DAY - 2

OVERVIEW

Preprocessing

Requirement of Preprocessing

Types of Preprocessing Techniques

Introducing Pandas and Handling Data

Handling Missing Values (Imputing) What is an Outlier?

Handling Categorical Variables (Encoding)

What is Scaling? and its Requirement

Working on Preprocessing Models

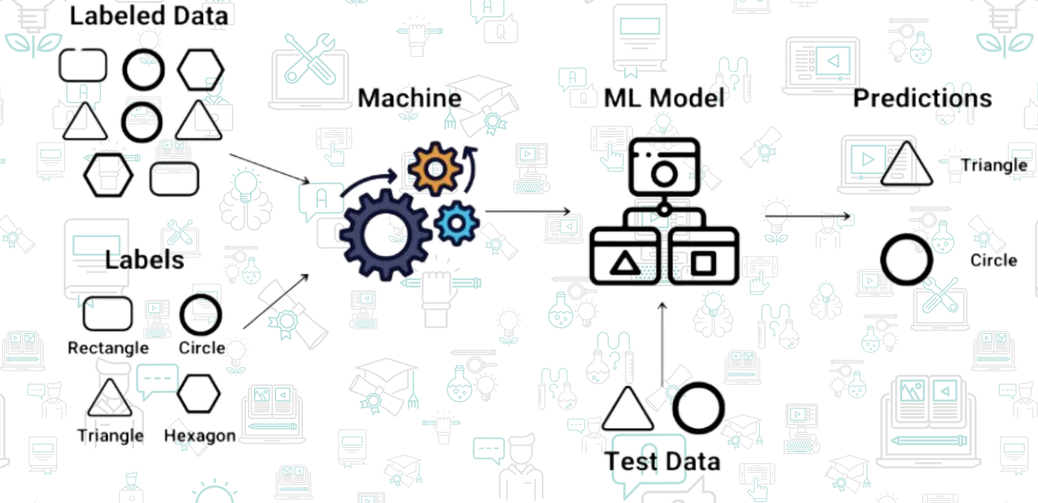
KNN

Decision Tree

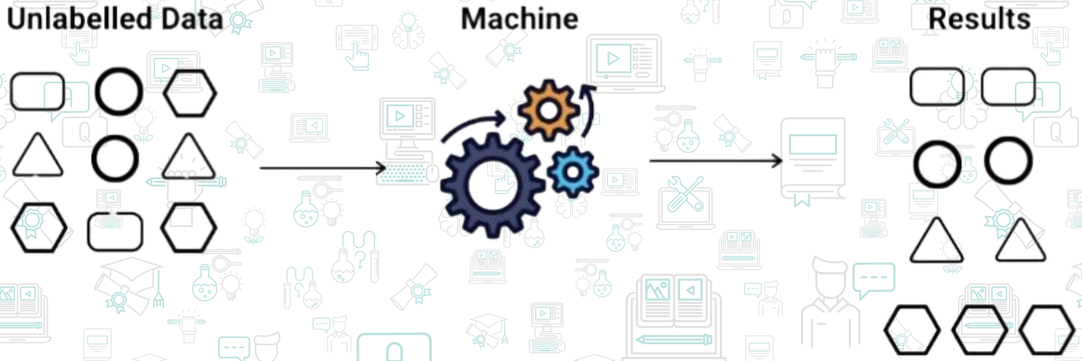
SUPERVISED AND UNSUPERVISED LEARNING

Supervised learning uses labeled data to train models, while **unsupervised learning** discovers patterns in unlabeled data

Supervised Learning



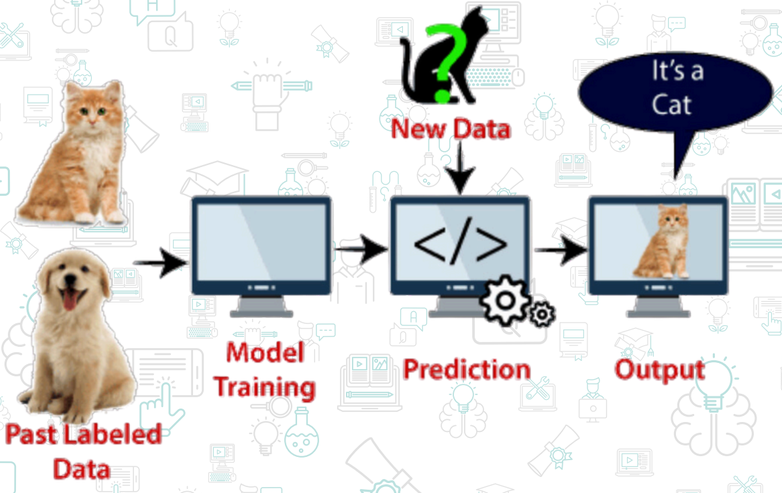
Unsupervised Learning



SUPERVISED LEARNING

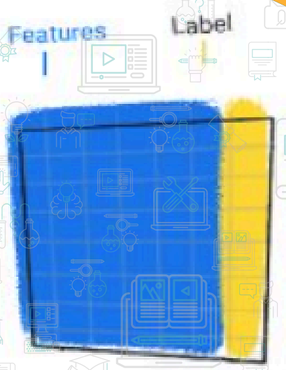
This type of learning trains datasets to teach models for the desired output

The model learns to map input data to known output labels by minimizing the error between its predictions and the actual outcomes



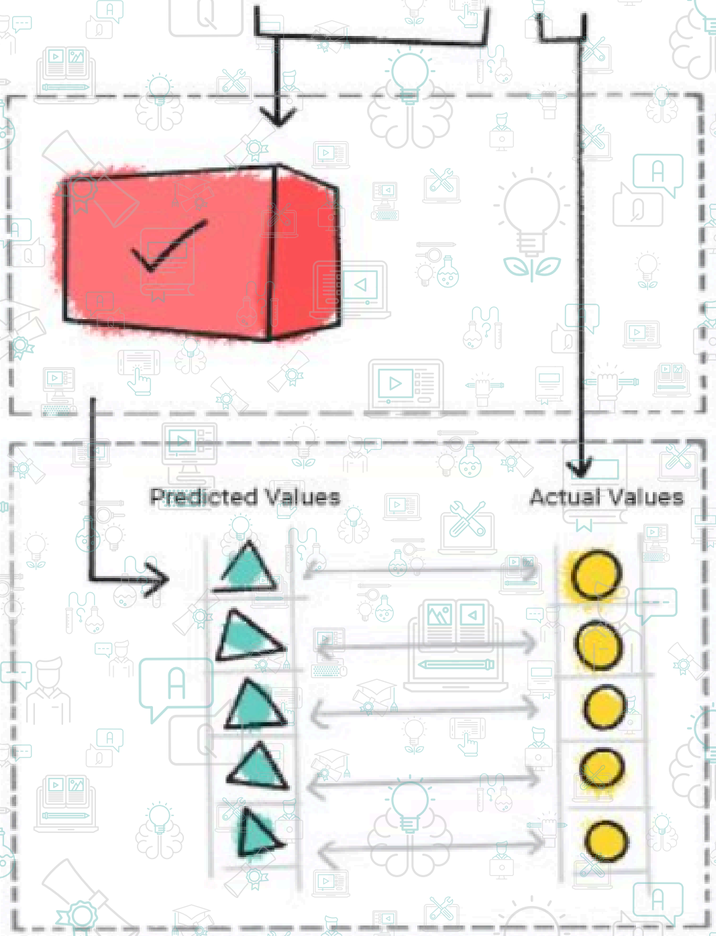
1

Use a **trained ML Model** to predict a value from a given dataset with labeled examples.



2

Compare **Predicted Values** with **Actual Values**.

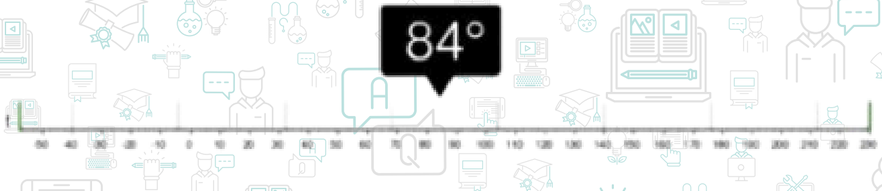


SUPERVISED LEARNING

WHAT ARE REGRESSION AND CLASSIFICATION

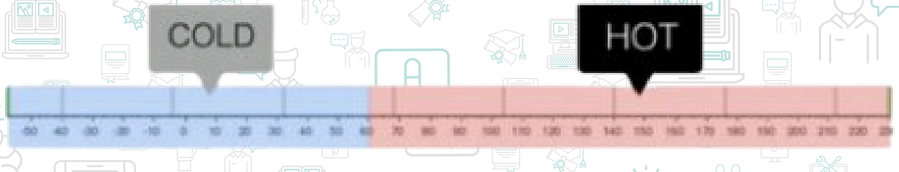
Regression

What will be the temperature tomorrow?

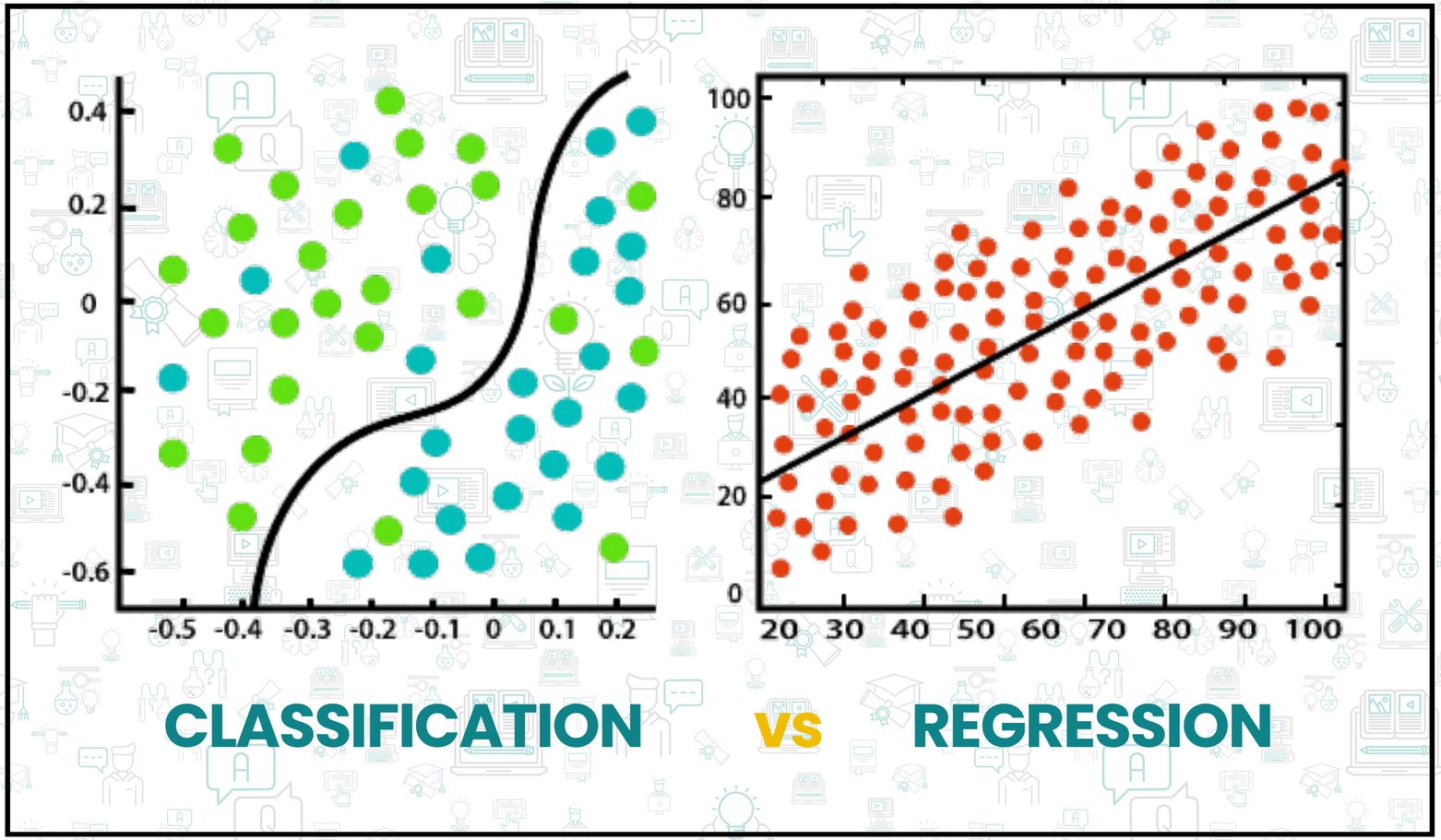


Classification

Will it be hot or cold tomorrow?



An Algorithm to accurately assign test data into specific categories



CLASSIFICATION vs REGRESSION

Understanding the relationship between dependent and independent variables.

PRACTICAL APPLICATIONS OF SUPERVISED LEARNING

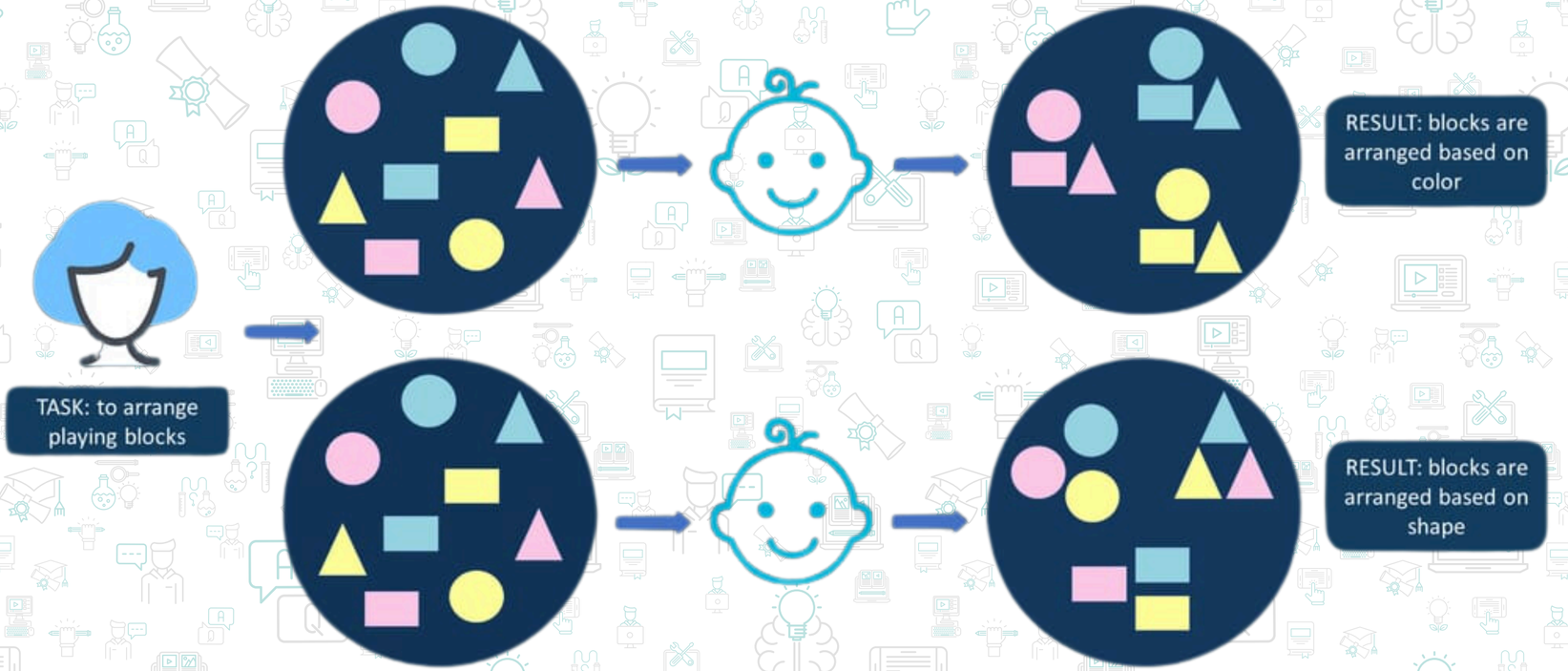
Image and Object Recognition
Predictive Analysis
Customer Sentiment Analysis
Spam Detection
Fraud Detection
Disease Diagnosis
Stock Price Prediction
Credit Scoring
Churn Prediction
Speech Recognition

CHALLENGES OF SUPERVISED LEARNING

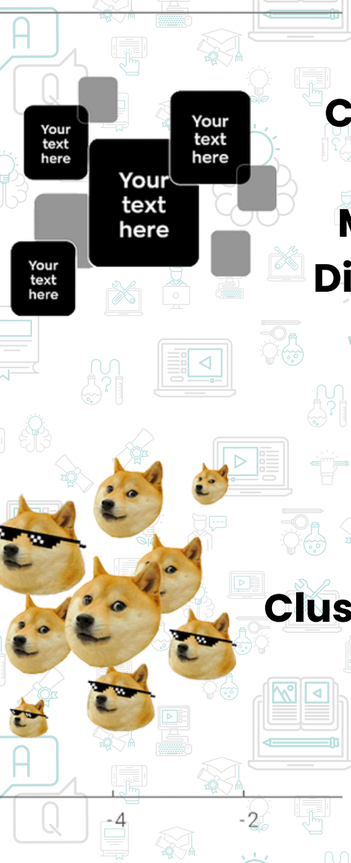
Expertise Required: Needs specialized knowledge
Time-Intensive Training: Takes considerable time
Human Error in Data: Errors affect learning
Labeled Data Dependence: Requires pre-labeled data
Scalability Issues: High resource demands

UNSUPERVISED LEARNING

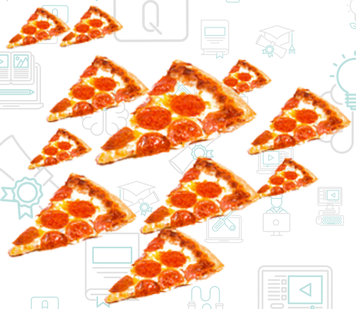
Explores datasets to uncover hidden patterns and relationships without predefined output labels. The model learns to identify structures and groupings in the input data by finding inherent similarities and differences within the data itself

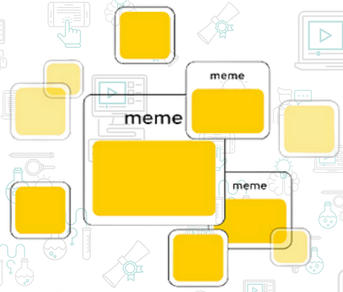


PRACTICAL APPLICATIONS OF UNSUPERVISED LEARNING



- Customer Segmentation
- Anomaly Detection
- Market Basket Analysis
- Dimensionality Reduction
- Topic Modeling
- Image Compression
- Feature Extraction
- Pattern Recognition
- Data Summarization
- Clustering for Data Exploration





CHALLENGES OF UNSUPERVISED LEARNING

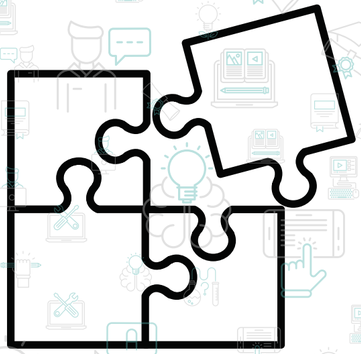
Evaluation Difficulty: Hard to assess performance

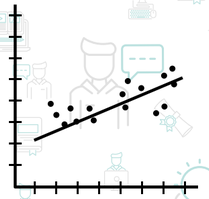
Interpreting Results: Patterns may be unclear

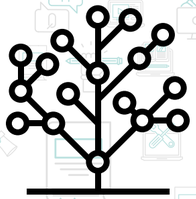
Algorithm Selection: Choosing the right method

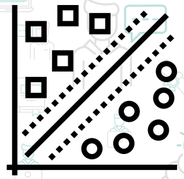
Scalability Issues: Struggles with large datasets

Sensitivity to Noise: Affected by data outliers









SUPERVISED CLASSIFIERS

Linear Regression - ALGEBRA

Logistic Regression - ALGEBRA

Decision Trees - INFORMATION GAIN / ENTROPY

Random Forest - PROBABILITY

Support Vector Machines (SVM) - GEOMETRY / OPTIMIZATION

k-Nearest Neighbors (k-NN) - DISTANCE METRICS

Naive Bayes - PROBABILITY

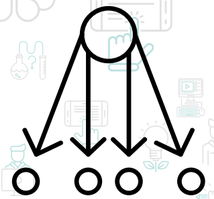
UNSUPERVISED CLASSIFIERS

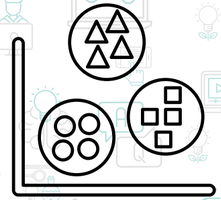
K-Means Clustering - CENTROIDS

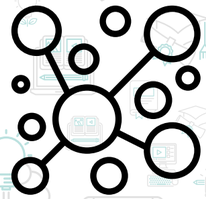
Hierarchical Clustering - DENDOGRAMS

DBSCAN - DENSITY

Principal Component Analysis (PCA) - DIMENSIONALITY REDUCTION







PREPROCESSING

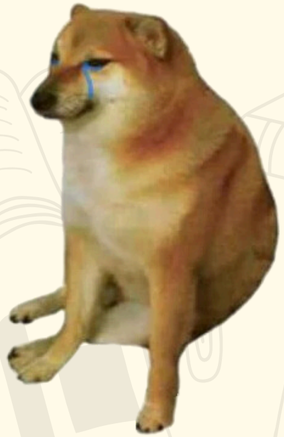
REQUIREMENT OF PREPROCEESSING

Improves **Data Quality**

Enhances Model **Performance**

Reduces computational complexity

Ensures data **compatibility**



PRACTICE DATA

REAL DATA

DIFFERENT STATES OF DATA

Discrete: Countable values, distinct categories (e.g., **students**)
Ordinal: Ordered categories without consistent differences (e.g., **ranks**)
Interval: Numeric, equal intervals, no true zero (e.g., **temperature**)
Ratio: Numeric, equal intervals, true zero (e.g., **weight**)
Nominal: Unordered categories (e.g., **colors, fruits**)
Continuous: Any value within a range (e.g., **time**)

EXPLORING THE DATA

Use plots (histograms, box plots) to understand data distribution
Data can be categorical or numerical (discrete or continuous)
Columns may need renaming for clarity and consistency
Datasets can contain duplicate records, which should be removed
Null values often require imputation or removal
Outliers can distort results and may need to be identified and treated

TYPES OF PREPROCESSING

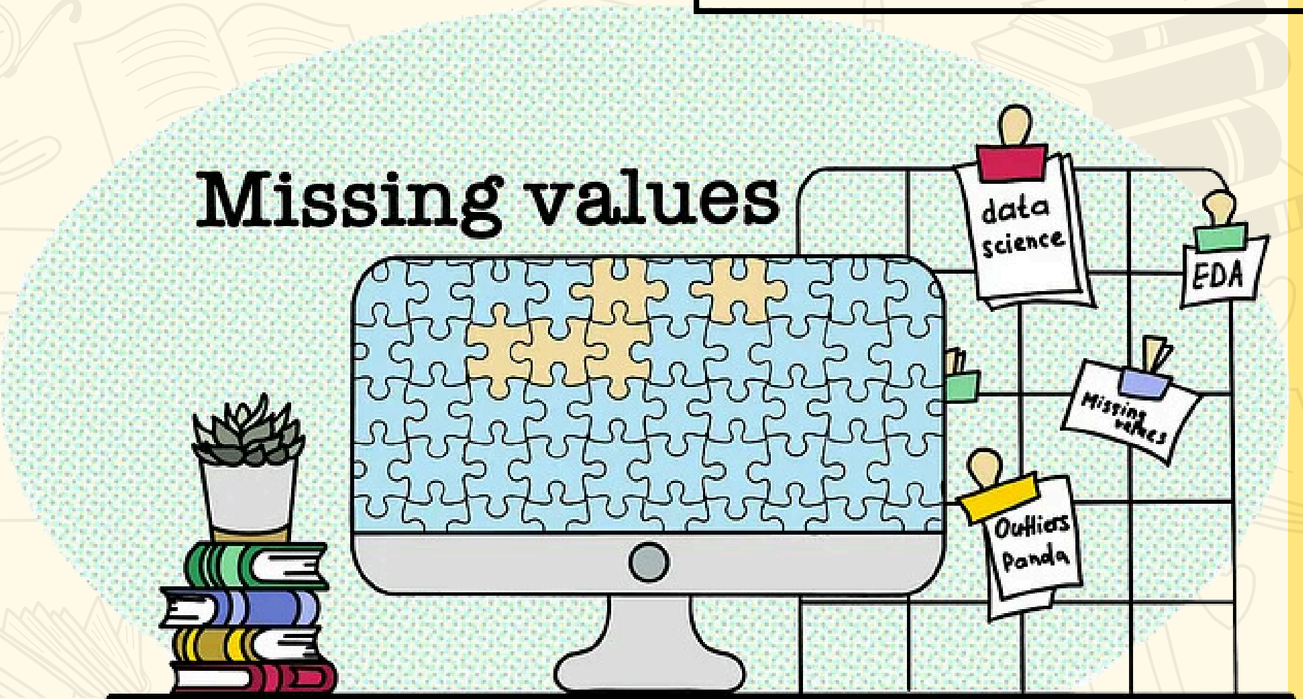
Handling Missing values - IMPUTING

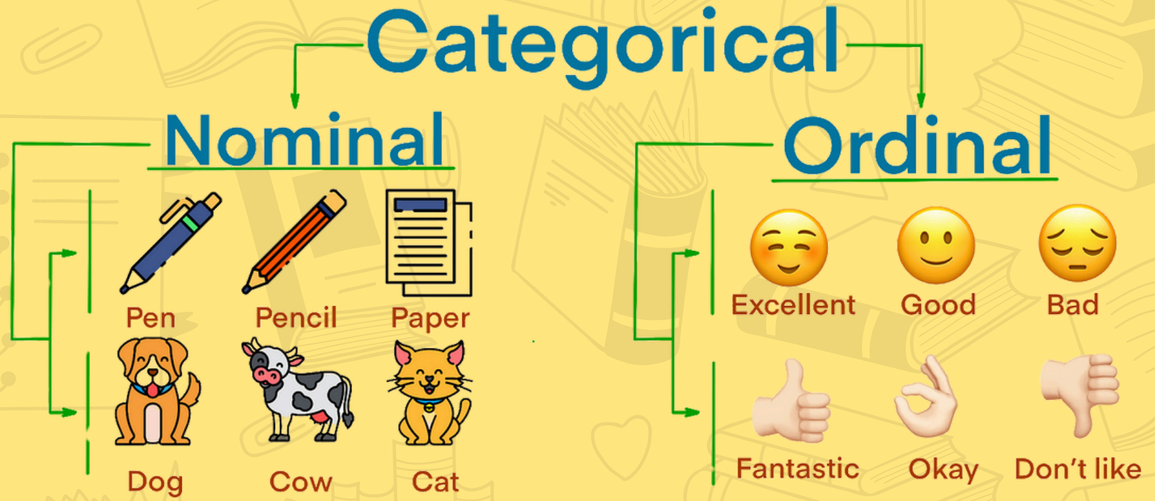
Categorical variables - ENCODING

Outliers

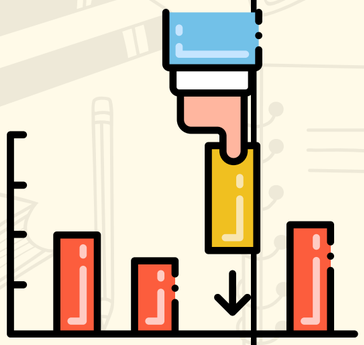
Data Transformation

Dimensionality Reduction





HANDLING MISSING VALUES (IMPUTING)



Mean/Mode/Median Imputation

Replaces missing values with the mean of the feature.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
```

Constant Imputation

Replaces missing values with a constant value.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='constant', fill_value=0)
```

Forward/Backward Fill

Fills missing values with previous or next value in a sequence.

```
import pandas as pd
df.fillna(method='ffill') or df.fillna(method='bfill')
```

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean()

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

One Hot Encoding

Places
New York
Boston
Chicago
California
New Jersey

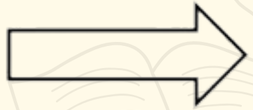
New York	Boston	Chicago	California	New Jersey
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

A Binary Column is created for each Unique Category in the variable

```
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
encoder = OneHotEncoder()
X_encoded = encoder.fit_transform(X)
```

Ordinal Encoding

Grades
A
B
C
D
Fail



Grades	Encoded
A	4
B	3
C	2
D	1
Fail	0

Used when the categories have a Natural Ordering

One Hot Encoding

Places
New York
Boston
Chicago
California
New Jersey

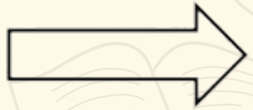
New York	Boston	Chicago	California	New Jersey
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

A Binary Column is created for each Unique Category in the variable

```
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
encoder = OneHotEncoder()
X_encoded = encoder.fit_transform(X)
```

Ordinal Encoding

Grades
A
B
C
D
Fail



Grades	Encoded
A	4
B	3
C	2
D	1
Fail	0

Used when the categories have a Natural Ordering

```
from sklearn.preprocessing import LabelEncoder
```

Count Encoding

Feature
Apple
Banana
Apple
Banana
Banana

Feature	Count
Apple	2
Banana	3

Feature	Encoded
Apple	2
Banana	3
Apple	2
Banana	3
Banana	3

Encoding categorical variables by counting the number of times a category appears

Label Encoding

Places
New York
Boston
Chicago
California
New Jersey

Places	Map
New York	1
Boston	2
Chicago	3
California	4
New Jersey	5

Places	Encoded
New York	1
Boston	2
New York	1
California	4
Boston	2

Each unique category is assigned a Unique Integer value.

```
from category_encoders import CountEncoder
```

```
from sklearn.preprocessing import LabelEncoder
```

Count Encoding

Feature
Apple
Banana
Apple
Banana
Banana

Feature	Count
Apple	2
Banana	3

Feature	Encoded
Apple	2
Banana	3
Apple	2
Banana	3
Banana	3

Encoding categorical variables by counting the number of times a category appears

Label Encoding

Places
New York
Boston
Chicago
California
New Jersey

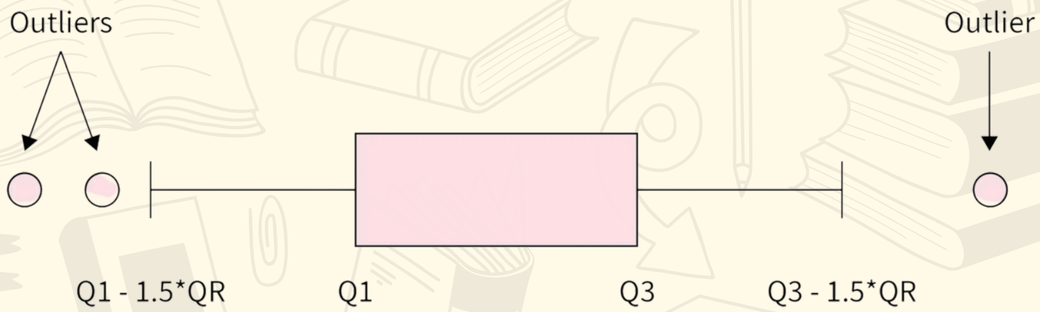
Places	Map
New York	1
Boston	2
Chicago	3
California	4
New Jersey	5

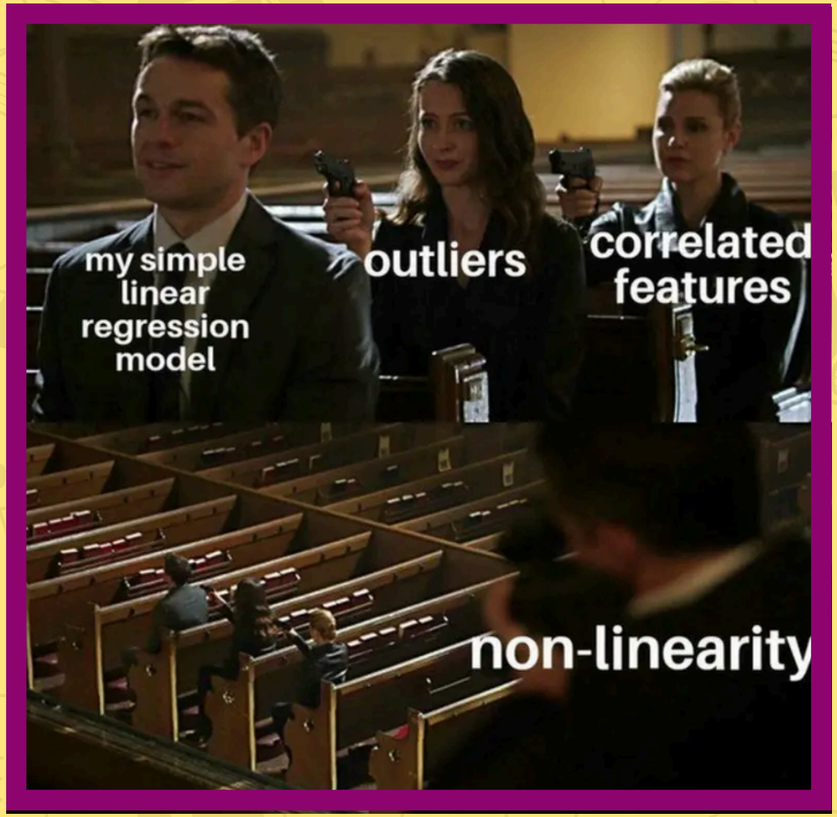
Places	Encoded
New York	1
Boston	2
New York	1
California	4
Boston	2

Each unique category is assigned a Unique Integer value.

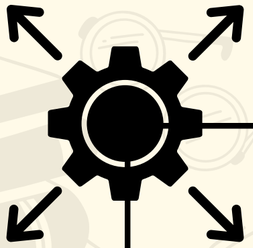
```
from category_encoders import CountEncoder
```

Using IQR, we can define a dataset's upper and lower bounds. The upper bound is defined as Q3 + 1.5*IQR, and the lower bound is defined as Q1 – 1.5*IQR. Any observations or data points that reside beyond and outside these bounds can be considered outliers.





REMOVING OUTLIERS



DATA TRANSFORMATION

WHY?

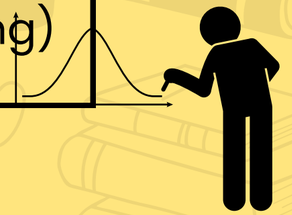
Improving **Data Quality** and **Consistency**
Enhancing **Analytical Performance**
Making Data More **Understandable**
Facilitating Comparisons

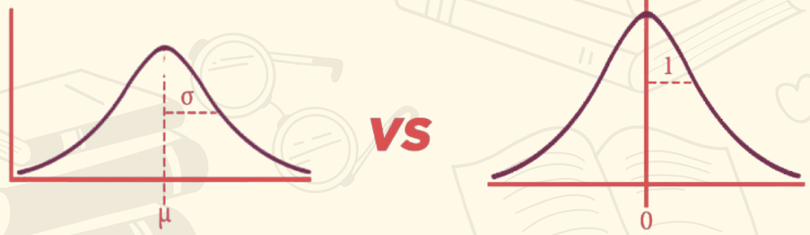
HOW?

Different Scales - Normalization
Scales data to a consistent range (e.g., 0 to 1)

Different Scales - Standardization
Adjusts data to have mean 0 and standard deviation 1

Feature Importance - Feature Scaling
Adjusts feature values to ensure equal contribution (e.g., Min-Max scaling)





Normalization Standardization

VISUALIZE