Day 2 : 19th December 2023

Group - 17, 21, 25, 29

Names :
Garvitaa Agarwal
Vaidika Padigela
Vishnu Priya (25) — Cohort 5
D. Greethika

Demystifying AI & ML:

- Day-to-day life: face recognition, autonomous cars, voice assistants, AI tools like chatgpt, etc.
- The margin of error is higher for a human, which might be the reason why AI can beat humans in games/calculations.

Eg:
For autonomous cars, it must know the destination location, obstacle detection. Based on the decisions it makes from the input, it can reach the destination.

It extracts features from the data and makes decisions from that.

Machine Learning has existed since the 1950's but it has only been used from 2010's. This is because more people have started to use electronic devices and more data has been collected for the ML model to train.
Another reason is that the computers have more processing speed which is actually capable of training the model.

Data is like the fuel for ML and it tries to understand the data.

Modern AI:
It is something executable unlike traditional AI, which can only be seen in papers.

Q) 1,3,5,7,9… What is the next number?
A) 11 (Odd Numbers)
    2x +1 -> {This is the objective function for the series}

Q) 1,3,9,19,33… What is the next number?
A) 51
    2(x)^2 + 1 -> {objective function}

The second question took more time than the first one as it is more complicated. How do we solve such problems?

- You have to find a pattern from the examples.
- Use it to predict the next number

When the solution is already set, we can solve it through a traditional code. But if there are multiple solutions and you wish to find the most optimised solution, ML can be used. Wherever, you cannot find a solution but you have some examples, you can use ML.

The objective is not to find a solution but to make an algorithm that can solve all questions in the given series.

Simplified View of ML:

Data → Algorithms →f(x)

<u>Algorithms</u> are nothing but mathematical procedures which can identify patterns and fins solutions.

Q) (1,3), (2,6), (3,9), (4,12).... What is the next point?
A) (5,15)

The above question can be solved using the graph y = 3x

But the problem in solving this question might be the <u>noise</u> in the data.

What makes it difficult?
- Noise in measurements
- Missing values
- Function is complex or uncertain

Most of these are due to human error. This makes it harder for the ML model to make an algorithm.

Q) Given a set of numbers{7, 26, 17, 11, 25, 32, 5, 8, 92}, partition into two sets
A) odd and even
Single and two digit,etc

The number of ways to partition a set of n distinct elements into subsets can be calculated using the **Bell Number**. The Bell Number represents the number of ways a set with n elements can be partitioned.
For a set of nine elements, there are 418 different ways to partition them. So, there are 418 different ways to segregate the set {7, 26, 17, 11, 25, 32, 5, 8, 92} into various subsets.

The above is Unsupervised Learning.

Two Prominent Learning Paradigms:
- **Unsupervised Learning:**
  Learn patterns from unlabelled data. Often look for structure.
  In this, there are no labels to segregate the data. We can only assume the ways to segregate them. But assumptions are not always accurate.

- **Supervised Learning:**
  It is the machine learning task of inferring a function from labelled training data.Segregate based on the given labels.

  There is another paradigm.

- **Semi - Supervised Learning:**
  It is a blend of supervised and unsupervised learning.
  Eg: google photos sees the labelled photo of a person and tries to label the rest of them the same.

<u>Data + interpretations(x, y) = (sample, label)</u>

Depending on the given data, we can decide if it is supervised or unsupervised.

More examples help the ML system devise an objective function accurately.

How to find the optimal parameters?
- Optimization problem
    1. Find the best coefficients for a given data/problem
- Training
- Computing

Traditional Programming:

$$\begin{tikzcd}
\text{Data} \arrow[dr] & & \\
& \text{computer} \arrow[r] & \text{Output} \\
\text{Program} \arrow[ur] & &
\end{tikzcd}$$

Machine Learning:

$$\begin{tikzcd}
\text{Data} \arrow[dr] & & \\
& \text{computer} \arrow[r] & \text{Program} \\
\text{input/output} \arrow[ur] & &
\end{tikzcd}$$

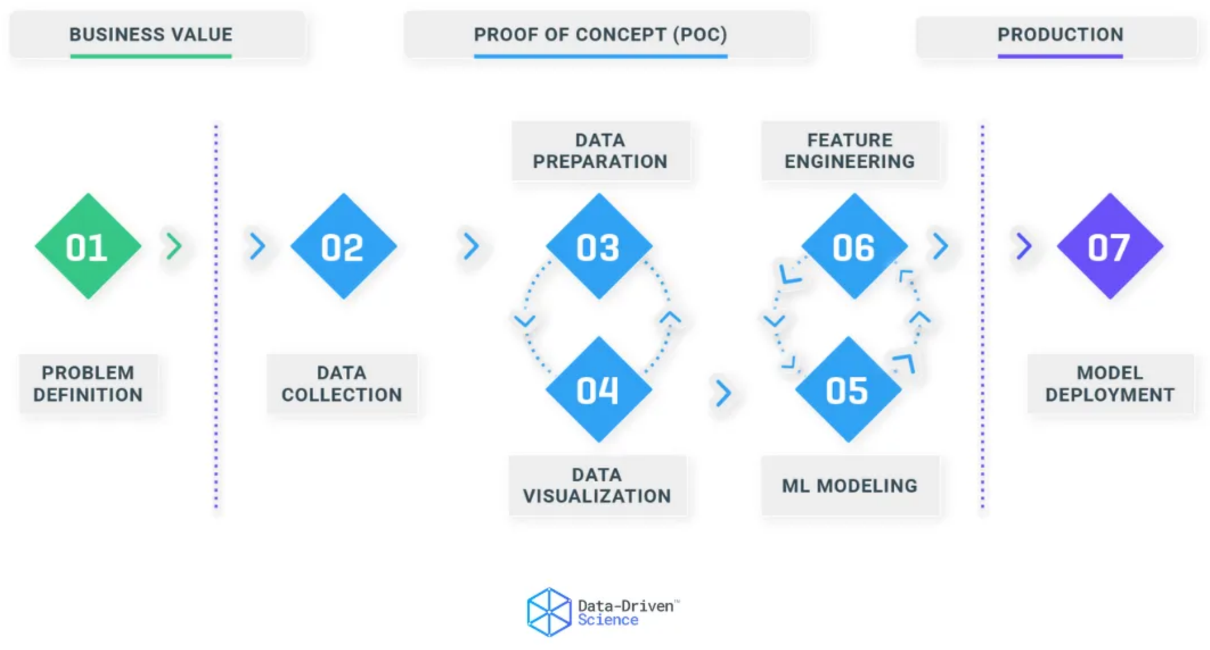
<u>ML Principles and Practical Issues:</u>

Characteristics of a good ML problem:
- Clear use case (What we are expecting the model to do)
- Relevant Data
- Decision Making

ML Pipeline:

1. Problem Definition
2. Data Collection
3. Data Preparation
4. Data Visualization
5. ML Modeling
6. Feature Engineering
7. Model Deployment

https://medium.com/@datadrivenscience/7-stages-of-machine-learning-a-framework-33d39065e2c9



Problem definition can be coming from the clients

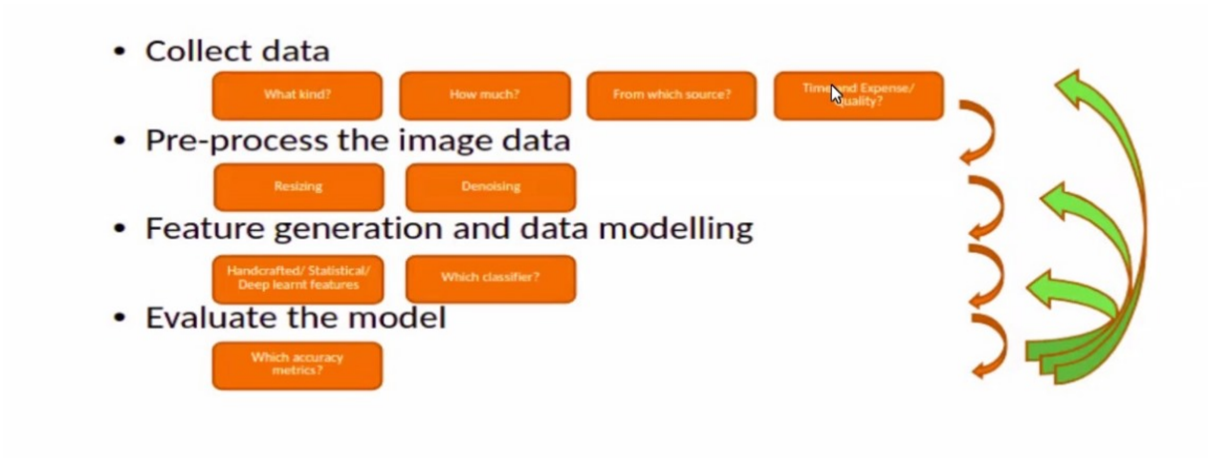
→ ML requires Domain understanding.

Once we understand the best model, the model has to be deployed. Deployment is the final step

Feature Engineering:
We can engineer new features from the existing models and create new ones.

For example, if we need to allow cats and not dogs we need to
Collect data(images), extract features,
y = f(x); y → predicted label
x → images of cats and dogs

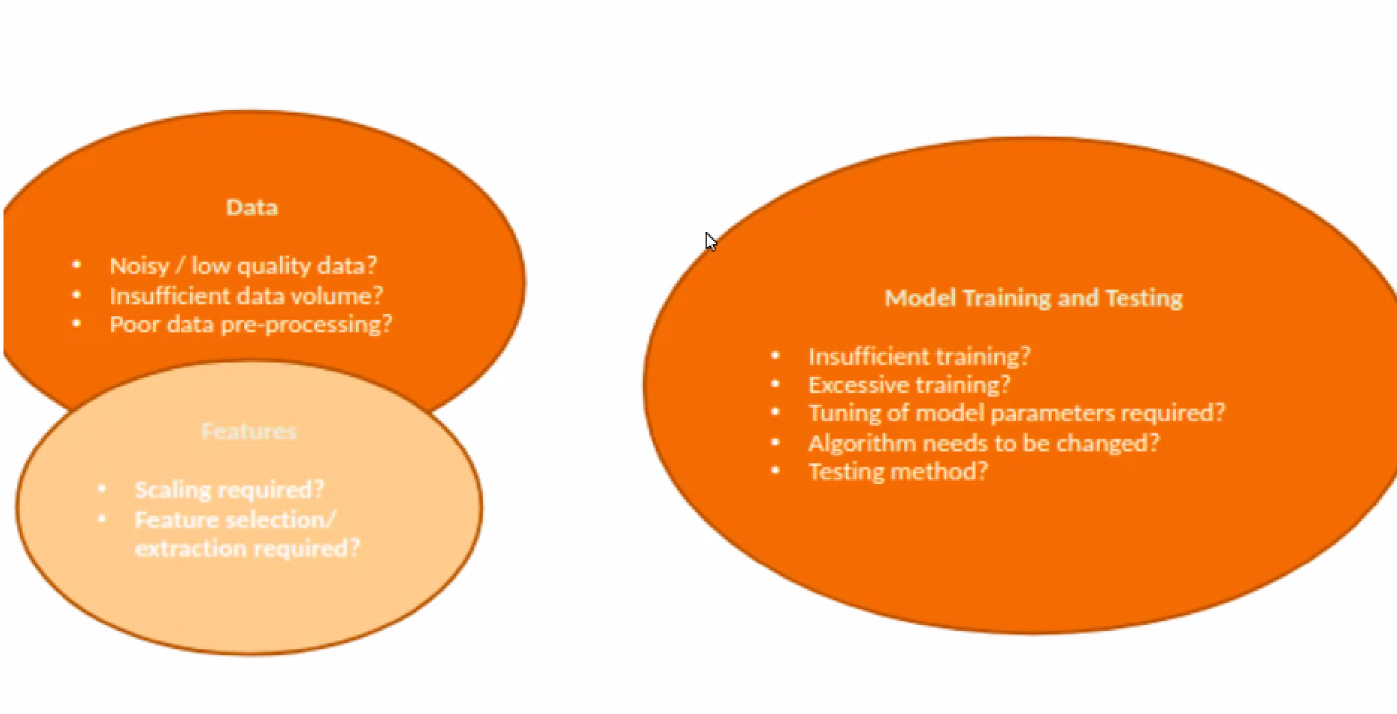
Build an ML Classifier:



- Collect data
  - What kind?
  - How much?
  - From which source?
  - Time and Expense/ Quality?
- Pre-process the image data
  - Resizing
  - Denoising
- Feature generation and data modelling
  - Handcrafted/ Statistical/ Deep learnt features
  - Which classifier?
- Evaluate the model
  - Which accuracy metrics?

- Collect data
    - What kind?
    - How much?
    - From which source?
    - Time & quality?
- Pre-process the image data (Because images will have noise)
    - Resizing
    - Denoising

- Feature generation and data modelling
    - Handcrafted/Statistical/Deep learnt features
    - Which classifier?
- Evaluate the model
    - What accuracy metrics?


Data
Noisy / low quality data?
Insufficient data volume?
Poor data pre-processing?
Features
Scaling required?
Feature selection/ extraction required?
Model Training and Testing
Insufficient training?
Excessive training?
Tuning of model parameters required?
Algorithm needs to be changed?
Testing method?

Q) What are the issues with model training and testing?
A) Over-fitting (Excessively trained, Model will try to memorise the data instead of evaluating it)

Randomly split the entire dataset into:

- Training Set
- Validation set
- Test Set

Training Set:
A dataset used for training the model

Validation Set :
Data used for fine tuning the model

Test Set:
Data only used for testing the model

Train the model in the training set and estimate model performance in the test set. The ratio of training and testing is usually 70:30.

If you provide irrelevant data to the model, the performance would go down. It will always give you an under-fitting model.

Eg: Training for cats and dogs and testing for lions and tigers.

Train, validate and repeat until the required performance is reached. Then test it.

Randomly split the entire dataset into:

- **Training set**: A dataset used for training the model. 60% to 80% of data.
- **Test set:** Data only used for testing the model. 20% to 30% of data.
- **Validation set:** 10% to 20% of data.

Drawbacks of this kind of data:

- Unfortunate split
- Small datasets require more training data

When you have a small amount of data, segregating them is quite difficult.

How to overcome this problem?
If the data has been correctly split, then the problem can be resolved.

Common Splitting Strategies:

- **K - Fold Cross Validation:**
  Multiple Splits: the more times we split the data and train, the more accurate the model is (reset and train, so aggregate and take average of all splits as the performance).

  This technique splits your data into 'K' groups or folds. The model is trained on K-1 folds and tested on the remaining fold. This process is repeated K times, each time using a different fold as a test set. The performance is averaged over the K trials.

- **Leave one Out Cross Validation:**
  It involves using a single observation as the test set and the remaining observations for training. This process is repeated for each observation in the dataset.

  A special case of K-fold where K equals the total number of instances in your dataset. Each instance is used as a separate test set exactly once while all remaining instances form the training set.

- **Stratified K-Fold Cross Validation:**
  Similar to K-Fold, but this method preserves the proportions of classes within each fold, which is beneficial if you have imbalanced classes.

  Mostly **K - Fold Cross Validation** is used

Choosing the Right Strategy:

- **Use LOO if:**
  Your dataset is small and you want the most accurate estimate of the model's performance. However, it's computationally expensive for large datasets. LOOCV is suitable for small datasets where the number of samples is very limited.

- **Use K-fold if:**
  You want a good balance between accuracy and computational cost. It's more efficient than LOO for large datasets and provides a less variable performance estimate. K-fold cross-validation is commonly used when you have a moderate-sized dataset.

Feature	**K-Fold Cross-Validation**	**Leave-One-Out Cross-Validation**
Splitting size	k roughly equal folds	1 data point for testing, n-1 for training
Iterations	k times	n times (number of data points)
Performance estimation	Average of k results	Average of n results
Computational cost	Lower	Higher
Variance of performance estimate	Higher	Lower (closer to true performance)

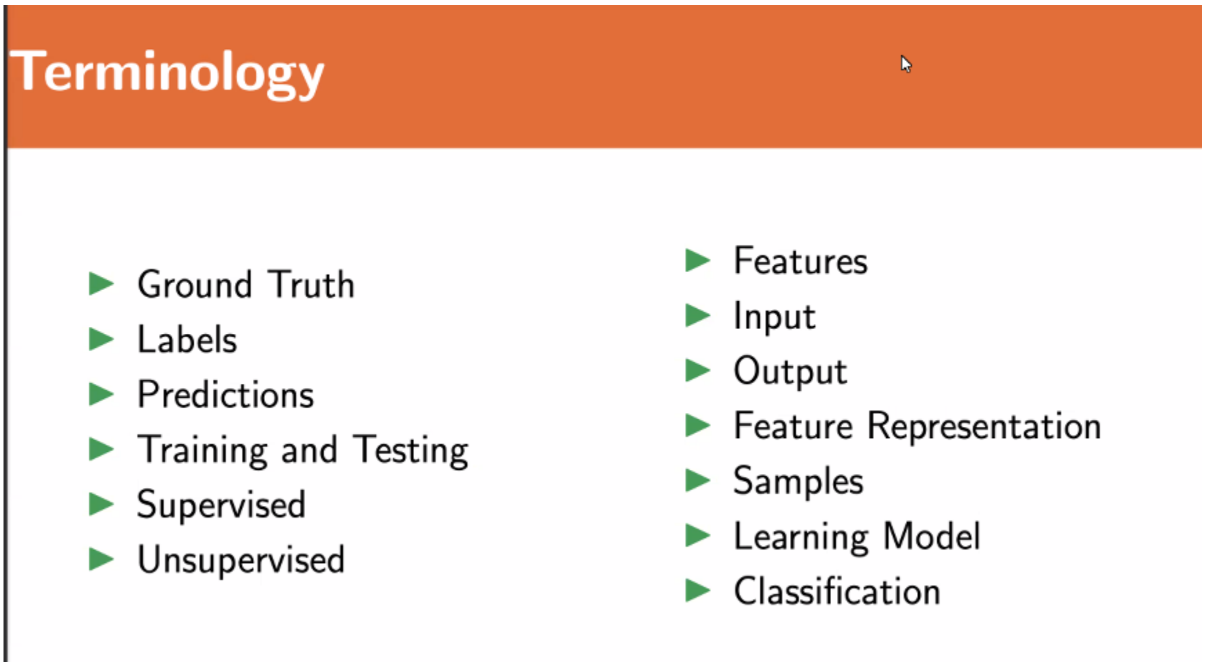
Broad Classification of Machine Learning Systems:

Reinforcement Learning:
No samples/data is provided. Feedback/reward policy system. Works in the same way a human learns how to speak/walk.
We look at the different scenarios and assume what to do and what to not. Pleasure and pain. In a given situation, it will make a decision based on the current data it has, and if it is wrong, it will try to understand which one will be wrong.

Feedback is important for reinforcement learning.
The algorithm in instagram also uses the same for suggesting reels and posts in the for-you page.

Eg:
A baby has hot water and normal water in front of it. If it does choose hot water, in the future it will make sure it does not choose hot water. It can be remembered by the vapour on the hot water, or the bubbles.

Terminology Used in ML:



Terminology

- Ground Truth
- Labels
- Predictions
- Training and Testing
- Supervised
- Unsupervised
- Features
- Input
- Output
- Feature Representation
- Samples
- Learning Model
- Classification

Ground Truth: The actual, true, or correct values or outcomes in a dataset. It serves as a reference for evaluating the performance of a model

The "ground truth" is like the cheat sheet or correct answers that help you check if the computer is doing a good job. If the computer says a picture is a cat, but your ground truth says it's a dog, then the computer needs more practice.

Labels: Labels are the known outputs or categories assigned to each data point in a supervised learning dataset. In classification, for example, labels represent the classes or categories.

In simpler terms, labels help the computer learn and remember what different things look like. It's like teaching a friend the names of different animals.The labels are your way of helping the computer understand and recognize things in the pictures.

Predictions: The outputs or values generated by a model based on input data. Predictions are compared to the ground truth to assess the model's performance.

Training and Testing:
Training: The phase where a machine learning model learns from a labelled dataset, adjusting its parameters to minimise the difference between its predictions and the ground truth.
Testing: The phase where the model is evaluated on new, unseen data to assess its ability to generalise beyond the training set.

Supervised: A type of machine learning where the algorithm is trained on a labelled dataset, learning from input-output pairs. The goal is to make predictions on new, unseen data.

Unsupervised: A type of machine learning where the algorithm explores patterns, structures, or relationships within unlabeled data. There are no predefined output labels.

Features: Features are the input variables or attributes that describe each data point in a dataset. Features are used by machine learning models to make predictions.

Input: The data or information provided to a machine learning model as it processes and makes predictions. Inputs consist of features that describe the characteristics of the data.

Output: The result or prediction generated by a machine learning model based on input data. In supervised learning, this is compared to the ground truth to evaluate performance.

Feature Representation: The way in which features are transformed or encoded to be suitable for input into a machine learning model. Feature representation is crucial for effective model learning.

Samples: Individual data points or instances in a dataset. Each sample is characterised by a set of features and, in supervised learning, an associated label.

Learning Model: A mathematical or computational representation that captures the relationship between input features and output predictions. The model is trained on data to make predictions on new, unseen data.

Classification:A type of supervised learning task where the goal is to assign input data to predefined categories or classes. The output is a categorical label representing the class membership.

Which learning to follow, depends on what we're trying to predict
<u>Supervised:</u>
- Classification - finitely possible outcomes (even if 1000, the outcome is out of these only)
- Regression - infinitely possible outcomes on numerical data (continuous data)

<u>Unsupervised:</u>
- Clustering - Group the data based on the similarity of the data
- Associations - Identify how on product is associated with another
  Eg: toothbrush and toothpaste

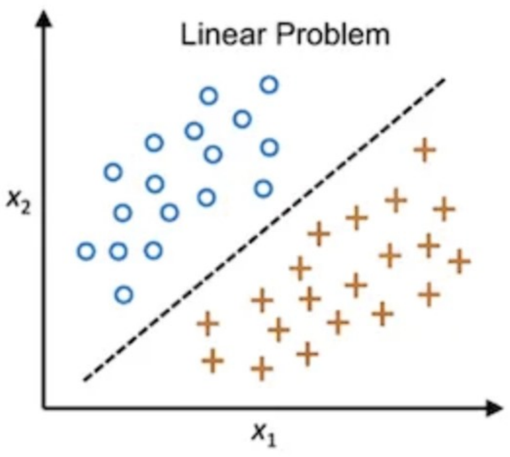
Classification Algorithm Method - Linear Classifier:

A linear classifier in supervised learning is a method for classifying data into different categories. It uses a linear equation to separate different classes in a dataset. This linear equation can be a straight line in 2D, a plane in 3D, or a hyperplane in higher dimensions.

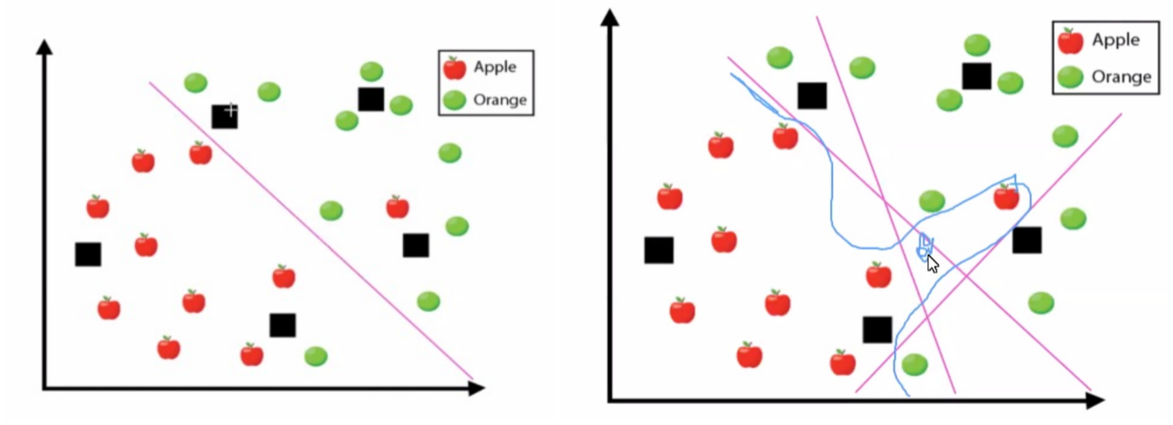
Linear Classifier:

- Recall the abstraction:
    - y = f(ω, x)

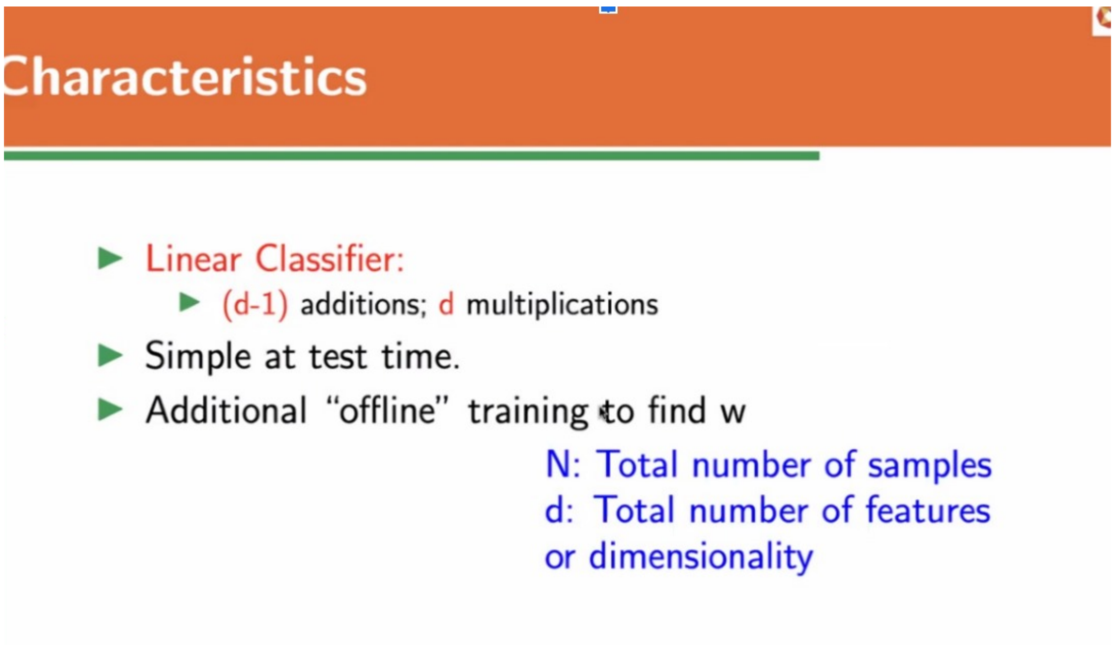
- f = Σωx = W ^ T . X



We cannot make the model guess all the cases correctly. There will always be noise. Hence the reason why we can sometimes leave some data points even if they do not fit into the line



It is important to make sure that testing data has less error as compared to the training data.



Characteristics

- Linear Classifier:
  - (d-1) additions; d multiplications
- Simple at test time.
- Additional "offline" training to find w

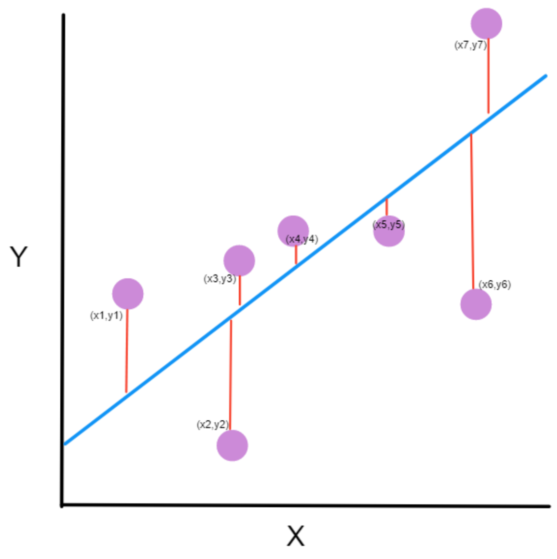
N: Total number of samples
d: Total number of features or dimensionality

Neural Networks:
Neural structures are inspired heavily from human brains.

Linear Regression:

Used when the data is almost linear.

Find the best fit line. The line which is having the least error and has least deviation from most data points.
Linear regression takes most data points and checks if the line is closest to most data points.



<u>**Cost function**</u>: Error/Deviation from actual values to output.

Cost Function (also known as loss function) quantifies the difference between the predicted values from the model and the actual observed values in the dataset. The goal in linear regression is to minimise the cost function.

Cost function = 1/2 * ∑ (ya - yb)²

Linear regression uses a specific cost function called **Sum of squared error**.

Our objective is to minimise the cost function

Cost Function

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}[\underset{\substack{\uparrow\\ \text{Predicted Value}}}{h_\Theta(x_i)} - \underset{\substack{\uparrow\\ \text{True Value}}}{y_i}]^2$$

Gradient Descent

$$\boxed{\Theta_j = \Theta_j - \underset{\substack{\uparrow\\ \text{Learning Rate}}}{\alpha}\frac{\partial}{\partial\Theta_j}J(\Theta_0, \Theta_1)}$$

Gradient Descent:
First create a random line, find cost function, update weights to minimise cost function using gradient descent. This is done until cost function cannot be reduced any further, that's the best fit line.

NOTE:
ML cannot provide insights about data not there in the dataset. If the data itself is biased, then nothing can be done.
Non-numeric data can be converted to numeric in a specific way.

Q) Implement the linear regression
Scikit Learn library for linear regression
Numpy is numerical computational library
Pandas is used for data manipulation/process
Pyplot is a library for graphs
A)

Nominal Data:
Does not have specific order

Ordinal Data:

Has a specific order.
Functions:
Y = mx +c
In ML, y and x are constants
        m and c are variables
m, c = weights
This is because x & y are fixed and known.
The problem is to choose the right weights such that the error is minimum.

Error function : measure of how good an approximation is

Q)Write a program to find the square root of a number.
The Newton-Raphson Iteration
 Steps - guess a random number as the square root
Divide the given number by the guessed number
If the quotient is equal to the guessed number then it is the square root
Otherwise the square root lies between the quotient and the number chosen
Take the middle of quotient and the chosen number
Divide the given number by it
If the quotient is equal to the number then it is the square root.

```
#code to find the square root of a number
def findsqrt(num):
 sqrt = 10
 while(sqrt * sqrt != num):
   quotient = num / sqrt
   sqrt = (sqrt + quotient) / 2
 return sqrt
n = findsqrt(49)
print(n)
```

10 is the guess and we can change it to anything