

Chapter 11: The AI 90% of Consultants Don't Know

In the last chapter, we mastered the cheat code for creating polished, client-ready workshop decks in under an hour. We learned how to *package* our expertise. Now, we move from presentation to substance. We're going to cover the one core skill that separates the six-figure experts from the prompt engineers who are just "cosplaying" as consultants.

I'm going to let you in on a secret. Machine learning used to be my one true (nerdy) love.¹

And then, Generative AI was born, and most people stopped caring about it.

I come from a 10-year background of working not in generative AI, but with machine learning (ML) models, stats, prediction models, and regression algorithms. I learned Natural Language Processing (NLP) manually, back when you had to write endless lines of Python code just to figure out if a sentence was positive or negative. There's an appreciation you gain for the techniques that existed before you could just use natural language to accomplish a task.¹

This "old world" knowledge isn't outdated. It's the secret weapon. It's the bedrock of real, defensible, high-value consulting.

The Cosplay Consultant vs. The Six-Figure Expert

Let's get this out of the way. **One skill separates six-figure consultants from prompt engineers cosplaying as experts: knowing when NOT to use ChatGPT (and language models).**¹

You hear the rhetoric on YouTube all the time. It's the siren song of the "AI Gold Rush": "All you need to do to get \$50,000 or \$80,000 a month is learn AI automation, build automations, and you will be sipping margaritas on the beach with a Lambo".¹

Realistically, that's probably not going to work in most companies. You need a combination of the AI-generalist skillset. The 90% of "consultants" out there only know Generative AI. They know ChatGPT, they know Claude, and they know how to write a prompt. They're chasing what's "sexy." But the old AI isn't sexy, and as I've said before, **"sexy does not always mean**

ROI".¹

The expert—the true consultant—is solving the high-value, high-ROI problems using the "boring" tools that actually run the world.

A Backdoor Cheat Code to What Enterprises *Really* Want

If you want a cheat code for what companies are actually hiring for, go look at their job descriptions. This literally gives you a backdoor into what people, what teams, what enterprises are really looking for.¹

I get job offers on LinkedIn weekly, and I've made an observation: 90% of roles that want something related to language models or GenAI *also* want you to have a background in machine learning.¹

I screenshotted the last one I got. The title was "Assistant Vice President of AI Engineering." Look at the requirements:

- "Design, build, and serve LLMs to solve complex business challenges."
- "Experience in AI engineering and/or **machine learning** with a focus on LLMs (NLP)."
- "Deep expertise in writing, reviewing, and productionizing code." ¹

When a client posts a job description, they are broadcasting their true, internal needs. Their need isn't for a "prompt engineer." It's for an architect who understands *both* worlds. When you walk into a discovery call speaking this hybrid language, you are immediately elevated from a vendor to a high-value peer. You're not pitching them something new; you're aligning with a corporate priority they've already budgeted for.

Why Your Banking App Isn't Built on ChatGPT

So why do enterprises still care so much about this "old" AI? It comes down to one core concept: **the scale problem**.¹

Most of the things on your phone that you rely on every day—the apps that *must* be reliable—are still using machine learning, not GenAI.

- **Banking Apps:** My banking app is still using ML to predict whether or not I should be approved for a mortgage, a credit card, or a loan.¹
- **Biometric Analysis:** When you verify your driver's license on an app, it's using "older-school image recognition models with ML, not vision LLMs".¹

Why? For two critical reasons:

1. **Cost (Dollars):** "At scales of millions of computations per day, you probably still won't use GenAI. It's still way too expensive".¹
2. **Reliability (Determinism):** GenAI models have hallucination rates. Vision LLMs are "still not fully there at scale".¹ A bank *needs* its loan approval algorithm to be deterministic, not "creative." The "old world advantage" is "a lot more determinism and reliability".¹

As a consultant, you have to be the one who understands this. You have to be the one to tell the client, "Yes, we can use GenAI for your chatbot, but for your high-volume, high-stakes fraud detection system, we *must* use traditional ML."

The Artist vs. The Predictor: What to Use When

This is the simplest way to think about the two worlds of AI.

Dimension	Generative AI (The Artist)	Traditional ML (The Predictor)
Core Task	Creating & Interpreting	Predicting & Classifying
Data Type	Unstructured (text, images)	Structured (numbers, categories)
Output	Creative & Probabilistic	Deterministic & Accurate
Key Use Case	"Talking to humans" ¹	"Predicting something" ¹

A Consultant's Field Guide to "Old-School" AI

Okay, I can already feel my old professor vibes seeping in, so I'll keep things grassroots.¹ You don't need to become an ML engineer overnight. You just need to be able to speak to the two main categories to sound smarter than 90% of the people in this field.

Category 1: Supervised Learning (Predicting a Specific Future)

This is the one you'll encounter most often. "Supervised" simply means you have "something you were predicting for, with labeled training data".¹ You have a dataset of past outcomes, and you want to predict a *future* outcome.

My past use case was building algorithms to see "whether or not I could get pattern recognition of... someone who should be trusted with a mortgage".¹

This breaks down into three simple types:

1. **Classification:** Predicting a True/False outcome. "Should this person be approved for the loan? Yes or no".¹
2. **Multi-class Classification:** Predicting a non-binary outcome. "If approved, what loan amount? 6+ possibilities".¹
3. **Regression:** Predicting a continuous value. "Predicting... the exact loan amount, not just the categories".¹

Here's the good news: You now have a luxury I didn't have. I had to write all this code by hand. You can now use tools like **Claude Code** or **Cursor** to automate the process of writing a lot of this ML code.¹

Category 2: Unsupervised Learning (Finding the Hidden Tribes)

This is where the AI learns for itself. "You don't tell anything to the AI. You give it a dataset and it tries to mathematically find ways to look for similarities based on what it sees".¹

The primary technique here is **Clustering**. The algorithm "tries to segregate a dataset into different groups".¹

A real-world example: I used to do customer segmentation on Coca-Cola data. We would

"give it all customer data and say: Go find me who these people are based on attributes *without me telling you who they should be.*" The algorithm would come back and create 6-8 distinct clusters of customer types, revealing hidden patterns in the market.¹

The Solution Architect's Matrix: A 12-Scenario Field Guide

Now that you have the theory, here is the practical, tactical playbook. These are the scenarios you will face, and these are the expert-level answers that separate you from the amateurs.¹

The AI Solution Architect's Matrix

Scenario	Recommendation	Core Rationale (Why?)
1. Fraud Detection	Machine Learning	Scale, cost, need for high accuracy. GenAI is too expensive and not deterministic enough for thousands of transactions per second.
2. Customer Support	Generative AI	GenAI's "bread and butter." It's built for natural language and talking to humans. (Can be hybrid, with ML for routing).
3. Lead Scoring	Machine Learning	This is predictive analytics on structured data (job title, company size). It's a classic regression/classification problem.

4. Content Personalization	Hybrid (ML + GenAI)	This is the superpower. ML does the initial segmentation (finds the cluster), then GenAI generates the personalized content for that specific cluster.
5. Pricing Optimization	Machine Learning	This is math-heavy, structured data (cost, demand) and requires precision. You "don't want hallucinated pricing that could cost millions."
6. Chatbots	Generative AI	LLMs are now so good at conversation that they are the primary tool. (ML can still help with intent detection/routing).
7. Churn Prediction	Machine Learning	This is a binary outcome (will churn/won't churn) based on structured usage data. It's a perfect classification model problem.
8. Recommendation Systems	Hybrid (but complex)	These systems are "very math heavy." GenAI is now being sprinkled in to "alleviate a lot of the manual work" previously required.
9. Image Analysis	Depends on Scale	<p>Low Scale (1000s/week): GenAI vision models can work if there's "error tolerance of 15-20%."</p> <p>High Scale/Fraud:</p>

		Traditional ML (computer vision) is required for cost and reliability.
10. Document Processing	Flexible (Hybrid)	Depends on what you're extracting and how much structure the documents have. Often a mix of ML, GenAI, and other tools.
11. Forecasting	Machine Learning	This uses specific ML models (like ARIMA or SARIMA). GenAI can <i>help you write the code</i> , but ML runs the actual algorithm.
12. Inventory Optimization	Machine Learning	Because of the high volume of metrics, numbers, and elements involved, ML is the best choice.
Bonus: Sentiment Analysis	95% Generative AI	This used to be a pain with old NLP. GenAI is built for language and is far superior. A hybrid model can add a "deterministic failsafe" if needed.

Your Decision-Making "Cheat Sheet"

If that matrix is the encyclopedia, this is the cheat sheet you need to memorize. These are the three fundamental rules.¹

Use Generative AI when:

- You are **creating content**.
- You are **talking to humans** (or having an AI talk to humans).
- You are dealing with **unstructured data** (like text).
- You need **creative output**.
- You **don't need 100% accuracy** (95% is okay; there's some tolerance).

Use Traditional ML when:

- You are **predicting something**.
- You **need 95%+ accuracy** (high stakes, deterministic).
- The problem is **numeric in nature**.
- You are dealing with **very structured data** (math-heavy).
- **Money at scale is involved** (cost per computation matters).

Use Hybrid when:

- You have **complex, multi-step workflows**.
- You need **ML to do the math/categorization/clustering**.
- You need **GenAI to interpret the results** of that clustering.

The "Superpower": Where Two Worlds Meet

Look at that last bullet point under "Hybrid." That's the real secret. That's the superpower.¹

In the "old world" with that Coca-Cola data, the ML algorithm was great at *finding* the clusters, but a human (me) had to manually go through the tables to interpret what those clusters *meant*.¹ Now, you unleash the hybrid model.

You use ML to do the heavy, deterministic math (the "what").

You then feed its output to GenAI to provide the natural language interpretation (the "so what").

This hybrid approach is the antidote to the "cold truth that 70% of AI projects fail".¹ They fail because people get obsessed with the outcome and use the wrong tool for the job. They try

to make GenAI do complex math and get frustrated when it fails, because they don't understand the journey "from zero to building proper infrastructure".¹

From "Bad Consultant" to "True Expert"

Let's put this all together. This is the moment that defines your career. A client comes to you and says:

"We want to use AI to predict customer churn."¹

The "bad consultant"—the "prompt engineer cosplaying as an expert"—says this:

*"Great! We'll build a ChatGPT agent to analyze your customers."*¹

This is wrong. It's a shallow, tool-first solution. You're using a creative tool for a mathematical prediction problem. You've instantly lost all credibility.

The "good consultant"—the true expert, the solution architect—says this:

*"This is a classification problem with structured data. We need ML for prediction accuracy. We could, however, add GenAI to generate personalized retention messages once ML identifies the at-risk customers. Let me show you the hybrid approach."*¹

Boom.

In that one response, you have (1) correctly diagnosed the problem ("classification problem"), (2) selected the right tool for the core task (ML), and (3) intelligently added GenAI as a *value-add*, not a crutch. That is the expertise signal.¹

The Final Positioning

This is the final positioning. You're not just an "AI consultant" who knows prompts.¹

You are a consultant who understands:

- When GenAI is perfect.

- When it's the completely wrong tool.
- When traditional ML is required.
- And—most importantly—how to combine both for superpowers.¹

That knowledge alone separates you from 90% of people calling themselves AI consultants.¹

Now that you have the deep *substance* that 90% of consultants lack, how do you apply it? How do you look at a client's messy tech stack and decide *which* tools to use for *which* problems? That's the next step. It's time to stop *thinking* like a consultant and start *building* like one. It's time to become an AI Solution Architect.¹

Works cited

1. Consulting Playbook Day 11_The AI 90% of Consultants Don't Know.txt