data, several research goals can be identified. First, cross-validation can be used to explore what supervised machine learning models have the most potential to most accurately predict the presence of a sleep disorder. Using these results, the three models with the highest potential can be selected to train more robust models on. This can be used to better gauge what machine learning model can most effectively predict the presence of a sleep disorder with lifestyle and physiological data.

Using the correlation matrix, correlation metrics were compared with the target variable, sleep disorder, to determine which features to include. Any correlation with the person ID is disregarded. A threshold of 0.4 was selected which accounts for the inclusion of four features: age, occupation, BMI category, and blood pressure.

Step 4: Model Building and Comparison

The three models selected for training based on cross validation results were the decision tree, random forest, and k-nearest-neighbors (kNN). For each of these models, parameter optimization was performed prior to training. Each model was trained using the same four aforementioned selected features, as well as the same train-test split. The test size was 20% of the data, leaving 80% for training, and the random state was set to 42. After each model was trained, predictions were made using the test set, the accuracy was computed, confusion matrices were produced, classification reports were printed, and auROC curves were created. These evaluation metrics were stored for each model for final comparison.

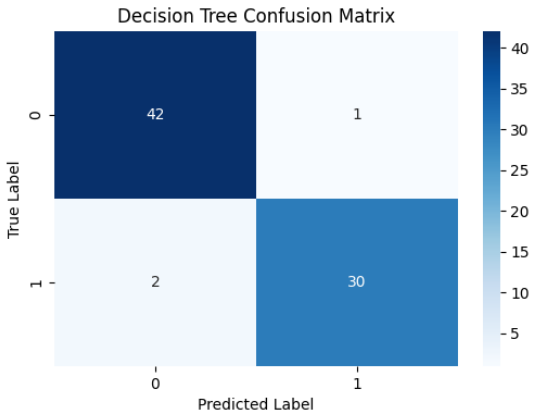
To evaluate the results of each model, a confusion matrix and area under the receiving operating characteristics curve (auROC) were developed. Classification reports showing precision, recall, and f1-scores were

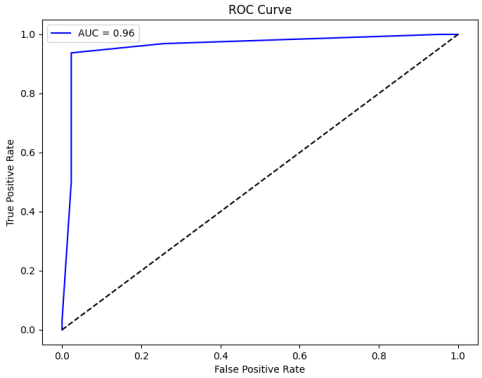
also created. These metrics give insight on how well each model can distinguish between classes and what classification types are accounting for the error in predicting the test set.

Decision Tree

Prior to training the model, grid search was used for parameter optimization. The optimized parameters were maximum depth, minimum samples split, minimum samples leaf, and criterion. Maximum depth was tested for 3, 5, 10, and none. Minimum samples split was tested for 2, 5, and 10. Minimum samples leaf was tested for 1, 2, and 4. Criterion which controls the level of impurity was optimized for the gini index or entropy. The selected parameters based on grid search optimization were 3, 2, 1, and gini, respectively.

Based on the confusion matrix, the decision tree model had both high precision and recall, with slightly higher likelihood of a false negative result than a false positive result. This results in a high f1 score of 0.95, as well as a strong ROC curve with an AUC of 0.961.

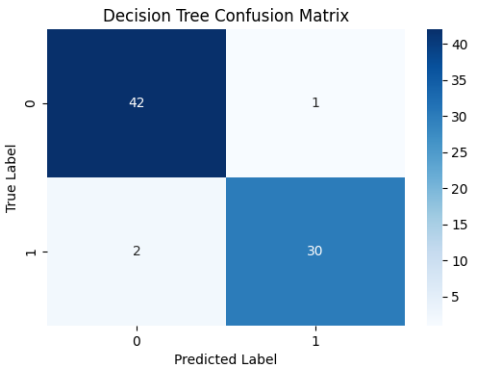


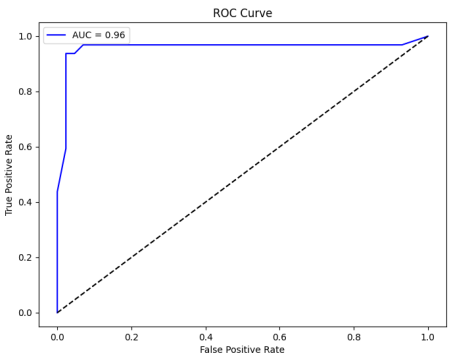


Random Forest

Similar to the decision tree parameter optimization, the random forest parameters were optimized prior to training using grid search. The five optimized parameters were n estimators, maximum depth, minimum samples split, minimum samples leaf, and criterion. N estimators was tested for 50, 100, and 200. Maximum depth was tested for 5, 10, 20, and none. Minimum samples split was tested for 2, 5, and 10. Minimum samples leaf was tested for 1, 2, and 4. Criterion was optimized for the gini index or entropy. The selected parameters were 100, 5, 2, 1, and gini, respectively.

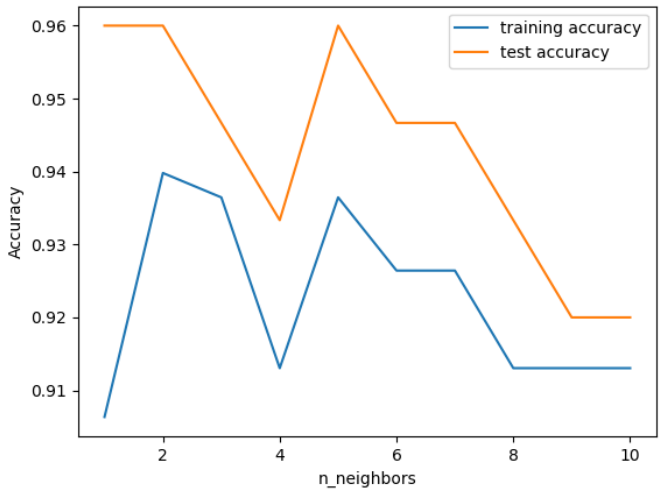
Like the decision tree, the confusion matrix indicates that the random forest model had both high precision and recall, with false negatives being slightly more prone than false positives. An identical f1 score was achieved of 0.95, while a slightly inferior, though still strong, ROC AUC score of 0.958 was achieved.



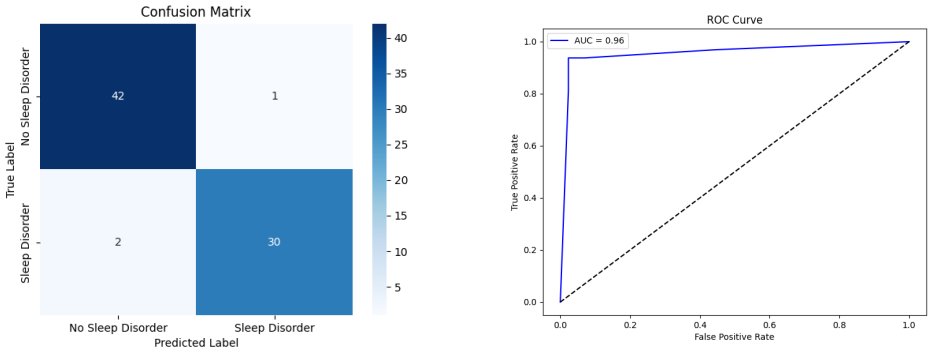


K-Nearest-Neighbors (kNN)

Alternative to the decision tree and random forest models, kNN was only optimized for one parameter, the number of neighbors. K neighbors classifier was used to test what value of k best maximized both training and testing accuracies on a range from k=1 to k=10. The results of this were plotted on a line graph, pictured below. The orange line represents test accuracy, and the blue line represents training accuracy. The simultaneous peak of the two plots at k=5 shows that that is where both accuracies are maximized, indicating k=5 is the most optimal parameter.



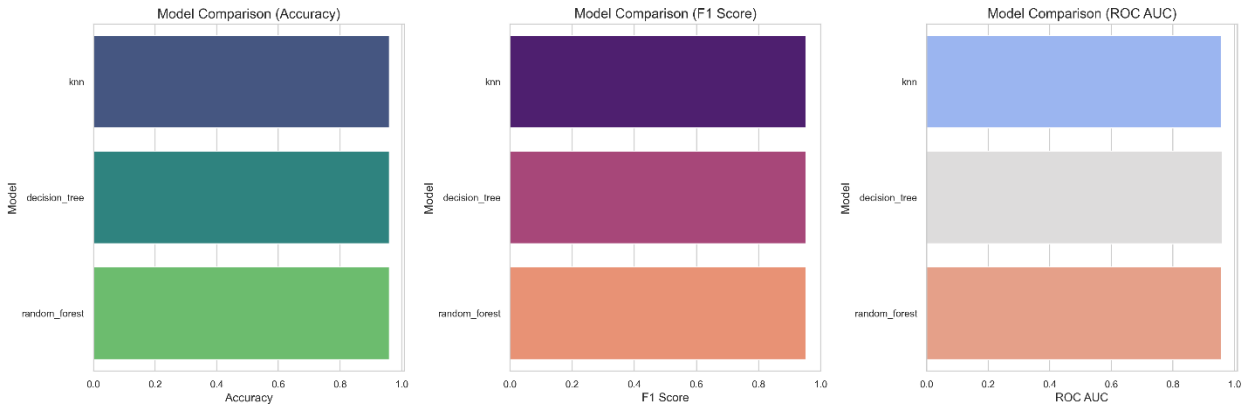
Analysis of the kNN model indicates similar performance to the decision tree and random forest. Again, confusion matrix results show a slightly higher tendency to present false negatives than false positives, though precision and recall overall are high. The kNN model had an identical f1-score of 0.95 to the other models. However, the kNN model achieved a slightly lower ROC AUC score of 0.957.



Results

Given the statistically close performance of these models, the ROC AUC metric is the most suitable for comparison. This metric summarizes the auROC curve into a single metric, most amplifying the small differences between the other model metrics. The accuracy, f1 score, and ROC AUC score for the kNN, decision tree, and random forest models are shown in the table below. As mentioned, the accuracy and f1 scores for these models are so similar there is no statistically significant difference, if there is one at all. This is further highlighted in the bar plots below where the difference in bar length is not visually apparent for any of the metrics, including ROC AUC. Though the ROC AUC scores are close, they do highlight the minimal differences in performance as seen in the last column of the table. These scores indicate that the decision tree model performs the best, followed by random forest and then kNN.

	Accuracy	F1 Score	ROC AUC
kNN	0.96	0.952	0.957
Decision Tree	0.96	0.952	0.961
Random Forest	0.96	0.952	0.958



Step 5: Conclusion and Recommendation

The results indicate that the decision tree, random forest, and kNN models all performed well in predicting the presence of a sleep disorder using lifestyle and physiological data. Given the close results, the ROC AUC score, being the most sensitive metric to variation in performance, is the best metric to distinguish between the models. Each had high accuracy and f1-scores, but the decision tree model performed the best based on ROC AUC scores with a score of 0.961, though the others closely follow. This makes the decision tree model the most optimal recommendation for predicting the preens of a sleep disorder with physiological and lifestyle data. Further study is required to determine how predictive power could be maximized with other model types or datasets that are different in size or features.