Model Evaluation

Machine Learning for Behavioral Data

March 18, 2025

Today's Topic

Week	Lecture/Lab
1	Introduction
2	Data Exploration
3	Regression
4	Classification
5	**Model Evaluation**
6	Time Series Prediction
7	Time Series Prediction
8	Time Series Prediction

Complete pipeline for one use case:

- **Data exploration**
- **Prediction**
- **Model evaluation**

Getting ready for today's lecture…

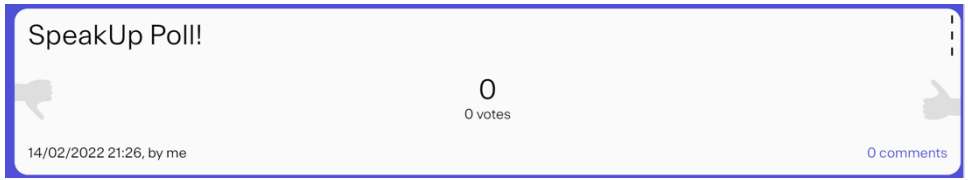
- **If not done yet**: clone the repository containing the Jupyter notebook and data for today's lecture into your Noto workspace
- SpeakUp room for today's lecture:

https://go.epfl.ch/speakup-mlbd2025

Short quiz about the past…

Which of the following is an example of an ensemble method?

a) Decision Tree
b) k-Nearest Neighbor
c) Logistic Regression
d) Random Forest



https://go.epfl.ch/speakup-mlbd2025

Short quiz about the past…

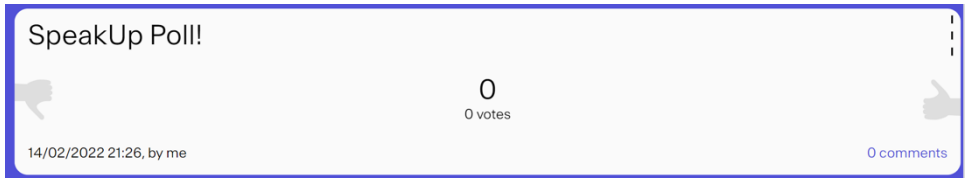
We are going to predict the species of an Iris Flower. The dependent variable (species) contains three possible values: Setosa, Versicolor, and Virginica. Which classification methods are appropriate for this task?

a) Decision Tree

b) k-Nearest Neighbor

c) Logistic Regression

d) Random Forest

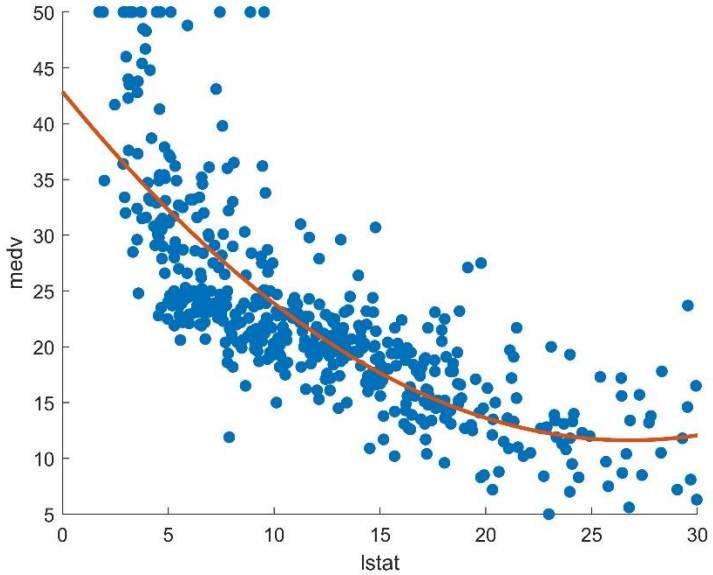


https://go.epfl.ch/speakup-mlbd2025

Today

- **Model Assessment and Selection**
- Reporting of Results

How good is my model?



$$medv = 42.86 - 2.33 \cdot lstat + 0.04 \cdot lstat^2$$

$$R^2 = 0.64$$

SpeakUp Chat!

0

0 votes

14/02/2022 21:25, by me

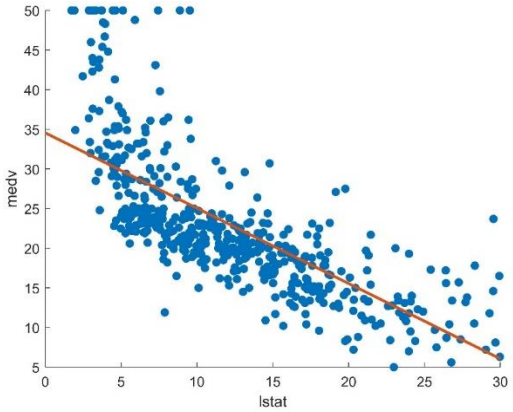
0 comments

lstat: Percentage of lower status of the population

medv: Median value of owner-occupied homes in $1000s

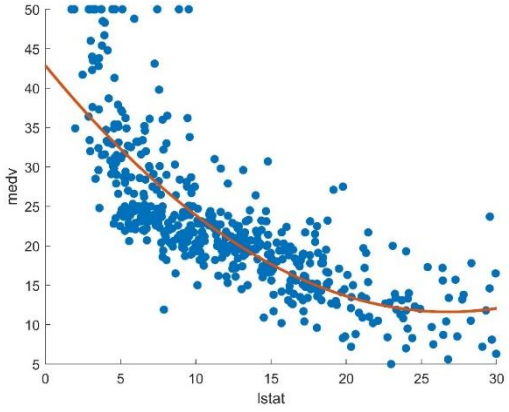
https://go.epfl.ch/speakup-mlbd2025

Which model is better?



$$medv = 34.55 - 0.95 \cdot lstat$$

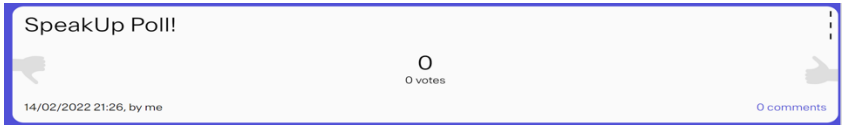
$$R^2 = 0.54$$



$$medv = 42.86 - 2.33 \cdot lstat + 0.04 \cdot lstat^2$$

$$R^2 = 0.64$$

a) The left model
b) The right model
c) I need more information



https://go.epfl.ch/speakup-mlbd2025

Model Evaluation

- **Model assessment**: having chosen a final model, estimating its prediction error on new data (*generalization*).

Model Evaluation

- **Model assessment**: having chosen a final model, estimating its prediction error on new data (*generalization*).
- **Model selection:** estimating the performance of different models in order to choose the best one.

Theory: Notation

- We are given a sample data set:

$$T = \{(y_n, \boldsymbol{x}_n)\} \text{ with } n = 1, \dots N$$

- The sample data set T has been drawn from an (**unknown**) underlying data model D with image $\mathrm{X} \times \Upsilon$:

$$(y_n, \boldsymbol{x}_n)\ i.i.d. \sim D$$

- We have learnt a model f for predicting y based on $\boldsymbol{x}$. How good is f?

Theory: Expected Loss

$$Err_D(f) = \mathrm{E}_D[L(y, f(\boldsymbol{x}))]$$

where y and $\boldsymbol{x}$ are randomly drawn from D, i.e. $Err_D(f)$ is the *expected* error of f over all samples chosen according to D. $Err_D(f)$ is denoted as *expected/true risk/loss*.

Theory: Expected Loss

$$Err_D(f) = \mathrm{E}_D[L(y, f(\boldsymbol{x}))]$$

where y and $\boldsymbol{x}$ are randomly drawn from D, i.e. $Err_D(f)$ is the *expected* error of f over all samples chosen according to D. $Err_D(f)$ is denoted as *expected/true risk/loss*.

- We cannot compute $Err_D(f)$, since we don't know D.
- We are, however, given a sample data set T, drawn from D. We can use T to approximate the expected loss.

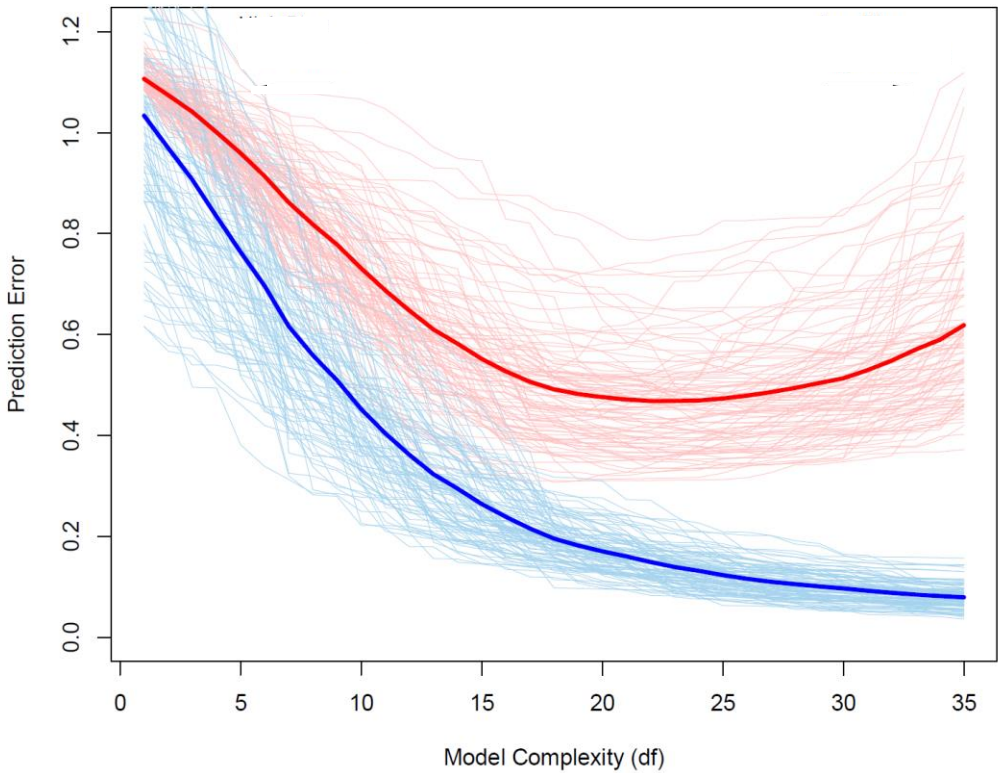
Theory: Training Error

- First idea: we compute the error of f on T to *empirically* approximate $Err_D(f)$:

$$\overline{err}(f) = \frac{1}{|T|} \sum_{(x_n, y_n) \in T} L(y_n, f(x_n))$$

- We call $\overline{err}(f)$ the *empirical error* (or simply the *training error*).

Training error underestimates expected loss



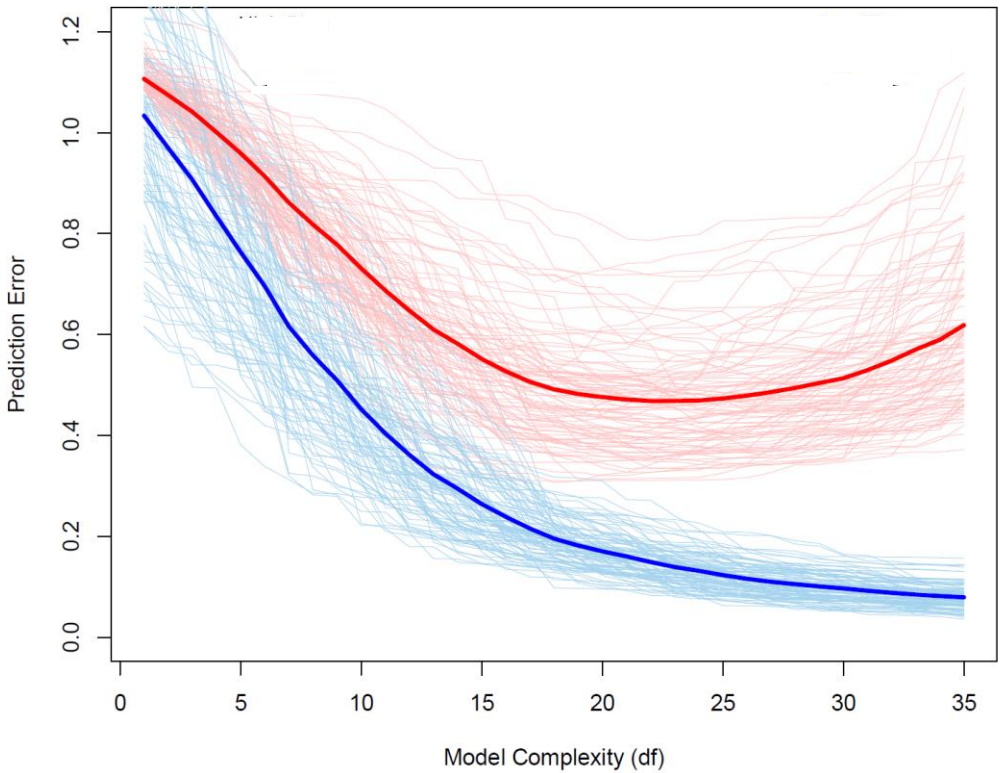
Err

$\overline{err}$

- Using synthetic data, we can compute the expected loss (we **know** D).

[Illustration from: Elements of Statistical Learning]

Training error underestimates expected loss



Err

$\overline{err}$

- Using synthetic data, we can compute the expected loss (we **know** D).
- The training error underestimates the expected loss

We need a better solution to estimate Err_D

[Illustration from: Elements of Statistical Learning]

Model Assessment & Selection

1. **Train, Validate, Test**
2. Resampling Methods
3. Information Criteria

Fixed assignment

- We randomly split the data into a training data set and a test data set (e.g., 80/20 split or 70/30 split)

- *Training set*: used for fitting the model
- *Test set*: assessment of the generalization error of the model

Test Error

- We randomly split the data set T into a training data set and a test data set (e.g., 80/20 split or 70/30 split)

$$Err_{test}(f_{train}) = \frac{1}{|T_{test}|} \sum_{(x_n, y_n) \in T_{test}} L(y_n, f_{train}(x_n))$$

Fixed assignment: Selection & Generalization

- We randomly split the data set T into a training data set and a test data set (e.g., 80/20 split or 70/30 split)

- We further (randomly) split the training data set T_{train} and reserve a part of it for validation (e.g., 80/20 split)

Fixed assignment: Selection & Generalization

- *Training set*: used for fitting the models
- *Validation set*: used to estimate the prediction error for model selection (e.g., hyperparameter tuning)
- *Test set*: assessment of the generalization error of the chosen model

Inefficient use of data

- Not very efficient use of data (requires a large amount of data)
- How large? Depends on
  - Signal to noise ratio of data set
  - Complexity of the models we want to fit

Assessment & Selection

1. Train, Validate, Test
2. **Resampling Methods**
   - **Cross Validation**
   - Bootstrapping
3. Information Criteria

K-Fold Cross-Validation

- Randomly divide data into K parts (folds)
- For $k = 1, \ldots, K$
  - Fit the model to the other $K - 1$ folds $\{1, \ldots, K\} \setminus \{k\}$
  - Compute the prediction error on k
- Combine the K estimates of prediction error

$K = 5$

$k = 1$	Test	Train	Train	Train	Train
$k = 2$	Train	Test	Train	Train	Train
$k = 3$	Train	Train	Test	Train	Train
$k = 4$	Train	Train	Train	Test	Train
$k = 5$	Train	Train	Train	Train	Test

Combining the K estimates of prediction error

$K = 5$

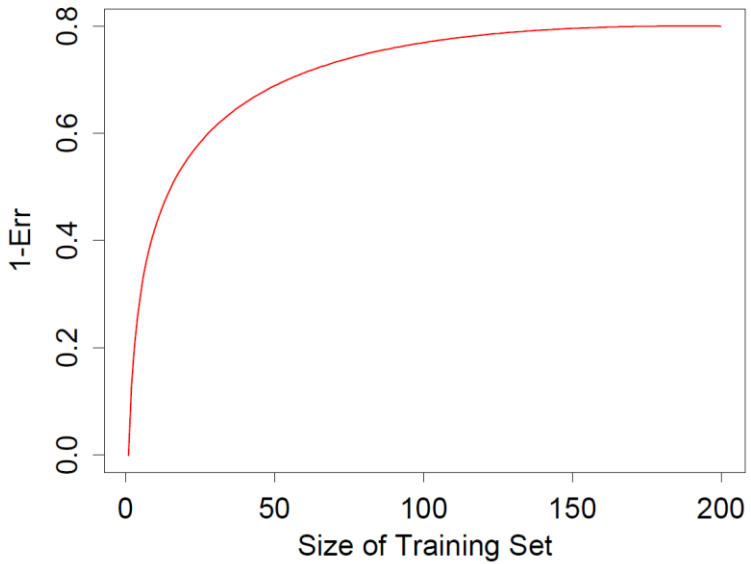
k = 1	Test	Train	Train	Train	Train
k = 2	Train	Test	Train	Train	Train
k = 3	Train	Train	Test	Train	Train
k = 4	Train	Train	Train	Test	Train
k = 5	Train	Train	Train	Train	Test

$$PE^{CV} = \frac{1}{K}\sum_{k \le K} PE_k = \sum_{i=1}^{N} L(y_i, f^{-\kappa(i)}(x_i))$$

Leave-one-out cross validation

- $K = N$, i.e. each fold just contains one sample
  - For a sample i, we train f on all samples but i and then use f to predict on i
  - Can have high variance (the N training sets are very similar to each other, the test sets very different)
  - High computational burden (f needs to be fit N times)
  - Avoids issues with fold selection

How do we choose K?



- We don’t know the learning curve of our model
- In practice: 5-fold and 10-fold cross validation are recommended as a good compromise

Variants of cross validation

- Flat
  - Each sample has equal chance of being placed into each fold
- Stratified
  - Fold selection is biased such that some variable is equally represented in each fold
    - This can be the variable we are trying to predict (y)
    - This can be a variable that is thought to be an important context

Stratification

- Other levels of stratification are possible (e.g., school, learning environment, demographic information)
- Consideration:
  - Where will the model be used (e.g., what is the potential application)?
  - Make sure you stratify at this level
- Can also be combined (e.g., we can stratify by predicted variable y and by user)
- Stratification can also be used in the fixed assignment (train – validate – test) setting

Example: Classification Problem

- Given: classification problem, synthetic data set with
  - **binary** class label y (balanced)
  - 5000 features (*independent* of class labels y)
- What is the expected loss of any classifier f on this data set?

a) 10%
b) 30%
c) 50%
d) 70%
e) 90%

https://go.epfl.ch/speakup-mlbd2025

Example: Suggested Procedure

1) Select the 100 features that have the highest correlation with the class labels y
2) Use a 1-nearest neighbor classifier based on just these 100 selected features
3) Use cross validation to estimate the prediction error of the classifier

➡ Over 50 simulations of this procedure: $PE^{CV} = 0.03$

Be careful!

- Multistep modeling procedure
  - Cross-validation must be applied to the **entire sequence** of modeling steps. In particular, **samples must be "left out" before any selection or filtering steps** are applied.
  - Exception: **unsupervised** cleaning or screening steps can be done before samples are "left out".
- This of course holds also for the train-validate-test case!

Your Turn – Model Assessment

- In your student notebook, we provide examples on Train-Test Split and Cross Validation model evaluation.
- Your task:
  - Run the Train-Test Split and the Cross Validation model evaluations
  - What is the difference in accuracy/AUC between the two methods? Where does this difference come from?

Assessment & Selection

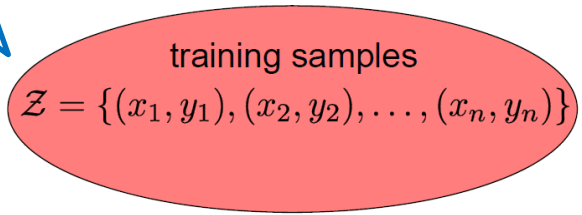
1. Train, Validate, Test
2. **Resampling Methods**
   - Cross Validation
   - **Bootstrapping**
3. Information Criteria

Bootstrapping

- General tool for assessing statistical accuracy
- Basic idea:
  - Given: training data set $\boldsymbol{Z} = \{z_1, \ldots, z_N\}$ with $z_i = (\boldsymbol{x_i}, y_i)$, where N is the number of samples
  - Randomly draw N pairs $(\boldsymbol{x_i}, y_i)$ with replacement from the training data set
  - Repeat B times (e.g., $B = 100$) -> B bootstrap data sets
  - Fit model to each bootstrap data set, observe behavior across bootstrap data sets

Bootstrapping

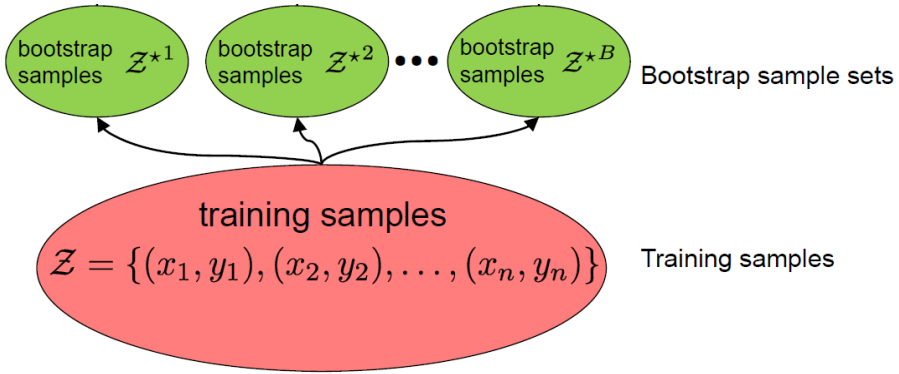
- We are interested in $S(Z)$
- $S(Z)$ can be any quantity computed from the data Z, e.g., the prediction at some input point
- We want to compute an aspect of the distribution of $S(Z)$ (e.g., the variance)



Training samples

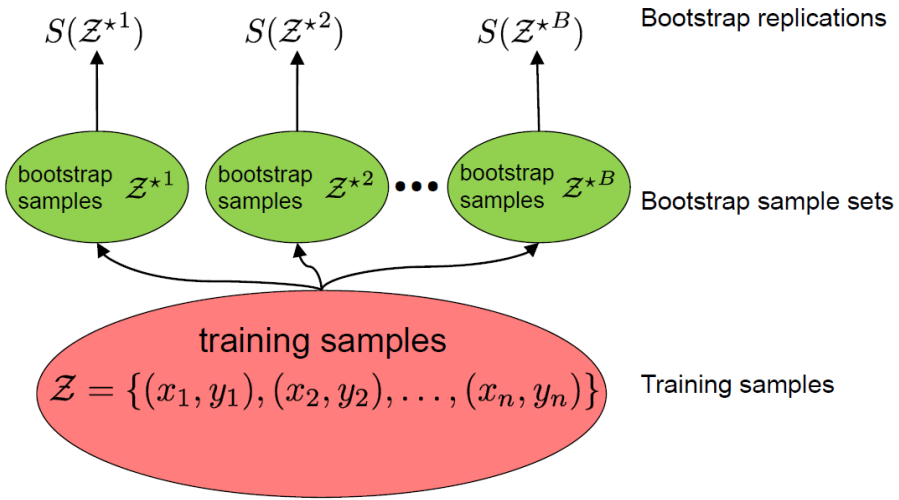
[Illustration credit: ML lecture, ETH Zurich]

Bootstrapping



[Illustration credit: ML lecture, ETH Zurich]

Bootstrapping



[Illustration credit: ML lecture, ETH Zurich]

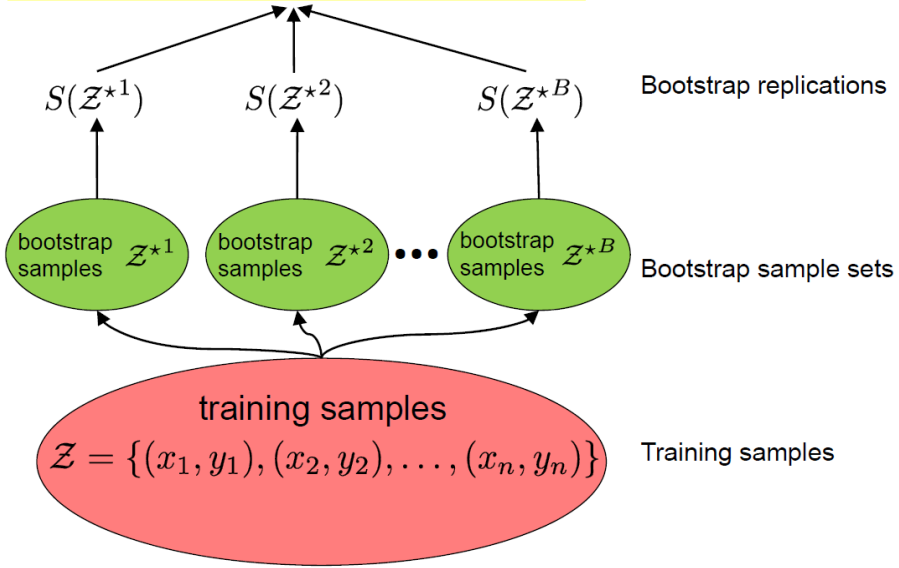
Bootstrapping

$$\overline{S} = \frac{1}{B}\sum_{b \le B} S(\mathcal{Z}^{\star b})$$

Replication mean

$$\sigma^2(S) = \frac{1}{B-1}\sum_{b \le B}\left(S(\mathcal{Z}^{\star b}) - \overline{S}\right)^2$$

Replication variance



Bootstrap replications

Bootstrap sample sets

Training samples

[Illustration credit: ML lecture, ETH Zurich]

Bootstrapping for prediction

- Idea: leave-one-out bootstrap

$$\widehat{Err}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, f^{*b}(\boldsymbol{x_i}))$$

where C^{-i} denotes the set of indices of bootstrap samples b, that do **not** contain observation i

.632 bootstrap error

- It can be shown that the average number of distinct samples in each bootstrap sample b is about $0.632 \cdot N$
- $\widehat{Err}^{(1)}$ will therefore tend to overestimate the true loss
- Solution:

$$\widehat{Err}^{(.632)} = .368 \cdot \overline{err} + .632 \cdot \widehat{Err}^{(1)}$$

Assessment & Selection

1. Train, Validate, Test
2. Resampling Methods
3. **Information Criteria**
   - AIC
   - BIC

Information Criteria

- The training error is an overly optimistic estimate of the *expected loss*
- Information criteria try to estimate the *optimism* in the training error and correct for it

Akaike Information Criterion (AIC)

$$AIC = -\frac{2}{N} \cdot loglik + 2 \cdot \frac{d}{N}$$

- d is the number of parameters of our model f
- $loglik$ is the log-likelihood (logarithm of the likelihood) of the sample data T given our model f
- N is the number of samples in the data set, i.e. $|T| = N$

Bayesian Information Criterion (BIC)

$$BIC = -2 \cdot loglik + \log(N) \cdot d$$

- d is the number of parameters of our model f
- $loglik$ is the log-likelihood (logarithm of the likelihood) of the sample data T given our model f
- N is the number of samples in the data set, i.e. $|T| = N$
- BIC penalizes complex models more heavily than AIC

Information Criteria - Considerations

- Applicable for models where the fitting is done under a maximum likelihood setting
- *Effective* number of parameters:
  - Choosing the best fitting model with d features, can result into more than d parameters being fit (e.g., regularization)
  - Determining the *effective* number of parameters can be difficult for more complex models (e.g., trees)

What method should we choose?

1. Train, Validate, Test
2. Resampling Methods
   - Cross Validation
   - Bootstrapping
3. Information Criteria
   - AIC
   - BIC

What method should we choose?

Depends on the purpose of the model:

- Interpretation: information criteria are useful for unsupervised cases or when the model is built for interpretation purposes (e.g., which features of the regression model explain the data better?)
- Prediction:
  - All presented methods (train-test, cross validation, bootstrap) are reasonable choices
  - We can combine them freely to do model selection and assessment
  - In practice: while bootstrap provides accurate estimates of test error, it also requires a high amount of extra work

Example Combinations

Your Turn – Model Selection & Assessment

- In your student notebook, we provide an incorrect example to tune to hyperparameters of the model and then evaluate the prediction error of that model in terms of accuracy of AUC
- Your task:
  - Explain why it is incorrect.
  - Describe how it could be fixed.

Today

- Model Selection and Assessment
- **Reporting of Results**

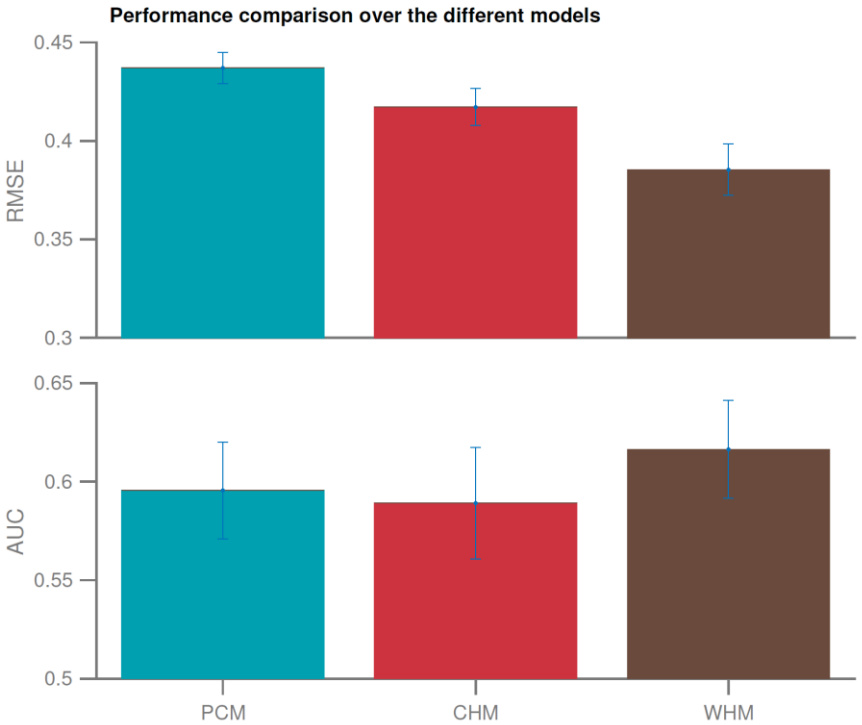
Back to the question: is my model any good?

- A friend tells you that he has found a great model for early prediction of drop out in online courses
- She gives you the following information:
  - The data set stems from N students taking course c
  - 30% of the students dropped the course
  - She has evaluated her model using cross validation
  - The accuracy of the model is 0.8 and the AUC is 0.83

Provide comparisons!

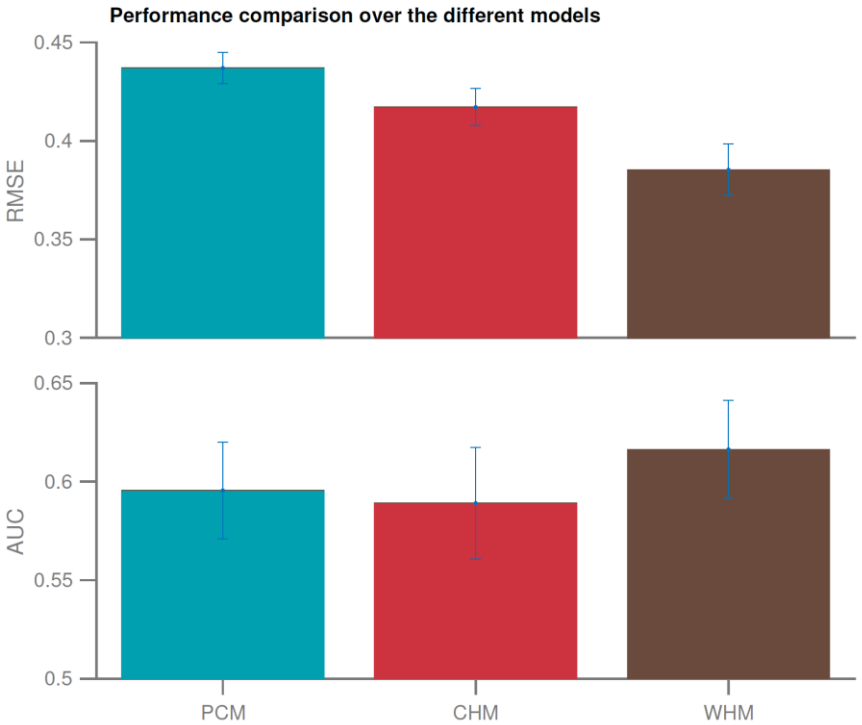
- Reporting performance metrics for a model f on a data set T is not enough: achievable performance (what is good?) depends on the data set!
- We need to provide comparisons to **baseline** models:
  - **Minimum** baseline: compare to a random model
  - Additional comparisons, depending on the goal:
    - Suggestion of a new type of model structure for a problem: need to compare to previously suggested structures
    - Suggestion of new features for a problem: need to compare performance between a model using and not using the new features

Quantification of Uncertainty



[Figure from Käser et al., LAK 2017]

Quantification of Uncertainty



- We usually report the *mean* prediction error over all samples
- We should quantify the uncertainty of the performance metric!
- Uncertainty can be computed across samples where applicable (or across folds)
- Error bars can denote
  - Standard deviation
  - Standard error ($\sqrt{\sigma^2/N}$)

[Figure from Käser et al., LAK 2017]

Today

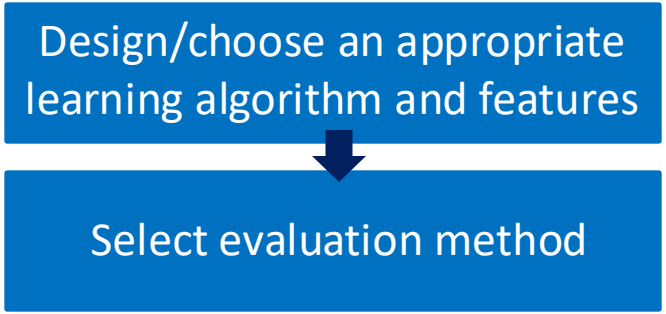
- Model Assessment and Evaluation
- Reporting of Results

Summary: Pipeline

Design/choose an appropriate learning algorithm and features

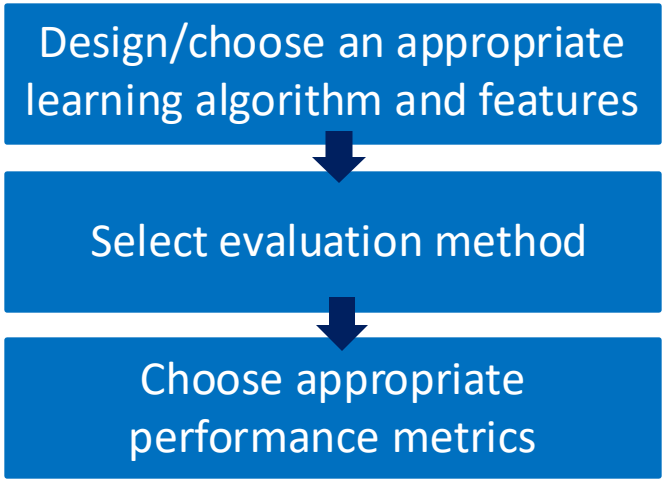
- Problem (e.g., classification or regression)
- Data set (e.g., size, type)
- Task (e.g., dropout prediction)

Summary: Pipeline



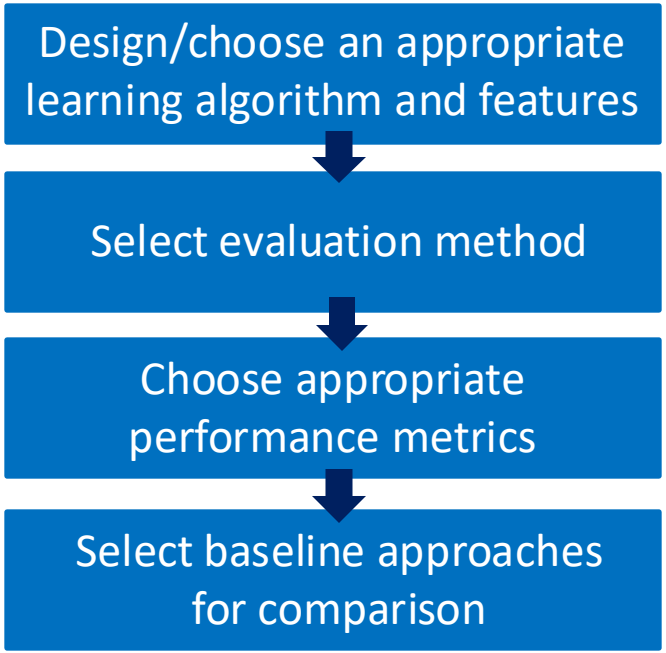
- Data set (e.g., size and type)
- Learning algorithm (e.g., model complexity)
- Goal (e.g., interpretation or prediction)
- Potential application (e.g., generalization to new users -> stratification)

Summary: Pipeline



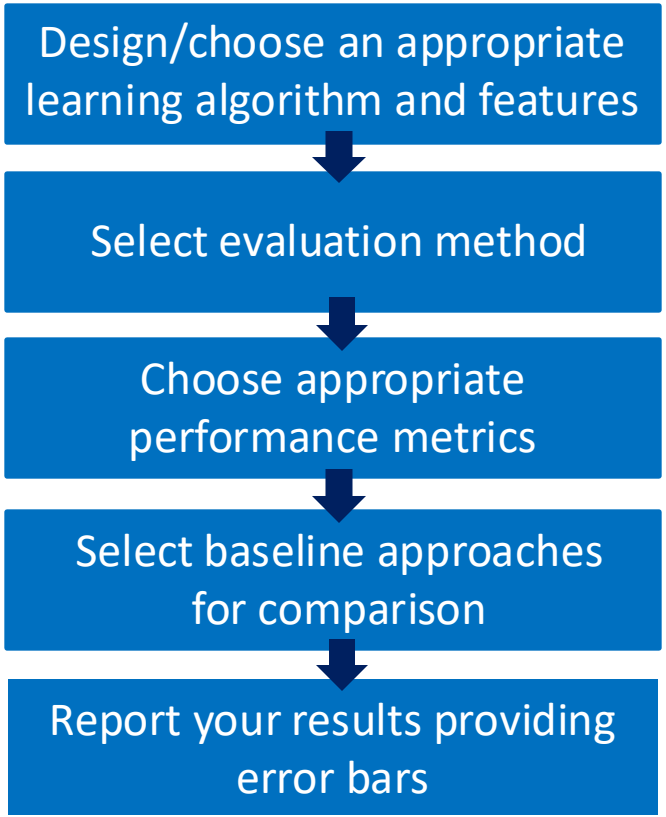
- Problem (e.g., classification or regression)
- Learning algorithm (e.g., model type)
- Potential application (i.e., what is important?)

Summary: Pipeline

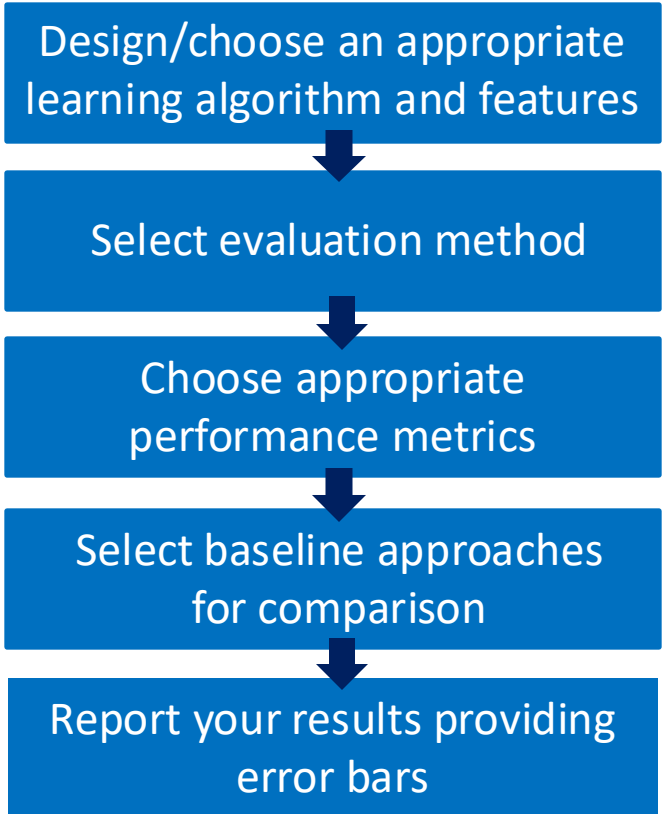


What do I want to show/prove, i.e. what is my claim (e.g., suggestion of new neural network architecture, suggestion of new features for a specific problem)?

Summary: Pipeline



Summary: Pipeline



There are many ways to solve a given task (e.g., predicting student performance). It is important that:

- You provide a clean and complete evaluation of your solution
- You are able to justify your decisions for each step