Machine Learning for econometrics

Statistical learning and penalized regression

Matthieu Doutreligne

January 10, 2025

A lot of today's content is taken from the excellent sklearn mooc (Estève et al., 2022)

Today's program

- Last session: importance of causal variable status
- Today: predictive inference in high dimensions
  - Statistical learning basics
  - Penalized linear regression for predictive inference
  - Hands-on with scikit-learn
-

Today's program

- Last session: importance of causal variable status
- Today: predictive inference in high dimensions
  - Statistical learning basics
  - Penalized linear regression for predictive inference
  - Hands-on with scikit-learn
- Next session:
  - Flexible models: Trees, Random Forests, Gradient Boosting
  - Practical scikit-learn

Table of contents

Statistical learning framework

Statistical learning, ie. predictive inference

Goal

- Predict the value of an outcome based on one or more input variables.

Setting

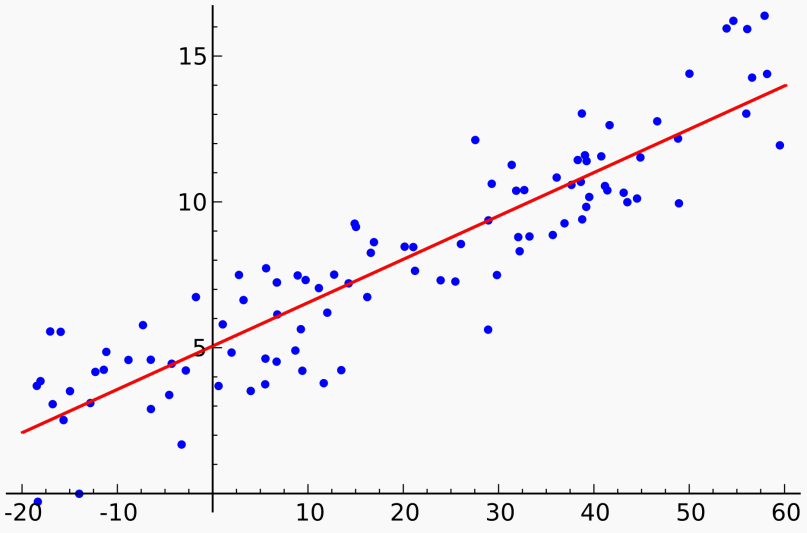
- Data: n pairs of (features, outcome), $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ identically and independently distributed (i.i.d.) from an unknown distribution P.
- Goal: find a function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that approximates the true value of y ie. for a new pair (x, y), we should have:

$$\hat{y} = \hat{f}(x) \approx y$$

Vocabulary

Finding the appropriate model $\hat{f}$ is called learning, training or fitting the model.

Statistical learning, two classes of problems

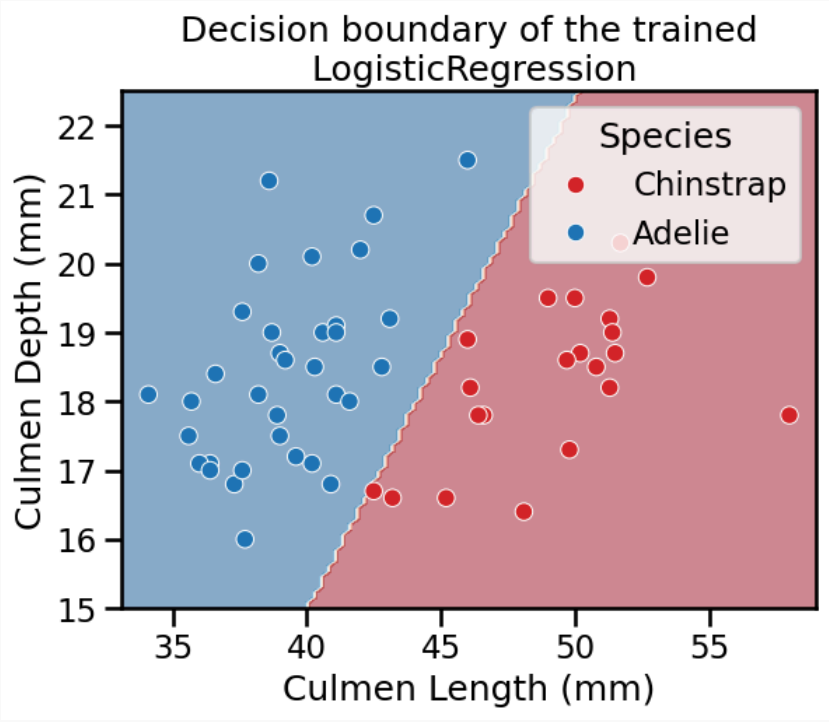


Regression

- The outcome is continuous: eg. wage prediction
- The error is often measured by the mean squared error (MSE):

$$\mathrm{MSE} = \mathbb{E}\left[\left(Y - \hat{f}(X)\right)^2\right]$$

Statistical learning, two classes of problems



Classification

- Outcome is categorical: eg. diagnosis, loan default, ...
- Error is often measured with accuracy:

$$\text{Misclassification rate} = \mathbb{E}\left[\mathbb{1}\left(Y \neq \hat{f}(X)\right)\right]$$

with $\hat{f} \in \{0, 1\}$ for binary classification

Motivation: why prediction?

Why do we need prediction for ?

Statistical inference

- Goal: infer some intervention effect with a causal interpretation
- Require to regress "away" the relationship between the treatment or the outcome and the confounders -> more on this in sessions on Double machine learning.

Predictive inference

- Some problems in economics requires accurate prediction without a causal interpretation (Kleinberg et al., 2015)
- Eg. Stratisfying on a risk score (loan, preventive care, ...)

Do we need more than linear models?

Let:

- p is the number of features
- n is the number of observations

Maybe no

- Low-dimensional data: $n \gg p$
- No non-linearities, no or few interactions between features

Maybe yes

- High-dimensional data: ie. $p \gg n$
- Non-linearities, many interactions between features

Do we need more than linear models?

When do we have "high-dimension"?

- Is $p \gg n$ a common setting in economics?
- Consider the wage dataset:
  - $n = 5150$ individuals
  - $d = 18$ variables
  -

When do we have "high-dimension"?

- Is $p \gg n$ a common setting in economics?
- Consider the wage dataset:
  - $n = 5150$ individuals
  - $d = 18$ variables
  - But, categorical variables, non-linearities and interactions increase the real number of features:
    - non-linearities: add polynomials of degree 2: $p = 2 \times 18 = 36$
    - interactions:
      - Of degree 2: $\binom{d}{2} = \binom{18}{2} = 153$
      - All interactions: $2^d = 2^{18} - 18 - 1 = 262125$

Is this common?

Yes

- Categorical or text data are increasingly common
- Image data is high-dimensional by nature
- Automation of data collection and storage leads to more collections of variables

Some examples:

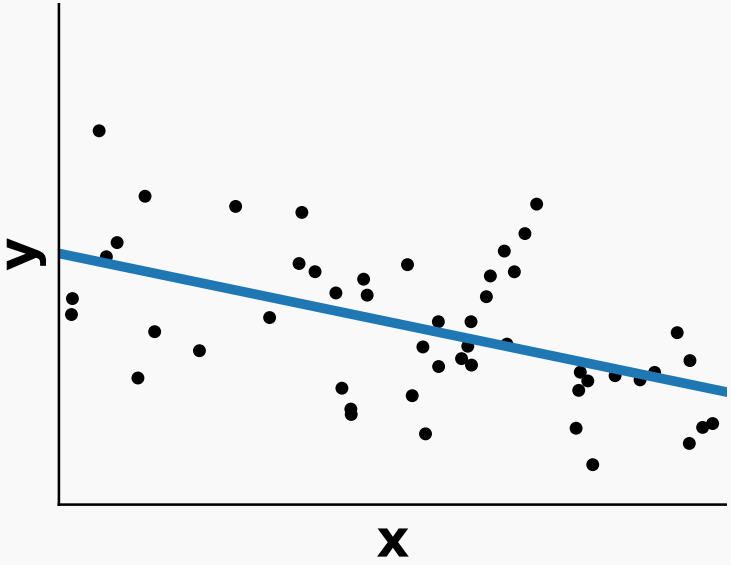
- The Current Population Survey (CPS) dataset has hundreds of variables, many of which are categorical
- The Système National des Données de Santé (SNDS) in France collects all reimbursments : many hundreds of variables, many of which are categorical

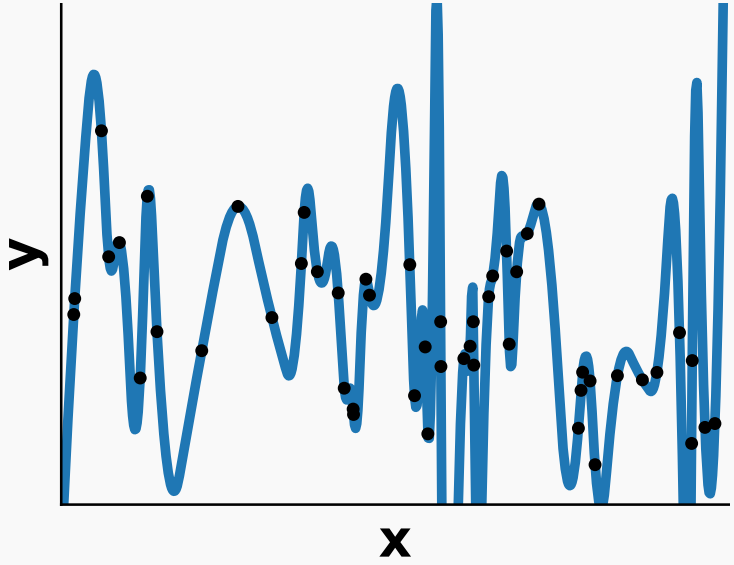
Is this common?

- The population referencement dataset from INSEE lists 800 pairs of (variables, categories).

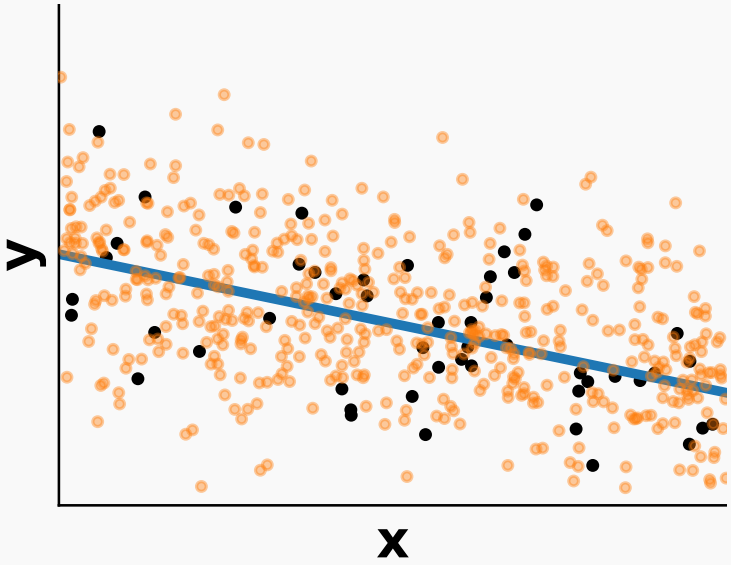
Statistical learning theory and intuitions

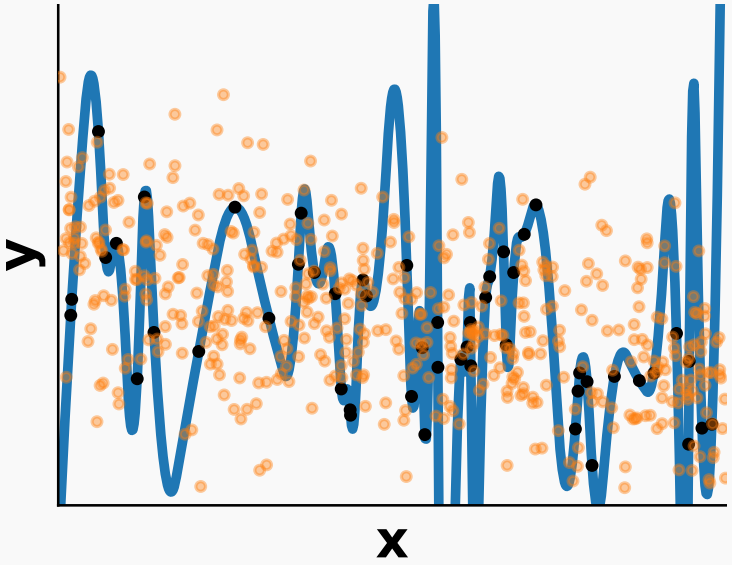
Which data fit do you prefer?





Which data fit do you prefer? (new data incoming)

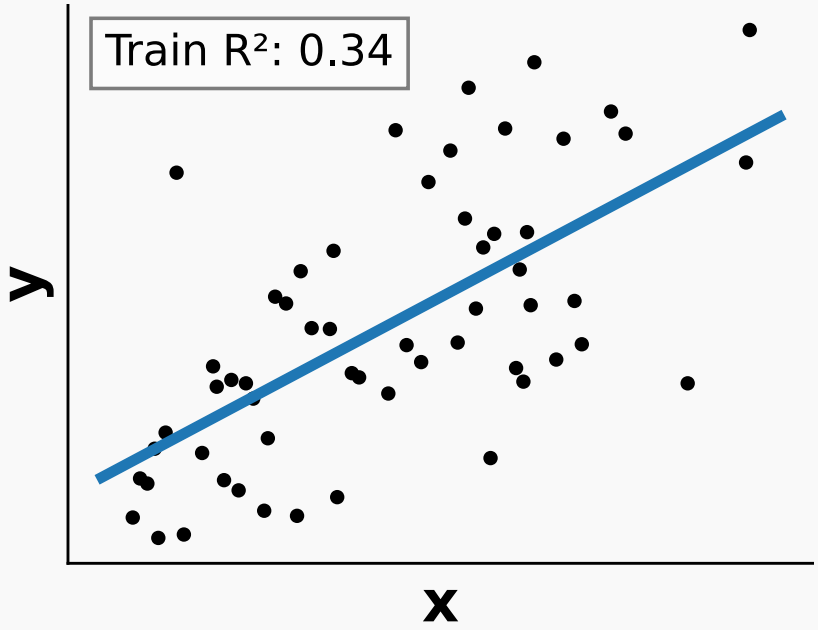




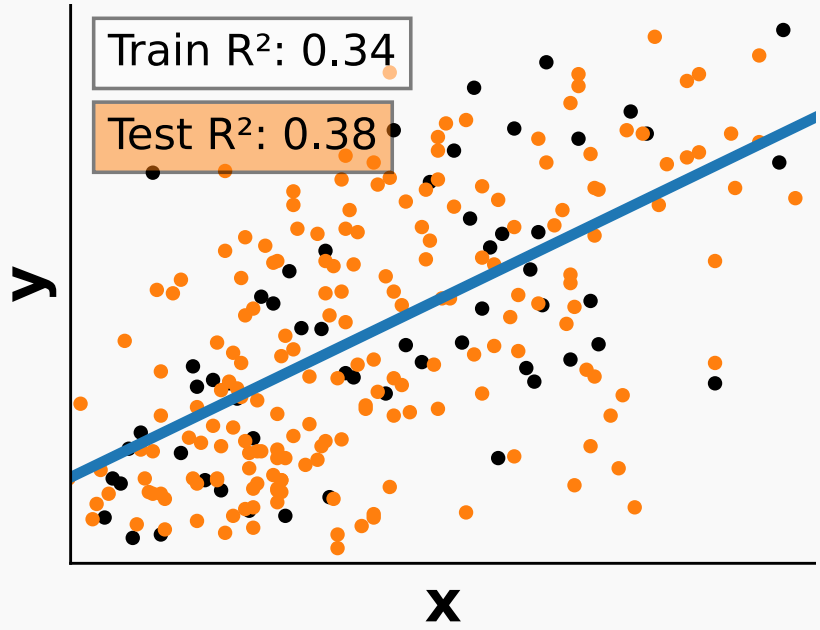
- Answering this question might be hard.
- Goal: create models that generalize.
- The good way of framing the question is: how will the model perform on new data?

Train vs test error: simple models

Measure the errors on the training data = fitting



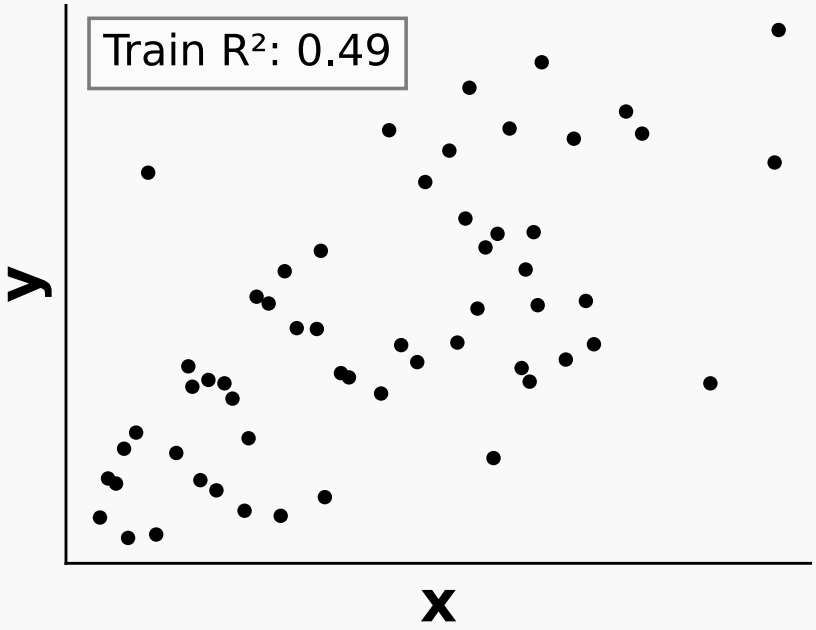
Measure the performances on test data = generalization



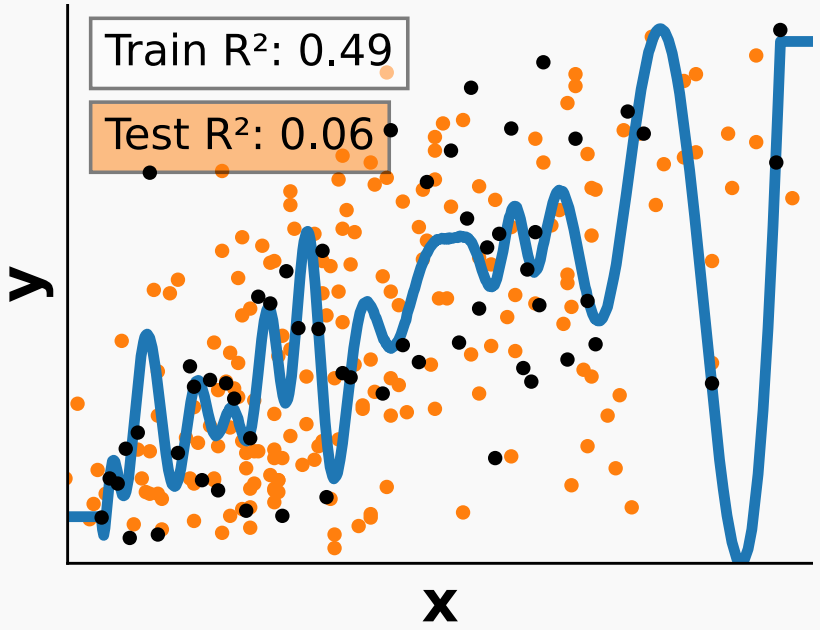
🎉 Here, no problem of overfitting: train vs test error are similar.

Train vs test error: flexible models

Measure the errors on the training data = fitting

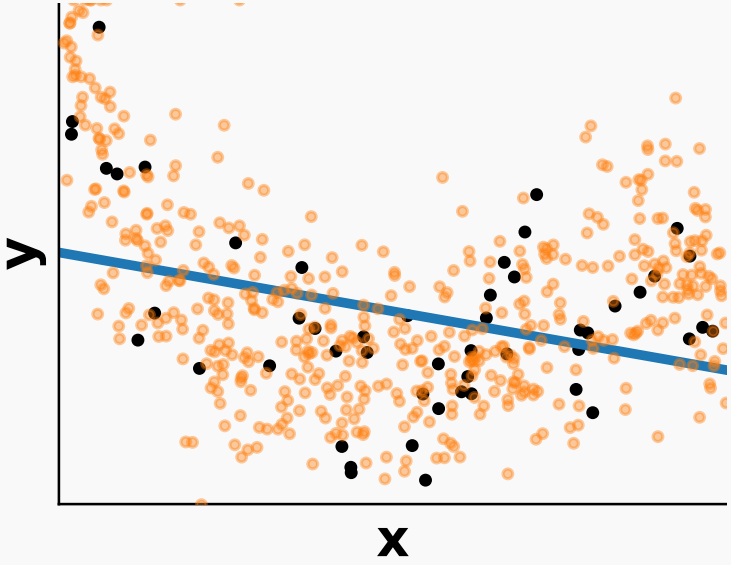


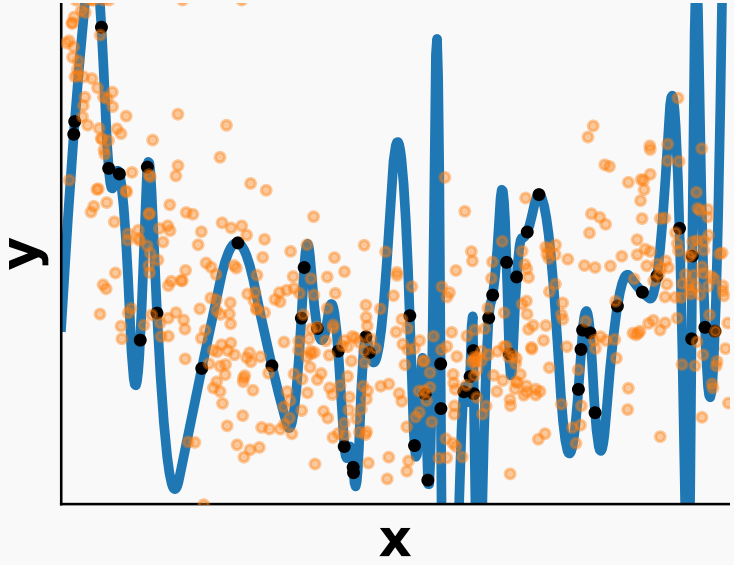
Measure the performances on test data = generalization



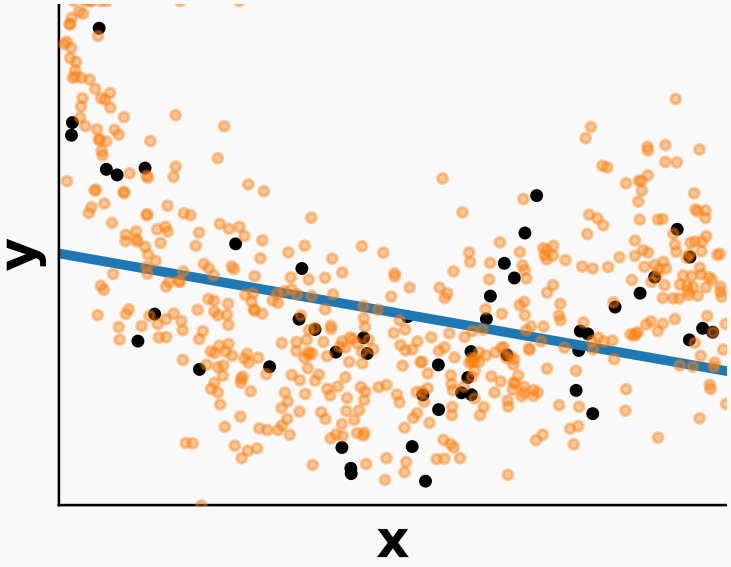
😫 Overfitting: the model is too complex and captures noise.

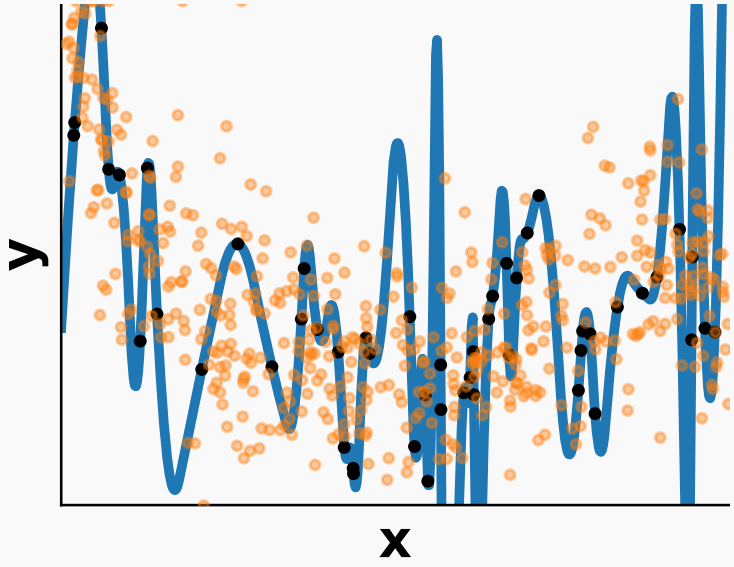
How to choose the complexity of the model?





How to choose the complexity of the model?





This trade-off is is called Bias variance trade-off.

- Let's recover it in the context of statistical learning theory.

Empirical Risk Minimization

- Define a loss function ℓ that defines proximity between the predicted value $\hat{y} = f(x)$ and the true value y: $\ell(f(x), y)$
- Usually, for continuous outcomes, the squared loss is used: $\ell(f(x), y) = (f(x) - y)^2$
- We choose among a (finite) family of functions $f \in \mathcal{F}$, the best possible function $f^\star$ minimizes the risk or expected loss $\mathcal{E}(f) = \mathbb{E}[(f(x) - y)^2]$:

$$f^\star = \operatorname{argmin}_{f \in \mathcal{F}} \mathbb{E}\left[(f(x) - y)^2\right]$$

Empirical risk minimization: estimation error

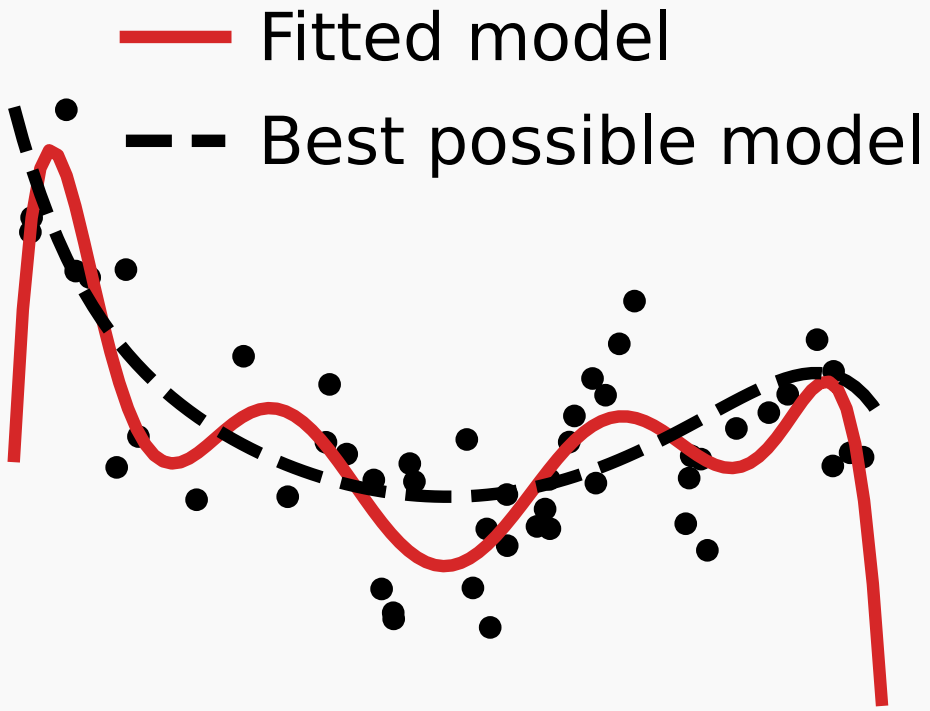
- In finite sample regimes, the expectation is not accessible since we only have access to a finite number of data pairs
- In practice, we minimize the empirical risk or average loss $R_{\text{emp}} = \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2$:

$$\hat{f} = \operatorname{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2$$

- This creates the estimation error, related to sampling noise:

$$\mathcal{E}\left(\hat{f}\right) - \mathcal{E}(f^\star) = \mathbb{E}\left[\left(\hat{f}(x) - y\right)^2\right] - \mathbb{E}\left[\left(f^\star(x) - y\right)^2\right] \geq 0$$

High estimation error means overfit



Model is too complex

- The model is able to recover the true generative process
- But its flexibility captures noise

Too much noise

Not enough data

Bayes error rate: Randomness of the problem

- Interesting problems exhibit randomness

$y = g(x) + e$ with $E(e|x) = 0$ and $\mathrm{Var}(e|x) = \sigma^2$

- The best possible estimator is $g(\cdot)$, yielding the Bayes error, the unavoidable error:

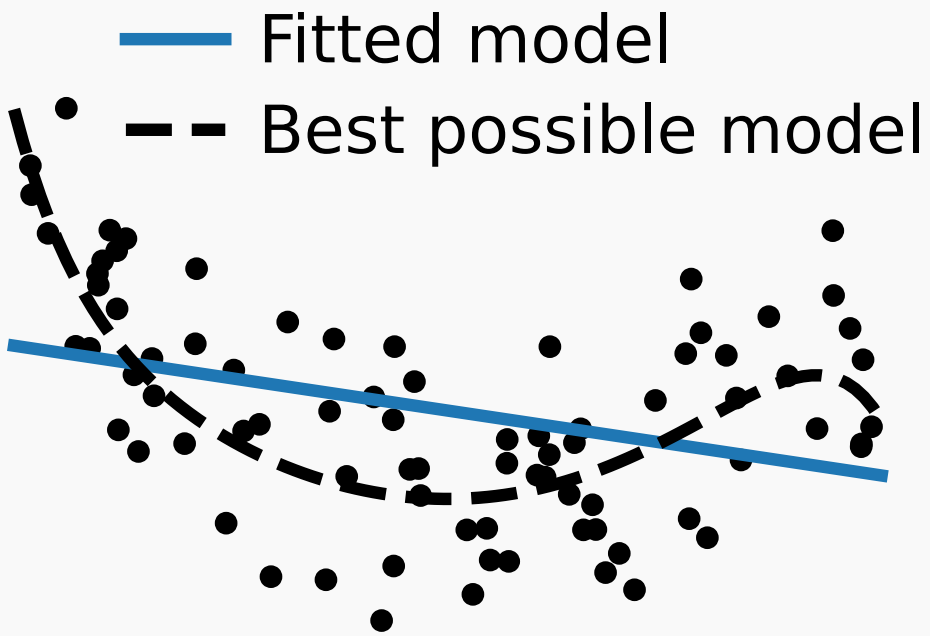
$$\mathcal{E}(g) = \mathbb{E}\big[(g(x) + e - g(x))^2\big] = \mathbb{E}\big[e^2\big]$$

Empirical risk minimization: approximation error

- In practice you don't know the class of function in which the true function lies : $y \approx g(x)$: Every model is wrong !
- You are choosing the best possible function in the class of functions you have access to: $f^\star \in \mathcal{F}$ eg. linear models, polynomials, trees, ...
- This creates the approximation error:

$$\mathcal{E}(f(\star)) - \mathcal{E}(g) = \mathbb{E}\Big[(f^\star(x) - y)^2\Big] - \mathbb{E}\big[(g(x) - y)^2\big] \geq 0$$

High approximation error means underfit



Model is too simple for the data

- its best fit does not approximate the true generative process
- Yet it captures little noise

Low noise

Rapidly enough data to fit the model

Bias variance trade-off: Putting the pieces together

Decomposition of the empirical risk of a fitted model $\hat{f}$

$$\mathcal{E}\left(\hat{f}\right) = \underbrace{\mathcal{E}(g)}_{\text{Bayes error}} + \underbrace{\mathcal{E}(f^\star) - \mathcal{E}(g)}_{\text{approximation error}} + \underbrace{\mathcal{E}\left(\hat{f}\right) - \mathcal{E}(f^\star)}_{\text{estimation error}}$$

Bias variance trade-off: Putting the pieces together

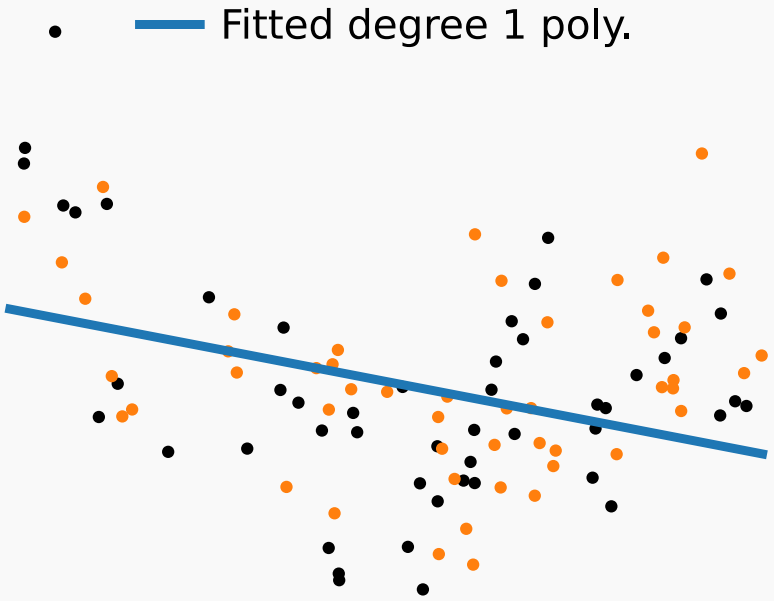
Decomposition of the empirical risk of a fitted model $\hat{f}$

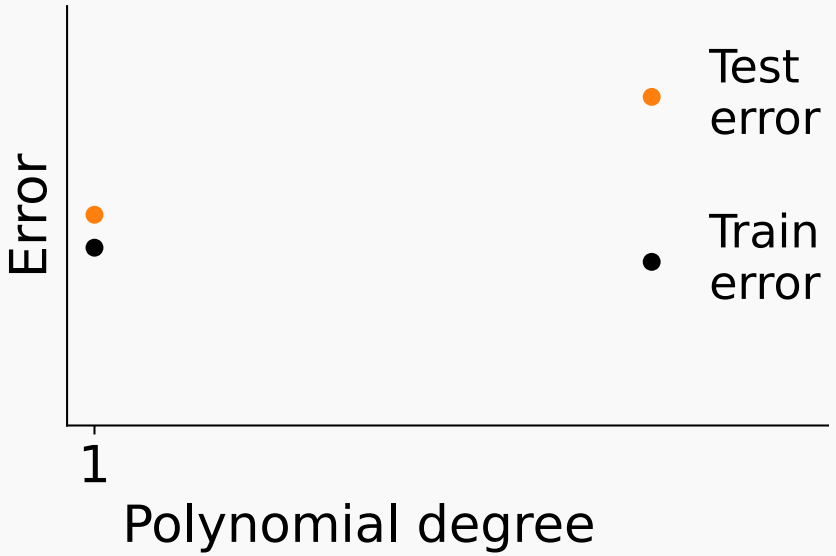
$$\mathcal{E}\left(\hat{f}\right) = \underbrace{\mathcal{E}(g)}_{\text{Bayes error}} + \underbrace{\mathcal{E}(f^\star) - \mathcal{E}(g)}_{\text{approximation error}} + \underbrace{\mathcal{E}\left(\hat{f}\right) - \mathcal{E}(f^\star)}_{\text{estimation error}}$$

Controls on this trade-off

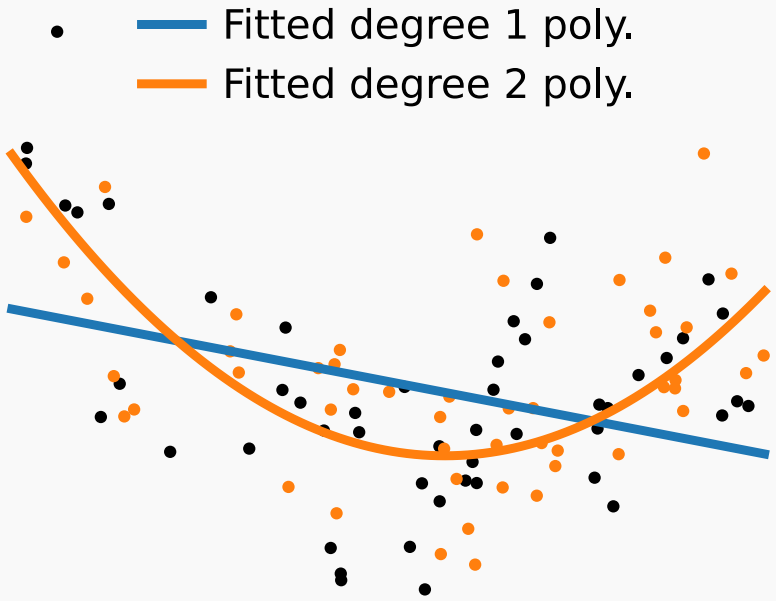
- Increase/decrease the size of the hypothesis family : $\mathcal{F}$ ie. more or less complex models.
- Increase your sample size: n ie. more observations.

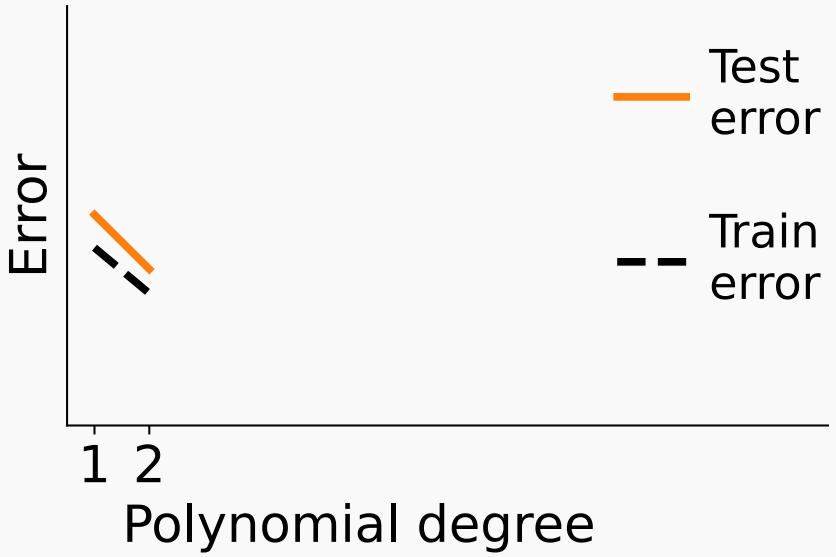
Train vs test error: increasing complexity



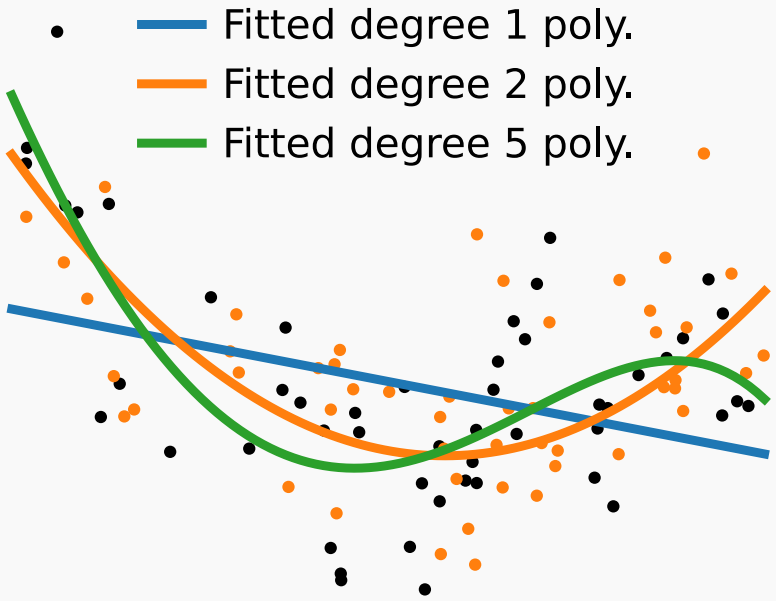


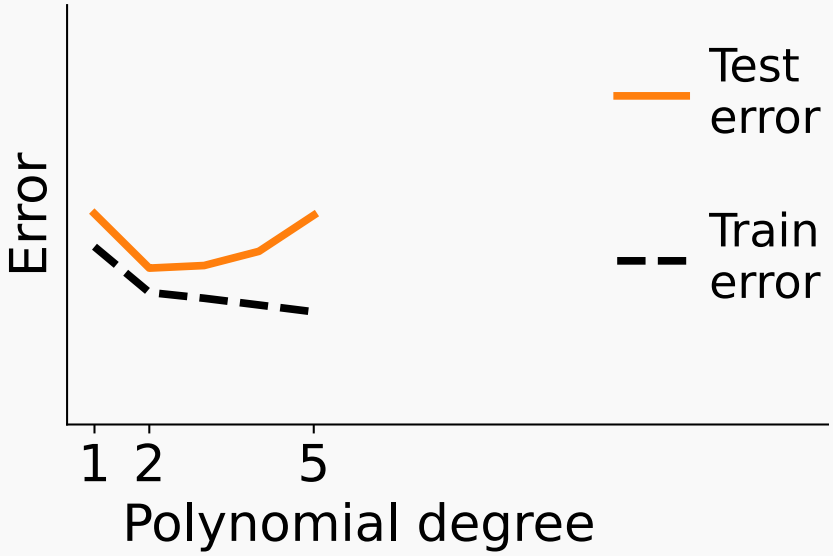
Train vs test error: increasing complexity



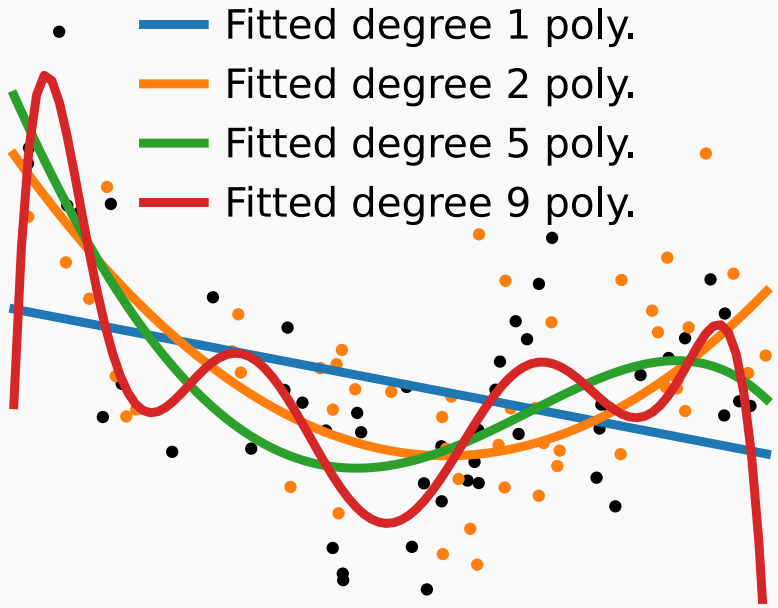


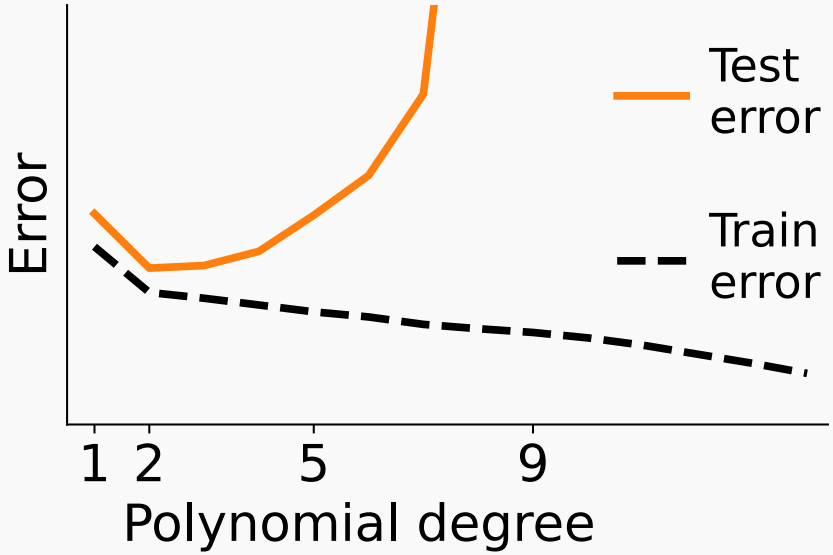
Train vs test error: increasing complexity



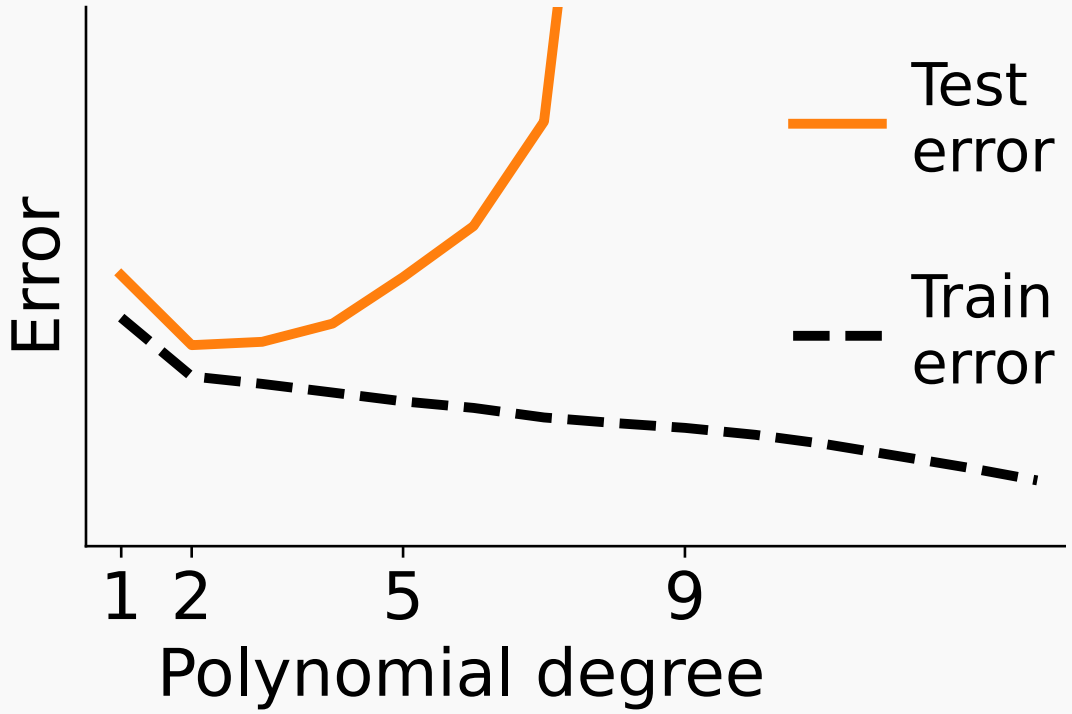


Train vs test error: increasing complexity



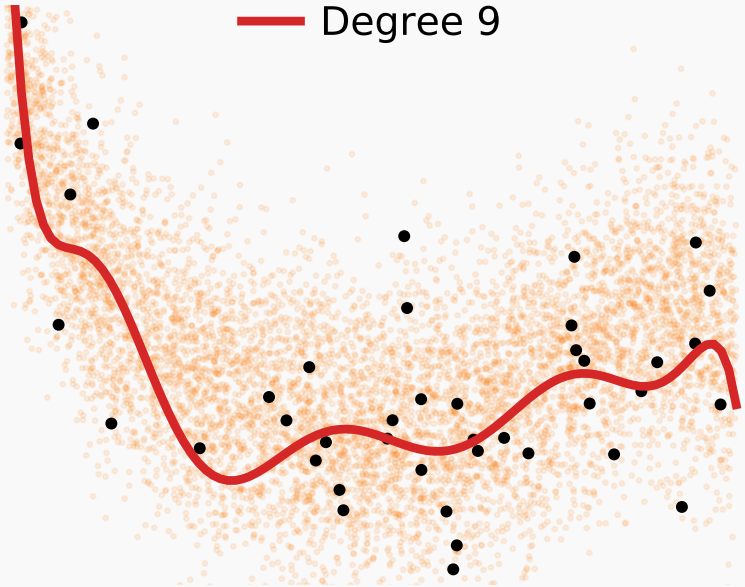


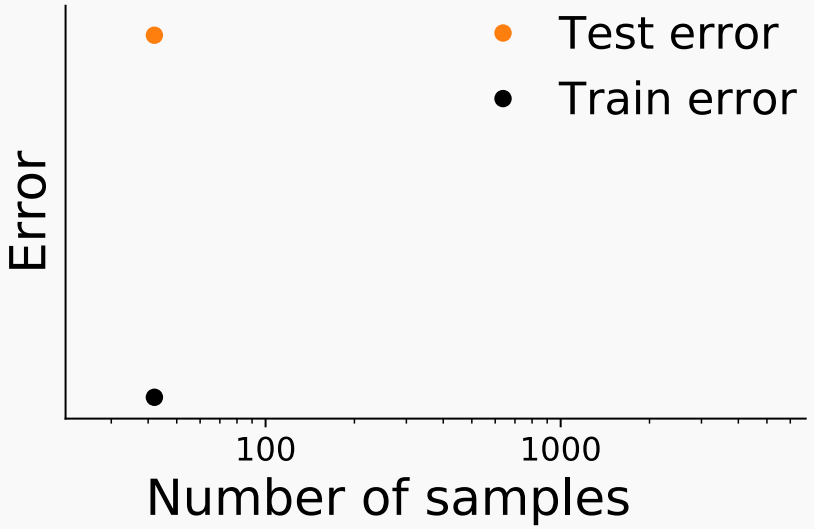
Train vs test error: increasing complexity



Underfit Sweet Spot Overfit

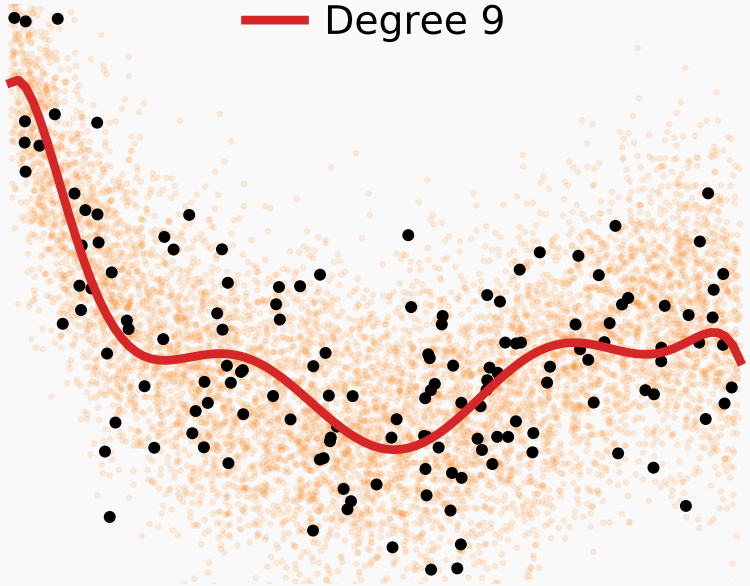
Varying sample size

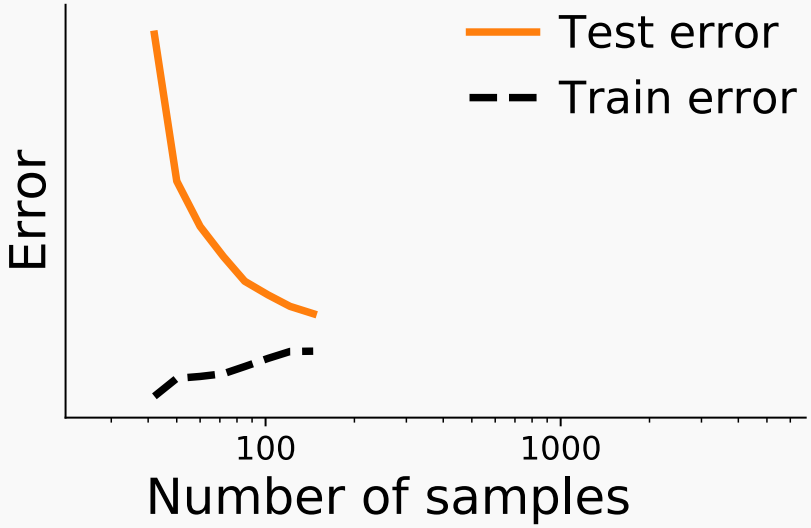




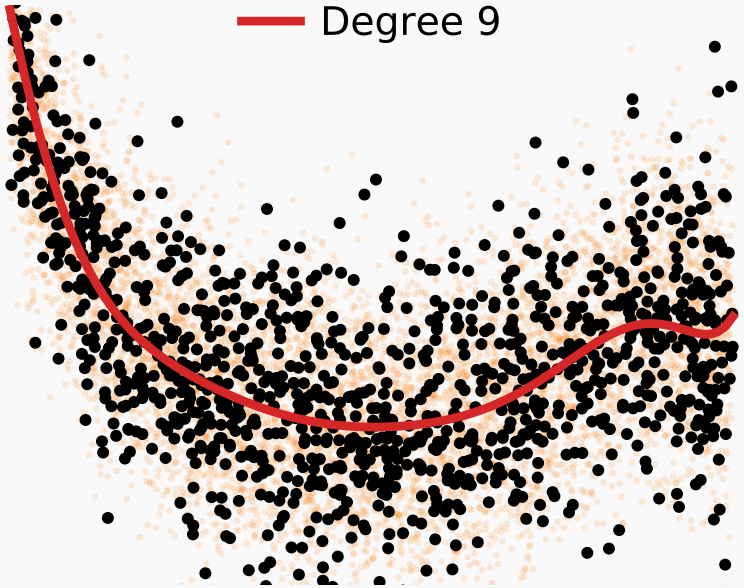
Overfit

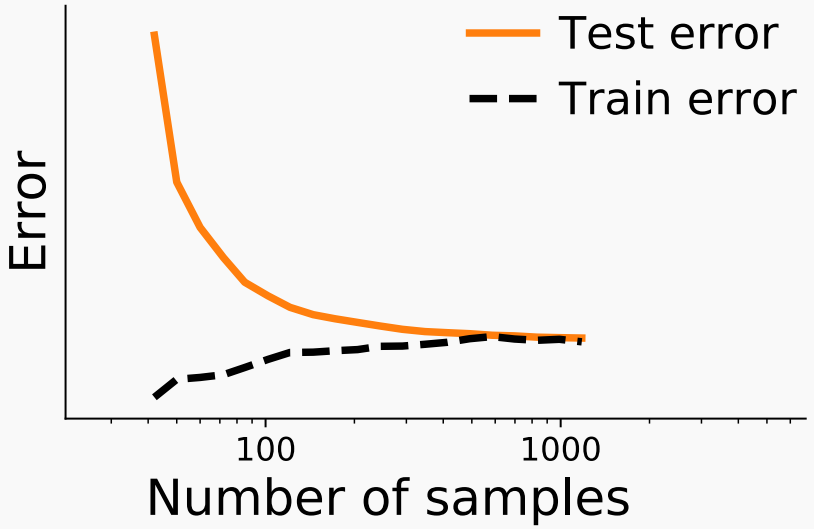
Varying sample size





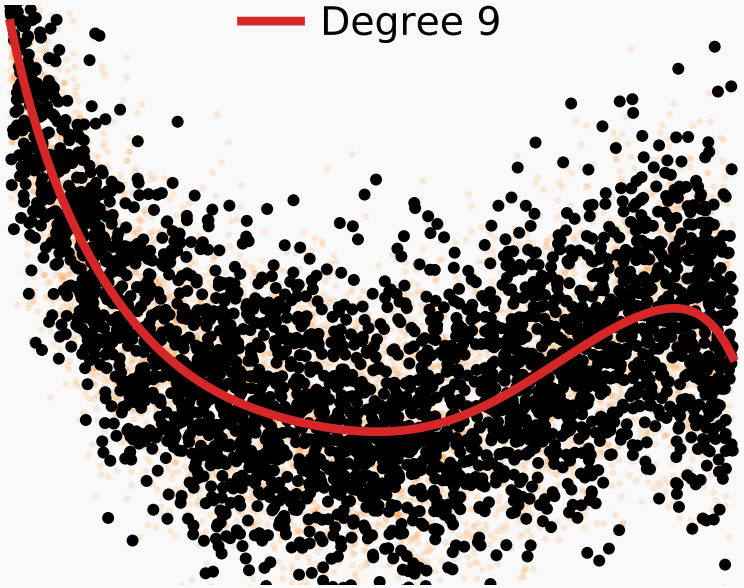
Varying sample size

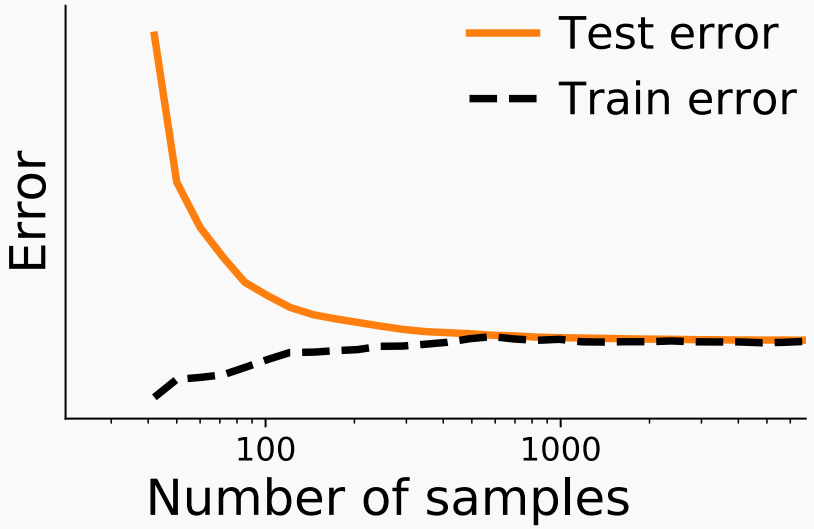




Sweet spot?

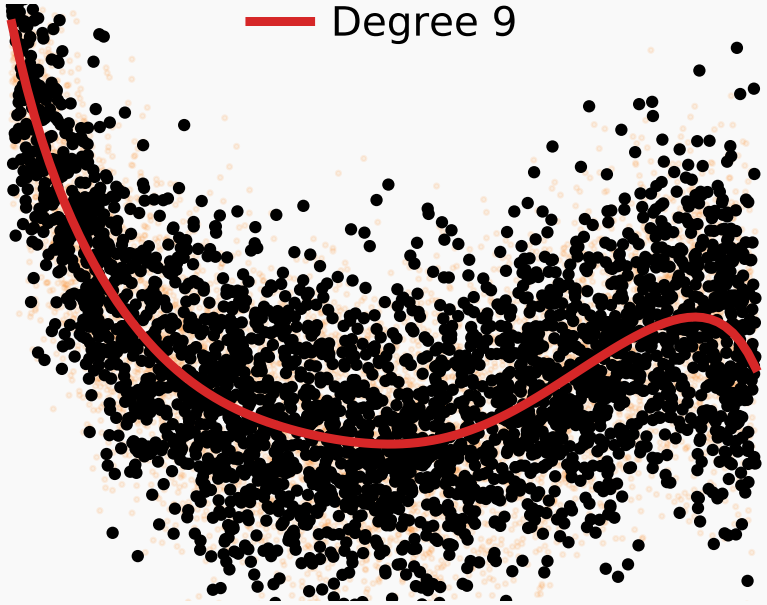
Varying sample size





Diminishing returns?

Varying sample size



The error of the best model trained on unlimited data.

Here, the data is generated by a polynomial of degree 9.

We cannot do better.

Prediction is limited by noise.

Explore common families of models suited to tables data

Today

- Penalized linear regression: Lasso and Ridge
- Hands-on with scikit-learn

Next session

- Flexible models: Trees, Random Forests, Gradient Boosting
- Practical model selection: Cross-validation
- Practical scikit-learn

Lasso for predictive inference

Linear model reminder: Linear regression

y is a linear combination of the features $x \in \mathbb{R}^p$

$$Y_i = X_i^T \beta_0 + \varepsilon_i$$

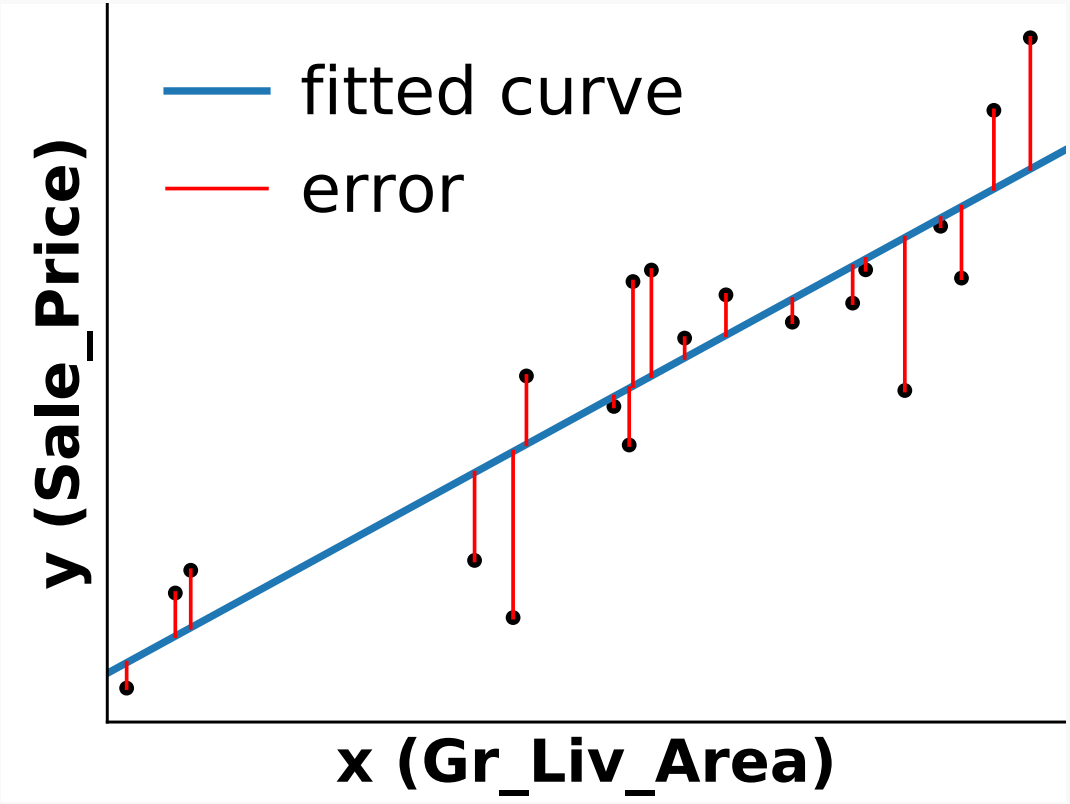
- ε the random variable of the error term.
- $\beta_0 \in \mathbb{R}^{p \times 1}$ the *true* coefficients.

Usually, we assume that the errors are normally distributed and independent of X_i:

$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\varepsilon_i \perp\!\!\!\perp X_i$ Model are typically fitted by linear algebra methods (Hastie, 2009).

Linear model reminder: Linear regression

$$Y_i = X_i^T \beta_0 + \varepsilon_i$$



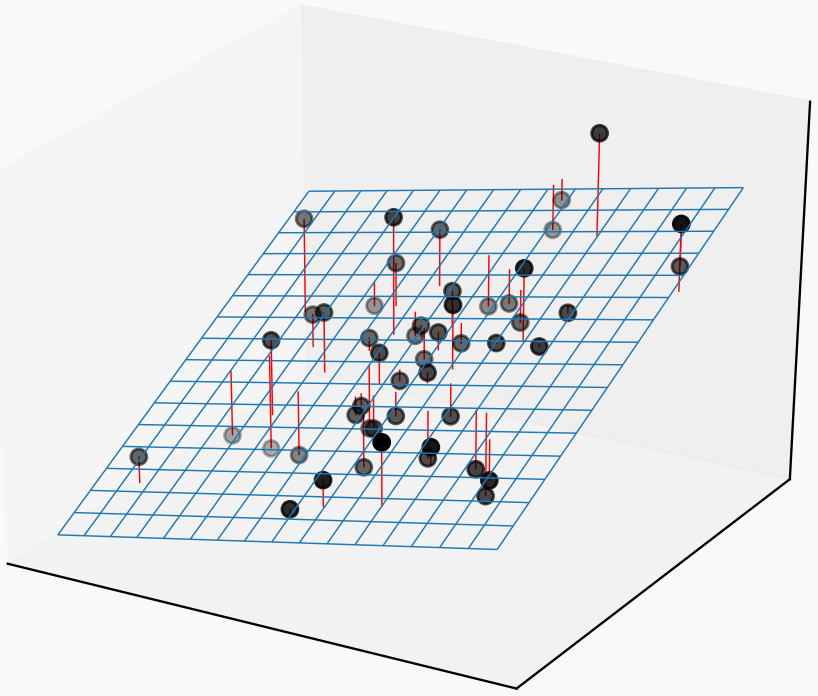
Linear model reminder: Linear regression

Model are typically fitted by linear algebra methods (Hastie, 2009).

Linear model reminder: Linear regression

TODO: Metrics

Linear regression: Two dimension illustration



Linear model reminder: classification, logistic regression

The logit of the probability of the outcome is a linear combination of the features $X_i \in \mathbb{R}^p$:

$$\ln\left(\frac{p(Y_i=1|X_i)}{p(Y_i=0|X_i)}\right) = X_i^T\beta_0$$

which is equivalent to:

$$p(Y_i = 1|X_i, \beta_0) \overset{\text{def}}{=} p(X_i, \beta_0) = \frac{1}{1+\exp(-X_i^T\beta_0)}$$

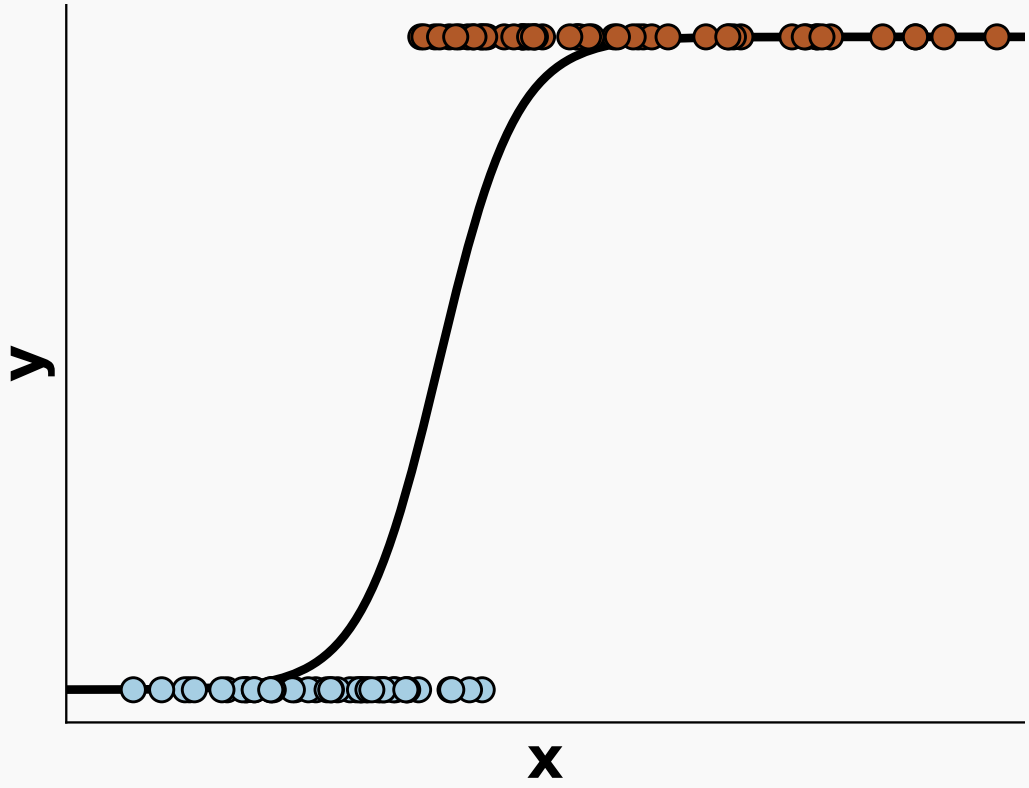
The statistical model is a bernoulli: $B(p(x, \beta_0))$

Models are fitted by maximizing the likelihood by iterative optimization (Hastie, 2009)[1].

[1] eg. coordinate descents (liblinear), second order descent (Newton's method), gradient descent (SAG)...

Linear model reminder: classification, logistic regression

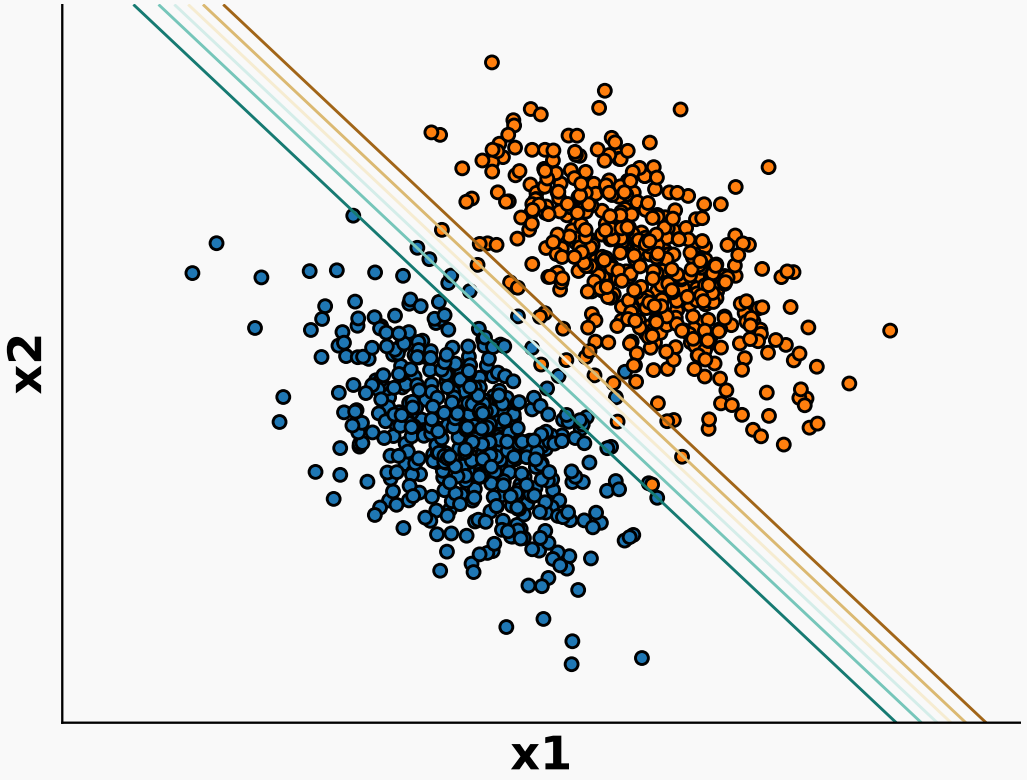
$$p(Y_i = 1|X_i, \beta_0) \stackrel{\text{def}}{=} p(X_i, \beta_0) = \frac{1}{1+\exp(-X_i^T \beta_0)}$$



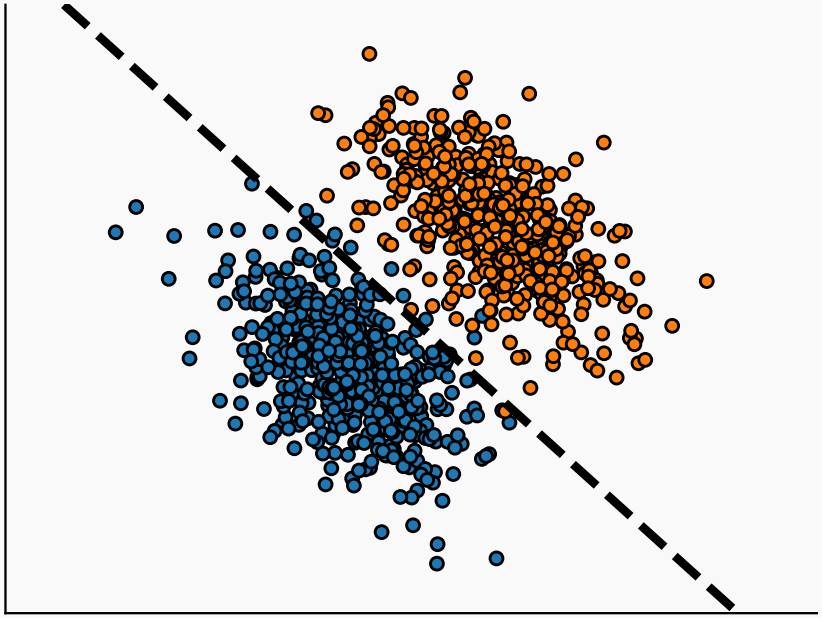
Linear model reminder: classification, logistic regression

TODO: Metrics

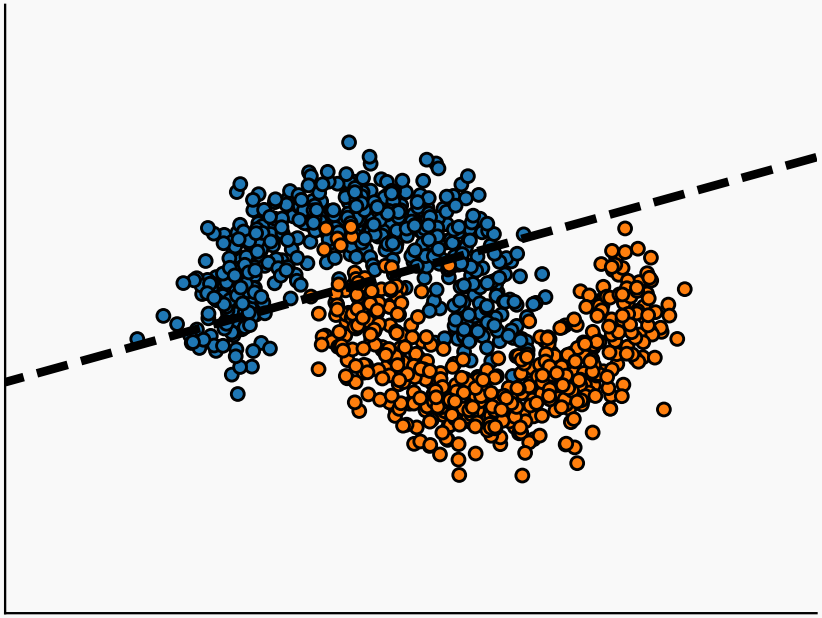
Logistic regression: Two dimension illustration



Linear models are not suited to all data



Almost linearly separable data.



Data not linearly separable.

Linear model pros and cons

Pros

- Converge quickly
- Hard to beat when n_features is large but we still have n_samples >> n_features
- Linear models work well if
  - the classes are (almost) linearly separable
  - or the outcome is (almost) linearly related to the features.

Linear model pros and cons

Cons

Sometimes

- the best decision boundary to separate classes is not well approximated by a straight line.
- there are important non-linear relationships between the features and the outcome.

Linear model pros and cons

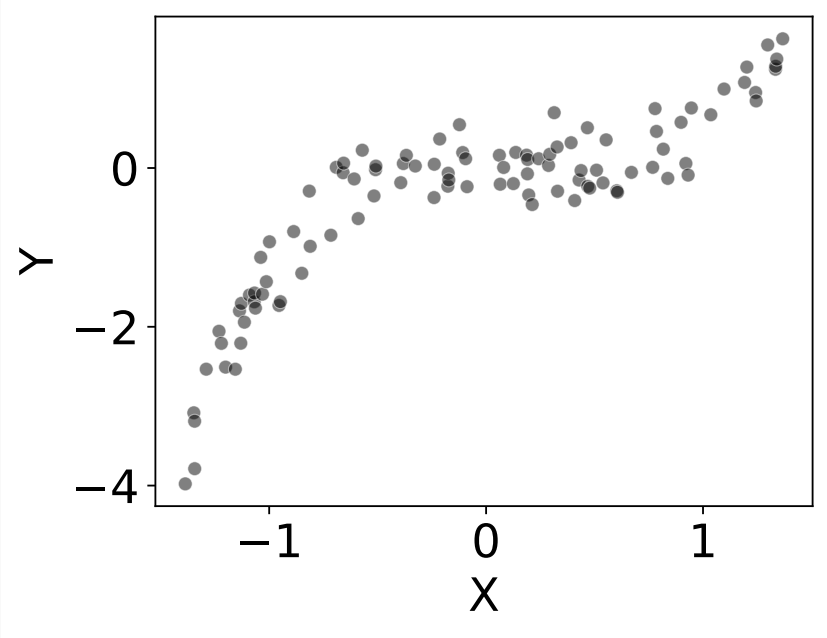
Cons

Sometimes

- the best decision boundary to separate classes is not well approximated by a straight line.
- there are important non-linear relationships between the features and the outcome.

💡 Either use non-linear models, or perform transformations on the data, to engineer new features.

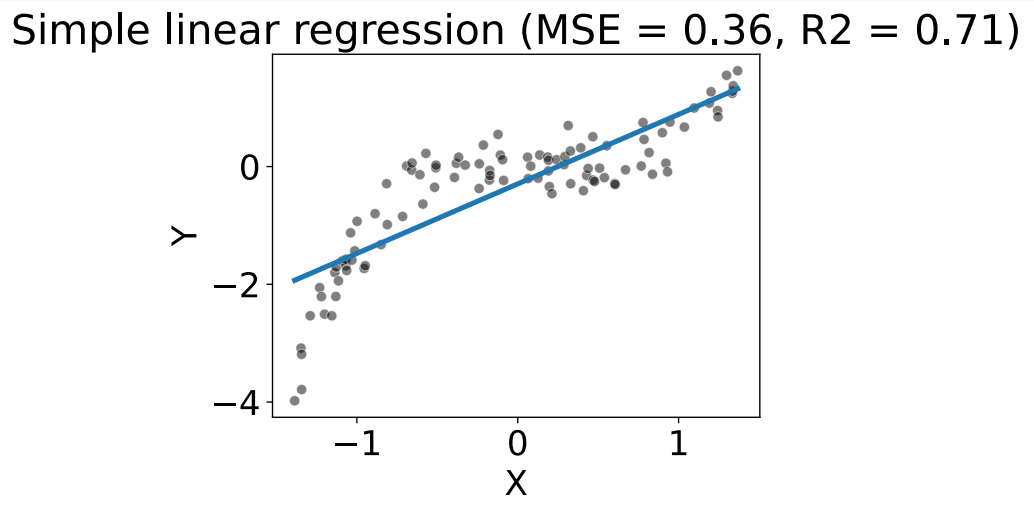
Transformation of the features: Example



Non-linear relationship between the features and the outcome:

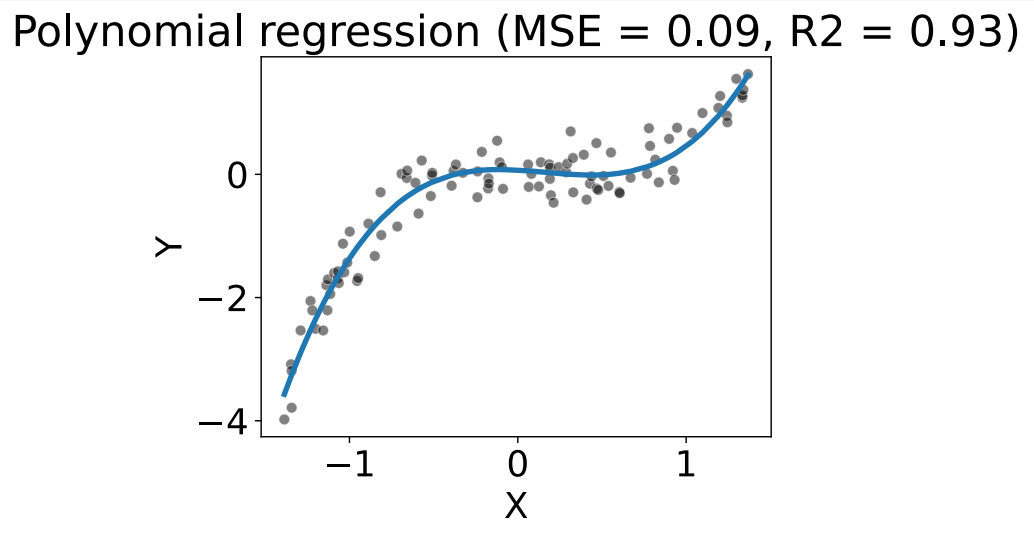
$$Y = X^3 - 0.5 \times X^2 + \varepsilon$$

Transformation of the features: Example



Vanilla Linear regression fails to capture the relationship.

Transformation of the features: Example



Solution:

- Expand the feature space with polynoms of the features:

$$X = [X, X^2, X^3]$$

- Run a linear regression on the new feature space.

$$Y = [X, X^2, X^3]^T \hat{\beta}$$

Sparsity

Lasso: intuition

Post-Lasso: intuition

Pitfalls on using Lasso for variable selections

Ridge regression

Elastic net

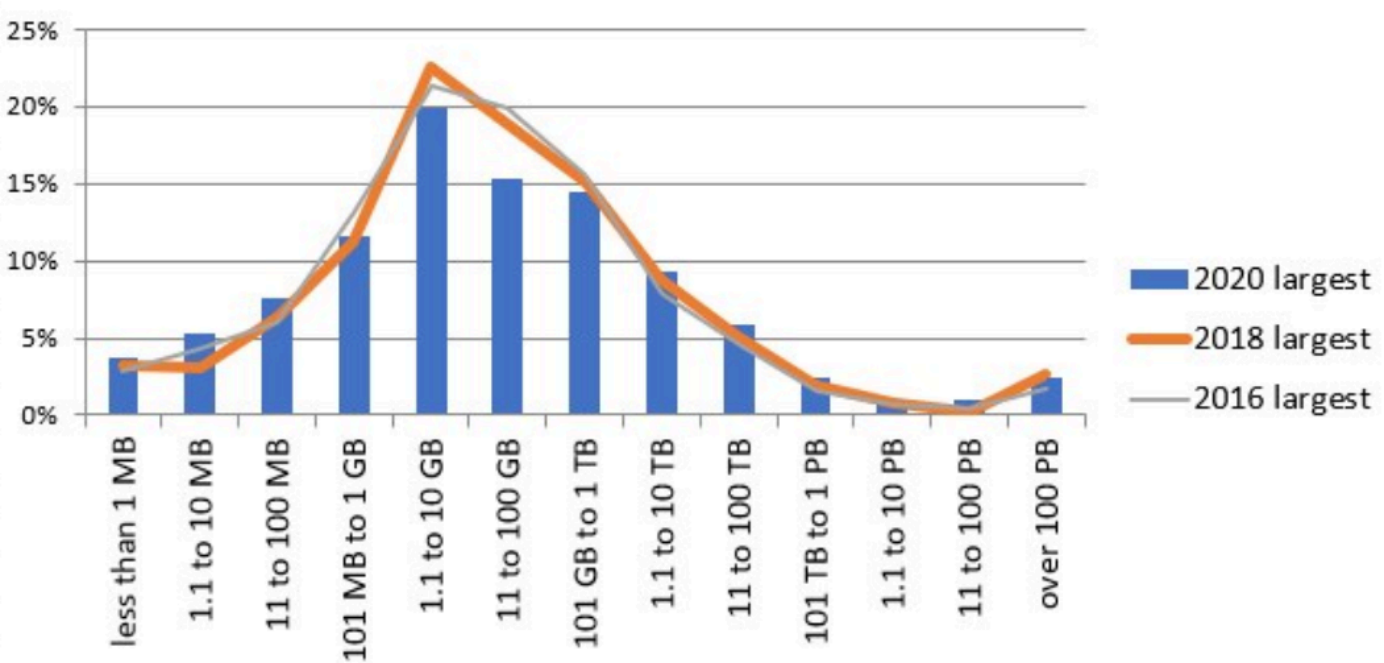
A word on deep learning

Why not use deep learning everywhere?

- Success of deep learning in image, speech recognition and text
- Why not so used in economics?

Limited data settings

- Typically in economics everywhere, we have a limited number of observations

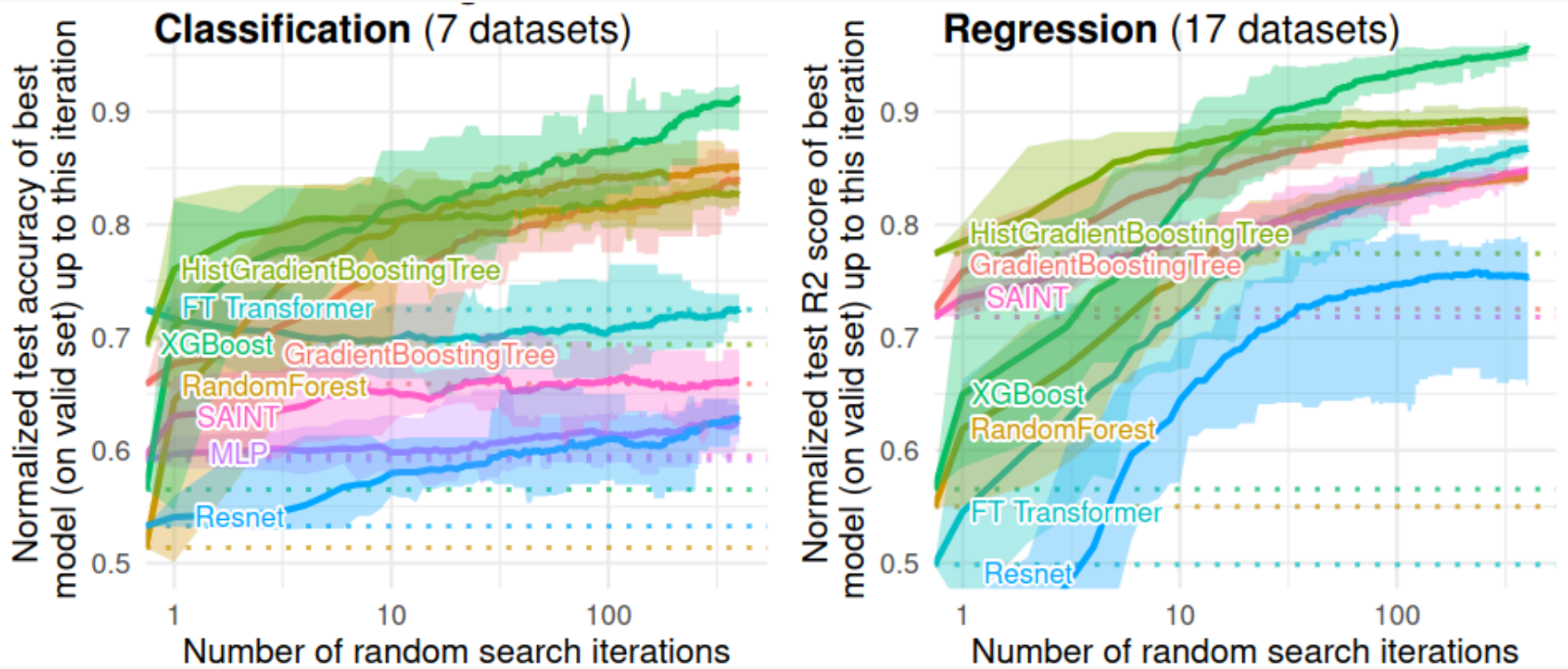


Typical dataset are mid-sized. This does not change with time.[2]

[2] https://www.kdnuggets.com/2020/07/poll-largest-dataset-analyzed-results.html

Deep learning underperforms on data tables

Tree-based methods outperform tailored deep learning architectures (Grinsztajn et al., 2022)



DAG for a RCT: the treatment is independent of the confounders

Take home messages: Bias-variance trade-off

High bias == underfitting

- systematic prediction errors
- the model prefers to ignore some aspects of the data
- mispecified models

High variance == overfitting:

- prediction errors without obvious structure
- small change in the training set, large change in model
- unstable models

Take home messages: Lasso and Ridge

Practical session

Resources

- https://web.stanford.edu/~swager/stats361.pdf
- https://www.mixtapesessions.io/
- https://alejandroschuler.github.io/mci/
- https://theeffectbook.net/index.html

Bibliography

Estève, L., Lemaitre, G., Grisel, O., Varoquaux, G., Amor, A., Lilian, Rospars, B., Schmitt, T., Liu, L., Kinoshita, B. P., hackmd-deploy, ph4ge, Steinbach, P., Boucaud, A., Muite, B., Boisberranger, J. du, Notter, M., Pierre, P, S., … parmentelat. (2022). INRIA/scikit-learn-mooc: Third MOOC session. Zenodo. https://doi.org/10.5281/zenodo.7220307

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data?. Advances in Neural Information Processing Systems, 35, 507–520.

Hastie, T. (2009,). The elements of statistical learning: data mining, inference, and prediction. Springer.

Kleinberg, J., Ludwig, J., Mullainathan, S., & Obermeyer, Z. (2015). Prediction policy problems. American Economic Review, 105(5), 491–495.