ENCS5341
Machine Learning and Data Science

Model Selection and Regularization

Model Selection

- Model selection is the application of a principled method to determine the complexity of the model, e.g. choosing a subset of predictors, choosing the degree of the polynomial model etc.

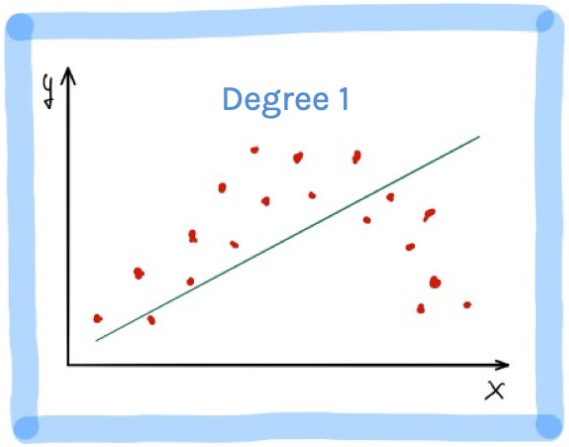
- A strong motivation for performing model selection is to avoid overfitting, which can happen when:

  - there are too many predictors:
    - the feature space has high dimensionality
    - the polynomial degree is too high

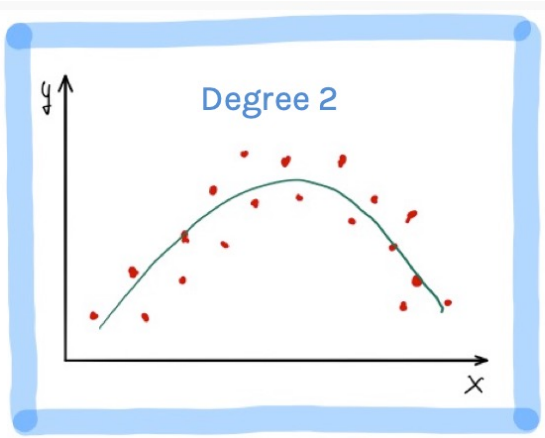
  - the coefficients values are too extreme

Polynomial Regression Example

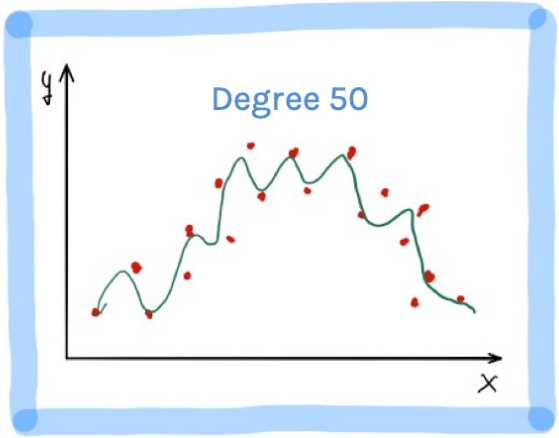
- Fitting a polynomial model requires choosing a degree.



Underfitting: when the degree is too low, the model cannot fit the trend.



We want a model that fits the trend and ignores the noise.

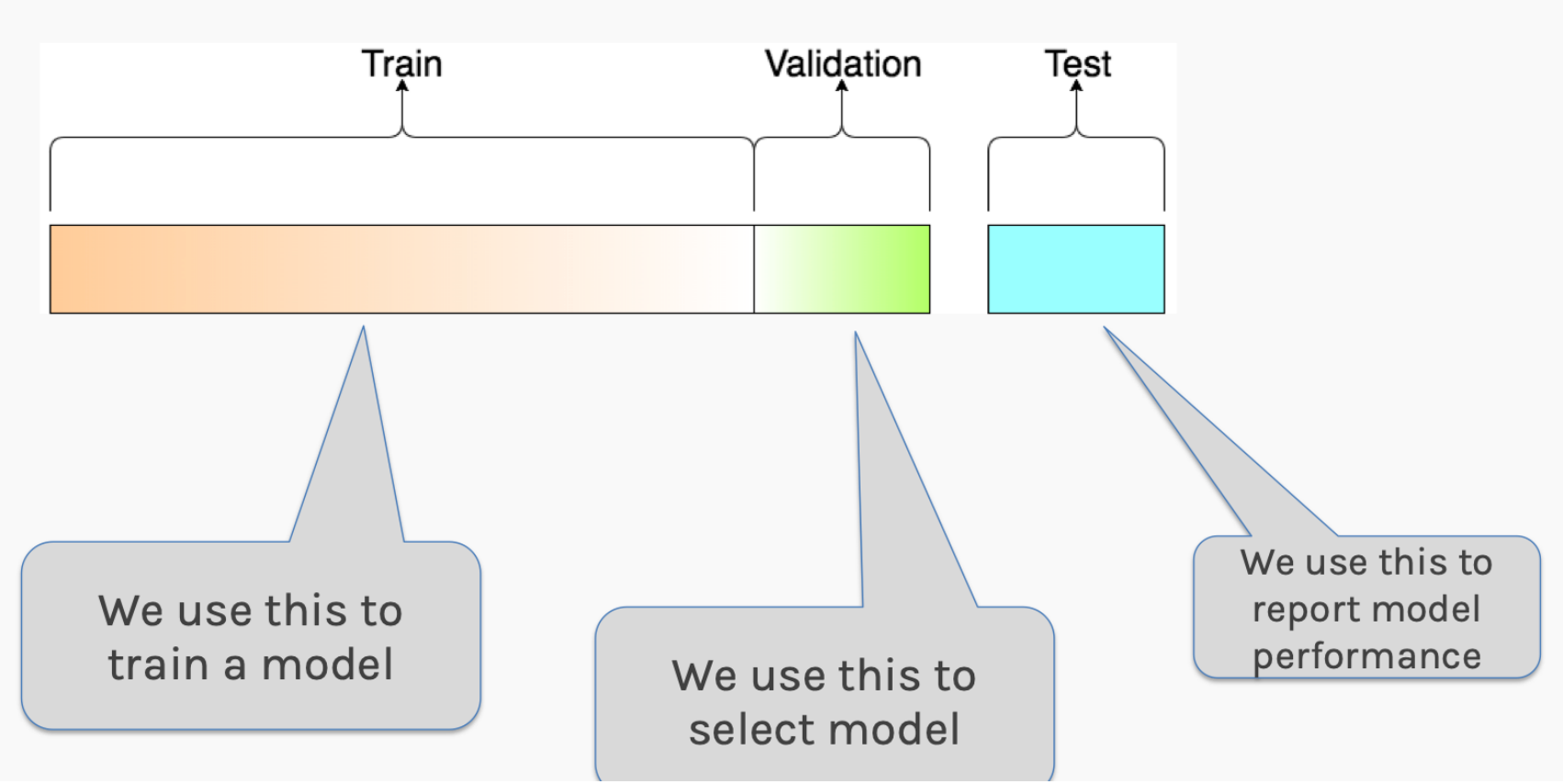


Overfitting: when the degree is too high, the model fits all the noisy data points.

Generalization Error

- We know to evaluate the model on both train and test data, because models that do well on training data may do poorly on new data (overfitting).
- The ability of models to do well on new data is called generalization.
- The goal of model selection is to choose the model that generalizes the best.
- Always evaluate models as they are predicting future data.
    - If the data is seen during training, we cannot use it for evaluation.

Train-Validation-Test



Model Selection

- Question: How many different models when considering **d** predictors (only linear terms) do we have?

Example: 3 predictors (X_1, X_2, X_3)

- Models with 0 predictor:
  - M0:
- Models with 1 predictor:
  - M1: X_1
  - M2: X_2
  - M3: X_3
- Models with 2 predictors:
  - M4: $\{X_1, X_2\}$
  - M5: $\{X_2, X_3\}$
  - M6: $\{X_3, X_1\}$
- Models with 3 predictors:
  - M7: $\{X_1, X_2, X_3\}$

2^d models

Stepwise Variable Selection and Validation

Selecting optimal subsets of predictors (including choosing the degree of polynomial models) through:

- Stepwise variable selection - iteratively building an optimal subset of predictors by optimizing a fixed model evaluation metric each time.

- Validation – selecting an optimal model by evaluating each model on validation set.

Stepwise Variable Selection: Forward method

In forward selection, we find an 'optimal' set of predictors by iterative building up our set.

- Start with the empty set P_0, construct the null model M_0.
- For k = 1, ... , d:
  - Let M_{k-1} be the model constructed from the best set of $k-1$ predictors, P_{k-1}.
  - Select the predictor $X_{n,k}$, not in P_{k-1}, so that the model constructed from $P_k = X_{n,k} \cup P_{k-1}$ optimizes a fixed metric.
  - Let M_k denote the model constructed from the optimal P_k.
- Select the model M amongst $\{M_0, M_1, \ldots, M_d\}$ that optimizes a fixed metric (this can be validation MSE, R^2, … etc.)

Stepwise Variable Selection Computational Complexity

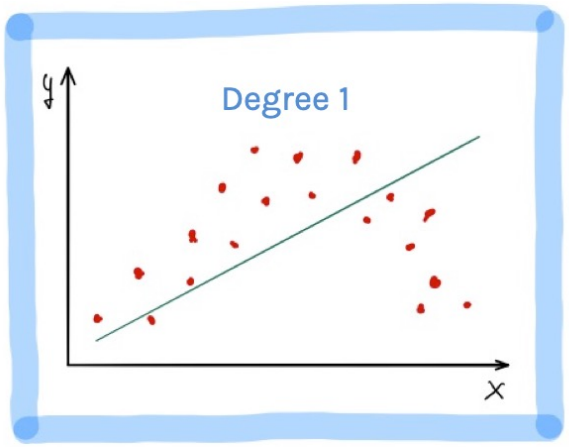
How many models did we evaluate?

- 1st step, **d Models**
- 2nd step, **d-1 Models** (add 1 predictor out of d-1 possible)
- 3rd step, **d-2 Models** (add 1 predictor out of d-2 possible)
- …

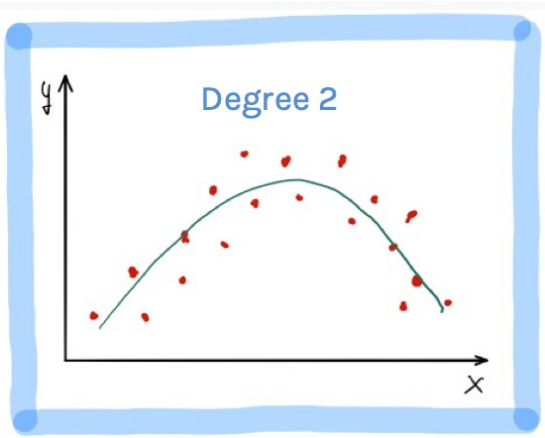
$O(d^2) << 2^d$ for large d

Choosing the degree of the polynomial model

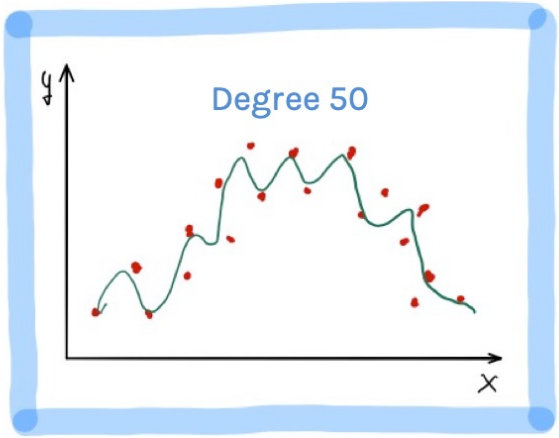
- Fitting a polynomial model requires choosing a degree.



Underfitting: when the degree is too low, the model cannot fit the trend.



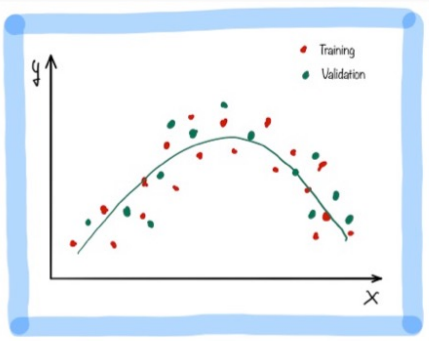
We want a model that fits the trend and ignores the noise.



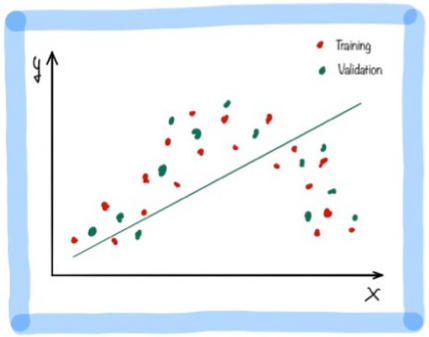
Overfitting: when the degree is too high, the model fits all the noisy data points.

Choosing the degree of the polynomial model

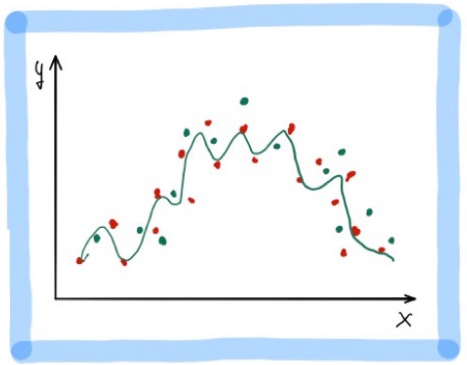
Best model: validation error is minimum.

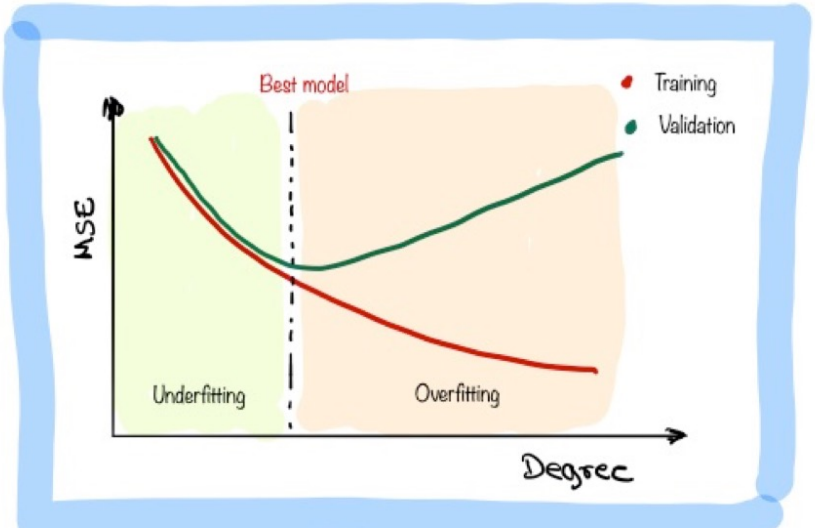


Underfitting: train and validation error is high.



Overfitting: train error is low, validation error is high.

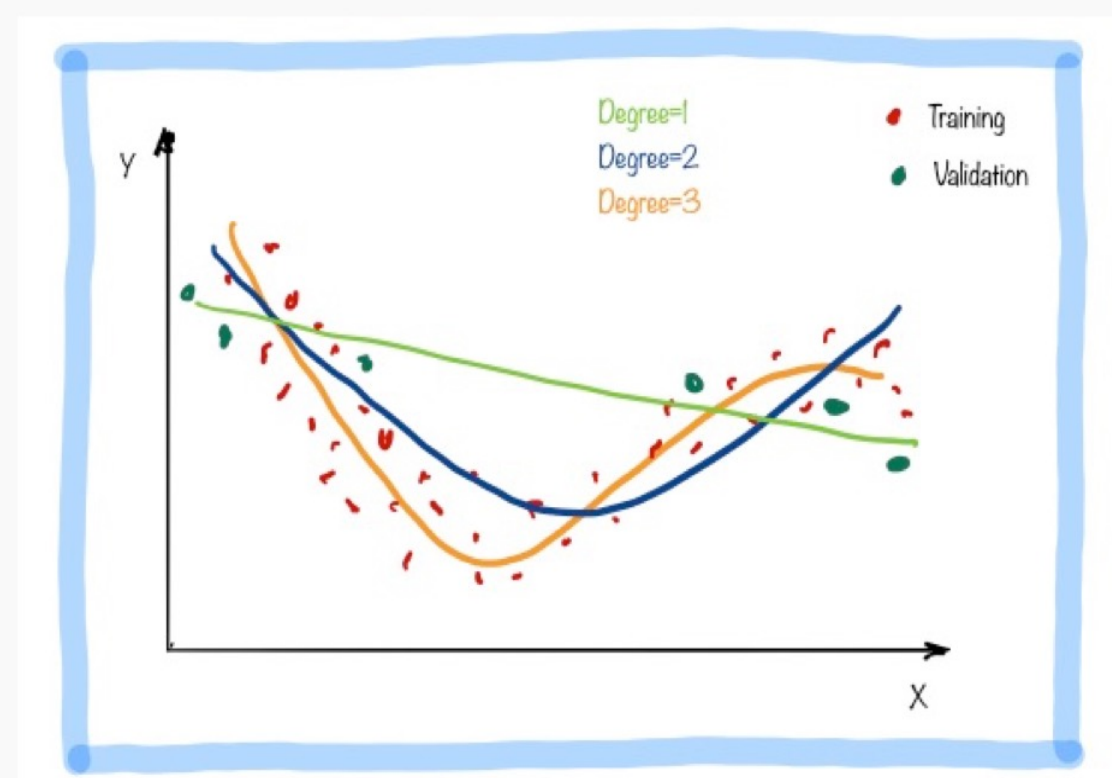




Cross Validation: Motivation

- Using a single validation set to select amongst multiple models can be problematic - **there is the possibility of overfitting to the validation set**.

- Example: It is obvious that degree=3 is the correct model but the validation set by chance favors the linear model.



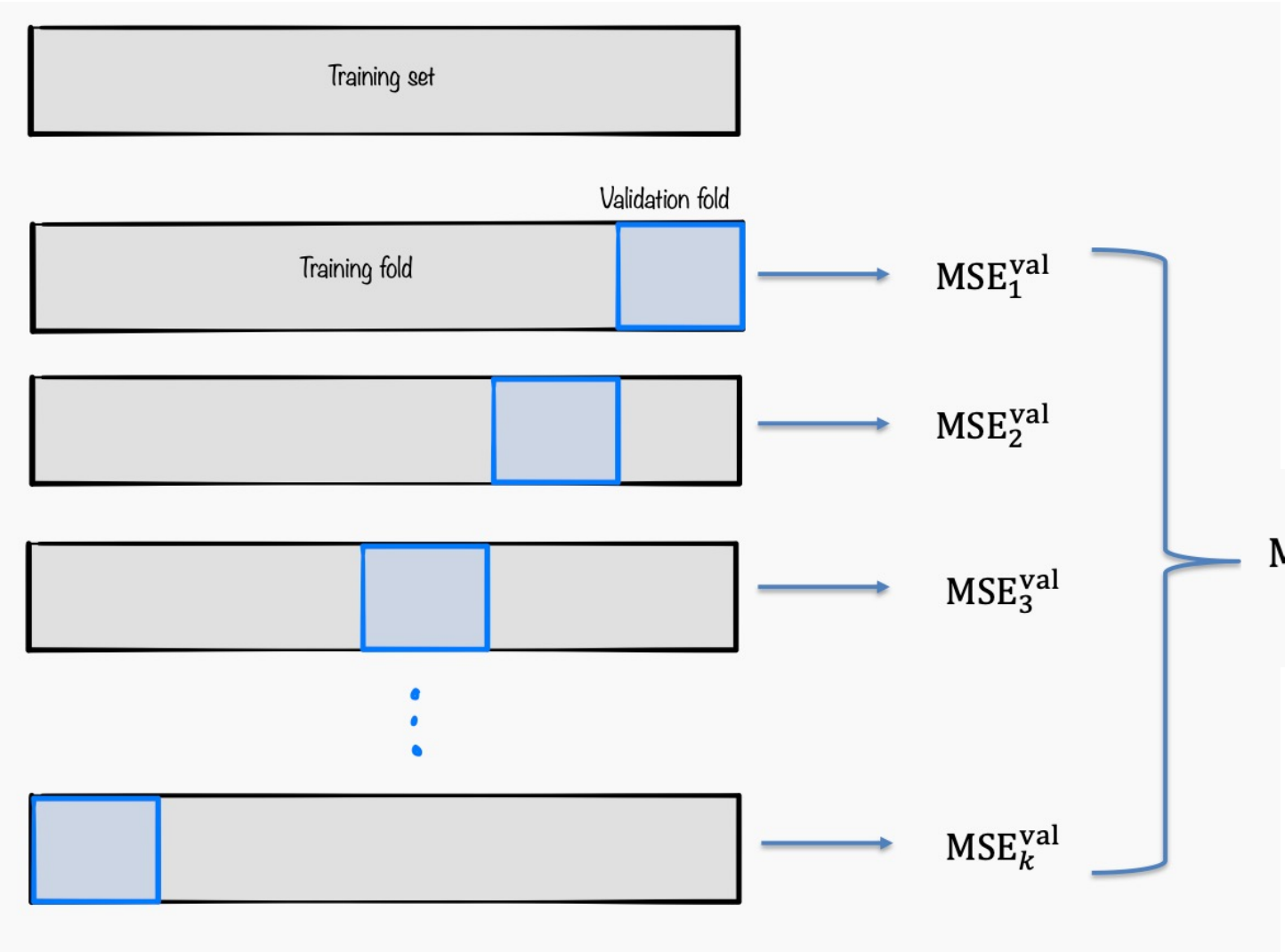
Cross Validation: Motivation

- Using a single validation set to select amongst multiple models can be problematic - **there is the possibility of overfitting to the validation set**.

- One solution to the problems raised by using a single validation set is to evaluate each model on multiple validation sets and average the validation performance.

- One can randomly split the training set into training and validation multiple times but randomly creating these sets can create the scenario where important features of the data never appear in our random draws.

Cross Validation



K-Fold Cross Validation

- Given a data set $\{X_1, \dots, X_n\}$, where each $\{X_1, \dots, X_n\}$ contains J features.

- To ensure that every observation in the dataset is included in at least one training set and at least one validation set we use the **K-fold validation**:

  - split the data into K uniformly sized chunks, $\{C1, \dots, C_K\}$

  - we create K number of training/validation splits, using one of the K chunks for validation and the rest for training.

- We fit the model on each training set, denoted $\hat{f}c_{-i}$, and evaluate it on the corresponding validation set, $\hat{f}c_{-i}(C_i)$. The **cross validation is the performanc**e of the model averaged across all validation sets:

$$CV(\text{Model}) = \frac{1}{K}\sum_{i=1}^{K} L(\hat{f}_{C_{-i}}(C_i))$$

where L is a loss function.

Leave-One-Out

- Or using the **leave one out** method:
  - validation set: $\{X_i\}$
  - training set: $X_{-i} = \{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$
- for $i = 1, \dots, n$:
  - We fit the model on each training set, denoted $\hat{f}_{X_{-i}}$, and evaluate it on the corresponding validation set, $\hat{f}_{X_{-i}}(x_i)$.
- The **cross validation score** is the performance of the model averaged across all validation sets

$$CV(\text{Model}) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{f}_{X_{-i}}(X_i))$$

where L is a loss function.

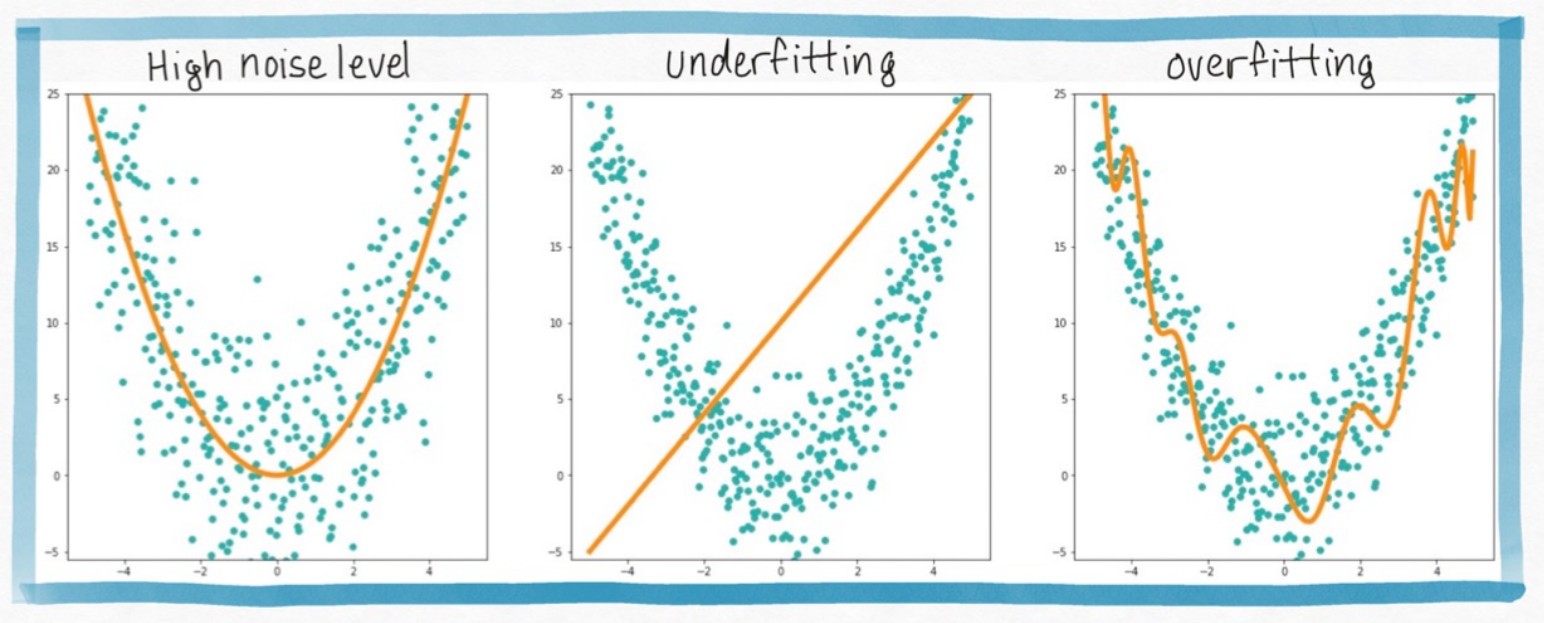
Regularization error, bias vs variance

Test Error and Generalization

- We know to evaluate models on both train and test data because models can do well on training data but do poorly on new data.

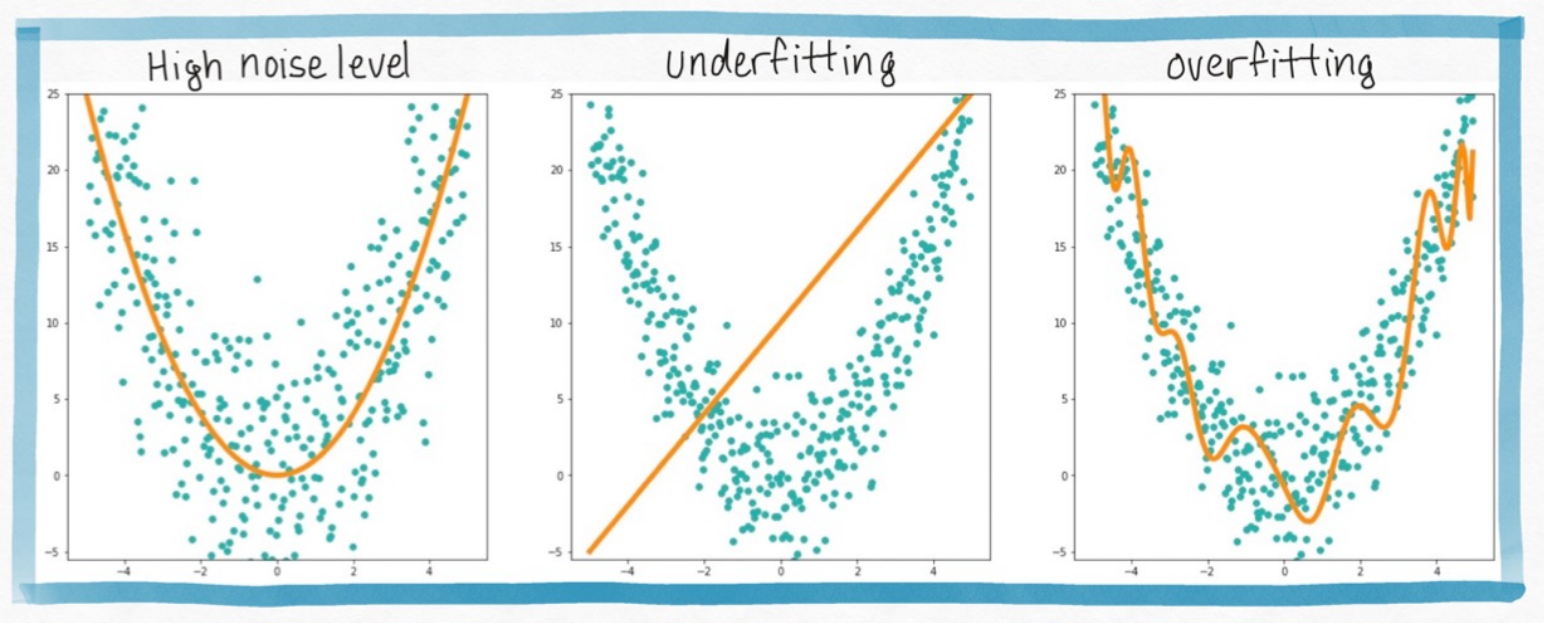
- When models do well on new data is called generalization.

- There are at least three ways a model can have a high test error.



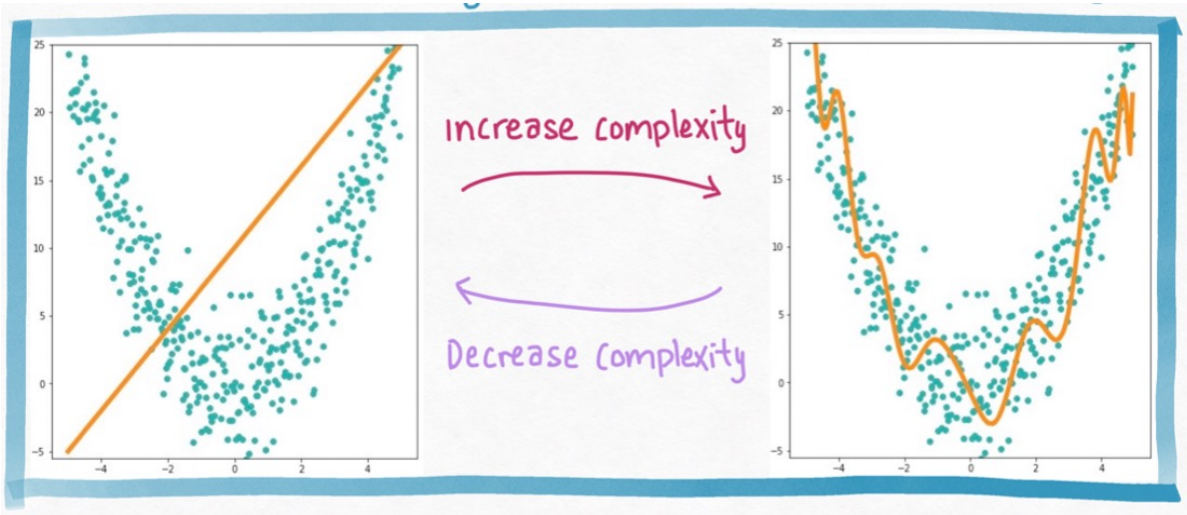
Irreducible and Reducible Errors

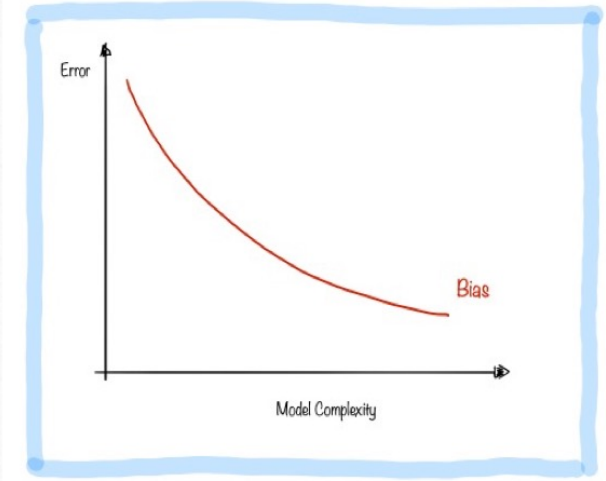
- We distinguished the contributions of noise to the generalization error:
- Irreducible error: we can't do anything to decrease error due to noise.
- Reducible error: we can decrease error due to overfitting and underfitting by improving the model.



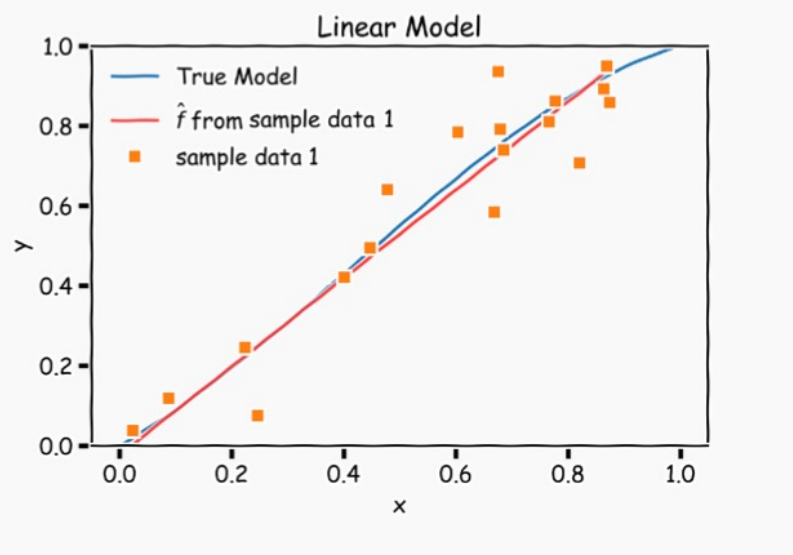
The Bias-Variance: Bias

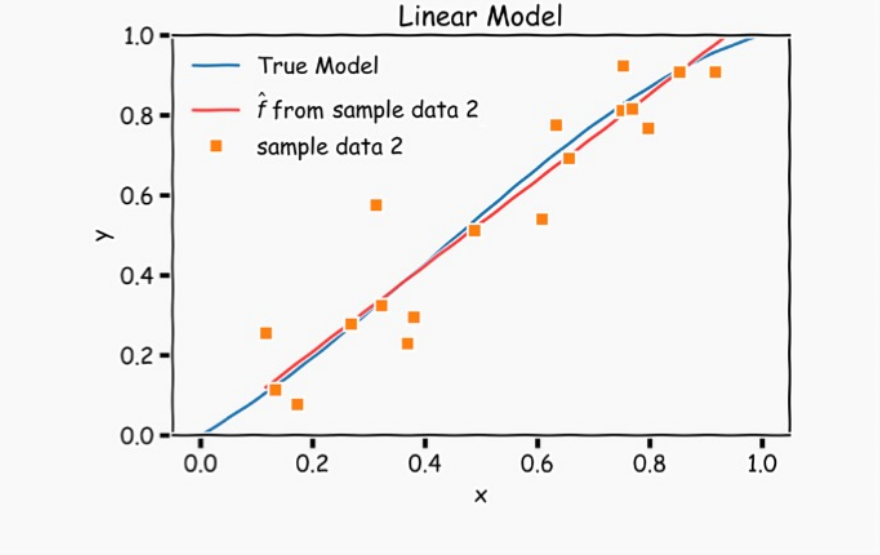
- Reducible error comes from either underfitting or overfitting. There is a trade-off between the two sources of errors:

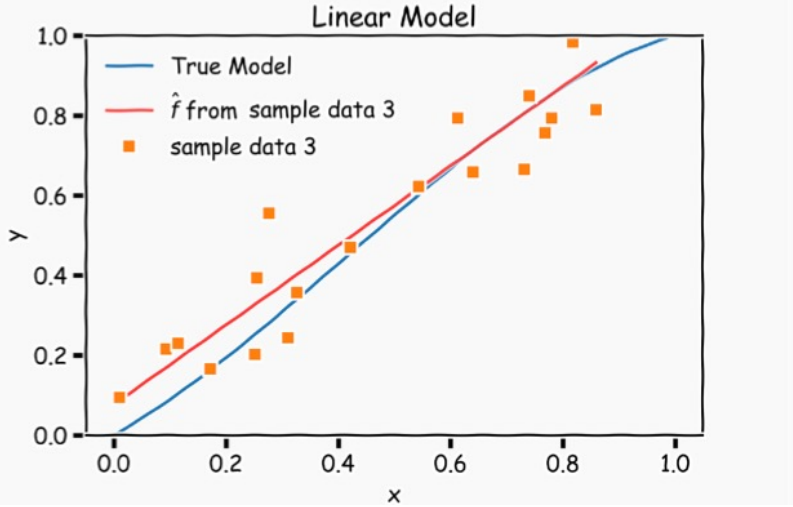


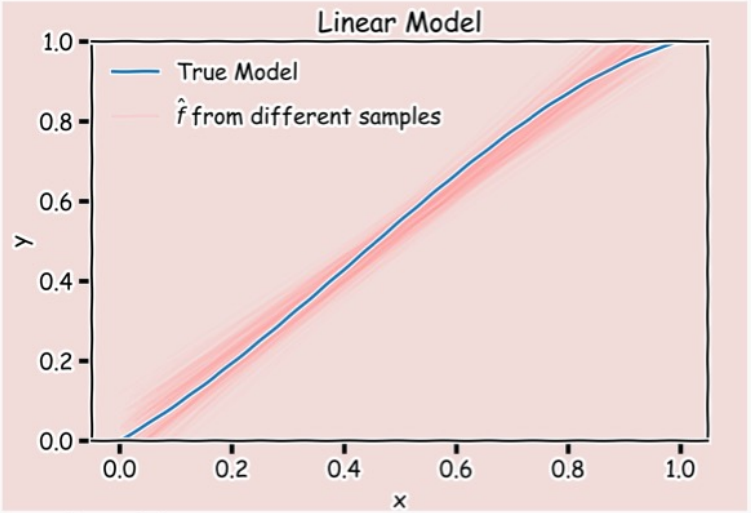


Bias vs Variance: Variance

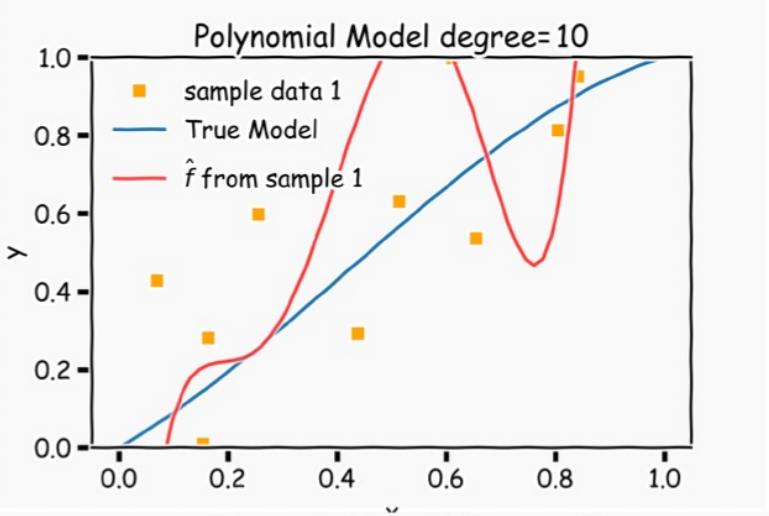


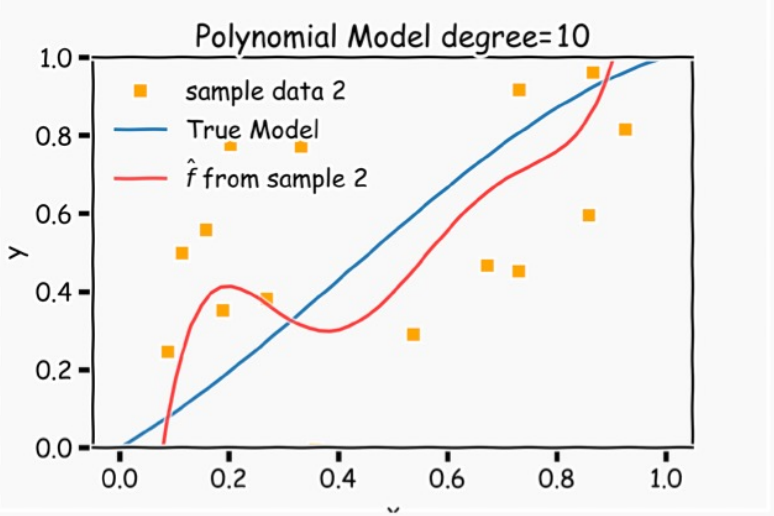


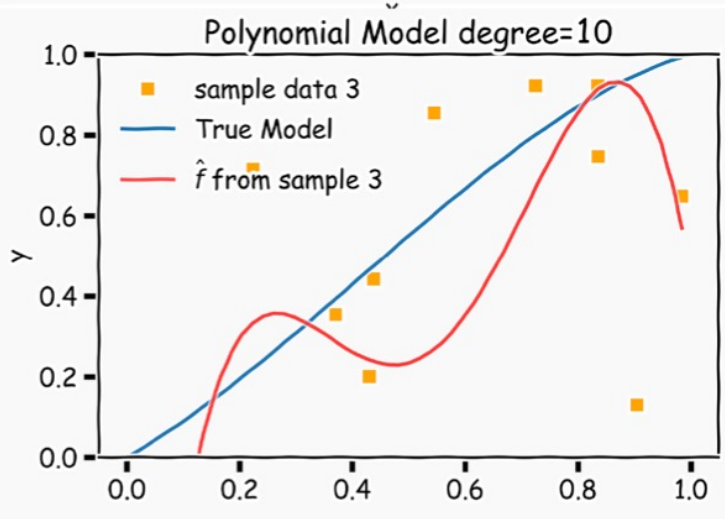


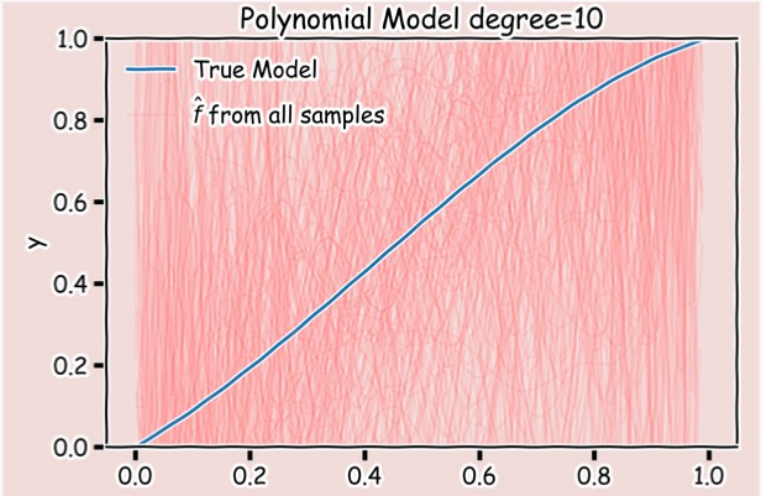


Bias vs Variance





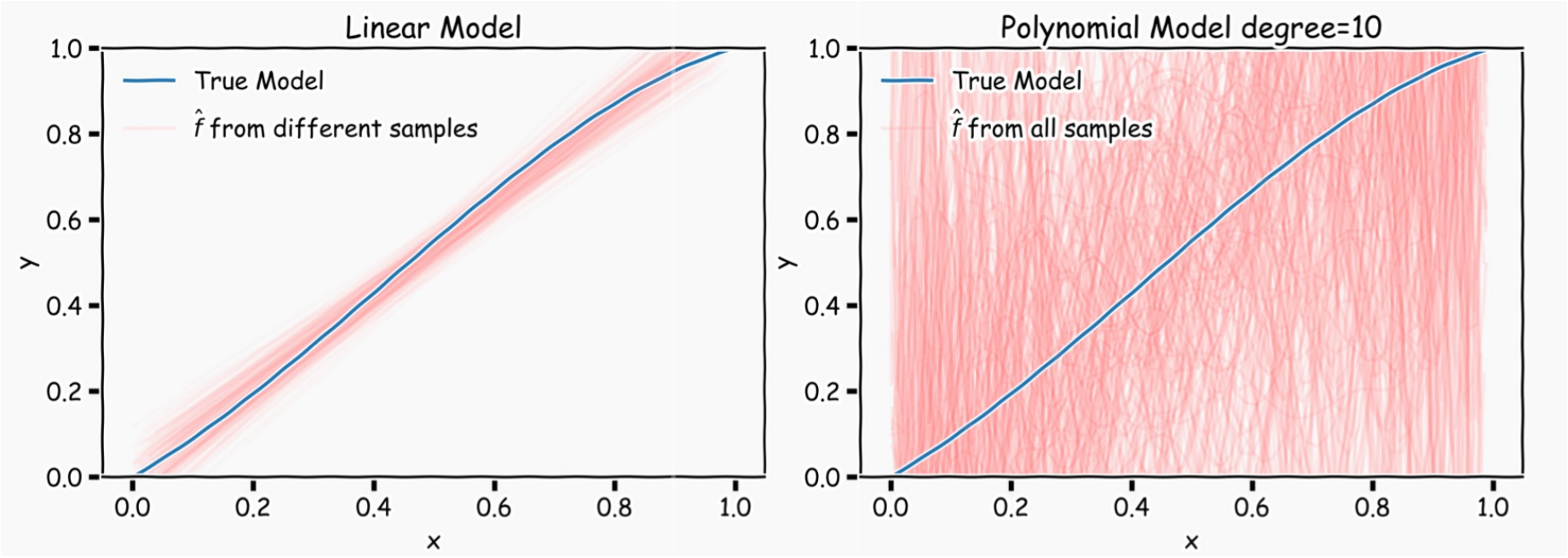




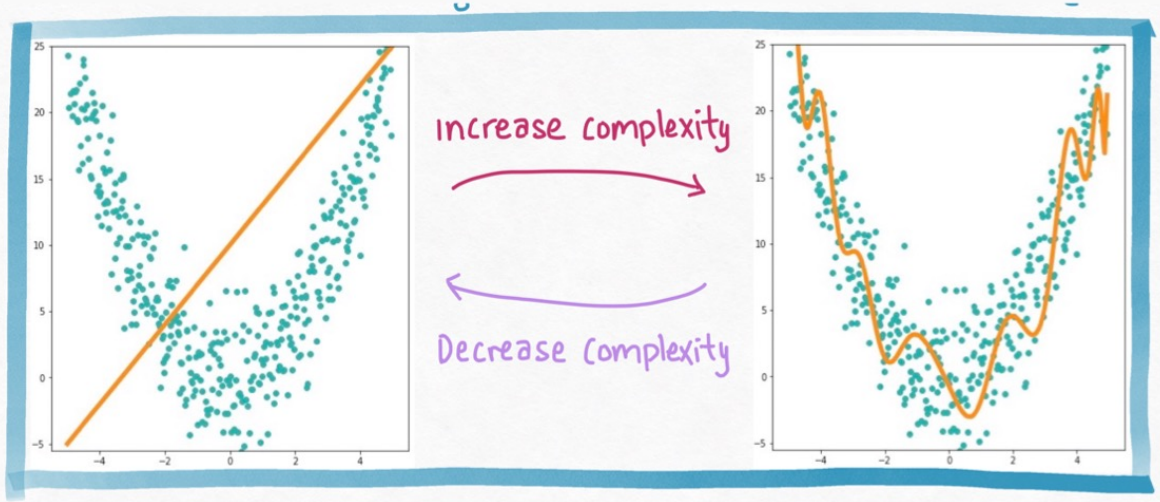
Bias vs Variance

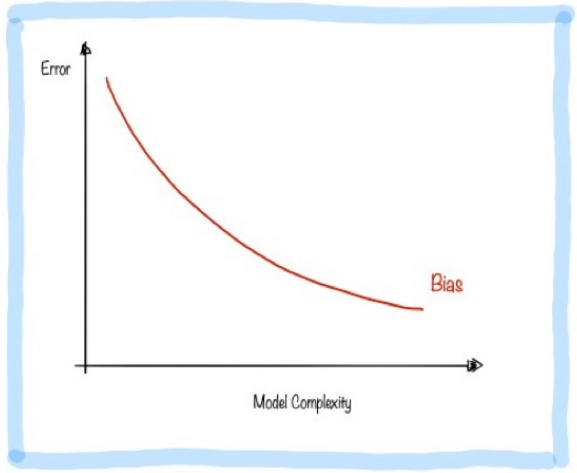
- **Left**: 2000 best fit straight lines, each fitted on a different 20-points training set.

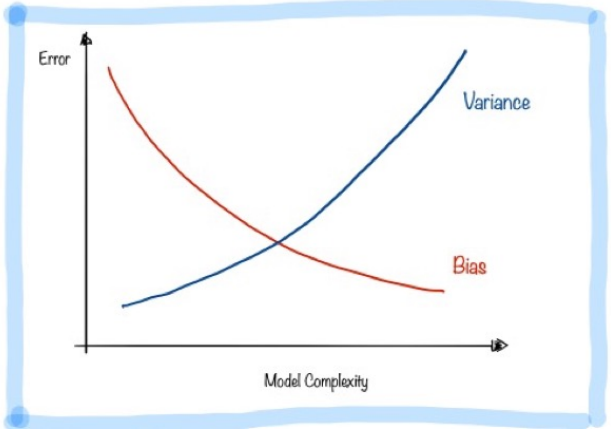
- **Right**: Best-fit models using degree 10 polynomials



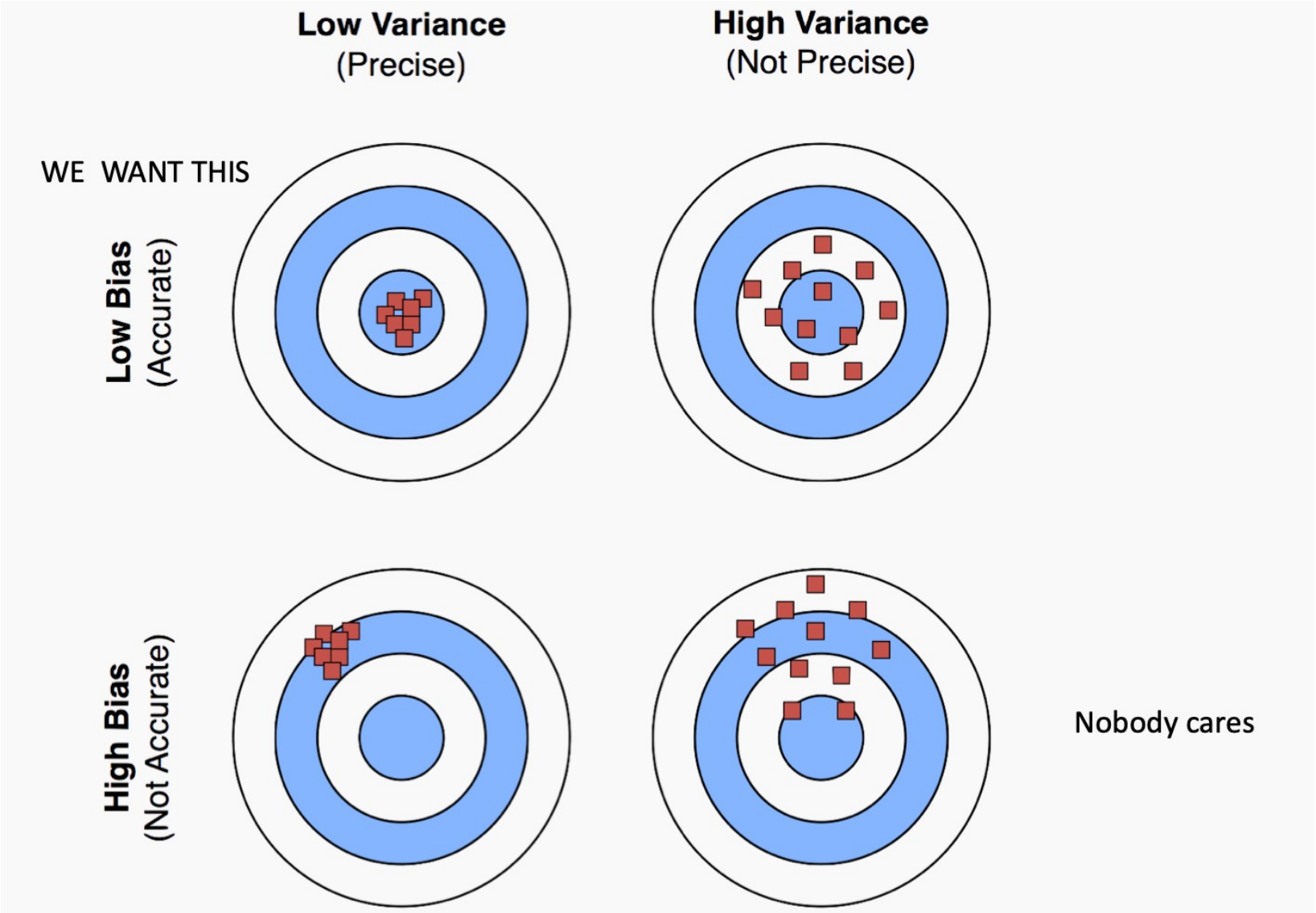
The Bias-Variance Trade Off







The Bias-Variance Trade Off



Overfitting

- Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

- Overfitting we can happen when:
  - there are too many predictors:
    - the feature space has high dimensionality
    - the polynomial degree is too high
  - the coefficients values are too extreme

- Model selection can be used the avoid the first case of overfitting

- For the 2nd case, we use another way of avoiding overfitting: Regularization

Regularization: An Overview

- The idea of regularization revolves around modifying the loss function L; in particular, we add a regularization term that penalizes some specified properties of the model parameters

$$L_{reg} = L(\boldsymbol{w}) + \lambda R(\boldsymbol{w})$$

where λ is a scalar that gives the weight (or importance) of the regularization term.

- Fitting the model using the modified loss function L_{reg} would result in model parameters with desirable properties (specified by R).

LASSO Regression

- Since we wish to discourage extreme values in model parameter, we need to choose a regularization term that penalizes parameter magnitudes. For our loss function, we will again use MSE.

- Together our regularized loss function is:

$$L_{LASSO}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}(\, y_i - \boldsymbol{w}^T\boldsymbol{x}_i\,)^2 + \lambda \sum_{j=1}^{d}|w_j|$$

- Note that $\sum_{j=1}^{d}|w_j|$ is the l_1 norm of the vector $\boldsymbol{w}$

$$\sum_{j=1}^{d}|w_j| = \|\boldsymbol{w}\|_1$$

Ridge Regression

- Alternatively, we can choose a regularization term that penalizes the squares of the parameter magnitudes. Then, our regularized loss function is:

$$L_{Ridge}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}(\, y_i - \boldsymbol{w}^T\boldsymbol{x}_i\,)^2 + \lambda \sum_{j=1}^{d} w_j^2$$

- Note that $\sum_{j=1}^{d} w_j^2$ is the square of the l_2 norm of the vector $\boldsymbol{w}$

$$\sum_{j=1}^{d} w_j^2 = \|\boldsymbol{w}\|_2^2$$

Ridge, LASSO - Computational complexity

- Solution to ridge regression:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_d)^{-1} \mathbf{X}^T \mathbf{y}$$

- LASSO has no conventional analytical solution, as the L1 norm has no derivative at zero. We can, however, use the concept of subdifferential or subgradient to find a manageable expression.

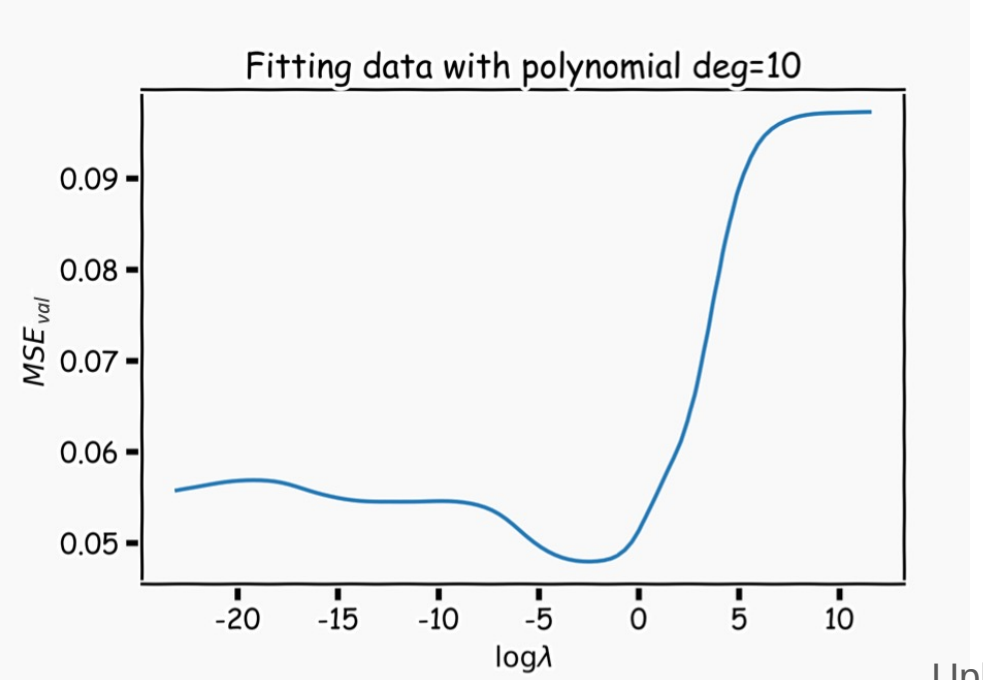
Choosing λ

- In both ridge and LASSO regression, we see that the larger our choice of the regularization parameter λ, the more heavily we penalize large values in w.
- If λ is close to zero, we recover the MSE, i.e. ridge and LASSO regression is just ordinary regression.
- If λ is sufficiently large, the MSE term in the regularized loss function will be insignificant and the regularization term will force w_{ridge} and w_{LASSO} to be close to zero.
- To avoid ad-hoc choices, we should select λ using validation or better cross-validation.

Regularization Parameter with a Validation Set

The solution of the Ridge/Lasso regression involves three steps:

- Select λ
- Find the minimum of the ridge/Lasso regression loss function and record the MSE on **the validation set**.
- Find the λ that gives the smallest MSE on the validation set.



The Geometry of Regularization