DATA SCIENCE PROJECT

SC1015 SC8 Group 6

- Chen Xin Han
- Peh Yu Ze
- Ong Tsien Jin

DATASETS USED

data.world

"Stress Analysis"

Primary Dataset

"Human Stress Detection in and through Sleep"

Secondary Dataset

kaggle

PROBLEM DEFINITION

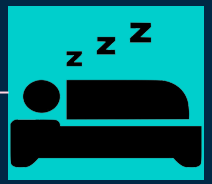
- Main Problem:
  - Our group set out to explore the connection between emotions and stress levels to predict sleep quality.

Using emotions from the primary dataset

Emotions

To predict stress levels from the primary dataset

Stress Levels

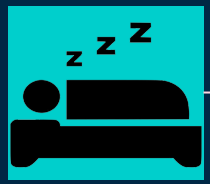


Then to predict sleep quality in the secondary dataset

Sleep Quality

PROBLEM DEFINITION

- Sub-Problem:
  - How can we use numeric sleeping data to predict stress level if emotions are unavailable.



Then to predict sleep quality in the secondary dataset

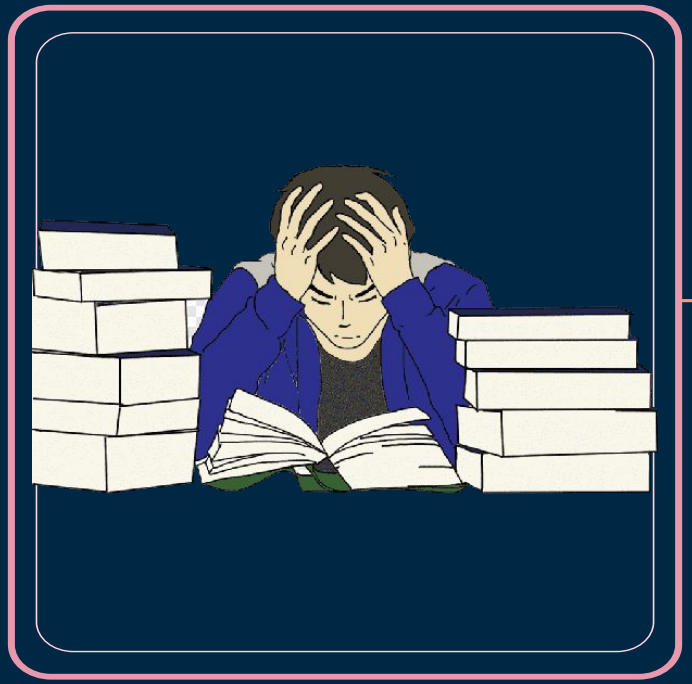
Sleep Quality

To predict stress levels from the primary dataset

Stress Levels

PRACTICAL MOTIVATION

- Society is very fast paced
- Most people will be stressed due to various reasons such as study or work
- Hard for one to be aware of one’s own actual stress levels day to day
- We seek to translate how they feel, into stress level, by asking them about their emotions.

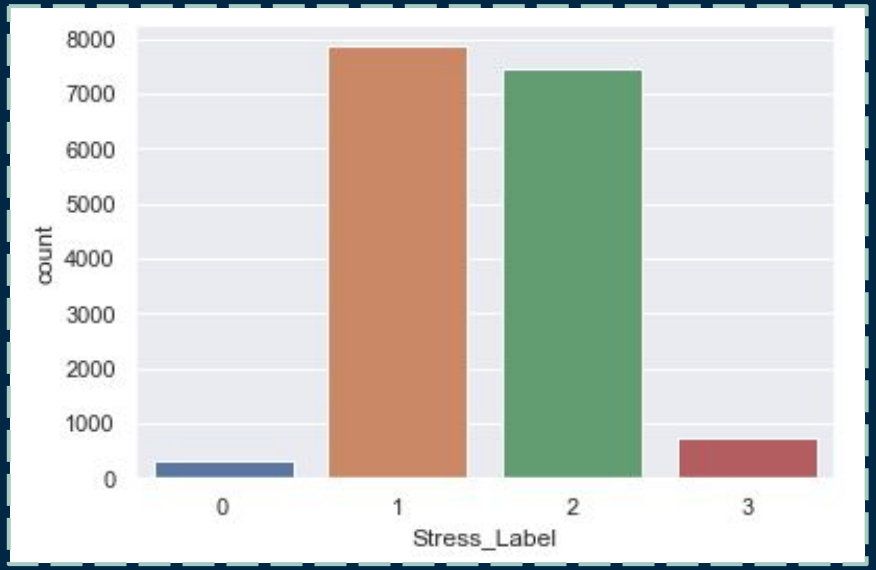


EXPLORATORY DATA ANALYSIS (PRIMARY DATASET)

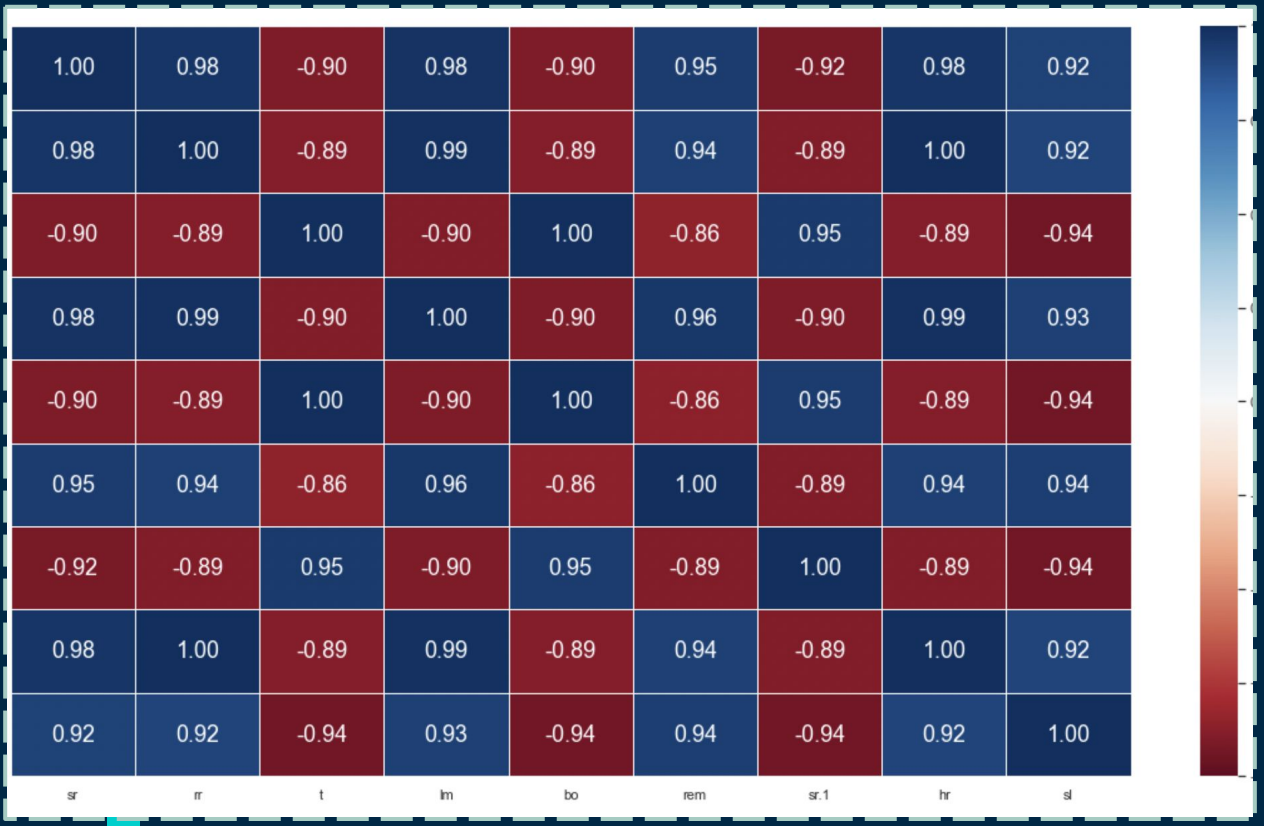
- Correlation between each variable to the stress levels is quite low, at about 0.33
- There is no one emotional factor that affects stress level the most, all the emotion variables are equally important in predicting stress level

EXPLORATORY DATA ANALYSIS (PRIMARY DATASET)

- We can observe that this dataset is biased towards Stress_Label of value 1 and 2 accounting for almost all the data entries
- This may potentially hamper the classification accuracy of our machine models



EXPLORATORY DATA ANALYSIS (SECONDARY DATASET)



- Sleep quality has a strong correlation in predicting stress level
- Each sleep quality variable has correlation from 0.92 to 0.94
- Similar to the primary data set, all variables have the same importance in predicting stress as there is no one factor that affects stress level the most.
-

CLEANING OF DATASETS

PRIMARY DATASET

- Has **4** stress levels
- 0 - no
- 1 - mild
- 2 - moderate
- 3 - severe

SECONDARY DATASET

- Has **5** stress levels
- 0 - low/normal
- 1 – medium low
- 2 - medium
- 3 - medium high
- 4 - high

- By suitability, we re-categorised the **secondary dataset** into **4** levels of stress:
- 0 - low/normal --> 0 - no
- 1 - medium low --> 1 - mild
- 2 - medium, 3 - medium high --> 2 - moderate
- 4 - high --> 3 - severe

CLEANING OF DATASETS

- By suitability, we re-categorised the **secondary dataset** into **4** levels of stress:
- 0 - low/normal --> 0 - no
- 1 - medium low --> 1 - mild
- 2 - medium, 3 - medium high --> 2 - moderate
- 4 - high --> 3 - severe

```
# Re-categorising the 5 different sleep levels into 4, with 0 to 0, 1 to 1, 2/3 to 2, and 4 to 3
sleepData["sl"] = sleepData["sl"].replace([2, 3], 2)
sleepData["sl"] = sleepData["sl"].replace([4], 3)
```

CLEANING OF DATASETS

PRIMARY DATASET

- Labeled the dataset, from text to numeric values to facilitate machine learning
- "No": 0
- "Mild": 1
- "Moderate": 2
- "Severe": 3

```
# encoding the labels

intensity = {
    'No': 0,
    'Mild': 1,
    'Moderate': 2,
    'Severe': 3,
}

newData = {}

for colName, colData in dataAll.iteritems():
    newArr = []
    for item in colData:
        newArr.append(intensity[item])

    newData[f'{colName}_Label'] = newArr

dataLabeled = pd.DataFrame.from_dict(newData)
dataLabeled


### writing data
# dataLabeled.to_csv('../data/labeled_data.csv', index=False)
```

CLEANING OF DATASETS

SECONDARY DATASET

- Converted the column of "Body temperature" in the dataset from fahrenheit to celsius, so that it will be more applicable to us and easier for users to understand and input their data

```
# Converting the body temperature from farenheit to celsius
def convert_to_celsius(x):
    return (x - 32) * 5/9

sleepData["t"] = sleepData["t"].apply(convert_to_celsius)
```

MACHINE LEARNING

PRIMARY DATASET

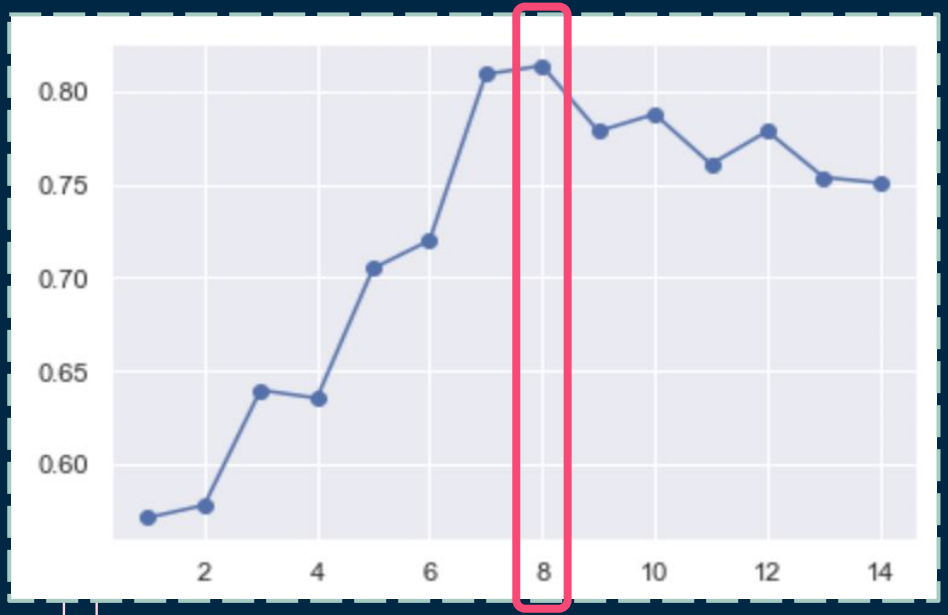
- Uses categorical variables (emotion levels) to classify stress levels
- **Decision Trees**
- **Support Vector Machines (SVMs)**

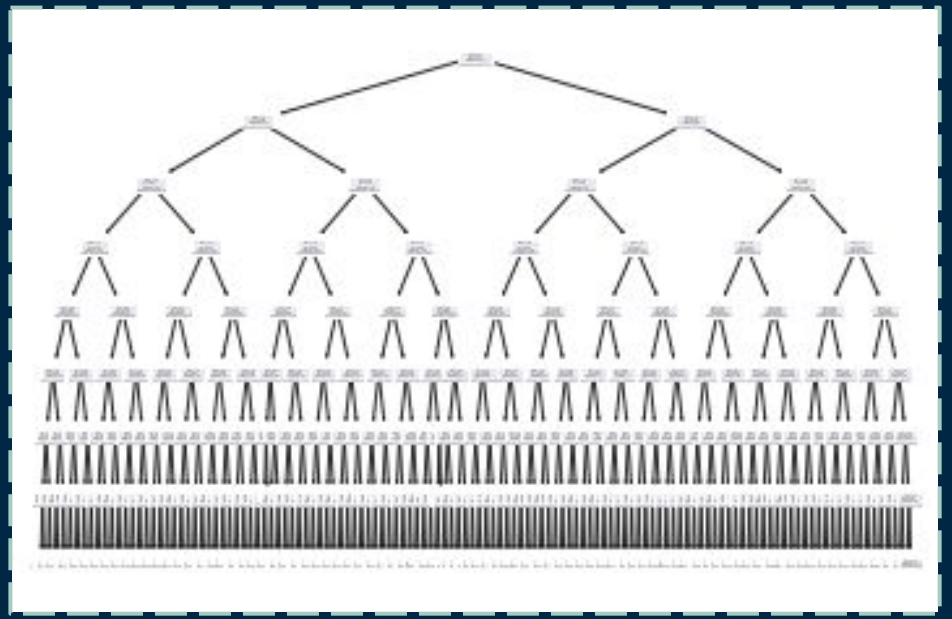
SECONDARY DATASET

- Uses numerical variables during sleep
- **Linear Regression**

MACHINE LEARNING (DECISION TREES)

- To determine the optimal depth for the decision tree for our primary data set





MACHINE LEARNING (DECISION TREES)

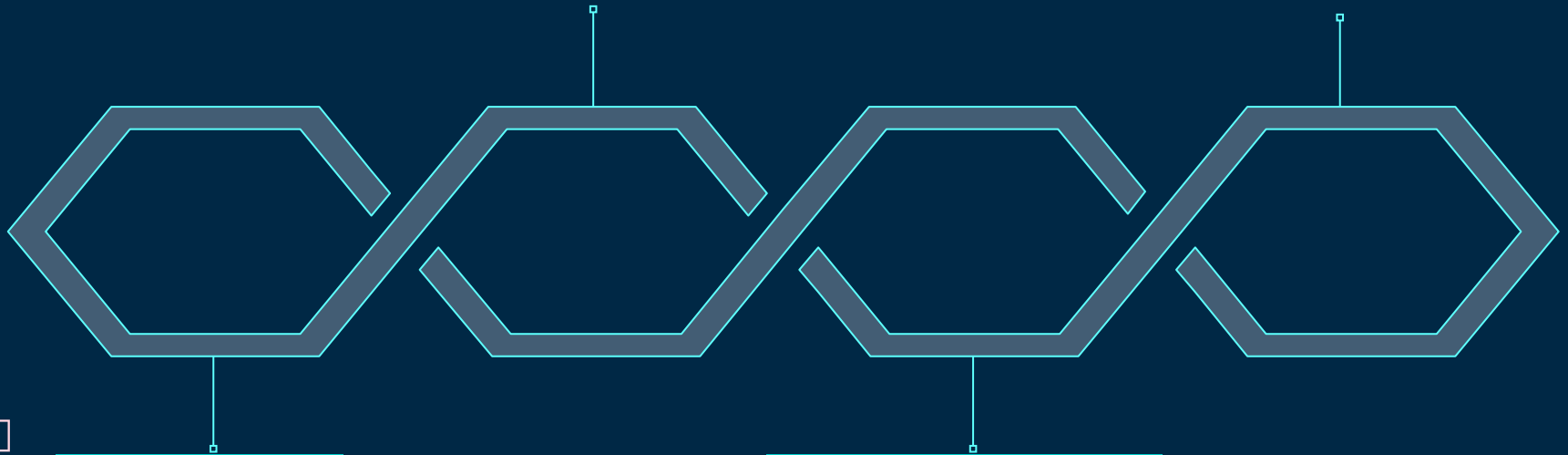
- Classification accuracy was not enough (**~0.79**)
- May not be the best model for our data set as decision trees only take into account one variable at every decision stage instead of considering all of the variables at once
- Therefore we went to find a better model for our dataset, and we found **Support Vector Machines (SVMs)**

```
print(f"Best accuracy: {max(list)}\nDepth: {list.index(max(list))+1}")
```

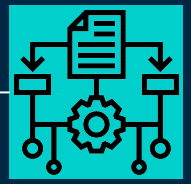
```
Best accuracy: 0.7924931339639915
Depth: 8
```

MACHINE LEARNING (SVMs)

- Form hyper-planes or a set of hyper-planes in an infinite dimensional space to classify data points into labeled subspaces



WHY WE CHOOSE SVC LINEAR (LIN) KERNEL?



01

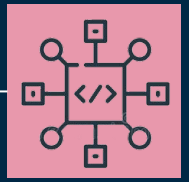
DECISION TREE

SVC LIN Kernel takes into more factors instead of just **one** at every decision stage

02

LINEAR SVM

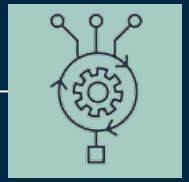
SVC LIN Kernel has an allowable margin of error which in turn results in a higher accuracy rating



03

RBF KERNEL

Complexity of the RBF model grows with the size of the data, resulting in it being more expensive than Linear Kernel in the long run



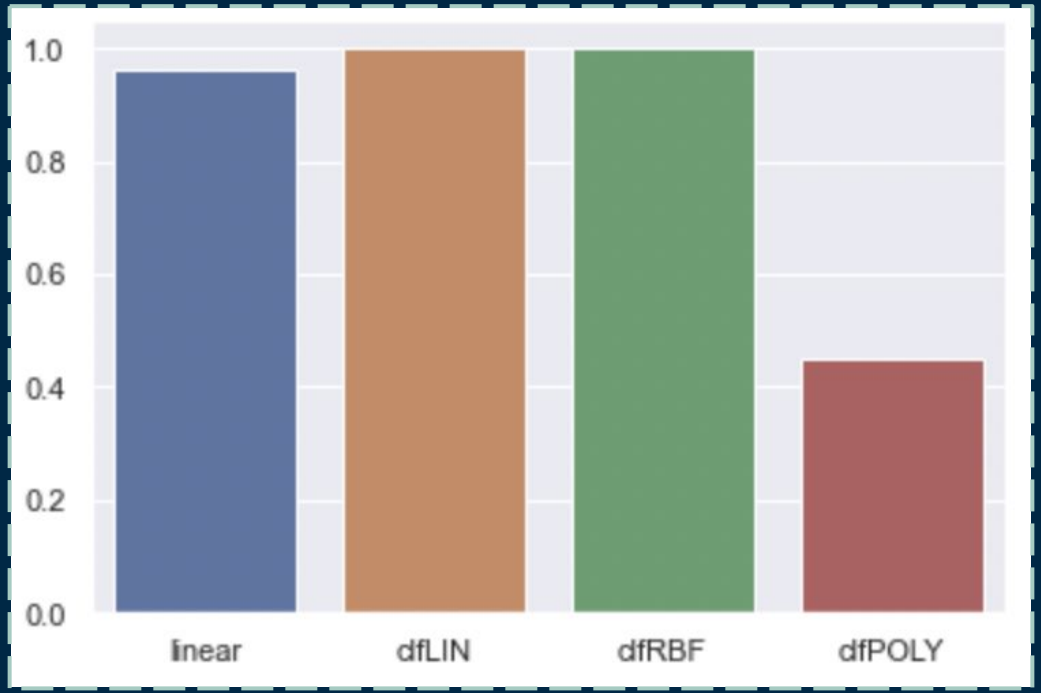
04

POLY KERNEL

SVC LIN Kernel is significantly less time and space consuming as compared to poly kernel

WHY WE CHOOSE SVC LINEAR (LIN) KERNEL?

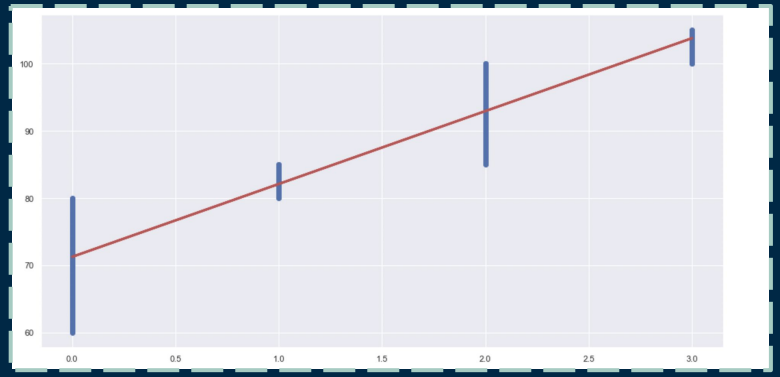
- No surprise that it has the **highest** classification accuracy of **1.0**, the highest among all the other models, and on par with SVC RBF Kernel.



LINEAR REGRESSION ON SECONDARY DATASET

- We used Linear regression to use **Stress Level (sl)** to predict **Eye Movement (rem)** and **Sleeping Hours (sr.1)**

Predicting Eye Movement (rem)

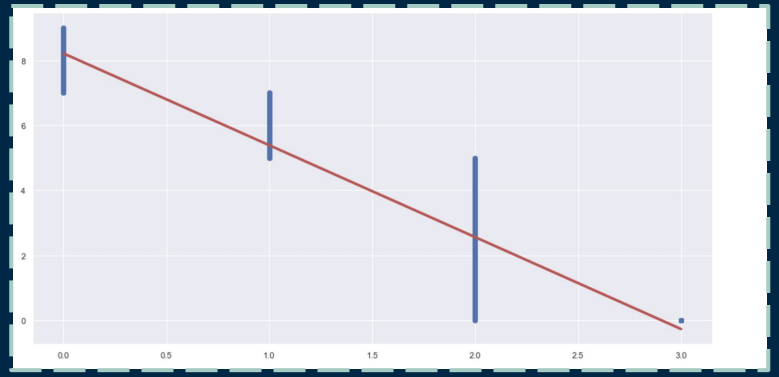



```
----- Predicting for "rem" below using Stress Level "sl": -----
Intercept of Regression          : b =  [71.22890778]
Coefficients of Regression       : a =  [[10.84877231]]

Goodness of Fit of Model         Train Dataset
Explained Variance (R^2)         : 0.8833926860765319
Mean Squared Error (MSE)         : 15.99938196851222

Goodness of Fit of Model         Test Dataset
Explained Variance (R^2)         : 0.8888131777071462
Mean Squared Error (MSE)         : 17.48203325760369
```

Predicting Sleeping Hours (sr.1)



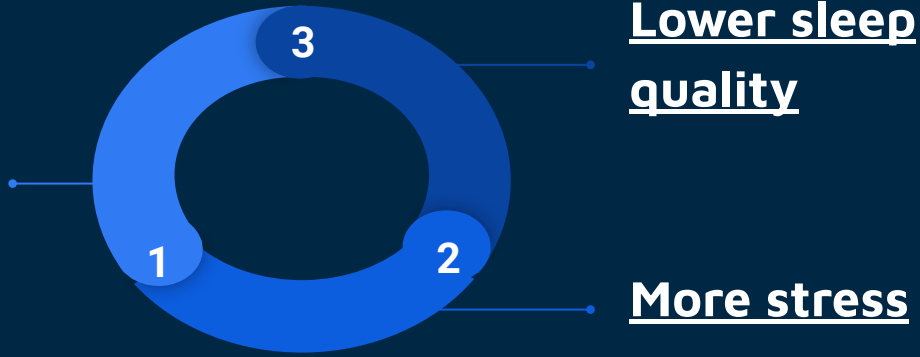

```
----- Predicting for "sr.1" below using Stress Level "sl": -----
Intercept of Regression          : b =  [8.21593749]
Coefficients of Regression       : a =  [[-2.82863585]]

Goodness of Fit of Model         Train Dataset
Explained Variance (R^2)         : 0.8835417416237671
Mean Squared Error (MSE)         : 1.0860937836795233

Goodness of Fit of Model         Test Dataset
Explained Variance (R^2)         : 0.8658333758165624
Mean Squared Error (MSE)         : 1.2440861184704632
```

OUTCOME

- **Two** models:
  - Predict stress levels using emotions
  - Predict sleep quality using stress levels
- Able to predict sleep quality using emotions, with stress levels as the common data column to link between the two datasets
- Discovered that people with higher level of emotions tend to experience a lower quality of sleep

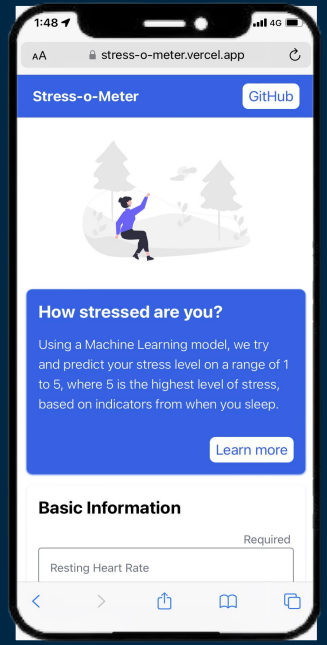


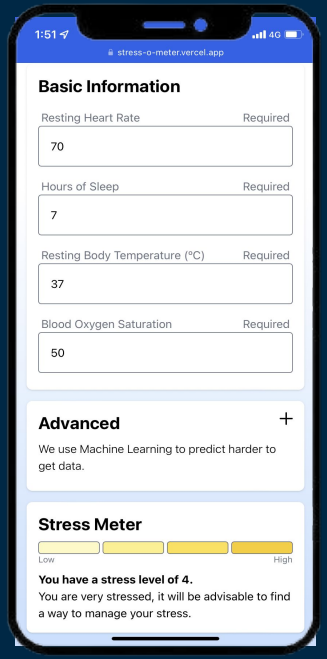
OUTCOME

- Both emotions and sleep quality affects one's stress levels
- One can look into ways to **manage their emotions** throughout the day to lower their stress levels to achieve a higher quality of sleep (e.g meditating)
- One can also **improve their sleeping quality** by changing the settings of one's bedroom before sleeping (e.g by sleeping in a cooler room, the comfortable level of bed and pillow suitable for themselves, regular and adequate sleeping hours) for a **lower stress level**

PHONE APP

- For users to predict their stress levels
- With this app, we strive to make people aware of their own actual stress levels so that they would take breaks more often and sleep better





thank you!