Q1 ) difference between AI , ML , DL ? what is AI, ML , DL ?

- let consider AI as universal set. anything you do , that comes under AI and definition of ai is ability of the computer to work , act and mimic like a human.

- ability to work independently and intelligently without human intervention.

- and machine learning is a subset of an AI , it provide stat tool to forcast and predict , classify , recommendation and analyse.

- it uses different algorithm and stat model or tool to do the prediction , analysis , classification, recomendation or forcast.

- than deep learning comes under machine learning and it is advance concept here we try to mimic the human brain and use neural network algorithm to do the job done like object detection, object recognition object classification , etc.

Q2 ) type of ML? , difference between Supervised learning and unsupervised learning?

what is classification and regression. ? type of supervised learning?

- in supervised learning there are multiple independent variable and one dependent variable.

- and based on the target or dependent variable supervised learning is divided into two type

- we consider the problem as regression problem when our dependent variable or feature is continuous or in range.

eg . price of house. height of person.

algorithm we can use are linear regression , lasso and ridge , decision tree regression , random forest regression, xgboost

- and we consider the problem as class cation problem when our dependent variable or feature is discrete variable or feature.

eg . customer will churn or not , email is spam or not

algorithm we can use are logistic regression (for binary classification) , lasso and ridge classification, decision tree classification , random forest classification, xgboost classification

in unsupervised learning there is no output , we used different algorithm and try to find the trend , pattern , cluster in the dataset . algorithm we can use are kmeans clustering , hirarical clustering, dbscan clustering , etc.

eg customer segmentation.

# Algorithms

## Linear regression

What is linear regression?

* Linear regression is a basic statistical method used to understand the relationship between dependent and independent variable.
* In simple terms, it's about finding the best straight line that fits through a set of data points.
* In y = mx+c quation of linear equation , The goal is to find the slope (m) and the y-intercept (c) that minimize the difference between the actual data points and the predicted values on the line.
* Technique like MSE , RMSE are used to evaluate the accuracy of the model trained by the linear regression.

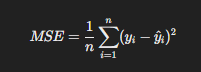
Example :

Imagine you want to predict someone's height based on their shoe size.

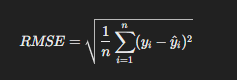
Loss Functions (to evaluate the performation)

## MSE & RMSE & MAE

## MSE : Mean Square Error

* MSE is the average of the squared differences between the predicted and actual values.
* 
* Example output: Mean Squared Error (MSE): 0.555
* **Interpretation**: The average squared difference between the predicted and actual house prices is 0.555.

## RMSE : Root Mean Square Error

* RMSE is the square root of the MSE. It brings the metric back to the same units as the original data, making it easier to interpret.
* These are most appropriate when large errors are particularly undesirable and when the errors' units should match the original data for easier interpretation. 

## Mean Absolute Error (MAE)

 Example output: Mean Absolute Error (MAE): 0.531

 **Interpretation**: On average, the model's predictions are off by $53,100.

Regularization techniques : techniques are used to prevent overfitting by adding a penalty term to the loss function of a model.

There are more linear regression algorithm but they are advance version of the simple linear regression. While **Linear Regression** Minimizes the mean squared error without any regularization.

Where **L2 Regularization (Ridge)**: Adds squared **magnitude** of coefficients as penalty term.

**L1 Regularization (Lasso)**: Adds **absolute** value of coefficients as penalty term.

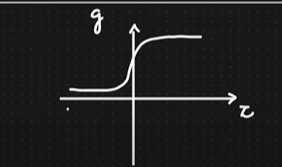
## Ridge and lasso regresser.

## Cross-Validation

## Logistic regression

Logistic regression is a statistical method used for binary classification problems, where the goal is to predict one of two possible outcomes (e.g., yes/no, true/false, 0/1).

Take a linear equation of best fit line and apply sigmoid function on top of it. You will hava a curve in the line hance make it able to classify better.



For classification we use the confusion matrix to evaluate the model and use precision and recall to evaluate certain senarios.

## Naive Bayes

The Naive Bayes algorithm is a classifier algorithm based on Bayes' Theorem or probability.

it assumes that all features are independent of each other, in order to make calculation easy.

Works well for problems like spam detection and sentiment analysis.



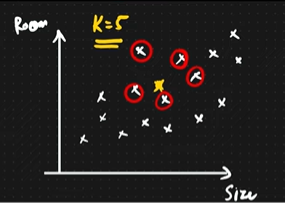
## K Nearest Neighbour (KNN)

This algorithm can be both regression algorithm and classification algorithm.

It finds out k nearest data points based k value using Eucledian distance or manhathan distance.

if we are using this algorithm for classification then we find out maximum k nearest data points fall in which category.

if we are using this algorithm for regression then we find out average of k nearest data points.



## Decision Tree

* It is basically classification and regression algorithm , and unlike other algorithm like linear and logistic regression it is non linear algorithm.
* We keep on splitting the table until we get the leaf node where we can split more
* Classification
  + We can calculate the information gain to select the best feature to start splitting with and keep on repeating.
  + And we check the impurity of split after calculating entropy(0,1) or gini\_index(0,0.5). then we take a decision to where it require to split more or not. For larger dataset must use gini\_index because it uses simple maths while entropy uses log which takes more time.
* Regression : in Regression we use the variance reduction technique to split the tree.

Decision Tree Classifier are prone to overfitting hence : (Hyperparaet)

**Post Pruning** And **Pre Pruning** In Decision Tree Classifier

Lets say you reach the leaf nodes , and you have the overfitting situation. **So Pruning helps us to cut down extra splits or braches of the tree which are unnecessary and it is hyper parameter technique** , let say I got the result in 3 depth of the tree , further splitting makes my model overfit to the training dataset.

In post **Post Pruning** we decide the depth of the tree after the model is train. (small dataset)

In post **Pre Pruning** we decide the depth of the tree after the model is train. (large dataset)

Cross validation (CV) technique to find out the best accuracy and also do the hyper parameter tuning

with the training dataset.

Let say CV=5



Take the accuracy of all the splits and take a mean of accuracy so we got the training data accuracy and than check the final accuracy with the test data accuracy .

Than we can take the decision model is either overfitting or underfitting.

Overfitting, Underfitting, general model

Model evaluation:

**In regression model :**

Train == bias

Test == variance

Low or high based on error

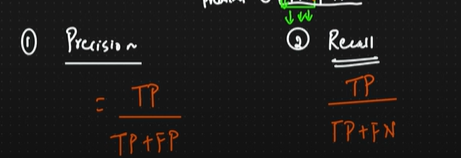
Underfitting : training error Is very high (high bias) , testing error is also high (high variance)

Overftting : training error is low (low bias) , test error is high (high variance)

General model : training error is low (low bias) , test error is low (low variance)

**In classification model :**

**Confusion matrix :**

galat tha galat aya

galat tha Sahi aya

Sahi tha galat aya

Sahi tha sahi aya

**Precision :**

Aim is to lower the false positive.

Ex. Spam or not , here we will minimise FP , because it is ok to see the spam mail but if important mail must not go to spam

**Recall :**

Aim is to lower the false negative.

Ex. fever or not , here we will minimise FN , disease must be detected first , if predicted wrong in the first place than it is wrong.

# EDA

Faste EDA with the library

1. **Profile pandas**
2. **Sweetviz**

pip install sweetviz

import sweetviz as sv

import seaborn as sns

df = sns.load\_dataset('tips')

my\_report = sv.analyze(df)

my\_report.show\_html()

1. **dtale**

pip install dtale

import dtale

import seaborn as sns

df = sns.load\_dataset('tips')

d = dtale.show(df)

d.open\_browser()

1. **Autoviz**

pip install autoviz

from autoviz import AutoViz\_Class

import seaborn as sns

%matplotlib inline

df = sns.load\_dataset('tips')

# df.head()

AV = AutoViz\_Class()

filename = df

# target\_variable = "Horsepower"

dft = AV.AutoViz(

    "",

    sep=",",

    # depVar=target\_variable,

    dfte=df,

    header=0,

    verbose=2,

    lowess=False,

    chart\_format="svg",

    max\_rows\_analyzed=500,

    max\_cols\_analyzed=20,

    save\_plot\_dir=None

)

mean square error

id is differentiable, because we can find derivative at any point.

local minima , gloabal minima.

not robust to outlier

hence there is mean absolute error

we dont square hence it is robust to outlier. but hard to optimize.

Videos:

DSwithBappy

**12 Data Science Apps**: https://www.youtube.com/watch?v=JwSS70SZdyM

**End to end project** : https://www.youtube.com/watch?v=1m3CPP-93RI

**Mlops :** <https://www.youtube.com/watch?v=dPmH3G9NQtY>

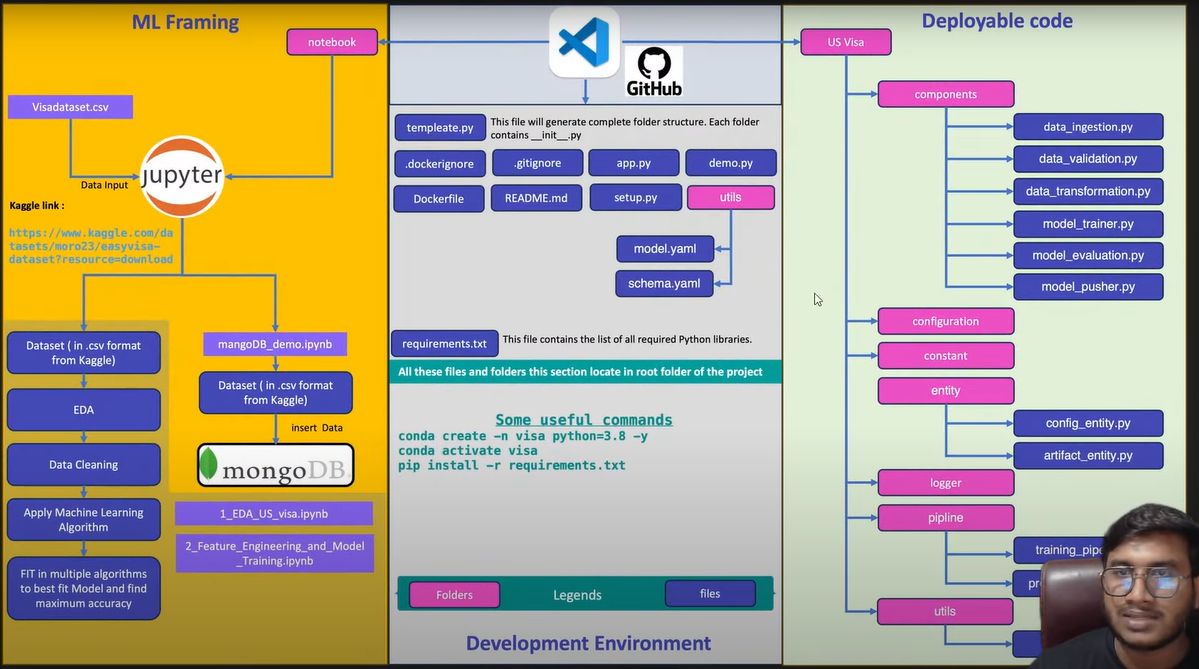
**Mock interview** : <https://www.youtube.com/watch?v=Naro4q5k4AM>

**Yolo sign language** : https://www.youtube.com/playlist?list=PLkz\_y24mlSJYWpwFbU8fyaBSwihoVHiJz

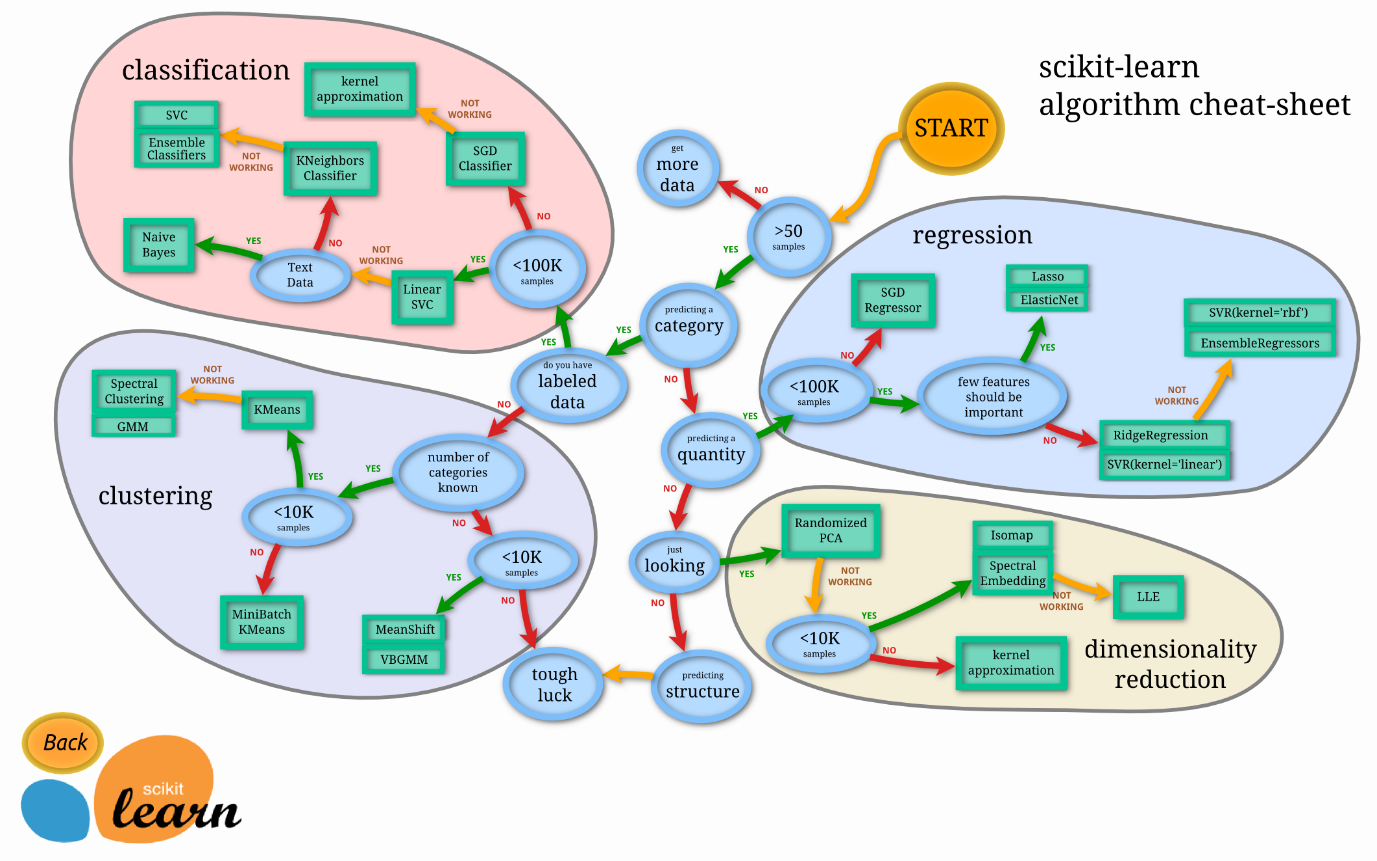
**MLOps Production Ready Machine Learning Project :**

* <https://www.youtube.com/playlist?list=PLkz_y24mlSJZvJOj1UXiJPVRQrNFdNEXX>
* https://www.youtube.com/playlist?list=PLkz\_y24mlSJZrqiZ4\_cLUiP0CBN5wFmTb

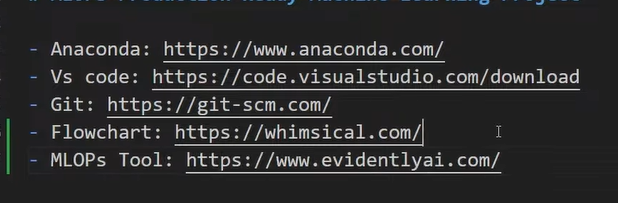
# Project Structure



1. Load the data from the db
2. EDA
3. ML algoritm
4. Hyperparameter tunning
5. Best model
6. Git repo
7. Requirement.txt
8. Template
9. Create a github repo
   1. Create readme.md
   2. Create .gitignore
10. Copy the url and git clone ………………………………….. in cmd to fetch all the file and than open the vs code



Useful links :



## Create a ML Project Structure Automatically using templete

import os

from pathlib import Path

project\_name = "us\_visa"

list\_of\_files = [

    f"{project\_name}/\_\_init\_\_.py",

    f"{project\_name}/components/\_\_init\_\_.py",

    f"{project\_name}/components/data\_ingestion.py",

    f"{project\_name}/components/data\_validation.py",

    f"{project\_name}/components/data\_transformation.py",

    f"{project\_name}/components/model\_trainer.py",

    f"{project\_name}/components/model\_evaluation.py",

    f"{project\_name}/components/model\_pusher.py",

    f"{project\_name}/configuration/\_\_init\_\_.py",

    f"{project\_name}/constants/\_\_init\_\_.py",

    f"{project\_name}/entity/\_\_init\_\_.py",

    f"{project\_name}/entity/config\_entity.py",

    f"{project\_name}/entity/artifact\_entity.py",

    f"{project\_name}/exception/\_\_init\_\_.py",

    f"{project\_name}/logger/\_\_init\_\_.py",

    f"{project\_name}/pipline/\_\_init\_\_.py",

    f"{project\_name}/pipline/training\_pipeline.py",

    f"{project\_name}/pipline/prediction\_pipeline.py",

    f"{project\_name}/utils/\_\_init\_\_.py",

    f"{project\_name}/utils/main\_utils.py",

    "app.py",

    "requirements.txt",

    "Dockerfile",

    ".dockerignore",

    "demo.py",

    "setup.py",

    "config/model.yaml",

    "config/schema.yaml",

]

for filepath in list\_of\_files:

    filepath = Path(filepath)

    filedir, filename = os.path.split(filepath)

    if filedir != "":

        os.makedirs(filedir, exist\_ok=True)

    if (not os.path.exists(filepath)) or (os.path.getsize(filepath) == 0):

        with open(filepath, "w") as f:

            pass

    else:

        print(f"file is already present at: {filepath}")