Introduction to Machine Learning

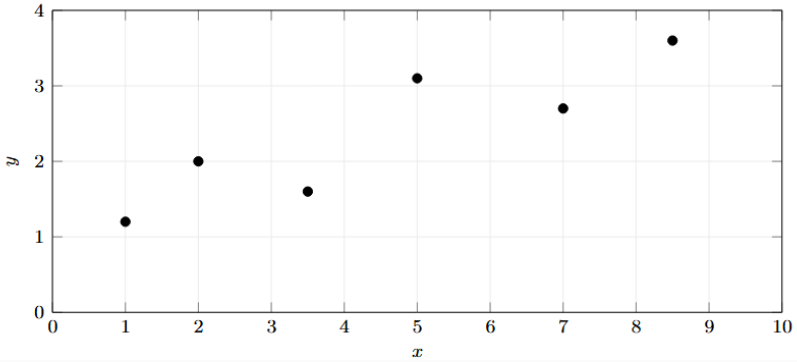
Lecture 8:
Supervised Learning: Hypothesis Class & Bias-Variance Trade-off

Prof. Subrahmanya Swamy Peruru
Department of Electrical Engineering
IIT Kanpur

Supervised Learning

Given: Labelled training data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

Goal: Learn $\hat{f}: \mathcal{X} \rightarrow \mathcal{Y}$



Data with Single feature x

Training Loss: $\underset{f}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2$ (what is the best function that minimizes this objective?)

A best possible solution:

Achieves the best possible training error

$$\hat{f}(\mathbf{x}_i) = y_i \ \forall i, \qquad \hat{f}(\mathbf{x}) = 0 \text{ for all other } \mathbf{x}.$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}(\mathbf{x}_i) - y_i\right)^2 = 0$$

Why did we end up getting such a bad function as the best possible solution of our optimization problem?

Since we did not restrict the function space!

Inductive Bias: Assumptions on Function Structure

No restriction on function structure
⇒ Memorization of training data

Learning is about generalization: How accurately can we predict target y at *unseen* inputs $\mathbf{x}$.

Impose assumptions on function structure (Inductive Bias)
⇒ Generalization on unseen data

These assumptions are called inductive bias
They may come from domain knowledge or mathematical principles (e.g., smoothness).

The assumptions are encoded into a **hypothesis class** $\mathcal{H}$

$$\operatorname{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

Example 1: Local Similarity

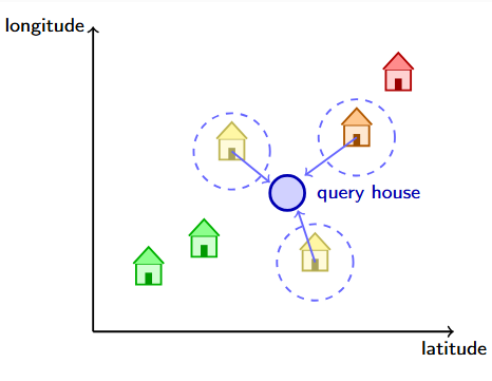
Problem: House price prediction based on location

Assumption: Nearby houses have similar prices

$$\mathbf{x} \approx \mathbf{x}_i \quad \Rightarrow \quad y \text{ likely close to } y_i$$

The output at $\mathbf{x}$ can be inferred from outputs of nearby inputs.

Consequence: Predict using nearby points ⇒ **nearest-neighbor methods** (**K-NN algorithm**)

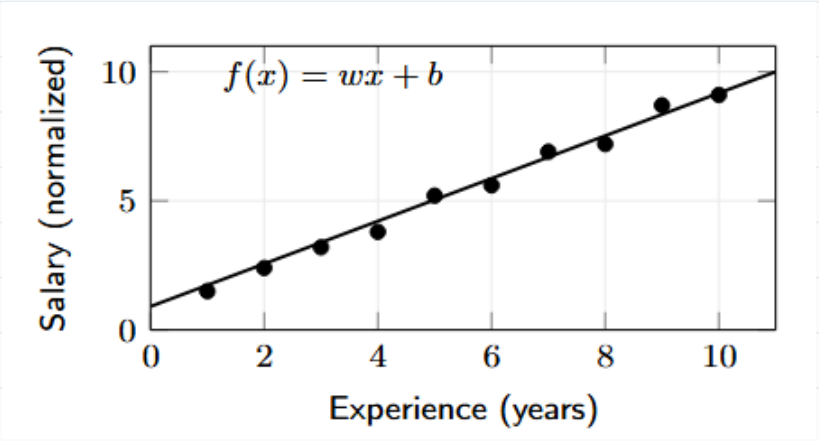


Example 2: Global Linearity

Problem: Salary prediction based on experience

Assumption: the relationship is approximately linear (globally).

$$f(x) = wx + b$$



Estimating Generalization Using Test Data

We care about how well a learned model f performs on *unseen data* (**generalization performance**).

We approximate generalization performance by using a **test set** (held out, not used for training).

$$\mathcal{S}_{\text{test}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$$

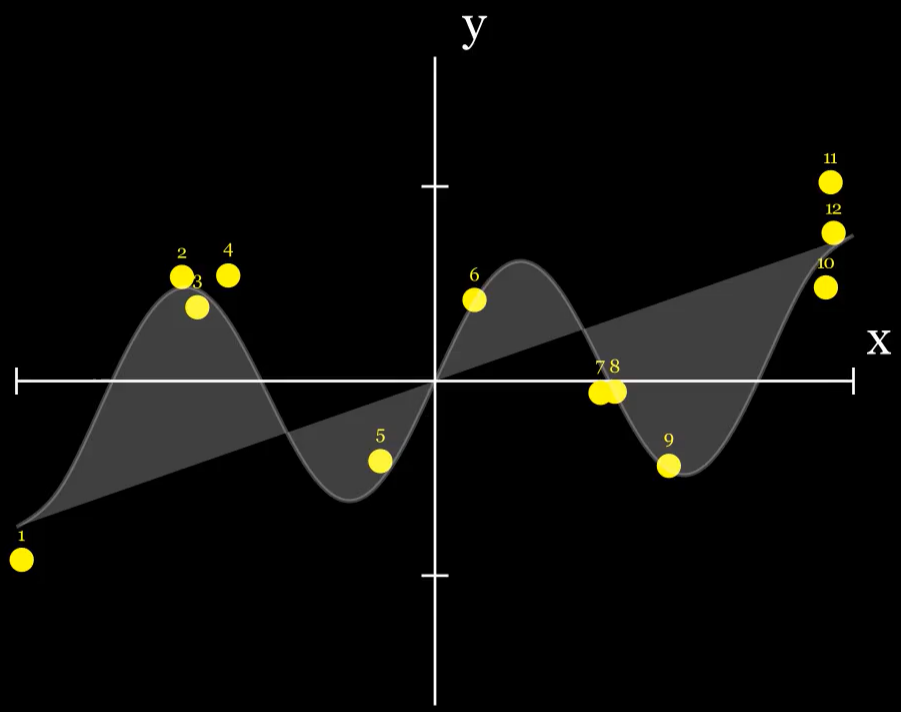
The test data is never touched during the training process:

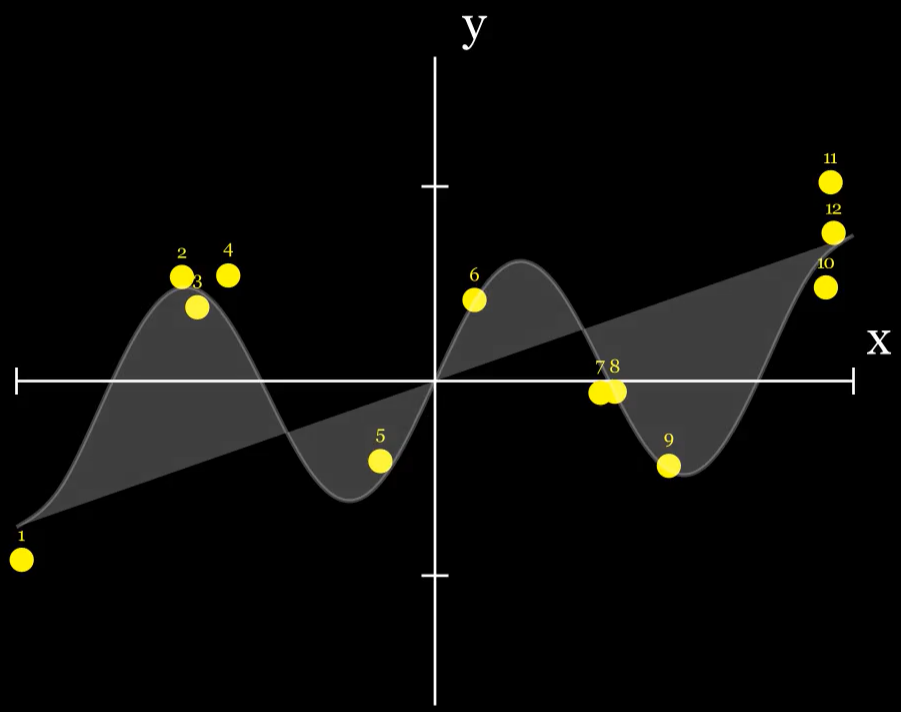
- It is **not** used to fit the model
- It is **not** used to decide the hypothesis class / flexibility
- It simply serves as a proxy for unseen data during evaluation

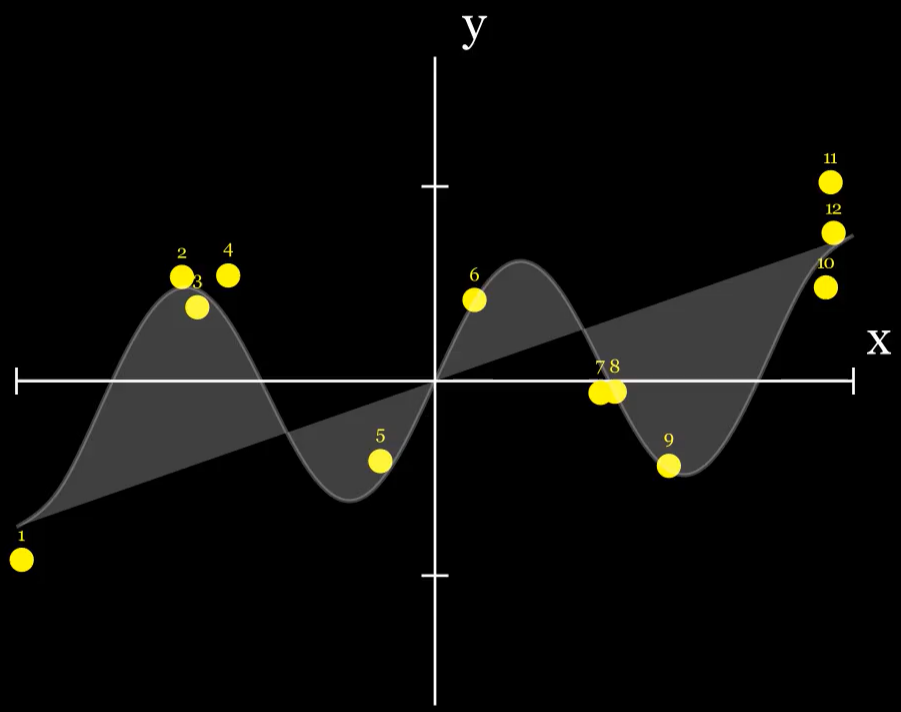
Test Error:

$$\frac{1}{m}\sum_{i=1}^{m} \ell\left(f(\mathbf{x}_i), y_i\right)$$

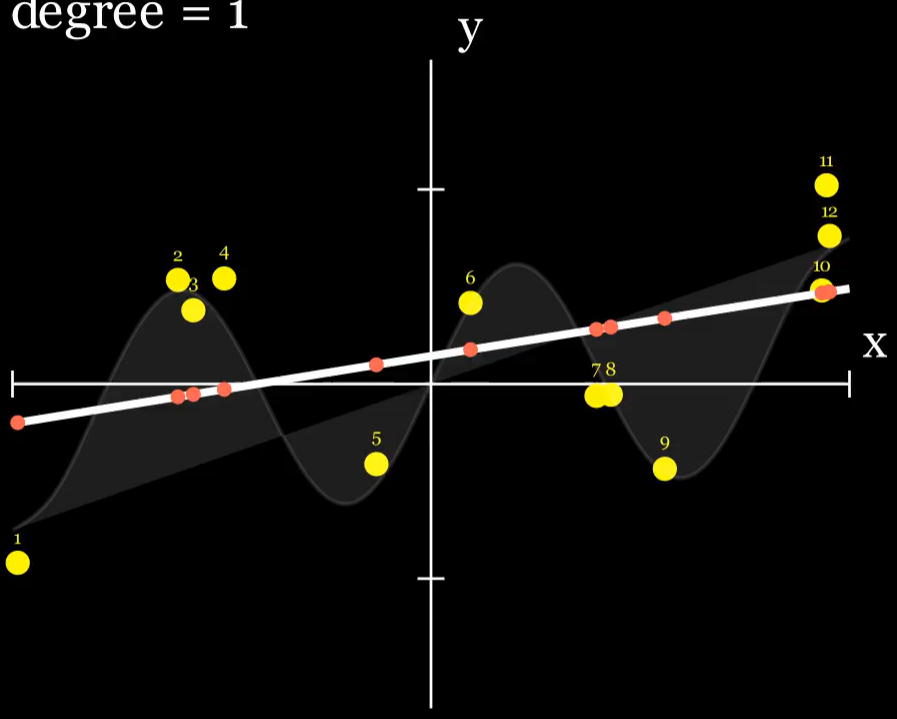
How does Train error and Test error vary with model flexibility?

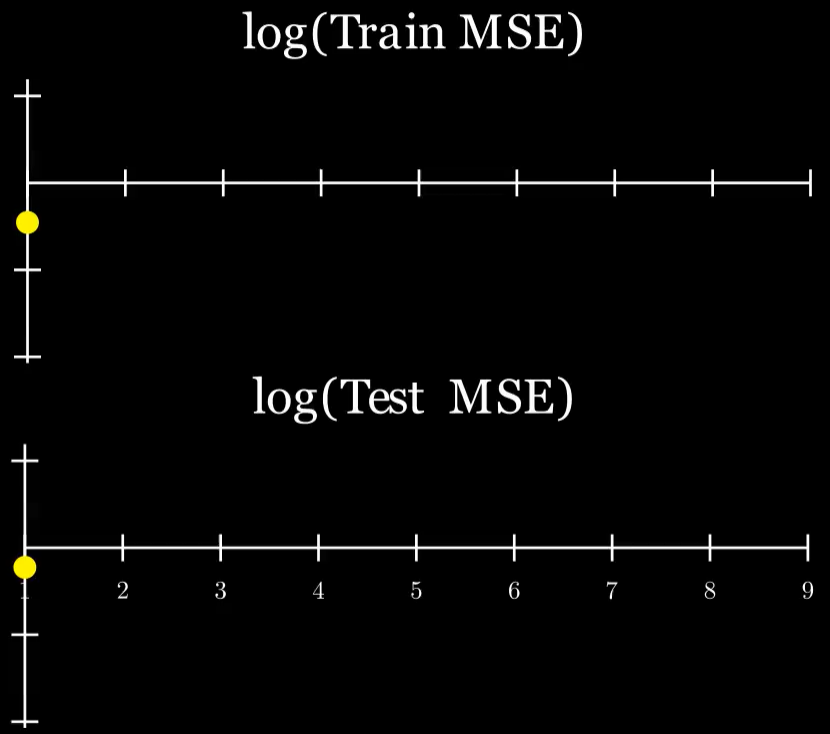

y
x
1
2
3
4
5
6
7
8
9
10
11
12


y
x
1
2
3
4
5
6
7 8
9
10
11
12


y
x
1
2
3
4
5
6
7
8
9
10
11
12

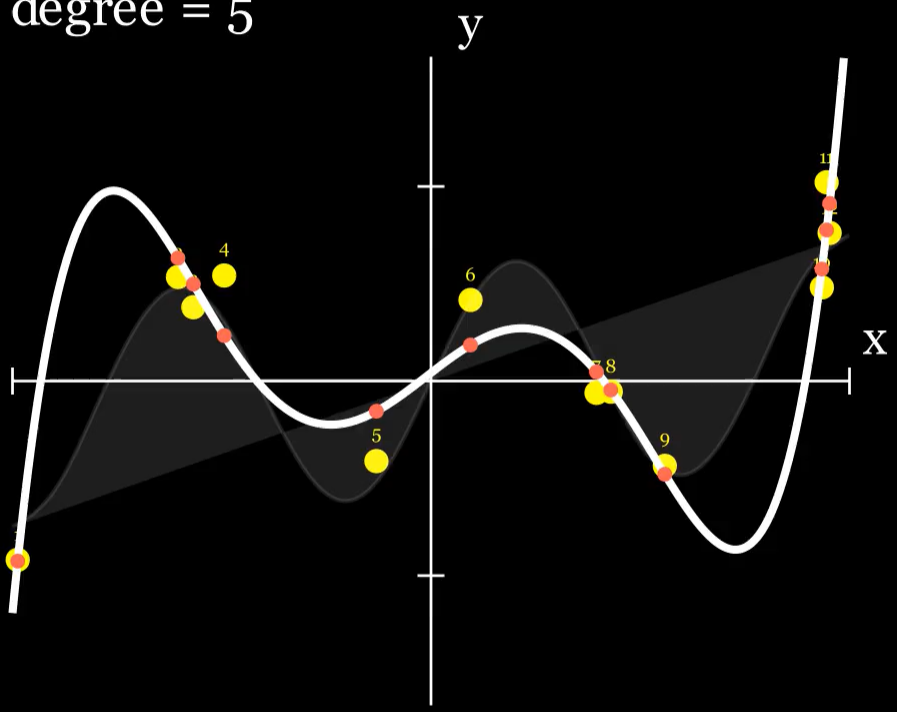
Train MSE: 0.351558
Test MSE: 0.595714





Train MSE: 0.111174
Test MSE: 0.462694

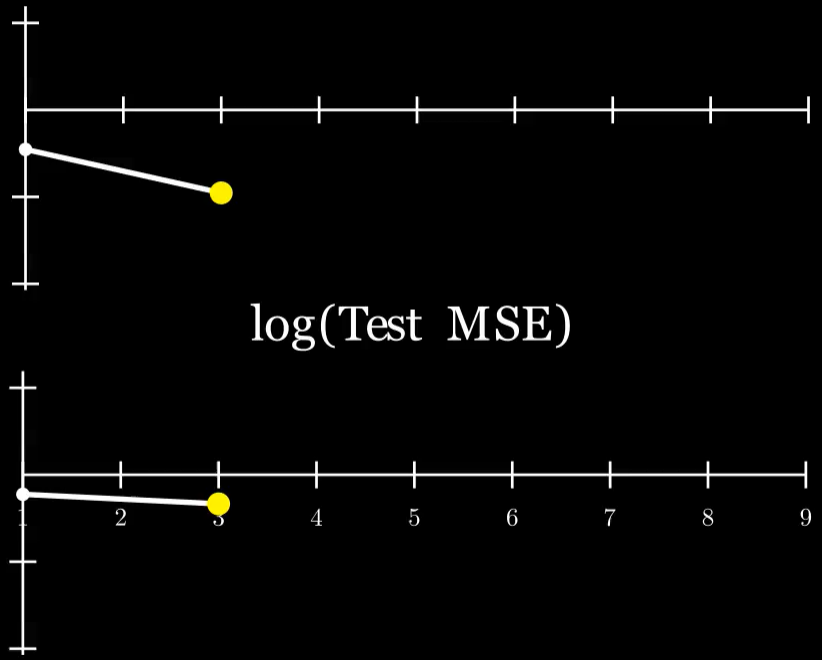
degree = 5



- training data
- prediction value

log(Train MSE)

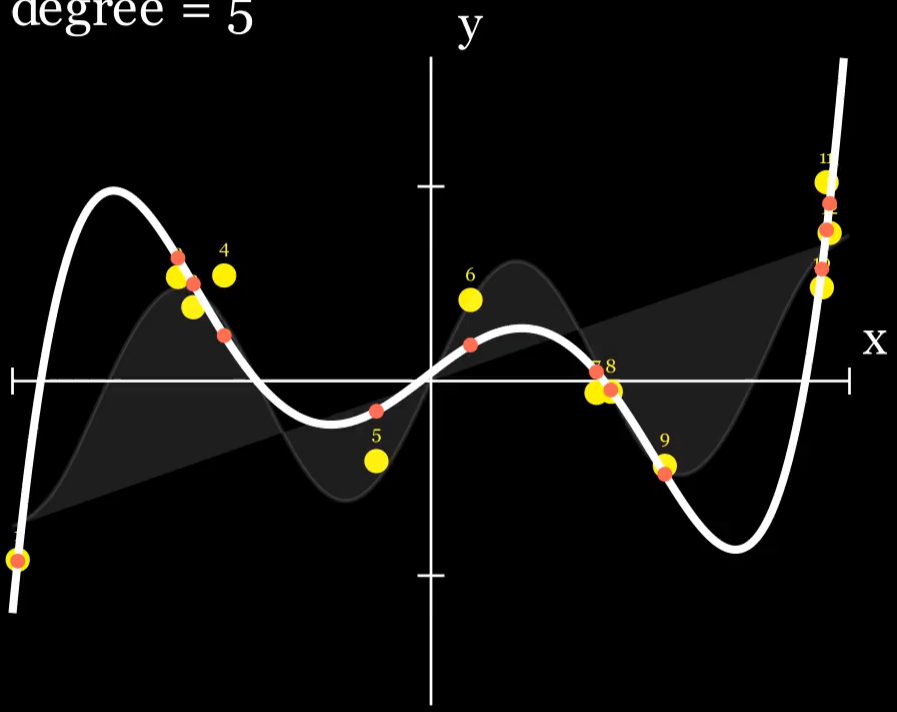
log(Test MSE)



degree

Train MSE: 0.111174
Test MSE: 0.462694

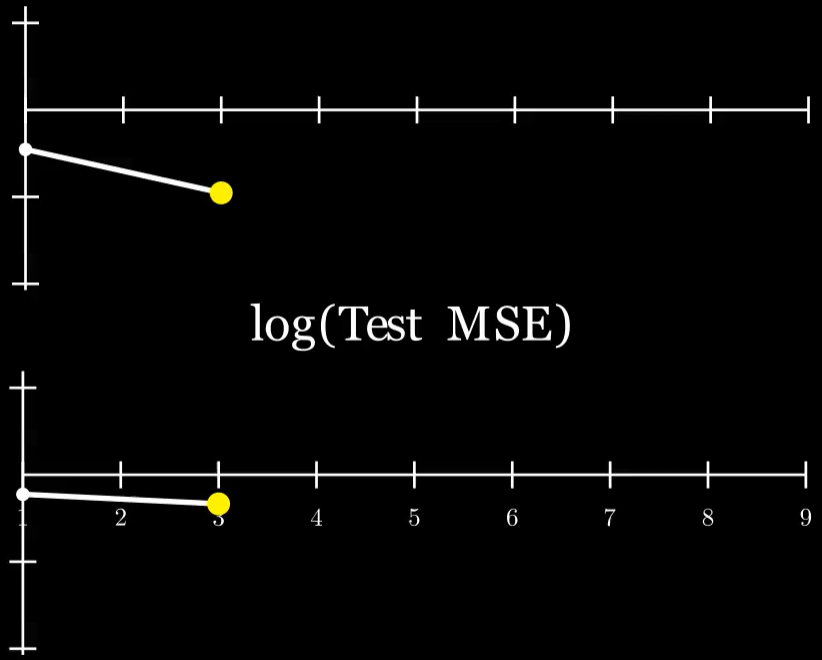
degree = 5



- training data
- prediction value

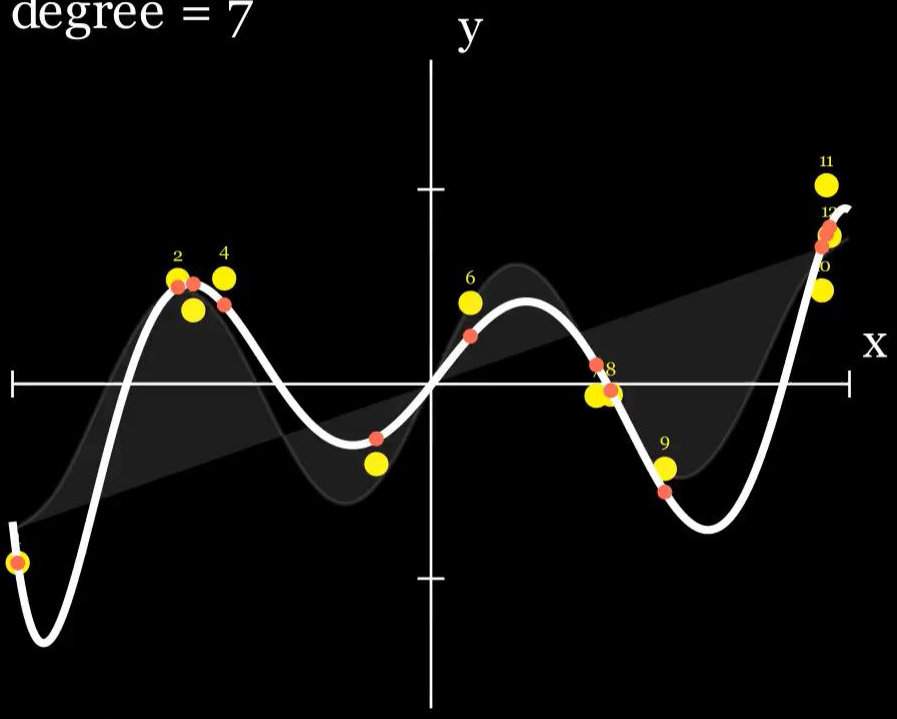
log(Train MSE)

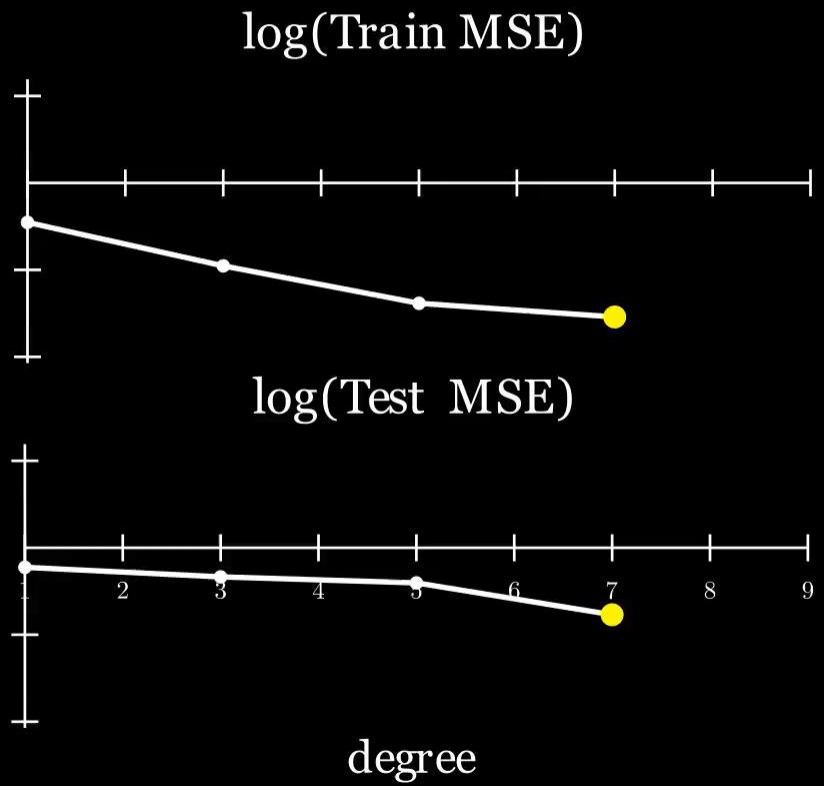
log(Test MSE)



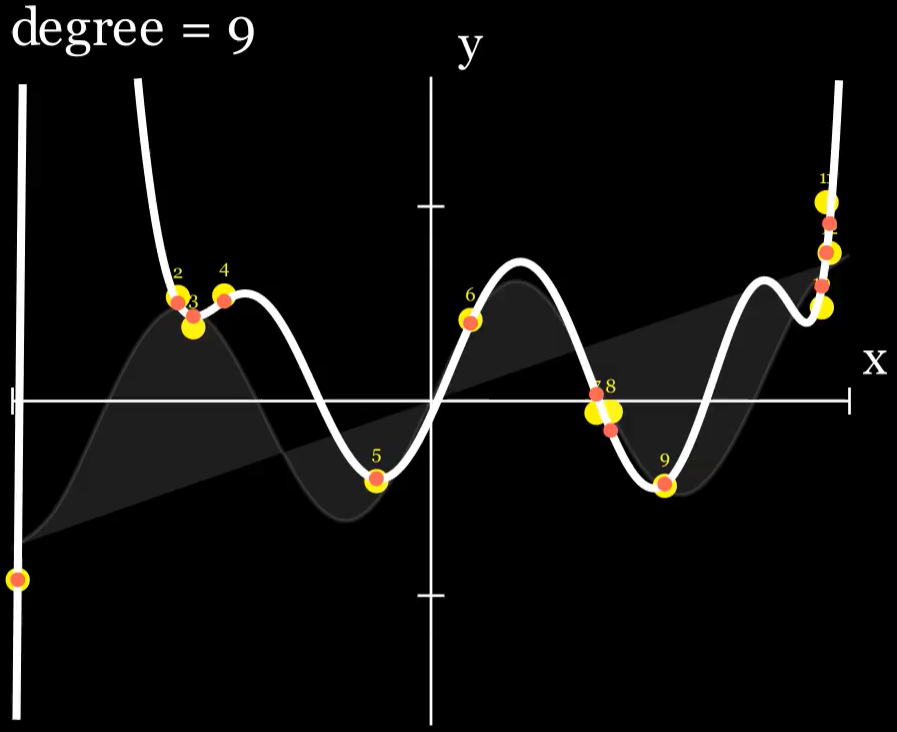
degree

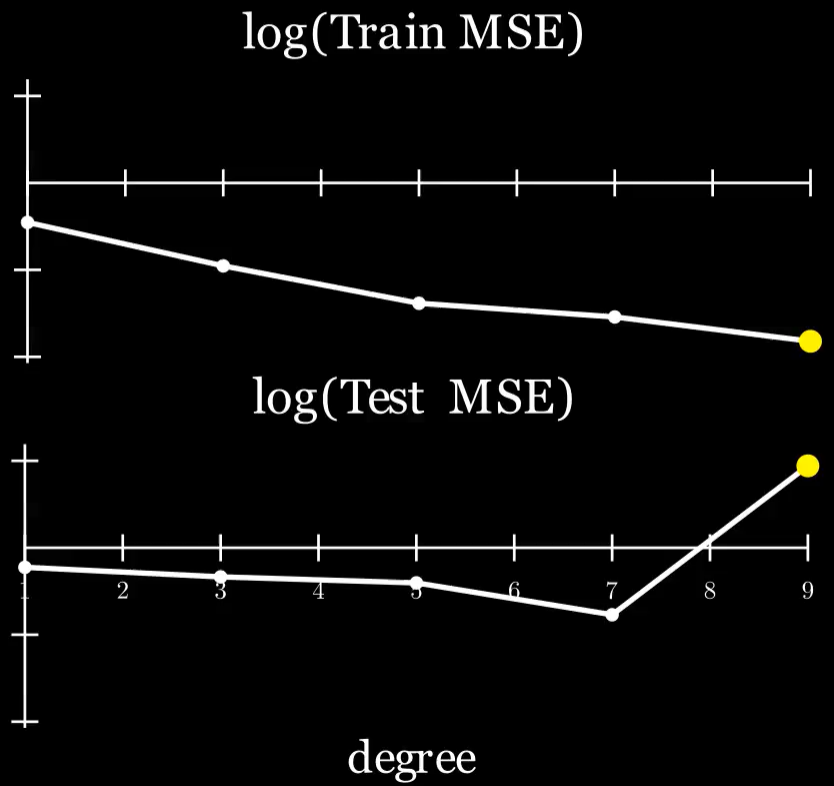
Train MSE: 0.028738
Test MSE: 0.169631



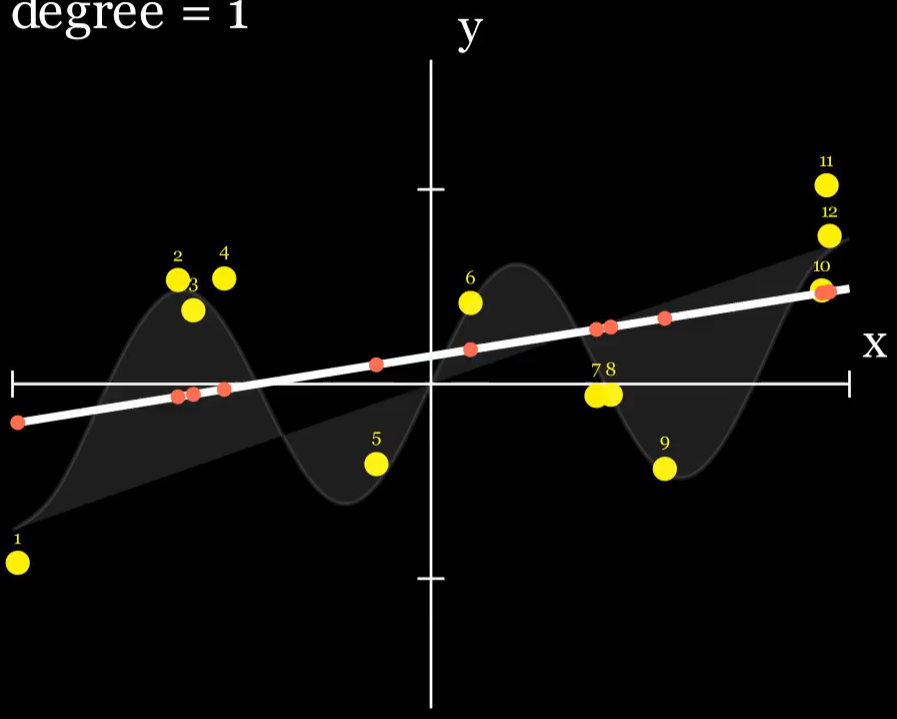


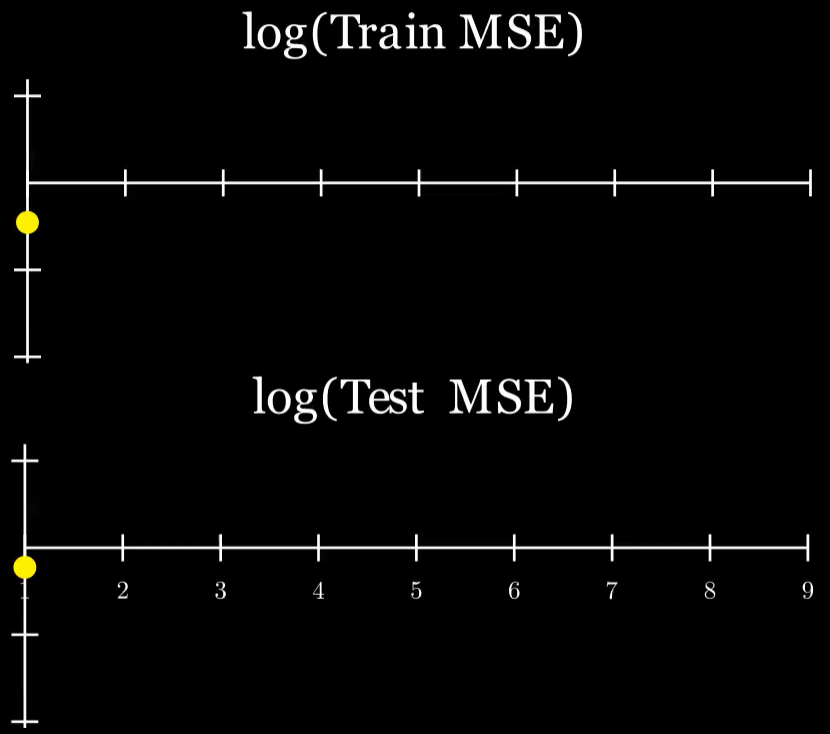
Train MSE: 0.015098
Test MSE: 8.752367





Train MSE: 0.351558
Test MSE: 0.595714





Why Does Test Error First Decrease and Then Increase?

Bias-Variance Trade-off

Probabilistic View of Supervised Learning

Assume there is an underlying data distribution

$$(X, Y) \sim \mathcal{D}.$$

True Regression Function:

$$f^\star = \operatorname{argmin}_f \mathbb{E}_{(X,Y)\sim\mathcal{D}}[(Y - f(X))^2]$$

We cannot find true regression f^*:
To find this f^* we need to know the true data distribution $\mathcal{D}$, which is not known to us.

Learn an approximate function from data :
We observe a random training dataset of n i.i.d. samples:

$$\mathcal{S} = \{(X_i, Y_i)\}_{i=1}^n \sim \mathcal{D}$$

A learning algorithm train on the random dataset $\mathcal{S}$
and output an approximate function $\hat{f}_{\mathcal{S}}$.

Exercise: Show that the true regression function $f^\star$ is given by:

$$f^\star(X) = \mathbb{E}[Y \mid X].$$

Bias–Variance Decomposition

Fix a test input value $\mathbf{x}$. Consider the expected test error at $\mathbf{x}$:

$$\mathbb{E}_{S,Y}\left[(Y - \hat{f}_S(\mathbf{x}))^2 \mid X = \mathbf{x}\right].$$

We can decompose into three terms:

$$(\mathbb{E}_S[\hat{f}_S(\mathbf{x})] - f^\star(\mathbf{x}))^2 \; + \; \mathbb{E}_S\left[(\hat{f}_S(\mathbf{x}) - \mathbb{E}_S[\hat{f}_S(\mathbf{x})])^2\right] \; + \; Var\,(Y \mid X = \mathbf{x})$$

Bias² **Variance** **Part of Y that cannot be explained by X**

$$\boxed{\text{Expected Test error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}}$$

Exercise: Derive the Bias-Variance decomposition

$$\mathbb{E}_{S,Y}\left[(Y - \hat{f}_S(\mathbf{x}))^2 \mid X = \mathbf{x}\right]$$
$$= (\mathbb{E}_S[\hat{f}_S(\mathbf{x})] - f^\star(\mathbf{x}))^2 + \mathbb{E}_S\left[(\hat{f}_S(\mathbf{x}) - \mathbb{E}_S[\hat{f}_S(\mathbf{x})])^2\right] + \; Var\,(Y \mid X = \mathbf{x})$$

Bias–Variance Decomposition

$$\text{Expected Test error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias: Error from restrictive modeling assumptions
(systematic mismatch from the true function)

Variance: *Due to sensitivity to the training data*
(model changes significantly across datasets for the same data distribution)

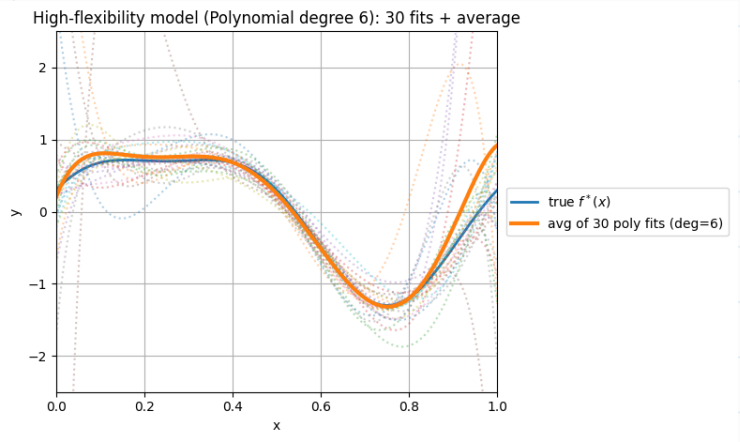
Irreducible Error: Error due to factors the features cannot explain
(cannot be reduced by any model or by any amount of data)

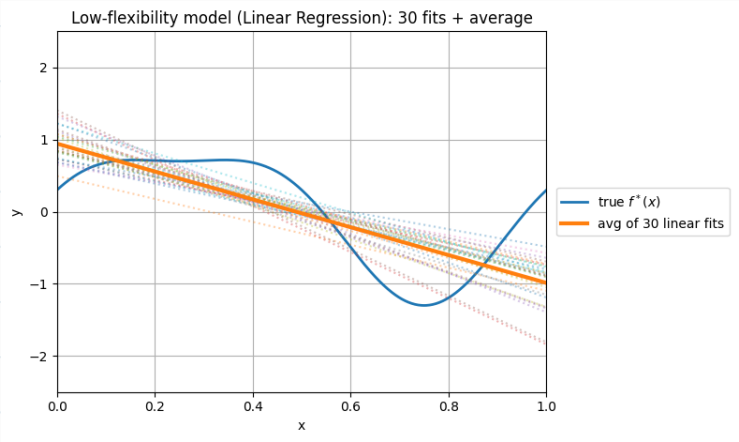
Demo on Bias-Variance Trade-off

Scan this QR for Demo Link

Bias–Variance Tradeoff (Demo)

$$\text{Expected Test error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$





Less flexible models: higher bias, lower variance (**underfitting**)

Highly flexible models: lower bias, higher variance (**overfitting**)

Can we use Test Set to decide model flexibility?

No! Using test error to select model complexity is like:

Peeking at the exam paper to decide how much to study.

What do we do instead ?

Use mock exam papers.

Validation Set:

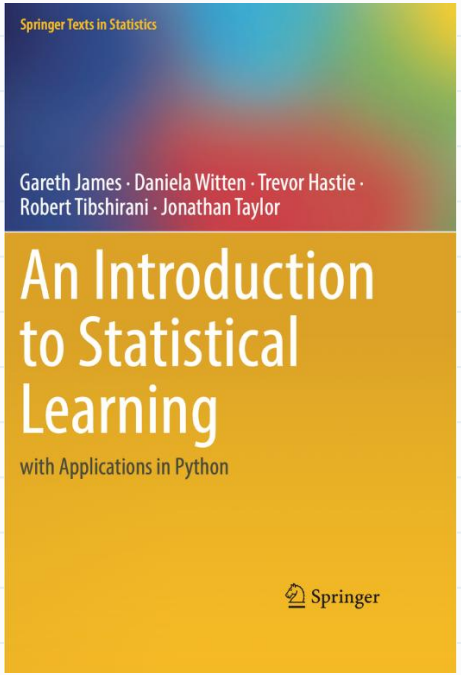
Split the available training data into two parts:

- **Training set** – used to fit the best function in your hypothesis class
- **Validation set** – used to select the hypothesis class (model flexibility, hyperparameters)

Cross-validation Technique

Reference

Read Chapter 2 of this book.



Download Link: https://www.statlearning.com/

Thank You