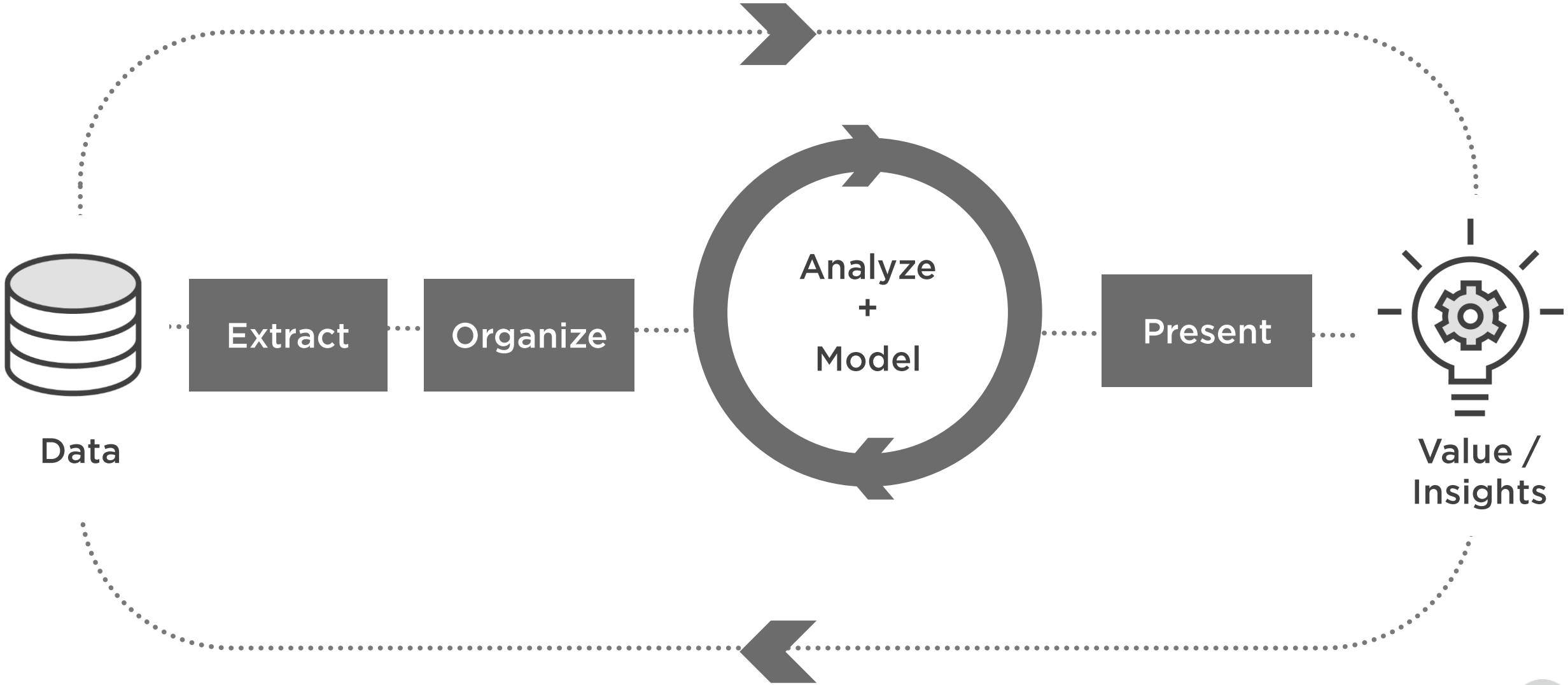
# Building and Evaluating Predictive Models – Part 1

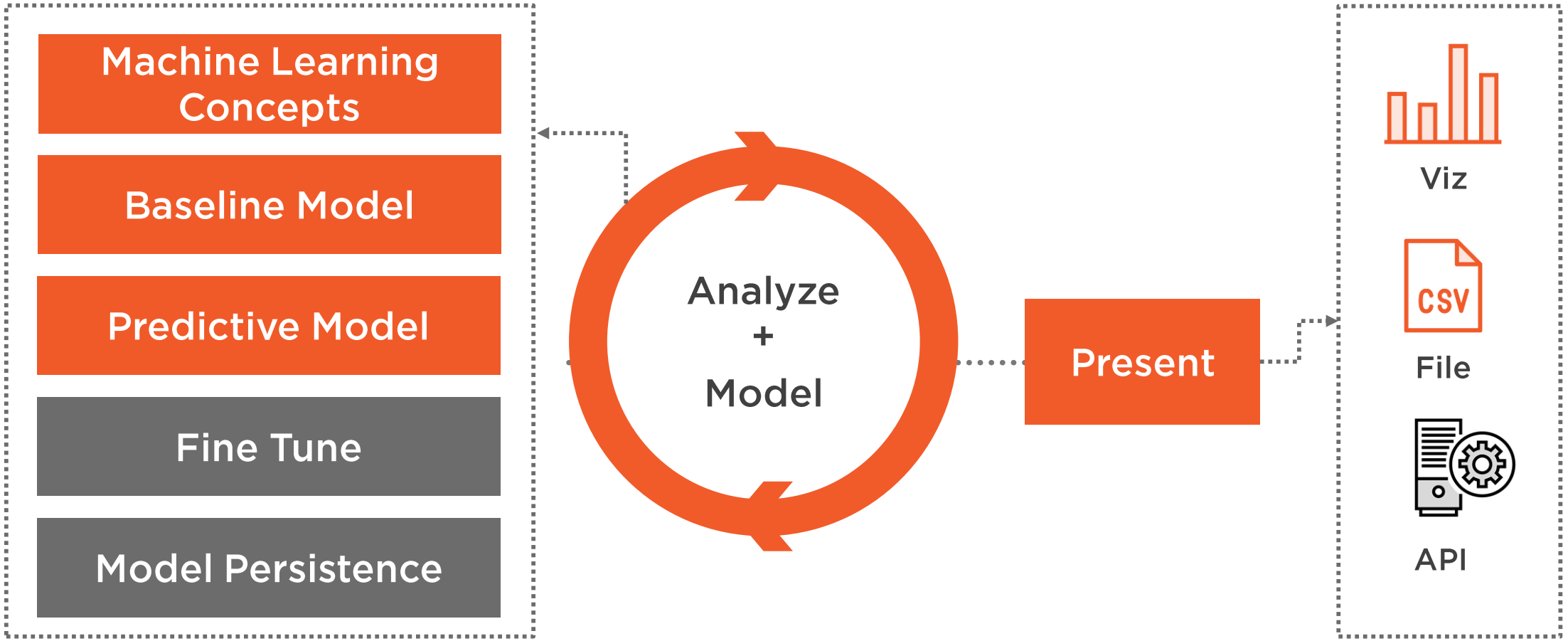
**Abhishek Kumar**
AUTHOR

@meabhishekkumar

# Data Science Project Cycle



# Data Science Project Cycle



# Overview (Concepts)

- Machine learning basics
- Titanic disaster challenge
- Classifier
- Metrics
- Baseline model
- Logistic regression model

# Overview (Tools)

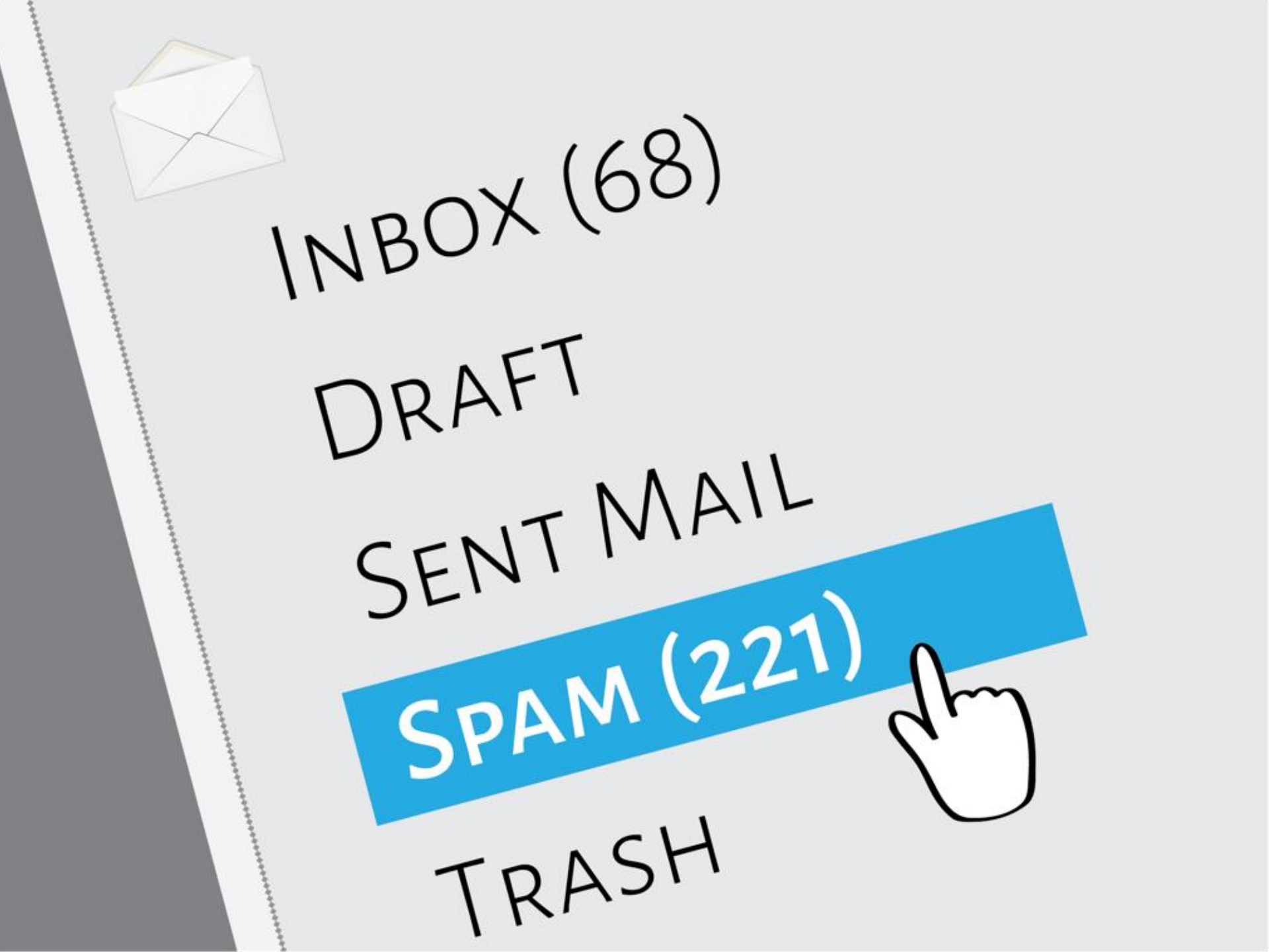
**Python**

- Numpy
- Pandas
- Scikit-Learn

# Machine Learning Basics

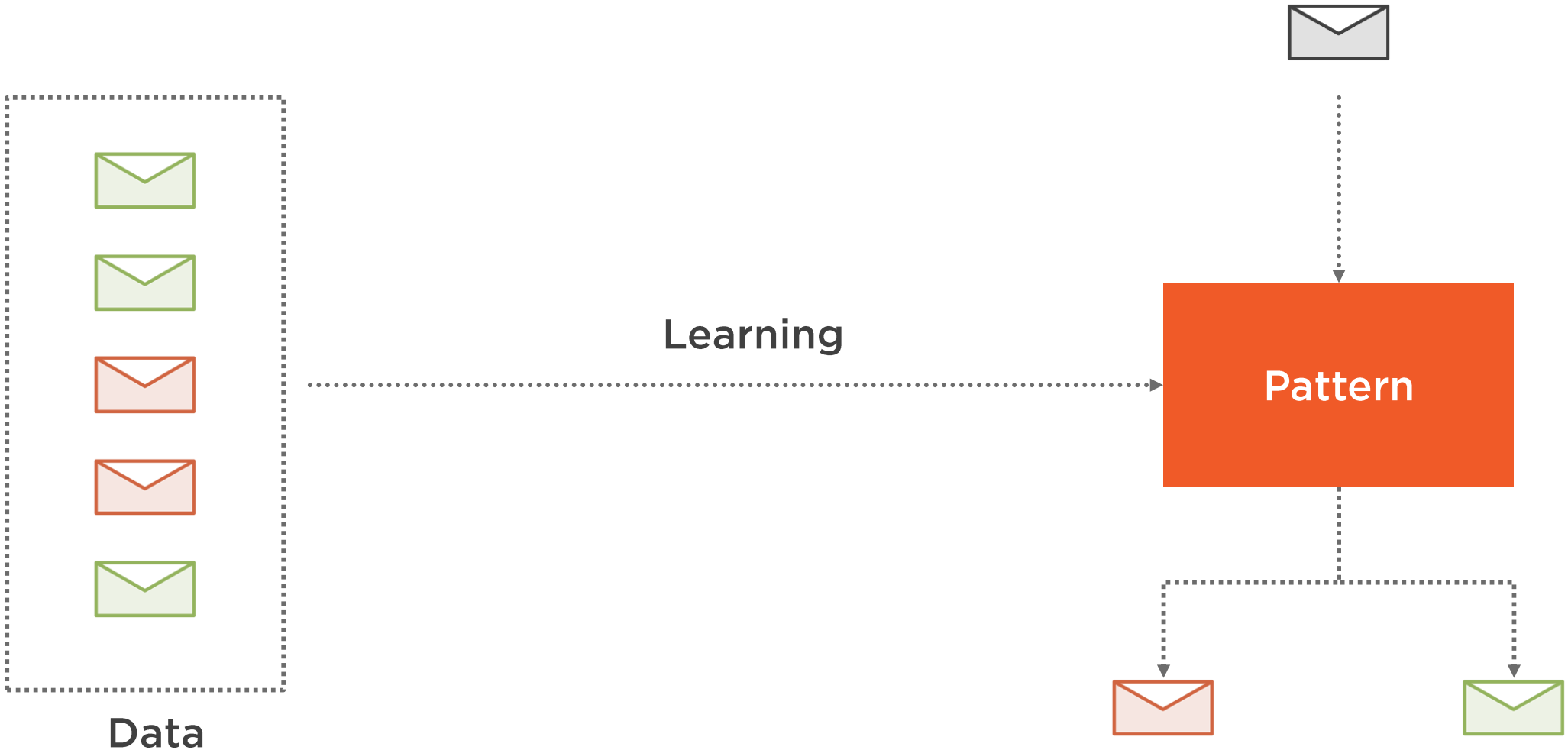
# Machine Learning

Learning from data or examples

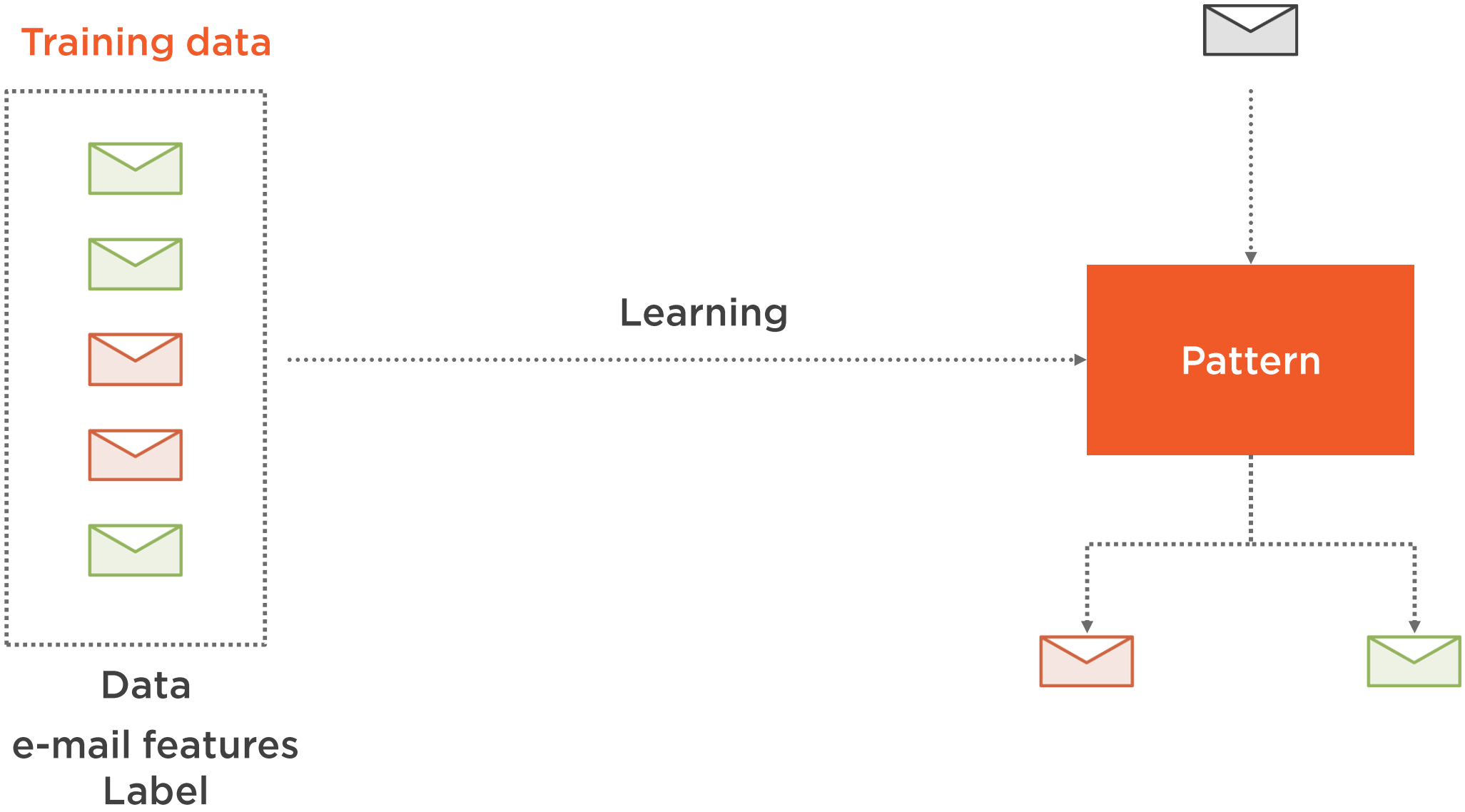



# Spam Detection

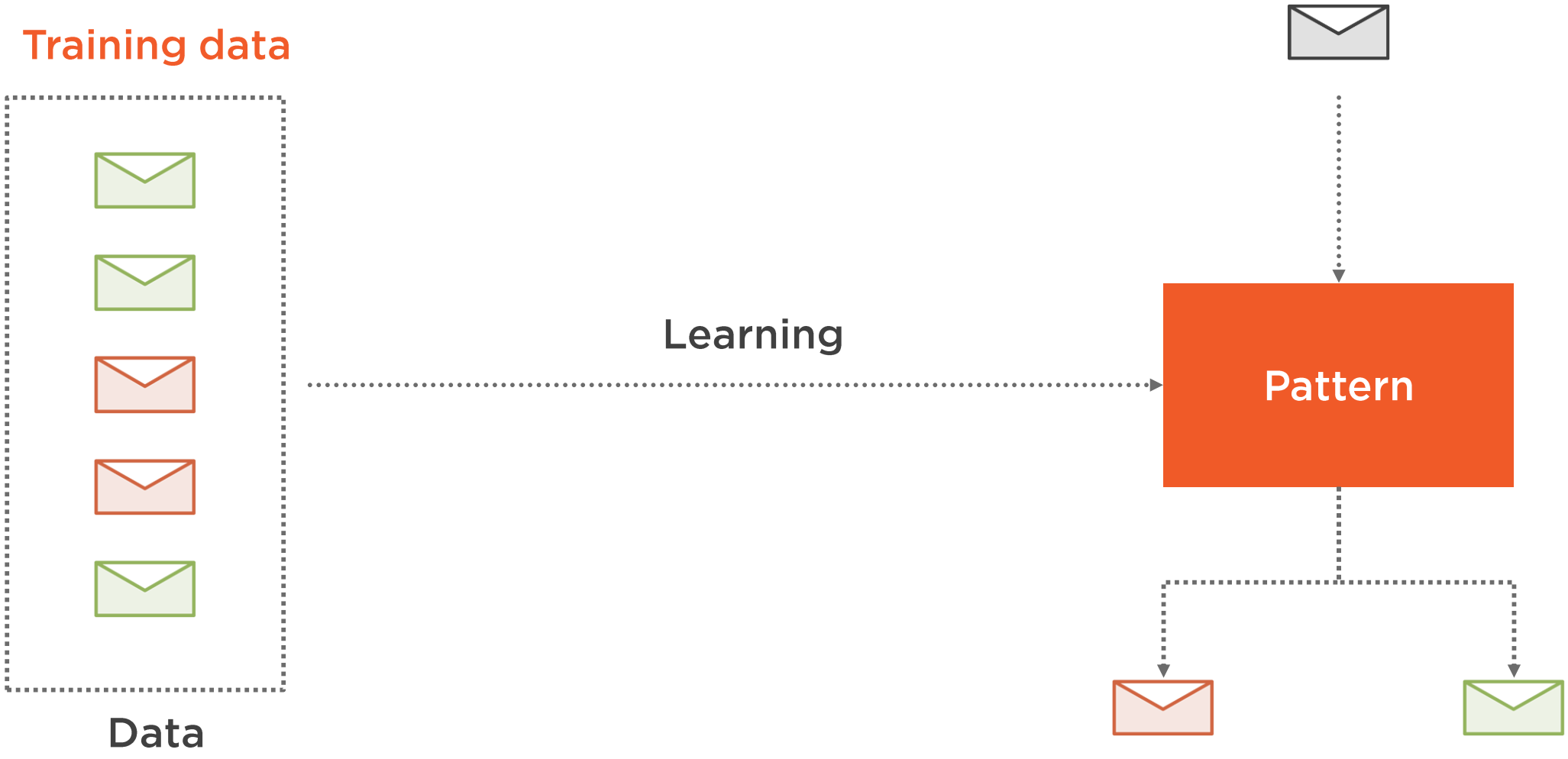
# Spam Detection



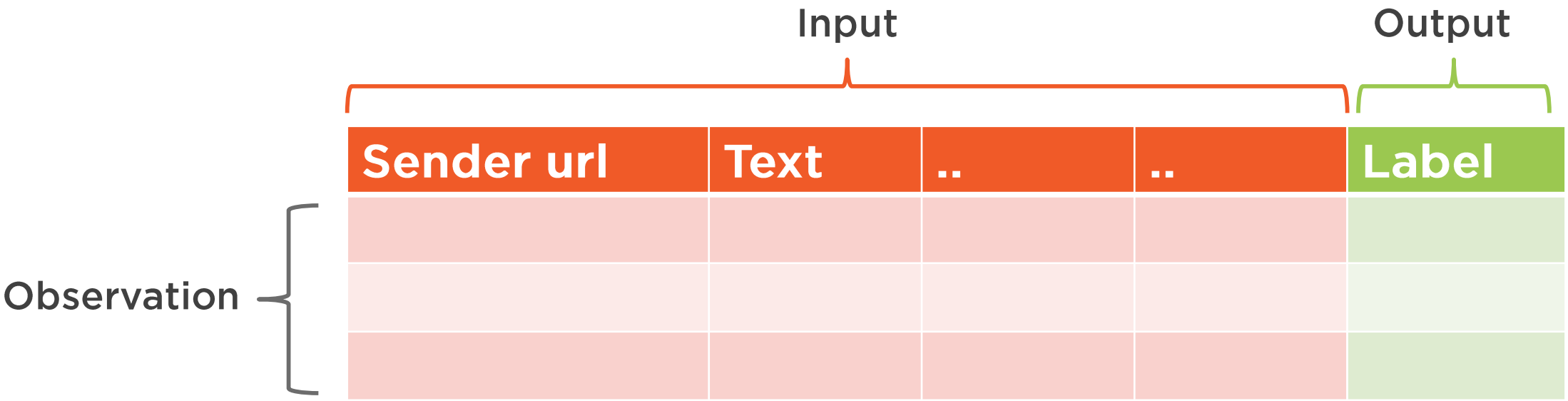
# Training Data



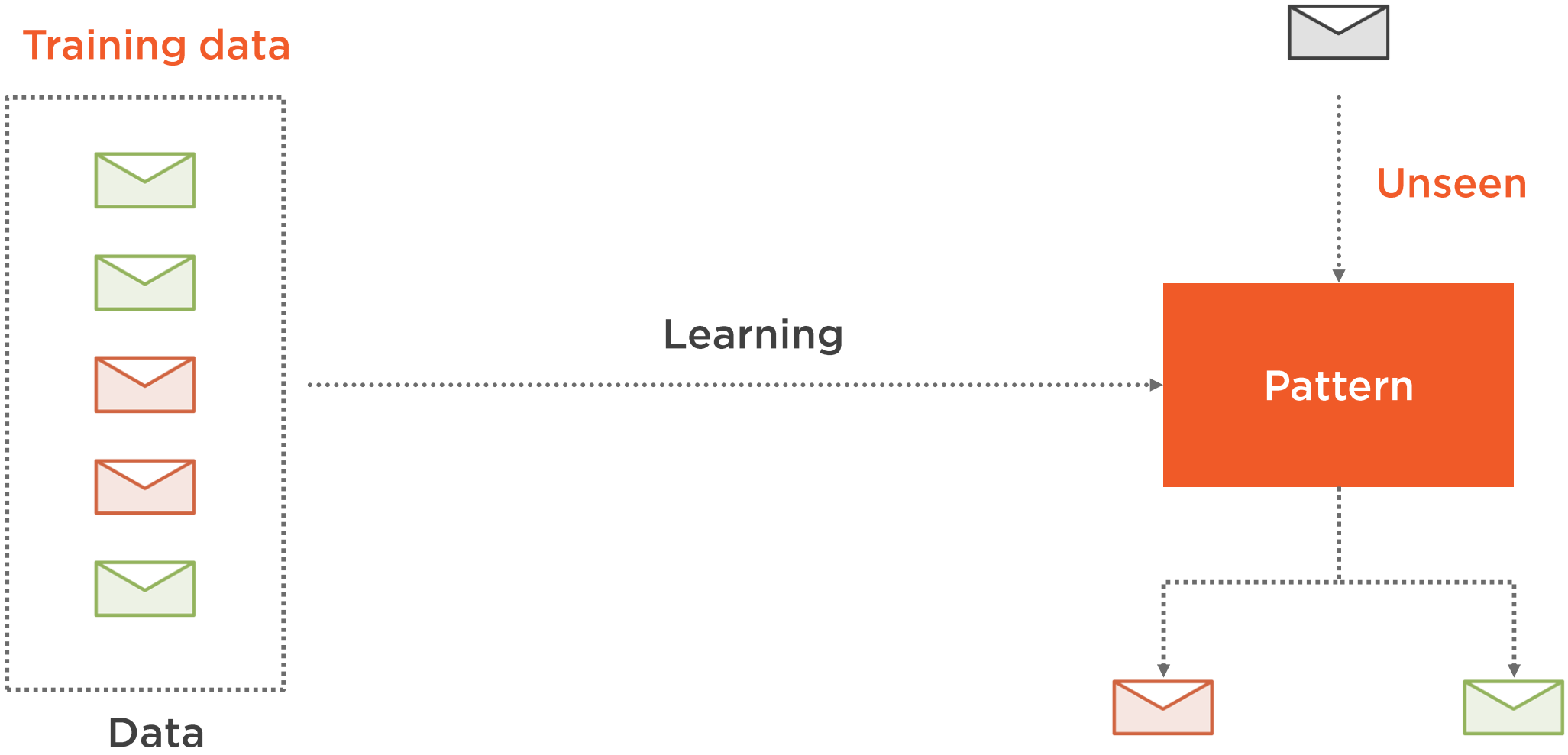
# Input and Output Feature



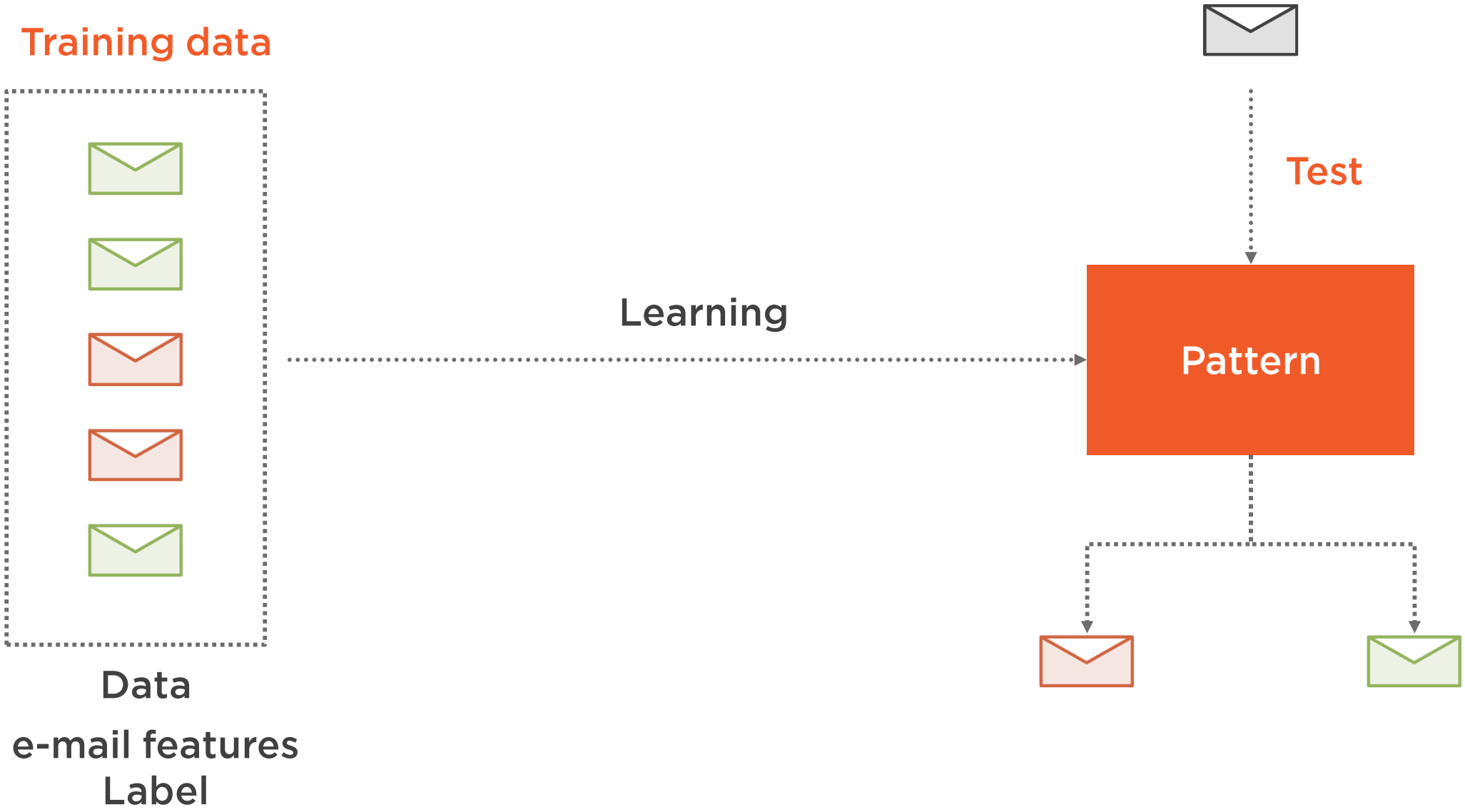
# Representation



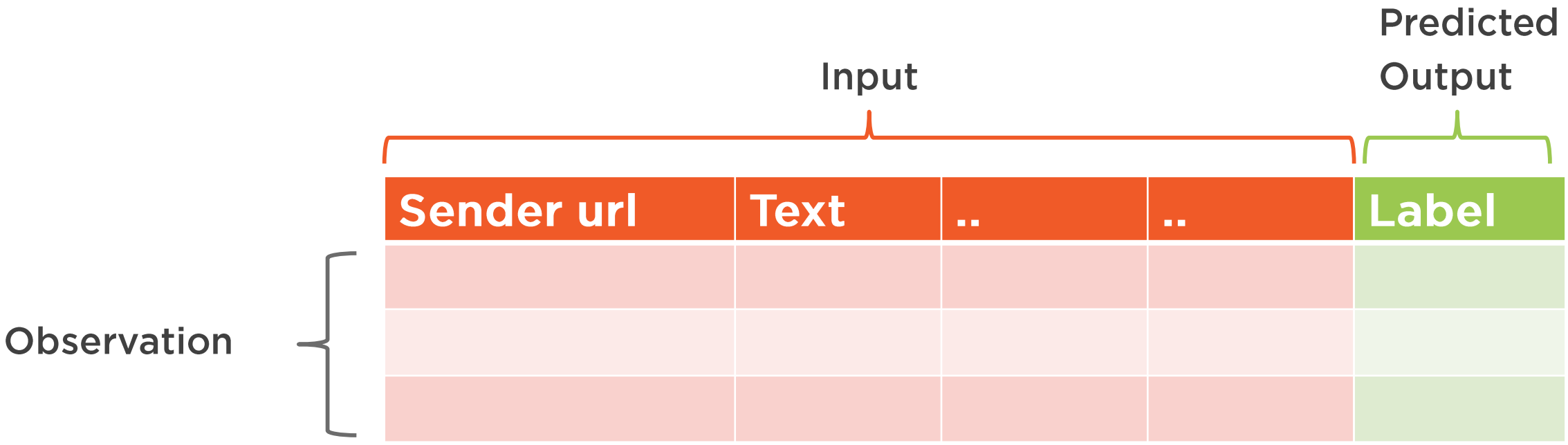
# Generalization



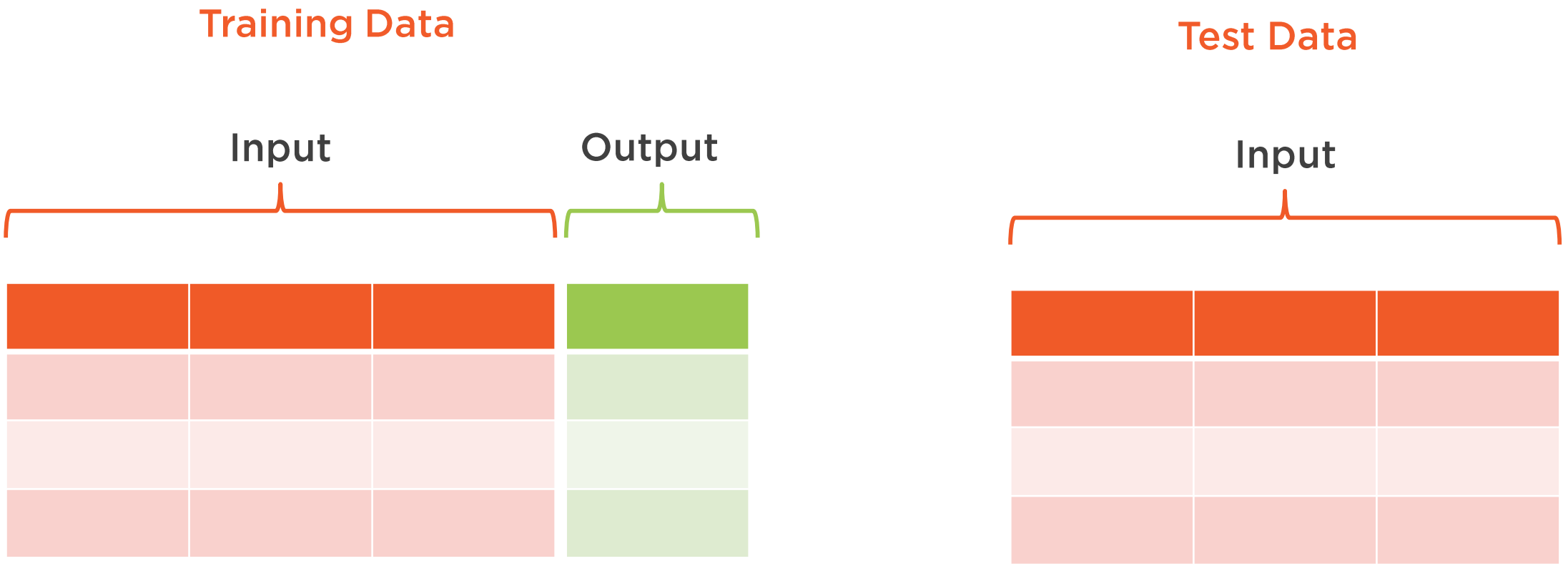
# Generalization



# Test Data



# Supervised Learning



# Titanic Disaster

## Training Data

Input

Output

| Age | Fare | ... | Survived |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

## Test Data

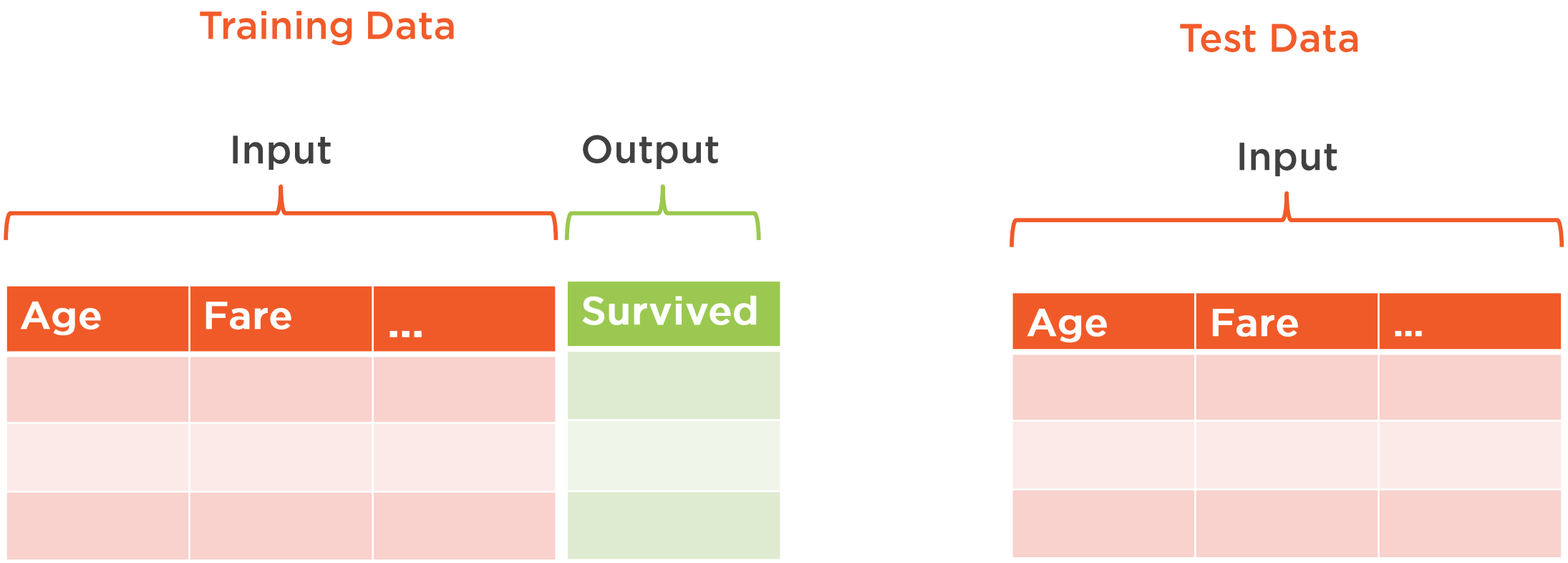
Input

| Age | Fare | ... |
|---|---|---|
| | | |
| | | |
| | | |

Class : Survived ( 1 )

Class : Not Survived ( 0 )

# Classification




Output : Discrete labels

Class : Survived ( 1 )

Class : Not Survived ( 0 )

# Regression

## Training Data

Input

Output

| Height | Width | .. | Mileage |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

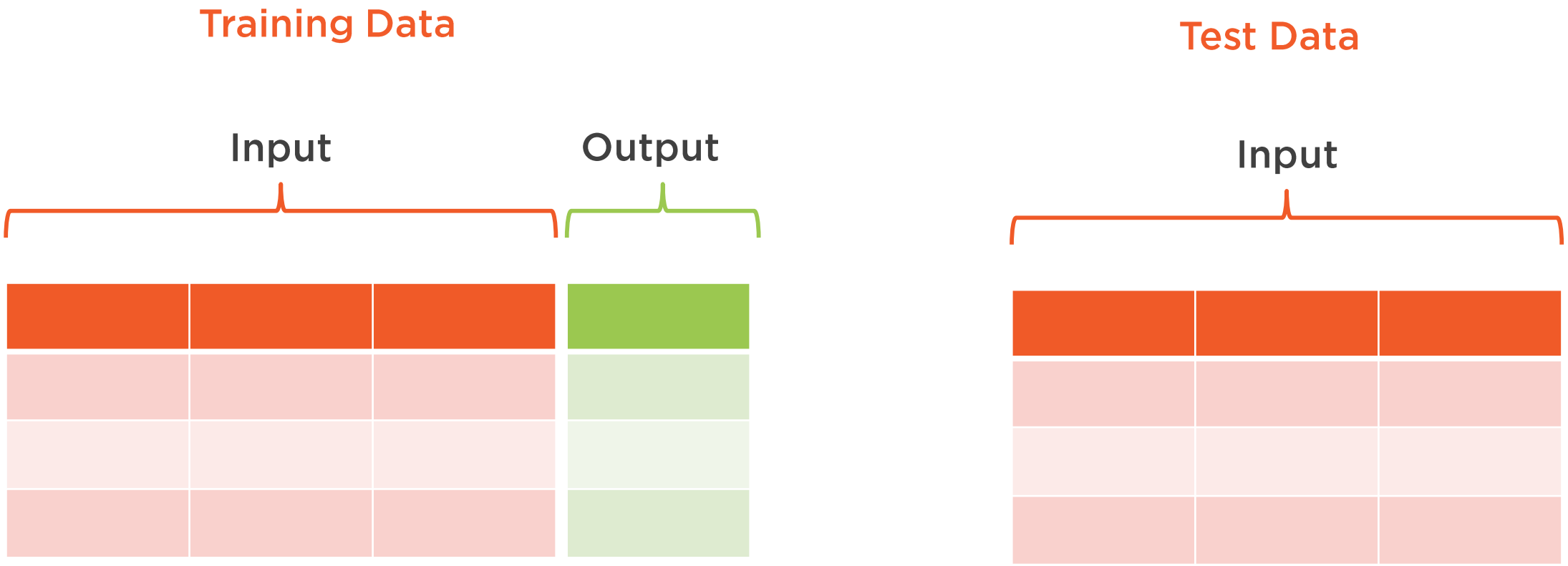
## Test Data

Input

| Height | Width | ... |
|---|---|---|
| | | |
| | | |
| | | |

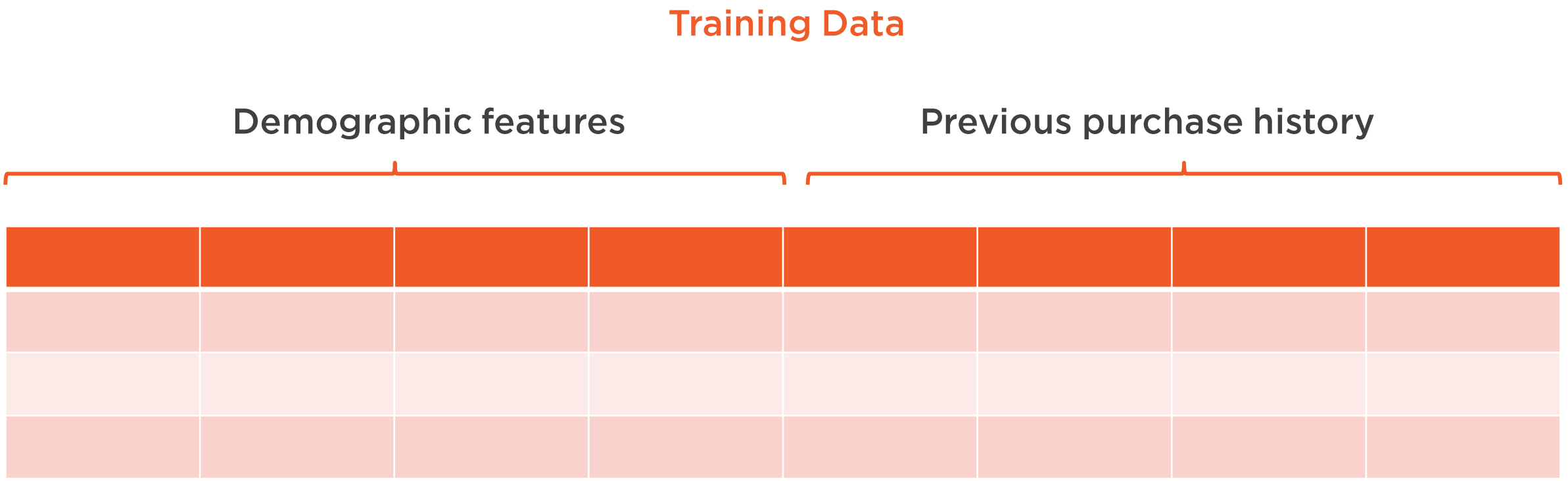
Output : Continuous values

# Unsupervised Learning



# Customer Segmentation

# Customer Segmentation



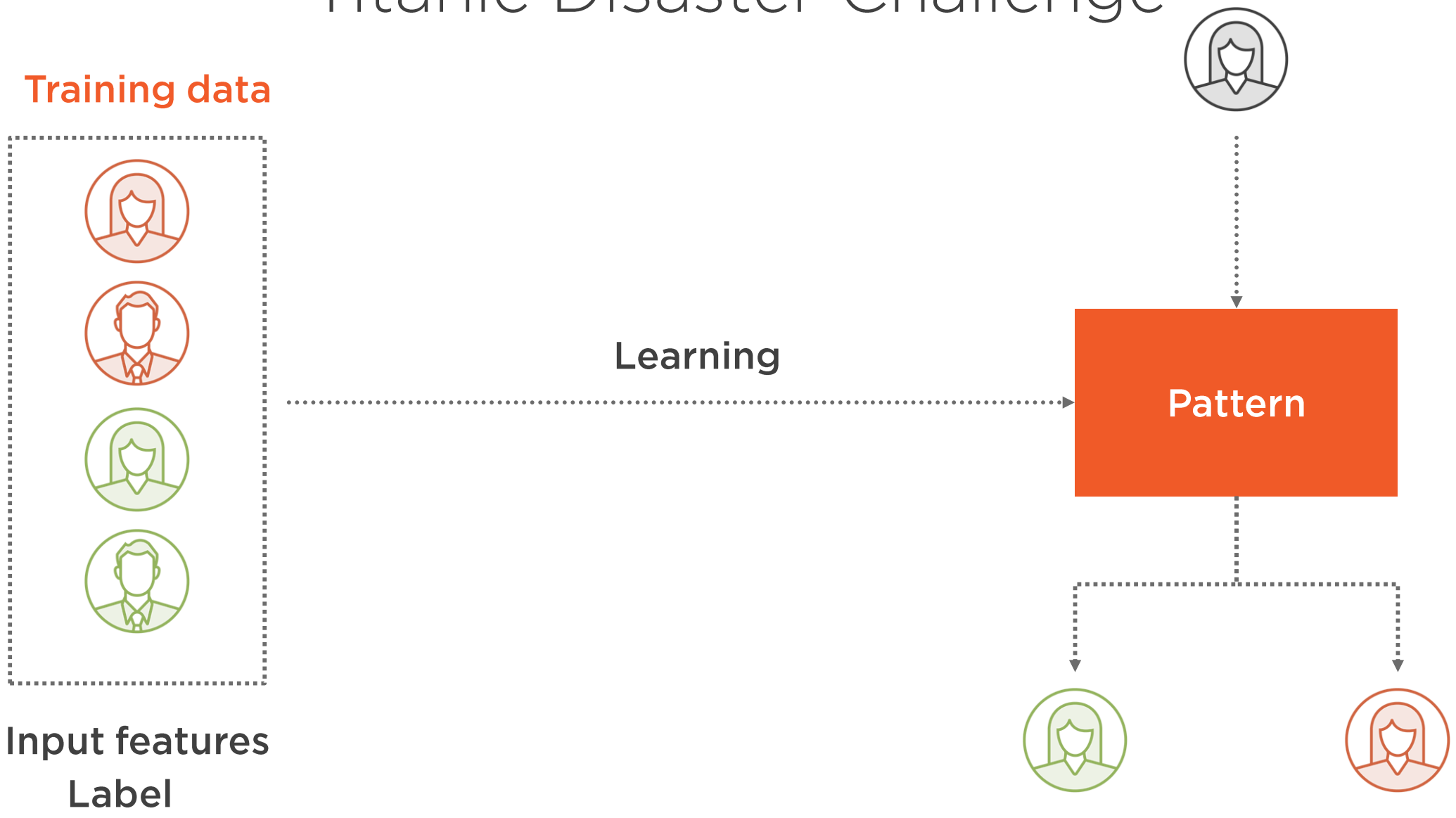
# Customer Segmentation

# Clustering

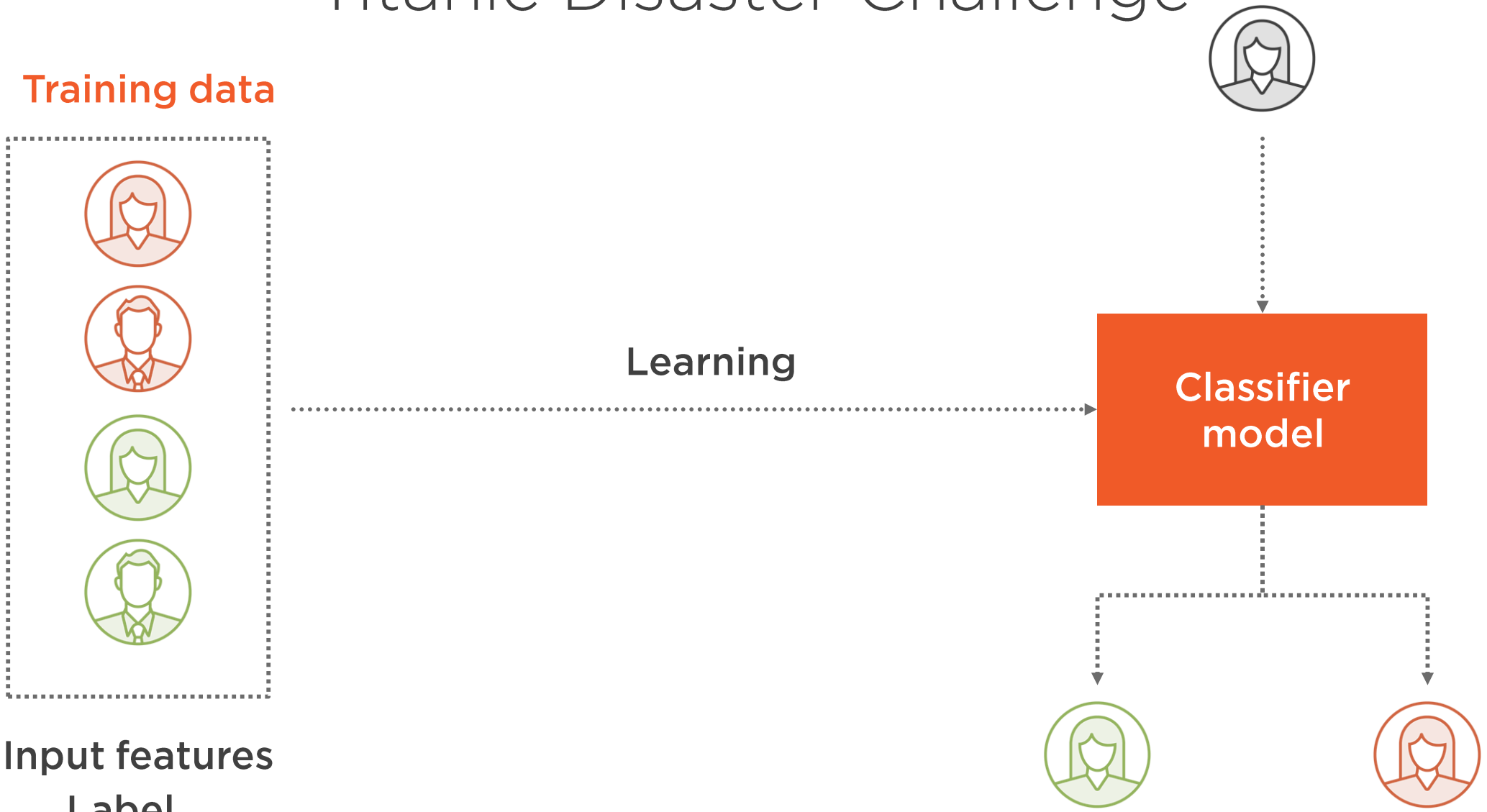
# Titanic Disaster Challenge

- Both input and output in training data
- Supervised learning problem
- Classification task
- Binary classification

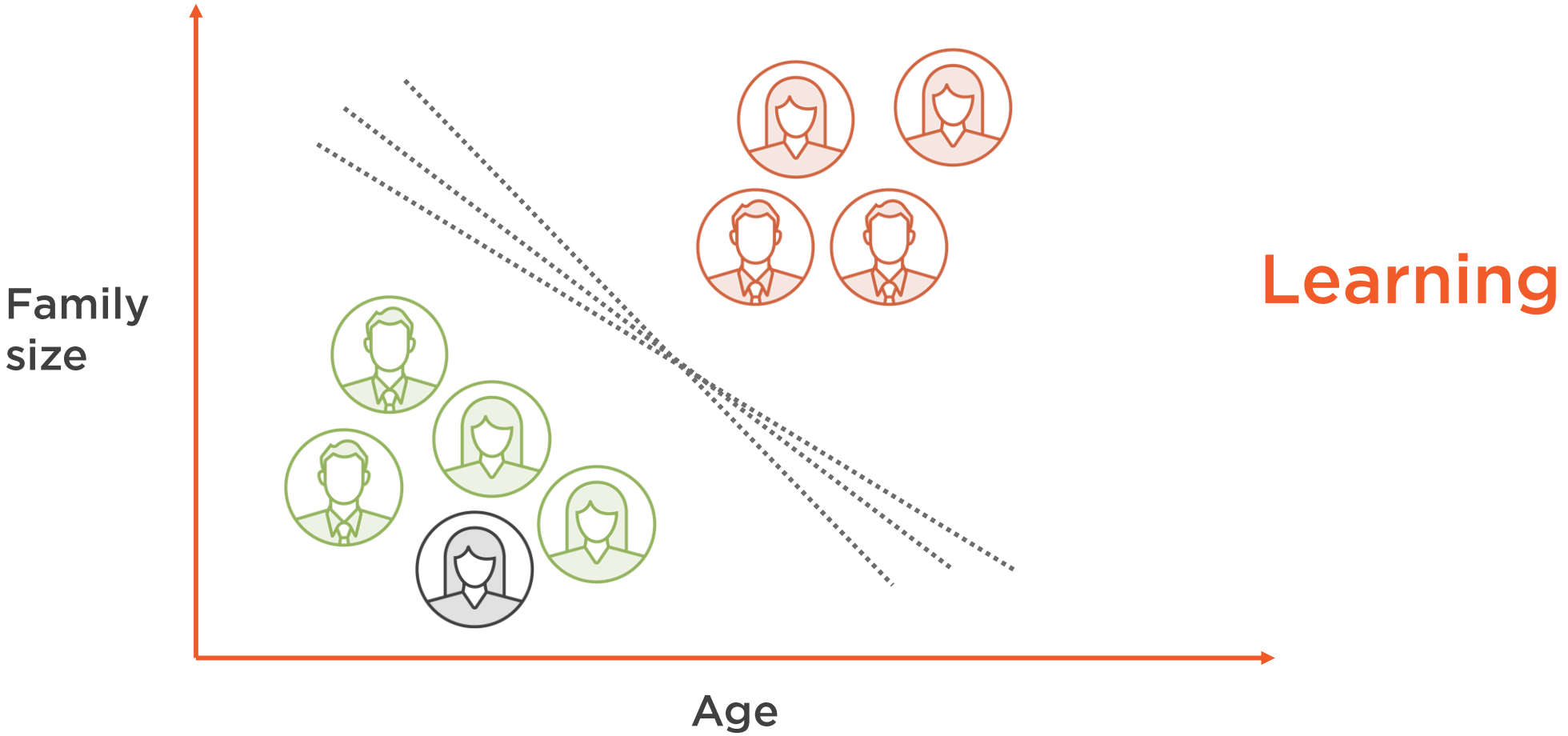
# Titanic Disaster Challenge



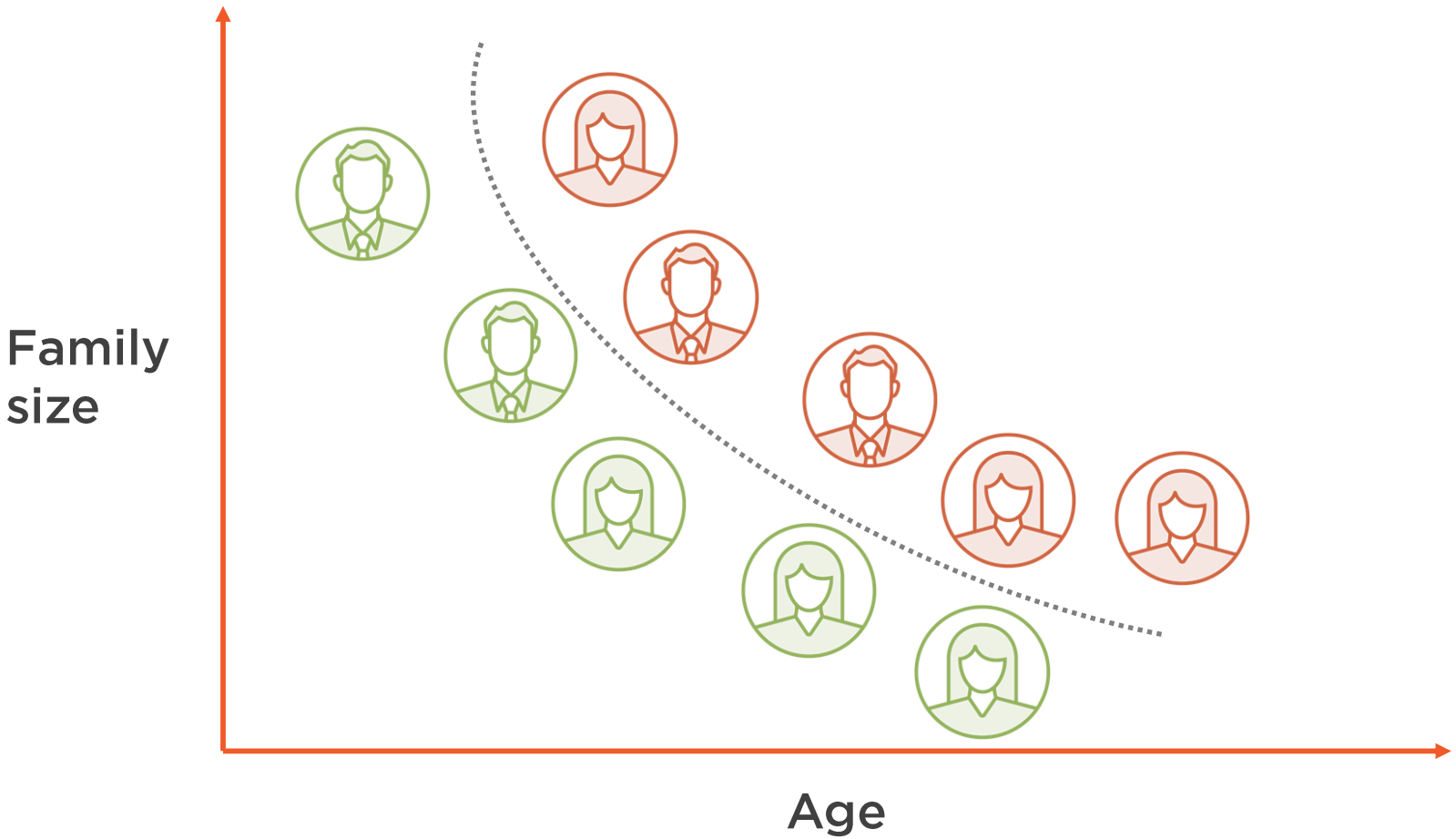
# Titanic Disaster Challenge

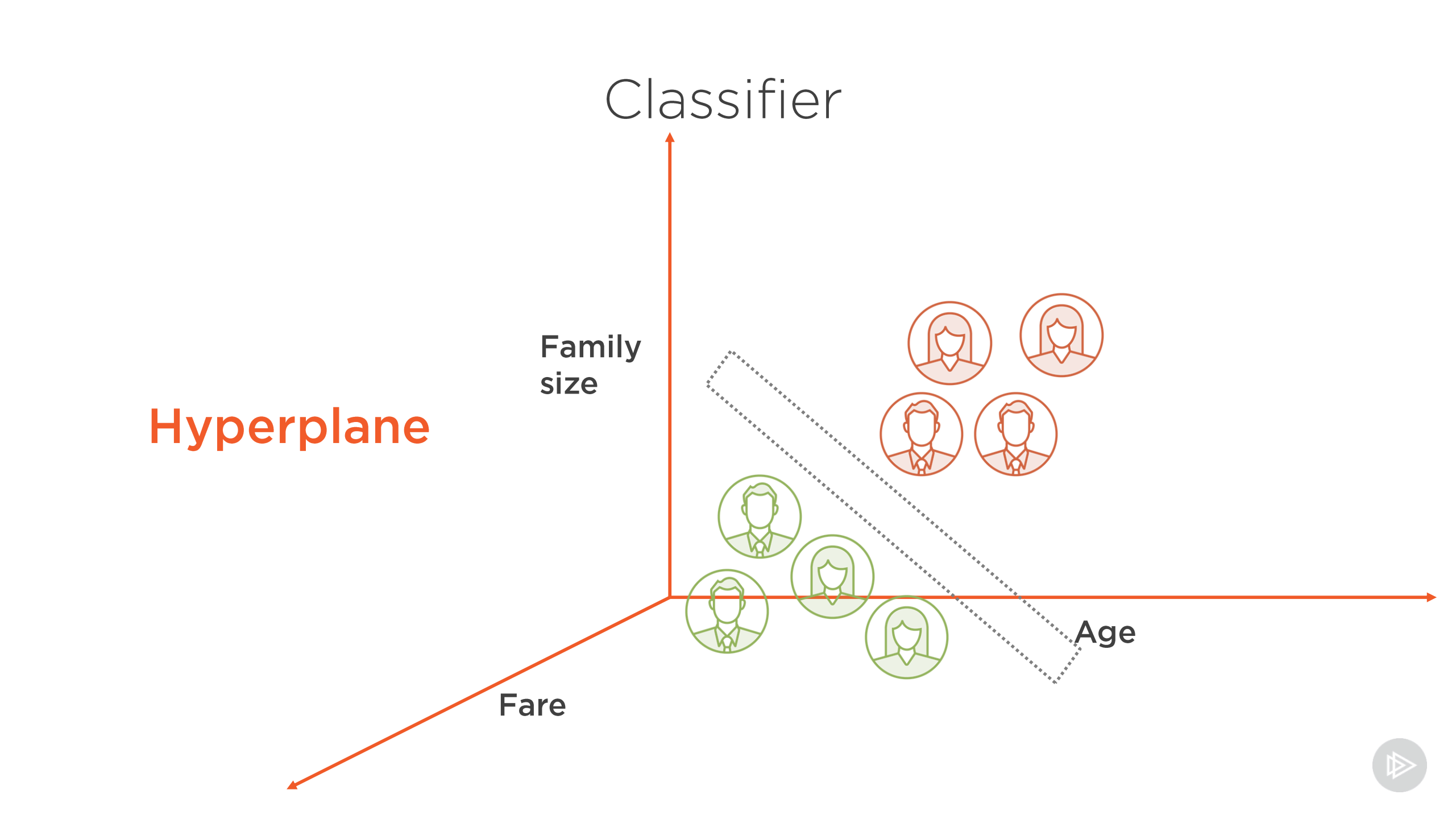


# Classifier

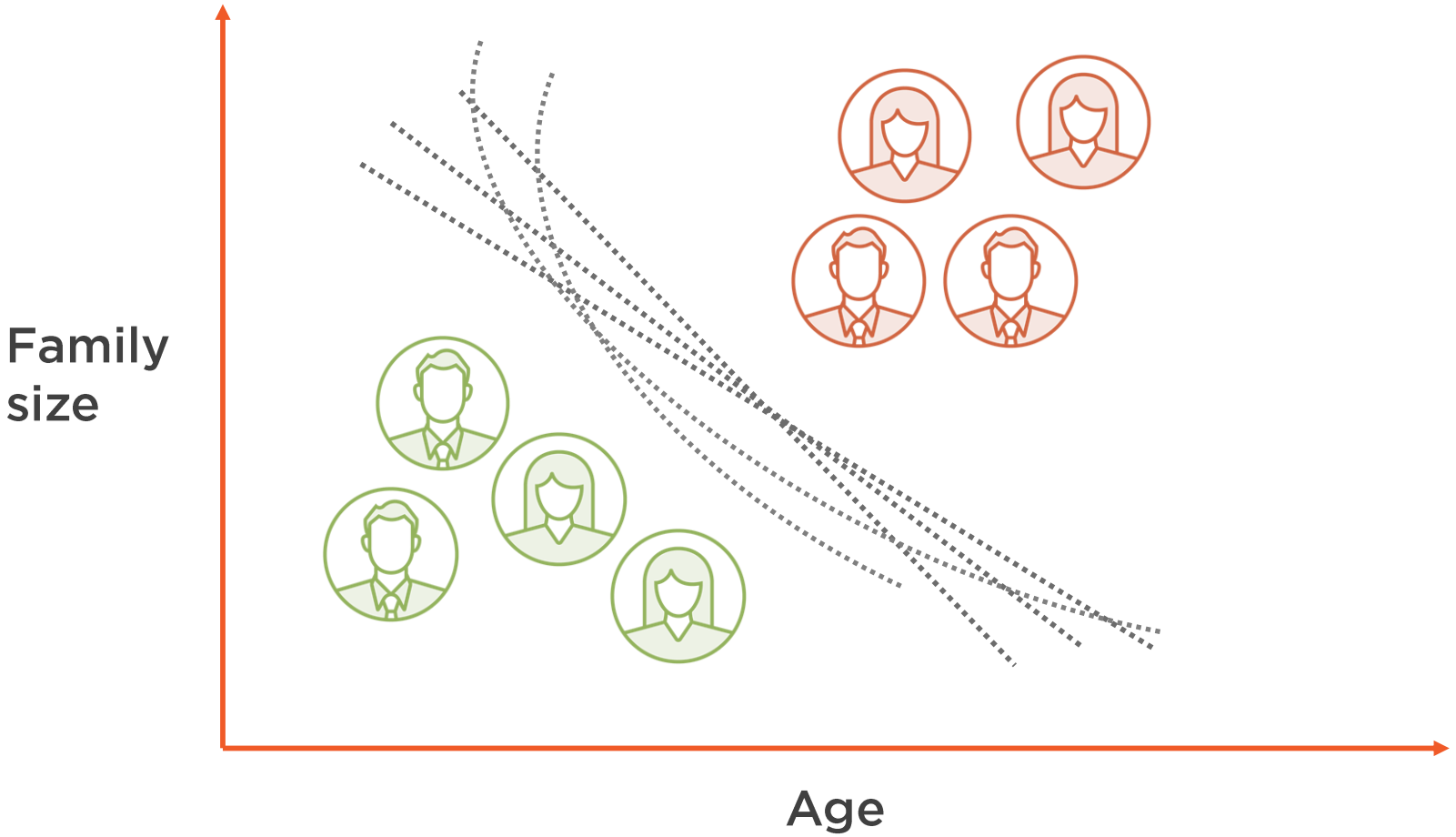


# Classifier




Classifier
Family size
Hyperplane
Age
Fare

# Classifier



# Classifiers

- Logistic regression
- Support vector machine
- Neural networks
- Random forest

“If you can’t measure it, you can’t improve it.”

**Peter Drucker**

# Performance Metrics

**Accuracy**

**Precision**

**Recall**

# Accuracy

| Id | $F_1$ | ... | $F_n$ | Output | Output | |
|---|---|---|---|---|---|---|
| 1 | | | | 1 | 1 | 1 |
| 2 | | | | 0 | 0 | 2 |
| 3 | | | | 1 | 0 | |
| 4 | | | | 1 | 1 | 3 |
| 5 | | | | 0 | 1 | |
| 6 | | | | 1 | 1 | 4 |
| 7 | | | | 1 | 0 | |
| 8 | | | | 0 | 1 | |
| 9 | | | | 1 | 1 | 5 |
| 10 | | | | 1 | 1 | 6 |

Test data

Predicted output

Actual output

Accuracy = Correct count / Total Count

Accuracy = 6 / 10 = 0.6 ( 60%)

# Precision

## Confusion Matrix

| | Predicted Negative | Predicted Positive |
|---|---|---|
| Actual Negative | True Negative ( TN ) | False Positive ( FP ) |
| Actual Positive | False Negative ( FN ) | True Positive ( TP ) |

| | Predicted Negative | Predicted Positive |
|---|---|---|
| Actual Negative | 20 | 40 |
| Actual Positive | 30 | 60 |

## Precision

**What fraction of positive predictions are correct?**

$$\frac{TP}{Total\ Positive\ Predictions}$$

$$\frac{TP}{TP + FP}$$

**Precision = 60 / (60 + 40 ) = 0.6**

# Recall

## Confusion Matrix

| | Predicted Negative | Predicted Positive |
|---|---|---|
| Actual Negative | True Negative ( TN ) | False Positive ( FP ) |
| Actual Positive | False Negative ( FN ) | True Positive ( TP ) |

| | Predicted Negative | Predicted Positive |
|---|---|---|
| Actual Negative | 20 | 40 |
| Actual Positive | 30 | 60 |

## Recall

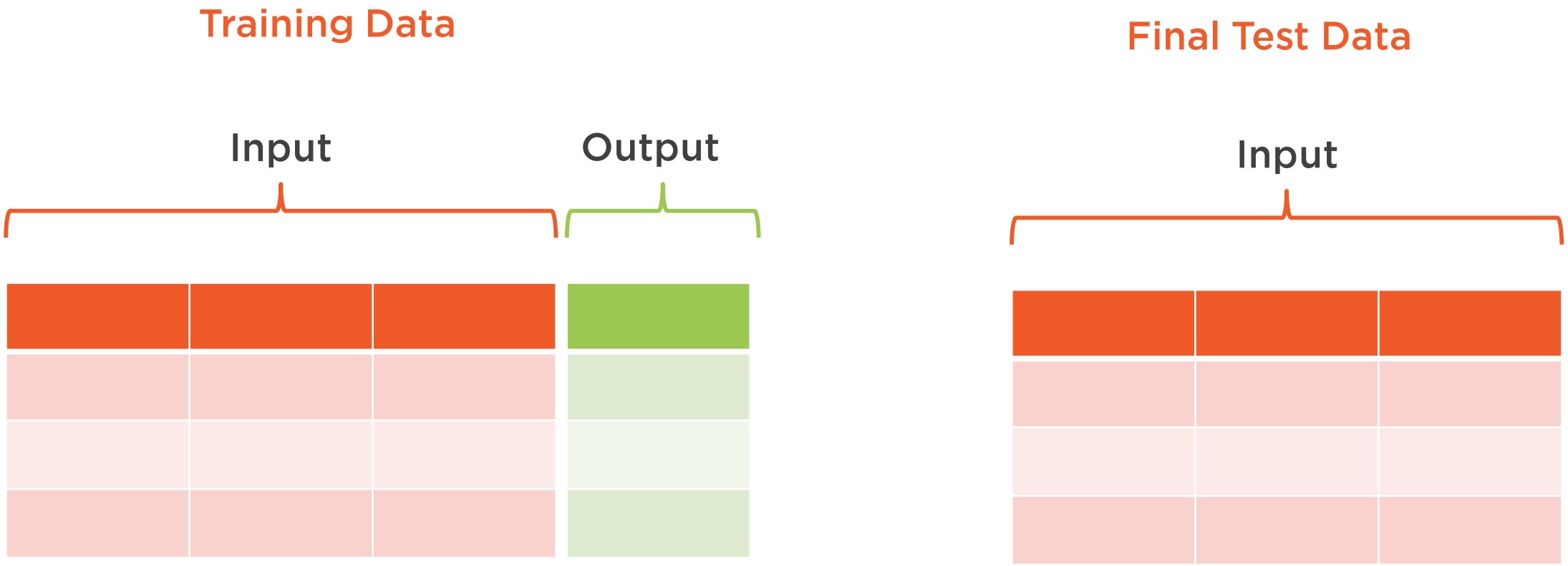
What fraction of positive cases you predicted correctly?

$$\frac{TP}{Total\ Positive\ Cases}$$

$$\frac{TP}{TP + FN}$$

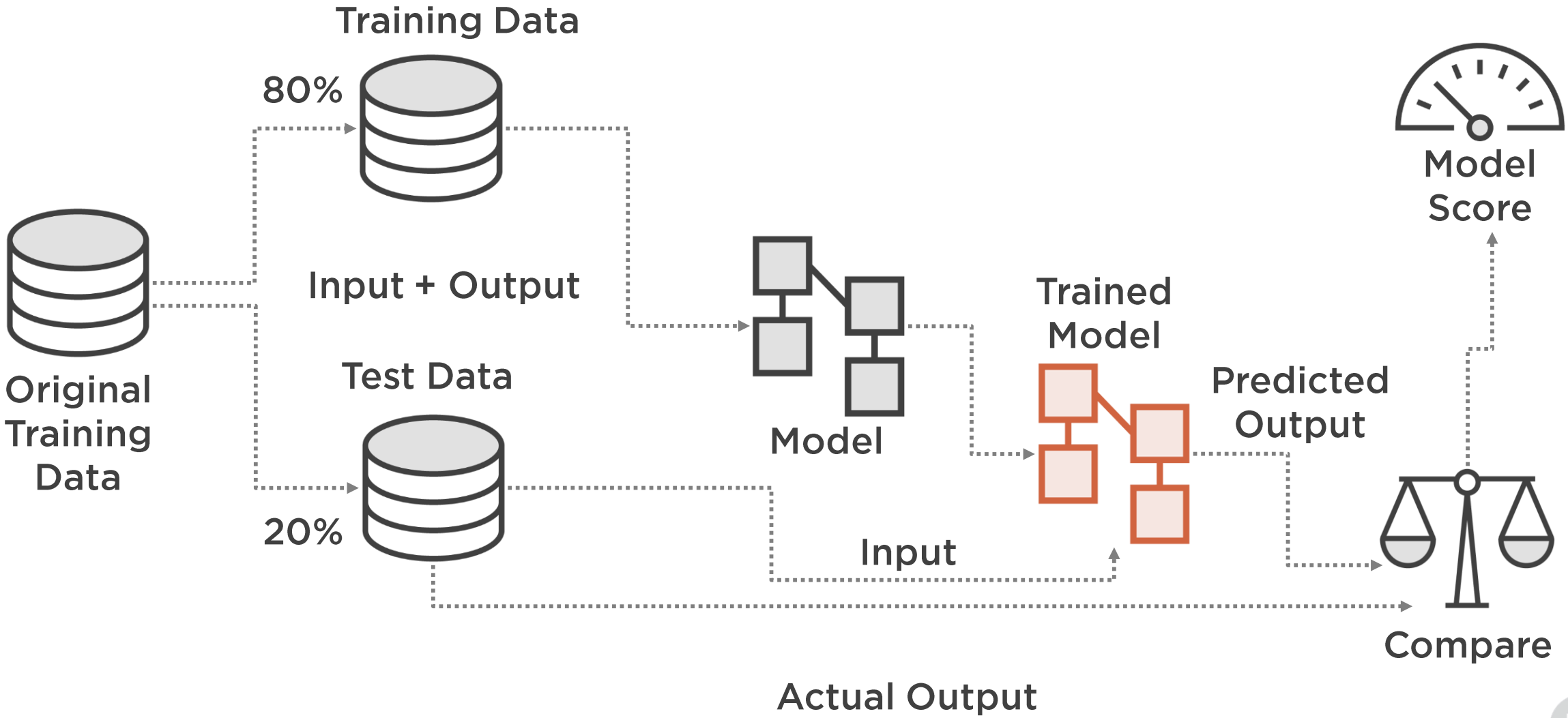
Recall = 60 / (60 + 30 ) = 0.67

# Classifier Evaluation




Train Test Split

# Classifier Evaluation



# Baseline Model

**First Step**

**Don’t Skip**

**Compare**

# Baseline Model for Classification

| Class | Count |
| --- | --- |
| 1 | 60 |
| 0 | 40 |

Baseline model = Class 1

Baseline model accuracy = 60 / ( 60 + 40 ) = 0.6

- Output majority class
- Predictive model should have better performance than baseline

# Demo

## Preparing data for machine learning model

# Demo

## Building and evaluating baseline model

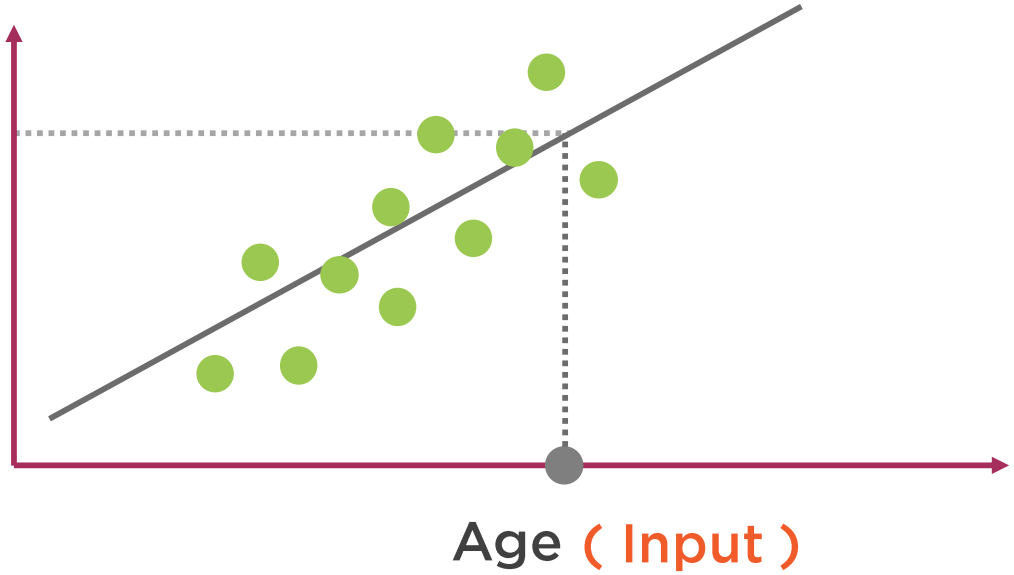
# Demo

## Making first Kaggle submission

# Linear Regression Model

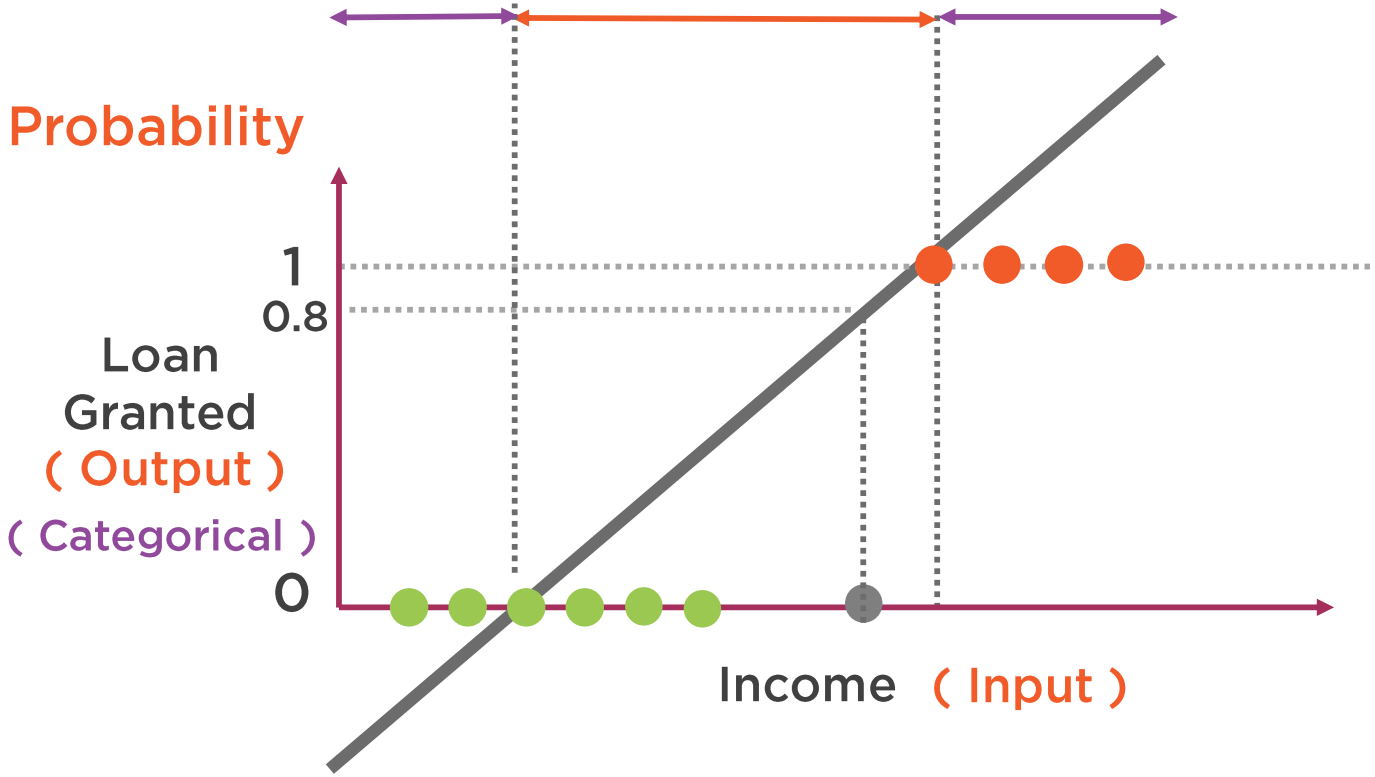
$$Y = c + m * x$$

$$Income = b0 + b1 * Age$$



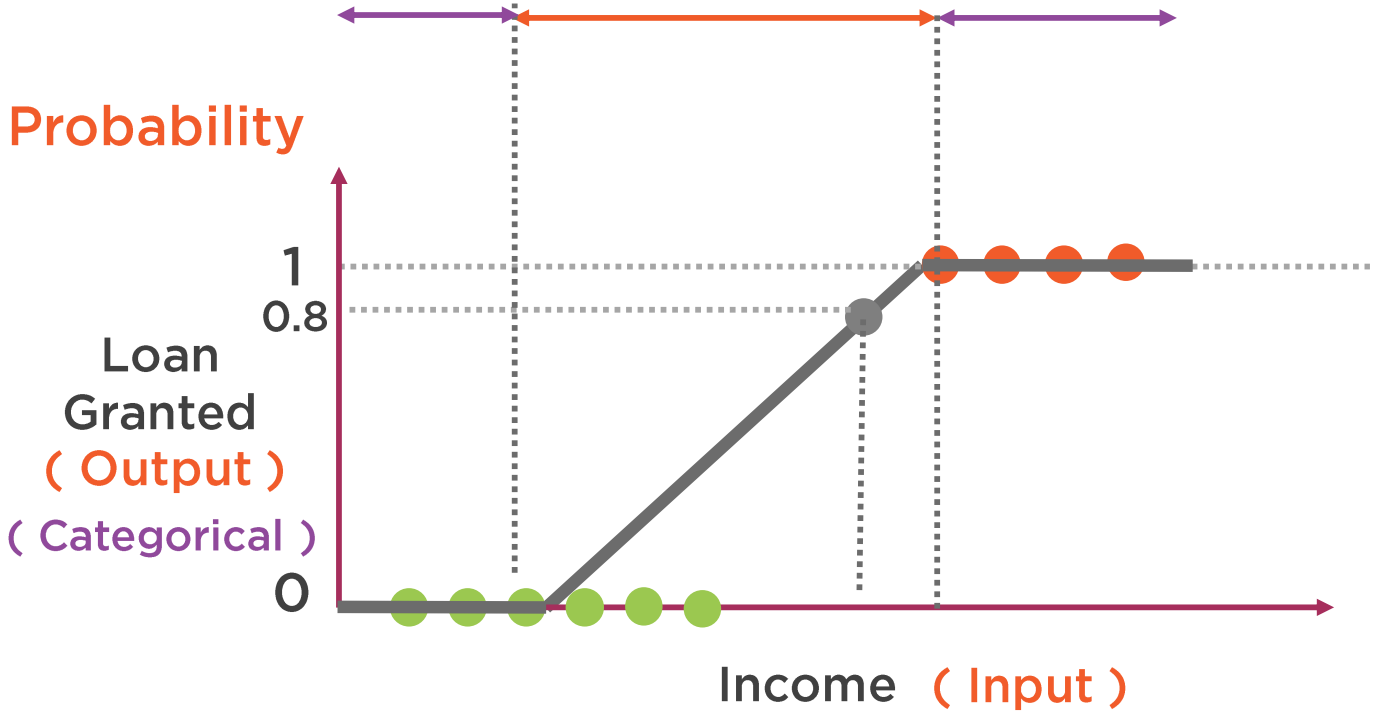

- Supervised learning problem
- Regression task
- Model coefficients : b0, b1

# Logistic Regression Model



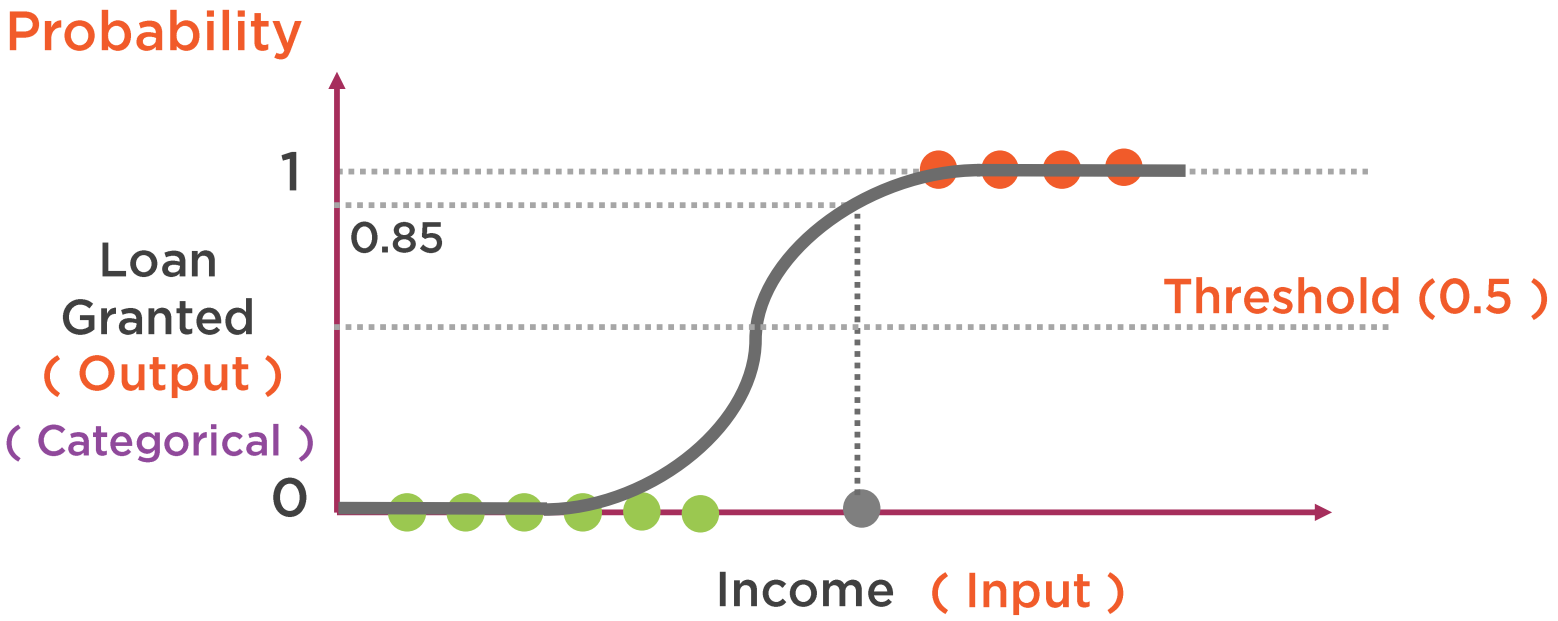

- Supervised learning problem
- Classification task
- Binary classification

# Logistic Regression Model



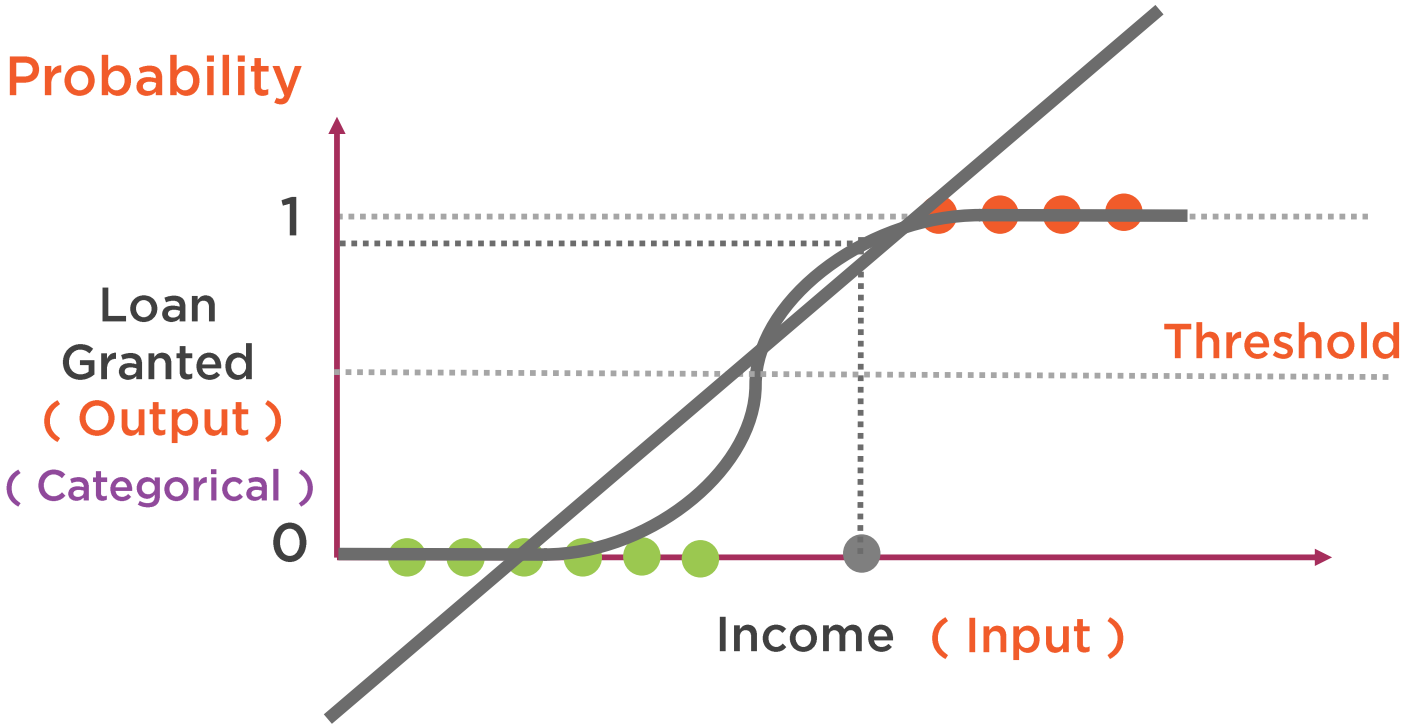

- Supervised learning problem
- Classification task
- Binary classification

# Logistic Regression Model




- Supervised learning problem
- Classification task
- Binary classification
- Sigmoidal curve

# Logistic Regression Model




Granted = b0 + b1 * income 1

$P = \frac{1}{1+ e^{-Granted}}$ 2

P > threshold : Class 1

P <= threshold : Class 0

# Demo

## Building logistic regression using Scikit-Learn

# Demo

## Making second Kaggle submission

# Summary

Machine learning foundation

Baseline model

Logistic regression model