Data Mining

Topic 5
Data Preparation

Recap on Module topics

Module topics follow CRISP-DM

✓ Business Understanding (done today)

✓ Data Understanding & EDA

❑ Data Preparation Techniques

➢ Data Mining (more to do)

- Classification
- Clustering
- Association analysis (time permitting)

✓ Model Evaluation

Lecture Overview

1. Why do we need to carry out Data Preparation?
2. Data selection
3. Handling missing data
4. Handling errors and outliers
5. Sampling

Steps 2 – 5: Data Cleaning

Lecture Overview

6. Scaling / Normalising the data
    - Min-max
    - Z-transform

7. Dimensionality Reduction

8. Attribute construction

9. Data type conversions

Steps 6 – 9: Data Transformations

1. Why do data preparation

Data preparation can comprise of up to 80% of the time spent analysing a dataset.

Why do we need to pre-process data?

- Maybe we have too much or too little data
  - Too much: can sample rows, filtering attributes
  - Too little: add additional sources of data, bootstrap
- Data in the dataset may be incomplete or noisy
  - Missing values
  - Fields that are obsolete or redundant
  - Outliers
- Data may be too detailed
  - Binning
- Or not in the correct format
  - Convert: nominal to numeric; numeric to binary, etc…
- Or possibly combining two attributes in some way exposes a pattern not evident from the original attributes.

Objectives when preparing data

1. To assist the mining algorithm in finding the patterns in the dataset
   - Improving quality
   - Exposing as much information content as possible to the mining tool

2. Reduce the dimensionality (num of attributes) or size (number of rows) in the data set

DO NOT CHANGE the information content (patterns) of the data set when preparing it

Why Is Data Preprocessing Important?

- To improve the accuracy of the mining algorithm
- And to improve the validity and usefulness of the results

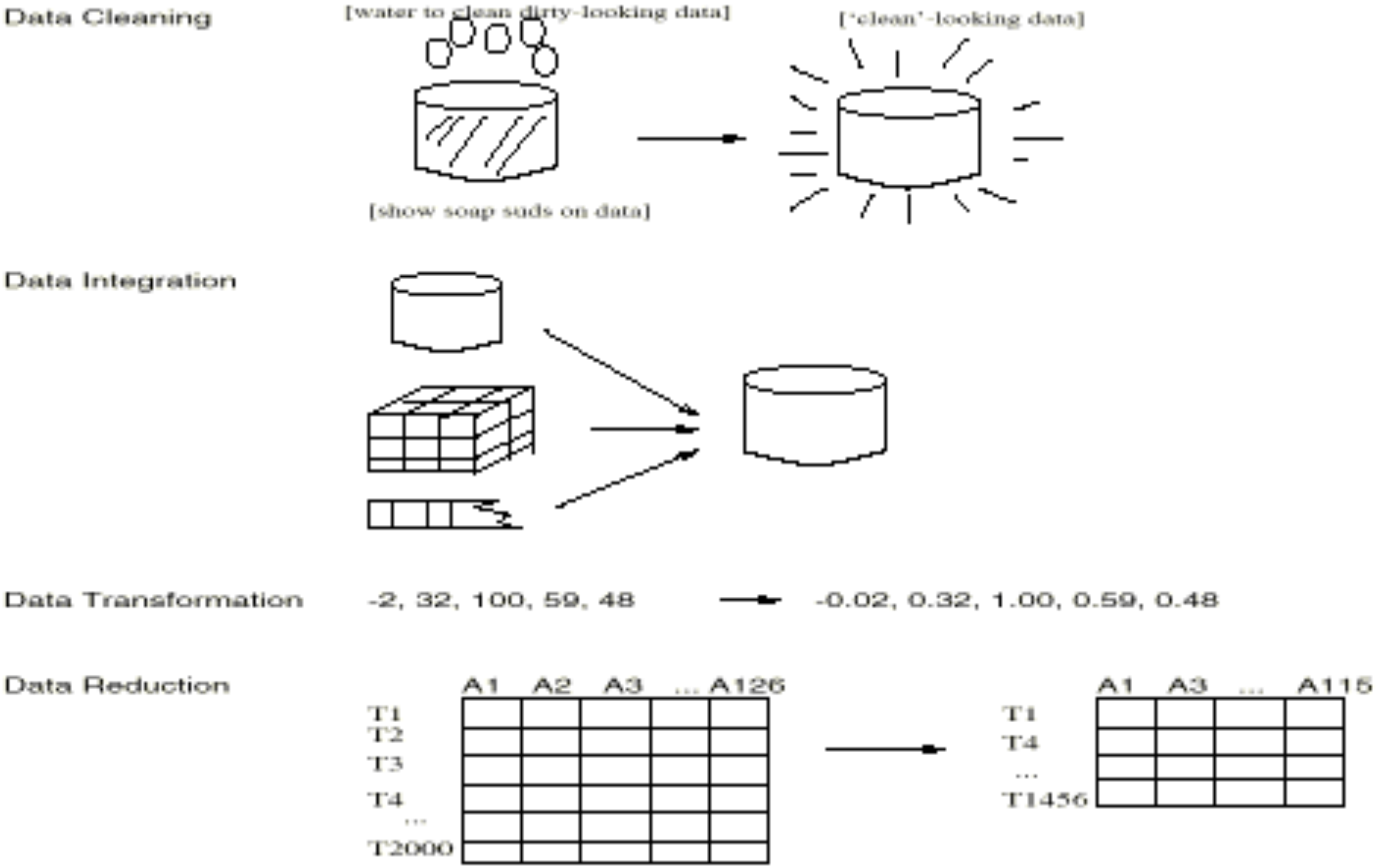
Poor quality data = poor quality mining results!

- Quality decisions must be based on quality data

Major Tasks in Data Preprocessing

1. Data cleaning
   - Selecting data
   - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
2. Data integration
   - Integration of multiple databases, data cubes, or files
3. Data transformation
   - Normalization, aggregation, generate new attributes
4. Data reduction
   - Obtains reduced representation in volume but produces the same or similar analytical results

Forms of data preprocessing



Data Cleaning

Data Integration

Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction

2. Data Cleaning

a) Selecting data

b) Filling missing values

2.a. Data Selection

- Having explored the data during the data understanding phase, data selection involves choosing which data from the data set is worth including in the analysis/mining.
- The decision on what rows and columns to include is a subjective one, and will be influenced by your understanding of the data set.
- Primarily there are two reasons for excluding data:

1. **Low information content,** either because of the:
   - **Measurement** type (e.g. nominal),
   - **Quality** of the data: A high proportion of missing values or errors in the column.
   - Information is **duplicated** in other columns (RM: remove correlated attribute)

2. The data **is not relevant** to the required outcome. Based on your knowledge of the data set, it is unlikely that this column will have any bearing on the final outcome. This decision could be made as part of an effort to reduce the dimensionality of the data set..

Selecting Rows and Columns in rapidminer

- Rapid miner provides a number of selection and filters for selecting rows (data) and columns (attributes) under:
  - data transformations / filtering to select rows in the dataset
  - data transformations / attribute set reduction and transformations/selection to select particular columns in the dataset
- Operators select attributes and filter examples allow the user to specify which column(s) or row(s) to exclude.

RAPID MINER

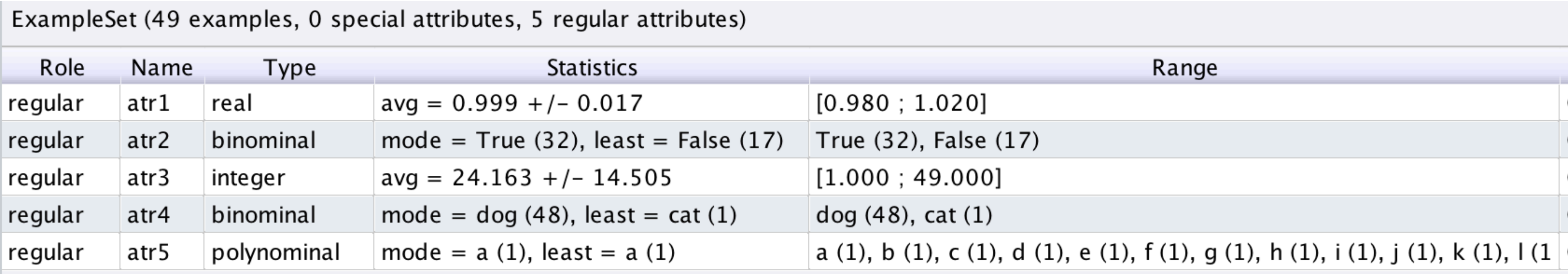
Selecting Rows and Columns in rapidminer

- Rapidminer can also identify SOME useless attributes based on the following criteria:
  - The attribute has the same value in almost every row
    - For numeric attributes, this means an attribute with a very small standard deviation.
  - The attribute is nominal, and has a different values in almost every row

Operator name: Remove Useless Attributes

Exercise

- Which of the following attributes could be labeled as useless, and why?



ExampleSet (49 examples, 0 special attributes, 5 regular attributes)

Role	Name	Type	Statistics	Range
regular	atr1	real	avg = 0.999 +/- 0.017	[0.980 ; 1.020]
regular	atr2	binominal	mode = True (32), least = False (17)	True (32), False (17)
regular	atr3	integer	avg = 24.163 +/- 14.505	[1.000 ; 49.000]
regular	atr4	binominal	mode = dog (48), least = cat (1)	dog (48), cat (1)
regular	atr5	polynominal	mode = a (1), least = a (1)	a (1), b (1), c (1), d (1), e (1), f (1), g (1), h (1), i (1), j (1), k (1), l (1

2. b. Missing data

Many mining algorithms will accept a certain percentage of missing data. If left missing/empty, the mining tool will either:

- Ignore them
- Automate the replacement

However you get better results if the problem is handled before mining starts, either by filling the missing values, or removing them.

Rapidminer operator:
Impute Missing Values

Missing values

- As a general rule:
  - if a **column** of data has a **high** proportion of missing values (> 40%), then it is **removed** from the data set.
  - If a column has a **low** proportion of missing values (<5%), then just those **rows** that contain missing values can be **removed**.

Removing a small number of rows from a large data set generally does not affect patterns in the data.

In removing a column, you will lose all patterns involving that column. You could preserve the missing value pattern for the column (explained below.)

Replacing missing values

- If its not practical to remove the missing values, then the alternative is to fill the missing values.

The primary objective when filling missing values is not to guess what the original value might have been,

but to fill it with a value that causes the least harm to the existing patterns of the data set.

Handling missing data

- How do we replace missing data? The most common options are to:

  i. Use a **statistic measure** such as mean (numerical variables) or mode (categorical variables) to replace the missing value

  - Does the least harm to the variables distribution, and is easy to calculate

  ii. Imputation: set the column which has missing data as the class label, and use a classification/prediction algorithm to predict the missing value.

  - Does the least hard to the relationships between variables, but is more time consuming to do.

i. Replacing Missing Data

With the mean or mode

For numeric columns, replace missing value with the mean, so for example:

1,?,3,4,?,4,5

becomes:

1,3.4,3,4,3.4,4,5

For categorical columns, replace missing value with the mode, so for example:

male,?,male,male,?,female, female

becomes:

male,male,male,male,male, female,female

i. Replacing Missing Data

With the class mean or mode

Replacing missing data with the mean gets more accurate results if each class is done seperately:

Income	Gender	Class label
20000	male	C0
23000	male	C0
27000	?	C0
?	female	C0
34000	female	C1
37000	male	C1
?	female	C1
32000	?	C1

becomes

Income	Gender	Class label
20000	male	C0
23000	male	C0
27000	male	C0
23333	female	C0
34000	female	C1
37000	male	C1
34333	female	C1
32000	female	C1

- Income is replaced with the average income for that class
- Gender is replaced with the most frequent gender (mode) for that class

ii. Replacing Missing Data

Preserve relationships between variables

- Generally modeling tools are investigation how one variable varies with respect to other variables
- Method to replace missing values should preserve these existing relationships

E.g. Suppose you have a data set with income level and size of house. If all missing income levels are replaced with a single value that preserves the mean, then no account is taken of the relationship between house size and income. In fact the relationship could get distorted as a result of filling missing values with the mean.

ii. Replacing Missing Data

Preserve relationships between variables

- Problem:
  - The missing value should be replaced by a value that preserves the relationship between variables
  - However the goal of mining the data is to discover what these relationships are
  - They are not known in advance when the data is being prepared.
- So what do you do?

- Note: the goal here is **not** to find a value close to the original value, but to find one which does not distort existing relationships

Replacing Missing Data

2. Preserve relationships between variables

Answer: Use a classification algorithm to predict the missing values based on existing patterns in the dataset:

- Set the column with the missing values as the class label
- Select the columns to be used to predict this class label (i.e. related columns that do not have missing data)
- Train a model with rows that do not have missing values
- Apply the model to the rows with missing values

- This is known as imputing the missing value.

RAPIDMINER Operator is:

Impute Missing Values

Missing value pattern

- Before replacing any values, it may be worth recording what values were missing. Maintaining the pattern in which missing values occur can be itself useful information.
- Missing Value Pattern (MVP)
  - is a column of flags which have a value of P, for present, and E, for empty.

Cust ID	**Age**	**Salary**	**Drivers License**	**MVP**
001	25		Y	PEP
002	25	€25,000	Y	PPP
003		€30,000		EPE
004				EEE
005			N	EEP

Handling missing values in Rapid Miner

Rapid Miner supports two algorithms for filling missing values:

1.**Replace Missing Values** If a value is missing, it is replaced by one of the functions "minimum", "maximum", "average", and "none", which is applied to the non missing attribute values of the example set. "none" means, that the value is not replaced. For nominal attributes the mode is used for the average, i.e. the nominal value which occurs most often in the data. For nominal attributes, and replacement type zero the first nominal value defined for this attribute is used. The replenishment "value" indicates that the user defined parameter should be used for the replacement. There is an example of this in RapidMiners samples folder under **preprocessing/07_missing value replenishment.xml**

2.**Impute Missing Values**: imputes missing values by learning models for each attribute (except the label) and applying those models to the data set. The learner which is to be applied has to be given as inner operator. In order to specify a subset of the example set in which the missing values should be imputed (e.g. to limit the imputation to only numerical attributes).

Delete attribute

Replace with the mean

Exercise

Delete row

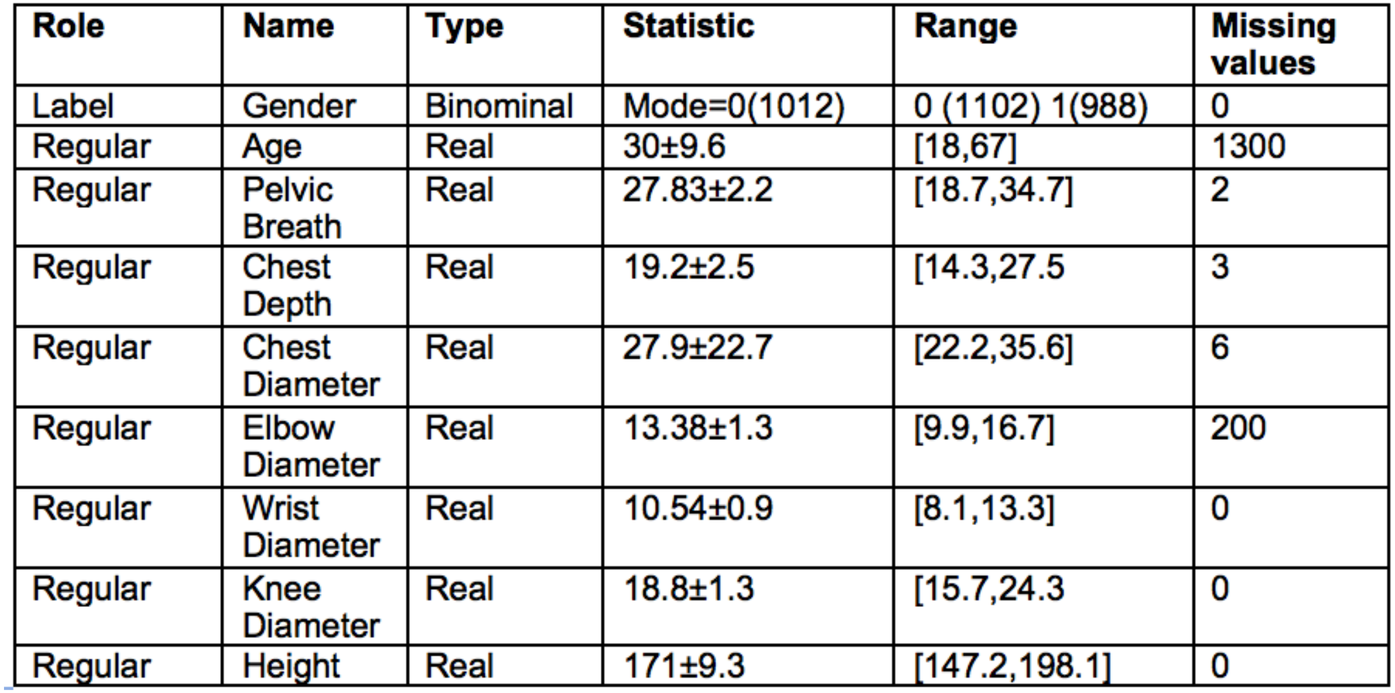
Impute values from other attributes

What approaches would you take to handle the missing data in the following dataset:

- The dataset has 20 attributes and 1000 rows.
  - One attribute has over half of it’s values missing.
  - Another attributes has 100 missing values, but seems to be very relevant to predicting the class label.
  - A third attribute also has 100 rows missing, but has low information content as it has a different nominal value in each row.
  - All remaining missing values are confined to five rows of data.

Past exam question

- The table below shows the meta data for a dataset of skeletal measures, used to determine **gender**. The dataset has 8 attributes, and **2000** rows. For each of the **five** attributes with missing data, recommend a suitable approach for handling their missing values. Justify each of your recommendations. (**9 marks**)



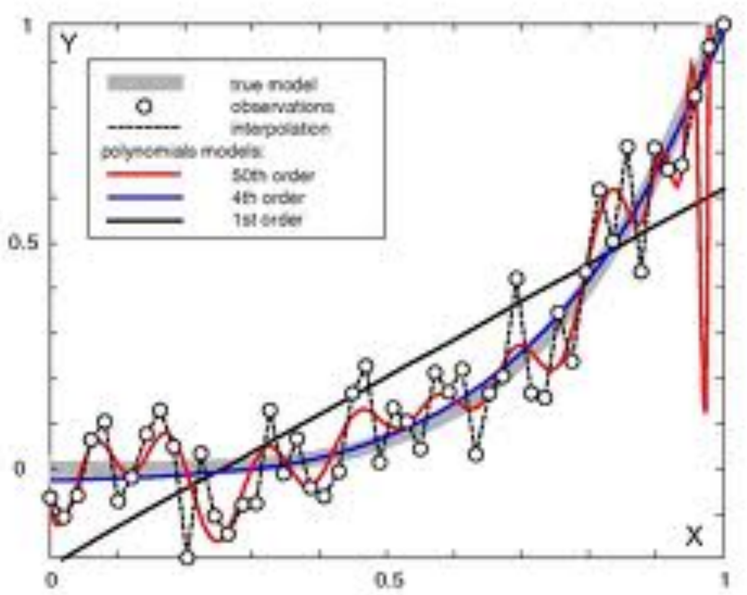
Role	Name	Type	Statistic	Range	Missing values
Label	Gender	Binominal	Mode=0(1012)	0 (1102) 1(988)	0
Regular	Age	Real	30±9.6	[18,67]	1300
Regular	Pelvic Breath	Real	27.83±2.2	[18.7,34.7]	2
Regular	Chest Depth	Real	19.2±2.5	[14.3,27.5	3
Regular	Chest Diameter	Real	27.9±22.7	[22.2,35.6]	6
Regular	Elbow Diameter	Real	13.38±1.3	[9.9,16.7]	200
Regular	Wrist Diameter	Real	10.54±0.9	[8.1,13.3]	0
Regular	Knee Diameter	Real	18.8±1.3	[15.7,24.3	0
Regular	Height	Real	171±9.3	[147.2,198.1]	0

Summary - Missing values

- Replacing missing values is important
- The objective is NOT to determine what the missing value was, but to find a replacement value that does the least harm to existing relationships in the data set
- It may be useful to record the MVP before replacing any values
- Values can be replaced by
  - Preserving the variability of the variable itself – mean or mode
  - Or preferable preserving the relationships between it and other variables

- No method is ideal. It’s a trade off between computation time and yield

4. Poor quality data: Noise, Errors and Outliers



Handling errors and outliers

- ***Noise*** is any random error or variance in a measured variable
  - Visible as greater variance in variable values or
  - An outlier value

How to Handle Noisy Data?

- Effect of noise can be reduced by ***smoothing*** out a variables values using:
  a) Binning (one column)
  b) Regression (related columns)
  c) Aggregation or Clustering (combining rows)

  Used to smooth out the data to reduce/remove effect of outliers (noise)

4.a) Binning

also called discretisation

Binning can be used on noisy columns to reduce the impact of noise

OR

Can be used to convert a numeric attribute to a nominal attribute

Binning

- **Binning** methods smooth a sorted data value by consulting its ‘neighbourhood’ (i.e. the values around it).
- The sorted values are distributed into a number of ‘buckets’ or ‘bins’
- For data smoothing, bins are calculated such that each bin has approximately the same number of values.
- Then within each bin
  - replace each value by the mean or mode of that bin,
  - the median of that bin, or
  - smooth by bin boundaries which means rounding each value to the nearest boundary value in its bin.

Binning

Equi-width binning: This is the most common method of binning data.

- Example: If the number set goes from 1 to 49, and you want 5 bins, the bins are 1-9, 10-19, 20-29, 30-39, 40-49.

- In general, if A is the lowest value in the list, and B is the highest value in the list, and you want N bins, the width of each bin is (B-A)+1/N.
- An appropriate label is given to each bin.

Essentially you are converting the variable into a categorical variable, reducing both information content and noise.

Equi-width Binning

Decide on bins and give each bin a label . . . .

category 1	0 to 9
category 2	10 to 19
category 3	20 to 29
category 4	30 to 39
category 5	40 to 49

Original attribute	*Generated attribute*	*Or Smooth to Bin Mean*
1	category 1	4
3	category 1	4
4	category 1	4
7	category 1	4
11	category 2	14
14	category 2	14
17	category 2	14
23	category 3	25
26	category 3	25
27	category 3	25
30	category 4	33
32	category 4	33
36	category 4	33
47	category 5	48
48	category 5	48
49	category 5	48

Equi-depth (frequency) binning

- Equi-width binning does not work well for data with outliers, or uneven distributions. For example take the following list for 'Number of Cars': 0,0,0,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,3,7,15
- Dividing this into 3 bins, for example, results in most of the data going into the first and second bin.

- An alternative approach is to calculate bins such that each bin has approximately the same number of values.

  Then within each bin, replace each value by the mean or mode of that bin, the median of that bin, or smooth by bin boundaries which means rounding each value to the nearest boundary value in its bin.

Equal-depth (frequency) partitioning: Divides the range into *N* intervals, each containing approximately same number of samples

Edqui-depth Binning Example

Sorted data (price in euro): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

1. Partition into (equi-depth) bins:

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

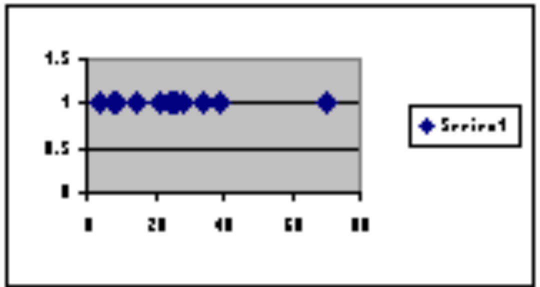
2. Smoothing by bin means:

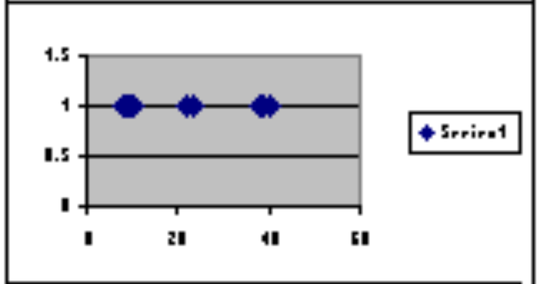
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

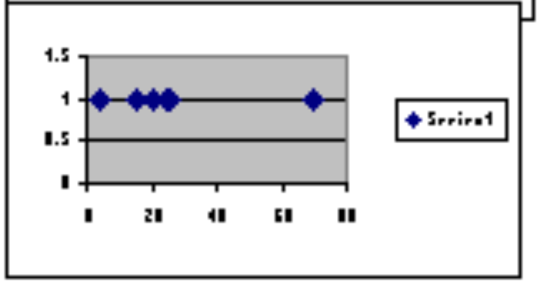
3. Smoothing by bin boundaries:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

smooth.xls







Question: Would this work for categorical variables?

Exercise

The table has a list of 12 values in the range [1,15]

1. Convert the list into three equi-width bins called small, medium and large
2. Convert the list into three equi-depth bins, again called small, medium and large
3. Which method makes more sense for this list of numbers?

List of values	Equi-width bin label	Equi-depth bin label
1		
2		
3		
4		
4		
5		
6		
8		
10		
12		
14		
15		

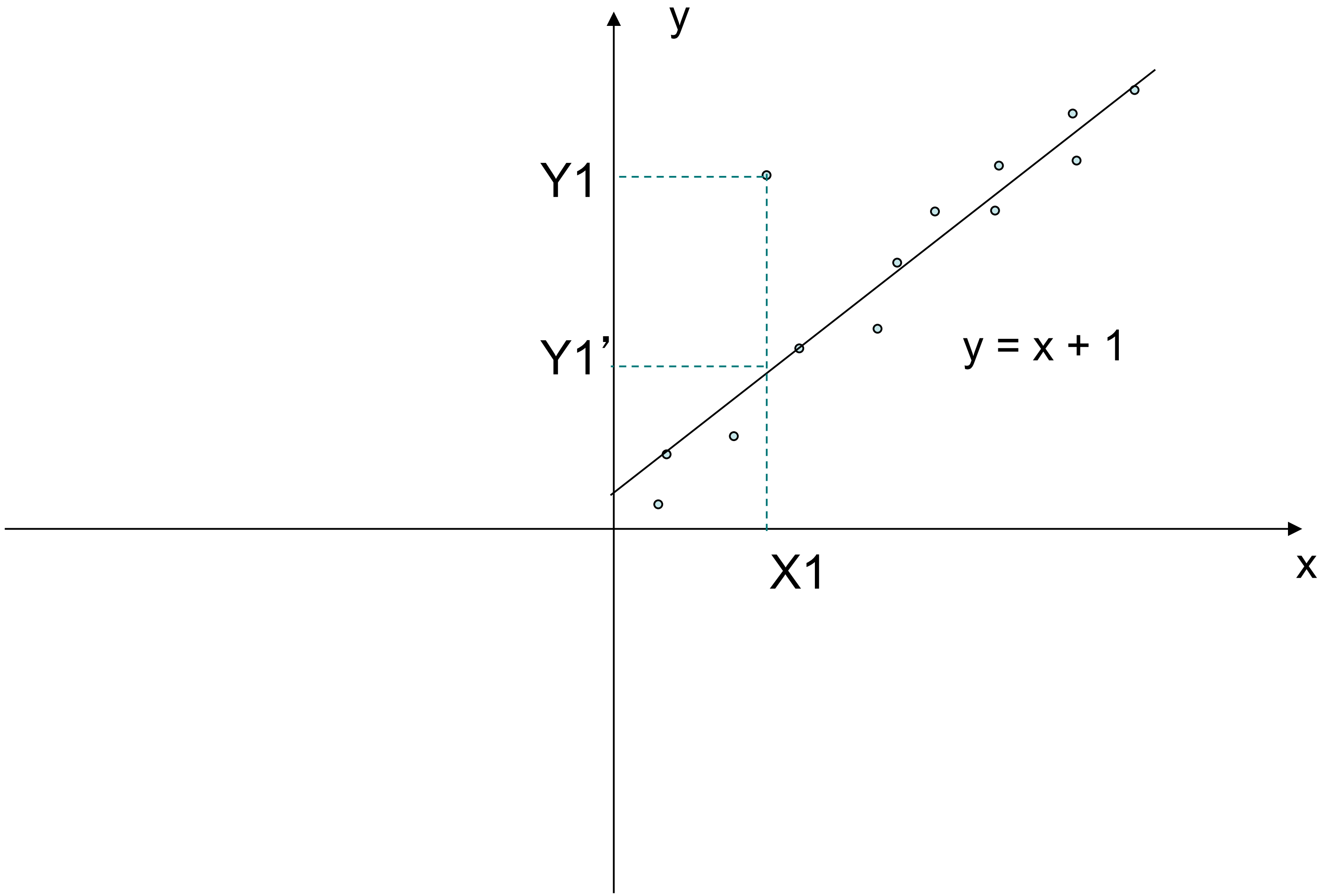
4. b) Regression

Predicting one numeric variable from another

Regression

- Binning only applies to one column of data at a time.
- A second technique for smoothing data, and so reduce the impact of noise or errors, is to use **regression**.
- This requires plotting the variable against one or more related variables, and using a regression algorithm to find a 'best fit' function for the variables values.
  - i.e. best line to fit two variables, so that one variable can be used to predict the other
- Each point would then have its values changed to one that lies directly on the regression function.

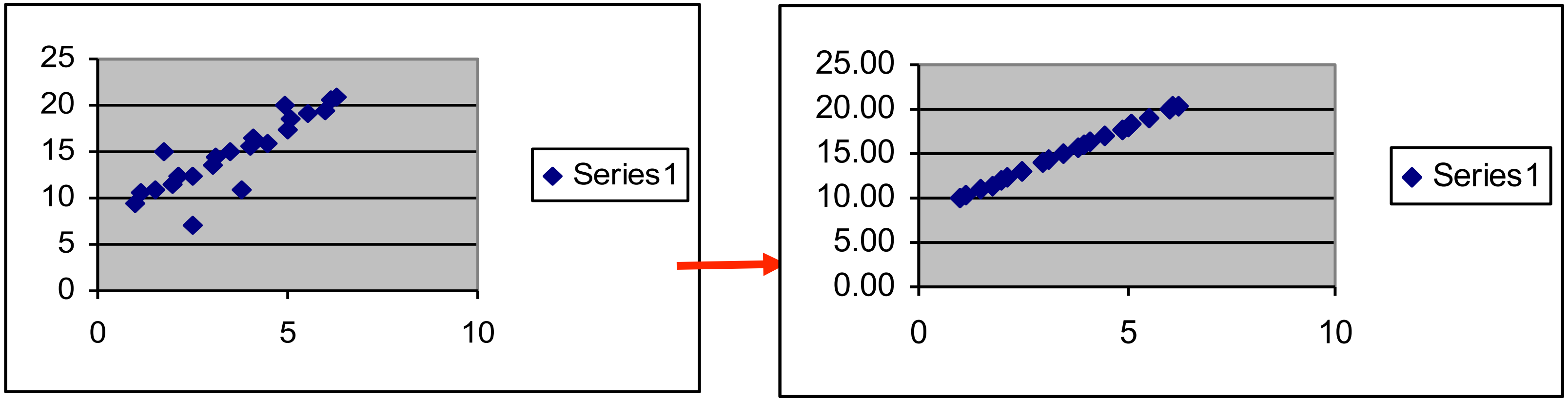
Regression



Regression on the first graph below recognized the formula 2(x)+8=y.
Recalculating Y using this formula resulted in the second, smoothed graph.

Data before regression

Data after regression



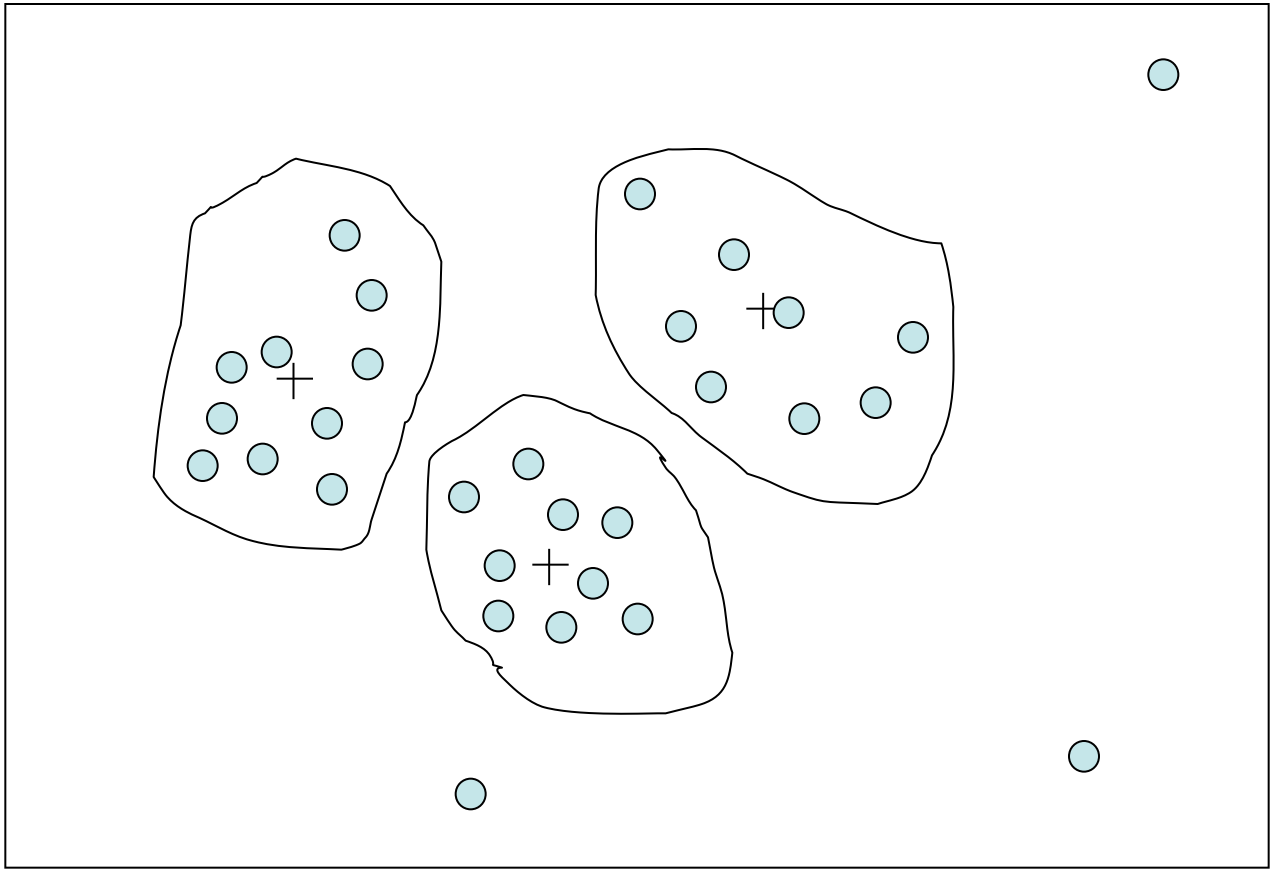
smooth.xls

4. c) Outlier detection

Outlier detection

- Outlier detection is a range of techniques that look for rows which are dissimilar to other rows in the dataset.
- An outlier is an unusual combination of values across a number of variables.
- Remember k Nearest Neighbour?
    - Outliers can be identified as the rows that have a largest distance from other rows in the dataset.
- Each outlier can then be examined manually to identify if it is erroneous, or worth keeping.

Outlier Detection



Data Smoothing: Summary

- Binning reduces variability in a single variable, as is a good technique to use for decision tree modeling.
- Regression reduces noise in a variable if it is related (correlated) to another variable in the dataset
- Outlier detection can identify outlier rows where the combination of values is unusual

Outlier detection in RapidMiner

- Take a look at samples/processes/02_preprcessing/18_OutlierDection
- This process uses Detect Outlier (distance) where you say hoy many outliers to find (e.g. 10), and RM will return the 10 rows most different from other rows in the dataset.
- An alternative operator is Detect Outlier (LOF) which returns a number indicating the extent to which a row is an outlier. Plotting a histogram of the result gives an indication of what the correct number of outliers is. Try it . . .

5. Scaling

Also called normalising or discretization

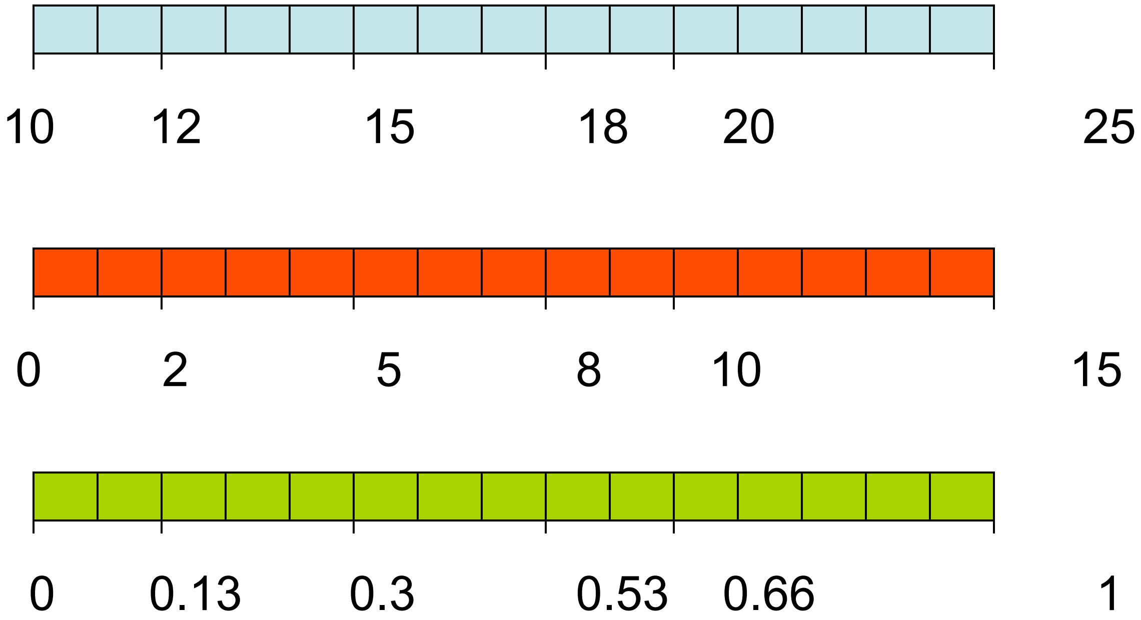
Normalisation Overview

- Mining tools work best if all numeric data are in a similar range.
- Converting all numeric data to fall between a particular range is called **normalising the data**.
- We will also look at normalising the distribution curve of the data, i.e. chaging an attribute so that it has an average of 0, and a standard deviation of ±1.

Normalising the data in this context has NOTHING in common with putting data into 3rd normal form for database design

Min-Max Normalisation

- To normalise a set of values so that they fall between 0 and 1 is straight forward:
- Take the following list if integers: 10, 12, 15, 18, 20, 25



1. Subtract 10 from each number.
2. Multiply each number by 1/15 = 0.06

Have you lost any information by doing this?

Exercise

- Normalise the following list of integers to a range between 0 and 1
  - 25, 41, 30, 48, 50, 35

Normalise using Z-transforms

- Setting the range before hand assumes that the full range of valid values is known, which is not always the case.
- **Example:** Mortgage applicants. Model was prepared from a sample with max salary of 100,000. Outliers were ignored, and given a null score. People with salaries > 100,000 were given a null score, and so deemed to be too big a risk for a mortgage!

Normalise using Z-transforms

- A Z-transform scales an attribute so that the mean is 0 and the standard deviation is 1.
- This allows for any range of values.

- A z-transform is calculated by:
  1. subtracting the mean from each value,
  2. dividing the result by the standard deviation.

Normalise using Z-transforms

Using the same list of numbers again:

	Original list	1. Substract the mean	2. Divide by the standard deviation
	10	-6.67	-1.21
	12	-4.67	-0.85
	15	-1.67	-0.30
	18	1.33	0.24
	20	3.33	0.61
	25	8.33	1.51
mean	16.67		0
st. dev.	5.50		1

Normalisation in RapidMiner

- Rapidminers operator for normalisation is Normalize. Its parameters allow you select how to normalise the data:
  - Range transformation: you set the min and max values.
  - Z-transform: scales the attributes so that each column has a mean of 0, and variance of 1.

Exercises

1. Use binning to reduce the impact of noise in the following variable. The values have already been sorted:

   4,6,8,8,13,15,15,17,24,25,27,36

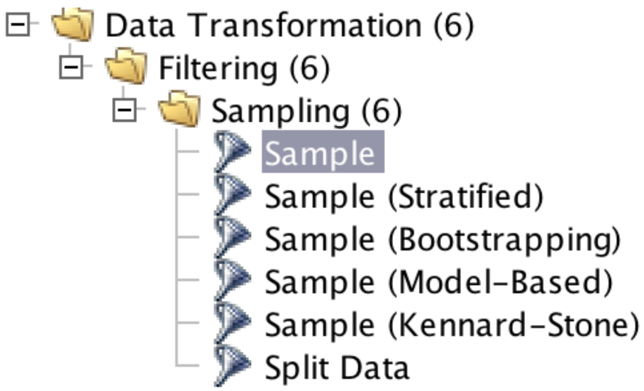
2. Plot the following data, and illustrate in the diagram where the points might be adjusted to if using linear regression to smooth the data.

X	1	3	5	7	8	10	11	14	15
Y	11	25	23	27	36	48	45	55	66

3. Normalise the following list of integers to the scale [0,1]:

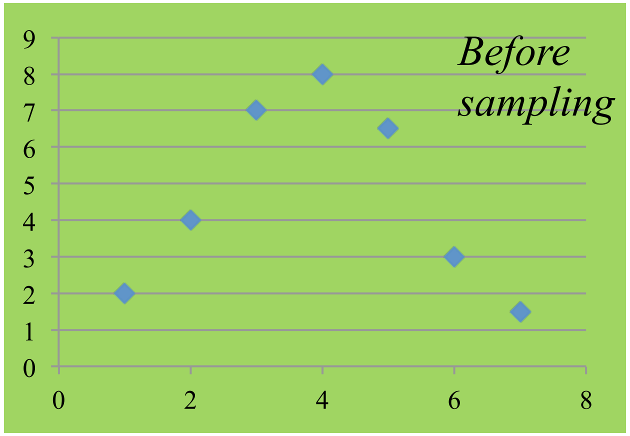
   1,2,4,6,7,8,9

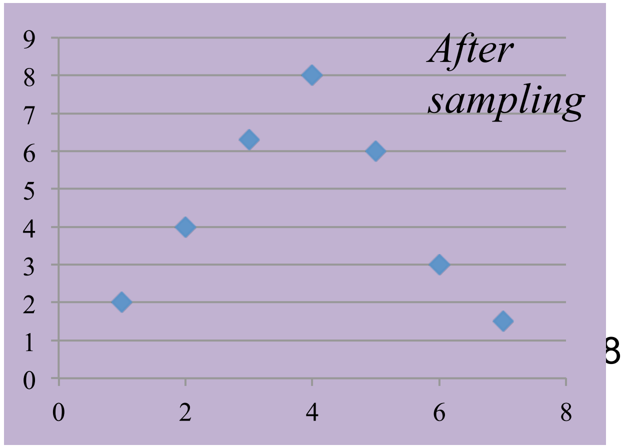
6. Sampling


Data Transformation (6)
Filtering (6)
Sampling (6)
Sample
Sample (Stratified)
Sample (Bootstrapping)
Sample (Model-Based)
Sample (Kennard-Stone)
Split Data

Sampling

- Sampling is used to select a subset of the data set for analysis, and is done to reduce the size of the data set allowing more expensive algorithms to be used.
- Using a sample will work almost as well as using the entire data set provided the sample is representative. In other words the sample must contain the same patterns as the original data set.
- One way to measure this is the compare means and/or standard deviations in the sample to those in the original data set.





Sampling

The sample size must be large enough to ensure that it represents all classes of objects and patterns from the original data set. The more complex the data set, the larger that sample size needs to be.

The optimal size for a sample can be hard to determine. One approach is to use progressive sampling. Start with a small sample, and assess the accuracy of the models produced by that sample. The accuracy of these models will change as the sample size increases, but at some point the rate of change in the models accuracy levels off, indicating that the sample size is approaching an optimal size.

Sampling techniques

The most common sampling techniques used is random sampling, where there is equal probability of selecting any row in the data set.

- The user would specify in advance the percentage of rows to be included in the sample.

However if a data set contains many different types of objects, random sampling can fail to adequately represent all object types.

An alternative is stratified sampling where groups of objects are specified in advance, and then objects[1] are selected randomly from each group. Typically objects are grouped by class variable.

[1] An object is a row of data

Sampling

award	attendance	hours studying	level of interest
1h	1	200	0.9
1h	0.9	140	1
2h1	0.5	250	0.8
2h2	0.7	175	0.6
pass	0.5	95	0.5
2h2	0.8	120	0.5
pass	0.6	110	0.6
1h	1	180	0.9
2h2	0.7	145	0.6

A random sample of 50% may not include students getting 2h1 and so could miss one of the groups of students completely.

*In stratified sampling, **award** would be identified as the class label to be used to group the data. The sampled data would include rows from each group, i.e. examples of students for each award.*

Other sampling methods

- Stratified sampling works well if all possible classes of objects are known, and identifiable, in advance, but frequently this is not the case.

- There are a range of other sampling techniques that attempt to maximise the variability of samples in the sample set. One of the most popular, and is implemented in Rapid Miner, is a method developed by Kennard and Stone. . .

Kennard-Stone sampling

Kennard Stone sampling is a sequential technique:

1. The first two rows selected are the two rows which are furthest apart in the data set

2. Subsequent rows are added by adding the row which is furthest from the objects currently in the sample.

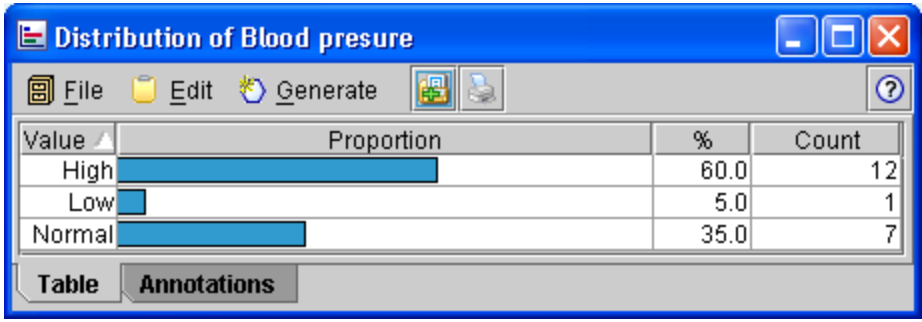
Sampling with replacement

- Bootstrap sampling is a technique which can be used to increase the size of the dataset.
- Standard sampling techniques do not allow the same row to be picked twice. Bootstrap **sampling does** and so is referred to as sampling with replacement, i.e. the same row can be picked a number of times.

Continuing from the student example on an earlier slide, bootstrap sampling could include the 2h1 student twice or more.

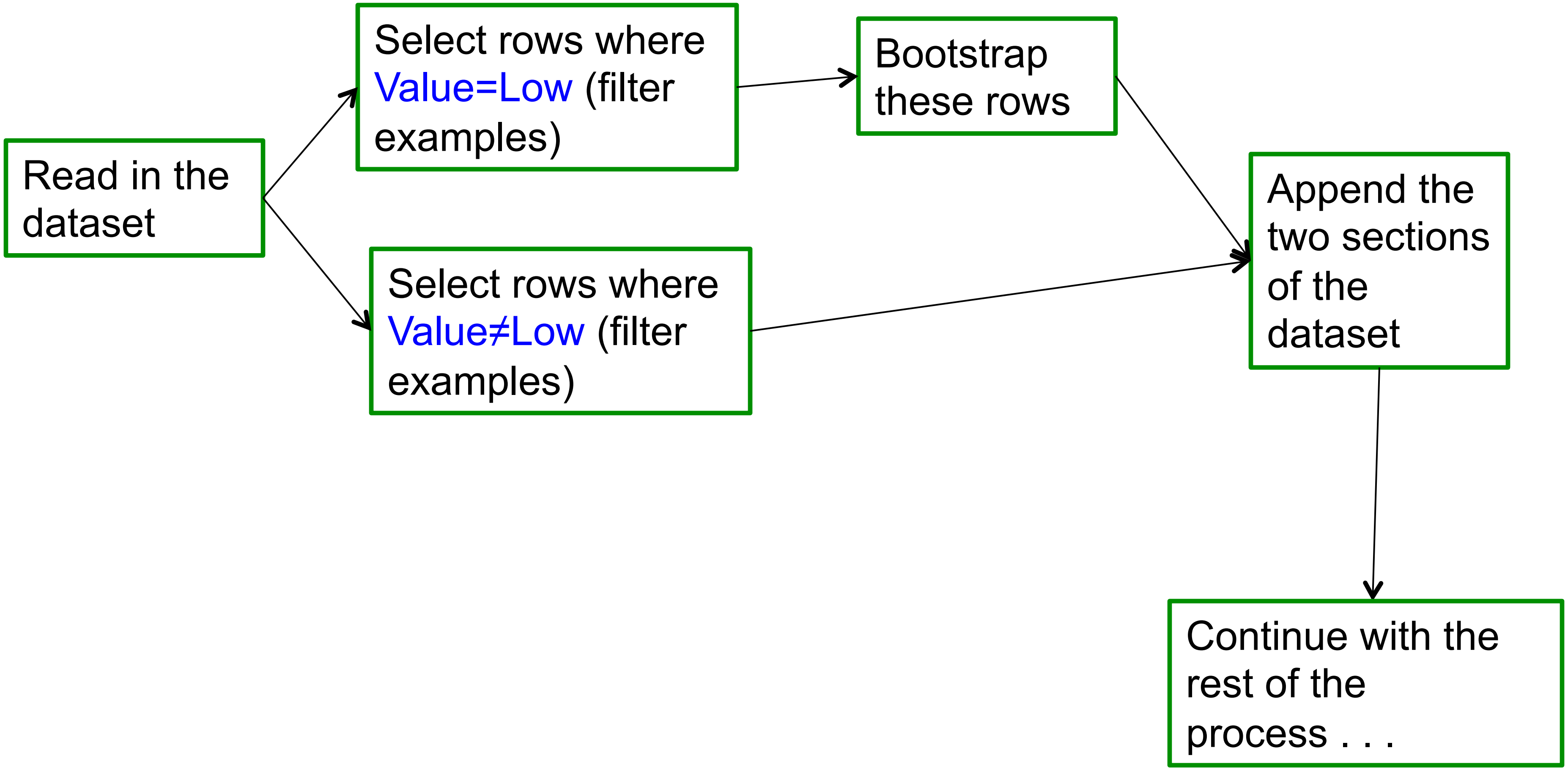
Class imbalance

- If one identifiable segment of the data constitutes only a small percentage of the data set or sample, then many mining algorithms will simply ignore that segment.
  - For example, suppose you want to analyse the factors that cause high blood pressure, or low blood pressure. Your data set might have a categorical variable called ‘blood pressure’, with three possible values: *high*, *normal* and *low*. Suppose only 5% of the data set represented people with low blood pressure. Many mining algorithms would consider this group insignificant, since they make up such a small percentage of the population.
- To force this group to be considered, the rows representing this group can be bootstrapped. For example the second row below representing low blood pressure could be duplicated seven or more times to bring it in line with other categories. This does not alter any patterns in the data set.



Value	Proportion	%	Count
High		60.0	12
Low		5.0	1
Normal		35.0	7

Rapidminer workflow



Read in the dataset

Select rows where Value=Low (filter examples)

Bootstrap these rows

Select rows where Value≠Low (filter examples)

Append the two sections of the dataset

Continue with the rest of the process . . .

7. Reducing dimensionality

Reducing the number of attributes
in a dataset

Dimensionality reduction

- Datasets frequently have too much data making it harder for mining algorithms to find patterns.
- In addition, many data **preparation** techniques result in **additional** columns being **added** to the data set, making the problem worse.
- Reducing the dimensionality of a data set has a number of benefits:
  1. Eliminates irrelevant features, and reduces the amount of noise in the data set.
  2. Patterns are easier to find in a denser space.
  3. Processing time and memory requirements are reduced.
  4. Results are easier to interpret as they involve less variables.
  5. More complex mining algorithms can be used.

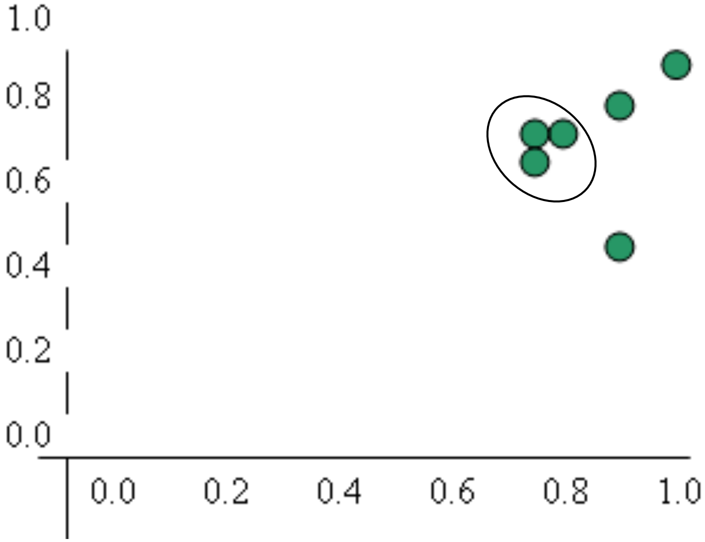
Problems with high dimensionality . . .

- The role of a mining tool is to spot **relationships** between variables, i.e. groups of rows that are similar to each other and so represent a pattern.

- Take the following table of height and weight characteristics of a football team:

Height	Weight
1.00	1.00
0.75	0.87
0.69	0.76
0.69	0.75
0.60	0.69
0.49	0.84

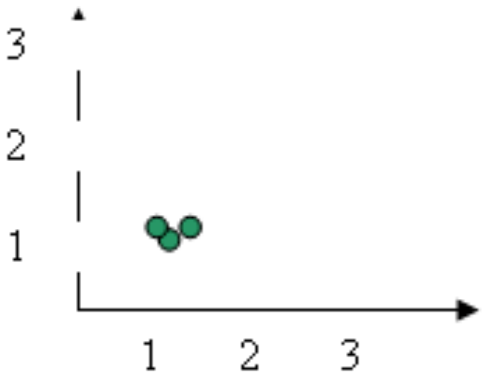
A 2-dimensional plot for these points is as follows:



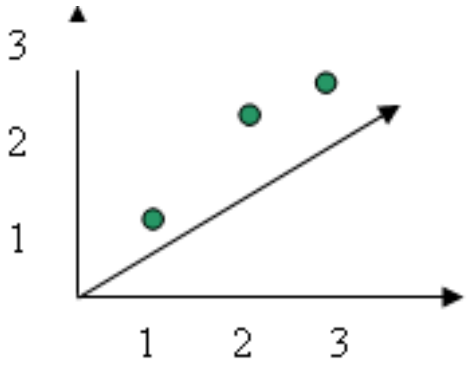
Three points appear to be similar, forming a cluster . . .

When viewed in 3-D (i.e. adding a thir dimension), the points are not so close together:

The three points in the cluster below are: (1, 1), (1.2, 0.9), (1.3, 1.1)



Adding a third dimension gives: (1, 1, 6); (1.2, 0.9, 0), (1.3, 1.1, 4)



Curse of dimensionality . . .

- When viewed in “3-D”, the points are no longer close together. Every dimension added to a dataset disperses points (rows) further apart, making it harder to find rows that are similar and so form a pattern.

Dimensionality reduction

- There are a variety of ways to reduce the number of dimnesions (variables) in the data set:

  1. Remove attributes that do not contribute greatly to the overall objective of the mining project.
  2. Remove attributes that hold the same information content
  3. Principal component analysis and related techniques.

Options 1 and 2 have already been covered under Data Selection. The following slides will focus on option 3.

Dimensionality reduction

Example: Take the following three variables: **Height, weight and girth.**

If you know any two, the third can be deduced. Therefore while three variables are needed to capture the values, only two variables are needed to capture the information content.

So the question arises: what is the minimum number of attributes needed to capture the information content of the dataset?

This question may be answered to some extent based on your domain knowldege, but there are mathematical techniques to do this . . .

Dimensionality reduction

Principal Component Analysis

- Principal Component Analysis is a compression technique for identifying the principle components, or sources of variation, in a dataset.
- Note: It’s computationally expensive, and the resulting variables are combinations of the original variables in the dataset, and so cannot easily be interpreted.

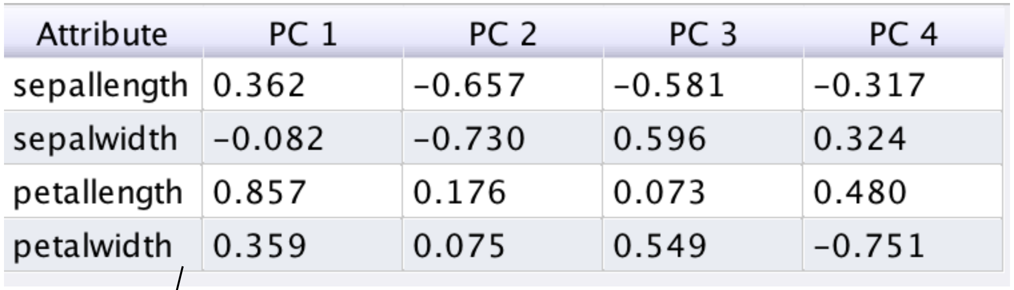
You can specify:

- how many variables to reduce the dataset to
- or how much of the variance (information) from the dataset include

and PCA will generate a dataset with less attributes but covering a significant portion of the information from the original dataset.

The output from PCA . . .

Eigen Vectors:

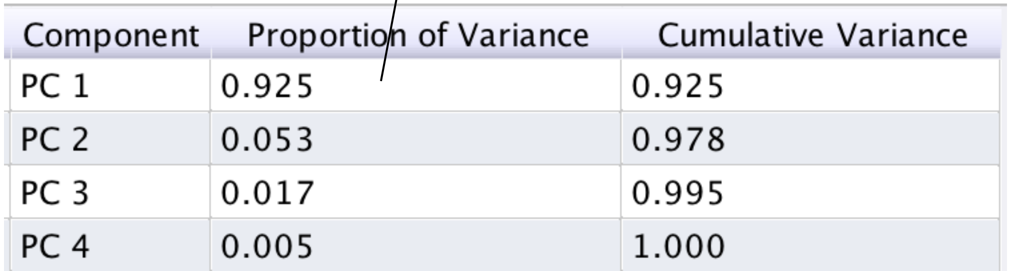


Attribute	PC 1	PC 2	PC 3	PC 4
sepallength	0.362	−0.657	−0.581	−0.317
sepalwidth	−0.082	−0.730	0.596	0.324
petallength	0.857	0.176	0.073	0.480
petalwidth	0.359	0.075	0.549	−0.751

Which attributes each Principal Component is calculated from.

How much of the information from the dataset is captured by each principal component

Eigen Values:



Component	Proportion of Variance	Cumulative Variance
PC 1	0.925	0.925
PC 2	0.053	0.978
PC 3	0.017	0.995
PC 4	0.005	1.000

Example of PCA

Have a look at the RapidMiner process under samples/ 04_attributes/03_PrincipalComponents

- The process reduces the IRIS dataset down to two attributes, which between them hold 95% of the variability of the original dataset.

Also take a look at lect5-PCA process on moodle which adds two validation blocks to the process above, one which models the original dataset, and the other which models the principal components.

- Is two principal components sufficient or would three be needed to maintain the accuracy of the results?

8. Attribute Construction

- Attribute construction is generating new variables from calculations made on existing variables in the data set, and is done to expose additional information to a mining algorithm. Like data transformation, attribute construction is project specific, and domain specific.
- *Example 1: Consider a data set containing information about historical artefacts, and suppose these artefacts were made from a number of materials (e.g. wood, clay, bronze, etc.) Density, calculated from mass/volume, would be more useful in getting an accurate classification of the artefact.*
- *Example 2: Consider a web log which includes IP address and a timestamp. It would be useful to add a session ID which is essentially a count that is incremented when the IP address changes or there is a timelag of more the xx minutes between one request and the next?*

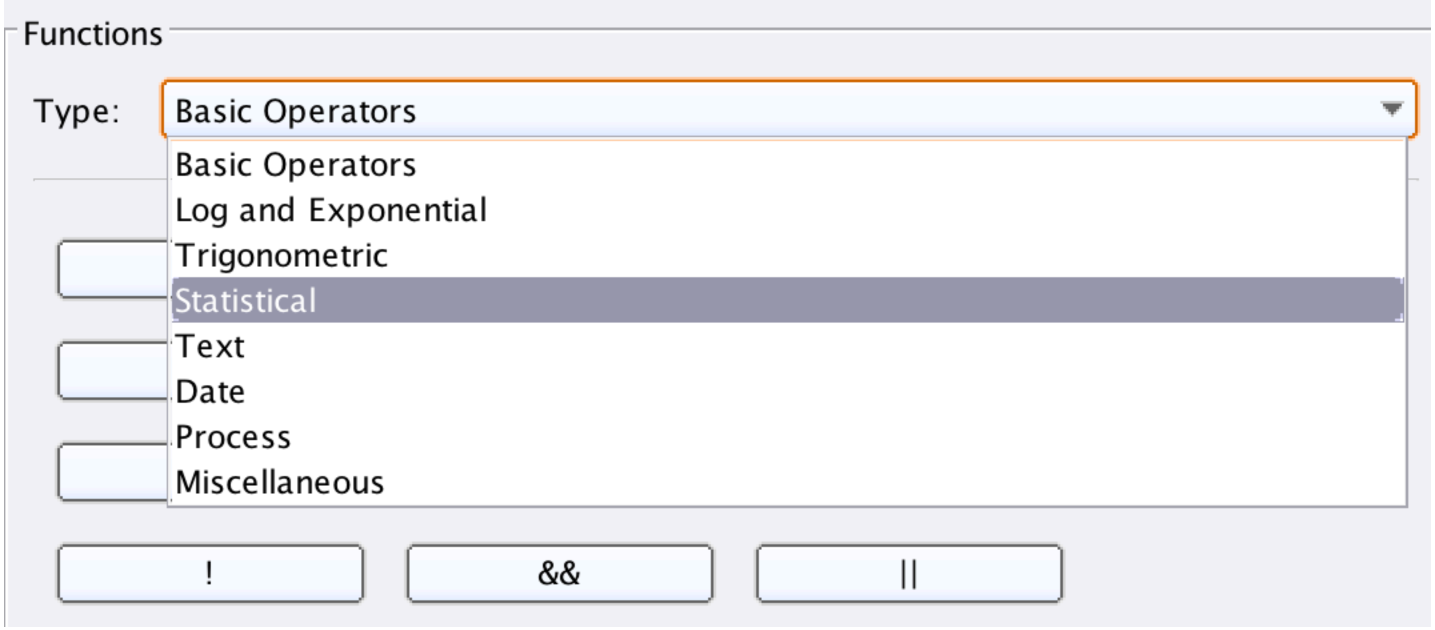
8 Attribute construction

- Exercise: Suggest attributes you could construct from the following:

Income	Loan	Number in family	Weekly shopping bill
20000	5000	1	100
50000	5000	4	200
30000	20000	2	150

8 Attribute construction

- Rapidminer miner has a large range of functions than can be applied to attributes to generate new data, available under the operator: Generate Attributes.
- Take a look at samples/02_preprocessing/ 12_UserDefinedFeatureGeneration
- Click on the Edit List parameter for new attribute definitions.



Functions

Type: Basic Operators

9. Type Conversion

Some learners are limited in the data types they can process, so it may be necessary to convert form one data type to another as follows . ..

Conversions for Numeric Data

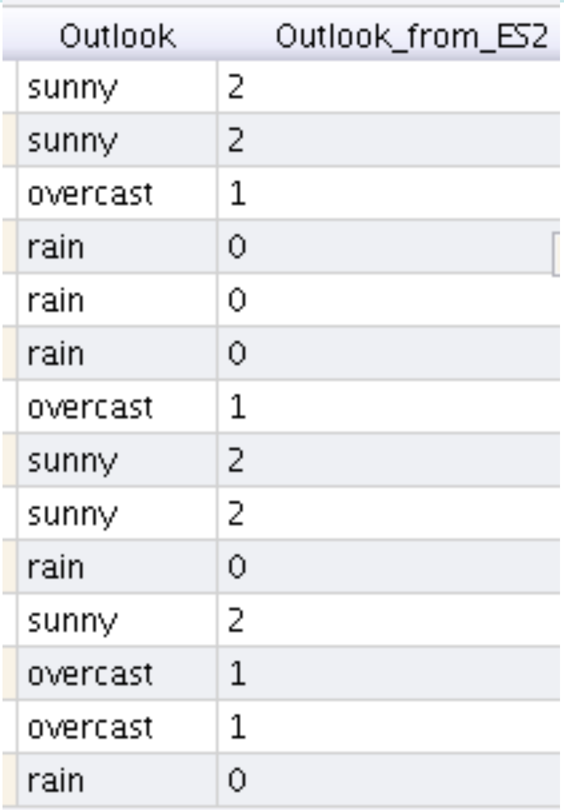
Numeric Atr.	Binominal (True = in the range [15-20])	Polynominal
10	False	10
12	False	12
15	True	15
18	True	18
20	True	20
25	False	25

Convert to Binominal:
All values with a certain range are set to True, other values are set to False

Convert to Polynominal:
Changes the datatype. e.g. 3.5 change to character string '3.5'

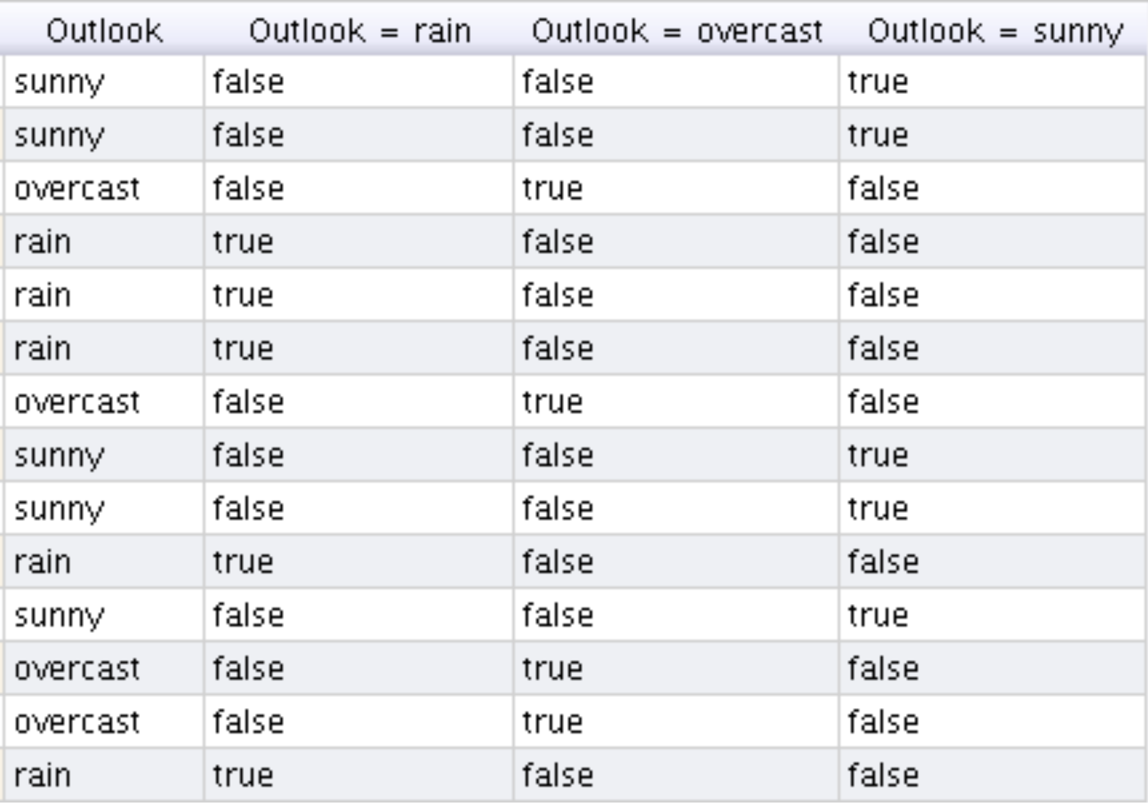
Conversions for nominal data

Convert to Numeric:
Each distinct nominal value is allocated a number sequentially starting at 0



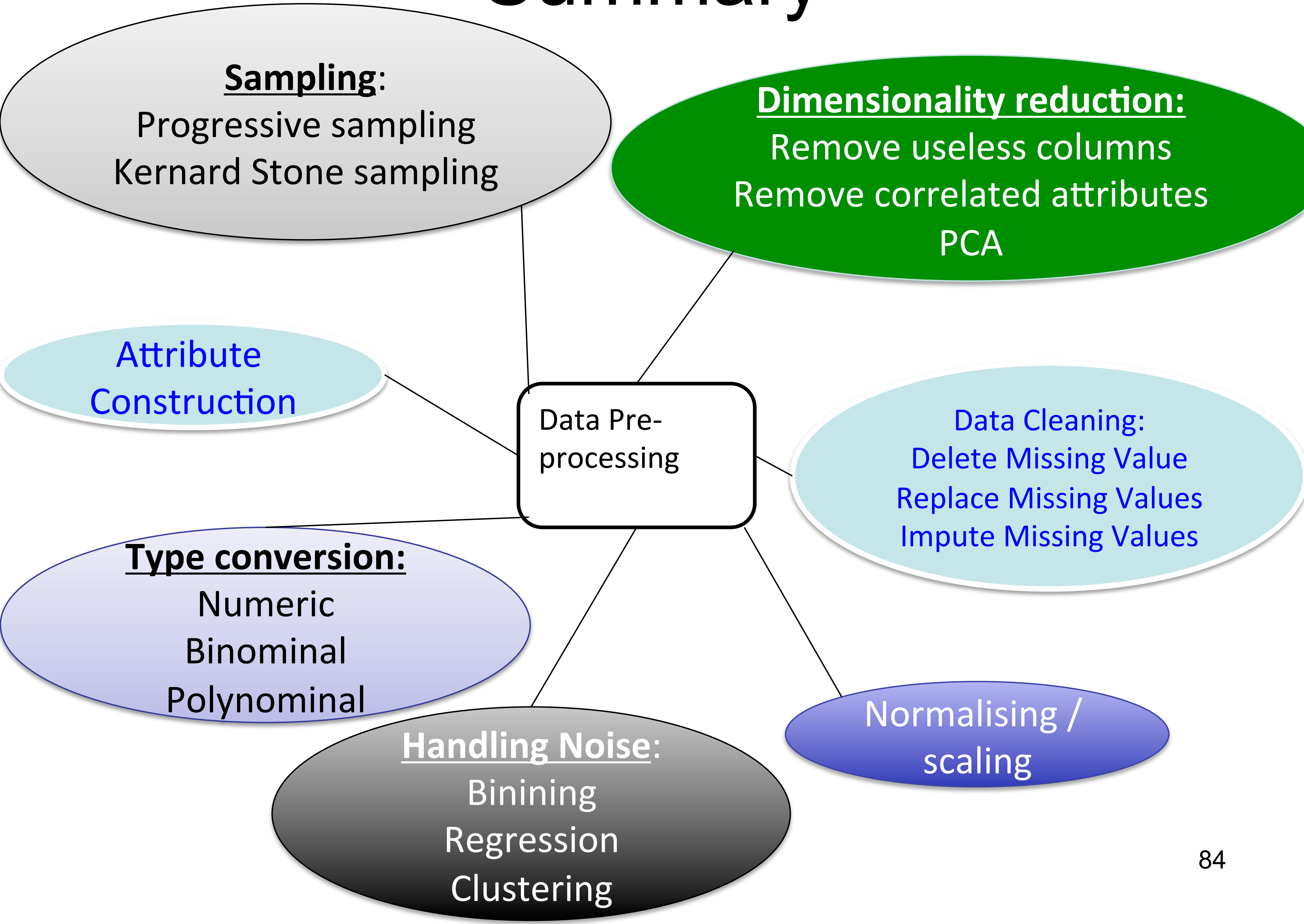
Outlook	Outlook_from_ES2
sunny	2
sunny	2
overcast	1
rain	0
rain	0
rain	0
overcast	1
sunny	2
sunny	2
rain	0
sunny	2
overcast	1
overcast	1
rain	0

Convert to Binominal:
Each distinct nominal value becomes a new attribute



Outlook	Outlook = rain	Outlook = overcast	Outlook = sunny
sunny	false	false	true
sunny	false	false	true
overcast	false	true	false
rain	true	false	false
rain	true	false	false
rain	true	false	false
overcast	false	true	false
sunny	false	false	true
sunny	false	false	true
rain	true	false	false
sunny	false	false	true
overcast	false	true	false
overcast	false	true	false
rain	true	false	false

Summary



Data Pre-processing

Sampling:
Progressive sampling
Kernard Stone sampling

Dimensionality reduction:
Remove useless columns
Remove correlated attributes
PCA

Attribute Construction

Data Cleaning:
Delete Missing Value
Replace Missing Values
Impute Missing Values

Type conversion:
Numeric
Binominal
Polynominal

Handling Noise:
Binining
Regression
Clustering

Normalising / scaling