Supervised Machine Learning

Classification, Regression, Time Series

Recap – Machine Learning Framework

$$y = f(x)$$

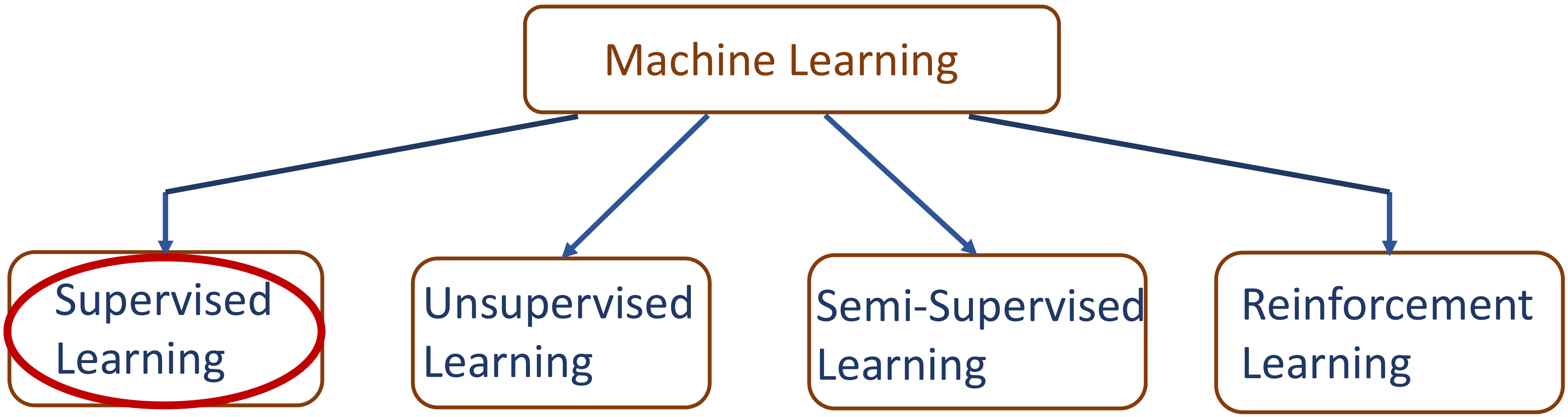
output

prediction function

feature or representation

- The input is converted to a vector **x**
- The output is a value indicated by **y**
- Depending on the nature of **x** and **y**, we define different types of learning

Categorization of ML Based on Learning



Supervised Learning

- Machine learning that are designed to learn by examples

- It is trained with labelled data
  - Feature vectors: $\boldsymbol{x_{ij}},\ \boldsymbol{i = 1..N},\ \ \boldsymbol{j = 1..M}$
  - Output values: $\boldsymbol{y_i},\ \boldsymbol{i = 1..N}$

- It maps the input to an output based on previous input-output pairs, through a mapping function, $\boldsymbol{Y = f(X)}$

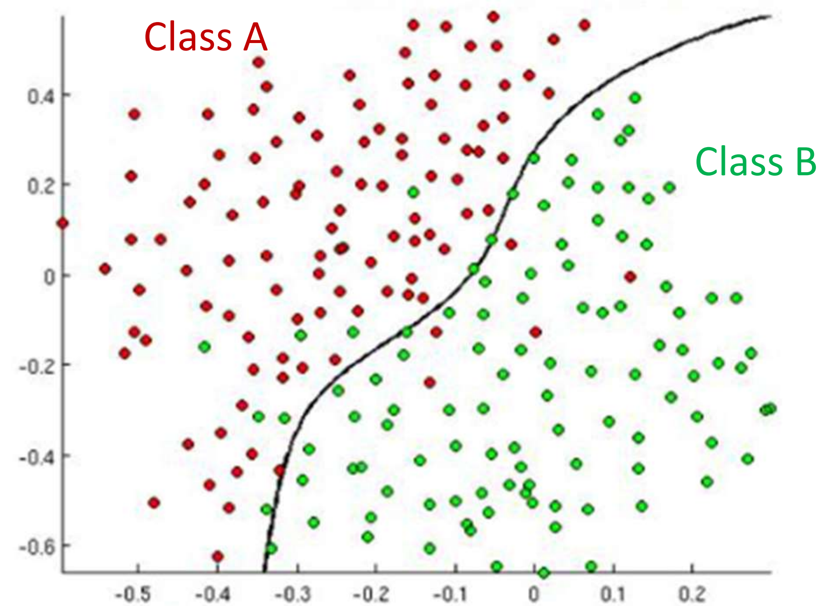
- Depending on the nature of *y*, we define:
  1. Classification
  2. Regression

Classification

- Classification predicts a discrete value/class label

- Some common classification algorithms include:
  - Decision Trees
  - Support Vector Machines
  - Naïve Bayes Classifier

- Classification is often used for tasks such as:
  - Spam filtering
  - Image classification
  - Text classification



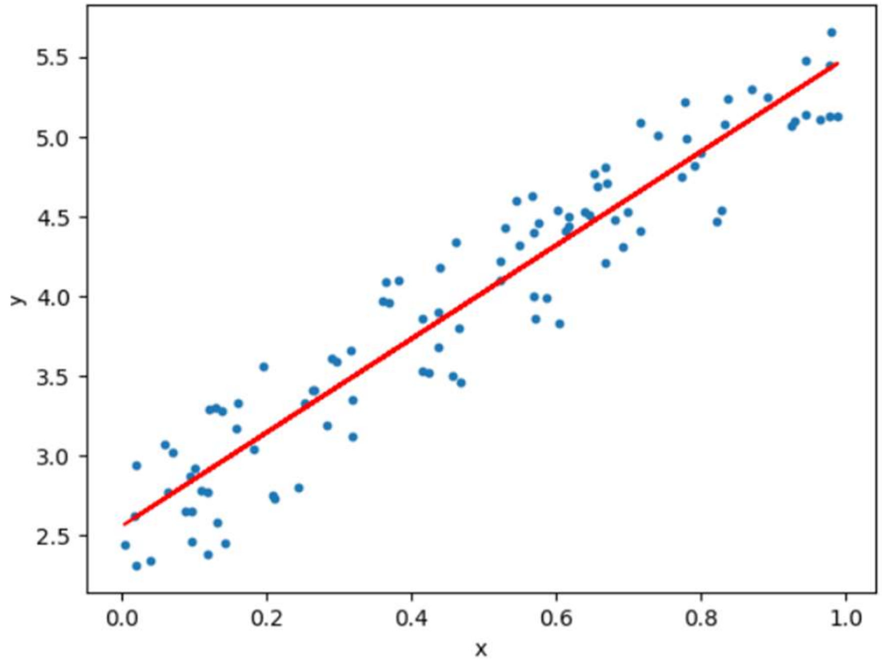
Regression

- Regression is a type of machine learning that predicts a continuous value.

- Some common regression algorithms include:
  - Linear regression
  - Polynomial regression

- Regression is often used for tasks such as:
  - Predicting the price of a house
  - Predicting the number of sales
  - Predicting the risk of a disease

e.g., a house's [Area, Age] (**x**) vs. its Price(*y*)



y = f(x), interpolating (approximating) a function from examples

Time Series Model

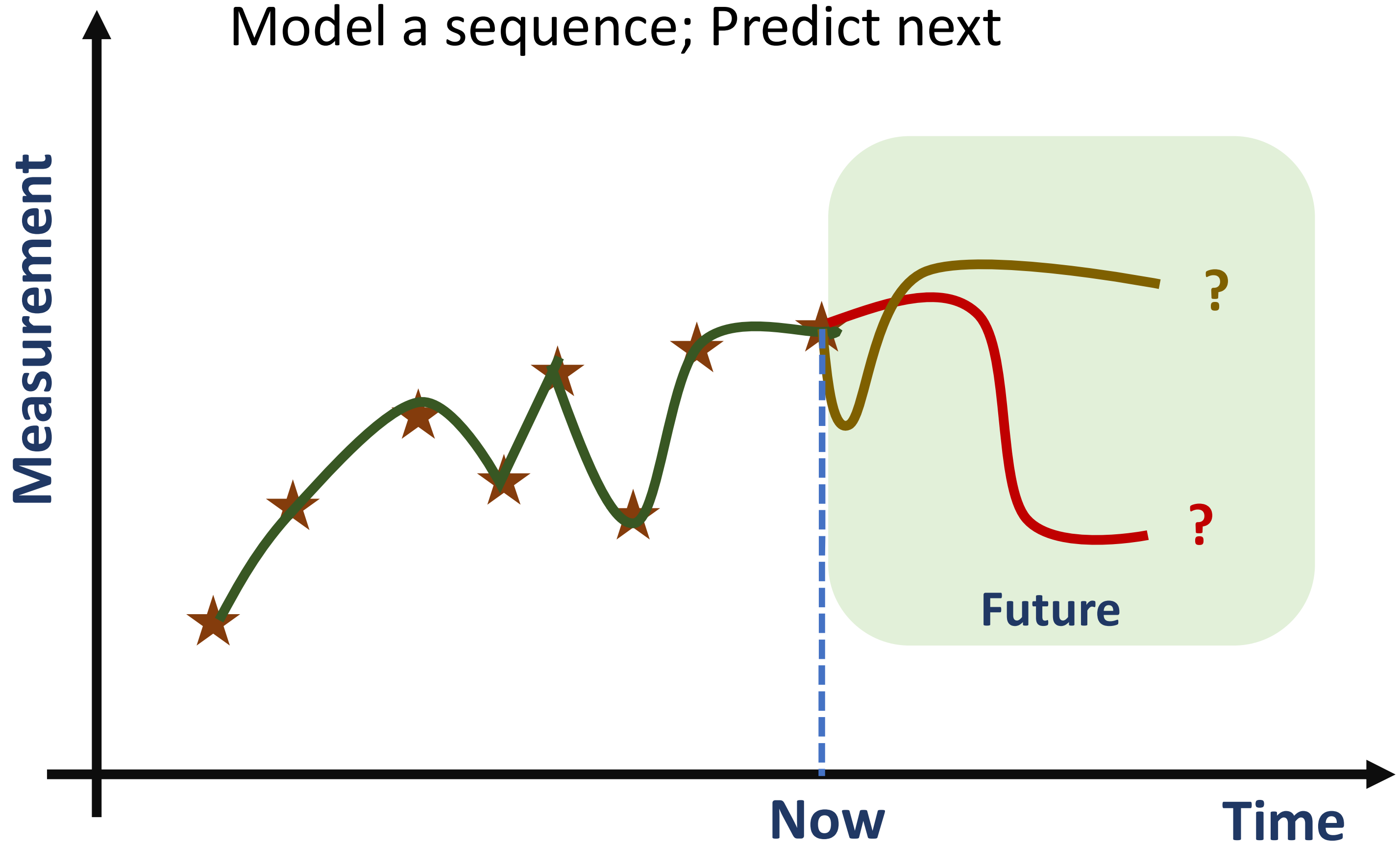
- Time series is a sequence of observations often ordered in time
- Popular Problem: Given a sequence, predict future samples
- Applications:
  - Meteorology,
  - Finance,
  - Marketing, etc

Year	Sales (in Million)
1921	251
1931	279
1941	319
1951	261
1961	439
1971	348
1981	585

- We want a machine learning model to understand sequences, not samples
- Assume we have a sequence of measurements, and we want to take N sequential measurements and predict the next one

Time Series Prediction

Model a sequence; Predict next



Nearest Neighbour Classifier

Classification with Association by Similarity

Nearest Neighbour Classifier

X_{test}: [42.5, 39]

→ ***Is the person diabetic or not?***

- How do we compare a test sample to a classification?
- We find distance to feature vectors of known classes - Association by Similarity
- We assign label of that sample which is nearest to the test sample

BMI	Age	Diabetic
32.6	49	1
34.2	23	1
22.4	31	0
25.7	43	0
29.8	15	0
31	58	1
43.2	65	0
37.6	52	1

Nearest Neighbour Classifier

BMI	Age	Diabetic
32.6	49	1
34.2	23	1
22.4	31	0
25.7	43	0
29.8	15	0
31	58	1
43.2	65	0
37.6	52	1

Feature Vector	Label
X_1 $[32.6, 49]$	1
X_2 $[34.2, 23]$	1
X_3 $[22.4, 31]$	0
X_4 $[25.7, 43]$	0
X_5 $[29.8, 15]$	0
X_6 $[31.0, 58]$	1
X_7 $[43.2, 65]$	0
X_8 $[37.6, 52]$	1

Nearest Neighbour Classifier

X_{test}: [42.5, 39]

Feature Vector	Label	Distance
X_1 $[32.6, 49]$	1	**14.07**
X_2 $[34.2, 23]$	1	**18.02**
X_3 $[22.4, 31]$	0	**21.63**
X_4 $[25.7, 43]$	0	**17.27**
X_5 $[29.8, 15]$	0	**27.15**
X_6 $[31.0, 58]$	1	**22.21**
X_7 $[43.2, 65]$	0	**26.01**
X_8 $[37.6, 52]$	$\boxed{1}$	$\boxed{\mathbf{13.89}}$

$$\sqrt{(\mathbf{42.5-32.6})^2+(\mathbf{39-49})^2}$$
$$= \mathbf{14.07}$$

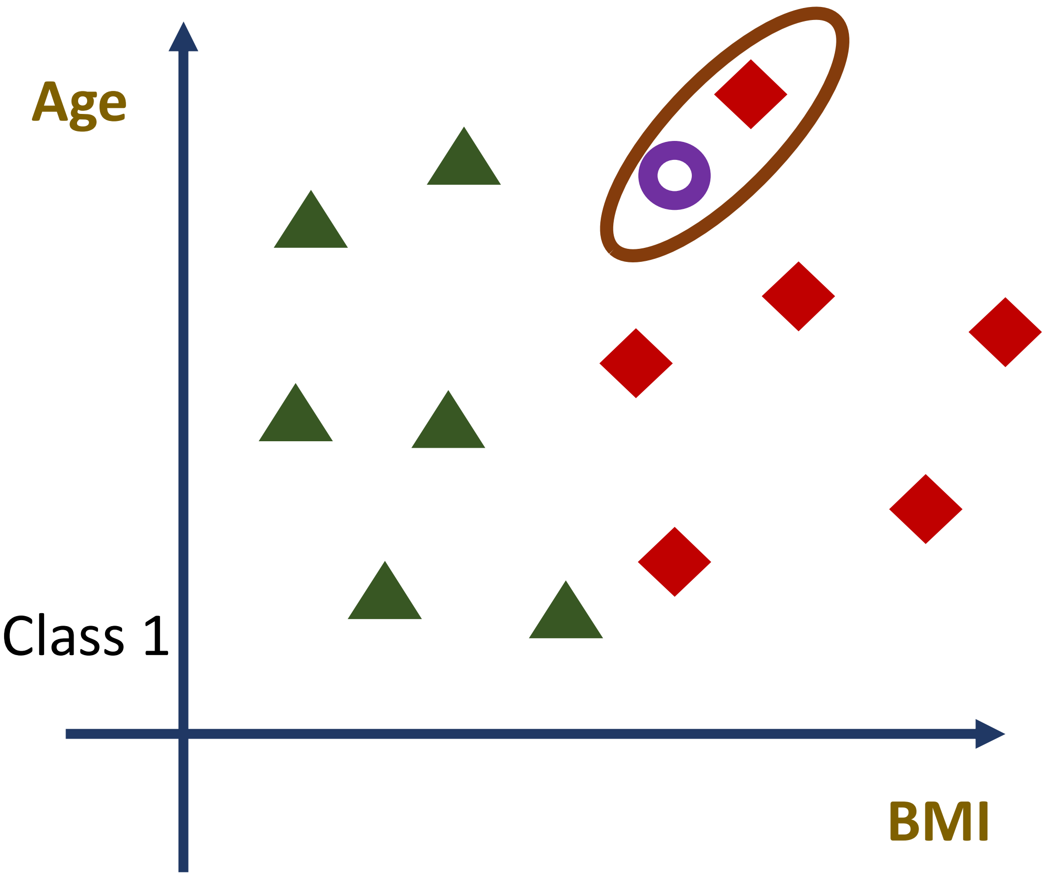
✓ ***The test sample has diabetes***

Visualization in Feature Space

Class 1

Class 0

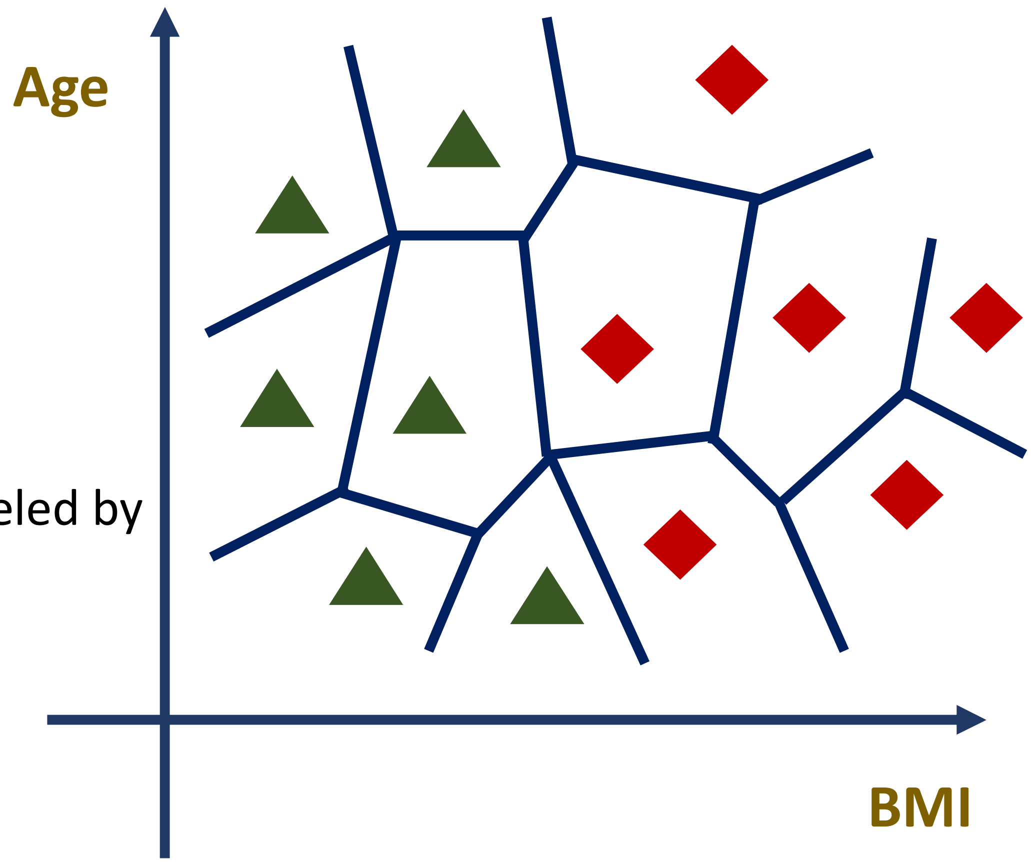
- The nearest sample is a
- We assign the test sample to Class 1



Class Boundaries

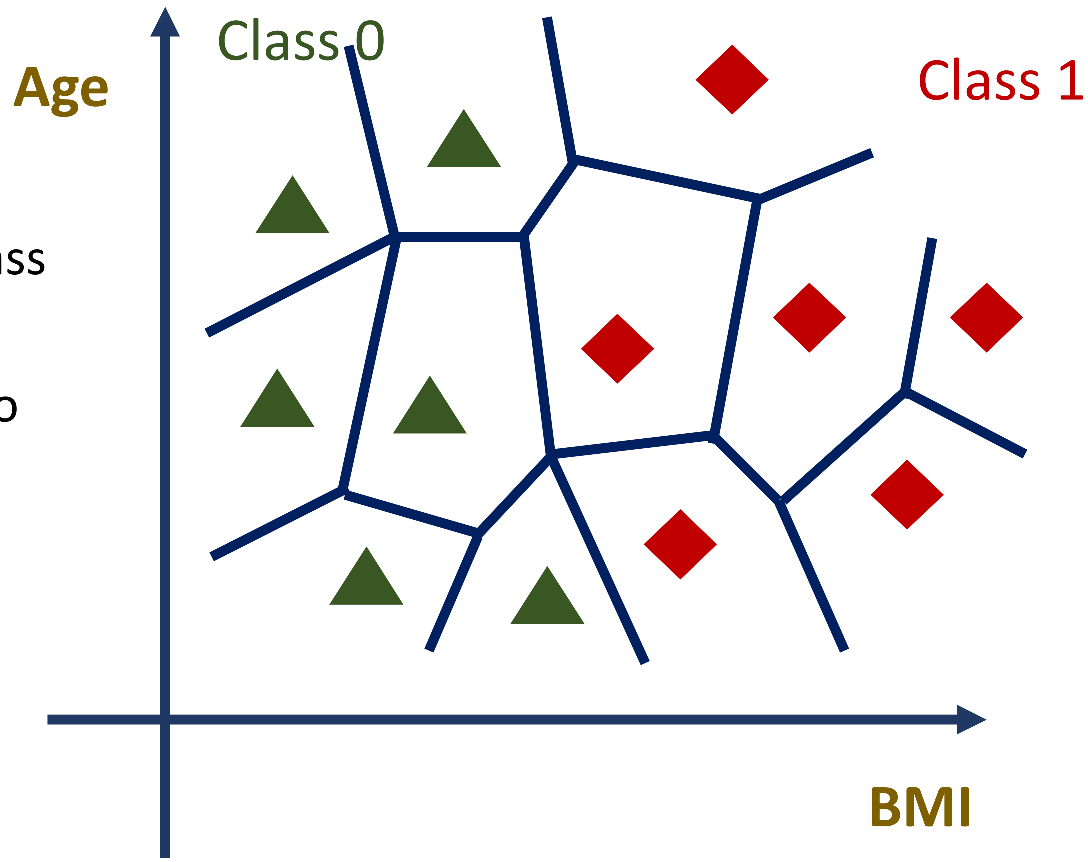
- The classifier effectively partitions the feature space into cells consisting of all points closer to a given training point (x_1,x_2) than to any other training points.

- All points in such a cell are thus labeled by the category of the training point – **Voronoi tessellation** of the space



Class Boundaries – Partition of Feature Space

- We now ignore boundaries between samples of the same class
- The decision boundary is found to be piece-wise linear



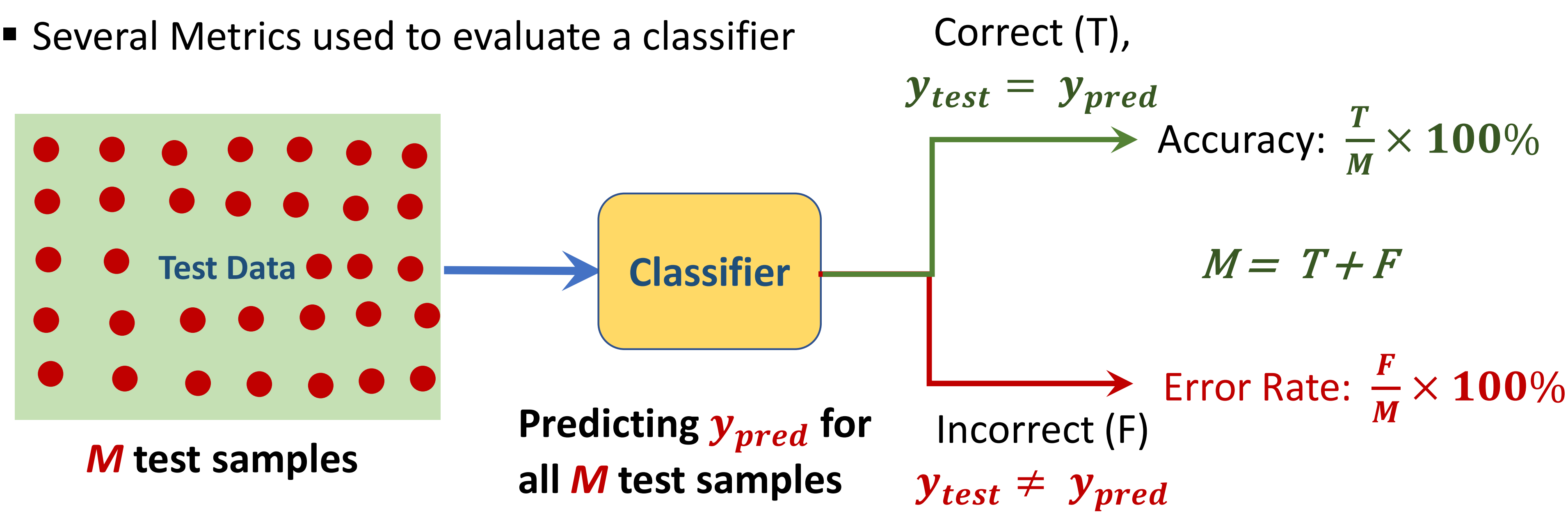
The Classification Algorithm

Problem

- Given:
  - A set of training n training samples: $(\mathrm{x_i}, y_\mathrm{i})$
  - A set of m test samples: $(x_{test}, y_{test}), m \ll n$
- Find:
  - Label (x_{test}) using Similarity Measure and return y_{pred}
- Find accuracy of the prediction
  - Evaluation of a classifier

Evaluating a Classifier

- Several Metrics used to evaluate a classifier



- In a 2-class classifier, what does an accuracy of 65% tell us about the classifier?

Supervised Machine Learning - Methodology

- **Step 1**: A set of training $\boldsymbol{n}$ training samples: $(\boldsymbol{x_i}, \boldsymbol{y_i}), \boldsymbol{i} = \mathbf{1}, .., \boldsymbol{n}$

- **Step 2**: We need to correctly predict labels of unseen $\boldsymbol{m}$ test sample: $(\boldsymbol{x_{test}}, \boldsymbol{y_{test}}),\ \boldsymbol{test} = \mathbf{1}, .., \boldsymbol{m}$. Predicted value = $\boldsymbol{y_{pred}}$

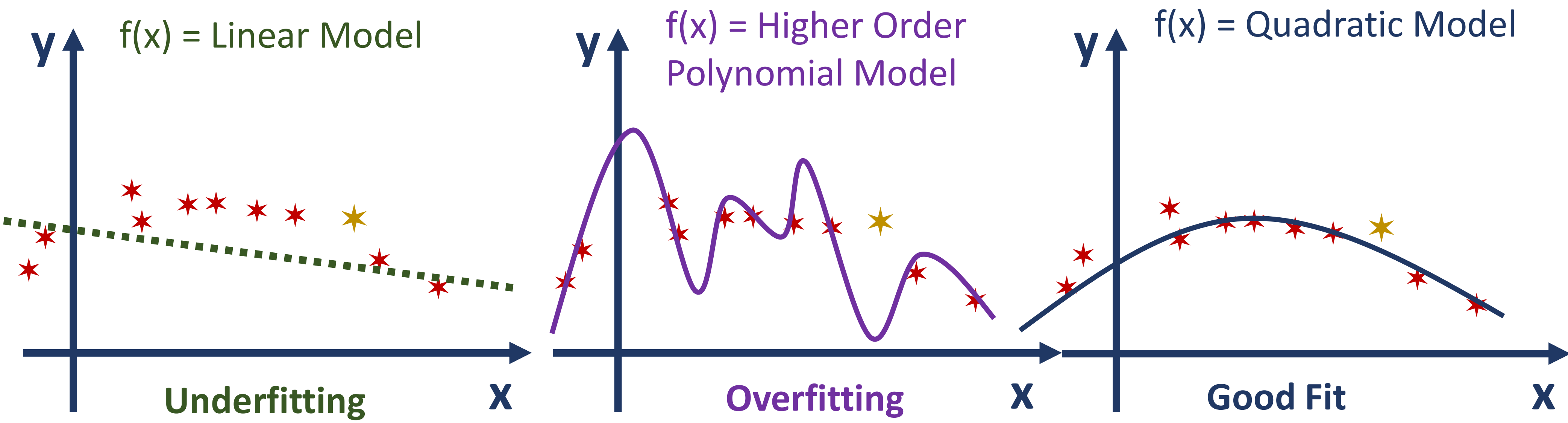
- **Step 3**: We need to maximize the accuracy on unseen $\boldsymbol{m}$ test sample: $(\boldsymbol{x_{test}}, \boldsymbol{y_{test}}),\ \boldsymbol{test} = \mathbf{1}, .., \boldsymbol{m}$. $\boldsymbol{y_{pred}} = \boldsymbol{y_{test}}$

- **Assumption**: Test samples come from the same distribution as training samples

- Can we have situations where we do well on training samples but perform badly on test samples? Rote Learning / Memorization

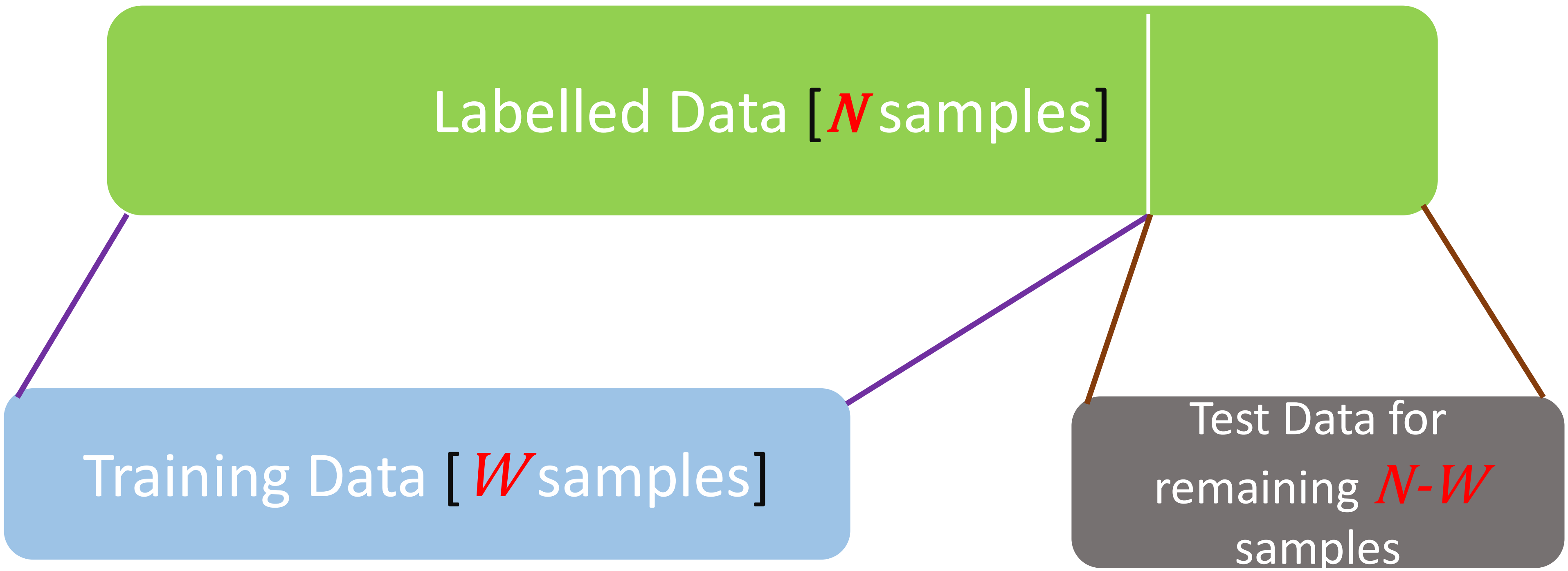
Overfitting vs Generalization

- We try to fit the data with different models in various orders of complexity



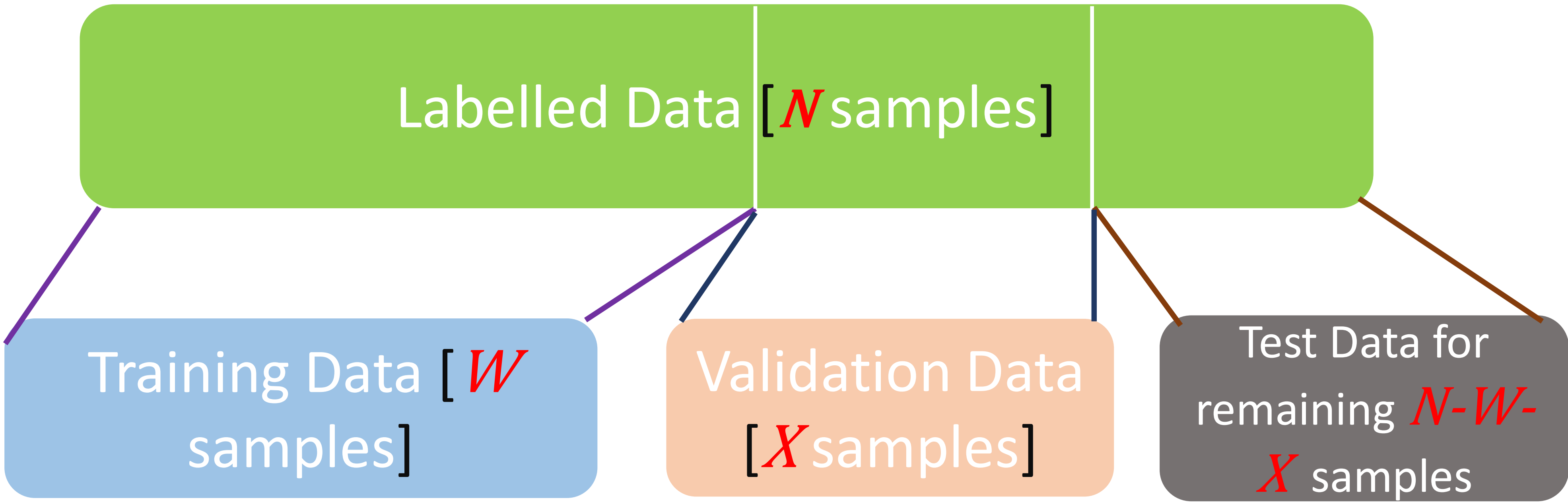
- Which of the above model will perform best on the unseen test data?

ML Based on Training-Testing Data

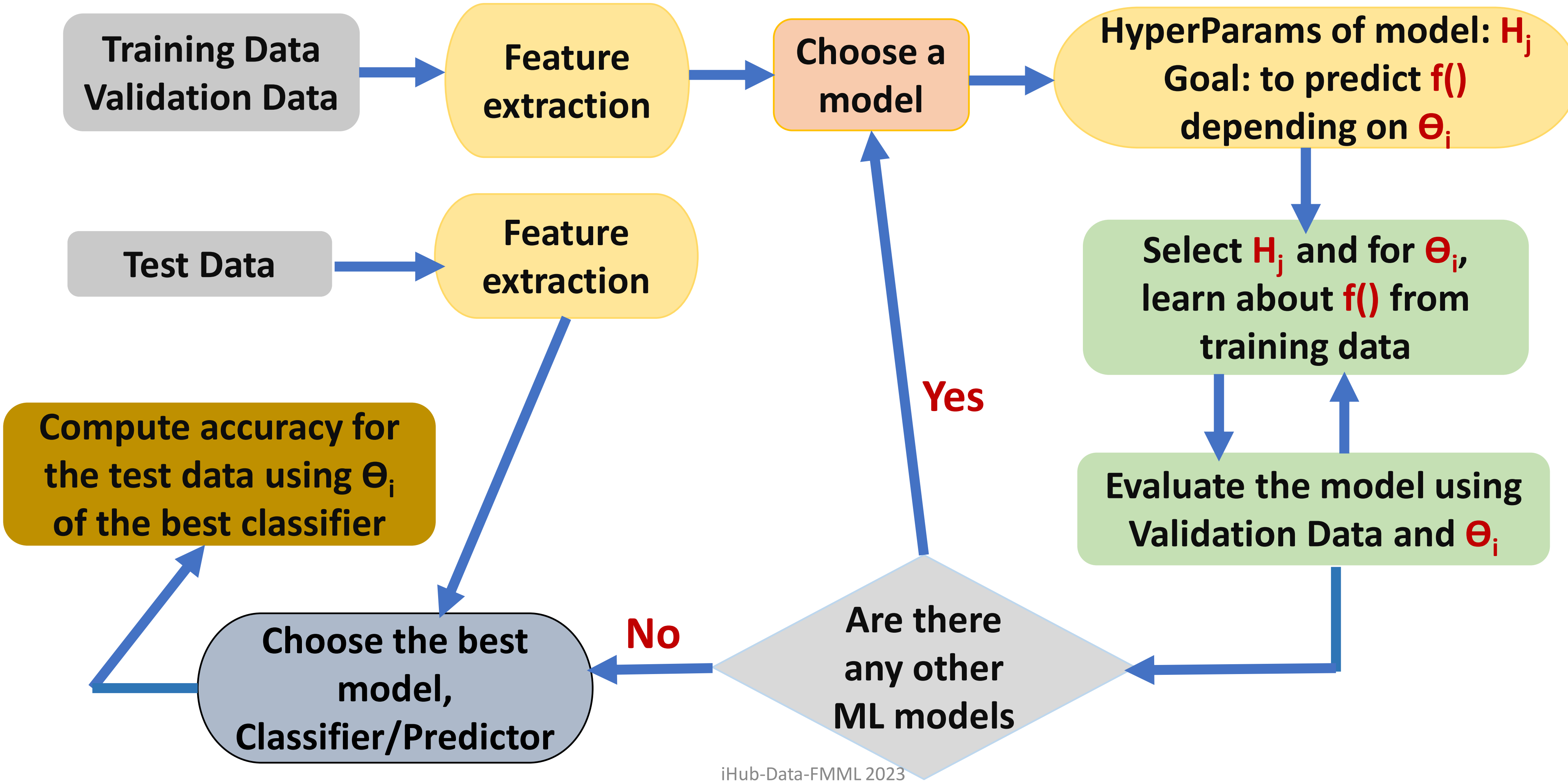


- Take care to not leak information from Test Data into the Model with repeated testing

ML Based on Training - Validation - Testing Data



- The validation model is repeatedly used during development
- The test data is used once for the final prediction



Training Data
Validation Data
Feature extraction
Choose a model
HyperParams of model: H_j
Goal: to predict f()
depending on Θ_i
Test Data
Feature extraction
Select H_j and for Θ_i, learn about f() from training data
Yes
Compute accuracy for the test data using Θ_i of the best classifier
Evaluate the model using Validation Data and Θ_i
No
Are there any other ML models
Choose the best model, Classifier/Predictor
iHub-Data-FMML 2023