50.038 Computational Data Science

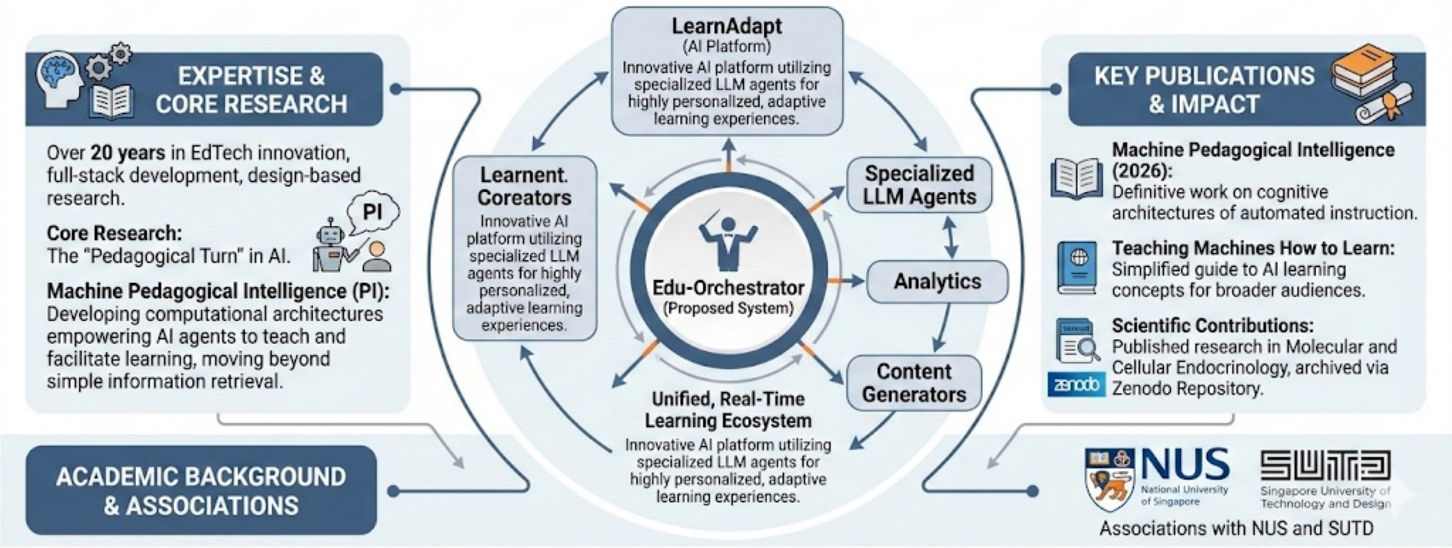
SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Week 2: The Raw & The Cooked – Features, Data, and Architectures

From Raw Signals to High-Dimensional Intelligence

Nizam Kadir

About Me - Nizam Kadir



EXPERTISE & CORE RESEARCH

Over **20 years** in EdTech innovation, full-stack development, design-based research.

Core Research: The "Pedagogical Turn" in AI.

Machine Pedagogical Intelligence (PI): Developing computational architectures empowering AI agents to teach and facilitate learning, moving beyond simple information retrieval.

ACADEMIC BACKGROUND & ASSOCIATIONS

LearnAdapt (AI Platform)
Innovative AI platform utilizing specialized LLM agents for highly personalized, adaptive learning experiences.

Learnent. Coreators
Innovative AI platform utilizing specialized LLM agents for highly personalized, adaptive learning experiences.

Edu-Orchestrator (Proposed System)

Specialized LLM Agents

Analytics

Content Generators

Unified, Real-Time Learning Ecosystem
Innovative AI platform utilizing specialized LLM agents for highly personalized, adaptive learning experiences.

KEY PUBLICATIONS & IMPACT

Machine Pedagogical Intelligence (2026): Definitive work on cognitive architectures of automated instruction.

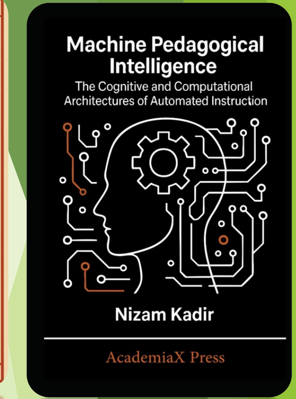
Teaching Machines How to Learn: Simplified guide to AI learning concepts for broader audiences.

Scientific Contributions: Published research in Molecular and Cellular Endocrinology, archived via Zenodo Repository.

zenodo
NUS National University of Singapore
SUTD Singapore University of Technology and Design

Associations with NUS and SUTD

TEACHING MACHINES HOW TO LEARN
Artificial Intelligence Made Easy
NIZAM KADIR



Machine Pedagogical Intelligence
The Cognitive and Computational Architectures of Automated Instruction
Nizam Kadir
AcademiaX Press

LinkedIn https://www.linkedin.com/in/nizam-kadir

nizam_kadir@mymail.sutd.edu.sg

https://www.NizamKadir.com

Getting to know you…

1. **Dream Big, Start Small…** (What problems do you want to solve with the skills gained in this course?)
2. **Start with the end in the mind** (how are you going to do well in this course? Lab check-offs? Project Work? Final Exams (MCQs)?)
3. **Can I predict your grades based on the survey below?** How can I teach machines to learn from data, to make a strong prediction of your grades, or how can I use the data collected to improve the course, or make you learn better?

Scan for Survey

Student Selector

https://forms.office.com/r/p7JQLfZvgs

Admin/Academic Matters

- **Show-and-Tell Your Previous Work: 2% Lab Check-Off (Guaranteed) for the Week**
  - Prepare a presentation of your previous work involving any of the topics, and share during the week which we cover the topics.
  - Duration: max. 5 minutes per selected work
  - Max. Slides: 10 slides (must have GitHub link to your codes, etc), Email me your slides 1 week in advance)
  - You will show-and-tell as an opening to the week's topics.
  - Only 2 Show-and-Tell presentations per week (1 per lecture session)
  - First-comes-first-serve (email me, which topics, which week, what is your Show-and-Tell work, share with me your GitHub link). I will select the top 2 for each week/topics.
- **Use of AI.** AI will be assumed to have been used by everyone. So, please use AI to augment your learning. I will focus on the process, rather than the product of your learning.

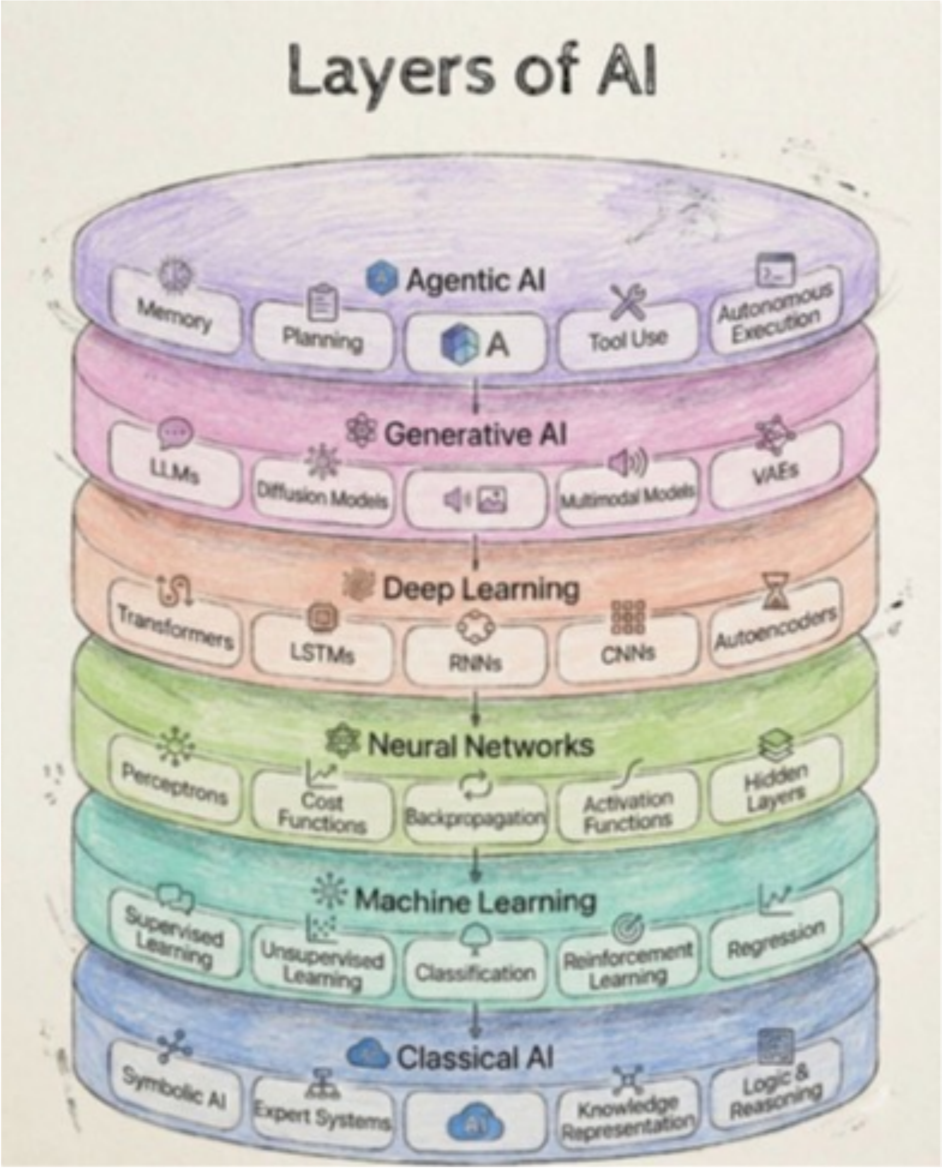
Critical Thinking For the Week

- What remains uniquely human if AI writes most code?
- What risks arise from AI systems writing themselves?
- How should data scientists be evaluated when coding is automated?

Scan for Resource

☆ Student Selector

https://nizamkadir.com/the-ouroboros-moment-why-claude-writing-itself-marks-the-end-of-coding-as-we-know-it/



The **hierarchical architecture of Computational Data Science**, highlighting how cutting-edge capabilities like Agentic AI rely heavily on the fundamental principles of Machine Learning and Neural Networks.

So, where does **Data / Features** fit into this?

What will we be covering for this week?

- **Lecture Part 1 (1.5 Hours):** The Physics of Data – Foundations, Quality, and Architectures.
- **Lecture Part 2 (1.5 Hours):** Structure from Chaos – Dimensionality Reduction and NLP.
- **Lab Session (2 Hours):** Polars, UMAP, and LLM-based feature extraction.

Learning Objectives for this Week

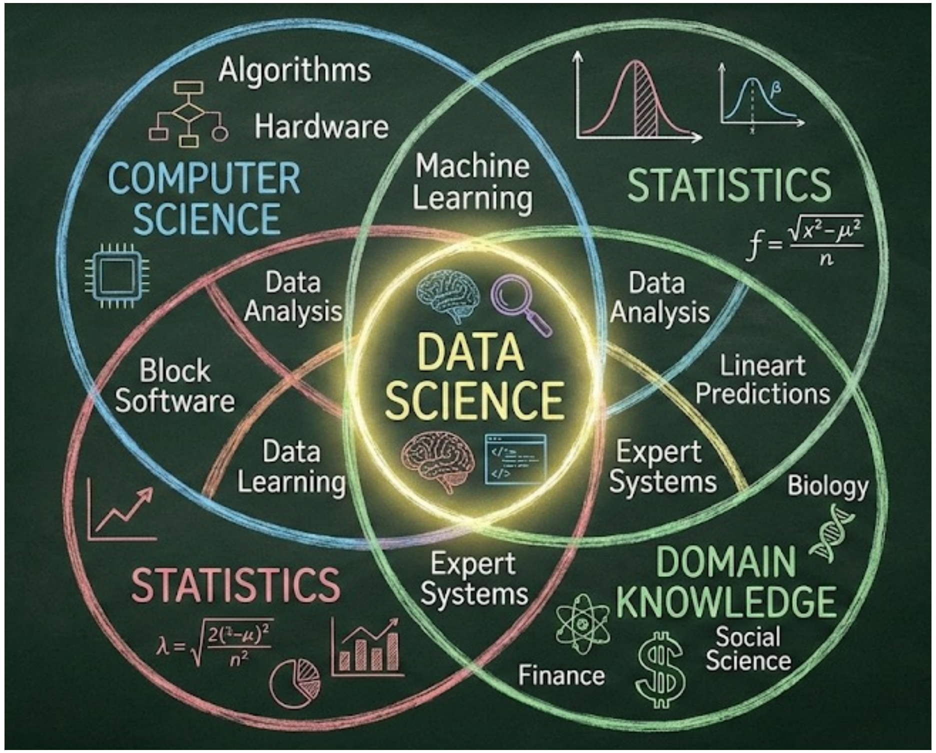
By the conclusion of this week's lesson, students are expected to:

- **Analyze Data Fundamentals:** Differentiate between feature types (Nominal, Ordinal, Interval, Ratio) and select appropriate target variables for Classification vs. Regression.
- **Master Data Quality & Pre-processing:** Identify common quality issues (noise, outliers, missing values) and implement cleaning strategies like standardization and normalization.
- **Navigate the Modern Stack:** Compare and contrast the performance benefits of **Polars** (lazy evaluation, columnar) versus traditional **Pandas** for large-scale data.
- **Apply Dimensionality Reduction:** Implement and interpret linear (PCA) and non-linear (t-SNE, UMAP) techniques to visualize high-dimensional structures.
- **Execute Advanced Feature Extraction:** Move beyond basic NLP (TF-IDF) to apply **Zero-Shot Feature Extraction** using Large Language Models (LLMs) on unstructured text.

Lecture Part 1
The Physics of Data

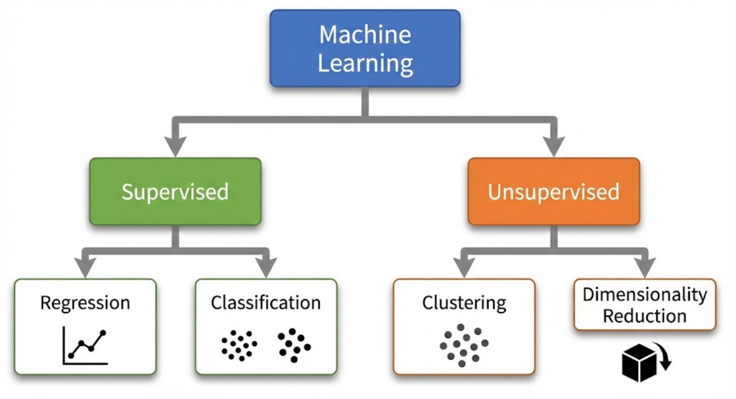
Foundations, Quality, and Architectures.

Defining our Territory



- **Data Science:** The full lifecycle (Collection → Cleaning → Modeling → Deployment).
- **Data Analysis:** The Detective. Retrospective. *"What happened?"* (Descriptive Stats).
- **Machine Learning:** The Oracle. Prospective. *"What will happen?"* (Predictive Models).

The Taxonomy of Learning



- **Supervised (Labeled):**
  - *Regression*: Predicting continuous values (Price, Temperature).
  - *Classification:* Predicting discrete labels (Spam/Not Spam, Churn/Stay).
- **Unsupervised (Unlabeled):**
  - *Clustering*: Finding groups (Customer Segmentation).
  - *Dimensionality Reduction*: Compression (PCA, UMAP).

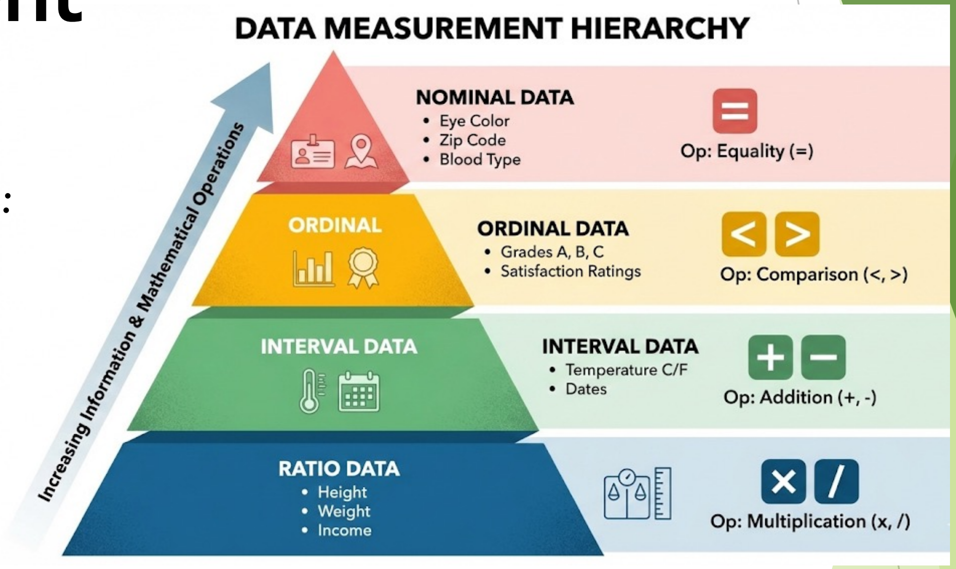
Activity (Think-Pair-Share)

”You want to predict the exact dollar amount a customer will spend next month. Is this Classification or Regression?"

Student Selector

The Atomic Unit: Features & Stevens' Scales of Measurement

- **Nominal:** *Labels with no order* (Eye Color, Zip Code). *Op: Equality (=).*
- **Ordinal:** *Labels with rank* (Grades A, B, C). *Op: Comparison (<, >).*
- **Interval:** *Differences are meaningful, no true zero* (Temperature C/F, Dates). *Op: Addition (+, -).*
- **Ratio:** *Ratios are meaningful, true zero exists* (Height, Weight, Income). *Op: Multiplication (x, /).*



DATA MEASUREMENT HIERARCHY

Increasing Information & Mathematical Operations

NOMINAL DATA
- Eye Color
- Zip Code
- Blood Type

Op: Equality (=)

ORDINAL

ORDINAL DATA
- Grades A, B, C
- Satisfaction Ratings

Op: Comparison (<, >)

INTERVAL DATA

INTERVAL DATA
- Temperature C/F
- Dates

Op: Addition (+, -)

RATIO DATA
- Height
- Weight
- Income

Op: Multiplication (x, /)

Why does it matter? *You cannot calculate the "average" Zip Code. It's mathematically meaningless even though it looks like a number. You cannot say 20°C is "twice as hot" as 10°C, because 0°C is arbitrary. You can say $200 is twice as much as $100. Knowing these properties prevents you from using the wrong algorithm. Neural networks, for example, assume Ratio inputs; if you feed them Nominal data without encoding, they will fail.*

Let's Explore Further!

Activity (Group Discussion)

- **The "Ordinal" Gray Area:** Ordinal data (like 5-star Amazon reviews). Is a 4-star rating really "twice as good" as a 2-star rating? This is a huge debate in survey analysis.
- **Encoding for Machine Learning:** Neural Networks fail with raw Nominal data. We could look at *how* we fix that (e.g., One-Hot Encoding vs. Label Encoding).
- **The Impact on Statistics:** We could look at which statistical tests (t-tests, Chi-square, ANOVA) are "legal" for each specific data type.
- The distinctions between Nominal, Ordinal, Interval, and Ratio (often remembered by the acronym **NOIR**) dictate everything from which summary statistics are valid to which machine learning algorithms will converge.
- Treating a variable as the wrong type, like averaging zip codes (Nominal) or assuming the difference between "Agree" and "Strongly Agree" is the same as "Disagree" and "Neutral" (Ordinal vs. Interval), can lead to fundamentally flawed insights.

★ Student Selector

Discrete vs. Continuous Variables

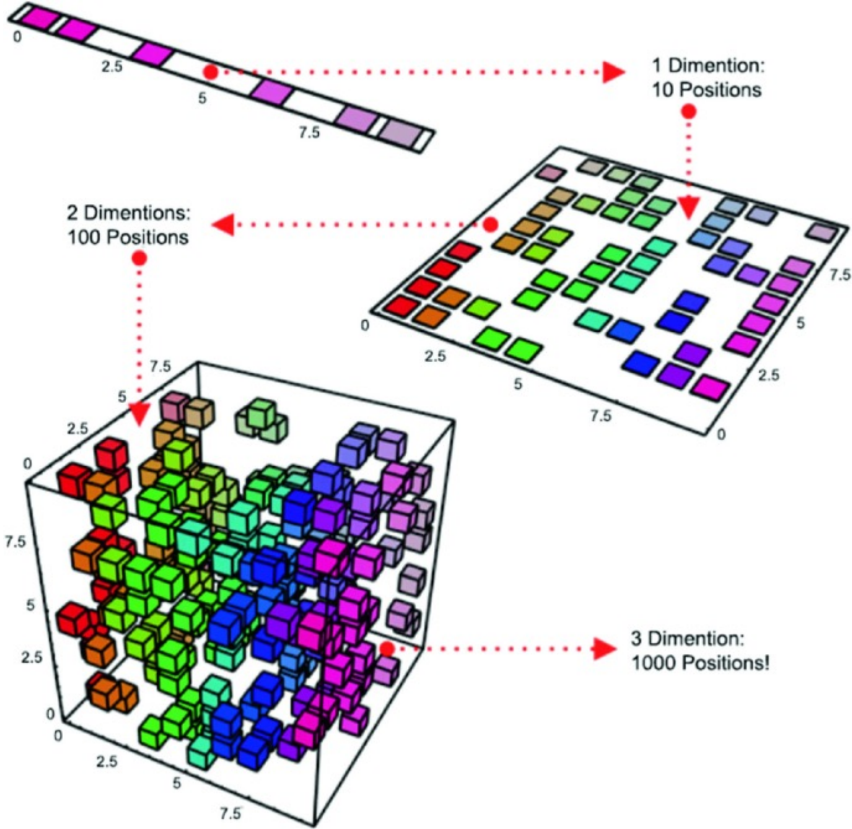
- **Discrete:** Countable, finite steps (Number of children, Digital image pixels).
- **Continuous:** Infinite gradations within a range (Time, Weight, Analog signal).

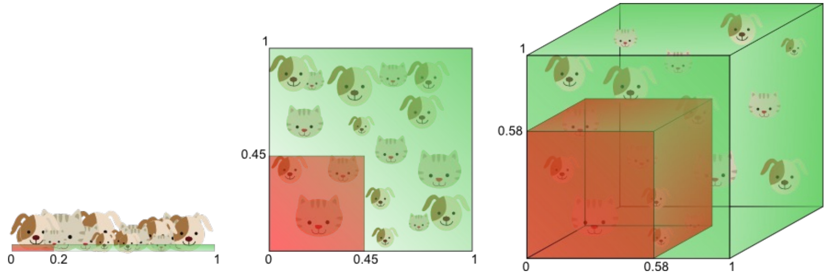
Categorical vs. Continuous variables leads to Classification vs. Regression

This distinction drives our target variable choice. If your target is discrete (churn/no churn), you are building a Classifier. If it is continuous (lifetime value), you are building a Regressor. We can also transform between them—discretizing continuous data (Age -> Age Groups) or encoding categorical data (Red ->).

The "Curses" of Data

- **Dimensionality:** High feature count (D).
- **The Curse:** As D increases, volume explodes. Data becomes sparse. Distance becomes meaningless.
- **Sparsity:** Matrices filled mostly with zeros (common in Text/Recommender Systems).
- **Resolution:** Granularity (Hourly vs. Weekly data).





Imagine searching for a needle in a haystack. *Now imagine the haystack is the size of the galaxy. That is high-dimensional space. To maintain statistical significance, the amount of data required grows exponentially with each added feature. This is why we need Dimensionality Reduction, which we will cover in Part 2.*

Data Quality — The Dirty Reality

- **Noise:** Random variance/error in measurement.
- **Outliers:** Deviants from the distribution. (Signal or Error?).
- **Missing Values:**
  - *MCAR*: Missing Completely At Random.
  - *MNAR*: Missing Not At Random (e.g., High earners hiding income).
- **Inconsistency:** "CA", "Calif.", "California".

Activity: A messy spreadsheet

	A	B	C	D	E	F	G
1	**Customer ID**	**Name**	**Age**	**State**	**Income (k)**	**Satisfaction Score (1-10)**	**Notes**
2	1001	John Smith	28	CA	65	8	
3	1002	Jane Doe	34	N/A	72	9.5	
4	1003	Bob Johnson	-5	Calif.		3	
5	1004	Alice		NY	120	7	
6	1005	Charlie Brown	42	California	55000	6	
7	1006	David	29	TX	NaN	2	
8	1007	Eve	31	New York	88	10	
9	1008	Frank	55	CA			
10	1009	Grace	150	FL	45	1	
11	1010	Heidi	22	ca	52	8	
12	1011	Ivan	40		250		
13	1012	Judy	33	TX	68	7.89231	

★ **Student Selector**

Pre-processing — Sampling & Aggregation

- **Aggregation:** Summarizing data (Daily Sales → Monthly Sales). Reduces noise, changes scale.
- **Sampling:**
  - *Simple Random:* Equal probability.
  - *Stratified*: Preserves class ratios (Crucial for Imbalanced Data like Fraud/Churn).

Include Data Loaders

In Deep Learning, we use Data Loaders to handle sampling dynamically. Instead of loading 1TB of data into RAM, a Data Loader streams small "batches," shuffling and sampling on the fly. This allows us to train on datasets larger than our machine's memory.

Feature Engineering — The Art Form

- **Feature Creation:** Making new info from old. (Mass & Volume → Density).
- **Discretization:** Smoothing noise (Age 24.5 →"20-30").
- **Binarization:** Thresholding (Prob > 0.5 → 1).
- **Feature Interactions:** Creating *A* x *B* to capture non-linear effects.

Sometimes the signal isn't in *A* or *B*, but in their interaction. If you are predicting house prices, "Width" and "Length" are okay, but "Square Footage" (Width x Length) is the driver. We call these Interaction Features. Conversely, if two features are identical (redundant), we must remove one to prevent Multicollinearity.

The Modern Stack — Pandas vs. Polars

pandas

The OG of data analysis

- **Pandas:** The Standard.
  - *Pros:* Huge ecosystem, flexible.
  - *Cons:* Single-threaded, high memory usage, eager execution.

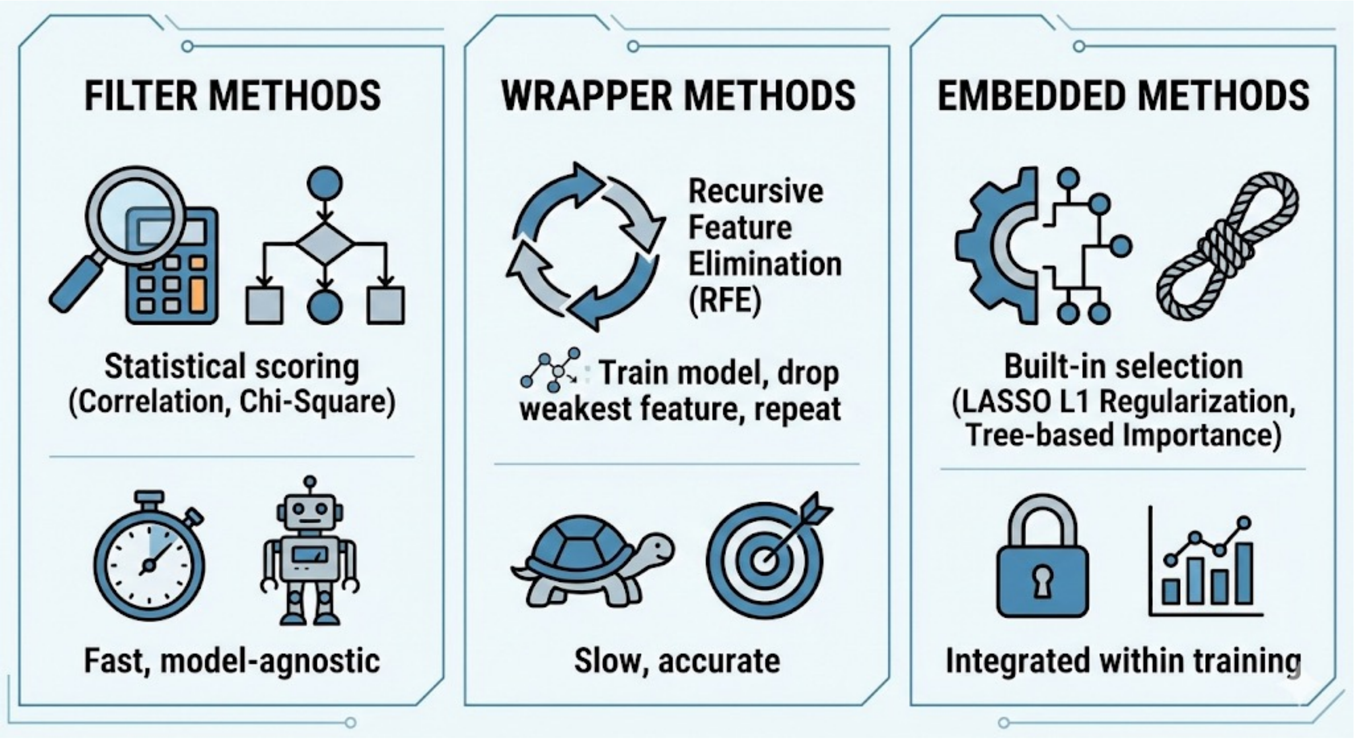
polars

The blazing-fast newcomer

- **Polars:** The Challenger.
  - *Pros:* Multi-threaded (Rust), Lazy Evaluation, Columnar (Apache Arrow).
  - *Performance:* 10-100x faster on large aggregations.

Pandas is built on NumPy and processes data row-by-row. Polars is built on Rust and Apache Arrow, using a columnar format. This means Polars can perform operations in parallel across all your CPU cores. It also uses "Lazy Evaluation”; it doesn't run your code line-by-line. It looks at the whole query, optimizes it (e.g., filtering *before* loading), and then executes. This is a game-changer for big data on a single machine.

Feature Selection Strategies



FILTER METHODS

Statistical scoring (Correlation, Chi-Square)

Fast, model-agnostic

WRAPPER METHODS

Recursive Feature Elimination (RFE)

Train model, drop weakest feature, repeat

Slow, accurate

EMBEDDED METHODS

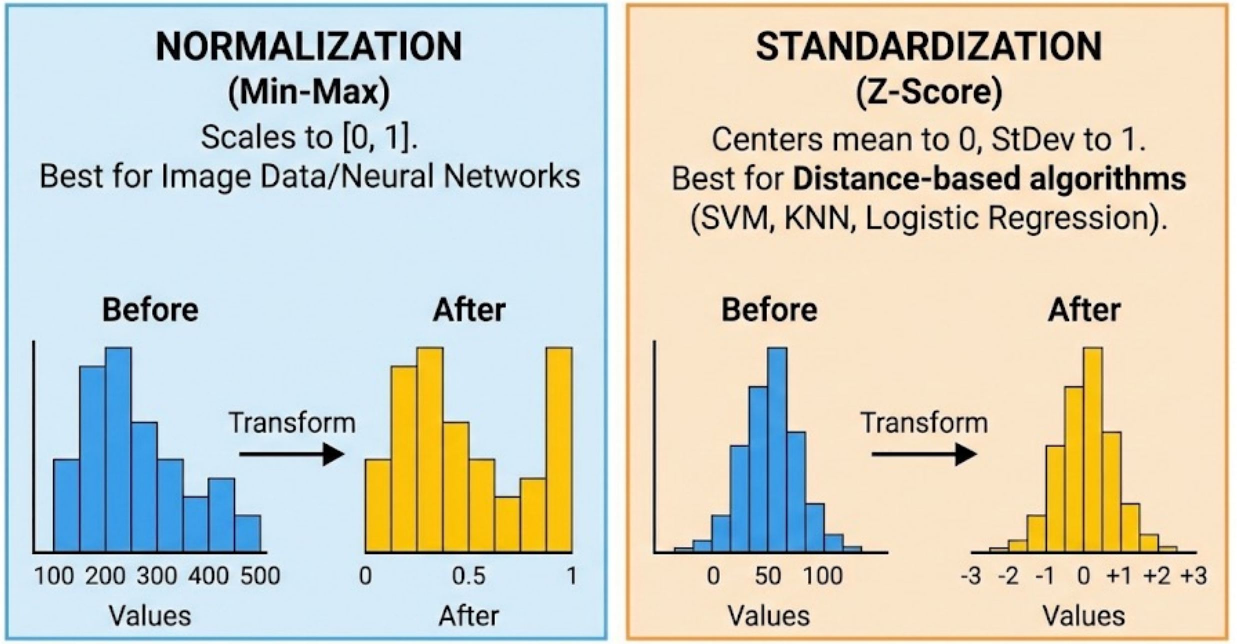
Built-in selection (LASSO L1 Regularization, Tree-based Importance)

Integrated within training

Filter methods are like a sieve; they just check the features against the target. Wrapper methods are like a tryout; they test how the model performs with specific teams of features. Embedded methods are efficient because the model selects features while it learns.

Standardization vs. Normalization

- **Normalization (Min-Max):** Scales to [min,max]. Best for Image Data/Neural Networks.
- **Standardization (Z-Score):** Centers mean to 0, StDev to 1. Best for Distance-based algorithms (SVM, KNN, Logistic Regression).



NORMALIZATION (Min-Max)

Scales to [0, 1].
Best for Image Data/Neural Networks

STANDARDIZATION (Z-Score)

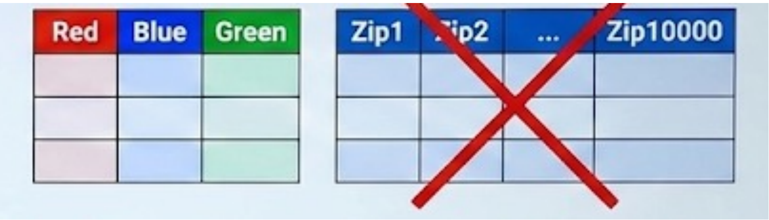
Centers mean to 0, StDev to 1.
Best for **Distance-based algorithms** (SVM, KNN, Logistic Regression).

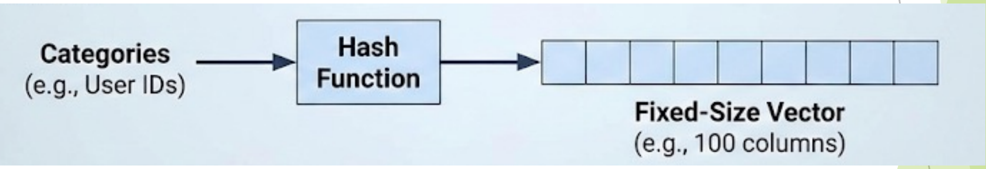
CRITICAL WARNING:

- Fit scaler on **TRAINING** data only.
- Transform **TEST** data using Training parameters.
- *Never* fit on the whole dataset. This is **Data Leakage**.

Handling High Cardinality

- **One-Hot Encoding:** Good for few categories (Red, Blue, Green). Bad for many (Zip Codes).
- **Feature Hashing (The Hashing Trick):** Maps categories to a fixed-size vector using a hash function. Fast, low memory, but collisions are possible.




Categories (e.g., User IDs)

Hash Function

Fixed-Size Vector (e.g., 100 columns)

If you have 10,000 User IDs, One-Hot Encoding creates 10,000 columns. Your RAM will crash. Feature Hashing maps these 10,000 IDs to a fixed number of columns (say, 100) using a hash function. It's a trade-off: you save massive memory, but two different IDs might map to the same column (collision).

Review & Quiz

1. Is "Zip Code" a Ratio variable?
2. If I fit my scaler on the Test set, what have I done?
3. True or False: Polars uses Eager execution by default.

Bring writing materials for all lessons!

Scan for Quiz

Student Selector

https://forms.office.com/r/2z5zTG94LR

Lecture Part 2
Structure from Chaos

Dimensionality Reduction and NLP.

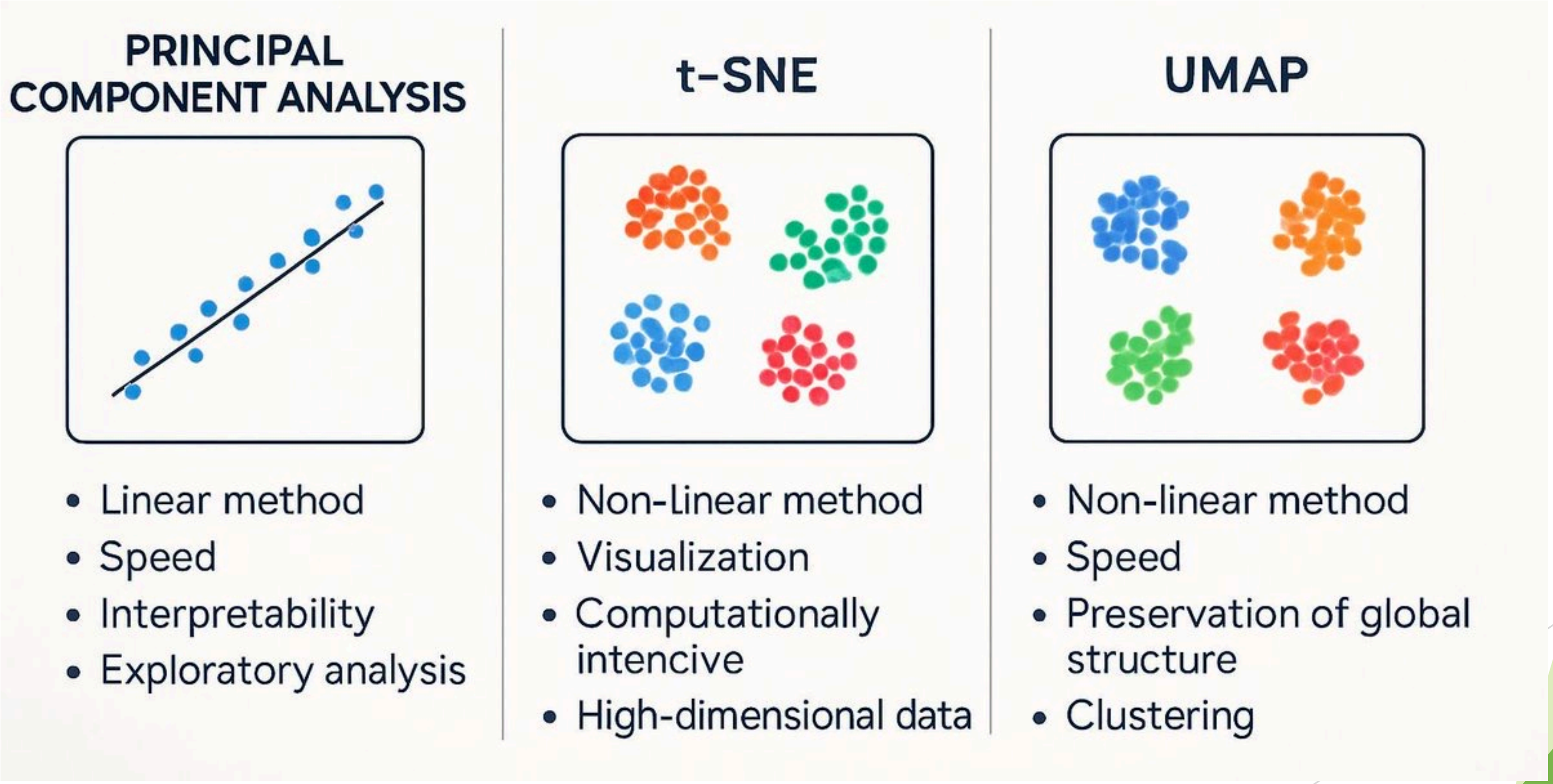
Activity: Write down notes on the piece of paper as you watch the following video. You will then discuss with other students around you. You may be chosen to present to the class, what is **‘The Curse of Dimensionality’.** You will then submit the notes to me. Make sure you write your name and today’s date. Your notes will then be peer-reviewed.

The Curse of Dimensionality

- In high dimensions, data becomes sparse.
- "Distance" loses meaning (everyone is far from everyone).
- We need to project data to lower dimensions (2D or 3D) to see it. Dimensionality Reduction

★ Student Selector

Dimensionality Reduction: PCA vs t-SNE vs UMAP



PRINCIPAL COMPONENT ANALYSIS

- Linear method
- Speed
- Interpretability
- Exploratory analysis

t-SNE

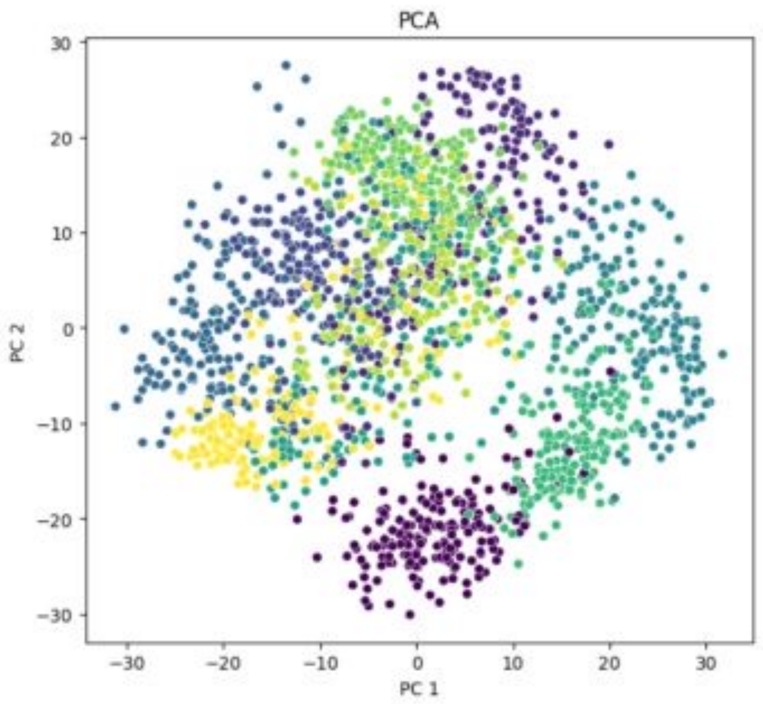
- Non-Linear method
- Visualization
- Computationally intencive
- High-dimensional data

UMAP

- Non-linear method
- Speed
- Preservation of global structure
- Clustering

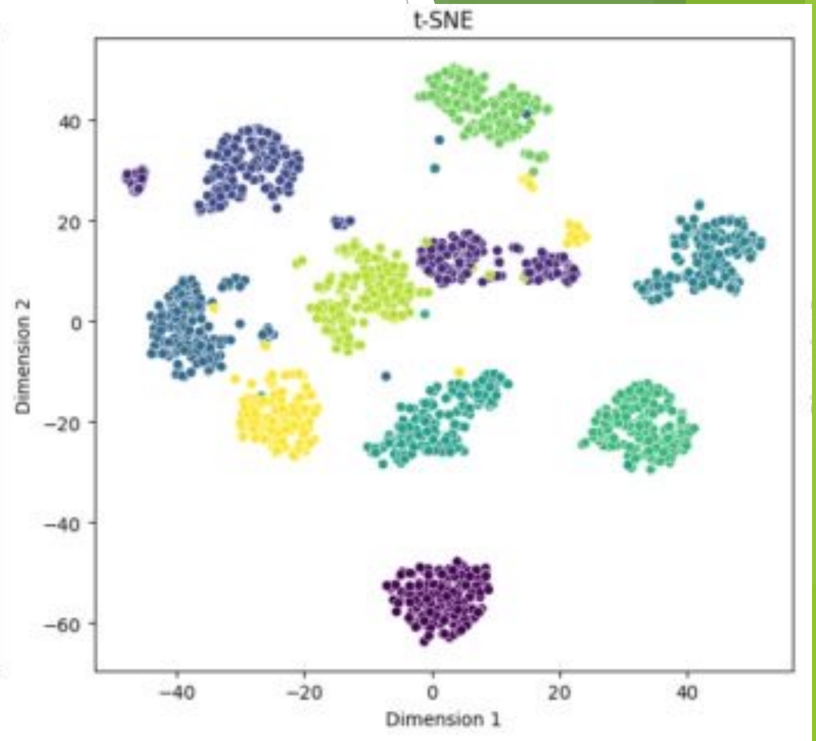
Linear Reduction — PCA

- **Principal Component Analysis**
  - Finds the "axes of greatest variance."
  - Rotates the data to fit these axes.
- **Pro:** Fast, Global, Deterministic.
- **Con:** Linear only. Cannot unroll a "Swiss Roll" shape.



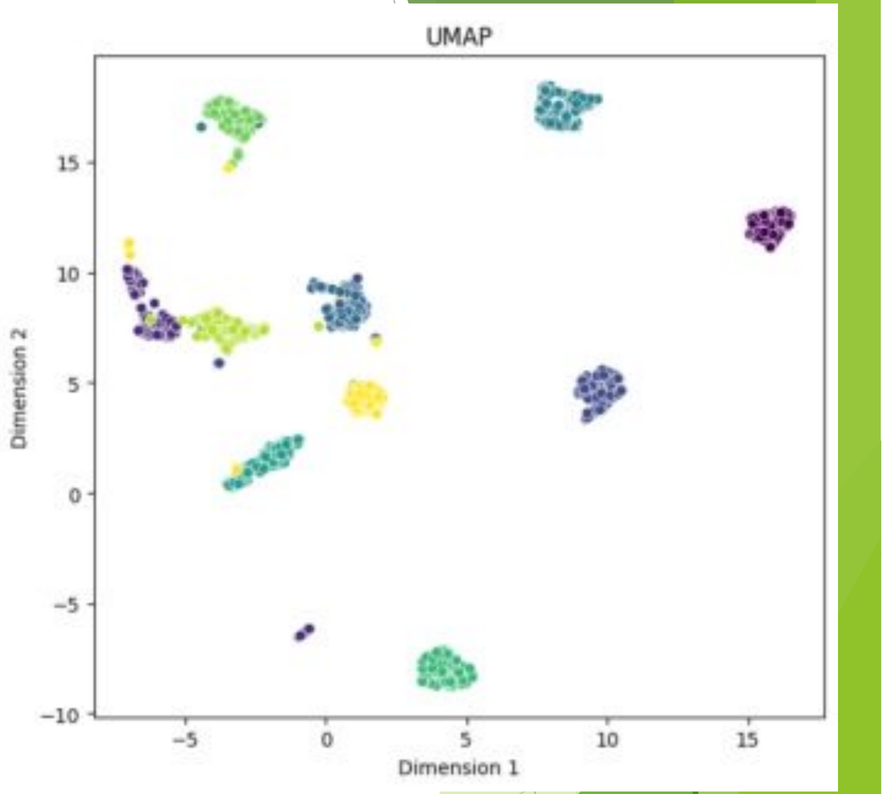
Manifold Learning — t-SNE

- **t-Distributed Stochastic Neighbor Embedding**
  - Non-linear. Focuses on keeping similar neighbors close.
  - Uses probability distributions (Gaussian in high-D, Student-t in low-D).
- **Drawbacks:**
  - Slow ($O(N^2)$).
  - Global structure is lost (Cluster distance is meaningless).
  - Stochastic (different result every time).



The New King — UMAP

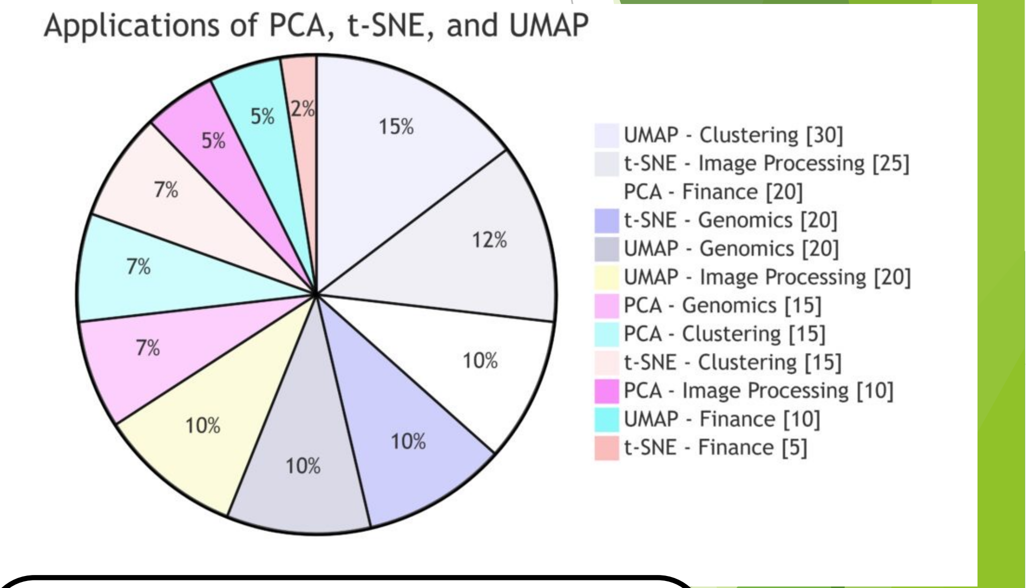
- **Uniform Manifold Approximation and Projection.**
  - Balances Local AND Global structure.
- **Speed:** Much faster than t-SNE (*O (N log N)*).
- **Math:** Based on Riemannian geometry and algebraic topology.



Just as a mapmaker tries to flatten the spherical Earth onto a 2D sheet of paper (distorting distances or shapes in the process), these algorithms try to flatten complex 100-dimension data onto a 2D screen. UMAP is simply a better map projection that preserves both the countries (clusters) and the oceans between them (global distance).

Choosing the Best Technique for Your Data

- When it comes to choosing the right **dimensionality reduction technique**, think about what you value most: interpretability, speed, or the ability to capture non-linear patterns.
  - **Go with PCA** if you need a quick, interpretable solution that works well for linearly related data.
  - **Use t-SNE** when working with complex, clustered data and you want a strong, detailed visualization.
  - **Choose UMAP** for large datasets or when both local and global structure matter, like in biological or text data.
- In many cases, it's worth trying out both **t-SNE** and **UMAP** side-by-side, especially if you're unsure which one will best represent your data's structure.



Student Selector

Source: https://aicompetence.org/pca-vs-t-sne-vs-umap/

Text Processing 101 — The Old Way

- **Tokenization:** Cutting text into units (words).
- **Stopwords:** Removing "the", "and".
- **Stemming vs. Lemmatization:**
  - *Stemming:* Chopping ("Running" → "Run"). Fast, crude.
  - *Lemmatization:* Dictionary look-up ("Better" → "Good"). Accurate, slow.
- **Representation:** Bag-of-Words (Count), TF-IDF (Weighted Frequency).

Deep Dive — TF-IDF

What is this specific document actually about?

- **Formula:** $TF \times \log(\frac{N}{DF})$
  - *TF (Term Frequency)*: How often word appears in *this* document.
  - *IDF (Inverse Document Frequency)*: How rare the word is in *all* documents.
- **Intuition:** If a word is rare globally but frequent here, it is a **keyword.**
- Brief intro only, better methods coming.

- N = Total number of documents.
- DF = Number of documents containing the word.

Example for Class: Imagine you have a library of 1,000 books

Formula: $TF \times \log(\frac{N}{DF})$

- **Word: "The"**
  - High TF (appears often in the book).
  - Low IDF (appears in all 1,000 books).
  - Result: TF-IDF is near **0**.
- **Word: "Voldemort"**
  - High TF (appears often in *Harry Potter*).
  - High IDF (appears in only 7 out of 1,000 books).
  - Result: TF-IDF is **Very High**.

★ **Student Selector**

The Paradigm Shift — LLM Feature Extraction

- Old Way: RegEx (Regular Expression), Keyword Matching.
- New Way: **Zero-Shot Extraction** with LLMs.
  - Using models (GPT, T5, BERT) to convert unstructured text into structured JSON.
- **Example:**
  - *Input:* "I called three times and nobody answered!"
  - *Extraction:* {"Issue": "Customer Service", "Sentiment": "Negative", "Urgency": "High"}.
- **Relevance:** This turns text columns into categorical features we can feed into our Churn Model (predicts if a customer would cancel their subscription).

Think of the Old Way as a library search engine that only looks for exact book titles. Think of the New Way as a librarian who has read every book and can answer, 'Give me a story about a sad robot,' even if 'sad' isn't in the title

Survival Analysis vs. Churn Prediction

- **Binary Churn Classification:** "Will they leave?" (Yes/No).
  - Good for immediate action.
- **Survival Analysis:** "When will they leave?" (Time-to-Event).
  - Good for Customer Lifetime Value (CLV).
- **Trade-off:** Survival analysis handles "Censored Data" (customers who haven't left *yet*) better than simple classification.

Standard classification has a flaw: If a customer has been with us 5 years and hasn't churned, the model just sees "0". Survival analysis sees "survived 60 months" and uses that duration to predict future probability. It's a more nuanced view of retention.

The Problem: Churn models (usually running on algorithms like Random Forest or Logistic Regression) love numbers and categories. They hate sentences.

The Solution: This process turns a "Text Column" (useless for the model) into three new "Feature Columns" (Issue Type, Sentiment Score, Urgency Level) that act as powerful predictors for the Churn Model.

The Feature Auction

★ Student Selector

- **Learning Outcome:** Feature selection intuition and domain knowledge application.
- **Concept:** Divide yourselves into teams. Each team has a a budget of $100.
- **Action:** You are given the potential features for the IBM Telco Churn dataset (customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn)
- **Goal:** Your team must bid on the **features** that you all think will be **most predictive of Churn**.
- **Reveal:** After bidding, I will show you the "Feature Importance" ranking (from a Random Forest model). Teams that bought high-importance features win.

Review & Quiz

- True or False: PCA can effectively unroll a non-linear "Swiss Roll" shape.
- Which dimensionality reduction technique is Stochastic and yields a different result every time?
- What modern "Zero-Shot" technique replaces complex RegEx for extracting features like Sentiment?

Scan for Quiz

★ Student Selector

https://forms.office.com/r/rUVighkKX1

Week 2 Lecture Wrap-up

- The "Physics" of Data
  - **Measurement Matters:** We established that data isn't just numbers; it has a structure (NOIR). Ignoring the difference between Nominal and Ratio data leads to "mathematical nonsense," such as averaging Zip Codes.
  - **Quality is Contextual:** We explored how "Missingness" tells a story (MCAR vs. MNAR) and why Data Leakage during scaling (fitting on the whole dataset) is a critical failure point.
- The Modern Architectures
  - **Speed & Scale:** We contrasted the legacy "Eager" execution of **pandas** with the high-performance "Lazy" execution of **Polars**, preparing us for datasets that exceed memory limits .
  - **Structure from Chaos:** We moved beyond linear PCA. We learned that **UMAP** has dethroned t-SNE by preserving both local clusters *and* the global relationships between them .
- The NLP Paradigm Shift
  - **From Counting to Understanding:** We transitioned from the "Old Way" of counting words (TF-IDF/Bag-of-Words) to the "New Way" of using **LLMs** for Zero-Shot Feature Extraction, turning unstructured text into structured predictive features .
- Next Step: The Laboratory
  - **Objective:** Apply these concepts to build a robust **Churn Prediction Pipeline** using Polars, UMAP, and Hugging Face transformers .

The Laboratory Session
The Churn Pipeline

Duration: 2 Hours **Platform:** Google Colab **Data Source:** Telco Customer Churn (IBM) + Synthetic Text.

- **Lab Objectives** This lab combines standard curriculum requirements with cutting-edge industry practices.
  - **Benchmarking:** Empirically prove **Polars** superiority over **Pandas**.
  - **Robust Engineering:** Implement a pipeline that strictly prevents **Data Leakage** during scaling.
  - **Advanced Viz:** Use **UMAP** and **Plotly** for interactive high-dimensional visualization.
  - **AI Integration:** Use a Hugging Face **Zero-Shot Pipeline** to perform feature extraction on text.