Regression

CISC 7026: Introduction to Deep Learning

University of Macau

ML

Many problems in ML can be reduced to **regression** or **classification**

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one
- Is this a dog or muffin?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?

ML

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one

- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?
- What color is this object?

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many
- How much money will I make?
- How much rain will there be tomorrow?
- How far away is this object?

Classification asks which one
- Is this a dog or muffin?
- Will it rain tomorrow? Yes or no?
- What color is this object?

Let us start with regression

Linear Regression

1. Define an example problem

Linear Regression

1. Define an example problem
2. Define our machine learning model f

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

1. **Define an example problem**
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

Task: Given your education, predict your life expectancy

Linear Regression

Task: Given your education, predict your life expectancy

X : Years in school

Linear Regression

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Linear Regression

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y$$

Linear Regression

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y$$

Goal: Given someone's education, you can predict how long they will live

Linear Regression

1. **Define an example problem**
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

1. Define an example problem
2. **Define our machine learning model f**
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

Soon, f will be a deep neural network

Linear Regression

Soon, f will be a deep neural network

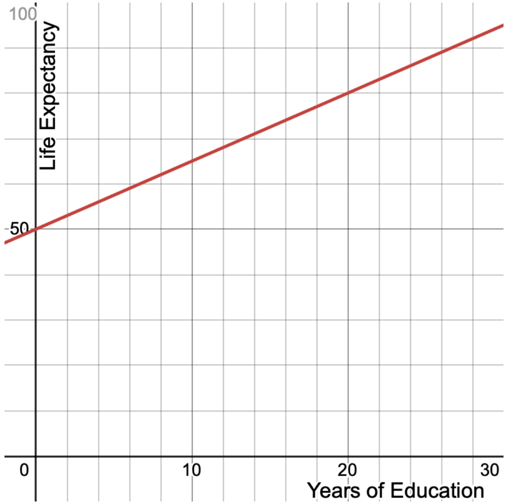
For now, it is easier if we make f a **linear function**

Linear Regression

Soon, f will be a deep neural network

For now, it is easier if we make f a **linear function**

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}\right) = \theta_1 x + \theta_0$$

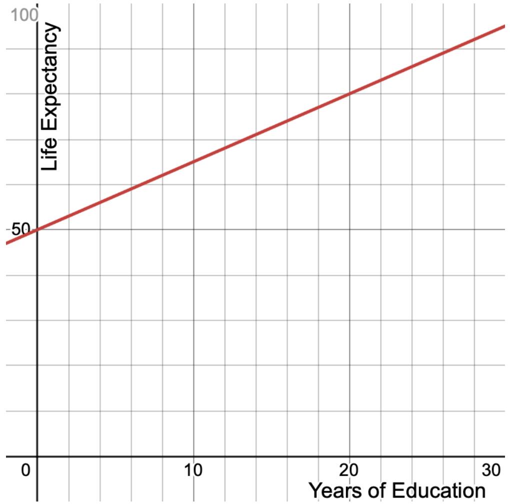


Linear Regression

Soon, f will be a deep neural network

For now, it is easier if we make f a **linear function**

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}\right) = \theta_1 x + \theta_0$$



Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$ that makes $f(x, \boldsymbol{\theta}) = y$

Linear Regression

1. Define an example problem
2. **Define our machine learning model f**
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. **Define a loss function $\mathcal{L}$**
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

Linear Regression

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

Question: How do we find $\boldsymbol{\theta}$? (Hint: We want $f(x, \boldsymbol{\theta}) = y$)

Linear Regression

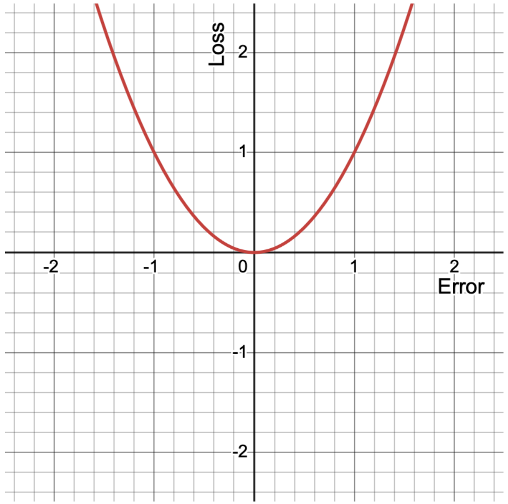
Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

Question: How do we find $\boldsymbol{\theta}$? (Hint: We want $f(x, \boldsymbol{\theta}) = y$)

Answer: We will minimize the **loss** (error) between $f(x, \boldsymbol{\theta})$ and y

E.g.,

$$\min_{\boldsymbol{\theta}} (f(x, \boldsymbol{\theta}) - y)^2 = 0$$



Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

$$\mathcal{L}(x, y, \boldsymbol{\theta})$$

Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

$$\mathcal{L}(x, y, \boldsymbol{\theta})$$

The loss function tells us how close $f(x)$ is to y

Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

$$\mathcal{L}(x, y, \boldsymbol{\theta})$$

The loss function tells us how close $f(x)$ is to y

By **minimizing** the loss function, we make $f(x) = y$

Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

$$\mathcal{L}(x, y, \boldsymbol{\theta})$$

The loss function tells us how close $f(x)$ is to y

By **minimizing** the loss function, we make $f(x) = y$

There are many possible loss functions, but for now we will use the **mean-square error**

Linear Regression

We compute the loss using the **loss function** $\mathcal{L}$

$$\mathcal{L}(x, y, \boldsymbol{\theta})$$

The loss function tells us how close $f(x)$ is to y

By **minimizing** the loss function, we make $f(x) = y$

There are many possible loss functions, but for now we will use the **mean-square error**

$$\text{error}(y, \hat{y}) = (y - \hat{y})^2$$

Linear Regression

Let's derive the error function

Linear Regression

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

Linear Regression

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

Linear Regression

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

$$(f(x, \boldsymbol{\theta}) - y)^2 = 0$$

Square for minimization

Linear Regression

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

$$(f(x, \boldsymbol{\theta}) - y)^2 = 0$$

Square for minimization

$$\text{error}(f(x, \boldsymbol{\theta}), y) = (f(x, \boldsymbol{\theta}) - y)^2$$

Linear Regression

Let's derive the error function

$$f(x, \boldsymbol{\theta}) = y$$

f(x) should predict y

$$f(x, \boldsymbol{\theta}) - y = 0$$

Move y to LHS

$$(f(x, \boldsymbol{\theta}) - y)^2 = 0$$

Square for minimization

$$\text{error}(f(x, \boldsymbol{\theta}), y) = (f(x, \boldsymbol{\theta}) - y)^2$$

Linear Regression

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Linear Regression

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

By minimizing $\mathcal{L}$, we can find the parameters $\boldsymbol{\theta}$

Linear Regression

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

By minimizing $\mathcal{L}$, we can find the parameters $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Linear Regression

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

By minimizing $\mathcal{L}$, we can find the parameters $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: Any issues with $\mathcal{L}$?

Linear Regression

We can write the loss function for a single datapoint x_i, y_i as

$$\mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

By minimizing $\mathcal{L}$, we can find the parameters $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Question: Any issues with $\mathcal{L}$?

Answer: We only consider a single datapoint! We want to learn $\boldsymbol{\theta}$ for the entire dataset

Linear Regression

For a single x_i, y_i:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Linear Regression

For a single x_i, y_i:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \operatorname{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

For the entire dataset:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \operatorname{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Linear Regression

For a single x_i, y_i:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

For the entire dataset:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Minimizing this loss function will give us the optimal parameters!

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. **Define a loss function $\mathcal{L}$**
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. **Use $\mathcal{L}$ to learn the parameters θ of f**
5. Investigate the results

Linear Regression

Question: How do we minimize:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Linear Regression

Question: How do we minimize:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(x_i, y_i, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \text{error}(f(x_i, \boldsymbol{\theta}), y_i) = \min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (f(x_i, \boldsymbol{\theta}) - y_i)^2$$

Answer: For now, magic! We need more knowledge before we can derive this.

Linear Regression

First, construct a **design matrix** $\boldsymbol{X}$ containing input data x and a constant 1 for the bias. Also construct a $\boldsymbol{y}$ vector!

$$\boldsymbol{X} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

And remember the parameters $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix},$$

$$\boldsymbol{\theta} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

The solution to linear regression

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. **Use $\mathcal{L}$ to learn the parameters θ of f**
5. Investigate the results

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. **Investigate the results**

Example

Back to the example...

Example

Back to the example...

Task: Given your education, predict your life expectancy

Example

Back to the example...

Task: Given your education, predict your life expectancy

X : Years in school

Example

Back to the example...

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Example

Back to the example...

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Goal: Learn the parameters $\boldsymbol{\theta}$ such that

$$f(x, \boldsymbol{\theta}) = y$$

Example

Back to the example...

Task: Given your education, predict your life expectancy

X : Years in school

Y : Age of death

Goal: Learn the parameters $\boldsymbol{\theta}$ such that

$$f(x, \boldsymbol{\theta}) = y$$

You will be doing this in your first assignment!

Example

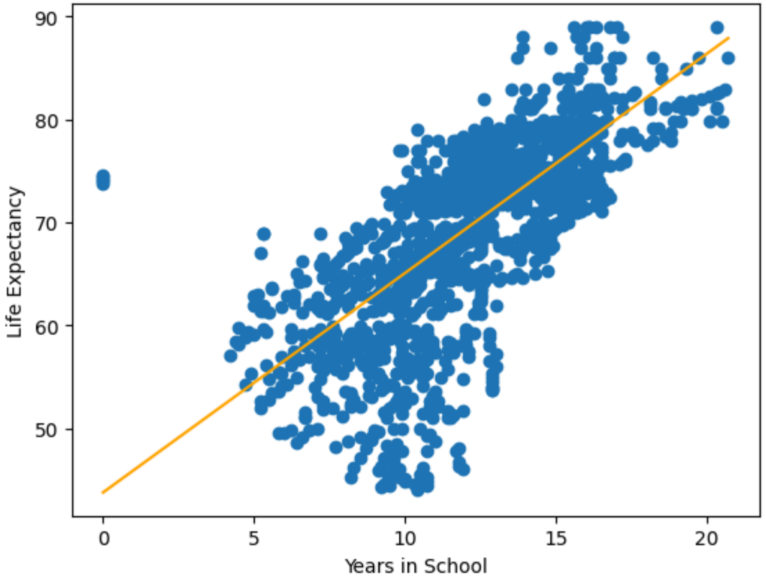
Back to the example...

Task: Given your education, predict your life expectancy

Example

Back to the example...

Task: Given your education, predict your life expectancy



Tips for assignment 1

Tips for assignment 1

```python
def f(theta, design):
  # Linear function
  return theta @ design
```

Tips for assignment 1

```python
def f(theta, design):
  # Linear function
  return theta @ design
```

Not all matrices can be inverted! Ensure the matrices are square and the condition number is low

```python
A.shape
cond = jax.numpy.linalg.cond(A)
```

Linear Regression

1. Define an example problem

Linear Regression

1. Define an example problem
2. Define our machine learning model f

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f

Linear Regression

1. Define an example problem
2. Define our machine learning model f
3. Define a loss function $\mathcal{L}$
4. Use $\mathcal{L}$ to learn the parameters θ of f
5. Investigate the results

Relax

We figured out linear regression!

We figured out linear regression!

- Outliers

We figured out linear regression!

- Outliers
- Can we go beyond linear?

We figured out linear regression!

- Outliers
- Can we go beyond linear?
- Overfitting

We figured out linear regression!

- Outliers
- Can we go beyond linear?
- Overfitting
- Test and train splits

We figured out linear regression!

- Outliers
- Can we go beyond linear?
- Overfitting
- Test and train splits

Example

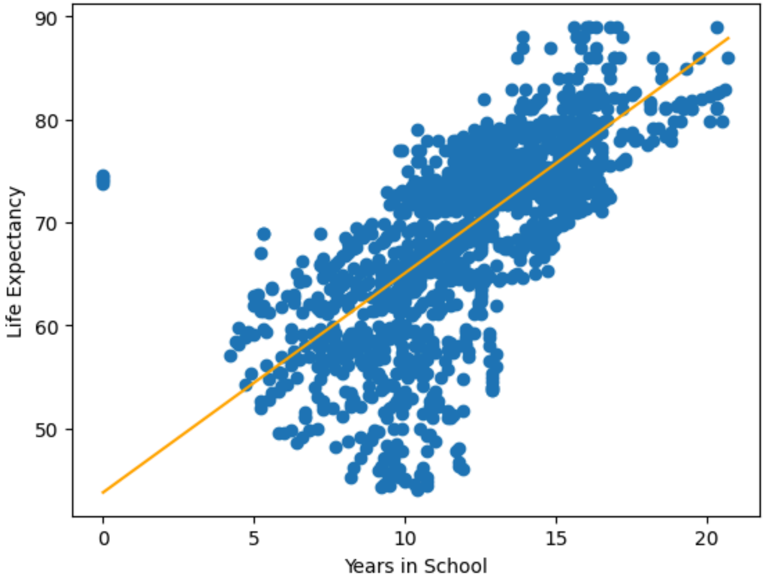
Back to the example...

Task: Given your education, predict your life expectancy

Example

Back to the example...

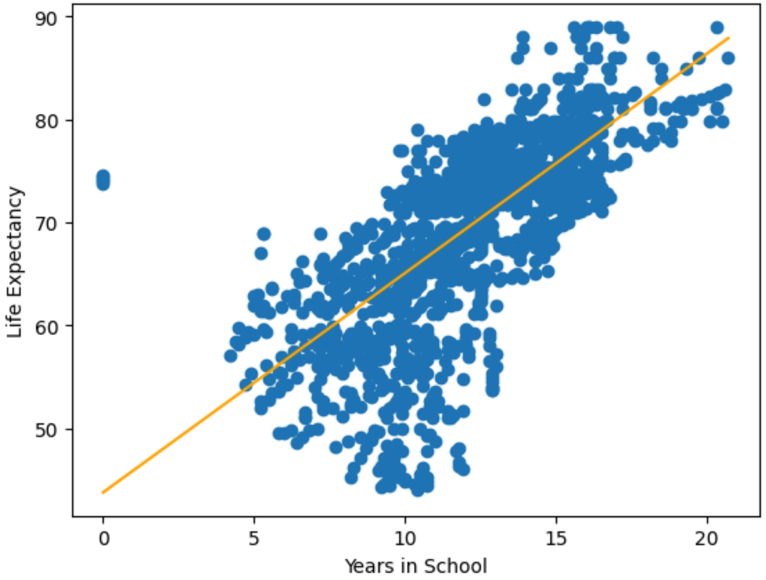
Task: Given your education, predict your life expectancy



Example

Back to the example...

Task: Given your education, predict your life expectancy



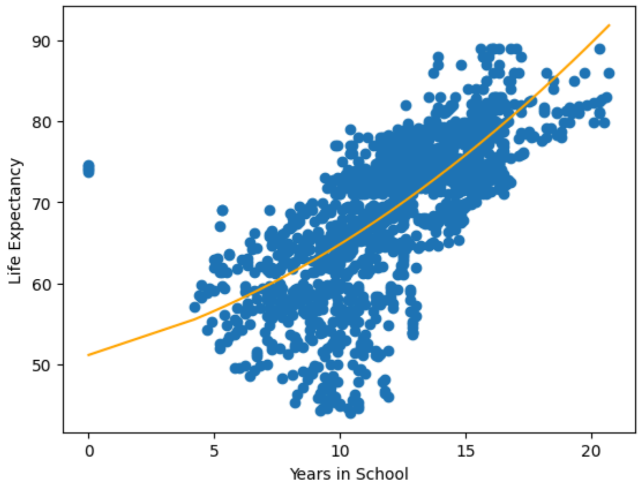
Could we do better than a linear function f?

Example

Could we do better than a linear function f?

Example

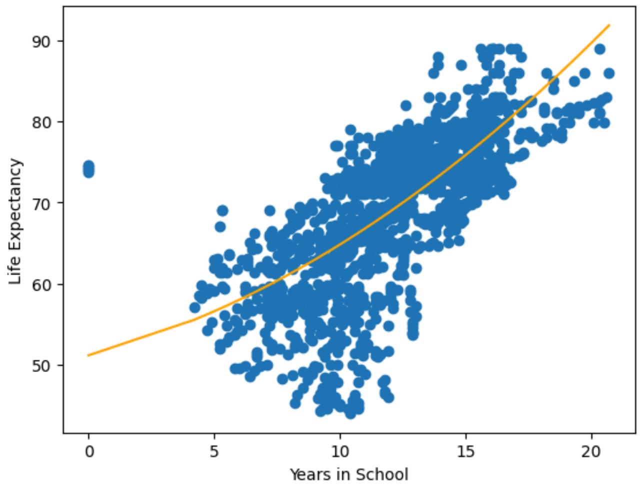
Could we do better than a linear function f?



What if we used a polynomial instead?

Example

Could we do better than a linear function f?



What if we used a polynomial instead?

$$f(x) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Example

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \right) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Example

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \right) = \theta_n x^n + \theta_{n-1} x^{n-1}, \dots, \theta_1 + x^1 + \theta_0$$

$$f(x, \boldsymbol{\theta}) = \begin{bmatrix} \theta_n & \theta_{n-1} & \dots & \theta_1 & b \end{bmatrix} \begin{bmatrix} x^n \\ x^{n-1} \\ \vdots \\ x^1 \\ 1 \end{bmatrix}$$

Example

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 x^1 + \theta_0$$

Example

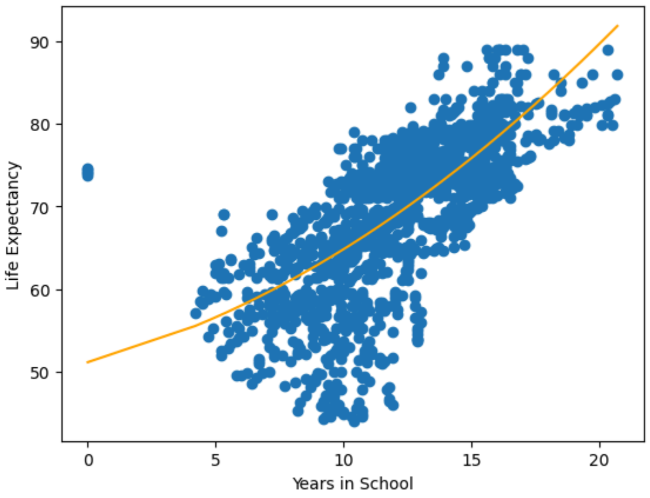
$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 x^1 + \theta_0$$

How do we choose n? Let us try different n

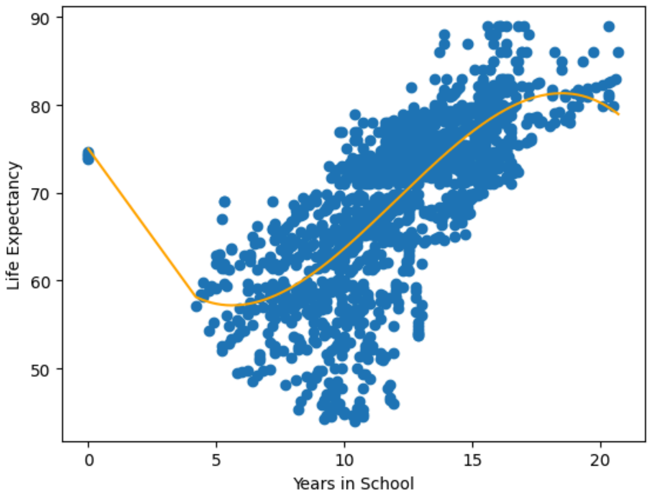
Example

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 x^1 + \theta_0$$

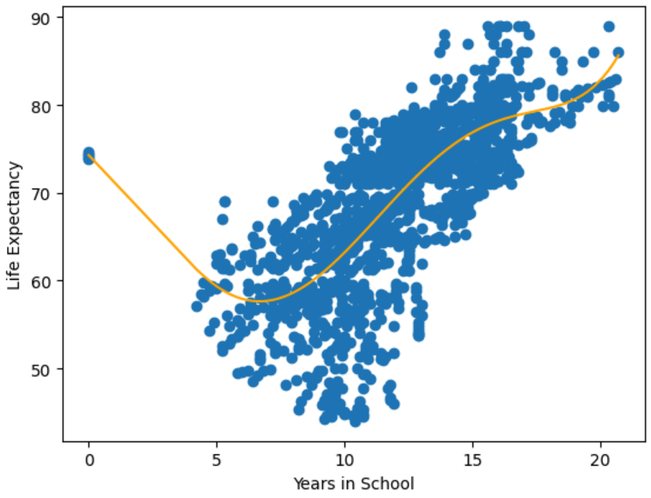
How do we choose n? Let us try different n



$n = 2$



$n = 3$



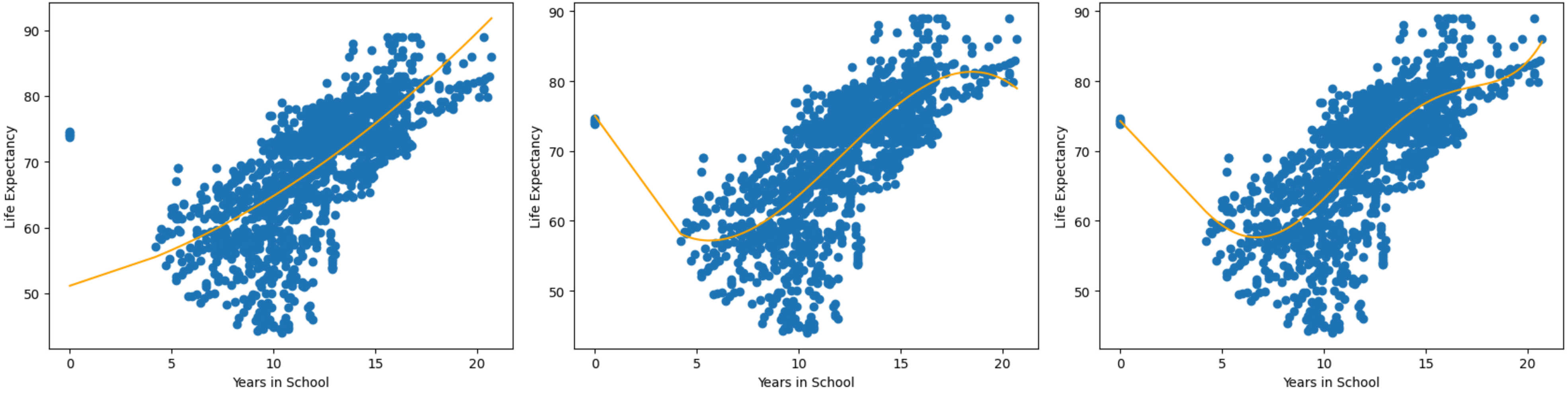
$n = 5$

Example

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$

Example

$$f(x, \boldsymbol{\theta}) = \theta_n x^n + \theta_{n-1} x^{n-1}, ..., \theta_1 + x^1 + \theta_0$$



$n = 2$ $\qquad$ $n = 3$ $\qquad$ $n = 5$

Question: Which n should we pick? Why?

Example

Data is inherently noisy

Example

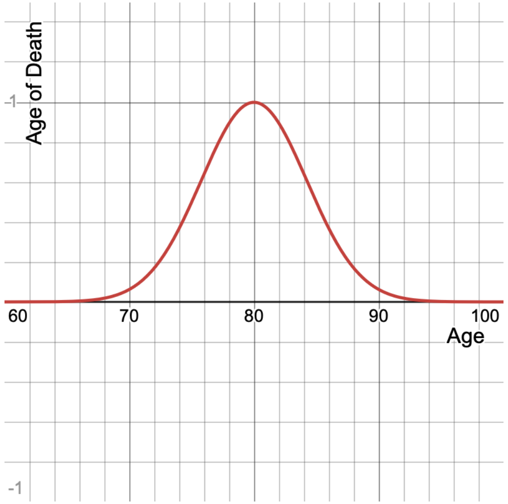
Data is inherently noisy

The world is governed by random processes

Example

Data is inherently noisy

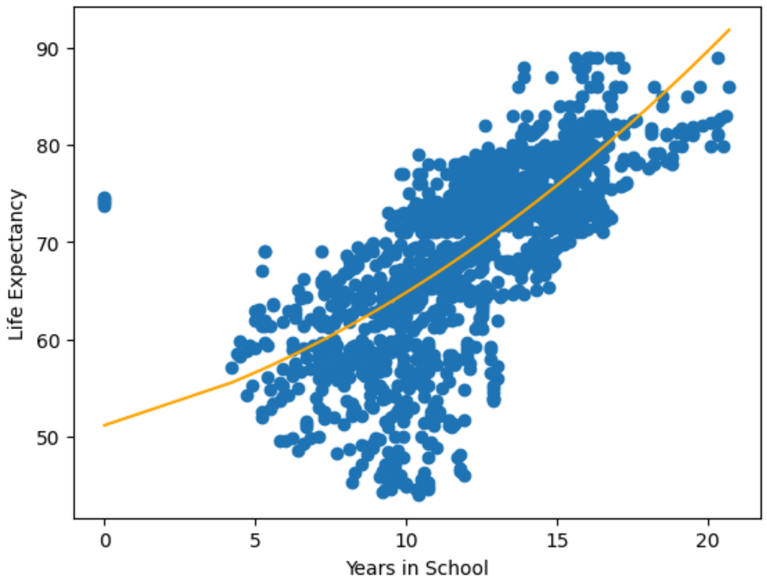
The world is governed by random processes



,

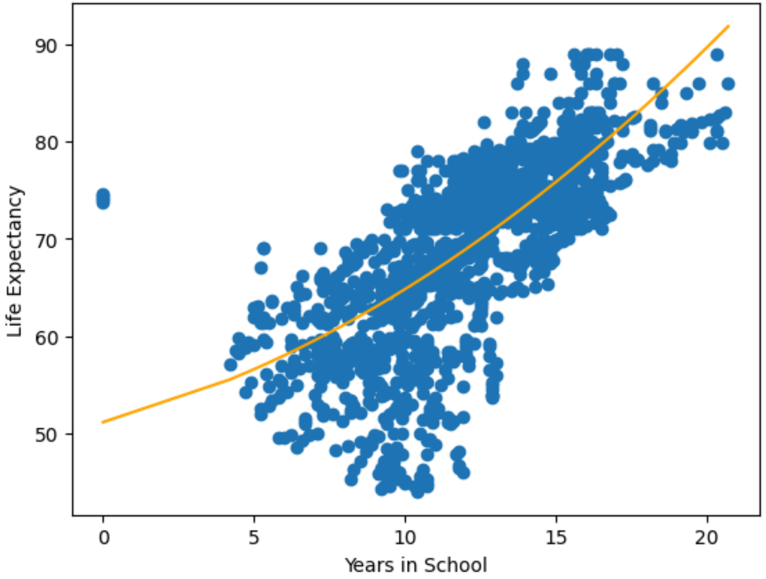
Example

This is just an estimate



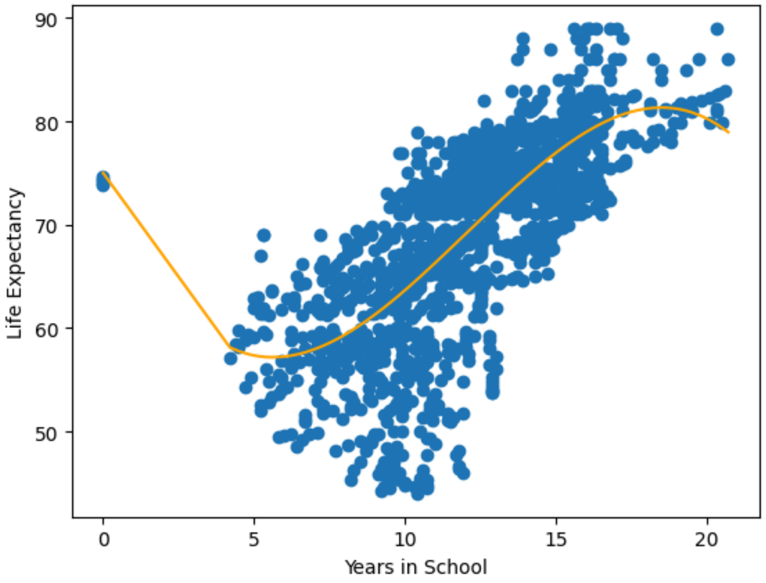
Example

This is just an estimate

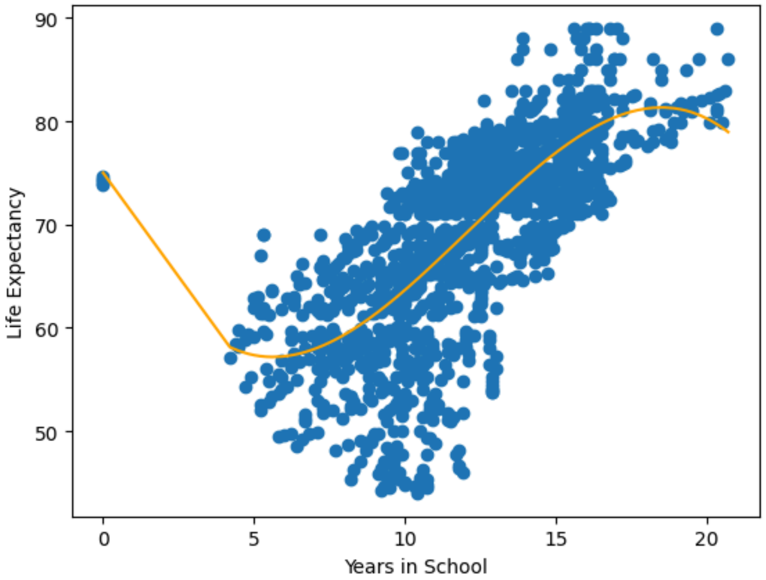


Going to school for 20 years will not save you from a hungry bear

Example

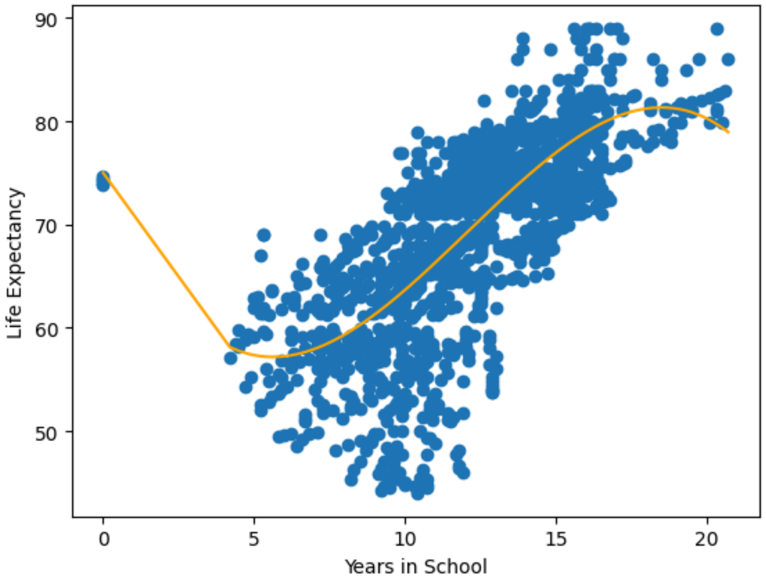


Example



When we fit to noise instead of the trend, we call it **overfitting**

Example



When we fit to noise instead of the trend, we call it **overfitting**

Overfitting is bad because our predictions will be inaccurate

Example

How can we measure overfitting?

Example

How can we measure overfitting?

Learn our parameters from one subset of data: **training dataset**

Example

How can we measure overfitting?

Learn our parameters from one subset of data: **training dataset**

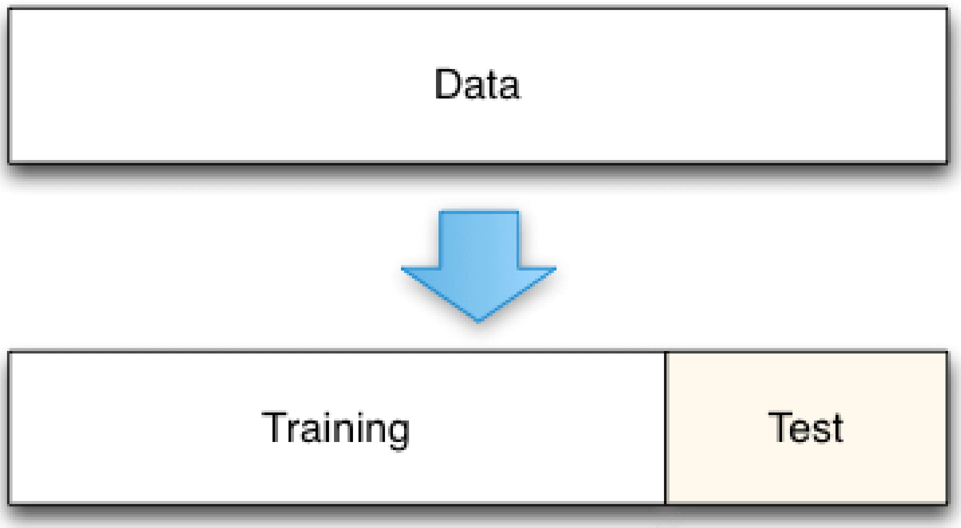
Test our model on a different subset of data: **testing dataset**

Example

How can we measure overfitting?

Learn our parameters from one subset of data: **training dataset**

Test our model on a different subset of data: **testing dataset**



Example

Question: How do we choose the training and testing datasets?

Example

Question: How do we choose the training and testing datasets?

$$\mathcal{D}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathcal{D}_{\text{train}} = \begin{bmatrix} x_4 \\ x_1 \\ x_3 \end{bmatrix}$$

$$\mathcal{D}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix} \qquad \mathcal{D}_{\text{test}} = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}$$

Answer: Always shuffle the data

Example

Question: How do we choose the training and testing datasets?

$$\mathcal{D}_{\text{train}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathcal{D}_{\text{train}} = \begin{bmatrix} x_4 \\ x_1 \\ x_3 \end{bmatrix}$$

$$\mathcal{D}_{\text{test}} = \begin{bmatrix} x_4 \\ x_5 \end{bmatrix} \qquad \mathcal{D}_{\text{test}} = \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}$$

Answer: Always shuffle the data

ML relies on the **Independent and Identically Distributed (IID)** assumption

Example

- Overfitting
- Outliers
- Regularization
- Etc