TUTE SHEET ON

Data Discretization

Data discretization is a process in data science where continuous data is converted into discrete bins or intervals. This is often used in data preprocessing to simplify the data, reduce its size, and prepare it for certain types of analysis or machine learning algorithms that require discrete input features.

Common Methods for Data Discretization

1. **Equal Width Binning (Uniform Binning):**
   - **Formula:** Divide the range of the data into equal-width intervals.
   - **Steps:**

     1. Calculate the range of the data: $\text{Range} = \text{Max} - \text{Min}$
     2. Determine the number of bins (k) you want.
     3. Calculate the bin width: $\text{Bin Width} = \frac{\text{Range}}{k}$
     4. Create intervals based on the bin width.

   Example:

   - Suppose we have a dataset: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20].
   - We choose 4 bins:
     - Bin 1: [2, 6]
     - Bin 2: [7, 11]
     - Bin 3: [12, 16]
     - Bin 4: [17, 20]
     - The value 20 would be included in an additional bin or extend the last bin.

Example: Equal Width Binning

Dataset:

Consider a dataset of the ages of 15 individuals: 18,22,25,28,30,35,37,40,42,45,50,52,55,60,65

Step 1: Determine the Range of the Data

First, find the minimum and maximum values in the dataset.

- **Min** = 18
- **Max** = 65
- **Range** = Max - Min = 65−18=47

Step 2: Decide the Number of Bins

Let's say we want to divide this data into 4 bins.

Step 3: Calculate the Bin Width

The bin width is calculated as:

$$\text{Bin Width} = \frac{\text{Range}}{\text{Number of Bins}} = \frac{47}{4} \approx 11.75$$

For simplicity, we can round this to 12, so the bin width will be 12.

Step 4: Create the Intervals

Now, we create the intervals based on the bin width.

1. **Bin 1**: [18, 30] (18 to just before 30)
2. **Bin 2**: [30, 42] (30 to just before 42)
3. **Bin 3**: [42, 54] (42 to just before 54)
4. **Bin 4**: [54, 66] (54 to just before 66)

Note that the last bin goes up to 66 to include the maximum value of 65.

Step 5: Assign the Data to Bins

Now, place each data point into the appropriate bin:

- **Bin 1**: [18, 22, 25, 28]
- **Bin 2**: [30, 35, 37, 40]
- **Bin 3**: [42, 45, 50, 52]
- **Bin 4**: [55, 60, 65]

Final Result:

The dataset has been discretized into 4 bins with the following intervals:

1. **Bin 1 (18 to just before 30):** [18, 22, 25, 28]
2. **Bin 2 (30 to just before 42):** [30, 35, 37, 40]
3. **Bin 3 (42 to just before 54):** [42, 45, 50, 52]
4. **Bin 4 (54 to 66):** [55, 60, 65]

2. **Equal Frequency Binning (Quantile Binning):**
    - **Formula:** Divide the data such that each bin contains the same number of data points.
    - **Steps:**

        1. Sort the data.
        2. Determine the number of bins (k).
        3. Divide the sorted data into k bins with an equal number of elements.

    - **Example:**

        - Suppose we have the same dataset: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20].
        - We choose 4 bins:
            - Bin 1: [2, 4, 6]
            - Bin 2: [8, 10, 12]
            - Bin 3: [14, 16]
            - Bin 4: [18, 20]

Example: Equal Frequency Binning (Quantile Binning)

Dataset:

Consider a dataset of exam scores for 12 students: 45,50,52,55,60,62,65,70,72,75,80,85

Step 1: Decide the Number of Bins

Let's say we want to divide the data into 3 bins.

Step 2: Sort the Data

First, sort the data (if it's not already sorted): 45,50,52,55,60,62,65,70,72,75,80,85

Step 3: Determine the Number of Data Points per Bin

The number of data points per bin is calculated as:

$$\text{Number of Data Points per Bin} = \frac{\text{Total Number of Data Points}}{\text{Number of Bins}} = \frac{12}{3} = 4$$

So, each bin will have 4 data points.

Step 4: Create the Bins

Now, divide the sorted data into 3 bins, each containing 4 data points.

1. **Bin 1**: [45, 50, 52, 55]
2. **Bin 2**: [60, 62, 65, 70]
3. **Bin 3**: [72, 75, 80, 85]

Step 5: Assign the Data to Bins

Each data point is now grouped into its corresponding bin based on the number of data points per bin.

Final Result:

The dataset has been discretized into 3 bins with the following intervals:

1. **Bin 1 (Low Scores):** [45, 50, 52, 55]
2. **Bin 2 (Mid Scores):** [60, 62, 65, 70]
3. **Bin 3 (High Scores):** [72, 75, 80, 85]

2. **Clustering-based Binning (e.g., using K-means):**
    - **Formula:** Use clustering algorithms to group data into bins based on similarity.
    - **Steps:**

        1. Apply a clustering algorithm like K-means.
        2. The data points are assigned to the nearest cluster center.
        3. Each cluster is treated as a bin.

    - **Example:**

        - If you have data [1, 2, 10, 11, 20, 21], applying K-means with k=3 might yield clusters:
            - Cluster 1: [1, 2]
            - Cluster 2: [10, 11]
            - Cluster 3: [20, 21]

Example: Clustering-based Binning Using K-means

Dataset:

Consider a dataset of 15 house prices (in thousands):
100,105,110,115,120,125,200,205,210,215,220,300,305,310,315

Step 1: Determine the Number of Clusters

Let's say we want to create 3 clusters (which will correspond to 3 bins).

Step 2: Apply the K-means Algorithm

K-means clustering works by minimizing the variance within clusters and maximizing the variance between clusters. Here's how it works:

1. **Initialization**: Randomly initialize 3 cluster centroids.
2. **Assigning Points to Clusters**: Assign each house price to the nearest centroid.
3. **Updating Centroids**: Calculate the new centroid of each cluster based on the mean of the points assigned to it.
4. **Reassignment**: Reassign points to the nearest centroid and repeat the process until the centroids no longer change significantly.

Step 3: Resulting Clusters

Assuming that after running K-means, we get the following clusters:

- **Cluster 1** (Low prices): [100, 105, 110, 115, 120, 125]
- **Cluster 2** (Mid prices): [200, 205, 210, 215, 220]
- **Cluster 3** (High prices): [300, 305, 310, 315]

Step 4: Create Bins Based on Clusters

Each cluster identified by K-means corresponds to a bin:

1. **Bin 1 (Low prices):** [100, 105, 110, 115, 120, 125]
2. **Bin 2 (Mid prices):** [200, 205, 210, 215, 220]
3. **Bin 3 (High prices):** [300, 305, 310, 315]

Final Result:

The dataset has been discretized into 3 bins based on clustering:

1. **Bin 1 (Cluster 1):** Lower house prices: [100, 105, 110, 115, 120, 125]
2. **Bin 2 (Cluster 2):** Middle house prices: [200, 205, 210, 215, 220]
3. **Bin 3 (Cluster 3):** Higher house prices: [300, 305, 310, 315]

Example of Data Discretization

Consider a dataset containing ages: `[22, 25, 28, 30, 35, 37, 40, 45, 50, 52, 55]`.

Equal Width Binning (3 bins):

- Range = 55−22=33
- Bin width = 33/3=11
- Bins:
    - Bin 1: [22, 33]
    - Bin 2: [33, 44]
    - Bin 3: [44, 55]

Resulting bins:

- Bin 1: [22, 25, 28, 30]
- Bin 2: [35, 37, 40]
- Bin 3: [45, 50, 52, 55]

Equal Frequency Binning (3 bins):

- Sorted data: [22, 25, 28, 30, 35, 37, 40, 45, 50, 52, 55]
- Bins:
    - Bin 1: [22, 25, 28, 30]
    - Bin 2: [35, 37, 40, 45]
    - Bin 3: [50, 52, 55]

These methods can be selected based on the nature of the data and the requirements of the analysis or modeling process.

Common Methods for Supervised Data Discretization

1. **Entropy-Based Discretization (Decision Tree-based):**
    - This method uses the concept of information gain (used in decision trees) to determine the best points to split the continuous attribute into discrete intervals.
    - **Steps:**
        1. For each potential split point in the data, calculate the information gain.
        2. Choose the split that provides the highest information gain.
        3. Repeat this process recursively to create bins, stopping when the information gain is below a certain threshold or when a stopping criterion (e.g., minimum number of samples in a bin) is met.
    - **Example:**

- Suppose you have the following data on age with a binary target (say, "Will Buy" and "Won't Buy"):

```
Age: [22, 25, 28, 30, 35, 37, 40, 45, 50, 52, 55]
Target: [Yes, No, Yes, Yes, No, Yes, No, Yes, No, Yes, Yes]
```

- The algorithm would evaluate potential split points in the "Age" feature to find where the information gain is maximized concerning the target variable. The data would be split into bins accordingly.

2. **ChiMerge:**
   - ChiMerge is a chi-square statistic-based method where continuous data is discretized by merging intervals that have similar class distributions.
   - **Steps:**

     1. Sort the data by the continuous attribute.
     2. Each unique value starts as its own interval.
     3. Compute the chi-square statistic for every pair of adjacent intervals.
     4. Merge the pair with the smallest chi-square value, indicating that the class distribution between the two intervals is not significantly different.
     5. Repeat until a stopping criterion (e.g., a chi-square threshold or a desired number of intervals) is met.

   - **Example:**

     - Using the same "Age" dataset:

```
Age: [22, 25, 28, 30, 35, 37, 40, 45, 50, 52, 55]
Target: [Yes, No, Yes, Yes, No, Yes, No, Yes, No, Yes, Yes]
```

     - The ChiMerge method would start with each age value as a separate bin and then iteratively merge adjacent bins that have similar distributions of the target class ("Yes" or "No"), resulting in larger intervals that maintain predictive power.

2. **Class-Attribute Interdependence Maximization (CAIM):**
   - CAIM aims to maximize the dependence between the class labels and the discretized intervals of the continuous attribute.
   - **Steps:**

     1. For each possible split point, calculate the CAIM criterion, which measures the interdependence between the class labels and the intervals.
     2. Select the split that maximizes the CAIM criterion.
     3. Continue splitting until no further improvement in the CAIM criterion is possible.

   - **Example:**

     - Similar to the other methods, you would apply CAIM to the "Age" dataset to find the splits that best differentiate the class labels "Will Buy" and "Won't Buy."

Example of Supervised Data Discretization (Using Entropy-Based Discretization)

Consider the following example:

- **Dataset:**

```
Age: [22, 25, 28, 30, 35, 37, 40, 45, 50, 52, 55]
Target: [Yes, No, Yes, Yes, No, Yes, No, Yes, No, Yes, Yes]
```

- **Steps:**
  1. **Calculate Entropy for the entire dataset:**
     - 6 "Yes" and 5 "No", so entropy $H(S)$ is:

$$H(S) = -\left(\frac{6}{11}\right)\log_2\left(\frac{6}{11}\right) - \left(\frac{5}{11}\right)\log_2\left(\frac{5}{11}\right)$$

  2. **Evaluate potential split points (e.g., Age = 30):**
     - Calculate the entropy for data subsets (Age ≤ 30 and Age > 30) and find the information gain.
     - The split that maximizes the information gain is chosen.
  3. **Repeat recursively for the resulting intervals.**
- **Result:**
  - You might end up with intervals like:
    - Age ≤ 30: Mostly "Yes"
    - 30 < Age ≤ 45: Mixed
    - Age > 45: Mostly "No"