श्रवण,मनन,निदिध्यासन

# Artificial Intelligence


Vaibhav Zodge

7020616260

info.dnyanyog@gmail.com

https://www.dnyanyog.org

https://github.com/zodgevaibhav

# Artificial Intelligence

*"Artificial Intelligence a techniques which enable machines to mimic human behavior"*

## Why ??

**Cost Reduction** 💰 **Speed & Efficiency** ⚡ **Scalability** 📈 **Accuracy & Consistency** 🎯 **Handling Complex Big Data** 🔍

# Why Human Behavior ?

- Think 🧠
- Learn 📚
- Store Information 💾
- Analyse 🔍
- Decide ✅
- Predict 🔮
- Take Actions ⚡
- Relate ❤️

We want to reduce human efforts : Software does the same job

Machine should be able to **further reduce human works or replace human at some kind of work**

Humans think and decide on answers, solve the problems, draw pictures

Humans keep learning from experiences, from Rewards or from punishments

Humans are good at relating the things, guess work on the basis of past experience or knowledge

Humans takes salary :)

# What Learning AI Means ?

*"Artificial Intelligence a techniques which enable machines to mimic human behavior"*

## Model Creation/Training

### Machine Learning

Regression  Clustering

Classification  Dimensionality Reduction

### Deep Learning

ANN  RNN (Time series like data)

CNN (Image, grid like data)  Transformer (NLP, Language Understanding, Image Understanding)

**Model Use**

RAG, RAG Pipeline, Agentic AI, Langchain, Embedding etc…

**Syllabus :**
https://github.com/zodgevaibhav/gen-ai-learning?tab=readme-ov-file

# AL/ML/DL/GenAI

**Artificial Intelligence**

A techniques which enable machines to mimic human behavior

We want machine to speak, listen, think, act like machine

AI Can be possible without Machine Learning

Ex. IVR, Text to Speech, Old Chatbots (Alexa)

**Machine Learning**

A code which make AI possible called ML.

It is subset of AI which use statistical methods to enable machines to improve the experience

ML is also considered as extension to statistic math.

**Algorithm** : Regression, Classification, etc…

It required programming, logic, etc…

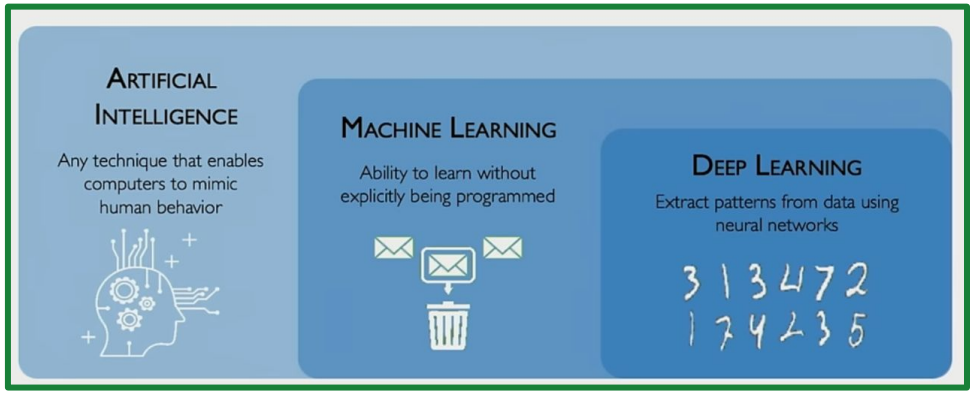
**Deep Learning**

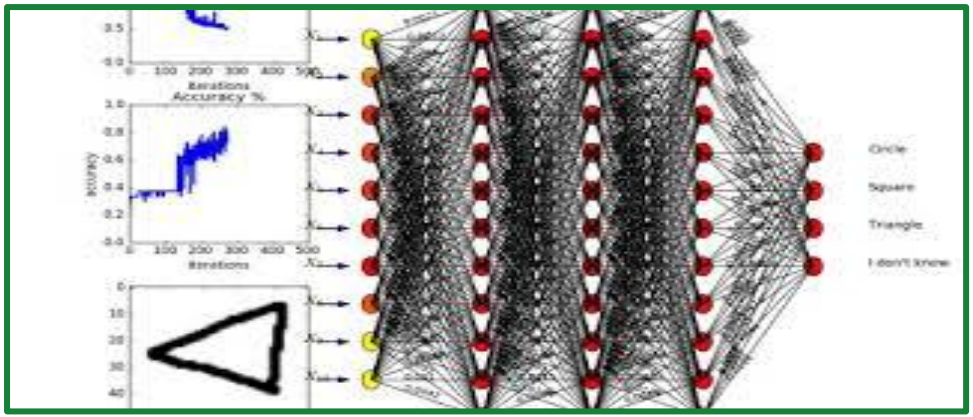
Based on human brain which have billions of neuron

Help to do computation in Machine Learning feasible

It is subset of ML which makes computation of **multi-layer neural network feasible**

Human brain have neural n/w hence scientist implemented similar architectural mode artificial neural network for computation. **ANN**




https://youtube.com/shorts/fQ376G-Ek1E?si=HbX4WRKbzJCHV_UR

# Know GenAI (Generative AI)

**Gen AI**

AI which can use deep learning and ml and create new contents like text, images, video, audio

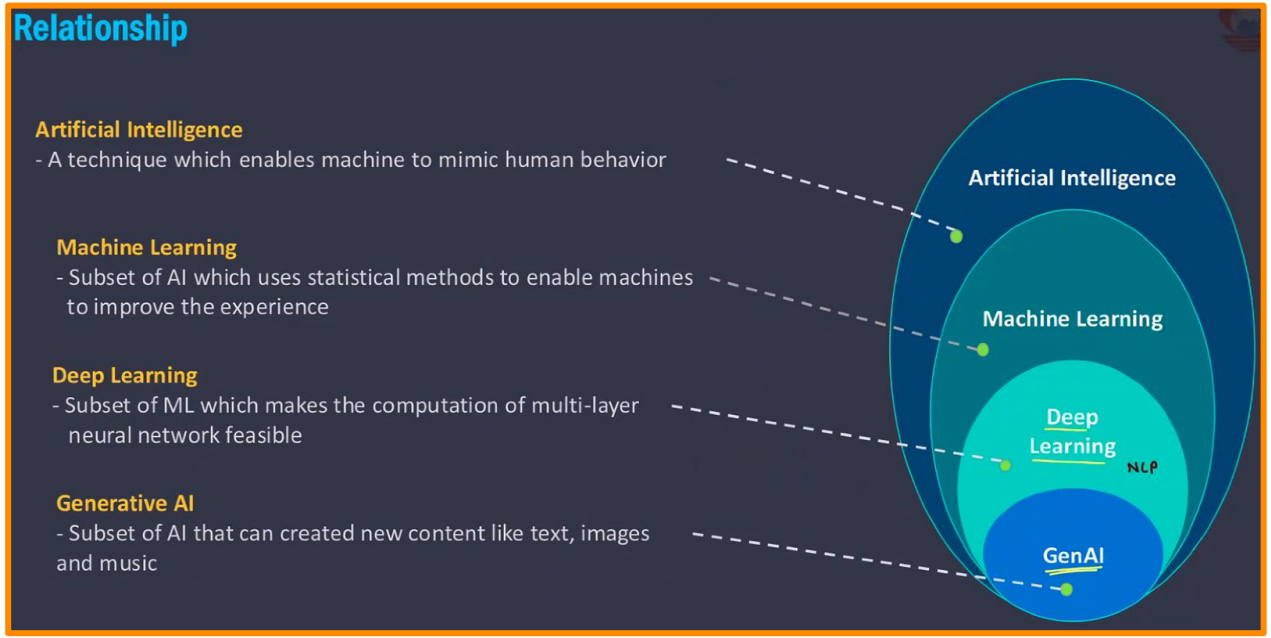
Generative meaning it can generate data

Can generate text, audio or video, images

AI stands for Artificial Intelligence

Gen AI & NLP is part of Deep Learning.

Deep Learning is part of ML and ML is part of AI



# Intelligence

**Intelligence** is the ability to

- Acquire
- Understand
- Apply knowledge and skills

To solve the problems or the make decisions

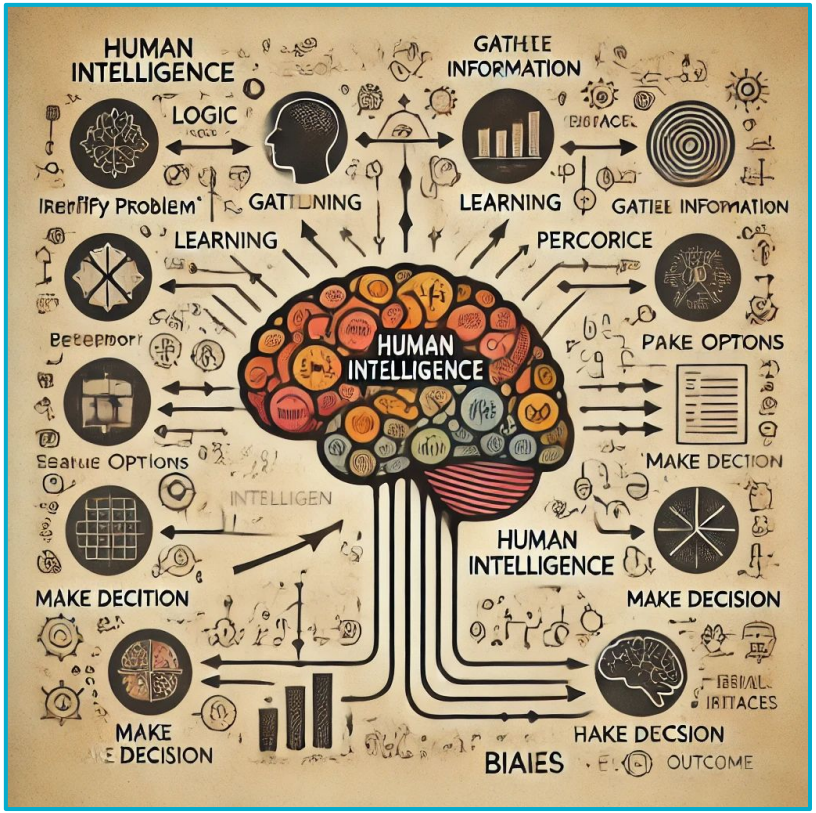
“Past experience or knowledge” is important to take the decisions

Taking Decision is important aspect of Intelligence

Recognize the problem, analyze it using past experience and knowledge, determine possible solutions, and decide on the best course of action.

Decisions can be Strategic, Tactic, Financial, and more

Humans take decisions every moment, seconds



# Understand Data & Decisions

How will you choose restaurant ?

- Cuisine Preference
- Budget
- Location
- Ambience & Atmosphere
- Reviews & Ratings
- Service Quality
- Hygiene & Cleanliness
- Availability & Reservations
- Extra Features

Data/knowledge plays key role in taking decision

Data can be past experience or new information (Searched or explored)

Also we should have clear problem statement (What is needed ?)

# Let's Solve one Problem or Take Decision

When **a = 2** Then what is value of **b ?**

We won't be able to find the value of **b** confidently because outcome is not well defined

**Let's try again :**

When a = 2 then b = 4
When a = 3 then b = 9
When a = 4 then b = 46
When a = 5 then b = 25
When a = 6 then b = 36
When a = 7 then b = ?
When a = 8 then b = ?

When we get additional data and it's expected values then we able to find the value of 7 & 8

To find value of 7 & 8 we analyse the data we had and tried to find relation between then

ex. What is relation between a & b i.e 2&4, 3&9

After analysis we found the relation i.e. b is square of a

And then we set the formula in our mind i.e b=a*a

Now it's easy to decide values of b i.e 7 * 7 = 49

**Terminologies:**

Data we are going to predict or find is called dependent data

Data we are going to use to predict or find is called Independent data

We must have only one data point to find or predict

Meaning at a time our formula should give only one output

Hence, **we must have only one dependent variable**

We can have more than one independent variables

All the data points called as variable

Independent data

Dependent data

| a | b |
|---|---|
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |
| 6 | 36 |
| 7 | 49 |
| 8 | 64 |

# Understand Different Kinds of Use Cases

# Predict Salary on the basis of YoE : Data


What should be the salary of an engineer having 5 years of experience ?

We can't guess until we know something ? what should we know ?

We must know salary of few persons, like…

Person 1 : YoE - 1 : Salary - 10,000
Person 2 : YoE - 2 : Salary - 15,000
Person 3 : YoE - 3 : Salary - 20,000
Person 4 : YoE - 4 : Salary - 25,000

} Data

Once we know the salary of few person we can guess or predict desired person's salary i.e **30,000**

*"**Data** is important to guess or predict"*

Data = Input & Correct Output

# Predict Salary on the basis of YoE : Relation


What should be the salary of an engineer having 5 years of experience ?

**But how did we guess it from the data ?**

Person 1 : YoE - 1 : Salary - 10,000
Person 2 : YoE - 2 : Salary - 15,000
Person 3 : YoE - 3 : Salary - 20,000
Person 4 : YoE - 4 : Salary - 25,000

} Data

We tried to find the **relation** between YoE and the Salary

We found that **minimum** salary is 10,000

Every year salary is getting **increased** by 5000

So the formula to calculate salary is ?

**Salary = 5000 × YoE + 5000**

Then by using formula we can predict the data

*"**Relation** between data is called **formula**"*

*Formula = Model*

# Data (input + correct o/p) + Relation + Learning = Supervised Learning

We Derived formula/model by analysing the data (input + correct output)

Using the formula we can now predict the salary of any given experience

**Formula** is key for prediction

To find the formula we used/analysed Input data with correct answers

When we learn (find formula) using the input data with correct answers then that type of learning called "Supervised Learning"

The name comes from "Supervision"

Correct output from the data Supervise/Guide the learning, hence called Supervised Learning

Person 1 : YoE - **1** : Salary - **10,000**
Person 2 : YoE - **2** : Salary - **15,000**
Person 3 : YoE - **3** : Salary - **20,000**
Person 4 : YoE - **4** : Salary - **25,000**

Data

**Salary = 5000 × YoE + 5000**

# Supervised Learning : Regression

We tried to find relation between given numbers & the correct output numbers

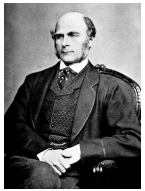
When we find relation between two of such numbers is called **“Regression Analysis”**

Person 1 : YoE - **1** : Salary - **10,000**
Person 2 : YoE - **2** : Salary - **15,000**
Person 3 : YoE - **3** : Salary - **20,000**
Person 4 : YoE - **4** : Salary - **25,000**

Data

**Salary = 5000 × YoE + 5000**

## Regression Analysis :



Francis Galton
Cousin of Darwin

Francis Galton Cousin of Darwin studied the heights of parents and their children.

**He noticed**
Very **tall** parents had children who were a bit **shorter** than them (closer to average).
Very **short** parents had children who were a bit **taller** than them (closer to average).

In other words, the children’s heights **“regressed to the mean”** (moved closer to the average population height).

Galton called this phenomenon **“regression toward mediocrity”**, later shortened to **regression**.

**The statistical method he used to study this relationship** (parent → child height) **became known as “Regression Analysis.”**

# Supervised Learning : Regression : Linear Relations

After analysis of salary data, the relation between YoE and Salary is,

$Y = aX + b$

Y = Salary = Output
X = YoE = Input
a = Rate of Salary Change = Constant (slope)
b = Minimum Salary = Constant (Y-Intercept) => Minimum Salary at 0 experience

Person 1 : YoE - **1** : Salary - **10,000**
Person 2 : YoE - **2** : Salary - **15,000**
Person 3 : YoE - **3** : Salary - **20,000**
Person 4 : YoE - **4** : Salary - **25,000**

Data

**Salary = 5000 × YoE + 5000**


When we plot input and Output data on graph and see the relation between them as straight **line**, such relation called **Linear Relation**

Since this Linear Relation works with numbers only, it falls into Regression Category and called "**Linear Regression**"

*"Linear regression is a statistical method used to model the relationship between a one or more inputs variables and single output by fitting a linear equation to the observed data"*

# Terminologies Consolidated

## Machine Learning

### Supervised Learning

- **Regression** (Predict continuous numbers)
  - Linear
  - Polynomial
  - Ridge/Lasso
- **Classification** (Predict distinct Categories)
  - Logistic
  - SVM
  - Decision Tree
  - kNN
  - Random Forest
  - Neural Network

### Unsupervised Learning

- **Clustering**
  - k-Means
  - Hierarchical
  - DB-SCAN
  - GMM (Gaussian Mixture Model)
- **Dimensionality Reduction**
  - PCA (Principal Component Analysis)
  - t-SNE
  - LDA (Linear Discriminant Analysis)
  - Auto Encoder
- **Association Rule Learning**
  - Apriori
  - Eclat

# Terminologies

**Terminologies:**

Data we are going to use to predict or find is called Independent (input) data

Data we are going to predict or find is called dependent (output) data

We must have only one data point to find or predict.

Meaning at a time our formula should give only one output

Hence, **we must have only one dependent variable**

We can have more than one independent variables

The formula we found/learned from the already given values of a & b is a*a is called **model**

When we have or we use given independent & dependent data to find/learn the formula/model then we call it **Supervised Learning.** Because we got guidance/help to find the formula

When we don't have dependent variable/data (we don't need to predict) then finding/learning the formula/model called as **Unsupervised Learning** - will see this much later

Independent data

Dependent data

| a | b |
|---|---|
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |
| 6 | 36 |
| 7 | 49 |
| 8 | 64 |

# AI and Libraries

**Artificial Intelligence** is the **Computer Program** which

- Acquire
- Understand
- Apply knowledge and skills

To solve problems and make decisions

Meaning, whatever we or our brain did in our earlier example, should be done by Computer Program

To support this programming there are lot of libraries are available

**pandas** – A library for data manipulation and analysis, offering powerful DataFrame structures.

```
pip install pandas
```

**numpy** – A fundamental package for numerical computing in Python, providing support for large, multi-dimensional arrays.

```
pip install numpy
```

**scikit-learn** – A machine learning library with tools for classification, regression, clustering, and preprocessing.

```
pip install scikit-learn
```

**matplotlib** – A plotting library for creating static, animated, and interactive visualizations in Python.

```
pip install matplotlib
```

**seaborn** – A statistical data visualization library built on top of Matplotlib for attractive and informative graphs.

```
pip install seaborn
```

**Data**

When a = 2 then b = 4
When a = 3 then b = 9
When a = 4 then b = 4
When a = 5 then b = 5
When a = 6 then b = 36
When a = 7 then b = ?
When a = 8 then b = ?

**Formula/Model**

**b = a * a**

# Machine Learning Process : Steps

- Look at the big picture : Understand what is needed
- Get the data
- Discover and visualize the data to gain insights (EDA Process)
- Prepare the data for Machine Learning algorithms (Data Scrubbing process)
- Select a model and train it
- Test the model
- Fine-tune your model
- Launch, monitor, and maintain your system


# Let’s do some coding before more theory

# Salary Prediction : Data Analysis

## Load and Inspect the Data

- Check data types
- If any missing values
- Check Min, Max, std, Mean
  - Mean is used to find the average value,
  - Median is used to find the middle value,
  - Mode is used to find the most frequently occurring value
- Ensure Data is clean and ready for analysis

## **Visualize the data** and see how it fit's in X & Y (Input and Output)

- Use scatter plots which helps to understand relationship of data
- Visual inspection shows Trend (linear or nonlinear)

## **Measure the relationship** to gain more confidence (Covariance & Correlation )

### Correlation

Correlation is a normalized form of covariance
It is a measure of how much two variables change together
Correlation ranges from -1 to 1
1 means that the variables are perfectly correlated
0 means that the variables are not correlated
-1 means that the variables are perfectly inversely correlated

### Covariance

Covariance is a measure of **how much two variables change together**
Positive covariance means that the variables are directly proportional
Negative covariance means that the variables are inversely proportional
Zero covariance means that the variables are not related

- Covariance of x and y = Σ((x - mean(x)) * (y - mean(y))) / (n-1)
- Covariance of x = Covariance of y
- [[cov(x,y)]] = cov(x,x) = cov(y,y)

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

df = pd.read_csv("salary_data.csv")
print("Columns: ",df.columns)
df.info()
print("Describe: ",df.describe())
print("Mean Salary: ", df['Salary'].mean())
print("Median Salary: ", df['Salary'].median())
print("Mode Salary: ", df['Salary'].mode()[0])


plt.scatter(df['Experience'], df['Salary'])
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.title('Experience vs Salary')
plt.show()  # Uncomment to see graph

covariance = np.cov(df['Experience'], df['Salary'])
print("Covariance: ", covariance)

correlation = df['Experience'].corr(df['Salary'])
print("Correlation: ", correlation)
```

Ref:
https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.SupervisedLearning/1.RegressionAnalysis/2.ModelToFindSalary/2.1.SalaryPredictionDataAnalysis.py

# Predict Salary (1D data for training) : Linear Regression

**Problem Statement :**

Predict the salary on the basis of Years of Experience

To find formula we need some data to analyse and find formu

(train the mode)

**Understand Code :**

Import the required packages

Here we used panda library to manipulate the data (tables)

Library sklearn is machine learning library helps to data processing, training the mode, testing the model and do the prediction

Used CSV file to store training data

We always need 2 dimensional independent data hence we just dropped dependent data from the data frame (data set)

**Why 2D?**

**Machine learning models work with datasets, not single values.**

Even if predicting for one value, the model expects an array with **rows (samples) and columns (features)**.

**Ensures consistency** when predicting multiple values at once.

Salary is dependent data which is always one dimensional

We used LinearRegression algorithm (why ? will see later)

```python
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load the dataset from a CSV file
df = pd.read_csv("salary_data.csv")

# Separate the features (independent variables) and the target
(dependent variable)
x = df.drop('Salary', axis=1)  # Features: all columns except 'Salary'
y = df['Salary']  # Target: 'Salary' column

# Initialize the Linear Regression model
model = LinearRegression()

# Fit the model to the data
model.fit(x, y)

# Predict the salary for 15 years of experience
salaries = model.predict(pd.DataFrame([[15]], columns=['Experience']))

# Print the predicted salary
print("Salary of 15 years of experience is:", salaries[0])
```

Ref: https://github.com/zodgevaibhav/gen-ai-learning/tree/main/2.ModelToFindSalary

# Understand model.predict() function

Predict function needs “Independent Data” or Input data & Column Names

Predict function returns the result in array format

During Prediction pass the data with same shape with same column names

- Model training remembers feature names.
- If names don’t match, sklearn warns you to prevent wrong predictions.

There will be always one and one dimensional output/prediction array

As output is 1D array, this does mean we can send multiple data for prediction

```
model.predict(pd.DataFrame([[15],[20],[25]],
columns=['Experience']))
```

We can skip the column names but then there are chances of incorrect prediction which sklearn does ward

```
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load the dataset from a CSV file
df = pd.read_csv("salary_data.csv")

x = df.drop('Salary', axis=1)  # Features: all columns
except 'Salary'
y = df['Salary']  # Target: 'Salary' column

# Initialize the Linear Regression model
model = LinearRegression()

# Fit the model to the data
model.fit(x, y)

# Predict the salary for 15 years of experience
salaries = model.predict(pd.DataFrame([[15]],
columns=['Experience']))

# Print the predicted salary
print("Salary of 15 years of experience is:",
salaries[0])
```

Ref: https://github.com/zodgevaibhav/gen-ai-learning/tree/main/2.ModelToFindSalary

# Understand Steps of Machine Learning Steps

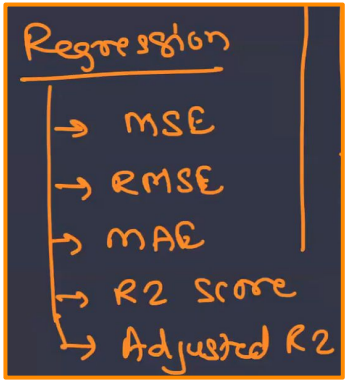
# Machine Learning Process : Steps

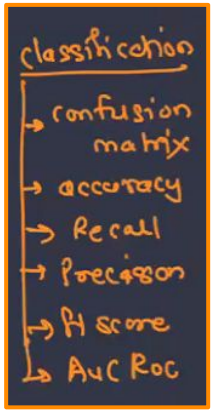
1. What do I want to cook? (Problem)
2. Get ingredients (Data)
3. Taste/check ingredients (EDA)
4. Wash, cut, prepare (Preprocessing)
5. Pick recipe (Model)
6. Cook (Train)
7. Taste & adjust (Evaluate/Tune)
8. Serve (Present)
9. Improve recipe next time (Deploy & Monitor)

# End to End Process : Looking at Big Picture

**Frame the Problem**

- The first question to ask your boss is what exactly the business objective is
- Building a model is probably not the end goal
- How does the company expect to use and benefit from this model?
  - Will this model will be final product
  - Will this model will be used under some product
- Knowing the objective is important because it will determine
  - How you frame the problem (Whether it is Regression, Classification problem or clustering)
  - Which algorithms you will select
  - Which performance measure you will use to evaluate your model. How do I know model is behaving right or wrong.
  - How much effort you will spend tweaking it




**Select a Performance Measure**

- Your next step is to select a performance measure
- A typical performance measure for regression problems is the Root Mean Square Error (RMSE)
- It gives an idea of how much error the system typically makes in its predictions, with a higher weight for large errors

# End to End Process : Get The Data (Data Collection)

- Decide the data source (Get from API, Scrape the data, buy the data, CSV, JSON)
- Download the data and make it available for the further learning
- Take a Quick Look at the Data Structure
    - Understand the data set and features are
        - What kind of columns & rows are available
        - Which are relevant according to objective
    - Evaluate the features and decide which one(s) are needed
        - Which are independent data
        - Which is dependent data
- Create a Test Set
- Keep some records aside for testing and validation (20% of data for testing)

# End to End Process : Discover and Visualize the Data to Gain Insights

- Visualize the data
  - Convert data in to diagram, charts, graphs or tables
- Use libraries like matplotlib or seaborn for data visualization
- Understand the pattern and relationship
  - Data analysis to be done here from visualization
- Look for correlation
  - Is house price dependent on location ? Yes (Correlation found)
  - Is house price depend on #rooms ? Yes (Correlation found)
  - Is house price is depend on id ? No (No correlation)
- Experiment with attribute combinations

| id | location | #rooms | prics |
|---|---|---|---|
| 1 | Kalyani nagar | 2 | 1.2 cr |
| 2 | Sinhgad | 3 | 1.2 cr |
| 3 | Magarpatta | 2 | 4 cr |
| 4 | Shivaji Nagar | 5 | 1.5 cr |

Independent Column

Dependent Column

Correlation

# End to End Process : Prepare the Data for Machine Learning Algorithms

**Data Cleaning**

Process of cleaning the data set to prepare it for ML algorithm

**Steps**

Check for the missing data

Check for wrong data types

Add features if needed

Remove unwanted features

**Feature Scaling**

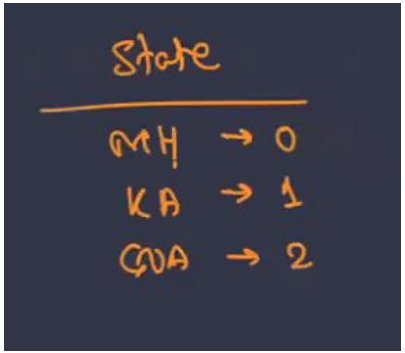
ML algorithms don't perform well when the input numerical attributes have very different scale

Scale the features to bring all of them to a single scale

**Handle categorical / text data**

Use transformers to convert "categorical or textual" to numerical. We must use numerical data

We can use encoders to convert textual data in to number



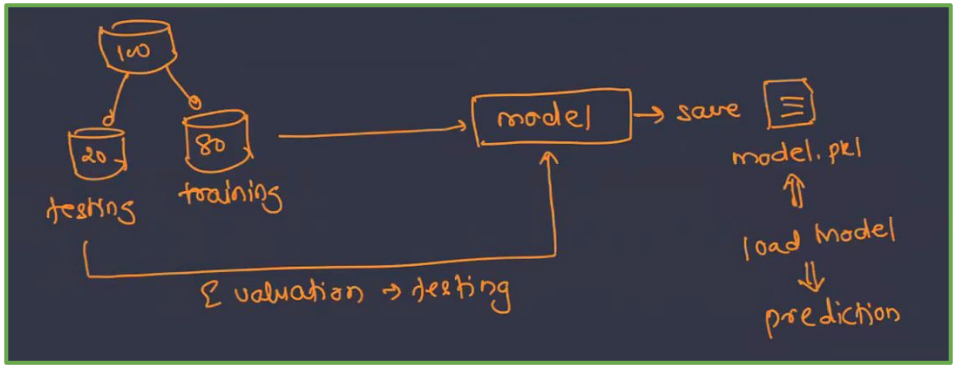
# End to End Process : Select and Train a Model

Training the model using train data set

- Create a model using selected algorithm
- Save the model for future use

Evaluation the model

- Evaluate the model to see if there is any chance to improve the accuracy
- Techniques
  - Cross Validation



# Understand various equations

# End to End Process : Fine-Tune Your Model

Grid Search

- One option would be to fiddle with the hyperparameters manually, until you find a great combination of hyperparameter values
- This would be very tedious work, and you may not have time to explore many combinations
- You can also automate this process using libraries like sci-kit

Randomized Search

- The grid search approach is fine when you are exploring relatively few combinations
- But when the hyperparameter search space is large, it is often preferable to use randomized search

Ensemble Methods

- Another way to fine-tune your system is to try to combine the models that perform best
- The group (or “ensemble”) will often perform better than the best individual model, especially if the individual models make very different types of errors.

Analyze the Best Models and Their Errors

Evaluate Your System on the Test Set

# End to End Process : Launch, Monitor, and Maintain Your System

Deploy the application for the end users

Monitor the application's performance

If the data keeps evolving, update your datasets and retrain your model regularly

- More the data better the accuracy

You should probably automate the whole process as much as possible (using MLOps)

- Collect fresh data regularly and label it
- Write a script to train the model and fine-tune the hyperparameters automatically.
- This script could run automatically, for example every day or every week, depending on your needs.
- Write another script that will evaluate both the new model and the previous model on the updated test set, and deploy the model to production if the performance has not decreased (if it did, make sure you investigate why)

# Predict Price (2D data for training) : Linear Regression

**Problem Statement :**

Predict the price on the basis of mileage and age of car

To find formula we need some data to analyse and find formula (train the mode)

**Understand Data & Code:**

Here Age and Mileage is the independent data

Unlike previous example here we are giving two variables for prediction

Price is the only dependent data

As seen in code we dropped price column and given rest column as feature

While prediction we must need to give two values as we trained model using two values

| Age | Mileage | Price |
|---|---|---|
| 7 | 16394 | 21547.3 |
| 4 | 74032 | 21447.4 |
| 13 | 8890 | 16582.5 |
| 11 | 46606 | 17364.7 |
| 8 | 92313 | 16879.35 |

```
import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.read_csv("car_price_data.csv")

# Separate the features (independent variables) and
the target (dependent variable)
x = df.drop('Price', axis=1)  # Features: all columns
except 'Price'
y = df['Price']  # Target: 'Price' column

model = LinearRegression()
model.fit(x, y)

predictions = model.predict([[2,20000]])
print(predictions)
```

Ref: https://github.com/zodgevaibhav/gen-ai-learning/tree/main/3.ModelToFindPriceBasedOnMileageAndAgeOfCar

# How Prediction works ? (Predict Salary (1D data for training) : Linear Regression)

**We used Linear Regression Algorithm :**

Linear Regression is a Supervised Learning Algorithm used to predict the continuous variables

**Linear Regression :**

When output change with constant/linear rate of input

Salary is directly proportional to years of experience

Considered change in year of experience make change in salary at constant rate

Whenever we want to predict the value on such linear rate of change then we should use Linear Regression Algorithm

Examples :

Salary Prediction on the basis of experience

Car Price prediction on the basis of Age and Mileage

How price prediction on the basis of Squarefoot

Supervised Learning :

Supervised learning meaning learning use independent data.

Independent Data guide/supervise the model for prediction hence called SL

Continuous Variables:

Continuous variable is type of variable which can take any numeric value

It just mean any number for any given range

## 1. Data Analysis & Preprocessing

**Understanding the Data:**

- The model first **analyzes the dataset** to understand the relationship between dependent (target) and independent (predictor) variables.
- Example: **Salary Prediction** → Years of Experience (X) vs. Salary (Y).

**Handling Missing & Outlier Values:**

- Missing values are **handled (imputation, removal, etc.)**.
- Outliers are **detected and analyzed** as they might impact the regression model.

**Checking for Linearity:**

- A **scatter plot** is used to see if there is a linear relationship.

## 2. Finding the Best-Fit Line (Model Training Process)

**Mathematical Representation:**

- The model assumes the relation follows the equation of a line: $Y=mX+cY = mX + cY=mX+c$ where:
  - YYY = Dependent variable (Salary)
  - XXX = Independent variable (Years of Experience)
  - mmm = Slope (Rate of increase in salary per year)
  - ccc = Intercept (Base salary with 0 experience)

**Cost Function (Error Measurement):**

- To find the best-fit line, the model calculates **errors (differences between actual and predicted values).**
- The most common cost function used is **Mean Squared Error (MSE):**

**Finding the Optimal Parameters (m & c):**

- The model **adjusts the values of mmm (slope) and ccc (intercept)** to minimize the MSE.
- This process is called **optimization**, typically done using **Gradient Descent** or **Ordinary Least Squares (OLS)**.

# How Prediction works ? (Predict Salary (1D data for training) : Linear Regression)

## 1. Data Analysis & Preprocessing

**Understanding the Data:**

- The model first **analyzes the dataset** to understand the relationship between dependent (target) and independent (predictor) variables.
- Example: **Salary Prediction** → Years of Experience (X) vs. Salary (Y).

**Handling Missing & Outlier Values:**

- Missing values are **handled (imputation, removal, etc.)**.
- Outliers/issues are **detected and analyzed** as they might impact the regression model.


## 2. Finding the Best-Fit Line (Model Training Process)

**Mathematical Representation:**

- The model assumes the relation follows the equation of a line: $Y = mX+c$
  where:
  - Y = Dependent variable (Salary)
  - X = Independent variable (Years of Experience)
  - m = Slope (Rate of increase in salary per year)
  - c = Intercept (Base salary with 0 experience)
- Model file the slope value and constant (intercept) from the data trend.
- Then using mX + C formula it calculates the salary.
  Assume calculated values from trends are : m (slope) = 10,000 and c (Intercept) = 30,000
  Calculate the salary for Years of Experience **5**

  $Y = (10,707 \times 5) + 20,000 = 73,538$

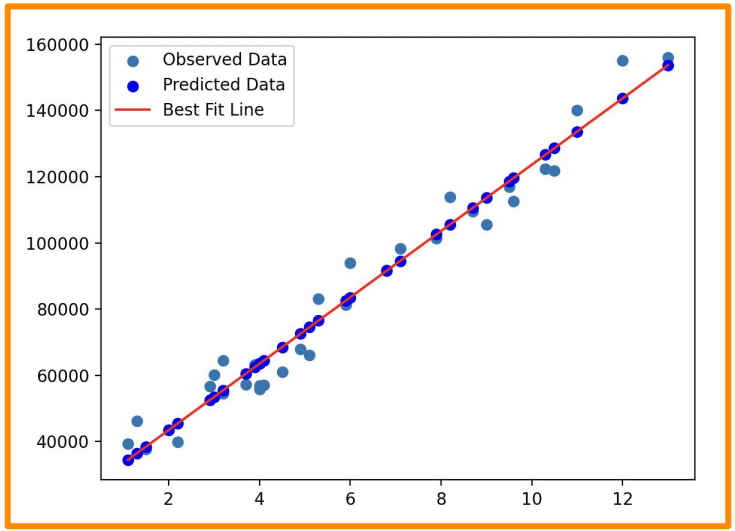
# Visualize Predicted Values and how the best fit line shows

As we seens model analyse the given data, find the relation and draw the best fit line

Best fit line is representation of the prediction

Hence if we want to visualize the best fit line then draw scatter on input and predicted o/p

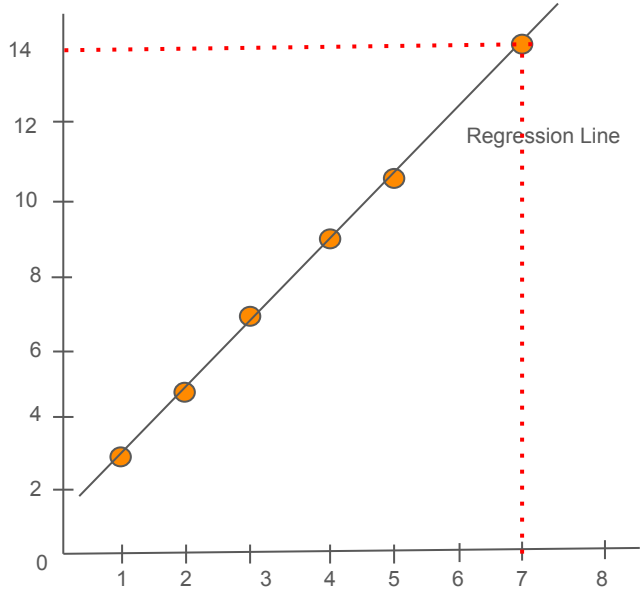
Also if we draw plot (line) on predicted values then will get best fit line




```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

# Load the dataset from a CSV file
df = pd.read_csv("salary_data.csv")

# Separate the features (independent variables) and the target (dependent
variable)
x = df.drop('Salary', axis=1)
y=df['Salary']

# Initialize the Linear Regression model
model = LinearRegression()

# Fit the model to the data
model.fit(x,y)

# Since we have model which is trained, we can plot the BEST FIT REGRESSION LINE
plt.scatter(df['Experience'], df['Salary'],label='Observed Data')
plt.scatter(df['Experience'], model.predict(x), color='blue',label='Predicted
Data')
plt.plot(df['Experience'], model.predict(x), color='red',label='Best Fit Line')
plt.legend()
plt.show()
```

# Understand Linear Regression

| x | y |
|---|---|
| 1 | 3 |
| 2 | 5 |
| 3 | 7 |
| 4 | 9 |
| 5 | 11 |
| 7 | ? |



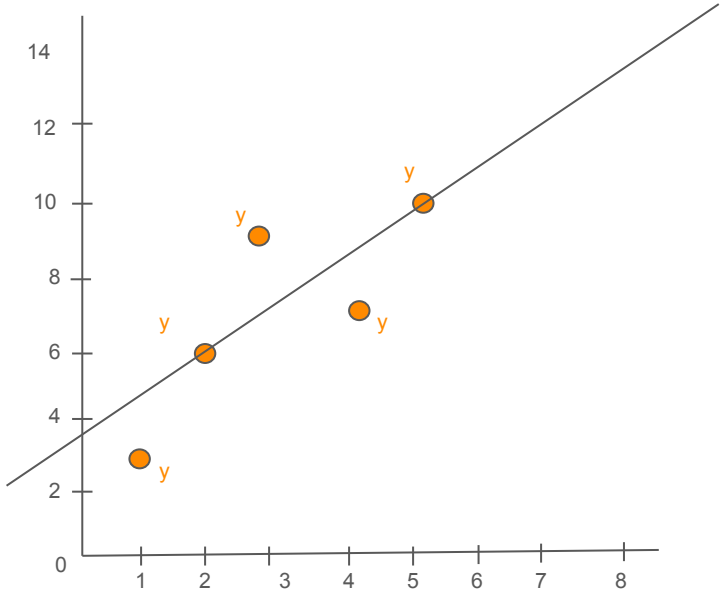
# Understand Linear Regression

Many time will not get Linear Straight Line

Hence we need to validate if my drawn line is near to best fit line ?

y

| x | y |
|---|---|
| 1 | 3 |
| 2 | 6 |
| 3 | 9 |
| 4 | 7 |
| 5 | 10 |
| 7 | ? |



# Confidence on Formula

Let's go back to our example…

When a = 2 then b = 4
When a = 3 then b = 9
When a = 4 then b = 4
When a = 5 then b = 5
When a = 6 then b = 36
When a = 7 then b = ?
When a = 8 then b = ?

Formula we derived out of above dependent and independent variable is

**b = a * a**

Independent data

Depender data

| a | b |
|---|---|
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |
| 6 | 36 |
| 7 | 49 |
| 8 | 64 |