

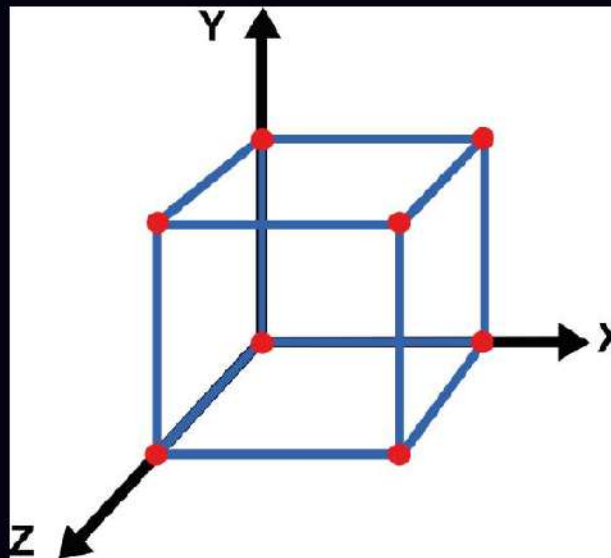
Multi Dimensional Decision Making Python Data Analysis

Bhashkar Paudyal



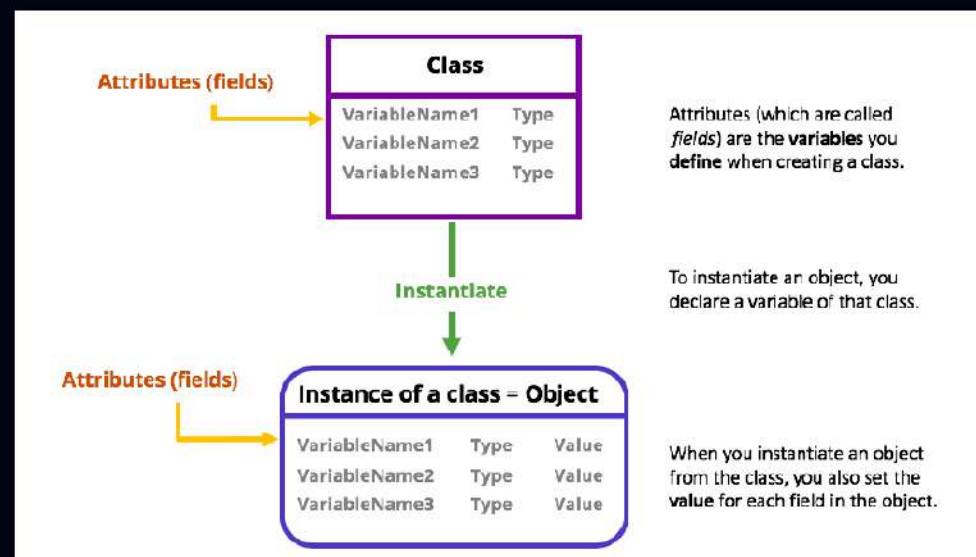
What is Multi Dimensional Decision Making?

- Multi dimensional decision making is a process of making a decision
- Based on multiple dimensions caused by multiple
- factors
- or variables
- or parameters



Purpose of Multi Dimensional Decision Making

- Choose which objects of same Class is best
- Classes have attributes
- These attributes are parameters for mddm
- MDDM chooses which object to use on basis of the values of these attributes



Functioning of MDDM

- uses;
- independent variables as input
- dependent variable as target decision
- takes the product of all values of attributes
- finds the volume of multi-dimensional shape
- finds the percentage of this calculated volume

$$\frac{\prod_{i=1}^{len(parameter)} parameter[i]}{\prod_{i=1}^{len(parameter)} parameter[i].max()} \times 100\%$$

Algorithm of Multi Dimensional Decision Making

1. Start
2. Create a list of parameters
3. Ask for value of parameters
4. Append the value to the list
5. Ask if user wants to add value to another parameter
6. If yes, go to step 3
7. If no, go to step 8
8. Process mddm(multi dimensional decision making) value
9. $mddm = \frac{\text{product_of_parameters}}{(\text{parameter_max_rating})^{\text{len(parameters)}}}$
10. display mddm

Implementation

```
1 def main():
2     print("Welcome to Multi Dimensional Decision Making")
3     parameters = []
4     max = []
5     choice = "y"
6     product = 1
7     result = 1
8     runtime = 0
9     while choice == "y":
10         if choice == "y":
11             print("Enter the parameters")
12             for i in range(int(input("Enter the number of parameters: "))):
13                 parameters.append(int(input("Enter the parameter: ")))
14                 max.append(float(input(f"Enter the max value for parameter{i}: ")))
15                 result *= parameters[i] / max[i]
16             print(result * 100)
17             choice = input("do you want to add values for another object?: ")
18
19
20 if __name__ == "__main__":
21     main()
```