Data Preprocessing

Adapted from Discovering Knowledge in Data: An Introduction to Data Mining, Second Edition, by Daniel Larose and Chantal Larose, John Wiley and Sons, Inc.,

Where & Why Do We Preprocess Data?

CRISP-DM Review

Business Research Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment

- Focusing on Data Understanding and Data Preparation Process CRISP-DM process
- For data mining purposes, database values must undergo <u>data cleaning</u> and <u>data transformation</u>

- Raw data often unprocessed, incomplete, noisy
- May contain:
  - Obsolete/redundant fields
  - Missing values
  - Outliers
  - Data in form not suitable for data mining
  - Values not consistent with common sense

Data Cleaning – Example

Can You Find Any Problems in This Tiny Data Set?

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	—40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Data-set of USA Customer Transactions

CustomerID field is assumed to be fine; But Zip Code, Gender?

Handling Missing Data

- Examine *cars* dataset containing records for 261 automobiles manufactured in 1970s and 1980s
- Suppose that some fields are missing for certain records, like in figure below:

	mpg	cubicinches	hp	brand
1	14.000	350	165	US
2	31.900		71	Europe
3	17.000	302	140	US
4	15.000	400	150	
5	37.700	89	62	Japan

- Delete Records Containing Missing Values?
  - Dangerous, as pattern of missing values may be systematic
  - Valuable information in other fields lost
- Three alternative methods available – Not entirely satisfactory
  - Method 1 – Replace with a Constant
  - Method 2 – Replace with Mode or Mean
  - Method 3 – Replace with Random values

Data Imputation methods

- Better approach
  - i.e. Imputed value based on other characteristics of the record
  - We ask "what would be the most likely value for the missing data given all other attributes for this record?"
- A General Approach
  1. Impute the values of the variable with the fewest missing values. Use only variables with no missing values as predictors
  2. Impute the values of the variable with the next fewest missing values
  3. Repeat Step 2 until all missing values imputed
  - **Continuous Variables** –e.g. some records in a Cereals dataset is missing a **potassium** value
  - Build a Multiple Regression Model with the dataset e.g.

$$= 6.004 - 1.7741(\text{Fat}) + 0.06557(\text{Calories}) + 0.9297(\text{Protein}) + 0.013364(\text{Sodium}) - 0.7331(\text{Fiber}) + 4.406(\text{Nabisco}) + 2.7(\text{Ralston})$$

  - **Categorical Variables** – Use a classification techniques e.g. CART
    - Response Variable (Target Variable) is the missing value.
    - If marital status is a missing value for a number of records build a decision tree with the dataset to predict the marital status

CART model for imputing *maritalstatus*

- loans in [0 1] [Mode: married] (103)
  - income_Z <= 0.812 [Mode: single] (73)
    - age_Z <= 0.774 [Mode: single] (55)
      - age_Z <= -0.266 [Mode: single] (33)
        - income_Z <= -0.947 [Mode: married] ⇨ **married** (3; 1.0)
        - income_Z > -0.947 [Mode: single] ⇨ **single** (30; 0.8)
      - age_Z > -0.266 [Mode: single] ⇨ **single** (22; 0.864)
    - age_Z > 0.774 [Mode: married] ⇨ **married** (18; 0.722)
  - income_Z > 0.812 [Mode: married] ⇨ **married** (30; 0.967)
- loans in [2 3] [Mode: other] (65)
  - loans in [0 1 2] [Mode: married] (49)
    - age_Z <= -0.682 [Mode: married] ⇨ **married** (13; 0.923)
    - age_Z > -0.682 [Mode: other] (36)
      - income_Z <= 0.213 [Mode: other] ⇨ **other** (30; 0.667)
      - income_Z > 0.213 [Mode: married] ⇨ **married** (6; 1.0)
  - loans in [3] [Mode: other] ⇨ **other** (16; 1.0)

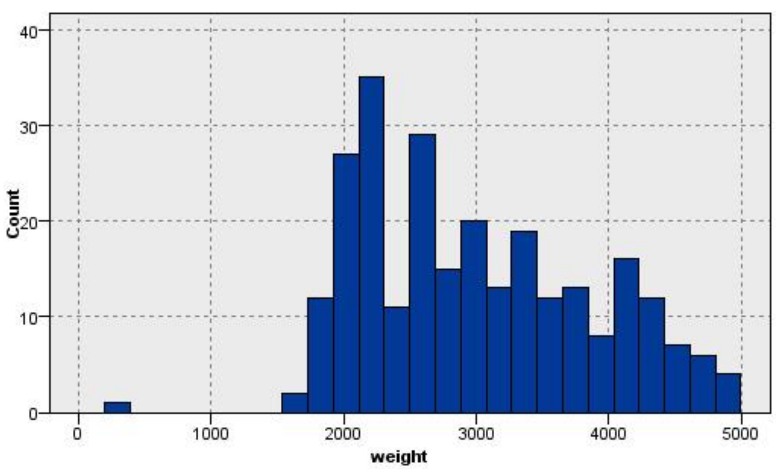
Here we have customers with missing marital statusCustomer Record has loans = 1, mortgage = y, age_Z = 1.45, income_Z = 1.498

?? Impute the marital status value for this customer using the Decision Tree above

Note: other classification techniques are also used

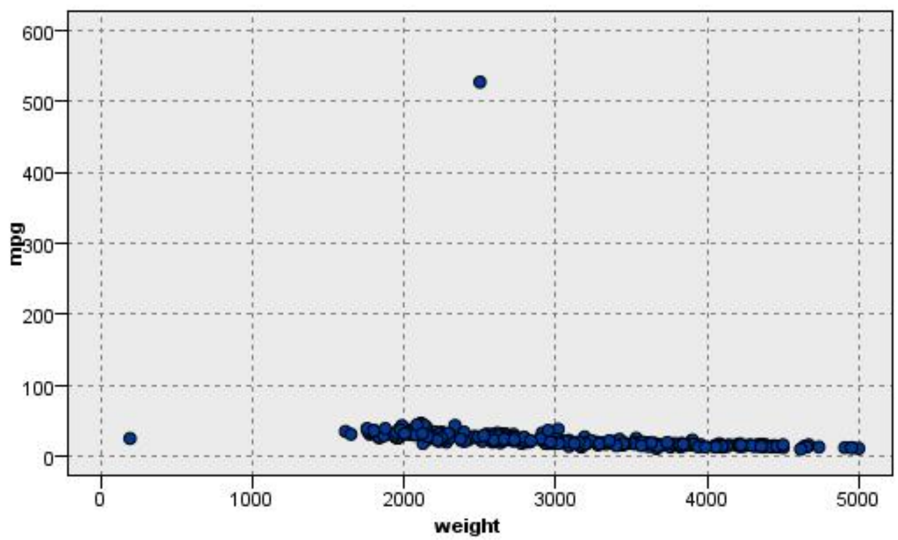
Graphical Methods for Identifying Outliers *(cont'd)*

- Outliers are extreme values that go against the trend of the remaining data
- Method #1 - Histogram
  - A histogram examines values of numeric fields
  - Example: Histogram shows vehicle weights for *cars* data set* (cars2.txt)
    - The extreme left-tail contains one outlier weighing several hundred pounds (192.5)
    - Should we doubt validity of this value? This is too light for a car.
    - Possibility: Original value was 1925 pounds. Requires further investigation.



Graphical Methods for Identifying Outliers *(cont'd)*

- Method #2 – Two-dimensional Scatter Plot
  - Two-dimensional scatter plots help determine outliers in more than one variable
  - Example: Scatter plot of *mpg* against *weightlbs* shows two possible outliers
    - Most data points cluster together along x-axis
    - However, one car weighs 192.5 pounds and other gets over 500 miles per gallon?
    - Important: A record may be outlier in a particular dimension, but not in the other



Measures of Center

Measures of center

Estimate where the center of a particular variable lies

- Most common *measures of center*
  - Mean, Median and Mode
    - They are a special case of *measures of location*, which indicate where a numeric variables lies (examples: percentiles and quantiles)

Measures of Spread

Measures of Spread - Recap

- Measures of location not enough to summarize a variable
- Example: Table with P/E ratios for two portfolios (below)
  - Portfolio A – Spread with one very low and one very high value
  - Portfolio B – Tightly clustered around the center
  - P/E ratios for each portfolio is distinctly different, yet **they both** have P/E ratios Clearly, measures of center do not provide a complete picture
- **Measures of spread** or measure of variability complete the picture by describing how spread the data values of each portfolio are

Stock Portfolio A	Stock Portfolio B
1	7
11	8
11	11
11	11
16	13

Measures of Spread

Measures of Spread - Refresh

- Typical measures of variability include
  - **Range** (maximum – minimum)
  - **Variance –** how far asset of numbers are spread out
  - **Standard Deviation** – Sensitive to the presence of outliers (because of the squaring involved – see below)
  - **Mean Absolute Deviation** – Preferred in situations involving extreme values
  - **Interquartile Range**
- Sample Standard Deviation is defined by

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

  - Interpreted as "typical" distance between a field value and the mean
  - Most field values lie within two standard deviations of the mean

Data Transformation

- Variables tend to have ranges different from each other
- In a dataset, two fields may have ranges:
  - **Annual Income:** **[€15,000, €300,000]**
  - **Employee Age:** **[16, 70]**
- Some data mining algorithms adversely affected by differences in variable ranges
  - Variables with greater ranges tend to have larger influence on data model's results
  - Therefore, <u>numeric</u> field values should be <u>normalized</u>
- Standardizing scales the effect each variable has on results
- kNearest Neighbour, Neural Networks and other algorithms that make use of distance measures benefit from normalization

Data Transformation: Min-Max Normalization

Find Min-Max normalization for cars weighing 1613, 3384 and 4997 pounds, respectively

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Where:

min(X) = 1613

max(X) = 4997

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	?	X* = ?	Represents the minimum value in this variable, and has min-max normalization ?
Mid-range vehicle	X = 3384	?	X* = ?	Weight exactly half-weight between the ligthest and the heaviest vehicle, and has min-max normalization of ?
Heaviest vehicle	X = 4997	?	X* = ?	Heaviest vehicle of the dataset has min-max normalization of ?

Min-Max normalization will always have a value between 0 and 1.

Data Transformation: Min-Max Normalization

Find Min-Max normalization for cars weighing 1613, 3384 and 4997 pounds, respectively

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Where:

min(X) = 1613

max(X) = 4997

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	$X^* = \frac{1613 - 1613}{4997 - 1613}$	X* = 0	Represents the minimum value in this variable, and has min-max normalization of zero.
Mid-range vehicle	X = 3384	$X = \frac{3884 - 1613}{4997 - 1613}$	X* = 0.5	Weight exactly half-weight between the ligthest and the heaviest vehicle, and has min-max normalization of 0.5.
Heaviest vehicle	X = 4997	$X = \frac{4997 - 1613}{4997 - 1613}$	X* = 1	Heaviest vehicle of the dataset has min-max normalization of one.

Min-Max normalization will always have a value between 0 and 1.

Data Transformation: Z-score Standardization

- Widely used in statistical analysis
- Takes difference between field value and field value mean
- Scales this difference by field's standard deviation

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

Find Z-score standardization for cars weighing 1613, 3384 and 4997 lbs, respectively

Summary Statistics for weightlbs

weightlbs
Statistics

Mean	3005.490
Min	1613
Max	4997
Range	3384
Standard Deviation	852.646

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	?	X* ≈ ?	Data values below the mean will have negative Z-score standardization.
Mid-range vehicle	X = 3384	?	X* ≈ ?	Values falling exactly on the mean will have ? Z-score
Heaviest vehicle	X = 4997	?	X* ≈ ?	Data values above the mean will have a positive Z-score standardization

Data Transformation: Z-score Standardization

Find Z-score standardization for cars weighing 1613, 3384 and 4997 pounds, respectively

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

Where:

mean(X) = 3005.49

SD(X) = 852.65

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	$X^* = \frac{1613 - 3005.49}{852.646}$	X* ≈ -1.63	Data values below the mean will have negative Z-score standardization.
Mid-range vehicle	X = 3384	$X = \frac{3384 - 3005.49}{852.646}$	X* ≈ 0.44	Values falling exactly on the mean will have zero (0) Z-score
Heaviest vehicle	X = 4997	$X = \frac{4997 - 3005.49}{852.646}$	X* ≈ 2.34	Data values about the mean will have a negative Z-score standardization

It is also possible to find the associated data value for a give Z-score

Data Transformation: Decimal Scaling

- Ensures that normalized values lies between -1 and 1
- Defined as:

$$X^* = \frac{X}{10^d}$$

where d represents the number of digits in the data value with the largest absolute value.

- For the weight data, the largest absolute value is $|4997| = 4997$, with $d=4$ digits
- Decimal scaling for the minimum and maximum weights are:

$$Min: X^*_{decimal} = \frac{1613}{10^4} = 0.1613$$

$$Max: X^*_{decimal} = \frac{4997}{10^4} = 0.4997$$

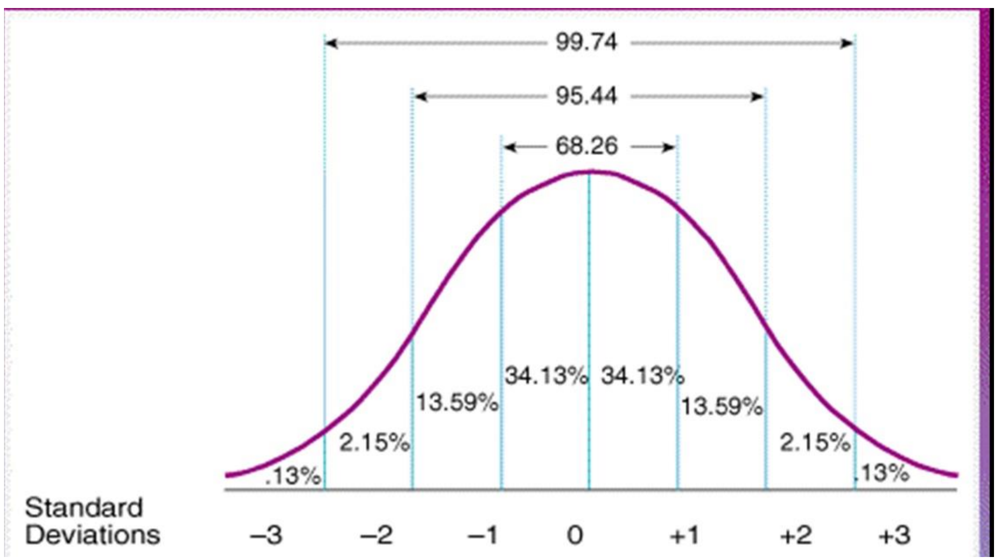
weightlbs

Statistics

Mean	3005.490
Min	1613
Max	4997
Range	3384
Standard Deviation	852.646

Transformations to achieve normality

- Some data mining algorithms and statistics methods require *normally distributed* variables
- Normal distribution (Gaussian Distribution)
  - Continuous probability distribution known as the 'bell curve' (symmetric)
  - Centered and mean μ (myu) and spread given by σ (sigma)



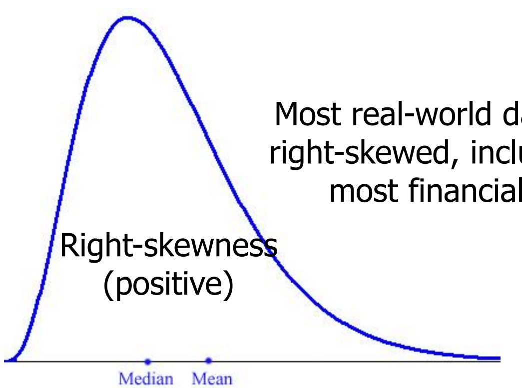
Standard normal Z-distribution
with μ=0 and σ=1

Transformations to achieve normality

- Some data mining algorithms and statistics methods require *normally distributed* variables
- Statistics for measuring **the skewness of a distribution**:

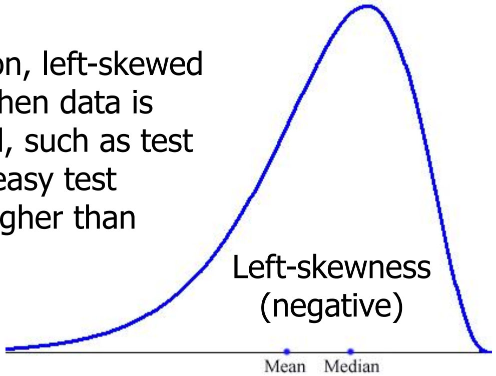
$$Skewness = \frac{3(mean\ - median)}{standard\ deviation}$$

- Right-skewness data – Is positive, as mean is greater than the median
- Left skewness data – Mean is smaller than the median, generating negative values
- Perfectly symmetric data – mean, median and mode are equal, so skewness is zero



Most real-world data is right-skewed, including most financial data

Not as common, left-skewed data occurs when data is right-censored, such as test scores on an easy test (cannot get higher than 100).



Transformations to achieve normality

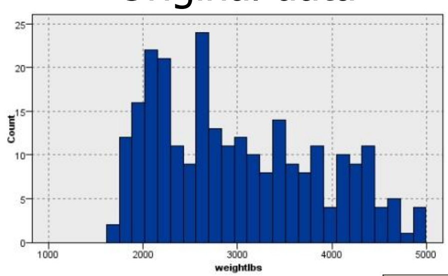
- To eliminate skewness, we must apply a transformation to the data
  - This makes the data symmetric and makes it “more normally distributed”
- Common transformations are:

Natural Log	Square Root	Inverse Square Root
$\ln(y)$	$\sqrt{y}$	$\frac{1}{\sqrt{y}}$

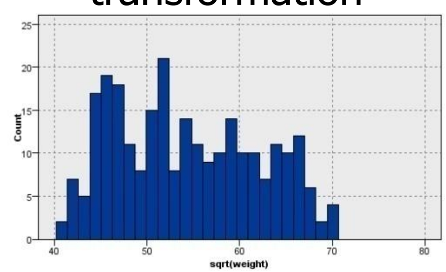
Square root transformation somewhat reduces skewness, while ln reduces it further while inverse square root reduces further again (y = weightlbs)

Important: There is nothing special about the inverse square root transformation. It just worked with the skewness in the weight data

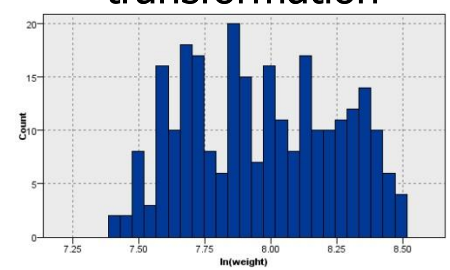
Original data



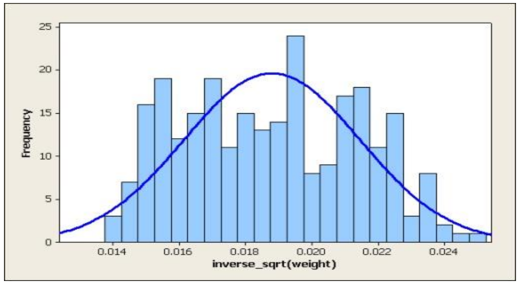
Square root transformation



Natural log transformation



Inverse Square Root transformation

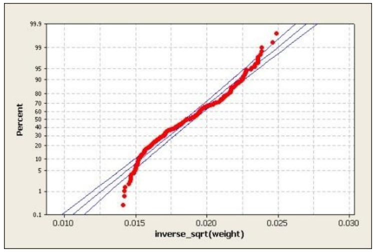


Notice that while we have achieved symmetry, we have not reached normality (the distribution does not match the normal curve)

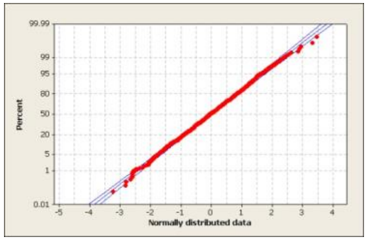
Transformations to achieve normality (*cont'd*)

- After achieving symmetry, we must also check for normality
- The Normal Probability Plot
  - Plots the quantiles for a particular distribution *against* the quantiles of the standard normal distribution
  - Similar to percentile, p^{th} *quantile* of a distribution is value x_p, such that $p\%$ of the distribution values are less than or equal to x_p
  - If the bulk of the points fall on a straight line, the distribution is normal; systematic deviations indicate nonnormality
- As expected, the normal probability plot for the inverse_sqrt(weigth) indicates nonnormality

Normal probability plots



Plot for inverse_sqrt(weight) has systematic deviations that indicate nonnormality



Plot for normally distributed data

Numerical Methods for Identifying Outliers

- **Using Z-score Standardization to Identify Outliers**
  - Outliers are Z-score Standardization values either <u>less than -3</u>, or <u>greater than 3</u>
  - Values much beyond range [-3, 3] require further investigation to <u>determine their validity</u>
    - Should not automatically omit outliers from analysis
  - However, Mean and Standard Deviation are both sensitive to the presence of outliers
    - Sample Mean and sample standard deviation are both part of the formula for z-score standardization
    - If an outlier is added or deleted from the dataset, mean and standard deviation is affected
  - When selecting a method for evaluating outliers, should not use measures which are themselves sensitive to outliers

Numerical Methods for Identifying Outliers *(cont'd)*

- Using Interquartile Range (IQR) to Identify Outliers
  - Robust statistical method and less sensitive to presence of outliers
  - Data divided into four quartiles, each containing 25% of data
    - First quartile (Q1) 25th percentile
    - Second quartile (Q2) 50th percentile (median)
    - Third quartile (Q3) 75th percentile
    - Fourth quartile (Q4) 100th percentile
  - IQR is measure of variability in data

Numerical Methods for Identifying Outliers *(cont'd)*

- IQR = Q3 – Q1 and represents spread of middle 50% of the data
- Data value defined as outlier if located:
  - 1.5 x (IQR) or more below Q1; or
  - 1.5 x (IQR) or more above Q3
- For example, set of test scores have 25th percentile (Q1) = 70, and 75th percentile (Q3) = 80
- 50% of test scores fall between 70 and 80 and Interquartile Range (IQR) = 80 – 70 = 10
- Test scores are identified as outliers if:
  - Lower than Q1 – 1.5 x (IQR) = 70 – 1.5(10) = 55; or
  - Higher than Q3 + 1.5 x (IQR) = 80 + 1.5(10) = 95

Binning Numerical Variables

- Some algorithms require categorical predictors
- Continuous predictors are partitioned as bins or bands
  - Example: *House value* numerical variable partitioned into: *low, medium or high*
- Four common methods:

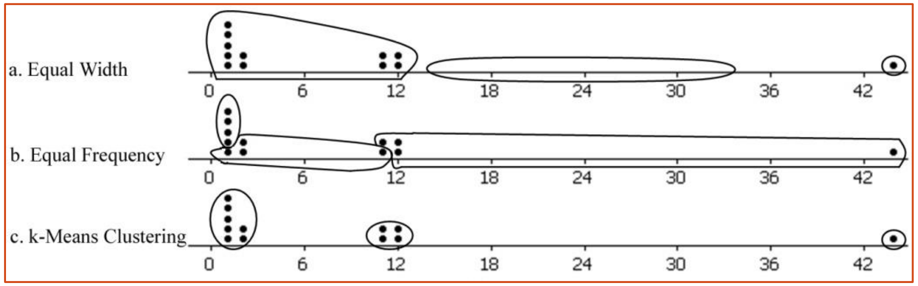
Method	Description	Notes
1. Equal width binning	Divides predictor into k categories of equal width, where k is chosen by client/analyst	Not recommended, since width of bins can be affected by presence of outliers
2. Equal frequency binning	Divides predictor into *k* categories, each having *k/n* records, where *n* is the total number of records	Assumes that each category is equally likely, which is not warranted
3. Binning by clustering	Uses clustering algorithm, like *k-means clustering* to automatically calculate "optimal" partitioning	Methods 3 and 4 are preferred
4. Binning based on predictive value	Methods 1 to 3 ignore the target variable; this method partitions numerical predictor based on the effect each partition has on the value of the target variable	

Binning Numerical Variables *(cont'd)*

- Example: Discretize $X = \{1,1,1,1,1,2,2,11,11,12,12,44\}$ into $k=3$ categories

Method	Low	Medium	High
a. Equal Width	$0 \le X < 15$ Contains all values except one	$15 \le X < 30$ Contains no data	$30 \le X < 45$ Contains single outlier
b. Equal Frequency	First four data values {1,1,1,1}	Next four data values {1,2,2,11}	Last four data values {11,12,12,44}
c. k-means Clustering	{1,1,1,1,2,2}	11,11,12,12	{44}

- How is that in Equal Frequency, values {1,1,1,1,1} are split into two categories? Equal values should belong to the same category
- As illustrated in image below, k-means clustering identifies apparently intuitive partitions



Reclassifying categorical variables

- Equivalent of binning numerical variables
- **Algorithms like Logistic Regression and C4.5 decision tree are suboptimal with too many categorical values**
- Used to reduce the number of values in a categorical field
- Example:
  - Variable *counties* {32 values} → Variable *region* {North, South, East, West}
  - Instead of 32 values, analyst/algorithm handle only 4 values
  - Alternatively, could convert *county* into *provinces*, with values {Munster, Leinster, Connacht, Ulster} or into *economic_level*, (poor_counties, rich_counties, midrange_counties)
- Data analyst should select reclassification that fits business/research problem

Removing variables that are not useful

- Some variables will not help the analysis
  - Unary variables – Take only a single value (a constant).
    - Example – In an all-girls private school, variable gender will always be female, thus not having any effect in the data mining algorithm
  - Variables which are very nearly unary – Some algorithms will treat these as unary. Analyst should consider whether removing.
    - Example - In a team with 99.9% females and 0.05% males, the variable gender is nearly unary.

Variables that should probably not be removed

Variables with 90% or more missing values

- Consider that there may be a pattern in missingness
- Imputation becomes challenging
- Example: Variable donation_dollars in self-reported survey
  - Top 10% donors might report donations, while others do not – the 10% is not representative
  - Preferable to construct a flag variable, *donation_flag*, since missingness might have predictive power
  - If there is reason to believe that 10% is representative, then proceed to imputation using regression or decision tree

Variables that should probably not be removed *(cont'd)*

Strongly correlated variables

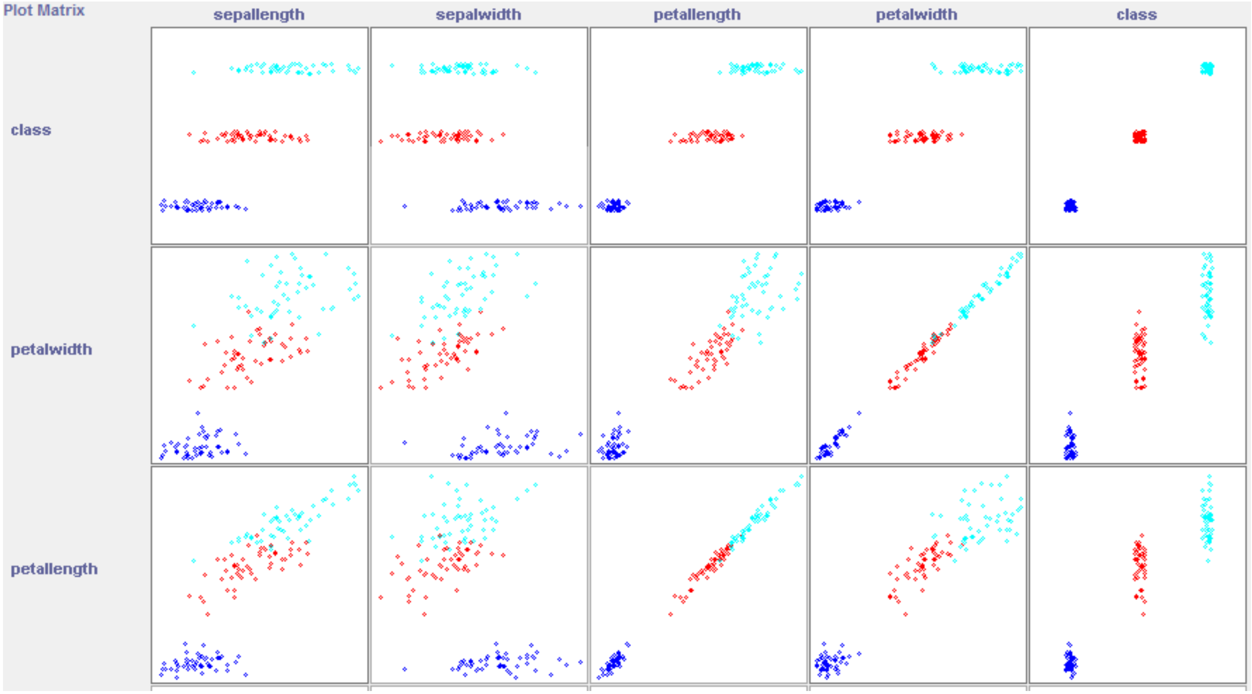
- Important information might be discarded when removing correlated variables
- Example: Variables *precipitation* and '*attendance at the beach*' are negatively correlated
  - They might double-count aspect of the analysis or cause instability in model results – prompting analyst to remove one variable
  - Should perform Principal Component analysis instead, to convert into a set of uncorrelated principal components

Removal of duplicate records

- Records might have been inadvertently copied, creating duplicates
  - Duplicate records lead to overweighting of their data values – therefore, they should be removed
- Example – If ID field is duplicated, then remove it
- But, consider genuine duplicates
  - When the number of records is higher than all possible combination of field values, there will be genuine duplicates

Dealing with Correlated Variables

- Using correlated variables in data model:
- Highly correlated attributes may essentially be measuring the same phenomena. (Multicollinearity)
- May create model instability and produces unreliable results
- Some algorithms like naïve Bayes, removing highly correlated attributes reduces these features attributes voting twice.

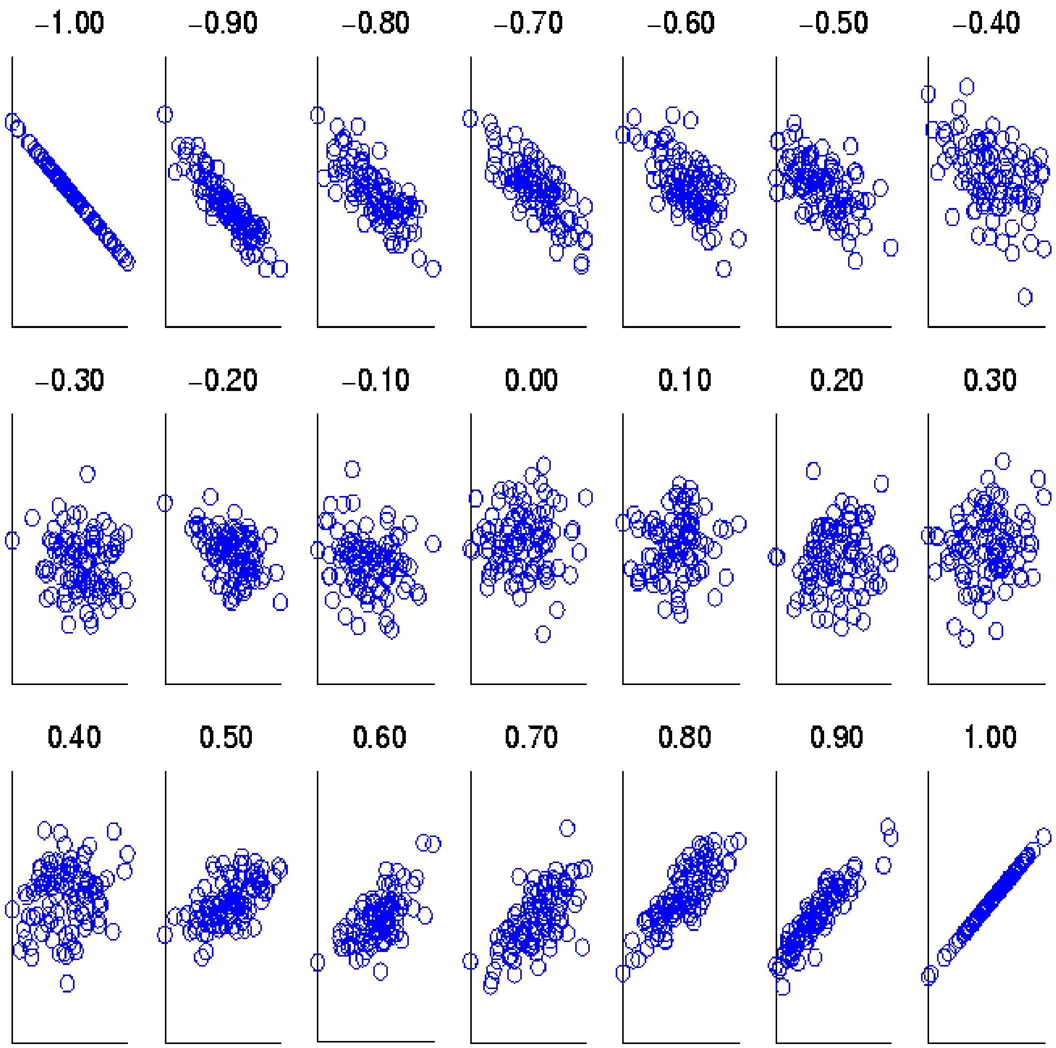


Matrix plot of Iris Data below. Any correlations?

Visualization of Correlation

- Examine attribute to class correlation.
- There are better methods, like attribute selection (next)
- But there is grounds for a visual inspection. Attribute selection may miss correlations that may be of value towards the class attribute.
- Its not often the case, but in some past experience, visualization has produced attributes not identified via attribute selection, that increased the performance of the model.
- Also attributes with low correlation may introduce noise into the model and could be removed.

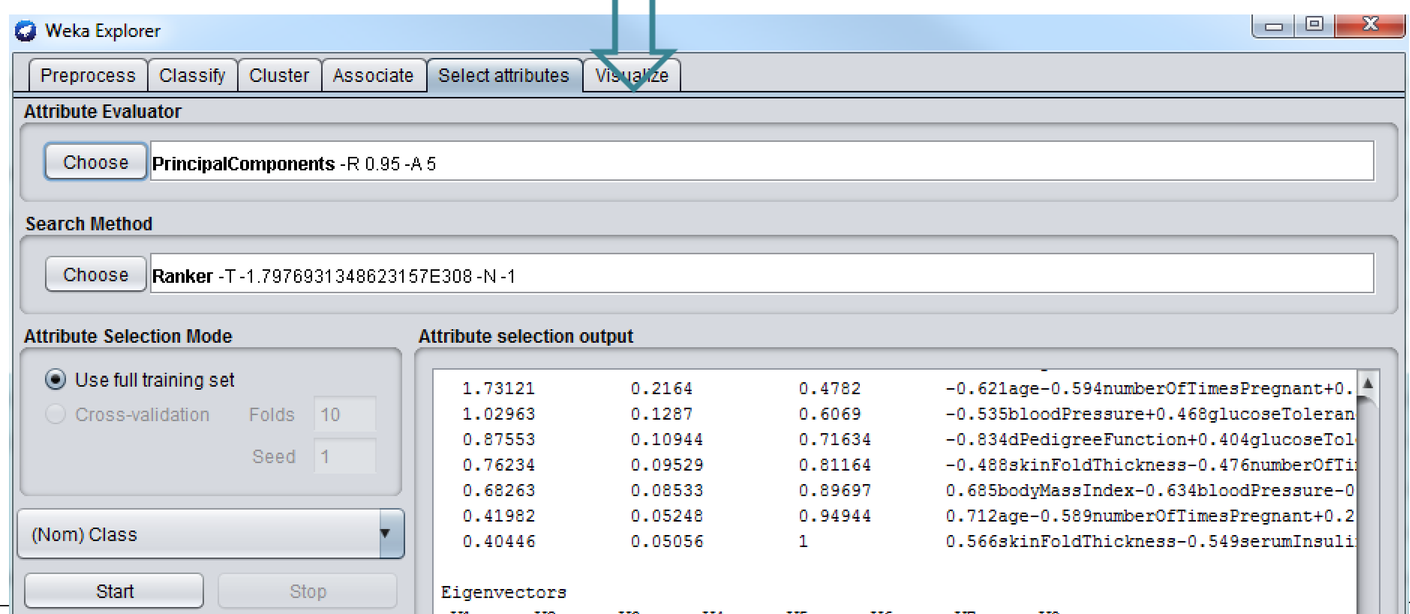
Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Attribute Selection

- We will look at three techniques:
  - correlationAttributeEval
  - InfoGainAttributeEval
  - principalComponents
- WEKA provides tool just for this task.



Correlation Attribute Selection

- Lets run correlation attribute selection:
- Attribute Evaluator: correlationAttributeEval
- This runs Pearson correlation analysis, on each attribute (feature) and compares it to the target variable (class).
- This works for both classification and regression algorithms

Correlation Attribute Selection

Diabetes data set:

The highest ranked attribute : 2
The lowest ranked attribute : 3

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 class):
        Correlation Ranking Filter
Ranked attributes:
 0.4666   2 plas
 0.2927   6 mass
 0.2384   8 age
 0.2219   1 preg
 0.1738   7 pedi
 0.1305   5 insu
 0.0748   4 skin
 0.0651   3 pres

Selected attributes: 2,6,8,1,7,5,4,3 : 8
```

Some of these attributes could in fact reduce the performance of the model.

Information Gain Attribute Selection

- Popular attribution selection technique. – Used in C4.5 Decision Tree classifier

- Information gain measures the amount of **information in bits** about the class prediction, if the only information available is the presence of a feature and the corresponding class distribution. **Concretely, it measures the expected reduction in entropy** (uncertainty associated with a random feature) (Mitchell, 1997)

- Those attributes that contribute more information will have higher information gain value and can be selected while those that do not add much information will have a lower score and can be removed.

- Popular attribution selection technique.

- Used in conjunction with other techniques.

Information Gain Attribute Selection

- Information Gain

- Similar findings to Correlation
- Minor lower attribute shift in ranking.

Let us use a 0.05 arbitrary cut-off

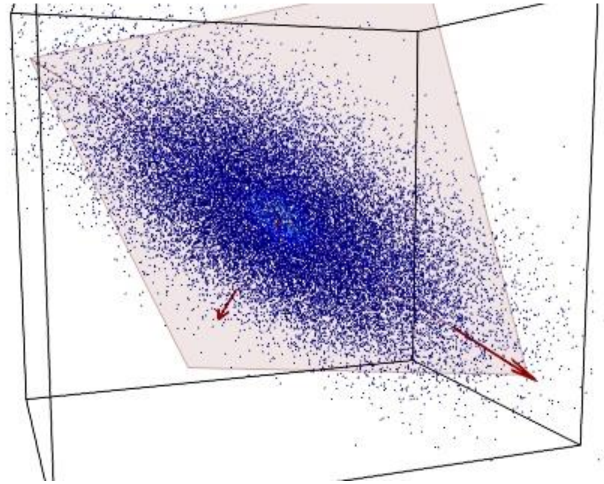
what attributes are removed?

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 class):
        Information Gain Ranking Filter

Ranked attributes:
 0.1901    2 plas
 0.0749    6 mass
 0.0725    8 age
 0.0595    5 insu
 0.0443    4 skin
 0.0392    1 preg
 0.0208    7 pedi
 0.014     3 pres

Selected attributes: 2,6,8,5,4,1,7,3 : 8
```

Data Reduction: PCA

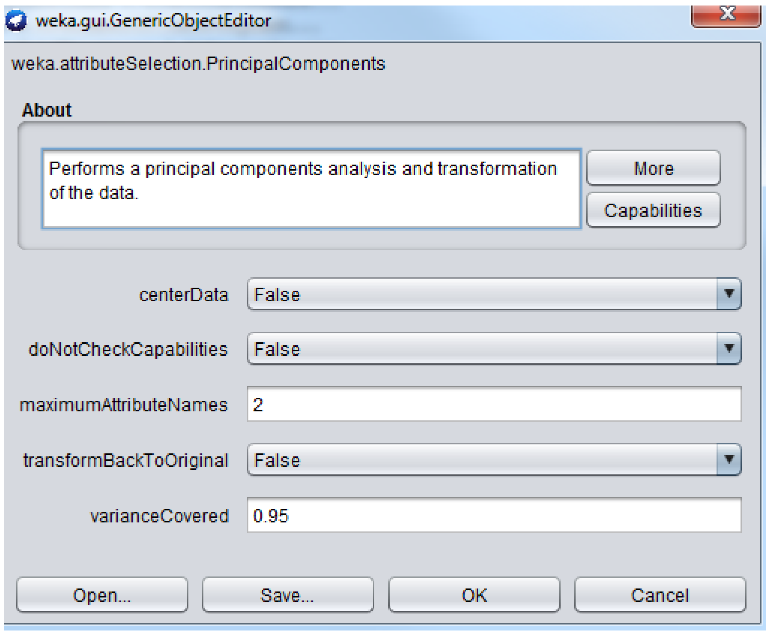
- PCA plots all data points in a multi-dimensional space.

- PCA essentially performs an orthogonal transformation (rotation of data in multi-dimensions) to find the covariance eigenvectors with the largest eigenvalues which represented the largest distribution or effect of the data set, hence selecting the principal component.



Data Reduction: PCA in WEKA

- PCA is also useful as an attribution selection technique.
- We may have to add some criteria before we run the algorithm, select the Evaluator by clicking on it:
- MaximumAttributeNames
- This will help with attribute selection and running PCA separately.



Data Reduction: PCA in WEKA

- PCA is using max 5 attributes:

eigenvalue	proportion	cumulative	
2.09438	0.2618	0.2618	-0.452mass-0.44skin-0.435insu-0.393plas-0.36pres...
1.73121	0.2164	0.4782	-0.621age-0.594preg+0.332skin+0.251insu-0.184pres...
1.02963	0.1287	0.6069	-0.535pres+0.468plas+0.433pedi-0.362mass+0.337insu...
0.87553	0.10944	0.71634	-0.834pedi+0.404plas+0.35 insu-0.081preg-0.071age...
0.76234	0.09529	0.81164	-0.488skin-0.476preg+0.466plas-0.347insu+0.328pres...
0.68263	0.08533	0.89697	0.685mass-0.634pres-0.271insu+0.194preg+0.094plas...
0.41982	0.05248	0.94944	0.712age-0.589preg+0.282skin-0.192pres-0.132insu...
0.40446	0.05056	1	0.566skin-0.549insu+0.45 plas-0.342mass-0.212age...

Eigenvectors

V1	V2	V3	V4	V5	V6	V7	V8	
-0.1284	-0.5938	-0.0131	-0.0807	-0.4756	0.1936	-0.5888	0.1178	preg
-0.3931	-0.174	0.4679	0.4043	0.4663	0.0942	-0.0602	0.4504	plas
-0.36	-0.1839	-0.5355	-0.056	0.328	-0.6341	-0.1921	-0.0113	pres
-0.4398	0.332	-0.2377	-0.038	-0.4879	0.0096	0.2822	0.5663	skin
-0.435	0.2508	0.3367	0.3499	-0.3469	-0.2707	-0.132	-0.5486	insu
-0.4519	0.101	-0.3619	-0.0536	0.2532	0.6854	-0.0354	-0.3415	mass
-0.2706	0.1221	0.4332	-0.8337	0.1198	-0.0858	-0.0861	-0.0083	pedi
-0.198	-0.6206	0.0752	-0.0712	-0.1093	-0.0334	0.7121	-0.2117	age

Data Reduction: PCA in WEKA

- PCA can also be run in the pre-process tab.

- Be careful, and do not reduce every attribute.

- Use a mixture of techniques and examine the outputs carefully to justify your mode.

Attribute removal

- Once you have identified attributes to remove, we can easily do this in WEKA.
- Both informationGain and correlation ranked bloodPressure (pres) as the lowest attribute.
- Remove it from the data set and examine any differences in the accuracy….

A word about ID fields

- ID fields have different value for each record
- Might be hurtful, with algorithm finding spurious relationships between ID field and target
- Recommendation: Filter ID fields from data mining algorithm, but do not remove them from the data, so that analyst can still differentiate the records