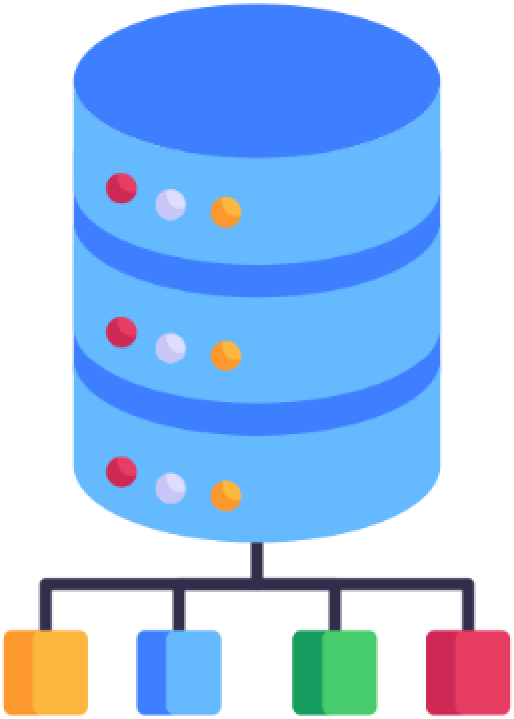
Exploratory Data Analysis (EDA)

Outline

- Features and Feature Types
- Dataset Types, Properties and Sources
- Data Exploration – Univariate
- Data Exploration - Bivariate
- Data Exploration - Multivariate

Features and Feature Types

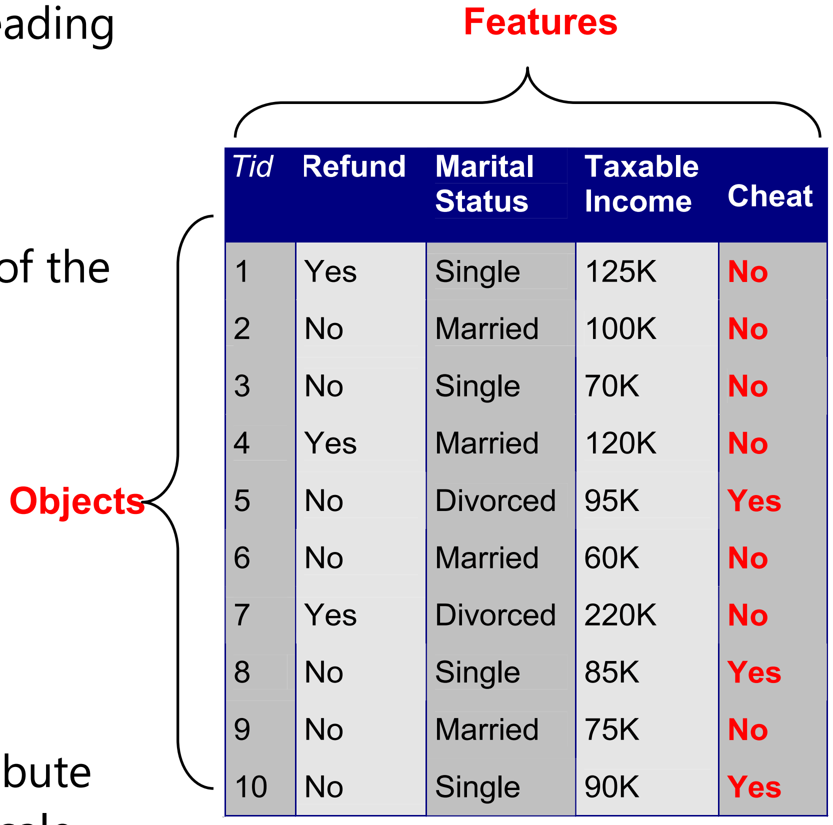


Steps for Doing Machine Learning

1. **Define Problem & Success Metrics**: Frame it as an ML task
2. **Acquire & Explore Data (EDA)**: Gather data and uncover initial insights
3. **Prepare Data & Engineer Features**: Clean, transform, and create predictive features
4. **Select Model & Establish Baseline**: Choose algorithms and a benchmark for comparison
5. **Train & Tune Model**: Train the model and optimize its hyperparameters
6. **Evaluate Final Model**: Test performance on unseen data against success metrics
7. **Deploy for Inference**: Integrate the model to make live predictions
8. **Monitor & Maintain**: Continuously track performance and retrain as needed

Anatomy of Data

- **Data object:** An individual sample or entity
  - e.g., a customer, a transaction, or a sensor reading
  - Also known as a record, instance, sample, or entity
- **Features:** A specific property or characteristic of the data object
  - e.g., Age, Income, Temperature
  - Also known as attribute, variable, field, or characteristic
- **Raw vs. Derived Features:**
  - **Raw:** collected or measured value of an attribute according to an appropriate measurement scale
  - **Derived:** constructed from data in one or more raw features (e.g., calculating "Debt-to-Income Ratio" from "Total Debt" and "Annual Income")



Features

Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Derived Features

- **Aggregates:** defined over a group or period, e.g., count, sum, average, minimum, or maximum of the values
- **Flags:** indicate presence or absence of some characteristic within a dataset, e.g., a flag indicating whether or not a bank account has been overdrawn
- **Ratios:** capture relationship between two or more raw data values, e.g., a ratio between a loan applicant’s salary and the amount for which they are requesting
- **Mappings:** convert continuous features into categorical features, e.g., map the salary values to low, medium, and high
- **Others:** no restrictions to the ways in which we can combine data to make derived features, e.g., use satellite photos to count the number of cars in the parking lots and use this as a proxy measure of activity within a competitor’s stores!

Goals for Derived Features

- To **improve** the accuracy and performance of ML models by transforming the raw data into a more meaningful representation that can better capture the underlying relationships in the data

- To help to **reduce** the dimensionality of a dataset and make it easier to visualize and understand the relationships between variables

Feature Types

Type	Subtype	Examples
Categorical (Qualitative)	**Nominal**	Product type, name
	Ordinal	Size measured as small<medium<large
	Binary	Spam email (yes/no, true/false, 0/1)
	Date / Time	Job start date
Numerical (Quantitative)	**Discrete** Countable values (integers)	Number of students in a class
	Continuous Measurable values (infinite decimals).	Height, weight, salary

Understanding the type of variables is crucial for selecting appropriate statistical methods, visualization techniques, and ML algorithms

Categorical Features

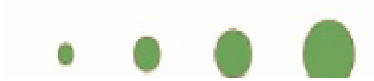
- **Categorical data are strings that represent qualitative data**
  - **Often selected from a group of categories, also called labels**
- **Nominal**, e.g., country of birth, gender, eye color, etc.

  – No inherent order or ranking

  – Operators applicable: **=, ≠**

  – 1:1 transformation permissible, e.g. ID: 974 ⇒ Qatar

- **Ordinal**, e.g. grade (A, B, C, D, F), degree (bachelor, master, PhD), height (tall, medium, short), etc.



  – Represent categories that can be meaningfully ordered

  – Operator applicable: **=, ≠, <, >, ≥, ≤**

  – Order-preserving transformation permitted,

  • e.g. height (tall, medium, short) to (1, 2, 3)