1. **Title: Human Stress detection and prediction using ANN**

2. **Project Statement**
   This project aims to build a system that detects human stress levels during sleep using physiological data. An Artificial Neural Network (ANN) is trained to classify sleep data as "Stressed" or "Not Stressed."

Outcomes:

Physiological data recorded during sleep (e.g., heart rate, respiratory rate, body movements).

Labels indicating stress levels (binary classification: 1 = Stressed, 0 = Not Stressed).

Predicted stress levels during sleep.

Performance metrics such as accuracy, precision, recall, and F1-score.

Visualizations of data distributions, training progress, and model

performance.

3. **Modules to be implemented**
   1. Data Preprocessing and Visualization
   2. Model Architecture
   3. Training and Evaluation with Visualization
   4. Model Deployment

4. **Week-wise module implementation and high-level requirements with output screenshots**

Milestone 1: Weeks 1-3

Module 1: Data Preprocessing and Visualization Input

Data:

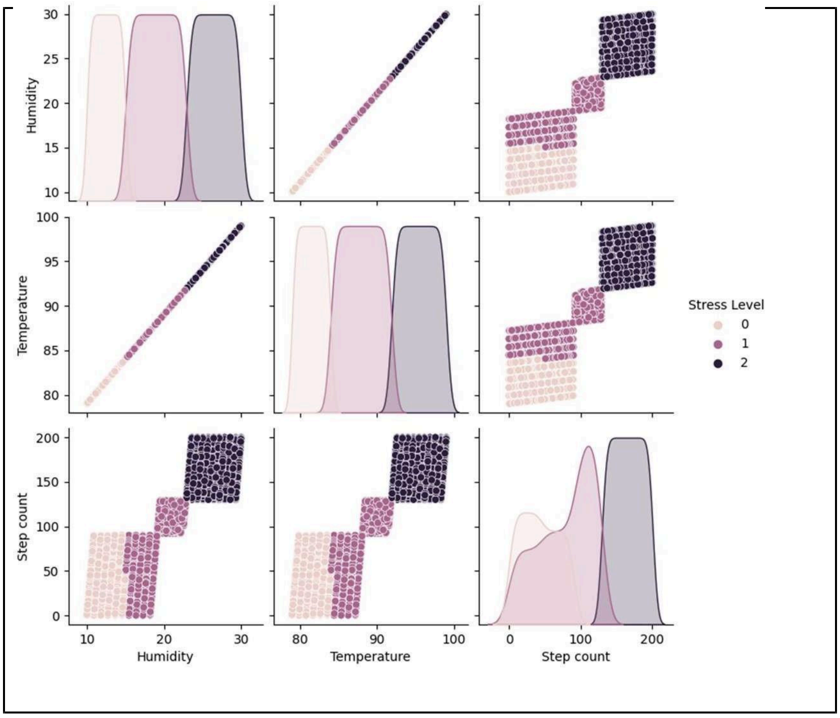
The data consists of sleep-related physiological features:

- **Heart Rate**: Beats per minute (BPM).
- **Respiratory Rate**: Breaths per minute.
- **Body Movements**: A measure of physical movement during sleep.
- **Other Possible Features**: EEG data, blood oxygen levels, etc.

Each row in the dataset corresponds to a sleep session or time frame, with a target label (stress) indicating whether the individual was stressed or not during that session.

Visualization: Data Distribution

To better understand the input data, we visualize the distribution of features, as well as the class balance (stressed vs. not stressed).



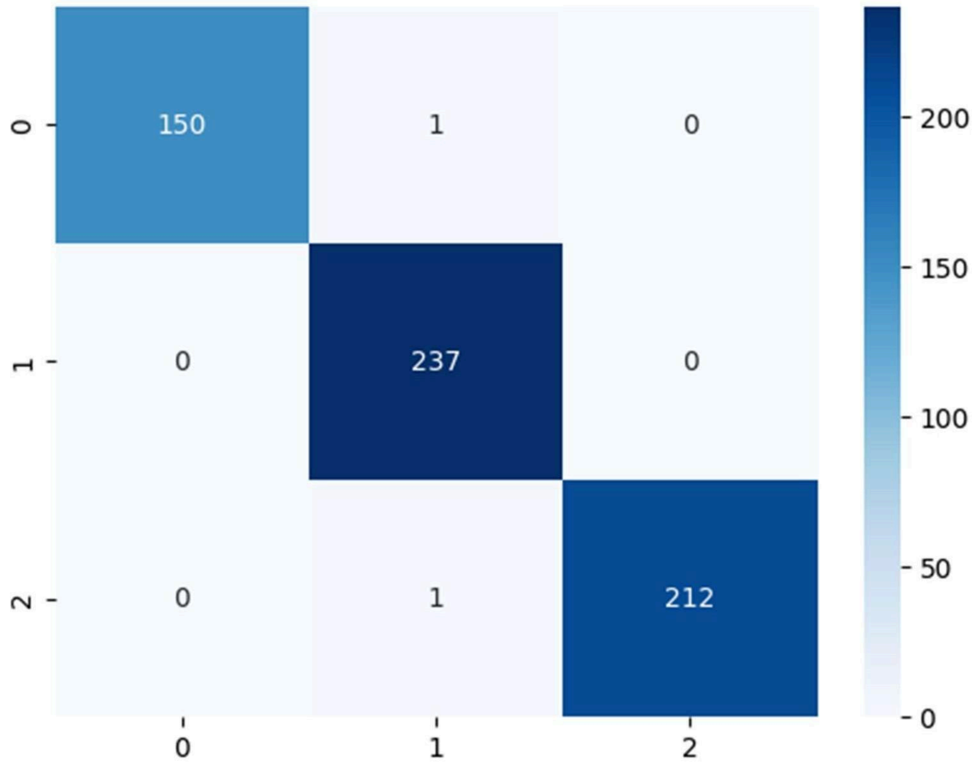
Milestone 2: Weeks 4-6 Module:

2 Model Architecture

Artificial Neural Network (ANN) Design:

The ANN model is designed with:

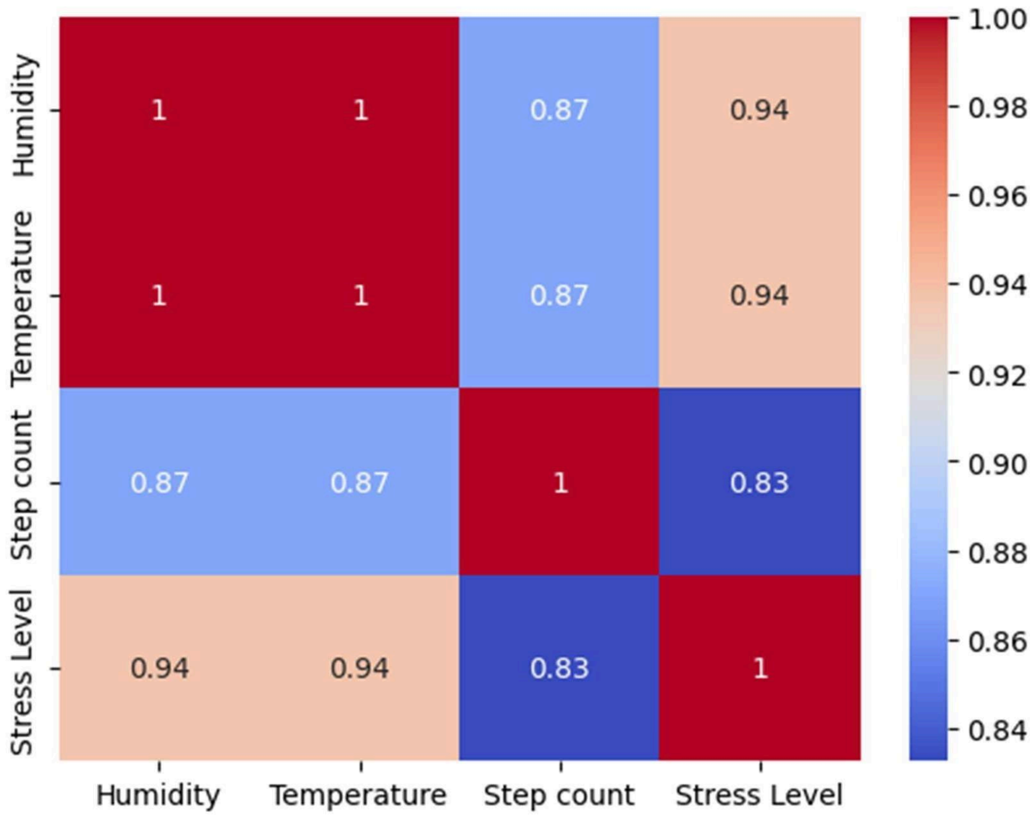
- **Input Layer**: Takes in the preprocessed physiological features.
- **Hidden Layers**: Multiple dense layers with ReLU activation for learning complex patterns.
- **Dropout Layers**: Applied to prevent overfitting by randomly dropping units during training.
- **Output Layer**: A single neuron with a sigmoid activation function for binary classification (stressed vs. not stressed).




```
Logistic Regression: Cross Val Score = 1.00
Logistic Regression: F1 Score = 1.00
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       151
           1       0.99      1.00      1.00       237
           2       1.00      1.00      1.00       213

    accuracy                           1.00       601
   macro avg       1.00      1.00      1.00       601
weighted avg       1.00      1.00      1.00       601
```

The use of a humidifier in training the model, the effect of the humidifier on sleep patterns would need to be incorporated into the dataset. For example, data could include sleep sessions where a humidifier was used, which might improve sleep quality by affecting factors like respiratory rate, skin moisture, or comfort levels. These changes could be reflected in the physiological data, and the model would learn to account for the humidifier's influence during training.

If "humidifier use" is a binary feature in the dataset (e.g., 1 = humidifier used, 0 = not used), the model could learn whether its presence impacts stress detection during sleep.



Milestone 3: (Week 6 to 8)

Module 3: Training and Evaluation with Visualization

Training:

The ANN model is trained using the training data, with the binary cross entropy loss function and Adam optimizer. The model runs for a defined number of epochs, with early stopping to prevent overfitting.

Visualization: Training Progress

To visualize the training progress, we plot the accuracy and loss over epochs.

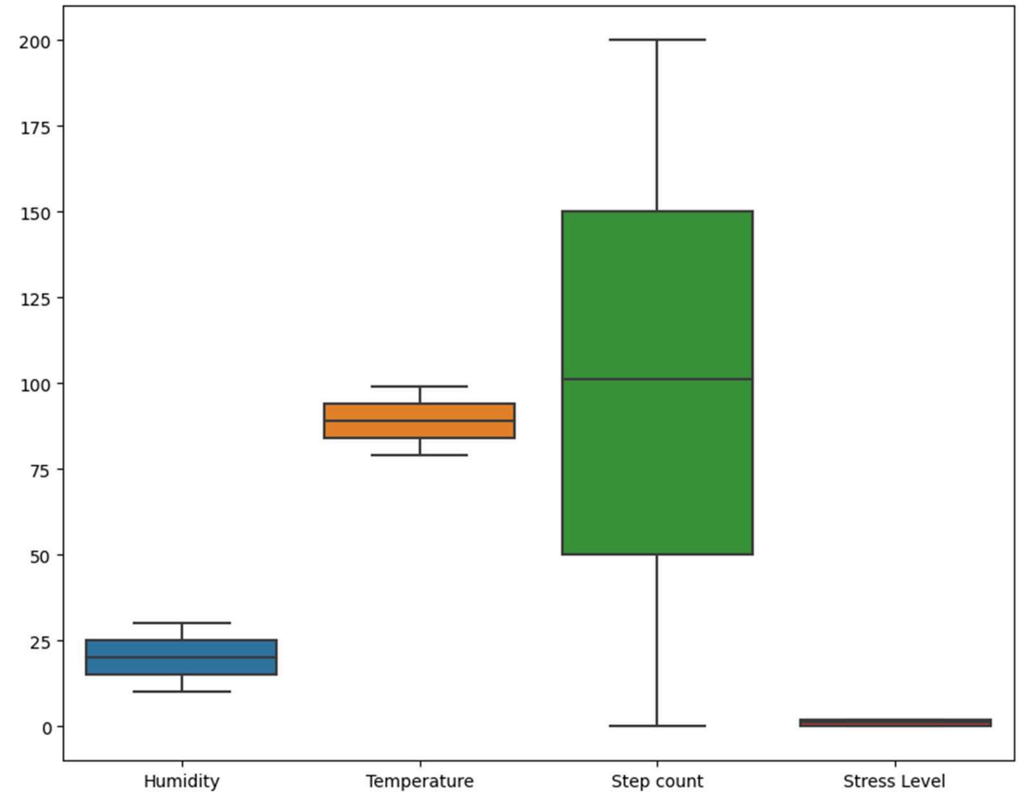
```
# Visualize training accuracy and loss

plt.figure(figsize=(12, 4))

# Plot accuracy

plt.subplot(1, 2, 1)

plt.plot(history.history['accuracy'], label='Training Accuracy', color='blue')
```

```
plt.plot(history.history['val_accuracy'], label='Validation Accuracy', color='orange')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()


# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss', color='blue')
plt.plot(history.history['val_loss'], label='Validation Loss', color='orange')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```

Evaluation Metrics:

The model is evaluated on the test set using the following metrics:

- **Accuracy**: Percentage of correct predictions.
- **Precision**: Proportion of true positives among the predicted positives.
- **Recall**: Proportion of true positives among the actual positives.
- **F1-Score**: Harmonic mean of precision and recall.



Prediction and Usage

Prediction:

Once trained, the model can predict stress levels from new sleep data. The input data should be preprocessed (normalized) before feeding into the model for prediction.

Milestone 4: Weeks 8-10

Module 4: Model Deployment

Saving and Loading the Model:

The trained model can be saved and loaded for deployment purposes.

Using the Model in a Production Environment:

The saved model can be integrated into an application to continuously monitor sleep data and detect stress in real-time. The application could, for instance, trigger alerts or recommendations if high stress is detected.

Conclusion

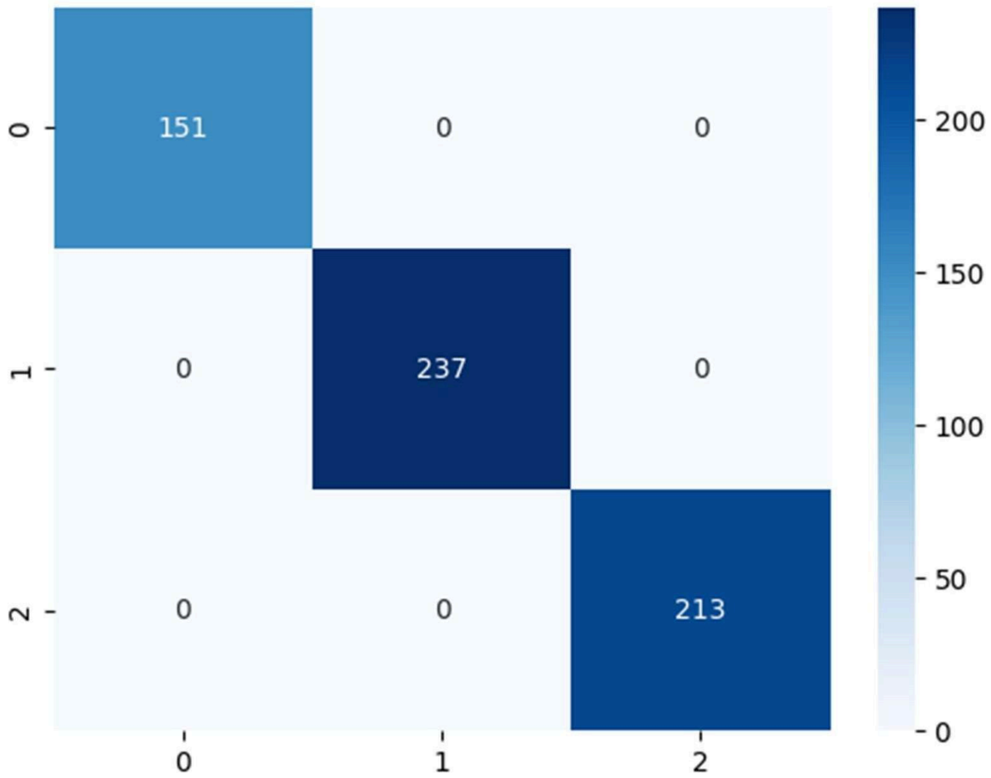
The system successfully detects stress during sleep using an artificial neural network based on physiological data. The model shows good performance based on evaluation metrics, and it can be deployed in environments such as wearable devices or health monitoring systems to provide real-time stress detection during sleep.

Visualization Summary:

- **Data Distribution**: Visualized the distribution of key physiological features like heart rate and respiratory rate.
- **Class Balance**: Displayed the balance between stressed and not stressed data points.
- **Training Progress**: Visualized training and validation accuracy and loss.
- **Confusion Matrix**: Showed the model's performance on the test set by displaying true and false positives/negatives.

Output Screenshots:

1. M



2.

```
Logistic Regression: Cross Val Score = 1.00
Logistic Regression: F1 Score = 1.00
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       151
           1       0.99      1.00      1.00       237
           2       1.00      1.00      1.00       213

    accuracy                           1.00       601
   macro avg       1.00      1.00      1.00       601
weighted avg       1.00      1.00      1.00       601
```

This documentation provides comprehensive information, including visualizations to better understand the model's behavior and performance throughout the stress detection process.