Week 1 - Data Profiling and Cleaning

Dr. Smitha K G
Senior Lecturer,
College of Computing and Data Science,
Nanyang technological University

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Course Structure

5 Jan – 14 Feb 2026, 6 weeks

One pre-recorded lecture to watch before Tuesday class

Tuesday: Consultative format where you come with questions/doubts

Saturday: Lectures with hands on coding exercises (Bring your laptops and chargers). We will be using the python notebook in anaconda platform

Assessment: 2 Quizzes on week 3 and 6 and 1 assignment

Quiz 1 on 24/01/26 Sat, 9.00pm-10.30am (regular classes after that)

Quiz 2 on 10/02/26 Tue, 6.30 pm-20.00pm

Data science thinking process

Ask questions and define the problem.

Prepare data by collecting and storing the information.

Process data by cleaning and checking the information.

Analyze data to find patterns, relationships, and trends.

Share data with your audience.

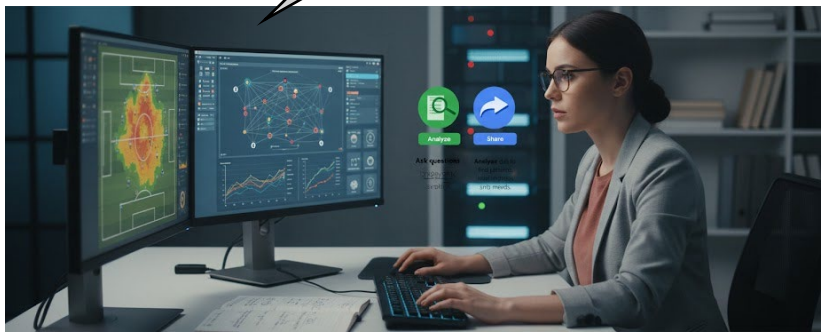
Act on the data and use the analysis results.

Problem: A football team **Phoenix FC** struggling with inconsistency.

How can my team get better?

Coach Davies: Experienced, intuitive, but increasingly open to new methods

Maybe I can help based on the data?



Dr. Anya : Brilliant, focused data scientist, recently brought in by the club.

Ask: Defining the cracks in the foundation

I need to know player speed and distance run, especially in the second half.

hit the target, but not the net. Where are our shots coming from?

we're giving the ball away too easily. let me know the pass completion rates, by player and by zone.

stupid free kicks… are we just ill-disciplined, or something else? I want to see fouls committed, their location, and who's making them

Ok. Noted. We're looking at **player endurance, possession efficiency, goal-scoring threat, and defensive discipline**

Prepare: Gathering the digital gold

Physical and mental fitness is needed. Let me work on strategies which can boost them

Data from old matches: match reports, referee reports, doctors' reports, GPS data from the players wearable trackers for location tracking

For shot locations, a new tagging system that categorized shots not just by on/off target, but by their expected goals value based on the position on the pitch.

Video analysis: To ensure every training session and match was meticulously tagged for player actions

Every pass, every sprint, every shot, every foul – it needs to be logged with precision including Player IDs, timestamps, field coordinates

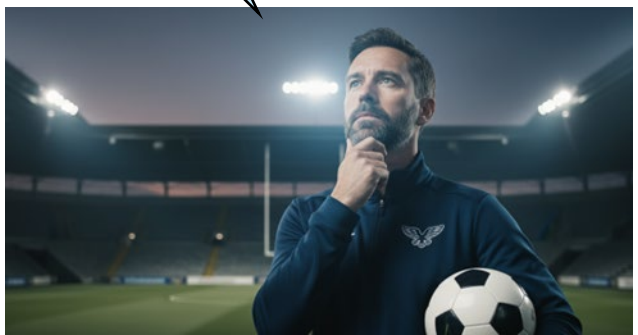
Pass completion was tracked not just overall, but broken down by short, medium, and long passes, and successful vs. unsuccessful.

Process : Polish the raw gem

"Player 7's GPS data shows him running 20km in one half," she muttered to herself, eyes narrowed.

Impossible to have 20km in one half… Some error

"Clear outlier. Filtering." Wrote scripts to identify and correct these anomalies..



Raw numbers into meaningful metrics.
Total distance run became 'average distance per 90 minutes'.
Fouls were normalized by 'fouls per defensive action'.
Shot locations were aggregated into 'shot heatmaps' and 'average xG' per shot'.

Raw data is messy. GPS trackers occasionally drop signals, human taggers make errors, and differing definitions can lead to inconsistencies.

Pass data needed to be cross-referenced with video to confirm accuracy. Some 'fouls' were logged as 'dispossessions' – need to standardize the data

Analyze: uncovering hidden truth

shot location analysis: Despite taking many shots, their low average xG per shot revealed a poor quality of attempts, taken mostly from tight angles or outside the penalty area.

Pass completion rates revealed an interesting pattern: while overall accuracy was decent, long passes from the defensive third into the attacking third had an alarmingly low success rate, often leading to immediate counter-attacks.

The **player speed and distance data** showed a significant drop-off in high-intensity sprints for the central midfielders and full-backs after the 70th minute in most games- **inform coach about fatigue**

Thanks, Anya. No surprises, but the data confirms the severity. Okay, let's move to action: What are the top three, non-negotiable drills we need to implement tomorrow morning to address the 70th-minute drop-off and the low xG shots?

fouls committed. a significant cluster of fouls is during transition moments after losing possession through those failed long passes, pointed to players being caught out of position and resorting to desperate measures when tired.

Share: Bridging intuition and evidence

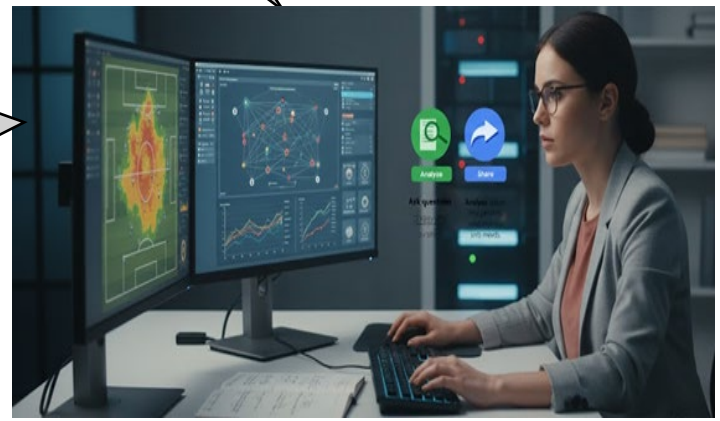
"And these fouls… they're not random, are they?"

shot location heatmap – a thick cluster outside the 18-yard box. "So much wasted effort.“. No wonder our average xG per shot is low

Possession: "our long passes from defense are a major vulnerability. Look at this heatmap showing where we lose the ball most frequently… often directly leading to opposition attacks."

"your instincts were spot on. We see a clear **drop in high-intensity efforts** from our key engine room players in the last 20 minutes, confirming your fatigue theory." She showed a graph of average sprint distance over 90 minutes.

"They correlate strongly with the fatigue window and those failed long passes, suggesting structural issues rather than just indiscipline."



Act: the road to transformation

For fatigue: They adjusted training regimens, focusing on endurance drills that mimicked late-game intensity. Substitutions became more data-driven, preemptively rotating key players to maintain high energy levels.
For pass completion: They implemented drills specifically for long-ball accuracy from the back, emphasizing when and when not to attempt them.
For shot locations: Attacking drills were redesigned to prioritize getting the ball into high-xG areas, encouraging players to make the extra pass rather than taking a low-percentage shot.
For fouls: Defensive shape during transitions was drilled relentlessly, teaching players how to maintain compact lines even when tired, reducing the need for desperate fouls.

Picture credits @Google Gemini AI

Data science pipeline

Defining the problem

What type of task is this?

What type of data do we have?

Getting the data

Collecting new data

Data is relevant to your specific problem and high-quality

Data collection is time consuming and expensive(need experts to pay for labeling)

Or use existing data

Check quality, relevance to our problem

Any legal or ethical liabilities

Understanding and Preparing the data

Data structure: size of dataset, number of features/variables, type of data (numerical vs categorical vs other)

Data quality: missing values? outliers? errors?

Data distribution: range of values, patterns and trends in the data

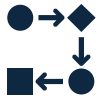
Data relrelationships: correlations between variables?

Cleaning (handling missing data, outliers, remove or correct errors)

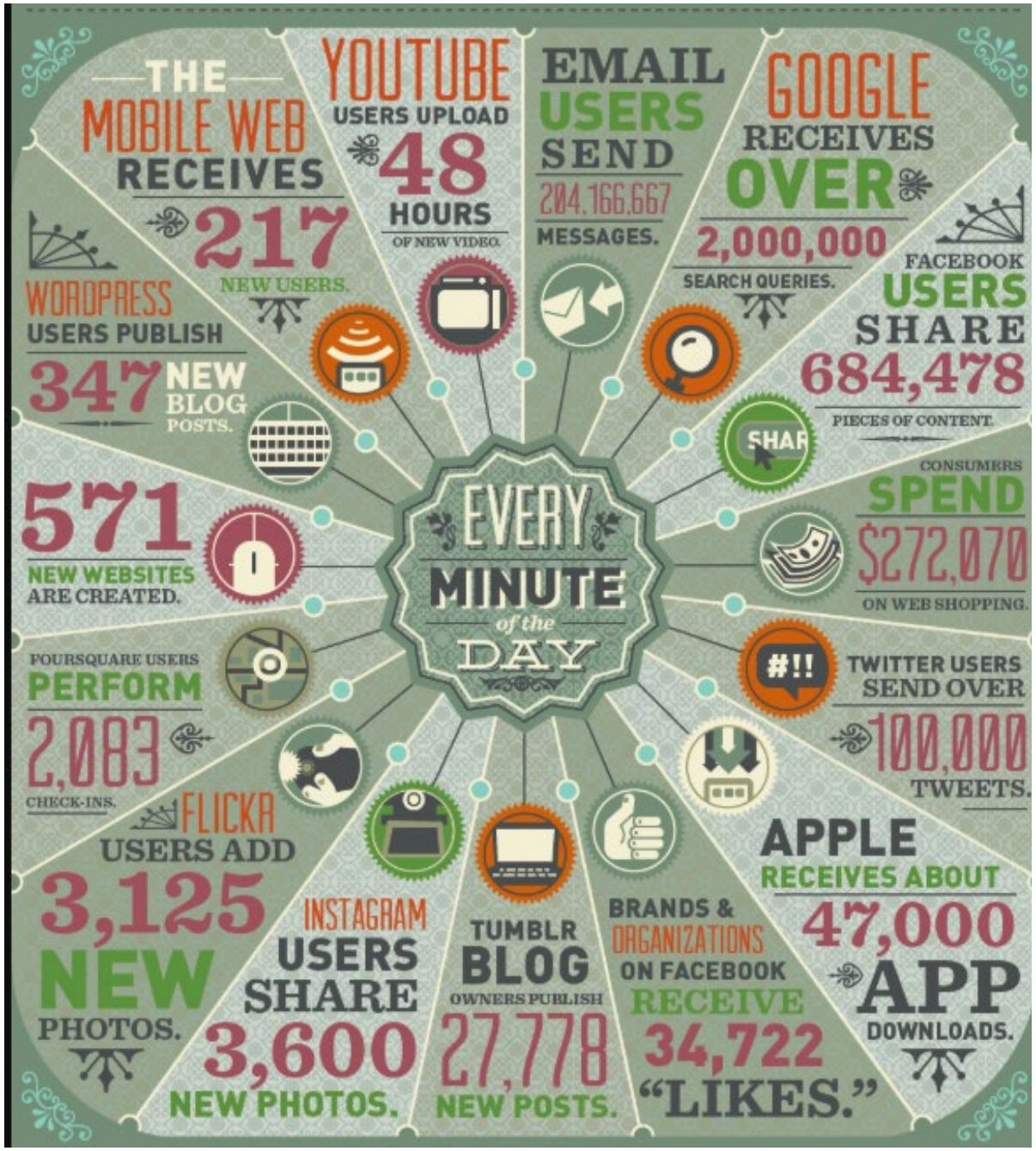
Transforming data (Normalization, encoding)

Feature engineering

Training the model



Evaluating, Deploying and Maintaining the model



The Rise of Data

World	**8 Billion**
Internet	**5.65 Billion**

YouTube	**500 hrs uploaded**
Facebook	**1.7M content shared**
Gmail	**231M emails sent**
Instagram	**66K photos shared**
Twitter	**347K tweets**

The figures state active users per month

Data Never Sleeps 10.0 | Domo

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75.0	7.2

Safety	Doors	Seats	Condition
high	4	2	unacc
med	5more	more	good
high	5more	more	vgood
high	2	2	unacc
high	2	2	unacc
low	4	more	acc

Car Evaluation dataset from ISL by James et al.

Structured Data

Categorical/ Mixed Data

Highly Organized Data Clearly Defined Variables Easy to Mine and Analyze

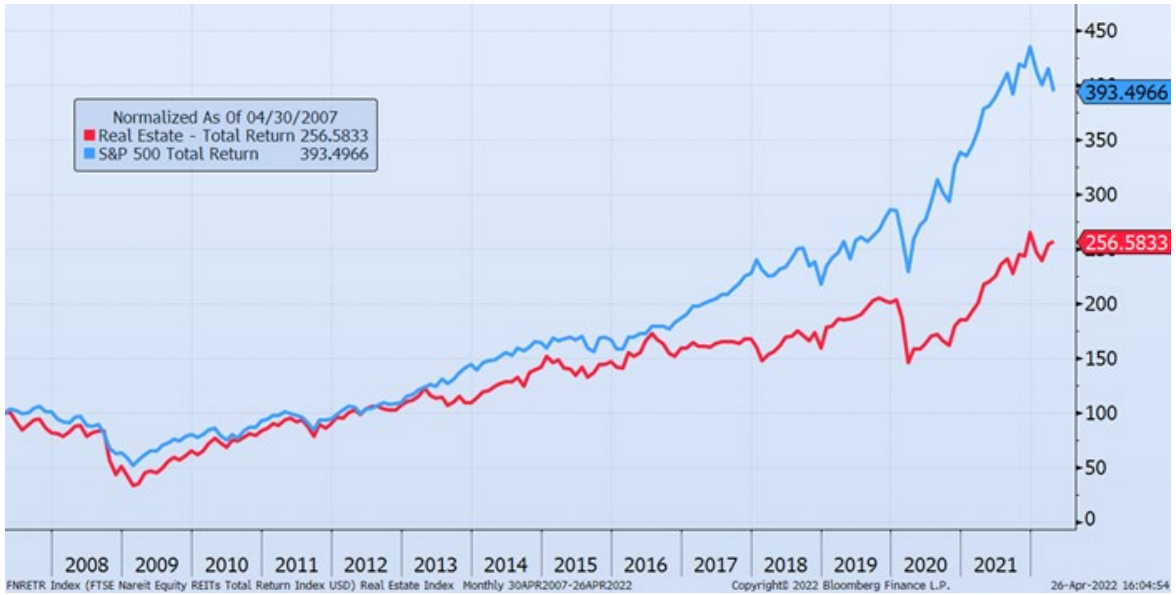
Factor/Level/Class Variables

Example Source

- Spreadsheets (Excel, CSV)
- Sensors and Devices
- Sales transactions (product ID, category, price, date)
- Patient demographics (age, gender, weight, diagnosis code)
- Transportation-Ride-sharing trip data (pickup, dropoff, duration, fare)

Structured Data

Time Series Data



House Prices vs. Stock Data from Bloomberg

- Highly organized data with time stamp
- Clearly defined with regular intervals (consistent intervals)
- Chronological dependency(values depend on previous time steps)
- Easy to mine and analyze(each time point can have multiple features or measured variables –temperature , humidity)

Example Source

- Stock and Equity Markets
- Weather Data over Time
- Prices and Promotions

Structured Data

Singapore MRT Network from MRT Website

Network Data

- **Fixed Schema (Nodes and Edges):** Network data is fundamentally organized into two fixed components
  - ✓ **Nodes (Vertices):** The entities (e.g., a person, a train station). Each node has defined, structured attributes
  - ✓ **Edges (Links):** The connections between the nodes (e.g., "is friends with," “connects between"). Each edge has defined attributes
- **Tabular Storage:** Network data is often stored and managed using tabular formats (like relational databases or specialized graph databases), meaning it has a defined column structure.

Example Source

- Social Networks and Web
- Transport Networks (MRT)
- Financial Transactions
- Supply chain (logistics)

Examples for structured network

Network Type	Node (Entity)	Edge (Relationship)	Structured Attributes (Data Points)
Social Network	Users (e.g., John, Jane)	"Is Friends With," "Follows"	Node Attributes: Age, Location, Posts Count. Edge Attributes: Date followed, Duration of friendship.
Logistics/Supply Chain	Warehouses, Suppliers	"Ships To," "Supplies"	Node Attributes: Capacity, GPS Coordinates. Edge Attributes: Shipping cost, Route ID, Estimated travel time.
Cybersecurity/IT	Computers, IP Addresses	"Communicates With," "Pings"	Node Attributes: Operating System, User ID. Edge Attributes: Timestamp, Port Number, Data Volume Transferred.
MRT system	MRT Stations	“Tracks," “Segments“connecting two stations	Node Attributes: Station code, Line, Interchange status Edge Attributes: Distance, Travel time, Cost and Direction
Academic Citations	Research Papers	"Cites," "References"	Node Attributes: Author, Publication Year, Journal. Edge Attributes: Citation type (e.g., Primary, Secondary).

Unstructured Data

Text Data

- High Dimensionality (The Curse of Vocabulary)
  - Example: If your model sees the words "run," "running," "ran," and "runner," it treats them as four separate features, even though they all relate to the root concept "run.“
  - Challenge: This vast number of features slows down training and makes models hard to interpret, requiring techniques like Dimensionality Reduction (e.g., embedding).
- Context Dependency (Meaning is Not Local)
  - Example: "The music has too much bass." (Sound frequency)"He caught a huge bass." (Fish)
  - Challenge: Traditional ML models struggle with this context, requiring advanced techniques like Transformers and Word Embeddings (e.g., Word2Vec, BERT) that capture meaning based on neighborhood.
- Ambiguity and Variability (The Messiness of Humans)
  - Example: "That product is great" (Positive)"Great service... NOT." (Negative: Sarcasm)
  - Challenge: Data preparation for text requires extensive cleaning steps like Lemmatization (reducing words to their base form), Stop Word Removal (deleting common words like "the," "a"), and sophisticated Tokenization to handle these variations.
- Sequence Matters (The Order of Operations)

Unstructured Data

Image Data

- Extreme High Dimensionality(large number of pixels)
  - Challenge**:** Trying to run traditional algorithms on 150,000+ independent features is computationally prohibitive and ineffective. CNNs solve this by using convolutional filters to learn local patterns (like edges and shapes) instead of treating every pixel individually.
- Spatial correlation (proximity context)
- Semantic gap
  - Example: The task is to classify an image as "danger." The machine must learn that a specific arrangement of red pixels and text constitutes a "STOP sign" and that the sign means "Danger.“
  - Challenge: This is the goal of Computer Vision—to bridge the gap between low-level pixel data and high-level human semantic understanding.
- Invariance challenge(same object but different look)
  - Challenge: The model must achieve invariance (the ability to recognize the object despite transformation). This requires training on massive, highly varied datasets and using pooling layers in CNNs to summarize features while retaining their meaning.

Looks like good food! – from the Canteen

Unstructured Data

Video Data

- Dimensionality explosion (width x height x channels x time/frames and batches)
  - Challenge: The processing load is immense. Specialized models like 3D Convolutional Neural Networks (3D-CNNs) are used to analyze this spatio-temporal data cube.
- Temporal correlation (Sequence and action)
  - Challenge: Models must include memory (like Recurrent Neural Networks or LSTMs) to link the features extracted from Frame N to the features in Frame N+1.
- Modality fusion
  - Challenge: Requires complex multimodal fusion models that process the visual and audio data streams separately and then combine them to create a single, coherent prediction (e.g., detecting "aggression" in a crowd).
- Semantic event detection (moment location)
  - Challenge: This requires models capable of temporal segmentation and object tracking across many frames. This is when the goal is not to classify video but to localize a specific event.

Unstructured Data

Voice Data

- Sequential and time dependent waveform
  - Challenge: Like video and text, models need to maintain temporal dependency. This is handled by Recurrent Neural Networks (RNNs) or Temporal Convolutional Networks which look at segments of the waveform over time.
- The conversion of features (spectrogram:- Time , frequency and pitch)
  - Challenge: The data scientist must choose the correct feature extraction method (e.g., Mel-Frequency Cepstral Coefficients or MFCCs) to summarize the audio without losing critical information about the speech or sound event.
- Acoustic variability
  - Challenge: Speech recognition systems need to be invariant to speaker identity and emotion while still recognizing the phonetic content. This requires massive, diverse training datasets and specialized layers to normalize speaker characteristics.
- Noise and environmental interference
  - Challenge: The first stage of processing often involves aggressive noise reduction and filtering techniques to isolate the target speaker's voice from the environmental chaos, ensuring the subsequent machine learning model receives a clean signal.

Data Profiling (from slides)

Step / Discovery Type	Goal	Process	Typical Statistical / Analytical Tools	Typical Outputs
Structure Discovery	Verify schema, data types, formats	Validate structure, datatypes, patterns	Pattern analysis, data type inference, format checks, regex validation	Schema summary, type mismatches
Content Discovery	Assess data quality and value distributions	Examine actual values (Identifies missing values, anomalies and inconsistencies in data)	Descriptive statistics, frequency analysis, outlier detection, missing value analysis	Missing value , mean, std. dev., outliers
Relationship Discovery	Identify dependencies and correlations between attributes	Identify column/table relationships (Analyses connection between different data elements and tables)	Correlation coefficients, chi-square tests, covariance, association rule mining	Correlation matrix, key integrity

Very essential for ensuring data readiness for analysis and informed decision making

Why data profiling matters- Example

ID	Name	Age	Email	Country	Revenue	Product_ Views
1	John Doe	22	john@gmail.com	Singapore	500	15
2	Jane Lee		jane_lee@gmail.com	SG	120.5	8
3	Mike Tan	-5	mike_tan	Malaysia	85	25
4	John Doe	29	john@gmail.com	Singapore	500	15
5	Sarah Wu	35	sarah@yahoo.com	USA	900	5

Key Benefits:

- Improves data quality and reliability
- Supports better decision-making
- Prevents costly errors in analytics or reports
- Prepares data for cleaning and transformation

Column	Profile Result	What It Tells Us (structure discovery)
ID	Unique: 5 values	No duplicates. (df.count()=5)
Name	4 unique values, “John Doe” appears twice	Possible duplicate records.
Age	Range: -5 to 29; 1 missing value	Error (negative age) + missing data.
Email	1 invalid format (“mike_tan”)	Data quality issue — not a valid email.
Country	3 unique values (“Singapore”, “Malaysia”, ‘USA")	SG and Singapore – Same country

Structure helps identify errors such as missing ages or invalid emails

Content discovery

.min()	Detects low outliers	df['Age'].min()	The minimum age is -5, which is an invalid data point.
.max()	Finds the highest value	df['Revenue'].max()	The maximum revenue is 900.0
.describe() (Numeric)	Statistical Summary. Provides range, central tendency, and spread	df.describe()	The mean Age (21.5) is artificially low due to the -5 outlier. The 25%, 50%, and 75% quartiles help assess data distribution.
.describe() (Categorical)	Standardization Check. Provides unique counts and frequency of the top value	df[['Name', 'Country']].describe(include='object')	Country has 4 unique values but only 5 entries (SG vs. Singapore), showing a standardization issue that needs cleaning.

Statistic	ID	Age	Revenue	Product_Views
count	5	4	5	5
mean	3	21.5	421.1	13.6
std	1.58	16.9	348.69	7.33
min	1	-5	85	5
25%	2	17.75	120.5	8
50% (Median)	3	25.5	500	15
75%	4	31.5	500	15
max	5	35	900	25

Statistic	Name	Country
count	5	5
unique	4	4
top	John Doe	Singapore
freq	2	2

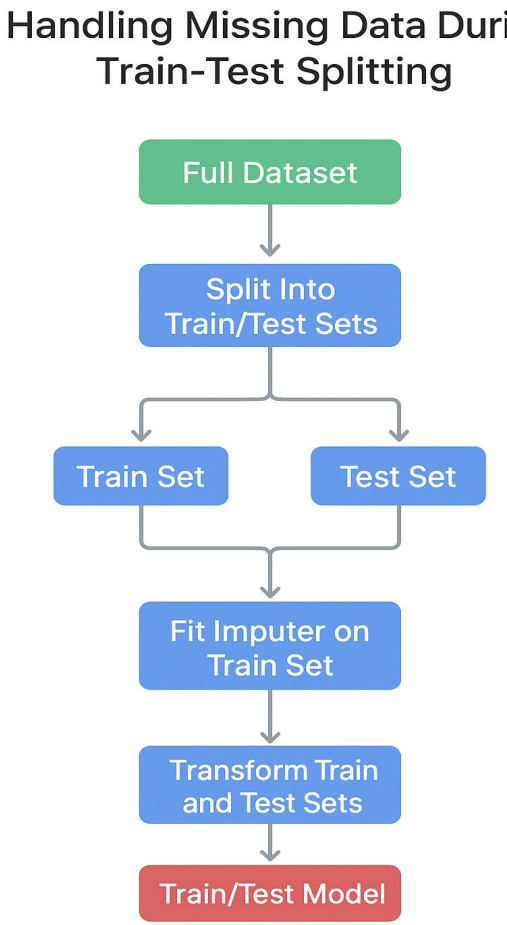
Relationship discovery

Function	Purpose in Relationship Discovery	Syntax Example	Key Insight
.corr()	**Measures co-variance.** Calculates the strength and direction of the linear relationship between pairs of numeric variables (e.g., Pearson's r).	df[['Age', 'Revenue', 'Product_Views']].corr()	A **strong negative correlation (-0.908)** between **Age** and **Product_Views** suggests that older customers view fewer products before purchasing.

ID	Name	Age	Email	Country	Revenue	Product_Views
1	John Doe	22	john@gmail.com	Singapore	500	15
2	Jane Lee		jane_lee@gmail.com	SG	120.5	8
3	Mike Tan	-5	mike_tan	Malaysia	85	25
4	John Doe	29	john@gmail.com	Singapore	500	15
5	Sarah Wu	35	sarah@yahoo.com	USA	900	5

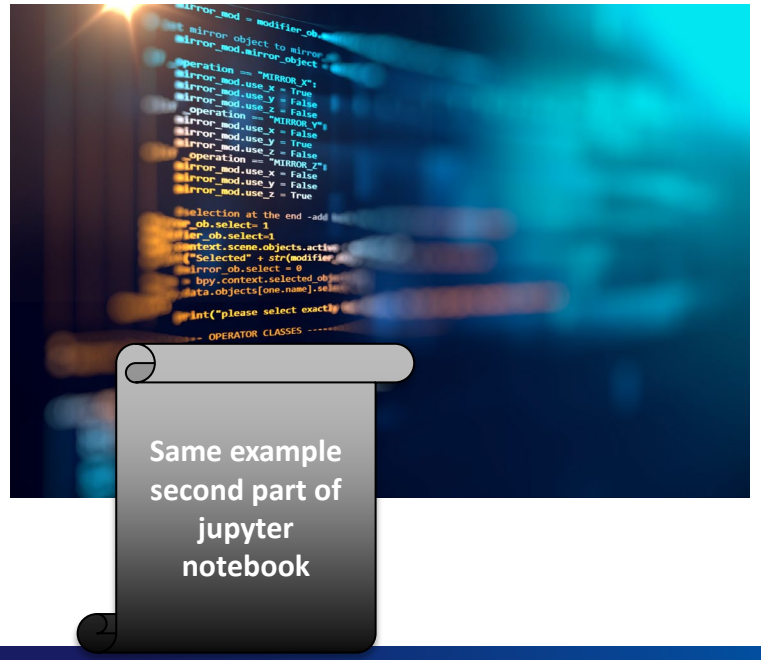
Golden rule in machine learning

- Never compute imputation values (mean, median, etc.) using the full dataset before splitting!
  - Because if you use all data to fill missing values, you are allowing information from the test set to influence the training process — this is data leakage- you may get optimistic results
  - Split the data into training and test sets
  - Fit the imputer only on the training set
  - Transform both training and test sets using that fitted imputer



Data cleaning

- Data cleaning (also called data cleansing or data scrubbing) is to ensure quality of data and for meaningful insights. Data cleaning is the process of detecting and correcting errors, inconsistencies, and inaccuracies in a dataset to ensure that it is accurate, consistent, and ready for analysis or machine learning. It usually follows data profiling and involves handling issues like:
  - Missing or incomplete values
  - Duplicated rows
  - Outliers or invalid values
  - Inconsistent data formats and units
  - Typographical or case errors
  - Encoded or garbled data
  - Truncated data
  - Unnecessary Metadata



Issue	Features / Description	Challenges	Example
Missing or Incomplete Values	Some fields are empty or partially filled.	Reduces accuracy; may bias models; requires imputation or removal.	Missing age or EEG segment.
Duplicated Rows	Identical records appearing more than once.	Inflates sample size, distorts analysis, misleading results.	Two identical emotional ratings.
Outliers / Invalid Values	Extremely high or low values outside valid range.	Hard to decide if real or error; may skew averages.	Heart rate = 900 bpm; negative reaction time.
Inconsistent Formats / Units	Data recorded using different formats or units.	Difficult integration; incorrect comparisons.	“37°C” vs “98.6°F”; “2025-02-01” vs “01/02/25”.
Typographical or Case Errors	Spelling or capitalization mistakes.	Causes grouping errors; not easily auto-corrected.	“Happy”, “happy”, “Hapy”.
Encoded / Garbled Data	Corrupted or wrongly encoded characters.	Loss of meaning; may need re-encoding.	“Ã©motion” instead of “émotion”.
Truncated Data	Information cut off due to storage or parsing error.	Partial records reduce validity; often unnoticed.	“P01” instead of “P01023”.
Unnecessary Metadata	Extra non-analytical fields.	Adds clutter, noise, and confusion.	“Last Edited By”, “Export Timestamp”.

Fitness data set example

Method / Toolh	Approach	Example / Usage
1. Identify Missing Values	Detect NaN or None values in datasets.	df.isnull().sum() or df.info()
2. Remove Missing Values	Drop rows/columns with missing data when data loss is minimal.	df.dropna(inplace=True)
3. Impute with Mean/Median/Mode	Replace missing values with statistical measures.	df['age'].fillna(df['age'].mean(), inplace=True)
4. Forward/Backward Fill	Fill using previous or next valid value (useful for time series).	df.fillna(method='ffill') / df.fillna(method='bfill')
5. Predictive Imputation	Use ML models (KNN, regression) to predict missing values.	from sklearn.impute import KNNImputer
6. Domain-based Imputation	Replace missing data based on expert knowledge (e.g., baseline emotion = neutral).	Manual or rule-based filling
7. Binning Imputation	Convert continuous values into bins (e.g., age groups) and impute using bin statistics (mode, median, or distribution).	df['age_bin'] = pd.qcut(df['age'], q=4)Then fill using df['age_bin'].mode()

Outlier detection

- Z score: Best for normally distributed data
- Calculates how far each point is from the mean in units of standard deviation.

$$Z = \frac{x - \mu}{\sigma}$$

- $|Z| > 3 \rightarrow$ Outlier
  (sometimes thresholds of 2.5 or 4 are used)
- **Advantages**
  - Simple to compute
  - Works well for symmetric / Gaussian data
- **Limitations**
  - Not reliable for skewed or non-normal data
  - Very sensitive to extreme values

- IQR: Best for non parametric data
- Uses quartiles Q1 = 25th percentile, Q3 = 75th percentile, IQR = Q3 - Q1
- A point is an outlier if it lies outside:

$$[Q1 - 1.5 \cdot IQR,\ Q3 + 1.5 \cdot IQR]$$

- **Advantages**
  - Robust against non-normal data
  - Not influenced by extreme values
  - Easy to visualize using box plots
- **Limitations**
  - Not ideal for **high-dimensional data**
  - Cannot capture complex patterns

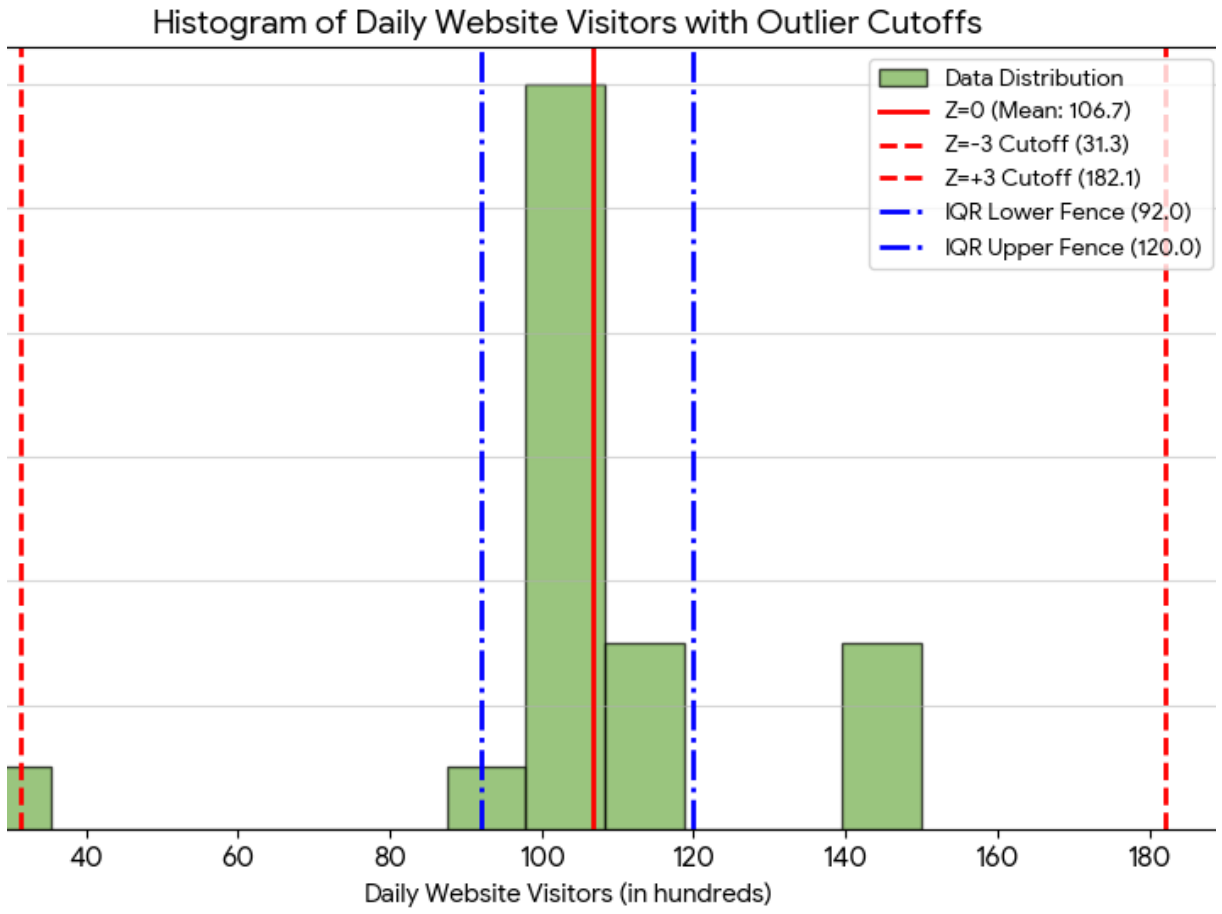
- Full Dataset (N=20):we are tracking daily website visitors (in hundreds) for 20 days. Most days are consistent, but there are a few spikes.
  25, 95, 100, 101, 102, 103, 103, 104, 104, 105, 105, 106, 107, 108, 109, 110, 112, 140, 145, 150
- Mean (μ)= $\mu = \frac{\sum x}{N} = \frac{2134}{20} = 106.7$
- **Standard Deviation (σ):**

$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{N}} \approx 25.12$$

- **Z-Score Outlier Lines:**
  - $Z = 0$ **Line (Mean):** 106.7
  - $Z = +3$ **Line (Upper Cutoff):** $\mu + 3\sigma = 106.7 + 3(25.12) = \mathbf{182.06}$
  - $Z = -3$ **Line (Lower Cutoff):** $\mu - 3\sigma = 106.7 - 3(25.12) = \mathbf{31.34}$

Outliers Detected: Only the value 25 is below the lower cutoff of 31.34.The Problem: None of the high values (140, 145, 150) are flagged as outliers because the extreme high values have inflated the mean and standard deviation.

- **Quartiles (Q_1, Q_2, Q_3):**
  - N=20. We find the 25th, 50th, and 75th percentiles of the sorted data.
  - Q_1 (25th percentile) is the average of the 5th and 6th values: (102 + 103) / 2 = 102.5
  - Q_2 (Median) is the average of the 10th and 11th values: (105 + 105) / 2 = 105.0
  - Q_3 (75th percentile) is the average of the 15th and 16th values: (109 + 110) / 2 = 109.5
- Interquartile Range (IQR}):
  - IQR = Q_3 - Q_1 = 109.5 - 102.5 = 7.0
- **IQR Outlier Fences**
  - **Lower Fence:** Q_1 - 1.5 IQR = 102.5 - 1.5(7.0) = 92.0
  - **Upper Fence:** Q_3 + 1.5 IQR = 109.5 + 1.5(7.0) = 120.0



IQR is robust and uses the middle 50% of the data, so the fences are not affected by the extreme values.

Isolation forest

- Instead of modeling normal data, it **isolates outliers**:
- Builds random trees, Outliers get isolated **quickly** because they are few and different
- Shorter path length → more likely an outlier
  - Large datasets
  - High-dimensional data
  - Complex, nonlinear patterns
- **Advantages**
  - Works extremely well with high-dimensional data
  - Captures nonlinear patterns
  - Fast, scalable, works for big data
  - Does NOT assume any distribution
- **Limitations**
  - Harder to explain than Z-score or IQR
  - Has hyperparameters (contamination rate)

Isolation Forest works by playing a game of "Twenty Questions" to find the odd ones out. The normal days (like 105 visitors) need many questions (splits) to separate them from their many neighbours. The odd days (like 25 or 150 visitors) are quickly isolated in just one or two questions because they have no close neighbours, making them the outliers.

The low outlier at 25 and the high outliers at 140, 145, 150 are far away from the main cluster of 95-112.

Method	Type	Best For	Weakness	Example Use
Z-Score	Parametric	Normal distributions	Fails on skewed data	Height, weight, test scores
IQR	Non-parametric	Skewed distributions	Weak for high-dim data	House prices, salaries
Isolation Forest	ML-based	High-dimensional, nonlinear	Needs tuning	Fraud detection, network anomalies

Few other outlier detection methods

- **Local Outlier Factor (LOF):**A density-based method that measures how isolated an observation is with respect to its surrounding neighbourhood.
  - LOF calculates a score by comparing the density around a point (A) to the average density around its nearest neighbours (B).If A's density is similar to B's, the LOF score is close to 1 (Normal), If A's density is much lower than B's, the LOF score is much greater than 1 (Outlier).
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** a clustering algorithm, but it intrinsically handles outlier detection by labelling points that do not belong to any cluster as "Noise.“
  - Define a neighbourhood radius and the minimum number of points required to form a dense region. Points in dense regions are classified as Core Points. Points near a dense region are Border Points. Points that are not core or border points are labelled as Noise (Outliers)
- **One-Class Support Vector Machine (OCSVM)** : specialized version of the Support Vector Machine (SVM) algorithm used for Novelty Detection, which is very similar to outlier detection
  - OCSVM draws a complex hypersphere (a boundary in high-dimensional space) that separates the vast majority of data points from the origin. Any new data point that falls outside this learned boundary is classified as an outlier.

Comparison: Outlier methods

Data Characteristic	Recommended Method(s)
Normal Distribution	Z-Score (Simple & Fast)
Skewed/Non-Normal	**IQR** (Robust) or **Isolation Forest** (Fast & Scalable)
Varying Density/Local Clusters	**LOF** or **DBSCAN**
High-Dimensional, Complex Data	**Isolation Forest** or **One-Class SVM**