

## Project Development Phase

### Model Performance Test

|               |   |
|---------------|---|
| Date          | 21 November 2023  |
| Team ID       | Team-593059   |
| Project Name  | The Sleep Oracle Anticipating Health and Lifestyle Through Data |
| Maximum Marks | 10 Marks  |

### Model Performance Testing:

In the project , The Sleep Oracle Anticipating Health and Lifestyle Through Data we are using 4 Classification Model which are

1. Logistic Regression
2. XGB boost
3. Decision Tree
4. Random Forest
5. Random forest ( Hyper Parameter Tuning )

| S.No.        | Parameter | Values  | Screenshot  |         |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
|--------------|-----------|---|---|---------|-----------|--------|----------|---------|---|------|------|------|----|---|------|------|------|----|---|------|------|------|----|----------|--|--|------|----|-----------|------|------|------|----|--------------|------|------|------|----|
| 1.           | Metrics   | <p><b>Classification Model (Confusion Matrix, Accuracy Score, and Classification Report ) :</b></p> <p><b>1. Logistic Regression :</b><br/>Accuracy Score : 0.66666</p> <p><b>2. XGB boost :</b><br/>Accuracy Score : 0.90666</p> <p><b>3. Decision Tree :</b><br/>Accuracy Score : 0.90666</p> <p><b>4. Random Forest :</b><br/>Accuracy Score : 0.90666</p> | <p><b><u>Logistic Regression :</u></b></p> <pre>[74] accuracy = accuracy_score(y_test, y_pred)       conf_matrix = confusion_matrix(y_test, y_pred)       classification_rep = classification_report(y_test, y_pred)</pre> <pre>[75] accuracy  0.6666666666666666</pre> <pre>[76] confusion_matrix(y_test,y_pred)  array([[ 7, 13,  0],        [ 3, 37,  1],        [ 0,  8,  6]])</pre> <pre>print(classification_rep)</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.70</td><td>0.35</td><td>0.47</td><td>20</td></tr><tr><td>1</td><td>0.64</td><td>0.90</td><td>0.75</td><td>41</td></tr><tr><td>2</td><td>0.86</td><td>0.43</td><td>0.57</td><td>14</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.67</td><td>75</td></tr><tr><td>macro avg</td><td>0.73</td><td>0.56</td><td>0.60</td><td>75</td></tr><tr><td>weighted avg</td><td>0.70</td><td>0.67</td><td>0.64</td><td>75</td></tr></tbody></table> |         | precision | recall | f1-score | support | 0 | 0.70 | 0.35 | 0.47 | 20 | 1 | 0.64 | 0.90 | 0.75 | 41 | 2 | 0.86 | 0.43 | 0.57 | 14 | accuracy |  |  | 0.67 | 75 | macro avg | 0.73 | 0.56 | 0.60 | 75 | weighted avg | 0.70 | 0.67 | 0.64 | 75 |
|              | precision | recall  | f1-score  | support |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 0            | 0.70      | 0.35  | 0.47  | 20      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 1            | 0.64      | 0.90  | 0.75  | 41      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 2            | 0.86      | 0.43  | 0.57  | 14      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| accuracy     |           |   | 0.67  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| macro avg    | 0.73      | 0.56  | 0.60  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| weighted avg | 0.70      | 0.67  | 0.64  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |

## XGB boost :

```
0s ✓ accuracy_score(y_test,y_pred1)
0.9066666666666666
```

```
0s ✓ [85] confusion_matrix(y_test,y_pred1)
array([[16,  2,  2],
       [ 1, 40,  0],
       [ 1,  1, 12]])
```

```
0s ✓ [86] print(classification_report(y_test,y_pred1))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.89      | 0.80   | 0.84     | 20      |
| 1            | 0.93      | 0.98   | 0.95     | 41      |
| 2            | 0.86      | 0.86   | 0.86     | 14      |
| accuracy     |           |        | 0.91     | 75      |
| macro avg    | 0.89      | 0.88   | 0.88     | 75      |
| weighted avg | 0.91      | 0.91   | 0.91     | 75      |

## Decision Tree:

```
0s ✓ [89] accuracy_tree = accuracy_score(y_test, y_pred_tree)
print("Decision Tree Accuracy:", accuracy_tree)
Decision Tree Accuracy: 0.9066666666666666
```

```
0s ✓ [90] classification_report_tree = classification_report(y_test, y_pred_tree)
print("Decision Tree Classification Report:\n", classification_report_tree)
```

```
0s ✓ [91] confusion_matrix(y_test,y_pred_tree)
```

```
array([[16,  2,  2],
       [ 1, 40,  0],
       [ 1,  1, 12]])
```

## Random Forest :

```
✓ [98] accuracy = accuracy_score(y_test, y_pred)
0s conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

```
✓ [99] print(f"Accuracy: {accuracy}")
0s
```

Accuracy: 0.9066666666666666

```
✓ [100] print("Confusion Matrix:")
0s print(conf_matrix)
```

Confusion Matrix:

```
[[16  2  2]
 [ 1 40  0]
 [ 1  1 12]]
```

```
✓ [101] print("Classification Report:")
0s print(classification_rep)
```

Classification Report:

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.89      | 0.80   | 0.84     | 20      |
| 1            | 0.93      | 0.98   | 0.95     | 41      |
| 2            | 0.86      | 0.86   | 0.86     | 14      |
| accuracy     |           |        | 0.91     | 75      |
| macro avg    | 0.89      | 0.88   | 0.88     | 75      |
| weighted avg | 0.91      | 0.91   | 0.91     | 75      |

| 2.           | Tune the Model | <p>Hyperparameter Tuning -</p> <p>Random Forest (Hyper Parameter Tuning ) :</p> <p>Accuracy Score : 0.90666</p> | <div><div>✓<br/>0s</div><div>[109] print(f"Accuracy: {accuracy}")</div><div>Accuracy: 0.9066666666666666</div></div> <div><div>✓<br/>0s</div><div><div>▶</div><div>print("Confusion Matrix:")<br/>print(conf_matrix)</div></div><div><div>➔</div><div>Confusion Matrix:<br/>[[16  2  2]<br/>[ 1 40  0]<br/>[ 1  1 12]]</div></div></div> <div><div>✓<br/>0s</div><div>[111] print("Classification Report:")<br/>print(classification_rep)</div><div>Classification Report:<table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.89</td><td>0.80</td><td>0.84</td><td>20</td></tr><tr><td>1</td><td>0.93</td><td>0.98</td><td>0.95</td><td>41</td></tr><tr><td>2</td><td>0.86</td><td>0.86</td><td>0.86</td><td>14</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>75</td></tr><tr><td>macro avg</td><td>0.89</td><td>0.88</td><td>0.88</td><td>75</td></tr><tr><td>weighted avg</td><td>0.91</td><td>0.91</td><td>0.91</td><td>75</td></tr></table></div></div> |         | precision | recall | f1-score | support | 0 | 0.89 | 0.80 | 0.84 | 20 | 1 | 0.93 | 0.98 | 0.95 | 41 | 2 | 0.86 | 0.86 | 0.86 | 14 | accuracy |  |  | 0.91 | 75 | macro avg | 0.89 | 0.88 | 0.88 | 75 | weighted avg | 0.91 | 0.91 | 0.91 | 75 |
|--------------|----------------|---|---|---------|-----------|--------|----------|---------|---|------|------|------|----|---|------|------|------|----|---|------|------|------|----|----------|--|--|------|----|-----------|------|------|------|----|--------------|------|------|------|----|
|              | precision      | recall  | f1-score  | support |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 0            | 0.89           | 0.80  | 0.84  | 20      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 1            | 0.93           | 0.98  | 0.95  | 41      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| 2            | 0.86           | 0.86  | 0.86  | 14      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| accuracy     |                |   | 0.91  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| macro avg    | 0.89           | 0.88  | 0.88  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |
| weighted avg | 0.91           | 0.91  | 0.91  | 75      |           |        |          |         |   |      |      |      |    |   |      |      |      |    |   |      |      |      |    |          |  |  |      |    |           |      |      |      |    |              |      |      |      |    |