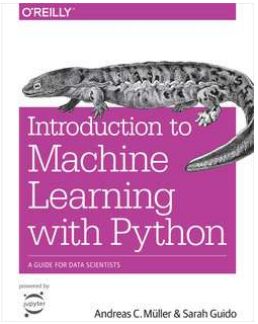
Applied Machine Learning
Controlling complexity

BSc course Informatiekunde 2026
https://staff.fnwi.uva.nl/a.visser/education/AML

Arnoud Visser
Intelligent Robotics Lab & Computer Vision Lab
Informatics Institute
Universiteit van Amsterdam
A.Visser@uva.nl

Illustrations courtesy of Maarten Marx, Sarah Guido, Yolanda Hagar, and many others.



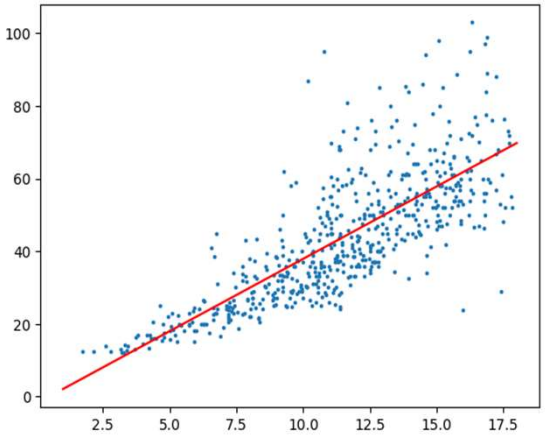
Section 2.2-2.3

Three ways of modelling

- Modelling 775 datapoints

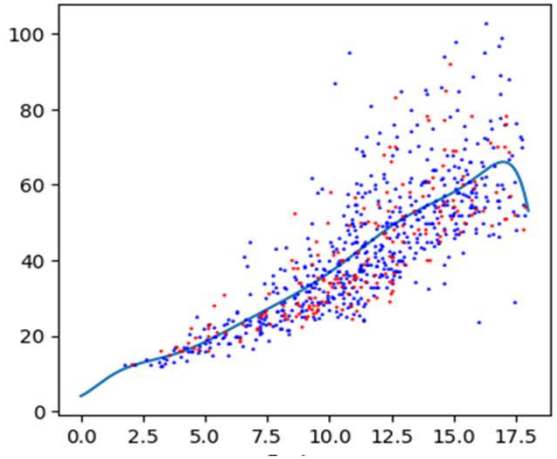
regensburg_pediatric_appendicitis

Linear model



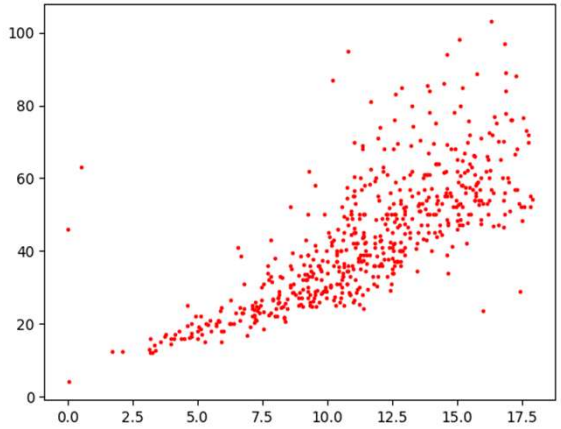
2 values ($y = \alpha + \beta x$)

Polynomial model



10 values ($y = \alpha + \ldots + \kappa x^9$)

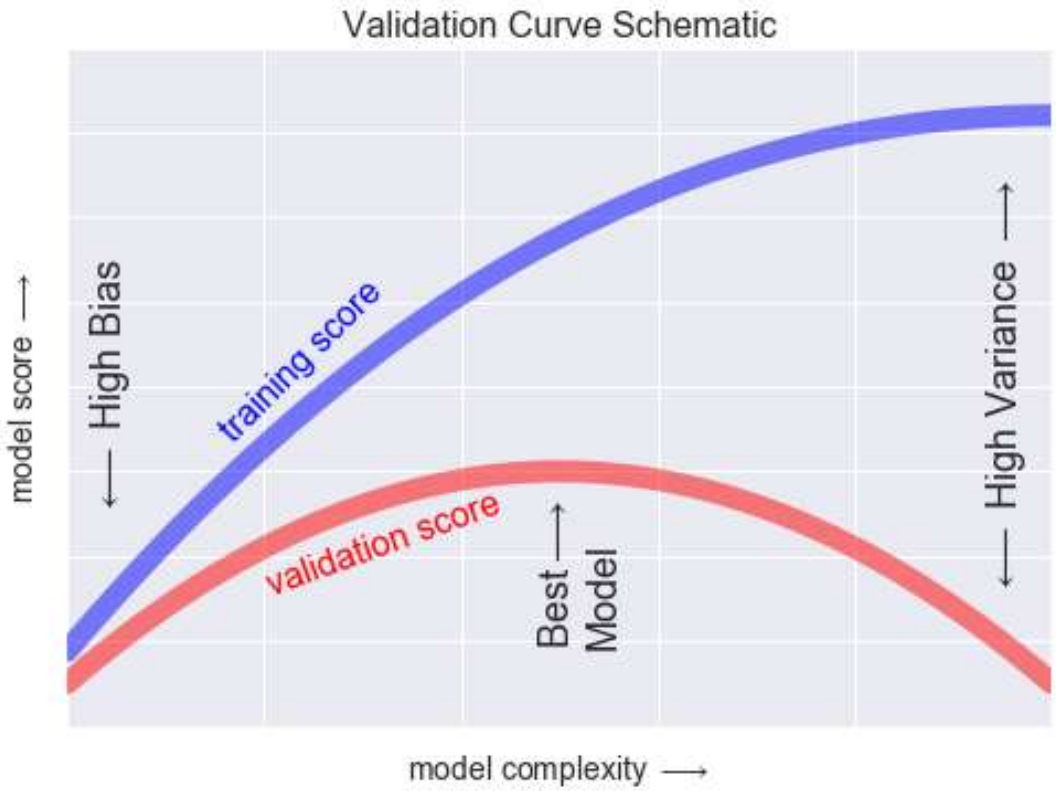
1−nn model



584 training values

What is good model?

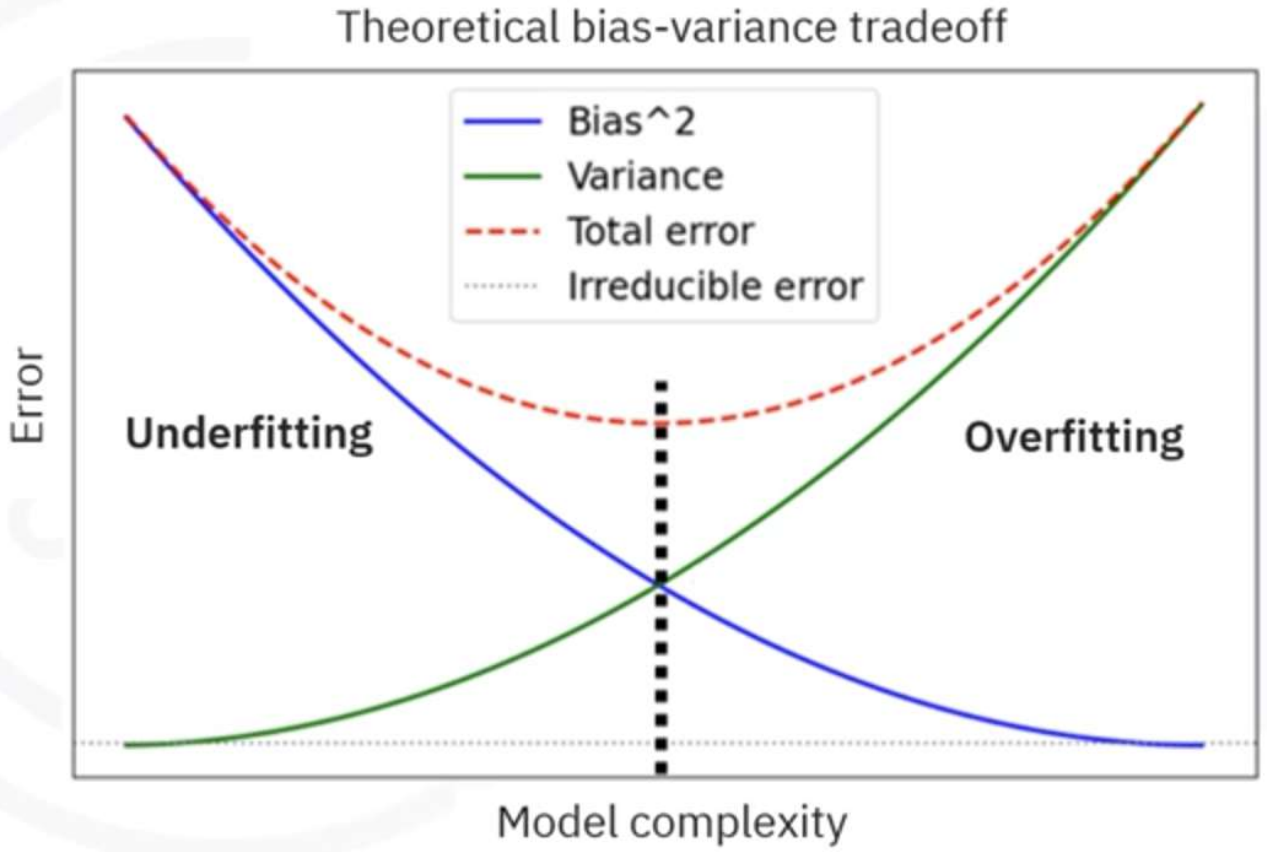
- Look at the validation curve



Page 366

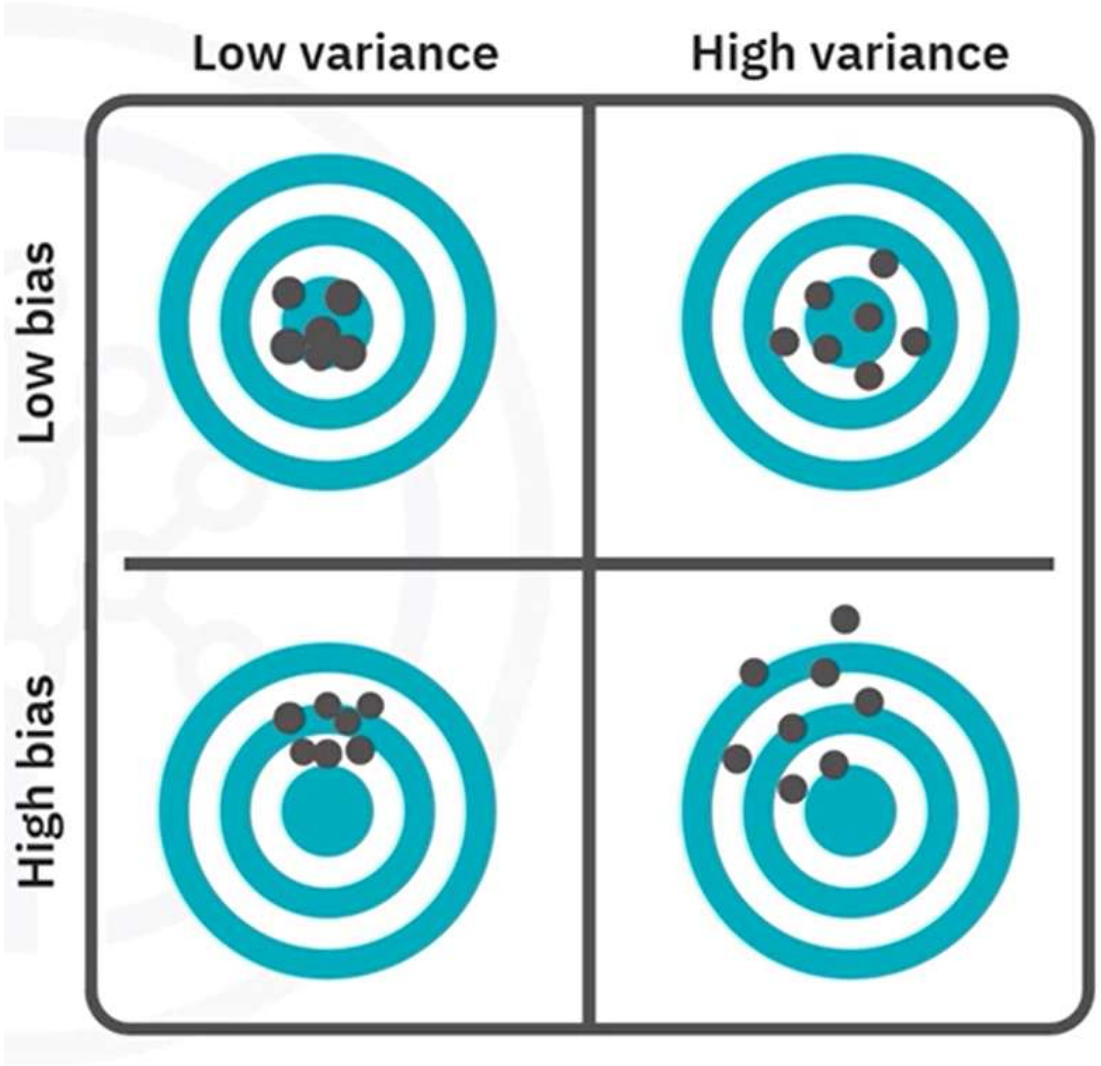
What is good model?

- Look at the RMSE



Courtesy Joseph Santarcangelo & Jeff Grossman

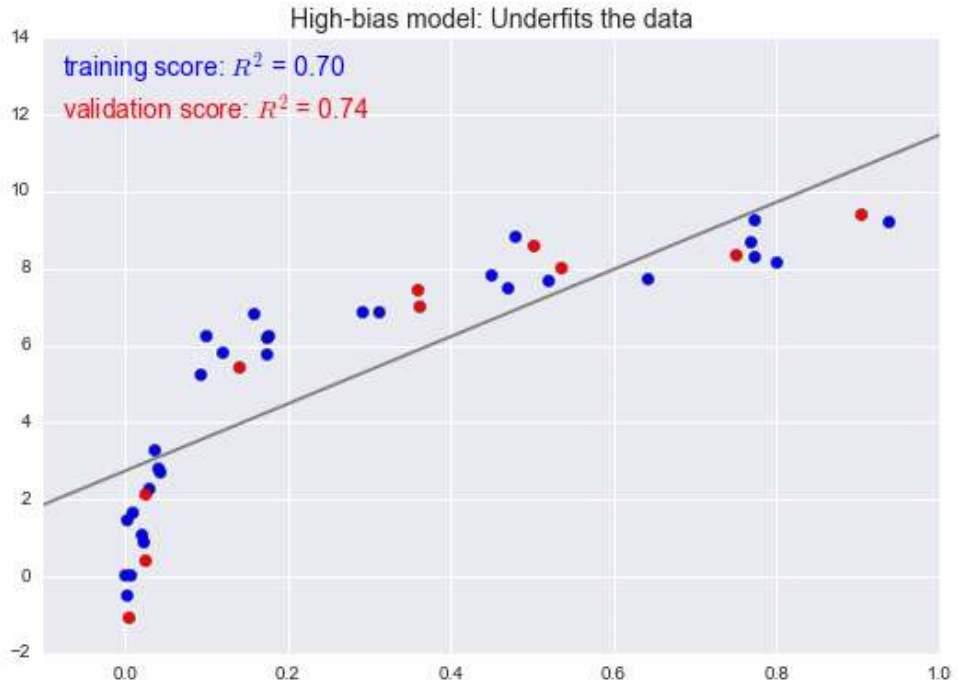
Bias versus Variance



Courtesy Joseph Santarcangelo & Jeff Grossman

High-bias model

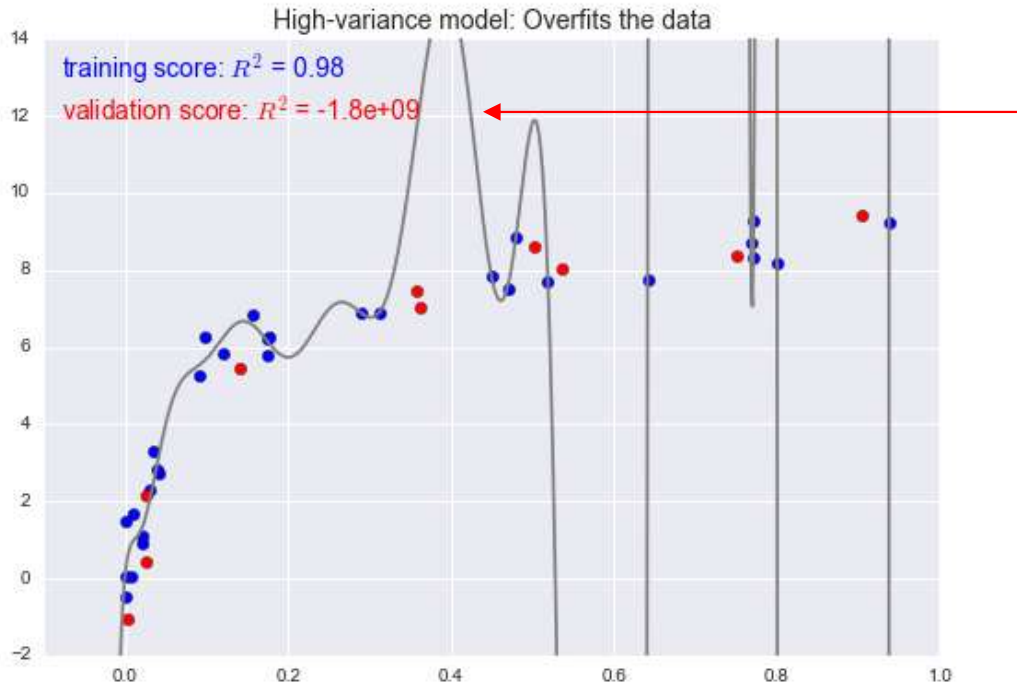
- R^2 : measures the proportion of variability in Y explained by the regression model.



Page 365

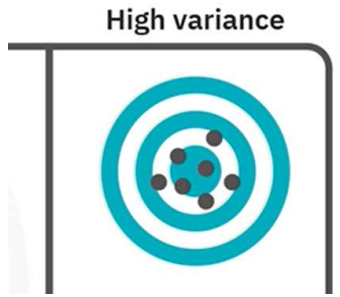
High-variance model

- R^2 : measures the proportion of variability in Y explained by the regression model.



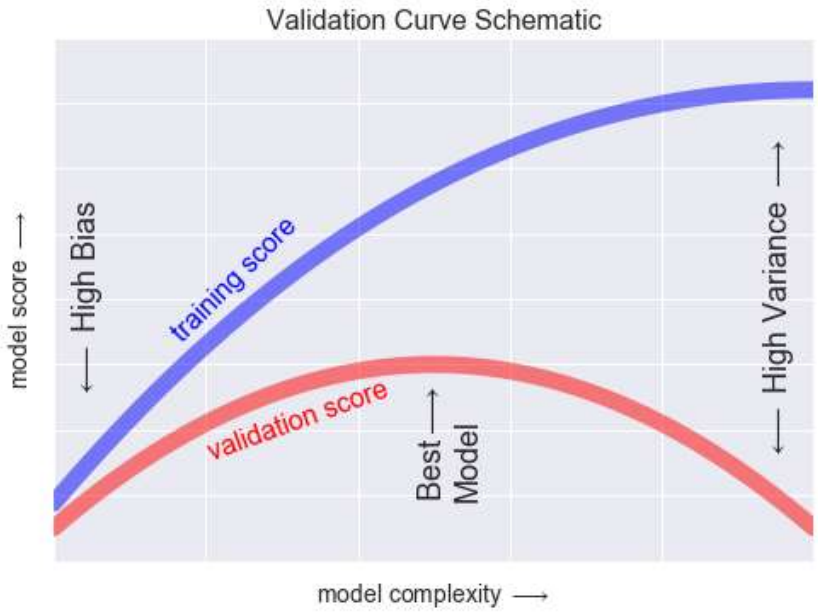
Negative R^2 : you better could have taken the mean()

Page 365



What is good model?

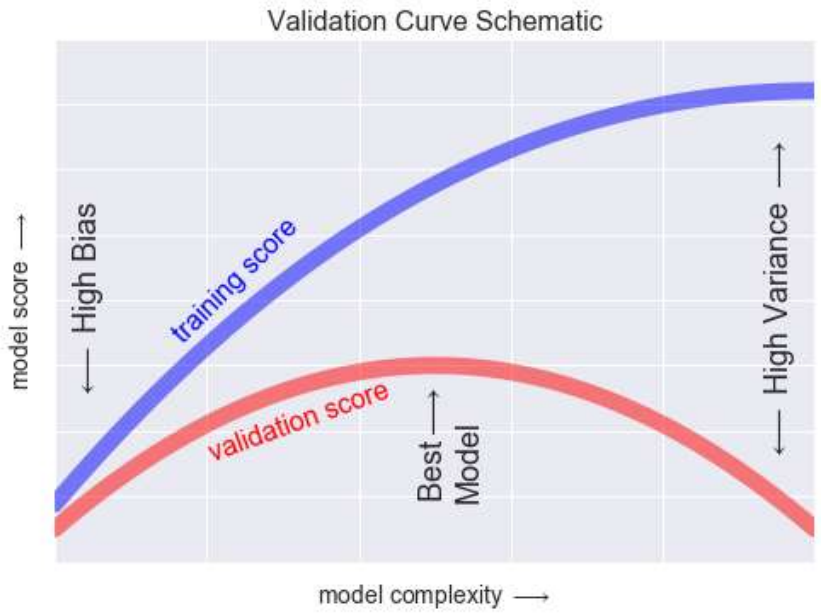
- How could you improve the model?



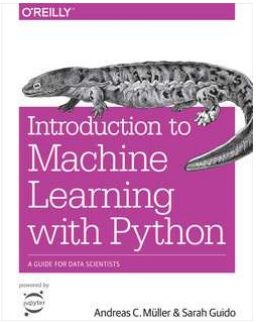
- Use a more complicated / more flexible model
- Use a less complicated / less flexible model

Control complexity

- Does a hyperparameter α exist which tunes the complexity?



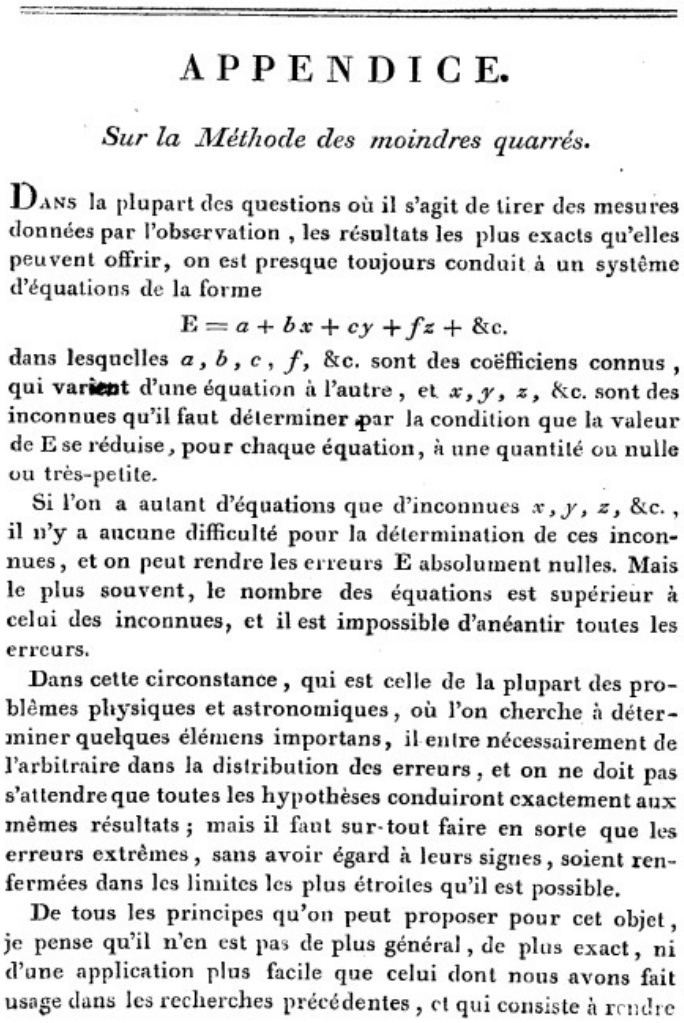
- For linear regression it does:
  - Ridge α
  - Lasso α



Page 51

Introduction to Machine Learning

Origin of `fit(X_train, y_train)` : “least square method”



APPENDICE.

Sur la Méthode des moindres quarrés.

DANS la plupart des questions où il s'agit de tirer des mesures données par l'observation, les résultats les plus exacts qu'elles peuvent offrir, on est presque toujours conduit à un système d'équations de la forme

$$E = a + bx + cy + fz + \&c.$$

dans lesquelles $a, b, c, f,$ &c. sont des coëfficiens connus, qui varient d'une équation à l'autre, et $x, y, z,$ &c. sont des inconnues qu'il faut déterminer par la condition que la valeur de E se réduise, pour chaque équation, à une quantité ou nulle ou très-petite.

Si l'on a autant d'équations que d'inconnues $x, y, z,$ &c., il n'y a aucune difficulté pour la détermination de ces inconnues, et on peut rendre les erreurs E absolument nulles. Mais le plus souvent, le nombre des équations est supérieur à celui des inconnues, et il est impossible d'anéantir toutes les erreurs.

Dans cette circonstance, qui est celle de la plupart des problèmes physiques et astronomiques, où l'on cherche à déterminer quelques élémens importans, il entre nécessairement de l'arbitraire dans la distribution des erreurs, et on ne doit pas s'attendre que toutes les hypothèses conduiront exactement aux mêmes résultats; mais il faut sur-tout faire en sorte que les erreurs extrêmes, sans avoir égard à leurs signes, soient renfermées dans les limites les plus étroites qu'il est possible.

De tous les principes qu'on peut proposer pour cet objet, je pense qu'il n'en est pas de plus général, de plus exact, ni d'une application plus facile que celui dont nous avons fait usage dans les recherches précédentes, et qui consiste à rendre

Simple linear regression

linear model

$$y = \alpha + \beta x$$

Several points with error ε_i

$$y_i = \alpha + \beta x_i + \varepsilon_i .$$

Error ε_i should be as small as possible

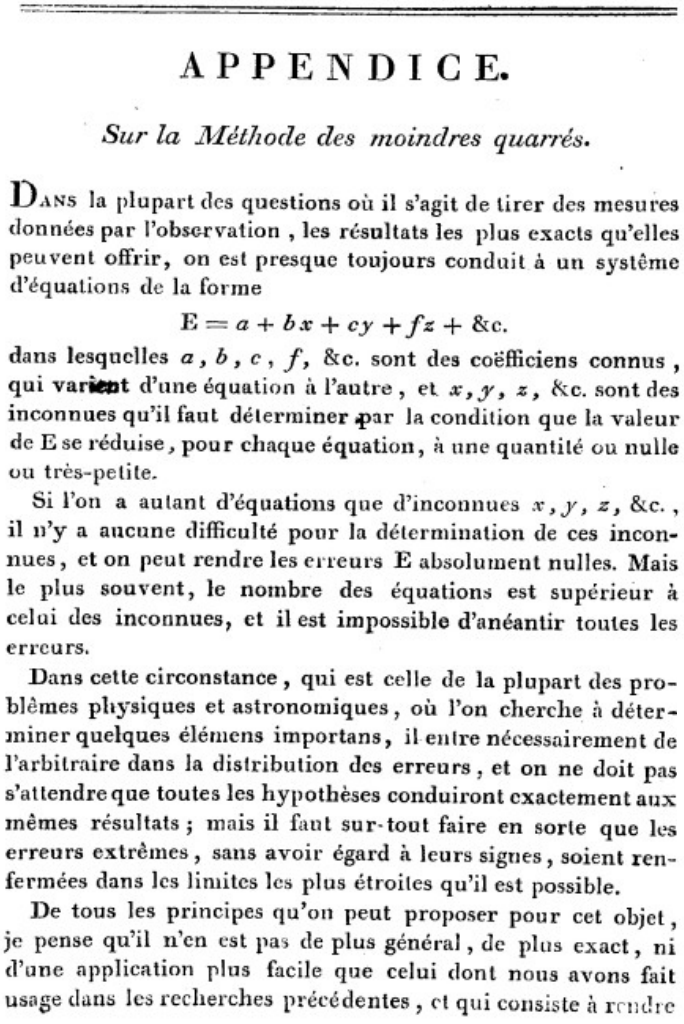
$$\hat{\varepsilon}_i = y_i - \alpha - \beta x_i .$$

Minimize the sum of all errors ε_i

$$\sum_{i=1}^{n} \hat{\varepsilon}_i^{\,2} = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 .$$

Ridge – additional constraints

Origin of `fit(X_train, y_train)` : “least square method” extended



APPENDICE.

Sur la Méthode des moindres quarrés.

DANS la plupart des questions où il s'agit de tirer des mesures données par l'observation, les résultats les plus exacts qu'elles peuvent offrir, on est presque toujours conduit à un système d'équations de la forme

$$E = a + bx + cy + fz + \&c.$$

dans lesquelles $a, b, c, f,$ &c. sont des coëfficiens connus, qui varient d'une équation à l'autre, et $x, y, z,$ &c. sont des inconnues qu'il faut déterminer par la condition que la valeur de E se réduise, pour chaque équation, à une quantité ou nulle ou très-petite.

Si l'on a autant d'équations que d'inconnues $x, y, z,$ &c., il n'y a aucune difficulté pour la détermination de ces inconnues, et on peut rendre les erreurs E absolument nulles. Mais le plus souvent, le nombre des équations est supérieur à celui des inconnues, et il est impossible d'anéantir toutes les erreurs.

Dans cette circonstance, qui est celle de la plupart des problèmes physiques et astronomiques, où l'on cherche à déterminer quelques élémens importans, il entre nécessairement de l'arbitraire dans la distribution des erreurs, et on ne doit pas s'attendre que toutes les hypothèses conduiront exactement aux mêmes résultats; mais il faut sur-tout faire en sorte que les erreurs extrêmes, sans avoir égard à leurs signes, soient renfermées dans les limites les plus étroites qu'il est possible.

De tous les principes qu'on peut proposer pour cet objet, je pense qu'il n'en est pas de plus général, de plus exact, ni d'une application plus facile que celui dont nous avons fait usage dans les recherches précédentes, et qui consiste à rendre

Simple Ridge

linear model

$$y = \alpha + \beta x$$

Several points with error ε_i

$$y_i = \alpha + \beta x_i + \varepsilon_i.$$

Error ε_i should be as small as possible

$$\hat{\varepsilon}_i = y_i - \alpha - \beta x_i.$$

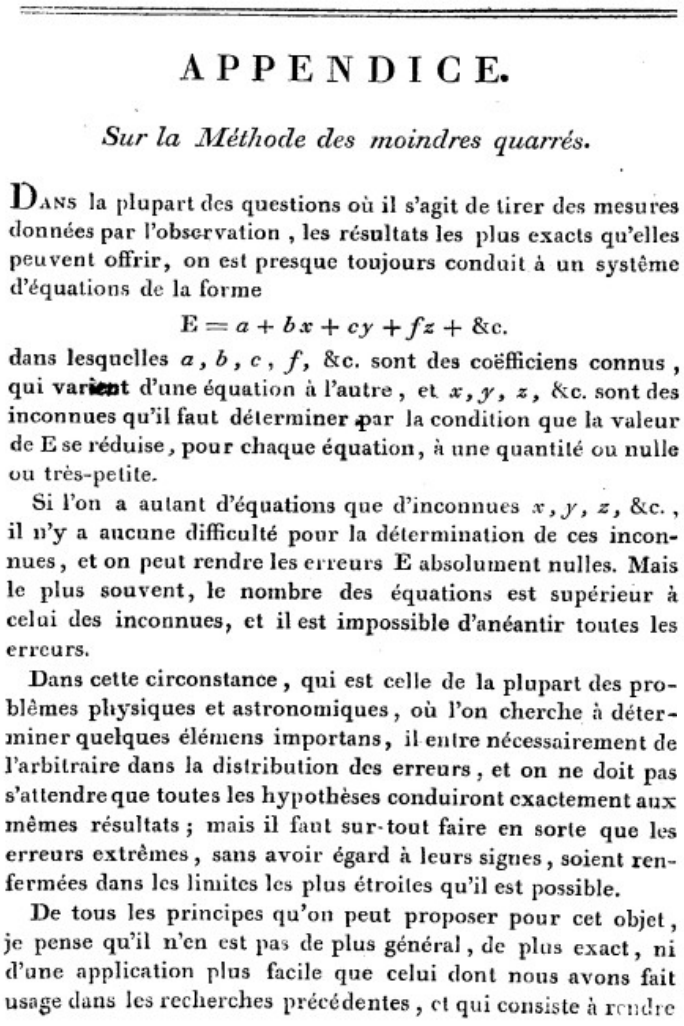
Minimize the loss

$$L2 = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 + \sum_{j=1}^{g} w_j^2$$

L2 Regularization

Lasso – additional constraints

Origin of `fit(X_train, y_train)` : “least square method” extended



APPENDICE.

Sur la Méthode des moindres quarrés.

DANS la plupart des questions où il s'agit de tirer des mesures données par l'observation, les résultats les plus exacts qu'elles peuvent offrir, on est presque toujours conduit à un système d'équations de la forme

$$E = a + bx + cy + fz + \&c.$$

dans lesquelles $a, b, c, f,$ &c. sont des coëfficiens connus, qui varient d'une équation à l'autre, et $x, y, z,$ &c. sont des inconnues qu'il faut déterminer par la condition que la valeur de E se réduise, pour chaque équation, à une quantité ou nulle ou très-petite.

Si l'on a autant d'équations que d'inconnues $x, y, z,$ &c., il n'y a aucune difficulté pour la détermination de ces inconnues, et on peut rendre les erreurs E absolument nulles. Mais le plus souvent, le nombre des équations est supérieur à celui des inconnues, et il est impossible d'anéantir toutes les erreurs.

Dans cette circonstance, qui est celle de la plupart des problèmes physiques et astronomiques, où l'on cherche à déterminer quelques élémens importans, il entre nécessairement de l'arbitraire dans la distribution des erreurs, et on ne doit pas s'attendre que toutes les hypothèses conduiront exactement aux mêmes résultats; mais il faut sur-tout faire en sorte que les erreurs extrêmes, sans avoir égard à leurs signes, soient renfermées dans les limites les plus étroites qu'il est possible.

De tous les principes qu'on peut proposer pour cet objet, je pense qu'il n'en est pas de plus général, de plus exact, ni d'une application plus facile que celui dont nous avons fait usage dans les recherches précédentes, et qui consiste à rendre

Simple Lasso(α)

linear model

$$y = \alpha + \beta x$$

Several points with error ε_i

$$y_i = \alpha + \beta x_i + \varepsilon_i.$$

Error ε_i should be as small as possible

$$\hat{\varepsilon}_i = y_i - \alpha - \beta x_i.$$

Minimize the loss

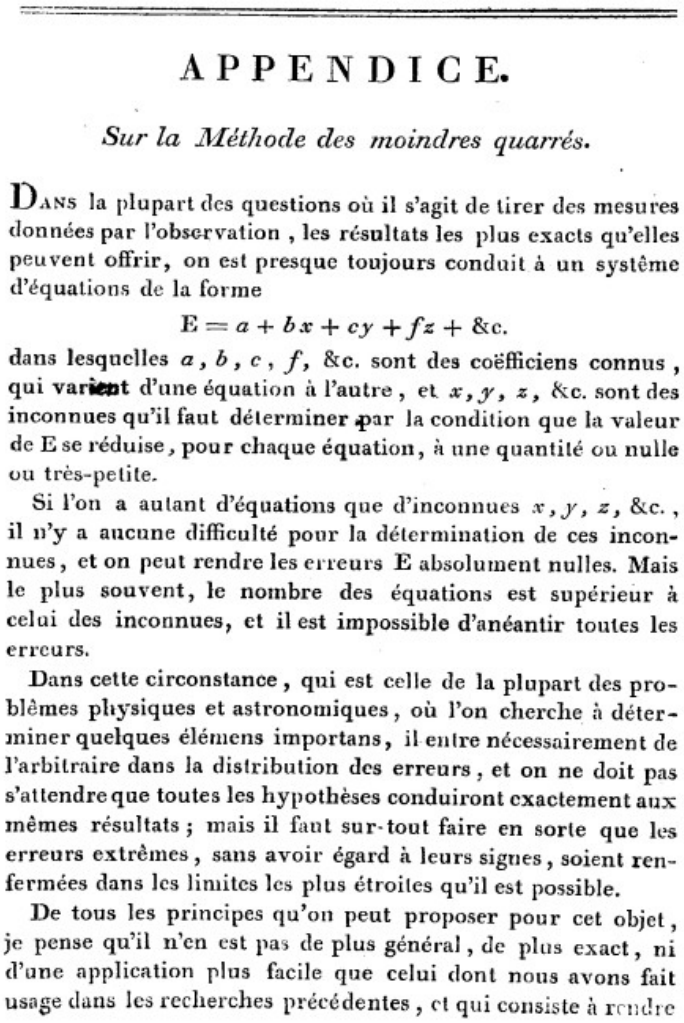
$$L1 = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 + \alpha \sum_{j=1}^{g} |w_j|$$

hyperparameter

L1 Regularization

Ridge – additional constraints

Origin of `fit(X_train, y_train)` : “least square method” extended



APPENDICE.

Sur la Méthode des moindres quarrés.

DANS la plupart des questions où il s'agit de tirer des mesures données par l'observation, les résultats les plus exacts qu'elles peuvent offrir, on est presque toujours conduit à un système d'équations de la forme

$$E = a + bx + cy + fz + \&c.$$

dans lesquelles $a, b, c, f,$ &c. sont des coëfficiens connus, qui varient d'une équation à l'autre, et $x, y, z,$ &c. sont des inconnues qu'il faut déterminer par la condition que la valeur de E se réduise, pour chaque équation, à une quantité ou nulle ou très-petite.

Si l'on a autant d'équations que d'inconnues $x, y, z,$ &c., il n'y a aucune difficulté pour la détermination de ces inconnues, et on peut rendre les erreurs E absolument nulles. Mais le plus souvent, le nombre des équations est supérieur à celui des inconnues, et il est impossible d'anéantir toutes les erreurs.

Dans cette circonstance, qui est celle de la plupart des problèmes physiques et astronomiques, où l'on cherche à déterminer quelques élémens importans, il entre nécessairement de l'arbitraire dans la distribution des erreurs, et on ne doit pas s'attendre que toutes les hypothèses conduiront exactement aux mêmes résultats; mais il faut sur-tout faire en sorte que les erreurs extrêmes, sans avoir égard à leurs signes, soient renfermées dans les limites les plus étroites qu'il est possible.

De tous les principes qu'on peut proposer pour cet objet, je pense qu'il n'en est pas de plus général, de plus exact, ni d'une application plus facile que celui dont nous avons fait usage dans les recherches précédentes, et qui consiste à rendre

Simple Ridge(α)

linear model

$$y = \alpha + \beta x$$

Several points with error ε_i

$$y_i = \alpha + \beta x_i + \varepsilon_i.$$

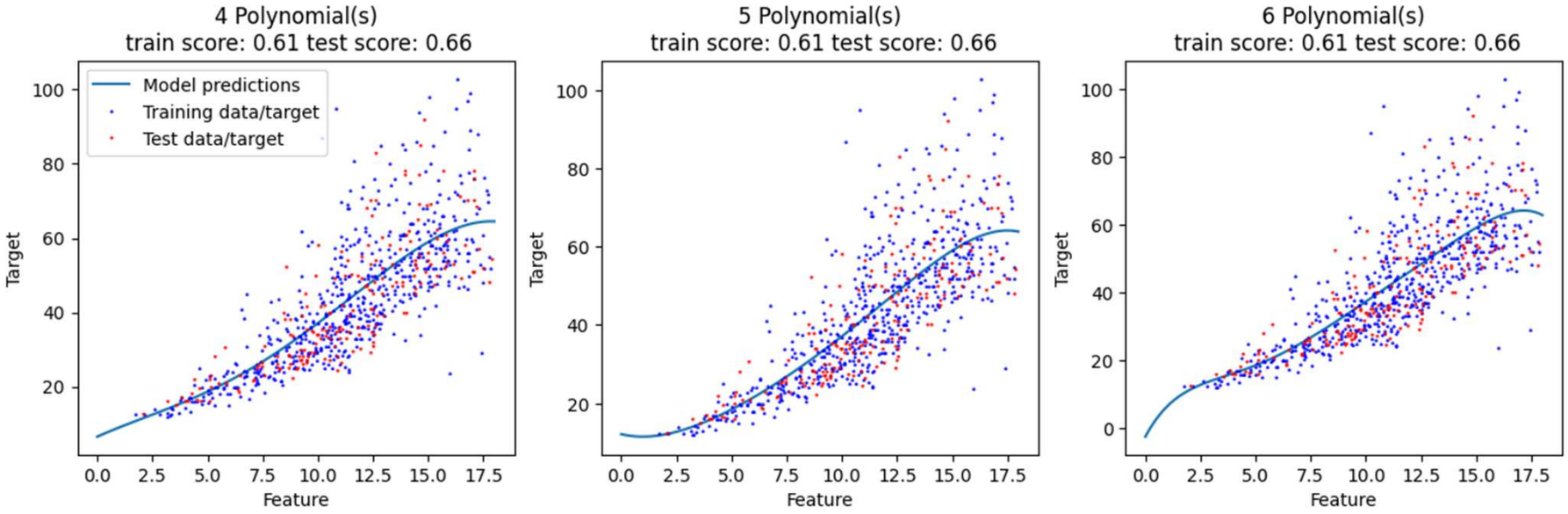
Error ε_i should be as small as possible

$$\hat{\varepsilon}_i = y_i - \alpha - \beta x_i.$$

Minimize the loss

$$L2 = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 + \alpha \sum_{j=1}^{g} w_j^2$$

Polynomial linear regression



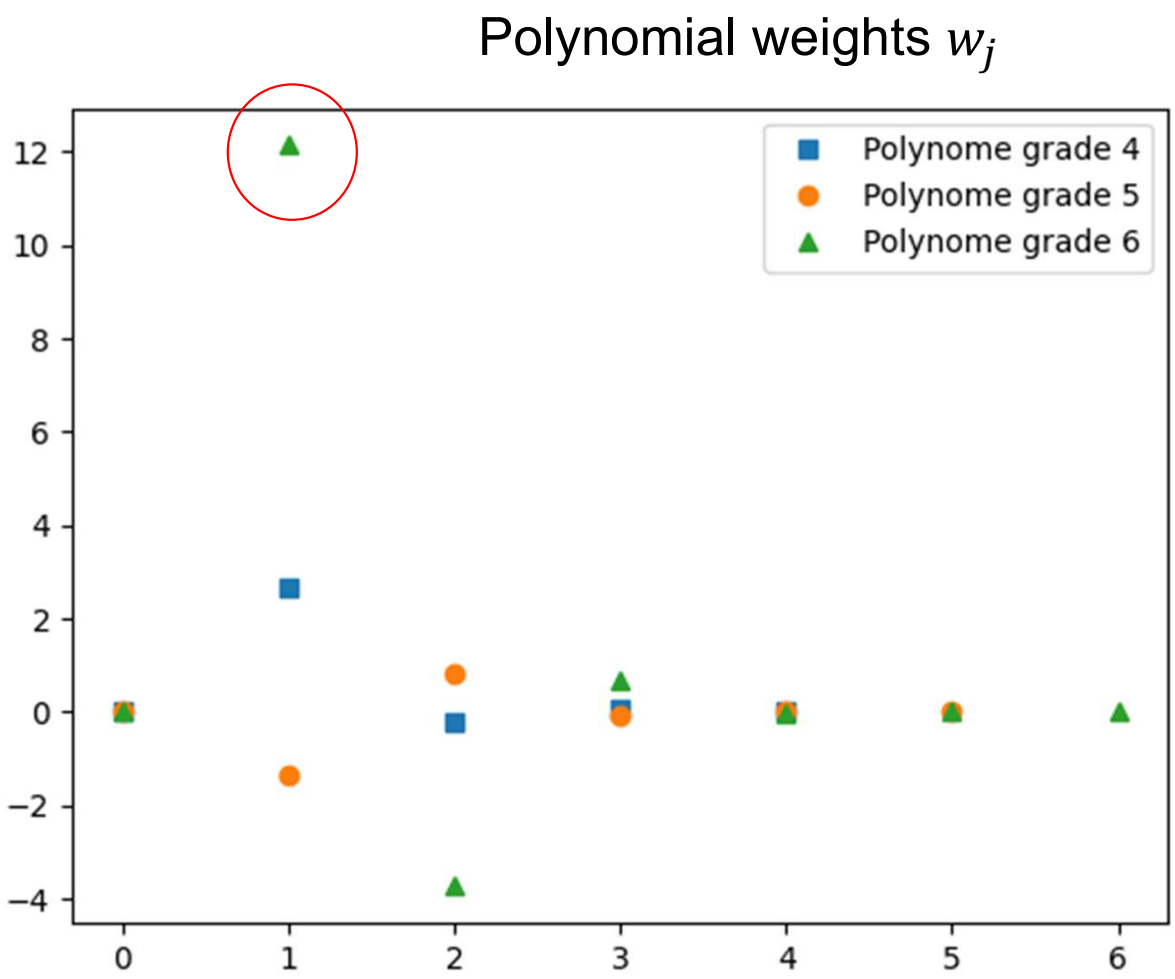
$\sum_{j=1}^{g} \lvert w_j \rvert$	2.91	2.23	16.6
$\sum_{j=1}^{g} w_j^2$	7.12	2.47	161.7

```
print("sum weights for grade 4: ", sum(abs(reg4.coef_)))
print("sum weights for grade 5: ", sum(abs(reg5.coef_)))
print("sum weights for grade 6: ", sum(abs(reg6.coef_)))

print("\n")

print("sum weights^2 for grade 4: ", sum(reg4.coef_**2))
print("sum weights^2 for grade 5: ", sum(reg5.coef_**2))
print("sum weights^2 for grade 6: ", sum(reg6.coef_**2))
```

Polynomial linear regression



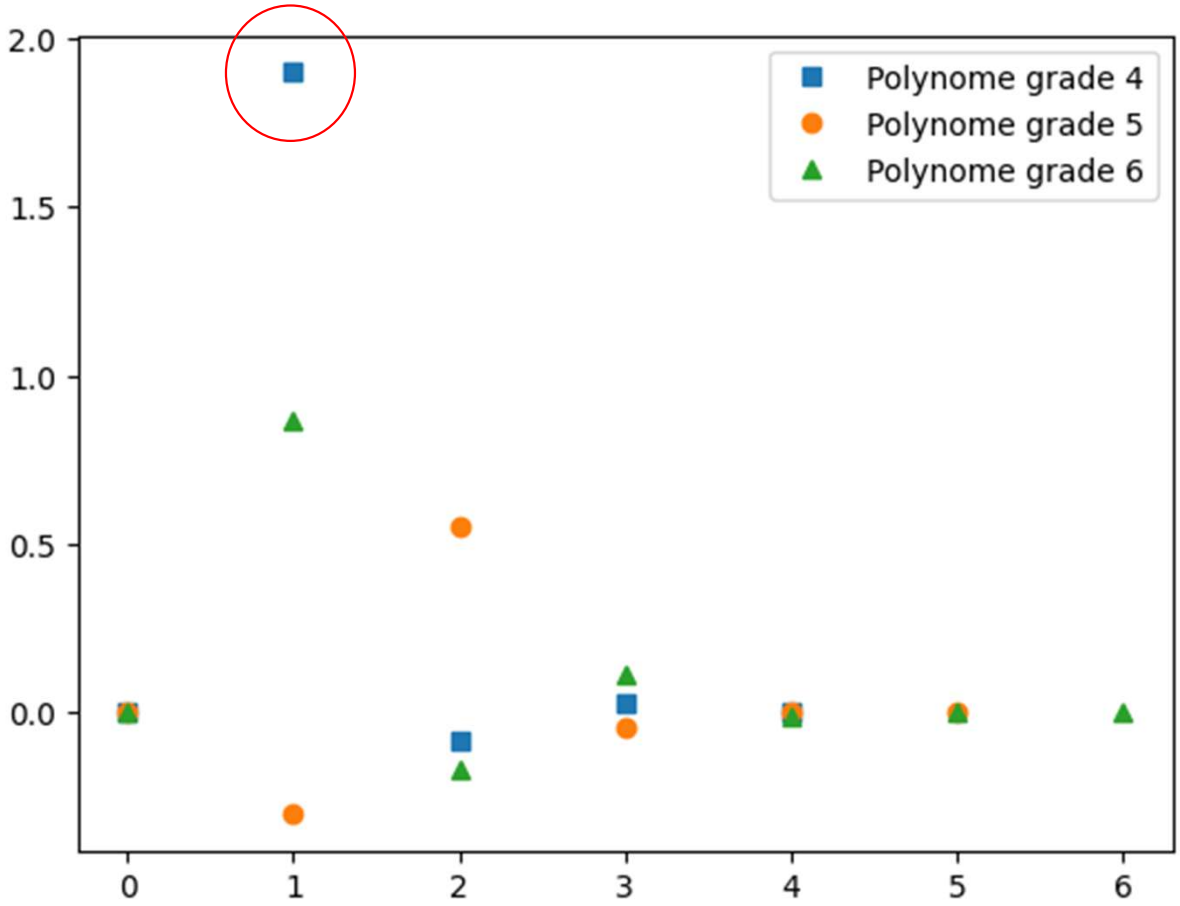
$\sum_{j=1}^{g}|w_j|$ $\sum_{j=1}^{g}w_j^2$

2.91 / 7.12

2.23 / 2.47 ←

16.6 / 161.7

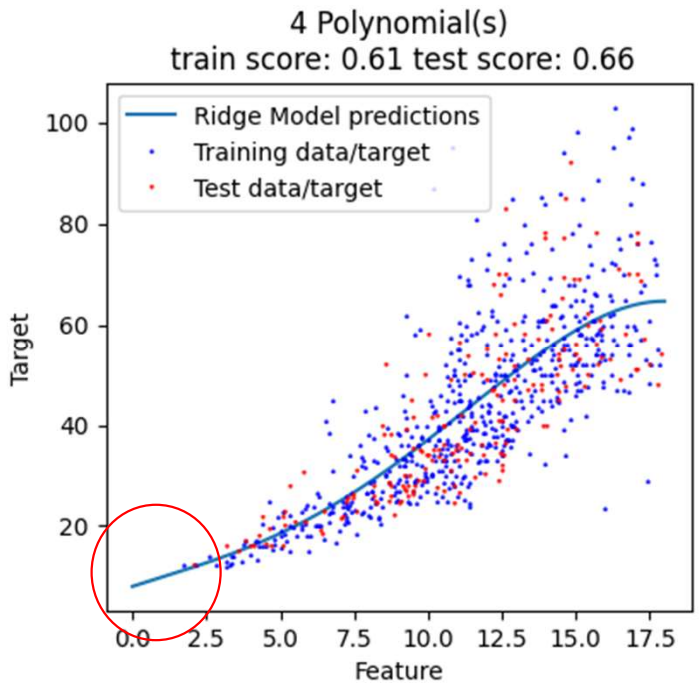
Reduce weights with Ridge

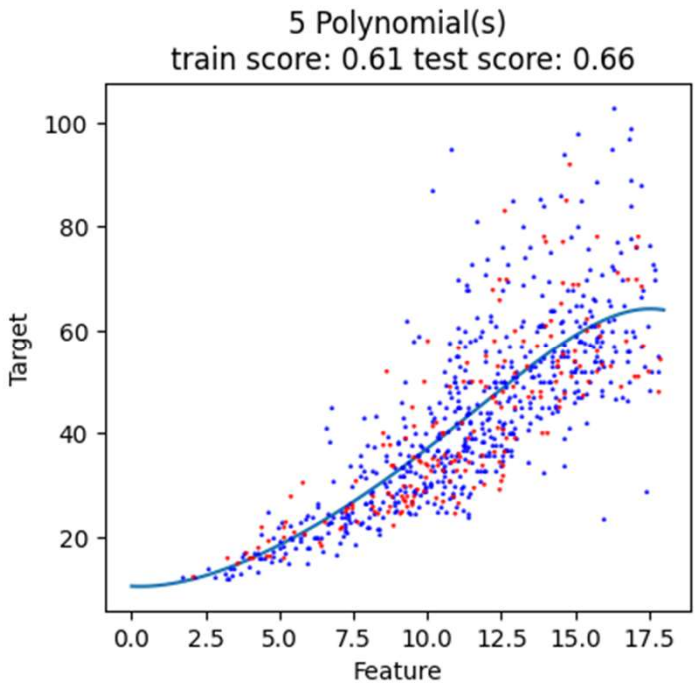


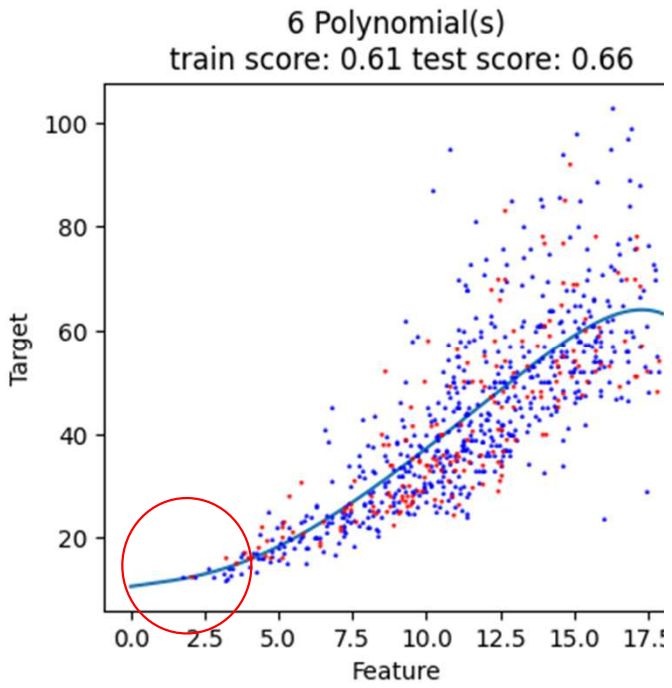
$\sum_{j=1}^{g}|w_j|$ $\sum_{j=1}^{g}w_j^2$

- / 3.61
- / 0.40
- / 0.79

Polynomial linear regression with Ridge







$\sum_{j=1}^{g} |w_j|$

$\sum_{j=1}^{g} w_j^2$

-	-	-
7.12 → 3.61	2.47 → 0.40	161.7 → 0.78

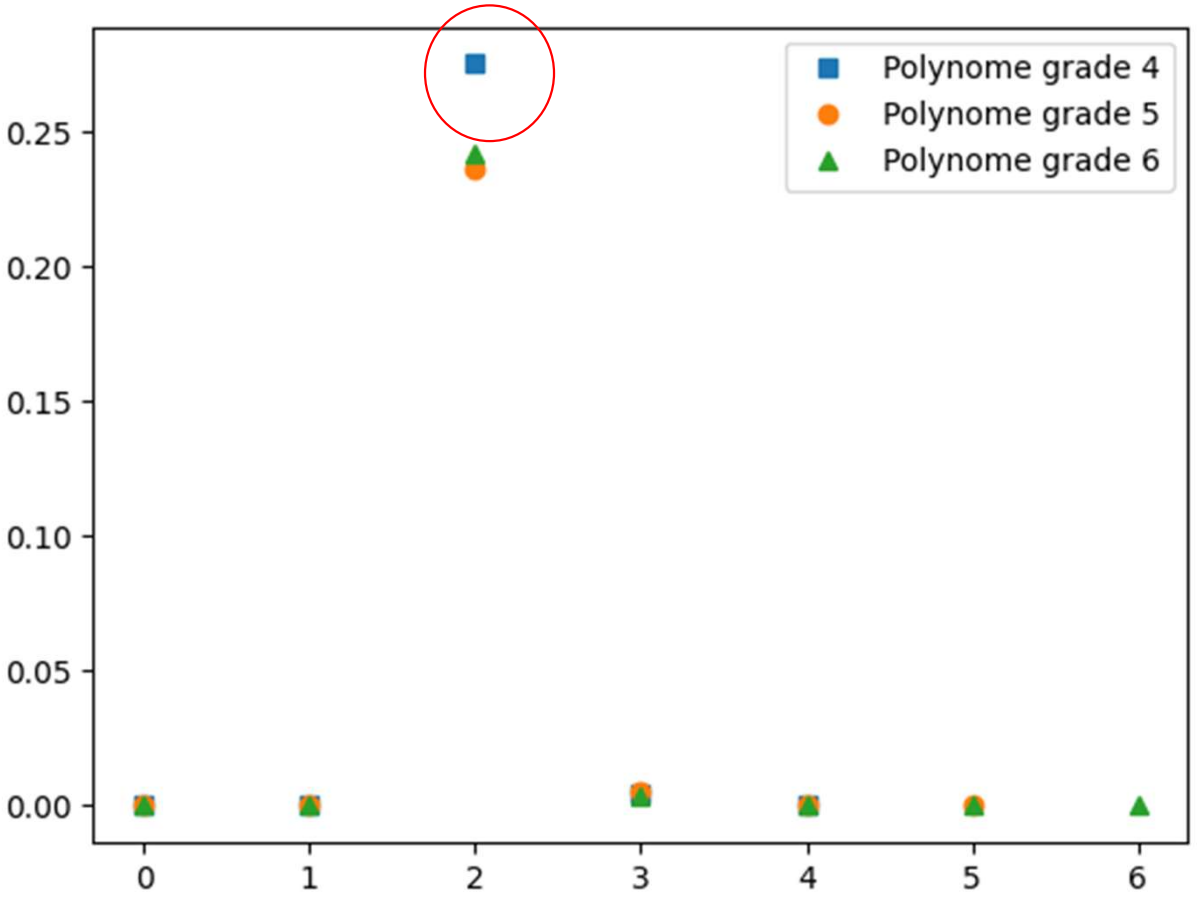
```
print("sum weights for grade 4: ", sum(abs(reg4.coef_)))
print("sum weights for grade 5: ", sum(abs(reg5.coef_)))
print("sum weights for grade 6: ", sum(abs(reg6.coef_)))

print("\n")

print("sum weights^2 for grade 4: ", sum(reg4.coef_**2))
print("sum weights^2 for grade 5: ", sum(reg5.coef_**2))
print("sum weights^2 for grade 6: ", sum(reg6.coef_**2))
```

Reduce weights with Lasso

Polynomial weights w_j



$\sum_{j=1}^{g}|w_j|$ $\sum_{j=1}^{g}w_j^2$

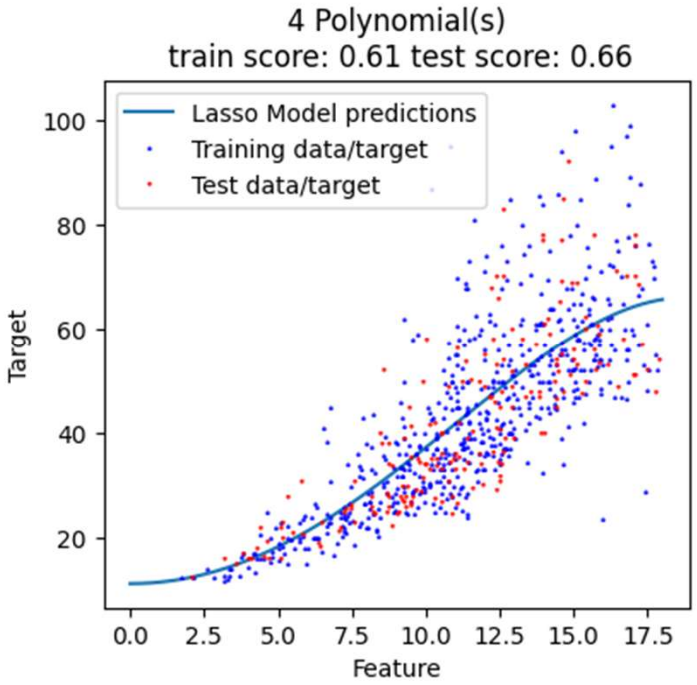
0.28 / -

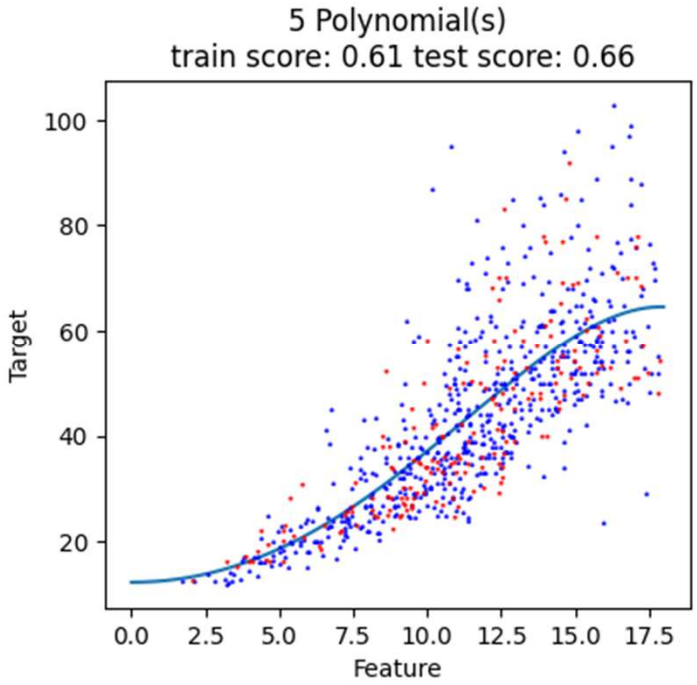
0.24 / -

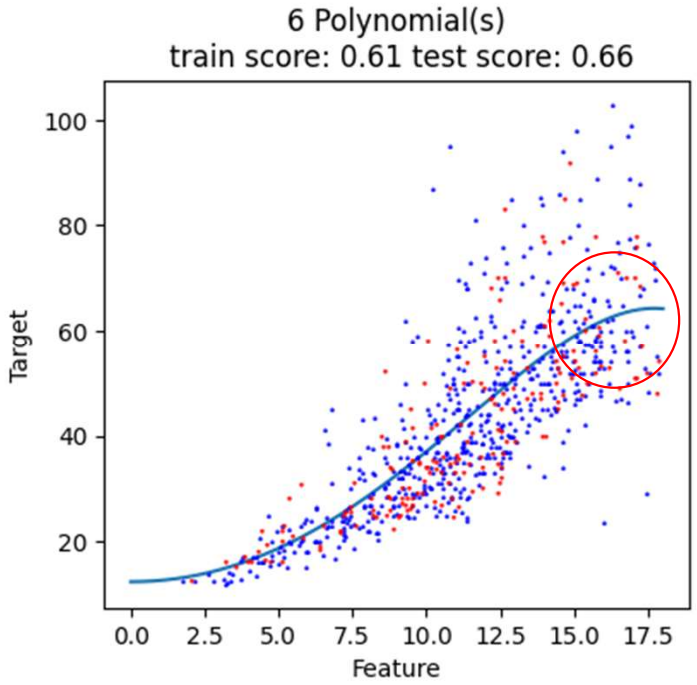
0.25 / -

If there is a group of highly correlated variables,
Lasso selects one variable from a group and ignore the others

Polynomial linear regression with Lasso







$\sum_{j=1}^{g} |w_j|$

2.91 → 0.28

2.23 → 0.24

16.6 → 0.25

$\sum_{j=1}^{g} w_j^2$

-

-

-

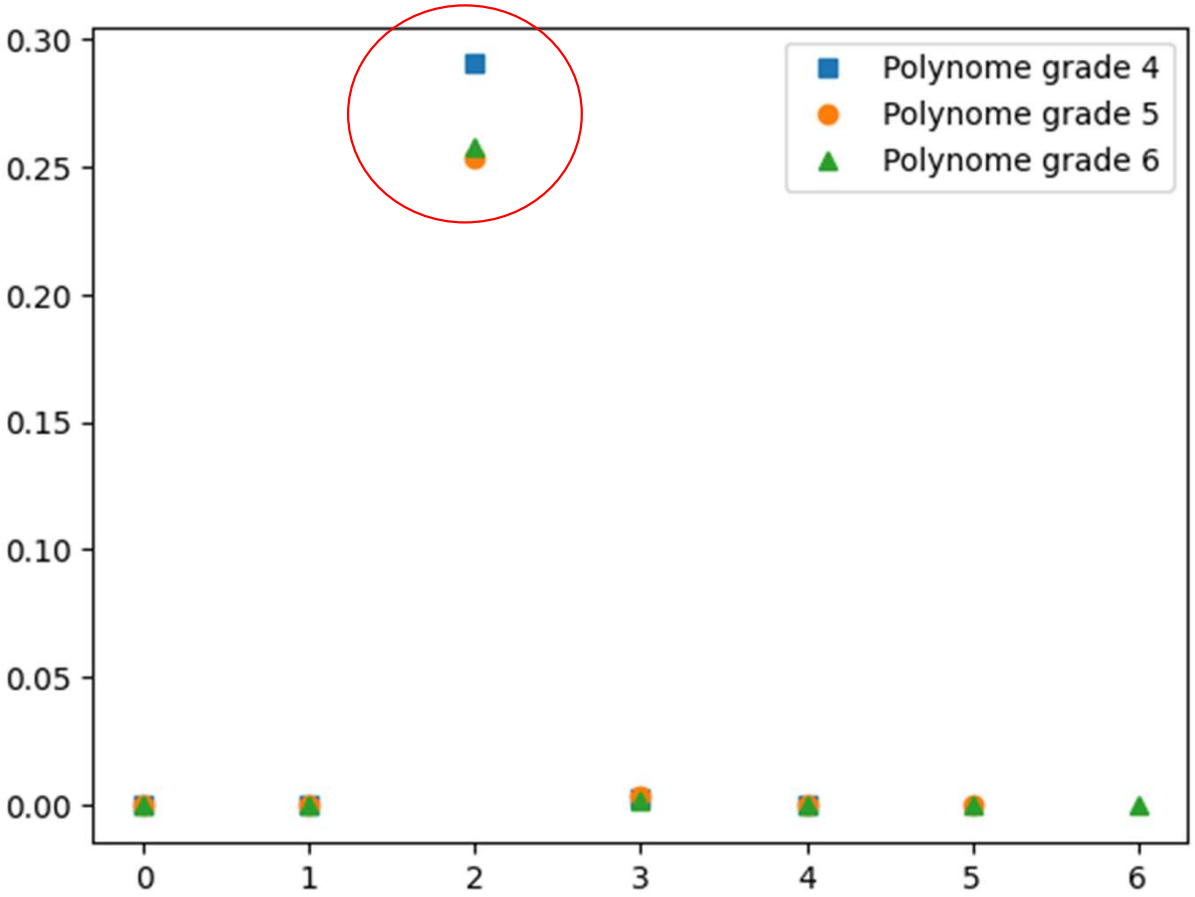
```
print("sum weights for grade 4: ", sum(abs(reg4.coef_)))
print("sum weights for grade 5: ", sum(abs(reg5.coef_)))
print("sum weights for grade 6: ", sum(abs(reg6.coef_)))

print("\n")

print("sum weights^2 for grade 4: ", sum(reg4.coef_**2))
print("sum weights^2 for grade 5: ", sum(reg5.coef_**2))
print("sum weights^2 for grade 6: ", sum(reg6.coef_**2))
```

Reduce weights with ElasticNet

Polynomial weights w_j



$\sum_{j=1}^{g}|w_j|$ $\sum_{j=1}^{g} w_j^2$

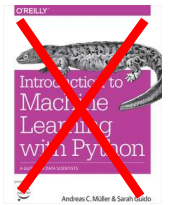
0.29 / 0.08

0.26 / 0.06

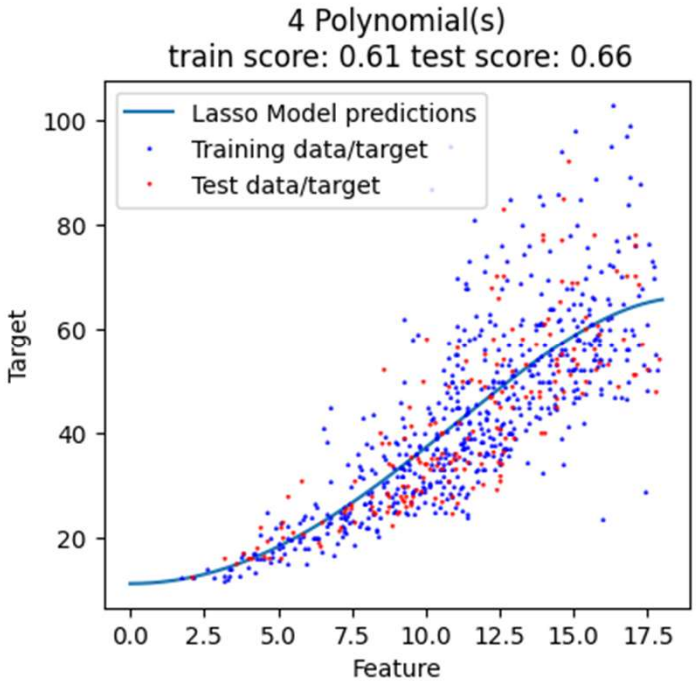
0.26 / 0.07

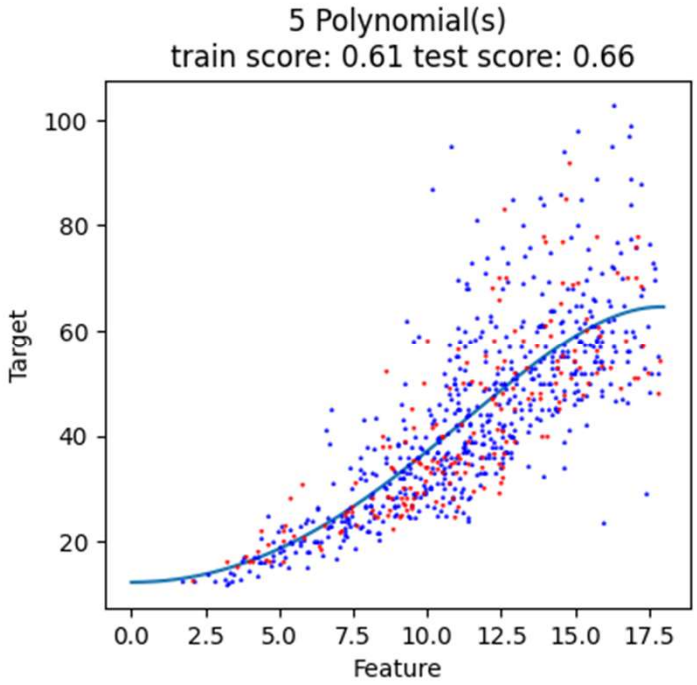
ElasticNet optimizes the Loss-function with includes both L1 & L2

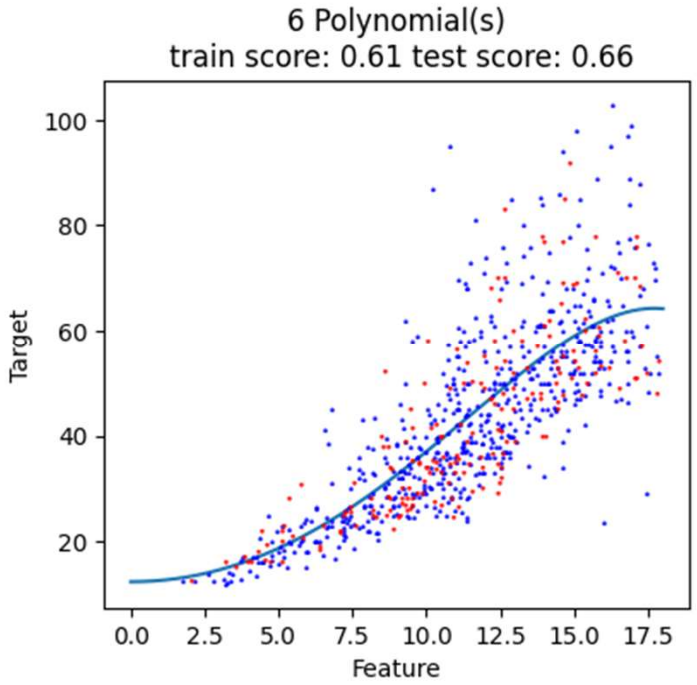
L1+L2 Regularization



Polynomial linear regression with ElasticNet







$\sum_{j=1}^{g} |w_j|$

2.91 → 0.29
7.12 → 0.08

2.23 → 0.26
2.47 → 0.06

16.6 → 0.26
161.7 → 0.07

$\sum_{j=1}^{g} w_j^2$

```
print("sum weights for grade 4: ", sum(abs(reg4.coef_)))
print("sum weights for grade 5: ", sum(abs(reg5.coef_)))
print("sum weights for grade 6: ", sum(abs(reg6.coef_)))

print("\n")

print("sum weights^2 for grade 4: ", sum(reg4.coef_**2))
print("sum weights^2 for grade 5: ", sum(reg5.coef_**2))
print("sum weights^2 for grade 6: ", sum(reg6.coef_**2))
```

Three regulization models

- Which one to choose?

- Lasso / L1 Regularization
- Ridge / L2 Regularization
- Elastic Net / L1 + L2 Regularization

Lasso / L1 Regularization

- Benefits:
  - Creates sparse models - unnecessary features don't contribute to their prediction
  - Can speed up inference (in NNs)

- Disadvantages:
  - Doesn’t work well in a high-dimensional cases
  - Unnecessary when you already did Feature Selection
  - When the prediction is “smeared out” over several correlated features, you can loose essential information.

Ridge / L2 Regularization

- Benefits:
  - Reduces the weights, but doesn't force them to zero
  - Works well for high-dimensional data with many correlated features

- Disadvantages:
  - The result is hard to interpret, with all (correlated) features contributing a tiny bit to the prediction.

Elastic Net / L1 + L2 Regularization

- Benefits:
  - Very robust
  - Works well if the result is close to Lasso or Ridge

- Disadvantages:
  - Can introduce unnecessary extra bias
  - Additional hyperparameter

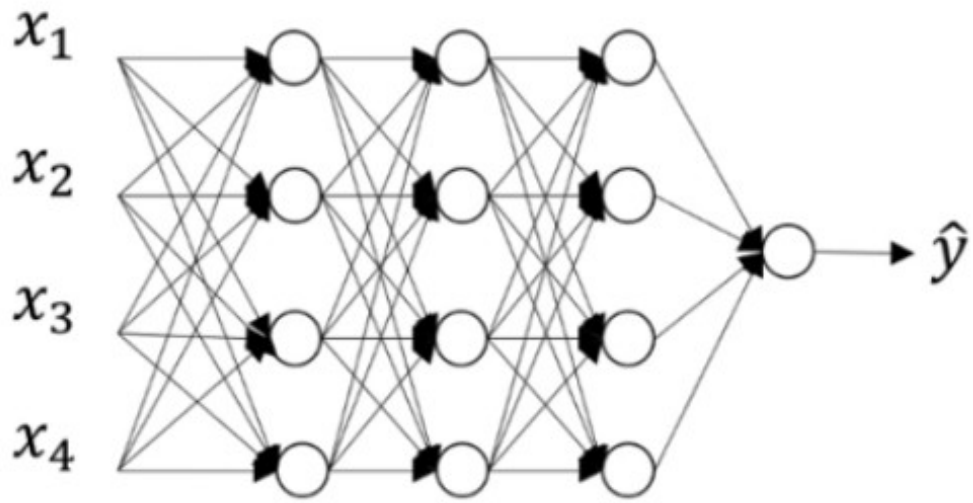
Best regularization choice

- If you prepared your data well (normalization & feature selection)
  - No regularization is needed
- If your dataset is already very sparse
  - Choose Ridge / L2 regularization
- If your dataset is large with many correlations
  - Choose Ridge / L2 regularization
- If your dataset is dense with many uncorrelated features
  - Choose Lasso / L1 regularization
- If you don't know
  - Choose ElasticNet / L1 + L2 regularization

Christiaan Versloot, Should I start with L1, L2 or Elastic Net Regularization?

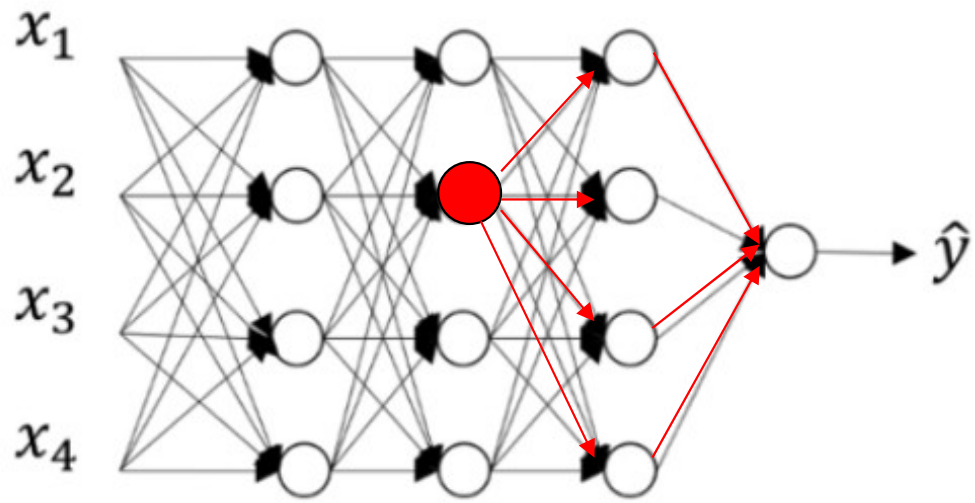
regularization in NNs

- In NNs overfitting is a real problem



regularization in NNs

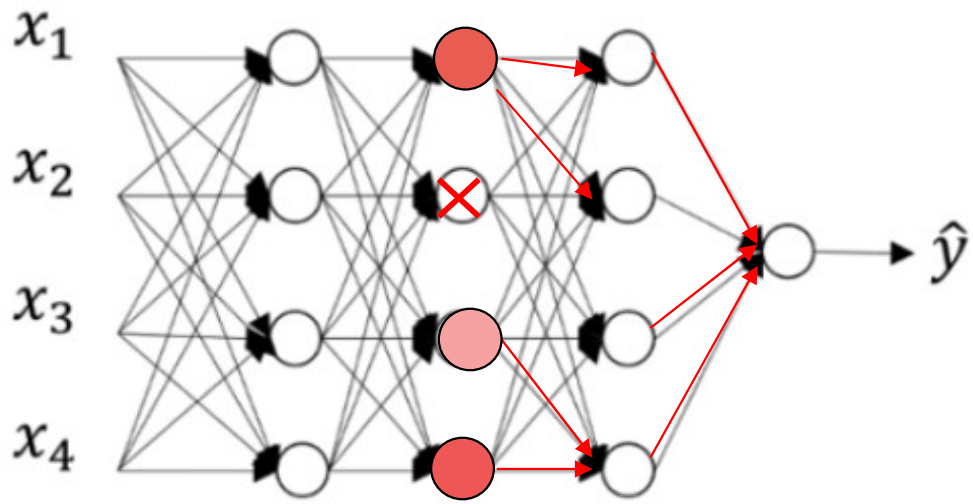
- In NNs overfitting is a real problem



- Dominant node

regularization in NNs

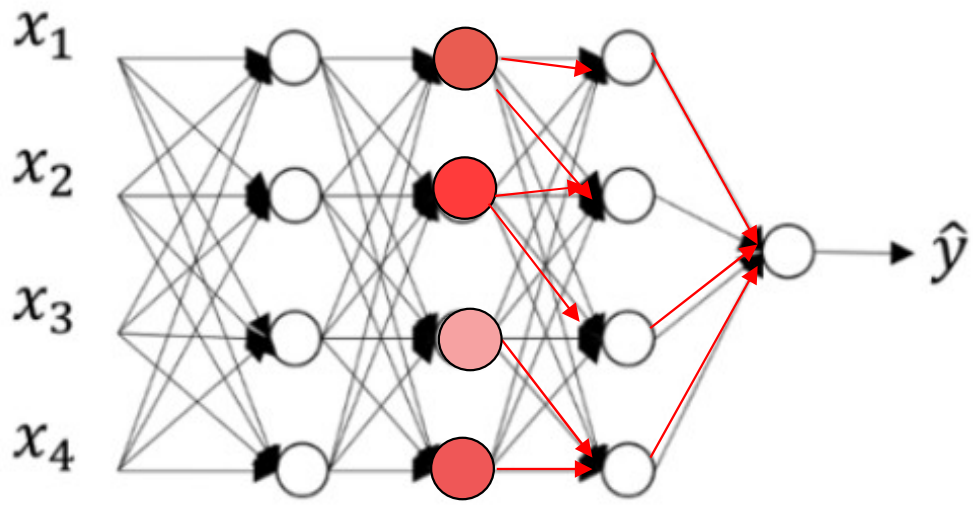
- In NNs overfitting is a real problem



- Dropout of dominant node 

regularization in NNs

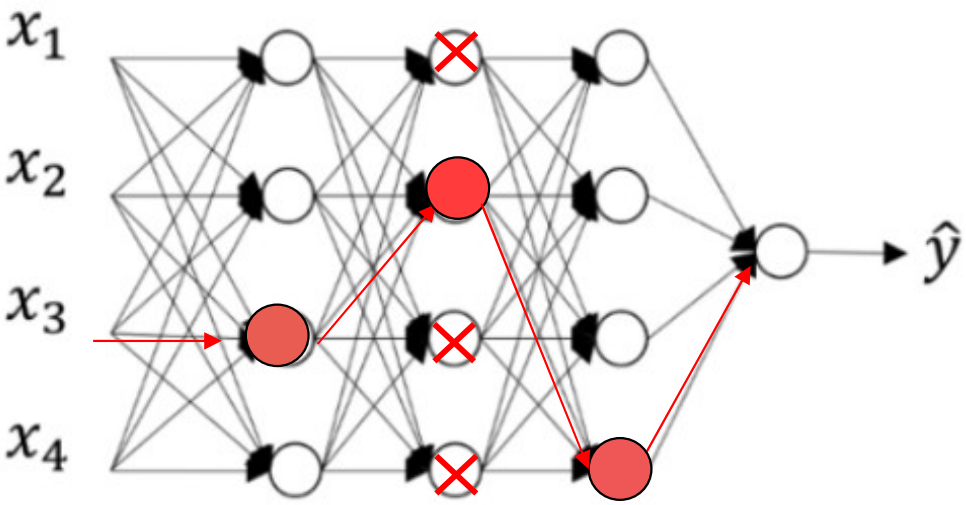
- In NNs overfitting is a real problem



- L2 regularization of the dominant node ("smeared out")

regularization in NNs

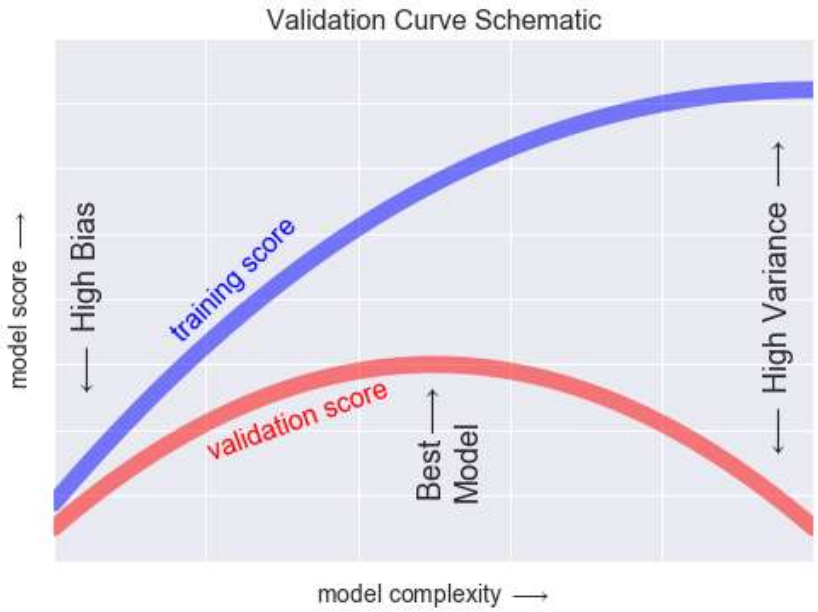
- In NNs overfitting is a real problem



- L1 regularization can simplify the network

What is good model?

- How could you improve the model?

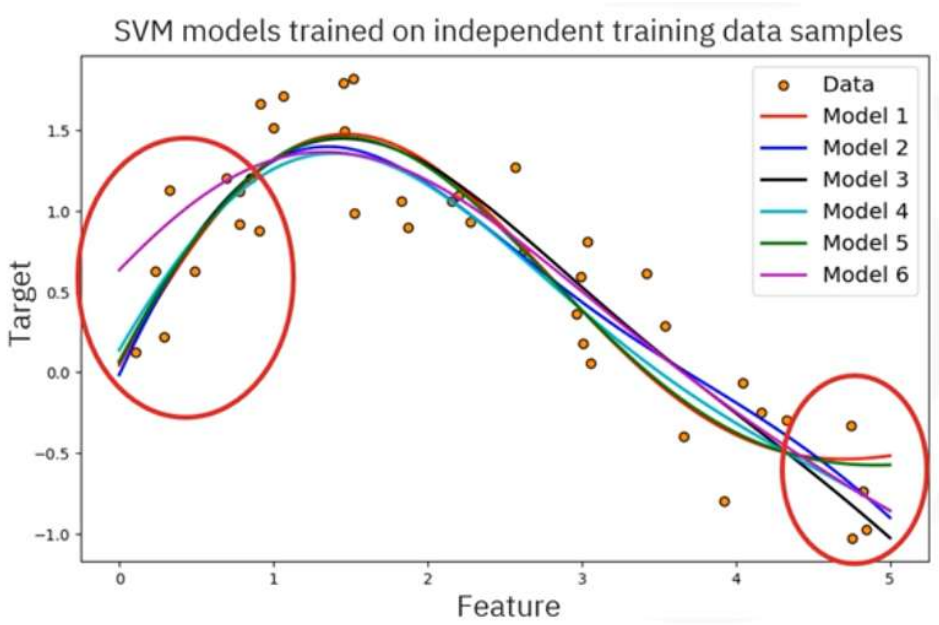


- Use a more complicated / more flexible model
- Use a less complicated / less flexible model

Page 363

What is good model?

- Variance in the model predictions

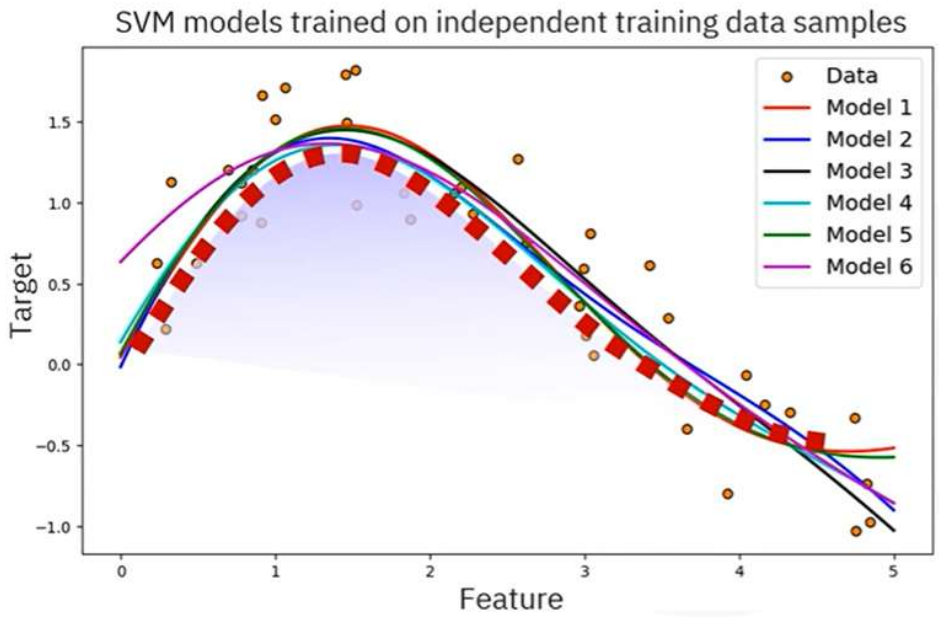


- If the curves align the remaining variance in data
- If there are differences at between the curves, model variance still exist

Courtesy Joseph Santarcangelo & Jeff Grossman

What is good model?

- Aggregate multiple models to reduce model variance

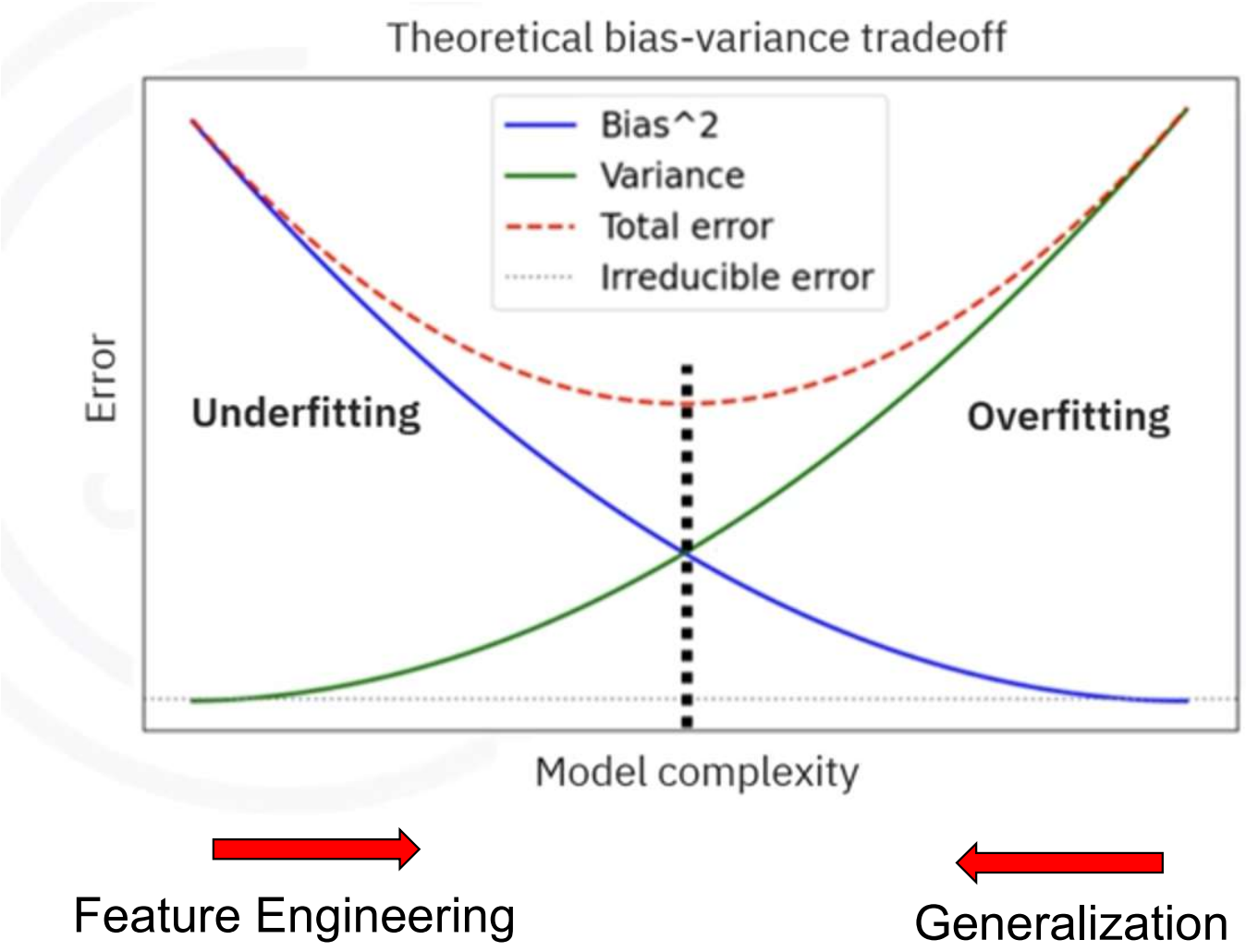


- Bootstrap aggregating or bagging
- Mitigate overfitting on data

Courtesy Joseph Santarcangelo & Jeff Grossman

What is good model?

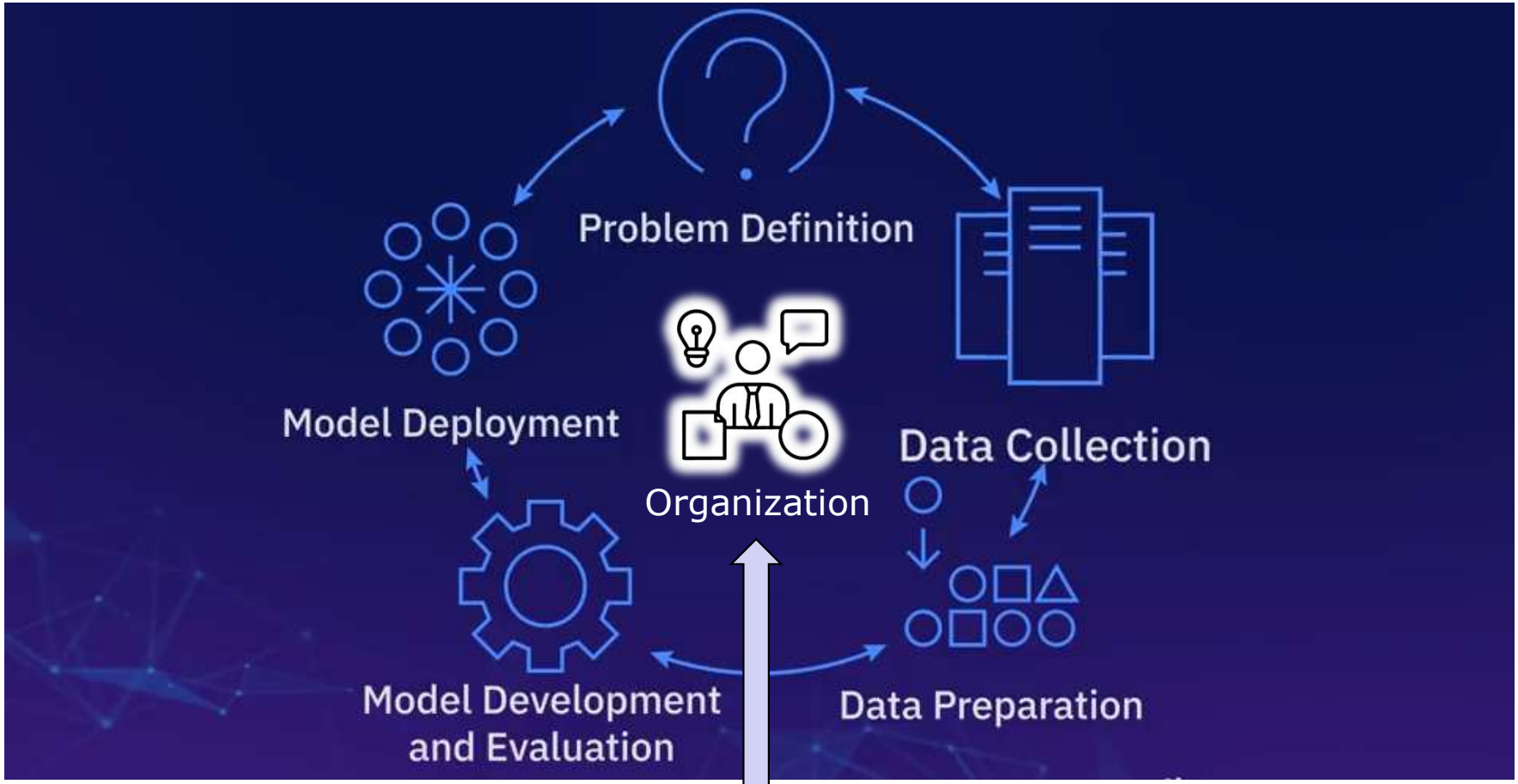
- Balance



Courtesy Joseph Santarcangelo & Jeff Grossman

Introduction to Machine Learning

Building a predictive model is a circular process.

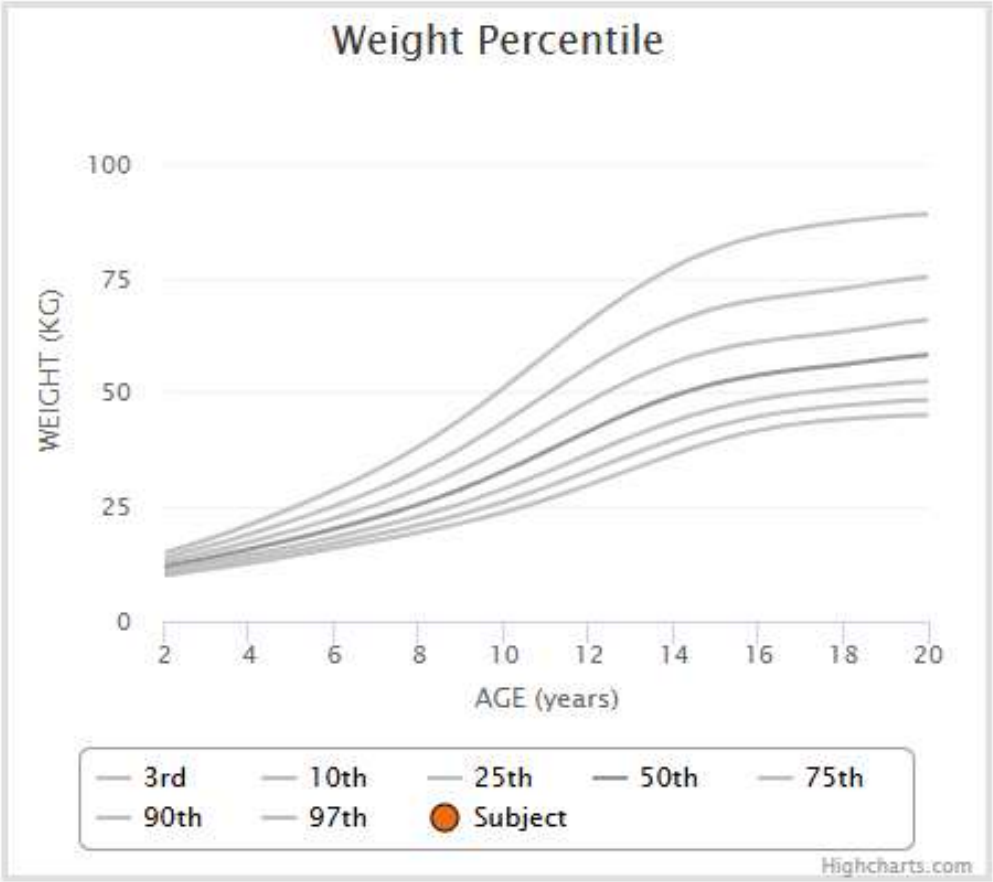


Expertise

- Depends on the organization
- Interpret the information in the right way

Expert knowledge

Healthy weight is between 5th and 85th percentile



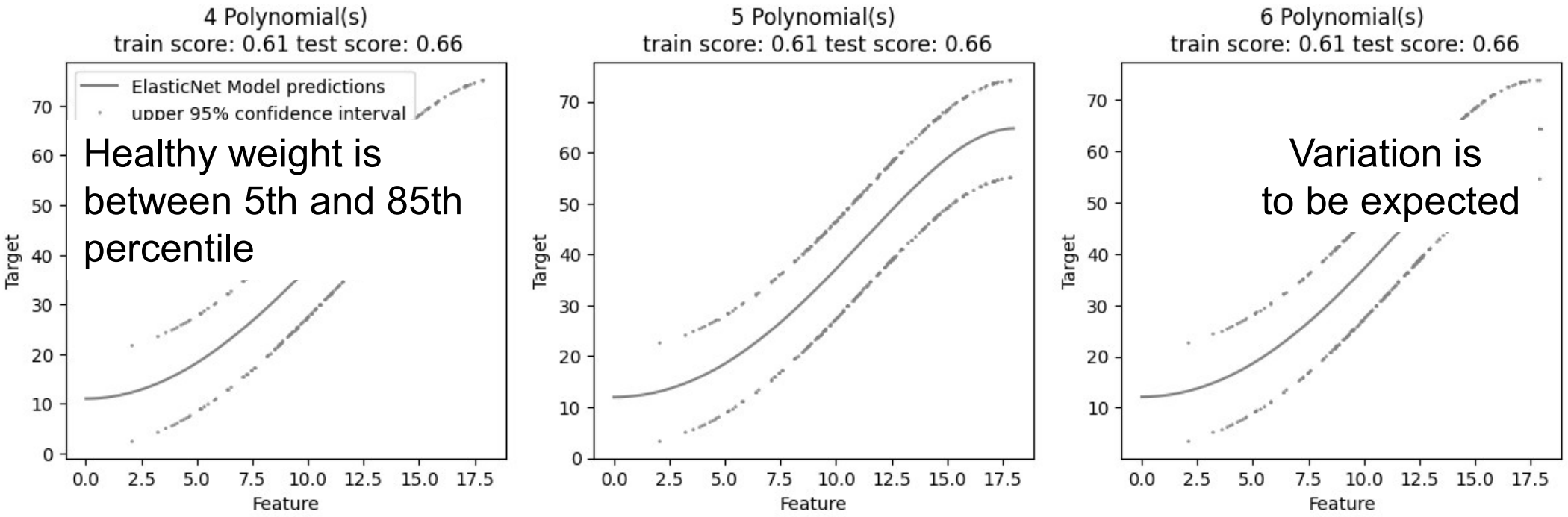
Variation is to be expected

[Age-based Pediatric Growth Reference Charts]

American Academy of Pediatrics

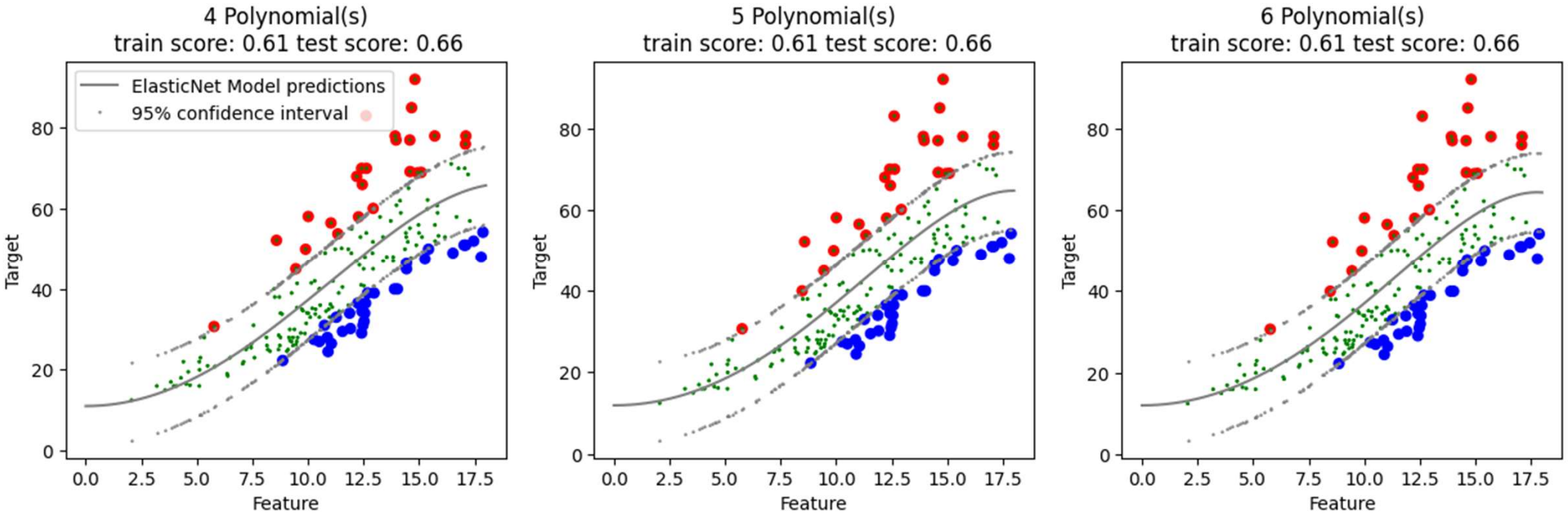
This calculator can help to determine whether a child is at a healthy weight for his/her height, age and gender.
The amounts of body fat, muscle, and bone change with age, and differ between boys and girls.
This calculator automatically adjusts for differences in height, age and gender.

Expert knowledge



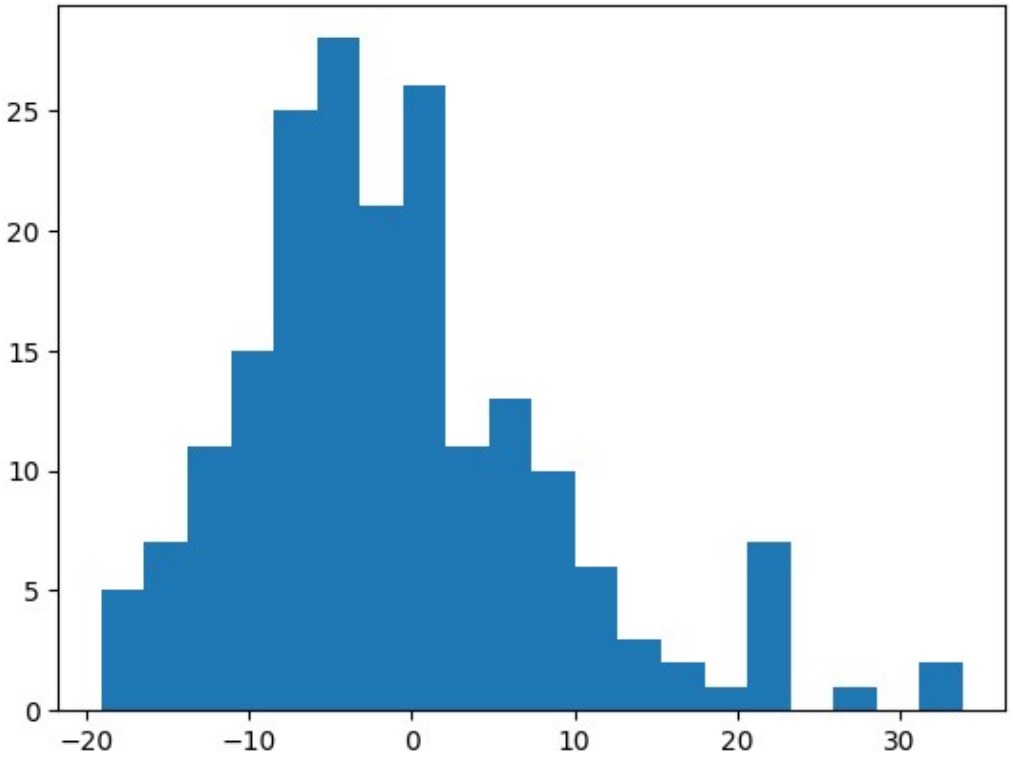
Be aware of the confidence interval of your model

Expert knowledge



12% above the 95% confidence interval
16% below the 95% confidence interval

Expert knowledge

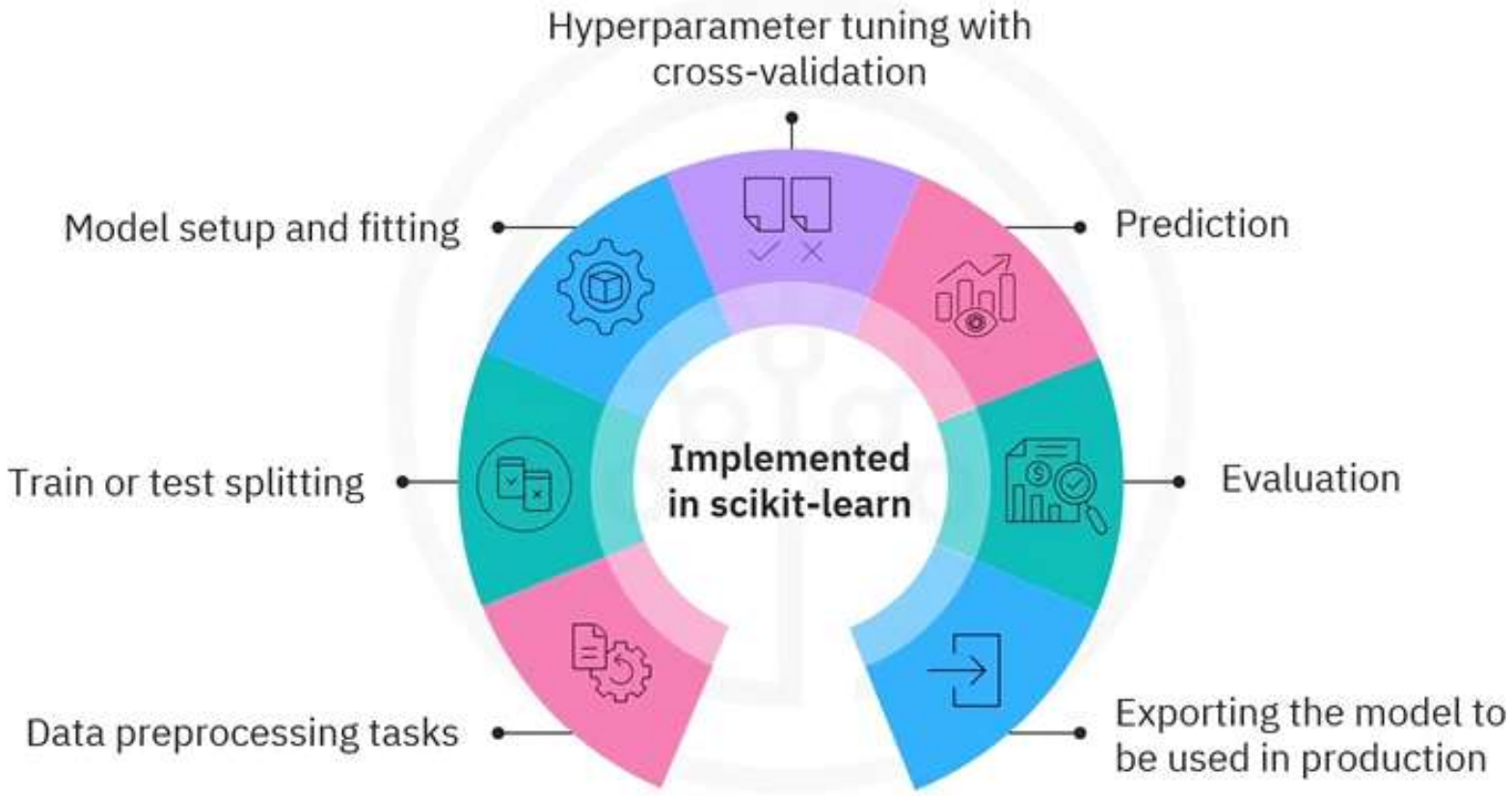


Histogram of residuals

Nearly Normal distribution
Skewed to heavier weights
mean shifted from median

Introduction to Machine Learning

Building a predictive model is a circular process.

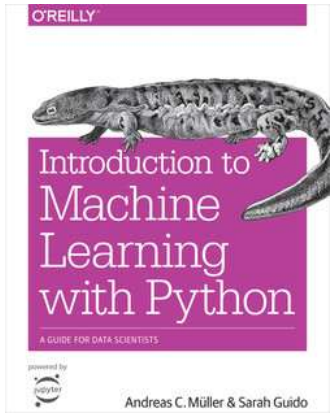


Joseph Santarcangelo & Joss Grossman, Machine Learning with Python, IBM Skills network, 2025

Introduction to Machine Learning

- 2.2 Generalization, overfitting & underfitting
- 2.3.3 Regularization

- Strengths:
  - Complexity of model can be controlled
- Weakness:
  - Additional hyperparameter.



Andreas C. Müller, Sarah Guido, Introduction to Machine Learning with Python, O'Reilly Media, October 2016

Conclusion

Learning outcomes of this course covered today

- Making the model less complex improves the generalization