MLF Week 1 : Introduction to machine learning

Lecture 1 : What is Machine Learning

Outline

1. **What is Machine Learning?**
2. The Wonders of Machine Learning
3. Data, Models and ML Tasks
4. Supervised Learning
    1. Regression
    2. Classification
5. Unsupervised Learning
    1. Dimensionality Reduction
    2. Density Estimation

Machine Learning Definition

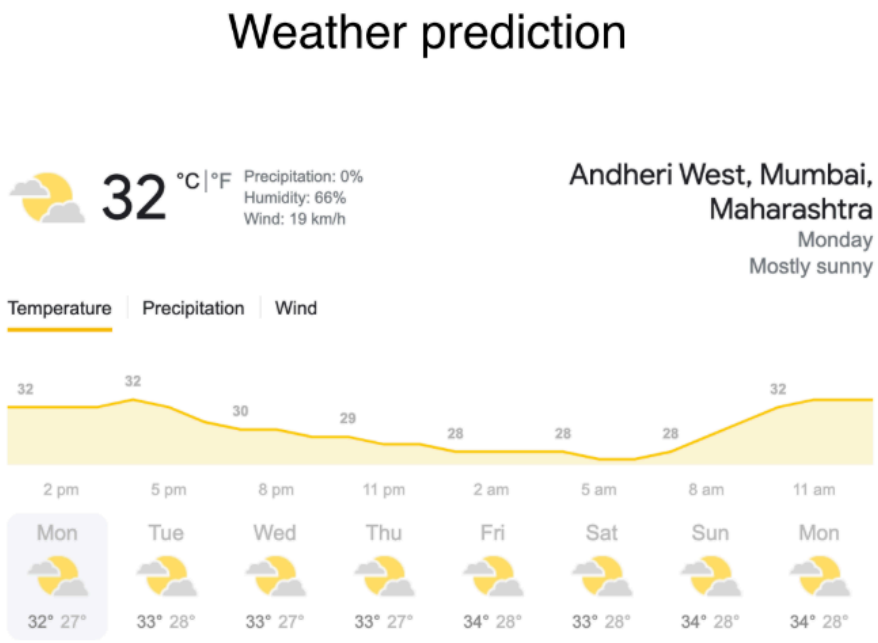
> Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data.

Machine Learning (ML) is a **branch of Artificial Intelligence (AI)** that focuses on building systems that can **learn from data and improve their performance over time without being explicitly programmed**.

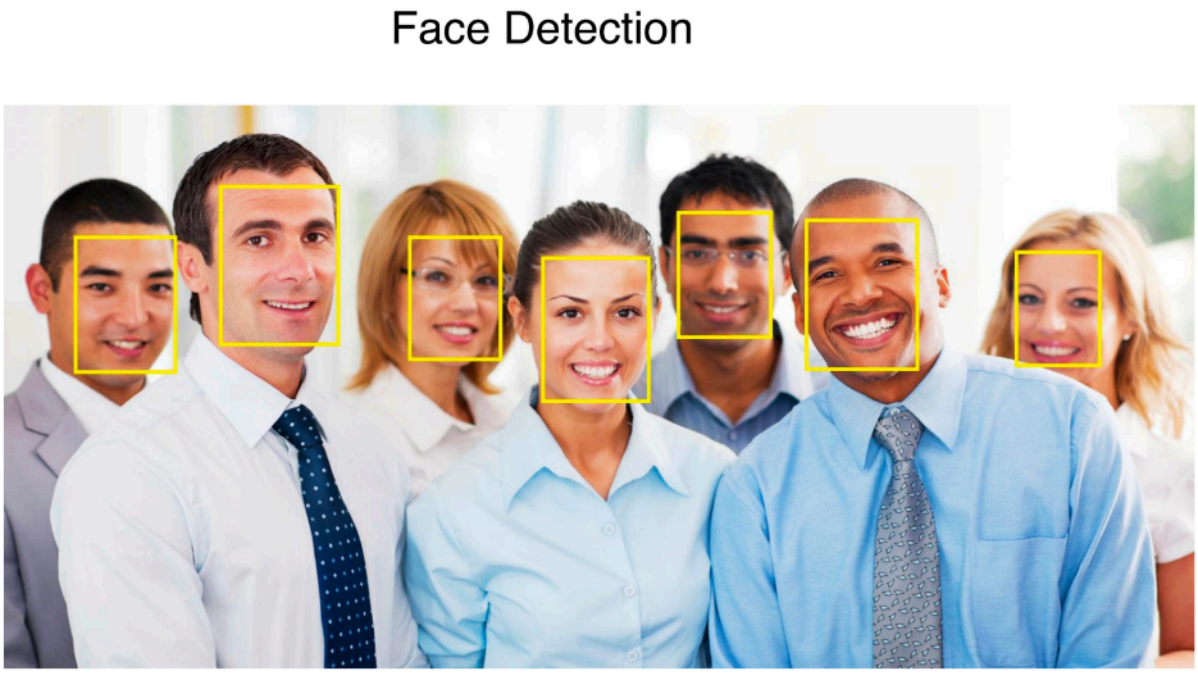
Instead of writing step-by-step instructions for solving a problem, in ML we provide data to algorithms, and these algorithms find patterns, make predictions, or take decisions automatically.

ML Tasks : Ex

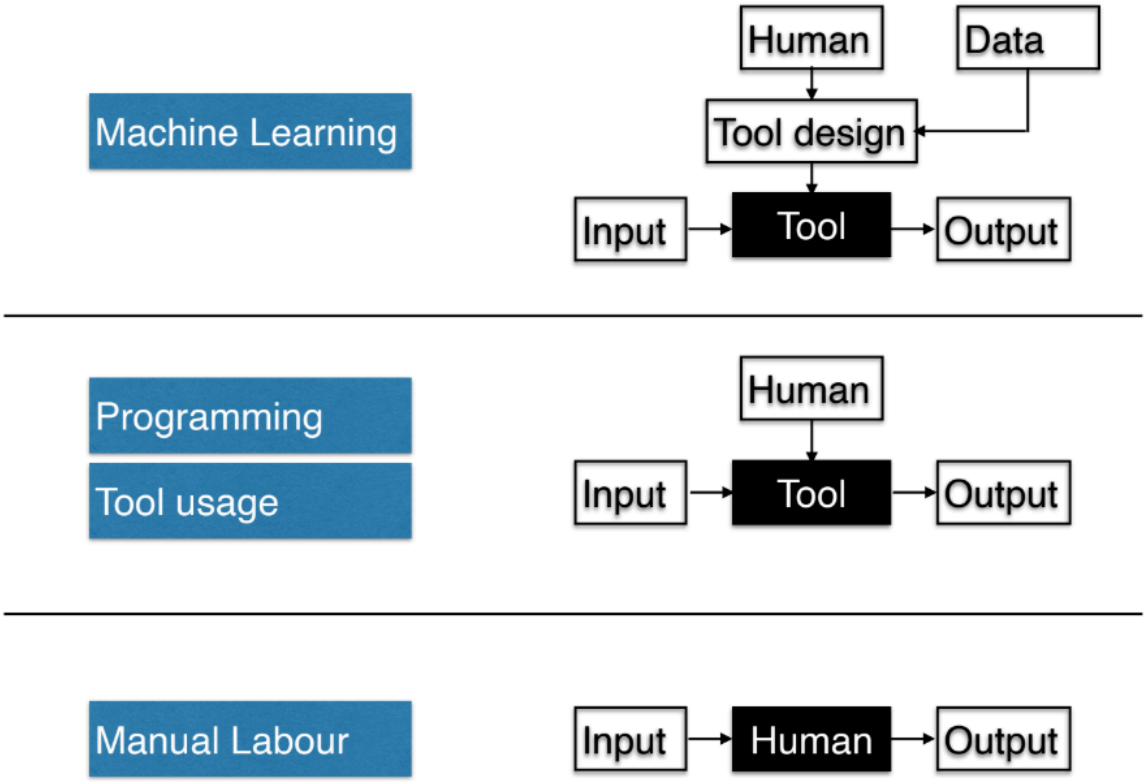
1.



2.



Task Hierarchy

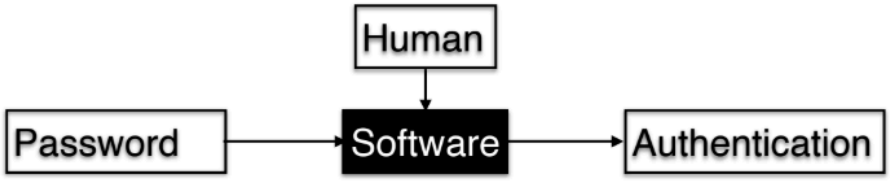


Why and When Machine Learning?

1. Programming/Human Labour Fails
    1. Scale/Speed/Cost of human labor
    2. Inability to express rules using language.
    3. Don't know the exact rules transforming input to output.
2. Machine Learning can succeed.
    1. Have lots of example data
    2. Have some structural idea on the rules

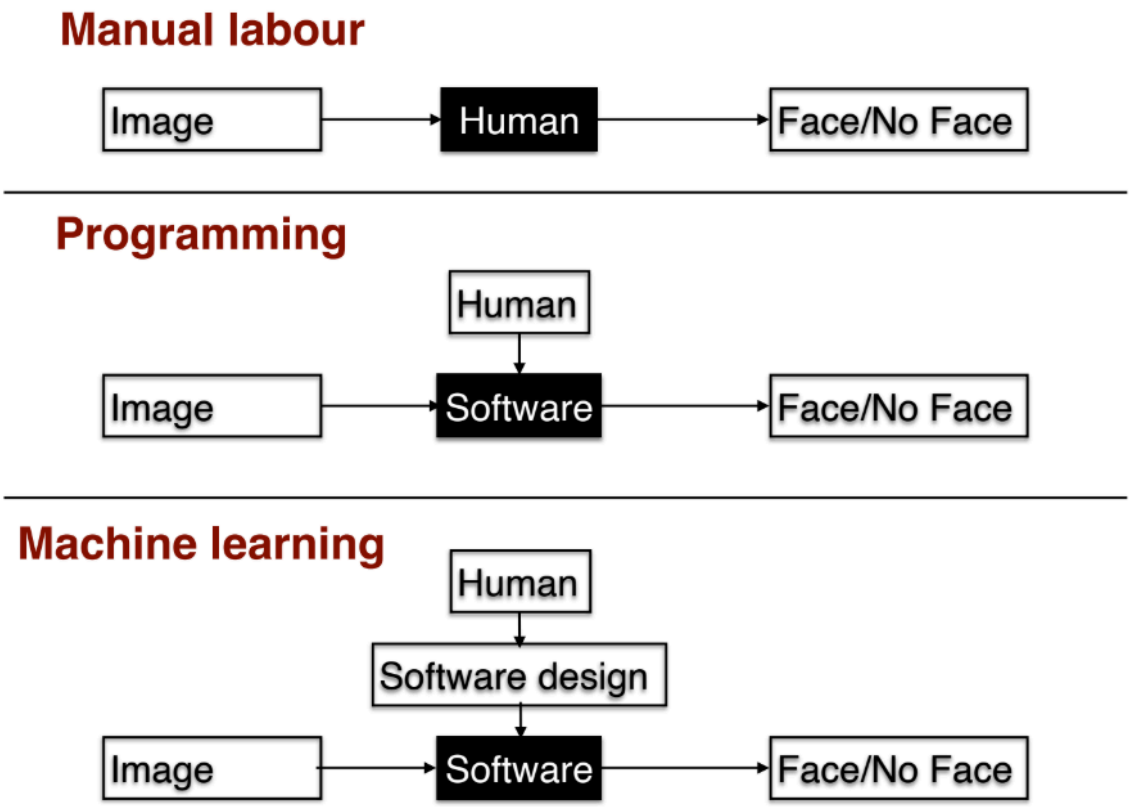
Task Analysis : Password Verify

Programming



- Problems with Manual Labour
  - Scale: Having humans check login details of every login is impractical.
- Problems with programming
  - None.
- Machine Learning not required.

Task Analysis : Face Detection

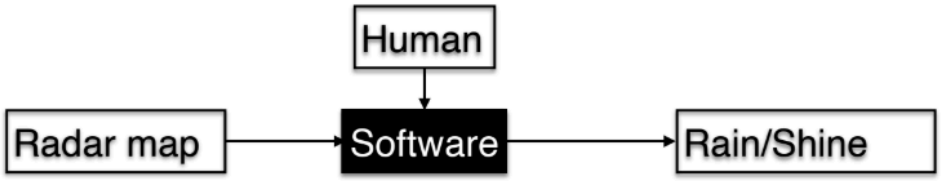


- Problems with Manual Labour
  - Scale: Having humans check all faces of every image is impractical.
- Problems with programming
  - Expressing face/not face in code is impossible.
- Case for Machine Learning
  - Lots of images available.

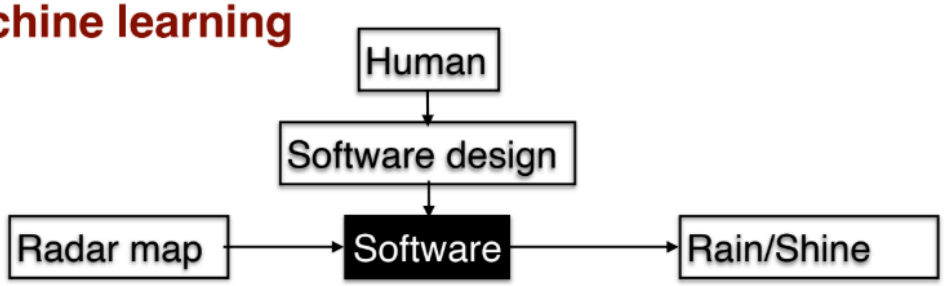
Task Analysis : Weather Prediction

Manual labour

Programming



Machine learning



- Problems with Manual Labour
  - Humans just do not know the full rules and can't process that much information.
- Problems with programming
  - Cannot code unknown rules.
- Case for Machine Learning
  - Lots of weather data available.

The Wonders of Machine Learning

1.What is Machine Learning??

2.The Wonders of Machine Learning

3.Data, Models and ML Tasks

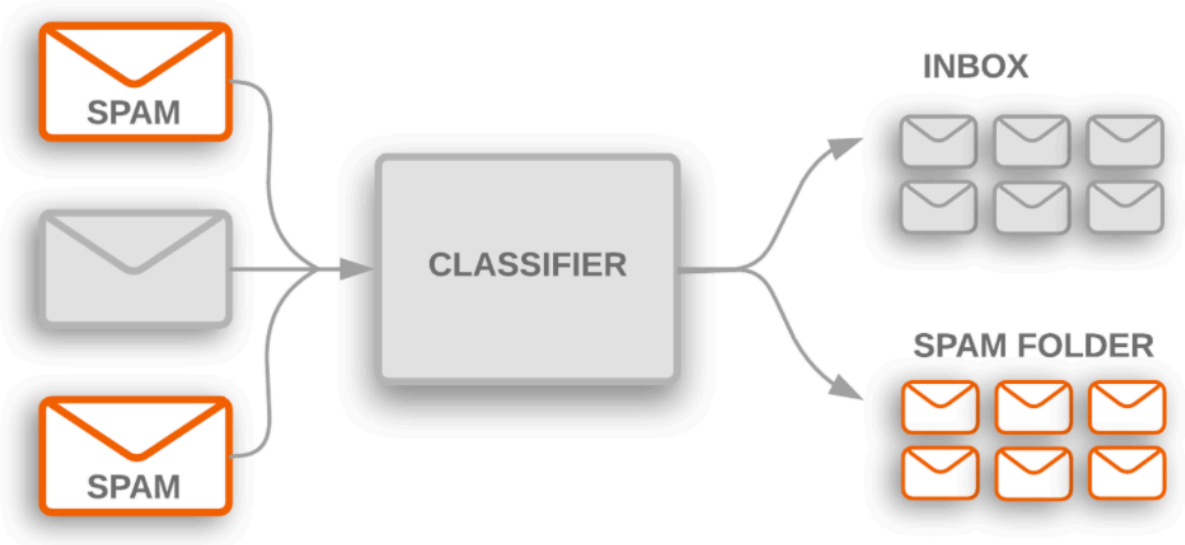
4.Supervised Learning

1. Regression
2. Classification

5. Unsupervised Learning

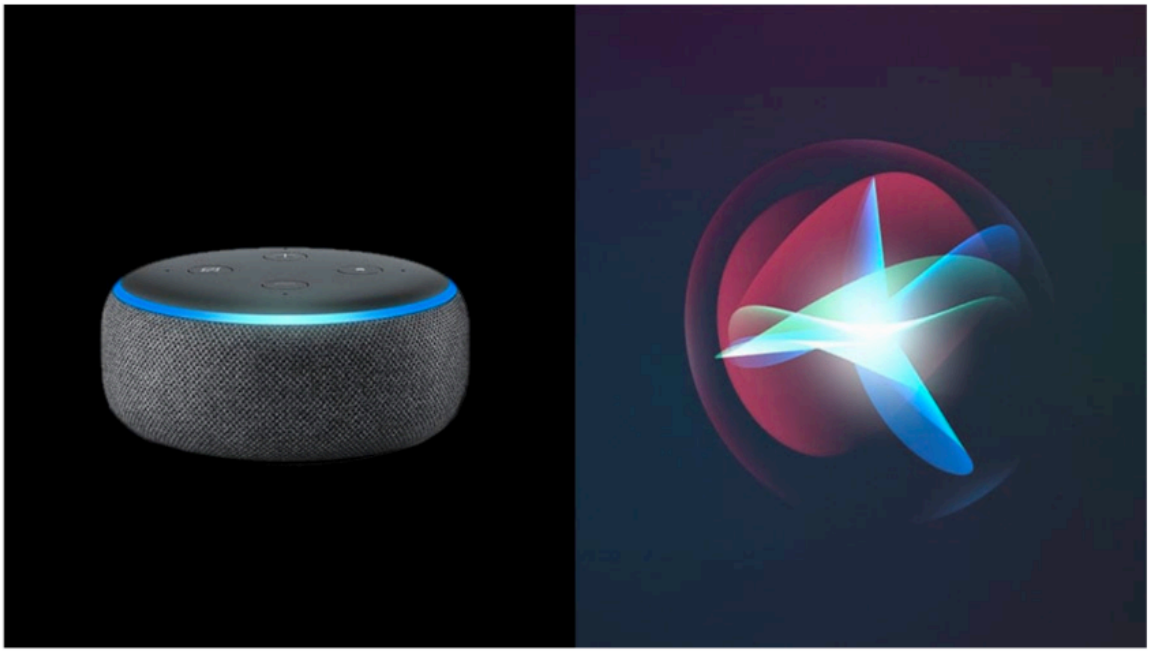
1. Dimensionality Reduction
2. Density Estimation

Machine Learning in your Inbox

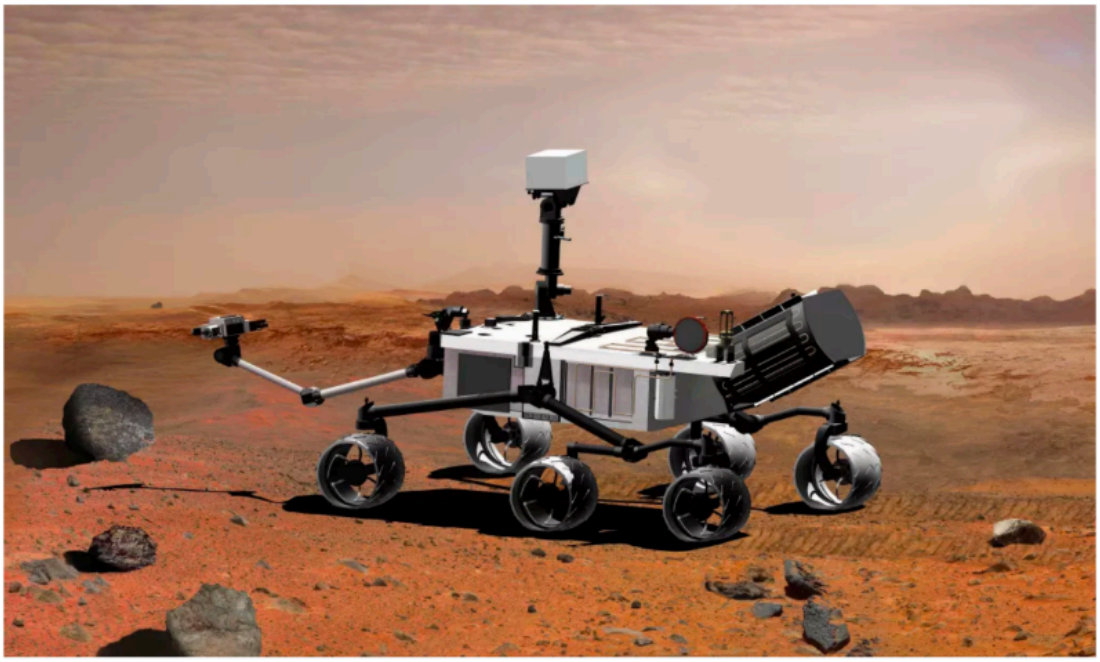


Machine Learning in your Shopping Cart

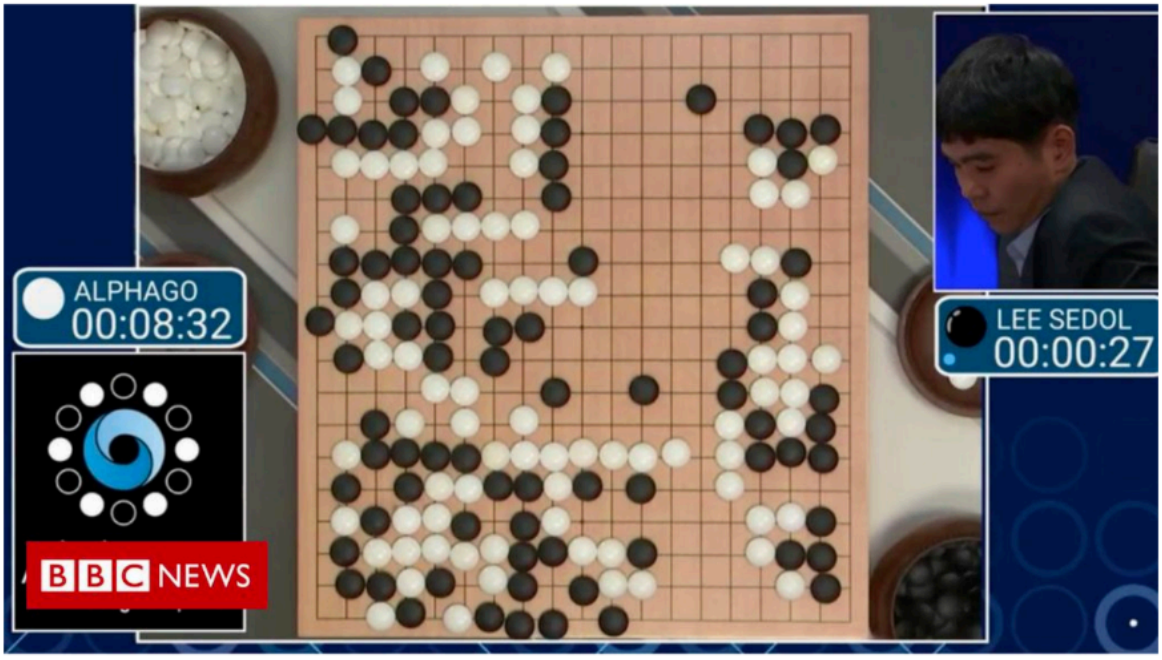
Machine Learning in your Smart Assistant



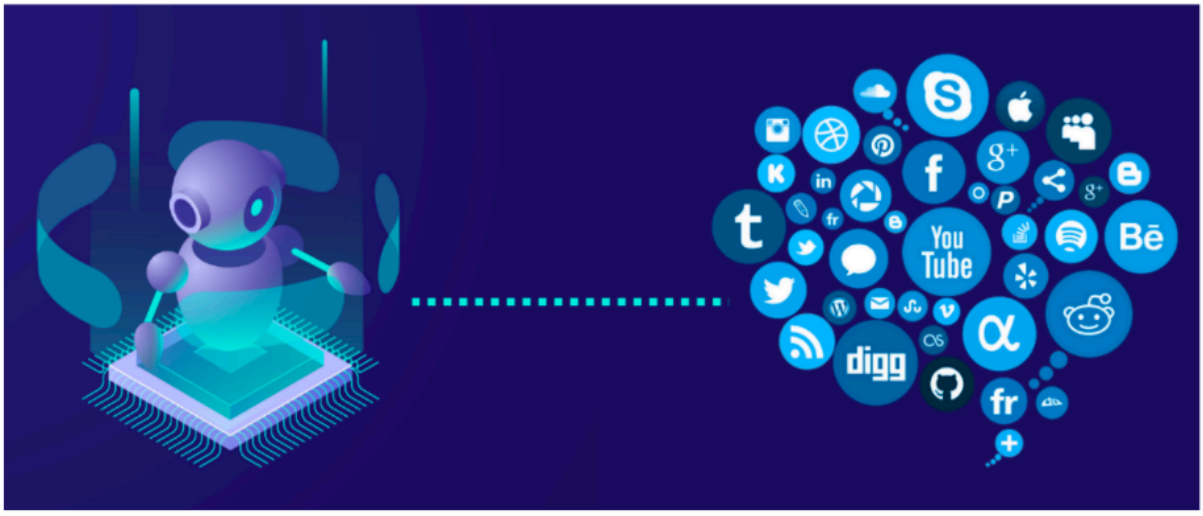
Machine Learning in Robot AIs



Machine Learning in Games



Machine Learning in Marketing



QN : 2
1st Try to understand the problem statement

Lecture 2 : Data, Models and ML Tasks

Outline

1. What is Machine Learning??
2. The Wonders of Machine Learning
3. **Data, Models and ML Tasks**
4. Supervised Learning
    1. Regression
    2. Classification
5. Unsupervised Learning
    1. Dimensionality Reduction
    2. Density Estimation

What is Data?

Data is any **raw fact, figure, or information** that can be collected, stored, and processed. It can be numbers, text, images, audio, video, or even signals from sensors.

On its own, data may not have meaning — but when **organized and analyzed**, it becomes **information** and then **knowledge**.

Data is a collection of vectors.

E.g.

3	9	1.9	5.0	House 1
2	7	2.1	3.2	House 2
4	12	2.8	6.6	House 3
5	16	0.9	9.8	House 4
5	15	3.1	8.5	House 5
4	11	1.6	6.9	House 6

Metadata is information on the data.

E.g. : (# rooms, Area in 100 sq.ft, Distance to metro in km, Price in 10 lakhs)

What is a Model?

A model is a mathematical simplification of reality.

Some examples:

The Ideal Gas model

Inverse square law for gravitational attraction

Moore's Law for semiconductors

Cobb–Douglas model in Economics

"All models are wrong, but some are useful"

George Box

In **Machine Learning (ML)**, a **model** is a **mathematical representation of a real-world process** that is trained to make predictions or decisions based on data.

Think of it as the **"brain" of ML** that learns patterns from past data and applies them to new, unseen data.

Types of Models in ML

- Predictive Model
  - Regression Model
  - Classification Model
  - …
- Probabilistic Model
-

1.Predictive Models

A **Predictive Model** is a type of machine learning model that uses **historical data** to **make predictions about future or unknown outcomes**.

It doesn't just explain past data — it tries to **forecast what will happen next**.

a. Regression Model

- A **regression model** is used when the output (target) is a **continuous numeric value**. **(real-valued numbers)**
- **Goal:** Predict "how much" or "how many."
- **Examples:**
  - Predicting house price (₹50,00,000).
  - Predicting temperature tomorrow (30.5°C).
  - Predicting a student's exam score (85 marks).

Regression Model

Model the price of a house based on its area and distance to metro.

Example good model:

Price = 0.5*Area - Distance

b. Classification Model

- A **classification model** is used when the output (target) is a **category or class label (discrete value)**.
- **Goal:** Predict “which class.”
- **Examples:**
  - Is an email **spam or not spam**?
  - Will a customer **buy or not buy** a product?
  - What digit is this handwritten number (0–9)?

Classification Model

Model whether a house is closer than 2kms to a metro based on price and area

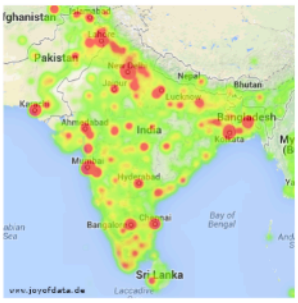
Example good model:

Answer = Close if 2*ROOMS-PRICE < 1
Far otherwise

2.Probabilistic Model

A **probabilistic model** is a type of model that represents **uncertainty** using the rules of probability.
Instead of making a single "hard" prediction, it outputs a **probability distribution** over possible outcomes.

In short: It tells you **not just what might happen, but also how likely it is**.



What is the probability that a randomly chosen person is in lat-long : (25N,30E) ?

What is the probability that a given tweet was generated by Mr. Chopra?

Learning Algorithms

A **learning algorithm** is the **method or procedure** that a machine learning system uses to **learn patterns from data** and improve its performance on a task.

Think of it like a **recipe**:

- **Ingredients** = Data
- **Recipe (Algorithm)** = Step-by-step process to adjust the model
- **Final Dish (Model)** = Trained system that can make predictions

Types of Learning Algorithms

1. Supervised Learning Algorithms

Learn from labeled data (input + correct output).

- Examples:
    - **Linear Regression** → Predict house prices
    - **Logistic Regression** → Spam or not spam
    - **Support Vector Machines (SVM)**

2. Unsupervised Learning Algorithms

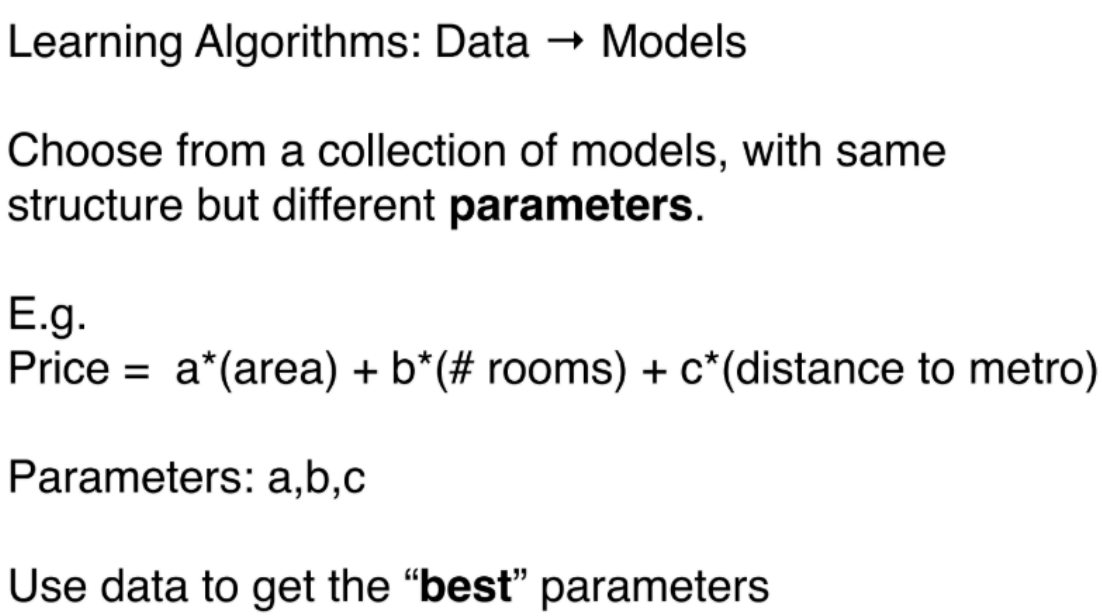
Learn from unlabeled data (no correct output).

- Examples:
    - **K-Means Clustering** → Group customers
    - **PCA (Principal Component Analysis)** → Dimensionality reduction

3. Reinforcement Learning Algorithms

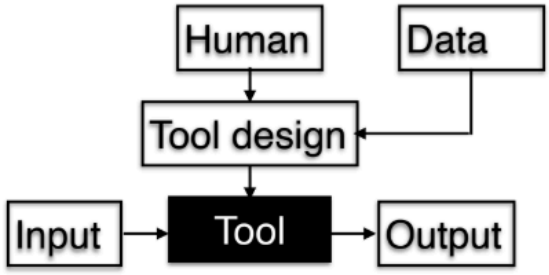
Learn by interacting with an environment and receiving rewards/penalties.

- Examples:
    - **Q-Learning**
    - **Deep Q-Networks (DQN)**

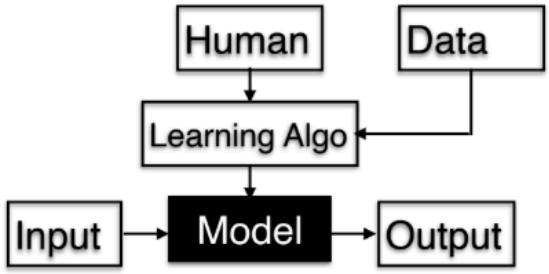


Machine Learning Tasks Revisited

Machine Learning



Machine Learning



Supervised = Learn with labels (predict values/categories).
Unsupervised = Learn without labels (find patterns/clusters).

Lecture 3 : Supervised Learning - Regression

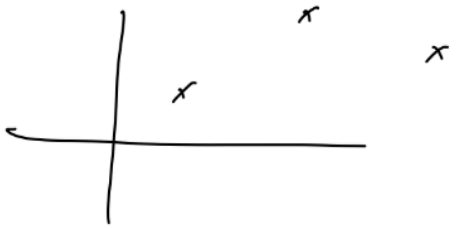
Notation

$x^1 = [1, 2, 3]$
$x^2 = [7, 9, 9]$
$x_2^2 = 9$
$\begin{pmatrix} 1.3 \\ -7.6 \\ 5.9 \end{pmatrix} \in \mathbb{R}^3$

- $\mathbb{R}$: real numbers, $\mathbb{R}_+$: Positive reals, $\mathbb{R}^d$: d-dimensional vector of reals.
- $\mathbf{x}$: vector. x_j: j^{th} co-ordinate. $\|\mathbf{x}\|$: Length of vector $\mathbf{x}$.
- $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n$: Collection of n vectors.
- x_j^i : j^{th} co-ordinate of i^{th} vector.
- $(x_1)^2$: Square of the first co-ordinate of the vector $\mathbf{x}$
- $\mathbf{1}(2 \text{ is even}) = 1$, $\mathbf{1}(2 \text{ is odd}) = 0$.

$X = \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix}$
$x_3 = 7$
$\|x\|^2 = x_1^2 + x_2^2 + \ldots + x_d^2$

Supervised Learning



- Supervised learning is curve-fitting.
- Given $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$ $\in \mathbb{R}^d$
- Find a model f such that $f(\mathbf{x}^i)$ is 'close' to y^i

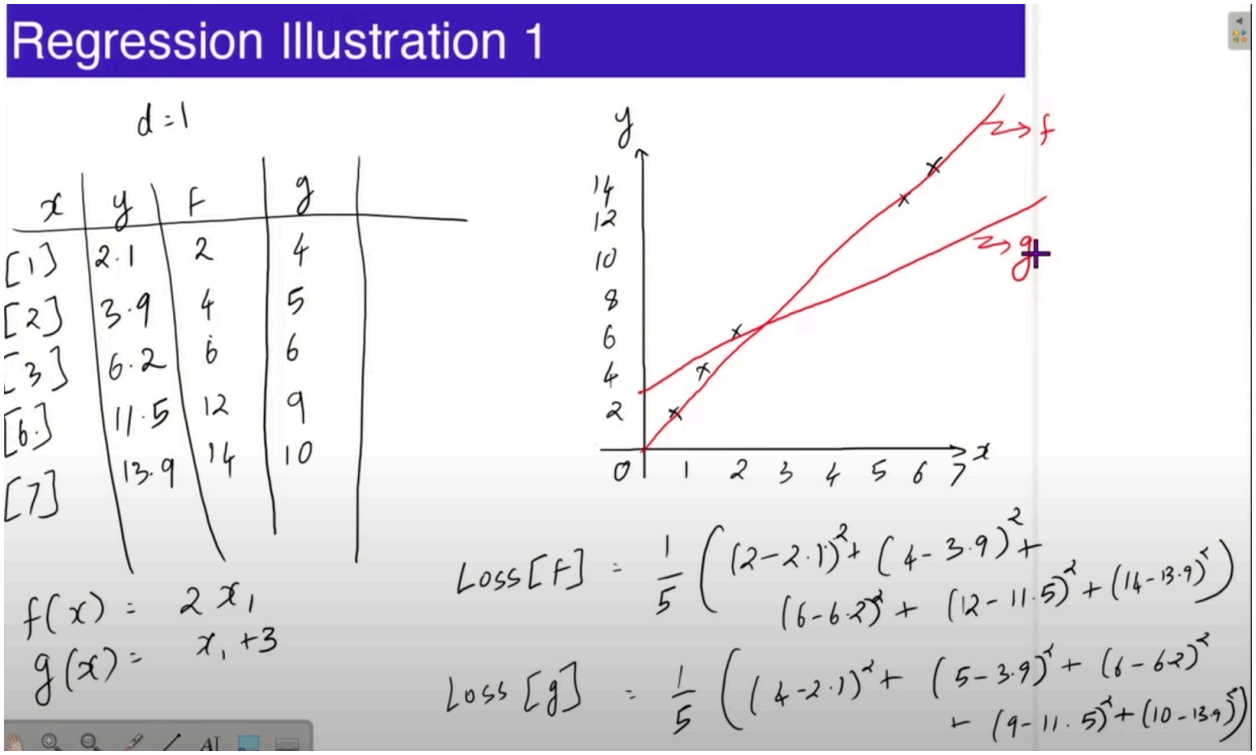
f is a function

Regression

- E.g. Predict house price from room, area, distance.
- Training data: $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$ ($\mathbf{x}^i \in \mathbb{R}^3$, $y^i \in \mathbb{R}$)
- $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \mathbb{R}$
- Algorithm outputs a model $f : \mathbb{R}^d \to \mathbb{R}$
- Loss$[f] = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^i) - y^i)^2$ = Squared loss
- $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \sum_{j=1}^{d} w_j x_j + b$

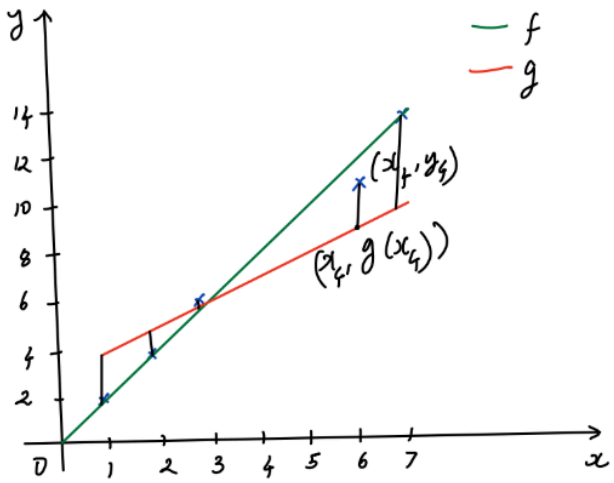
Linear Parameterisation = w_1 (# rooms) + w_2 (area) + w_3 (distance) + b

Regression Illustration 1



$d = 1$

x	y	f	g
[1]	2.1	2	4
[2]	3.9	4	5
[3]	6.2	6	6
[6]	11.5	12	9
[7]	13.9	14	10



$f(x) = 2x_1$

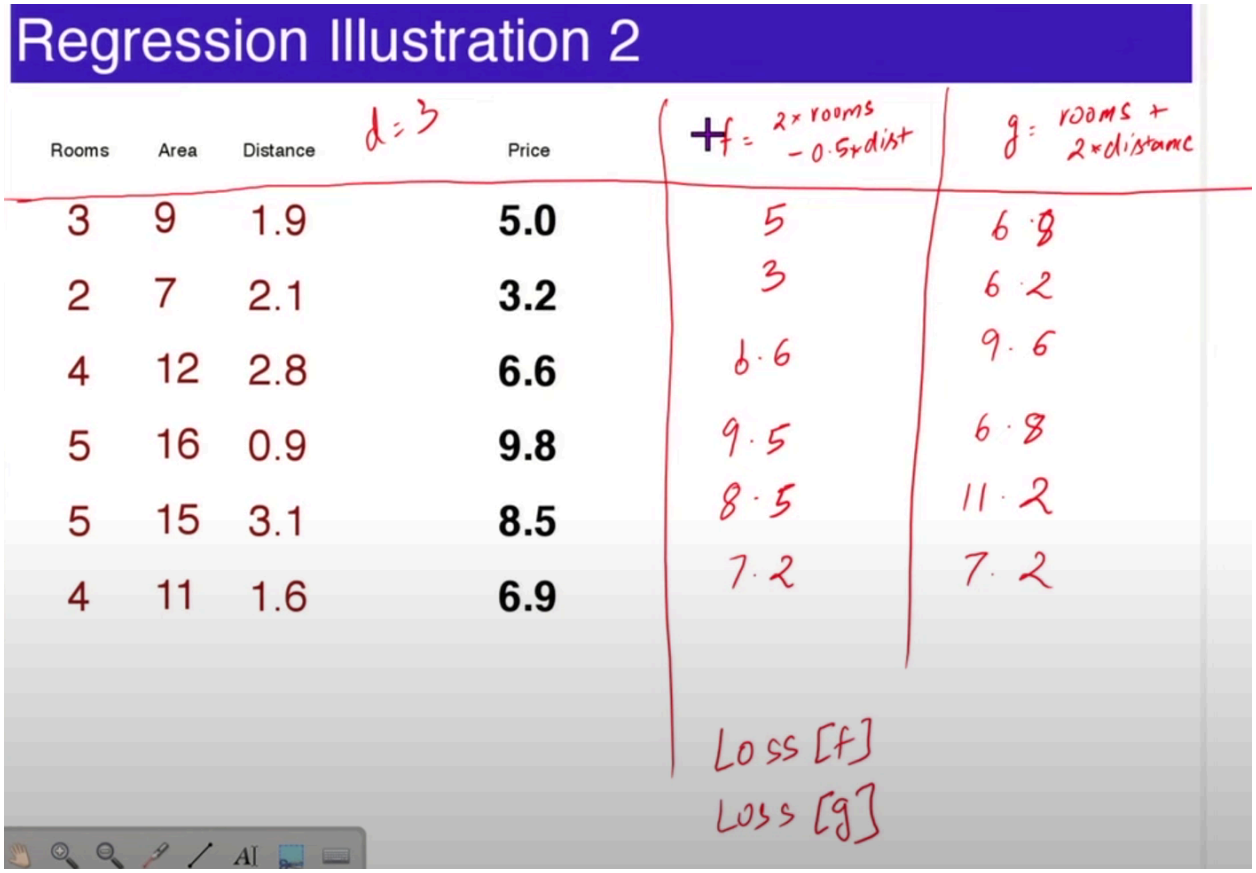
$g(x) = x_1 + 3$

$$\text{Loss}[f] = \frac{1}{5}\left((2-2.1)^2 + (4-3.9)^2 + (6-6.2)^2 + (12-11.5)^2 + (14-13.9)^2\right)$$

$$= \frac{1}{5}(0.3)$$

$$\text{Loss}[g] = \frac{1}{5}\left((4-2.1)^2 + (5-3.9)^2 + (6-6.2)^2 + (9-11.5)^2 + (10-13.9)^2\right)$$

Regression Illustration 2



Rooms	Area	Distance	Price	f = 2 * Rooms − 0.5 * dist	g = Rooms + 2 * distance
3	9	1.9	**5.0**	5.05	6.8
2	7	2.1	**3.2**	2.95	6.2
4	12	2.8	**6.6**	6.6	9.6
5	16	0.9	**9.8**	9.5	6.8
5	15	3.1	**8.5**	8.5	11.2
4	11	1.6	**6.9**	7.2	7.2

Lecture 4 : Supervised Learning - Classification

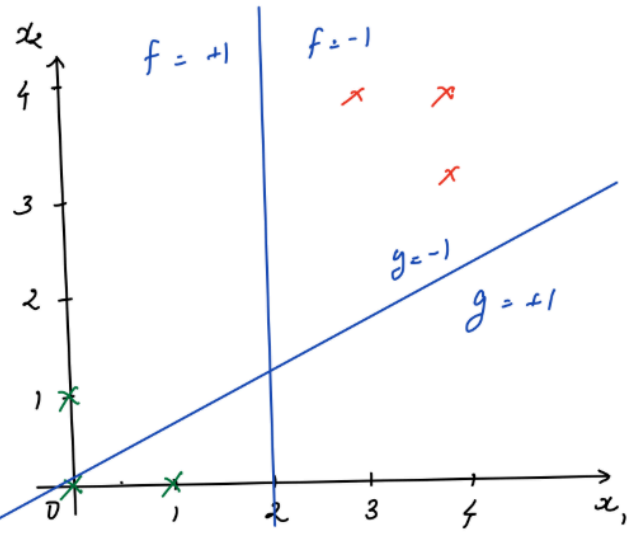
Classification

- E.g. Predict if rooms>3 from area and price.
- Training data: $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^n, y^n)\}$
- $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \{+1, -1\}$
- Algorithm outputs a model $f : \mathbb{R}^d \rightarrow \{+1, -1\}$
- Loss$[f] = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(f(\mathbf{x}^i) \neq y^i)$ = Fraction of training data classified wrongly by f
- $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$

  Linear separator

Classification Illustration 1

x	y	f	g
$[0,0]$	$+1$	$+1$	$+1$
$[0,1]$	$+1$	$+1$	-1
$[1,0]$	$+1$	$+1$	$+1$
$[4,4]$	-1	-1	-1
$[3,4]$	-1	-1	-1
$[4,3]$	-1	-1	-1



$f(x) = \text{Sign}(2 - x_1)$

$g(x) = \text{Sign}(x_1 - 2x_2)$

$\text{Loss}[f] = \frac{1}{6}(0) = 0$

$\text{Loss}[g] = \frac{1}{6}(1) = \frac{1}{6}$

Classification Illustration 2

Area	Price	Rooms
9	5.0	-1
7	3.1	-1
12	6.9	+1
16	9.7	+1
15	8.5	+1
11	7.1	+1

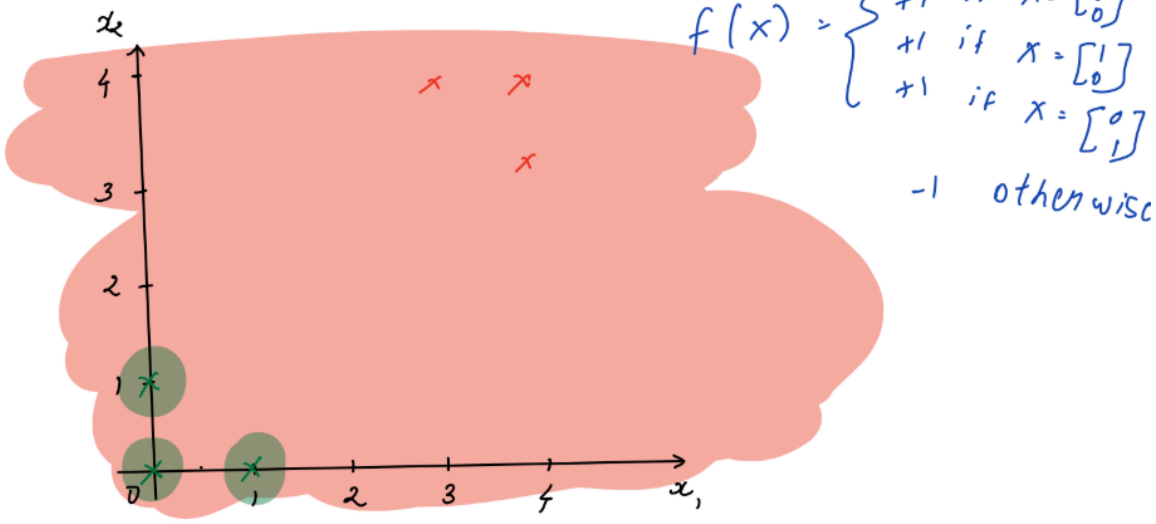
Rooms=1 or 2 or 3 is encoded as -1.
Rooms > 3 is encoded as +1.

f	g	h
-1	-1	-1
-1	-1	-1
+1	+1	-1
+1	+1	+1
+1	+1	-1
+1	+1	-1

$f(x)$: sign (area - 10)
$g(x)$: sign (Price - 6)
$h(x)$: sign (Price - 9)

Evaluating Learned Models : Test Data

- Learning algorithm uses training data $(x^1, y^1), \ldots, (x^n, y^n)$ to get model f.
- But evaluating the learned model must **not** be done on the training data itself.
- Use test data that is **not** in the training data for model evaluation.



Model Selection : Validation Data

- Learning algorithms just find the "best" model in the collection of models given by the human.
- How to find the right collection of models?
- This is called model selection, and it is done by using another subset of data called **validation data** that is distinct from train and test data.

$$\text{Price} = w_1 * (\#\text{rooms}) + w_2 (\text{area}) + w_3 (\text{distance}) + b$$

Lecture 5 : Unsupervised Learning: Dimensionality Reduction

Unsupervised Learning

- Unsupervised learning is 'understanding data'
- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Build models that compress, explain and group data.

(**Understanding**)

Unsupervised Learning Application

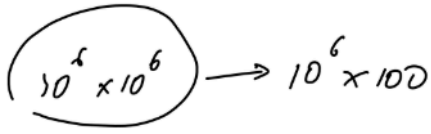
Tweet 1

⋮

Tweet 999999

Group the million tweets into 10 manageable groups

Dimensionality Reduction



E.g.: Represent a million gene expression levels of a million people, using just 100 numbers per person.

Dimensionality reduction: compression and simplification.

2.

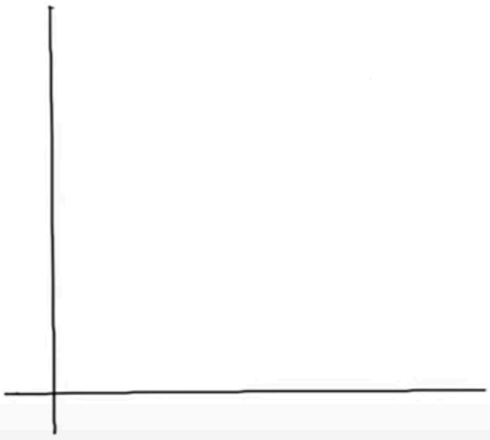
- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Encoder $f : \mathbb{R}^d \to \mathbb{R}^{d'}$
- Decoder $g : \mathbb{R}^{d'} \to \mathbb{R}^d$
- Goal : $g(f(\mathbf{x}^i)) \approx \mathbf{x}^i$
- Loss $= \frac{1}{n} \sum_{i=1}^{n} \|g(f(\mathbf{x}^i)) - \mathbf{x}^i\|^2$

$d' << d$

Dimensionality Reduction Illustration

$d = 2, d' = 1$

x	f	g
$[1, 0.8]$	0.2	$[0.2, 0.2]$
$[2, 2.2]$	-0.2	$[-0.2, -0.2]$
$[3, 3.2]$	-0.2	$[-0.2, -0.2]$
$[4, 3.8]$	0.2	$[0.2, 0.2]$



$f(x) = x_1 - x_2$

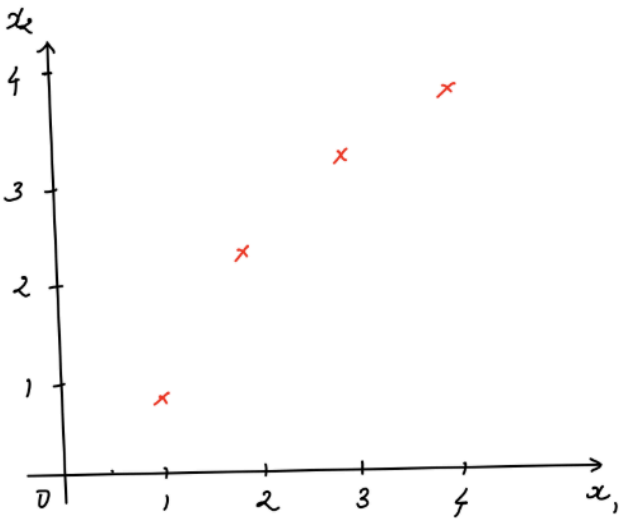
$g(u) = [u, u]$

$d = 2, d' = 1, n = 4$

	f	g
$[1, 0.8]$	0.2	$[0.2, 0.2]$
$[2, 2.2]$	-0.2	$[-0.2, -0.2]$
$[3, 3.2]$	-0.2	$[-0.2, -0.2]$
$[4, 3.8]$	0.2	$[0.2, 0.2]$

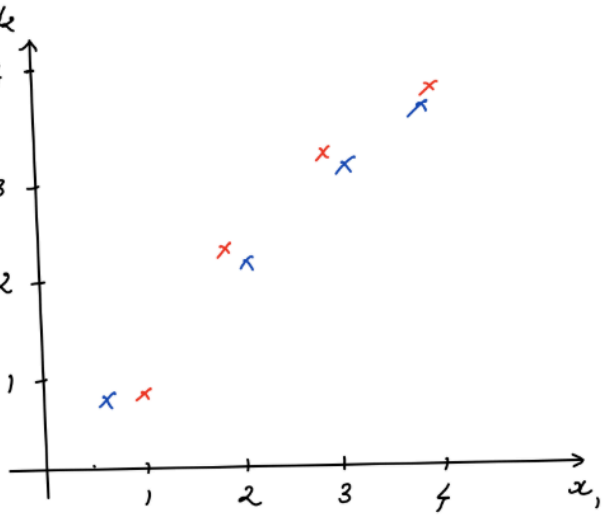
$f(x) = x_1 - x_2$

$g(u) = [u, u]$



$d = 2$, $d' = 1$, $n = 4$

	$\tilde{f}$	$\tilde{g}$
$[1, 0.8]$	0.9	$[0.9, 0.9]$
$[2, 2.2]$	2.1	$[2.1, 2.1]$
$[3, 3.2]$	3.1	$[3.1, 3.1]$
$[4, 3.8]$	3.9	$[3.9, 3.9]$



$f(x) = x_1 - x_2$

$g(u) = [u, u]$

$\tilde{f}(x) = \frac{x_1 + x_2}{2}$

$\tilde{g}(u) = [u, u]$

$10^6 \times 100$

$\hat{f}$, $\tilde{g}$

QN : 5

Lecture 6 : Unsupervised Learning: Density Estimation

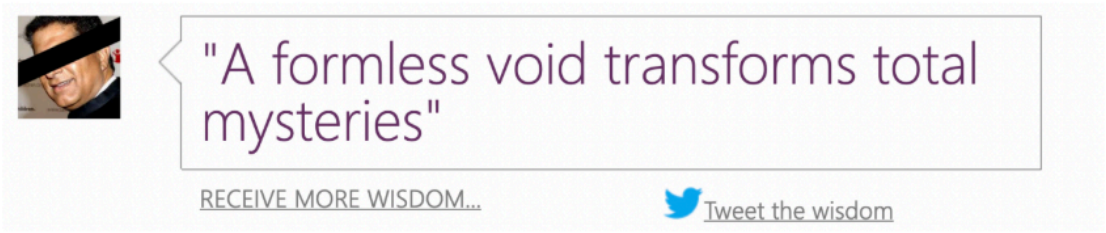
Density Estimation

E.g.: Assuming tweets from an account are independently generated randomly. Create a robot account that generates more such tweets.

$f(\text{Tweet})$ = Score of the tweet

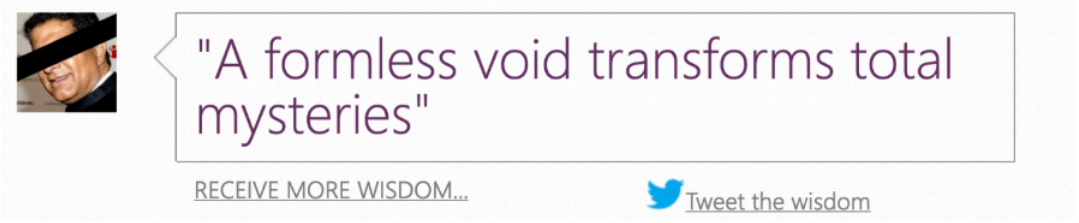
wisdomofchopra.com

It has been said by some that the thoughts and tweets of Mr. Chopra are indistinguishable from a set of profound sounding words put together in a random order, particularly the tweets tagged with "#cosmisconciousness". This site aims to test that claim! Each "quote" is generated from a list of words that can be found in Deepak Chopra's Twitter stream randomly stuck together in a sentence.



Disclaimer: This is intended for entertainment purposes only. It in no way reflects the thoughts of any real person.

Density Estimation



To generate such sentences randomly, we need to be able to assign a probability score to every possible 128 character sentence, giving high scores to those that are likely to be from the original source.

A density estimation model takes in several samples from a random source, and outputs a model that assigns a probability score to every possible instance.

Density Estimation

- Data: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$
- $\mathbf{x}^i \in \mathbb{R}^d$
- Probability mapping $P : \mathbb{R}^d \to \mathbb{R}_+$ that 'sums' to one.
- Goal : $P(\mathbf{x})$ is large if $\mathbf{x} \in$ Data, and low otherwise.
- Loss $= \frac{1}{n} \sum_{i=1}^{n} -\log\left(P(\mathbf{x}^i)\right)$

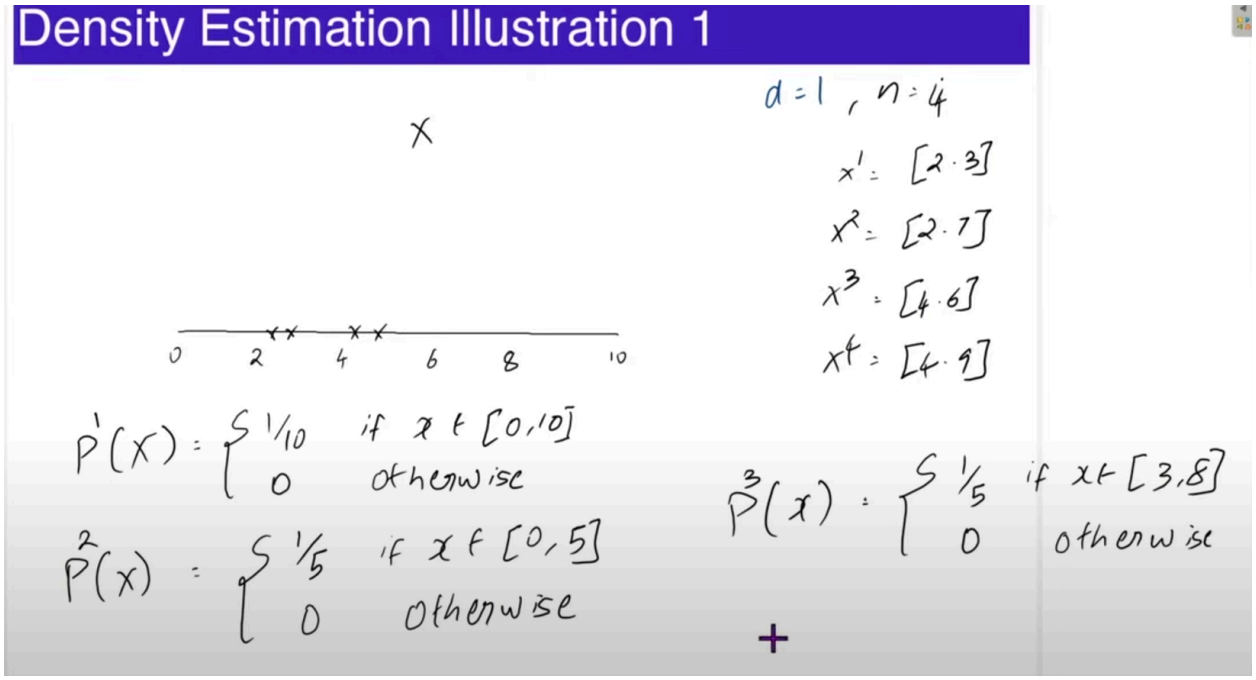
Σ Sum over all tweets $\{a..z\}^{128}$

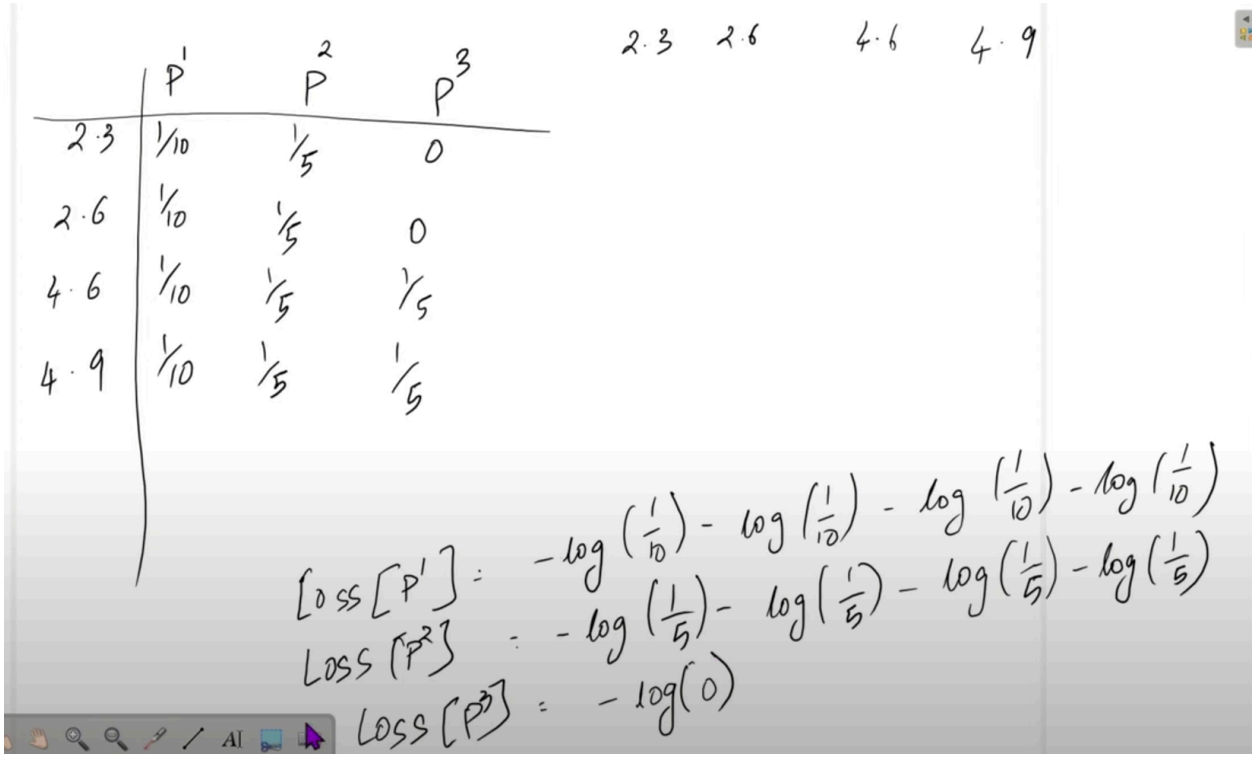
$P(\text{tweet}) = 1$

$P(x^i)$ is large.

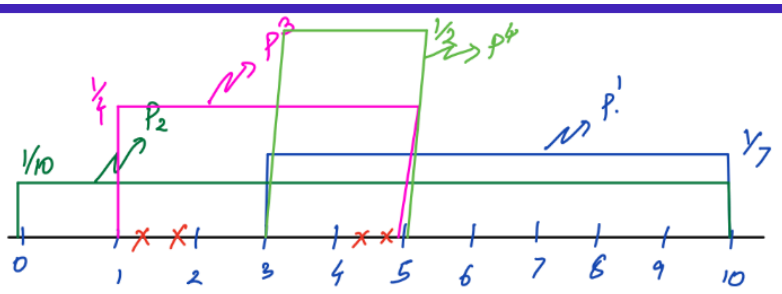
$P(\text{anything}) = 10^{10}$

Density Estimation Illustration 1





Density Estimation Illustration 2



$x^1 = [1.2]$

$x^2 = [1.9]$

$x^3 = [4.3]$

$x^4 = [4.8]$

p^1 : Uniform in $[3,10]$ $0, 0, \frac{1}{7}, \frac{1}{7}$

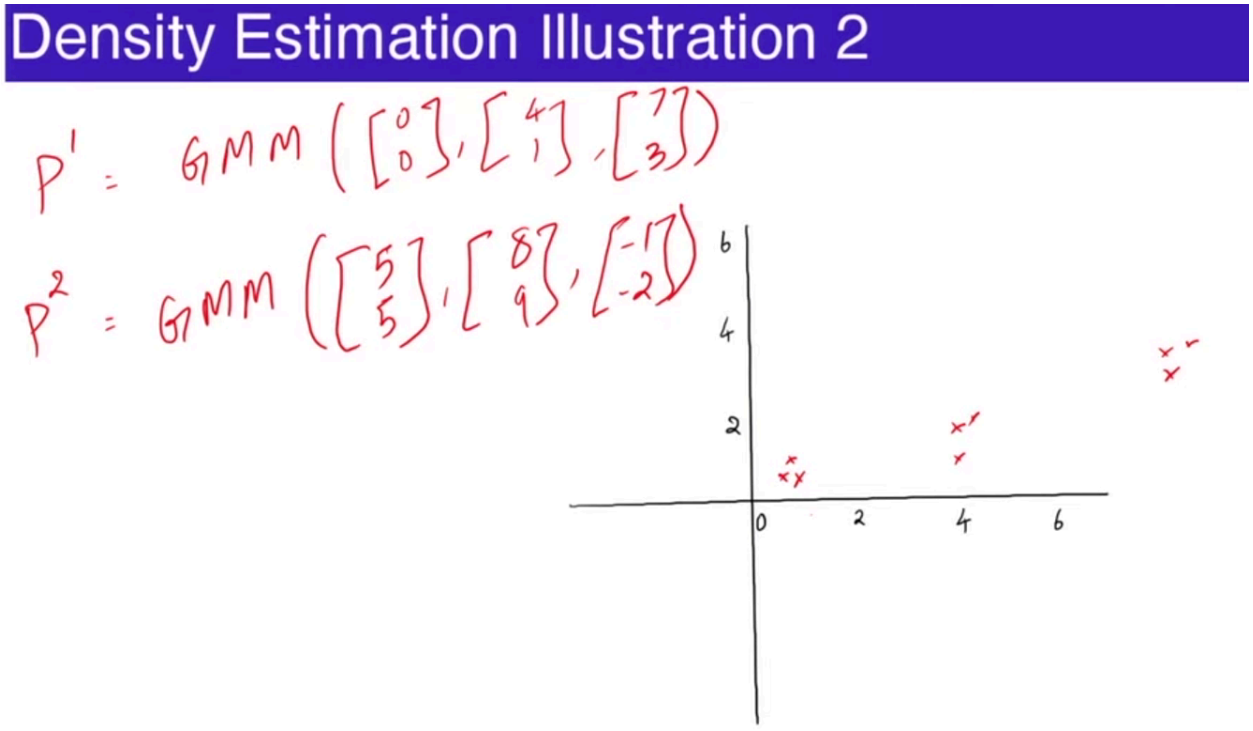
p^2 : Uniform in $[0,10]$ $\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}$

p^3 : Uniform in $[1,5]$ $\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$

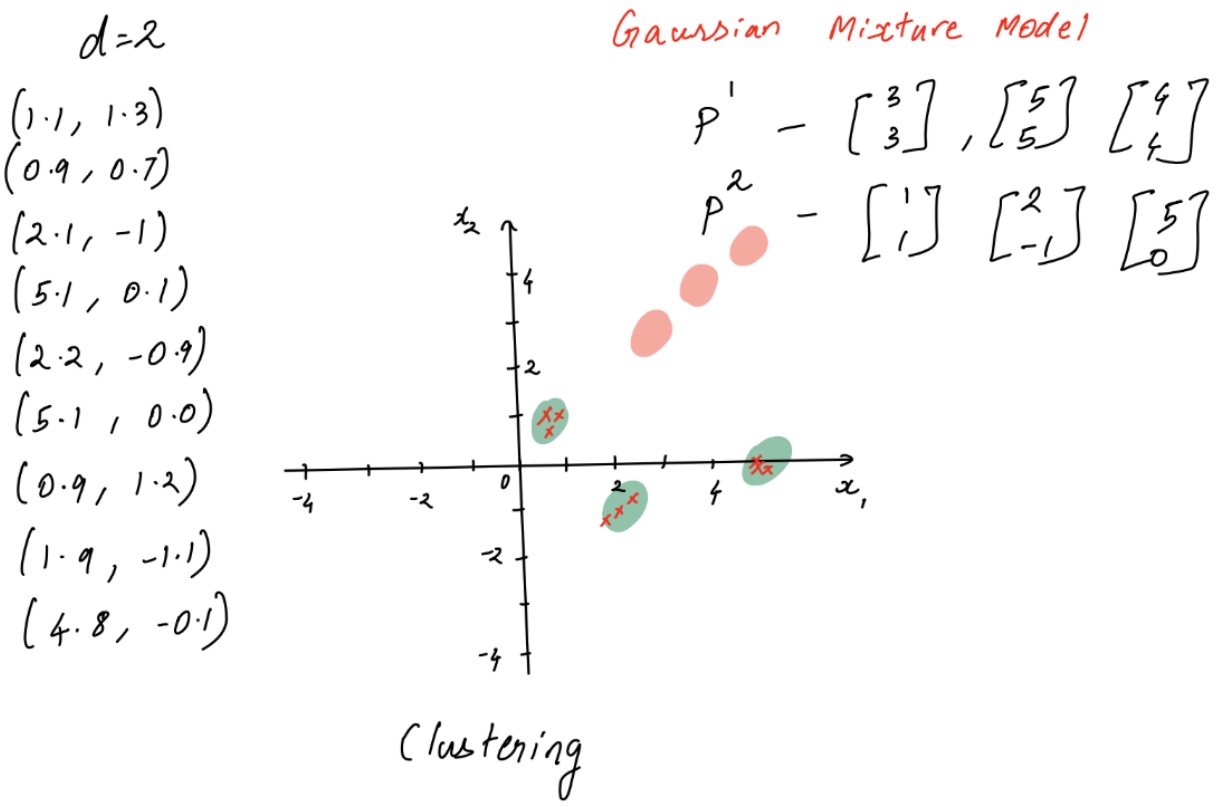
p^4 : Uniform in $[3,5]$ $0, 0, \frac{1}{2}, \frac{1}{2}$

$loss[p^4] = loss[p^1] = \infty > loss[p^2] > loss[p^3]$

Density Estimation Illustration 3



Density Estimation Illustration 4



What is a Gaussian Mixture Model (GMM)?

A **Gaussian Mixture Model** is a **probabilistic model** that assumes data points are generated from a **mixture of several Gaussian distributions (Normal distributions)** with unknown parameters.

- Each cluster is represented by a **Gaussian (bell curve)**.
- GMM is an extension of **K-Means clustering**, but it is **more flexible** because instead of hard assignments ("point belongs to cluster A"), it gives **probabilities** ("point belongs to cluster A with 70%, B with 30%").

QN : 4

Tutorial 1.1 : To ML or not to ML

Socratic Method

"I cannot teach anybody anything. I can only make them think."

Look at the data in the table below and **decide** whether a Machine Learning Approach is required to solve this problem.

Data	label
S1 = [3,5,4]	0
S2 = [3,4,5]	1
S3 = [4,2,1]	0
S4 = [6,7,8]	1
S5 = [1,2,3]	1
S6 = [1,1,1]	1
S7 = [1,2,0]	0

[4,5,3] ⟶ ?

Yes, It requires ML approach.

No, It doesn't require ML approach.

Show me more data points to make a fair decision

Not possible to make a decision

Secret Revealed

Consider a problem of identifying whether an integer sequence of length $N = 3$, say, $S = [3, 5, 4]$) is sorted in ascending order or not. Does this problem absolutely require machine learning approach to solve?.Justify

Obviously **Not** necessary, :-)

Justifications:

1.We know the rule and can transform it into code.
2. The Solution is also scalable

However, It could be cast as Machine Learning problem!

Let us pose it as a Machine Learning problem, then think about the data collection (generation) process

Data	label
S1 = [3,5,4]	0
S2 = [3,4,5]	1
S3 = [2,1,4]	0
S4 = [6,7,8]	1
S5 = [1,2,3]	1
S6 = [1,1,1]	1
S7 = [1,2,0]	0

$S_i = [S_1, S_2, S_3]$
$[0\text{-}9]$

Metadata associated with the dataset?

1.Number of data points: 1000

2.Element type: Non-negative Integers

3.Licence: CC

Model Type

Supervised Classification

Model:

Assumption:

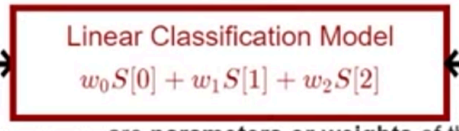
You have only the "data and corresponding labels", no other information.

1. Hand Crafted model:

```
if s[0] <= s[1]:
  if s[1] <= s[2]:
    print('1')
  else:
    print('0')
```

2. Machine learning model:

Data
S1 = [3,5,4]
S2 = [3,4,5]
S3 = [2,1,4]
S4 = [6,7,8]
S5 = [1,2,3]
S6 = [1,1,1]
S7 = [1,2,0]

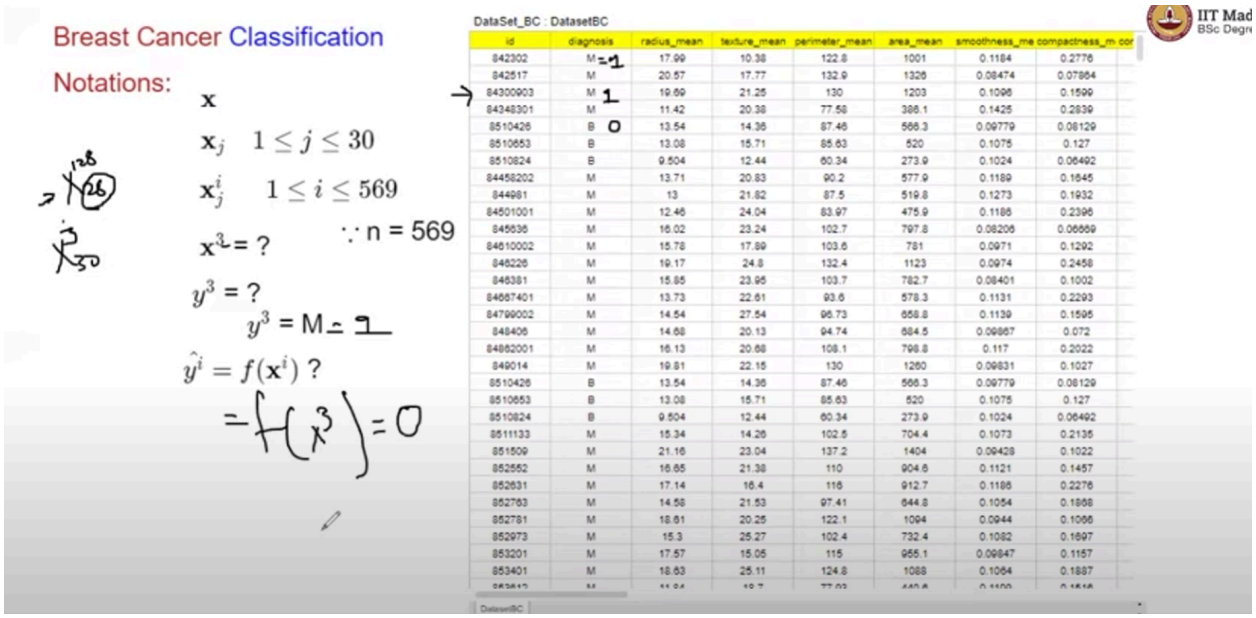


Label
0
1
0
1
1
1
0

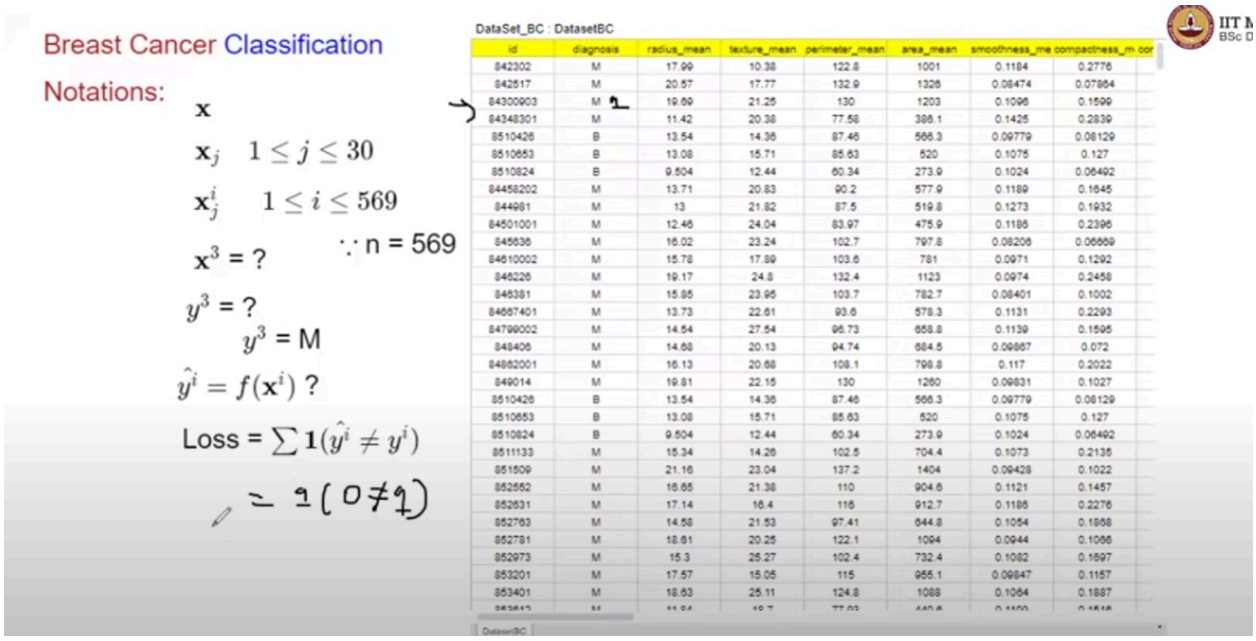
w_0, w_1, w_2 are **parameters or weights** of the model. The **best values** for the parameters will be learned from the data

Tutorial 1.2 : Illustration with a real world dataset

1.



2.



3.

Breast Cancer Classification

Notations:

$\mathbf{x}$

$\mathbf{x}_j \quad 1 \leq j \leq 30$

$\mathbf{x}_j^i \quad 1 \leq i \leq 569$

$\because$ n = 569

$\mathbf{x}^3$ = ?

y^3 = ?

y^3 = M

$\hat{y^i} = f(\mathbf{x}^i)$?

Loss = $\sum \mathbf{1}(\hat{y^i} \neq y^i)$

Which function is suitable?

Let us first consider **two features** to make the decision,

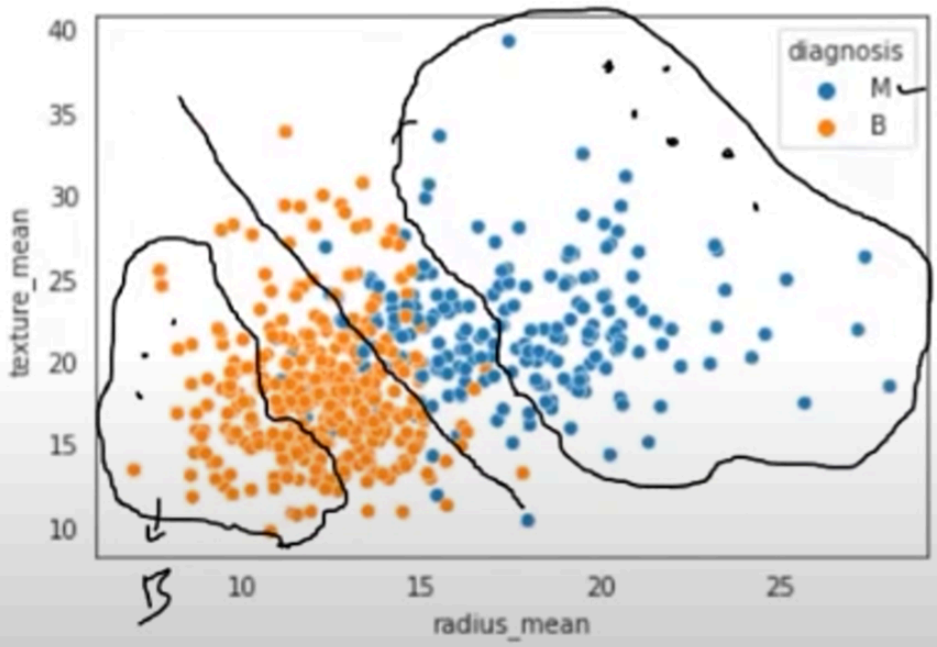
, therefore, $1 \leq j \leq 2$.

DataSet_BC : DatasetBC

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me	compactness_m	cor
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	
849014	M	19.81	22.15	130	1260	0.09831	0.1027	
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	
852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	
852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	
852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	
853201	M	17.57	15.05	115	955.1	0.09847	0.1157	
853401	M	18.63	25.11	124.8	1088	0.1064	0.1887	

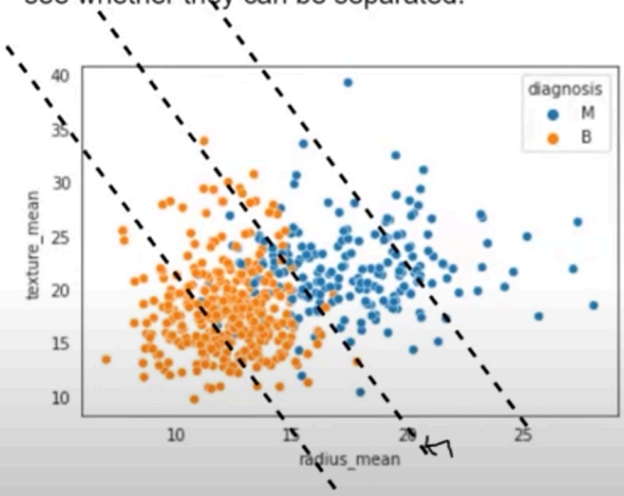
4.

Let us plot those two features as data points in the cartesian coordinates and see whether they can be separated.



5.

Let us plot those two features as data points in the cartesian coordinates and see whether they can be separated.



Shall we go with linear function for $f(\mathbf{x})$?

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 = \mathbf{w}^{\mathbf{T}}\mathbf{x}$$

How do we find the (optimal) values of w_0, w_1, w_2?

That's the Job of Learning algorithms.

6.

Train,Validation and Test Data

Train Set:

80% of total :455

Validation Set:

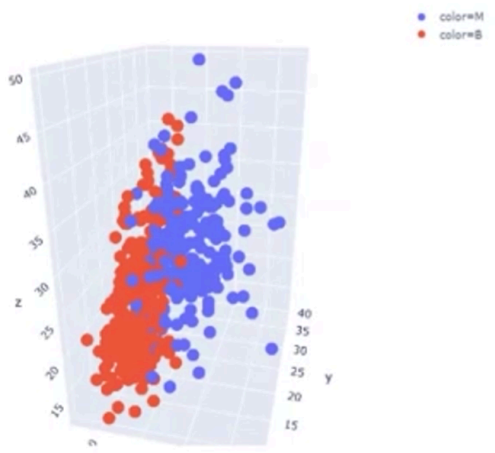
20% of training :91

Test Set:

20% of total: 204

7.

What if we add a third feature?



Could we still use linear function?

Of course Yes!

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 = \mathbf{w}^{\mathbf{T}}\mathbf{x}$$

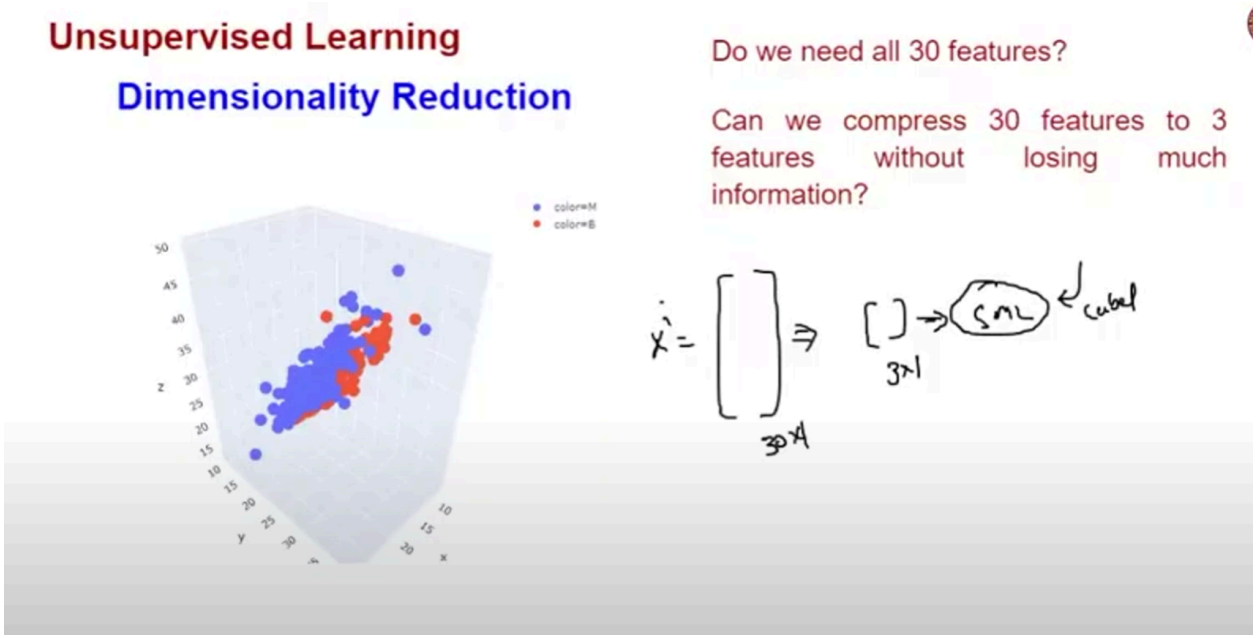
Obtain decent (?) performance.

What if we add a fourth feature?

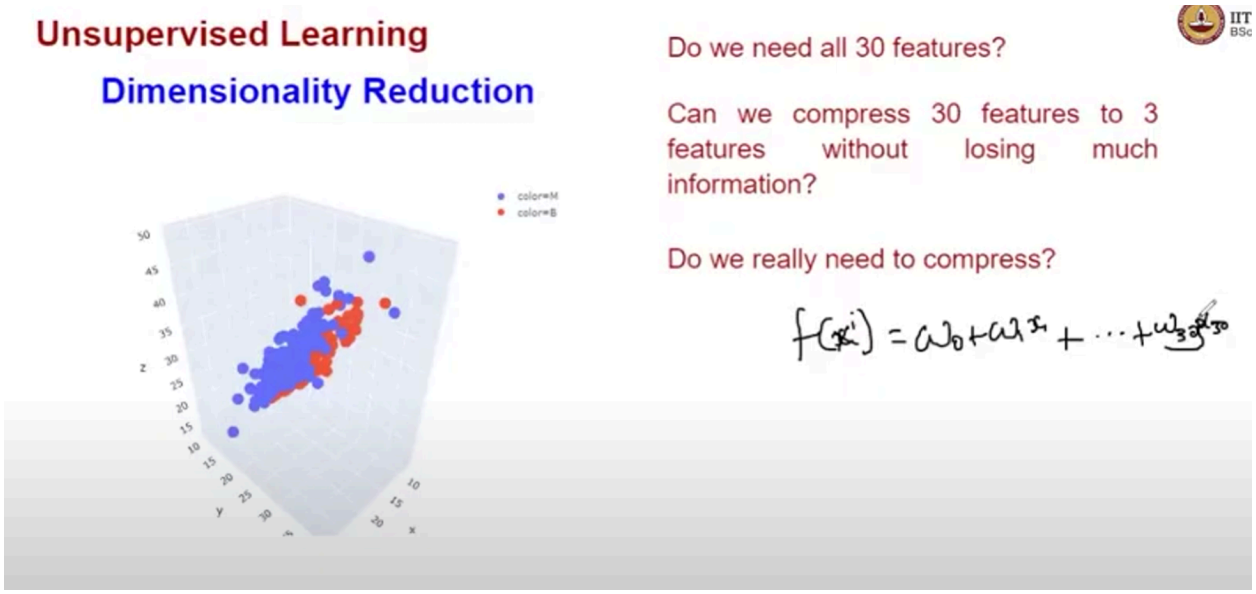
Do we need all 30 features?

Tutorial 1.3 : Dimensionality Reduction and Density Estimation with Applications

1.



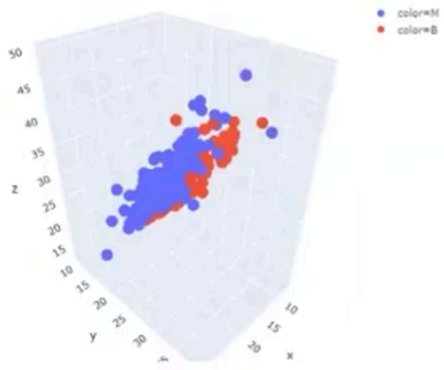
2.



3.

Unsupervised Learning

Dimensionality Reduction



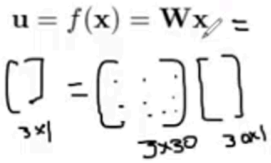
Do we need all 30 features?

Can we compress 30 features to 3 features without losing much information?

Do we really need to compress?

What might be the encoder function?

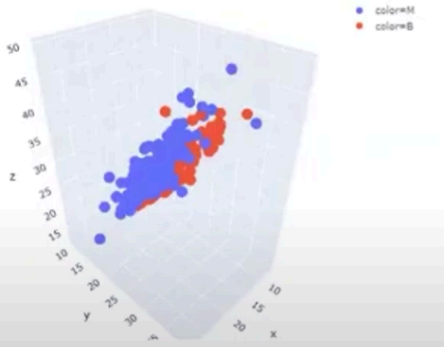
$$\mathbf{u} = f(\mathbf{x}) = \mathbf{W}\mathbf{x}$$



4.

Unsupervised Learning

Dimensionality Reduction



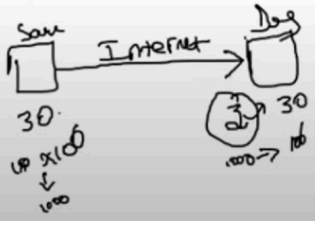
Do we need all 30 features?

Can we compress 30 features to 3 features without losing much information?

Do we really need to compress?

What might be the encoder function?

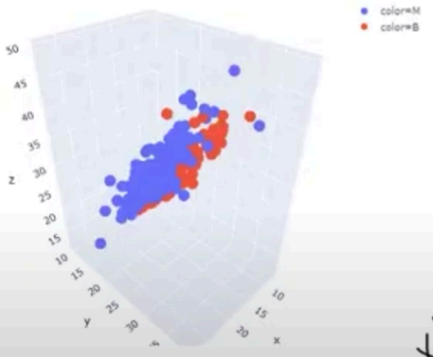
$$\mathbf{u} = f(\mathbf{x}) = \mathbf{W}\mathbf{x}$$



5.

Unsupervised Learning

Dimensionality Reduction



Do we need all 30 features?

Can we compress 30 features to 3 features without losing much information?

Do we really need to compress?

What might be the encoder function?

$$\underline{\mathbf{u}} = f(\underline{\mathbf{x}}) = \mathbf{W}\mathbf{x}$$

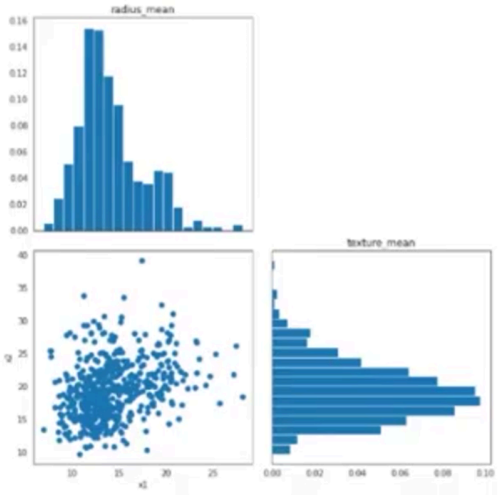
What might be the decoder function?

$$\mathbf{x} = g(\mathbf{u}) = \mathbf{W}^{\mathrm{T}}\mathbf{u}$$

6.

Unsupervised Learning

Density Estimation



Assumption:

All samples are assumed to be from some probability distribution.

We have totally ,n=569, samples.
However, it seems in sufficient.

Can we Increase the number of samples?

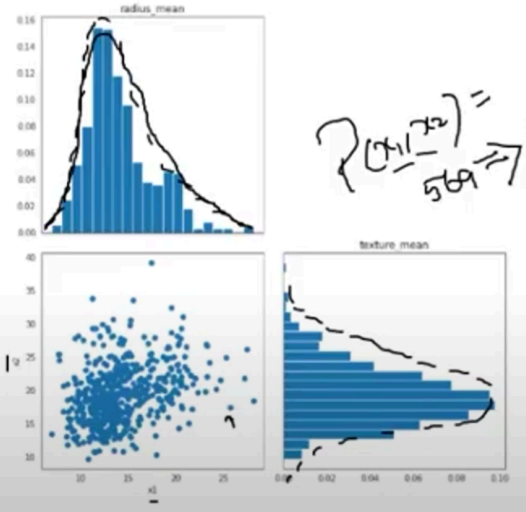
Or

Yes, take data from more cancer patients.

Generate the samples by estimating the PDF for each features.

7.

Unsupervised Learning
Density Estimation



Assumption:

All samples are assumed to be from some probability distribution.

We have totally ,n=569, samples.
However, it seems in sufficient.

Can we Increase the number of samples?

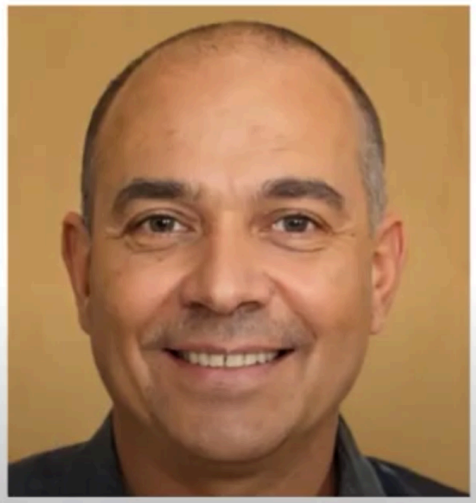
Or

Yes, take data from more cancer patients.

Generate the samples by estimating the PDF for each features.

8.

Do you Know him?



No one in the world knows him. Why?

https://thispersondoesnotexist.com/

Generated by the deep learning model StyleGAN (Generative Adversarial Networks)

Idea:

Estimate the parameter of PDF by using millions of face images and sample a new face from it!

PA : 1,3,4,5,7,12,13
GA : 2,6,9,11,13,14,15,18
In 13 check each and find smallest SSE

 # Note - Always Check Your Calculations