Exploring the foundation of AI: The Role of Discrete Mathematics in Machine Learning



|  |  |
| --- | --- |
| **NAME** | **STUDENT ID** |
| Abdul Rafay Zahid | 65540 |
| Emaz Ali Khan | 65566 |
| Sumaika Asif | 65651 |

DATE OF SUBMISSION:

Exploring the foundation of AI: The Role of Discrete Mathematics in Machine Learning

**SUBMITTED BY**

Abdul Rafay Zahid (65540)

Emaz Ali Khan (65566)

Sumaika Asif (65651)

**SUPERVISED BY**

Sir. Muhammad Talha



REPORT SUBMITTED TO THE FACULTY OF COMPUTING, KARACHI INSTITUTE OF ECONOMICS AND TECHNOLOGY, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SPRING 2025

# Introduction: Discrete Mathematics as the Pillar of AI

# Artificial Intelligence (AI) and Machine Learning (ML) are based on ideas from discrete mathematics, which deals with separate and countable items. It includes important topics like logic, sets, functions, relations, and graphs. These topics are very useful in AI. For example, logic helps machines make decisions by following clear rules, similar to how people solve problems. Set theory helps organize and group data, which is important in tasks like choosing the right features and managing databases. Functions show how inputs lead to outputs, which is the basic idea behind machine learning models. Relations explain how different things are connected, which is helpful in building recommendation systems and knowledge maps. Overall, discrete mathematics helps in designing smart and reliable AI systems. As AI continues to improve, having a good understanding of discrete math becomes more important for creating machines that can think, learn, and make good decisions.

# Set Theory and Relations: Organizing Data

# Set theory and relations are very important in organizing and understanding data in Artificial Intelligence (AI) and Machine Learning (ML). A set is simply a group of unique items, and in AI, these items can be data points, features, or labels. Set theory helps us perform operations like combining sets (union), finding common elements (intersection), or checking if one set is part of another (subset). These ideas are useful when we structure or filter data. In clustering, which is a type of unsupervised learning, algorithms like K-Means and DBSCAN group similar data points into clusters. Each cluster can be seen as a subset of the whole dataset where items are alike in some way. In classification, which is a type of supervised learning, algorithms like Decision Trees or Support Vector Machines sort data points into predefined categories. Each category is like a set containing certain types of data. Relations, on the other hand, show how different items are connected. For example, in a recommendation system, a relation might show which users liked which products. These relationships help in building systems like collaborative filters or social networks. Overall, using sets and relations helps AI systems understand data better, make smart decisions, and work more efficiently.

# ChatGPT Image May 24, 2025, 06_22_58 PM

# Propositional Logic and Rule-Based Systems

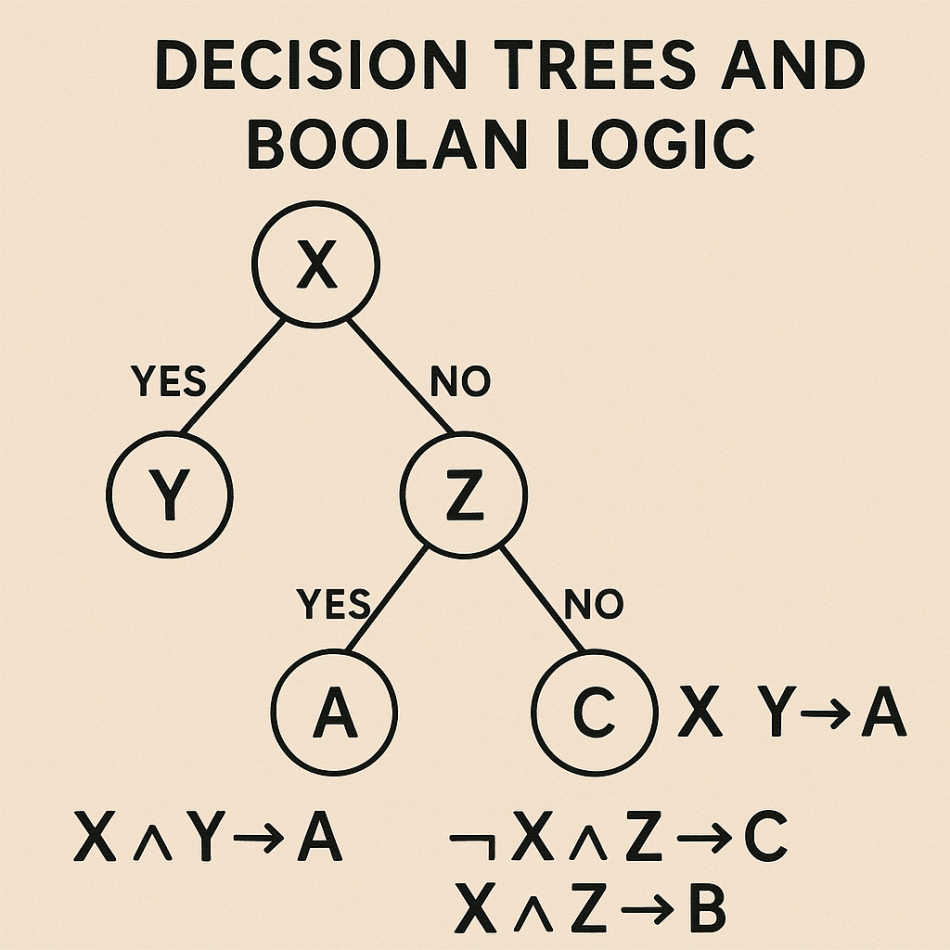
Propositional logic is an important part of artificial intelligence, especially in systems that use rules to make decisions. It deals with statements that can either be true or false, and uses logical words like AND, OR, NOT, and IF-THEN to combine them. Rule-based systems use this logic to act like human decision-making. They work by using IF-THEN rules, where a certain condition leads to a certain conclusion. For example, the rule “IF Fever AND Cough THEN Flu” means that if both symptoms are present, the system will decide the person likely has the flu.

This kind of logic is used in expert systems for tasks like medical diagnosis, in games to control character behavior, in legal systems to apply rules of law, and in robots or machines to make quick decisions. Rule-based systems are easy to understand, can be changed easily, and allow users to see how decisions were made. However, they can become hard to manage if there are too many rules, and they cannot handle more complex ideas that involve variables, which require more advanced logic. Overall, propositional logic helps AI systems think in a clear and logical way, and it has been a key part of building smart systems.

# ChatGPT Image May 24, 2025, 06_14_46 PM

# Decision Trees and Boolean Logic

Decision trees are simple yet powerful tools in machine learning used for classification (grouping) and regression (predicting values). They work like flowcharts, where each node asks a yes/no question using Boolean logic. For example: “IF Outlook is Sunny AND Humidity is High, THEN Play = No.” Each path through the tree combines such rules to reach a decision. Popular decision tree algorithms include **ID3** (uses information gain), **C4.5** (handles different data types), and **CART** (works for both classification and regression). They’re widely used in tasks like medical diagnosis, credit scoring, and spam detection. Decision trees are easy to understand, require little data preparation, and give quick results. However, they can overfit if too complex and may change completely with small data changes. Overall, they blend simple logic with effective learning, making them ideal when both accuracy and clarity are important.



# Graph Theory in Neural Networks and Recommendation Systems

Graph theory helps AI model relationships using nodes (entities) and edges (connections). Neural networks can be seen as directed graphs, where neurons are nodes and edges show data flow. This applies to models like CNNs, RNNs, and Transformers.

In recommendation systems, bipartite graphs connect users to items (e.g., movies), based on interactions like clicks or ratings. These help suggest new items through pattern analysis.

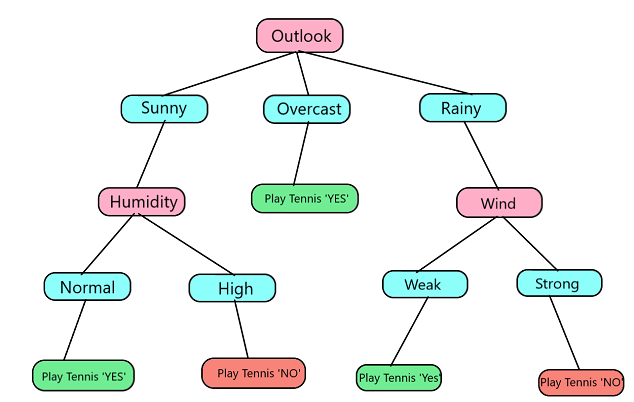
Graph algorithms like shortest path (Dijkstra), PageRank, and clustering support tasks like navigation, ranking, and community detection.

Knowledge graphs use nodes and labeled edges to show real-world relationships (e.g., Elon Musk → founder\_of → SpaceX), aiding search and language understanding.

# ChatGPT Image May 24, 2025, 06_20_00 PM

# Case Study

To demonstrate the practical application of discrete mathematics in AI, a simple case study was developed using C#. The system predicts whether a person should play tennis based on weather conditions by simulating a decision tree. The user provides inputs for outlook (Sunny, Overcast, Rain), humidity (High, Normal), and wind conditions (true/false). Based on these values, the program applies Boolean logic through nested conditional statements to reach a decision. For example, if the outlook is "Sunny" and the humidity is "High," the program outputs "Don't Play." If the outlook is "Overcast," the output is always "Play," and if it is "Rain" with no wind, the output is "Play"; otherwise, "Don't Play." This case study clearly illustrates how decision-making logic can be implemented using propositional logic and control structures, which are rooted in discrete mathematics. It also emphasizes how interpretable models like decision trees can be coded manually for small-scale expert systems, forming a bridge between theoretical logic and real-world AI applications.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Wind** | **Play Tennis** |
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

# Why Discrete Structures Are Foundational to AL Models?

Discrete structures provide the core building blocks for artificial intelligence. They offer a clear and logical framework for organizing, analyzing, and making decisions with data. **Logic** allows AI to perform reasoning, draw conclusions, and simulate human thinking. For example, rule-based systems and decision-making models rely on logical statements to function properly.

**Set theory** helps group and classify data—this is essential for machine learning tasks like clustering or labeling data into categories. **Functions** define the relationship between inputs and outputs, which is the heart of how AI models make predictions. Every model, from simple linear regression to deep learning, is based on some form of function mapping.

**Relations** show how different entities connect or interact, which is crucial in building graphs, networks, and databases. These are widely used in recommendation systems, knowledge graphs, and social network analysis.

Together, these discrete mathematical tools make AI models structured, efficient, and easier to understand. They ensure AI systems are not just powerful but also logical and trustworthy, which is why discrete structures are so important in building intelligent solutions to real-world problems.

# Conclusion

# The role of discrete mathematics in artificial intelligence and machine learning is both foundational and indispensable. Core concepts such as logic, sets, functions, relations, and graph theory provide the structural backbone for intelligent systems to process data, learn from it, and make informed decisions. Whether it's decision trees applying Boolean logic, neural networks structured as directed graphs, or rule-based systems relying on propositional logic, discrete structures allow complex AI systems to be both interpretable and mathematically robust. As AI continues to evolve, a deep understanding of these discrete mathematical foundations will remain essential for designing reliable, scalable, and ethical models. This integration of theory and application underscores the importance of discrete mathematics not only as a theoretical discipline but as a driving force in the future of intelligent technology.

# References

1. Gadish, N. (n.d.). Mathematics for Machine Learning and Artificial Intelligence. University of Pennsylvania. Retrieved from [https://www.sas.upenn.edu/~ngadish/notes/18204/MLAI.pdf](https://www.sas.upenn.edu/~ngadish/notes/18204/MLAI.pdf" \t "_new)
2. IJSRMSS. (2023). The Role of Discrete Mathematics in Machine Learning and Artificial Intelligence. ISROSET. Retrieved from [https://www.isroset.org/pub\_paper/IJSRMSS/10-ISROSET-IJSRMSS-09633.pdf](https://www.isroset.org/pub_paper/IJSRMSS/10-ISROSET-IJSRMSS-09633.pdf" \t "_new)
3. PNR Journal. (2023). The Application of Discrete Structures in AI and ML. Retrieved from [https://www.pnrjournal.com/index.php/home/article/download/7809/10369/9480](https://www.pnrjournal.com/index.php/home/article/download/7809/10369/9480" \t "_new)

# Appendices

# Code:

using System;

class DecisionTreePlayTennis

{

static void Main()

{

// Prompt user to enter the weather condition inputs

Console.WriteLine("=== Tennis Play Decision Predictor ===\n");

Console.Write("Enter Outlook (Sunny, Overcast, Rain): ");

string outlook = Console.ReadLine().Trim();

Console.Write("Enter Temperature (Hot, Mild, Cool): ");

string temperature = Console.ReadLine().Trim(); // Optional input for completeness

Console.Write("Enter Humidity (High, Normal): ");

string humidity = Console.ReadLine().Trim();

Console.Write("Is it Windy? (true/false): ");

bool windy;

// Try parsing the input safely

while (!bool.TryParse(Console.ReadLine().Trim(), out windy))

{

Console.Write("Invalid input. Please enter 'true' or 'false' for Windy: ");

}

// Call the method that applies decision logic

string decision = PredictPlayTennis(outlook, humidity, windy);

// Display the final decision

Console.WriteLine($"\n=> Decision: {decision}");

}

/// <summary>

/// This function acts like a decision tree to determine if one should play tennis

/// based on outlook, humidity, and wind conditions.

/// </summary>

static string PredictPlayTennis(string outlook, string humidity, bool windy)

{

// Decision Tree Logic

if (outlook.Equals("Sunny", StringComparison.OrdinalIgnoreCase))

{

// If sunny, check humidity

if (humidity.Equals("High", StringComparison.OrdinalIgnoreCase))

{

return "Don't Play (Sunny + High Humidity)";

}

else if (humidity.Equals("Normal", StringComparison.OrdinalIgnoreCase))

{

return "Play (Sunny + Normal Humidity)";

}

else

{

return "Invalid humidity input.";

}

}

else if (outlook.Equals("Overcast", StringComparison.OrdinalIgnoreCase))

{

// Always play if overcast

return "Play (Overcast conditions)";

}

else if (outlook.Equals("Rain", StringComparison.OrdinalIgnoreCase))

{

// If rainy, check wind conditions

if (windy)

{

return "Don't Play (Rainy + Windy)";

}

else

{

return "Play (Rainy + Not Windy)";

}

}

else

{

return "Invalid outlook input.";

}

}

}

# Output:

