Python Data Products

Course 3: Making Meaningful Predictions from Data

Lecture: Over and underfitting

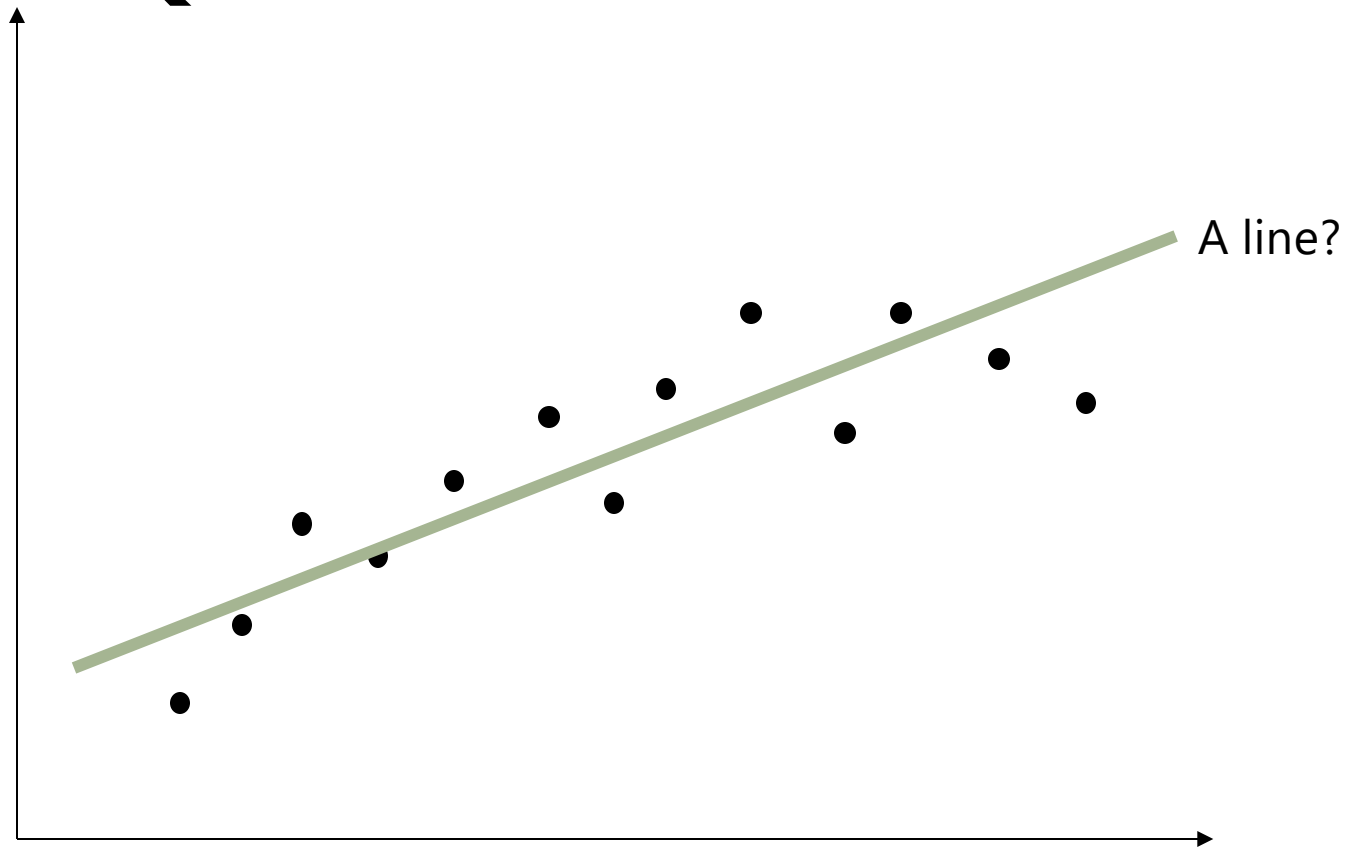
Learning objectives

In this lecture we will...

- Introduce the concept of **overfitting**
- expand our previous discussion training and test sets

Example

Q: What model is the best fit to this data?



Example

A high-degree polynomial might be the best fit to the data (in terms of the mean-squared error), but intuition tells us this is not a good solution

Overfitting

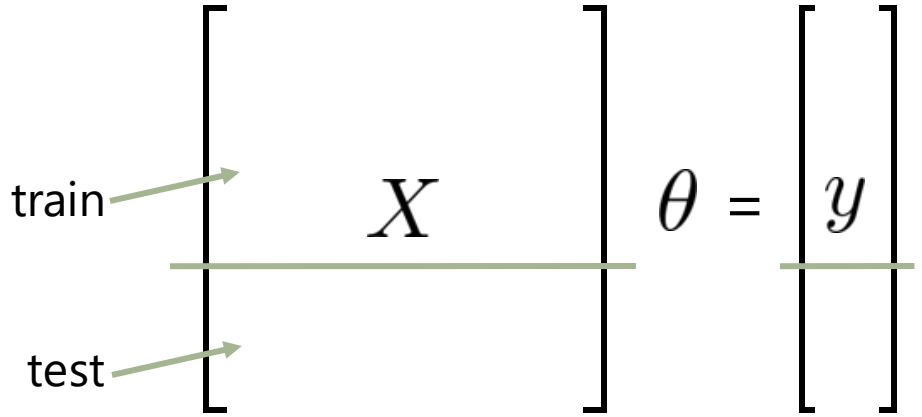
Q: But can't we get an R^2 of 1 (MSE of 0) just by throwing in enough random features?

A: Yes! This is why MSE and R^2 (or other statistics) should be evaluated on data that **wasn't** used to train the model

A good model is one that **generalizes to new data**

Training and test sets

Training data	Testing data



Training and test sets

- We first split our data into a **training** and a **test** set
  - The **training set** is used to tune the model parameters (i.e., θ)
  - The **test set** is used to evaluate the model's performance on unseen data

Training and test sets

How should the training and test sets be selected?

- The training and test sets should be **non-overlapping** samples of the data
- They should each be a **random** sample of the data

Training and test sets

The **size** of the training and test sets should be chosen to balance various considerations:

- The training set should be large enough (compared to the model complexity) so that we don't overfit too badly
- The test set should be large enough so that it is representative of the variance in the data
- We might also be constrained by running time, etc.

Summary of concepts

- Explained the difference between training performance versus generalization ability
- Showed how a training and test set can be used to measure generalization ability