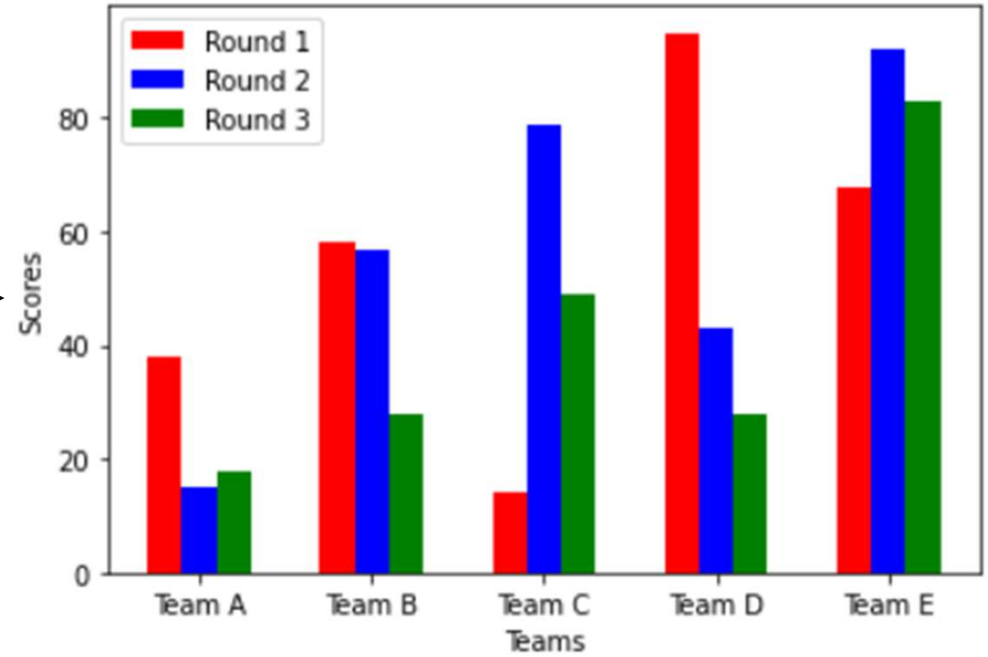
Data Visualization

Plotting Techniques

Data Visualization

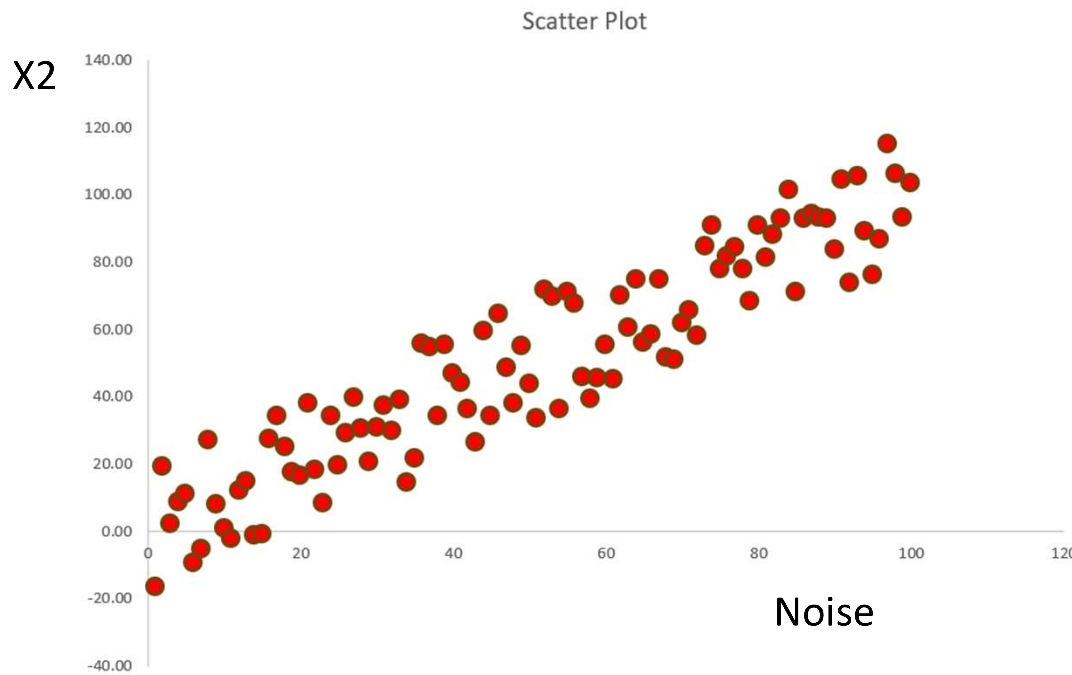
Team	Round 1	Round 2	Round 3
A	38	15	18
B	58	57	28
C	14	79	49
D	95	43	28
E	68	92	83

→



Data Visualization

Noise	X2
2.01	-7.09
9.90	-6.07
1.73	8.53
9.36	5.96
34.55	20.02
39.66	6.70
17.70	17.38
20.39	12.60
39.82	-10.25
23.02	16.85
10.91	13.54
28.44	13.96
14.93	20.62
12.98	26.66
31.48	33.70
17.50	17.77
23.46	35.35
4.20	10.80
18.71	37.51
11.43	25.62
37.53	37.76
0.52	7.14
23.50	38.78
30.05	19.07
39.48	21.85
32.44	19.82
25.85	12.66
1.21	20.79
22.09	18.53
29.89	10.12
25.53	15.66
13.46	33.86
11.30	50.47
26.01	42.72
0.30	46.51
18.92	35.51



Data Visualization

- Data visualization deals with a visual representation of data and is part of data analysis.
- It is the process of translating data into a chart, graph, or other visual components.

❑ **Data visualization can be used for:**

  - Making data engaging and easily digestible
  - Identifying trends and outliers within a set of data
  - Highlighting the important parts of a set of data

Variables

- Variables refer to characteristics, properties, or attributes that can be measured, observed, or recorded for a particular entity or unit within a dataset
- Types of Variables: Dependent Variables, Independent Variables

❑ Based on the nature:

Qualitative (Categorical): It describes the quality of something or someone. It is descriptive information. For e.g., skin color, eye color gives us qualitative information about a person.

Quantitative (Numerical): It provides numerical information, like how much, how many, or how often. Can be continuous or discrete. For e.g., the height and weight of a person.

Univariate Analysis

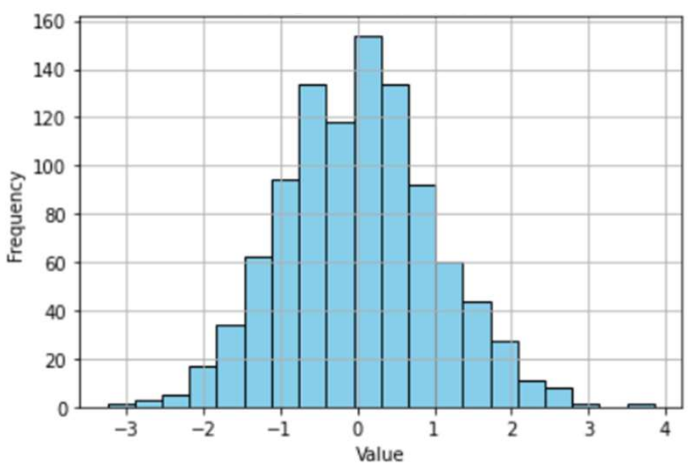
- Univariate Analysis is used in statistics to describe a data type that contains only one attribute or characteristic

Data = [0.49,-1.13,2.64,0.15,-1.23,-0.24,1.58,0.77,-0.47, 0.54, -2.46, -0.44, 0.22, 0.76,1.24, -0.54,-2.11,0.12,0.02,-1.15,....., 0.32,-0.12,0.19]

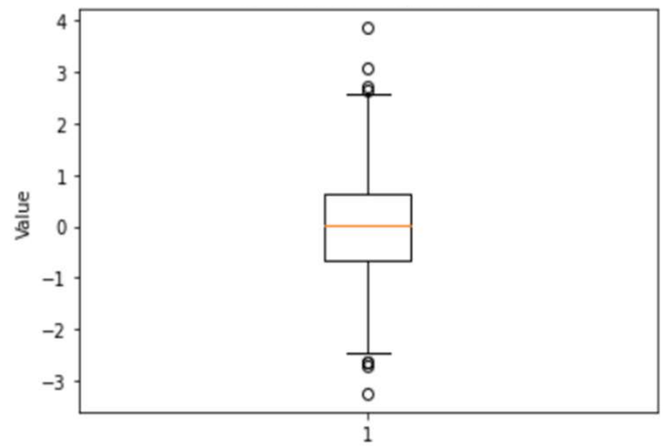
- Includes checking the central tendency (mean, median and mode), the range, the maximum and minimum values and the standard deviation of a variable

Univariate Analysis

Data = [0.49,-1.13,2.64,0.15,-1.23,-0.24,1.58,0.77,-0.47, 0.54, -2.46, -0.44, 0.22, 0.76,1.24, -0.54,-2.11,0.12,0.02,-1.15,....., 0.32,-0.12,0.19]



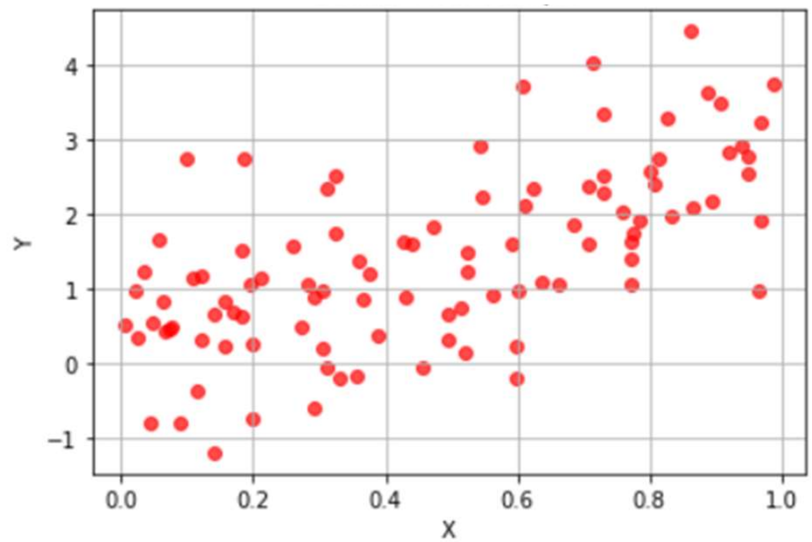
Histogram: Frequency distribution graph



Box Plot: Compare the spread of the variables and get an insight into outlier

Bivariate Analysis

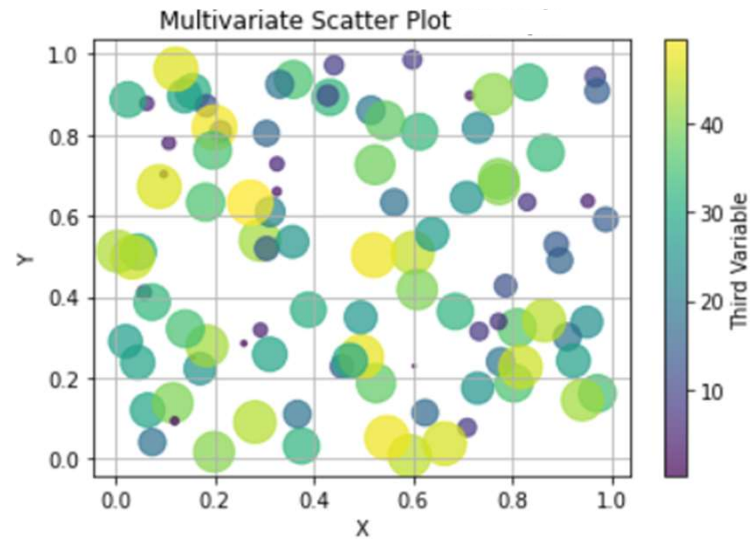
- The bivariate analysis is mainly used to compare two sets of data to find a relationship between the two variables.
- Remember, that if one variable influences the change in another variable, then you have an independent and dependent variable.
- Ex:- Scatter Plot, Heatmap, Contour Plot, Bivariate Line Chart, Pair Plot, etc.

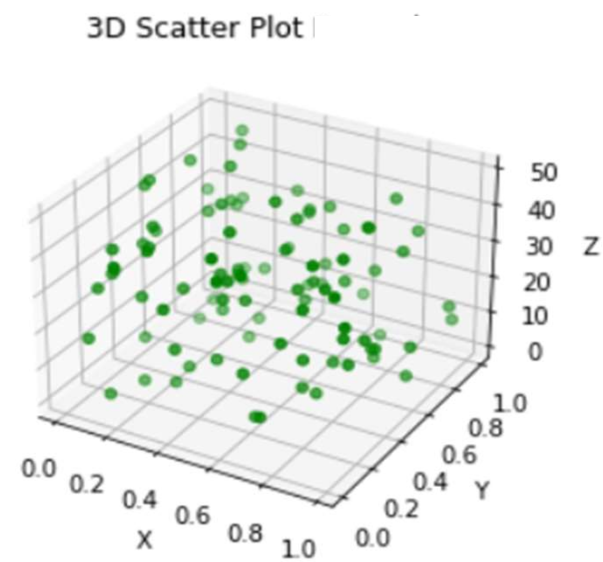


Scatter Plot: Captures the correlation between the two

Multivariate Analysis

- Multivariate analysis is used to reveal the relationship among several variables simultaneously.
- Assists in making informed decisions by considering multiple variables and their interactions simultaneously.
- Ex: - Grouped Box Plot, Multivariate Scatter Plot, and 3D scatter plot.





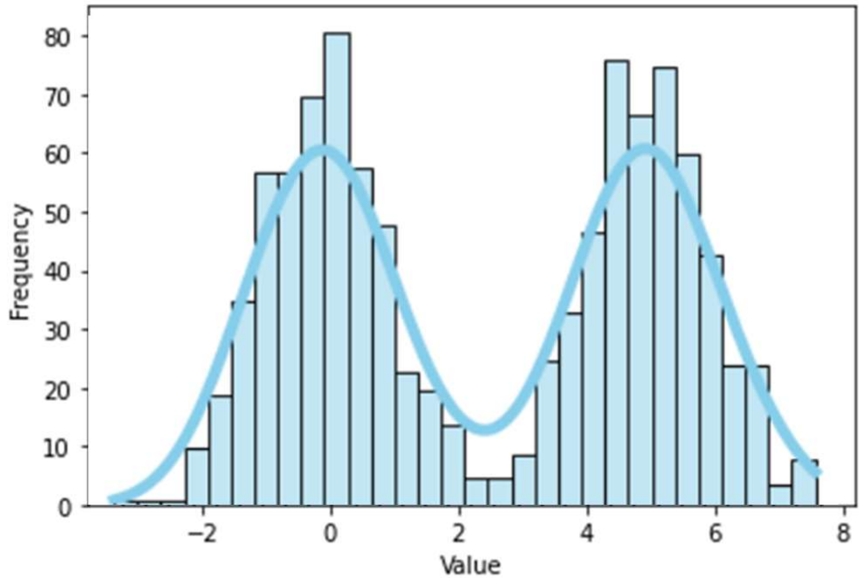
Visualization Techniques

- Distribution of data points: Box plot, Histogram
- Comparison of data points: Multi-line chart, Bar plot, Line chart [to show trends in data]
- Relationship/Correlation of data points: Scatter plot
- Composition of data points: Pie chart, Stacked Area chart, Stacked Bar chart

Violin Plot

Data = [0.49,-1.13,2.64,0.15,-1.23,-0.24,1.58,0.77,-0.47, 0.54, -2.46, -0.44, 0.22, 0.76,1.24, -0.54,-2.11,0.12,0.02,-1.15,....., 0.32,-0.12,0.19]

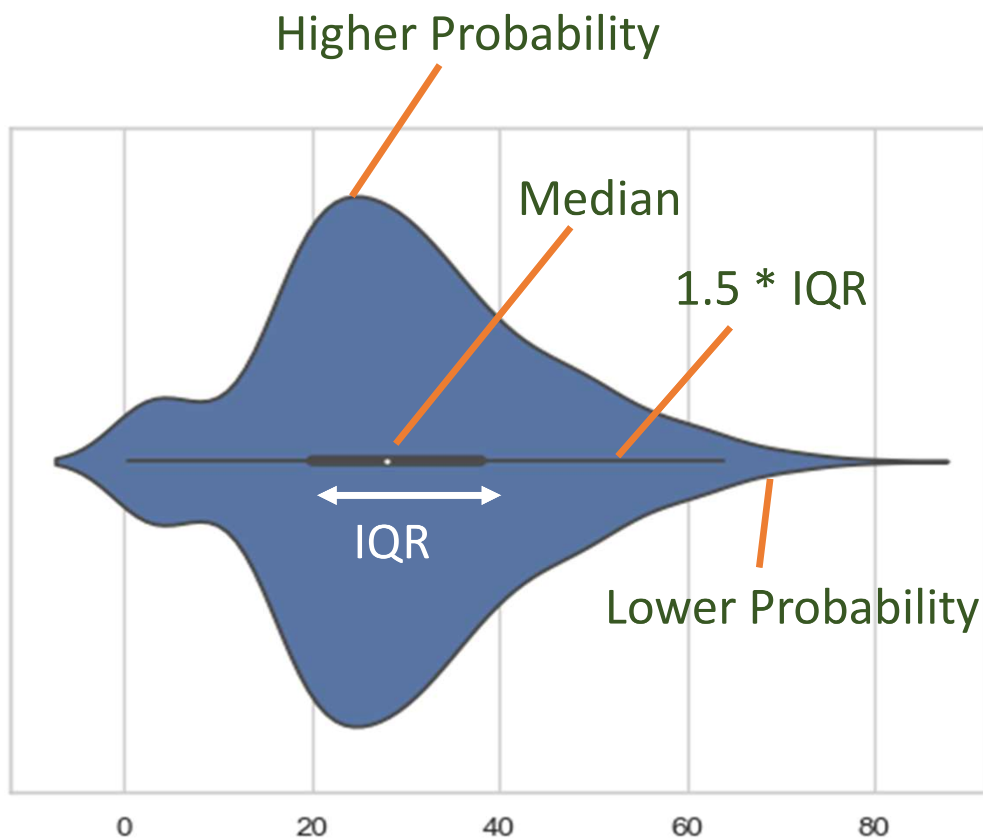
- The box plot is convenient for comparing summary statistics (such as range and quartiles), but it doesn't let you see variations in the data.
- Are most of the values clustered around the median, or around the minimum/maximum?
- The histogram and kernel density estimation helps you in seeing the variations in the data, but you miss the outliers



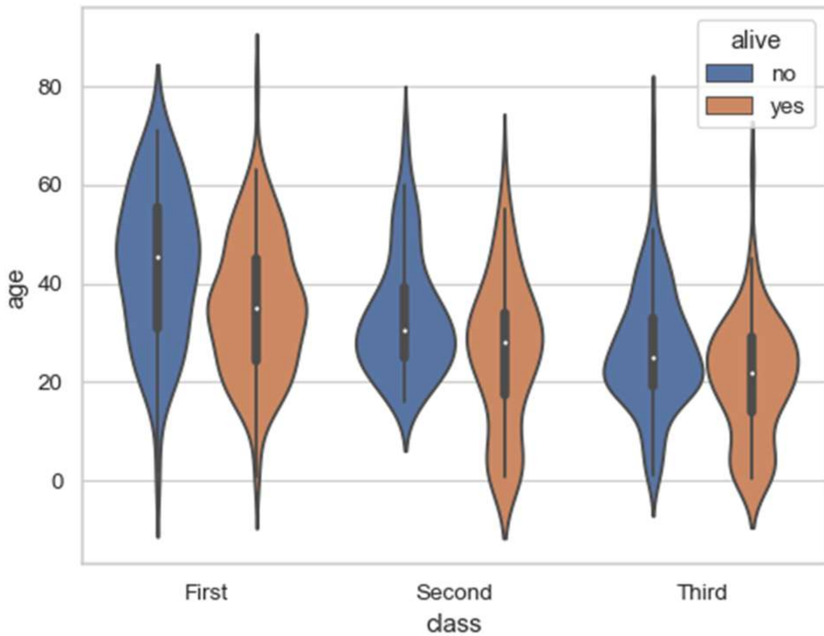
Can we combine both?

Violin Plot

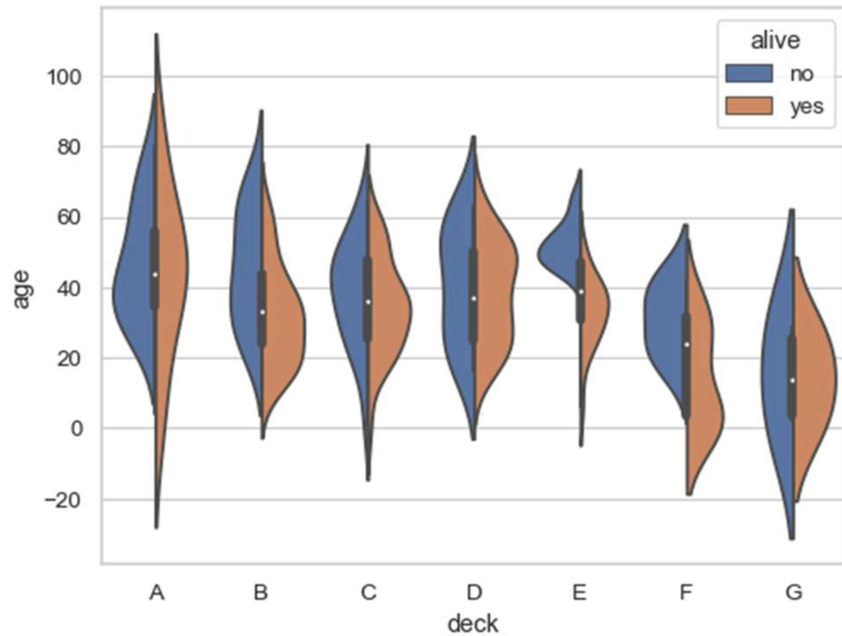
- It is a combination of box plot and a kernel density plot, which shows peaks in the data
- It depicts the summary statistics and the density of each variable



Violin Plot



Vertical violins, grouped by two variables



Split violins to take up less space

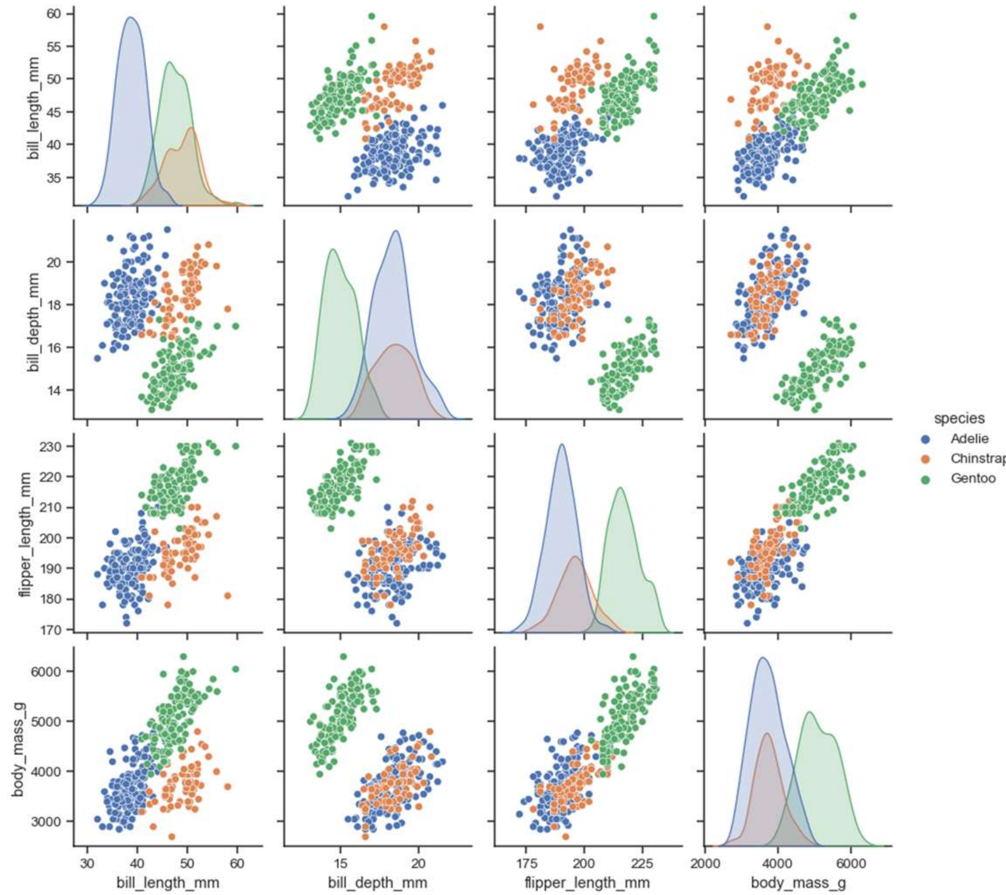
Plot Courtesy: Seaborn Documentation [seaborn.pydata.org]

Pair Plot

- Pair plot visualizes given data to find the relationship between them and plots pairwise relationships in a data-set

- It is used for exploring the relationship between multiple variables at once

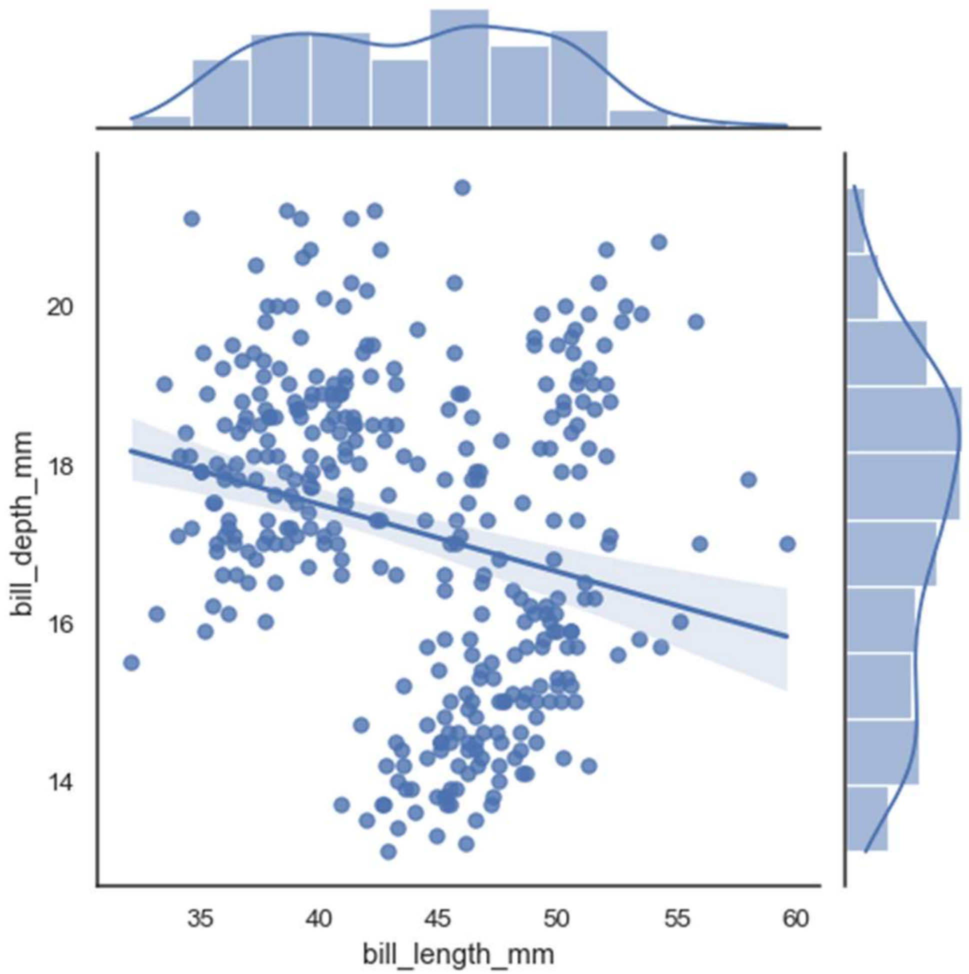
- Plots in a matrix format
  - Diagonal subplots are the univariate histograms for each attribute
  - Off-diagonal entries are the scatter plots



Plot Courtesy: Seaborn Documentation [seaborn.pydata.org]

Joint Plot

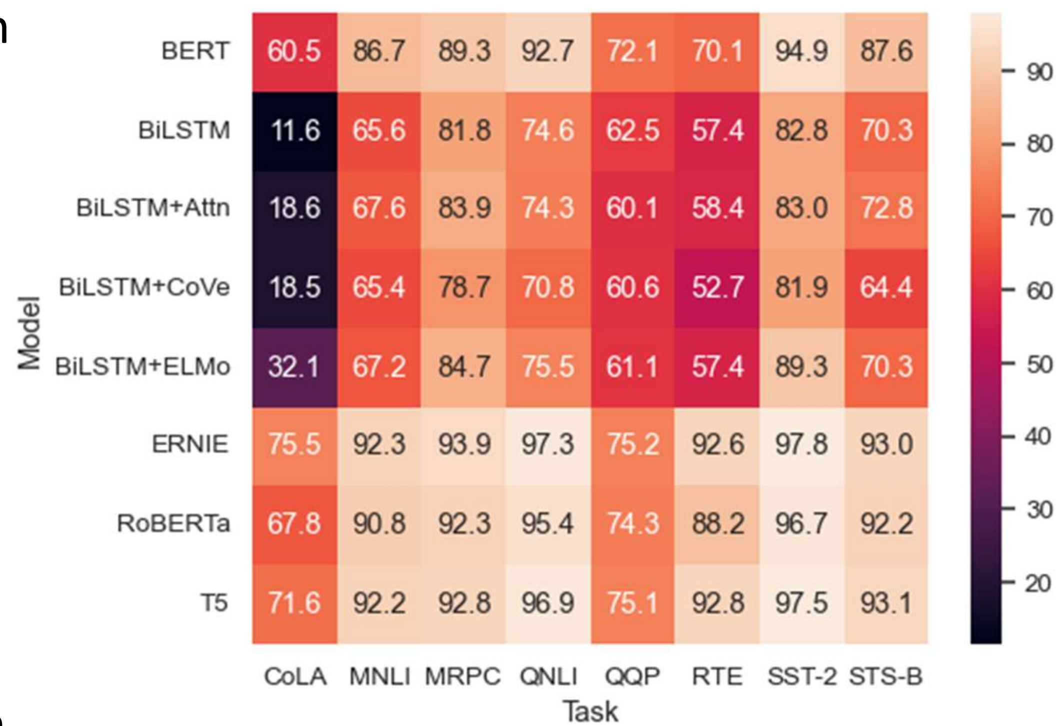
- Joint plot combines univariate and bivariate plots to visualize relationship between two variables
- It consists of a scatter plot for the bivariate relationship, with additional marginal plots for each variable
- Helps in understanding the correlations and distributions of two variables simultaneously



Plot Courtesy: Seaborn Documentation [seaborn.pydata.org]

Heatmap

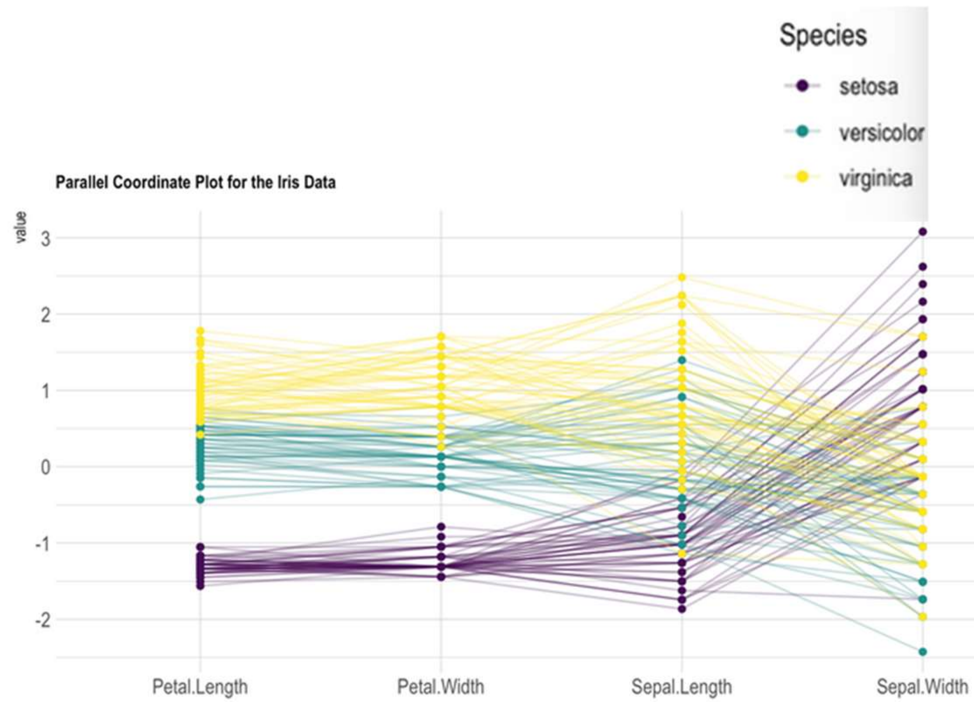
- A heatmap is a color-coded representation of a 2-dimensional data, representing the magnitude of individual values within a dataset
- Colours are used to represent the magnitude, intensity, with the colour gradient scheme ranging from a lighter colour (low values) to a darker colour (high values)
- Displays the correlations or relationships in a correlation matrix



Plot Courtesy: Seaborn Documentation [seaborn.pydata.org]

Parallel Co-ordinates

- Parallel Co-ordinates allows for the comparison of multiple data records, by using parallel lines to connect points based on multiple numerical variables
    - Each vertical line is a dimension
    - A data item is connected by line segments
    - Large number of samples clutters the visualization



Dimensionality Reduction

Dimensionality

- The number of input variables or features for a dataset is referred to as its dimensionality.

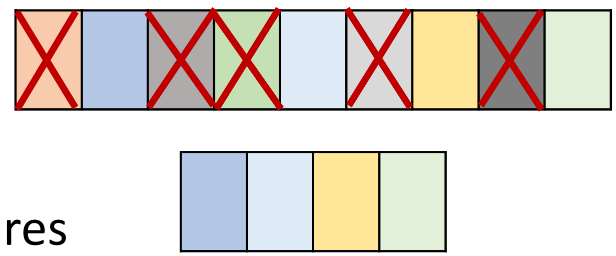
- The difficulties related to training machine learning models due to high dimensional data is referred to as 'Curse of Dimensionality'.

- *When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the "essence" of the data. This is called dimensionality reduction.*

— Page 11, Machine Learning: A Probabilistic Perspective, 2012.

Dimensionality Reduction

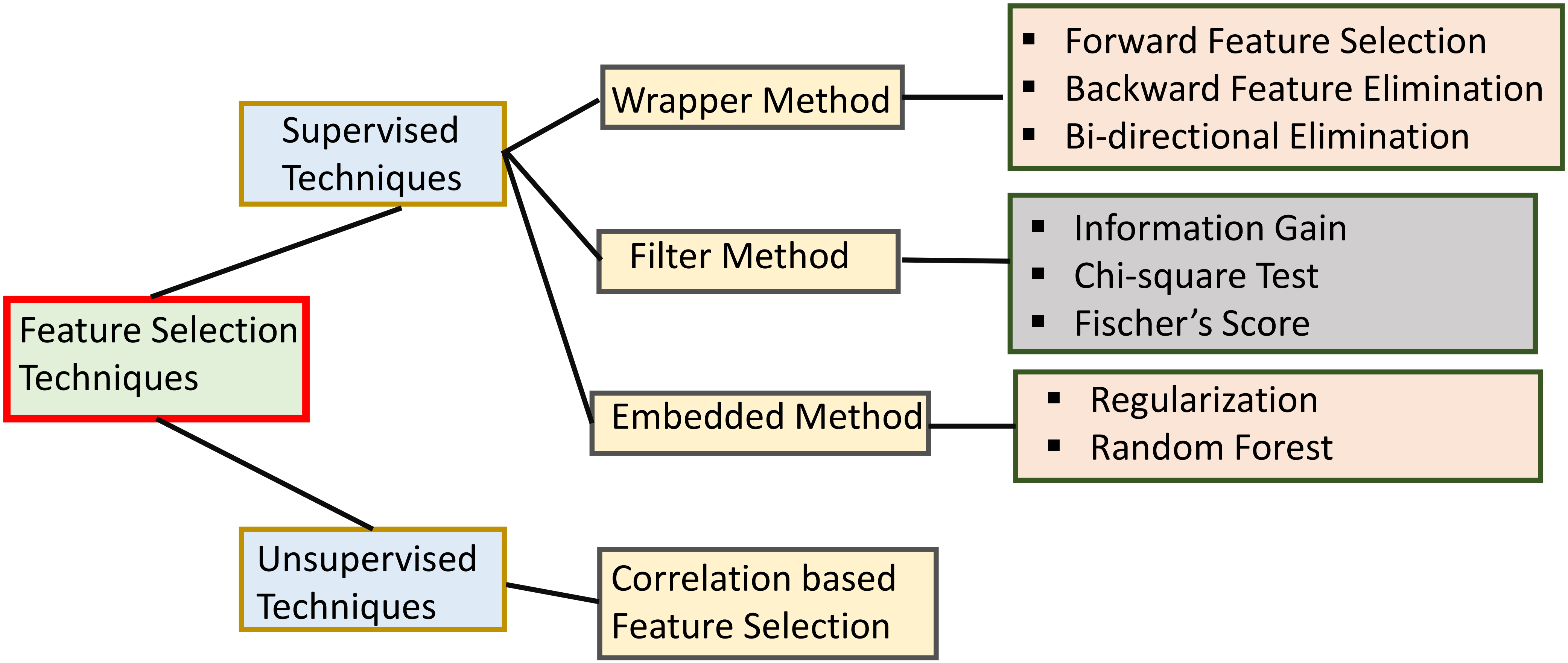
- Feature Selection
  - Select the most relevant subset of features
  - Reducing the number of irrelevant or redundant features



- Feature Extraction
  - extracting/deriving information from the original features set to create a new features subspace
  - compress the data with the goal of maintaining most of the relevant information

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 2.5 & 0.6 & 1.3 & 0.8 \\ 3 & -1 & 0.7 & 4.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Feature Selection Techniques



Forward-Feature Selection

- Iteratively selects one feature at a time, evaluating the model's performance after adding each feature and keeping the best subset of features that maximizes or minimizes the chosen performance metric.

- **Step 1**: Initialization: Initialize an empty set **S** to store the selected features
- **Step 2**: Loop over Features: For each feature X_i to be added to **S**
  - Find X_i that best improves the model's performance metric when added to **S**
  - Update the selected X_i to **S**
- **Step 3**: Termination:
  - Repeat **Step 2** until adding any remaining feature does not improve the model's performance.
  - Return **S** as the selected feature subset

Forward Feature Selection

Sr. no	Feature_idx	Avg_Score
1	(10,)	0.541
2	(5,10)	0.638
3	(5,8,10)	0.682
4	(5,7,8,10)	0.696
5	(4,5,7,8,10)	0.715
6	(3,4,5,7,8,10)	0.721
7	(3,4,5,7,8,9,10)	0.724
8	(1,3,4,5,7,8,9,10)	0.728
9	(0,1,3,4,5,7,8,9,10)	0.729
10	(0,1,2,3,4,5,7,8,9,10)	0.730
11	(0,1,2,3,4,5,6,7,8,9,10)	0.732

Backward Feature Elimination

- Start with all available features, iteratively remove one feature at a time, and evaluate the model's performance.
- If the performance improves, we keep the feature removed; otherwise, we add it back.
- The final set of features that maximizes or minimizes the chosen performance metric is returned as the selected feature subset.

```
BackwardFeatureElimination(X, y):
   S = {all features}  # Initialize with all features
   best_score = EvaluateModel(X, y, S)  # Evaluate initial model
using all features

   while there are remaining features in S:
      for feature in S:
         Remove feature from S
         Train a model using the features in S
         Evaluate model performance using a chosen metric
         If model performance improves compared to
best_score:
            Update best_score to the new performance
         else:
            Add feature back to S

   return S  # Return the remaining features after elimination
```

Backward Elimination

Sr. no	Feature_idx	Avg_Score
11	(0,1,2,3,4,5,6,7,8,9,10)	0.732
10	(0,1,2,3,4,5,7,8,9,10)	0.730
9	(0,1,3,4,5,7,8,9,10)	0.729
8	(1,3,4,5,7,8,9,10)	0.728
7	(3,4,5,7,8,9,10)	0.724
6	(3,4,5,7,8,10)	0.721
5	(4,5,7,8,10)	0.715
4	(5,7,8,10)	0.696
3	(5,8,10)	0.682
2	(5,10)	0.638
1	(10,)	0.541

Bi-directional Elimination

- Combines forward and backward feature selection techniques to iteratively select a subset of features that optimizes a model performance metric

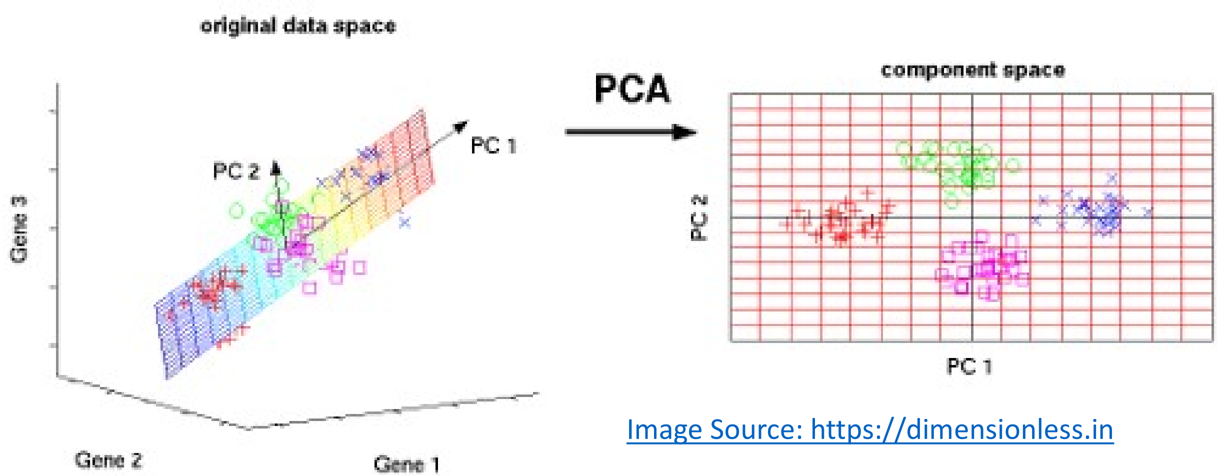
❖ **Step 1**: Initialization: Initialize an empty set **S** to store the selected features and choose direction

❖ **Step 2**: Loop over Features: For direction chosen as 'forward' or 'backward'
  - If 'forward': Perform forward selection adding the best feature that improves the model's performance metric
  - If 'backward': Perform backward elimination, removing the least significant feature

❖ **Step 3**: Termination:
  - Repeat **Step 2** until adding/eliminating any remaining feature does not improve the model's performance.
  - Return **S** as the selected feature subset

Feature Extraction

- Aims to reduce the number of features in a dataset by creating new features from the existing ones (discarding the original ones)
- **Feature Extraction Techniques**:
  - **Principal Component Analysis**: Linear transformation techniques by finding orthogonal axes that capture the most variance



[Image Source: https://dimensionless.in](https://dimensionless.in)

Feature Extraction

- Aims to reduce the number of features in a dataset by creating new features from the existing ones (discarding the original ones)

- **Feature Extraction Techniques**:
  - **Isomap, t-SNE**(t-distributed Stochastic Neighbor Embedding): Non-linear dimensionality reduction technique that emphasizes the local structure of the data

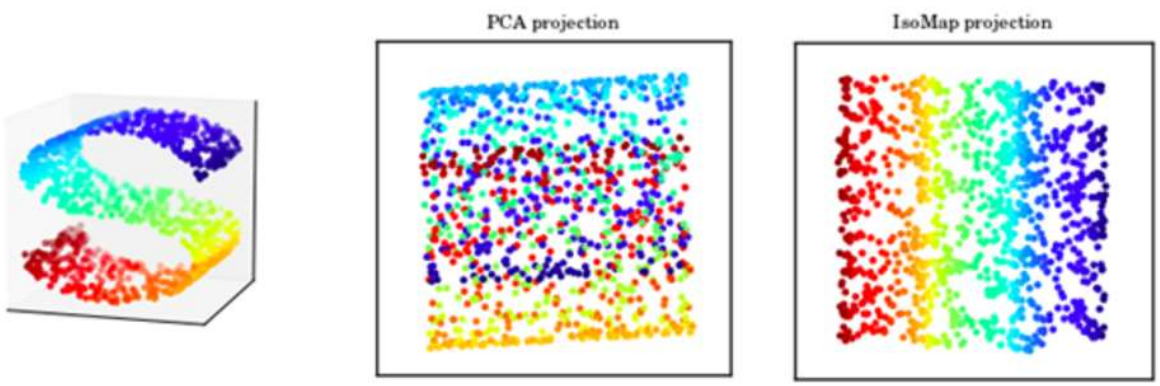


Image Source : https://ciera.northwestern.edu