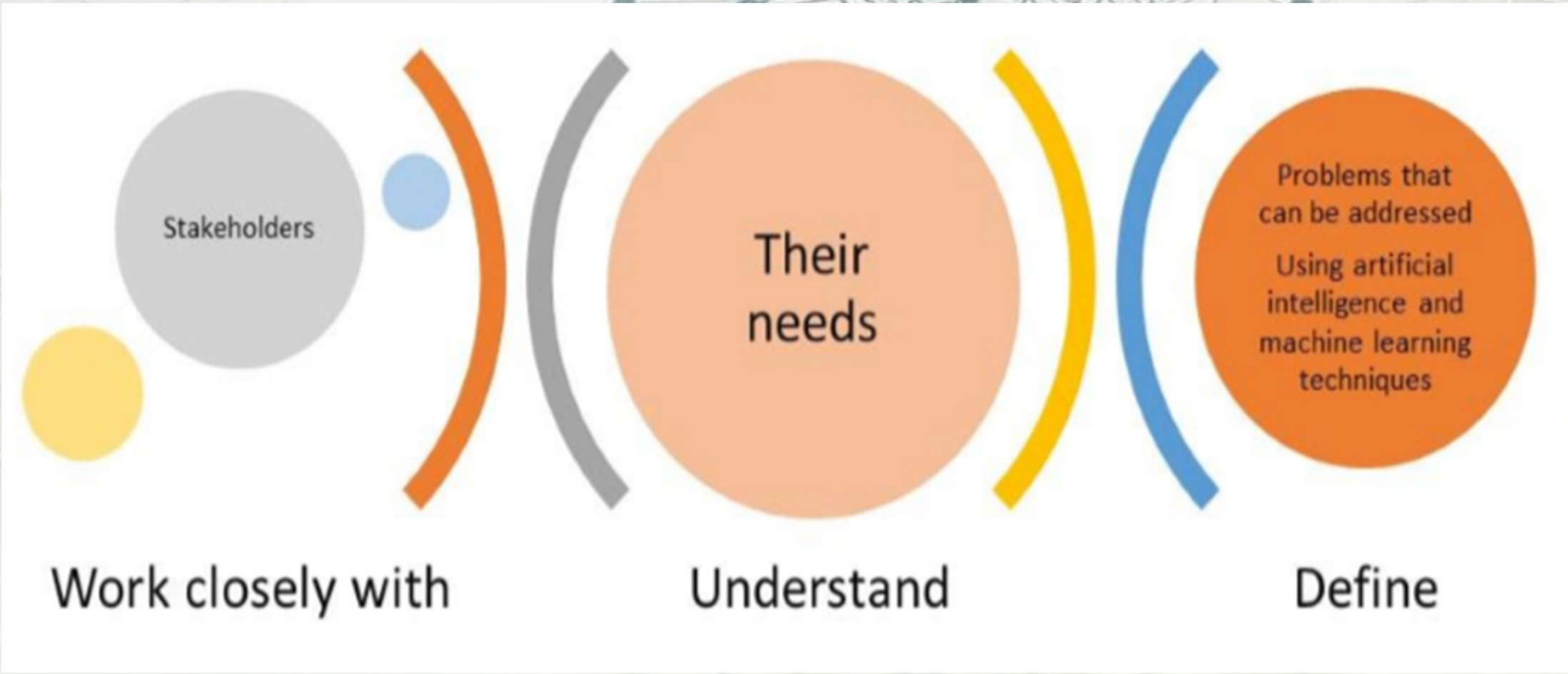
About Data

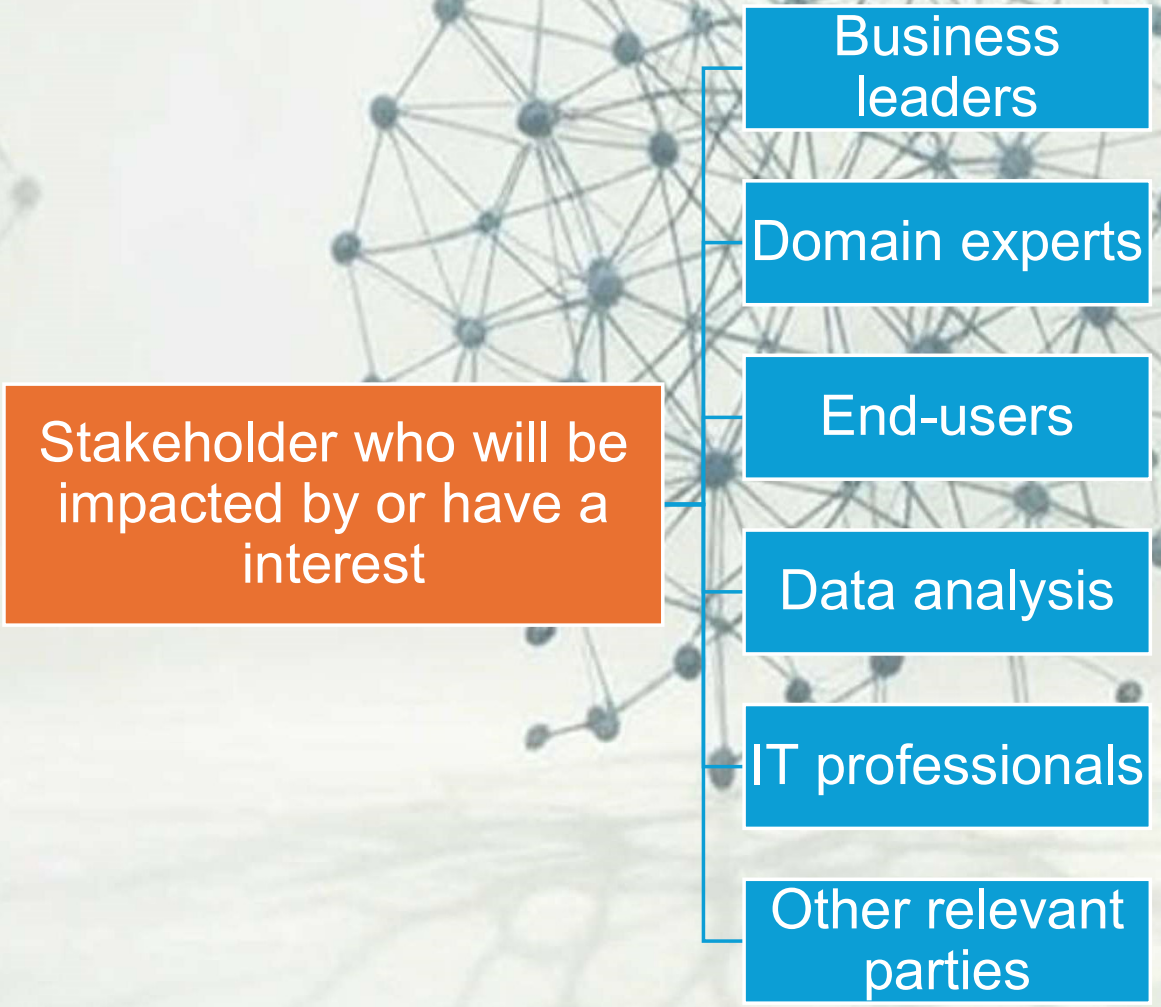
Created By

The easylearn academy

Problem Definition



Identify Stakeholders

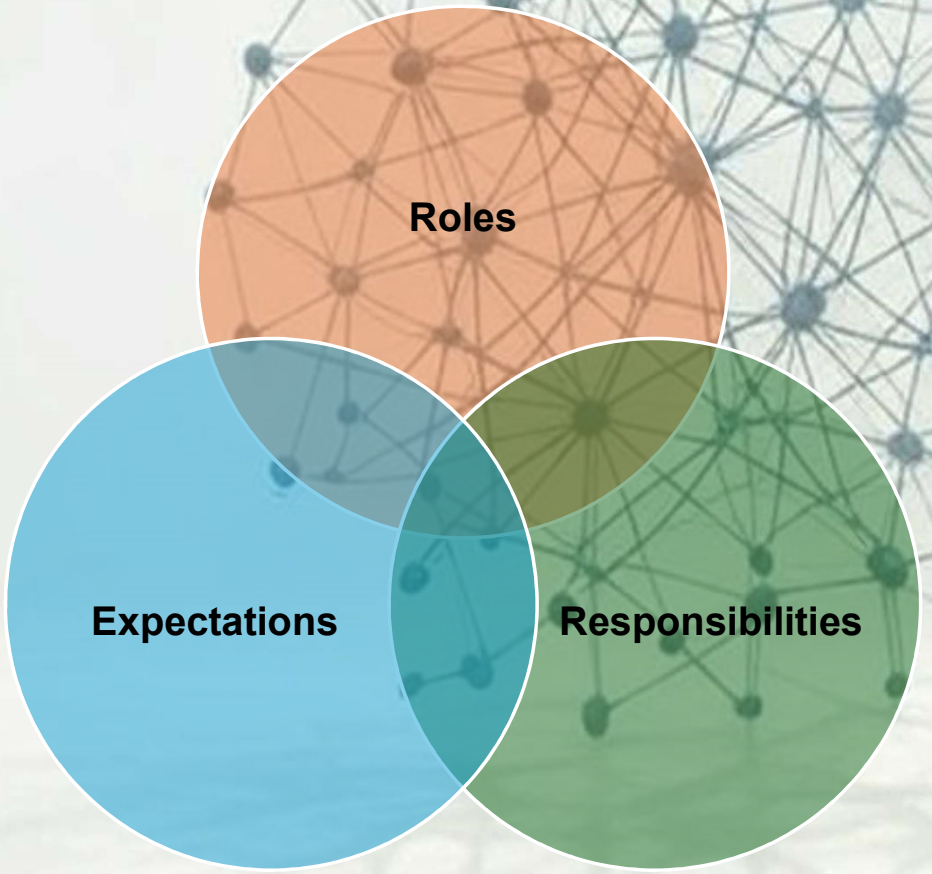


Engage Stakeholders & Listening

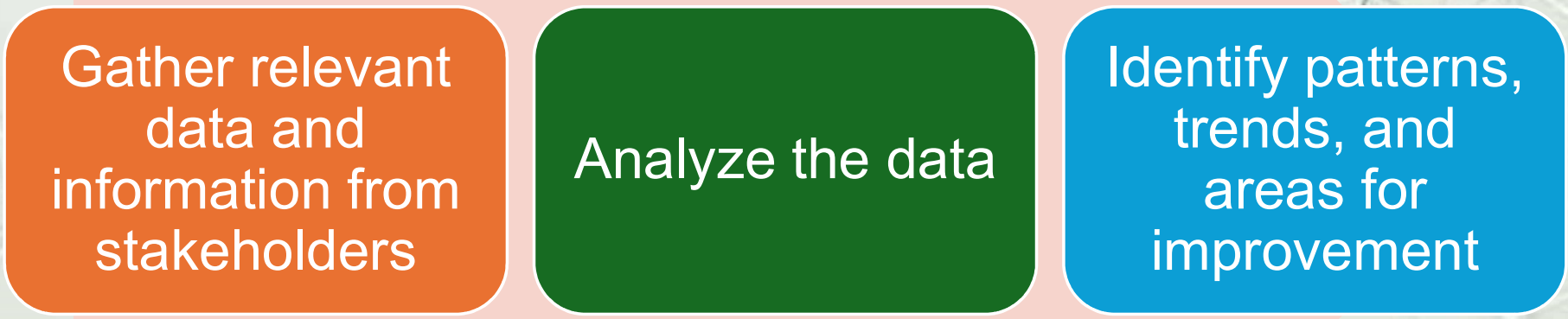
Ask Questions

- What are your key requirements?
- What are the main challenges do you face?
- What are your current pain points?
- Can you please given examples to illustrate these?

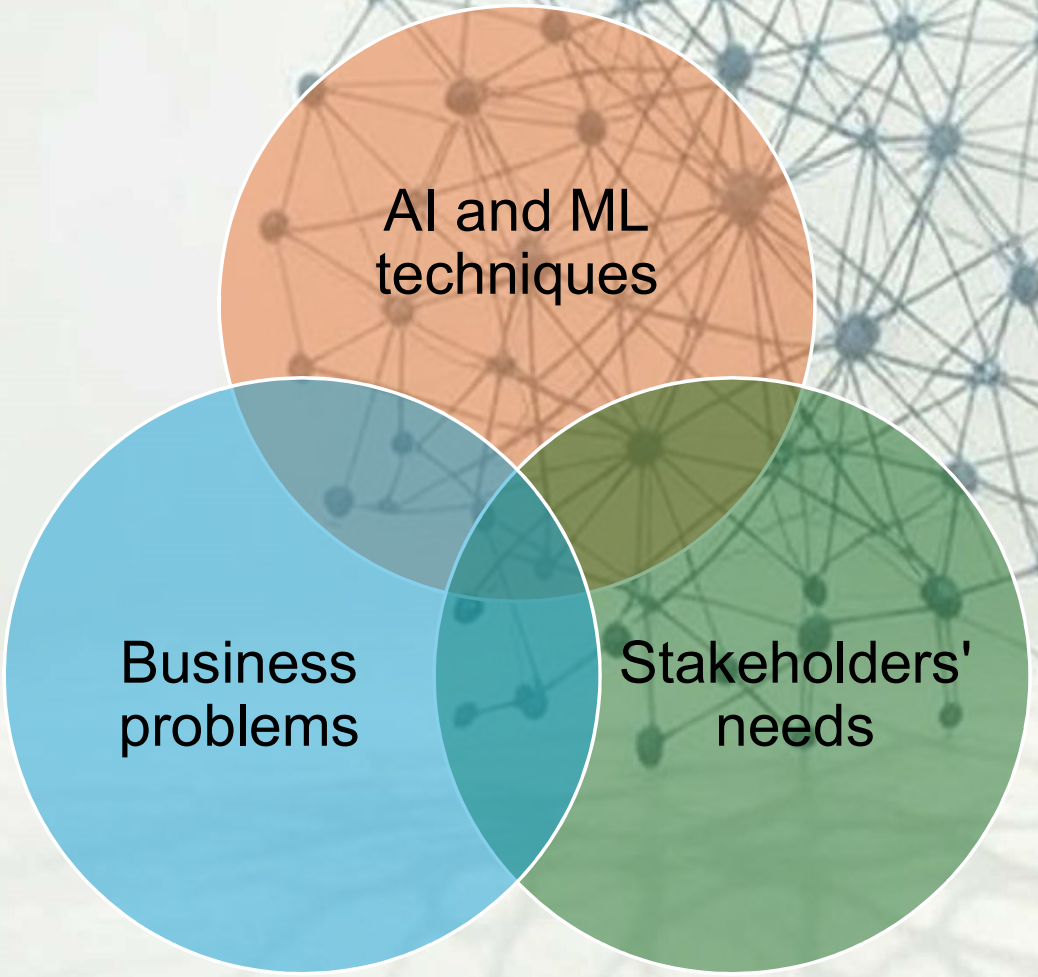
Understand Requirements

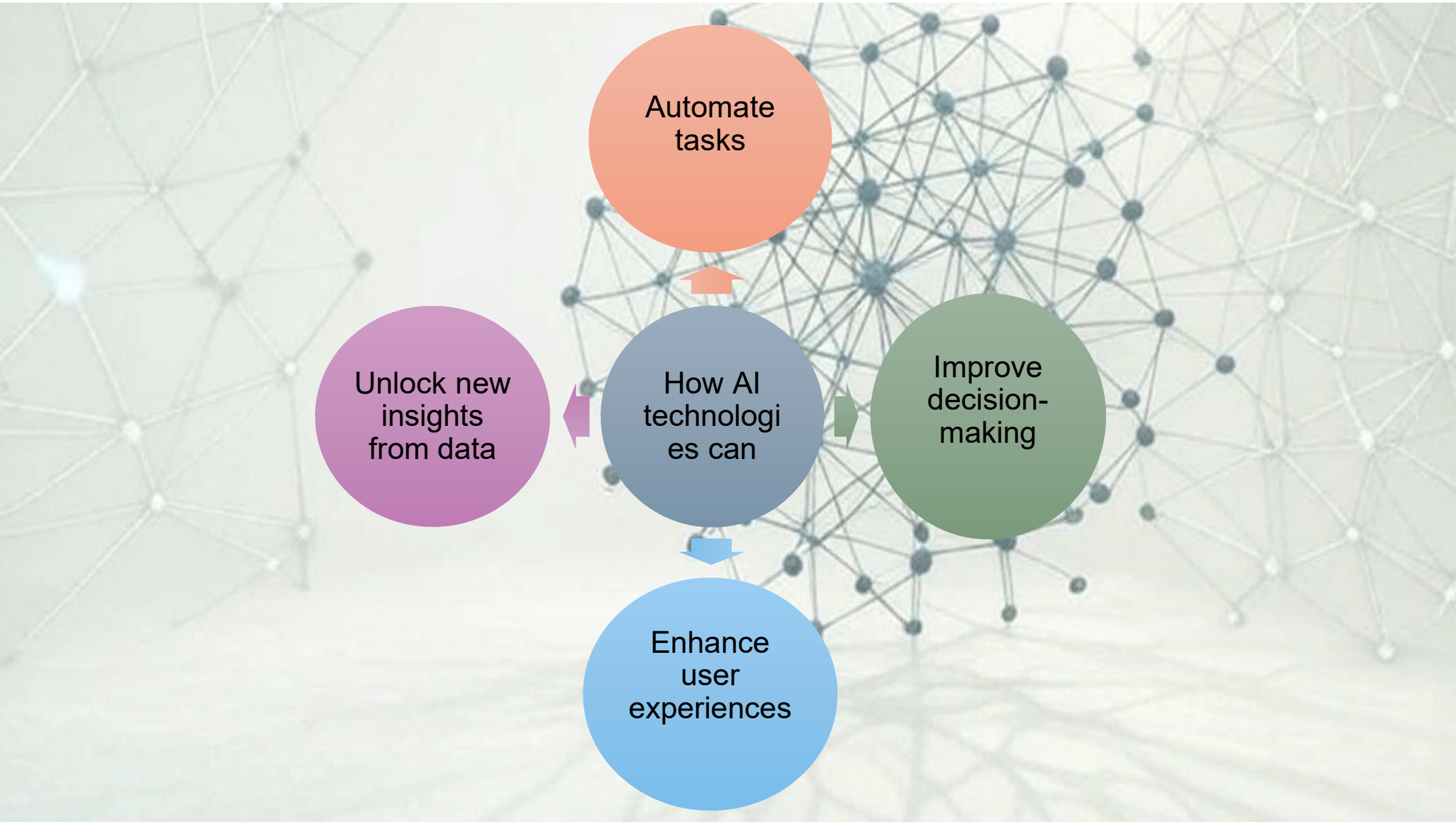


Collect Data

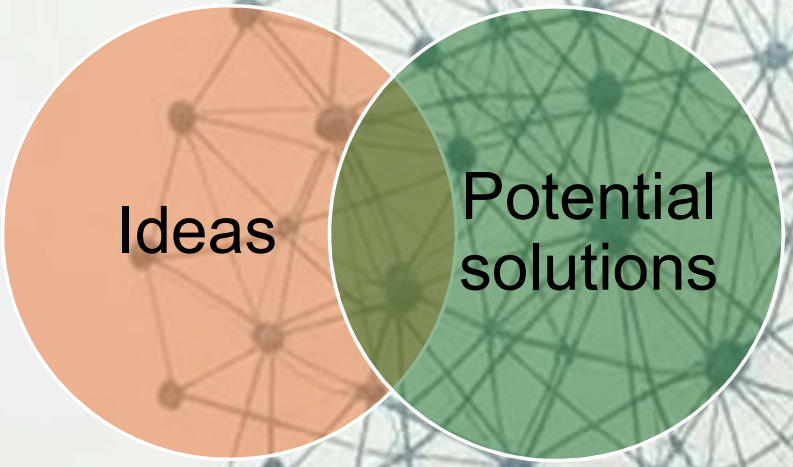


Map Needs to AI Opportunities




Automate tasks
Unlock new insights from data
How AI technologies can
Improve decision-making
Enhance user experiences

About Solutions

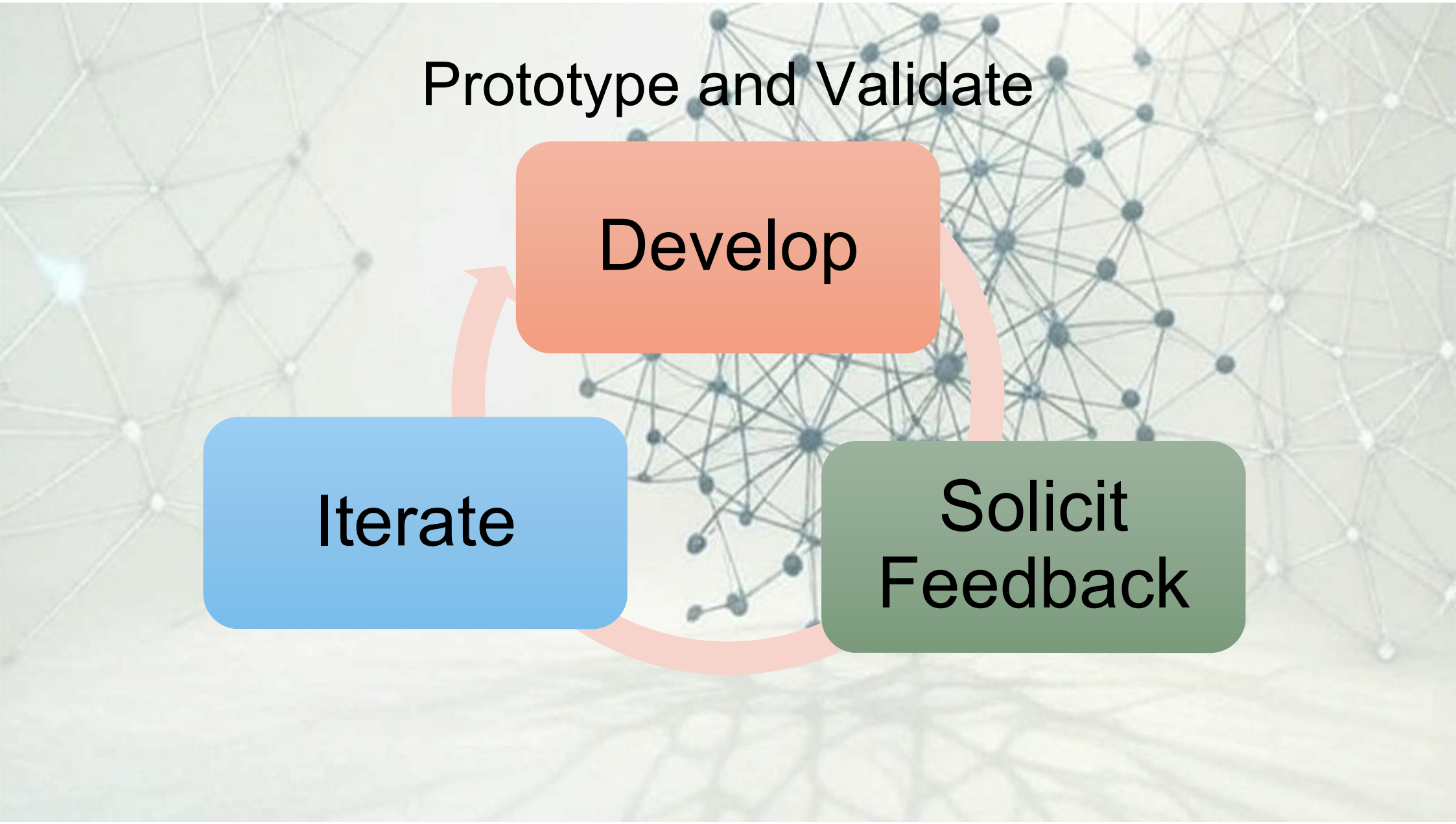


Define Success Metrics

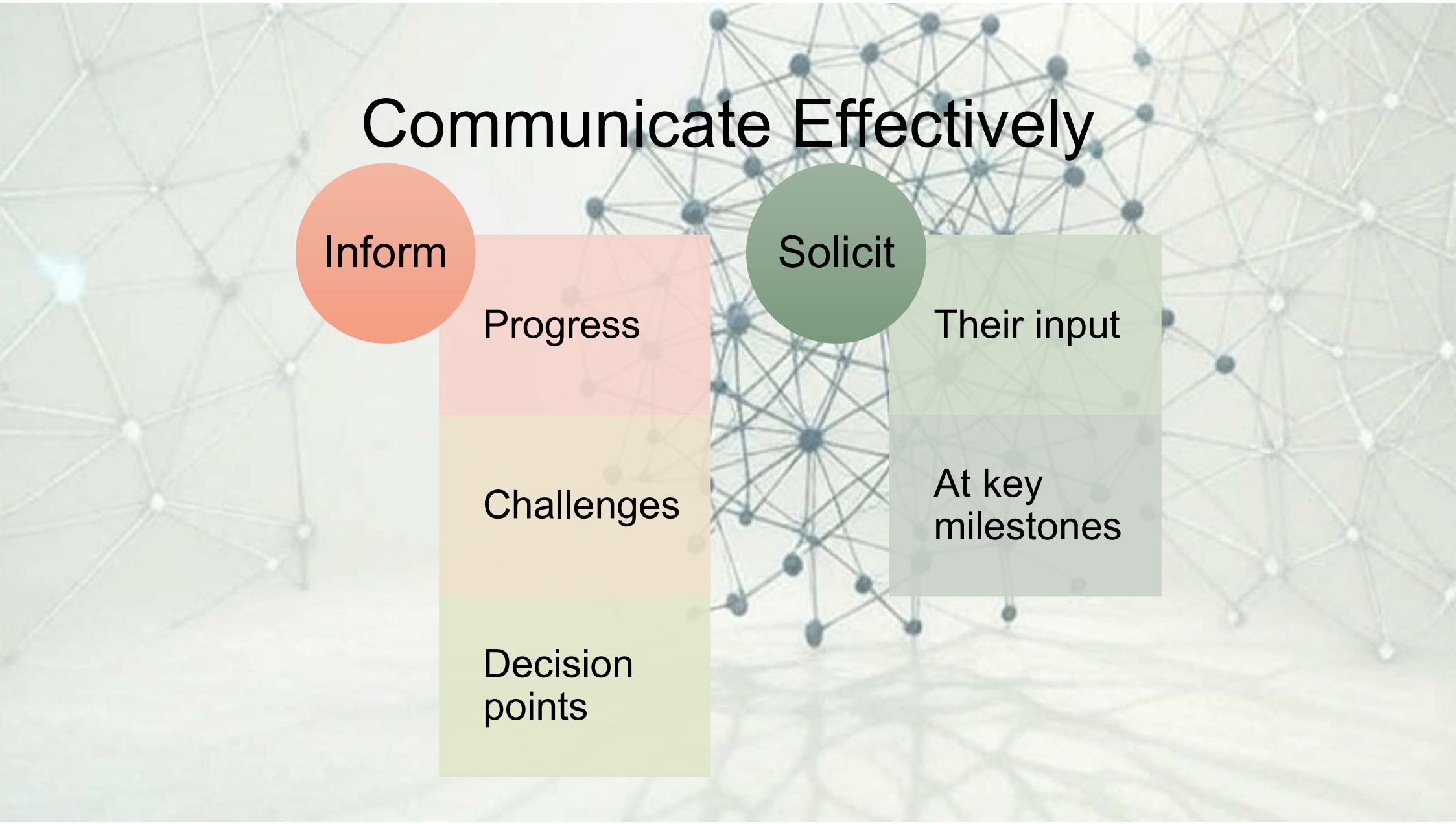
- Work with stakeholders
- Define clear and measurable success metrics
- Use to evaluate the effectiveness of AI solutions

Proof of Concepts vs. Prototype

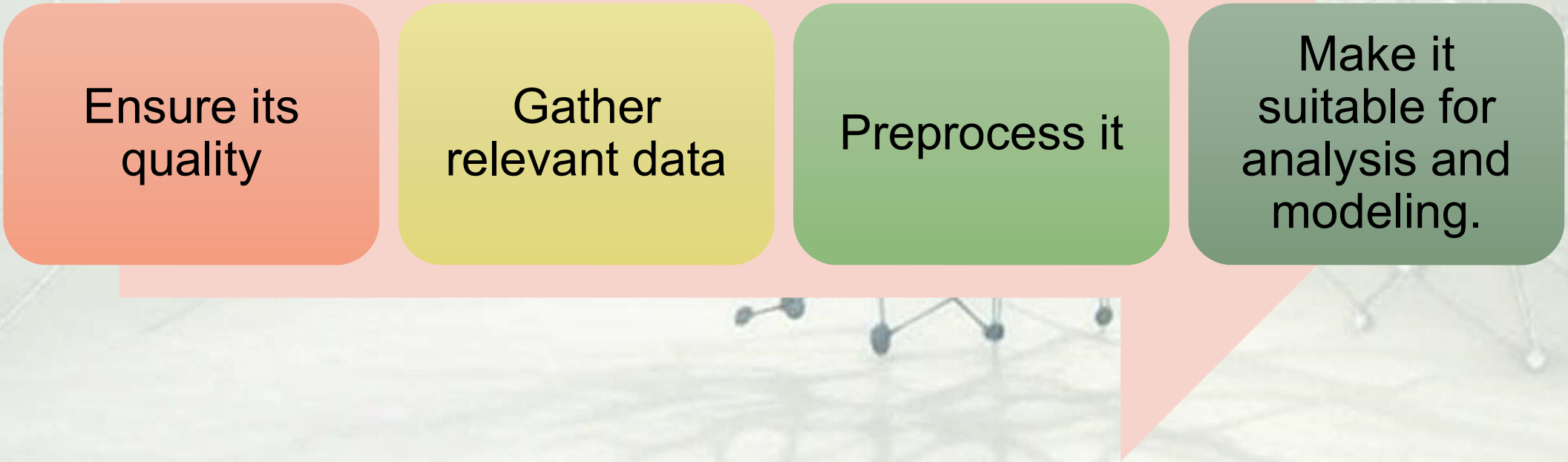
Proof of Concepts(POC)	Prototype
• Theoretical demonstration of a product/process/concept. • Determine whether an idea can be turned into a reality. • Test whether the idea is viable and explore the idea's potential t be developed or built. • Verify that the idea will function as envisioned. • Address how the proposed product or service will support organizational goals, objectives or other business requirements as a secondary goal.	• Very early draft of a product/process/concept. • Meant to turn a POC idea into a slimmed-down version of the end product that can be tested and evaluated for usability, functionality, and design. • Not expected to have all the features and functions of a market-ready product, nor is it expected to contain all the usability or aesthetics of a final product. • Gives stakeholders, project managers, executives and potential investors a draft of what the final product might be. • Allows makers to determine how best to develop the product when it moves into full production for a final, market-ready item


Prototype and Validate
Develop
Solicit Feedback
Iterate

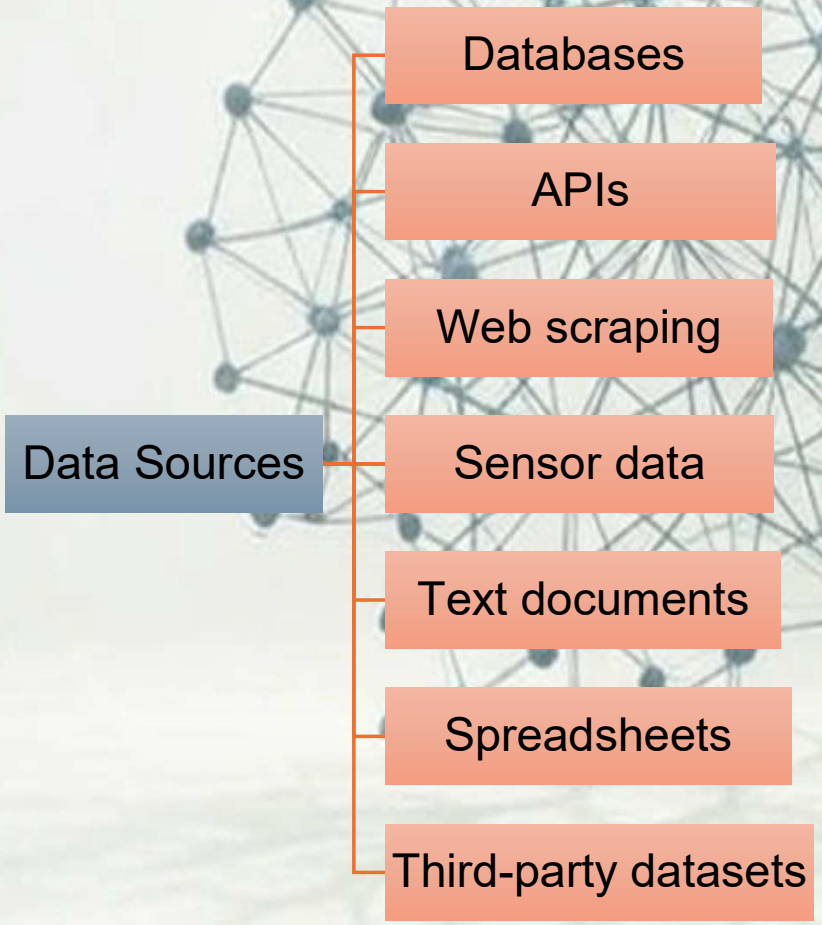
Communicate Effectively



How to collect and preprocess data



Identify Data Sources



Public Sources for Database

Kaggle Datasets

- **Kaggle**: One of the largest platforms for datasets, particularly for machine learning and data analysis. Great for structured data.

Google Dataset Search

- **Google Dataset Search**: A search engine for datasets across the web.

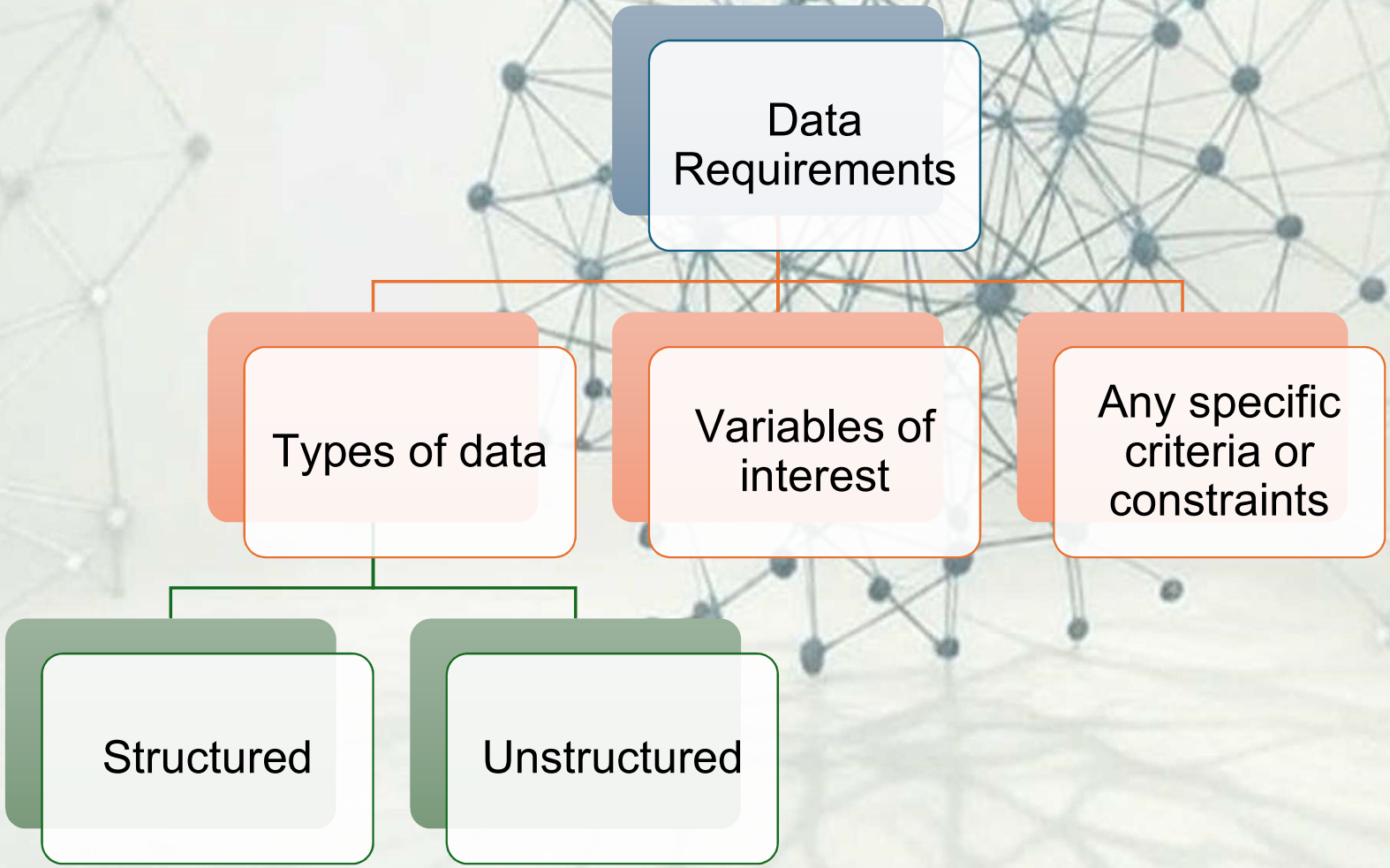
Ensure Ethical and Legal Considerations

The data is publicly available or you have permission to use it.

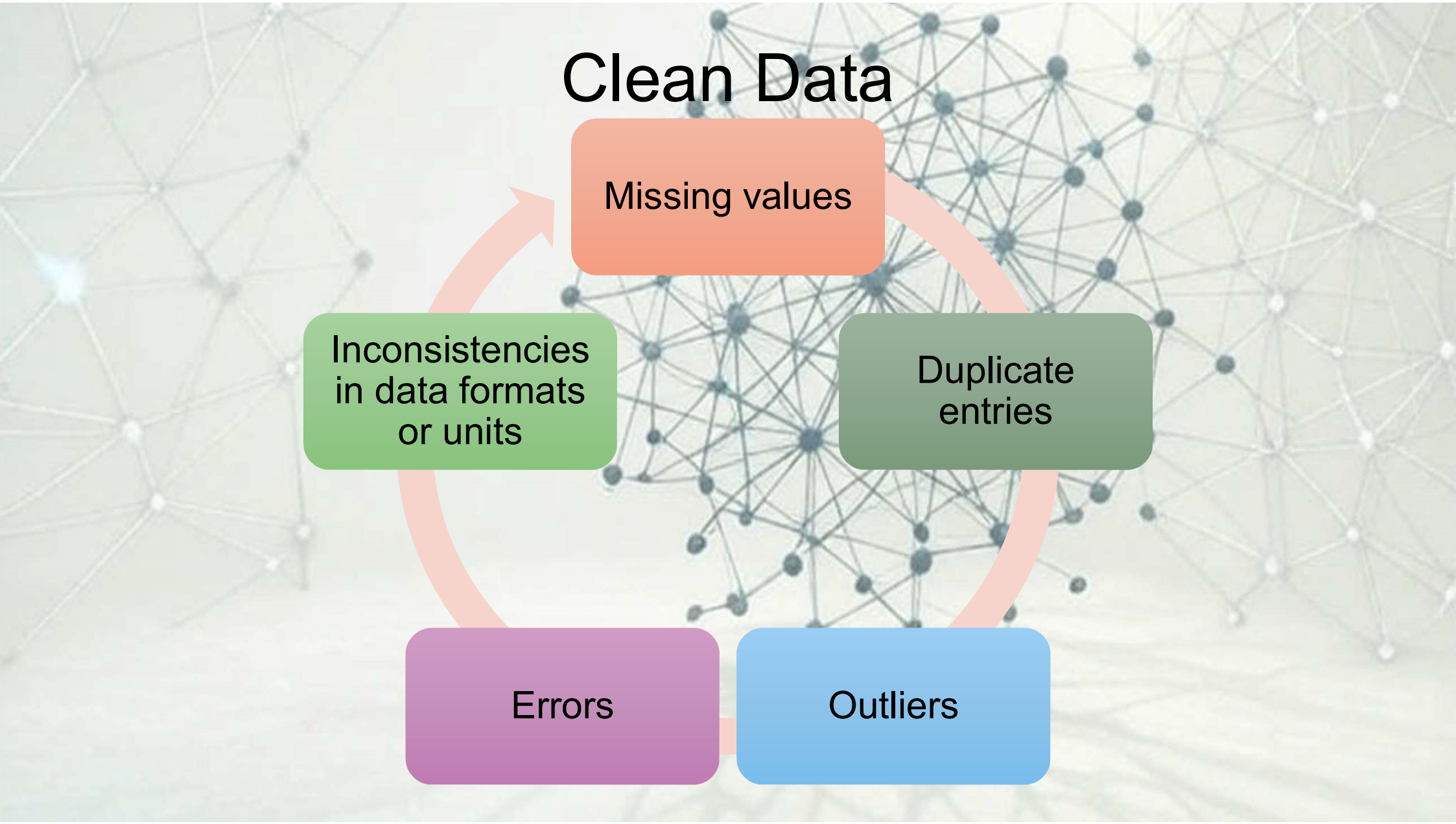
Cite sources where appropriate and respect licensing agreements.

If you're working with personal data, ensure compliance with data protection regulations (like GDPR in Europe or CCPA in California).

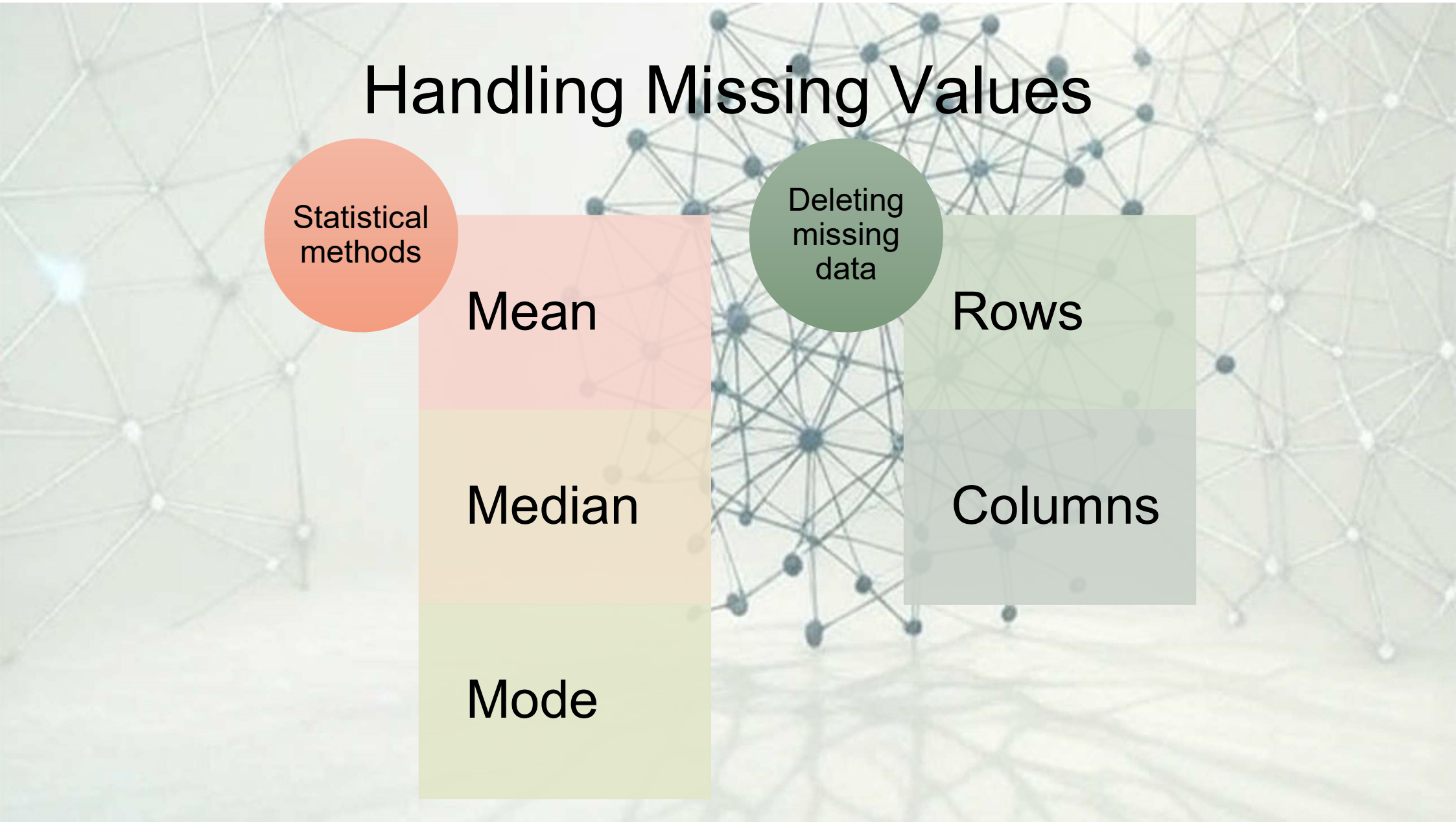
Define Data Requirements



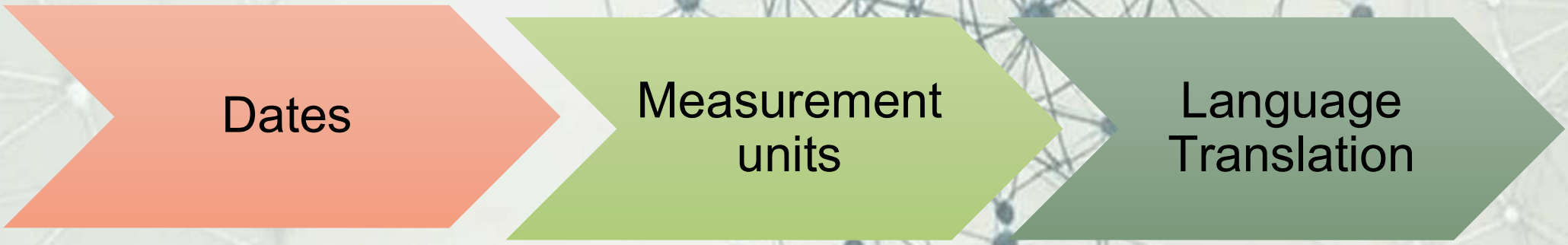
Assess Data Quality


Clean Data
Missing values
Duplicate entries
Outliers
Errors
Inconsistencies in data formats or units

Handling Missing Values



Standardizing data formats



What Next?

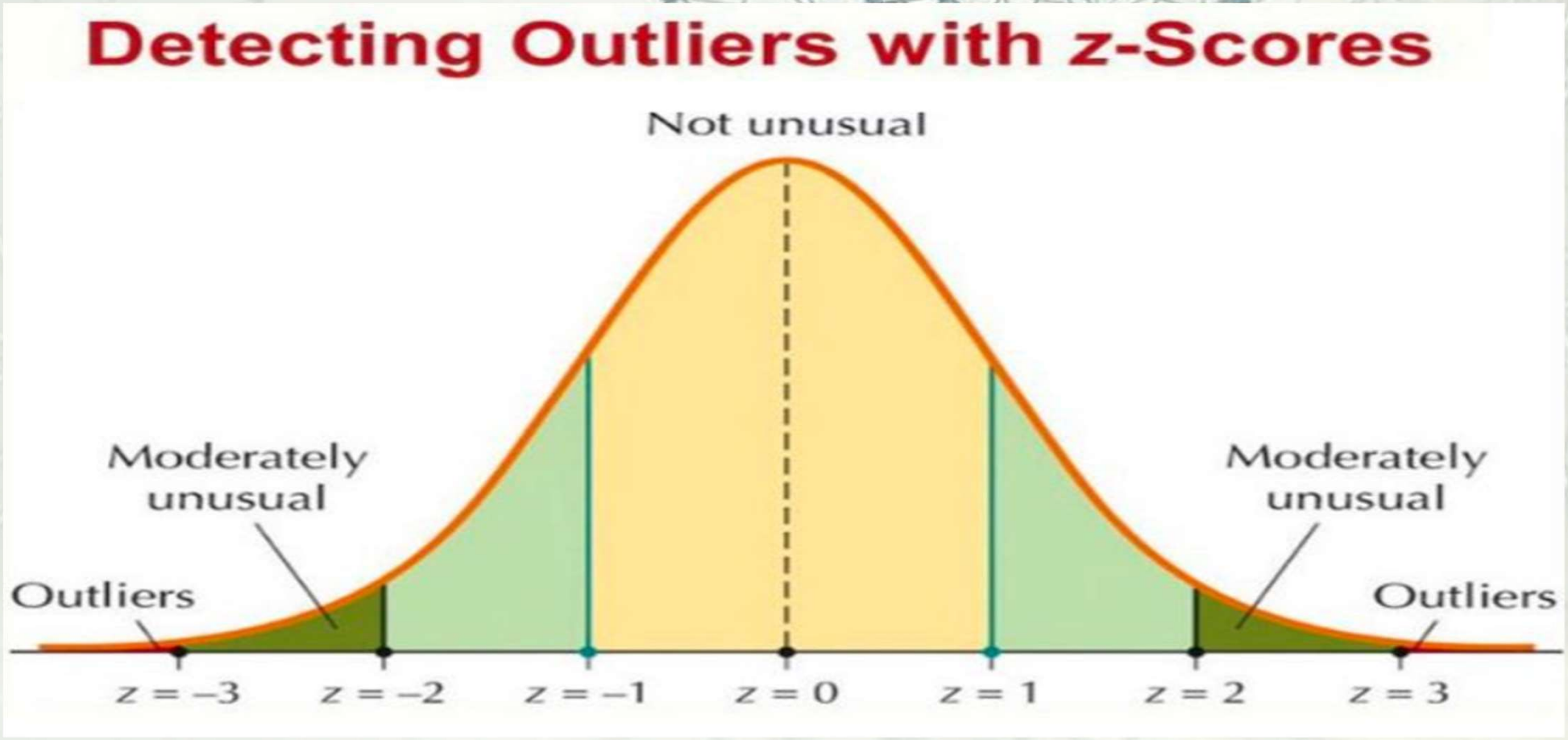
Removing Duplicates

- Remove duplicates data without losing important information

Encoding Categorical Variables

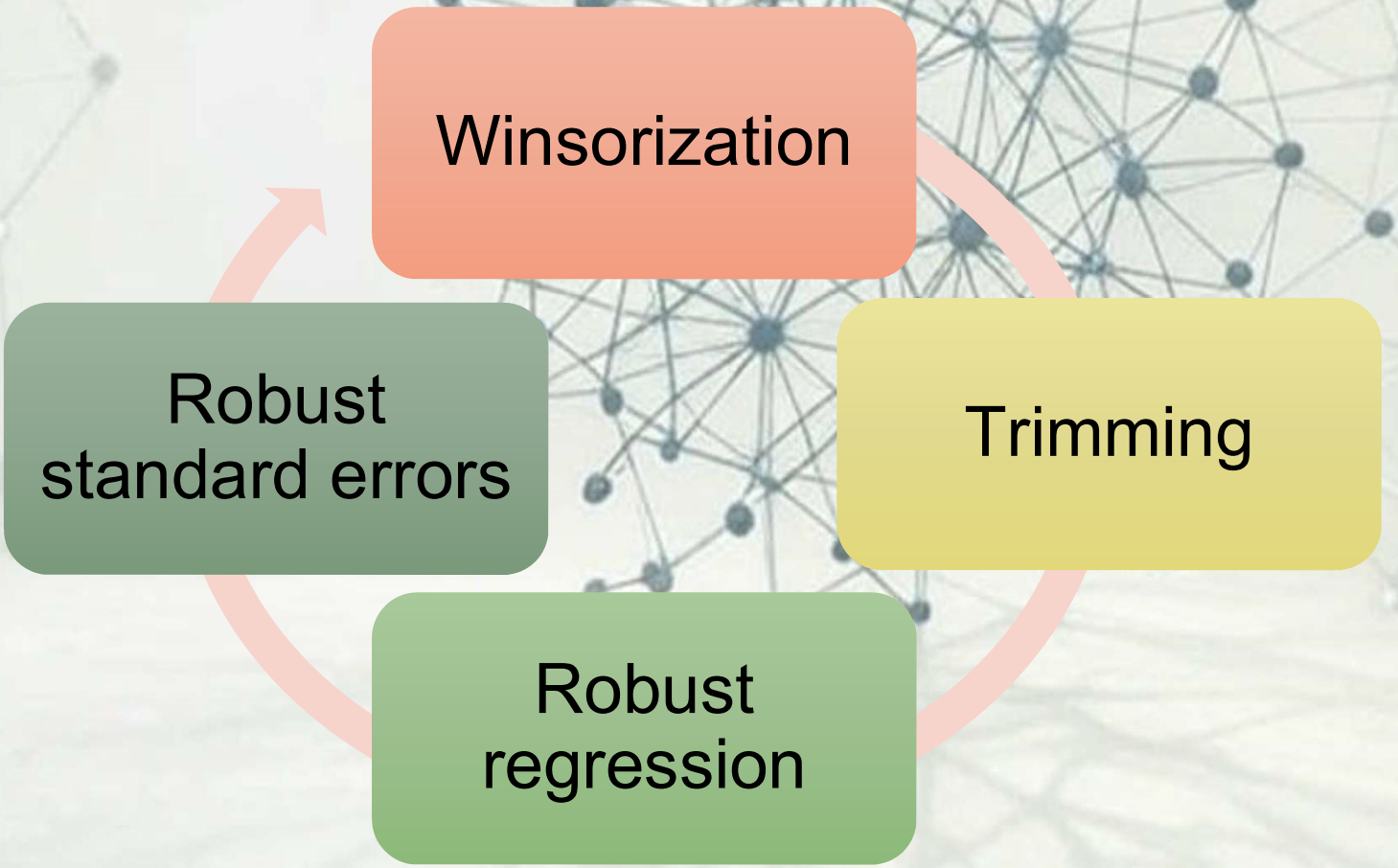
- Categorical variables represent data that can be divided into multiple categories but cannot be ordered or measured.
- Each category can be identified by a distinct label, and data points are allocated to these categories based on qualitative properties

Dealing with outliers- What are Outliers?



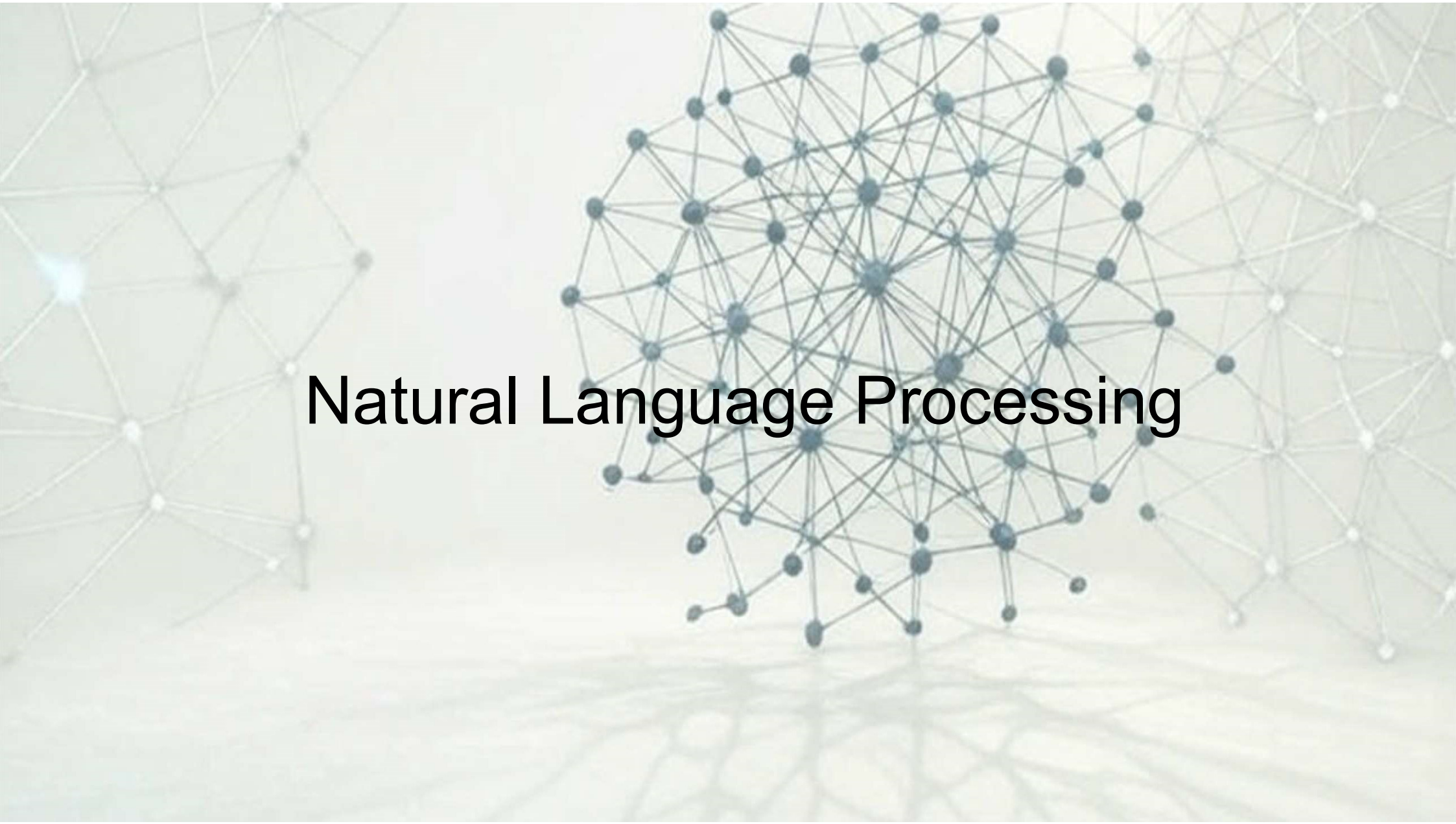
Dealing with outliers

Robust Methods for Handling Outliers



Visualization of data

- Matplotlib
- Seaborn
- Excel


Natural Language Processing

Word embedding's

- Suppose we have a small corpus of text documents consisting of three sentences:
  1) "The cat sat on the mat.“
  2) "The dog played in the park."
  3) "The bird sang in the tree.“
- To create word embedding's, we can use the Word2Vec algorithm, which learns distributed representations of words based on their co-occurrence patterns in the corpus. After training the Word2Vec model, each word in the vocabulary is represented by a DENSE VECTOR IN A CONTINUOUS VECTOR SPACE.

- Here's a simplified example of word embedding's for the words in our corpus:
  - "the": [0.2, -0.4, 0.1]
  - "cat": [-0.3, 0.2, -0.5]
  - "dog": [0.4, -0.1, 0.3]
  - "bird": [0.1, 0.5, -0.2]
  - "sat": [-0.2, -0.3, 0.4]
  - "played": [0.3, -0.4, -0.1]
  - "sang": [-0.1, 0.3, 0.2]
  - "on": [0.2, 0.1, -0.3]
  - "in": [0.3, -0.2, 0.1]
  - "mat": [0.4, 0.2, 0.3]
  - "park": [-0.2, 0.3, 0.1]
  - "tree": [-0.3, 0.4, -0.2]

What do the numbers mean in the word embedding's

- Suppose we have a small corpus of text documents consisting of three sentences:
  1) "The cat sat on the mat."
  2) "The dog played in the park."
  3) "The bird sang in the tree."
- To create word embedding's, we can use the Word2Vec algorithm, which learns distributed representations of words based on their co-occurrence patterns in the corpus.
- After training the Word2Vec model, each word in the vocabulary is represented by a dense vector in a continuous vector space.

- Here's a simplified example of word embedding's for the words in our corpus:
  - "the": [0.2, -0.4, 0.1]
  - "cat": [-0.3, 0.2, -0.5]
  - "dog": [0.4, -0.1, 0.3]
  - "bird": [0.1, 0.5, -0.2]
  - "sat": [-0.2, -0.3, 0.4]
  - "played": [0.3, -0.4, -0.1]
  - "sang": [-0.1, 0.3, 0.2]
  - "on": [0.2, 0.1, -0.3]
  - "in": [0.3, -0.2, 0.1]
  - "mat": [0.4, 0.2, 0.3]
  - "park": [-0.2, 0.3, 0.1]
  - "tree": [-0.3, 0.4, -0.2]

How are the word embedding’s used for Feature Extraction

- Suppose we have a small corpus of text documents consisting of three sentences:
  - "The cat sat on the mat.“
  - "The dog played in the park.“
  - "The bird sang in the tree.“
- To create word embedding’s, we can use the Word2Vec algorithm, which learns distributed representations of words based on their co-occurrence patterns in the corpus.
- After training the Word2Vec model, each word in the vocabulary is represented by a dense vector in a continuous vector space.

- Here's a simplified example of word embedding’s for the words in our corpus:
  - "the": [0.2, -0.4, 0.1]
  - "cat": [-0.3, 0.2, -0.5]
  - "dog": [0.4, -0.1, 0.3]
  - "bird": [0.1, 0.5, -0.2]
  - "sat": [-0.2, -0.3, 0.4]
  - "played": [0.3, -0.4, -0.1]
  - "sang": [-0.1, 0.3, 0.2]
  - "on": [0.2, 0.1, -0.3]
  - "in": [0.3, -0.2, 0.1]
  - "mat": [0.4, 0.2, 0.3]
  - "park": [-0.2, 0.3, 0.1]
  - "tree": [-0.3, 0.4, -0.2]

How are the word embedding's used for Feature Extraction