Introduction to Data Science

Session 1 — Core Concepts, Real Stories & Roles

Welcome to your first step into the world of data science. By the end of this session, you'll be able to define data science, recognize its real-world power, and understand who does what in a data team.

SESSION 1 FOUNDATIONS

Session Outcomes

By the end of this session, you will be able to:

01

Define Data Science

Explain it clearly in plain English — well enough to describe it to a friend.

02

Differentiate What It Is & Isn't

Avoid the most common misconceptions students bring into class.

03

Explain Real-World Applications

Use simple case stories to ground abstract ideas in everyday life.

04

Describe the 3 Pillars & 5 Roles

Understand what each pillar contributes and how roles collaborate.

Session Flow

Here's how we'll move through today's material — a structured journey from warm-up to recap.

Warm-Up

Where do you see data science in daily life? (2–3 mins)

Section 1

What is Data Science? The Ask → Collect → Understand → Act loop

Section 2

Netflix, Google Maps & Amazon success stories

Section 3

The 3 Pillars of Data Science

Section 4

5 Data Roles + Pass-the-Baton story

What Is Data Science?

> Data science is the habit of asking a clear question, gathering the right facts, finding patterns using math, and turning those patterns into actions that improve over time.

If you remember only one thing today, remember this loop:

- Ask ❓
- Collect 🗃️
- Understand 🧠
- Act 🚀

Breaking Down the Four Verbs

Ask ❓ — Sharp the Question

Convert a vague wish into a precise question. Without a clear target, you can't measure success.

Collect 🗃️ — Gather Clean Data

Gather, clean, and organize relevant data. Messy inputs lead to misleading outputs.

Understand 🧠 — Spot Patterns

Use statistics or algorithms to find patterns. This explains what's happening and what may happen next.

Act 🚀 — Drive Real Change

Turn insights into a decision, rule, report, or feature — then track and improve. Insights that don't change reality are wasted.

✉️ Story 1 — The Spam Filter

How your inbox stays clean — automatically.

Ask — "Can we automatically keep junk out of inboxes?" Convert a vague annoyance into a measurable classification problem.

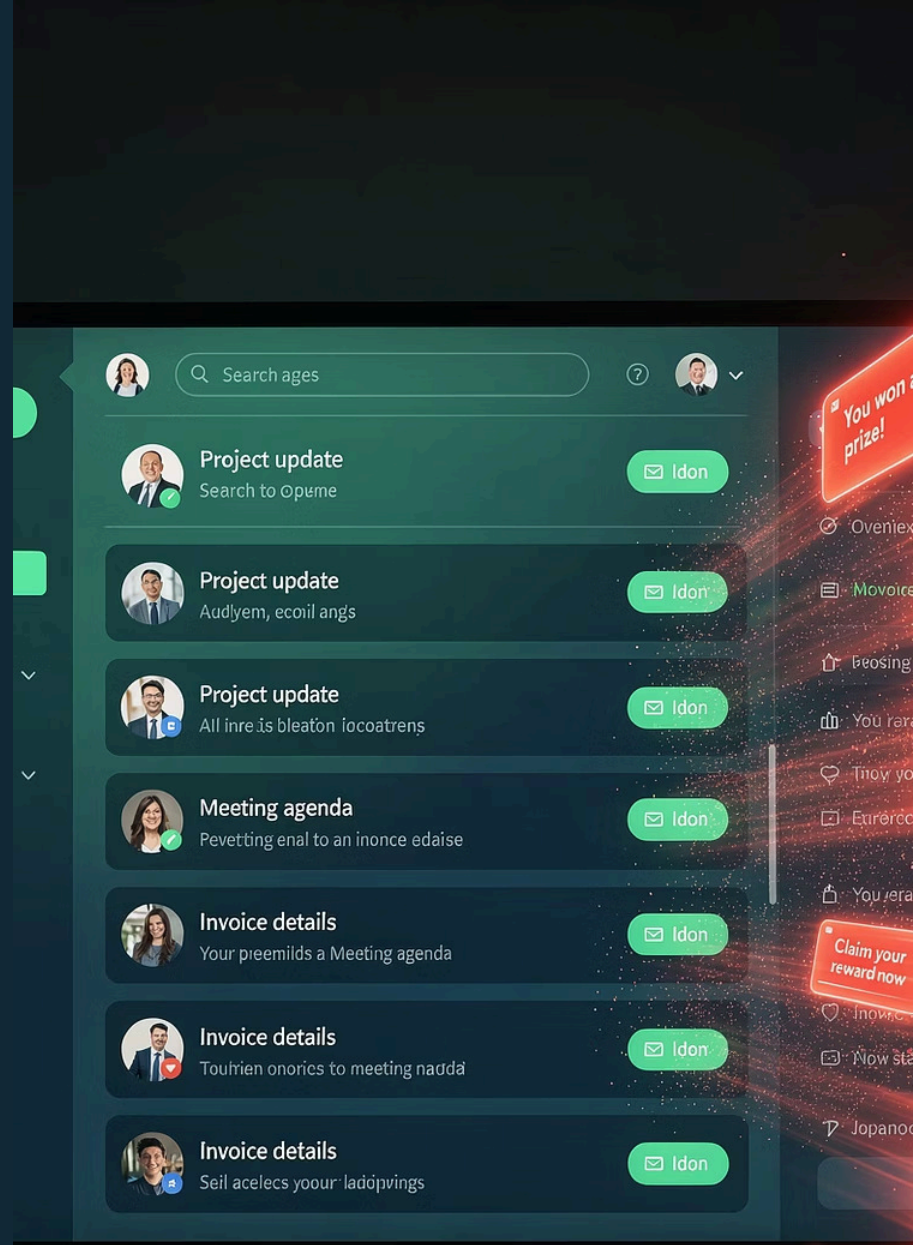
Collect — "Millions of past emails, each labeled spam or not spam." Includes sender domain, subject line keywords, link patterns, and user behavior.

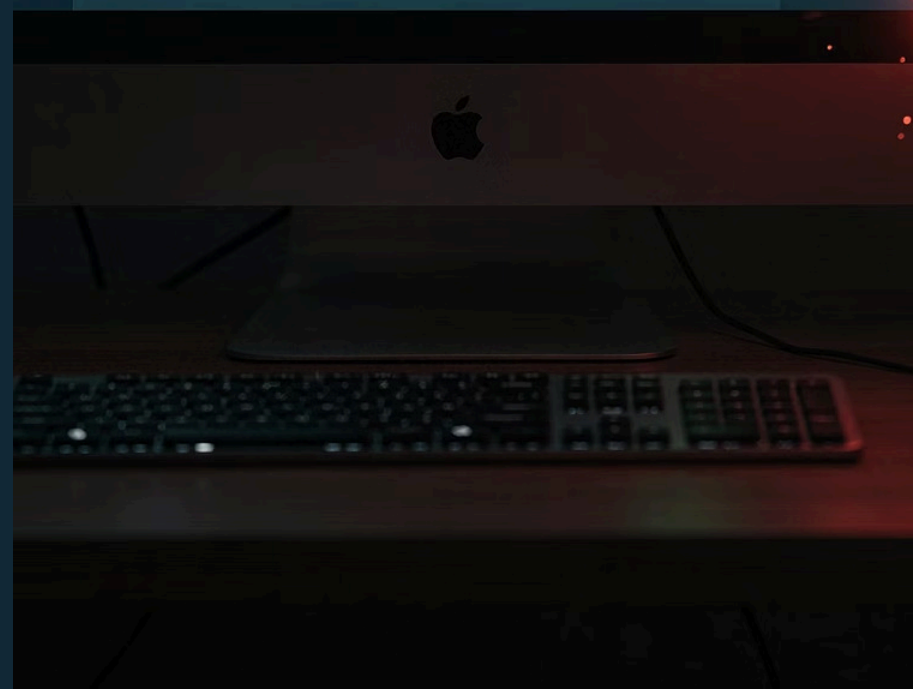
Understand — "Spam emails share patterns: certain phrases, suspicious domains, unusual formatting." A model learns to score each email by risk level.

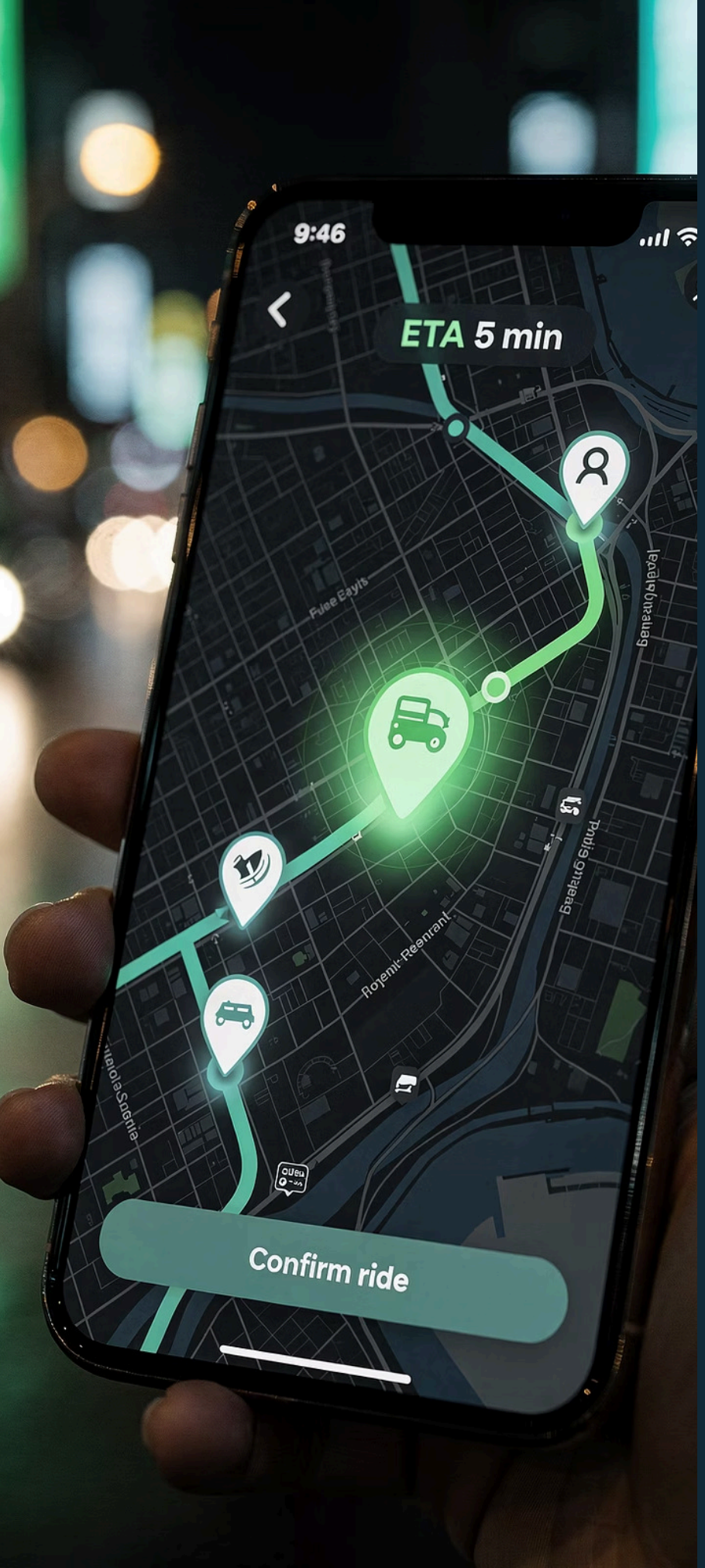
Act — "Every incoming email is scored. High score → routed to Spam folder instantly." Rules update as new spam patterns emerge.

Impact — "Less junk, better engagement, fewer phishing attacks." Gmail blocks 99.9% of spam using this exact loop.

🍎 Tip: — "What happens if the training data has mislabeled emails?" → Introduces the concept of garbage in, garbage out.







🚗 Story 2 — Ride-Hailing ETA

How Uber & Grab predict your pickup time to the minute.

Ask — "How long until the driver arrives and completes the trip?" Accuracy matters — wrong ETAs cause cancellations and lost trust.

Collect — "GPS traces, live traffic feeds, historical trip data, weather, time of day, and driver speed profiles." Billions of data points updated in real time.

Understand — "Compare this trip to thousands of similar past routes." Models learn that Friday evenings add 8 min, rain adds 5 min, school zones slow things down.

Act — "Display ETA in the app. Recalculate every 30 seconds as conditions change." Driver and rider both see live updates.

Impact — "Fewer cancellations, better driver allocation, higher satisfaction scores." Uber processes 18 million trips per day using this loop.

🍎 Tip : Ask — "What if there's a sudden road closure not in the data?" → Introduces the idea that models need live data streams, not just historical data.

🕐 Story 3 — Daily Step Goal

How your fitness tracker sets a goal that's hard enough — but not impossible.

Ask — "What daily step goal will push the user without demotivating them?" Too easy = no benefit. Too hard = user gives up.

Collect — "Last 30 days of step counts, heart rate trends, sleep data, and day-of-week patterns." Your Monday behavior is very different from your Sunday behavior.

Understand — "Weekday vs weekend patterns emerge. The model identifies your personal baseline and peak capacity." It knows you average 6,200 steps on Tuesdays but 9,800 on Saturdays.

Act — "Goal increases by 5–10% if you've been consistent. Decreases slightly after missed days." Personalized nudges sent at your most active time of day.

Impact — "Goals feel achievable → motivation stays high → long-term habit formation." Fitbit users with adaptive goals walk 20% more steps per week on average.

🍎 Tip: This is a great example of a feedback loop — the model learns from your behavior and adjusts.

What Data Science Is NOT

☐ Not Just a Pretty Chart

Charts show *what* happened. Data science explains *why*, predicts *what's next*, and recommends *what to do*.

One-liner: A chart shows data; data science extracts meaning.

☐ Not Coding Without Purpose

Writing complex code doesn't equal data science. Coding is a tool to answer a question or build a system.

One-liner: Good code supports insights; great data science supports decisions.

☐ Not Guesswork or Gut Feeling

Data science is evidence-based. Intuition can frame the question, but conclusions must come from data.

One-liner: Data science removes bias, rather than reinforcing it.

🥐 Activity 1 — The Bakery Croissant Problem

"Apply the 4-verb loop to a real business problem."

🎯 The Challenge

A local bakery bakes croissants fresh every morning. Too many = waste and lost profit. Too few = missed sales and unhappy customers. Your job: design a data science solution.

Your Task — 1 sentence each:

Ask ❓ — Write the precise business question the bakery needs to answer.

Collect 🗃️ — What data sources should they track? (Think: time, weather, events)

Understand 🧠 — What patterns would you look for in the data?

Act 🚀 — What specific action should the bakery take tomorrow morning?

✅ Sample Strong Answer

Ask:

"How many croissants should we bake each day to minimise waste while meeting demand?"

Collect:

Daily sales logs, day of week, local weather, school holidays, nearby events, leftover count at closing.

Understand:

Weekends sell 40% more. Rainy days drop footfall by 15%. School holidays spike morning traffic.

Act:

Bake baseline 80 + apply rules: +20% Sundays, -10% rainy days, +30% school holidays. Review accuracy weekly.

🍎**Tip:** After sharing answers, reveal the "Act" step. Ask — "What happens if the bakery only uses last week's data?" → Introduces the risk of small sample sizes and seasonal bias.

Section 2 — Big-League Success Stories

Netflix, Google Maps, and Amazon prove that data science is not magic ✨ — it's the same loop repeated at massive scale.

The Core Insight

Every billion-dollar data product follows the same 4 verbs: **Ask → Collect → Understand → Act**. The scale changes; the logic doesn't.

Why These Stories?

Students already use these products daily. Connecting familiar experiences to data science concepts makes abstract ideas immediately tangible.

🎬 Story #1 — Netflix Recommendations

Ask

What should we show each person next so they enjoy it and keep subscribing?

Collect

Plays, pauses, likes/dislikes, search history, device context, content metadata (genre, actors, length).

Understand

Group similar viewers; learn what each group enjoys next. Run A/B tests on rows, thumbnails, and trailers.

Act

Rank and personalize the homepage in real time when you open the app.

> 💡 Key idea: "Spot patterns in similar people, then personalize one person at a time." Impact: More watching → lower churn → higher revenue.

🗺️ Story #2 — Google Maps ETA

"How Google predicts your arrival time — to the minute."

Ask

"How long will this exact trip take right now, given live conditions?" Wrong ETAs erode trust. Accuracy is the product.

Collect

"Live GPS speeds from millions of phones, historical street timings, accident reports, road closures, weather alerts, and time-of-day patterns." Every phone on the road is a live sensor.

Understand

"Roads are modelled as a network. The model learns how slowdowns ripple — a jam on one street affects three others." It compares your route to thousands of similar past trips in real time.

Act

"Display ETA the moment you start navigation. Recalculate every few seconds as conditions change." Reroute suggestions appear automatically when a faster path opens up.

Impact

"Better ETAs → fewer delays, fewer wrong turns, higher user trust." Google Maps serves 1 billion users per month using this exact loop.

🍎 Tip — "What data does your phone send to Google Maps right now, without you knowing?" → Opens a discussion on data privacy, consent, and the trade-off between convenience and surveillance.

📦 Story #3 — Amazon Demand Forecasting

Ask ❓

How many units of each item should we stock, and where, to deliver fast without overstock?

Collect 🗃️

Years of sales, price changes, promotions, seasonality, holidays, local events, and weather signals.

Understand 🧠

Learn seasonal patterns, detect spikes (sudden trends), and predict a **range** — not one perfect number.

Act 🚀

Automate purchase orders and warehouse placement nightly. Less waste, fewer stockouts, faster delivery.

> 💡 Key idea: Forecasting is about **risk management**, not perfect prediction.

🧠 Activity 2 — Reflection

Choose the project that surprised you most: **Netflix / Maps / Amazon**

Discussion Question:

> Which step — Ask, Collect, Understand, or Act — was the toughest in your chosen story, and why?

"**Collect** is hardest because data is messy and scattered across many sources."

"**Ask** is hardest because the wrong question leads to a completely useless model."

"**Act** is hardest because deploying changes affects real users and carries real risks."

Section 3 — The Three Pillars of Data Science

> Think of a three-legged stool 🪑 — if one leg is weak, the whole thing wobbles.

⚙️ Data Engineering

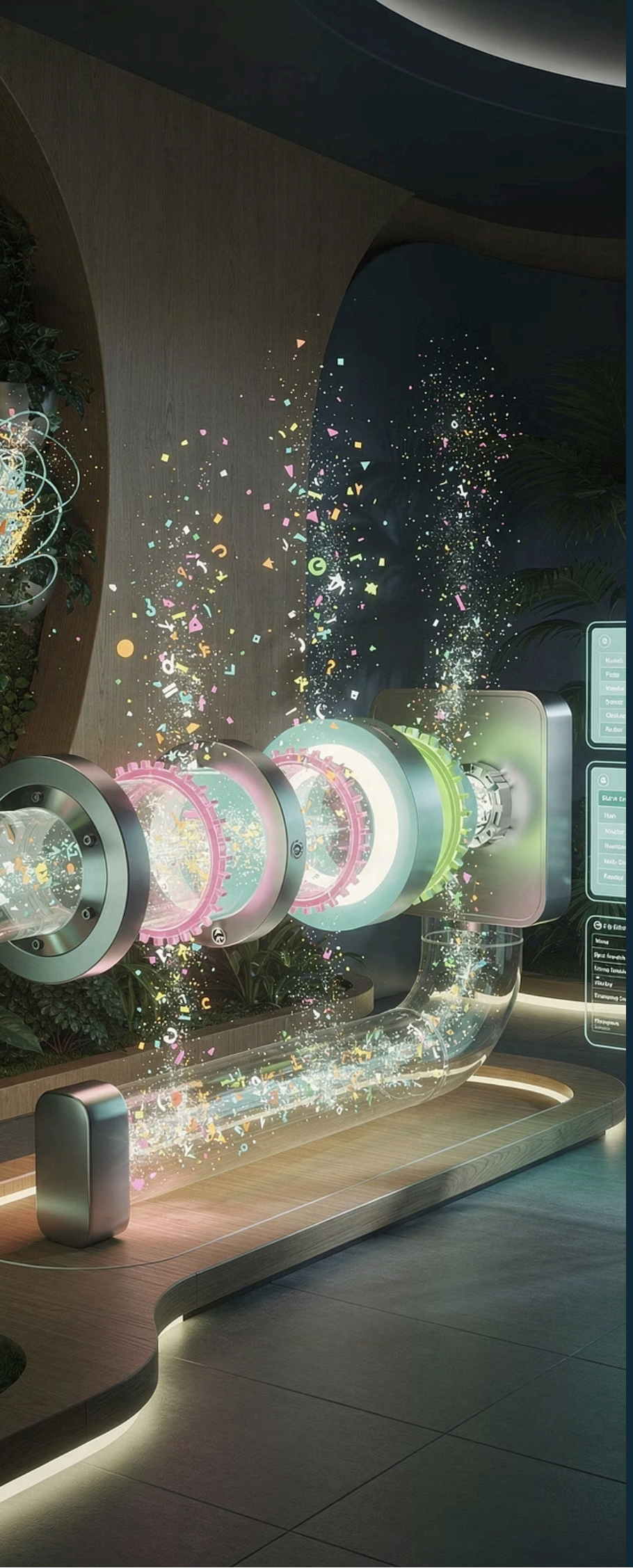
Clean + reliable data for everyone downstream.

🧪 Statistical Modelling & Experimentation

Separating truth from noise with rigorous methods.

🤖 Machine Learning & AI

Automated predictions and real-time decisions at scale.



⚙️ Pillar 1 — Data Engineering

The "Data Plumber" 🔧 — making raw data clean, usable, and trusted.

🎯 What They Do

Data Engineers build and maintain the pipelines that move raw data from source systems into clean, reliable tables that analysts and scientists can actually use.

Key point: Great plumbing saves everyone downstream from headaches. Bad data = wrong decisions.

🔁 Build Pipelines

Automate the flow of data from databases, APIs, and files into a central warehouse.

🧹 Clean & Standardise

Fix formats, handle missing values, remove duplicates, enforce a single source of truth.

✅ Ensure Reliability

Set up data quality checks and alerts so broken data never reaches the team silently.

📋 At a Glance

Goal	Move raw data → tidy tables everyone can trust
Tasks	Pipelines, format cleaning, missing value handling, deduplication
Deliverables	Clean tables, automated pipelines, data quality dashboards
Real Example	A grocery chain uploads receipts nightly → removes duplicates → marks returns → analysts see clean daily sales by 6am

🧪 Pillar 2 — Statistical Modelling & Experimentation

The "Pattern Detective" 🕵️ — separating real results from lucky flukes.

🎯 What They Do

Statistical Modellers and Experimenters use rigorous methods to find patterns in data and test whether changes actually work — or just got lucky.

Key point: Without statistics, you might ship a change that only worked by chance — and hurt users later.

📊 Find Patterns

Use regression, clustering, and trend analysis to understand what's driving outcomes.

🧪 Run Experiments

Design A/B tests with proper control groups to measure the true impact of changes.

📏 Measure Confidence

Use confidence intervals and p-values to know when a result is real vs. noise.

📋 At a Glance

Goal	Spot patterns + measure if a change truly works
Tasks	A/B tests, trend analysis, confidence intervals, explaining results simply
Deliverables	"Version B increased clicks by ~7% (95% confidence)" — clear, honest reporting
Real Example	An e-commerce site tests a green vs. blue "Buy" button on 50,000 users → finds green converts 4.2% better → ships it globally

🍎Tip— "If you flip a coin 10 times and get 7 heads, does that prove the coin is biased?" → Introduces the idea of sample size and statistical significance in a memorable way.

🤖 Pillar 3 — Machine Learning & AI

"The 'Prediction Engine' 🚀 — learning from data to make real-time decisions at scale."

🎯 What They Do

Machine Learning engineers train models on historical data so the system can automatically predict outcomes, classify inputs, or personalise experiences — without being explicitly programmed for every case.

Key point: ML moves insights from "monthly reports" to "real-time action."

🧠 Train Models

Feed labelled examples to algorithms so they learn patterns automatically.

✅ Validate & Test

Check model accuracy on unseen data to ensure it generalises, not just memorises.

🔄 Monitor & Retrain

Track model performance over time and retrain when real-world patterns shift (model drift).

📋 At a Glance

Goal	Predict or classify new data using patterns learned from examples
Tasks	Train model, validate on held-out data, deploy via API, monitor drift
Deliverables	A live model/API that answers "Will this user churn?" in milliseconds
Real Example	A music app learns your skip/like history → predicts songs you'll enjoy → auto-queues them before you ask

🍎 Tip: Ask — "What happens if a model is trained only on data from one city, then deployed globally?" → Introduces the concept of training data bias and why diverse, representative data matters.

🏙️ Mini-Case — All 3 Pillars Together

Scenario: A city wants to predict water pipe bursts before they happen ❄️

⚙️ Data Engineering

Collects pipe age records, historical repair logs, pressure sensor readings, and weather data. Cleans and merges them into one unified, trusted table updated nightly.

Output: A single reliable dataset every team can use.

🧪 Statistical Modelling

Analyses risk factors across thousands of pipes. Finds that pipes over 40 years old burst 3× more often, and that sub-zero weeks increase risk by 60%.

Output: A ranked list of the highest-risk pipes by postcode.

🤖 Machine Learning

Trains a model to score every pipe daily based on age, pressure, weather forecast, and repair history. Flags pipes above a risk threshold for pre-emptive inspection.

Output: A live risk score per pipe, updated every morning.

✅ Combined Result: Pipe burst incidents reduced by 34%. Repair costs down 22%. Zero emergency floods in the pilot district. — All three pillars working in concert. Remove one and the system fails.

Section 4 — Who Does What? The 5 Data Roles

You're focusing on **Data Scientist** in this course — but understanding the full ecosystem is essential to working effectively in any data team.

Data Engineer

Builds pipelines and clean, trusted data. **Win:** Everyone sees the same numbers.

Data Scientist ⭐

Frames problems, explores, experiments, models, explains. **Win:** "Now I know what to do."

ML Engineer

Deploys models as fast, reliable APIs. **Win:** API runs at scale without failing.

Data Analyst

Dashboards and "what happened?" reporting. **Win:** PM spots trend early.

MLOps Engineer

Monitors, retrains, and sets drift alerts. **Win:** Model stays accurate over time.

🏃 Pass-the-Baton: News Recommendations

A website launches a "recommended articles" feature. Watch how each role hands off to the next.

Data Engineer

Builds and maintains pipelines

ML Engineer

Deploys model as API

Data Scientist

Trains and validates models

MLOps Engineer

Monitors and retrains models

Each role has a clear responsibility. No single person does everything — the magic is in the handoff.

⚙️ Step 1 & 🔬 Step 2 — Engineer & Scientist

A news website launches a 'recommended articles' feature. The first two roles lay the foundation.

⚙️ Data Engineer

Ingest — Collects raw article click events from the website in real time across all users and devices.

Clean — Removes bot traffic, fixes broken timestamps, deduplicates repeated sessions.

Structure — Organises data into a reliable, version-controlled warehouse layer. Every downstream team gets the same clean, consistent view of user behaviour.

🤝 **Handoff → Clean, trusted data table passed to the Data Scientist.**

🔬 Data Scientist

Explore — Analyses reading patterns: night owls vs morning readers, topic clusters, session length by category.

Model — Trains a collaborative filtering recommendation model on the clean click data.

Validate — Designs an A/B test to measure engagement lift. Presents findings and model confidence to stakeholders.

🤝 **Handoff → Trained model + performance report passed to the ML Engineer.**

🚀 Step 3 & 🔄 Step 4 — ML Engineer & MLOps Engineer

The model is built — now it needs to be deployed, monitored, and kept accurate over time.

🚀 ML Engineer

Wrap — Packages the trained model into a REST API endpoint accessible by the website in real time.

Serve — Returns "top 5 recommended articles for user X" in under 50ms per request.

Log — Records every API call — latency, errors, and edge cases — for reliability and debugging.

🤝 **Handoff → Live API in production passed to the MLOps Engineer.**

🔄 MLOps Engineer

Retrain — Schedules automatic model retraining every week using the latest click data from the pipeline.

Alert — Sets a threshold: if click-through rate drops below 2.1% baseline → triggers an investigation automatically.

Rollback — Maintains version control on all models — if a new version underperforms, instantly reverts to the previous one.

🤝 **Handoff → Stable, monitored model passed to the Data Analyst.**

📊 Step 5 — Data Analyst & Outcome

The final handoff — turning model performance into business insight and closing the feedback loop.

📊 Data Analyst

Dashboard — Builds a real-time editor dashboard tracking clicks by hour, trending topics, and top recommended articles.

Insight — Identifies early wins: "**Tech articles recommended at 8pm get 3×** more clicks than morning slots."

Loop — Reports findings back to the Data Scientist, triggering the next round of model improvement and retraining.

🤝 **Closes the loop: Analyst insights → back to Data Scientist for the next iteration.**

✅ Final Outcome

📈 **+31%** — Increase in article click-through rate after 4 weeks of recommendations

⏱️ **2 min** — Average reading time per recommended article (up from 45 seconds)

🔁 **5 Roles** — Each contributed one piece. Together they shipped a complete production data product.

💡 **Key Lesson** — No single role could have done this alone. Data science is a team sport.

Key Takeaways

1 The Loop is Everything

Ask → **Collect** → **Understand** → **Act** is the universal framework. Every data product — from spam filters to Amazon warehouses — follows this loop.

2 Three Pillars, One Stool

Data Engineering, Statistical Modelling, and ML & AI each play a critical role. Weaken one and the whole system wobbles.

3 Roles Collaborate, Not Compete

Data Scientists sit at the center — but they rely on Engineers, Analysts, and MLOps to bring insights to life and keep them alive.