PADERBORN UNIVERSITY

Linear Models

Explainable Artificial Intelligence

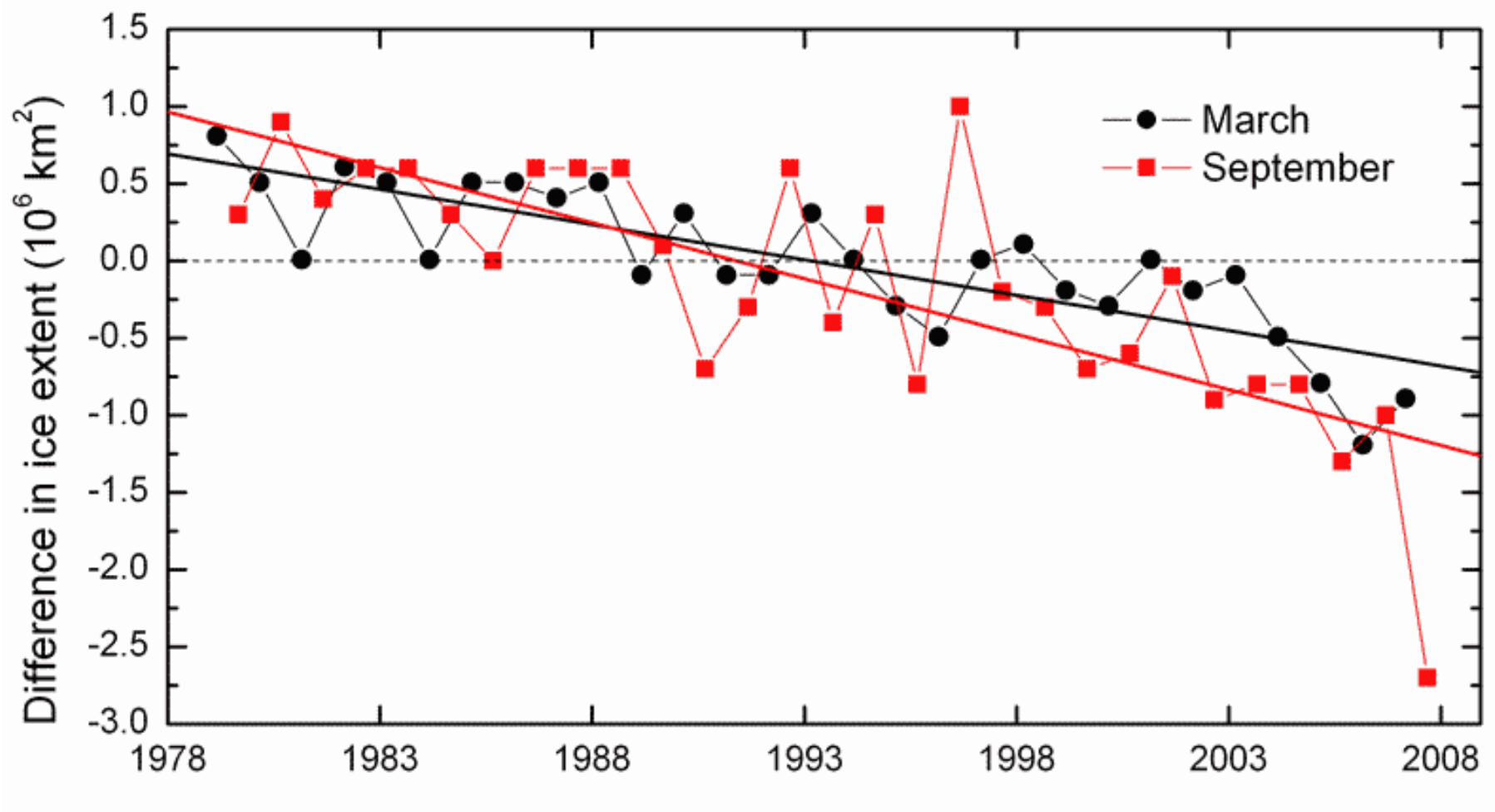
Dr. Stefan Heindorf

Outlook

- Linear regression
- Logistic regression

Linear regression

Example: arctic sea ice extent



NOAA - J. Richter-Menge, S. Nghiem, D. Perovich, I. Rigor (https://commons.wikimedia.org/wiki/File:Arctic_Sea_Ice_Extent,_March_&_September,_1979-2007.png), „Arctic Sea Ice Extent, March & September, 1979-2007“, marked as public domain

How does linear regression work?

Linear model

- Predict the target y as a weighted sum of the feature values $x_1, \ldots, x_p$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

$\hat{y}$	predicted target (should approximate the true target y)
β_0	intercept
β_j	learned feature weights or coefficients

- Linearity of the learned relationship makes the interpretation easy
- Linear regression models are widely used by statisticians, computer scientists and other people

Linear regression

Matrix notation

For a single datapoint $\mathbf{x}^{\mathbf{T}} = (1, x_1, \ldots, x_p)$ **:**

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \mathbf{x}^{\mathbf{T}}\boldsymbol{\beta}$$

For multiple datapoints $\mathbf{x}^{(1)^{\mathbf{T}}}, \ldots, \mathbf{x}^{(n)^{\boldsymbol{T}}}$**:**

$$\hat{\boldsymbol{y}} = \mathbf{X}\boldsymbol{\beta}$$

where

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(n)} \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)^{\mathbf{T}}} \\ \mathbf{x}^{(2)^{\mathbf{T}}} \\ \vdots \\ \mathbf{x}^{(n)^{\mathbf{T}}} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_p^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix}, \qquad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}$$

Notation:

- scalars are in *italics* (x_j)
- vectors use **bold lower case** letters ($\mathbf{x}$)
- matrices use **BOLD UPPER CASE** letter ($\mathbf{X}$)

Linear regression

Learning

- Predict the target y as a weighted sum of the feature values $\mathbf{x} = (1, x_1, \ldots, x_p)$

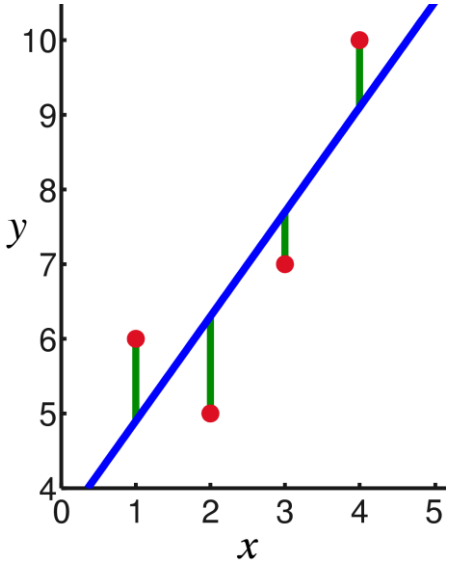
$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \mathbf{x}^{\mathbf{T}} \boldsymbol{\beta}$$

(where $x_0 := 1$)

- Given multiple datapoints $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ with true targets $y^{(1)}, \ldots, y^{(n)}$, the goal is to find the weights $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)$ that minimize the **squared** loss $\left(\hat{y}^{(i)} - y^{(i)}\right)^2$

$$\boldsymbol{\beta} = \arg \min_{\beta_0, \ldots, \beta_p} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

- Minimization problem can be solved exactly or approximately with various methods
  - Exact solution: see next slide
  - Backpropagation: see later lecture on neural networks



Linear regression: analytical solution

[cf., Hastie et al. 2019, Equations 2.3-2.6, 3.3-3.6, Kolter et al. 2020, Section 4.4]

Using the fact that $\|\mathbf{v}\|_2^2 = \mathbf{v}^\mathbf{T}\mathbf{v}$ for an arbitrary vector $\mathbf{v}$, the term

$$\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \sum_{i=1}^{n}\left(y^{(i)} - \mathbf{x}^{(i)\mathbf{T}}\boldsymbol{\beta}\right)^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

becomes minimal when its gradient (= "derivate") w.r.t. $\boldsymbol{\beta}$ becomes 0 (necessary condition):

$$-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

(for further details, see next slides)

If $\mathbf{X}^\mathbf{T}\mathbf{X}$ is nonsingular (=invertible), then the unique solution is given by

$$\boxed{\boldsymbol{\beta} = \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathbf{T}\mathbf{y}}$$

Note: To prove that the $\boldsymbol{\beta}$ obtained is indeed the local minimum, one needs to differentiate once more to obtain the Hessian matrix and show that it is positive definite (sufficient condition)

Linear regression: analytical solution

Excursus: Computation of gradient

$$\|\mathbf{y} - \mathbf{X\beta}\|_2^2 = (\mathbf{y} - \mathbf{X\beta})^T(\mathbf{y} - \mathbf{X\beta})$$

cost function

Transformation (see Kolter et al. 2020)

$$\begin{aligned}
&(\mathbf{y} - \mathbf{X\beta})^T(\mathbf{y} - \mathbf{X\beta}) \\
&= \left(\mathbf{y^T} - \boldsymbol{\beta}^\mathbf{T}\mathbf{X^T}\right)(\mathbf{y} - \mathbf{X\beta}) \\
&= \mathbf{y^T y} - \mathbf{y^T X\beta} - \boldsymbol{\beta}^\mathbf{T}\mathbf{X^T y} + \boldsymbol{\beta}^\mathbf{T}\mathbf{X^T X\beta} \\
&= \mathbf{y^T y} - 2\mathbf{y^T X\beta} + \boldsymbol{\beta}^\mathbf{T}\mathbf{X^T X\beta}
\end{aligned}$$

Applying the rules for gradients (see Kolter et al. 2020)

$$\begin{aligned}
&\nabla_{\boldsymbol{\beta}}(\mathbf{y^T y} - 2\mathbf{y^T X\beta} + \boldsymbol{\beta}^\mathbf{T}\mathbf{X^T X\beta}) \\
&= \nabla_{\boldsymbol{\beta}}\left(\mathbf{y^T y}\right) - \nabla_{\boldsymbol{\beta}}(2\mathbf{y^T X\beta}) + \nabla_{\boldsymbol{\beta}}(\boldsymbol{\beta}^\mathbf{T}\mathbf{X^T X\beta}) \\
&= \mathbf{0} - \left(2\mathbf{y^T X}\right)^\mathbf{T} + 2\mathbf{X^T X\beta} \\
&= -2\mathbf{X^T}(\mathbf{y} - \mathbf{X\beta})
\end{aligned}$$

same as closed form

Linear regression: analytical solution

Excursus: Computation of gradient

this checks that its minimum

Hessian ("gradient of the gradient")

$$\begin{aligned} & \nabla_{\beta}(-2\mathbf{X}^{\mathbf{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \\ &= \nabla_{\beta}\left(-2\mathbf{X}^{T}\boldsymbol{y}+2\mathbf{X}^{T}\mathbf{X}\boldsymbol{\beta}\right) \\ &= \mathbf{0}+2\mathbf{X}^{\mathbf{T}}\mathbf{X} \end{aligned}$$

Positive semi-definite

- **Definition:** A matrix $\mathbf{A}$ is positive semi-definite if for all vectors $\boldsymbol{z} \in \mathbb{R}^{\boldsymbol{n}} \backslash \{\mathbf{0}\}$:
$$\mathbf{z}^{\mathbf{T}}\mathbf{A}\mathbf{z} \geq \mathbf{0}$$
- **Here:** $\mathbf{z}^{\mathbf{T}}2\mathbf{X}^{\mathbf{T}}\mathbf{X}\mathbf{z} = 2(\mathbf{X}\mathbf{z})^{\mathbf{T}}\mathbf{X}\mathbf{z} = 2\|\mathbf{X}\mathbf{z}\|_2^2 \geq \mathbf{0}$

Positive definite

- **Definition:** A matrix $\mathbf{A}$ is positive definite if for all vectors $\mathbf{z} \in \mathbb{R}^{\boldsymbol{n}} \backslash \{\mathbf{0}\}$:
$$\mathbf{z}^{\mathbf{T}}\mathbf{A}\mathbf{z} > \mathbf{0}$$
- **Here:** $\mathbf{2}(\mathbf{X}\mathbf{z})^{\mathbf{T}}(\mathbf{X}\mathbf{z}) > \mathbf{0}$ (because $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ is nonsingular, $\mathbf{X}$ has full rank)

Bike rental

Predict number of rented bikes on given day

What is this?
How is this computed?

	Weight	SE	|t|
(Intercept)	2742.4	339.0	8.1
seasonSPRING	454.8	169.9	2.7
seasonSUMMER	263.3	218.7	1.2
seasonFALL	636.9	152.7	4.2
holidayY	-441.1	239.0	1.8
workdayY	242.0	101.9	2.4
weatherMISTY	-345.3	120.2	2.9
weatherBAD	-1804.1	306.0	5.9
temp	51.0	10.3	5.0
hum	-21.7	4.5	4.8
windspeed	-46.5	9.5	4.9
cnt_2d_bfr	0.6	0.0	19.3

Linear regression Estimating variance and standard error of weights

[Hastie, 2009 pp. 44 – 49; Hayashi 2020 p. 27 – 29; Sheather 2009 chapter 5, p. 134]

- Feature weights $\hat{\beta}_j$ are estimated
- Assumption: target outcome Y given the features $\mathbf{x}$ follows a normal distribution

$$\mathbf{Y} = E\left(Y \middle| \mathbf{X}_1, \ldots, \mathbf{X}_\mathrm{p}\right) + \epsilon = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \epsilon$$

  where the error ε is a Gaussian random variable with expectation zero and variance σ^2, written $\varepsilon \sim N(0, \sigma^2)$
- Different draws from random distribution, yield different target outcomes $\mathbf{Y}$ (and hence, different weights $\hat{\boldsymbol{\beta}}$ in trained model)
- Variance and standard deviation of feature weights $\hat{\boldsymbol{\beta}}$ can be estimated from this (Think: train model multiple times with different true y values for instances)
- It can be shown that

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}\sigma^2)$$

- Typically, one estimates the variance σ^2 by

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

- Standard error $\mathrm{SE}(\beta_j) = \sqrt{Var(\beta_j)} = \sqrt{(\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}_{jj}\hat{\sigma}^2}$

  (diagonal elements of variance-covariance matrix $\left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}\sigma^2$ contain variances)

Linear regression
Feature importance

Feature importance measured with t-statistic

- The t-statistic $t_{\hat{\beta}_j}$ is the estimated weight $\hat{\beta}_j$ of a feature scaled with its standard error $SE(\hat{\beta}_j) = \hat{\sigma}$

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

- t-statistic increases with increasing weight
- t-statistic decreases with increasing standard error
  (= the less certain we are about the correct value)

Interpretation

- The t-statistic can be interpreted as feature importance
- The higher the t-statistic, the more important the feature is

Linear regression

Advantages

Linearity

- Makes the estimation procedure simple
- Weights are easy to understand
- Widespread in use

Supports confidence intervals

- Confidence interval: range for the weight estimate that covers the “true” weight with a certain confidence
- Example: 95% confidence interval for a weight of 2 could range from 1 to 3
- Interpretation: If we repeated the estimation 100 times with newly sampled data, the confidence interval would include the true weight in 95 out of 100 cases, given that the linear regression model is the correct model for the data.

Whether the model is the “correct” model depends on assumptions

- Linearity
- Normality
- Homoscedasticity
- Independence
- Fixed features
- Absence of multicollinearity

💡 Assumptions are optional

You only need the assumptions to get further things out of the linear model like confidence intervals.

Linear regression
Assumptions

Linearity

- Prediction is a linear combination of feature values
  (which is both its greatest strength and its greatest limitation)
- Linearity leads to interpretable models
- Linear effects are easy to quantify and describe

Normality

- Assumption: target outcome given the features follows a normal distribution
- If assumption violated: estimated confidence intervals of feature weights invalid

Homoscedasticity (constant variance)

- The variance of the error terms $\hat{y}^{(i)} - y^{(i)}$ is assumed to be constant over the entire feature space
- Assumption often not fulfilled in practice
- Example: Prediction of house prices. For higher prices often higher variance

Linear regression
Assumptions

Independence

- Assumption: each instance is independent of any other instance
- Example: If you perform repeated measurements, such as multiple blood tests per patient, the data points are not independent

Fixed features

- Input features are considered "fixed"
- Fixed means that they are treated as "given constants" and not as statistical variables. This implies that they are free of measurement errors
- This is an unrealistic assumption but greatly simplifies the model

Absence of multicollinearity

- If features are strongly correlated, this messes up the estimation of the weights
- If two features are correlated: arbitrary which feature receives the weight

💡 In practice

→ Assumptions are often violated
→ Nevertheless, linear regression, standard error, and t-statistic are often useful (often assumptions are only "slightly" violated)

Linear regression

Interpretation of feature weight depends on feature type

Numerical feature:

- Example: size of a house
- Interpretation: An increase of feature x_k by one unit increases the prediction for y by β_k units when all other feature values remain fixed

Binary feature: Feature that takes on one of two possible values

- Example: “house has garden”
- Interpretation: Change from 0 to 1, increases prediction by feature weight

Categorical feature: A feature with a fixed number of possible values

- Example: feature “floor type” with categories “carpet”, “laminate” and “parquet”
- Typically represented with one-hot-encoding: one binary feature per category
- Interpretation:
  - Decrease prediction by weight of old category and
  - Increase by weight of new category

Intercept $\boldsymbol{\beta_0}$: the feature weight for the “constant feature” x_0 which is always 1 for all instances

- Interpretation: model prediction when all feature values are 0

Bike rental

Predict number of rented bikes on given day

	Weight	SE	|t|
(Intercept)	2742.4	339.0	8.1
seasonSPRING	454.8	169.9	2.7
seasonSUMMER	263.3	218.7	1.2
seasonFALL	636.9	152.7	4.2
holidayY	-441.1	239.0	1.8
workdayY	242.0	101.9	2.4
weatherMISTY	-345.3	120.2	2.9
weatherBAD	-1804.1	306.0	5.9
temp	51.0	10.3	5.0
hum	-21.7	4.5	4.8
windspeed	-46.5	9.5	4.9
cnt_2d_bfr	0.6	0.0	19.3

How do you interpret the previous table?

Interpretation of weights

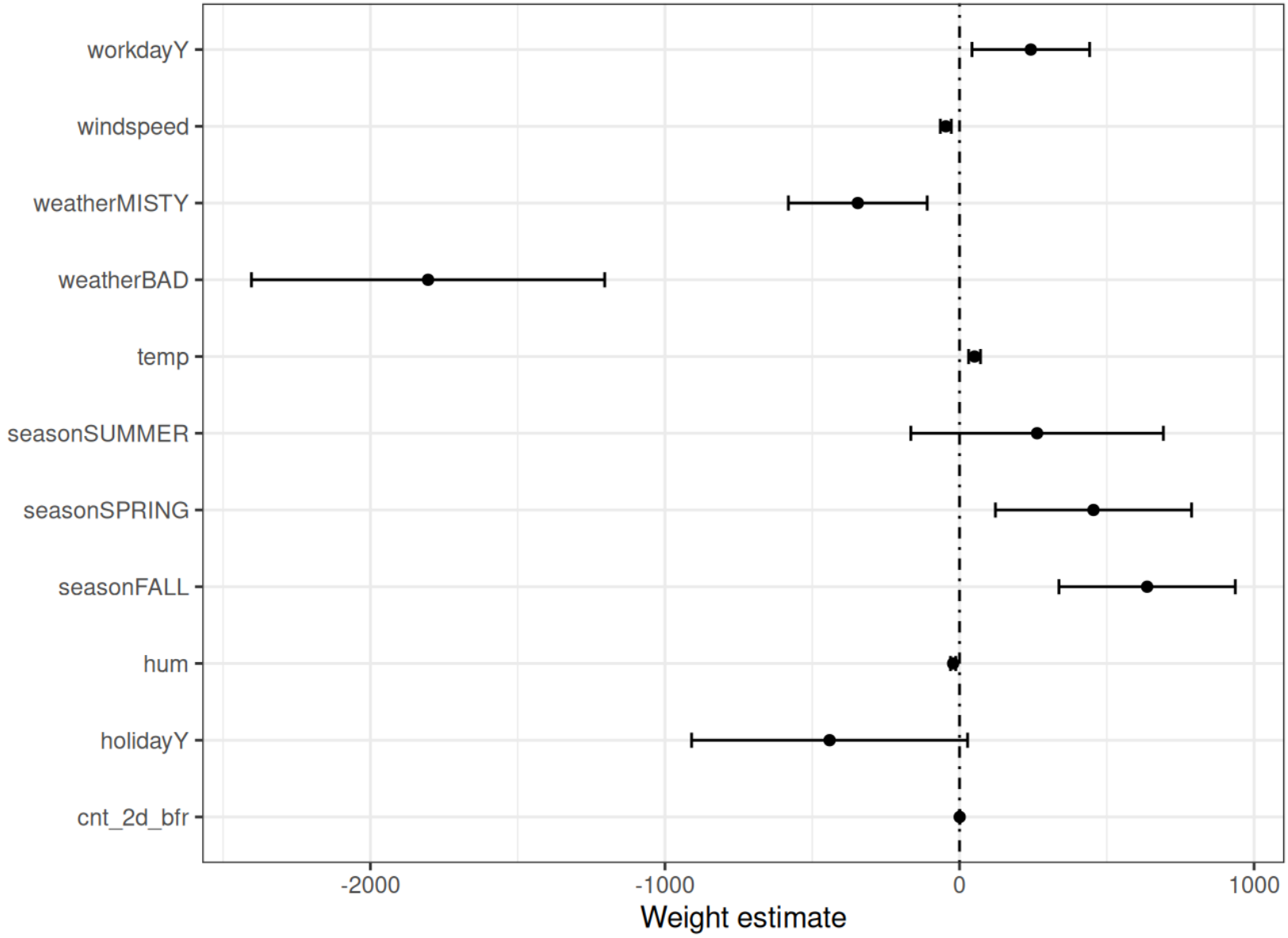
- Interpretation numerical feature (temperature): An increase of the temperature by 1 degree Celsius increases the predicted number of bicycles by 51.0, when all other features remain fixed.
- Interpretation of a categorical feature ("weatherBAD"): The estimated number of bicycles is -1804.1 lower when it is raining, snowing or stormy, compared to good weather – again assuming that all other features do not change.
- Why is weight of `seasonSPRING` much higher than `seasonSUMMER`?
  - Possibly due to correlated features: In the summer, the temperature is higher and this already explains why more bicycles are rented. In spring, bicycles might be rented although the temperature is still low.
- Which feature has the lowest t-statistic?
  - `seasonSUMMER`
- Which feature has the highest t-statistic?
  - `cnt_2d_bfr`

⚠ Don't compare weights of features with different scales

The units a feature is measured in affects the magnitude of coefficients/weights. For example, if you multiply a feature by 1000, like when converting from kilogram to gram, then the new coefficient will be smaller by a factor of 1/1000.

Visualization of feature weights

(weight plot with 95% confidence intervals)



Normalization of feature values

Problem of weight plot (and weight table):

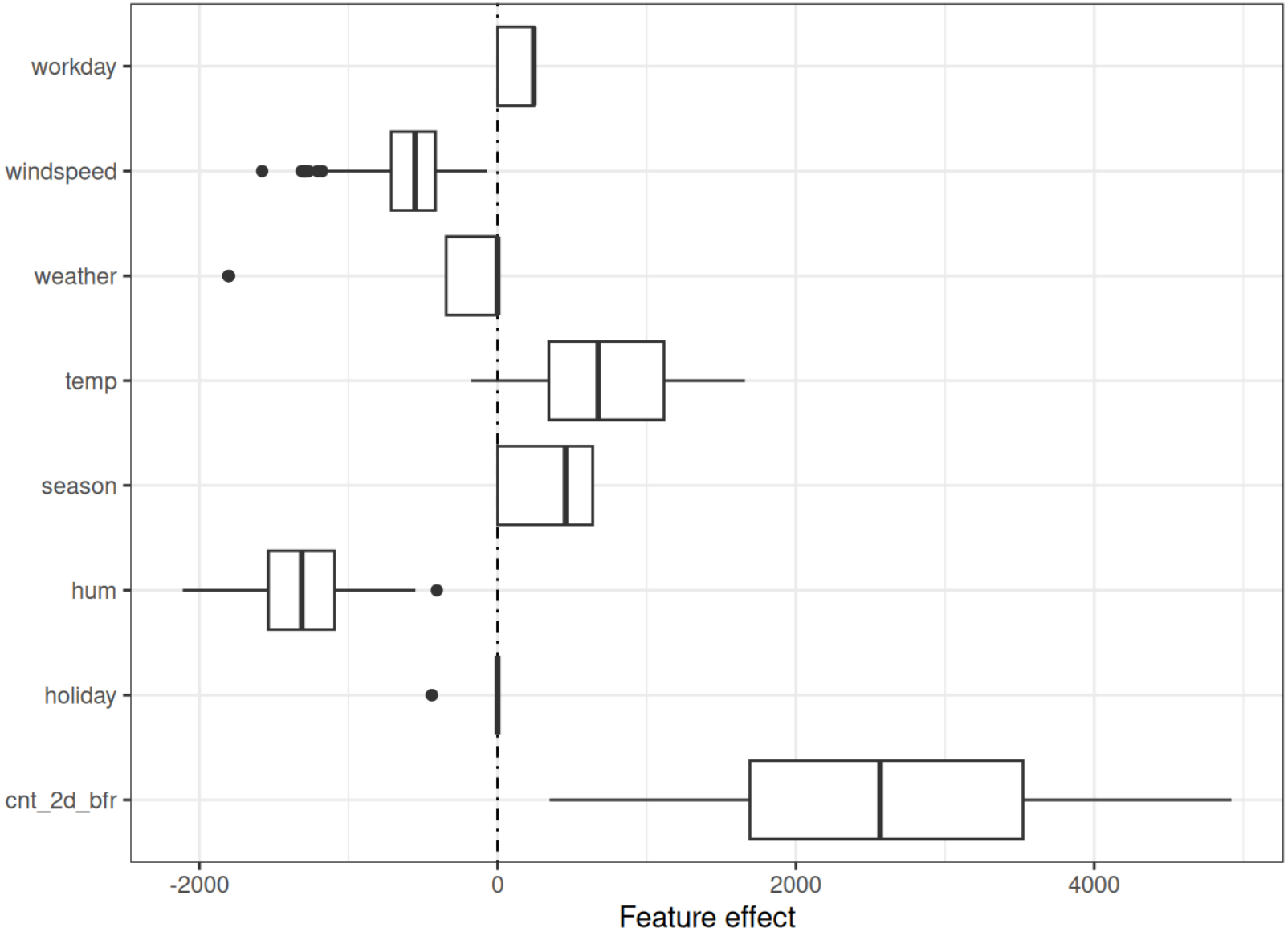
- Features are measured on different scales
- Weather: reflects difference between good and rainy/stormy/snowy weather
- Temperature: reflects increase of 1 degree Celsius

→ Make features more comparable by scaling them before fitting the model (zero mean and standard deviation of one)

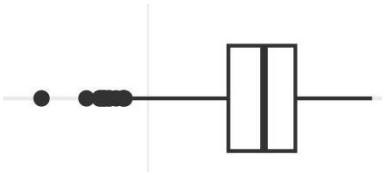
→ Alternative: use effect plot (see following slides)

Effect plot

Feature effect = feature value times feature weight



Effect plot



Problem

- Weights depend on scale of the feature
  (e.g., measure a persons height in meters or centimeters)
- The weight can be different although the actual effect of the feature is the same
- Low variance features: almost all instances have similar contribution from feature

Solution: effect

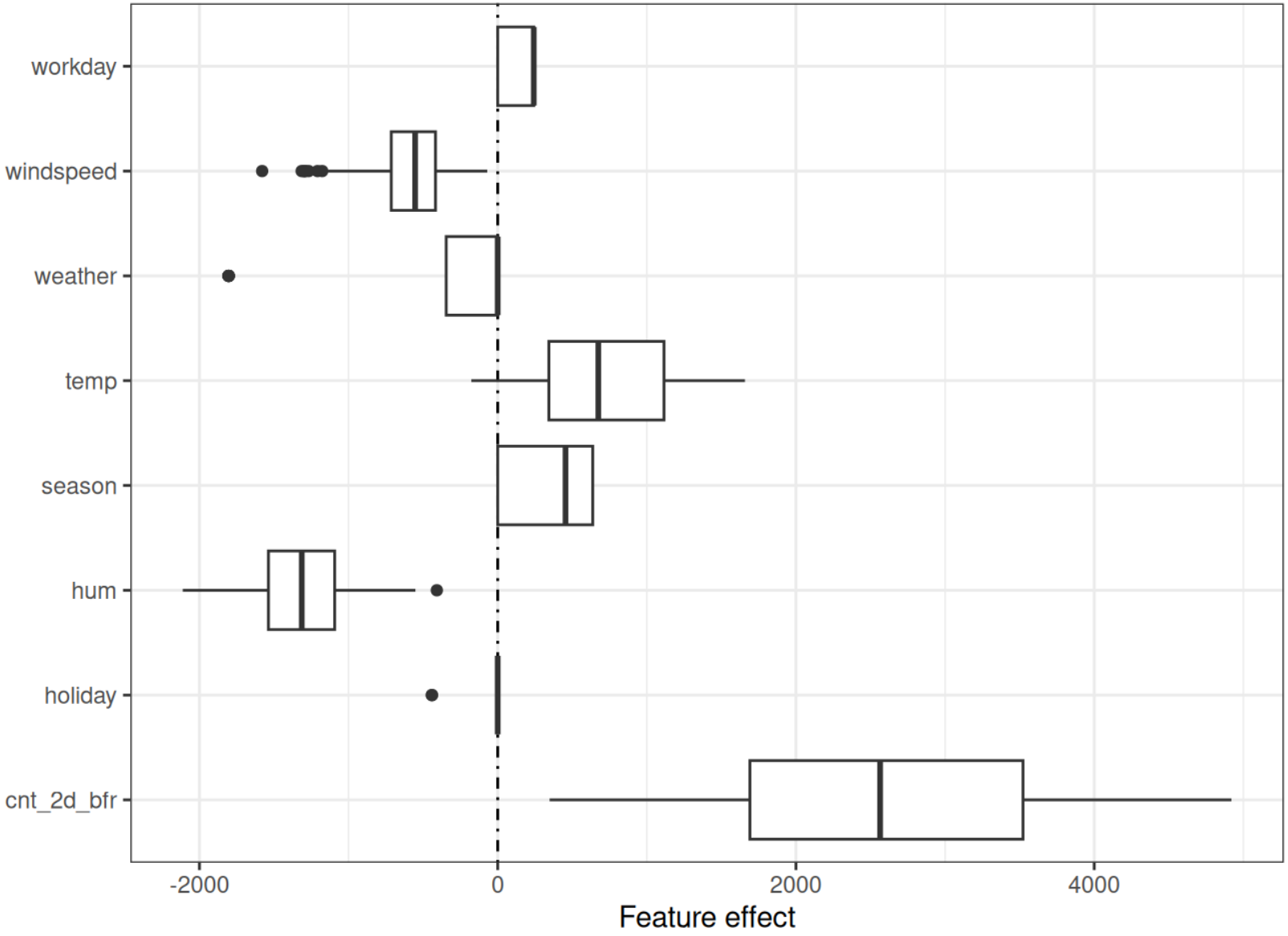
- Feature effect = weight per feature times the feature value of an instance

$$\text{effect}_j^{(i)} = \beta_j x_j^{(i)}$$

- Boxplot: show distribution across all instances
  - Lower and upper limits: 25% to 75% of effect quantiles
    (25% of instances have a lower effect, 75% of instances have a lower effect)
  - Vertical line in middle: median effect
    (50% of the instances have a lower effect)
  - Dots: Outliers (1.5 * IQR below the first quartile,1.5 * IQR above the third quartile, IQR, interquartile range, contains the second and third quartile, i.e., 50% of points)
  - Lower and upper whiskers (horizontal lines)
    connect the **observed** points below the first quartile and above the third quartile **that are not outliers**
- Note: categorical feature effects can be summarized in a single boxplot

Effect plot

Feature effect = feature value times feature weight



How do you interpret the previous plot?

Interpretation of effect plot

Large positive effects (on expected number of rented bicycles)

- Temperature: `temp`
- Rented bikes two days before: `cnt_2d_bfr`

Large negative effects (on expected number of rented bicycles)

- Humidity: `hum`
- Wind Speed: `windspeed`

Large interquartile range (IQR)

- Rented bikes two days before: `cnt_2d_bfr`
  (large range of feature values)
- Temperature: `temp`
  (large range of feature values)

Small interquartile range (IQR)

- Holiday: small weight and only a binary feature (mostly 0)
- Working Day: small weight and only a binary feature,
  less than 25% of days are holidays → Boxplot around 0)

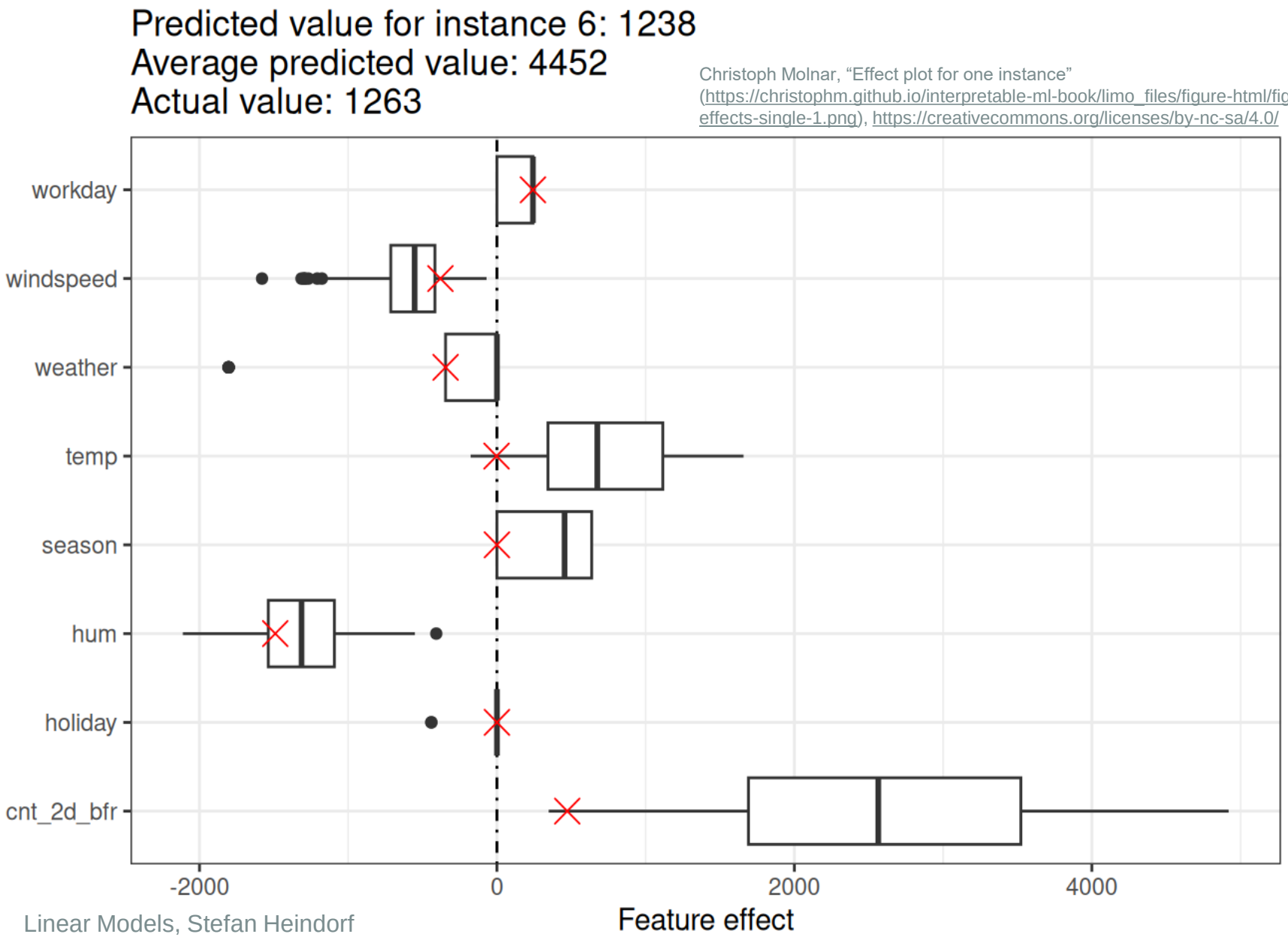
Explain individual predictions

Feature values for instance 6

Feature	Value
season	WINTER
holiday	N
workday	Y
weather	MISTY
temp	-0.052723
hum	68.6364
windspeed	8.182844
cnt_2d_bfr	822
cnt	1263

Individual effect plot: explain individual predictions

Feature values for instance 6



Christoph Molnar, "Effect plot for one instance" (https://christophm.github.io/interpretable-ml-book/limo_files/figure-html/fig-linear-effects-single-1.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

Encoding of categorical features

Discussion of different options [Seabold 2023, UCLA]

Example: six instances, three categories

- 1: A 3: B 5: C
- 2: A 4: B 5: C

Treatment coding (reference category: A)

- First column always 1
  (intercept β_0= mean prediction of reference category)
- Second, third columns: B, C

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Effect coding (reference: grand mean)

- First column: always 1
  (intercept β_0= grand mean:
  mean of means prediction of all categories)
- Second, third columns: B, C

$$\begin{pmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Dummy coding

- First, second, third columns: A, B, C
- No intercept necessary (avoids multiple solutions)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Do linear models create good explanations?

Recall human-friendly explanations from first lecture

Contrastive

- Somewhat contrastive
- The weights times feature values (feature effects) explain outcome contrastive to “median-instance” (see individual effect plot)

Selective

- Linear models not selective by default
- Selectivity can be achieved
  - Train sparse models (most feature weights are 0, see LASSO below)
  - Only include few features in the model (see feature selection techniques below)

Truthful

- If linear equation is an appropriate model for the relationship between features and outcome
- Less truthful if many non-linearities and feature interactions

LASSO

Least absolute shrinkage and selection operator

Current optimization problem

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)T}\boldsymbol{\beta} \right)^2$$

LASSO adds regularization term $\lambda\|\boldsymbol{\beta}\|_1$:

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{x}^{(i)T}\boldsymbol{\beta} \right)^2 + \lambda\|\boldsymbol{\beta}\|_1$$

where

$$\|\boldsymbol{\beta}\|_1 = |\beta_0| + |\beta_1| + \cdots + |\beta_p| = \sum_{0}^{p} \beta_i$$

is L1-norm, which penalizes large weights

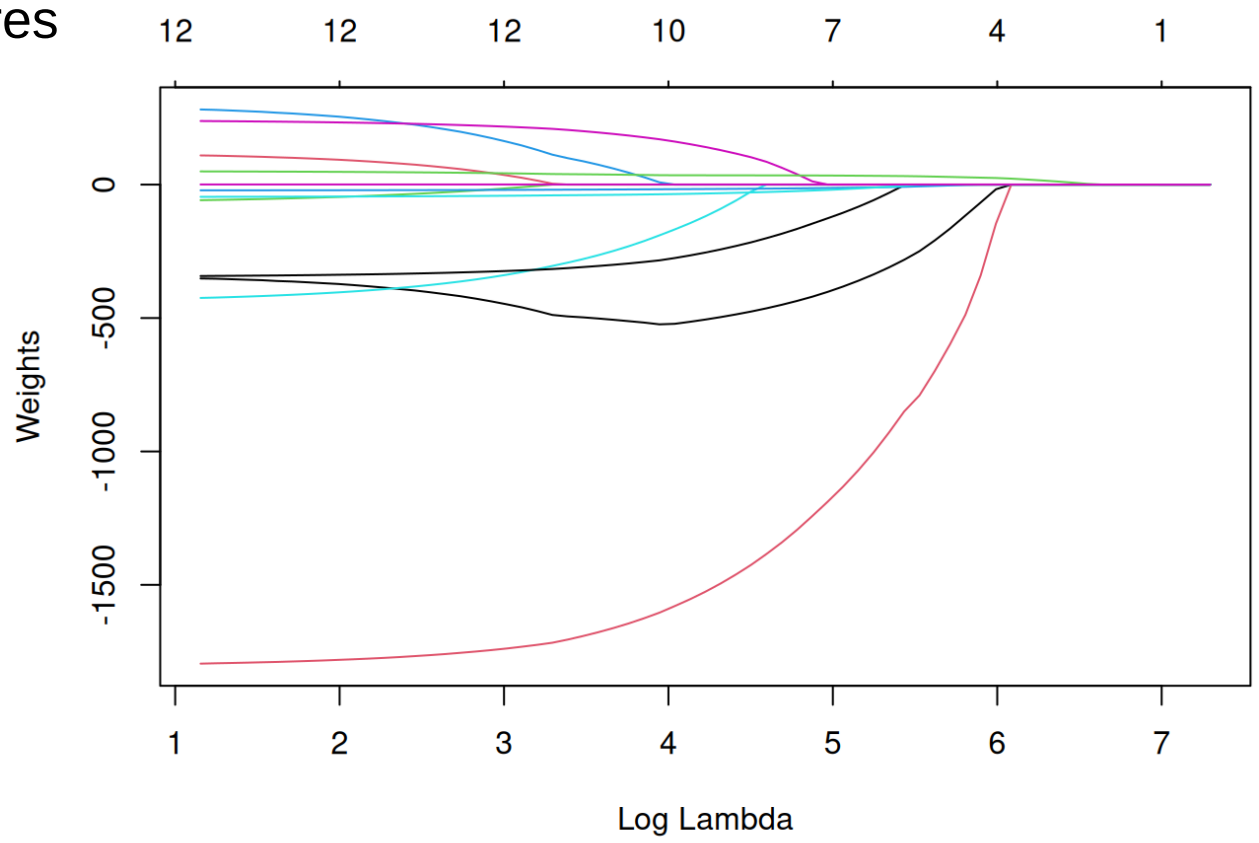
- Many of the weights receive an estimate of 0, others are shrunk
- Parameter lambda (λ) controls strengths of regularization (usually tuned by cross-validation)

Feature weights depending on Lambda

Example

- With increasing lambda, fewer features receive non-zero weights
- Each feature is represented by different color
- Number above plot shows number of non-zero weights

Features



Example with LASSO

Tune lambda to obtain **2 features**

	Weight
seasonWINTER	0.00
seasonSPRING	0.00
seasonSUMMER	0.00
seasonFALL	0.00
holidayY	0.00
workdayY	0.00
weatherMISTY	0.00
weatherBAD	0.00
temp	22.45
hum	0.00
windspeed	0.00
cnt_2d_bfr	0.48

Feature selection

Alternative to LASSO to obtain a linear model with few features

Pre-processing methods

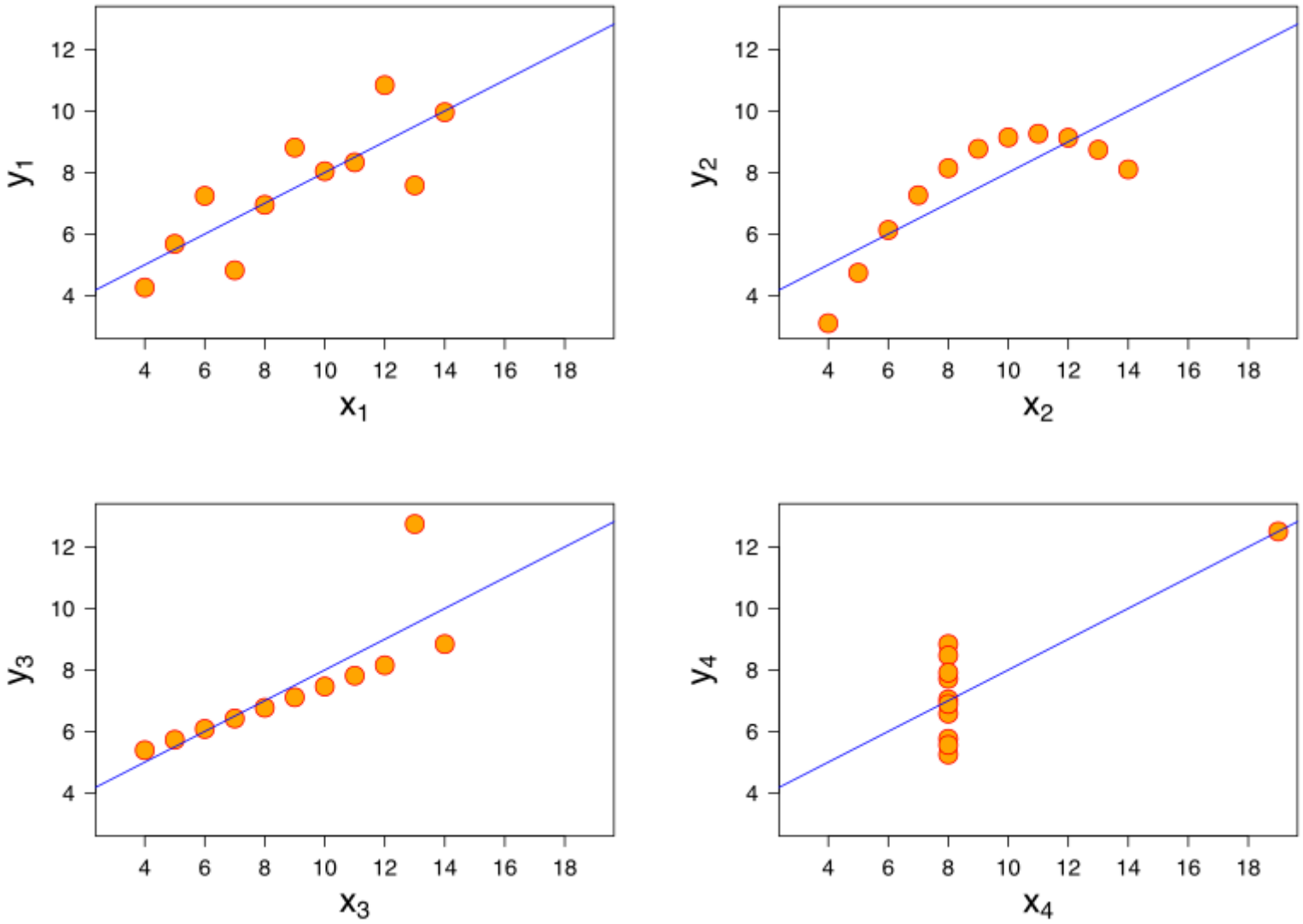
- Manually selected features: use expert knowledge to select or discard features
  - Drawback 1: cannot be automated
  - Drawback 2: need someone who understands the data
- Univariate selection: Only consider features that exceed a threshold of correlation between the feature and the target
  - Drawback: Only considers features individually
  - Some features might not show a correlation until the linear model has accounted for some other features

Step-wise methods

- Forward selection
  - Fit model with one feature. Do this with each feature. Select model that works best
  - Repeat process and always add a feature to current best model until a stopping criterion is reached (e.g., maximum number of features)
- Backward selection
  - Similar to forward selection
  - Instead of adding features, start with model that contains all features and try removing one feature at a time to get the highest performance increase
  - Repeat until some stopping criterion is reached

Interpretation

Be careful! Same model, different underlying data



Linear regression
Advantages

- Modeling of predictions as a weighted sum makes it transparent
- LASSO and/or feature selection can keep the number of features small
- Linear regression is widely used
  - Many people have experience with it
  - Lots of teaching material on it
  - Linear regression can be found in R, Python, Java, Julia, Scala, Javascript, …
- Mathematically straightforward to estimate the weights
- Guaranteed to find optimal weights (given all assumptions of the linear regression model are met by the data)
- Solid statistical theory
  - Confidence intervals (e.g., for weights)
  - Statistical tests (e.g., for weights)
- Many extensions of linear models available (generalized linear models [GLM], generalized additive models [GAM])

Linear regression

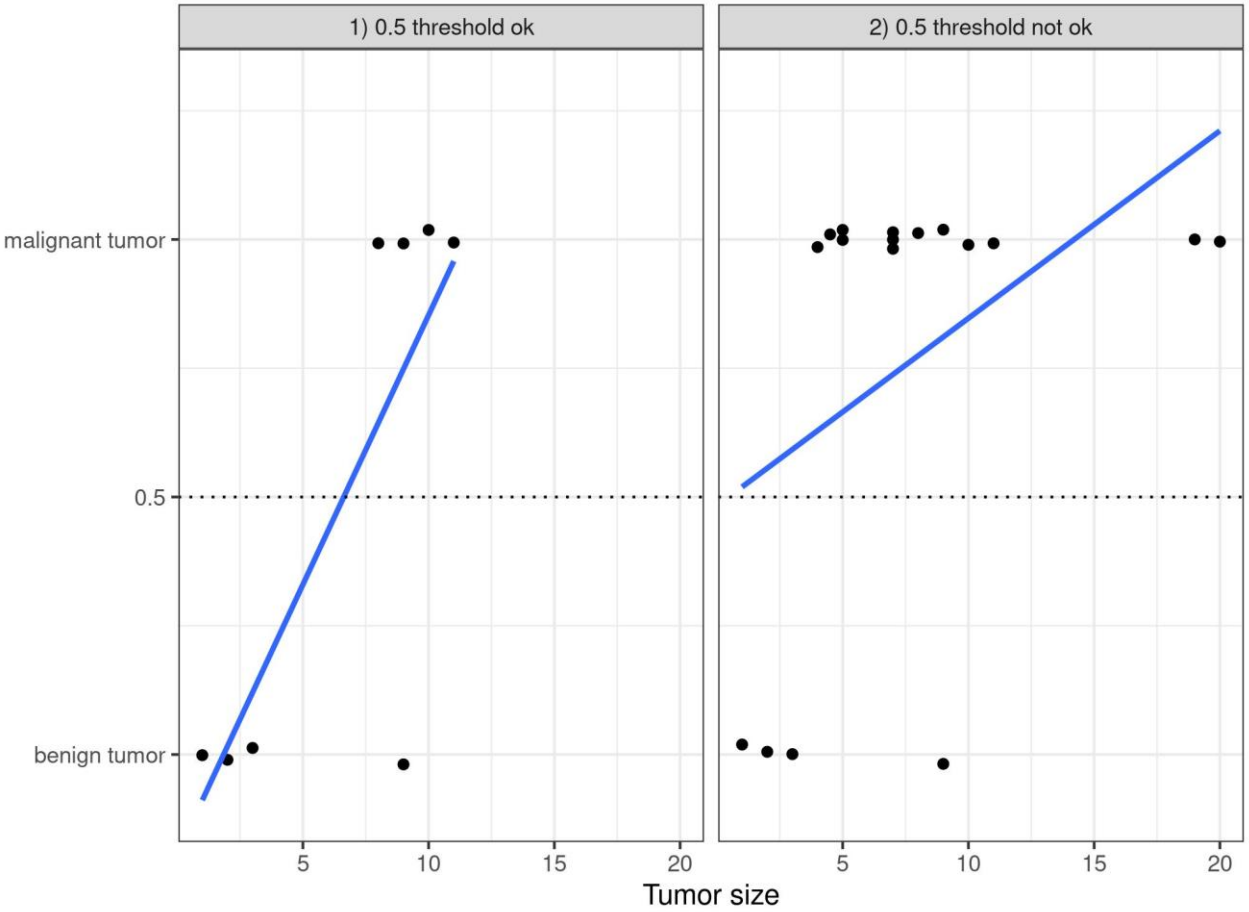
Disadvantages

- Can only represent linear relationships
- Each nonlinearity or interaction has to be hand-crafted and explicitly given to the model as an input feature
- Linear models are often not that good regarding predictive performance, because learned relationships usually oversimplify how complex reality is
- The interpretation of a weight can be unintuitive because it depends on all other features
  - Feature with high positive correlation with the outcome y and another feature might get a negative weight in the linear model, because, given the other correlated feature, it is negatively correlated with y in the high-dimensional space
  - Completely correlated features make it impossible to find unique solution for the linear equation
- Example: Model to predict house price
  - Features house size and number of rooms are highly correlated
  - If you take both features, it might happen, that the size of the house is the better predictor and gets a large positive weight
  - The number of rooms might end up getting a negative weight, because, given that a house has the same size, increasing the number of rooms could make it less valuable or the linear equation becomes less stable, when the correlation is too strong

Logistic regression

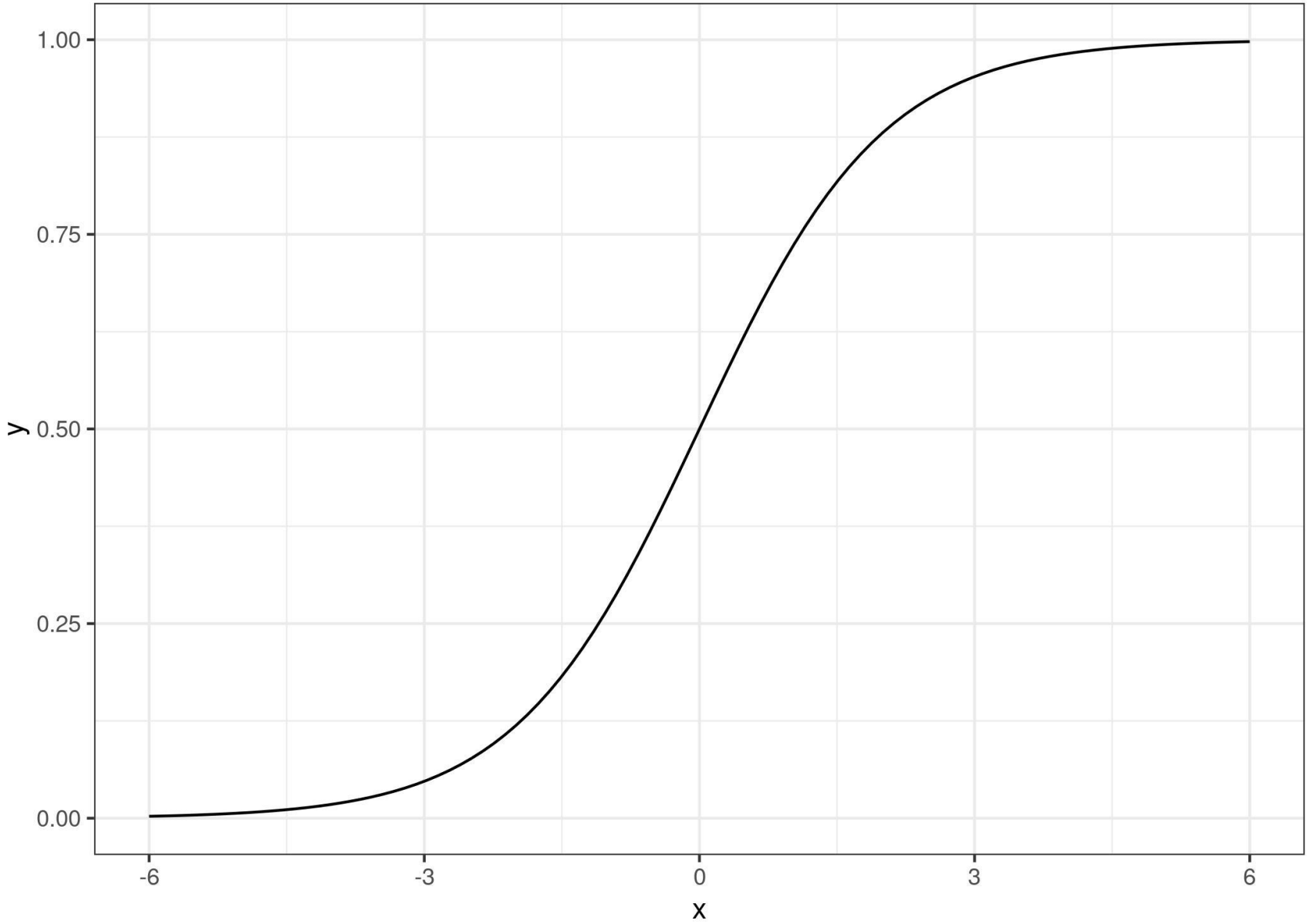
Why linear regression is not suitable for classification?

- In principle, linear regression can be used for classifications
- But: Linear model can output numbers below 0 and above 1
  → cannot be interpreted as probabilities
- Logistic regression often gives better results for classification problems



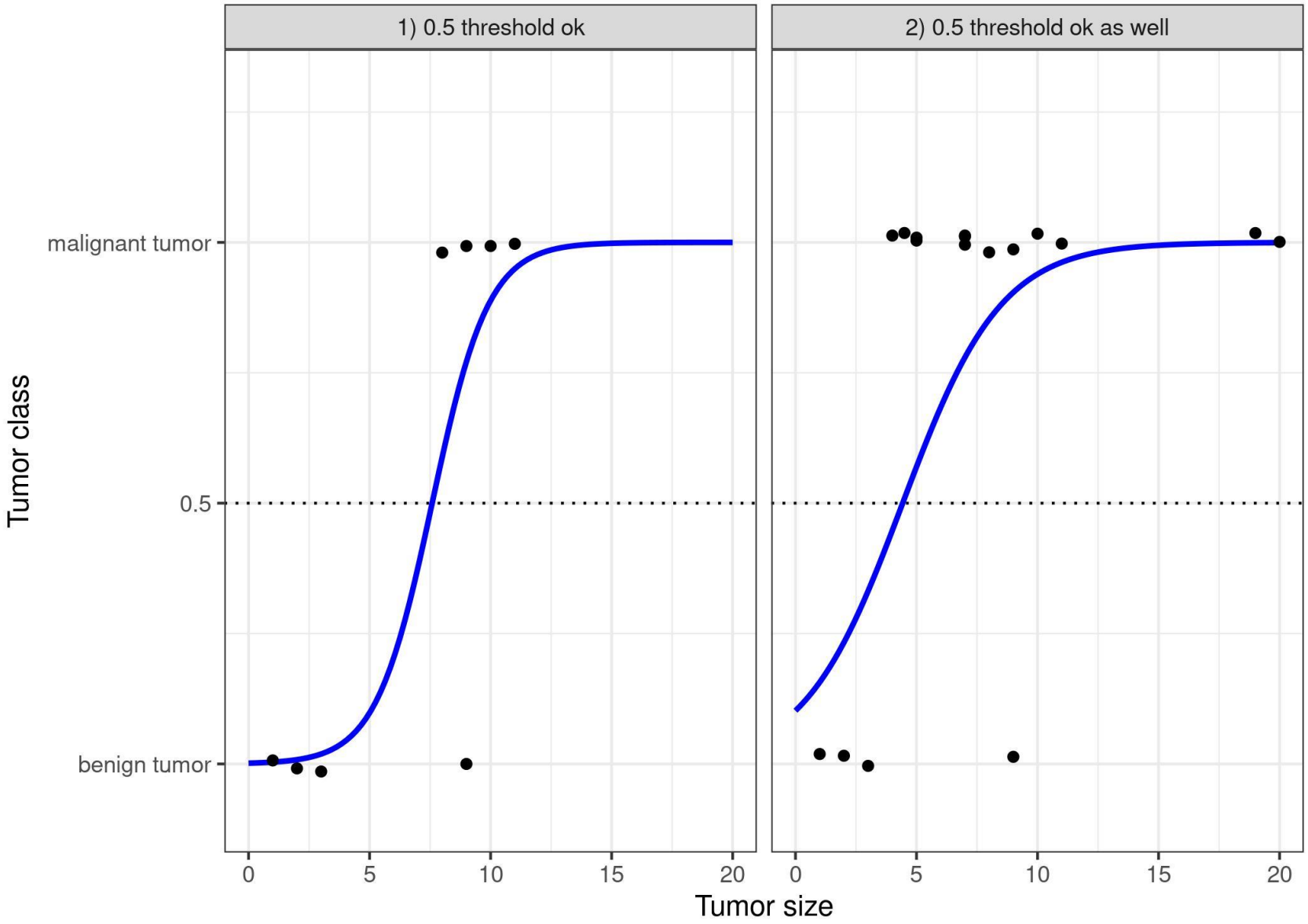
Logistic regression

Logistic function



Logistic regression

Example



Logistic regression

Do not fit a straight line

Use a logistic function

$$\text{logistic}(\eta) = \frac{1}{1 + \exp(-\eta)}$$

and fit

$$\hat{y} = P(y = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p))}$$

Logistic regression

Interpretation: Odds

With probability p

$$\text{odds} = \frac{p}{1-p}$$

Example: Rolling a six-sided dice

- Probability of rolling a 3 $1/6$ (1 out of 6 events)
- Odds of rolling a 3 $1/5$ (1 vs. 5 events)

- Probability of rolling a 2 or 4 $2/6 = 1/3$ (2 out of 6 events)
- Odds of rolling a 2 or 4 $2/4 = 1/2$ (2 vs 4 events)

Logistic regression

Interpretation: Odds

kicktipp

◀ Final ▶ | Bonus

Odds for your predictions

12/17/22 4:00 PM	Croatia	Morocco	**1-0**	2.35 / 3.45 / 3.00
12/18/22 4:00 PM	Argentina	France	**0-1**	2.70 / 3.00 / 2.90

Logistic regression

Interpretation: linear model for the log odds

Logistic regression is linear model for the log odds:

$$\boxed{\ln(\text{odds}) = \ln\left(\frac{P(y=1)}{P(y=0)}\right) = \mathbf{x}^{\mathbf{T}}\boldsymbol{\beta}}$$

because

$$\hat{y} = P(y=1) = \frac{1}{1+\exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)} = \frac{1}{1+\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}$$

and

$$\ln\left(\frac{P(y=1)}{P(y=0)}\right) = \ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \ln\left(\frac{\frac{1}{1+\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}}{1-\frac{1}{1+\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}}\right)$$

$$= \ln\left(\frac{\frac{1}{1+\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}}{\frac{\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}{1+\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}}\right) = \ln\left(\frac{1}{\exp(-\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})}\right) = \ln\left(\exp(\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta})\right) = \mathbf{x}^{\mathbf{T}}\boldsymbol{\beta}$$

Logistic regression

Interpretation: feature changes as odds ratios

From

$$\ln(\text{odds}) = \mathbf{x}^{\mathbf{T}}\boldsymbol{\beta}$$

it follows that

$$\text{odds} = \exp\left(\mathbf{x}^{\mathbf{T}}\boldsymbol{\beta}\right)$$

What happens when one feature value x_j is increased by 1 unit?

$$\begin{aligned}\frac{odds_{x_j+1}}{odds_{x_j}} &= \frac{\exp\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_j\left(x_j + 1\right) + \cdots + \beta_p x_p\right)}{\exp\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \cdots + \beta_p x_p\right)} \\ &= \exp\left(\beta_j(x_j + 1) - \beta_j x_j\right) \\ &= \exp(\beta_j)\end{aligned}$$

Increasing the feature value x_j by 1 unit
→ increases the odds by a factor of $\exp(\beta_j)$

Logistic regression

Example: Predict penguin P(female)

	Weight	Odds ratio	Std. Error
(Intercept)	95.86	4.28931e+41	29.29
bill_depth_mm	-2.06	0.13	0.83
bill_length_mm	-0.53	0.59	0.19
flipper_length_mm	-0.16	0.85	0.11
chonkinessRegular_Penguin	0.65	1.92	1.25
chonkinessAbsolute_Unit	-0.84	0.43	1.66

Example 1: An increase in a penguin's bill length increases the odds of being female by a factor of 0.59

Example 2:

Words of caution:

- Holds when all other features stay the same
- Correlation does not imply causation
  (to establish causality randomized controlled trials necessary)

Logistic regression

Advantages & disadvantages

Advantages

- Widely used by many different people
- Does not only allow classification, but yields probabilities
- Can be extended from binary classification to multi-class classification (multinomial regression)

Disadvantages

- Restrictive expressiveness (e.g., interactions must be added manually)
- Other models may have better predictive performance
- Interpretation is more difficult than linear regression because the interpretation of the weights is multiplicative and not additive

References

- **Hayashi**, Fumio. *Econometrics*. Princeton University Press, **2011**.
- **Hastie**, Trevor, Robert Tibshirani, Jerome H. Friedman, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. New York: springer, **2009**. https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf
- **Kolter**, Zico. Linear Algebra Review and Reference. **2020**. https://cs229.stanford.edu/notes2022fall/cs229-linear_algebra_review.pdf
- **Seabold**, Skipper, et al. Patsy: Contrast Coding Systems for categorical variables. **2023** https://www.statsmodels.org/stable/contrasts.html https://www.statsmodels.org/stable/examples/notebooks/generated/contrasts.html
- **Sheather**, Simon. *A modern approach to regression with R*. Springer Science & Business Media, **2009**.
- **UCLA**: Statistical Consulting Group. Introduction to SAS. https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-how-do-i-cite-web-pages-and-programs-from-the-ucla-statistical-consulting-group/