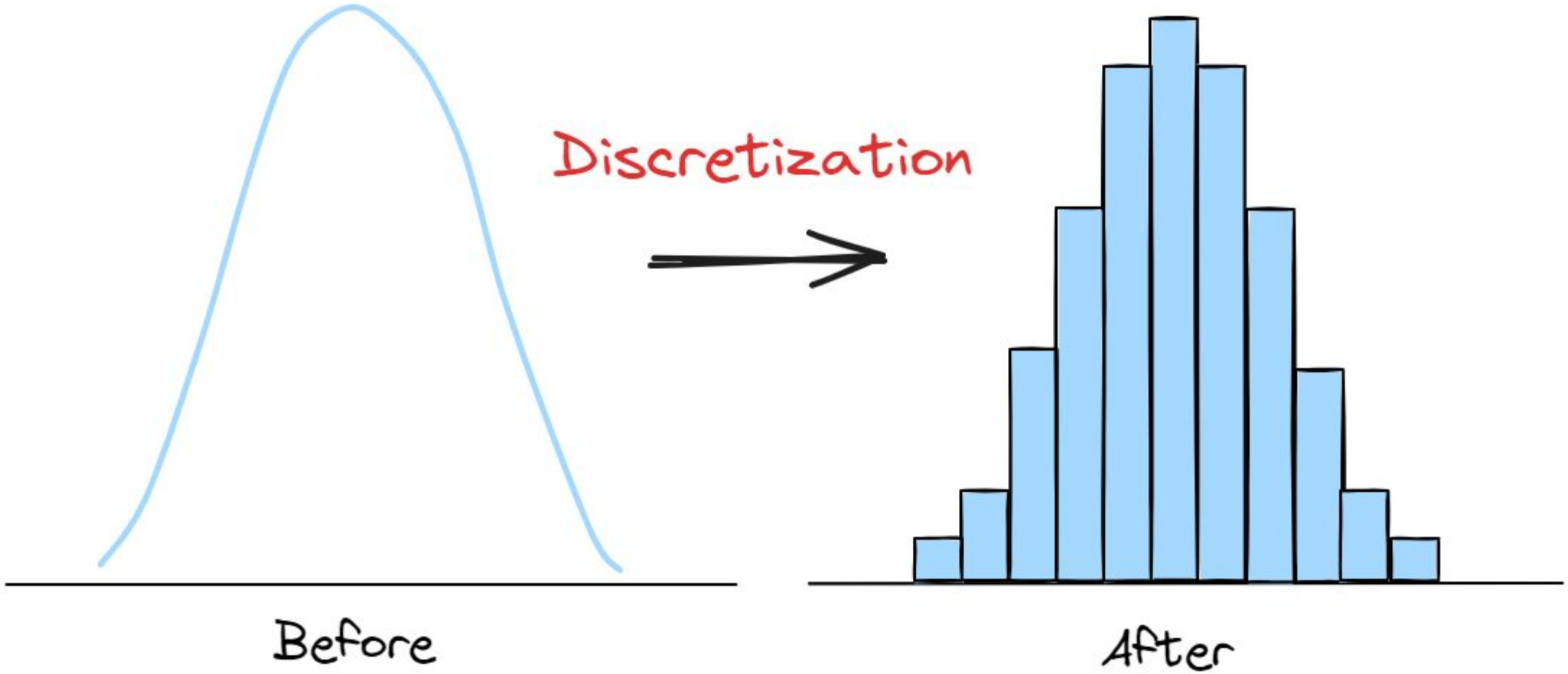
Data Discretization in Data Science



Data discretization

- Data discretization, also known as binning, is the process of grouping continuous values of variables into contiguous intervals. This procedure transforms continuous variables into discrete variables, and it is commonly used in data mining and data science, as well as to train models for artificial intelligence.
- In discretization, we convert continuous variables into discrete features. To do this, we compute the limits of the contiguous intervals that span the entire variable value range.

Continuous vs. Discrete Data:

- **Continuous Data**: Data that can take any value within a range, such as height, weight, temperature, etc.
- **Discrete Data**: Data that can take only specific, distinct values, often categorized into intervals or groups, such as age groups (e.g., 20-30 years, 31-40 years).

Data Binning

Large Continuous Data

Grouped into

Small Discrete Bins

Purpose of Discretization:

- Simplify the data for better understanding and interpretation.
- Reduce noise by grouping similar values.
- Improve the performance of machine learning algorithms that require categorical data.
- Help in identifying meaningful patterns in data.

Methods of Data Discretization

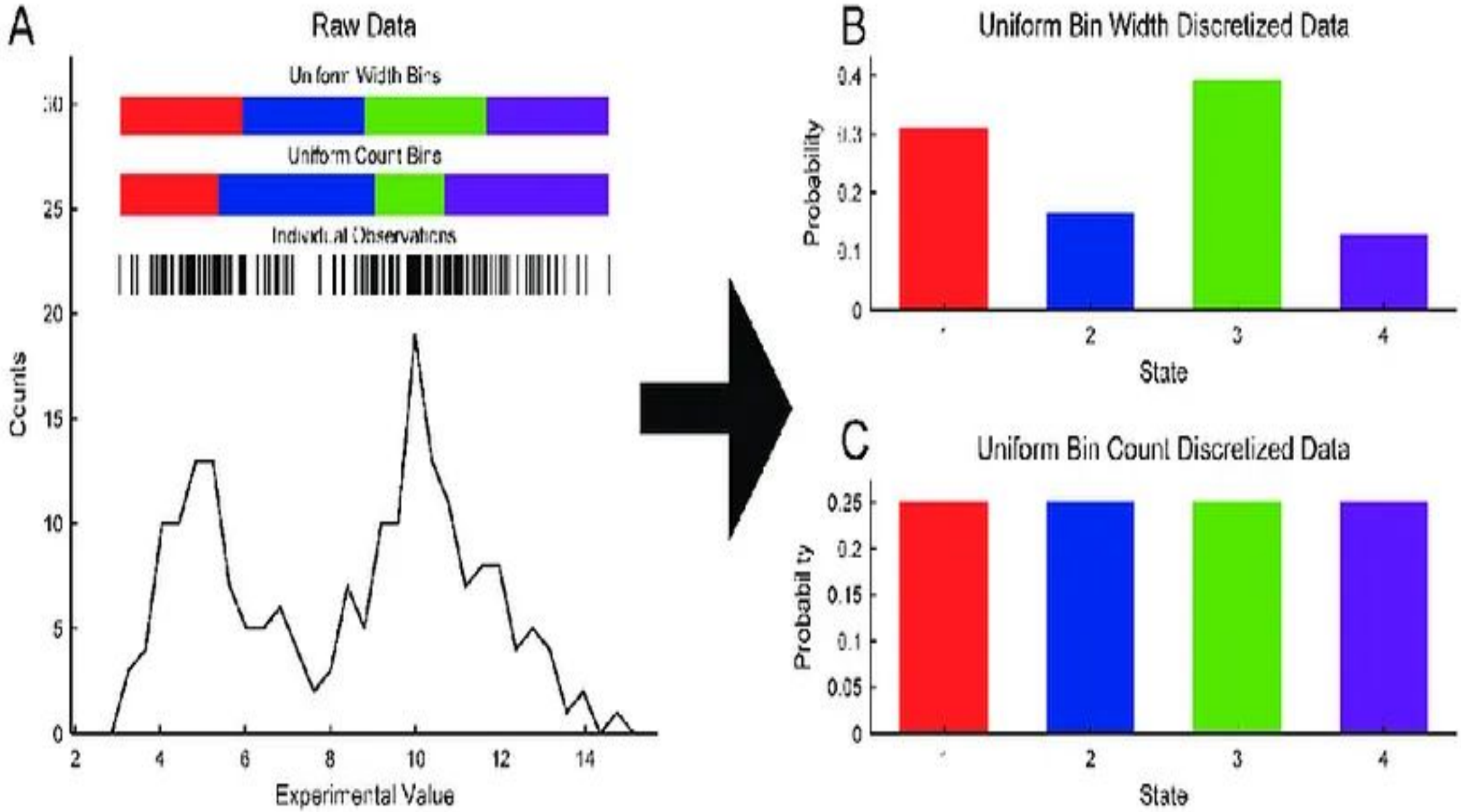
Data discretization can be categorized into two main types: **unsupervised** and **supervised** discretization.

1. Unsupervised Discretization

Unsupervised discretization methods do not consider the output variable(these methods don't consider the target when finding interval limits). They simply divide the continuous data into bins based on certain criteria.

Common techniques include:

- **Equal Width Binning (Interval Binning, Uniform Binning)**:
- **Equal Frequency Binning**:
- **Clustering-Based Binning**:


A
Raw Data
Uniform Width Bins
Uniform Count Bins
Individual Observations
Counts
Experimental Value
B
Uniform Bin Width Discretized Data
Probability
State
C
Uniform Bin Count Discretized Data
Probability
State

Equal Width Binning (Interval Binning):

- **Definition**: The range of the data is divided into equal-sized intervals or bins.
- **Example**: Suppose you have a dataset with ages ranging from 18 to 60. If you want to divide this range into 4 equal-width bins:
  - Bin 1: 18-28 years
  - Bin 2: 29-39 years
  - Bin 3: 40-50 years
  - Bin 4: 51-60 years
- **Use Case**: This method is simple and easy to implement, but it may not always be effective if the data is not uniformly distributed.

Binning

→ Equal Width Binning

$$w = \left(\frac{max - min}{x}\right)$$

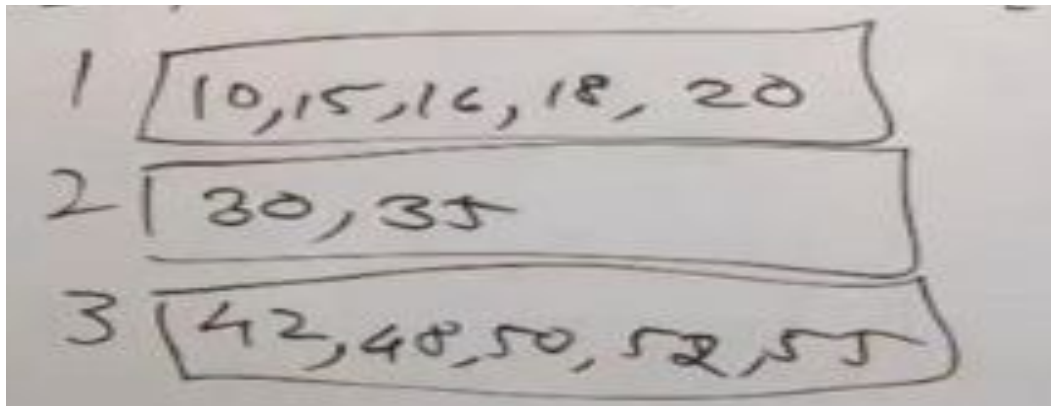
A: [10, 15, 16, 18, 20, 30, 35, 42, 48, 50, 52, 55]

$x = 3 \quad w = \frac{(55 - 10)}{3} = 15$

1 → [min, min + w − 1] ⟿ [10, 10 + 15 − 1 (24)]

2 → [min + w, min + 2w − 1] ⟿ [25, 10 + 30 − 1 (39)]

3 → [min + 2w, max] ⟿ [40, 55]



Equal Frequency Binning

- **Definition**: The data is divided into bins such that each bin contains approximately the same number of data points.
- **Example**: Consider a dataset of 100 student scores ranging from 0 to 100. If you want to create 4 bins:
  - Bin 1: Scores in the range of 0-25 (25 students)
  - Bin 2: Scores in the range of 26-50 (25 students)
  - Bin 3: Scores in the range of 51-75 (25 students)
  - Bin 4: Scores in the range of 76-100 (25 students)
- **Use Case**: This method ensures that all bins have a similar number of observations, making it useful for datasets with skewed distributions.

Binning

→ Equal frequency Binning

$$f = \frac{n}{x} - 3 = \frac{12}{3} (= 4)$$

A. [10, 15, 16, 18, 20, 30, 35, 42, 48, 50, 52, 55]

1 → 10, 15, 16, 18

2 → 20, 30, 35, 42

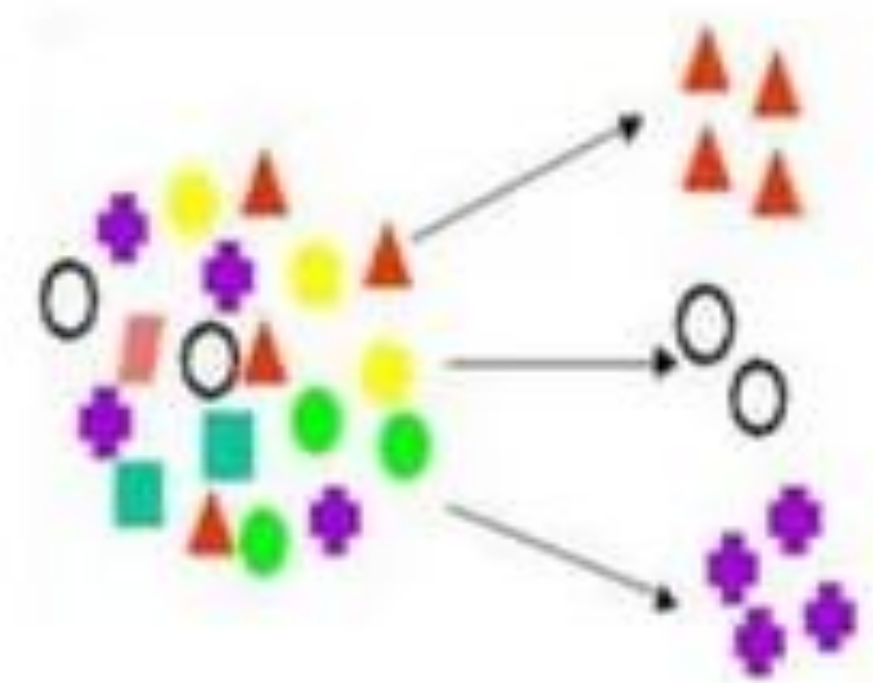
3 → 48, 50, 52, 55

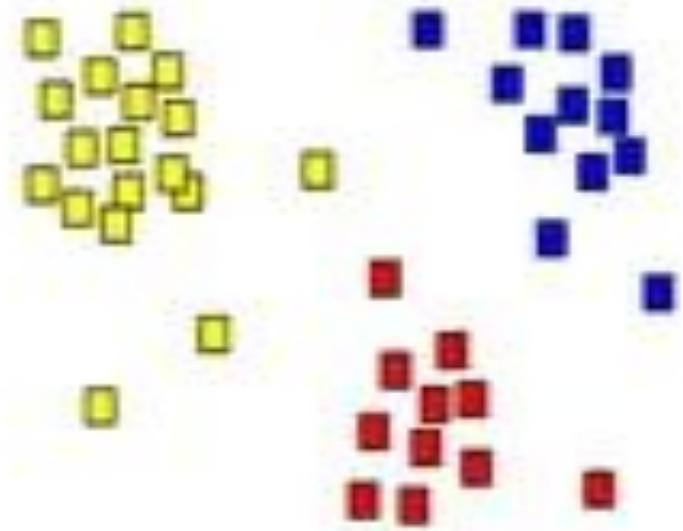
- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28
- **Equal width**
  - Bin 1: 0, 4 [-,10)
  - Bin 2: 12, 16, 16, 18 [10,20)
  - Bin 3: 24, 26, 28 [20,+)
- **Equal frequency**
  - Bin 1: 0, 4, 12 [-, 14)
  - Bin 2: 16, 16, 18 [14, 21)
  - Bin 3: 24, 26, 28 [21,+)

Clustering-Based Binning:

- **Definition**: Clustering algorithms like K-means are used to group data points into clusters, which are then treated as bins.
- **Example**: If a clustering algorithm groups students into clusters based on their performance, each cluster can be treated as a bin, representing students with similar performance levels.
- **Use Case**: This method is useful when the data naturally forms clusters and when you want the bins to reflect these natural groupings.

Examples of Clustering





2. Supervised Discretization

- Supervised discretization methods consider the output variable during the discretization process, ensuring that the bins created are predictive of the output(use the target to find interval limits and the optimal number of divisions).
- **Entropy-Based Binning**:
- **ChiMerge**:
- **Fayyad & Irani's Method**:

Entropy-Based Binning (Decision Tree-Based Discretization):

- **Definition**: This method uses decision tree algorithms to determine the best split points, maximizing information gain relative to the target variable.
- **Example**: Suppose you want to discretize age for predicting whether someone will buy a product. If the decision tree finds that people under 30 and people over 50 are most likely to buy, it may create bins like:
  - Bin 1: Age ≤ 30
  - Bin 2: Age > 30 and ≤ 50
  - Bin 3: Age > 50
- **Use Case**: This method is highly effective in creating bins that are directly relevant to the prediction task.

ChiMerge:

- **Definition**: A chi-square-based method that merges bins with the smallest chi-square values until a stopping criterion is met. This ensures that the bins are statistically independent with respect to the target variable.
- **Example**: Suppose you have income data for predicting loan approval. ChiMerge might merge income ranges that have similar approval rates, resulting in bins like:
  - Bin 1: Income ≤ $30,000
  - Bin 2: Income > $30,000 and ≤ $60,000
  - Bin 3: Income > $60,000
- **Use Case**: ChiMerge is useful when you want to ensure that the bins created are statistically significant in relation to the target variable.

Chi-Square Test

Watch → / Actor ↓	Y	N	(R) Total ↓
M	140	44	184
F	178	38	216
(C) Total →	318	82	400

$$\chi^2_c = \sum \frac{(O_i - E_i)^2}{E_i}$$

$$E = \frac{RT \times CT}{T} = \frac{184 \times 318}{400} = 146$$

$$= (184 \times 82)/400 = 38$$

$$= (216 \times 318)/400 = 172$$

$$= (216 \times 82)/400 = 44$$

H_0: No connection

H_1: full connection

$\alpha = 0.05$

df = (Rows - 1) * (Cols - 1)

Actor	Watch	O	E	$(O-E)^2$	$(O-E)^2/E$
M	Y	140	146	36	0.246
M	N	44	38	36	0.947
F	Y	178	172	36	0.209
F	N	38	44	36	0.818
					2.220

Fayyad & Irani’s Method:

- **Definition**: A recursive partitioning method that selects cut points by maximizing information gain, similar to decision tree splitting.
- **Example**: In a dataset predicting customer churn based on monthly spending, this method might create bins that effectively separate high-risk and low-risk customers based on spending patterns.
- **Use Case**: This method is useful for creating a small number of highly informative bins.

Advantages of Data Discretization

- **Improves Model Interpretability:** Discretization simplifies the data, making it easier to interpret and understand.
- **Reduces Overfitting:** By grouping continuous values into bins, discretization can help reduce the model's complexity and prevent overfitting.
- **Enhances Performance:** Some machine learning algorithms perform better with discrete data, especially when the relationships between variables are non-linear.
- **Supports Certain Algorithms:** Algorithms like Naive Bayes, decision trees, and association rule mining naturally work better with discrete data.

Challenges in Data Discretization

- **Loss of Information:** Discretization can lead to a loss of information, especially when the original continuous data contains meaningful variations within the bins.
- **Determining Optimal Number of Bins:** Choosing the right number of bins is crucial. Too few bins might oversimplify the data, while too many bins might not provide any benefit over the original continuous data.
- **Potential Bias:** Supervised discretization can introduce bias if not done carefully, especially if the bins are not generalizable.

Applications of Data Discretization

- **Preprocessing for Decision Trees:** Many decision tree algorithms work better when the input features are discrete.
- **Data Summarization:** Discretization is used in summarizing data by converting continuous attributes into categorical attributes, making the data easier to understand and interpret.
- **Association Rule Mining:** In market basket analysis, discretization is used to convert continuous variables like purchase amount into categories like “Low,” “Medium,” and “High.”