BITS Pilani

BITS Pilani
Hyderabad Campus

Prof.Aruna Malapati
Department of CSIS

innovate achieve lead

BITS Pilani
Hyderabad Campus

Data Preprocessing

Today's Learning objective

- Overview of data Preprocessing approaches
- Data Cleaning
- Feature Extraction
- Data Reduction

Data pre-processing

1. Data cleaning: handling errors and missing values
2. Feature extraction: creating new features by combining and transforming existing ones
   - a crucial step! ⇒ what patterns can you find application-specific require understanding of the domain
3. Data reduction
   - sampling
   - feature selection
   - dimension reduction by transformations

Data Cleaning

- Strategies to handle Missing values
  - If a feature has many missing values, prune the feature with correct values.
  - If a record has many missing values, prune the record
  - Impute missing values
  - If the modeling technique allows missing values, just replace them with special values (like “NA”)

Feature extraction

- scaling and normalization: numerical → numerical
- discretization: numerical → categorical
- binarization: categorical → binary (0/1)
- creating similarity graphs: any type → graph
- transformations for dimension reduction: create new, less redundant features and keep the best ones, both feature extraction and data reduction

Scaling and Normalization

- Features with large magnitudes dominate the aggregate functions like Euclidean distances.
- Hence, we can transform all features to the same scale or standardize distributions.
- Normalization is particularly useful for classification algorithms.
  - min-max normalization
  - z-score normalization
  - Normalization by decimal scaling

Scaling and Normalization (Contd..)

- min-max scaling:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \text{(new range [0, 1])}$$

- mean normalization:

$$y = \frac{x - mean(x)}{\max(x) - \min(x)} \quad \text{(new range } [-1, 1], \ mean(y) = 0)$$

- Beware! min and max may be outliers

Min-max normalization

Transform the data from measured units to a new interval from new_minF to new_maxF for feature :

$$v' = \frac{v - min_F}{max_F - min_F}(new\_max_F - new\_min_F) + new\_min_F$$

where v is the current value of feature F.

Suppose that the minimum and maximum values for the feature income are \$120,000 and \$98,000, respectively. We would like to map income to the range 0.0,1.0 . By min-max normalization, a value of \$73,600 for income is transformed to:

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0.0) + 0 = 0.716$$

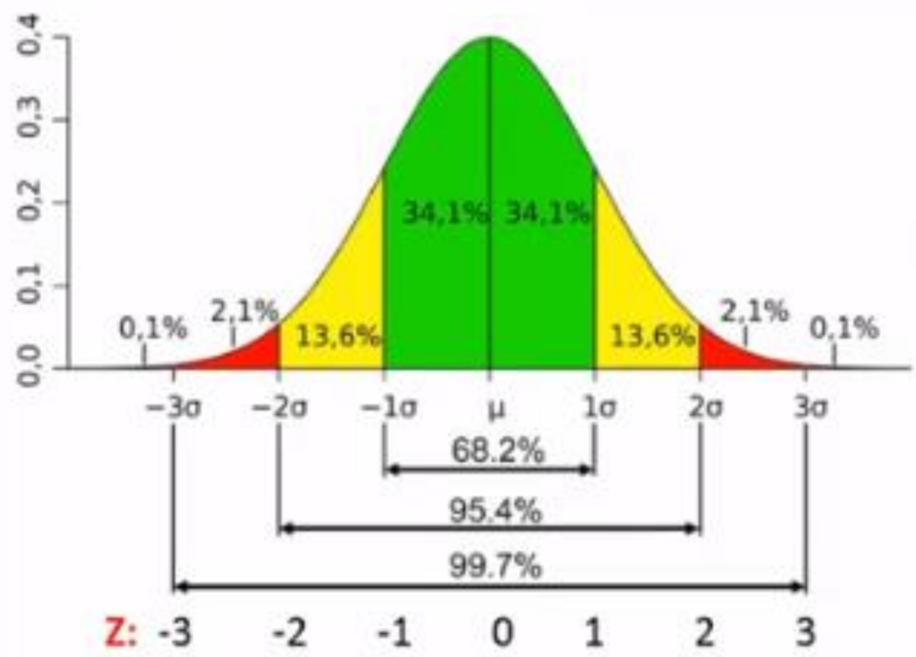
Standardization or Z-Score Normalization

If the distribution is normal:

$$z = \frac{x - mean(x)}{stdev(x)}$$

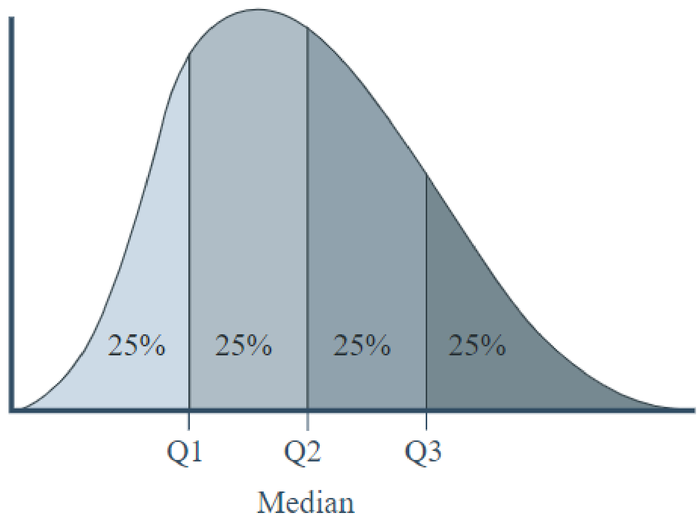
$$mean(z) = 0$$

$$stdev(z) = 1$$



Robust Scaling

- If many outliers mean and stdev are biased ⇒ robust scaling using median and interquartile range:

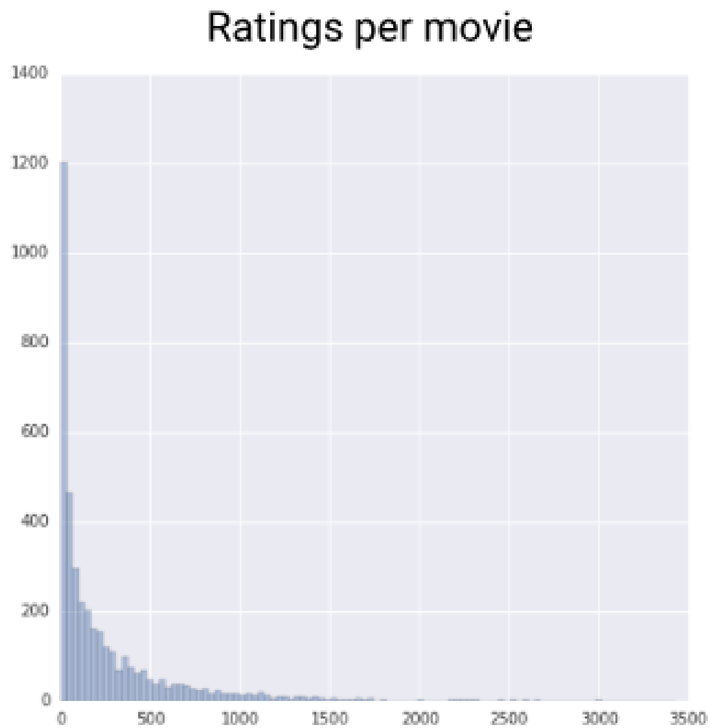


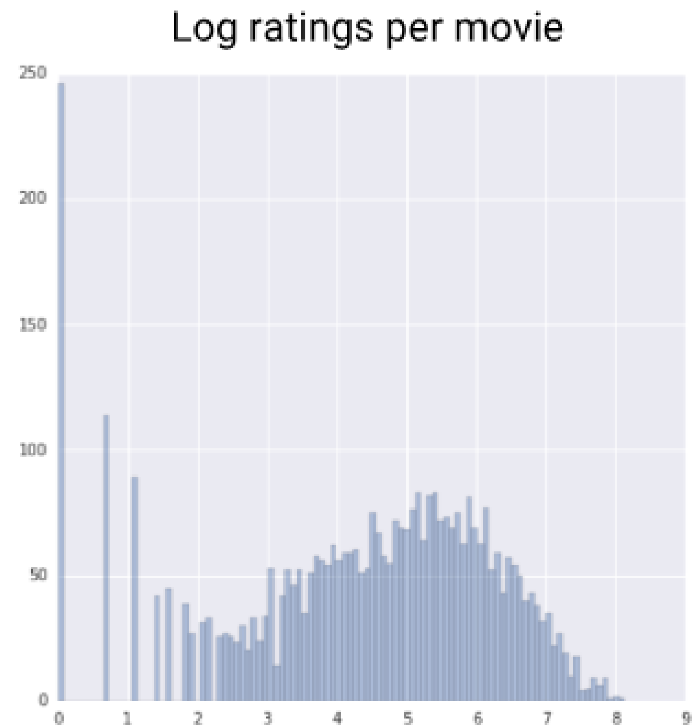
$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)}$$

- Lower Quartile (QL) or First Quartile (Q1) : 25% of the data falls below this percentile 50th percentile
- Median or Second Quartile (Q2) : 50% of the data falls below this percentile 75th percentile
- Upper Quartile (QU) or Third Quartile (Q3): 75% of the data falls below this percentile

Log Transformation

- Sometimes y = log2(x) helps to make distribution less skewed or even normal.





Discretization
numerical → categorical

- Discretization of continuous attributes is most often performed one attribute at a time, independent of other attributes.
- This approach is known as static attribute discretization on the other end of the spectrum is dynamic attribute discretization, where all attributes are discretized simultaneously while taking into account the interdependencies among them.

Discretization

- Unsupervised discretization
  - Equal-interval binning
  - Equal-frequency binning

- Class labels are ignored
- The best number of bins k is determined experimentally

- Supervised discretization
  - Entropy-based discretization
  - It tries to maximize the “purity” of the intervals (i.e. to contain as less as possible mixture of class labels)

Unsupervised Discretization

- Require the user to specify the number of intervals and/or how many data points should be included in any given interval.
- The following heuristic is often used to choose intervals:
  - The number of intervals for each attribute should not be smaller than the number of classes (if known).
  - The other popular heuristic is to choose the number of intervals, n_{Fi}, for each attribute, F_i (i=1,…,n,) where n is the number of attributes), as follows: $n_{Fi} = M/3 * C$ where M is the number of training examples and C is the number of known categories.

Unsupervised Discretization

- **Equal-interval binning**
  - Divide the attribute values x into k equally sized bins
  - If $\mathbf{x_{min}} \le \mathbf{x} \le \mathbf{x_{max}}$ then the bin width $\boldsymbol{\delta}$ is given by

$$\delta = \frac{x_{max} - x_{min}}{k}$$

  - Construct bin boundaries at $\mathbf{x_{min}} + \mathbf{i\delta}$, i = 1,…, k-1

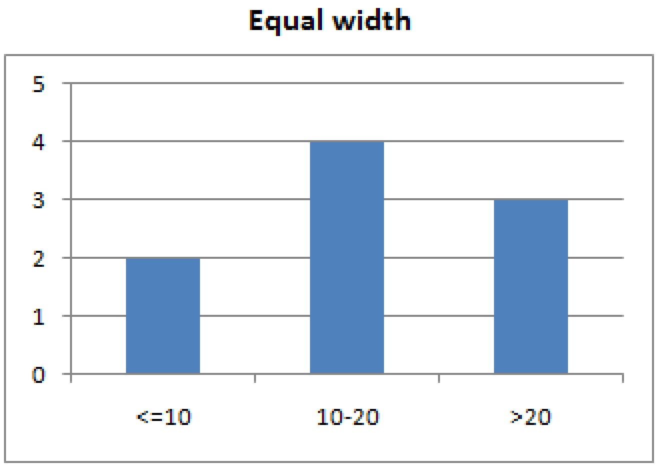
- **Disadvantage**: Outliers can cause problems

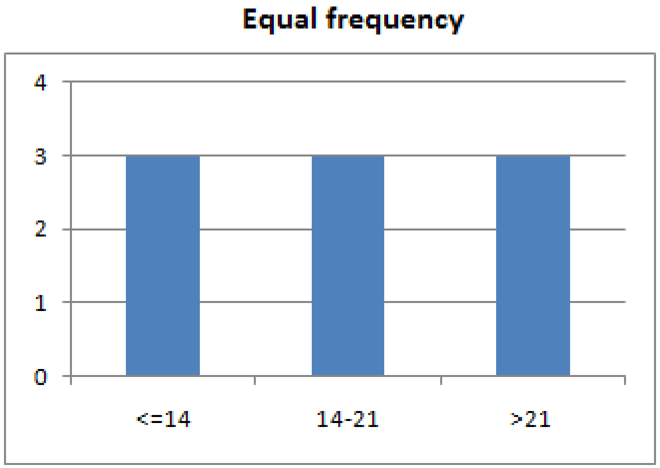
Unsupervised Discretization

- Equal-frequency binning
- An equal number of values are placed in each of the k bins.
- Disadvantage: Many occurrences of the same continuous value could cause the values to be assigned into different bins.

Example

- **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,+)





Supervised Discretization

- Suppose you are analyzing risk of Alzheimer's disease and you split age data at age 16, age 24, and age 30.

  Your bins look something like this:

  <=16

  16...24

  24...30

  >30

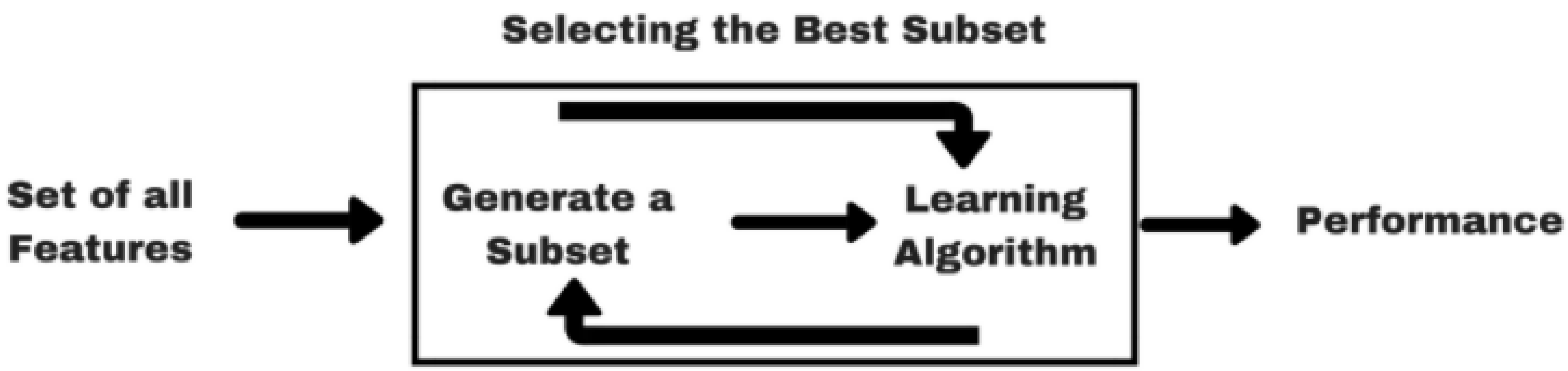
  Now you have a giant bin of people older than 30, where most Alzheimers patients are, and multiple bins split at lower values, where you're not really getting much information.

- Because of this issue, we want to make **meaningful splits** in our continuous variables.

Feature Subset Selection Techniques

- Brute-force approach: Try all possible feature subsets as input to data mining algorithm
- Filter approaches: Compute a score for each feature and then select features according to the score.
- Wrapper approaches: score feature subsets by seeing their performance on a dataset using a classification algorithm.
- Embedded approaches: Select features during the process of training.

Wrapper Methods



Sequential Forward Selection (SFS)

1. Start with an empty feature set
2. Try each remaining feature
3. Estimate classifier performance for adding each feature
4. Select feature that gives max improvement
5. Stop when there is no significant improvement

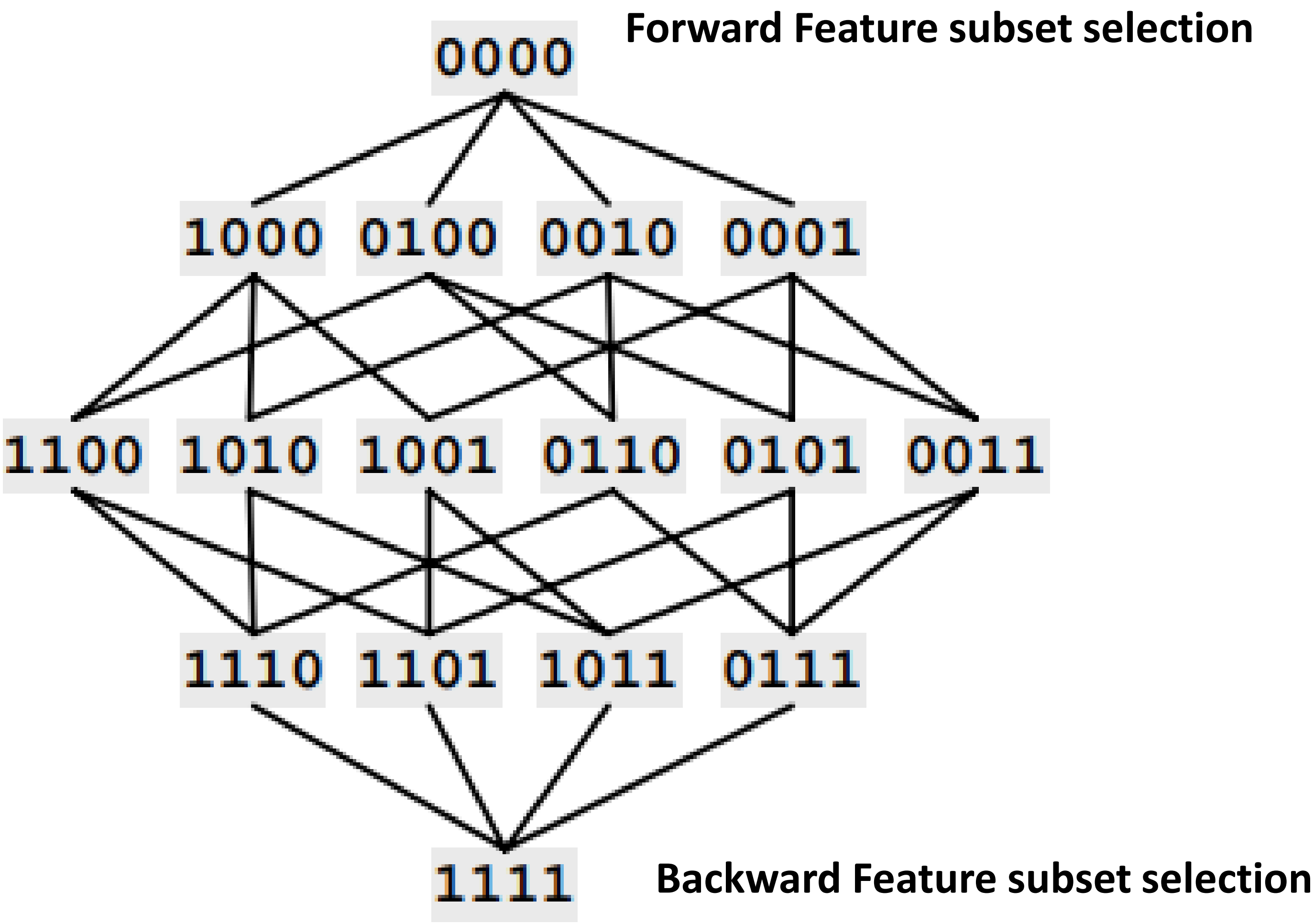
Disadvantage: Once a feature is retained, it cannot be discarded; nesting problem

Sequential Backward Selection (SBS)

1. Start with an full feature set
2. Try removing feature
3. Drop the feature with smallest impact on classifier performance

Disadvantage: SBS requires more computation than SFS

Search space for feature selection



Embedded Method for feature selection

- Embedded methods perform feature selection and training of the algorithm in parallel.
- Example
- Lasso Regression
- Decision Trees

Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes
- Three general methodologies:
  - Feature Extraction: multimedia features(low,middle,high level fetaures)
    - domain-specific
  - Mapping Data to New Space
  - Feature Construction: combining features

Take home message

- Missing values can be handled by eliminating features or records or by imputation methods.
- Feature extraction methods like scaling, normalization, and discretization need to be applied based on the problem.
- Data reduction methods will be applied to reduce the number of features required to build the model.