

Learning Simple Functions I

Fundamentals of Artificial Intelligence

Fabien Cromieres

Kyoto University

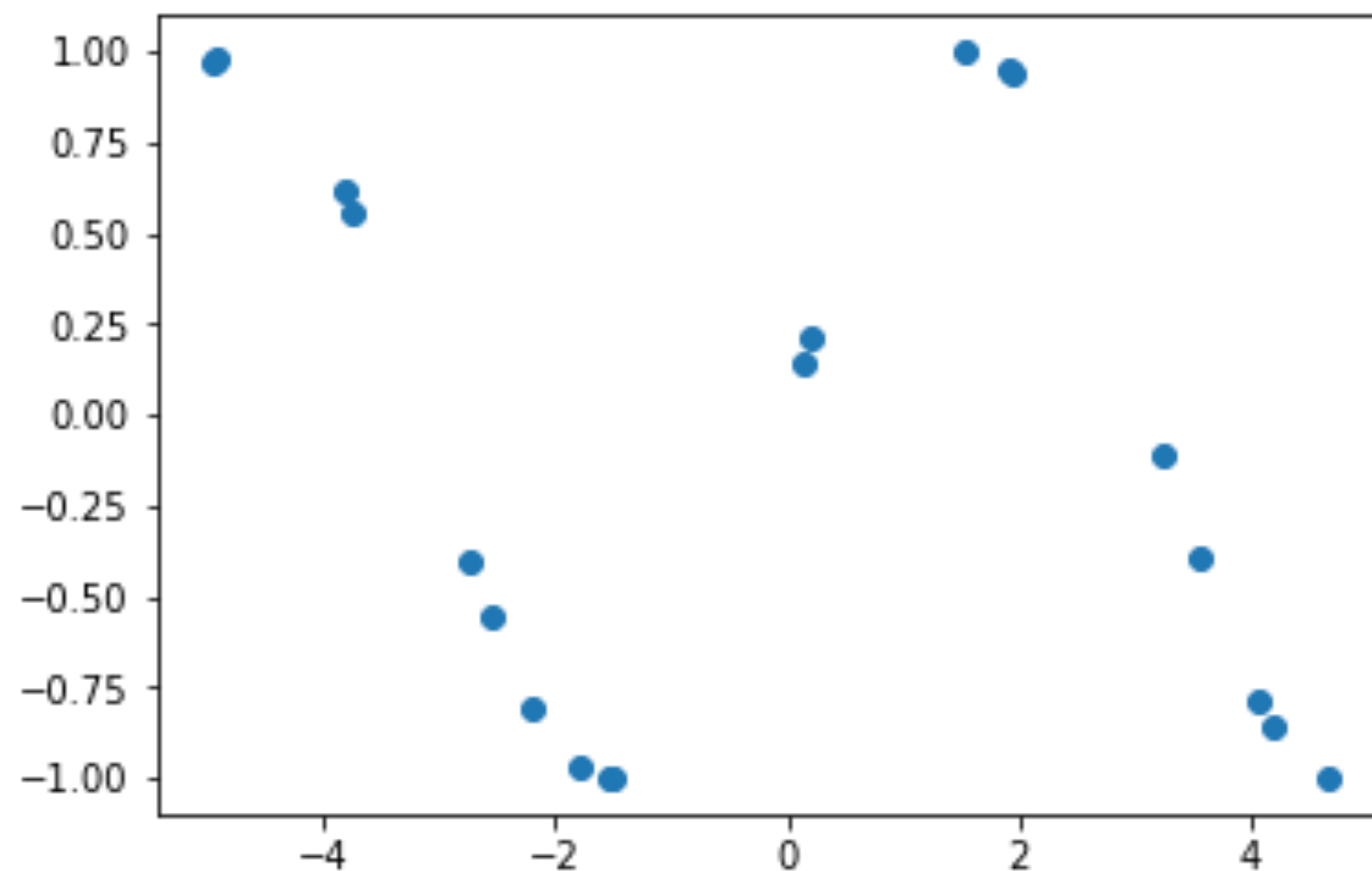
<http://lotus.kuee.kyoto-u.ac.jp/~fabien/lectures/IA/>

Learning a function from examples

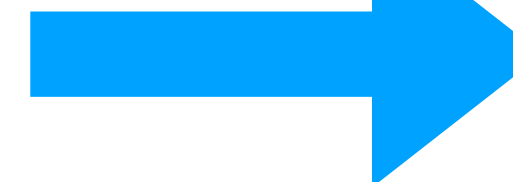
- Main idea today: Learning a function from examples

We are given example values of $f(x)$ for some x

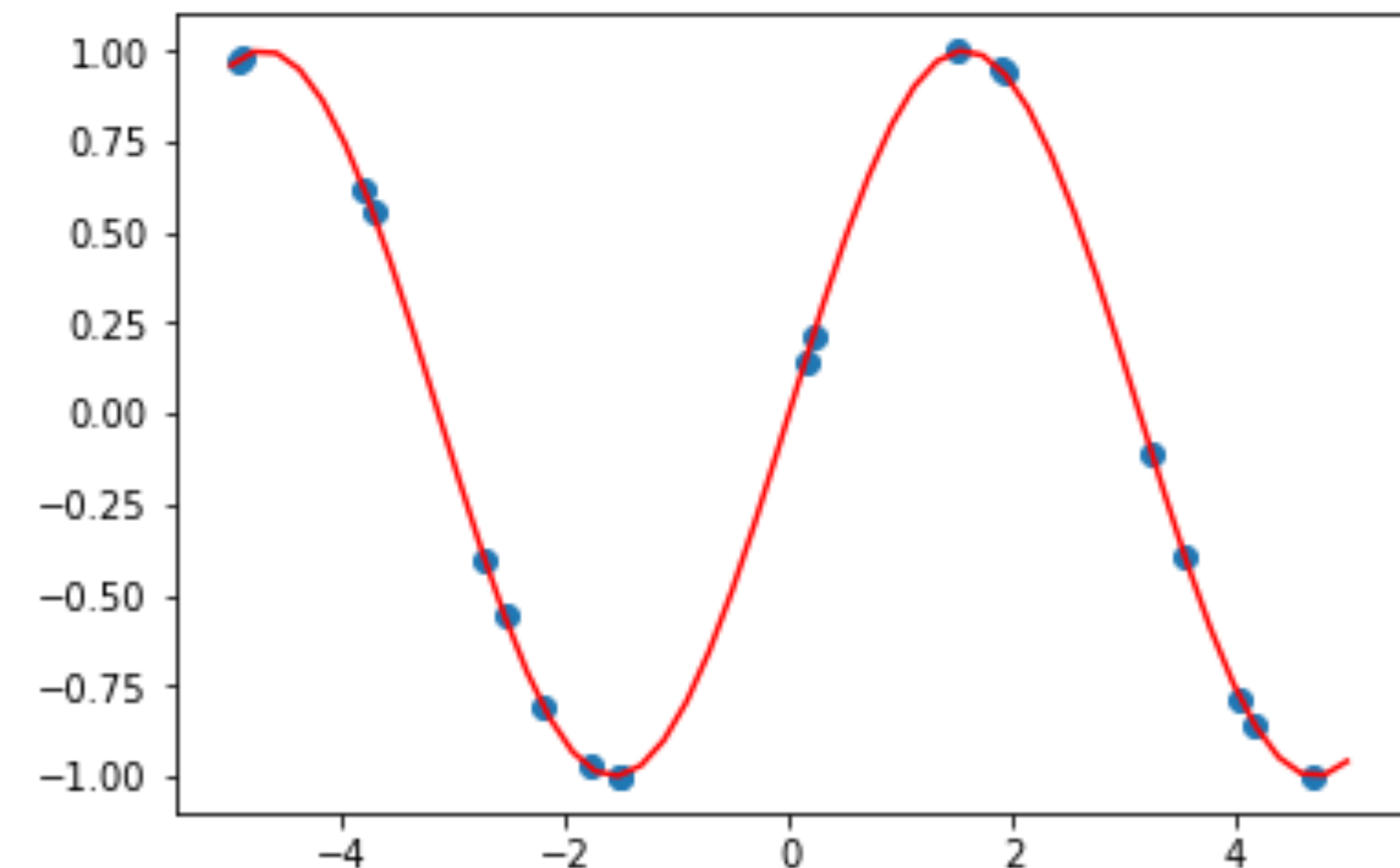
x	f(x)
-3.324820	0.182204
1.294681	0.962122
-4.141596	0.841473
2.928655	0.211333
-2.374387	-0.694126
4.555678	-0.987746
.....	



Learning



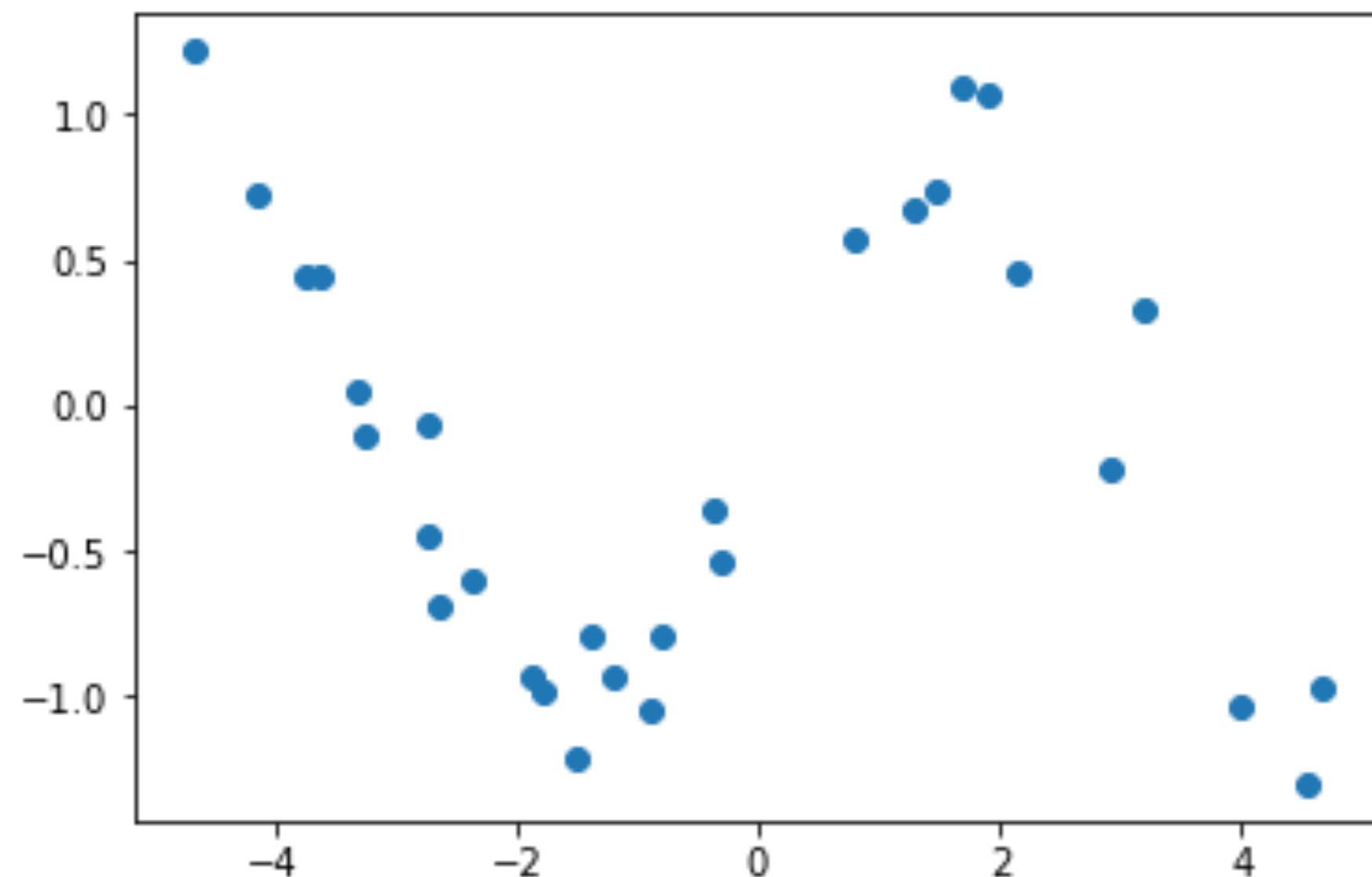
What would be the value of the function for all x ?



Learning a function from examples

- We might not always want that the function we learn match exactly the examples
- In practice, we might have to consider “imperfect” or “noisy” examples

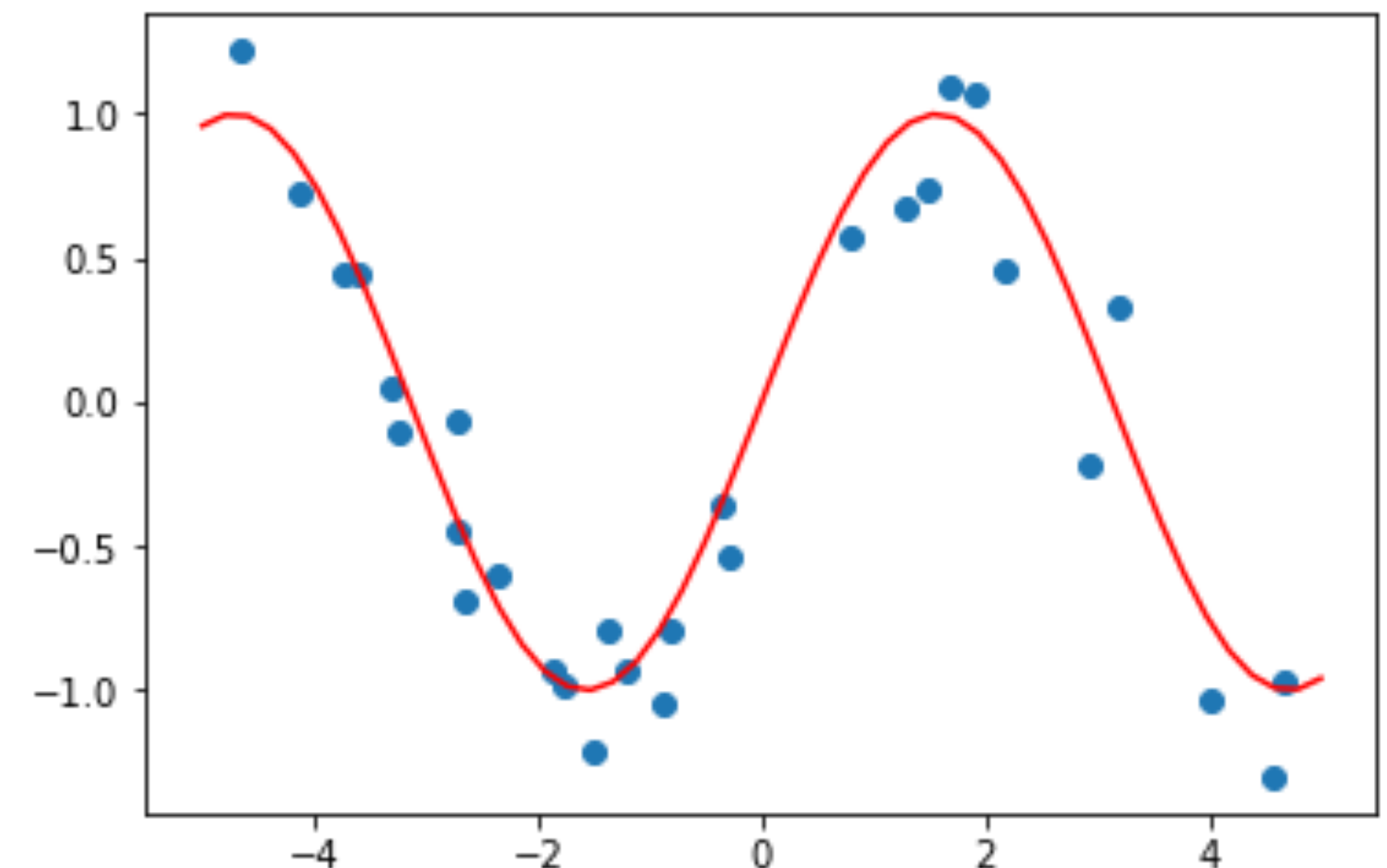
What we are given:



Learning



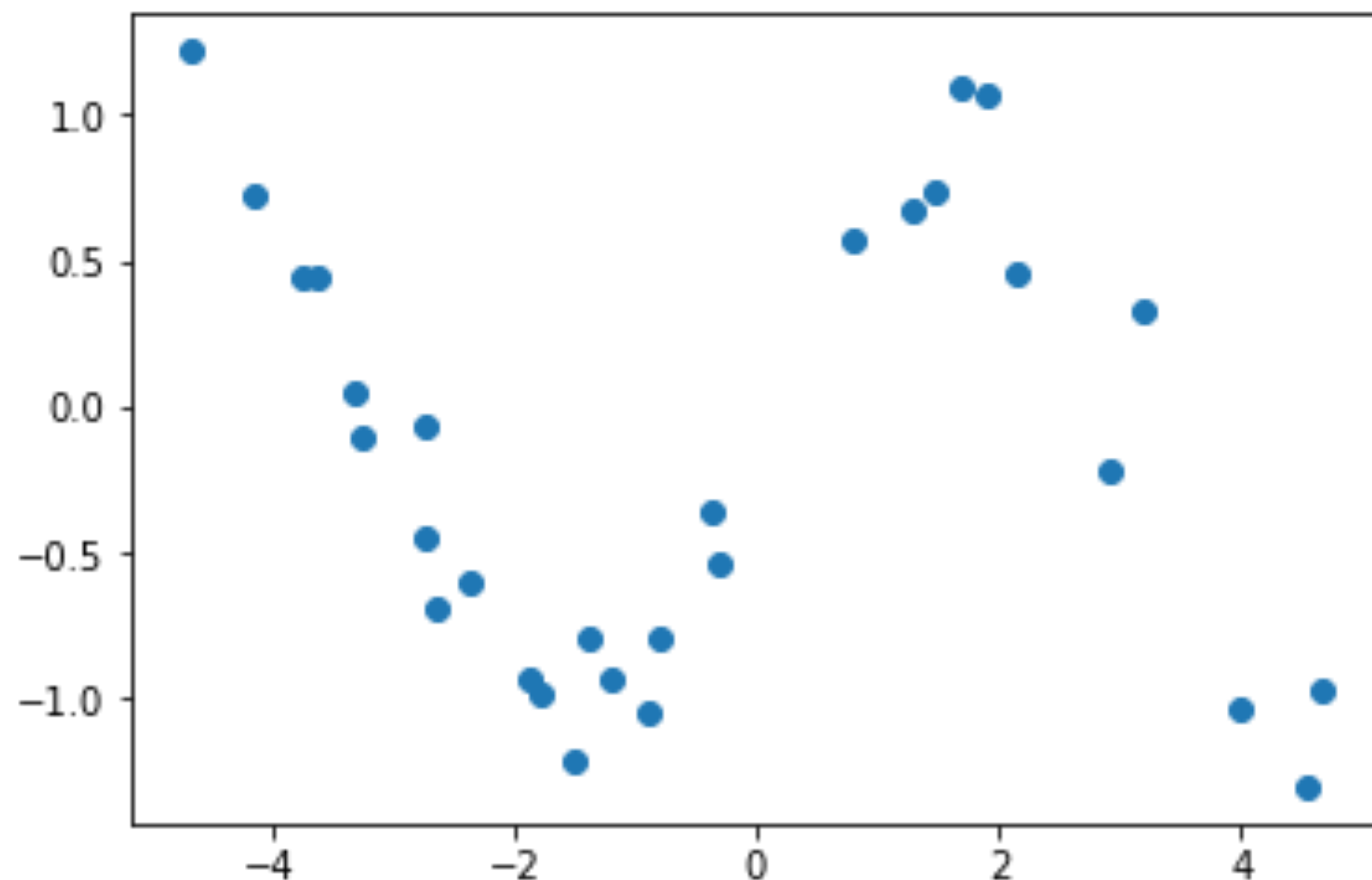
What we want to learn :



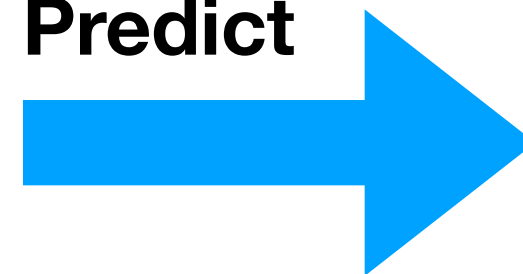
Predicting from examples

- This can also be seen as a prediction task
- Given examples of the value of the function, can we predict its value for points not given in the example?

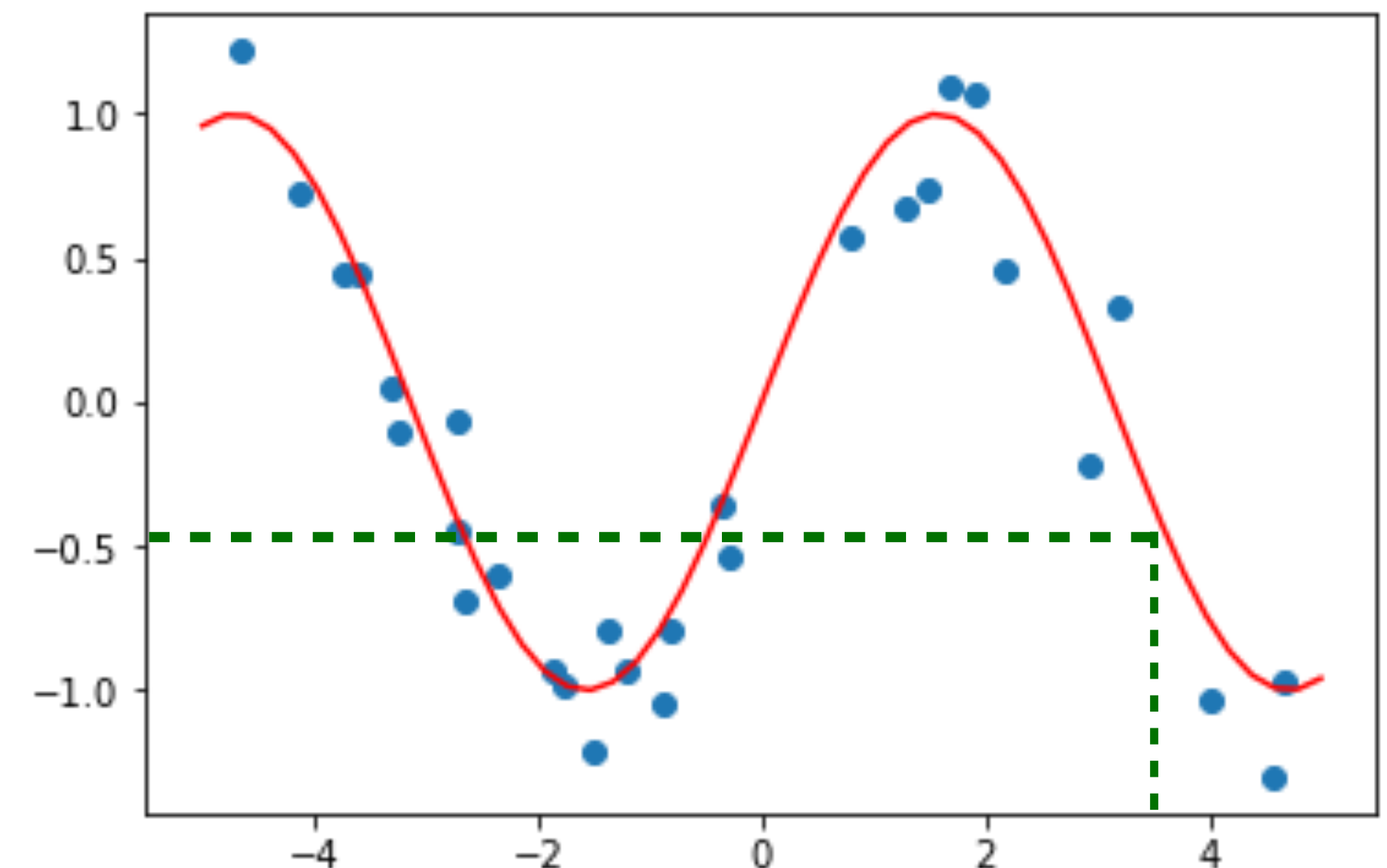
What we are given:



Predict



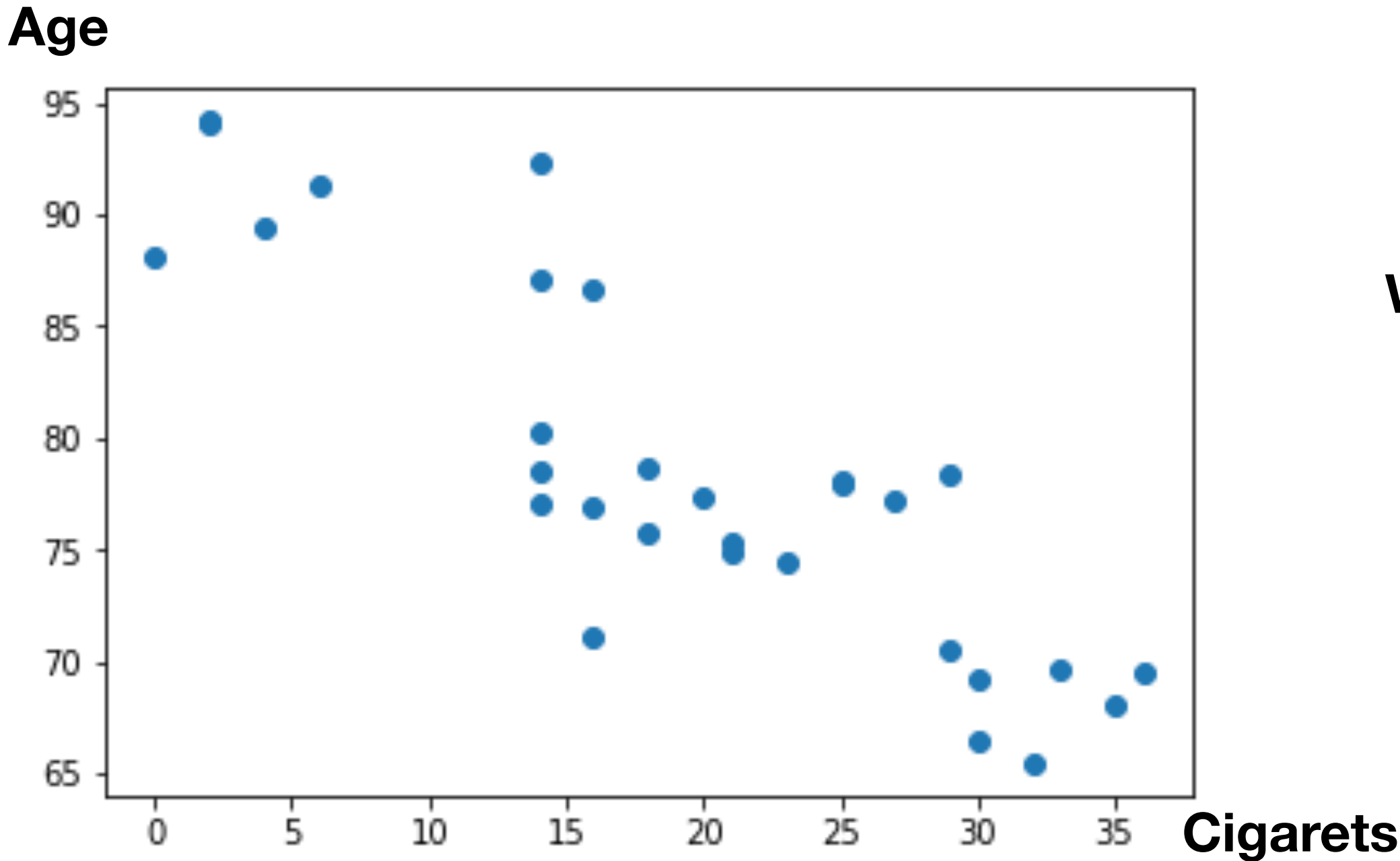
We predict the value at 3.5 should be -0.45



Example: influence of smoking on life expectancy

- For a more practical context, let us consider some Health-related situation
- Let us suppose we gathered the data from many persons about:
 - How many cigarets they were smoking per day
 - How old they died
- Now, knowing that somebody smokes x cigarets per day, we would like to **predict** what age he is most likely to die

daily cigarets	age of death
32.0	73.471399
7.0	88.237207
30.0	82.077261
17.0	85.576741
27.0	76.190373
15.0	84.899030
20.0	72.598501
28.0	77.018773
....	



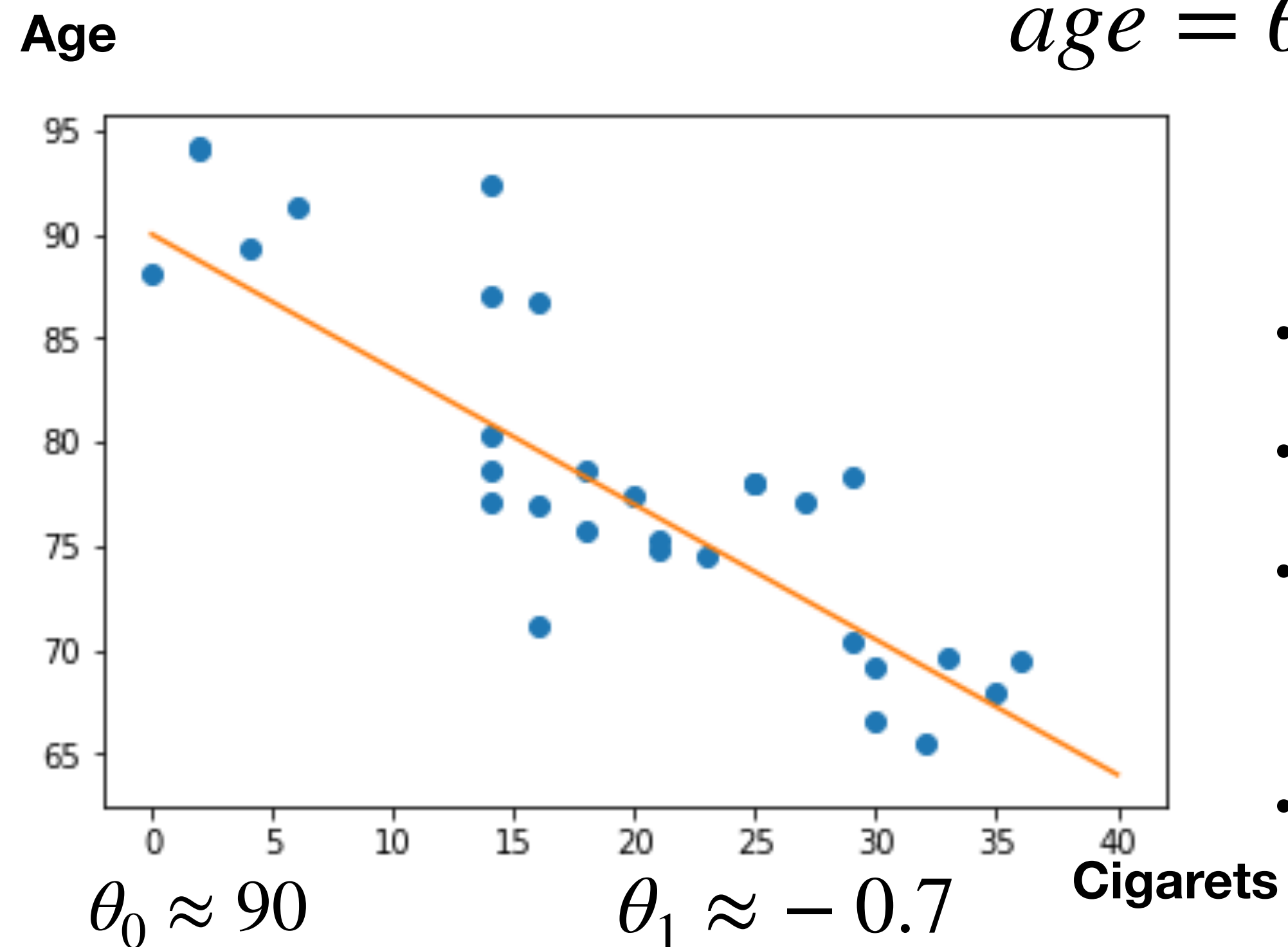
What age am I most likely to die if I smoke 10 cigarets per days?

Linear regression

- One possible answer is to do a ***linear regression***
- We assume there is a *linear relation* between loss in life expectancy and number of cigarettes smoked
- Noting the life expectancy (or age of death) as *age* and the number of cigarettes smoked as *cig*, we suppose:

$$age = \theta_0 + \theta_1 \times cig$$

Unknown Parameters



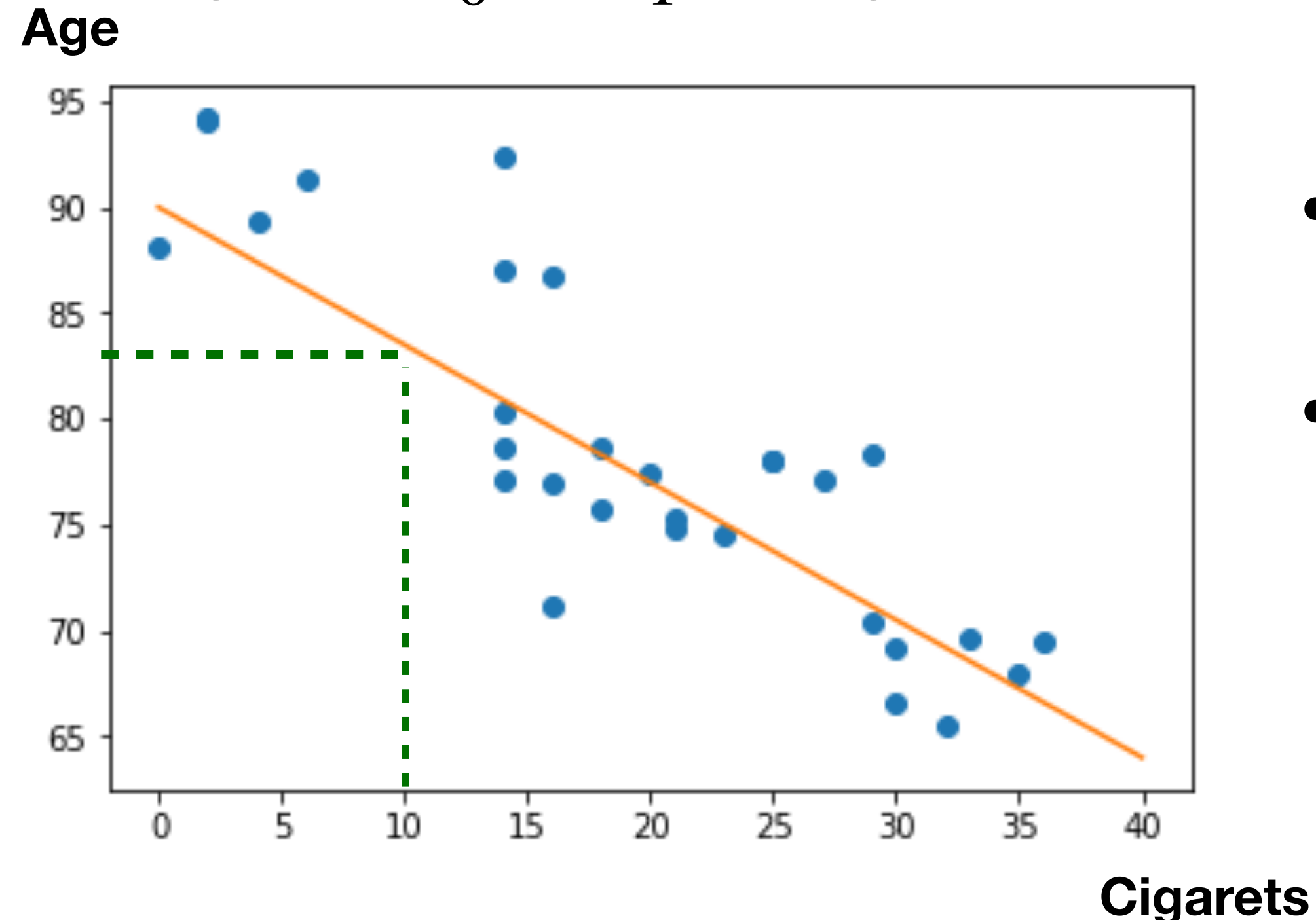
- The examples points are not exactly on a line
- This is because cigarettes are just one of many things that influence life expectancy
- Two persons smoking the same number of cigarettes might have different health condition, eating habits, do different amount of sport, etc... But we do not have this information in our data.
- This is why we said earlier we expect examples to be noisy: in general, we do not know all of the hidden factors explaining the output.

Linear regression

- The linear regression consist in finding the good values for the parameters
- In our Machine Learning terminology, we could also say we are “learning” the function that compute the life expectancy from the number of cigarets smoked

$$age = \theta_0 + \theta_1 \times cig$$

$$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$$



- Now, we can make our prediction:
- What age am I most likely to die if I smoke 10 cigarets per day?
- $90 - 0.7 \times 10 = 83$ year old

Linear Regression

- Linear regression is a common tool in **statistics**
- It can also be seen as a simple **Machine Learning** task
- Point of view is a bit different (exaggerating a bit):
- In statistics, we are also interested in interpreting the parameters, like the slope of the linear approximation
 - Smoking one additional pack of cigaret is associated with losing 2.5 years of life expectancy
- In Machine Learning, we do not care so much about the parameters. We care about the “predictive” power
 - If I know how many pack of cigarets a person smoke, can I predict what age she will die?

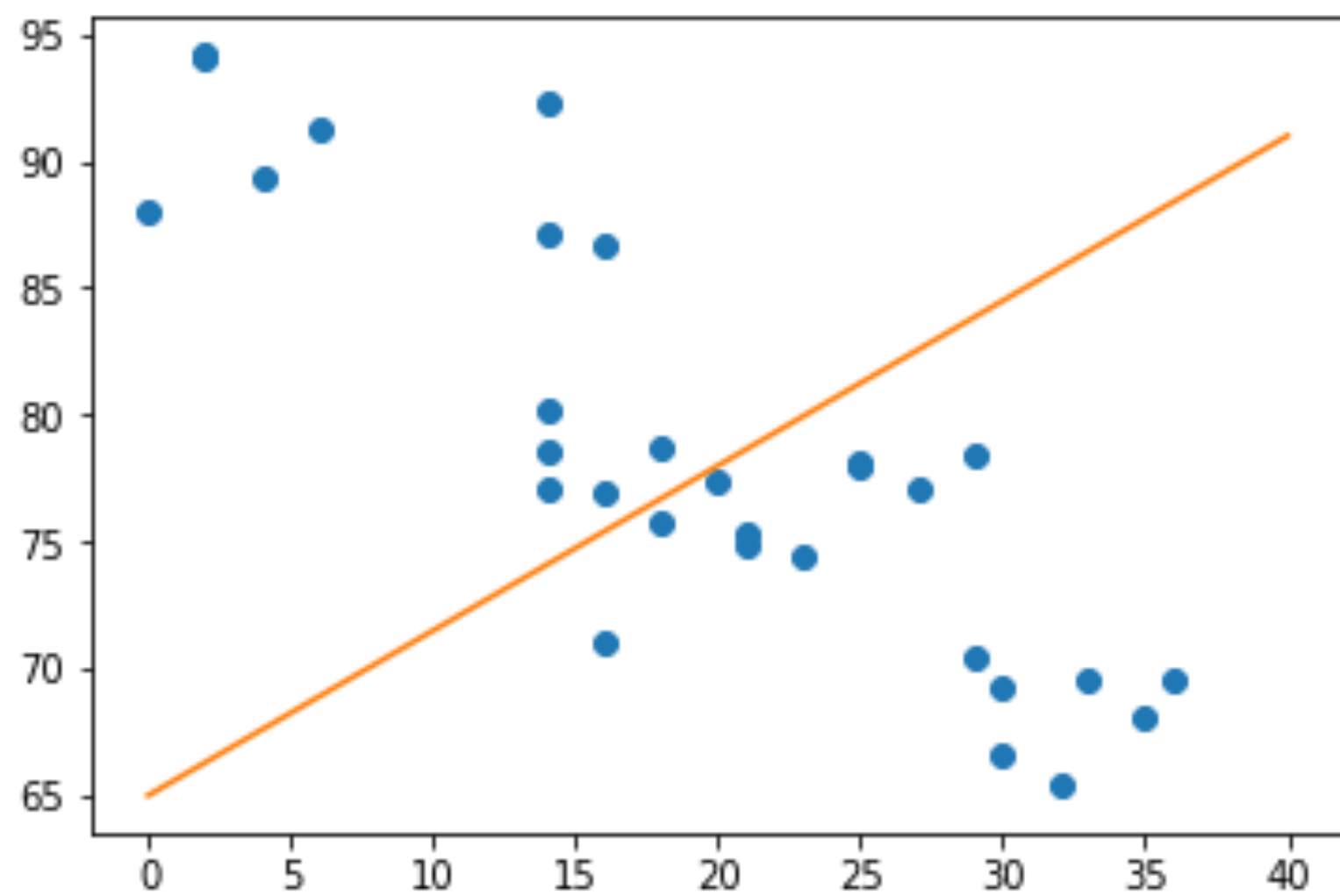
Linear regression

- How are we going to find the values of the parameters Θ_0, Θ_1 ?

- We need a criterion to evaluate the **quality** of Θ_0, Θ_1 .

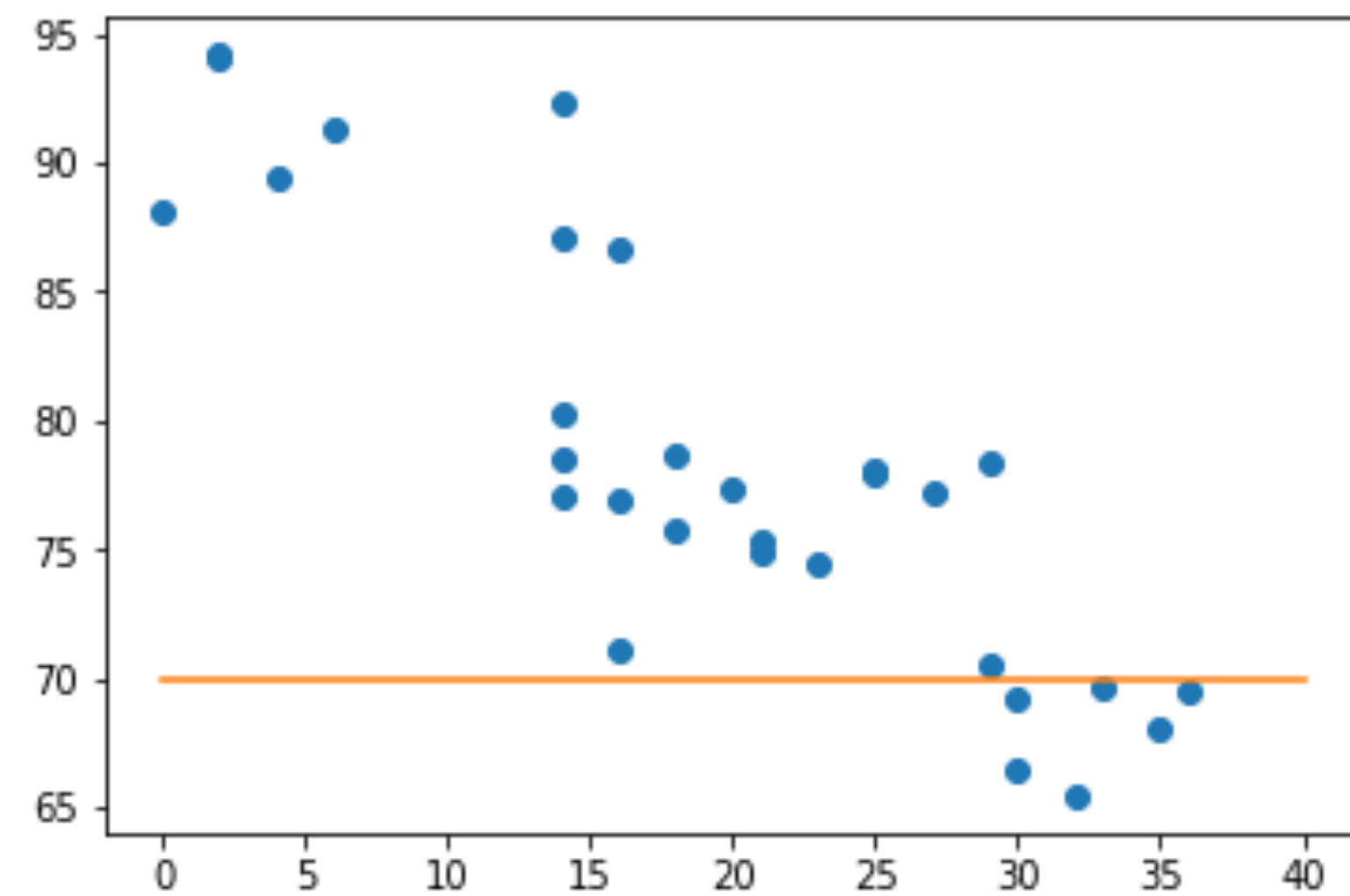
$$age = \theta_0 + \theta_1 \times cig$$

Age

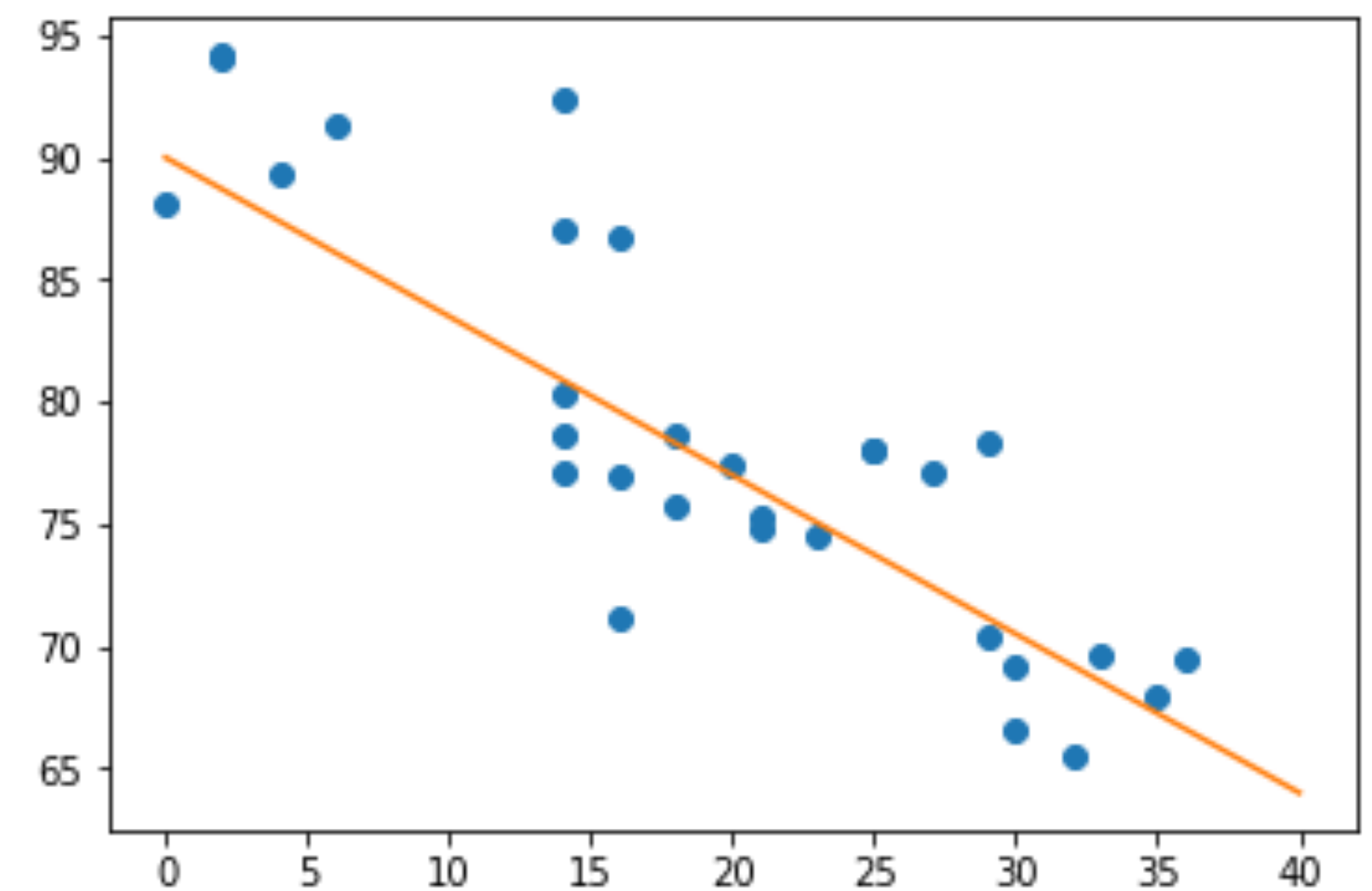


Cigarettes

$$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$$



$$\theta_0 \approx 70 \quad \theta_1 \approx 0$$



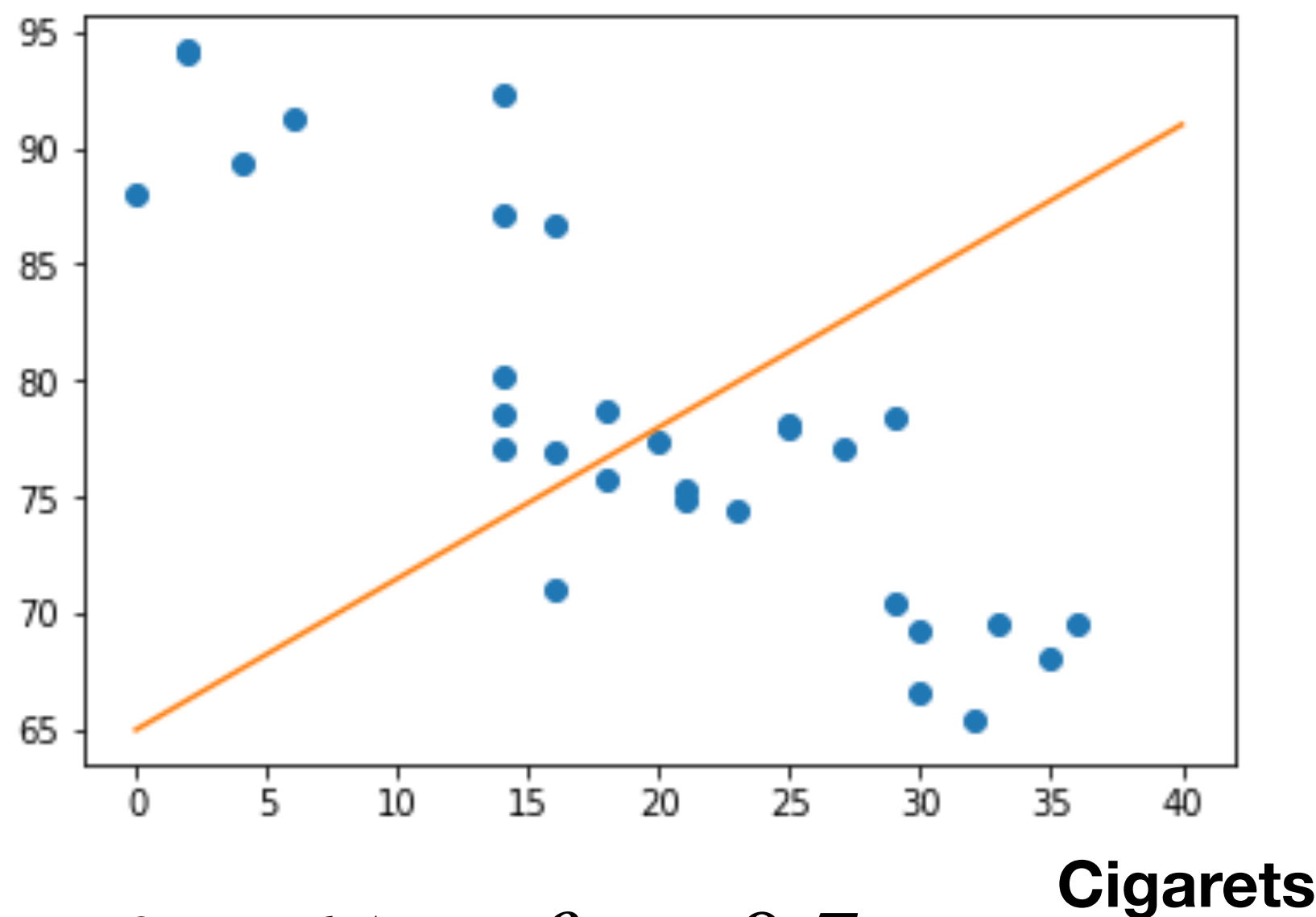
$$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$$

Linear regression

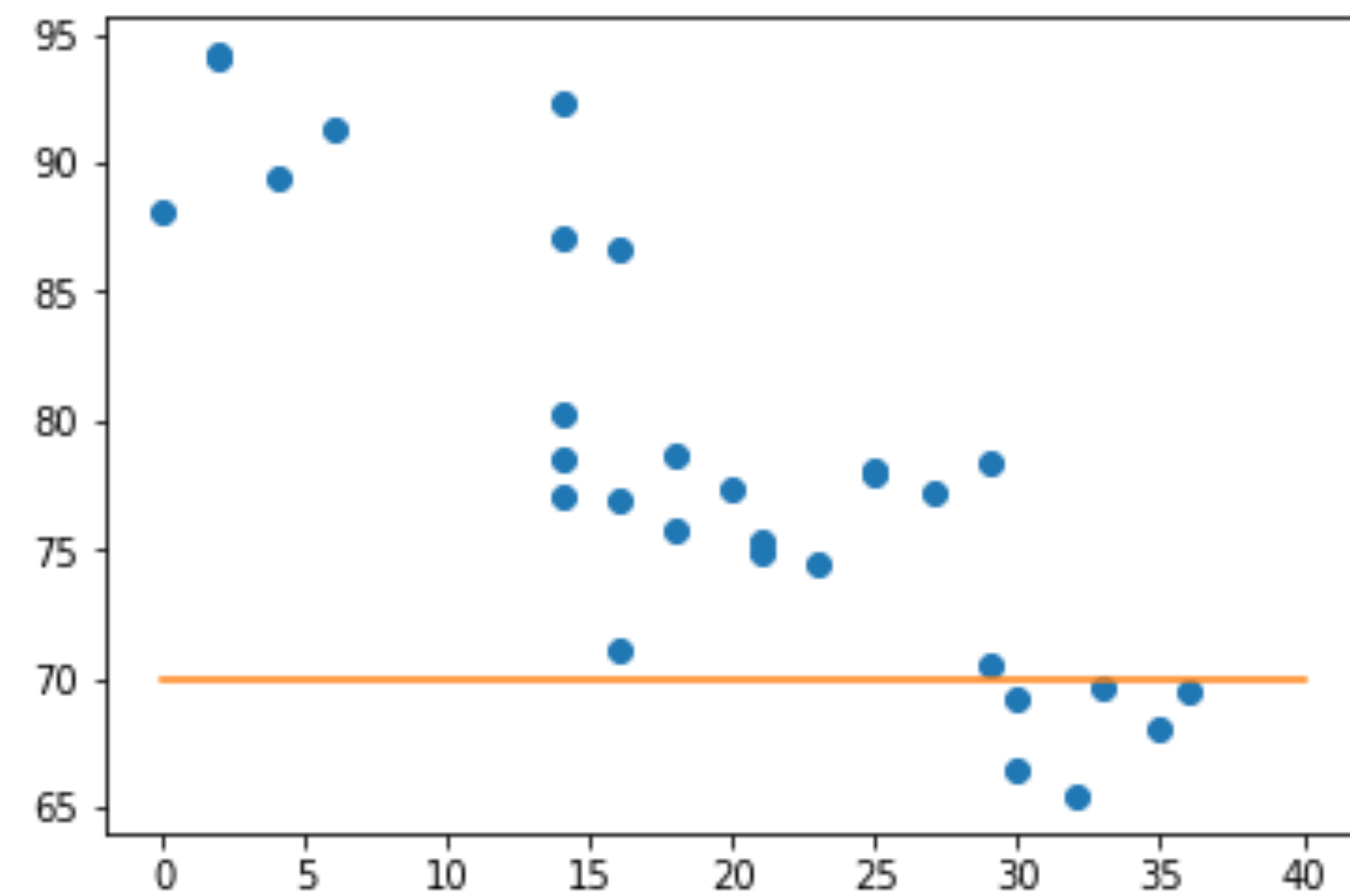
- How are we going to find the values of the parameters Θ_0, Θ_1 ?
- We need a criterion to evaluate the **quality** of Θ_0, Θ_1 .

$$age = \theta_0 + \theta_1 \times cig$$

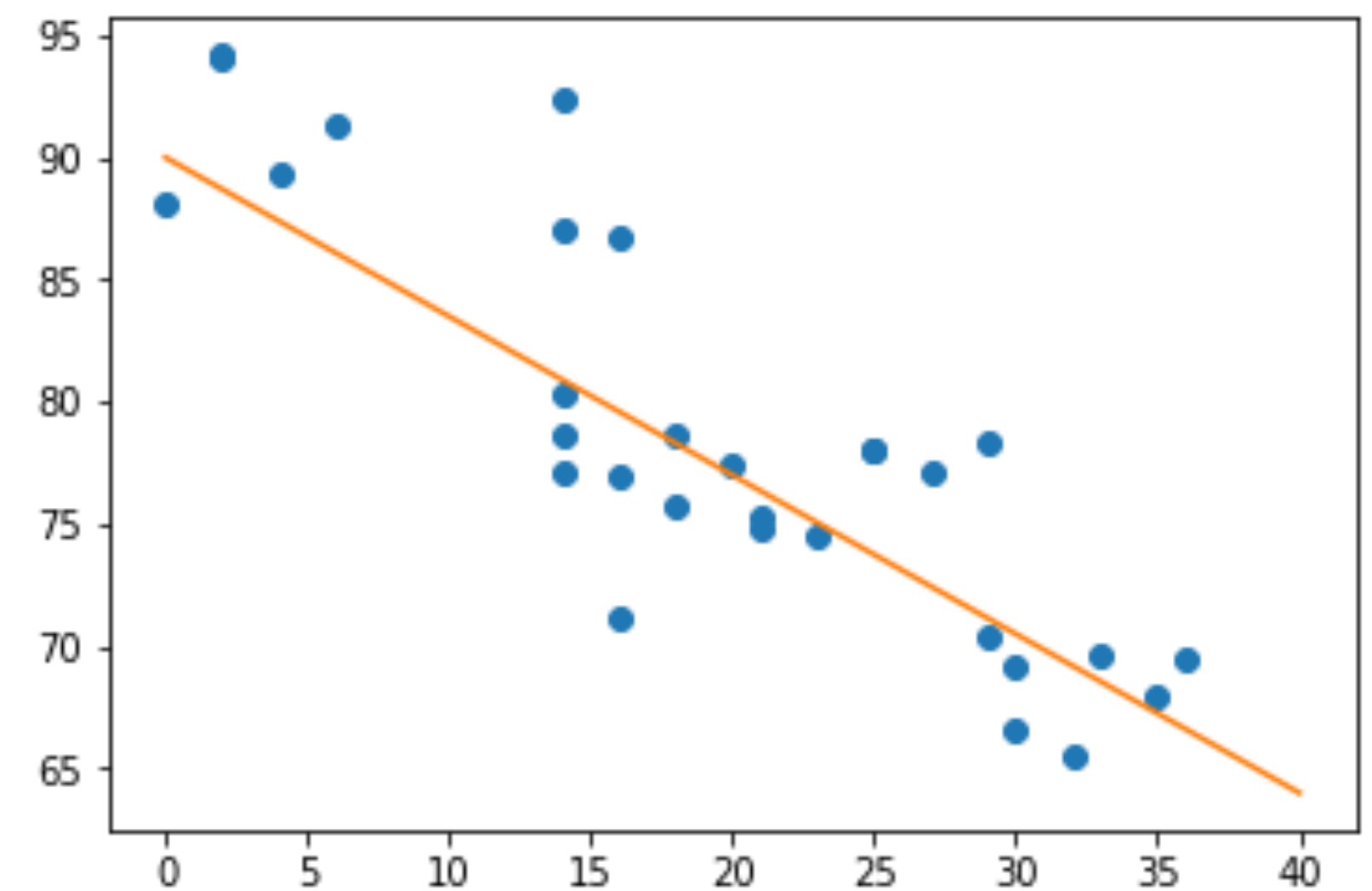
Age



$$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$$



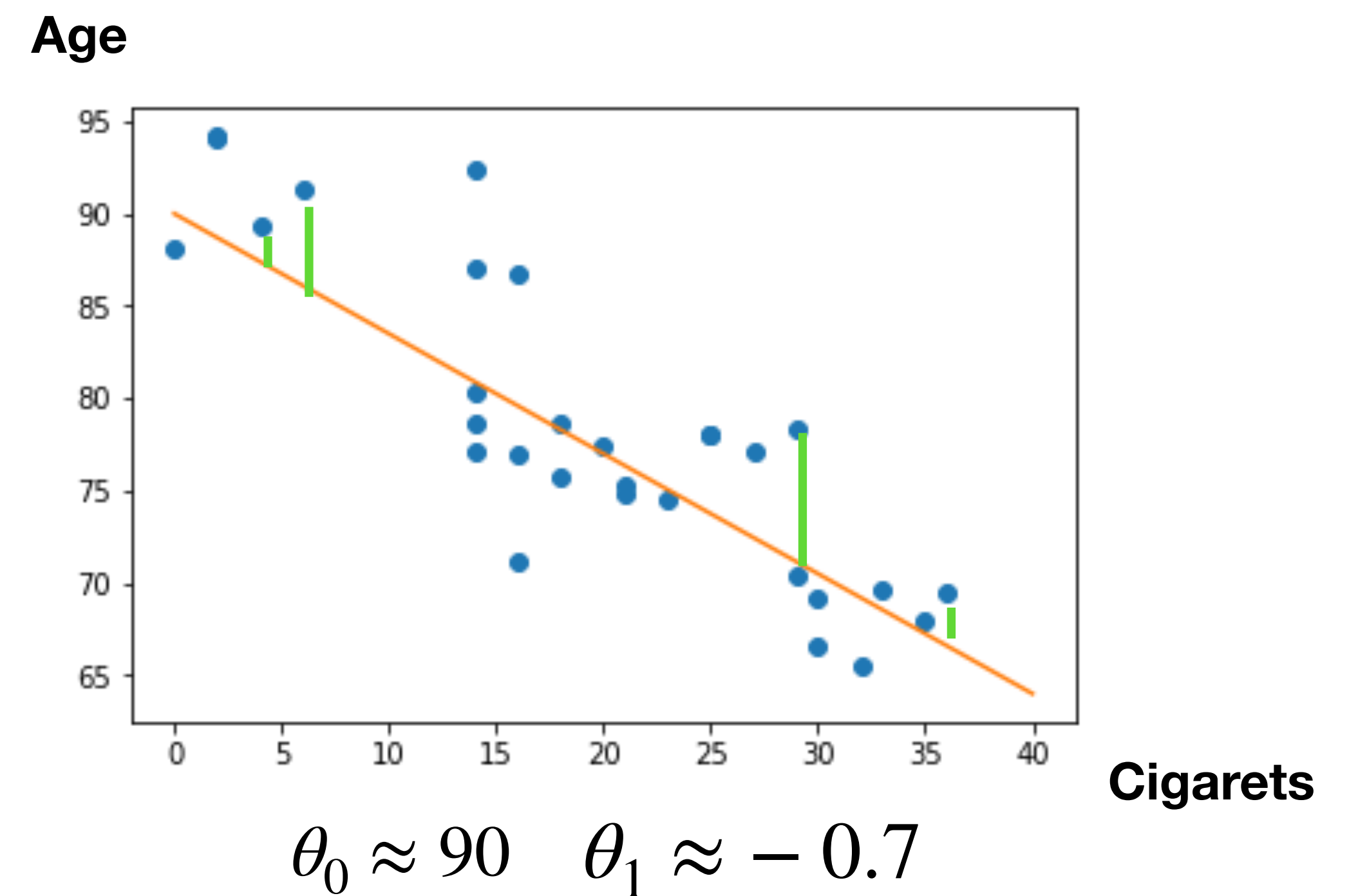
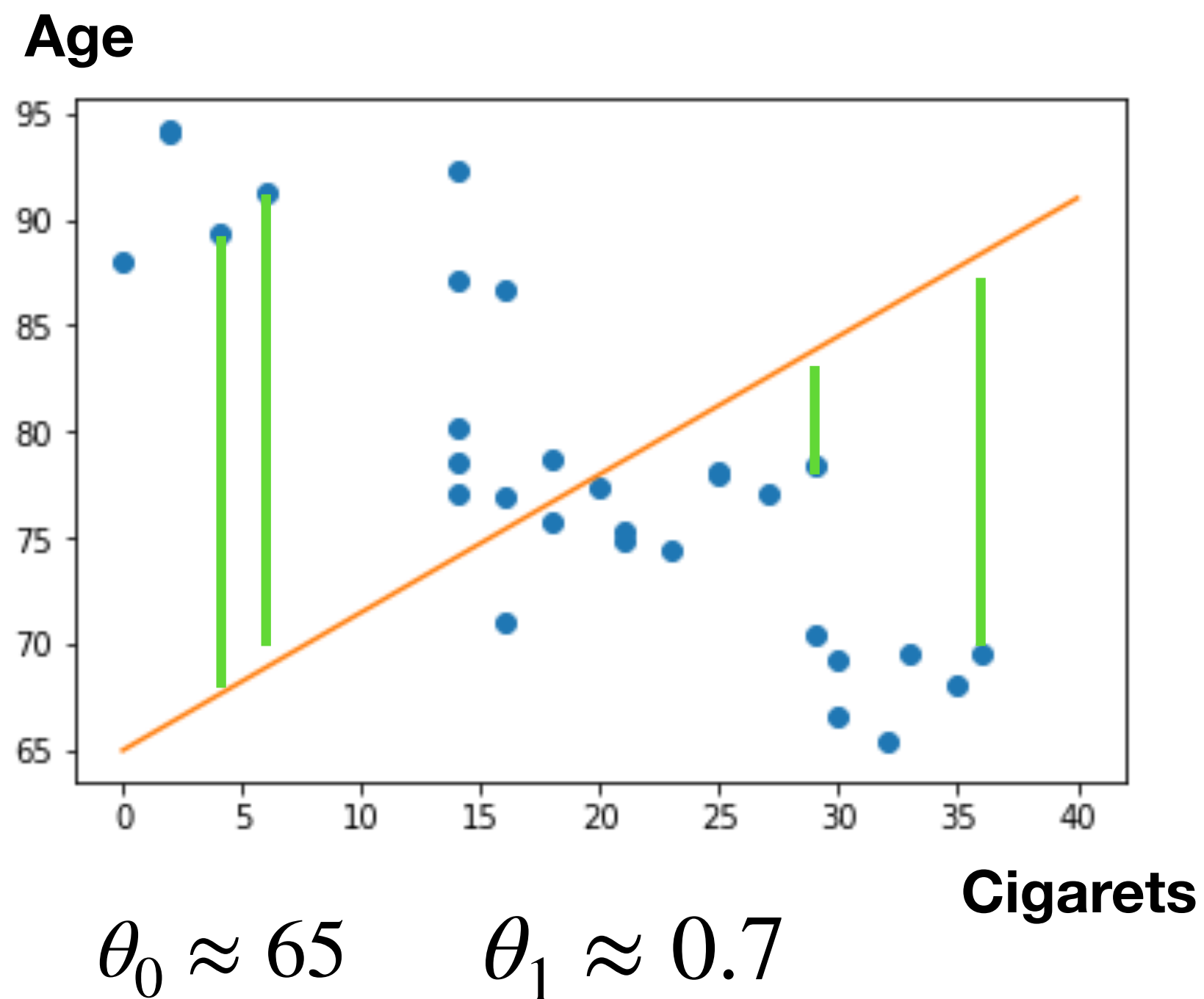
$$\theta_0 \approx 70 \quad \theta_1 \approx 0$$



$$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$$

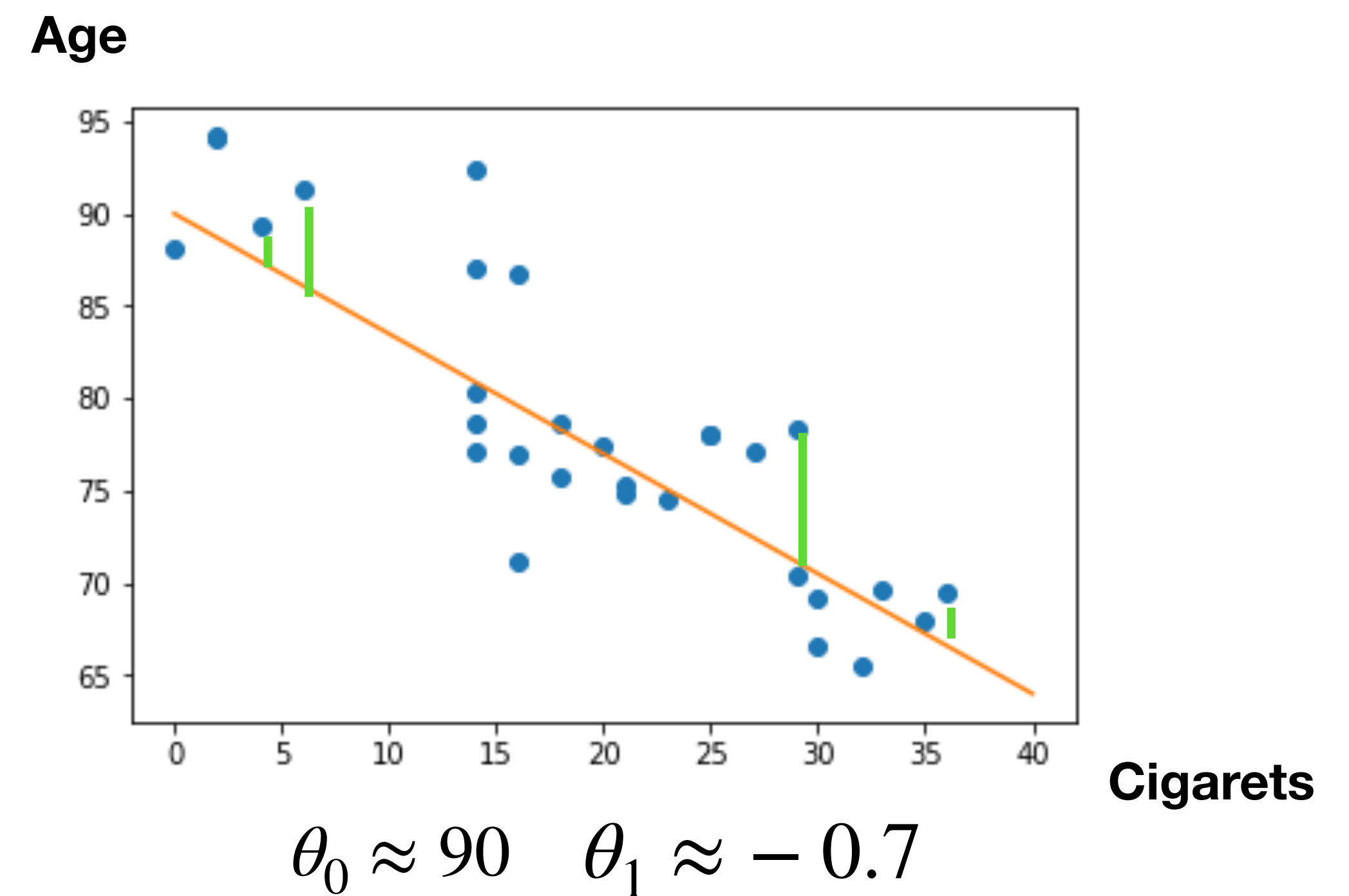
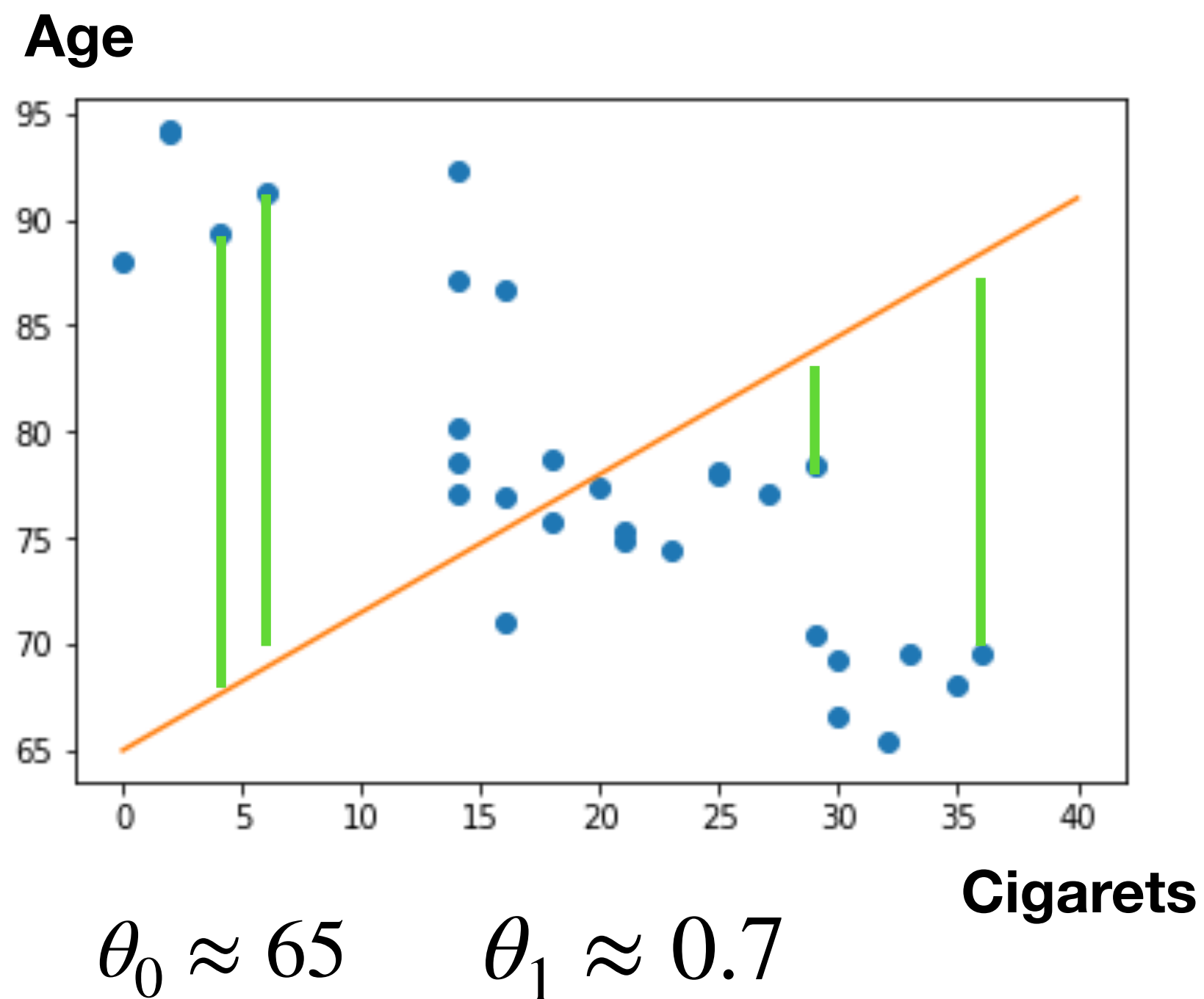
Mean Squared Distance

- We consider the distance between the examples and the model predictions



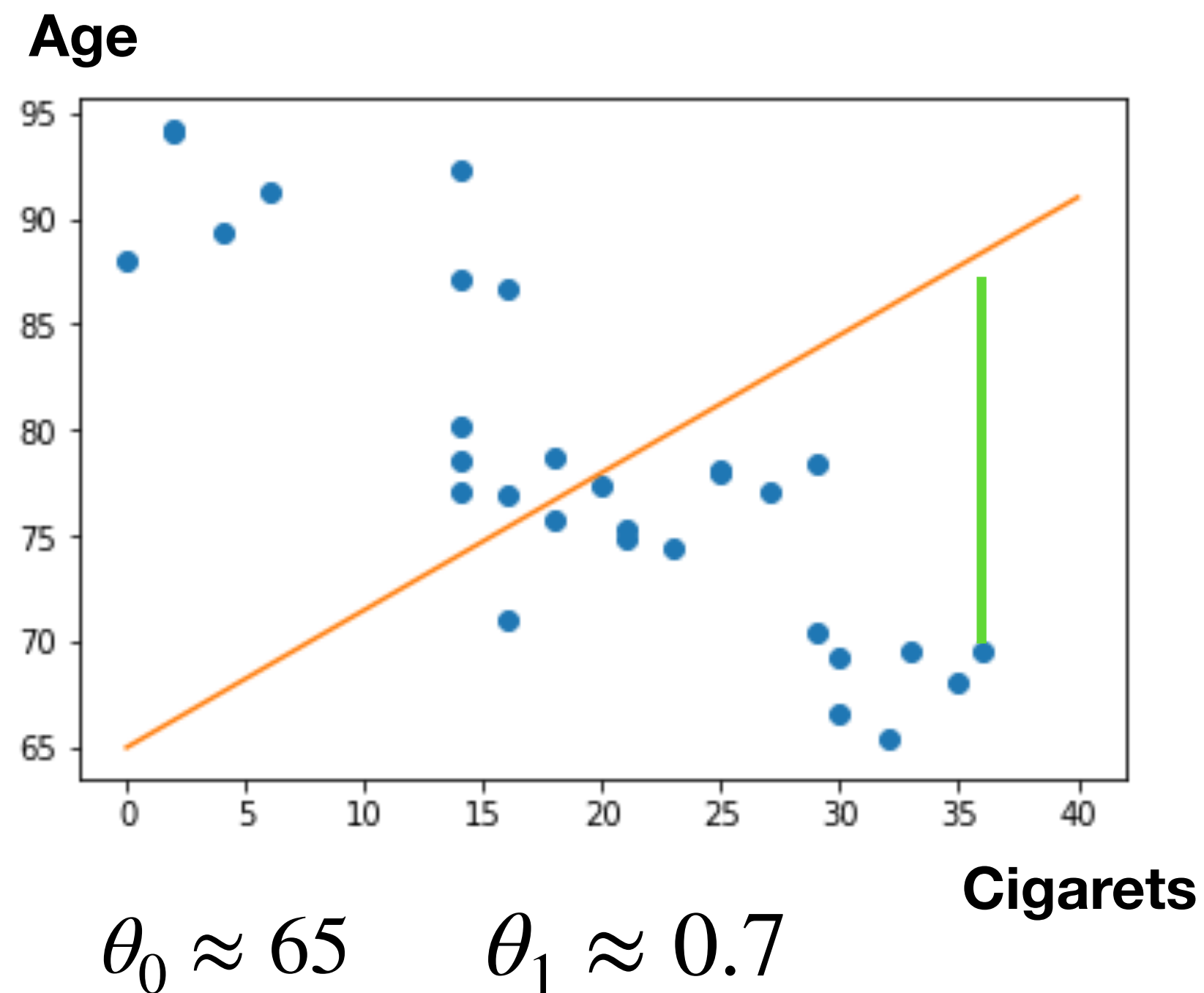
Mean Squared Distance

- We consider the distance between the examples and the model predictions



Mean Squared Distance

- We consider the distance between the examples and the model predictions



Person 7 in our data was smoking 36 cigarets per days,
and died at age 69

Our model, with the given Θ_0, Θ_1 was predicting
he would die at age 87

Difference is: -18 years

Squared distance is $18^2 = 356$

We compute this squared distance for all
persons in our dataset and compute the mean

$$\text{MeanSquaredDistance} = \frac{1}{N} \cdot \sum_i (f(x_i) - y_i)^2$$

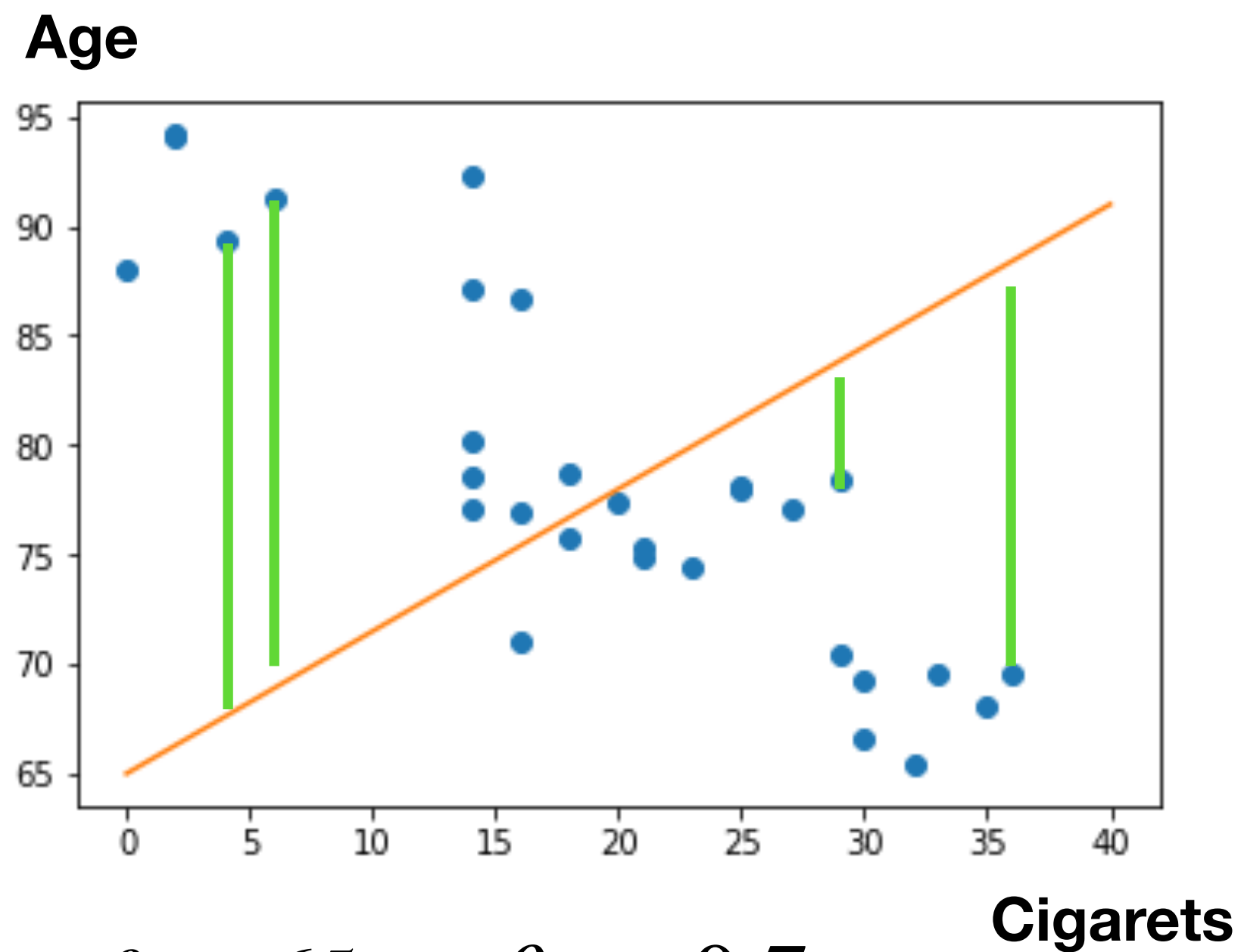
x_i : number of cigarets smoked by person i

y_i : age person i died

$f(x_i)$: prediction of our model

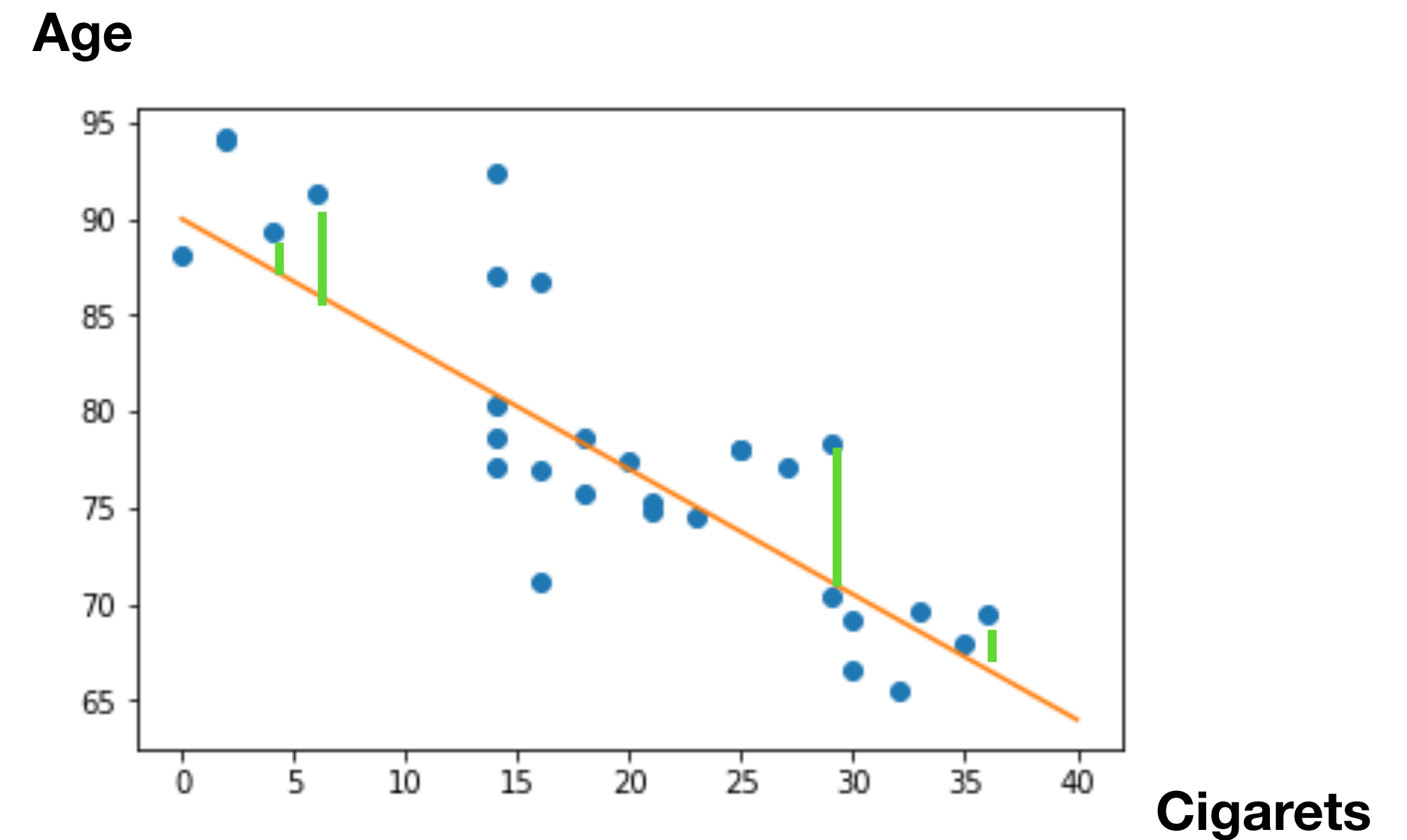
Mean Squared Distance

- The mean square distance now gives us a criterion for finding which parameters are best



$$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$$

$$\text{MeanSquaredDistance} = 294.7$$



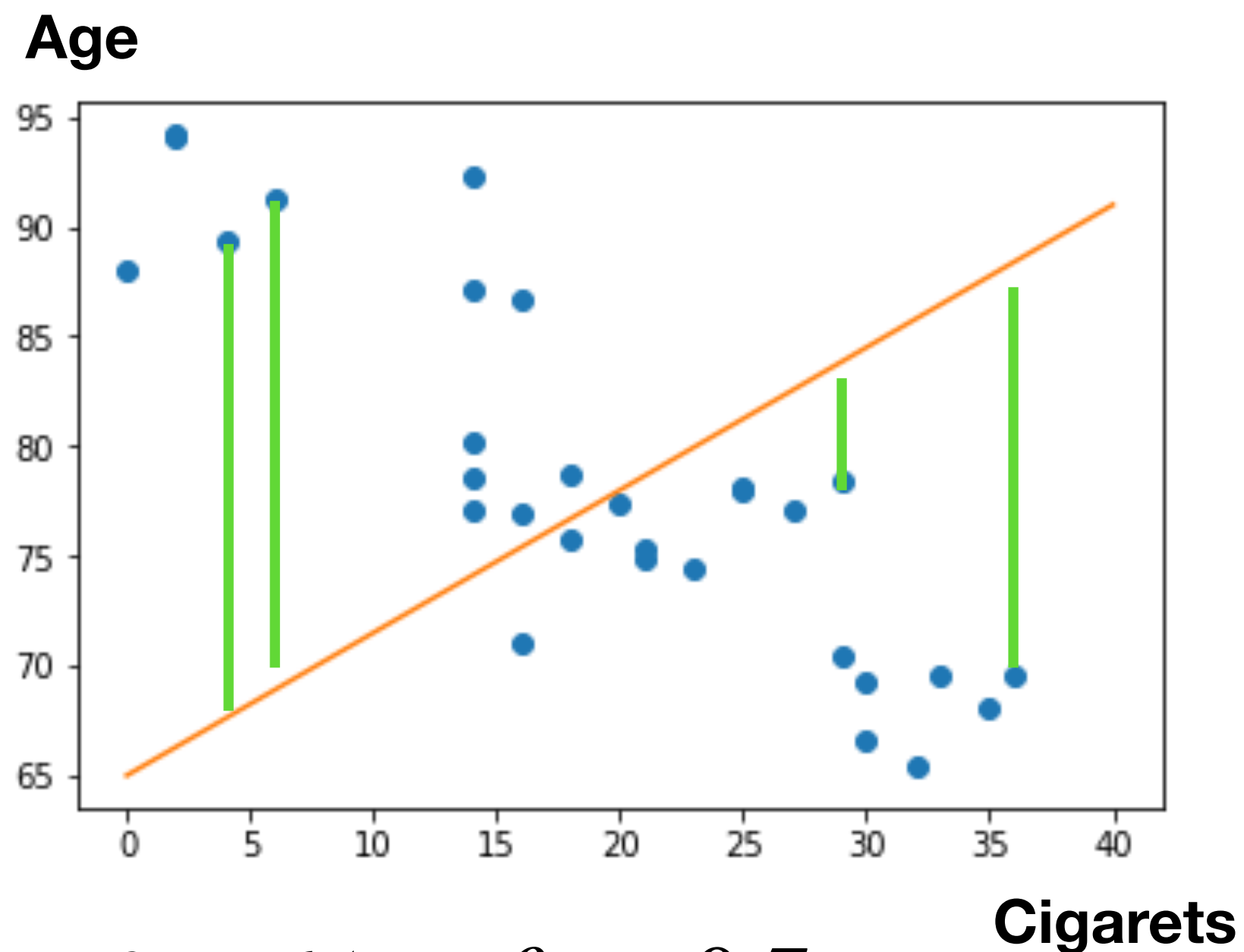
$$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$$

$$\text{MeanSquaredDistance} = 18.3$$

Minimizing the Mean Squared Distance

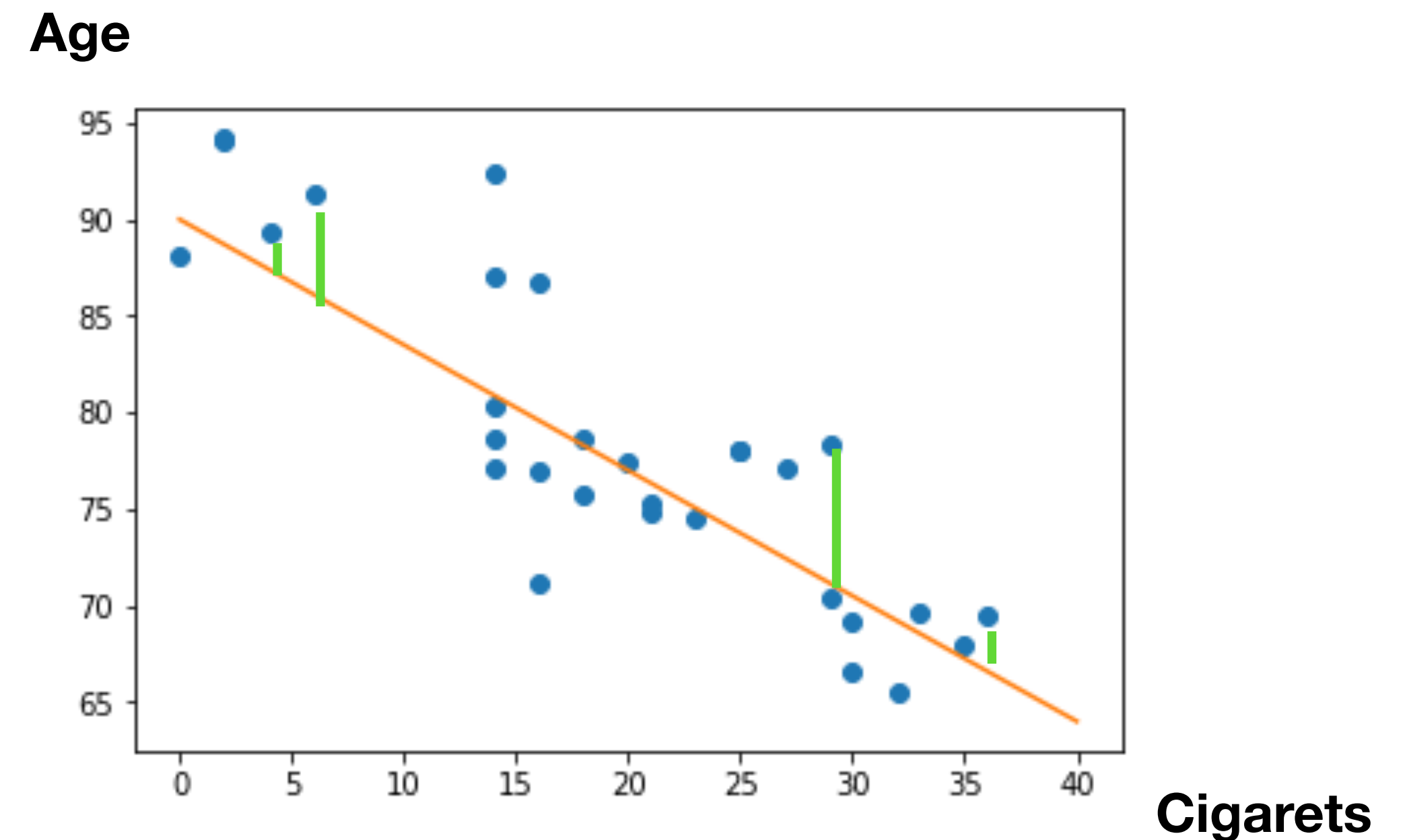
$$age = \theta_0 + \theta_1 \times cig$$

- Now, we know what we want: we want to find Θ_0, Θ_1 such that the Mean Squared distance is the smallest possible



$$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$$

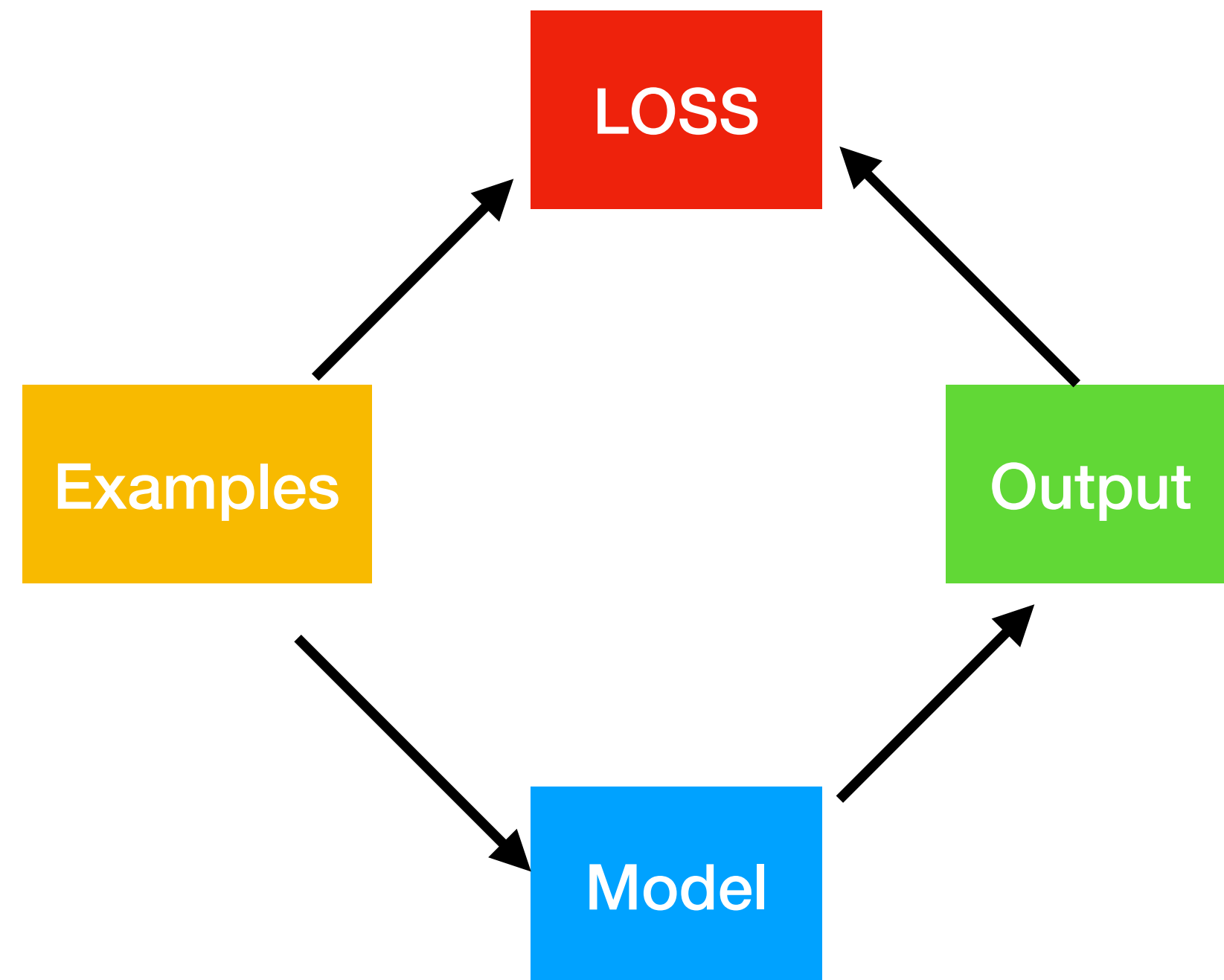
$$MeanSquaredDistance = 294.7$$



$$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$$

$$MeanSquaredDistance = 18.3$$

The General Supervised Machine Learning setting:



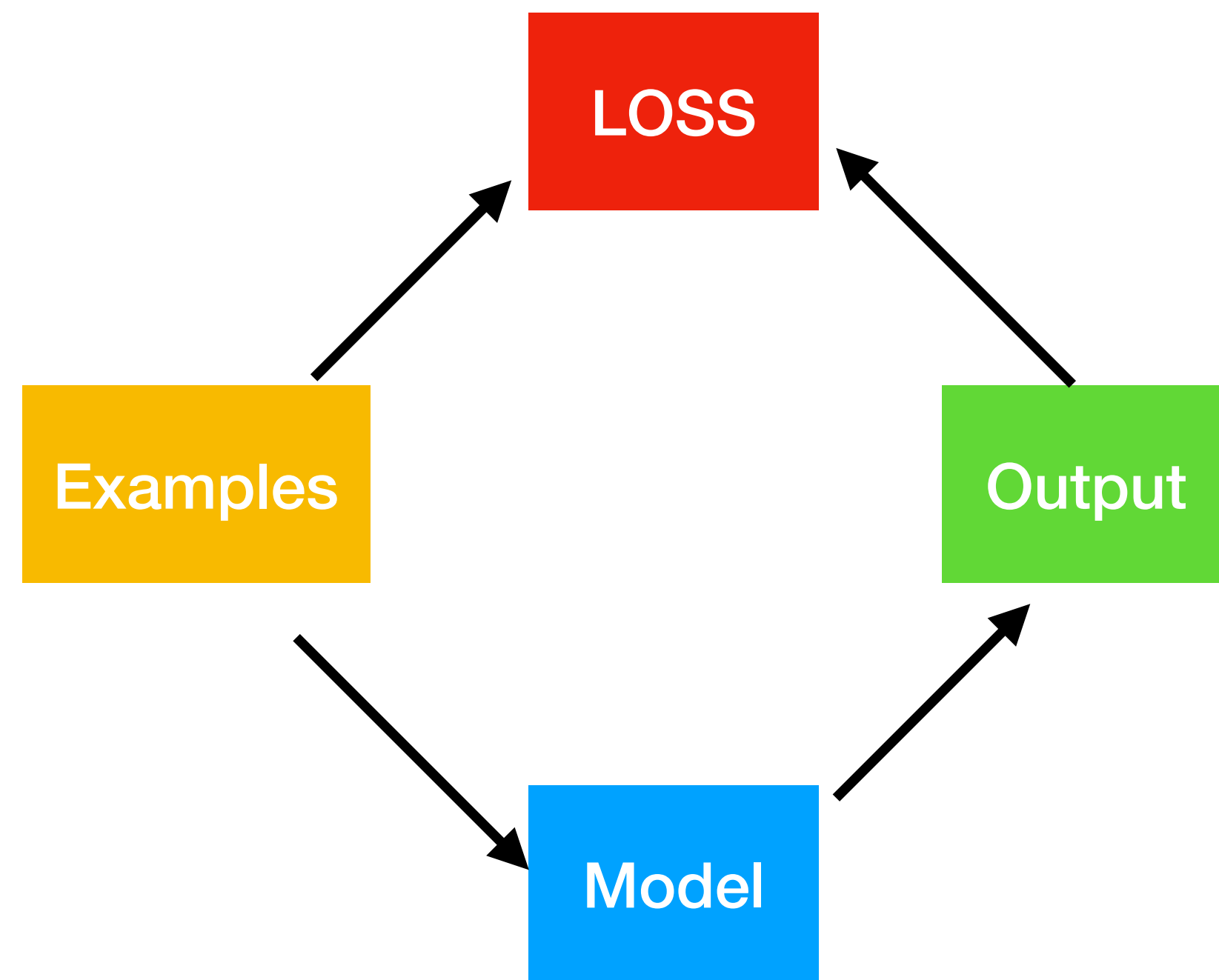
We have some examples

We have a model with some parameters

We have a loss function that compute the difference between the example and the prediction of our model

We minimize the loss to obtain the best parameters for our model

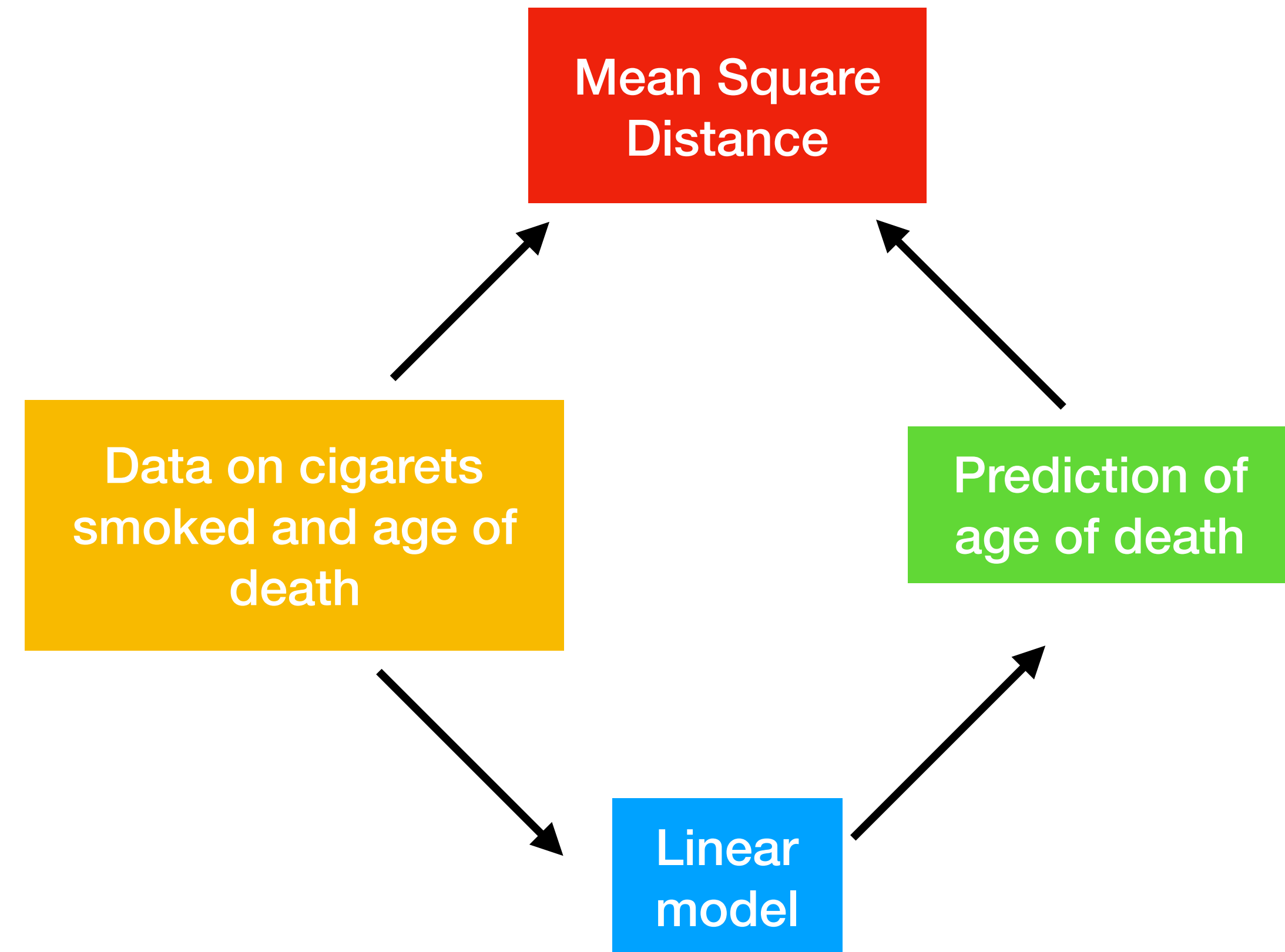
The General Supervised Machine Learning setting:



We have some examples (cigaretts/age data)

We have a model with some parameters
(linear function with parameters Θ_0, Θ_1)

We have a loss function that compute the difference
between the example and the prediction of our
model (Mean Squared Distance)



We minimize the loss to obtain the
best parameters for our model

Minimizing the Mean Squared Distance

- We saw one method for “easily” minimizing a function: **gradient descent**
- We can apply it here
- We need to express the Mean Squared Distance as a function of Θ_0, Θ_1

$$\textit{MeanSquaredDistance} = \frac{1}{N} \cdot \sum_i (f(x_i) - y_i)^2$$

x_i : number of cigarettes smoked by person i

y_i : age person i died

$f(x_i)$: prediction of our model

N : total number of persons in our data

With $f(x_i) = \theta_0 + \theta_1 \times x_i$

(just a rewriting of our model:)

$$\textit{age} = \theta_0 + \theta_1 \times \textit{cig}$$

Minimizing the Mean Squared Distance

- Therefore, the Mean Squared Distance as a function of Θ_0, Θ_1 is :

$$\textit{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

- What is the gradient?

Minimizing the Mean Squared Distance

$$\textit{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

- What is the gradient?

Minimizing the Mean Squared Distance

$$\textit{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

- What is the gradient?

$$\left| \begin{array}{l} \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \\ \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \end{array} \right|$$

Gradient Descent for Linear Regression

- Knowing the gradient, what would be the formula for the gradient descent update?

$$\theta_0 := ?$$

$$\theta_1 := ?$$

$$\begin{vmatrix} \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \\ \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \end{vmatrix}$$

Gradient Descent for Linear Regression

- Knowing the gradient, what would be the formula for the gradient descent update?

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \quad \left| \begin{array}{l} \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \\ \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \end{array} \right|$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Gradient Descent for Linear Regression

$$\text{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with $\theta_0, \theta_1 = 0$, $lr=0.1$

$$MSD(0,0) = ?$$

$$\theta_0 := \theta_0 - ?$$

$$\theta_1 := \theta_1 - ?$$

Our data:

daily cigaretts	age of death
32.0	73
7.0	88
17.0	85

Gradient Descent for Linear Regression

$$MeanSquaredDistance(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with $\Theta_0, \Theta_1 = 0$, $lr=0.1$

daily cigaretts	age of death
32.0	73
7.0	88
17.0	85

$$MSD(0,0) = \frac{1}{3} \cdot ((0 + 0 \times 32 - 73)^2 + (0 + 0 \times 7 - 88)^2 + (0 + 0 \times 17 - 85)^2) = 6766$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{3} \cdot ((0 + 0 \times 32 - 73) + (0 + 0 \times 7 - 88) + (0 + 0 \times 17 - 85)) = 16.4$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{3} \cdot (32 \times (0 + 0 \times 32 - 73) + 7 \times (0 + 0 \times 7 - 88) + 17 \times (0 + 0 \times 17 - 85)) = 293.1$$

Gradient Descent for Linear Regression

$$\text{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$
$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$
$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Our data:

daily cigaretts	age of death
32.0	73
7.0	88
17.0	85

Interesting question: can we get rid of the sigma?

The mean squared distance and its gradient are computed as an average over the data examples

If instead, we just compute the gradient for a random example, does it work?

Gradient Descent for Linear Regression

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with $\theta_0, \theta_1 = 0$, $lr=0.1$

$$\theta_0 := \theta_0 - lr \times \frac{2}{3} \cdot ((0 + 0 \times 32 - 73) + (0 + 0 \times 7 - 88) + (0 + 0 \times 17 - 85)) = 16.4$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{3} \cdot (32 \times (0 + 0 \times 32 - 73) + 7 \times (0 + 0 \times 7 - 88) + 17 \times (0 + 0 \times 17 - 85)) = 293.1$$

Instead, choose an example at random (ie. Person 2)

$$\theta_0 := \theta_0 - lr \times 2 \cdot ((0 + 0 \times 7 - 88)) = 17.6 \quad \theta_1 := \theta_1 - lr \times 2 \cdot (7 \times (0 + 0 \times 7 - 88)) = 123$$

daily cigaretts	age of death
32.0	73
7.0	88
17.0	85

Gradient Descent for Linear Regression

$$\text{MeanSquaredDistance}(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Our data:

daily cigaretts	age of death
32.0	73
7.0	88
17.0	85

Interesting question: can we get rid of the sigma?

The mean squared distance and its gradient are computed as an average over the data examples

If instead, we just compute the gradient for a random example, does it work?

YES! (provided the learning rate is decreased over time:
it is Stochastic Gradient Descent)

Stochastic Gradient Descent

- Stochastic gradient descent says that we can replace the average of the gradient over all examples by the gradient given by a randomly chosen example
- Slower Convergence
- But if we have one million examples: one million times faster to compute!

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \quad \xrightarrow{\text{Choose example } i \text{ randomly}} \quad \theta_0 := \theta_0 - lr \times 2 \cdot (\theta_0 + \theta_1 \times x_i - y_i)$$

- In practice, we often average over a few examples (instead of just one). This is called mini-batch gradient descent

Linear Regression: other methods

- Because Linear Regression is a very simple form of Machine Learning (we model data with a simple function), there are methods more direct to minimize the Mean Squared Distance (eg. Normal Equations)
- However, for many equations and many variables, Stochastic Gradient Descent can still be the most efficient solution
- There are many existing implementations of Linear Regression, so in practice you would not need to do the gradient descent by yourself anyway
- For example, in python, we can use the function *linregress* in the package *stats* of the library *scipy*:

```
In [1061]: from scipy import stats
```

```
In [1062]: stats.linregress(daily_cig_smoked, life_expectancy)
```

```
Out[1062]: LinregressResult(slope=-0.710478500965002, intercept=92.29875345880524, rvalue=-0.8609731085193286, pvalue=1.0335137971251332e-09, stderr=0.07932348624135113)
```

Next

- What if I have more than one variable I want to use?
 - eg. Using Daily number of cigarettes, weight, BMI index and sport activity to predict age of death
- What if I want to learn a function more complex than a linear function?
- We will see that next time. But we will see that in practice the process is always the same:
 - Define the model
 - Define the loss
 - Do a gradient descent on the loss