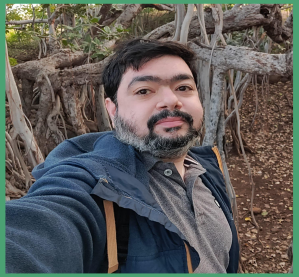
श्रवण,मनन,निदिध्यासन

Data Preparation

(Label Encoding, Column Transformer, Scaling,...)



Vaibhav Zodge

7020616260

info.dnyanyog@gmail.com

https://www.dnyanyog.org

https://github.com/zodgevaibhav

Problems with Data ?

Given RAW data may not be direct fit to train the model, because, it might…

- Contains missing values
- Has categorical features (text, labels, categories, *Refer slide* ***Types of Features***)
- Has different scales (salary in lakhs vs. age in years)
- Has irrelevant or redundant columns

If we don't prepare data properly, the model:

- Might misinterpret categories as numeric orders (label encoding issue)
- Could get biased toward features with large scales
- May perform poorly or fail to converge

Data Pre Processing

Handling Missing Data

Models can’t handle NaN or NULL values.

Drop missing rows/columns (if few).

Impute values (mean, median, most-frequent, or ML-based imputation).

Ref: https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/3.DataImputing.py

Encoding Categorical Variables

Models needs number and they don't work with Text

Label Encoding – assigns numbers (Toyota=0, BMW=1). Works for ordinal features.

Ref: https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/1.LabelEncoding.py

One-Hot Encoding – creates binary columns (Toyota=1, BMW=0). Works for nominal (unordered) features.

Magnitude of Categorical Labels may create problem. Ex. Model might try to relate 1 & 0 as weight or importance

Hence need to find way that model will not get the encoded number in the considerable format

Ref: https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/2.2.OneHotEncoding.py

Target Encoding / Frequency Encoding – replaces with target mean or frequency.

Feature Scaling

Some models (like Linear Regression, KNN, Neural Networks) are sensitive to feature scales.
Meaning, age = 20 & Salary=30,000, due to “Salary having numerical large size, model may thing Salary more important

MinMax Scaler - MinMaxScaler (Normalization) : Scales data to a fixed range - usually 0 to 1
StandardScaler (Standardization) : Scales data to have mean=0 and variance=1
Which means it centers the data around 0 and scales it based on standard deviation. Example : If mean=50 and std=10, then value 60 will be transformed to (60-50)/10 = 1

Ref : https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/4.DataScalingMinMax_Standard.py

Data Pre Processing

Handling Missing Data

Models can’t handle NaN or NULL values.

Drop missing rows/columns (if few).

Impute values (mean, median, most-frequent, or ML-based imputation).

Ref: https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/3.DataImputing.py

Data Pre Processing

Splitting Data

- A model is first trained on the training dataset.
- After training, we need to evaluate its performance using data where the correct outputs (labels) are already known.
- To achieve this, we split the available dataset into two parts:
- Training set – used by the model to learn patterns.
- Testing set – used to check how well the model performs on unseen data.
- This ensures that we are not just memorizing the training data but actually generalizing to new data.
- Sometimes, we also use a validation set (or cross-validation) for hyperparameter tuning before testing.
- The key idea is: Train → Validate → Test to build a reliable model.

Ref: https://github.com/zodgevaibhav/gen-ai-learning/blob/main/1.1.SupervisedLearning/1.RegressionAnalysis/2.2.DataPrepationExample/5.TrainTestData.py

Others

Remove redundant, irrelevant data

Remove vague data

Combine multiple data (if relevant)

Balance data (reduce bias or dominant data)

Types of Features

Feature Type	Subtype	Description	When to Use / Problem it Solves
Categorical	**Nominal**	Categories with no order. Ex: City = {Paris, Tokyo}	Used when classes are just labels. Need One-Hot encoding to avoid false order.
	Ordinal	Categories with natural order. Ex: Rating = {Poor < Good < Excellent}	Useful when rank matters but distances are not equal. Use Label/Ordinal encoding.
	High-cardinality	Many unique categories (e.g. `zipcode`, `user_id`)	Handle with target encoding, embeddings, or frequency encoding to avoid sparse features.
	Multi-label	One record belongs to multiple categories. Ex: Movie genres = {Action, Comedy}	Used when multiple classes apply simultaneously. Requires multi-hot encoding.
Numerical	**Discrete**	Countable integers. Ex: `number_of_children`	Useful for count data, often modeled with Poisson/Count models.
	Continuous	Values within a range. Ex: `height = 170.2 cm`	Standard ML input. Often scaled (Standard/MinMax) for algorithms sensitive to magnitude.
	Interval	Continuous values without true zero. Ex: `year`, `Celsius temp`	Differences matter, but ratios don't. Handle carefully in scaling.
	Ratio	Continuous values with true zero. Ex: `weight`, `salary`	Ratios are meaningful, scaling often applied. Standard in regression.
Binary	**Boolean**	{True, False} or {0,1}. Ex: Clicked Ad = {Yes, No}	Directly used as 0/1. No special encoding needed.
	Dichotomous Categorical	Two categories like {Male, Female}	Can map to 0/1. Often merged into binary features.

Types of Features

Text	**Unstructured**	Raw sentences. Ex: “I love this product!”	Requires NLP preprocessing: tokenization, cleaning.
	Structured tokens	Bag-of-Words, TF-IDF	Converts text into numeric vectors. Good for ML models.
	Embeddings	Dense representation (Word2Vec, BERT)	Captures semantic meaning of text for deep learning/NLP tasks.
Date/Time	**Date only**	Calendar date. Ex: `2025-08-30`	Use for seasonal analysis, trends. Extract year/month/day.
	Time only	Clock time. Ex: `14:25:33`	Useful in patterns like business hours, peak times.
	Datetime/Timestamp	Full datetime value	Use to derive cyclical features (hour, weekday).
	Cyclical features	Transform periodic data (e.g., sine/cosine for hours)	Helps ML understand circular data (e.g., 23:00 close to 01:00).
Derived / Engineered	**Mathematical transformations**	log(x), sqrt(x), polynomial terms	Used to stabilize variance, handle skewed data.
	Aggregates	Moving avg, rolling sum, cumulative count	Used in time-series, finance, sensor data.
	Interaction features	Feature1 × Feature2	Used when relation between features is important.
Special Modalities	**Image**	Pixels, CNN embeddings	For computer vision problems.
	Audio	Spectrogram, MFCCs	For speech/music recognition.
	Time-series / Sensor	Sequential signals (IoT, ECG, stock prices)	Used in forecasting, anomaly detection.

DNYANYOG EDUCATION