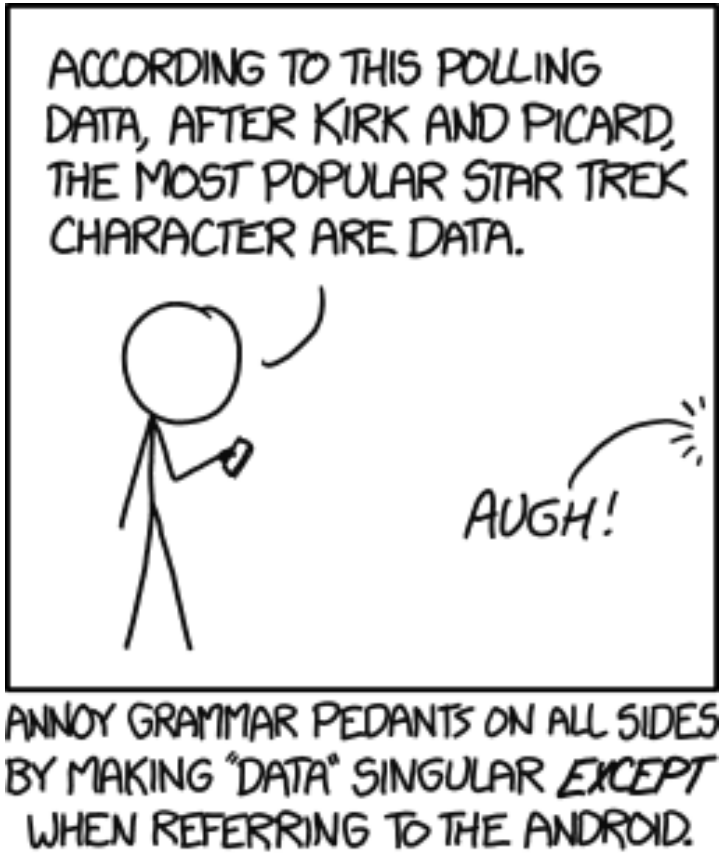


Image. xkcd comics - # 1429

Data Mining: Data

Data Types, Data Preprocessing

CS 4821 - CS 5831 - s24

some slides adapted from: G. Piatetsky-Shapiro; Han, Kamber, & Pei; P. Smyth; C. Volinsky; A. Mueller; Tan, Steinbach, & Kumar; J. Taylor; G. Dong; M. Zaki & W. Meria

Data, Types of Attributes

Data Preparation Overview

Data Preperation - Cleaning / Missing Data

Data Preparation - Integration

Data Preparation - Reduction

Data Transformation

Data, Types of Attributes

What is Data?

- Data comes in many different forms for analysis:
  - Tabular Data
  - Relational Data
  - Time-series data
  - Spatial data
  - Graph data
  - ...

Tabular Data

- Collection of data objects and their attributes

- Row: A collection of attributes describe an object
  aka: record, sample, instance
- Column: An attribute (or feature) is a property or characteristic of an object
  aka: variable, field

Attributes

Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Attribute Values

- Attribute values are numbers or symbols assigned to an attribute

- Distinction between attributes and attribute values
  - Same attribute can map to different attribute values
    Ex. Height can be measured in feet or meters
  - Different attributes can map to same set of values
    Ex. Attribute values for ID and age are both integers
    Properties of attribute values can be different

Types of Attributes

- Categorical / Qualitative: set-valued domain
  - Nominal - Categories, states, or "names of things"
    Examples: ID numbers, eye color, zip codes, ...
  - Ordinal - Values have a meaningful order (ranking), but magnitude between successive values is not known
    Examples: grades, sizes, Likert items

- Numerical / Quantitative: real-valued or integer-valued domain
  - Interval - Measured on a scale of fixed and equal-sized units; no true zero-point
    Examples: Temperature in degree F/C, calendar dates
  - Ratio - Measured but with a defined zero point
    Examples: Temp. in Kelvin, length, counts, money

Types of Attributes

Attribute Type	Description	Examples	Operations
Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. (=,)	zip codes, employee ID numbers, eye color, sex: {*male, female*}	mode, entropy, contingency correlation, χ^2 test
Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {*good, better, best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Types of Attributes

Attribute Level	Transformation	Comments
Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
Ordinal	An order preserving change of values, i.e., *new_value = f(old_value)* where *f* is a monotonic function.	An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by { 0.5, 1, 10}.
Interval	*new_value =a * old_value + b* where a and b are constants	Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
Ratio	*new_value = a * old_value*	Length can be measured in meters or feet.

Types of Attributes: Another Perspective

- Discrete - has only a finite or countably infinite set of values
  Examples: zip codes, counts, binary values
  Representation: integer
- Continuous - real number of values
  Examples: temperature, height, weight
  Representation: floating point

Tabular Data

Data that consists of collection of records, each of which have a fixed set of attributes.

- Perhaps, the simplest format to deal with is a "flat file" or a data table
  - In 'R', a data frame object
  - In 'Python', a 'pandas' DataFrame, or 'numpy' array
  - In 'Matlab', a table
- If data objects, have same fixed set of numeric attributes, then data objects are points in a multi-dimensional space (each dimension is a distinct attribute)
  - This data can be represented by a $n \times p$ **data matrix**

Data Matrix

- A data set is a collection of data objects representing entities
- Data objects are described by attributes

$$\mathbf{X}_{n\times p} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{bmatrix}$$

- n is the number of objects, samples, samples, etc. $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ip})$
- p is the number of features, attributes, variables, etc. $X_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$

Algebraic View

For a data matrix $\mathbf{X}_{n\times p}$, each row is a p-dimensional vector

$$\mathbf{x}_i = (x_{i1} \;\; x_{i2} \;\; \cdots \;\; x_{ip}) \in \mathbb{R}^p$$

where each column or attribute is a n-dimensional vector:

$$X_j = (x_{1j} \;\; x_{2j} \;\; \cdots \;\; x_{nj})^T \in \mathbb{R}^n$$

Example: Data Matrix

Table 1: University Data

ID	GPA	ACT
12345678	3.50	28
87654321	3.72	30
24687531	3.17	25
13578642	2.98	29
83576124	3.42	32

For 5 students we have their `ID`, `GPA`, and `ACT` this would then be the matrix $\mathbf{X}_{5\times 3}$

Example: Data Matrix (2)

- If we add `Major` to the data set, then there is also a nominal (categorical or discrete) variable.

Table 2: University Data

ID	GPA	ACT	Major
12345678	3.50	28	CS
87654321	3.72	30	Math
24687531	3.17	25	ECE
13578642	2.98	29	MIS
83576124	3.42	32	ME

Data Preparation Overview

Data Preparation

- Real world data is dirty
  - incomplete - lacking attribute values, lacking attributes for the problem
  - noisy - may have errors and outliers
  - inconsistent - discrepancies in how it was captured
- Without quality data, results of data mining may be worthless

Garbage in, garbage out

- Investigate your data
- Prepare data for next steps of analysis

Tasks in Data Preprocessing

- Data Cleaning
  - check data quality
  - missing data
- Data Integration
  - integration of multiple data sources
- Data Reduction
  - obtain a reduced representation of data that produces same/similar results
- Data Transformation and Discretization
  - normalize, discretize, etc.; prepare data for additional analyses

Data Cleaning

- incomplete: lacking attribute values, lacking certain attributes of interest, only has aggregate data
  - e.g., Occupation = " " (missing data)
- noisy or outliers: containing errors or outliers
  - e.g., Salary="-10" (an error); Compensation="$181K/hr" (outlier)
- inconsistent: contains discrepancies in codes or names
  - e.g., Age="42", Birthday="03/07/2009"
  - e.g., ratings using both "1, 2, 3", and "A, B, C"
- intentional: disguised missing data
  - e.g., every record has Birthday="Jan 1"
- duplication: duplicate data
- systemic vs. random: aspect to consider for noise, missing data, etc.

Data Preperation - Cleaning / Missing Data

Missing Data

- Data is not always available
- Reasons for missing data:
  - equipment malfunction
  - data not entered properly
  - data not available
  - data processing error
  - ...
- Missing data itself may have significance

Handling Missing Data

- What do you do?
  - **There is no single right answer!**

- What to do depends on
  - how much data is missing? are they important values?
  - the techniques to be used, can they handle missing values?
  - is the data missing at random?
  - try to check robustness of results

Handling Missing Data

- Missing values can be encoded in many ways
  - “”
  - ” ”
  - “NA”
  - “N/A”
  - ?
  - ???
  - “Unknown”
  - -1
  - ...
- Often missingness itself is informative

Handling Missing Data - Drop Row

- Drop the sample that has missing data
  - use when the values are missing at random
  - possible if small % of data is missing;

ID	City	Degree	Age	Married ?
1	Lisbon	NaN	25	0
2	Berlin	Bachelor	25	1
3	Lisbon	NaN	30	1
4	Lisbon	Bachelor	30	1
5	Berlin	Bachelor	18	0
6	Lisbon	Bachelor	NaN	0
7	Berlin	Masters	30	1
8	Berlin	No Degree	NaN	0
9	Berlin	Masters	25	1
10	Madrid	Masters	25	1

ID	City	Degree	Age	Married ?
2	Berlin	Bachelor	25	1
4	Lisbon	Bachelor	30	1
5	Berlin	Bachelor	18	0
7	Berlin	Masters	30	1
9	Berlin	Masters	25	1
10	Madrid	Masters	25	1

- Issues with this approach:
  - if all missing data is eliminated and this data is used to train a model; then, if testing data also has missing elements the model can not handle this
  - if missingness relates to outcome and drop all samples with missing values, you are biasing the results

Handling Missing Data - Drop Column

- Drop the column that has missing data
  - use when the values are missing at random
  - rule of thumb remove when the missing values are significantly more than the other values

ID	City	Salary	Married ?
1	Lisbon	45,000	0
2	Berlin	NaN	1
3	Lisbon	NaN	1
4	Lisbon	NaN	1
5	Berlin	NaN	0
6	Lisbon	NaN	0
7	Berlin	NaN	1
8	Berlin	NaN	0
9	Berlin	NaN	1
10	Madrid	NaN	1

ID	City	Married ?
1	Lisbon	0
2	Berlin	1
3	Lisbon	1
4	Lisbon	1
5	Berlin	0
6	Lisbon	0
7	Berlin	1
8	Berlin	0
9	Berlin	1
10	Madrid	1

Handling Missing Data - Imputation

- Fill in missing value
  - Use a global constant: *unknown*
  - Use attribute information: the attribute mean/mode/median
  - Use attribute & class information: the attribute mean for all samples in the same class
  - Use a random value
  - Use learning-based imputation
    Nearest neighbor or model-based

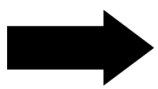
Missing Data - Mean/Median Imputation

Method involves replacing the missing value with a measure of central tendency of the column where the missing value is from

- Numeric Variables - replace with Mean

Average_Age = 26.0

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1



ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	26	0
7	Berlin	30	1
8	Berlin	26	0
9	Berlin	25	1
10	Madrid	25	1

Missing Data - Mean/Median Imputation

Method involves replacing the missing value with a measure of central tendency of the column where the missing value is from

- Categorical Variables - replace with Mode

Most frequent Degree = 'Bachelor'

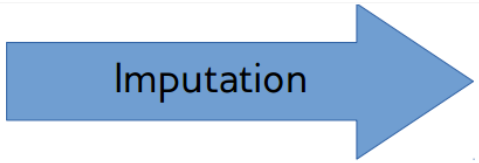
ID	City	Degree	Married ?
1	Lisbon	NaN	0
2	Berlin	Bachelor	1
3	Lisbon	NaN	1
4	Lisbon	Bachelor	1
5	Berlin	Bachelor	0
6	Lisbon	Bachelor	0
7	Berlin	Masters	1
8	Berlin	No Degree	0
9	Berlin	Masters	1
10	Madrid	Masters	1

ID	City	Degree	Married ?
1	Lisbon	Bachelor	0
2	Berlin	Bachelor	1
3	Lisbon	Bachelor	1
4	Lisbon	Bachelor	1
5	Berlin	Bachelor	0
6	Lisbon	Bachelor	0
7	Berlin	Masters	1
8	Berlin	No Degree	0
9	Berlin	Masters	1
10	Madrid	Masters	1

Example: Median Imputation

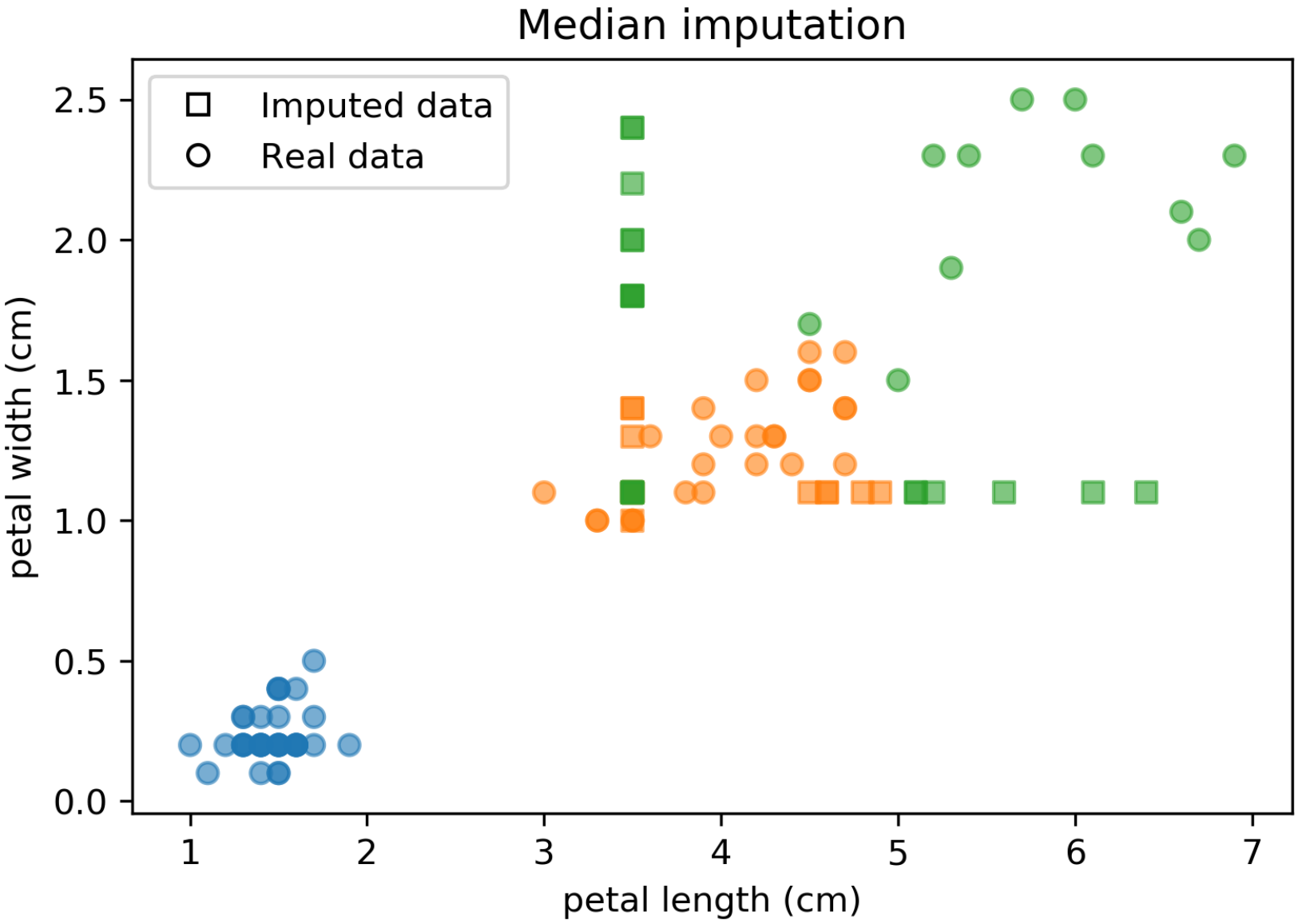
```
[[ 6.   2.9  4.5  1.5]
 [ 5.9  3.   5.1  1.8]
 [ 4.4  3.   1.3  0.2]
 [ 5.1  3.3  nan  nan]
 [ 5.   3.5  1.6  0.6]
 [ 5.4  3.4  nan  nan]
 [ 5.7  3.8  nan  0.3]
 [ 5.6  2.5  3.9  nan]
 [ 7.7  2.6  6.9  2.3]
 [ 5.8  2.7  5.1  1.9]
 [ 6.7  3.1  5.6  2.4]
 [ 4.8  3.4  1.9  nan]
 [ 7.2  3.2  6.   1.8]
 [ 4.4  2.9  nan  nan]
 [ 6.9  3.2  5.7  2.3]
 [ 5.5  4.2  1.4  nan]
 [ 6.3  2.3  4.4  1.3]
 [ 7.   3.2  4.7  1.4]
 [ 5.8  2.7  nan  nan]
 [ 6.8  2.8  4.8  1.4]
 [ 5.4  3.9  1.7  nan]
 [ 7.6  3.   6.6  2.1]
 [ 7.7  2.8  6.7  2. ]
 [ 5.   3.3  nan  0.2]
 [ 5.9  3.   4.2  1.5]
 [ 6.1  2.8  4.   1.3]
 [ 5.   3.6  1.4  0.2]
 [ 7.4  2.8  6.1  1.9]
 [ 6.3  2.5  5.   1.9]
 [ 6.7  3.3  5.7  2.5]]
```

```python
from sklearn.impute import SimpleImputer
imp = SimpleImputer(strategy="median").fit(X_train)
X_median_imp = imp.transform(X_train)
```




```
array([[ 6.   ,  2.9  ,  4.5  ,  1.5  ],
       [ 5.9  ,  3.   ,  5.1  ,  1.8  ],
       [ 4.4  ,  3.   ,  1.3  ,  0.2  ],
       [ 5.1  ,  3.3  ,  4.116,  1.462],
       [ 5.   ,  3.5  ,  1.6  ,  0.6  ],
       [ 5.4  ,  3.4  ,  4.116,  1.462],
       [ 5.7  ,  3.8  ,  4.116,  0.3  ],
       [ 5.6  ,  2.5  ,  3.9  ,  1.462],
       [ 7.7  ,  2.6  ,  6.9  ,  2.3  ],
       [ 5.8  ,  2.7  ,  5.1  ,  1.9  ],
       [ 6.7  ,  3.1  ,  5.6  ,  2.4  ],
       [ 4.8  ,  3.4  ,  1.9  ,  1.462],
       [ 7.2  ,  3.2  ,  6.   ,  1.8  ],
       [ 4.4  ,  2.9  ,  4.116,  1.462],
       [ 6.9  ,  3.2  ,  5.7  ,  2.3  ],
       [ 5.5  ,  4.2  ,  1.4  ,  1.462],
       [ 6.3  ,  2.3  ,  4.4  ,  1.3  ],
       [ 7.   ,  3.2  ,  4.7  ,  1.4  ],
       [ 5.8  ,  2.7  ,  4.116,  1.462],
       [ 6.8  ,  2.8  ,  4.8  ,  1.4  ],
       [ 5.4  ,  3.9  ,  1.7  ,  1.462],
       [ 7.6  ,  3.   ,  6.6  ,  2.1  ],
       [ 7.7  ,  2.8  ,  6.7  ,  2.   ],
       [ 5.   ,  3.3  ,  4.116,  0.2  ],
       [ 5.9  ,  3.   ,  4.2  ,  1.5  ],
       [ 6.1  ,  2.8  ,  4.   ,  1.3  ],
       [ 5.   ,  3.6  ,  1.4  ,  0.2  ],
       [ 7.4  ,  2.8  ,  6.1  ,  1.9  ],
       [ 6.3  ,  2.5  ,  5.   ,  1.9  ],
       [ 6.7  ,  3.3  ,  5.7  ,  2.5  ]])
```

Example: Median Imputation



Missing Data - Mean/Median Imputation

Advantages:

- Easy to implement

Disadvantages:

- distorts the distribution of the dataset
- distorts the variance of the dataset by reducing the variance
- distorts the co-variance

Improvement:

- look at mean/median per class for classification data

Missing Data - Random Imputation

Method involves substituting the missing values with values extracted from the original variable

The two randomly picked Ages are 25 & 30

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1

ID	City	Age	Age-Imputed	Married ?
1	Lisbon	25	25	0
2	Berlin	25	25	1
3	Lisbon	30	30	1
4	Lisbon	30	30	1
5	Berlin	18	18	0
6	Lisbon	NaN	25	0
7	Berlin	30	30	1
8	Berlin	NaN	30	0
9	Berlin	25	25	1
10	Madrid	25	25	1

Missing Data - Random Imputation

Advantages:

- does not distort the variance
- does not distort the distribution

Disadvantages:

- not widely used
- element of randomness

Missing Data - Arbitrary Value

Method uses an arbitrary value to replace the missing values

Arbitrary Value = -1

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	-1	0
7	Berlin	30	1
8	Berlin	-1	0
9	Berlin	25	1
10	Madrid	25	1

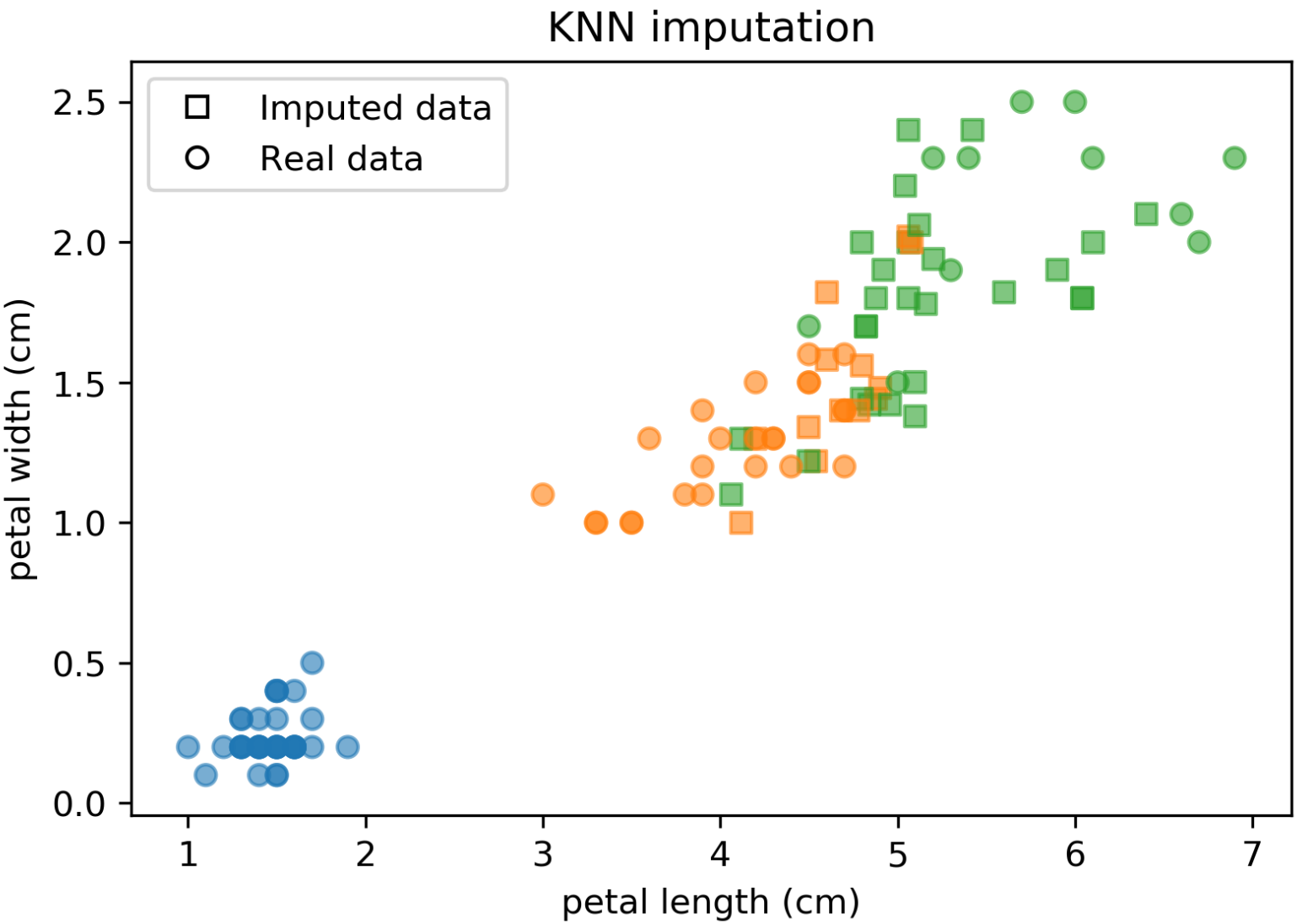
- A transformer convert the data set to a binary

Missing Data - KNN Imputation

- Find k nearest neighbors that have non-missing values.
- Fill in all missing values using the average of the neighbors.

- Challenges must select k and distance measure to determine neighbors

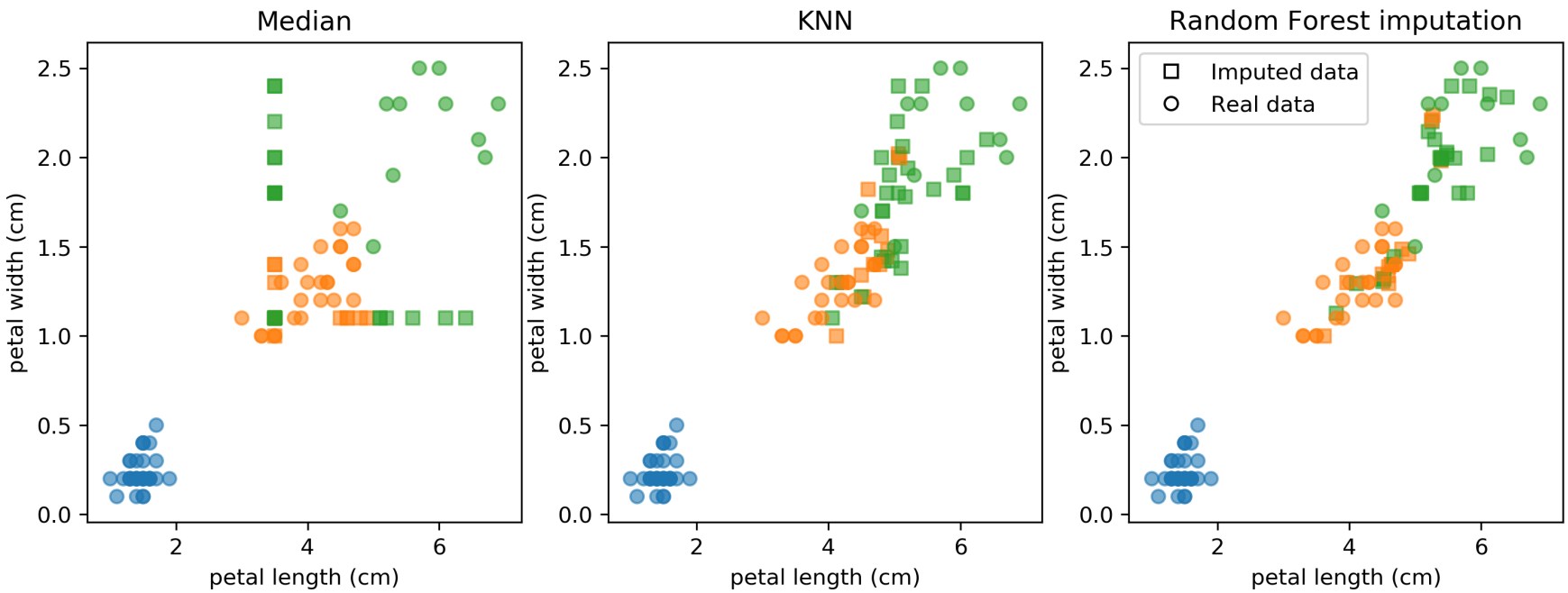
Missing Data - KNN Imputation



Missing Data - Model Imputation

- Train regression model for missing values
- Iterative Imputation
  - models each feature with missing values as a function of other features, in an iterated round-robin fashion
  - at each step:
    - a feature column is designated as output `y` and the other feature columns are treated as inputs `X`
    - a regressor is fit on `(X, y)` for known `y` and to predict the missing values of `y`
  - this is done for each feature in an iterative fashion, and then is repeated for max_iter imputation rounds

Handling Missing Data - Comparison



Data Cleaning as a Process

Take time at this step

- use metadata, statistics, domain knowledge (e.g., domain, range, dependency, distribution)
- check field overloading
- use commercial tools
  - Data scrubbing: use simple domain knowledge to detect errors and make corrections
  - Data auditing: manually and programmatically analyze the data to detect relationships and violations

Data Preparation - Integration

Data Integration

Combine data from multiple sources into a coherent collection

- Ex. Entity identification problem
  identify real world entities from multiple sources,
  e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources are different
  - why? different representations, different scales

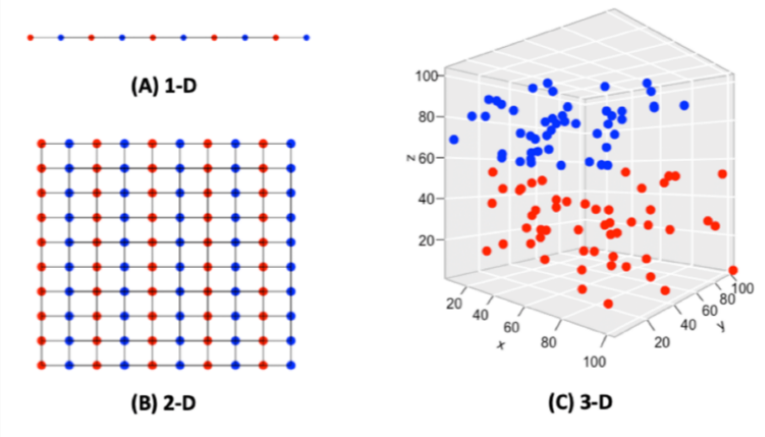
Redundancy in Data Integration

- Redundant data often occur when integration of multiple databases
  - Object identification: the same attribute or object may have different names in different databases
  - Derivable data: one attribute may be a "derived" attribute in other table
- Careful integration of data from multiple sources may help reduce or avoid redundancies and inconsistencies
- Redundant attributes may be detected by correlation or covariance analysis
- ETL tools support deduplication

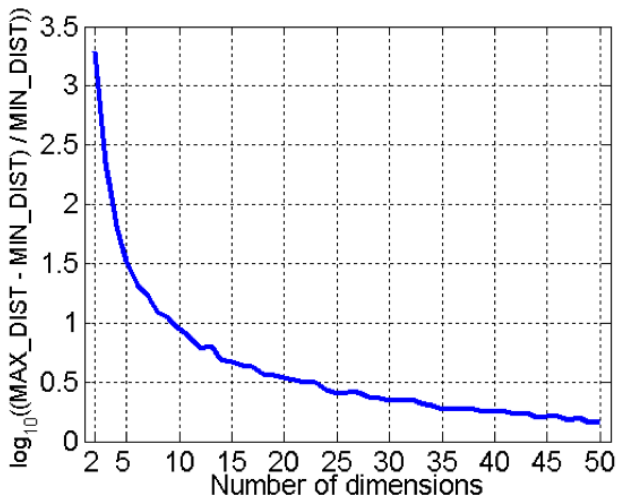
Data Preparation - Reduction

Curse of Dimensionality

- When dimensionality increases, the size of the data space grows exponentially.
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful
  - Density $\rightarrow 0$
  - All points tend to have same Euclidean distance to each other



Experiment: Randomly generate 500 points. Compute difference between max and min distance between any pair of points



Data Reduction

- Obtain a reduced representation of the data set that is smaller in volume that produces the same (or almost the same) analytical results
- Strategies
  - Dimensionality reduction: wavelet transforms, principal component analysis (PCA), feature subset selection, feature creation
  - Numerosity reduction: regression and log-linear models, histograms, clustering, sampling, data cube aggregation
  - Data compression: transform data to obtain a reduced or "compressed form"

Data Preprocessing Tasks

- Data Cleaning
  - check data quality
  - missing data
- Data Integration
  - integration of multiple data sources
- Data Reduction
  - obtain a reduced representation of data that produces same/similar results
- Data Transformation and Discretization
  - normalize, discretize, etc.; prepare data for additional analyses

Data Transformation

Data Transformations

- Aggregation
- Sampling
- Functional
- Discretization
- Binarization
- Scaling
- ...

Aggregation

- Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - Data reduction
    Reduce the number of attributes or objects
  - Change of scale
    Cities aggregated into regions, states, countries, etc.
  - More "stable" data
    Aggregated data tends to have less variability

Sampling

- Statisticians *often* sample because obtaining the entire set of data of interest is too expensive or time consuming.

The key principle for effective sampling is:

- Using a sample will work almost as well as using the entire data sets, if the sample is representative.
- A sample is representative if it has approximately the same property (of interest) as the original set of data.

Types of Sampling

Replacement

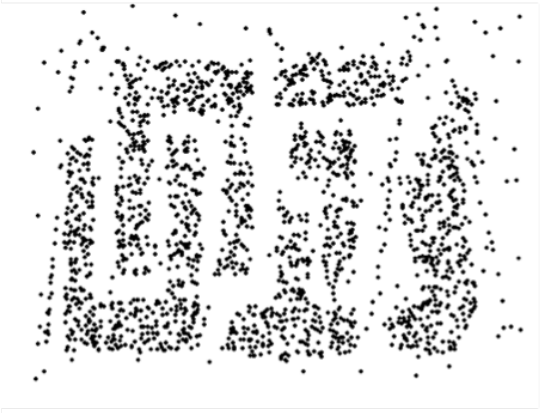
- Sampling without replacement
  - As each item is selected, it is removed from the population
- Sampling with replacement
  - The same object can be selected multiple times

Selection

- Simple Random Sampling
  - There is an equal probability of selecting any particular item
- Stratified Sampling
  - Split data into several partitions, then draw random samples from each partition

Sample Size Impact

8000 points

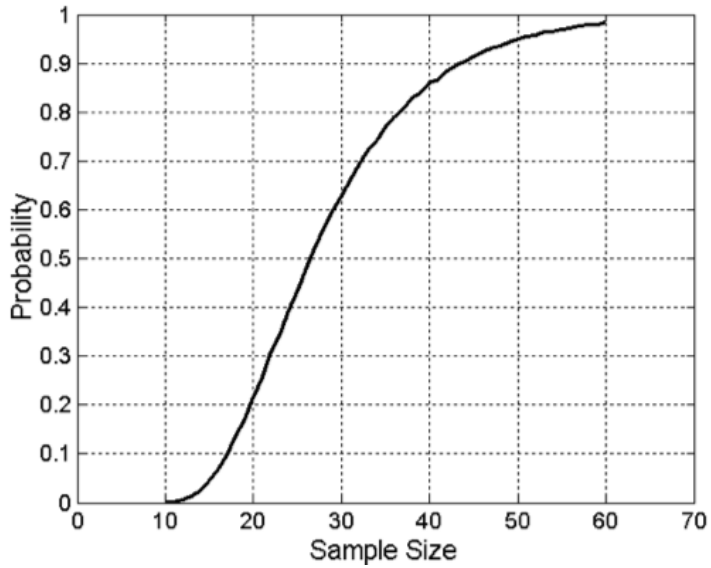


2000 Points

500 Points

Sample Size Impact

- What sample size is necessary to ensure at least one sample from each of 10 groups?



- Sample size determination
  - Stats: confidence interval or desired statistical power
  - DM/ML: cross-validated performance, more is better*

Functional Transformation

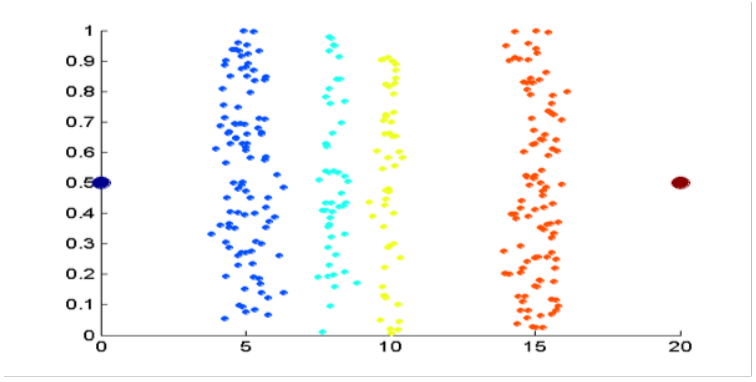
An attribute uses a function to map the entire set of values to a replacement set of values:

Example Functions:

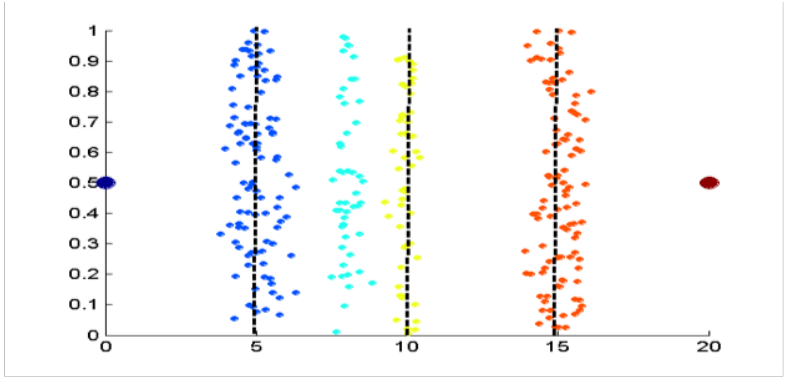
- $\log(x)$
- x^k
- $|x|$
- e^x
- ...

Discretization

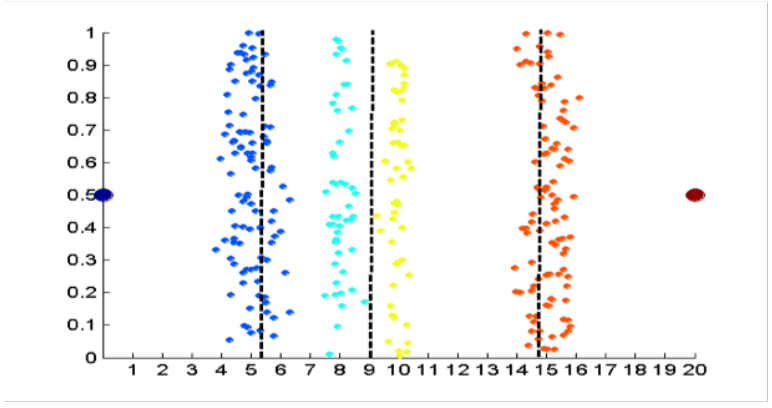
- Discretization is the process of converting a continuous attribute into an ordinal attribute
- Example of Unsupervised Discretization



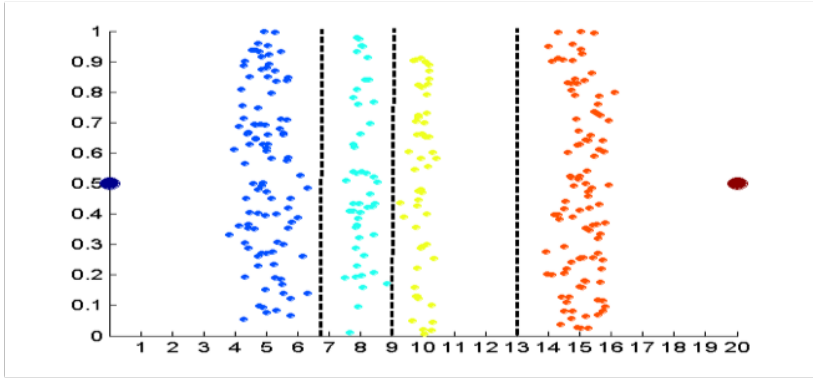
Data



Equal interval width



Equal frequency



K-means

Binarization

- Binarization maps a continuous or categorical attribute into one of more binary variables.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Categorical Data

Language Differences

- R generally has support for data with categorical variables implicitly (both nominal and ordinal).
  - Examples:
    - Color: $\{green, blue, brown, red\}$
    - Area: $\{Manhattan, Brooklyn, Queens, Bronx, StatenIsland\}$
    - Vegan: $\{yes, no\}$
- Python, specifically `scikit-learn` generally assumes/requires all input to be continuous numbers.

Ordinal Encoding

	boro	salary	vegan
0	Manhattan	103	No
1	Queens	89	No
2	Manhattan	142	No
3	Brooklyn	54	Yes
4	Brooklyn	63	Yes
5	Bronx	219	No

Ordinal Encoding

```
df['boro_ordinal'] = df.boro.astype("category").cat.codes
```

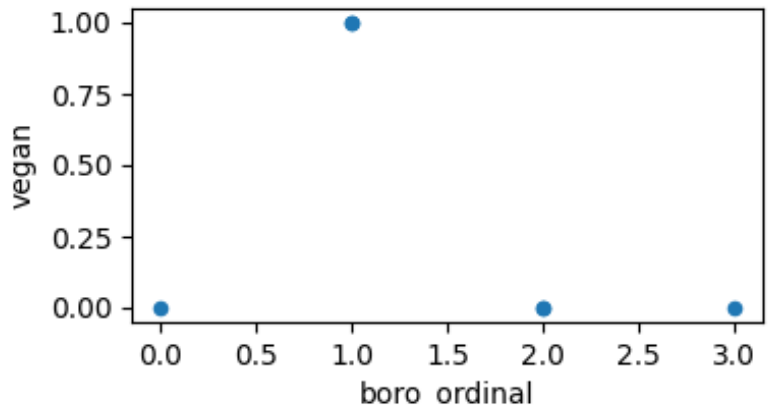
	boro	salary	vegan
0	Manhattan	103	No
1	Queens	89	No
2	Manhattan	142	No
3	Brooklyn	54	Yes
4	Brooklyn	63	Yes
5	Bronx	219	No

	boro	boro_ordinal	vegan
0	Manhattan	2	No
1	Queens	3	No
2	Manhattan	2	No
3	Brooklyn	1	Yes
4	Brooklyn	1	Yes
5	Bronx	0	No

Ordinal Encoding

```
df['boro_ordinal'] = df.boro.astype("category").cat.codes
```

	boro	boro_ordinal	vegan
0	Manhattan	2	No
1	Queens	3	No
2	Manhattan	2	No
3	Brooklyn	1	Yes
4	Brooklyn	1	Yes
5	Bronx	0	No



What's Wrong? Defining a linear relation, and order between the categories

One-Hot Encoding

```
pd.get_dummies(df)
```

	boro	salary	vegan
0	Manhattan	103	No
1	Queens	89	No
2	Manhattan	142	No
3	Brooklyn	54	Yes
4	Brooklyn	63	Yes
5	Bronx	219	No

	salary	boro_Bronx	boro_Brooklyn	boro_Manhattan	boro_Queens	vegan_No	vegan_Yes
0	103	0	0	1	0	1	0
1	89	0	0	0	1	1	0
2	142	0	0	1	0	1	0
3	54	0	1	0	0	0	1
4	63	0	1	0	0	0	1
5	219	1	0	0	0	1	0

What's Wrong?

Alternative: `OneHotEncoder` from `sklearn`

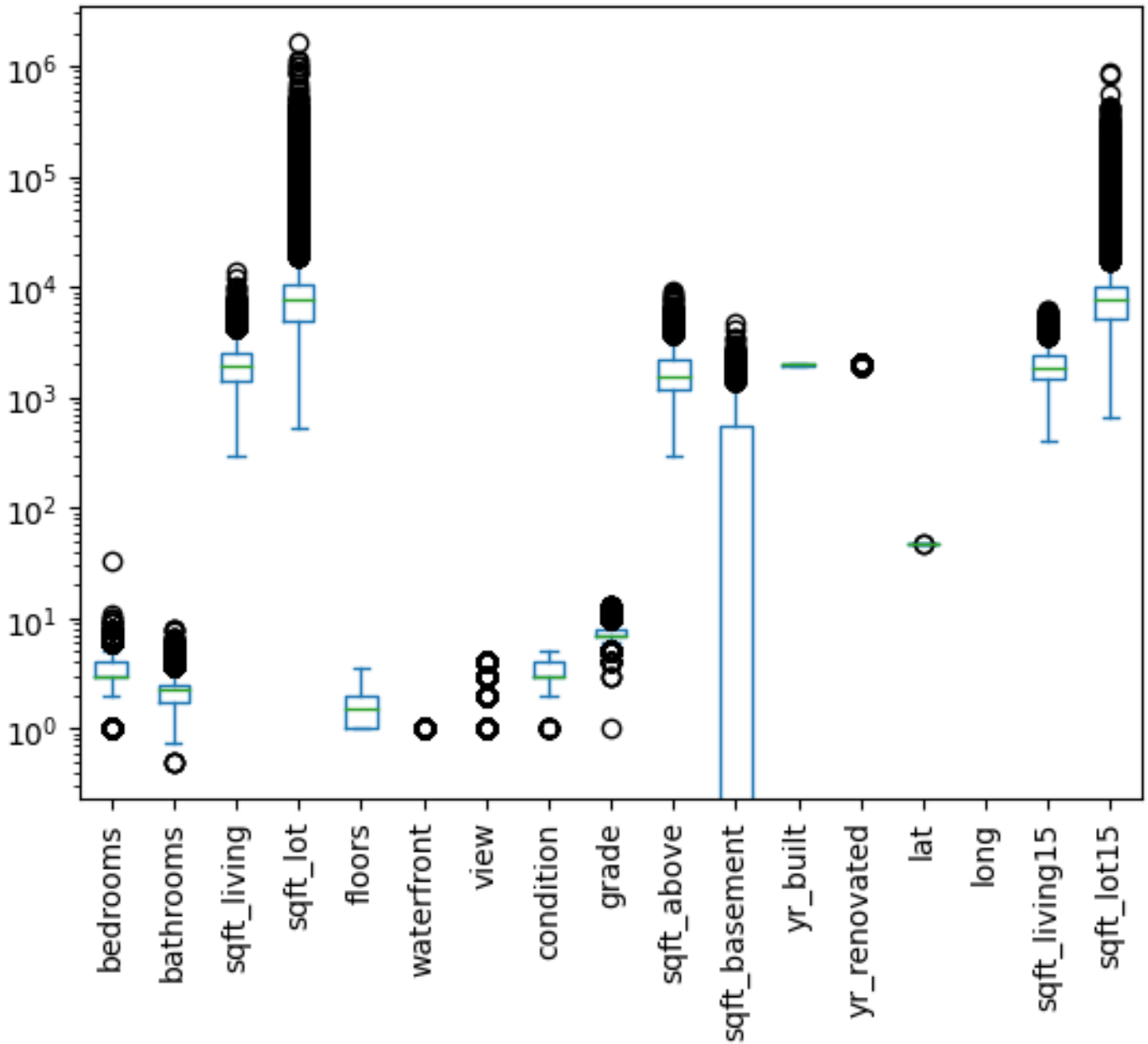
Dummy variables and colinearity

- What's Wrong?
  - One-hot is redundant (last column is 1 - sum of others)
  - Can introduce co-linearity
- Solution? Can drop a column
- Issue? Choice of which column to drop matters for penalized models
- In Classification, may keep all to help with interpretability of the model.

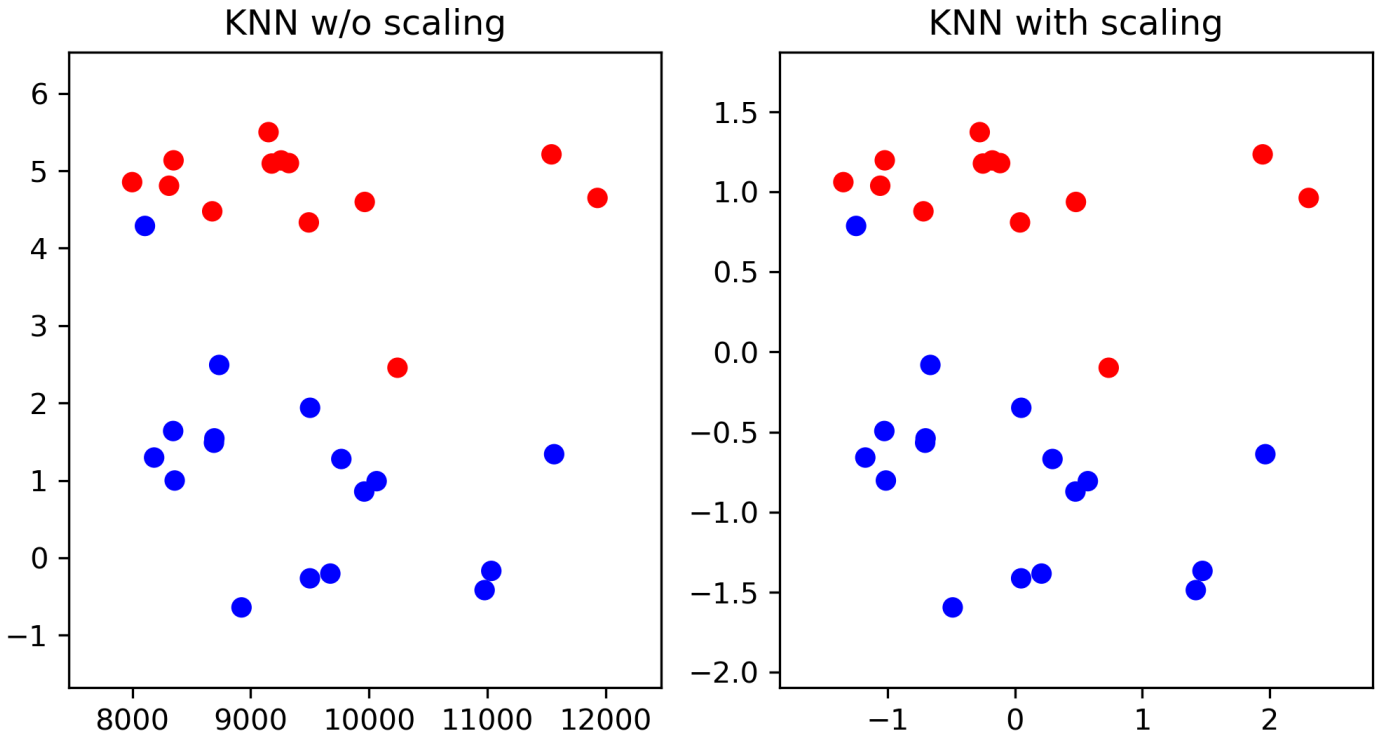
Target Encoding (Impact Encoding)

- What if you have a high cardinality categorical features
  - Example: US states, US zip codes
- Don't create many one-hot encoded variables. Instead create one "response encoded" variable
- Different encoding depending on problem type:
  - Regression - "average price in zip code"
  - Binary classification - "building in this zip code have a likelihood p for class 1"
  - Multi-class classification - "One feature per class: probability distribution"

Scaling

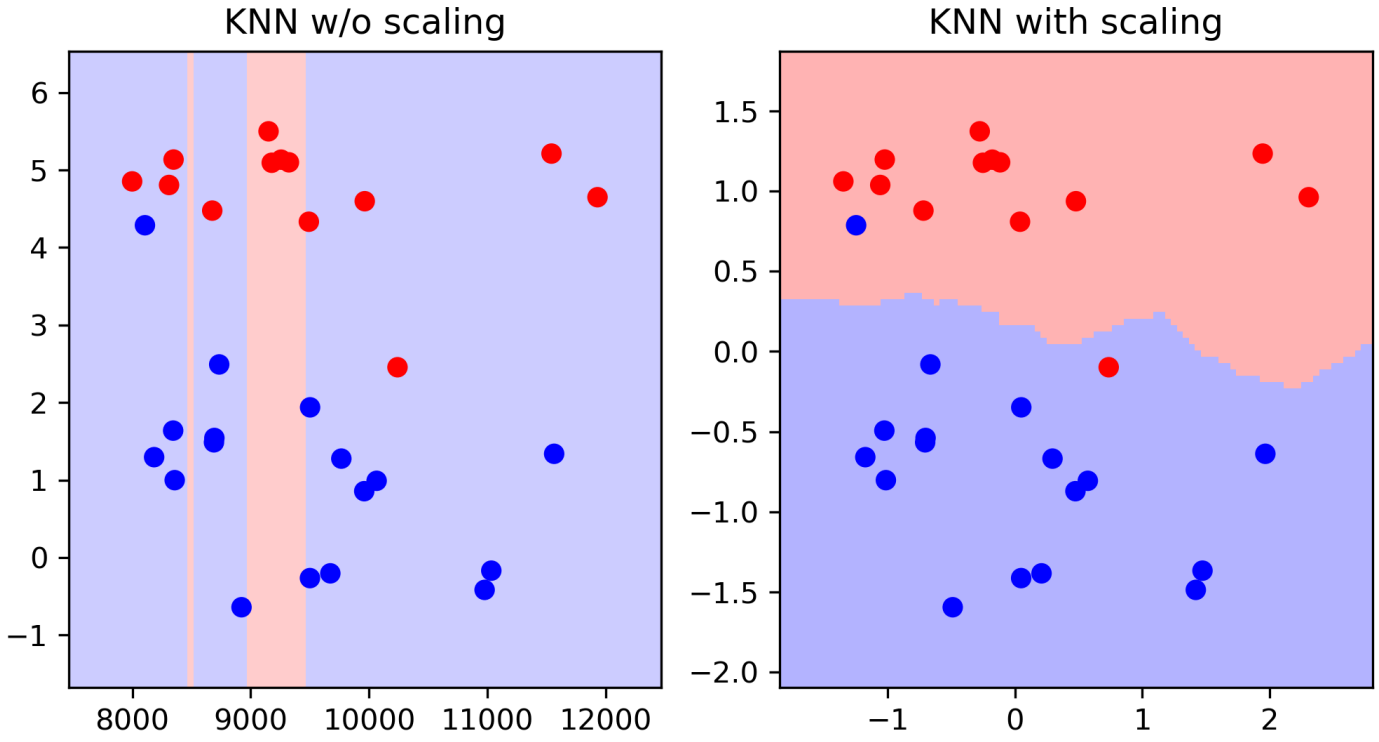

10^6
10^5
10^4
10^3
10^2
10^1
10^0
bedrooms
bathrooms
sqft_living
sqft_lot
floors
waterfront
view
condition
grade
sqft_above
sqft_basement
yr_built
yr_renovated
lat
long
sqft_living15
sqft_lot15

Scaling



StandardScaler used on right plot

Scaling



What's Wrong? KNN classifier shown with background color

Scaling Methods: Min-Max (linear)

- Min-Max scaling subtracts the minimum and divides by the range to scale to be between 0 and 1 for each variable
- Min-Max scaling for given attribute A:
  - Map values to be between, parameters: $[min_A^*, max_A^*]$

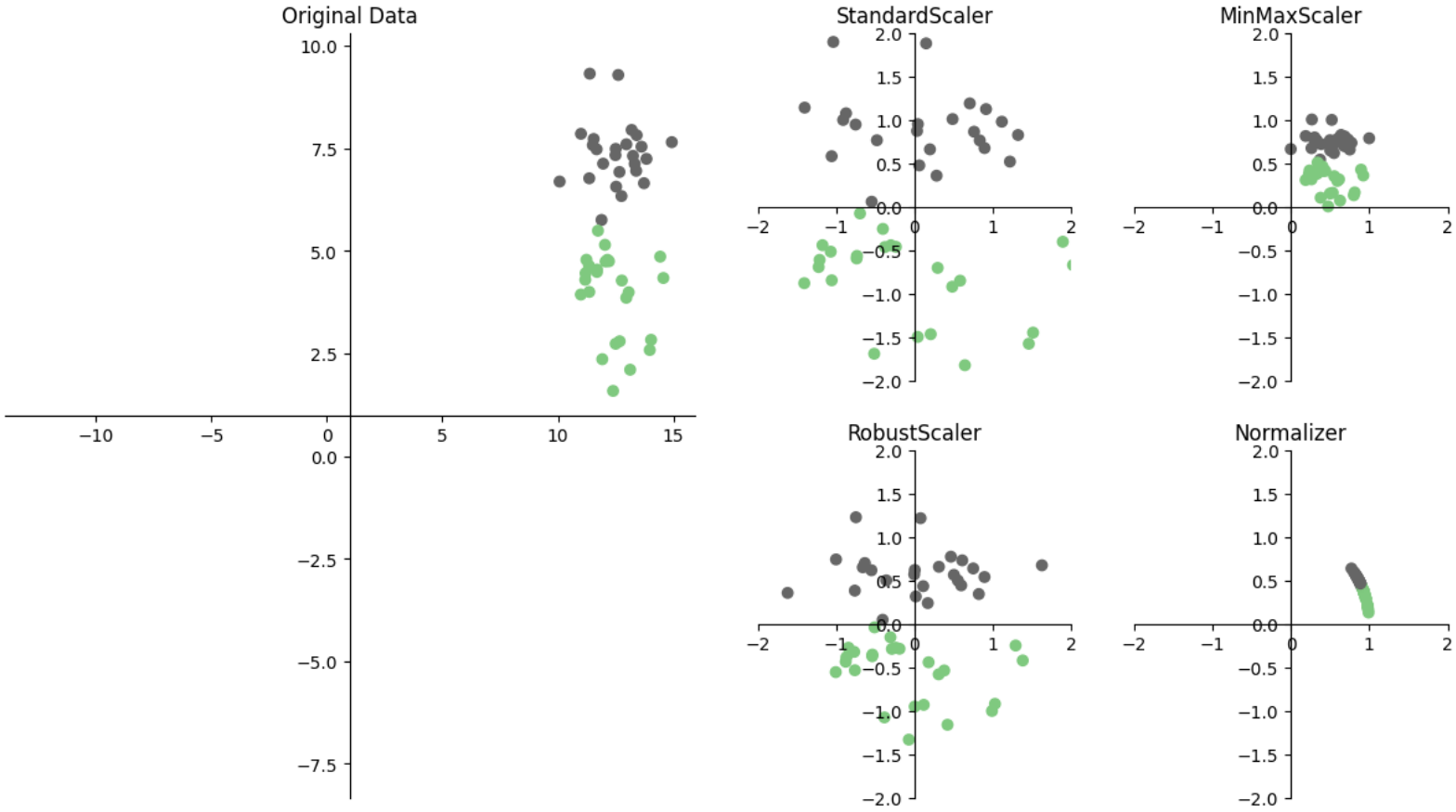
$$\hat{v} = \frac{v - min_A}{max_A - min_A}(max_A^* - min_A^*) + min_A^*$$

Scaling: Standard (z-score, Gaussian)

- Standard subtracts the mean and divides by the standard deviation for each variable.
- Result: All variables have a zero mean and standard deviation of 1.
- Method for given attribute A
  - Parameters: μ_A - mean, σ_A - std. dev.

$$\hat{v} = \frac{v - \mu_A}{\sigma_A} = \frac{v - \bar{x}_A}{\hat{\sigma}_A}$$

Scaling Methods



Scaling with Sparse Data

- Data with many zeros where only non-zero entries are stored
- Any scaling where you subtract of the mean or another value, will make the data "dense" and blow up the RAM
- Only scale, don't center (`MaxAbsScaler`)

Overall Scaling