安徽理工大学
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

• 人工智能专业 学科基础教育必修模块

Python与机器学习

Python and Machine Learning

Chapter 1：Introduction

• Lecturer ：孟亦凡

• E-mail：myf@aust.edu.cn

AIMC LAB

0.0 Course Introduction

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

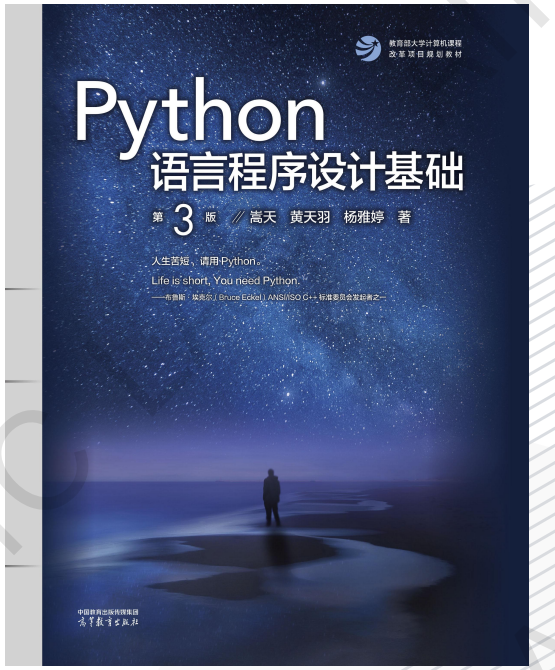
Python与机器学习

学时：48学时

成绩评定：

总评=考试（60%）+平时（40%）

平时=作业（40%）+表现（30%）+课堂（30%）



Python语言程序设计基础(第3版)
高等教育出版社，嵩天 等

机器学习
清华大学出版社，周志华

0.0 Course Discipline（课堂纪律）

安徽理工大学
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

1.不迟到早退，迟到5分钟考勤与课堂成绩扣50%；

2.上课不随意进出教室，随意离开教室扣课堂成绩50%；

3.迟到学生请直接入座第一排；

4.课堂主动回答问题，答对课堂成绩加10分/次；被动回答问题，答对加5分/次。

5.作业线上拍照提交，截止日期前必须提交，不会提供延期。未提交作业则本次作业不得分。

AIMC LAB

1 Introduciton

- What is it? —— What is machine learning?
- How to learn? —— How to master machine learning effectively?
- What to use? —— Why Python?
- Why is it? —— Why are Python and machine learning a "perfect match"?

AIMC LAB

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

CONTENTS

AIMC LAB

1.1 What is machine learning?

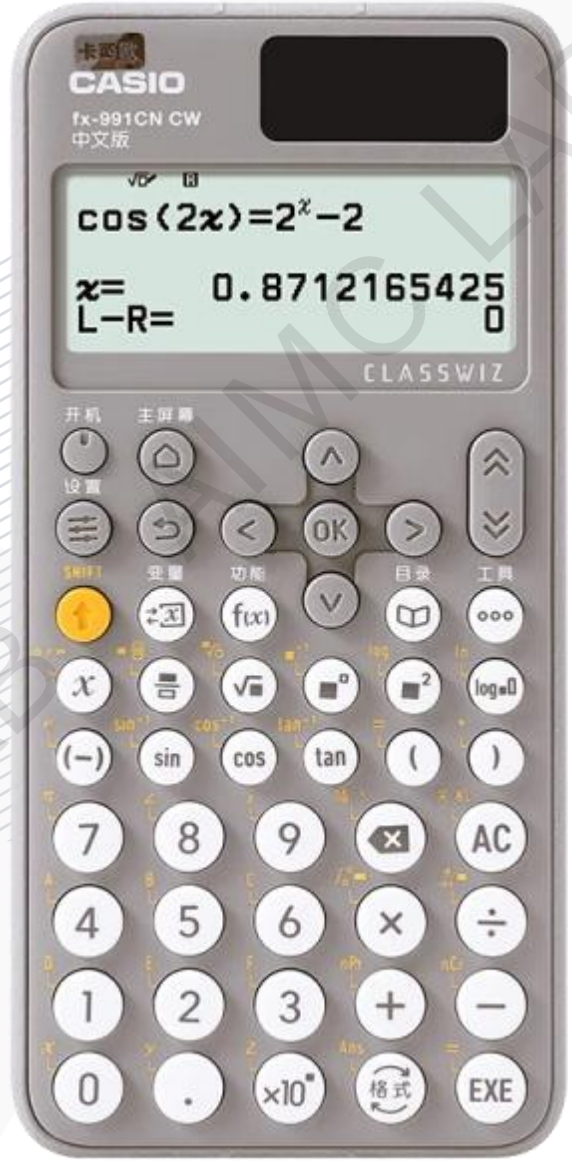
1.1.1 From Traditional Programming to Machine Learning

Traditional programming (e.g., C++):

Mode: Input +Process → Output

Core: The programmer needs to know and write all the rules in advance.

Example: Write a calculator program. You must precisely define the rules for +, -, *, / operations. Input 2 and 3, the rule is +, and the program outputs 5.



AIMC LAB

1.1 What is machine learning?

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Machine Learning：

-Mode: Input + Output → Rules (Models)

-Core: The computer automatically learns the implicit rules from a large number of "input-output" pairs. After learning, given a new input, it can predict the unknown output.

1.1 What is machine learning?

Example: Spam filter.

-Input: The contents of thousands of emails.

-Output: The label (spam or non-spam) of email.

-Learning process: The algorithm analyzes the relationship between the content of the email and the tags are likely to be spam.

-Result: Learn a model. When a new email arrives, the model can automatically determine whether it is spam or not.

1.2 How to Learn Machine Learning?

Learning Target：

- **Build intuition（建立直觉）: understand why algorithms work, not just how they work.**
- **Master the process（掌握流程）: master the complete Pipeline from data to model.**
- **Formation of thinking（形成思维）: the ability to view and solve real-world problems with machine learning thinking.**
- **Ability（具备能力）: Have the ability to implement, debug and optimize models.**

AIMC LAB

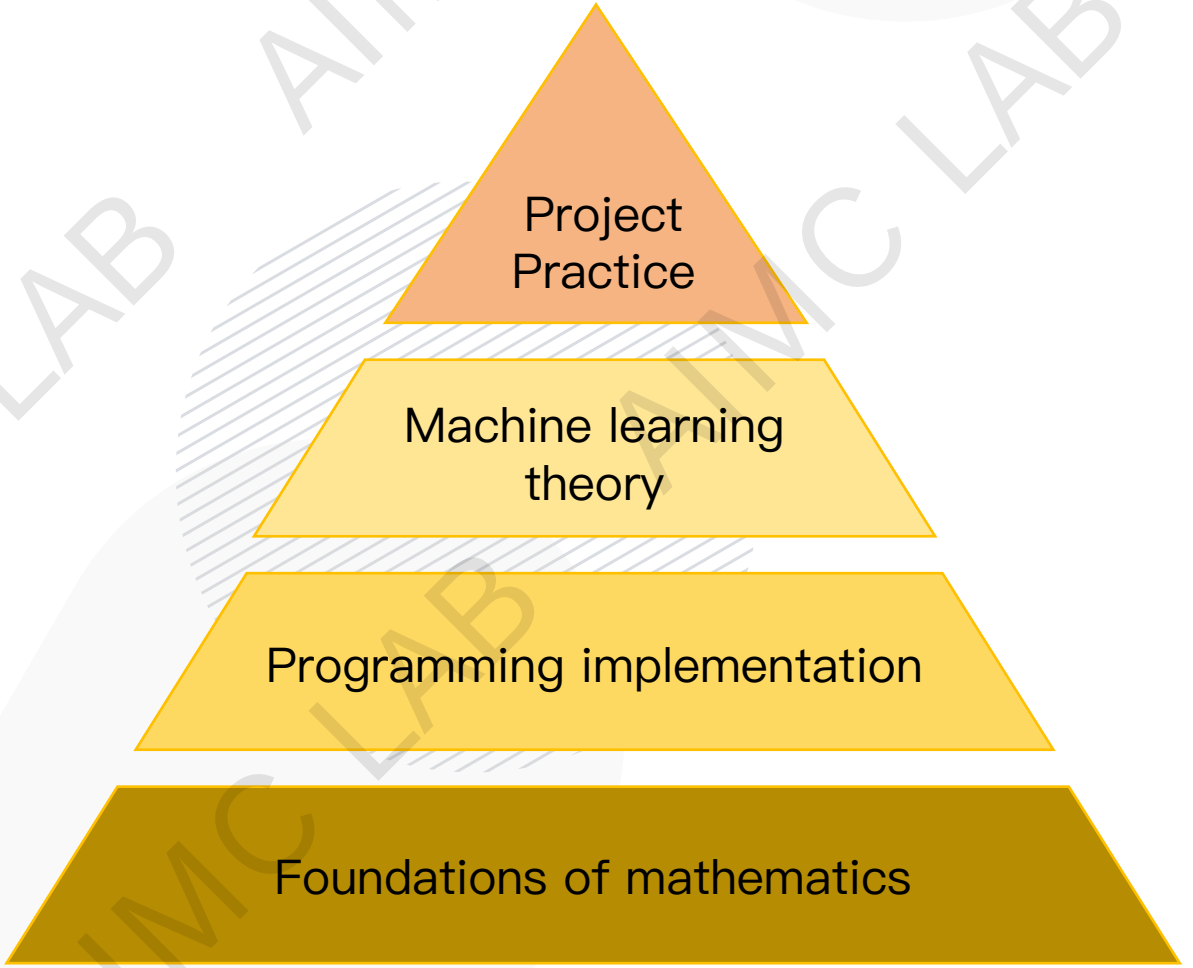
1.2 How to Learn Machine Learning?

1. Mathematical foundation

- **Linear Algebra（线性代数）**
- **Probability and Statistics（概率论与统计）**
- **Calculus（微积分）**

2. Programming： Python

- **Core Library：NumPy（计算基础），Pandas（数据处理），Matplotlib（可视化），Scikit-learn（传统ML）。**



AIMC LAB

1.3 What is Python?

A computer is a device that operates data according to instructions.

-Functionality（功能性）

The operation of data : Data calculation, Input & output processing, result storage.

-Programmability（可编程性）

Automatically, predictably and accurately perform the operator's intention according to a set of instructions.

AIMC LAB

1.3 What is Python?

What is a programming language?

A programming language is an artificial language used to interact (communicate)

- Programming languages are simpler（简单）, more rigorous（严谨） and more precise（精确） than natural languages.
- Programming languages are used to interact between humans and computers.

AIMC LAB

1.3 What is Python?

Oct 2025	Oct 2024	Change	Programming Language	Ratings	Change
1	1		Python	24.45%	+2.55%
2	4	↑	C	9.29%	+0.91%
3	2	↓	C++	8.84%	-2.77%
4	3	↓	Java	8.35%	-2.15%
5	5		C#	6.94%	+1.32%
6	6		JavaScript	3.41%	-0.13%
7	7		Visual Basic	3.22%	+0.87%
8	8		Go	1.92%	-0.10%
9	10	↑	Delphi/Object Pascal	1.86%	+0.19%
10	11	↓	SQL	1.77%	+0.13%

AIMC LAB

1.3 What is Python?

Our foundation: C++

- Powerful static type language
- Close to the hardware, excellent performance
- Fine memory control
- Suitable for system level development, game engine, high performance computing

Our goal: Machine learning

- Core: data, algorithms（算法）, rapid experimentation（快速实验）, model iteration（模型迭代）
- Requirements: rapid prototyping（原型设计）, powerful data processing power, rich library ecology（生态）

AIMC LAB

1.3 What is Python?

The question is: To "designing intelligent driving algorithms" (machine learning), we need more efficient（高效） tools.

What is the Chinese meaning of python?

蟒蛇

AIMC LAB

1.3 What is Python?

Python [`paiθ ə n]，译为“蟒蛇”

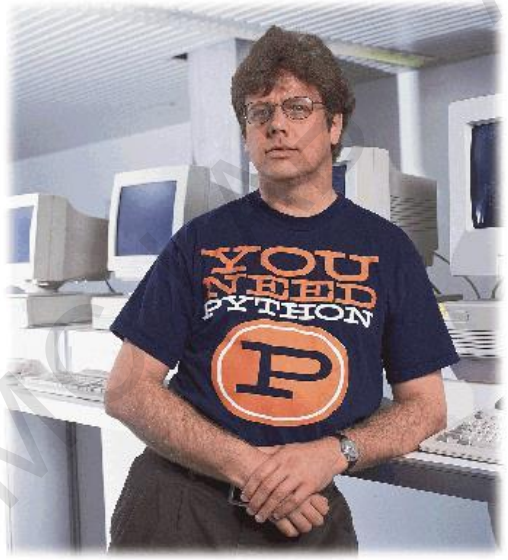
The Python language is owned by the Python Software.

PSF is a non-profit（非盈利） organization dedicated to protecting the open source and development of the Python language

AIMC LAB

1.3 What is Python?

- 2000.10, **Python 2.0** was officially released, solving many problems in the interpreter and runtime environment and ushering in an era of widespread Python application.
- 2008.12, **Python 3.0** has been released, with many significant improvements at the syntax level and within the interpreter.



Guido van Rossum

Python语言创立者

AIMC LAB

1.3 What is Python?

- **The cost of the change**: Python 3.x series code is not backward compatible with Python 2.x series code.

- In 2010, the final version of the Python 2.x series was released, Python 2.7, bringing the 2.x series of releases to an end.

AIMC LAB

1.3 What is Python?

Hello World

- C:

```
#include <stdio.h>
int main(void)
{
        print(“Hello World\n”);
        return 0;
}
```

- **Python:**

```
print(“Hello Word”)
```

AIMC LAB

1.3 What is Python?

python

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

The d

- Bea
- Cla
- Sim
- Fla
- Hig

Core

"

Comp

giving programmers great flexibility, but also increasing complexity.

1.3 What is Python?

python

安徽理工大学
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison I: Syntax & Readability

Feature	C++	Python
Statement End	Semicolon ;	Newline
Code Blocks	Curly Braces {}	Indentation (Mandatory, part of the syntax) 缩进（强制性的，是语法的一部分）
Variable Declaration	Explicit type required int a = 10;	Direct Assignment a = 10 (Dynamically Typed)
Example	if (x > 5) { cout << "Hello"; }	if x > 5: print("Hello")

AIMC LAB

1.3 What is Python?

python

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison I: Syntax & Readability

- Python syntax is closer to natural language, like writing pseudocode（伪代码）.
- Mandatory indentation enforces a uniform, highly readable style, crucial for frequently reviewed and modified ML projects.
- "Life is short, you need Python." — Spend less time on syntactic details, more on the problem itself.

AIMC LAB

1.3 What is Python?

python

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison II: Type System

Feature	C++	Python
Type Checking	Statically Typed (Compile-time)	Dynamically Typed (Runtime)

Variable Assignment	string s = "hello";
	s = 42; // Error!
		s = "hello" # s is now a str
		s = 42 # s is now an int, LEGAL!

Advantage	High performance, early error catching, safer.	Flexible, concise code, rapid development.
Disadvantage	Less flexible, compile-time overhead.	Runtime type errors possible, performance overhead.

AIMC LAB

1.3 What is Python?

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison II: Type System

- C++ is like a rigorous engineer; everything must be planned.
- Python is like an agile scientist; it encourages rapid experimentation.
- In the ML research and experimentation phase, flexibility is far more important than type safety.

AIMC LAB

1.3 What is Python?

python

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison III: Performance & The "Glue" Language

- **C++: Compiled to machine code, high performance. Suitable for computation-intensive cores (e.g., graphics rendering).**
- **Python: Interpreted, lower performance.**

AIMC LAB

1.3 What is Python?

python

安徽理工大學 ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison III: Performance & The "Glue" Language

But! Python is a "Glue Language"

- **The Key: Core Python libraries (e.g., NumPy) are implemented in C/C++ under the hood.**
- **Workflow（工作流）: You use Python's high-level, clean syntax to describe tasks; the heavy computation is done by efficient C/C++ code.**

AIMC LAB

1.3 What is Python?

python
安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison IV: Ecosystem（生态） & Libraries

- **C++ Ecosystem: Powerful, but fragmented, with a high barrier to entry.**
- **Python Ecosystem (Especially in Data Science & AI): Unbelievably rich, unified, and easy to use.**

AIMC LAB

1.3 What is Python?

python

安徽理工大學 ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

Core Comparison IV: Ecosystem & Libraries

Domain	Core Library	Purpose
Scientific Computing	NumPy	Foundational package for numerical computation with powerful n-dimensional arrays.
Data Manipulati		
Visualizati	Seaborn	zations.
Machine Learning	Scikit-learn	Library for traditional ML algorithms (classification, regression, clustering). Unified and clean API.
Deep Learning	TensorFlow PyTorch	Building and training neural network models. The mainstream tools for modern AI research.

In AI, libraries are productivity. Python has the most complete toolchain in this field.

AIMC LAB

1.4 Machine Learning and Python

Why Python is the #1 Choice for ML?

- **High Development Efficiency: Concise syntax and dynamic typing allow data scientists to quickly translate ideas into code.**
- **Powerful Ecosystem: Mature libraries support the entire pipeline, from data cleaning to model deployment, creating a virtuous cycle.**

AIMC LAB

1.4 Machine Learning and Python

Why Python is the #1 Choice for ML?

- **Vibrant Community: Solutions to almost any problem are available; new research is often implemented in Python first.**
- **The "Glue" Nature: Perfectly marries the high performance of C++ with the ease of use of Python.**

AIMC LAB

1.4 Machine Learning and Python

Why Python is the #1 Choice for ML?

C++: A Swiss Army Knife（瑞士军刀） – powerful, precise, you can build anything with it, but it requires training.

Python (in AI): LEGO Bricks（乐高积木） – provides high-performance, pre-built modules (libraries) that let you rapidly and flexibly build complex data products.

AIMC LAB

1.X 课程小结 / Course Summary

安徽理工大學
ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

本章核心要点回顾 / Key Points Review

1. 理解机器学习本质 / Understanding ML Essence

从规则编程到数据驱动范式转变

From rule-based programming to data-driven paradigm shift

2. 掌握学习方法论 / Mastering Learning Methodology

数学基础 + 编程实现 + 理论理解 + 项目实践

Math foundation + Programming + Theory + Project practice

AIMC LAB

1.X 课程小结 / Course Summary

3. 认识Python优势 / Recognizing Python Advantages

简洁语法、动态类型、丰富生态、胶水语言特性

Concise syntax, dynamic typing, rich ecosystem, glue language

4. 理解完美匹配原因 / Understanding Perfect Match Reasons

Python高效原型设计能力与ML快速实验需求高度契合

Python's rapid prototyping perfectly matches ML's experimental needs

AIMC LAB

1.X 课程小结 / Course Summary

学习路径建议 / Learning Path Suggestions

❶ 夯实基础 | Solidify Foundation

熟练掌握Python基本语法与数据类型

Master Python basic syntax and data types

❷ 工具熟悉 | Tool Familiarization

逐步掌握NumPy、Pandas、Matplotlib等核心库

Gradually master core libraries: NumPy, Pandas, Matplotlib

❸ 项目驱动 | Project-Driven

通过实际项目加深对机器学习流程的理解

Deepen understanding through practical projects

AIMC LAB

1.X 课程小结 / Course Summary

下章预告 / Next Chapter Preview

第2章：Python编程基础

思考题： 基于已有C++基础，您认为Python的哪些特性最能提升机器学习开发效率?

Question: Based on your C++ background, which Python features do you think can most improve ML development efficiency?

AIMC LAB