Getting started. Eda + prepproces

[Getting Started with Machine Learning: Exploratory Data Analysis & Data Preparation/Preprocessing]

ML pipeline summary:

1) Data collection
2) Data preprocessing
3) Feature engineering and dividing data into train,test set, validate set

4) Selecting the model and training the model

5) Model eval + hyper peramter tuning

6) Testing
7) Deploymenyt
8) Monitor + maintain

Rule of thumb: garbage in , garbage out, (no matter how good model)

Scaling law: more data, more resopurces MIGHT lead to better model performance #debateble

- EDA : exploratory data analysis #its more like an art than rigid science
- Data preprocessing and preparearion

Eda: basically it's a systematic n creative process of cleaning, inspecting , transformaing data to uncover insights , understand data, summarize and some vizualizing etc

1) Understand the data landscape, types, missing stuff and the distribution
2) Unwind patterns like correlation, trends
3) Detect anamolies and outlier
4) Then feature engineering
5) Final: formulate hypothesis (eg: I feel that sales will go up, or this product is bad ....)

##ask data the right questions, itll spit out the facts

CDA: directional version of EDA to confirm hypothesis

Data collection, after u have identified and defined the problem, collect the data, identify the source of data:

1) Primary (own survey, conduct experiment + observation) basically data creation
2) secondary sources (using available data, maybe govt resources, research data etc) by scrapping. Or just use API like kaggle, UCI repo, physioNet

Key Considerations for Data Collection

- **Data Privacy & Security**: Ensure compliance with regulations like **GDPR (General Data Protection Regulation)** to avoid legal issues.
- **No PII (Personally Identifiable Information)**: Avoid collecting sensitive user data without proper consent.
- **Data Quality**: Verify accuracy, consistency, and completeness before analysis.

Or u can get in legal trouble

#documenting EDA is a Must, itll be evry helpful later on

Data cleaning: handling missing values
- Deal with duplicates
- Fixing typos, inconsistent formatting
- Irregular units

Data transofrmation: (computer undestandsnumbers)

Handling diff data types

Image	pixel
Audio:	spectogram
Text:	tf idf, word2vec

Encoding categorical data

1) Label encoding:
   Converts each category into a unique number.
   - Example: ['Apple', 'Banana', 'Cherry'] → [0, 1, 2]
   - Works well for **ordinal** data (e.g., "low", "medium", "high"

2) One hot encoding; put 1 and rest 0:
   Basically using vectors (binary vectors)
   Best for **non ordinal data**

eg IRIS: setosa; [1,0,0] verginia: [0,1,0] virsica: [0,0,1]

Feature engineering:
Feature Scaling Techniques: Normalization, Standardization, and Log Transformation

1) Normalization (min max scaling) transforms values into a **fixed range**, typically **[0,1]** (or sometimes [-1,1]) using the formula: X' = X - Xmin / Xmax - Xmin
   Eg: blood pressure range is 80 to 300 but weifght is 40 to 10o usually, so its must be normalised so one feature does not become more ehavy
2) Standardization (z score normalization): Standardization transforms data so that it has a **mean of 0 and a standard deviation of 1** using the:
   x- mean/std deviation

   Log transformation is used to **reduce the impact of outliers** and **make skewed distributions more normal**. It is applied as:
   X'=log(X+1)
   (We add **1** to avoid issues with **log(0)**.)

What is Data Discretization?
Data discretization is the process of converting **continuous numerical data** into **discrete categories** or **bins**. It helps in:

- **Reducing complexity** by transforming continuous data into a finite set of intervals.
- **Enhancing interpretability** for decision trees and rule-based models.
- **Improving performance** in some machine learning models.

[Mastering Missing Values in ML: Listwise Deletion & Univariate Imputations | mean,median,mode,etc]

Handling missing values :
What is the source of this mising values? Is it faults during data colllection, data tempering?

Humans incredibly deal with missing values , like if we see tail of dog, we picture dog in brain, bnu tm,ost simple ML models do not have this ability unless explicitly trained

Why we must deal with missing values,:
We do not want our model to learn wrong or incomplete distribution

Task fo Machine learning is to generize not memorize

Why are these machines fast and efficient? Is it because of standardized representation? In contrast, human representation of data is not standardized at all—for example, the color red in my brain might be represented differently than in my friend's.

The difficulty in transferring brain data between people is **deeply tied to differences in data representation and distribution in each brain.,**

How Can We Get Data from the Brain?
Our brain communicates using **electrical and chemical signals**, and we have some ways to capture these signals:

- **EEG (Electroencephalography)**
  - Measures **electrical activity** in the brain using electrodes on the scalp.
  - Used for **brain-computer interfaces (BCIs)**, sleep studies, and seizure detection.
  - **Limitation:** Low spatial resolution (only gives a general idea of activity).
- **fMRI (Functional Magnetic Resonance Imaging)**
  - Measures **blood flow** in the brain to see which areas are active.
  - Used in neuroscience research and **mind-reading experiments**.
  - **Limitation:** Expensive, slow, and not real-time.
- **Neural Implants (Brain Chips)**
  - Tiny electrodes are implanted **inside** the brain to record activity at a much finer level.
  - Examples: **Neuralink (by Elon Musk), BrainGate, Utah Array.**
  - **Limitation:** Highly invasive—requires brain surgery!
- **Optogenetics (Experimental)**
  - Uses **light** to control neurons genetically modified to respond to it.
  - Used in research but **not yet for humans at scale**.

2. Can We "Inject" Brain Data into Someone Else?
Right now, **not fully**, but scientists are making progress!

- **Brain-to-Brain Communication (B2B) Experiments**
  - **Scientists have successfully linked two rat brains** so one rat's brain activity controlled another's.
  - In 2014, researchers did a **human-to-human brain interface** where one person moved another's hand **just by thinking** (via EEG + TMS).
- **Neuralink & BCIs for Thought Transmission**
  - Future brain chips **could** allow direct communication between humans, like "telepathy."
  - Some scientists think we could **download memories or skills** (like in The Matrix), but this is still **far away from reality.**

In Machine Learning (ML), we work with various types of data, including:

1. **Structured Data**
   - Data that is organized into a predefined format, typically in rows and columns (like a database or spreadsheet).
   - **Example**: Customer information in a table with fields like Name, Age, Salary, City.
2. **Unstructured Data**
   - Data that does not have a fixed format or organization. It is often rich in information but harder to process.
   - **Example**: Text from emails, images, videos, or audio recordings.
3. **Semi-Structured Data**
   - Data that has some structure but is not strictly formatted like relational databases. Often stored in formats like JSON, XML, or NoSQL databases.
   - **Example**: A JSON file with nested fields containing user details and preferences.
4. **Time-Series Data**
   - Data collected over time, where the order of data points is crucial. Often used for forecasting or trend analysis.
   - **Example**: Stock market prices, temperature readings over time, or website traffic logs.
5. **Ordinal Data**
   - Categorical data with a meaningful order, but the differences between categories are not necessarily uniform.
   - **Example**: Customer satisfaction levels (Low, Medium, High), or education levels (High School, Bachelor's, Master's, PhD).
6. **Categorical Data**
   - Data that represents discrete groups or categories, with no inherent order.
   - **Example**: Types of pets (Dog, Cat, Bird) or colors (Red, Blue, Green).

Each type of data requires different preprocessing techniques and ML models to extract insights effectively

Geo spatial features: when u are working with space for eg: logitudinal and latitudes data

Aspect	MCAR (Missing Completely At Random)	MAR (Missing At Random)	MNAR (Missing Not At Random)
Definition	Data is missing **purely by chance**, independent of both observed and unobserved variables.	The probability of missing data **depends on observed variables** but not on the missing values themselves.	The probability of missing data depends on the **missing values themselves** (unobserved variables).
Example	A survey respondent **randomly** skips a question because they got distracted.	In a health survey, **older** respondents tend to skip technology-related questions, but age is recorded.	In a mental health survey, people with **severe depression** are less likely to respond, and depression severity is missing.
Implication	No systematic bias; missing data does not affect conclusions.	Can introduce bias but is manageable using statistical techniques.	High risk of bias; missing values hold meaningful information.

- **MCAR is the best-case scenario** but rare. No bias is introduced.
- **MAR is more common** and can be corrected using proper statistical techniques.
- **MNAR is the most problematic**, as missing values contain meaningful information, requiring deeper analysis or additional data collection.

Nearly impossible to find which kind of missing it is,so we just to closest assumption
~ask domain expert

Sd.isnull().sum() #in ur notebook
Df.info()
#ipynb: bar chart for missing values, correlation and heatmap

#correlation does not alwqays necessarily mean causation
#see if correlation is due to missing values

Cleaning too little leads to noisy or incorrect results, but **over-cleaning** can cause problems, especially in real-time applications.

Data Latency (Slow Processing)

- In real-time applications (e.g., fraud detection, stock trading), **speed matters more than perfect data**

categories or bins. It helps in:

- **Reducing complexity** by transforming continuous data into a finite set of intervals.
- **Enhancing interpretability** for decision trees and rule-based models.
- **Improving performance** in some machine learning models.

Active Learning is a **semi-supervised machine learning technique** where the model selectively **chooses which data points** should be labeled by a human expert to **improve model performance with less labeled data**.

How It Works:

1. **Model is trained on a small labeled dataset**.
2. The model **identifies uncertain/uninformative samples**.
3. These samples are **sent to a human expert** for labeling.
4. The newly labeled data is added to the training set, and the model is retrained.
5. The process repeats until the model reaches the desired accuracy.

Example:

Imagine you're training an image classifier for **cats and dogs** but only have labels for **10% of images**. Instead of randomly labeling more images, Active Learning helps you **label only the most useful images**—ones where the model is uncertain (e.g., blurry images, weird angles).

Use Case:

- When **labeled data is expensive** or time-consuming to obtain.
- Used in **medical imaging, speech recognition, NLP** where expert labeling is needed.

#see if correlation is due to missing values

Cleaning too little leads to noisy or incorrect results, but **over-cleaning** can cause problems, especially in real-time applications.

Data Latency (Slow Processing)

- In real-time applications (e.g., fraud detection, stock trading), **speed matters more than perfect data**.
- Too many cleaning steps (e.g., complex outlier removal, multiple imputation techniques) **increase processing time**, slowing down decision-making.

☑ **Solution:** Use lightweight cleaning methods like **mean/mode imputation** instead of complex algorithms.

Data Drift and Changing Patterns

- In real-time systems (like recommendation engines), data patterns change **constantly**.
- Over-cleaning based on old rules (e.g., fixed thresholds) **may discard useful new trends**.

☑ **Solution:** Use **adaptive cleaning** with models that update dynamically.

Computational Cost

- Real-time systems **must process data quickly**.
- Complex cleaning steps (like deep-learning-based imputations) **consume CPU/GPU resources**, making systems inefficient.

☑ **Solution:** Use **simpler, faster techniques** (e.g., forward fill, interpolation) for missing values.

CCA: **Listwise deletion (also called Complete Case Analysis, CCA)** is a method for handling missing data in machine learning and statistics. It **removes any row (data instance) that has missing values** in one or more columns.

When u are deleting data point or sampling, we must make sure that it is properly representing the distribution of original data, or else thing go in waste\

- If you remove too many values **from a specific group**, your dataset no longer reflects reality.
- Example: If missing values are mostly from **older patients**, removing them might make your dataset **skew younger**, leading to biased healthcare predictions.

1 Univariate Imputation (Single Variable)

✔ **Definition:**

- **Uses only one column** (the one with missing values) to fill in missing data.
- **Ignores relationships** with other columns.
- Simple but might not capture dependencies between features.

✔ **Common Methods:**

- **Mean/Median/Mode Imputation** (for numerical data)
- **Most Frequent Category** (for categorical data)
- **Forward Fill / Backward Fill** (for time series)
- **Arbitary or random value**

2 Bivariate Imputation (Two Variables)

✔ **Definition:**

- Uses **two variables** to fill in missing values.
- Accounts for relationships between features.
- **More accurate** than univariate imputation if the variables are strongly correlated.

✔ **Example Methods:**

- **Linear Regression Imputation** → Predict missing values based on another variable.
- **KNN Imputation** → Finds similar data points using two variables

Is Interpolation Always Bivariate?

✓ **Mathematically (Newton's, Lagrange, etc.)** → **Yes, interpolation is bivariate or multivariate because it requires x-values to estimate y-values.**

✓ **In Data Science (missing value imputation)** →

- If we interpolate **a single column over time** → People call it "univariate" (but technically it's still functionally bivariate).
- If we interpolate **one feature using another** → Clearly bivariate.