DATA PREPROCESSING

SAMIA RAHMAN

WHAT IS DATA PREPROCESSING?

Transforming raw data->usable format for machine learning training

MISSING VALUE HANDLE

Handling Missing Data

1	size(sqkm)	no_of_bedrooms	price
2	2000	4	10000000
3	3000	NaN	23
4	1500	0	7300452
5	20	5	1222454
6	980	3	5000000
7	nan	2	4000000
8	1250	4	6783421
9	1500	15	7366452
10	23000	3	150000000
11	2000	-4	4590000
12	3500	5	25000000
13	unknown	2	4000000
14	2350	3	16000000
15	1530	3	10000000

House_price.csv

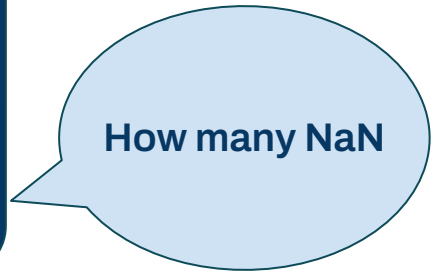
Handling Missing Data

size(sqkm)	no_of_bedrooms	price
2000	4	10000000
3000	NaN	23
1500	0	7300452
20	5	1222454
980	3	5000000
nan	2	4000000
1250	4	6783421
1500	15	7366452
23000	3	150000000
2000	-4	4590000
3500	5	25000000
unknown	2	4000000
2350	3	16000000
1530	3	10000000

Handling Missing Data

First comes check

```
import pandas as pd
pd.read_csv("House_price.csv)
Print(df.isnull().sum())
```



Handling Missing Data

Customized check

```
missing_placeholders = ["nan",
"unknown", 0, "n/a"]
df.replace(missing_placeholders, np.nan,
inplace=True) print(df.isnull().sum())
```

np.nan makes everything NaN

You can also use visualization techniques like heatmap, bar plot for missing data detection

Missing Data Detected

1. Deletion

```
df_cleaned = df.dropna()
```

	size(sqkm)	no_of_bedrooms	price
2	2000	4	10000000
3	3000	NaN	23
4	1500	0	7300452
5	20	5	1222454
6	980	3	5000000
7	nan	2	4000000
8	1250	4	6783421
9	1500	15	7366452
10	23000	3	150000000
11	2000	-4	4590000
12	3500	5	25000000
13	unknown	2	4000000
14	2350	3	16000000
15	1530	3	10000000

	size(sqkm)	no_of_bedrooms	price
2	2000	4	10000000
3	20	5	1222454
4	980	3	5000000
5	1250	4	6783421
6	1500	15	7366452
7	23000	3	150000000
8	3500	5	25000000
9	2350	3	16000000
10	1530	3	10000000

Missing Data Detected

2. Imputation

- Mean
- Median
- Mode
- KNN
- Regression
- Decision Tree

Missing Data Detected

Mean

Replace missing values with the mean of the column.

size(sqkm)	no_of_bedrooms	price
2000	4	10000000
3000	NaN	23
1500	0	7300452
20	5	1222454
980	3	5000000
nan	2	4000000
1250	4	6783421
1500	15	7366452
23000	3	150000000
2000	-4	4590000
3500	5	25000000
unknown	2	4000000
2350	3	16000000
1530	3	10000000

(4+5+3+2+4+15+3+5+2+3+3)/11 = 4.45 =4

size(sqkm)	no_of_bedrooms	price
2000	4	10000000
3000	4	23
1500	4	7300452
20	5	1222454
980	3	5000000
nan	2	4000000
1250	4	6783421
1500	15	7366452
23000	3	150000000
2000	4	4590000
3500	5	25000000
unknown	2	4000000
2350	3	16000000
1530	3	10000000

Missing Data Detected

Median

Replace missing values with the median of the column. $\text{Median} = \begin{cases} \text{Middle value (sorted data)} & \text{if odd number of values} \\ \frac{\text{Value at } n/2 + \text{Value at } (n/2+1)}{2} & \text{if even number of values} \end{cases}$

Mode

Replace missing values with the most frequent value in the column.

Regression

Predict missing values using a regression model trained on other features.

KNN

Fill missing values based on the nearest neighbors' values.

DT

Based on entropy value.

Missing Data Detected

Imputation suitability

- **Mean(**numerical data with no significant outliers**)**
- **Median(**numerical data, can handle outliers)

  The median represents the middle value of a sorted dataset. It divides the data into two halves, and it is not influenced by the magnitude of extreme values (outliers)
- **Mode(**categorical data**)**
- **KNN, Regression, DT(**both numerical and categorical data**)**

Missing Data Detected

Imputation (less common techniques)

- **Forward Fill (f-fill)**:
  - Use the previous value to fill missing data. Best for time series data where the previous value is a good estimate for the missing value.
- **Backward Fill (b-fill)**:
  - Use the next value to fill missing data. Suitable when future values are more relevant for filling missing data.

DATA TRANSFORMATION

Data transformation

Encoding:

Encoding is the process of converting categorical or textual data into numerical formats that machine learning algorithms can understand.

- **One-Hot Encoding**:
  - Converts categorical variables into binary vectors.
  - Example: "Red, Blue, Green" → `[1, 0, 0], [0, 1, 0], [0, 0, 1]`.
- **Label Encoding**:
  - Converts categories into numeric labels.
  - Example: "Red, Blue, Green" → `0, 1, 2`.
- **Ordinal Encoding**:
  - Assigns ordered integers to ordinal categorical data.
  - Example: "Low, Medium, High" → `1, 2, 3`.

One hot encoding

Name	Favorite colour
Rima	Red
Sima	Blue
Rahim	Green
Karim	Blue
Mehek	Red
Rain	Red
Nila	Green

Name	Favorite colour	Red	Blue	Green
Rima	Red	1	0	0
Sima	Blue	0	1	0
Rahim	Green	0	0	1
Karim	Blue	0	1	0
Mehek	Red	1	0	0
Rain	Red	1	0	0
Nila	Green	0	0	1

Label encoding

Name	Favorite colour
Rima	Red
Sima	Blue
Rahim	Green
Karim	Blue
Mehek	Red
Rain	Red
Nila	Green

Name	Favorite colour	Favorite colour new
Rima	Red	0
Sima	Blue	1
Rahim	Green	2
Karim	Blue	1
Mehek	Red	0
Rain	Red	0
Nila	Green	2

Label encoding vs Ordinal encoding

Label encoding	Ordinal encoding
No meaningful order such as colour red=0, blue=1, green=2	Meaningful order. Such as BSc=0, MSc=1, PhD=2
May introduce false relationships.	Reflects the ordinal relationship.

Key questions

1. Why not One-hot encoding always?
2. We can always use label encoding when we need label or ordinal encoding, so why the ordinal encoding required?
3. Code difference between label and ordinal encoding

Data transformation

Scaling

Scaling involves normalizing numerical data to ensure that all features contribute equally to the model. Features with larger ranges can dominate the model's calculations, leading to bias.

- **Normalization**:
  - **Mean normalization:**Scales data to a estimated range, typically [-1,1]

$$X' = \frac{X - \text{mean}(X)}{\max(X) - \min(X)}$$

  - **Min-max normalization:**Scales data to a fixed range, typically [0, 1]

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Mean Normalization

no_of_bedrooms
4
4
4
5
3
2
4
15
3
4
5
2
3
3

Here,
lowest range= min = 2
highest range = max = 15
selected x=5
mean=(4+4+4+5+3+2+4+15+3+4+5+2+3+3)/14 = 4.35
after performing mean max
x' = (x-mean)/(max-min)
= (5-4.35)/(15-2)
= 0.05

size(sqkm)	no_of_bedrooms	price
2000	-0.027472527	10000000
3000	-0.027472527	23
1500	-0.027472527	7300452
20	0.049450549	1222454
980	-0.104395604	5000000
nan	-0.181318681	4000000
1250	-0.027472527	6783421
1500	0.818681319	7366452
23000	-0.104395604	150000000
2000	-0.027472527	4590000
3500	0.049450549	25000000
unknown	-0.181318681	4000000
2350	-0.104395604	16000000
1530	-0.104395604	10000000

size(sqkm)	no_of_bedrooms	price
-0.059399478	-0.027472527	-0.052982295
-0.015883377	-0.027472527	-0.119648818
-0.081157528	-0.027472527	-0.070979284
-0.145561358	0.049450549	-0.111499277
-0.103785901	-0.104395604	-0.086315633
-0.103785901	-0.181318681	-0.092982301
-0.092036554	-0.027472527	-0.074426158
-0.081157528	0.818681319	-0.070539284
0.854438642	-0.104395604	0.880351182
-0.059399478	-0.027472527	-0.089048967
0.005874674	0.049450549	0.047017721
0.005874674	-0.181318681	-0.092982301
-0.044168842	-0.104395604	-0.012982289
-0.079852045	-0.104395604	-0.052982295

Need?

1. Centered around 0(mean=0)

★ Centering helps numerical optimization algorithms (like gradient descent) converge faster by starting closer to the minimum.

2. Helps the model make accurate predictions by avoiding biases toward larger-scaled features.

Mean vs Min-Max

Mean normalization	Min-Max normalization
Centered around 0	Not centered around 0
No fixed range. Depends on distribution of original data.If the data is symmetric around the mean, the range might be approximately $[-0.5,0.5]$. For skewed data, the range might extend further, e.g., $[-1,0.3]$	Fixed range $[0,1]$
used when data needs to be centered around 0. Such as gradient decent or PCA algorithm	Used when equal range for all column is important such as in neural network

Scaling

- **Standardization (Z-Score Scaling)**:
  - Scales data to have a mean of 0 and a standard deviation of 1.

Z: The z-score (standardized value). $$Z = \frac{X - \mu}{\sigma}$$

X: The raw data point (value to be standardized).

μ: The mean of the dataset.

σ: The standard deviation of the dataset.

Age	height (ft)
2	2
4	3
6	3
8	4
10	4

Apply standardization on age column.

→ I'll just determine the standardized value of 8 here.

$$\mu = \frac{2+4+6+8+10}{5} = 6$$

$$\sigma^2 = \frac{\Sigma(x_i - \mu)^2}{n} = \frac{(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-8)^2 + (10-6)^2}{5}$$

$$= 8$$

$$\sigma = \sqrt{8} = 2.83$$

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

$$= \frac{8-6}{2.83}$$

$$= 0.706$$

Normalization vs Standardization

Normalization	Standardization
Rescales data to a fixed range, typically [0,1].	Rescales data to have a mean of 0 and standard deviation of 1 (z-scores).
More sensitive	Less sensitive on outliers due to standard deviation scaling.
Use case: When algorithms requires fixed boundary such as gradient descent, sparse data(e.g: one hot encoded data)	Use case: When outliers are present in data, algorithm like SVM, logistic regression, PCA where its assumed that input features are centered around 0 and have a standard deviation of 1.

Problem with standardization

Can't handle sparse data well.
Sparse datasets often contain many zero values (e.g., one-hot encoded data).
Zero values (X=0) get transformed into $Z=-\mu/\sigma$, which makes the data dense rather than sparse. Sparse data loses its sparsity, increasing memory and computation requirements.

Sparse data often has specific meanings for non-zero values (e.g., word frequencies in TF-IDF matrices or one-hot encoded values).
Standardization distorts these relationships because it changes their magnitude relative to the zeros.

Key questions

1. Why normalization performs poor on outliers compared to standardization?
2. Why standardization doesn't work well with sparse data?

Self study

Robust Scaling: Outlier handling performance is better than previously discussed

OVERVIEW

Model name	Colour	Space(GB)	Response time	Price
Xiaomis42	Red	64	2	20000
Vivov2320	Blue	48	3	15000
Samsung7	Red	NaN	1	50000
iPhonei12	Blue	78	1	0
Xiaomis33	Unknown	-4	2	10000

Model name	Colour	Space(GB)	Response time	Price
Xiaomis42	Red	64	2	20000
Vivov2320	Blue	48	3	15000
Samsung7	Red	NaN	1	50000
iPhonei12	Blue	78	1	0
Xiaomis33	Unknown	-4	2	10000

Model name	Colour	Space(GB)	Response time	Price
Xiaomis42	Red	64	2	20000
Vivov2320	Blue	48	3	15000
Samsung7	Red	64	1	50000
iPhonei12	Blue	78	1	23000
Xiaomis33	Red	64	2	10000

Model name	Colour	Space(GB)	Response time	Price	Red	Blue
Xiaomis42	Red	64	2	20000	1	0
Vivov2320	Blue	48	3	15000	0	1
Samsung7	Red	64	1	50000	1	0
iPhonei12	Blue	78	1	23000	0	1
Xiaomis33	Red	64	2	10000	1	0

Model name	Space(GB)	Response time	Price	Red	Blue
Xiaomis42	64	2	20000	1	0
Vivov2320	48	3	15000	0	1
Samsung7	64	1	50000	1	0
iPhonei12	78	1	23000	0	1
Xiaomis33	64	2	10000	1	0

Space(GB)	Response time	Price	Red	Blue
64	2	20000	1	0
48	3	15000	0	1
64	1	50000	1	0
78	1	23000	0	1
64	2	10000	1	0

Space(GB)	Response time	Price	Red	Blue
0.53333	0.5	0.25	1	0
0	1	0.125	0	1
0.53333	0	1	1	0
1	0	0.325	0	1
0.53333	0.5	0	1	0

HOMEWORK

Perform missing value handling and data transformation on

Income.csv

Requirement

1. Provide the code
2. Provide the transformed csv
3. In the code for each action mention why you did that(e.g: I've applied standardization technique because dataset has no sparse data)

CONCLUSION

1

Handling missing data

- Check if has any
- If has, then delete or follow imputation techniques

2

Encoding

- One-hot encoding
- Label encoding
- Ordinal encoding

3

Scaling

- Mean normalization
- Min max scaling
- Z-score
- Robust scaling

THANK YOU!