Week 4 Unit 1: **Classification Analysis with Decision Trees**

openSAP
open.sap.com

Classification Analysis with Decision Trees

Play golf?

- Play Golf?
  - Windy? (Yes)
    - Outlook? (Rain) — Don’t Play
    - Outlook? (Sunny, O/cast)
      - Humidity? (Low) — Play
      - Humidity? (High)
  - Windy? (No)
    - Outlook? (O/cast, Rain) — Play
    - Outlook? (Sunny)
      - Humidity? (Low) — Play
      - Humidity? (High) — Don’t Play

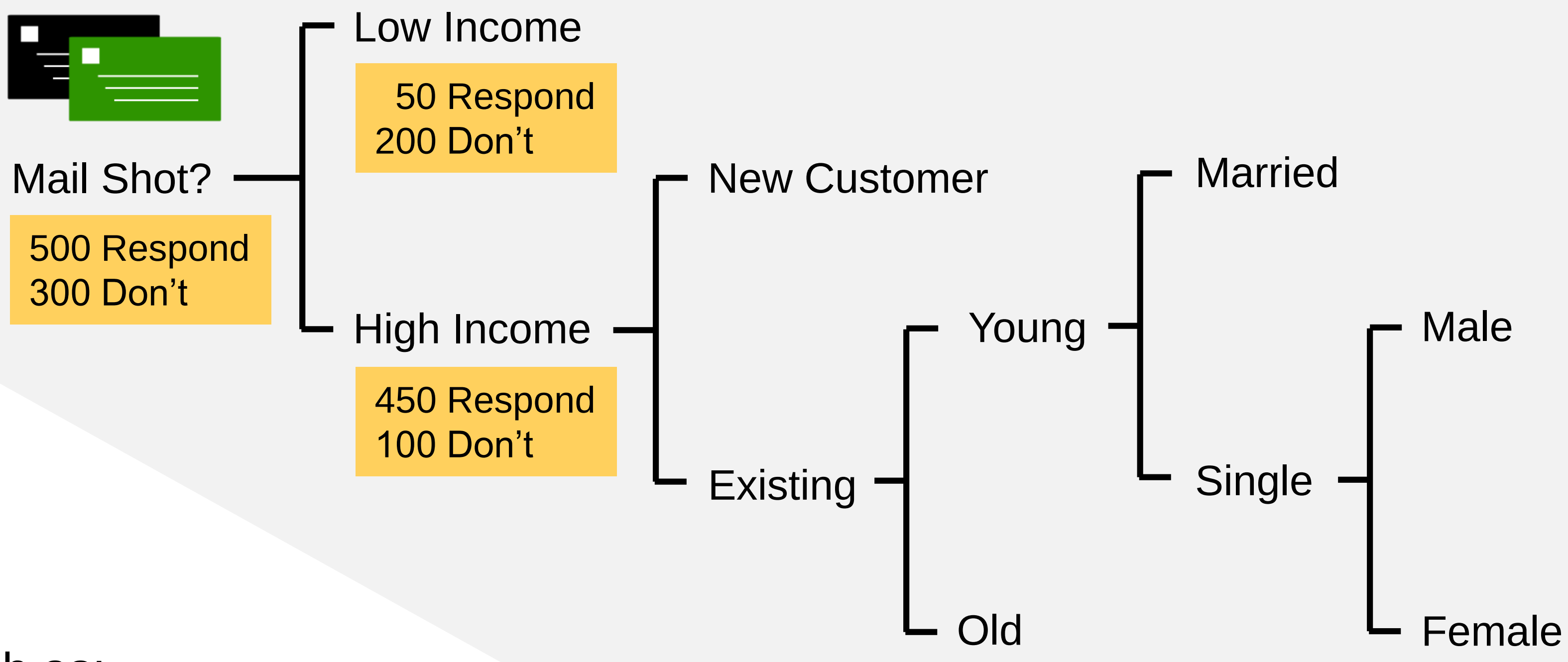
From which we can derive the rules:

Don't play if the outlook is rain and it’s windy
Don’t play if humidity is high, outlook sunny, and it’s not windy

Otherwise, go play!

Classification Analysis with Decision Trees

Mail shot campaign

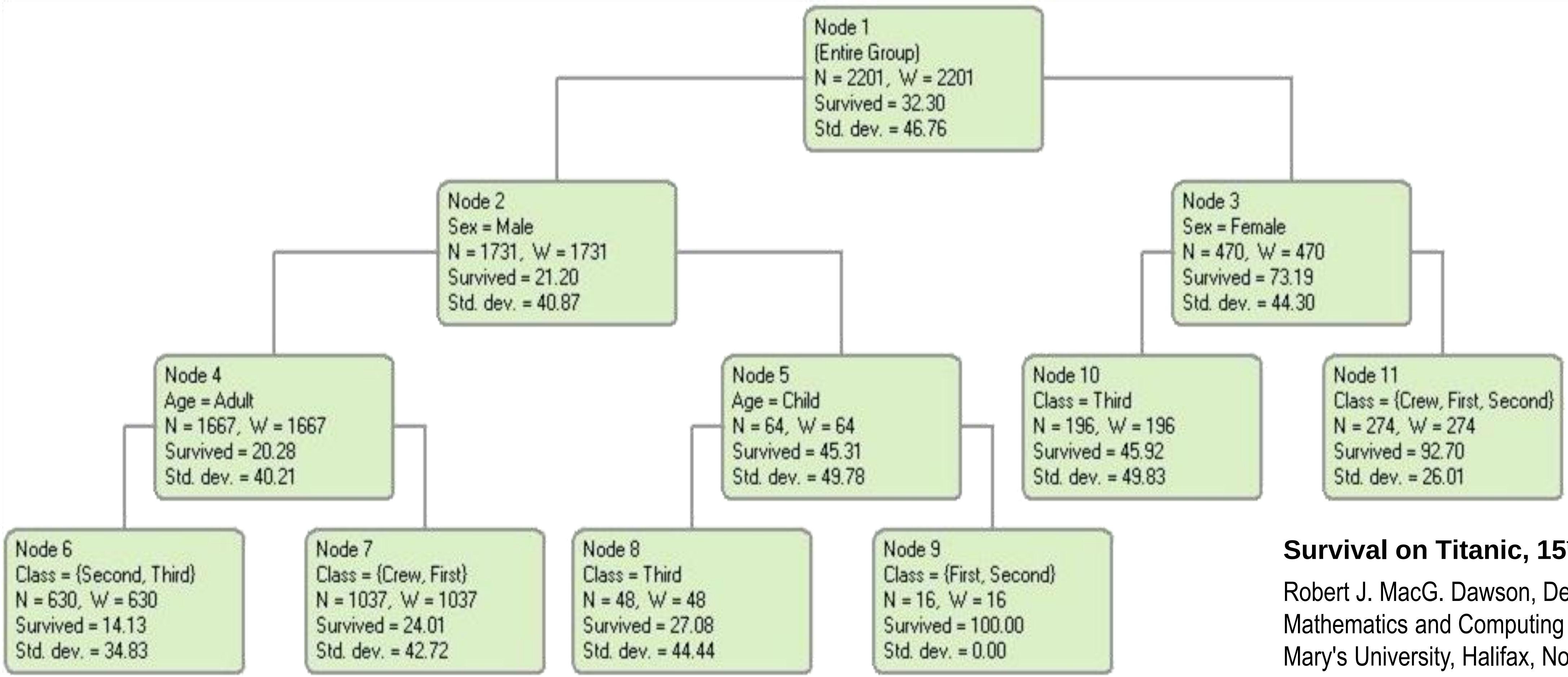


From which we might derive a rule such as:

Young, single females who are existing customers with a high income are most likely to respond to our mail shot.

Classification Analysis with Decision Trees

Titanic



Survival on Titanic, 15th April 1912

Robert J. MacG. Dawson, Department of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, Canada

Classification Analysis with Decision Trees

Play golf: Chi-square automatic interaction detector (CHAID) example

Outlook	Temp (°F)	Humidity (%)	Windy?	Class
Sunny	75	70	Yes	Play
Sunny	80	90	Yes	Don't Play
Sunny	85	85	No	Don't Play
Sunny	72	95	No	Don't Play
Sunny	69	70	No	Play
Overcast	72	90	Yes	Play
Overcast	83	78	No	Play
Overcast	64	65	Yes	Play
Overcast	81	75	No	Play
Rain	71	80	Yes	Don't Play
Rain	65	70	Yes	Don't Play
Rain	75	80	No	Play
Rain	68	80	No	Play
Rain	70	96	No	Play

Classification Analysis with Decision Trees

Play golf: CHAID example

Scenario	Outlook	Temp (°F)	Humidity (%)	Windy?	Class
1	Sunny	2	1	Yes	Play
2	Sunny	2	2	Yes	Don't Play
3	Sunny	3	2	No	Don't Play
4	Sunny	2	2	No	Don't Play
5	Sunny	1	1	No	Play
6	Overcast	2	2	Yes	Play
7	Overcast	3	2	No	Play
8	Overcast	1	1	Yes	Play
9	Overcast	3	1	No	Play
10	Rain	2	2	Yes	Don't Play
11	Rain	1	1	Yes	Don't Play
12	Rain	2	2	No	Play
13	Rain	1	1	No	Play
14	Rain	1	2	No	Play

<= 70 is 1 >= 71 and <= 80 is 2 >= 81 is 3	<= 75 is 1 > 75 is 2

Classification Analysis with Decision Trees

Play golf: CHAID example

Outlook	Observed		
	Play	Don't	
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5
	9	5	14

Outlook	Expected		
	Play	Don't	
Sunny	3.21	1.79	5.00
Overcast	2.57	1.43	4.00
Rain	3.21	1.79	5.00
	9.00	5.00	14.00

Chi Squared			
	Play	Don't	
Sunny	0.46	0.83	1.2844
Overcast	0.79	1.43	2.2222
Rain	0.01	0.03	0.0400
	1.2667	2.2800	3.5467

Chi Squared	3.5467
Degrees of Freedom	2
Chi Test	0.1698

Temp.	Observed		
	Play	Don't	
1	4	1	5
2	3	3	6
3	2	1	3
	9	5	14

Temp.	Expected		
	Play	Don't	
1	3.21	1.79	5.00
2	3.86	2.14	6.00
3	1.93	1.07	3.00
	9.00	5.00	14.00

Chi Squared			
	Play	Don't	
1	0.19	0.35	0.5378
2	0.19	0.34	0.5333
3	0.00	0.00	0.0074
	0.3852	0.6933	1.0785

Chi Squared	1.0785
Degrees of Freedom	2
Chi Test	0.5832

Humidity	Observed		
	Play	Don't	
1	4	1	9
2	5	4	5
	9	5	14

Humuidity	Expected		
	Play	Don't	
1	5.79	3.21	9.00
2	3.21	1.79	5.00
	9.00	5.00	14.00

Chi Squared			
	Play	Don't	
1	0.55	1.53	2.0765
2	0.99	2.75	3.7378
	1.5432	4.2711	5.8143

Chi Squared	0.8365
Degrees of Freedom	1
Chi Test	0.3604

Windy	Observed		
	Play	Don't	
Yes	3	3	6
No	6	2	8
	9	5	14

Windy	Expected		
	Play	Don't	
Yes	3.86	2.14	6.00
No	5.14	2.86	8.00
	9.00	5.00	14.00

Chi Squared			
	Play	Don't	
Yes	0.19	0.34	0.5333
No	0.14	0.26	0.4000
	0.3333	0.6000	0.9333

Chi Squared	0.9333
Degrees of Freedom	1
Chi Test	0.3340

Most significant split is Outlook

If you look at the observed values for Outlook, you will note that the 4/0 split for Overcast is the clearest split and contributes most to the chi-squared calculation.

Classification Analysis with Decision Trees

Play golf: CHAID example

Classification Analysis with Decision Trees

Play golf: CHAID example

For Outlook = Sunny

Temp	**Observed**		
	Play	Don't	
1	1	0	1
2	1	2	3
3	0	1	1
	2	3	5

Temp	**Expected**		
	Play	Don't	
1	0.40	0.60	1.00
2	1.20	1.80	3.00
3	0.40	0.60	1.00
	2.00	3.00	5.00

Chi Squared			
	Play	Don't	
1	0.90	0.60	1.5000
2	0.03	0.02	0.0556
3	0.40	0.27	0.6667
	1.3333	0.8889	2.2222

Chi Squared	2.2222
Degrees of Freedom	2
Chi Test	0.3292

Humidity	**Observed**		
	Play	Don't	
1	0	3	3
2	2	0	2
	2	3	5

Humidity	**Expected**		
	Play	Don't	
1	1.20	1.80	3.00
2	0.80	1.20	2.00
	2.00	3.00	5.00

Chi Squared			
	Play	Don't	
1	1.20	0.80	2.0000
2	1.80	1.20	3.0000
	3.0000	2.0000	5.0000

Chi Squared	5.0000
Degrees of Freedom	1
Chi Test	0.0253

Most significant

Windy	**Observed**		
	Play	Don't	
Yes	1	1	2
No	1	2	3
	2	3	5

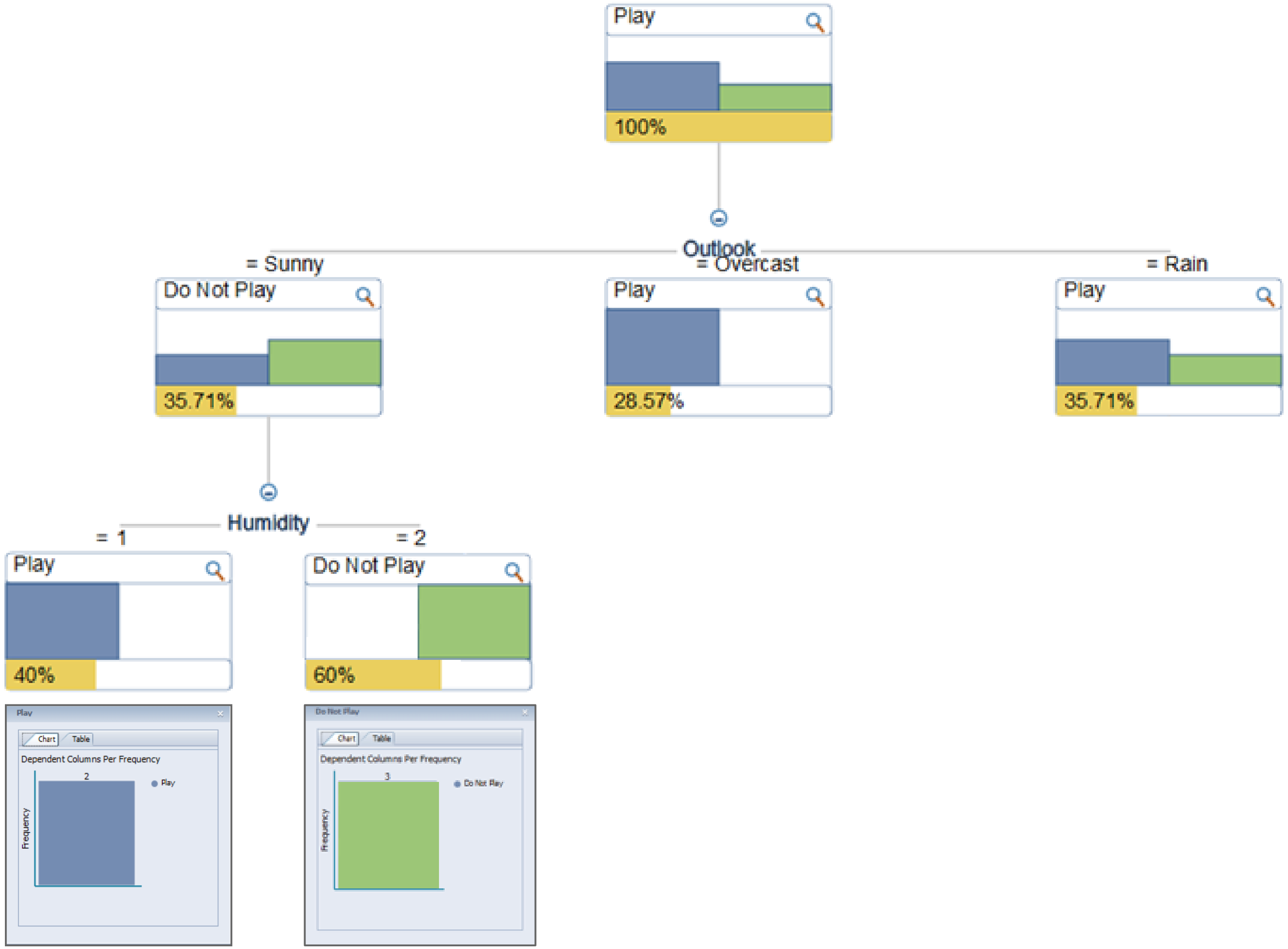
Windy	**Expected**		
	Play	Don't	
Yes	0.80	1.20	2.00
No	1.20	1.80	3.00
	2.00	3.00	5.00

Chi Squared			
	Play	Don't	
Yes	0.05	0.03	0.0833
No	0.03	0.02	0.0556
	0.0833	0.0556	0.1389

Chi Squared	0.1389
Degrees of Freedom	1
Chi Test	0.7094

Classification Analysis with Decision Trees

Play golf: CHAID example



Classification Analysis with Decision Trees
Confusion matrix

Actual

```
SELECT * FROM PAL_TRAINING_TAB
```

	OUTLOOK	TEMP	HUMIDITY	Windy	CLASSLABEL
1	Sunny	2	2	Yes	Play
2	Sunny	2	1	Yes	Do Not Play
3	Sunny	3	1	No	Do Not Play
4	Sunny	2	1	No	Do Not Play
5	Sunny	1	2	No	Play
6	Overcast	2	1	Yes	Play
7	Overcast	3	1	No	Play
8	Overcast	1	2	Yes	Play
9	Overcast	3	2	No	Play
10	Rain	2	1	Yes	Do Not Play
11	Rain	1	2	Yes	Do Not Play
12	Rain	2	1	No	Play
13	Rain	1	1	No	Play
14	Rain	1	1	No	Play

Predicted

```
SELECT * FROM PAL_RESULT_TAB
```

	ID	CLASSLABEL
1	1	Play
2	2	Do Not Play
3	3	Do Not Play
4	4	Do Not Play
5	5	Play
6	6	Play
7	7	Play
8	8	Play
9	9	Play
10	10	Do Not Play
11	11	Do Not Play
12	12	Play
13	13	Play
14	14	Play

Actual and Predicted Values for Golf Example

Confusion Matrix

2 X 2 Matrix **Model Accuracy: 0.92**

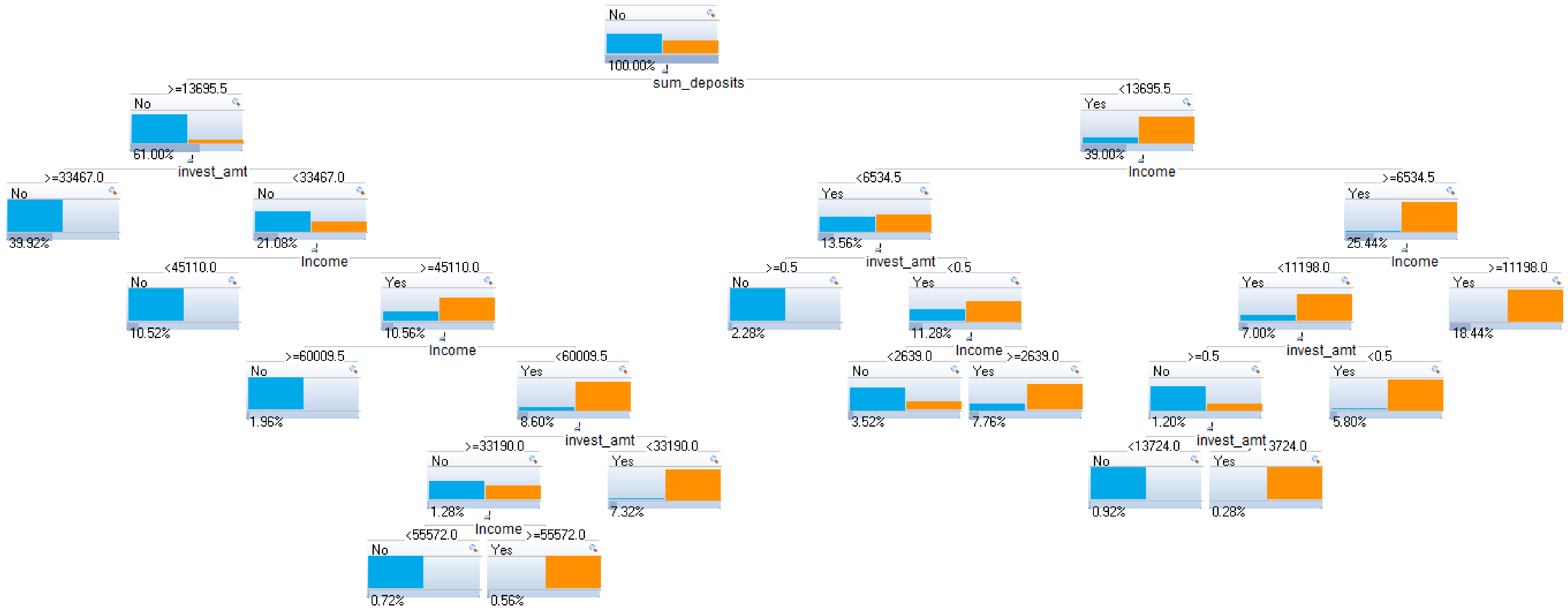
Actual → Predicted ↓	0	1
0	741	55
1	27	177

Derivatives

Derivative → Class ↓	Precision	Sensitivity	Specificity	Negative Prediction
0	0.96	0.93	0.87	0.76
1	0.76	0.87	0.93	0.96

Classification Analysis with Decision Trees

Realistic example



Classification Analysis with Decision Trees

Summary

- **Strengths**
  - The tree-type output is very visual and easy to understand
  - They are able to produce ‘understandable’ rules
  - They can perform classification without requiring much computation
  - They can handle both continuous and categorical variables
  - They provide a clear indication of variable importance
- **Weaknesses**
  - Clearly sensitive to the ‘first split’
  - Some algorithms require binary target variables
  - They can be computationally expensive
  - They generally examine just a single field at a time
  - They are prone to over-fitting

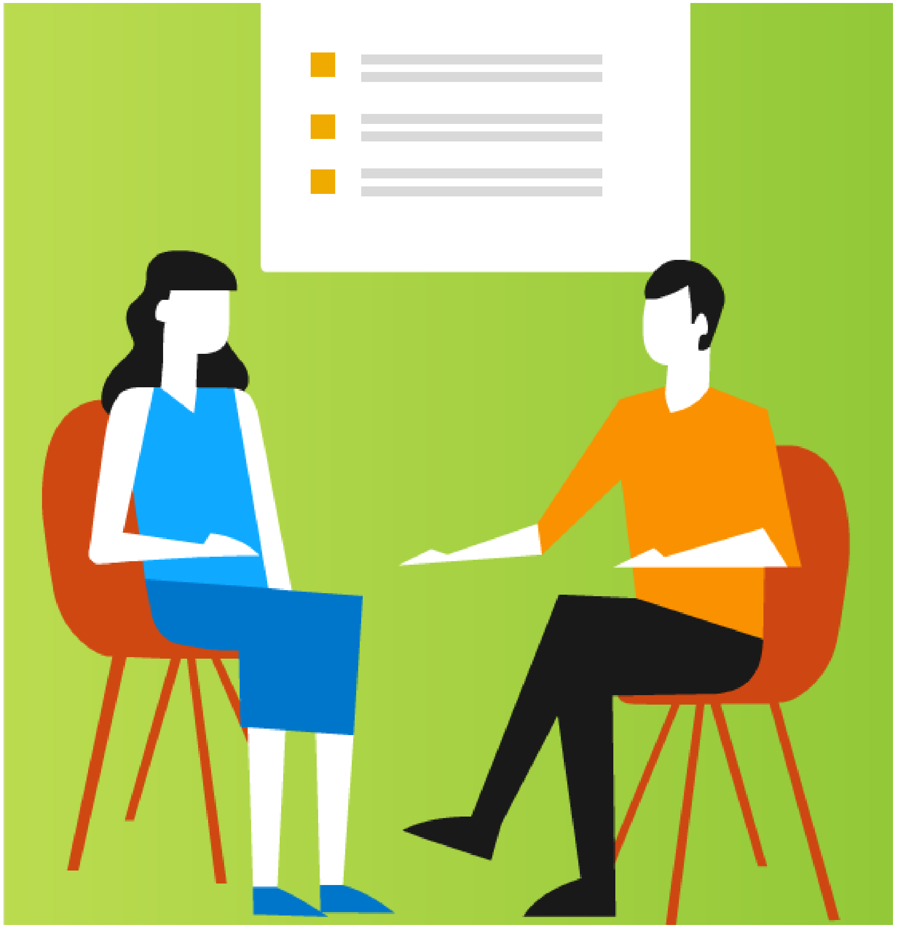
Thank you

Contact information:

open@sap.com

Additional Material

- Decision Trees
  - C4.5 algorithm
  - CNR tree
  - Random forests



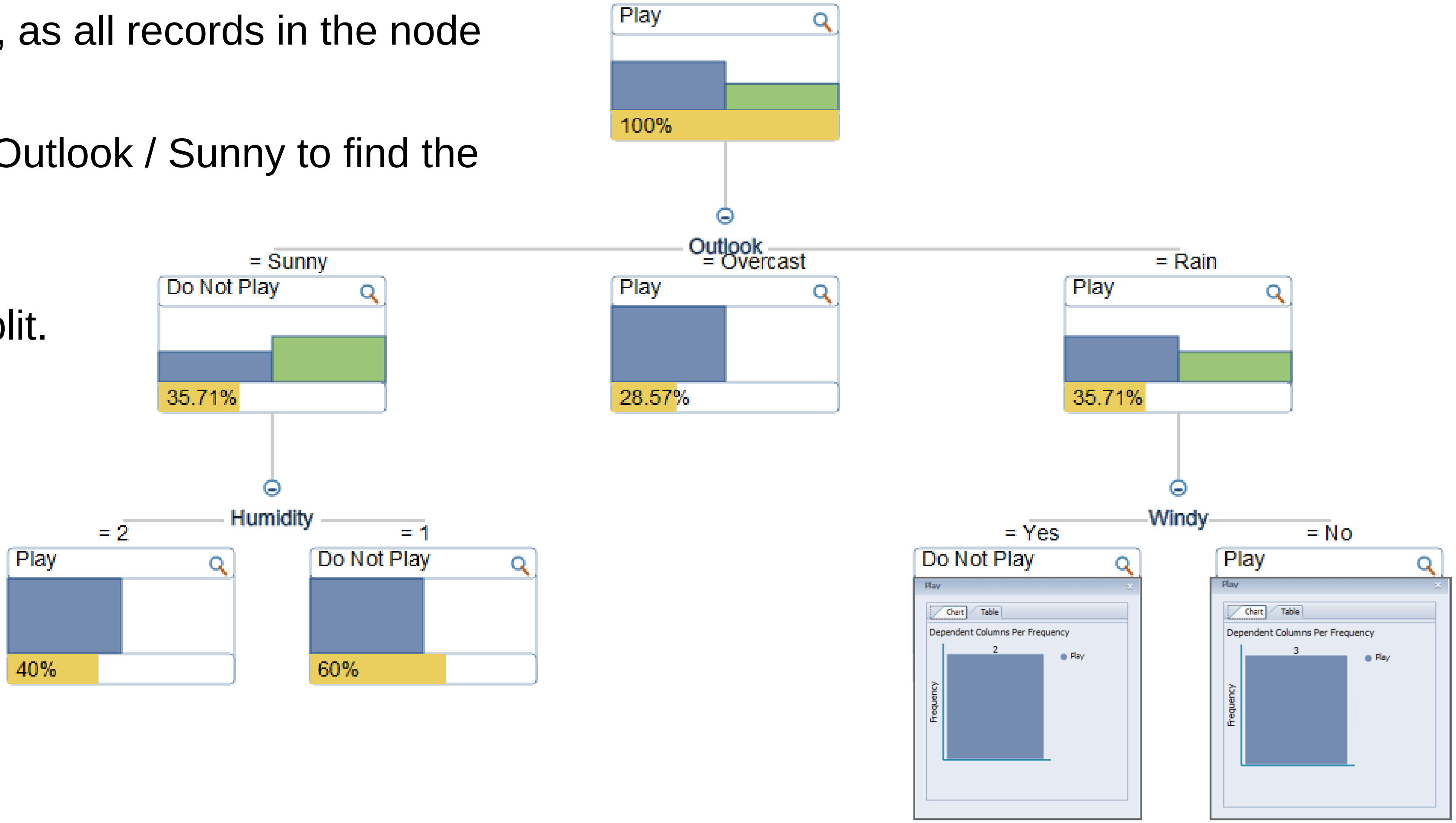
Classification Analysis with Decision Trees
C4.5 algorithm

- C4.5 is another classification tree algorithm that works by splitting the data based on the variable that provides the maximum information gain to the model from the split. Information is measured using a concept called **entropy**.
- Entropy is a concept from information theory and is a measure of information content, using a scale 1 – completely uncertain, to 0 – completely certain.
- For example, a coin has heads on one side, tails on the other and there is 50/50 probability of heads or tails when the coin is tossed. The entropy is defined as 1. The outcome is completely uncertain – it could be heads or tails. If the probability is 25/75 then the entropy is a little lower – we're less uncertain. If the coin had heads on both sides, the outcome would be completely certain. The entropy is then defined as 0.
- The goal in these decision trees is to get a very low entropy in order to make the most accurate decisions and classifications.
- In order to calculate the entropy, the following formula is used $E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$

Classification Analysis with Decision Trees

C4.5 algorithm: Play golf?

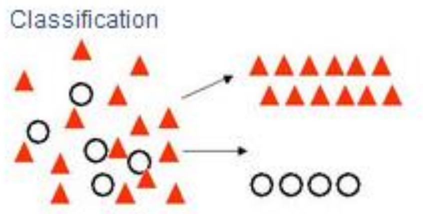
- Outlook, Overcast is a leaf node, as all records in the node are Play. The entropy is zero.
- Then we repeat the process for Outlook / Sunny to find the best split.
- Then we repeat the process for Outlook / Rain to find the best split.



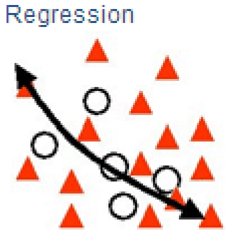
Classification Analysis with Decision Trees

CNR tree

- Another classification variation is CNR tree or C&RT, sometimes simply referred to as CART.
- This uses recursive partitioning to split the training records into segments with similar output field values.
- The C&R tree node starts by examining the input fields to find the best split, measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is subsequently split into two more subgroups, and so on, until one of the stopping criteria is triggered.
- All splits are binary, i.e. only two subgroups are created at each split.
- C&RT uses classification trees for discrete outcomes and regression trees for continuous outcomes:
  - **Classification Trees**: Where the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall



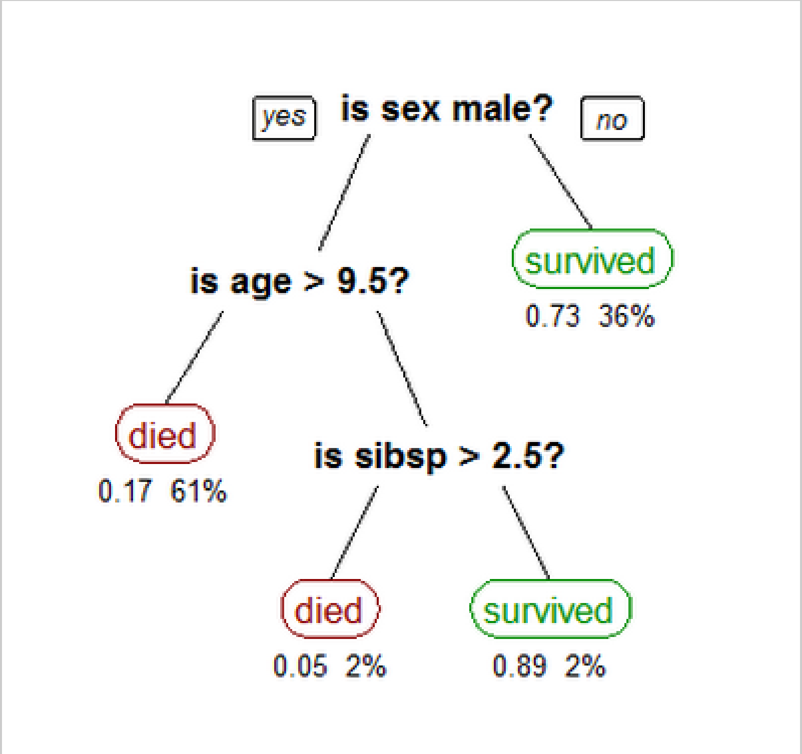
  - **Regression Trees**: Where the target variable is continuous and the tree is used to predict its value



Classification Analysis with Decision Trees

CNR tree

- CART is structured as a sequence of questions.
- The answers to the questions determine what the next question (if any) should be.
- The result of these questions is a tree-like structure where the ends are terminal nodes, at which point there are no more questions.



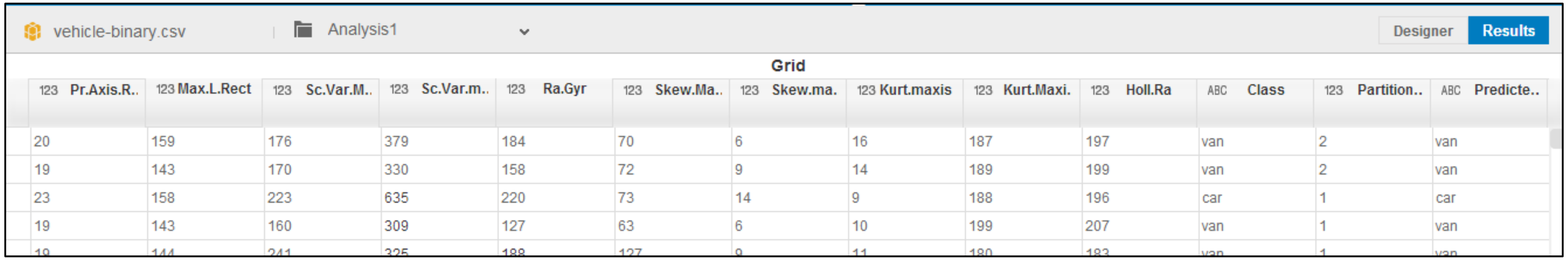
A tree showing survival of passengers on the Titanic ("sibsp" is the number of spouses or siblings aboard). The figures under the leaves show the probability of survival and the percentage of observations in the leaf.

Source: Wikipedia

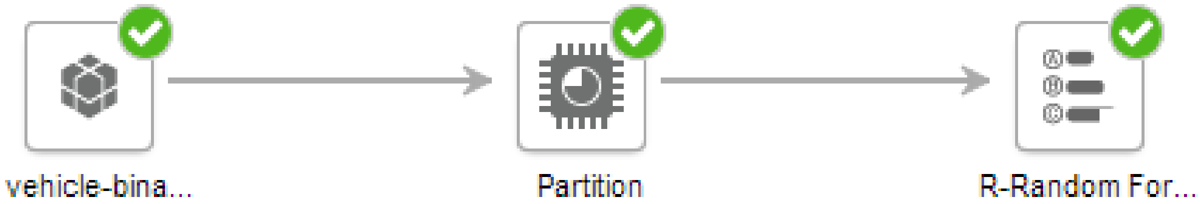
Classification Analysis with Decision Trees

Random forests

- Random forests are an ensemble learning method for classification, regression, and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Random decision forests correct decision trees' habit of over-fitting to their training set.
- Ensemble methods generally provide more robust models.
- Example in PA – various characteristics of a vehicle that have been collected for two types of vehicles: cars and vans…



Pr.Axis.R..	Max.L.Rect	Sc.Var.M..	Sc.Var.m..	Ra.Gyr	Skew.Ma..	Skew.ma.	Kurt.maxis	Kurt.Maxi.	Holl.Ra	Class	Partition..	Predicte..
20	159	176	379	184	70	6	16	187	197	van	2	van
19	143	170	330	158	72	9	14	189	199	van	2	van
23	158	223	635	220	73	14	9	188	196	car	1	car
19	143	160	309	127	63	6	10	199	207	van	1	van



Week 4 Unit 2: **Classification Analysis with KNN, NN, and SVM**

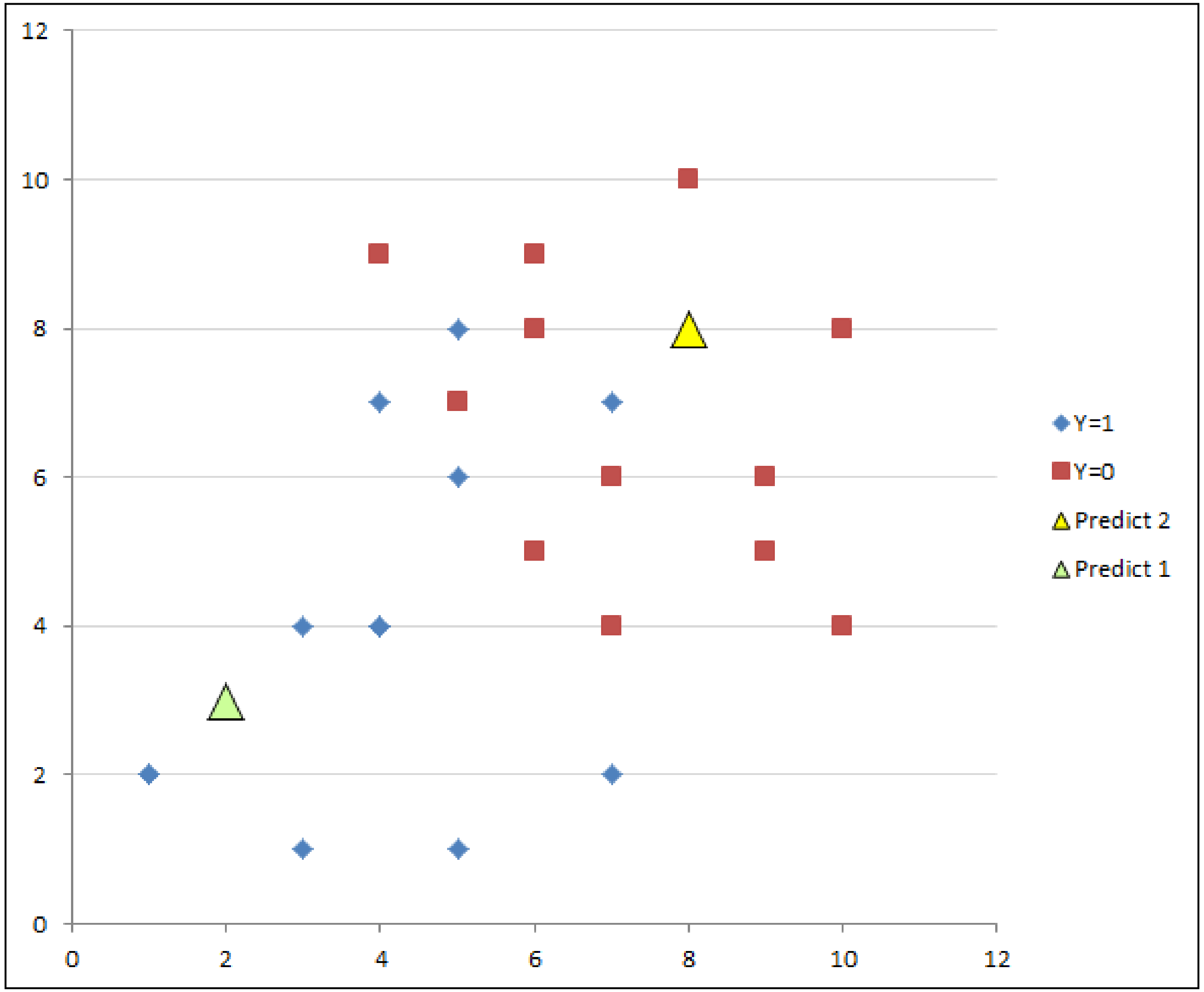
openSAP
open.sap.com

Contents

- The k-nearest neighbor algorithm **KNN**
  - Introduction
  - Worked example (see Appendix)
  - Summary

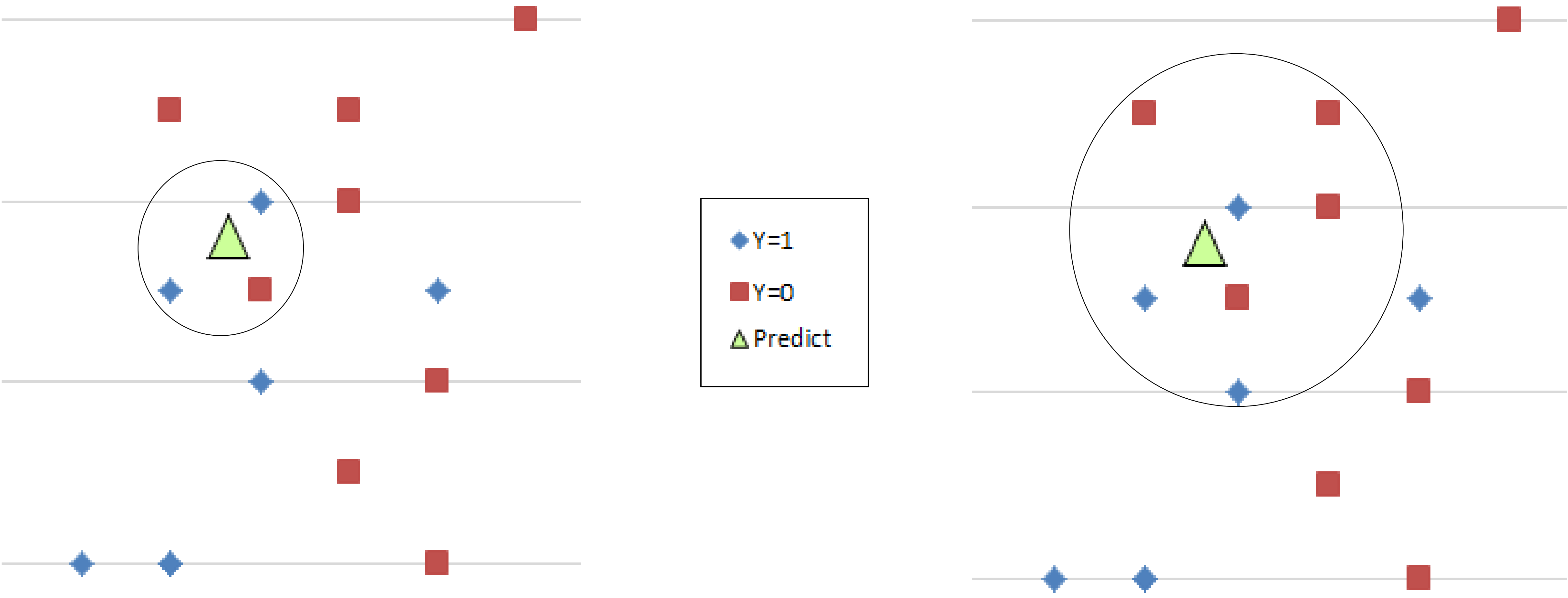
Classification Analysis with KNN, NN, and SVM

The k-nearest neighbor algorithm: introduction



Classification Analysis with KNN, NN, and SVM

The k-nearest neighbor algorithm



Classification Analysis with KNN, NN, and SVM

The k-nearest neighbor algorithm: summary

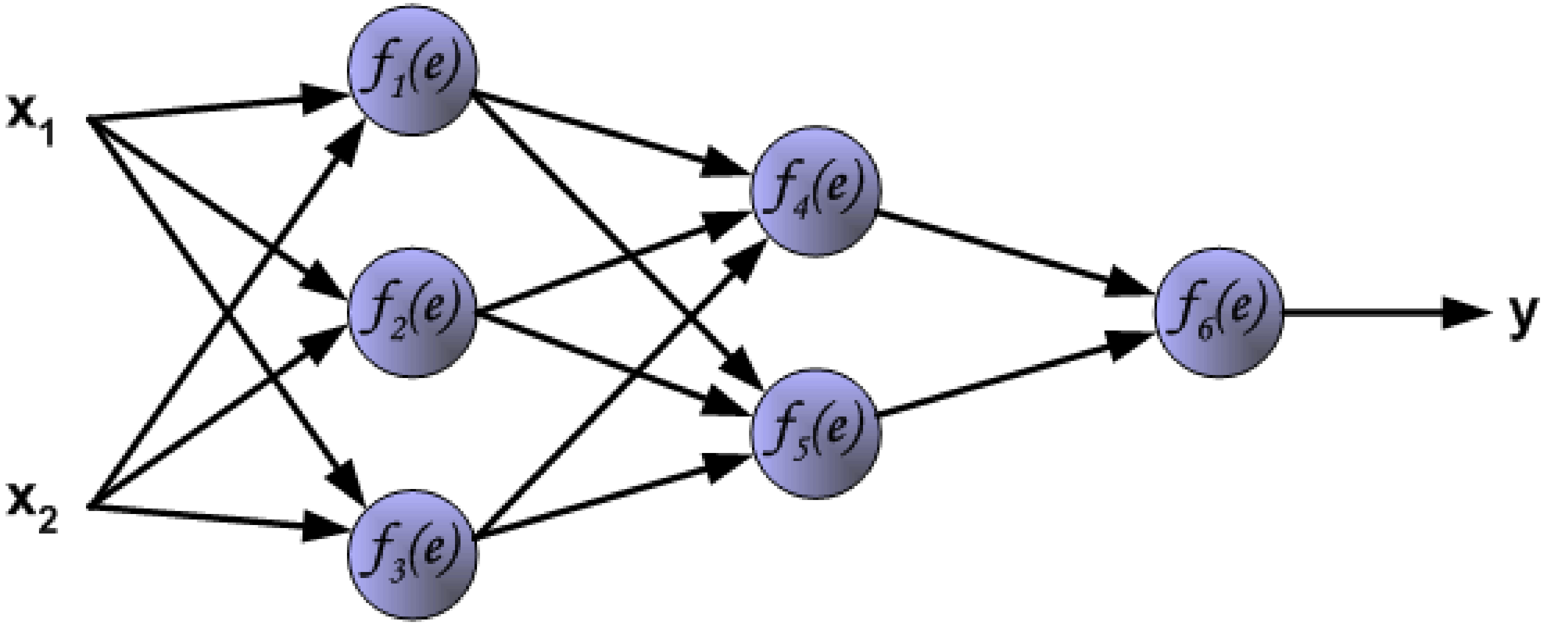
- **Strengths**
  - It is beautifully simple and logical
- **Weaknesses**
  - It may be driven by the choice of k, which may be a bad choice.
  - Generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct.
  - The accuracy of the algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance.
  - In binary (two class) classification problems, it is helpful to choose k to be an odd number, as this avoids tied votes.
  - It is important to review the sensitivity of the solution to different values of k.

Contents

- Neural networks
  - Introduction
  - Worked example (see Appendix)
  - Car trips example (see Appendix)
  - Deep learning
  - Summary

Classification Analysis with KNN, NN, and SVM

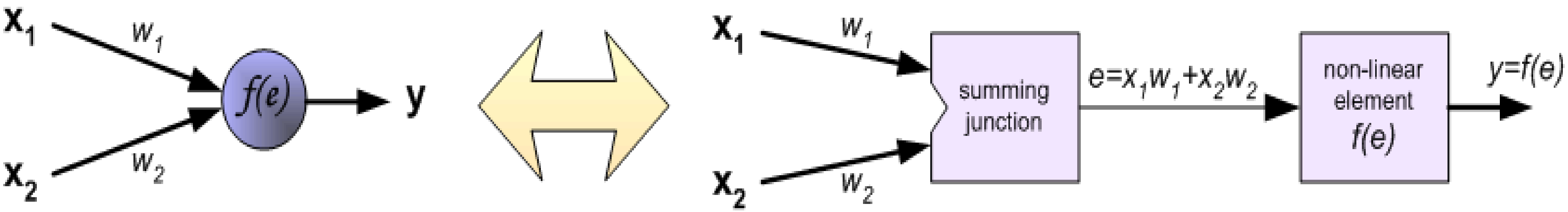
Neural network: introduction



The diagrams are from the Department of Electronics
AGH University of Science and Technology, Krakow, Poland

Classification Analysis with KNN, NN, and SVM

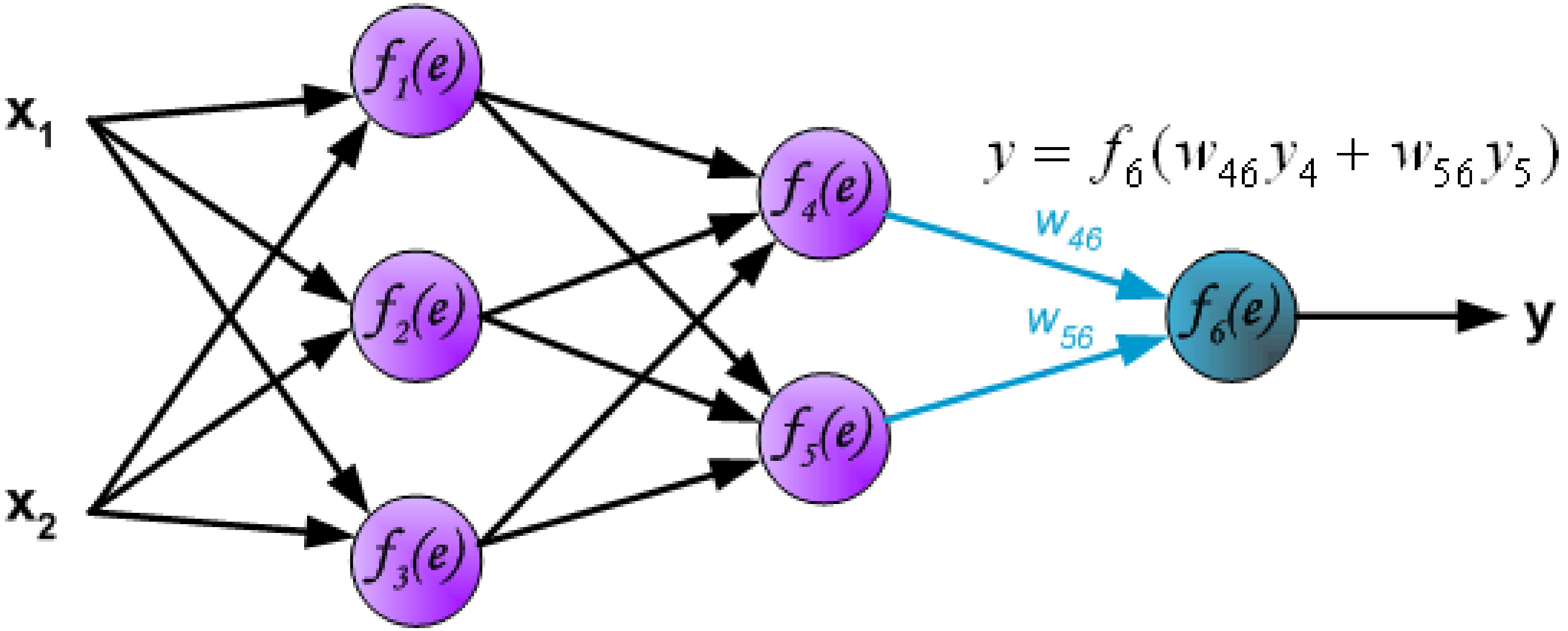
Neural network: introduction



The diagrams are from the Department of Electronics
AGH University of Science and Technology, Krakow, Poland

Classification Analysis with KNN, NN, and SVM

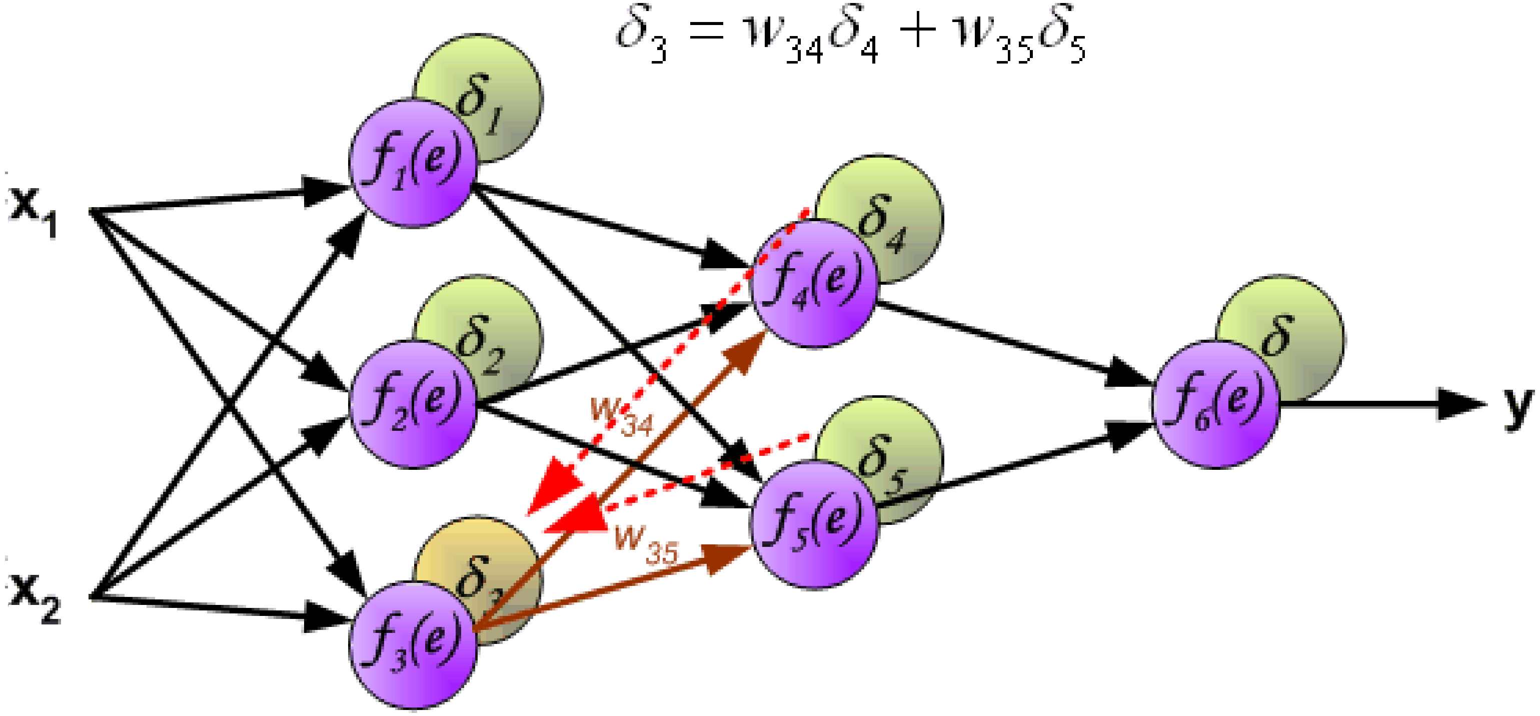
Neural network: introduction



The diagrams are from the Department of Electronics
AGH University of Science and Technology, Krakow, Poland

Classification Analysis with KNN, NN, and SVM

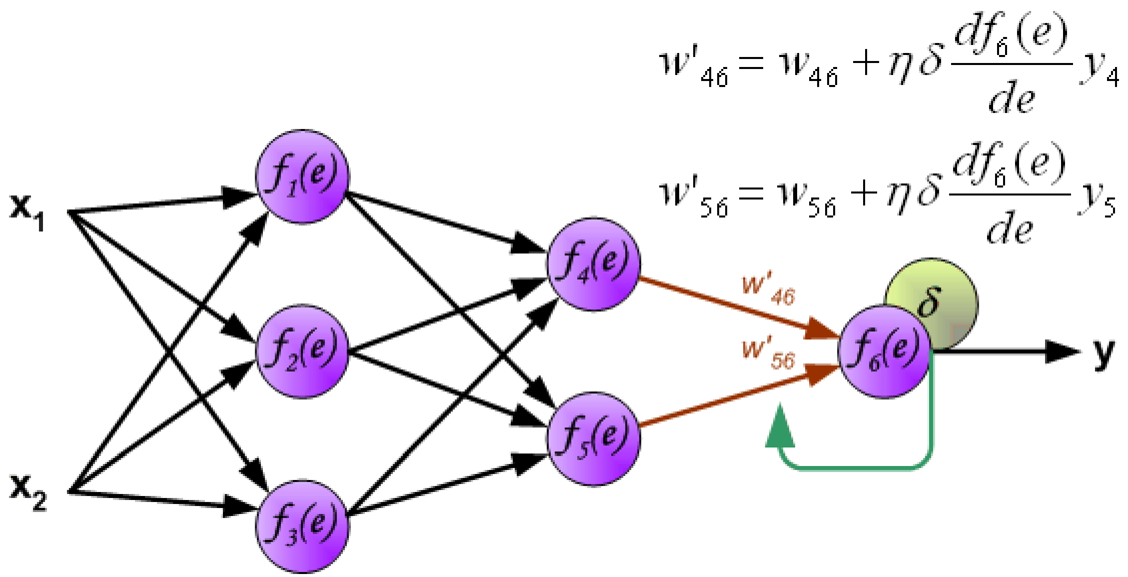
Neural network: introduction



The diagrams are from the Department of Electronics
AGH University of Science and Technology, Krakow, Poland

Classification Analysis with KNN, NN, and SVM

Neural network: introduction



The diagrams are from the Department of Electronics
AGH University of Science and Technology, Krakow, Poland

Classification Analysis with KNN, NN, and SVM

Summary

- **Strengths**
  - They can handle a wide range of problems
  - They can produce good results even in complicated domains
  - They can handle both categorical and continuous variables
- **Weaknesses**
  - Black box – hard to explain results
  - They may converge prematurely to an inferior solution
  - Potential to over-fit

Classification Analysis with KNN, NN, and SVM

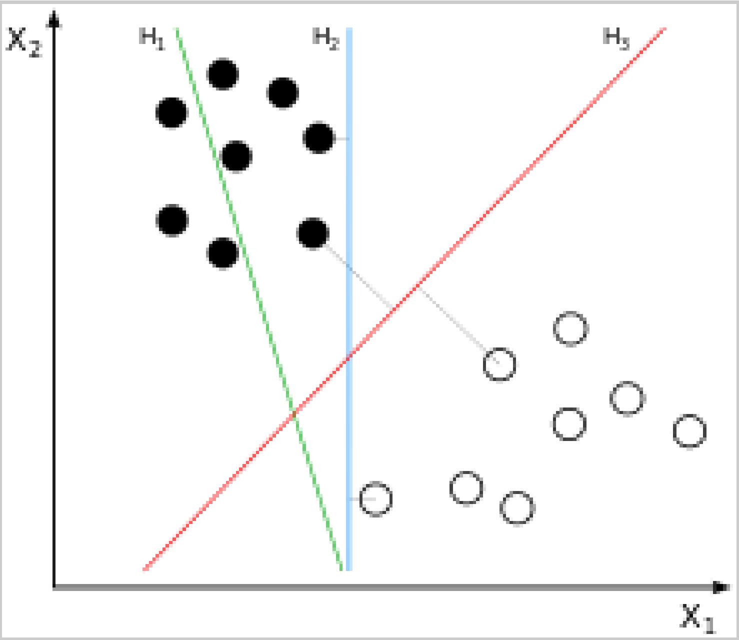
Contents

Contents

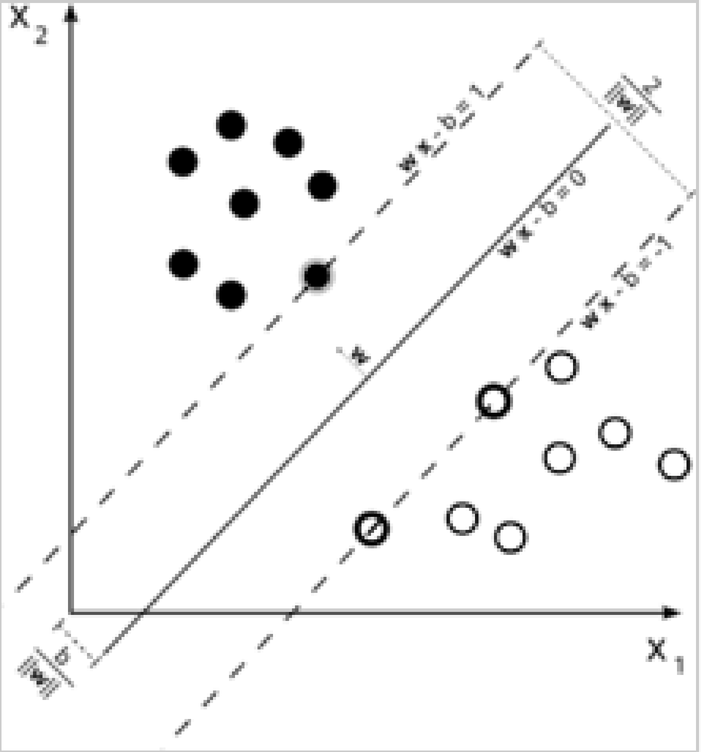
- Support vector machines (SVMs)
  - Introduction
  - SVMs in SAP PAL
  - Summary

Classification Analysis with KNN, NN, and SVM

Introduction



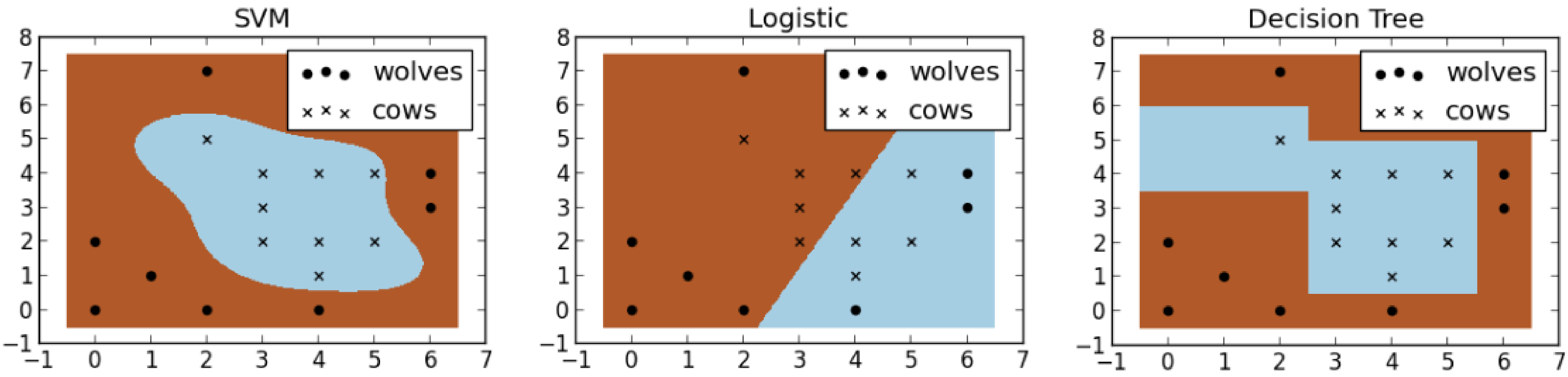
H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximum margin.



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Classification Analysis with KNN, NN, and SVM

SVMs vs other classification techniques



Greg Lamp
http://www.yaksis.com/posts/why-use-svm.html

Classification Analysis with KNN, NN, and SVM

SVM summary

- **Strengths**
  - SVMs are capable of both classification and regression.
  - Non-linear SVMs use a non-linear kernel. Non-linear SVM means that the boundary that the algorithm calculates doesn't have to be a straight line. The benefit is that you can capture much more complex relationships between your data points without having to perform difficult transformations. The downside is that the training time is much longer, because it is much more computationally intensive.
  - SVMs have a regularization parameter, which can help avoid over-fitting.
  - The “kernel trick” enables you to build in expert knowledge about the problem by engineering the kernel.
- **Weaknesses**
  - Requires full labeling of input data
  - For classification, the SVM is only directly applicable for two-class tasks. Therefore, algorithms that reduce the multi-class task to several binary problems have to be applied.
  - Parameters of a solved model are difficult to interpret.

Thank you

Contact information:

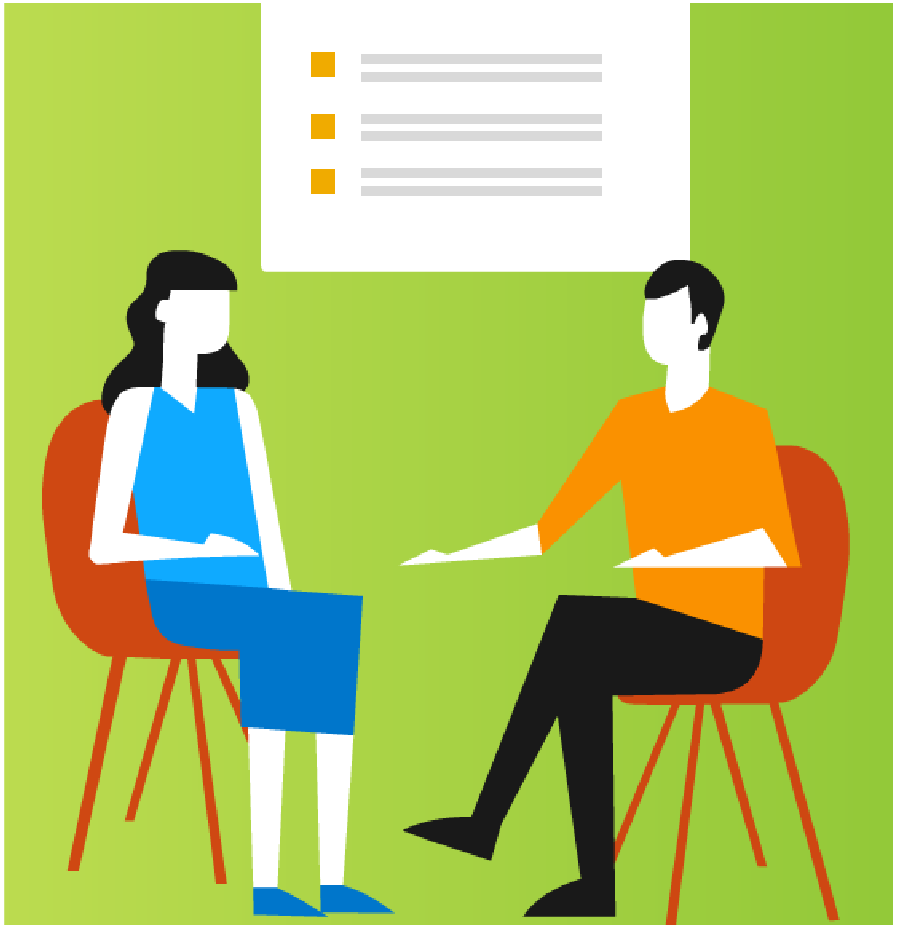
open@sap.com

Classification Analysis with KNN, NN, and SVM

Appendix

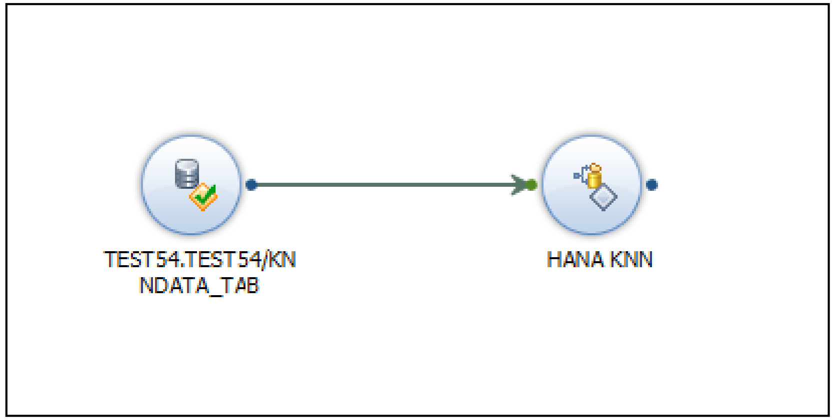
Additional material

- **KNN**
  - Worked example
- **Neural network**
  - Worked example
  - Car trips example
  - Deep learning

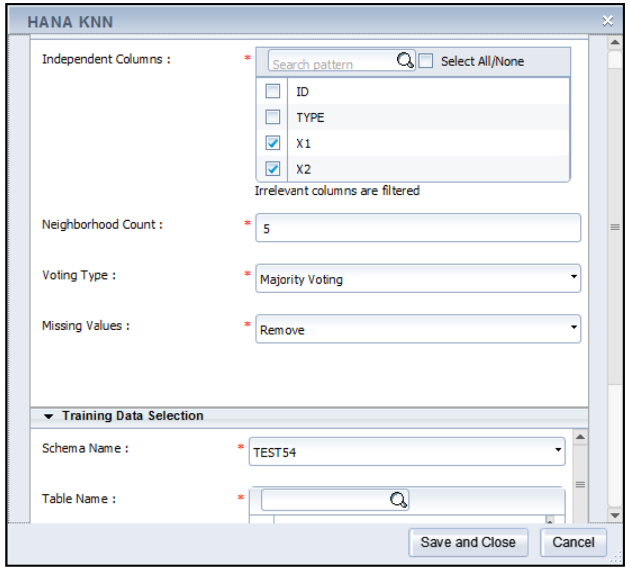


Classification Analysis with KNN, NN, and SVM

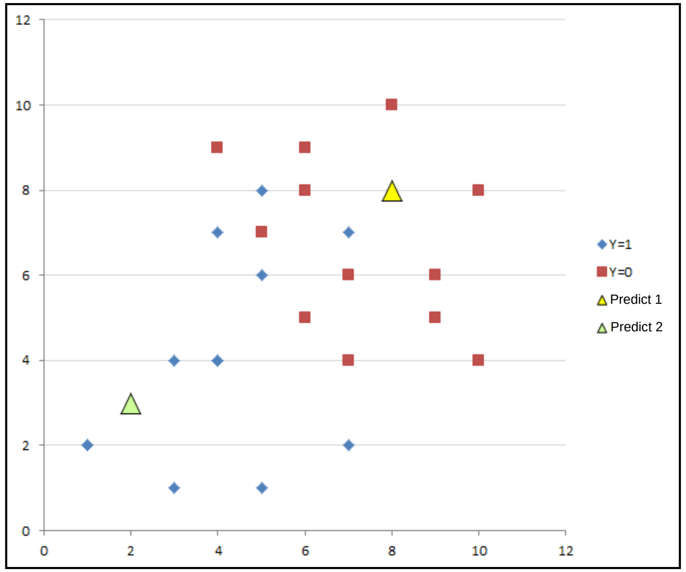
The k-nearest neighbor algorithm: worked example



Using SAP Predictive Analytics



Parameter Settings K=5



Visualization

	ID	X1	X2
1	1	8.0	8.0
2	2	2.0	3.0

Input Data

Prediction →

	ID	Type
1	1	0
2	2	1

Output Assignments

Classification Analysis with KNN, NN, and SVM

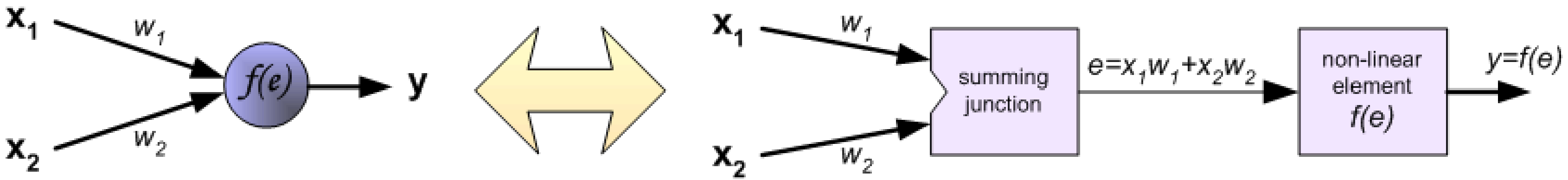
Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR				AND				NOR				NAND				XOR (Exclusive OR)		
Input 1	Input 2	Output		Input 1	Input 2	Output		Input 1	Input 2	Output		Input 1	Input 2	Output		Input 1	Input 2	Output
0	0	0		0	0	0		0	0	1		0	0	1		0	0	0
0	1	1		0	1	0		0	1	0		0	1	1		0	1	1
1	0	1		1	0	0		1	0	0		1	0	1		1	0	1
1	1	1		1	1	1		1	1	0		1	1	0		1	1	0

- We are looking for a transfer function that takes the inputs X_i and derives the output Y.
- It will comprise the sum of each input value, multiplied by a weight W_i compared to a threshold value.
- If it is greater than, it returns the threshold value to Y = 1, or if less than to Y = 0. (i.e. True or False).
- In neural terms, we say that the output Y is "firing" if it equals true (1) and that it is "resting" if it equals false (0).



Classification Analysis with KNN, NN, and SVM

Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

NOR

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR (Exclusive OR)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- For the OR truth table, we can use the following transfer function Y = (1.5 * X1 + 1.5 * X2 >= 1.0)
  The weights are W1 = 1.5, W2 = 1.5 and the threshold value is 1.0

OR Y := (1.5 * X1 + 1.5 * X2 >= 1.0)

Input 1	Input 2	W1	W2	H1	Y H1 >=1.0
0	0	1.5	1.5	0	0
0	1	1.5	1.5	1.5	1
1	0	1.5	1.5	1.5	1
1	1	1.5	1.5	3	1

Classification Analysis with KNN, NN, and SVM

Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

NOR

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR (Exclusive OR)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- For the AND truth table, we can use the following transfer function Y = (0.75 * X1 + 0.75 * X2 >= 1.0)
  The weights are W1 = 0.75, W2 = 0.75 and the threshold value is 1.0

AND	Y := (0.75 * X1 + 0.75 * X2 >= 1.0)				
					Y
Input 1	Input 2	W1	W2	H1	H1 >=1.0
0	0	0.75	0.75	0	0
0	1	0.75	0.75	0.75	0
1	0	0.75	0.75	0.75	0
1	1	0.75	0.75	1.5	1

Classification Analysis with KNN, NN, and SVM

Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

NOR

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR (Exclusive OR)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- For the NOR truth table, we can use the following transfer function Y = (1.5 * X1 + 1.5 * X2 =< 1.0) However, we want the threshold equation to be consistent as >=, which we can achieve as follows –

Y = (1.5 * X1 + 1.5 * X2 =< 1.0)

Y = (-1.5 * X1 + -1.5* X2 >= -1.0)

Y = (2.0*C + -1.5 * X1 + -1.5 * X2 >= +1.0)

NOR Y := (2.0 + -1.5 * X1 + -1.5 * X2 >= 1.0)

Input 1	Input 2	W1	W2	C	H1	Y H1 >=1.0
0	0	-1.5	-1.5	2	2	1
0	1	-1.5	-1.5	2	0.5	0
1	0	-1.5	-1.5	2	0.5	0
1	1	-1.5	-1.5	2	-1	0

Classification Analysis with KNN, NN, and SVM

Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

NOR

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR (Exclusive OR)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- For the NAND truth table, we can use the following transfer function Y = (0.75 * X1 + 0.75 * X2 =< 1.0) However, we want the threshold equation to be consistent as >=, which we can achieve as follows –

Y = (0.75 * X1 + 0.75 * X2 =< 1.0)

Y = (-0.75 * X1 + -0.75 * X2 >= -1.0)

Y = (2.0*C + -0.75 * X1 + -0.75 * X2 >= +1.0)

NAND Y := (2.0 + -0.75 * X1 + -0.75 * X2 >= 1.0)

Input 1	Input 2	W1	W2	C	H1	Y H1 >=1.0
0	0	-0.75	-0.75	2	2	1
0	1	-0.75	-0.75	2	1.25	1
1	0	-0.75	-0.75	2	1.25	1
1	1	-0.75	-0.75	2	0.5	0

Classification Analysis with KNN, NN, and SVM

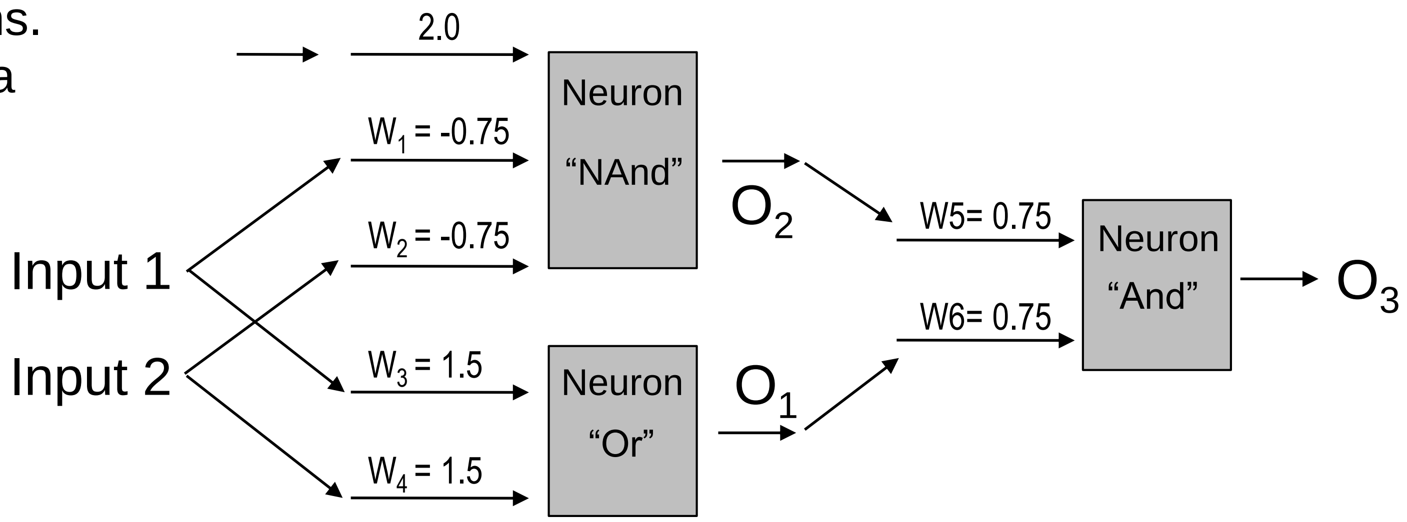
Neural network: worked example

- Worked examples – OR, AND, NOR, NAND, XOR

Truth Tables for OR, AND, NOR, NAND and XOR

OR			AND			NOR			NAND			XOR (Exclusive OR)		
Input 1	Input 2	Output	Input 1	Input 2	Output	Input 1	Input 2	Output	Input 1	Input 2	Output	Input 1	Input 2	Output
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
0	1	1	0	1	0	0	1	0	0	1	1	0	1	1
1	0	1	1	0	0	1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1	0	1	1	0

- For the XOR truth table, we have three neurons. A collection of neurons connected together is a "network" of neurons. Thus, the "XOr" function has been created using a "neural network".



Classification Analysis with KNN, NN, and SVM

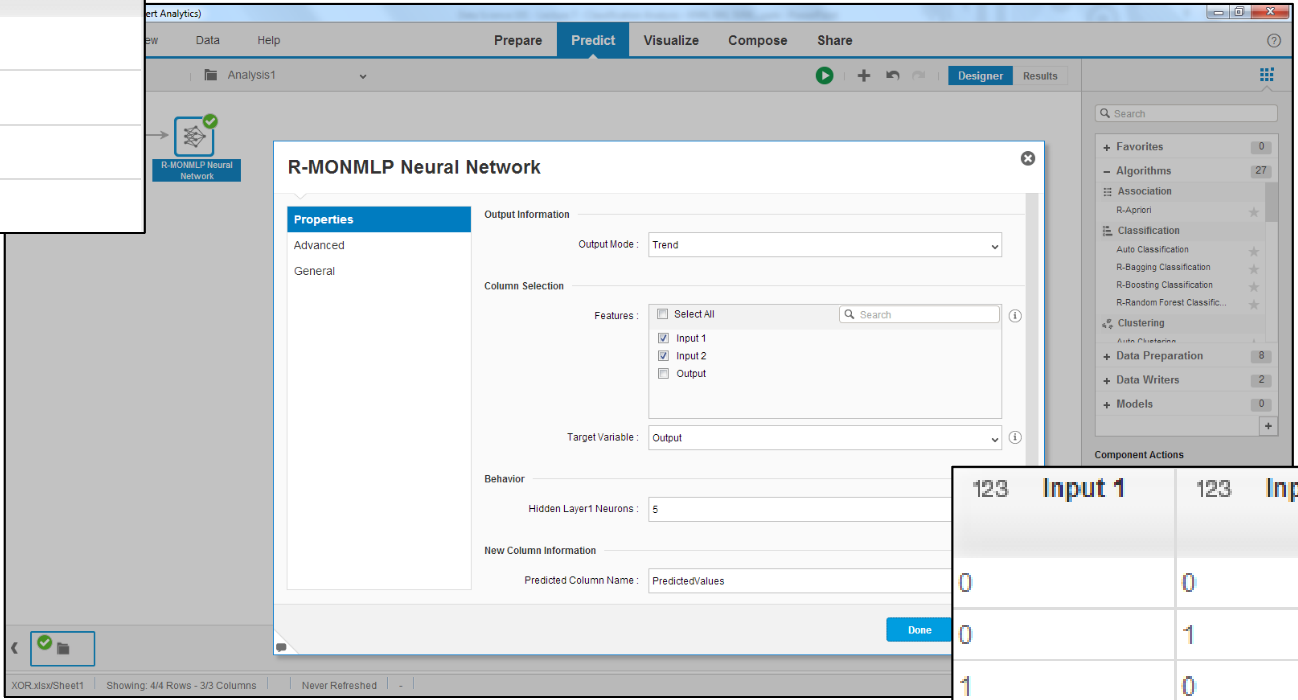
Neural network: worked example

- Training the network: A training algorithm is a step-by-step procedure for setting the weights to appropriate values to produce the desired function. By applying the training algorithm to the neural network, you are "training" the weights to your desired values.
- For example, suppose that we had a neural network with just one neuron in it that we want to train to duplicate the "And" function - Y = (0.75 * X1 + 0.75 * X2 >= 1.0)
- Suppose also that we do not initially know that the appropriate weights for this network should be W1 = 0.75 and W2 = 0.75 to produce the "And" function. We might initially set the weights W1 and W2 to 0.0 and evaluate the function. If the output is too low, increase the weights which had inputs that were "1". If the output is too high, decrease the weights which had inputs that were "1". Continue looping through this process until each possible input combination gives the right output.
- Alternatively we might set the weights to 0, 0.25, 0.5, 0.75, 1.0 and evaluate the function with these values, then iterate to the best fitting weights.

Classification Analysis with KNN, NN, and SVM
Neural network example in SAP Predictive Analytics

- The XOR truth table in SAP PA

123 Input 1	123 Input 2	123 Output
0	0	0
0	1	1
1	0	1
1	1	0



123 Input 1	123 Input 2	123 Output	123 Predicte..
0	0	0	-0.00
0	1	1	1.00
1	0	1	1.00
1	1	0	0.00

Classification Analysis with KNN, NN, and SVM

Neural network example in SAP Predictive Analytics

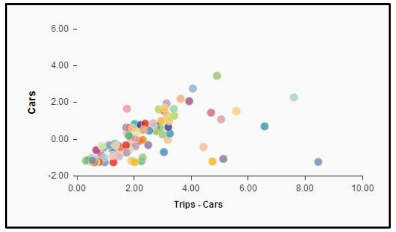
- We have data for 100 locations, giving the population size, number of houses, cars, average income, and employment level. For urban planning purposes, we wish to predict the number of car trips that will be made for new locations given these variables.
- The data has already been normalized to ensure each variable has equal weight in the analysis.

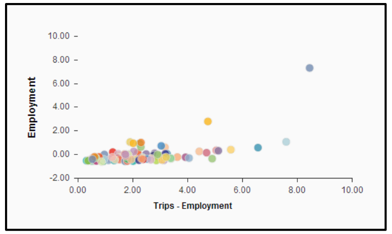
	A	B	C	D	E	F	G
1	Location	Population	Houses	Cars	Income	Employment	Trips
2	Location 1	-1.416	-1.386	-1.248	-0.828	7.328	8.460
3	Location 2	-1.335	-1.329	-1.207	-1.063	0.910	2.250
4	Location 3	-1.265	-1.181	-1.186	-1.373	-0.533	0.317
5	Location 4	-1.281	-1.056	-1.191	-1.251	1.043	1.907
6	Location 5	-1.427	-1.423	-1.223	-0.977	2.791	4.746
7	Location 6	-1.143	-1.079	-1.154	-1.241	-0.540	0.375
8	Location 7	-1.347	-1.371	-1.251	-1.521	-0.323	0.540
9	Location 8	-1.411	-1.402	-1.245	-1.067	-0.215	0.768
10	Location 9	-1.395	-1.426	-1.267	-0.796	-0.211	0.605
11	Location 10	-1.051	-0.967	-1.013	-0.144	0.616	2.298
12	Location 11	-0.371	-0.543	-0.717	-0.151	0.022	1.708
13	Location 12	-0.450	-0.616	-0.886	-1.191	-0.428	0.843
14	Location 13	1.536	1.142	0.434	-0.810	0.093	2.801
15	Location 14	0.223	0.068	-0.547	-0.024	-0.335	1.239
16	Location 15	1.365	1.155	0.445	-0.519	-0.105	2.756
17	Location 16	-0.280	0.016	-0.320	-0.797	-0.454	1.304
18	Location 17	0.837	0.844	0.623	0.123	-0.335	2.385
19	Location 18	-0.442	-0.455	-0.370	0.804	-0.014	1.883
20	Location 19	-0.758	-0.587	-0.458	0.347	-0.343	1.242
21	Location 20	1.059	3.034	0.979	-0.602	-0.206	2.946
22	Location 21	-0.070	0.018	-0.103	0.010	0.131	2.183
23	Location 22	-0.074	0.149	0.085	1.031	-0.304	1.868
24	Location 23	0.830	1.182	0.717	-0.208	-0.072	2.772
25	Location 24	-0.320	-0.286	-0.273	-0.242	-0.433	1.313
26	Location 25	0.259	0.341	-0.492	-0.825	-0.349	1.472
27	Location 26	1.001	0.329	-0.284	-0.813	-0.171	1.852
28	Location 27	0.183	0.185	-0.036	-0.259	0.595	3.177
29	Location 28	0.953	0.598	0.006	-0.626	-0.264	2.037
30	Location 29	-0.353	-0.311	-0.398	-0.557	0.243	2.037
31	Location 30	1.079	1.176	0.507	-0.252	-0.285	2.349
32	Location 31	0.778	0.811	0.632	0.035	-0.410	2.169
33	Location 32	-1.441	-1.440	-1.270	-1.446	0.190	1.272
34	Location 33	-0.273	-0.476	-0.941	-1.522	0.044	1.467
35	Location 34	-1.152	-1.158	-1.104	-1.202	-0.324	0.648
36	Location 35	0.310	0.283	-0.581	-1.152	-0.012	1.838
37	Location 36	0.531	0.988	-0.583	-1.374	-0.260	1.548
38	Location 37	-1.141	-1.171	-1.170	-0.635	-0.508	0.387
39	Location 38	-1.417	-1.419	-1.266	-1.031	-0.009	0.966
40	Location 39	0.576	1.113	0.257	-0.591	-0.112	2.996
41	Location 40	1.832	1.733	2.193	0.871	-0.231	3.631

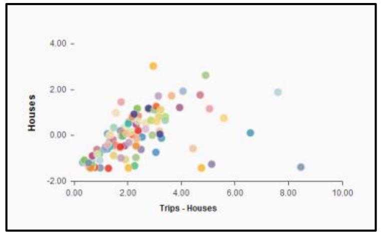
Classification Analysis with KNN, NN, and SVM

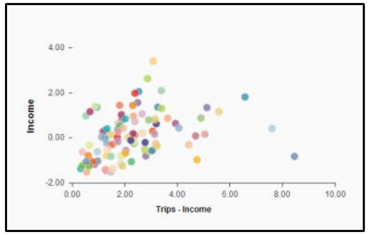
Neural network example in SAP Predictive Analytics

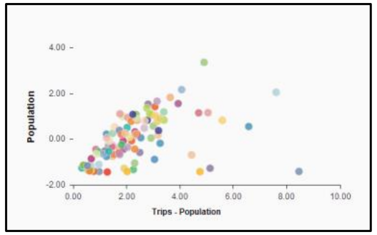
- We should start with an initial data exploration.
- We see a few strong relationships between the independent variables and a few outliers, but overall, no clear pattern.

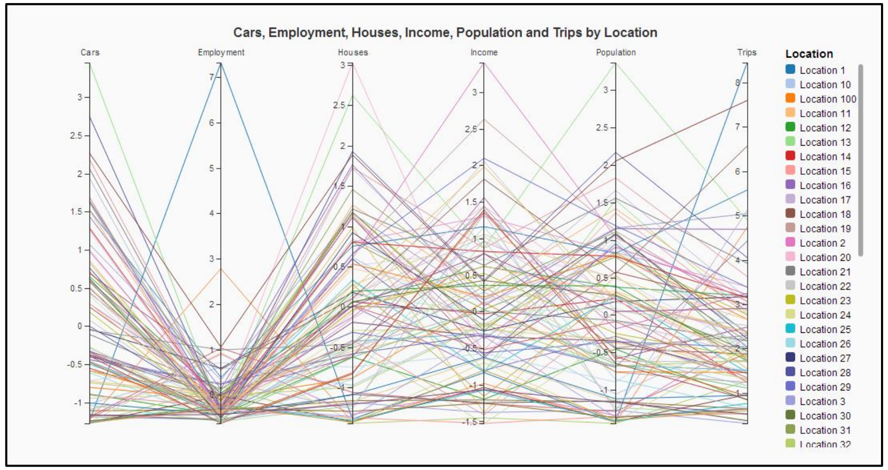








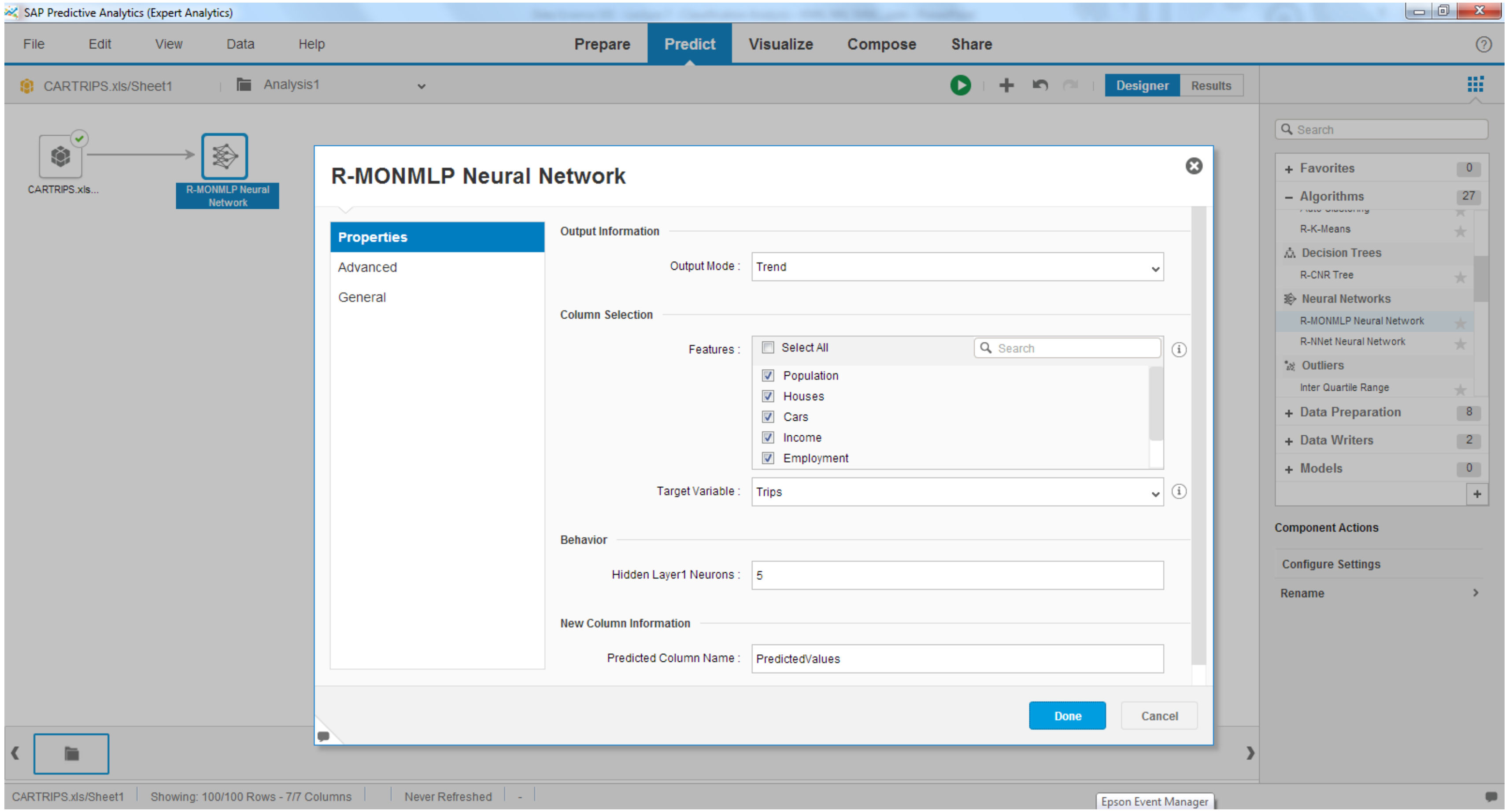




Classification Analysis with KNN, NN, and SVM

Neural network example in SAP Predictive Analytics

- Define the analysis



Classification Analysis with KNN, NN, and SVM

Neural network example in SAP Predictive Analytics

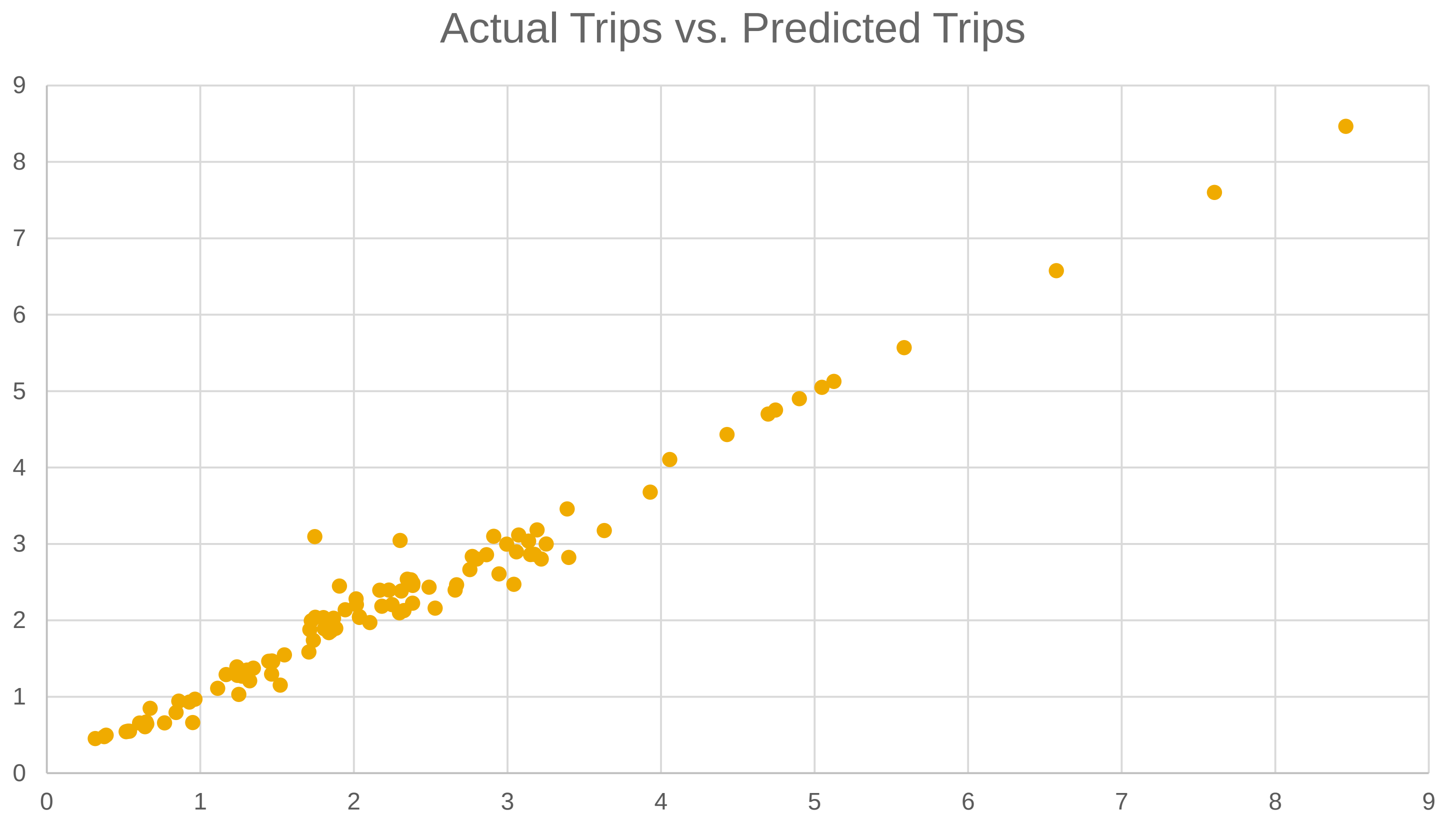
- Run the analysis

ABC Location	123 Population	123 Houses	123 Cars	123 Income	123 Employm..	123 Trips	123 Predicte..
Location 1	-1.42	-1.39	-1.25	-0.83	7.33	8.46	8.46
Location 2	-1.34	-1.33	-1.21	-1.06	0.91	2.25	2.20
Location 3	-1.27	-1.18	-1.19	-1.37	-0.53	0.32	0.45
Location 4	-1.28	-1.06	-1.19	-1.25	1.04	1.91	2.45
Location 5	-1.43	-1.42	-1.22	-0.98	2.79	4.75	4.75
Location 6	-1.14	-1.08	-1.15	-1.24	-0.54	0.38	0.48
Location 7	-1.35	-1.37	-1.25	-1.52	-0.32	0.54	0.55
Location 8	-1.41	-1.40	-1.24	-1.07	-0.22	0.77	0.66
Location 9	-1.40	-1.43	-1.27	-0.80	-0.21	0.60	0.65
Location 10	-1.05	-0.97	-1.01	-0.14	0.62	2.30	2.10
Location 11	-0.37	-0.54	-0.72	-0.15	0.02	1.71	1.58
Location 12	-0.45	-0.62	-0.89	-1.19	-0.43	0.84	0.79
Location 13	1.54	1.14	0.43	-0.81	0.09	2.80	2.80
Location 14	0.22	0.07	-0.55	-0.02	-0.33	1.24	1.39
Location 15	1.36	1.16	0.44	-0.52	-0.11	2.76	2.66
Location 16	-0.28	0.02	-0.32	-0.80	-0.45	1.30	1.35
Location 17	0.84	0.84	0.62	0.12	-0.33	2.38	2.49
Location 18	-0.44	-0.46	-0.37	0.80	-0.01	1.88	1.89
Location 19	-0.76	-0.59	-0.46	0.35	-0.34	1.24	1.28
Location 20	1.06	3.03	0.98	-0.60	-0.21	2.95	2.61
Location 21	-0.07	0.02	-0.10	0.01	0.13	2.18	2.18

Classification Analysis with KNN, NN, and SVM

Neural network example in SAP Predictive Analytics

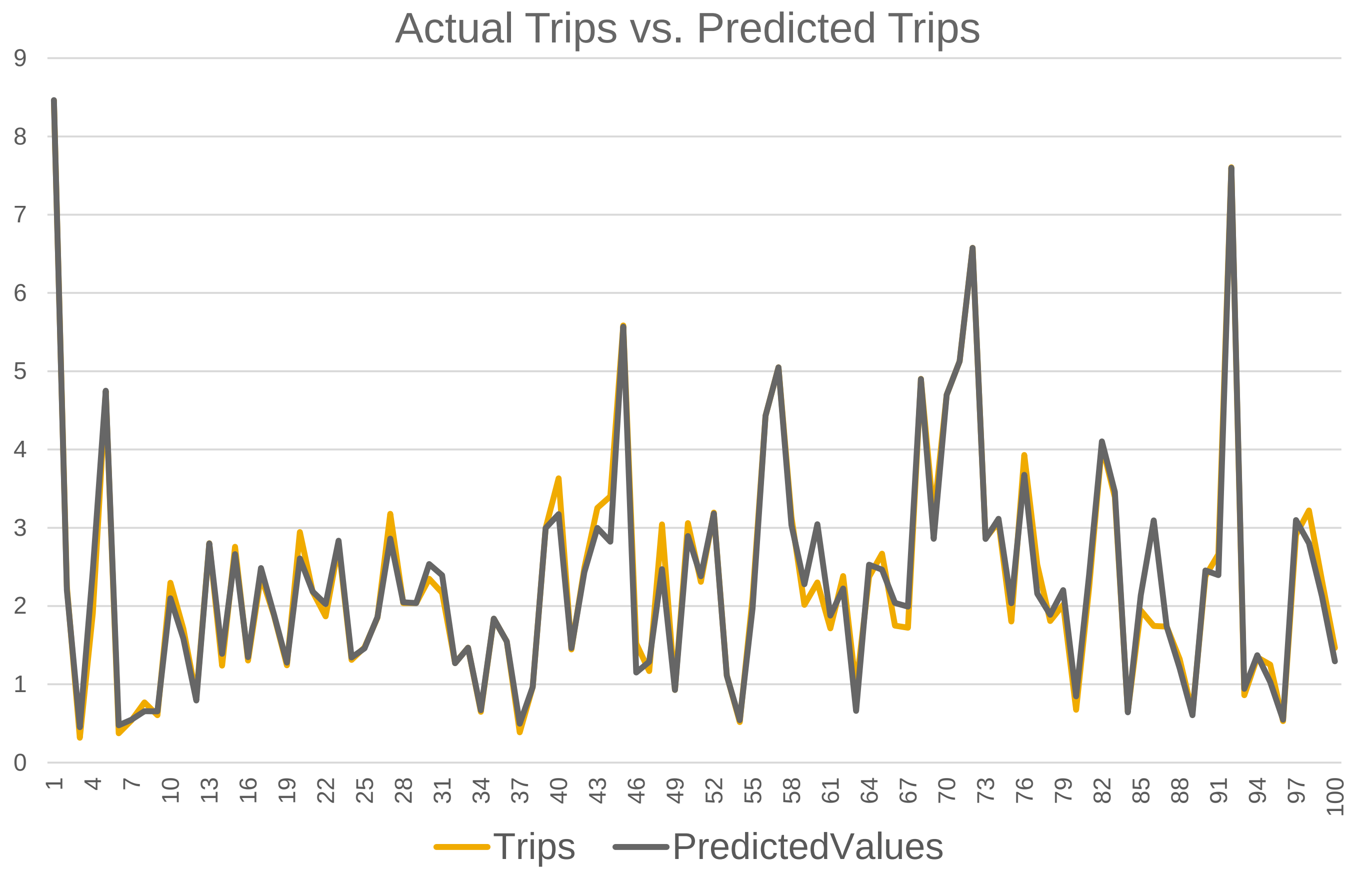
- View the results after exporting to the Visualization tab.



Classification Analysis with KNN, NN, and SVM

Neural network example in SAP Predictive Analytics

- View the results after exporting to the Visualization tab.



Classification Analysis with KNN, NN, and SVM

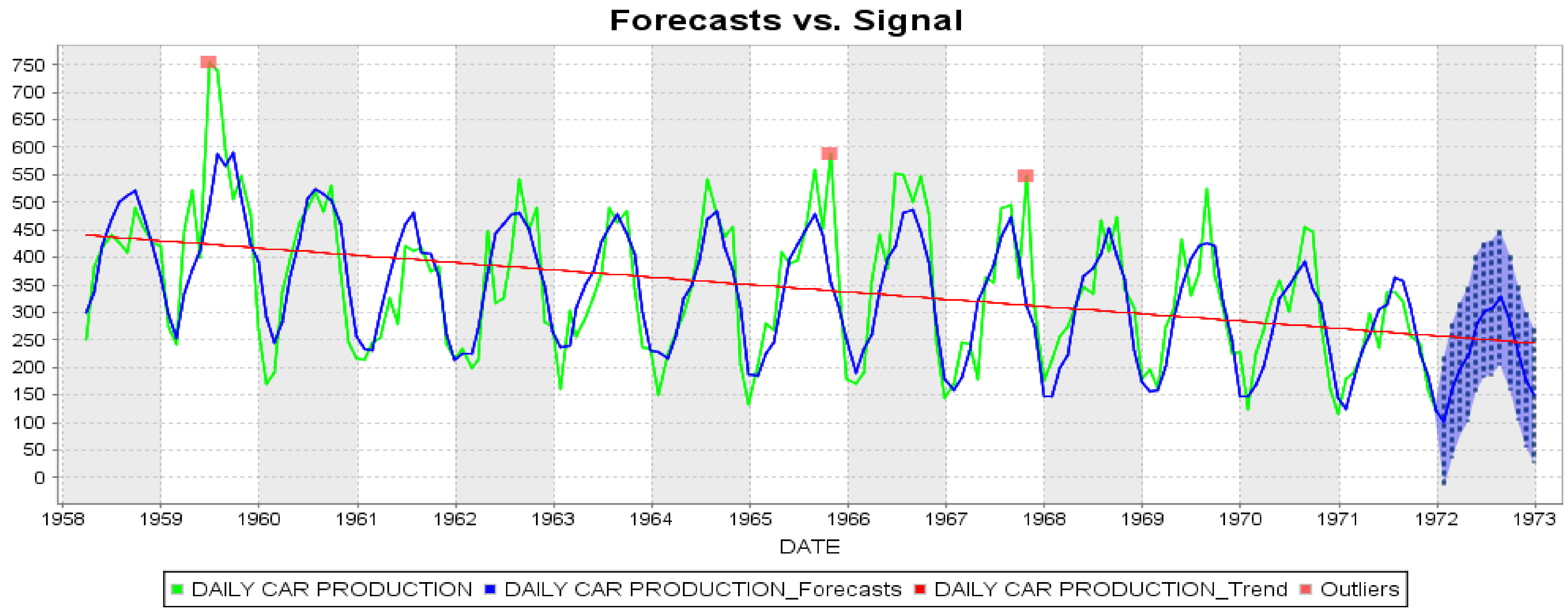
Deep learning

- *Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks, and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition, and bioinformatics, where they have been shown to produce state-of-the-art results on various tasks.* Wikipedia
- Geoffrey Hinton is known as the "Father of Deep Learning". He was a professor at Toronto University, and after his research, got hired in the Google Brain team. He is quoted as saying "Deep learning is an algorithm which has no theoretical limitations of what it can learn; the more data you give and the more computational time you provide, the better it is". See http://www.cs.toronto.edu/~hinton/
  - http://www.kdnuggets.com/2014/12/geoff-hinton-ama-neural-networks-brain-machine-learning.html
  - http://www.kdnuggets.com/2014/12/geoffrey-hinton-talks-deep-learning-google-everything.html
  - For further information http://deeplearning.net/
- Deep learning has been characterized as a buzzword, or a rebranding of neural networks.
- A word of caution in an article by Zachary Chase Lipton, who regularly contributes to KDnuggets – "Recent press has challenged the hype surrounding deep learning, trumpeting several findings which expose shortcomings of current algorithms. However, many of deep learning's reported flaws are universal, affecting nearly all machine learning algorithms."
- This report summarizes potential applications of AI -
  - http://www.giiresearch.com/report/trac328668-artificial-intelligence-enterprise-applications.html
- Early applications in SAP are being considered for medical images, medical text analysis, and sentiment analysis.

Week 4 Unit 3: **Time Series Analysis**

Time Series Analysis

Introduction

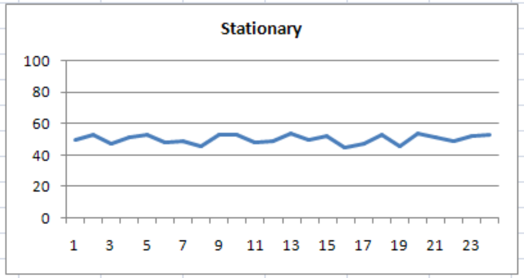


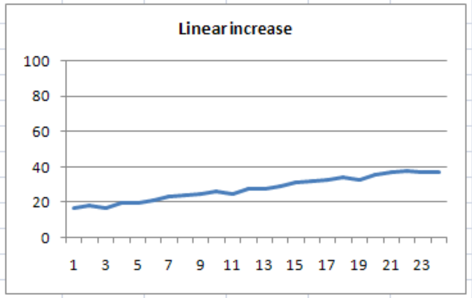
Time Series Analysis

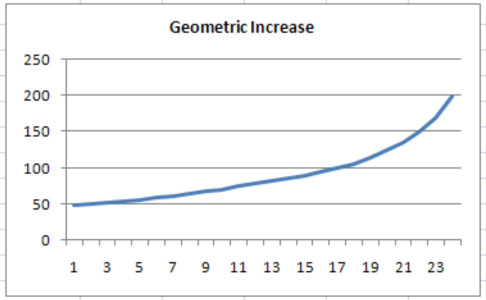
Stationary, trend, & seasonality

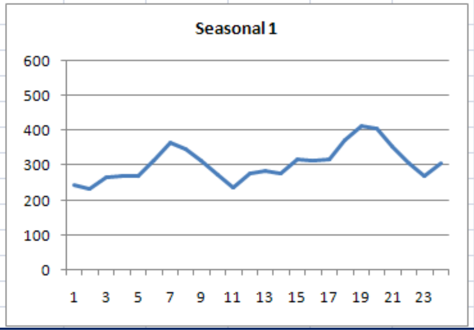
Basic series

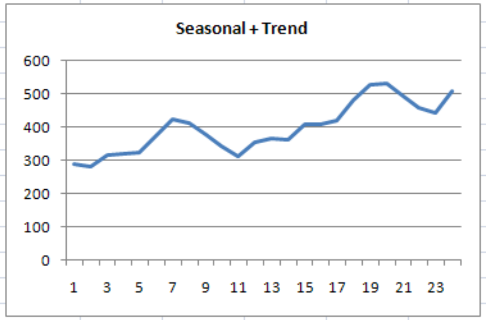
Period	Stationary	Linear inc	Geometri	Seasonal	Seasonal + Trend
1	50	17	48	242	290
2	53	18	49	233	282
3	47	17	51	267	318
4	51	20	53	269	322
5	53	20	55	270	325
6	48	21	58	315	373
7	49	23	61	364	425
8	46	24	64	347	411
9	53	25	67	312	379
10	53	26	70	274	344
11	48	25	74	237	311
12	49	28	78	278	356
13	54	28	82	284	366
14	50	29	86	277	363
15	52	31	90	317	407
16	45	32	95	313	408
17	47	33	100	318	418
18	53	34	105	374	479
19	46	33	115	413	528
20	54	36	125	405	530
21	51	37	135	355	490
22	49	38	150	306	456
23	52	37	170	271	441
24	53	37	200	306	506



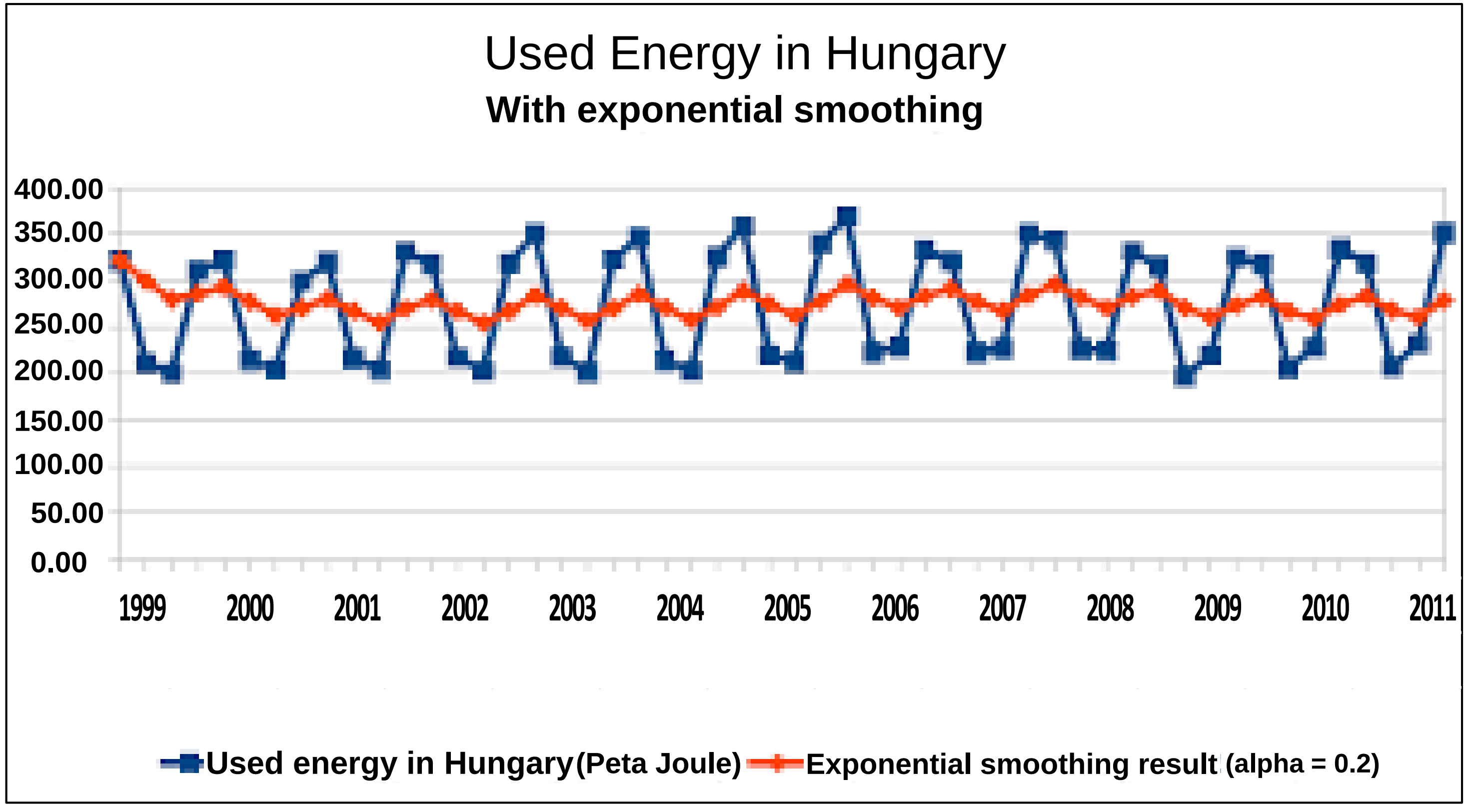








Time Series Analysis
Exponential smoothing



Time Series Analysis

Accuracy measures

- MAPE is one of the most popular measures for forecasting time series error.
- MAPE is the **Mean Absolute Percentage Error**.
- This is an accuracy measure of the forecast:

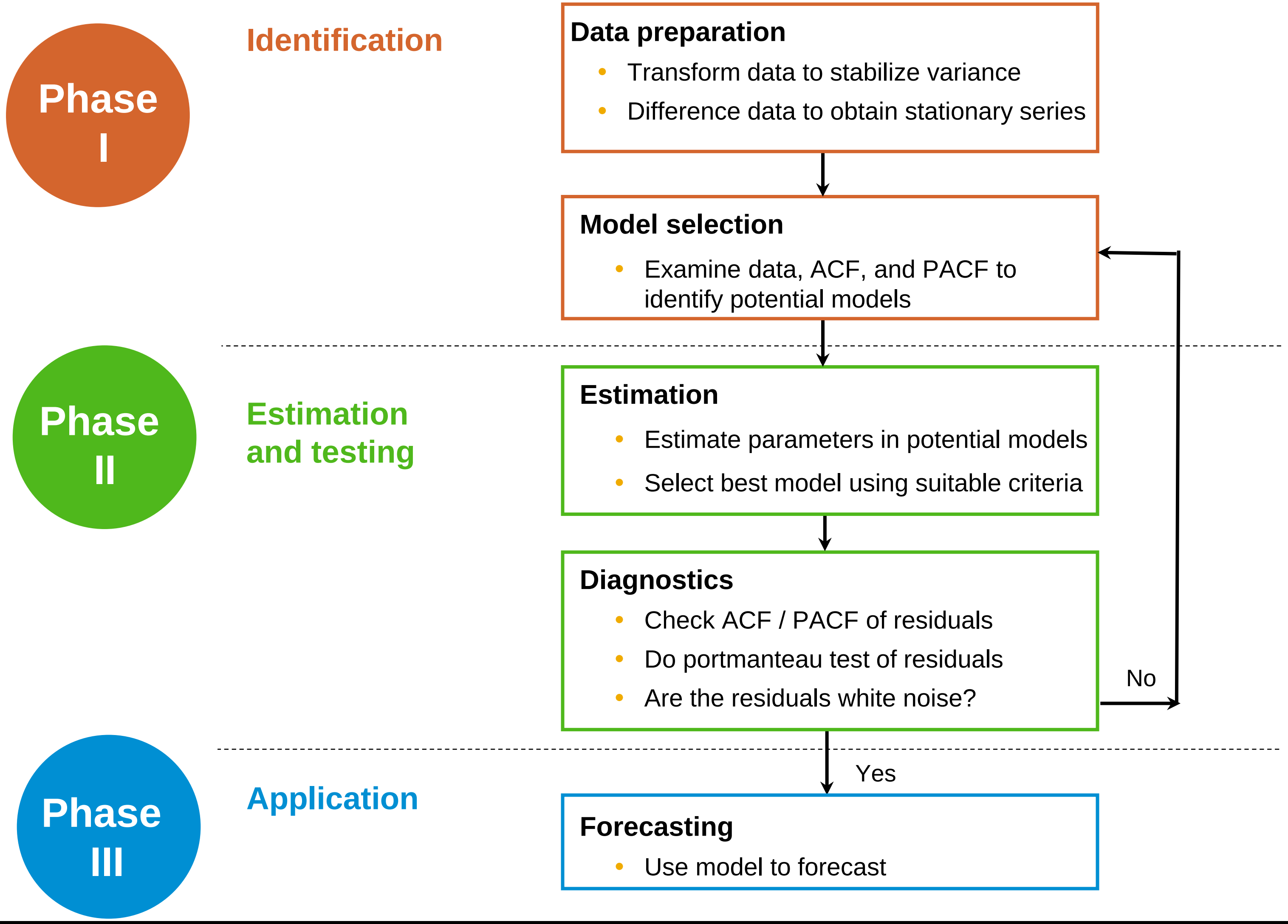
$$\text{MAPE}= \frac{1}{n} \times \sum_{t=1}^{n} \left|\frac{A_t - Ft}{A_t}\right|$$

where A_t is the actual value and F_t is the forecast value.

Description
Specifies measure name: • MPE: Mean percentage error • MSE: Mean squared error • RMSE: Root mean squared error • ET: Error total • MAD: Mean absolute deviation • MASE: Out-of-sample mean absolute scaled error • WMAPE: Weighted mean absolute percentage error • SMAPE: Symmetric mean absolute percentage error • MAPE: Mean absolute percentage error

Time Series Analysis

Autoregressive integrated moving average (ARIMA)



Thank you

Contact information:

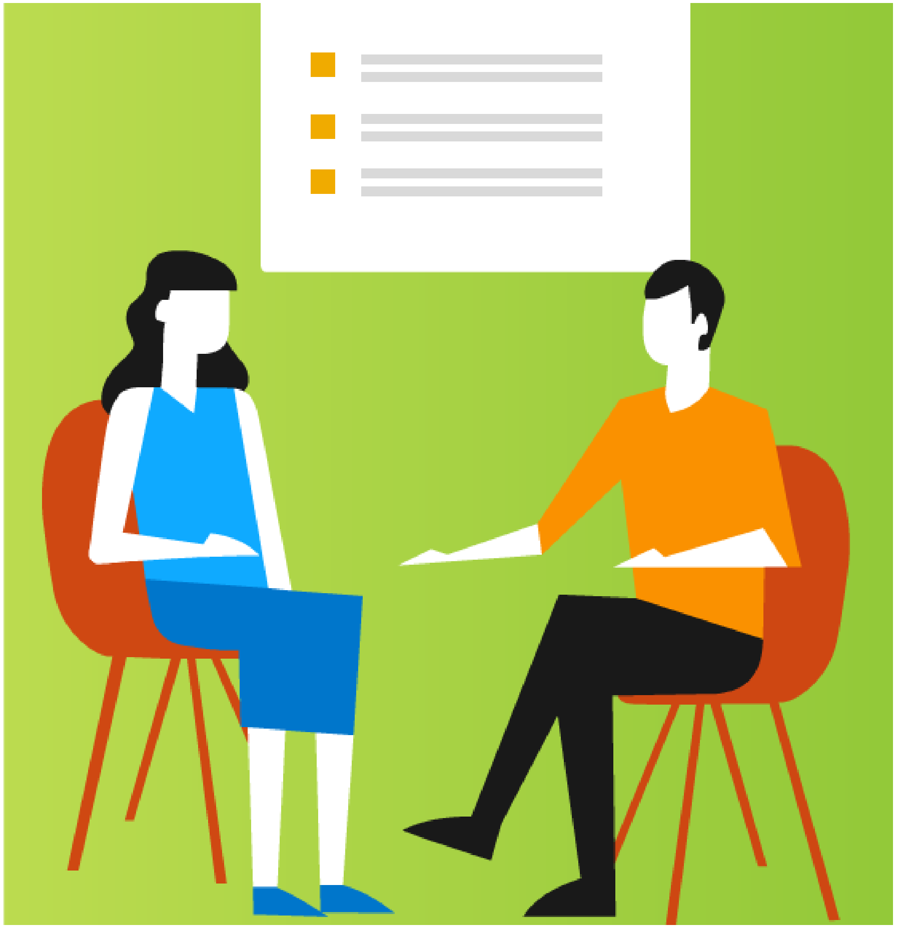
open@sap.com

Time Series Analysis
Appendix

Additional Material

Time series

- Single exponential smoothing: worked example
- Double exponential smoothing: worked example
- Triple exponential smoothing: worked example
- ARIMA details

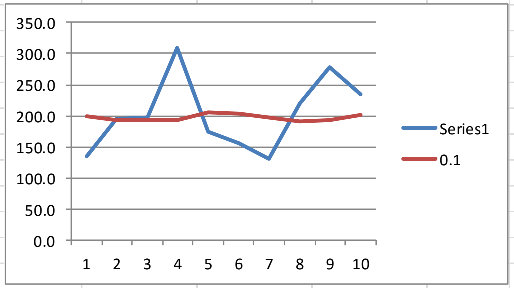


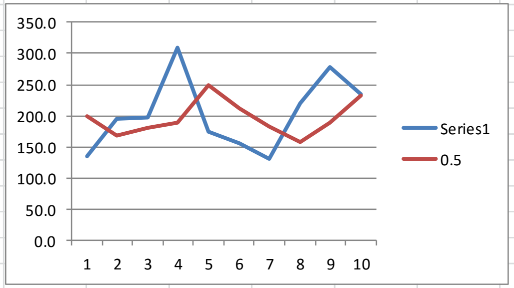
Time Series Analysis

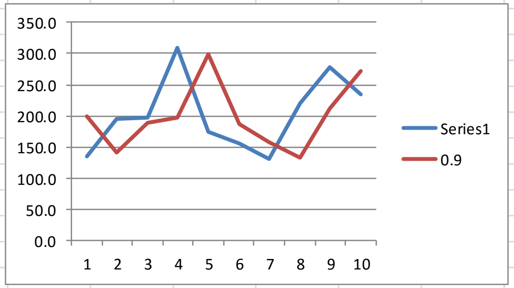
Exponential smoothing: worked example

Single or simple exponential smoothing – a weighted average of the past

Period	Actual	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
1	200.0															
2	135.0	200.0	200.0	200.0	-65.0	-65.0	-65.0	65.0	65.0	65.0	48.15	48.15	48.15	4225	4225	4225
3	195.0	193.5	167.5	141.5	1.5	27.5	53.5	1.5	27.5	53.5	0.77	14.10	27.44	2.25	756.25	2862.25
4	197.5	193.7	181.3	189.7	3.8	16.3	7.8	3.8	16.3	7.8	1.95	8.23	3.97	14.8225	264.0625	61.6225
5	310.0	194.0	189.4	196.7	116.0	120.6	113.3	116.0	120.6	113.3	37.41	38.91	36.54	13447.88	14550.39	12833.49
6	175.0	205.6	249.7	298.7	-30.6	-74.7	-123.7	30.6	74.7	123.7	17.50	42.68	70.67	938.2888	5578.223	15294.64
7	155.0	202.6	212.3	187.4	-47.6	-57.3	-32.4	47.6	57.3	32.4	30.69	37.00	20.88	2262.748	3288.306	1047.632
8	130.0	197.8	183.7	158.2	-67.8	-53.7	-28.2	67.8	53.7	28.2	52.16	41.29	21.72	4598.402	2880.67	797.3121
9	220.0	191.0	156.8	132.8	29.0	63.2	87.2	29.0	63.2	87.2	13.17	28.71	39.63	839.2398	3989.699	7599.712
10	277.5	193.9	188.4	211.3	83.6	89.1	66.2	83.6	89.1	66.2	30.12	32.10	23.86	6984.392	7935.608	4384.775
11	235.0	202.3	233.0	270.9	32.7	2.0	-35.9	32.7	2.0	35.9	13.92	0.87	15.27	1070.298	4.165745	1287.248
12		205.6	234.0	238.6												
Analysis of Errors																
Test Period 2 to 11																
Mean Error					5.56	6.80	4.29									
Mean Absolute Error					47.76	56.94	61.32									
Mean Absolute Percent Error					24.58	29.20	30.81									
Mean Squared Error					3438.332	4347.237	5039.368									





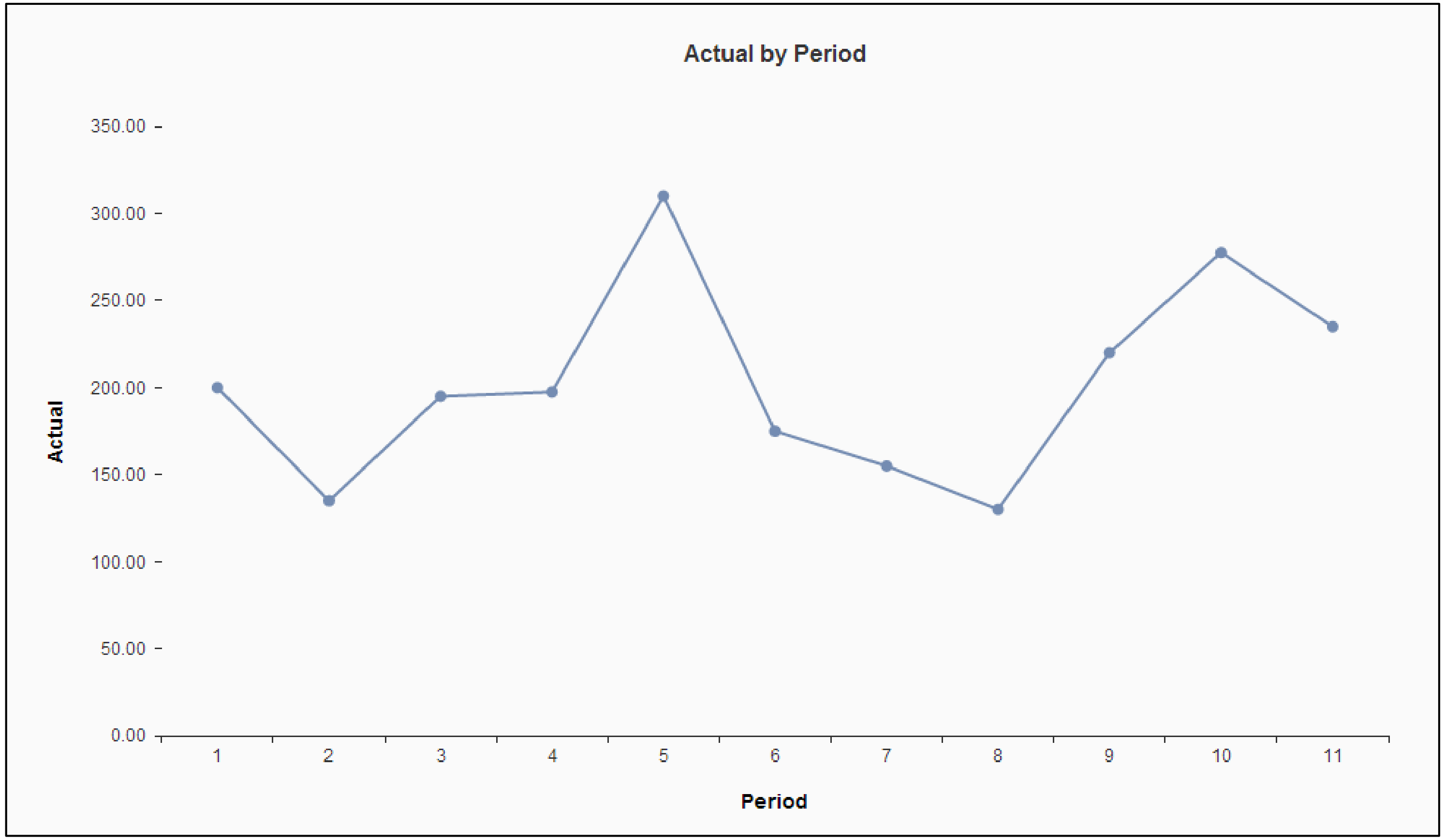


Time Series Analysis

Exponential smoothing in SAP Predictive Analytics

Single Exponential Smoothing

123 Period	123 Actual
1	200.00
2	135.00
3	195.00
4	197.50
5	310.00
6	175.00
7	155.00
8	130.00
9	220.00
10	277.50
11	235.00

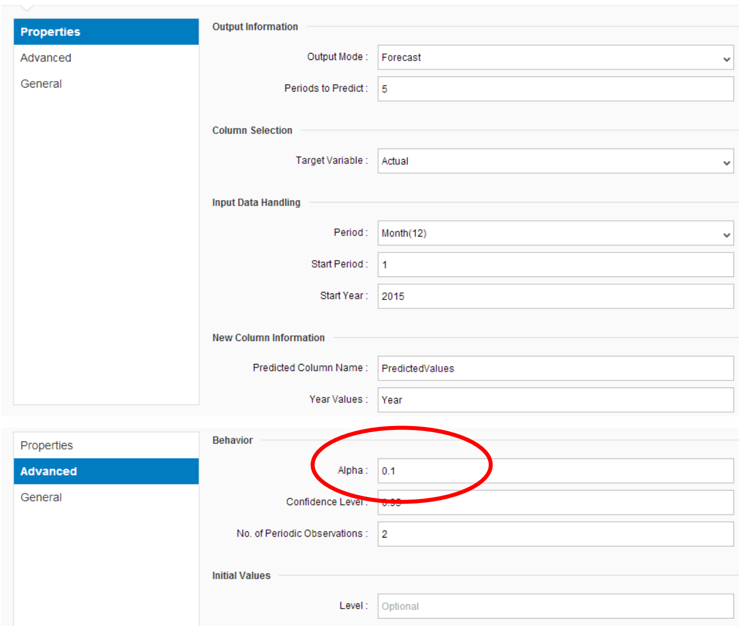


Time Series Analysis

Exponential smoothing in SAP Predictive Analytics

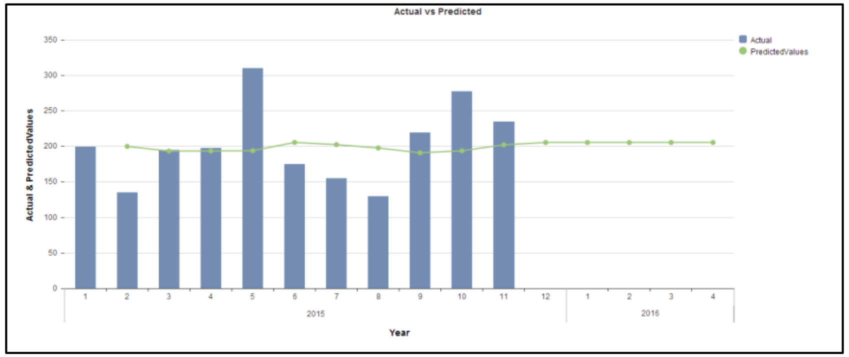
Single Exponential Smoothing

Alpha = 0.1



Parameter Settings

123 Year	123 Month	123 Period	123 Actual	123 Predicte..
2015	1	1	200.00	
2015	2	2	135.00	200.00
2015	3	3	195.00	193.50
2015	4	4	197.50	193.65
2015	5	5	310.00	194.04
2015	6	6	175.00	205.63
2015	7	7	155.00	202.57
2015	8	8	130.00	197.81
2015	9	9	220.00	191.03
2015	10	10	277.50	193.93
2015	11	11	235.00	202.28
2015	12			205.56
2016	1			205.56
2016	2			205.56
2016	3			205.56
2016	4			205.56



Output

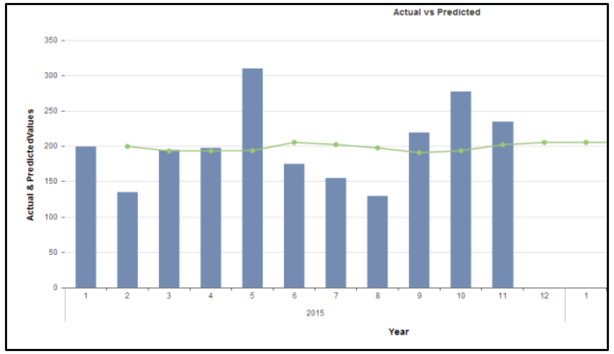
Time Series Analysis

Exponential smoothing in SAP Predictive Analytics

Single Exponential Smoothing

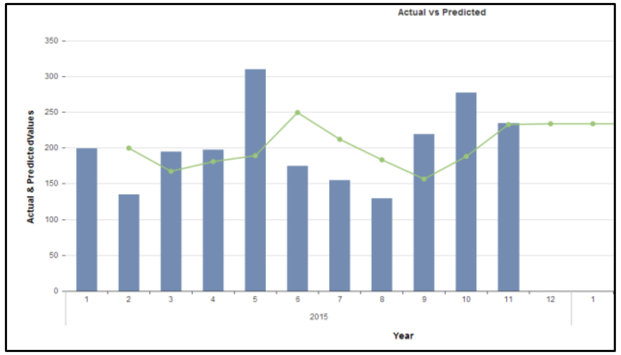
Alpha = 0.1

123 Period	123 Actual	123 Predicte..
1	200.00	
2	135.00	200.00
3	195.00	193.50
4	197.50	193.65
5	310.00	194.04
6	175.00	205.63
7	155.00	202.57
8	130.00	197.81
9	220.00	191.03
10	277.50	193.93
11	235.00	202.28
		205.56
		205.56
		205.56
		205.56
		205.56



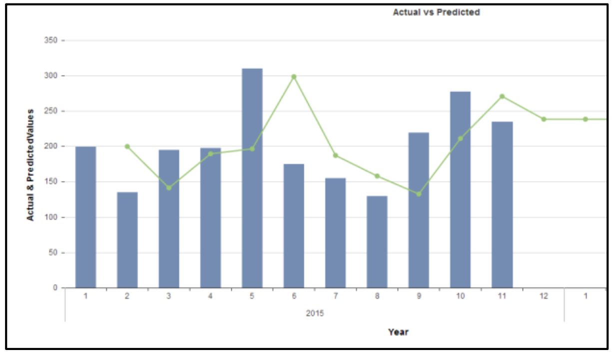
Alpha = 0.5

123 Period	123 Actual	123 Predicte..
1	200.00	
2	135.00	200.00
3	195.00	167.50
4	197.50	181.25
5	310.00	189.38
6	175.00	249.69
7	155.00	212.34
8	130.00	183.67
9	220.00	156.84
10	277.50	188.42
11	235.00	232.96
		233.98
		233.98
		233.98
		233.98
		233.98



Alpha = 0.9

123 Period	123 Actual	123 Predicte..
1	200.00	
2	135.00	200.00
3	195.00	141.50
4	197.50	189.65
5	310.00	196.72
6	175.00	298.67
7	155.00	187.37
8	130.00	158.24
9	220.00	132.82
10	277.50	211.28
11	235.00	270.88
		238.59
		238.59
		238.59
		238.59
		238.59



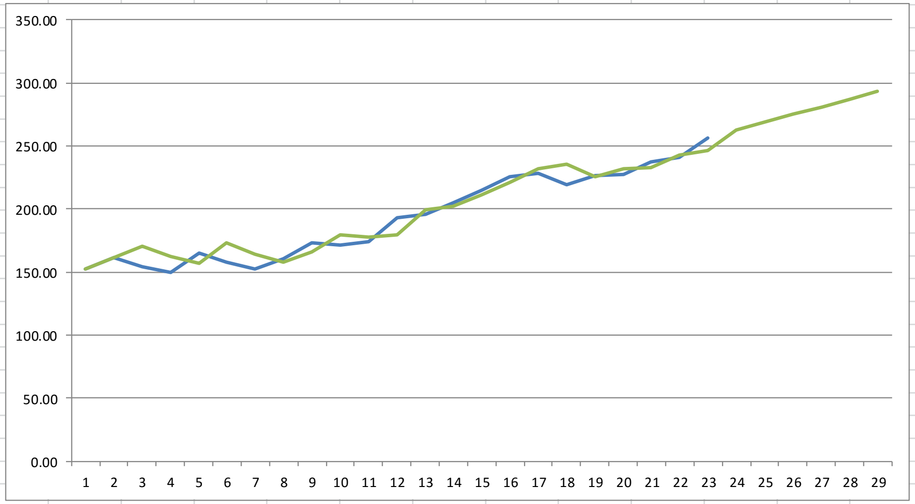
Time Series Analysis

Double exponential smoothing: worked example

Double exponential smoothing that applies two smoothing constants: one for the stationary element and the other for the trend

Double Exponential Smoothing

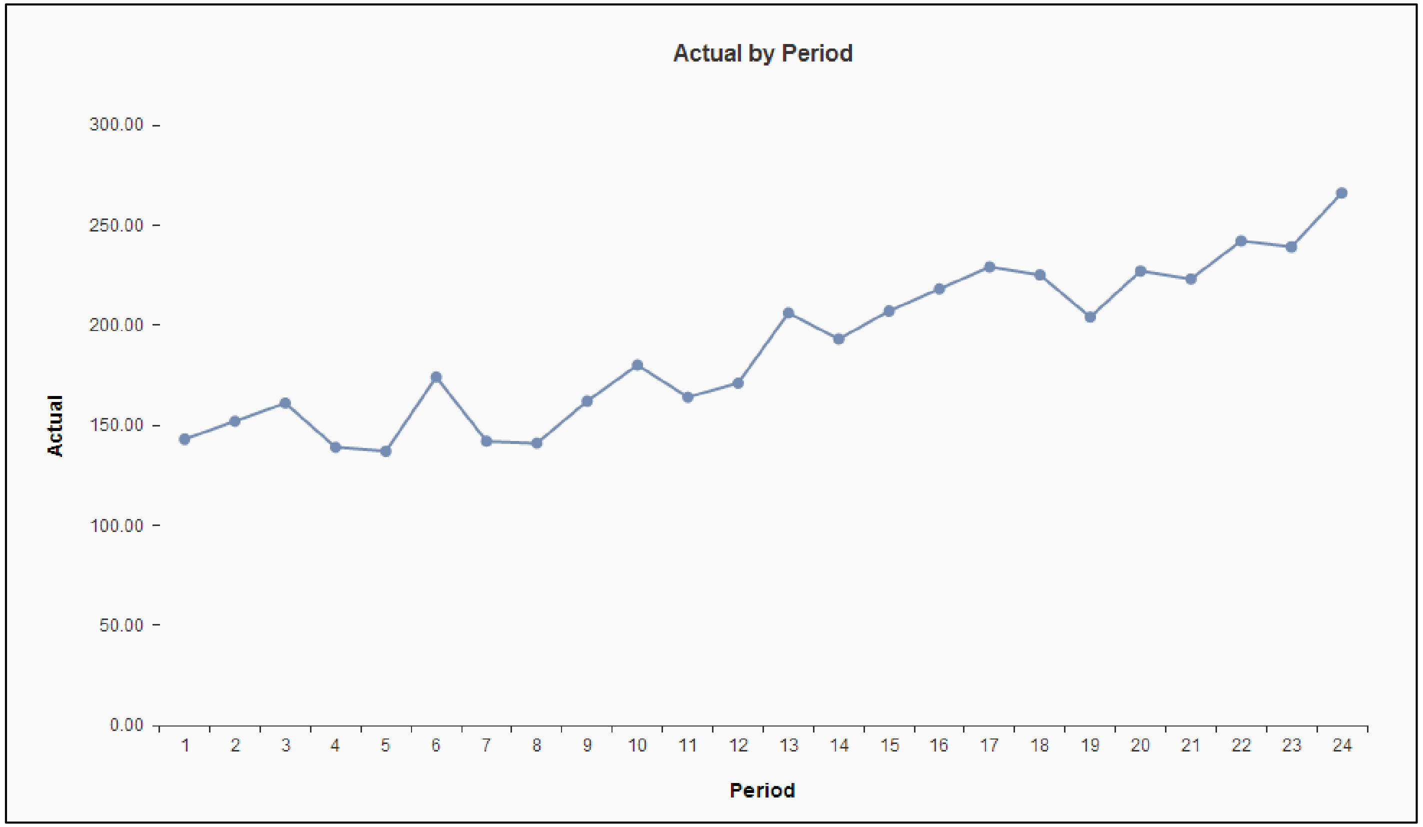
Period	Actual	lt	bt	Ft	
1	143.0	143.00	9.00		
2	152.0	152.00	9.00	152.00	
3	161.0	161.00	9.00	161.00	
4	139.0	154.47	7.88	170.00	
5	137.0	149.65	6.97	162.35	
6	174.0	165.33	7.59	156.62	
7	142.0	157.43	6.48	172.92	
8	141.0	152.43	5.65	163.91	
9	162.0	160.05	5.79	158.08	
10	180.0	172.93	6.30	165.84	
11	164.0	171.60	5.75	179.24	
12	171.0	174.17	5.53	177.36	
13	206.0	192.88	6.47	179.70	
14	193.0	196.17	6.25	199.35	
15	207.0	204.71	6.41	202.41	
16	218.0	214.57	6.66	211.12	
17	229.0	225.12	6.94	221.23	
18	225.0	228.52	6.68	232.06	
19	204.0	219.57	5.56	235.21	
20	227.0	226.07	5.63	225.13	
21	223.0	227.34	5.31	231.69	
22	242.0	237.33	5.65	232.65	
23	239.0	240.99	5.51	242.98	
24	266.0	256.27	6.21	246.49	
25				262.48	1
26				268.69	2
27				274.90	3
28				281.11	4
29				287.32	5
30				293.53	6
Alpha	0.501				
Beta	0.072				



Time Series Analysis
Double exponential smoothing: example

Double Exponential Smoothing

Period	Actual
1	143.00
2	152.00
3	161.00
4	139.00
5	137.00
6	174.00
7	142.00
8	141.00
9	162.00
10	180.00
11	164.00
12	171.00
13	206.00
14	193.00
15	207.00
16	218.00
17	229.00
18	225.00
19	204.00
20	227.00
21	223.00
22	242.00
23	239.00
24	266.00

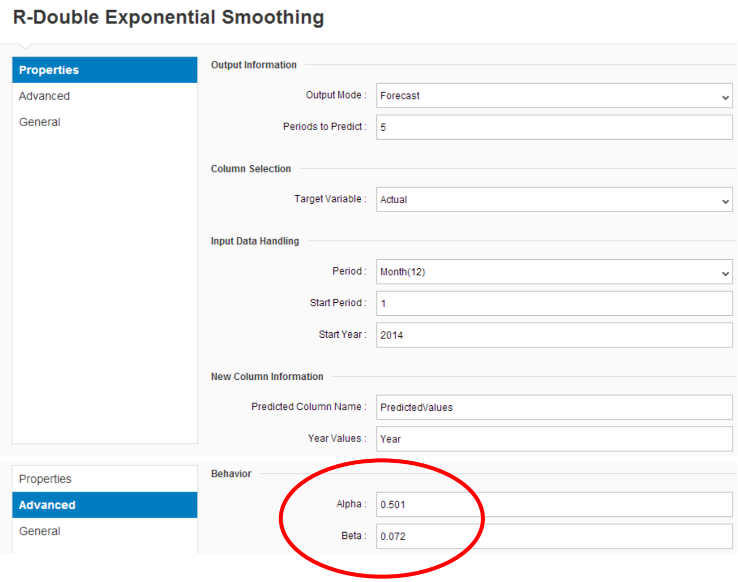


Time Series Analysis

Double exponential smoothing in SAP Predictive Analytics

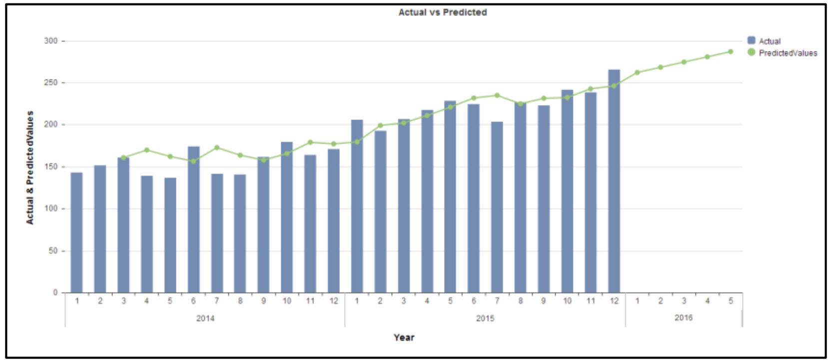
Double Exponential Smoothing

Alpha = 0.501 and Beta = 0.072



Parameter Settings

Year	Month	Period	Actual	Predicte..
2014	1	1	143.00	
2014	2	2	152.00	
2014	3	3	161.00	161.00
2014	4	4	139.00	170.00
2014	5	5	137.00	162.35
2014	6	6	174.00	156.62
2014	7	7	142.00	172.92
2014	8	8	141.00	163.91
2014	9	9	162.00	158.08
2014	10	10	180.00	165.84
2014	11	11	164.00	179.24
2014	12	12	171.00	177.36
2015	1	13	206.00	179.70
2015	2	14	193.00	199.35
2015	3	15	207.00	202.41
2015	4	16	218.00	211.12
2015	5	17	229.00	221.23
2015	6	18	225.00	232.06
2015	7	19	204.00	235.21
2015	8	20	227.00	225.13
2015	9	21	223.00	231.69
2015	10	22	242.00	232.65
2015	11	23	239.00	242.98
2015	12	24	266.00	246.49
2016	1			262.48
2016	2			268.69
2016	3			274.90
2016	4			281.11
2016	5			287.32



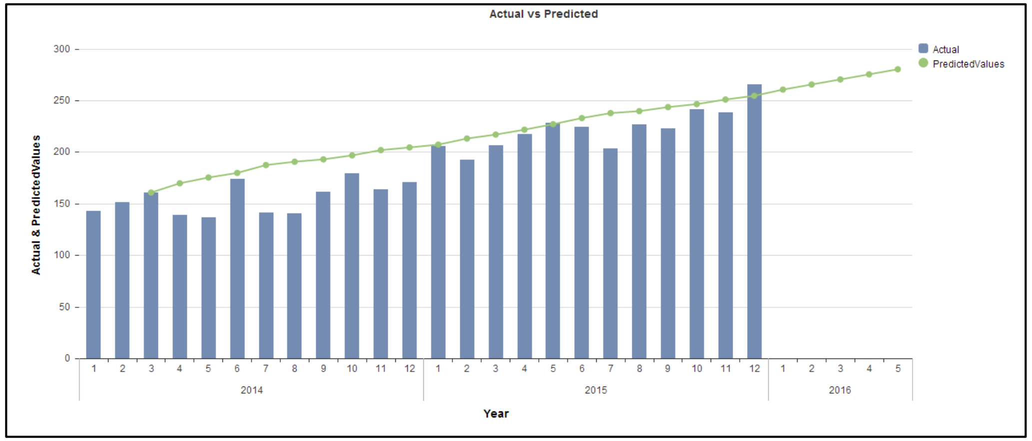
Output

Time Series Analysis

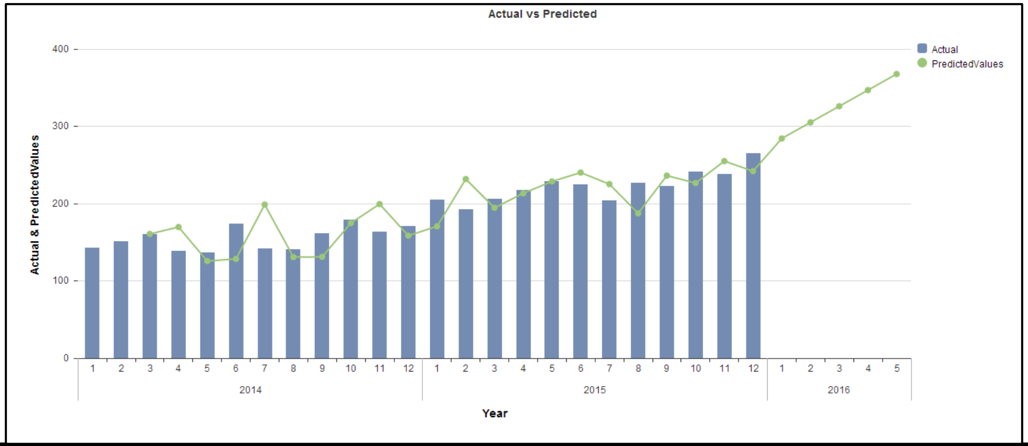
Double exponential smoothing in SAP Predictive Analytics

Double Exponential Smoothing

- **Alpha = 0.1, Beta 0.1**



- **Alpha = 0.9, Beta = 0.9**



Time Series Analysis

Triple exponential smoothing: worked example

Triple exponential smoothing for stationary and trend and seasonality

Period	Actual	Single Smoothing	Seasonal Smoothing	Trend Smoothing	Forecast
1	362		0.95		
2	385		1.01		
3	432		1.14		
4	341	380.00	0.90	9.75	
5	382	398.99	0.95	10.26	
6	409	404.68	1.01	10.01	414.64
7	498	433.90	1.14	11.06	471.43
8	387	433.70	0.90	10.44	399.29
9	473	487.08	0.95	12.81	423.22
10	513	505.25	1.01	13.10	506.40
11	582	512.86	1.14	12.80	589.59
12	474	527.89	0.90	12.92	471.57
13	544	565.05	0.95	14.25	515.88
14	582	575.31	1.01	14.03	586.92
15	681	597.11	1.14	14.46	670.26
16	557	619.21	0.90	14.88	548.66
17	628	653.76	0.95	15.96	605.17
18	707	692.87	1.01	17.24	678.47
19	773	685.01	1.14	15.86	807.73
20	592	667.09	0.90	14.00	628.87
21	627	661.07	0.95	12.90	650.26
22	725	708.00	1.01	14.77	683.05
23	854	746.07	1.14	16.05	821.77
24	661	741.57	0.90	14.92	683.41
25					721.98
26					782.25
27					894.38
28					718.27
29					778.94
30					842.78

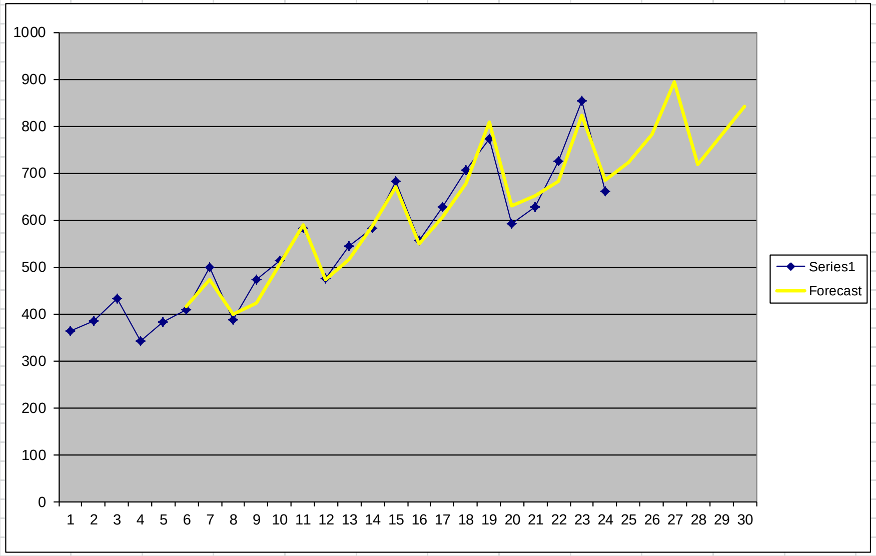
Alpha	0.822
Beta	0.055
Gamma	0.055

Initial value of Single Smooth 380.00 Average of the first 4 periods

Initial values of Seasonal Smoothing calculated as Period i = period i / average of first four periods (for quarterly data)

Intial values of Trend Smoothing calculated as difference between each quarter for the first 2 years averaged to give a trend estimate

20
24
66
46
9.75

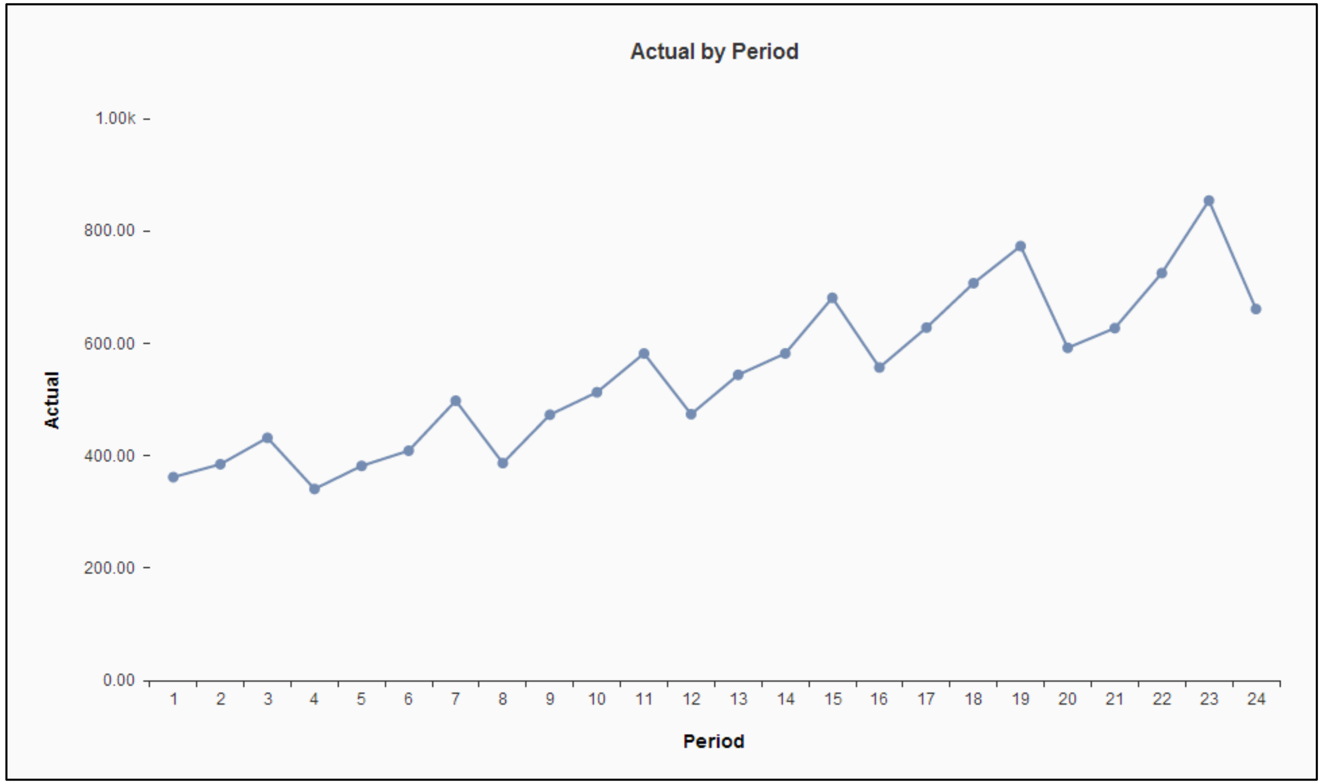


Time Series Analysis

Triple exponential smoothing: example

Triple Exponential Smoothing

123 Period	123 Actual
1	362
2	385
3	432
4	341
5	382
6	409
7	498
8	387
9	473
10	513
11	582
12	474
13	544
14	582
15	681
16	557
17	628
18	707
19	773
20	592
21	627
22	725
23	854
24	661



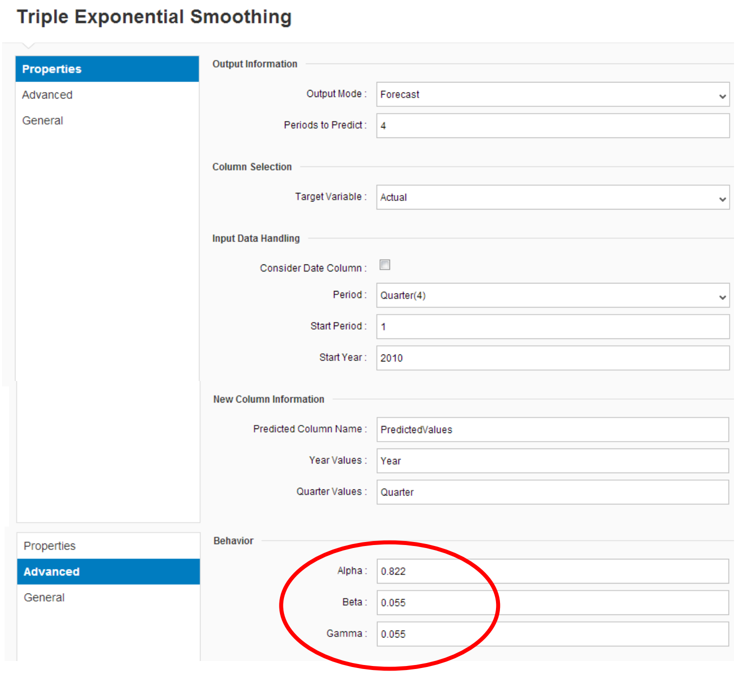
Time Series Analysis

Triple exponential smoothing in SAP Predictive Analytics: example

Triple Exponential Smoothing

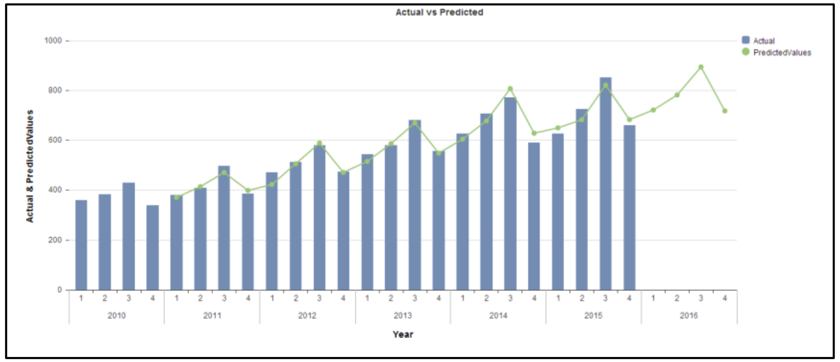
Alpha = 0.822 ; Beta = 0.055 ;

Gamma = 0.055



Parameter Settings

Year	Quarter	Period	Actual	Predicte..
2010	1	1	362	
2010	2	2	385	
2010	3	3	432	
2010	4	4	341	
2011	1	5	382	371.29
2011	2	6	409	414.64
2011	3	7	498	471.43
2011	4	8	387	399.29
2012	1	9	473	423.22
2012	2	10	513	506.40
2012	3	11	582	589.59
2012	4	12	474	471.57
2013	1	13	544	515.88
2013	2	14	582	586.92
2013	3	15	681	670.26
2013	4	16	557	548.66
2014	1	17	628	605.17
2014	2	18	707	678.47
2014	3	19	773	807.73
2014	4	20	592	628.87
2015	1	21	627	650.26
2015	2	22	725	683.05
2015	3	23	854	821.77
2015	4	24	661	683.41
2016	1			721.98
2016	2			782.25
2016	3			894.38
2016	4			718.27



Output

Time Series Analysis

ARIMA autoregressive integrated moving average

- Autocorrelation coefficients provide a key method of identifying patterns in time series - trend, seasonality, and randomness.
- Correlation is the association between two variables. The degree of correlation is measured by the correlation coefficient.
- The autocorrelation coefficient describes the association between time series values of the same variable but at different time periods or lags.

The Autocorrelation Coefficient of time lag k is given by

$$r_k = \frac{\sum_{t=1}^{n-k}(X_t - \overline{X})(X_{t+k} - \overline{X})}{\sum_{t=1}^{n}(X_t - \overline{X})^2}$$

where

r_k is the autocorrelation coefficient of time period lag k

n is the number of observations

X_t is the value of the time series variable at time t

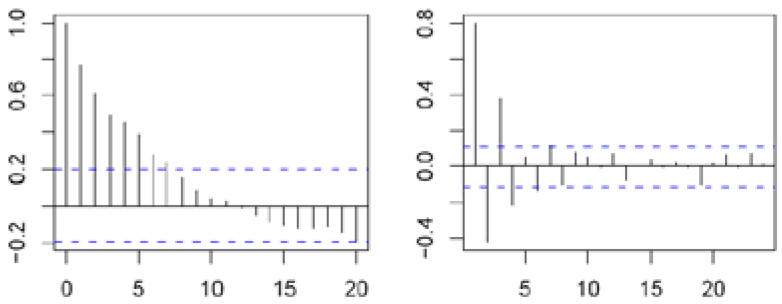
and $\overline{X}$ is the mean of all the data

- AR(1) $Y_t = c + \alpha_1 Y_{t-1} + e_t$ AR(2) $Y_t = c + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + e_t$
- MA(1) $Y_t = c + e_t - \beta_1 e_{t-1}$ MA(2) $Y_t = c + e_t - \beta_1 e_{t-1} - \beta_2 e_{2-1}$
- ARMA(1,1) or ARIMA(1,0,1) $Y_t = c + \alpha_1 Y_{t-1} + e_t - \beta_1 e_{t-1}$

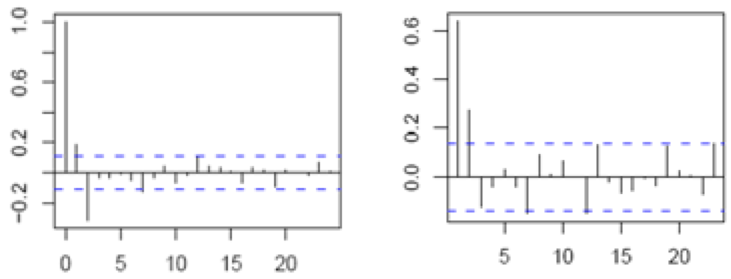
Time Series Analysis

ARIMA autoregressive integrated moving average

Example of exponential decay



Example of significant lags

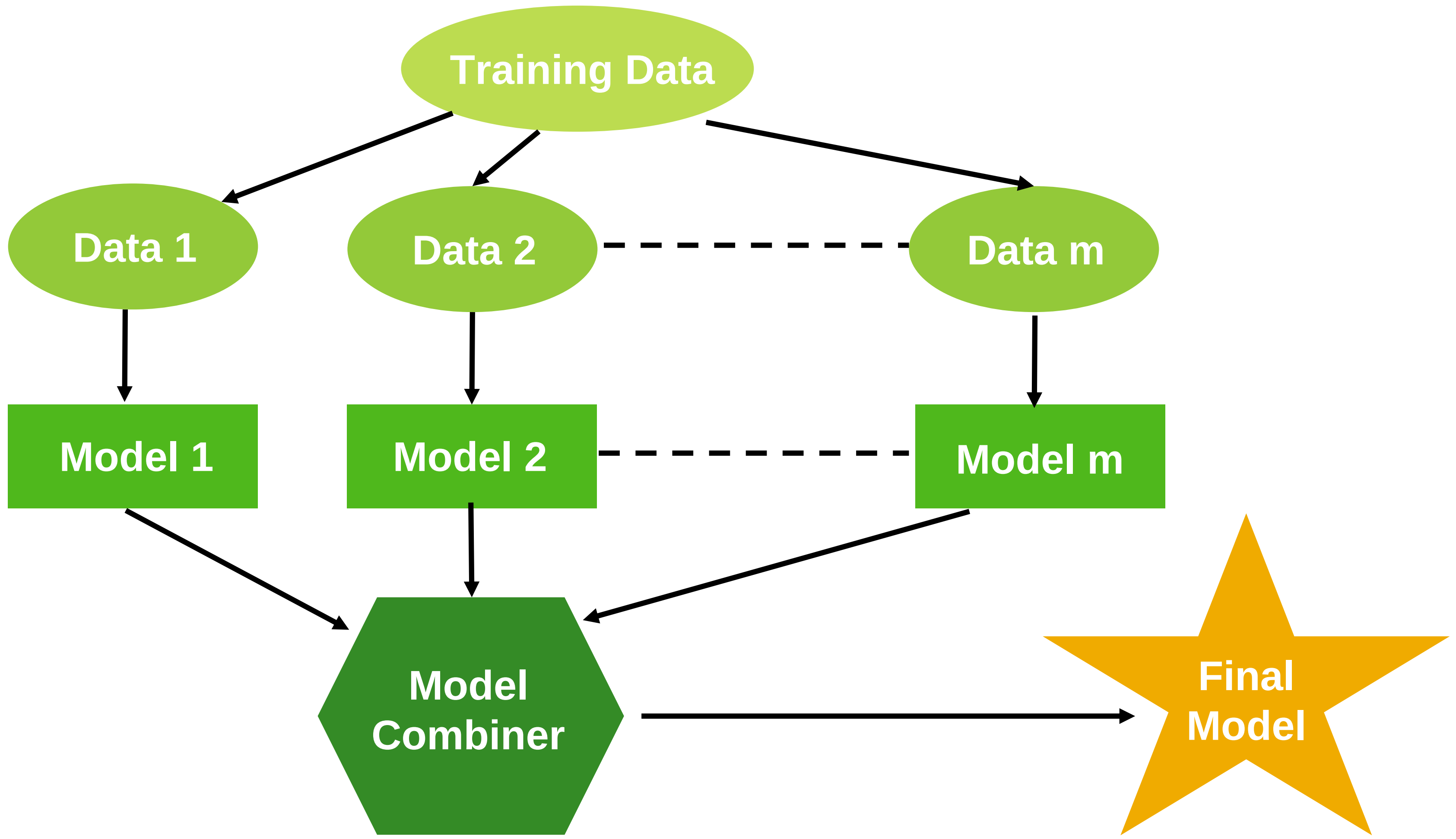


Model	ACF	PACF
White Noise	All zeros	All zeros
AR(p)	Exponential Decay	P significant lags before dropping to zero
MA(q)	q significant lags before dropping to zero	Exponential Decay
ARMA(p,q)	Decay after q^{th} lag	Decay after p^{th} lag

Week 4 Unit 4: **Ensemble Methods**

Ensemble Methods

Introduction

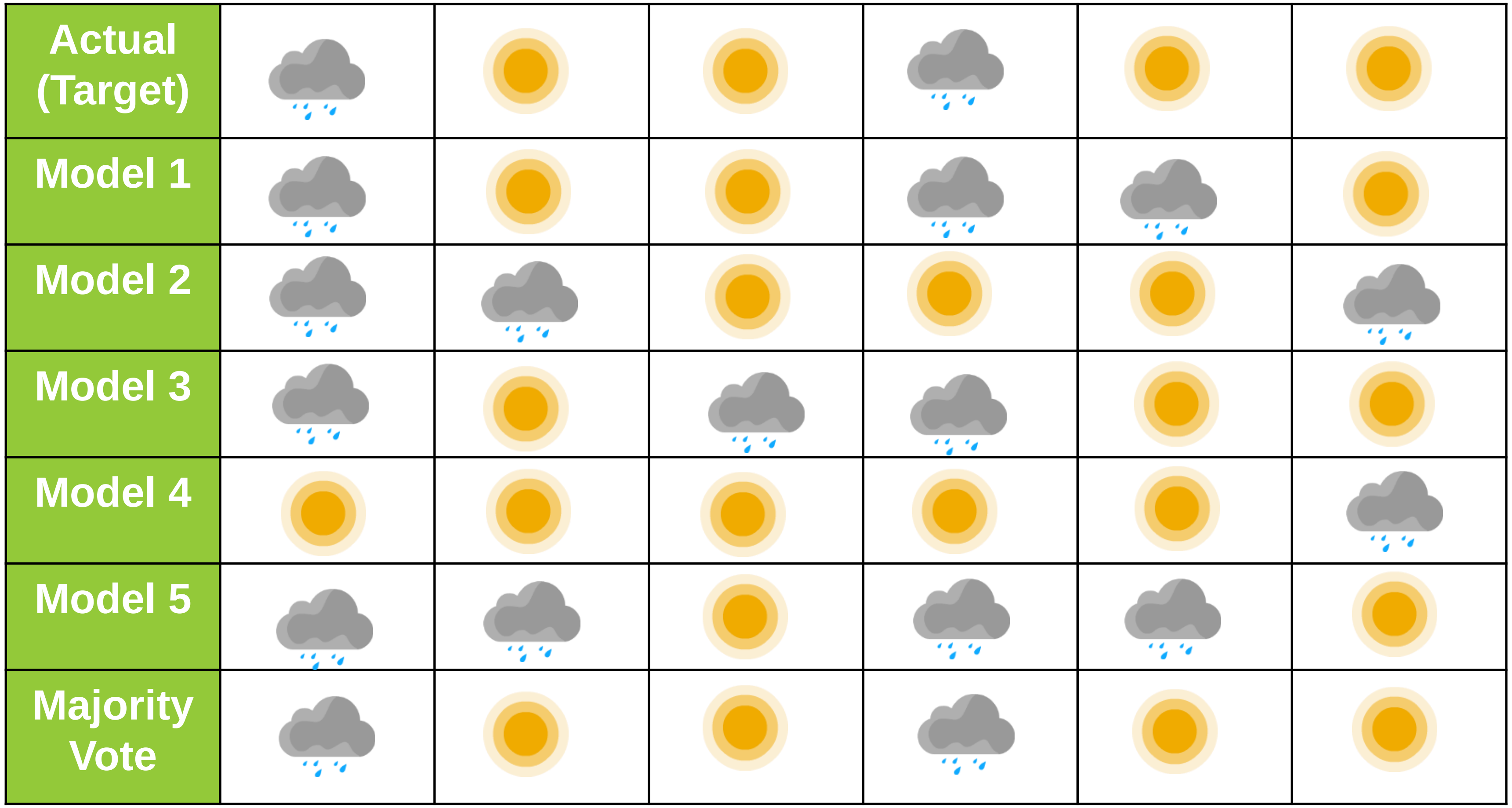


Ensemble Methods

Voting and averaging-based ensemble methods

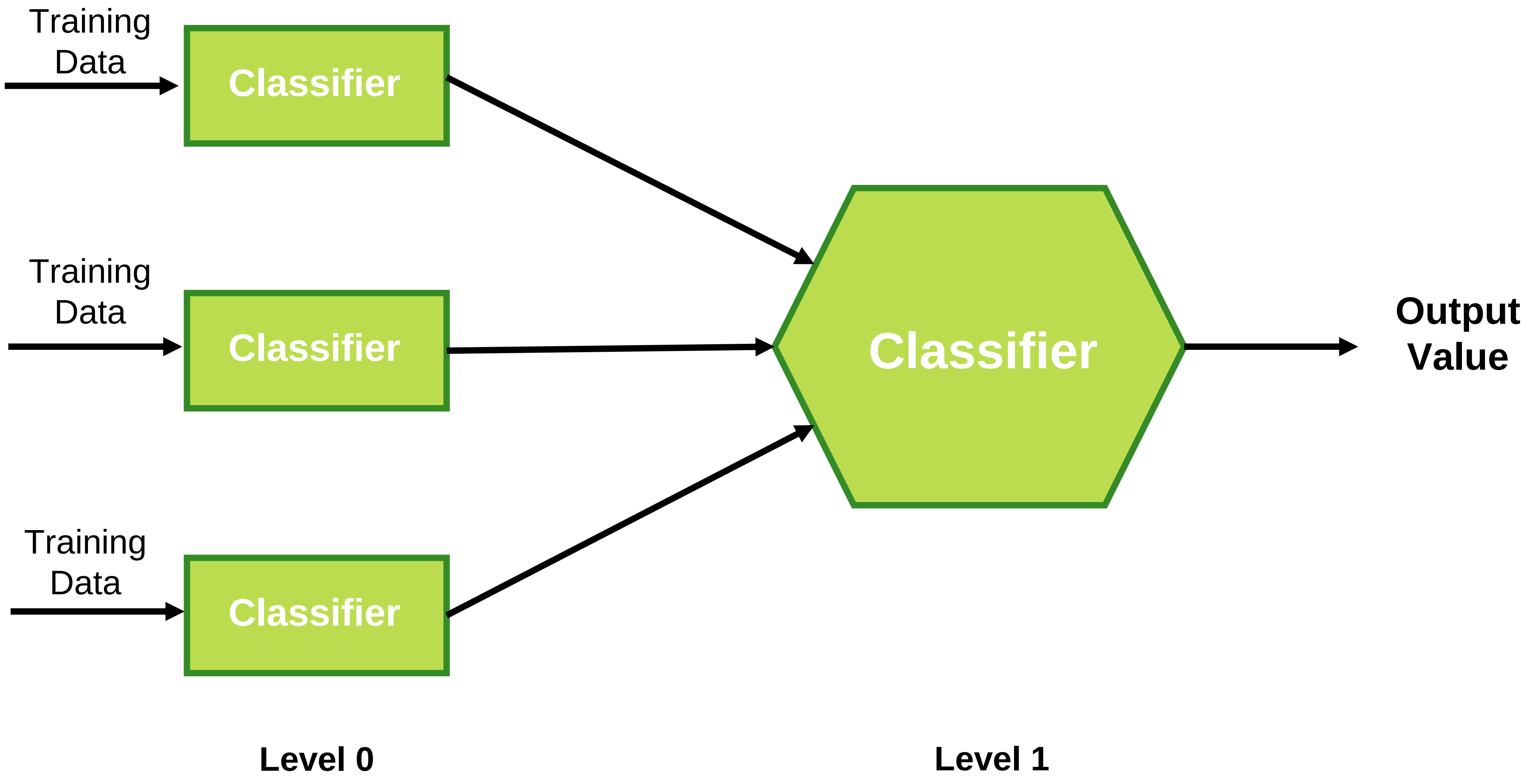
- Voting and averaging are two of the easiest ensemble methods. They are both easy to understand and implement.
- Voting is used for classification and averaging is used for regression.

Weather Forecast



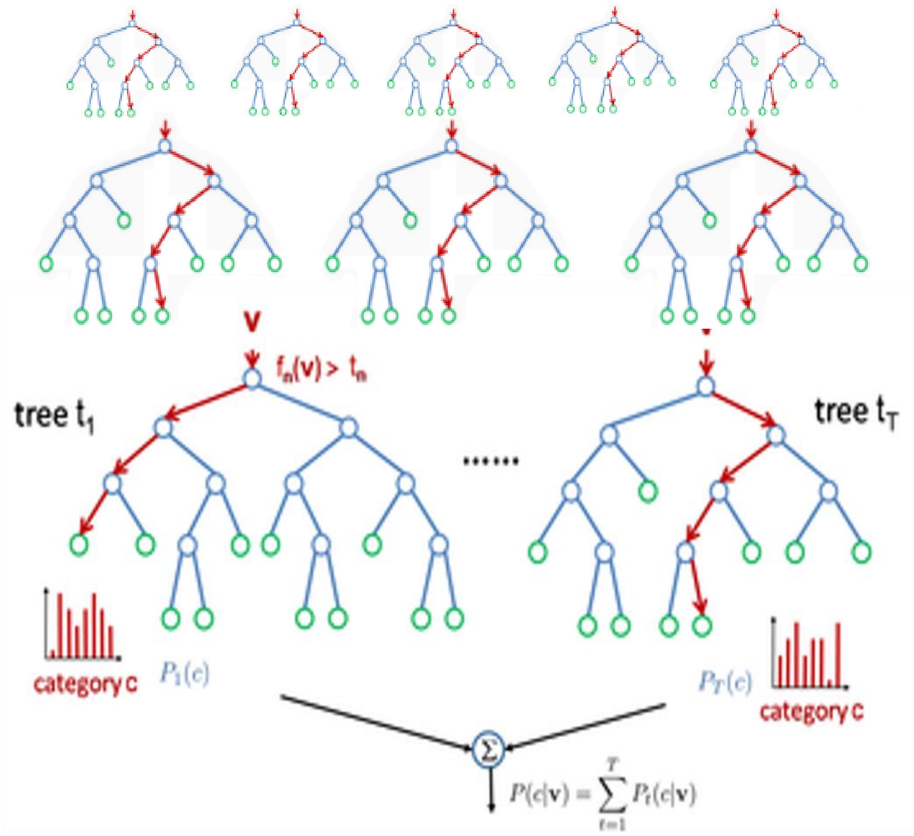
Ensemble Methods

Stacking multiple machine learning models



Ensemble Methods

SAP HANA Predictive Analysis Library: random forest



A random forest of many decision trees

Bootstrap sample $\Rightarrow$ $f_1(x)$

Bootstrap sample $\Rightarrow$ $f_2(x)$

Bootstrap sample $\Rightarrow$ $f_3(x)$

…

Bootstrap sample $\Rightarrow$ $f_M(x)$

Combine $f_1(x),\ldots, f_M(x)$ $\Rightarrow$ $f(x)$

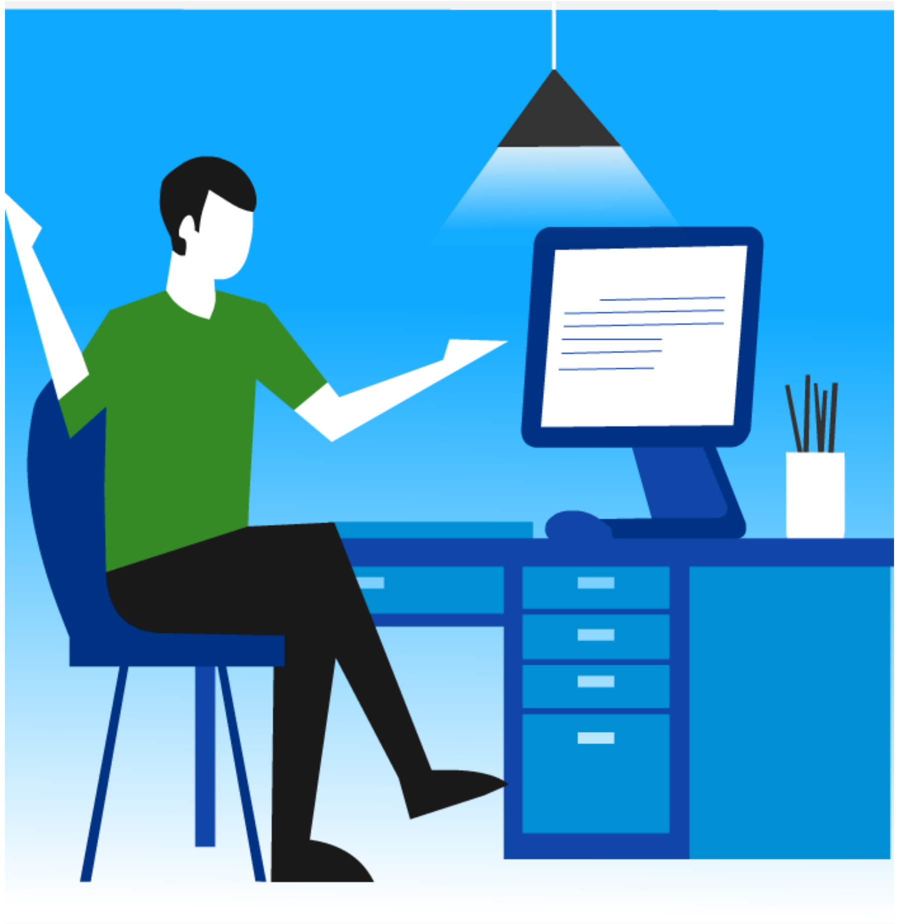
MODEL AVERAGING

$f_i(x)$'s are "base learners"

Ensemble Methods

Summary

- All three approaches combine several machine learning techniques into one predictive model.
- They average out biases (boosting)
- They reduce the variance (bagging)
- They improve the predictive power (stacking)
- They are unlikely to over-fit



Thank you

Contact information:

open@sap.com

open.sap.com

Week 4 Unit 5: **Simulation & Optimization**

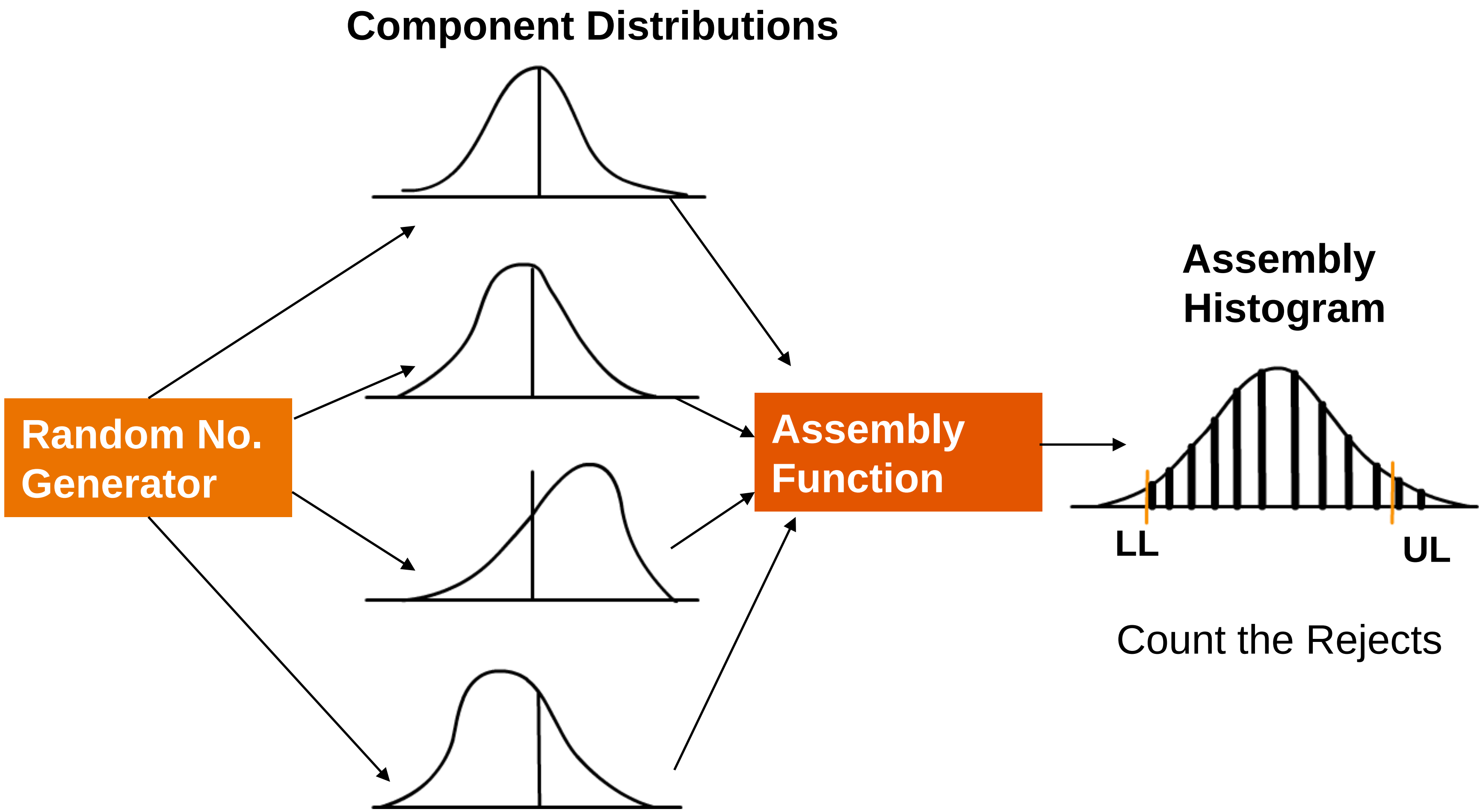
Simulation & Optimization

Monte Carlo simulation: introduction

- Monte Carlo simulation is a mathematical technique that allows you to account for risk in quantitative analysis and decision making.
- It uses repeated random sampling to obtain the distribution of an unknown probabilistic entity.

Simulation & Optimization

Monte Carlo simulation: simple example



Simulation & Optimization
Finance example

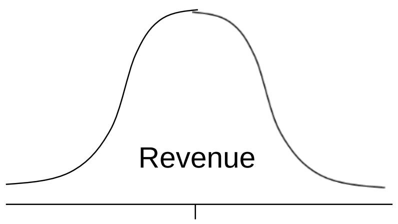
Our simple analysis shows an NPV of £8,382 – this is a “deterministic” approach that could be improved with simulation.

Example Monte Carlo Simulation Example					
John MacGregor July 2014					
	Year 1	**Year 2**	**Year 3**	**Year 4**	**Year 5**
Deterministic values					
Product Revenue	£0	£3,000	£8,000	£18,000	£30,000
Product Costs	£1,000	£1,000	£2,500	£7,000	£10,000
Product Margin	-£1,000	£2,000	£5,500	£11,000	£20,000
Overheads	£1,500	£2,000	£2,500	£3,000	£3,500
Total Profit	-£2,500	£0	£3,000	£8,000	£16,500
Capital Investment	£10,000	£2,000			
Cash Flow	-£12,500	-£2,000	£3,000	£8,000	£16,500
Net Present Value of Investment	£8,382				

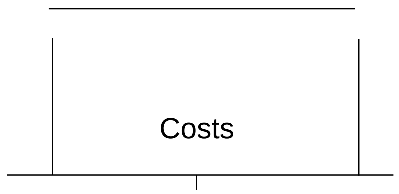
Simulation & Optimization
Finance example

- How simulation works:

- Product Revenue – sample from a normal distribution using the mean of the deterministic model and estimated variance:



- Product Costs – sample from a rectangular distribution, for example:



- Derive Product Margin
- Likewise for Overheads and Capital Investment to derive the project NPV
- Repeat N times, to create the probability distributions of the Product Margin, Profit, Cash Flow, and NPV

Simulation & Optimization

Finance example

… Run N times (the number of trials or simulations) from which we derive N values of the NPV, and thus the probability distribution of the NPV from which we can estimate Pr(NPV >0), Pr(NPV >10,000) ...

Run 3

Example Monte Carlo Simulation Example	Year 1	Year 2	Year 3	Year 4	Year 5
Deterministic values					
Product Revenue	£0	£2,956	£9,211	£17,003	£27,009
Product Costs	£1,250	£998	£2,401	£5,989	£11,969
Product Margin	-£1,250	£1,958	£6,810	£11,014	£15,040
Overheads	£1,459	£1,799	£2,009	£2,309	£3,006
Total Profit	-£2,709	£159	£4,801	£8,705	£12,034
Capital Investment	£10,132	£1,522			
Cash Flow	-£12,841	-£1,363	£4,801	£8,705	£12,034
Net Present Value of Investment	£7,272				

Run 2

Net Present Value of Investment £10,204

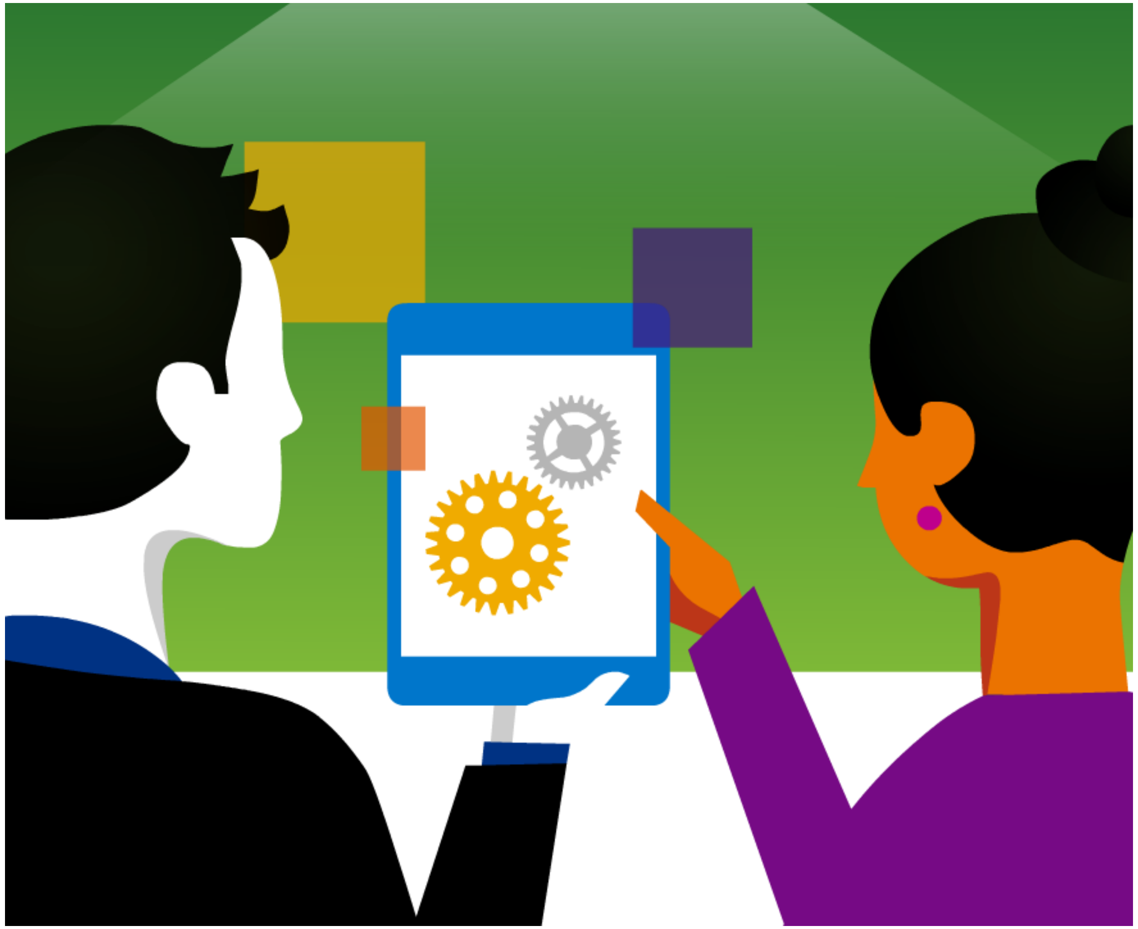
Run 1

Net Present Value of Investment £4,140

Simulation & Optimization

Monte Carlo simulation: summary

- In finance and business, Monte Carlo simulation can be used to explore the probability of outcomes.
- It offers a probabilistic approach, in contrast to the usual deterministic approach to model building.
- It offers a logical / visual approach to problem solving.
- It is part of the very significant topic for operations researchers (older name for data scientists ☺).



Simulation & Optimization

Optimization: introduction

How many of product X and Y should we produce to maximize profits?

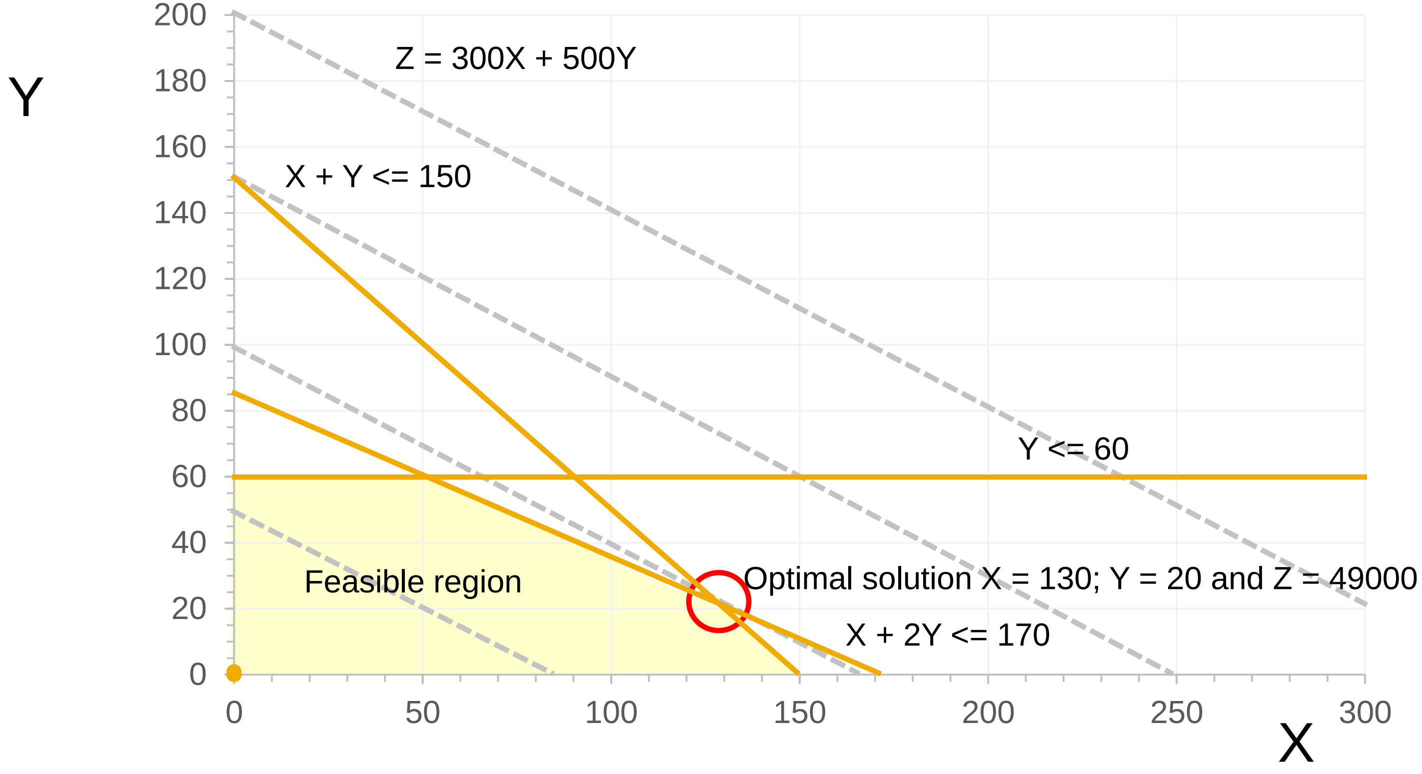
Maximise Z = 300X + 500Y (this is called the "objective function")

subject to the constraints -

X + 2Y <= 170

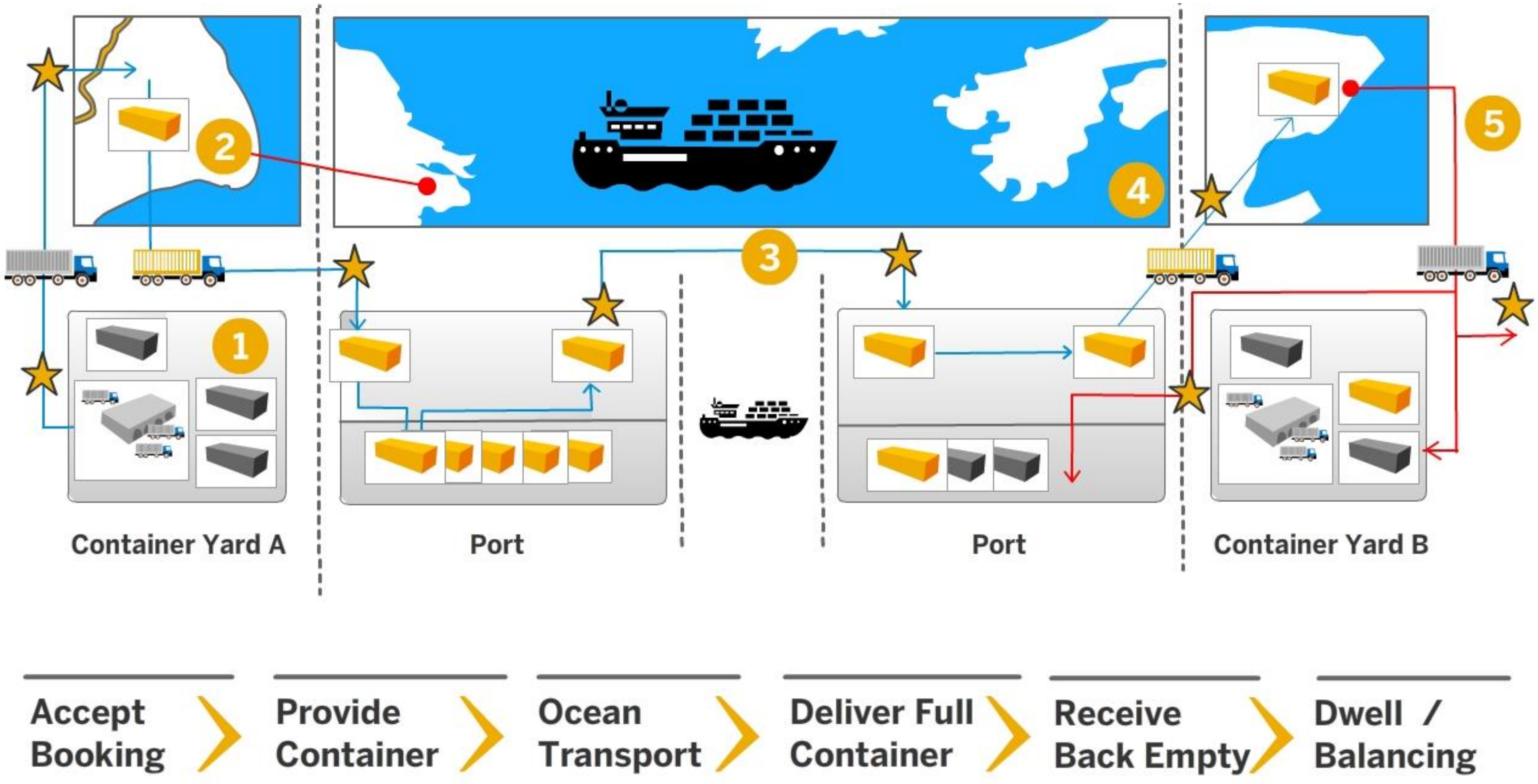
X + Y <= 150

Y <= 60



Simulation & Optimization

Optimization in SAP Transportation Resource Planning



Thank you

Contact information:

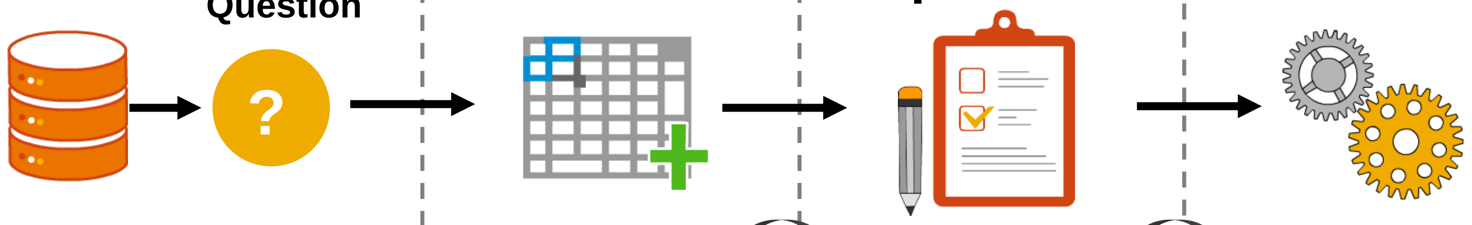
open@sap.com

Week 4 Unit 6: **Automated Modeling**

open.sap.com

Automated Modeling

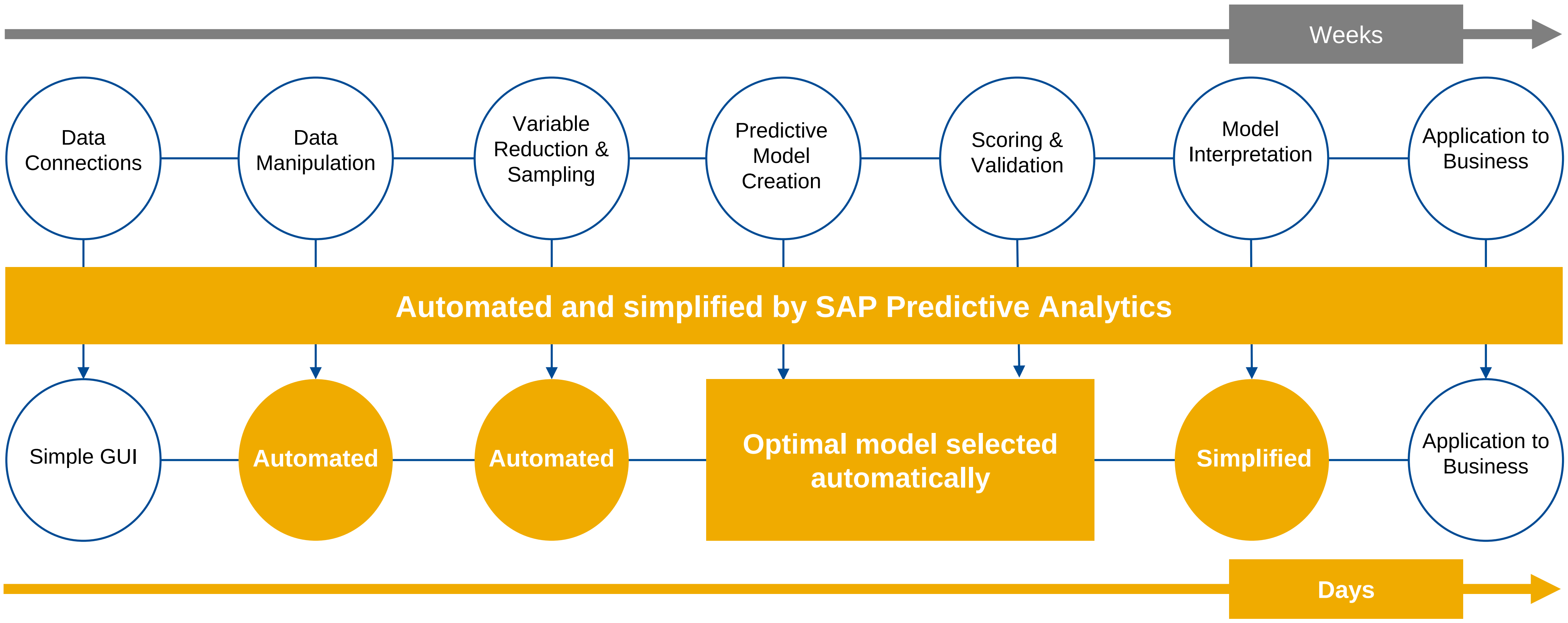
Introduction – Traditional vs. automated modeling approach



	Data → Business Question	Build ADS	Prepare Data	Build Model
Traditional Tools		**Create SQL manually:** • Write SQL code • Execute it on a regular basis	**Choose the right techniques:** • Treat missing values • Identify outliers • Identify correlations	**Choose the right algorithm with the right parameters:** SVM, decision trees, KNN, neural network, K-means
SAP Predictive Analytics (automated)		**Create SQL with GUI:** • Create SQL using an easy interface • Create ADS with time-stamped population	**All automatically handled**	**Simply choose the data mining function:** classification, regression, forecasting…

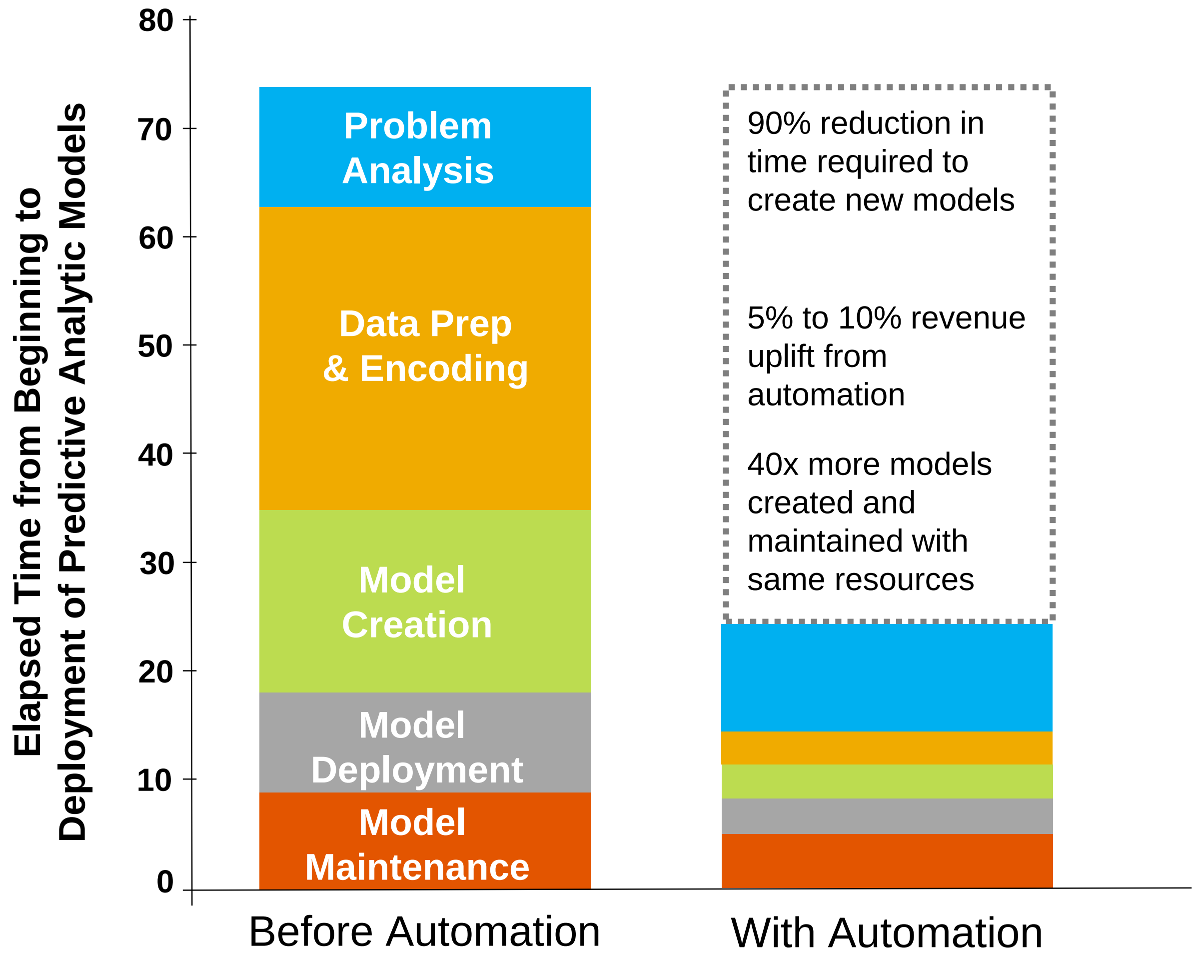
Automated Modeling

Introduction – Traditional vs. automated modeling approach



Automated Modeling

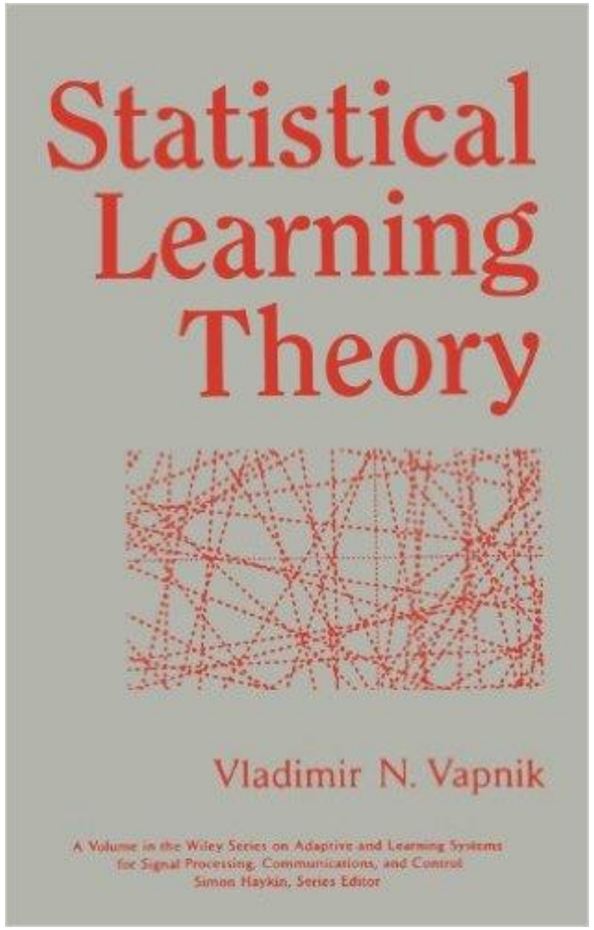
Automated modeling benefits



Automated Modeling

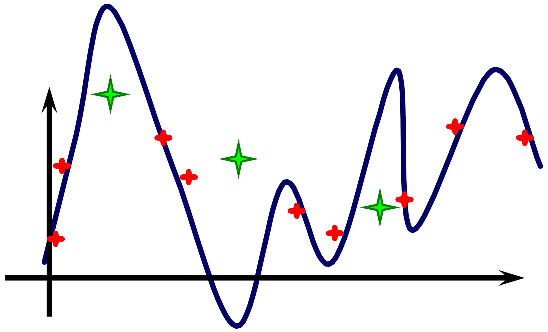
Vapnik and structural risk minimization (SRM)

Professor Vladimir N. Vapnik



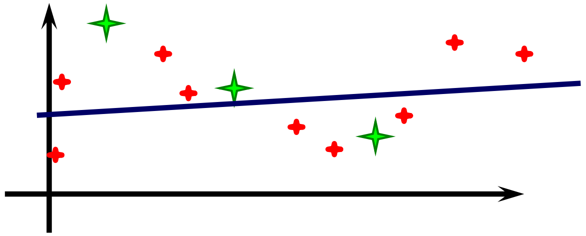
Automated Modeling

Basics of SRM – Automatically selecting the best model



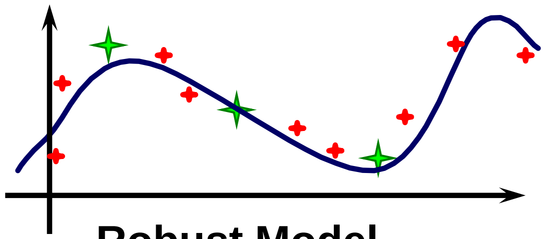
Over-Fit Model/Low Robustness

(No Training Error, High Test Error)



Under-Fit Model/High Robustness

(High Training Error = High Test Error)



Robust Model

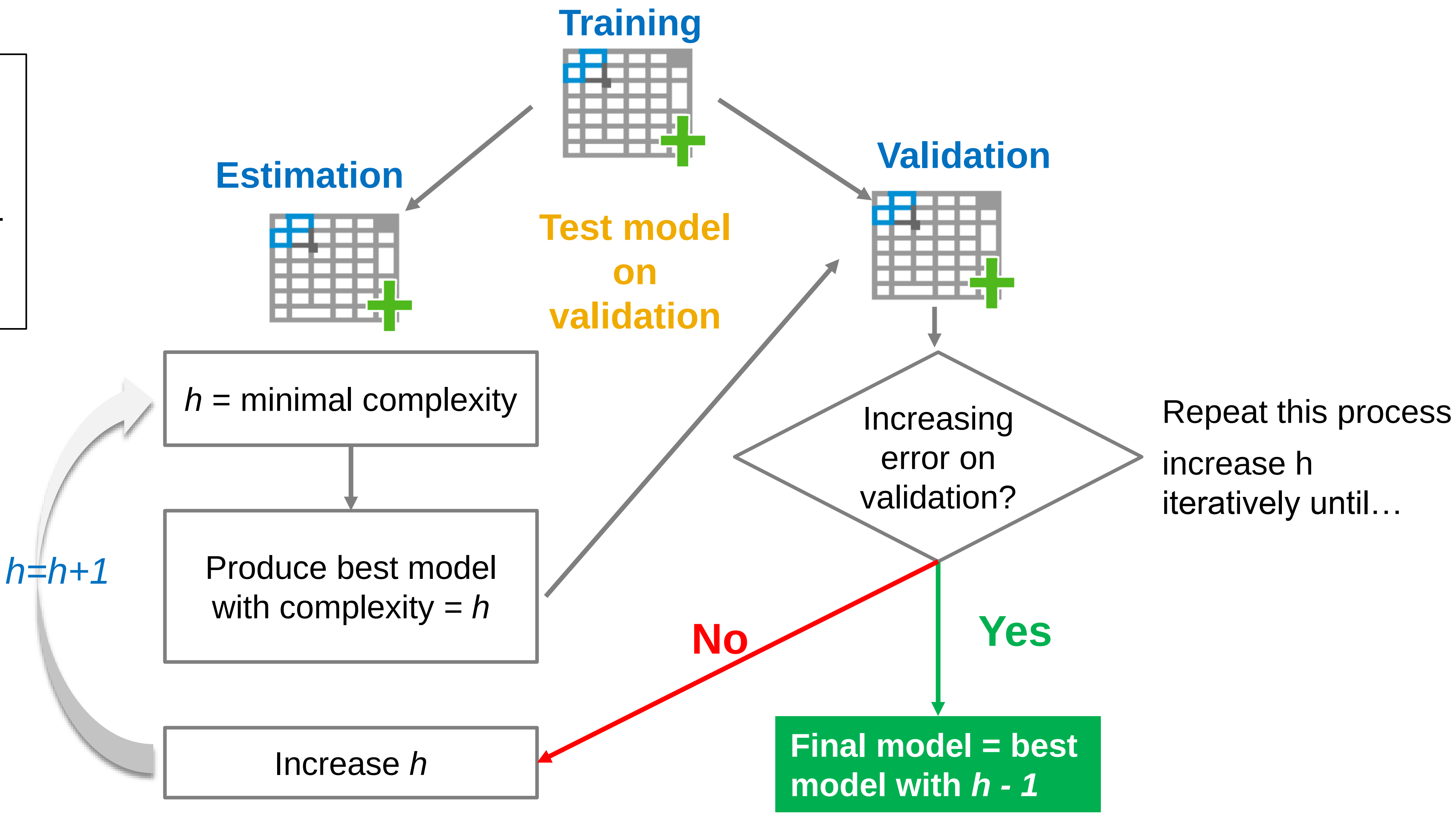
(Low Training Error ≈ Low Test Error)

Model Built
Known Data
New Data

Automated Modeling

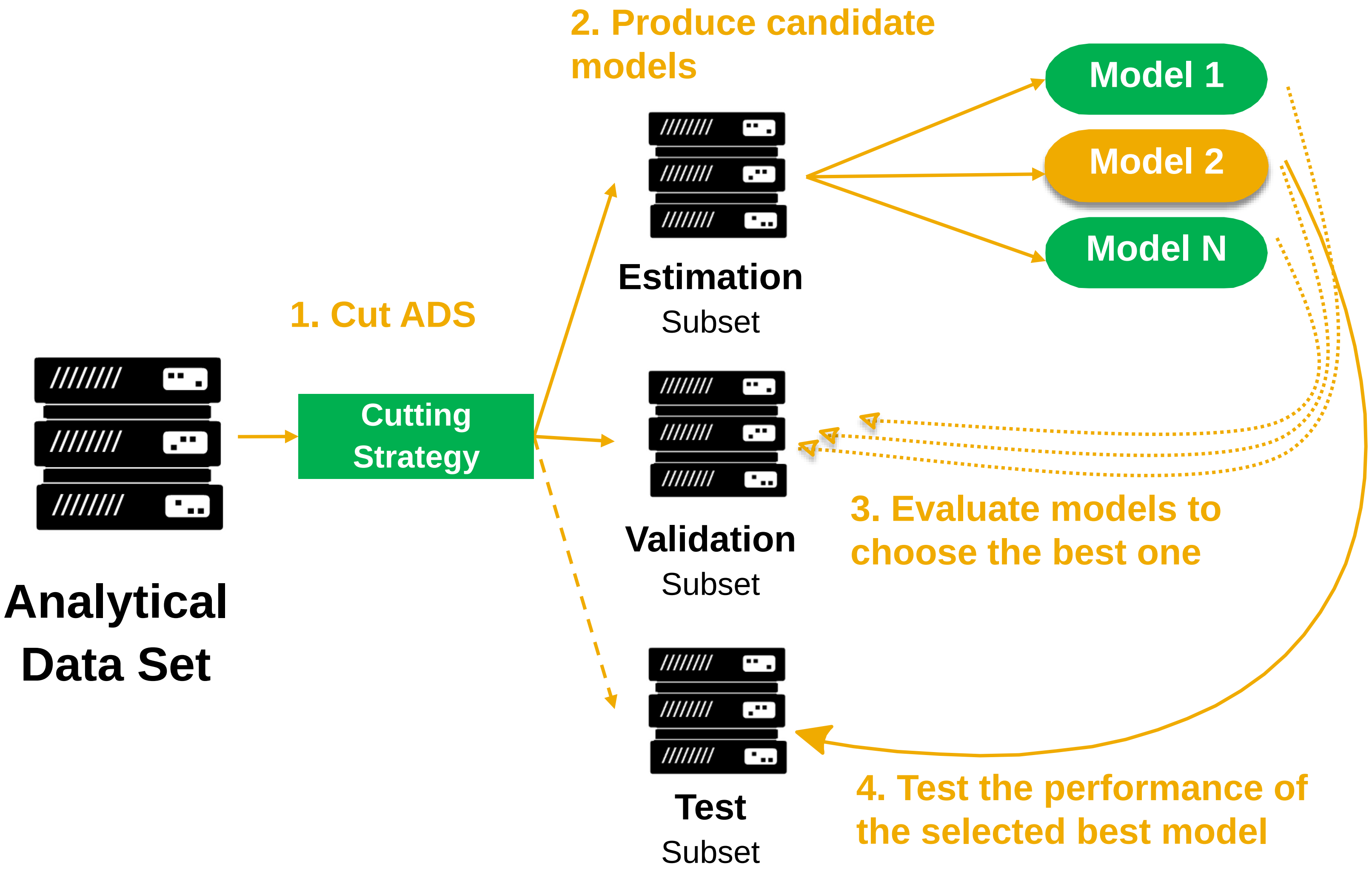
Vapnik's principles – Process overview

- h is a measure of the **complexity** of the model.
- It is called the Vapnik-Chervonenkis (VC) dimension

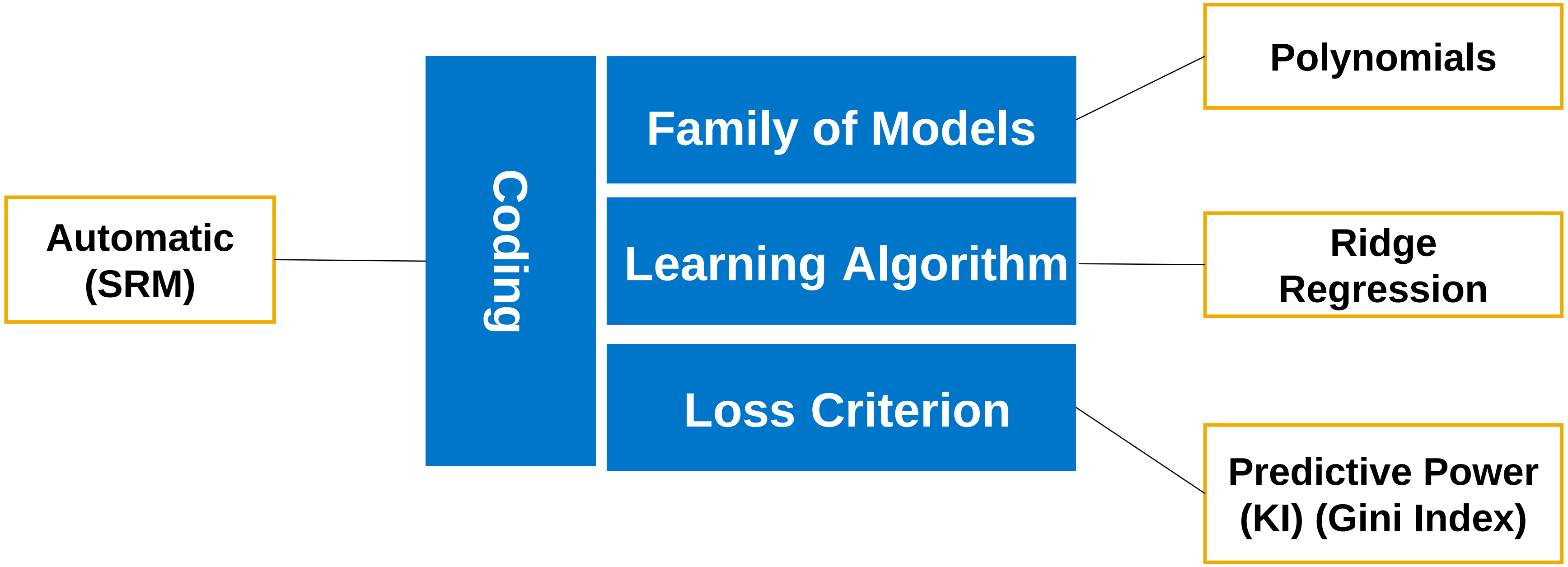


Automated Modeling

Cutting strategy in model selection



Automated Modeling
SRM implementation



Automated Modeling

Missing values in SAP Automated Analytics?

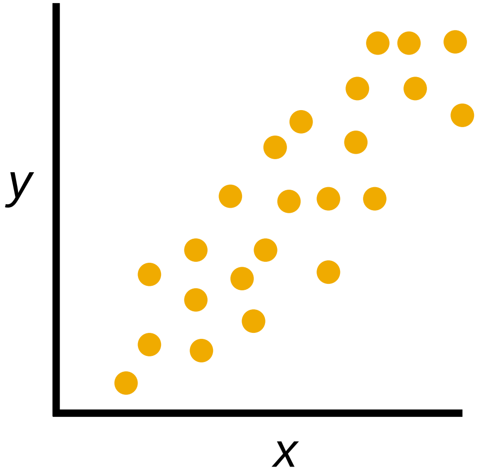
Missing values rarely occur by chance.

- Missing values in SAP Automated Analytics are not excluded – they are replaced with a constant called KxMissing and then treated by the model as any other category.

CITY
Paris
London
KxMissing
New York
KxMissing
KxMissing

Automated Modeling

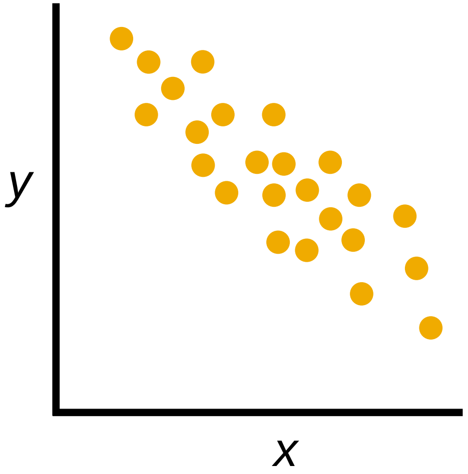
How are correlations handled?



Positive correlation

The observations lie close to a straight line, which has a positive gradient.

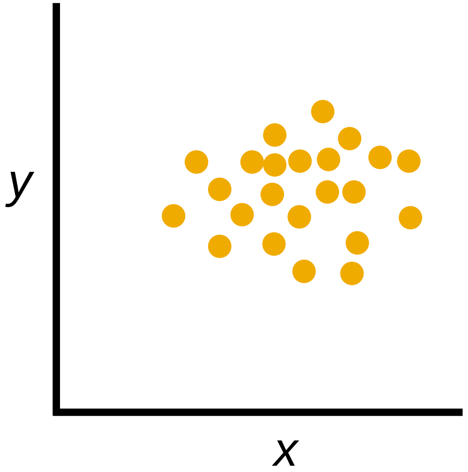
This shows that as one variable increases the other increases.



Negative correlation

The observations lie close to a straight line, which has a negative gradient.

This shows that as one variable increases the other decreases.



No correlation

There is no pattern in the observations.

This shows that there is no connection between the two variables.

Thank you

Contact information:

open@sap.com