METHODS 3: MULTILEVEL STATISTICAL MODELLING AND MACHINE LEARNING

COURSE OVERVIEW (SECOND HALF)

W6: Machine Learning Intro

Moving the goal away from explanations towards prediction and getting Python running

W7: Linear Regression Revisited (machine learning)

How to constrain our models to make them more predictive

W8: Logistic regression (machine learning)

Categorizing responses based on informed guesses

W9: Dimensionality reduction, Principled Component Analysis (PCA)

What to do with very rich data?

W10: Organizing and preprocessing messy data

How to clean up?

W11: Final evaluation and wrap-up of course

Ask anything

COURSE OVERVIEW (SECOND HALF)

W6: Machine Learning Intro

Moving the goal away from explanations towards prediction and getting Python running

W7: Linear Regression Revisited (machine learning)

How to constrain our models to make them more predictive

W8: Logistic regression (machine learning)

Categorizing responses based on informed guesses

W9: Dimensionality reduction, Principled Component Analysis (PCA)

What to do with very rich data?

W10: Organizing and preprocessing messy data

How to clean up?

W11: Final evaluation and wrap-up of course

Ask anything

TODAYS PLAN

Aim for today:

- Revisit ML concepts, in visual way (and more repetition than anticipated)
- Exercises on:
    - Bias/variance
    - CV
    - Regularization

TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

CATCH-UP + DISCLAIMER

CATCH-UP

- How are you holding up?

- Any comments on the course for Lau or me?

CATCH-UP

- Something you would like (re)visited for next class?
  - *Revisit concept of __*
  - *Python specifics*
    - *Class() objects .*
    - *List comprehensions*
    - *Dict, set, tuple, list*
  - *More time with assignment instead*
  - *... something else?*

CATCH-UP

- Take two minutes with your partner to discuss:

- Something you would like (re)visited for next class?
    - *Revisit concept of __*
    - *Python specifics*
        - *Class() objects*
        - *List comprehensions*
        - *Dict, set, tuple, list*
    - *More time with assignment instead*
    - *… something else?*

DISCLAIMER

DISCLAIMER

- Missing mathy, technical background that you guys have

DISCLAIMER

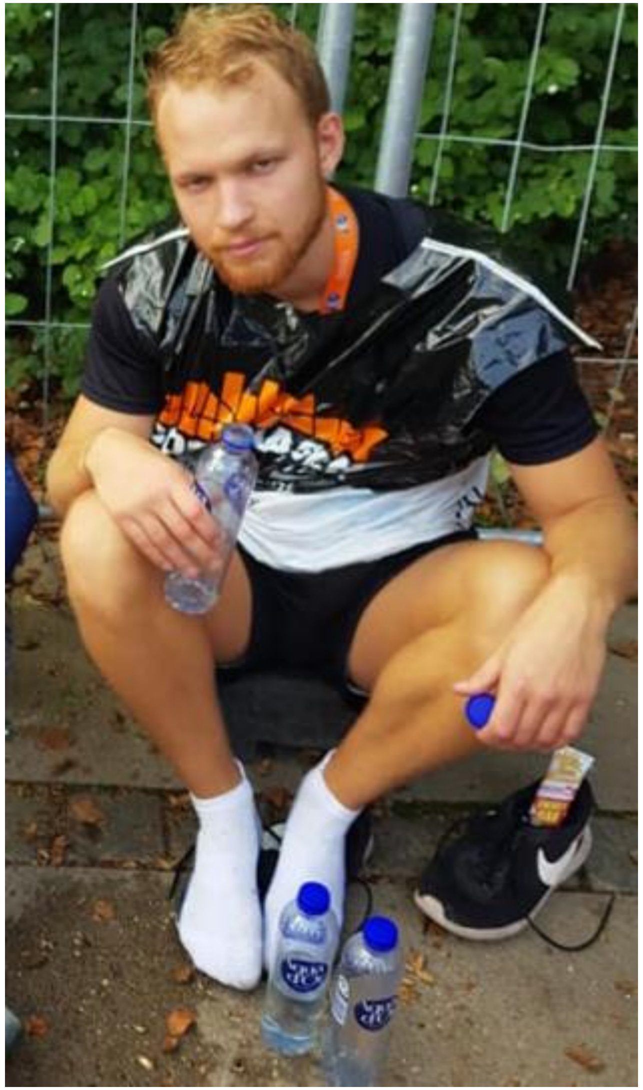
- Missing mathy, technical background that you guys have
- Everything from scratch (and week to week)

DISCLAIMER

- Missing mathy, technical background that you guys have
- Everything from scratch (and week to week)
- Feedback – both of us for classes?

Me, when learning entire curriculum had changed this summer

DISCLAIMER



- Missing mathy, technical background that you guys have
- Everything from scratch (and week to week)
- Feedback – both of us for classes?

DISCLAIMER

… however!

- Good conceptual understanding of ML

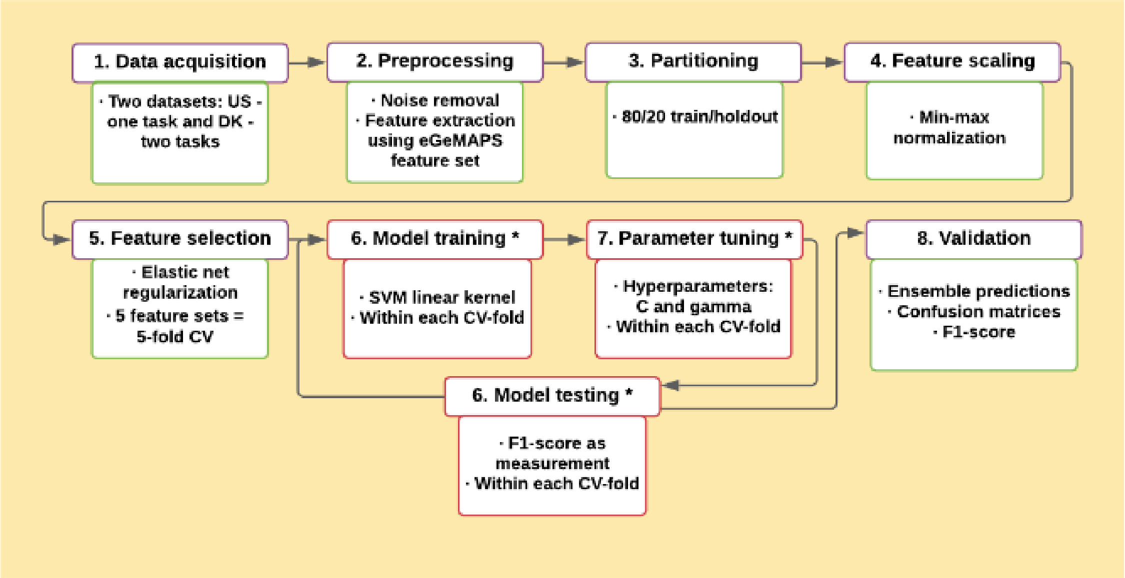


Figure 1. Overview of the machine learning pipeline. Purple refers to the general steps whereas green refers to specifics. "US" indicates the US English study, "DK" the two Danish studies. eGeMAPS indicates the Extended Geneva Minimalistic Acoustic Parameter Set (Eyben et al xxx). Train and holdout indicate respectively the portions of the dataset on which the

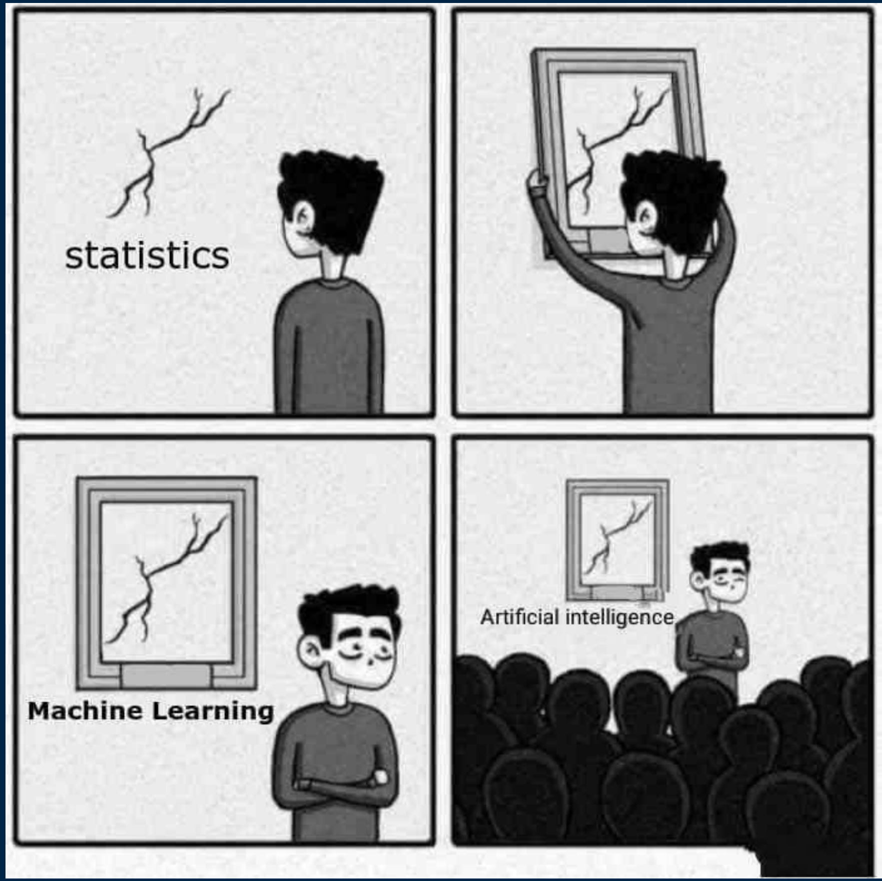
Preprint extract, w. Stine Nyhuus:

Vocal markers of Autism Spectrum Disorder: assessing the generalizability of machine learning models

TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

MACHINE LEARNING



MACHINE LEARNING

- Supervised learning
    - Continuous
    - Classification

- Unsupervised learning
    - Clustering

- Reinforcement learning

MACHINE LEARNING

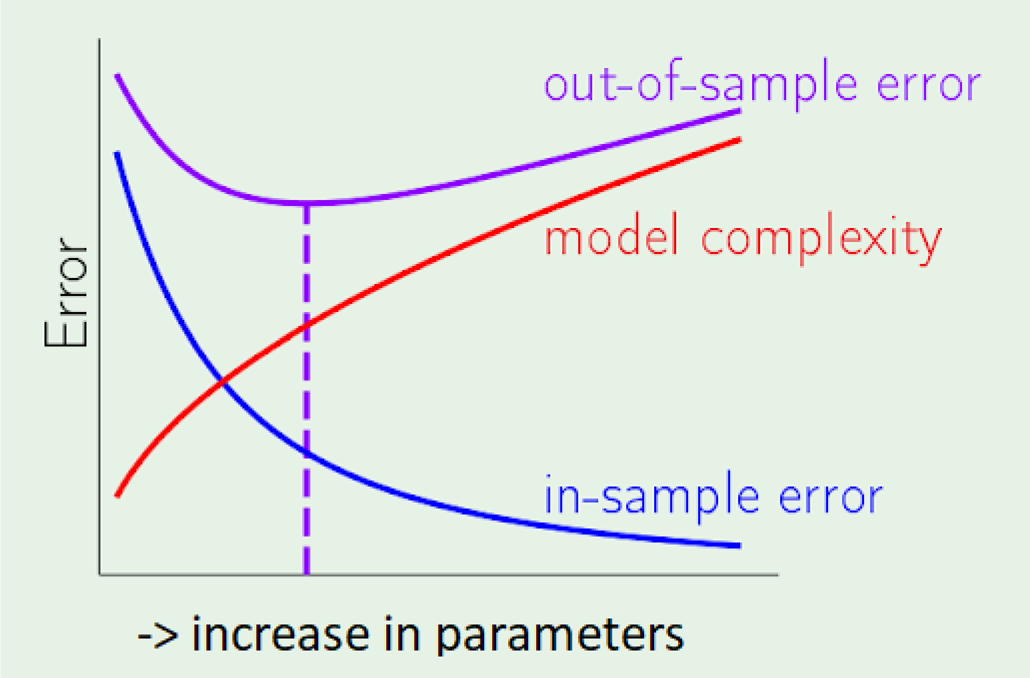
- **Supervised learning**
  - **Continuous**
  - **Classification**

- Unsupervised learning
  - Clustering

- Reinforcement learning

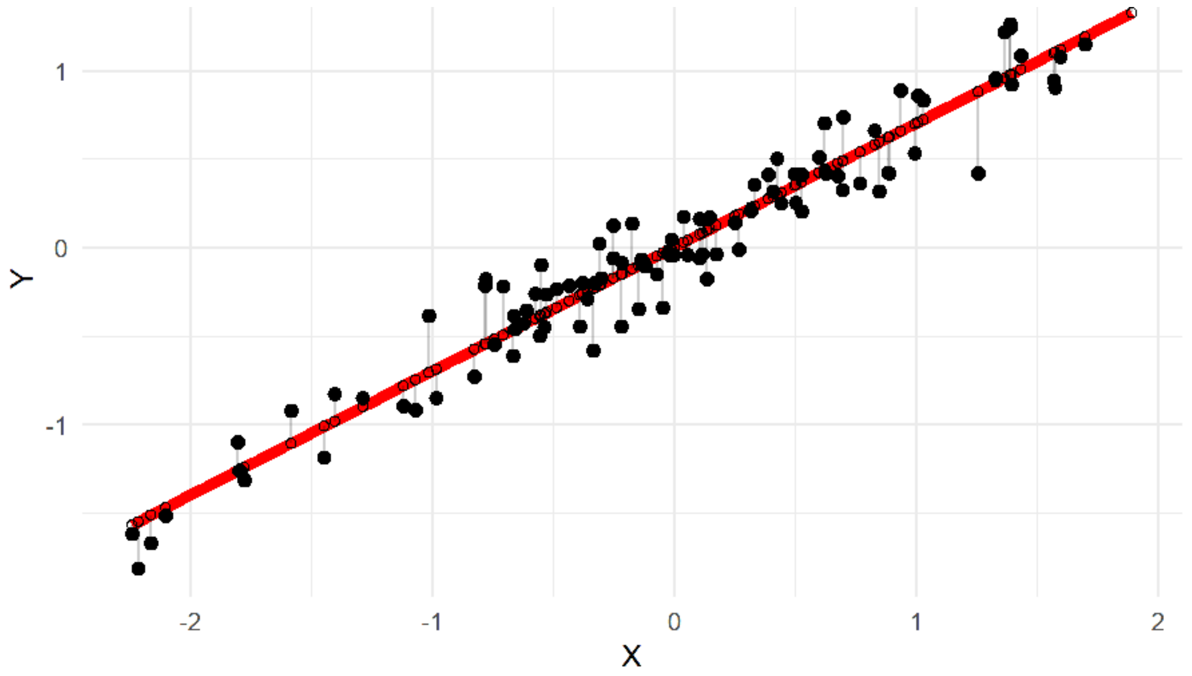
MACHINE LEARNING (SUPERVISED LEARNING)

- R^2 < AIC/BIC < **Out-of-sample error**

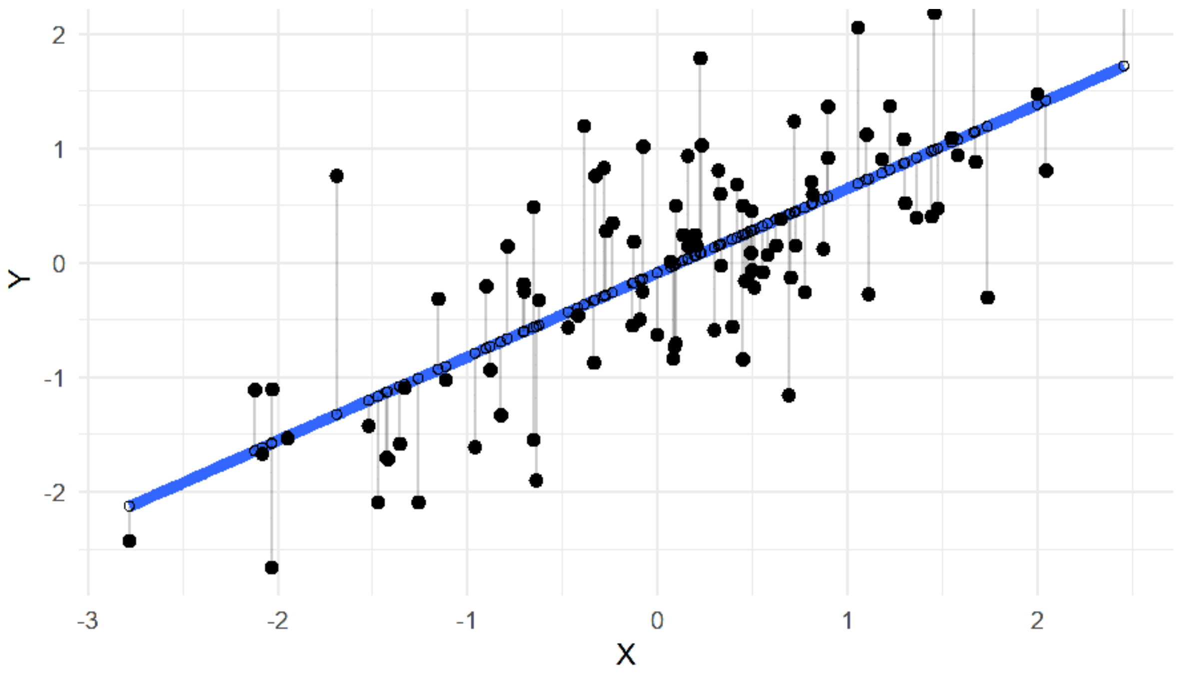


… or epochs/iterations for e.g. NN

MACHINE LEARNING (SUPERVISED LEARNING)



Within-sample-error



Out-of-sample-error

MACHINE LEARNING (SUPERVISED LEARNING)

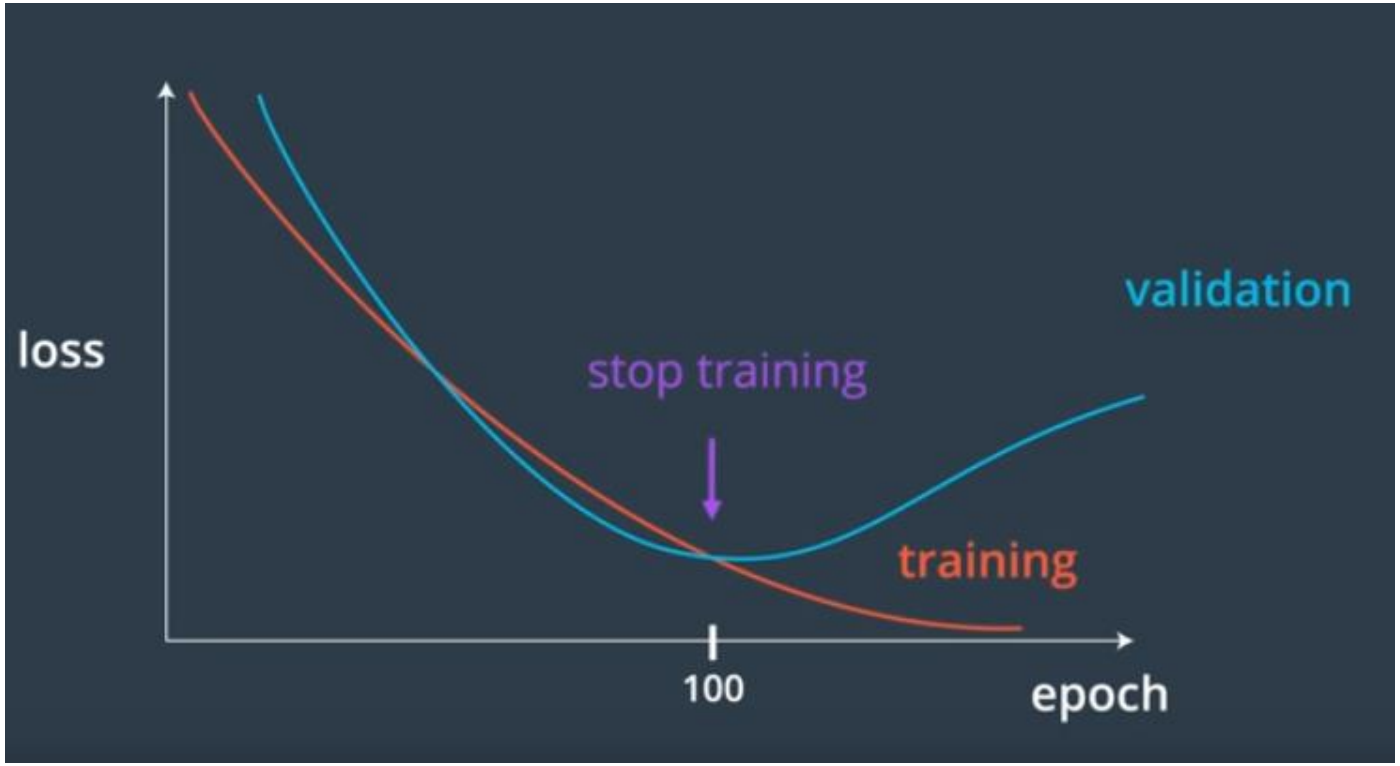
- So how do we optimize for out-of-sample-performance?

MACHINE LEARNING (SUPERVISED LEARNING)

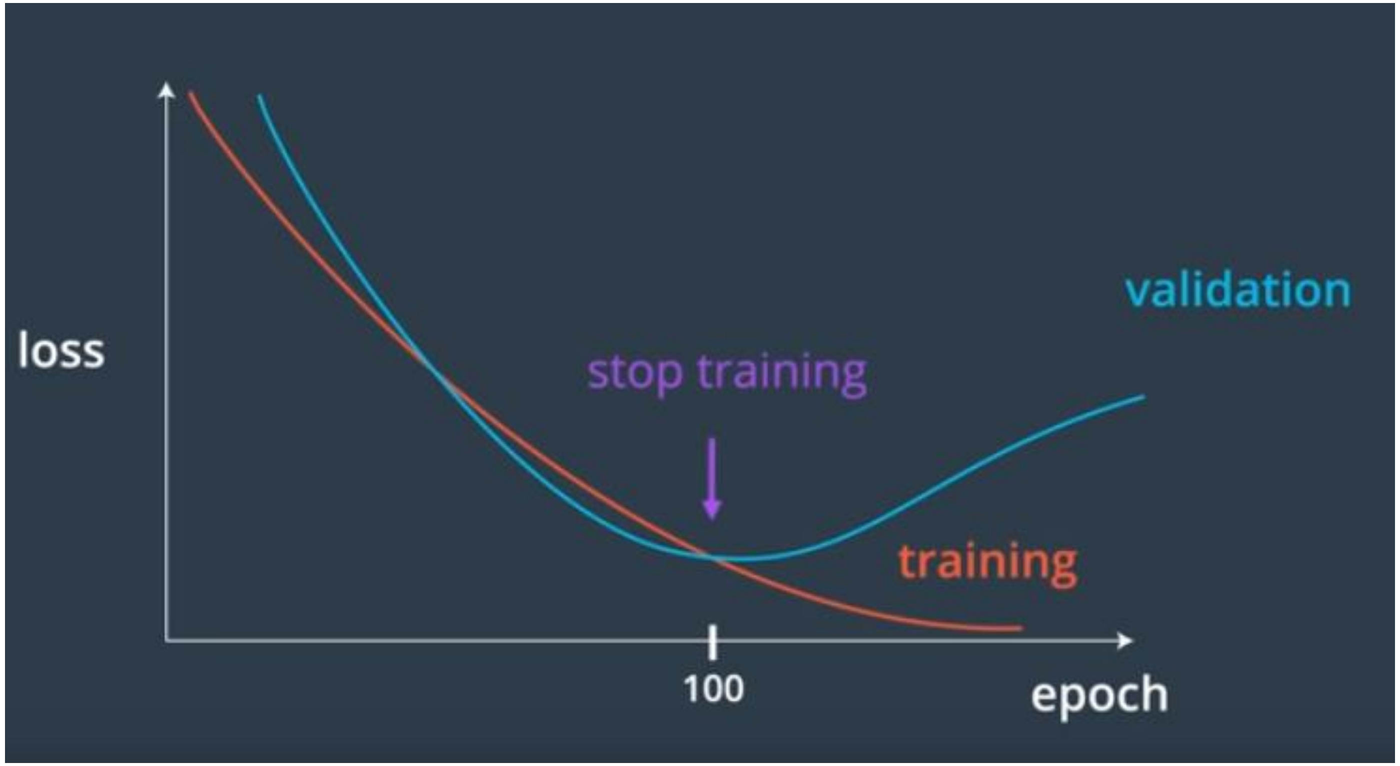
1. Split data
2. Train model
3. Validate model
4. *(Repeat 2 + 3, w. new hyperparameters (e.g. lambda))*
5. Test model (only once)

MACHINE LEARNING (SUPERVISED LEARNING)

1. Split data
2. Train model
3. Validate model
4. *(Repeat 2 + 3, w. new hyperparameters (e.g. lambda))*
5. Test model (only once)



MACHINE LEARNING (SUPERVISED LEARNING)



1. Split data
2. Train model
3. Validate model
4. *(Repeat 2 + 3, w. new hyperparameters (e.g. lambda))*
5. Test model (only once)

- Why do we only test model once?
- Disadvantages of approach? (Think CV)

MACHINE LEARNING (SUPERVISED LEARNING)

- Disadvantages of train-validation-test splits
  - Having only 1 random split: Poor estimate of out-of-sample performance (we might have variation in performance, if we were to repeat but we wouldn't know)
  - We're only using some of our data for the training (the models that we're training are suboptimal, given less data)

TODAYS PLAN

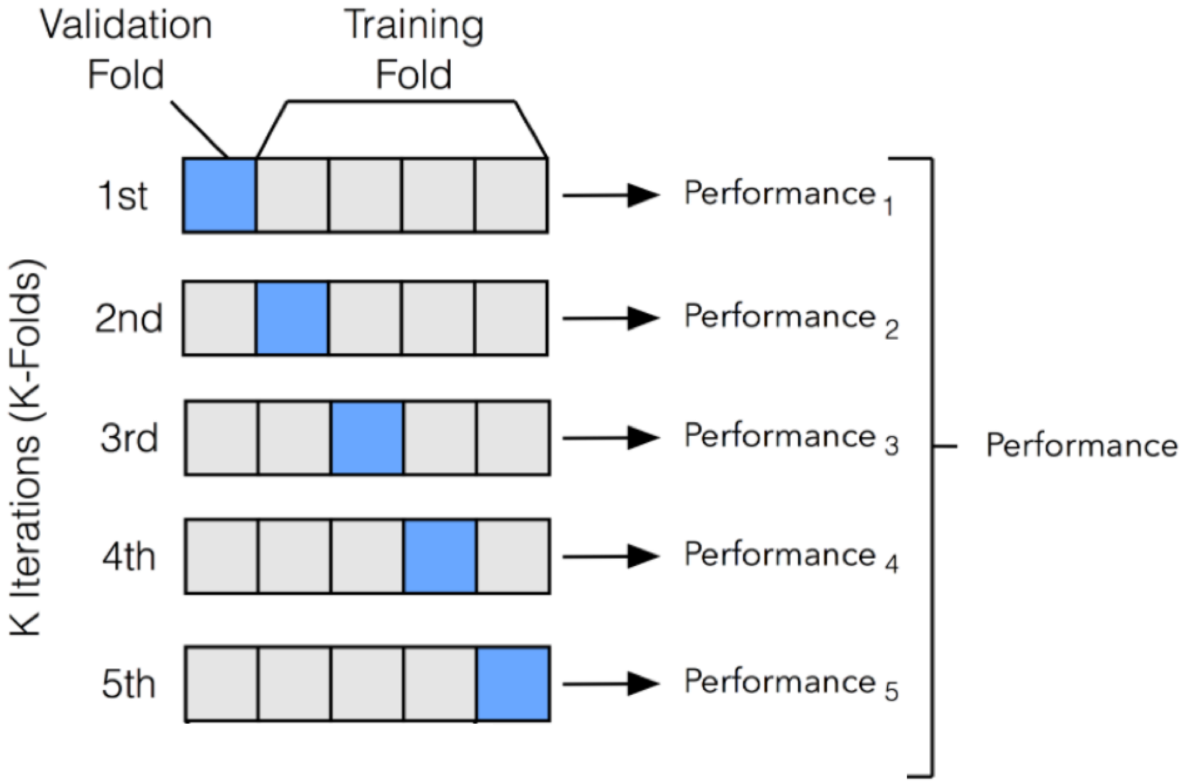
- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

CROSS-VALIDATION

CROSS-VALIDATION

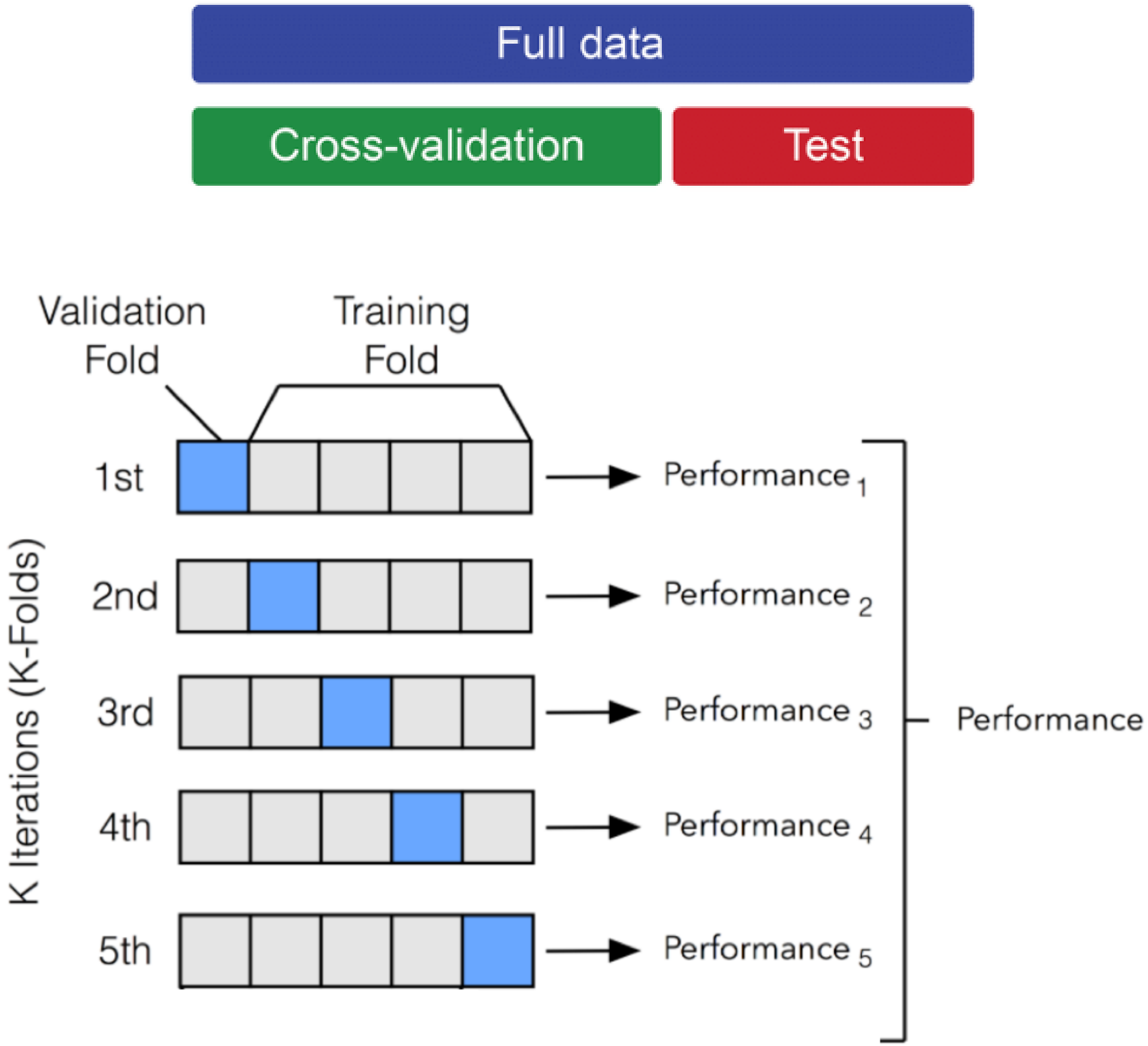
CROSS-VALIDATION

1. Split data
2. Cross-validation (train-validate with k-splits)
3. *(Repeat 2, w. new hyperparameters (e.g. lambda))*
4. Test model



CROSS-VALIDATION

- Utilizes all data
- Multiple performance scores



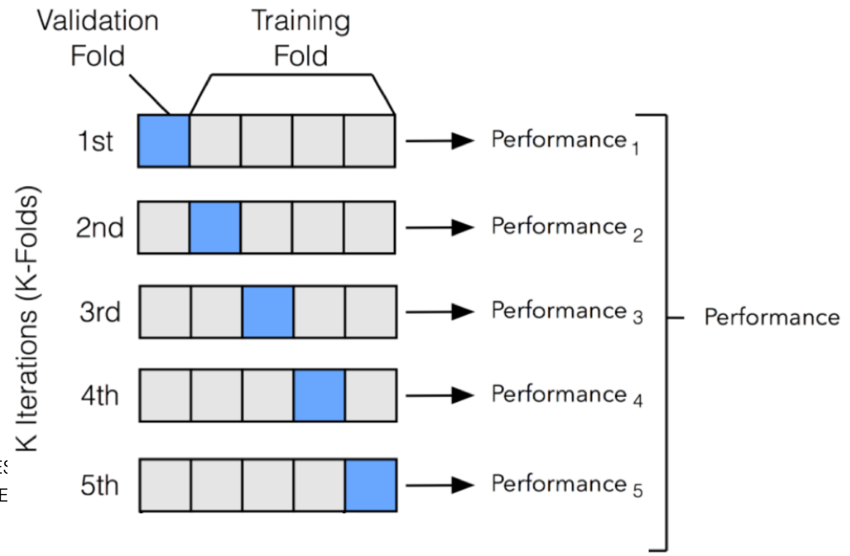
CROSS-VALIDATION

- Train-validation-test approach disadvantages:
  - Having only 1 random split -> poor estimate of out-of-sample performance
  - We're only using some of our data for the training

- Cross-validation advantages:
  - Multiple splits -> higher confidence in estimate of out-of-sample performance
  - Using all the data

=

We can better optimize for out-of-sample performance



TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

REGULARIZATION

REGULARIZATION

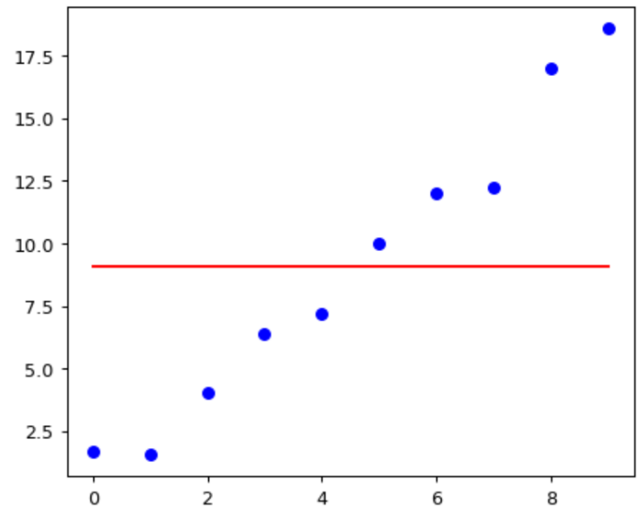
- Prior to regularization we have utilized OLS method (or similar)
  =
- Global minimum for solution space, for e.g.. convex function

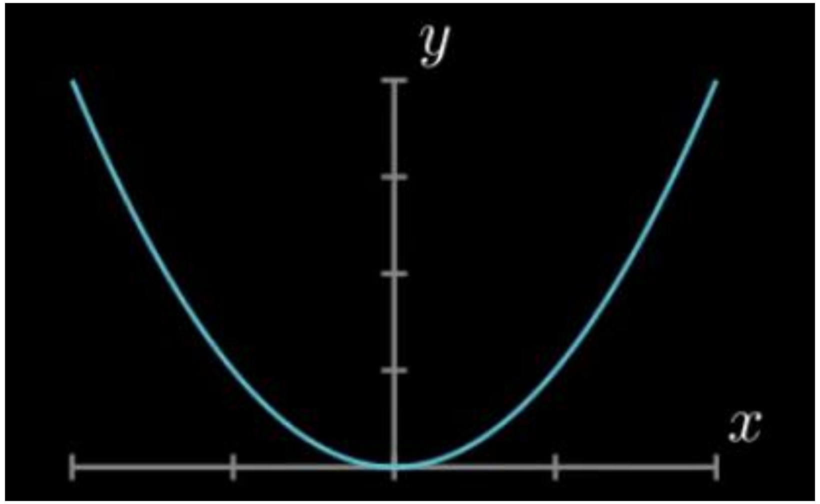
- Example: Last weeks exercises, finding intercept model

REGULARIZATION

- Prior to regularization we have utilized OLS method (or similar)
  =
- Global minimum for solution space, for e.g.. convex function

- Example: Last weeks exercises, finding intercept model





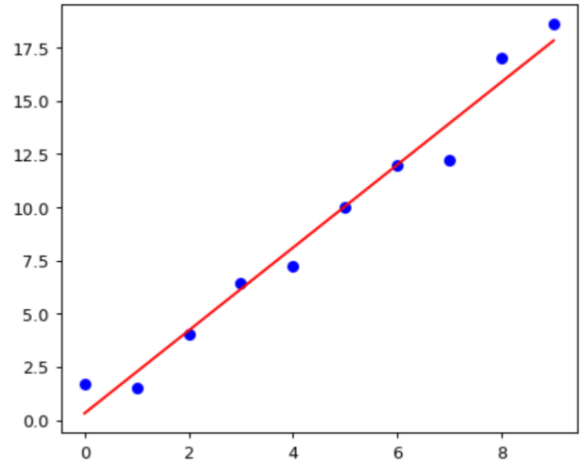
y = loss function

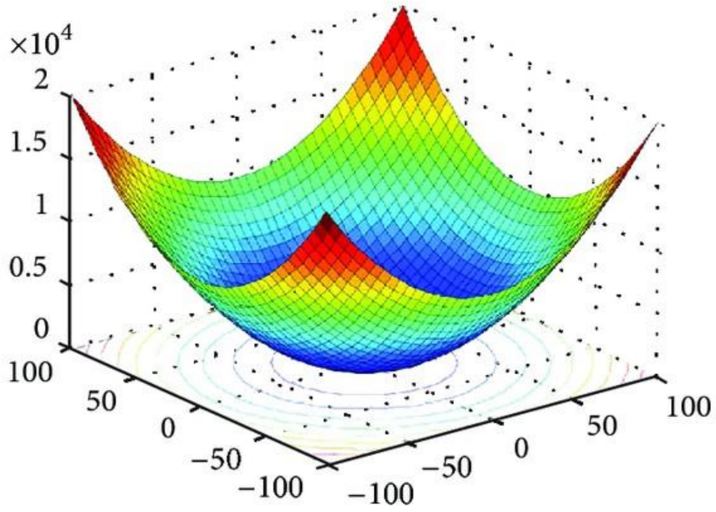
x = beta0

REGULARIZATION

- Prior to regularization we have utilized OLS method only (or similar)
  =
- Global minimum for solution space, for convex function

- Example: Last weeks exercises, finding intercept model…
  - … or model including slope





y = loss function

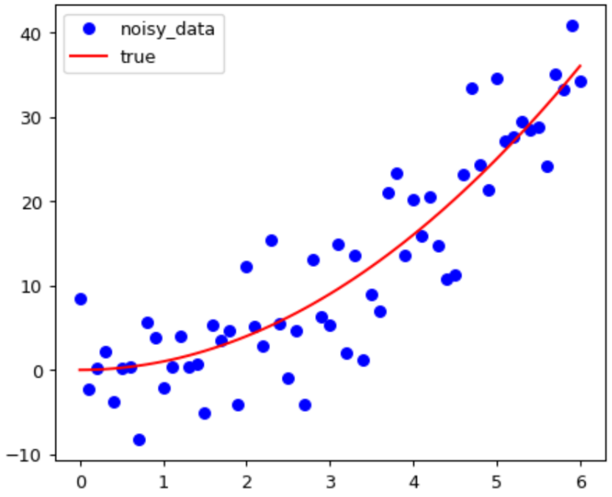
x = beta0

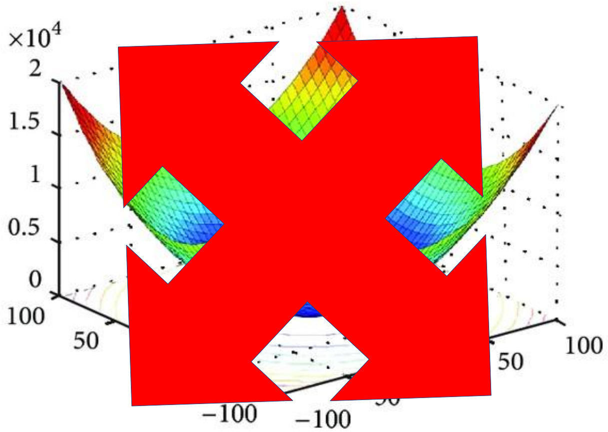
z = beta1

REGULARIZATION

- Prior to regularization we have utilized OLS method
  =
- Global minimum for solution space, for convex function

- Example: Last weeks exercises, finding intercept model…
  - … or model including slope
  - … + independent variables or higher order polynomial fitting





y = loss function

x = beta0

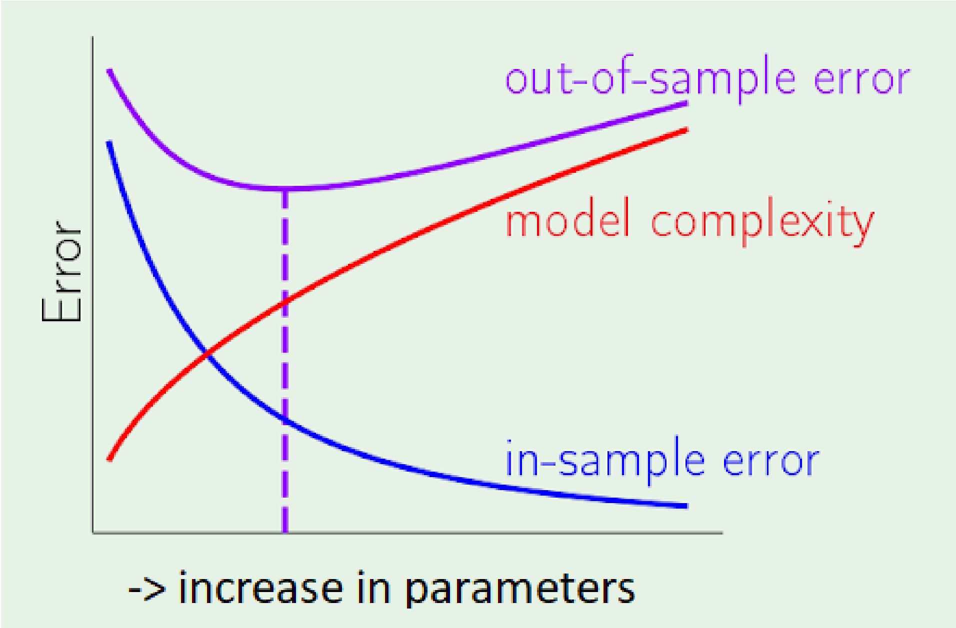
z = beta1

w = beta2

REGULARIZATION

- Problem:
  - Global minimum of training data
    ≠
  - Global minimum
- Solution?

1. Split data
2. Cross-validation (train-validate with k-splits)
3. (Repeat 2, w. new hyperparameters (e.g. lambda))
4. Test model



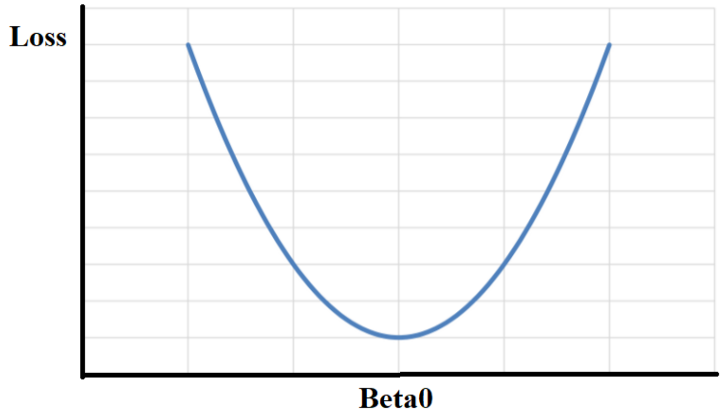
REGULARIZATION

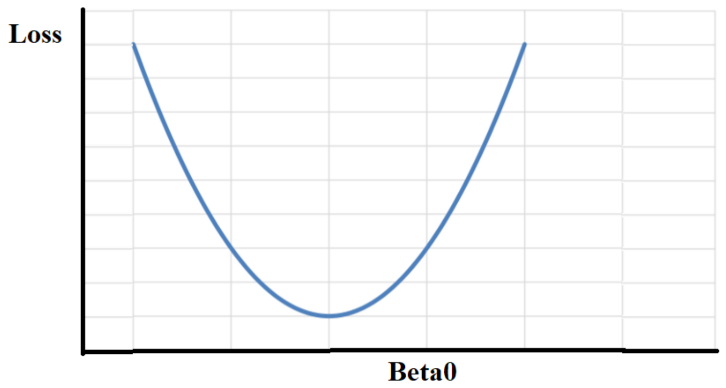
- Still finding global minimum in multidimensional space
- … but new loss function

- L2 (Ridge): $$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$
- L1 (LASSO): $$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} |w_i|$$
- ElasticNet: $$Loss = Error(y, \hat{y}) + \lambda \left(\frac{1-\alpha}{1} \sum_{j=1}^{m} \hat{\beta}_j^{\,2} + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

REGULARIZATION

- Still finding global minimum in multidimensional space
- … but new loss function





- L2 (Ridge): $Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$
- L1 (LASSO): $Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} |w_i|$
- ElasticNet: $Loss = Error(y, \hat{y}) + \lambda \left(\frac{1-\alpha}{1} \sum_{j=1}^{m} \hat{\beta}_j^{\,2} + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$

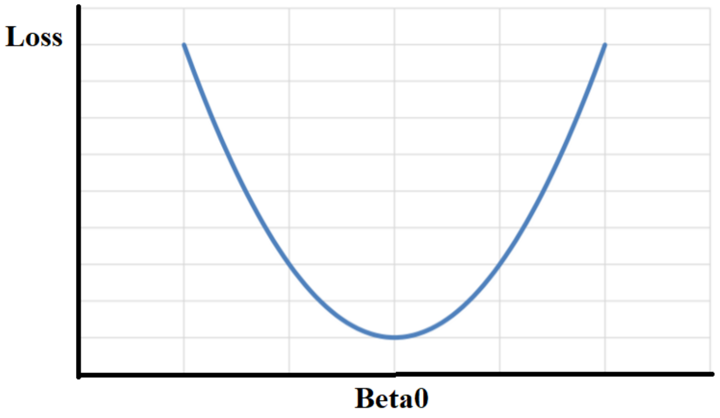
REGULARIZATION

- Still finding global minimum in multidimensional space
- … but new loss function
- … which in turn means new solution space and new global minimum
- *(not "just" setting to zero -> some confusion from last lecture)*

REGULARIZATION

- Still finding global minimum in multidimensional space
- … but new loss function
- … which in turn means new solution space and new global minimum

- Loss = OLS



- Loss = OLS + penalty

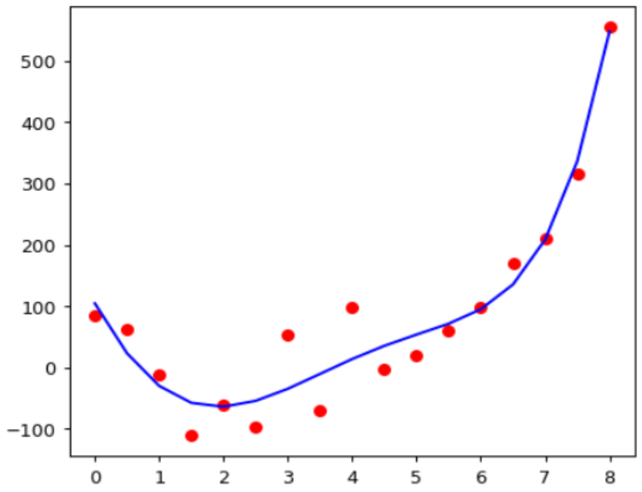
- Large beta -> harder penalty
- Large λ -> harder penalty
  - *(larger change on right hand plots)*

REGULARIZATION

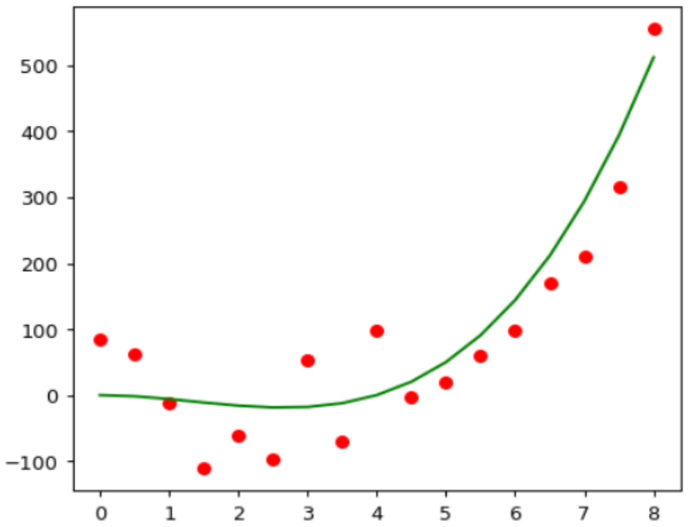
- … So lower betas with regularization
    - Why might this be good?

REGULARIZATION

- … So lower betas with regularization
  - Why might this be good?



- Blue = better fit on training

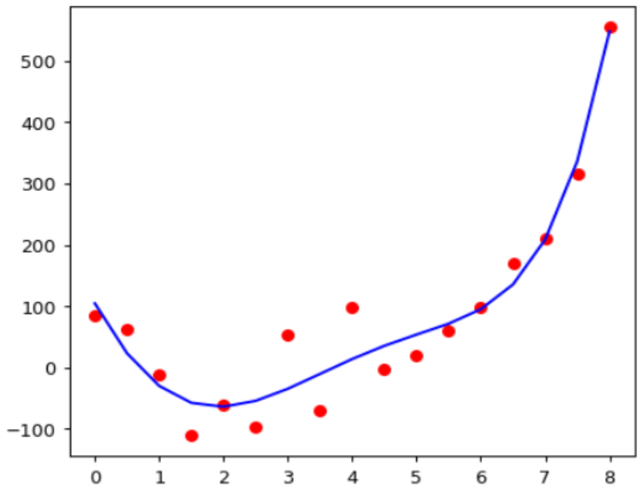


- Green = Better fit on test.

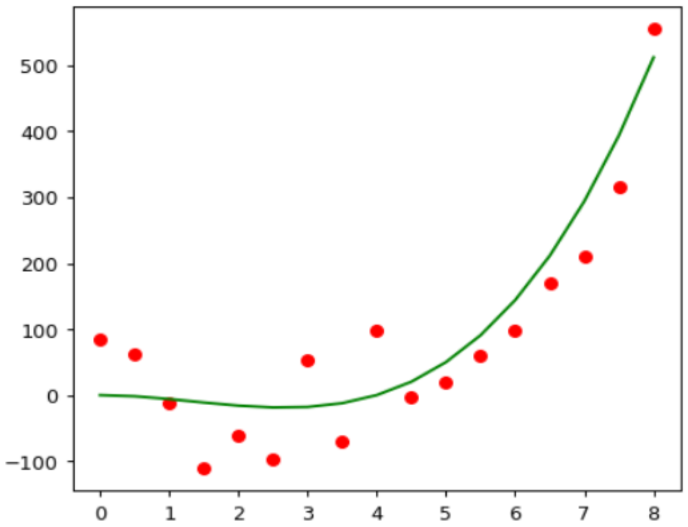
  Or is it? We have to CV/validate to find out!

REGULARIZATION

- … So lower betas with regularization
  - Why might this be good?
  - Reduction in collinearity
  - "Curves are less curvy" for polynomials



- Blue = better fit on training



- Green = Better fit on test.

  Or is it? We have to CV/validate to find out!

REGULARIZATION

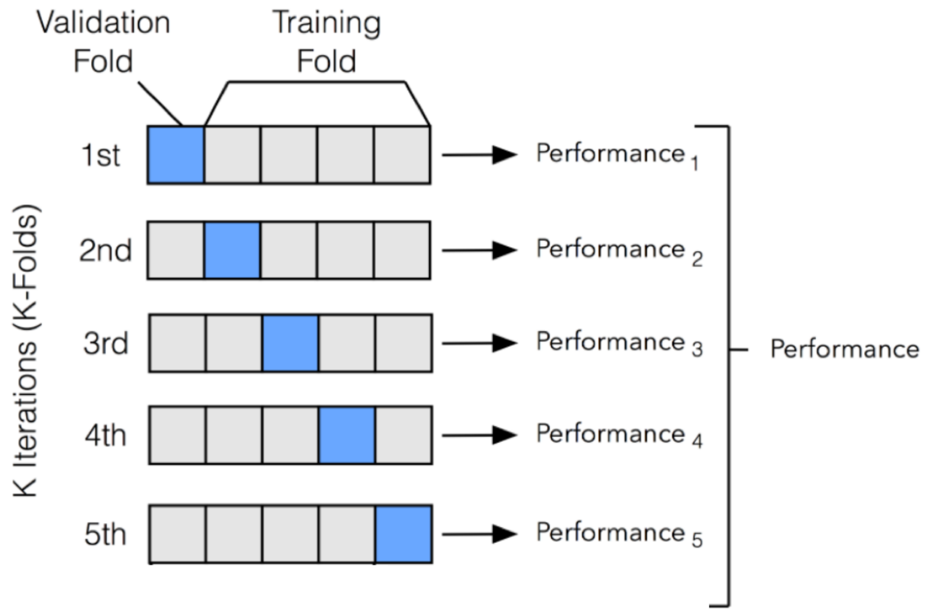
- Ridge (L2)
  - Betas approach 0 asymptotically
  - Behaviour:
    - If two predictors are correlated, both betas will be low
- LASSO (L1)
  - Minimum solution of cost function may have beta = 0 (we're not setting them to zero "afterwards")
  - May be used for feature selection
  - Behaviour:
    - If two predictors are correlated, one may be 0

REGULARIZATION

- Which lambda?
  - Optimize to out-of-sample performance

REGULARIZATION

- Which lambda?
  - Optimize to out-of-sample performance
  - (Using grid search)

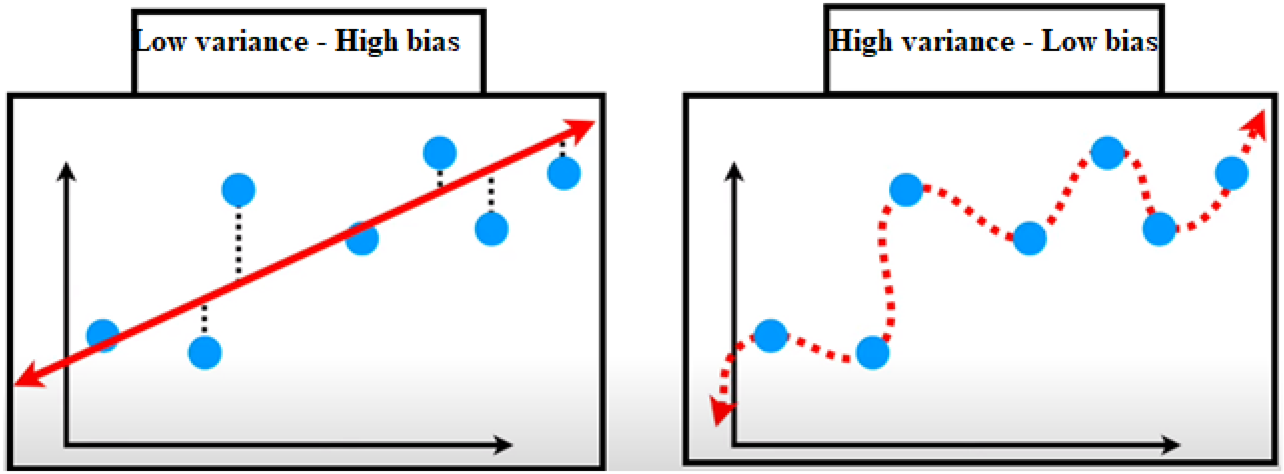


BIAS AND VARIANCE

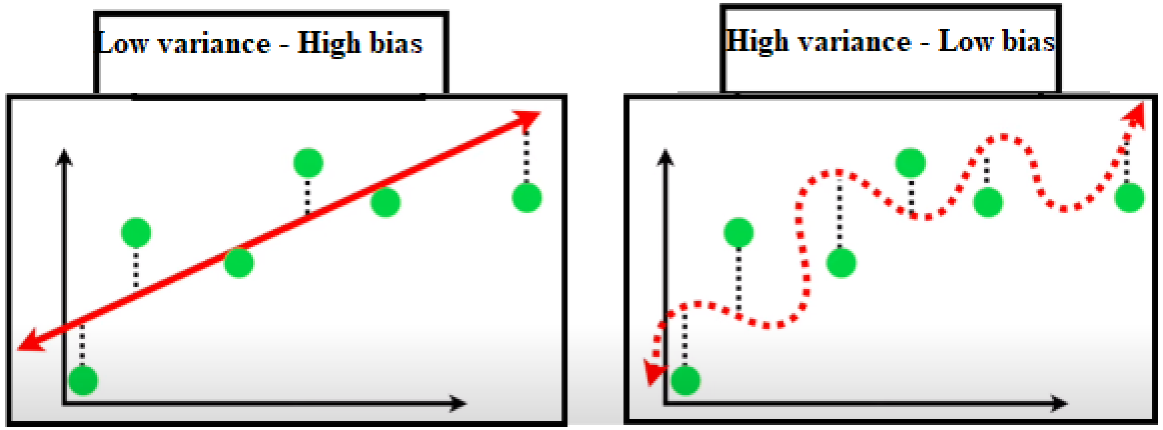
BIAS AND VARIANCE

- High variance, low bias:
  - When testing to new datasets, accuracy varies
  - Sometimes predicts well, other times poorly

- Low variance, high bias:
  - Consistent prediction accuracy
  - Squared residuals similar across testing datasets



Blue = train set



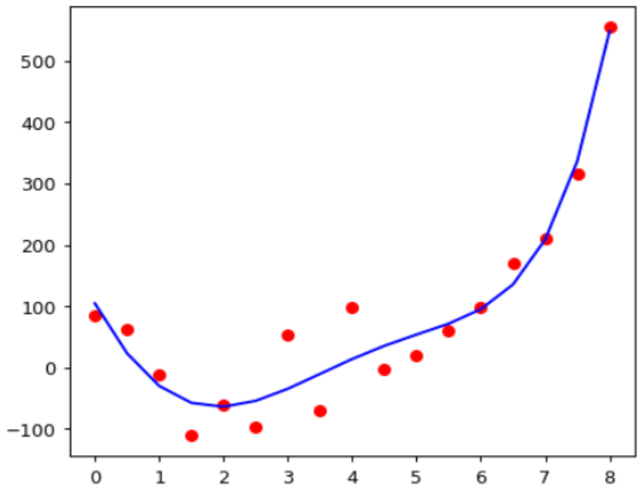
Green = test set

BIAS AND VARIANCE

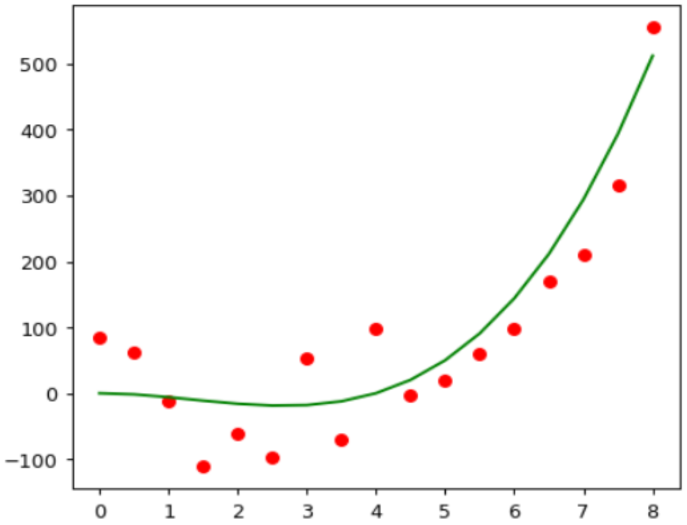
- Regularization and bias/variance
- I simulated some data + models, to show what shrunk betas may look like

BIAS AND VARIANCE

- Regularization and bias/variance
- I simulated some data + models, to show what shrunk betas may look like



- Blue = better fit on training



- Green = Better fit on test.

  Or is it? We have to CV/validate to find out!

TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

EXERCISE TIPS

EXERCISE TIPS

- List comprehensions
- *(if else statements may be included)** ***draw***

```
new_values = []
for v in values:
        new_v=v+7
        new_values.append(new_v)
```

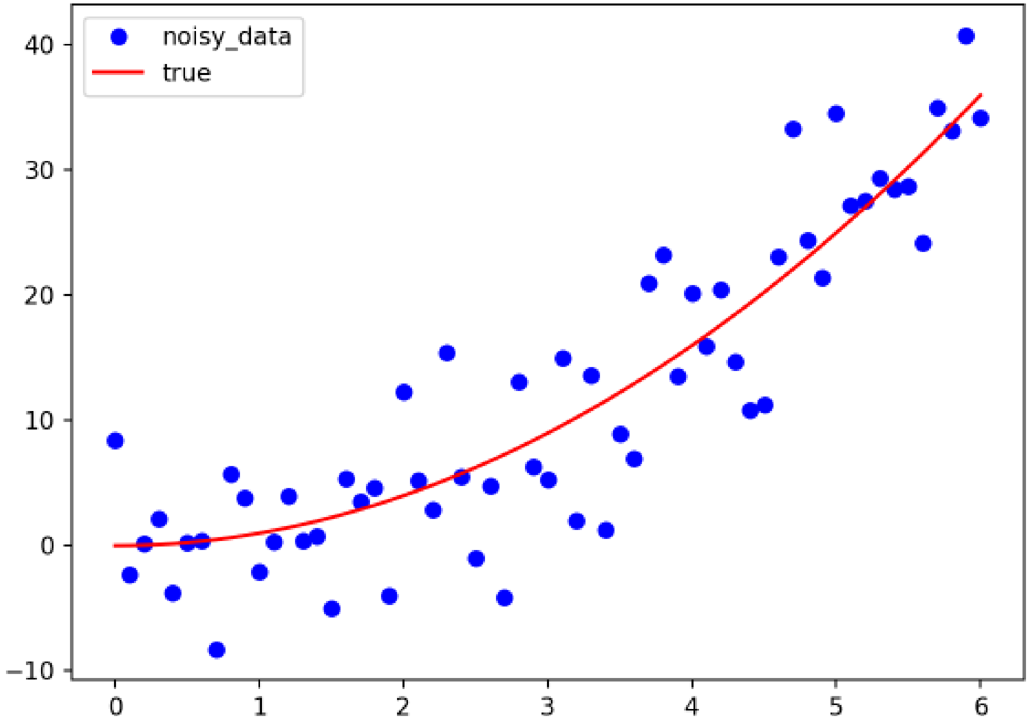
=

```
new_values = [v + 7 for v in values]
```

EXERCISE TIPS

- Plots
  - plt.figure() initializes new plot - skip, if you want new layers on top
  - No reason to use “plt.scatter()”

```
plt.figure() # Initialize new plot
plt.plot(x, y_noise, 'bo')
plt.plot(x, y_true, 'r-')
plt.legend(['noisy_data', 'true'])
plt.show()
```



EXERCISE TIPS

- Predictor variable formats in sklearn:
  - x.shape(50,) != x.shape(50, 1)

- How to reshape dimensions?
  - x.reshape(-1, 1)
  - x.reshape(50, 1)

$$\begin{pmatrix} \texttt{y1hat} \\ \texttt{y2hat} \\ \texttt{y3hat} \end{pmatrix} = \begin{pmatrix} 1 & 1.2 & 1.44 \\ 1 & 2.4 & 5.76 \\ 1 & 3.6 & 12.96 \end{pmatrix} \times \begin{pmatrix} \texttt{b0} \\ \texttt{b1} \\ \texttt{b2} \end{pmatrix}$$

EXERCISE TIPS

- Other functions you may want to use
  - np.arange(0,10,1)
  - np.random.normal(loc=0, scale=1, size=10)

EXERCISE TIPS

- Sci-kit learn documentation
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

TODAYS PLAN

- Catch-up + disclaimer
- Machine learning overview
- Cross-validation
- Regularization
- Bias and variance
- Exercise tips
- Exercise work

EXERCISES

EXERCISES

1. Pull exercises from upstream
2. Create copy of practical exercise + give unique filename ending
3. Work on new copied version of exercise
4. *(Feedback; classes do not match exercises -> questions after class? code-specific "questions" cryptpad)*

EXERCISES

Sci-kit learn documentation

x.shape(50,) != x.shape(50, 1)

X.reshape(-1, 1)

1. Pull exercises from upstream
2. Create copy of practical exercise + give unique filename ending
3. Work on new copied version of exercise
4. *(Feedback; classes do not match exercises -> questions after class? code-specific "questions" cryptpad)*

```
plt.figure() # Initialize new plot
plt.plot(x, y_noise, 'bo')
plt.plot(x, y_true, 'r-')
plt.legend(['noisy_data', 'true'])
plt.show()
```

List comprehensions

$$\begin{pmatrix} y1hat \\ y2hat \\ y3hat \end{pmatrix} = \begin{pmatrix} 1 & 1.2 & 1.44 \\ 1 & 2.4 & 5.76 \\ 1 & 3.6 & 12.96 \end{pmatrix} * \begin{pmatrix} b0 \\ b1 \\ b2 \end{pmatrix}$$

np.arange(0,10,1)

np.random.normal(loc=0, scale=1, size=10)

AARHUS UNIVERSITY

Feedback?