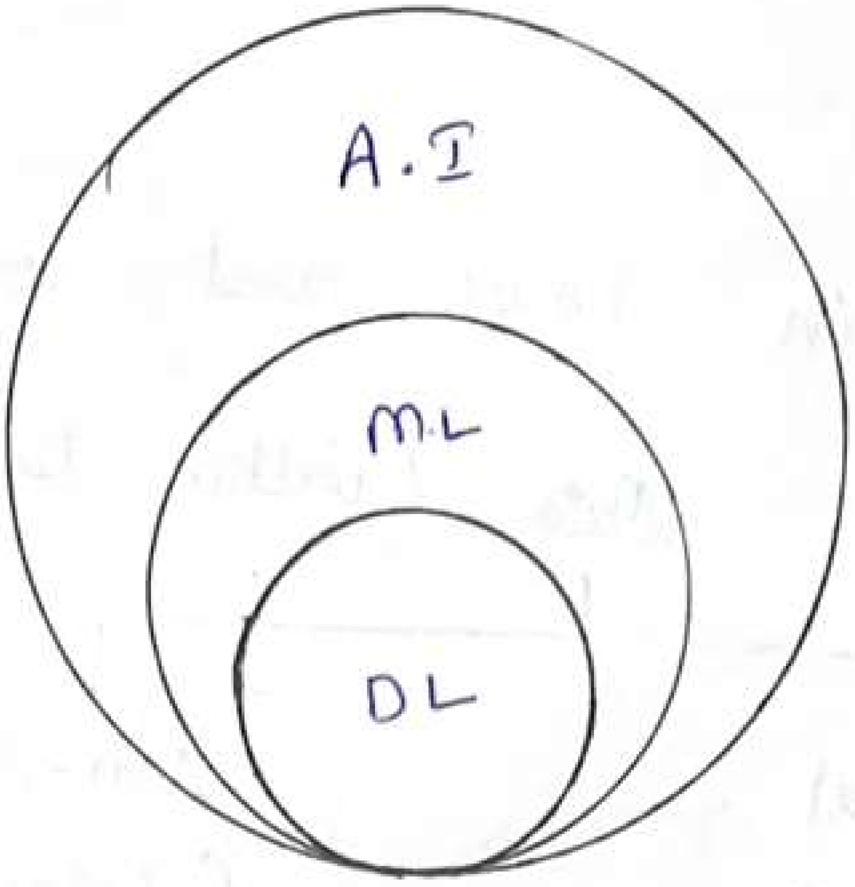
3/08/2023

Machine Learning

Introduction



AI (Artificial Intelligence) → which mimics human intelligence

ML (Machine Learning) → which is a subset of A.I. Learns from historical data to do some tasks.

DL (Deep Learning) → subset of M.L. Deals with complex algorithms for complex tasks.

The tasks which we are to perform are directly related to the data which we have in hand.

Data

Types :- Tabular | Image | Text

Tabular Data :-

Represented in rows and columns.

Data (within tabular)

Numerical (Continuous) | Non-Numerical (Categorical)

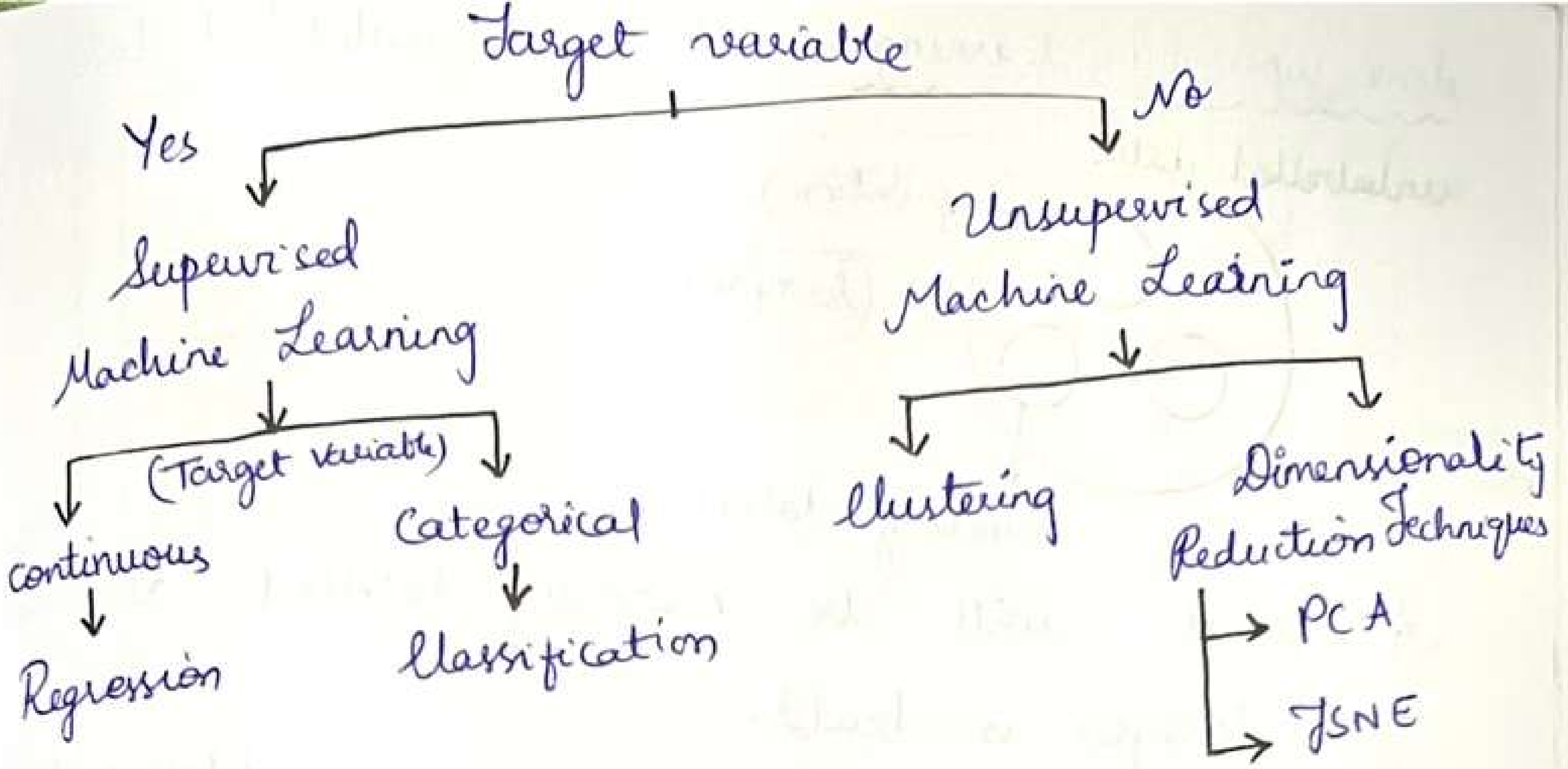
→ In the tabular data, we have independent variables, also known as the input variables, and the target variable, known as the output variable. (dependent variable).

→ In some cases, the Target varable is not mentioned (or) not present.

→ We have different approaches (or) learning techniques depending on availability of target variable.

→ If the Target variable is present / specified we call it as a 'labelled data.'

→ If the target variable is not present / not specified, we call it as an 'unlabelled data.'



Our intuition will always be that the ML Model has to learn the 'pattern' of the data.

The entire row in the tabular data is referred as 'record'.

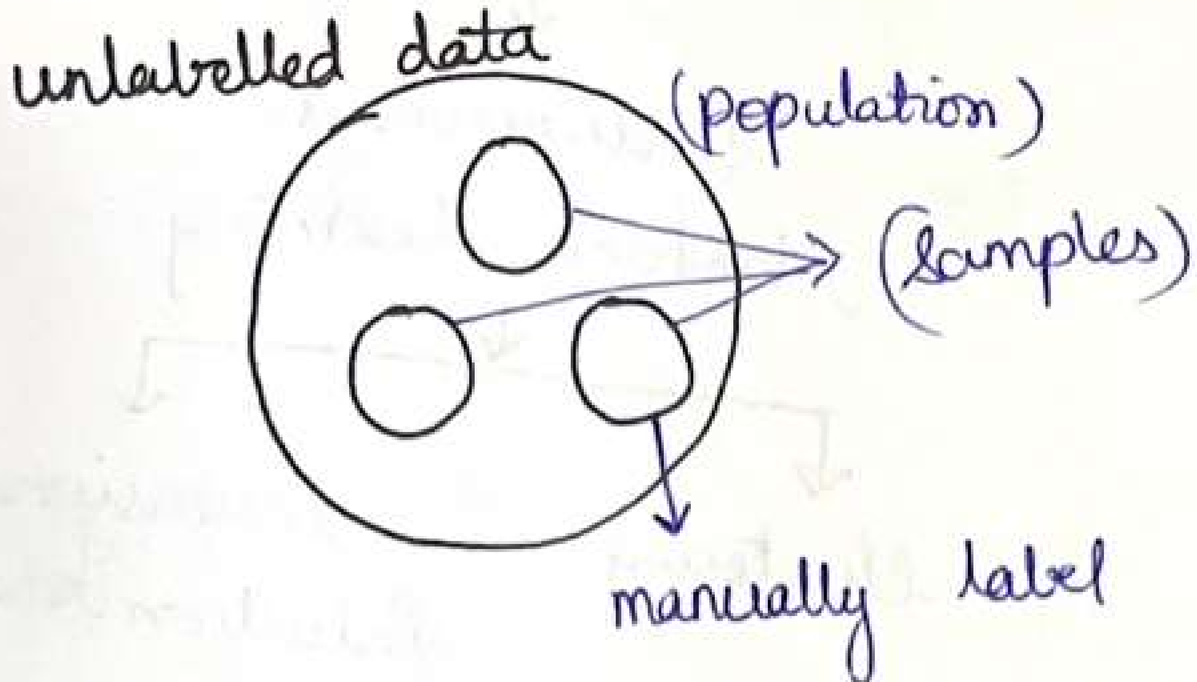
Dimensionality Reduction Techniques :-

Each feature variable is considered as a dimension.

More dimensions => Slow learning / no learning.

Apart from Supervised and Unsupervised ML, we also have Semi-supervised ML and Reinforcement Learning.

Semi-supervised learning:- We have unlabelled data



A sample will be manually labelled and the classifier is build.

And then using this classifier and labelled data ~~to~~ we perform semi-supervised learning.

Reinforcement Learning:-

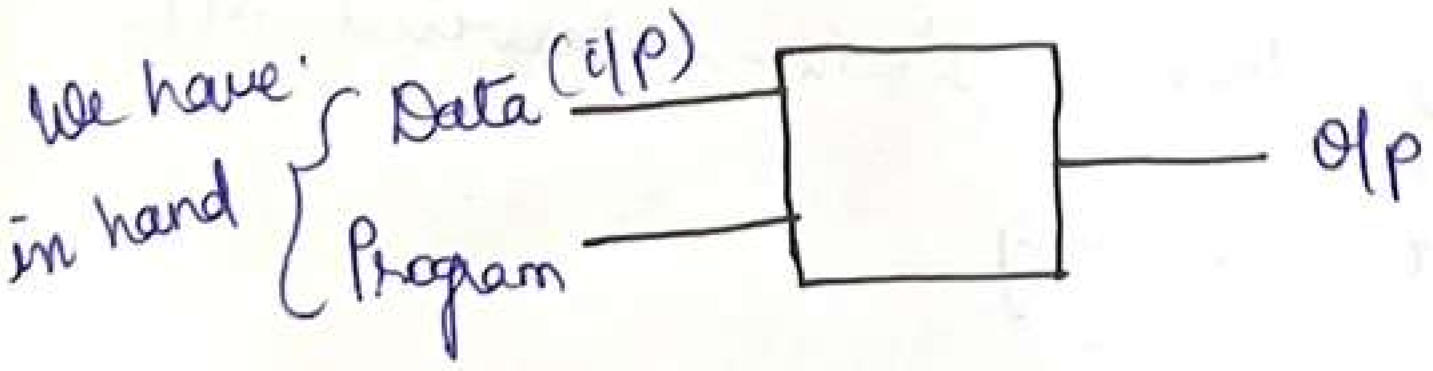
Happens based on Rewarding system.

The rewards are based on the achievment of task or not.

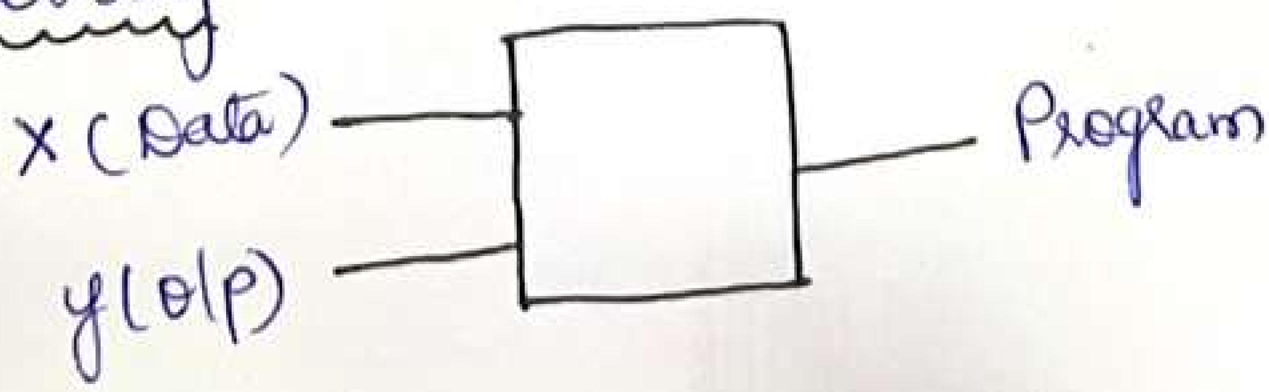
* We will not delve into semi-supervised and reinforcement learning in the due of the course *

4|08|2023

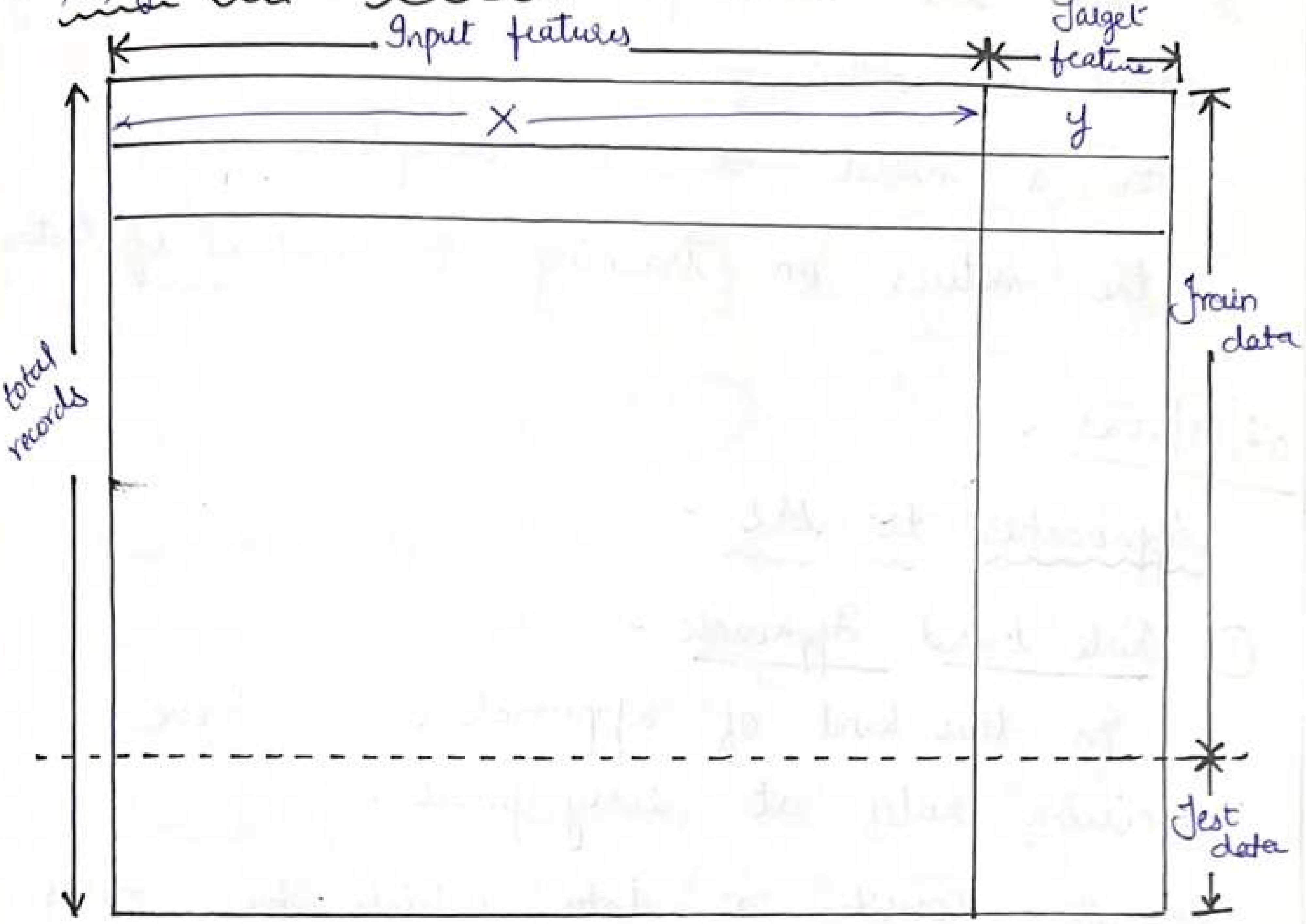
Traditional programming:-



Machine learning:-



X, y and Train, test :-



In general, the complete data is splitted into train and test by 80 : 20 ratio.

If the target is categorical, then in the test data further we do a 15% and 5% split for testing and validation purposes.

Also, we go for Stratified splitting preferably for categorical target variable.

If any new datapoint is introduced to the model, where it doesn't follow the characteristics of training data, then the model can missclassify the datapoint.

What the model learns during the training are called as 'parameters'.

To know the model performance, we have something called as metrics.

For a model to be acceptable →

the metrics on training ≈ metrics of testing.

07/08/2023

Approaches to ML :-

① **Rule-based Approach** :-

In this kind of approach, we have decision rules at every point.

Example:- Consider a data which has marks related to Math, Physics and Chemistry subjects, we need to predict the result whether Fail / First Class with Distinction / First Class / Second class

There are certain assumptions that can be made. Like -

≥ 32 → Pass , < 32 → Fail

≥ 80% → FCD

≥ 60% → FC

≥ 40% → SC

< 40% → Fail

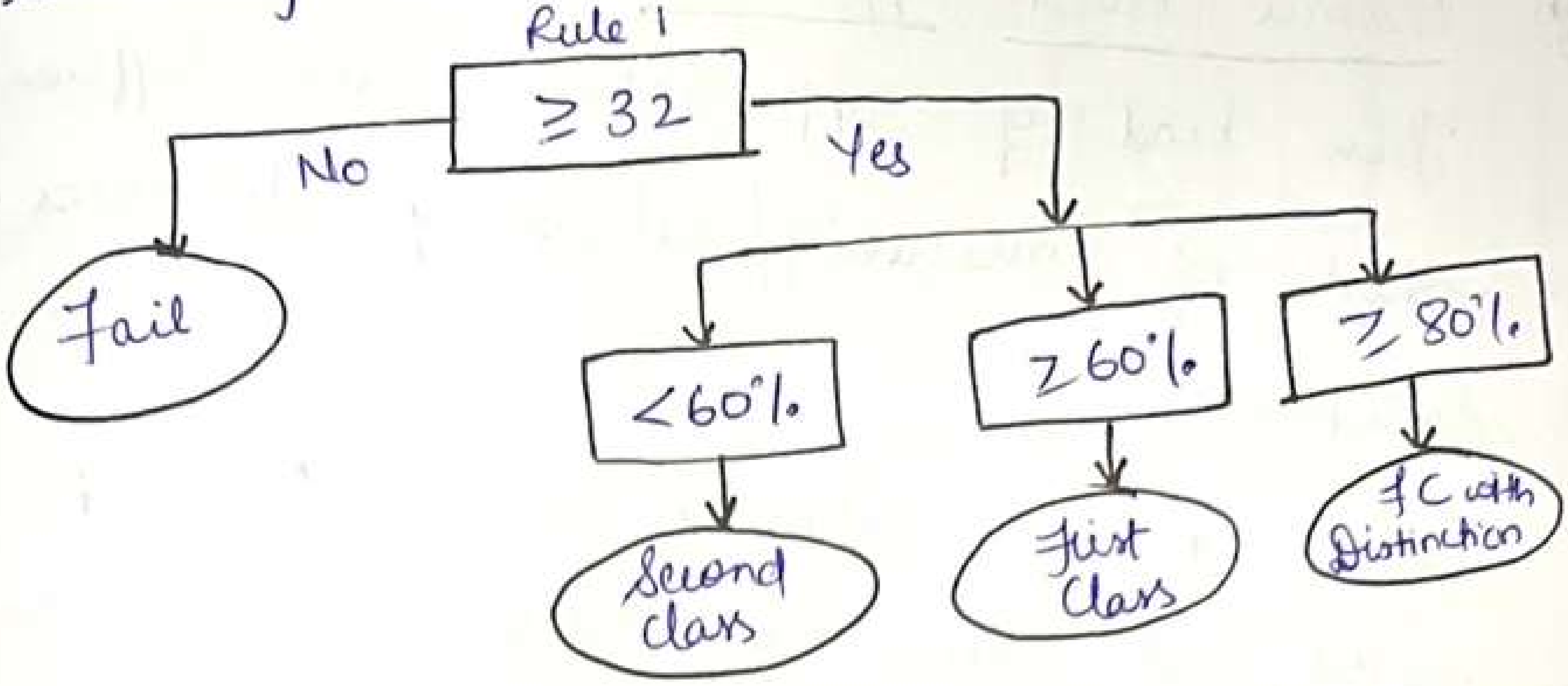
Math	Physics	Chemistry	Total	Result
100	30	100	230	Fail

X - independent features

feature - generated fom existing input features.

y - Target feature

Structuring the rules -



This kind of structure where there are decision rules at every point / every layer is called "Decision trees".

Sometimes, it becomes highly difficult to use all the input features, as the algorithm can become highly complex.

In such situations, we do something called as feature engineering, where we create / generate a new feature from existing features.

This helps in reducing complexity of the algorithm and also results in higher accuracy of the model.

From decision trees, we go down to other rule-based approaches like-

Random Forests (multiple decision trees)

and

Ensemble methods

② Distance - Based Approaches :-

These kind of approaches uses different methods of measuring/ calculating distances between data points.

When a new datapoint is introduced to the model, it checks how close is the point to certain group or other datapoints.

The different types of distances are:-

- Minkowski
- Euclidean
- Manhattan

Using Minkowski distance and altering the p-value we can calculate the other dimensions as well.

Steps involved in distance-based approach are:-

1- Calculate the distance between given data point from the other datapoints in the dataset.

2- Sort the calculated distances in ascending order.

3- (For classification tasks) - from the sorted distances, find the k^{th} nearest neighbours and classify the target of the data point based on the label of the high frequency of those neighbours.

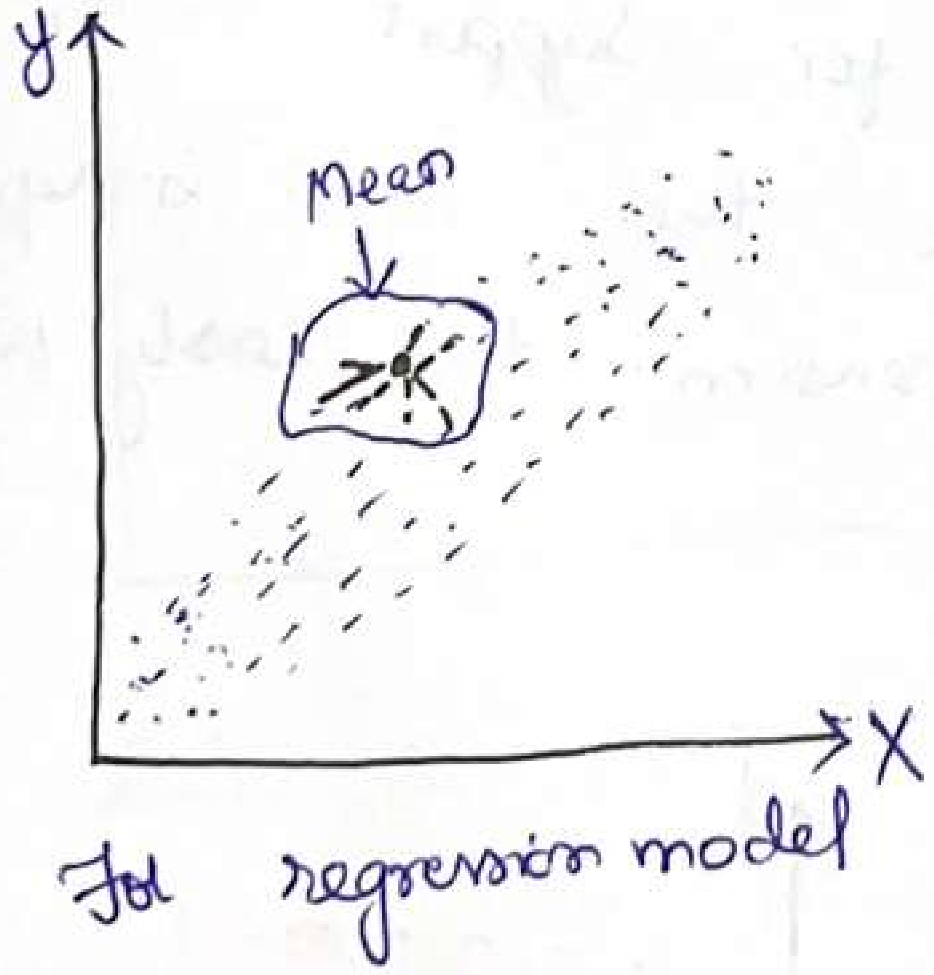
(for regression tasks) - take the mean of the k^{th} nearest neighbours and assign it to the target.

The best number of neighbours we could select is $\sqrt{n}$ rounded to the nearest odd number.

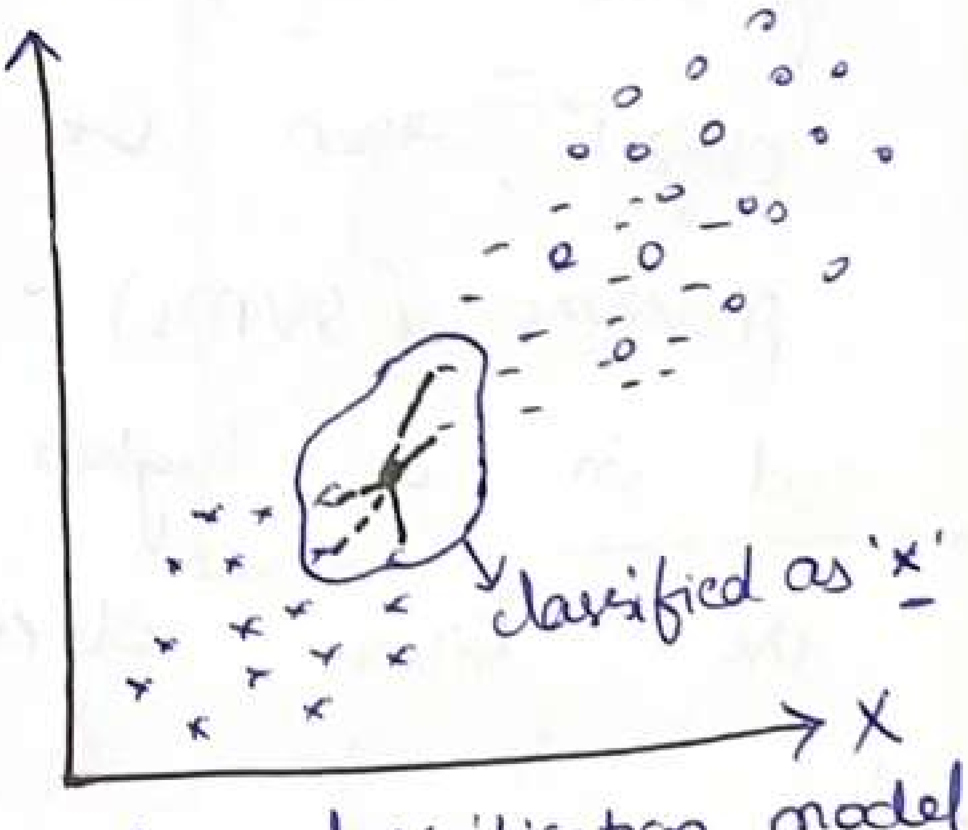
(n → sample count from the training data).

If $\sqrt{n}$ is even and there is a equal division of the classes, then the target might be misclassified.

And when $\sqrt{n}$ is odd, there is a pretty bit chance that one class frequency is higher than the other.

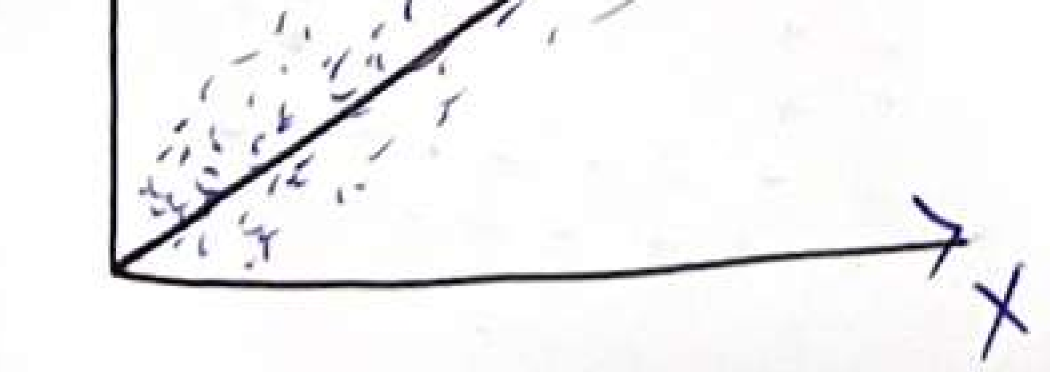


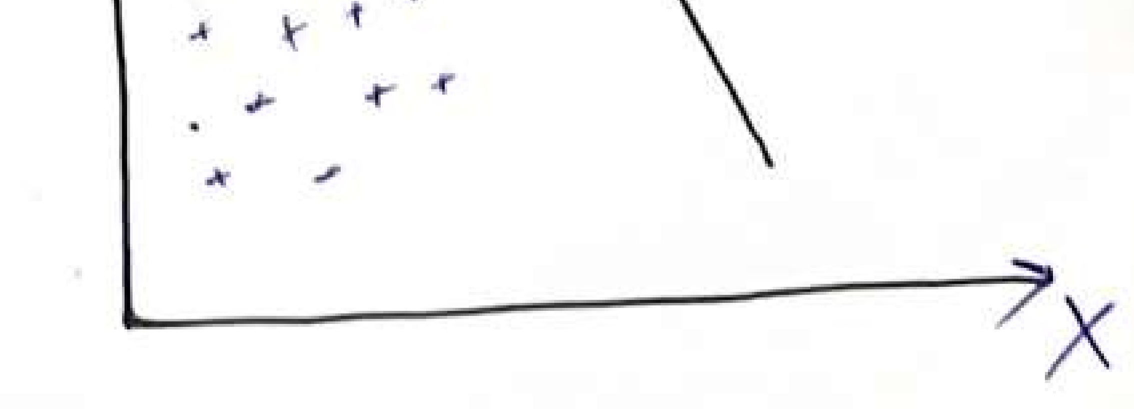
For regression model



For classification model

③ <u>Boundary - Based Approaches:-</u>





these kind of approaches is achieved by fitting a linear object - (in regression) through most of the densly packed datapoints and (in classification) to separate the defined classes.

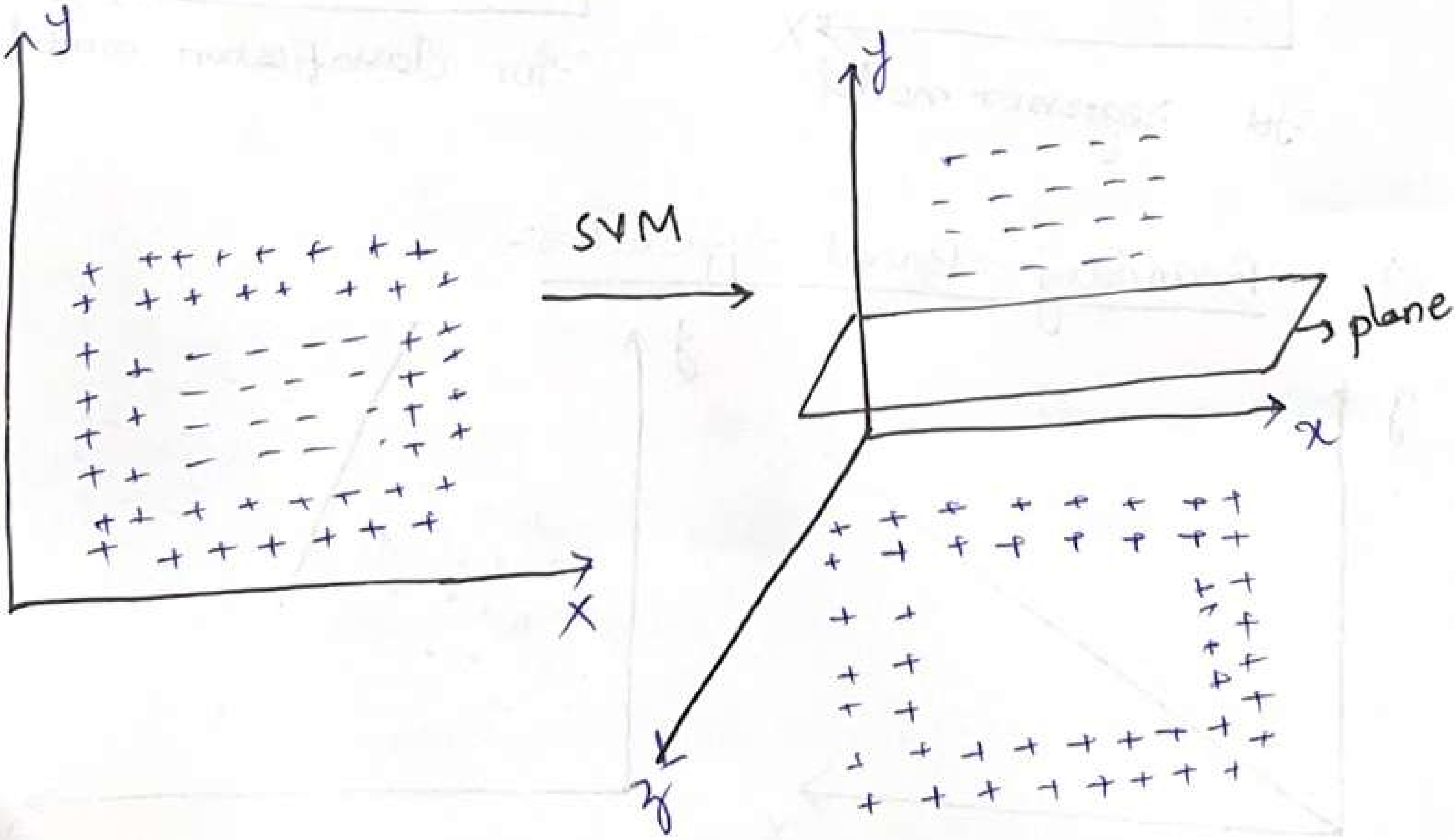
The linear object can be a line in 2D, plane in 3D and hyperplane in 4D and >4D.

Boundary - Based Approach

(Regression) Linear Regression

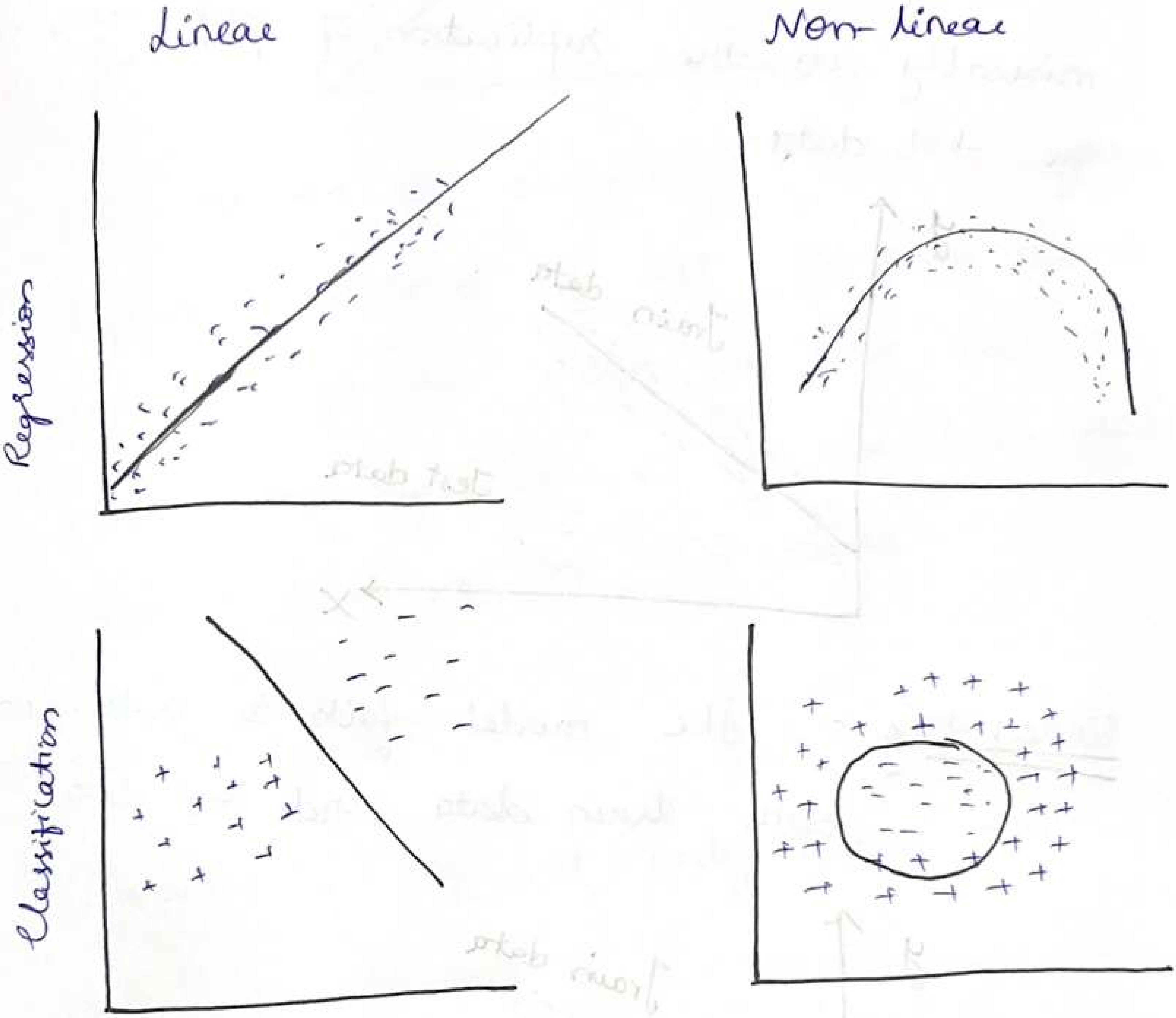
(Classification) Logistic Regression

If the data, initially, doesnot fit a linear object, then we go for Support Vector Machines (SVMs) where the data is represented in a higher dimension to easily fit the linear object.



SVMs can be applied to both (linear) regression models and classification models.

Kinds of data:-

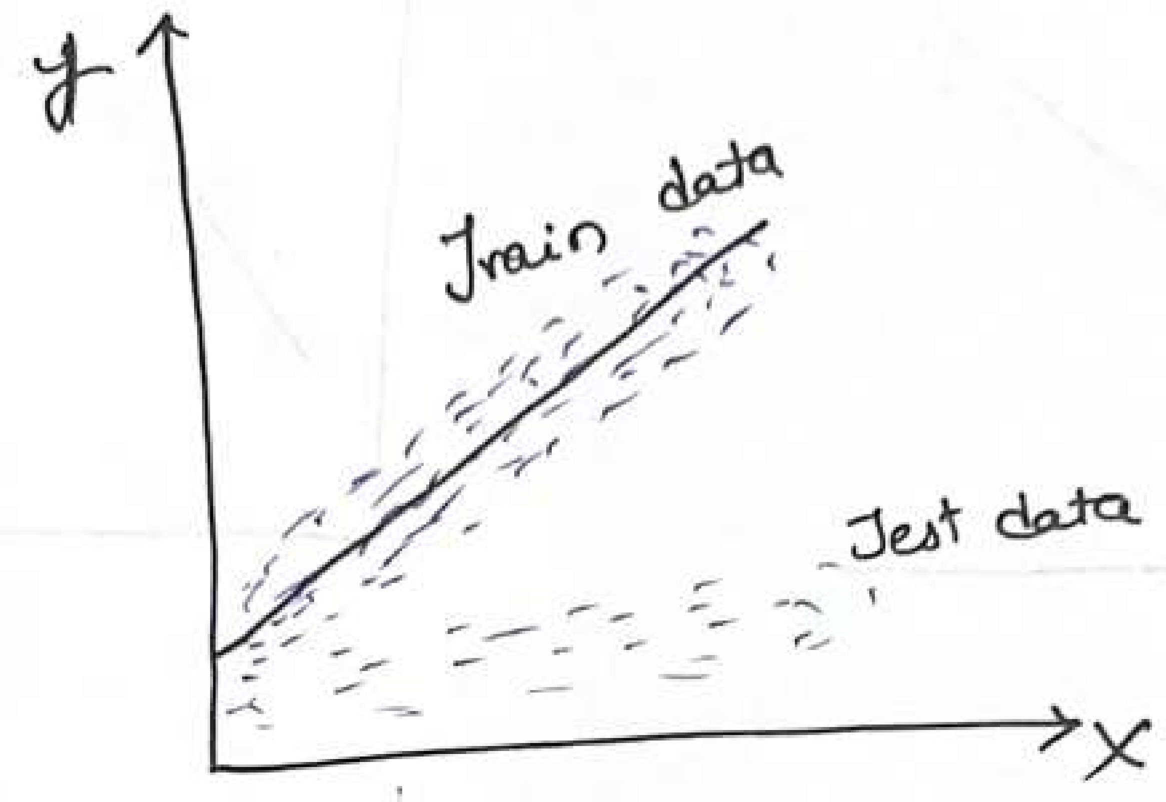


Data is said to be linear if it can fit a linear object to perform necessary tasks. If the linear object couldnot be fitted then the data is non-linear.

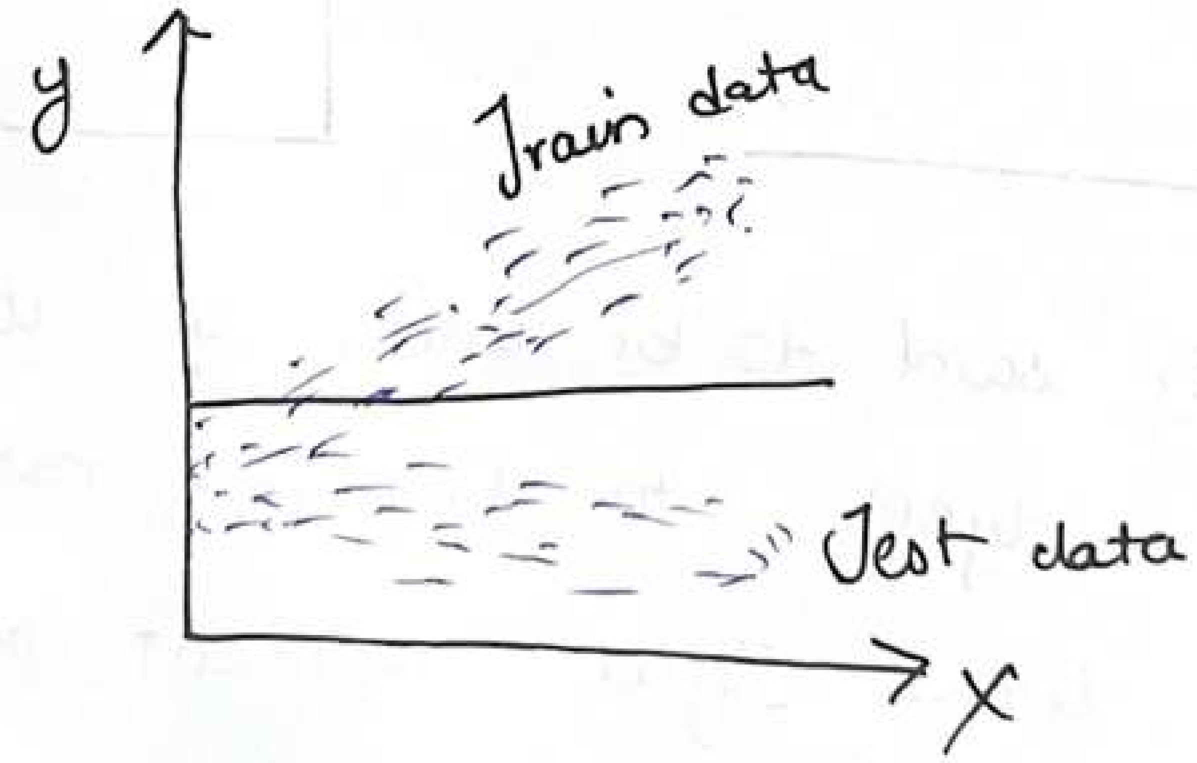
* we can transform non-linear data to linear data using feature engineering.

Three diffrent cases of model perfolmance -

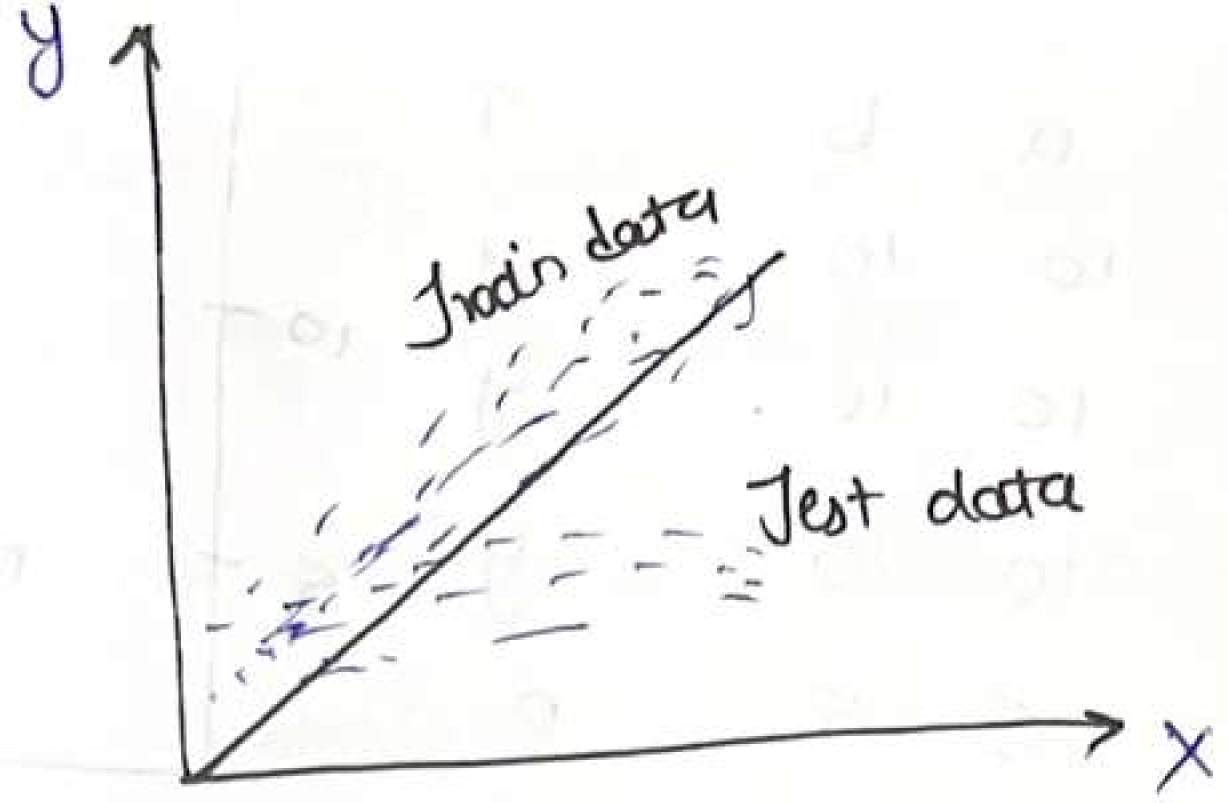
Overfitting :- The model perfoms exceptionally well on train data and fails miserably on the replication of perfolmance on the test data.



Underfitting :- The model fails to perfolm on both train data and test data.



Best / Optimal fit :- The model replicates optimal metrics for both train data and test data.



The problem of overfitting and underfitting happens because of not choosing appropriat approach for the data we have and the model learning so well on the train data that fails on the test data.

08|08|2023

* Model is efficient only when the data is efficient *

Data preprocessing:-

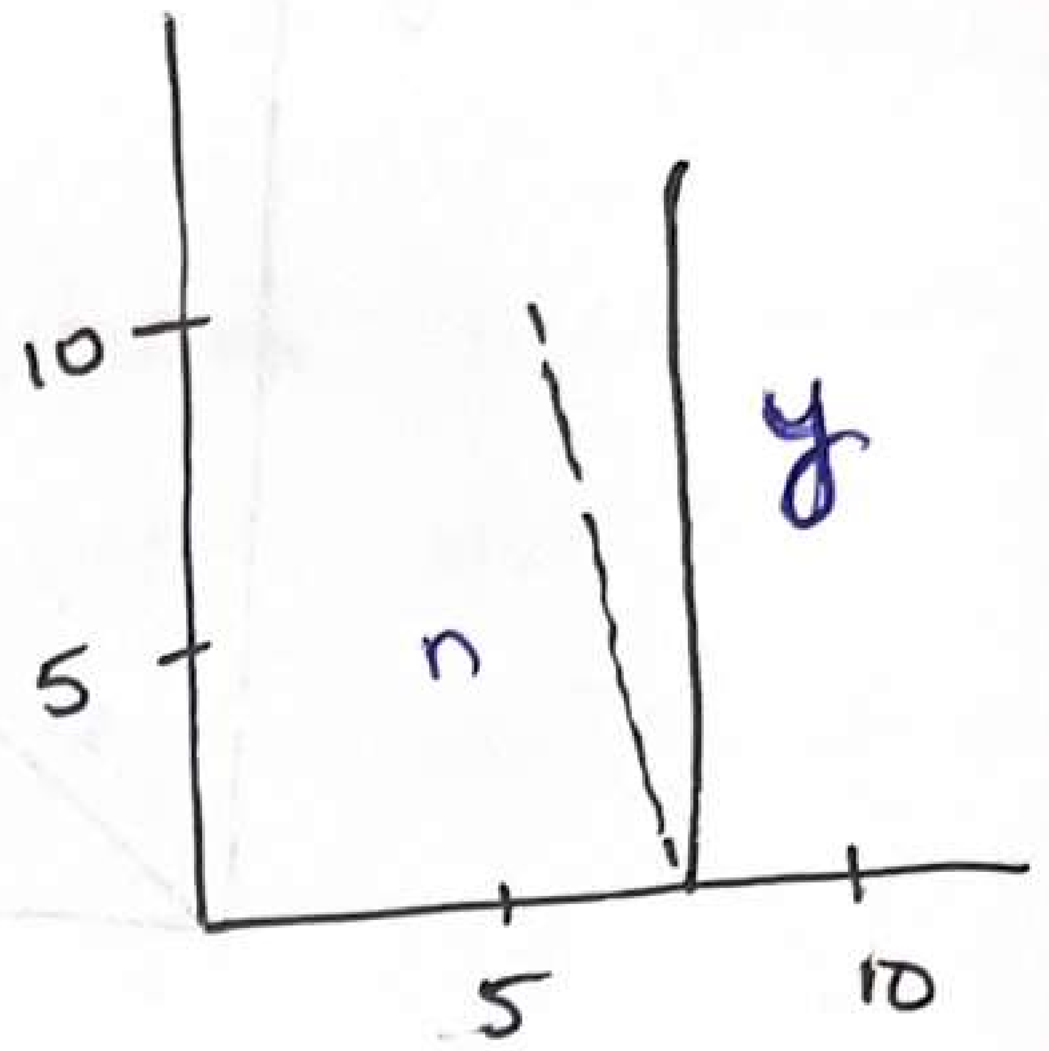
Step 1:- Dedeuplication (dropping the duplicates):-

Problem with having the duplicates in the data is they tend to affect the linear object by tilting it towards high density points.

The model also learns more from the duplicates.

Example:-

a	b	T
10	10	y
10	10	y
10	10	y
5	5	n



---- expected boundary

—— boundary affected by duplicate values

Step 2 :- Imputing missing values :-

It is not desirable to have missing values in our data as missing values = missing data. This can lead to poor performance of the model.

While imputing the missing values, it is to be taken care that the mean and standard deviation of the feature should remain same or near to same before imputing and after imputing.

Primary reason for this is to maintain the distribution of the column as we will be seeing similar data in the future.

Step 3 :- Outliers / Anomalies handling:-

It is important for us to detect and handle outliers and anomalies in our data as they can have an affect on our approaches to the learning and thereby affecting the perfolmance of the model.

Steps ① ,② and ③ of data preprocessing helps us to understand what to do in our next steps. (feature selection).

→ Data Transformation :-

In our data, we can have continuous and categorical variables.

It is important for transforming our data to a desirable feormat before giving this data to the model.

Categorical variables Encoding :-

Example:-

a	b	c
(cont.)	(cont.)	(cat.)
10	10	x → 2nd high preference
7	7	y → low preference
5	5	z → high preference

If the categorical data is ordinal, then the values are provided / given to class based on preference.

This type of encoding is called as "Ordinal encoding".

If the categorical data is nominal, then we do some-thing called as '**One-hot encoding**'

Example:-

b	c
5	x
3	y
2	z

⇒

b	c_x	c_y	c_z
5	1	0	0
3	0	1	0
2	0	0	1

Results in multicollinearity between the x-features

Drop one - one hot encoding :-

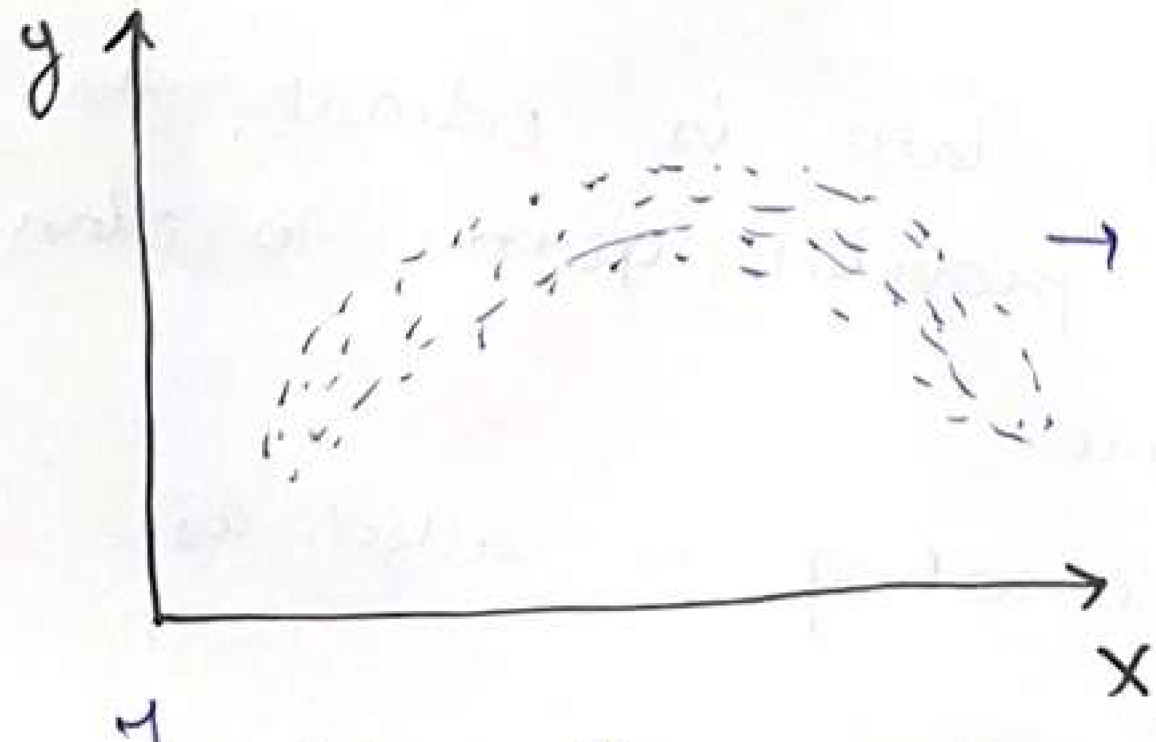
drop one of the features from c_x, c_y, c_z.

Preferably 'c_x'

b	c_y	c_z
5	0	0
3	1	0
2	0	1

no multicollinearity

Continious variables transfolmation :-



→ Stretch the data to the right so that the data forms a near to linear data

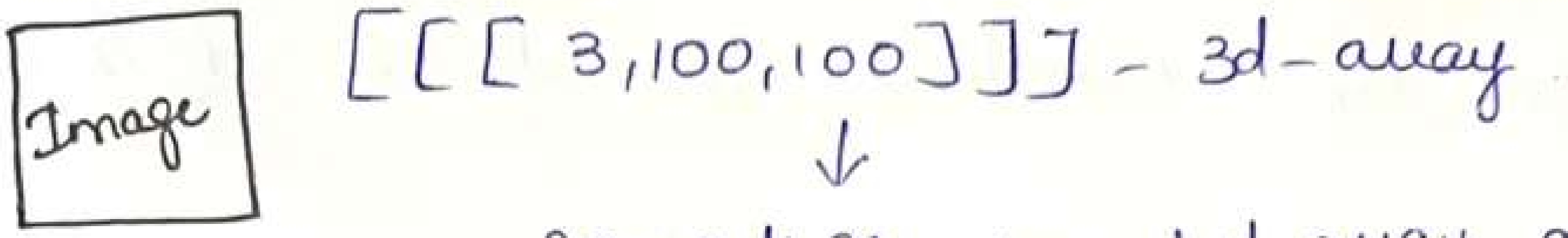
Transfolm the continuous data involves Converting Non-normal data to normal data and non-linear to linear data.

To do this we have power transformer and functional transformer.

In functional transformer we can build user-defined transformer.

Image Data :-

Images are represented as ndarrays.



converting to 1-d array and assinging it in a record / entry

← X →	y	
← Image data →	cat	class 1
	dog	class 2

We use, PIL → Python Image Library, for performing preprocessing of Images.

Initial preprocessing techniques are -

resize, cropping, color correction, sharpness.

Text Data :-

We have traditional approaches and deep learning approaches.

Traditional approaches → Bag of words, n-grams, Tfidf vectors

Deep learning approaches → word 2 vector, glove embeddings, Bert embeddings.

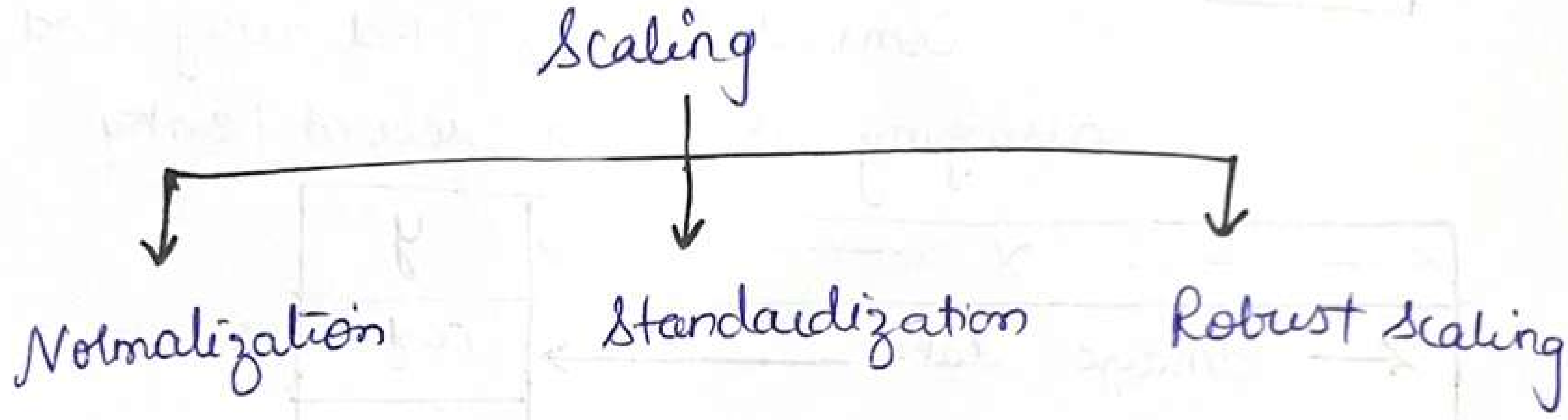
09/08/2023

Feature scaling :-

→ We need to avoid weighing one column more than the other column.

→ We achieve this by scaling our continuous data.

Machine learning models tend to favour those data features where magnitude is higher. All the variables should be of same scale.



Preprocessing techniques which are applied on historical training data, same learning has to be applied on the test data / unseen data.

Step 1 :- Split the data to X & y variables and then to Train and test.

The train & test split can be done according to the data we have at hand.

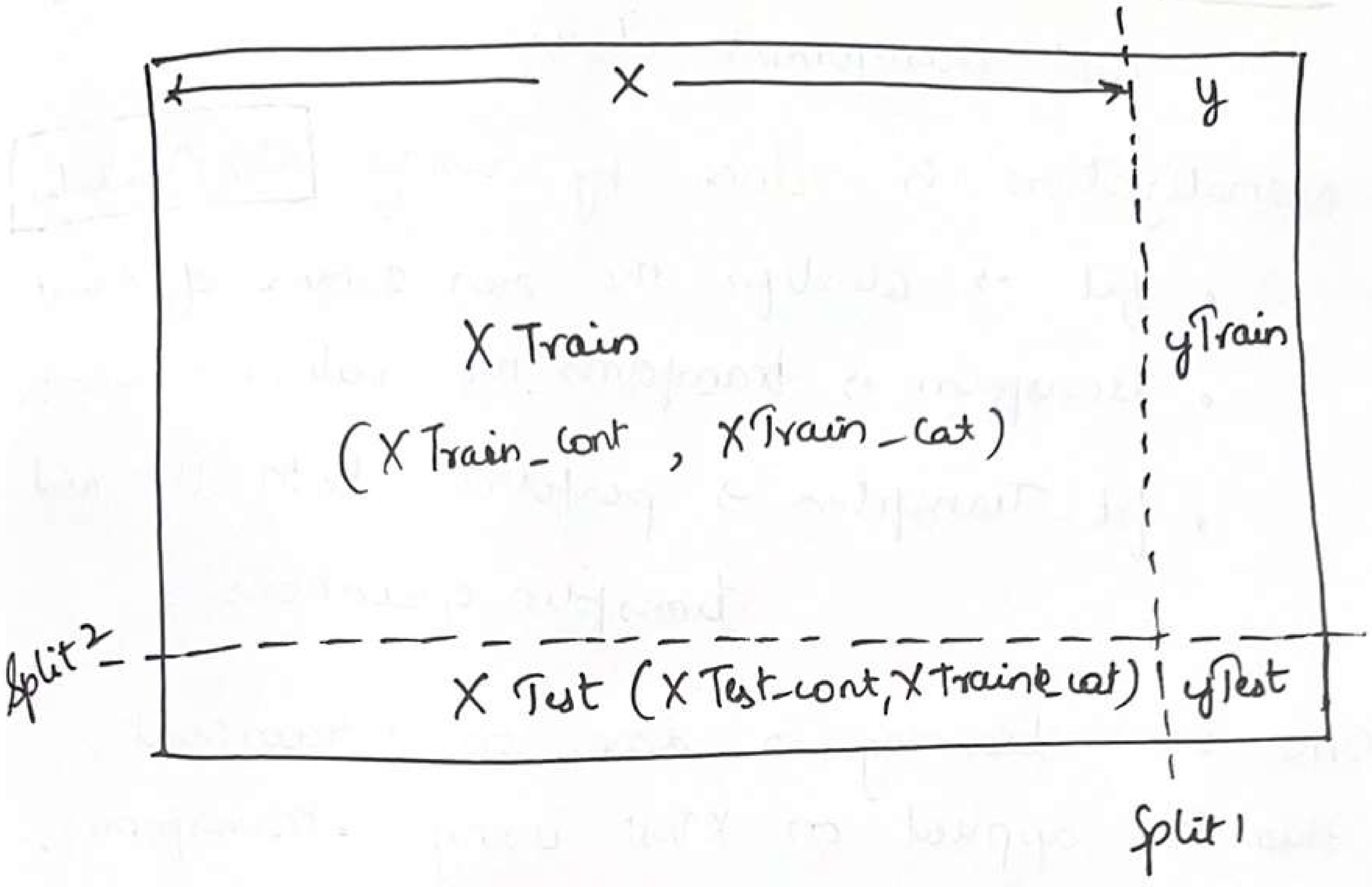
Step 2 :- Split the X_Train, X_test to

X_train_cont   X_test_cont

X_train_Cat   X_test_cat

Step 3 :- We scale our X_Train_continuous with appropriate scaling technique (.fit_transform).

Step 4 :- We apply this learning from scaling X_Train_cont on the X_Test_cont (. transform)

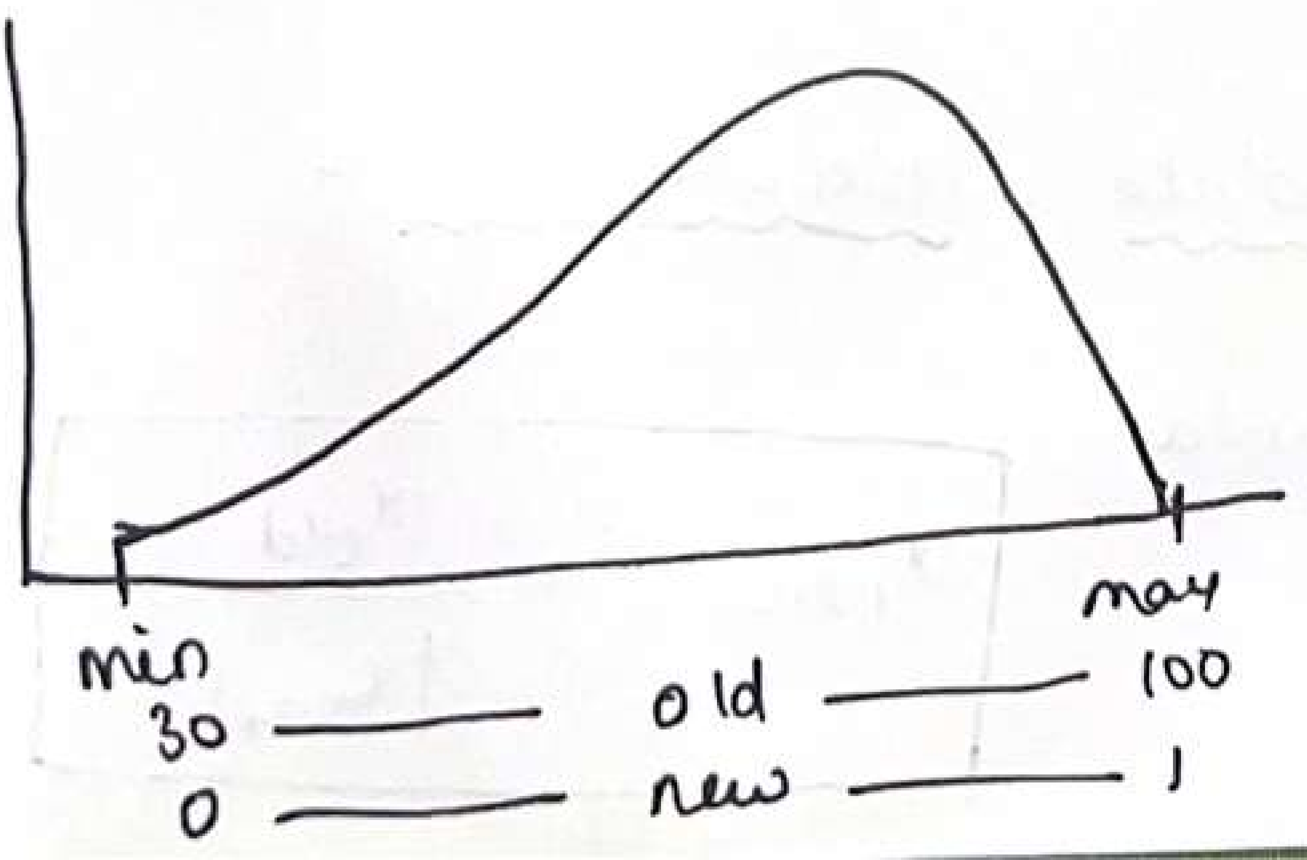

X
y
X Train
(X Train_cont , XTrain_cat)
yTrain
Split 2
X Test (X Test_cont, X Traine cat)
yTest
Split 1

* Normalization :-

$$\boxed{\text{formula } (x_{new}) = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}}$$

The data is scaled down such that it ranges between 0 & 1 (both included).

0 being minimum value and 1 being maximum value.



The distribution is shifted to the origin at the first quadrant, no matter where the distribution was (-ve or +ve values of x).

'sklearn' - library contains all preprocessing and transformer classes.

Normalization is done by using $\boxed{\text{MinMaxScaler}}$

- fit → identifies the min & max of X cont.
- transform → transforms the values between 0&1.
- fit_transform → performs both fit and transform operations.

Once the learning is done on X train cont, this is applied on X Test using (.transform).

There are other types of normalization.

→ Mean normalization :-

The distribution of the transformed data is centered over the 'mean'. ⇒ mean = 0, range = 1

Formula

$$\boxed{x_{new} = \frac{x_{old} - \bar{x}}{x_{max} - x_{min}}}$$

→ Max-absolute normalization :-

Formula

$$\boxed{x_{new} = \frac{x_{old}}{|x_{max}|}}$$

The new values range from [-1 to 1] and the mean =0.

* Scaling will not alter the distribution, rather it will only scale the x-axis.

* **Standardization :-**

We rely on standardization as it scales the data such that, mean =0 and standard deviation =1.

Formula,

$$x_{new} = \frac{x_{old} - \bar{x}}{S_x}$$

$\bar{x}$ → mean sample

S_x → standard deviation of sample

* **Robust Scaling :-**

We use this kind of scaling when we have outliers.

Formula,

$$x_{new} = \frac{x_{old} - Q_2}{Q_3 - Q_1}$$

$$= \frac{x_{old} - x_{median}}{IQR}$$