Course Reform and Practice of "Python Programming" for Artificial Intelligence and Big Data Applications

Qian Liu* Jiejuan Guo

Yinchuan University of Science and Technology, Yinchuan, 750000, China
**Corresponding author: liuqian95.91@163.com*

Abstract: *With the rapid development of artificial intelligence (AI) and big data technologies, Python has become a widely used programming language in related fields. However, the traditional "Python Programming" course has been unable to meet the demands of applications in AI and big data in terms of content and teaching models. Therefore, this paper presents a comprehensive reform of the "Python Programming" course, optimizing course content by incorporating modules on big data processing and artificial intelligence. Additionally, a project-driven teaching model and a diversified evaluation system have been implemented. By introducing innovative teaching tools and resources, students' practical abilities and innovative thinking have been enhanced. The practical outcomes of the course reform are significant, with students showing notable improvements in their programming skills in AI and big data. In the future, this study will continue to optimize course content to address the ever-changing technological demands.*

Keywords: *Python programming; artificial intelligence; big data; course reform; project-driven teaching; practical effects*

Introduction

With the rapid advancement of artificial intelligence and big data technologies, the demand for technical personnel with data analysis and AI development capabilities continues to grow. As one of the most influential programming languages in current AI and big data applications, Python's teaching is particularly important. However, the traditional "Python Programming" course primarily covers basic syntax and programming techniques, lacking the core technologies related to big data and AI, and has insufficient practical and application-oriented elements. Moreover, the teaching methods predominantly rely on traditional lectures, with a lack of project-based practice and innovation capability development. This study aims to reform the course by optimizing the structure of the "Python Programming" course to enhance its practicality and relevance, especially in applications related to AI and big data. The necessity of the course reform lies in cultivating students' comprehensive abilities, as the reformed course will better equip students with the skills needed to tackle new technological challenges, enhancing their competitiveness in fields such as data processing, machine learning, and deep learning.

1. Current Status of the "Python Programming" Course

1.1 Limitations in Course Content

The current "Python Programming" courses predominantly focus on basic syntax and fundamental programming concepts, primarily covering Python's basic data types, control structures, functions, and modules. While this content helps students master foundational programming skills, it lacks depth when addressing the complex applications of artificial intelligence and big data. The curriculum does not adequately cover cutting-edge technologies such as big data processing, machine learning algorithms, and deep learning frameworks, making it difficult for students to apply Python to modern data-intensive projects after completing the course. Additionally, many courses overlook the integration of Python with data science libraries (e.g., NumPy, Pandas, Matplotlib), failing to fully explore Python's potential in data analysis and visualization, which restricts students' understanding and application abilities in data-driven projects. [1]

1.2 Traditional Teaching Methods

Most current "Python Programming" courses adopt traditional lecturing methods, where instructors primarily provide theoretical explanations and code demonstrations while students follow along to write basic programs. This teaching model, centered on one-way information transfer, struggles to stimulate students' active learning and innovation capabilities. With relatively few practical components, the knowledge students gain in class is not adequately applied in real-world contexts, leaving their learning superficial. Furthermore, the courses often lack project-based practical teaching, resulting in students having limited opportunities to use Python programming to solve complex real-world problems. This approach not only fails to enhance students' programming thinking but also restricts their in-depth exploration of frontier technologies such as big data and artificial intelligence.

1.3 Insufficient Development of Student Abilities

Due to the limitations in course content and traditional teaching methods, the current "Python Programming" courses struggle to effectively cultivate students' comprehensive abilities to tackle challenges in big data and artificial intelligence. First, students mainly engage with basic programming techniques, lacking practical experience in handling complex data processing tasks, making them ill-equipped for projects that require working with large datasets. Second, the evaluation systems often emphasize final exams or single programming tasks, neglecting to assess students' problem-solving skills, innovative thinking, and teamwork abilities. As a result, students may grasp Python's basic syntax but lack sufficient flexibility and creativity in practical applications, particularly in areas like big data analysis and machine learning model development. Additionally, the lack of interdisciplinary content hinders students' ability to integrate Python programming with other fields, such as statistics, data science, and artificial intelligence, further diminishing their competitiveness in the future job market. [2]

2. Specific Measures for Reforming the "Python Programming" Course for AI and Big Data Applications

2.1 Optimizing Course Content and Adding AI and Big Data Modules

To enhance the practicality and foresight of the "Python Programming" course, the optimization of course content should focus on big data processing and AI applications. First, additional modules on Python's applications in the big data field should be introduced, covering data preprocessing, data cleaning, data visualization, and management and analysis of large datasets. The use of specific technologies such as the Pandas, NumPy, and Matplotlib data science libraries will provide students with foundational skills for handling complex data tasks. These tools can process structured data as well as semi-structured and unstructured data, such as JSON and XML files. Furthermore, the curriculum should address common challenges in data analysis, such as handling missing values and detecting outliers, helping students become familiar with routine operations and optimization strategies for complex datasets.

Secondly, regarding the AI field, content related to machine learning and deep learning should be introduced, covering basic algorithms (such as regression, classification, and clustering) and the applications of deep learning frameworks (like TensorFlow, Keras, and PyTorch). By showcasing practical case studies on model training, parameter tuning, and model evaluation, students can gain an in-depth understanding of algorithm principles and their applications in various scenarios. With the addition of these modules, students will master key applications of Python in AI and big data, laying a solid foundation for real-world industry applications. [3]

Moreover, by incorporating specific industry cases—such as Python applications in financial data analysis, image recognition, and natural language processing—the practical relevance of the course can be further enhanced. For instance, in the financial sector, students could learn how to utilize Python for risk analysis and forecasting; in image recognition, students could build image classification models; while in natural language processing, the course could introduce practical applications like text classification and sentiment analysis. This industry-driven case teaching approach will equip students with valuable programming skills, enhance their ability to tackle complex data challenges, and provide strong support for their future career development.

2.2 Adopting a Project-Driven Teaching Model to Enhance Practicality

To address the lack of practical application in traditional teaching methods, the course reform should adopt a project-driven teaching model that organically integrates theoretical instruction with practical application. This method requires students to apply Python programming in real datasets or actual problem scenarios, cultivating their problem-solving abilities and teamwork awareness. Through this approach, students can not only grasp theoretical knowledge but also enhance their practical skills through hands-on experience. For example, students could engage in big data analysis projects that allow them to experience the complete data processing workflow, from data acquisition and cleaning to analysis and visualization.

Additionally, students could work on AI projects, such as implementing simple machine learning or neural network models, allowing them to practice AI algorithms and understand the model training and optimization process. These projects could encompass common machine learning tasks such as regression, classification, and clustering, enabling students to master core AI concepts by adjusting algorithm parameters and evaluating model performance. To further enhance practicality, instructors could introduce more complex AI application projects, such as real-world problems in natural language processing (NLP) or computer vision, helping students flexibly apply Python programming skills in challenging scenarios.

This project-driven teaching model not only enhances students' practical application capabilities but also increases their learning initiative and innovative thinking. During the project, instructors can guide students in exploratory learning, allowing them to identify problems through hands-on practice and propose solutions, thus mastering Python programming skills and AI techniques in real-world contexts. Instructors should focus on stimulating students' innovative awareness, encouraging them to propose unique solutions beyond conventional tasks.

2.3 Constructing a Diversified Teaching Evaluation System

Traditional evaluation systems typically focus on exam scores, which fail to comprehensively reflect students' overall abilities in programming practice, innovative thinking, and teamwork. Therefore, course reform needs to establish a diversified teaching evaluation system that encompasses multiple dimensions, including formative assessment, project outcomes, and team collaboration performance. This evaluation system not only pays attention to students' performance in theoretical exams but also emphasizes their practical abilities, innovative capabilities, and problem-solving skills in actual programming and projects. Formative assessments can be conducted by recording students' performance, participation, and proactive problem-solving in projects to comprehensively evaluate their learning progress and practical programming skills. Instructors can assess students' engagement in classroom discussions, contributions to project designs, and strategies for addressing challenges to evaluate their progress and application abilities comprehensively. [4]

Moreover, the quality of project outcomes should become a crucial evaluation criterion. Instructors can grade students based on several aspects, including the effectiveness of project implementation, code quality, data processing accuracy, and model performance. For instance, in data analysis projects, instructors can examine how students clean and process data, whether they can appropriately apply Python libraries to complete tasks, and the clarity and effectiveness of the final data visualizations. In AI projects, factors such as model accuracy, optimization of training time, algorithm selection, and innovativeness should also be central to the evaluation. By employing these detailed assessment criteria, instructors can better understand students' actual programming abilities and provide targeted feedback.

Team collaboration and innovation should also be incorporated into the evaluation system. The division of roles within team projects, communication and collaboration skills, and the innovative solutions proposed during projects will be key assessment areas. Particularly in interdisciplinary projects, students' coordination and teamwork levels can reflect their collaborative potential in future workplaces. Instructors can evaluate team collaboration through the collaborative processes among students, peer evaluations within the team, and the final outcomes of the project. Additionally, the assessment of innovation can be based on students' proposals for novel algorithms, unconventional programming approaches, or optimization solutions. Through the construction of a diversified evaluation system, the issue of "one-size-fits-all" evaluation criteria can be effectively avoided, allowing for a more accurate measurement of students' abilities and potential in AI and big data applications, ensuring comprehensive enhancement in both theory and practice.

2.4 Introducing Innovative Teaching Tools and Resources

To improve teaching effectiveness and enhance students' autonomous learning abilities, course reform should actively introduce innovative teaching tools and resources. First, by utilizing online learning platforms and virtual laboratories, students can engage in self-directed learning and programming practice outside of class. For example, instructors can provide datasets and relevant AI project templates, allowing students to train models and analyze data in a virtual environment, overcoming the limitations of traditional classroom time and space. This approach enables students to flexibly arrange their learning pace and repeatedly practice key technical elements, significantly enhancing their learning efficiency and outcomes.

Second, by leveraging cloud computing resources, students can utilize cloud servers for processing large datasets and training models, addressing the issue of insufficient computing power on local devices. This cloud-based technical support allows students to experience industrial-level AI and big data processing scenarios, such as handling massive data and distributed computing through platforms like AWS and Google Cloud. Not only does this help students better understand how to tackle real-world computing constraints, but it also enhances their awareness of distributed computing and resource management, cultivating their ability to handle complex data tasks.

Additionally, automated code assessment systems can be introduced in teaching, utilizing AI to automatically grade students' code assignments and provide immediate feedback, helping students promptly identify issues and make improvements. These automated assessment systems can evaluate not only the correctness of students' code but also provide a comprehensive evaluation based on code execution efficiency and adherence to standards. This not only improves the efficiency of instructors' grading but also ensures that every student receives timely and detailed feedback, aiding them in quickly correcting mistakes and optimizing their code. Through this immediate feedback mechanism, students can enhance their programming skills more rapidly and gain a deeper understanding of common issues and optimization strategies in AI and big data programming.

3. Practical Effects of the Reform of the "Python Programming" Course for Applications in Artificial Intelligence and Big Data

3.1 Improvement in Students' Big Data and AI Programming Skills

Through the course reform, students have significantly improved their capabilities in big data processing and artificial intelligence programming. The optimized course content not only covers fundamental Python syntax but also places a strong emphasis on modules related to data analysis, machine learning, and deep learning. Students have acquired practical skills in data preprocessing, algorithm implementation, and handling large datasets. In the project-driven teaching model, students have the opportunity to apply their knowledge in real projects, completing the entire process from data collection and cleaning to model training and optimization. Particularly, through in-depth learning of data science libraries like Pandas and NumPy, as well as hands-on experience with AI frameworks like TensorFlow and Keras, students can proficiently utilize Python for complex AI algorithm development and data analysis tasks. This practice-oriented teaching model enables students to move beyond theoretical learning and achieve profound improvements in programming skills through actual operations, laying a solid foundation for their future careers in the fields of artificial intelligence and big data. [5]

3.2 Significant Improvement in Innovative Thinking and Team Collaboration Skills

In the practical teaching following the course reform, students have demonstrated a marked increase in their innovative thinking and teamwork abilities. The project-driven teaching method encourages students to actively participate in solving real problems rather than passively receiving knowledge. Each project requires students to explore independently, propose innovative solutions, and validate the feasibility of their solutions through experimentation and analysis. For instance, in AI projects, students need to design suitable models for different datasets and enhance their performance through algorithm tuning and model optimization, thereby cultivating their innovative capabilities. Moreover, the team collaboration model in projects teaches students how to work together to complete complex tasks through communication and division of labor. Each group member undertakes different modules—from data processing and model design to result presentation—necessitating constant collaboration and mutual feedback throughout the process. This cooperative learning approach not only enhances students'

teamwork skills but also equips them with the ability to integrate various technologies and knowledge across disciplines, leading to innovative solutions that are crucial for their future professional development.

3.3 Teaching Feedback and Directions for Continuous Optimization

The effectiveness of the course reform has also been validated through teaching feedback. Students have highly evaluated the reformed course content and teaching model, generally agreeing that the new course material is more practical and that the project practice segments allow them to learn core technologies in big data and artificial intelligence within real-world contexts. Simultaneously, instructors have continuously optimized course structures and teaching strategies based on data analysis and student feedback. For example, instructors can adjust the difficulty of the teaching content or introduce more challenging AI and big data application cases based on students' performances in projects. Additionally, feedback from the automated code assessment system enables instructors to identify common issues students face during programming, such as frequent algorithm implementation errors or insufficient performance optimization, allowing for targeted explanations in subsequent classes. Moving forward, the course will continually introduce new teaching tools and cutting-edge technologies and regularly adjust the syllabus to ensure students are always learning the latest industry advancements. Through ongoing feedback and improvements, the impact of the course reform will become even more pronounced, providing strong support for cultivating talents with innovative capabilities and practical application skills in artificial intelligence and big data. [6]

Conclusion

Through the reform of the "Python Programming" course, the course content has become more aligned with the demands of artificial intelligence and big data applications, leading to a significant enhancement in students' programming abilities in areas such as big data processing and AI algorithm implementation. Additionally, the project-driven teaching model has effectively stimulated students' innovative thinking and teamwork skills, resulting in positive feedback on the teaching outcomes. Based on this, future courses will continue to introduce more cutting-edge technologies and align with the latest trends in industry development, ensuring that students are always equipped to handle complex technical environments. Furthermore, teaching methods and evaluation systems will be continually optimized to better support students' personalized learning and long-term development.

Fund project

Yinchuan University of Science and Technology Project No.: 2023XJJG017 Project name: "Python Program Design" curriculum reform and practice

References

[1] Zhao, H. (2024). Construction of an Online Course for Python Programming in a Mobile Interactive Environment. China Information World, (05), 216-218.
[2] Hu, Q., & Yang, J. (2024). Teaching Reform of Assembly Language Programming Course. Computer Education, (08), 92-96.
[3] Qu, X., Wu, W., & Hou, X. (2024). Teaching Practice of Object-Oriented Programming Course under the Background of New Engineering. Journal of Shanxi Datong University (Natural Science Edition), 40(04), 22-25.
[4] Xu, W. (2024). Research on the Teaching Reform of Python Programming Course Based on the OBE Concept. Computer Knowledge and Technology, 20(21), 169-171+177.
[5] Wang, W., & Yu, H. (2024). Practical Exploration of Python Programming Course under the "Double High" Construction Background. Fujian Education, (30), 49-53.
[6] Gao, H., & Chen, Z. (2024). Empowering Teaching Reform of Programming Courses with Artificial Intelligence. Computer Education, (07), 41-43+48.