A study on the development and practice of programming courses based on financial intelligence thinking -An example of “Python in Finance’ course

Yanan Chen[1], Yi Peng[2*]

[1]School of Management, Guangzhou City University of Technology, Guangzhou, 510800, China
[2]School of Economics, Shenzhen Polytechnic University, Shenzhen, 518055, China
[*]Corresponding author: pengyi@szpu.edu.cn

***Abstract*：** *With the rapid development of information technology and the continuous change of economic environment, financial and business thinking has become one of the necessary core competences for modern finance and economics students. Taking the course ‘Python in Finance’ as an example, this paper discusses how to develop and practice programming courses in finance and economics majors to cultivate students' financial business thinking and programming skills. Through the innovation of curriculum design, teaching methodology and assessment mechanism, this paper aims to provide reference and reference for programming class courses in finance and economics majors.*

Keywords: *Financial Intelligence Thinking；Python in Finance；Programming Courses*

1. Introduction

In a rapidly changing economic environment, finance students face unprecedented challenges and opportunities. To adapt to these changes, they need to acquire not only solid financial knowledge, but also diverse skills such as data analysis, programming, and problem solving. This comprehensive set of skills is the core competitiveness of today's financial talent, and financial thinking is the key to building on this foundation. Financial literacy means not only understanding and applying financial knowledge, but more importantly, being able to flexibly use such knowledge and skills to solve practical economic problems and make informed decisions.

With the rapid development of big data and artificial intelligence technology, data has become the "oil" of the new era, and how to efficiently mine and utilize it has become the key to success for companies and individuals. In this context, programming skills have become an important and indispensable skill for modern finance and economics students [1]. Through programming, students can process and analyze large data sets, extract valuable information, and provide a scientific basis for financial decision-making[2-3]. For example, analyzing financial data using Python or R can help investors identify market trends and optimize asset allocation; using machine learning algorithms to predict credit risk can improve the risk management level of financial institutions[4]. Therefore, it has become particularly urgent to develop and practice programming courses, especially those that can closely integrate programming skills with financial knowledge. Such courses aim to develop students' ability to apply programming to solve financial problems in real work scenarios[5-6]. For example, through project-based learning, students can participate in actual financial data analysis projects, from data collection and cleaning to modeling and analysis, and finally producing reports or solutions.

2. The Importance of Financial Thinking in the Finance and Economics Profession

Financial thinking refers to the ability to apply financial knowledge and skills to solve practical economic problems. In the fast-changing economic environment, this ability is not only crucial for personal finance management, but also one of the core competencies necessary for modern finance and economics majors. Financial thinking can help individuals better understand economic phenomena, make sound financial decisions, and effectively manage personal or organizational finances. The following are four important aspects of financial literacy thinking and its specific applications in finance and economics majors:

Data analysis and decision-making: In modern financial management, data analysis has become increasingly important. Financial data is not only huge in quantity, but also varied, including financial statements, market data, customer data and so on. Financial thinking can help students process and analyze these data efficiently so as to make scientific and reasonable decisions. For example, by analyzing historical financial data, future financial trends can be predicted to provide a basis for the strategic planning of an enterprise[5]. In addition, data analysis can help identify potential risk points and take preventive measures in advance to reduce financial risks. By using programming tools such as Python, students can perform data cleaning, preprocessing and visualization more efficiently to better understand and interpret financial data.

Automation and optimization: With the development of information technology, automation tools are increasingly used in financial management. Financial thinking not only requires students to have basic financial knowledge, but also certain programming skills in order to write automation scripts and tools to improve the efficiency of financial work. For example, by writing Python scripts, financial data can be automatically obtained from multiple data sources, data cleaning and pre-processing can be performed to generate financial reports. This not only saves a lot of labor and time, but also reduces human errors and improves data accuracy and reliability. In addition, automation tools can be used for daily financial tasks such as bill processing and invoice reconciliation to further improve efficiency.

Risk management: Risk management is an important part of financial management. Financial thinking can help students identify and manage financial risks and develop effective risk management strategies. For example, by analyzing market data, potential market risks can be identified and investment portfolios can be adjusted in time to reduce losses. In addition, financial thinking also requires students to have a sense of crisis and be able to react quickly when unexpected events occur and take appropriate countermeasures to protect the financial security of the enterprise. By learning the theories and methods of risk management, students can better cope with a variety of financial risks and provide enterprises with more robust financial support.

Innovative ability: In a competitive market environment, the ability to innovate is the key to maintaining the competitive advantage of enterprises. Financial thinking not only requires students to have solid financial knowledge, but also innovative thinking and problem-solving skills. Through the learning of programming skills, students can develop new financial tools and applications to improve their innovation and competitiveness. For example, students can develop a financial management system to help companies manage financial data, financial analysis and decision-making. In addition, students can develop small programs to solve specific financial problems and improve work efficiency. Through innovative practice, students can apply what they have learned to practical problems and develop the ability to solve complex problems.

In conclusion, the importance of financial thinking in finance and economics majors cannot be ignored. By cultivating students' financial thinking, we can not only improve their financial knowledge and skills, but also cultivate their innovative thinking and problem-solving ability, laying a solid foundation for their future career development.

3. Curriculum design

3.1 Course objectives

The core objective of the Python in Finance course is to develop students' financial thinking and programming skills to achieve a high degree of integration between the two. Through a systematic teaching approach, the course aims to enable students to learn and master the basic syntax of the Python programming language as well as commonly used library functions. On this basis, students will be able to use Python to efficiently collect, clean and analyse financial data in order to write automation scripts that can improve the efficiency of financial work. The course also covers the construction of simple financial models to support forecasting and decision making, developing students' computational thinking skills and ability to solve real-world financial problems. Through this series of well-designed teaching sessions, students will be able to realise the perfect combination of theory and practice in the field of finance.

3.2 Course content

3.2.1 Python Basics and Data Processing

This module aims to provide an in-depth exposition of the core concepts and syntactic structures of the Python programming language, covering key elements such as variable definitions, data type classification, control flow structures, function construction and modular programming. Within this framework, students will learn in depth the basic operating techniques of NumPy and Pandas, two libraries widely used in the field of data processing, in order to master a systematic approach to reading, cleaning, pre-processing and transforming data. In addition, the module will explain in detail the strategies of data normalisation, the techniques of normalisation processing and the practical application of coding methods. Meanwhile, students will learn how to present data information intuitively using two powerful visualisation libraries, Matplotlib and Seaborn. Through the systematic study of this module, students will not only build a solid foundation in Python programming, but will also be able to efficiently process and analyse financial data, laying a solid foundation for subsequent in-depth research.

3.2.2 Financial data acquisition and analysis

The core teaching content of this module aims to provide an in-depth exploration of the collection of financial data and its methodology. Students will systematically learn and master various common sources of financial data, including but not limited to stock market data, corporate financial statements and macroeconomic data. In addition, the module will focus on how to efficiently acquire real-time financial data from financial websites using advanced API interface technologies. Further, students will learn how to store these captured data in standardised formats in CSV files, Excel tables and databases. Through the in-depth study and practice of this module, students will be able to skillfully use various means to collect financial data from different channels, and have the ability to conduct preliminary data processing and analysis.

3.2.3 Automated scripting and financial modeling

This module aims to provide an in-depth look at the use of Python scripting in the area of financial automation, detailing how to automate financial workflows through programming. In the process, students will acquire the core skills of writing Python scripts that are able to extract financial data from diverse data sources on a regular basis and store and manage it efficiently. In addition, students will learn to utilise Python's timed task scheduling tools, such as APScheduler, in order to automate the timed execution of tasks to enhance productivity. Further, this module will provide a detailed introduction to building and applying common financial models, including but not limited to cash flow models, capital budgeting models, and portfolio optimisation models. Students will learn how to use these models for financial forecasting and decision support, providing a scientific basis for practical problem solving. Through the systematic learning of this module, students will not only be able to significantly improve the efficiency and accuracy of financial work, but also be able to construct and apply financial models to meet the challenges and needs of practical work.

3.2.4 Project practice and integrated application

This module aims to deepen students' mastery and application of professional knowledge and skills through the implementation of practical projects. During the process, students will be divided into groups, each group needs to select a real-life problem related to finance for investigation, such as stock price forecasting, financial risk assessment, portfolio optimisation and other topics. Under the careful guidance of the teacher, students will carry out a series of practical activities, including data collection, processing, model construction and comprehensive analysis of the results. On this basis, students will collaborate to complete a project presentation and review their writing code. During this session, teachers and students will provide valuable feedback and constructive comments. Through the learning and practice of this module, students can not only successfully apply theoretical knowledge to practical problem solving, but also effectively improve their problem solving ability. At the same time, the module also focuses on cultivating students' teamwork and communication skills, with a view to demonstrating more comprehensive professionalism in their future careers.

Through in-depth study of the above four well-designed modules, students will be able to enhance their understanding and mastery of Python in finance. They will not only be able to use Python to perform complex data analysis and accurate model building, but also be able to integrate programming skills with deep knowledge of financial theory, thus demonstrating their outstanding ability to solve real financial problems. This process will significantly enhance students' financial thinking and give them a competitive edge in the workplace, adding an indispensable value-add to their careers.

4. Teaching practice and evaluation of effectiveness

4.1 Preparation for teaching

To ensure that the course "Python in Finance" runs smoothly, preparation for teaching is crucial. First, the instructor is trained in Python programming and financial knowledge to ensure that the instructor has sufficient teaching ability. The training includes the basic syntax of Python, the use of common libraries, financial data analysis and model construction. Second, prepare rich teaching resources, including textbooks, courseware, video tutorials and practice questions. The textbook chooses Python programming books suitable for finance and economics majors, such as Python for Data Analysis and Python for Finance. The courseware is produced in detail, covering the main content and key knowledge points of each module. Video tutorials are used to explain the difficulties and key points in the course to help students better understand and master the knowledge. Finally, a stable experimental environment is built to provide the software and data resources needed by students. The experimental environment includes the installation of Python and its commonly used libraries, rich financial data resources, and online programming platforms such as Jupyter Notebook or Google Colab, which ensures that students are able to perform programming exercises and project development locally or in the cloud.

4.2 Teaching and learning implementation

Teaching and learning delivery is at the centre of many aspects of the success of the programme. Each class is carefully designed to cover both theoretical and practical components to ensure that students are able to effectively translate their theoretical knowledge into real-world problem solving skills. In the theoretical part of the class, the teacher will focus on the basic concepts and advanced methods of Python programming and financial analysis, aiming to build a solid theoretical foundation for students. At the same time, students will combine theoretical knowledge with practice through programming exercises in the classroom, and the teacher will provide on-site guidance and Q&A in the process to solve the problems encountered by students in practice. In addition, in order to promote active participation and teamwork, the course adopts the form of group discussion, allowing students to discuss real cases in groups and work together to find solutions to problems, so as to cultivate their teamwork skills in the process of problem solving. The course further strengthens students' practical skills with several mini-projects and a comprehensive large-scale project. The mini-projects mainly involve basic data analysis tasks, while the large-scale project requires students to apply their knowledge and skills to solve more complex real-world problems. Upon completion of the projects, students were required to work in groups to present their projects and review the code, and the teacher and their classmates would give feedback and suggestions based on the completion of the projects. Through this series of practical activities, students can not only consolidate and deepen the knowledge they have learnt, but also significantly improve their ability to solve practical problems, laying a solid foundation for future career challenges.

4.3 Student evaluation

In order to assess the effectiveness of students‘ learning, the programme adopts a diversified assessment system, which aims to ensure the comprehensiveness and objectivity of the assessment results, so as to accurately reflect students’ learning outcomes. Weekday grades include not only students' performance in class, but also the quality of their homework and their active participation in group discussions. Teachers will grade students based on their participation in class, the quality of their assignments and their performance in discussions. As for the project grade, it is mainly based on the overall quality of project completion and the innovativeness it reflects. Teachers may organise project debriefing sessions to provide a platform for students to present their project results, as well as code review and feedback to promote in-depth understanding and skill enhancement. The final exam consists of a theoretical knowledge test and a practical programming test. The former is designed to test students' understanding and mastery of the course knowledge points, while the latter requires students to complete an actual programming task within a specified period of time in order to assess their practical programming ability. Through this series of fine-tuned evaluation tools, teachers are able to fully grasp the learning status of students, identify potential problems in time and take corresponding measures to continuously optimise teaching strategies and course content.

To sum up, through the systematic teaching preparation, efficient teaching implementation and scientific and rigorous student evaluation system of the course ‘Python in Finance’, this course not only significantly promotes the cultivation of students‘ financial thinking and programming skills, but also significantly enhances students’ practical ability and competitive advantage in the professional market. Looking ahead, we aim to continue to refine and optimise the course design and teaching methodology, in order to further improve the quality and effectiveness of the course and lay a solid foundation for the cultivation of composite talents who can meet the needs of the times.

5. Teaching Effect and Reflection

Through the teaching practice of the course "Python in Finance", students not only mastered the basic syntax of Python programming and commonly used libraries, such as NumPy, Pandas, Matplotlib, etc., but also were able to proficiently collect, clean and analyze financial data. They learned how to acquire data from multiple data sources, handle missing values and outliers, perform data preprocessing, and use visualization tools to present analysis results. In addition, students were able to write automation scripts to improve the efficiency of financial work, such as automating the generation of financial reports and batch processing of transaction data. More importantly, they are able to construct simple financial models for forecasting and decision support, such as predicting sales using linear regression and assessing credit risk using the random forest algorithm. Through the practice of real projects, students' practical ability and problem solving ability have been significantly improved. Many students said that the course was rich in content and practicality, which greatly enhanced their financial thinking and professional competitiveness. For example, one student successfully developed an automated stock trading system at the end of the course, while another used the knowledge gained to help a family business optimize its inventory management strategy. These achievements not only demonstrate the students' learning outcomes, but also reflect the practical value of the course.

However, we also identified some areas for improvement. Firstly, some students found it difficult to understand and apply programming concepts when they first started to engage in programming, especially in understanding and applying programming concepts. This group of students needs more guidance and support, including the provision of additional tutorial materials and the conduct of small-group tutorial sessions. Secondly, the difficulty of certain program tasks needs to be further adjusted to suit students of different levels. The current set of tasks may be too complicated for beginners and not challenging enough for students with a certain level of foundation. In the future, we will design the program tasks in a graded manner based on students' feedback and performance to ensure that each student can find the right practice content for him/herself.

6. Conclusion

The course ‘Python in Finance’ effectively promotes the overall development of students' financial thinking and programming skills through its systematic curriculum design, unique teaching strategies and precise scientific assessment system. The rich content of the course not only systematically teaches the basic theories of Python programming, including the core concepts of variables, functions, loops and conditional statements, but also provides in-depth explanations of programming libraries that are widely used in the financial field, such as NumPy, Pandas and Matplotlib. Through this course, students will not only be able to master these fundamental knowledge, but also be able to apply their programming skills flexibly in solving real financial problems, thus significantly enhancing their practical skills and competitiveness in the workplace.

In terms of teaching methods, we have adopted a variety of innovative means, such as the flipped classroom, project-driven learning and group work. The flipped classroom allows students to learn the basics on their own through video tutorials before class, and then focuses on discussing and solving real-world problems in the classroom, which greatly improves learning efficiency. Project-driven learning allows students to consolidate what they have learnt and improve their ability to solve complex problems through real cases and project tasks. Group work not only promotes communication and co-operation among students, but also fosters the spirit of teamwork. A scientific assessment mechanism is also one of the keys to the success of the programme. In addition to traditional exams and assignments, we have set up several practical projects, requiring students to submit complete project reports and codes to assess their comprehensive application skills. In addition, regular classroom quizzes and group discussion records are also included in the final grade to comprehensively examine students' knowledge mastery and practical skills. In the future, we will continue to optimise the course content and teaching methods to further enhance the teaching effectiveness of the course. Specific measures include introducing more up-to-date financial datasets and industry cases so that students can be exposed to the most cutting-edge technologies and application scenarios; and designing more open-ended project tasks to encourage students to give full play to their creativity in solving practical financial challenges. At the same time, we will strengthen the cooperation with enterprises and introduce more real-world projects to improve the practicality and attractiveness of the course.

In conclusion, the course ‘Python in Finance’ has achieved remarkable teaching results and provided valuable programming skills and practical experience for finance and economics students. In the future, we will continue to work hard to provide students with more high-quality programming courses to help them achieve greater success in their careers. References

Fund Project

2023 Guangdong Provincial Education Science Planning Project (Higher Education Special)"Research on the Mechanism of Higher Education Empowering the Achievement of Common Prosperity from the Perspective of Industrial Structure Upgrading" (2023GXJK634).

Guangdong Province Quality Engineering Vocational Education Teaching Reform Research and Practice Project"Research on the Development and Practice of Programming Design Courses for Higher Vocational Finance and Economics Majors Based on Computational Thinking Cultivation in case of　Application of Python in Finance" in 2023(2023JG333).

2024 Guangzhou City University of Technology campus level scientific research project (57-K0224033）

References

[1] Chen Danlei. Research on the teaching process of Python programming course in the context of new engineering[J]. Knowledge Window(Teacher Edition),2024,(09):89-91.
[2] Zhao Xiyang. Application of Python in financial big data[J]. Journal of Anhui Police Officers Vocational College,2024,23(01):124-128.
[3] Wang Xian. Research on the teaching reform of financial big data foundation based on Python[J]. Liaoning Higher Vocational Journal,2024,26(04):55-58.
[4] Lu Yonglai. Research on the construction and application of teaching case base for the course ‘Python Data Analysis’[J]. Office automation,2024,29(19):41-44.
[5] Liu Yang,LI Wei,ZHANG Yangfan,et al.Python in financial data mining and analysis[J]. Electronic Technology,2023,52(11):274-275.
[6] Rudy, Li Su, Liu Shengzhi, et al. Application and practice of project-based teaching in the course of ‘Python language programming’[J]. Science and Technology Wind,2024,(27):120-122.