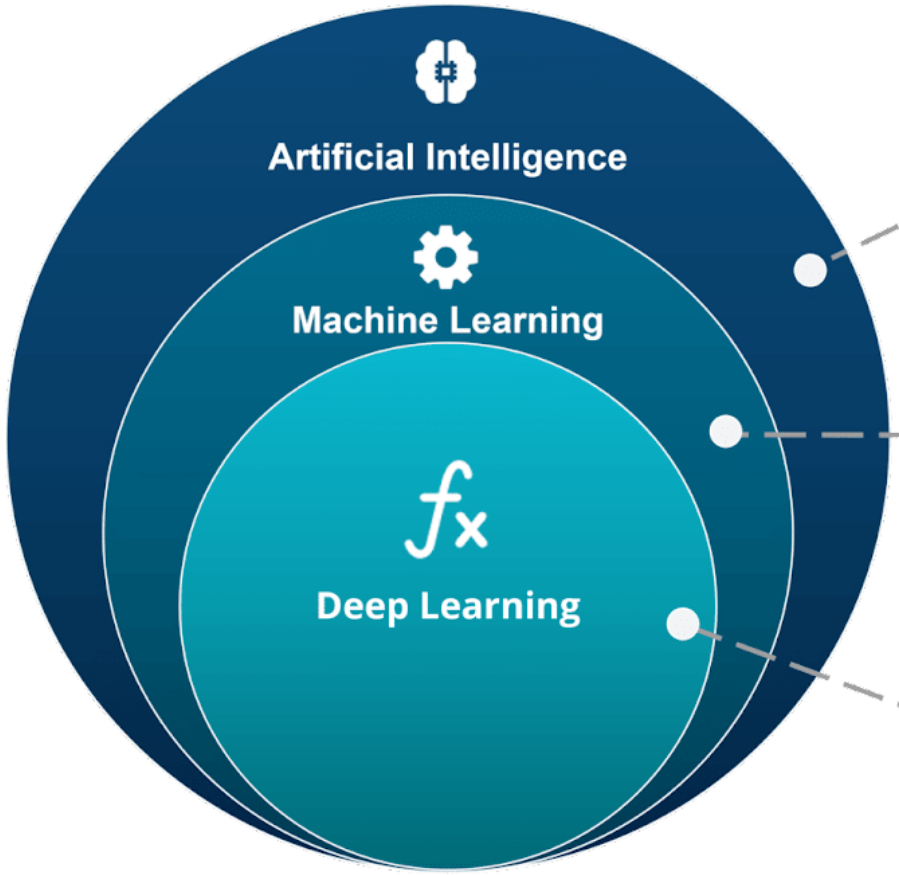
Session

Introduction To Machine Learning

Agenda

- Introduction to machine learning
- Supervised Learning
  - Linear Regression
  - Gradient Descent
  - Linear Regression vs Logistic Regression

Introduction To Machine Learning



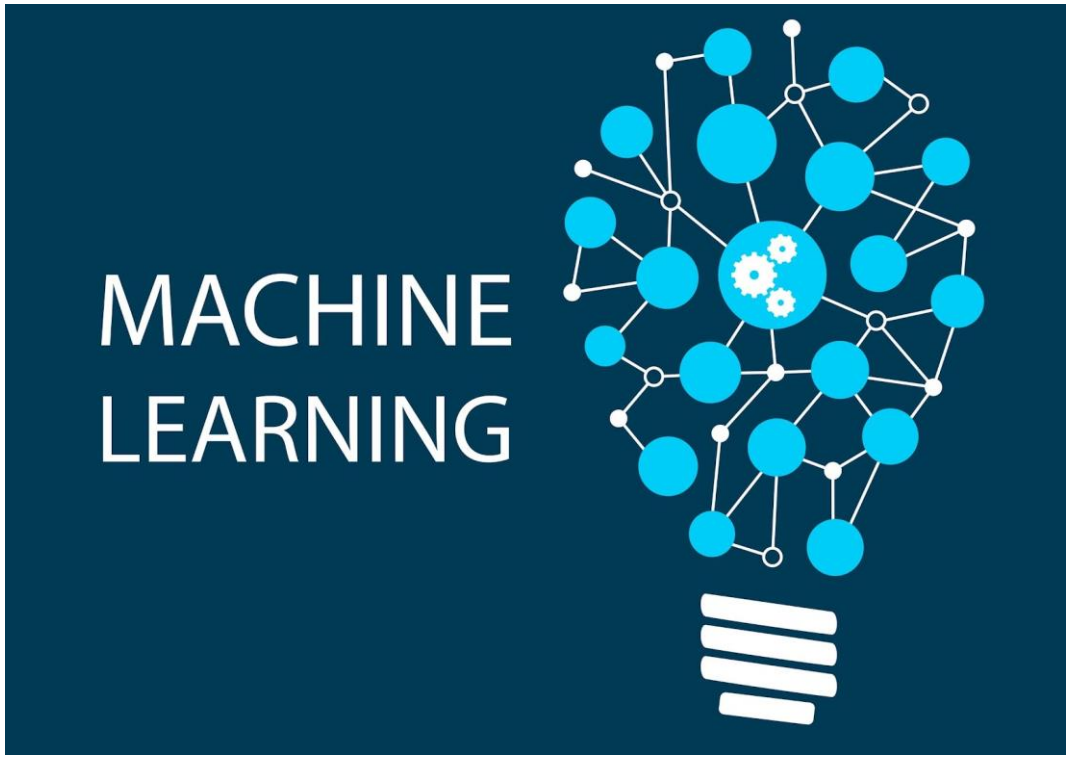
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

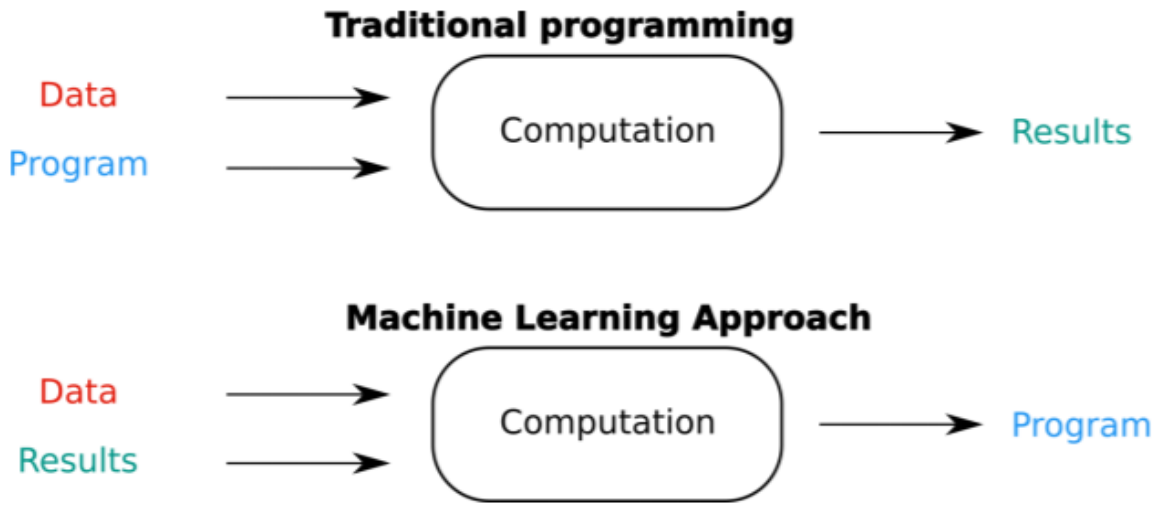
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Introduction To Machine Learning

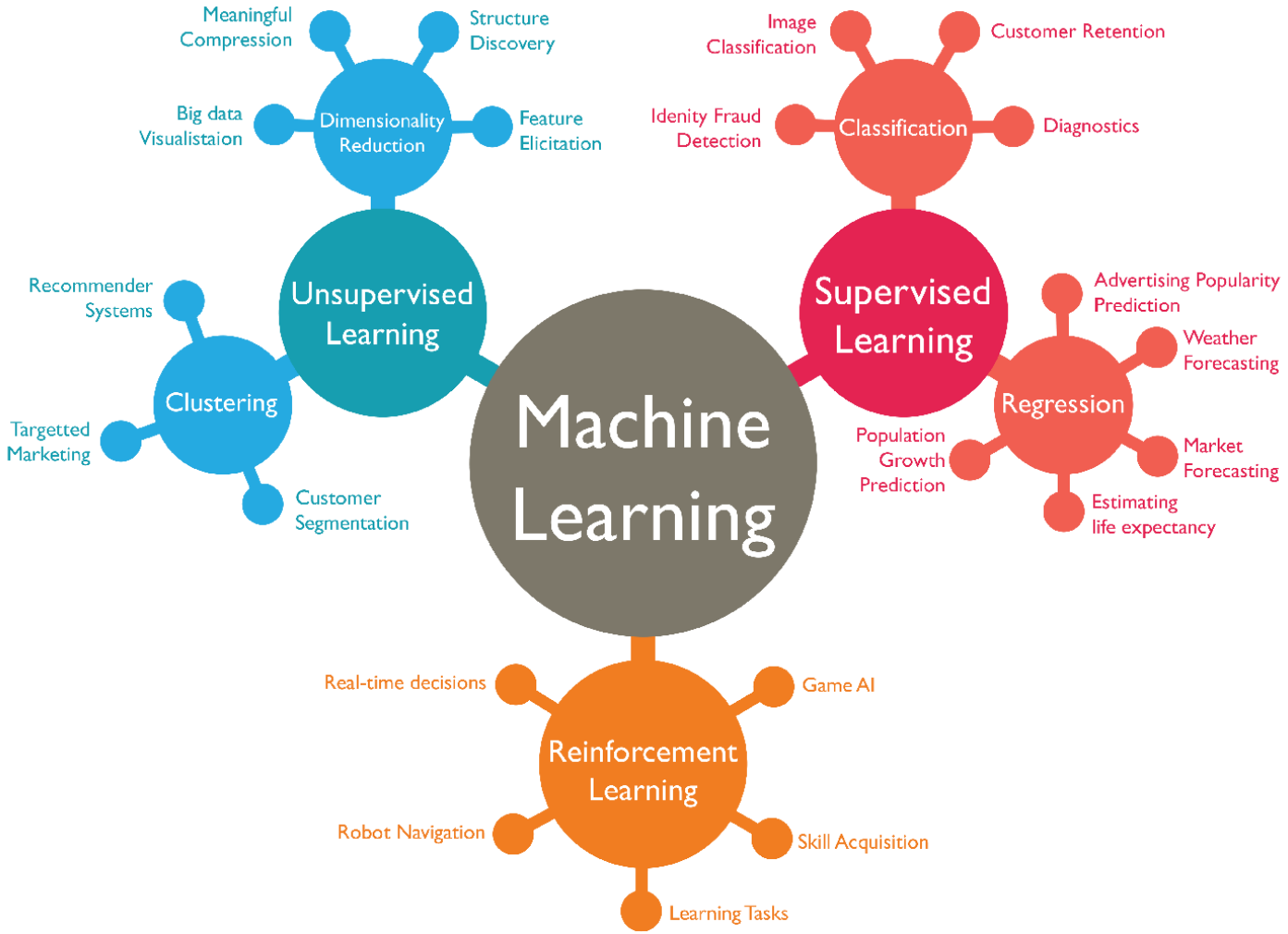
- **Machine learning** is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed



Introduction To Machine Learning

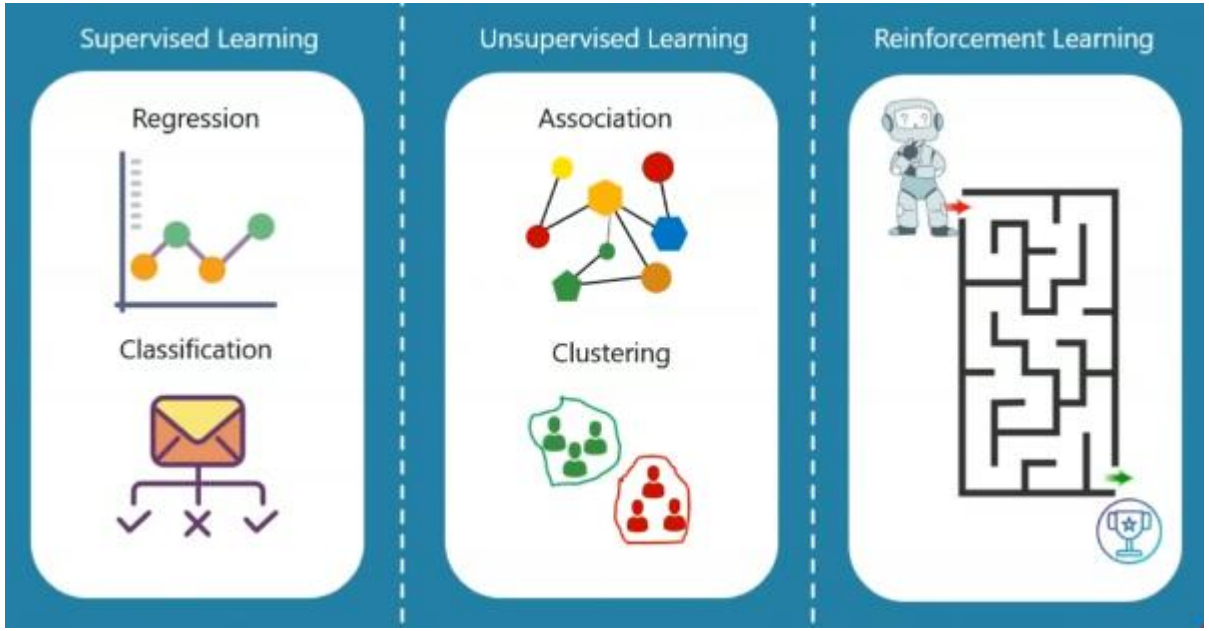


Introduction To Machine Learning

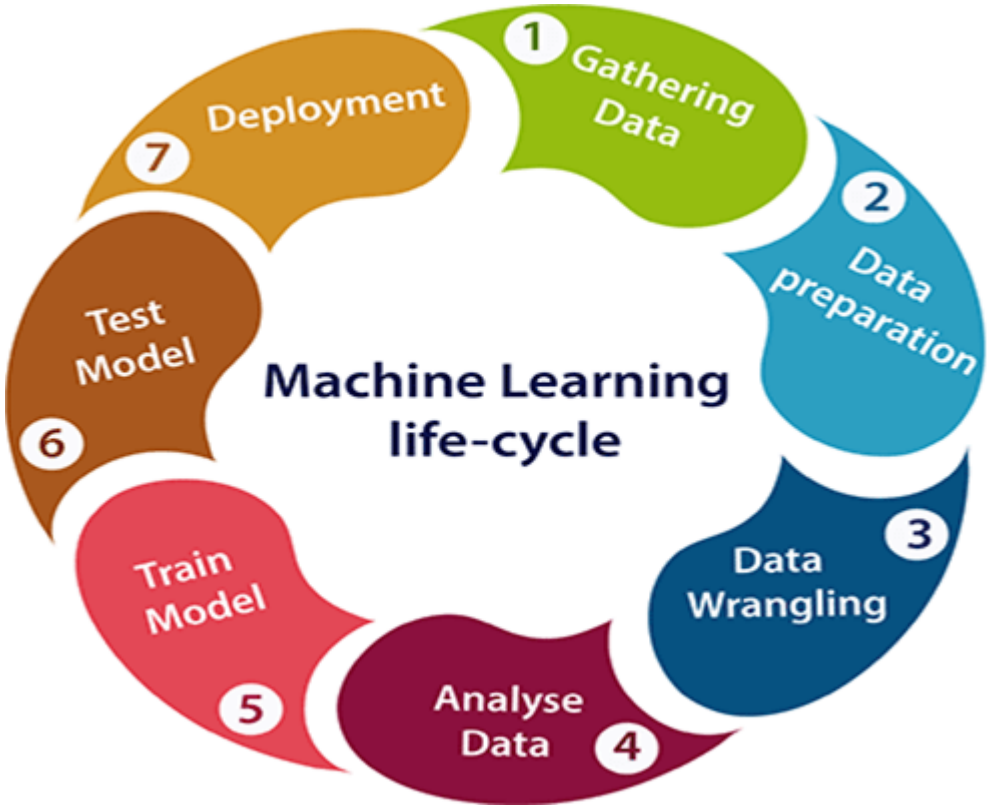


Introduction To Machine Learning

- Supervised Learning
  - Regression
  - Classification
- Unsupervised Learning
  - Clustering
  - Association
- Reinforcement Learning



Introduction To Machine Learning



Introduction To Machine Learning

- Programming Language
  - Python
  - C++
  - R
  - …
- Many Libraries
  - Scikit-learn ← Classical machine learning
  - PyTorch
  - TensorFlow
  - Keras
  - …

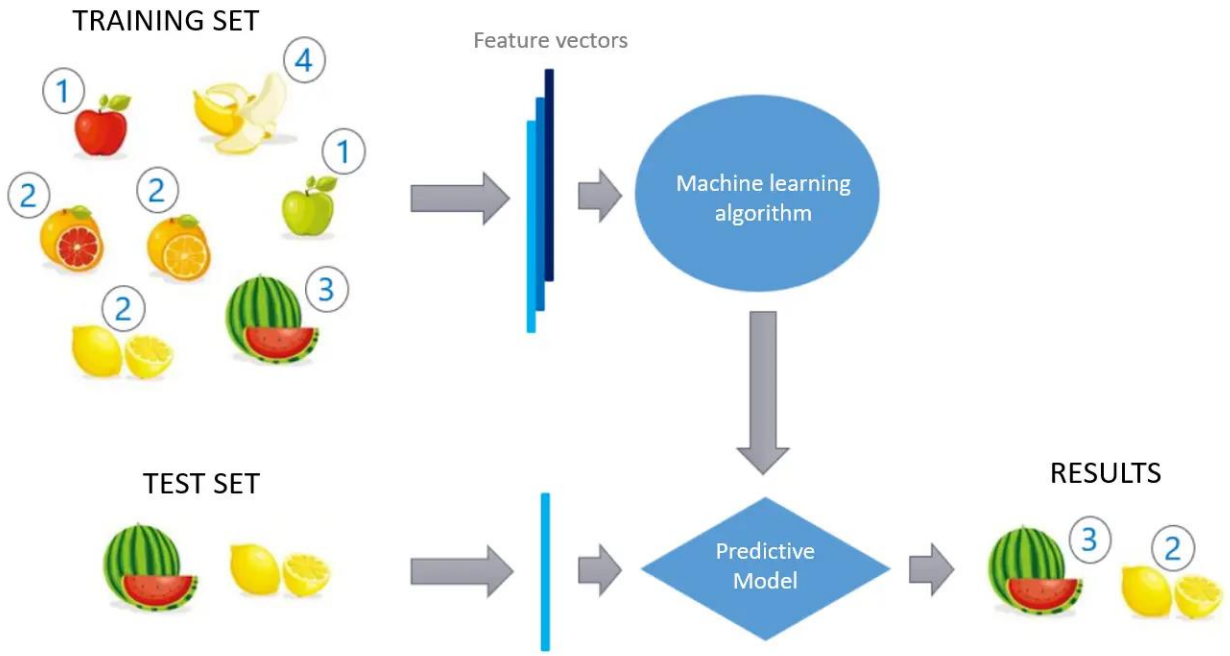
} Deep learning frameworks

Agenda

- Introduction to machine learning
- Supervised Learning
  - Linear Regression
  - Gradient Descent
  - Linear Regression vs Logistic Regression

Supervised Learning

- Apply what has been learned in the past to new data using **labeled examples** to predict future events.



Agenda

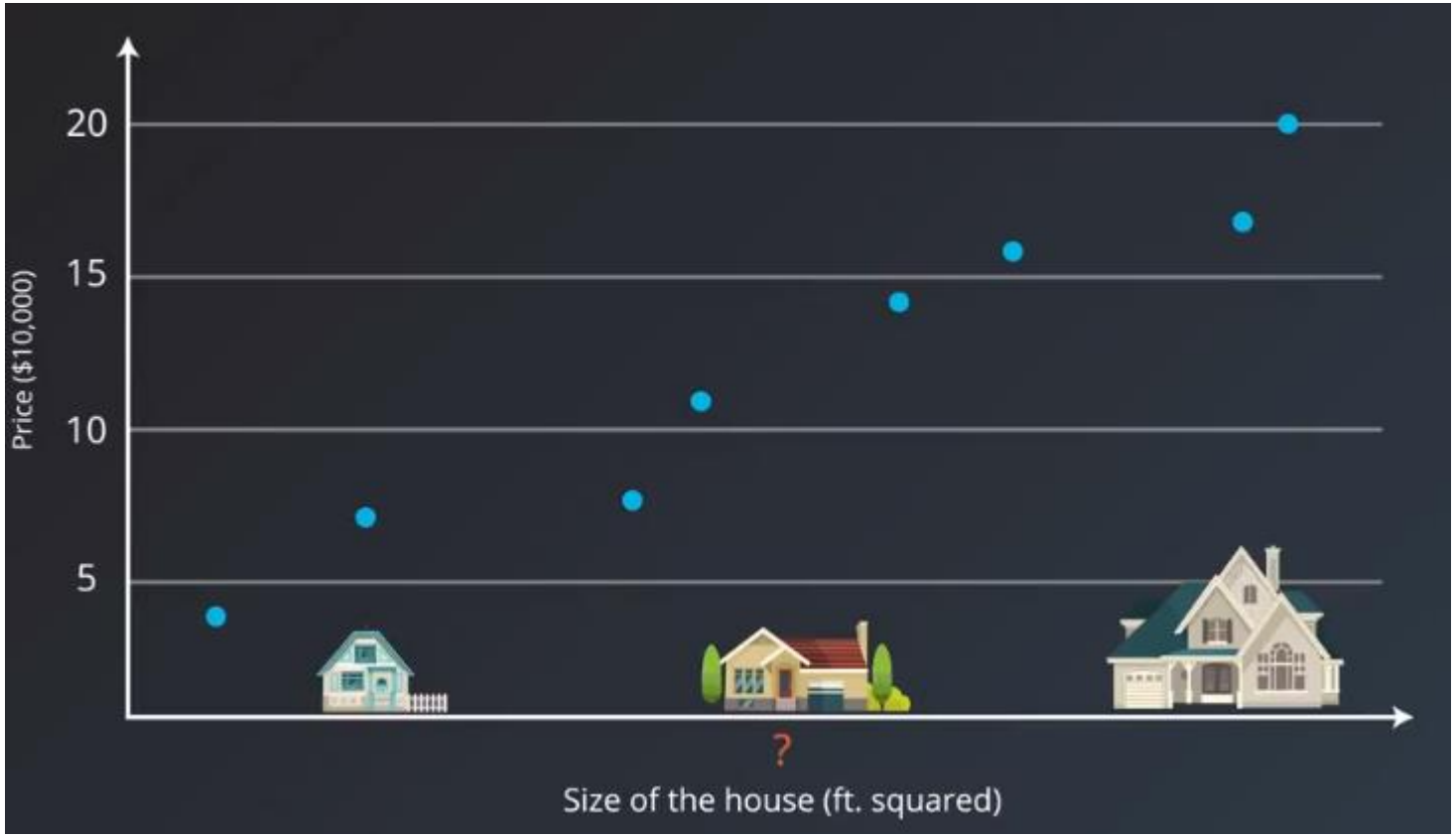
- Introduction to machine learning
- Supervised Learning
  - Linear Regression
  - Gradient Descent
  - Linear Regression vs Logistic Regression

Supervised Learning: Linear Regression

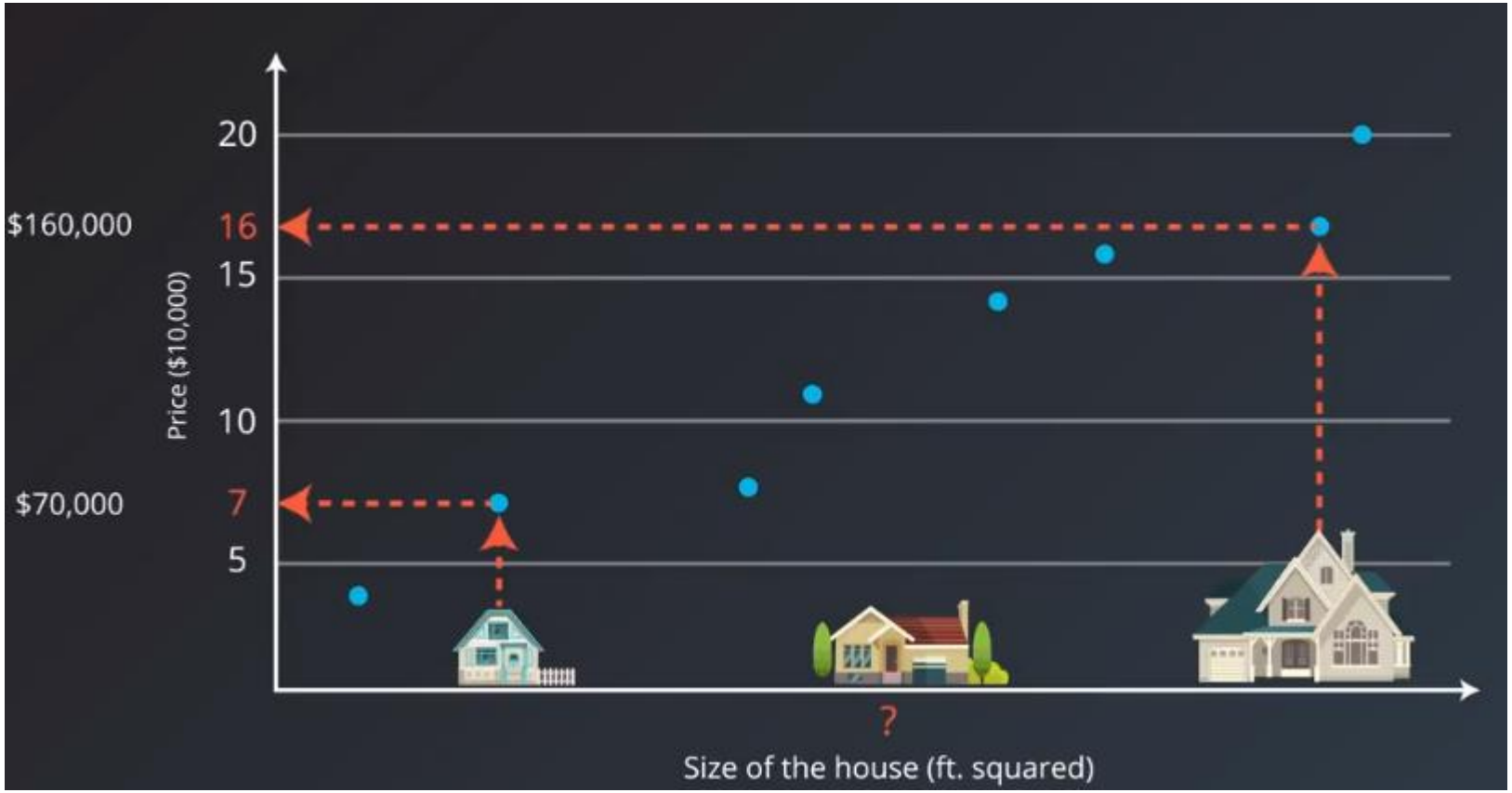
- Let’s say we want to estimate the price of the medium sized house

- What do we do?

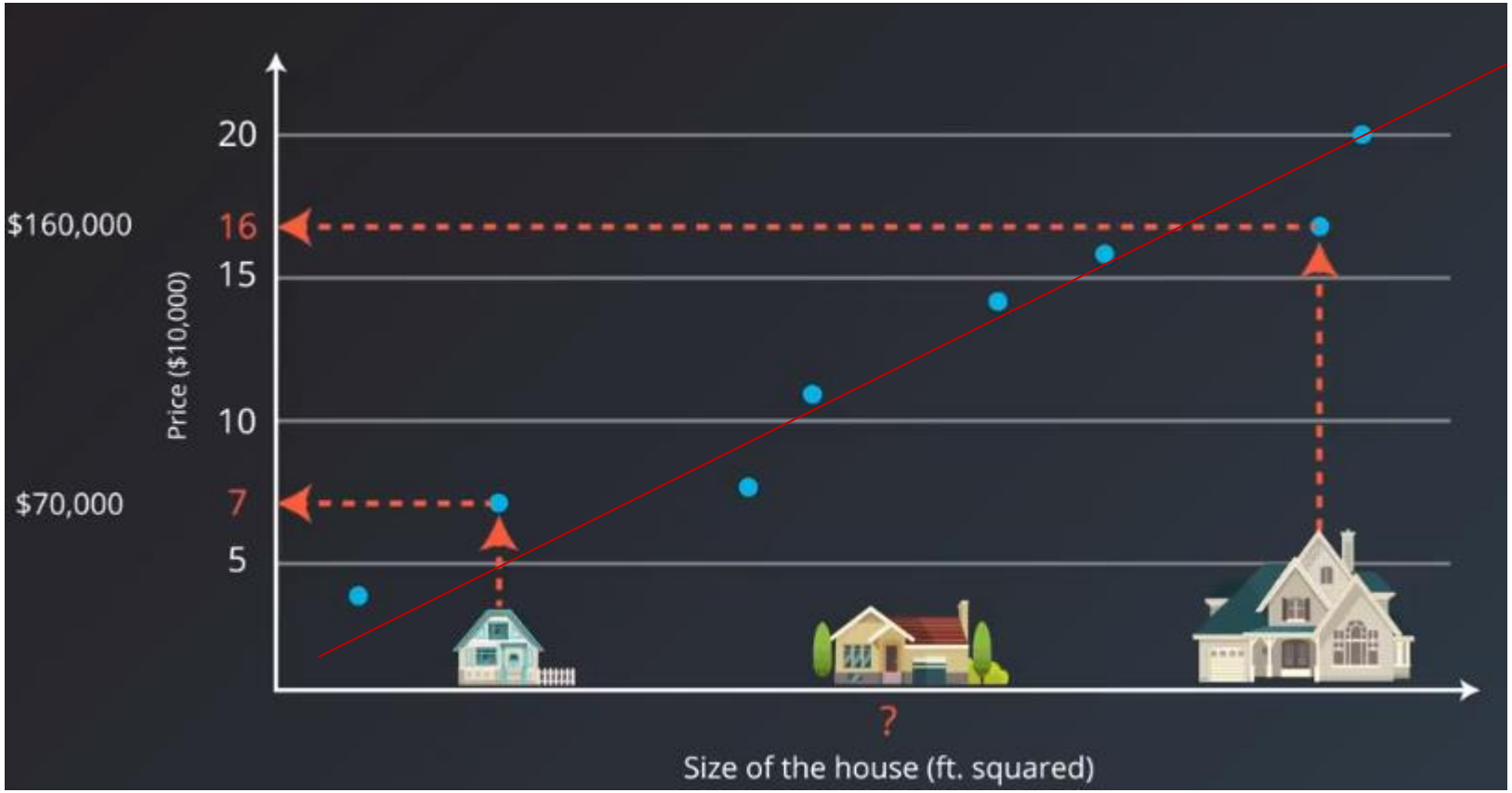
Supervised Learning: Linear Regression



Supervised Learning: Linear Regression

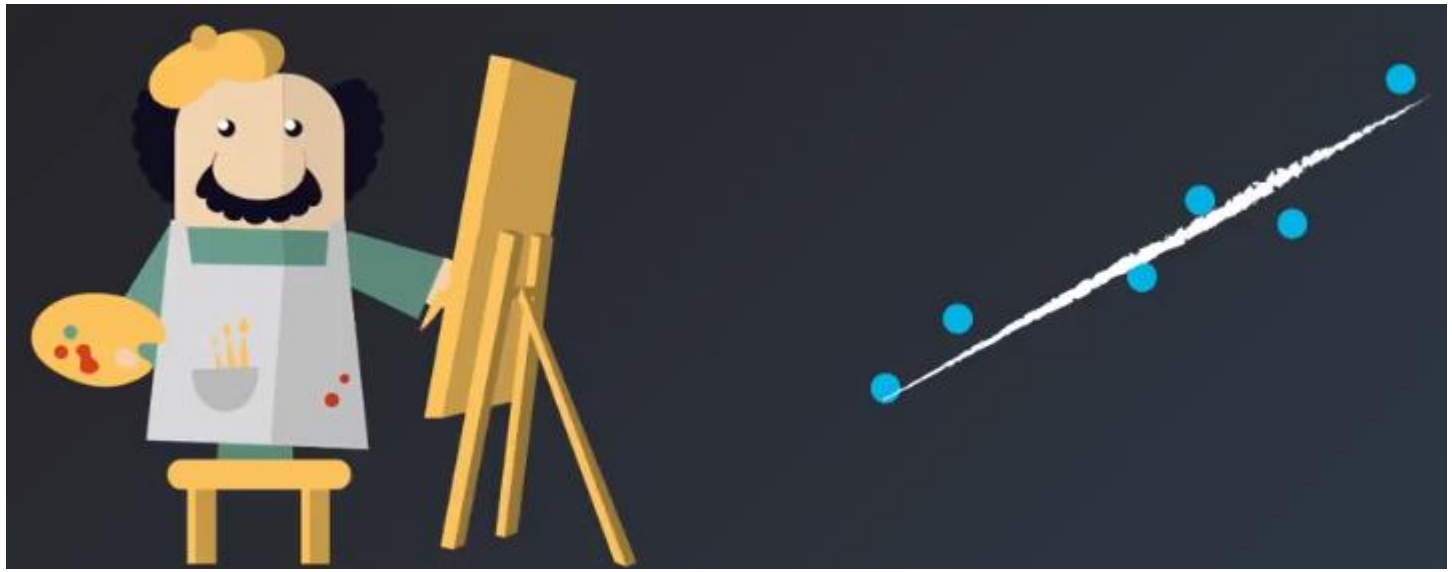


Supervised Learning: Linear Regression



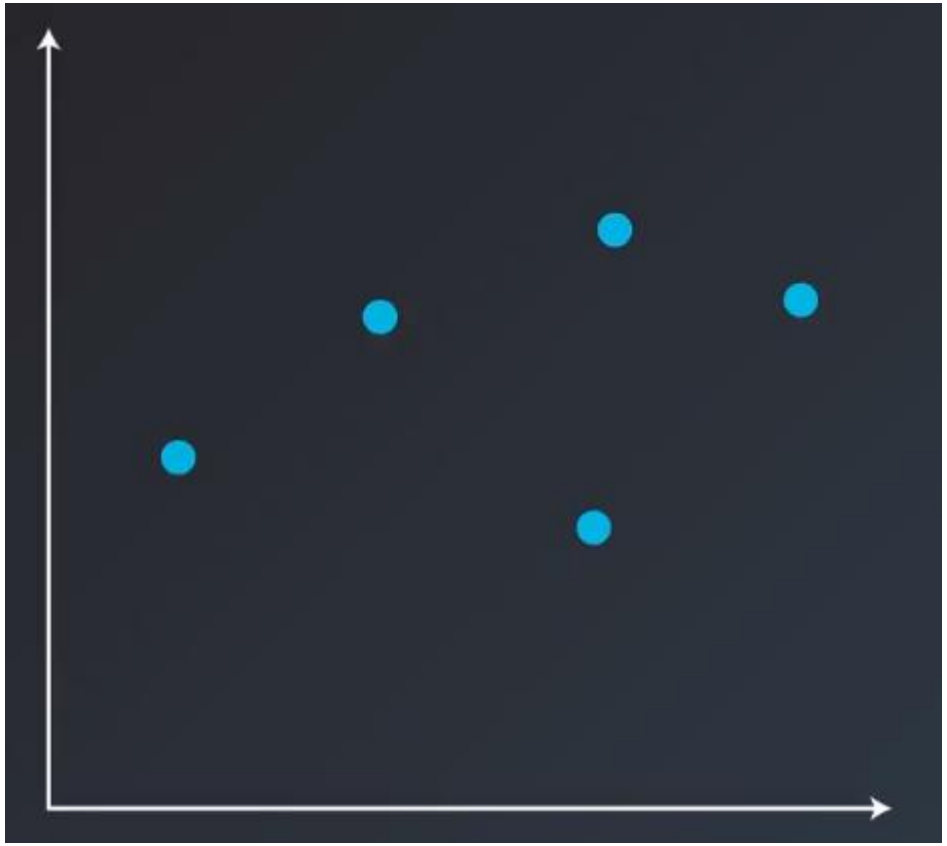
Supervised Learning: Linear Regression

- You can think of Linear Regression as a painter drawing best fit line through your data



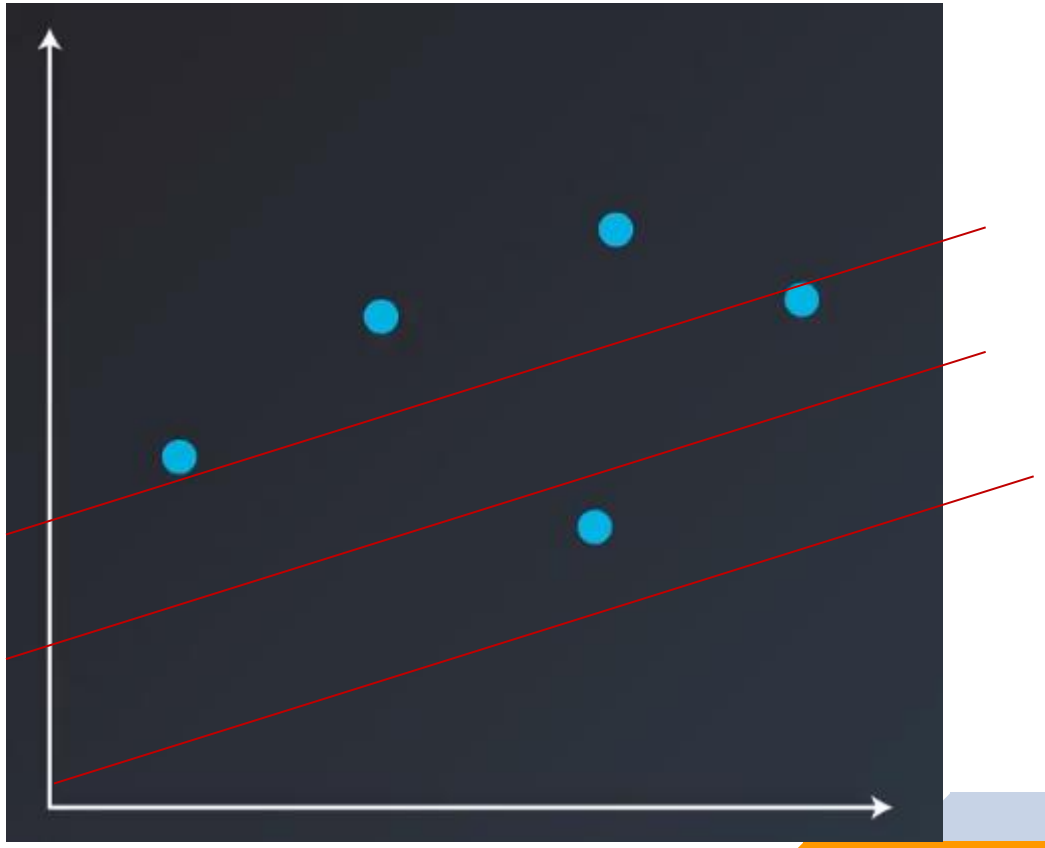
Supervised Learning: Linear Regression

- Now how do we find that line?



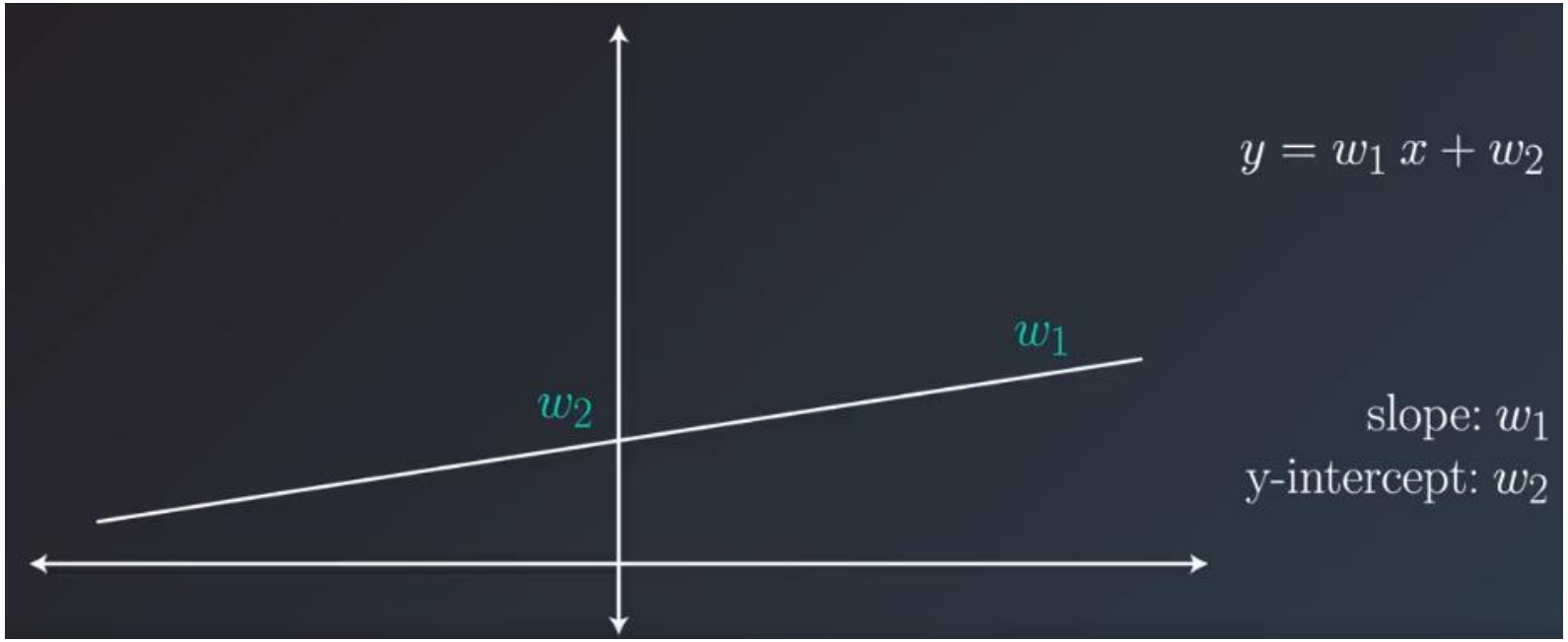
Supervised Learning: Linear Regression

- Now how do we find that line?



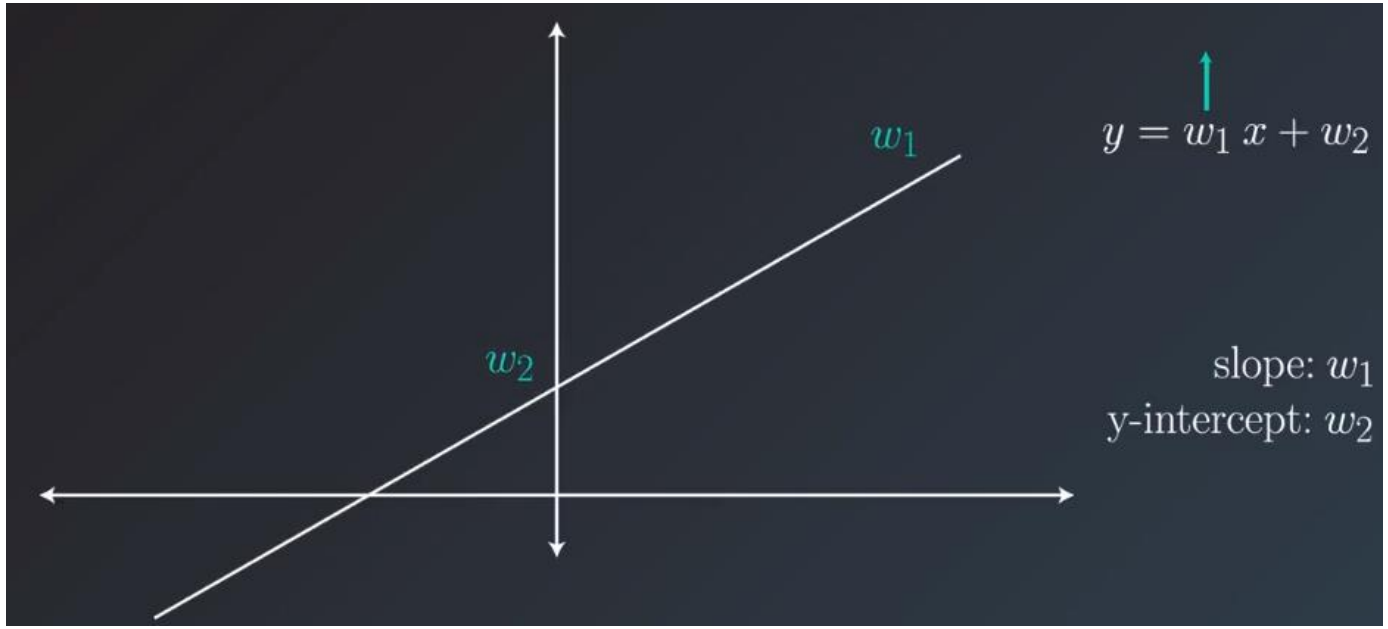
Supervised Learning: Linear Regression

- Now…
- How do we move that line (in mathematical terms)



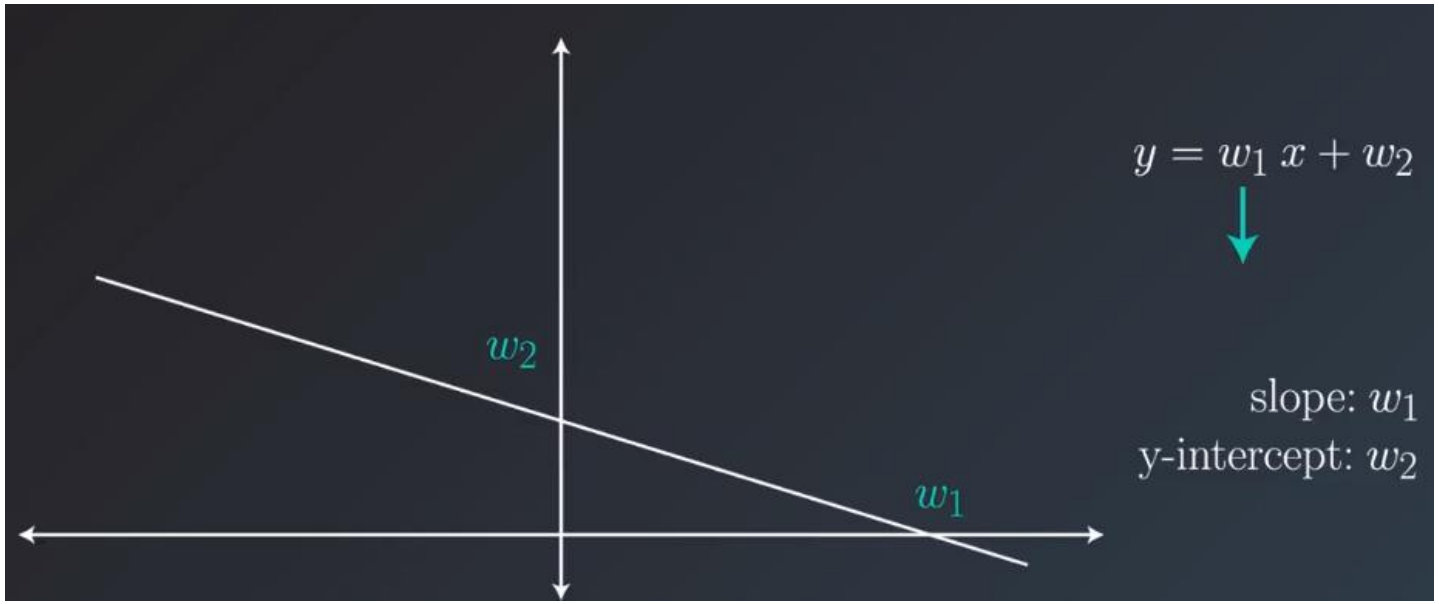
Supervised Learning: Linear Regression

- Increase the slope (w1)



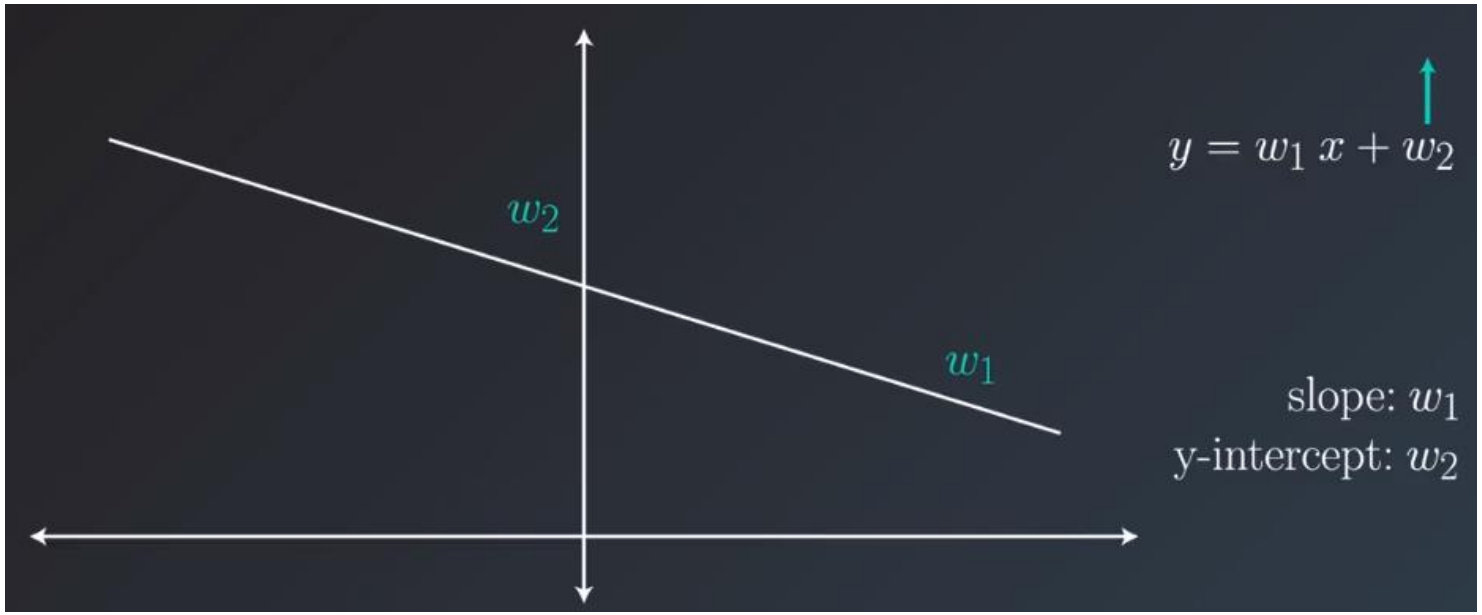
Supervised Learning: Linear Regression

- Decrease the slope (w1)



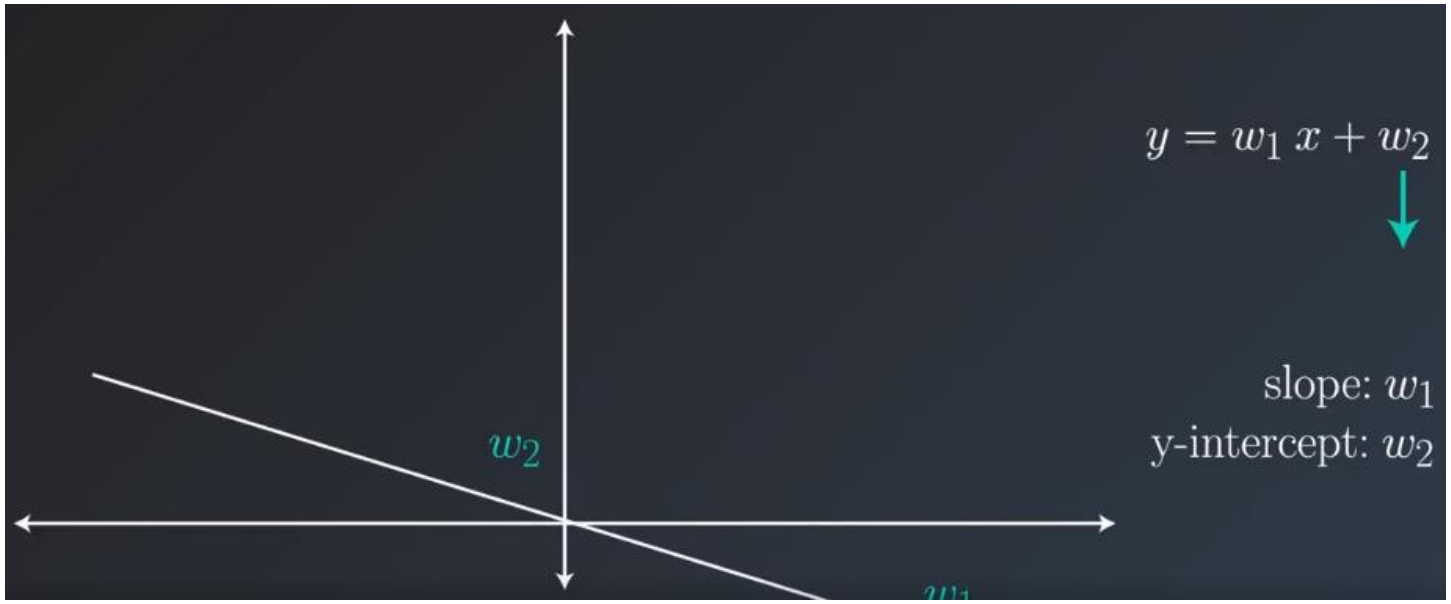
Supervised Learning: Linear Regression

- Increase y-intercept (w2)



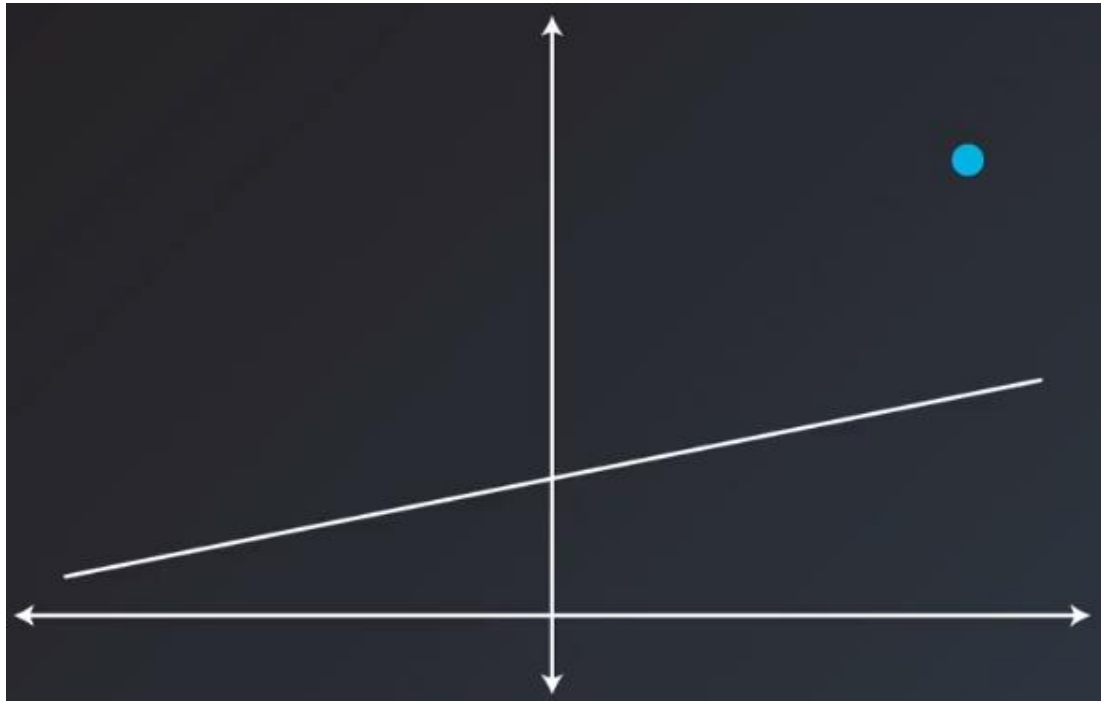
Supervised Learning: Linear Regression

- Decrease y-intercept (w2)



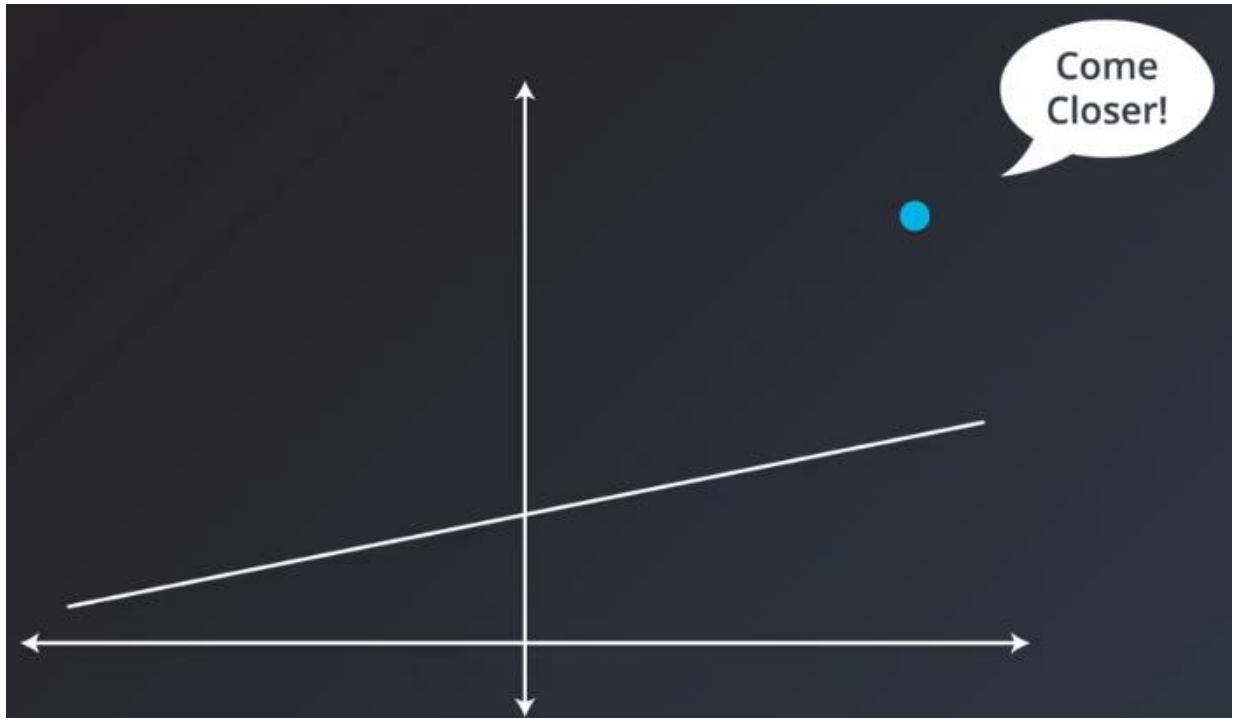
Supervised Learning: Linear Regression

- Now that we know how to move the line.
- Its time to move the line towards the point
- For that we have two techniques:
  - Absolute trick
  - Square trick

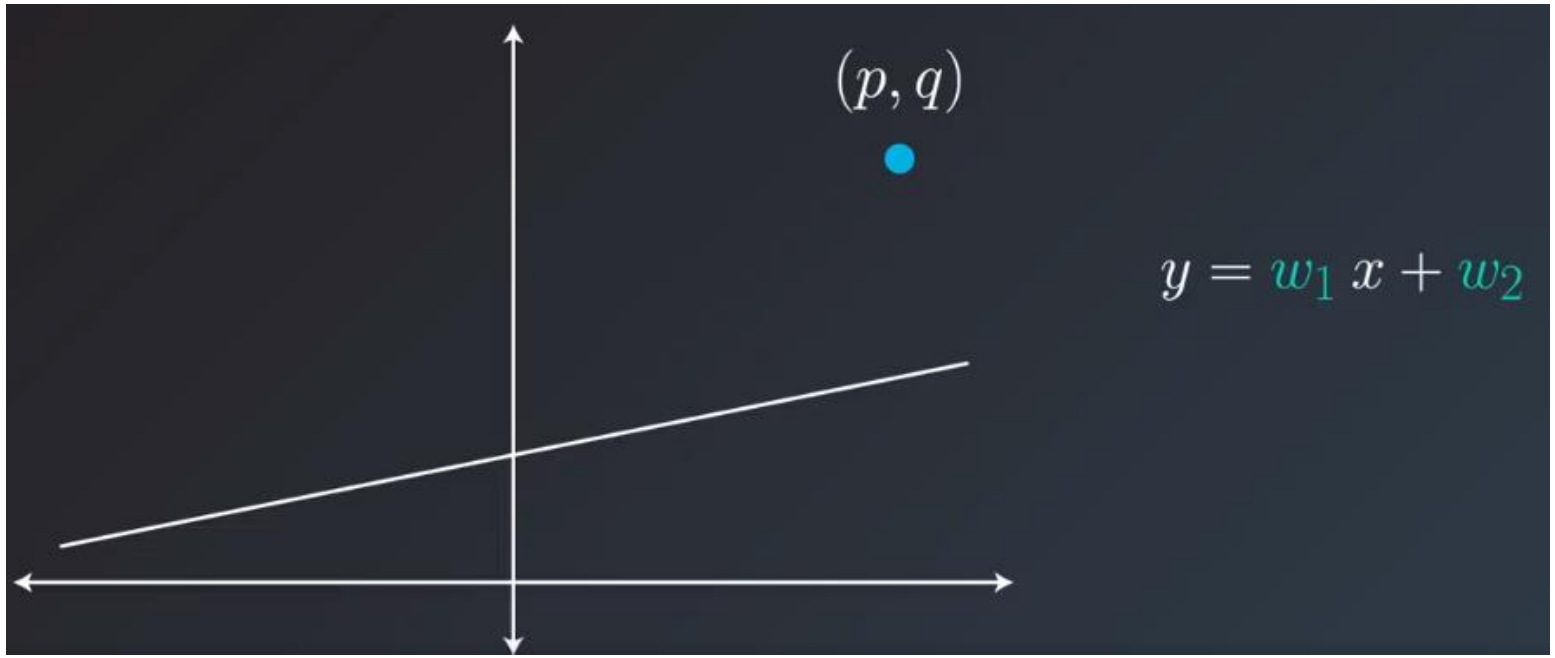


Supervised Learning: Linear Regression

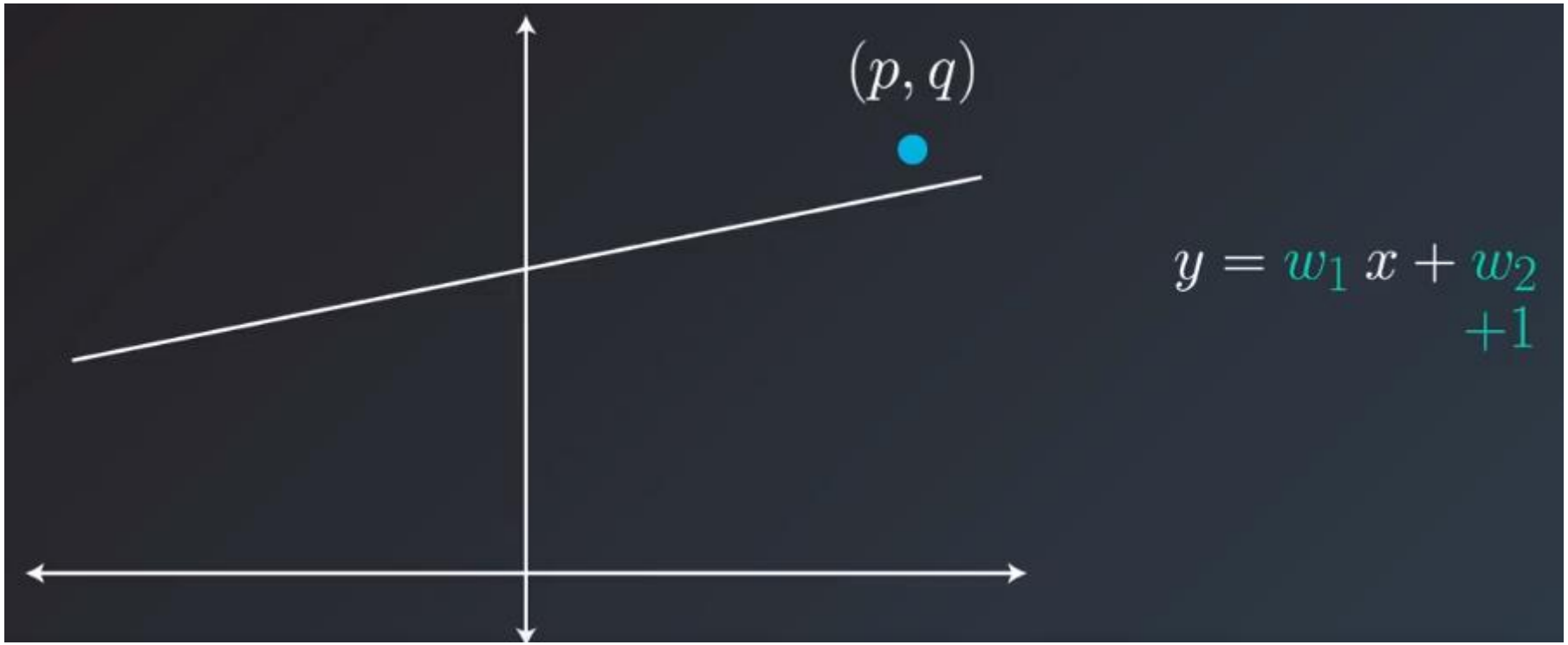
- Absolute trick
- Square trick



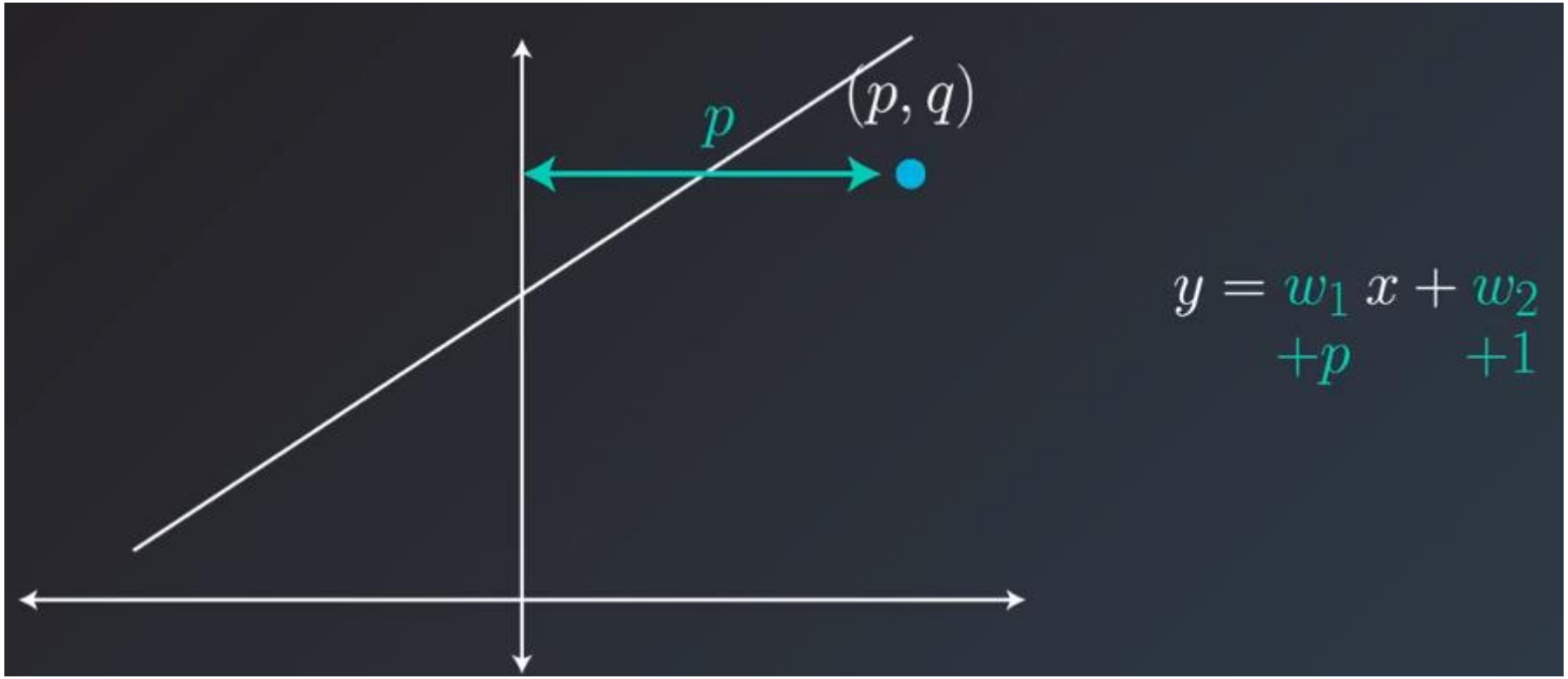
Supervised Learning: Linear Regression



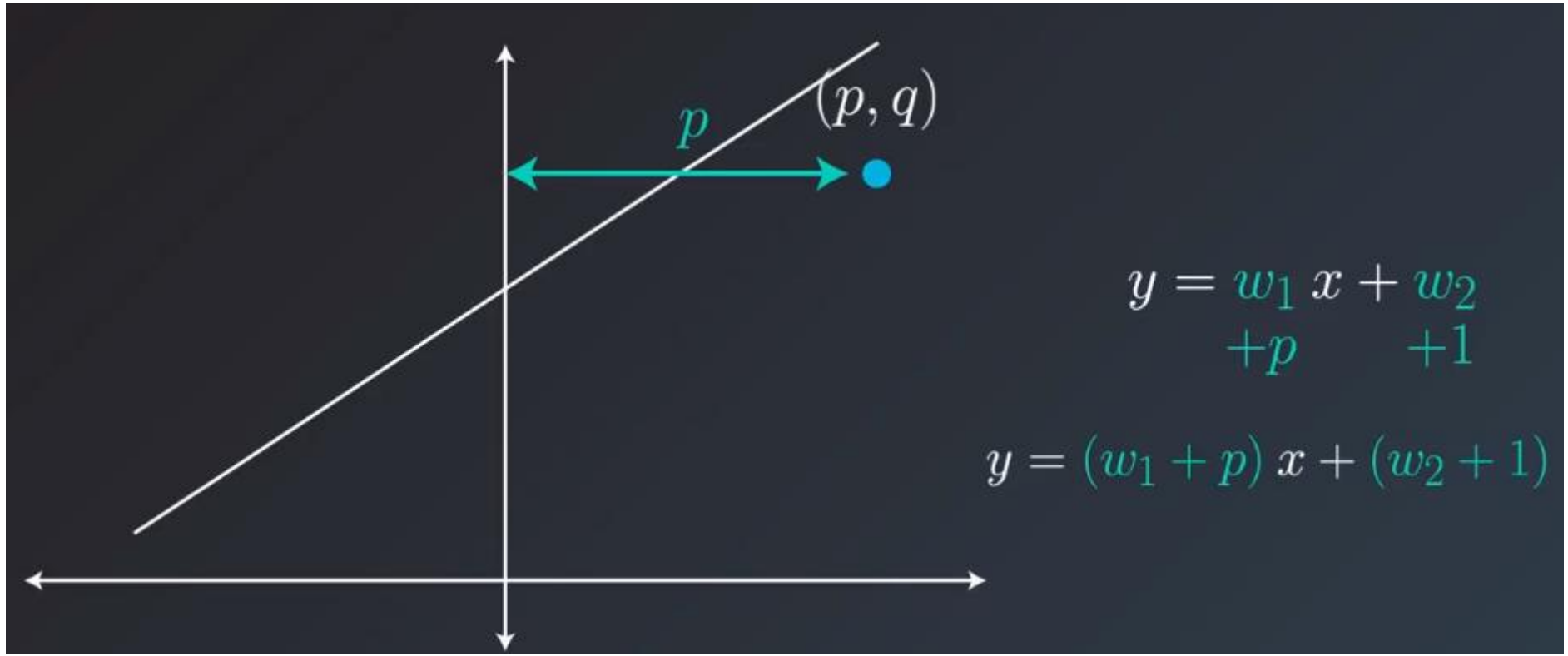
Supervised Learning: Linear Regression



Supervised Learning: Linear Regression



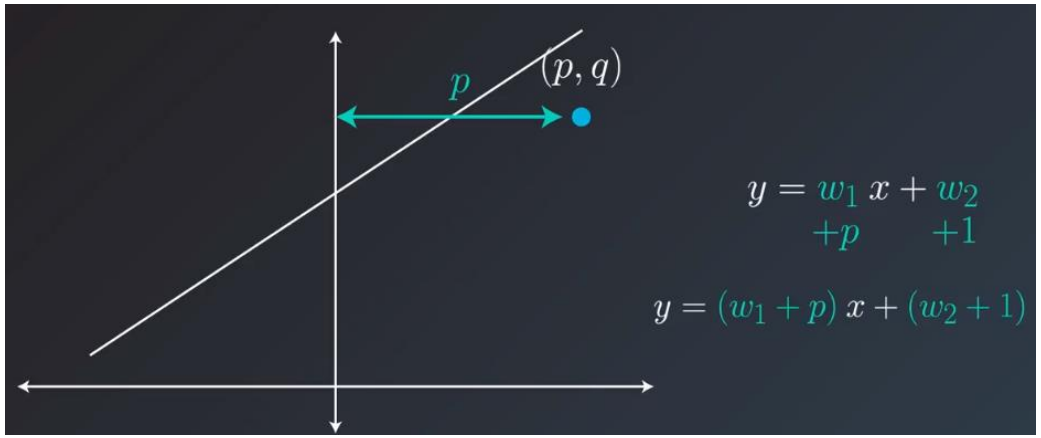
Supervised Learning: Linear Regression



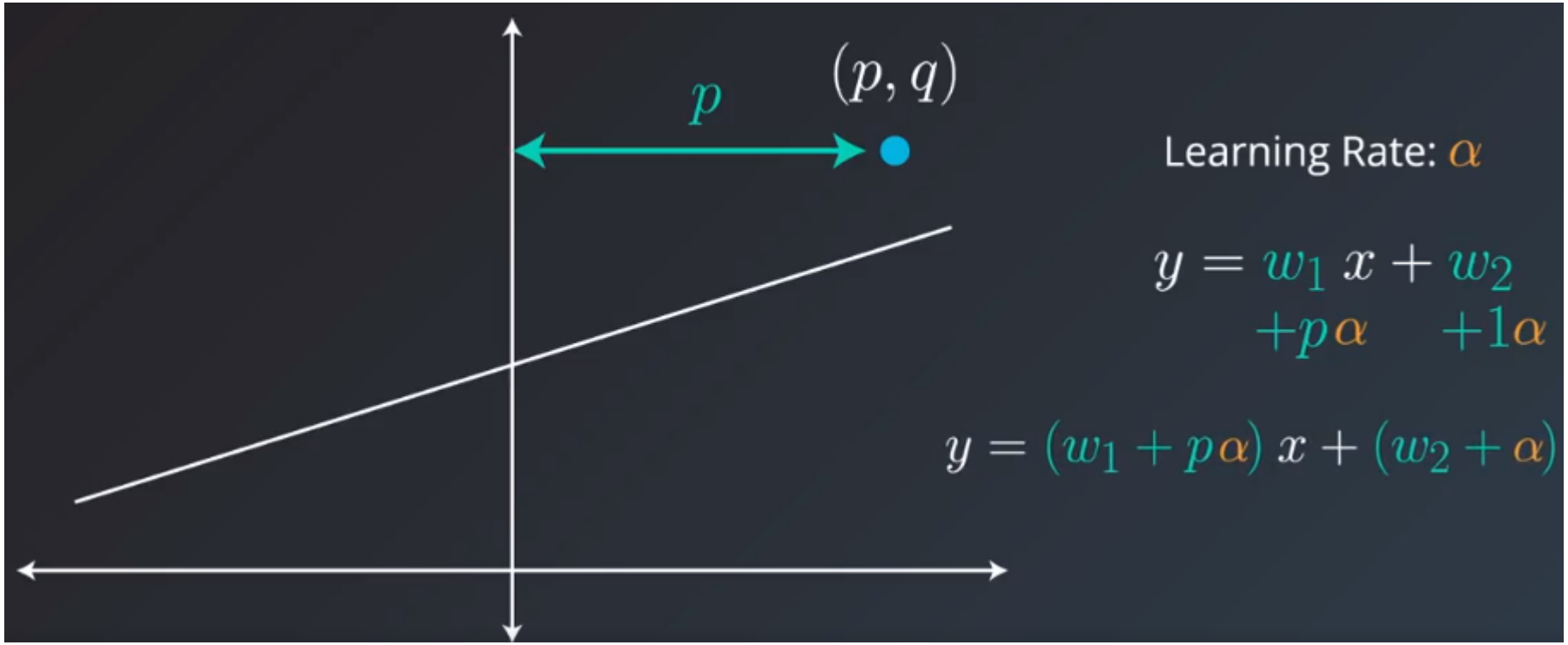
Supervised Learning: Linear Regression

- We successfully moved the line towards the point.
- Notice however that we moved the line too much. And the line has now moved past the point

- We don't like to do that in machine learning. So we have to take smaller steps

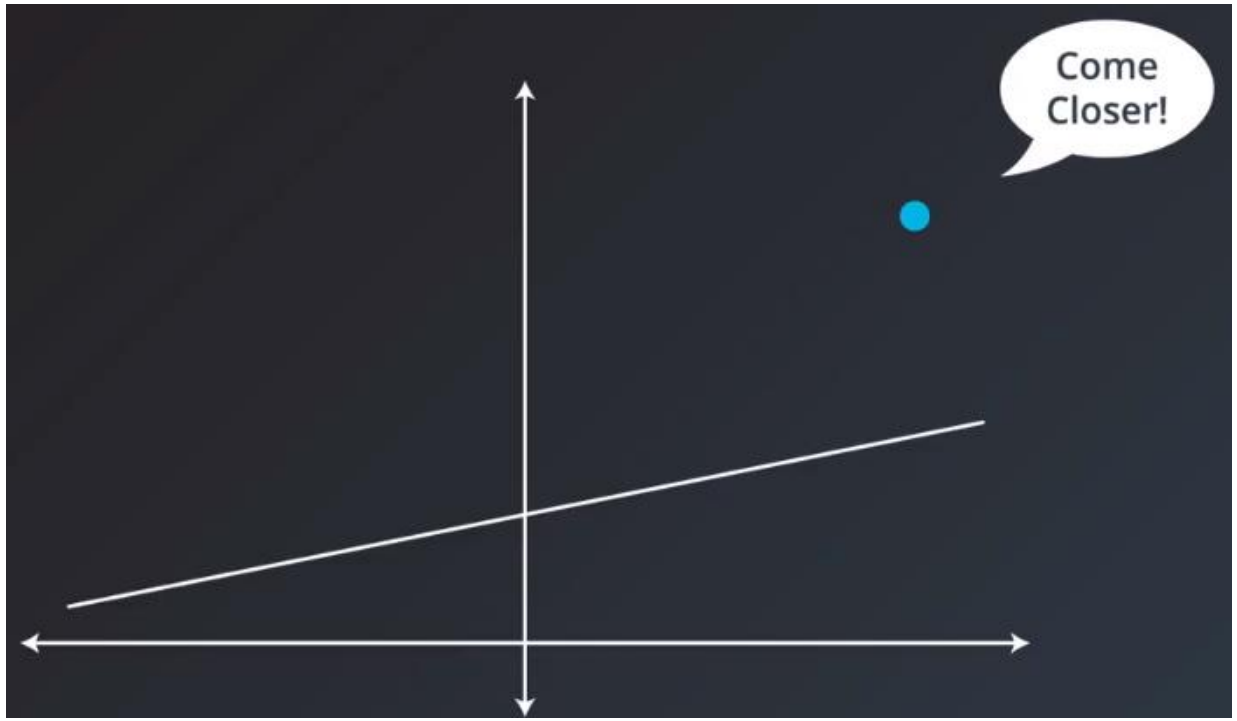


Supervised Learning: Linear Regression



Supervised Learning: Linear Regression

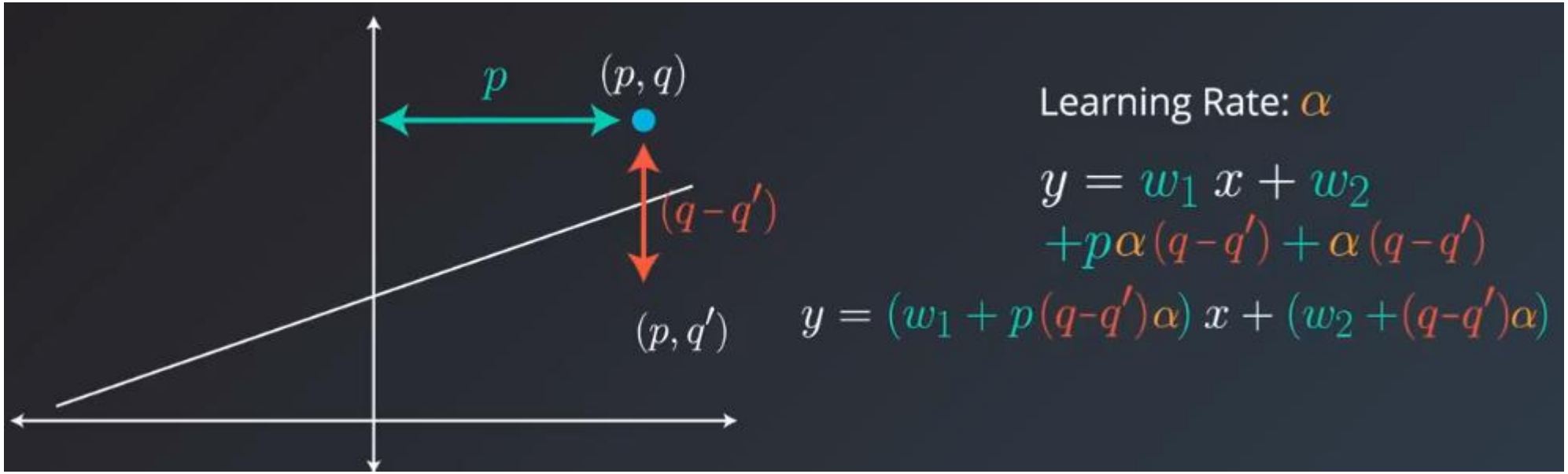
- Absolute trick
- Square trick



Supervised Learning: Linear Regression

- The absolute trick doesn’t consider how far or close the point is from the line.
- Taking the same step everytime

Supervised Learning: Linear Regression



Agenda

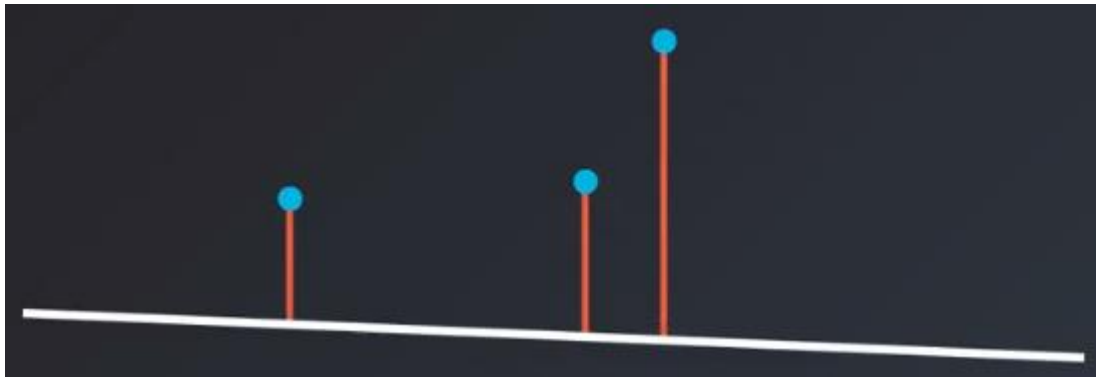
- Introduction to machine learning
- Supervised Learning
  - Linear Regression
  - Gradient Descent
  - Linear Regression vs Logistic Regression

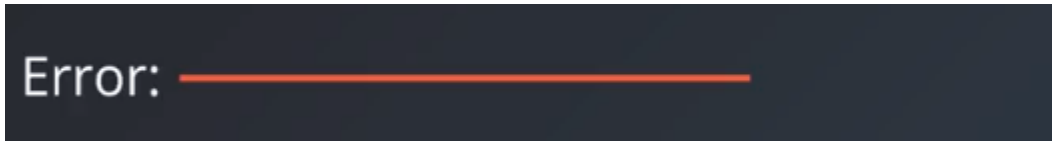
Supervised Learning: Gradient Descent

Gradient Descent

Supervised Learning: Gradient Descent

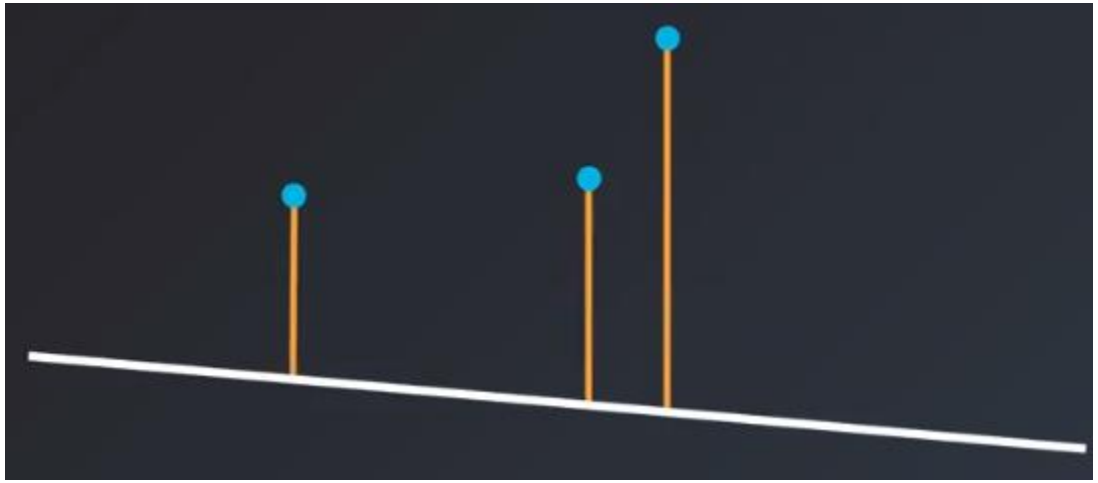
Gradient Descent

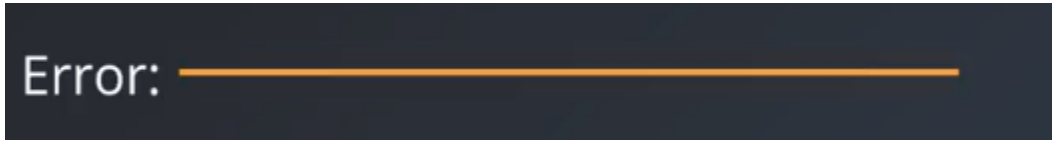




Supervised Learning: Gradient Descent

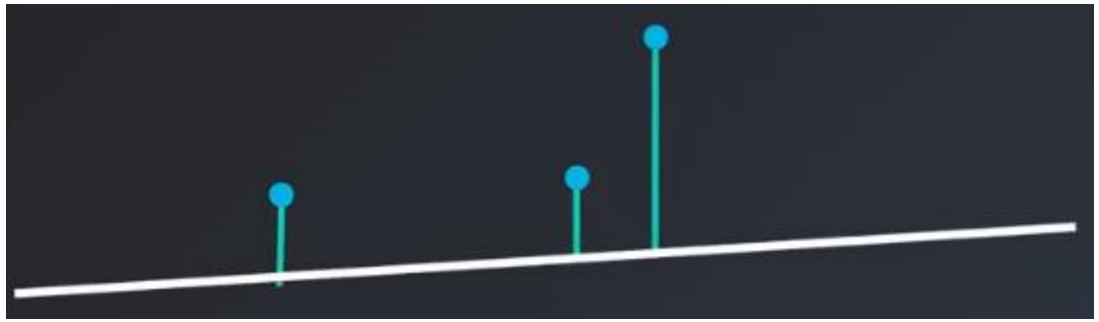
Gradient Descent

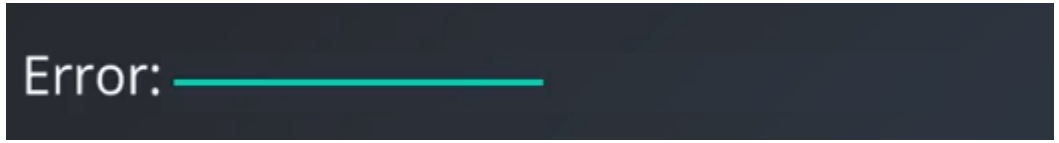




Supervised Learning: Gradient Descent

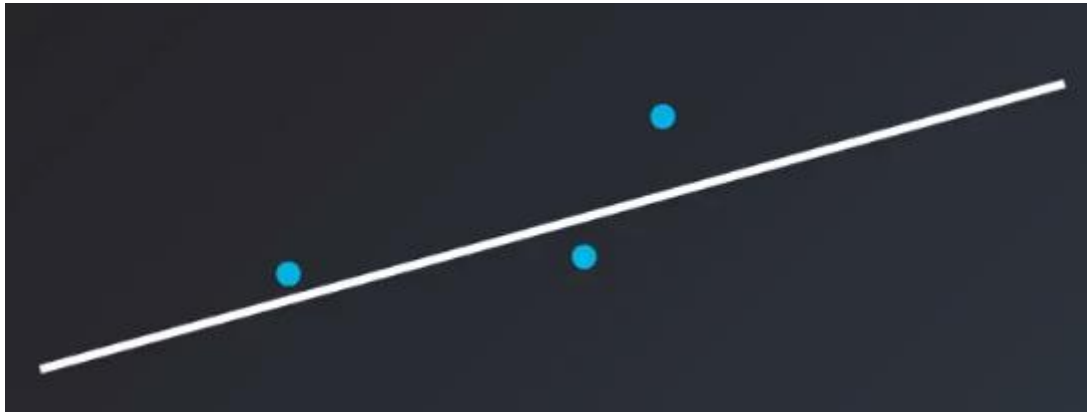
Gradient Descent





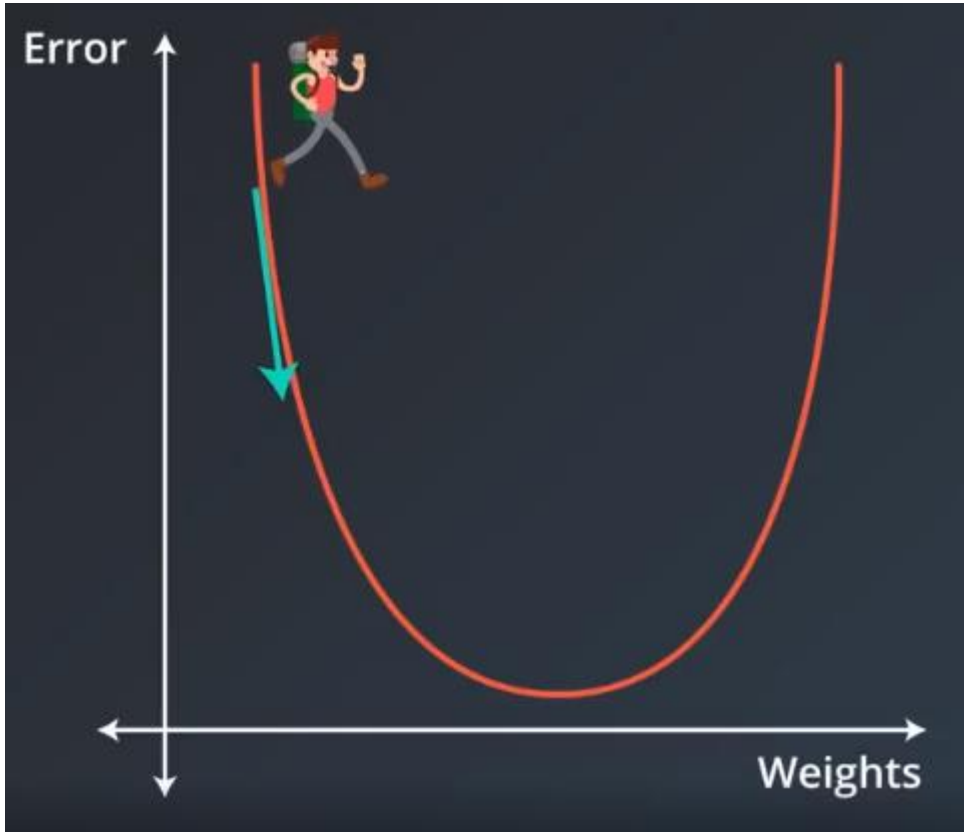
Supervised Learning: Gradient Descent

Gradient Descent



Supervised Learning: Linear Regression

Gradient Descent



Error Function

- Gradient of Error Function

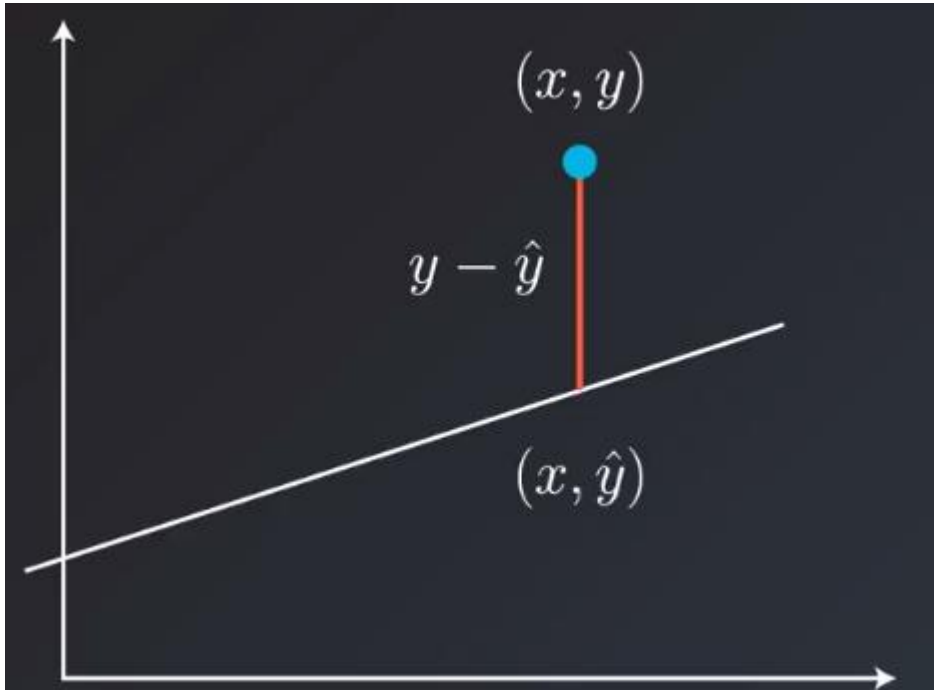
$$w_i \rightarrow w_i - \alpha \frac{\partial}{\partial w_i} Error$$

Supervised Learning: Linear Regression

- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error

Supervised Learning: Linear Regression

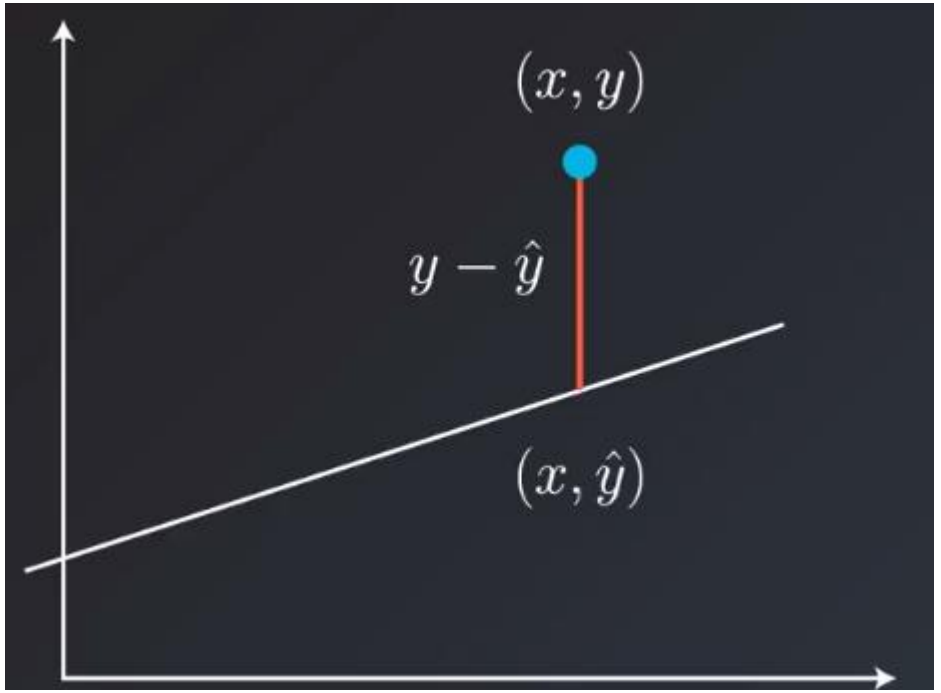
- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error



Supervised Learning: Linear Regression

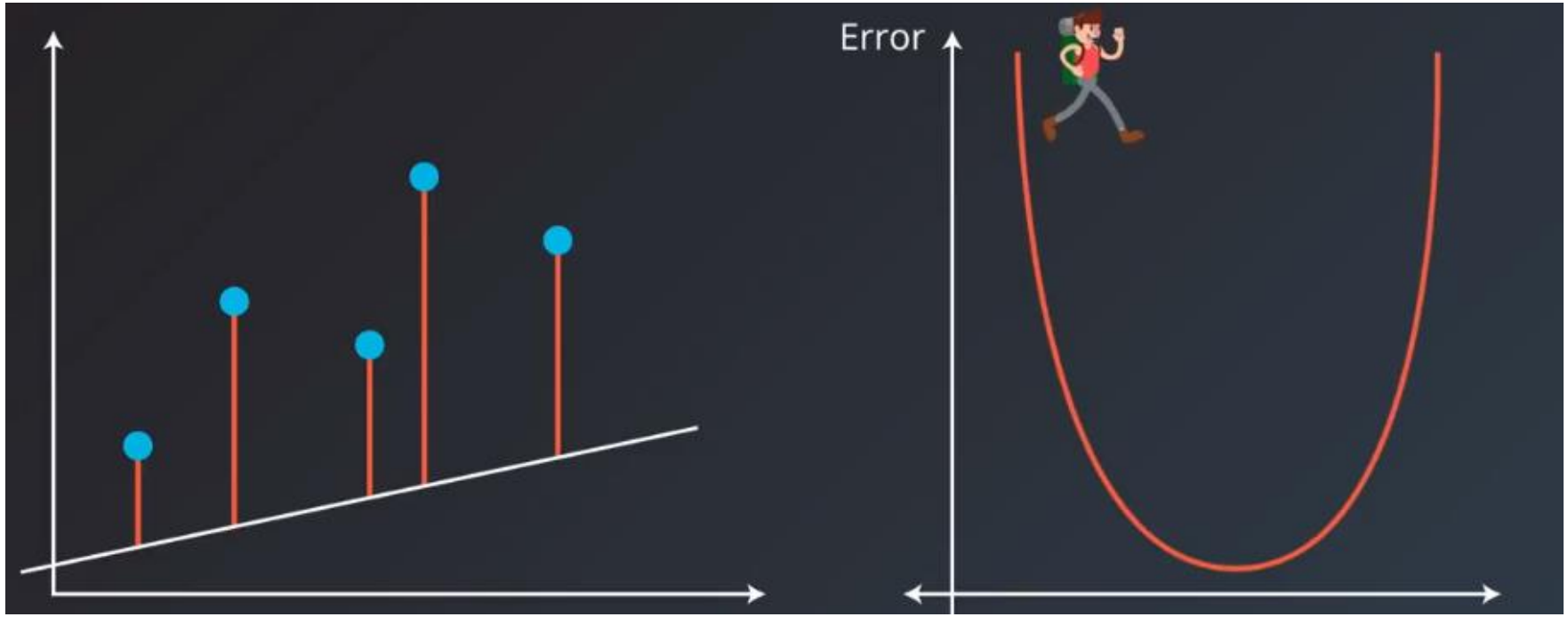
- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error

$$Error = \frac{1}{m} \sum_{i=1}^{m} |y - \hat{y}|$$



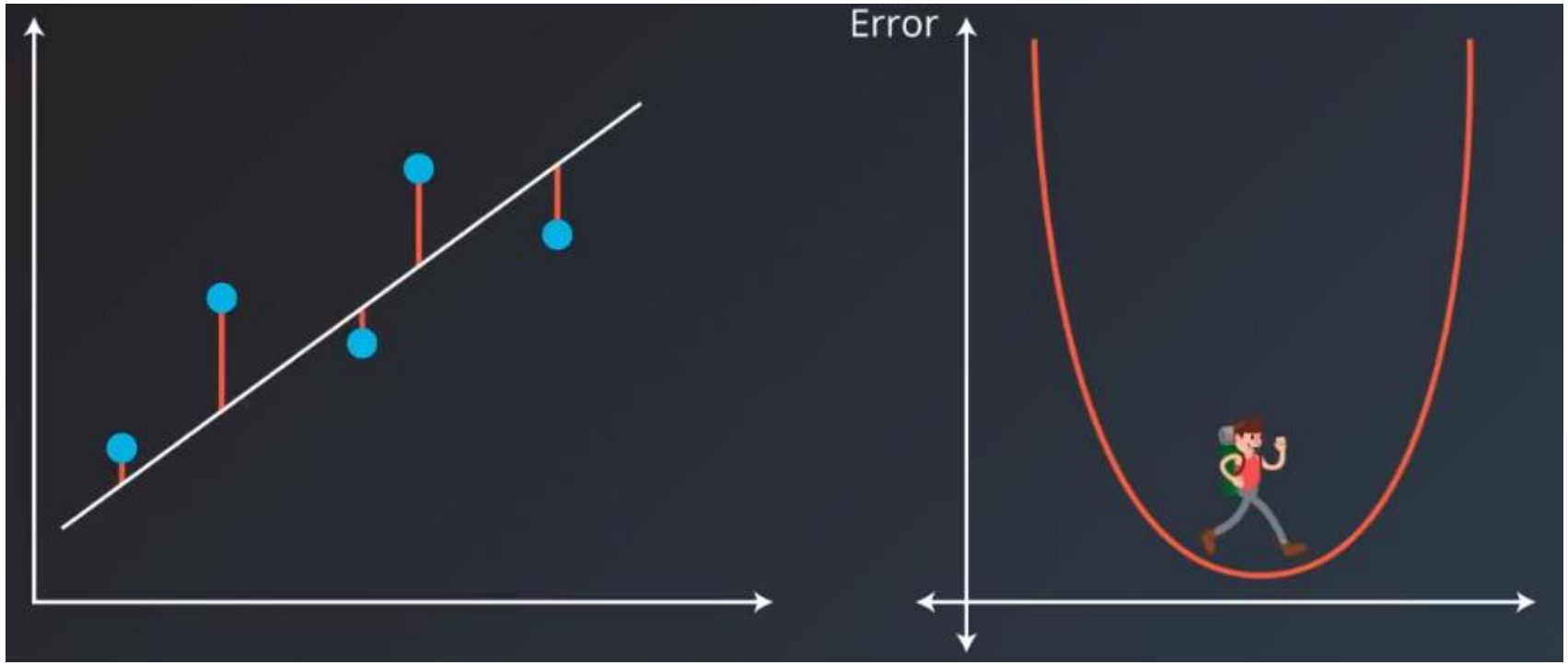
Supervised Learning: Linear Regression

- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error



Supervised Learning: Linear Regression

- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error



Supervised Learning: Linear Regression

- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error

$$Error = \frac{1}{2m}\sum_{i=1}^{m}(y - \hat{y})^2$$

Supervised Learning: Linear Regression

- Error Functions:
  - Mean Absolute Error
  - Mean Squared Error

1) Absolute Error Function

$$Error = \frac{1}{m} \sum_{i=1}^{m} |y - \hat{y}|$$

2) Squared Error Function

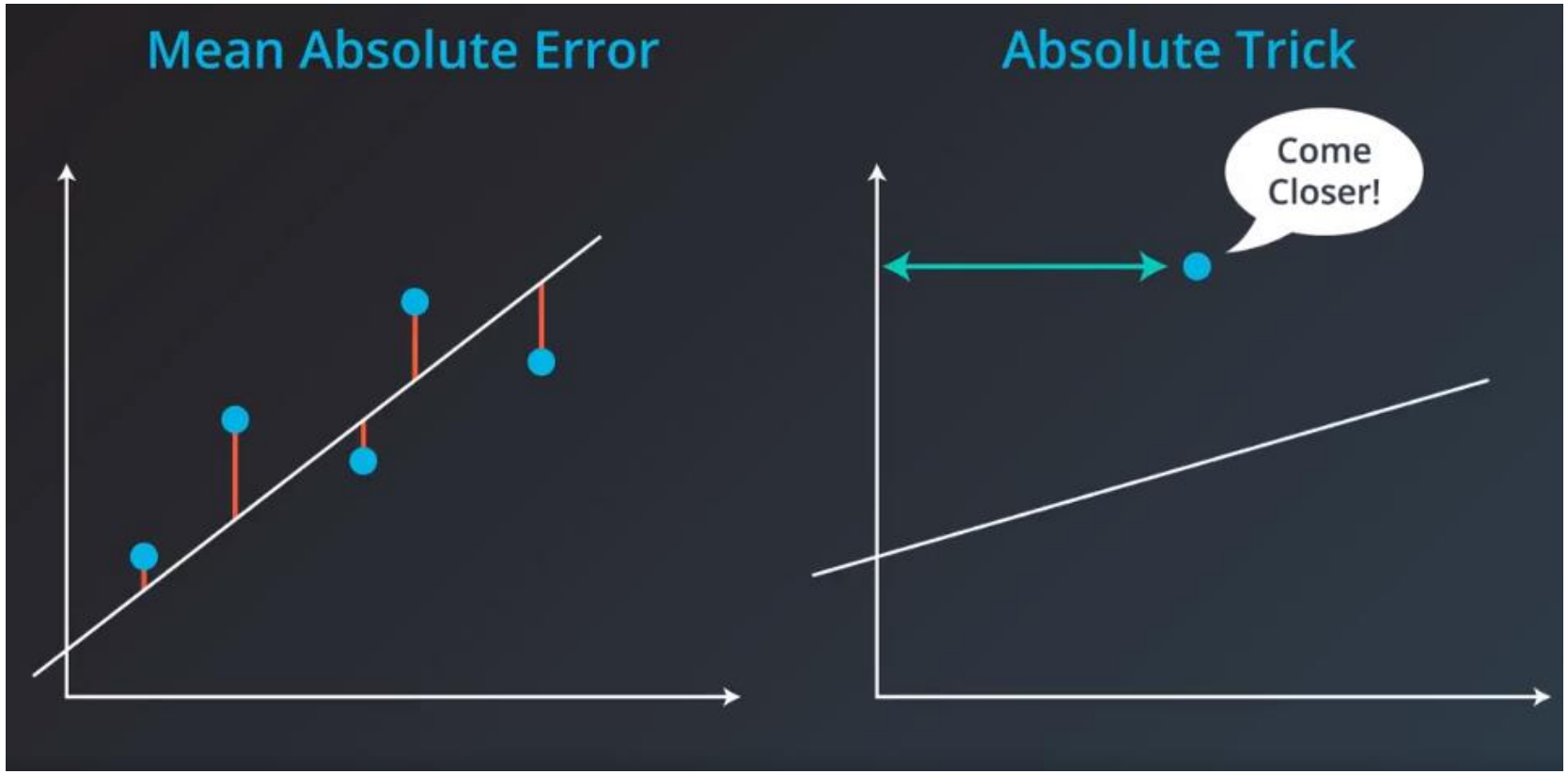
$$Error = \frac{1}{2m} \sum_{i=1}^{m} (y - \hat{y})^2$$

Supervised Learning: Linear Regression

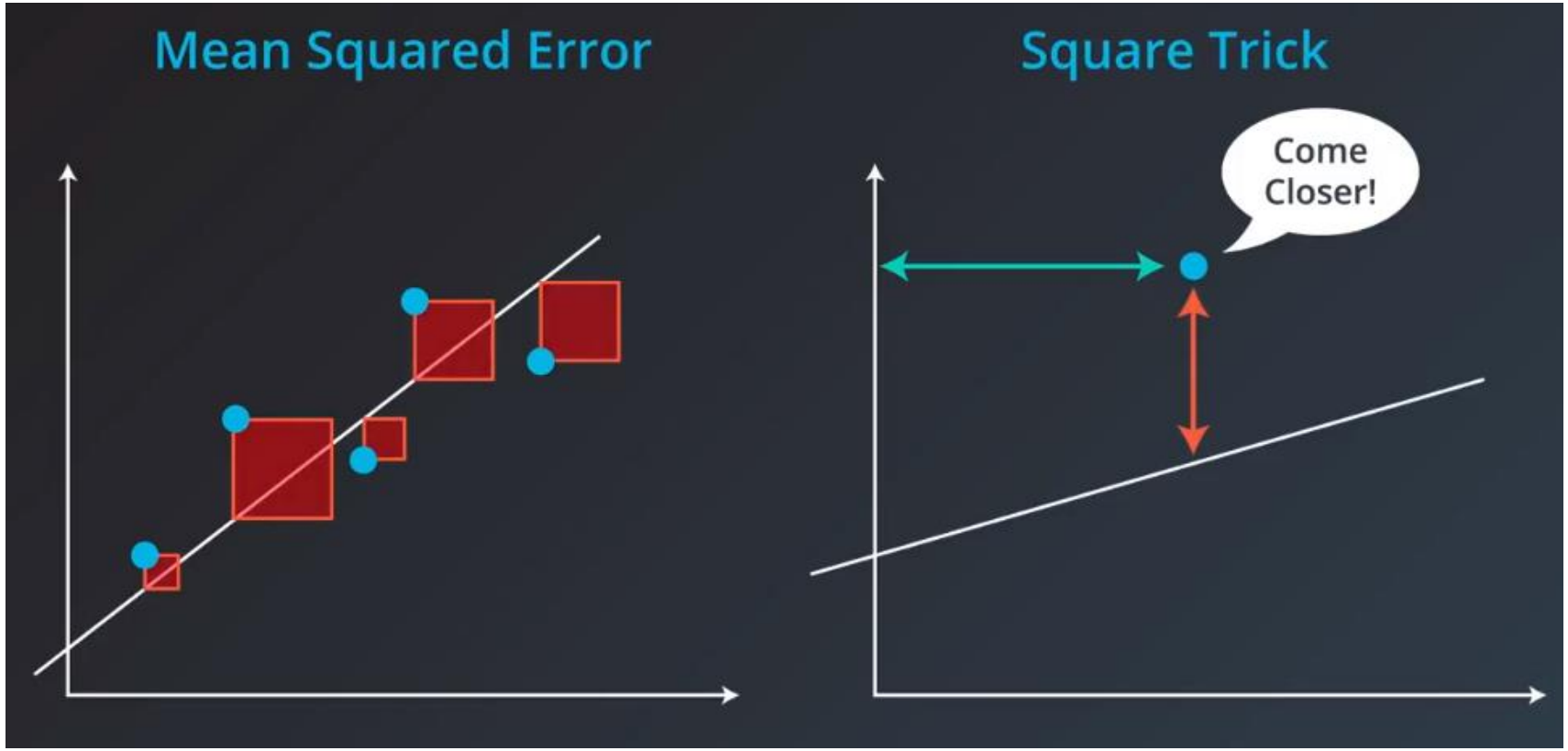
- We have learned 2 methods to decrease the error. Tricks, and Error functions.

- Turns out they are both the exact same thing

Supervised Learning: Linear Regression

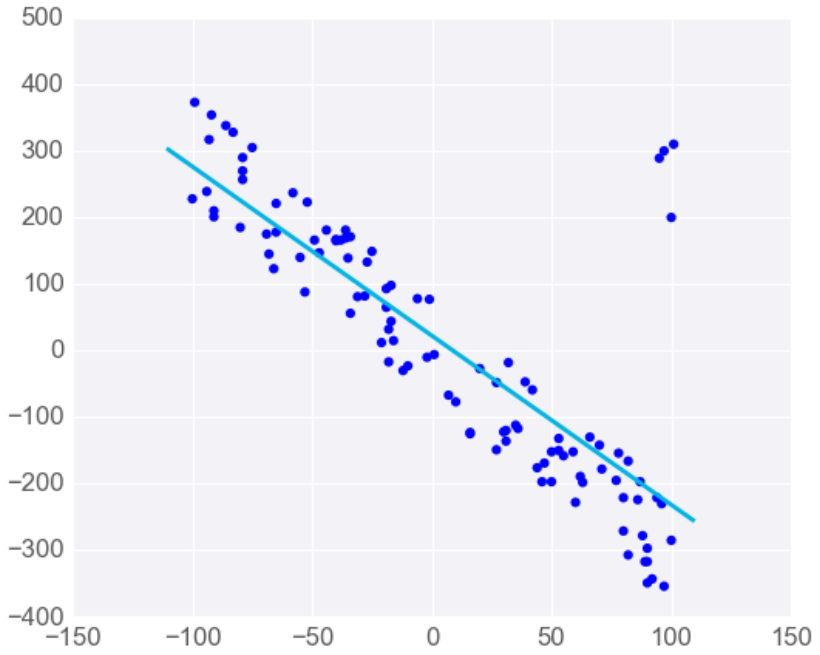


Supervised Learning: Linear Regression



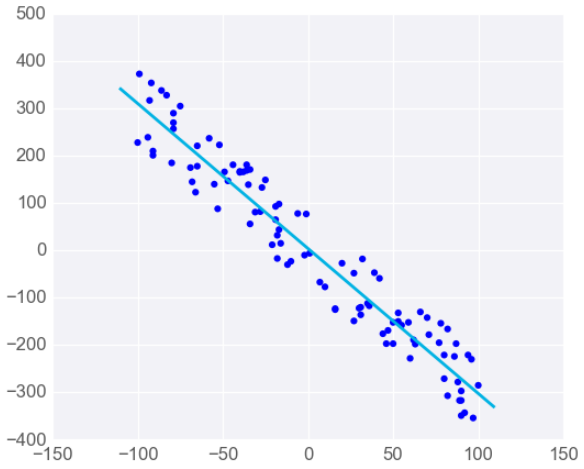
Supervised Learning: Linear Regression

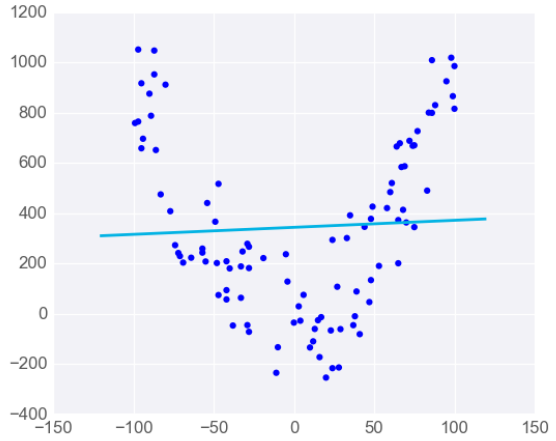
- Linear regression rules:
  - Affected by outliers.
  - It works best when data is linear



Supervised Learning: Linear Regression

- Linear regression rules:
  - Affected by outliers.
  - It works best when data is linear





Supervised Learning: Linear Regression

- Linear Regression in scikit-learn

```
>>> from sklearn.linear_model import LinearRegression
>>> model = LinearRegression()
>>> model.fit(x_values, y_values)
```

```
>>> print(model.predict([ [127], [248] ]))
[[ 438.94308857, 127.14839521]]
```

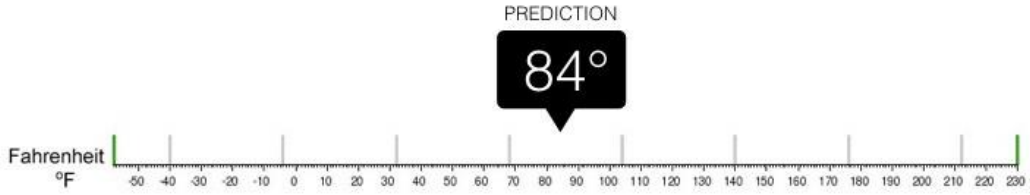
Agenda

- Introduction to machine learning
- Supervised Learning
  - Linear Regression
  - Gradient Descent
  - Linear Regression vs Logistic Regression

Supervised Learning: Linear Regression vs Logistic Regression

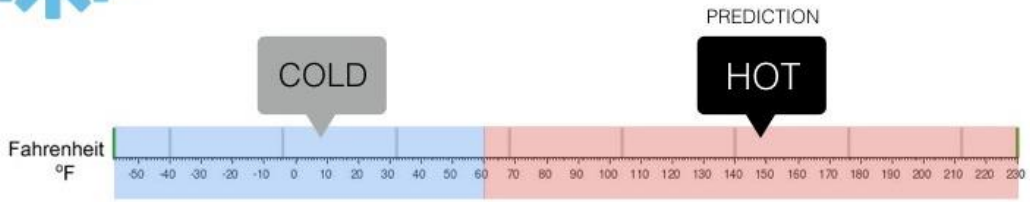
Regression

What is the temperature going to be tomorrow?



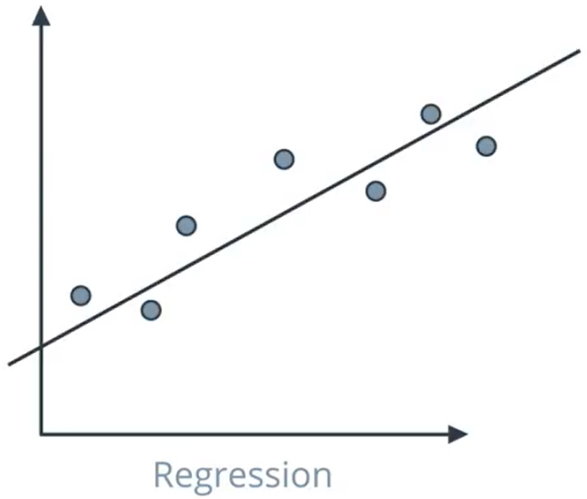
Classification

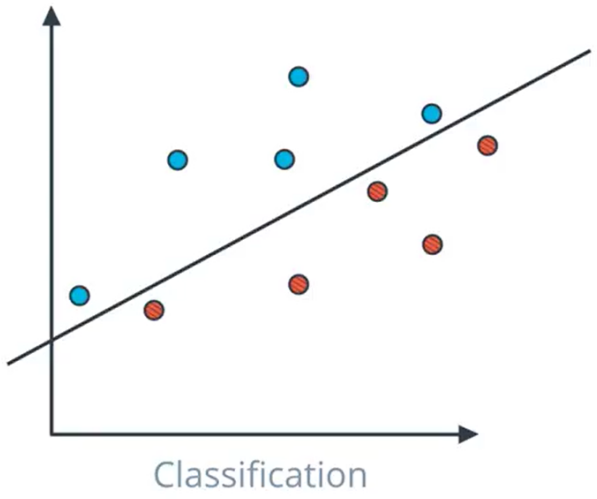
Will it be Cold or Hot tomorrow?



Supervised Learning: Linear Regression vs Logistic Regression

REGRESSION AND CLASSIFICATION





Regression returns a numeric value
We use Linear Regression

Classification returns a numeric state.
We use Logistic Regression

Supervised Learning: Linear Regression vs Logistic Regression

- Logistic Regression in scikit-learn

```
# import the class
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression()
```