Data Reshaping

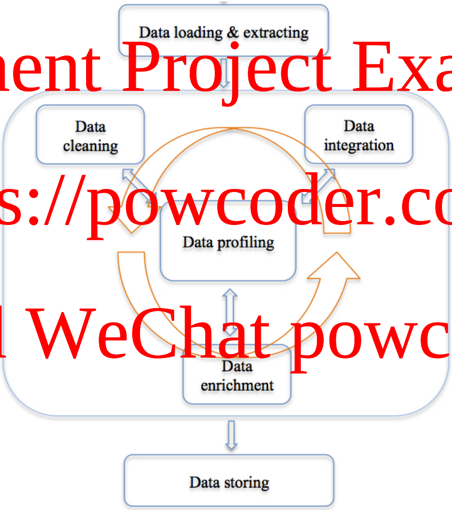
Faculty of Information Technology, Monash University, Australia

FIT5196 week 09

Outline

1. Data Transformation
   - Data Normalisation/Scaling
   - Transformation by generating new features
   - Nominal to Numeric Transformation
2. Data Discretisation
3. Feature Engineering & Data Sampling
4. Summary

Data Wrangling Process



Outline

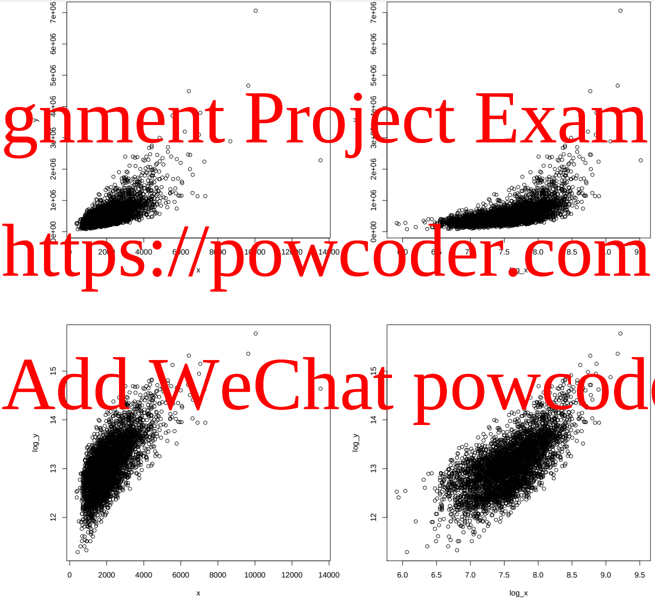
1 Data Transformation
- Data Normalisation/Scaling
- Transformation by generating new features
- Nominal to Numeric Transformation

2 Data Discretisation

3 Feature Engineering & Data Sampling

4 Summary

Data Transformation



Data Transformation

- Why: Raw attributes are usually not good enough to obtain accurate predictive model.
  - k-nearest neighbours (KNN) with an Euclidean distance measure if want all features to contribute equally

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_i (p_i - q_i)^2}$$

  - logistic regression, SVMs, perceptrons, neural networks etc. if you are using gradient descent/ascent-based optimisation, otherwise some weights will update much faster than others

$$\Delta w_j = -\eta \frac{\partial J}{\partial w_j} = \eta \sum_i (t^{(i)} - o^{(i)}) x_j^{(i)}$$

    so that $w_j := w_j + \Delta w_j$
  - linear discriminant analysis, principal component analysis, kernel principal component analysis since you want to find directions of maximising the variance (under the constraints that those directions/eigenvectors/principal components are orthogonal); you want to have features on the same scale since you'd emphasise variables on "larger measurement scales" more.

Data Transformation

- Data transformation
  - A series of manipulation steps to transform the original attributes or to generate new attributes with better properties that will help the predictive power of the model.
  - To achieve properties that enhance the modelling and analysis (linearity, statistical or visual interpretability).
  - Methods
    - Normalisation/Scaling methods
    - Transformation by generating new features (i.e., variables or attributes)

Outline

1 Data Transformation
- Data Normalisation/Scaling
- Transformation by generating new features
- Nominal to Numeric Transformation

2 Data Discretisation

3 Feature Engineering & Data Sampling

4 Summary

Data Transformation — Normalisation

There are two types of data normalisation:

- Standardisation (z-score normalisation): where the focus is on shifting the distribution of data to have mean of 0 and standard deviation of 1.
- Scaling: where the focus is on rescaling data value range to a specific interval.
  - Min-Max normalisation
  - Decimal scaling

Data Normalisation — Standardisation

Z-score Normalisation

- Rescale the features (or variables) so that they will have the properties of a standard normal distribution with

$$\mu = 0 \;\&\; \sigma = 1.0$$

- How?

$$x' = \frac{x - \mu}{\sigma}$$

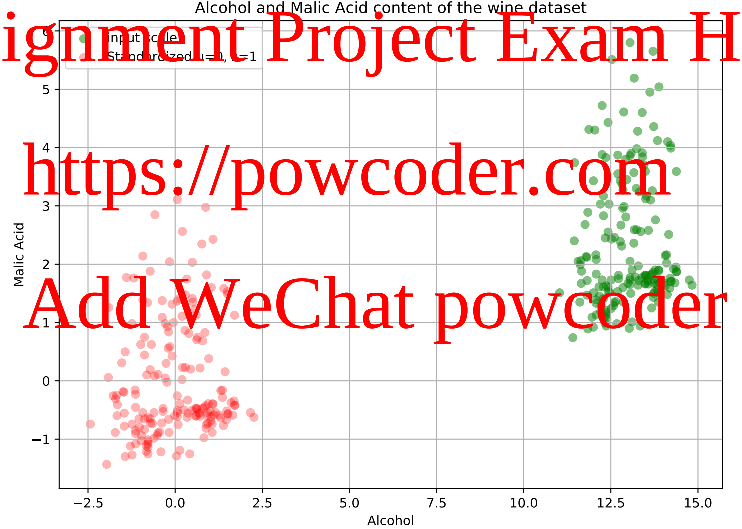
where

$$\mu = \frac{1}{n}\sum_i x_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_i (x_i - \mu)^2}$$

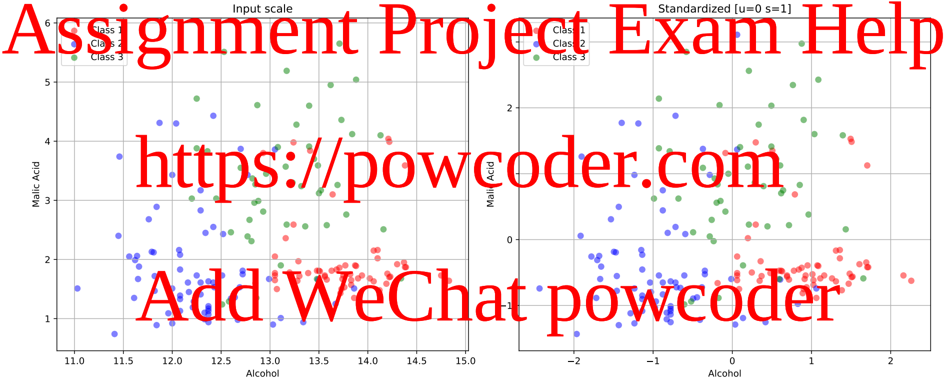
Data Normalisation — Standardisation

Z-score Normalisation



Data Normalisation — Standardisation

Z-score Normalisation



Data Normalisation — Min-Max Scaling

Min-Max Scaling

- Rescale the features (or variables) that their values are in a specific range $[X'_{min}, X'_{max}]$.
- How?

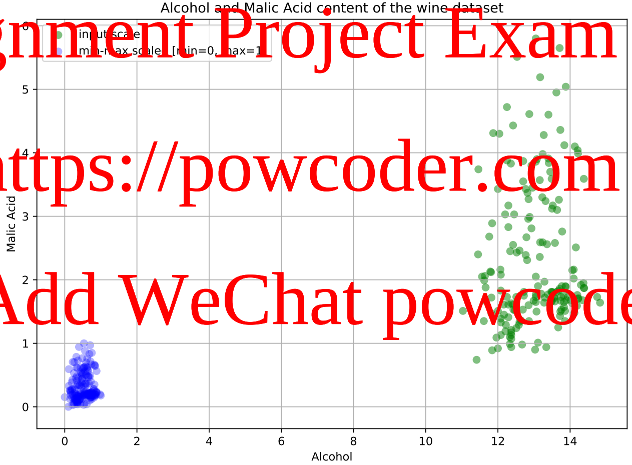
$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} (X'_{max} - X'_{min}) + X'_{min}$$

If the fixed range is [0,1]

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Data Normalisation — Min-Max Scaling

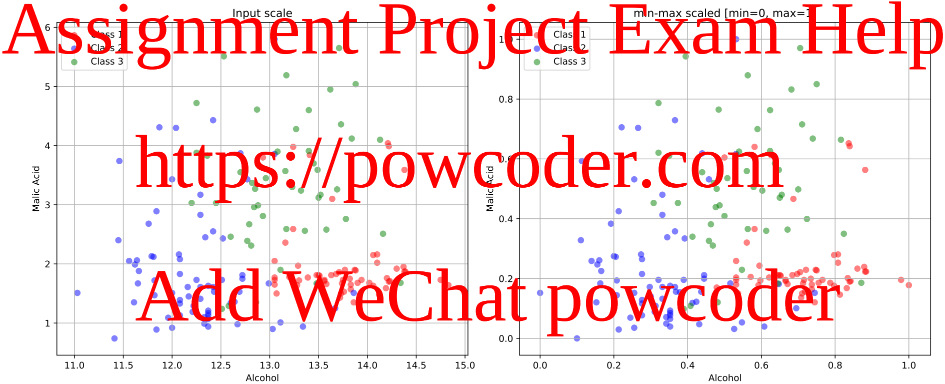
Min-Max Scaling



We will end up with smaller standard deviations, which can suppress the effect of outliers

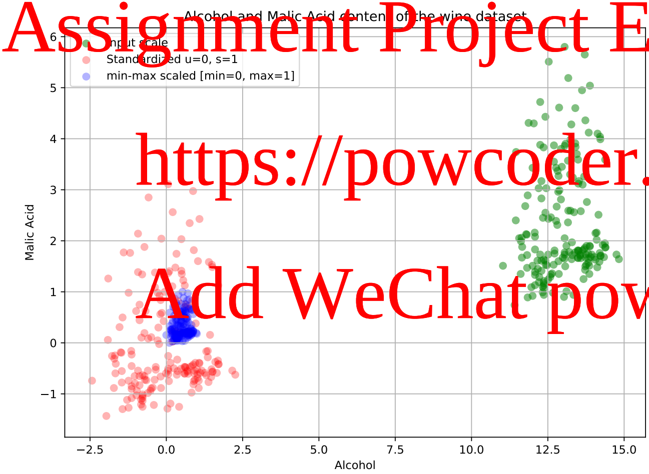
Data Normalisation — Min-Max Scaling

Min-Max Scaling



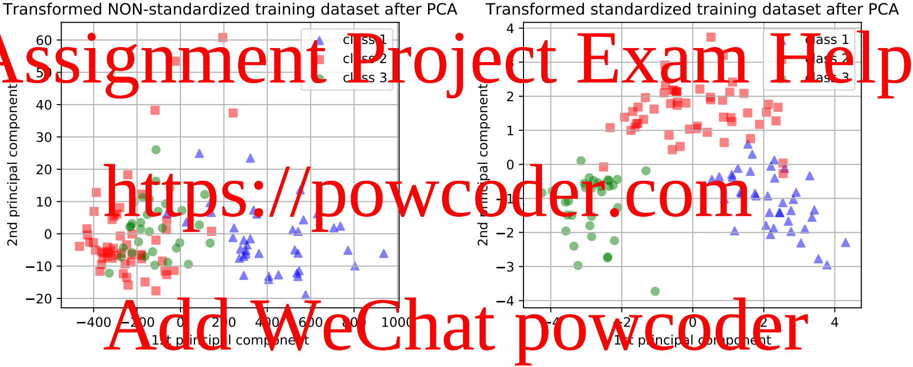
Data Normalisation — Standardisation vs Min-Max

“Standardisation or Min-Max scaling?”: depends on the application



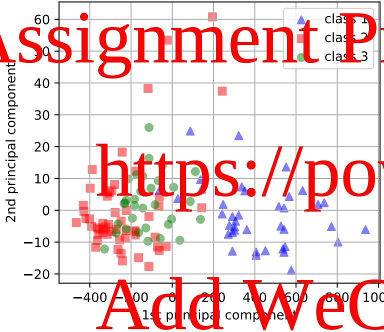
- PCA: standardisation
- Image processing: pixel intensities have to be normalised to fit within a certain range (i.e., 0 to 255 for the RGB colour range)
- ANN: data that on a 0-1 scale

Data Normalisation — Standardisation vs Min-Max

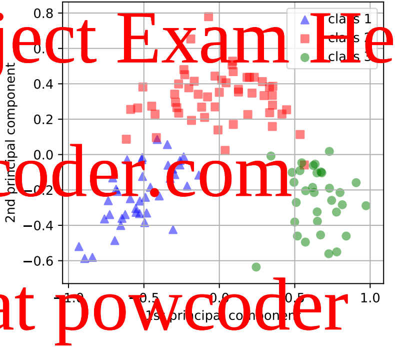


Data Normalisation — Standardisation vs Min-Max

Transformed NON-standardized training dataset after PCA



Transformed min_max scaled training dataset after PCA



Data Normalisation — Decimal Scaling

- Shift the decimal place of a numeric value such that the maximum absolute value will be always less than 1
- How:

$$x' = \frac{x}{10^c}$$

  where c is the smallest integer such that $max(|x'|) < 1$.

- Example:
  - $-500 \leq x \leq 45 \Rightarrow -0.500 \leq x \leq 0.045$
  - How to convert?

Data Normalisation — Decimal Scaling

- Shift the decimal place of a numeric value such that the maximum absolute value will be always less than 1

- How:

$$x' = \frac{x}{10^c}$$

where c is the smallest integer such that $max(|x'|) < 1$.

- Example:
  - $-500 \leq x \leq 45 \Rightarrow -0.500 \leq x \leq 0.045$
  - How to convert?
    - $x_{max} = max(abs(x)) = 500$
    - $c = ceil(log_{10}(x_{max})) = 3.0$
    - $x/ = 10.0^{3.0} = x/1000.0$

Outline

1. Data Transformation
   - Data Normalisation/Scaling
   - Transformation by generating new features
   - Nominal to Numeric Transformation
2. Data Discretisation
3. Feature Engineering & Data Sampling
4. Summary

Data Transformation

Data Transformation is a process of re-expressing data in a form that is more suitable for analysis.

- Reasons for data transformation:
    - Fix skewness in data
    - Enhance data visualisation
    - Better interpretability
    - Improve the compatibility of data with assumptions underlying a modelling process

- Methods: different mathematical formulas from statistical analysis
    - linear transformation
    - log transformation
    - Power transformation
    - Box-Cox Transformation
    - others: Quadratic transformation, (non-)polynomial approximation of transformation, rank transformation

Data Transformation

Linear Transformation

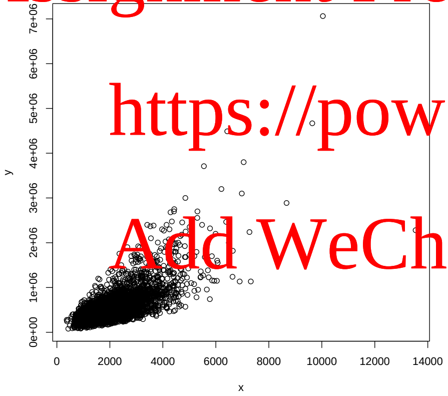
- Linear transformation preserves the linear relationship between the features.
- Aggregate the information contained in various features.
- Linear transformation function: Given a subset of the complete set of attributes, $X_1, X_2, \ldots, X_m$,

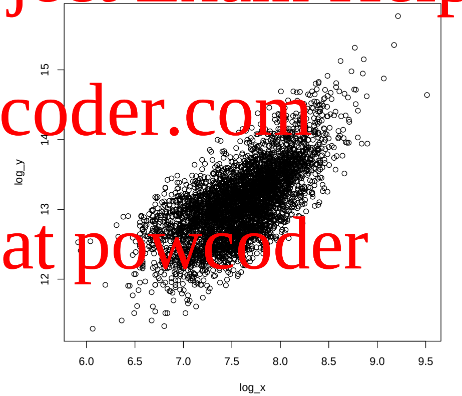
$$X_{agg} = w_0 + \sum_{i=1}^{m} w_i X_i$$

- Examples:
  - Celsius to Fahrenheit
  - Miles to Kilometers
  - Inches to Centimeters

Data Transformation

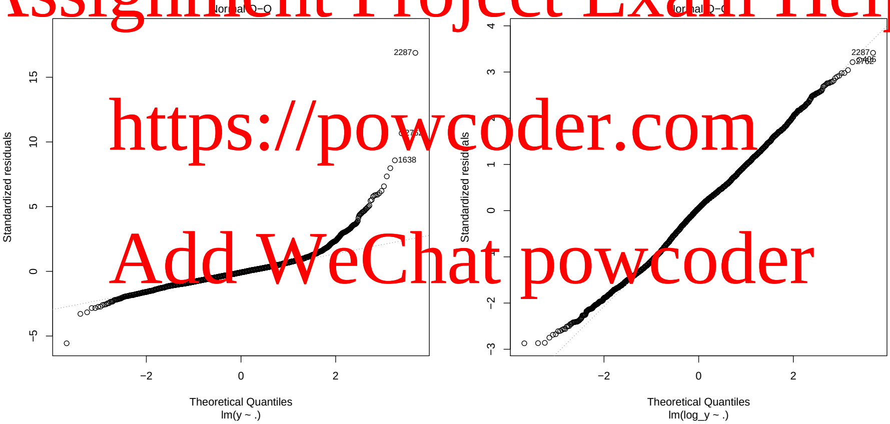
Log transformation makes highly skewed distributions less skewed





Data Transformation

Log transformation makes highly skewed distributions less skewed

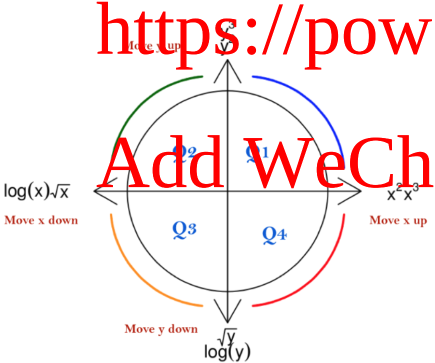


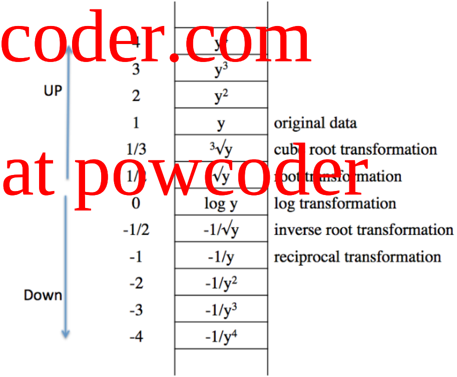
Data Transformation

Power Transformation

- *Tukey and Mosteller's Bulging Rule*: The idea is that it might be interesting to transform X and Y at the same time, using some power functions

$$Y_i^q = \beta_0 + \beta_1 X_i^p + \eta_i$$

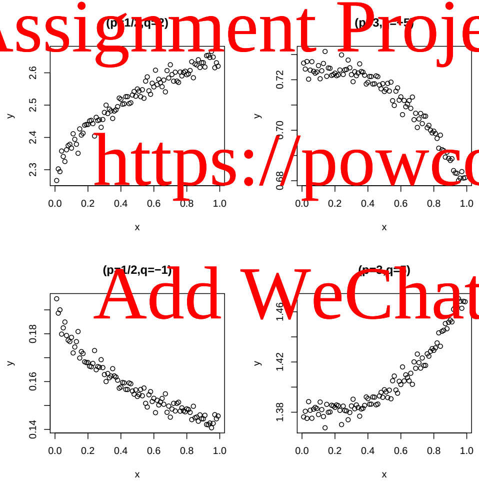




4	y^4	
3	y^3	
2	y^2	
1	y	original data
1/3	$\sqrt[3]{y}$	cube root transformation
1/2	$\sqrt{y}$	root transformation
0	log y	log transformation
-1/2	$-1/\sqrt{y}$	inverse root transformation
-1	-1/y	reciprocal transformation
-2	$-1/y^2$	
-3	$-1/y^3$	
-4	$-1/y^4$	

Data Transformation

Power Transformation

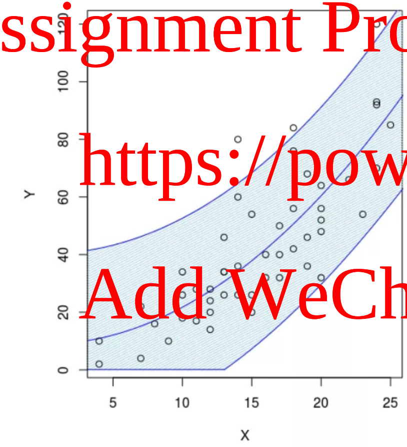


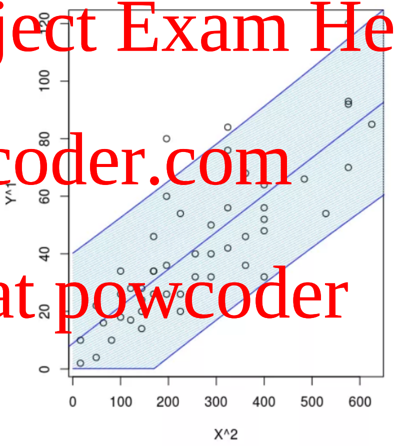
$$Y_i^q = \beta_0 + \beta_1 X_i^p + \eta_i$$

More information can be found https://www.r-bloggers.com/tukey-and-mostellers-bulging-rule-and-ladder-of-powers/

Data Transformation

Power Transformation

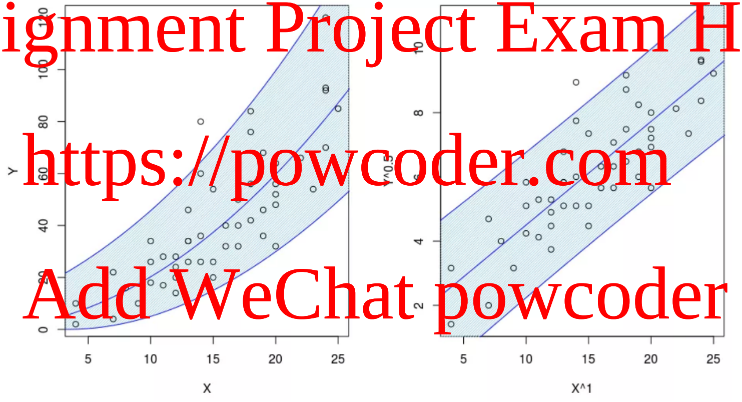




Data Transformation

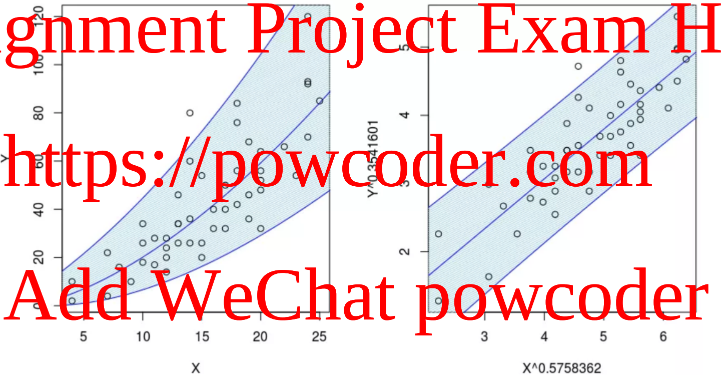
MONASH University

Power Transformation



Data Transformation

Power Transformation



- Seek optimal transformations: learnt *p* and *q* with L-BFGS

Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

$$y = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$

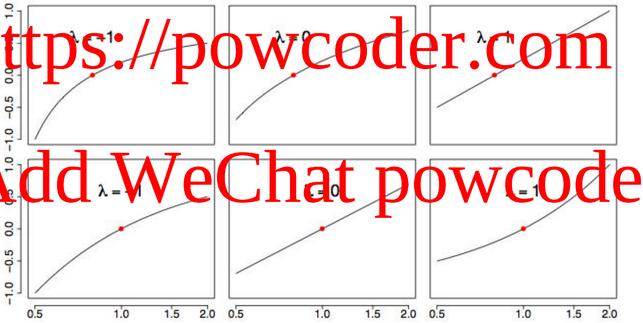
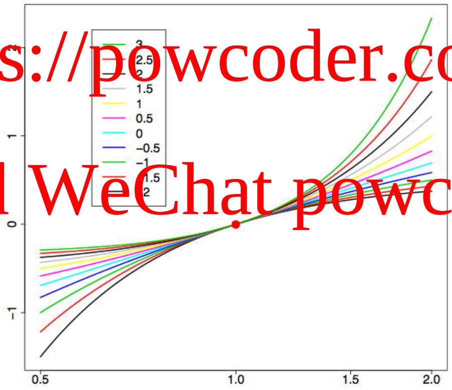


Figure: Examples of the Box-Cox transformation x'_λ versus x for $\lambda = -1, 0, 1$. In the second row, x'_λ is plotted against $log(x)$. The red point is at $(1, 0)$.

Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

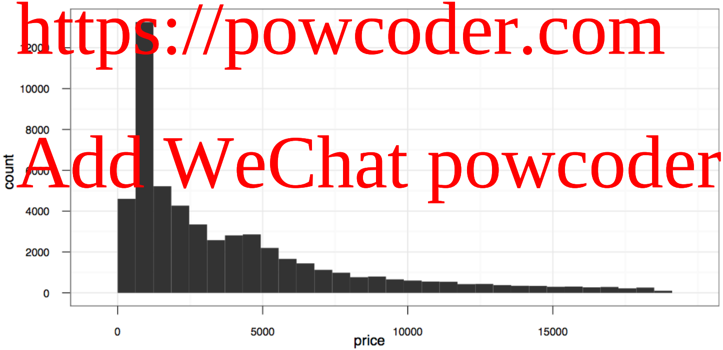
$$y = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$



Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

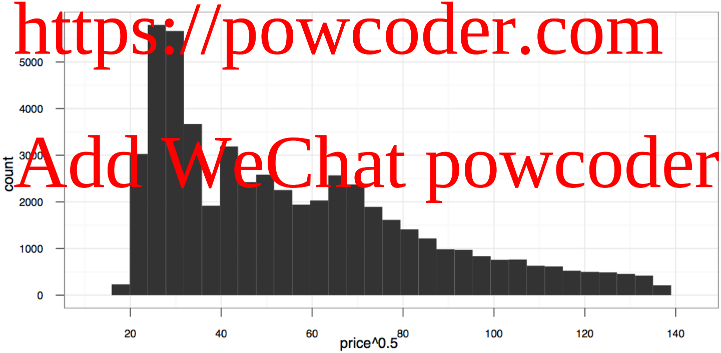
$$y = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$



Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

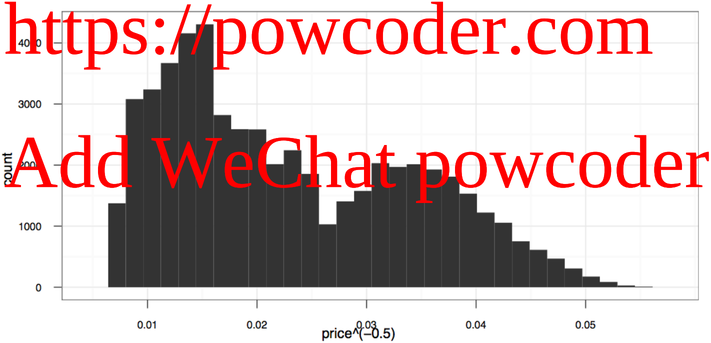
$$y = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$



Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

$$y = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$



Data Transformation

The Box-Cox Transformation: transforms a continuous variable into an almost normal distribution.

- With negative values in the attributes

$$y = \begin{cases} \frac{(x+c)^{\lambda-1}}{g\lambda} & \text{if } \lambda \neq 0 \\ \frac{\log(x+c)}{g} & \text{if } \lambda = 0 \end{cases}$$

where

- A parameter c: offset the negative values
- g: scale the resulting values, often considered as the geometric mean of the data.
- λ: greedily search λ so that the resulting attribute is as close as possible to the normal distribution.

Outline

1. Data Transformation
   - Data Normalisation/Scaling
   - Transformation by generating new features
   - Nominal to Numeric Transformation

2. Data Discretisation

3. Feature Engineering & Data Sampling

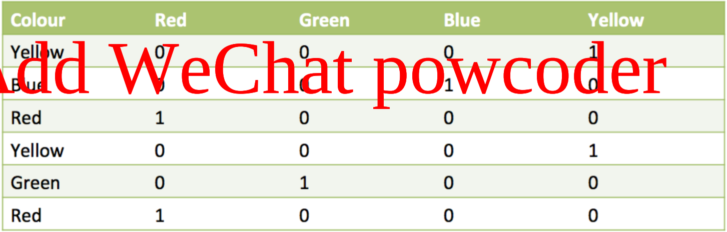
4. Summary

Nominal to Numeric Transformation

- Why?
  - Many machine learning algorithms only accept numeric value, while in many applications we have nominal attributes.
- How?
  - Integer substitution: map each nominal value in the domain to numeric value
  - Example: assume we have a color attribute with Red, Green, Blue and Yellow value
    - Red ⇒ 1
    - Green ⇒ 2
    - Blue ⇒ 3
    - Yellow ⇒ 4
  - What is the problem?
    - Implies a sort of ranking that doesn?t actually exists in the original data.
    - The outcome of the mining algorithms would be sensitive to the numeric values we choose to use.

Nominal to Numeric Transformation

- Why?
  - Many machine learning algorithms only accept numeric value, while in many applications we have nominal attributes.
- How?
  - Integer substitution: map each nominal value in the domain to numeric value
  - Example: assume we have a color attribute with Red, Green, Blue and Yellow value
  - One-hot encoding



Colour	Red	Green	Blue	Yellow
Yellow	0	0	0	1
Blue	0	0	1	0
Red	1	0	0	0
Yellow	0	0	0	1
Green	0	1	0	0
Red	1	0	0	0

Outline

1. Data Transformation
2. Data Discretisation
3. Feature Engineering & Data Sampling
4. Summary

Data Discretisation

- The process of converting or partitioning continuous variables to discretised or nominal variables.
    - Find concise data representations as categories which are adequate for the learning task retaining as much information in the original continuous attribute as possible
    - Effects of discretisation
        - Smooth data
        - Reduce noisy
        - Reduce data size
        - Enable specific methods using nominal data

Data Discretisation

- Methods
  - Binning
  - Entropy discretisation
  - Concept hierarchy

Data Discretisation — Binning

- An unsupervised algorithm (doesn't care about the dependent variable) that splits ordered data into predefined number of bins

- Two approaches
  - Equal-width binning
    - Given a range of values, $[x_{min}, x_{max}]$, we divide the value range into intervals with approximately same width, w

$$w = \frac{x_{max} - x_{min}}{n}$$

where n is the number of bins. Or you can specify the value of w

  - Equal-depth binning
    - Divides the range into n intervals, each containing approximately same number of samples.

- Binning with
  - mean value
  - median values
  - bin boundaries

Data Discretisation — Binning

- Task: discretise $\{34, 64, 88, 55, 94, 59, 10, 25, 44, 48, 69, 15\}$
  - sort the values in ascending order

    $$\{10, 15, 25, 34, 44, 48, 55, 59, 64, 69, 88, 94\}$$

  - Equal-width binning with $n = 4$

    $$\{10, 15, 25\}, \{34, 44, 48\}, \{55, 59, 64, 69\}, \{88, 94\}$$

    - mean value

      $$\{16.6, 16.6, 16.6\}, \{42, 42, 42\}, \{61.75, 61.75, 61.75, 61.75\}, \{91, 91\}$$

    - median value

      $$\{15, 15, 15\}, \{44, 44, 44\}, \{61.5, 61.5, 61.5, 61.5\}, \{91, 91\}$$

    - boundaries

      $$\{10, 10, 25\}, \{34, 48, 48\}, \{55, 55, 69, 69\}, \{88, 94\}$$

Data Discretisation — Binning

- Task: discretise $\{34, 64, 88, 55, 94, 59, 10, 25, 44, 48, 69, 15\}$
  - sort the values in ascending order

$$\{10, 15, 25, 34, 44, 48, 55, 59, 64, 69, 88, 94\}$$

  - Equal-depth binning with $n = 4$

$$\{10, 15, 25\}, \{34, 44, 48\}, \{55, 59, 64\}, \{69, 88, 94\}$$

    - mean value

$$\{16.6, 16.6, 16.6\}, \{42, 42, 42\}, \{59.3, 59.3, 59.3\}, \{83.6, 83.6, 83.6\}$$

    - median value

$$\{15, 15, 15\}, \{44, 44, 44\}, \{59, 59, 59\}, \{88, 88, 88\}$$

    - boundaries

$$\{10, 10, 25\}, \{34, 48, 48\}, \{55, 55, 64\}, \{69, 94, 94\}$$

Data Discretisation — Binning

Advantage/disadvantage of each method:

- Equal-width binning
  - Is simple but sensitive to outliers
  - Not well handles skewed data
- Equal-depth binning
  - Scales well by keeping the distribution of the data

Data Discretisation — Entropy Discretisation

- Entropy

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b(p(x_i))$$

  - Coin toss: $p(head) = p(tail) = 1/2$

$$H(X) = -p(head)\log_2(p(head)) - p(tail)\log_2(p(tail)) = -2 \times \frac{1}{2}\log_2(1/2) = 1$$

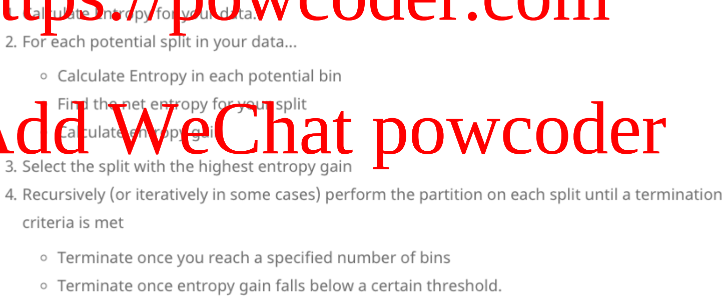
  - Coin toss: $p(head) = 0.7$ and $p(tail) = 0.3$

$$H(X) = -0.7\log_2(0.7) - 0.3\log_2(0.3) = 0.881 < 1$$

- Entropy discretisation: a method takes into account the class labels in discretisation.

Data Discretisation — Entropy Discretisation

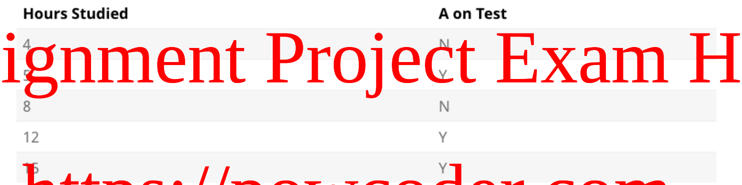
- Entropy discretisation: a method takes into account the class labels in discretisation.
  - Ideas
    - Data should be split into intervals that maximise the information, measured by Entropy,
    - Partitioning should not be too fine-grained, to avoid over-fitting.
  - Algorithm



1. Calculate Entropy for your data.
2. For each potential split in your data…
   - Calculate Entropy in each potential bin
   - Find the net entropy for your split
   - Calculate entropy gain
3. Select the split with the highest entropy gain
4. Recursively (or iteratively in some cases) perform the partition on each split until a termination criteria is met
   - Terminate once you reach a specified number of bins
   - Terminate once entropy gain falls below a certain threshold.

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

Data Discretisation — Entropy Discretisation



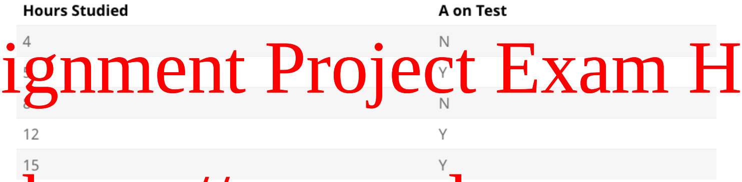
Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Entropy of the data:

$$H(X) = -\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5}) = 0.529 + 0.442 = 0.971$$

Data Discretisation — Entropy Discretisation



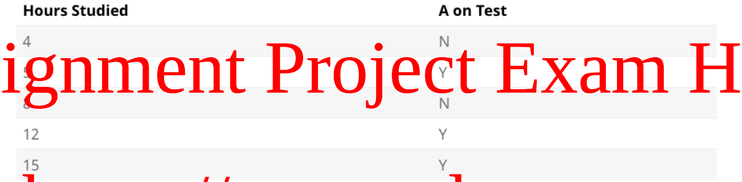
Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Split at 4.5

$$
\begin{aligned}
H(X <= 4.5) &= -\frac{1}{1}\log_2(1) - \frac{0}{1}\log_2(0) = 0 + 0 = 0 \\
H(X > 4.5) &= -\frac{3}{4}\log_2(\frac{3}{4}) - \frac{1}{4}\log_2(\frac{1}{4}) = 0.311 + 0.5 = 0.811 \\
H(X_{new}) &= H(X <= 4.5) + H(X > 4.5) = \frac{1}{5}0 + \frac{4}{5}0.811 = 0.6488 \\
G(X_{new}) &= 0.971 - 0.6488 = 0.322
\end{aligned}
$$

Data Discretisation — Entropy Discretisation



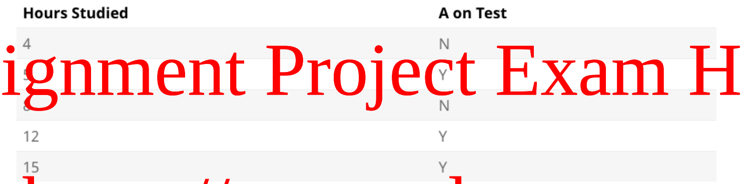
Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Split at 6.5

$$
\begin{aligned}
H(X <= 6.5) &= -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) = 1 \\
H(X > 6.5) &= -\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3}) = 0.918 \\
H(X_{new}) &= H(X <= 6.5) + H(X > 6.5) = \frac{2}{5}1 + \frac{3}{5}0.917 = 0.951 \\
G(X_{new}) &= 0.971 - 0.951 = 0.02
\end{aligned}
$$

Data Discretisation — Entropy Discretisation



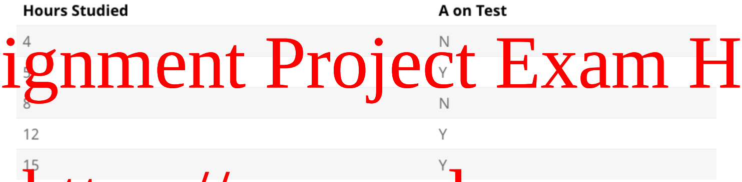
Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Split at 10

$$
\begin{aligned}
H(X <= 10) &= -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) = 0.918 \\
H(X > 10) &= -\frac{2}{2}\log_2(\frac{2}{2}) - \frac{0}{2}\log_2(\frac{0}{2}) = 0 \\
H(X_{new}) &= H(X <= 10) + H(X > 10) = \frac{3}{5}0.917 + \frac{2}{5}0 = 0.551 \\
G(X_{new}) &= 0.971 - 0.551 = 0.42
\end{aligned}
$$

Data Discretisation — Entropy Discretisation



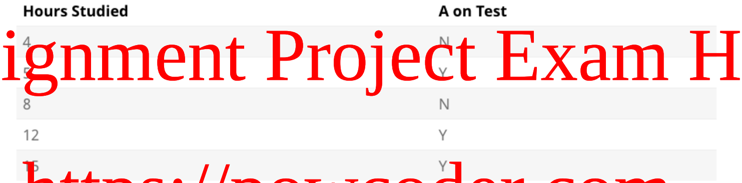
Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Split at 13.5

$$
\begin{aligned}
H(X <= 13.5) &= -\frac{2}{4}\log_2(\frac{2}{4}) - \frac{2}{4}\log_2(\frac{2}{4}) = 1.0 \\
H(X > 13.5) &= -\frac{1}{1}\log_2(\frac{1}{1}) - \frac{0}{1}\log_2(\frac{0}{1}) = 0 \\
H(X_{new}) &= H(X <= 13.5) + H(X > 13.5) = \frac{4}{5}1.0 + \frac{1}{5}0 = 0.8 \\
G(X_{new}) &= 0.971 - 0.8 = 0.171
\end{aligned}
$$

Data Discretisation — Entropy Discretisation

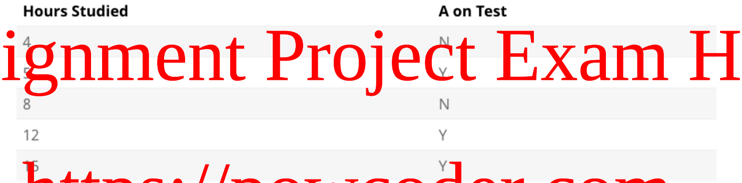


Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

- Split at 4.5: $G(X_{new}) = 0.322$
- Split at 6.5: $G(X_{new}) = 0.02$
- Split at 10: $G(X_{new}) = 0.42$
- Split at 13.5: $G(X_{new}) = 0.171$

Data Discretisation — Entropy Discretisation



Hours Studied	A on Test
4	N
5	Y
8	N
12	Y
15	Y

Figure is adapted from http://kevinmeurer.com/a-simple-guide-to-entropy-based-discretization/

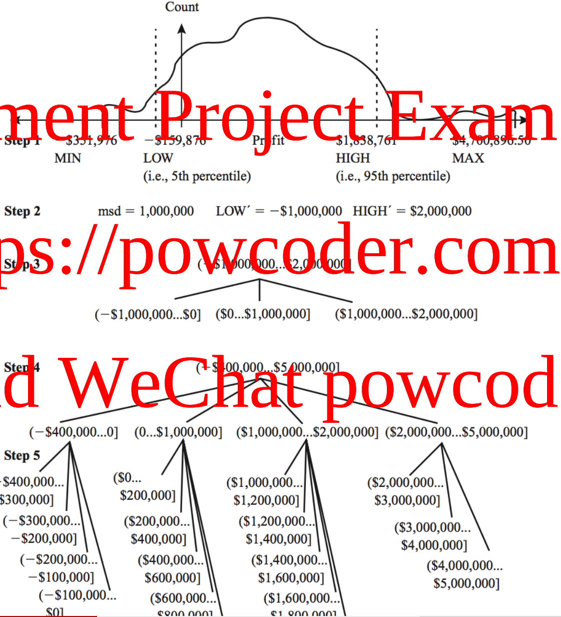
- When to step the algorithm
  - Terminate when a specified number of bins has been reached
  - Terminate when information gain falls below a certain threshold.

Concept Hierarchy for numerical data

A simple 3-4-5 rule can be used to segment numeric data (attribute values) into relatively uniform,“natural” intervals.

- If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width.
- If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals intervals
- If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

Segmentation by natural partitioning



Count

Step 1 −$351,976 MIN −$159,876 LOW (i.e., 5th percentile) Profit $1,838,761 HIGH (i.e., 95th percentile) $4,700,896.50 MAX

Step 2 msd = 1,000,000 LOW′ = −$1,000,000 HIGH′ = $2,000,000

Step 3 (−$1,000,000...$2,000,000]

(−$1,000,000...$0] ($0...$1,000,000] ($1,000,000...$2,000,000]

Step 4 (−$400,000...$5,000,000]

(−$400,000...0] (0...$1,000,000] ($1,000,000...$2,000,000] ($2,000,000...$5,000,000]

Step 5

(−$400,000... −$300,000]
(−$300,000... −$200,000]
(−$200,000... −$100,000]
(−$100,000... $0]

($0... $200,000]
($200,000... $400,000]
($400,000... $600,000]
($600,000... $800,000]

($1,000,000... $1,200,000]
($1,200,000... $1,400,000]
($1,400,000... $1,600,000]
($1,600,000... $1,800,000]

($2,000,000... $3,000,000]
($3,000,000... $4,000,000]
($4,000,000... $5,000,000]

Outline

1 Data Transformation

2 Data Discretisation

3 Feature Engineering & Data Sampling

4 Summary

Feature Engineering

- Feature extraction (or generation)
  - Generate new features from raw data or other features
  - Goals
    - Produce more meaningful/descriptive/discriminant features

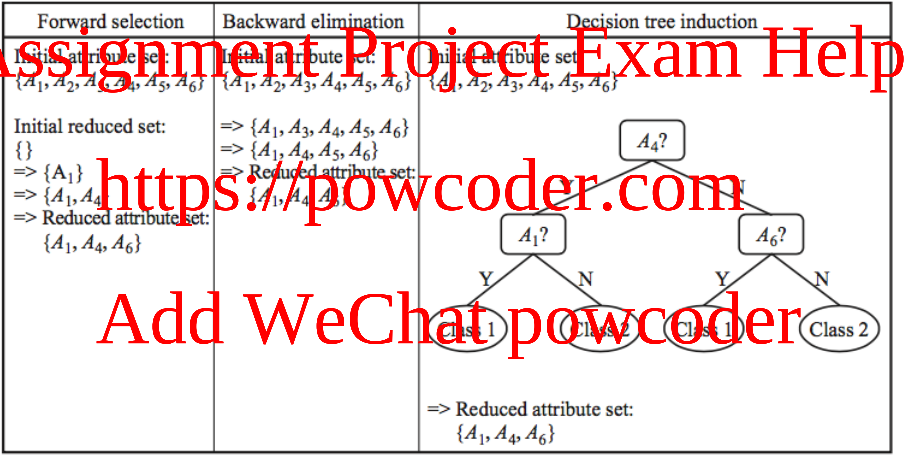
- Feature selection
  - Select a subset of available features based on some criteria
  - Goals
    - Remove irrelevant data
    - Increase predictive accuracy of learned models
    - Improve learning efficiency
    - Reduce the model complexity and increase its interpretability

Feature Subset Selection

Feature subset selection reduces the data set size by removing irrelevant or redundant features.

- Goal: find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes
- Methods
  - Stepwise forward selection.
  - Stepwise backward elimination.
  - Combination of forward selection and backward elimination
  - Decision tree induction.

Feature Subset Selection



Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: {} => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$

A_4?
Y
N
A_1?
A_6?
Y
N
Y
N
Class 1
Class 2
Class 1
Class 2

Figure is from "Data mining: know it all"

Data Sampling Methods

- Sampling methods are used to choose a representative subset of the data
- Why?
  - Reduce the volume of data
  - Fix imbalance distribution
  - Creating training, validation, testing sets.

Data Sampling Methods

- Methods: Suppose that a large dataset, D, contains N tuples, the ways we can used to do data reduction:
    - Simple random sample without replacement (**SRSWOR**) of size *s*:
        - Draw s of the N tuples from D (s < N), where the probability of drawing any tuple in D is 1/N
    - Simple random sample with replacement (**SRSWR**) of size *s*.
        - Similar to SRWOR, except that after a tuple is drawn, it is placed back in D so that it may be drawn again.

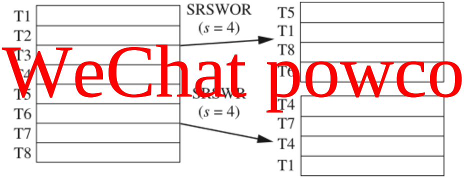
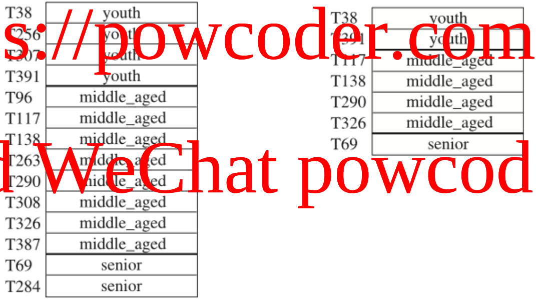


Figure is from "Data mining: know it all"

Data Sampling Methods

- Methods: Suppose that a large dataset, D, contains N tuples, the ways we can used to do data reduction:
  - Stratified sample
    - If D is divided into mutually disjoint parts called strata, a stratified sample of D is generated by obtaining an SRS at each stratum



T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Figure is from "Data mining: know it all"

Summary

- Data transformation:
  - Normalisation/Scaling
  - Data transformation generating new features
  - Nominal to numerical transformation
- Data Discretization
- Feature selection and data sampling