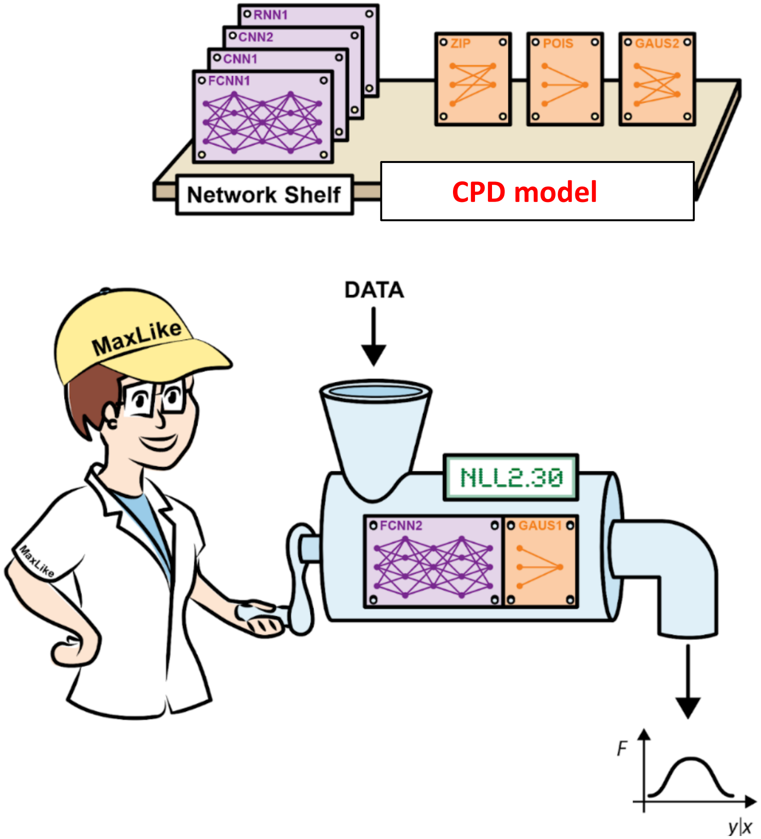
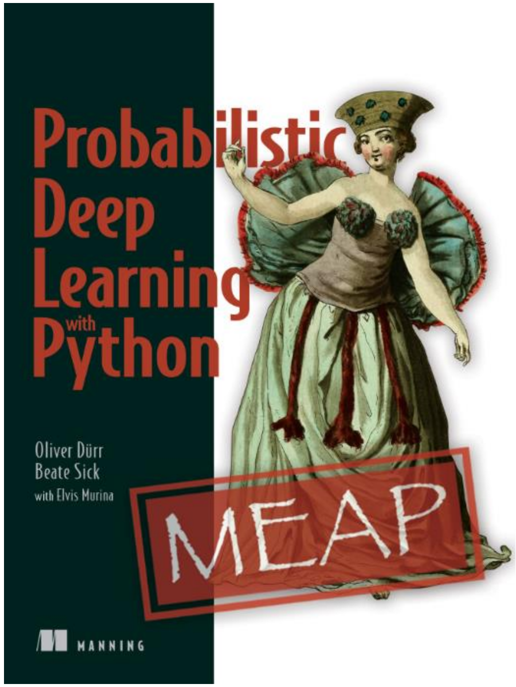
Intro to probabilistic DL models





Zurich University of Applied Sciences

zh aw School of Engineering

IDP Institute of Data Analysis and Process Design

Beate Sick

University of Zurich UZH

ETH zürich

What is a probabilistic model?

Simple regression via a NN: no probabilistic model in mind

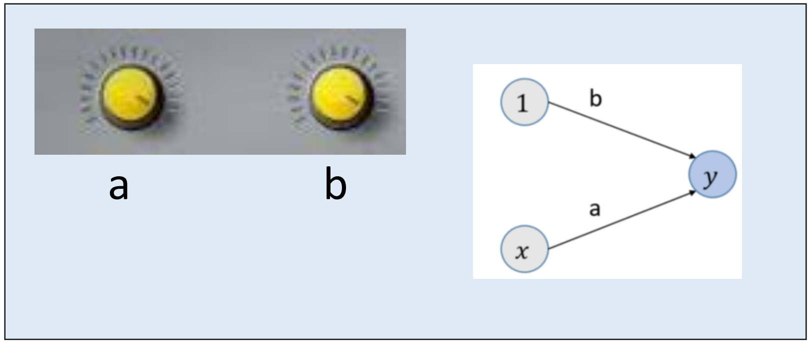
age

Input x

Systolic blood pressure

Output y

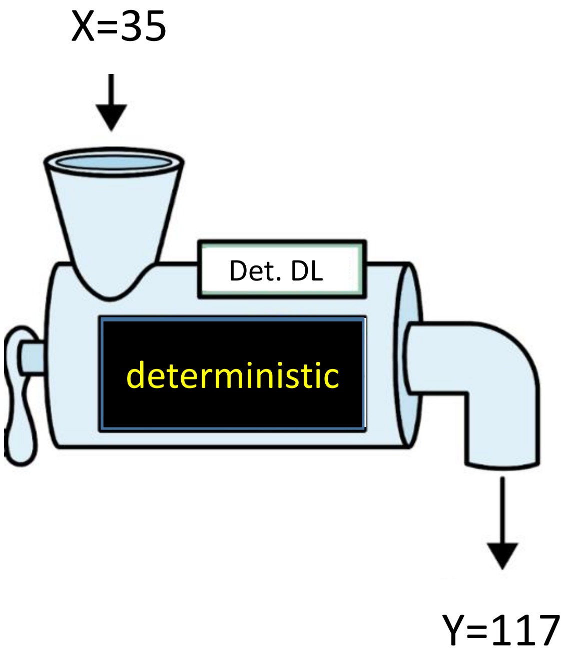
$x = 35$ →



→ $s\hat{b}p = 117$

One input x (age) → one predicted outcome (sbp)

Traditional versus probabilistic regression DL models

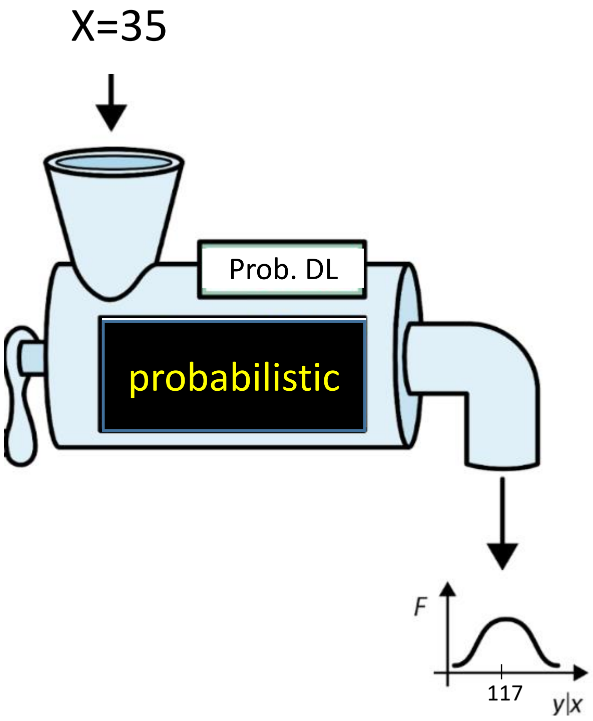


X=35

Det. DL

deterministic

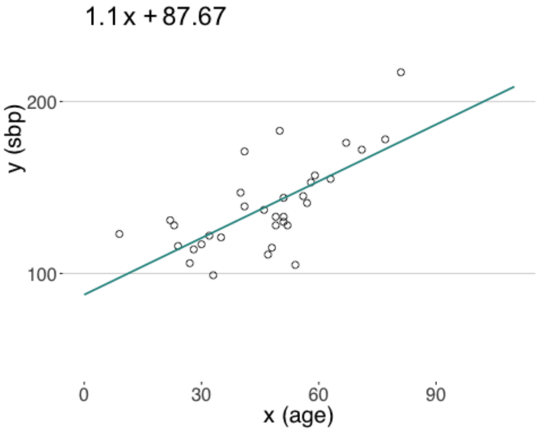
Y=117

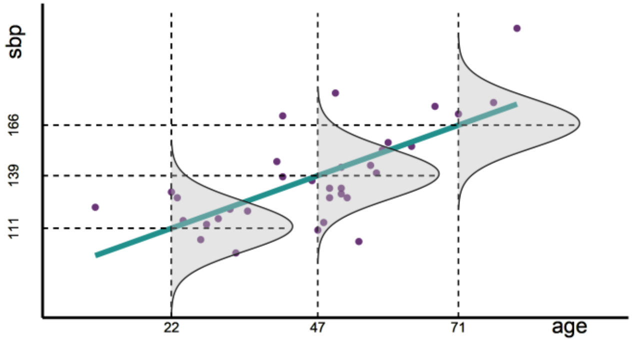


X=35

Prob. DL

probabilistic



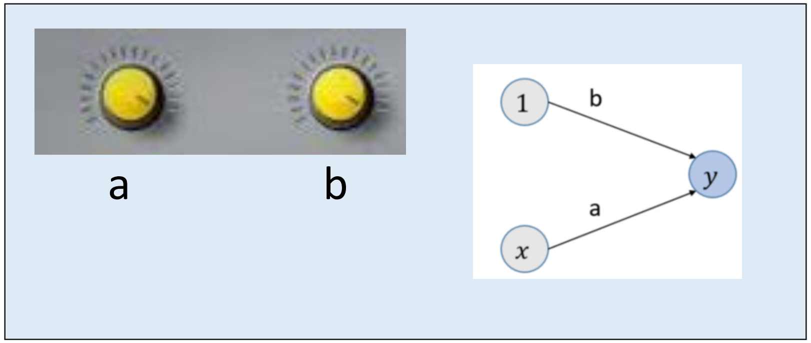


Binary classification: no probabilistic model in mind

Fake or real?

Quantify transparency

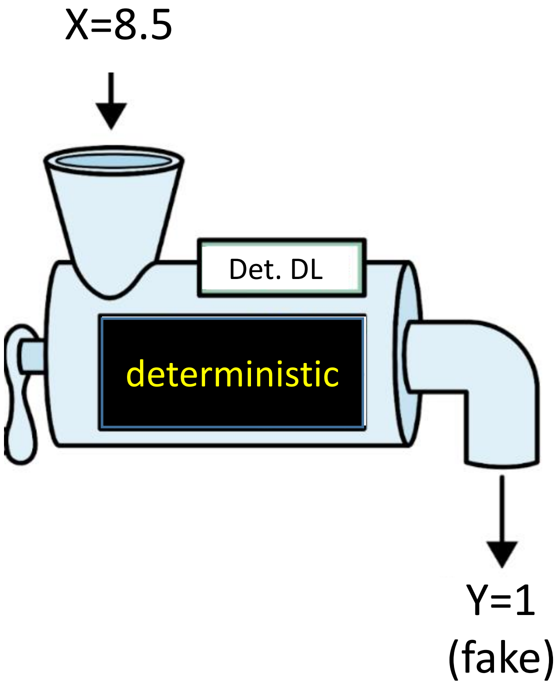
$x = 8.5$



fake

One input x → one predicted outcome

Traditional versus probabilistic classification DL models



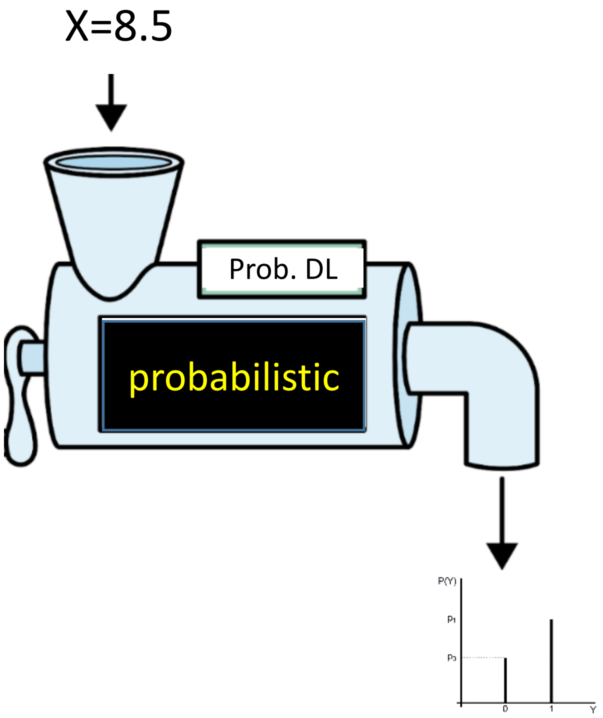
X=8.5

Det. DL

deterministic

Y=1

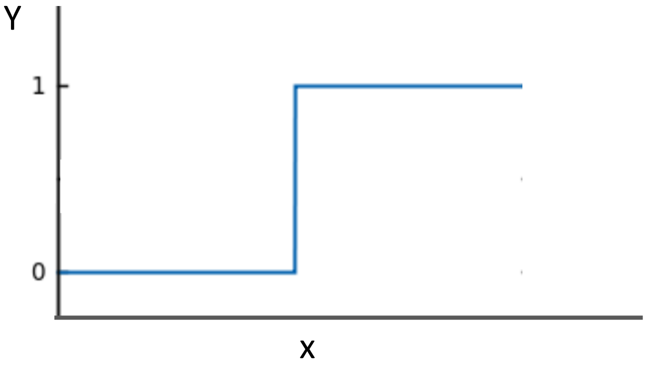
(fake)

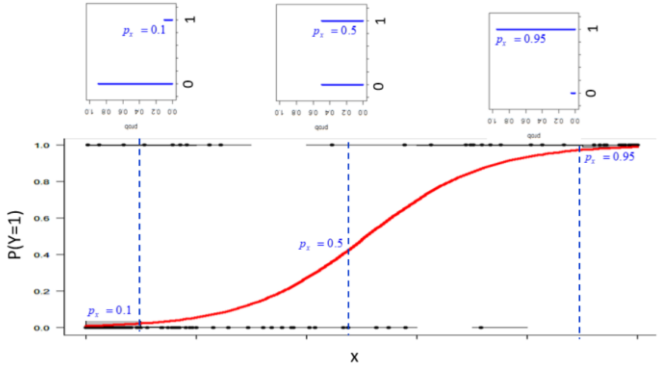


X=8.5

Prob. DL

probabilistic





Why is it important to know about probabilities?

Philosophical reasons:

“It is scientific to say what is more likely and what is less likely...”
Richard Feynman

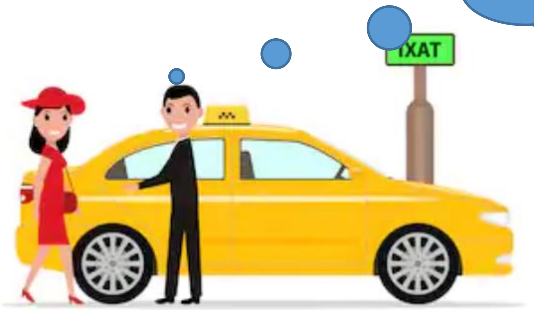
Practical reasons:

We often want to optimize expected costs which requires CPD for computing.

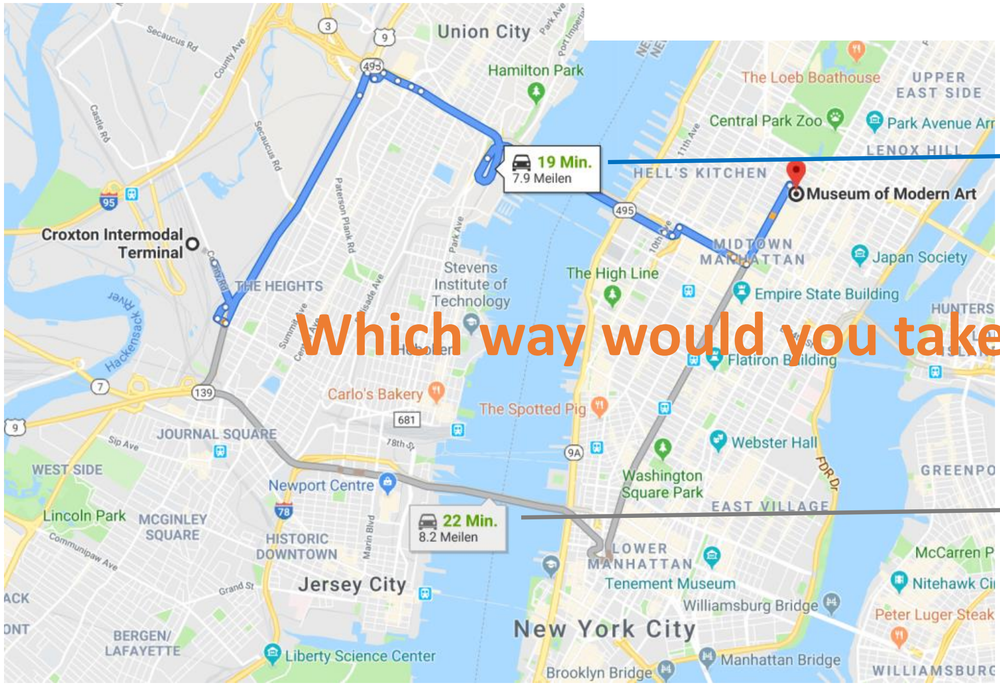
x

Probabilistic travel time prediction

You'll get 500$ tip if I arrive at MOMA within 25 minutes!

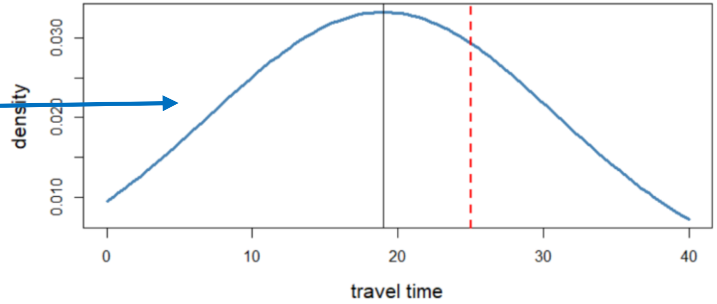


Let's use my probabilistic travel time gadget!

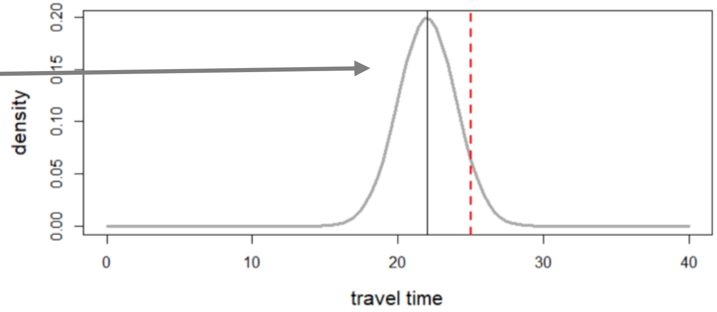


Which way would you take?

Chance to get tip: 69%



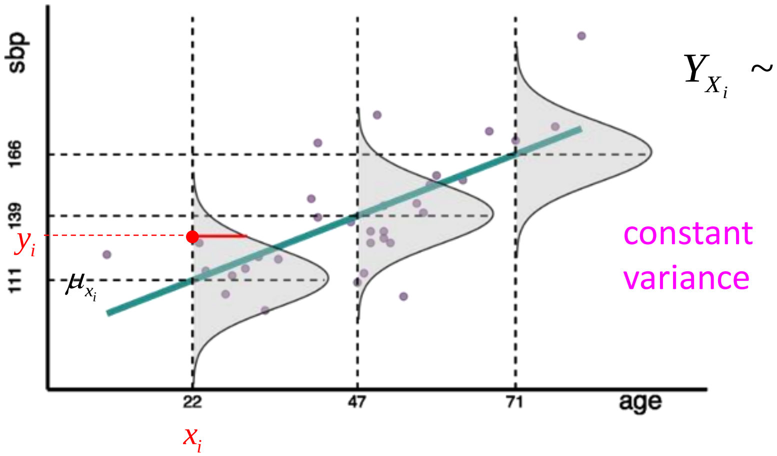
Chance to get tip: 93%



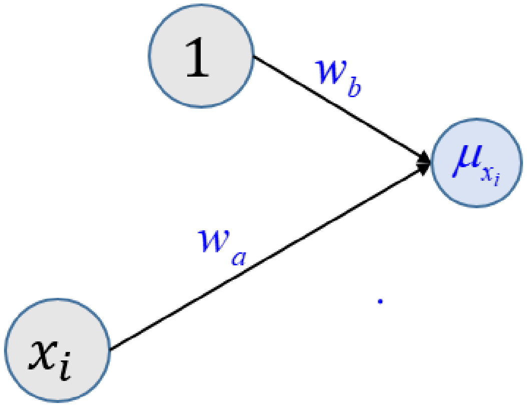
How to fit a probabilistic model?

How to train a NN to output the parameter of a CPD?

→ use the beautiful maximum likelihood principle



$$Y_{X_i} \sim \mathrm{N}(\mu_{x_i}, \sigma^2)$$



Maximum likelihood:

$$\mathbf{w}_{\mathrm{ML}} = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \, \mathrm{e}^{-\frac{(y_i-\mu_{x_i})^2}{2\sigma^2}}$$

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} -\log\left(\cancel{\frac{1}{\sqrt{2\pi\sigma^2}}}\right) + \frac{(y_i-\mu_{x_i})^2}{2\sigma^2}$$

$$(\hat{w}_a, \hat{w}_b)_{\mathrm{ML}} = \underset{w_a, w_b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (w_a \cdot x_i + w_b)\right)^2$$

gradient descent with MSE loss

$\hat{w}_a$ $\hat{w}_b$

How to get the variance which is assumed to be constant?

The constant variance drops out in the MSE optimization. There are two ways to get it **after the fitting.** Then μ_{x_i} are the predicted means.

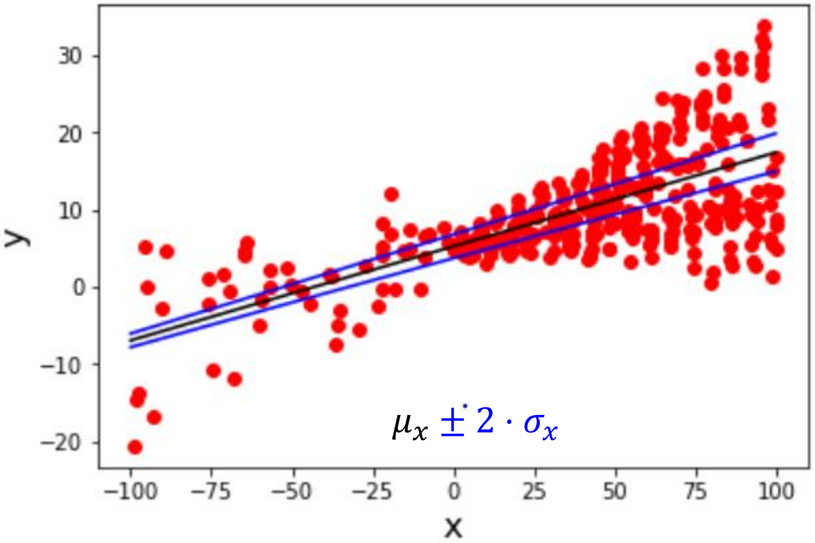
- From residuals after fitting, it's.

$$\hat{\sigma} = \frac{1}{n-2}\sum\left(y_i - \mu_{x_i}\right)^2$$

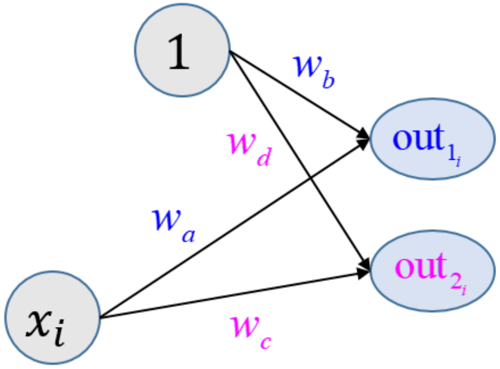
- By optimizing the variance σ in the NLL

$$\sum_{i=1}^{n} -\log(\frac{1}{\sqrt{2\pi\sigma^2}}) + \frac{\left(y_i - \mu_{x_i}\right)^2}{2\sigma^2}$$

Fit a probabilistic regression with non-constant variance



$$Y_{X_i} \sim \mathrm{N}(\mu_{x_i}, \sigma_x^2)$$



$$\mu_{x_i} = \mathrm{out}_{1_i}$$

$$\sigma_{x_i} = e^{\mathrm{out}_{2_i}}$$

Minimize the negative log-likelihood (NLL):

$$\hat{w} = argmin \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{2\pi\sigma_{x_i}^2}}\right) + \frac{(y_i - \mu_{x_i})^2}{2\sigma_{x_i}^2}$$

gradient descent with NLL loss

$\hat{w}_a$ $\hat{w}_b$

$\hat{w}_c$ $\hat{w}_d$

Modelling the standard deviation (positive values)

- The variance or standard deviation are both positive
- Neural networks output is not constrained.
- Two common approaches to fix this (exp or softplus)

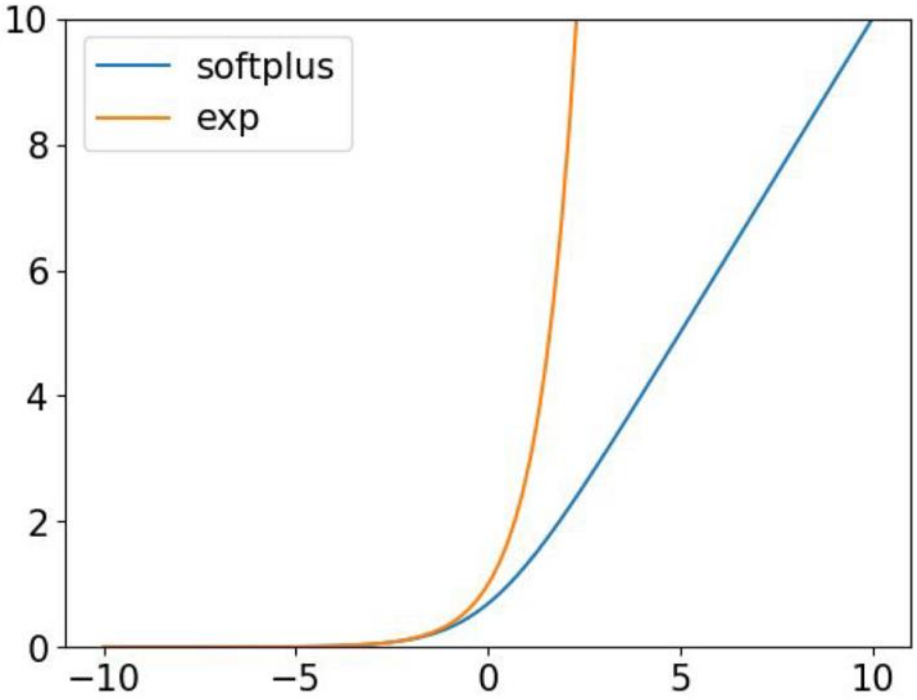
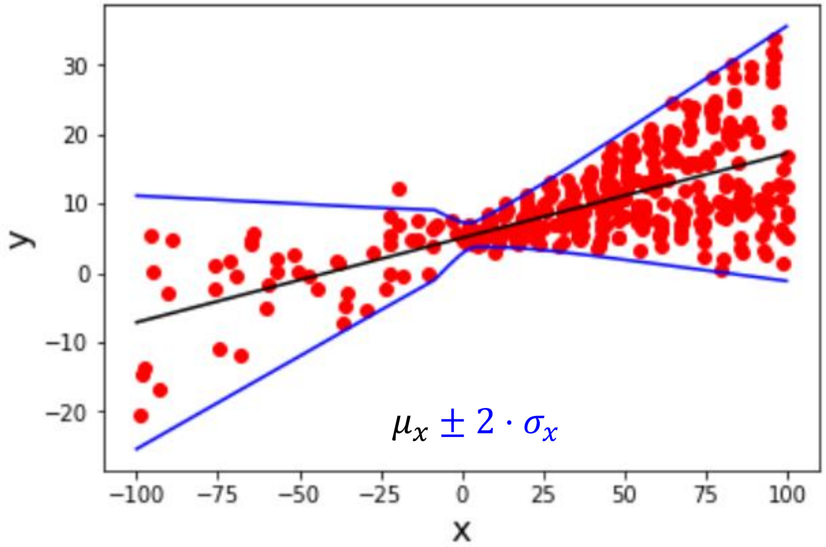
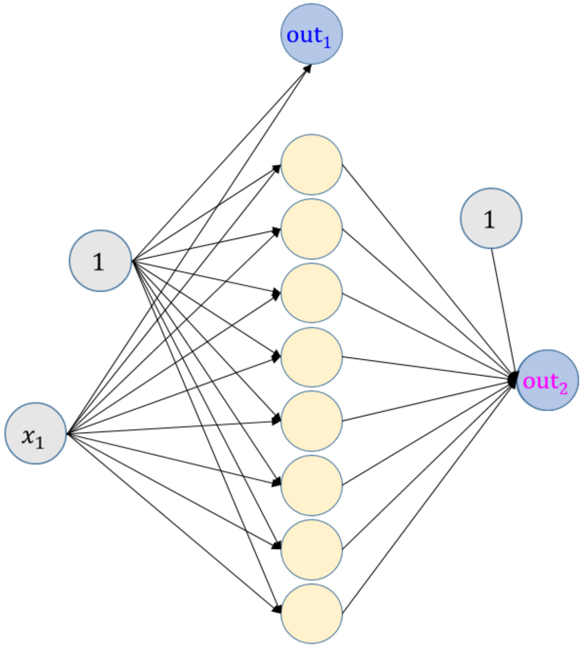


Figure 5.sp: The softplus function compared with the exponential function. Both functions map arbitrary values to positive values.

Fit a probabilistic regression with flexible non-constant variance

$$Y_{X_i} = (Y|X_i) \sim N(\mu_{x_i}, \sigma_x^2)$$





$$\mu_{x_i} = \text{out}_{1_i}$$

$$\sigma_{x_i} = e^{\text{out}_{2_i}}$$

Minimize the negative log-likelihood (NLL):

$$\hat{\mathbf{w}}_{\text{ML}} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \frac{\left(y_i - \mu_{x_i}\right)^2}{2\sigma_{x_i}^2}$$

gradient descent with NLL loss

$$\hat{w}_1,\ \hat{w}_{.2},\ \ldots,\ \hat{w}_{.27}$$

Note: we do not need to know the “ground truth for s” – the likelihood does the job!

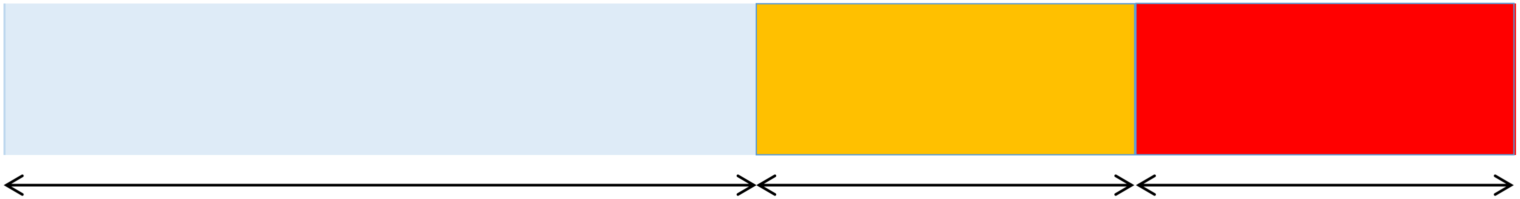
How to evaluate a probabilistic prediction model?

Check prediction quality on NEW data

It's difficult to make predictions, especially about the future

Nils Bohr, physics Nobel price 1922

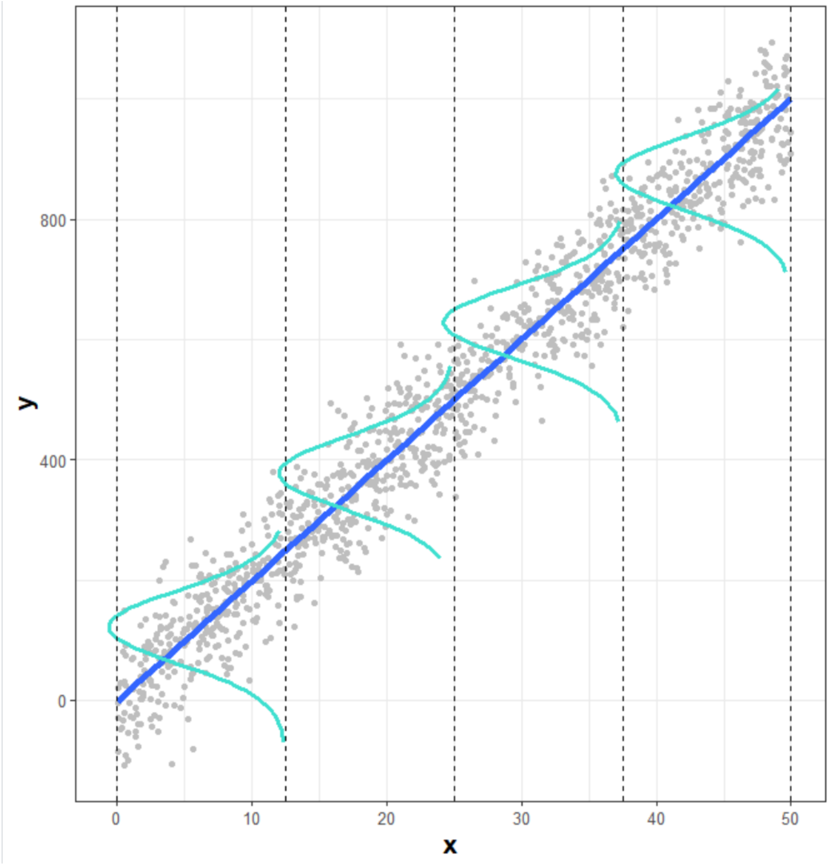
Common data split:



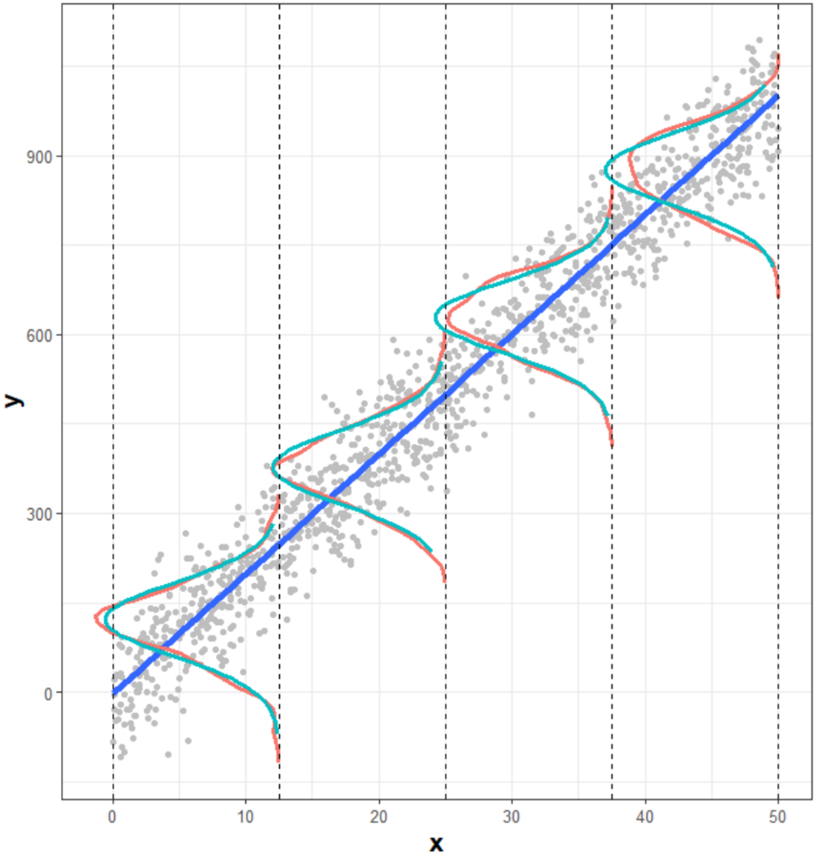
Train-data(50%) Validation-data(25%) Test-data(25%)

Visually: Do predicted and observed outcome distribution match?

Validation data along with predicted outcome distribution (Gauss with const σ)

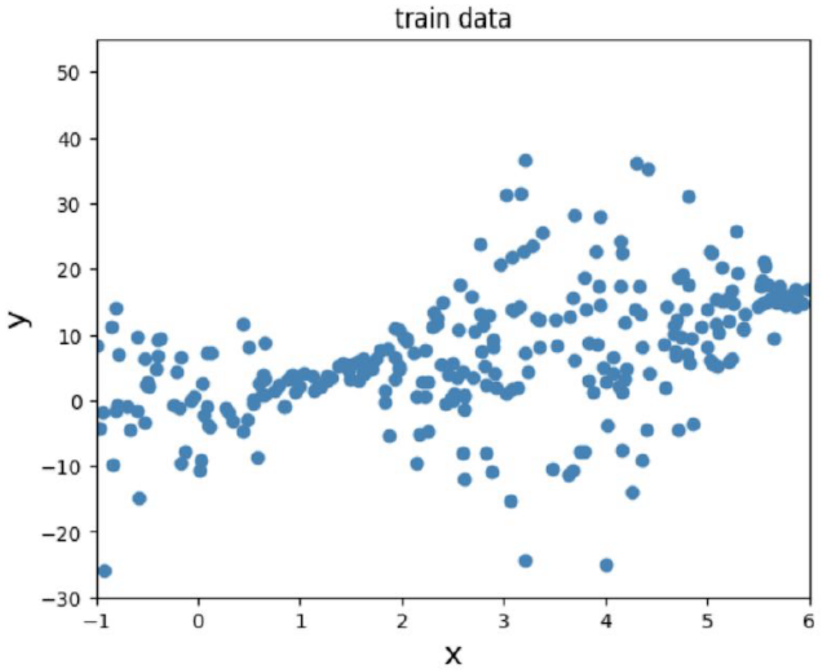


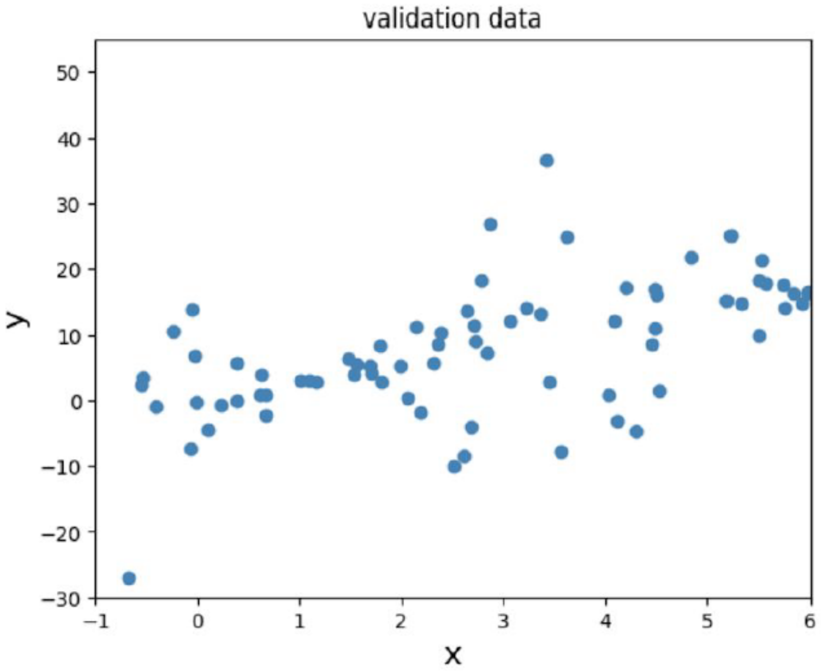
Validation data along with predicted and observed outcome distribution



A large validation data set is needed to ensure underlying assumption:
observed distribution = data generating distribution

Simulate some challenging data for linear regression models

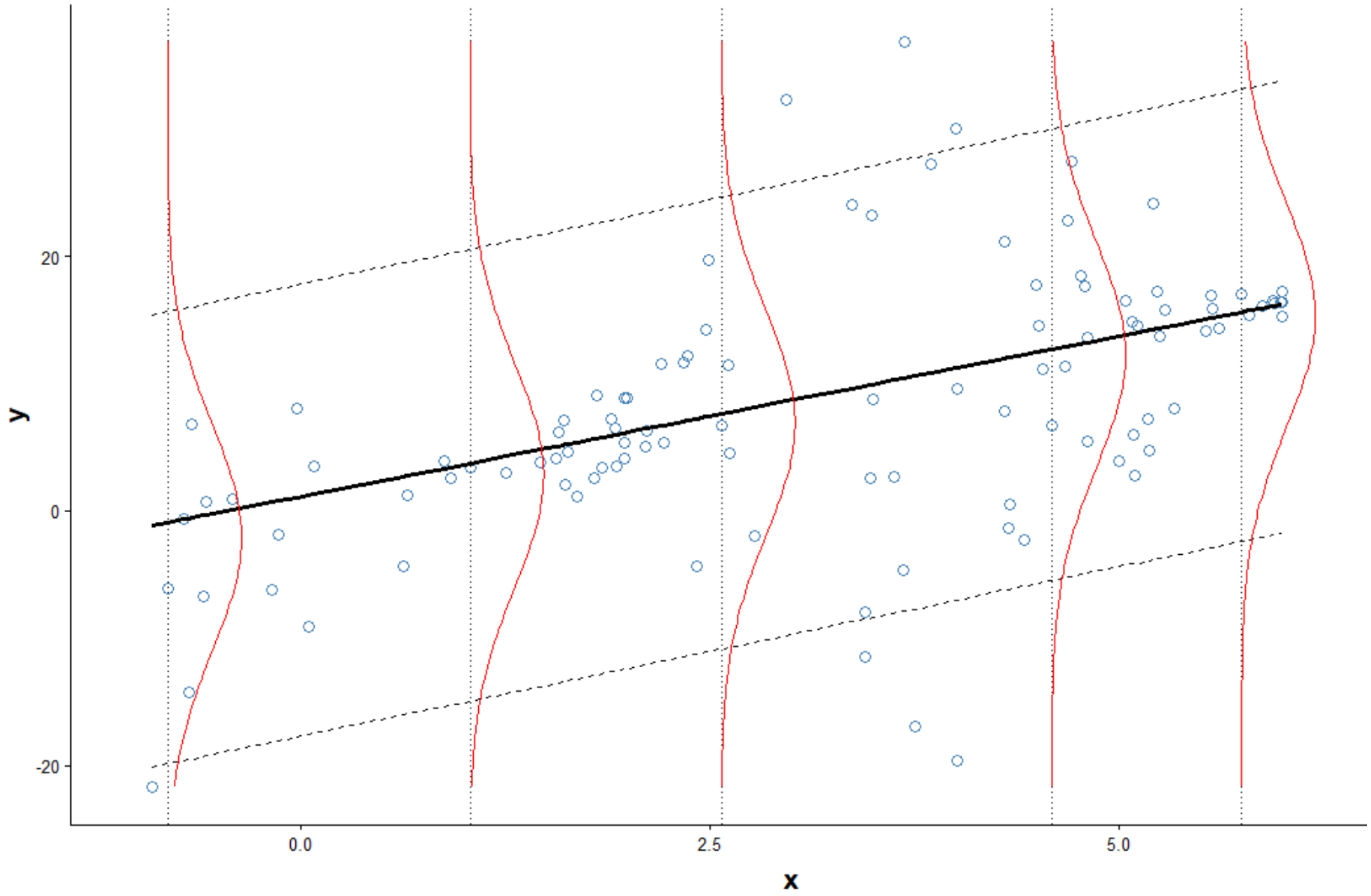




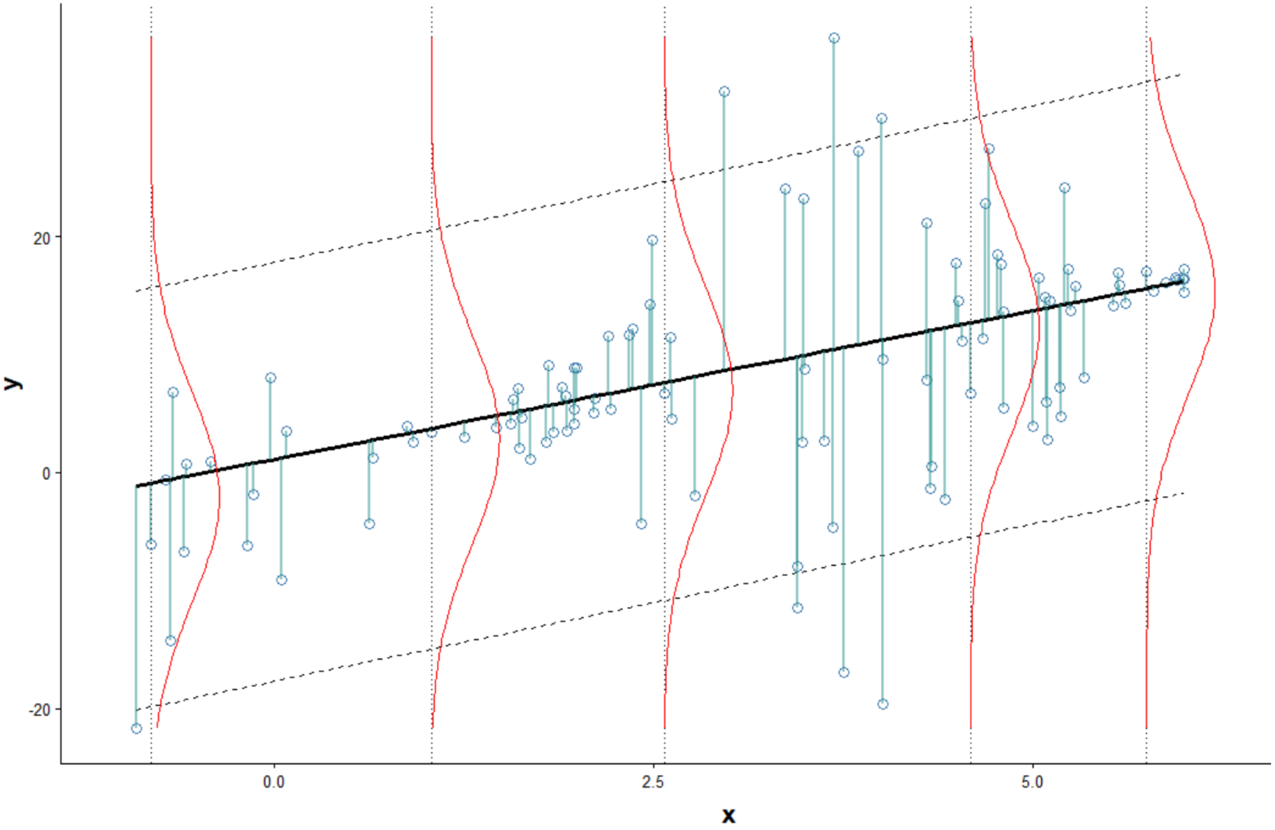
Model_1 (linear regression with constant variance): $(y\,|\,x) \sim N(\mu_x, \sigma^2)$

Model_2 (linear regression with flexible variance): $(y\,|\,x) \sim N(\mu_x, \sigma_x^2)$

Predicted outcome distribution from model_1 (constant σ)



Root mean square error (RMSE) or mean absolute error (MAE)

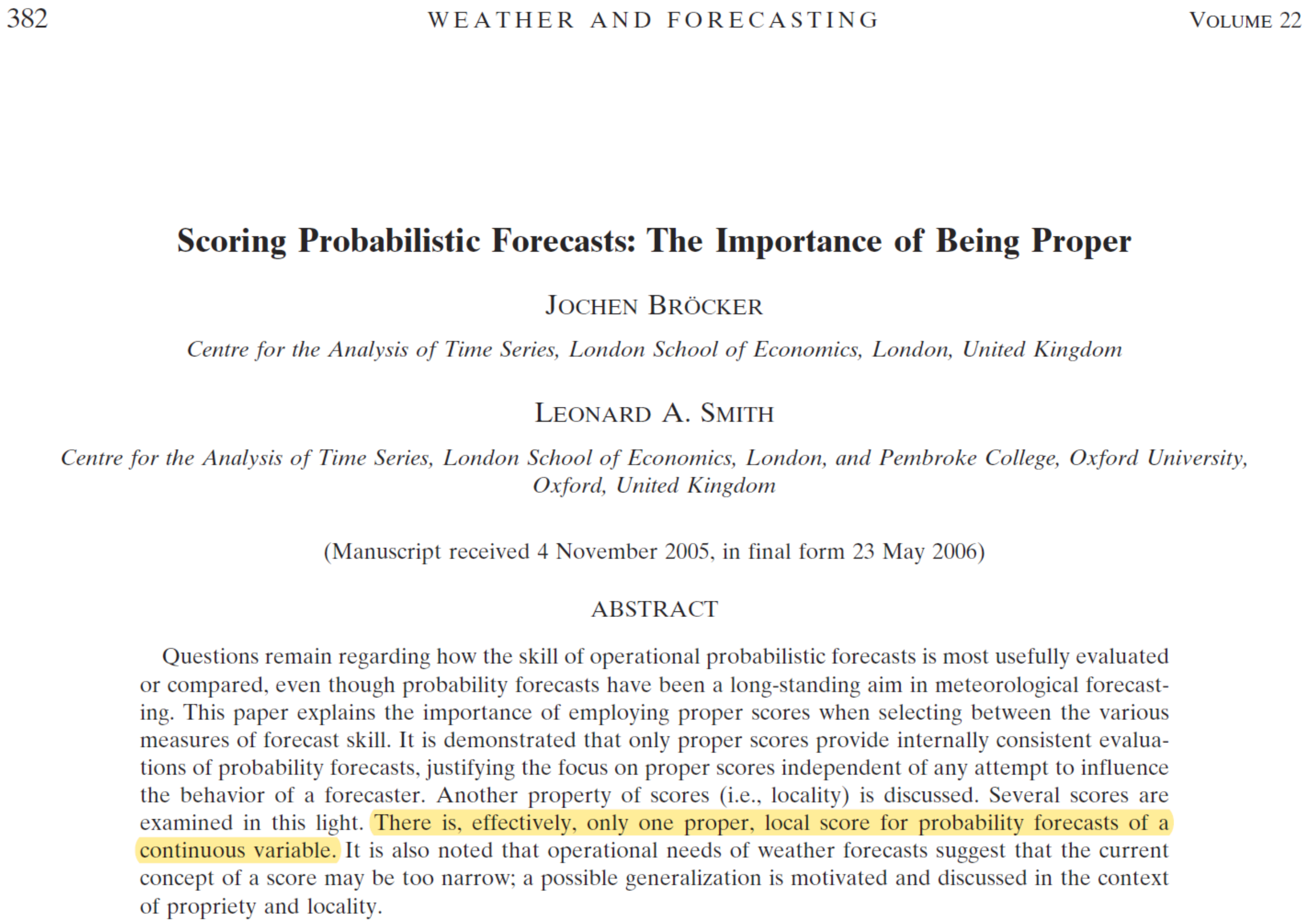


$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_{x_i})^2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{\mu}_{x_i}\right|$$

RMSE and MAE alone do not capture performance for probabilistic models!

Both only depend on the mean (μ) of the CPD, but not on it’s shape or spread (σ) and are not appropriate to evaluate the quality of the predicted distribution of a probabilistic model.



Scoring Probabilistic Forecasts: The Importance of Being Proper

JOCHEN BRÖCKER

Centre for the Analysis of Time Series, London School of Economics, London, United Kingdom

LEONARD A. SMITH

Centre for the Analysis of Time Series, London School of Economics, London, and Pembroke College, Oxford University, Oxford, United Kingdom

(Manuscript received 4 November 2005, in final form 23 May 2006)

ABSTRACT

Questions remain regarding how the skill of operational probabilistic forecasts is most usefully evaluated or compared, even though probability forecasts have been a long-standing aim in meteorological forecasting. This paper explains the importance of employing proper scores when selecting between the various measures of forecast skill. It is demonstrated that only proper scores provide internally consistent evaluations of probability forecasts, justifying the focus on proper scores independent of any attempt to influence the behavior of a forecaster. Another property of scores (i.e., locality) is discussed. Several scores are examined in this light. There is, effectively, only one proper, local score for probability forecasts of a continuous variable. It is also noted that operational needs of weather forecasts suggest that the current concept of a score may be too narrow; a possible generalization is motivated and discussed in the context of propriety and locality.

https://journals.ametsoc.org/doi/full/10.1175/WAF966.1

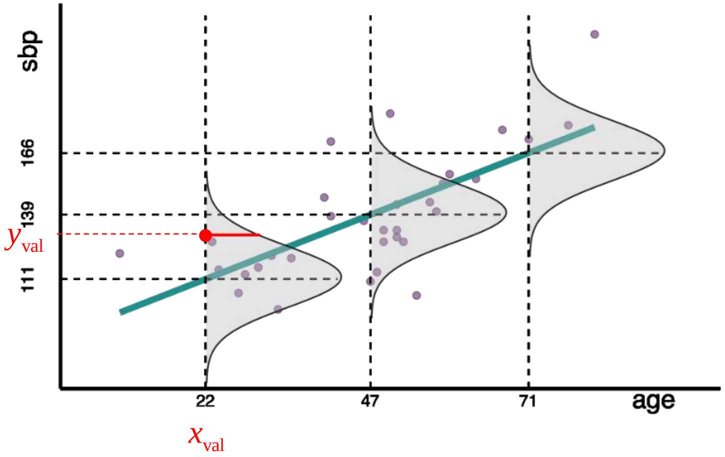
Scores to evaluate probabilistic prediction models

- We need validation data: (x_{val}, y_{val})

- We need predicted outcome distribution, given x: $p_{pred}(y|x)$

- The score *S* takes *one instance* and yields a real number (smaller is better)

$$S(p_{pred}(y|x_{val}), y_{val})$$



Example 1: NLL (aka log-score, ignorance):

$$S_{NLL}\big(p_{pred}(y|x_{val}), y_{val}\big) = -\log(p_{pred}(y_{val}|x_{val}))$$

Example 2: weighted MSE:

$$S_{wMSE}\big(p_{pred}(y|x_{val}), y_{val}\big) = \int (y_{val} - y)^2\, p_{pred}(y|x_{val})\ \mathrm{dy}$$

Empirical loss as average score

- If we use a validation set with n instances (x_{val_i}, y_{val_i}) to evaluate the model, the average score is used as empirical loss:

$$\text{empirical loss } = \frac{1}{n}\sum_{i=1}^{n} S\left(p_{\text{pred}}(y \mid x_{val_i}), y_{val_i}\right)$$

- The empirical loss approximates the expected loss:

$$\text{expected loss} = \int_y S\left(p_{\text{pred}}(y \mid x'), y'\right) \cdot p_{true}(y', x') \; dx' dy'$$

p_{pred} : predicted distribution
p_{true} : data generating distribution

Local scores

A score is local if the predicted distribution is evaluated only at the actual observed outcome of the validation data

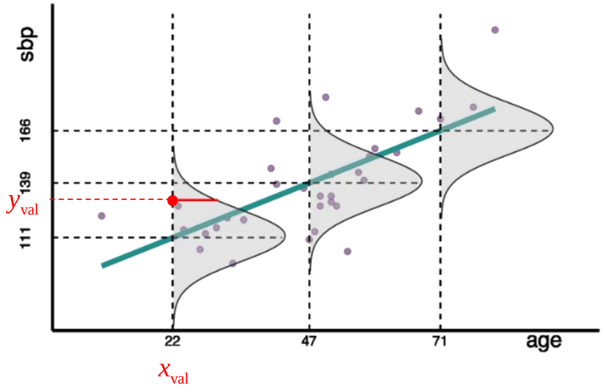
$$S(p_{pred}(y|x_{val}), y_{val}) = S(p_{pred}(y_{val}|x_{val}), y_{val})$$

Example 1: NLL (aka log-score, ignorance):

$$S_{NLL}(p_{pred}(y|x_{val}), y_{val}) = -\log(p_{pred}(y_{val}|x_{val}))$$

Example 2: linear score

$$S_{NL}(p_{pred}(y|x_{val}), y_{val}) = -p_{pred}(y_{val}|x_{val})$$



Proper Scores

For a proper score holds:

The expected value of a *proper score* takes its minimal (optimal) value, if predicted distribution p_{pred} = p_{true} data generating distribution

The expected value of a *strictly* proper score takes its minimal value, *only* if predicted distribution p_{pred} = p_{true} data generating distribution

$\int\int S(p_{true}(y|x'),y')p_{true}(y',x')dy'dx' < \int\int S\big(p_{pred}(y|x'),y'\big)p_{true}(y',x')dy'dx'$ if $p_{pred} \neq p_{true}$

The score with true cdf

The score with predicted cdf

The log-score is strictly proper

To show: $\int\int S(p_{pred}(y|x'),y')p_{true}(y',x')dy'dx' > \int\int S(p_{true}(y|x'),y')p_{true}(y',x')dy'dx'$

$$\int_{x,y} S\left(p_{pred}(y\,|\,x'),y'\right)\cdot p_{true}(y',x')\ \ dx'dy' \ = \int_{x,y} S\left(p_{true}(y\,|\,x'),y'\right)\cdot p_{true}(y',x')\ \ dx'dy' \ +$$

$$\underbrace{\left\{\int_{x,y} S\left(p_{pred}(y\,|\,x'),y'\right)\cdot p_{true}(y',x')\ \ dx'dy' \ - \int_{x,y} S\left(p_{true}(y\,|\,x'),y'\right)\cdot p_{true}(y',x')\ \ dx'dy'\right\}}_{>0 \quad \text{for strictly proper scores } S}$$

Proof that NLL is strictly proper $\quad S_{NLL}\left(p(y\,|\,x_{val}),y_{val}\right) = -\log\left(p(y_{val}\,|\,x_{val})\right)$

$$\int S_{\mathrm{NLL}}\left(p_{\mathrm{pred}}(y\,|\,x'),y'\right)\cdot p_{\mathrm{true}}(y',x')\ \ dx'dy' \ - \int S_{\mathrm{NLL}}\left(p_{\mathrm{true}}(y\,|\,x'),y'\right)\cdot p_{\mathrm{true}}(y',x')\ \ dx'dy'$$

$$= \int -\log\left(p_{\mathrm{pred}}(y'\,|\,x')\right)\cdot \underbrace{p_{\mathrm{true}}(y',x')}_{=p_{true}(y'|x')\cdot p(x')}\ \ dx'dy' \ - \int -\log\left(p_{\mathrm{true}}(y'\,|\,x')\right)\cdot \underbrace{p_{\mathrm{true}}(y',x')}_{=p_{true}(y'|x')\cdot p(x')}\ \ dx'dy'$$

$$= \int \log\left(\frac{p_{\mathrm{true}}(y'\,|\,x')}{p_{\mathrm{pred}}(y'\,|\,x')}\right)\cdot p_{\mathrm{true}}(y'\,|\,x')dy'\,p_{\mathrm{true}}(x')\ \ dx' \ = \mathrm{KL}\left(p_{\mathrm{true}}(\cdot\,|\,x);\,p_{\mathrm{pred}}(\cdot\,|\,x)\right) > 0 \qquad \forall\ p_{pred} \ \neq \ p_{true}$$

The linear score is not proper

The linear score is not proper, meaning p_{true} does not yield the best expected score.

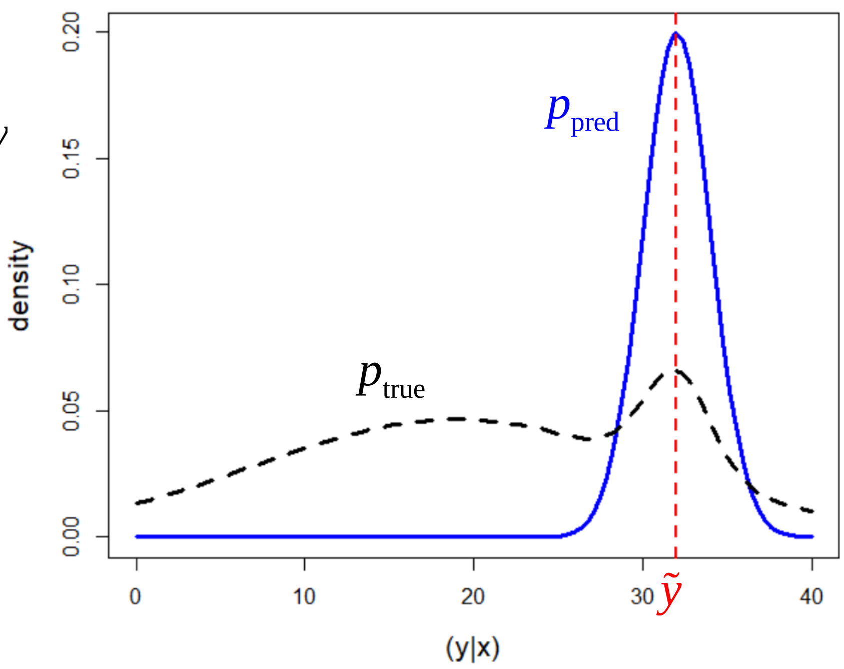
$$S_{\text{lin}}\left(p(y|x_{val}), y_{val}\right) = -p(y_{val}|x_{val})$$

$$E_{p_{\text{true}}}(S_{p_{\text{pred}}}) = \int_y S\left(p_{true}(y|x'), y'\right)\cdot p_{true}(y',x')\ \ dx'dy$$

$$= \int_y -p_{true}(y'|x')\cdot p_{true}(y',x')\ \ dx'dy'$$

If p_{true} is not constant, then there is a $\tilde{y}$ higher than mean probability:

$$-p_{true}(\tilde{y}|x') < E_{p_{\text{true}}}(S_{p_{\text{true}}})$$



Proof:

Construct p_{pred} that scores better than p_{true}: $p_{\text{pred}}(\tilde{y}|x') = \frac{1}{\sigma}\cdot\text{kernel}\left(\frac{y'-\tilde{y}}{\sigma}\right)$

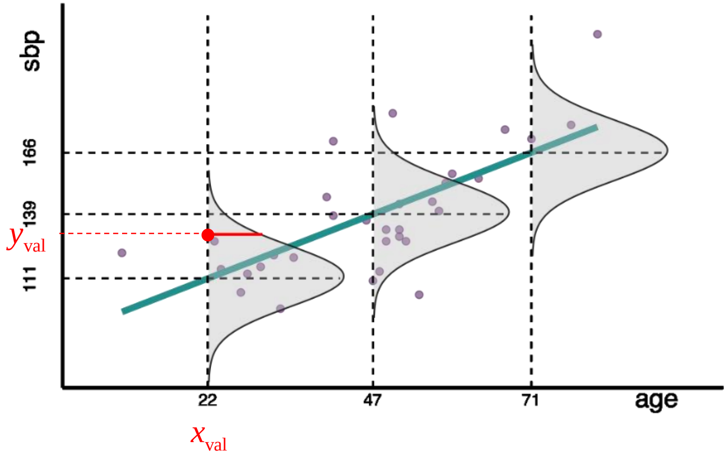
$$E_{p_{\text{true}}}(S_{p_{\text{pred}}}) = \int_y -p_{\text{pred}}(y'|x')\cdot p_{true}(y',x')\ \ dx'dy' \ \rightarrow\ -p_{\text{true}}(\tilde{y}|x')\ <\ E_{p_{\text{true}}}(S_{p_{\text{true}}})$$

The uniqueness of the log-score

It is provable that **the log-score** is the **only** smooth, proper and local score for continuous variables

(Bernardo, J. M., 1979: Expected information as expected utility. *Ann. Stat.*, **7,** 686–690)

$$S_{\mathrm{NLL}}\left(p(y\,|\,x_{val}), y_{val}\right) = -\log\left(p(y_{val}\,|\,x_{val})\right)$$



Prominent Scores for binary classifiers

Definition 9.9 (Scoring rules for binary predictions) Let $Y \sim \mathrm{B}(\pi)$ be the predictive distribution for a binary event, i.e.

$$f(y) = \begin{cases} \pi & \text{for } y = 1, \\ 1 - \pi & \text{for } y = 0. \end{cases}$$

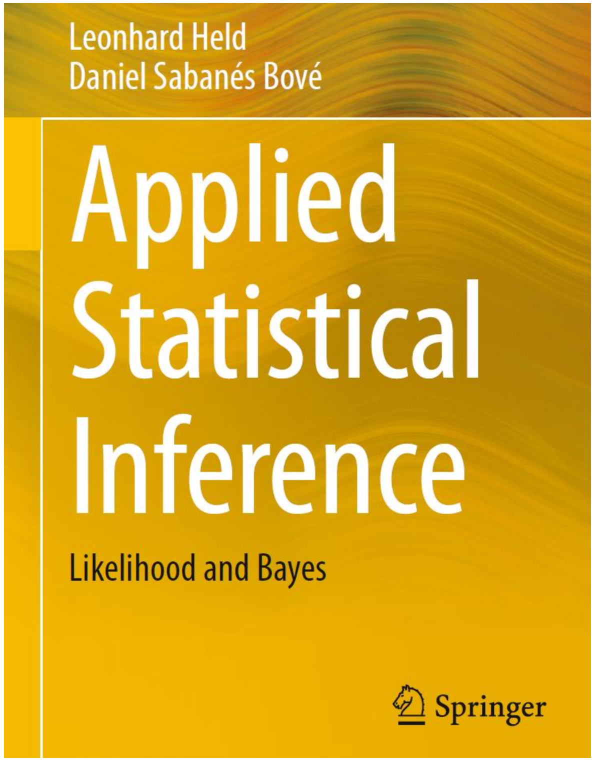
The *Brier score* BS, the *absolute score* AS and the *logarithmic score* LS are defined as

Strictly proper: $\mathrm{BS}\big(f(y), y_o\big) = (y_o - \pi)^2,$

Not proper: $\mathrm{AS}\big(f(y), y_o\big) = |y_o - \pi|$ and

Strictly proper: $\mathrm{LS}\big(f(y), y_o\big) = -\log f(y_o),$

respectively.



Leonhard Held
Daniel Sabanés Bové

Applied Statistical Inference

Likelihood and Bayes

Springer

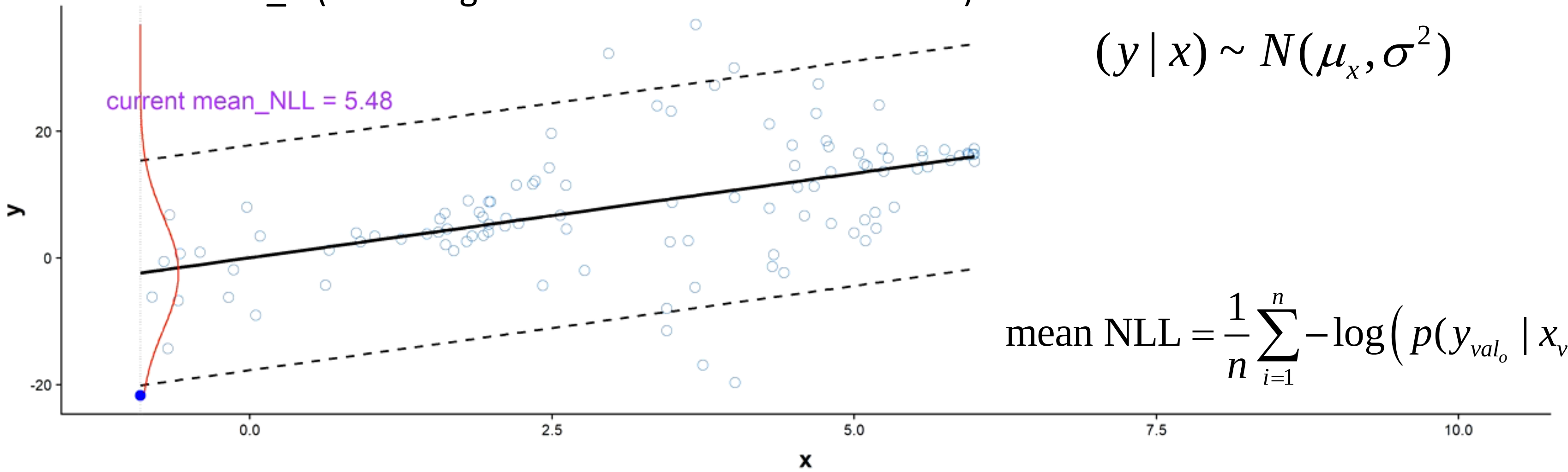
Remark: For binary classification, the log score is not the only strictly proper score.

NLL as general cure-all in probabilistic modeling

- Maximize likelihood ↔ minimize negative log-likelihood (NLL)
- The log-score (NLL) is strictly proper score for regression.
- The log-score (NLL) is also strictly proper for classification models.
- To train a probabilistic model: minimize NLL!
- To evaluate or compare probabilistic models: use the validation NLL!

Use validation NLL to compare probabilistic models

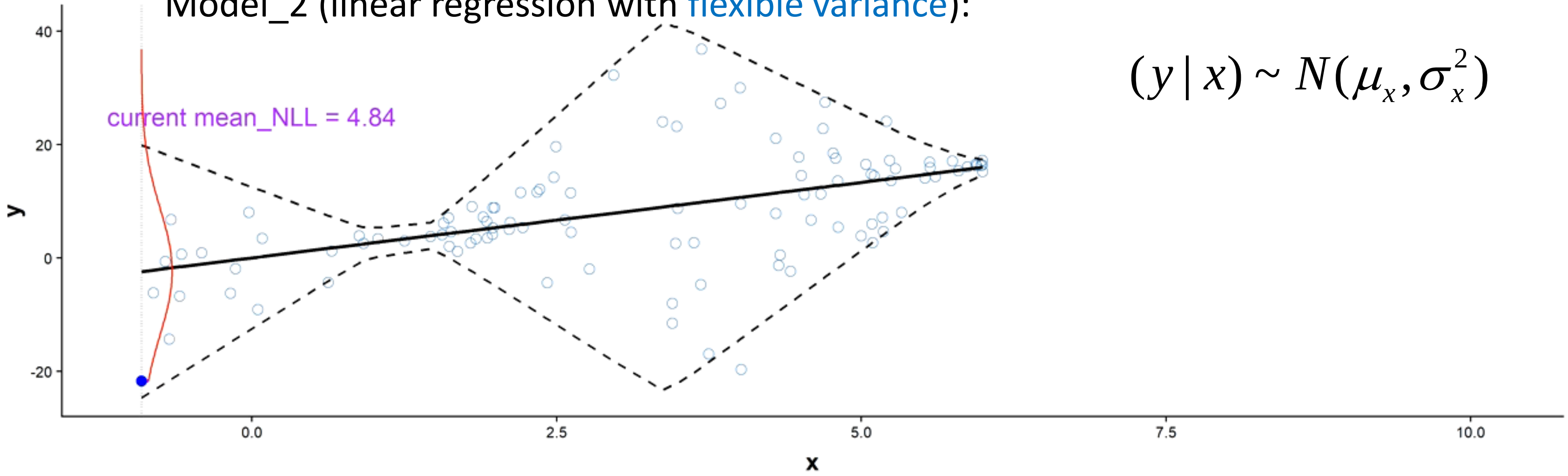
Model_1 (linear regression with constant variance):



$$(y \mid x) \sim N(\mu_x, \sigma^2)$$

$$\text{mean NLL} = \frac{1}{n}\sum_{i=1}^{n} -\log\left(p(y_{val_o} \mid x_{val_i})\right)$$

Model_2 (linear regression with flexible variance):



$$(y \mid x) \sim N(\mu_x, \sigma_x^2)$$

How to develop a highly performant probabilistic model for count data?

Probabilistic models for count data

Goal: Probabilistic model for deer activity conditioned on the time (in day and year).

wild	year	time	daytime	weekday
0	2002.0	0.000000	night.am	Sunday
0	2002.0	0.020833	night.am	Sunday
...	....	....	....	...
1	2002.0	0.208333	night.am	Sunday
0	2002.0	0.229167	pre.sunrise.am	Sunday
0	2002.0	0.270833	pre.sunrise.am	Sunday

The columns have the following meaning:

wild: the number deers killed in a road accident in Bavaria

year: the year (from 2002 to 2009 in the training set from 2010 to 2011 for the test set)

time: the number of days to the first event. These numbers are measured in fractions of a day.

Data on deer related car accidents in the years 2002 until 2011 in Bavaria, Germany.
Target variable (wild): use number of deers killed during 30 minute period as surrogate

Modeling count data

Goal:
Predict CPD for y=#deers-killed-in-30min, given x (time and derived variables).

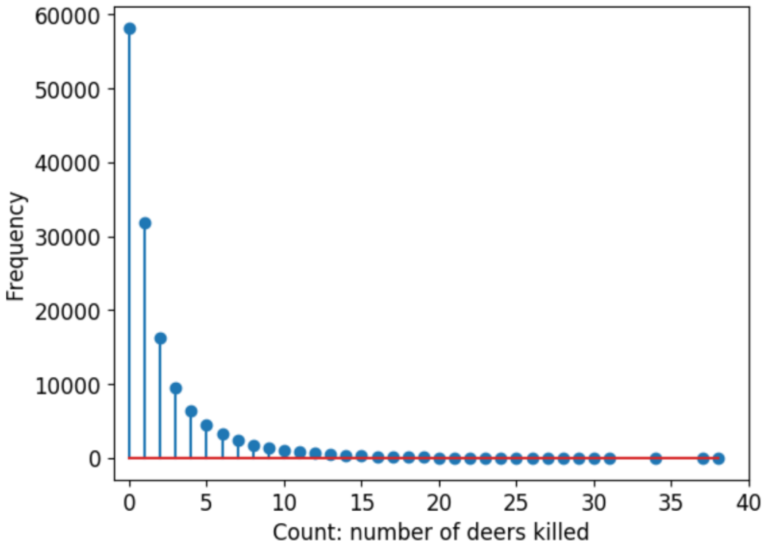
Possible CPD models:

$$(y \mid x) \sim \mathrm{Pois}(\lambda_x)$$

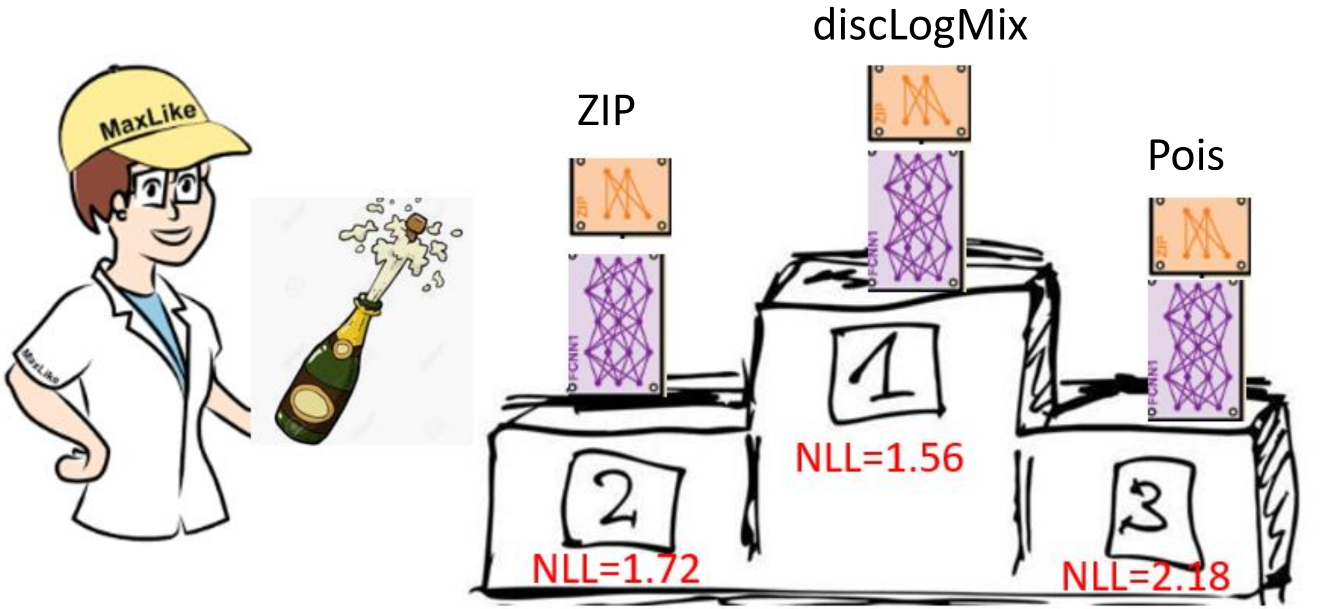
$$(y \mid x) \sim \mathrm{ZIP}({}^{z}p_x, \lambda_x)$$

$$(y \mid x) \sim \mathrm{discretizedLogisticMix}({}^{1}p_x, {}^{2}p_x\, {}^{3}p_x, {}^{1}\mu_x, {}^{2}\mu_x, {}^{3}\mu_x, {}^{1}\sigma_x, {}^{2}\sigma_x, {}^{3}\sigma_x)$$

Marginal distribution of deer killed in 30min



Validation NLL allows to rank different probabilistic models



Take home messages

- A probabilistic model predicts for each input a whole outcome CPD

- Use the NLL for training, evaluating and comparing probabilistic models