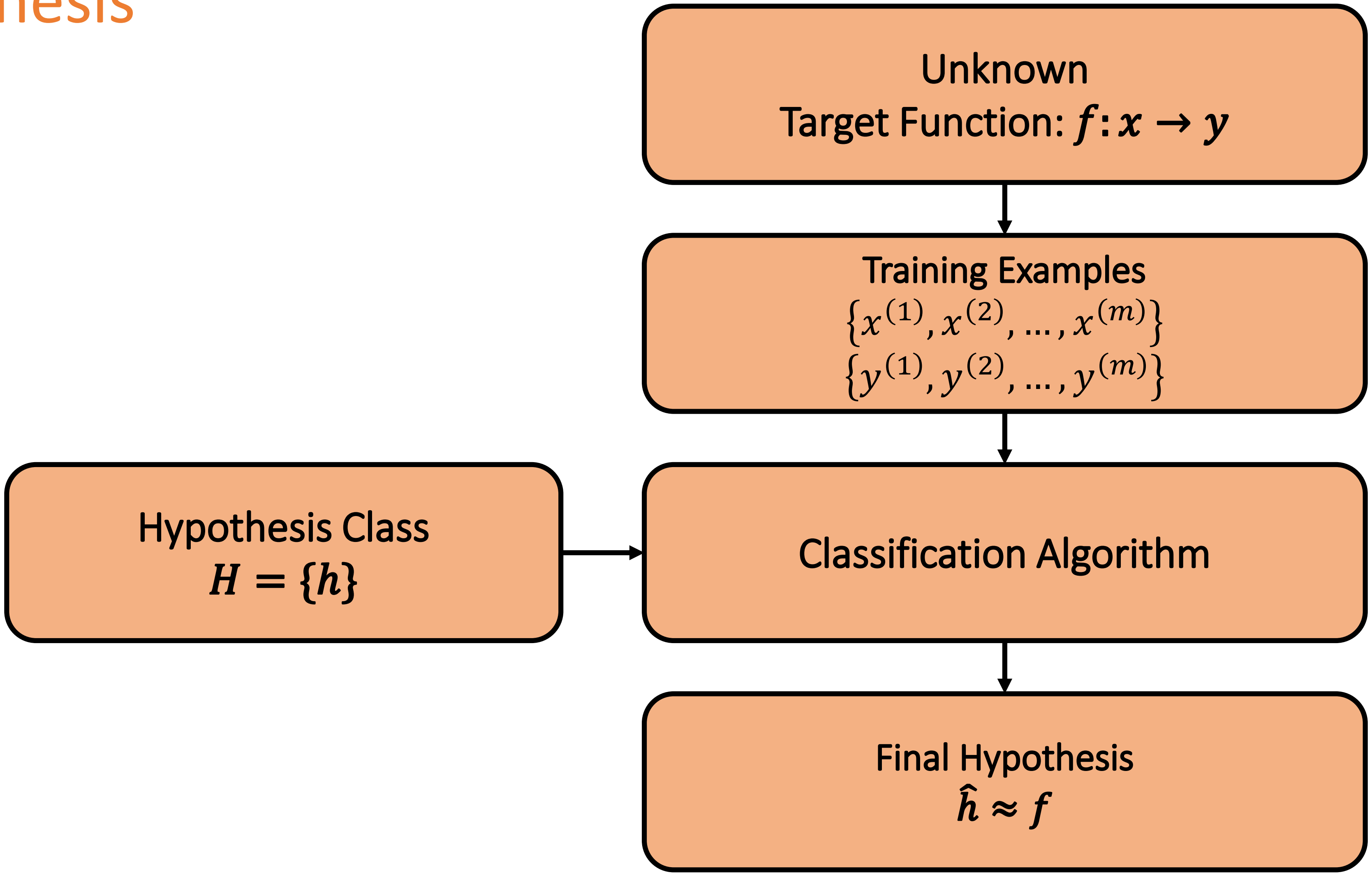
Machine Learning in Healthcare

#L07-Practical consideration on training a model

Technion-IIT, Haifa, Israel

Asst. Prof. Joachim Behar
Biomedical Engineering Faculty, Technion-IIT
Artificial intelligence in medicine laboratory (AIMLab.)
https://aim-lab.github.io/
Twitter: @lab_aim

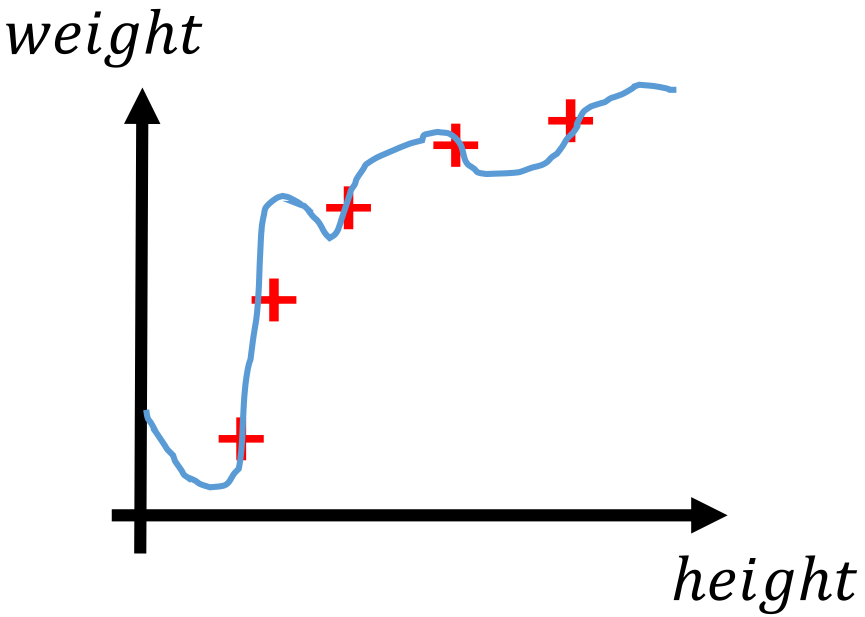
Hypothesis


Unknown
Target Function: $f: x \rightarrow y$
Training Examples
$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
$\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$
Hypothesis Class
$H = \{h\}$
Classification Algorithm
Final Hypothesis
$\hat{h} \approx f$

Evaluating a model

Overfitting

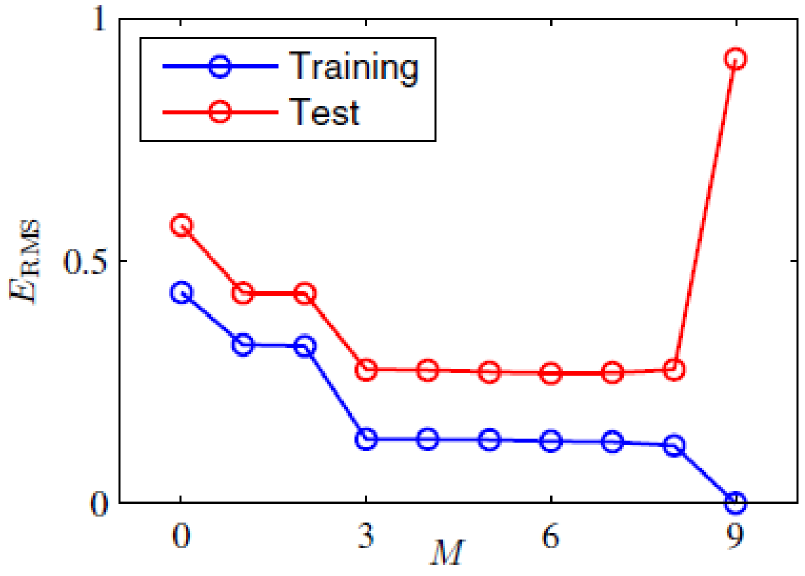
- You trained a model with its $J \to 0$. You feel very proud!
- Then you go out in the real world and start making predictions. Surprise, results are not as expected! What happened?
- Very likely your model is overfitting the training examples leading to bad generalization.

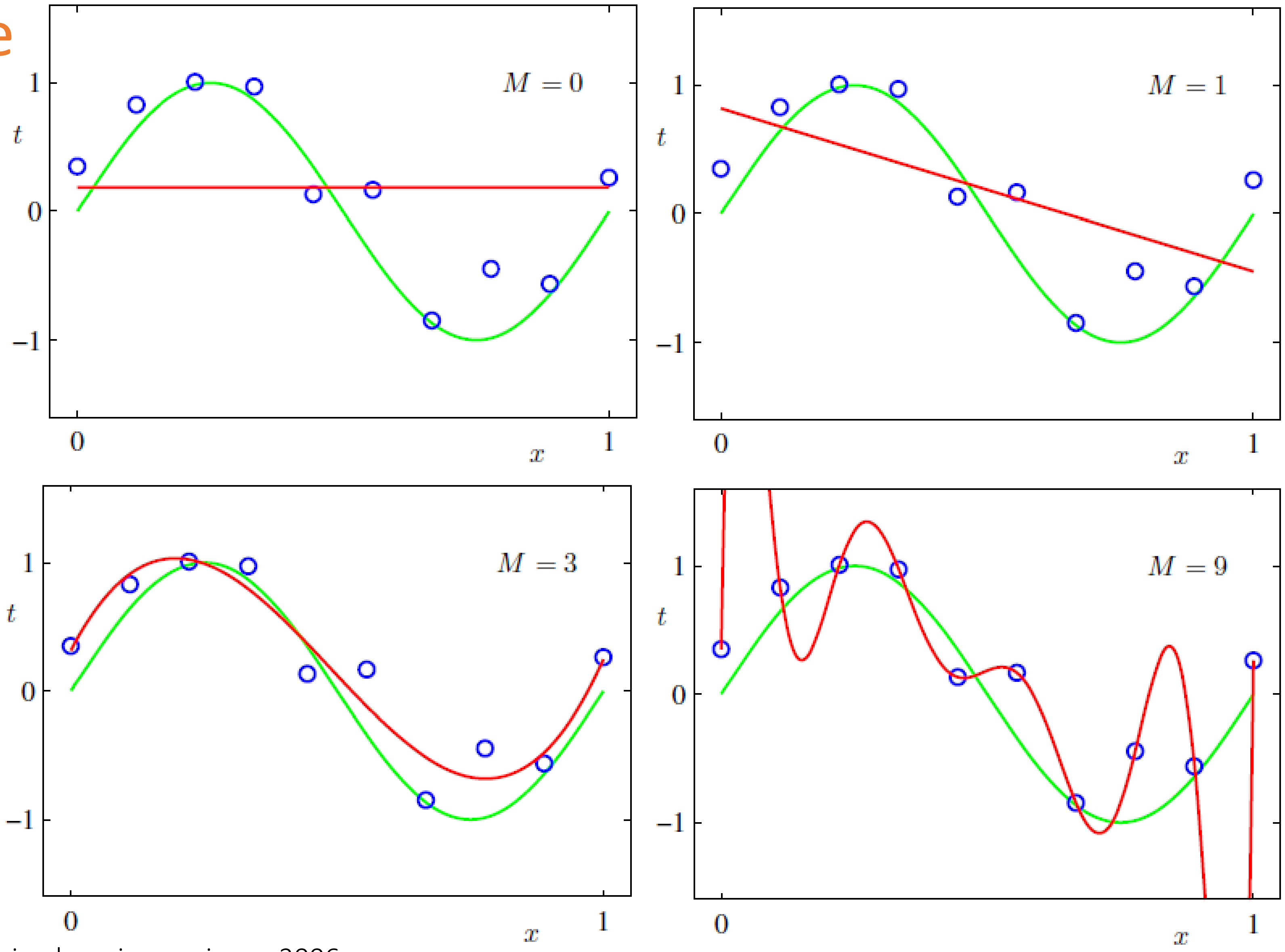


$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Reminder: bias-variance

- Under fitting (high bias),
- Overfitting (high variance).





Images: Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

AIMLab.

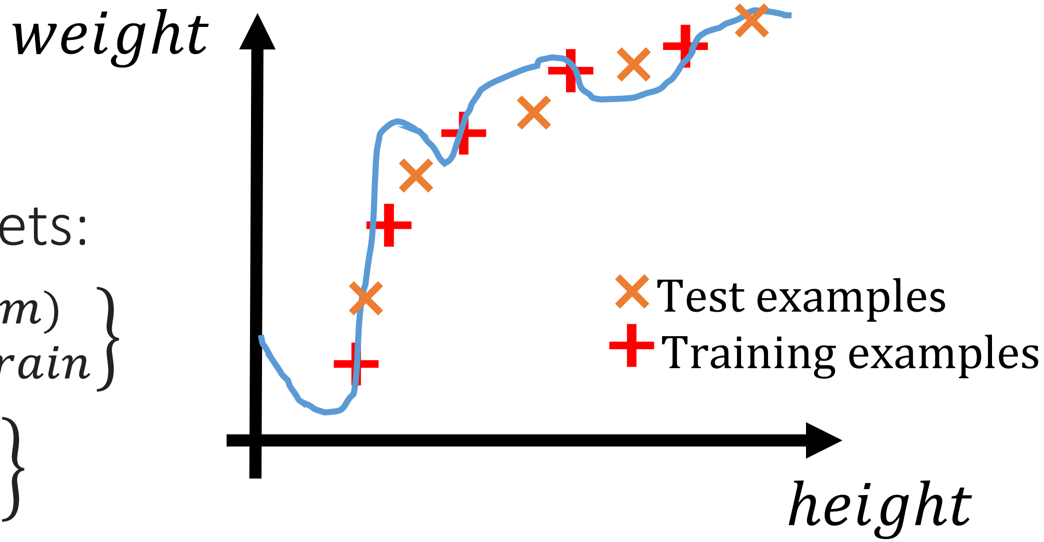
Test set

- How can we evaluate our model?
- **Training set** and **Test set**.
- We divide our dataset into training and test sets:

$$\left\{x_{train}^{(1)}, x_{train}^{(2)}, \dots, x_{train}^{(m)}\right\}, \left\{y_{train}^{(1)}, y_{train}^{(2)}, \dots, y_{train}^{(m)}\right\}$$

$$\left\{x_{test}^{(1)}, x_{test}^{(2)}, \dots, x_{test}^{(p)}\right\}, \left\{y_{test}^{(1)}, y_{test}^{(2)}, \dots, y_{test}^{(p)}\right\}$$

- Typically ~70% training and ~30% test.



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Going back to our history from slide 1. What would have happened if we had used a training and test set?
  - E.g. $J_{train} \to 0$ and $J_{test} \to 2.3$
  - We would have observed a gap between the training and test errors!

Diagnosing a model

- Say your model does not get the performance that you were expecting.
  - Do you need more data?
  - Is your model too simple/complex?
  - Do you need to regularize it?
  - Do you need to design new features?
- How do you figure out where to invest your time?

Model selection

Diagnosing a Model

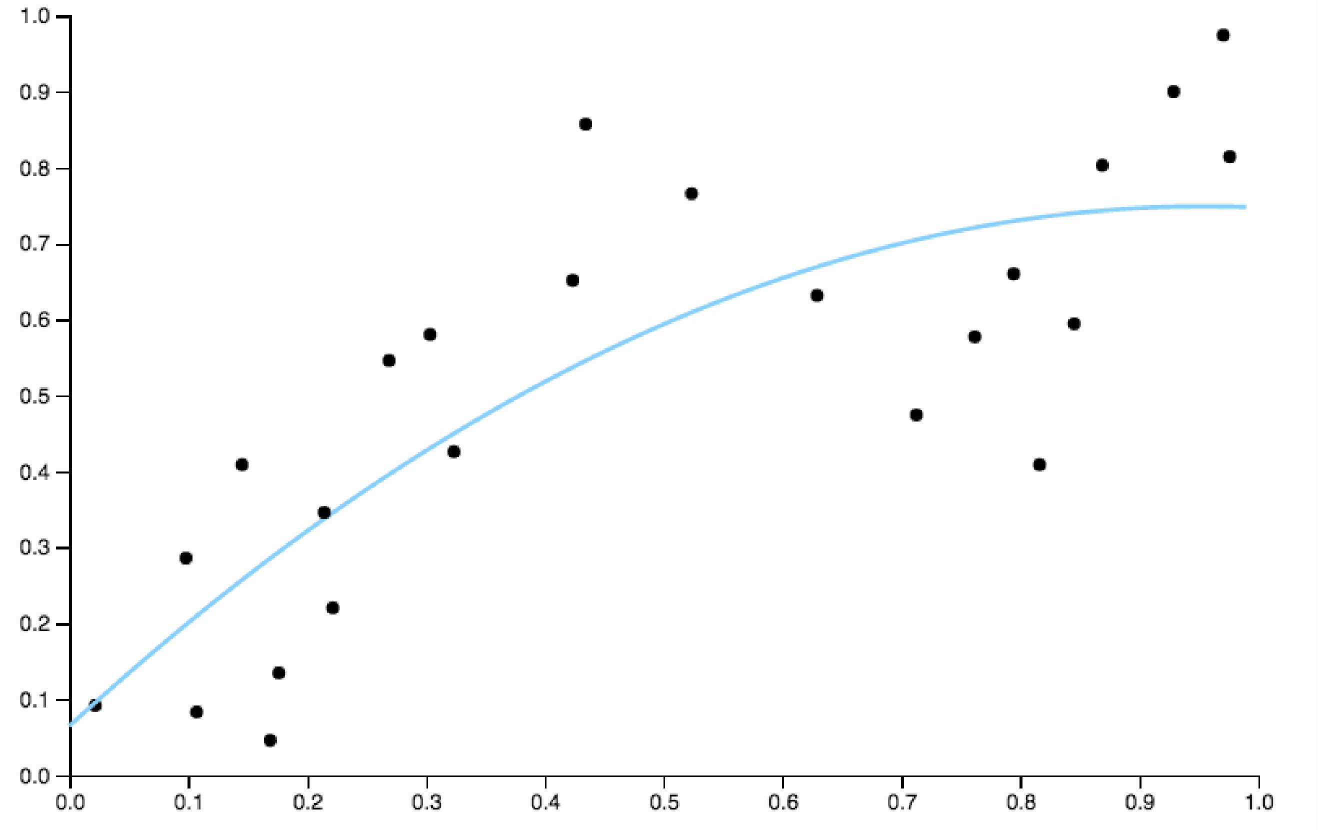
- Consider we evaluated our model and it does not generalize well i.e. $J_{Test} \gg J_{Train}$.
- How can we improve?
  1. Degree of the polynomial d? ➔ complexity of the hypothesis class.
  2. Value of the regularization parameter λ? ➔ dealing with overfitting.
  3. Do we need to increase the amount of data m? ➔ learning curve.

Tradeoff bias-variance

- We will see how to explore which avenue is the most promising. But first let's clarify again the difference between the notion of **model complexity** and **regularization**.

Tradeoff bias-variance

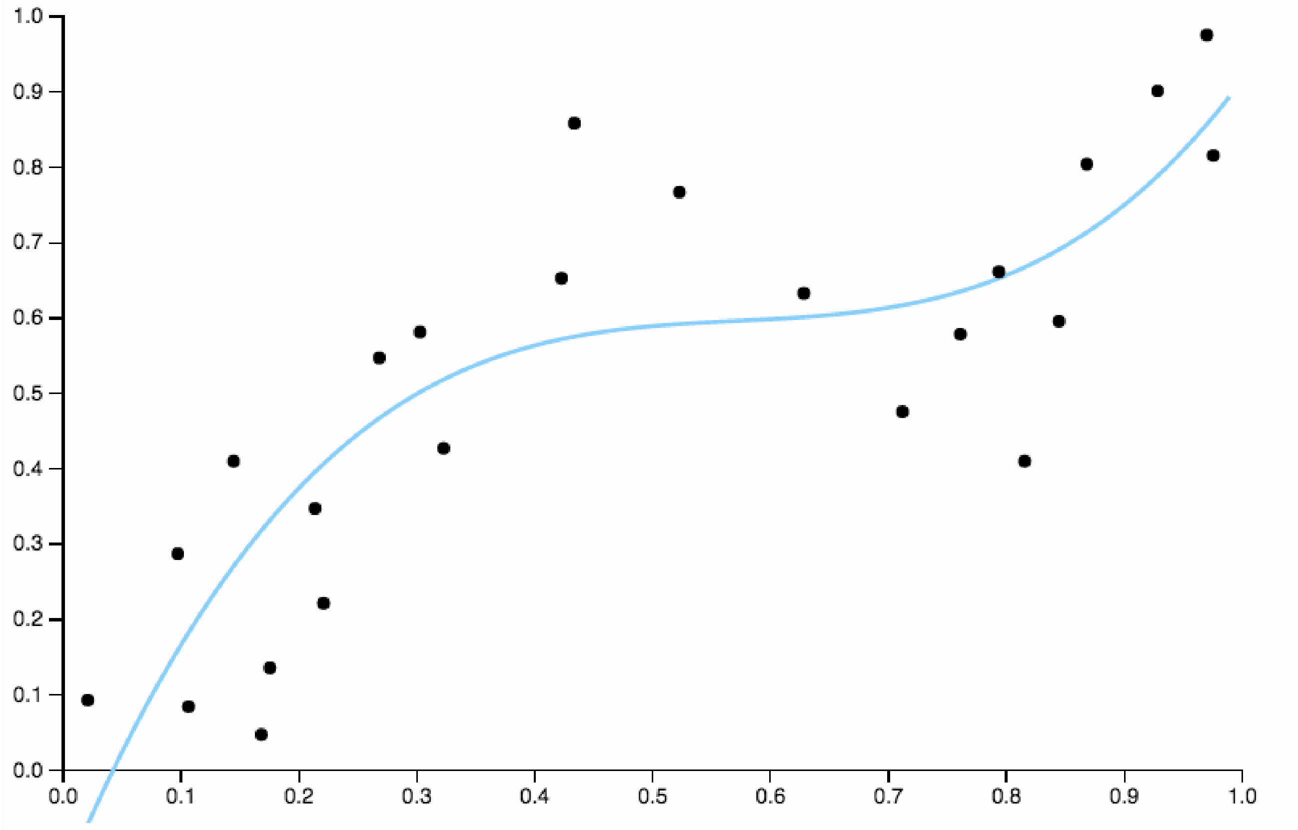
- Low d, model is too simple.



http://madrury.github.io/smoothers/

Tradeoff bias-variance

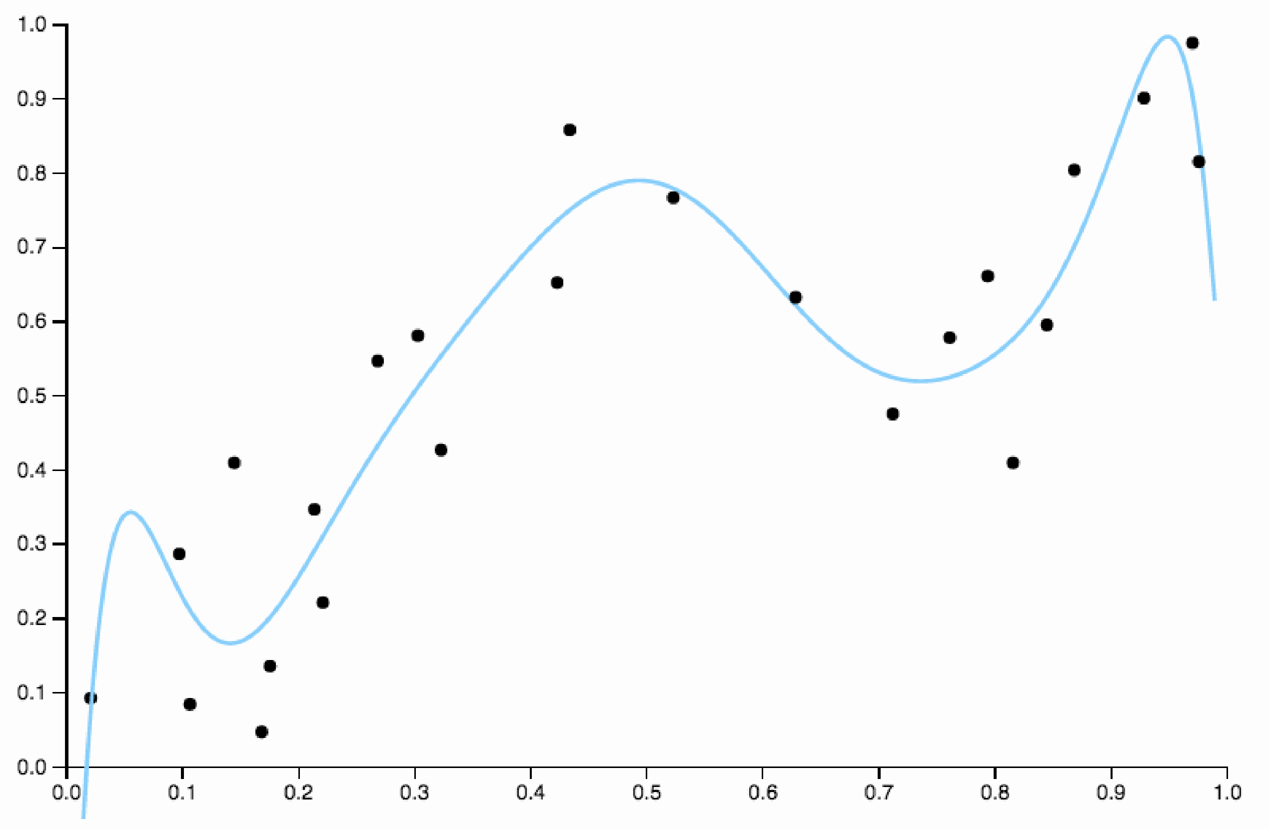
- Increase d. Better, still not enough!



http://madrury.github.io/smoothers/

Tradeoff bias-variance

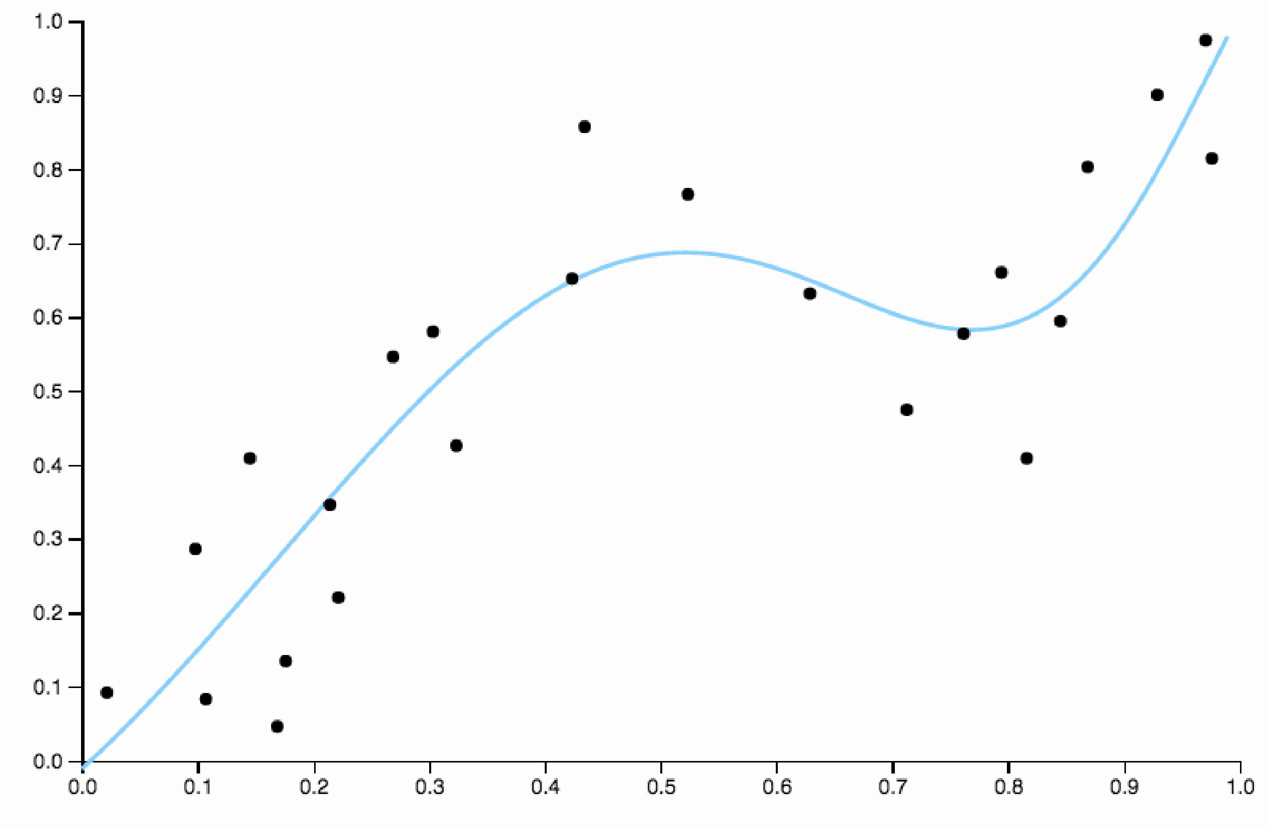
- We increase more d. But we now get overfitting!



http://madrury.github.io/smoothers/

Tradeoff bias-variance

- We keep the same d but now use some regularization. Sweet spot!



http://madrury.github.io/smoothers/

Tradeoff bias-variance

- Choose d so that the hypothesis class is about right.
- Then use regularization to use the power of a more complex model while not using it to its “full extend”.
- This will provide you with a **tradeoff between bias and variance**.

Diagnosing a model

- Consider we evaluated our model and it does not generalize well i.e. $J_{Test} \gg J_{Train}$.
- How can we improve?

1. Degree of the polynomial d? ➔ complexity of the hypothesis class.
2. Value of the regularization parameter λ? ➔ dealing with overfitting.
3. Do we need to increase the amount of data m? ➔ learning curve.

Tradeoff bias-variance

Validation set

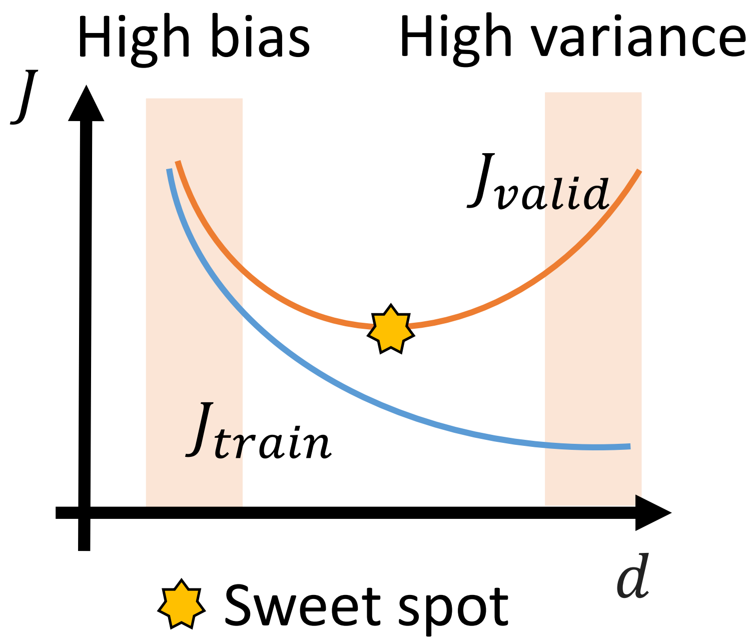
- Degree of the polynomial $\mathbf{d}$? ➔ complexity of the hypothesis class.
  - $h_w(x)$ assumed to be a polynomial of degree d.
  - Compute $J_{train}(w)$ for different values of d.
  - Select d based on best performance on the test set $J_{test}(w)$.
  - However, how do we ensure good generalization? Indeed, we just used the test set to tune our model.
    - We call this problem <u>information leakage</u>.

Validation set

- Degree of the polynomial $\mathbf{d}$? → complexity of the hypothesis class.
  - We need to divide our model in three parts (instead of two like before):
    - **Training, validation and test sets**.
    - A typical split is 60%-20%-20%.
  - We use the validation set to tune the d parameter (by minimizing $J_{valid}(w)$) and reporting the final error of the model on the test set $J_{test}(w)$.
    - This is <u>the</u> good practice!

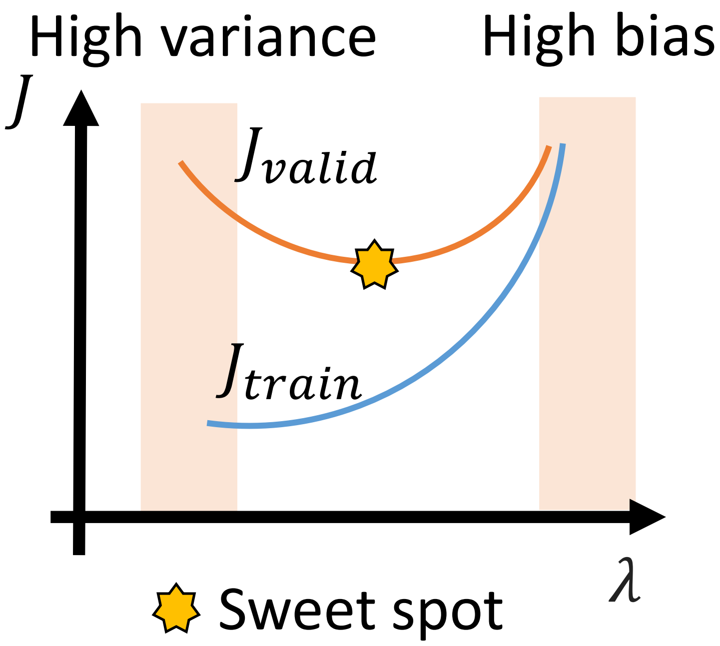
Validation set

- Degree of the polynomial d? ➔ complexity of the hypothesis class.



Validation set

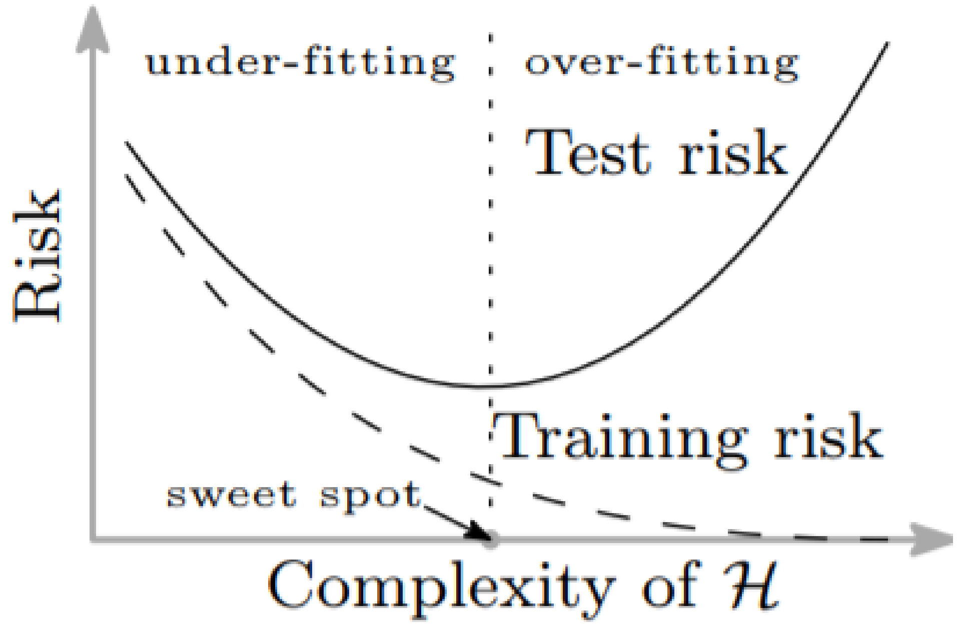
- Value of the regularization parameter λ? ➔ dealing with overfitting.



- $J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} w_j^2\right]$
  - Blue: the regularization term.
  - λ: Regularization parameter. It controls the tradeoff bias-variance.
  - $\lambda \rightarrow 0$: no regularization.
  - $\lambda \rightarrow \infty$: underfitting ($h_w(x) = w_0$).

Bias-variance tradeoff

- The control of the **function class complexity** is used to balance between bias and variance and find the **sweet spot that is a good tradeoff**.
- The control of the function class complexity may be **explicit via the choice of the hypothesis class or implicitly through regularization**.



(a) U-shaped "bias-variance" risk curve

Belkin, Mikhail, et al. "Reconciling modern machine learning and the bias-variance trade-off." arXiv preprint arXiv:1812.11118 (2018).

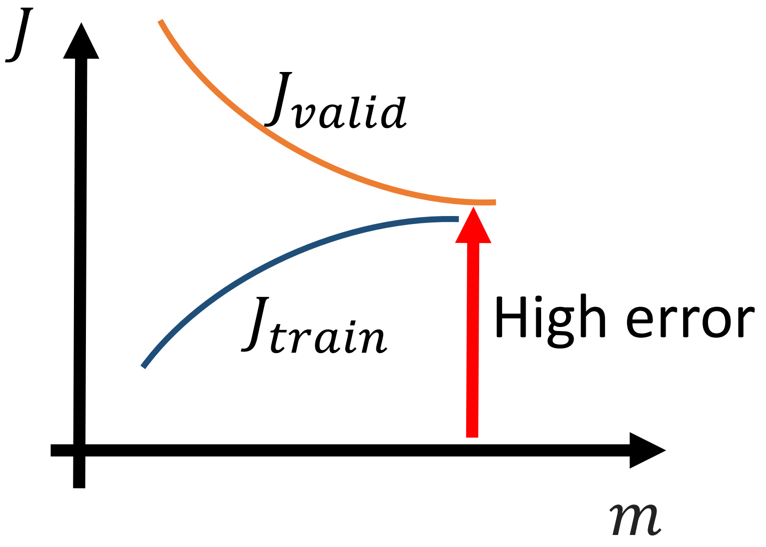
Learning curves

Learning curves

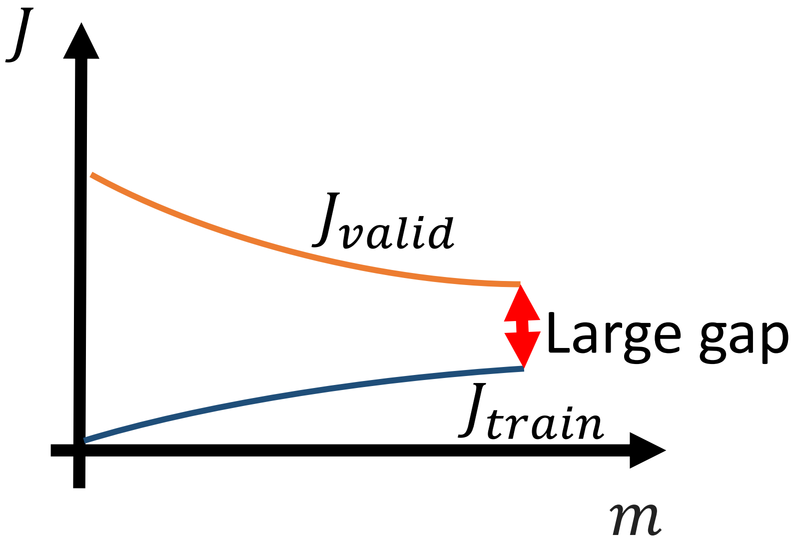
- Do we need to increase the amount of data m? ➔ learning curve
  - Do we have enough examples for building our model?
  - **Learning curve**: graph that is used to compare the performance of a model on training and validation datasets over a varying number of training examples.
  - Performance will generally improve as the number of training examples increases.
  - We want to graph $J_{train}(w)$ and $J_{valid}(w)$ as a function of m.

Learning curves

- **Do we need to increase the amount of data m? ➔ learning curve**

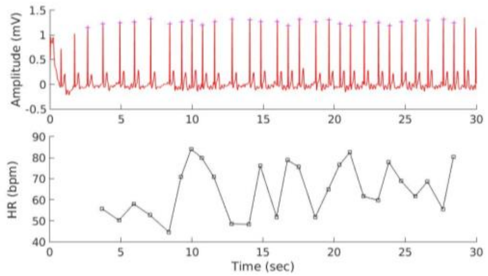


Diagnosis: High bias
➔ Getting more data will not help.



Diagnosis: High variance
➔ Getting data might improve.

Learning curves: example

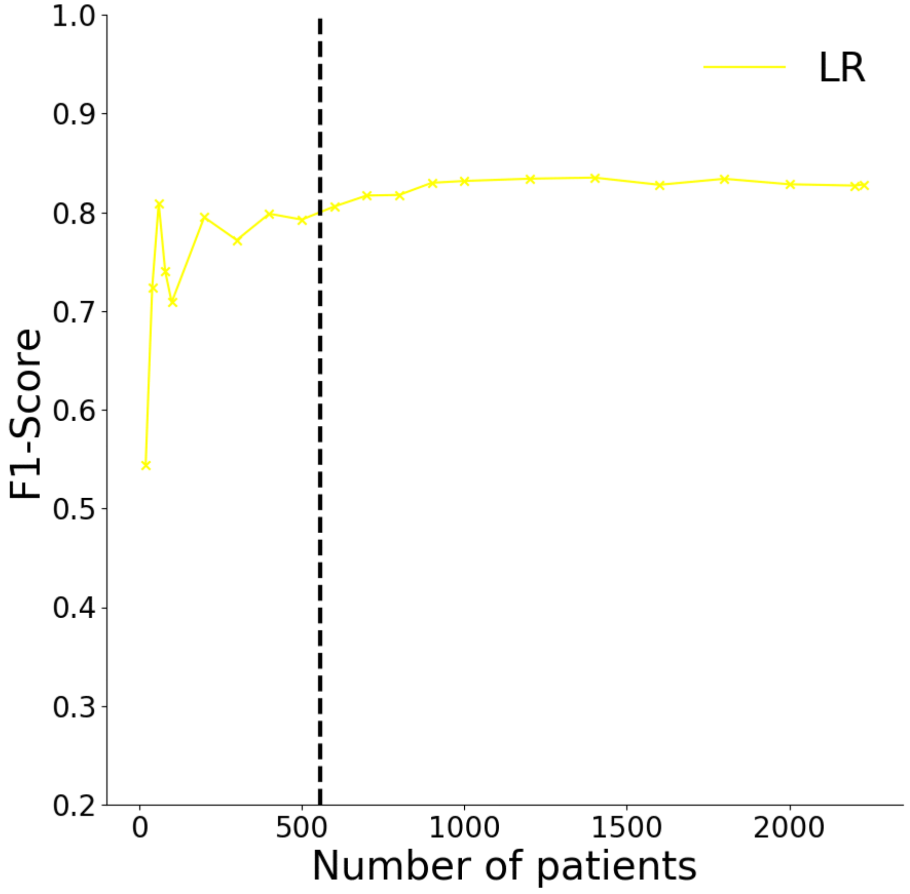


- Binary classification: AF, non-AF.
- A total of n=2,891 unique patients recording.
  - 2,612 non-AF and 279 AF.
- Each recording is 24h long, totaling 68,800h.
- Model elaboration:
  - 80-20% train-test set split.
  - 5-fold cross validation on the training set.

Chocron, A., Oster, J., Biton, S., Franck, M., Elbaz, M., Zeevi, Y. Y., & Behar, J. (2020). Remote atrial fibrillation burden estimation using deep recurrent neural network. *arXiv preprint arXiv:2008.02228*.

Learning curves: example

- We create the learning curve while adding patients one by one.
- When adding patients we alternate with AF and non-AF so that we have 50-50%.
- We do this until we exhaust the AF examples - dotted black vertical line.
- Will we gain by increasing the number of patients?



Chocron, A., Oster, J., Biton, S., Franck, M., Elbaz, M., Zeevi, Y. Y., & Behar, J. (2020). Remote atrial fibrillation burden estimation using deep recurrent neural network. *arXiv preprint arXiv:2008.02228*.

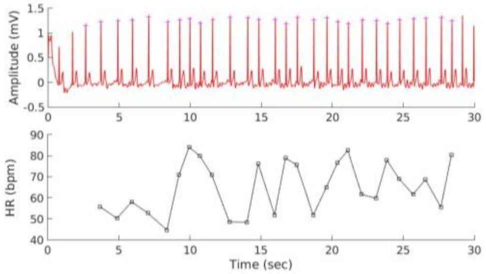
Error analysis

Error analysis

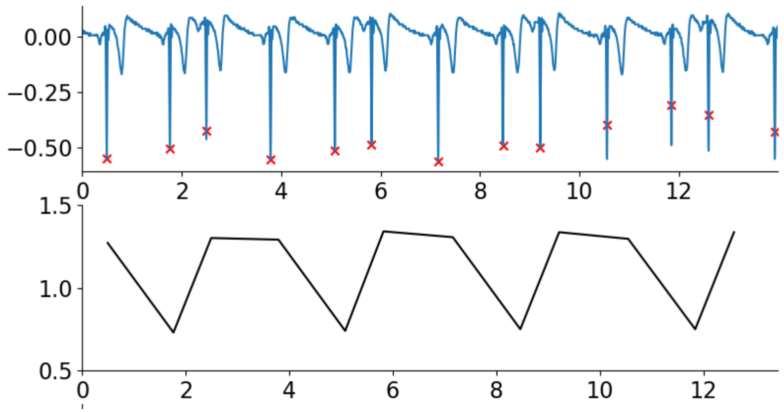
- What if the problem actually comes from the fact that the features are not descriptive enough/missing some characteristic of the data?
- Manually examine the examples in the validation set that your algorithm classify incorrectly. This is called “**error analysis**”:
  - Look for systematic trends the algorithm is making errors on.
  - Try to categorize.

AIMLab.

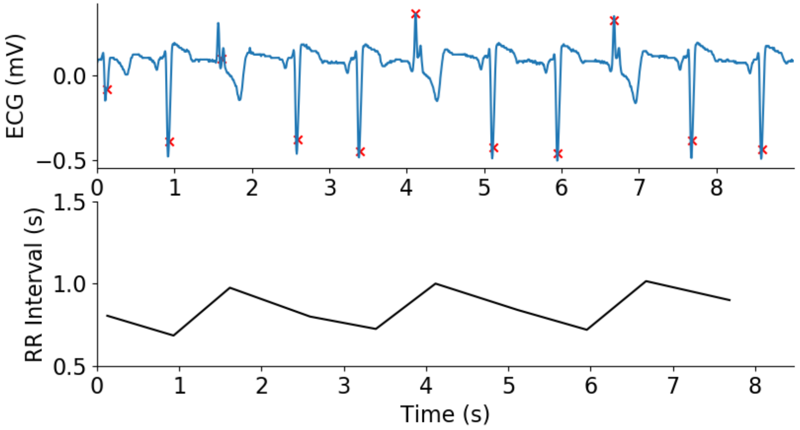
Error analysis



- E.g. Atrial fibrillation:
  - Is the misclassification due to the presence of noise?
  - Some specific non-AF rhythms that we do not learn well?
- We can start by looking at single examples:



Atrial premature contractions

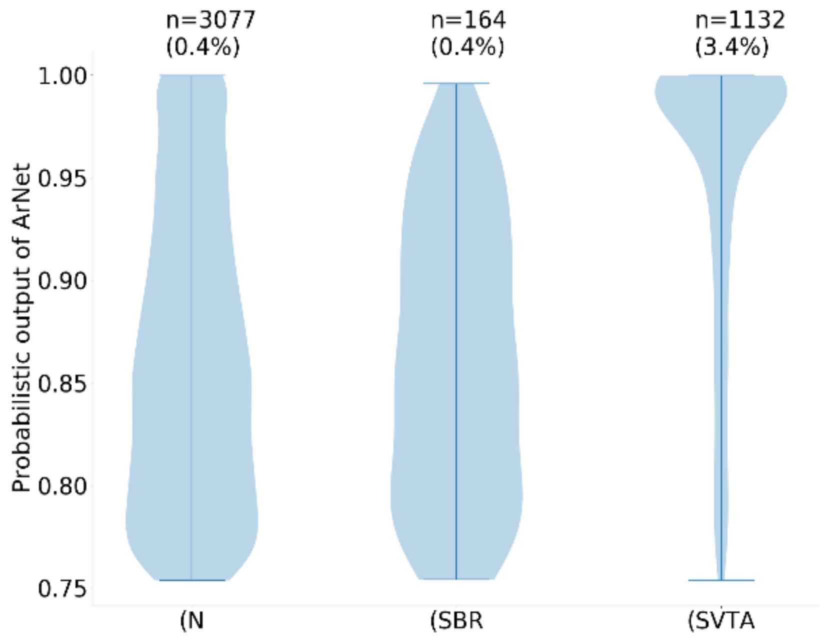


Trigeminy

Chocron, A., Oster, J., Biton, S., Franck, M., Elbaz, M., Zeevi, Y. Y., & Behar, J. (2020). Remote atrial fibrillation burden estimation using deep recurrent neural network. *arXiv preprint arXiv:2008.02228*.

Error analysis: examples

- Aggregation of examples:



- If you spot that some specific type of recurrent error then you can look at:
  - Adding features that may discriminate them from the other examples.
  - Augmenting their representation in the dataset if they are not well represented.

Chocron, A., Oster, J., Biton, S., Franck, M., Elbaz, M., Zeevi, Y. Y., & Behar, J. (2020). Remote atrial fibrillation burden estimation using deep recurrent neural network. *arXiv preprint arXiv:2008.02228*.

Generalization performance

Generalization performance

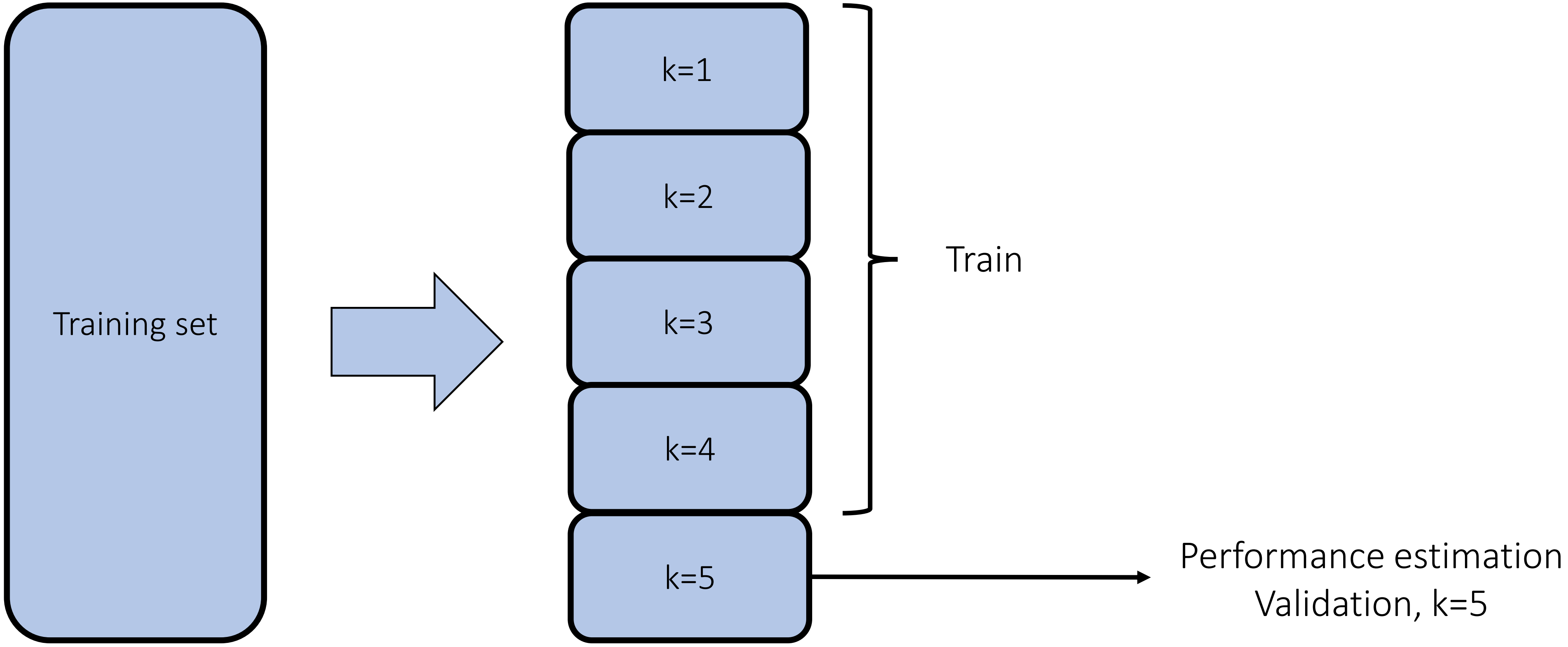
- **Generalization performance** relates to how accurately the algorithm is able to predict outcome values for previously unseen data.
- In order to set our hyperparameters we created a validation set to assess how the model will generalize to an independent dataset. This is called **cross-validation**.
- However, in practice **we want to use multiple rounds of cross-validation** to reduce variability. It will provide a more accurate estimate of model prediction performance.
- How do we do that?

Generalization Performance

- Recall: for training our model we:
  - Use a **training set** from which the model learns.
  - Use a **validation set** that enables us to perform hyperparameters selection.
  - Use a **test set** which we use to assess the final model performance. This set is not used to train the model or tune its hyperparameters.
- How can we assess **generalization performance** to best decide on our hyperparameters?

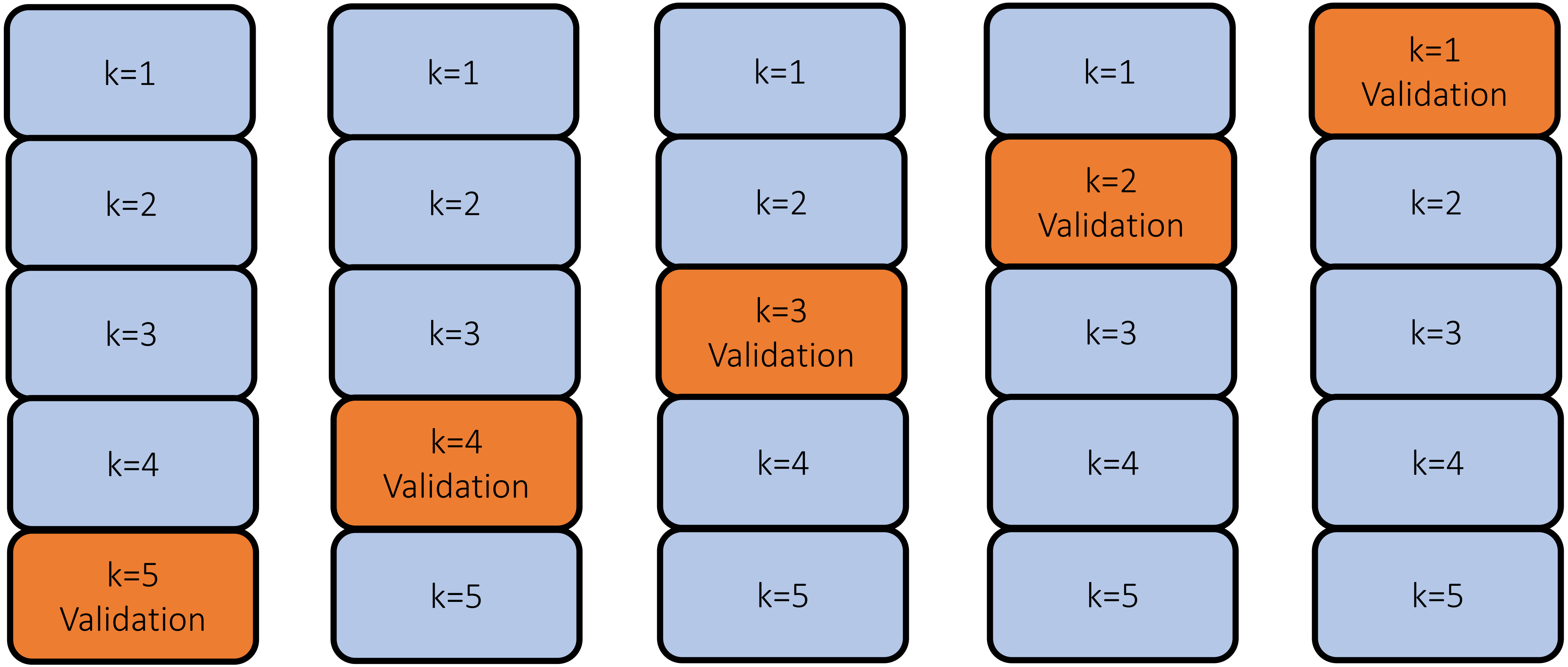
K-fold cross validation

- Divide the training set into k folds. Say here $k = 5$



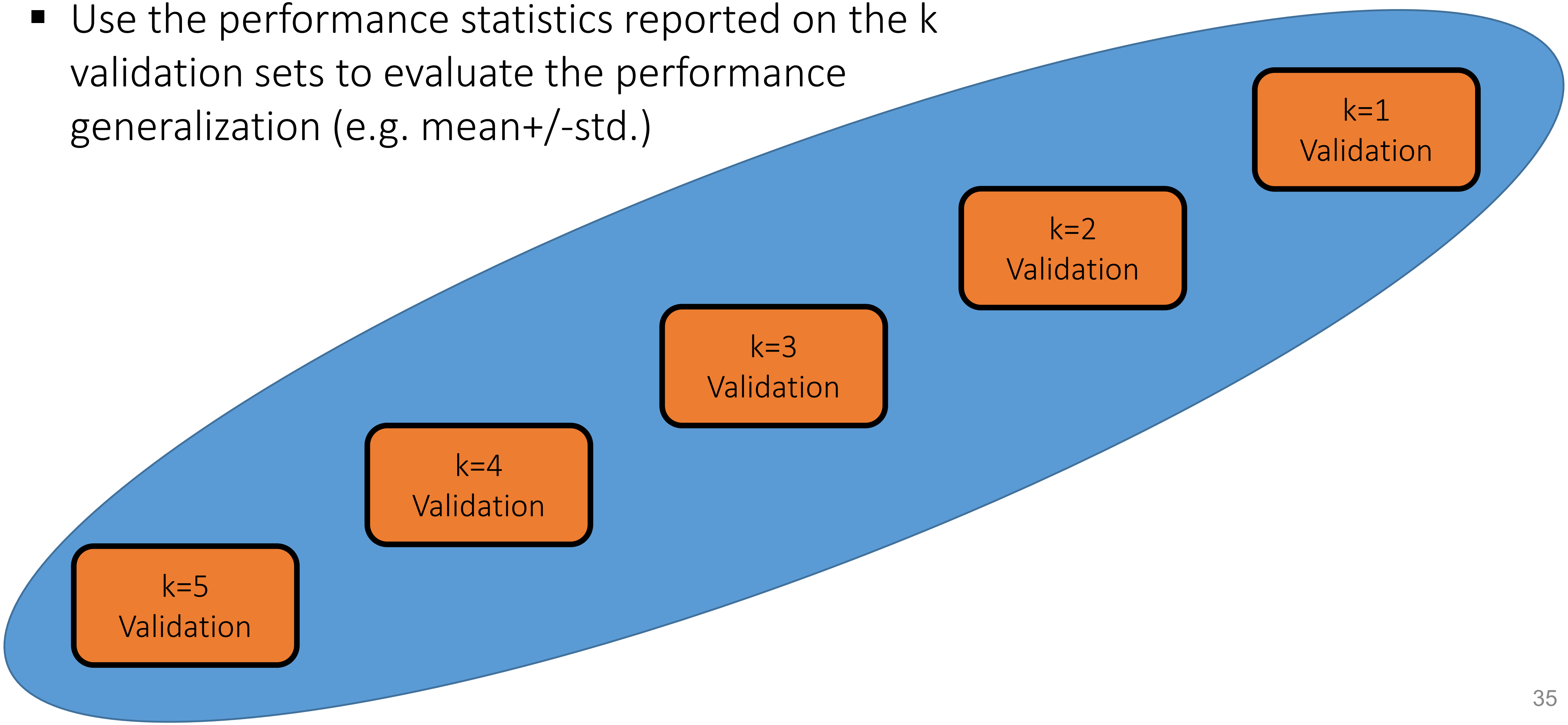
K-fold cross validation

- Repeat on each of the k folds. Each time consider 4 as training and 1 as validation:



K-fold cross Validation

- Use the performance statistics reported on the k validation sets to evaluate the performance generalization (e.g. mean+/-std.)



Cross validation

- Cross validation has different flavors.
  - K-fold cross validation.
    - What we just saw.
  - Leave-one-out cross-validation.
    - Can be computationally expensive.
  - Repeated random sub-sampling validation.
    - May not cover all the training examples if not enough iterations.
- **Important note:** to be able to compare performance statistics accross multiple experiments, the split train-validation-test must be the same – write a script to be able to reproduce this exact split.

Table 1
Study database. The diagnosis is based on the ICSD-3 and AASM 2017 guidelines and using the recommended rule for hypopnea.

Diagnosis	Number	Percentage (%)
Non-OSA	503	56.7
Mild OSA	206	23.2
Moderate OSA	103	11.6
Severe OSA	75	8.5
Total	887	

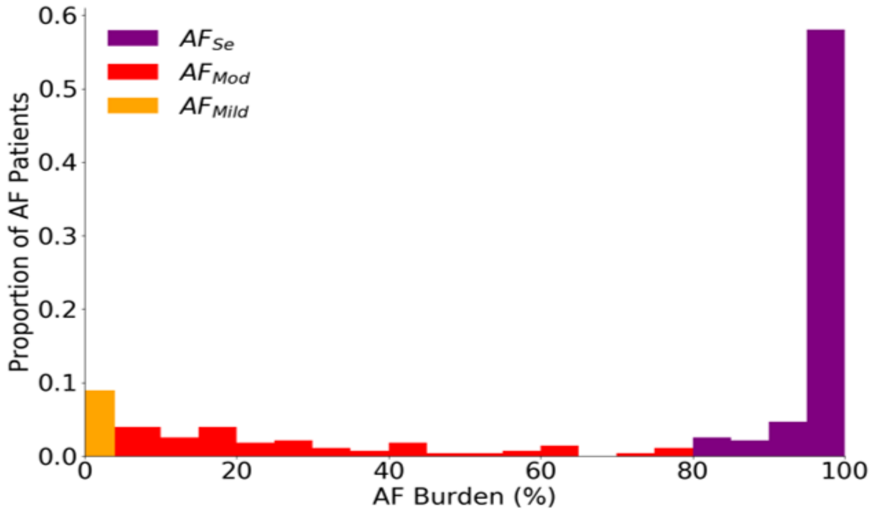
Behar, Joachim A., et al. "Feasibility of Single Channel Oximetry for Mass Screening of Obstructive Sleep Apnea." EClinicalMedicine (2019).

Stratification

- When doing cross validation consider performing **stratification** to ensure the different folds have the same proportion of each classes.
- Examples:

Table 1
Study database. The diagnosis is based on the ICSD-3 and AASM 2017 guidelines and using the recommended rule for hypopnea.

Diagnosis	Number	Percentage (%)
Non-OSA	503	56.7
Mild OSA	206	23.2
Moderate OSA	103	11.6
Severe OSA	75	8.5
Total	887	



Behar et al. "Feasibility of Single Channel Oximetry for Mass Screening of Obstructive Sleep Apnea." EClinicalMedicine (2019).
Chocron et al. Remote atrial fibrillation burden estimation using deep recurrent neural network. arXiv preprint arXiv:2008.02228.

Nested cross validation

- What if we want to estimate the **generalization error** of our model?
- **Nested cross validation** consists of:
    - An inner loop: fit a model to each training set, select hyperparameters by maximizing the performance over the validation set.
    - The outer loop where you estimate the generalization error by averaging test set scores over several dataset splits.

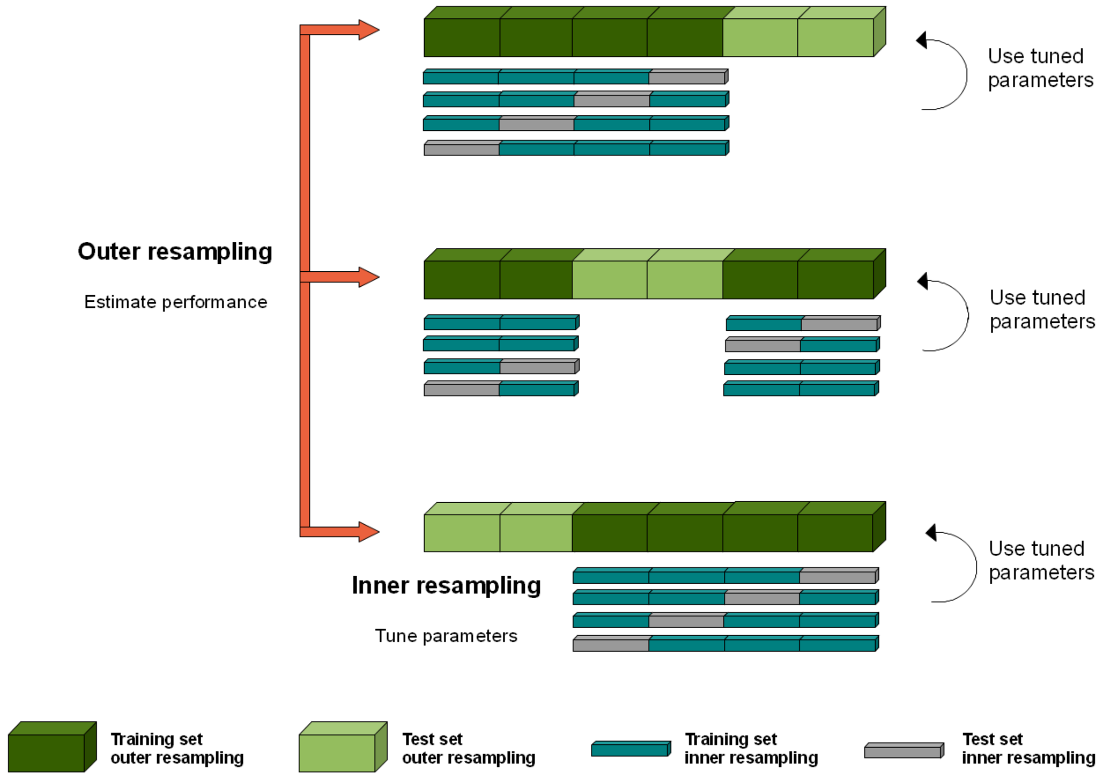


Image: https://weina.me/nested-cross-validation/

Nested cross validation

- Nested Cross Validation will provide you with a good idea of the generalization capacity of your model by:
  - Analyzing the generalization error: e.g. is the standard deviation of the error very high?
  - Analyzing model settings consistency (e.g. features selected, value of the weights in LR, hyperparameters values).

Nested cross validation

Feature extraction

4 SpO_2 features | 8 demographic features

Machine Learning Optimization → Gradient Descent

Outer loop : 5 folds → use tuned hyperparameters in each fold and optimize parameters across the 5 folds.

Inner loop : 100 sub-folds → select hyperparameters so that statistics F_1 is maximized across the 100 sub-folds.

Table 5
Performance of the models (average and standard deviation for the test sets) evaluated against the AHI R2017.

Statistics/model	AUROC	Ac	F_1	NPV	PPV	Se	Se-mild	Se-moderate	Se-severe	Sp
NoSAS	0.83 ± 0.03	0.72 ± 0.03	0.58 ± 0.07	0.69 ± 0.03	0.81 ± 0.04	0.46 ± 0.08 (176/384)	0.33 ± 0.04 (67/206)	0.54 ± 0.14 (57/103)	0.69 ± 0.14 (52/75)	0.92 ± 0.02 (463/503)
STOP-BANG	0.77 ± 0.04	0.72 ± 0.02	0.65 ± 0.04	0.73 ± 0.03	0.70 ± 0.03	0.61 ± 0.05 (233/384)	0.47 ± 0.06 (97/206)	0.71 ± 0.13 (75/103)	0.81 ± 0.11 (61/75)	0.81 ± 0.02 (405/503)
LR-SB	0.87 ± 0.04	0.75 ± 0.04	0.76 ± 0.04	0.89 ± 0.04	0.66 ± 0.04	0.90 ± 0.04 (345/384)	0.84 ± 0.04 (172/206)	0.97 ± 0.03 (100/103)	0.97 ± 0.06 (73/75)	0.64 ± 0.05 (324/503)
LR-ODI	0.92 ± 0.01	0.85 ± 0.03	0.83 ± 0.03	0.87 ± 0.02	0.84 ± 0.04	0.82 ± 0.03 (315/384)	0.70 ± 0.04 (143/206)	0.94 ± 0.04 (97/103)	1.00 ± 0.00 (75/75)	0.88 ± 0.03 (443/503)
LR-SpO_2	0.92 ± 0.02	0.85 ± 0.02	0.82 ± 0.03	0.87 ± 0.03	0.82 ± 0.01	0.83 ± 0.05 (317/384)	0.70 ± 0.07 (143/206)	0.96 ± 0.04 (99/103)	1.00 ± 0.00 (75/75)	0.86 ± 0.01 (435/503)
OxyDOSA	0.94 ± 0.02	0.86 ± 0.03	0.84 ± 0.04	0.90 ± 0.03	0.82 ± 0.03	0.87 ± 0.04 (335/384)	0.77 ± 0.05 (158/206)	0.99 ± 0.02 (102/103)	1.00 ± 0.00 (75/75)	0.85 ± 0.03 (428/503)

Four sets of classifiers were evaluated for comparison. These are denoted: LR-SB for which classifiers were trained using all the demographic features used for the STOP-BANG questionnaire; LR-ODI for which classifiers were trained using the oxygen desaturation index as the sole feature; LR-SpO_2 for which classifiers were trained using all the oxygen saturation features; OxyDOSA for which classifiers were trained using features selected from all oxygen saturation and the demographic features available from the STOP-BANG questionnaire. Statistics are reported for the test sets.

Behar, Joachim A., et al. "Feasibility of Single Channel Oximetry for Mass Screening of Obstructive Sleep Apnea." EClinicalMedicine (2019).

Important notes on performance generalization

- Summary:
  - **Training set:** What you develop the model on.
  - **Validation set:** Data excluded from model development but used in model hyperparameters selection.
  - **Test set:** Data excluded from ALL development, and used once to evaluate the final model.
- Be VERY aware of **information leakage** i.e. accidental use of information from the test set. You want to avoid that!

Important notes on generalization

- Examples of information leakage pitfalls:
  - Normalizing using all the data, then splitting into train/test.
  - Using data from the same patient in train and test.
  - Training a model, evaluating on test, iterating.

Take Home

- Tools for **diagnosing a model**:
  - Complexity of the hypothesis class,
  - Dealing with overfitting through regularization,
  - Dataset-size and learning curve.
  - Error analysis.

Take home

- To search for adequate hyperparameters and evaluate your model **performance** and **generalization performance** divide your data into: **train, validation and test sets**:
  - Use a **training set** from which the model learns.
  - Use a **validation set** that enables to perform hyperparameters selection.
  - Use a **test set** which we use to assess the final model performance. This set is not used to train the model or tune its hyperparameters.
- Use a flavor of **cross-validation** when optimizing model hyperparameters.
  - K-fold cross validation.
  - Leave-one-out cross-validation.
  - Repeated random sub-sampling validation.
  - Nested Cross Validation.

Take Home

- Perform **stratification** to ensure the different folds have the same proportion of each classes
- A practical approach:
  - Start with a simple algorithm, obtain results on cross-validation. Then plot learning curves and get an idea of where you can improve.
  - Avoid “premature optimization” and let evidence guide your design.
- Do not use information from your test set, this is **information leakage**, that will lead to bad generalization of the model performance.

References

[1] Machine Learning Basic Concepts. CDT Lectures Notes 2013. Alistair Johnson.

[2] Coursera, Andrew Ng. Advice for applying machine learning.

[3] Belkin, Mikhail, et al. "Reconciling modern machine learning and the bias-variance trade-off." arXiv preprint arXiv:1812.11118 (2018).