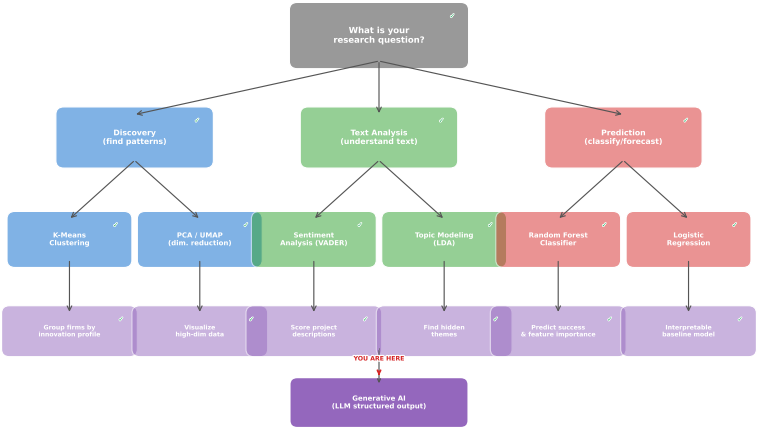
ML Technique Decision Tree



Your research question determines your method. GenAI extends all three branches.

A Tool That Extends Every Branch

Imagine a tool that has read
every innovation paper ever written.

Every annual report. Every patent filing.

But it also hallucinates, makes things up,
and is supremely confident when wrong.

Your job: get useful output without being misled.

GenAI does not fit neatly into Discovery, Text, or Prediction — it extends all three. A major upgrade and a new risk.

How Large Language Models Work

The core idea:

Predict the next word, billions of times, on the entire internet.

"The innovation project was a _____"

After training on trillions of words, the model learns:

- Grammar and structure
- Facts and relationships
- Reasoning patterns
- Domain-specific knowledge

Emergent capabilities:

Next-word prediction at scale produces unexpected abilities:

- Summarization
- Translation
- Classification
- Code generation
- Reasoning about novel problems

Key insight

Brilliant pattern-matcher. Dangerous factual claimer. They don't "understand" — they predict useful continuations.

GPT-4, Claude, Llama: billions of parameters trained on trillions of tokens. Scale drives capability.

What LLMs Can and Cannot Do for Research

Can do well:

- Classify text into categories
- Extract structured data from prose
- Summarize long documents
- Generate first drafts
- Code qualitative data consistently
- Translate between languages

Cannot rely on:

- Factual accuracy (hallucinations)
- Consistent counting/math
- Access to unpublished data
- Replacing human judgment
- Original theoretical contributions
- Causal reasoning

The rule: Use LLMs for *processing* at scale. Verify results with human oversight.

Always disclose AI assistance in your research. Follow your institution's AI policy.

Prompt Engineering: Getting Useful Output

The basics:

A "prompt" is your instruction to the LLM. Better prompts = better results.

Key techniques:

1. **Be specific:** "Classify as product/process/organizational innovation"
2. **Give examples:** Show 2–3 labeled cases
3. **Set the role:** "You are an innovation researcher"
4. **Constrain output:** "Respond only with JSON"

Example prompt:

```
Classify this innovation project description into one
of: [product, process, organizational, marketing].

Also extract: industry, key technology, innovation
stage.

Return as JSON with fields: type, industry, technology,
stage.

Description: ``{text}''
```

How you ask determines what you get. Vague questions get vague answers. Structured prompts get structured data.

Structured Output: Systematic Data Coding

The problem:

You have 500 project descriptions. Manual coding would take weeks.

The solution:

Force the LLM to output *structured* data (JSON) matching your codebook.

Benefits:

- Code 500 descriptions in minutes
- Consistent schema (no typos)
- Reproducible with same prompt
- Scalable to thousands

Example output:

```
{
  "type": "product",
  "industry": "healthtech",
  "technology": "AI diagnostics",
  "stage": "prototype",
  "sentiment": "positive",
  "novelty_level": "substantial",
  "key_challenge": "regulatory"
}
```

For your thesis

Use LLM coding as a *second coder*. Compare with manual coding for inter-rater reliability.

Instead of free-text responses, force the AI to fill in a structured form. Now it is a research instrument, not a chatbot.

Live Demo: Structured Output for Research

Jupyter Notebook: 06_structured_output.ipynb

- Load pre-generated LLM responses (cached)
- Examine the prompt template used
- Analyze coding consistency
- Compare LLM coding vs. ground truth
- Discuss: when to trust automated coding

Demo uses cached responses (no live API). The notebook shows how to set up live calls with your own API key.

Ethics and Responsible Use

Disclosure requirements:

- Report which LLM you used
- Document your prompts
- State what was AI-generated
- Follow journal guidelines
- Check institutional policy

Reproducibility:

- Save all prompts and responses
- Note model version and date
- LLM outputs may change over time

Risks to manage:

- Hallucination: LLM invents plausible-sounding but false information
- Bias: Training data biases transfer to outputs
- Privacy: Don't send confidential data to APIs
- Over-reliance: AI assists, doesn't replace judgment

Golden rule

Human oversight on every output.

Can you cite an AI? Can you trust its classifications? Responsible use requires transparency and verification.

GenAI as Research Assistant: Use Cases

Literature review:

- Summarize 50 papers in hours
- Extract key findings systematically
- Identify gaps and contradictions
- Generate initial taxonomy

Hypothesis generation:

- "Given these patterns, suggest 5 testable hypotheses"
- Brainstorm mechanisms
- Challenge your assumptions

Data coding & analysis:

- Code open-ended survey responses
- Extract variables from annual reports
- Classify patents by innovation type
- Sentiment analysis at scale

Writing support:

- Drafting methodology sections
- Improving clarity and structure
- Translating between languages

Use GenAI to accelerate, not to replace. The insight and theory must be yours.

Your Turn: Where Does This Fit?

"If an LLM classifies 10,000 abstracts for you,
is that YOUR analysis or the AI's?
Where does authorship begin?"

Take 60 seconds. Discuss with your neighbor.

Map Check: Where Are We?

Branches explored:

- ✓ All 7 tools deployed: clustering, PCA, sentiment, topics, RF, logistic, GenAI

Next on the map:

- The complete map: your thesis path
- Choosing your own route through the tree

Every tool on the map has been explored. Next: assembling the complete picture and finding your path.