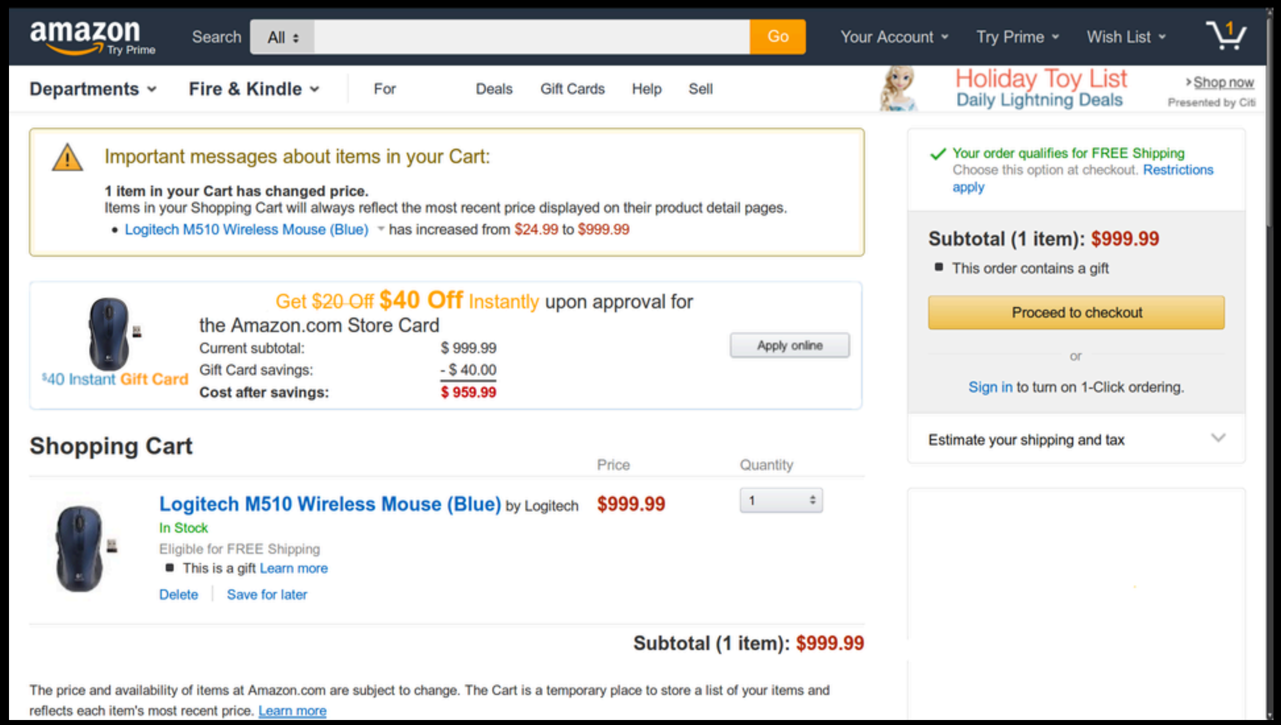
Foundations of Machine Learning

ALGIERS STEAM MAKERSPACE

Taxonomy of ML

Learning paradigms and algorithms

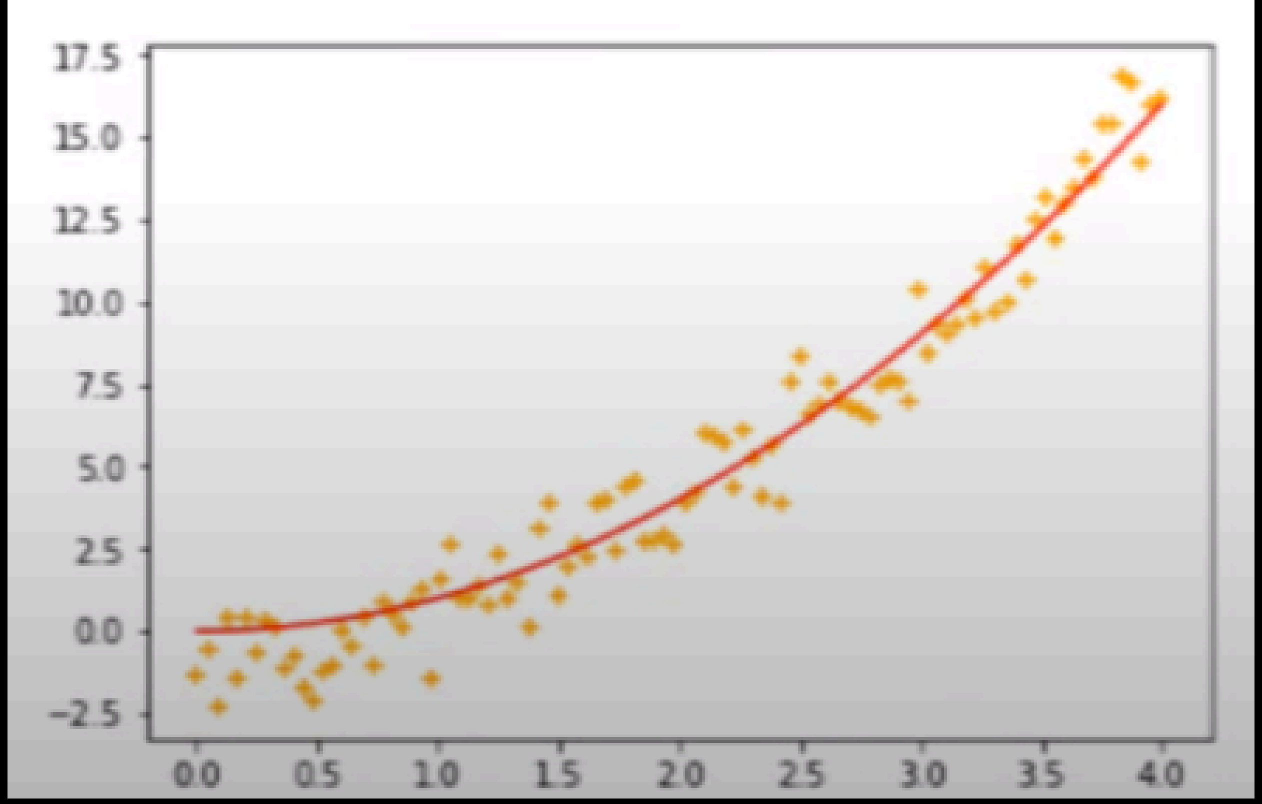
Machine Learning



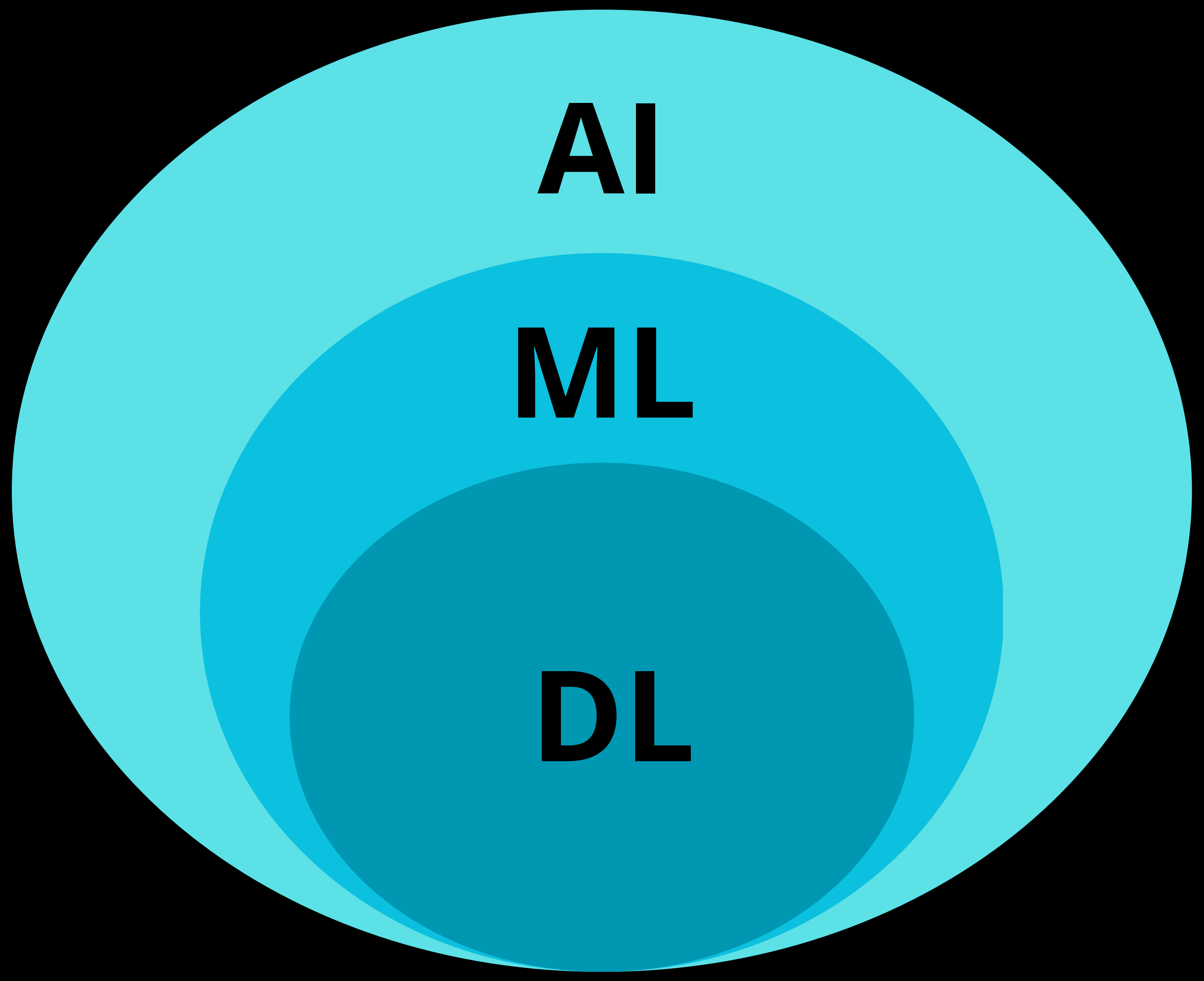
Machine Learning

"Giving a machine the ability to learn without being explicitly programmed."
- Arthur Samuel

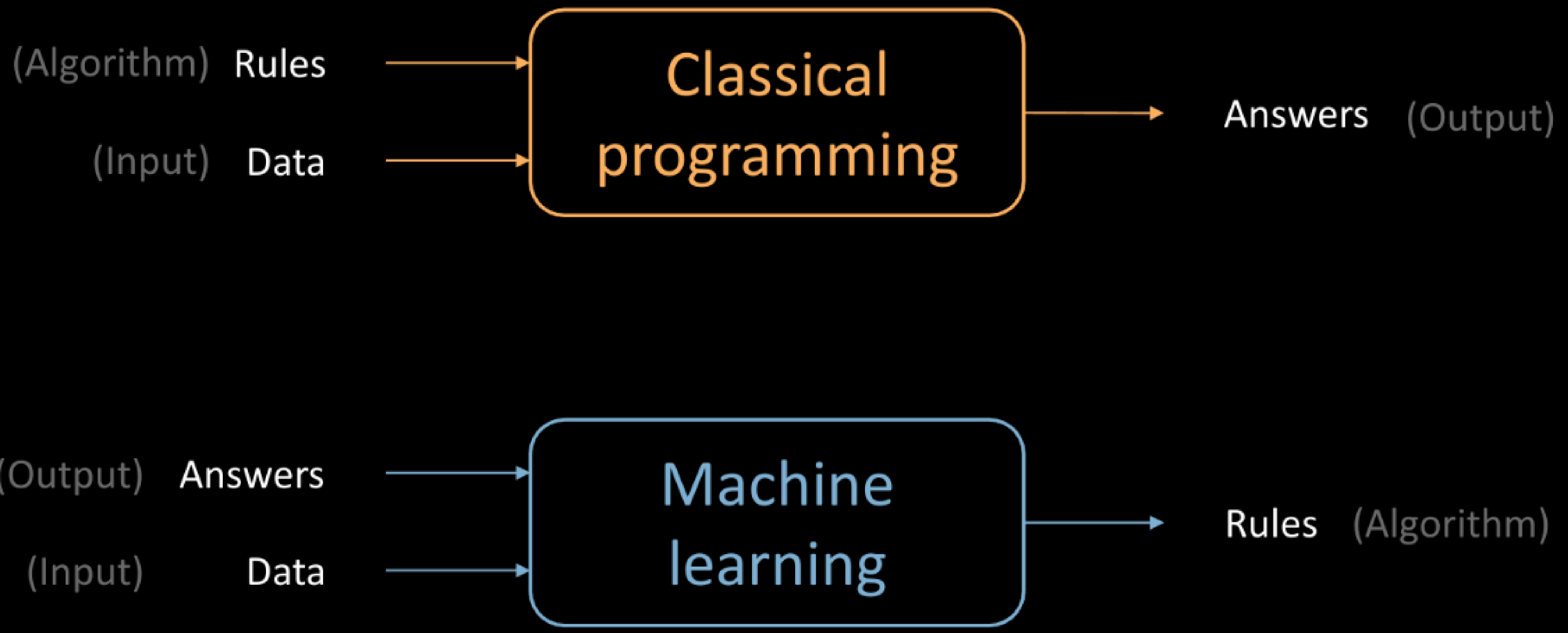
→ Developing a model from data



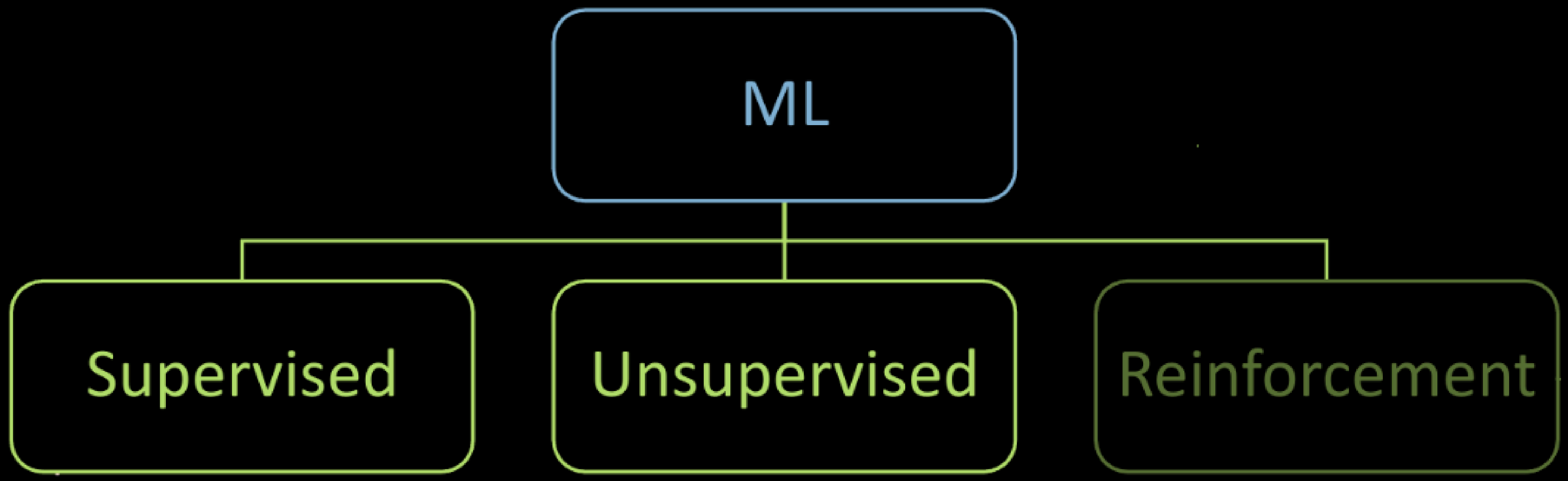
AI, ML, DL



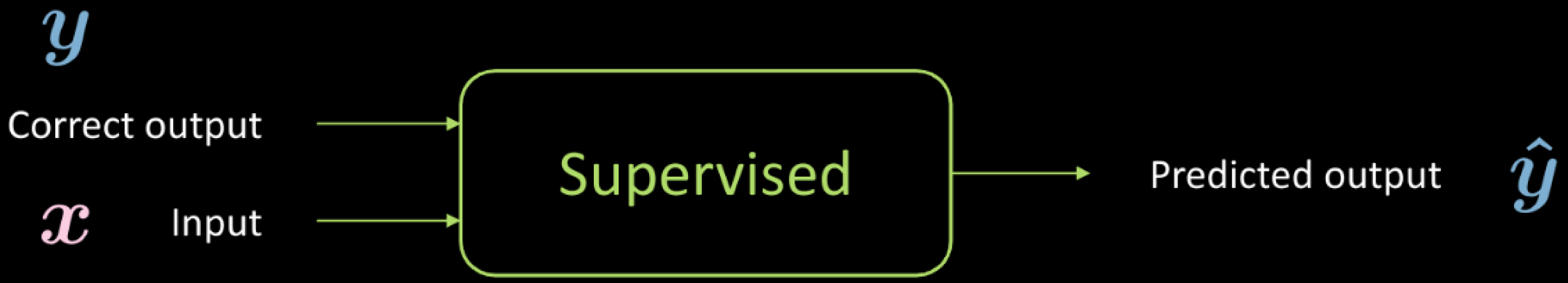
Machine Learning (ML)



ML paradigms taxonomy



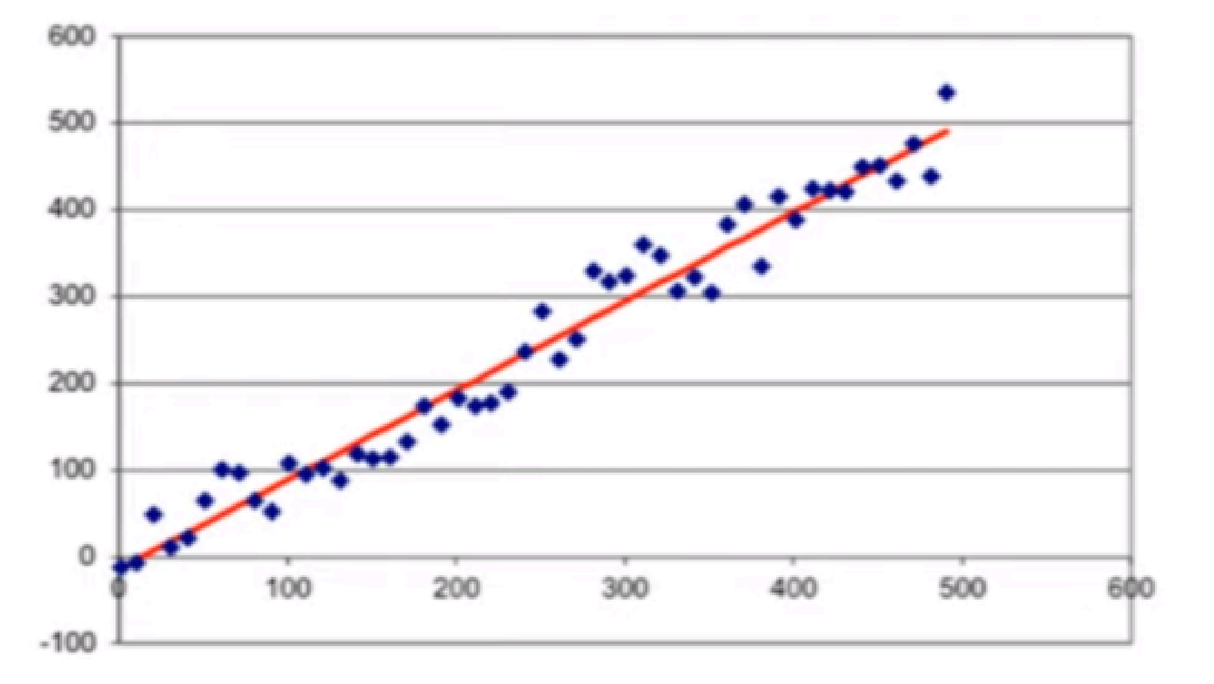
ML paradigms taxonomy



ML paradigms taxonomy : Supervised

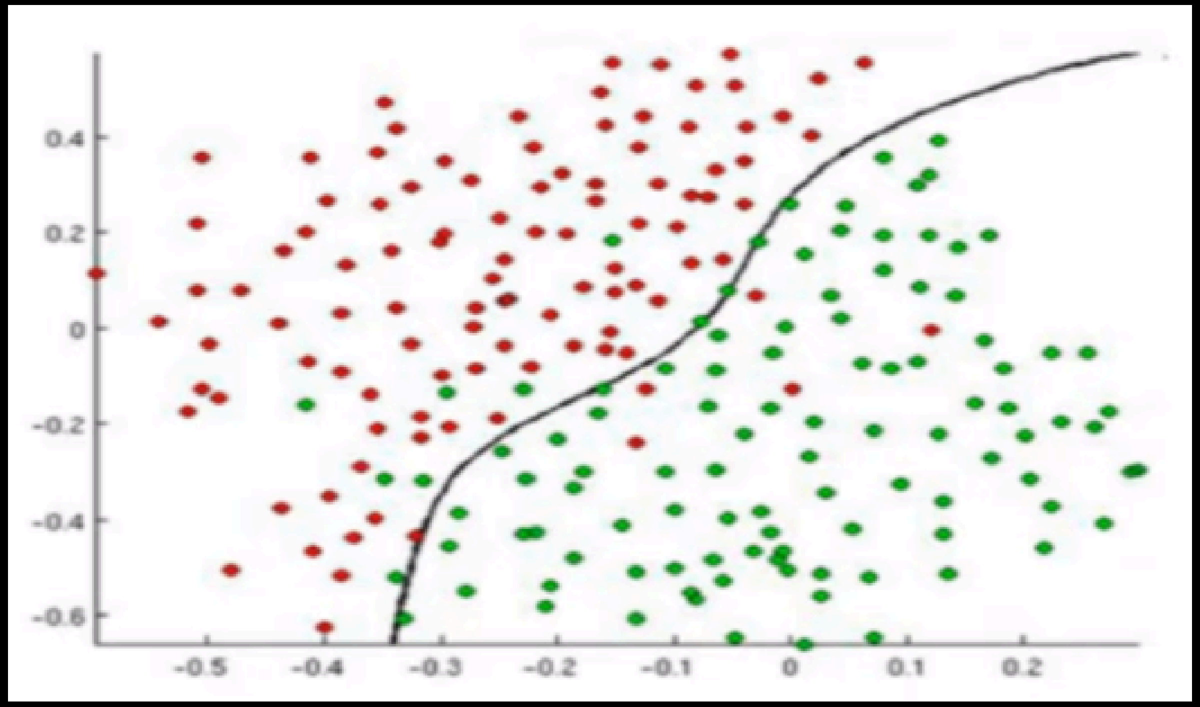
Allows solving two types of problems

Rregression



Price of an apartment
(y is continuous)

Classification



Email Spam / non Spam
(y is discrete)

ML paradigms taxonomy

Supervised Learning: The 4 Stages

1. Dataset

Example of a dataset on apartments.

Features

Target y x_1 x_2 x_3

Price	Area	Quality	Postal address
313,000	90	3	95000
720,000	110	5	93000
250,000	40	4	44500
290,000	60	3	67000
190,000	50	3	59300
...	...	...	...

m

n

$x_3^{(2)}$

By convention:

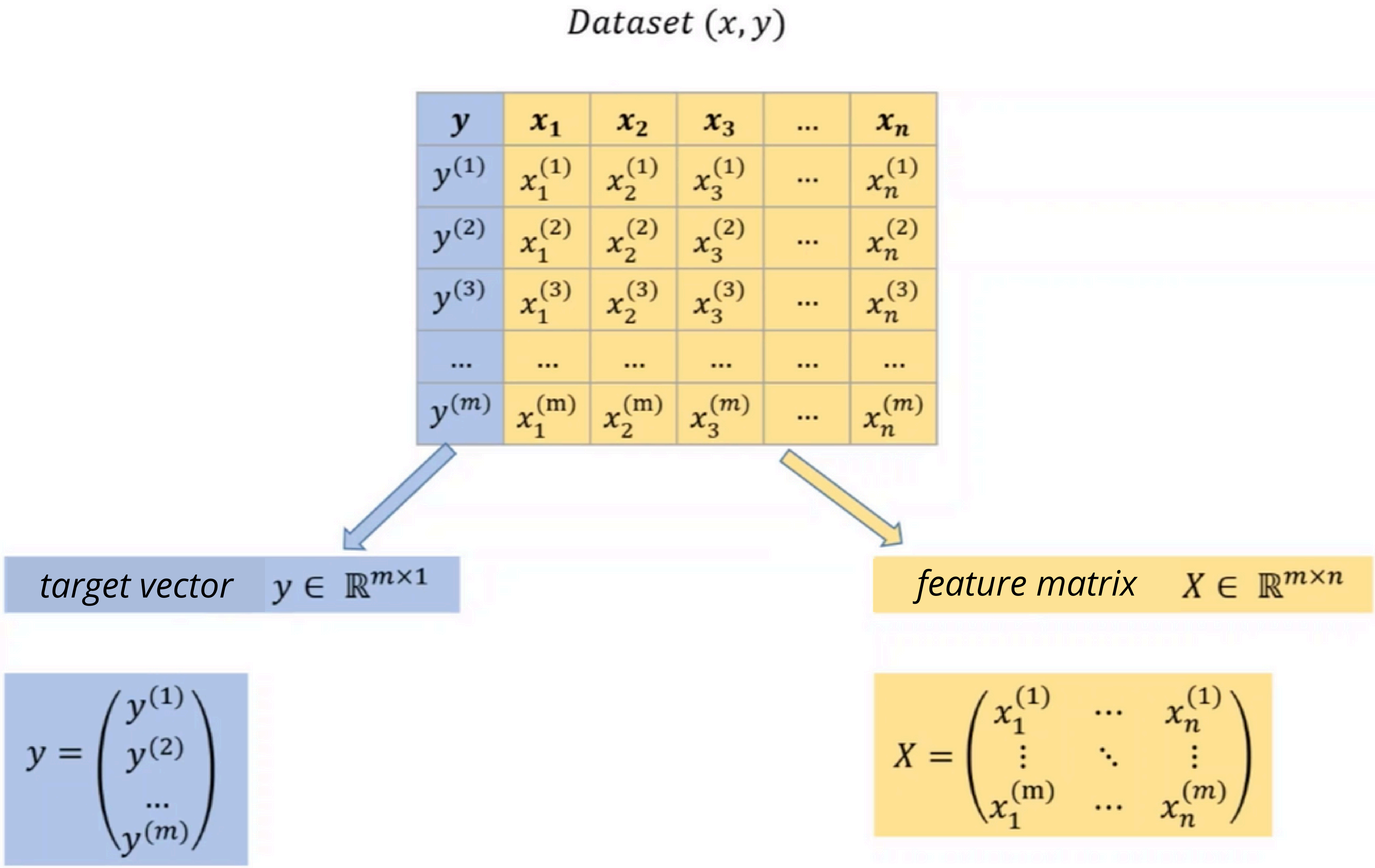
- ***m:*** represents the number of examples (or data points).
- ***n:*** represents the number of features (or attributes) per example.

by convention, we denote:

$$x^{(example)}_{feature}$$

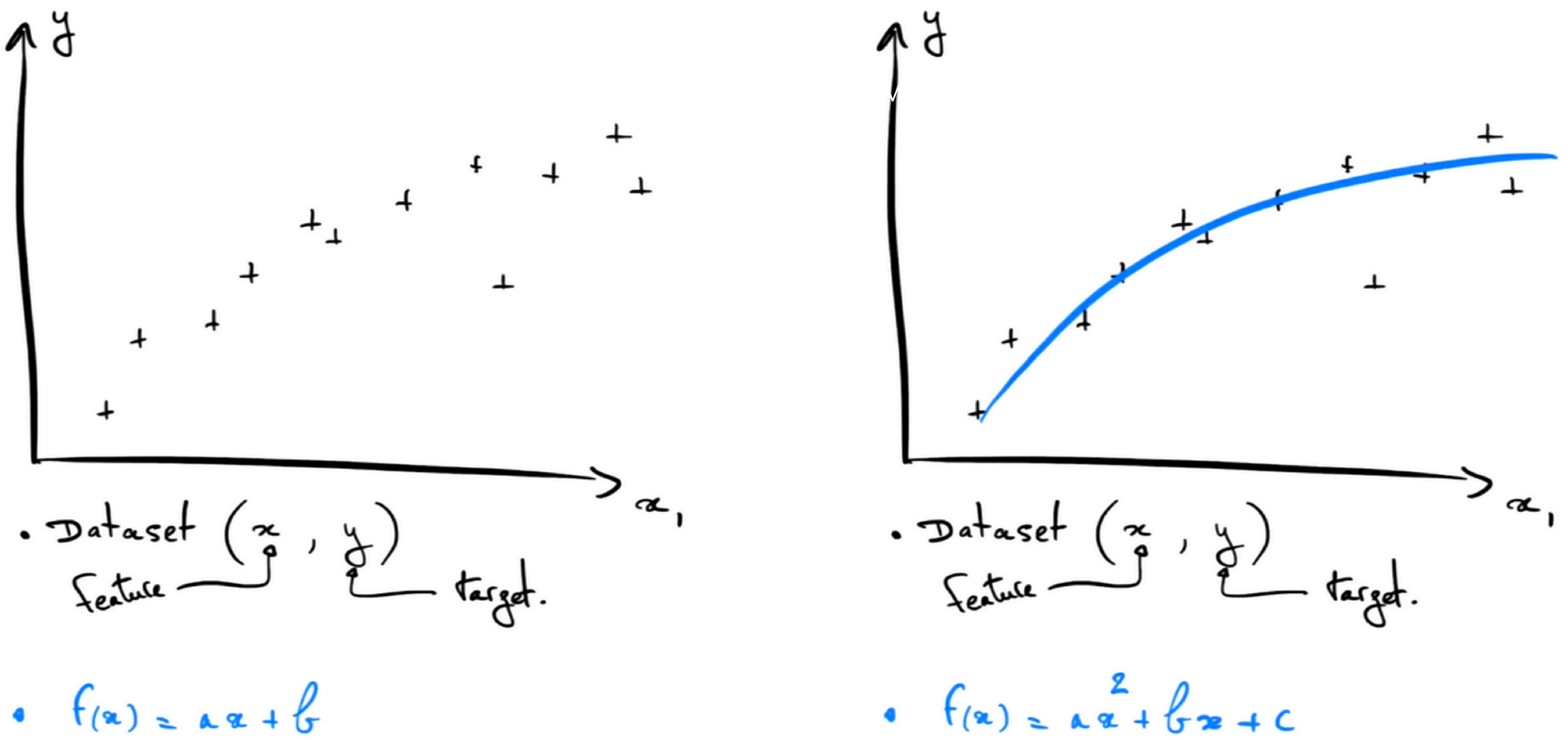
Supervised Learning: The 4 Stages

1. Dataset



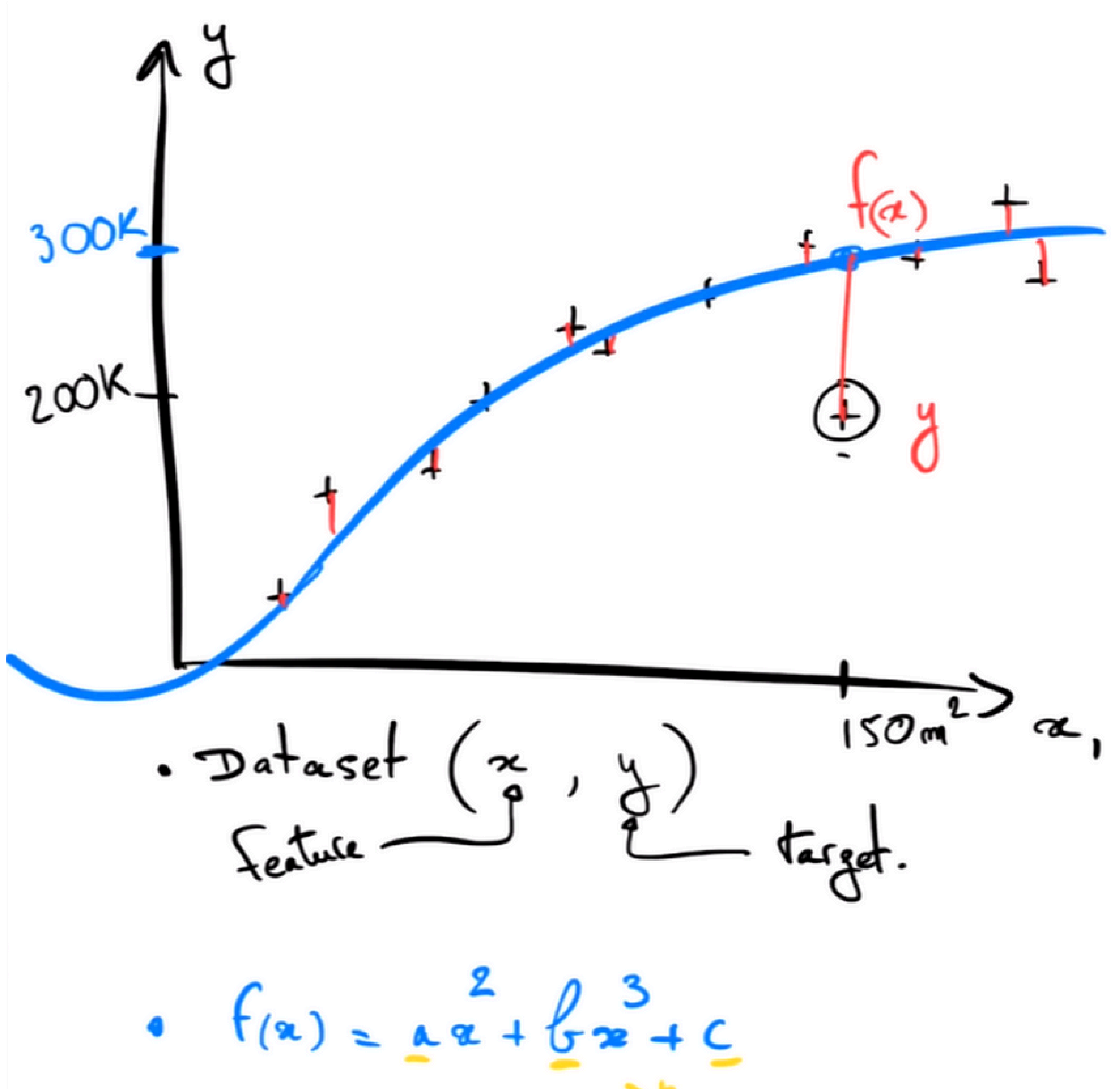
Supervised Learning: The 4 Stages

2. Model: Parameters



Supervised Learning: The 4 Stages

3. Cost Function



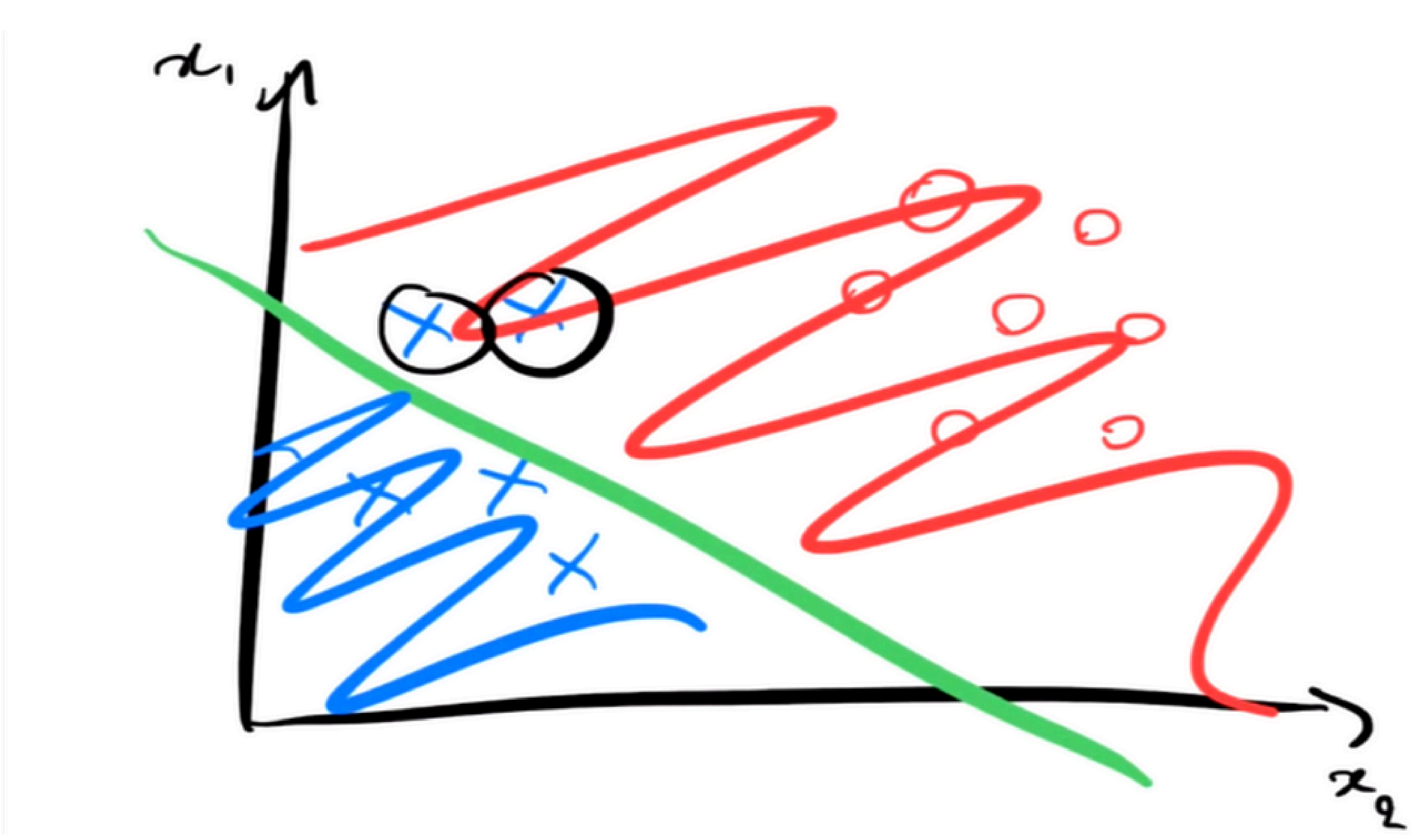
The cost function is the set of errors between the model's predictions and the true values of the dataset.

Supervised Learning: The 4 Stages

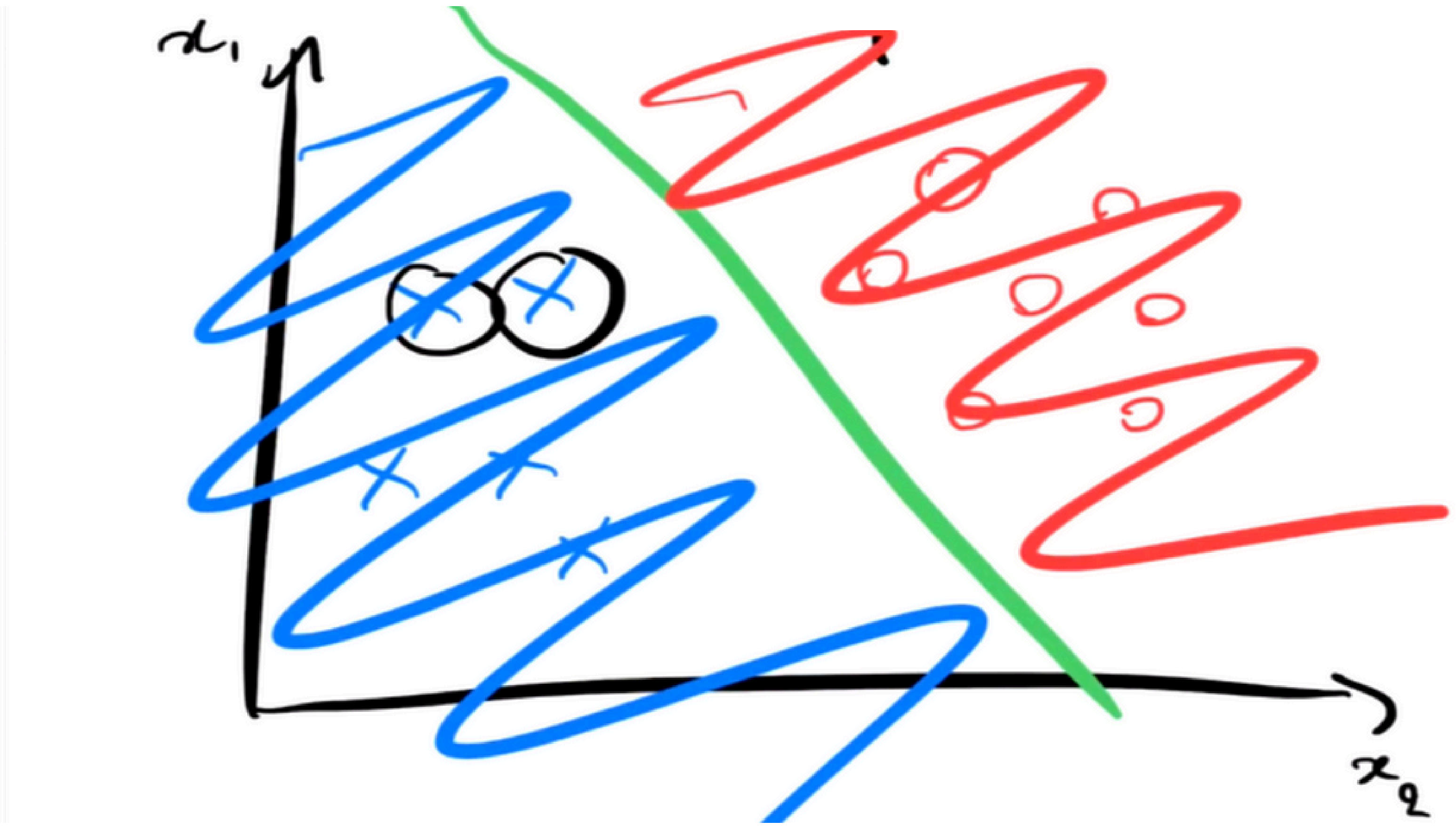
4. Optimization Algorithm

Aims to find the best model parameters to minimize the cost function.

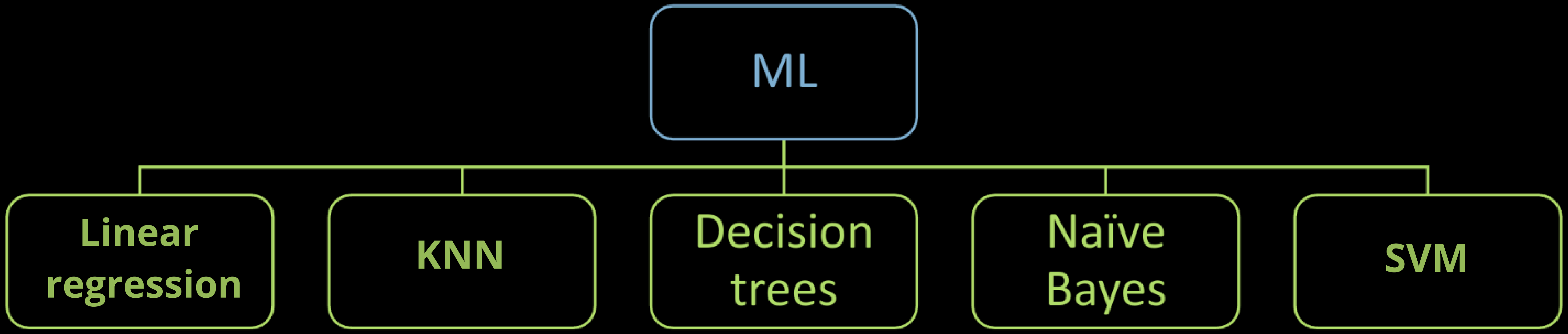
high cost



low cost



Supervised Learning: ML ALgorithms



Linear Regression

1. Dataset
2. Modal
3. Cost Function
4. Optimization Algorithm

Linear Regression

- **Dataset :** (x, y)
- **Modal :** $f(x) = ax + b$
- **Cost function :** $J(a, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(ax^{(i)} + b - y^{(i)}\right)^2$
- **Gradients :**

$$\frac{\partial J(a,b)}{\partial a} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}\left(ax^{(i)} + b - y^{(i)}\right)$$

$$\frac{\partial J(a,b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(ax^{(i)} + b - y^{(i)}\right)$$

- **Gradient Descent algorithm :**

$$a = a - \alpha\frac{\partial J(a,b)}{\partial a}$$

$$b = b - \alpha\frac{\partial J(a,b)}{\partial b}$$

Practice : Numpy

A library for numerical computing in Python.

Linear Regression with matrices

- **Dataset** : (x, y)
- **Modal** : $f(x) = ax + b$
- **Cost function** : $J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(ax^{(i)} + b - y^{(i)}\right)^2$
- **Gradients** :

$$\frac{\partial J(a,b)}{\partial a} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}\left(ax^{(i)} + b - y^{(i)}\right)$$

$$\frac{\partial J(a,b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(ax^{(i)} + b - y^{(i)}\right)$$

- **Gradient Descent algorithm** :

$$a = a - \alpha\frac{\partial J(a,b)}{\partial a}$$

$$b = b - \alpha\frac{\partial J(a,b)}{\partial b}$$

- Simplify calculations
- Develop more complex models without complicating the task

Linear Regression with matrices

The model F

$$X = \begin{bmatrix} x^{(1)} & 1 \\ \dots & \dots \\ x^{(m)} & 1 \end{bmatrix} \quad \theta = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\underset{m \times (n+1)}{X} \qquad \underset{(n+1) \times 1}{\theta}$$

$$\underset{m \times 1}{F} = X.\theta$$

Linear Regression with matrixes

Cost function

$$\underset{1 \times 1}{J(\theta)} = \frac{1}{2m}\sum(\underset{m \times 1}{X.\theta} - \underset{m \times 1}{Y})^2$$

Linear Regression with matrixes

Gradient

$$\underset{(n+1)\times 1}{\frac{\partial J(\theta)}{\partial \theta}} = \frac{1}{m} X^T.(X.\theta \ - Y)$$

Linear Regression with matrices

Gradient Descent

$$\theta \coloneqq \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

Practice : Numpy
Linear Regression

Sklearn

A machine learning library for Python.

ML with Sklearn

1. **Select an estimator and specify its parameters**

$$\underbrace{\textit{model}}_{\text{Object}} = \underbrace{\textit{LinearRegression}}_{\text{Constructor}}\underbrace{\textit{(......................)}}_{\text{Hyperparameters}}$$

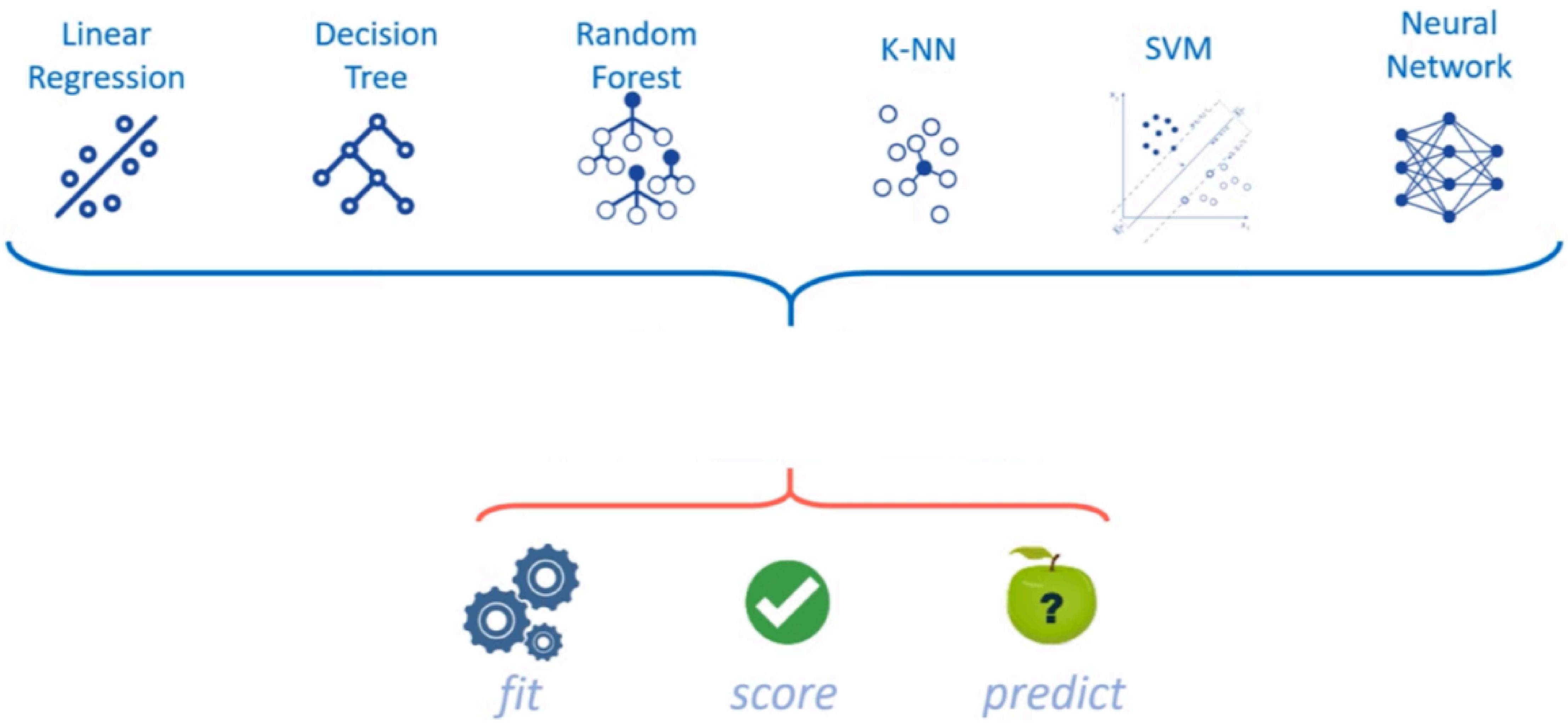
Examples

```python
model = SGDRegressor(eta0 = 0.3) # learning_rate = 0.3
```

```python
model = RandomForestClassifier(n_estimators=100)
```

Sklearn Interface

With 3 methods present in all classes!



fit(X, y): This method is used to train the model on the given data.

predict(X): This method is used to make predictions using the trained model.

score(X, y): This method is used to evaluate the performance of the model.

Sklearn in 4 Lines

Linear Regression

```
model = LinearRegression()
model.fit(X, y)
model.score(X, y)
model.predict(X)
```

Decision Tree

```
model = DecisionTreeClassifier()
model.fit(X, y)
model.score(X, y)
model.predict(X)
```

Sklearn Supervised Learning

1. **Select an estimator and specify its hyperparameters:**
   `model = LinearRegression(...)`

2. **Train the model on the data X, y (divided into 2 NumPy arrays):**
   `model.fit(x, y)`

3. **Evaluate the model:**
   `model.score(x, y)`

4. **Use the model:**
   `model.predict(X)`

Sklearn Modules

sklearn.**linear_model :**

Linear Regression
Ridge
SGDRegressor

sklearn.**neighbors :**

KNeighborsClassifier

sklearn.**svm :**

SVC
SVR

Sklearn Modules

```
from sklearn.linear_model import LinearRegression
```

```
import sklearn
```

```
model = LinearRegression()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most
<ipython-input-2-28d4d21f64c7> in <module>
----> 1 model = LinearRegression()

NameError: name 'LinearRegression' is not defined
```

Practice: Supervised Learning with Sklearn

Workflow of Supervised Learning algorithms

Step 1: Data Gathering

- Scikit-learn library
- Kaggle dataset

Workflow of Supervised Learning algorithms

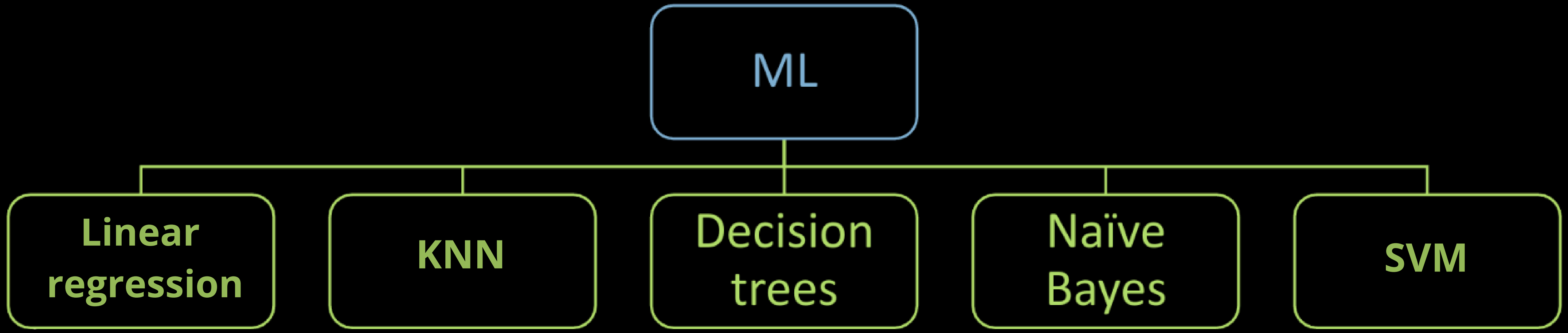
Step 2: Data pre-processing

- **Data Cleaning**: the process of filling in blanks, correcting errors, and deleting unnecessary data.
- **Data Transformation**: Techniques including aggregation, normalization, and feature selection.
- **Data Reduction**: the process of extracting relevant data from a large amount of data to meet a specific requirement. It entails attribute selection and dimension reduction.

Workflow of Supervised Learning algorithms

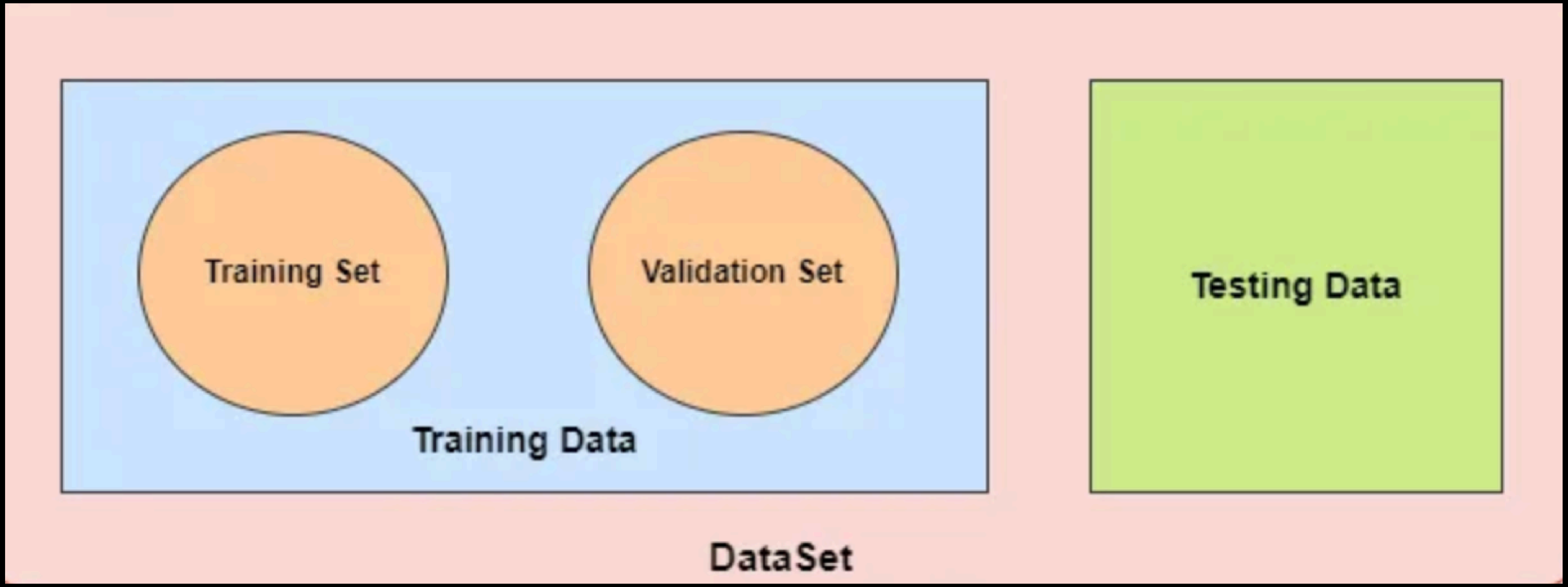
Step 3: Decide on a model.

Many models have been developed. Some are more suited to visuals, while others are better suited to numerical data, text, and so on. It's important to pick the right model.



Workflow of Supervised Learning algorithms

Step 4: Split the Dataset



Workflow of Supervised Learning algorithms

Step 5: Train the model

The model is trained by feeding datasets in this step.

Workflow of Supervised Learning algorithms

Step 6: Evaluation

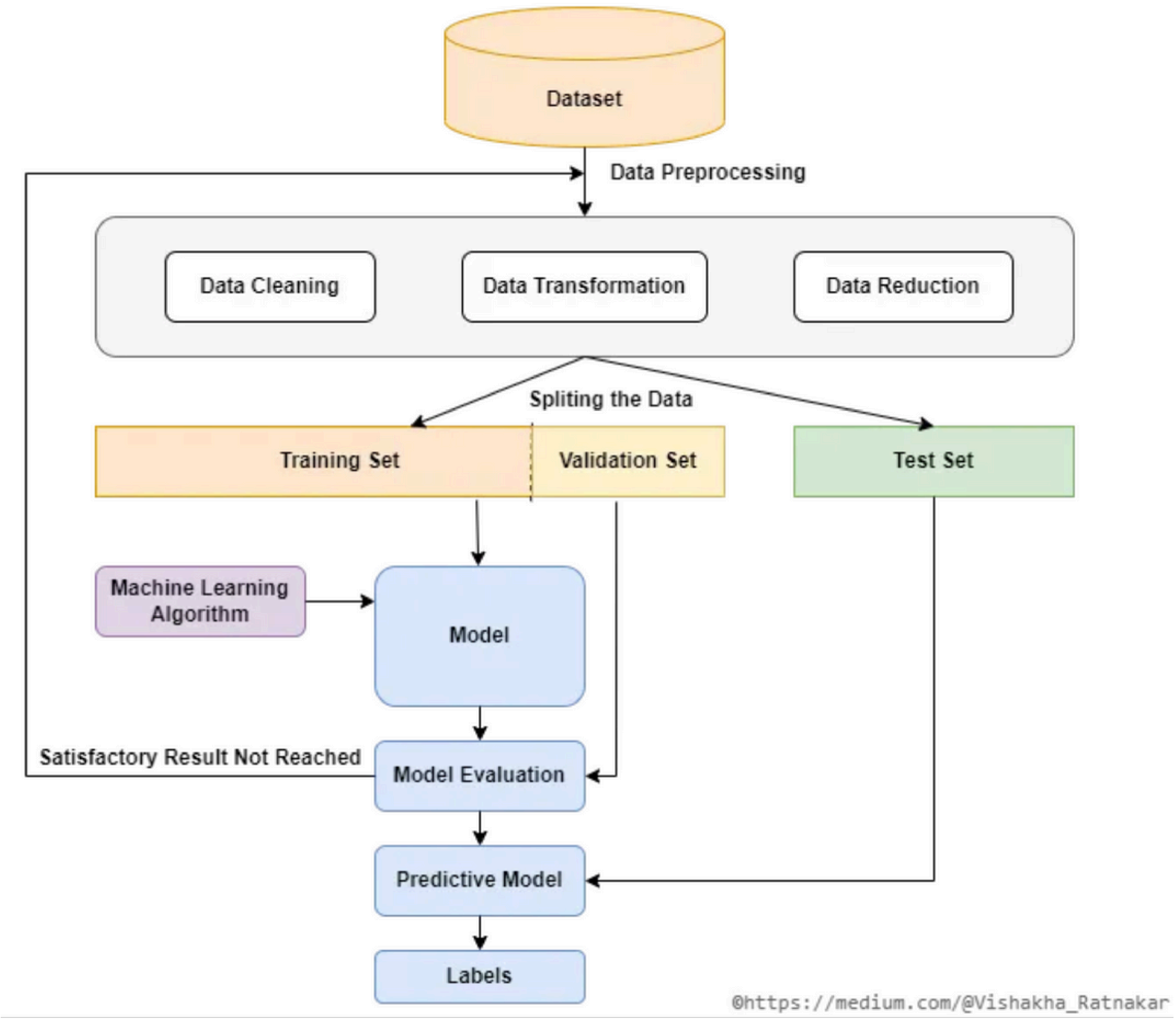
Basic performance measure for Classification task.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

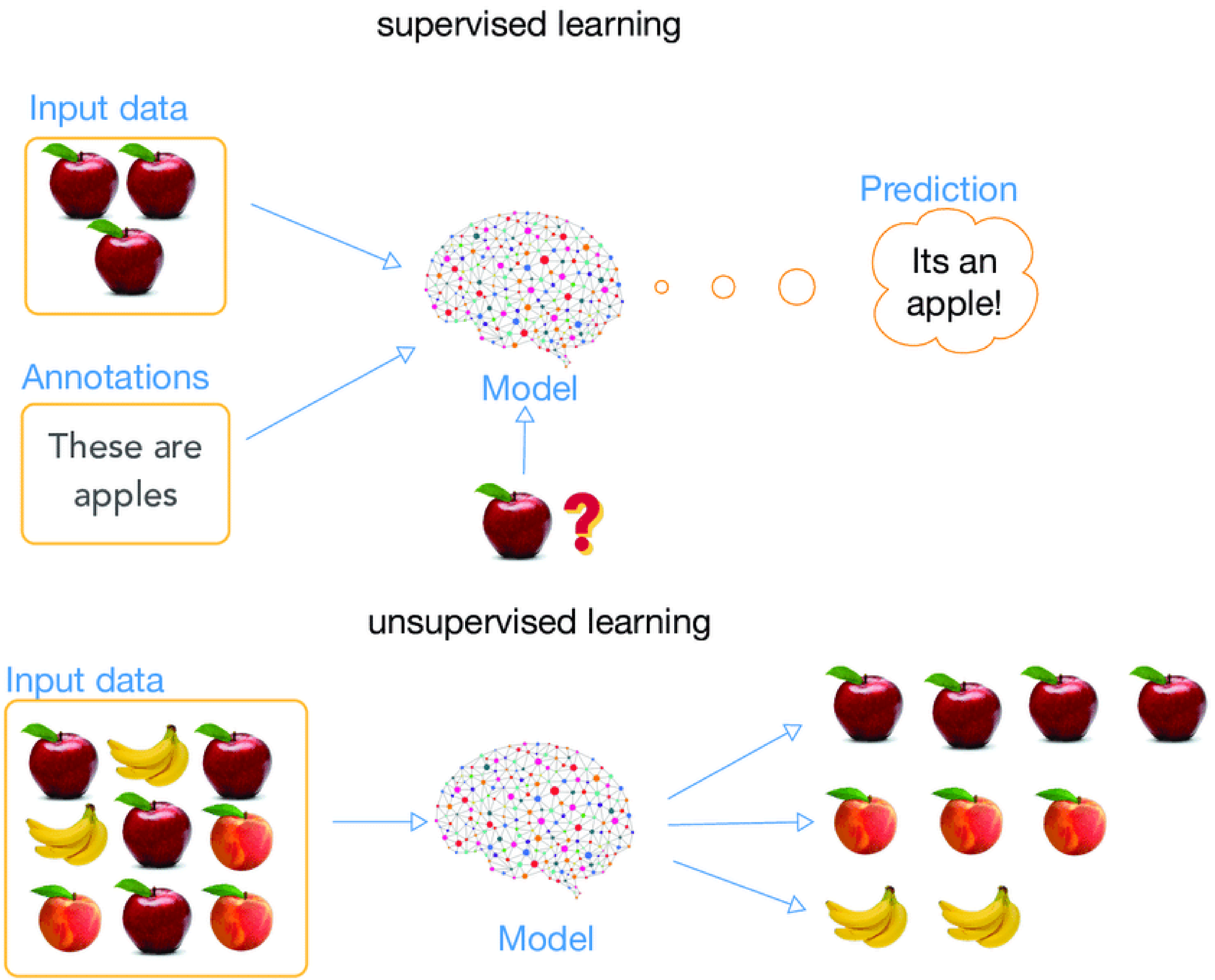
Basic performance measure for Regression task.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Workflow of Supervised Learning algorithms



Unsupervised Learning



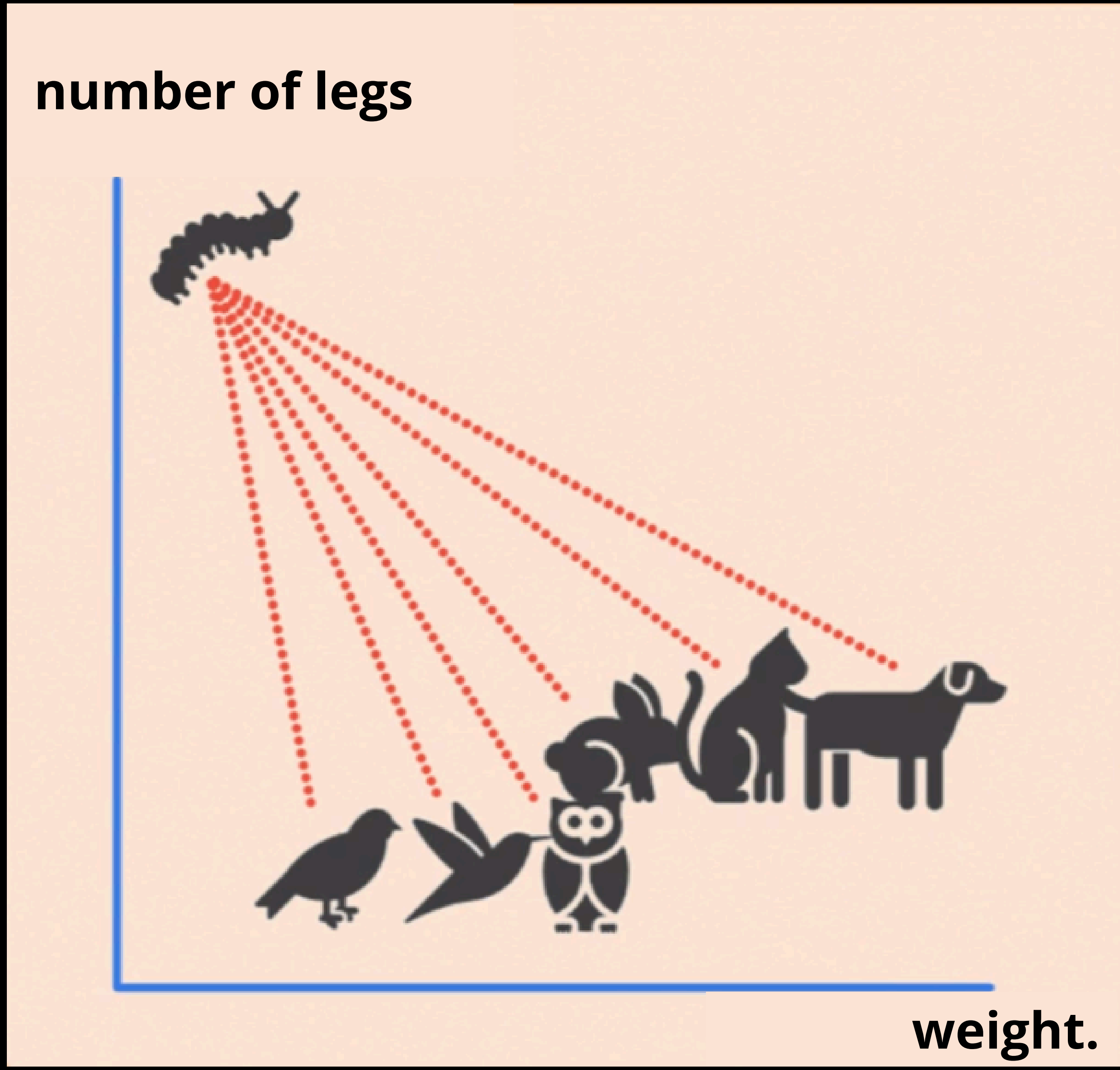
Unsupervised Learning : Applications

Clustering

Unsupervised Learning : Applications

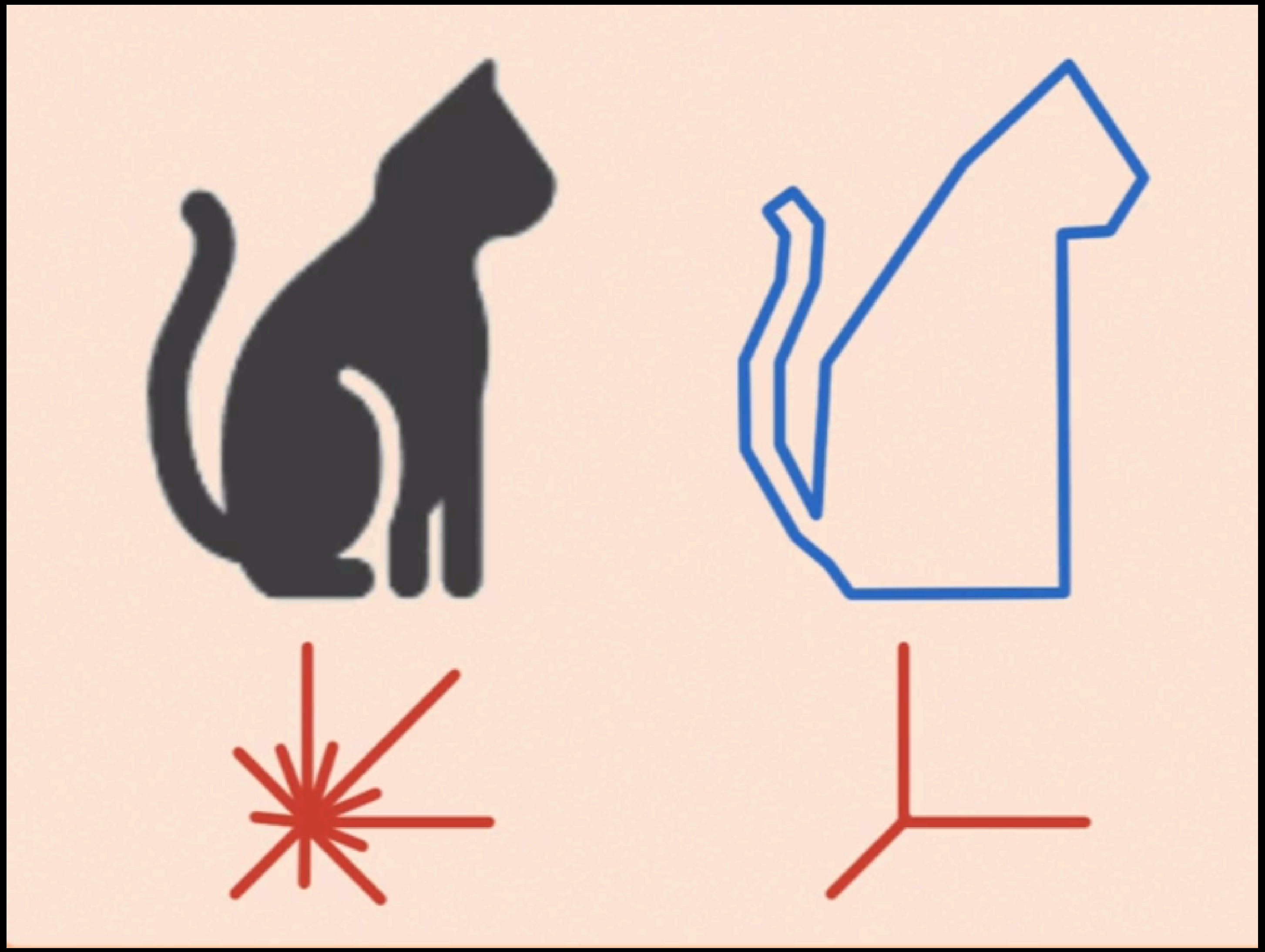
Anomaly detection

identifying data points, events, or observations that deviate significantly from the majority of the data. These anomalies or outliers may indicate critical incidents, such as fraud, network intrusion, structural defects, or medical conditions.



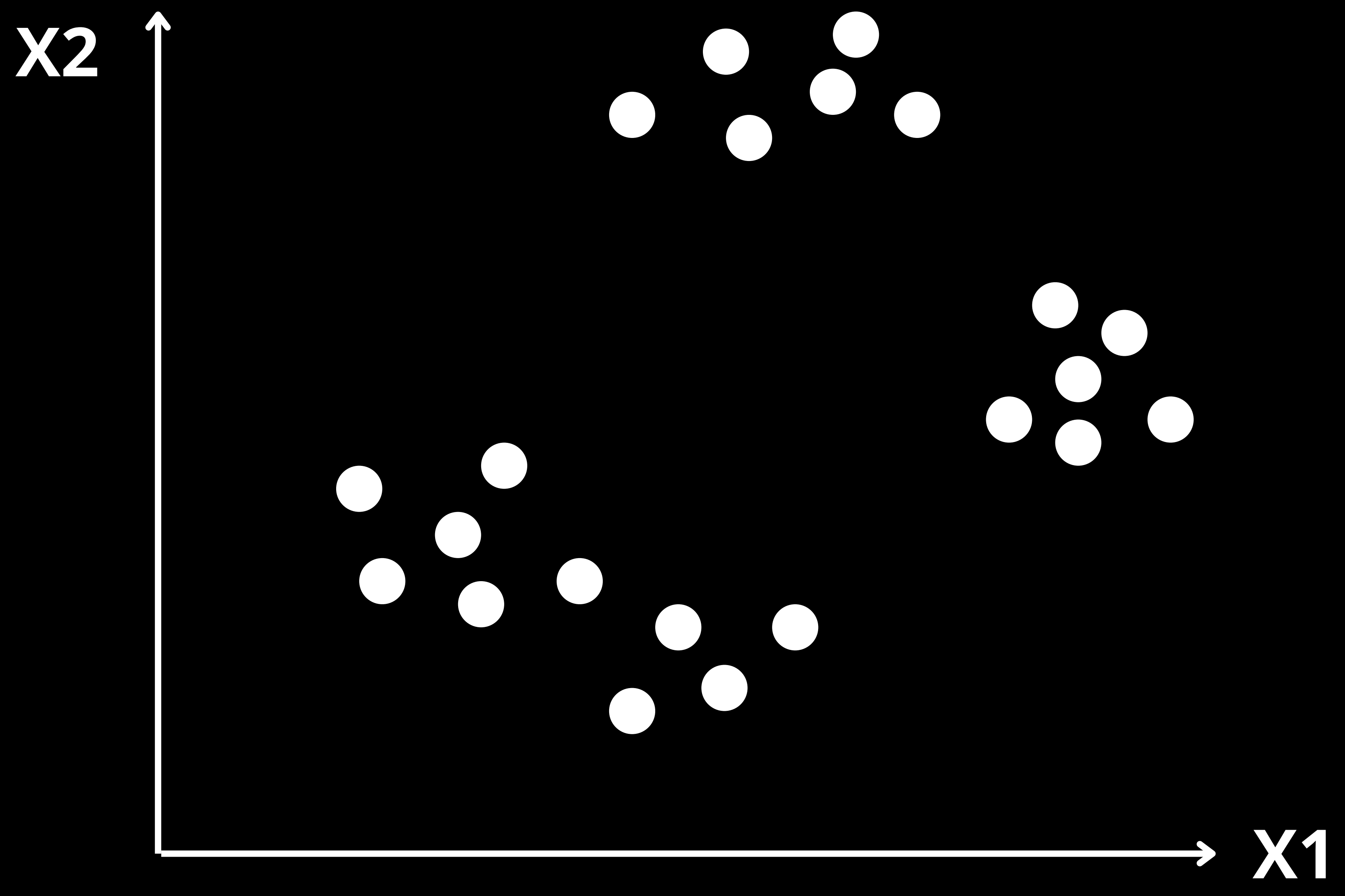
Unsupervised Learning : Applications

Dimentionality reduction: PCA

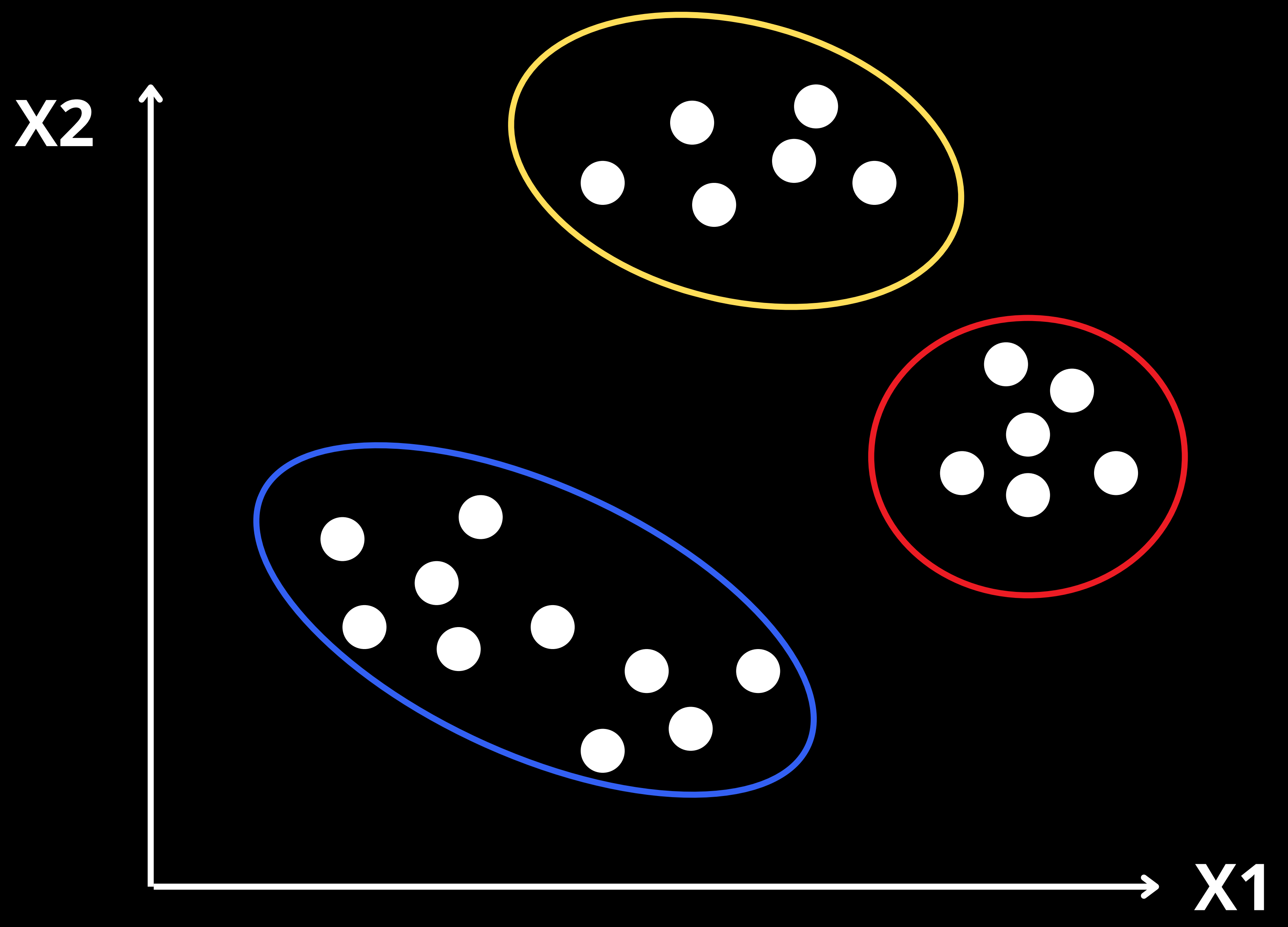


used to reduce the number of input variables or features in a dataset while retaining as much information as possible. This process simplifies the dataset, making it easier to visualize, understand, and model.

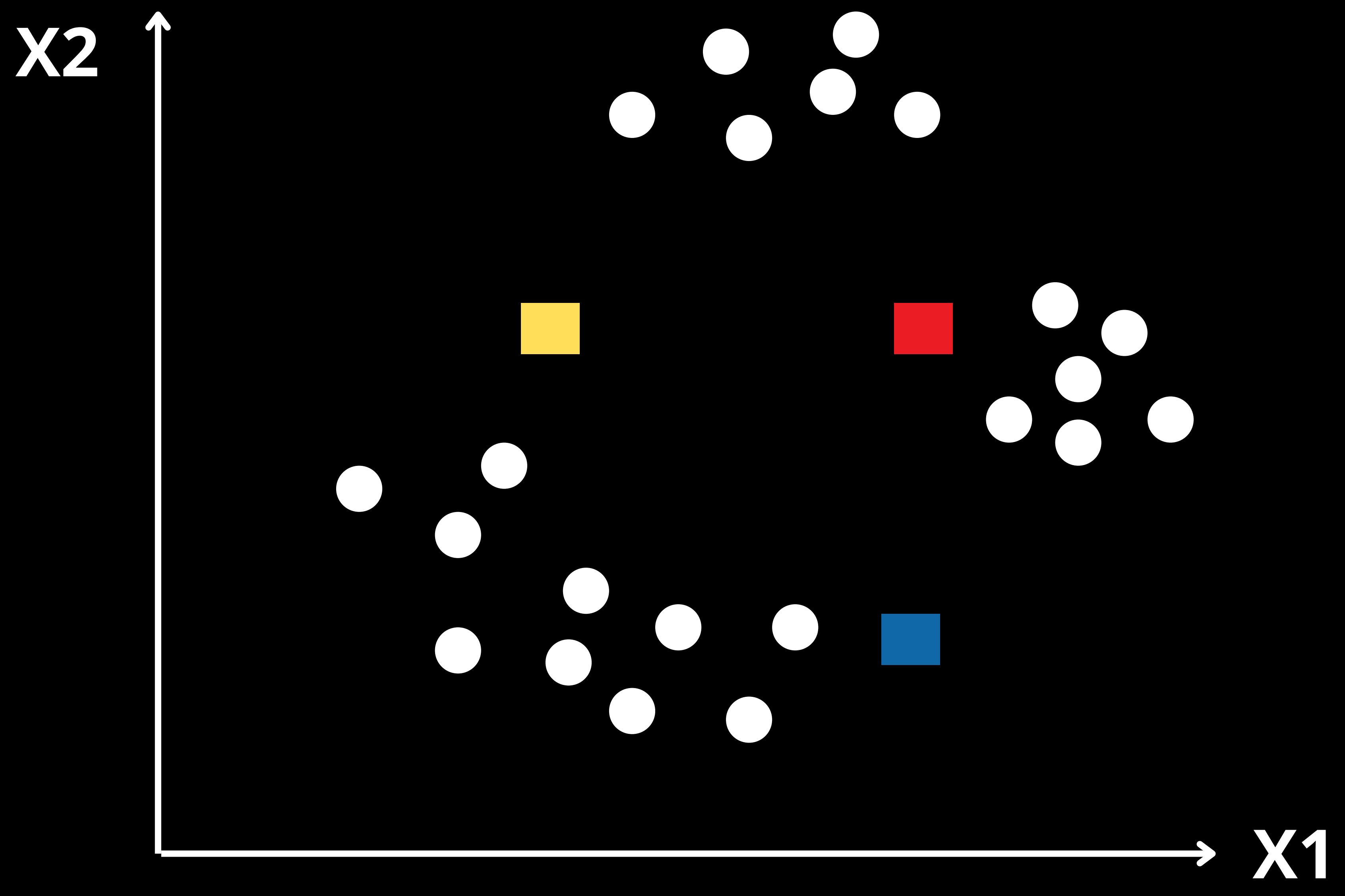
K-Means Clustering



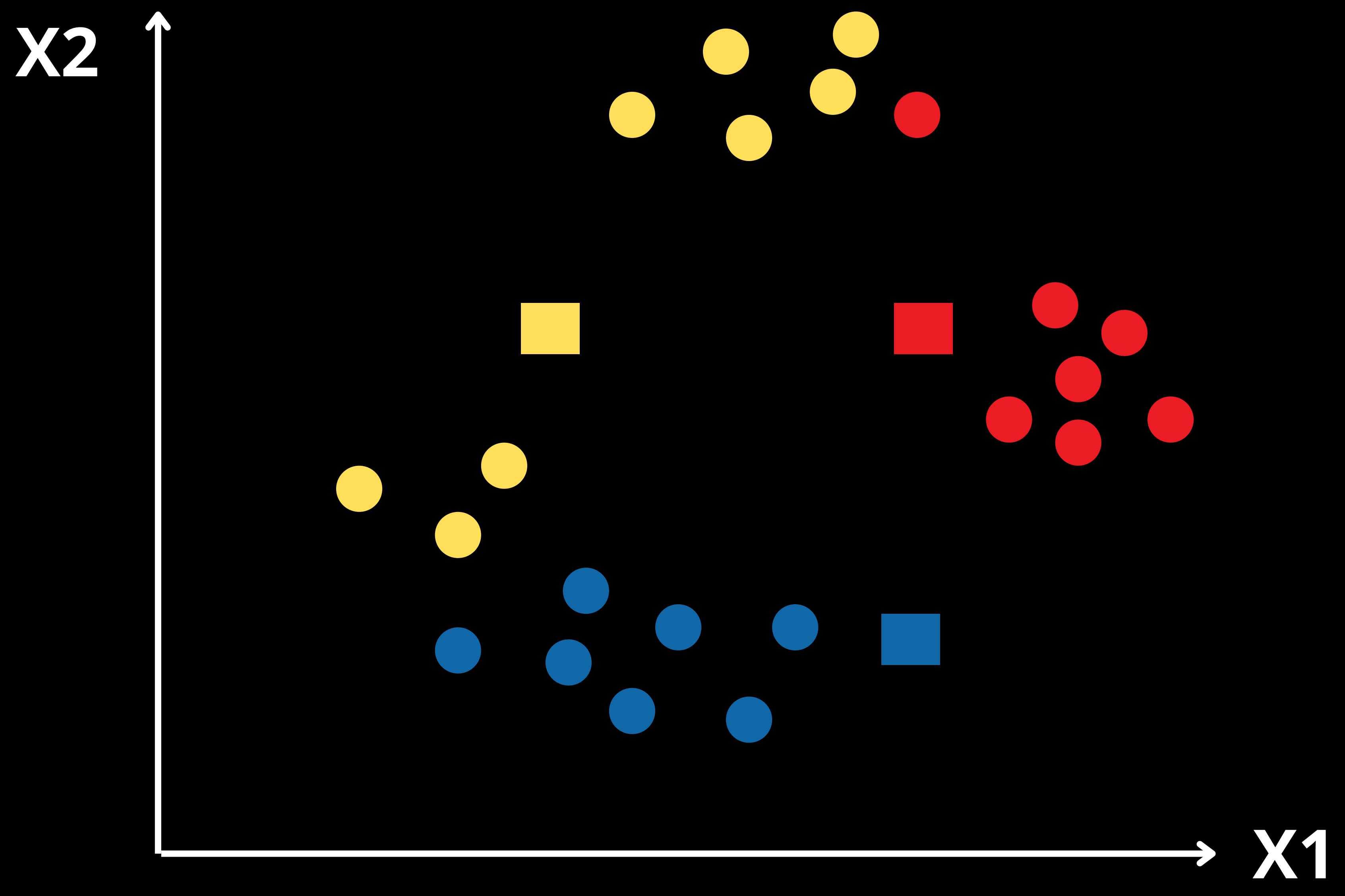
K-Means Clustering



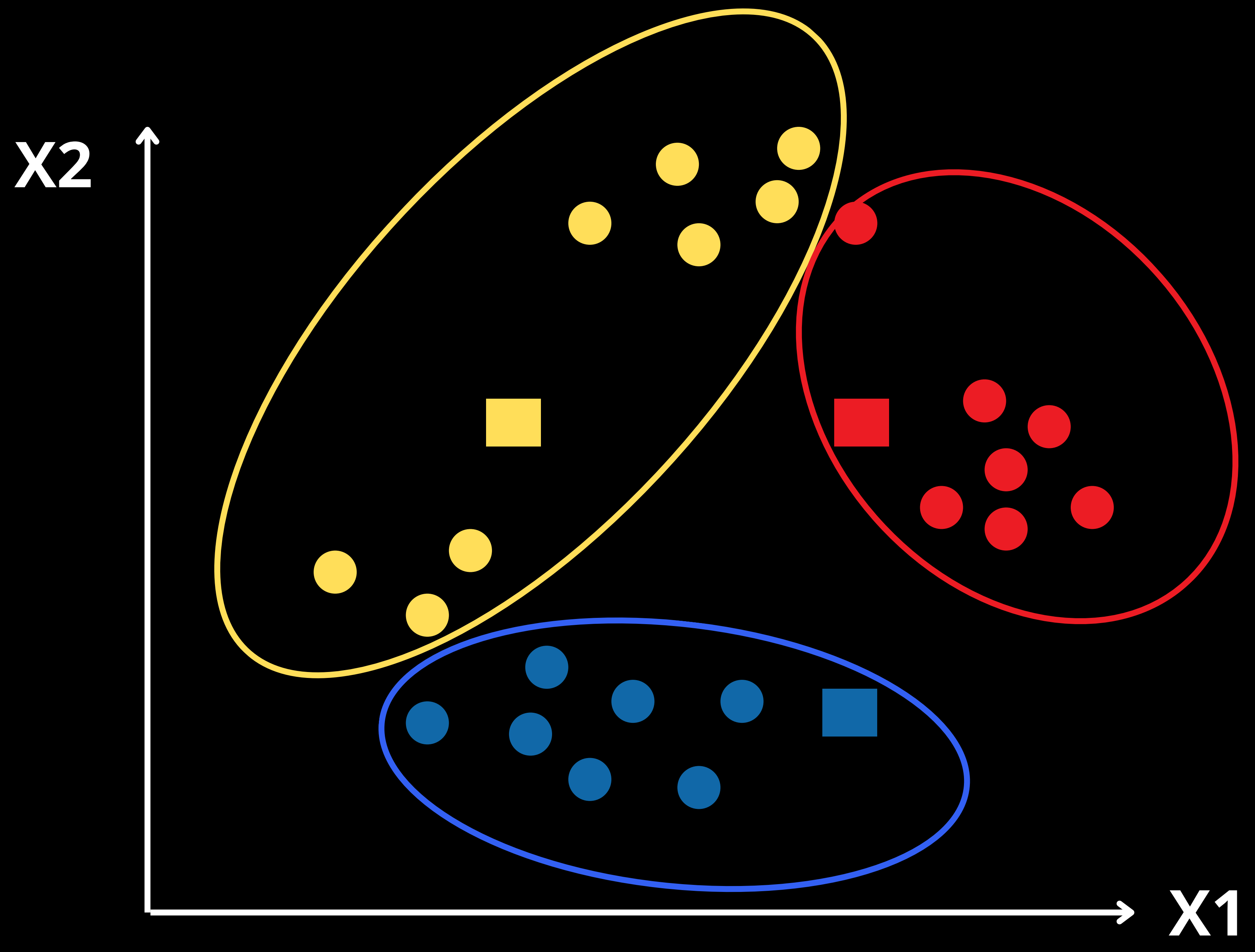
K-means Clustering



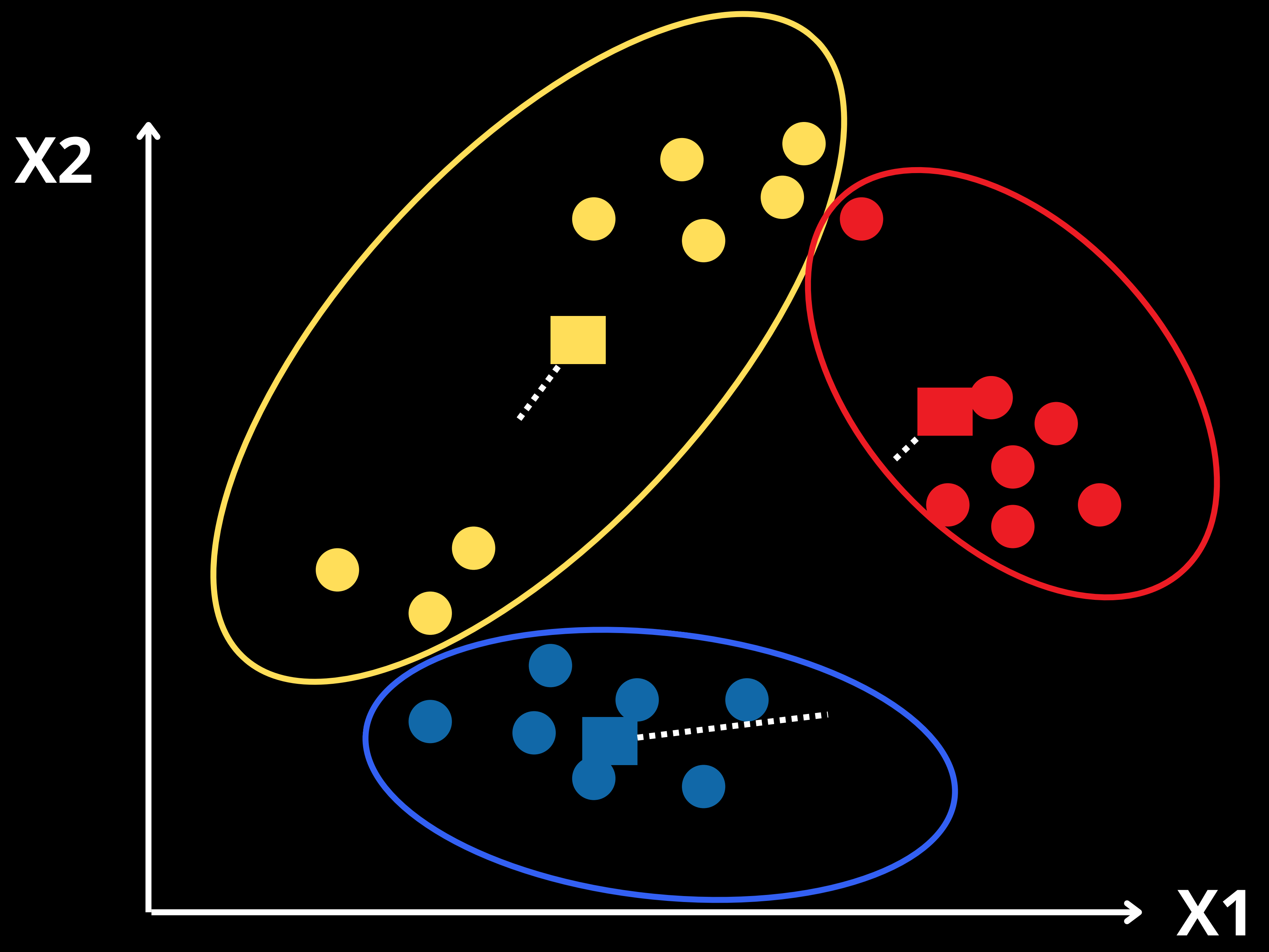
K-means Clustering



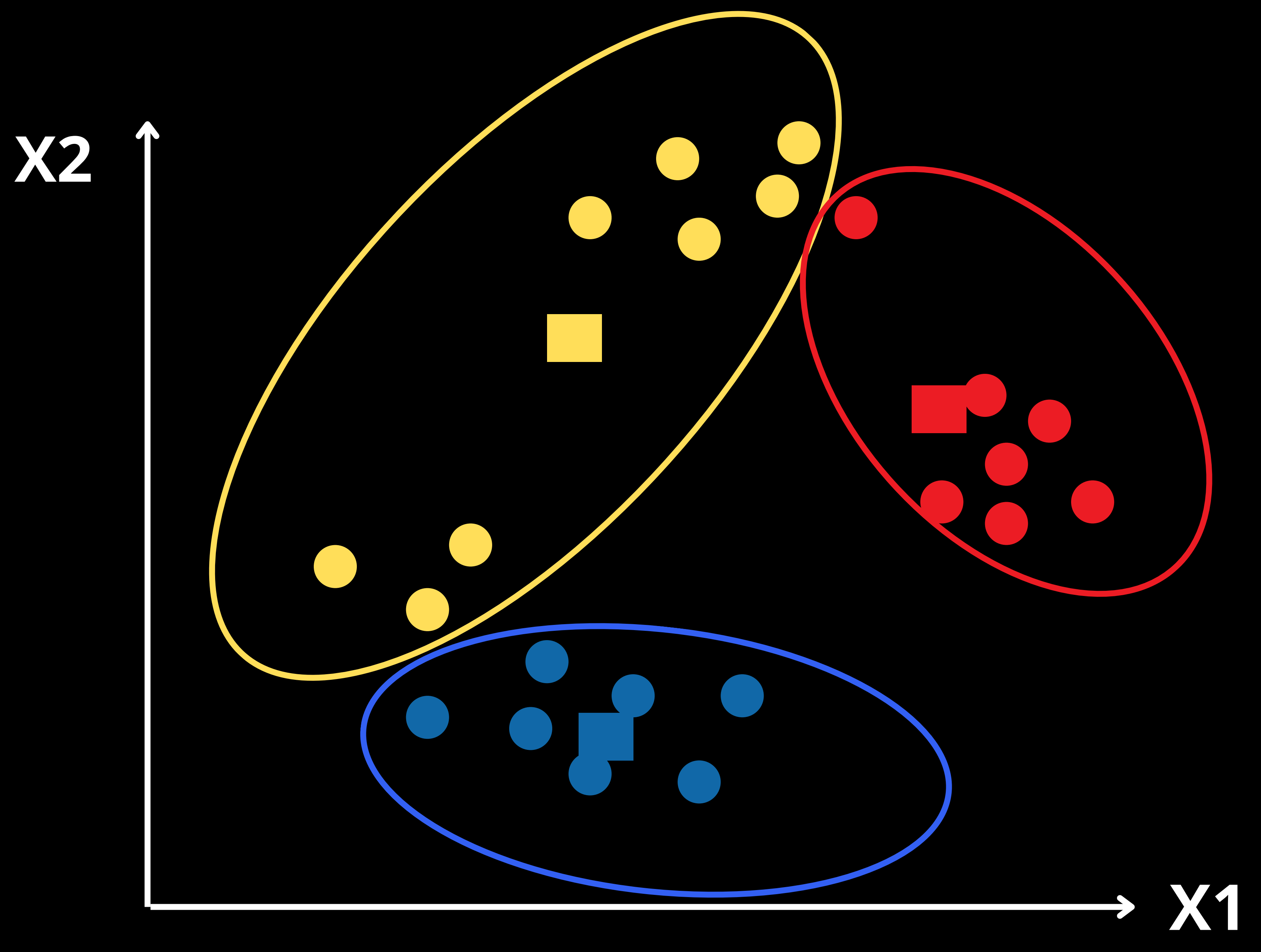
K-means Clustering



K-means Clustering

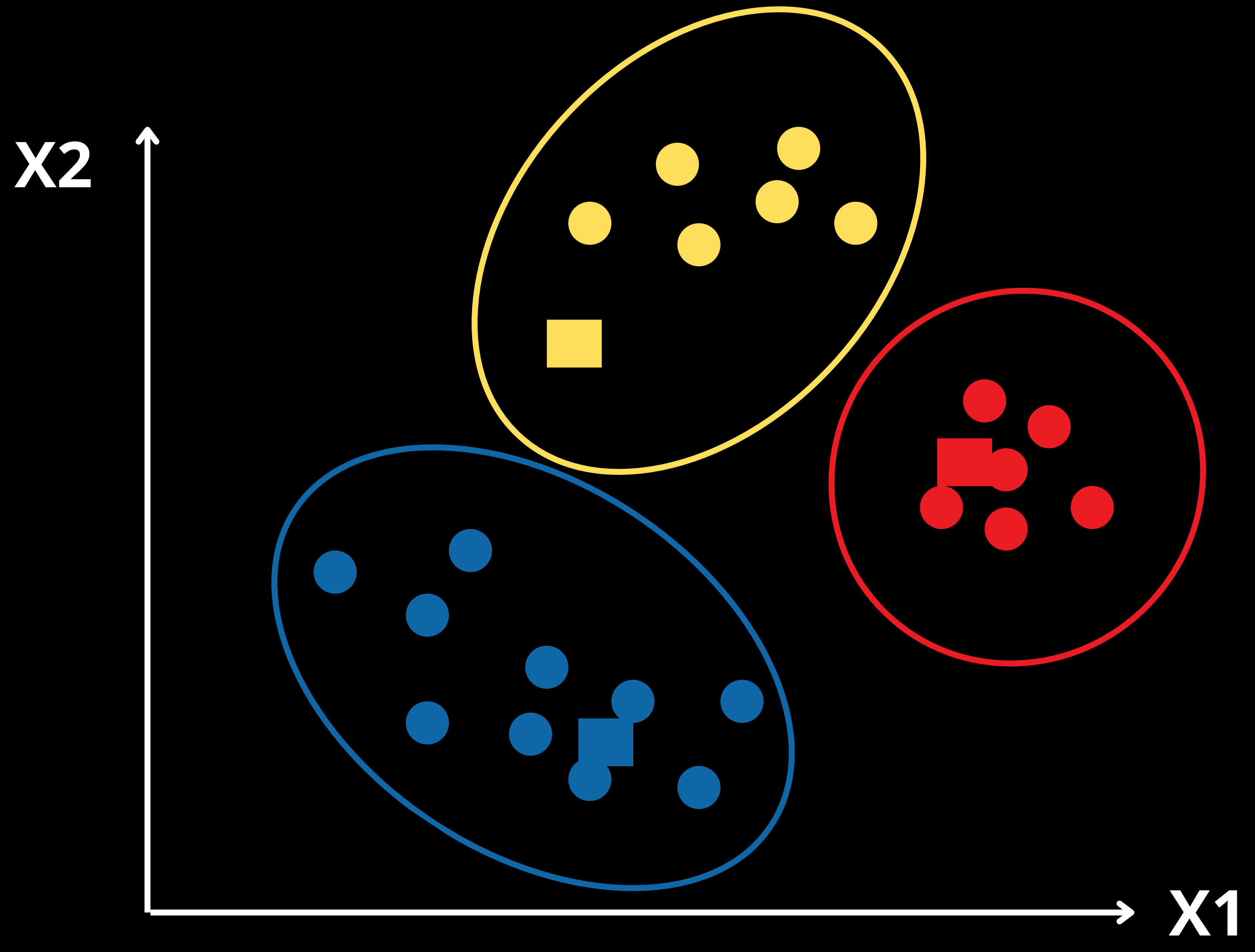


K-means Clustering

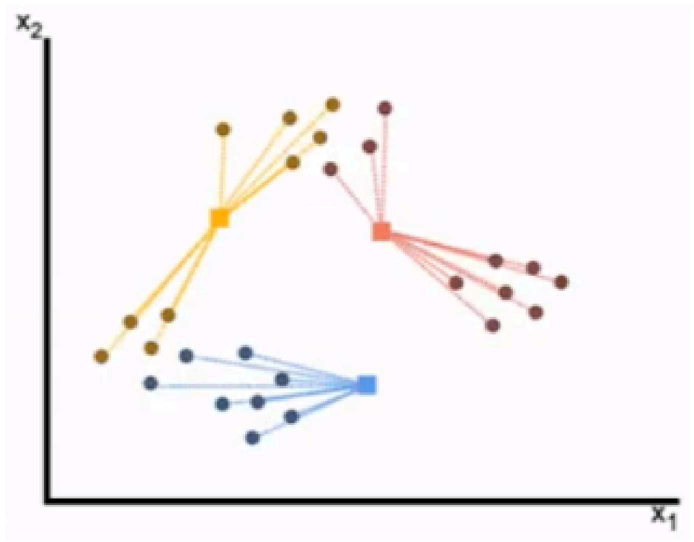


K-means Clustering

K-means Clustering

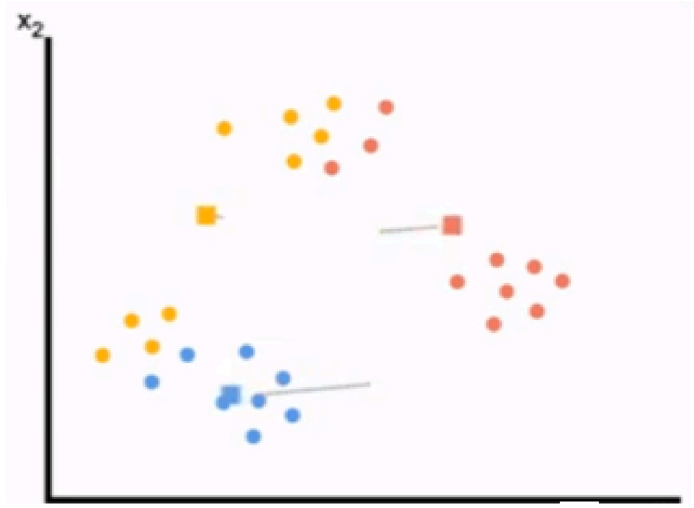


K-means Clustering



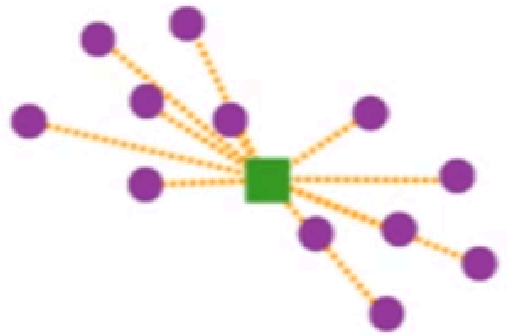
K-Means is an iterative algorithm that operates in two steps:

1. **Assignment:** Each point is assigned to the nearest cluster center.
2. **Update:** The cluster center is moved to the mean of the assigned points.



K-means Clustering

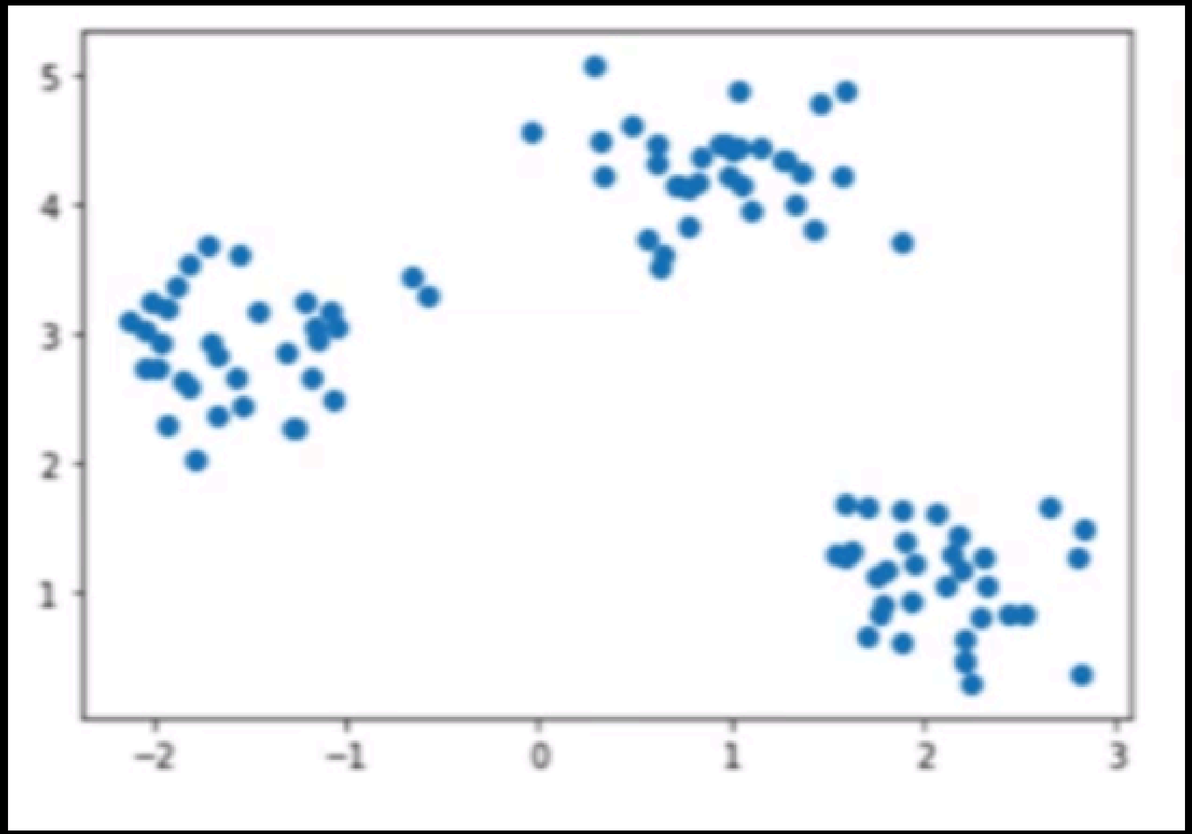
In summary: K-Means seeks the positions of the centers that minimize the distance between the points of a cluster and the center of that cluster.

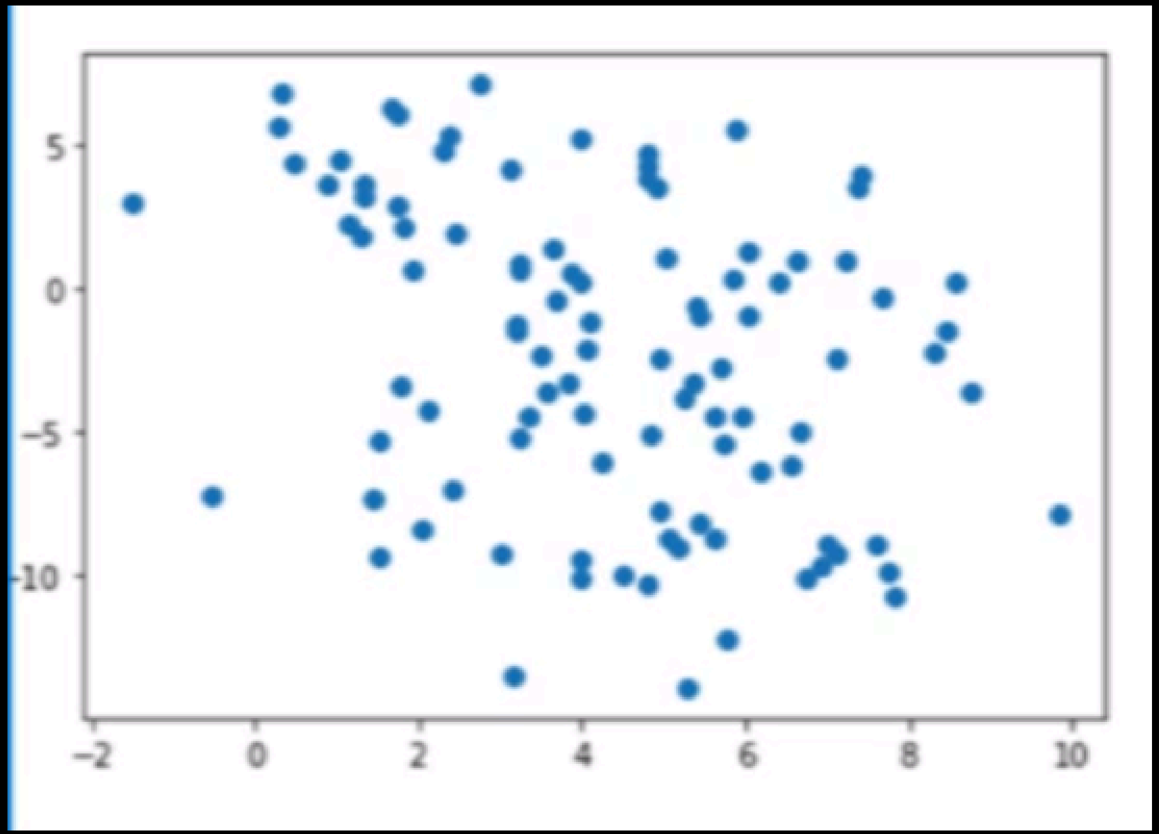


$$\boxed{\sum_{i=0}^{n} \underset{\mu_j \in C}{min} \left(\left\| x_i - \mu_j \right\|^2\right)}$$

K-means Clustering: Elbow method

How to know the number of clusters for the algorithm?





K-means Clustering: Elbow method

Plotting the inertia against the number of clusters and looking for the "elbow" point where adding more clusters results in only a marginal reduction in inertia.