EXPERT SYSTEM :

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The system helps in decision making for complex problems using **both facts and heuristics like a human expert**. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain.

- Use Case:
  - Expert systems are used in various fields such as medicine, finance, engineering, and customer service.
  - Examples include medical diagnosis, financial advising, fault diagnosis in machinery, and troubleshooting technical problems.
- Human Expert VS Expert System:

Aspect	**Human Expert**	**Expert System**
Learning Curve	Years of education and experience	Requires programming and knowledge input
Availability	Limited by time and resources	Available 24/7, consistent performance
Bias	May be influenced by personal bias	Impartial, based solely on programmed rules
Scalability	Limited by human capacity	Scalable, can handle large volumes of data
Speed	Limited by human processing speed	Can process information rapidly

Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

An expert system mainly consists of three components:

- **User Interface:**
  - Facilitates interaction between user and expert system.
  - Takes user queries in readable format and passes to inference engine.
  - Displays output to user.

- Helps non-expert users communicate with system.

- **Inference Engine:**

  - Main processing unit.

  - Applies inference rules to knowledge base to derive conclusions.

  - Extracts knowledge from knowledge base.

  - Types: Deterministic (based on facts and rules) and Probabilistic (contains uncertainty).

  - Modes: Forward Chaining (from known facts and rules) and Backward Chaining (starts from goal and works backward).

- **Knowledge Base**:

  - Stores knowledge from domain experts.

  - Essential storage for expert system.

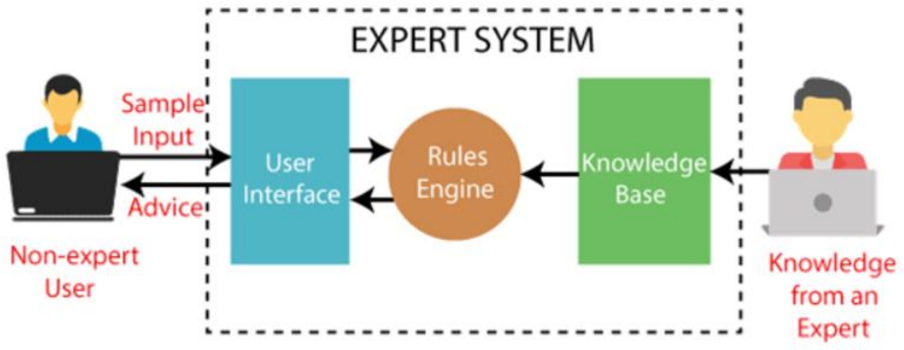
  - Similar to a database.

  - Contains information and rules of domain.

  - Collections of objects and their attributes (e.g., Lion: mammal, not domestic).

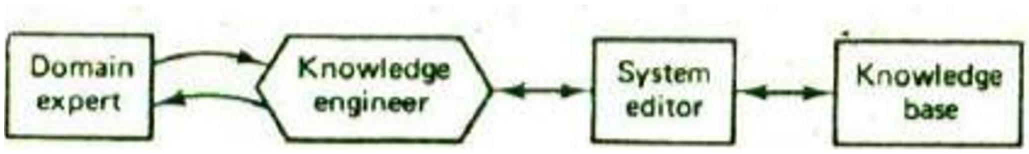
  - Components of Knowledge Base:

    a. Factual Knowledge: Accepted facts by knowledge engineers.
    b. Heuristic Knowledge: Based on practice, guesswork, evaluation, and experience.
    c. Knowledge Representation: Formalizing knowledge using If-else rules.
    d. Knowledge Acquisition: Extracting, organizing, and storing domain knowledge in the knowledge base.



- **Knowledge Acquisition:**

  - Knowledge acquisition is the process of capturing and transferring expertise from human experts to the ES via rules, objects, and frame based ontologies.
  - It involves identifying, extracting, and structuring knowledge relevant to the problem domain.



- **Methods:**
  - **Manual Acquisition:** Direct input from human experts through interviews, questionnaires, or documentation review.
  - **Automated Acquisition:** Utilizes tools such as data mining, natural language processing, or machine learning to extract knowledge from existing sources or data repositories.
  - **Interactive Acquisition:** Collaboration between human experts and the ES, where experts interact with the system to refine and enhance its knowledge base.

- **ES Development Life Cycle:**

  - Identification:
  1. Identify the problem area where an expert system can be beneficial.
  2. Assess the feasibility and potential impact of implementing an expert system.
  3. Define the goals and objectives of the expert system project.

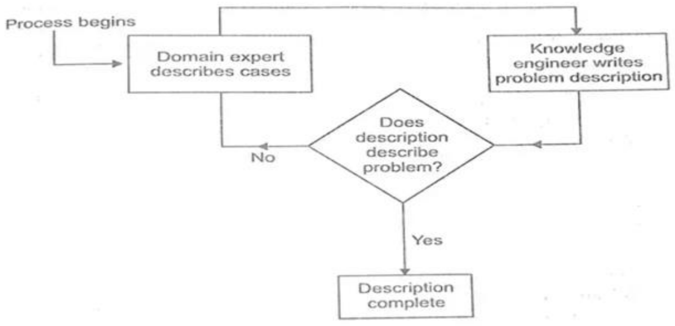

Process begins
Domain expert describes cases
Knowledge engineer writes problem description
Does description describe problem?
No
Yes
Description complete

Fig: Iterative Process of identifying the problem in Expert System

  - Conceptualization:
  1. Develop a conceptual framework for the expert system.
  2. Determine the scope and boundaries of the system.
  3. Define the knowledge sources and expertise required.

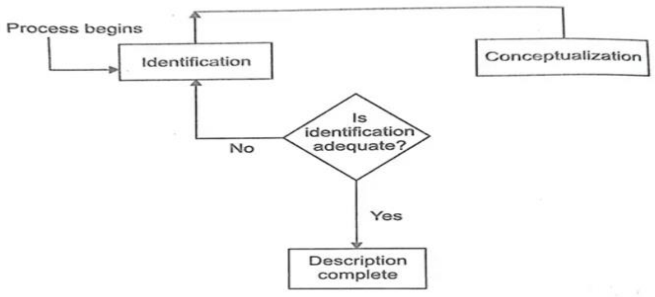


Fig: Conceptual Phase of Expert System Development Life Cycle

  - Formalization:
  1. Formalize the knowledge representation scheme.
  2. Capture and organize domain-specific knowledge into a knowledge base.
  3. Define the inference mechanisms and reasoning strategies.

  - Implementation:
  1. Develop the software architecture and user interface.

2. Implement the knowledge base, inference engine, and working memory.
3. Integrate the various components into a functional expert system.

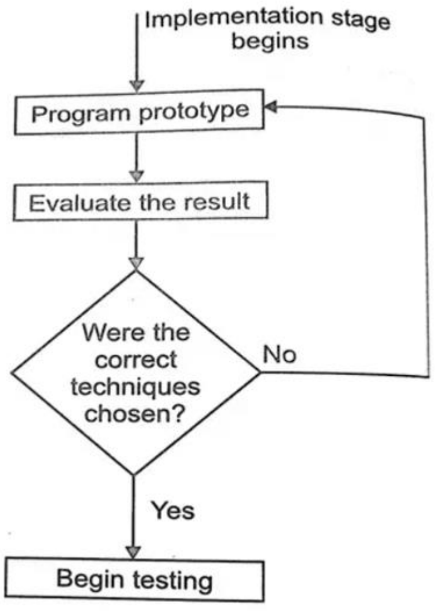
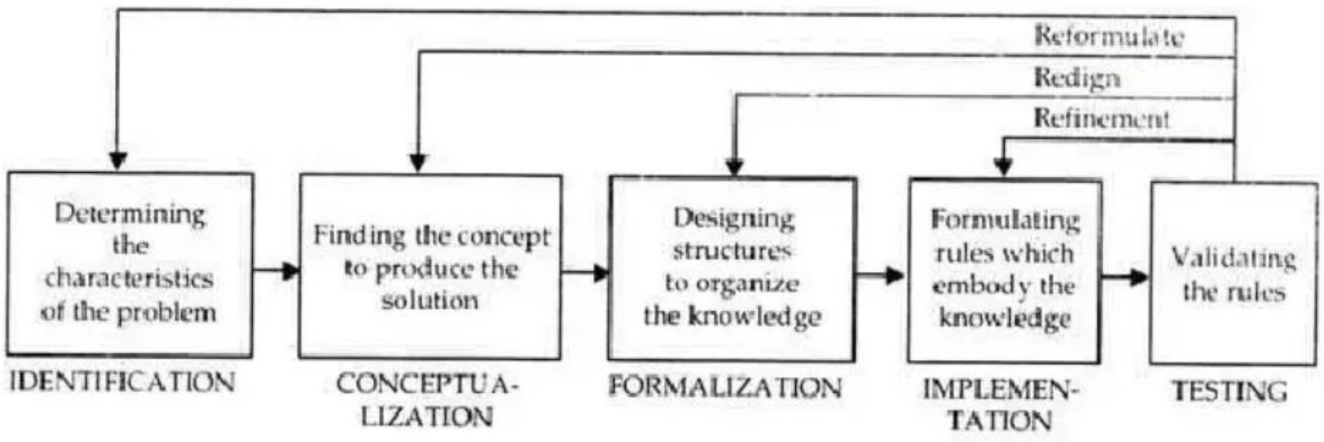

Implementation stage begins
Program prototype
Evaluate the result
Were the correct techniques chosen?
No
Yes
Begin testing

Fig: Implementation Phase of Expert System Development Life Cycle

- Testing:

1. Validate the expert system against predefined criteria and requirements.
2. Conduct thorough testing to identify and resolve any issues or errors.
3. Evaluate the performance and effectiveness of the expert system through simulation and real-world testing.


Reformulate
Redign
Refinement
Determining the characteristics of the problem
IDENTIFICATION
Finding the concept to produce the solution
CONCEPTUA-LIZATION
Designing structures to organize the knowledge
FORMALIZATION
Formulating rules which embody the knowledge
IMPLEMEN-TATION
Validating the rules
TESTING

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Why ES?

- No memory limitations
- High efficiency
- Expertise in domain
- Not affected by emotions
- High security
- Considers all facts
- Regular updates improve performance

Some popular examples of the Expert System:

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.
- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.

- **SHINE:** This system is designed for analyzing and monitoring the real-time or non-real-time systems which are designed by NASA.
- **MUDMAN:** MUDMAN is an expert system that helps the on-site engineer do the job more consistently designed by N L Baroid Company.
- **PROSPECTOR:** PROSPECTOR helps the United States in geological survey and exploration of geological minerals. Designed by the SRI's Artificial Intelligence Center.
- **XCON**: XCON is a rule-based expert system that is designed to automatically select the computer system components based on the customer's requirements.
- **CaDet**: CaDet is an Expert System that is designed to help in identifying cancer at early stages.

DENDRAL:

- DENDRAL (Dendritic Algorithm) was one of the earliest expert systems developed in the 1960s.
- It aimed to interpret mass spectrometry data to identify chemical compounds.
- **DENDRAL has following features:**
  - Knowledge representation: Production rules and algorithm for generating graph structure are constructed by META- DENDRAL. META DENDRAL is a program which uses learning techniques to construct rules for an expert system automatically.
  - Reasoning: DENDRAL uses forward chaining.
  - Heuristics: DENDRAL uses generate and test method.
  - Explanation: the user can supply the information and the system can request information as required.
- **Procedure :**

  - Input:

  a. Spectra data obtained from mass spectrometry serves as input.

  - **Preliminary Analysis:**

  a. Determines necessary compounds based on the spectral data.

  b. Identifies forbidden compounds using both spectral data and expert knowledge.

  - Generate and Test:

  a. Structure Enumerator:

  i. Generates all possible compounds based on necessary and forbidden lists.

  ii. Outputs chemical formulas for potential compounds.

  b. Spectra Synthesizer:

  i. Generates simulated spectra data for each potential compound.

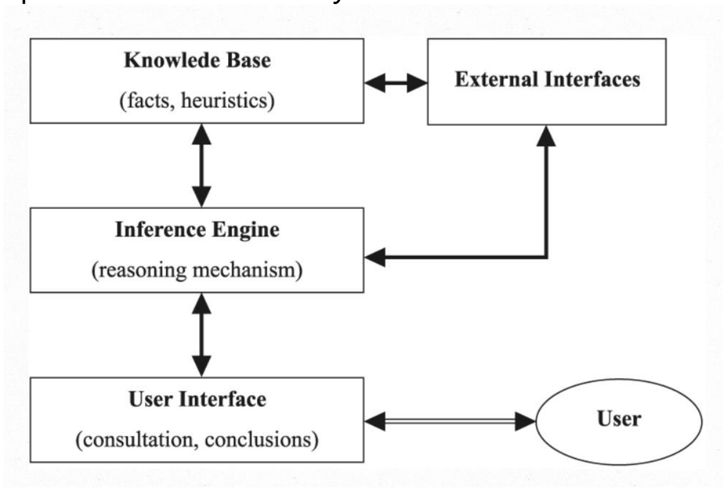
  c. Matcher:

  i. Compares synthesized spectra with actual spectra.

  ii. Selects the compound with the best fit as the identification.

  - Working:

- The mass spectrometer subjects the sample to high-energy electrons, causing it to fragment into charged ions of various sizes.
- The ions are sorted by passing them through a magnetic field, which deflects ions with higher charge-to-mass ratios more than those with lower ratios.
- This process results in a mass spectrogram showing the relative abundance of each ion, which is used for compound identification by DENDRAL.



- **Disadvantages of DENDRAL :**

  - **Complexity:** DENDRAL's structure enumeration process can become computationally intensive and time-consuming for large and complex molecules.

  - **Limitations in Handling Uncertainty:** may struggle to handle uncertainty and variability in spectral data, leading to inaccuracies in compound identification.

  - **Dependence on Input Quality:** accuracy of DENDRAL's identification heavily relies on the quality and reliability of the input mass spectrometry data.

  - **Domain Specificity:** DENDRAL's expertise is limited to organic chemistry, making it less suitable for identifying compounds outside this domain.

  - **Expertise Dependency:** DENDRAL's performance is contingent on the availability of expert knowledge for defining necessary and forbidden compounds, which may not always be comprehensive or readily accessible.

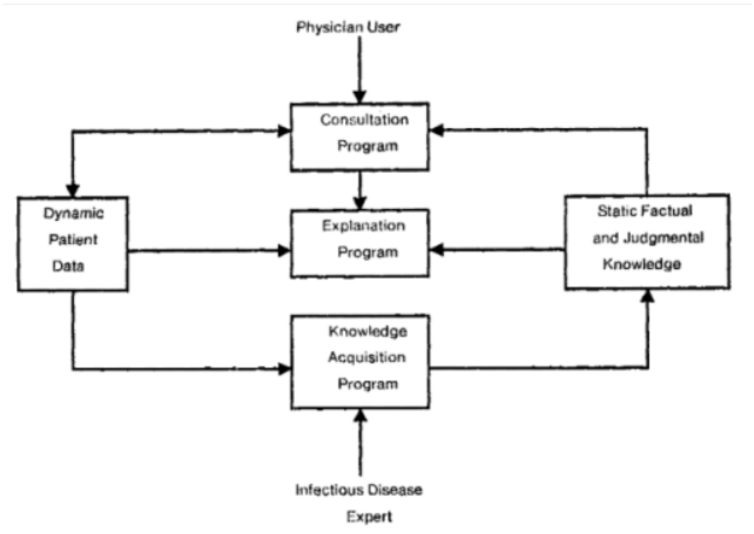
MYCIN:

- MYCIN: Medical Diagnosis Expert System
- Developed at Stanford University in the mid-1970s.
- Designed for medical diagnosis of bacterial and meningitis infections.
- **MYCIN Features:**
  - Domain: Focuses on medical diagnosis for bacterial and meningitis infections.
  - Task: Interviews physicians, makes diagnosis, and recommends therapies.
  - Input: Receives answers to queries about symptoms and patient history.
  - Output: Provides an ordered set of diagnoses and therapy recommendations.

- Architecture: Utilizes rule-based exhaustive backward chaining with uncertainty handling.
- Tools: Programmed in LISP, with a shell called EMYCIN (empty MYCIN).
- Results: While not in general use, MYCIN was groundbreaking in the field of diagnostic consultation systems.

- **Procedure :**
  - **Knowledge Acquisition:**
    - Medical experts specializing in bacterial infections provide domain knowledge.
    - Input includes causes, symptoms, and other relevant information.
  - **Knowledge Base Update:**
    - Expert knowledge is organized and updated in the knowledge base (KB) of MYCIN.
  - **Problem Identification:**
    - A doctor presents MYCIN with a new problem: identifying bacterial presence.
    - Details of the patient, including symptoms and medical history, are inputted.
  - **Questionnaire:**
    - MYCIN prompts the patient to fill out a questionnaire for general information.
    - Information like gender, age, etc., is collected to aid diagnosis.
  - **Solution Generation:**
    - MYCIN applies IF-THEN rules using its inference engine on the collected data.
    - It analyzes facts stored within the KB to derive a solution.
  - **Response:**
    - MYCIN provides a response to the patient through the user interface.
    - Diagnosis, treatment recommendations, or further actions are communicated.
- **Case Study Analysis:**
  - **Problem:** Diagnosing bacterial infections in patients.
  - **Solution:**
    - Data Input: MYCIN receives patient data, symptoms, and medical history.
    - Knowledge Representation: Uses IF-THEN rules based on expert knowledge to analyze data.
    - Inference Engine: Derives diagnosis by applying rules to patient information.
    - Output: Provides diagnosis and treatment recommendations through the user interface.
  - **Outcome:**
    - MYCIN efficiently diagnoses bacterial infections based on patient data.
    - Enables doctors to make informed decisions and provide timely treatment.
    - Demonstrates the effectiveness of expert systems in medical diagnosis and decision-making.
- **Disadvantages of MYCIN:**
  - **Limited Scope:** MYCIN is specialized for bacterial infections and may not be applicable to other medical conditions.

- **Dependence on Input:** Accuracy relies on the completeness and accuracy of input data provided by users.
- **Lack of Human Judgment**: Cannot replicate the intuition and judgment of experienced human doctors.
- **Overreliance on Rules:** May overlook rare or atypical cases not covered by existing rules.
- **Complexity:** Maintenance and updating of the knowledge base require continuous effort and expertise.


Physician User
Consultation Program
Dynamic Patient Data
Explanation Program
Static Factual and Judgmental Knowledge
Knowledge Acquisition Program
Infectious Disease Expert

Types of Expert System:

- **Rule-Based ES:**
  - Uses IF-THEN rules for decision-making.
  - Simple and transparent reasoning process.
  - Effective for well-defined problems with clear rules.
- **Fuzzy Logic ES:**
  - Deals with uncertainty and imprecision in decision-making.
  - Allows for gradual transition between true and false states.
  - Suitable for problems with vague or ambiguous information.
- **Frame-Based ES:**
  - Organizes knowledge into hierarchical structures called frames.
  - Captures relationships between objects and concepts.
  - Useful for representing complex, interconnected domains.
- **Neural ES:**
  - Inspired by the structure and function of the human brain.
  - Learns from data patterns through training.
  - Excels in pattern recognition and classification tasks.
- **Neuro-Fuzzy ES:**
  - Integrates fuzzy logic and neural networks.
  - Combines fuzzy reasoning with adaptive learning capabilities.
  - Provides robustness and flexibility for handling complex problems.

GAME PLAYING:

- **Game Playing VS Search :**

Aspect	Game Playing	Search
Objective	To make optimal moves to win a game.	To find solutions or answers to a problem.
Domain	Limited to games with defined rules.	Applicable to various domains and scenarios.
Strategy	Involves decision-making based on game rules.	Involves exploration of possible solutions.
Outcome	Winning or losing the game.	Finding optimal or satisfactory solutions.
Complexity	Deals with complex game states and strategies.	Deals with diverse problem-solving scenarios.
Example	Chess, Go, Tic-Tac-Toe.	Pathfinding, route optimization, scheduling.

- **Types of Games :**

	Deterministic	**Chance Moves**
Perfect information	Chess, Checkers, go, Othello	Backgammon, monopoly
Imperfect information	Battleships, blind, tic-tac-toe	Bridge, poker, scrabble, nuclear war

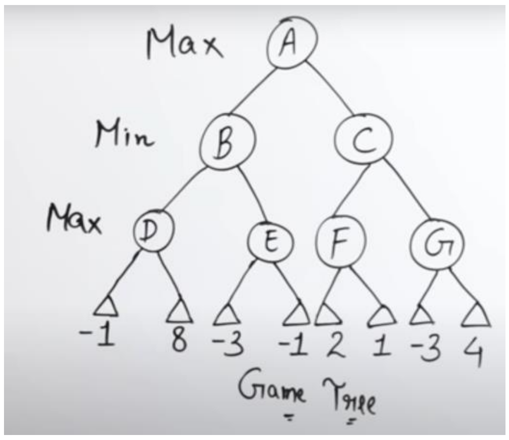
- **Perfect information:** A game with the perfect information is that in which agents can look into the complete board. Agents have all the information about the game, and they can see each other moves also. Examples are Chess, Checkers, Go, etc.
- **Imperfect information:** If in a game agents do not have all information about the game and not aware with what's going on, such type of games are called the game with imperfect information, such as tic-tac-toe, Battleship, blind, Bridge, etc.
- **Deterministic games:** Deterministic games are those games which follow a strict pattern and set of rules for the games, and there is no randomness associated with them. Examples are chess, Checkers, Go, tic-tac-toe, etc.
- **Non-deterministic games:** Non-deterministic are those games which have various unpredictable events and has a factor of chance or luck. This factor of chance or luck is introduced by either dice or cards. These are random, and each action response is not fixed. Such games are also called as stochastic games. Example: Backgammon, Monopoly, Poker, etc.

- **MINIMAX Algorithm :**

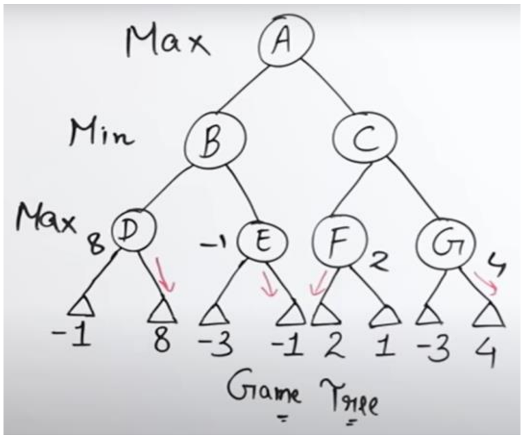
- Mini-max algorithm: Recursive or backtracking algorithm used in decision-making and game theory.
- Provides optimal move assuming opponent plays optimally.
- Utilizes recursion to search through game tree.
- Mostly used in game playing AI like Chess, Checkers, tic-tac-toe.
- Two players: MAX (maximizes) and MIN (minimizes) benefits.

- MAX seeks maximum benefit, MIN seeks minimum benefit.
- Performs depth-first search of complete game tree.
- Backtracks tree using recursion.

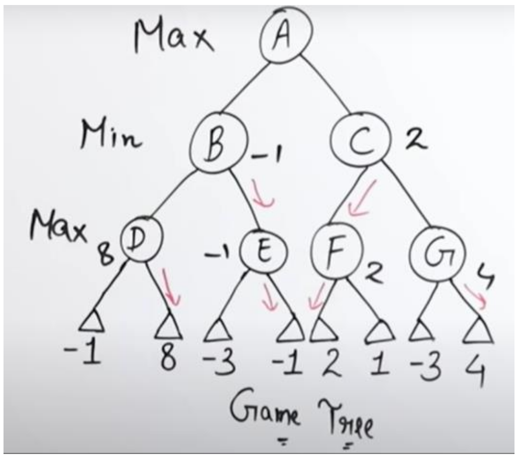
- **Working of MINIMAX Algorithm :**
  - Generates entire game tree, starting from initial state.
  - Applies utility function to terminal states, assigning values to nodes.
  - Maximizing player (Max) selects nodes with highest values.
  - Minimizing player (Min) selects nodes with lowest values.
  - Alternates between Max and Min players until reaching terminal states.
  - Backtracks, propagating values from terminal states up through the tree.
  - Final decision made at root node based on computed values.
  - Determines optimal move for each player assuming opponent plays optimally.



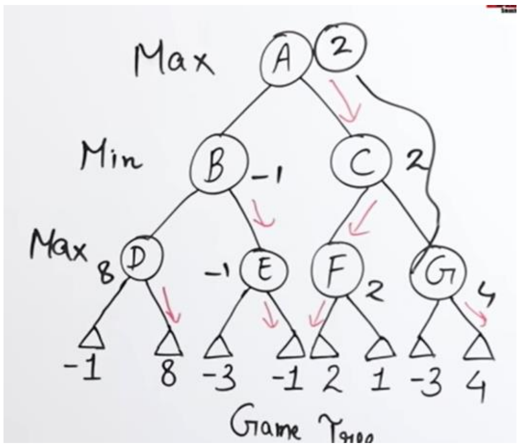
Step 1



Step 2



Step 3



Step 4

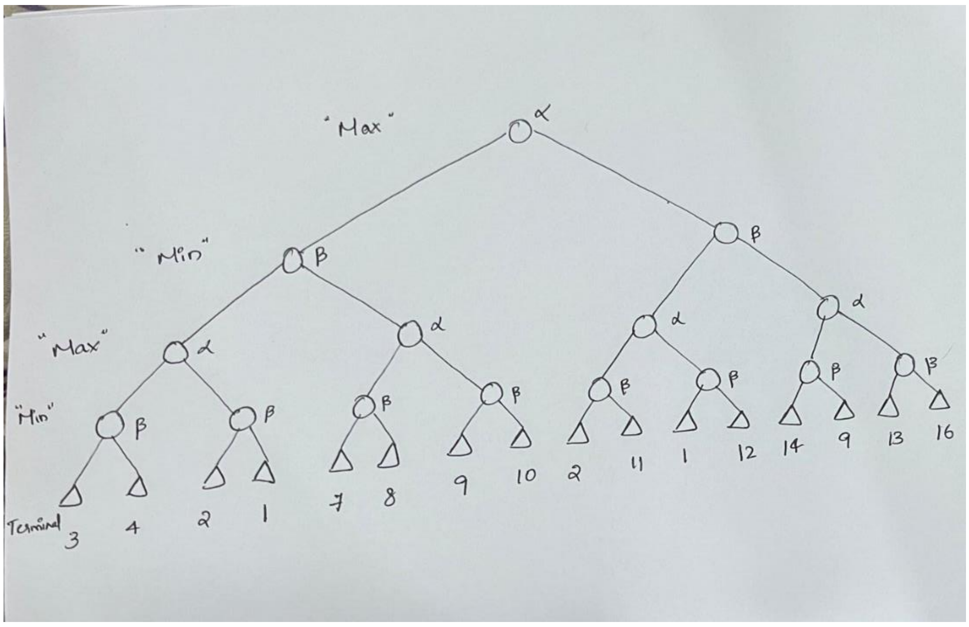
- **Properties of Mini-Max algorithm:**
  - **Complete-** Min-Max algorithm is Complete. It will definitely find a solution (if exist), in the finite search tree.
  - **Optimal-** Min-Max algorithm is optimal if both opponents are playing optimally.
  - **Time complexity-** As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is **O(b^m)**, where b is branching factor of the game-tree, and m is the maximum depth of the tree.
  - **Space Complexity-** Space complexity of Mini-max algorithm is also similar to DFS which is **O(bm)**.

- **Limitations of MINIMAX Algorithm:**
  - **Slow Performance:** Minimax algorithm becomes slow for complex games like Chess and Go.
  - **High Branching Factor:** Complex games have a large number of possible moves, leading to extensive exploration of the game tree.
  - **Many Choices:** Players have numerous options to consider, increasing computation time.
  - **Improvement with Alpha-Beta Pruning:** Alpha-beta pruning addresses this limitation by efficiently pruning branches of the game tree, reducing search time.

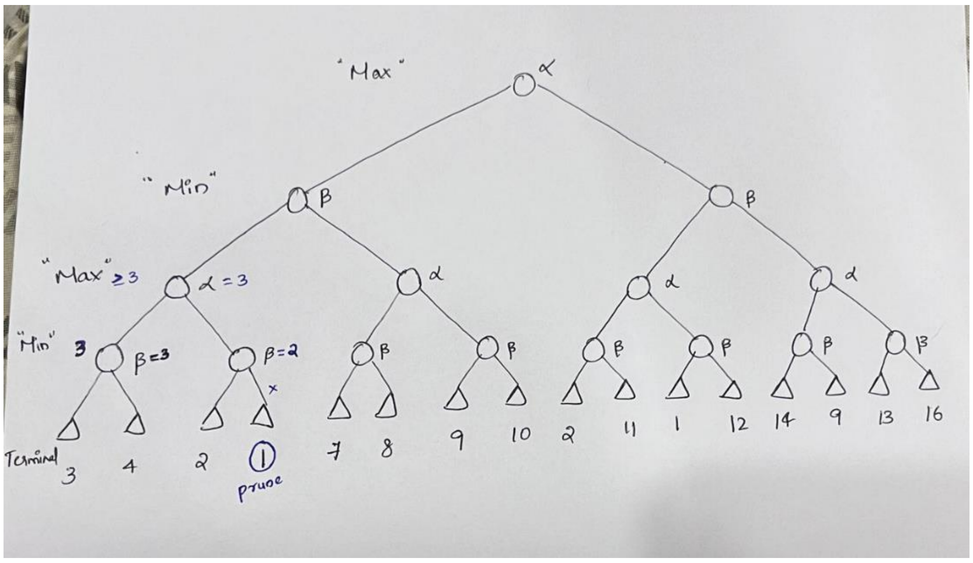
- **Alpha-Beta Pruning:**
  - **Alpha-beta Pruning:** Optimization technique for MiniMax algorithm.
  - **Reduces Search Space**: Cuts down the number of game states examined, reducing computation time.
  - **Threshold Parameters**: Alpha (highest-value choice for Maximizer), Beta (lowest-value choice for Minimizer).
  - **Pruning**: Eliminates nodes that do not affect final decision, improving efficiency.
  - **Alpha and Beta Values:** Initialized to $-\infty$ and $+\infty$ respectively.
  - **Same Decision as MiniMax:** Returns the same optimal move as MiniMax algorithm, but faster.
  - **Condition :** $\alpha >= \beta$

- **Key points about alpha-beta pruning:**
  - The Max player will only update the value of alpha.
  - The Min player will only update the value of beta.
  - While backtracking the tree, the node values will be passed to upper nodes instead of values of alpha and beta.
  - We will only pass the alpha, beta values to the child nodes.

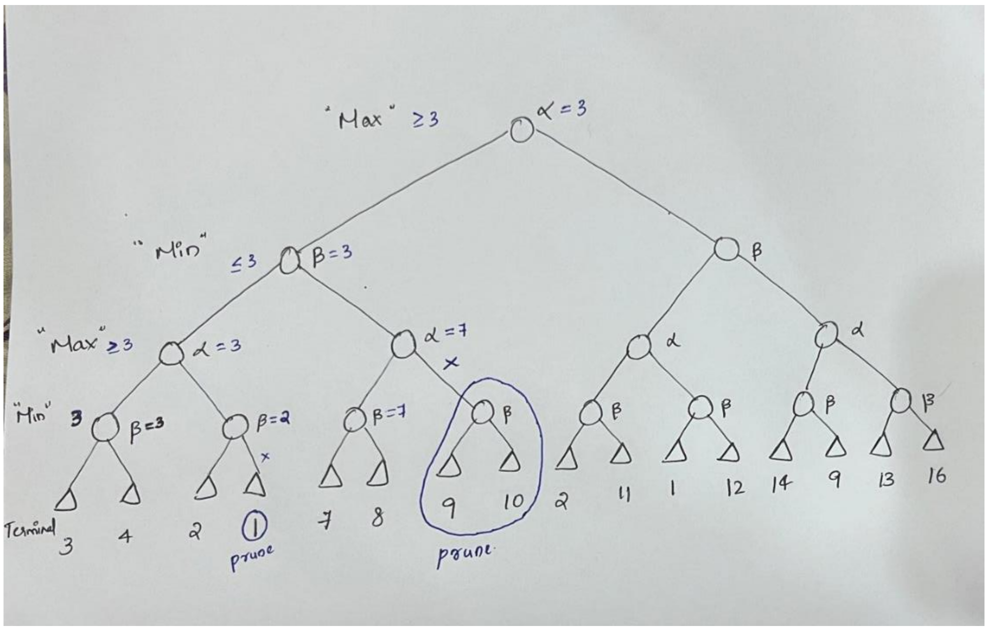
- **Working of Alpha-Beta Pruning:**
  - Max player starts at root node A with α = -∞ and β = +∞.
  - Values of α and β propagate down the tree.
  - At each node, α and β are updated based on Max and Min player turns.
  - Pruning occurs when α >= β, eliminating unnecessary branches.
  - Backtracking updates α and β values as algorithm progresses.
  - Optimal move determined with reduced search space.
  - Final result obtained efficiently, same as MiniMax algorithm.
  - Pruned nodes shown as never computed, reducing computation time.



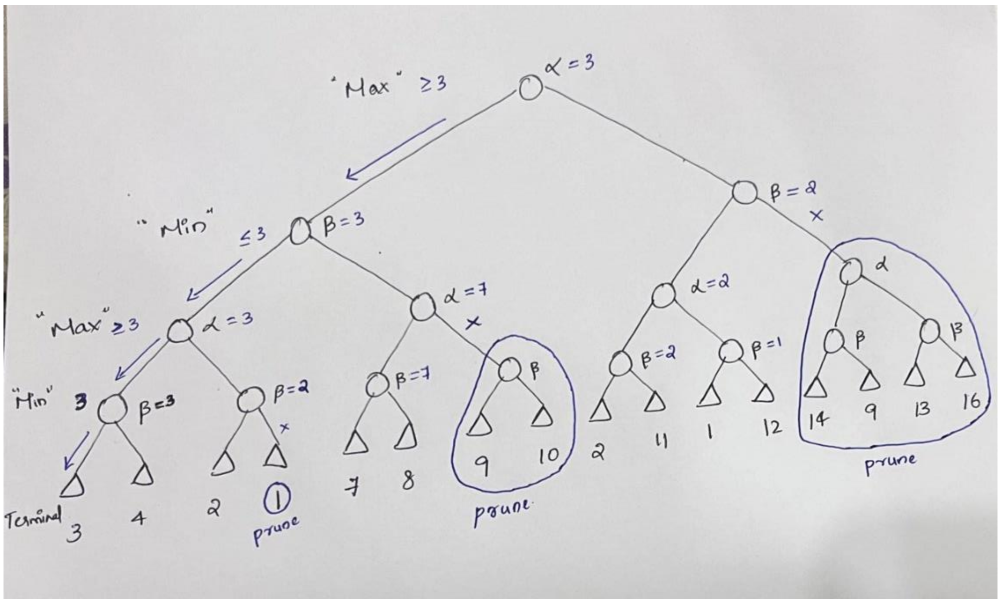
Step 1



Step 2



Step 3



Step 4

MODERN AI APPLICATIONS:

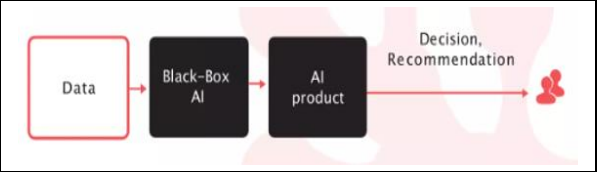
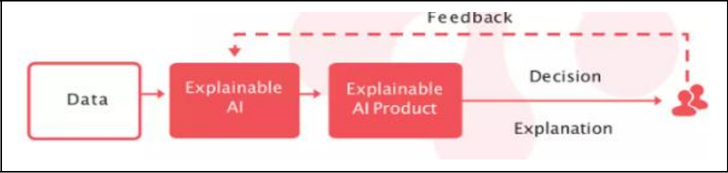
Modern AI applications utilize machine learning, natural language processing, computer vision, and other AI subfields to solve complex problems, enhance efficiency, and innovate solutions across diverse sectors, leveraging recent advancements in AI technologies and computing power.

- **Applications of AI :**
  - **E-commerce :** Recommender systems for personalized product suggestions, chatbots for customer service, and fraud detection algorithms.
  - **Education:** Adaptive learning platforms, intelligent tutoring systems, plagiarism detection software, and virtual classroom assistants.

- **Social Media:** Content recommendation algorithms, sentiment analysis tools, personalized advertising, and chatbots for customer engagement.
- **Healthcare:** Medical image analysis for diagnosis, predictive analytics for disease prognosis, virtual health assistants, and drug discovery algorithms.
- **Finance**: Algorithmic trading, fraud detection, credit scoring models, and customer service chatbots.
- Transportation: Autonomous vehicles, route optimization algorithms, predictive maintenance systems, and traffic management solutions.
- **Manufacturing:** Predictive maintenance, quality control using computer vision, autonomous robots for assembly, and supply chain optimization.
- **Entertainment:** Content recommendation algorithms for streaming platforms, personalized gaming experiences, and AI-generated music and art.

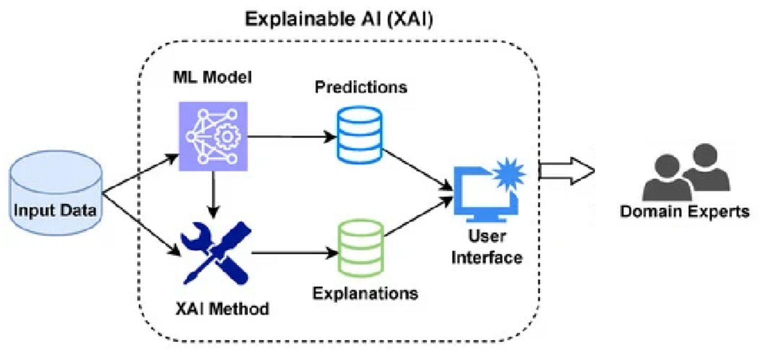
EXPLAINABLE AI (XAI):

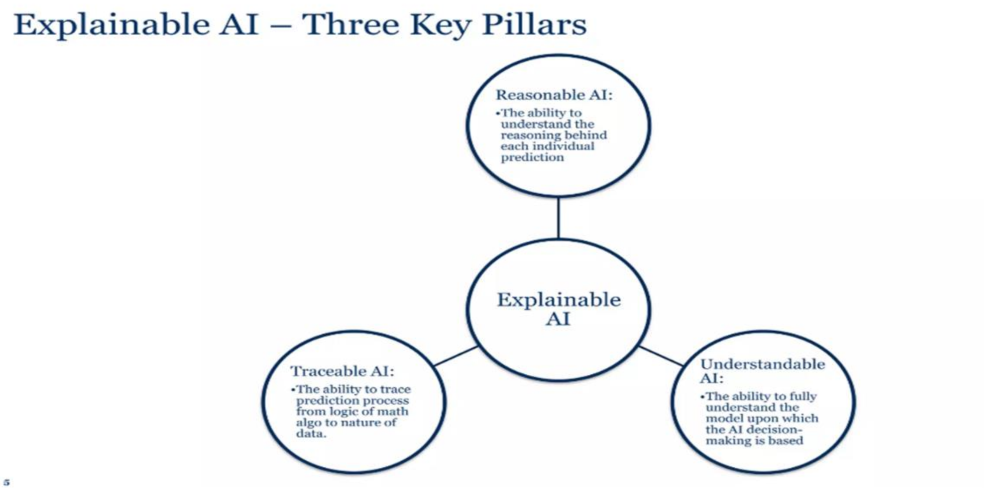
- It is an AI system that explains their decision making which is referred as Explainable AI or XAI.
- The goal of XAI is to provide verifiable explanations of how machine learning systems make decisions and let humans to be in the loop.
- Two ways to provide XAI:
  - Use ML approaches like decision tee, knowledge graphs and similarity models.
  - Develop new approaches to explain complicated neural networks.
- **Black Box AI VS Explainable AI:**

Aspect	Black Box AI	Explainable AI (XAI)
Transparency	Opacity, inner workings not understood	Transparency, decisions are interpretable
Interpretability	Difficult to understand reasoning	Decisions and behaviors are explainable
Trust	Lower trust due to lack of transparency	Higher trust due to understandable decisions
Accountability	Challenging to assign responsibility	Easier to assign responsibility for decisions
Application	Used in scenarios where explanation not critical	Essential in critical applications where transparency is crucial
Diagram	 Data Black-Box AI AI product Decision, Recommendation	 Feedback Data Explainable AI Explainable AI Product Decision Explanation

- Life Cyle of XAI :
  - **Data Understanding:** XAI techniques are used to analyze the training data for biases or fairness issues that might impact the AI model's decisions.
  - **Model Development & Training:** XAI helps understand how the model learns from the data, potentially revealing factors influencing its decision-making process.

- **Deployment & Monitoring:** XAI provides explanations for the AI model's outputs, increasing trust and transparency for users and stakeholders.





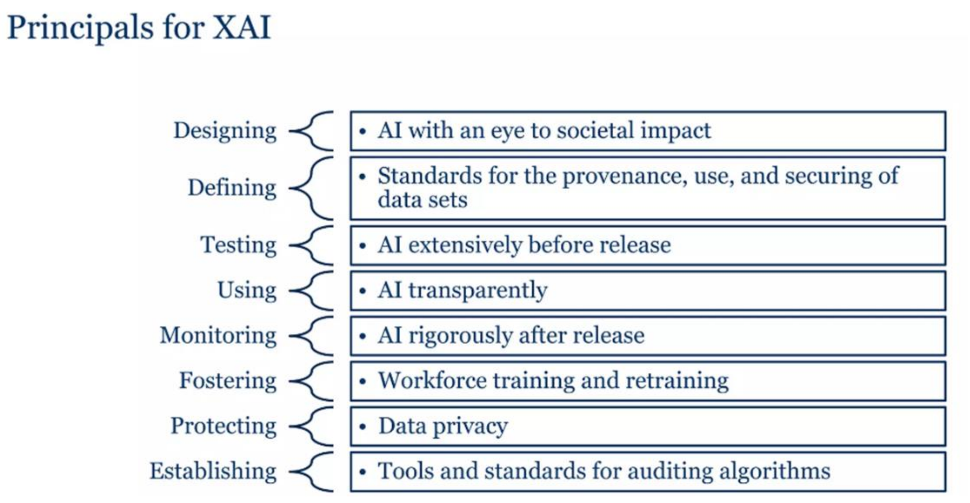
Explainable AI – Three Key Pillars

Reasonable AI:
- The ability to understand the reasoning behind each individual prediction

Traceable AI:
- The ability to trace prediction process from logic of math algo to nature of data.

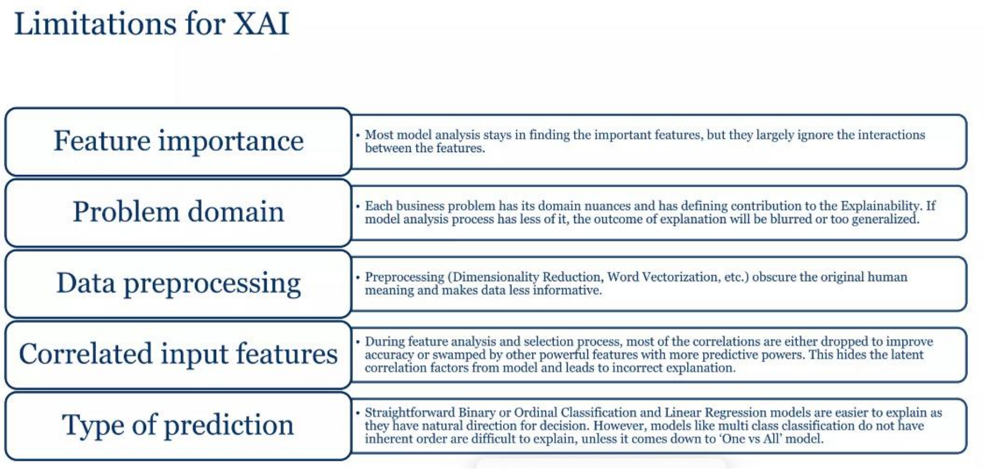
Understandable AI:
- The ability to fully understand the model upon which the AI decision-making is based

5



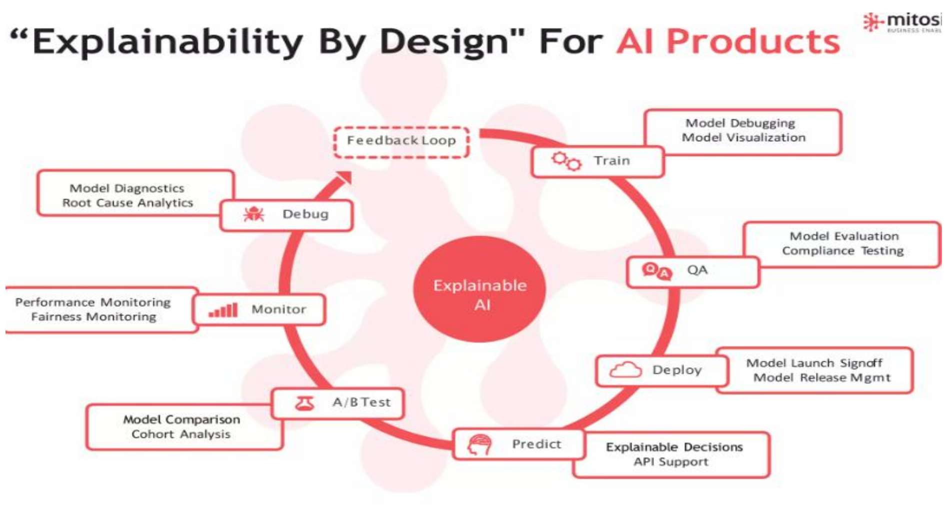
Principals for XAI

Designing	• AI with an eye to societal impact
Defining	• Standards for the provenance, use, and securing of data sets
Testing	• AI extensively before release
Using	• AI transparently
Monitoring	• AI rigorously after release
Fostering	• Workforce training and retraining
Protecting	• Data privacy
Establishing	• Tools and standards for auditing algorithms



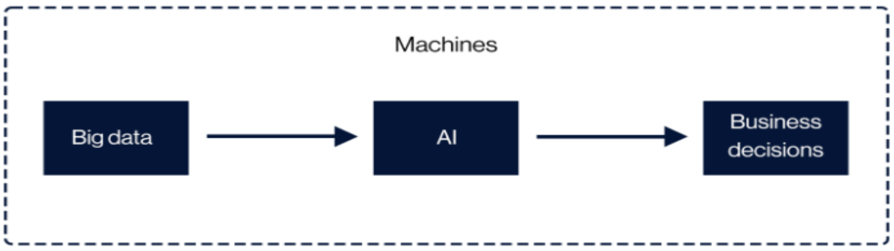
Limitations for XAI

Feature importance	• Most model analysis stays in finding the important features, but they largely ignore the interactions between the features.
Problem domain	• Each business problem has its domain nuances and has defining contribution to the Explainability. If model analysis process has less of it, the outcome of explanation will be blurred or too generalized.
Data preprocessing	• Preprocessing (Dimensionality Reduction, Word Vectorization, etc.) obscure the original human meaning and makes data less informative.
Correlated input features	• During feature analysis and selection process, most of the correlations are either dropped to improve accuracy or swamped by other powerful features with more predictive powers. This hides the latent correlation factors from model and leads to incorrect explanation.
Type of prediction	• Straightforward Binary or Ordinal Classification and Linear Regression models are easier to explain as they have natural direction for decision. However, models like multi class classification do not have inherent order are difficult to explain, unless it comes down to 'One vs All' model.

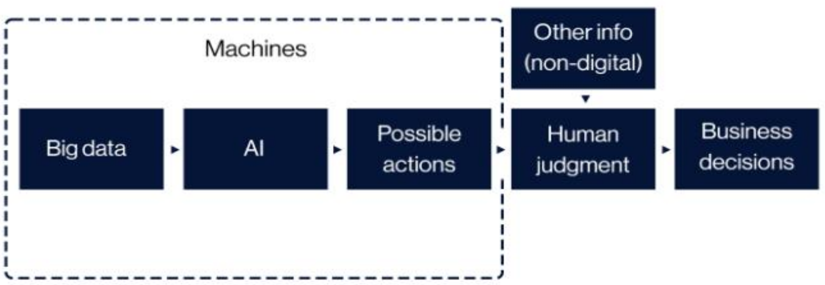


DECISION MAKING AI:

- Involves data processing tasks, such as trend analysis and decision suggestions, conducted by AI systems.
- AI platforms replace or assist humans in quantifying data for more accurate predictions and decisions.
- Utilizes AI algorithms to analyze data patterns and make informed choices, enhancing decision-making processes.



- **Role of AI in Decision Making:**
  - **Continuous Learning:** AI continually learns from data-driven decisions, improving its predictive capabilities over time.
  - **Data-Driven Models:** AI utilizes data collections to build models adept at making accurate predictions and categorizations.
  - **Real-Time Decision Making:** These models can analyze live data in real-time, providing instant predictions, categorizations, and recommendations.
  - **Enhanced Decision Making:** Businesses benefit from AI's ability to make precise, data-driven decisions quickly and efficiently, leading to improved commercial outcomes.



PREDICTIVE ANALYTICS (PA):

- Predictive analytics involves analyzing historical data to predict future outcomes and trends.
- PA enables organizations to make informed decisions, anticipate future events, mitigate risks, and optimize operations.
- **Types of PA:**
  - **Clustering Model:**
    - Definition: Groups similar data points into clusters based on their characteristics or attributes.
    - Purpose: Identifies patterns and structures within data, aiding in segmentation and targeted marketing.
  - **Classification Model:**
    - Definition: Predicts the category or class to which a new data point belongs.
    - Purpose: Used for tasks like spam detection, sentiment analysis, and disease diagnosis.
  - **Time-Series Model:**
    - Definition: Analyzes time-ordered data to forecast future values based on past observations.
    - Purpose: Helps predict trends, patterns, and seasonal variations, useful in financial forecasting, demand forecasting, and weather prediction.
- **Applications :**
  - **Finance:**
    - Fraud detection and prevention.
    - Credit risk assessment and scoring.
    - Stock market prediction and portfolio optimization.
  - **Healthcare:**
    - Disease diagnosis and prognosis.
    - Patient readmission prediction.
    - Drug discovery and personalized medicine.
  - **Telecommunications:**
    - Churn prediction to retain customers.
    - Network fault prediction and optimization.
    - Customer segmentation for targeted promotions.
  - **Marketing:**
    - Customer behavior analysis and segmentation.
    - Campaign optimization and ROI prediction.
    - Customer lifetime value prediction.
  - **Transportation:**
    - Route optimization for logistics and transportation.
    - Predictive maintenance for vehicles and fleets.
    - Demand forecasting for ride-sharing and public transportation services.

ADAPTIVE ANALYTICS IN AI:

- **Definition:** Adaptive Analytics in AI refers to the ability of systems to dynamically adjust and learn from new data, ensuring that models and algorithms remain relevant and effective in changing environments.
- **Key Components:**
  - **Real-time Learning:** Systems continuously learn and adapt in real-time as new data becomes available.
  - **Dynamic Models:** Adaptive analytics systems incorporate flexibility to adjust their models based on evolving patterns in the data.
  - **Context Awareness:** Understanding the context in which decisions are made allows for more adaptive and contextually relevant analytics.
- **Applications:**
  - **User Experience:** Adaptive analytics is applied in personalized content recommendations and user interface customization.
  - **Cybersecurity:** Systems adapt to new threats by learning from historical attack patterns and adjusting security protocols.
  - **Industrial Processes:** Adaptive analytics optimize manufacturing processes by adjusting parameters based on real-time data.