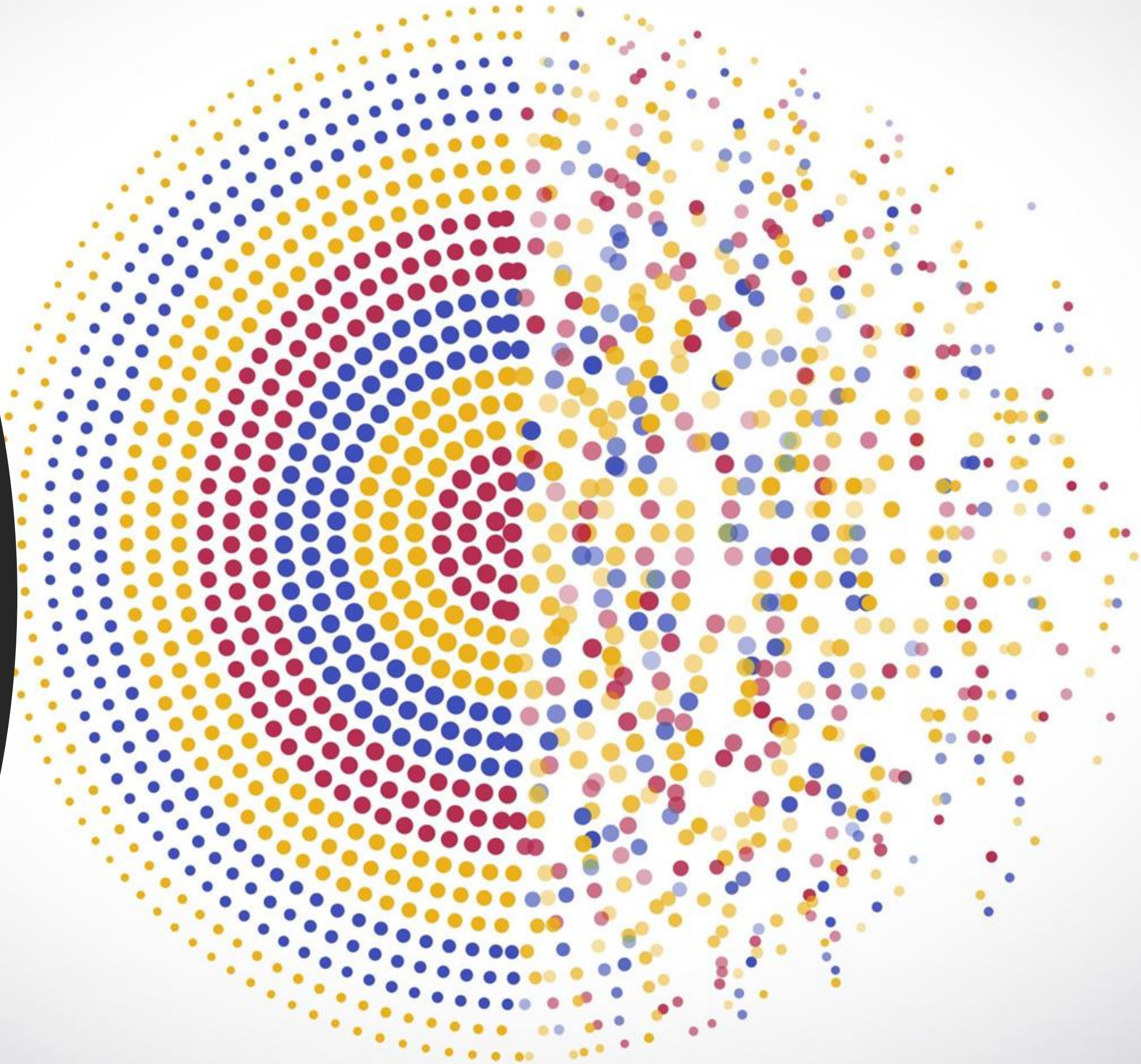
# Data Preprocessing

Dr. Mohamed AlHajri



# Data Preprocessing

- Aggregation
- Sampling
- Normalization
- Encoding
- Discretization

# Aggregation

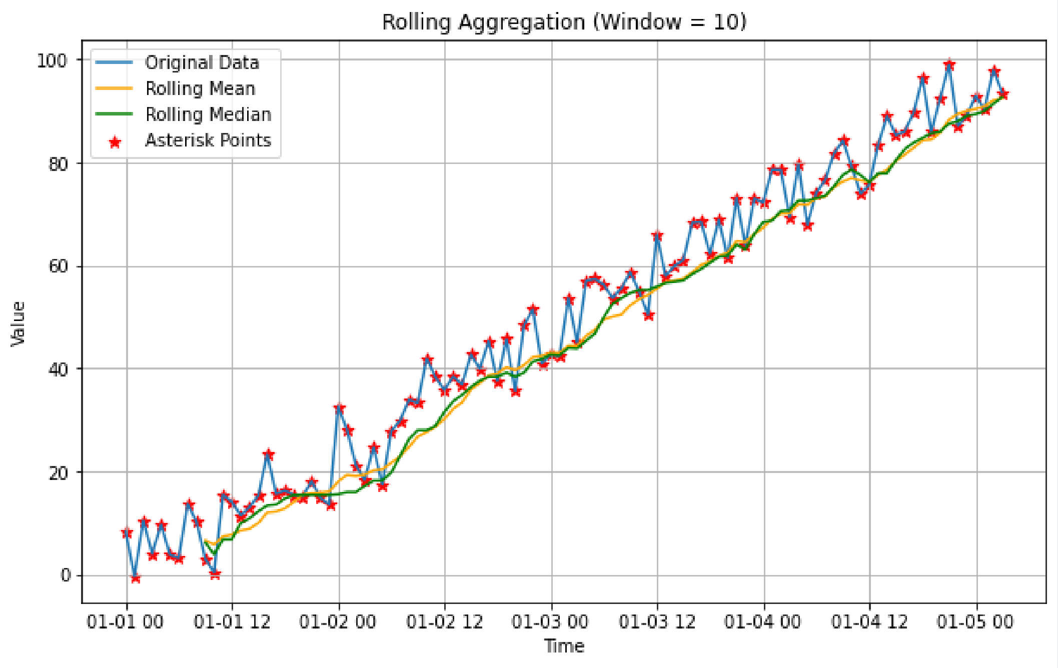
- Aggregation involves summarizing or combining data points to reduce the dataset's size and complexity. This is particularly useful when handling large time-series data, or data that needs to be summarized at different levels of granularity. Aggregation can be used to calculate various metrics such as sums, averages, or maximum values across groups or time windows.

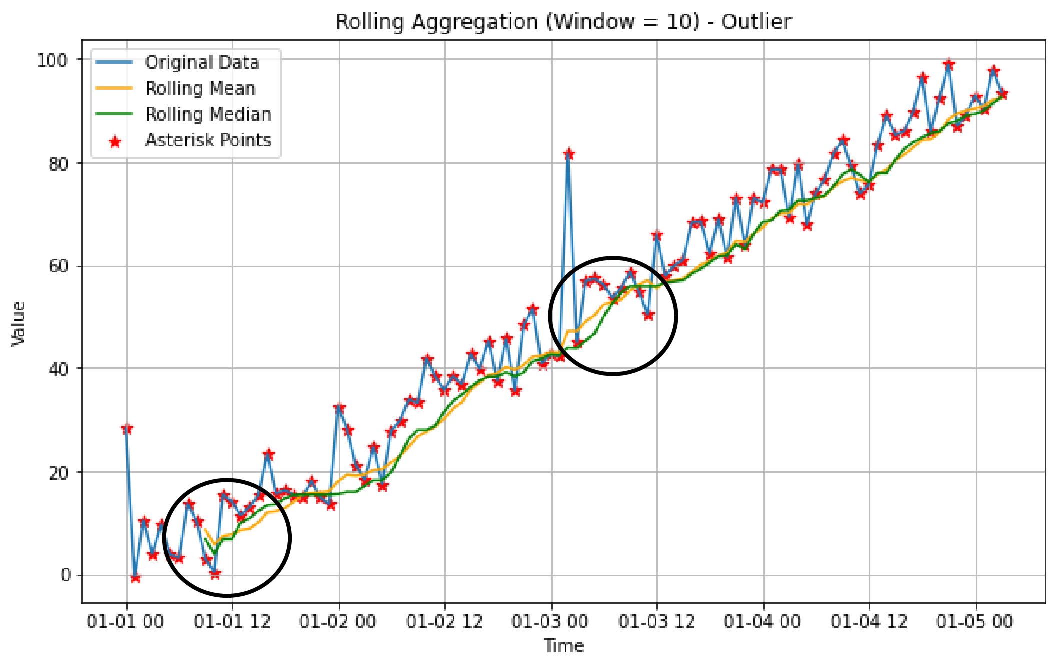
- Common Aggregation Techniques:
  - Sum Aggregation: Sums values within a group.
  - Mean Aggregation: Averages values in a group (e.g., mean temperature over a day).
  - Median Aggregation: Finds the median, which is less sensitive to outliers than the mean.
  - Windowed Aggregation: Aggregates values within a specific time or data window, often used in time-series data.
  - Spatial Aggregation: Combining geographically close data points (e.g., population density by region).

# Aggregation

| Aggregation Technique | Advantages | Disadvantages | When to Use |
|---|---|---|---|
| **Mean Aggregation** | Simple, works well for normally distributed data. | Sensitive to outliers. | Use when data is symmetric or without significant outliers. |
| **Median Aggregation** | Robust to outliers. | Does not capture fine-grained variations like the mean. | Use when data contains outliers or is skewed. |
| **Windowed Aggregation** | Helps track trends over time windows. | Can introduce lag in real-time applications. | Use for time-series data to smooth and detect trends. |

# Aggregation





# Sampling

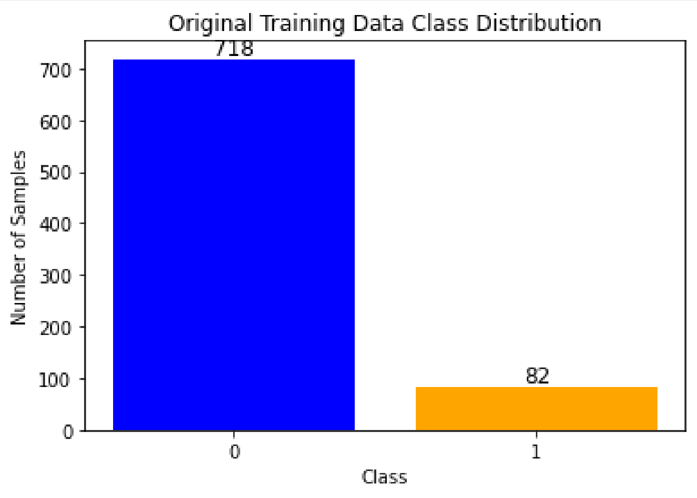
- Sampling is the process of selecting a subset of data from a larger population. It is used to reduce the amount of data for processing while maintaining a representative sample. Sampling is critical when working with large datasets or imbalanced data.

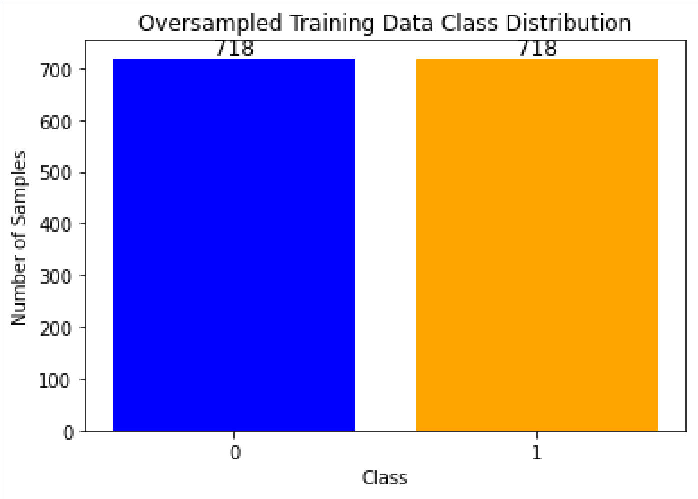
- Common Sampling Techniques:
  - Random Sampling: Selecting random instances from the dataset.
  - Stratified Sampling: Ensures each class or group is represented proportionally in the sample.
  - Oversampling/Undersampling: Used in cases of imbalanced classes, where the minority class is oversampled, or the majority class is undersampled.
  - Systematic Sampling: Data points are selected at regular intervals.

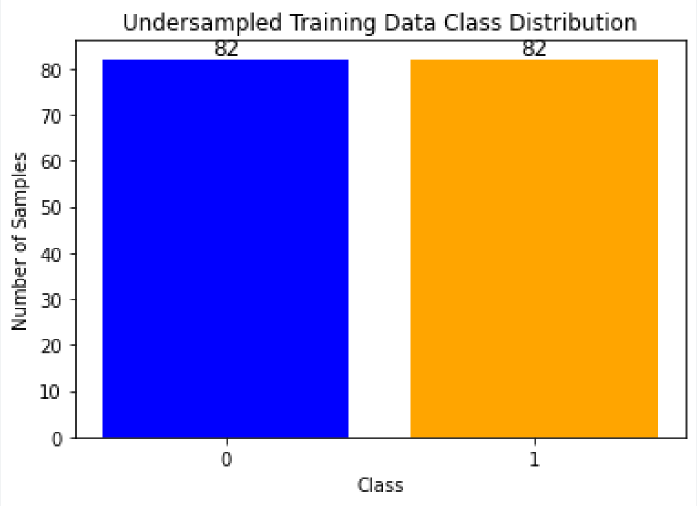
# Sampling

| Sampling Technique | Advantages | Disadvantages | When to Use |
|---|---|---|---|
| **Random Sampling** | Simple, effective for large, homogeneous datasets. | May not represent all classes proportionally (in imbalanced datasets). | Use for large datasets without inherent bias. |
| **Stratified Sampling** | Ensures proportional representation of all groups/classes. | Requires knowledge of class labels. | Use for imbalanced datasets with categorical data. |
| **Oversampling** | Effective for balancing minority classes. | May introduce overfitting if the minority class is repeated too often. | Use when dealing with imbalanced classification problems. |

# Sampling


Original Training Data Class Distribution
718
82
700
600
500
400
300
200
100
0
Number of Samples
0
1
Class


Oversampled Training Data Class Distribution
718
718
700
600
500
400
300
200
100
0
Number of Samples
0
1
Class


Undersampled Training Data Class Distribution
82
82
80
70
60
50
40
30
20
10
0
Number of Samples
0
1
Class

# Normalization

- Normalization is the process of scaling numeric data so that it fits within a specific range, often to improve model performance. Normalization is essential when features have different scales (e.g., age vs income). Different types of normalization handle different data characteristics.

- Common Normalization Techniques:
  - Min-Max Normalization: Scales features to a range between 0 and 1.

$$\frac{X - X_{min}}{X_{max} - X_{min}}$$

  - Z-Score Normalization: Centers data with mean 0 and variance 1.

$$\frac{X - \mu_X}{\sigma_X}$$

# Normalization

| Normalization Technique | Advantages | Disadvantages | When to Use |
|---|---|---|---|
| **Min-Max Normalization** | Simple, ensures all data is in [0, 1] range. | Sensitive to outliers; compresses the range of large values. | Use when data is bounded and doesn't have outliers. |
| **Z-Score Normalization** | Handles both positive and negative values well. | Less intuitive interpretation of scaled data. | Use when data is normally distributed. |

# Encoding

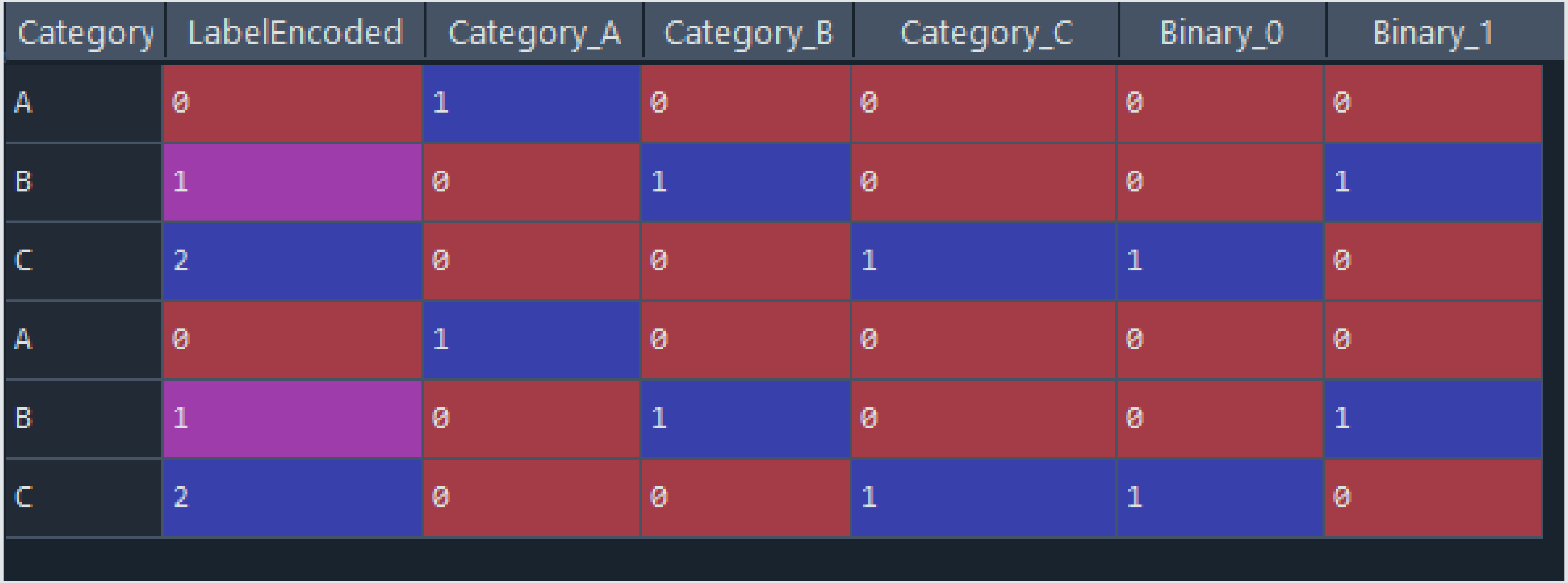
- Encoding is essential when dealing with categorical data. Since most machine learning algorithms require numerical input, categorical features must be converted into a numerical format. Different encoding techniques handle different data types (nominal vs ordinal) and cardinality.

- Common Encoding Techniques:
    - Label Encoding: Converts each category to a unique integer label.
    - One-Hot Encoding: Creates a binary column for each category.
    - Frequency Encoding: Replaces categories with their frequency in the dataset.
    - Binary Encoding: Converts categories to binary codes.

# Encoding

| Encoding Technique | Advantages | Disadvantages | When to Use |
|---|---|---|---|
| **Label Encoding** | Simple and memory-efficient. | Introduces an ordinal relationship that may not exist. | Use for ordinal categorical data (where order matters). |
| **One-Hot Encoding** | Avoids introducing ordinal relationships. | Can lead to high-dimensional data if categories are numerous. | Use for nominal categorical data with few categories. |
| **Binary Encoding** | Reduces dimensionality for high-cardinality categories. | May be less interpretable. | Use when the dataset contains high-cardinality categorical variables. |

# Encoding



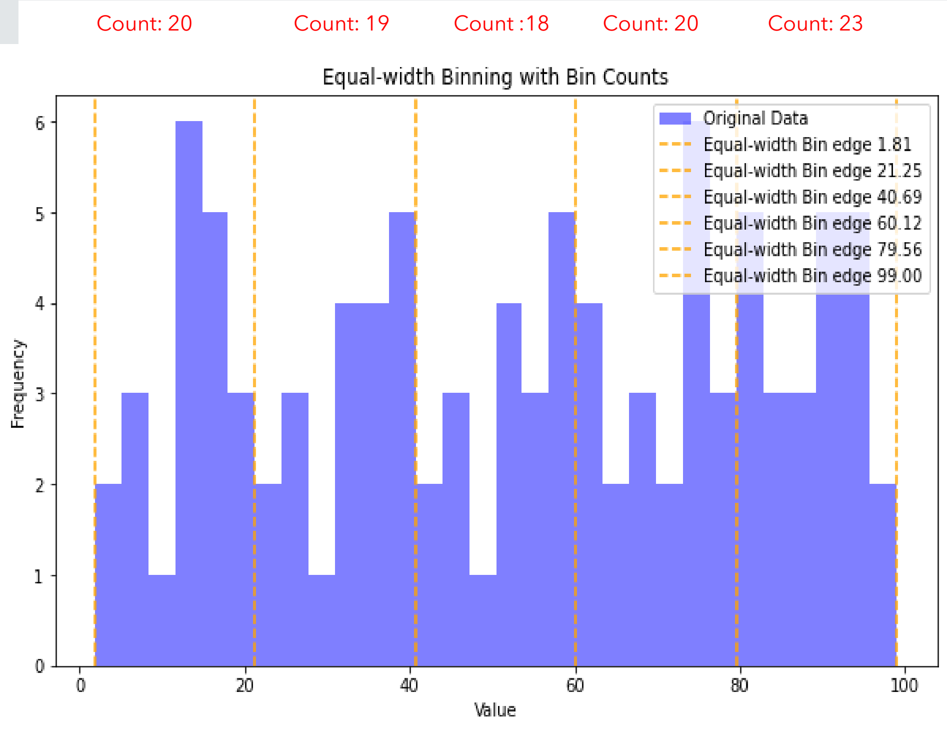
| Category | LabelEncoded | Category_A | Category_B | Category_C | Binary_0 | Binary_1 |
|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 1 | 0 | 0 | 1 |
| C | 2 | 0 | 0 | 1 | 1 | 0 |
| A | 0 | 1 | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 1 | 0 | 0 | 1 |
| C | 2 | 0 | 0 | 1 | 1 | 0 |

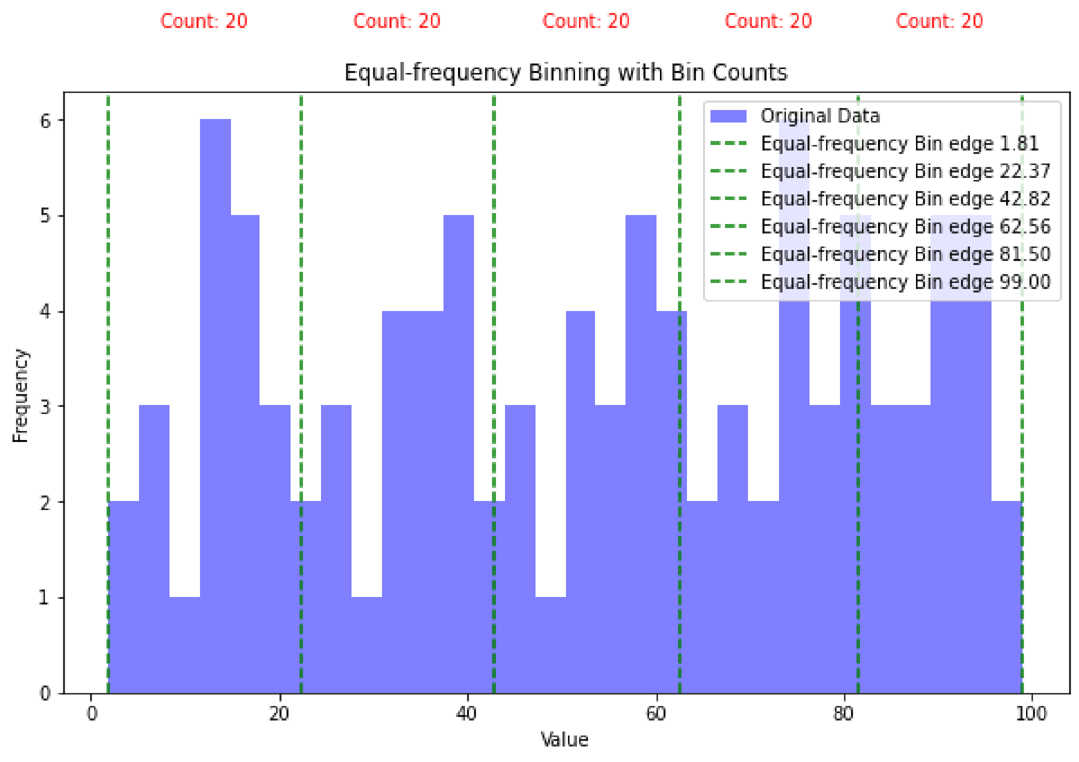
# Discretization

- Discretization is the process of converting continuous features into discrete bins or categories. This is useful when you want to apply models that assume categorical inputs or to reduce the granularity of features.

- Common Discretization Techniques:
    - Equal-width Binning: Divides the range of data into equal-width bins.
    - Equal-frequency Binning: Ensures each bin has approximately the same number of data points.
    - K-means Binning: Uses clustering algorithms (like K-means) to form bins based on cluster centers.

# Discretization


Count: 20
Count: 19
Count :18
Count: 20
Count: 23
Equal-width Binning with Bin Counts
Original Data
Equal-width Bin edge 1.81
Equal-width Bin edge 21.25
Equal-width Bin edge 40.69
Equal-width Bin edge 60.12
Equal-width Bin edge 79.56
Equal-width Bin edge 99.00
Frequency
Value


Count: 20
Count: 20
Count: 20
Count: 20
Count: 20
Equal-frequency Binning with Bin Counts
Original Data
Equal-frequency Bin edge 1.81
Equal-frequency Bin edge 22.37
Equal-frequency Bin edge 42.82
Equal-frequency Bin edge 62.56
Equal-frequency Bin edge 81.50
Equal-frequency Bin edge 99.00
Frequency
Value