Norges miljø- og biovitenskapelige universitet

Institutt for datavitenskap

Digitalisering på Ås

DAT121
Introduction to data science

3 Regression basics

3.1 Supervised learning
3.2 Regression using statsmodels
3.3 Validation and testing

Schedule for 21st, 22nd, and 23rd August

Norwegian University of Life Sciences

Monday, 21st August 2023

9.15 – 10.00 Q&A session
10.15 – 11.00 first lecture on regression
11.15 – 12.00 discussion and problem solving

13.15 – 15.00 project work and tutorial

Tuesday, 22nd August 2023

10.15 – 12.00 scheduling of group sessions and of the final presentations

13.15 – 15.00 project work and tutorial

Wednesday, 23rd August 2023

10.15 – 11.00 second lecture on regression
11.15 – 12.00 **interest group sessions**

13.15 – 15.00 project work and tutorial

Reasoning underlying modelling

Norwegian University of Life Sciences

Common aims in modelling are for a model (e.g., an objective function) to be

- **quantitatively** accurate, both for
  - descriptions, *i.e.*, it should reproduce the known data correctly,
  - predictions, *e.g.*, for interpolation and extrapolation from data.
- **qualitatively** accurate, *i.e.*, it should correctly reflect *the way* in which multiple variables relate to each other.

These expectations very roughly relate to the two main modes of reasoning:

- **inductive** reasoning, where conclusions are drawn from patterns in data sets or statistics over data: This is what we here mean by "learning."
- **deductive** reasoning, also just "reasoning," where a premise (logically, mathematically) implies the conclusion, which is thus rigorously proven.

Reasoning underlying modelling

Common aims in modelling are for a model (e.g., an objective function) to be

- **quantitatively** accurate, both for
  - descriptions, *i.e.*, it should reproduce the known data correctly,
  - predictions, *e.g.*, for interpolation and extrapolation from data.

Qualitative accuracy relies on theories, quantitative accuracy on empirical data.

These expectations very roughly relate to the two main modes of reasoning:

- **inductive** reasoning, where conclusions are drawn from patterns in data sets or statistics over data: This is what we here mean by "learning."

Deductive reasoning relies on theories, learning relies on empirical data.

3 Regression basics

3.1 Supervised learning

Classification of machine learning methods

Categorization of learning methods by the *mode of human-digital interaction*:

- **Supervised learning**, where an agent obtains input-output pairs directly or indirectly from its percepts; *e.g.*, lists $\mathbf{x}$ and $\mathbf{y}$ are taken from sensory input, and a model $f(\mathbf{x}) = \mathbf{y}_{\text{model}}$ is constructed, aiming toward $\mathbf{y}_{\text{model}} = \mathbf{y}$. The model function is not arbitrary, but based on a priori **hypotheses**.

supervised learning

hypothesis

The **model quality** can be assessed by **validation and testing**, *i.e.*, by evaluating how well the model predicts data on which it has not been trained.

Classification of machine learning methods

Categorization of learning methods by the *mode of human-digital interaction*:

- **Supervised learning**, where an agent obtains input-output pairs directly or indirectly from its percepts; *e.g.*, lists $\mathbf{x}$ and $\mathbf{y}$ are taken from sensory input, and a model $f(\mathbf{x}) = \mathbf{y}_{\text{model}}$ is constructed, aiming toward $\mathbf{y}_{\text{model}} = \mathbf{y}$. The model function is not arbitrary, but based on a priori **hypotheses**.

- **Unsupervised learning**, where lists of variable values $\mathbf{x}_0, \ldots, \mathbf{x}_n$ are given to the agent/algorithm without any a priori hypotheses. It is up to the agent/algorithm to detect any patterns in the data set autonomously.

- **Reinforcement learning**, like the above, but with feedback on the model quality provided to the agent at each iteration.

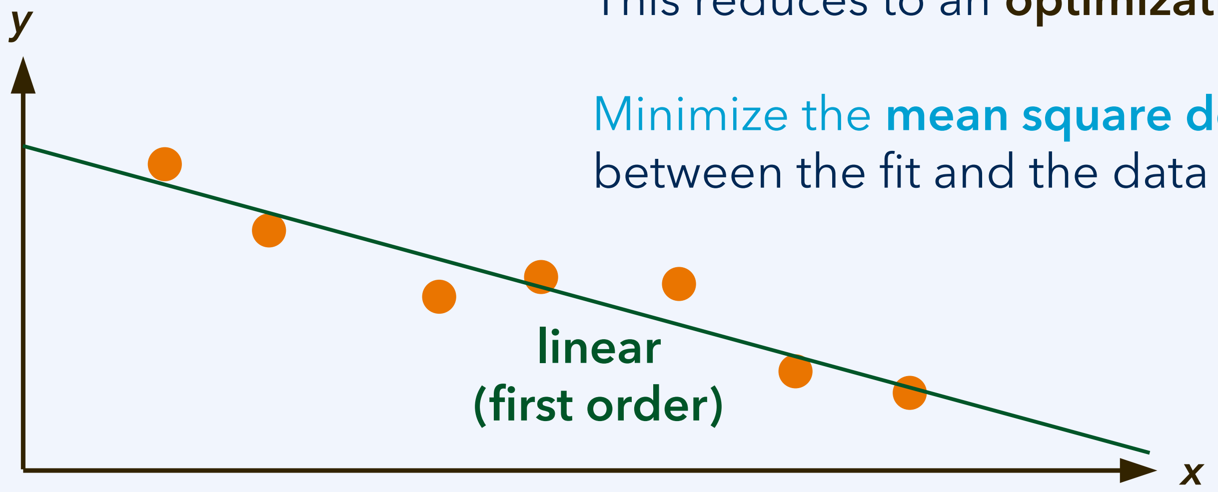
It is possible to combine these approaches, *e.g.*, by providing some a priori hypotheses about how the world functions, but not enough for a complete model.

Learning from data by regression

regression analysis

Data are typically affected by noise, random error, fluctuations, and similar phenomena that obscure to what extent variables are related to each other.

Regression analysis can help recover the **correlations between variables**.



This reduces to an **optimization problem**:

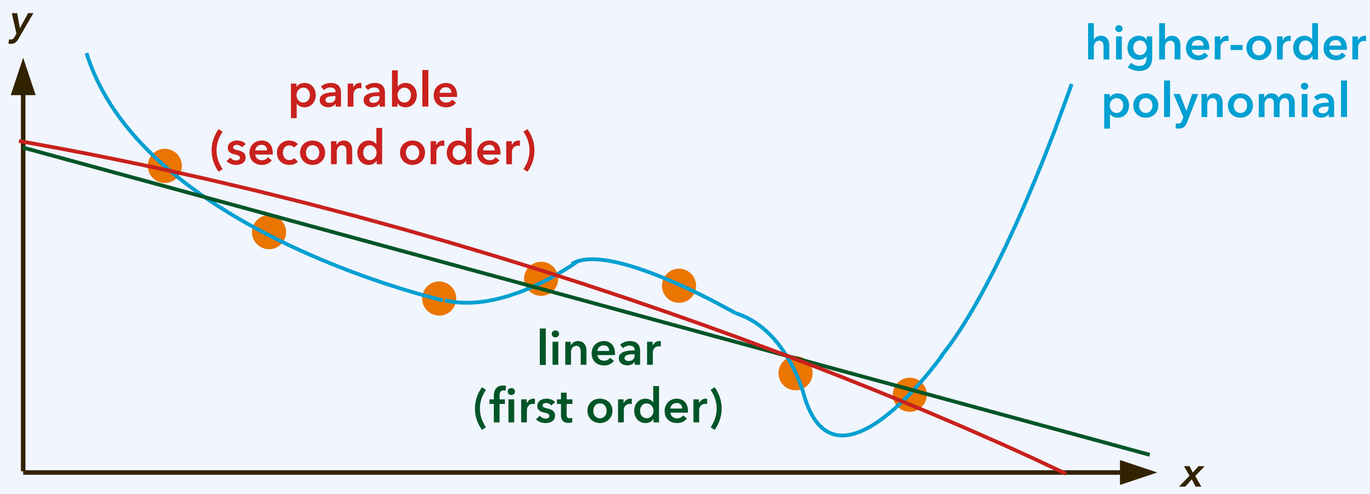
Minimize the **mean square deviation** between the fit and the data points.

This is also called an **ordinary least squares (OLS)** fit of a line to a data set.

Learning from data by regression

The **number of adjustable parameters** needs to reflect the amount of available information (not data, but data minus noise) and the complexity of the modelling problem. Von Neumann: "With four parameters I can fit an elephant[1]."

If this rule is disregarded, it leads to **overfitting**: Predictions become worse.

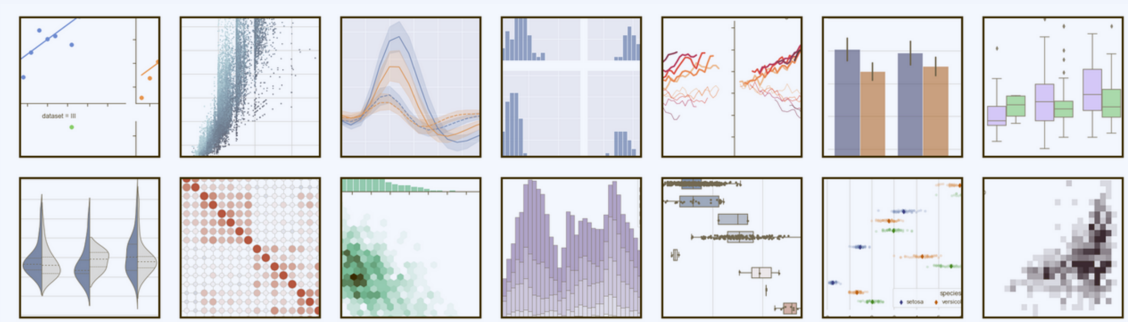


In supervised learning, the user specifies the type of model (*i.e.*, the **hypothesis**).

[1] J. Mayer *et al.*, *Am. J. Phys.* 78(6), 648–649, **2010**, actually draw such an elephant.

Regression and visualization using seaborn

We have used seaborn before, to visualize performance measurements. Functionalities of the **matplotlib** and **seaborn** libraries are presented in *Python for Data Analysis*, Chapter 9. There are many examples on the seaborn website:



Gallery of seaborn examples:

seaborn.pydata.org/examples/

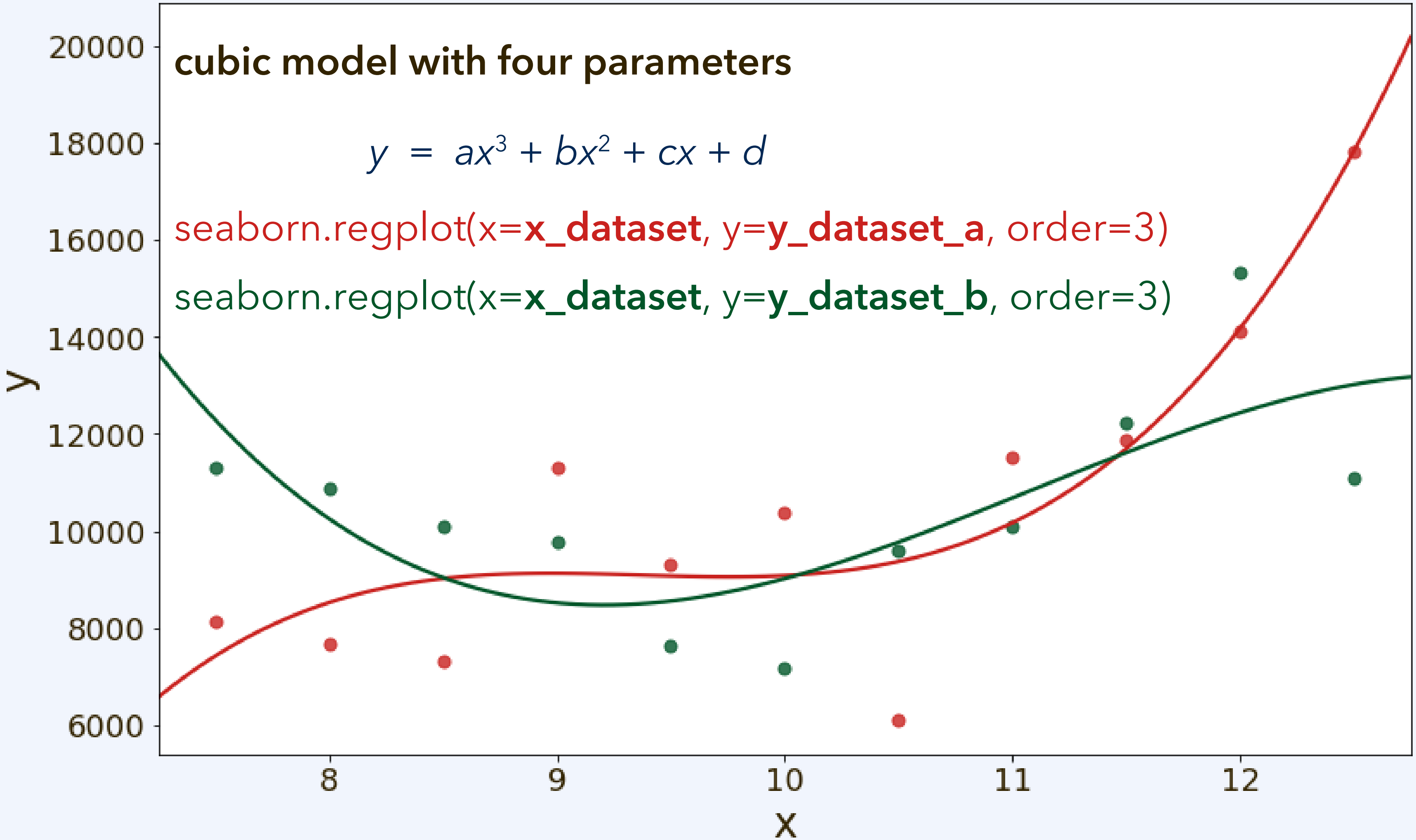
Regression analysis can help recover the **correlations between variables**. As an example, we consider two data sets, each generated by one of the following functions and affected by substantial noise:

$$f_a(x) = x^3 - 10\,x^2 + 1000\,x$$

$$f_b(x) = 10\,000$$

The regression can be done using seaborn, but only visually!
It is unfortunately *impossible to export the coefficients from the regression*.

Regression and visualization using seaborn



Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap

Digitalisering på Ås

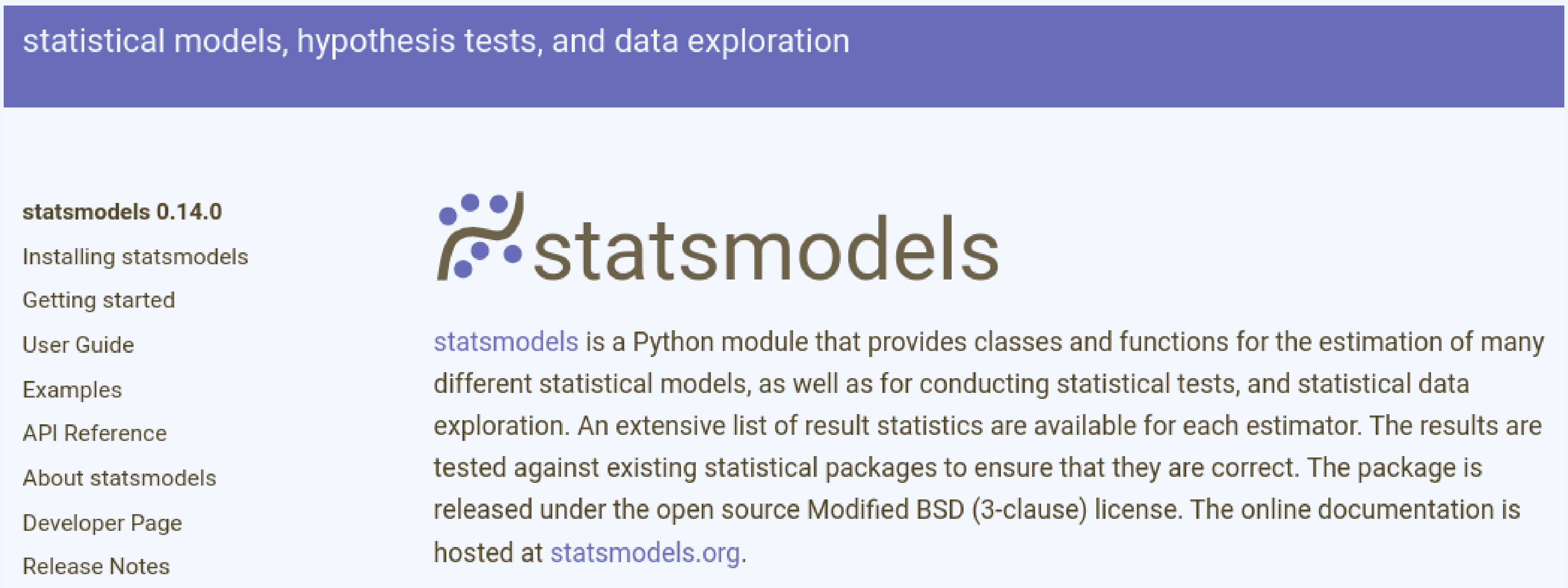
3 Regression basics

3.1 Supervised learning

3.2 Regression with statsmodels

The statsmodels library

We are interested in regression analysis not only as a visual tool. The library **statsmodels** (https://www.statsmodels.org/) is more suitable for this purpose.


statistical models, hypothesis tests, and data exploration

statsmodels 0.14.0

Installing statsmodels

Getting started

User Guide

Examples

API Reference

About statsmodels

Developer Page

Release Notes

statsmodels

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at statsmodels.org.

Chapter 10 of the *Python for Data Analysis* book discusses statsmodels, among other tools that can be used to analyse and aggregate data.

Linear regression using statsmodels

```python
import statsmodels.api as sm

x_array = sm.add_constant(np.asarray(x_dataset))
linear_fit_a = sm.OLS(np.asarray(y_dataset_a), x_array).fit()

print("Fit a):\n", linear_fit_a.summary())
```

```
Fit a):
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.574
Model:                            OLS   Adj. R-squared:                  0.526
Method:                 Least Squares   F-statistic:                     12.11
Date:                Mon, 29 Nov 2021   Prob (F-statistic):            0.00693
Time:                        15:11:38   Log-Likelihood:                -99.816
No. Observations:                  11   AIC:                             203.6
Df Residuals:                       9   BIC:                             204.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -4997.9028   4507.307     -1.109      0.296   -1.52e+04    5198.333
x1          1549.5537    445.200      3.481      0.007     542.441    2556.666
==============================================================================
Omnibus:                        5.158   Durbin-Watson:                   1.606
Prob(Omnibus):                  0.076   Jarque-Bera (JB):                1.722
Skew:                          -0.797   Prob(JB):                        0.423
Kurtosis:                       4.103   Cond. No.                         65.4
==============================================================================
```

$y = 1550\,x - 5000$

if the variables are independent, there is a **0.7%** probability of artificially creating (at least) such a strong correlation by chance

95% probability that the linear coefficient is between 542 and 2560

Linear regression using statsmodels

Compare data set b), with no actual underlying correlation between x and y.

```python
linear_fit_b = sm.OLS(np.asarray(y_dataset_b), x_array).fit()
print("Fit b):\n", linear_fit_b.summary())
```

```
Fit b):
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.124
Model:                            OLS   Adj. R-squared:                  0.026
Method:                 Least Squares   F-statistic:                     1.270
Date:                Mon, 29 Nov 2021   Prob (F-statistic):              0.289
Time:                        15:10:39   Log-Likelihood:                -99.022
No. Observations:                  11   AIC:                             202.0
Df Residuals:                       9   BIC:                             202.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5801.8258   4193.444      1.384      0.200   -3684.404    1.53e+04
x1           466.8272    414.199      1.127      0.289    -470.156    1403.810
==============================================================================
Omnibus:                        0.210   Durbin-Watson:                   1.045
Prob(Omnibus):                  0.900   Jarque-Bera (JB):                0.143
Skew:                           0.182   Prob(JB):                        0.931
Kurtosis:                       2.577   Cond. No.                         65.4
==============================================================================
```

$y = 467\,x + 5800$

if the variables are independent, there is a **28.9%** probability of artificially creating (at least) such a strong correlation by chance

95% probability that the linear coefficient is between –470 and +1400

The *p* value

Compare data set b), with no actual underlying correlation between x and y.

This quantity is called the "*p* value."

It indicates the probability of the same or a stonger apparent correlation between two variables (here, x and y), assuming that the null hypothesis is true.

if the variables are independent, there is a **28.9%** probability of artificially creating (at least) such a strong correlation by chance

Null hypothesis: There is no actual underlying correlation between x and y. Any appearance of such a correlation is due to chance.

By convention, correlations are typically seen as statistically insignificant if $p > 5\%$.

95% probability that the linear coefficient is between –470 and +1400

p value

Spurious correlations

https://tylervigen.com/spurious-correlations

There is always the risk of **statistical fallacies** when we overly rely on the *p* value. Assume we are particularly rigorous and require the *p* value to be lower than a level of significance of 0.01.

But we examine data for very many correlations.

Now we instruct our high-throughput data analysis system to evaluate:

- Is there a correlation between avocado consumption and cancer? No.
- … between liver disease and number of pets in the household? No.
  (… about a hundred more questions …)
- … between coronary disease and consumption of elk meat? Yes, $p < 0.01$.

Next month in an illustrated paper: "Eat elk meat to avoid heart attacks! A scientific study has proven …"

Nonlinear regression using statsmodels

Polynomial regression using a **statsmodels** OLS linear regression fit:

First create a matrix (2D numpy array) of 1, x, x^2, …, x^k values:

[	[1	7.5	56.25	421.875]
	[1	8.0	64.00	512.000]
	[1	8.5	72.25	614.125]
	[1	9.0	81.00	729.000]
	[1	9.5	90.25	857.375]
	[1	10.0	100.00	1000.000]
	[1	10.5	110.25	1157.625]
	[1	11.0	121.00	1331.000]
	[1	11.5	132.25	1520.875]
	[1	12.0	144.00	1728.000]
	[1	12.5	156.25	1953.125]]

x_expansion_2d_array = np.asarray(\
[[1, x, x*x, x*x*x] for x in **x_dataset**])

Then pass on to the OLS fit:

sm.OLS(np.asarray(**y_dataset_a**), \
x_expansion_2d_array).fit()

Remark: x, x^2, *etc.*, are not independent variables. The regression analysis (*e.g.*, *p* values) from statsmodels is affected by the correlation between them.

Nonlinear regression using statsmodels

significance level for the overall model (as opposed to only noise) based on *F* test

```
Fit a):
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.799
Model:                            OLS   Adj. R-squared:                  0.713
Method:                 Least Squares   F-statistic:                     9.265
Date:                Mon, 06 Dec 2021   Prob (F-statistic):            0.00781
Time:                        10:55:39   Log-Likelihood:                -95.687
No. Observations:                  11   AIC:                             199.4
Df Residuals:                       7   BIC:                             201.0
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -2.342e+05   1.79e+05     -1.311      0.231   -6.56e+05    1.88e+05
x1          7.821e+04   5.49e+04      1.424      0.198   -5.17e+04    2.08e+05
x2         -8367.3711   5558.872     -1.505      0.176   -2.15e+04    4777.273
x3           297.8664    185.111      1.609      0.152    -139.851     735.584
==============================================================================
Omnibus:                        2.762   Durbin-Watson:                   2.877
Prob(Omnibus):                  0.251   Jarque-Bera (JB):                1.069
Skew:                          -0.761   Prob(JB):                        0.586
Kurtosis:                       3.123   Cond. No.                     4.04e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.04e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

$y = 298\,x^3 - 8{,}370\,x^2 + 78{,}200\,x - 234{,}000$

p values for the **individual** coefficients

warning about correlation between the x, x^2, and x^3 values

Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap

Digitalisering på Ås

3 Regression basics

3.1 Supervised learning

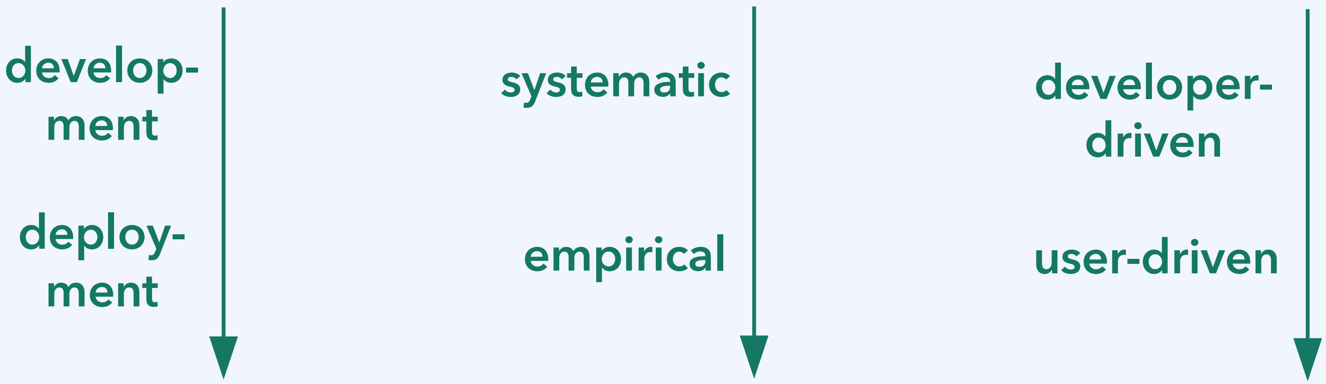
3.2 Regression with statsmodels

3.3 Validation and testing

Validation and testing in software engineering

Verification: Proof that the developed product complies with its specification.

– Where possible, provide a **rigorous logical/mathematical proof**; alternatively, provide documents following agreed standards/procedures.



Testing: Use-case driven evaluation of the final (or alpha, or beta) product.

– The considered **use cases** should be **representative**.
– They should be as unrelated as possible to any concrete scenarios considered during development.
– Ideally, **conducted by prospective users**; if unavailable, “play the user.”

Validation and testing in software engineering

Verification: Proof that the developed product complies with its specification.

- Where possible, provide a **rigorous logical/mathematical proof**; alternatively, provide documents following agreed standards/procedures.

Validation: Empirical evaluation to what extent user the requirements are met.

- All **requirements** need to be covered and demonstrated at least once.
- Ideally, **requirements** are not identical with the specification. They should be user-oriented; *e.g.*, epics and user stories in a requirements analysis from agile software engineering. **Feedback from users** is needed.

Testing: Use-case driven evaluation of the final (or alpha, or beta) product.

- The considered **use cases** should be **representative**.
- They should be **as unrelated as possible to** any concrete scenarios considered during development, including **the validation process**.
- Ideally, **conducted by prospective users**; if unavailable, "play the user."

Validation and testing in modelling

validation and testing

It is good practice to split the available data into three portions:

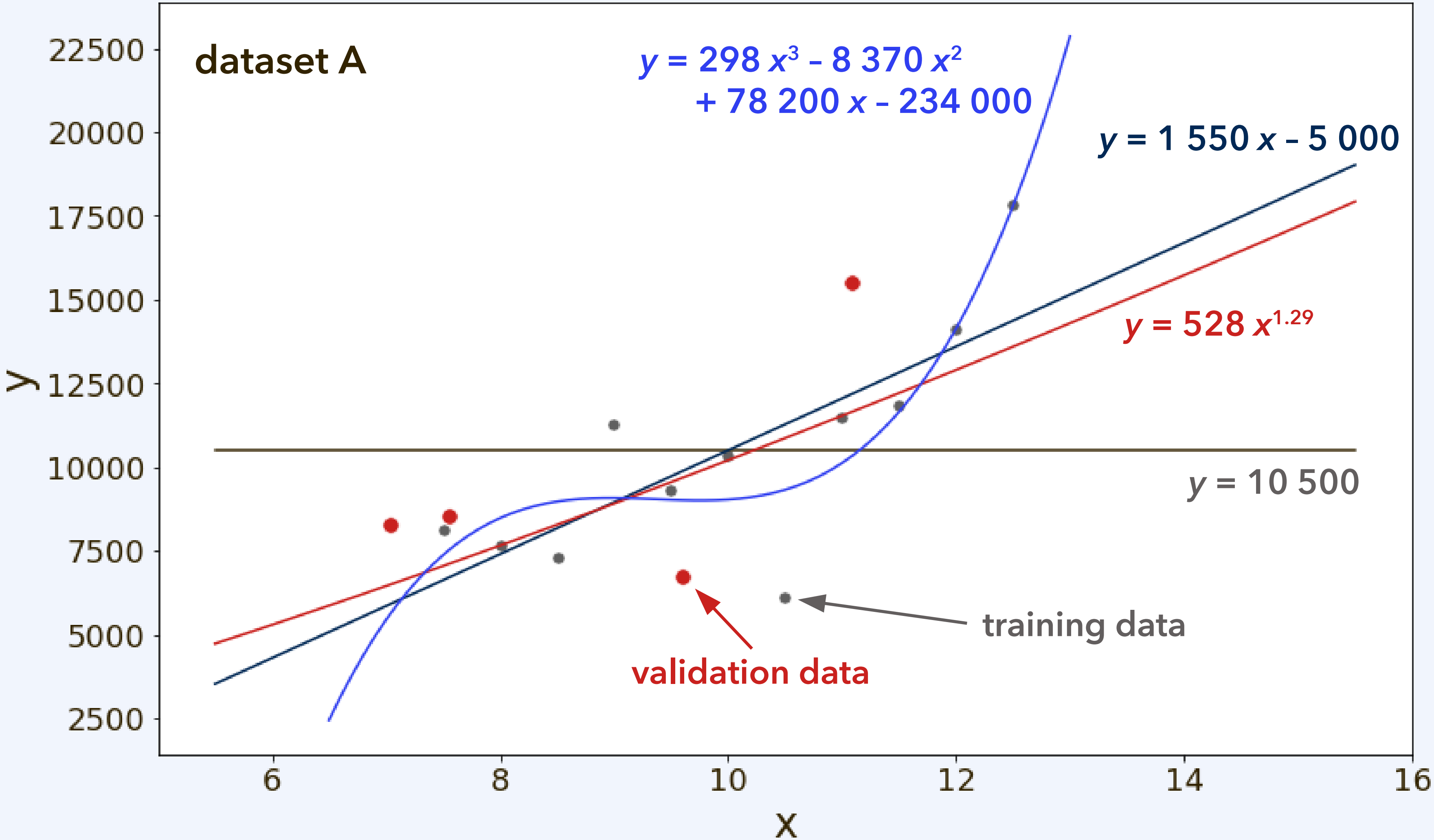
- **Training set:** Data that are used to compute the regression(s), or to construct model(s) by another method based on learning from data. This should be the largest portion of the overall data set.
- **Validation set:** Data reserved for evaluating multiple candidate models. How well do the models predict data with which they were not trained?
- **Test set:** What accuracy does the selected model have for predictions?

This approach works best if:

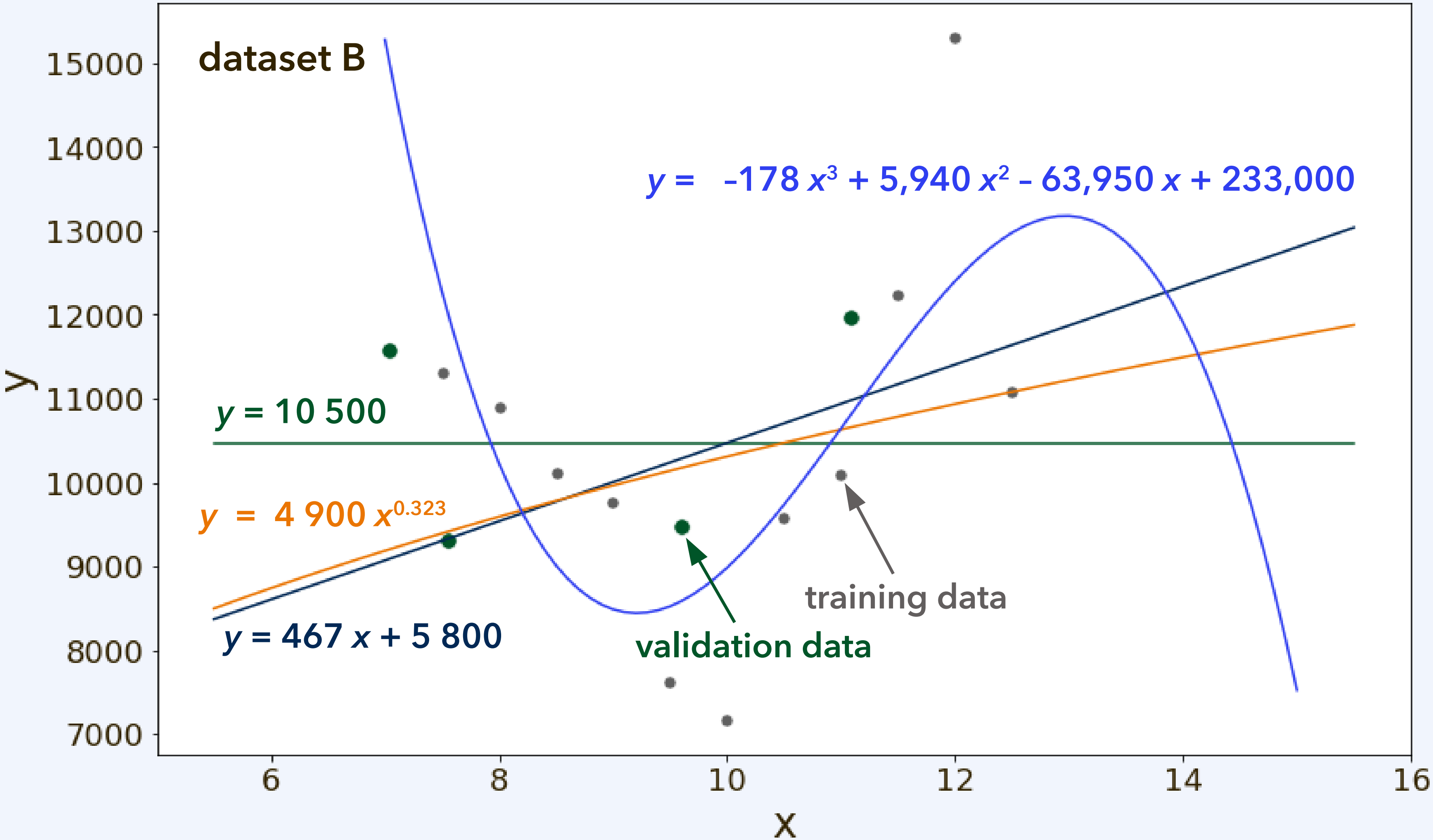
a) The training, validation, and test sets are equally representative of the phenomenon under investigation.

b) Except for this connection, they are as mutually independent as possible. (To ensure that this is really an independent validation/test).

Example: Model validation (data set A)



Example: Model validation (data set B)



Root mean square deviation from validation data

```
… from constant average:   3455.2
… from linear regression:  2725.4
… from bilog. regression:  2672.2
… from cubic regression:   3148.1
```

data set A

```
… from constant average:   1199.5
… from linear regression:  1392.3
… from bilog. regression:  1394.6
… from cubic regression:   2302.9
```

data set B

constant average value

Data set A: $y = 10{,}500$

Data set B: $\boldsymbol{y = 10{,}500}$ ← selected model

linear regression

Data set A: $y = 1550\,x - 5000$

Data set B: $y = 467\,x + 5800$

bilogarithmic regression

Data set A: selected model → $\boldsymbol{y = 528\,x^{1.29}}$

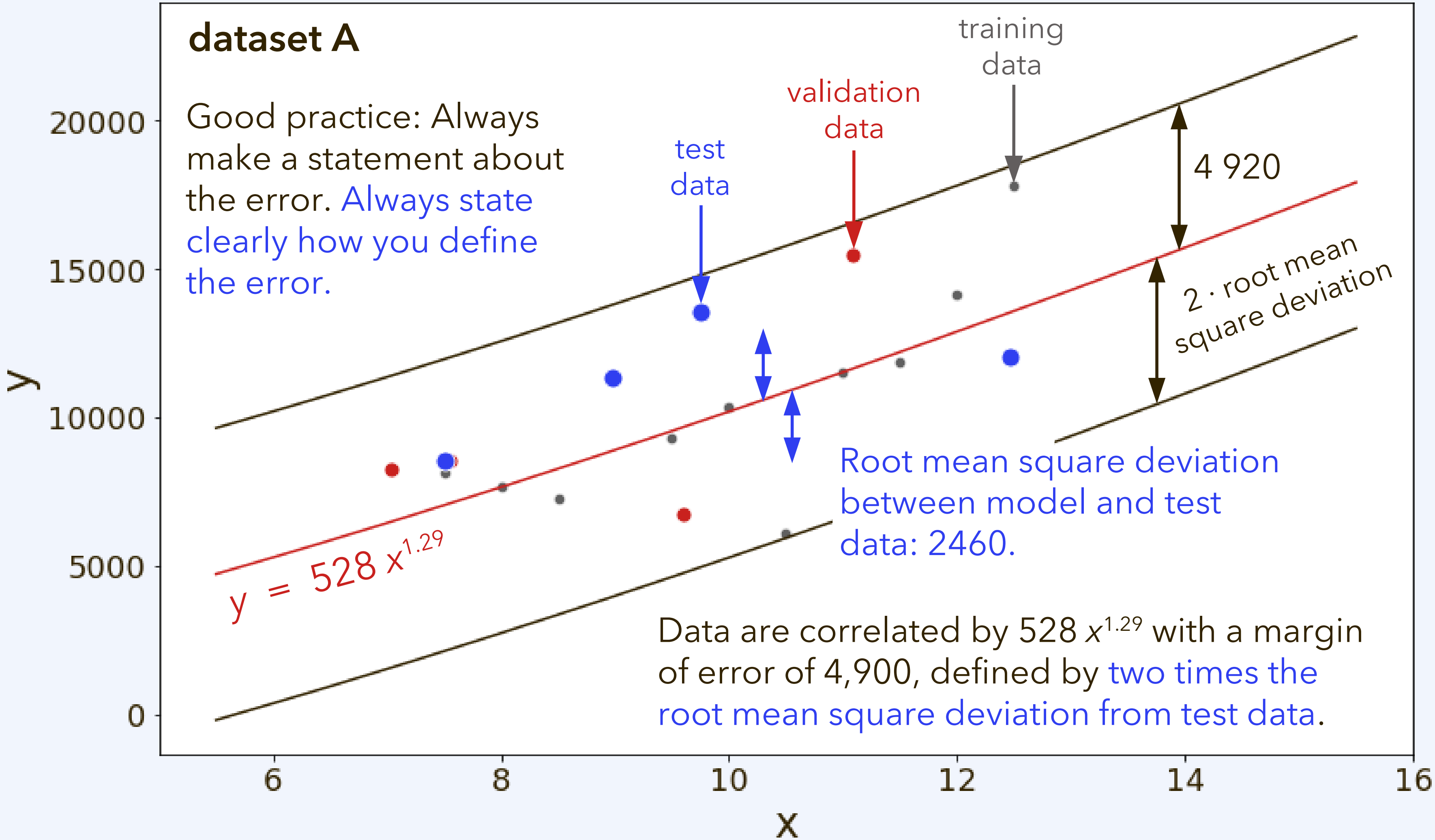
Data set B: $y = 4900\,x^{0.323}$

cubic regression

Data set A: $y = 298\,x^3 - 8{,}370\,x^2 + 78{,}200\,x - 234{,}000$

Data set B: $y = -178\,x^3 + 5{,}940\,x^2 - 63{,}950\,x + 233{,}000$

Model testing (example dataset A)



Noregs miljø- og biovitskaplege universitet

Institutt for datavitskap

Digitalisering på Ås

Conclusion

Glossary terms

Proposed glossary[1] terms:

- How do we best define them? Is the definition controversial?
- What is the best translation into Norwegian bokmål/nynorsk?
- Are there more key concepts that would require an agreed definition?

supervised learning

hypothesis

validation and testing

p value

regression analysis

[1] https://home.bawue.de/~horsch/teaching/dat121/glossary-en.html

Norges miljø- og biovitenskapelige universitet

Institutt for datavitenskap

Digitalisering på Ås

DAT121
Introduction to data science

3 Regression basics

3.1 Supervised learning

3.2 Regression using statsmodels

3.3 Validation and testing