FIT1043 Introduction to Data Science

Week 6

Ian K T Tan

School of Information Technology
Monash University Malaysia

With materials from Wray Buntine, Mahsa Salehi

Week 5 Coverage
Models
Machine Learning

MONASH University

Models and Machine Learning

Week 5 Coverage

Models

What is model?

What are predictive models?

How to evaluate predictive models?

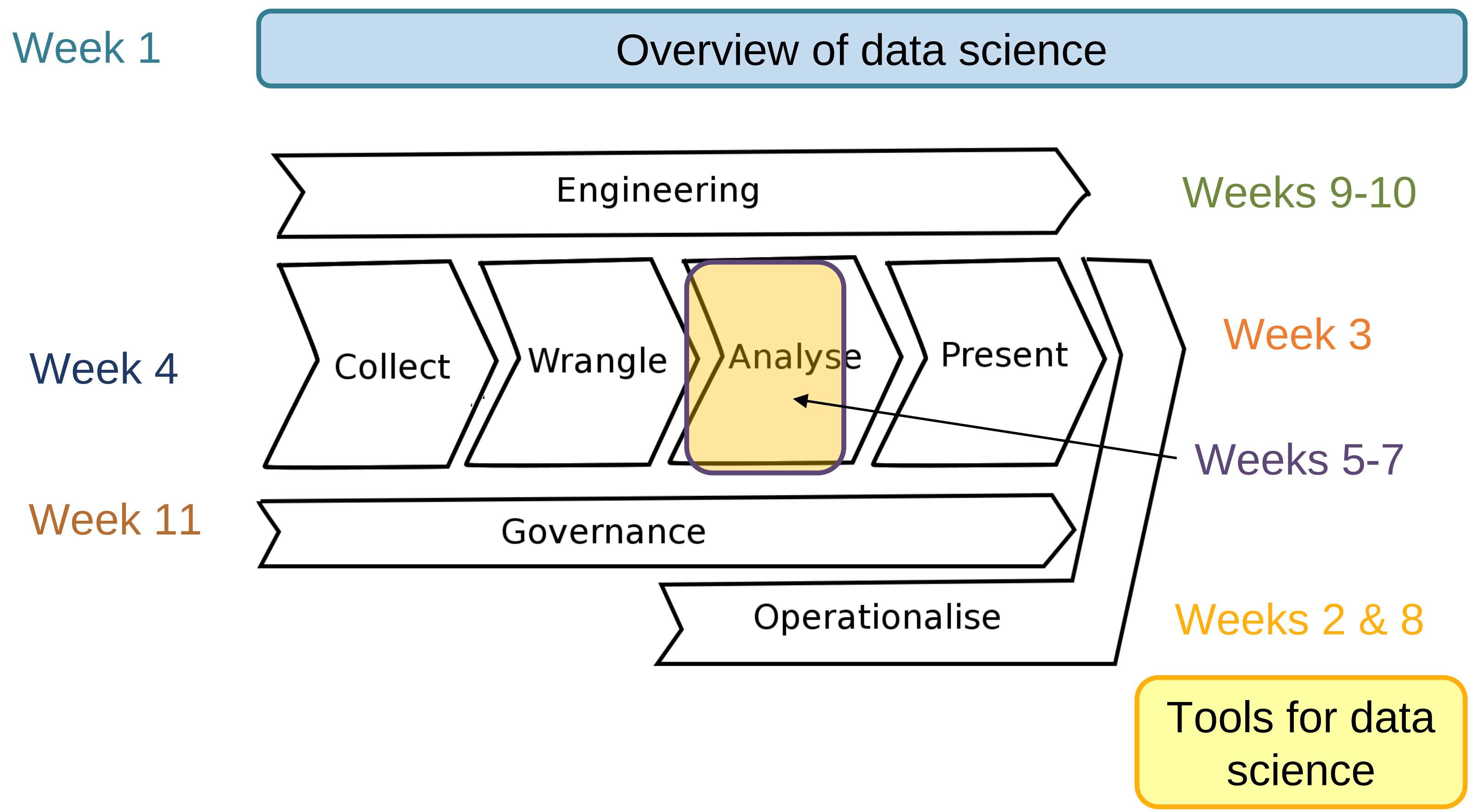
Machine Learning

Machine learning styles

What is learning theory

Linear Regression

Polynomial regression



MONASH University

Week	Activities	Assignments
1	Overview of data science	
2	Introduction to Python for data science	
3	Data visualisation and descriptive statistics	
4	Data sources and data wrangling	
5	Data analysis theory	Assignment 1
6	Regression analysis	
7	Classification and clustering	
8	Introduction to R for data science	Assignment 2
9	Characterising data and "big" data	
10	Big data processing	
11	Issues in data management	Assignment 3
12	Industry guest lecture (tentative)	

Introduction to Data Analysis

Week 6 Outline

- Linear regression terminology
- How to calculate model parameters
- Underfitting vs Overfitting
- Bias and Variance
  - No free lunch theorem
- Ensemble models
- Descriptive vs Predictive Data Analysis

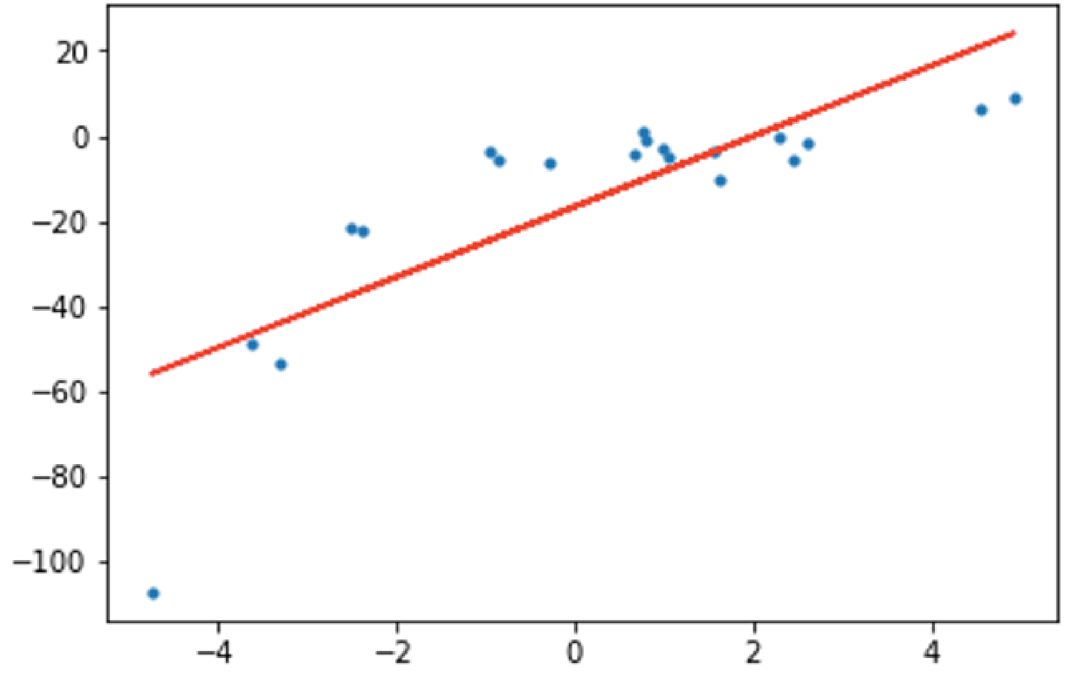
Learning Outcomes

Week 6

By the end of this week you should be able to:

- Fit linear regression and polynomial regression models to a given dataset (in tutorials)
- Explain overfitting and underfitting of different models
- Comprehend bias and variance trade-off
- Comprehend the importance of “No Free Lunch Theorem”
- Explain what ensemble models are

Data Analysis Algorithms
Regression



Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1, no. 10. New York: Springer series in statistics, 2001.

Regression

What is Regression?

- The study of relationship between variables.

Examples

- Identify the relation between
  - Salary and experience, education and role
  - …

Terminology

Variables

- Independent Variables/Inputs/Predictors
  - E.g. Experience, education, role (Categorical or Continuous Data Type)

- Dependent Variables/Outputs/Responses
  - E.g. Salary of employee (Continuous Data Type)

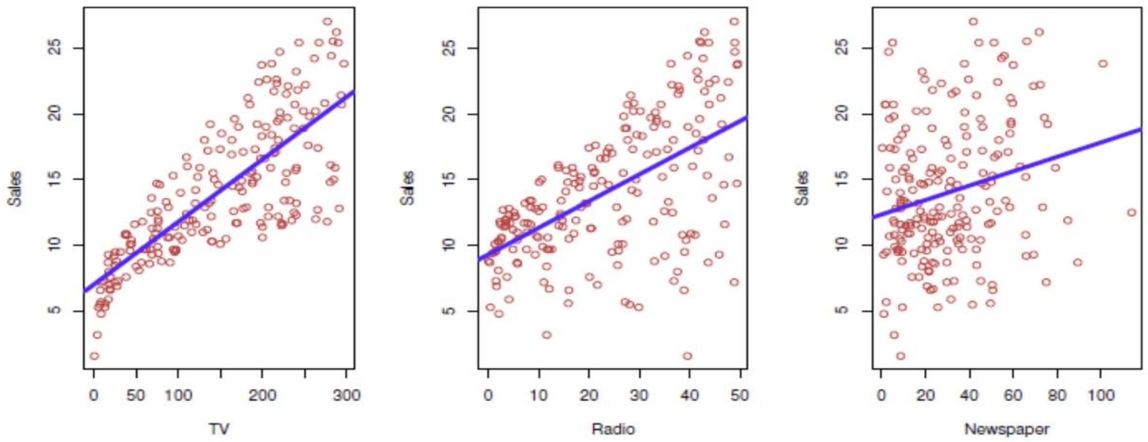
Observations

- An observations is a data point, row, or a sample in a dataset
  - E.g., an employee's salary, experience, education, or role.

MONASH University

When To Use Regression

- To determine how multiple variables are related
    - E.g., determine *if* and *to what extent* the experience or education impact salaries



Example: Sales ~ TV, Radio, Newspaper

- To forecast a value
    - E.g., predict electricity consumption given the outdoor temperature, time of day, and number of residents in that household

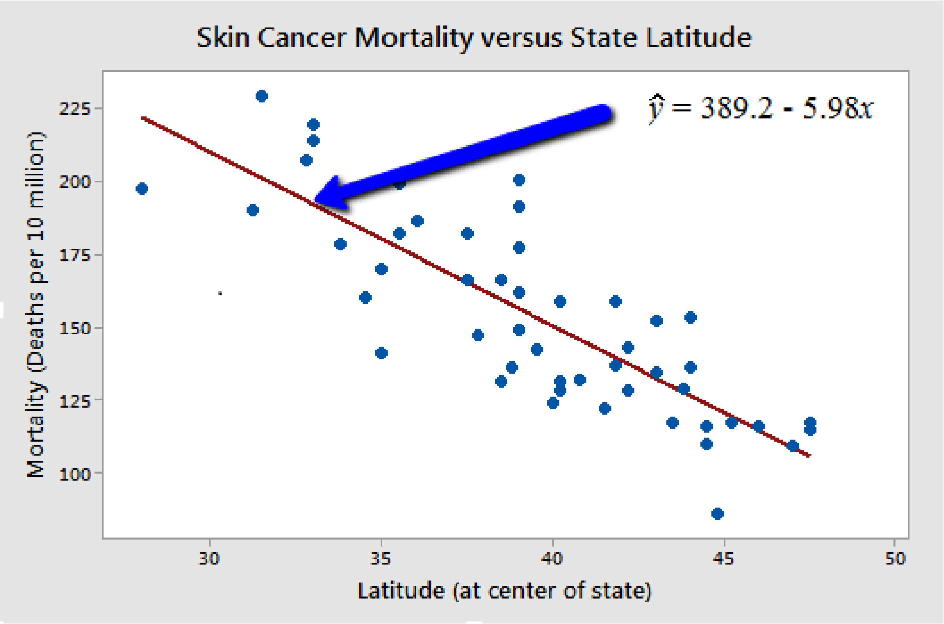
Simple Linear Regression

Two-Dimensional Space

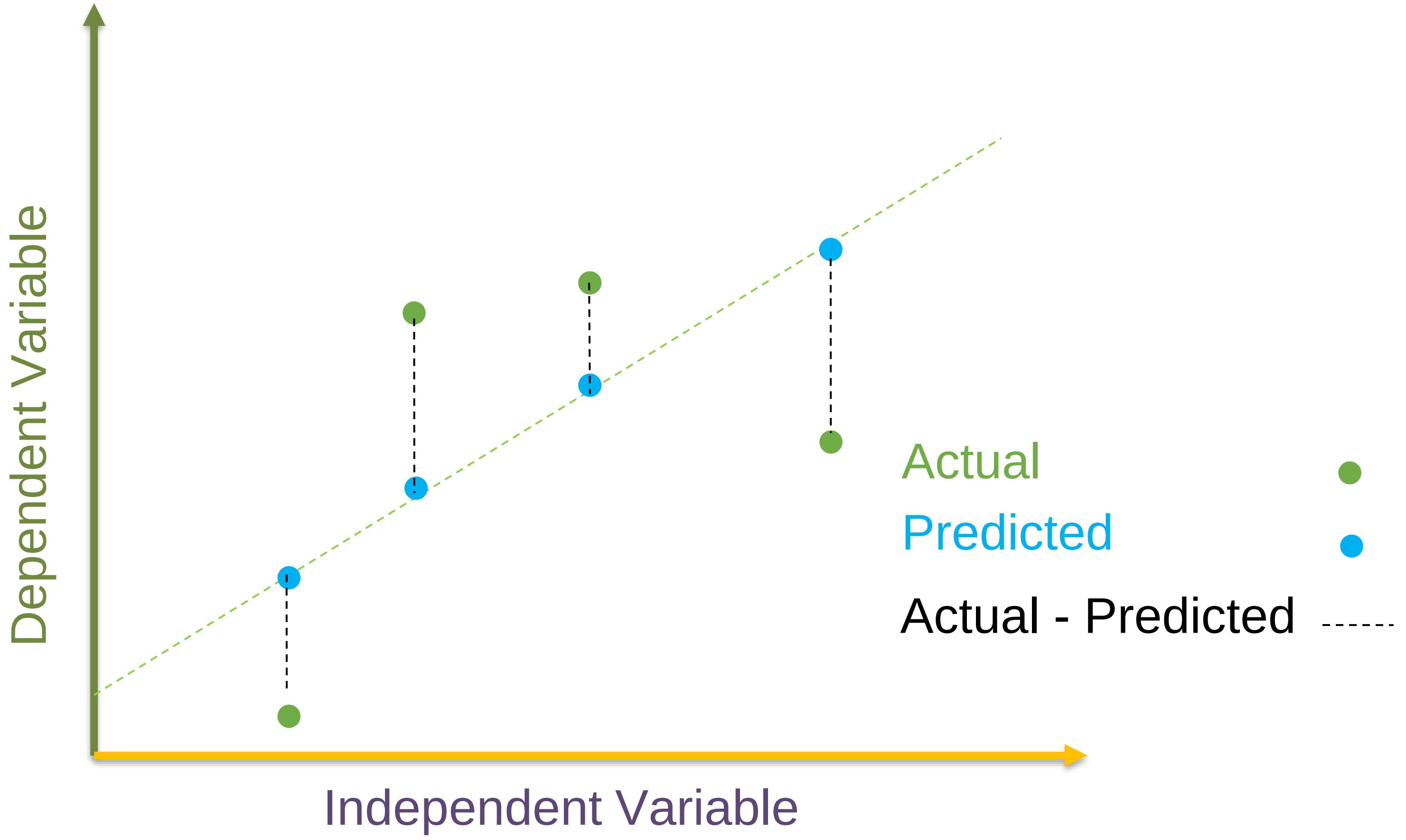
Regression fits a very simple equation to the data:

$$\hat{y}(x; \vec{a}) = a_0 + a_1 x$$

Here $\hat{y}(x; \vec{a})$ is prediction for y at the point x using the model parameters $\vec{a} = (a_0, a_1)$, i.e. the intercept and slope terms.

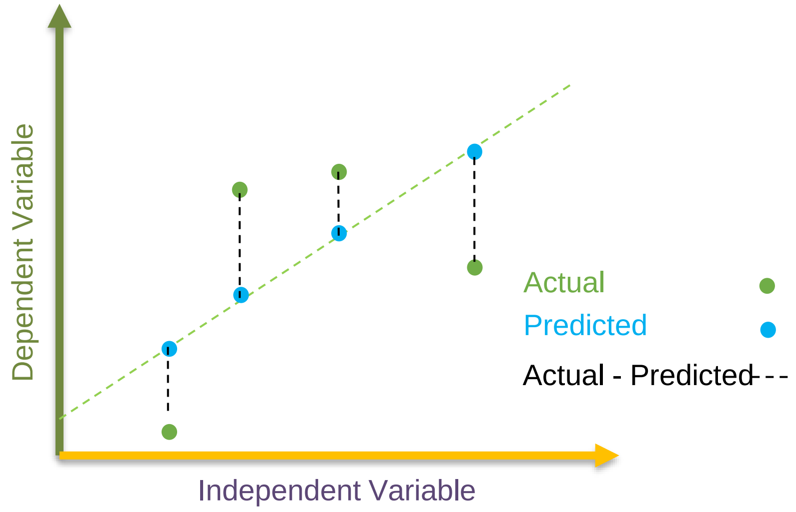


Best Fitting Line



The aim is for the predicted response, be as close as possible to the actual response.

Calculating the Parameters



$$\hat{y}(x; \vec{a}) = a_0 + a_1 x$$

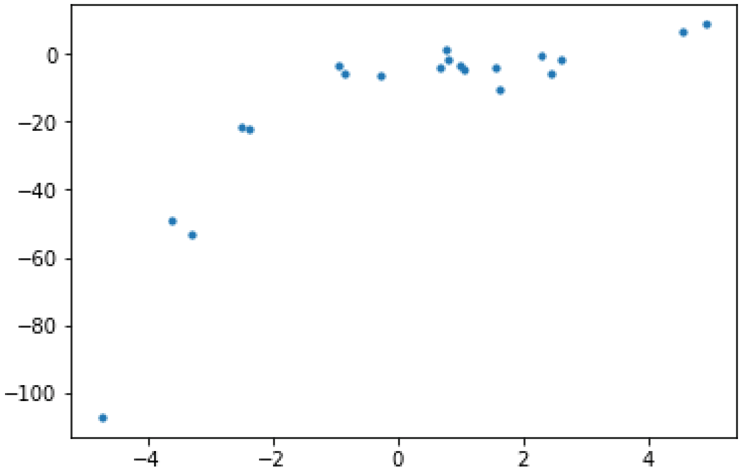
Given some data pairs $(x_1, y_1), \ldots, (x_N, y_N)$, we fit a model by finding the vector $\vec{a}$ that minimises the loss function:

$$\text{mean square error} = MSE_{train} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}(x_i; \vec{a}) - y_i\right)^2$$

MONASH University

Python Example

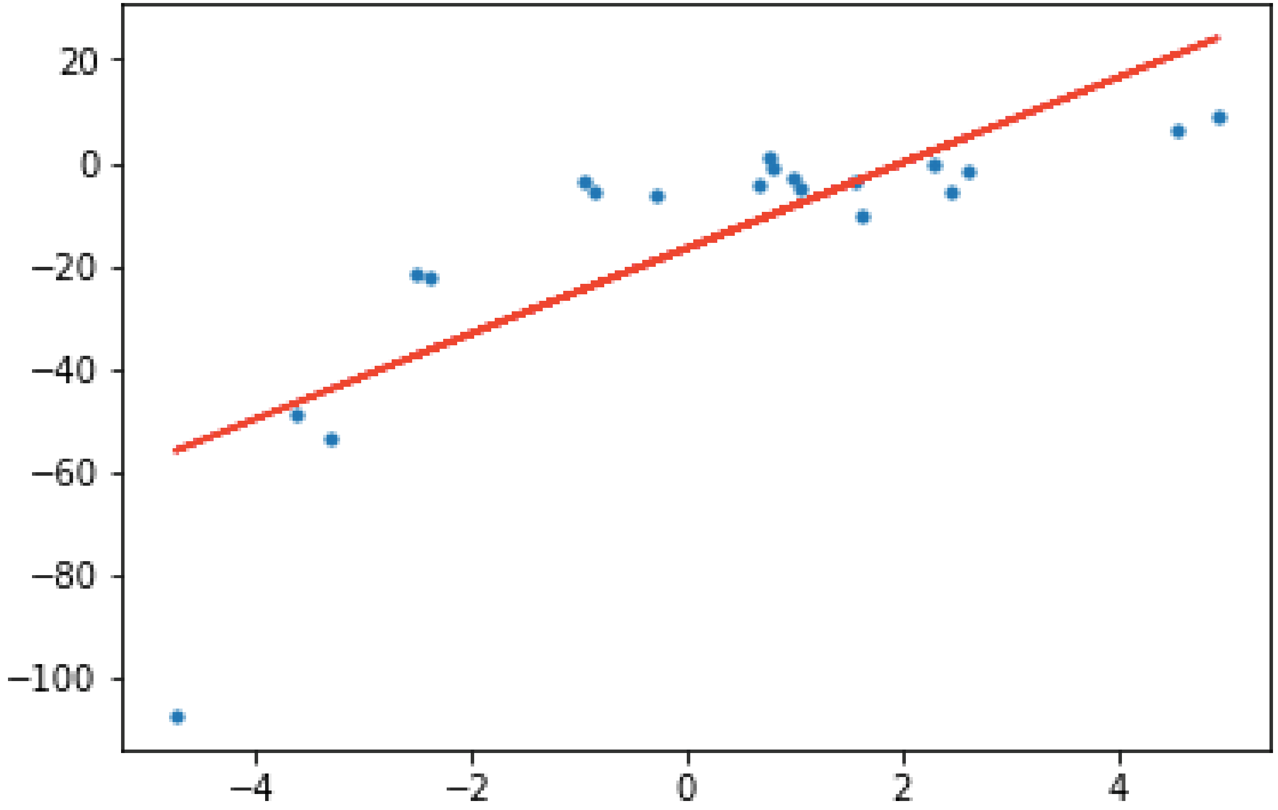
```python
#import required packages
import numpy as np
import matplotlib.pyplot as plt

#provide data
np.random.seed(0)
x = 2 - 3 * np.random.normal(0, 1, 20)
y = x - 2 * (x ** 2) + 0.5 * (x ** 3) + np.random.normal(-3, 3, 20)
plt.scatter(x,y, s=10)
plt.show()
```



Python Example

```
#import required packages
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

#provide data
np.random.seed(0)
x = 2 - 3 * np.random.normal(0, 1, 20)
y = x - 2 * (x ** 2) + 0.5 * (x ** 3) + np.random.normal(-3, 3, 20)

# transforming the data to include another axis
x = x[:, np.newaxis]
y = y[:, np.newaxis]

#create a liniear regression model
model = LinearRegression()
model.fit(x, y)
y_pred = model.predict(x)

#diplay the best fit line
plt.scatter(x, y, s=10)
plt.plot(x, y_pred, color='r')
plt.show()
```

Linear regression is unable to capture the patterns in the data. This is an example of under-fitting.



To overcome under-fitting, we need to increase the complexity of the model

Python Polynomial Regression

To overcome under-fitting, we need to increase the complexity of the model

Assume the polynomial relationship between the inputs and output.

E.g., 10th order (aka degree) polynomial

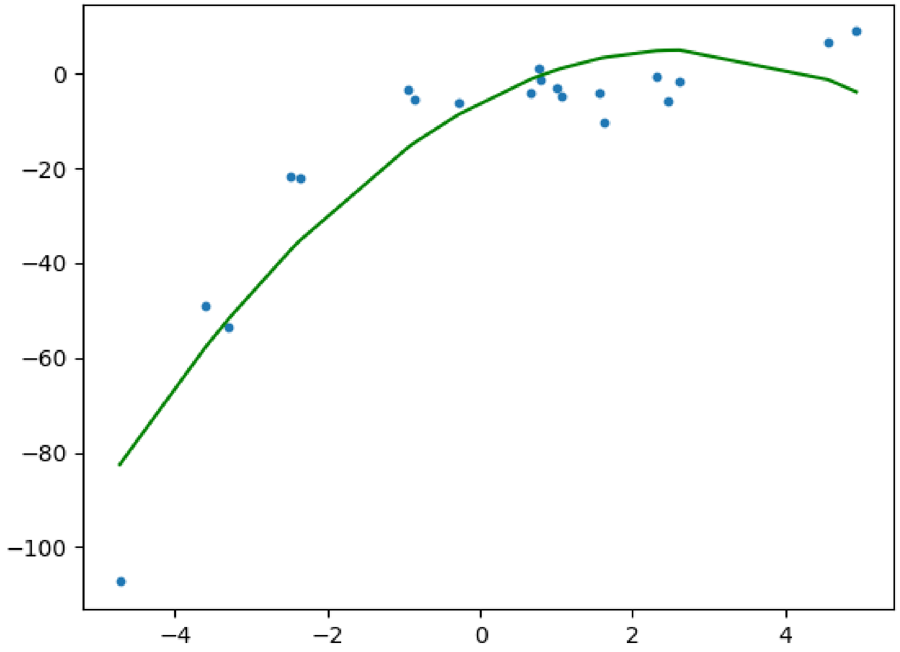
$$\hat{y}(x; \vec{a}) = a_0 + a_1 x$$

Python Polynomial Regression

2nd Degree Polynomial (Polynomial of order 2)

```
#import required packages
import operator
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures

#provide data
np.random.seed(0)
x = 2 - 3 * np.random.normal(0, 1, 20)
y = x - 2 * (x ** 2) + 0.5 * (x ** 3) + np.random.normal(-3, 3, 20)

# transforming the data to include another axis
x = x[:, np.newaxis]
y = y[:, np.newaxis]

#create polynomial regression
polynomial_features= PolynomialFeatures(degree= 2)
x_poly = polynomial_features.fit_transform(x)

model = LinearRegression()
model.fit(x_poly, y)
y_poly_pred = model.predict(x_poly)

rmse = np.sqrt(mean_squared_error(y,y_poly_pred))
r2 = r2_score(y,y_poly_pred)
print(rmse)
print(r2)

plt.scatter(x, y, s=10)
# sort the values of x before line plot
sort_axis = operator.itemgetter(0)
sorted_zip = sorted(zip(x,y_poly_pred), key=sort_axis)
x, y_poly_pred = zip(*sorted_zip)
plt.plot(x, y_poly_pred, color='m')
plt.show()
```

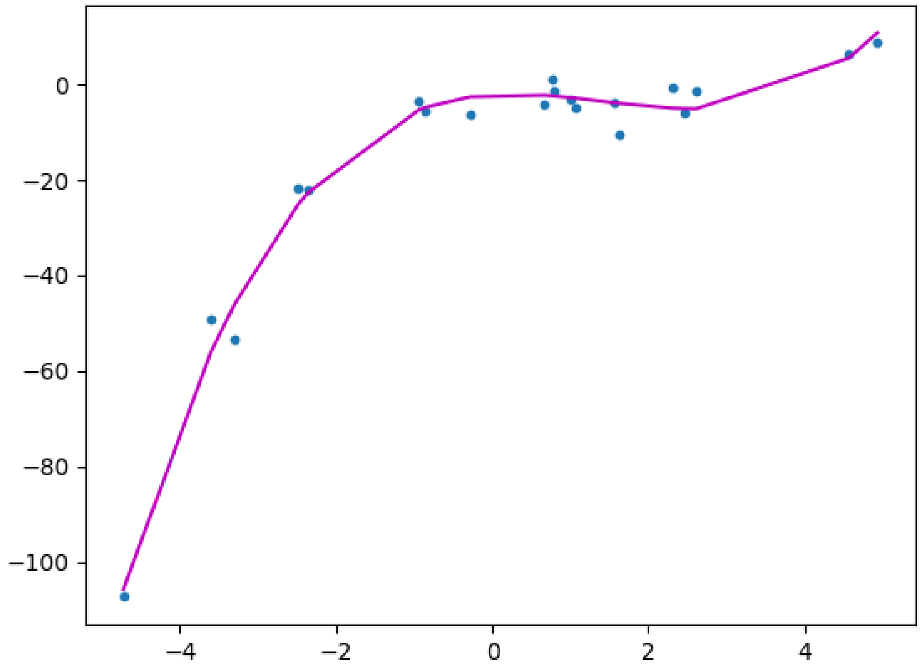
$$\hat{y}(x;\vec{a}) = a_0 + a_1 x + a_2 x^2$$



MONASH University

Python Polynomial Regression

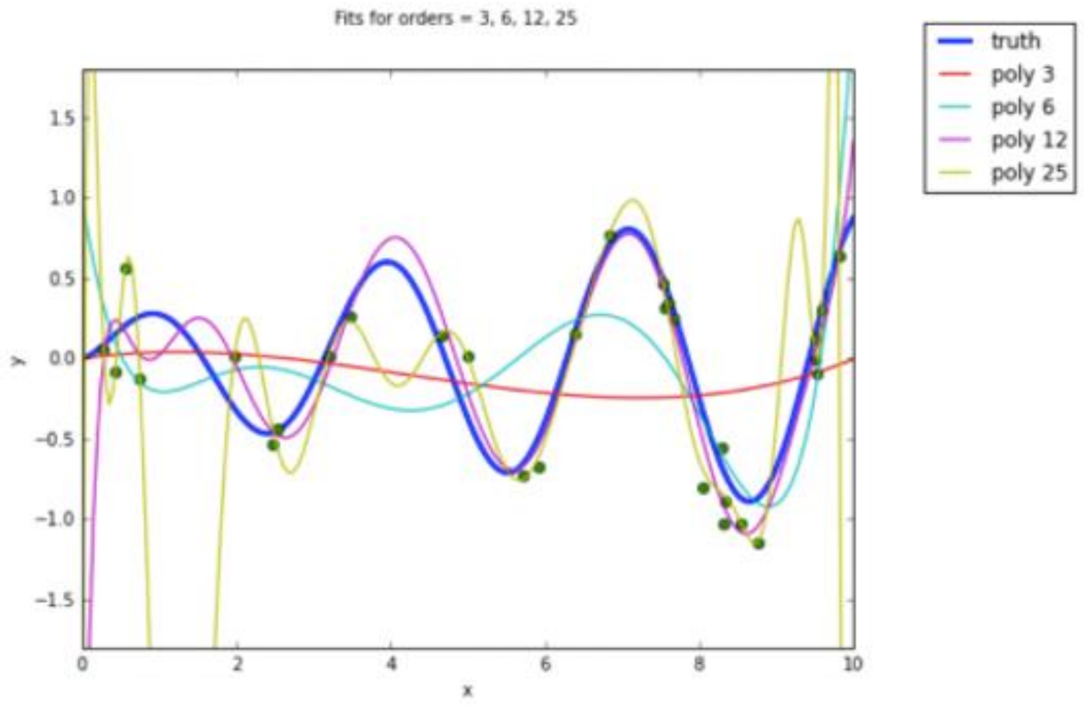
3rd Degree Polynomial (Polynomial of order 3)



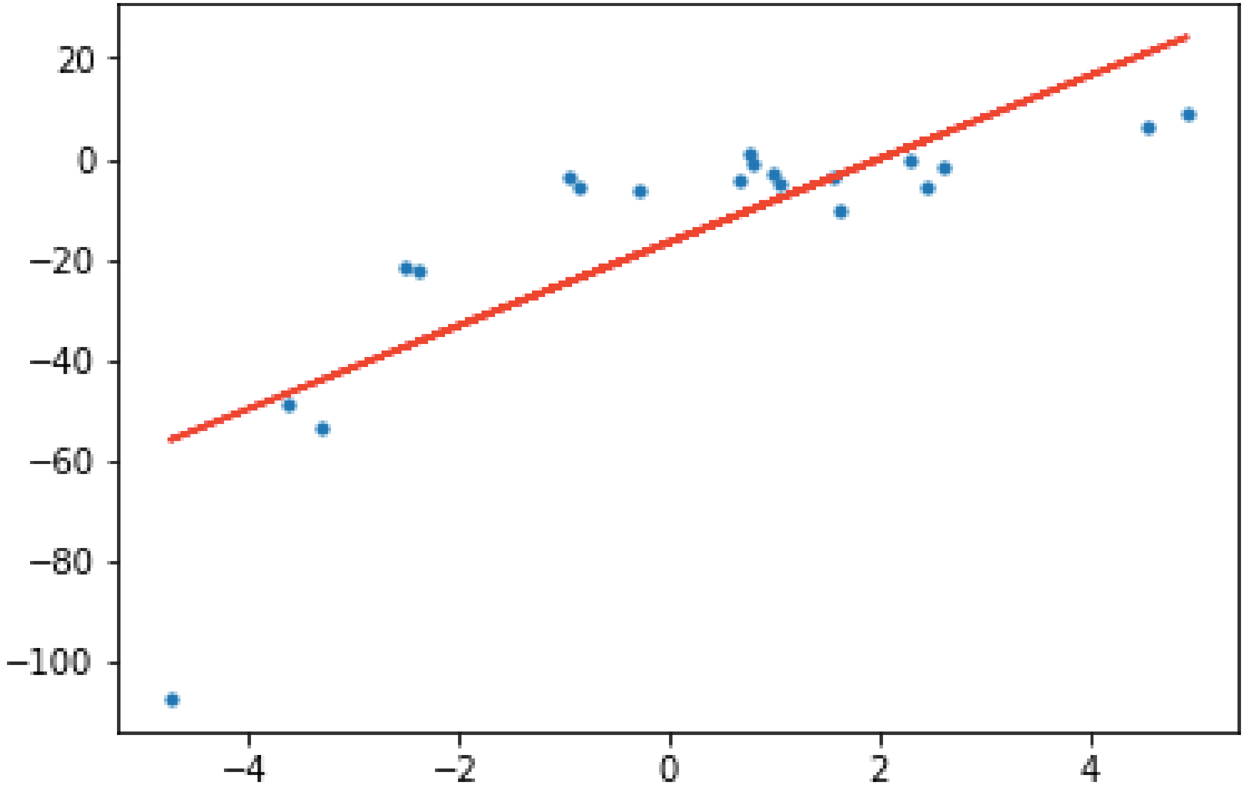
$$\hat{y}(x; \vec{a}) = a_0 + a_1 x + a_2 x^2$$

Underfitting and Overfitting

Model Fitting

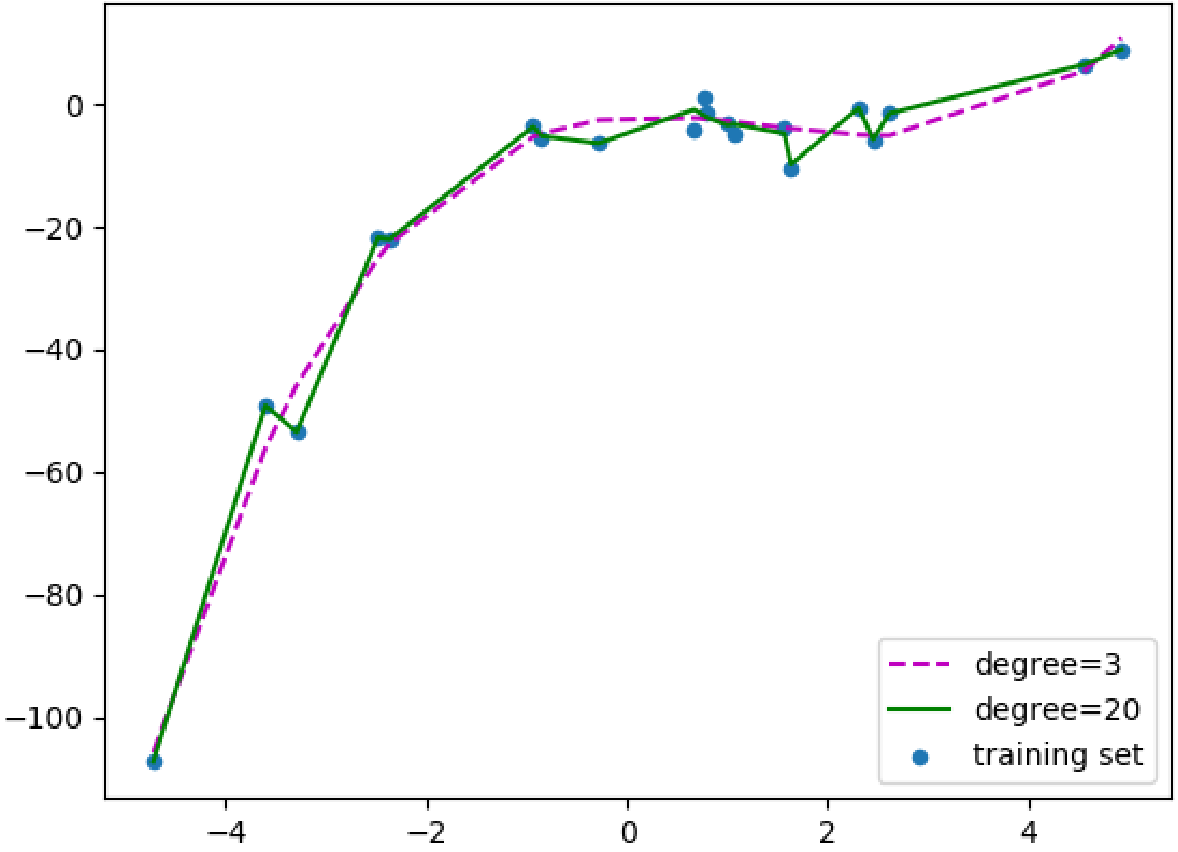


Underfitting and Overfitting



Under-Fitting

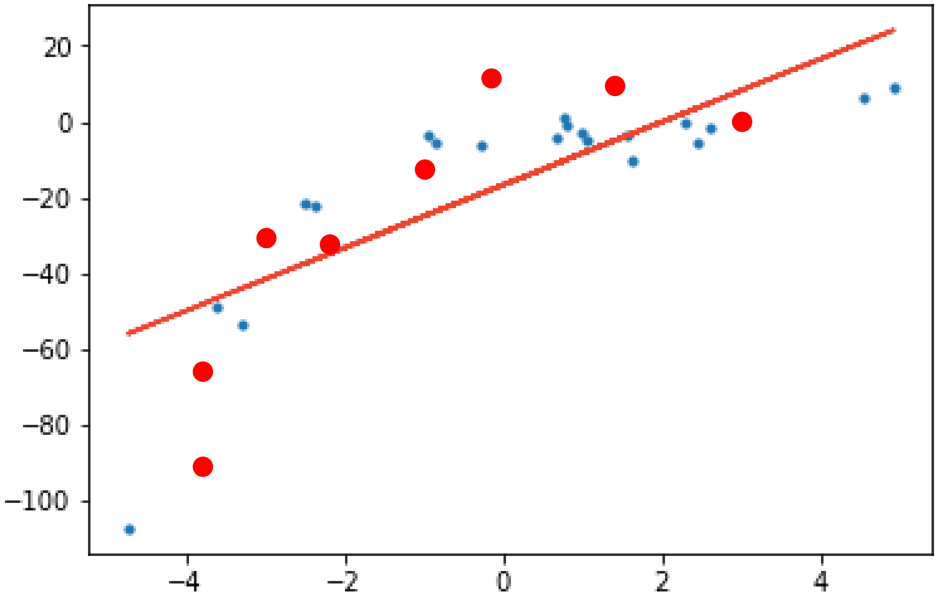
Underfitting and Overfitting



Over-Fitting

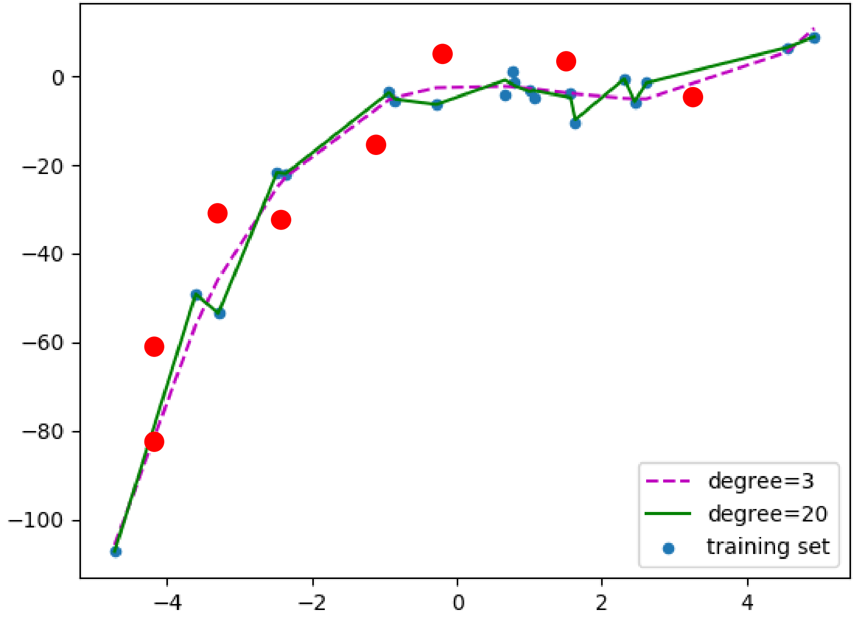
MONASH University

Underfitting and Overfitting

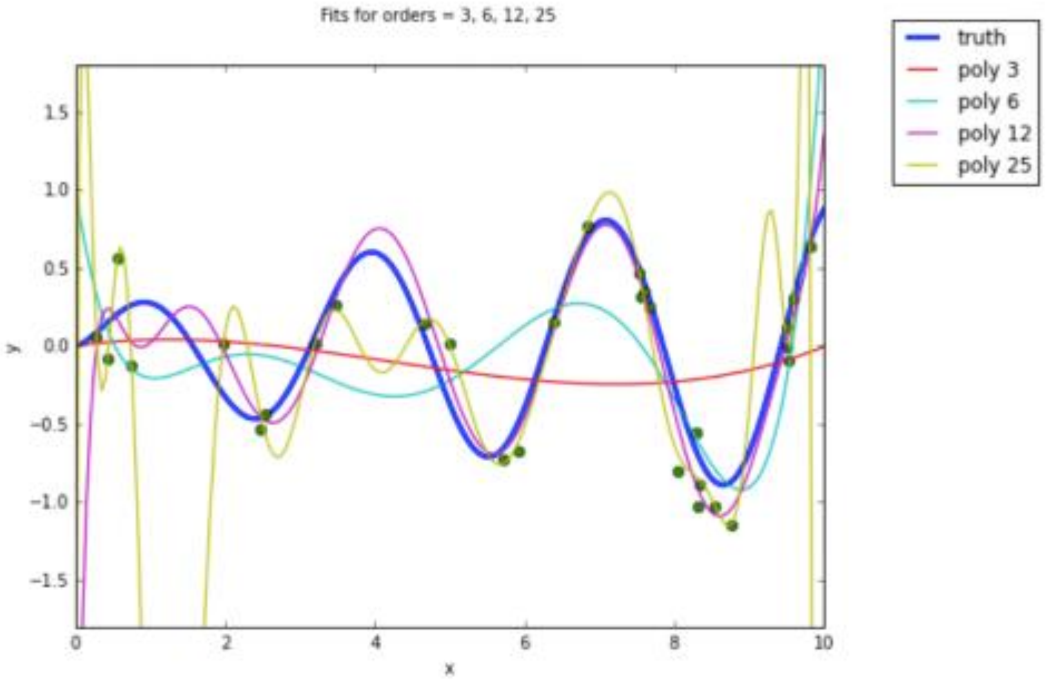


Under-Fitting

Over-Fitting



Overfitting



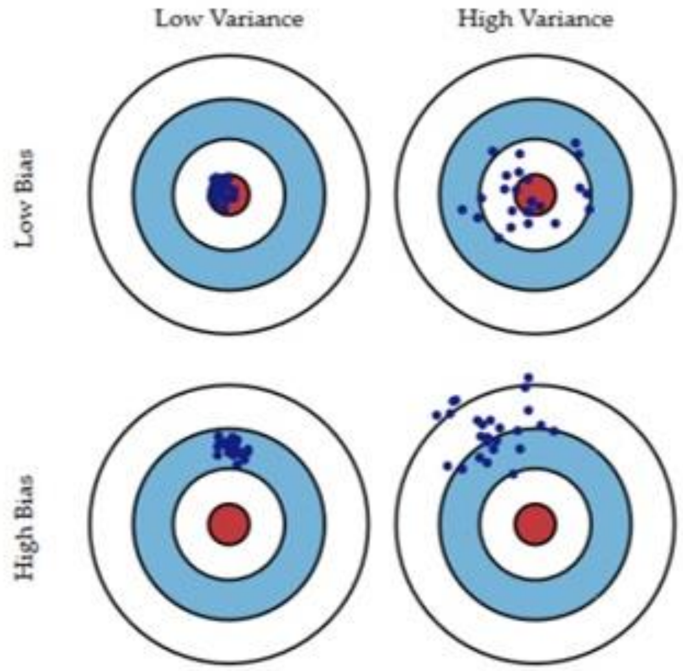
- The more parameters a model has, the more complicated a curve it can fit.
  - If we don't have very much data and we try to fit a complicated model to it, the model will make wild predictions.
  - This phenomenon is referred to as overfitting

MONASH University

Overfitting

- Small polynomial; cannot fit the data well; said to have high bias
- Large polynomial; can fit the data well; fits the data too well; said to have small bias
- If there is known error in the data, then a close fit is wasted:
- 25th degree polynomial does all sorts of wild contortions!
- Poor fit due to **high bias** called underfitting
- Poor fit due to **low bias** called overfitting

Bias and Variance

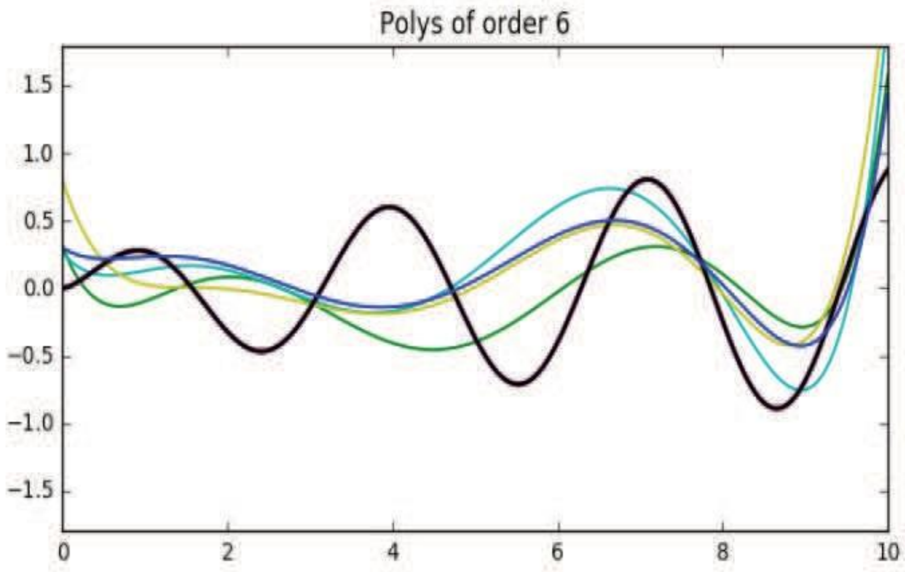


MONASH University

Training and Test Datasets

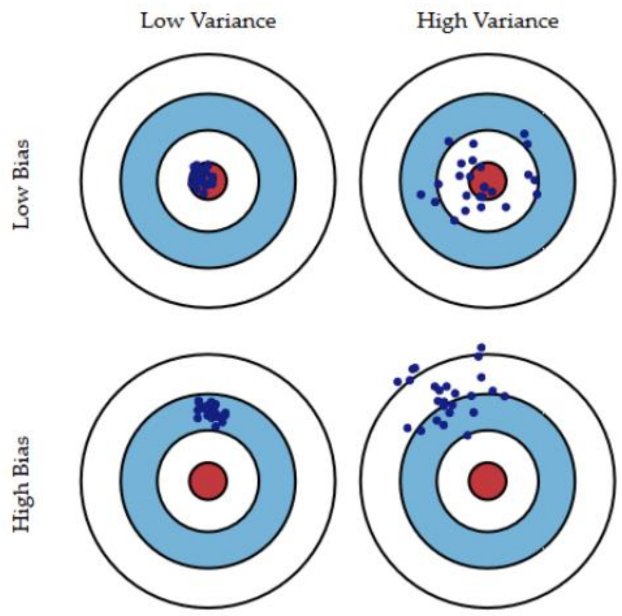
- Split up the data we have into two non-overlapping parts, a training set and a test set
- Do your learning, run your algorithm, build your model using the training set
- Run evaluation using the test set
- Don’t run evaluation on the training set
- How big to make the test set?

Bias and Variance



- Using different data points, the plot is for 6th degree polynomial models.
  - How do we measure how much does it differ from the desired model?
  - For each data point, does the prediction differ a lot each time?

Bias and Variance

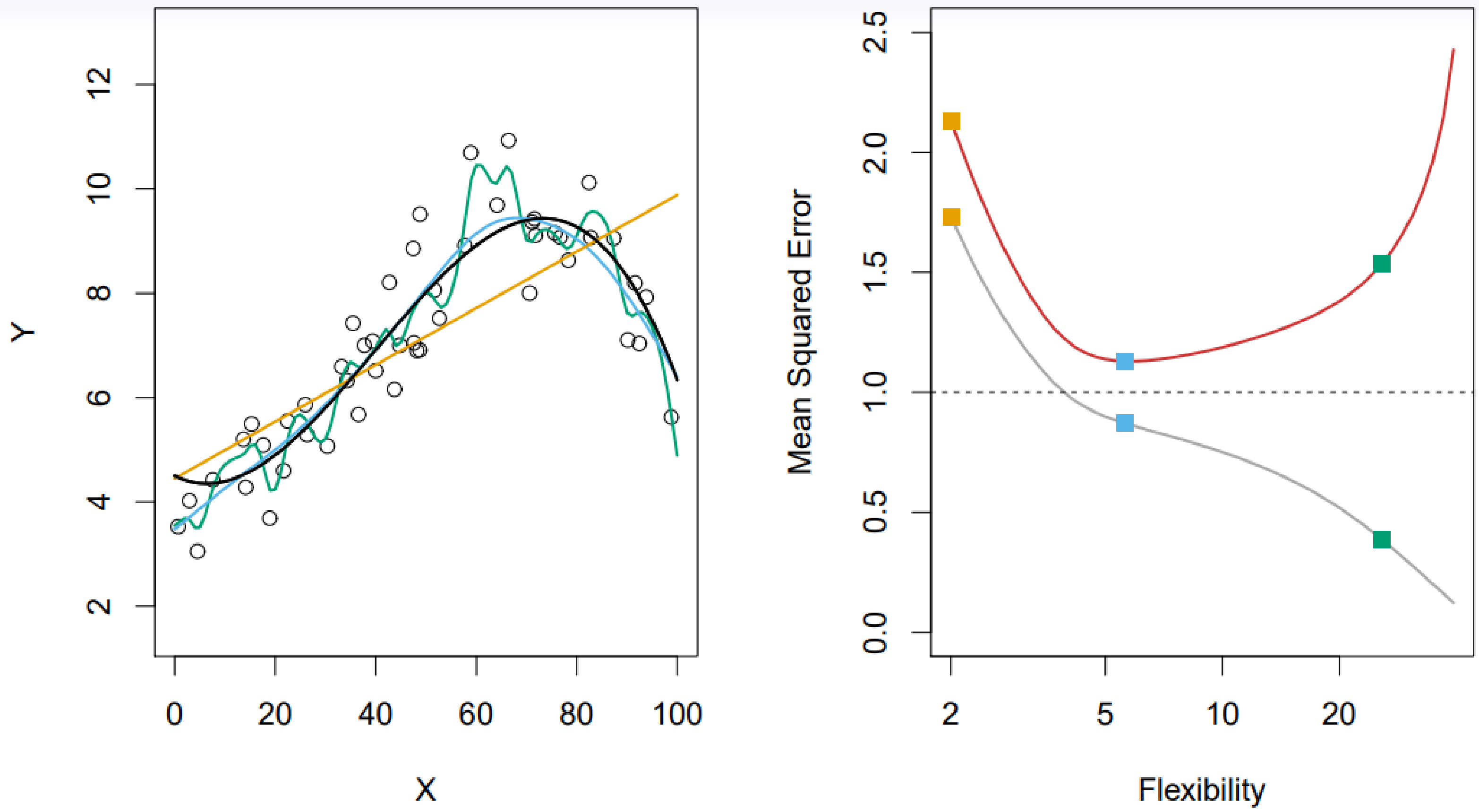


- **Bias**: measures how much the prediction differs from the desired regression function.
- **Variance**: measures how much the predictions for individual data sets vary around their average.

Bias vs Variance Trade-off

https://www.youtube.com/watch?time_continue=116&v=VusKAosxxyk (Up to 4 mins 26 secs)

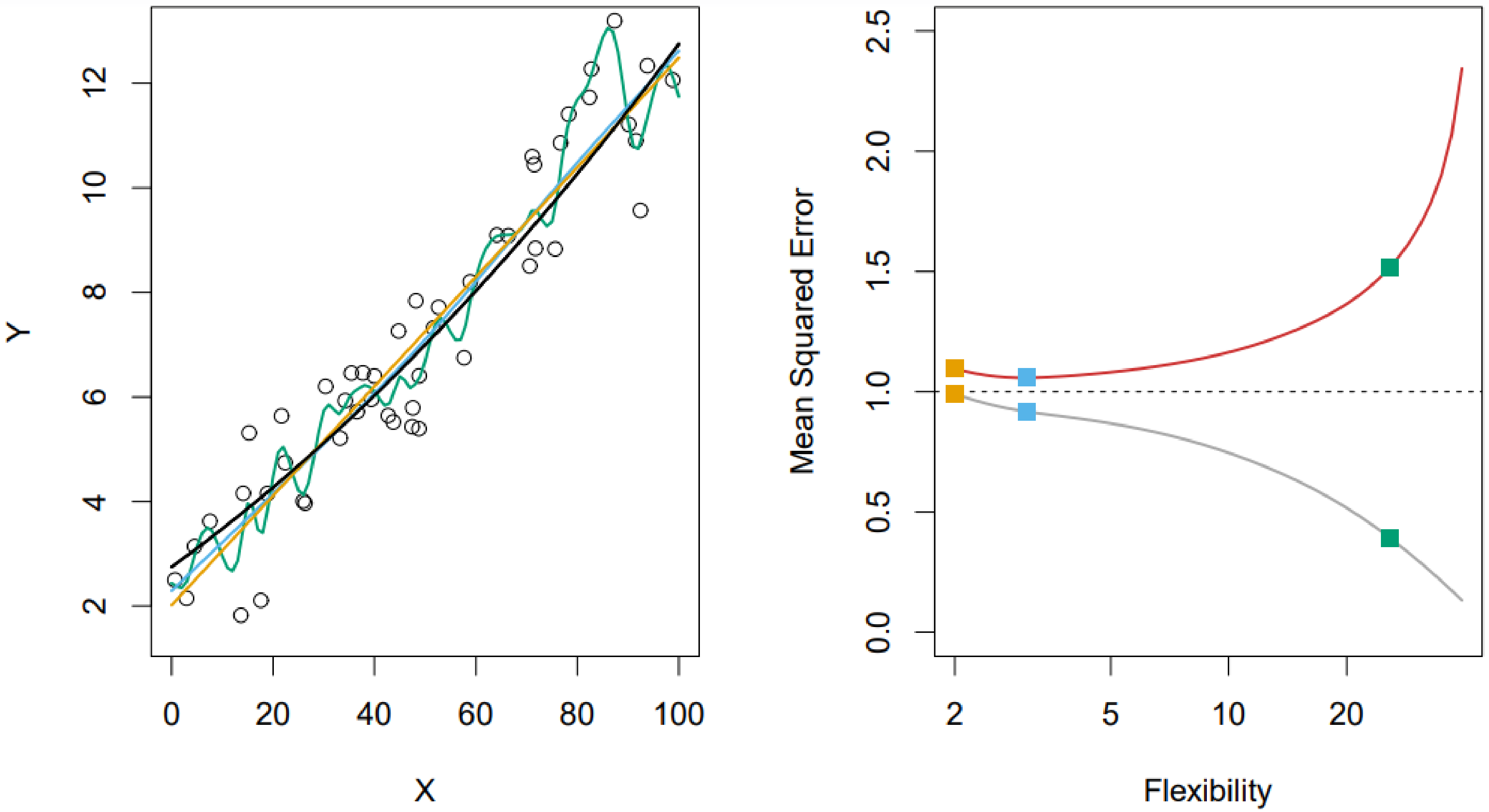
Scenario 1



Bias vs Variance Trade-off

https://www.youtube.com/watch?time_continue=268&v=VusKAosxxyk (Up to 4 mins 56 secs)

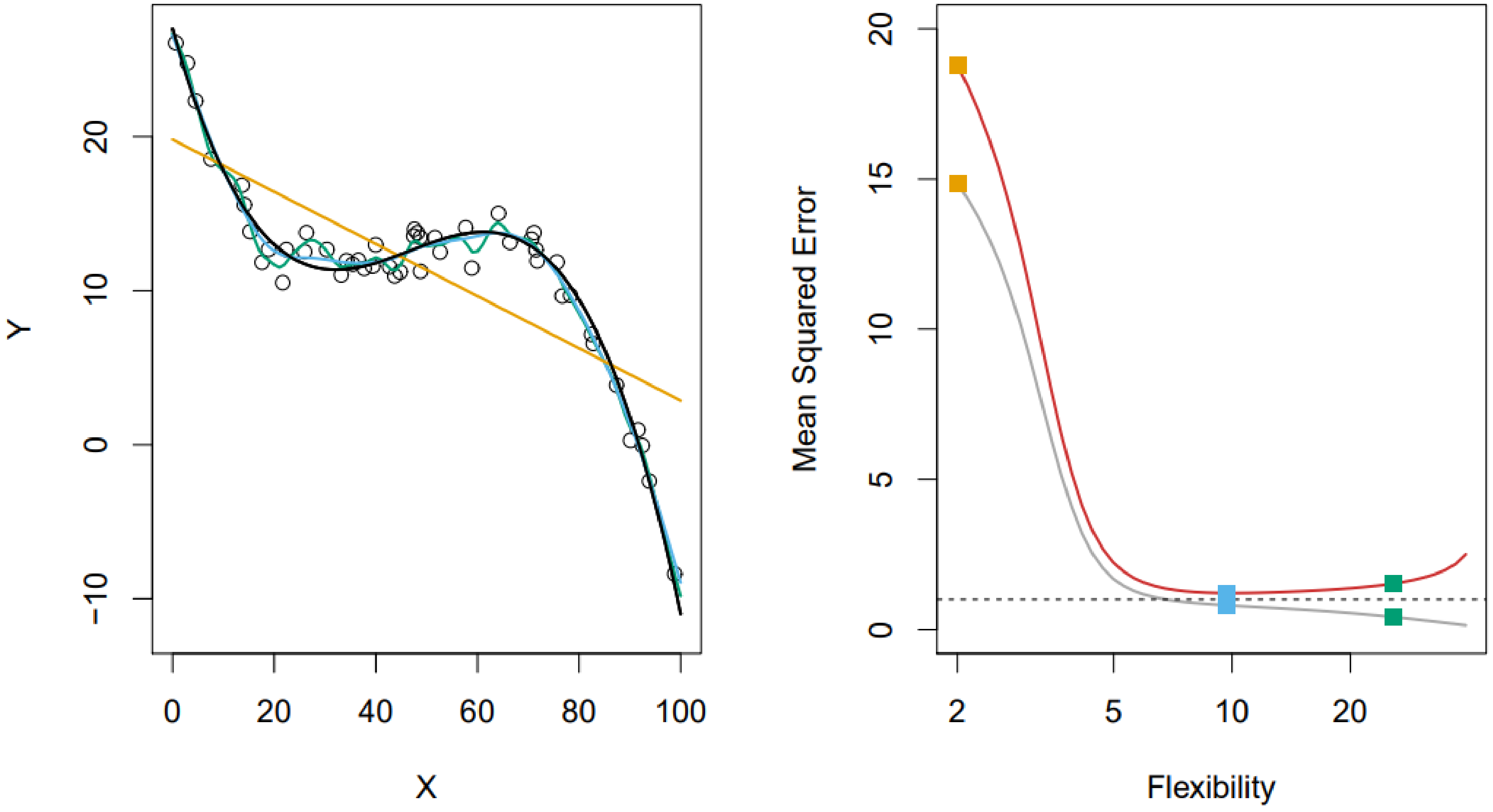
Scenario 2



Bias vs Variance Trade-off

https://www.youtube.com/watch?time_continue=295&v=VusKAosxxyk (Up to 5 min 50 secs)

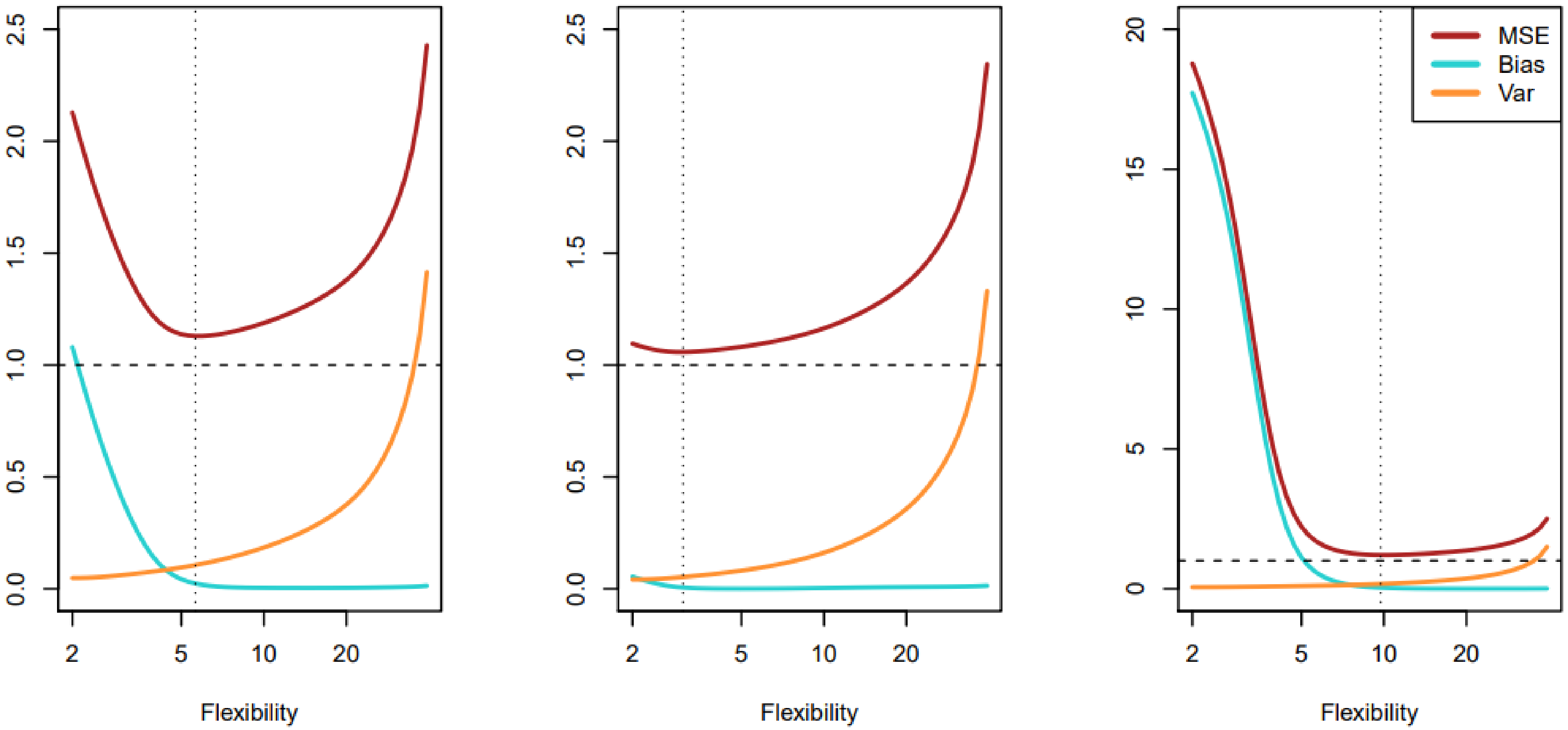
Scenario 3



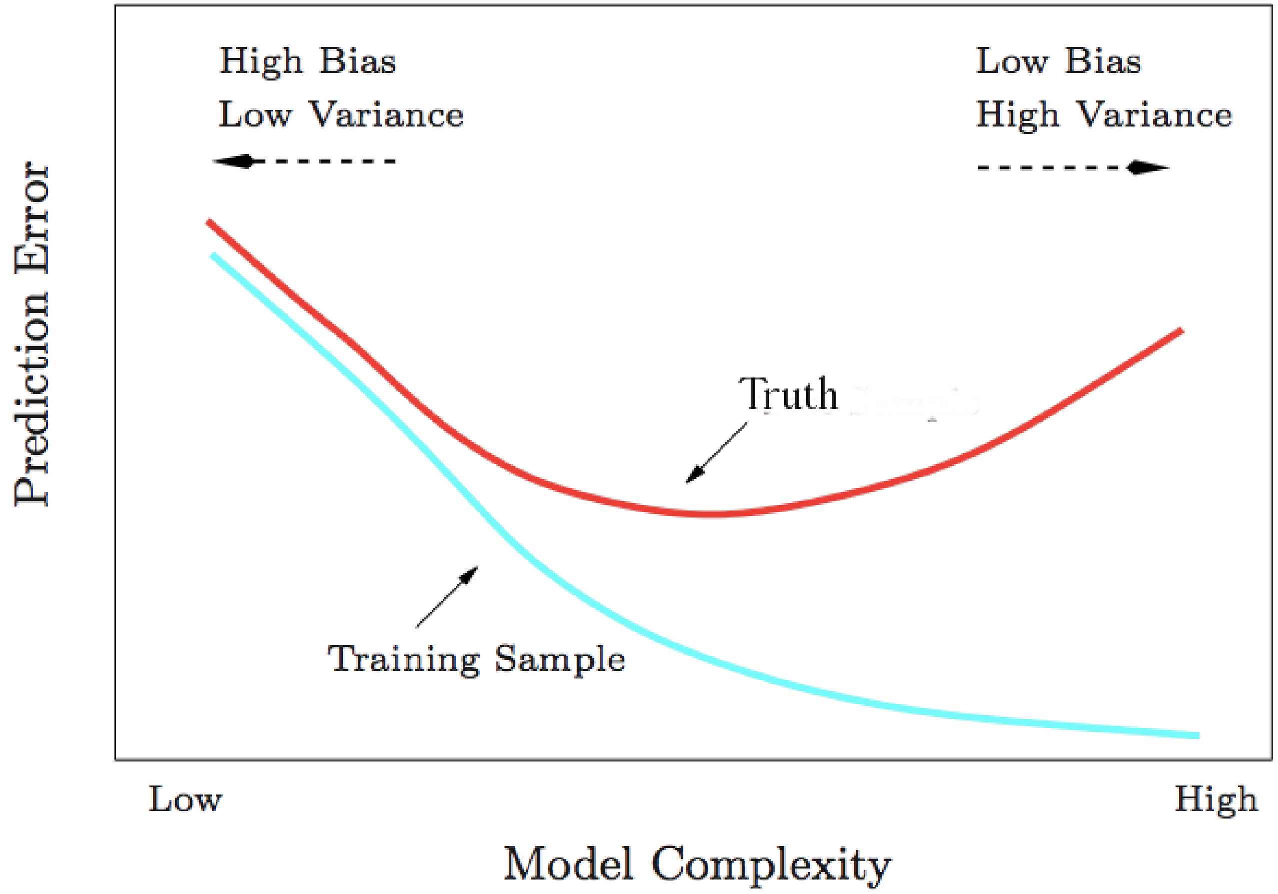
Bias vs Variance Trade-off

https://www.youtube.com/watch?time_continue=521&v=VusKAosxxyk

Optimum Degree



Bias vs Variance Trade-off



No Free Lunch Theorem

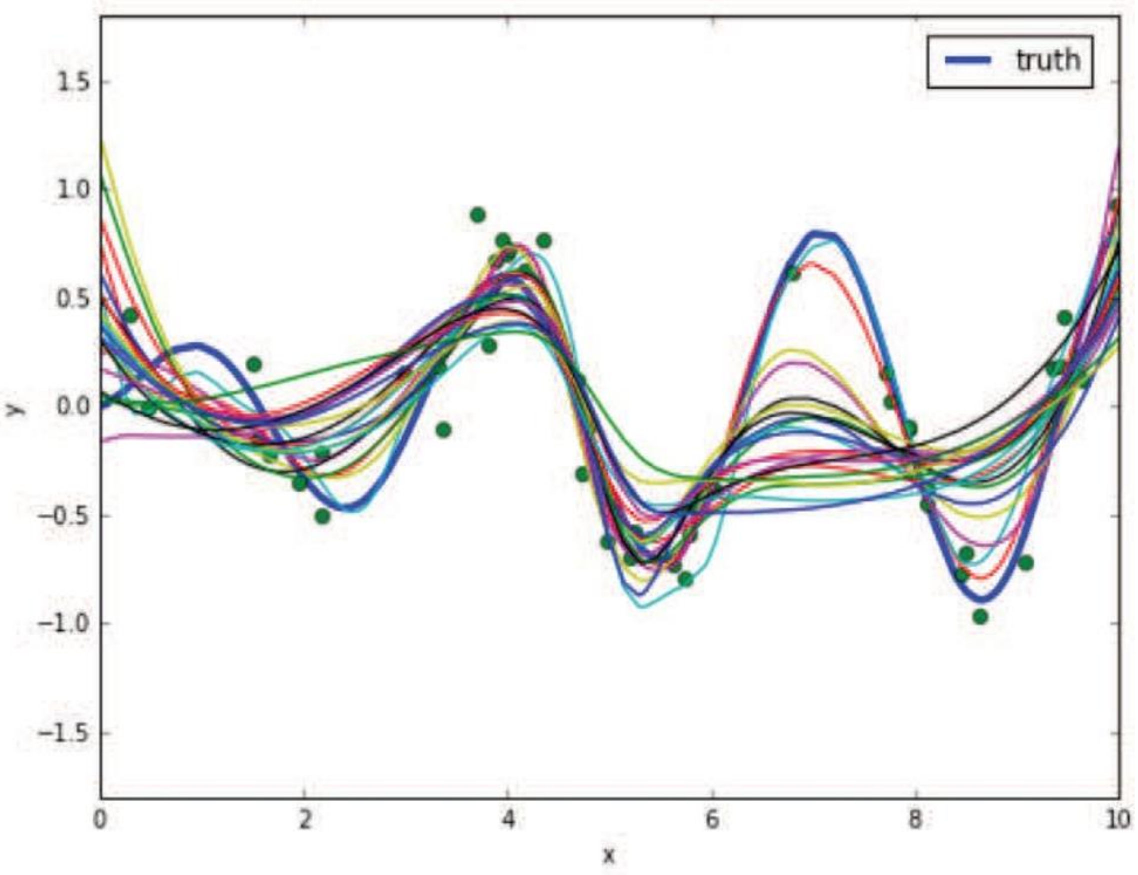
Wolpert and McCready proved:

- if a [learning] algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems

There is no universally good machine learning algorithm (when one has finite data)

- Naive Bayesian classification performs well for text classification with smaller data sets
- Linear Support Vector Machines perform well for text classification

Ensembles



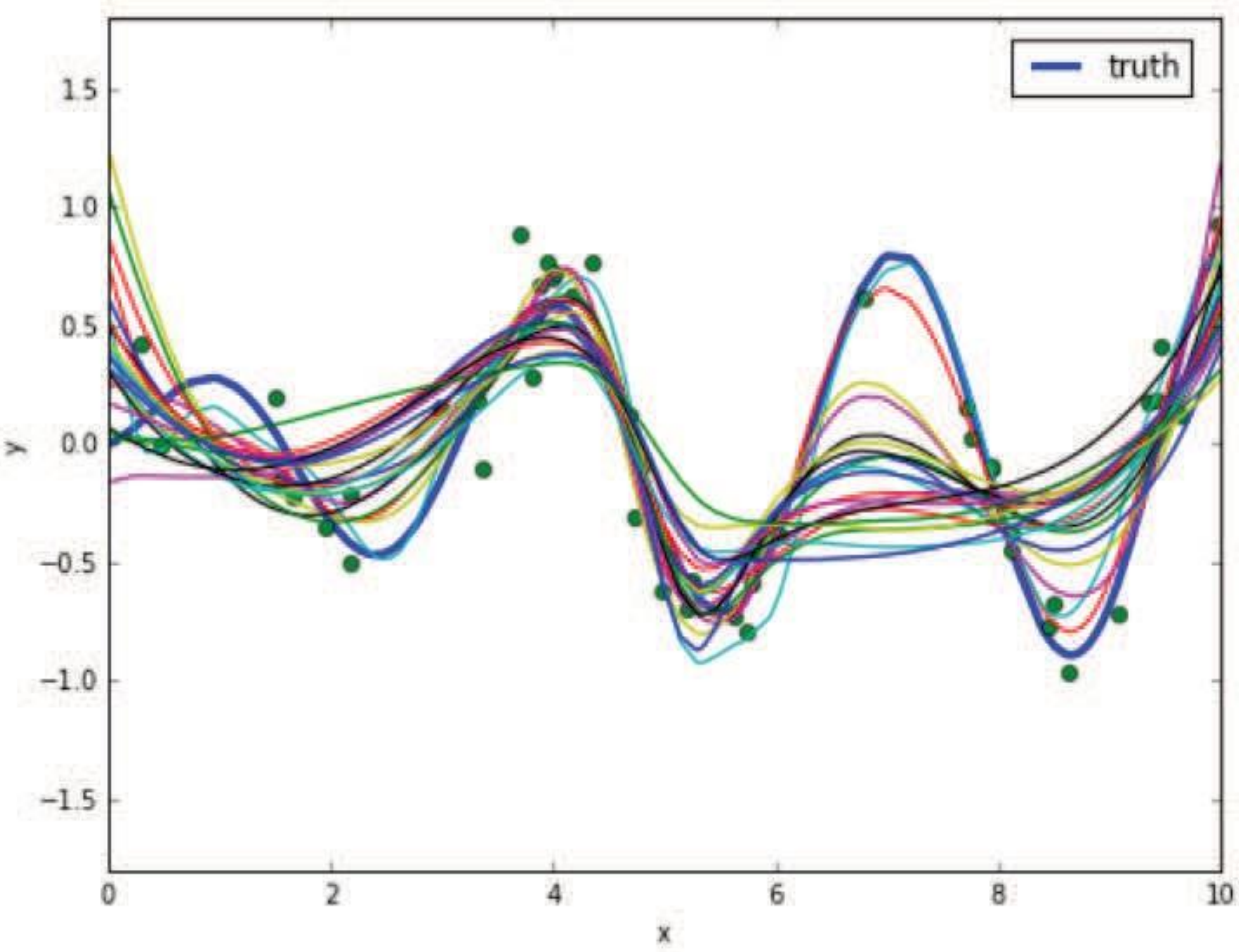
MONASH University

Ensembles

Given only data, we do not know the truth and can only estimate what may be the “truth”

An ensemble is a collection of possible/reasonable models

From this we can understand the **variability** and range of predictions that is realistic

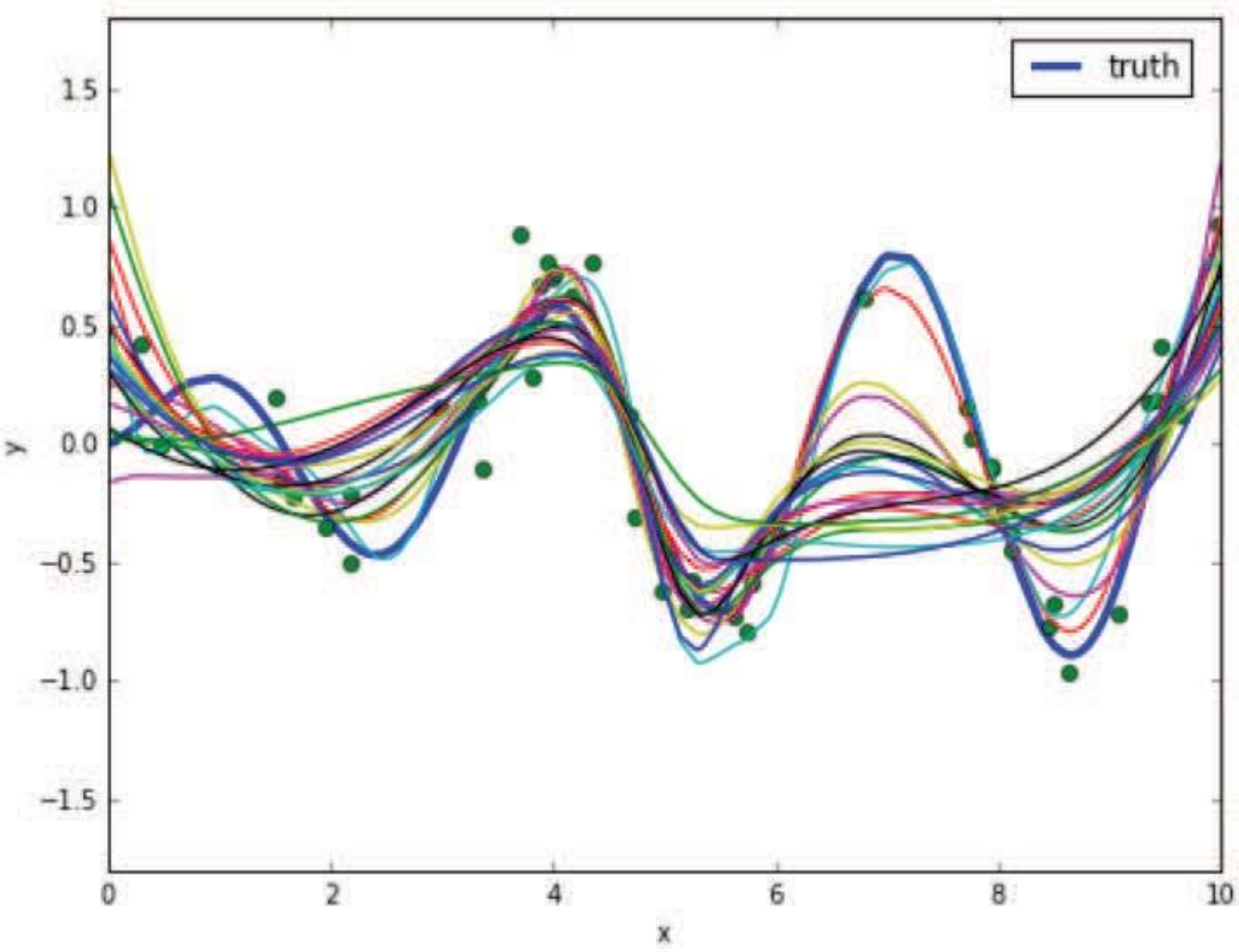


Ensembles

Generating an ensemble is a whole statistical subject in itself

Often we average the predictions over the models in an ensemble to improve performance

$$\hat{y}(x) = \frac{1}{M}\sum_{i=1}^{M} \hat{y}^{(i)}(x)$$

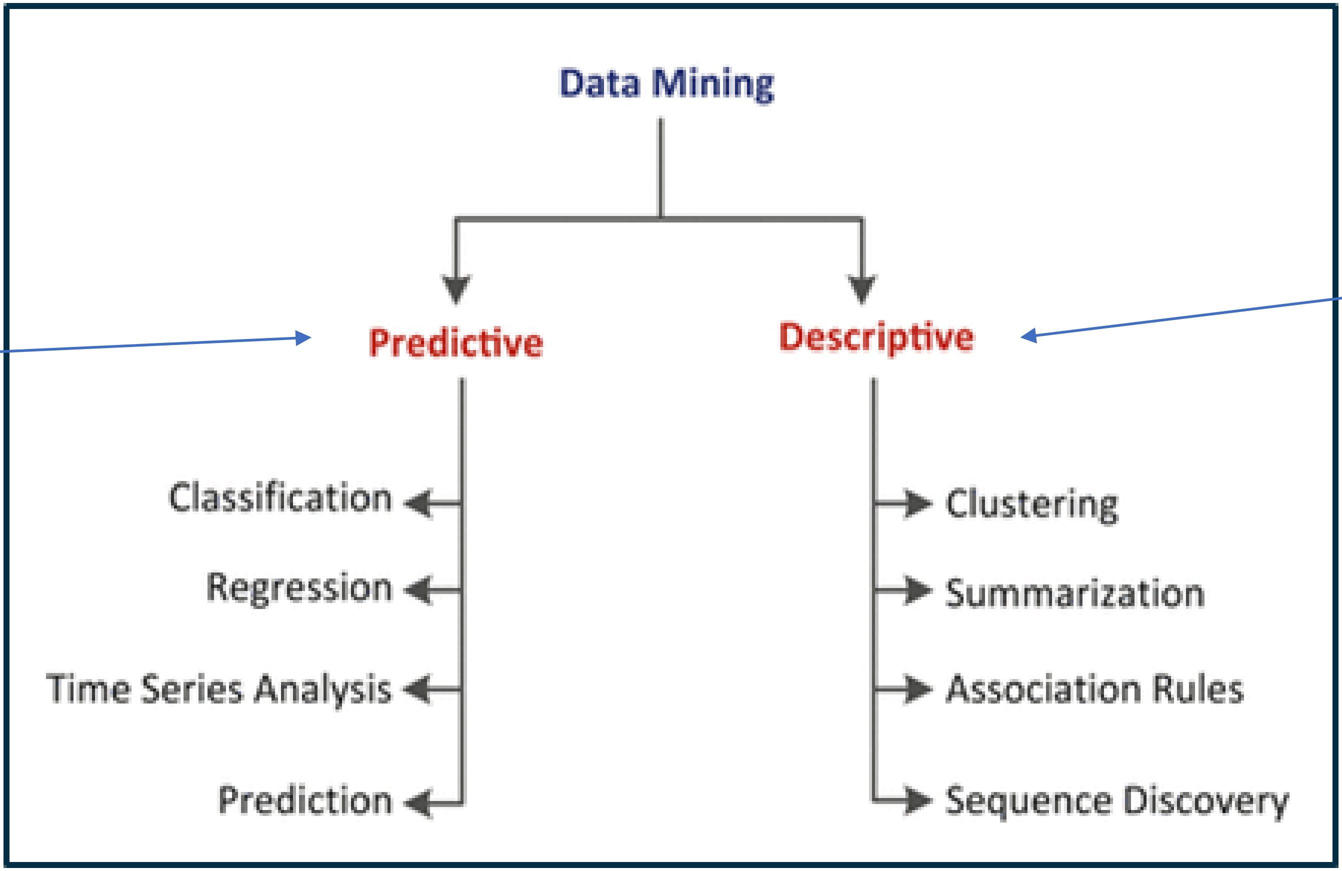


Summary

(So far)

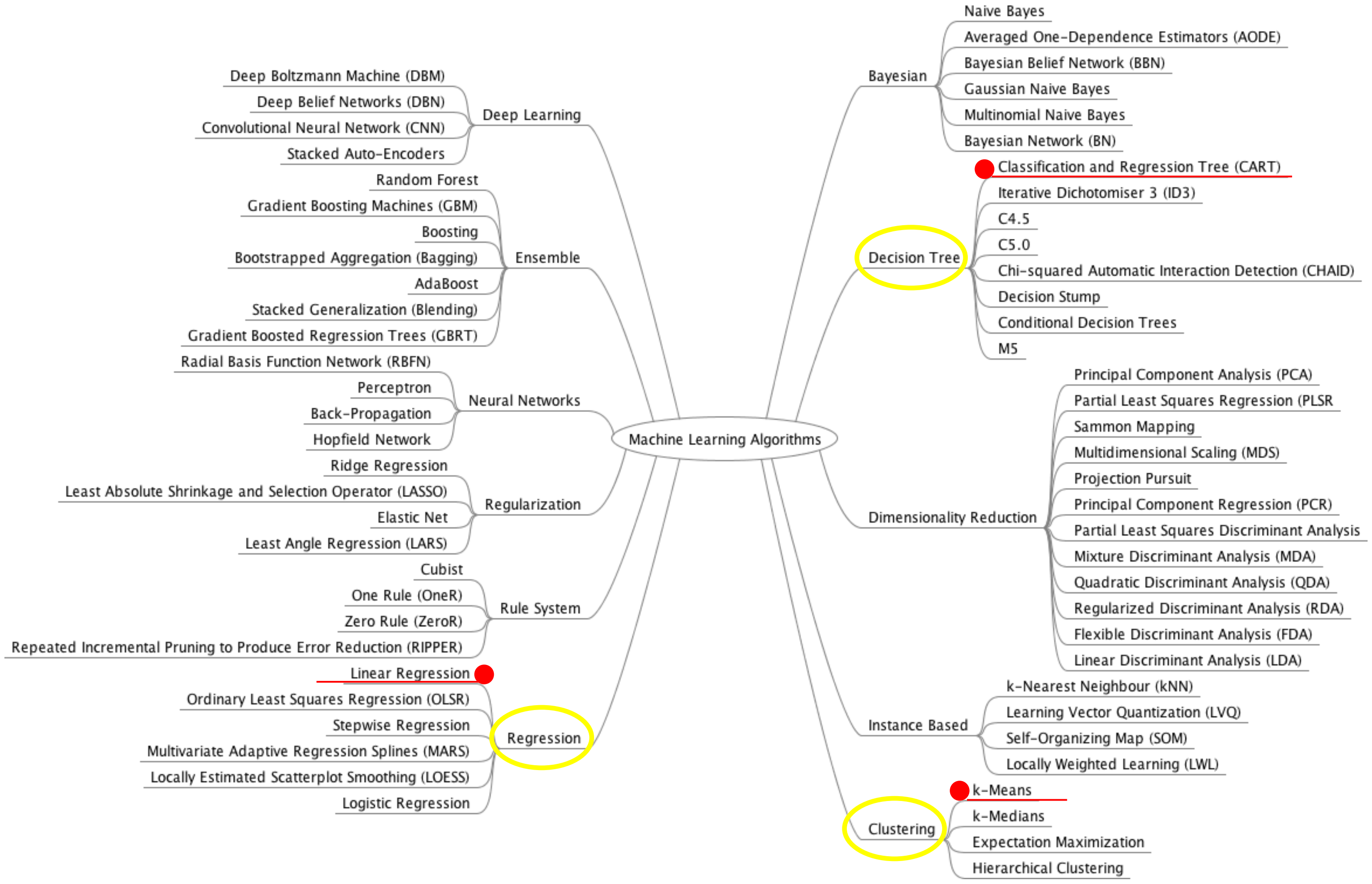
Predictive vs Descriptive

Make prediction using statistical and machine learning techniques



Gain insight from historical data

Machine Learning Algorithms



Recap: Learning Outcomes

Week 6

By the end of this week you should be able to:

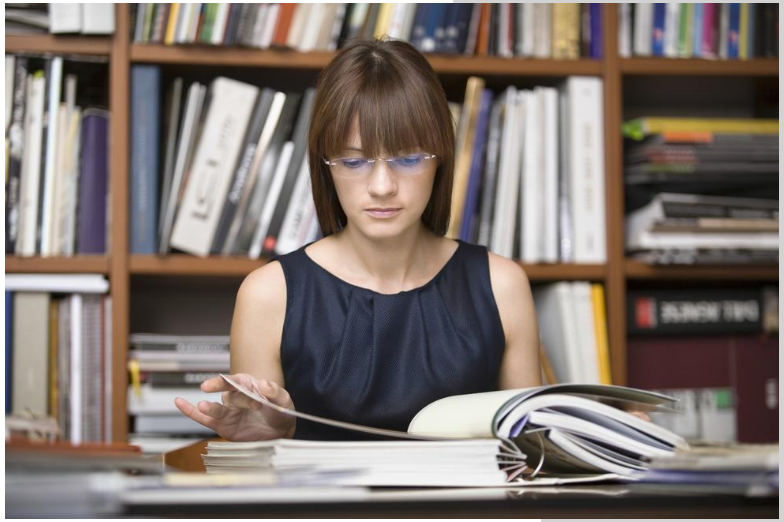
- Fit linear regression and polynomial regression models to a given dataset (in tutorials)
- Explain overfitting and underfitting of different models
- Comprehend bias and variance trade-off
- Comprehend the importance of “No Free Lunch Theorem”
- Explain what ensemble models are

Home Activities

Suggested Activities for the week

Videos

https://www.youtube.com/watch?v=VusKAosxxyk



Articles

Read Alex Guanga, “Machine Learning: Bias vs Variance”, becominghuman.ai, October 2018

Read Sydney Firmin, “There is no free lunch in data science”, KDnuggets, September 2019. Summarized in the paragraph “No Free Lunch for Supervised Machine Learning”.

Tutorials Week 6

Regression in Python, 3 Activities

- Examine the bias of the linear regression model
- Illustrates the fitting of a different type of model
- Illustrates ensembles