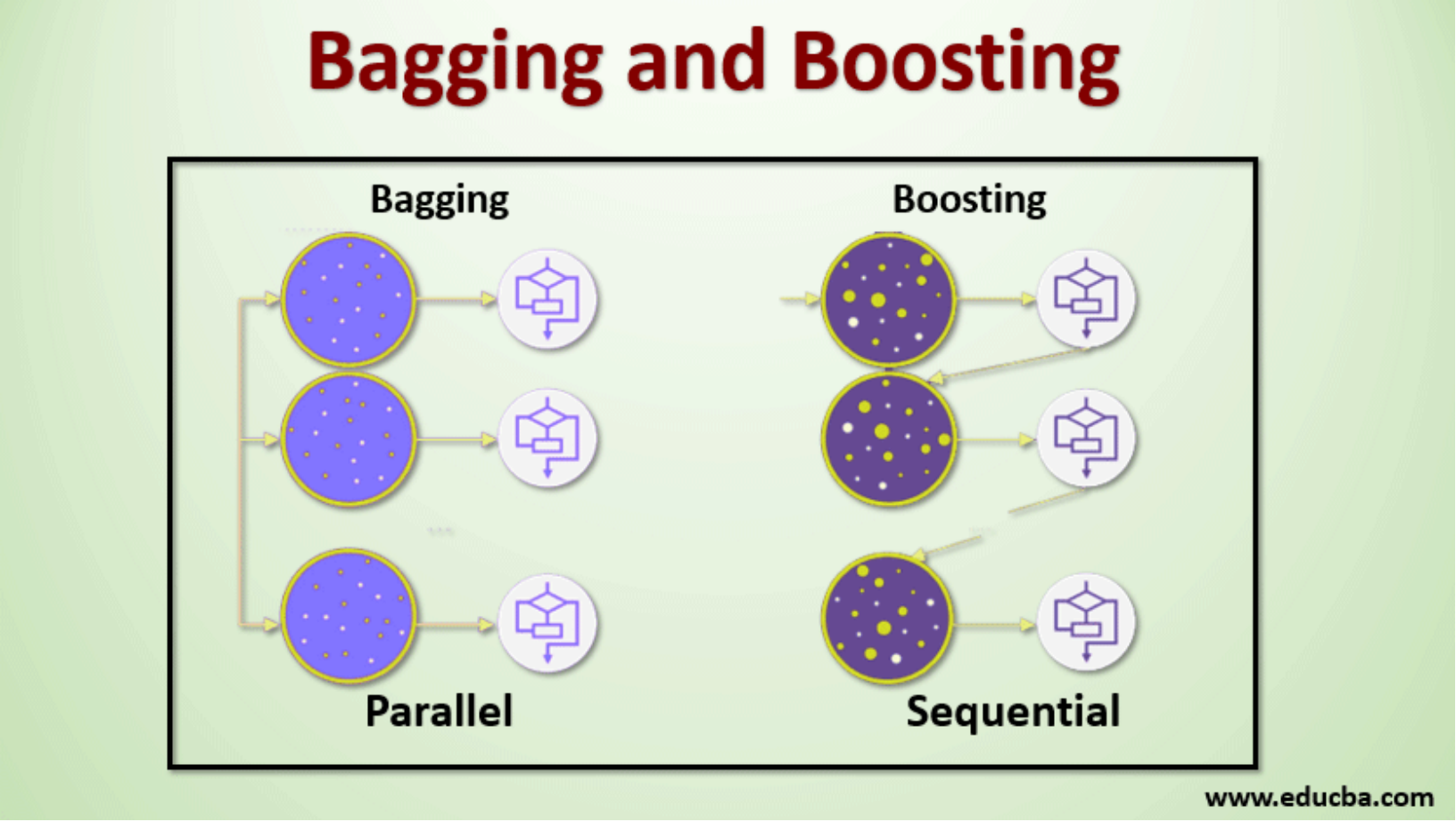
PH-227
AI and Data Science

- Aftab Alam
- Email : aftab@iitb.ac.in
- Ext. : 5564 or 8564

TAs:

Yashowardhan, Divyansh, Matam, Peela, Piyush

Ensemble Learning (Bagging, Boosting)



Cross Validation

- Cross validation is a resampling technique
- It helps to estimate how well a model will perform on an independent dataset

K-fold method

12	8	13	16	11	20	7	17	10	4	18	15	2	4	14	19

Pearson's Correlation Coefficient

- Pearson's correlation coefficient is a correlation coefficient that measures linear correlation between two sets of data

- It is defined as the ratio between the covariance of two variables and the product of their standard deviations

$$r = \frac{Cov(x,y)}{\sigma(x)\sigma(y)} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

- The above expression is therefore a number between -1 and +1. It is equal to unity when all the points lie on a straight line.

Pearson's Correlation Coefficient

$$r = \frac{Cov(x,y)}{\sigma(x)\sigma(y)} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

x	y
-6.0	6.4
2.0	4.7
0.2	8.0
7.0	2.0
-4.0	3.4

Pearson's Correlation Coefficient

$$r = \frac{Cov(x,y)}{\sigma(x)\sigma(y)} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

x	y
-6.0	6.4
2.0	4.7
0.2	8.0
7.0	2.0
-4.0	3.4

$\bar{x}$=-0.16, $\bar{y}$=4.9

$\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$=-23.08

$\sum_{i=1}^{N}(x_i - \bar{x})^2$ =104.91

$\sum_{i=1}^{N}(y_i - \bar{y})^2$=22.56

$$r = 0.473$$

Data Preprocessing & Data Cleaning

- This is an important topic which refers to the technique/procedures used to convert raw data into a clean, organized and structured format suitable for analysis and modeling

Advantages:

- Improving Data Quality
- Enhancing Model Performance
- Reducing Computational Complexity
- Ensuring Compatibility

Various Techniques

- **Handling Missing Values:** Imputation, Deletion, or Prediction-based methods
- **Handling Outliers:** Trimming, or Transformation-based methods
- **Normalization/Scaling:** Min-max scaling, z-score normalization, or robust scaling
- **Encoding Categorical Variables:** One-hot encoding, label encoding, or target encoding
- **Feature Extraction:** Selecting relevant features or transforming existing features
- **Data Splitting:** Splitting data into training, validation and test sets for model evaluation

How to deal with missing values in a dataset

- **Imputation:** This involves replacing missing values with some calculated value, such as the mean, median, or most frequent value of the corresponding feature.
- **k-Nearest Neighbors (kNN) Imputation:** This involves using the kNN algorithm to impute values based on the values of the nearest neighbors in the feature space

Employee ID	Age	Grade	Salary
1	29	76	2000
2	45	NaN	3500
3	NaN	97	4000
4	57	57	NaN

A simple example of Imputation

$$\begin{pmatrix} 1 & 2 & NaN \\ 4 & NaN & 6 \\ Nan & 8 & 9 \end{pmatrix}$$

- Lets use the Mean of each features to replace NaN

$$\begin{pmatrix} 1 & 2 & 7.7 \\ 4 & 5.0 & 6 \\ 2.5 & 8 & 9 \end{pmatrix}$$

Python program

```
From sklearn.impute import SimpleImputer

Import numpy as np


data = np.array ([
          [1,       2,         np.nan]
          [4,       np.nan,6         ]
          [np.nan,  8,         9      ]
])

Imputer = SimpleImputer(strategy= ‘mean’)
Imputed_data = Imputer.fit_transform(data)

Print(data)
Print(imputed_data)
```