ML-Intro

Friday, 29 July 2022 8:55 PM

WELCOME

Agenda :-

- → Summary - learnt
- → Overview (10k feet Overview)
  - → ML ? Vs SDE
  - → Tasks
  - → Learning
- → Applications

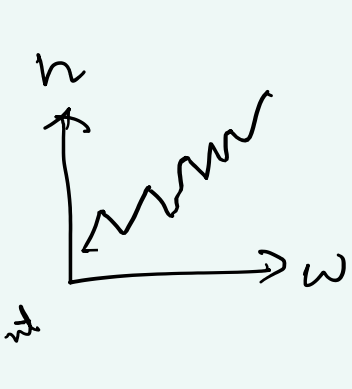
ML → ML-1 (2 months), (ML-2) (1 month) | DL

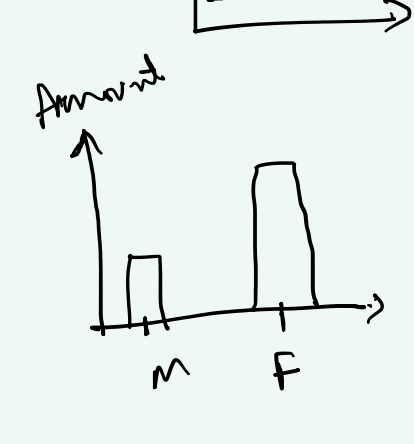
1. ① Python & DS Libraries
3. ② Prob & stats
1. ③ Coordinate Geometry (Linear Algebra)
2. ④ Calculus & optimisation

Numpy, Pandas, Seaborn

1V → Numerical : histogram, boxplot, kde, violinplot, distribution
1V → Categorical : Countplot, piechart

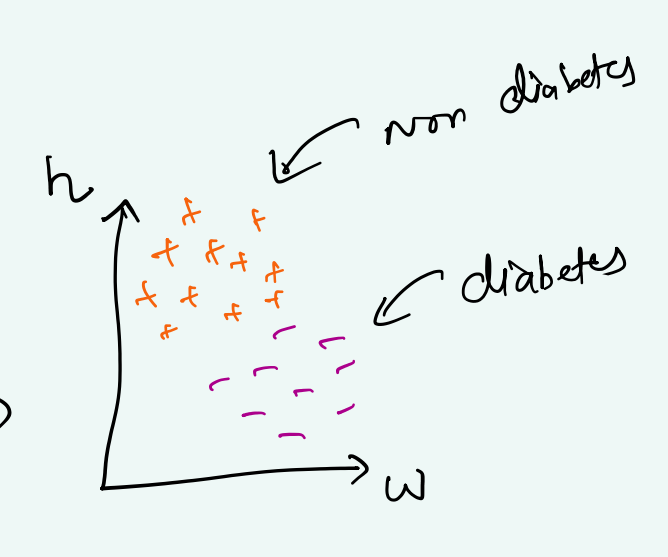
2V → N-N : Scatterplot, hexabin, lineplot
2V → N-C : boxplot / violin, barplot,
2V → C-C : Heatmap, → Contigency table, Countplot





Gender | Profession (M, F)

3V → N-N-C



Q→ P(diabetes) ??? → 300/1000 ⇒ 0.3

{ P (diabetic | Patient w = 120kg & h < 6') ⇒ 0.6

Patient$_i$ ⇒ w = 120kg & h = 6'

Case I : Patient$_i$ is diabetic
Case II : Patient$_i$ has 90% Probability of being diabetic

Linear Algebra.

Data → $x^{(i)}$ [h | w | a | g | ...] ↓ (1-dim vector)

n-datapoints → Matrix

ML Vs classical Programming

Arthur Samuel → ML Provides ability to learn (Pattern) w/o being explicitly programmed (→ telling actual logic)

Example : Email → Spam or Not Spam

Data : Text-Email

input → E_1 E_2 E_3 ... E_n
output → S NS NS S

Keywords :- { lottery, 100% off / free, asap, immediately, send money } → Spam

Rules: if → Spam; else → Non-Spam

Email → SDE Model (function) → Decision (S / NS)
(rules) (logic) → SDE Model

#Drawbacks :
- → manual logic
- → rigid rules → Worst, terrible (Synonyms)

ML-Pipeline

E_1, E_2, E_3, ..., E_n → S, NS, NS, S } → Training Data

→ ML Algo (Logistic Regression) → (Pattern) → h (Hypothesis function) (Trained ML Model)

New Email → h → Decision (S / NS) → Prediction

SDE: inputs →, logic (pattern) → [ ] → output

ML: Train data → [ ] → [ ] (input) → output

"Pattern Recognition"

SDE { 5, 12 → n % 2 == 0 → even / odd; manual logic building (Pattern Recog) }

ML { 2, 4, 6, 8, 16, 22 → even; 1, 3, 21, 19, 15, 11 → odd; Pattern / hypo. func. n % 2 == 0 }

Tom M. Mitchell :

Task (T) → S / NS classification

learn → Experience (E) (Training data) ML Algo to train

Has model learnt thing correctly? → Performance (P) (Metric)

Unseen Data / Test Data → Good Prediction.

100 New Email → 70 Email (NS), 30 Email (S) | 92/100 ⇒ 92%

ETP

	T	E	P
①	Predict Stock Price	Historical Price data	$P_{Predicted} \approx P_{Actual}$, difference = 0
②	Classify Images { O → T1, K → T2, M → T3 }	Images + Labels	Accuracy. [O] 100%; P(O) = ↑, P(K) = ↓, P(M) = ↓

Type of Tasks.

- → classification
- → Regression → Price, House Sale Price
- → clustering → (group), (Customer segmentation)
- → Recommendation
- → Time Series forecasting → Stock Prediction, Uber fare Prediction

Type of learning

1. ① Supervised Learning
2. ② Unsupervised Learning
3. ③ Reinforcement Learning
4. ④ Semi-Supervised.

T : Predict Blood glucose level. → continuous value

D (n × d): columns 1, 2, 3, ..., d, y; rows 1, 2, 3, i, ..., n; $x^{(1)}$, $x^{(i)}$; y_i → glucose level

$y_i \in \mathbb{R}$ → $(-\infty, \infty)$

$$\mathcal{D} = \{ (x^{(i)}, y^{(i)})_{i=1}^{n} ; x^{(i)} \in \mathbb{R}^d ; y^{(i)} \in \mathbb{R} \}$$

(input, output) Target output is real value. → Regression

D = columns 1, 2, 3, ..., d, y; rows 1, 2, 3, i, ..., n; $x^{(1)}$, $x^{(i)}$; y_i → T1 | Diabetic, T2 | Non-diabetic → Category. (n ×

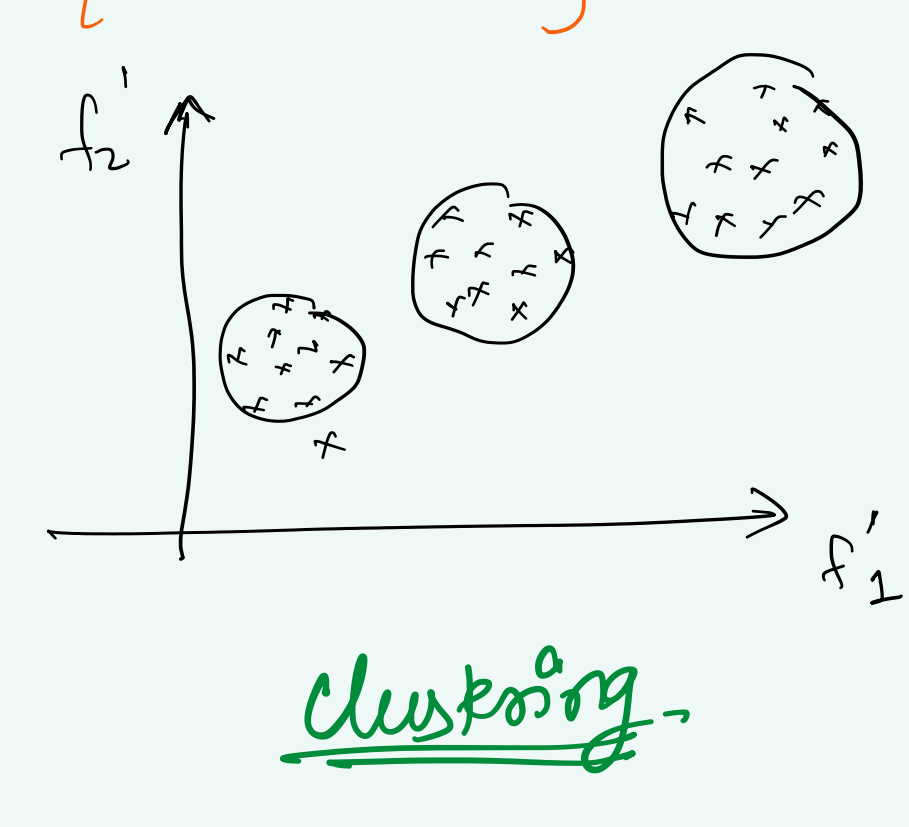
$$D = \{ (x^{(i)}, y^{(i)})_{i=1}^{n} ; x^{(i)} \in \mathbb{R}^d ; y^{(i)} \in \{0, 1\} \}$$

Target is categorical

Customer Segmentation → Low segment, Mid-Segment, High Segment

f_1 f_2 f_3 ... f_d; rows 1, 2, 3, i^{th}, ..., n; $x^{(i)}$, $x^{(k)}$

$$D = \{ (x^{(i)})_{i=1}^{n} \}$$



Clustering.

Amazon (100M) → is Prime? → $ spent in 3 months → location

Rec-Sys.

User → V10, V12, V16 ...
Recommend → V15, V36 ..

Youtube / Netflix → 100 M Videos

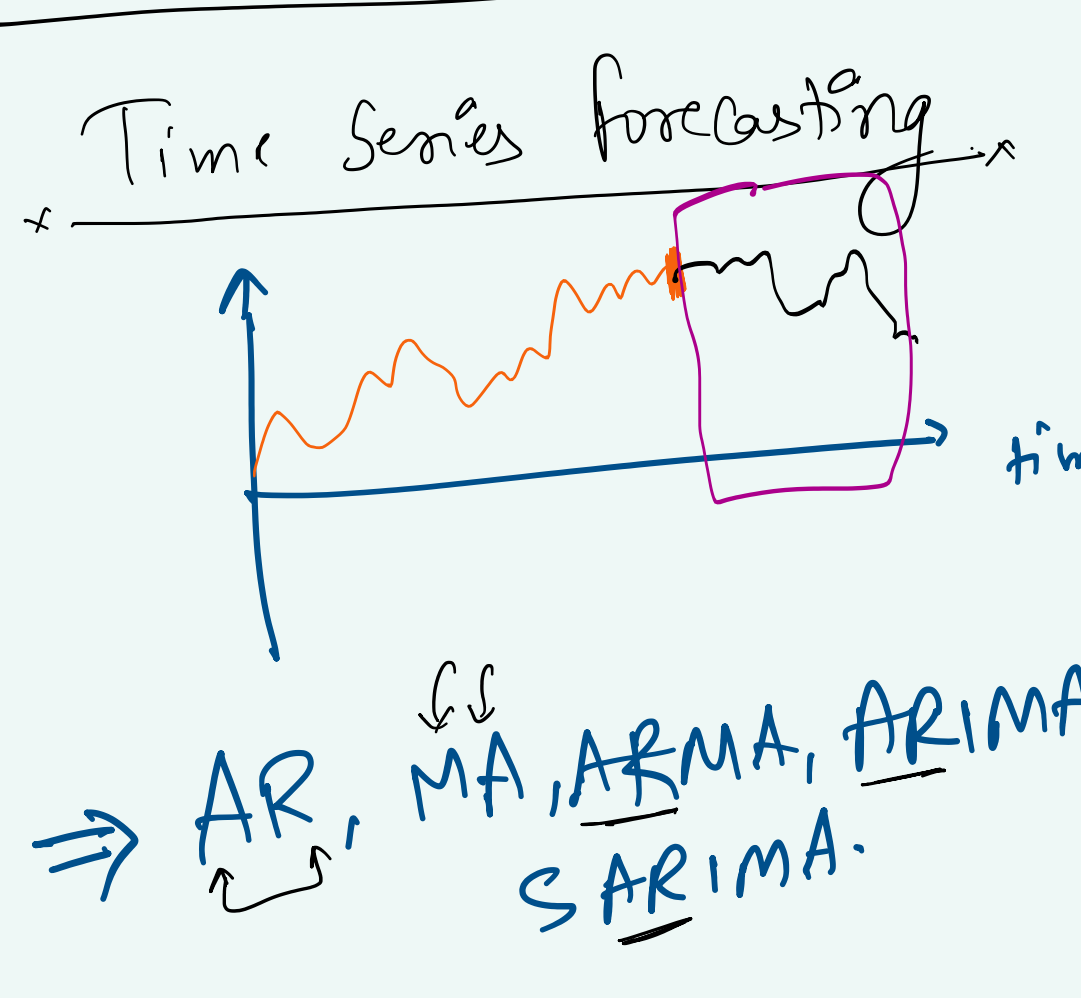
Superv | Unsuper.

Rec-Sys → item-item Similarity → Collaborative filtering → Matrix factorisation (SVD)

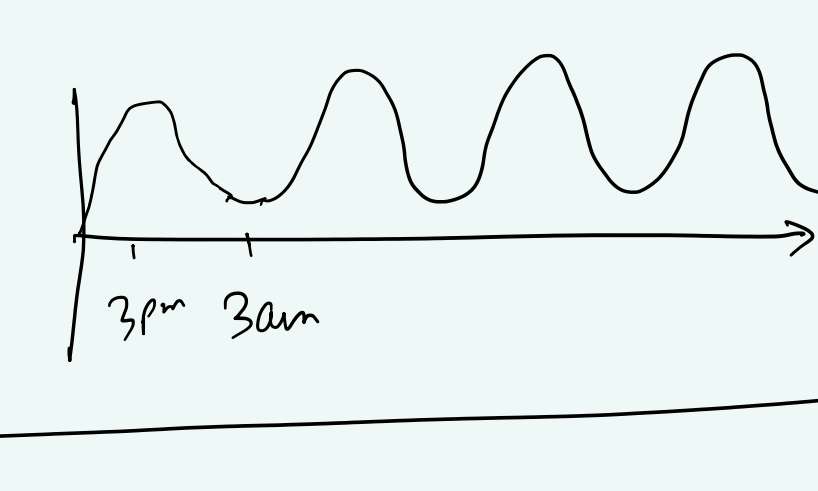
V5 ↔ V7

eval, evec

Time Series forecasting



→ Pattern
→ seasonality



⇒ AR, MA, ARMA, ARIMA, SARIMA.

Reinforcement learning