Traditional Prog

data →
computer → output
program →

① Intro

Page No.

Date :

Intro to ML

ML
Data →
computer → prog
Output →

- → Machine Learning pipeline
  - → Data Generation / Collection - generate, webscrape, kaggle
  - → Data Cleaning / Feature Engineering
  - → Algorithms to find patterns
  - → Deployment

ml :- machines are able to learn without being explicitly program

Supervised Learning

→ learn from data (labelled)
→ generalised predictions on future data and avoid "OVERFITTING"

TRAINING - 4

x — input

y — label

label → continuous / discrete value

continuous
~~[illegible]~~ regression
- house price prediction
- stock price prediction

discrete value
classification
- is it cat or dog?
  0 or 1
- → is it an apple?
  0 or 1

Representation

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^{i} \; x_2^{i} \; x_3^{i} \ldots x_n^{i} \\ \vdots \\ —— x_j —— \end{bmatrix} \quad y = \begin{bmatrix} \uparrow \\ y_i \\ \downarrow \end{bmatrix}$$

In regression
$f(x) \rightarrow y$
$x \in R^n$
$y \in R$

In classification
$f(x) \rightarrow y$
$x \in R^n$
$y \in$ one of the classes

Traditional Prog

$\text{data} \rightarrow$ computer $\rightarrow$ output
$\text{program} \rightarrow$

① Intro

Page No.

Date :

Intro to ML

ML
Data $\rightarrow$ computer $\rightarrow$ prog
Output $\rightarrow$

→ Machine Learning pipeline
- → Data Generation / Collection - generate, webscrape, kaggle
- → Data Cleaning / Feature Engineering
- → Algorithms to find patterns
- → Deployment

ml :- machines are able to learn without being explicitly program

Supervised Learning
- → learn from data (labelled)
- → generalised predictions on future data and avoid "OVERFITTING"

] TRAININ -G

x [ ] input

y [ ] label

label → continuous / discrete value

continuous
~~[illegible]~~ regression
- house price prediction
- stock price prediction

discrete value
classification
- is it cat or dog?
  0 or 1
- is it an apple?
  0 or 1

Representation

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^i & x_2^i & x_3^i & \dots & x_n^i \\ \text{---} & & x_j^i & & \text{---} \end{bmatrix} \qquad y = \begin{bmatrix} \uparrow \\ y_i \\ \downarrow \end{bmatrix}$$

In regression
$$f(x) \rightarrow y$$
$$x \in R^n$$
$$y \in R$$

In classification
$$f(x) \rightarrow y$$
$$x \in R^n$$
$y \in$ one of the classes

Batch learning

How algorithm works?

Training Data

↓

Algorithm

↓

x_{Test} → Hypothesis → y_{test} (Prediction)

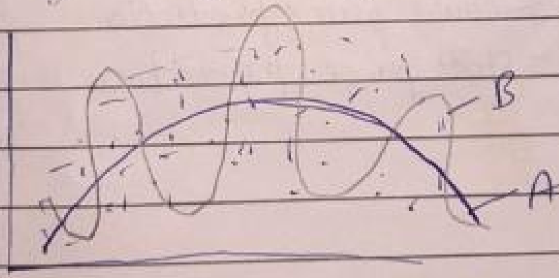
(unseen data)

↓

function parameters

In regression our goal is to reduce loss (error)
In classification our goal is to reduce misclassifications.

Overfitting ⟹ A model that fits training data well but doesn't give good generalized predictions for example.



now B covers a bigger range of points so will perform well on training data, but is not suitable for predictions whereas A is more suitable for a generalized prediction

Ex:- Spam detection, fake news prediction, document classification, NLP, image classification, sequence processing (speech, music), predictive analysis - house, stock price.

* Unsupervised Learning

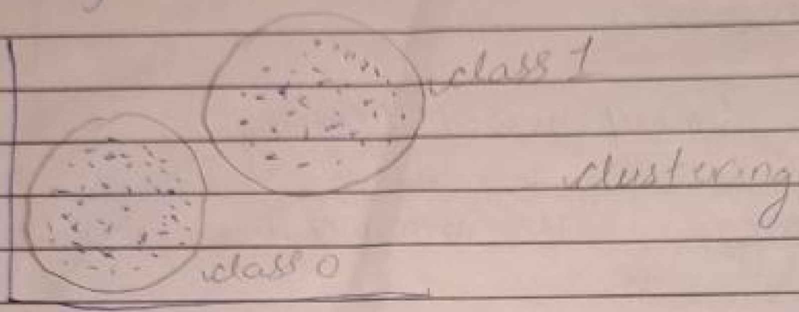
→ no supervision

→ no labelled data

→ It has to find patterns on its own and group similar data points.

→ clustering is used

→ only known x values, no knowledge of y.



ex:- tweet sentiment analysis, products bought together

* Semi - Supervised Learning

classification ex

→ we do not have labelled data prior to use

→ we manually label the data.

→ used in object segmentation

* Reinforcement Learning

dog bone whiste example

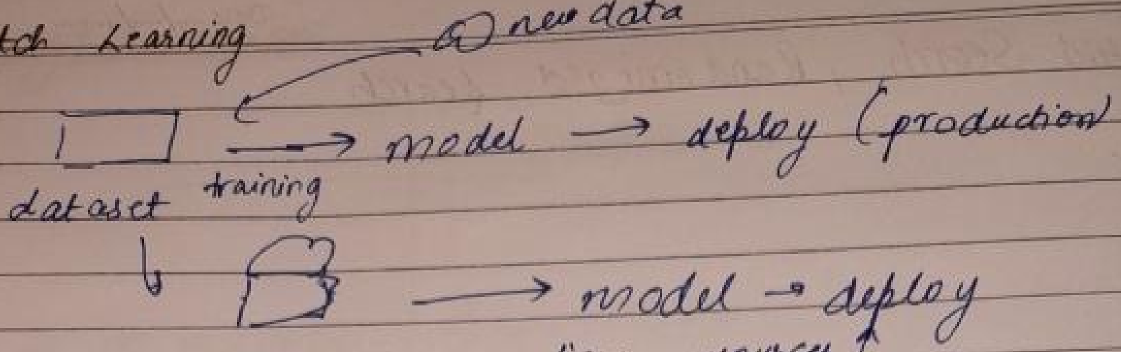
→ Hit and trial method

→ Agent interacts with the environment.

→ environment includes everything which agent interacts with.

→ If agent reaches success there's a reward, if it fails then it gets punishment/loss through which it learns. Most like humans.

Batch Learning



pros - version cons - time, resources ↑

Online learning

model → new data + data → model → new data + data

pros - time ↓ cons - bad data, no version maintenance

Instance learning

by heart learning
strict model

model based learning

↳ logic → decision

Challenges in ML

1. Insufficient amount of data - underfitting
2. nonrepresentative data
3. poor quality of data - outliers
4. irrelevant data