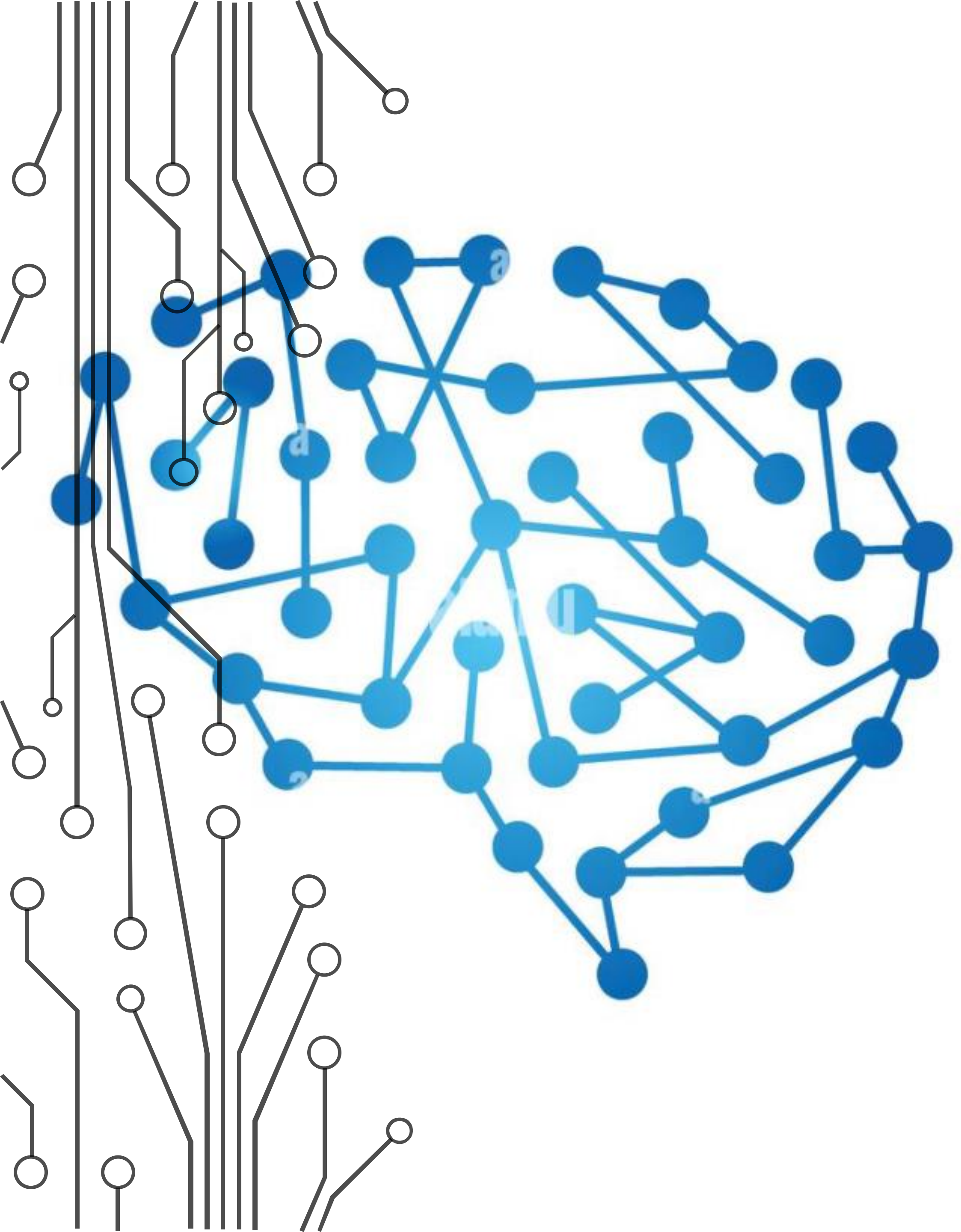
C379 EMERGING TECHNOLOGIES

LESSON 16: DATA PREPARATION

OFFICIAL (CLOSED) \ NON-SENSITIVE

L16 LEARNING OBJECTIVES

- **Explain the use of predictive analytics to predict trends by solving classification problems using data**
- **Explain and differentiate between train, validation, and test datasets for predictive analytics**
- **Prepare train, validation and test datasets for predictive analytics from data for a given scenario**
- List and describe three (3) different machine learning algorithms used to perform classification tasks in AI
- Apply these three (3) machine learning algorithms to train predictive models
- Test and evaluate the three (3) predictive models and identify the one that will produce the best solution for the given scenario
- Explain and effectively present results to end-user devices

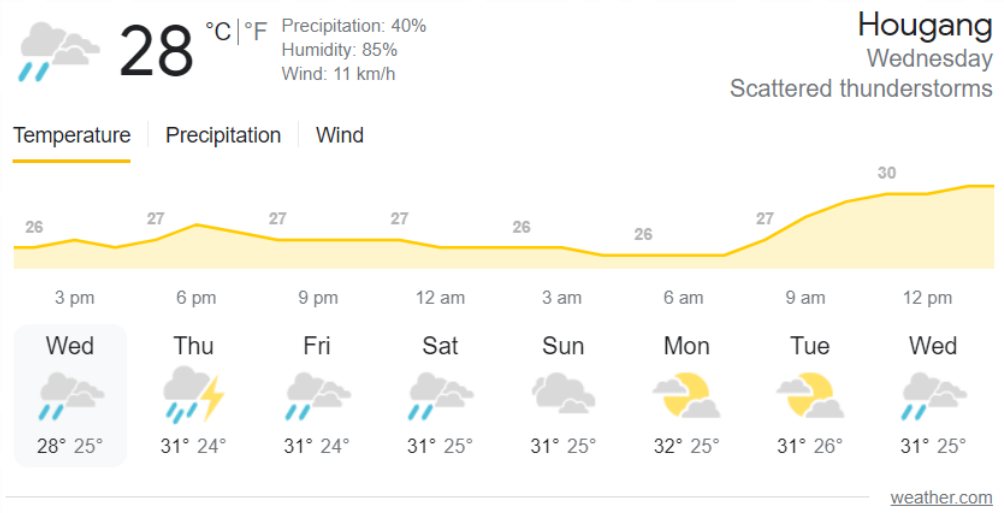


PREDICTIVE ANALYTICS

WEATHER PREDICTION

Take a moment to discuss some of these questions?

1. Where do you get your weather forecast?
2. How does the app predict tomorrow's weather?
3. What will be the prediction of the weather on this year's National Day?
4. Why the forecast is inaccurate most of the time?



2.00 pm to 4.00 pm
Updated at 2.00pm 24 Nov

PREDICTIVE ANALYTICS

TECH(talk)

What is Predictive Analytics?

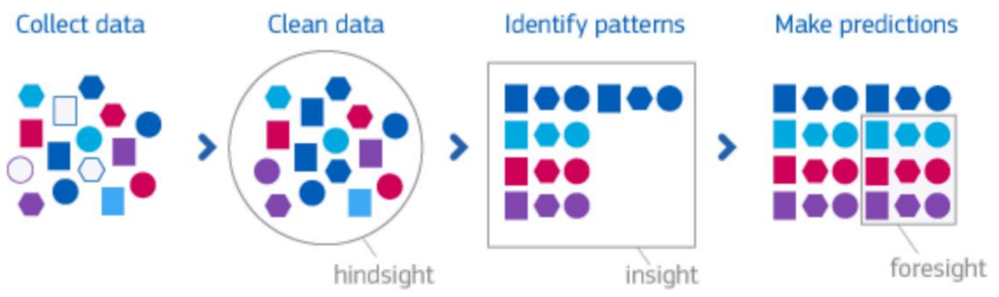
Transforming data into future insights

0:09 / 3:20

https://www.youtube.com/watch?v=cVibCHRSxB0

PREDICTIVE ANALYTICS

- Predictive analytics is the use of data, statistical algorithms and machine learning techniques to identify the likelihood of future outcomes based on historical data.
- The goal is to go beyond knowing what has happened to providing a best assessment of what will happen in the future.
- In this lesson, we will build AI models that predict categories. This class of machine learning problems is known as **classification** tasks.



OFFICIAL (CLOSED) \ NON-SENSITIVE

CLASSIFICATION TASKS

- Classification tasks belongs to the *supervised learning* branch of ML. These kinds of tasks are the most widely used in applications in industry.
- ML classification models can output 2 types of predictions:
  - **Predicted classes:** For every observation, the model will directly give the prediction of the class.
  - **Probabilities of each class:** For every observation and every class, the model will output probabilities of that observation belonging to the class.

OFFICIAL (CLOSED) \ NON-SENSITIVE

PREDICTING CATEGORIES USING CLASSIFICATION MODELS

- We want to use the variables (Chest Pain, Good Blood Circulation, etc) to predict if someone has heart disease.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	45	No
Yes	Yes	Yes	80	Yes
Yes	Yes	No	90	No
…	…	…	…	…

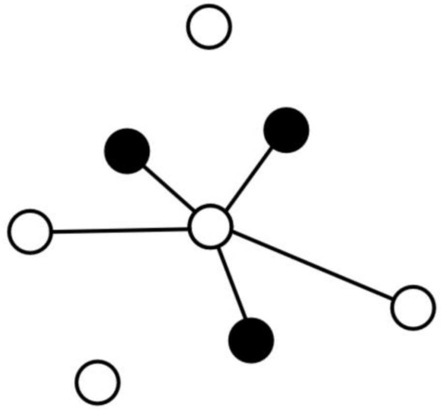
- When a new patient shows up, we can measure these variables to predict if he/she has heart disease or not.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	56	???

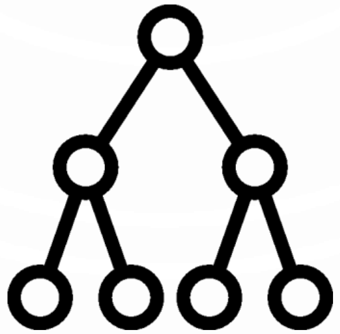
OFFICIAL (CLOSED) \ NON-SENSITIVE

PREDICTING CATEGORIES USING CLASSIFICATION MODELS

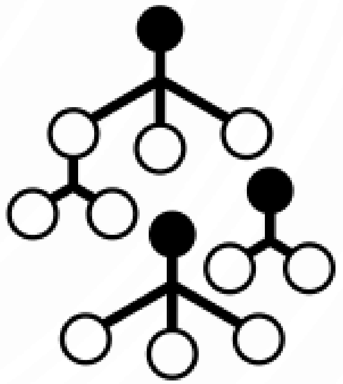
- However, first we have to decide which machine learning method would be best …



K-nearest neighbours…



or Decision Tree…



or Random Forest?

- And many more machine learning methods…

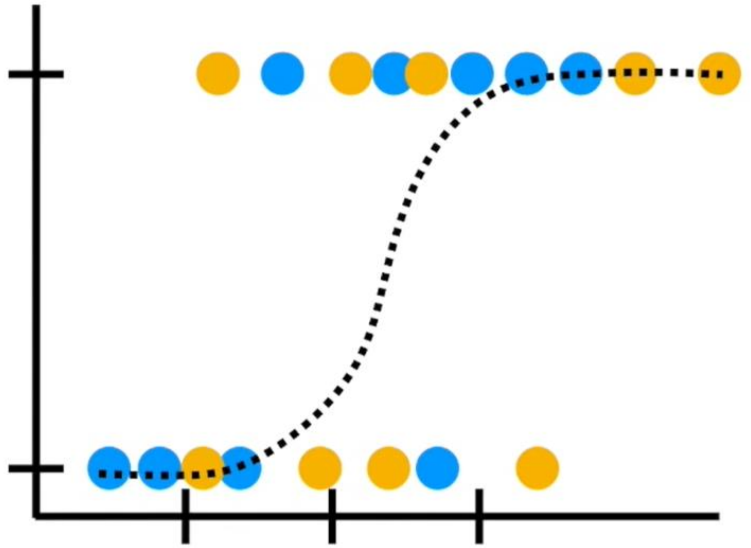
OFFICIAL (CLOSED) \ NON-SENSITIVE

PREDICTING CATEGORIES USING CLASSIFICATION MODELS

- From the data that we have collected about people with and without heart disease.
- From this data, we need to **estimate the parameters for the machine learning methods.**
- In other words, to use K-nearest Neighbours (KNN) method, we have to use some of the data to estimate the distance between each point.
- In machine learning lingo, estimating parameters is called "**training** the algorithm".

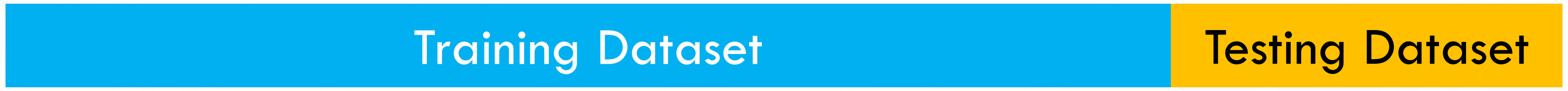
DATA PREPARATION

- Using machine learning, we need the data to….
  1. **Train** the machine learning methods
  2. **Test** the machine learning methods.
- A terrible approach would be to use all of the data to estimate the parameters (i.e. train the algorithm), because then there would not be any data left to test the method.
- **Reusing the same data** for both training and testing is a **bad** idea because we need to know how the method will work on data it was not trained on. Otherwise, it will lead to a problem known as *overfitting*.

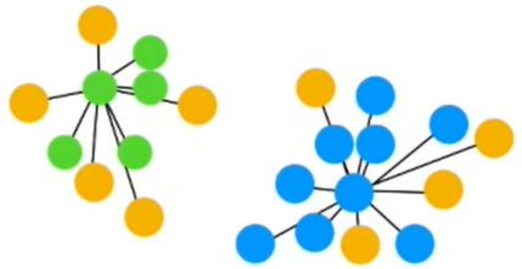
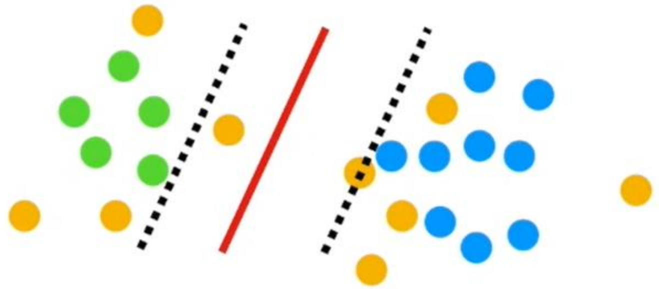


DATA PREPARATION

- A better approach would be to use the first 70%-80% of the data for **training,**
- and the last 20%-30% of the data for **testing.**



- We could then compare methods by seeing how well one categorised the test data.

OFFICIAL (CLOSED) \ NON-SENSITIVE

DIFFERENCE BETWEEN TRAINING, VALIDATION & TEST DATA

- **Training data**
  - Large portion of the dataset in a corpus should be reserved for training your predictive models
  - By default, **60% of your dataset is recommended to be set aside as the training set**
- **Validation data**
  - The validation set is a separate section of your dataset that you will use during training
  - The purpose of validation data is to get a sense of how well your model is doing on data that are not being used in training
  - **20% of your dataset is recommended to set aside as the validation set**
- **Test data**
  - To prevent overfitting the new model, evaluation metrics can be run on the test set at the end to get a sense of how well your predictive models will perform in production
  - **20% of your dataset is recommended to set aside as test set**

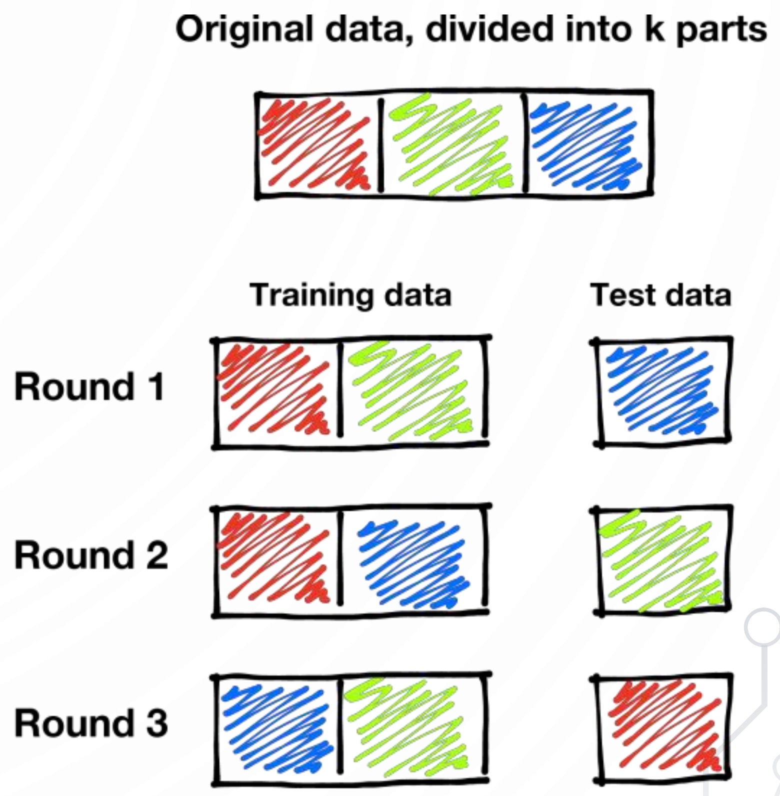
OFFICIAL (CLOSED) \ NON-SENSITIVE

COMMON PITFALLS IN THE TRAIN, VALIDATE & TEST SPLIT

- **Train/Test Bleed**
  - This is when some of your testing data are **overly similar** to your training data
  - When duplicating data in the dataset, ensure that these data do not enter different train, validation and test splits
- **Overemphasis on the Training Set**
  - The more data, the better the model. However, this mantra might tempt you to use most of your dataset for the training set and only to hold out 10% or less for validation and test.
  - Skimping on your validation and test sets could cloud your evaluation metrics with a limited subsample, and lead you to select a suboptimal model
- **Overemphasis on Validation and Test Set Metrics**
  - Validation and test set metrics are only as good as the data underlying them
  - They may not be fully representative of how well you model will perform in production

CHALLENGE OF USING VALIDATION DATA

- One unfortunate effect of validation set is that we can now use only 60% of the original data to train the predictive models.
- This can be an issue if we have a very limited amount of training data to begin with.
- Nevertheless, this problem can be addressed by using a technique known as ***cross-validation***.
- Cross-validation avoids holding out parts of the dataset to be used as validation data, but at the expense of additional computation.

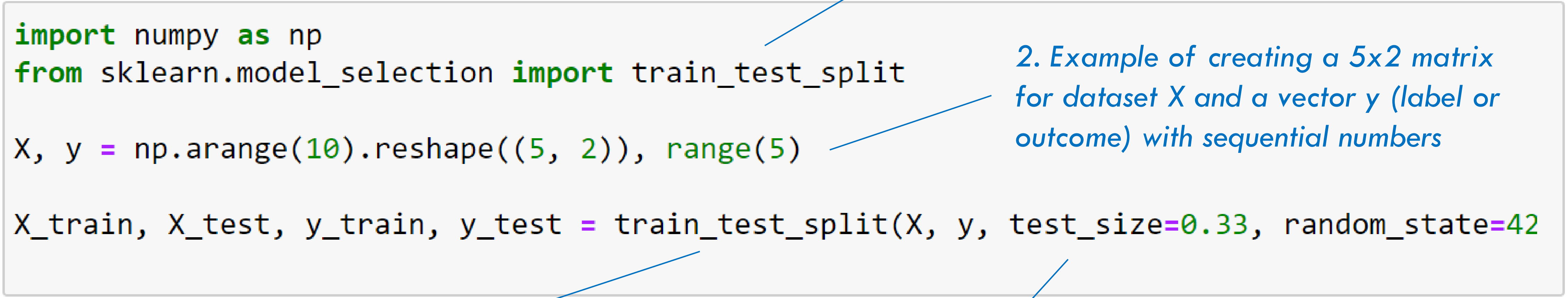


OFFICIAL (CLOSED) \ NON-SENSITIVE

PREPARE TRAINING AND TESTING DATA

1. Importing train_test_split method from scikit-learn's model_selection library

2. Example of creating a 5x2 matrix for dataset X and a vector y (label or outcome) with sequential numbers



```
import numpy as np
from sklearn.model_selection import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42
```

3. This method split the dataset into 4 different segments:
***X_train** – Data use for **training** the predictive model*
***X_test** – Data use for **testing** the model*
***y_train** – Labels corresponding to X_train*
***y_test** – Labels corresponding to X_test*

4. The parameters used for the method can be included:
***test_size** – proportion of dataset allocating for testing*
***random_state** – seed number used to shuffle the data*

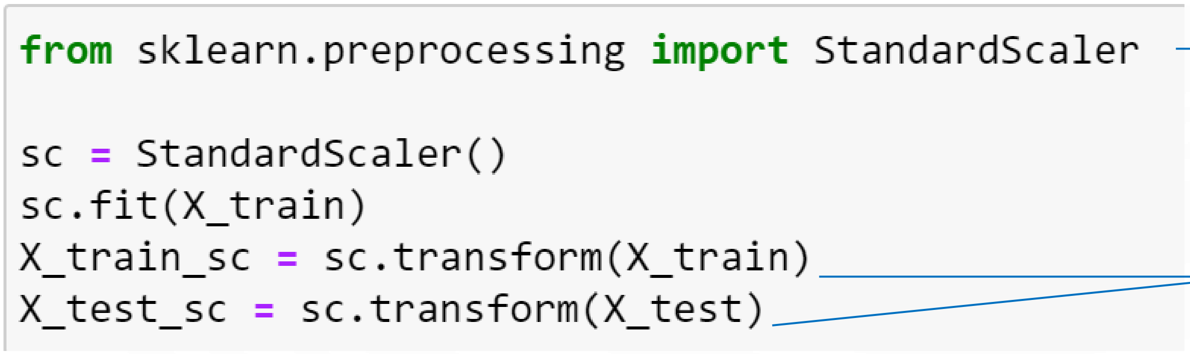
OFFICIAL (CLOSED) \ NON-SENSITIVE

APPLYING DATA SCALING

- Data Scaling is a recommended pre-processing step when working with many ML algorithms. It performs better when numerical input variables are scaled to a standard range
- One of the popular technique for scaling numerical data is **standardization.** It scales each input variable separately by subtracting the mean and dividing by the standard deviation:

$$y = \frac{x \ - mean}{stddev}$$

- The goal is to shift the distribution to have a mean of 0 and a standard deviation of 1.



```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
sc.fit(X_train)
X_train_sc = sc.transform(X_train)
X_test_sc = sc.transform(X_test)
```

1. Importing StandardScaler method from scikit-learn's preprocessing library

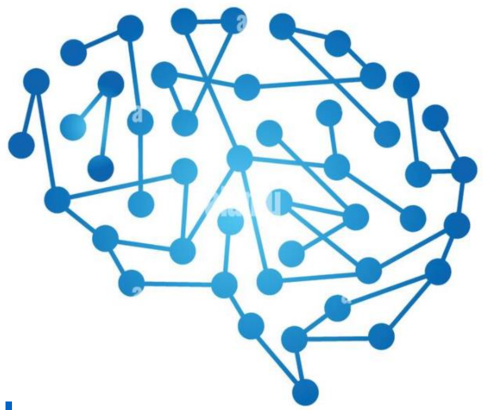
2. Apply standard scaling to the train & test dataset

PRACTICAL

LAB 16-1

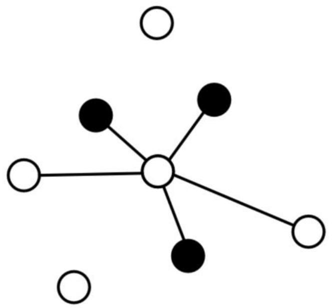
DATA PREPARATION

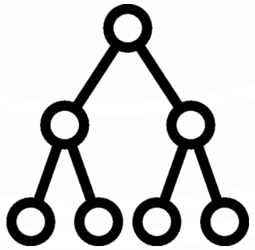
UNSCHEDULED ASYNCHRONOUS E-LEARNING

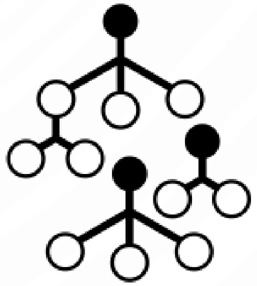


Watch the following videos before Lessons17 :

- KNN – https://www.youtube.com/watch?v=HVXime0nQeI







60 mins