WHAT IS MACHINE LEARNING

It is a science of getting computers to learn without being explicitly programmed

ARTIFICIAL GENERAL INTELLIGENCE: a theoretical type of artificial intelligence (AI) that aims to create machines with human-like intelligence.

TYPES OF ML:

Supervised learning

Unsupervised learning

SUPERVISED LEARNING**:**

Learn to predict x--->y or input to output

we give our machine the correct examples to learn from i.e. the correct label y for input x
the algorithm learns maps the input data to the desired output

it learns from giving the "right answers"

some real-world supervised learning scenarios are:

Spam Filtering (Binary Classification)

Input: Email content

Output**:** Spam? (0 = Not spam, 1 = Spam)

Application**:** Detecting spam emails.

Speech Recognition (Sequence Prediction)

Input: Audio data

Output: Text transcript

Application: Converting spoken language into written text.

Machine Translation (Text-to-Text Mapping)

Input**:** English text

Output: Spanish translation

Application: Translating languages using AI.

REGRESSION:

Regression is actually a type of supervised learning algorithm. Regression specifically refers to the task of predicting a continuous output variable based on

Input features-

1.choosing a perfect line which will fit to the data.

2.imp part of the algorithm is to predict right answers

3.task is to predict number

CLASSIFICATION ALGORITHM:

-trying to predict only small number of possible outputs(mostly two possible outputs)

-they predict categories e.g. picking a picture is a cat or dog

-predicts small, finite ,limited set possible outcomes such as 0,1,2 but not numbers 1.2,0.6

- We can have two inputs and then create a boundary to differentiate between the outputs

UNSUPERVISED LEARNING:

1. We are not given any labels or predefined outcomes.
2. we are given data that isn't associated with any output labels y
3. but algorithm finds a structure a pattern
4. we ask the algorithm to find out what patterns and structures that might be in the data
5. data comes with input x but not output labels y

CLUSTERING: Grouping data points into clusters based on similarity by taking data with no labels
e.g. finding the similar news articles that mentions similar words and grouping them into clusters
the clustering algorithm figures out on its own which words suggest same articles in the group
This clustering algorithm must find the clusters on its own.
e.g. big companies have huge databases of customer information using this data they automatically group customers into different market segments so that they can serve the customers better

LINEAR REGRESSION:
It belongs to the family of regression algorithms, which predict a continuous output based on given input features
Filling a straight line to the data
e.g. to predict the size of the house with the price of the house
Linear regression models the relationship between two variables by fitting a linear equation to the observed data. The idea is to find the best-fitting straight line (the regression line) through the points.

- it is a type of supervised learning model because we are first training our model by giving the model data that has the right answers . we give the model examples of both size of the house and price that model should predict for each house where here the prices i.e. the right answers are given for each dataset
- it's called a regression model because it predicts numbers like prices and sizes as the outputs (any supervised learning that predicts a number is adjacent to a regression problem linear regressions is one example of adjacent regression model)
- there infinitely many numbers a model could output

The basic formula for a linear regression model is:

$$y=\beta0+\beta1x$$

- y is the predicted output.
- x is the input feature.
- β0is the y-intercept (the value of y when x=0).
- β1is the slope of the line (the change in y for a one-unit change in x).

training set --

the dataset used to train a model is called a training set, we train the model so that they can predict accurately

it consists of input features and targets that is the output

- after training our learning algorithm produces a function f (hypothesis)
  it takes an input and gives an output y^(i.e. the estimate or prediction of y)
  the function f is called a model
- the models prediction y^ is the estimate value of y it may or may not be the true value
  here $y=\beta0+\beta1x$
- Linear regression with one variable (i.e., linear regression with a single input variable xx) is also known as univariate linear regression
- idea of a cost function is one of the most important and universal in ml

GRADIENT DESCENT:

-The systematic way of finding minimum values of J(w,b) of the cost function

- it is an algorithm we can use to minimize any function not just cost function of linear regression

-they can work with other cost functions that have more than two cost functions

- to find minimum J(w,b):we keep on changing w,b to reduce J(w,b) so that it reduces to minimum

1. **Gradient Descent Update Equations:**
   - The formula for updating the parameters is: w=w−α∂J(w,b)\ ∂w
   - Here:
     - α is the **learning rate**, which controls the step size.
     - ∂J\∂w and ∂J\∂bare the gradients (partial derivatives) of the cost function with respect to w and b

2. **Simultaneous Update vs. Sequential Update:**

Correct: Simultaneous Update

Both wand b are updated **simultaneously** without affecting each other's calculations

This ensures the updates are made using the original values of w and b.

tmp_w = w - α ∂J/∂w

tmp_b = b - α ∂J/∂b

w = tmp_w

b = tmp_b

Incorrect: Sequential Update

Here, the update of w occurs first, and then b is updated using the new value of w:

The problem with this approach is that it uses the updated value of www, leading to incorrect calculations and slower or unstable convergence.

tmp_w = w - α ∂J/∂w

w = tmp_w

tmp_b = b - α ∂J/∂b (w)

b = tmp_b

COST FUNCTION:

Tells us how well the model predictions match the actual outcomes so that we can try to make it better

Parameters of a model can be adjusted during training

We minimize the cost function during the training process, leading to more accurate models

Mean Squared Error (MSE):

J=1/n∑ (y−y^)2

It calculates the average squared difference between the actual and predicted values.

Mean Absolute Error (MAE):

MAE=1/n∑ |y−y^|

It calculates the average absolute difference between actual and predicted values.

The algorithm repeats the parameter update steps until convergence, meaning the cost function reaches a minimum or changes minimally across iterations.

-if the learning rate(alpha) is too small the gradient descent will work but it will be slow

- if the learning rate is too large then gradient descent may overshoot and never reach the minimum and may fail to converge

LOGISTIC REGRESSION:

Is similar to linear regression except it predicts whether something is true or false instead of predicting something continuous

Logistic Regression is a statistical method used for binary classification problems—situations where the outcome is either one thing or another (yes/no, true/false, 1/0). Imagine you're trying to foresee if someone's favourite colour is blue. Logistic Regression helps model this probability based on various input features.

At its core, Logistic Regression models the probability of a certain class or event. Unlike Linear Regression which predicts continuous values, Logistic Regression outputs a value between 0 and 1 by applying a logistic (sigmoid) function to a linear combination of input features.

Logistic regression can work with continuous data and discrete data but unlike normal regression we can't easily compare the complicated model to simple model

The Logistic Function

This function turns any value between -∞ and +∞ into a value between 0 and 1.

$$\text{Logistic}(z) = \frac{1}{1 + e^{-z}}$$

Where z is a linear combination of input features.

Odds and Log Odds

Logistic Regression works with odds, where:

$$\text{Odds} = \frac{p}{1 - p}$$

Training a Logistic Regression Model

To teach our model, we need a labelled dataset and an optimization algorithm called **gradient descent**.

1. **Collect Data**: Gather input data (features) and corresponding output (labels).
2. **Initialize Parameters**: Start with initial guesses for weights and biases.
3. **Compute Predictions**: Use current parameters to make predictions.
4. **Calculate Error**: Assess how far off predictions are using a **cost function**—typically **Cross-Entropy Loss** for classification problems.

5. **Update Parameters**: Adjust weights and biases using gradient descent to minimize the error.
6. **Iterate**: Repeat steps 3-5 until the model's predictions become consistently accurate.