MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA

INFORMATION AND COMMUNICATION TECHNOLOGIES
PhD study programme

Data Mining and Knowledge Discovery

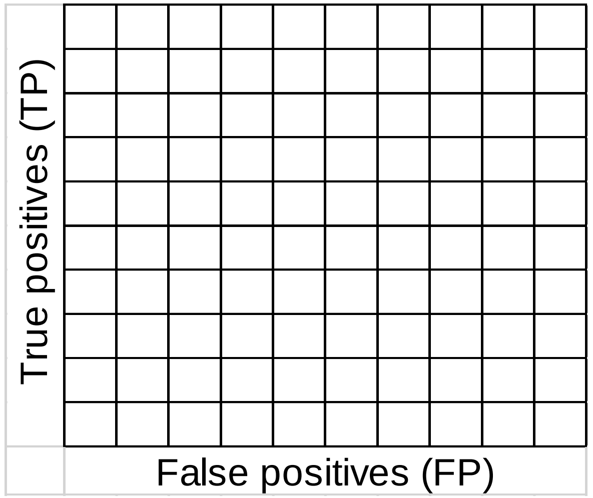
Petra Kralj Novak

December 1, 2020

http://kt.ijs.si/petra_kralj/dmkd3.html

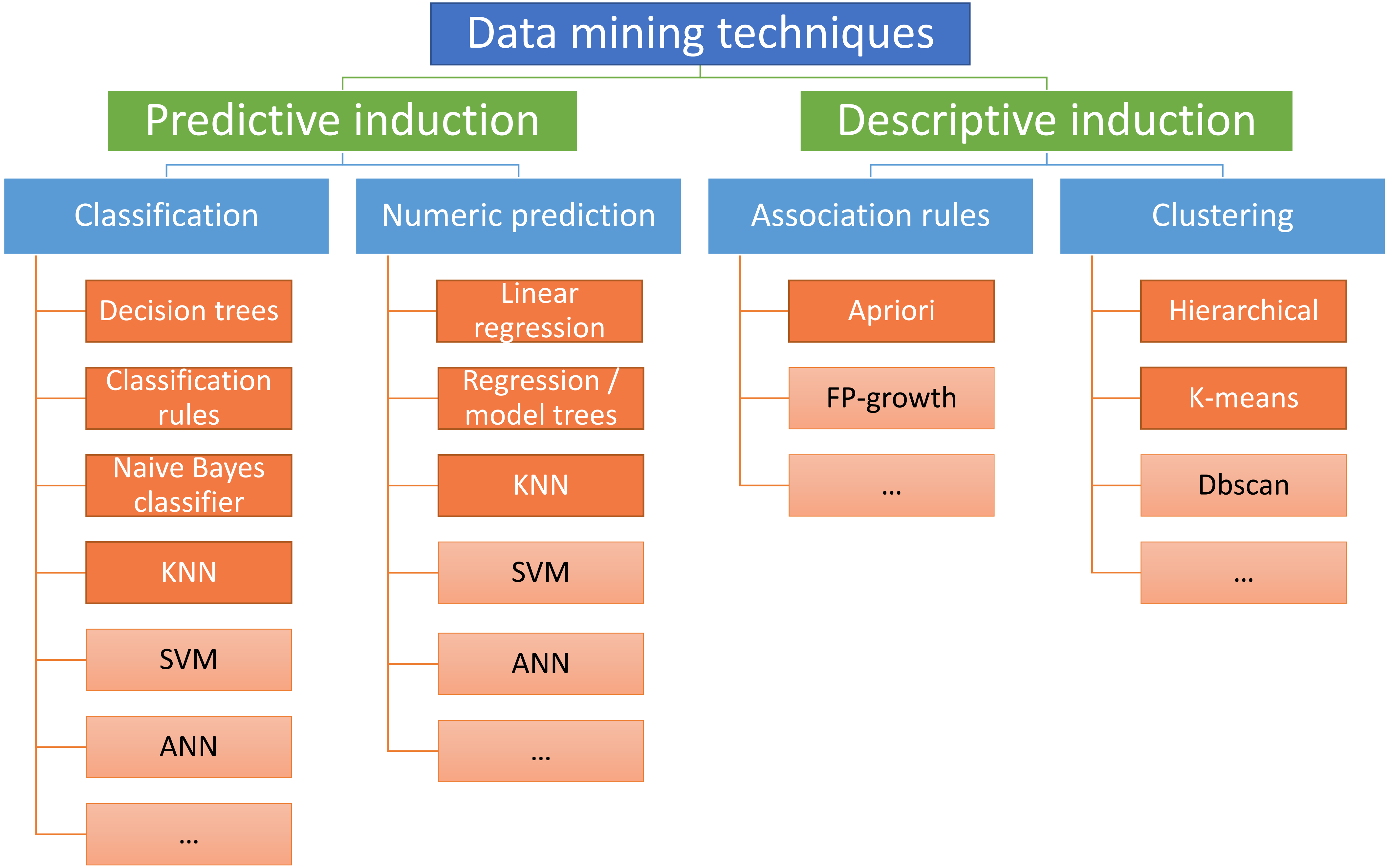
Exercise: ROC curve and AUC

	Actual class	Confidence classifier forclass Y	FP	TP	FPr	TPr
P1	Y	1				
P2	Y	1				
P3	Y	0.95				
P4	Y	0.9				
P5	Y	0.9				
P6	N	0.85				
P7	Y	0.8				
P8	Y	0.6				
P9	Y	0.55				
P10	Y	0.55				
P11	N	0.3				
P12	N	0.25				
P13	Y	0.25				
P14	N	0.2				
P15	N	0.1				
P16	N	0.1				
P17	N	0.1				
P18	N	0				
P19	N	0				
P20	N	0				



TPr = TP / All positives

FPr = FP / All negatives

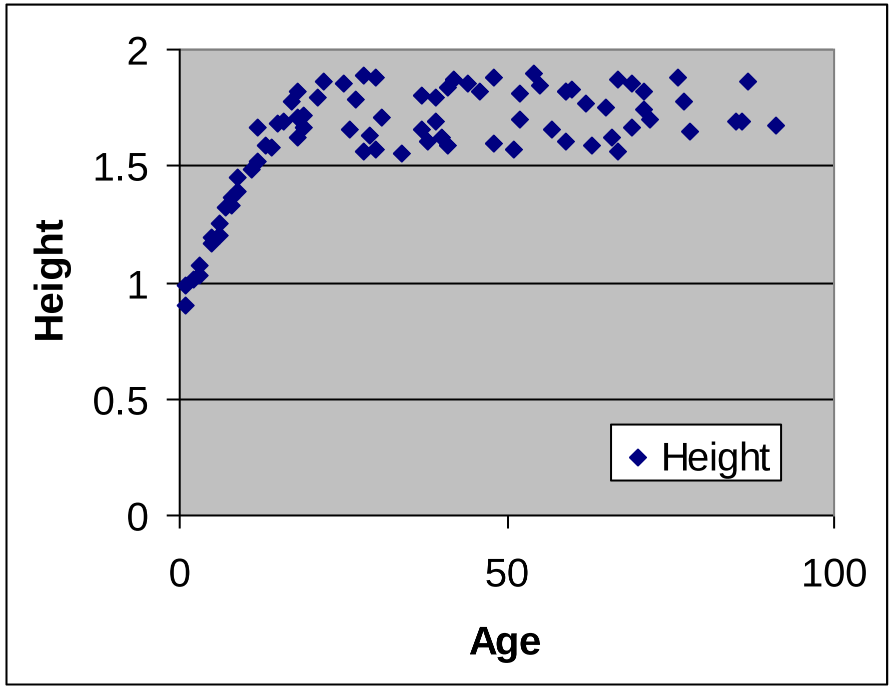

Data mining techniques
Predictive induction
Descriptive induction
Classification
Numeric prediction
Association rules
Clustering
Decision trees
Classification rules
Naive Bayes classifier
KNN
SVM
ANN
…
Linear regression
Regression / model trees
KNN
SVM
ANN
…
Apriori
FP-growth
…
Hierarchical
K-means
Dbscan
…

Numeric prediction

Regression

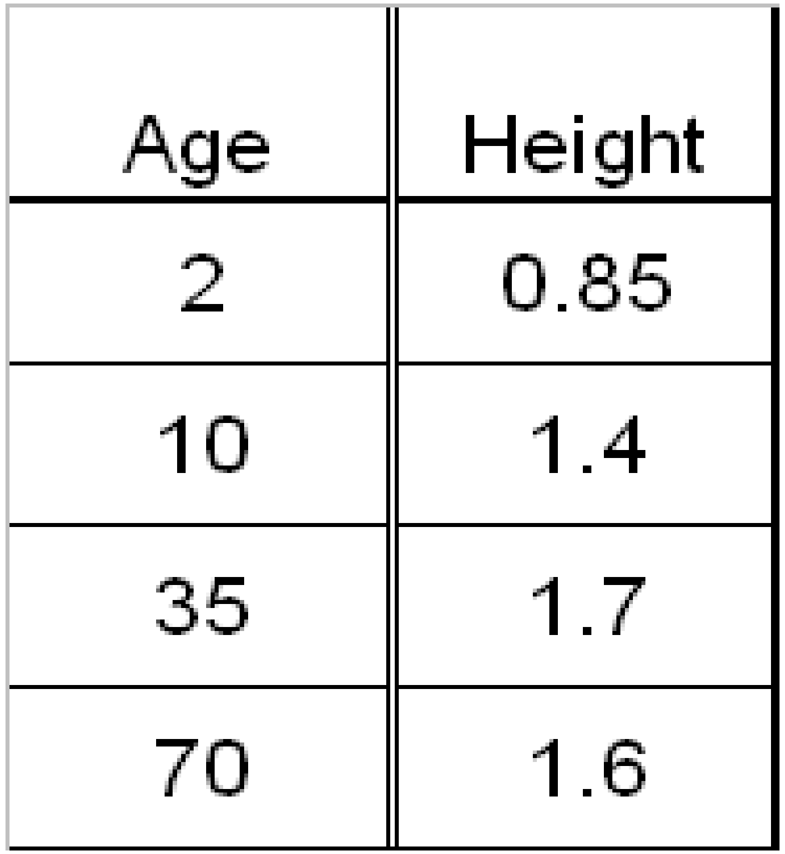
Example

- data about 80 people: Age and Height



Age	Height
3	1.03
5	1.19
6	1.26
9	1.39
15	1.69
19	1.67
22	1.86
25	1.85
41	1.59
48	1.60
54	1.90
71	1.82
...	...

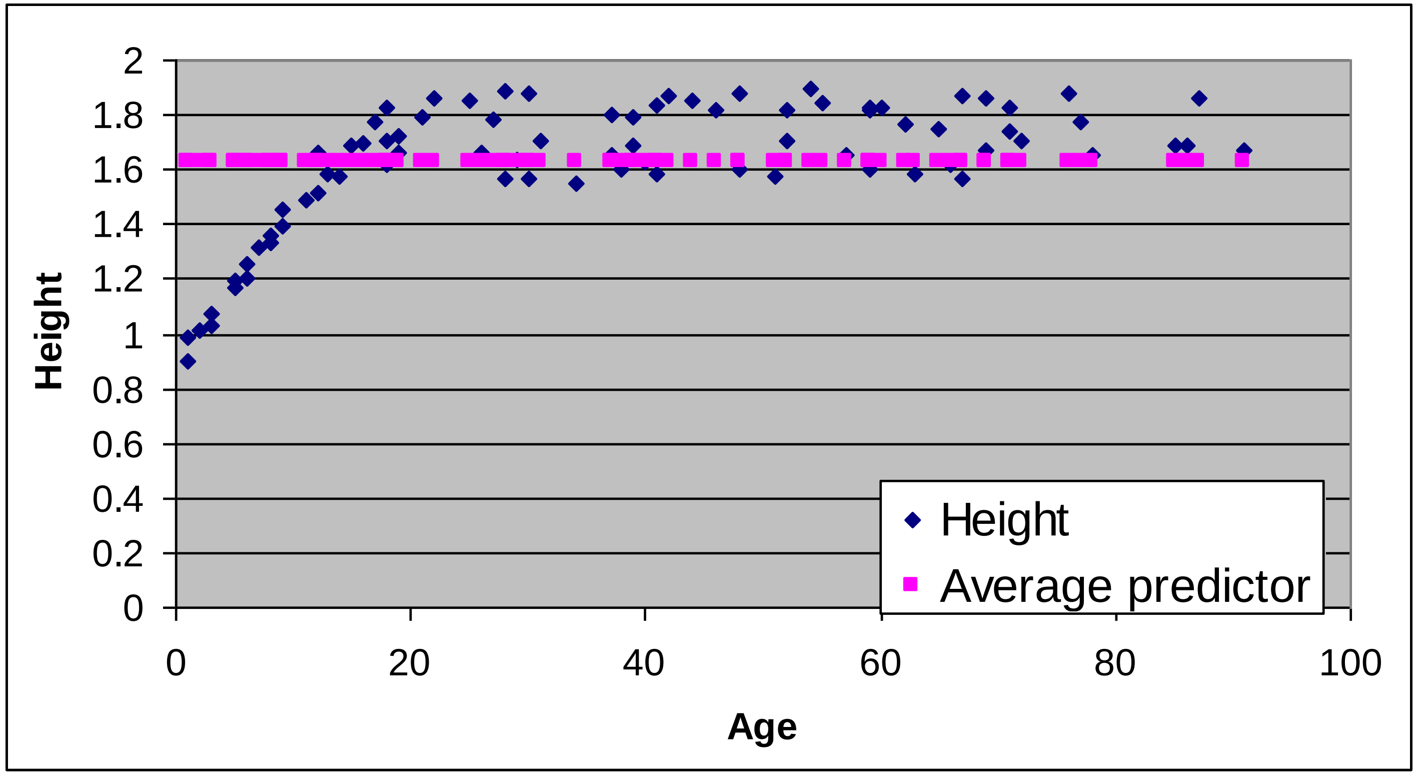
Test set



Age	Height
2	0.85
10	1.4
35	1.7
70	1.6

Baseline numeric predictor

- Average of the target variable



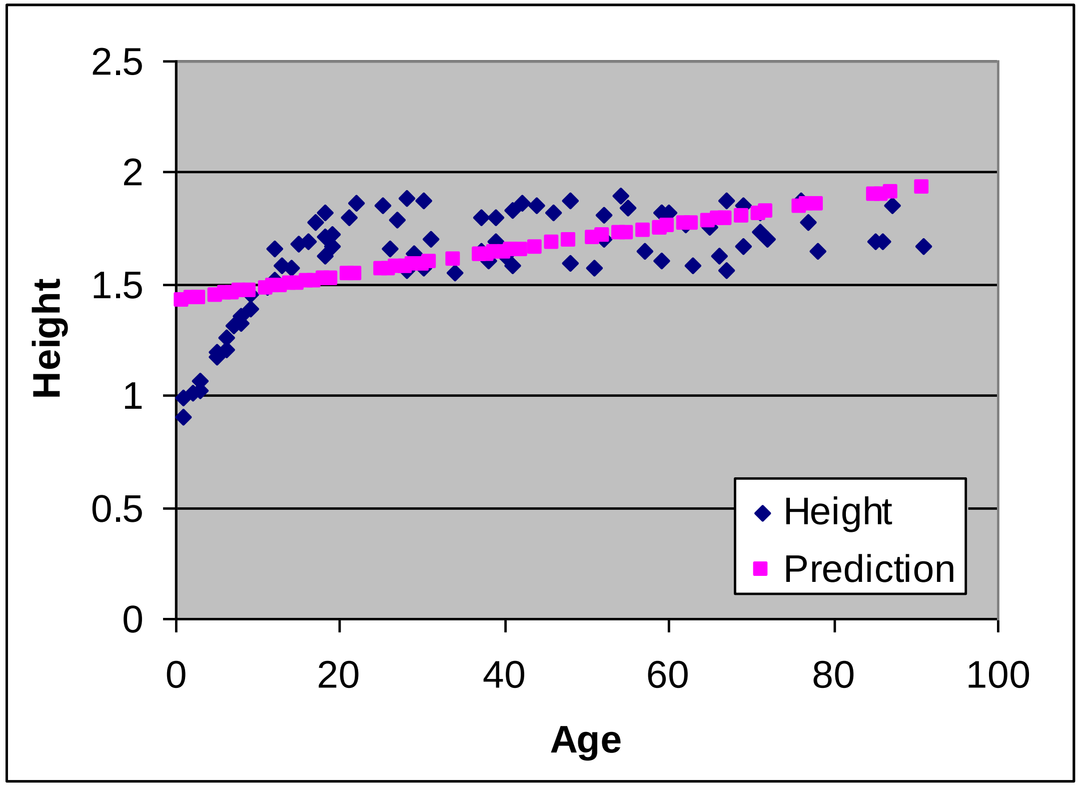
Baseline predictor: prediction

Average of the target variable is 1.63

Age	Height	Baseline
2	0.85	
10	1.4	
35	1.7	
70	1.6	

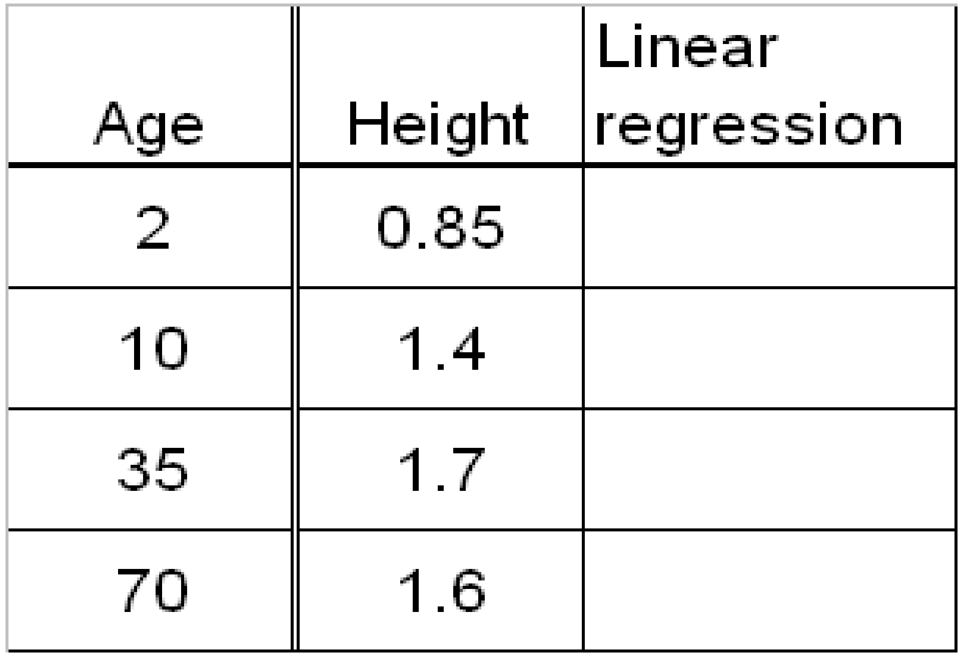
Linear Regression Model

Height = 0.0056 * Age + 1.4181



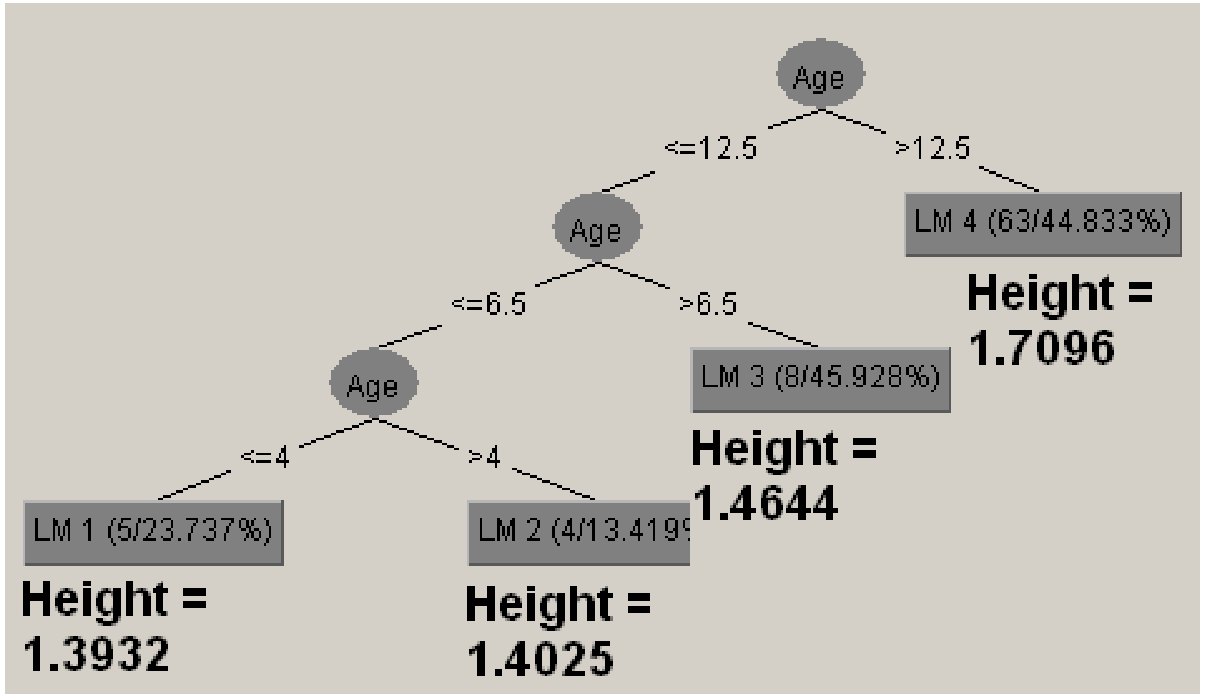
Linear Regression: prediction

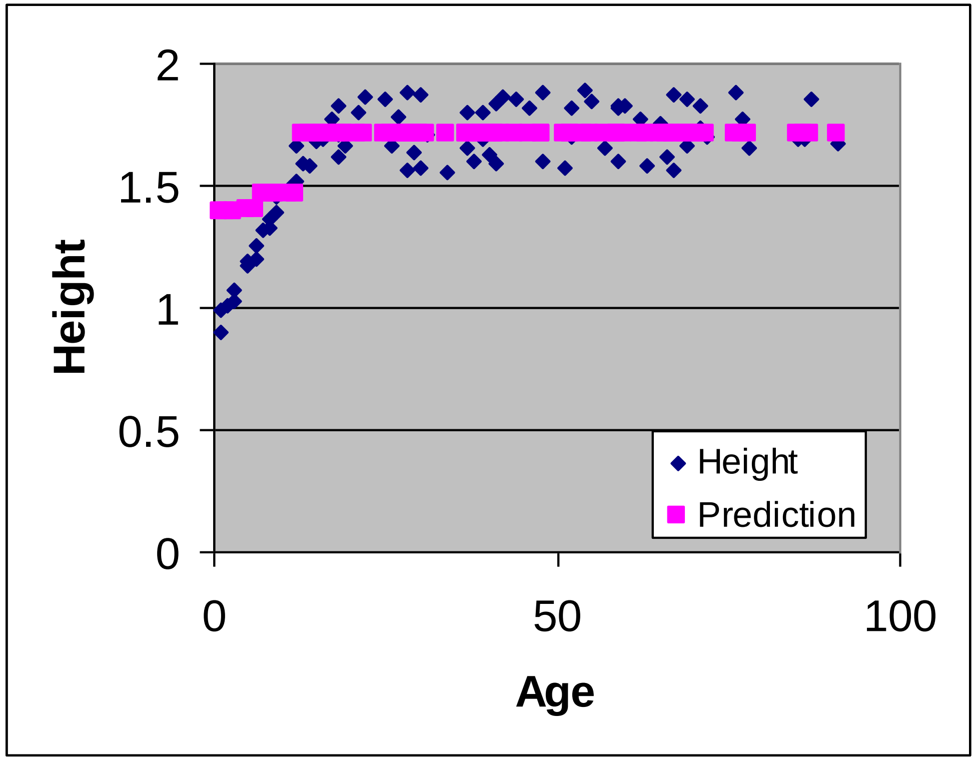
Height = 0.0056 * Age + 1.4181



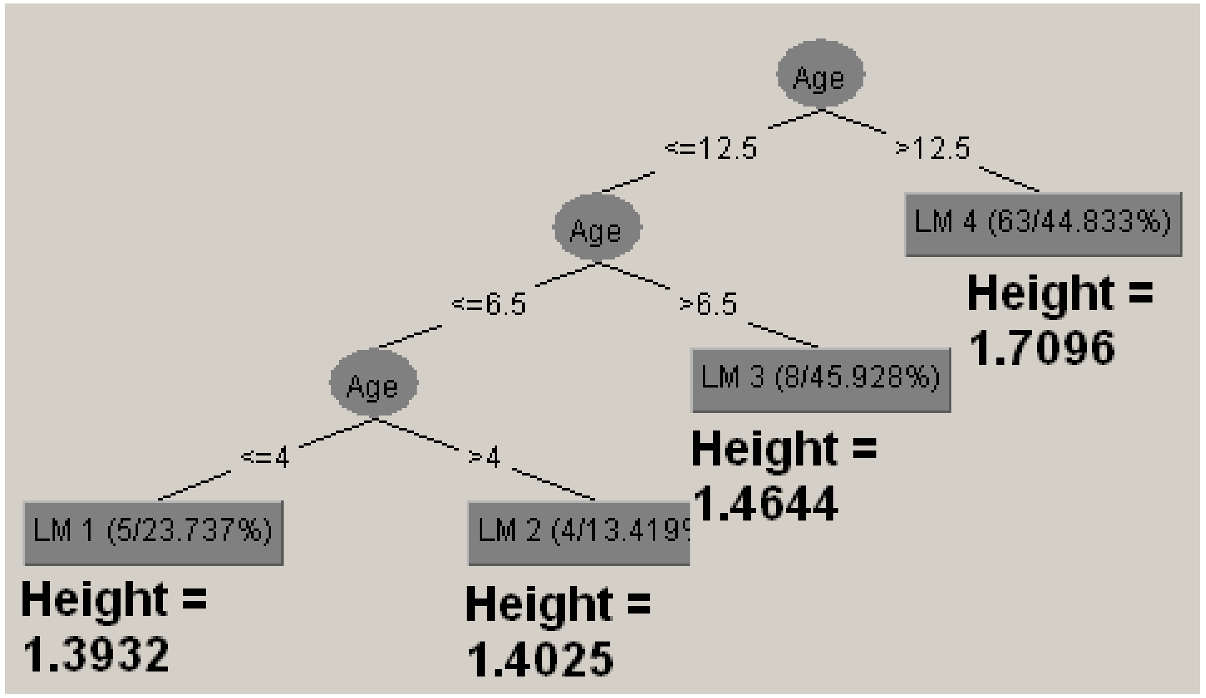
Age	Height	Linear regression
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Regression tree



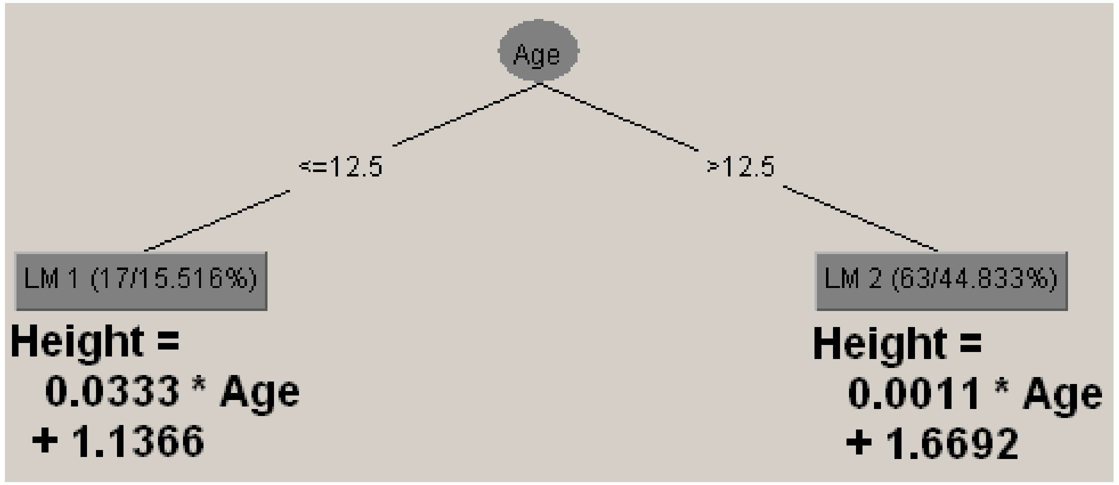


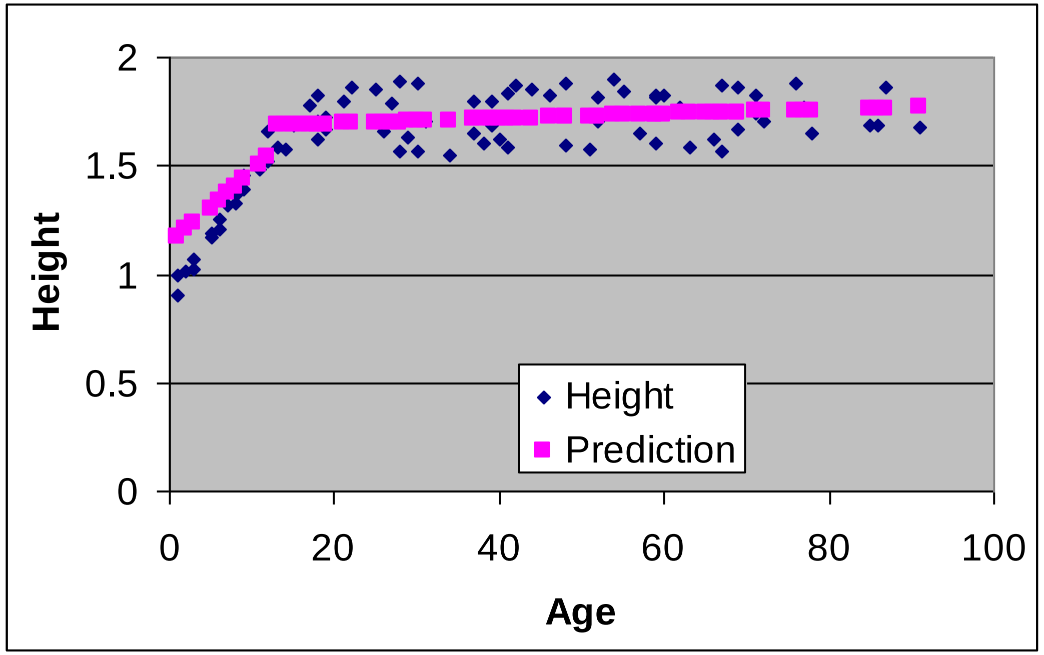
Regression tree: prediction



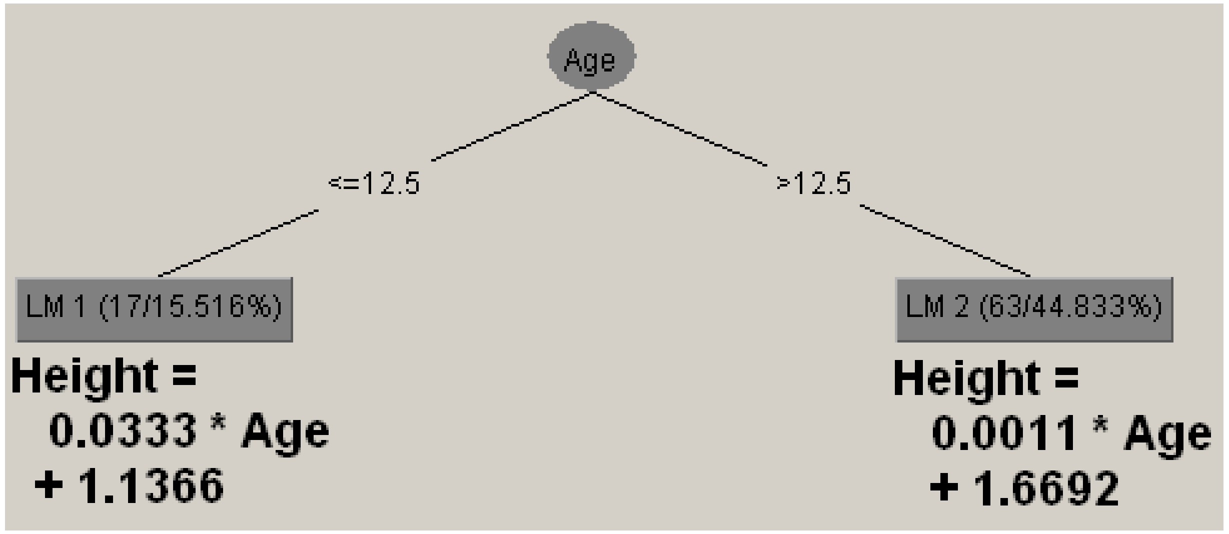
Age	Height	Regression tree
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Model tree


Age
<=12.5
>12.5
LM 1 (17/15.516%)
Height = 0.0333 * Age + 1.1366
LM 2 (63/44.833%)
Height = 0.0011 * Age + 1.6692


Height
Age
2
1.5
1
0.5
0
0
20
40
60
80
100
◆ Height
■ Prediction

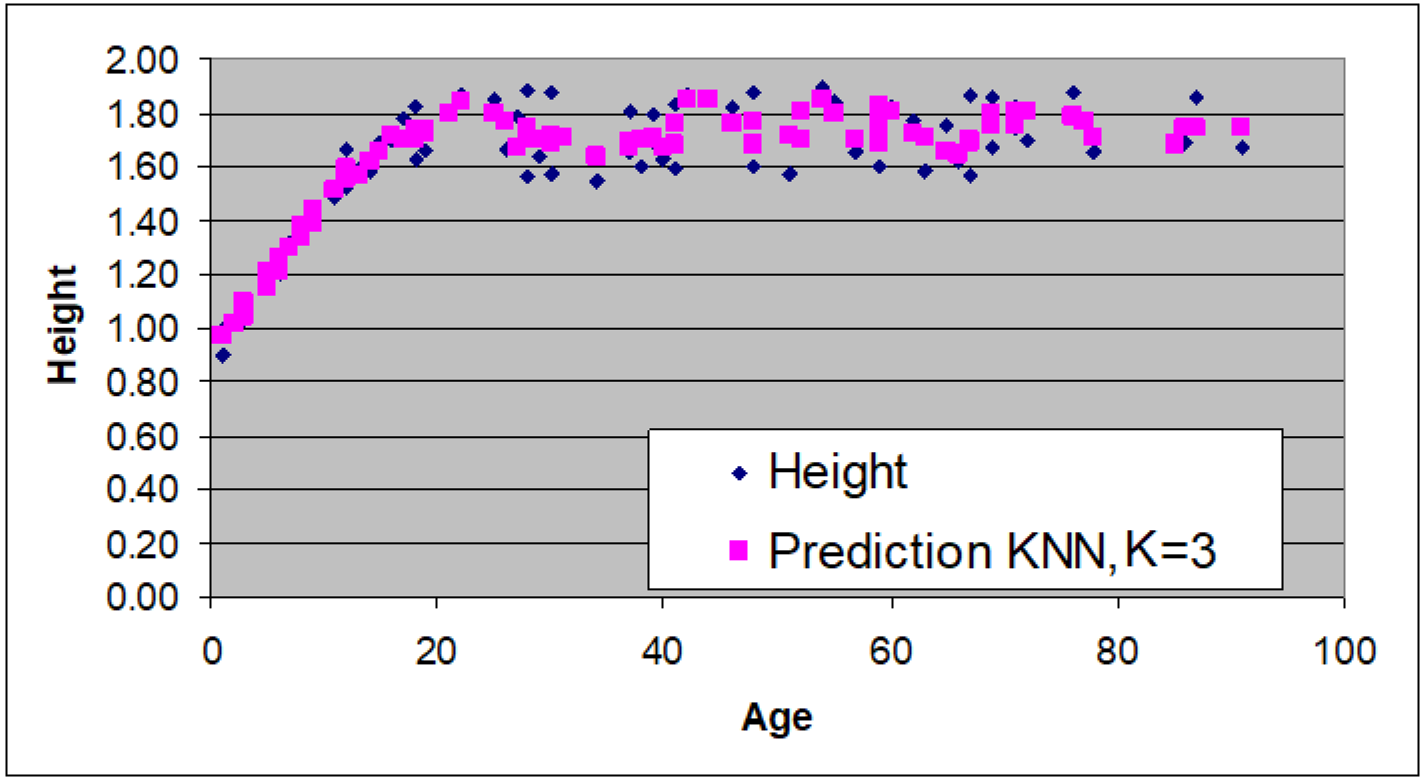
Model tree: prediction


Age
<=12.5
>12.5
LM 1 (17/15.516%)
LM 2 (63/44.833%)
Height = 0.0333 * Age + 1.1366
Height = 0.0011 * Age + 1.6692

Age	Height	Model tree
2	0.85	
10	1.4	
35	1.7	
70	1.6	

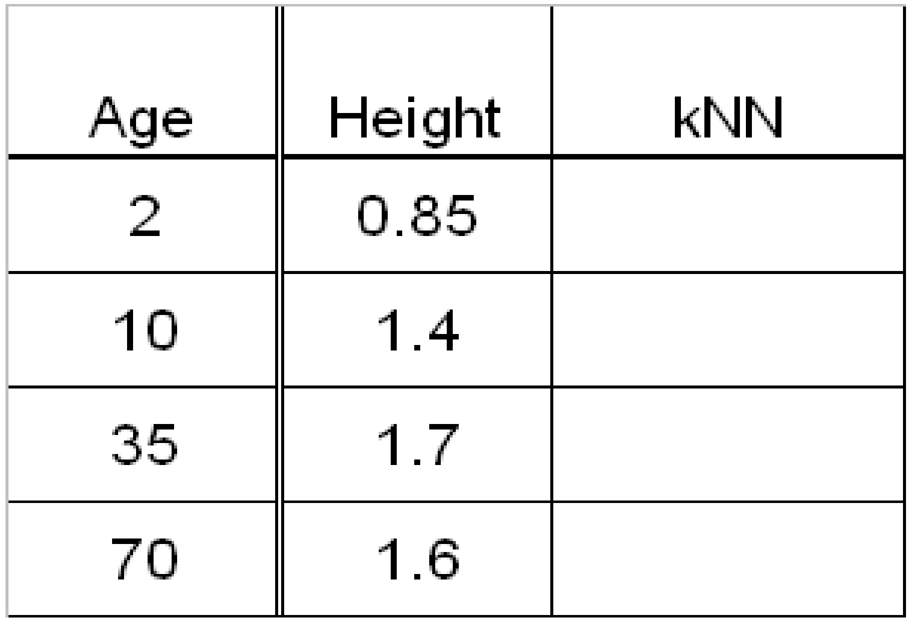
KNN – K nearest neighbors

- Looks at K closest examples (by non-target attributes) and predicts the average of their target variable
- In this example, K=3



KNN prediction

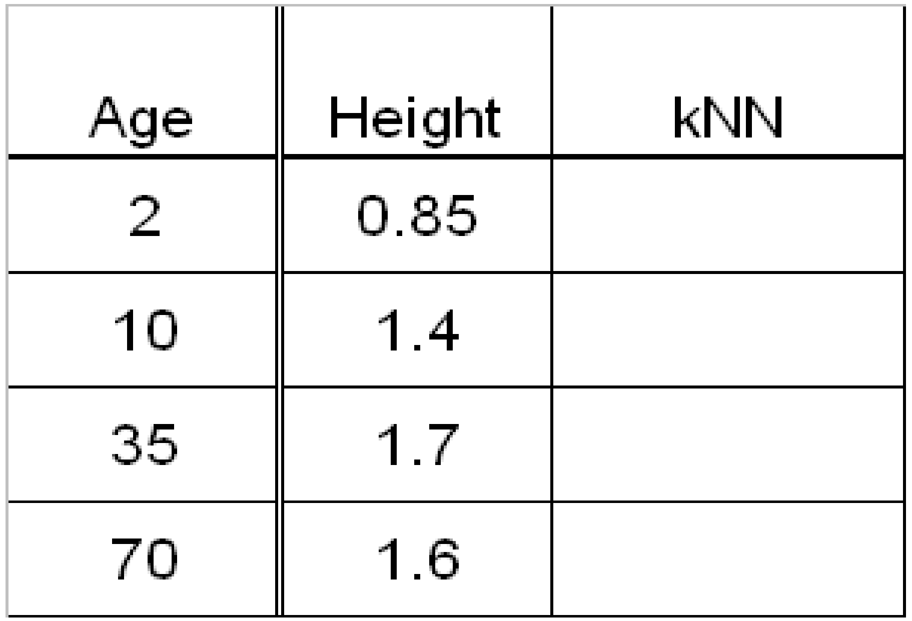
Age	Height
1	0.90
1	0.99
2	1.01
3	1.03
3	1.07
5	1.19
5	1.17



Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

KNN prediction

Age	Height
8	1.36
8	1.33
9	1.45
9	1.39
11	1.49
12	1.66
12	1.52
13	1.59
14	1.58



Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

KNN prediction

Age	Height
30	1.57
30	1.88
31	1.71
34	1.55
37	1.65
37	1.80
38	1.60
39	1.69
39	1.80

Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

KNN prediction

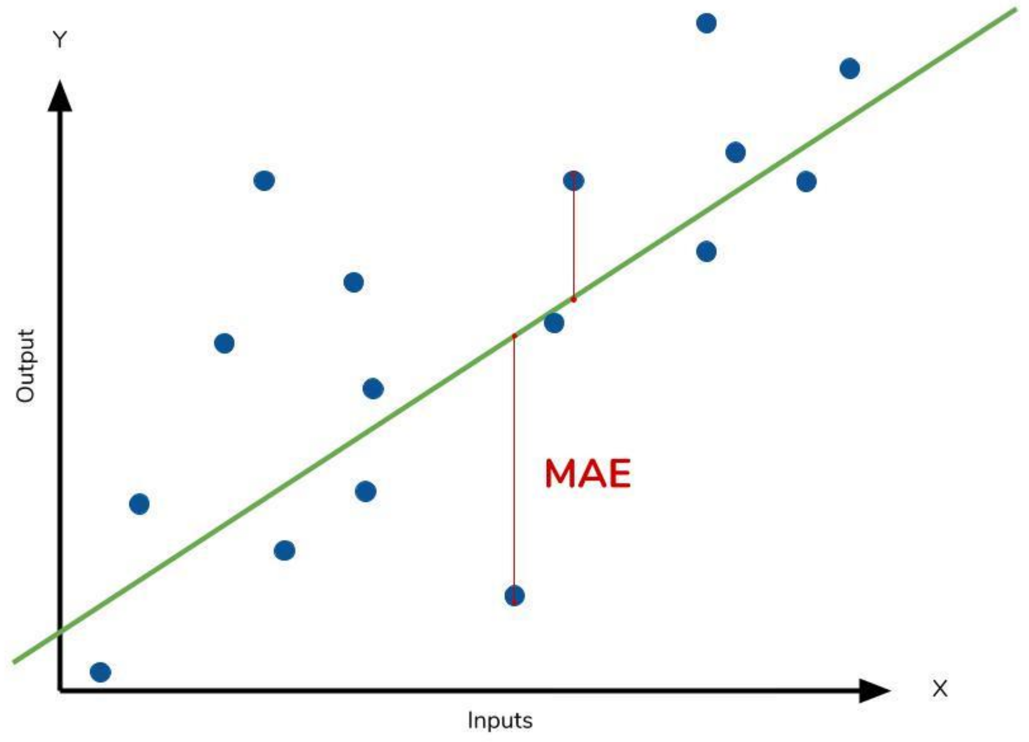
Age	Height
67	1.56
67	1.87
69	1.67
69	1.86
71	1.74
71	1.82
72	1.70
76	1.88

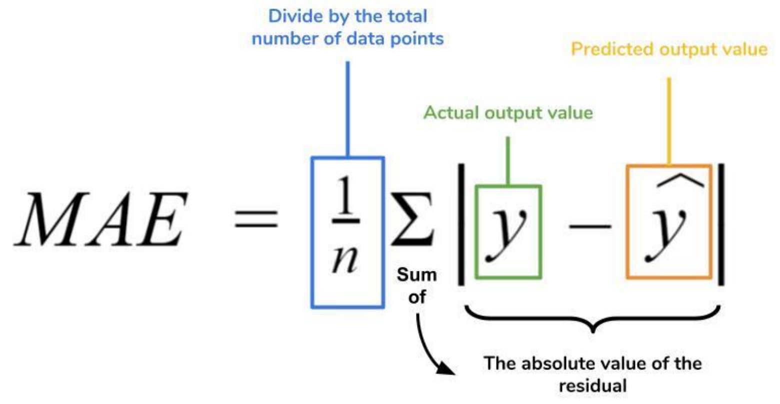
Age	Height	kNN
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Which predictor is the best?

Age	Height	Baseline	Linear regression	Regressi on tree	Model tree	kNN
2	0.85	1.63	1.43	1.39	1.20	1.00
10	1.4	1.63	1.47	1.46	1.47	1.44
35	1.7	1.63	1.61	1.71	1.71	1.67
70	1.6	1.63	1.81	1.71	1.75	1.77

MAE: Mean absolute error

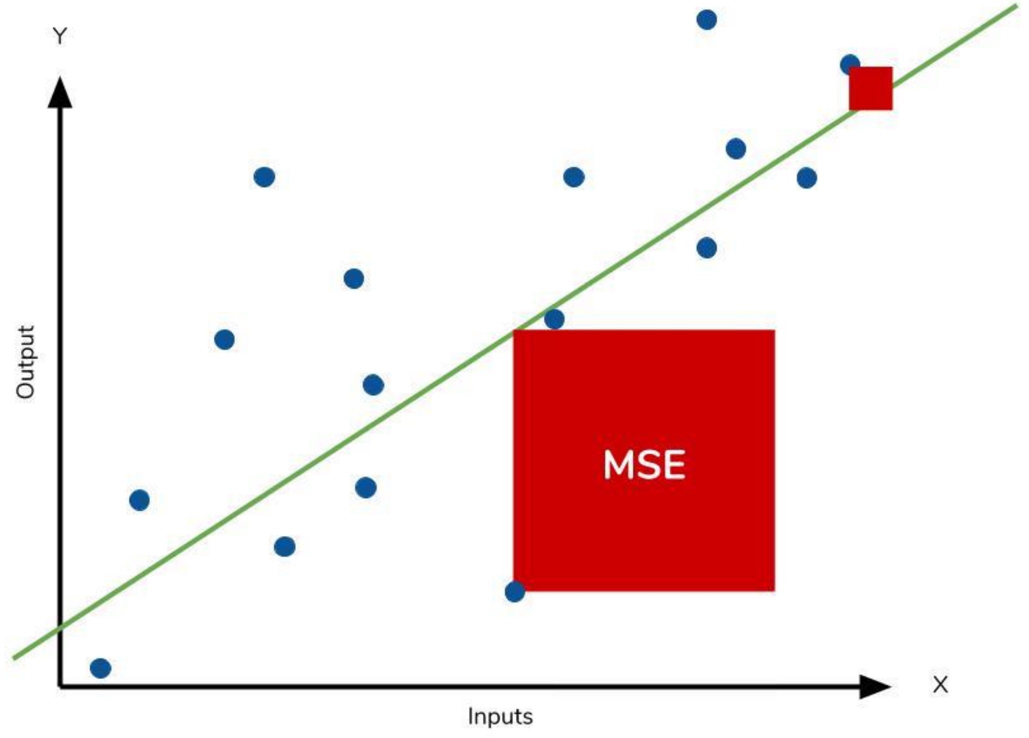



Divide by the total number of data points
Predicted output value
Actual output value
Sum of
The absolute value of the residual

$$MAE = \boxed{\frac{1}{n}} \sum \left| \boxed{y} - \boxed{\hat{y}} \right|$$

The average difference between the predicted and the actual values.
The units are the same as the unites in the target variable.

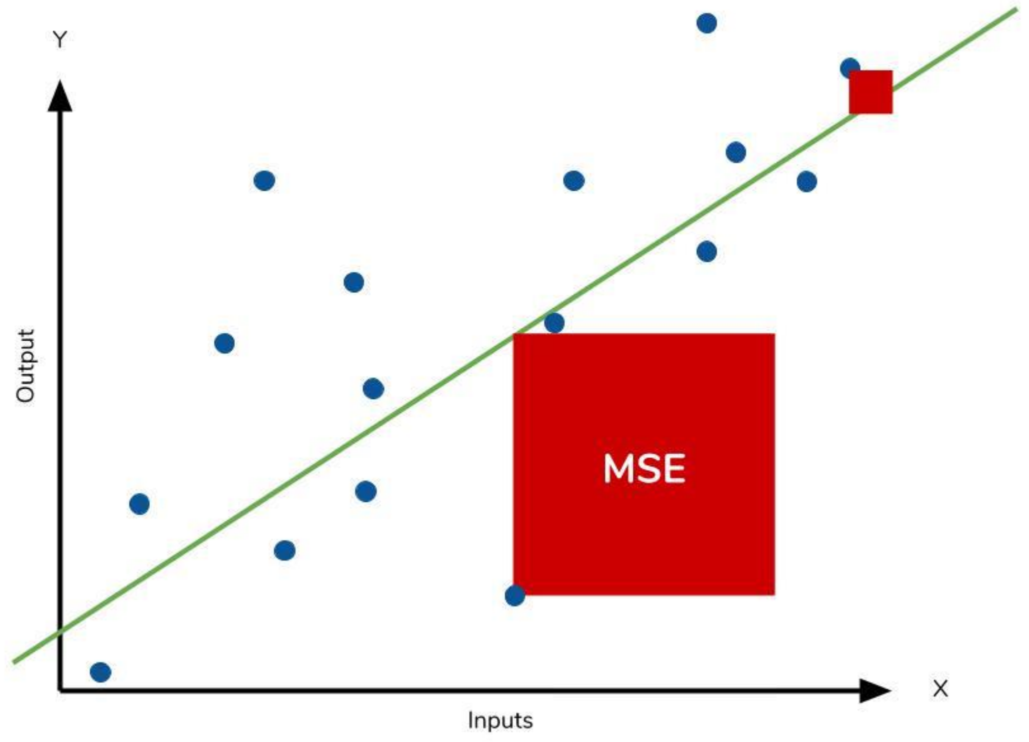
MSE: Mean squared error



$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \widehat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

Mean squared error measures the average squared difference between the estimated values and the actual value.
Weights large errors more heavily than small ones.
The units of the errors are squared.

RMSE: Root mean square error

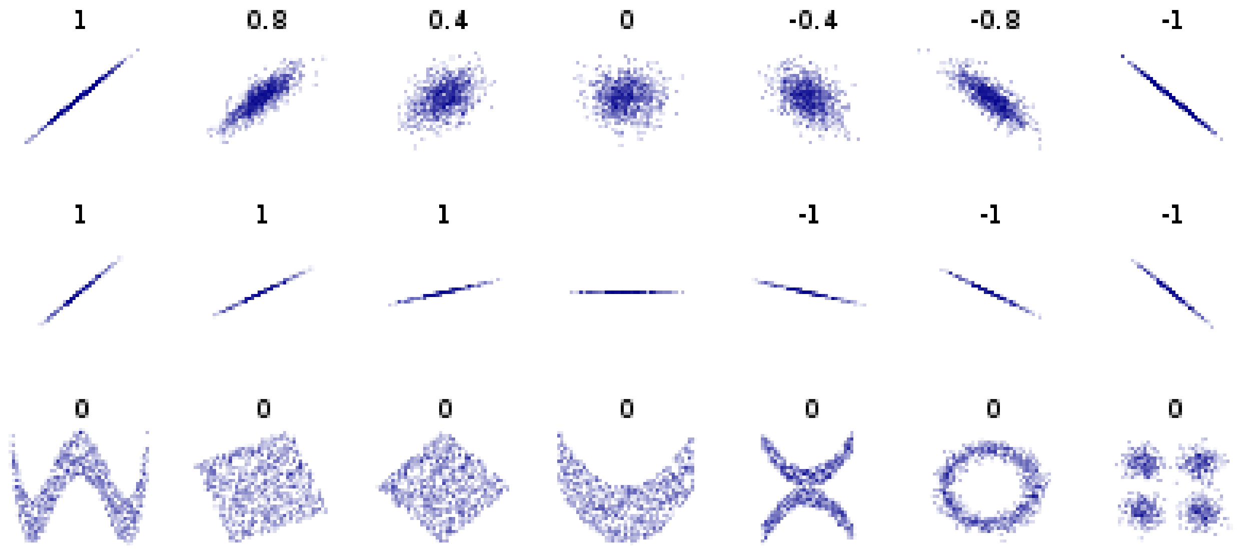


$$RMSE = \sqrt{MSE}$$

Taking the square root of MSE yields the root-mean-square error (RMSE), which has the same units as the quantity being estimated.

Correlation coefficient

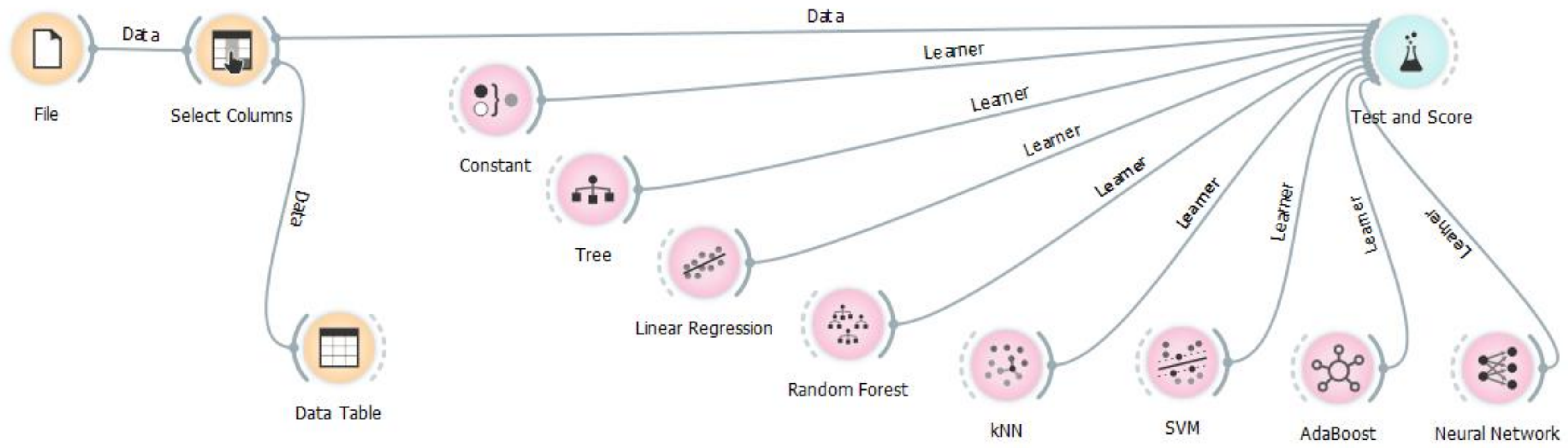
- Pearson correlation coefficient is a statistical formula that measures the strength between variables and relationships.



Similar to confusion matrix in the classification case.
No unit.

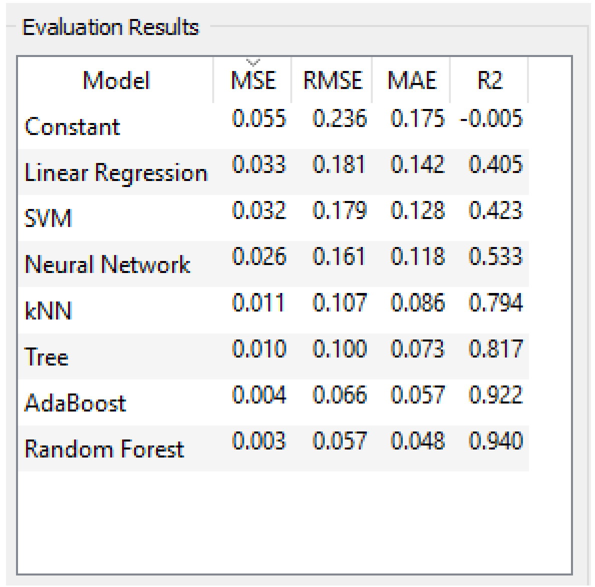
Numeric prediction in Orange

Models



Metrics

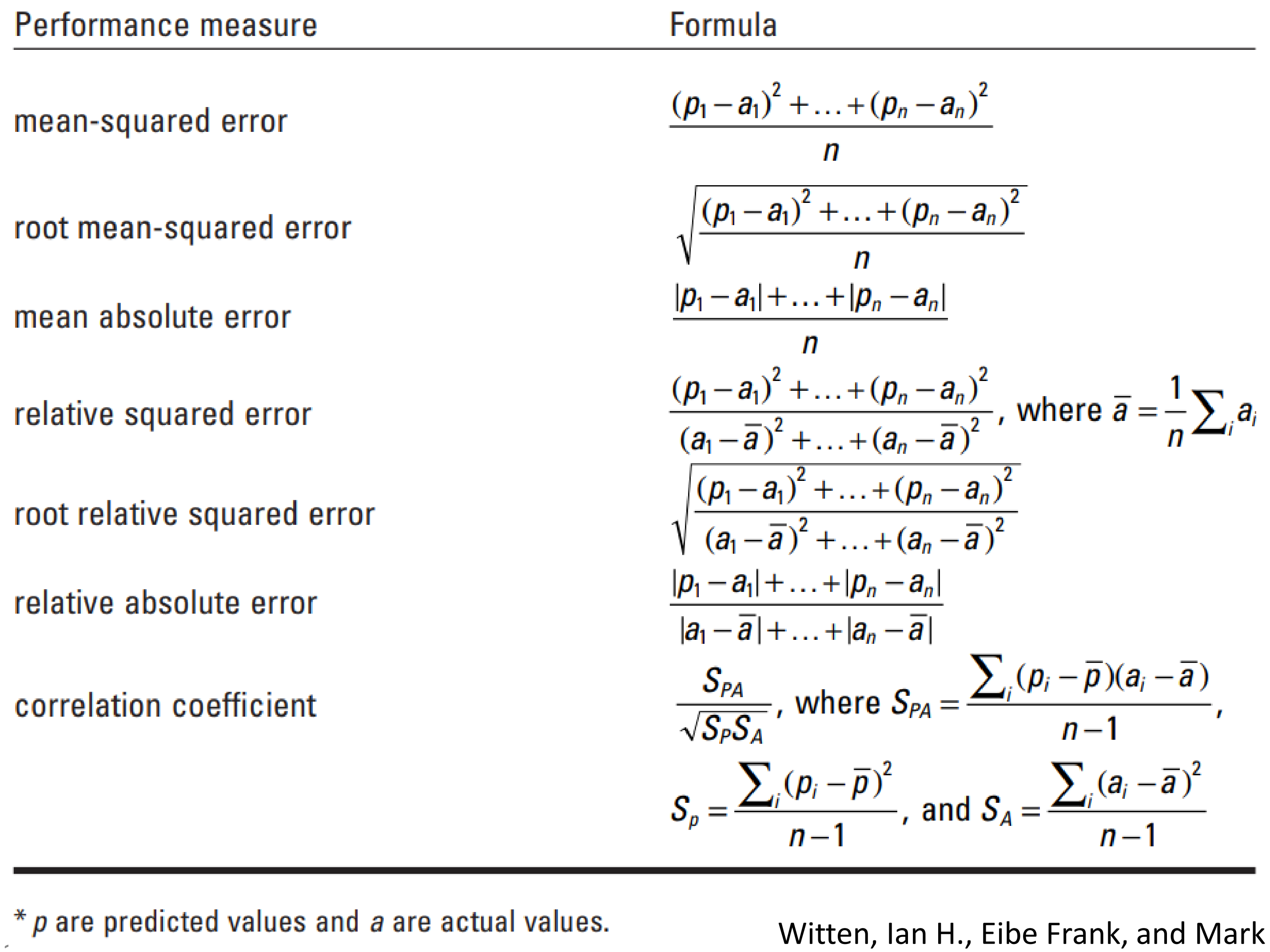
- MSE – mean squared error
- RMSE – root mean squared error
- MAE – mean absolute error
- R^2 – correlation coefficient


Evaluation Results

Model	MSE	RMSE	MAE	R2
Constant	0.055	0.236	0.175	-0.005
Linear Regression	0.033	0.181	0.142	0.405
SVM	0.032	0.179	0.128	0.423
Neural Network	0.026	0.161	0.118	0.533
kNN	0.011	0.107	0.086	0.794
Tree	0.010	0.100	0.073	0.817
AdaBoost	0.004	0.066	0.057	0.922
Random Forest	0.003	0.057	0.048	0.940

Numeric prediction	**Classification**
Data: attribute-value description	
Target variable: Continuous	**Target variable**: Categorical (nominal)
Evaluation: cross validation, separate test set, …	
Error: MSE, MAE, RMSE, …	**Error**: 1-accuracy
Algorithms: Linear regression, regression trees,…	**Algorithms**: Decision trees, Naïve Bayes, …
Baseline predictor: Mean of the target variable	**Baseline predictor**: Majority class

Performance measures for numeric prediction



Performance measure	Formula
mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i(p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i(p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i(a_i-\bar{a})^2}{n-1}$

* *p* are predicted values and *a* are actual values.

Witten, Ian H., Eibe Frank, and Mark A. Hall. "Practical machine learning tools and techniques." *Morgan Kaufmann* (2005): 578. pg. 178

Relative squared error

“The error is made relative to what it would have been if a simple predictor had been used. The simple predictor in question is just the average of the actual values from the **training** data. Thus relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the default predictor.”

Regression in scikit … 4_regression.py

```
import pandas as pd
from sklearn import dummy
from sklearn import linear_model
from sklearn import tree
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
from sklearn import metrics

print("_______________________________________________________________________________")
print("Regression models, train-test validation on regressionAgeHeight.csv. ")
print("_______________________________________________________________________________")

print(""" Load the data """)
csvFileName = r"./Datasets/regressionAgeHeight.csv"
df = pd.read_csv(csvFileName)
print(df.head())
print("data shape: ", df.shape)

feature_cols = ['Age']
target_var = 'Height'

X = df[feature_cols].values
y = df[target_var].values

""" Train-test split """
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

Regression in scikit … 4_regression.py

```python
""" Initialize the learners """
dummy = dummy.DummyRegressor()
regr = linear_model.LinearRegression()
reg_tree = tree.DecisionTreeRegressor(min_samples_leaf=8)
knn = KNeighborsRegressor(n_neighbors=2)

learner = reg_tree

"""" Train and apply """
learner.fit(X_train, y_train)
y_pred = learner.predict(X_test)

print ("\n Actual   Predicted")
for i in range(len(y_test)):
     print("{0:6.2f}  {1:8.2f}".format(y_test[i], y_pred[i]))

print("Performance:")
print("MAE  \t{0:5.2f}".format( metrics.mean_absolute_error(y_test,y_pred)))
print("MSE  \t{0:5.2f}".format( metrics.mean_squared_error(y_test,y_pred)))
print("R2   \t{0:5.2f}".format( metrics.r2_score(y_test,y_pred)))
```

Exercise: RRSE

- Use SciKit to compute RRSE of regression models
- RRSE = root relative squared error
    - Nominator: sum of squared differences between the actual and the expected values
    - Denominator: sum of squared errors (the sum of the squared differences between each observation and its group's mean)

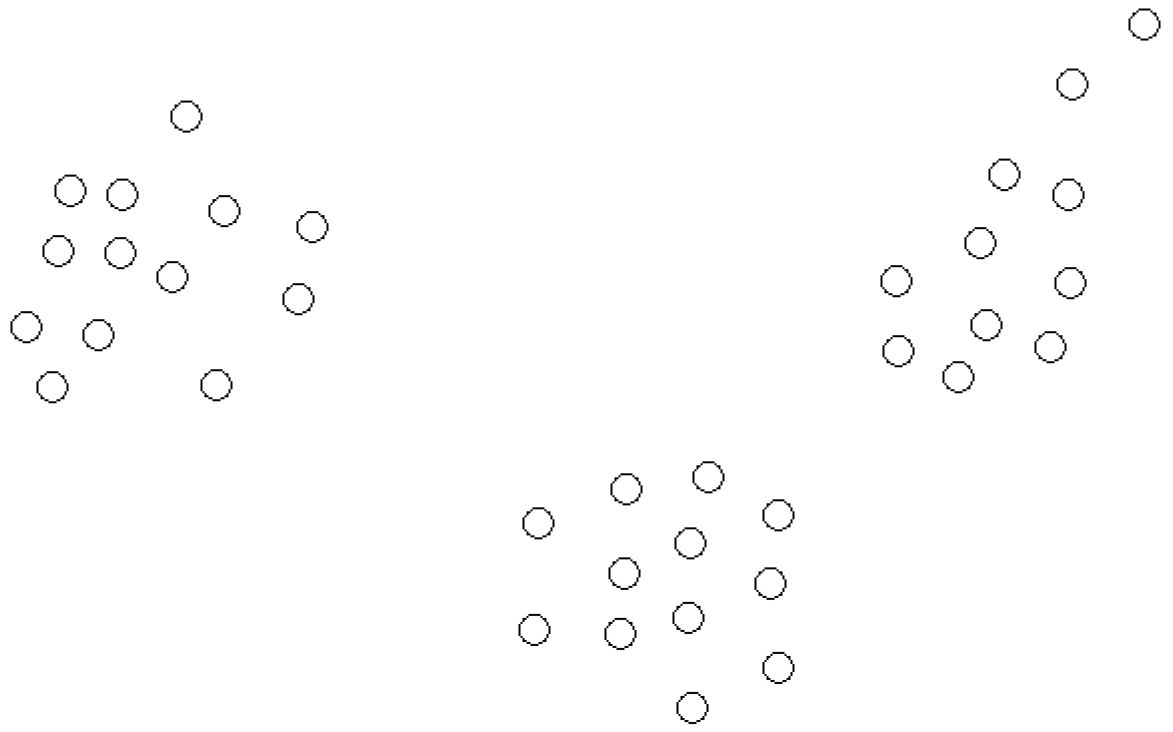
$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n}(\bar{a} - a_i)^2}}$$

p – predicted, a – actual, ā – the mean of the actual

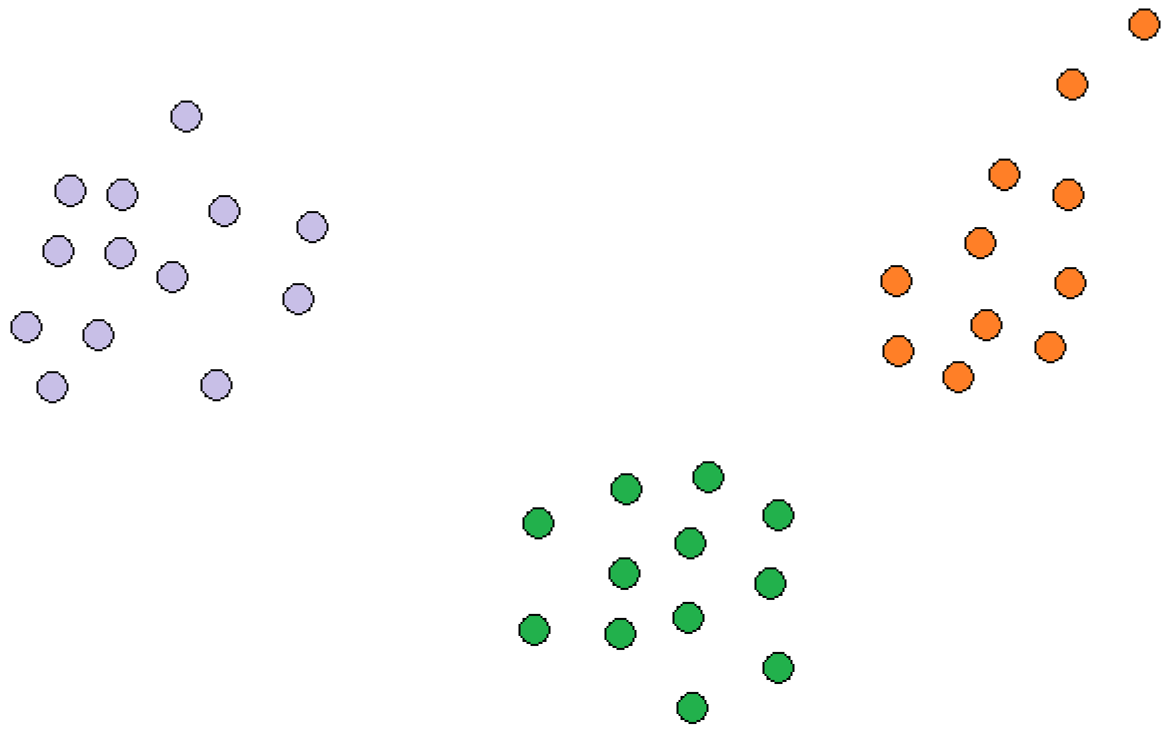
- RRSE: Ratio between the error of the model and the error of the naïve model (predicting the average)

Clustering

Clustering



Clustering

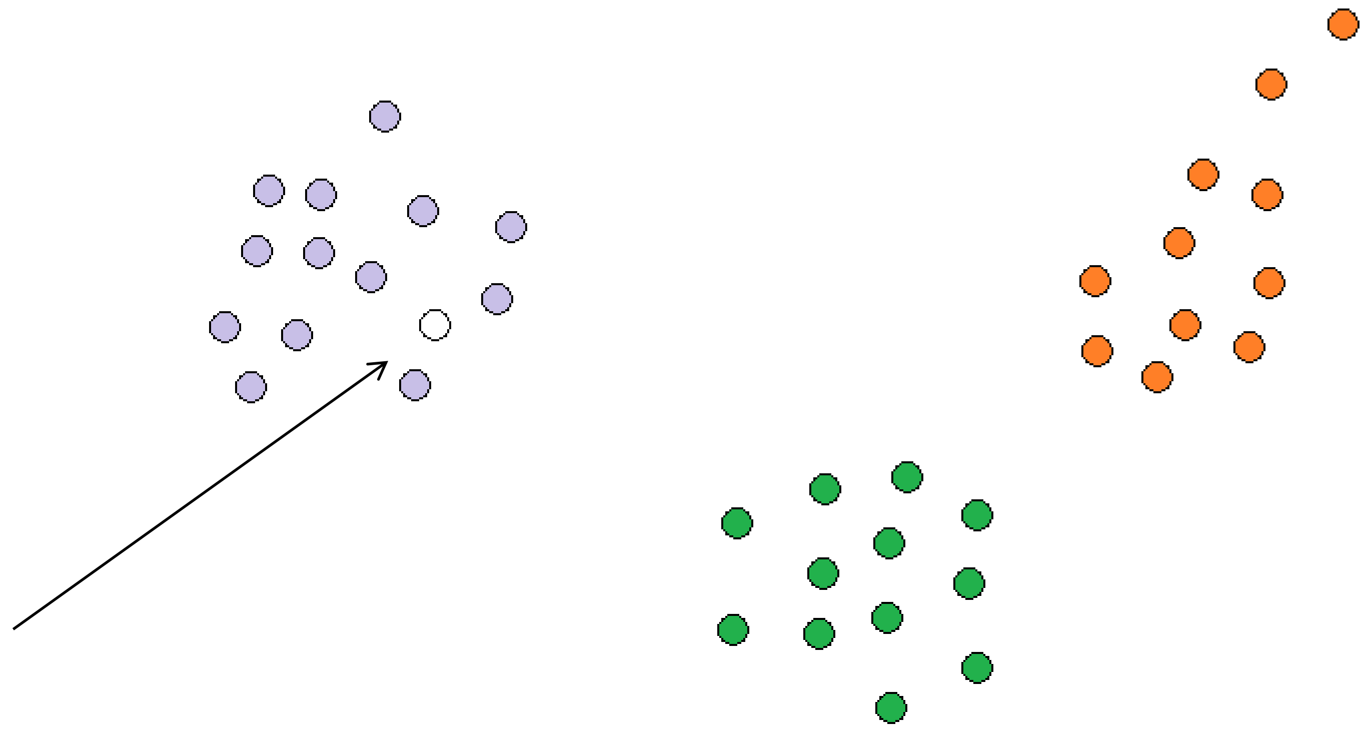


Clustering

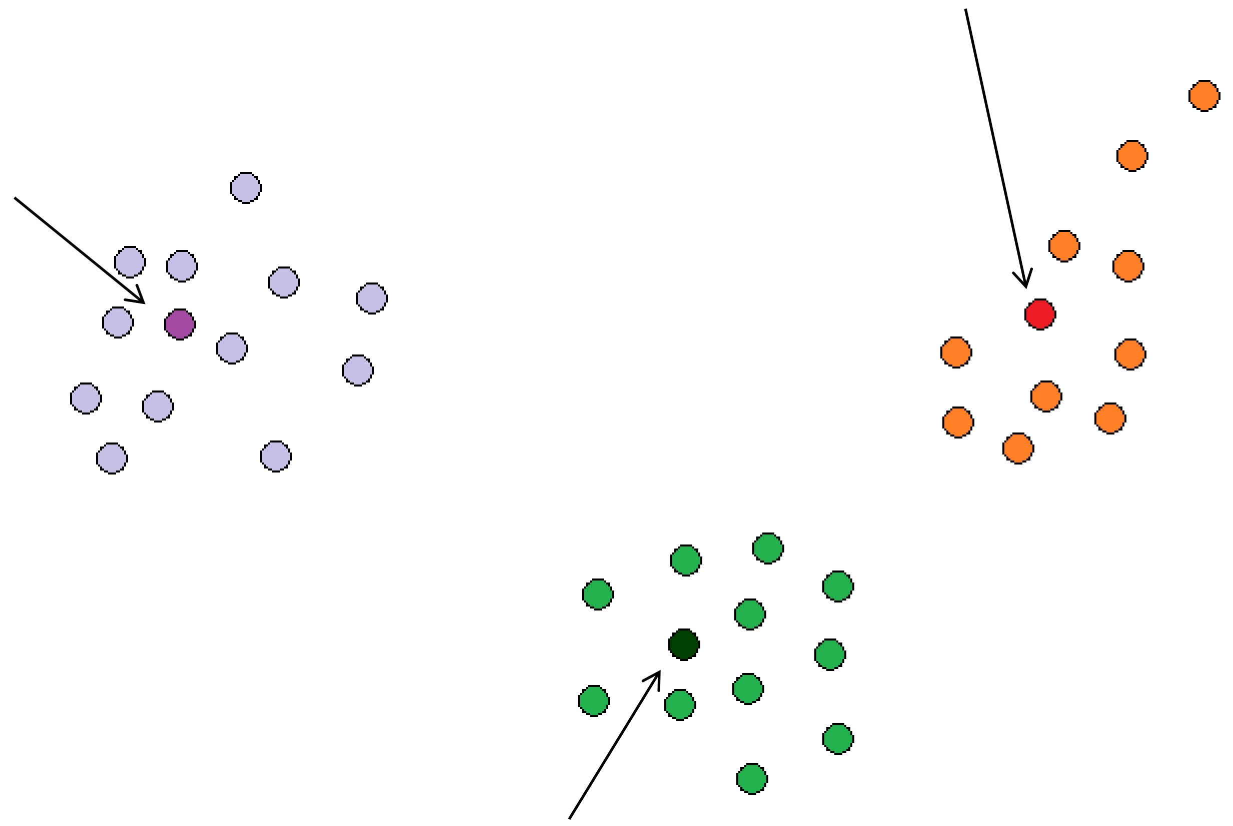
- … is the process of grouping the data instances into clusters so that objects within a cluster have high similarity but are very dissimilar to objects in other clusters.

- Wish list:
  - Identity clusters irrespective of their shapes
  - Scalability
  - Ability to deal with noisy data
  - Insensitivity to the order of input records

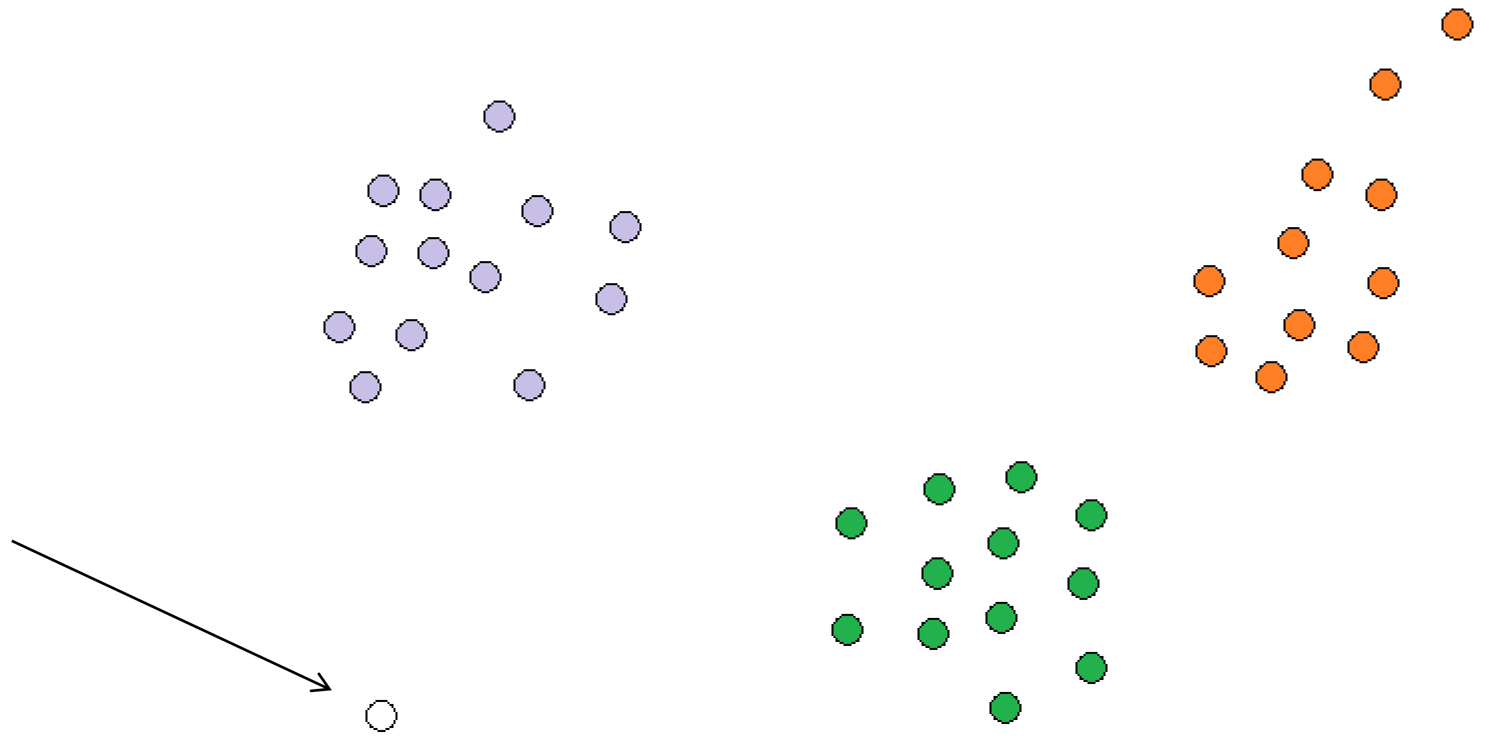
Unsupervised classification



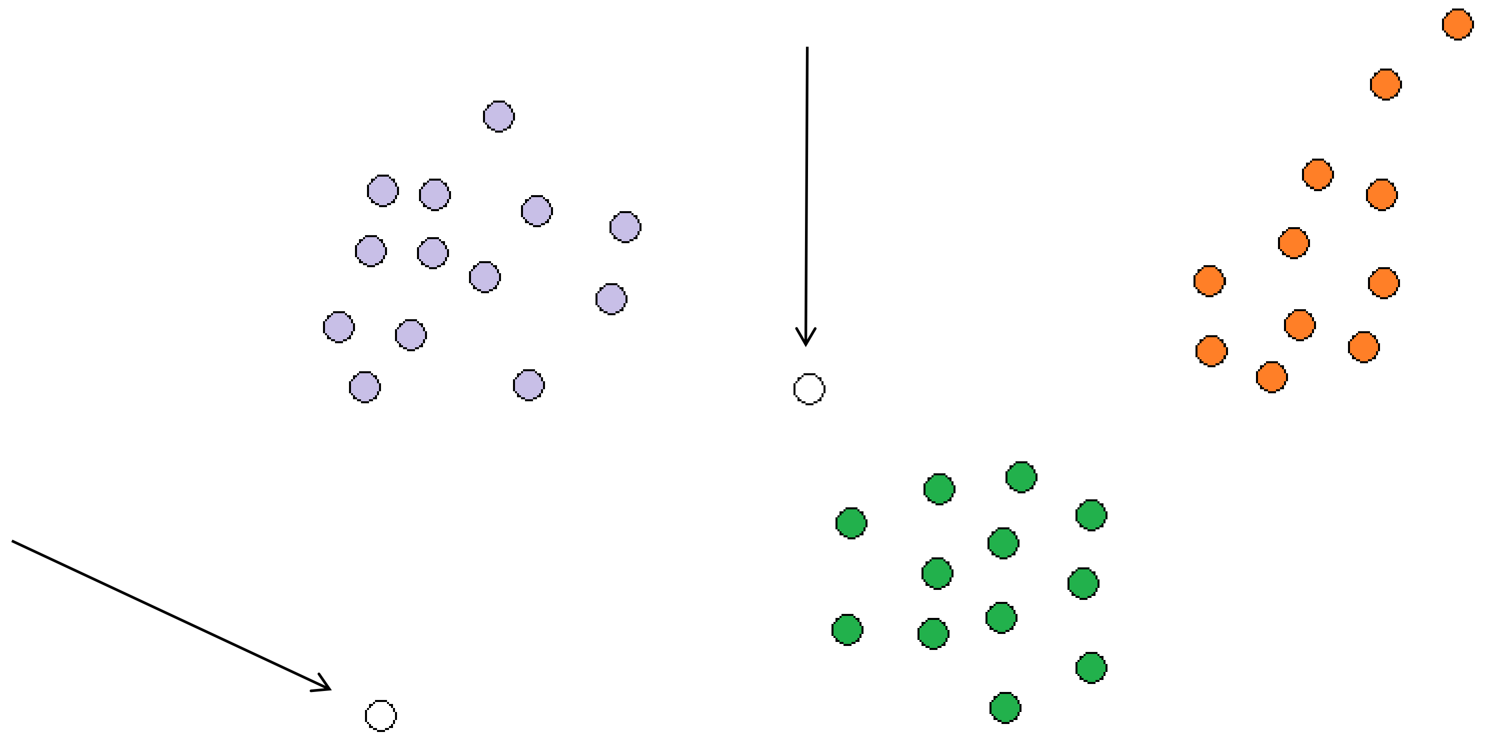
Data summarization: centroid, medoid



Outlier detection



Outlier detection



Applications

- Data mining
  - Unsupervised classification
  - Data summarization
  - Outlier analysis
  - …
- Customer segmentation and collaborative filtering
- Text applications
- Social network analysis

Clustering types

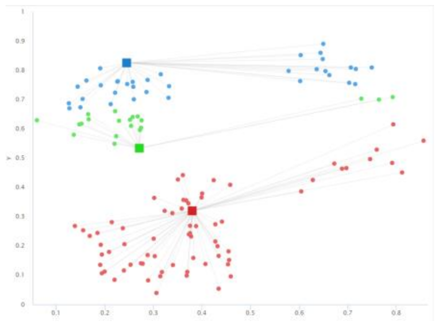
- **Partitioning**
  - k-means, k-medoids, k-modes
- **Hierarchical**
  - Agglomerative
- Grid-based
  - Multi-resolution grid structure
  - Efficient and scalable
- Density-based
  - A cluster is a dense region of points, which is separated by low density regions, from other regions of high density
  - Algorithms: DBSCAN, OPTICS, DenClue

Interactive k-Means (Educational)

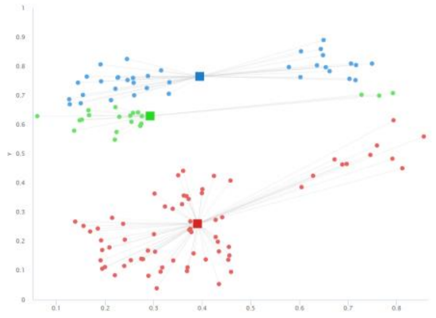
Interactive k-Means

K-Means example

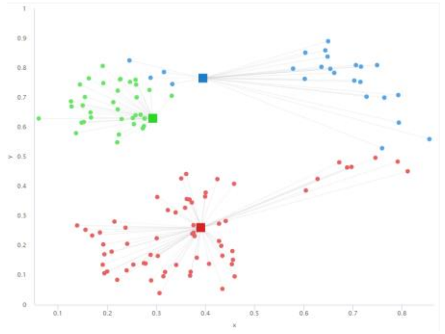
Random initialization



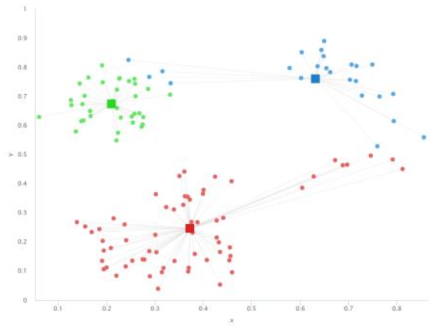
Centroid computation



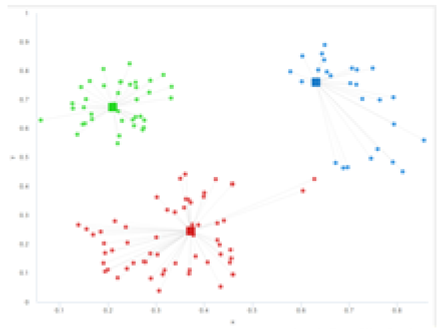
Assignment of points to the nearest centroid



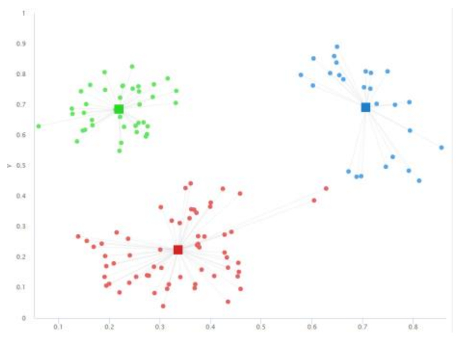
Centroid computation



Assignment of points to the nearest centroid



Centroid computation



K-means

1. Choose **k** random instances as cluster centers
2. Assign each instance to its closest cluster center
3. Re-compute cluster centers by computing the average (aka *centroid*) of the instances pertaining to each cluster
4. If cluster centers have moved, go back to Step 2

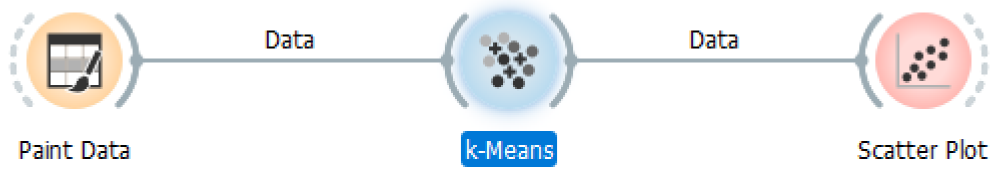
(Equivalent termination criterion: stop when assignment of instances to cluster centers has not changed)

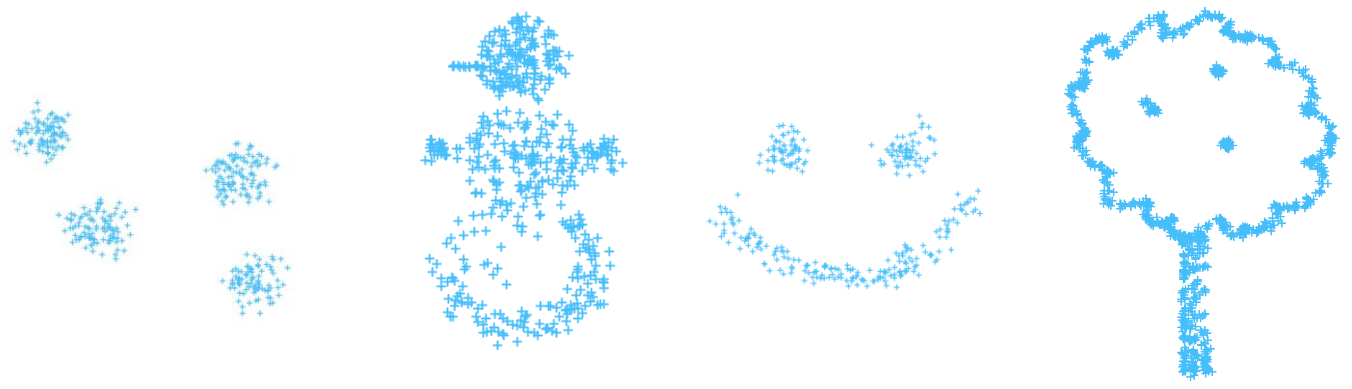
Alternatives: K-medoids, K-modes

- Might get stuck in local minima
- Silhuette for finding the optimal K

Lab exercise: clustering on drawings

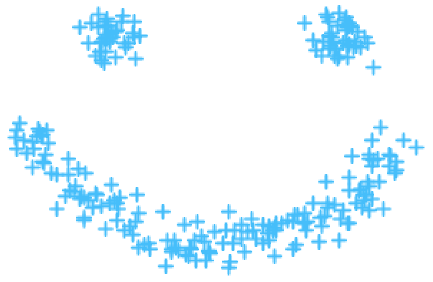
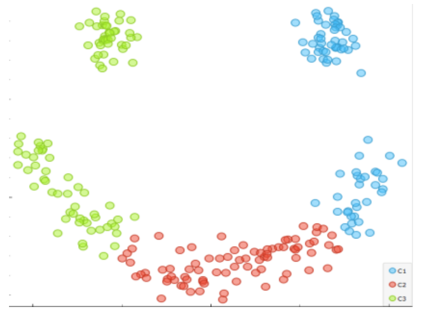
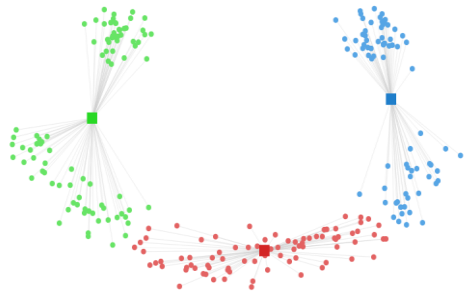
- Draw the following images in PaintData
  - Four snowballs
  - A snowman
  - A smiley face
  - An apple tree
- Compare
  - K-means
  - Hierarchical
  - DB-scan





Properties of k-Means

- The number of clusters ***k*** is fixed in advance
- It is fast, it always converges
- Can converge into a local minima (bad solution because of unlucky start)
- Finds convex (“spherical”) shaped clusters
- K-Means will cluster the data even if it can’t be clustered (e.g. data that comes from uniform distributions)

Clustering evaluation

- Clustering analysis doesn't have a solid evaluation metric

- External validation criteria
  - Using the ground truth to evaluate to evaluate the clustering result
- Internal validation criteria
  - Sum of distances to centroids
  - Intracluster to intercluster distance ratio
  - Silhouette coefficient

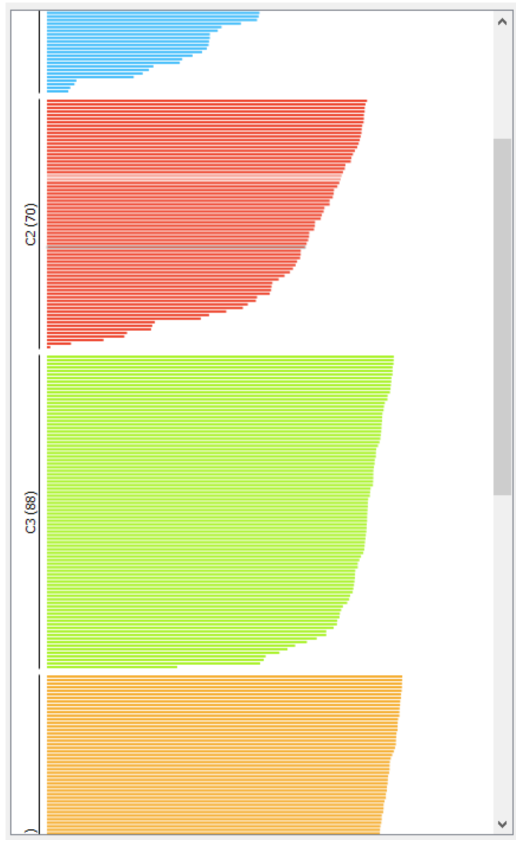
  - Parameter tuning – the "elbow" method

Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015. Chapter 6: cluster analysis, pgs 195 -201

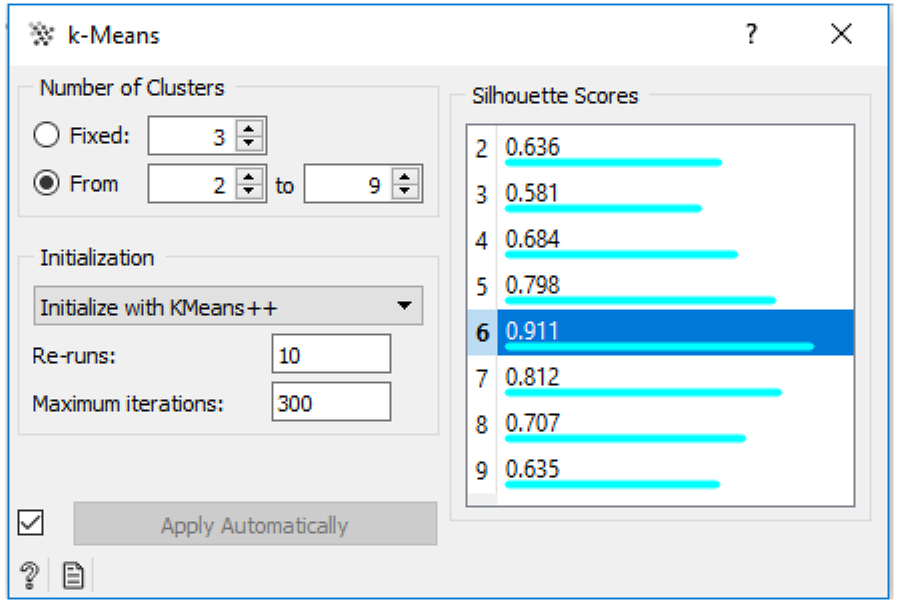
Silhouette coefficient

- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

- For example x_i , its silhouette coefficient is $s_i = (b_i - a_i) / \max(a_i, b_i)$
  - a_i average distance between x_i to all other examples in its cluster.
  - b_i average distance between x_i to the examples in the "neighboring" cluster
- The overall silhouette coefficient is the average of the data point-specific coefficients.



k-Means + Silhouette + „reruns“



k-Means

Number of Clusters

Fixed: 3

[✓ From] 2 to 9

Initialization

Initialize with KMeans++

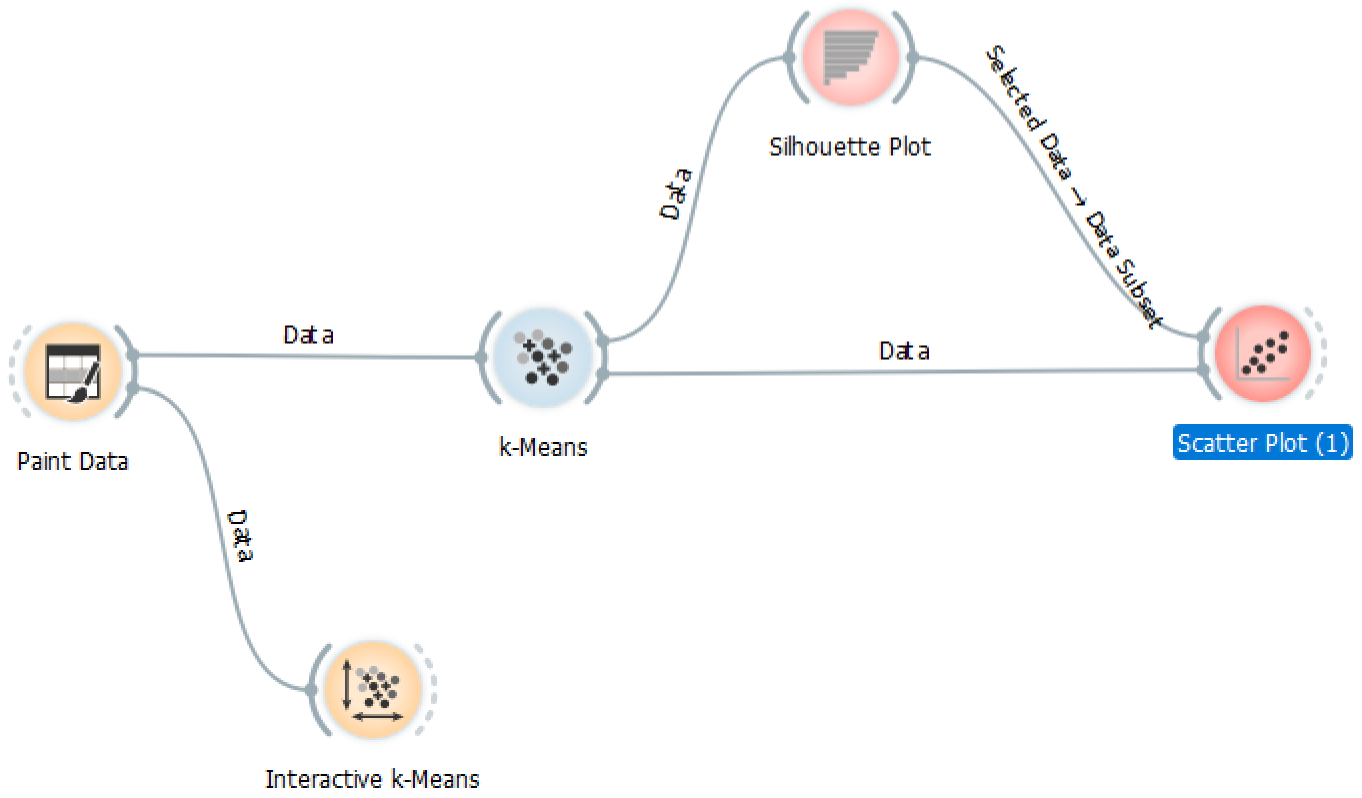
Re-runs: 10

Maximum iterations: 300

[✓] Apply Automatically

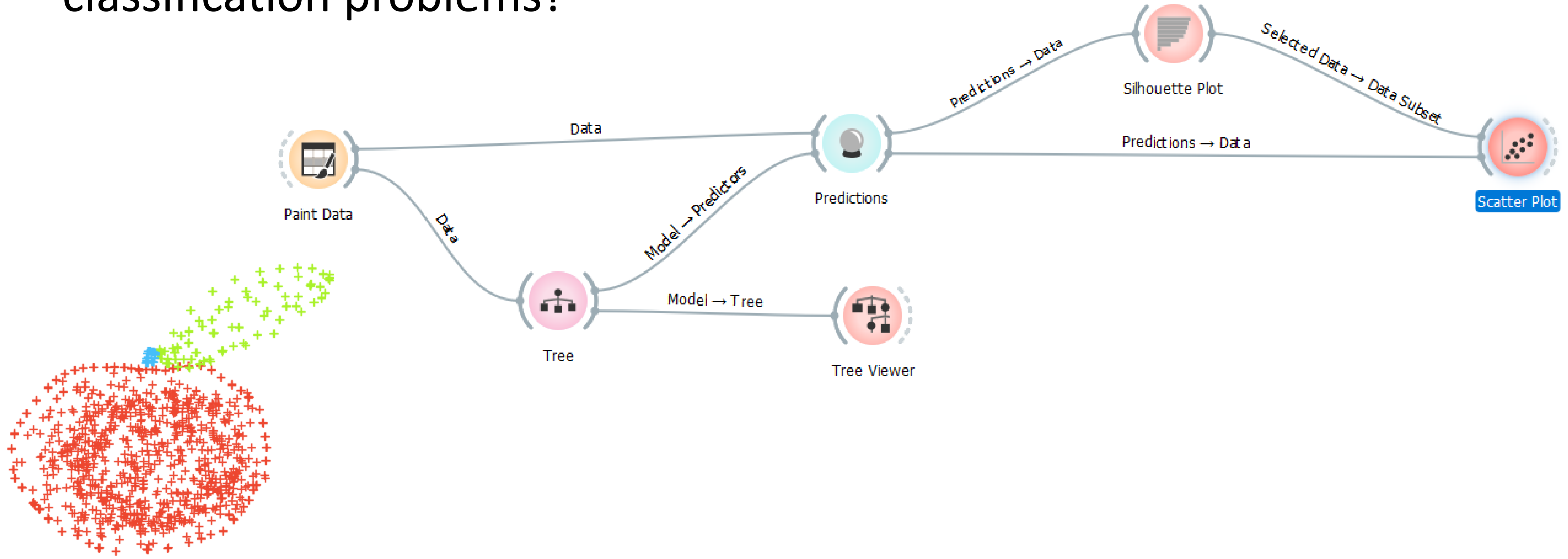
Silhouette Scores

2	0.636
3	0.581
4	0.684
5	0.798
6	0.911
7	0.812
8	0.707
9	0.635

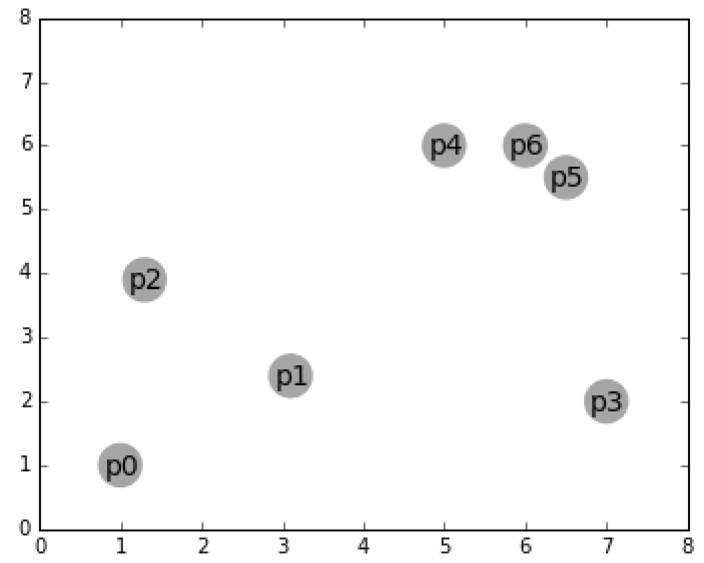


Orange workflow

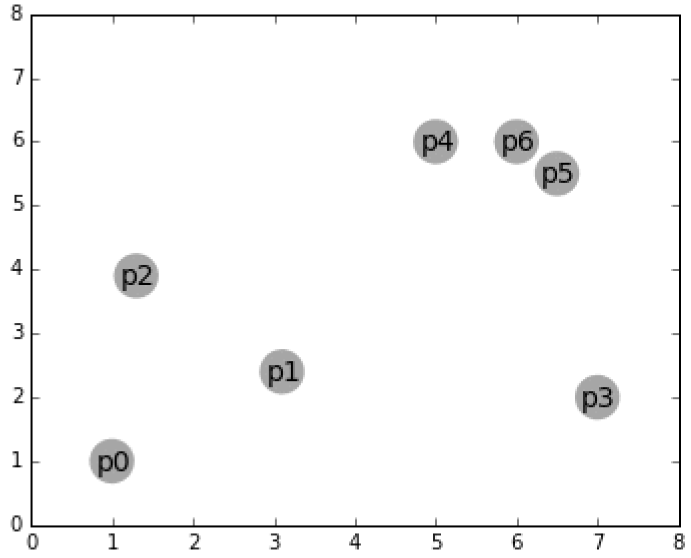
- How can we use the silhouette coefficient for searching for outliers in classification problems?

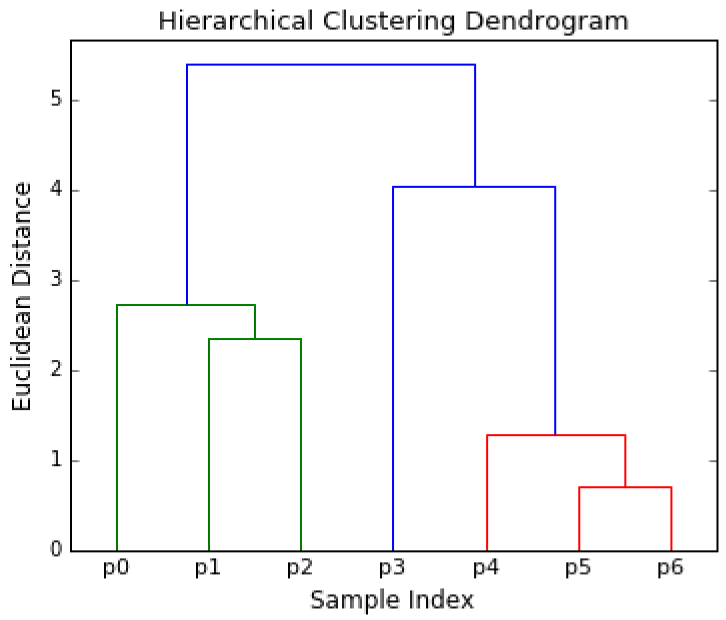


Agglomerative clustering - example



Agglomerative clustering - dendrogram





Agglomerative clustering

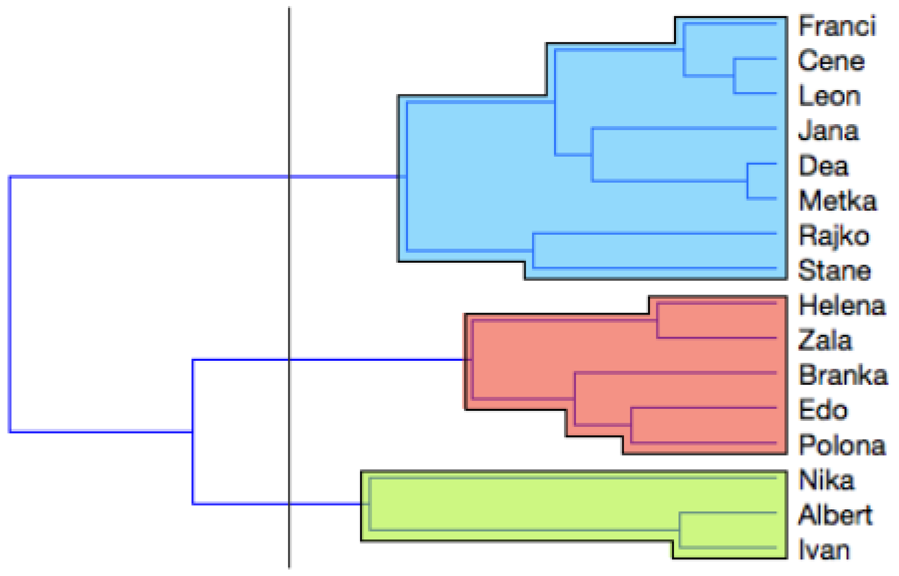
1. Start with a collection **C** of **n** singleton clusters
    - Each cluster contains one data point $\mathbf{c}_i = \{\mathbf{x}_i\}$
2. Repeat until only one cluster is left:
    1. Find a pair of clusters that is closest: min $\mathbf{D}(\mathbf{c}_i, \mathbf{c}_j)$
    2. Merge the clusters $\mathbf{c}_i$ and $\mathbf{c}_j$ into $\mathbf{c}_{i+j}$
    3. Remove $\mathbf{c}_i$ and $\mathbf{c}_j$ from the collection **C**, add $\mathbf{c}_{i+j}$

Some new index, not a sum

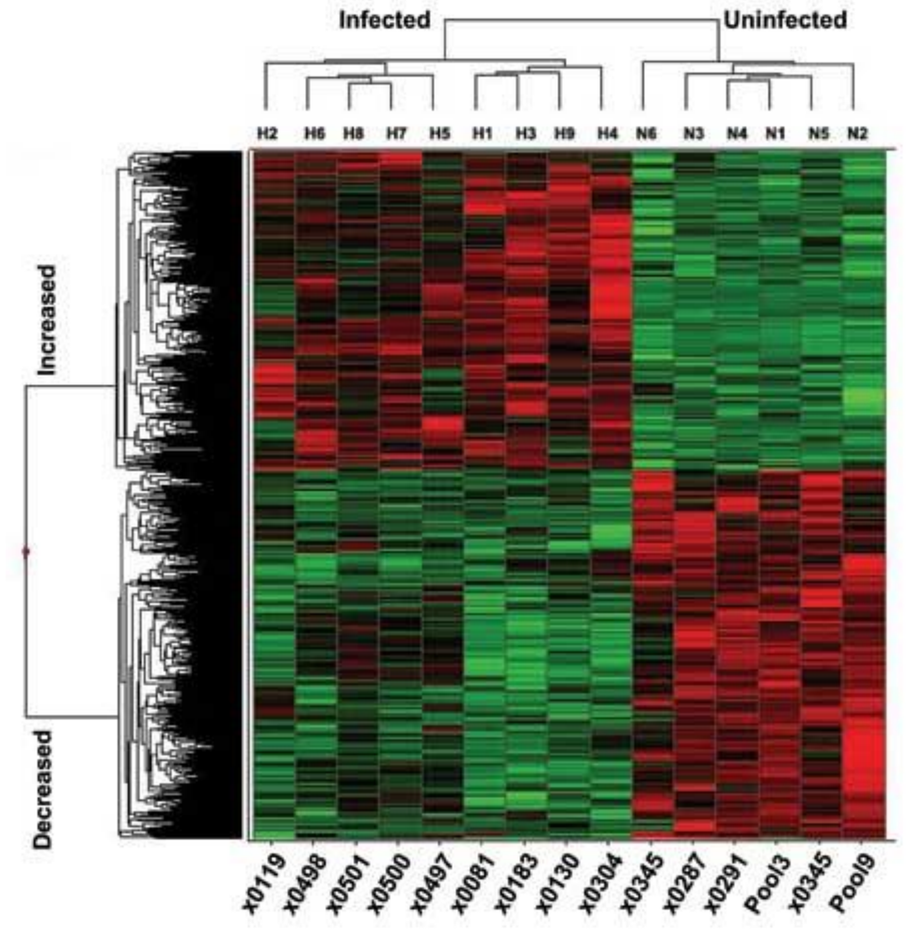
- Time and space complexity
- Sensitive to noisy data

Dendrogram

- The agglomerative hierarchical clustering algorithms result is commonly displayed as a tree diagram called a dendrogram.
- Dendrogram a tree diagram for showing taxonomic relationships.

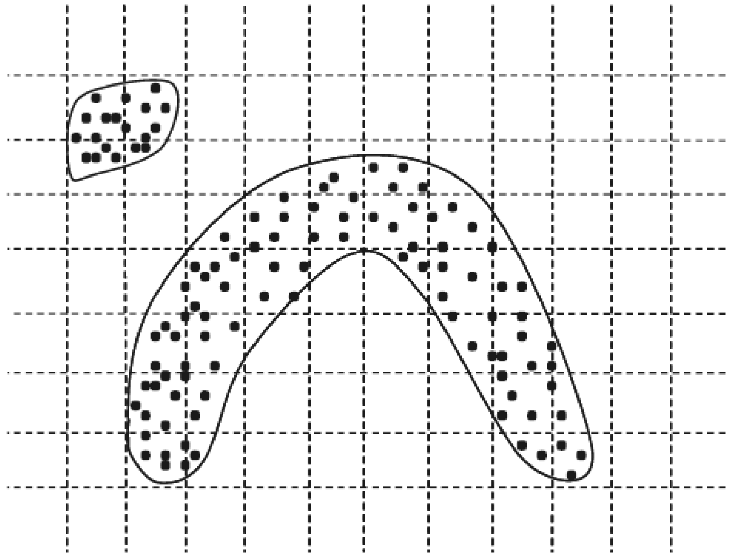


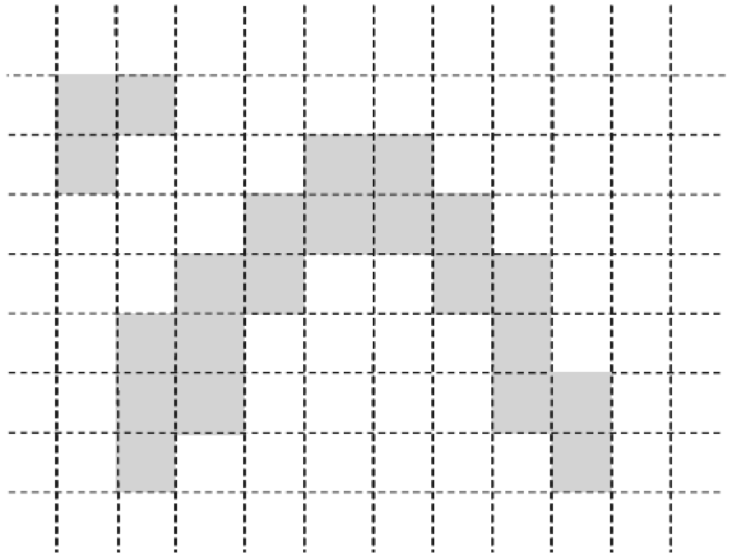
Example: Hierarchical clustering of genes



Grid-based (parameters **p** and **τ**)

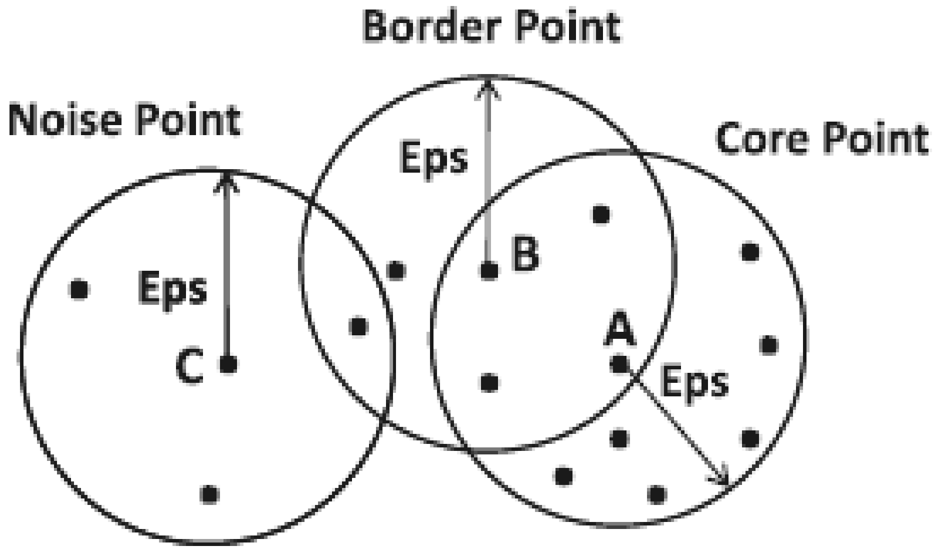
1. Discretize each dimension of **D** into **p** ranges
2. Determine dense grid cells at level **τ**
3. Create graph where dense grid cells are connected if they are adjacent
4. Determine connected components of graph
5. Return: points in each connected component as a cluster





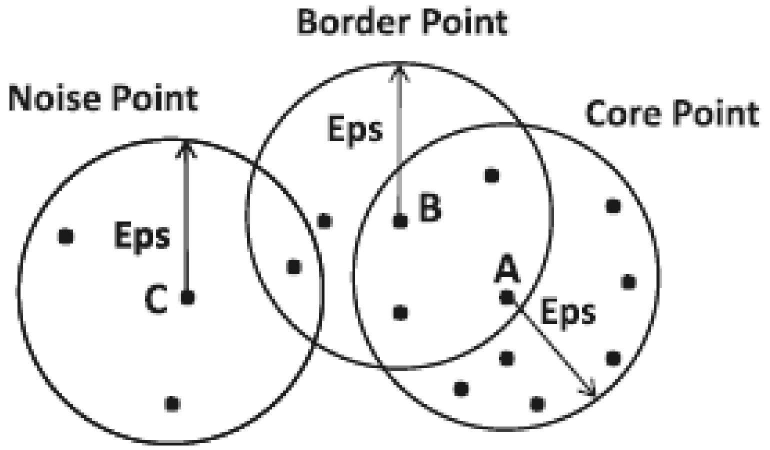
Density based clustering *DBSCAN*
(parameters: radius: *Eps*, density: τ)

- *Core point*:
  - contains at least τ data points within a radius *Eps*
- *Border point*:
  - not a core point
  - at least one core point within a radius *Eps*
- *Noise point*:
  - neither a core point nor a border point



Density based clustering *DBSCAN* (parameters: radius: *Eps*, density: τ)

1. Determine core, border and noise points at level (*Eps,* τ);
2. Create graph in which core points are connected if they are within *Eps* of one another;
3. Determine connected components in graph;
4. Assign each border point to connected component with which it is best connected;
5. **Return** points in each connected component as a cluster;



Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015. Chapter 6: cluster analysis, pg 183

DBSCAN properties

Similar to grid-based approaches, except that it uses circular regions as building blocks.

Advantages of DBSCAN:

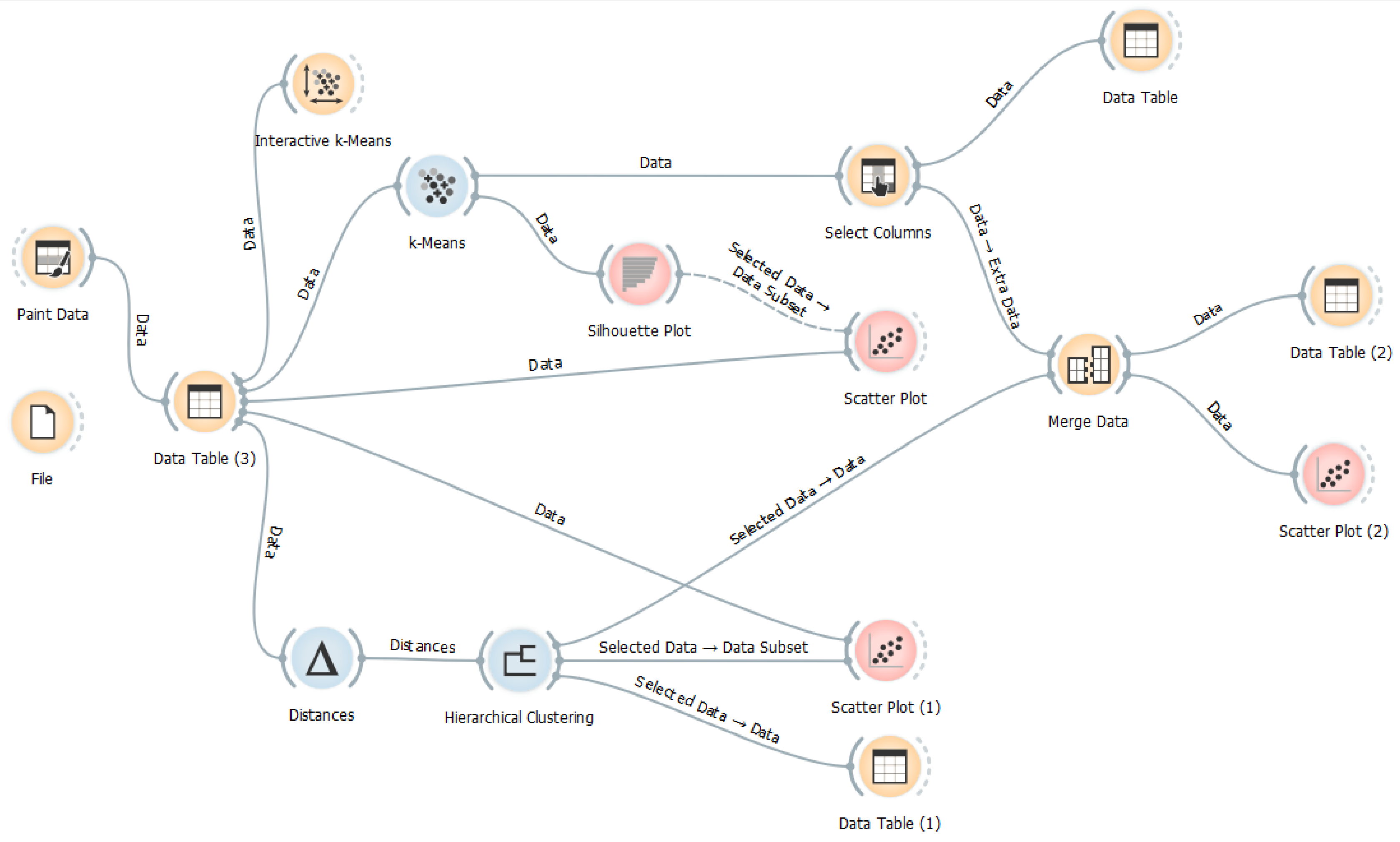
- Can detect clusters of arbitrary shape.
- Does not require the number of clusters as an input parameter.
- Not sensitive to outliers.

Disadvantages of DBSCAN:

- Computationally expensive in the first step (determining core, border and noise points)
- Susceptible to variations in the local cluster density.
- Struggles with high dimensionality data.

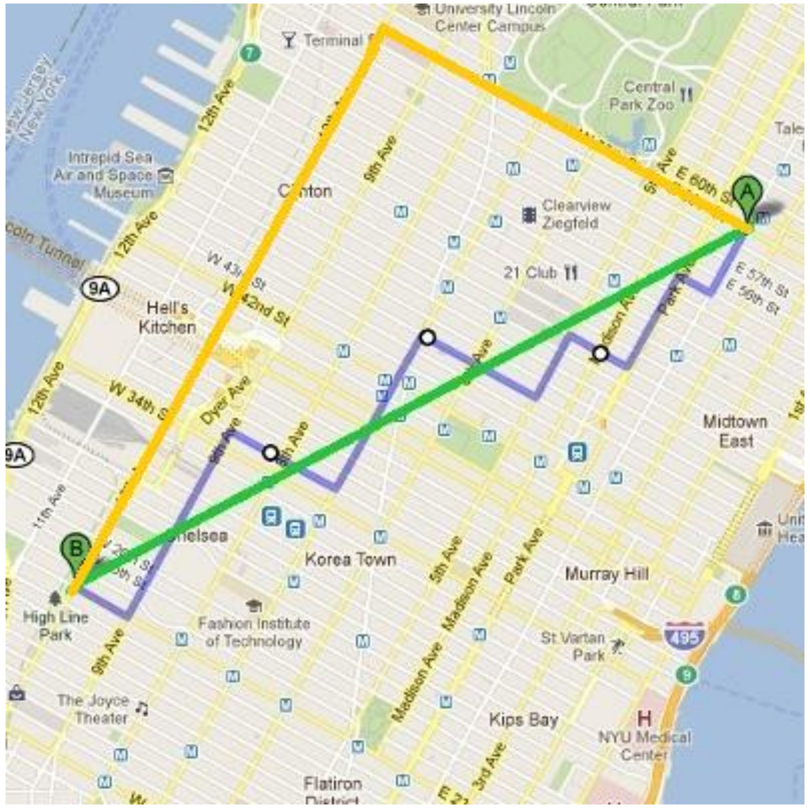
Lab work in Orange

- Comparison of hierarchical and k-Means clustering on
- painted data
- „wine.tab“, where we compare also to the real classes

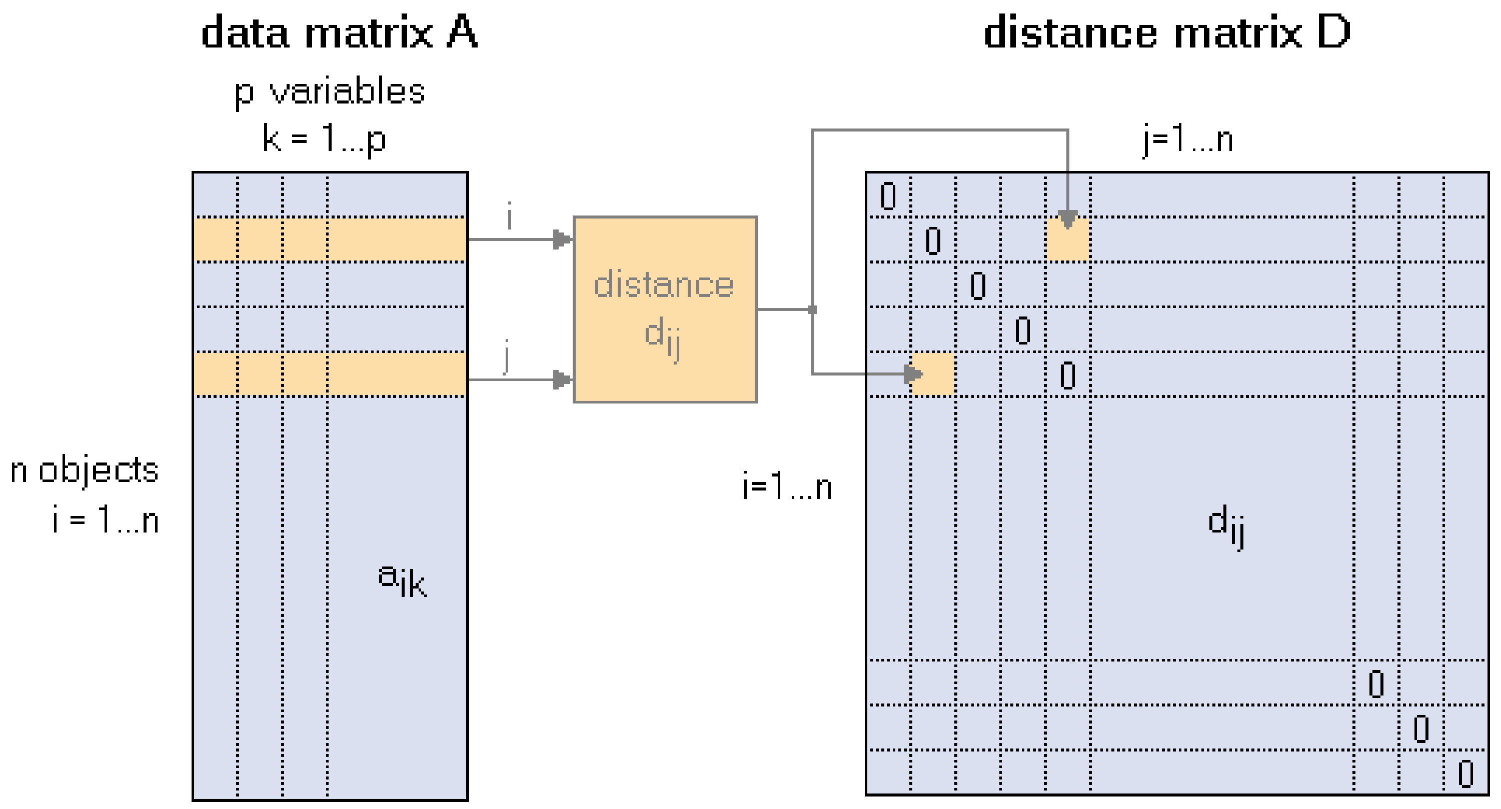


Similarity / distance measures

- The similarity measure depends on characteristics of the input data:
  - Attribute type: binary, categorical, continuous
  - Sparseness
  - Dimensionality
  - Type of proximity

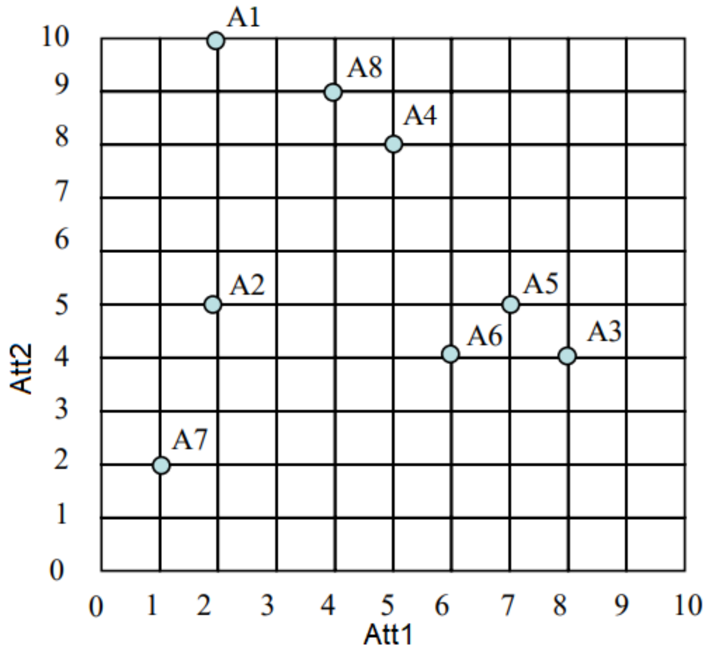


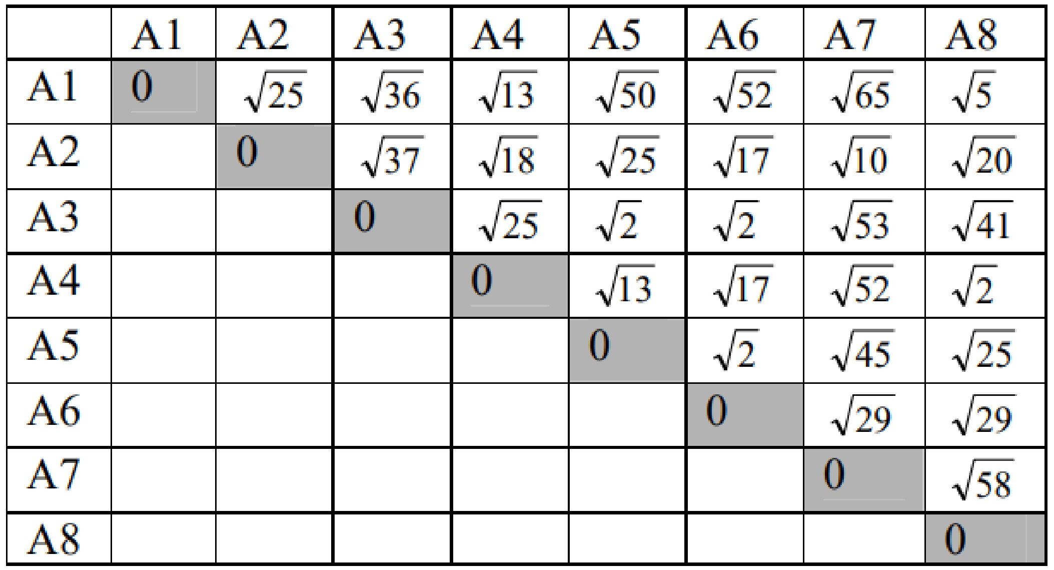
Distance matrix



Distance matrix example

	Att1	Att2
A1	2	10
A2	2	5
A3	8	4
A4	5	8
A5	7	5
A6	6	4
A7	1	2
A8	4	9

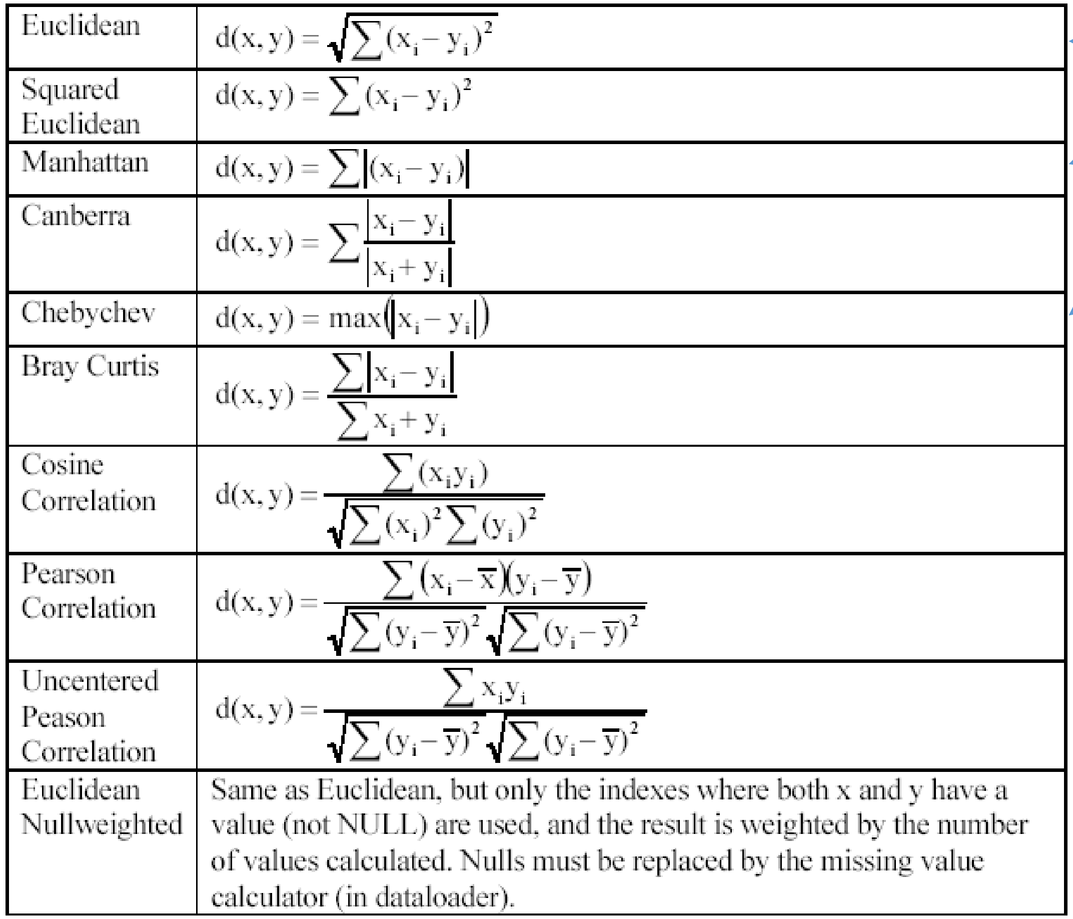




	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

$$\text{Euclidian} \xrightarrow{} Dist(A,B) = \sqrt[2]{(Att1(A) - Att1(B))^2 + (Att2(A) - Att2(B))^2}$$

Distance measures



Euclidean	$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$
Squared Euclidean	$d(x, y) = \sum (x_i - y_i)^2$
Manhattan	$d(x, y) = \sum \lvert (x_i - y_i) \rvert$
Canberra	$d(x, y) = \sum \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Chebychev	$d(x, y) = \max(\lvert x_i - y_i \rvert)$
Bray Curtis	$d(x, y) = \frac{\sum \lvert x_i - y_i \rvert}{\sum x_i + y_i}$
Cosine Correlation	$d(x, y) = \frac{\sum (x_i y_i)}{\sqrt{\sum (x_i)^2 \sum (y_i)^2}}$
Pearson Correlation	$d(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (y_i - \bar{y})^2} \sqrt{\sum (y_i - \bar{y})^2}}$
Uncentered Peason Correlation	$d(x, y) = \frac{\sum x_i y_i}{\sqrt{\sum (y_i - \bar{y})^2} \sqrt{\sum (y_i - \bar{y})^2}}$
Euclidean Nullweighted	Same as Euclidean, but only the indexes where both x and y have a value (not NULL) are used, and the result is weighted by the number of values calculated. Nulls must be replaced by the missing value calculator (in dataloader).

Minkowski distance

$$D(X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer. (Chapter 3)

Homework

- Similarity vs. distance
- List algorithms that are based on distance/similarity

Literature

- Max Bramer: Principles of data mining (2007)
  - 14. Clustering
- Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015. Chapter 6: Cluster analysis
- Aggarwal, Charu C. *Data mining: the textbook*. Springer, 2015. Chapter 2: Similarity and distances

Association rules

Motivation

What people buy in a given shopping experience.

- 25 Osco Drug stores
- 1.2 million market baskets

(A market basket is the stuff you put in
the physical cart and check out at the register.)

- An unexpected pattern

Between 5p.m. and 7p.m. **diapers → beer**

http://www.dssresources.com/newsletters/66.php

Association rules

- Determine associations between groups of items bought by customers.
- No predefined target variable(s).
- Find interesting, useful patterns and relationships.
- Data mining, business intelligence.

* Terminology from market basket analysis (transactions, items, itemsets, …)

Confidence and support

- The dataset consists of **n** transactions
- We have an association rule A→ B

The **support** of an itemset A is defined as the fraction of the transactions in the database T = {T1 . . . Tn} that contain A as a subset.

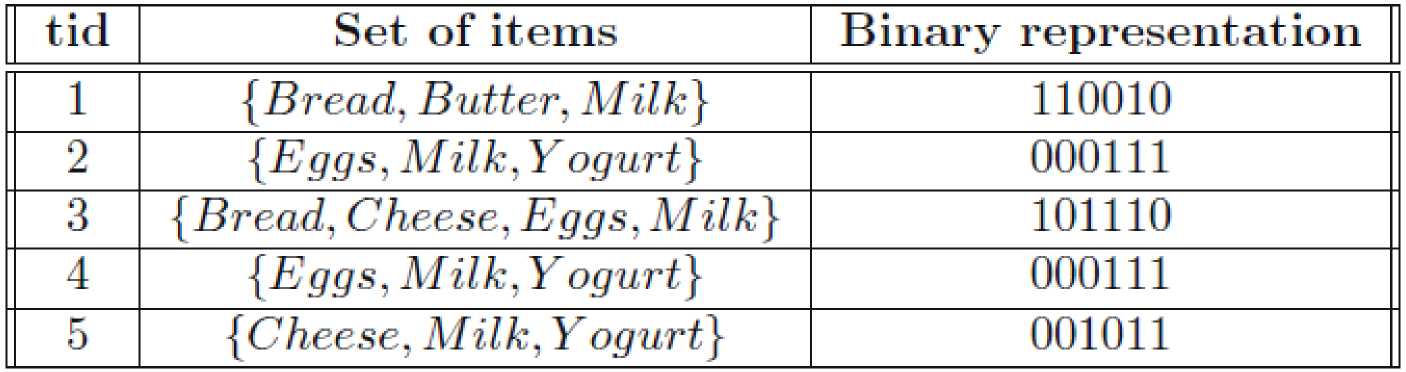
$$supp(A) = \frac{|A|}{n}$$

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

The **confidence** of the rule A→ B is the conditional probability of A and B occurring in a transaction, given that the transaction contains A.

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Example of a snapshot of a market basket data set



tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

- What do customers buy together?
- Which items imply the purchase of other items?

Charu C. Aggarwal : Data Mining: The Textbook (2015), pg. 95

Exercise

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

$$supp(A) = \frac{|A|}{n}$$

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Supp ({Bread}) =

Supp ({Milk, Yogurt}) =

Conf ({Milk} → {Yogurt}) =

Conf ({Yogurt} → {Milk}) =

Association rules

- Rules **A → B**, where A and B are conjunctions of items
- Task: Find **all** association rules that satisfy the minimum support and minimum confidence constraints

- **Support**:

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

- **Confidence**:

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Hard problem

- In practice
  - Millions of transactions
  - Many (thousands) of items
- Too many possible combinations
  - 1000 items for sale → $2^{1000} - 1$ candidate market baskets
- Solution
  - *Apriori algorithm*

Frequent itemsets: intuition

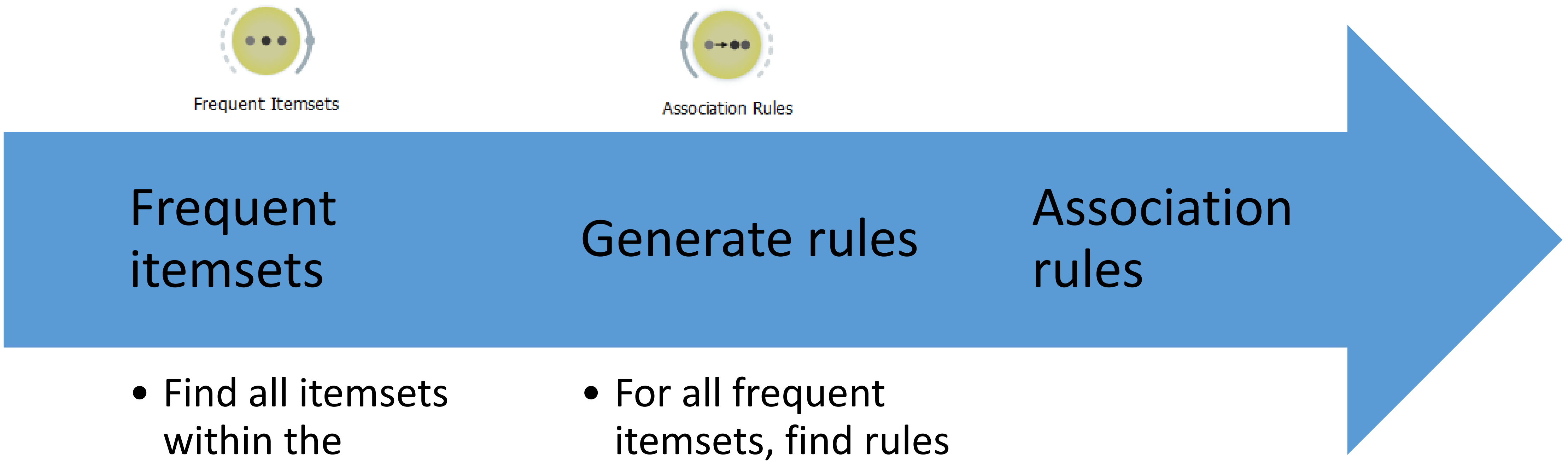
- We have ***n*** transactions containing (at least) {gloves, scarf and hat}
- What can we say about the number of transaction containing {gloves and scarf}?

*At least **n**.*

- The **anti-monotone property of support:** if we drop out an item from an itemset, support value of new itemset generated will either be the same or will increase.

$$\forall A, B : A \subseteq B \Rightarrow supp(A) \geq supp(B)$$

Apriori



- Find all itemsets within the *minSupport* constraint

- For all frequent itemsets, find rules which satisfy the *minConfidence* constraint

*Frequent itemsets = large itemsets, sometimes also frequent patterns

Apriori

```
Create $L_1$ = set of supported itemsets of cardinality one
Set $k$ to 2
while ($L_{k-1} \neq \emptyset$) {
    Create $C_k$ from $L_{k-1}$
    Prune all the itemsets in $C_k$ that are not
        supported, to create $L_k$
    Increase $k$ by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$
```

(Generates C_k from L_{k-1})

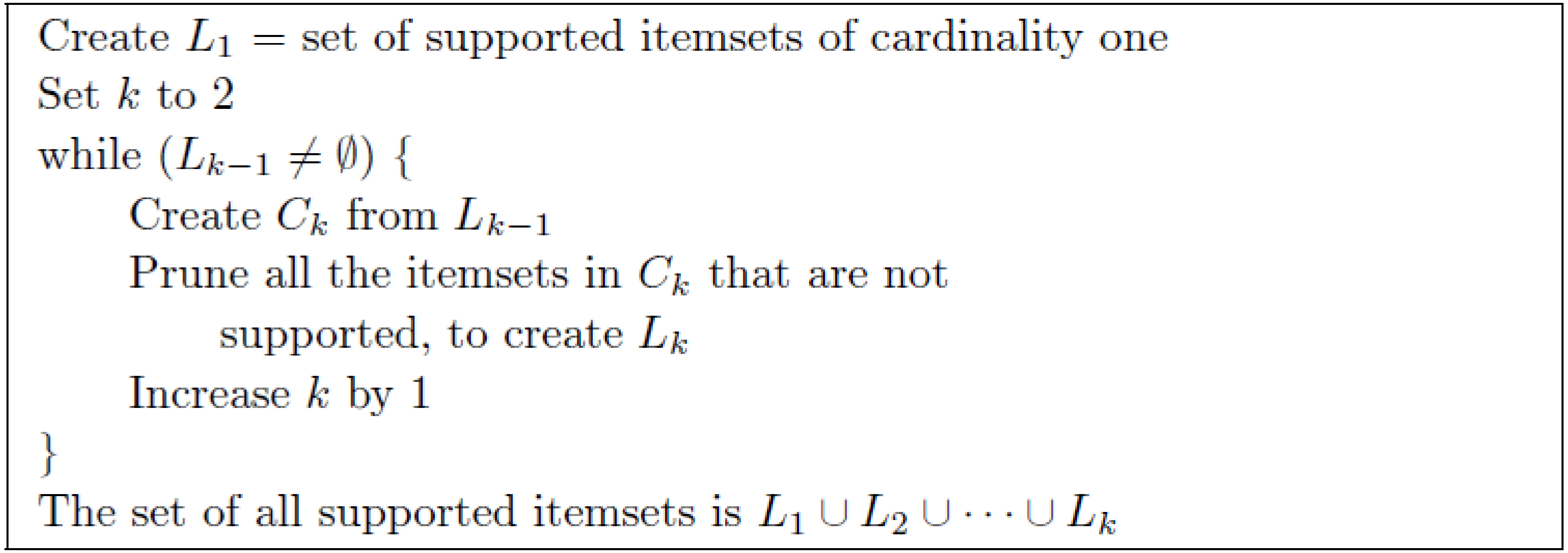
Join Step

Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.

Prune Step

```
For each member $c$ of $C_k$ in turn {
Examine all subsets of $c$ with $k-1$ elements
Delete $c$ from $C_k$ if any of the subsets is not a member of $L_{k-1}$
}
```

Apriori: Frequent itemset mining

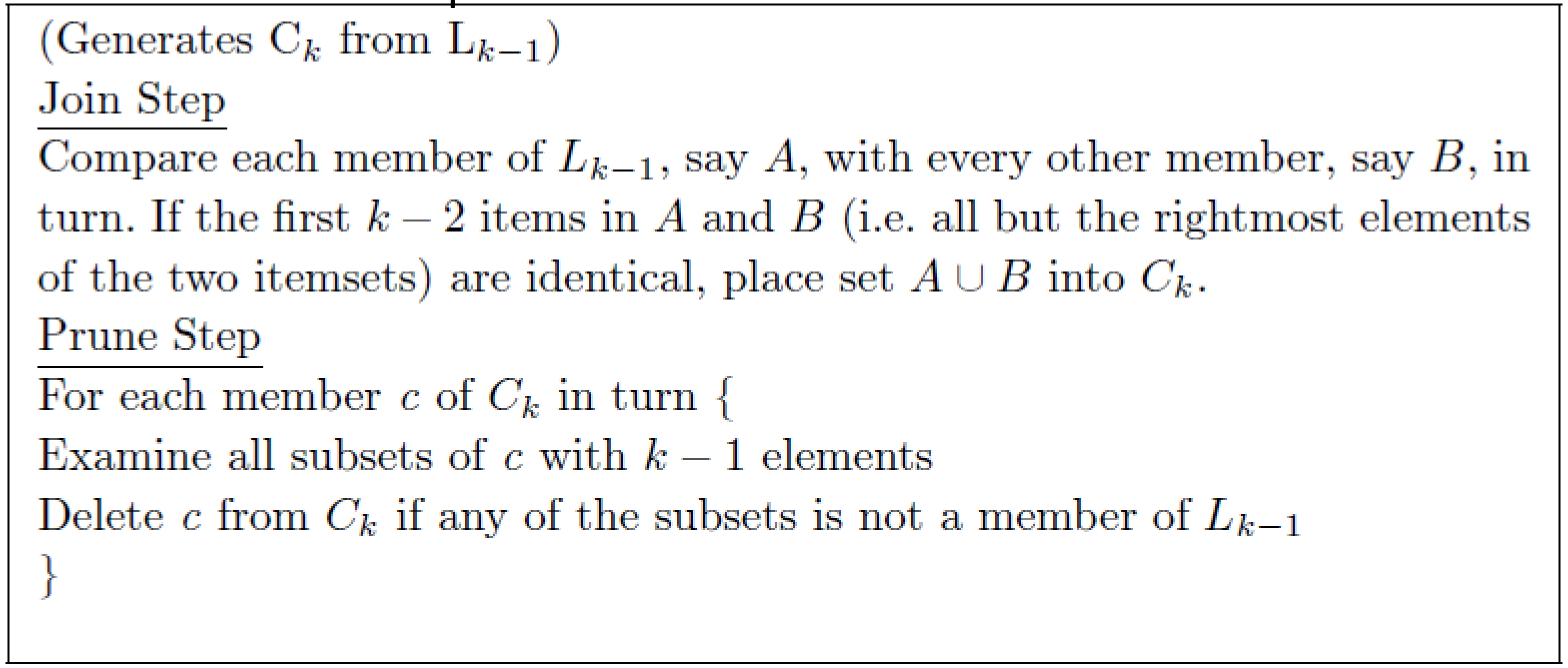


```
Create $L_1$ = set of supported itemsets of cardinality one
Set $k$ to 2
while ($L_{k-1} \neq \emptyset$) {
    Create $C_k$ from $L_{k-1}$
    Prune all the itemsets in $C_k$ that are not
        supported, to create $L_k$
    Increase $k$ by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$
```

- The items in the sets should be ordered (alphabetically, …)

Apriori: constructing the next level from the previous one

- Since items in the sets are ordered (alphabetically, …)
- Join Step:
  - Merge sets that have all the elements the same except for the rightmost one
- Prune Step:
  - Remove the set if any of its subsets are not on the previous level



(Generates C_k from L_{k-1})

Join Step

Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.

Prune Step

```
For each member $c$ of $C_k$ in turn {
Examine all subsets of $c$ with $k-1$ elements
Delete $c$ from $C_k$ if any of the subsets is not a member of $L_{k-1}$
}
```

Rules from frequent itesmets

- Generate rules with a certain confidence
- All the counts we need are in the lattice (no database scanning)
- Confidence of rules generated from the same itemset has an anti-monotone property
- No need to check all the rules, since

$$\text{Conf}(\{A,B\} \rightarrow \{C\}) >= \text{Conf}(\{A\} \rightarrow \{B,C\})$$
$$\text{Conf}(\{A,B,C\} \rightarrow \{D\}) \geq \text{Conf}(\{A,B\} \rightarrow \{C,D\}) \geq \text{Conf}(\{A\} \rightarrow \{B,C,D\})$$

*In general, confidence does not have an anti-monotone property: Conf(ABC → D) can be larger or smaller than Conf(AB → D)

Charu C. Aggarwal : Data Mining: The Textbook (2015), pg. 98

Exercise: Association rules

Generate frequent itemsets with support at least 2/6 and confidence at least 75%.

Items: **A**=apple, **B**=banana, **C**=coca-cola, **D**=doughnut

- Client 1 bought: A, B, C, D
- Client 2 bought: B, C
- Client 3 bought: B, D
- Client 4 bought: A, C
- Client 5 bought: A, B, D
- Client 6 bought: A, B, C

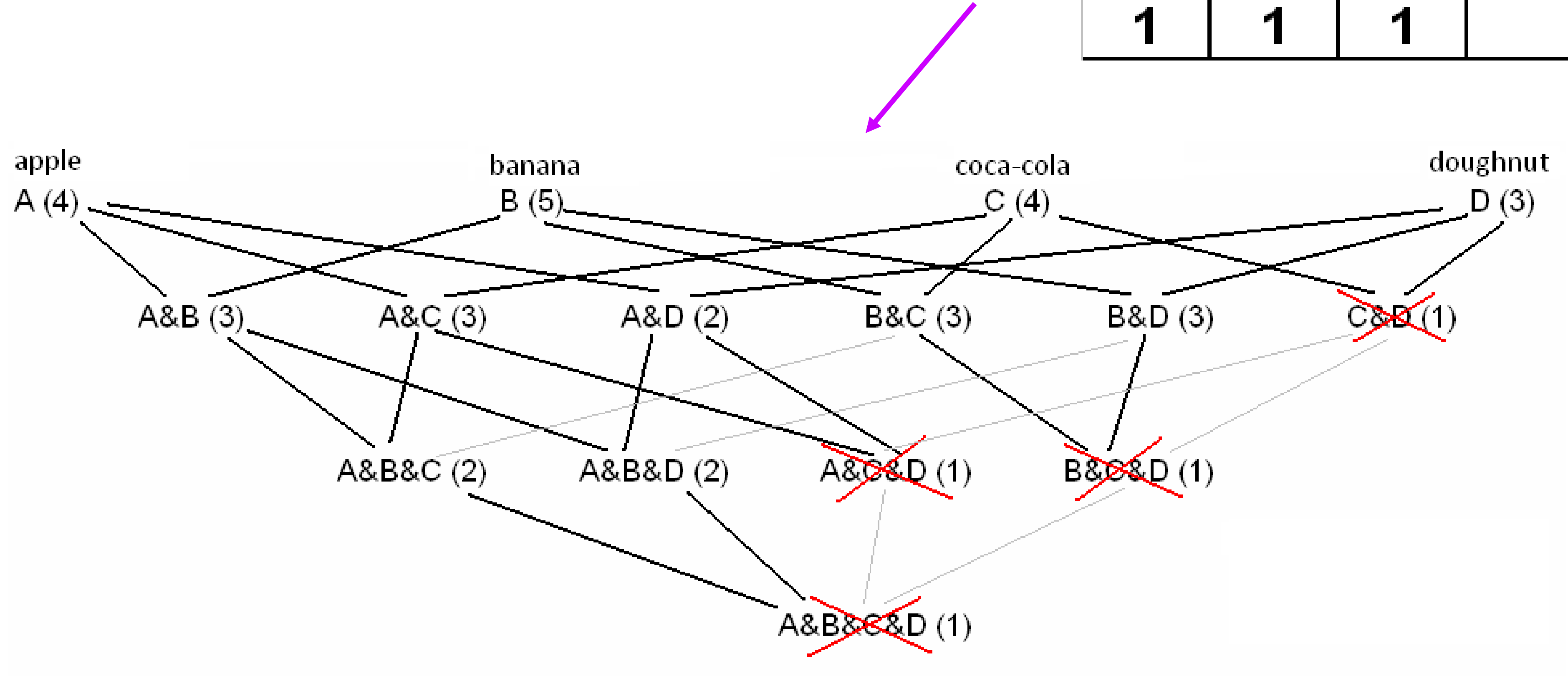
Exercise: Frequent itemsets

To ease the counting, we transcribe into a binary representation.

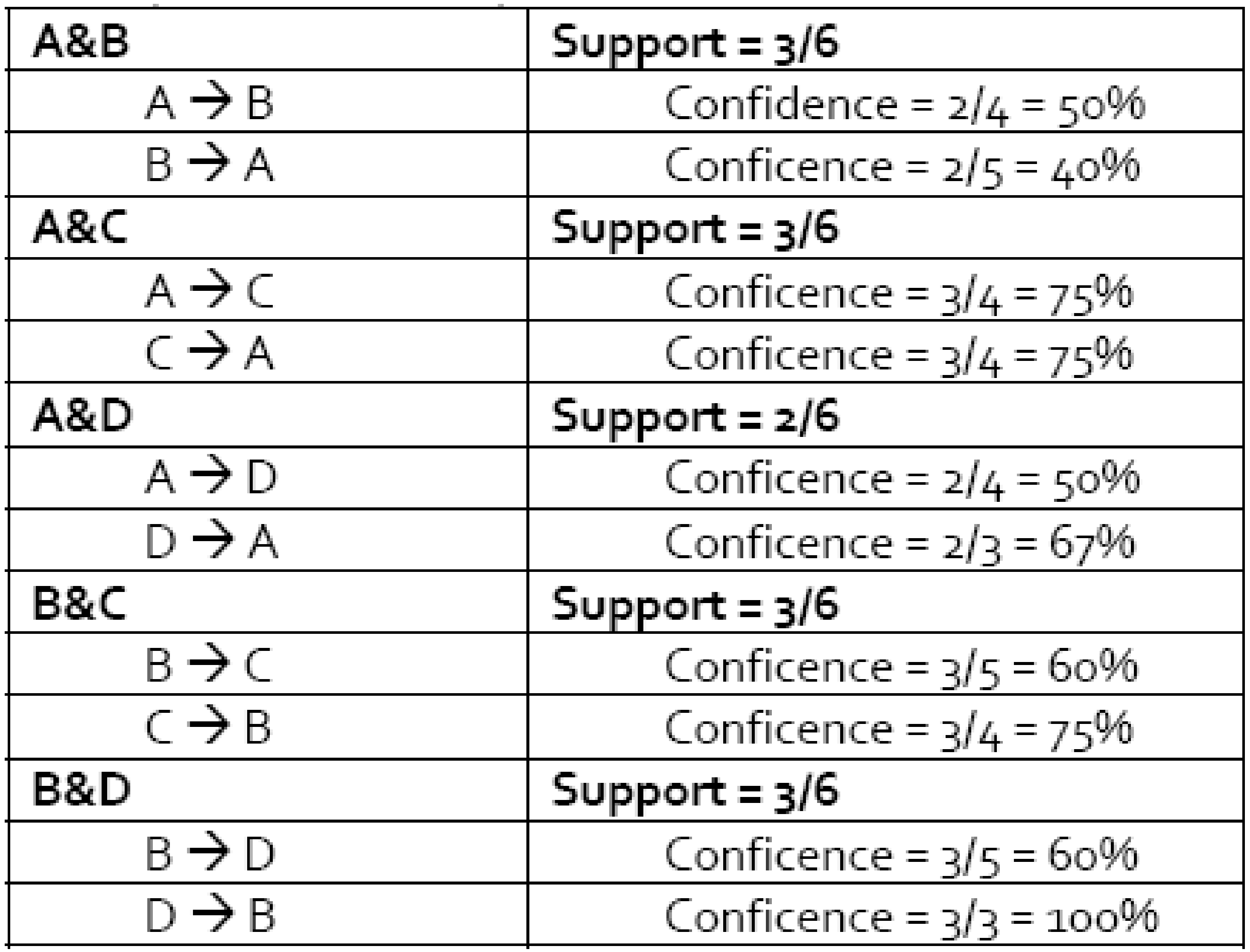
A	B	C	D
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	

Frequent itemsets

A	B	C	D
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	



Association rules …

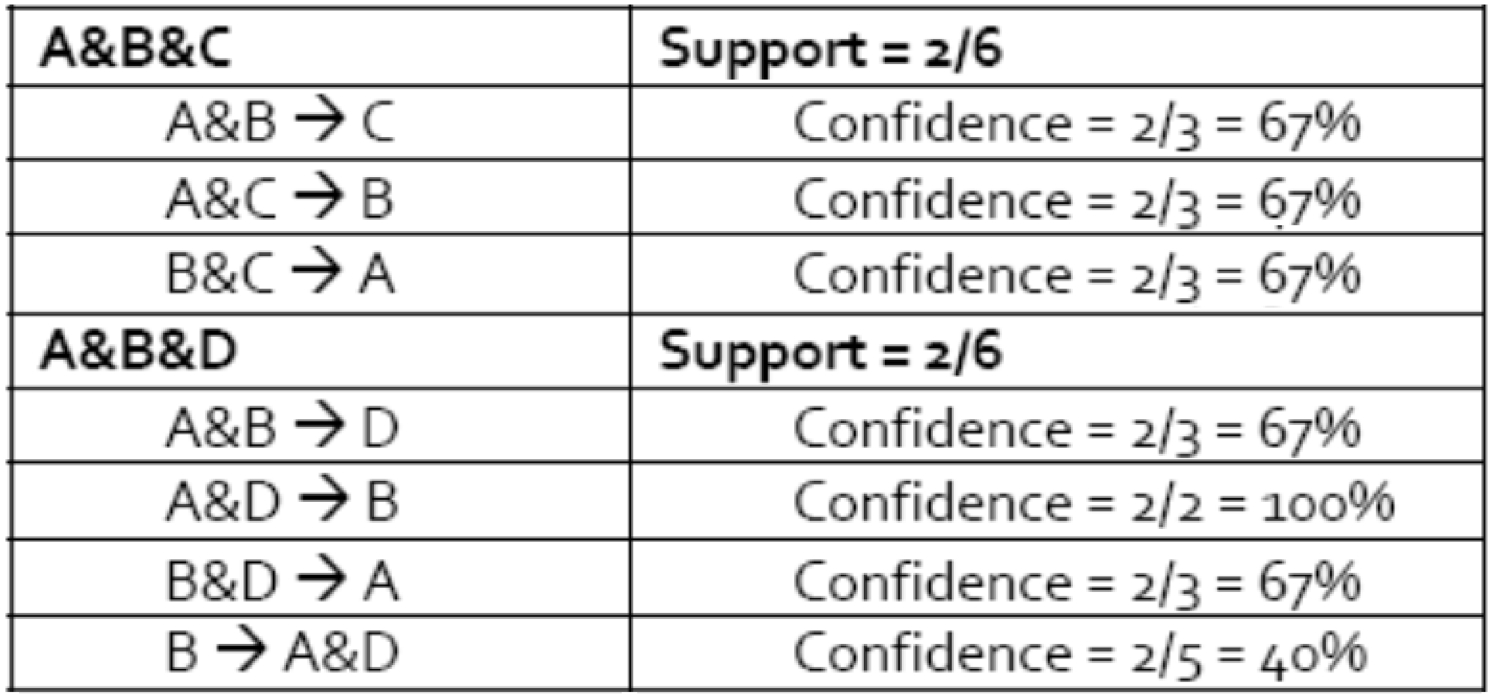


A&B	**Support = 3/6**
A → B	Confidence = 2/4 = 50%
B → A	Conficence = 2/5 = 40%
A&C	**Support = 3/6**
A → C	Conficence = 3/4 = 75%
C → A	Conficence = 3/4 = 75%
A&D	**Support = 2/6**
A → D	Conficence = 2/4 = 50%
D → A	Conficence = 2/3 = 67%
B&C	**Support = 3/6**
B → C	Conficence = 3/5 = 60%
C → B	Conficence = 3/4 = 75%
B&D	**Support = 3/6**
B → D	Conficence = 3/5 = 60%
D → B	Conficence = 3/3 = 100%

… association rules

No need to generate rules {A}→{B,C}, {B}→{C,A} , {C}→{A,B} because the rules with two items on the left side from the same itemset do not satisfy the minimum confidence constraint.

Similar for the itemset {A, B, D}



A&B&C	**Support = 2/6**
A&B → C	Confidence = 2/3 = 67%
A&C → B	Confidence = 2/3 = 67%
B&C → A	Confidence = 2/3 = 67%
A&B&D	**Support = 2/6**
A&B → D	Confidence = 2/3 = 67%
A&D → B	Confidence = 2/2 = 100%
B&D → A	Confidence = 2/3 = 67%
B → A&D	Confidence = 2/5 = 40%

Lift

- The lift of rule **L → R** measures how many more times the items in L and R occur together in transactions than would be expected if the itemsets L and R were statistically independent.

$$\text{lift}(L \rightarrow R) = \frac{\text{support}(L \cup R)}{\text{support}(L) \times \text{support}(R)}$$

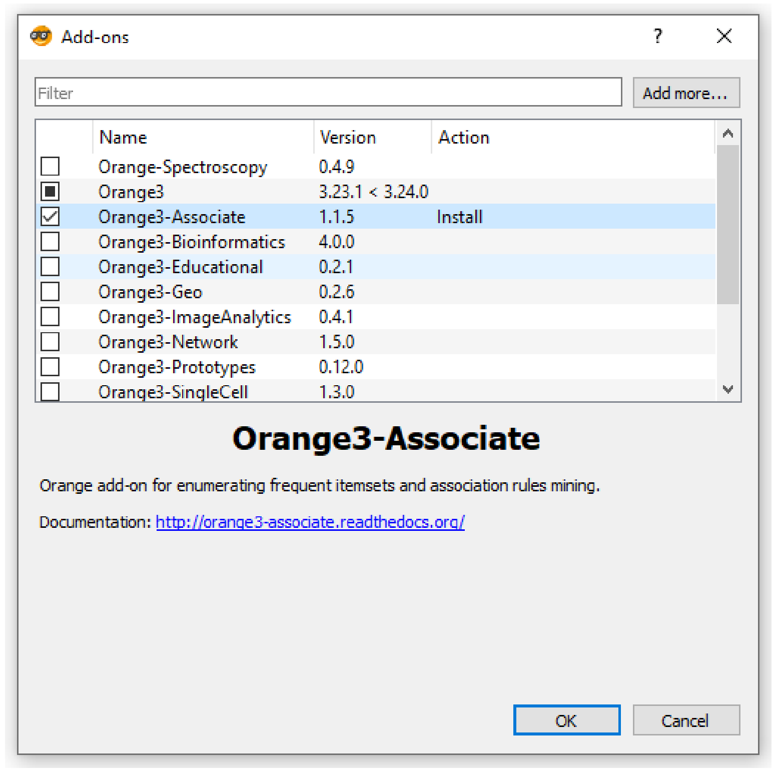
$$\text{lift}(L \rightarrow R) = \text{lift}(R \rightarrow L)$$

Leverage

- The leverage of rule **L → R** is the difference between the support for L ∪ R (i.e. the items in L and R occurring together in the database) and the support that would be expected if L and R were independent.

$$\text{leverage}(L \rightarrow R) = \text{support}(L \cup R) - \text{support}(L) \times \text{support}(R)$$

Association rules: Orange workflow


Add-ons

Filter | Add more...

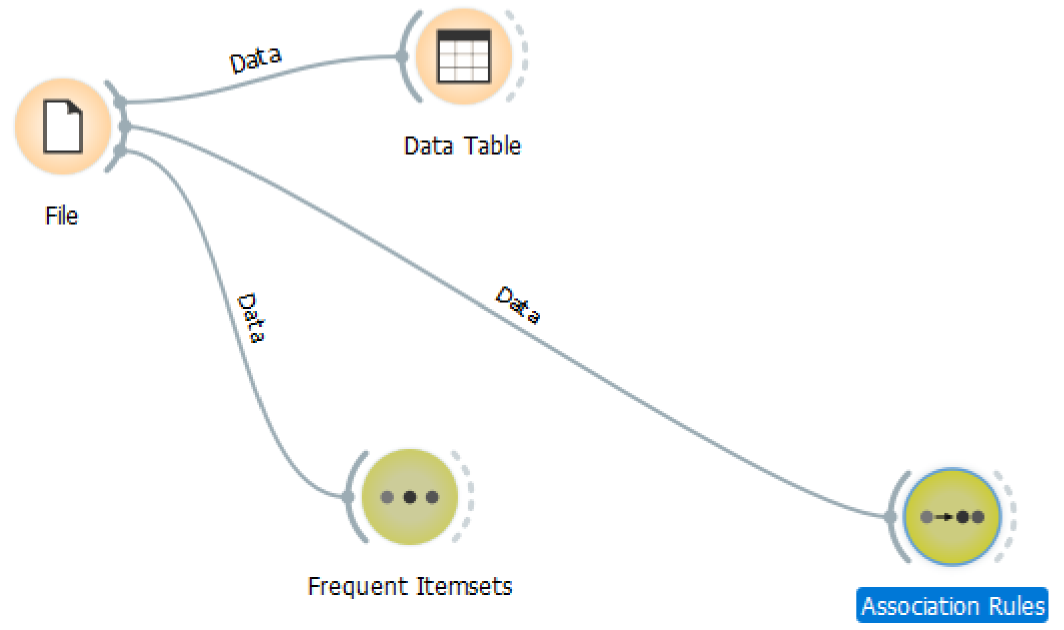
	Name	Version	Action
☐	Orange-Spectroscopy	0.4.9	
■	Orange3	3.23.1 < 3.24.0	
☑	Orange3-Associate	1.1.5	Install
☐	Orange3-Bioinformatics	4.0.0	
☐	Orange3-Educational	0.2.1	
☐	Orange3-Geo	0.2.6	
☐	Orange3-ImageAnalytics	0.4.1	
☐	Orange3-Network	1.5.0	
☐	Orange3-Prototypes	0.12.0	
☐	Orange3-SingleCell	1.3.0	

Orange3-Associate

Orange add-on for enumerating frequent itemsets and association rules mining.

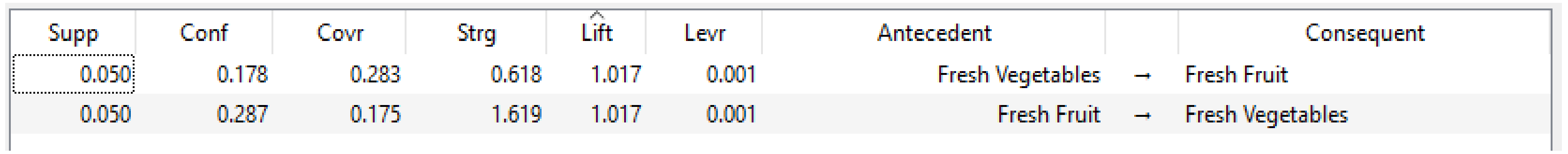
Documentation: http://orange3-associate.readthedocs.org/

OK | Cancel



* Start with a small minSupport and we increase it gradually (to avoid running out of memory)

Association rules quality measures in Orange



Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.050	0.178	0.283	0.618	1.017	0.001	Fresh Vegetables	→	Fresh Fruit
0.050	0.287	0.175	1.619	1.017	0.001	Fresh Fruit	→	Fresh Vegetables

- **support, confidence, lift, leverage**
- **coverage**: how often antecedent items are found in the data set (support of antecedent/data)
- **strength**: (support of consequent/support of antecedent)

Lab exercise

Datasets

- https://biolab.si/core/foodmart.basket
- https://github.com/digizeph/data_mining/blob/master/data/FoodMart.csv
- http://file.biolab.si/datasets/voting.tab

1. Compare the two datasets (files)
2. Generate frequent itemsets and association rules for both datasets. What is the difference?
3. Frequent itemsets and association rules for „Voting.tab“
4. Compute „conviction“ for a few rules

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}$$

Conviction: the ratio of the expected frequency that X occurs without Y if X and Y were independent divided by the observed frequency of incorrect predictions.

Beyond the basics

- Summarization:
    - finding maximal itemsets
    - closed itemsets
    - nonredundant rules
- *Constraint incorporation:*
    - application-specific constraints into the itemset generation process → much lower support-levels
- Multilevel association rules: considering item classes
    - (e.g., chips, peanuts, bretzels, etc., belonging to class “snacks”)

(Non) redundant patterns

Maximal Frequent Itemset

- *A frequent itemset is maximal at a given minimum support level minsup if it is frequent and no superset of it is frequent.*

Closed Itemsets

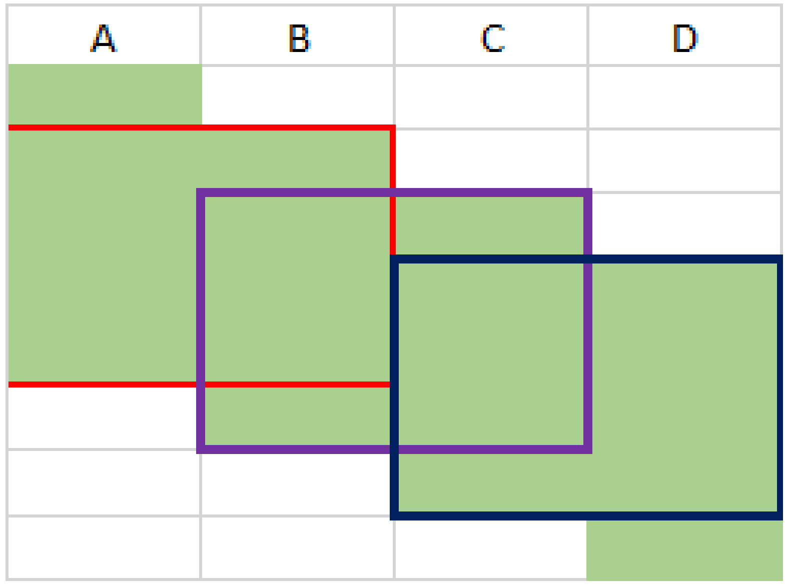
- *An itemset X is closed, if none of its supersets have exactly the same support count as X.*

Closed Frequent Itemsets

- *An itemset X is a closed frequent itemset at minimum support minsup, if it is both closed and frequent.*

Example

A	B	C	D
🟩			
🟩	🟩		
🟩	🟩	🟩	
🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩
	🟩	🟩	🟩
		🟩	🟩
			🟩



Find closed frequent itemsets with the minSupport level 5/8 and 4/8.

Constraint-based pattern mining

- Constraint-based pattern mining systems are systems that with minimal effort can be programmed to find different types of patterns satisfying constraints.
- They achieve this genericity by providing
  - (1) high-level languages for specifying constraints;
  - (2) generic search algorithms that find patterns for any task

Applications

- Market Basket Analysis
- Classification: rule based classifiers
- Clustering: CLIQUE algorithm for finding dense regions of data
- Outlier detection: the outliers are defined as transactions that are not "covered" by most of the association patterns in the data
- Recommendations and Collaborative Filtering: In collaborative filtering, the idea is to make recommendations to users on the basis of the buying behavior of other similar users. The problem of localized pattern mining is much more challenging because of the need to simultaneously determine the clustered segments and the association rules.

Literature

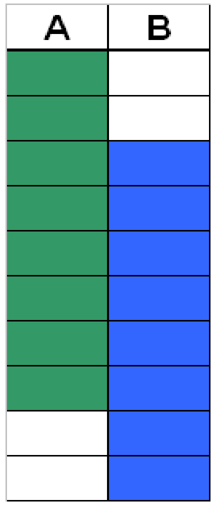
- Max Bramer: Principles of data mining (2007)
  - 13. Association Rule Mining II
- Charu C. Aggarwal : Data Mining: The Textbook (2015)
  - 4. Association pattern mining
  - 5. Association pattern mining: advanced concepts

- What is the "true story" about using data mining to identify a relation between sales of beer and diapers? http://www.dssresources.com/newsletters/66.php

Discussion questions

1. Transformation of an attribute-value dataset to a transaction dataset.
2. What would be the association rules for a dataset with two items A and B, each of them with support 80% and appearing in the same transactions as rarely as possible?
   a. minSupport = 50%, min conf = 70%
   b. minSupport = 20%, min conf = 70%
3. What if we had 4 items: A, ¬A, B, ¬ B

4. Compare decision trees and association rules regarding handling an attribute like “PersonID”. What about attributes that have many values (eg. Month of year)



A	B
🟩	
🟩	
🟩	🟦
🟩	🟦
🟩	🟦
🟩	🟦
🟩	🟦
🟩	🟦
	🟦
	🟦

Artificial intelligence

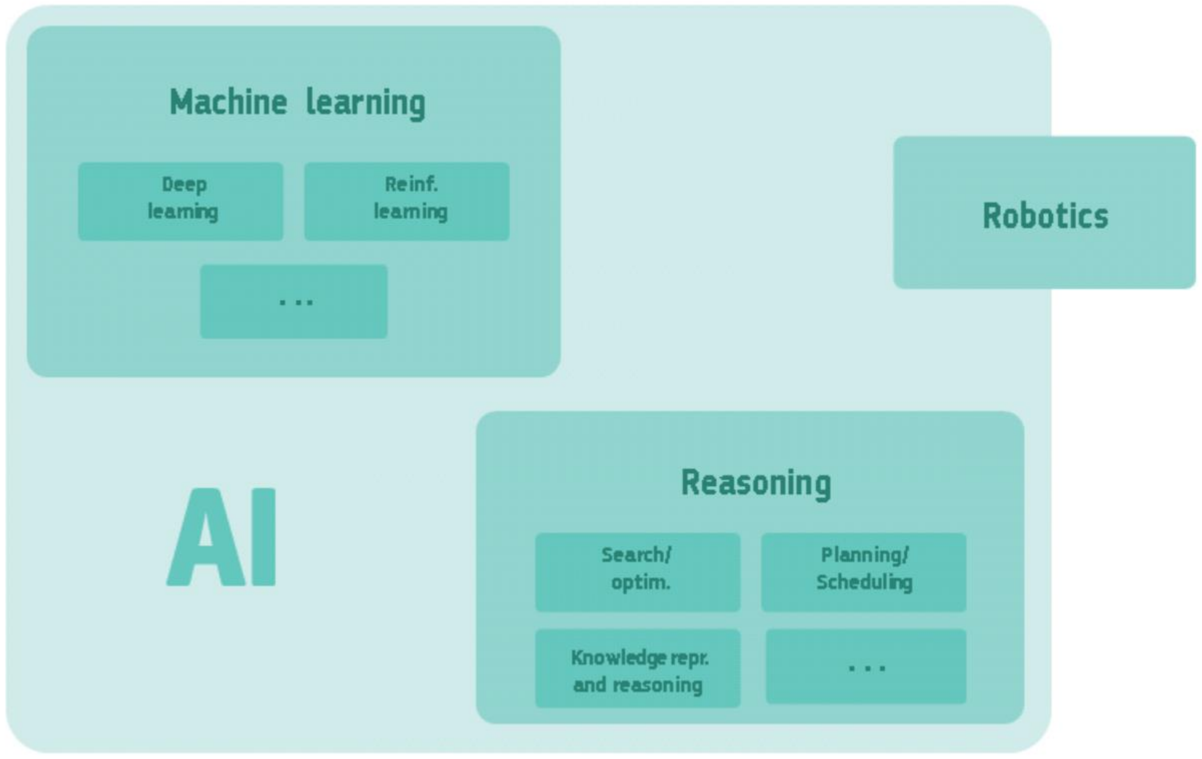
- Artificial intelligence (AI) refers to systems that display intelligent behavior by analyzing their environment and taking actions – with some degree of autonomy – to achieve specific goals.
- AI-based systems can be **purely software-based**, acting in the virtual world (e.g. voice assistants, image analysis software, search engines, speech and face recognition systems) or AI can be **embedded in hardware devices** (e.g. advanced robots, autonomous cars, drones or Internet of Things applications).

High-Level Expert Group on Artificial Intelligence: A definition of AI: main capabilities and scientific disciplines
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651

A simplified overview of AI’s sub-disciplines

Both machine learning and reasoning include many other techniques, and robotics includes techniques that are outside AI.

The whole of AI falls within the computer science discipline.



High-Level Expert Group on Artificial Intelligence: A definition of AI: main capabilities and scientific disciplines
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651

AI as a scientific discipline

- As a scientific discipline, AI includes several approaches and techniques, such as
  - machine learning (of which deep learning and reinforcement learning are specific examples),
  - machine reasoning (which includes planning, scheduling, knowledge representation and reasoning, search, and optimization),
  - and robotics (which includes control, perception, sensors and actuators, as well as the integration of all other techniques into cyber-physical systems).

High-Level Expert Group on Artificial Intelligence: A definition of AI: main capabilities and scientific disciplines
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651

AI system

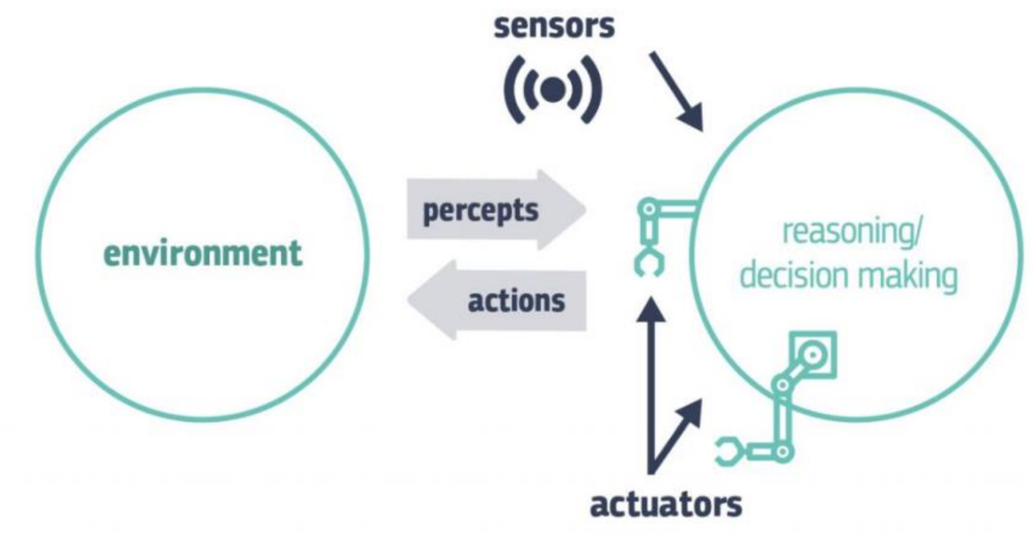


Figure 1: Schematic depiction of an AI system.

- AI system: any AI-based component, software and/or hardware.
- AI systems are (usually) embedded as components of larger systems, rather than standalone systems.

High-Level Expert Group on Artificial Intelligence: A definition of AI: main capabilities and scientific disciplines
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651

AI system

- “Artificial intelligence (AI) systems are software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by
  - perceiving their environment through data acquisition,
  - interpreting the collected structured or unstructured data,
  - reasoning on the knowledge, or processing the information, derived from this data
  - and deciding the best action(s) to take to achieve the given goal.
- AI systems can either
  - use symbolic rules or
  - learn a numeric model, and
  - they can also adapt their behavior by analyzing how the environment is affected by their previous actions.

High-Level Expert Group on Artificial Intelligence: A definition of AI: main capabilities and scientific disciplines
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651

Trustworthy AI

- According to the *High-Level Expert Group on AI's* **Ethics Guidelines for Trustworthy Artificial Intelligence**, trustworthy AI should be:
- (1) lawful - respecting all applicable laws and regulations
- (2) ethical - respecting ethical principles and values
- (3) robust - both from a technical perspective while taking into account its social environment

High-Level Expert Group on Artificial Intelligence: Ethics guidelines for trustworthy AI." *B-1049 Brussels* (2019).
https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

An example of automated discrimination

- A computer program for screening job applicants was used during the 1970s and 1980s in St. George’s Hospital Medical School, London, UK.
- The program used information from applicants’ forms, without any reference to ethnicity.
- However, the program was found to **unfairly discriminate against ethnic minorities and women** by inferring this information from surnames and place of birth, and lowering their chances of being selected for interview.

Stella Lowry and Gordon Macpherson. 1988. A blot on the profession. Brit. Med. J. Clin. Res. 296, 6623 (1988), 657

Guidance on Trustworthy AI

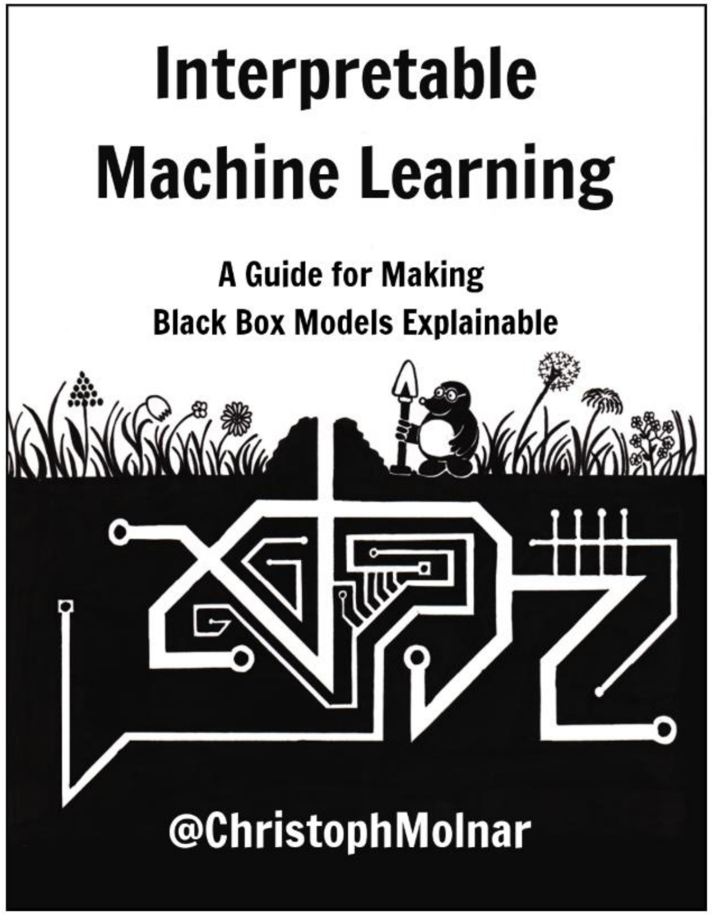
Ensure that the development, deployment and use of AI systems meets the seven key requirements for Trustworthy AI:

1. human agency and oversight,
2. technical robustness and safety,
3. privacy and data governance,
4. **transparency**,
5. diversity, non-discrimination and fairness,
6. environmental and societal well-being and
7. accountability.

Consider technical and non-technical methods to ensure the implementation of those requirements.

High-Level Expert Group on Artificial Intelligence: Ethics guidelines for trustworthy AI." *B-1049 Brussels* (2019).
https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

Transparency of ML



Molnar, Christoph. **Interpretable machine learning**. Lulu. com, 2019. https://christophm.github.io/interpretable-ml-book/

Black box

- A black box system can be viewed in terms of its inputs and outputs, **without any knowledge of its internal workings**.
- The opposite of a black box is a system where the inner components or logic are available for inspection, which is most commonly referred to as a **white box** (which can also come be called a "clear box" or a "glass box").

Interpreting black box systems

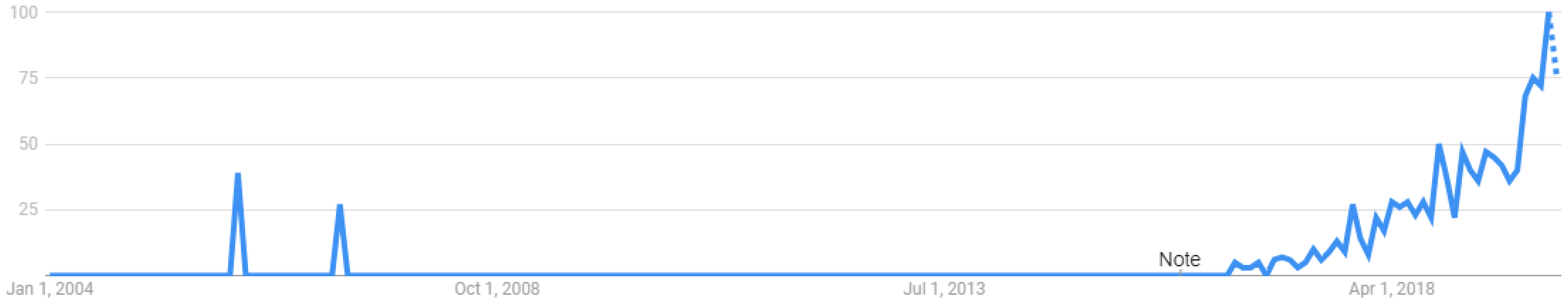
- What does it mean that a model is interpretable or transparent?
- What is an explanation?
- When is a model or an explanation comprehensible?
- Which is the best way to provide an explanation?
- Which are the problems requiring interpretable models/predictions?

- How much are we willing to lose in prediction accuracy to gain any form of interpretability?

XAI = Explainable AI

- XAI tends to refer to the movement, initiatives, and efforts made in response to AI transparency and trust concerns, more than to a formal technical concept

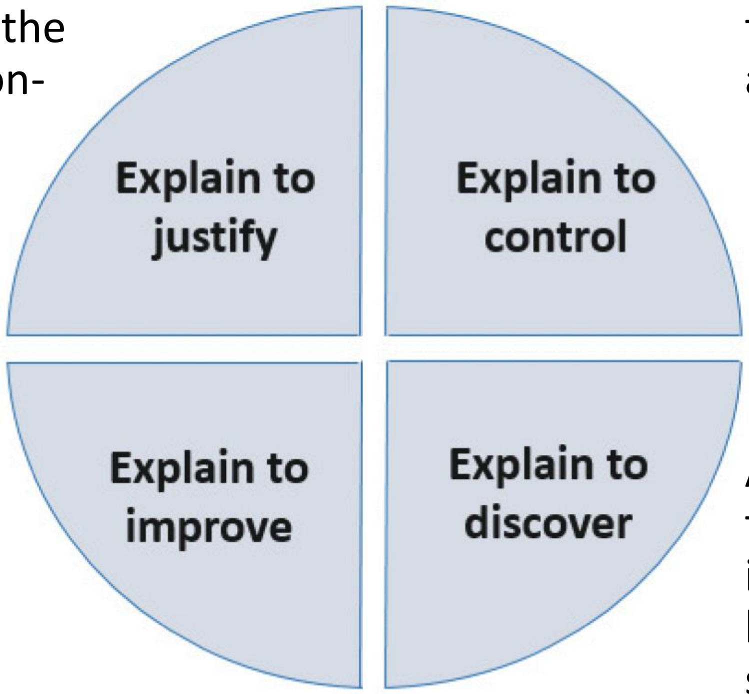
Google trends for “Explainable AI”



XAI: Need and Application Opportunities

Explanation for a decision: the need for reasons or **justifications for** that particular **outcome**, rather than a description of the inner workings or the logic of reasoning behind the decision-making process in general.

Understanding more about system behavior provides greater visibility over unknown vulnerabilities and flaws, and helps to rapidly identify and correct errors.



A model that can be explained and understood is one that can be more easily improved.

Asking for explanations is a helpful tool to learn new facts, to gather information and thus to gain knowledge. Only explainable systems can be useful for that.

Adadi, A., & Berrada, M. (2018). **Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)**. IEEE Access, 6, 52138-52160.

XAI methods

- **Intrinsic or post hoc?**

- **Model-specific or model-agnostic?**
  - Model-specific interpretation tools are limited to specific model classes.
  - Agnostic methods usually work by analyzing feature input and output pairs.

- **Local or global?**
  - Does the interpretation method explain an individual prediction or the entire model behavior?

Result of the interpretation method

- **Feature summary statistic**: feature importance, the pairwise feature interaction strengths…
- **Feature summary visualization**: Partial dependence plots are curves that show a feature and the average predicted outcome.
- **Model internals:** learned weights, tree, rules, …
- **Data point**: counterfactual explanations, ...
- **Intrinsically interpretable model**: approximate black box models (either globally or locally) with an interpretable model.

Shapley values

Molnar, Christoph. **Interpretable machine learning**. Lulu. com, 2019. https://christophm.github.io/interpretable-ml-book/

Chapter 5.9

Shapley values

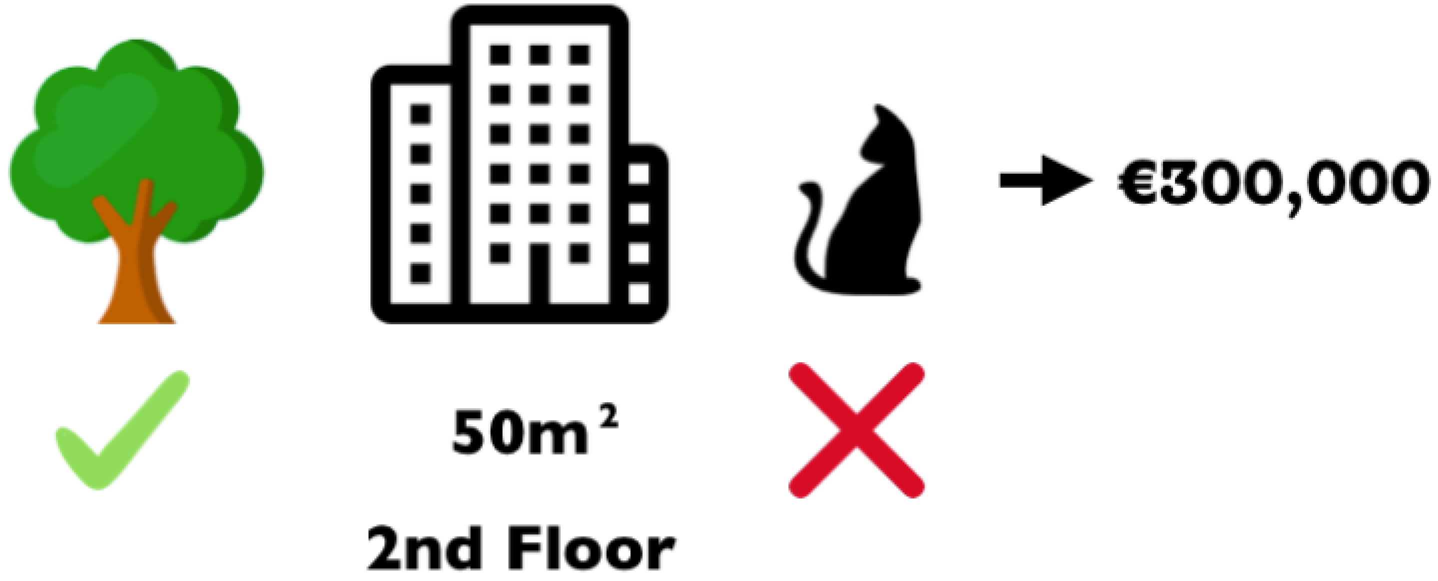
instance

Given the current set of feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction.

- Based on a method from coalitional game theory:

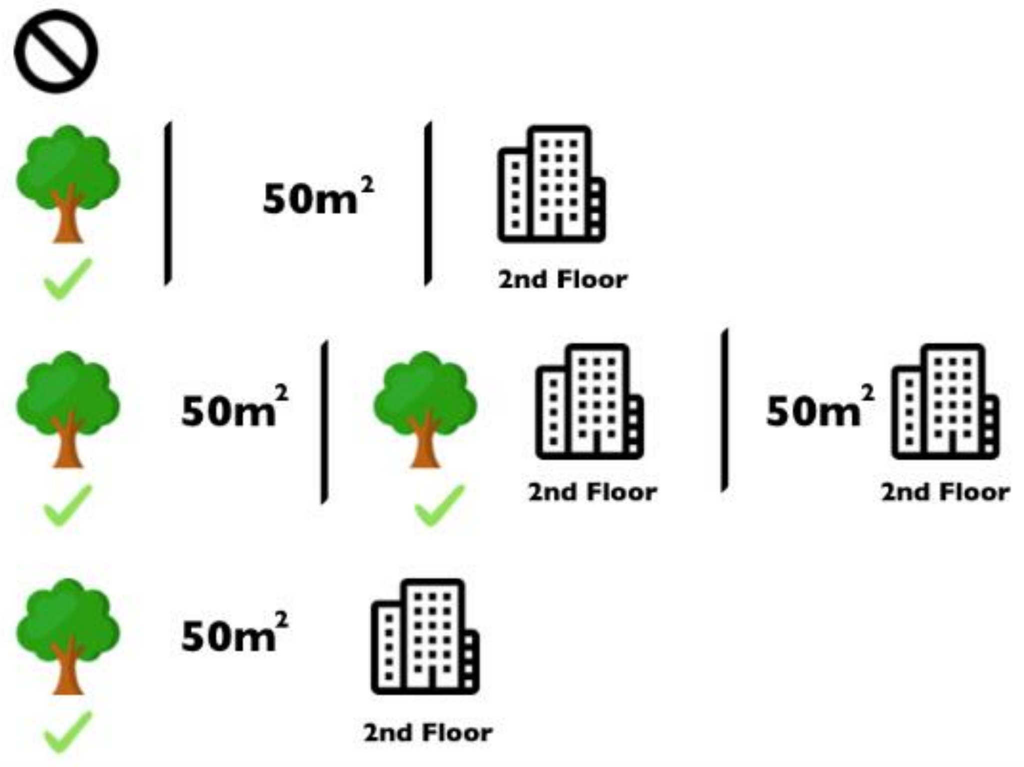
A group of differently skilled participants are all cooperating together for a collective reward. How should the reward be ***fairly*** *divided amongst the group?*

Shapley explained



The average prediction for all apartments is €310,000. How much has the feature value “cats not allowed” contributed to the prediction (compared to the average prediction)?

Computing Shapley : Coalitions



- 8 coalitions composed of the features in the instance “cats not allowed”
- For each coalition, compute the difference between model output of the coalition and model output of the coalition with the added feature value “cats not allowed”

Shapley values

The Shapley value is the average marginal contribution of a feature value across all possible coalitions.

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

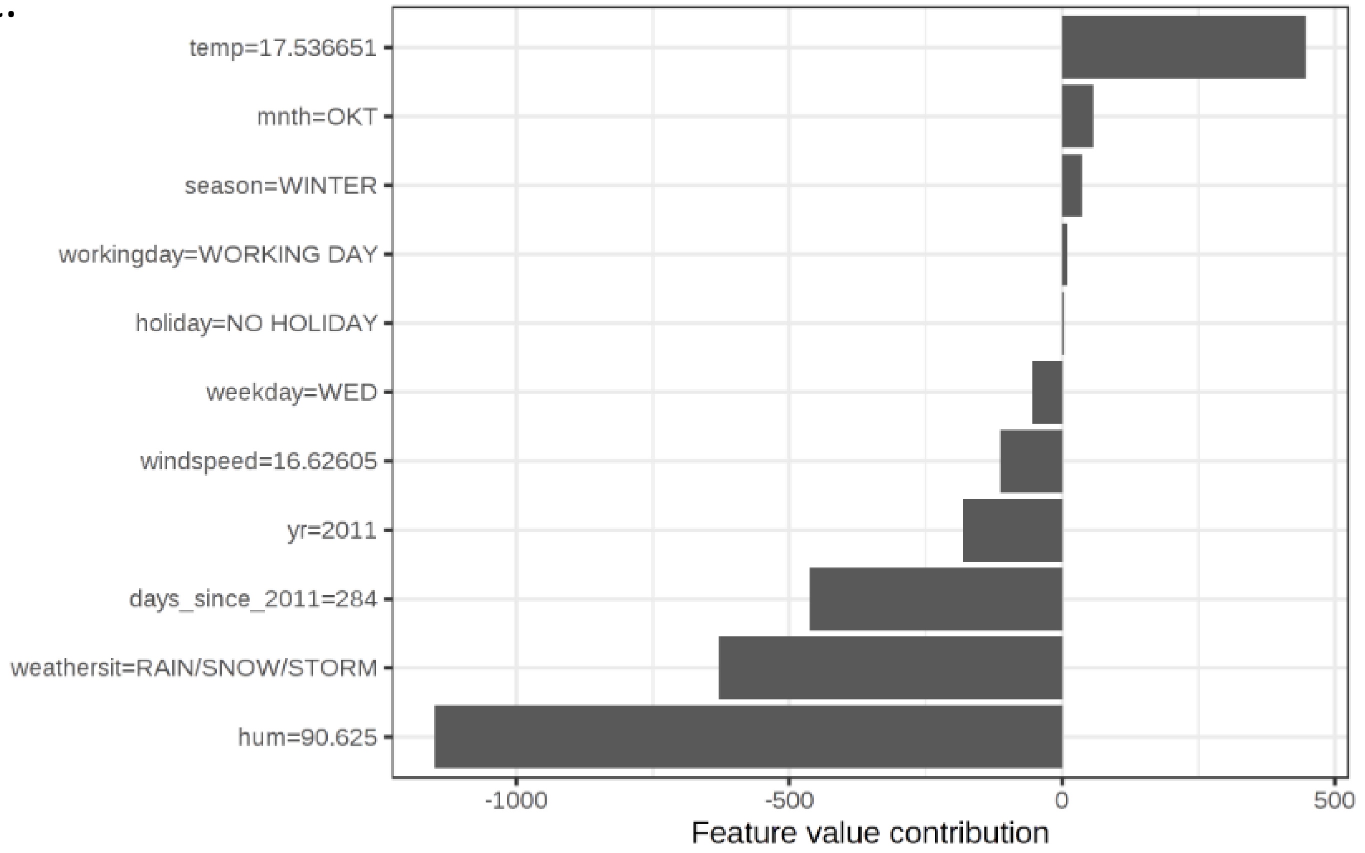
	Coalitional game	**Machine learning**
{x1,….,xp}	set of all players	Set of all attribute-value pairs (an instance)
i	The i-th player	The i-th feature-value pair
val	the function *val* gives the value (or payout) for any subset of those players	Predictive ML model

Shapley example

The interpretation of the Shapley value for feature value j is:

The value of the j-th feature contributed φj to the prediction of this particular instance compared to the average prediction for the dataset.

bike rental dataset

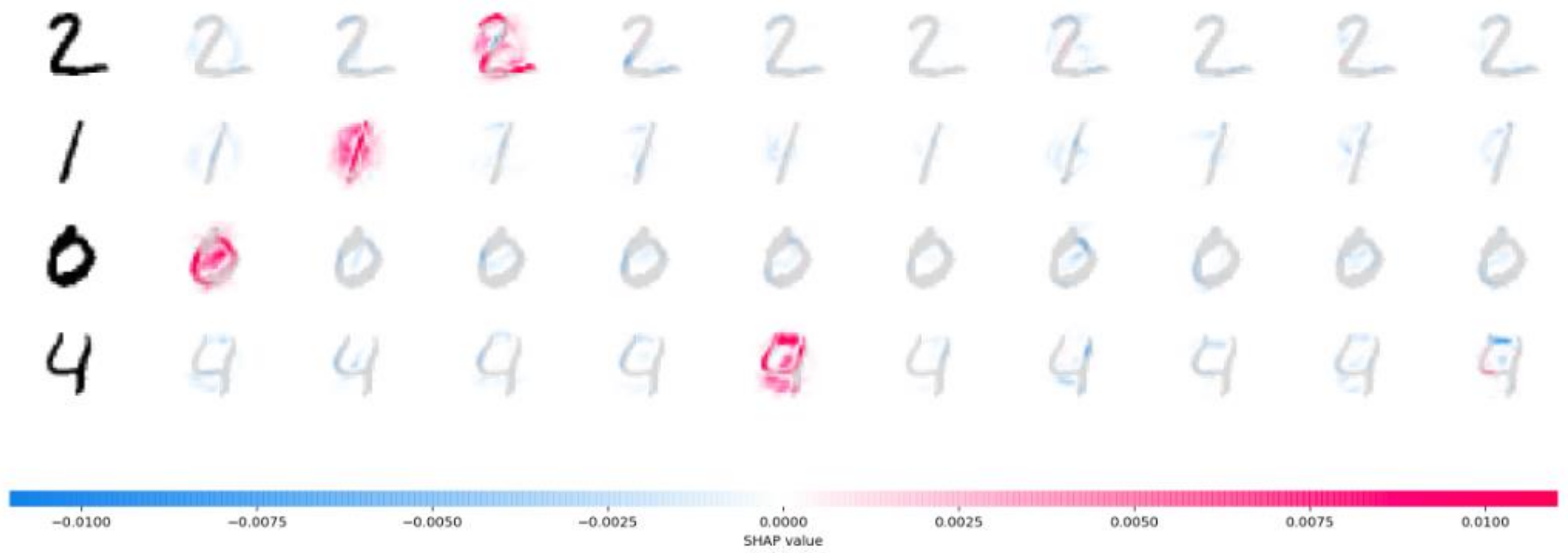


Shapley value summary

- The Shapley value is the average contribution of a feature value to the prediction in different coalitions.
- The Shapley value is NOT the difference in prediction when we would remove the feature from the model.
- The Shapley value works for both classification (if we are dealing with probabilities) and regression.
- **A lot of computing time**. In 99.9% of real-world problems, only the approximate solution is feasible.

- SHAP (SHapley Additive exPlanations) has a fast implementation for tree-based models (random forest,…)
- shap Python package

SHAP result example



The plot above explains ten outputs (digits 0-9) for four different images. Red pixels increase the model's output while blue pixels decrease the output. The input images are shown on the left, and as nearly transparent grayscale backings behind each of the explanations. The sum of the SHAP values equals the difference between the expected model output (averaged over the background dataset) and the current model output. Note that for the 'zero' image the blank middle is important, while for the 'four' image the lack of a connection on top makes it a four instead of a nine.

https://github.com/slundberg/shap

Example-based explanations

Counterfactual Explanations

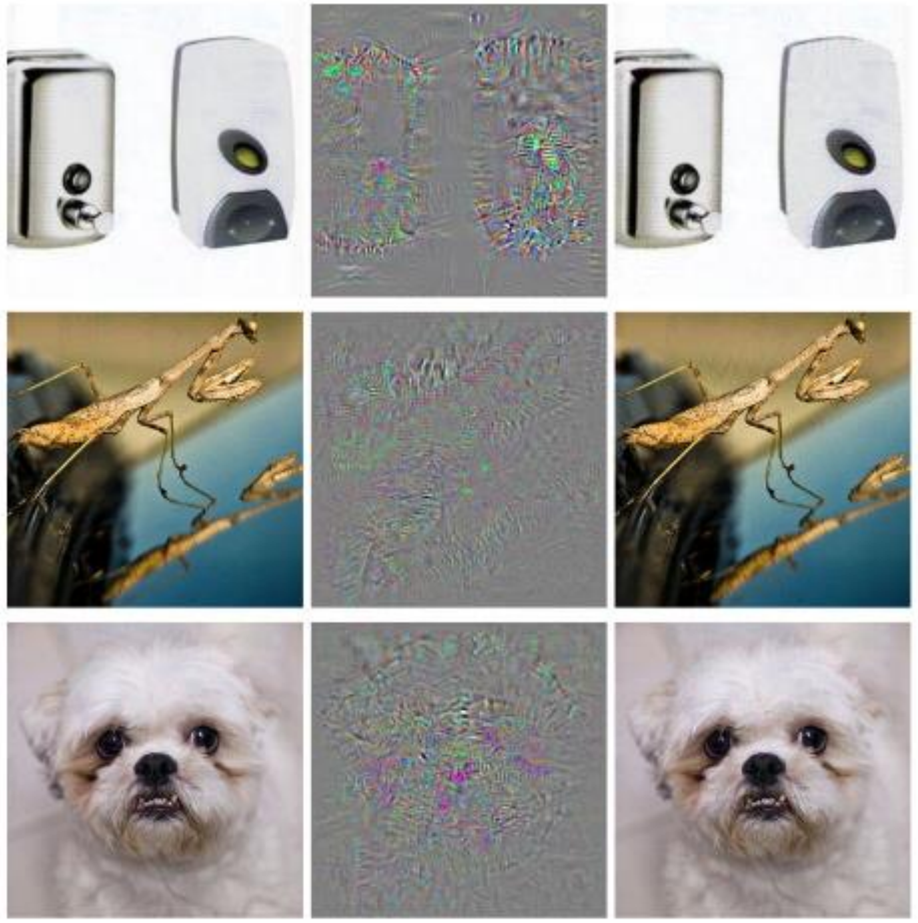
- A counterfactual explanation of a prediction describes **the smallest change to the feature values that changes the prediction** to a predefined output.

- "If X had not occurred, Y would not have occurred"

- Model-agnostic (it only works with the model inputs and output)
- Explain predictions of individual instances

Adversarial Examples

An adversarial example is an instance with small, intentional feature perturbations that cause a machine learning model to make a false prediction.



Adversarial examples for AlexNet by Szegedy et. al (2013). All images in the left column are correctly classified. The middle column shows the (magnified) error added to the images to produce the images in the right column all categorized (incorrectly) as ‘Ostrich’.

Literature

- Molnar, Christoph. **Interpretable machine learning**. Lulu. com, 2019. https://christophm.github.io/interpretable-ml-book/
- Adadi, A., & Berrada, M. (2018). **Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)**. IEEE Access, 6, 52138-52160.
- Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). **A survey of methods for explaining black box models**. ACM computing surveys (CSUR), 51(5), 1-42.
- High-Level Expert Group on Artificial Intelligence: **A definition of AI: main capabilities and scientific disciplines** https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60651
- High-Level Expert Group on Artificial Intelligence: **Ethics guidelines for trustworthy AI**." B-1049 Brussels (2019). https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai