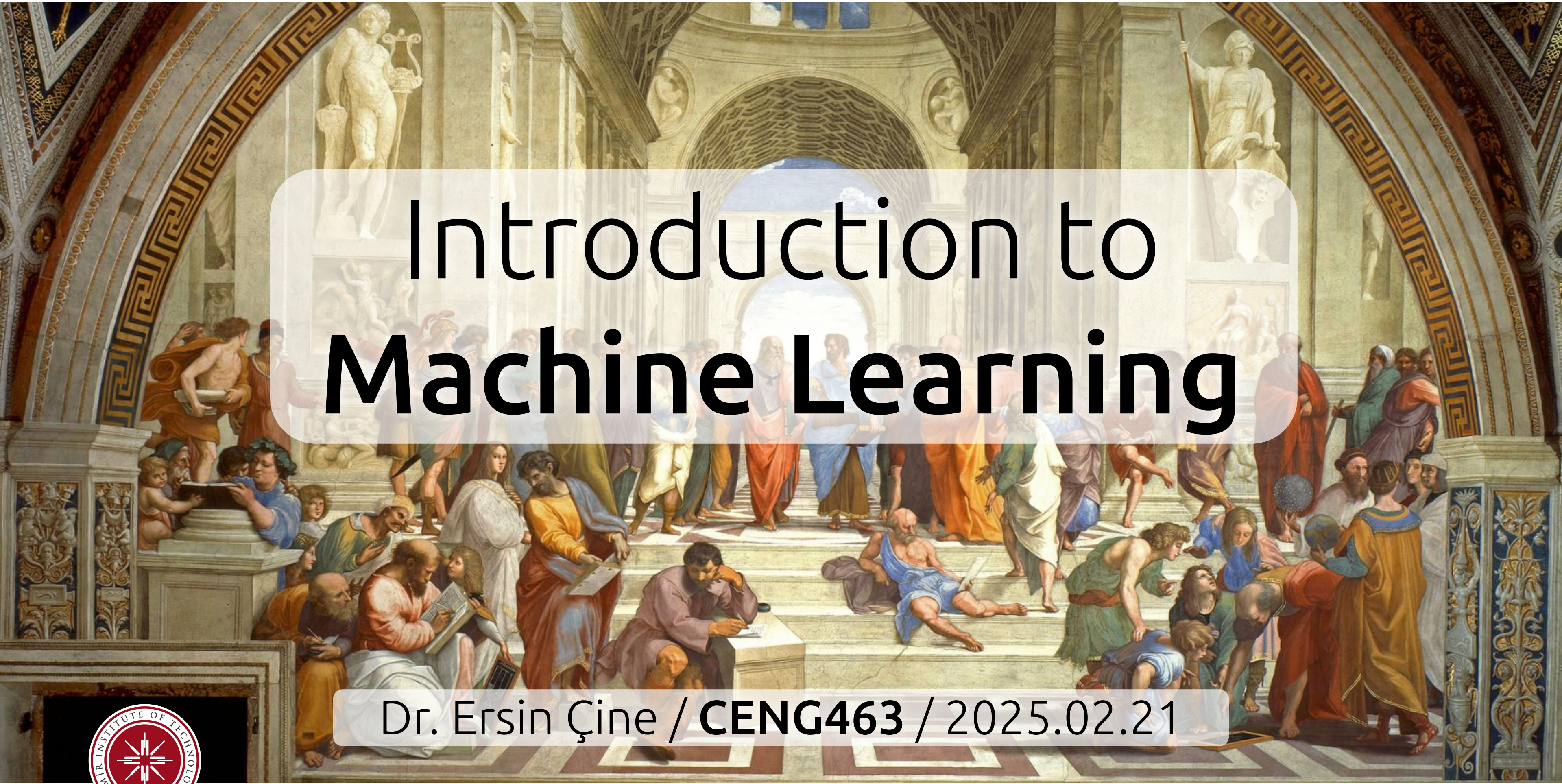


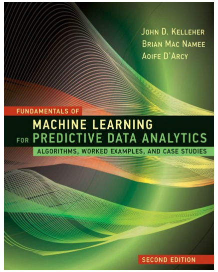
Introduction to **Machine Learning**

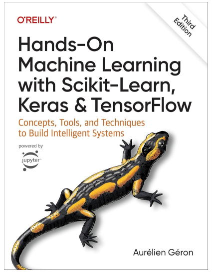
Dr. Ersin Çine / **CENG463** / 2025.02.21

Course Overview (Tentative)

Instructor: **Dr. Ersin Çine** (Email: **ersincine@iyte.edu.tr**; Office hours: **Monday 14:30–16:30**)
Assistant: **Ceren Sözeri** (Email: **cerensozeri@iyte.edu.tr**; Office hours: **Thursday 13.30–15.30**)


Primary Textbook: **Fundamentals of Machine Learning for Predictive Data Analytics** (Second Edition)
Supplementary Textbook: **Hands-On Machine Learning with Scikit-Learn & TensorFlow** (Third Edition)





Grading Policy: **20% Assignments + 10% Quizzes + 30% Midterm Exam + %40 Final Exam**

Syllabus (Tentative)

Introduction: 1 Week
Information-Based Learning: ~2 Weeks (1, 3.6, 4.1, 4.2, 4.3, 4.4.4, 4.4.5)
Similarity-Based Learning: ~1 Week (5.1, 5.2, 5.3, 5.4.1, 5.4.3, 5.4.6)
Probability-Based Learning: ~1 Week (6.1, 6.2, 6.3, 6.4.1)
Error-Based Learning: ~2 Weeks (7.1, 7.2, 7.3, 7.4.4, 7.4.5, 7.4.6, 7.4.7)
Deep Learning: ~2 Weeks (8.1, 8.2, 8.3)
Evaluation: ~1 Week (9.1, 9.2, 9.3, 9.4.1, 9.4.2, 9.4.3, 9.4.4, 9.4.5)
Unsupervised Learning: ~1 Week (10.1, 10.2, 10.3)
Reinforcement Learning: ~1 Week (11.1, 11.2, 11.3)
The Art of Machine Learning and Next Steps: ~1 Week (14)

What Is Artificial Intelligence (AI)?

AI is a field that focuses on creating systems capable of performing **tasks that typically require human intelligence**, such as problem-solving, reasoning, planning, perceiving the environment through vision, and understanding language.

What Is Machine Learning (ML)?

ML is a subset of AI.

"ML is the field of study that gives computers the ability to learn without being explicitly programmed."

"True **intelligence** is not about knowing everything,
but about knowing how to **learn** anything."

AI and ML Applications

What are some applications of AI and ML?

How can they be used by **students**, **engineers**, **scientists**, **doctors**, **lawyers**, **marketers**, **artists**, **entrepreneurs**, **professionals in other fields**, and **everyday people** in their daily lives?
(Share what you know or dream up something exciting!)

Plato

Idealist

Theoretical & abstract

Classical AI (Logic)

Aristotle

Realist

Practical & experimental

Machine Learning (Statistics)

Chess-Playing AI

Classical AI

Search possible moves and choose the move that leads to the best position (i.e., worst position for the opponent).

Manually defined **rules** (or heuristics):

- A bishop is three times more valuable than a pawn.
- Knights are better positioned in the center.
- …

Machine Learning

Predict the best move based on the huge training data (i.e., experience).

Compiled **examples** from past games:

- In a grandmaster's match, White won after this opening.
- From this position, Black lost their queen in three moves.
- …

"The path is long with rules,
but short with examples."

— Seneca

Chess-Playing AI

Classical AI

Search possible moves and choose the move that leads to the best position (i.e., worst position for the opponent).

Manually defined **rules** (or heuristics):

- A bishop is three times more valuable than a pawn.
- Knights are better positioned in the center.
- …

Machine Learning

Predict the best move based on the huge training data (i.e., experience).

Compiled **examples** from past games:

- In a grandmaster's match, White won after this opening.
- From this position, Black lost their queen in three moves.
- …

What Is Learning?

Learning a task $\Leftrightarrow$ Improving performance as experience increases

Task: An unknown function that we want to learn

$$f(x, y, z) = ?$$

Feature

Task: An unknown function that we want to learn

$$f(x, y, z) = ?$$

Experience: Input-output data examples for the function (i.e., training set)

$$f(2, 5, 2) = -6$$
$$f(2, 0, 3) = 6$$
$$f(-1, 1, 2) = -4$$

Training set

Task: An unknown function that we want to learn

f(x, y, z) = ?

Experience: Input-output data examples for the function (i.e., training set)

f(2, 5, 2) = -6
f(2, 0, 3) = 6
f (-1, 1, 2) = -4

Model

Task: An unknown function that we want to learn

f(x, y, z) = ?

Experience: Input-output data examples for the function (i.e., training set)

f(2, 5, 2) = -6
f(2, 0, 3) = 6
f (-1, 1, 2) = -4

Performance: Minimizing errors on the test set

f(4, 1, 7) = ?

Let's say we predict **f(4, 1, 7) = 28**

Assume the ground truth is **f(4, 1, 7) = 26**

Our **absolute error = 2**

Our **squared error = 4**

Test set

Task: An unknown function that we want to learn

$$f(x, y, z) = \boxed{xz - 2y}$$

Experience: Input-output data examples for the function (i.e., training set)

$$f(2, 5, 2) = -6$$
$$f(2, 0, 3) = 6$$
$$f(-1, 1, 2) = -4$$

Performance: Minimizing errors on the test set

$$f(4, 1, 7) = ?$$

Let's say we predict $f(4, 1, 7) = 28$

Assume the ground truth is $f(4, 1, 7) = 26$

Our **absolute error = 2**

Our **squared error = 4**

Test set

Task: An unknown function that we want to learn

$$f(x, y, z) = \boxed{xz - 2y}$$

Experience: Input-output data examples for the function (i.e., training set)

$$f(2, 5, 2) = -6$$
$$f(2, 0, 3) = 6$$
$$f(-1, 1, 2) = -4$$

Performance: Minimizing errors on the test set

$$f(4, 1, 7) = ?$$

Let's say we predict **$f(4, 1, 7) = 28$**

Assume the ground truth is **$f(4, 1, 7) = 26$**

Our **absolute error = 2**

Our **squared error = 4**

Stochastic functions

Task: An unknown function that we want to learn

$$f(x, y, z) = \boxed{xz - 2y}$$

Experience: Input-output data examples for the function (i.e., training set)

$$f(2, 5, 2) = -6$$
$$f(2, 0, 3) = 6$$
$$f(-1, 1, 2) = -4$$

Performance: Minimizing errors on the test set

$$f(4, 1, 7) = ?$$

Let's say we predict **f(4, 1, 7) = 28**

Assume the ground truth is **f(4, 1, 7) = 26**

Our **absolute error = 2**

Our **squared error = 4**

"No free lunch"

Task: Learning to add two numbers

add(x, y) = ?

Since this is a very important function, we have a special operator for this task:

x + y

Experience: Examples given in class (i.e., in the training set)

7 + 21 = 28

4 + 14 = 18

11 + 40 = 51

Performance: Scoring high on the exam (i.e., the test set)

6 + 32 = ?

17 + 2 = ?

Task: Learning to add two numbers

add(x, y) = ?

Since this is a very important function, we have a special operator for this task:

x + y

Experience: Examples given in class (i.e., in the training set)

7 + 21 = 28

4 + 14 = 18

11 + 40 = 51

Performance: Scoring high on the exam (i.e., the test set)

6 + 32 = ?

17 + 2 = ?

Overfitting

Task: Learning to add two numbers

add(x, y) = ?

Since this is a very important function, we have a special operator for this task:

x + y

Experience: Examples given in class (i.e., in the training set)

7 + 21 = 28

4 + 14 = 18

11 + 40 = 51

Performance: Scoring high on the exam (i.e., the test set)

6 + 32 = ?

17 + 2 = ?

Data augmentation

Task: Learning to add two numbers

add(x, y) = ?

Since this is a very important function, we have a special operator for this task:

x + y

Experience: Examples given in class (i.e., in the training set)

7 + 21 = 28

4 + 14 = 18

11 + 40 = 51

Performance: Scoring high on the exam (i.e., the test set)

6 + 32 = ?

17 + 2 = ?

Pretraining

Problem (Machine Translation)

translate(original_text, target_language) = translated_text

Once each character is represented as a number, it becomes a mathematical problem.

Examples

translate(“makine”, English) = “machine”

translate(“bir kedi”, English) = “a cat”

translate(“Benim adım Ersin.”, German) = “Mein Name ist Ersin.”

translate(“Düşünüyorum.”, English) = “I'm thinking.”

translate(“Bugün hava çok güzel.”, English) = “The weather is very nice today.”

⋮

Queries

translate(“makine öğrenmesi”, English) = ?

translate(“Nasılsın?”, German) = ?

⋮

Problem (Machine Translation)

translate(original_text, target_language) = translated_text

Once each character is represented as a number, it becomes a mathematical problem.

Examples

translate(“makine”, English) = “machine”

translate(“bir kedi”, English) = “a cat”

translate(“Benim adım Ersin.”, German) = “Mein Name ist Ersin.”

translate(“Düşünüyorum.”, English) = “I'm thinking.”

translate(“Bugün hava çok güzel.”, English) = “The weather is very nice today.”

⋮

Queries

translate(“makine öğrenmesi”, English) = ?

translate(“Nasılsın?”, German) = ?

⋮

Pretraining?

Problem (Machine Translation)

translate(original_text, target_language) = translated_text

Once each character is represented as a number, it becomes a mathematical problem.

Examples

translate(“makine”, English) = “machine”

translate(“bir kedi”, English) = “a cat”

translate(“Benim adım Ersin.”, German) = “Mein Name ist Ersin.”

translate(“Düşünüyorum.”, English) = “I'm thinking.”

translate(“Bugün hava çok güzel.”, English) = “The weather is very nice today.”

⋮

Queries

translate(“makine öğrenmesi”, English) = ?

translate(“Nasılsın?”, German) = ?

⋮

We won’t be doing natural language processing in this course.

Problem (Image Classification)

classify(photo) = object_in_the_photo

Once each pixel is represented as a number, it becomes a mathematical problem.

Examples

classify( **) = dog**

classify( **) = table**

...

Queries

classify( **) = ?**

⋮

Problem (Image Classification)

classify(photo) = object_in_the_photo

Once each pixel is represented as a number, it becomes a mathematical problem.

Examples

classify( **) = dog**

classify( **) = table**

...

Queries

classify( **) = ?**

⋮

Data augmentation?

Problem (Image Classification)

classify(photo) = object_in_the_photo

Once each pixel is represented as a number, it becomes a mathematical problem.

Examples

classify( **) = dog**

classify( **) = table**

...

Queries

classify( **) = ?**

We won't be doing computer vision in this course.

Problem (Tabular Data Classification)

titanic(name, age, gender, ticket_fare) = accident_outcome

Examples

titanic(“Braund, Mr. Owen Harris”, 22, Male, 7) = Died

titanic(“Heikkinen, Miss. Laina”, 26, Female, 7) = Survived

titanic(“Nasser, Mrs. Nicholas”, 14, Female, 30) = Survived

⋮

Queries

titanic (“Myles, Mr. Thomas Francis”, 62, Male, 9) = ?

⋮

Problem (Tabular Data Classification)

titanic(name, age, gender, ticket_fare) = accident_outcome

Examples

titanic(“Braund, Mr. Owen Harris”, 22, Male, 7) = Died

titanic(“Heikkinen, Miss. Laina”, 26, Female, 7) = Survived

titanic(“Nasser, Mrs. Nicholas”, 14, Female, 30) = Survived

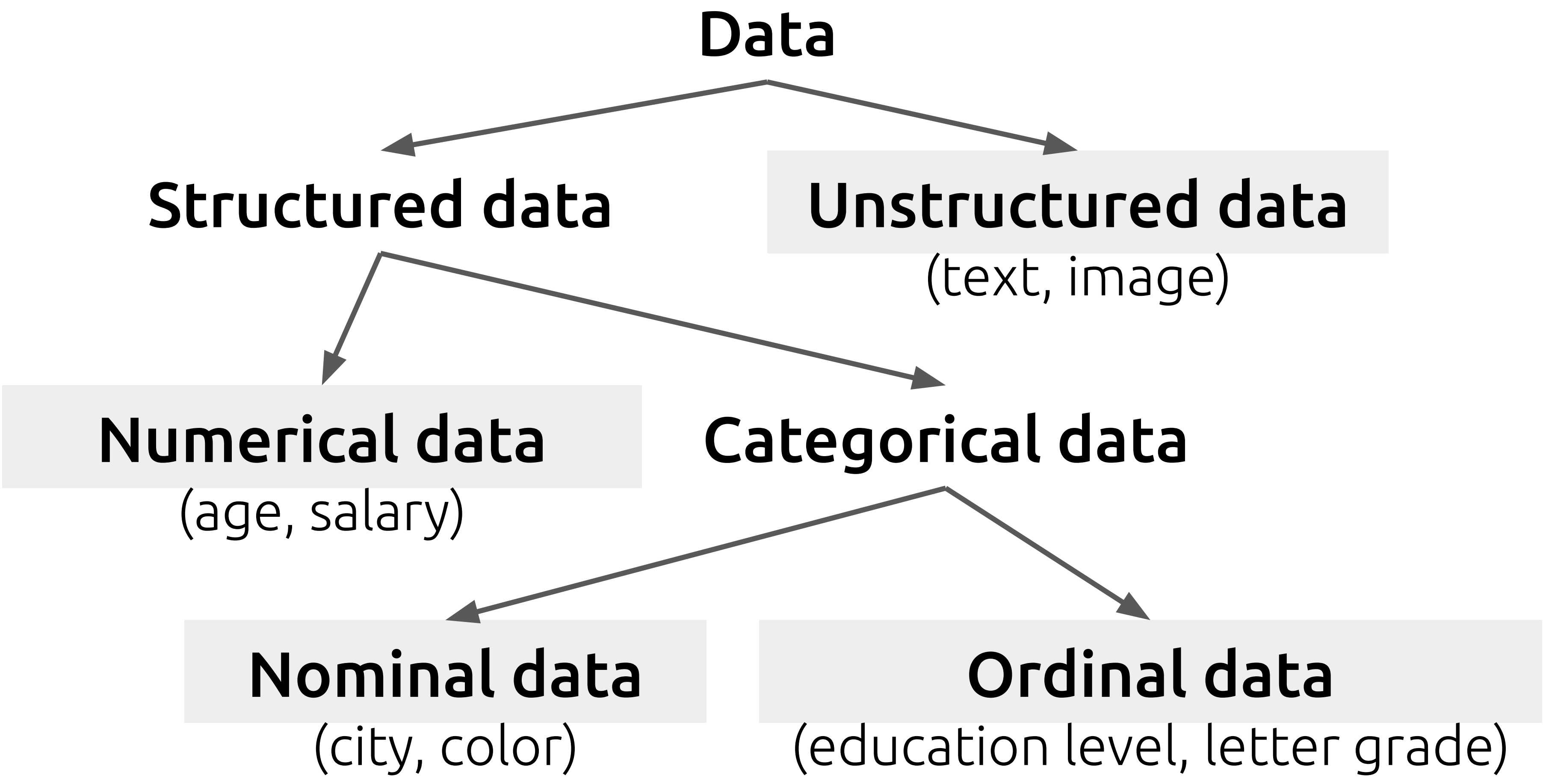
⋮

Queries

titanic (“Myles, Mr. Thomas Francis”, 62, Male, 9) = ?

⋮

Target variable


Data
Structured data
Unstructured data
(text, image)
Numerical data
(age, salary)
Categorical data
Nominal data
(city, color)
Ordinal data
(education level, letter grade)

Problem (Tabular Data Classification)

titanic(name, age, gender, ticket_fare) = accident_outcome

Examples

titanic(“Braund, Mr. Owen Harris”, 22, Male, 7) = Died

titanic(“Heikkinen, Miss. Laina”, 26, Female, 7) = Survived

titanic(“Nasser, Mrs. Nicholas”, 14, Female, 30) = Survived

⋮

Queries

titanic (“Myles, Mr. Thomas Francis”, 62, Male, 9) = ?

⋮

Problem (Tabular Data Classification)

titanic(name, age, gender, ticket_fare) = accident_outcome

Examples

titanic(“Braund, Mr. Owen Harris”, 22, Male, 7) = Died

titanic(“Heikkinen, Miss. Laina”, 26, Female, 7) = Survived

titanic(“Nasser, Mrs. Nicholas”, 14, Female, 30) = Survived

⋮

Queries

titanic (“Myles, Mr. Thomas Francis”, 62, Male, 9) = ?

⋮

Classification: Accuracy

Regression: Mean squared error

Problem (Tabular Data Classification)

titanic(name, age, gender, ticket_fare) = accident_outcome

Examples

titanic(“Braund, Mr. Owen Harris”, 22, Male, 7) = Died

titanic(“Heikkinen, Miss. Laina”, ?, Female, 7) = Survived

titanic(“Nasser, Mrs. Nicholas”, 14, ?, 30) = Survived

⋮

Queries

titanic (“Myles, Mr. Thomas Francis”, 62, Male, 9) = ?

⋮

Missing values

If the number of people in a photo is to be counted:
Detecting people can be learned.
(Then counting the detected "things" is easy.)

If the time on an analog clock is to be read from a photo:
Detecting the angles of the hour hand and minute hand can be learned.
(Once their angles are known, telling the time is easy.)

Proxy task

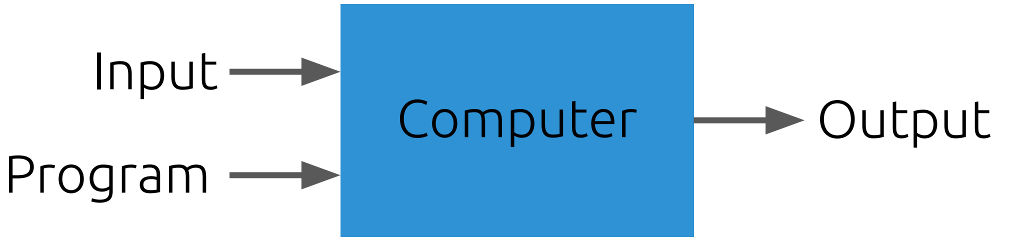
What move should White play?

Instead of learning the complex task of
“**finding the best move for White in a given position**,”
a simpler task can be learned:
“**Estimating White's probability of winning in a given position.**”
(We can perform the estimation for all possible positions and choose the move that leads to the best position.)

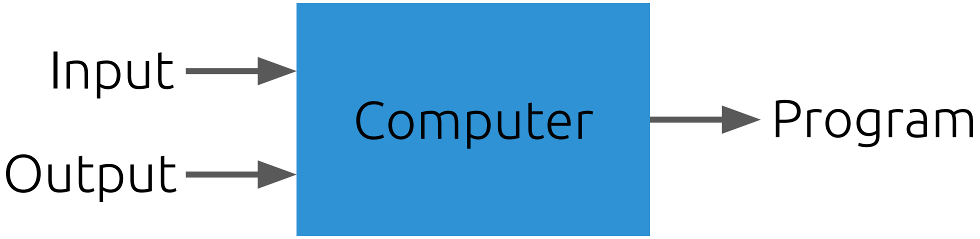
Proxy task

A Famous Illustration

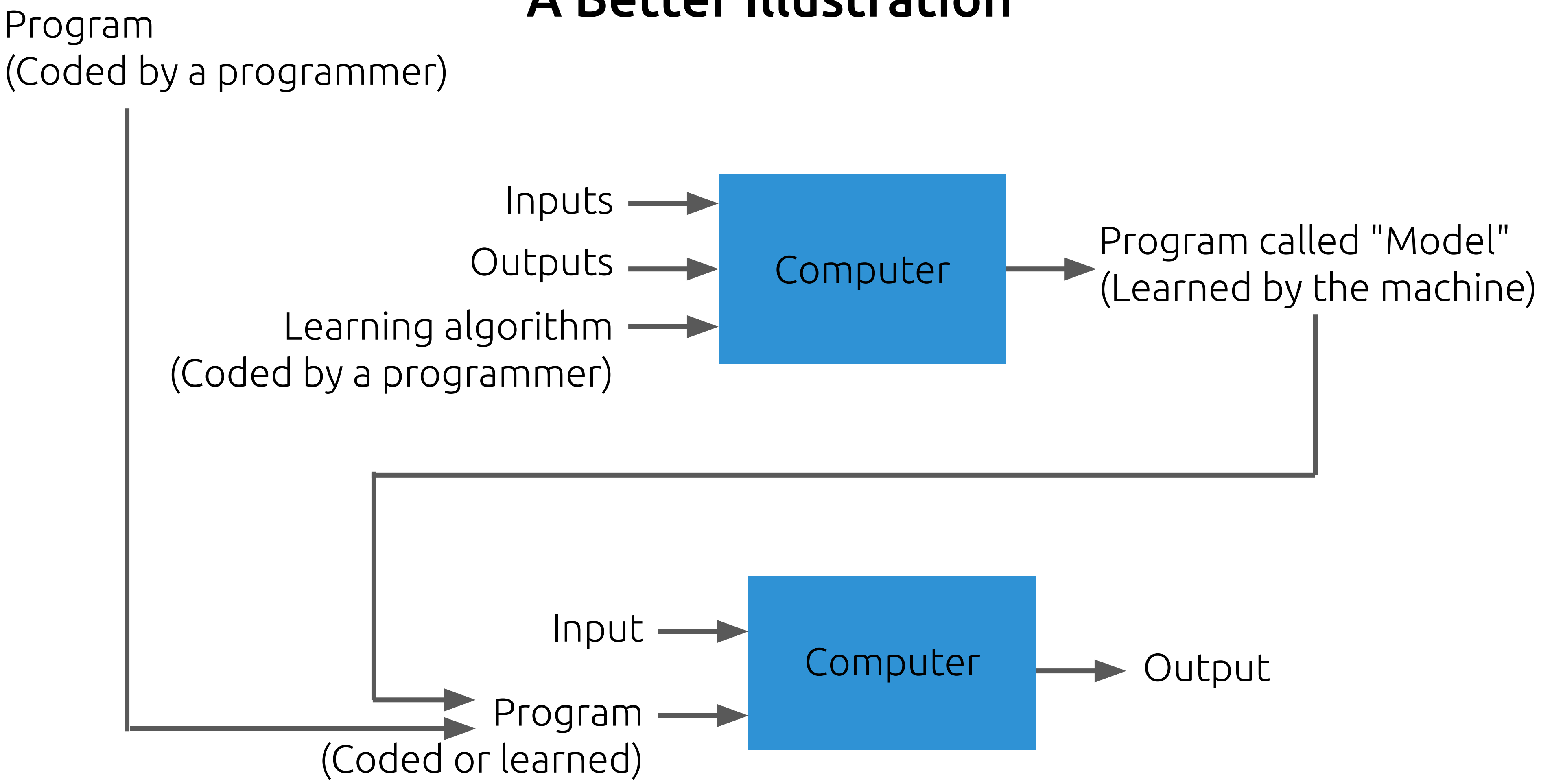
Software 1.0:
Traditional Programming



Software 2.0:
Machine Learning

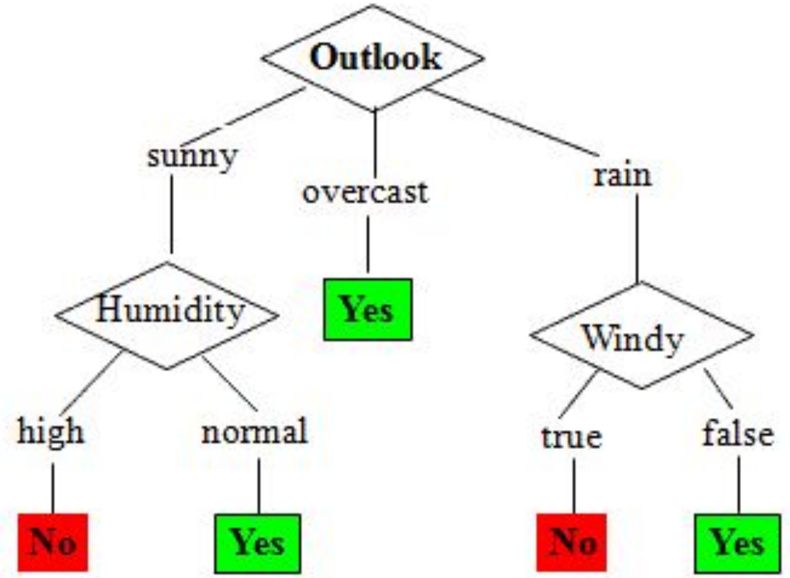


A Better Illustration



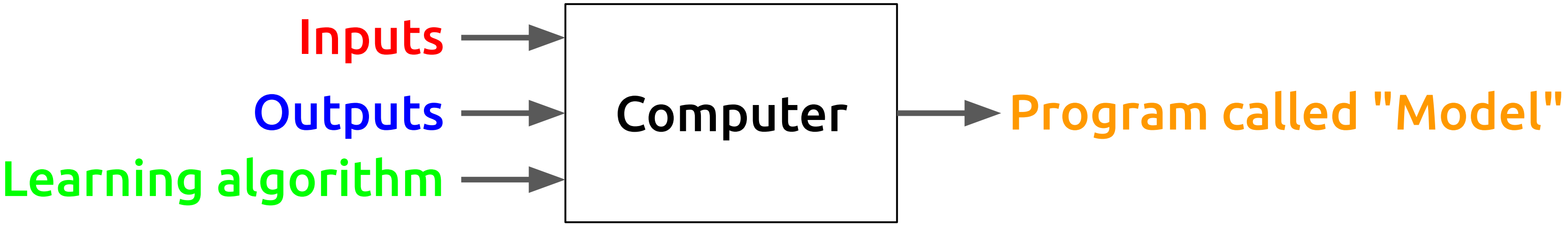
Model Example: Decision Tree

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No



Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	normal	false	?

Ensemble model



Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

ID3 - Algorithm

ID3 *(Examples, TargetAttribute, Attributes)*

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree Root, with label = most common value of *TargetAttribute* in *Examples*
- Otherwise Begin
  - A ← the attribute from *Attributes* that best classifies *Examples*
  - The decision attribute for *Root* ← A
  - For each possible value, vi, of A,
    - Add a new tree branch below *Root*, corresponding to the test A = vi
    - Let $Examples_{vi}$ be the subset of *Examples* that have value vi for A
    - If $Examples_{vi}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *TargetAttribute* in *Examples*
      - Else below this new branch add the subtree ID3($Examples_{vi}$, *TargetAttribute*, *Attributes* – {A})
- End
- Return *Root*