Department of AI&DS

MACHINE LEARNING
22AD2203R

Topic:

TREE MODELS

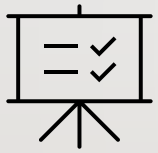
Session - 04

Dr. NARENDRA BABU TATINI
Associate Professor
Department of IoT

AIM OF THE SESSION

To build an accurate and efficient machine learning model that can handle both classification and regression tasks.

INSTRUCTIONAL OBJECTIVES



This session is designed to:

1. Understand the basic concepts and principles of decision tree learning.
2. Apply decision tree learning algorithms to solve classification and regression problems.

LEARNING OUTCOMES

At the end of this session, you should be able to:

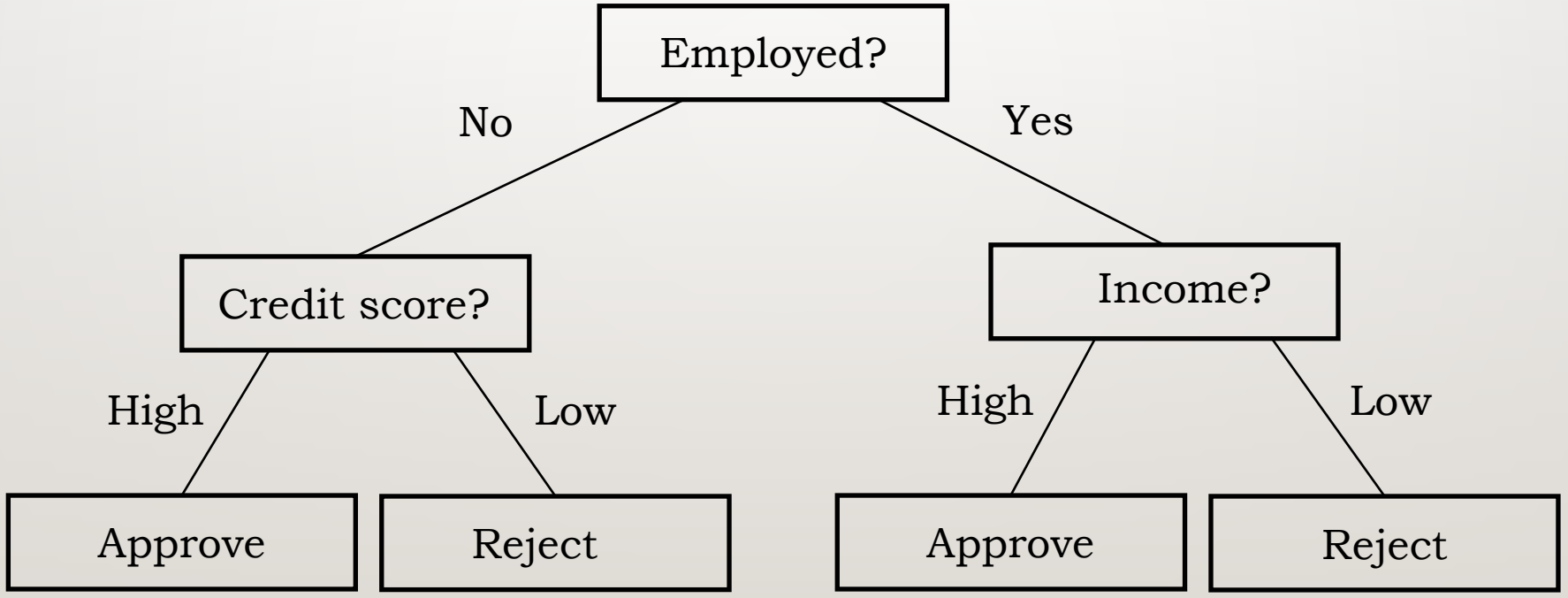
1. Describe the different algorithms used to construct decision tree models, and
2. Apply decision tree learning techniques to real-world datasets and solve classification or regression problems.

DECISION TREE

- Decision trees are a foundational concept in machine learning. They are predictive models that replicate the way humans make decisions. At their core, decision trees are hierarchical structures used for tasks like classification and regression.
- The structure of a decision tree is akin to a flowchart, with each node representing a decision point, each branch signifying a possible decision or criteria, and each leaf node offering a final prediction or classification.

DECISION TREE REPRESENTATION

A decision tree for whether to approve a loan.



DECISION TREE

- Decision trees work by asking a series of questions based on input features, guiding the process to a final decision or prediction. This sequential, tree-like decision-making process is what makes them distinct in the machine learning landscape.
- Decision trees are a key component of interpretability in machine learning. They provide transparency into the decision-making process, making them valuable for understanding why a model makes a particular prediction.

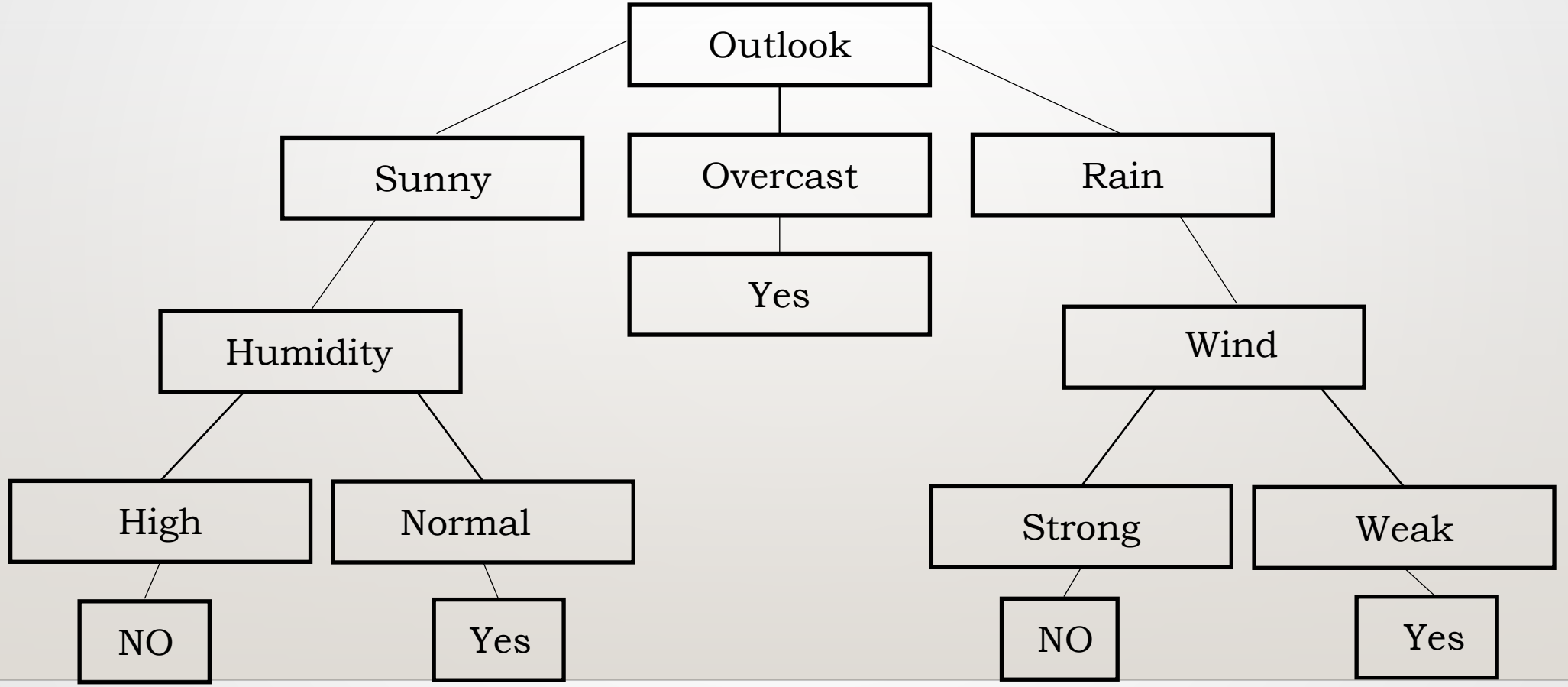
DECISION TREE FOR PLAY TENNIS

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

DECISION TREE FOR PLAY TENNIS

- Attributes and their values:
  - Outlook: Sunny, Overcast, Rain
  - Temperature: Hot, Mild, Cool
  - Humidity: High, Normal
  - Wind: Strong, Weak
- Target:
  - Play Tennis: Yes, No

DECISION TREE FOR PLAY TENNIS



ID3 ALGORITHM

- ID3 is a basic decision tree learning algorithm.
- ID3 stands for Iterative Dichotomiser 3.
- The ID3 algorithm was invented by Ross Quinlan in 1975.
- Used to generate a decision tree from a given data set by employing a top-down, greedy search, to test each attribute at every node of the tree.
- The resulting tree is used to classify future samples.
- It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain (G) or minimum Entropy (S).

ID3 ALGORITHM

- Create a Root node for the tree
- If all Examples are positive, Return the single-node tree Root, with label = +
- If all Examples are negative, Return the single-node tree Root, with label = -
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Examples.

ID3 ALGORITHM

- Otherwise Begin
  - A ← the attribute from Attributes that best* classifies Examples
  - The decision attribute for Root ← A
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test A = v_i
    - Let *Examples* v_i, be the subset of Examples that have value v_i for *A*
    - If *Examples* v_i , is empty
      - Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples
      - Else below this new branch add the subtree
        ID3(*Examples* v_i, Targe_tattribute, Attributes – {A}))
- End
- Return Root

WHICH ATTRIBUTE IS THE BEST CLASSIFIER?

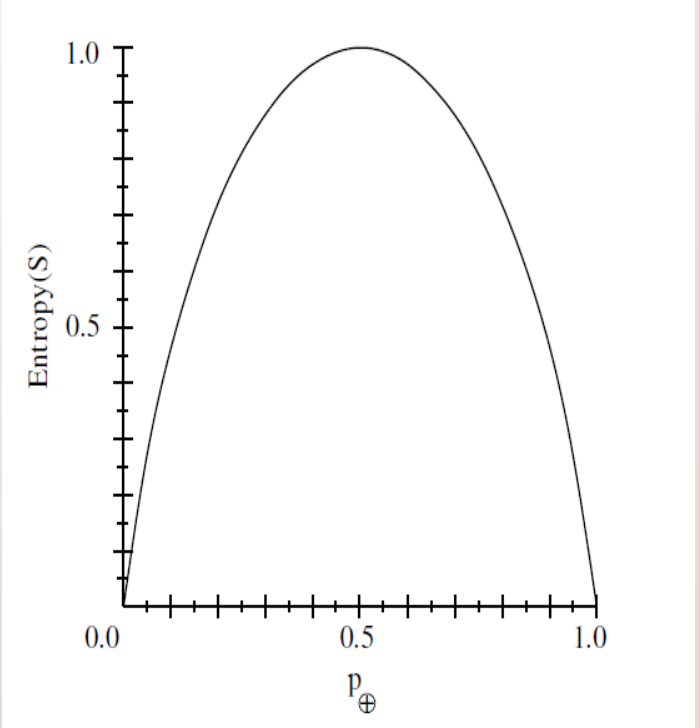
- The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree.
- A statistical property called ***information gain*** that measures how well a given attribute separates the training examples according to their target classification.
- ID3 uses this *information gain* measure to select among the candidate attributes at each step while growing the tree.
- In order to define information gain precisely, we use a measure commonly used in information theory, called ***entropy.***

ENTROPY

- Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is

$$Entropy\,(S) = \; p_+\,(-\log_2 p_+) + p_-\,(-\log_2 p_-)$$

$$Entropy\,(S) = \; -p_+ \log_2 p_+ - \; p_- \log_2 p_-$$



Where,

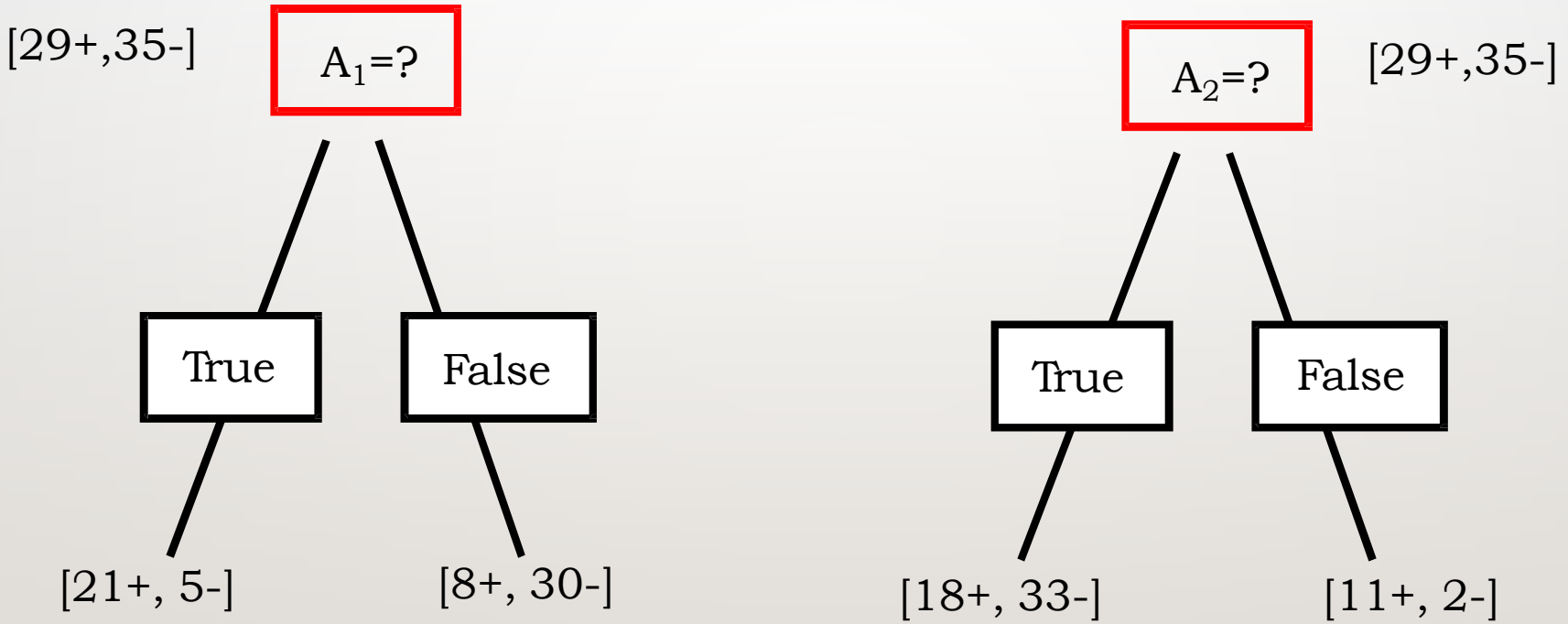
- p_+ is the proportion of positive examples in S.
- p_- is the proportion of negative examples in S.

INFORMATION GAIN

- ***Information gain,*** is the expected reduction in entropy caused by partitioning the examples according to this attribute.
- The information gain, Gain(S, A) of an attribute A, relative to a collection of examples S, is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

WHICH ATTRIBUTE IS THE BEST CLASSIFIER?



WHICH ATTRIBUTE IS THE BEST CLASSIFIER?

- $Entropy\ ([29+, 35-]) = -\frac{29}{64}\log_2\frac{29}{64} - \frac{35}{64}\log_2\frac{35}{64} = 0.99$

- $Entropy\ ([21+, 5-]) = -\frac{21}{26}\log_2\frac{21}{26} - \frac{5}{26}\log_2\frac{5}{26} = 0.71$
- $Entropy\ ([8+, 30-]) = -\frac{8}{38}\log_2\frac{8}{38} - \frac{30}{38}\log_2\frac{30}{38} = 0.74$
- $Gain\ (S, A_1) = Entropy\ ([29+, 35-]) - \frac{26}{64}\ Entropy\ ([21+, 5-]) - \frac{38}{64} Entropy\ ([8+, 30-]) = 0.27$

- $Entropy\ ([18+, 33-]) = -\frac{18}{51}\log_2\frac{18}{51} - \frac{33}{51}\log_2\frac{33}{51} = 0.94$
- $Entropy\ ([11+, 2-]) = -\frac{11}{13}\log_2\frac{11}{13} - \frac{2}{13}\log_2\frac{2}{13} = 0.62$
- $Gain\ (S, A_2) = Entropy\ ([29+, 35-]) - \frac{51}{64}\ Entropy\ ([18+, 33-]) - \frac{13}{64} Entropy\ ([11+, 2-]) = 0.12$

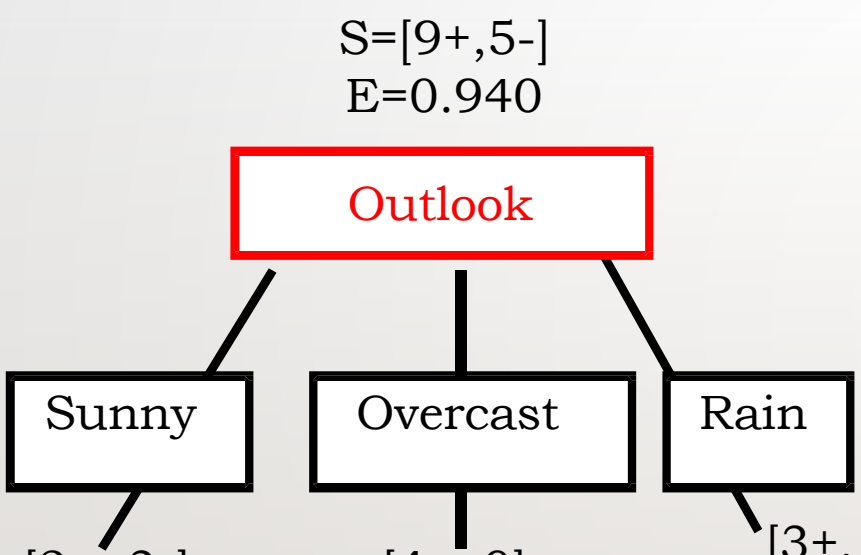
- **A_1 provides greater information gain than A_2, So A_1 is a better classifier than A_2.**

AN ILLUSTRATIVE EXAMPLE FOR PLAY TENNIS

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

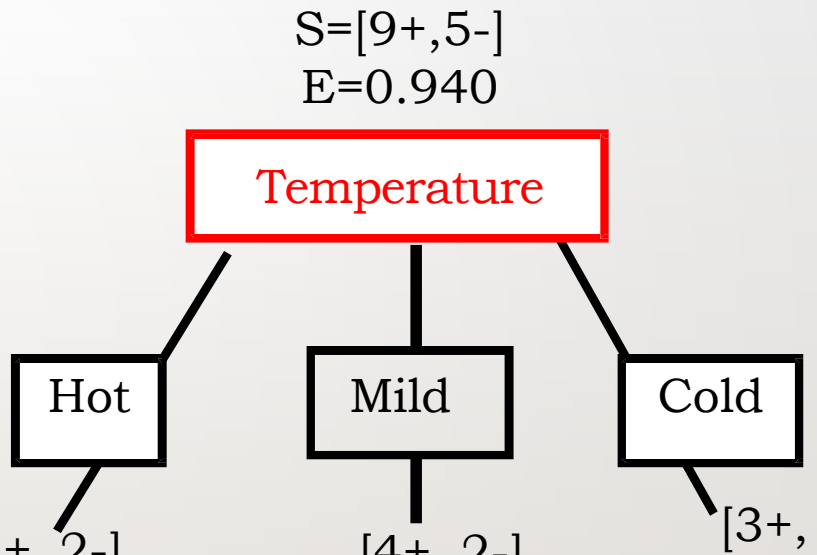
SELECTING NEXT ATTRIBUTE

- Entropy([9+,5-] = – (9/14) $\log_2$(9/14) – (5/14) $\log_2$(5/14) = 0.940



Gain(S, Outlook) = 0.940-(5/14)*0.971 -(4/14)*0.0 -(5/14)*0.0971

= **0.247**

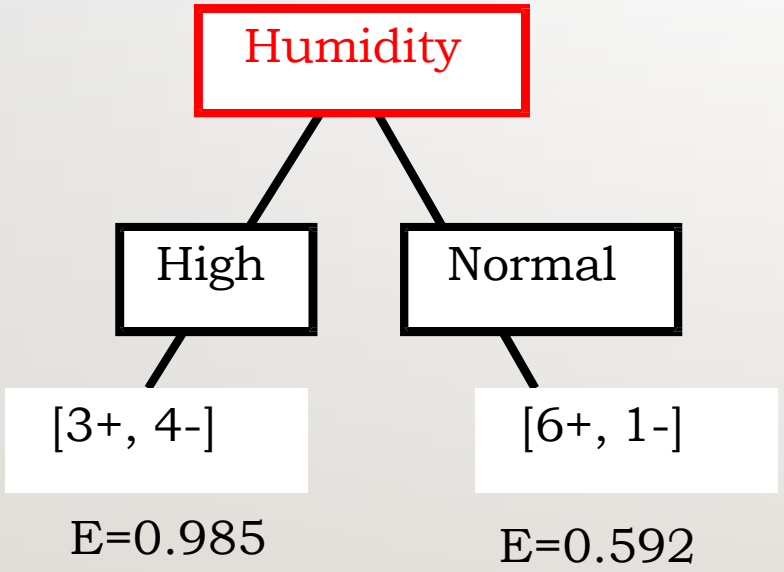


Gain(S, Temp) = 0.940-(4/14)*1.0 - (6/14)*0.911 - (4/14)*0.811

= **0.029**

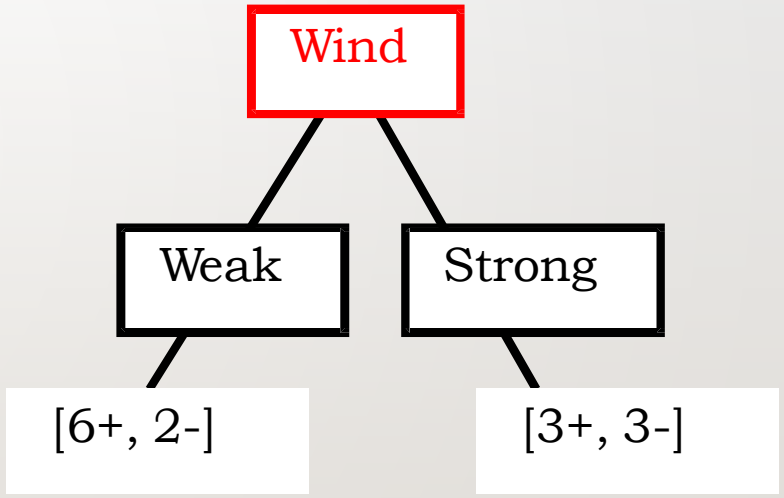
SELECTING NEXT ATTRIBUTE

S=[9+,5-]
E=0.940



Gain(S, Humidity) = 0.940-(7/14)*0.985-(7/14)*0.592
= **0.151**

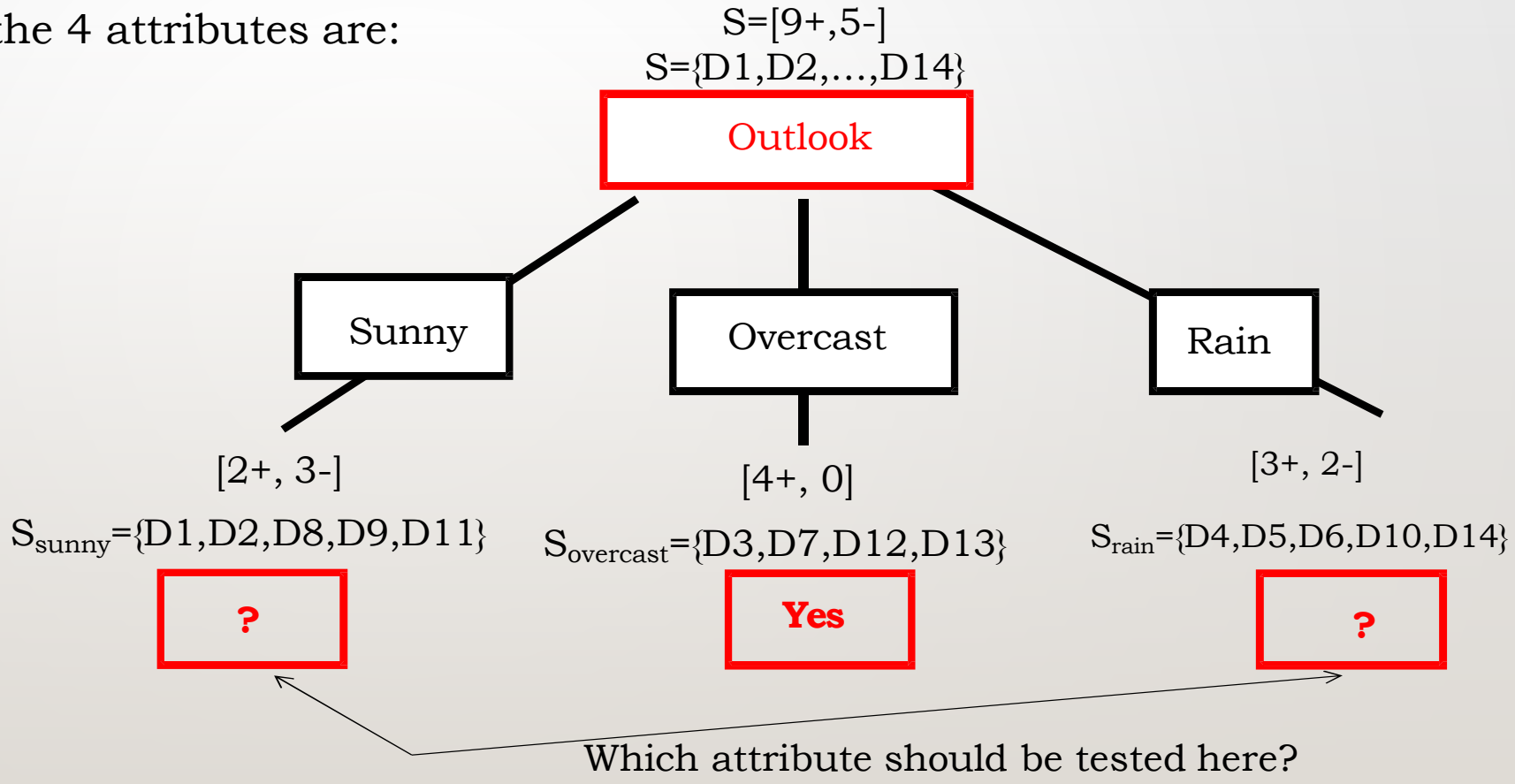
S=[9+,5-]
E=0.940



Gain(S, Wind) = 0.940-(8/14)*0.811-(6/14)*1.0
= **0.048**

BEST ATTRIBUTE-OUTLOOK

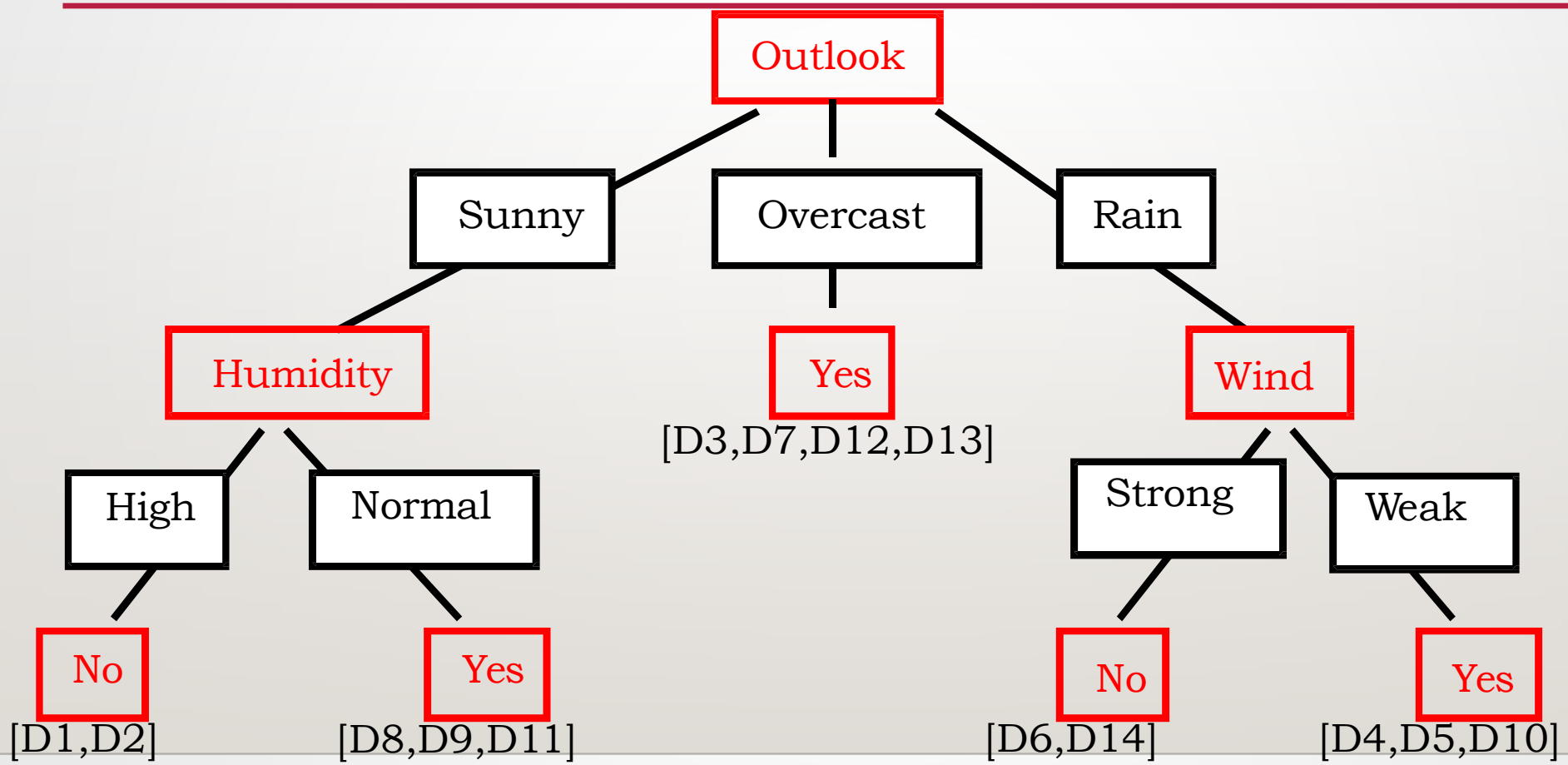
- The information gain values for the 4 attributes are:
  - Gain(S, Outlook) = **0.247**
  - Gain(S, Temp) = 0.029
  - Gain(S, Humidity) = 0.151
  - Gain(S, Wind) = 0.048



ID3-SUNNY

- Gain(S_{sunny} , Humidity) = 0.970-(3/5)0.0 – 2/5(0.0) = **0.970**
- Gain(S_{sunny} , Temp.) = 0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570
- Gain(S_{sunny} , Wind) = 0.970= -(2/5)1.0 – 3/5(0.918) = 0.019

ID3-RESULT


Outlook
Sunny
Overcast
Rain
Humidity
Yes
[D3,D7,D12,D13]
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes
[D1,D2]
[D8,D9,D11]
[D6,D14]
[D4,D5,D10]

CLASSIFICATION AND REGRESSION TREES (CART)

- Another well-known tree-based algorithm, CART, indicates that it can be used for both classification and regression.
- Classification is not wildly different in CART, although it is usually constrained to construct binary trees.

  For Example:

  A question that has three answers (say the question about when your nearest assignment deadline is, which is either 'urgent', 'near', or 'none')

  It can be split into two questions: first, '**is the deadline urgent?**', and then if the answer to that is 'no', second '**is the deadline near?**'

- The only real difference with classification in CART is that a different information measure is commonly used.

GINI IMPURITY

- The entropy that was used in ID3 as the information measure is not the only way to pick features. Another is **Gini impurity**.
- The 'impurity' in the name suggests that the aim of the decision tree is to have each leaf node represent a set of data-points that are in the same class. so that there are no mismatches. This is known as purity.
- If a leaf is pure then all of the training data within it have just one class.

  In which case, if we count the number of data points at the node, that belong to a class i (call it N(i)), then it should be 0 for all except one value of i.

GINI IMPURITY

- So suppose that you want to decide on which feature to choose for a split. The algorithm loops over the different features and checks how many points belong to each class.
- If the node is pure, then N(i) = 0 for all values of i except one particular one.
- So for any particular feature k you can compute:

$$G_k = \sum_{i=1}^{c} \sum_{j \neq i} N(i)N(j).$$

where c is the number of classes. In fact, you can reduce the algorithmic effort required by noticing that $\sum_i N(i) = 1$ (since there has to be some output class) and so $\sum_{j \neq i} N(j) = 1 - N(i)$. Then Equation is equivalent to:

$$G_k = 1 - \sum_{i=1}^{c} N(i)^2.$$

EXAMPLE

- An attribute with the smallest Gini Impurity is selected for splitting the node.
- If a data set D is split on an attribute A into two subsets D_1 and D_2 with sizes n_1 and n_2, respectively, the Gini Impurity can be defined as:

$$Gini_A(D) = \frac{n_1}{n} Gini(D_1) + \frac{n_2}{n} Gini(D_2)$$

- When training a decision tree, the attribute that provides the smallest $Gini_A(D)$ is chosen to split the node.
- In order to obtain information gain for an attribute, the weighted impurities of the branches is subtracted from the original impurity. The best split can also be chosen by maximizing the **Gini gain**. Gini gain is calculated as follows:

$$\triangle Gini(A) = Gini(D) - Gini_A(D)$$

	Count		Probability		Gini Impurity
	n_1	n_2	p_1	p_2	$1 - p_1^2 - p_2^2$
Node A	0	10	0	1	$1 - 0^2 - 1^2 = 0$
Node B	3	7	0.3	0.7	$1 - 0.3^2 - 0.7^2 = 0.42$
Node C	5	5	0.5	0.5	$1 - 0.5^2 - 0.5^2 = 0.5$

REGRESSION

- Regression analysis is a statistical method used to model the relationship between a dependent (target) variable and one or more independent variables.
- Provides a clear understanding of the factors that influence the target variable. By identifying these relationships, it enables the development of accurate machine learning models.
- Regression models are employed when the goal is to predict continuous data. They excel at estimating values within a range, making them valuable in various domains.

Example:

- ***Predicting Rainfall:*** Rainfall can depend on various factors, including humidity, temperature, wind direction, and wind speed. A regression model can help forecast rainfall amounts based on these variables.
- ***Predicting House Prices:*** The price of a house in a certain area may be influenced by a multitude of factors, such as size, locality, distance to schools, hospitals, metro stations, marketplaces, pollution levels, and the presence of parks. A regression model can take these variables into account to estimate house prices accurately.

REGRESSION

- *Dependent Variable:* The dependent variable, also known as the target variable, is the variable we aim to predict or understand. It represents the outcome of interest in our analysis.
- *Independent Variables:* Independent variables, often referred to as predictors or features, are the variables used to predict the values of the dependent variable. They serve as the input values in a predictive model.
- Independent variables can be thought of as the inputs, while the dependent variable is the output of these inputs. The relationships between the independent and dependent variables are the core of regression analysis.

Example:

- ***Dependent Variable:*** In the example of predicting house prices, the dependent variable is the "Price of the house." This is the value we want to estimate or predict.
- ***Independent Variables:*** The independent variables in this case are "size," "Locality," "distance from school, hospital, metro, and market," "pollution level," and "number of parks." These are the factors that influence or help predict the house price.

TYPES OF REGRESSION

Linear Regression

- It is the linear relationship between the dependent and independent variables.

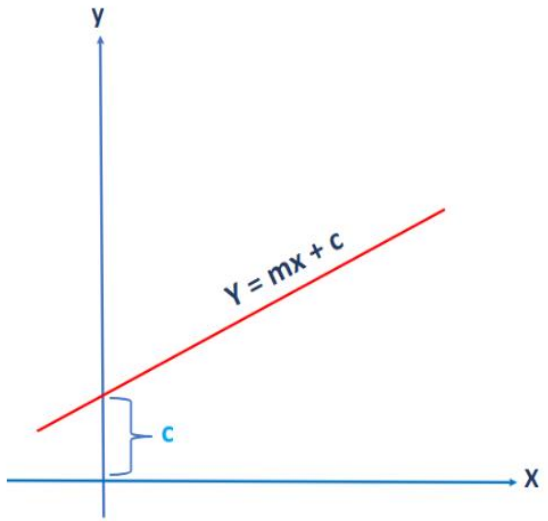
Example: In house price prediction, as the size of the house increases, the price of houses also increases. Here, the size of the house is an independent variable and the price of a house is a dependent variable.

- In a linear regression model, we try to find the best fit line (i.e. the best value of m and c) to reduce the error.
- **Note:**Error is the difference between the actual value and the predictive value.

Equation of Line:

$y = mx + c$

where,
m = slope of line
c = intercept
y = dependent variable
x = independent variable



Linear Regression Equation:

- **Single independent Variable**
  $y = a_0 + a_1 x$
- **Multiple independent variable**
  $y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \ldots + a_n x_n$

TYPES OF REGRESSION

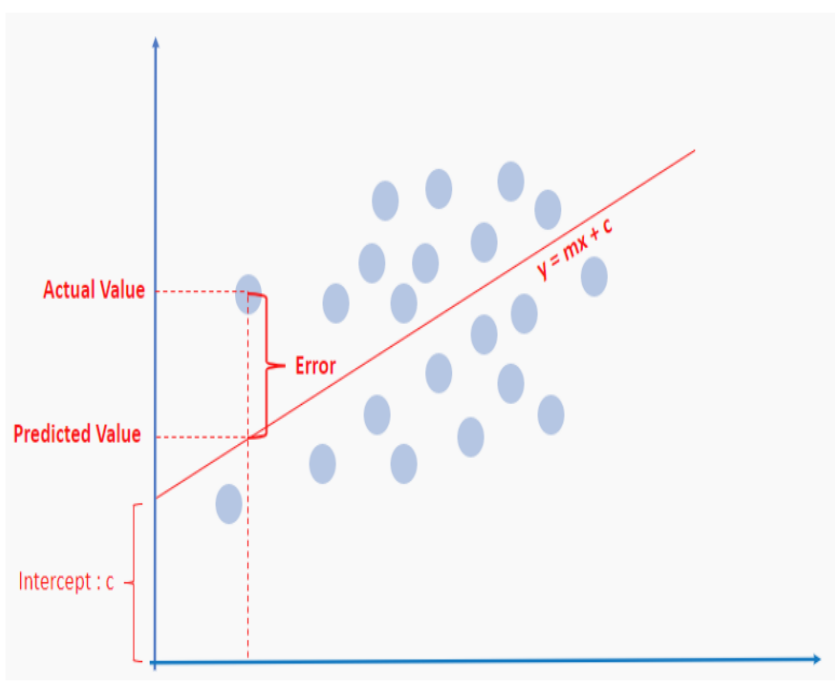
Mean square Error (MSE)

- In order to find the best fit line, we will use mean square error.
- Mean Square Error (MSE) is a critical metric used to measure the accuracy and quality of a statistical model.
- It is calculated as the average of the squared differences between the actual (observed) values and the predicted values. The formula for MSE is:

$$MSE = \frac{1}{n} * \sum (y - \hat{y})^2$$

Where:

- n is the number of data points.
- y represents the actual (observed) values.
- $\hat{y}$ represents the predicted values.



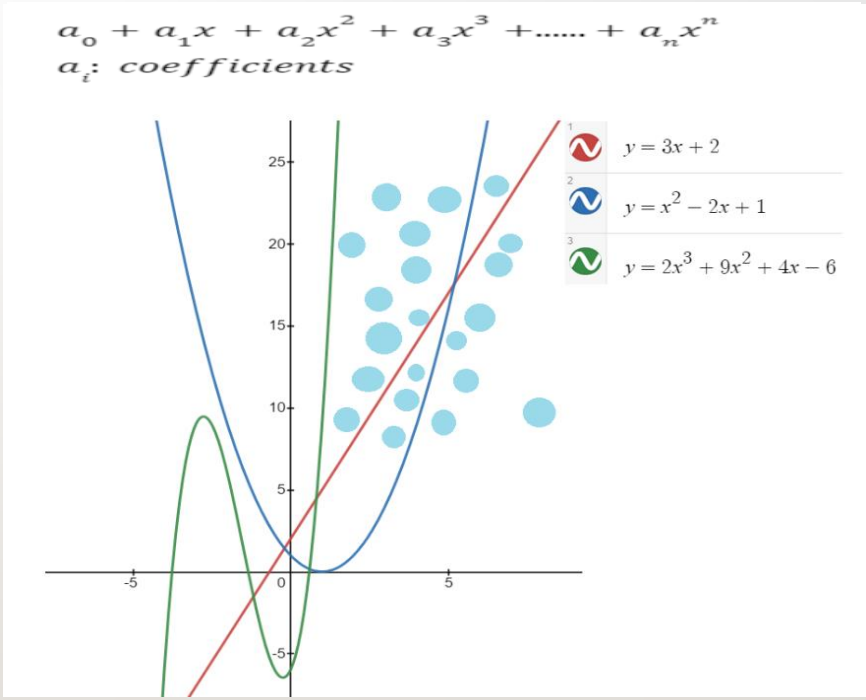
TYPES OF REGRESSION

- *Interpretation:* A lower MSE value indicates a better model fit. It signifies that the model's predictions are closer to the actual values.
- *Note:* MSE is widely used in model evaluation and selection, making it a valuable tool for assessing the performance of predictive models.

TYPES OF REGRESSION

Polynomial Regression

- It is a regression technique for modeling non-linear data by fitting polynomial functions to the relationship between independent and dependent variables.
- A polynomial is a mathematical expression consisting of variable powers multiplied by coefficients.
- Useful for capturing complex non-linear relationships in data, especially when points follow a curve.
- Allows the use of polynomial functions of various orders (e.g., quadratic, cubic) to adapt to the data's curvature.



TYPES OF REGRESSION

Logistic regression

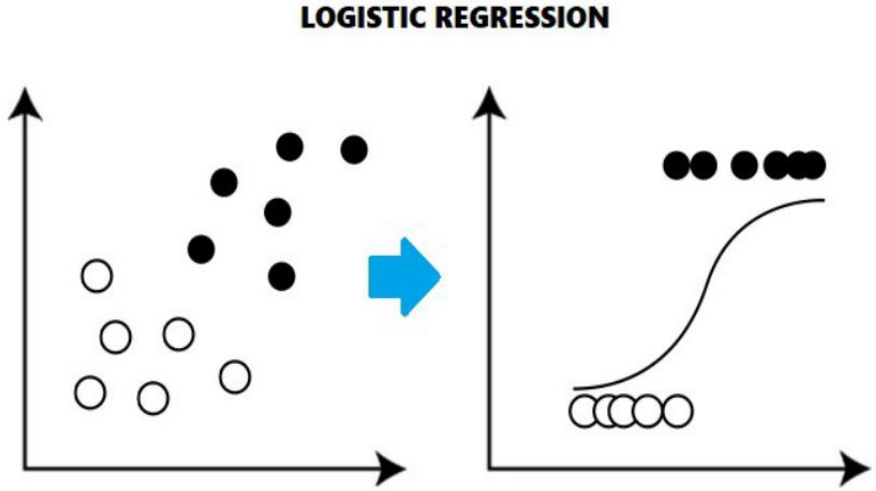
- Logistic regression is employed when computing the probability of mutually exclusive events, such as True/False, Pass/Fail, 0/1, or Yes/No.
- It is the choice when the dependent variable is discrete, taking only one of two values.
- It relies on the sigmoid function to model the relationship between independent variables and the probability of a specific outcome.

$$sig(x) = \frac{1}{1+e^{-x}}$$

$sig(x)$: *sigmoid function (value is between 0 and 1)*

x: *value of input*

$e = 2.718$



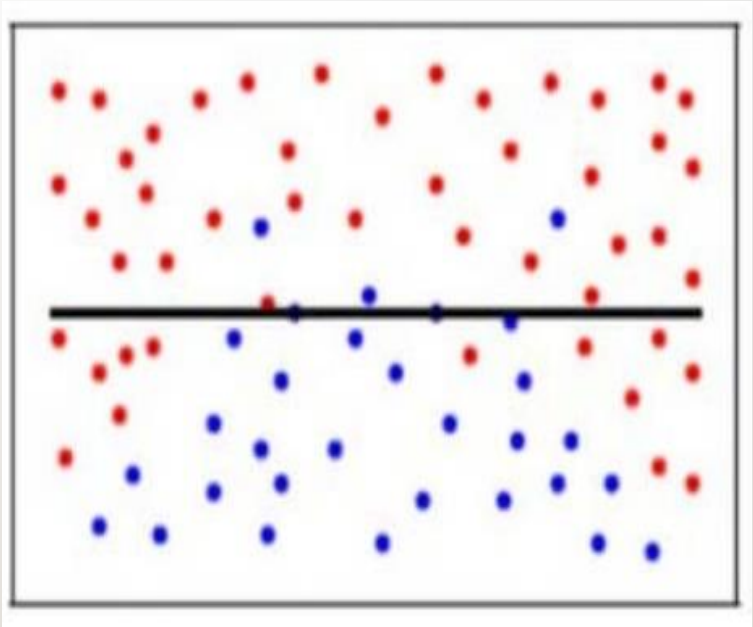
BIAS

- Bias is the disparity between the Predicted Value and the Expected Value in machine learning.
- Machine learning models make certain assumptions when they are trained on data. However, these assumptions may not always hold true when applied to testing or validation data.

Example: Suppose a model, using a large number of nearest neighbors, simplifies the prediction process by considering only a subset of parameters. For instance, it may assume that Glucose levels and Blood Pressure solely determine whether a patient has diabetes. This simplification makes strong assumptions about other parameters not influencing the outcome. This oversimplified model can be likened to underfitting, where the model predicts a simple relationship while the data suggests a more complex one.

BIAS

- The relationship between input variables (X) and the target variable (Y) is represented as Y = f(X) + e.
- where 'e' signifies the error that follows a normal distribution.
- The goal of the model f'(x) is to predict values as close to f(x) as possible. The bias of the model is mathematically expressed as: $Bias[f'(X)] = E[f'(X) - f(X)]$
- *Underfitting:* When the model's high bias leads to oversimplified generalizations and fails to capture variations effectively, it results in underfitting.



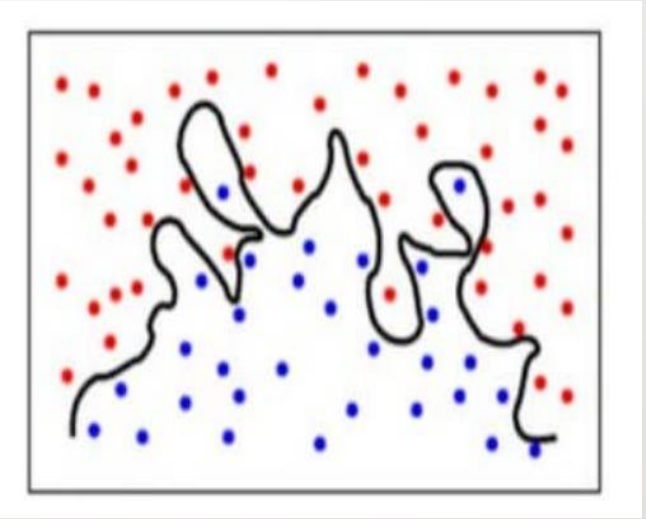
VARIANCE

- Variance in machine learning refers to the model's sensitivity to fluctuations or noise in the data.
- *High Variance:* A high-variance model takes into account not only the underlying patterns but also the noise in the training data. This means it learns too much from the training data, leading to issues when making predictions on new (testing) data.
- *Mathematical Representation:* The variance error in the model can be mathematically expressed as:

Variancef(x)) = E[X^2] - E[X]^2

VARIANCE

- *Overfitting:* When the model overlearns from the training data and captures even the noise, it is said to have high variance, resulting in overfitting.



Example:

- Consider a classification model that predicts whether a patient has diabetes. If the model, due to high variance, makes complex and overfitted predictions like "if the number of pregnancies is more than 3, glucose level is more than 78, diastolic blood pressure is less than 98, skin thickness is less than 23 mm, and so on for every feature, then the patient has diabetes," it is prone to making inaccurate predictions when exposed to new data. This can be costly and unreliable.

BIAS-VARIANCE TRADEOFF IN MACHINE LEARNING

- The bias-variance tradeoff is a fundamental concept in machine learning, striking a balance between two critical sources of error: bias and variance.
- Bias represents errors due to overly simplistic model assumptions. High bias results in underfitting, causing poor performance on training and unseen data.
- Variance reflects the model's sensitivity to fluctuations in the training data. High variance leads to overfitting, where the model captures noise and performs poorly on new data.
- The goal is to find the right level of complexity in a model to minimize both bias and variance. Achieving this balance is essential for good generalization to new data.

Self-Assessment Questions

1. Which of the following is a crucial step in designing a machine learning system?

(a) Selecting a programing language
(b) Choosing a machine learning algorithm randomly
(c) Skipping the evaluation phase
(d) Collecting and preparing a high-quality dataset

2. What is the purpose of feature engineering in machine learning system design?

(a) To automate the model training process
(b) To select the best machine learning algorithm
(c) To extract and transform raw data into informative features
(d) To validate the performance of the machine learning model

Self-Assessment Questions

3. What does a high variance in a machine learning model indicate?

A) The model makes overly simplistic assumptions.
B) The model overlearns from the training data.
C) The model underfits the data.
D) The model balances bias and variance effectively.

4. Which of the following strategies is NOT typically used to manage high bias in a machine learning model?

A) Regularization techniques.
B) Feature selection.
C) Increasing model complexity.
D) Ensemble methods.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Text Books:

1. Mitchell, Tom. Machine Learning. New York, NY: McGraw-Hill, 1997. ISBN: 9780070428072.
2. MacKay, David. Information Theory, Inference, and Learning Algorithms. Cambridge, UK: Cambridge University Press, 2003. ISBN: 9780521642989.

Reference Books:

1. EthemAlpaydin "Introduction to Machine Learning ", The MIT Press (2010).
2. Stephen Marsland, "Machine Learning an Algorithmic Perspective" CRC Press, (2009).

Sites and Web links:

1. Data Science and Machine Learning: https://www.edx.org/course/data-science-machinelearning.
2. Machine Learning: https://www.ocw.mit.edu/courses/6-867-machine-learning-fall-2006/.

THANK YOU

Team – MACHINE LEARNING