Anomaly Detection
Recommender Systems

SUSANA BRÁS

SUSANA.BRAS@UA.PT

Anomaly Detection – what?

Anomaly Detection – The process of identifying unusual patterns or deviations in data that seem suspicious, because they don't conform to established norms. It is based on the detection of rare events, outliers, or inconsistencies that might indicate errors, fraud, or security breaches.

Identifying deviations - Anomaly detection focuses on finding data points or patterns that significantly differ from the majority of the data or the expected behavior.

Applications - It's used in various fields, including cybersecurity (detecting intrusions), finance (fraud detection), manufacturing (quality control), and healthcare (monitoring patient conditions).

Benefits - Anomaly detection can help identify problems early, reduce damages, and improve the accuracy and quality of data.

Anomaly Detection – core principles

Anomaly Detection	Operates under the premise that anomalies are rare and significantly different from the majority of the data
	The process involves training a model on data that is labeled as normal or assumes that the majority of the data represents normal behavior
	The model then attempts to identify data points that deviate from this established norm
	The effectiveness of anomaly detection relies on the ability to accurately define what constitutes normal behavior, which can vary widely across different domains and applications

Anomaly Detection – why?

Improved data quality: Identifying and handling data anomalies can significantly improve data quality, which is essential for accurate and reliable data analysis. By addressing data anomalies, analysts can reduce noise and errors in the dataset, ensuring that the data is more representative of the true underlying patterns.

Enhanced decision making: Data-driven decision making relies on accurate and reliable data analysis to inform decisions. By identifying and handling data anomalies, analysts can ensure that their findings are more trustworthy, leading to better-informed decisions and improved outcomes.

Optimized machine learning performance: Data anomalies can significantly impact the performance of machine learning algorithms, as they can cause the model to fit the noise rather than the underlying pattern in the data. By identifying and handling data anomalies, analysts can optimize the performance of their machine learning models, ensuring that they provide accurate and reliable predictions.

Anomaly Detection – steps

Anomaly detection involves three main steps:

1. **Data preprocessing:** involves cleaning, transforming, and standardizing the data to make it suitable for anomaly detection.

2. **Anomaly identification:** involves applying one or more techniques to identify the anomalies in the data. These techniques can be based on statistical methods, machine learning algorithms, or domain knowledge.

3. **Anomaly analysis:** involves interpreting and explaining the anomalies, as well as taking appropriate actions to resolve them.

Anomaly detection is not a one-size-fits-all problem. Different types of data and domains may require different approaches and techniques. Therefore, it is important to understand the nature and context of the data, as well as the objectives and requirements of the anomaly detection task.

Anomaly Detection – categories

Point Anomaly

- These are individual data points that deviate significantly from the rest of the data.

Contextual Anomalies

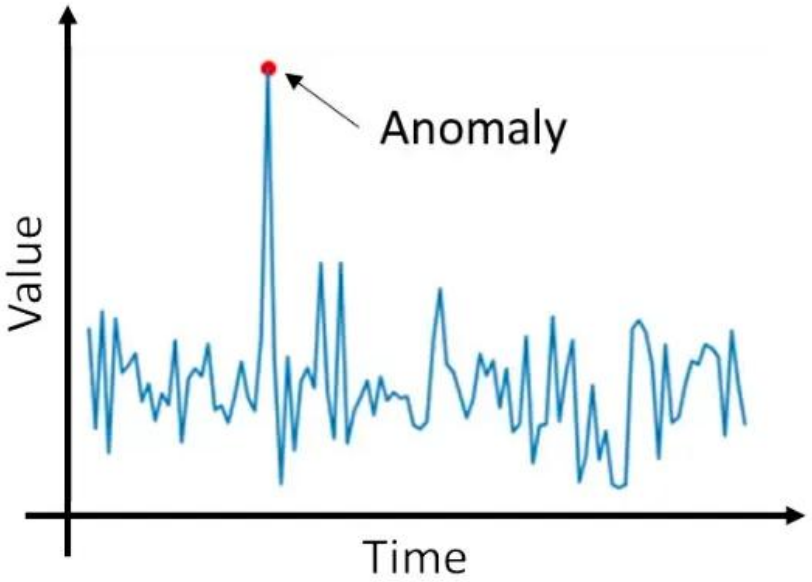
- These are data points that deviate significantly from the normal behavior of the data in a specific context. The context can be defined by temporal, spatial, or other attributes.

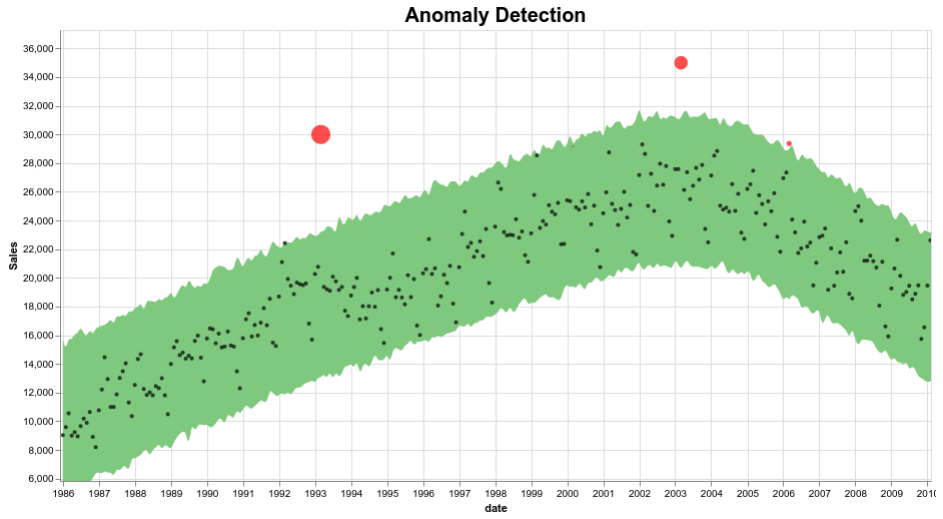
Collective Anomalies

- These are groups of data points that deviate significantly from the rest of the data as a whole. The individual data points may not be anomalous by themselves, but their collective behavior is anomalous.

Anomaly Detection – how?

Visualization – A powerful tool for detecting data anomalies, as it allows data scientists to quickly identify potential outliers and patterns in the data.



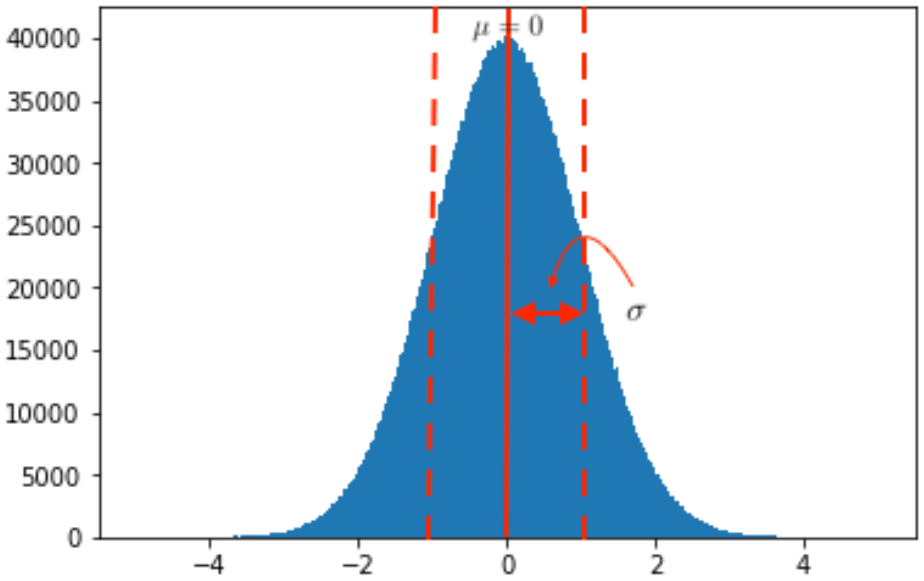


Anomaly Detection – how?

Statistical tests - by comparing the observed data with the expected distribution or pattern.

If $x \in \mathbb{R}$, and x follows Gaussian distribution with mean, μ and variance σ^2, denoted as,

$$x \sim \mathcal{N}(\mu, \sigma^2)$$



Standard normal Gaussian distribution (μ=0, standard deviation σ=1). Density is higher around μ and reduces as distance from mean increases. If we know parameters μ and σ, the probability of x in Gaussian distribution is:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

Anomaly Detection – how?

Machine learning algorithms - detect data anomalies by learning the underlying pattern in the data and then identifying any deviations from that pattern. Some of the most common ML anomaly detection algorithms include: isolation forest, One-Class SVM, k-NN, Naive Bayesian, Autoencoders, Local Outlier Factor (LOF), k-means

Supervised methods use the labeled data to train a classifier that can distinguish between normal and anomalous data points. (Anomaly Classification, Anomaly Prediction)

Unsupervised methods use the unlabeled data to learn the normal behavior of the data and identify the data points that deviate from it. (Anomaly Detection, Anomaly Scoring)


You
Uber
Eats

Recommender System

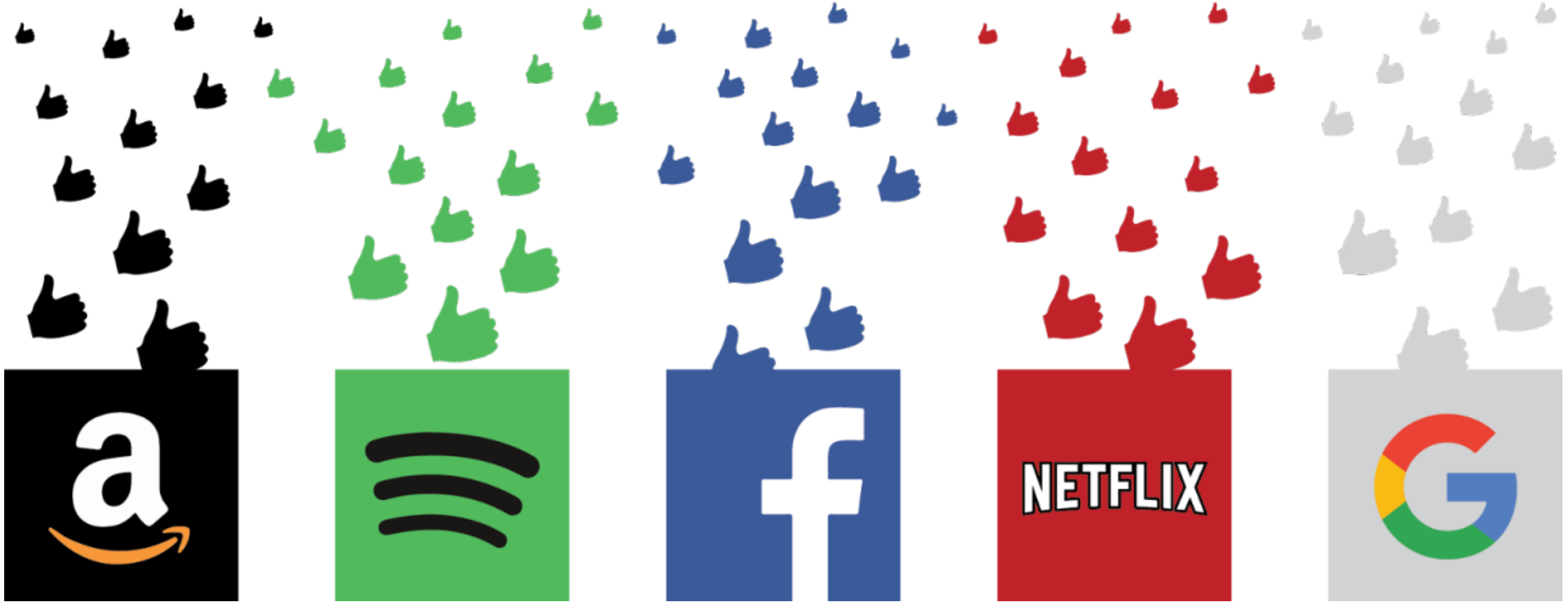
What?
Recommendation systems are AI-driven tools that use algorithms to suggest items to users based on their preferences and behavior.

Where?
Use cases are related to e-commerce, streaming services, and social media…

How?
These systems analyze individual user data to predict what other users will likely find interesting.

Why?
Enhancing user experience, boosting engagement, and personalization.
Increasing business revenue.

Surprise is a Python scikit for recommender systems:
https://surprise.readthedocs.io/en/stable/


NETFLIX

Recommender System - Types

- Content-Based Filtering: Recommending items based on user history and item characteristics.

- Collaborative Filtering: Predicting user preferences based on similar users' choices.
  - User-based
  - Item-based

- Hybrid Models: Combining multiple techniques.

Content-Based Filtering

CBF methods are constructed behind the following paradigm: "**Show me more of the same that I've liked**".

This approach recommends items similar to those the user liked before. The recommendations are based on item descriptions and a profile of the user's preferences.

The computation of the **similarity** between items is the most important part of these methods, and it is based on the content of the items themselves.

As the content of the item can be very diverse, and it usually depends on the kind of items the system recommends, a range of algorithms are usually used to abstract features from items.

Uses item metadata and user profiles
Learns a user's preferences and suggests similar items
Example: Movie genres, actors, keywords

Collaborative Filtering

CF methods are constructed behind the following paradigm: "**Tell me what's popular among my like-minded users**".

An important working hypothesis behind these kind of recommenders is that similar users tend to like similar items.

These approaches are based on collecting and analyzing large number of data related to the behavior, activities, and predicting what users will like based on their similarity to other users.

One of the main advantages is that it does not need to "understand" what the item it recommends is.

The main drawbacks of this kind of method is the need for a user community, as well as the cold-start effect for new users in the community.

The cold-start problem appears when the system cannot draw any, or an optimal, inference or recommendation for the users (or items) since it has not yet obtained sufficient information.

Collaborative Filtering

CF can be of two types:

• **User-based CF:** Find similar users to me and recommend what they liked. In this method, given a user, *U*, we first find a set of other users, *D*, whose ratings are similar to the ratings of *U,* and then we calculate a prediction for *U*.

• **Item-based CF:** Find similar items to those that I previously liked. In item-based CF, we first build an item–item matrix that determines relationships between pairs of items; then, using this matrix and data on the current user *U*, we infer the user's taste. Typically, this approach is used in the domain: people who buy *x* also buy *y*. This is a really popular approach used by companies like Amazon. Moreover, one of the advantages of this approach is that items usually do not change much, so their similarities can be computed offline.

Based on user-item interactions
Key idea: similar users like similar items
Techniques:
- User-based kNN
- Item-based kNN

Hybrid Recommenders

Hybrid approaches can be:

- making content-based and collaborative predictions separately and then combining them
- adding content-based capabilities to a collaborative approach (and vice versa)
- unifying the approaches into one model.

Similarity – what is it?

Similarity measures quantify the similarity between objects, data points, or vectors by mathematical definition.

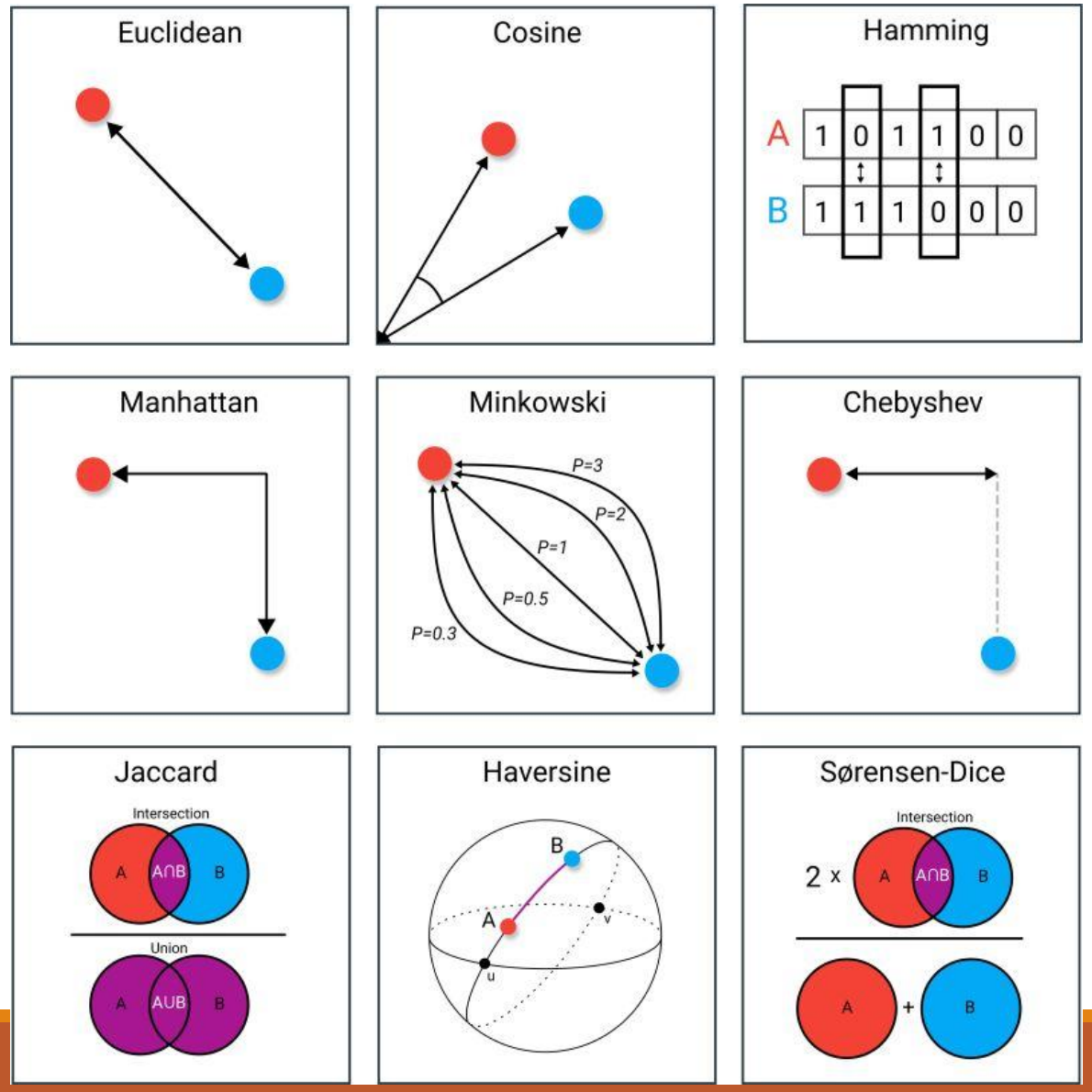
The state or fact of being similar measures how much two objects are alike.

In a data mining context can be described as a distance with dimensions representing feature space. If the distance is small, two objects are very similar, in high distance there is a low degree of similarity.

Understanding the concept of similarity in the vector space and employing appropriate measures is fundamental in solving a wide range of real-world problems, such as recommendations.

Several similarity measures can be used to calculate how close two vectors are in the embedding space.

Similarity – distance examples



Similarity – other approaches

- Correlation
- Entropy
- Mutual Information
- Normalized Relative Compression Measure
- Etc.

Similarity – which one?

Consider:

- **Type of data:** Some metrics are more appropriate for continuous data, while others are better suited for categorical or binary data.
- **Characteristics of the data:** Different metrics are sensitive to different aspects of the data, such as the magnitudes of differences between attributes or the angles between attributes. Consider which characteristics of the data are most important to your analysis and choose a similarity metric that is sensitive to these characteristics.
- **Goals of your analysis:** Different metrics can highlight different patterns or relationships in the data, so consider what you are trying to learn from your analysis and choose a distance metric that is well-suited to this purpose.

Algorithms – Matrix Factorization

Matrix factorization decomposes the user-item interaction matrix into smaller matrices.

These matrices represent latent factors that explain user preferences and item attributes.

By multiplying these matrices, we predict missing interactions and recommend items to users.

This method is efficient for handling sparse data and providing accurate recommendations.

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance forever	5	?	?	0
Cute puppies of love	?	4	0	?
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	?

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

Predicted ratings:

$$\begin{bmatrix} (\theta^{(1)})^T(x^{(1)}) & (\theta^{(2)})^T(x^{(1)}) & \dots & (\theta^{(n_u)})^T(x^{(1)}) \\ (\theta^{(1)})^T(x^{(2)}) & (\theta^{(2)})^T(x^{(2)}) & \dots & (\theta^{(n_u)})^T(x^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ (\theta^{(1)})^T(x^{(n_m)}) & (\theta^{(2)})^T(x^{(n_m)}) & \dots & (\theta^{(n_u)})^T(x^{(n_m)}) \end{bmatrix}$$

Algorithms – KNN

K-Nearest Neighbors (k-NN) is a recommendation technique that suggests items to a user based on the preferences of similar users.

K-NN works by:

Finding Similar Users: It calculates the similarity between users based on their interactions with items. Users who have similar preferences are considered neighbors.

Selecting “k” Neighbors: The technique selects the top “k” similar users as neighbors. These users’ liked items are used to make recommendations.

Aggregating Preferences: It aggregates the preferences of the neighbors, often through weighted averages or other similarity-based methods.

Making Recommendations: Items liked by the neighbors, but not yet seen by the target user, are recommended.

K-NN is simple to understand and implement, making it a popular choice for collaborative filtering. However, it may face challenges with sparse data and the “cold start” problem for new users or items.

Algorithms – DL

Main characteristics of DL:

- Deep learning has transformed recommendation systems
- Neural networks can capture complex patterns in behavior and content
- Great for unstructured data (text, images, etc.)
- Highly personalized, data-driven suggestions

DL as a task specific models?

- Despite progress, deep models remain task-specific
- Limited generalization beyond trained scenarios
- Emergence of Foundation Models, e.g., LLMs
- New AI paradigm: general-purpose, adaptable systems

Algorithms – LLMs

How LLMs Enhance Recommendation Systems:

- Act as zero-shot rankers using self-attention and inference
- Capture user preferences from histories and interactions
- Leverage rich text representations and external knowledge
- Improve both quality and diversity of recommendations

Adaptability and Explainability with LLMs:

- Natural language explanations via conversational interfaces
- Incorporate user feedback and context dynamically
- Handle cold start, content-based, and hybrid scenarios
- A new level of flexibility in recommender systems

Recommendation Systems - Challenges

Cold Start: Difficulty recommending for new users or items

Sparsity: Most users rate only a few items

Scalability: Managing large-scale data (millions of users/items)

Biases: Overexposure of popular items

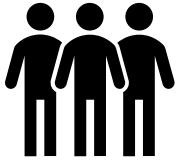
Ethical Issues: Filter bubbles, lack of fairness and diversity

Importance of domain and data understanding


AI

The process

Data Collected



Sample Population

Does not represent whole population

Actual Population

- Inaccuracy
- Inequality
- Incomplete
- Incompatible

Ethics

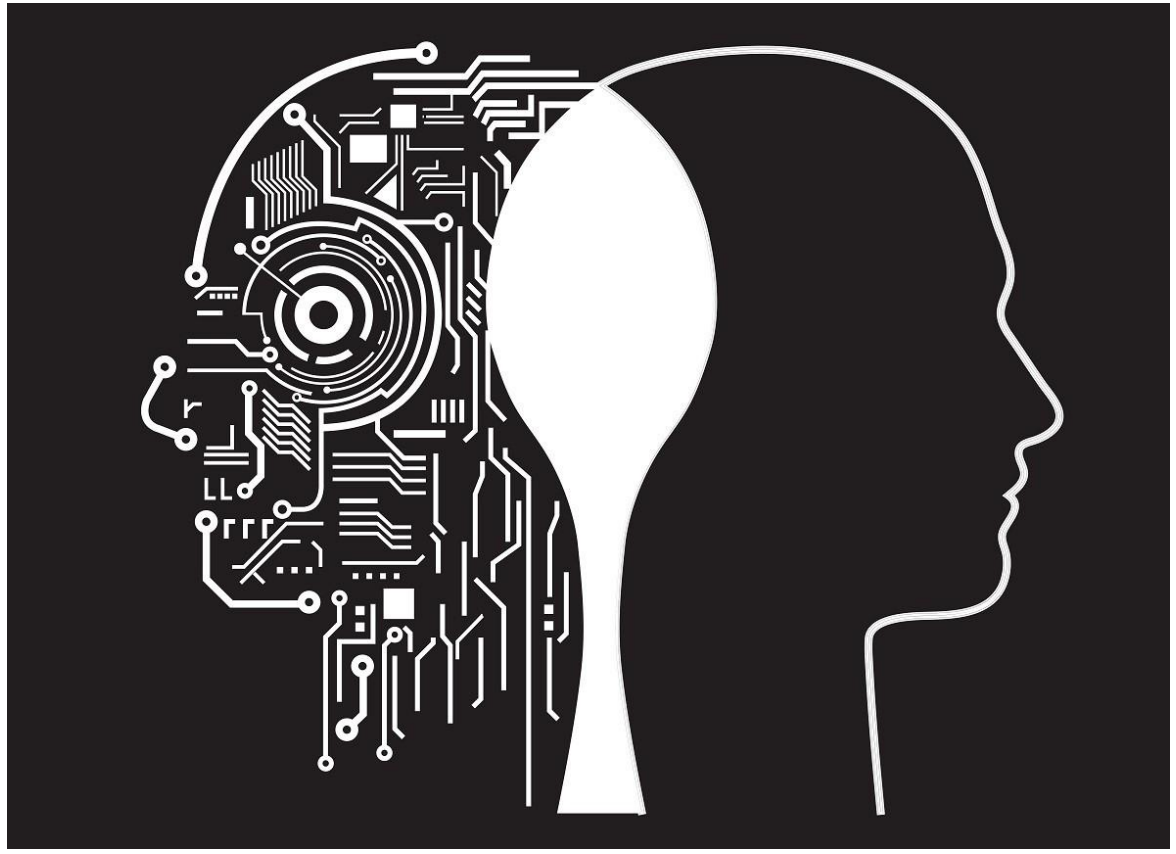
No matter what will the future and the applications of AI, there are something that always should be guarantee: **Ethics**.

Ethics

No matter what will the future and the applications of AI, there are something that always should be guarantee: **Ethics**.

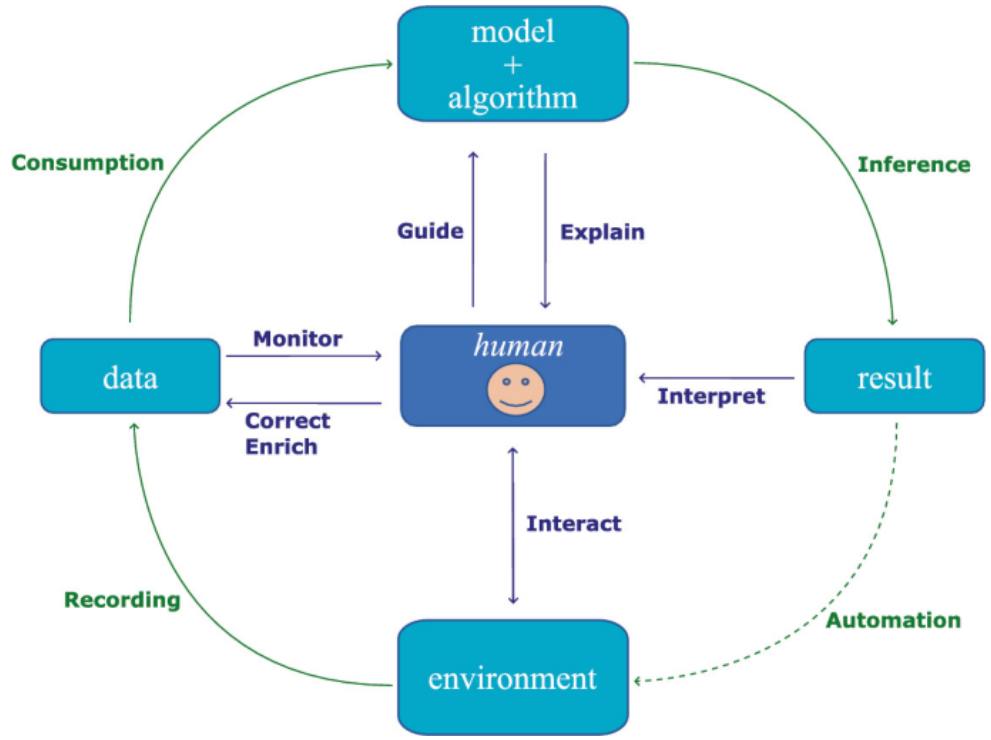
Ethical and Regulatory Concerns: As AI and data usage become more prevalent, ethical considerations and regulations around data **privacy**, **bias**, **transparency**, and accountability are likely to grow in importance. Striking a balance between innovation and responsible use will be crucial.

Acceptance

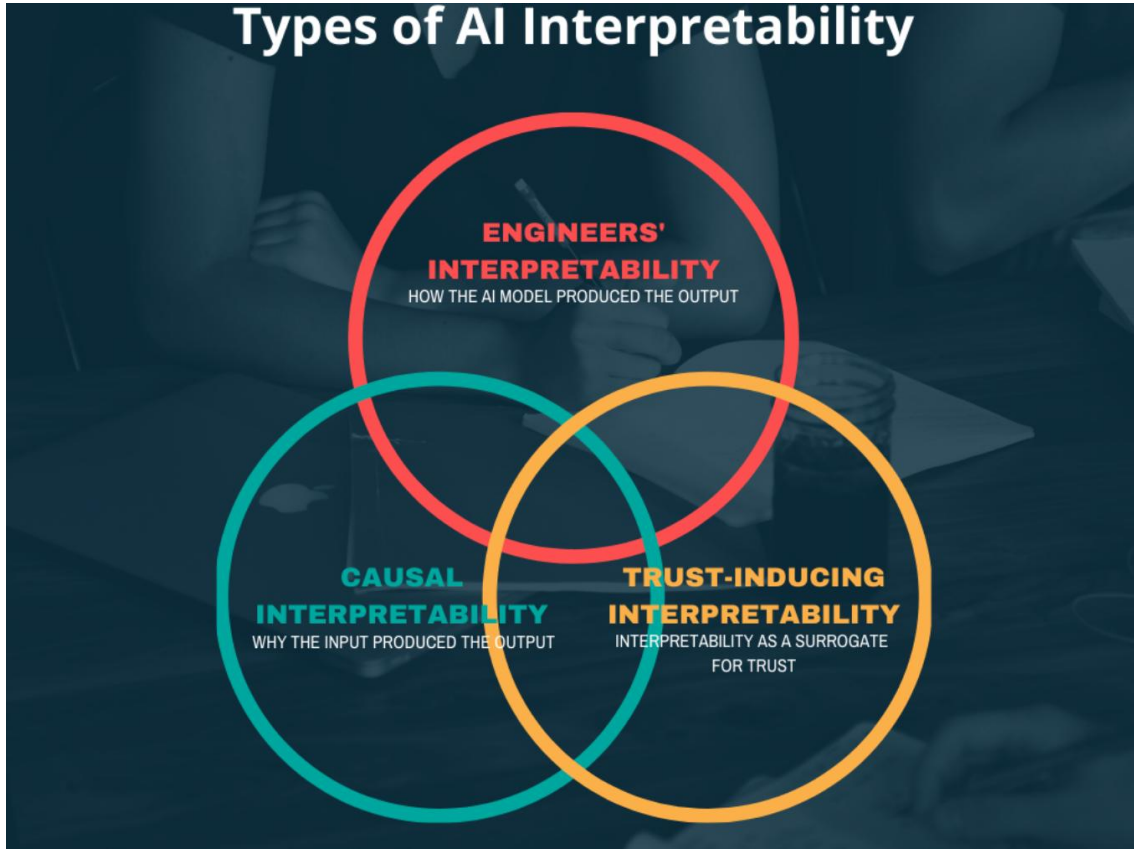


Acceptance

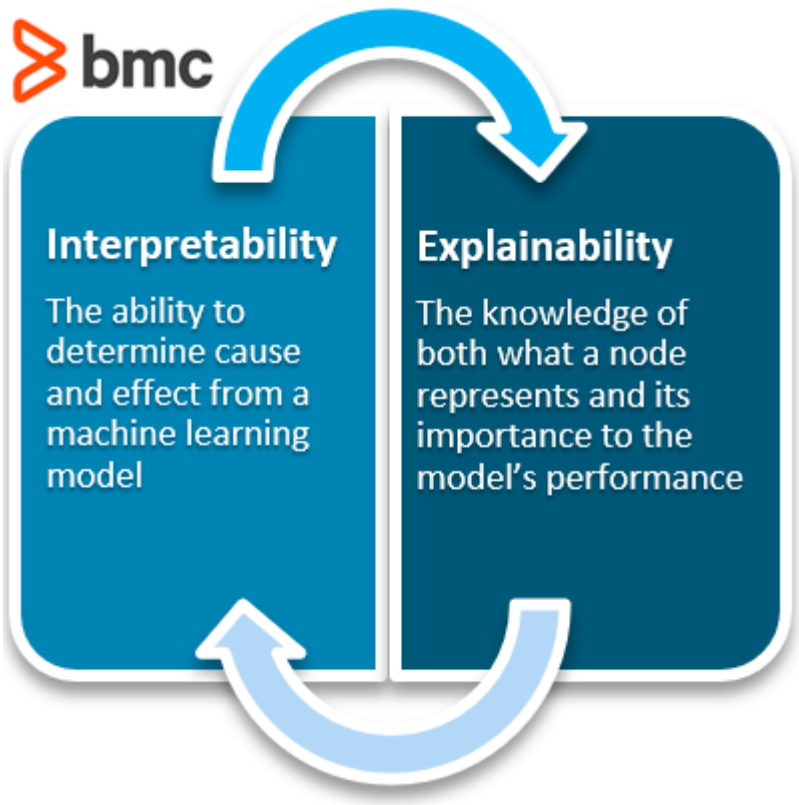
Acceptance of AI is dependent on the concept understanding.



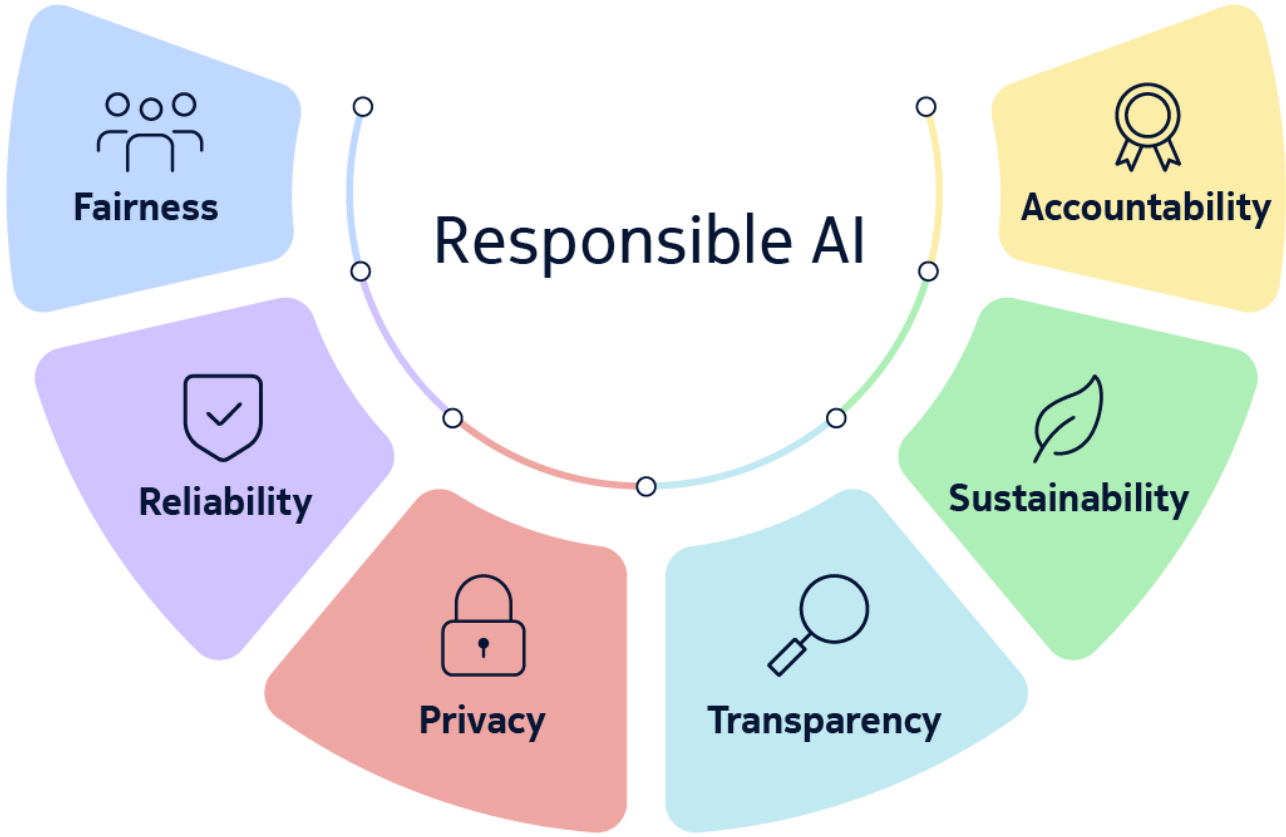
Interpretability and Explainability



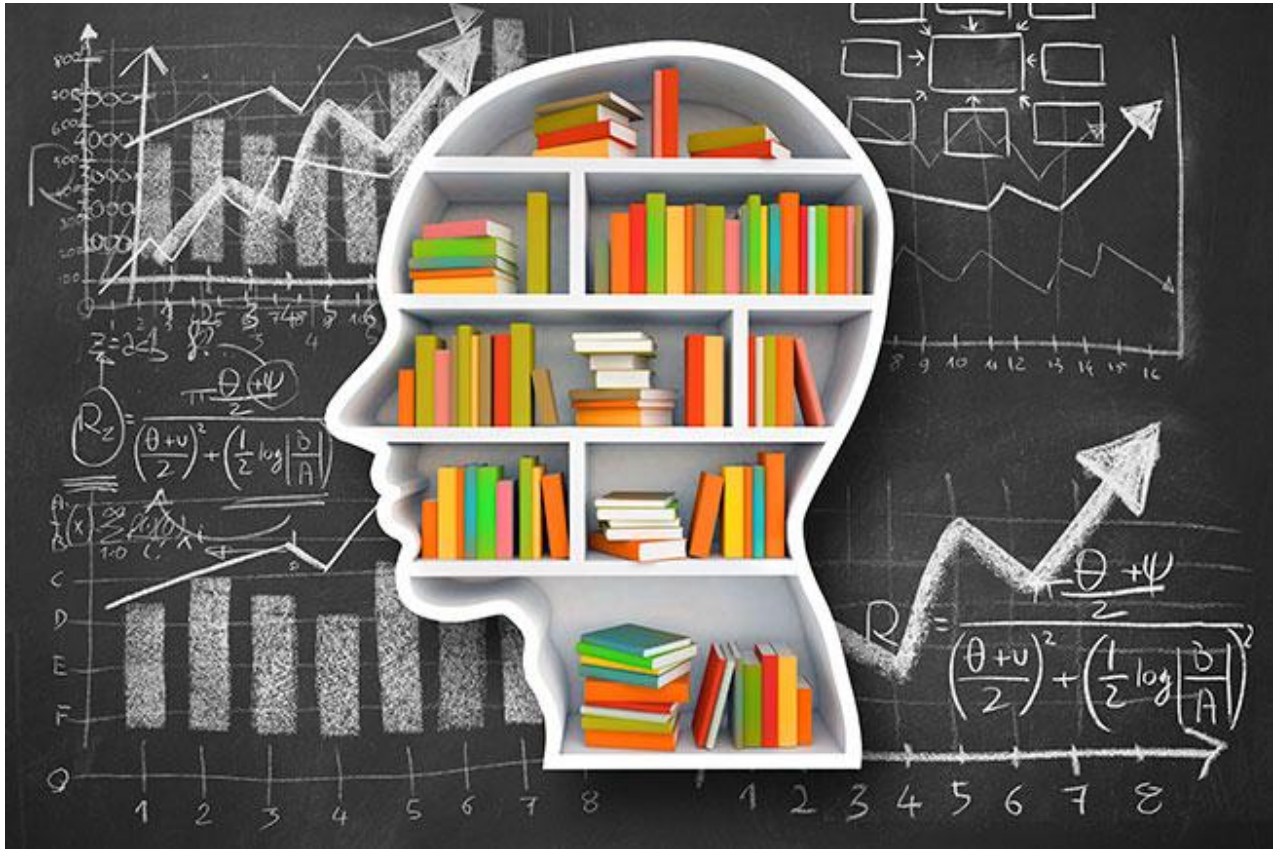
Interpretability and Explainability



Responsible AI


Responsible AI
Fairness
Accountability
Reliability
Sustainability
Privacy
Transparency

Important characteristic of an AI user



Important characteristic of an AI user

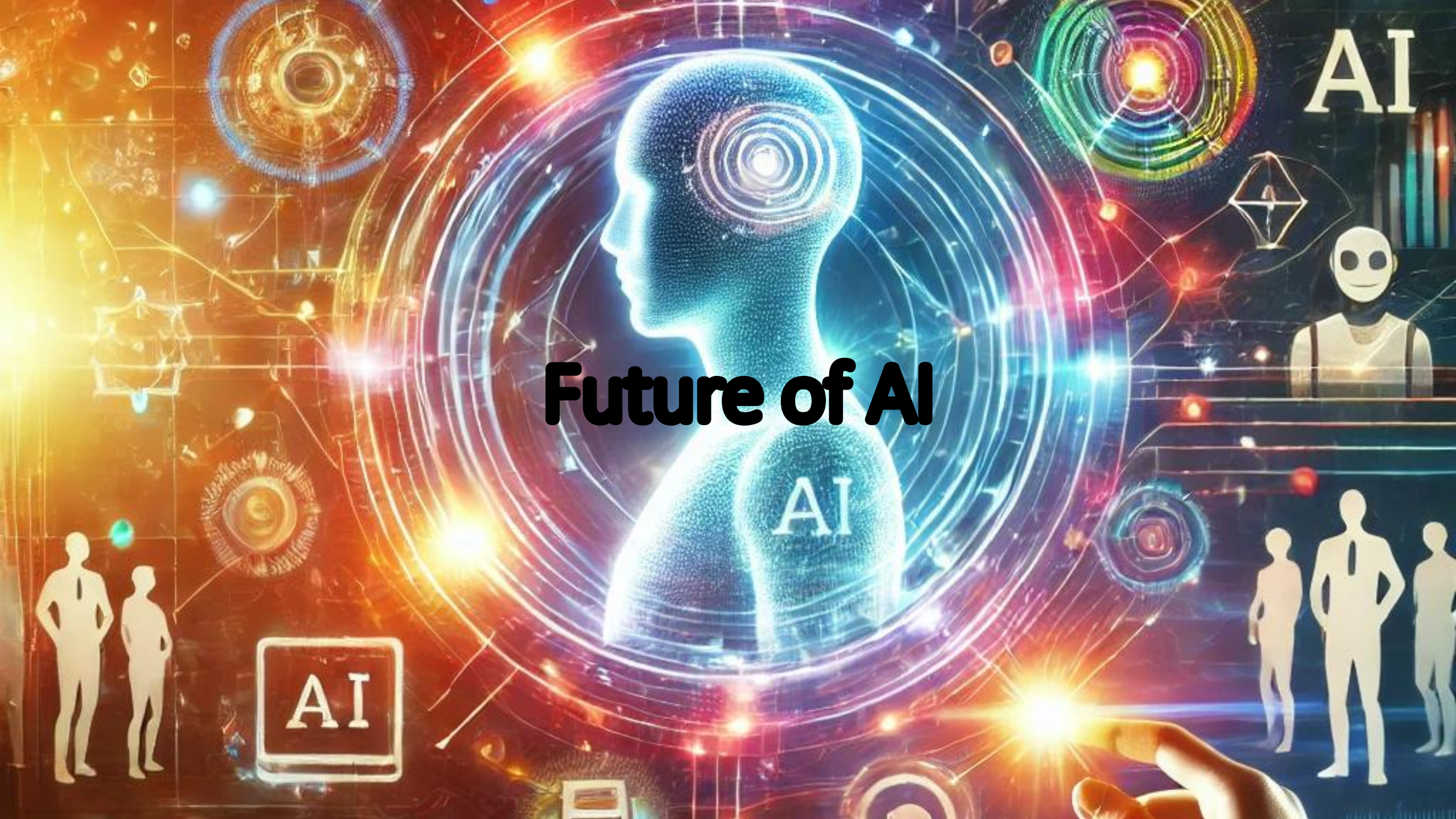
An important characteristic of an AI user is **clarity in communication**. Effectively interacting with AI often requires users to express their needs, questions, or problems clearly and concisely. Providing relevant details or context helps the AI understand the request and deliver accurate, tailored responses.

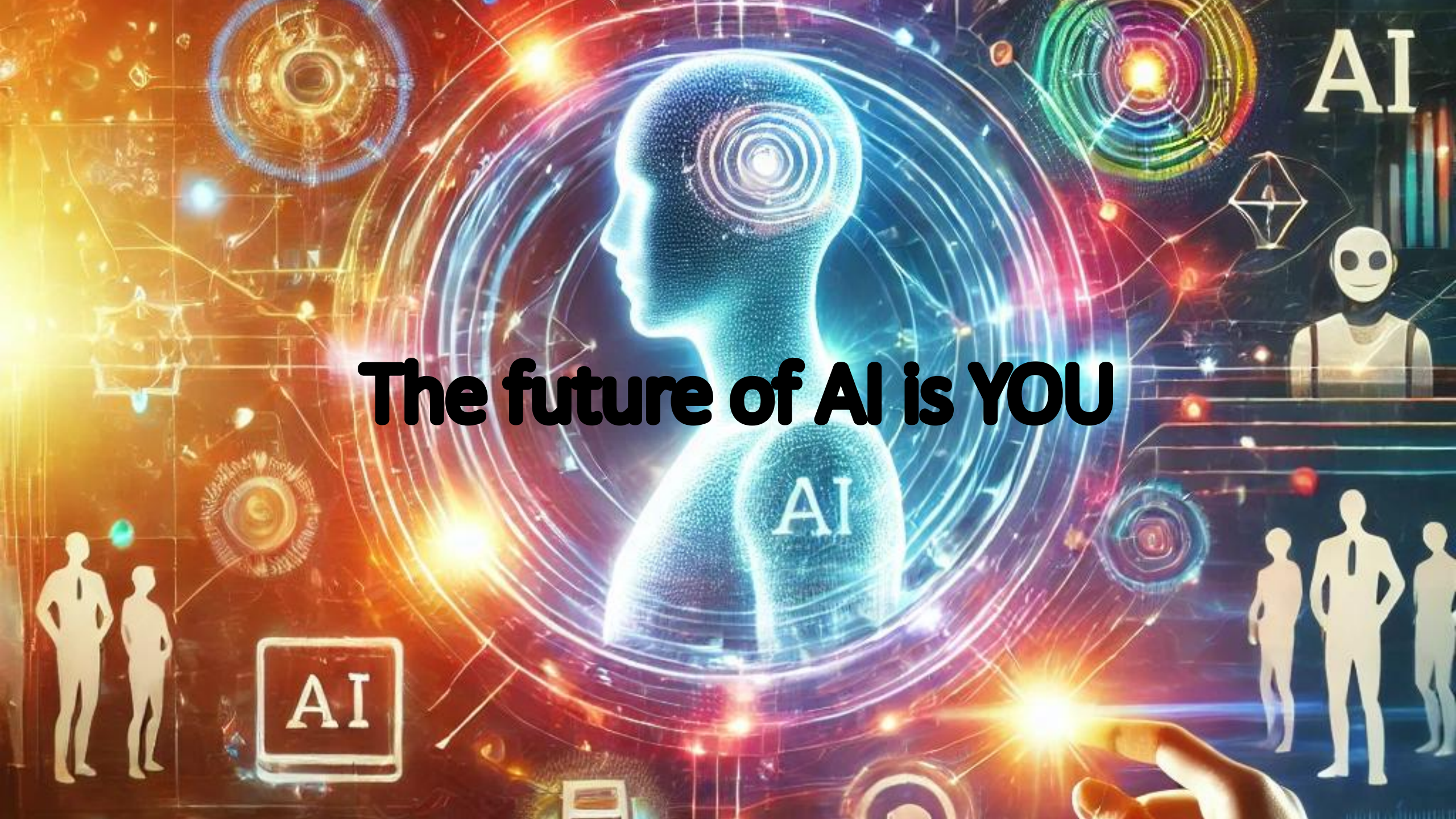
Is the AI system always right??

It's important to approach AI outputs critically, validate information when necessary, and use AI as a tool to assist rather than replace human decision-making.

No, an AI system is not always right. While AI can process large amounts of data and provide useful insights or solutions, it has limitations, such as:

1. **Dependence on Training Data**: AI systems rely on the data they are trained on. If the data contains biases, errors, or is incomplete, the AI's output may reflect those flaws.
2. **Context Limitations**: AI may misunderstand ambiguous or nuanced requests, especially if sufficient context is not provided.
3. **Dynamic Knowledge**: AI knowledge can become outdated if it doesn't have access to real-time updates or recent information.
4. **Complex or Unpredictable Scenarios**: AI may struggle with unique or highly complex situations it wasn't designed or trained to handle.
5. **Human Oversight**: AI systems lack human judgment, creativity, and emotional intelligence, which are often needed for nuanced decision-making.


Future of AI
AI
AI
AI
AI


AI
The future of AI is YOU
AI
AI