Introduction to Data Mining

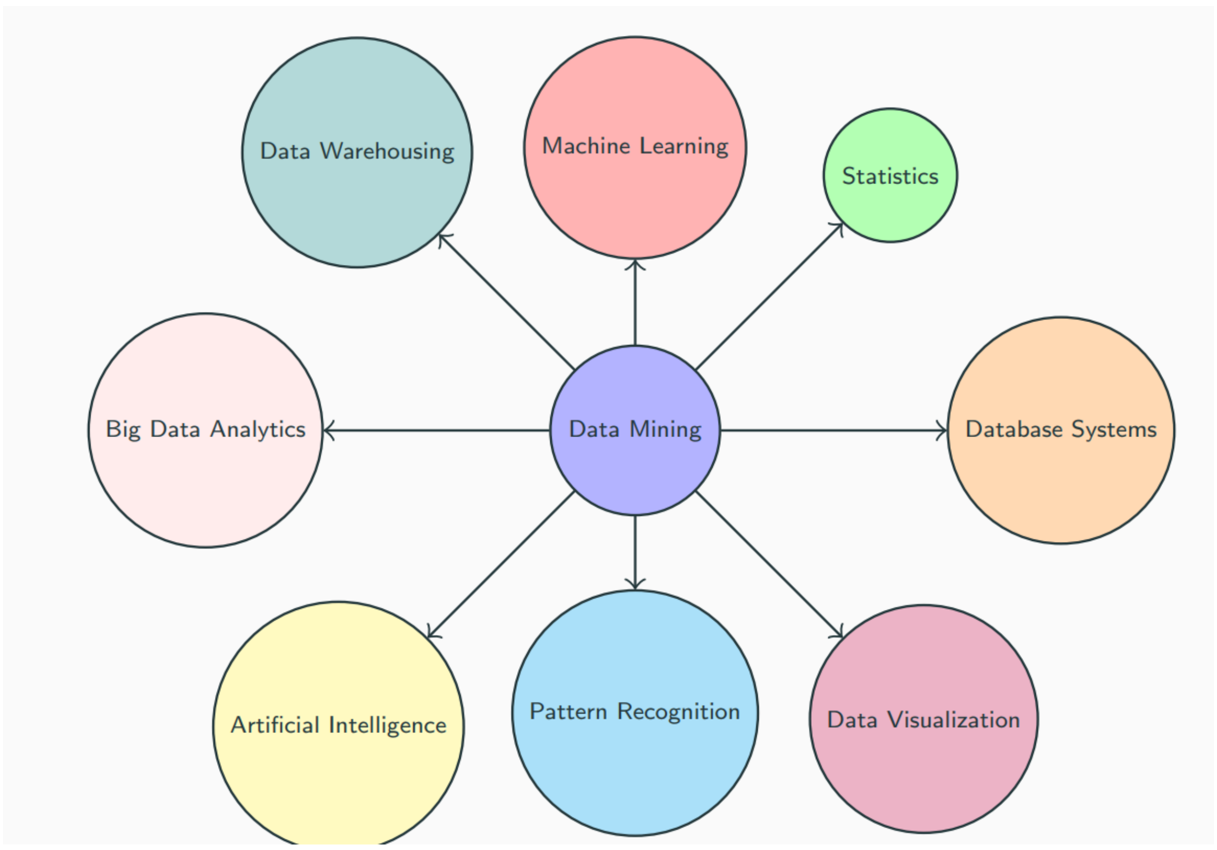
A.Belcaid 24/09/2024

1) What is Data Mining?

Definition: Data mining is the process of discovering patterns, correlations, and useful insights from large datasets. It draws from various fields such as statistics, artificial intelligence (AI), machine learning, and database systems to extract knowledge and turn raw data into valuable information.

Key skills involved in data mining include:

- **Data Warehousing**: Centralized data storage enabling easy access and analysis.
- **Machine Learning**: Algorithms that learn from data to make predictions.
- **Statistics**: Provides techniques for data analysis.
- **Database Systems**: Efficient data storage and querying.
- **Data Visualization**: Represents data insights visually for easier interpretation.
- **Pattern Recognition**: Identifies hidden patterns in data.
- **Big Data Analytics**: Deals with large-scale datasets to uncover insights.



2) What Makes a Data Scientist?

A data scientist combines technical expertise with analytical and communication skills to extract insights from data. Key responsibilities include:

- **Analyzing Data**: Discovering trends and patterns in data.
- **Managing Large Data Sets**: Organizing and cleaning big data for analysis.
- **Creating Visualizations**: Using tools like Matplotlib and Tableau to visually communicate insights.
- **Building Mathematical Models**: Using statistical and machine learning models to make predictions or improve decision-making.
- **Communicating Insights**: Presenting complex technical findings to non-technical audiences.

3) What is Data?

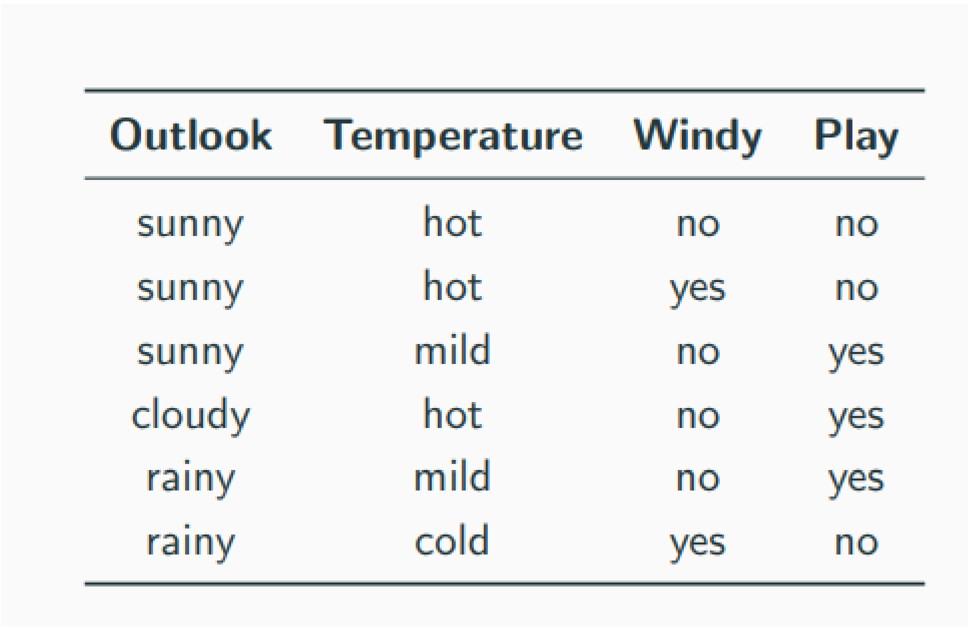
Data refers to raw facts and figures, which can take various forms such as numbers, text, or measurements. It can be categorized in different ways depending on the context or use case, such as structured, semi-structured, and unstructured, or as public, private, and restricted.

It may also be classified as big or small, internal or external, personal or non-personal, and sensitive or non-sensitive.

Data can appear in various formats, including tabular, graph, tree, text, image, video, and audio.

Data Representation

The next example that we will use is a **tabular dataset**, which is one of the most common types of structured data.



Outlook	Temperature	Windy	Play
sunny	hot	no	no
sunny	hot	yes	no
sunny	mild	no	yes
cloudy	hot	no	yes
rainy	mild	no	yes
rainy	cold	yes	no

To analyze data, we often need to transform categorical data into numerical formats. Two common methods are:

Method 1: Direct Numerical Assignment

In this method, we assign each category a unique number. The numbers serve as **labels** for the categories, with **no ranking or hierarchy** implied. This method is simple and works well for datasets with fewer categories.

- **Outlook**:
  - **Sunny**: 1
  - **Cloudy**: 2
  - **Rainy**: 3
- **Temperature**:
  - **Hot**: 1
  - **Mild**: 2
  - **Cold**: 3
- **Windy**:
  - **Yes**: 1
  - **No**: 0
- **Play**:
  - **Yes**: 1
  - **No**: 0

Example Encoding:

For the data point **F1 (1, 1, 0, 0)**:

Method 2: Binary Representation (One-Hot Encoding)

In this method, we use **binary encoding** to represent each category as a combination of 0's and 1's. Each category is assigned a unique binary string, with a separate bit (0 or 1) for each possible category.

Binary Representation of Attributes:

- **Outlook**:
  - **Sunny**: 100
  - **Cloudy**: 010
  - **Rainy**: 001
- **Temperature**:
  - **Hot**: 100
  - **Mild**: 010
  - **Cold**: 001
- **Windy**:

    - **Yes**: 1
    - **No**: 0
- **Play**:
    - **Yes**: 1
    - **No**: 0

Example Encoding:

For the data point **F1 (1,0,0, 1,0,0, 0,0)**:

- **Outlook**: Sunny (100)
- **Temperature**: Hot (100)
- **Windy**: No (0)
- **Play**: No (0)

Note:

The **binary representation (one-hot encoding)** method works well for categorical data with few categories, but it can become **inefficient** when dealing with a large number of categories, as the encoding size increases dramatically.

---

4) Interpreting Data

We use simple rules to predict outcomes. For example, based on a dataset of weather conditions, we might create rules like:

- If it’s windy, don’t play.
- If it’s hot and not windy, don’t play.
- If it’s not windy and not hot, play.

With these rules we have effectively created a model! To evaluate the model's effectiveness, we calculate **precision**. That can be done by training in on the tabular dataset itself.

Precision = number of successful prediction / actual result.

Outlook	Temperature	Windy	Play	Predection
Sunny	Hot	No	No	No
Sunny	Hot	Yes	No	No
Sunny	Mild	No	Yes	Yes
Cloudy	Hot	No	Yes	No
Rainy	Mild	No	Yes	Yes
Rainy	Cold	Yes	No	No

For this case our Precision is 5/6 = 0.83% .

Formally

- We have our data $\mathbf{X}$:
  - (with features: outlook, temp, and windy).
- Our data consists of smaller **instances**, 'some instance' is written as: $\mathbf{x}$.
- If we want to specifically point at a particular instance (say our first row), we write: $\mathbf{x}_1$.
- We can see our model as a function $\mathbf{f}$, that when given any instance $\mathbf{x}$, gives us a prediction $\hat{\mathbf{y}}$.

$$\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}) .$$

- The application of the model to some instance in our data can be written as $\mathbf{f}(\mathbf{x}) = \hat{\mathbf{y}}$. Our hope is that $\hat{\mathbf{y}}$ is the same as our target: $\mathbf{y}$.
- In other words, we want the prediction $\hat{\mathbf{y}}$ to be similar to $\mathbf{y}$

Recapultif

- **Features X**:
  - (outlook, temp., windy)
- **Target**:
  - (play)
- **Some instance**: $\mathbf{x}$
- **Some target**: $\mathbf{y}$
- **First Row** $\mathbf{x}_1$:
  - (sunny, hot, no)
- **First target**:
  - (no)
- **Model**: if it’s not windy and not hot → play ($\mathbf{f}(\mathbf{x})$)
- **Predictions by f**: $\hat{\mathbf{y}}_i$
- **Prediction for** $\mathbf{x}_1$: $\hat{\mathbf{y}}_1$ (no)