ELEC3544 · Fall 2026

Data Science with Foundation Models

Yang Liu

The University of Hong Kong



Lecture 4

Data Science Applications and Data Wrangling

From real-world impact to professional data preparation

Today's Learning Objectives

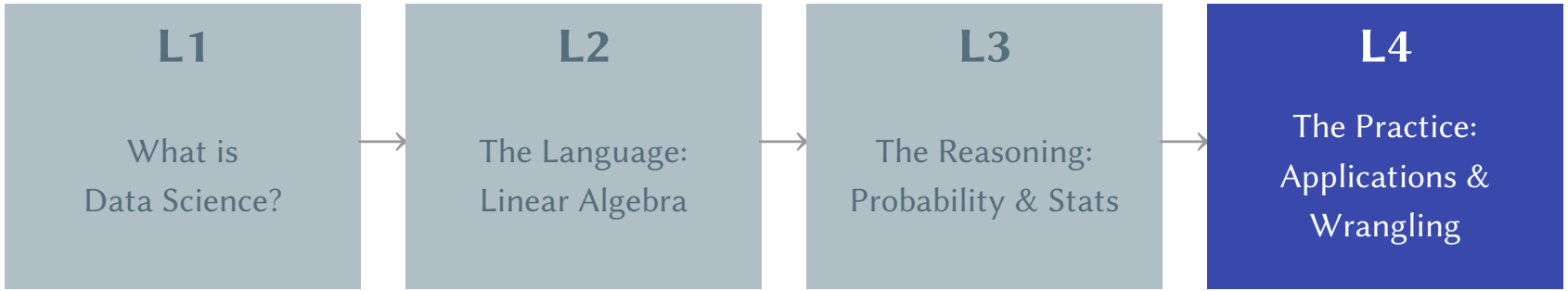
Part 1: Why and What — Applications & Ideation

1. Describe data science applications across multiple domains, identifying the common workflow pattern.
2. Formulate a project idea using structured CTA-assisted ideation, evaluating feasibility.

Part 2: How — Professional Data Wrangling

3. Conduct systematic data quality assessment, producing a structured quality report.
4. Choose and justify cleaning strategies based on missingness mechanisms and domain context.
5. Conduct discovery-driven EDA and prepare data through encoding and reproducible pipelines.
6. Direct CTA-assisted wrangling with strategic decision-making and verification.

Where We Are: Module 1 Journey



L4 bridges foundations (Module 1) to core methods (Module 2)

Module 2 ahead: Regression (L5) ⟶ Classification (L6) ⟶ Unsupervised Learning (L7)

What You've Built So Far

Your toolkit for today — concepts we'll use throughout this lecture:

From L2: Linear Algebra

Data as matrices and vectors

Variance, covariance,

PCA intuition

Dimensionality as feature count

From L3: Probability & Stats

Summary statistics and

distributions,

"Always visualize" principle,

Conditional probability,

Bayes' rule

From L1: Course Framework

VERIFY framework for AI outputs

Human judgment + AI capability

Data science pipeline overview

Key Insight: Today you'll see these tools in action: matrices become real datasets, distributions reveal data quality issues, and the VERIFY framework guides every CTA interaction.

Today's Structure: Two Parts

Part 1: "Why and What"

1A: Application Showcase

Four domains showing data science impact

1B: Project Ideation

Five-step CTA brainstorming process

Part 2: "How"

2A: Data Quality Assessment

Four-check protocol + CTA profiling

2B: Missingness & Cleaning

MCAR/MAR/MNAR framework

2C: EDA & Pipelines

Discovery cycle ⟶ reproducible pipeline

🤖 CTA Role Transition Today

Brainstorming Partner (generating and evaluating ideas) ⟶ Implementation Assistant (writing code, generating reports)

1. Interest ⟶ Project Idea 2. Automated Profiling 3. Cleaning Consequences 4. Discovery & Pipeline

PART 1A

Data Science Applications

Seeing the impact across domains

Data Science is Everywhere

What do disease prediction, fraud detection, climate forecasting, and smart traffic management have in common?



Healthcare

Prediction & Discovery



Finance

Risk & Pattern Detection



Climate

Monitoring & Prediction



Smart Cities

Urban Intelligence

🤔 Track these questions: What data? What techniques? What could go wrong?

Healthcare: Prediction and Discovery

Disease Prediction

Patient features (age, lab results, history) ⟶ predict diabetes risk

Classification problem — connects to L6

High stakes: wrong predictions have real consequences

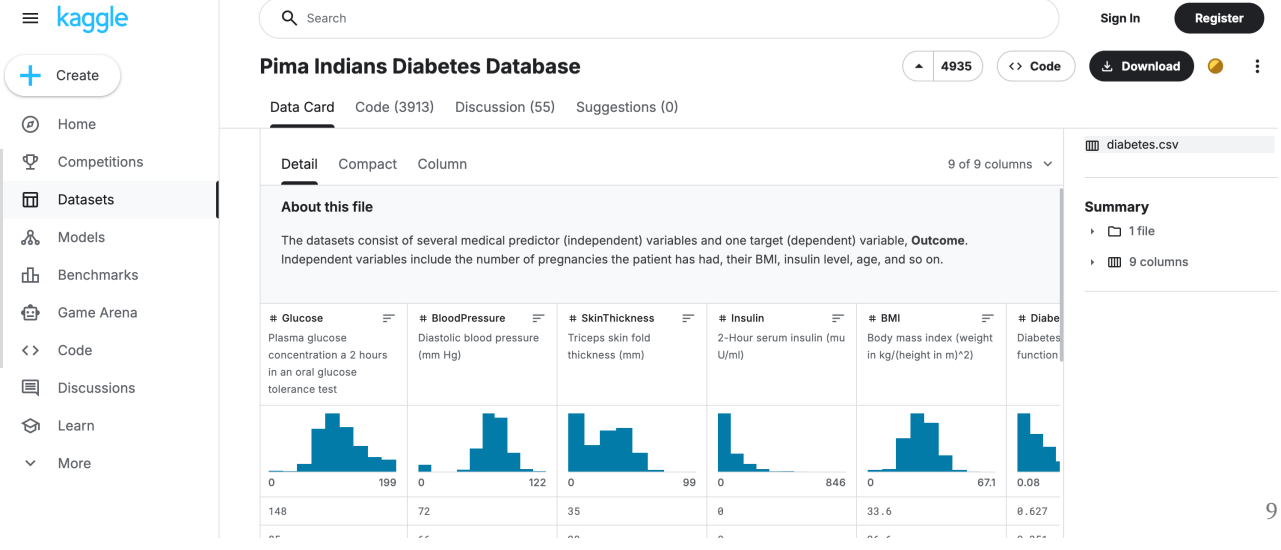
Medical Imaging

Deep learning detecting pneumonia from X-rays

Connects to L9 (Deep Learning)

Enabled by large annotated datasets

The data:
structured patient records — each row is a patient; each column is a measurement.



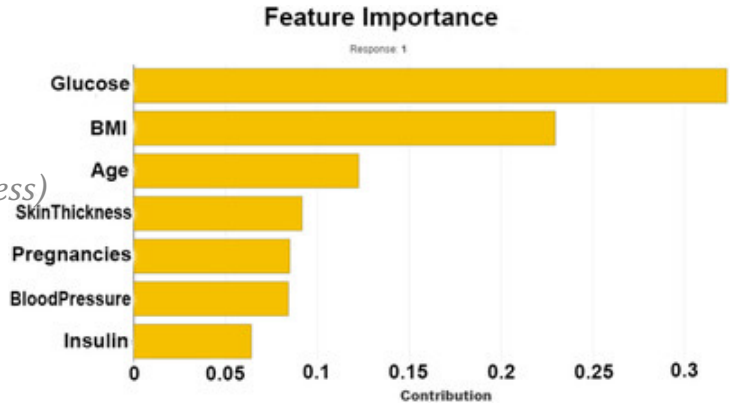
Healthcare: The Data Science Perspective

The Pipeline

Data: Electronic health records, lab results, medical imaging

Key Challenges

- Missing data — patients skip tests *(connects to Part 2: missingness)*
- Class imbalance — most patients are healthy *(previews L6)*
- Privacy regulations — HIPAA, patient consent
- High stakes — false negative in cancer screening



Impact

Early detection programs reducing disease progression — data-driven screening saves lives.

Finance: Risk and Pattern Detection

Fraud Detection

Identifying fraudulent transactions among millions

Extreme class imbalance (~0.1% fraud)

"Ever gotten a text from your bank asking 'was this you?'"

Credit Scoring

Predicting loan defaults from applicant profiles

Regression and classification (L5–L6)

Fairness concerns: who gets denied?

Sample Transaction Log

Timestamp	Amount	Merchant	Location	IsFraud
2024-03-15 09:23:41	$42.50	Starbucks Coffee	Hong Kong, HK	0
2024-03-15 09:24:03	$8,750.00	ElectroMart Online	Lagos, NG	1
2024-03-15 09:24:18	$15.99	Netflix	Hong Kong, HK	0
2024-03-15 09:25:02	$3,200.00	LuxWatch Ltd	Moscow, RU	1
2024-03-15 09:25:44	$67.30	PARKnSHOP	Kowloon, HK	0
2024-03-15 09:26:11	$23.80	MTR Corporation	Central, HK	0

The data: transaction logs — millions of rows, real-time streaming, each row is one purchase.

Finance: The Data Science Perspective

The Pipeline

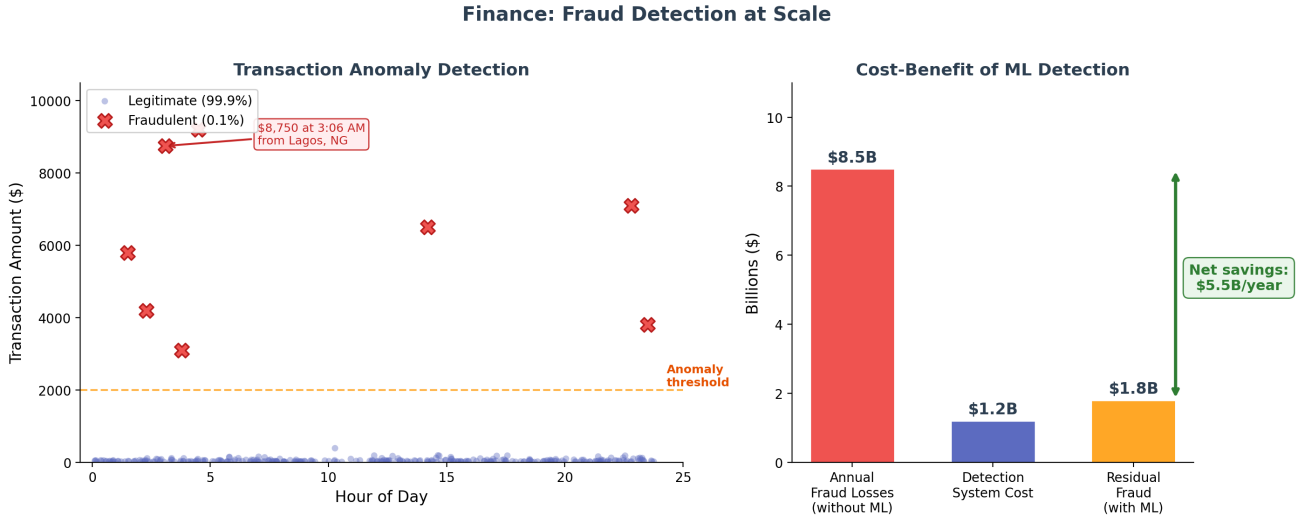
Data: Transaction logs (millions of rows, real-time), market data, customer profiles

Key Challenges

- Extreme imbalance — ~0.1% of transactions are fraudulent
- Adversarial behavior — fraudsters adapt to detection systems
- Real-time requirements — decisions in milliseconds
- Fairness concerns — who gets denied credit, and why?

Impact

Cost savings from fraud detection at scale — financial institutions save billions annually through automated pattern detection.



Climate: Monitoring and Prediction

Climate Monitoring & Prediction

Analyzing temperature records, forecasting extreme weather

Regression problem — connects to L5

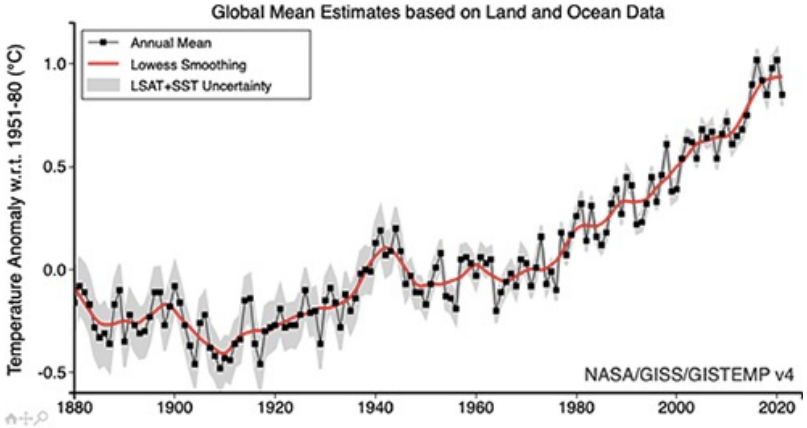
Decades of spatial and temporal data

Energy Optimization

Predicting solar/wind output for grid management

Regression (L5) for continuous forecasting

HK context: typhoon prediction, air quality



Weather Station Temperature Readings Across Hong Kong

Timestamp	HK Observatory (°C)	Tai Mo Shan (°C)	Waglan Island (°C)	Sha Tin (°C)	Cheung Chau (°C)
2024-09-07 00:00	27.1	21.8	26.9	26.5	27.0
2024-09-07 06:00	27.3	22.1	27.0	26.8	27.2
2024-09-07 12:00	**31.8**	25.4	30.5	**31.2**	30.8
2024-09-07 18:00	29.4	23.7	28.8	29.0	28.6
2024-09-08 00:00	27.5	22.0	27.2	26.9	27.3
2024-09-08 06:00	26.8	21.5	26.5	26.2	26.7
2024-09-08 12:00	**32.4**	26.1	**31.2**	**31.8**	**31.0**

Each row = one timestamp • Each column = one location • Repeated every 6 hours across stations

The data: time series from sensor networks — each row is a timestamp; columns are measurements across locations.

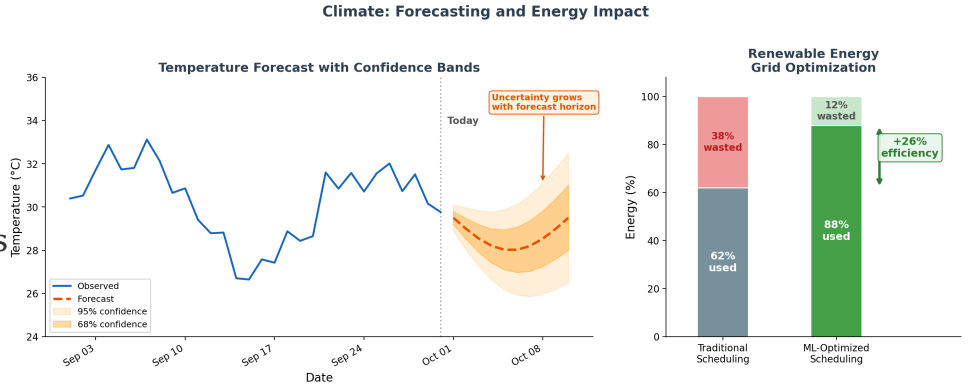
Climate: The Data Science Perspective

The Pipeline

Data: Satellite imagery, sensor networks, weather stations (decades of time series)

Key Challenges

- Temporal dependencies — today's weather depends on yesterday's
- Spatial correlations — nearby stations are correlated
- Massive scale — billions of data points across decades
- Weather prediction vs. climate projection — different timescales, different methods



Impact

Data-driven energy grid optimization — matching renewable supply to demand reduces waste and carbon emissions.

Smart Cities: Urban Intelligence

Transportation Optimization

Ride-sharing demand prediction, traffic flow analysis

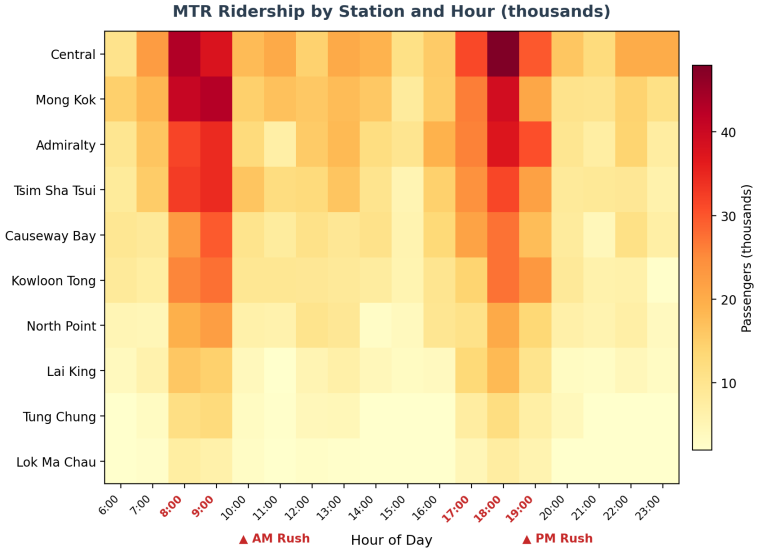
HK: Octopus card patterns, MTR crowding prediction
Clustering + regression methods

Urban Sensing

Air quality monitoring, energy consumption patterns

IoT sensor networks generating real-time data
Embeddings for location representation

Smart Cities: MTR Transit Data (Hong Kong)



Sample Transit Records (15-second intervals)

Timestamp	Station	Entries	Exits
08:15:03	Central	1,247	892
08:15:03	Mong Kok	1,105	756
08:15:03	Admiralty	983	1,124
08:15:18	Central	1,312	901
08:15:18	Mong Kok	1,089	812
08:15:18	Admiralty	1,021	1,198
08:15:33	Central	1,285	878

Each row = one reading • Sub-minute granularity
~5.7M daily Octopus taps across the network

The data: IoT sensors, transit records, mobile traces — each row is a reading, often at sub-minute granularity.

Smart Cities: The Data Science Perspective

The Pipeline

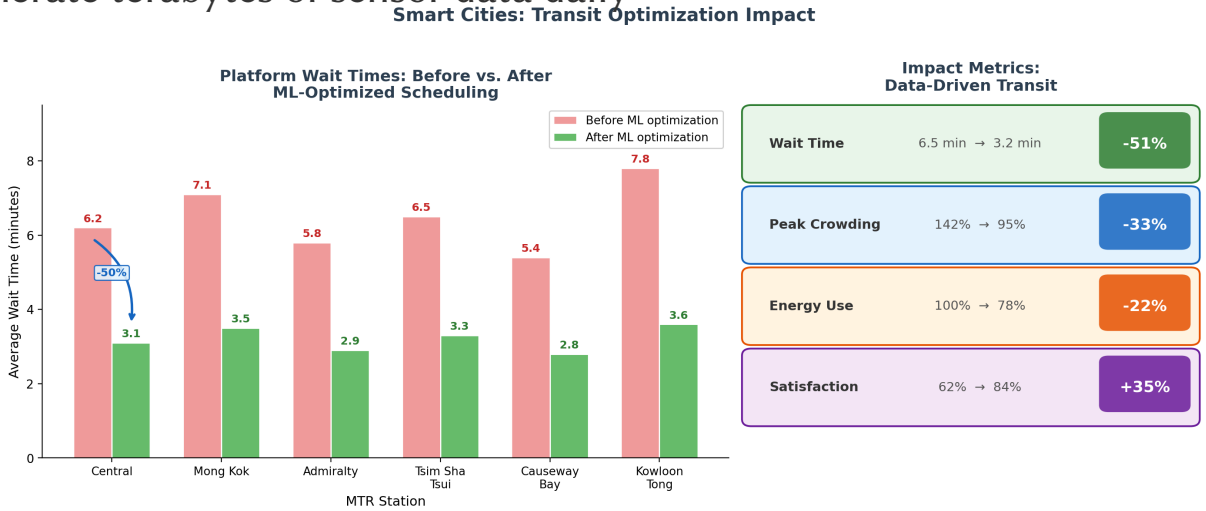
Data: IoT sensors, transit records, mobile traces, open government data

Key Challenges

- Heterogeneous sources — combining GPS, sensors, surveys, and social data
- Privacy — location data reveals personal patterns
- Real-time processing — traffic optimization requires instant decisions
- Scale — entire cities generate terabytes of sensor data daily

Impact

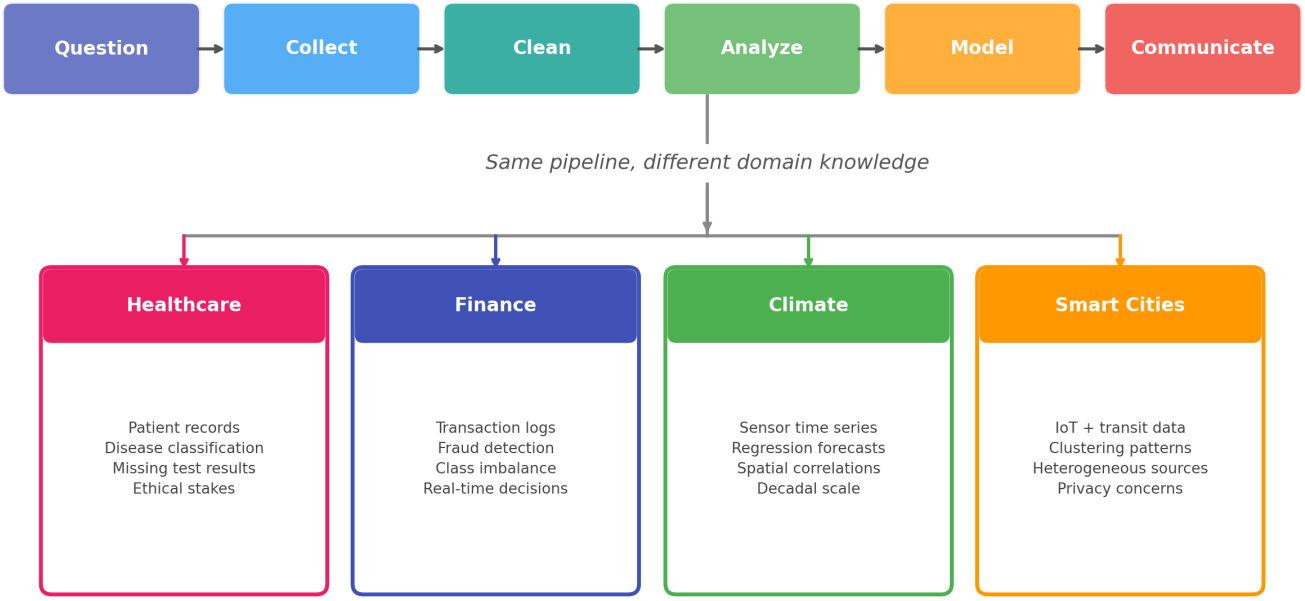
Transit optimization reducing commute time and crowding — data-driven scheduling improves quality of life for millions.



The Common Thread

Key Insight

All four domains follow the same pipeline: Question → Collect → Clean → Analyze → Model → Communicate. What changes is the domain knowledge. The pipeline is universal; the judgment is domain-specific.



The pipeline is universal. The judgment is domain-specific.

Connecting Domains to Course Techniques

Domain	Primary Techniques	Key Challenges	Lecture(s)
🏥 **Healthcare**	Classification + Metrics	Missing data, class imbalance	L4, L6
💰 **Finance**	Regression + Classification	Extreme imbalance, fairness	L5, L6
🌍 **Climate**	Regression + Visualization	Temporal dependencies, scale	L4, L5
🏙️ **Smart Cities**	Clustering + Embeddings	Heterogeneous data, privacy	L7

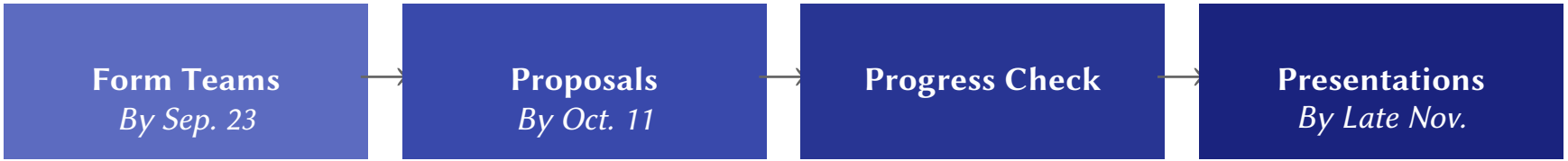
The methods in L5–L7 are the tools practitioners use across all these domains.

Your Turn: The Course Project

This is where you apply everything from the course.

- Team of 3 students
- Real dataset (not from class) — you find it, you own it
- Full data science pipeline: question → data → wrangling → modeling → communication

Project Timeline



Assessment: 30% of final grade • Demonstrates end-to-end data science capability

Today's Part 1B: learn the ideation process to find YOUR project idea.

What Makes a Good Project?

1. **Clear, answerable question**
   Not "understand social media" but "predict post engagement from timing and content features"

2. **Real, accessible data (>1,000 rows; labelled / paired data)**
   Publicly available or collectible within the project timeline

3. **Appropriate scope (6 weeks)**
   Manageable for a team of 2–3 — not a PhD thesis, not a homework problem

4. **Personal interest**
   You'll spend weeks on this — pick something you genuinely want to explore

5. **Full pipeline opportunity**
   Involves data wrangling, EDA, modeling, and interpretation — not just one step

✓ **Good:** "Predicting customer churn for a streaming service" ✗ **Too broad:** "Understanding social media"

Where to Find Data

📊 Kaggle Datasets

Thousands of curated datasets with community notebooks

🎓 UCI ML Repository

Classic benchmark datasets used in academic research

🔍 Google Dataset Search

Search engine specifically for datasets

🇭🇰 HK Open Data

data.gov.hk — local datasets for Hong Kong projects

🌐 World Bank / WHO / NOAA

Authoritative global data on economics, health, climate

📂 GitHub awesome-public-datasets

Community-curated list organized by domain

> **⚠️ Always Verify First**
> Always verify the dataset actually exists and has the features you need before committing to a project idea.

Tip: Start browsing datasets this week. Bring 2–3 candidates to your team discussion.

The Five-Step CTA Ideation Process

From first principle

1 — Start with Your Interest

Not data — YOUR curiosity matters the most

→

2 — Generate Questions

CTA brainstorms options

→

3 — Evaluate Feasibility

Data, scope, techniques

→

4 — Refine & Narrow

Iterate toward clarity

→

5 — Produce Project Sketch

One-paragraph proposal

> 💡 **Key Insight**
>
> The CTA generates options. YOU make every decision — which question, which dataset, which scope.

🤖 CTA Demo: From Interest to Project Idea

> **What We'll Explore**
> Watch how we take a vague interest and turn it into a concrete, feasible project idea using the five-step process.
> Let’s ask for a topic from the audience — it doesn't need to relate to the four domains we discussed.

Key Questions:

- How do you write a structured prompt with constraints — not just "give me ideas"?
- How do you evaluate CTA suggestions against your five criteria?
- How do you verify that a CTA-suggested dataset actually exists?
- How do you make the final idea yours — not just the CTA's output?

CTA Role: Brainstorming Partner **Dataset:** Audience-chosen topic

Pattern: Iterative Refinement (Creative)

CTA Demo 1: Key Moments

1. Structured Prompt with Constraints

Not "give me ideas" — specifying level, data requirements, techniques, timeline, and team size in the prompt.

2. Evaluating, Not Accepting

Assessing each CTA suggestion against the five criteria. "This one is interesting but where would I get the data?"

3. VERIFY: Validate

Checking if the CTA-suggested dataset actually exists. "Does this dataset have the claimed features?"

4. Making It Yours

Articulating WHY you chose this question. "Can I explain this project without the CTA?"

The CTA generated ideas, suggested datasets, and wrote a polished proposal — but every decision was the instructor's.

VERIFY in Action: Project Ideation

✓ VERIFY: Applying the Framework to Project Ideation

V	**Validate**	Does the dataset actually exist with the claimed features?
E	**Execute**	Can you download and load it? Try it before committing.
R	**Reason**	Does the approach logically address the research question?
I	**Inspect**	What could go wrong — data too small, features irrelevant, scope too large?
F	**Fix**	Iterate on scope and approach. Narrow or pivot as needed.
Y	**Yours**	Can you explain this project without the CTA? Is this YOUR idea?

Highlighted: V (Validate) and Y (Yours) — the two steps most critical for project ideation.

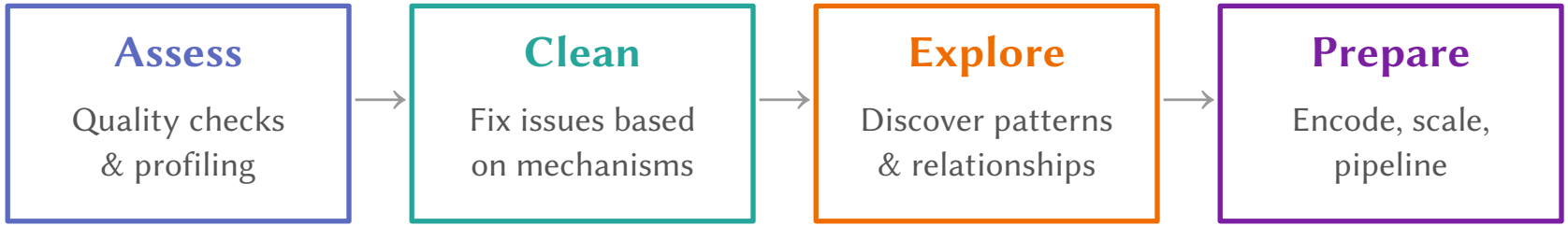
PART 2

Professional Data Wrangling

From raw data to model-ready features

The Professional Data Wrangling Workflow

The 80/20 Reality: Data scientists spend approximately 80% (or even more!) of their time on data preparation — cleaning, transforming, and exploring data before any modeling begins.



Running Example: Telco Customer Churn dataset (~7,043 customers, 21 columns, binary target). We'll use this dataset throughout Part 2.

Today: we walk through each step with real data and real CTA collaboration.

Meet the Telco Dataset

~7,043 customers

21 columns

Binary target: **Churn**

Demographics

`gender, SeniorCitizen, Partner, Dependents`

Account Information

`tenure, Contract, PaymentMethod, MonthlyCharges, TotalCharges`

Services

`PhoneService, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies`

Business Question: Predict which customers will churn (leave within the last month).

https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Step 1: Systematic Data Quality Assessment

1

Structural

Dimensions, data types, memory usage

2

Completeness

Missing values — where, how many, patterns?

3

Validity

Ranges, type mismatches, duplicates

4

Distributional

Summary stats, distribution shapes

Output: A Data Quality Report — your first professional deliverable

💡 **Key Insight:** Systematic assessment before cleaning. Don't fix what you haven't diagnosed — the four-check protocol ensures you understand your data before changing it.

Structural Assessment

Python / pandas

```
# Dimensions
df.shape
# Column types (most important!)
df.dtypes
# Full overview
df.info()
# Memory usage
df.memory_usage(deep=True)
```

What to Look For

Most important diagnostic:
`.dtypes`

Type mismatches are common and informative.

A numerical column stored as `object` means hidden quality issues.

Example: Telco Dataset
If `df.dtypes` reveals TotalCharges as `'object'` — not `'float64'` as expected. This signals a hidden data quality issue we need to investigate.

Completeness Assessment

```
# Count missing per column
df.isnull().sum()

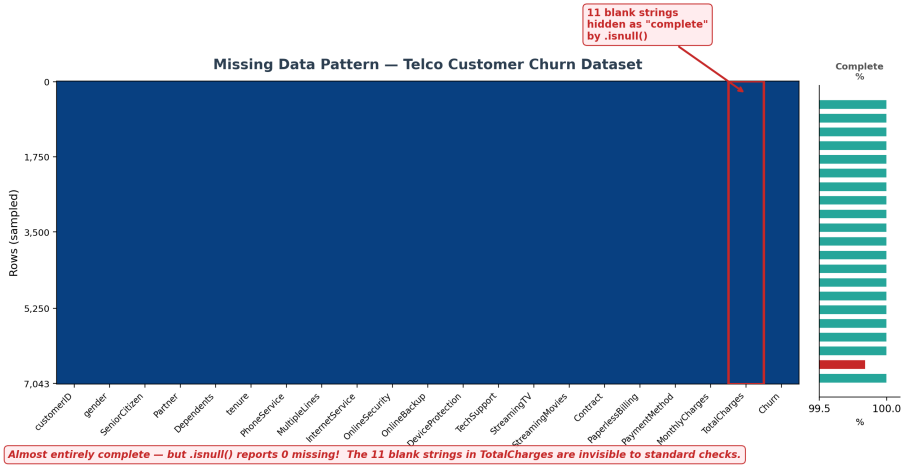
# Proportion missing
df.isnull().mean()
```

Critical Question: WHY is data missing? The answer determines your cleaning strategy.

⚠ **Watch Out:** "Missing" data isn't always `NaN` — can be blank strings, "N/A", -999, or other sentinel values. Always check beyond `.isnull()`.

Missingness Rules of Thumb

- **< 5%** Manageable — most strategies work
- **5–30%** Needs careful strategy choice
- **> 30%** Consider dropping the column



Validity Assessment

Out-of-Range Values

Negative ages? Charges above $10,000?
Domain knowledge defines valid ranges.

Duplicate Rows

`df.duplicated().sum()`
Investigate before dropping — are they true duplicates or repeat measurements?

Inconsistent Categories

"Female" vs. "female" vs. "F"
Use `.value_counts()` to find variants.

Cross-Field Inconsistencies

Tenure = 0 but TotalCharges = $5,000?
Relationships between columns reveal errors.

CTA can identify these efficiently — but interpreting whether suspicious values are errors requires domain knowledge.

Distributional Assessment

Beyond `df.describe()` — visualization reveals what statistics hide. This is how you OWN the data.

Histograms

Numerical shape — skewed? bimodal? uniform?

Box Plots

Outliers and quartile spread at a glance

Count Plots

Categorical balance — are classes roughly even?

KDE

Smooth density estimate — reveals subtle patterns

Look for: unexpected shapes • extreme skewness • suspicious uniformity • multimodal distributions

> 💡 **Callback to L3:** Remember Anscombe's quartet — statistics alone aren't enough. Always visualize. Four datasets with identical means, variances, and correlations looked completely different when plotted.

🤖 CTA Demo: Automated Profiling

with Human Interpretation

> **What We'll Explore**
>
> Let's have the CTA generate a complete data quality report for the Telco dataset. Think of how to write a structured request — and then check what the CTA gets right, and what only a human can catch.

Key Questions:

- How do you write a structured prompt with four explicit sections?
- How does the quality of your request affect the quality of the output?
- What does the CTA get right — and what only a human can catch?

CTA Role: Implementation Assistant Dataset: Telco Customer Churn

Pattern: Structured Requests + Verification Through Computation

CTA Demo 2: The Structured Request

The Structured Prompt

"I have a dataset about teleco
customer churn with ~7,000 rows
and 21 columns. Please produce
a data quality assessment covering ... :

(1) Structural overview
(2) Completeness
(3) Validity
(4) Distributional summary

Format as a structured report with
a prioritized issues list at the end."

Compare to:

"Is my data clean?"

Vague input ⟶ vague output

💡 Lesson

Quality of AI output depends on quality of your request.

Four explicit sections matched to the four-check protocol, with specified output format.

Notice: the prompt includes context (dataset description), structure (four sections), and format (report with prioritized issues).

CTA Demo 2: What the Human Catches

Finding: `SeniorCitizen`

CTA reports:
`int64`, values `[0, 1]`, mean 0.16

Human recognizes:
This is a binary categorical variable, not continuous. Its 0/1 values already provide valid indicator encoding.

Finding: `TotalCharges`

CTA flags:
`dtype` is `object` (string), not `float`

Human investigates:
Blank strings (not `NaN`) for zero-tenure customers — new customers who haven't been billed yet. Business explanation!

💡 **Key Insight:** The CTA computes accurately. The human interprets meaningfully. The CTA can't know SeniorCitizen is categorical without business context, and can't explain why TotalCharges has blanks without domain understanding.

Two takeaways: (1) structured requests produce comprehensive output, and (2) human interpretation catches what the CTA misses.

VERIFY in Action: Data Quality Assessment

✓ VERIFY: Applied to Data Quality Assessment

V	**Validate**	Are type classifications correct? Is `SeniorCitizen` really numerical?
E	**Execute**	Does the profiling code run on your dataset without errors?
R	**Reason**	Do the CTA's findings make sense given what we know about the business?
I	**Inspect**	What edge cases might the automated profile miss? (Sentinel values, type mismatches)
F	**Fix**	Iterate: ask follow-up questions when something looks suspicious.
Y	**Yours**	Can you explain each finding in your own words?

Highlighted: Validate and Inspect — the two steps most critical for data quality assessment.

The Data Quality Report as Professional Deliverable

A skill you'll use in your project and your career.

1	**Data Overview**	Source, dimensions, feature descriptions, collection context
2	**Findings by Check Type**	Structural, completeness, validity, distributional — organized and specific
3	**Prioritized Issues**	Ranked by impact on analysis — what must be fixed vs. what can wait
4	**Recommended Actions**	Specific, justified cleaning strategies for each issue identified

Now that we know what's wrong, how do we fix it? That depends on why the data is problematic.

Why is Data Missing? Three Mechanisms

The mechanism determines the valid cleaning strategy — this cannot be determined from data alone.

MCAR

Missing Completely at Random

Missingness unrelated to any variable — observed or unobserved.

Example: Lab tech drops test tubes — loss is purely random.

MAR

Missing at Random

Given observed variables, missingness is unrelated to missing values.

Example: Older patients skip fitness test — depends on age (observed).

MNAR

Missing Not at Random

Missingness depends on missing values, even given observed variables.

Example: High earners don't report income — depends on income (unobserved).

Determining the mechanism requires reasoning about the data-generating process — it cannot be determined from data alone.

Why is Data Missing? Three Mechanisms

Easiest to handle ——— *Hardest to handle*

MCAR

Missing Completely at Random

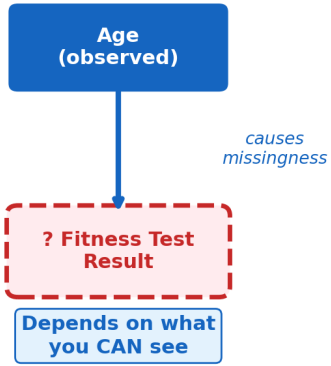


Pure accident

Lab tech drops test tubes
— loss unrelated to any variable

MAR

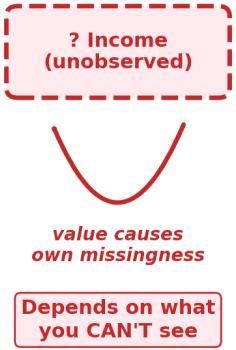
Missing at Random



Older patients skip fitness test
— depends on age (which you have)

MNAR

Missing Not at Random



Hardest case

High earners don't report income
— depends on income (which is missing)

Why Mechanisms Matter for Cleaning

The mechanism determines which imputation strategy is valid.

MCAR	**Deletion can be unbiased**	Listwise deletion loses data. Mean/median imputation can distort variability.
MAR	**Conditional imputation appropriate**	Use related observed variables and a suitable imputation model.
MNAR	**No simple fix**	Must model the missingness process itself, or acknowledge the limitation of the data.

> 💡 **Key Insight**
>
> The CTA will implement whatever strategy you request. YOUR job is determining which strategy is valid — and that requires understanding the mechanism.

"The mechanism is the judgment call. The implementation is just code."

Cleaning Strategy Toolkit

From simple to sophisticated — choose based on the missingness mechanism.

Strategy	Description	When to Use / Notes
Deletion	Listwise or pairwise removal of incomplete rows	Generally safest under MCAR; dataset large enough to absorb loss
Constant	Fill with a fixed value (e.g., 0, 'Unknown')	When 'missing' has natural meaning (e.g., no internet service)
Statistical	Mean, median, or mode imputation	Simple but distorts variance and distribution shape
Conditional	Impute using related observed variables	Appropriate under MAR with a suitable imputation model
Model-based	KNN, iterative imputer, or regression-based	More sophisticated; handles complex MAR patterns

Simple ⟶ Sophisticated

🤖 CTA Demo: Consequences of Cleaning Choices

What We'll Explore

TotalCharges has missing values for zero-tenure customers. The CTA will happily fill these in however we ask. But which strategy should we use? Let's try three approaches and see what happens to the data.

Key Questions:

1. How do different imputation strategies affect the distribution of `TotalCharges`?
2. Which approach produces artificial patterns that don't match reality?
3. How does the missingness mechanism tell us which strategy is valid?

CTA Demo 3: Three Strategies Compared

> 🤖 **Three imputation strategies, three very different results.**

Mean Imputation

Artificial spike at ~$2,280

Tells the model 11 new customers all spent exactly $2,280 — completely unrealistic for zero-tenure.

❌ **Distorts distribution**

Median Imputation

Spike at median value

Same problem at a different location — median is still unrealistic for brand-new customers.

❌ **Same issue, different value**

Conditional Imputation

Values near $0 for new customers

Imputes by tenure group — zero-tenure customers get values based on similar (1–3 month) customers. No artificial spike.

✓ **Preserves natural relationship**

Visual: overlaid or side-by-side histograms make the consequences immediately visible.

CTA Demo 3: Connecting to Mechanisms

Observation

`TotalCharges` is missing for customers with `tenure` = 0.
We assume missingness depends on tenure alone,
not on the unobserved `TotalCharges` value.

→

Working assumption: MAR

Conditional imputation accounts for tenure.
Its validity also depends on the imputation model
and assumptions for zero-tenure customers.

💡 **Key Insight**

Mean/median imputation introduces values that contradict the data-generating process — new customers didn't spend $2,280. Conditional imputation can use the relationship between tenure and charges. The MCAR/MAR/MNAR framework isn't abstract theory — it guided our strategy and the assumptions to check.

✓ **VERIFY Connection**

Inspect — examining distributional consequences of each strategy. Reason — connecting the chosen strategy to the missingness mechanism.

"Three strategies, three different results. The CTA implemented all three equally well — it doesn't know which is 'right.' That judgment came from you."

Beyond Missing Data: Other Cleaning Tasks

Fixing Type Mismatches

```
df['TotalCharges'] = pd.to_numeric(
    df['TotalCharges'], errors='coerce')
```

Converts string columns to numeric; `'coerce'` turns unparseable values to `NaN` for later handling.

Handling Duplicates

```
df.duplicated().sum()
# Investigate before dropping!
```

Are they true duplicates or repeat measurements? Domain context determines the action.

Standardizing Categories

```
df['col'] = (df['col'].str.strip()
  .str.lower().replace({'f': 'female'}))
```

"Female" vs. "female" vs. "F" — inconsistent labels cause silent errors in analysis.

Handling Outliers

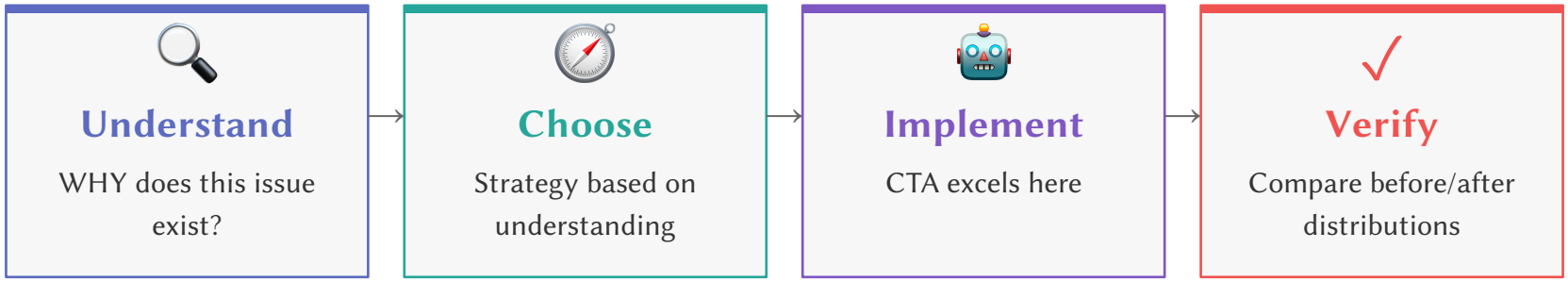
```
# Domain-dependent decision
```

Errors ⟶ fix or remove. Genuine extremes ⟶ keep. Different population ⟶ investigate separately.

CTA can identify these efficiently — but interpreting whether suspicious values are errors requires domain knowledge.

Consequence-Driven Cleaning: A Mindset

The professional approach to data cleaning.



Key Insight

The CTA implements; you decide and verify. Steps 1, 2, and 4 require human judgment. Step 3 is where the CTA accelerates your workflow.

Human: Understand why → Choose strategy → Verify consequences **CTA:** Implement efficiently

"Don't fix what you don't understand. Don't trust what you haven't verified."

Our data is now clean.

Or more precisely — we've made deliberate, justified cleaning choices.

Assessed quality ✓ → Understood mechanisms ✓ → Chose strategies ✓ → Verified consequences ✓

Next: What can we discover?

EDA is where data science gets creative.

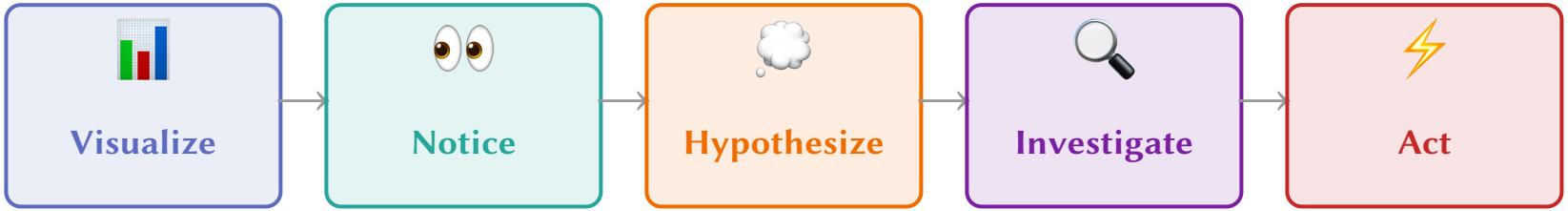
Each visualization provokes a question.

Each question motivates the next visualization.

Part 2C: Discovery-Driven EDA and Pipelines →

Exploratory Data Analysis: The Discovery Cycle

EDA isn't a checklist — it's an iterative cycle of creative discovery.



↻ *Each visualization provokes a question; each question motivates the next visualization.*

Quality Assessment vs. EDA

Quality Assessment

Systematic — checks every column methodically. Goal: find problems.

Exploratory Data Analysis

Creative and question-driven — follows interesting threads. Goal: find patterns and insights.

EDA Toolkit: Visualizations by Purpose

Choose your visualization based on the question you're asking.

📊

What does one variable look like?

Histogram
Count plot
KDE

📈

How do two variables relate?

Scatter plot
Grouped bar
Box plot

🎯

How does the target differ across groups?

Grouped bar chart of target rate

🔥

Patterns across many features?

Correlation heatmap
Pair plots

💡 **Callback to L3:** Remember Anscombe's quartet — four datasets with identical statistics but completely different shapes. Chart selection matters. Always match the visualization to your question.

EDA in Practice: Telco Dataset

First EDA look — three key observations that set up deeper investigation.

Tenure Distribution

Bimodal — many new customers and many long-term customers (loyalty), fewer in the middle.

🤔 *Why two clusters? Does this relate to churn?*

Monthly Charges

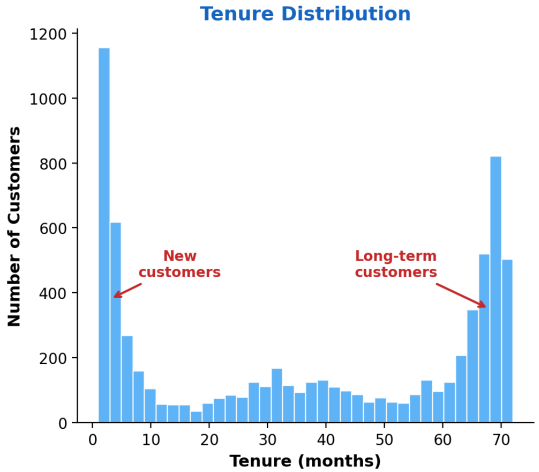
Roughly uniform with concentration at the low end. Wide range ($18–$118).

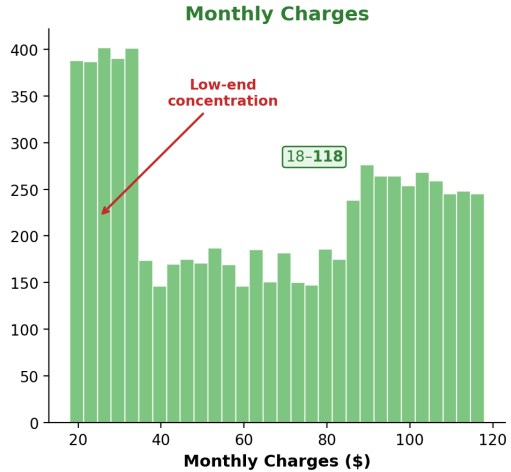
🤔 *Do high-charge customers churn more?*

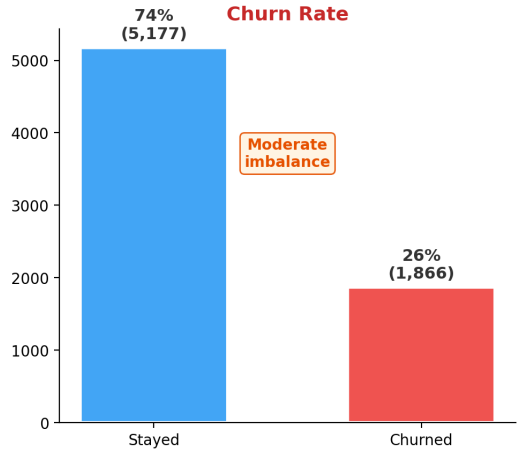
Churn Rate

~27% of customers churned — moderate class imbalance (not extreme).

🤔 *Which customer segments drive this?*







🤖 CTA Demo: Discovery Workflow

What We'll Explore

Let's do live EDA on the Telco dataset. We can start with one visualization, notice something interesting, and follow that thread. Watch how each finding leads to the next question — this is the discovery cycle in action.

Key Questions:

1. How does a targeted visualization request differ from "show me the data"?
2. How do you follow a discovery thread — letting one finding guide the next question?
3. What features emerge as strong predictors of churn?

CTA Role: Implementation Assistant Dataset: Telco Customer Churn

Pattern: Iterative Refinement + Challenging for Completeness

CTA Demo 4: The Discovery Thread

Each finding leads to the next question — the discovery cycle in practice.

1 Scatter: Charges vs. Tenure

"Create a scatter of MonthlyCharges vs. tenure, colored by Churn."

High-charge, low-tenure customers cluster as churned. New customers on expensive plans are at highest risk.

2 Follow-Up: Churn by Contract

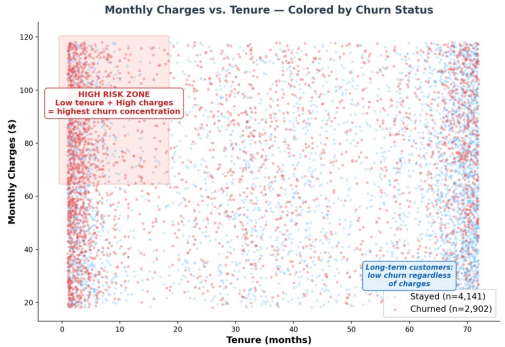
"Break down churn rate by contract type."

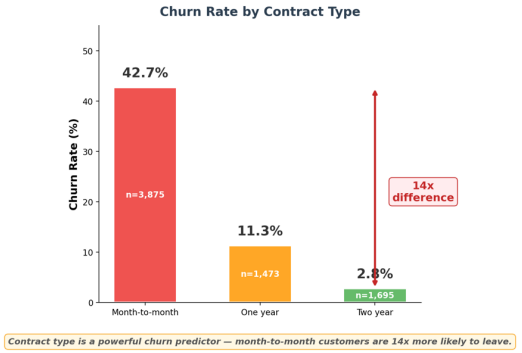
Month-to-month: ~43% churn. One year: ~11%. Two year: ~3%. A 14× difference — contract type is a powerful predictor.

3 Feature Insight

Discovery synthesis

Contract type, tenure, and charges are strong individual predictors. Their interactions may be even more informative for modeling in L6.





The discovery thread: scatter → contract breakdown → feature insight. Each finding raised the next question.

Categorical Encoding

ML models need numerical inputs. Categorical features must be encoded — but how?

One-Hot Encoding

Create binary columns for each category.

When to use: Nominal features (no natural order), few levels.

```
Contract:
  Month-to-month → [1, 0, 0]
  One year → [0, 1, 0]
  Two year → [0, 0, 1]
```

✓ Most common

Ordinal Encoding

Assign ordered integers to categories.

When to use: Naturally ordered features (e.g., education level: HS < BS < MS < PhD).

```
Satisfaction:
  Low → 1
  Medium → 2
  High → 3
```

✓ Preserves order

Label Encoding

Assign arbitrary integers to categories.

When to use: Generally avoid — except for tree-based models that handle splits naturally.

```
Contract:
  Month-to-month → 0
  One year → 1
  Two year → 2
```

⚠ Use with caution

Key rule: encoding must respect the structure of your data. One-hot is safest for nominal categories.

Encoding the Telco Data

Applying encoding decisions to our running example.

Encoding Decisions

Feature	Encoding	Reason
`Contract`	One-hot	No natural order
`PaymentMethod`	One-hot	No natural order
`InternetService`	One-hot	No natural order
`gender`	One-hot	Binary nominal
`SeniorCitizen`	One-hot	0/1 also valid

Python / sklearn

```
from sklearn.preprocessing import \
    OneHotEncoder

# Identify categorical columns
cat_cols = ['Contract',
  'PaymentMethod', 'InternetService',
  'gender', 'SeniorCitizen']

encoder = OneHotEncoder(
    sparse_output=False,
    handle_unknown='ignore')
encoded = encoder.fit_transform(
    df[cat_cols])
```

🤔 **Dimensionality Increase:** One-hot encoding increases the number of columns. The original 21 columns become ~30+ columns after encoding. Each categorical feature with k levels creates k new binary columns. This trade-off (more columns, but each is numerically meaningful) is usually worthwhile for model interpretability.

⚠️ **Watch Out:** `SeniorCitizen` is binary categorical; its 0/1 values already form a valid indicator encoding.

Scaling Review: When and Why

Callback to L1: min-max and z-score standardization — when do they matter for modeling?

✓ When Scaling Matters

Distance-based methods
KNN, SVM, PCA — features with larger ranges dominate distance calculations

Gradient-based methods
Logistic regression, neural networks — unscaled features cause slow or unstable convergence

→ When Scaling Doesn't Matter

Tree-based methods
Decision trees, random forests, gradient boosting — splits are scale-invariant

Gaussian Naïve Bayes
Positive affine scaling preserves Gaussian likelihood ratios

Z-score: $z = (x - \mu) / \sigma$
Centers at 0, scales by standard deviation

Min-max: $x' = (x - x_{min}) / (x_{max} - x_{min})$
Training range: [0, 1]; new values may exceed it

Rule of thumb: if features have very different ranges and you're using distance/gradient methods, scale.

Feature Engineering: First Taste

EDA-driven feature creation — the best features come from domain knowledge + EDA discoveries.

Average Monthly Charge

`TotalCharges / tenure`

Adjusts for tenure; pricing differences may remain. Use only when tenure > 0; handle tenure = 0 separately.

Domain knowledge: total spend grows with time; rate of spend is more informative.

Number of Services

Count of subscribed services

Aggregate engagement metric — customers using more services may be more sticky or more frustrated.

EDA insight: individual service flags are noisy; the total captures engagement level.

Tenure Bins

New (0–12) / Mid (13–36) / Long (37–72)

Based on the bimodal distribution discovered in EDA. Captures the natural customer lifecycle phases.

EDA discovery: tenure histogram showed distinct clusters at short and long tenures.

💡 Key Insight

Best features come from domain knowledge + EDA discoveries. The CTA can implement any transformation — you decide WHICH transformations are meaningful.

Pipeline Building

> Consolidate all wrangling steps — type fixing, imputation, encoding, scaling — into a single `sklearn Pipeline`. Then challenge for completeness: does it handle new data, unseen categories, and edge cases?

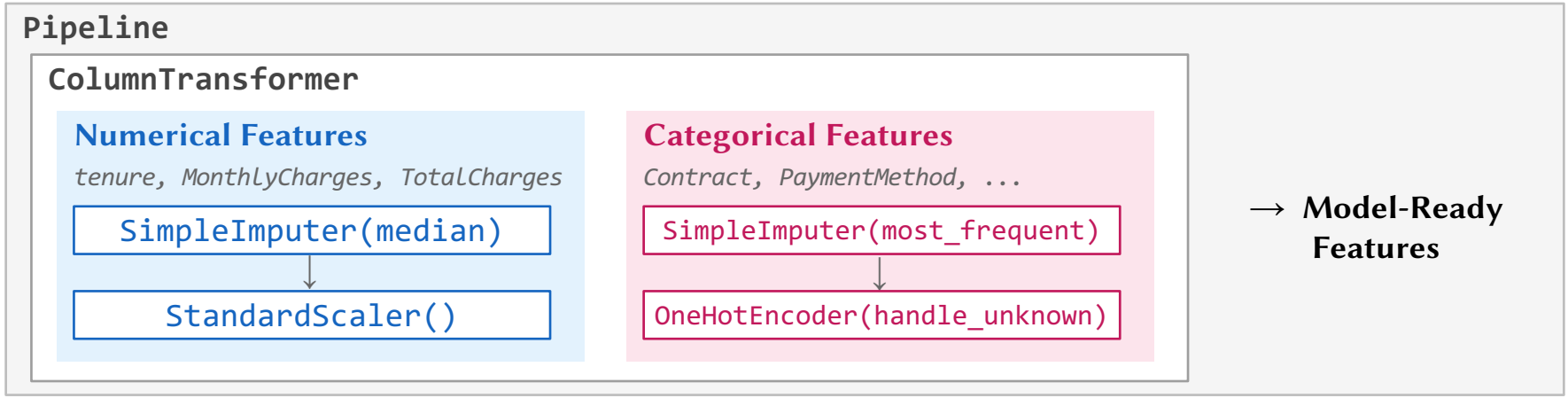
Challenging for Completeness — Key Questions:

1. What happens if the test data has a category level that wasn't in the training data?
2. Did you include SeniorCitizen in the categorical pipeline or the numerical one?
3. Should we add feature engineering steps (average monthly charge, number of services) into the pipeline?
4. Is the order of operations correct — should we impute before or after encoding?

Each question reveals a gap. The CTA addresses them, producing a more robust pipeline through iteration.

The Reproducible Pipeline

`sklearn ColumnTransformer + Pipeline` Structure

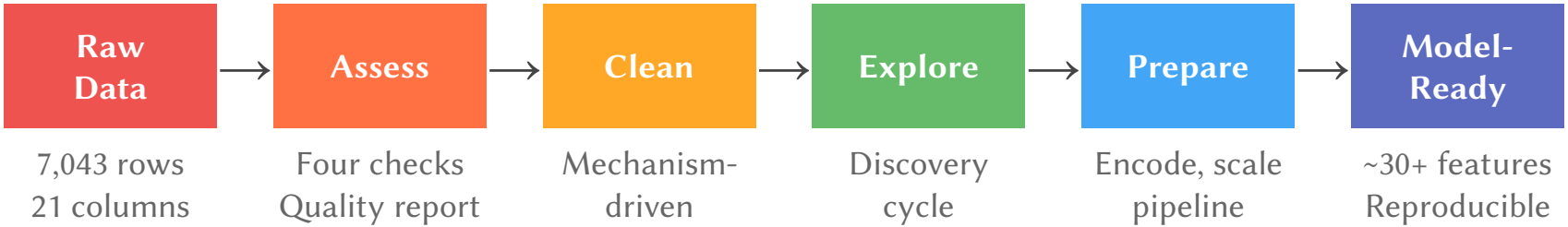


Why Reproducibility Matters

- Consistent train/test preprocessing — same transformations applied identically
- No data leakage — fitting only on training data, transforming both
- Easy to modify — swap components without rewriting the entire workflow

From Raw Data to Model-Ready Features

The complete Part 2 journey — from raw Telco data to features ready for L6 classification.



> 💡 **Key Insight**
> Every step involved human judgment. The CTA made each step faster. Your judgment made each step correct.

Telco dataset is now ready for Lecture 6: Classification

The cleaned, encoded features will serve as input for predicting customer churn — our first real classification problem.

Today's Key Takeaways

1. **Same pipeline, different domain knowledge**
   Healthcare, finance, climate, smart cities — all follow Question → Collect → Clean → Analyze → Model → Communicate.

2. **CTA as brainstorming partner — but the idea must be yours**

   Five-step ideation: interest → questions → feasibility → refine → sketch. The CTA generates; you decide.

3. **Four-check protocol as professional skill**

   Structural, completeness, validity, distributional — a systematic approach you'll use in projects and careers.

4. **Cleaning choices have consequences — understand why**
   MCAR/MAR/MNAR guides strategy. Simple imputation can distort relationships; conditional methods need suitable models and assumptions.

5. **EDA is discovery, not a checklist**

   Let findings guide questions. Consolidate into reproducible pipelines for consistent, leak-free preprocessing.

The CTA Role Transition

How your interaction strategy with the CTA evolved through the course so far.

L2–L3

Learning Partner

Explaining concepts, verifying understanding, building intuition through worked examples.

Interaction strategy:
Ask "why" and "how" questions. Use CTA to check your reasoning.

→

L4 Part 1

Brainstorming Partner

Generating and evaluating project ideas, exploring application domains, creative ideation.

Interaction strategy:
Structured prompts with constraints. Evaluate — don't accept blindly.

→

L4 Part 2

Implementation Assistant

Writing code, generating reports, building pipelines. CTA produces; you direct and verify.

Interaction strategy:
Structured requests. Challenge for completeness. Verify consequences.

Technically Intriguing Questions

Questions that reward curiosity — explore these with the CTA on your own.

🤔 Question 1

When would you deliberately choose NOT to impute missing data and instead drop rows? What are the tradeoffs?

Think about: small vs. large datasets, high missingness proportions, analysis types that are robust to missing data.

🤔 Question 2

Can you use PCA (from L2) as a data quality diagnostic? How would outliers appear in PCA space?

Try it: run PCA on a dataset with outliers. Connects L2 to L4 — PCA as diagnostic, not just dimensionality reduction.

🤔 Question 3

How would you assess data quality for unstructured data like text or images, where the four-check protocol doesn't directly apply?

Extends to: text quality (encoding, languages, boilerplate), image quality (resolution, labeling errors). Forward pointer to L7 and L9–L10.

Homework 1

Deadline: October 3 at 11:59 pm

Now you have your group

Course Project Proposal

Deadline: October 11 at 11:59 pm

Questions?

CONTACT INFORMATION



yangliu.mit.edu

Instructor: Prof. Yang Liu

Email: yliu12@hku.hk

Office Hours: 5:00 - 5:50 PM, Thursdays

Location: CB-714, Chow Yei Ching Building

Let me know if you want remote!

🤖 **CTA is your 24/7 interactive and responsive learning partner!**

TA contacts

Homework #1

Tuesday 4-5PM

Yushi Du:
u3012725@connect.hku.hk
https://us04web.zoom.us/j/78561597481

Zhengyang Hu:
u3010250@connect.hku.hk

Thursday 3-4PM

Jun Dai:
jundai332@connect.hku.hk
https://rochester.zoom.us/j/97202452759

Lulu Cai:
u303011209@connect.hku.hk

TA office hours on Zoom! Two slots (1 hour each) on Tuesdays and Thursdays