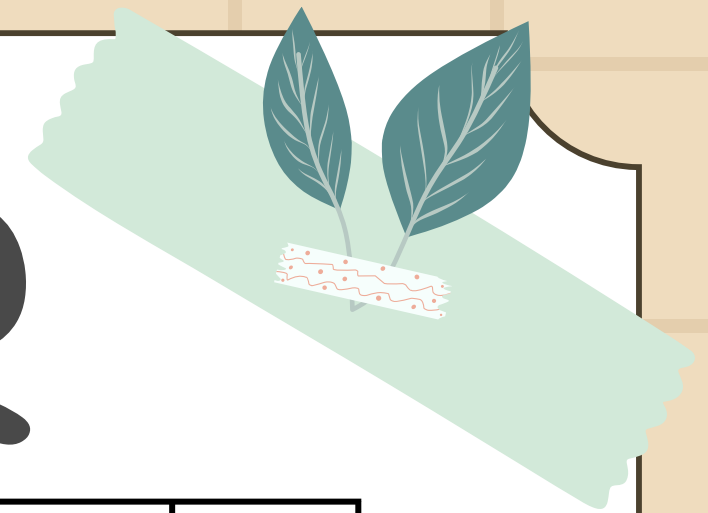
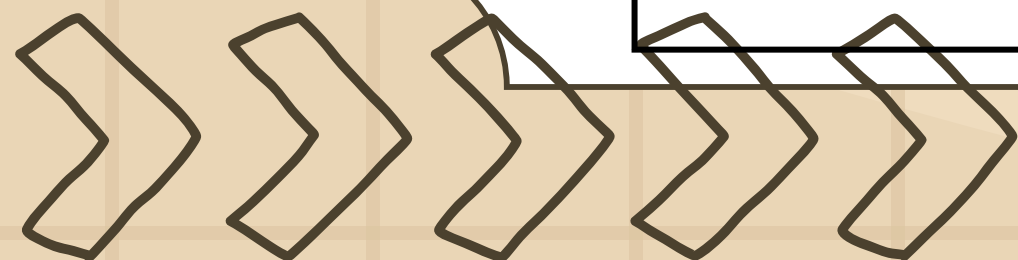


Presented by

STRESS SAVIOR



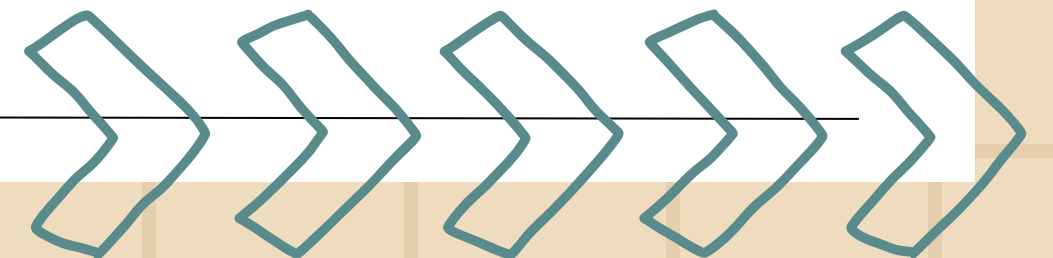
Celine Ong Zi Ying	22004878	1
Dominic Chong Rong Yau	22004881	1
Mooi Kai Jun	22004744	1
Lye Xin Tian	22004853	4
Kew Wei Ming	22004821	1



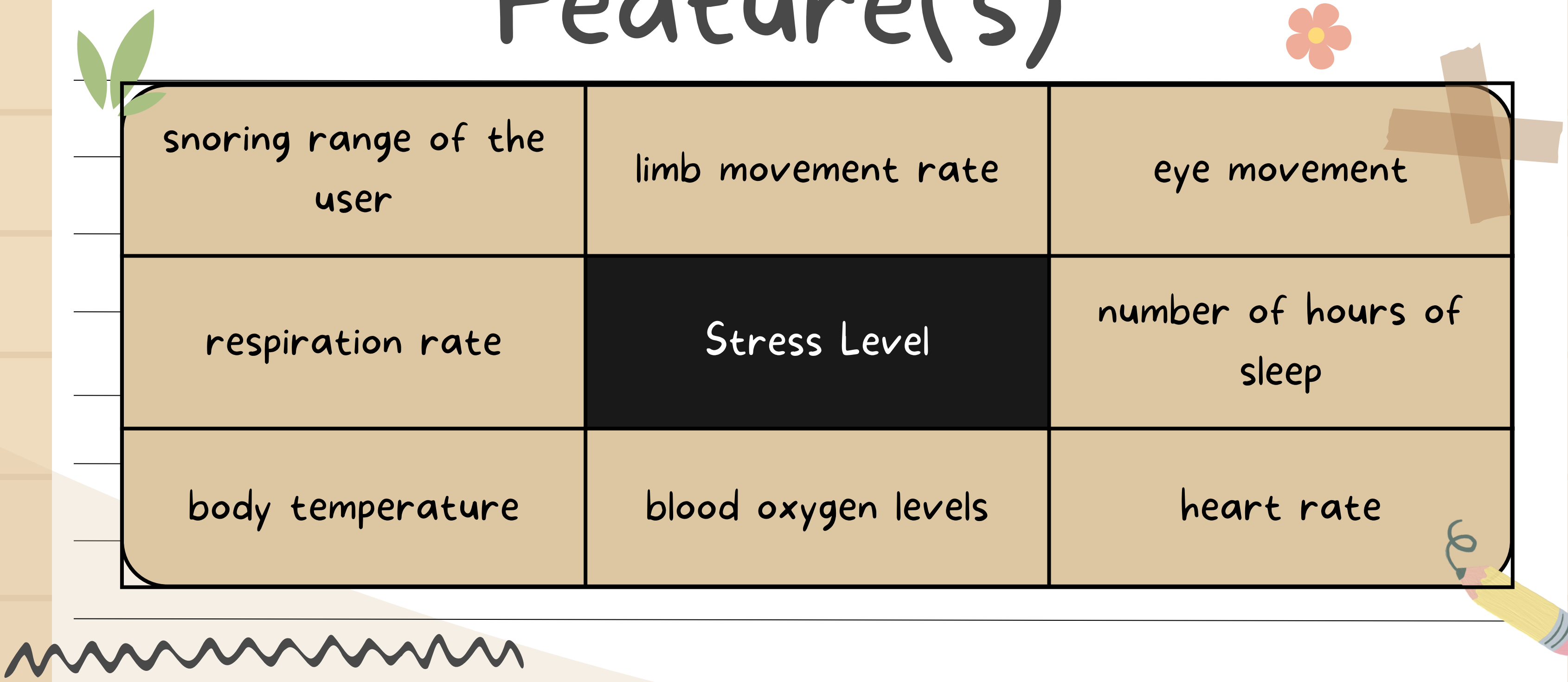


Project Overview

Nowadays, many people are busy on their works until they don't know whether they are stress or not. Especially university students, they are busy with thier assignments and projects until burning midnight oil or even skipping meal. All these factors can be used to determine the stress level of a person.



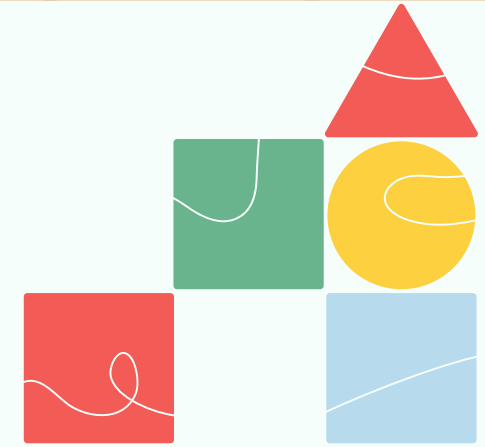
Feature(s)



snoring range of the user	limb movement rate	eye movement
respiration rate	Stress Level	number of hours of sleep
body temperature	blood oxygen levels	heart rate



Naive Bayes



01

Create Naive Bayes model.

02

Fit the model to training data.

03

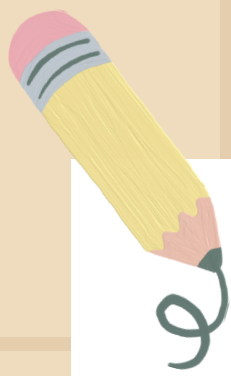
Evaluate performance of model.

04

Perform cross-validation to check if model is overfitting.

05

Visualize model using confusion matrix.



K-Nearest Neighbors



01

Create K-Nearest
Neighbors (KNN) model

02

Best Hyperparameter
Tuning

03

Fit the model into the
training data

04

Model Evaluation

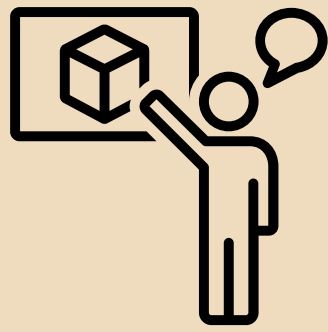
- on both validation and test set
- test set evaluation
- error calculation
- Cross-Validation

05

Visualization of model

- Classification Report
- Confusion Matrix
- Learning Curve

Demo



Classiication Report

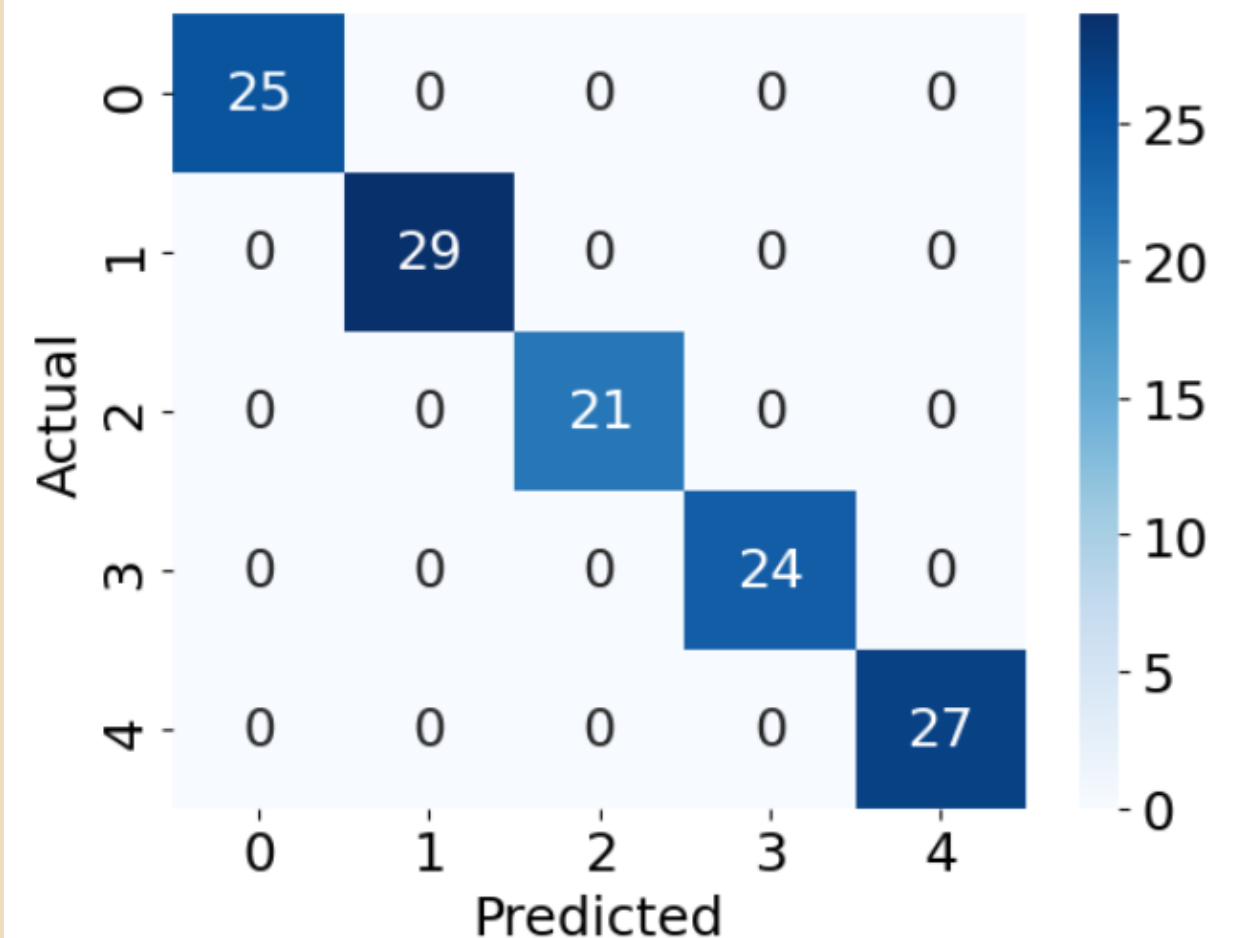
```
# Generate a classification report
report = classification_report(y_test, predict_test)#predict_knn
print("Classification Report:\n", report)
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	25
1	1.00	1.00	1.00	29
2	1.00	1.00	1.00	21
3	1.00	1.00	1.00	24
4	1.00	1.00	1.00	27
accuracy			1.00	126
macro avg	1.00	1.00	1.00	126
weighted avg	1.00	1.00	1.00	126

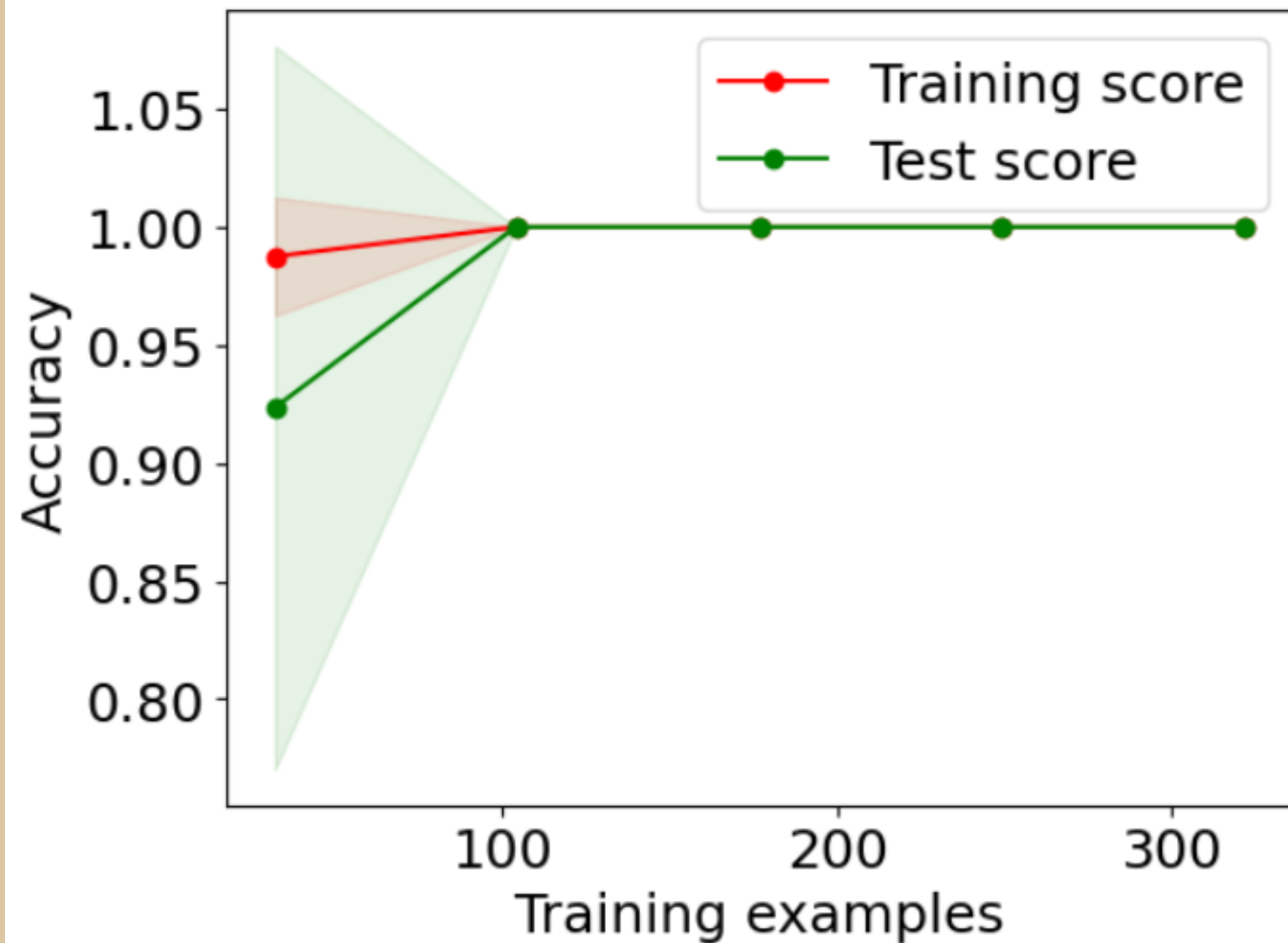
Confusion Matrix

```
# Generate a confusion matrix
confusion_matrix = confusion_matrix(y_test, predict_test) #predict_knn
cm_display = sns.heatmap(confusion_matrix, annot=True, cmap='Blues', fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



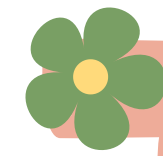
Demo

Learning Curve

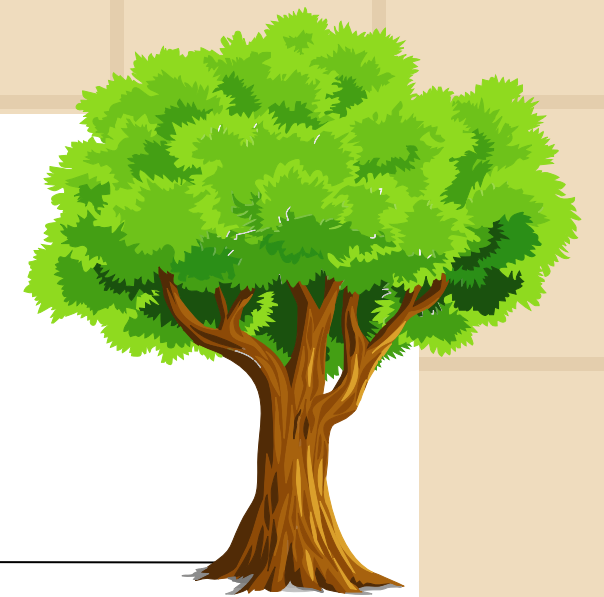


Through this learning curve :

- can ensure that the model is overfitting or not
- similar performance between the training and test scores
- indicate a **well-performing model with good generalization and is not overfitting or underfitting.**



Decision Trees



01

Create a decision tree model to train data.

02

Observe the performance of test data (using metrics)

- Confusion matrix
- Accuracy score
- Classification report

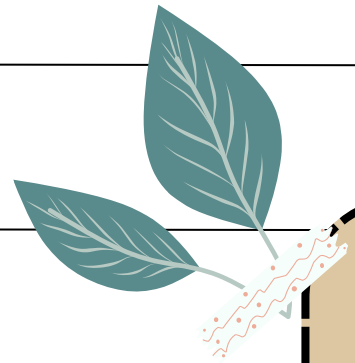
03

Determine the feature importance and plot the tree graph

04

Check if the decision tree model is overfitting

Hyperparameter tuning for Decision Tree

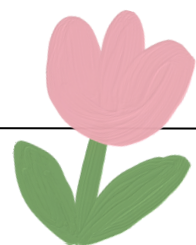


Using Grid Search with Cross-Validation

Pre-Pruning - a technique that refers to the early stopping of the growth of the decision tree.

1. Maximum depth
2. Minimum sample split
3. Minimum sample leaf
4. Maximum features
5. Criterion





THANK YOU

