MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

INFORMATION AND COMMUNICATION TECHNOLOGIES
Master study programme

Data and Text Mining

Petra Kralj Novak

December 2, 2019

http://kt.ijs.si/petra_kralj/dmtm2.html

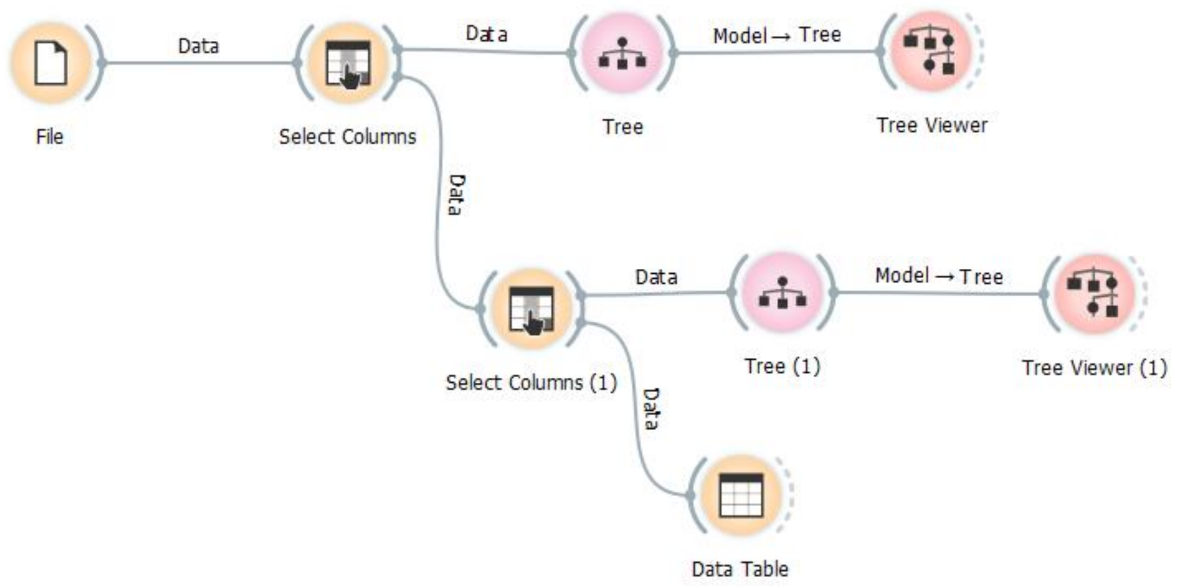
In previous episodes …

- 23-Oct-19
  - **Data**, data types
  - Interactive **visualization** (Orange)
  - **Classification** with decision trees (root, leaves, rules, entropy, info gain, TDIDT, ID3)
- 6-Nov-19
  - Classification: train – test (evaluate) - apply
  - **Decision tree** example (on blackboard)
  - Decision tree language bias (Orange workflow)
  - Homework:
    - InfoGain questions
    - Orange workflow
    - Reading "Classification and regression by randomForest" by Liaw & Wiener, 2002
- 25-Nov-19
  - **Evaluation**:
    - Methods: train-test, leave-one-out, randomized sampling,…
    - Metrics: accuracy, confusion matrix, precision, recall, F1,…
  - Homework: XOR, questions, precision and recall

Assignment 1

1. Sketch the real decision tree model behind the data of the XOR example.
2. What happens if we remove the attribute “C”? Guess first, then use an Orange workflow and find out.

A	B	C	AxorB
1	1	1	0
1	1	1	0
1	0	1	1
1	0	0	1
0	1	0	1
0	1	0	1
0	0	1	0
0	0	0	0



Assignment 2: Questions

1. What do we get when testing on the training set?
2. Can we always get a 100% accuracy on the training set?
3. When do we use “leave-one-out”?
4. What is stratified sampling?
5. When is classification accuracy “good”?

Assignment 3: Compute the precision, recall and F1 for both classifiers for the class Fraud

Two confusion matrices for two classifiers

		Predicted		
		Fraud	**Not Fraud**	
Actual	**Fraud**	**0**	**4**	4
	Not fraud	**0**	**9996**	9996
		0	10000	

		Predicted		
		Fraud	**Not Fraud**	
Actual	**Fraud**	**4**	**0**	4
	Not fraud	**300**	**9696**	9996
		304	9696	

For the class *Fraud*

- Precision=
- Recall=
- F1=

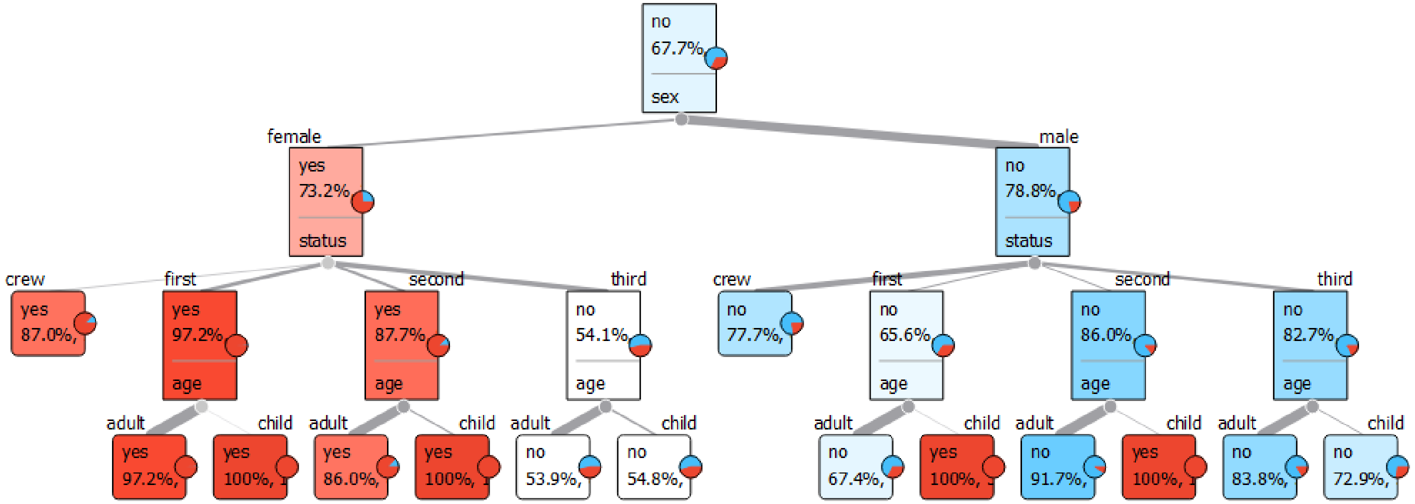
- Precision=
- Recall=
- F1=

Homework

- Express F1 in terms of the entries in the confusion matrix (TP, FP, TN, FN) and simplify the equation.

High precision and/or high recall?

- Can we make a model more precise (increase precision)?
- How sure is the model about a certain prediction?
- We can set different thresholds and get different binary classifiers.
- Find a trade-off between precision and recall appropriate for a problem at hand.



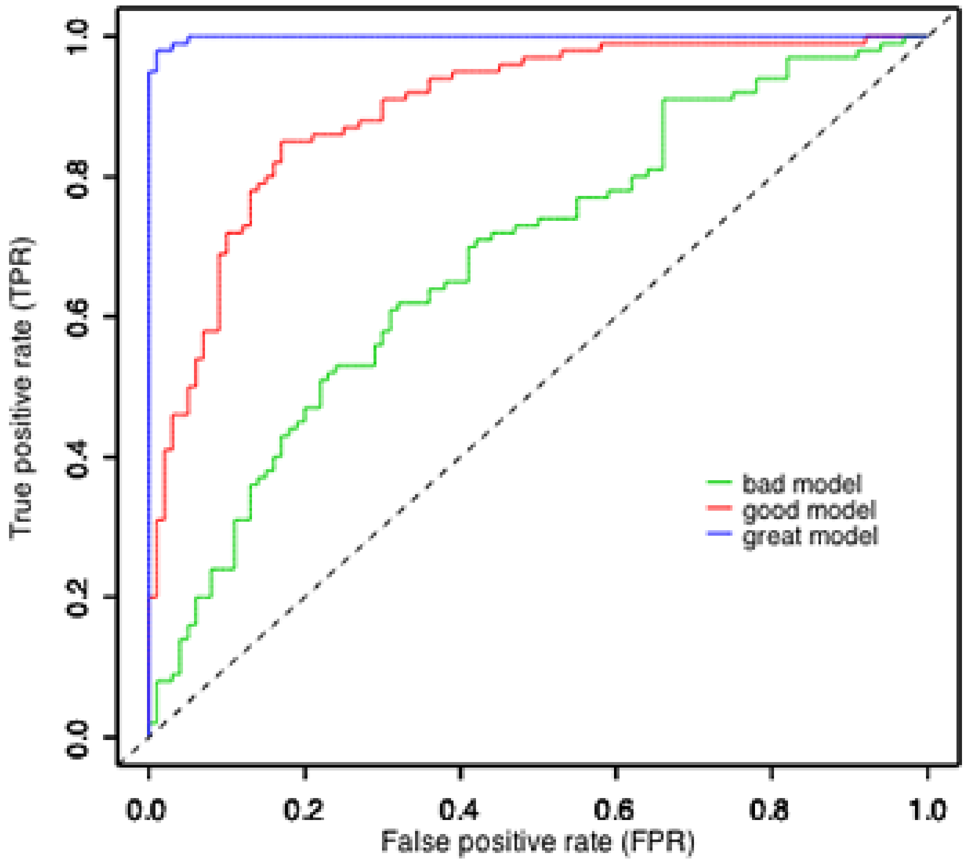
Probabilistic classification

- A **probabilistic** classifier is a classifier that is able to predict, given an observation of an input, a **probability** distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to.
- Ranking
- Tresholds/cutpoints

	Actual class	Confidence classifier forclass Y
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

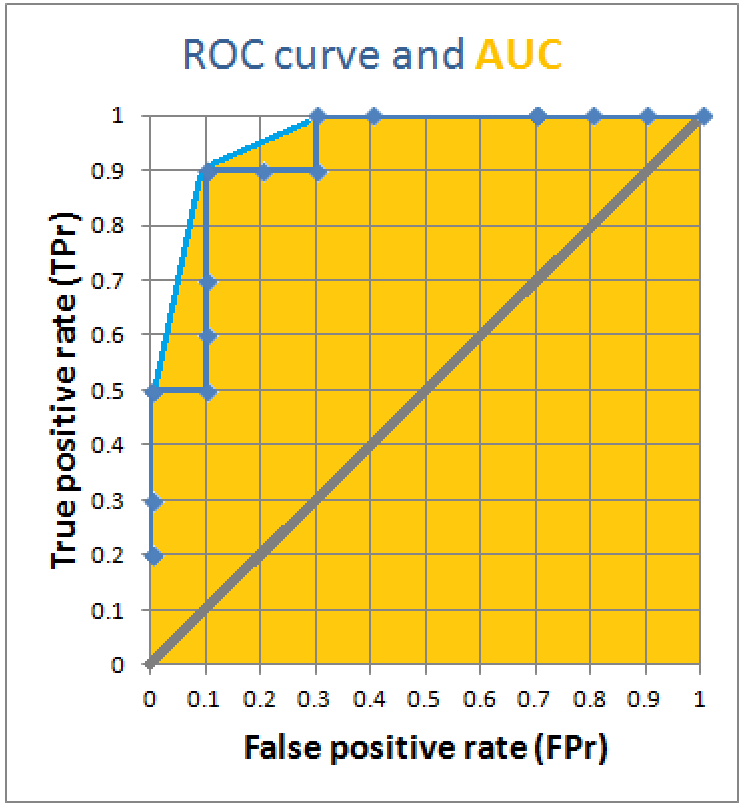
ROC curve and AUC

- **Receiver Operating Characteristic curve** (or ROC curve) is a plot of the true positive rate (TPr=Sensitivity=Recall) against the false positive rate (FPr) for different possible cutpoints.
- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve to the top left corner, the "better" the classifier.
- The diagonal represents the random classifiers (predicting the positive class with some probability regardless the data).



AUC - Area Under (ROC) Curve

- Performance is measured by the area under the ROC curve (AUC). An area of 1 represents a perfect classifier; an area of 0.5 represents a worthless classifier.
- The area under the curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

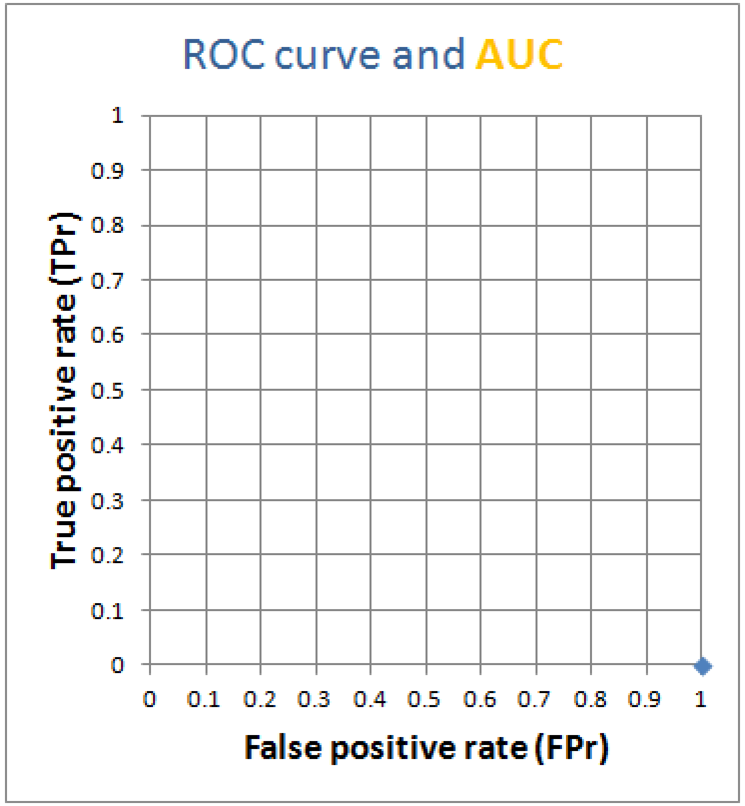


Exercise: ROC curve and AUC

	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				

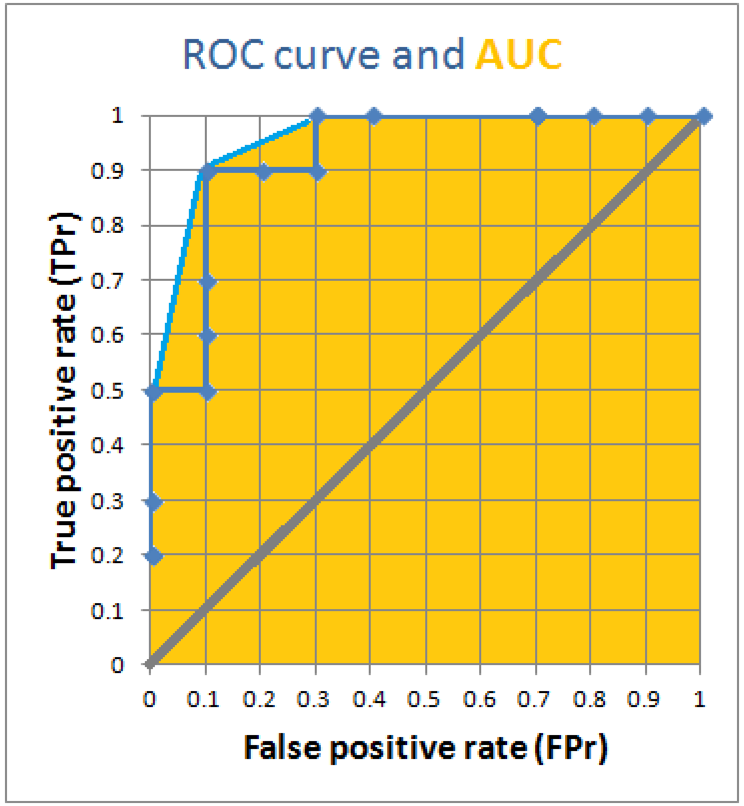
ROC curve and AUC

	Actual class	Classifier confidence forclass Y	FP	TP	FPr	TPr
P1	Y	1	0	2	0	0.2
P2	Y	1	0	2	0	0.2
P3	Y	0.95	0	3	0	0.3
P4	Y	0.9	0	5	0	0.5
P5	Y	0.9	0	5	0	0.5
P6	N	0.85	1	5	0.1	0.5
P7	Y	0.8	1	6	0.1	0.6
P8	Y	0.6	1	7	0.1	0.7
P9	Y	0.55	1	9	0.1	0.9
P10	Y	0.55	1	9	0.1	0.9
P11	N	0.3	2	9	0.2	0.9
P12	N	0.25	3	9	0.3	0.9
P13	Y	0.25	3	10	0.3	1
P14	N	0.2	4	10	0.4	1
P15	N	0.1	7	10	0.7	1
P16	N	0.1	7	10	0.7	1
P17	N	0.1	7	10	0.7	1
P18	N	0	8	10	0.8	1
P19	N	0	9	10	0.9	1
P20	N	0	10	10	1	1



ROC curve and **AUC**

	Actual class	Classifier confidence forclass Y	FP	TP	FPr	TPr
P1	Y	1	0	2	0	0.2
P2	Y	1	0	2	0	0.2
P3	Y	0.95	0	3	0	0.3
P4	Y	0.9	0	5	0	0.5
P5	Y	0.9	0	5	0	0.5
P6	N	0.85	1	5	0.1	0.5
P7	Y	0.8	1	6	0.1	0.6
P8	Y	0.6	1	7	0.1	0.7
P9	Y	0.55	1	9	0.1	0.9
P10	Y	0.55	1	9	0.1	0.9
P11	N	0.3	2	9	0.2	0.9
P12	N	0.25	3	9	0.3	0.9
P13	Y	0.25	3	10	0.3	1
P14	N	0.2	4	10	0.4	1
P15	N	0.1	7	10	0.7	1
P16	N	0.1	7	10	0.7	1
P17	N	0.1	7	10	0.7	1
P18	N	0	8	10	0.8	1
P19	N	0	9	10	0.9	1
P20	N	0	10	10	1	1



Area Under (the convex) Curve
AUC = 0.96

Probabilistic classification

A **probabilistic** classifier is a classifier that is able to predict, given an observation of an input, a **probability** distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to.

$$p(C_k \mid x_1, \ldots, x_n)$$

Naïve Bayes Classifier

Basic probability refresh

- Probability of A

$$P(A)$$

- Independence

$$P(A \cap B) = P(A)P(B)$$
$$P(A|B) = P(A)$$
$$P(B|A) = P(B)$$

- Conditional probability

$$P(A|B) = P(A, B)/P(B)$$

- Bayes' Rule

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

$$P(A|B, C) = \frac{P(B|A, C)P(A|C)}{P(B|C)}$$

The idea behind the Naïve Bayes Classifier

- We are interested in the probability of the class C given the attribute values X_1, X_2, X_3, …. , X_n

$$P(C|X_1X_2 \dots X_n)$$

- We „**naively**“ assume that all attribute values X_1, X_2, X_3, …. , X_n are mutually independent, conditional on the category C

$$P(X_1X_2 \dots X_n|C) \approx P(X_1|C) \cdot P(X_2|C) \cdot \dots \cdot P(X_n|C)$$

Homework

- Learn about the derivation of the Naïve Bayes formula https://en.wikipedia.org/wiki/Naive_Bayes_classifier

$$p(C_k, x_1, \ldots, x_n) \qquad = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})} \quad = \qquad \cdots \qquad = \quad p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Naïve Bayes Classifier

Conditional probability of attribute value v_i given class c

Attribute values

$$P(c_i \mid a_1 = v_1, a_2 = v_2, \dots, a_j = v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

Class c_i

* where ∝ denotes proportionality
* The results are not probabilities (they do not sum up to 1). The formula is simplified for easy implementation (and time complexity), while the results are proportional to the estimates of the probabilities of a class given the attribute values.

Exercise: Naïve Bayes Classifier

Color	Size	Time	Caught
black	large	day	**YES**
white	small	night	**YES**
black	small	day	**YES**
red	large	night	**NO**
black	large	night	**NO**
white	large	night	**NO**

$$P(c_i \mid a_1 = v_1, a_2 = v_2, \ldots, a_j = v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

- Does the spider catch a white ant during the night?
- Does the spider catch the big black ant at daytime?

Exercise: Naïve Bayes Classifier

Does the spider catch a white ant during the night?

Color	Size	Time	Caught
black	large	day	YES
white	small	night	YES
black	small	day	YES
red	large	night	NO
black	large	night	NO
white	large	night	NO

$$P(c_i \mid a_1 = v_1, a_2 = v_2, \ldots, a_j = v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

$$v_1 = \text{“}Color = white\text{”}$$

$$v_2 = \text{“}Time = night\text{”}$$

$$c_1 = YES$$

$$c_2 = NO$$

$$\begin{aligned} P(C_1|v_1, v_2) = & \\ &= P(\mathrm{YES}|C = w, T = n) \\ &= P(\mathrm{YES}) \cdot P(C = w|\mathrm{YES}) \cdot P(T = n|\mathrm{YES}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{1}{3} \\ &= \frac{1}{18} \end{aligned}$$

$$\begin{aligned} P(C_2|v_1, v_2) = & \\ &= P(\mathrm{NO}|C = w, T = n) \\ &= P(\mathrm{NO}) \cdot P(C = w|\mathrm{NO}) \cdot P(T = n|\mathrm{NO}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot 1 \\ &= \frac{1}{6} \end{aligned}$$

Exercise: Naïve Bayes Classifier

Does the spider catch the big black ant at daytime?

Color	Size	Time	Caught
black	large	day	YES
white	small	night	YES
black	small	day	YES
red	large	night	NO
black	large	night	NO
white	large	night	NO

$$P(c_i \mid a_1 = v_1, a_2 = v_2, \ldots, a_j = v_j) \propto P(c_i) \times \prod_{j=1}^{n} P(a_j = v_j \mid class = c_i)$$

Ant 2: Color = black, Size = large, Time = day

$$v_1 = \text{“}Color = black\text{”} = \text{“}C = b\text{”}$$

$$v_2 = \text{“}Size = large\text{”} = \text{“}S = l\text{”}$$

$$v_3 = \text{“}Time = day\text{”} = \text{“}T = d\text{”}$$

$$c_1 = YES$$

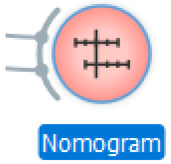
$$c_2 = NO$$

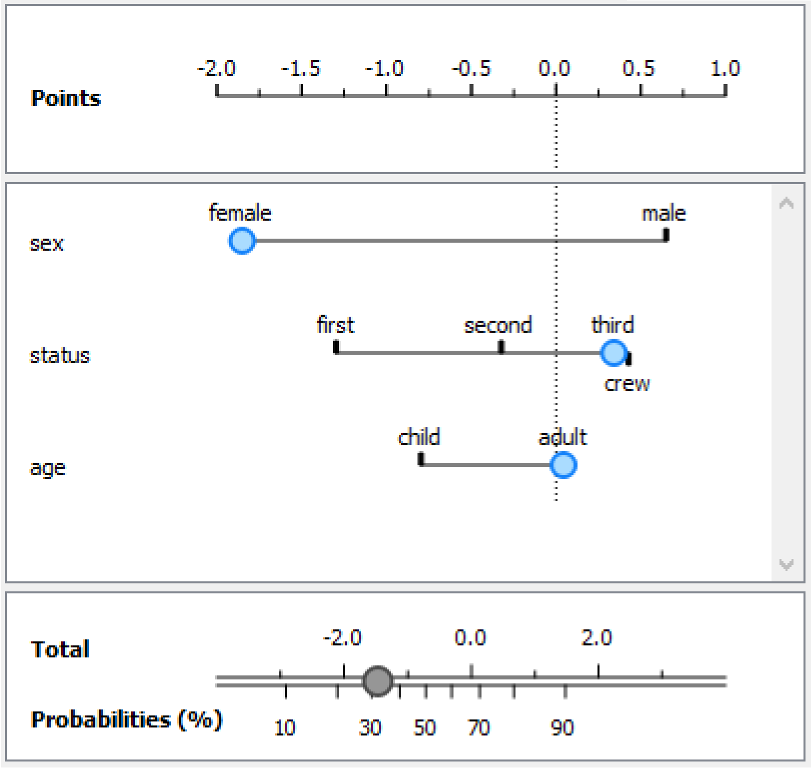
$$\begin{aligned} P(C_1|v_1, v_2, v_3) &= \\ &= P(\text{YES}|C = b, S = l, T = d) \\ &= P(\text{YES}) \cdot P(C = b|\text{YES}) \cdot P(S = l|\text{YES}) \cdot P(T = d|\text{YES}) \\ &= \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \\ &= \frac{4}{54} = \frac{2}{27} \end{aligned}$$

$$\begin{aligned} P(C_2|v_1, v_2, v_3) &= \\ &= P(\text{NO}|C = b, S = l, T = d) \\ &= P(\text{NO}) \cdot P(C = b|\text{NO}) \cdot P(S = l|\text{NO}) \cdot P(T = d|\text{NO}) \\ &= \frac{1}{2} \cdot \frac{1}{3} \cdot \frac{3}{3} \cdot 0 \\ &= 0 \end{aligned}$$

Use of Naïve Bayes

- Frequently used in practice
  - Medical diagnisis
    - The attributes are inherently chosen to be as independent as possible
    - NB is not sensitive to missing data
  - Simple text classification(features are words)
    - Classification of news into categories
    - Spam detection
  - ....
- Why?
  - Simple
  - Not sensitive to missing values
  - Uses all the available data
  - Very few parameters
  - Visualization with nomograms





Probability Estimation

Estimating probability

- In machine learning we often estimate probabilities from small samples of data and their subsets:
  - In the 5th depth of a decision tree we have just about 1/32 of all training examples.
- Estimate the probability based on the amount of evidence and of the prior probability
  - Coin flip: prior probability 50% - 50%
  - One coin flip does not make us believe that the probability of heads is 100%
  - More evidence can make us suspect that the coin is biased

Estimating probability

Relative frequency

- **P(c) = n(c) /N**
- A disadvantage of using relative frequencies for probability estimation arises with small sample sizes, especially if the probabilities are either very close to zero, or very close to one.
- In our spider example:

  P(Time=day|caught=NO) =

  = 0/3 = 0

n(c) … number of examples where c is true
N … number of all examples
k … number of possible events

Relative frequency vs. Laplace estimate

Relative frequency

- **P(c) = n(c) /N**
- A disadvantage of using relative frequencies for probability estimation arises with small sample sizes, especially if the probabilities are either very close to zero, or very close to one.
- In our spider example:

  P(Time=day|caught=NO) =

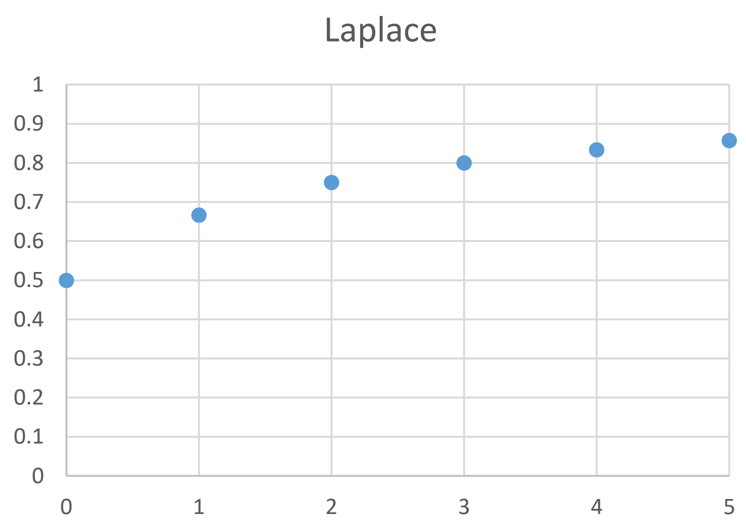
  = 0/3 = 0

n(c) … number of examples where c is true
N … number of all examples
k … number of possible events

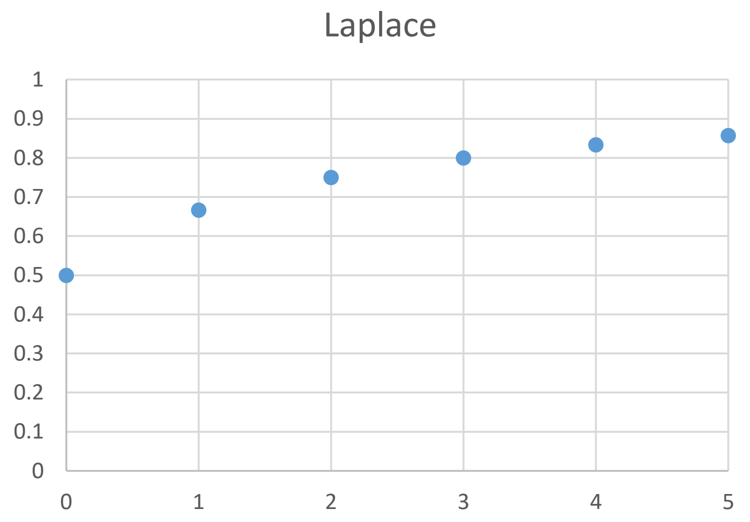
Laplace estimate

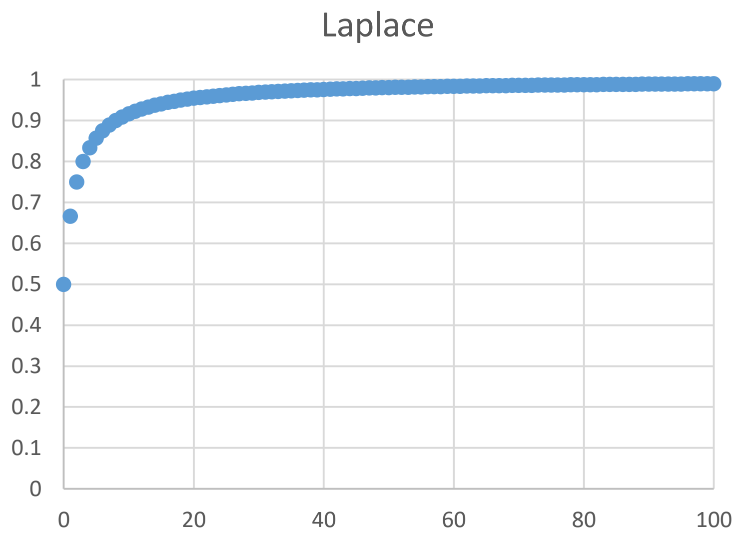
- Assumes uniform prior distribution over the probabilities for each possible event
- **P(c) = (n(c) + 1) / (N + k)**
- In our spider example: P(Time=day|caught=NO) = (0+1)/(3+2) = 1/5
- With lots of evidence it approximates relative frequency
- If there were 300 cases when the spider didn't catch ants at night: P(Time=day|caught=NO) = (0+1)/(300+2) = 1/302 = 0.003
- With Laplace estimate probabilities can never be 0.

Laplace estimate



Laplace estimate





Homework

- Compare the Naïve Bayes classifier with decision trees.
- How do we evaluate the Naïve Bayes classifier? Methods, metrics.
- Estimate the probabilities of C1 and C2 in the table below by relative frequency and Laplace estimate.

Number of events		**Relative frequency**		**Laplace estimate**	
Class C1	Class C2	P(C1)	P(C2)	P(C1)	P(C2)
0	2				
12	88				
12	988				
120	880				

Literature

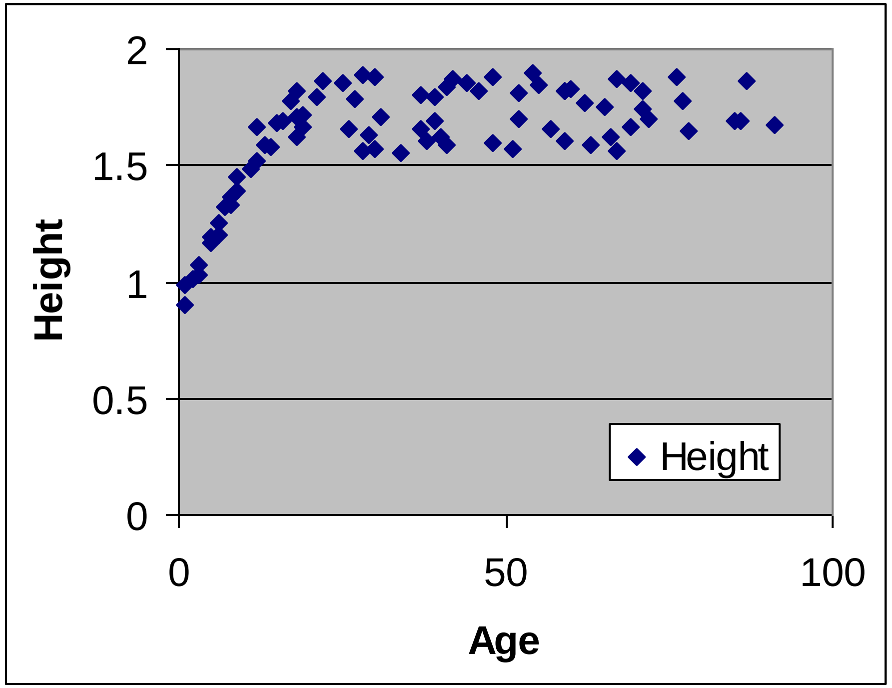
- Max Bramer: Principles of data mining (2007)
  - 2. Introduction to Classification: Naive Bayes and Nearest Neighbour

  On pg. 30, there is a mistake where it says “making the assumption that the attributes are independent” … it should be “conditionally independent given the class”. Refer to https://en.wikipedia.org/wiki/Naive_Bayes_classifier

Numeric prediction

Example

- data about 80 people: Age and Height



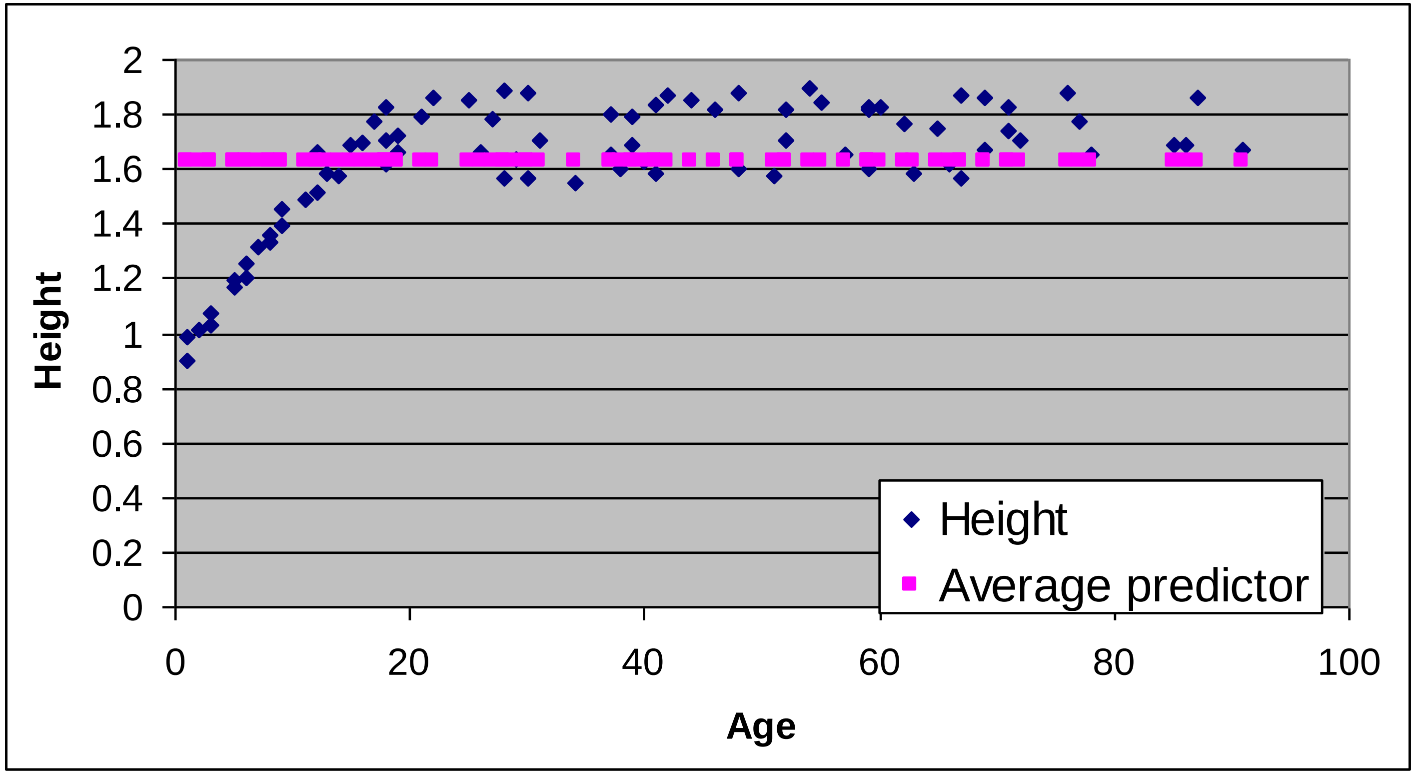
Age	Height
3	1.03
5	1.19
6	1.26
9	1.39
15	1.69
19	1.67
22	1.86
25	1.85
41	1.59
48	1.60
54	1.90
71	1.82
...	...

Test set

Age	Height
2	0.85
10	1.4
35	1.7
70	1.6

Baseline numeric predictor

- Average of the target variable



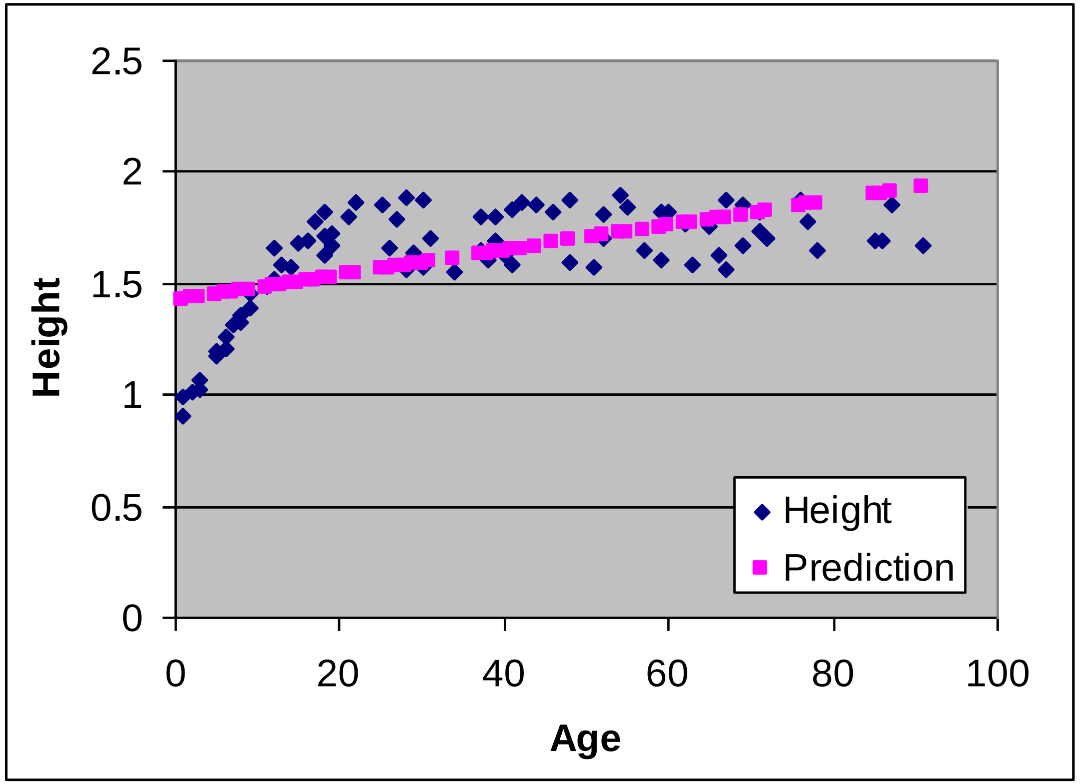
Baseline predictor: prediction

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

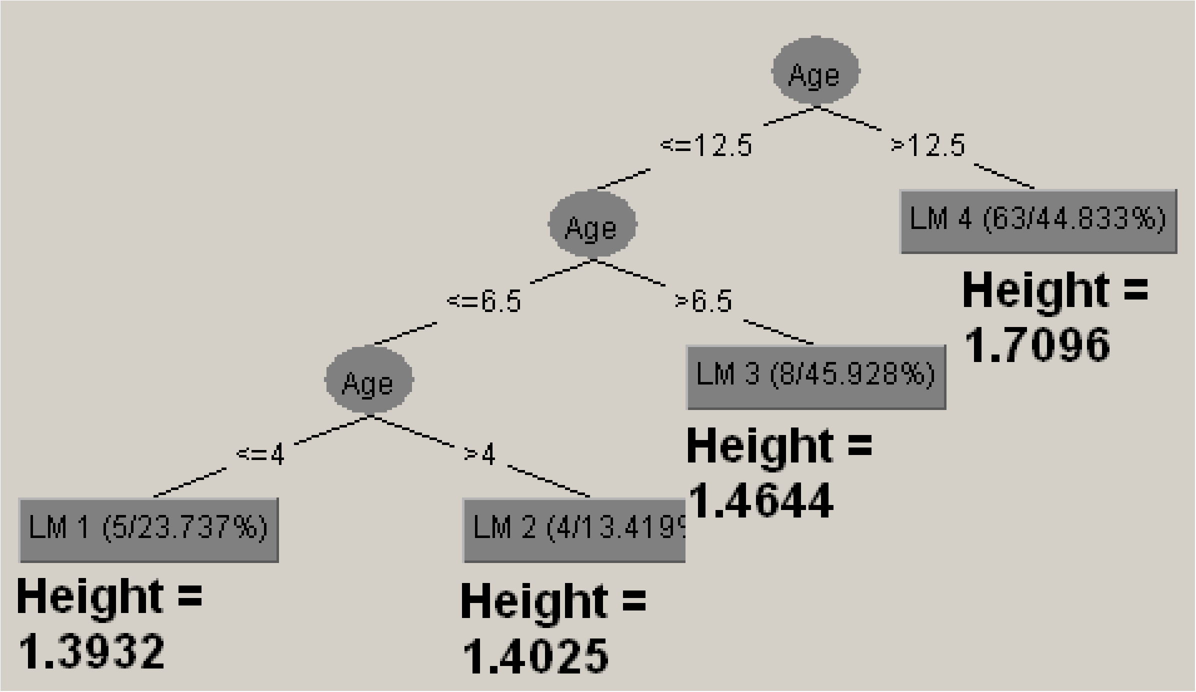


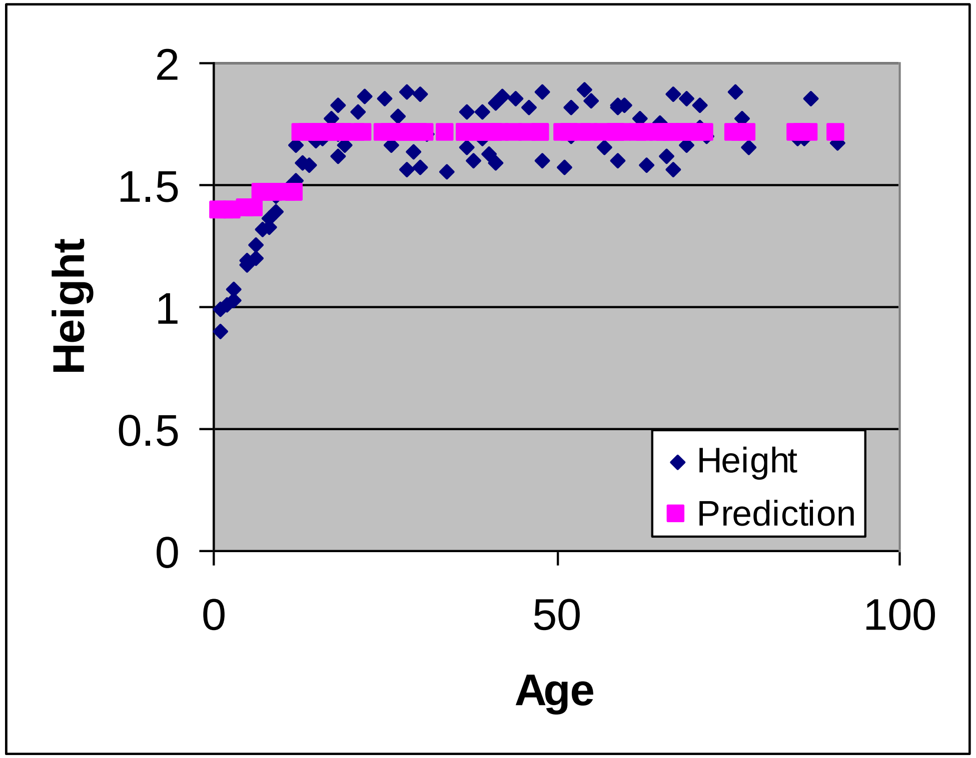
Linear Regression: prediction

Height = 0.0056 * Age + 1.4181

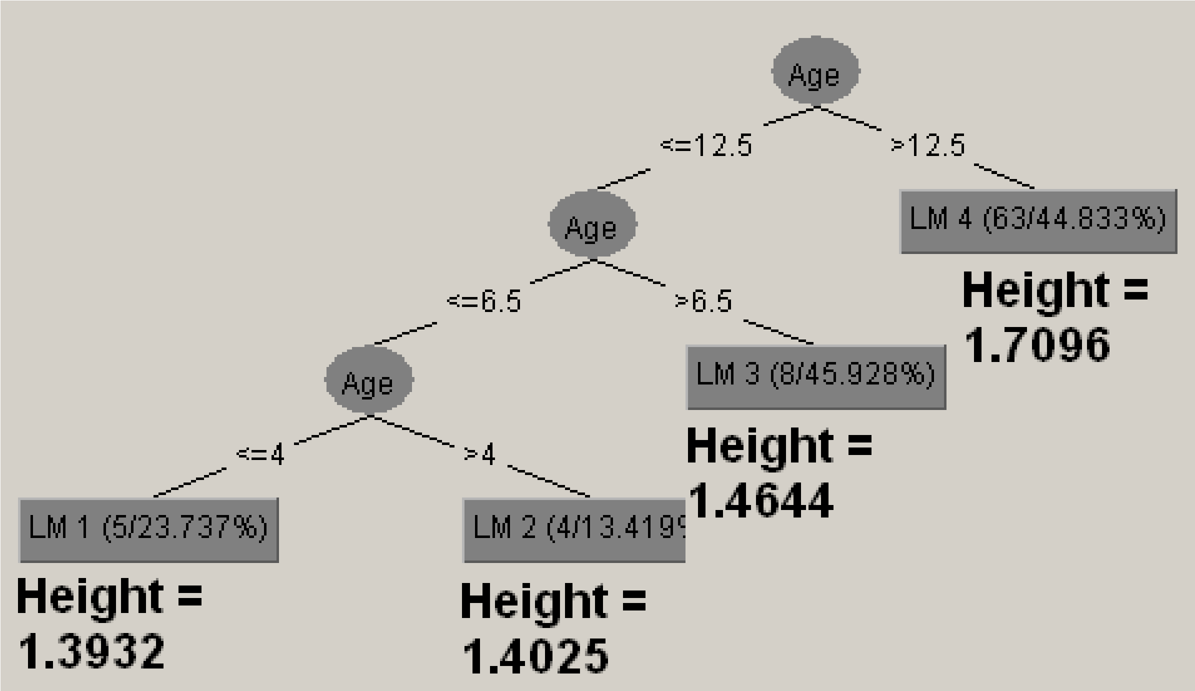
Age	Height	Linear regression
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Regression tree



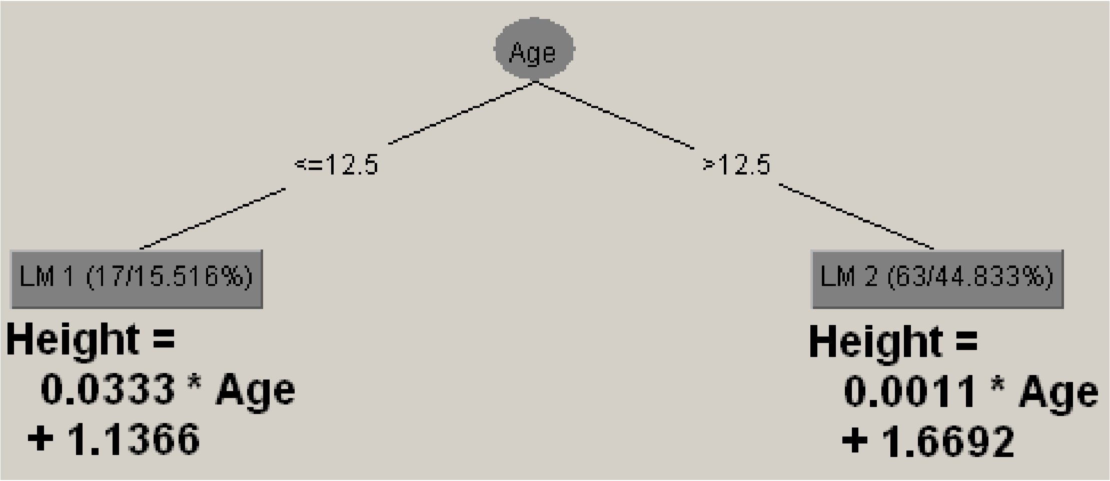


Regression tree: prediction

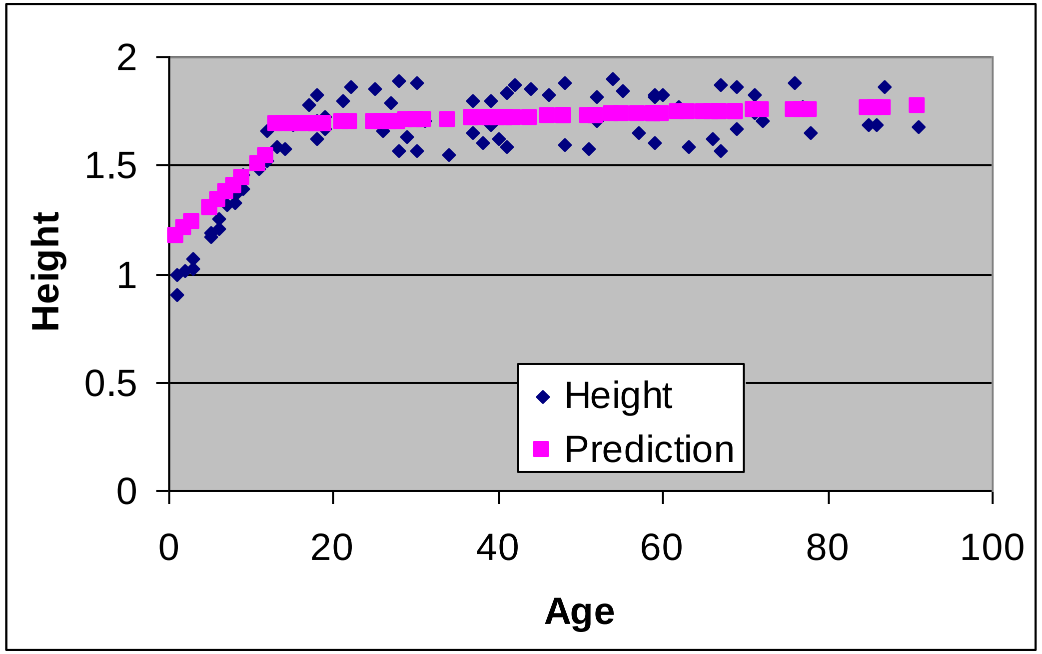


Age	Height	Regression tree
2	0.85	
10	1.4	
35	1.7	
70	1.6	

Model tree

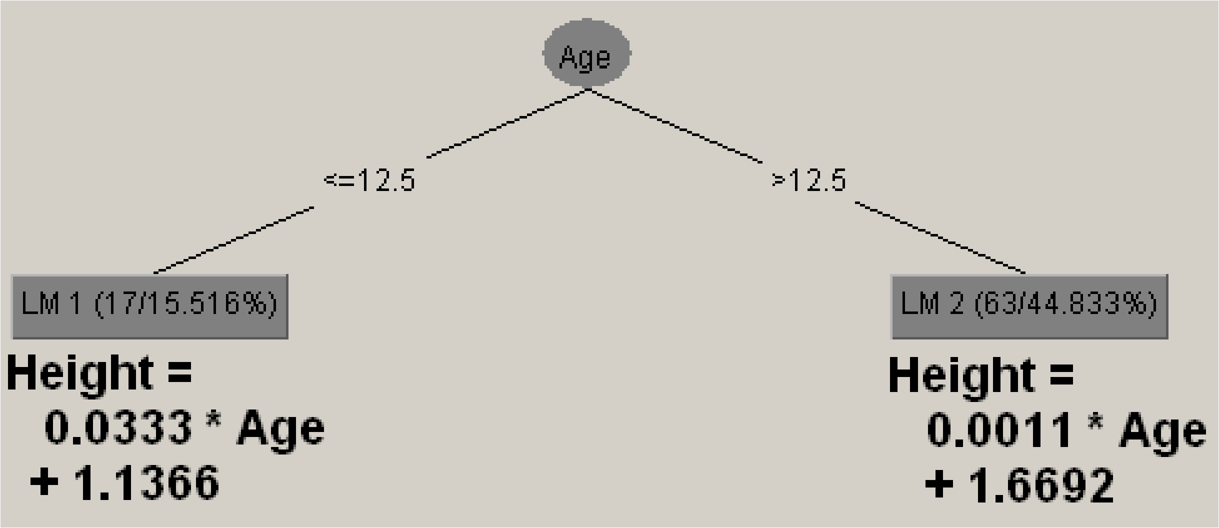


Age
<=12.5
>12.5
LM 1 (17/15.516%)
Height =
0.0333 * Age
+ 1.1366
LM 2 (63/44.833%)
Height =
0.0011 * Age
+ 1.6692



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

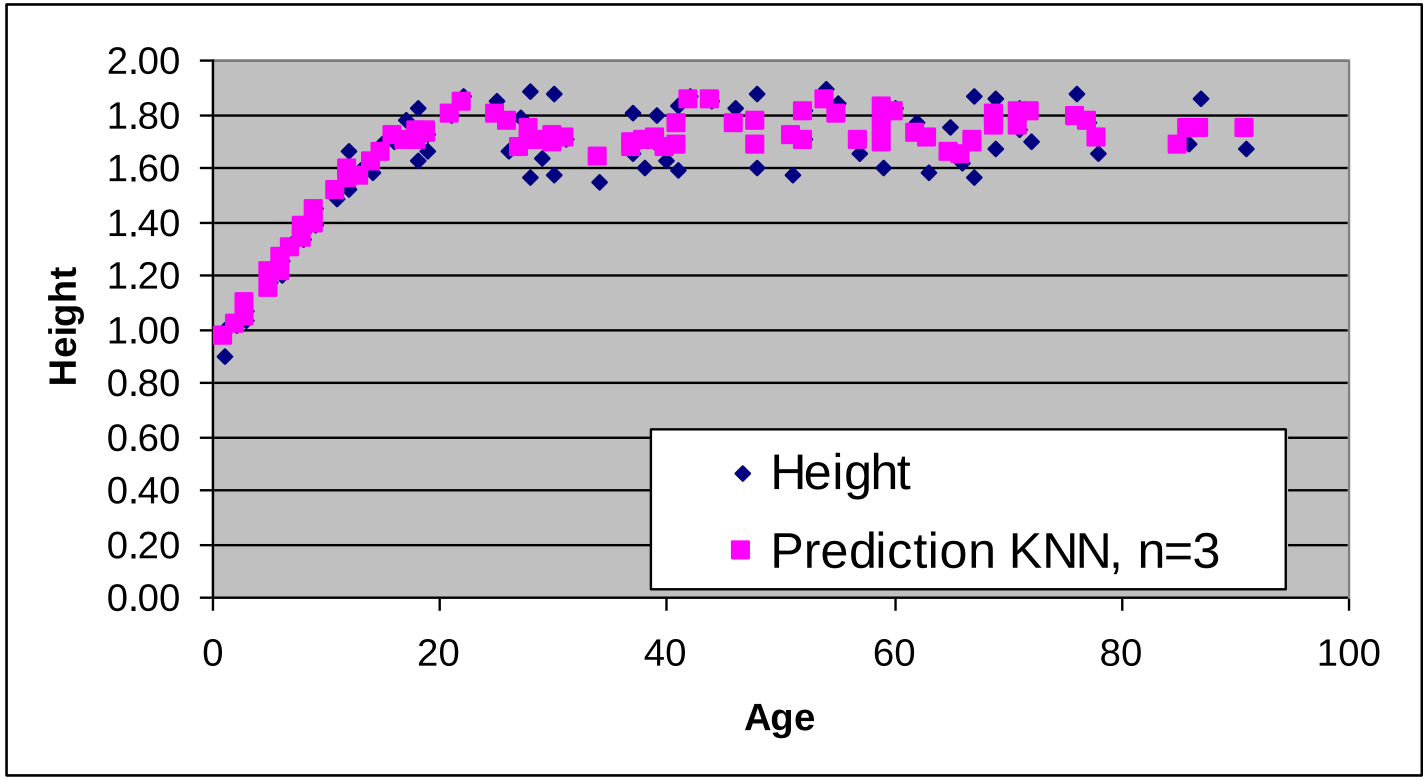
Model tree: prediction



Age	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	

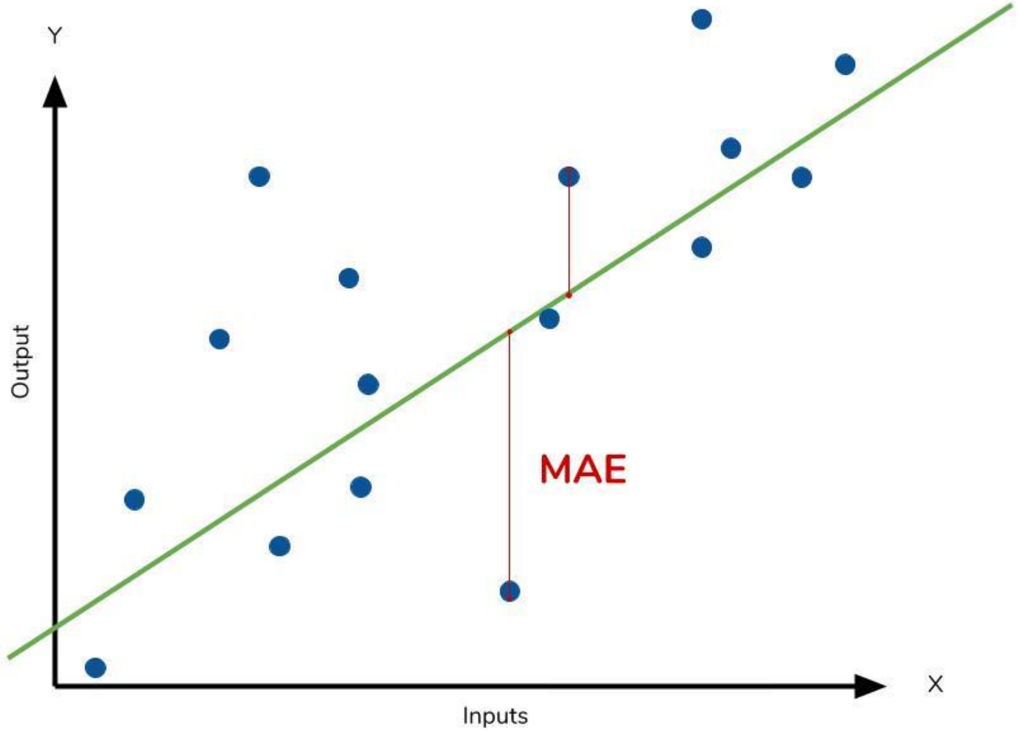
KNN video

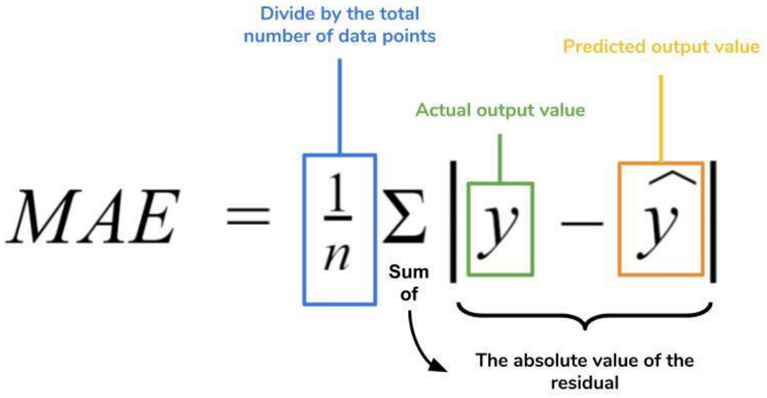
- http://videolectures.net/aaai07_bosch_knnc

Which predictor is the best?

Age	Height	Baseline	Linear regression	Regression 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

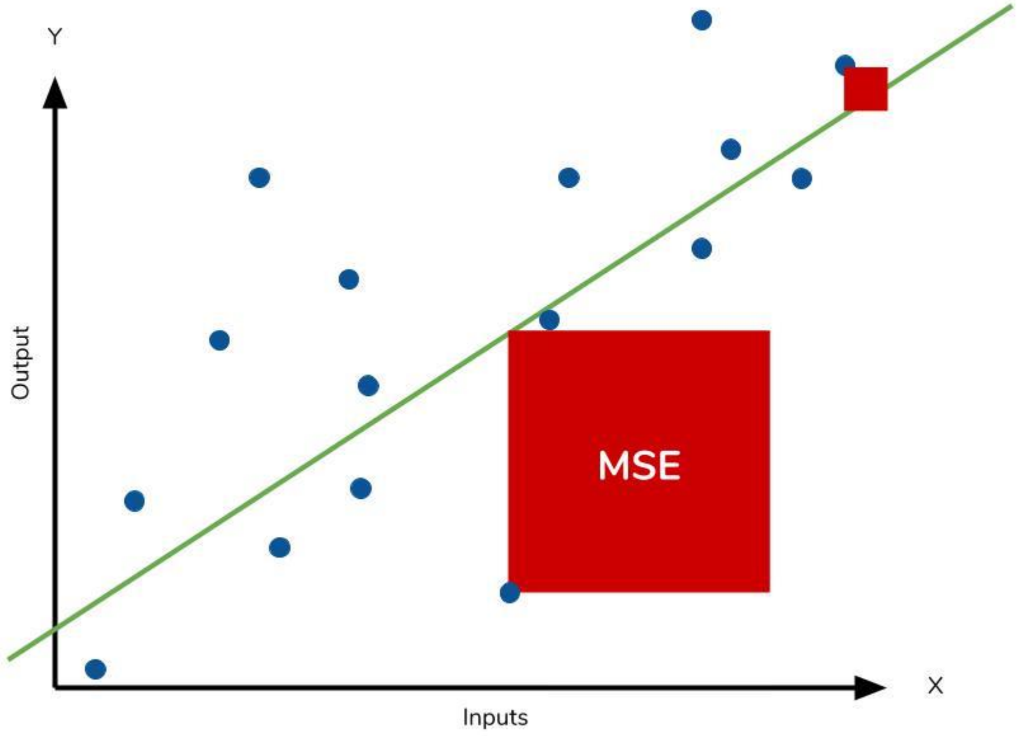
MAE: Mean absolute error





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

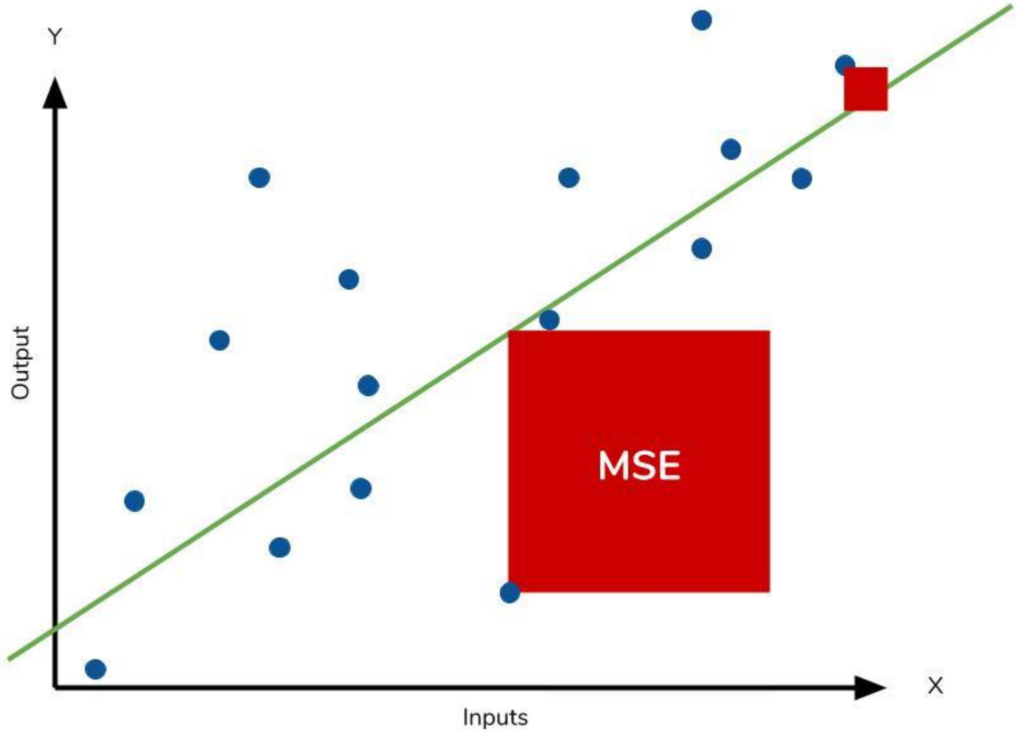
MSE: Mean squared error



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

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

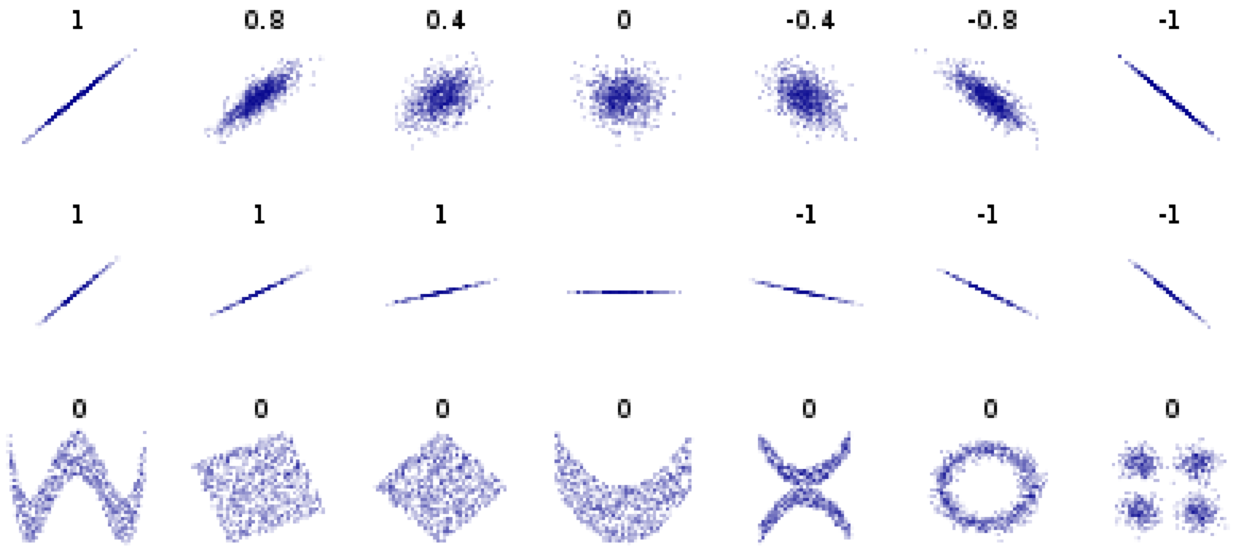


$$RMSE = \sqrt{MSE}$$

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

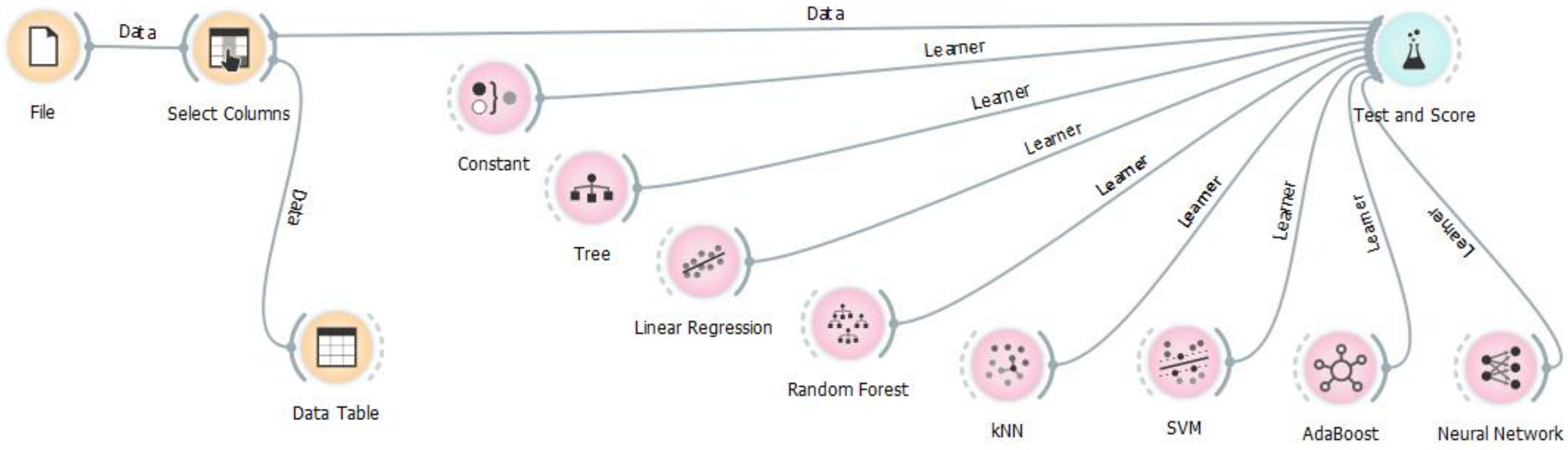
- Pearson correlation coefficient is a statistical formula that measures the strength between variables and relationships.



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

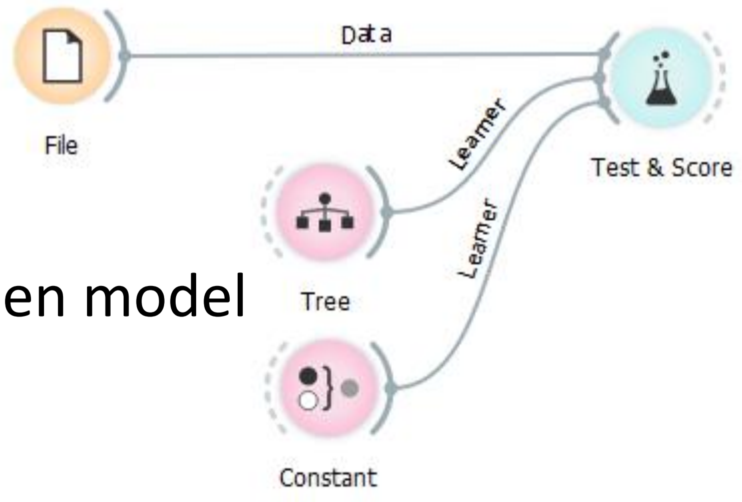
Homework

- Read

Loh, Wei-Yin. "Classification and regression trees." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1.1 (2011): 14-23.
https://onlinelibrary.wiley.com/doi/full/10.1002/widm.8

- Compare decision and regression trees.
- Rules of thumb when choosing the k parameter of KNN.

Homework



- Use Orange and a calculator to compute RRSE for a chosen model
- Data: regressionAgeHeight.csv

- RRSE = root relative squared error
  - Nominator: sum of squared differences between the actual and the expected values
  - Denominator: sum od squared errors

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

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)
  - Hint: If we divide both the nominator and the denominator by n we get RSE of the model and const model.