Introduction to Machine Learning

Agenda

- → Summary
- → ML ?
- → MLE Vs SDE
- → ML Tasks
- → Types of learning

Already Learnt

→ DS Libraries : Numpy, Pandas, Seaborn

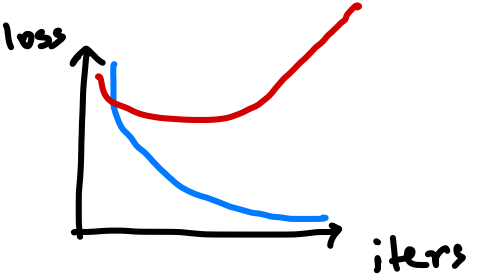
Numpy :

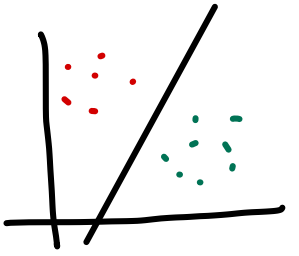
X & Y — `np.dot (X, Y)`

W, X — `np.dot (W, X)`

Pandas : read & manipulate dataframes

Matplotlib/ Seaborn :





Prob. & Stats

$P(\text{click} \mid \text{user}) = 0.96$

$P(\text{click} \mid \text{user}) = 0.88$

Std / Variance

Z-value . Standardisation $z = \frac{x - \mu}{\sigma}$ $\mu = 0$, $\sigma = 1$

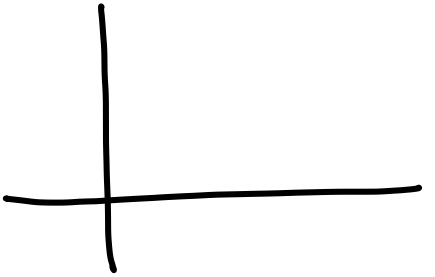
Coordinate Geom | Linear Algebra

$$y = mx + c$$

$$y = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d$$

$$\boxed{y = w^T x + w_0}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$



Calculus & Optimization

Loss func.

↓

Minimize loss

"Gradient Descent" → diff

↓

Partial diff

$$w_j = w_j - \eta \frac{\partial L}{\partial w_j}$$

ML v/s Classical Programming:

Definition by Arthur Samuel -

ML provides ability to learn without being explicitly programmed

Q.1 - But can a "dumb" computer perform anything by itself if don't tell what to do?

Q.2 - If somehow, let's assume that it is possible, shouldn't we use ML for everything?

Think : Why do we have SDEs then? Why not all of them MLEs?

Lets understand with a task

Given an email, you have to identify whether is spam or not spam

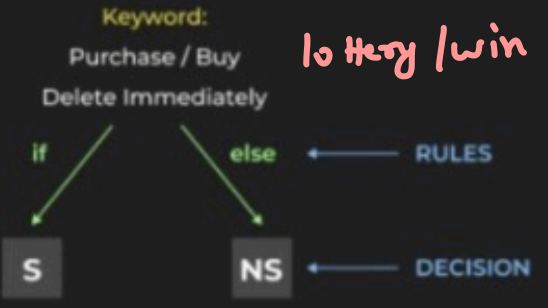
Example: Email → Spam vs Not Spam

How does the SDE pipeline look ?

1. Look at the data, find some "patterns" certain keywords, certain phrases

" Manually detect rule "

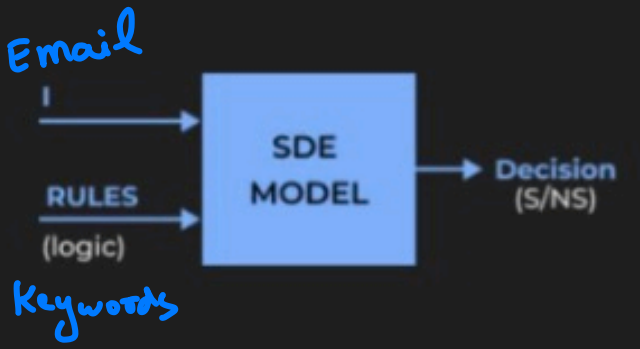
2. Build a long if-elif-else with lots of rule



lottery / win / 100%.

We can visualise the SDE approach as

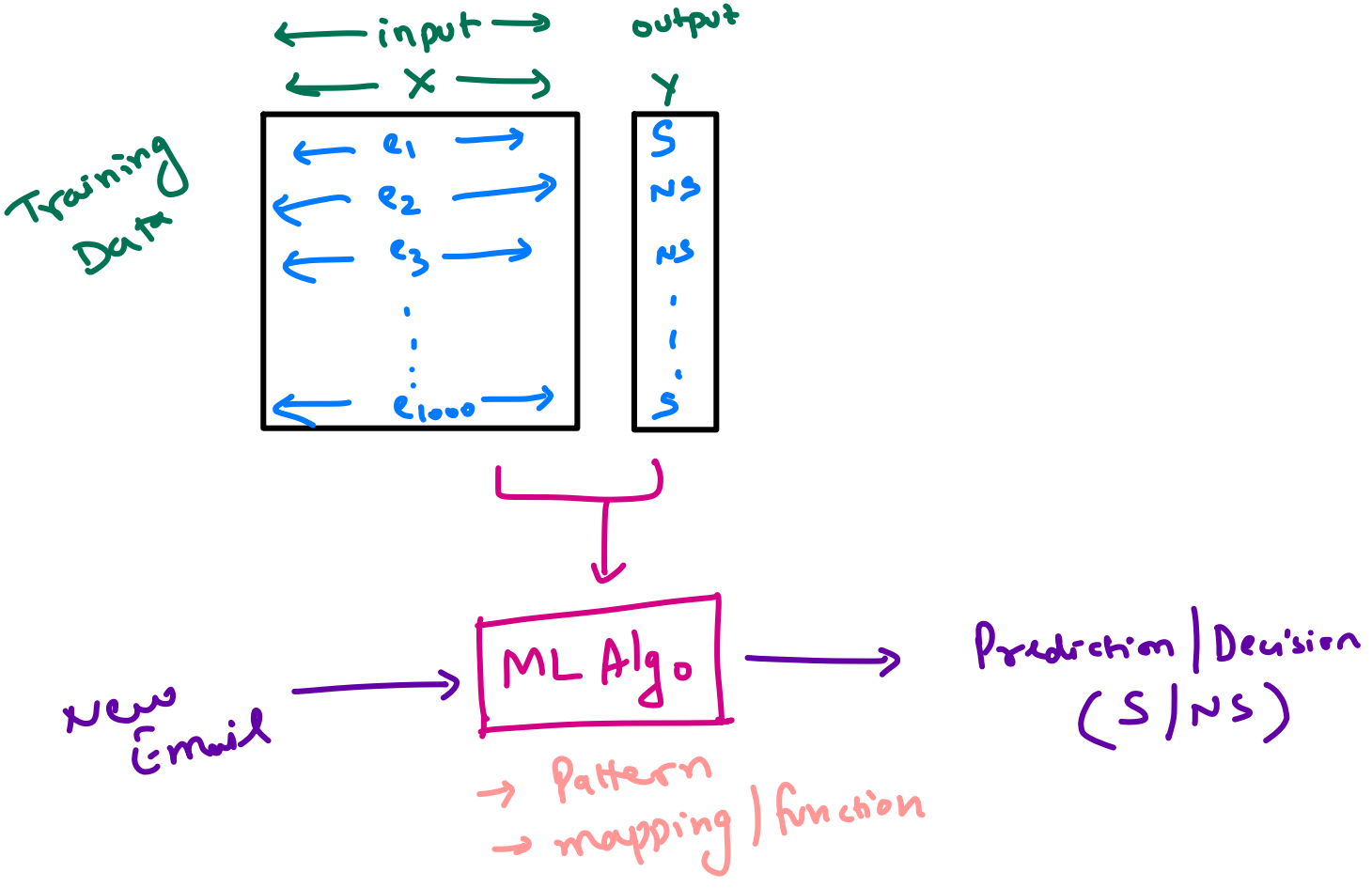
1. Inputs - Input text, Rules (written by programmer)
2. Output - Decision (Spam/Not Spam)



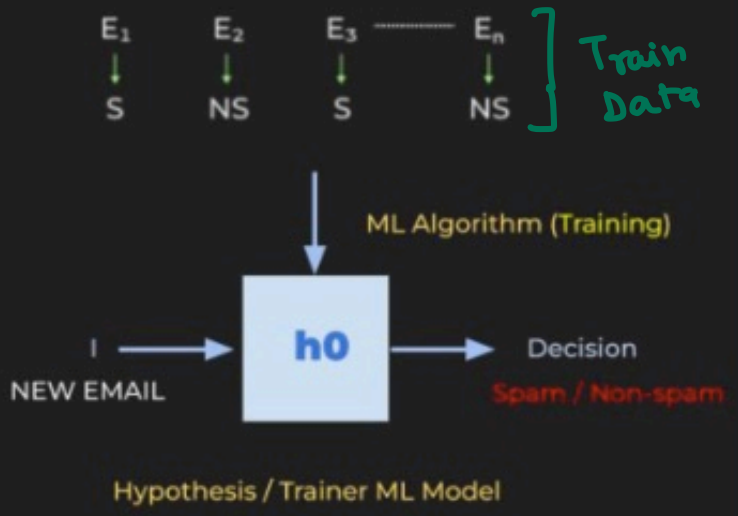
Can you think of any potential problems here?

1. **Rules are rigid** - will it handle synonyms?
2. **Lot of hard coding** - every new pattern, even similar, has to be added separately

MLE


input
X
output
Y
Training Data
e1
e2
e3
e1000
S
NS
NS
S
ML Algo
New Email
Prediction | Decision
(S/NS)
→ Pattern
→ mapping | function

How does the ML pipeline look ?



We can visualise the ML approach as

1. Inputs - Training Data (Text, Decision (Spam/Not Spam))
2. Learning Algorithm ML Algo
3. Output - Trained Model (also called Hypothesis)

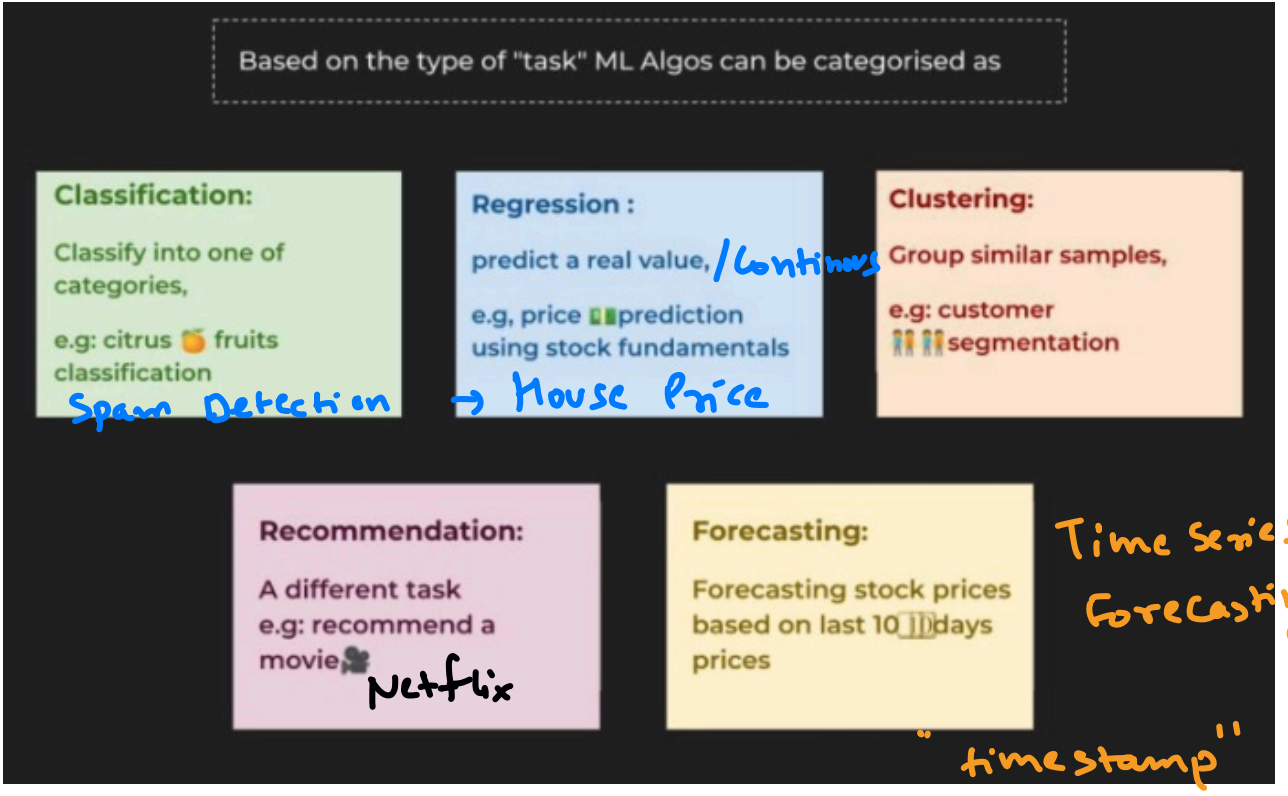
Given a "unknown" sample, we can use hypothesis to "predict" whether its a spam or not spam

Tom M. Mitchell

a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

E	T	P
Historical Email (S/NS)	Spam Detection	Accuracy
Past stock price data	Predict price Stock	Predicted ~ 1050 Actual ~ 918
____	____	____

ML Tasks


Based on the type of "task" ML Algos can be categorised as
Classification:
Classify into one of categories,
e.g: citrus fruits classification
Spam Detection
Regression :
predict a real value, / Continous
e.g, price prediction using stock fundamentals
→ House Price
Clustering:
Group similar samples,
e.g: customer segmentation
Recommendation:
A different task
e.g: recommend a movie
Netflix
Forecasting:
Forecasting stock prices based on last 10 days prices
Time Series Forecasting
"timestamp"

Which of the followings is/are true about classification?

42 users have participated

A	Classification can be defined as a predictive model mapping inputs to discrete outputs	29%
B	Class label prob. enables classification algos to predict continuous values.	7%
C	A classification algorithm can have both discrete and real-valued input variables.	12%
D	All of the options	52%

$P(y=S|x)$

↓

0.78

0.79

Based on the type of "learning" use in training the model

1

Supervised Learning:

Output labels used for training

2

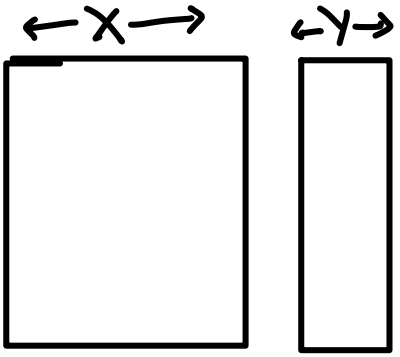
Unsupervised Learning:

Output labels are not used

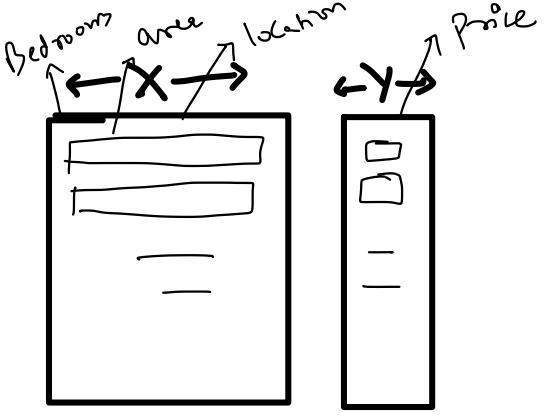
3

Reinforcement Learning:

To be discussed later

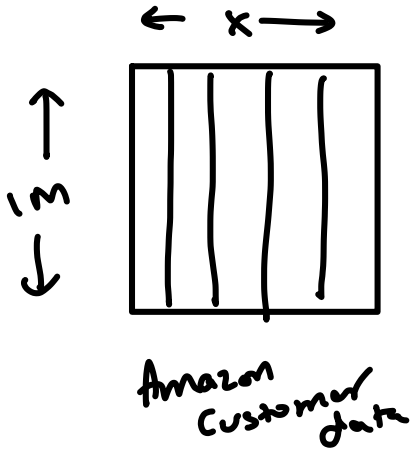


Spam Detection



House Price Prediction

Unsupervised



→ clustering

group similar customers

"3 Categories"

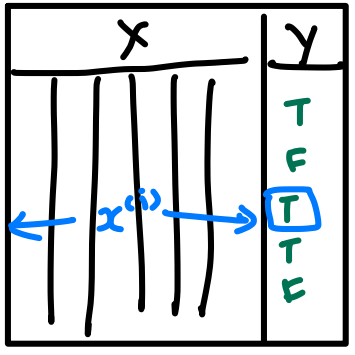
500K → Cat 1 → window shop, Students discount seekers

300K → Cat 2 → 24-30

200K → Cat 3 → rich, disc ↓ expensive

Classification

D=

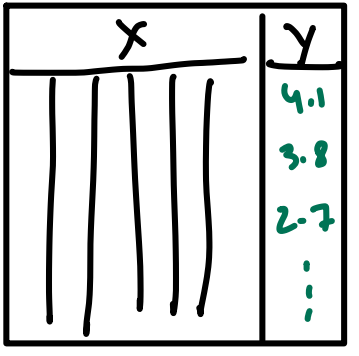


$$D = \left\{ (x^i, y^i)_{i=1}^N ;\ x_i \in R^d ,\ y_i \in \{0,1\} \right\}$$

Binary class fⁿ

Regression

D =



$$D = \left\{ (x^i, y^i)_{i=1}^N ;\ x_i \in R^d ,\ y_i \in R \right\}$$

R^1
R^2
$\vdots$
$R^d \rightarrow$ d-dim