EJPMR, 2025, 12(9), 504-512 **Review Article** **Impact Factor 7.065**

EUROPEAN JOURNAL OF PHARMACEUTICAL AND MEDICAL RESEARCH

www.ejpmr.com

ISSN (O): 2394-3211
ISSN (P): 3051-2573
Coden USA: EJPMAG

ARTIFICIAL INTELLIGENCE AND PYTHON PROGRAMMING IN UNDERGRADUATE PHARMACY EDUCATION: ADVANTAGES, CHALLENGES FOR TEACHERS AND STUDENTS, AND SUGGESTIONS FOR EFFECTIVE IMPLEMENTATION

Vikash Agnihotri[1], Ankur Patel[2] & Tapan Kumar Mahato[3*]

[1]B.Pharmacy College Rampura, Godhra, Gujarat, India.
[2]Sardar Patel College of Pharmacy, Anand, Gujarat, India.
[3]Faculty of Pharmacy, Bhupal Nobles' University, Udaipur, Rajasthan, India.

***Corresponding Author: Tapan Kumar Mahato**

Faculty of Pharmacy, Bhupal Nobles' University, Udaipur, Rajasthan, India.

DOI: https://doi.org/10.5281/zenodo.17464084

How to cite this Article: Agnihotri, V., Patel, A., & Mahato, T. K. (2025). Artificial intelligence and Python programming in undergraduate pharmacy education: Advantages, challenges for teachers and students, and suggestions for effective implementation. European Journal of Pharmaceutical and Medical Research, 12(9), 504–512.

Article Received on 10/08/2025 **Article Revised on 20/09/2025** **Article Published on 30/09/2025**

ABSTRACT

The proposed subject Artificial Intelligence (AI) & Python Programming for Pharmacy – I represent a pioneering addition to the Bachelor of Pharmacy curriculum as envisioned by the Pharmacy Council of India (PCI). This course equips pharmacy graduates with the essential foundations of AI, machine learning (ML), and Python programming, enabling them to operate effectively in data-driven and technology-enabled pharmaceutical environments. Beginning with Unit 1, students explore the history, key approaches, and knowledge-representation methods of AI, gaining a conceptual grounding in symbolic, statistical, and connectionist paradigms. Unit 2 introduces core ML paradigms, supervised, unsupervised, and reinforcement learning, along with representative algorithms such as Naïve Bayes, k-Nearest Neighbours, regression models, clustering methods, and basic neural networks, fostering analytical problem-solving. Unit 3 contextualizes these approaches within pharmaceutical sciences, highlighting applications in drug discovery, formulation, quality assurance, pharmacovigilance, and regulatory science. Units 4 and 5 progressively train students in Python, from installation, syntax, and data types to advanced programming constructs, data handling, functions, file management, and exception handling. By integrating AI/ML theory with Python programming, students are prepared to analyze pharmaceutical case studies, implement small coding exercises, and develop mini-projects relevant to real-world challenges in formulation, manufacturing, and patient care. The course uniquely addresses both technical and domain-specific requirements through pharmacy-centered examples, ensuring relevance and engagement. With its forward-looking vision, this subject positions B.Pharm graduates for leadership roles in computational drug discovery, precision medicine, and intelligent pharmaceutical systems, aligning education with the evolving demands of healthcare and industry.

KEYWORDS: Artificial Intelligence (AI), Machine Learning (ML), Python Programming, Pharmaceutical Applications, Drug Discovery and Development, Precision Medicine.

I. INTRODUCTION

Pharmacy is a vital healthcare discipline focused on drug discovery, development, manufacturing, and patient care. With the rapid digital transformation, Artificial Intelligence (AI) is emerging as a game-changer, offering data-driven solutions for formulation design, pharmacovigilance, clinical trials, and precision medicine. Machine Learning (ML), a core branch of AI, enables predictive modelling and decision support, while Python programming provides a simple yet powerful platform for implementing these algorithms. Integrating AI and Python into pharmacy education empowers future pharmacists with computational skills, preparing them for leadership in intelligent drug discovery, personalized therapies, and technology-enabled pharmaceutical systems. Let's discuss about advantages, challenges for teachers and students and steps for smooth implementation of all units of the subject Artificial

Intelligence (AI) & Python Programming for Pharmacy – I which is in first semester of Bachelor of pharmacy.

Unit 1 (Foundations of AI): Advantages, challenges for teachers and students and steps for smooth implementation in a practical way

Advantages

i. Strong foundational understanding: Understanding the historical progression of AI provides pharmacy students with valuable context. Early symbolic, rule-based systems mimicked decision-trees and treatment guidelines, but lacked adaptability. Statistical AI introduced data-driven reasoning, while connectionist models such as neural networks now dominate modern drug discovery.[6] In pharmacy, these paradigms translate into decision support, from simple drug–drug interaction checkers (symbolic) to advanced QSAR/QSPR predictive models (connectionist).[5] This helps students appreciate AI not as just coding, but as evolving frameworks for pharmaceutical problem-solving.

ii. Multidisciplinary relevance: AI in pharmacy thrives at the intersection of statistics, computer science, biology, and clinical sciences. For instance, knowledge representation techniques can be used to map relationships between herbal phytochemicals and therapeutic activities, while reasoning under uncertainty supports patient care when medical records are incomplete.[4] These applications highlight that AI is not an isolated discipline but a bridge across multiple scientific fields essential to pharmaceutical sciences.

iii. Decision-making in healthcare: AI enhances pharmaceutical decision-making by offering structured strategies for handling complex patient data. In precision medicine, ML models analyze genomics to recommend individualized therapies.[2] For example, dose adjustments in anticoagulant therapy can be guided by algorithms that consider patient genetic markers. Similarly, in hospital pharmacy, AI can support drug utilization reviews by suggesting safer alternatives when interactions are detected.[3] These examples show how AI frameworks improve evidence-based decision-making in pharmacy practice.

iv. Bridging Tech and Pharma: Pharmacy students often come from strong chemistry and biology backgrounds but lack computational exposure. Introducing AI concepts such as symbolic vs. statistical reasoning helps bridge this gap. For instance, AI-based graph models can represent drug–drug interactions visually, while image recognition systems can detect microscopic formulation defects more reliably than manual inspection.[4] By connecting computational logic with practical pharmacy examples, this unit ensures students see AI as an extension of pharmaceutical practice rather than a foreign subject.

v. Employability & leadership: Equipping pharmacy students with AI skills enhances their career opportunities in a data-driven pharmaceutical industry. Graduates can lead projects in computational drug discovery, regulatory science, pharmacovigilance, and intelligent manufacturing systems.[5] AI-trained pharmacists are also positioned to work with real-world evidence, which regulatory authorities increasingly use to guide approvals.[7] Thus, pharmacy graduates gain not only technical employability but also leadership capacity to shape the future of healthcare innovation.

Challenges for teachers

i. Interdisciplinary knowledge gap: One of the biggest challenges for teachers introducing AI to pharmacy students is the interdisciplinary knowledge gap. Pharmacy faculty are typically experts in pharmaceutics, pharmacology, or pharmacognosy, but may not have advanced training in AI concepts such as symbolic reasoning, probabilistic models, or neural networks. For example, explaining how Bayesian inference supports pharmacovigilance signal detection requires both statistical and computational expertise, which is often outside the comfort zone of traditional pharmacy educators.[4] Bridging this gap may require collaborative teaching with computer science experts or additional faculty development programs, so that complex AI principles can be explained in a pharmacy-relevant context.[3]

ii. Lack of domain-specific examples: Another challenge is the lack of pharmacy-specific examples in conventional AI teaching materials. Standard AI textbooks often rely on illustrations like chess games, spam detection, or weather forecasting, which pharmacy students may find irrelevant. This disconnect can make it difficult for teachers to maintain learner engagement. To contextualize AI, pharmacy educators must create tailored examples, such as demonstrating how knowledge representation techniques can map herbal drug–constituent–activity relationships or how decision trees can classify patients into responders and non-responders in a clinical setting.[5] However, developing these domain-specific examples requires extra time and preparation, and many teachers lack access to ready-made teaching resources designed for pharmacy.

iii. Balancing theory and application: A further difficulty lies in finding the right balance between theory-heavy concepts and practical applications. Unit 1 introduces abstract topics such as symbolic versus connectionist approaches, knowledge representation, and reasoning under uncertainty. If these are taught only theoretically, they may feel distant from pharmacy practice. Conversely, rushing into coding exercises without sufficient conceptual grounding may overwhelm students who have little programming background.[6] For example, teaching decision-making strategies solely through flowcharts may make the subject dry, while teaching only with Python scripts may confuse beginners. Teachers must therefore balance both, explaining theoretical principles and linking them to pharmaceutical case studies, such as using uncertainty reasoning to guide drug selection when patient data is incomplete.[2]

Challenges for students

i. Cognitive overload: For many pharmacy students, Artificial Intelligence is their first encounter with computational concepts, which can cause cognitive overload. Unlike laboratory-based experiments, AI requires them to engage with abstract ideas such as symbolic reasoning, probabilistic models, and knowledge representation. For example, understanding how Bayesian networks estimate the likelihood of adverse drug reactions may feel overwhelming compared to a traditional pharmacology practical. Without scaffolding, students risk becoming disengaged when faced with too many new concepts at once.[4]

ii. Limited coding integration: Since Unit 1 emphasizes conceptual understanding rather than programming, students may struggle to see the practical link between theory and coding. For instance, they may understand the idea of symbolic AI but fail to visualize how it would look in Python as a rule-based decision system for checking drug–drug interactions. Without small, early coding demonstrations, students may perceive AI as abstract theory rather than a practical tool for drug design, pharmacovigilance, or patient care.[3]

iii. Adjustment to new pedagogy: Pharmacy education has traditionally been experiment-heavy, focusing on laboratory practicals and formulation trials. Shifting to AI requires computational and statistical reasoning, which feels unfamiliar to many students. For example, reasoning under uncertainty demands comfort with probability, something many pharmacy students lack confidence in. This transition to algorithmic thinking can cause anxiety, especially among learners who identify as "non-math" students. Research shows that bridging this gap requires carefully designed pedagogy that mixes computational and pharmaceutical contexts.[5]

iv. Resource dependence: AI education requires computers, IDEs (Integrated Development Environment), and datasets, which may not be available in all pharmacy institutions. Students in under-resourced colleges often rely on shared labs, outdated hardware, or limited internet connectivity. Even when resources are available, pharmacy-relevant datasets (e.g., clinical trial data, pharmacovigilance records) are usually proprietary. As a result, students often work with artificial or very small datasets, which limits their ability to see how AI truly functions in large-scale pharmaceutical applications.[2, 7]

Suggestions for smooth implementation

i. Use case studies: The best way to make AI concepts relevant to pharmacy students is through domain-specific case studies. For example, symbolic AI can be demonstrated with a drug–drug interaction checker, where rule-based logic flags potential adverse combinations. Similarly, connectionist models like neural networks can be introduced with a QSAR (Quantitative Structure–Activity Relationship) example, predicting the biological activity of new compounds.[5] Case studies like these transform abstract theory into recognizable pharmaceutical applications, improving student engagement and comprehension.

ii. Integrate mini exercises: Pharmacy educators should design mini classroom exercises to help students apply theoretical concepts. For instance, while teaching knowledge representation, students could create a hierarchy mapping drug classifications e.g., antibiotics subdivided into penicillins, cephalosporins, and fluoroquinolones. Such simple but practical exercises reinforce learning while connecting AI methods to existing pharmacy knowledge.[3]

iii. Promote team-teaching models: Because AI spans both technical and pharmaceutical expertise, a collaborative teaching approach is highly effective. Pharmacy educators can explain pharmacological or regulatory contexts, while computer science experts demonstrate technical implementation. For example, in teaching pharmacovigilance, the pharmacy faculty might describe how adverse event reports are generated, while a computer science faculty member shows how text-mining algorithms classify ADRs. This team-teaching model ensures students see both perspectives and reduces the interdisciplinary gap.[4]

iv. Provide visual aids and flowcharts: Visual learning tools, such as diagrams and flowcharts, simplify complex AI paradigms. A comparative chart of symbolic vs. statistical approaches or a flow diagram of a Bayesian reasoning process can help students quickly grasp abstract ideas. In pharmacotherapy teaching, flowcharts are already widely used to explain treatment guidelines; applying the same method to AI ensures continuity in learning styles.[6]

v. Encourage real-world connections early: From the first sessions, educators should emphasize how AI relates to real-world pharmacy challenges. For instance, reasoning under uncertainty can be explained through clinical examples, deciding on an antibiotic when microbial sensitivity data is incomplete. Decision-making strategies can be connected to personalized medicine, where AI helps select chemotherapy regimens based on patient genetics.[2] By showing such applications early, students learn to see AI not as a separate subject, but as a practical tool integrated into pharmaceutical practice.

Unit 2 (Machine learning essentials): Advantages, challenges for teachers and students and steps for smooth implementation in a practical way

Advantages

i. Exposure to learning paradigms: Introducing students to supervised, unsupervised, and reinforcement learning provides them with a strong foundation for recognizing how different approaches solve different pharmaceutical problems. For instance, supervised learning can be applied to predict whether a tablet formulation passes stability testing based on temperature and pH values, while unsupervised learning can cluster patient populations according to therapeutic responses in clinical trials. Reinforcement learning can optimize intelligent manufacturing systems, where production parameters are continuously adjusted to maintain product quality. This broad exposure helps students appreciate the flexibility of ML for addressing real-world pharmacy challenges.[1]

ii. Practical relevance of algorithms: The inclusion of core algorithms such as Naïve Bayes, k-Nearest Neighbours (KNN), regression models, clustering methods, and basic neural networks makes Unit 2 highly practical. In pharmacovigilance, Naïve Bayes classifiers can categorize adverse drug reactions; KNN can classify new drug molecules by comparing them to structurally similar ones; logistic regression can model the probability of treatment success; and clustering algorithms like k-means can group herbal extracts based on their HPTLC fingerprints. Even simple neural networks provide students with an introduction to the architecture behind QSAR/QSPR models used in drug discovery.[9]

iii. Builds analytical thinking: By learning when and how to apply different algorithms, students cultivate analytical and problem-solving skills. Instead of memorizing facts, they begin to think critically about selecting the right method for the right dataset. For example, they may decide to use clustering to identify subgroups in a clinical trial dataset or regression to predict drug release profiles. This shift from rote learning to evidence-based reasoning prepares students for modern pharmaceutical research, where data-driven decisions are essential.[10]

iv. Early coding integration: Unit 2 also integrates practical coding exercises in Python, giving students early exposure to implementing algorithms. Even simple exercises, such as using logistic regression to predict whether cholesterol levels indicate cardiovascular risk or applying k-means clustering to dissolution data, allow them to see the transition from theory to application. This hands-on coding experience reduces fear of programming and boosts student confidence, while demonstrating the practical power of ML in pharmacy.[8]

Challenges for teachers

i. Mathematical and statistical gap: Teaching machine learning requires explaining concepts such as probability, linear algebra, regression, and optimization, which many pharmacy faculty may not be fully comfortable with. For example, while explaining logistic regression, a teacher must show how the algorithm predicts the probability of therapeutic success or treatment failure based on patient data. Without strong mathematical and computational training, teachers may struggle to simplify these concepts for students from non-computer science backgrounds.[1]

ii. Lack of pharmacy-specific datasets: Most standard AI/ML teaching datasets (like Iris flowers or MNIST handwritten digits) feel irrelevant to pharmacy students. To make the subject engaging, teachers must contextualize algorithms with pharmaceutical datasets such as dissolution profiles, HPTLC fingerprints, or adverse drug event reports. However, such datasets are often unavailable or proprietary, making it challenging for educators to design realistic classroom activities.[9]

iii. Limited time versus wide content: Unit 2 introduces multiple algorithms—Naïve Bayes, KNN, regression, clustering, and neural networks, within a short duration (7 hours). Covering theory, coding, and applications together can be overwhelming both for teachers and students. Teachers may end up oversimplifying neural networks or skipping practical demonstrations due to time constraints, reducing the effectiveness of learning.[8]

Challenges for students

i. Intimidation by mathematics: Many pharmacy students have strong foundations in chemistry and biology but weaker exposure to mathematics and statistics. Machine learning relies on probability, linear algebra, and optimization, which may intimidate students unfamiliar with quantitative reasoning. For example, calculating Euclidean distances in KNN or understanding cost functions in regression may seem abstract compared to hands-on pharmaceutics. This lack of confidence in mathematics often becomes a barrier to fully engaging with ML concepts.[1]

ii. Algorithm confusion: Since multiple algorithms are introduced within a short span, students often struggle to differentiate their applications. For example, they may confuse classification (logistic regression) with clustering (k-means), assuming both serve the same purpose. Without repeated reinforcement, these misconceptions persist, leading to difficulty in applying the right tool to the right problem in pharmaceutical datasets.[10]

iii. Transition from theory to coding: Even when students grasp algorithm theory, implementing it in Python is a challenge. Issues such as library imports, syntax errors, and data preprocessing often discourage beginners. For example, a student may understand how logistic regression predicts treatment outcomes but struggle to run the same model in scikit-learn due to technical barriers. This transition from conceptual knowledge to coding practice can create frustration if not scaffolded carefully.[8]

iv. Resource and dataset constraints: Effective ML learning requires access to computers, Python IDEs, and domain-relevant datasets. In resource-limited institutions, students may rely on shared labs with outdated software, reducing practice opportunities. Moreover, pharmacy-relevant datasets such as ADR reports or dissolution profiles are often proprietary or confidential, limiting exposure to realistic case studies. Students may therefore end up practicing only on toy datasets, which restricts their ability to visualize real-world pharmaceutical applications.[9]

Suggestions for smooth implementation

i. Use domain-specific case studies: To make machine learning concepts meaningful, educators should embed them in pharmacy-specific case studies. For example, logistic regression can be demonstrated by predicting whether a tablet batch will pass dissolution testing, while k-means clustering can be used to group herbal extracts based on HPTLC profiles. Reinforcement learning can be introduced using an example of adaptive process control in granulation or coating. These pharmacy-focused applications make abstract ML methods more engaging.[9]

ii. Provide guided mini-exercises: Small, structured exercises help students gradually build confidence. For instance, students could apply Naïve Bayes to classify adverse drug reaction (ADR) reports or use KNN to identify drug similarity based on physicochemical descriptors. By starting with simple datasets and gradually moving to pharmaceutical data, students can better understand how algorithms translate into real-world problem-solving.[10]

iii. Emphasize visualization and interpretation: Since pharmacy students may not have strong mathematical backgrounds, visual tools like scatter plots, clustering maps, and regression curves can simplify concepts. For example, plotting dissolution values into clusters helps students immediately “see” how ML groups similar formulations. Visualization bridges the gap between theoretical understanding and practical application, reducing math anxiety.[1]

iv. Encourage collaborative and team-based learning: Team-based or peer-learning models are highly effective. Stronger students can help beginners debug code, while interdisciplinary teaching teams (pharmacy + computer science faculty) can provide complementary expertise. For example, a pharmacy professor could explain ADR data, while a computer scientist demonstrates classification algorithms on that dataset. This approach reduces both teacher knowledge gaps and student frustration.[8]

v. Connect ML to real-world industry use cases: From the beginning, students should be shown how ML is transforming pharmaceutical industries, drug discovery pipelines, process optimization, clinical trial design, and pharmacovigilance. For example, students can learn how clustering helps identify patient subgroups in clinical trials, or how neural networks support QSAR modelling. Linking algorithms to current pharmaceutical innovations ensures students view ML as an essential professional skill.[9]

Unit 3 (AI/ML in Pharmaceutical Sciences): Advantages, challenges for teachers and students and steps for smooth implementation in a practical way

Advantages

i. Strong industrial relevance: AI/ML is increasingly applied across the pharmaceutical industry. For example, in formulation development, predictive models can optimize excipient ratios to achieve desired release profiles. In quality control, computer vision models can detect cracks or discoloration in tablets more efficiently than manual inspection. Similarly, supply-chain analytics powered by ML helps forecast demand and prevent drug shortages.[1]

ii. Research-oriented applications: AI/ML opens up vast research opportunities in pharmacy. For instance, QSAR/QSPR models predict drug activity and solubility, guiding medicinal chemistry decisions. Deep learning models like AlphaFold have revolutionized protein structure prediction, speeding up drug discovery. In clinical trial design, ML can optimize patient selection and predict dropout risks, making studies faster and more efficient.[9]

iii. Personalized medicine and regulatory impact: AI/ML enables precision medicine, tailoring treatments based on patient-specific genetic or clinical data. For example, algorithms can help determine the optimal warfarin dose for an individual by analyzing pharmacogenomic markers. Regulatory authorities, including the FDA, increasingly use AI to analyze real-world evidence from health records and pharmacovigilance databases to guide approvals, shaping the future of regulatory science.[8]

Challenges for teachers

i. Rapidly evolving field: The field of AI in pharmaceutical sciences evolves quickly, with new tools like deep generative models and transformer-based architectures emerging every year. Teachers may find it challenging to keep their teaching material updated with these innovations.[4]

ii. Limited access to datasets: Many AI applications require large proprietary datasets, such as clinical trial or pharmacogenomic data. Teachers often lack access to these, forcing them to rely on simplified or simulated datasets that may not fully capture the complexity of real-world pharmacy problems.[9]

iii. Bridging conceptual vs. applied learning: Explaining how theoretical ML concepts link to real pharmaceutical workflows is difficult. For instance, connecting logistic regression to patient stratification in trials requires both technical and domain knowledge.

Without careful integration, students may see AI as detached from pharmacy practice.[1]

Challenges for students

i. Overwhelming breadth of applications: Unit 3 spans formulation, manufacturing, QC, supply chain, pharmacovigilance, drug discovery, QSAR, clinical trials, personalized medicine, and regulatory science, all in limited hours. Students may feel overwhelmed by the breadth without sufficient depth.[10]

ii. Difficulty in connecting theory to practice: Students may understand abstract algorithms but struggle to see how they operate in real-world workflows. For example, clustering might be taught conceptually, but without industry examples, students may not visualize how it segments patients in a clinical trial.[8]

iii. Expectation gap: Students often expect extensive coding for drug discovery or clinical trials, but due to limited resources, the focus is often on case studies and theory. This mismatch between expectation and delivery can cause disengagement unless clarified early.[4]

Suggestions for smooth implementation

i. Use domain-specific case studies: Teachers should integrate realistic case studies, such as AI predicting dissolution profiles, or classifying ADRs using text mining. These help students understand how algorithms fit into pharmaceutical processes.[9]

ii. Incorporate mini-projects: Students can be given small projects, such as clustering dissolution data, running regression on drug release profiles, or using Naïve Bayes to classify ADRs. Even small-scale projects make AI applications tangible and relevant.[10]

iii. Visualize workflows: Flowcharts showing the role of AI in the drug development pipeline, from discovery to QC and pharmacovigilance, can help students grasp the big picture and contextualize individual algorithms.[10]

iv. Promote industry interaction: Inviting guest lecturers from pharmaceutical AI projects or organizing webinars can bridge the classroom–industry gap. Hearing about real use cases such as AI in vaccine distribution or clinical trial analytics motivates students and strengthens learning relevance.[8]

Unit 4 (Python setup and language basics): Advantages, challenges for teachers and students and steps for smooth implementation in a practical way

Advantages

i. Accessibility and relevance: Python is one of the most widely used programming languages in AI/ML and scientific computing because of its simplicity and readability. For pharmacy students, this makes it easier to begin coding compared to complex languages like C++ or Java. For example, students can quickly write a script to calculate drug doses or convert storage temperatures from Fahrenheit to Celsius, which shows them how Python directly supports pharmaceutical tasks.[10]

ii. Smooth onboarding with IDEs: Unit 4 introduces students to popular development environments such as Jupyter Notebook, VS Code, and PyCharm. Jupyter Notebook, in particular, allows stepwise execution and inline graph visualization, making it ideal for analyzing dissolution profiles or plotting drug release curves. This helps students connect coding tools with laboratory-style workflows they already understand.[9]

iii. Foundation for advanced learning: Learning Python's syntax rules, variables, and built-in data types lays a strong foundation for later units. For example, integers and floats can represent stability study values (e.g., pH 5.8, hardness 7 kg/cm²), while strings store drug names. These basics prepare students to later use Python libraries such as NumPy, Pandas, and Scikit-learn for pharmaceutical data analysis.[1]

iv. Confidence building through simple scripts: Writing and running their first Python script builds student confidence. For example, a script that prints a patient's medication schedule or calculates BMI reinforces the idea that coding is accessible and directly useful in pharmacy practice. These early wins reduce fear and motivate students to engage with more complex coding later.[8]

Challenges for teachers

i. Mixed student backgrounds: In pharmacy classrooms, some students may have prior coding exposure while others are complete beginners. Teachers must balance the pace so that advanced learners are not bored and beginners are not overwhelmed. This requires adaptive teaching strategies.[10]

ii. Technical setup issues: Installing Python and setting up IDEs can be time-consuming in institutions with limited resources or internet access. For example, downloading packages such as Pandas or Matplotlib may be difficult in resource-limited environments, reducing hands-on practice opportunities.[1]

iii. Lack of pharma-specific examples: Generic coding exercises such as "calculate the sum of two numbers" may feel irrelevant. Teachers must design pharmacy-centered coding tasks, like calculating pediatric doses or storing drug release data. Developing these examples takes time and interdisciplinary preparation.[9]

Challenges for students

i. Syntax sensitivity: Python is indentation-sensitive, which can confuse beginners. A misplaced tab or missing colon in a conditional may cause frustrating errors such as IndentationError. For example, when writing a dose-adjustment script, even a small spacing error may prevent execution.[8]

ii. Transition to computational thinking: Pharmacy students are trained in laboratory methods and may struggle to view problems as variables and logic flows. For instance, representing solubility data as floats (7.5) or drug names as strings ("Moxifloxacin") requires abstract computational reasoning. This mental shift can be difficult at first.[10]

iii. Initial fear of coding: Many pharmacy students perceive coding as a "technical" subject outside their domain. Even simple exercises such as writing a script to print drug names may intimidate them, leading to self-doubt if not guided effectively.[1]

iv. Resource dependence: Access to computers, updated IDEs, and datasets is not always guaranteed. Students in some institutions may rely on outdated lab systems or shared devices, reducing coding practice opportunities.[9]

Suggestions for smooth implementation

i. Provide pharmacy-based coding examples: From the beginning, coding tasks should be directly related to pharmacy. For example:

a. Declaring variables for drug names and doses.
b. Converting "400" (string) to 400 (int) to calculate dose strength.
c. Writing a script to calculate pediatric doses using Young's Rule or Clark's Rule. This ensures that students see coding as directly applicable to their profession.[8]

ii. Use stepwise demonstrations: Teachers can begin with Jupyter Notebook, which allows stepwise execution. For instance, first declare variables for drug release data, then gradually use them in calculations. This stepwise approach reduces overload.[1]

iii. Encourage collaborative learning: Pairing stronger students with beginners for coding exercises can reduce anxiety. For example, groups can work on writing a script that prints a patient's medication chart. This peer-assisted learning builds confidence.[10]

iv. Emphasize debugging skills: Teachers should guide students in reading error messages and correcting mistakes. For example, showing how to fix an Indentation Error or Type Error builds resilience and problem-solving skills in programming.[9]

Unit 5 (Python programming constructs & data handling): Advantages, challenges for teachers and students and steps for smooth implementation in a practical way

Advantages

i. Development of logical problem-solving: Unit 5 introduces constructs like operators, conditional statements, and loops, which train students to think algorithmically. For example, an if–else statement can automate dose adjustments in patients with renal impairment, while loops can print drug release values across multiple time points. This logical structuring of problems prepares students for computational problem-solving in pharmaceutical contexts.[10]

ii. Handling pharmaceutical data efficiently: Pharmacy students often work with large datasets—clinical results, dissolution profiles, or stability data. Python's lists and tuples allow them to efficiently store and process such data. For example, a list [10, 25, 40, 70] could represent % drug release at intervals, while slicing extracts values at specified time points. Tuples ensure immutability for pharmacopoeial standards (e.g., temperature ranges).[1]

iii. Reusable programming through functions: Functions encourage modular, reusable code. For instance, a function calculate_dose (weight, mg_per_kg) can be applied repeatedly for pediatric dosing calculations. Similarly, functions can calculate drug release or convert storage temperatures, making coding more efficient and systematic.[8]

iv. Digital record keeping with file handling: File handling (open, read, write, close) enables digital record management. Students can automate writing batch records, such as: Batch F2 – pH: 5.8 – Release: 78% and retrieve the data later. This skill aligns with digital documentation requirements in Good Manufacturing Practice (GMP).[9]

v. Error-resilient programming with exception handling: Using try–except–finally teaches students how to build robust programs that don't crash when unexpected data appears. For example, if a dataset includes "NaN" in patient age, the program can skip it and continue analysis. This mirrors the error-handling requirements in pharmaceutical software where safety is critical.[8]

Challenges for teachers

i. Balancing depth and breadth: Unit 5 covers many programming constructs in limited time, operators, conditionals, loops, collections, functions, file handling, and exceptions. Teachers often struggle to balance depth with coverage, risking superficial explanations or skipped coding demonstrations.[10]

ii. Creating pharma-relevant examples: Generic tasks like calculating factorials may disengage pharmacy students. Teachers must design pharmacy-relevant coding problems (e.g., simulating dissolution studies or calculating doses). Creating such examples requires extra preparation and domain knowledge.[1]

iii. Debugging difficulties: Students often get stuck in loops or misapply conditions, leading to infinite iterations or errors. Teachers need to spend significant time teaching debugging strategies, which can reduce time for covering other topics.[8]

Challenges for students

i. Logical thinking barriers: Students unfamiliar with computational thinking may struggle with translating real-world pharmacy rules into conditions. For example, coding a rule like: *"If patient age > 65 and creatinine clearance < 50 → reduce dose"* requires careful logic structuring.[9]

ii. Confusion with collections: Students may mix up lists, tuples, or slicing. For example, misusing slicing on dissolution data could lead to incorrect interpretation of results. This hampers their ability to use Python effectively for pharma data analysis.[10]

iii. Transition to larger programs: Up to this point, students may have written small scripts. Unit 5 requires combining loops, functions, and file handling, which can feel overwhelming. For example, writing a full program to calculate doses, save them to a file, and handle missing values can intimidate beginners.[1]

iv. Frustration with errors: Frequent Indentation Error or Type Error messages can frustrate students. Without proper debugging support, they may feel demotivated when programs fail, especially in multi-step coding tasks.[8]

Suggestions for smooth implementation

i. Start with pharmacy-centered coding: Teachers should replace generic programming exercises with pharmacy tasks. For example:

a. Use if–else to classify whether a drug passes stability testing.
b. Loop through drug release values across time points.
c. Store antibiotics in a list and filter by class (e.g., fluoroquinolones). This contextualizes coding in pharmacy practice.[9]

ii. Use stepwise learning: Introduce loops and functions incrementally. Begin with simple numeric ranges before moving to pharma datasets. This progressive learning reduces overload.[10]

iii. Demonstrate file handling with batch records: Instead of abstract files, use examples like batch production logs or patient data files. Writing and reading files that store drug release or dosing data gives students practical insight into pharma documentation.[1]

iv. Teach debugging as a core skill: Students should learn to read error messages and troubleshoot systematically. For example, demonstrating how to resolve an Indentation Error when writing a dose calculator script makes them more resilient coders.[8]

v. Encourage mini-projects: End the unit with mini-projects that combine all constructs. Example projects:

a. Pediatric dose calculator with file logging and error handling.
b. A dissolution profile analyzer that plots release data.
c. A program to classify patients as hypertensive or normal using conditional logic. These integrative tasks strengthen learning and confidence.[9]

II. CONCLUSION

The inclusion of AI & Python Programming for Pharmacy – I in the B.Pharm curriculum marks a transformative step toward bridging computational sciences with pharmacy education. By covering five structured units, the subject ensures a balance between conceptual understanding, analytical reasoning, and practical coding skills. Students benefit from learning the evolution of AI, core problem-solving paradigms, and knowledge-representation approaches, which build a strong foundation for applying ML algorithms to pharmaceutical contexts. The integration of supervised, unsupervised, and reinforcement learning paradigms, along with algorithms like Naïve Bayes, KNN, regression, clustering, and neural networks, provides learners with a versatile toolkit for addressing diverse challenges in drug development, formulation, clinical trials, and pharmacovigilance. Practical exposure to Python programming enhances computational literacy, allowing students to design scripts, manage datasets, implement decision-support algorithms, and maintain digital records with error resilience. Teachers may face challenges such as the interdisciplinary knowledge gap, lack of domain-specific datasets, and balancing theory with application, while students often struggle with mathematical foundations, coding errors, and limited resources. However, these barriers can be addressed through collaborative teaching, pharmacy-centered case studies, guided exercises, visualization tools, and mini-projects. Overall, this course fosters critical thinking, problem-solving, and innovation by encouraging students to integrate AI/ML techniques with coding practice. By the end of the course, graduates are not only able to describe and differentiate AI/ML paradigms but also capable of implementing them in pharmacy-relevant scenarios. This forward-looking inclusion by PCI ensures that Indian pharmacy education remains globally competitive and prepares students for leadership in precision medicine and intelligent healthcare systems.

III. ACKNOWLEDGEMENT

We sincerely thank all the authors of the references cited in this article. Their valuable contributions to the field of pharmacy have provided significant insights and guidance, which greatly assisted us in preparing this work.

IV. DISCLAIMER

This article was drafted with the assistance of ChatGPT, an AI language model, and subsequently reviewed and refined by the authors.

V. REFERENCES

1. Abdel Aziz, M. H., Rowe, C., Southwood, R., Nogid, A., Berman, S., & Gustafson, K. (2024). A scoping review of artificial intelligence within

pharmacy education. *American Journal of Pharmaceutical Education, 88*(1)l: 100615.
2. Bücker, M., Hoti, K., & Rose, O. (2024). Artificial intelligence to assist decision-making on pharmacotherapy: A feasibility study. *Exploratory Research in Clinical and Social Pharmacy, 15:* 100491.
3. Chalasani, S., Syed, J., Madhan, R., Patil, V., & Kumar, T. (2023). Artificial intelligence in the field of pharmacy practice: A literature review. *Exploratory Research in Clinical and Social Pharmacy, 12:* 100346.
4. Jarab, A. S., Abu Heshmeh, S. R., & Al Meslamani, A. Z. (2023). Artificial intelligence (AI) in pharmacy: An overview of innovations. *Journal of Medical Economics, 26*(1): 1261–1265.
5. Kandhare, P., Kurlekar, M., Deshpande, T., & Pawar, A. (2025). Artificial intelligence in pharmaceutical sciences: A comprehensive review. *Medicine in Novel Technology and Devices, 27:* 100375.
6. Kaul, V., Enslin, S., & Gross, S. A. (2020). History of artificial intelligence in medicine. *Gastrointestinal Endoscopy, 92*(4): 807–812.
7. Lester, C., Rowell, B., Zheng, Y., Co, Z., Marshall, V., Kim, J. Y., Chen, Q., Kontar, R., & Yang, X. J. (2025). Effect of uncertainty-aware AI models on pharmacists' reaction time and decision-making in a web-based mock medication verification task: Randomized controlled trial. *JMIR Medical Informatics, 13:* e64902.
8. Mortlock, R., & Lucas, C. (2024). Generative artificial intelligence (Gen-AI) in pharmacy education: Utilization and implications for academic integrity: A scoping review. *Exploratory Research in Clinical and Social Pharmacy, 15:* 100481.
9. Simpson, M. D., & Qasim, H. S. (2025). Clinical and operational applications of artificial intelligenceand machine learning in pharmacy: A narrative review of real-world applications. *Pharmacy (Basel), 13*(2): 41.
10. Zhang, X., Tsang, C. C. S., Ford, D. D., & Wang, J. (2024). Student pharmacists' perceptions of artificial intelligence and machine learning in pharmacy practice and pharmacy education. *American Journal of Pharmaceutical Education, 88*(12): 101309.