Fearture Manipulation

DATA PREPROCESSING & PREPARATION

Overview

- Introduction of Data Preparation/Preprocessing
  - What data preparation include
  - Why data preparation
  - Types of data preparation techniques

- Data Preparation Techniques
  - Training vs Testing, k-fold cross validation
  - Categorical Data Encoding
  - Normalisation
  - Discretisation

- Feature Manipulation
  - Dimensionality Reduction
  - Feature Construction
  - Feature Selection

Data Preprocessing/Preparation

- "data munging", prepare the final data set(s) for modelling
- Takes over 80% of time and effort in the project
- Five steps:
  - Data Selection: determine data sets to be used, select features, select instances
  - Data Cleaning: to correct, impute, or remove erroneous values, missing values
  - Data Construction: constructive data preparation operations, e.g. feature construction, instance generation, feature transformation
  - Integrate data: create new records or values by combined from multiple data source, merge data from different sources, aggregations
  - Format data: re-format data, convert to format convenient for modelling

Why Data Preprocessing?

- Data in the real world:
  - incomplete: missing attribute values
  - inconsistent: “03/07/2015”, “March 07, 2015”
  - noisy: containing errors or outliers, gender=“Male”, pregnant = “Yes”
  - large-scale/big data: with a large number of features and instances
    - different types: numeric, nominal, text, Web data, images, audio/video
- Different ML tools use different data formats; Different ML methods have different requirements
- How to use the data: whole set of data or subsets of data
- Garbage in, garbage out

Data Preprocessing Tasks

Tasks cover three major aspects:

- data format: for different tools, methods
- experiment preparation: how to use the data
- input data processing

Common data preprocessing tasks:

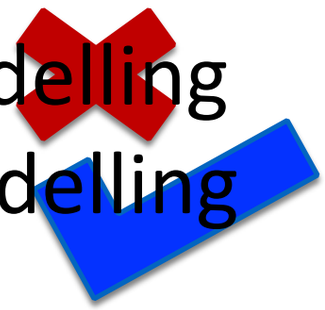
- data cleaning: identify and correct mistakes
- data transforms: change the scale/distribution of variables
- dimensionality reduction: create compact projections of the data.
- feature selection: identify relevant input variables
- feature construction: derive new variables from available data

Data Leakage

- The manner in which data preparation techniques are applied to data matters
- A Problem with naive data preparation - data leakage
- Information/knowledge about the holdout dataset, e.g. a test dataset, leaks into the data used to train the model
- result in an incorrect estimate of model's perdiction performance

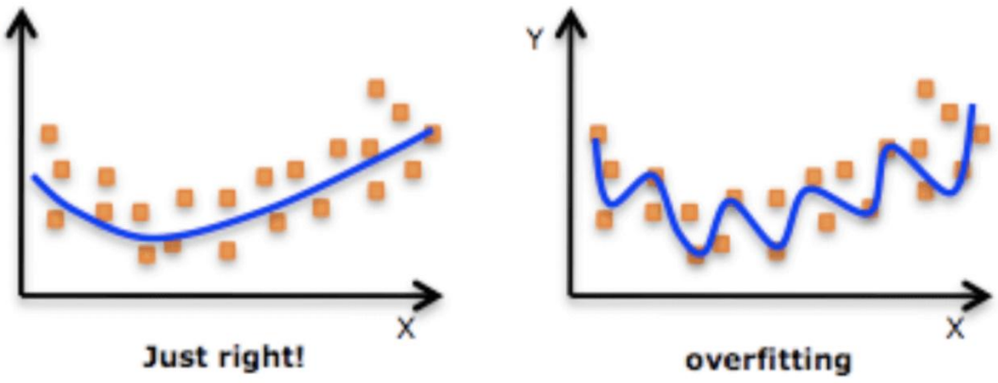
Data Preparation->Data Splitting-> Modelling

Data Splitting-> Data Preparation-> Modelling



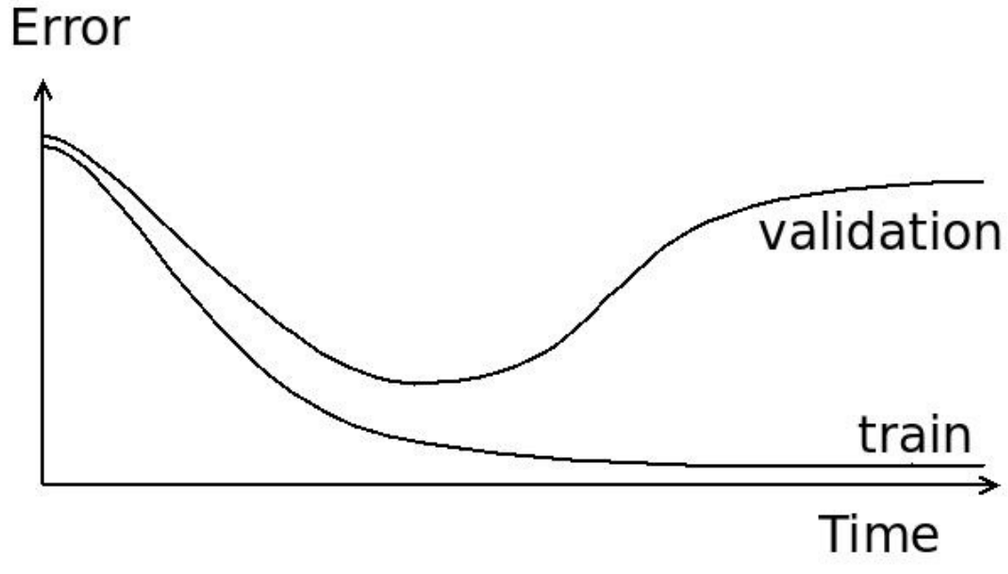
Training and Test Sets

- Training set: to learn/train a model
- Test set: to measure the performance of the model
  - Training—Test: 50%—50%; 2/3 — 1/3; 70%—30%
- Both need to be representative of the original data as a set
- Generalisation VS overfitting



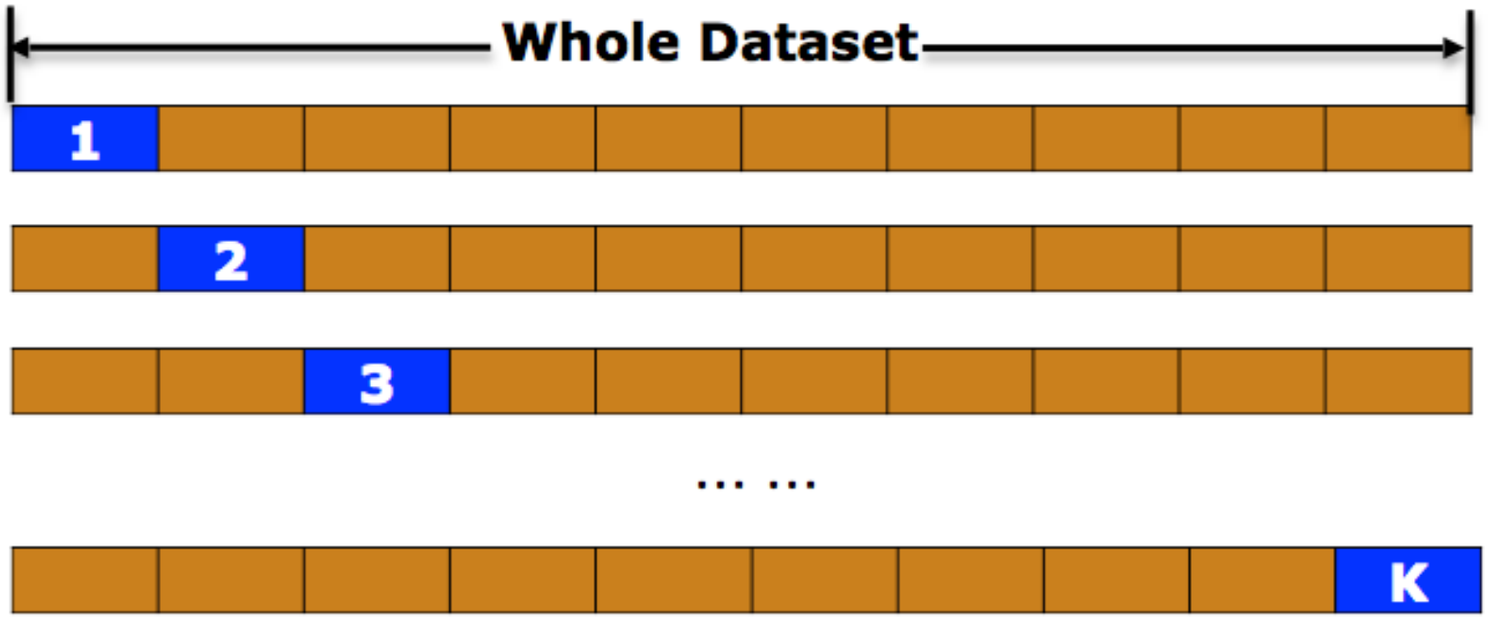
- Validation set: monitor the training process
- Validation set VS Test set
- Training —Test—Validation:

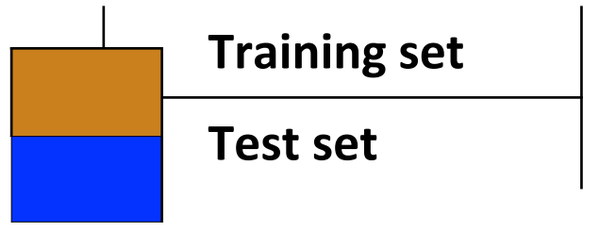
  1/3 — 1/3 — 1/3



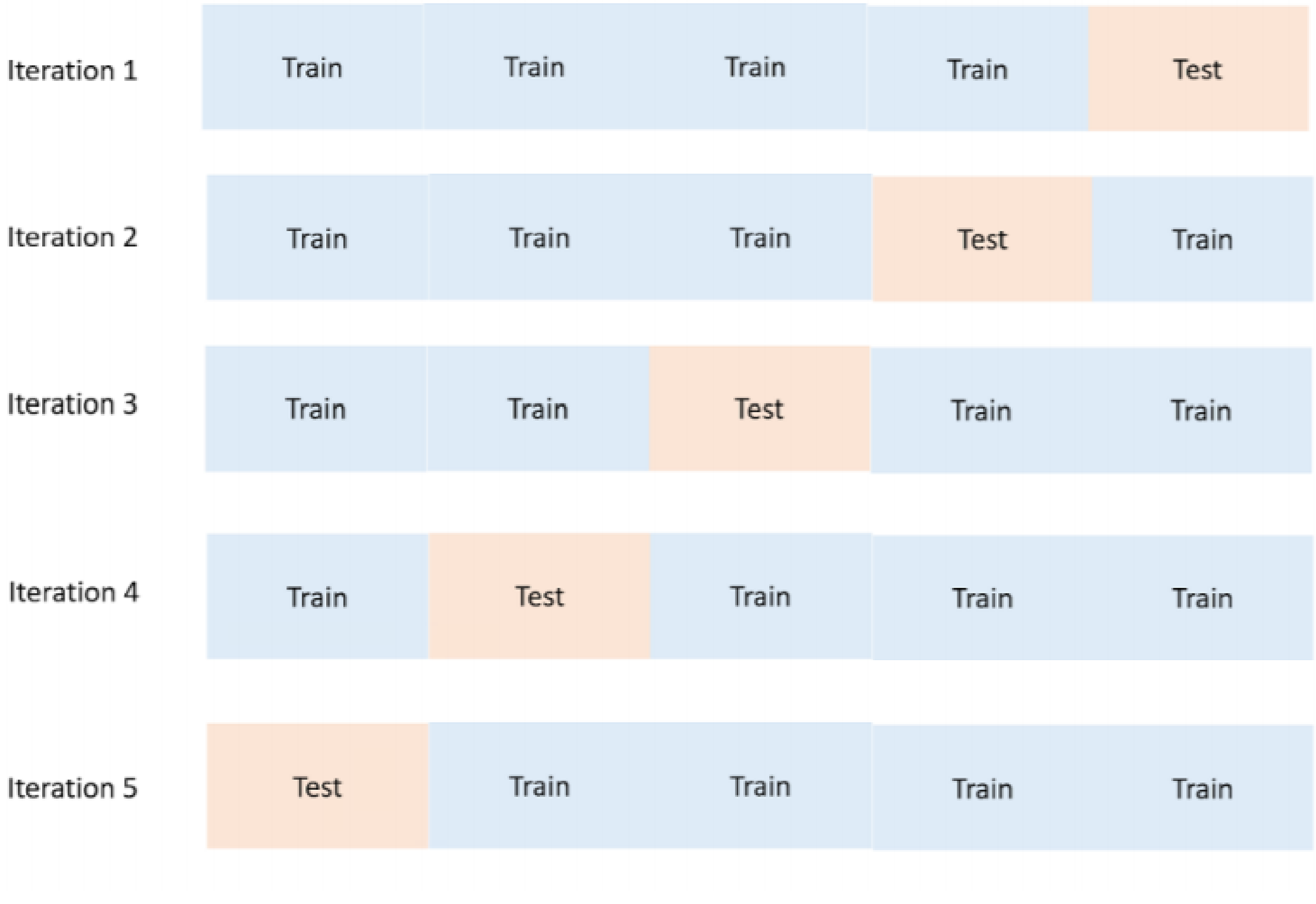
K-fold Cross Validation (K-CV)

- Used when only a small number of instances are available
  - Split the whole dataset to K folds with equal size
  - Use 1 fold as test set, and the other (K-1) folds as training set
  - Repeat K times to make sure each fold has a chance to be the test set
  - Average the K test performances (e.g. error rates)

- 



Example: 5-fold Cross Validation (K-CV)



Data Processing

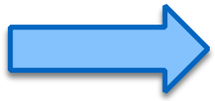
- Different types of data:
  - continuous data
  - discrete data: categorical/nominal; ordinal; Integer
  - other/special types of data (multi-media data):Text data, hyperlink data, image data
- Encoding categorical data: convert categorical data to numeric value
- Nomalisation/Scaling: transform columns/rows to a consistent set
- Discretisation: convert a numeric attribute to a nominal attribute
  - e.g. Temperature attribute from {50, 80} to {low, high}

- Handle missing data, noisy data, outliers, unbalanced data, redundant data

Categorial Data Encoding Scheme

- Categorical variables : contain label values rather than numeric values
- One Hot Encoding: for each unique value in a categorical column, a new column is added

Country
USA
UK
USA
France
USA
UK



dummy variables

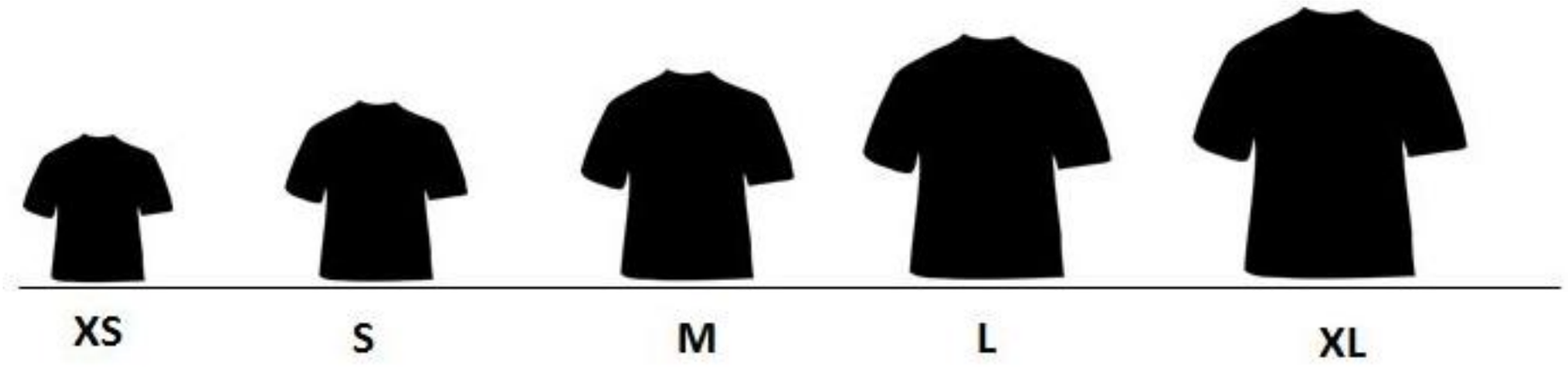
USA	UK	France
1	0	0
0	1	0
1	0	0
0	0	1
1	0	0
0	1	0

- Dummy Encoding: N-1 columns for N unique values

UK	France
0	0
1	0
0	0
0	1
0	0
1	0

Encoding Ordinal Variables

- Ordinal data: categorical data that have a natural rank order, e.g. *Poor, Good, Very Good, and Excellent*



- Ordinal encoding: assign integers to labels in certain order

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

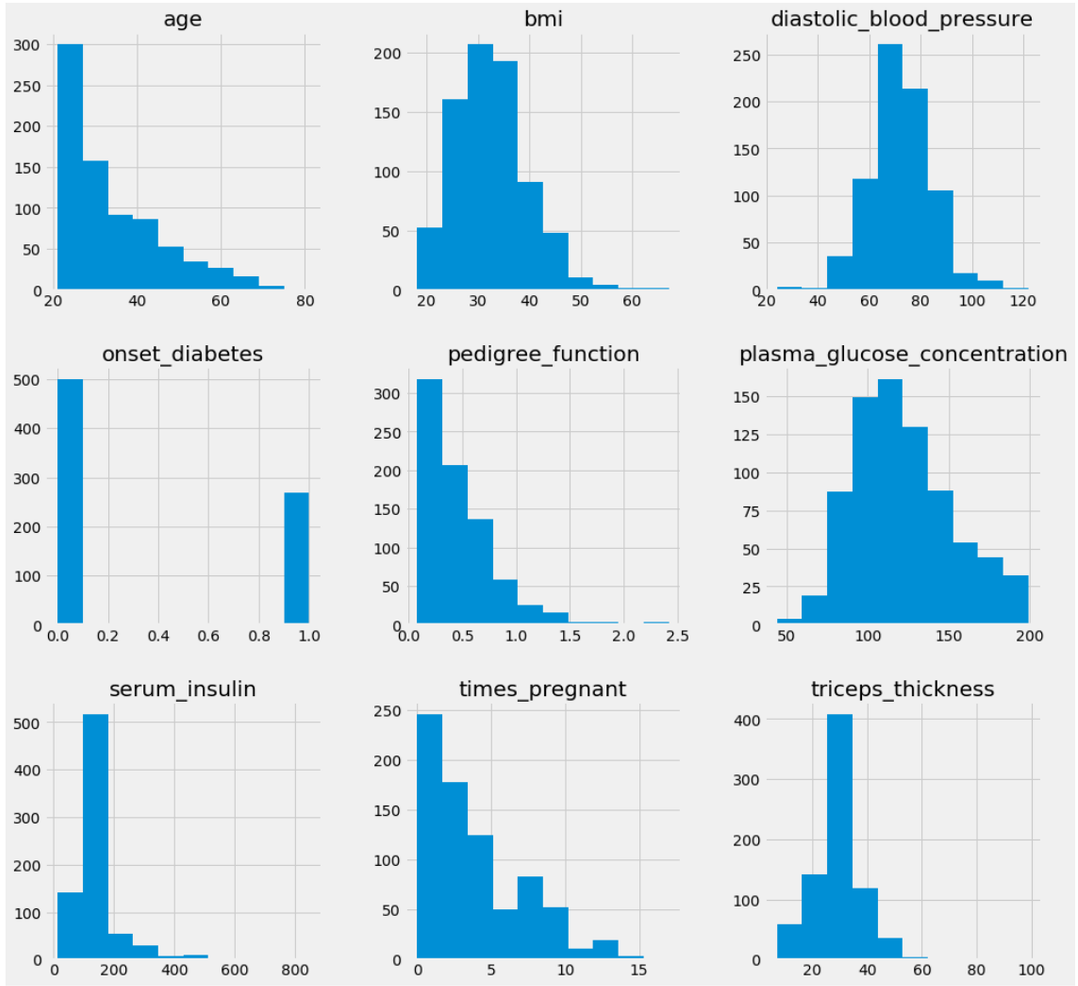
Normalisation/Scaling

- Numeric data, feature values in different ranges

- Some machine learning methods e.g. KNN, SVM, gradient descent, are affected greatly by the scale of the data

- Normalization transforms columns and/or rows to a consistent set of rules

- a common form - transform all features to be between a consistent and static range of values, e.g.[0, 1]



Example of variables with vastly different scales

Min-Max Normalisation/Scaling

Linear scaling

- To the range [0, 1]: $$x' = \frac{x - X_{min}}{X_{max} - X_{min}}$$

  the min are all zeros and the max values are all ones

- To a pre-defined range [New_{min}, New_{max}]:

$$x' = \frac{x - X_{min}}{X_{max} - X_{min}}(New_{max} - New_{min}) + New_{min}$$

- use the scikit-learn object MinMaxScaler

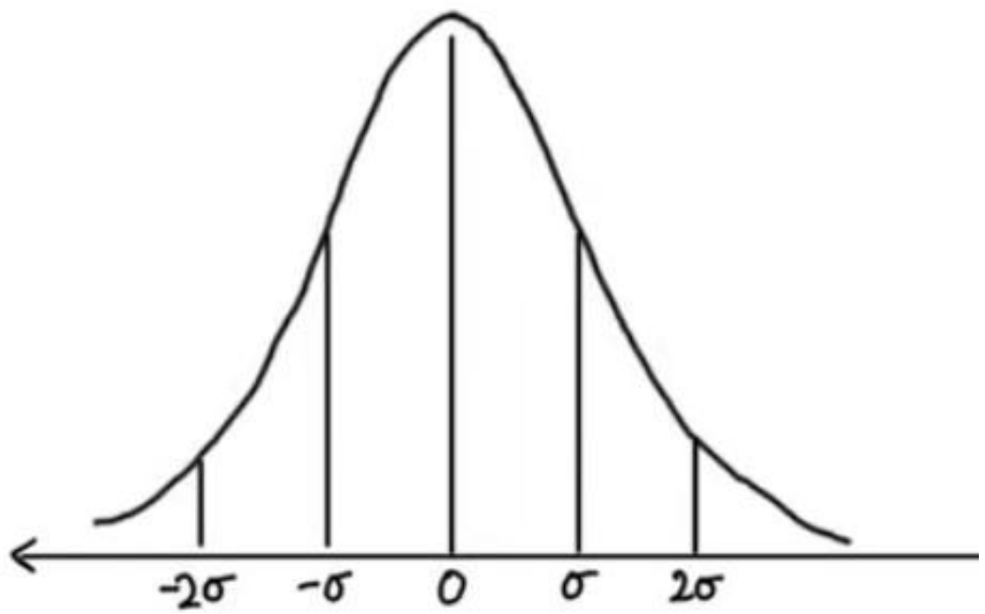
Z-Score Standardisation

- also referred to as center scaling
- center values around a mean of zero and a standard deviation of one utilising the statistical idea - a z-score/standard score

$$z = (x - \mu)/\sigma$$

μ is the mean, σ is the standard deviation of the feature

- use the scikit-learn object StandardScaler



Normalisation or Standardisation?

- Standardisation can give values that are both positive and negative centered around zero
- Normalisation makes different variables to have the same range
- If the distribution is normal, then it should be standardised, otherwise => normalise
- If in doubt => normalise
- might be a good idea to have a mixture of standardised and normalised variables=> standardised followed by normalised

Discretisation

- Discretisation/binning: a process of converting continuous numeric values such as price, age, and weight into discrete intervals
- Some algorithms prefer/require categorical inputs, e.g. DT, rule-based algorithms
- For data smoothing, handle outliers

Two types

- Unsupervised discretisation - does not depending on class label
- Supervised discretisation - depends on class label

- Use scikit-learn object KBinsDiscretizer

Discretisation: Equal-Width/Uniform

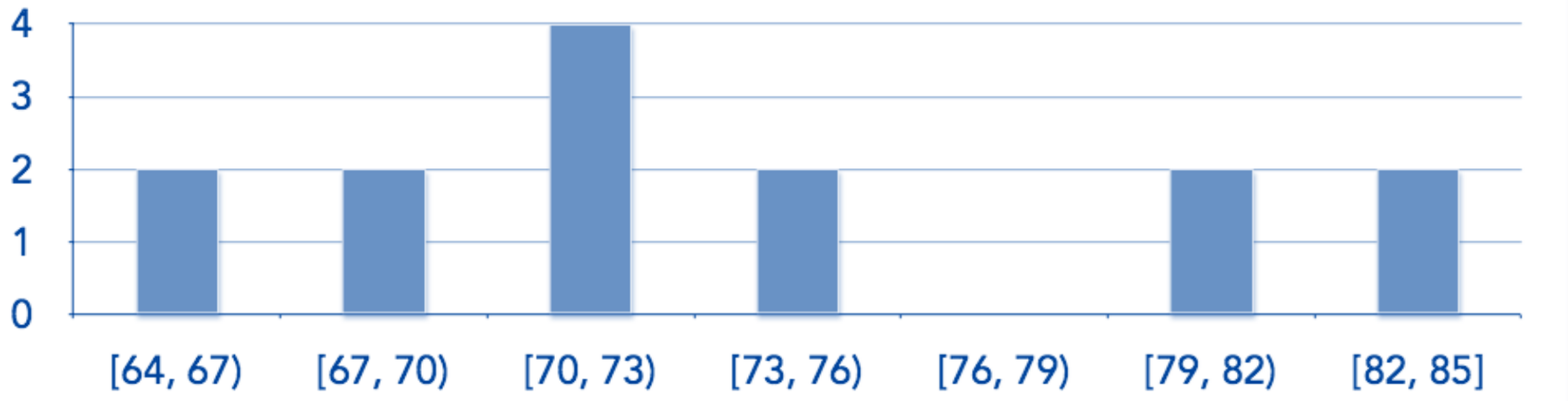
Convert a numeric attribute to a nominal/ordinal attribute with N possible values

- Find the Maximum and Minimum values of the attribute
- Divides the range [Min, Max] into N intervals of equal size
- The width of intervals: W=(Max - Min)/N

- KBinsDiscretizer(n_bin=7, encode='ordinal', strategy='uniform')

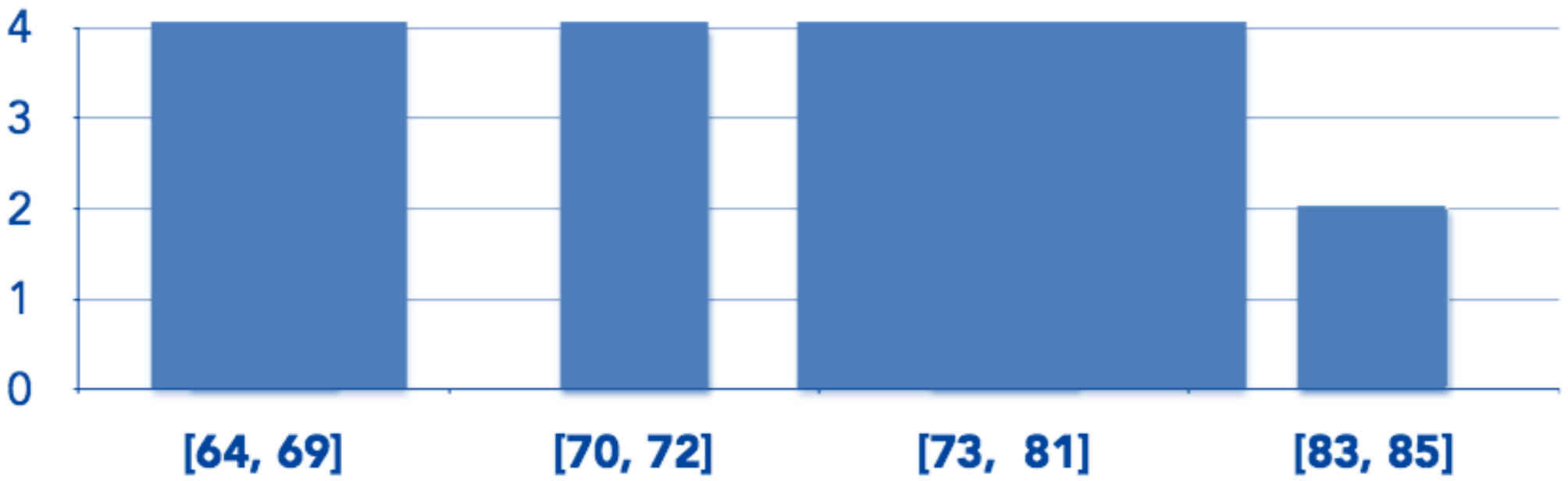
Example: Temperature values are 85, 80, 83, 70, 64, 65, 68, 71, 69, 72, 75, 75, 81,72

- Max=85, Min=64, N=7, W = (85-64)/7=3



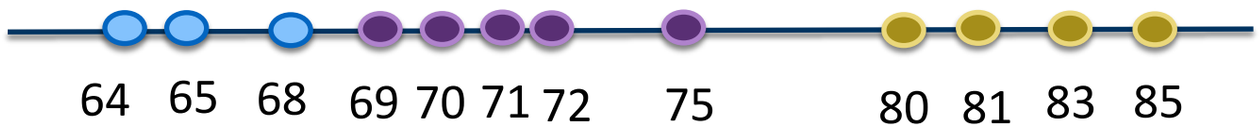
Discretisation: Equal-Depth/Frequency/Quantile

- Divides the range [Max, Min] into N intervals
- Each interval including approximately same number of instances
- KBinsDiscretizer(n_bin=4, encode='ordinal', strategy='quantile')
- Example:
  - Sort the 14 Temperature values
  - 64, 65, 68, 69, 70, 71, 72, 72, 75, 75, 80, 81, 83, 85
  - N=4



Clustering for Discretisation

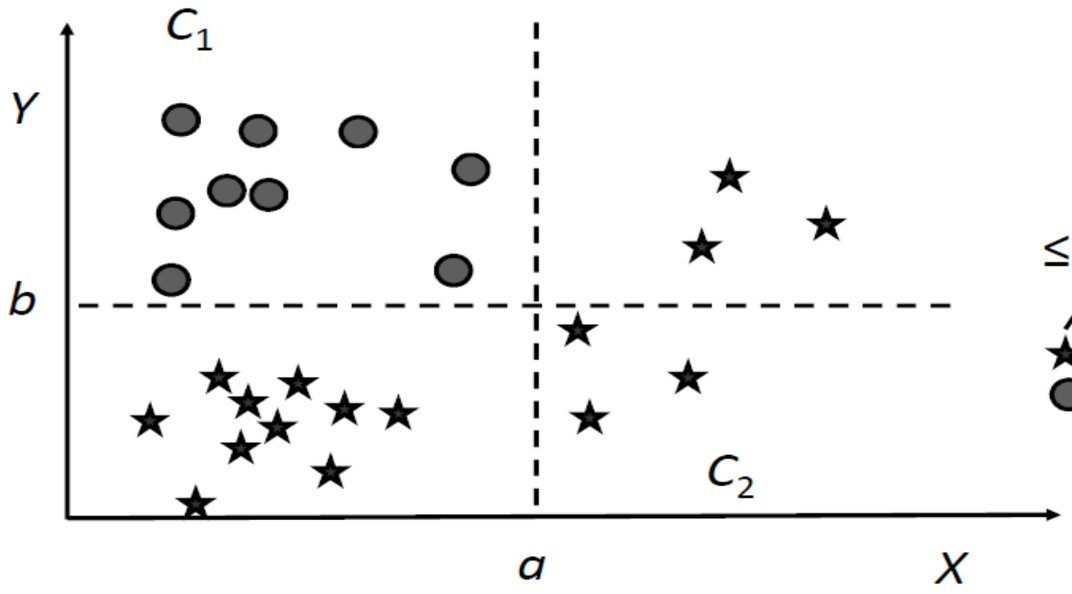
- Clustering algorithm can be applied to discretize a numeric attribute by partitioning the values into clusters
- K-means can find the number and boundaries of intervals
- Values in each bin have the same nearest center of a one dimensional K-means clustering
- KBinsDiscretizer(n_bin=3, encode='ordinal', strategy='kmeans')

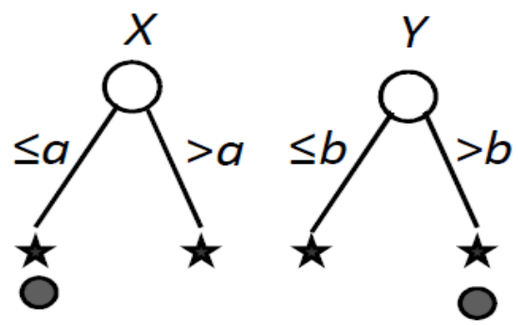


1R Discretisation

- Supervised method: consider class label
- Find a split point(s) to discretise a numeric feature to different categories
- For each split point:
  - Find the frequency of each class in each category
  - Assign the most frequent class label to each category
  - Calculate the error over all categories

  The best split point is the one with minimum error
- available in the R package dprep (function: disc.1r)

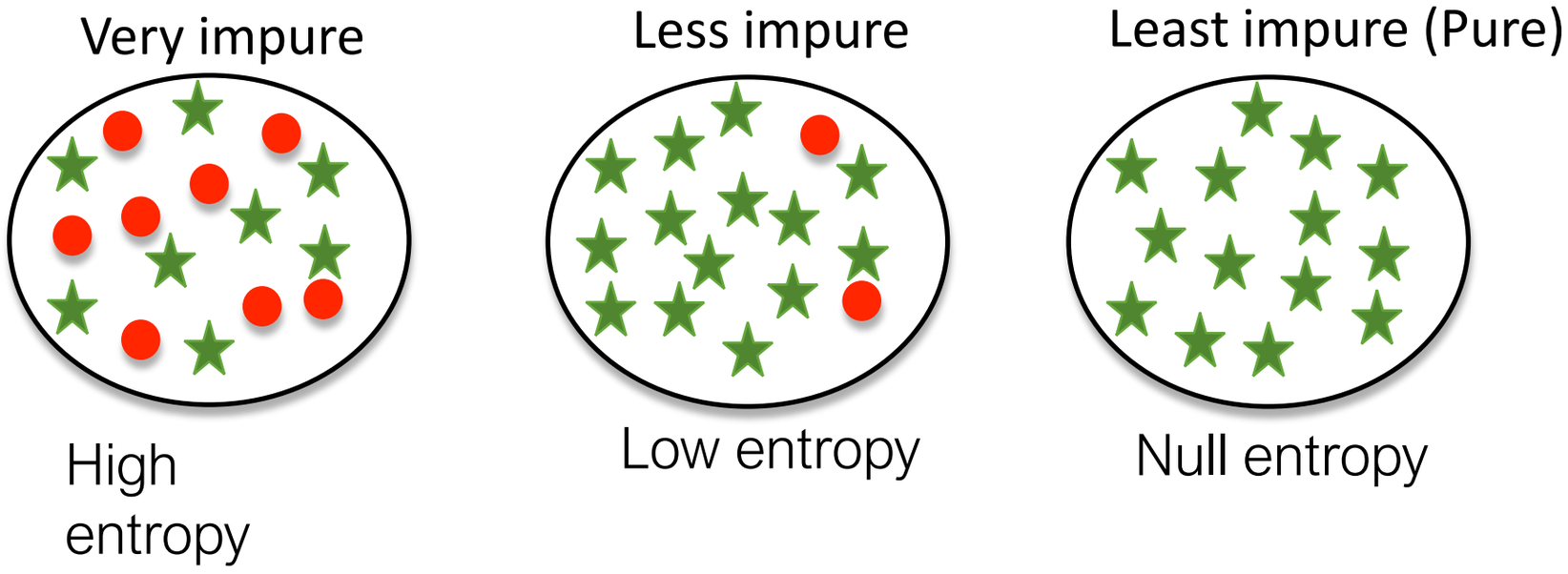




http://www.uta.fi/sis/tie/tl/index/Datamining6.pdf

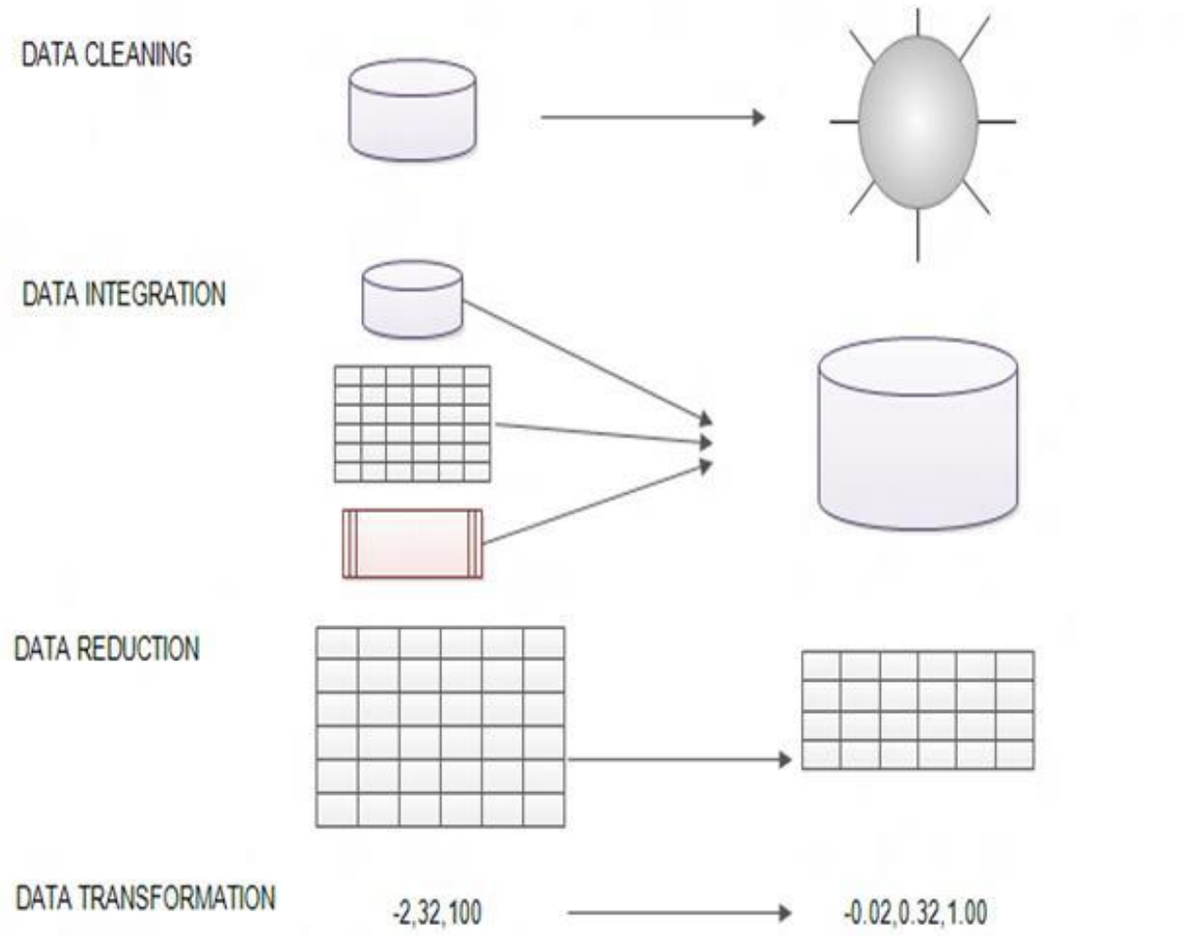
Entropy Based Discretisation

- **Entropy** measures the impurity or uncertainty in a group of examples
- ***E(S)*** measure the Entropy of the training set S, *Pc* is the **proportion** of class *Cc* in *S* $E(S) = -\sum_{c=1}^{N} p_c * log_2(p_c)$
- Recursively find a partition boundary T minimizing the **ne**t **e**ntropy in each potential subset $NE(S,T) = \frac{|S_1|}{|S|}E(S_1) + \frac{|S_2|}{|S|}E(S_2)$
- available in the R package dprep (function: disc.mentr)



Summary

- Data preprocessing is an important step in KDD/DM
- Encoding categorical data
- Data normalisation
- Data discretisation



Lecture 04 – Fearture Manipulation

FEATURE MANIPULATION

Overview

- Introduction of Data Preparation/Preprocessing
  - What data preparation include
  - Why data preparation
  - Types of data preparation techniques

- Data Preparation Techniques
  - Training vs Testing, k-fold cross validation
  - Categorical Data Encoding
  - Normalisation
  - Discretisation

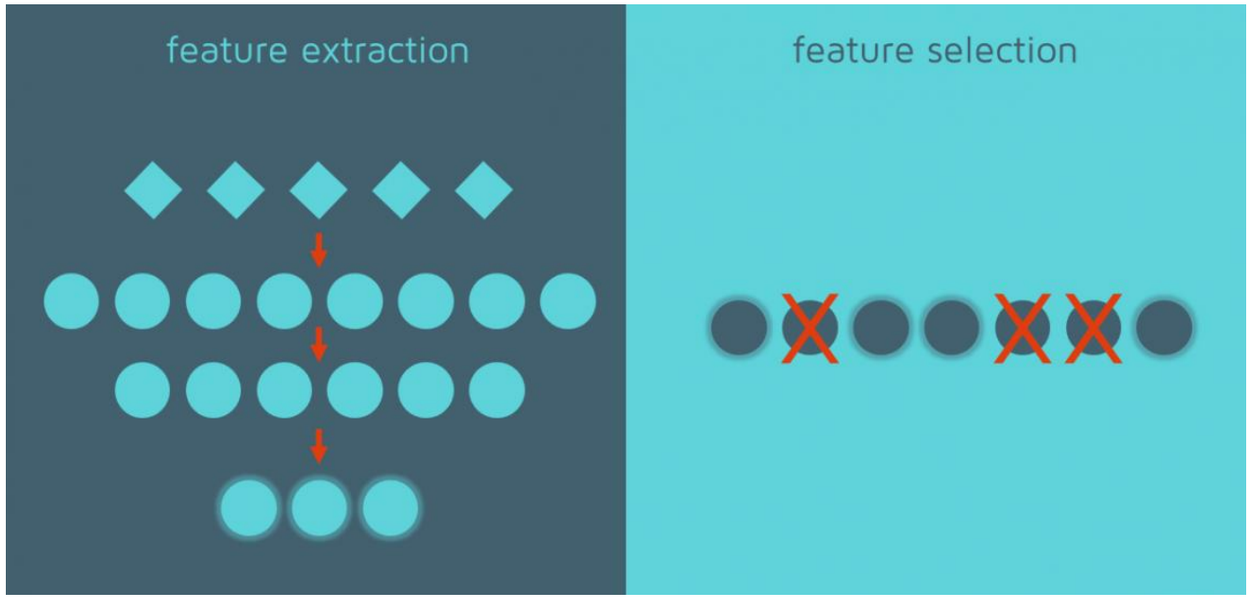
- Feature Manipulation – Dimensionality Reduction
  - Feature Extraction
  - Feature Selection

Feature Manipulation

Dimensionality reduction:

1) Feature extraction: apply a transformation on the original feature vector to reduce its dimensionality from *d* to *m*

2) Feature selection: select a small subset of original features



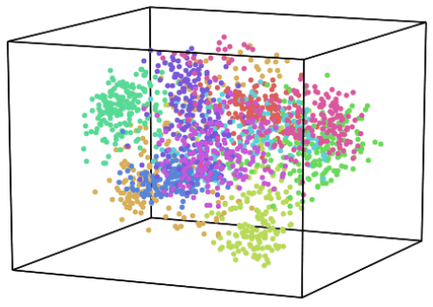
Dimensionality changing:

3) Feature construction: a process that discovers missing information about the relationships between features and augments the feature space by creating additional features

https://www.analyticsvidhya.com/blog/2021/04/guide-for-feature-extraction-techniques/

Why Feature Manipulation

- Raw data contains a mix of features, where not all are useful to generate the model



- Choose an optimal subset of features according to a certain criterion
- To reduce dimensionality and noise
- To visualize the data for model selection
- To improve performance (in terms of speed, predictive power, simplicity, interpretability of the model)

Edible or poisonous?

Attribute Information:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Feature Extraction

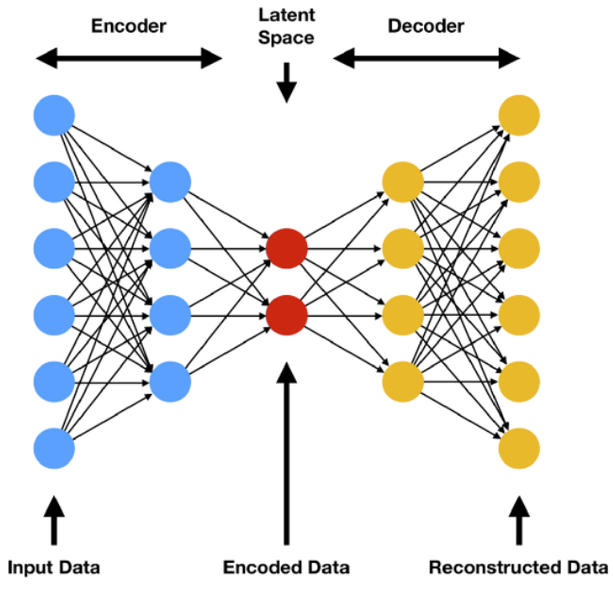
- Transform the original feature vector to reduce its dimensionality from *d* to *m, usually m < d*
- m new features retain most of the information
- Come at the cost of loss of individual feature identity
- Extracted features are not interpretable by humans
- Used for: data compression, noise reduction, improve modelling performance, reduce overfitting, data visualization

Two main approaches

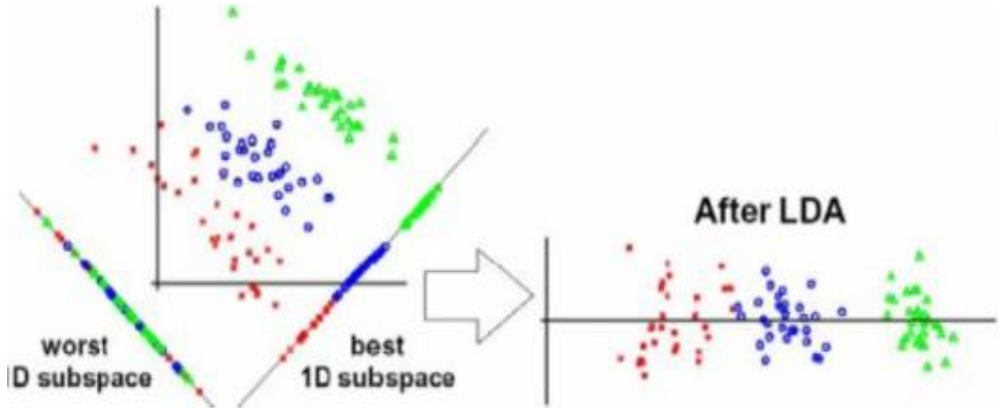
- Projection: project onto a lower-dimensional space in a way to preserve the distances
- Manifold learning: model the manifold the training instances lie

Feature Extraction Techniques

- Principle Component Analysis (PCA): linear, unsupervised, computing the uncorrelated principal components
- Independent Component Analysis (ICA): linear, unsupervised, take input data as a mixture of independent components, aim to identify these components
- Linear Discriminant Analysis: supervised, maximize the distance between the mean of each class and minimize the spreading within the class
  - sklearn.discriminant_analysis.**LinearDiscriminantAnalysis**
- Nonlinear methods: Autoencoder, T-SNE, …



https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/



Linear Discriminant Analysis

https://mlalgorithm.wordpress.com/2016/06/18/linear-discriminant-analysis-lda/

Feature Selection

- FS: reduce the number of input variables to those that are believed to be most useful, remove non-informative or redundant features
- Useful for high dimensional data, e.g. genomic DNA and text data

Methods

- Univariate (considers a feature independently of others)
  - Pearson correlation coefficient
  - Chi-square
  - F-score

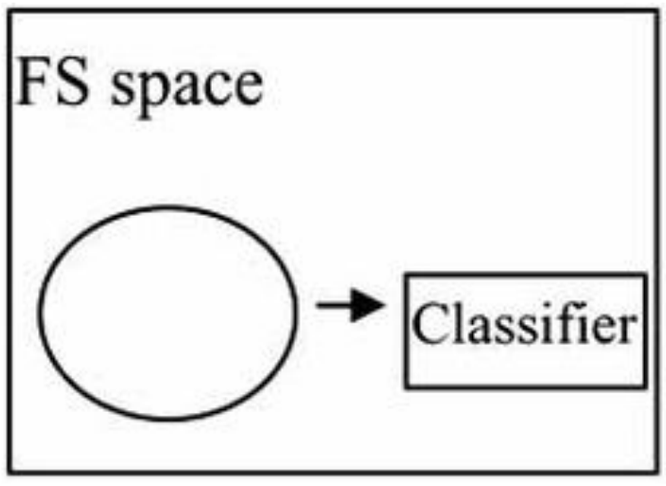
  and more, such as mutual information, Relief

- Multivariate (considers all features simultaneously)
  - Recursive feature elimination
  - Linear models such as support vector machine, and Lasso

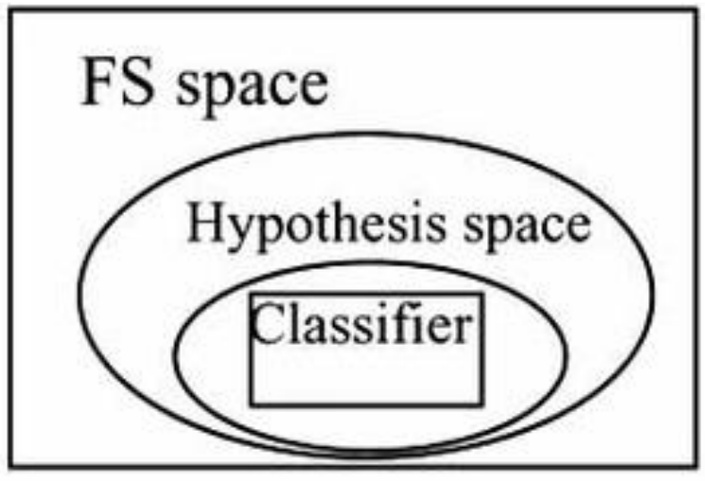
Filter, Wrapper and Embedded

- Feature Selection methods also (often) can be divided into Filter, Wrapper, and Embedded

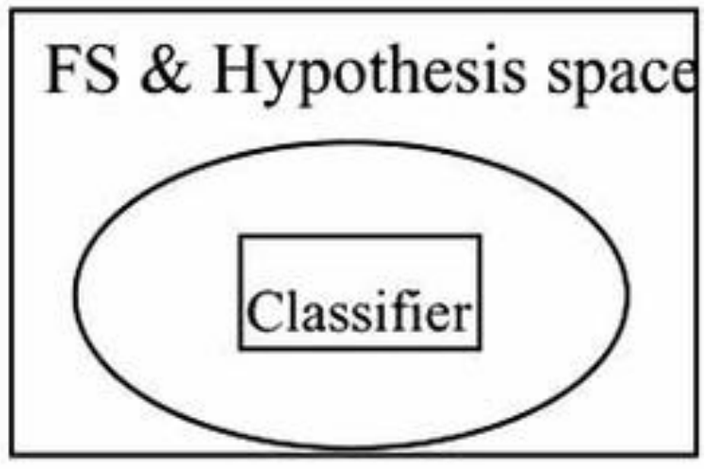
Filter select the feature subsets prior to the classifier model being built



Wrapper methods use a classifier to select the feature subsets prior to the classifier model being built



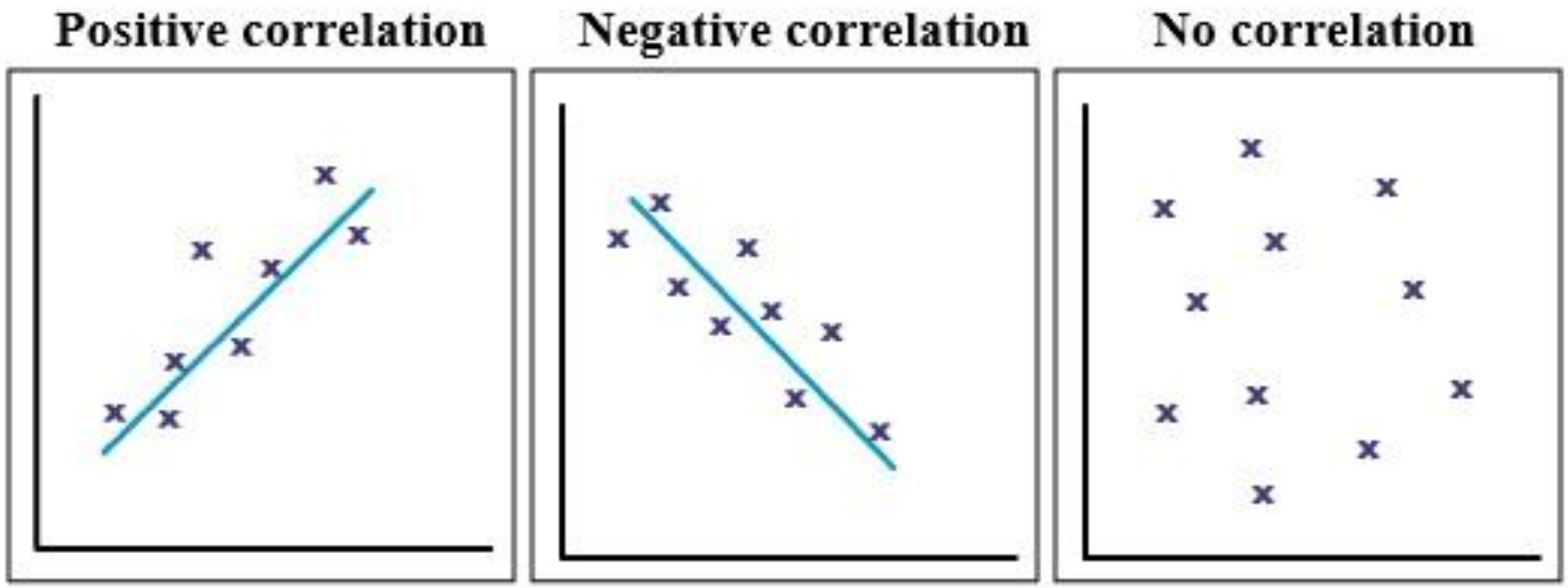
Embedded methods select the feature subsets at the same time as the classifier model is being built.



Univariate FS - Correlation based methods

Univariate methods measure some type of correlation between each input feature/variable and the target variable/class label

- Correlation: a statistical term describing the degree to which two variables move in coordination with one another
- Two variables move in the same direction/opposite directions, then have a positive correlation/negative correlation



Pearson correlation coefficient r

Measures the linear correlation between two continuous variables

- Formulas:

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{s_x s_y} \qquad r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

  cov refers to covariance, a measure of the joint variability of two variables, the sign of the covariance therefore shows the tendency between the variables.

- The correlation r is between -1 and 1, 1 means perfect positive correlation and -1 in the other direction

- Use sklearn.feature_selection.r_regression

Chi-square Test

- χ^2 (chi-square) test for discreate data

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- Test whether the observed frequencies match the expected frequencies

e.g. For two discrete features, A and B, let (Ai, Bj) denote the joint event of ($A = a_i$, $B = b_j$)

- *Observed: the observed frequency of the joint event (Ai, Bj)* $o_{ij} = Count(A = a_i, B = b_j)$
- *Expected: the expected frequency of (Ai, Bj),* $e_{ij} = \frac{Count(A=a_i) \times Count(B=b_j)}{n}$

Gender\ umbrella present	yes	no	Total
female	5	7	12
male	5	5	10
Total	10	12	22

o(female, yes) = 5

*e(female, yes) = 12*10/22≈5.455*

- larger χ^2 value, the more likely the variables are related
- Use sklearn.feature_selection.chi2

Some Other Measures for FS

- Mutual information - measures the reduction in uncertainty for one variable given a known value of the other variable, measures mutual dependency
- I(X; Y) measures the common information between two X and Y.
- Use sklearn.feature_selection.mutual_info_classif
  sklearn.feature_selection.mutual_info_regression

- Spearman: for continuous features/variable, nonlinear correlation, use scipy.stats.spearmanr
- ANOVA: between continuous feature and discreate label
  use sklearn.feature_selection.f_classif

FS based on Feature Importance

- Feature importance: scores to input features that indicates the relative importance of each feature
- highlight which features may be most relevant to the target
- Feature importance scores can be fed to a wrapper model to perform feature selection, also can use with recursive feature elimination

  e.g. sklearn.feature_selection.SelectFromModel

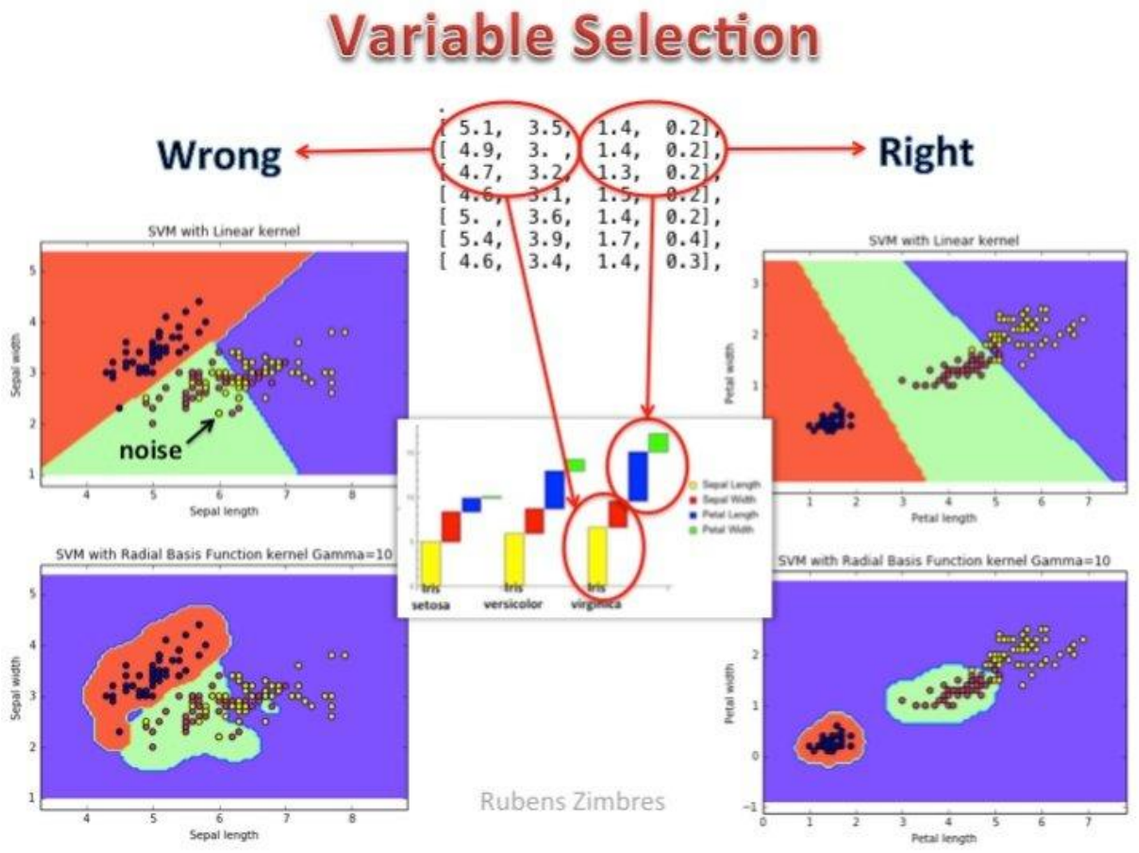
  sklearn.feature_selection.RFE

Feature importance score

- Permutation testing:
- (Linear) model coefficients: running logistic regression a few times and get the average on coefficients
- from decision tree: based on the reduction in the criterion, e.g. Gini or entropy, used to select split points
  - sklearn DecisionTreeRegressor -> feature_importances_

Univariate Feature Selection Limitations

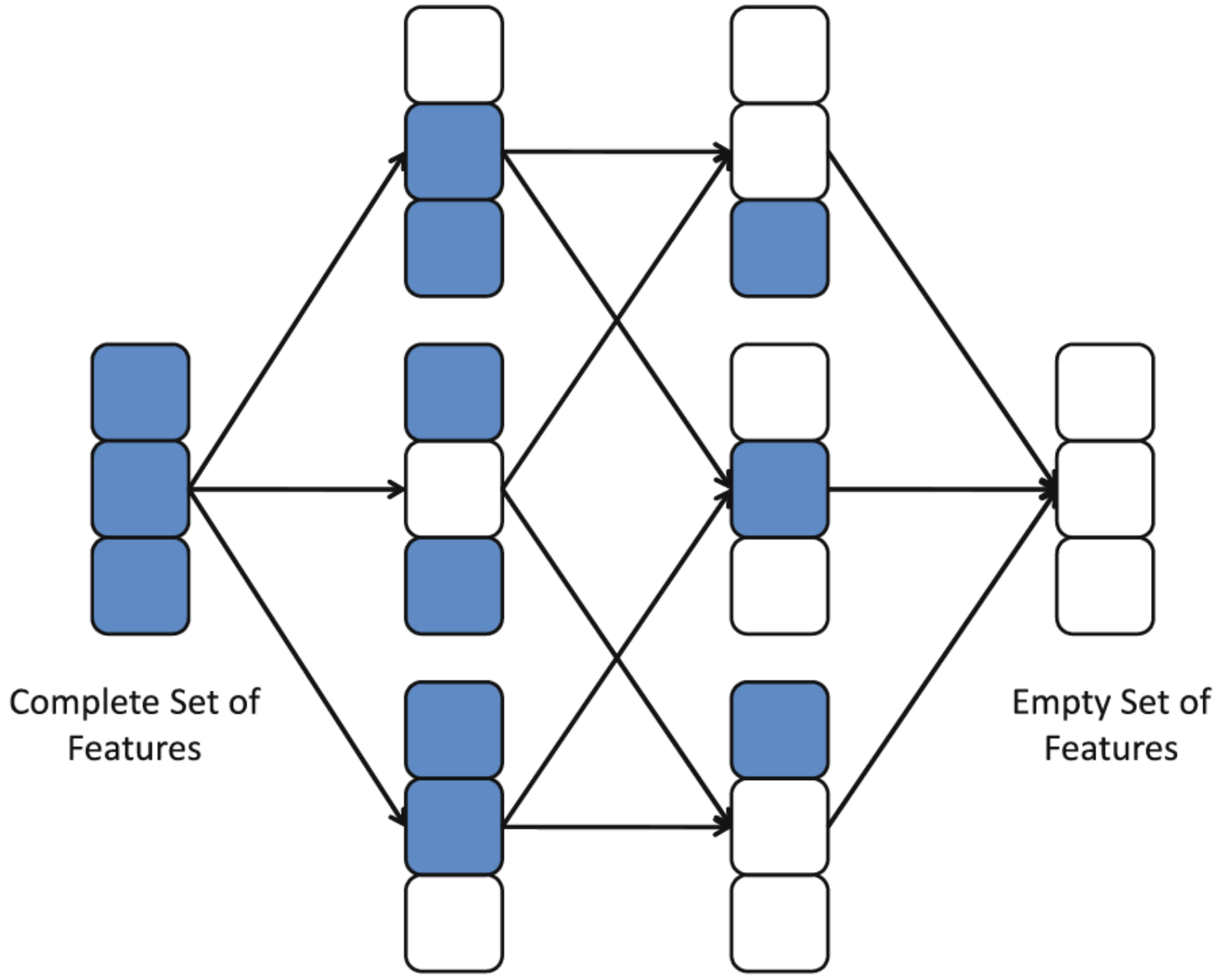
- Unclear how to tell in advance if feature selection will work
  - Only known way is to check but for very high dimensional data (at least half a million features) it helps most of the time
- How many features to select?
  - Perform cross-validation
- Does not consider feature interaction



https://www.kdnuggets.com/2017/06/practical-importance-feature-selection.html

Multivariate: Search for Subset of Features

Search Space:



Multivariate: Subset of Features

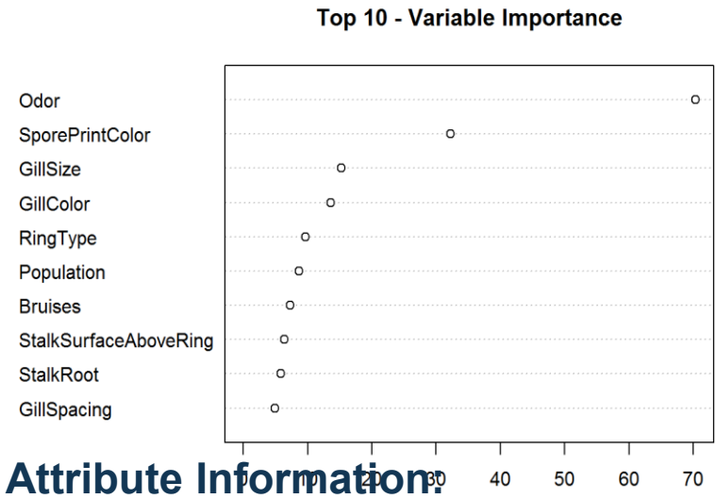
- Sequential Forward Generation (SFG): starting from an empty set, sequentially add the feature X that results in the highest objective function when combined with the features Y_k
  - works best when the optimal subset has a small number of features
  - unable to remove features that become obsolete after the addition of other features

- Sequential Backward Generation (SBG): starting from the full set, sequentially remove the feature x that results in the smallest decrease in the objective function
  - works best when the optimal subset has a large number of features
  - cannot re-evaluate a feature after discarded

More advanced Feature Selection

- FS can be considered as a search problem, where each state of the search space corresponds to a subset of features selected.
- The selection can be represented as a binary array: E.g. 0011101110110000001010
  - 1 if the feature is currently selected by the algorithm and
  - 0 if it not
- There should be a total of 2^n subsets where N is the number of features of a data set.

  2^{22} = 4,194,304 combinations to test!



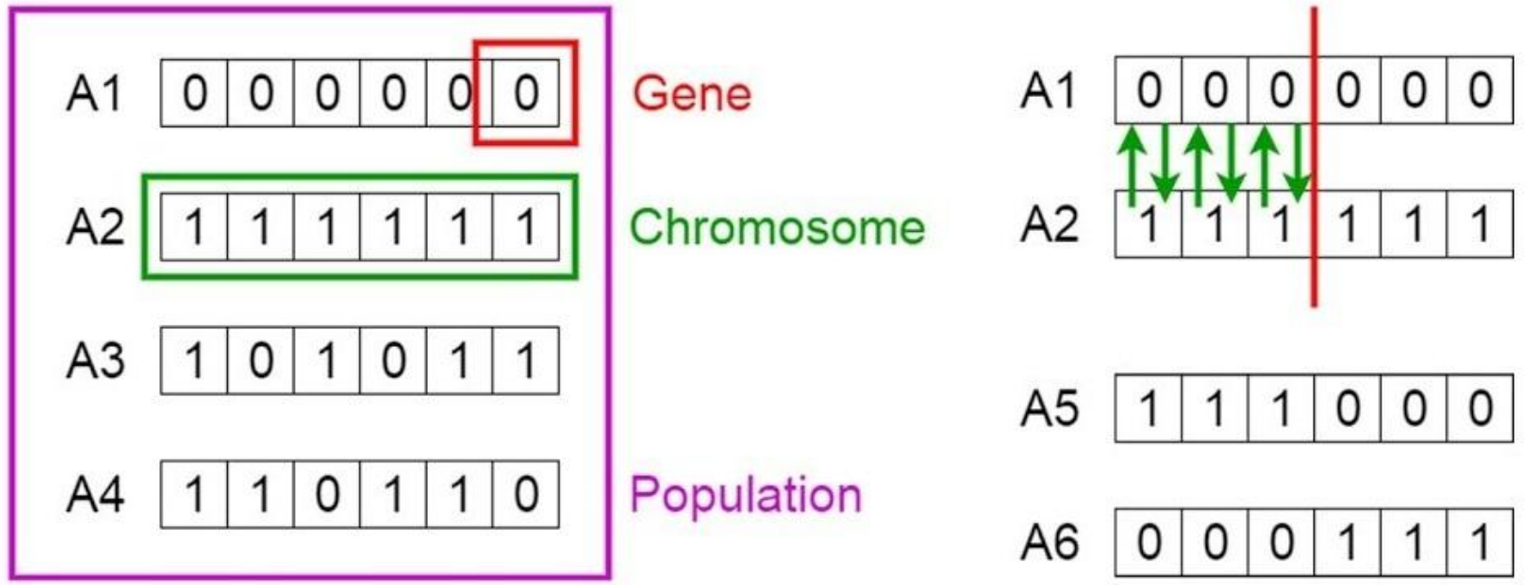
Attribute Information:

1	cap-shape	0
2	cap-surface	0
3	cap-color	1
4	bruises?	1
5	odor	1
6	gill-attachment	0
7	gill-spacing	1
8	gill-size	1
9	gill-color	1
10	stalk-shape	0
11	stalk-root	1
12	stalk-surface-above-ring	1
13	stalk-surface-below-ring	0
14	stalk-color-above-ring	0
15	stalk-color-below-ring	0
16	veil-type	0
17	veil-color	0
18	ring-number	0
19	ring-type	1
20	spore-print-color	0
21	population	1
22	habitat	0

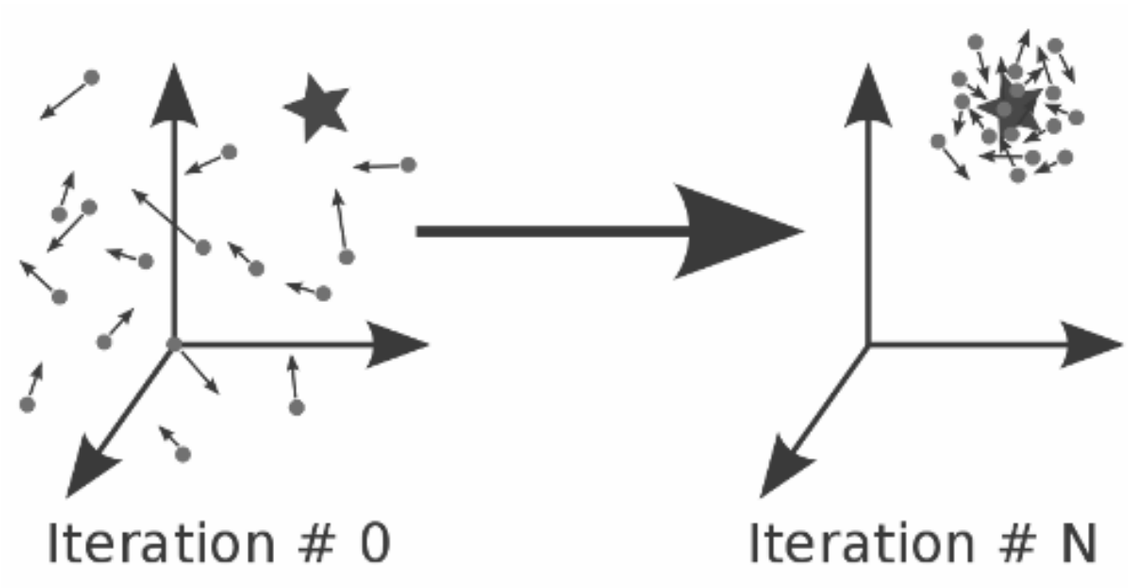
https://www.r-bloggers.com/random-forest-classification-of-mushrooms/

More advanced FS Methods

- Genetic Algorithm for FS



- Particle Swarm Optimization for feature selection



Summary

Feature Manipulation – Dimensionality Reduction

- Feature Extraction
  - Unsupervised vs Supervised: PCA, ICA, Autoencoder, … vs. LDA
  - Linear vs Nonlinear: Autoencoder, T-sne, … vs. PCA, ICA, LDA
- Feature Selection
  - Univariate FS
  - Multivariate FS