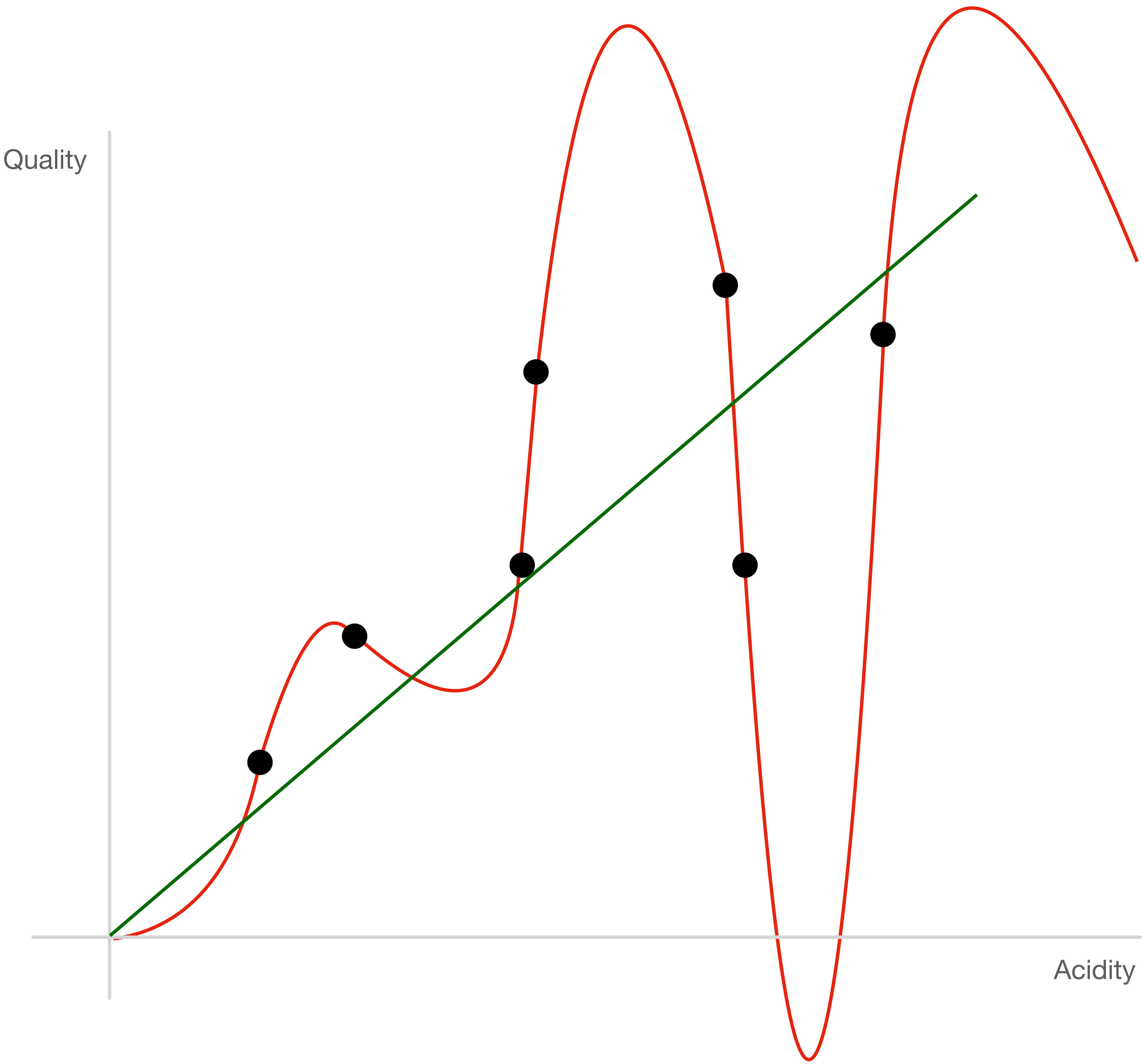
AIBridge

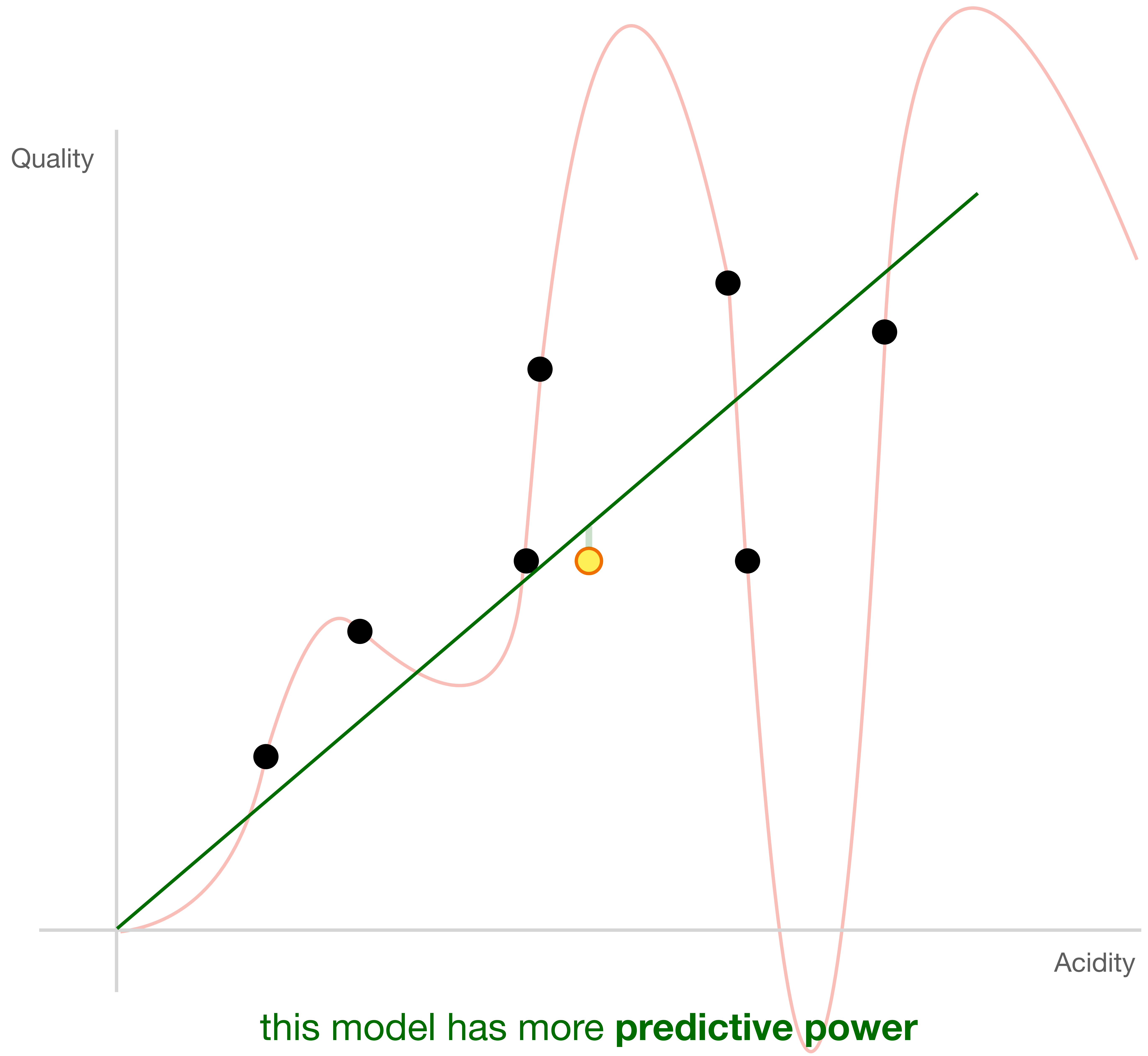
Lecture 7

overfitting



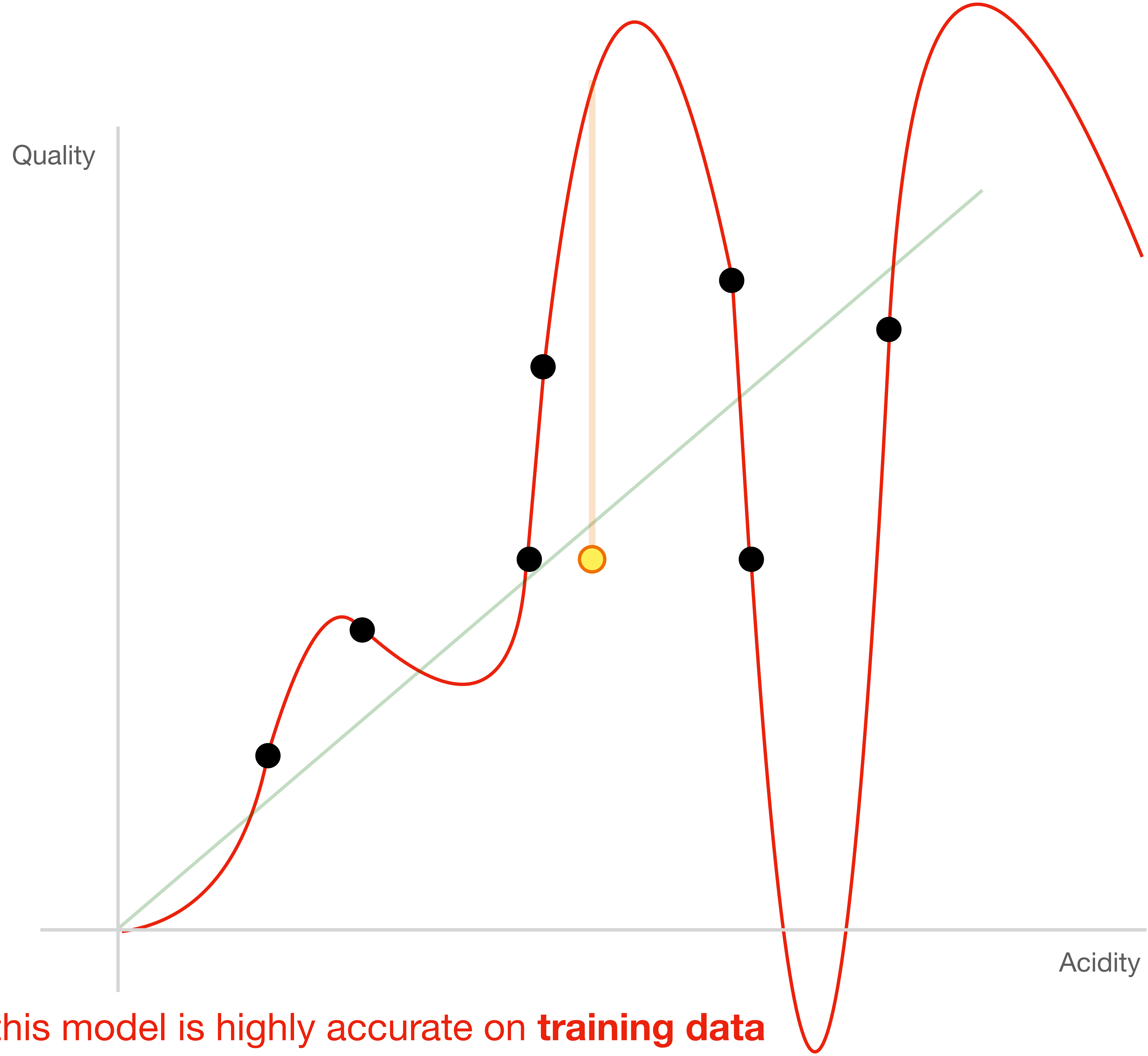
Which one is a better line?

overfitting



this model has more **predictive power**

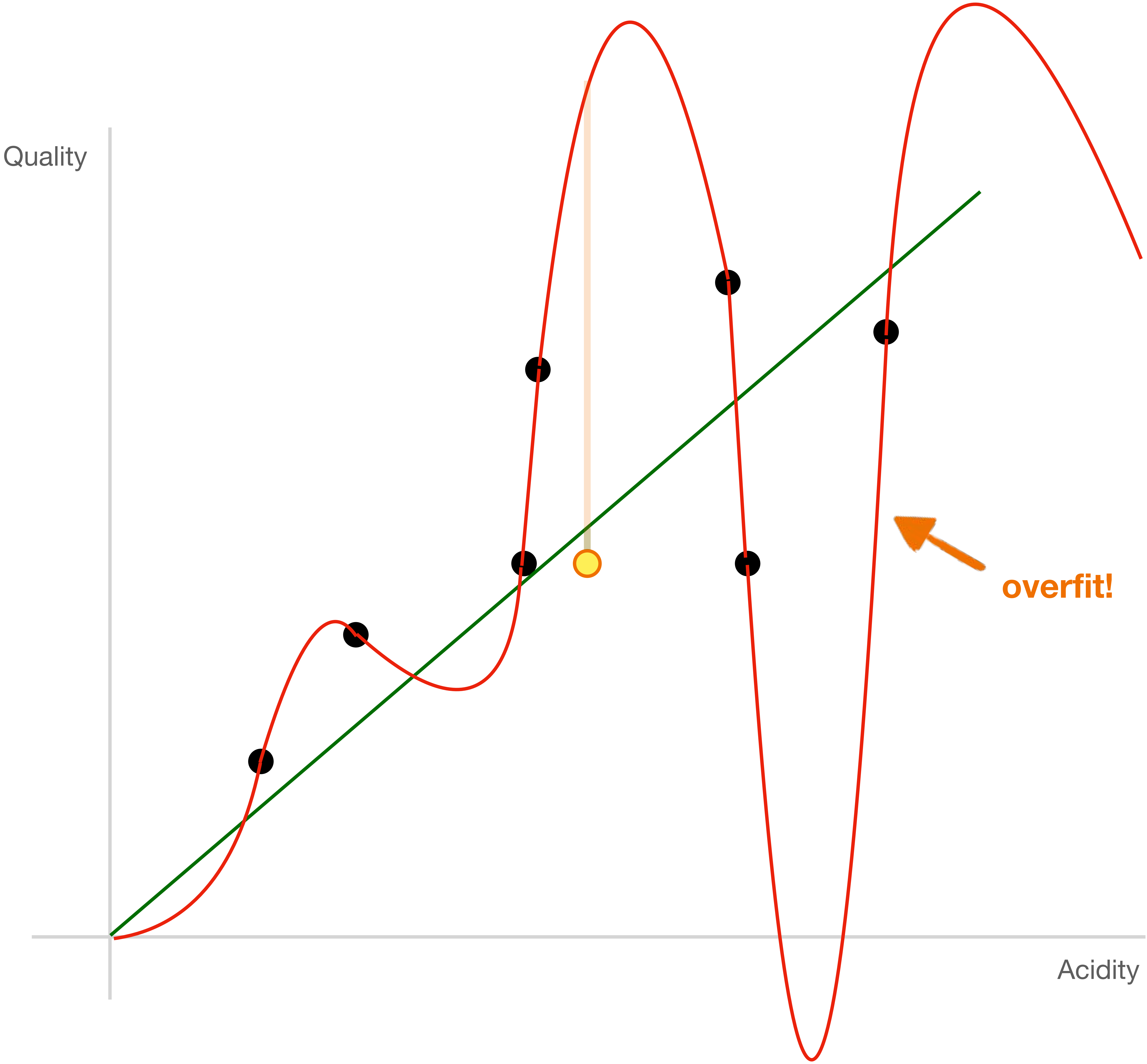
overfitting



this model is highly accurate on **training data**

but bad at predictions anywhere else

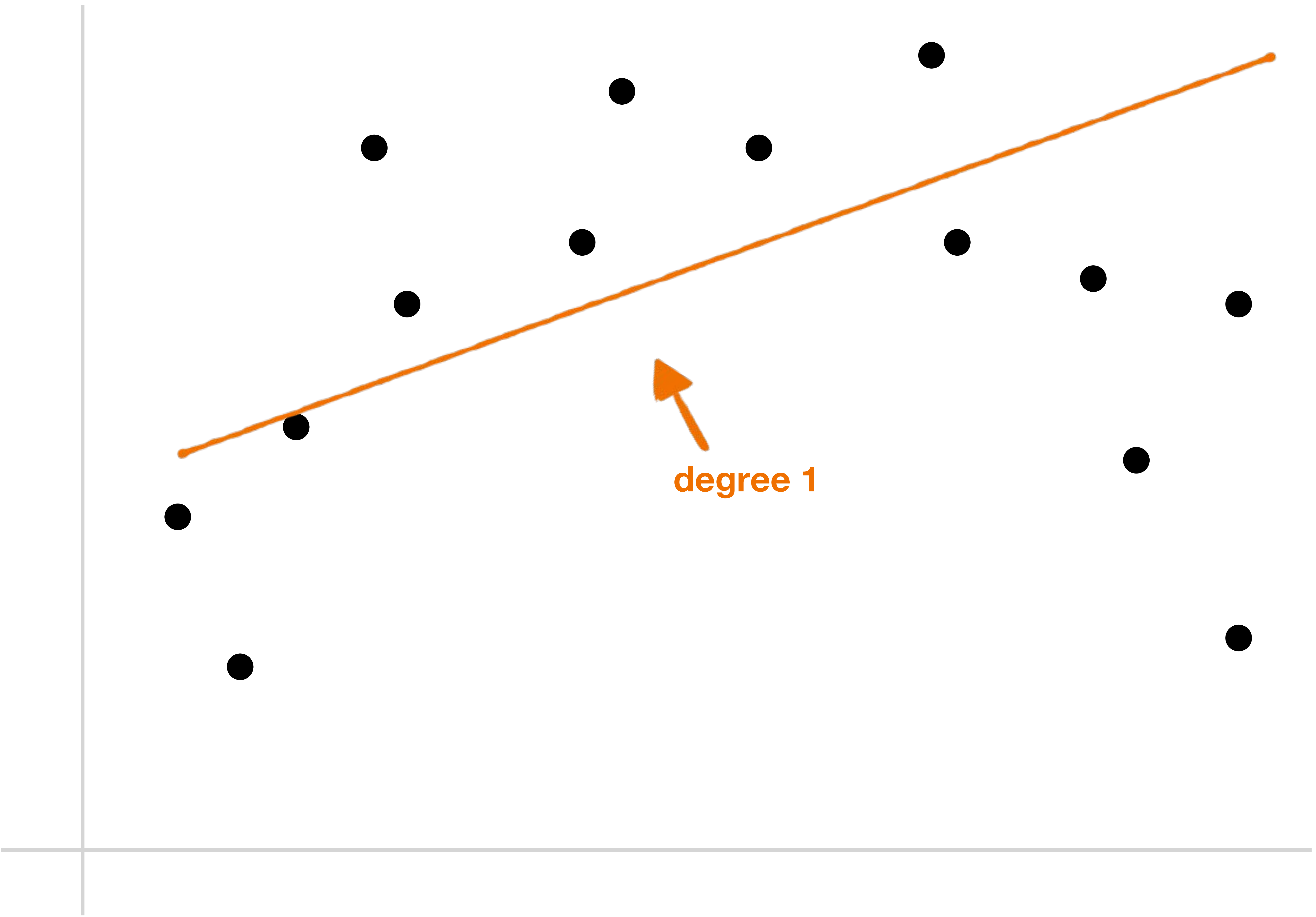
overfitting



- too-precise fits to original data without generalization is called **overfitting**

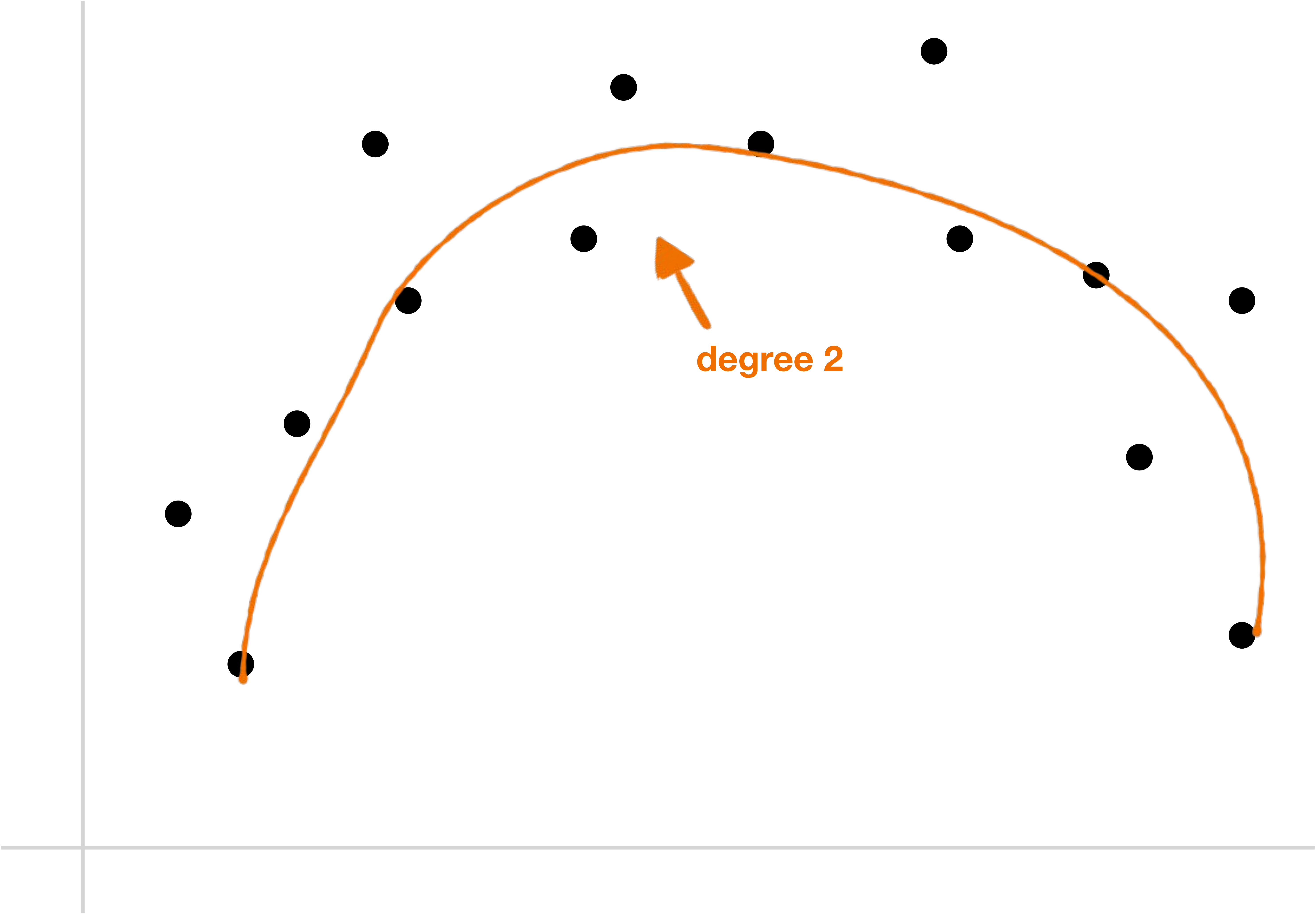
underfitting

underfit!

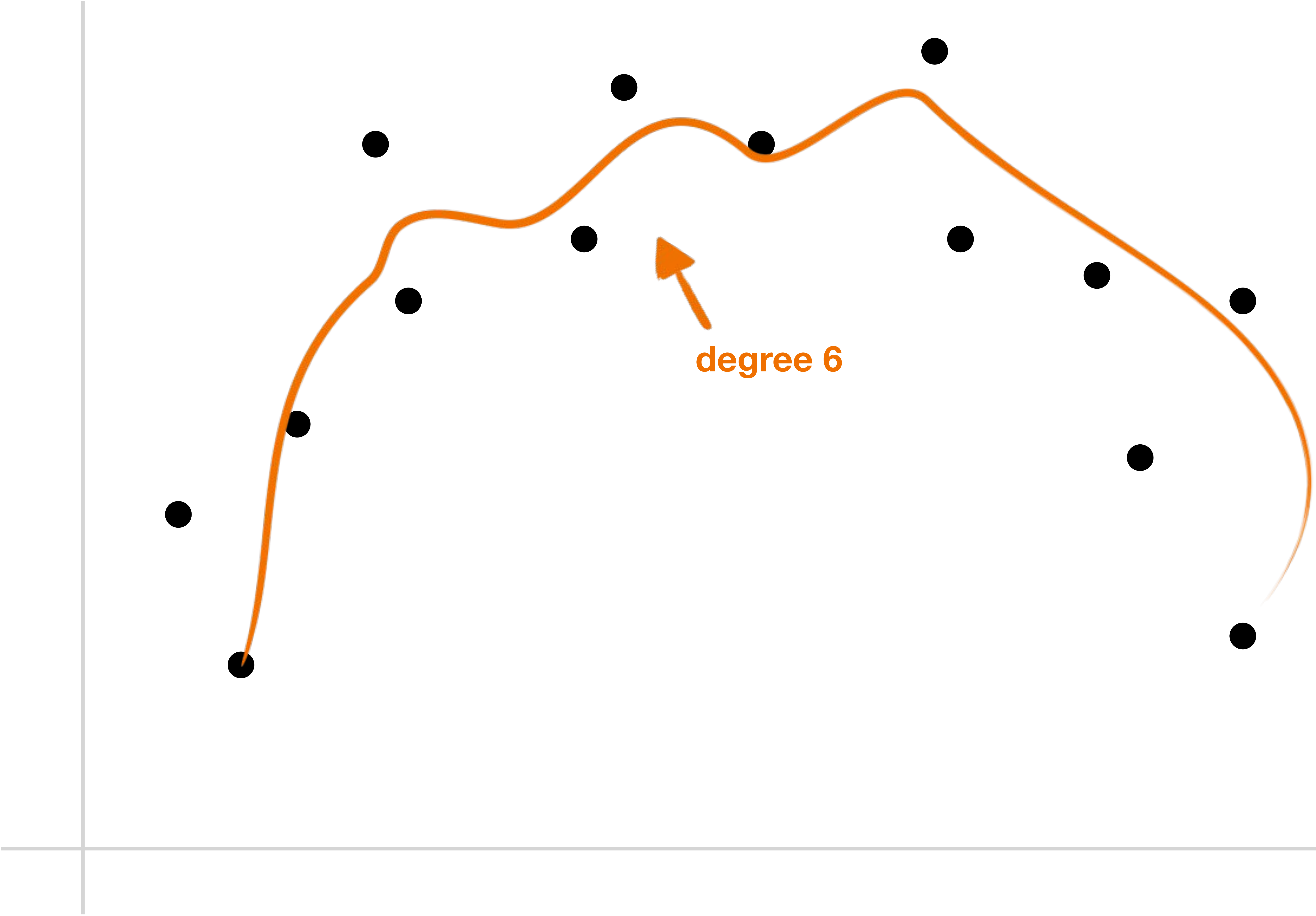


- model is unable to capture relationship between variables

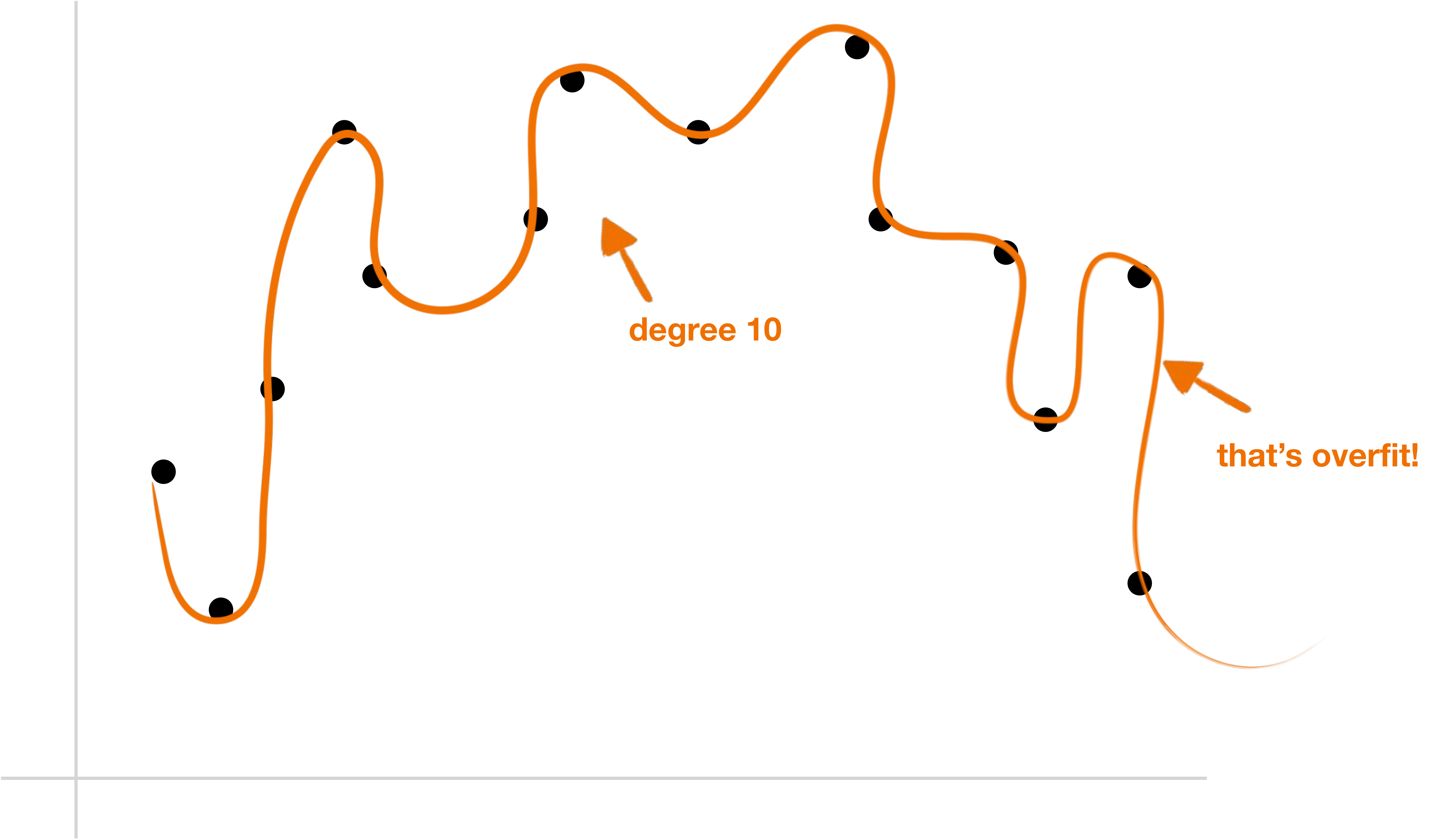
fitting



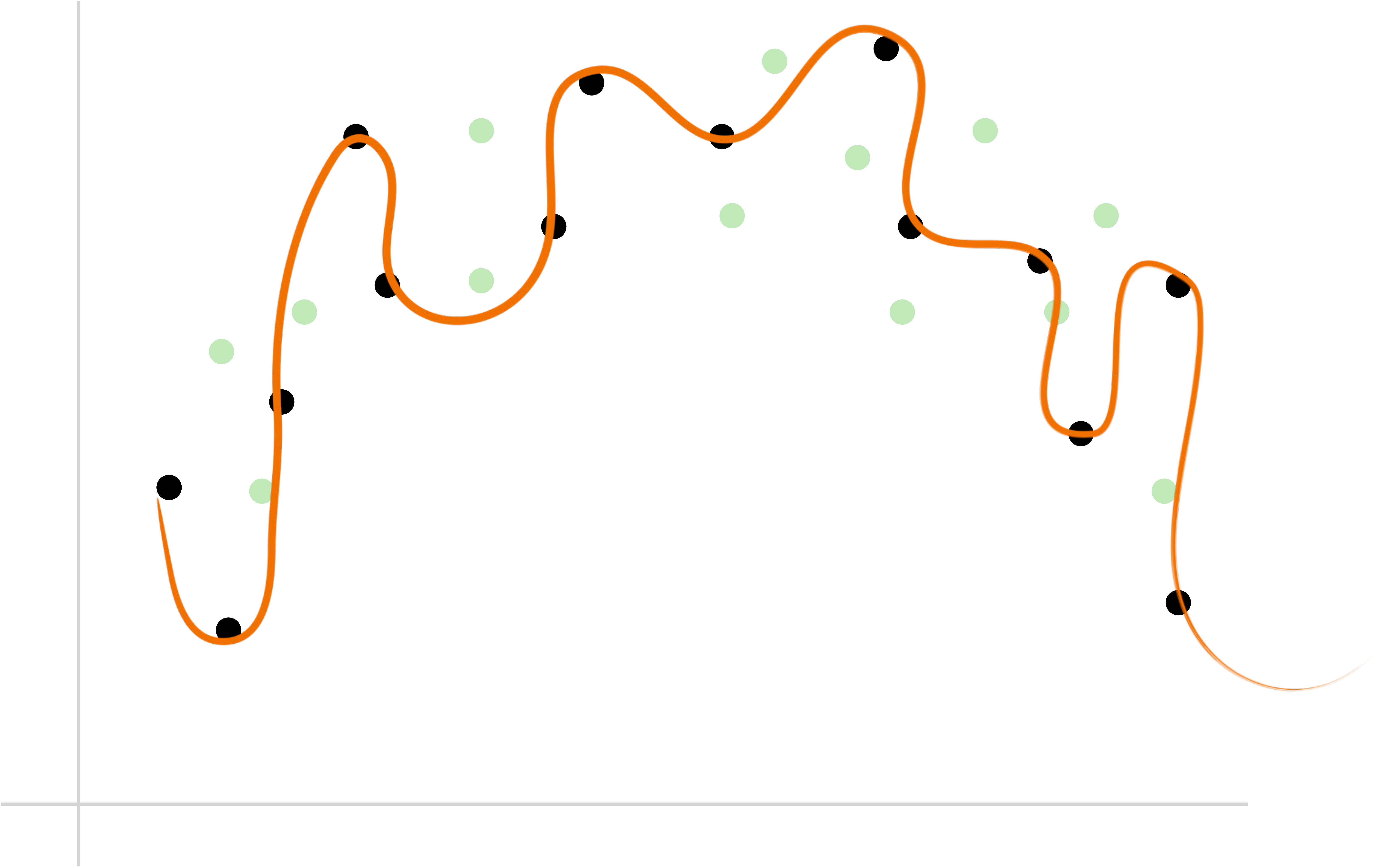
overfitting



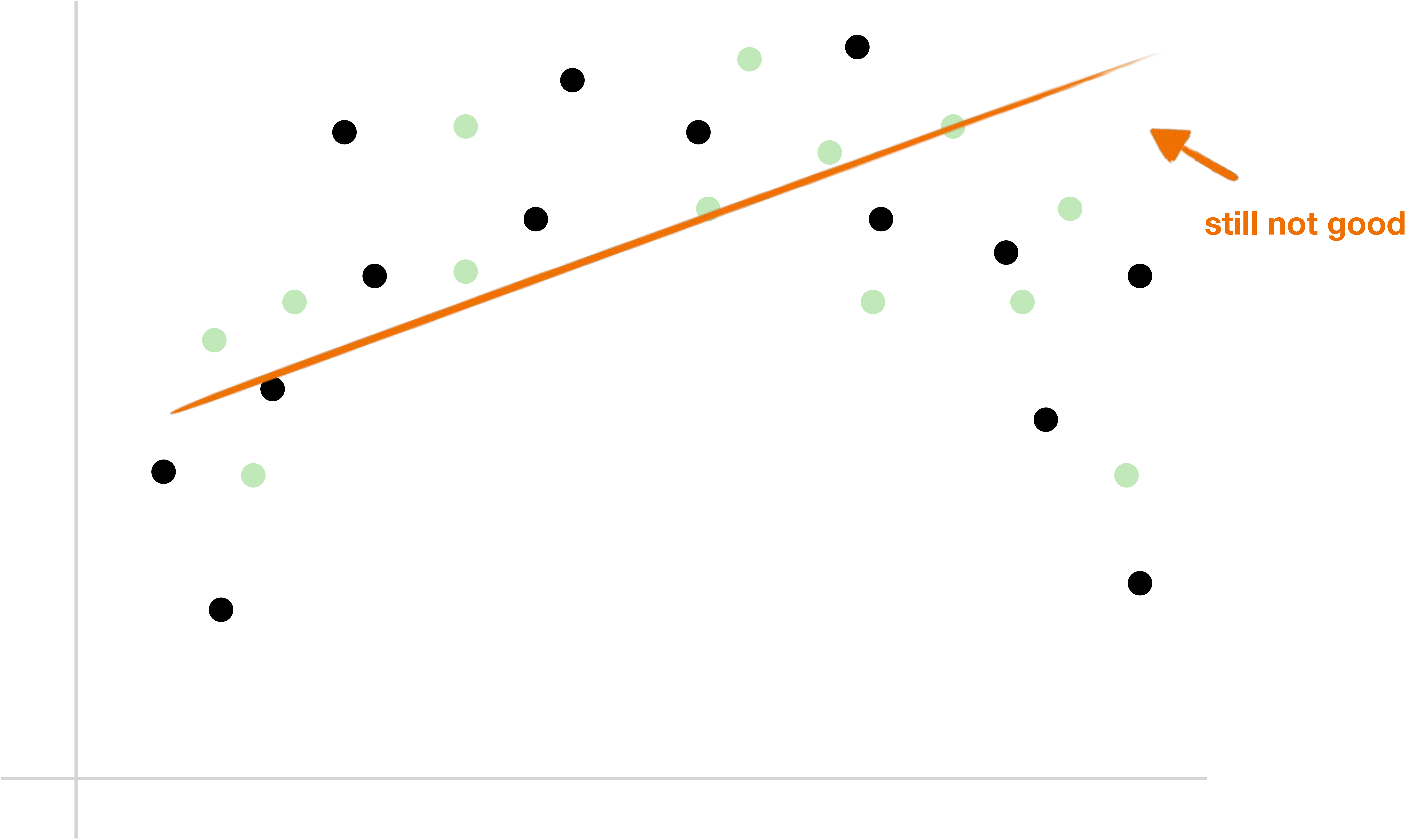
overfitting



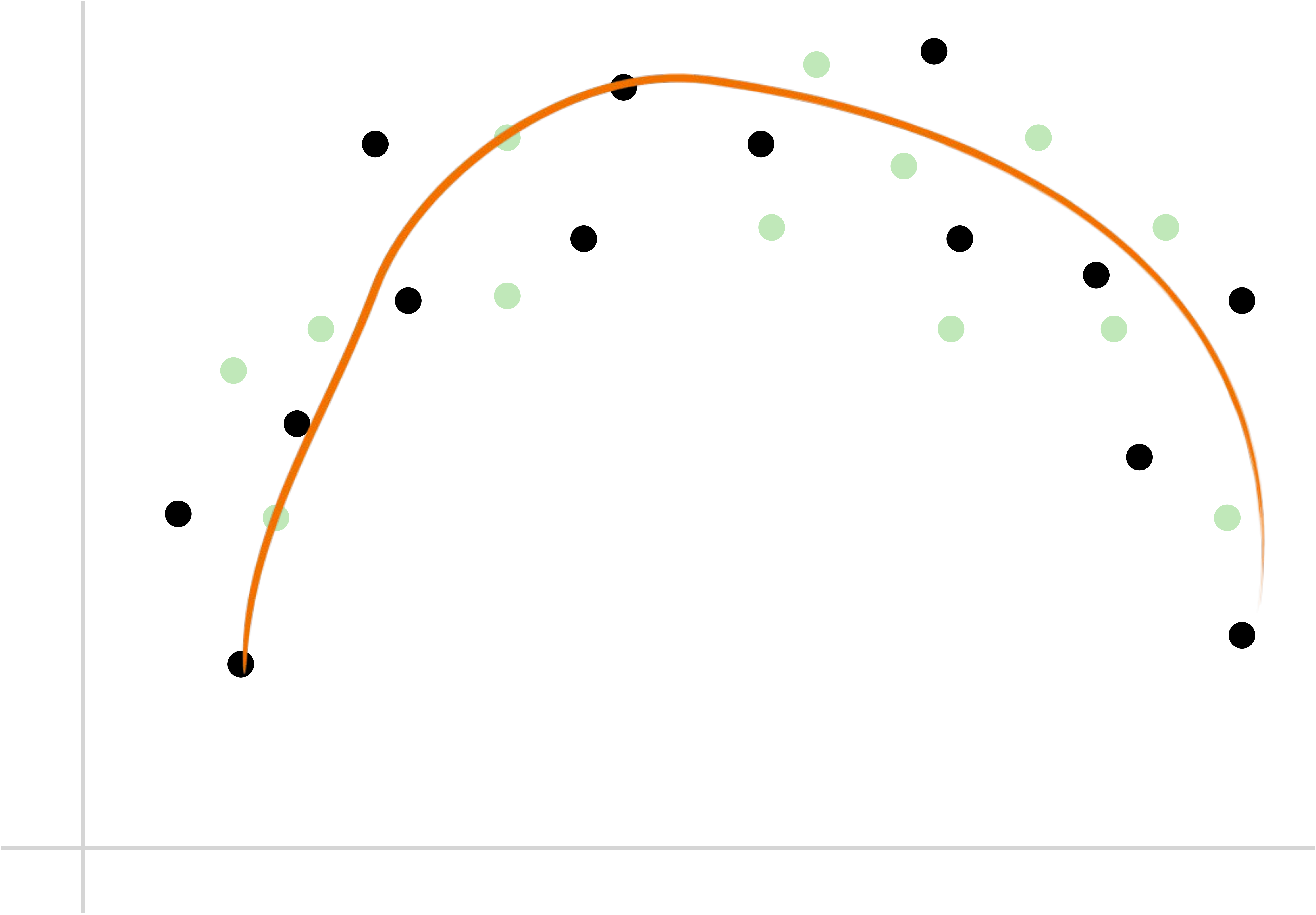
overfitting



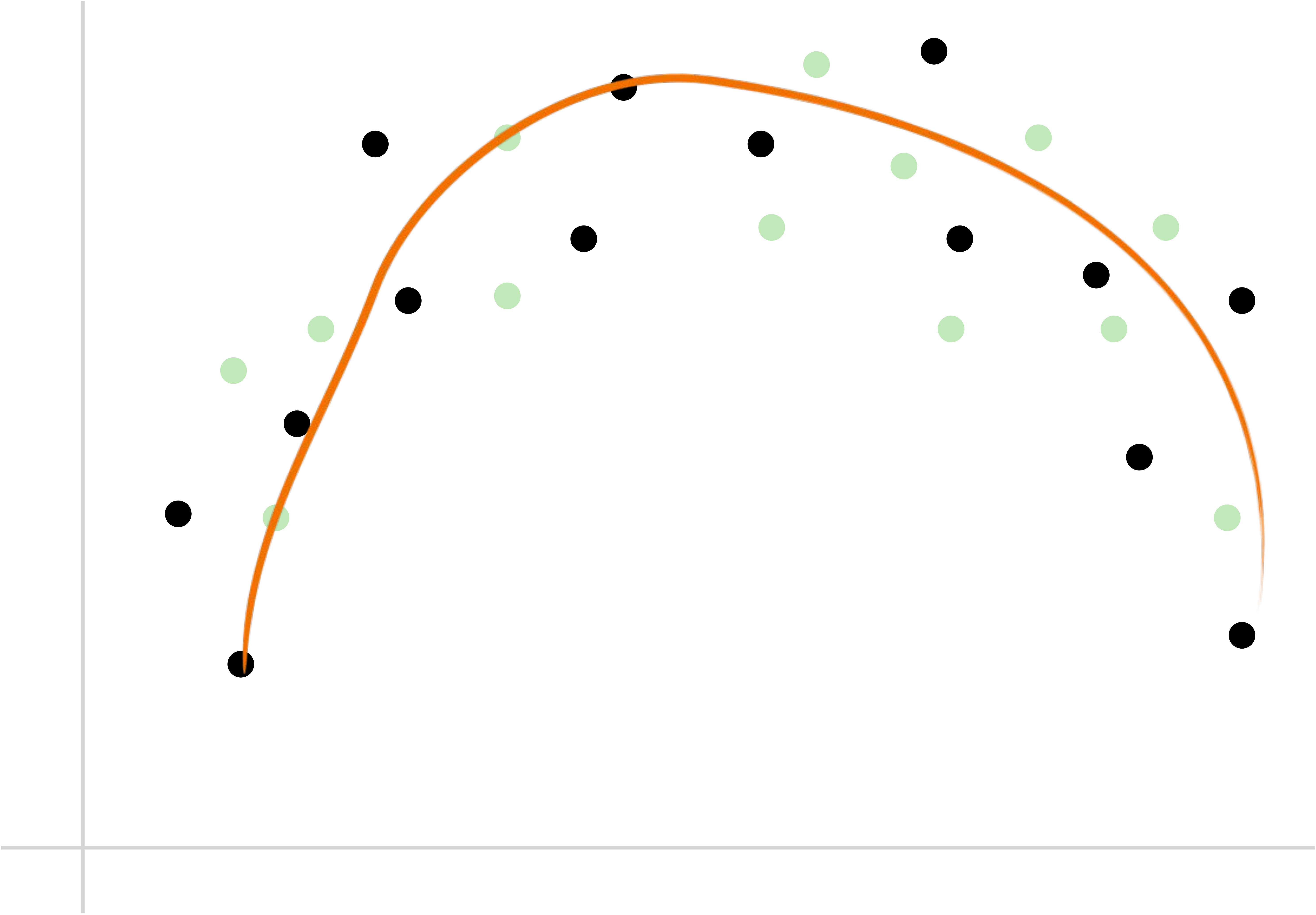
underfitting



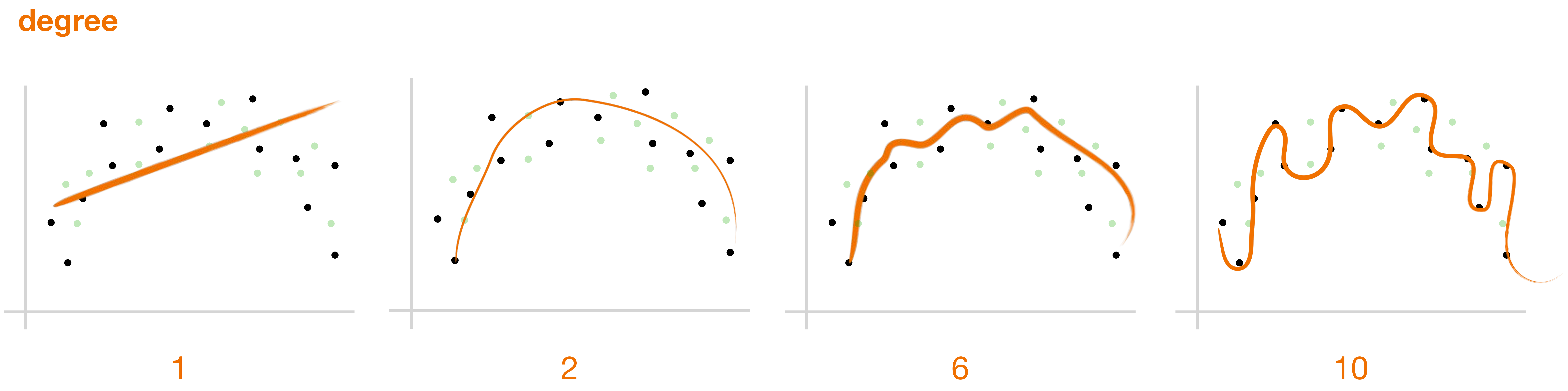
overfitting



overfitting



overfitting



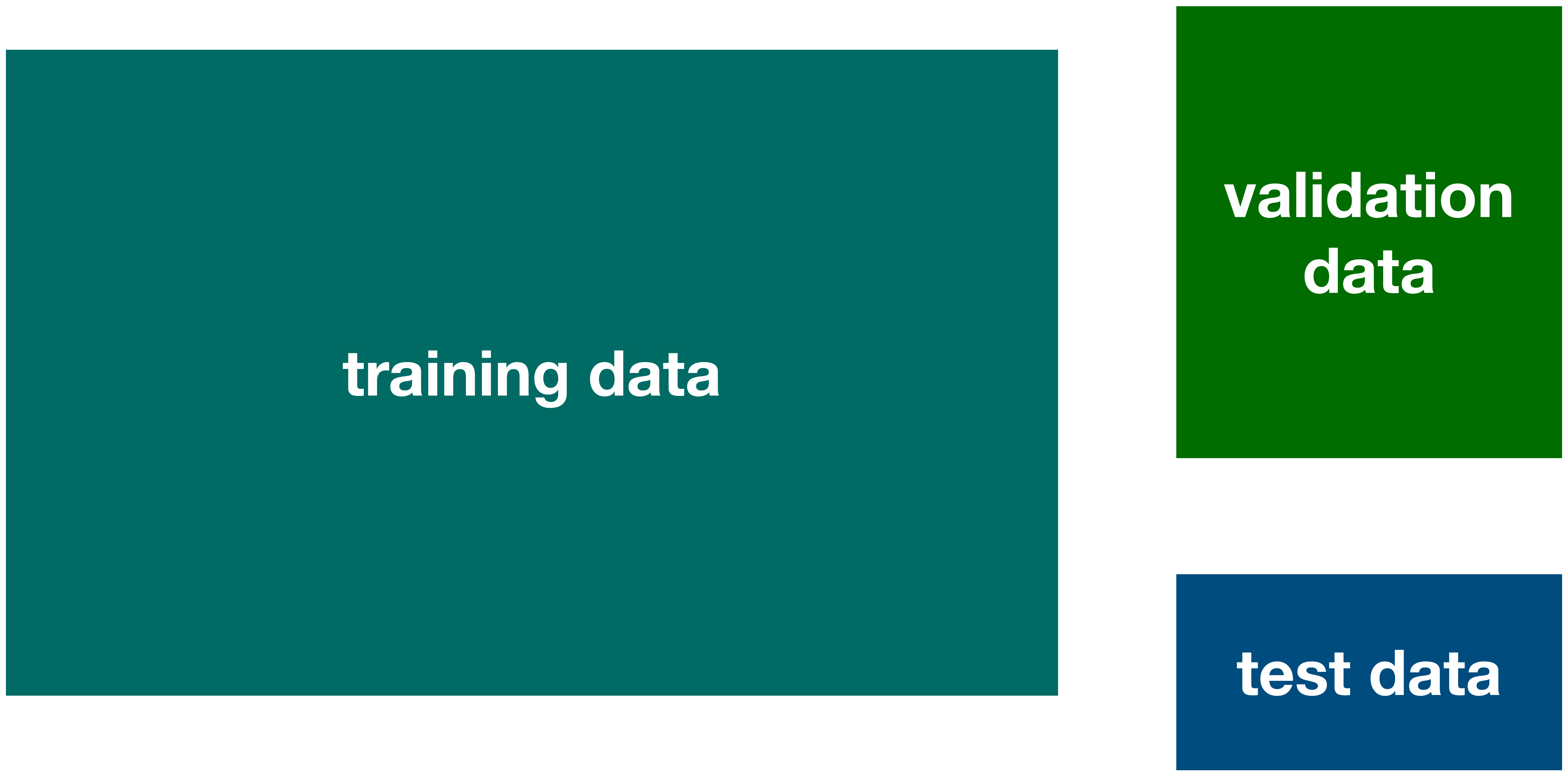
■ **overfitting** frequently takes place when the degree of a regression model is set too high

How do we address under/overfitting?

address overfitting

training data

address overfitting



address overfitting

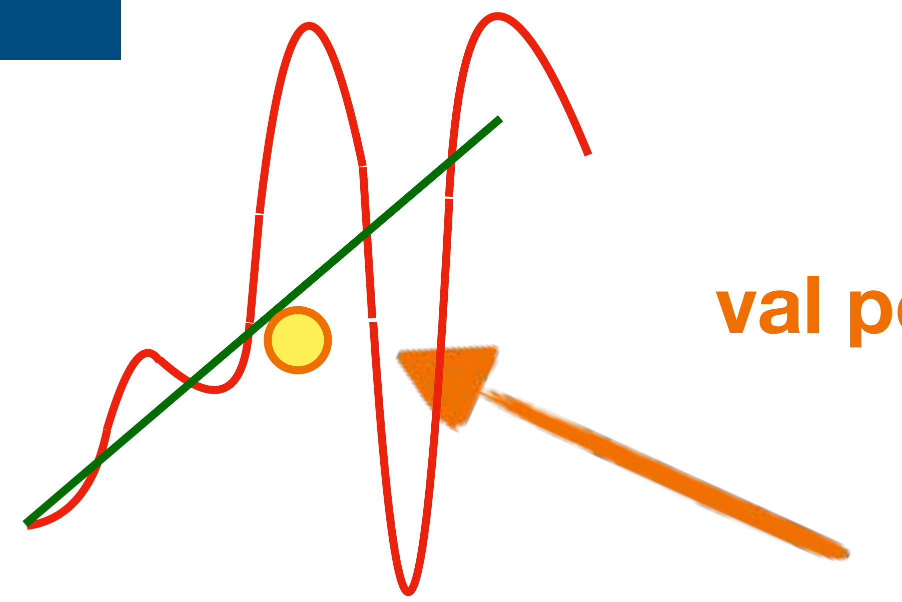
Model Has Seen

training data

Model
Hasn't Seen

validation
data

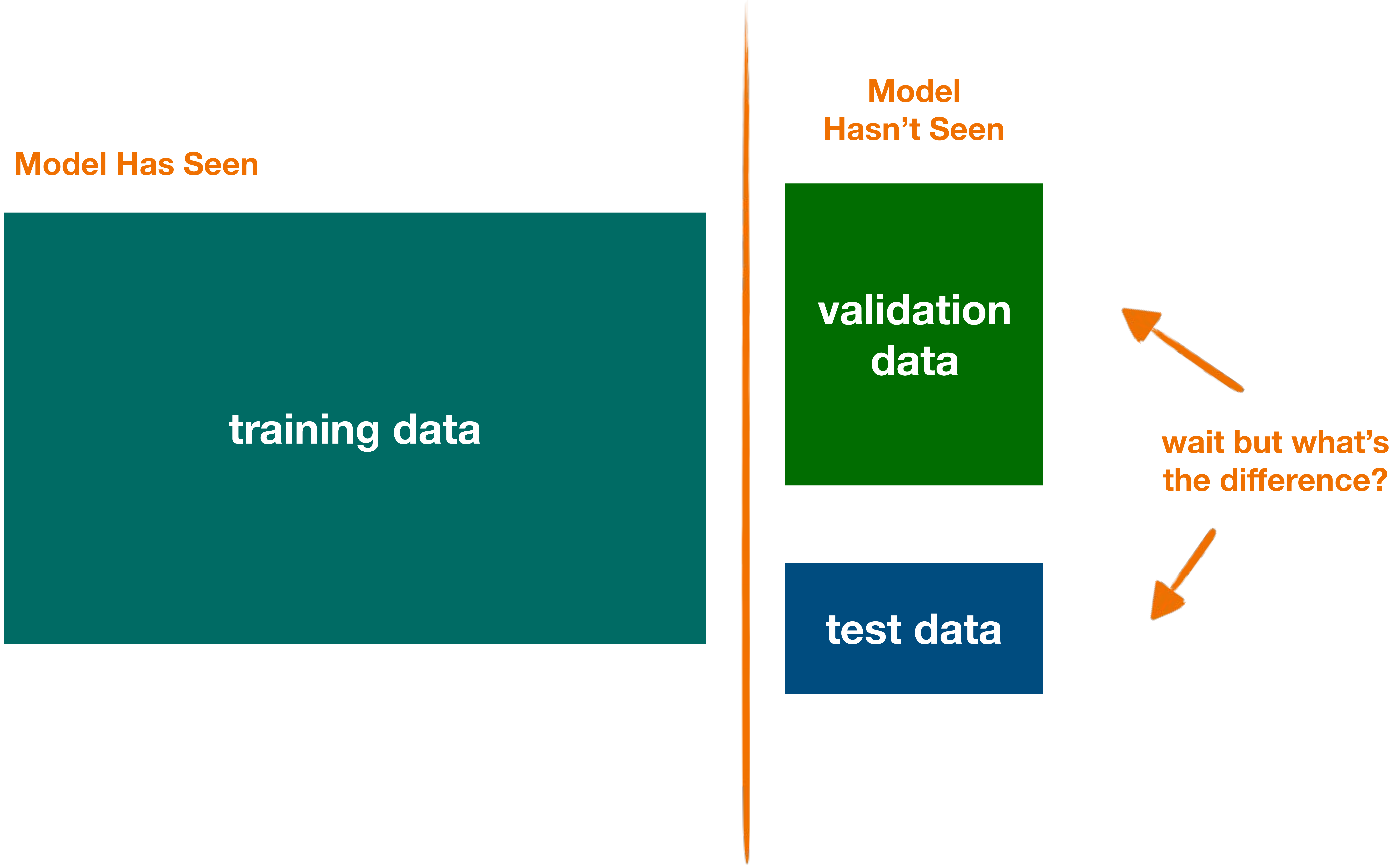
test data



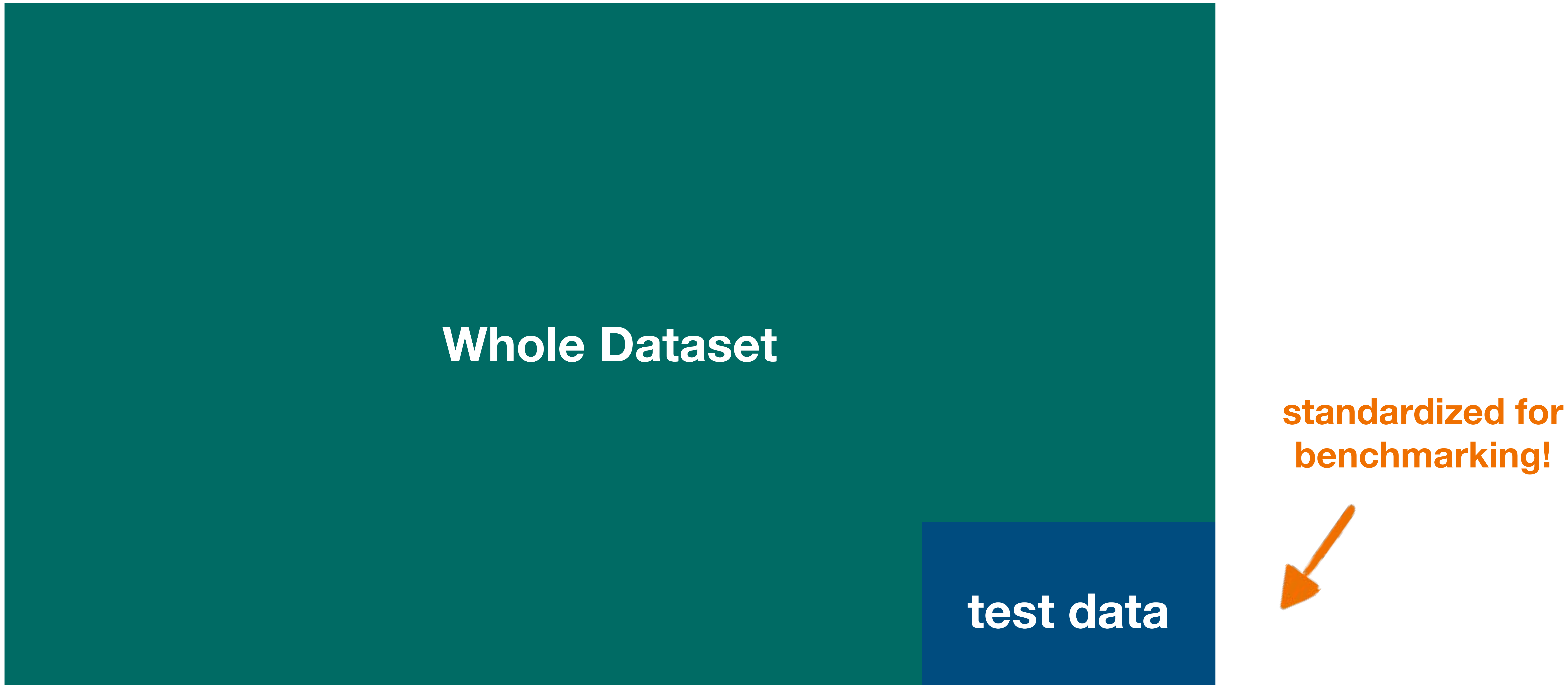
val point(s)!

- we use **validation** and **test** sets, small subsets of data the model hasn't seen before,

address overfitting



address overfitting



- **test sets** are, unlike validation sets, usually set by the data creator as common, unseen benchmark data.

overfitting can be dangerous

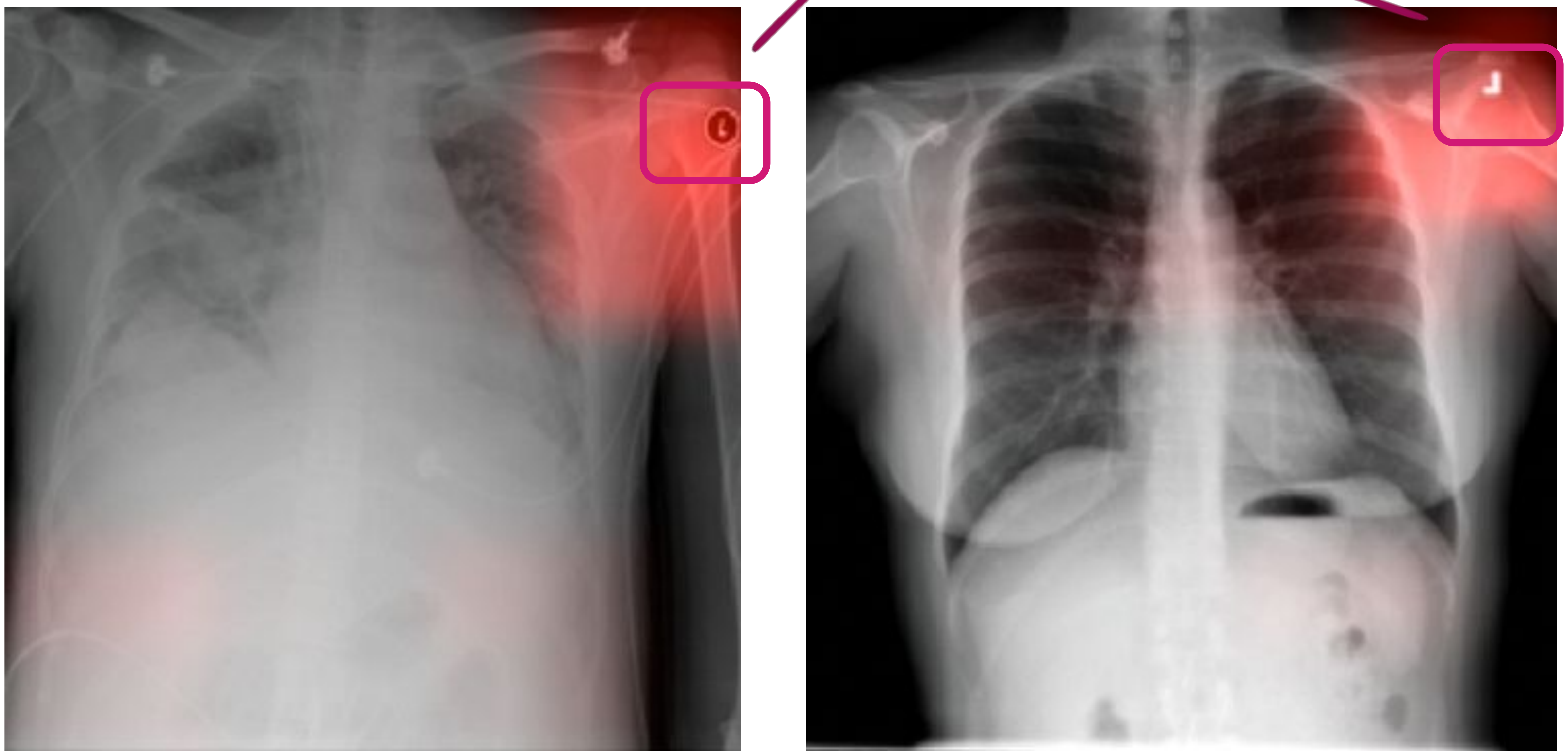
data ethics

data ethics

which one has pneumonia?

data ethics

different hospitals used
different markers!



- models, when not controlled for external factors, often **overfit** on easy targets

Feature selection
& Feature engineering

Motivation

- Performance could degrade when including input variables that are not relevant to the target variable.
- Overfitting for tasks with a smaller # of samples
- A large number of variables can be computationally expensive

Feature Selection

Typical techniques

- Remove features with low variance (e.g., zero variance)
- Remove features with low correlation based on statistical tests
- Sequential feature selection
  - Forward: iteratively add the best new features
  - Backward: iteratively remove the least useful feature
- https://scikit-learn.org/stable/modules/feature_selection.html

Feature Engineering

- Different from feature selection
- Example: predict time-to-sell of a house
- Input (features and label): square footage, lot size, transaction date, built date, and price
- Engineered features could include
  - Cost per sq. ft
  - House age
  - Zip code
  - School rating
- Data preprocessing (e.g., normalization, missing data) sometimes are also considered as feature engineering

Feature Selection

Typical process

- Brainstorming features
- Deciding what features to create
- Creating features
- Testing the impact of the identified features on the task
- Improving your features if needed
- Repeat

Features

- Feature selection
- Feature engineering
- PCA
- Differences

Improving Outcome

Improving Outcome

Debugging a learning algorithm

- A dataset
- Applied a machine learning algorithm
- Got a result, e.g., error rate 11%
- Is this a good result?

Credit: Advanced Machine learning, Andrew Ng, Coursera, for the debugging discussion .

Improving Outcome

Establish a baseline

- What is a reasonable level of error we can hope for?
  - Human level performance
  - Competing/existing algorithms
  - Educated guess based on experience

- Additional baselines
  - Random guess
  - Simple heuristics

Improving Outcome

Bias/variance

	Case 1	Case 2	Case 3
Baseline (e.g., human)	10.6%	10.6%	10.6%
Training error	11%	15.5%	11%
Validation error	16%	16%	12%

Improving Outcome

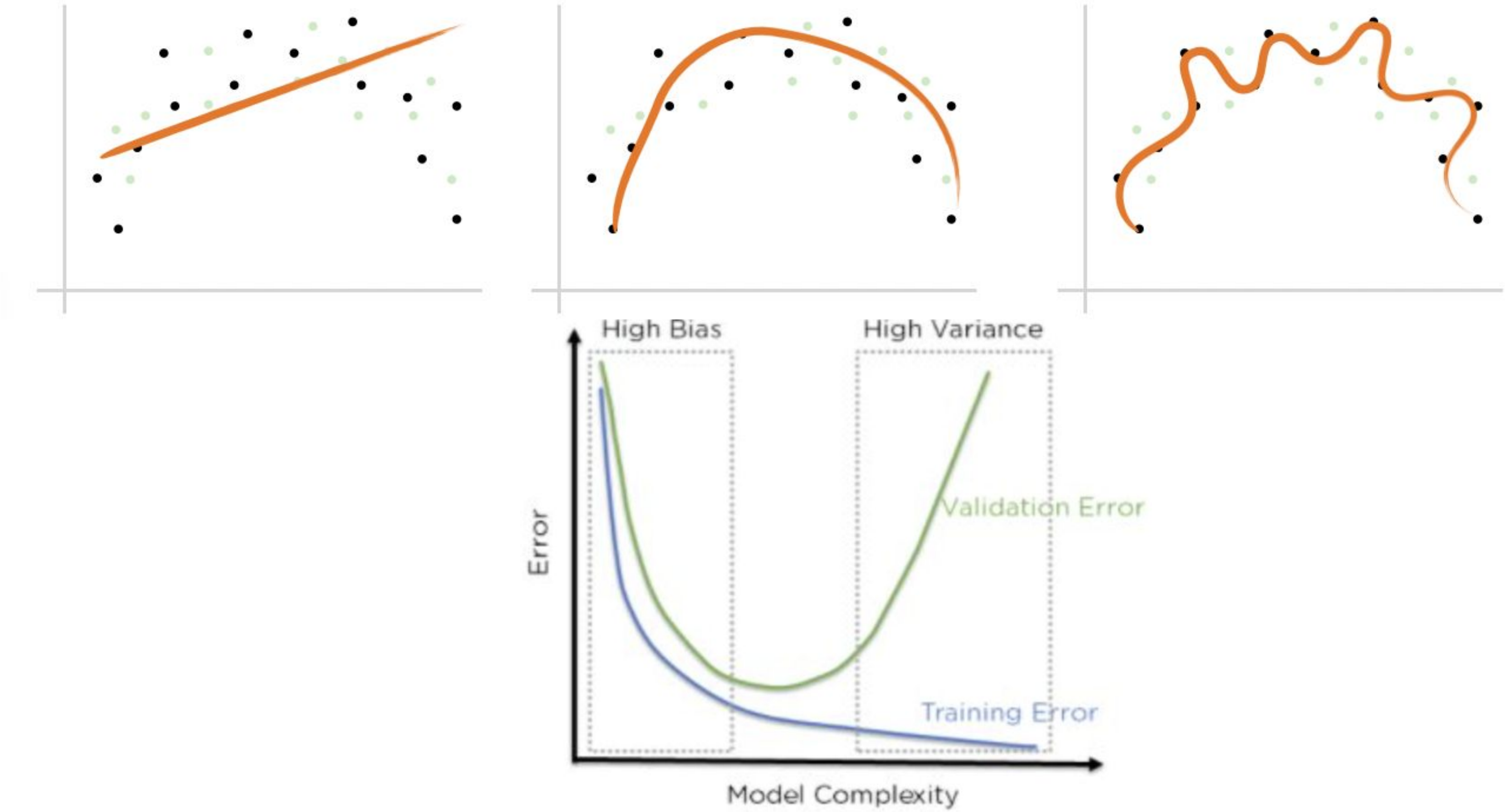
Debugging

- Bias: error from erroneous assumptions in the learning algorithm.
- Variance: error from sensitivity to small fluctuations in the training set.
- Q: how do they manifest?

Improving Outcome

Debugging

- High bias: training error high
- High variance: validation error high



Improving Outcome

Debugging

- High bias: training error high
- High variance: validation error high

- What can we do?

Improving Outcome

Debugging

- High bias: training error high
- High variance: validation error high

- Try getting additional features
- Try adding polynomial features
- Try decreasing regularization or use larger models
- Get more training samples
- Try smaller set of features
- Try increasing regularization or use smaller models

Improving Outcome

Debugging

- Try getting additional features (high bias)
- Try adding polynomial features (high bias)
- Try decreasing regularization or use larger models (high bias)

- Get more training samples (fixes high variance)
- Try smaller set of features (high variance)
- Try increasing regularization or use smaller models (high variance)

Improving Outcome

Error analysis

- Examine where the model went wrong
- Categorize the errors
- Focus on how to fix these errors (or most of them)

Improving Outcome

Example

- Food spoilage prediction
- Manually examine 100 examples where our model got wrong
- Categorize them based on common traits

- Southern CA: 21
- Valley: 10
- Raining weather: 50
- Packaging: 5

- More data and features for SoCal and raining days

A Real Example

- Gait analysis to classify stroke patient in recovery vs. control

When to Use Which Algorithm?

- Start simple
- Try the typical ones
- Sklearn guideline

Potential Pitfalls

Potential Pitfalls

Things that can go wrong

- Inconsistent preprocessing
- Data leakage

- Model is used on test data that has changed
- Selecting appropriate metrics
- Hidden confounders
- Spurious correlations
- Performance on subgroups may be missing
- Data biases

Things that can go wrong

- Inconsistent preprocessing (e.g., different scaling/normalization)
- Data leakage (e.g., temporal or mixing subjects)

- Model is used on test data that has changed
- Selecting appropriate metrics (e.g., is 99% accuracy good enough?)
- Hidden confounders (e.g., golf is correlated with heart attacks)
- Spurious correlations (e.g., hospital ID on images)
- Performance on subgroups may be missing
- Data biases (e.g., AI recruiter)

ML Practices

Be Cautious

- AI/ML is not a cure-all
- “All models are wrong, some are useful.” –George Box
- Understand your models, know the assumptions and limitations of the models
- Is AI a hype or a GE?

Typical steps to apply ML

- Data preprocessing
- Trying different ML algorithms
  - Training set, validation set, test set
- Diagnostics
  - More training samples
  - Increase/decrease feature set
  - Increase/decrease regularization
- Loop back

A ML Project

- Why ML is a suitable approach
  - Do not use ML for the purpose of using ML
  - Evaluate existing approaches and room for improvement
- Problem abstraction and formulation
  - Set appropriate goals
  - Model complexity, data availability, evaluation
  - Domain knowledge critical
- Data collection and data cleaning
  - What, where, and how
- ML algorithms
  - This is often the “easy” part
- Evaluation, sanity check, interpretation
- Iterate the process

Characteristics of Good Problems

- Existing solutions not satisfactory
  - Automate the process
  - Improve performance
- Data availability: suitable data available or obtainable
- Data quality and quantity
- Can evaluate proposed approaches
- Large complex problem beyond white-box modeling
- Understanding complex venue and large data

Project Examples

- Sanitation classification
- Tomato processing loss prediction
- Dietary recommendation
- Help breeders to run more efficient and targeted breeding programs
- Gait analysis
- Biomarker identification
- Disease outbreak prediction
- Weather – outbreak spatial temporal analysis
- Network traffic scheduling
- Encrypted traffic classification
- Traveling salesman