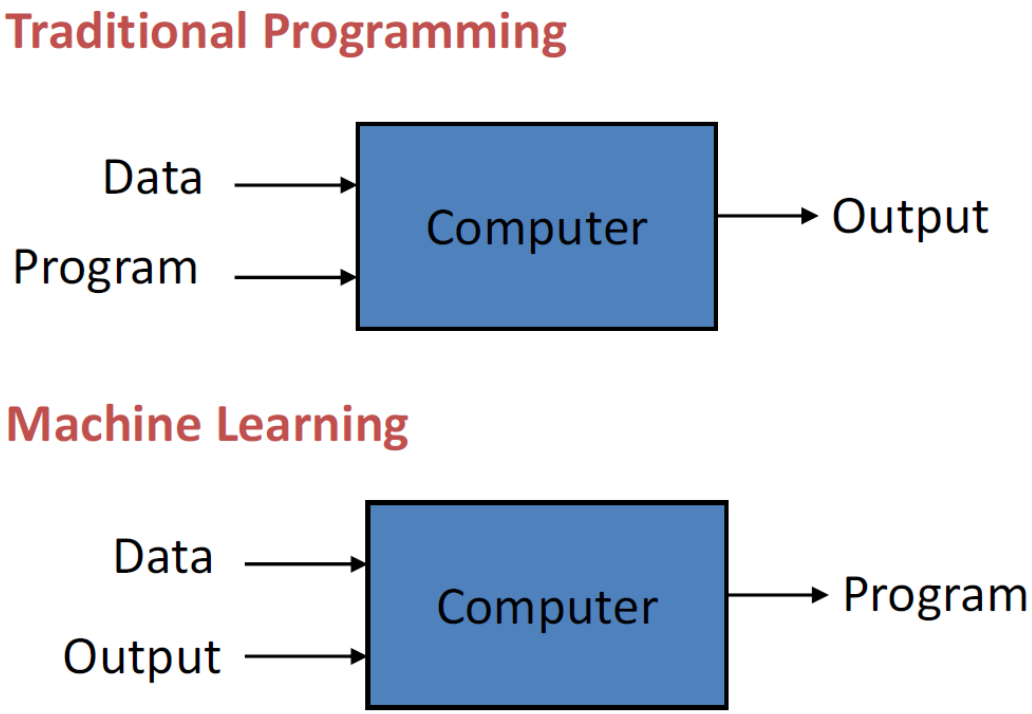
Introduction to Machine Learning

Introduction

- Machine learning models are computer programs that are used to recognize patterns in data or make predictions.
  - *Machine learning is a subset of artificial intelligence (AI) that focuses on building systems that learn from data and make decisions or predictions without being explicitly programmed for every task.*
  - *Instead of writing specific rules or algorithms for every task, machine learning models are trained on data and improve their performance based on experience.*

Machine Learning (ML)

"A computer program is said to learn from experience (E) with some class of tasks (T) and a performance measure (P) if its performance at tasks in T as measured by P improves with E"

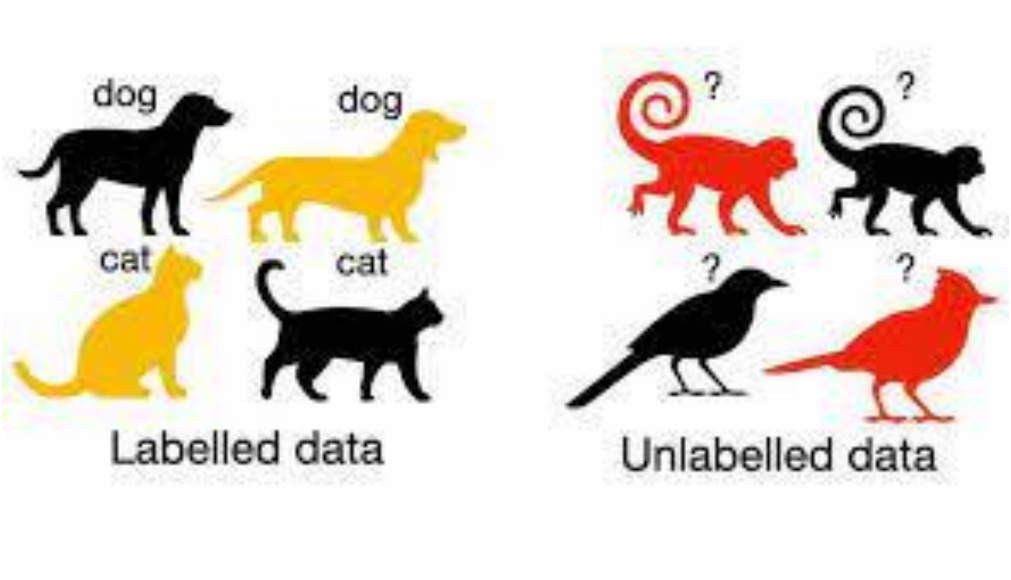


Types

- Supervised Learning
- Unsupervised Learning
- Semi supervised learning
- Reinforcement Learning

Supervised Learning

- Two phases: training, testing
- An algorithm learns to map input data to a desired output based on labeled examples (supervisor) provided in the training dataset.
- The algorithm uses features from the dataset to determine relationship between the input and the target.
- Terms
  - Training set, testing set
  - Model
  - Evaluation



Supervised Learning: Types

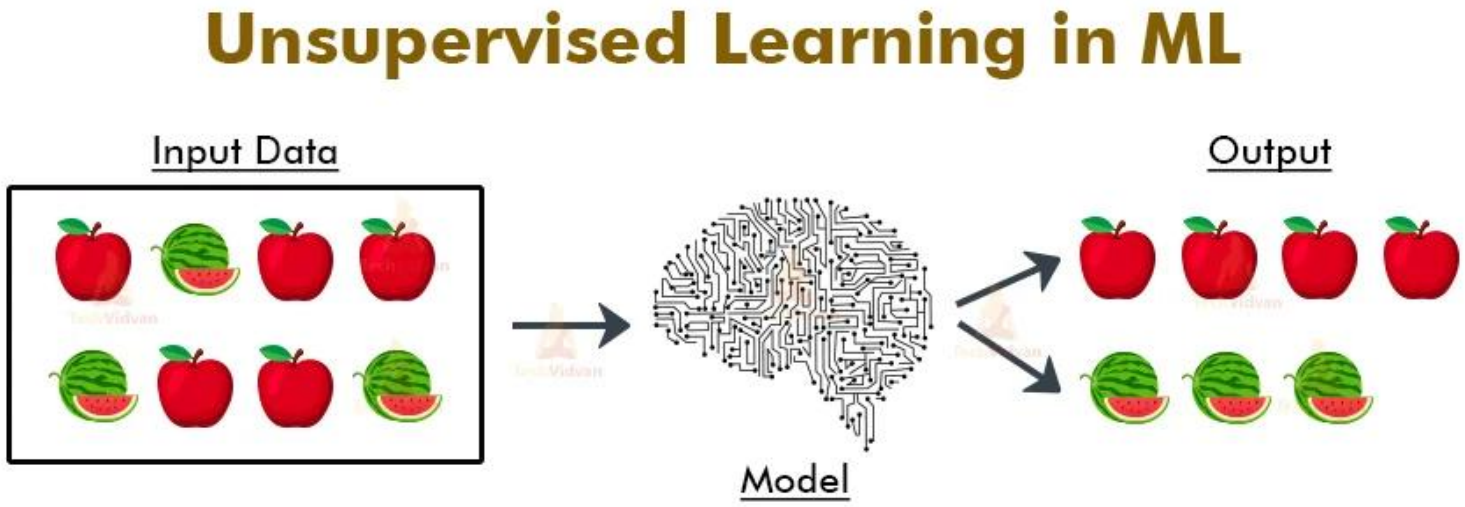
- Regression
  - the model predicts a continuous output.
  - E.g. predicting house prices based on features like size, number of bedrooms, and location.

- Classification
  - the model assigns data points to predefined classes.
  - E.g., classifying emails as spam or not spam.

Algorithms: Linear regression, decision trees, support vector machine

Unsupervised Learning

- Algorithm is trained on unlabeled data to discover patterns within the data.
- No predefined target outputs or labels.



Linear Regression

- Linear regression is used for modeling the relationship between a dependent variable (target) and one or more independent variables (predictors or features) by fitting a linear line to the observed data.
- The aim is to find the best fitting line that describes the relationship.
- Types
  - Simple Linear Regression
  - Multiple Linear Regression

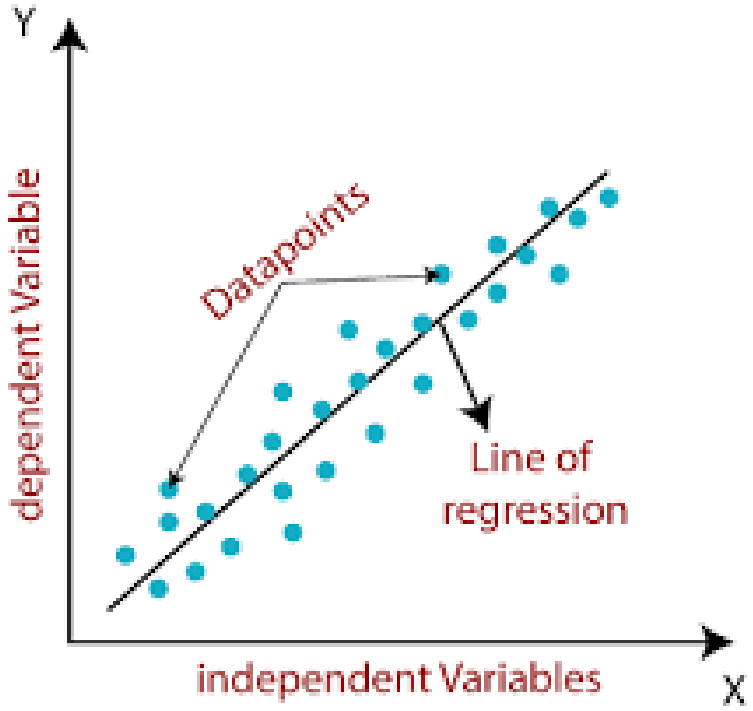


Fig. Simple Linear Regression

Simple Linear Regression

- One independent and one dependent variable.
- Simple Linear regression model is represented by the following equation: -

$$Y = \beta_0 + \beta_1 X$$

where, β_0 – y intercept, β_1 – $slope$

X – independent variable

Y – dependent variable

- The slope and the intercept of the best fitting line determines the relationship between the variables.

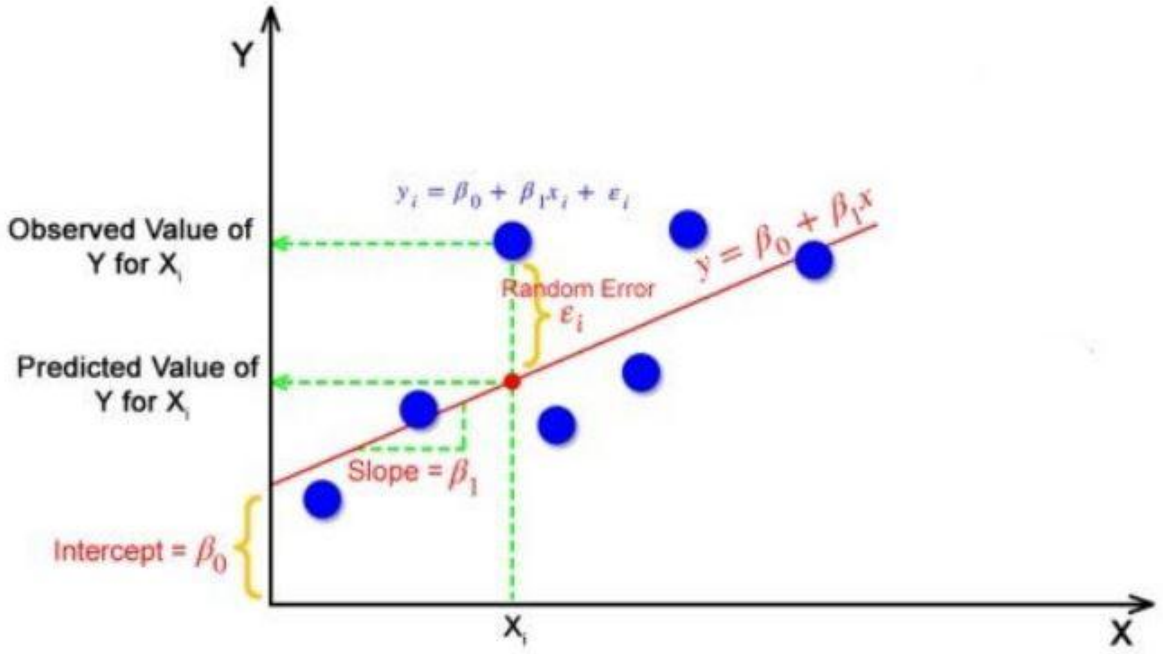
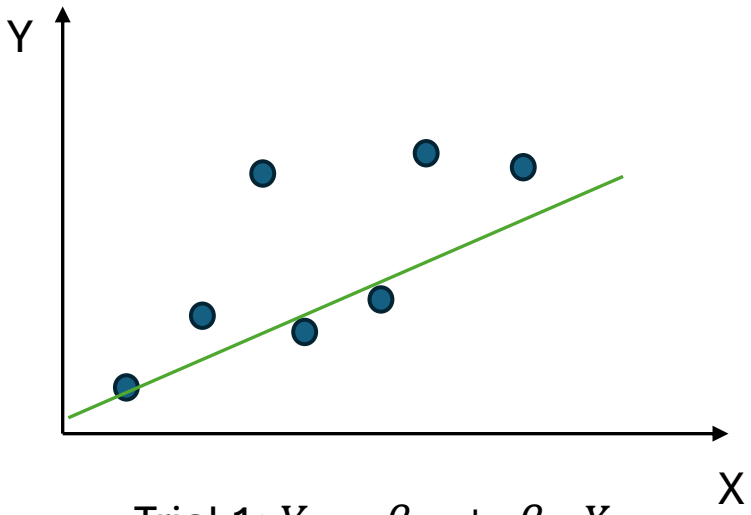
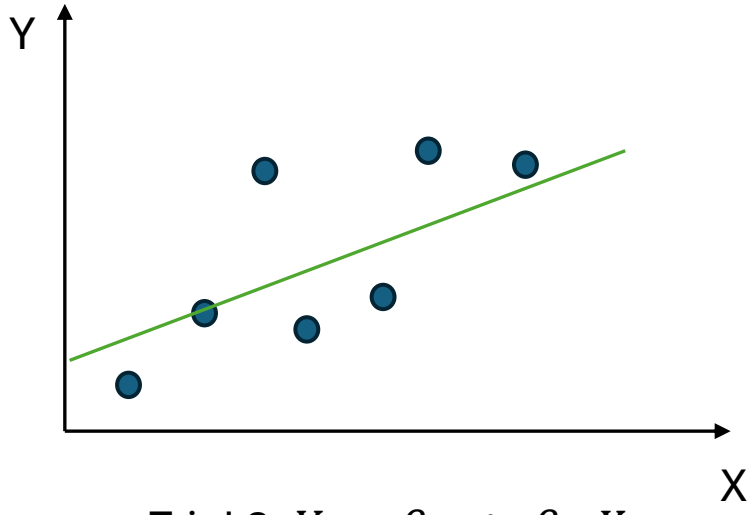


Fig. Simple Linear Regression

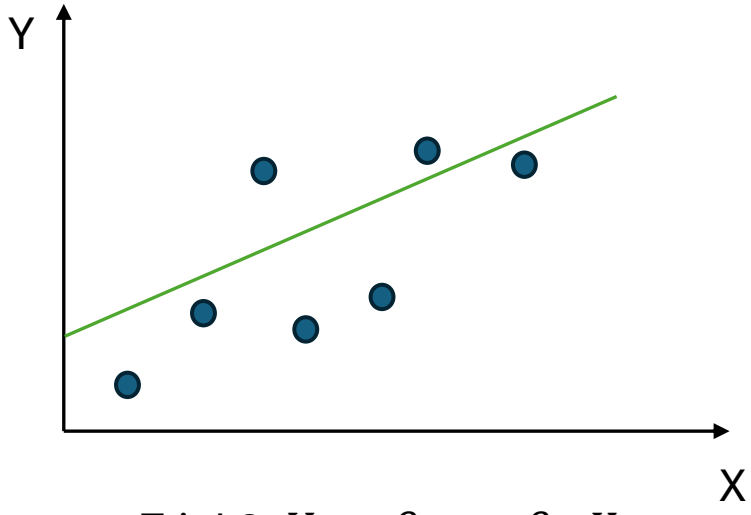
Simple Linear Regression: How is the best fitting line found?



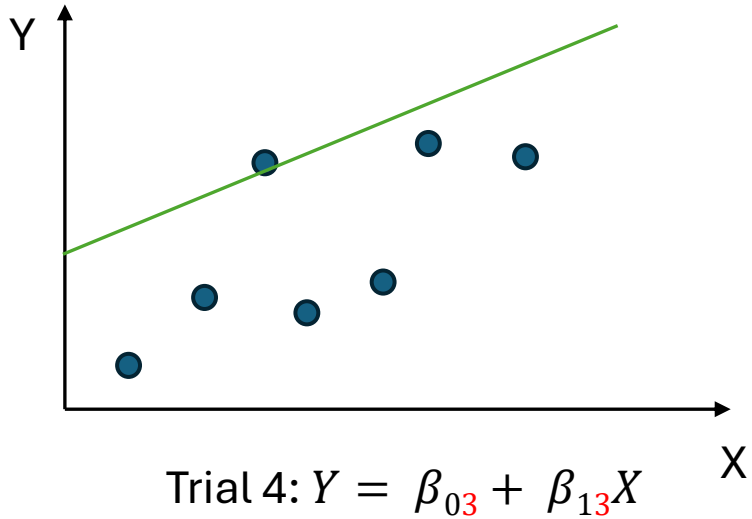
Trial 1: $Y = \beta_{01} + \beta_{11}X$



Trial 2: $Y = \beta_{02} + \beta_{12}X$



Trial 3: $Y = \beta_{03} + \beta_{13}X$



Trial 4: $Y = \beta_{03} + \beta_{13}X$

- Multiple trails of fitting a line are conducted, each time with a different line
- Error between the actual and predicted value is computed. The line with the least error is the best fitting line.

Simple Linear Regression: Residual Sum of Squares

- Residual = $y_i - y_i'$,
  - where, y_i = actual value, y_i' = predicted value
- Mean Squared Error (MSE) = $\frac{1}{n}\Sigma (y_i - y_i')^2$

$$= \frac{1}{n}\Sigma (y_i - (\beta_0 + \beta_1 x_i))^2$$

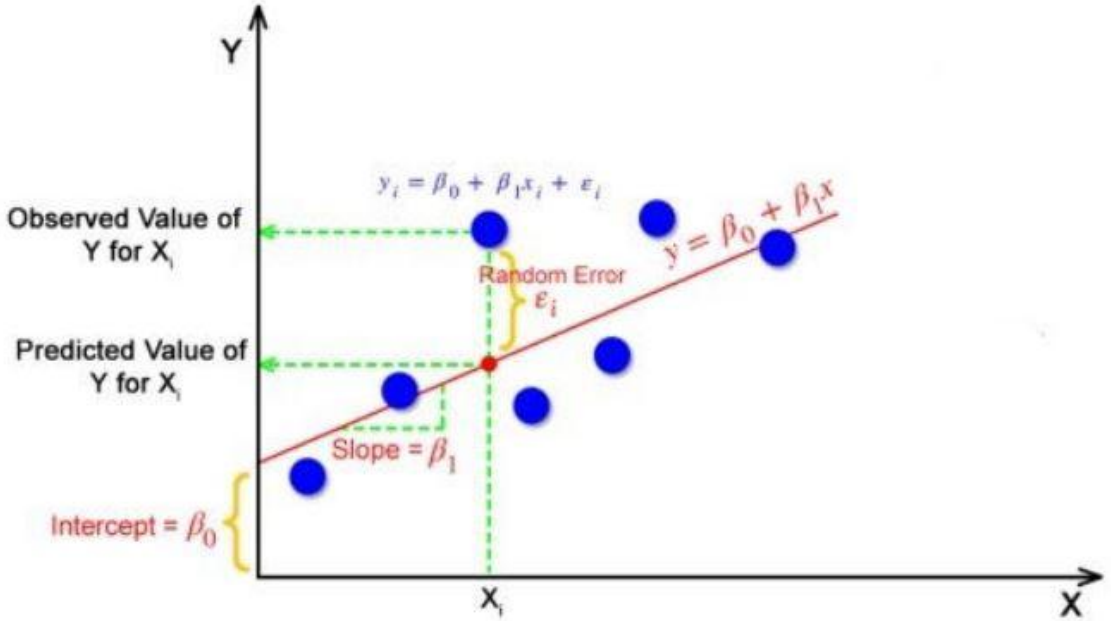


Fig. Simple Linear Regression

Link to video:
https://www.youtube.com/watch?v=owI7zxCqNY0&ab_channel=dataminingincae

Linear Regression: House Price Prediction

Area (sq. ft.)	Price (lakhs)
500	12
650	13.5
720	16.5
750	17
780	17.2
850	19
880	21
1050	24
1200	29
1550	41

Data set

80-20 split

Area (sq. ft.)	Price (lakhs)
500	12
650	13.5
720	16.5
750	17
780	17.2
850	19
880	21
1050	24

Training set

Area (sq. ft.)	Price (lakhs)
1200	29
1550	41

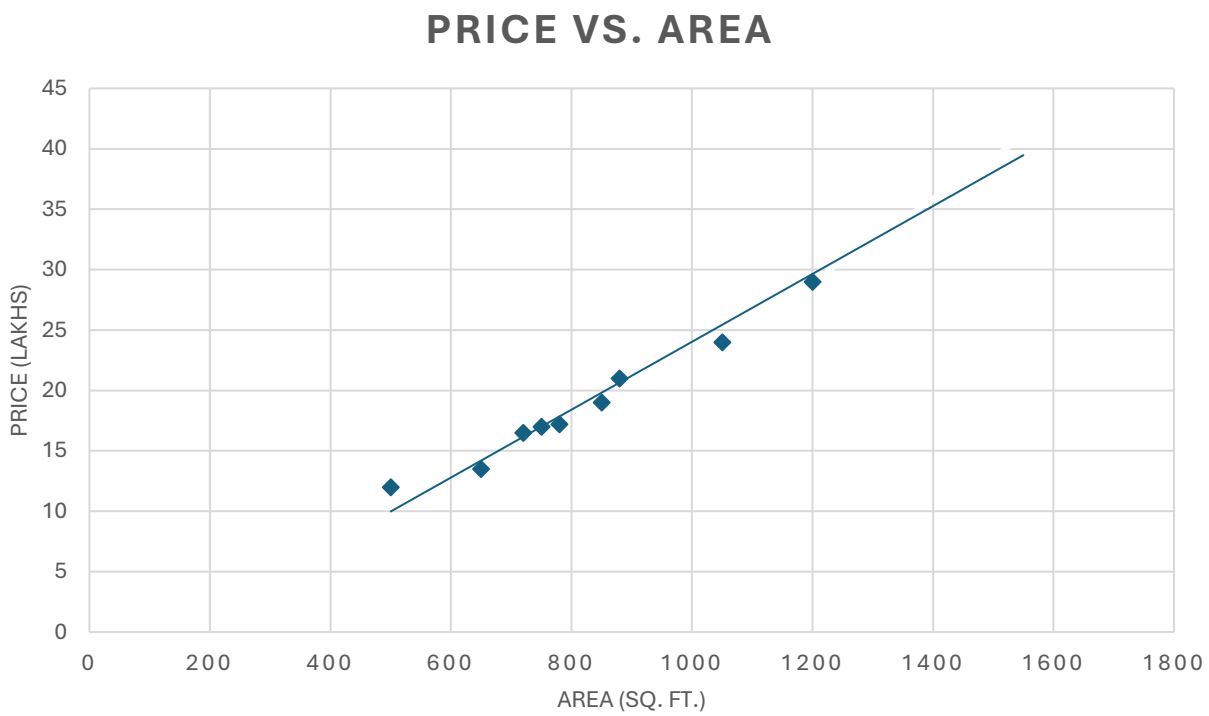
Testing set

- Training set is used for training (finding the best fit line).
- Testing set is used for evaluating the trained model.

Linear Regression: House Price Prediction (Training)

Area (sq. ft.)	Price (lakhs)
500	12
650	13.5
720	16.5
750	17
780	17.2
850	19
880	21
1050	24

Training set



- Best fitting line found by adjusting the slope (β_1) and y intercept (β_0).
- Suppose β_0 = 3 and β_1 = 6 are the determined values, then the equation of the best fitting line is $Y = 3 + 6X$

Linear Regression: House Price Prediction (Testing)

Area (sq. ft.)	Price (lakhs)
1200	29
1550	41

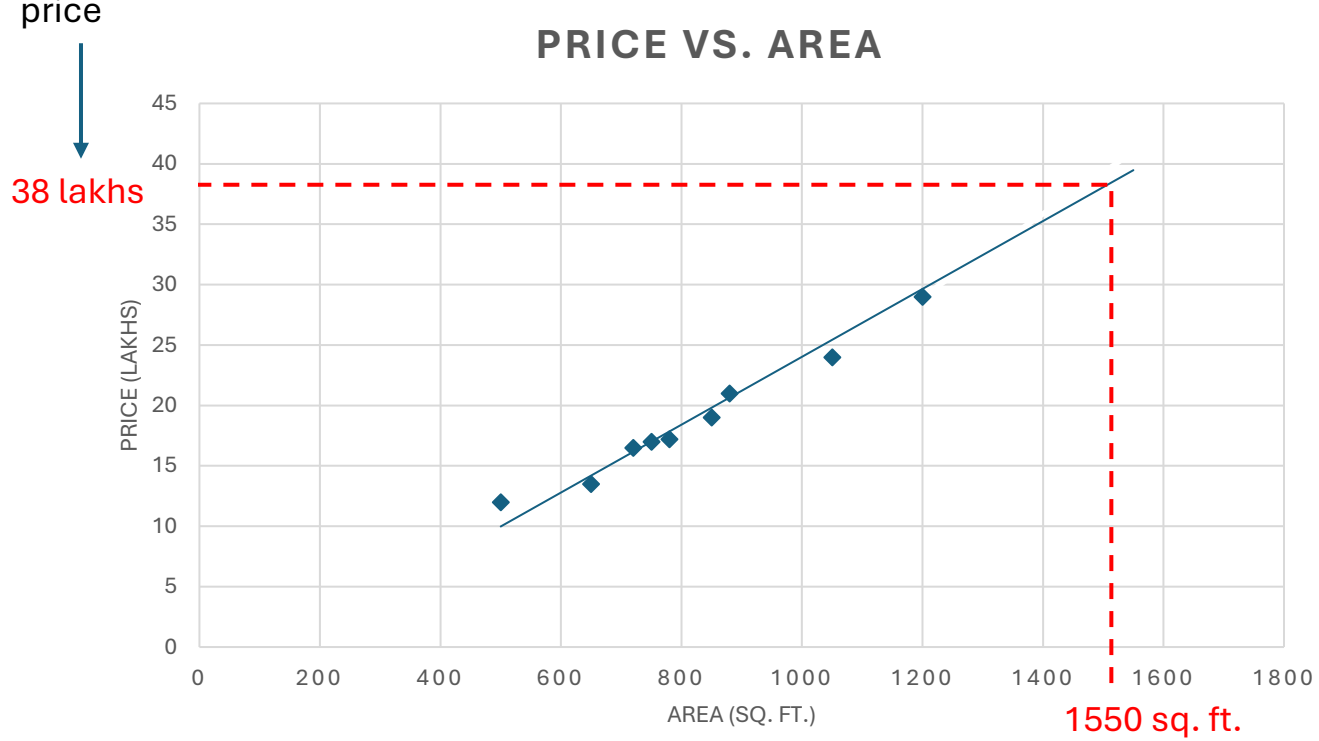
Testing set

Actual price

$$Y = 0.242x + 0.51$$

Predicted price

38 lakhs



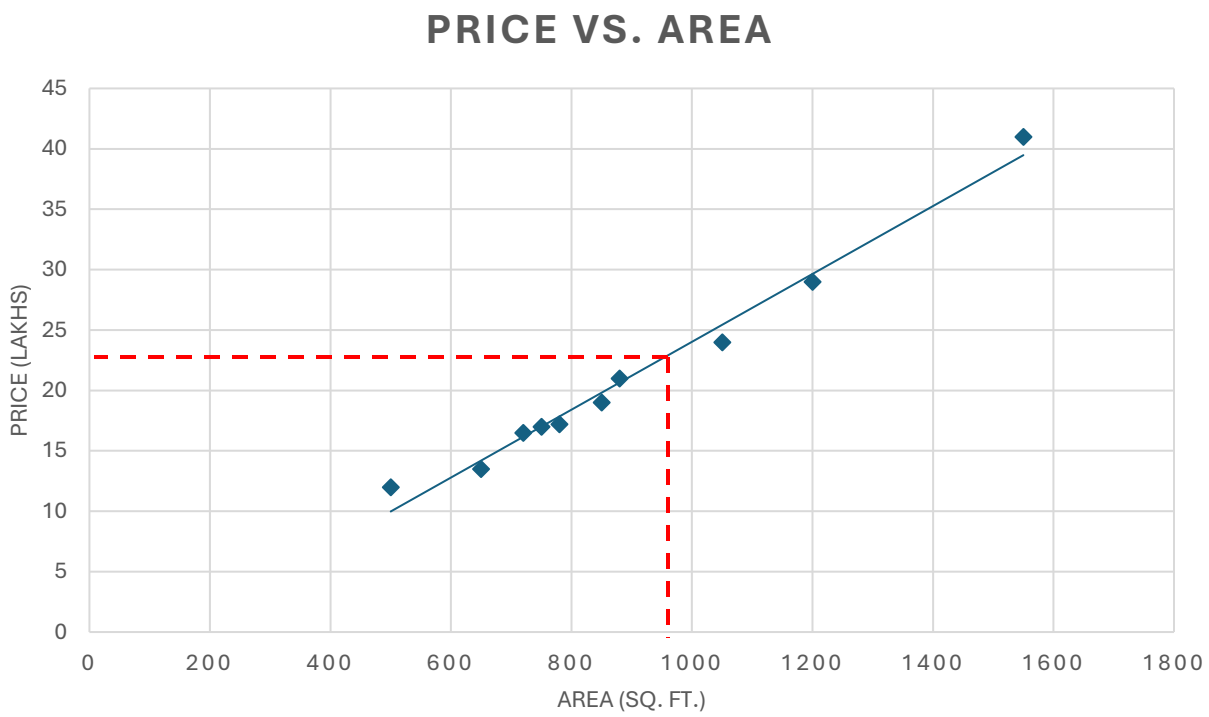
Price for the examples in the testing set predicted using the regression line.

Linear Regression: House Price Prediction

Prediction phase

Area (sq. ft.)	Price (lakhs)
500	12
650	13.5
720	16.5
750	17
780	17.2
850	19
880	21
1050	24
1200	29
1550	41

Data set

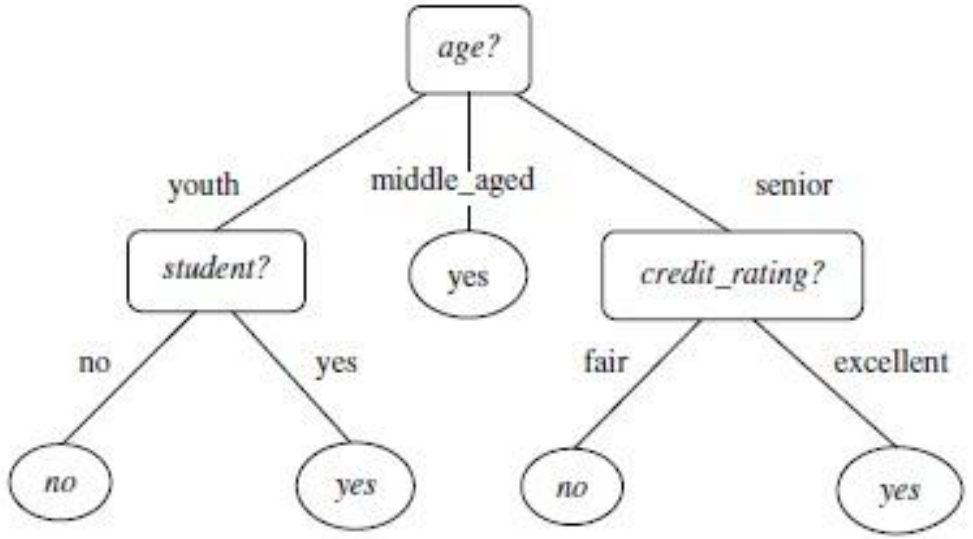


Price of a house with area of 900 sq. ft.?

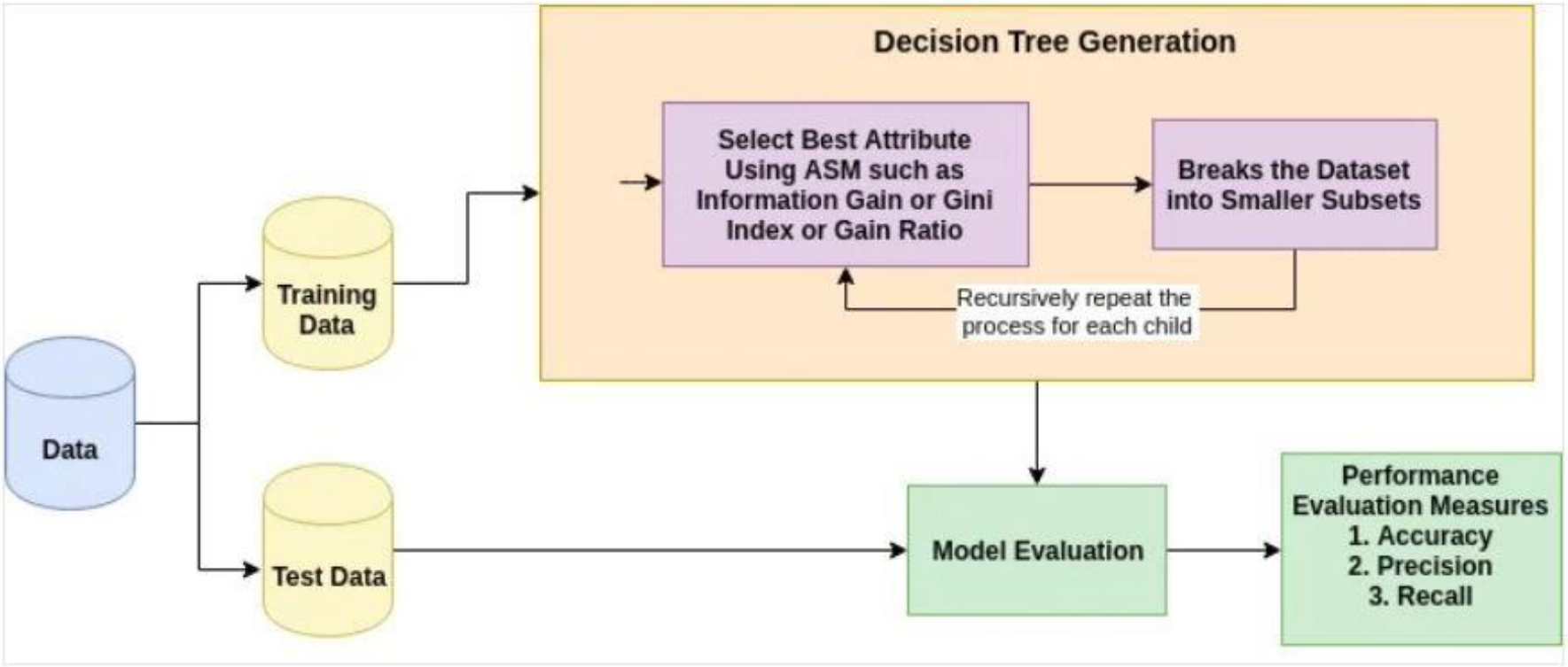
Decision Tree

- A Decision Tree is a supervised learning algorithm used for both classification and regression tasks.
- It works by splitting the data into subsets based on the most significant feature, creating a tree structure where each internal node represents a decision based on a feature, each branch represents the outcome of that decision, and each leaf node represents the final prediction.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



How does a Decision Tree work?



Training phase: Decision Tree Induction

Decision tree induction refers to the process of generating a decision tree from a given data set.

1. Select the Best Feature to Split:
    - Choose the feature that results in the largest gain in information.
2. Split the Dataset:
    - Divide the dataset into subsets based on the chosen feature and condition.
3. Repeat for Subsets
    - Apply the same process on each subset to create further decision nodes and leaf nodes, until:
        - All data points are correctly classified.
        - A stopping criterion is met (e.g., maximum depth, minimum samples in a leaf).

Training on the sample dataset

- To generate a decision tree from the dataset, the following steps are followed: -
  - Select one of the features *(age, income, student, credit_rating)* to be placed at the root.
  - Based on the feature selected, create *'x'* branches from the root, where *x* – number of unique values of the feature.
  - Follow the same process to place other features at the internal nodes.
- The leaf nodes will have the target classes.

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Fig. Dataset

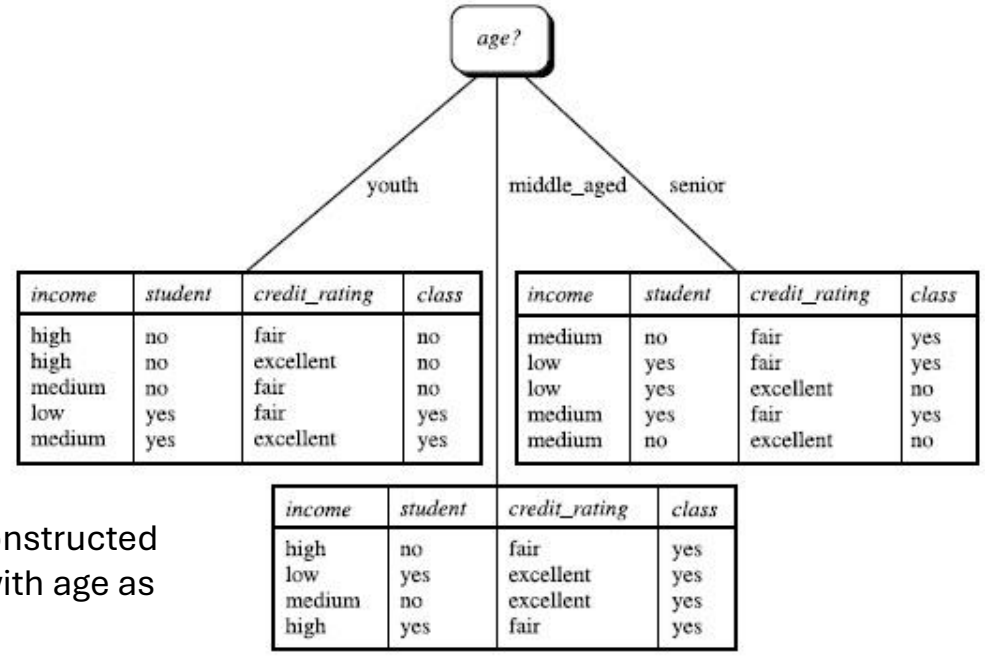


Fig. Partially constructed decision tree with age as the root node

Training phase: Attribute selection measures

- How was age selected to be placed at the root, out of the features (age, income, student, credit_rating)?
  - An attribute selection measure can be used to determine the best attribute.

- Attribute selection measures that can be used: -
  - Information gain
  - Gain ratio
  - Gini index

Information Gain

- Information gain is a metric that measures how much information is gained by splitting a data set on a particular feature.
- Information gain is calculated by comparing the entropy of the original dataset to the entropy of the child sets. A higher information gain indicates that the feature is more effective at splitting the data.
- The entropy of the original dataset (D) is given as follows: -

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

where, p_i – probability that that a tuple in D belongs to class C

- The entropy of the child set is given as follows: -

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

where, $|D_j|$ – number of samples in subset j
$|D|$ - total number of samples in the dataset
$Info(D_j)$ – entropy of subset D_j

- The information gain is given as follows: -

$$Gain(A) = Info(D) - Info_A(D).$$

Information gain calculation

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Entropy of dataset

buys_computer: yes

buys_computer: no

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

Entropy of the child set created when split on the attribute *age*

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$= 0.694 \text{ bits.}$$

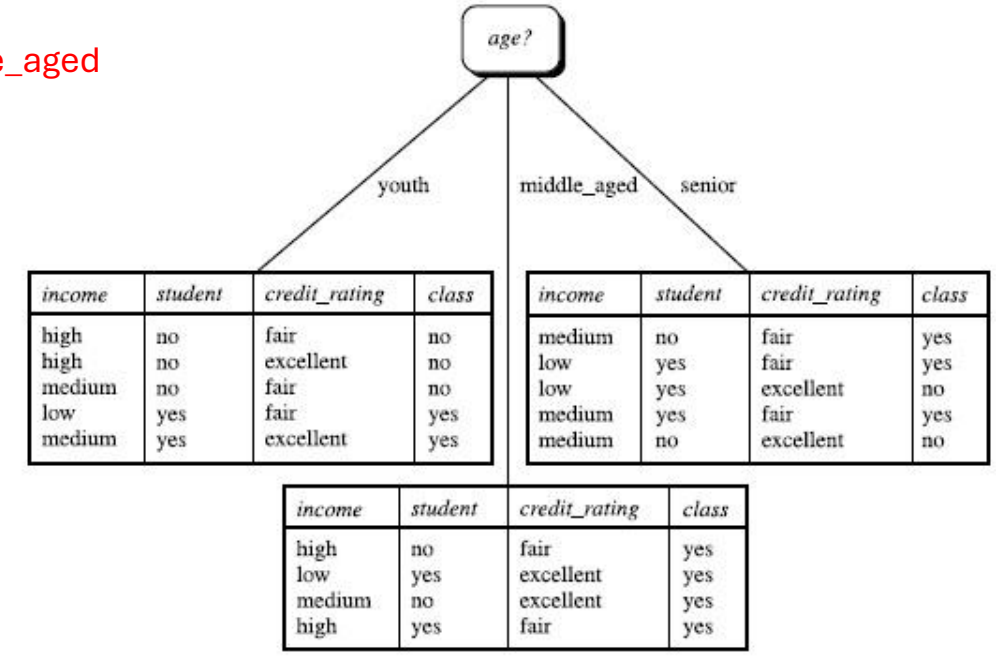
age: youth

age: middle_aged

age: senior

Information gain

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$



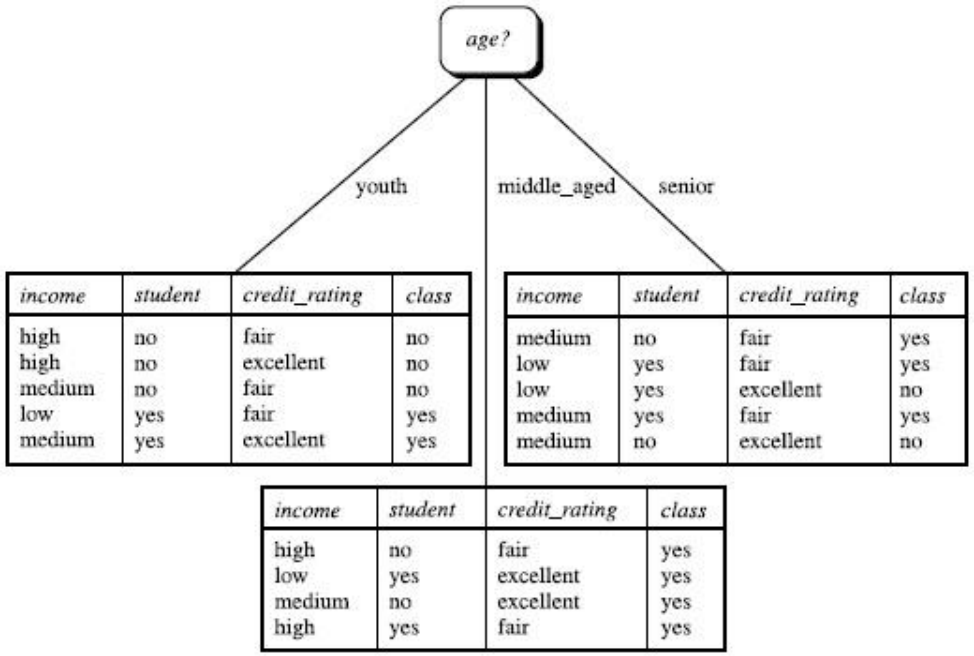
income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Information gain calculation

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

The gain for the other attributes are

Gain(income) = 0.029 bits

Gain(student) = 0.151 bits

Gain(credit_ rating) = 0.048 bits

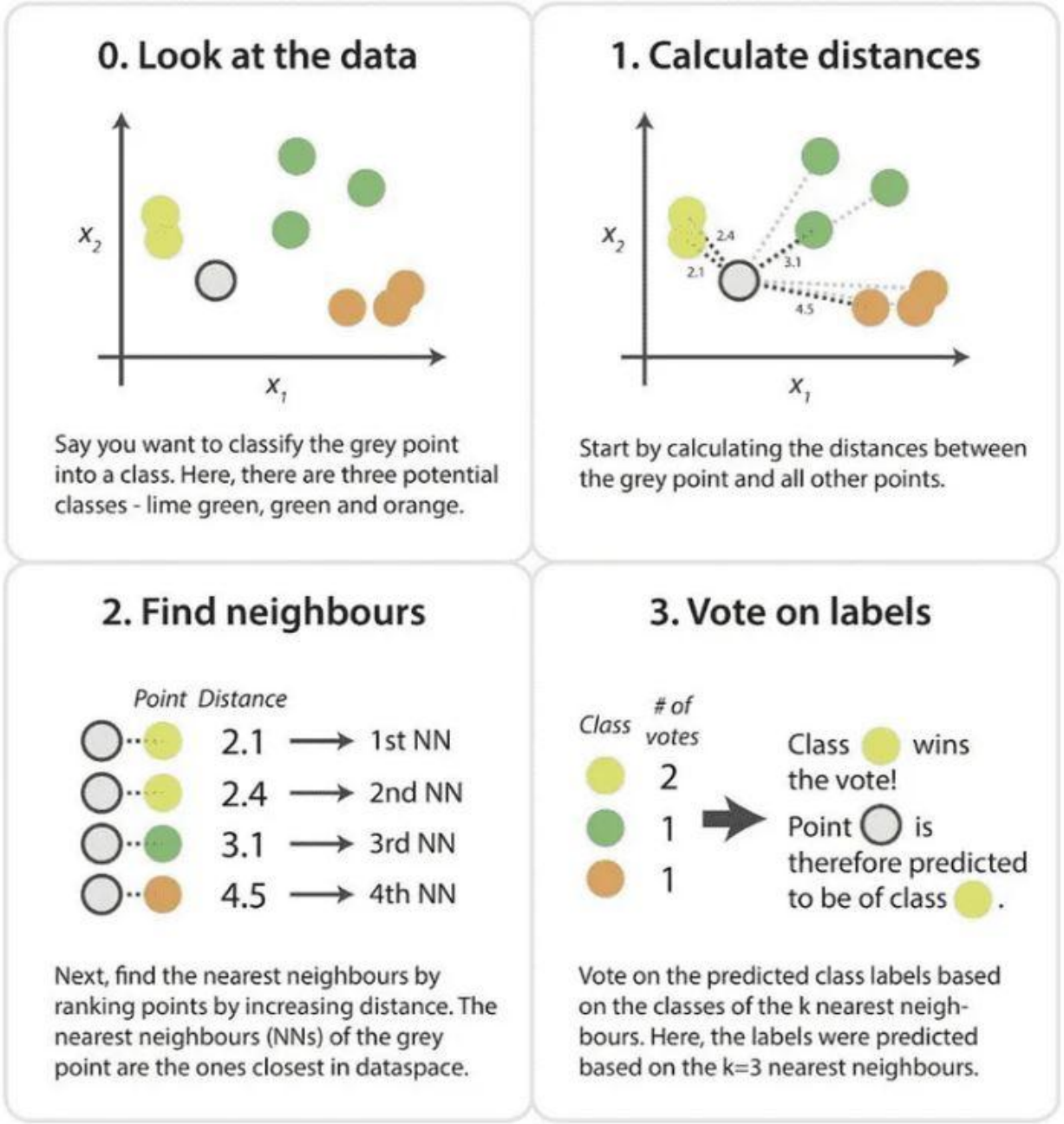
- Since, age has the highest gain value, it is selected to be placed at the root.
- The data is split as shown in the figure. The tuples for the partition *middle_aged* belong to the same category (class: yes), a pure partition. Hence a leaf node is created.
- For the other two impure partitions *(youth, senior)*, the same procedure is followed on the remaining features (income, student, credit_rating).

K Nearest Neighbor (KNN)

- The K Nearest Neighbor (KNN) is an algorithm that can be used for both classification and regression (mostly used for classification).
- It relies on the idea that similar data points tend to have similar labels or values.
- Training phase: stores the entire dataset (lazy learner)
- Testing phase: calculates distance between input data point to all the training examples using a distance metric (E.g. Euclidean dist., Manhattan dist., etc.)
- Parameter tuning: value of ‘k’ determined.
- Classify input value using the value of ‘k’.

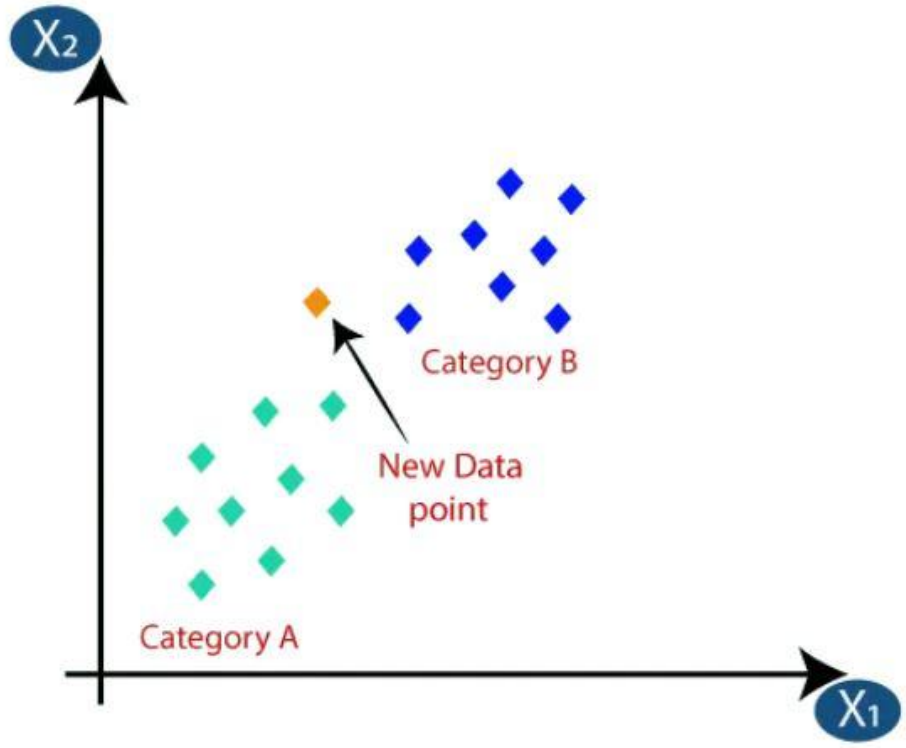
KNN

Step-1: Select the K number of the neighbors
Step-2: Calculate the Euclidean distance of K number of neighbors
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
Step-4: Among these k neighbors, count the number of the data points in each category.
Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.



KNN - Example

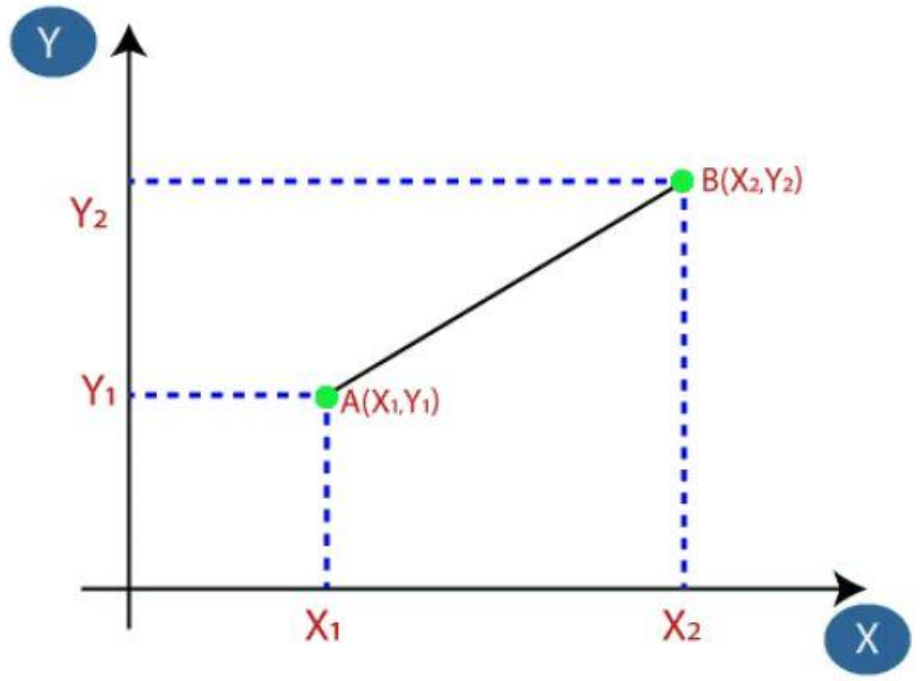
- There are two categories A and B, and we have a new data point to classify.
- Select the K number of the neighbors. Here the chose value of K is 5.



KNN - Example

- Calculate the Euclidean distance of K number of neighbors.

$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$



$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

KNN - Example

- Take the K nearest neighbors as per the calculated Euclidean distance.
  - *Based on the Euclidean distance*
  - *Nearest neighbors in category A = 3*
  - *Nearest neighbors in category B = 2*
- Assign the new data points to that category for which the number of the neighbor is maximum.
  - *The new data point is assigned to category A*

https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

Determining value of ‘K’

- The selection of the optimal ‘k’ value depends on the specific dataset and the characteristics of the problem, and it can significantly impact the performance of the algorithm.
- Considerations
  - Odd value of ‘k’ more suitable.
  - Smaller value of ‘k’ can be noisy, while larger can have bias and is more computationally expensive.
  - Cross validation
  - K = sqrt(N), where N = no. of samples in the dataset.

Example: Determine the ‘Default’ status for Andrew.

Customer	Age	Loan	Default
John	25	40000	N
Smith	35	60000	N
Alex	45	80000	N
Jade	20	20000	N
Kate	35	120000	N
Mark	52	18000	N
Anil	23	95000	Y
Pat	40	62000	Y
George	60	100000	Y
Jim	48	220000	Y
Jack	33	150000	Y
Andrew	48	142000	?

Training data

Customer	Age	Loan	Default	Euclidean distance
John	25	40000	N	1,02,000.00
Smith	35	60000	N	82,000.00
Alex	45	80000	N	62,000.00
Jade	20	20000	N	1,22,000.00
Kate	35	120000	N	22,000.00
Mark	52	18000	N	1,24,000.00
Anil	23	95000	Y	47,000.01
Pat	40	62000	Y	80,000.00
George	60	100000	Y	42,000.00
Jim	48	220000	Y	78,000.00
Jack	33	150000	Y	8,000.01
Andrew	**48**	**142000**	**?**	

Step 1: Calculate the distance to all data points

Example: Determine the ‘Default’ status for Andrew.

Customer	Age	Loan	Default	Euclidean distance	Minimum Euclidean Distance
John	25	40000	N	1,02,000.00	
Smith	35	60000	N	82,000.00	
Alex	**45**	**80000**	**N**	**62,000.00**	**5**
Jade	20	20000	N	1,22,000.00	
Kate	**35**	**120000**	**N**	**22,000.00**	**2**
Mark	52	18000	N	1,24,000.00	
Anil	**23**	**95000**	**Y**	**47,000.01**	**4**
Pat	40	62000	Y	80,000.00	
George	**60**	**100000**	**Y**	**42,000.00**	**3**
Jim	48	220000	Y	78,000.00	
Jack	**33**	**150000**	**Y**	**8,000.01**	**1**
Andrew	48	142000	?		

Step 2: Assume k = 5
Step 3: Find five minimum distance values.

Out of 5 chosen records there are three Ys and two Ns.
Andrew’s default status is taken as Y.