Module Descriptions

Computer Science (B.Sc.)

March 6, 2025

Table of contents

1 Introduction to Computer Science

1.1 Learning Outcomes

After successful completion of the module, students are able to:

- Explain the fundamental concepts and definitions of computer science and its historical evolution.
- Describe the architecture and components of computer systems, including their functions.
- Recognize the basic functioning of various types of computer networks and their architectures.
- Apply basic computational thinking to model and solve problems using computational methods.
- Understand and evaluate the role of information systems in business contexts.
- Identify and assess database systems, management support systems, and business intelligence tools.
- Discuss the principles of business process management and entity-relationship modeling.
- Recognize the importance of data security and protection in business applications.

1.2 Module Content

Introduction to informatics

- Definition and scope
- Evolution and history

Technical foundations

- Computer systems

    - Evolution of computer systems
    - Coding of information (bits and bytes, ASCII, Unicode, ...)
    - The architecture of computer systems (von Neumann, Turing, ...)
    - Components of a computer system (CPU, RAM, Storage, ...)
    - Software (BIOS, OS, Firmware, Drivers, GUI, Applications)

- Computer networks

  - Evolution and objectives
  - Types of networks (LAN, WAN, GAN)
  - Network coordination types and topologies
  - Internet, Intranet, and Extranet
  - Cloud computing

- Software

  - Introduction to programming languages
  - History, concepts, features, and paradigms
  - Structure of programming languages: lexical, syntactic, and semantic
  - Open source and proprietary software
  - Features of coding versus low- and no-coding applications

Basics of computational theory

- Computational thinking
- Modeling and solving problems with computers
- Logic and algorithmic thinking

Introduction to business informatics

- Definition and scope
- Information and database systems in business
- Management support systems and business intelligence
- Business process management
- Entity–relationship model
- Data security and data protection in business

Literature

Weber et al. (2022), Xu & Zhang (2021), Cormen et al. (2022), Deitel & Deitel (2019).
Brookshear & Brylow (2019), Scott (2009)

References

Brookshear, J. G., & Brylow, D. (2019). *Computer science: An overview, global edition* (13th ed.). Pearson.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT press.

Deitel, P. J., & Deitel, H. M. (2019). *Intro to Python for computer science and data science.* Pearson.

Scott, J. C. (2009). *But how do it know?*

Weber, P., Gabriel, R., Lux, T., & Menke, K. (2022). *Basics in business informatics* (2nd ed.). Springer.

Xu, Z., & Zhang, J. (2021). *Computational thinking: A perspective on computer science.* Springer.

2 Introduction to Programming with Python

2.1 Learning Outcomes

After successful completion of the module, students are able to:

- Articulate the key features of Python and contrast it with other programming languages.
- Set up and utilize IDLE and other GUIs for script-based programming in Python.
- Demonstrate proficiency in using various data types, variables, and operators in Python.
- Apply control flow mechanisms and iterations to solve programming problems.
- Manipulate and work with lists, tuples, and dictionaries to manage data.
- Handle errors and exceptions to ensure robust program execution.
- Present in written and oral form the ability to create and utilize functions to modularize code effectively.
- Implement basic data analysis and visualization techniques using essential libraries like NumPy, Pandas, and Matplotlib.

2.2 Module Content

- The features of Python
- Comparison to other programming languages
- IDLE and other GUIs to work with Python scripts
- Introduction to fundamental programming concepts
- Variables, operators, expressions, data types, and basic input/output operations
- Selection and control flow mechanisms (if statements)
- Basic iterations (loops)
- Functions
- Lists, tuples, and dictionaries
- Errors and exceptions management
- Principles of object-oriented programming
- Introduction to essential libraries (e.g., NumPy, Pandas, Matplotlib) for data analysis and visualization
- AI tools for Python programming

Literature

Gagniuc (2023), Deitel & Deitel (2019), Ritza (2022), Charatan & Kans (2022), Igual & Seguí (2017)

References

Charatan, Q., & Kans, A. (2022). *Programming in two semesters: Using Python and Java.* Springer.

Deitel, P. J., & Deitel, H. M. (2019). *Intro to Python for computer science and data science.* Pearson.

Gagniuc, P. A. (2023). *An introduction to programming languages: Simultaneous learning in multiple coding environments.* Springer.

Igual, L., & Seguí, S. (2017). *Introduction to data science: A Python approach to concepts, techniques and applications.* Springer.

Ritza. (2022). *Learn Python the right way: How to think like a computer scientist.* Ritza. https://learnpythontherightway.com/

3 Mathematics for Computer Science

3.1 Learning Outcomes

After successful completion of the module, students are able to:

- Demonstrate the importance of mathematics and logical reasoning in solving problems in computer science.
- Define key concepts and operations in probability theory and Boolean algebra and explain their applications in computer science.
- Describe basic terminology and types of graphs in graph theory, and explain their relevance to computer science problems.
- Categorize matrix operations and evaluate their applications in various computational contexts.
- Define the fundamental concepts of automata theory and explain the differences between deterministic and nondeterministic finite automata.
- Explain real-world applications of mathematical concepts in computer science and categorize how those applications solve specific problems.

3.2 Module Content

- Importance of mathematics in computer science
- Overview of mathematical logic
- Introduction to probability theory, including Bayes' theorem and its applications
- Boolean algebra
- Introduction to graph theory and terminology
- Matrix algebra
- Introduction to automata theory
- Applications of mathematical concepts in computer science

Literature

Hartmann (2023), Hastie et al. (2009), Deisenroth (2020)

References

Deisenroth, M. P. (2020). *Mathematics for machine learning.* Cambridge University Press.

Hartmann, P. (2023). *Mathematics for computer scientists: A practice-oriented approach.* Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

4 Software Applications for Computer Science

4.1 Learning Outcomes

After successful completion of the module, students are able to:

- Understand the major operating systems and their functions.
- Distinguish between open-source and proprietary software.
- Utilize command-line interfaces (CLI) and shell commands effectively.
- Apply script-based workflows in software applications.
- Evaluate and compare no-code and low-code applications with traditional coding applications.
- Explain cloud computing concepts and identify deployment models (IaaS, PaaS, SaaS).
- Address privacy and security aspects of software applications.
- Utilize various software applications relevant to computer science, including IDEs, data science tools, and collaboration platforms.
- Implement version control practices using Git and GitHub.

4.2 Module Content

Introduction

- Overview of major operating systems (Windows, Linux, macOS, iOS, Android)
- Open-source and proprietary software
- Command-line interface (CLI) usage and shell commands
- The features of a script-based workflow
- No-code and low-code applications in comparison to coding applications
- Overview of cloud computing concepts and deployment models (IaaS, PaaS, SaaS)
- Privacy and security aspects of software applications
- Deploy and manage software across environments: Docker and containerization

Software Applications

- Editors and Integrated Development Environments (IDEs)
- Document preparation (documents, reports, presentations, websites, etc.)

- – Markup and style sheet languages(LaTeX, Markdown, Quarto, HTML, CSS)
- – No-code and low-code applications (Office suites, Canva, Prezi, etc.)
- Reference management software (e.g., BibTeX, JabRef, Zotero)

- Web-based writing assistance applications (Grammarly, DeepL, ChatGPT)
- Applications for data science (e.g., R, RStudio, SPSS, Stata, Power BI)
- Version control and collaboration using Git and GitHub/Gitlab, covering key concepts like branching and merging
- Other collaboration applications and platforms (e.g., Slack, MS Teams, Zoom, Kaggle)
- Cloud computing services (e.g., AWS, Google Cloud, Microsoft Azure, Posit Cloud)
- Overview of applications to share, sync, and back up data (e.g., Nextcloud, Dropbox, Google Drive)
- Overview of software applications for project management (e.g., JIRA, Trello)

Literature

quarto.org (2025), The LaTeX Project (2025), Tanenbaum et al. (2023), Git community (2025)

References

Git community. (2025). *Git.* online. https://git-scm.com/

quarto.org. (2025). *Welcome to quarto: An open-source scientific and technical publishing system.* online. https://quarto.org/

Tanenbaum, A., Tanenbaum, A. S., & Bos, H. (2023). *Modern operating systems. Global edition* (5th ed.). Pearson.

The LaTeX Project. (2025). *LaTeX: A document preparation system.* https://www.latex-project.org/

5 Software Engineering

5.1 Learning Outcomes

After successful completion of the module, students are able to:

- Demonstrate proficiency in applying fundamental software engineering principles to real-world scenarios.
- Interpret software requirements and accurately translate them into viable specifications.
- Distinguish between various software development methodologies and define their appropriate applications in differing project contexts.
- Describe key concepts in software architecture and design patterns, articulating their importance in system design.
- Explain the role of software testing and quality assurance in the development process, ensuring that standards are met.
- Categorize different types of software tests and evaluate their effectiveness in achieving quality assurance goals.
- Demonstrate effective use of version control systems to manage code collaboratively.
- Document, evaluate, and present software development processes.

5.2 Module Content

- Introduction to software engineering principles
- Software development life cycle and methodologies (Waterfall, V-Model, Agile)
- Requirements engineering and software specification
- Software architecture and design patterns
- Version control systems in software engineering (e.g., Git, GitHub/GitLab)
- Software testing and quality assurance
- AI applications for software engineering

    - AI-powered code assistance and generation tools
    - Automated testing and debugging
    - AI in software development lifecycle

- Documentation and maintenance of software
- Practical application through case studies and projects

Literature

Sommerville (2020), Pressman & Maxim (2019), Martin (2008), Fowler (2019), Bass et al. (2021)

References

Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison Wesley.

Fowler, M. (2019). *Refactoring: Improving the design of existing code* (2nd ed.). Addison Wesley.

Martin, R. C. (2008). *Clean code.* Prentice Hall.

Pressman, R., & Maxim, B. (2019). *ISE software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Sommerville, I. (2020). *Engineering software products: An introduction to modern software engineering* (Global Edition). Pearson.

6 Operating Systems and Computer Architectures

6.1 Learning Outcomes

After successful completion of the module, students are able to:

- Demonstrate an understanding of the evolution of operating systems and their historical context.
- Interpret the key features and functionalities of various operating systems, including Windows, GNU/Linux, macOS, iOS, and Android.
- Define the basic principles of computer architecture, including hardware components and their interactions.
- Describe CPU architecture, instruction execution, and the role of the memory hierarchy (cache, RAM, and storage).
- Explain the significance of assembly languages in system-level programming.
- Evaluate the structure and functionality of file systems, including various data types and their management.
- Describe the principles of input/output management and the relationship between hardware devices and software.
- Use virtualization technologies to execute various operating systems.
- Evaluate current trends in operating systems and computer architectures, and their implications for future developments.

6.2 Module Content

- Introduction to operating systems
    - Evolution of operating systems
    - Key features and differences among popular operating systems such as Windows, GNU/Linux, macOS, iOS, and Android
- System programming fundamentals
- Introduction to computer architectures
    - Overview of hardware components and computer architecture principles
    - CPU architecture and instruction execution
    - Memory hierarchy: Cache, RAM, and storage

- Introduction to assembly languages
- Process, thread, and memory management
- Structure and functionality of file systems and data types
- Input/output management
- Concurrency and synchronization
- Virtualization

  - The role and the benefits of virtual machines in modern computing
  - Types of virtualization: full, para-virtualization, and containerization
  - Software applications for virtualization (Docker, KVM, VMware, Virtualbox, Proxmox)

- Trends in operating systems and computer architectures

Literature Comer (2024), Fox (2024), Tanenbaum et al. (2023)

References

Comer, D. (2024). *Essentials of computer architecture* (3rd ed.). Chapman & Hall/CRC.

Fox, C. (2024). *Computer architecture.* No Starch Press.

Tanenbaum, A., Tanenbaum, A. S., & Bos, H. (2023). *Modern operating systems. Global edition* (5th ed.). Pearson.

7 AI Applications

7.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the role of AI applications in various industries and everyday life.
- Differentiate between general AI and narrow AI applications and their implications.
- Evaluate the impact of AI technologies on sectors such as e-commerce, healthcare, and education.
- Explain the functionalities and applications of advanced technologies like Natural Language Processing (NLP) and computer vision.
- Compare examples of AI applications in everyday life, including lifestyle enhancements and robotics.
- Categorize the various aspects of data security related to AI solutions.
- Present legal and ethical considerations surrounding the deployment of AI technologies.
- Identify future trends in AI and their potential implications for various careers.
- Demonstrate practical applications of AI concepts and tools relevant to their career aspirations.

7.2 Module Content

- Introduction and overview to popular AI applications
- Overview of general purpose AI models and narrow AI applications
- Technical foundations of AI applications

  - Core algorithms and architectures (e.g., neural networks, decision trees)
  - Data requirements and preprocessing
  - Hardware considerations (GPUs, TPUs, quantum computing)

- Advanced AI technologies

  - Natural Language Processing (NLP)
  - Computer vision and image processing
  - Robotics and autonomous systems
  - Computer vision

- Enhancing businesses (E-commerce, Healthcare, Education, etc.) with AI
- AI in everyday life

    – Personal assistants

    – Navigation and transportation

    – Entertainment

    – Social media

- Overview of AI applications for science
- Data management and security

    – Big data analytics and AI

    – AI and data security

- Legal and ethical considerations
- Future trends
- Practical applications and career opportunities

Literature

8 Programming with Java

8.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the role of Java in modern software development and its importance across various application domains.
- Differentiate between the core concepts of object-oriented programming and procedural programming paradigms.
- Evaluate the similarities and differences between Java and Python, particularly in terms of syntax, data types, and error handling.
- Explain the different data types in Java, including primitive types and their wrapper classes, and categorize input and output operations based on standard and file-based techniques.
- Compare various control structures in Java, such as loops and conditional statements, and present decision-making approaches using flowcharts or pseudocode.
- Contextualize the implementation of classes, objects, methods, and arrays within the framework of object-oriented programming principles in Java.
- Differentiate between packages and interfaces and explain their significance in modular programming and code organization in Java.
- Present real-world applications of Java in different fields, such as web development, desktop applications, and mobile app.

8.2 Module Content

- Introduction to Java
- Object-oriented programming principles
- Differences between Python and Java
- Software applications for Java programming
- Input, output, and data types
- Control structures
- Java classes, objects, arrays and methods
- Packages and interfaces
- Exception handling and debugging techniques
- Software and AI applications for Java programming
- Applications of Java

Literature Downey & Mayfield (2019), Liang (2021), Charatan & Kans (2022), Schildt & Coward (2024)

References

Charatan, Q., & Kans, A. (2022). *Programming in two semesters: Using Python and Java.* Springer.

Downey, A. B., & Mayfield, C. (2019). *Think Java: How to think like a computer scientist* (2nd ed.). O'Reilly Media.

Liang, Y. D. (2021). *Introduction to Java programming and data structures: Comprehensive version* (12th ed.).

Schildt, H., & Coward, D. (2024). *Java: A beginner's guide* (10th ed.). McGraw-Hill Education.

9 Databases (SQL)

9.1 Learning Outcomes

After successful completion of the module, students are able to:

- Explain the advantages, costs, and risks of various database approaches.
- Apply the Entity-Relationship diagrams to create a logical representation of the data for an organization.
- Define the key concepts and components of relational databases, including tables, relationships, and schemas.
- Differentiate between various types of databases, including relational and NoSQL, and assess their specific use cases and advantages.
- Evaluate database designs using normalization principles to ensure efficient data storage and retrieval.
- Explain the various aspects of SQL, including DDL, DML, DQL, DCL, and TCL, and how they are applied in managing databases.
- Compare the performance characteristics and use cases of relational databases versus NoSQL databases to determine the most suitable option for given requirements.
- Present database design and query solutions, demonstrating effective communication of technical concepts and outcomes to both technical and non-technical audiences.

9.2 Module Content

- Introduction to database management systems
- Entity-Relationship (ER) diagrams and database modeling
- Introduction to SQL

  - Relational databases
  - Data Definition Language (DDL)
  - Data Query Language (DQL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)
  - Transaction Control Language (TCL)

- Database design and normalization
- Introduction to NoSQL databases (e.g., document stores, key-value stores) and comparison with relational databases
- Indexing and query optimization

- Practical applications using a relational database system (e.g., MySQL, PostgreSQL, or SQL Server)
- Software and AI applications for database managment systems

Literature

Beaulieu (2020), Watson (2022), Domdouzis et al. (2021), Hoffer et al. (2020)

References

Beaulieu, A. (2020). *Learning SQL: Generate, manipulate, and retrieve data* (3rd ed.). O'Reilly Media.

Domdouzis, K., Lake, P., & Crowther, P. (2021). Concise guide to databases: A practical introduction. In *Cham: Springer Nature Switzerland* (2nd ed.). Springer.

Hoffer, J. A., Topi, H., & Venkataraman, R. (2020). *Essentials of database management.* Pearson.

Watson, R. T. (2022). *Data management: Databases and analytics: Open edition.* Prospect Press. https://www.richardtwatson.com/open/

10 Applications of C++

10.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the principles of object-oriented programming within the framework of C++ and its applications.
- Differentiate between C++, Java, and Python in terms of syntax, features, and use cases.
- Explain key C++ concepts such as classes, objects, inheritance, and polymorphism.
- Present effective debugging techniques and error analysis methods pertinent to C++ programming.
- Explain and compare various applications of C++ across different domains such as game development, system software, and robotics
- Present insights into advanced applications of C++, including blockchain technologies, high-performance computing, and machine learning.
- Evaluate the suitability of C++ for network programming, virtual reality, and augmented reality applications.

10.2 Module Content

- Introduction to Object-Oriented Programming and C++
- Differences between C++, Java, and Python
- Key C++ Concepts

    - Classes and Objects
    - Inheritance and Polymorphism
    - Memory Management and Move Semantics
    - Multiple Inheritance
    - Operator Overloading
    - Exception Handling
    - Input/Output in C++
    - Lambda Functions

- Error Analysis and Debugging Techniques
- Examples of Applications of C++

    - App and Game Development

- System Software
- Embedded Systems and Robotics
- Blockchain Technologies
- High-Performance Computing
- Graphical User Interfaces (GUIs)
- Database Management Systems
- Web Browsers
- Network Programming
- Machine Learning and Data Science
- Virtual Reality (VR) and Augmented Reality (AR)

- Software and AI applications for C++ programming

Literature

Will (2025), Lippman et al. (2012), Iglberger (2022)

References

Iglberger, K. (2022). *C++ software design: Design principles and patterns for high-quality software.* O'Reilly Media.

Lippman, S. B., Lippman, S., & Moo, B. (2012). *C++ primer* (5th ed.). Addison-Wesley Professional.

Will, T. T. (2025). *C++: The comprehensive guide.* Rheinwerk Computing.

11 Mobile Application Development

11.1 Learning Outcomes

After successful completion of the module, students are able to:

- Identify and judge the foundations of mobile app development for Android and iOS.
- Develop mobile applications using cross-platform development tools, such as Flutter.
- Prepare and develop mobile app interfaces that adhere to human-computer interaction and interface design principles.
- Prepare and conduct thorough app testing, debugging, and optimization.
- Present in written and oral form the process of publishing and distributing apps, explaining the steps and requirements for successful deployment on app stores.

11.2 Module Content

- Foundations of mobile app development for Android and iOS
- Human-computer interactions and interface design principles
- Mobile app design essentials
- Cross-platform development with Flutter
- Mobile app components and APIs
- App testing, debugging, and optimization
- Publishing and distributing apps

Literature

Mahagamage (2023), Flutter (2025), Apple.com (2025), Android.com (2025), Rogers et al. (2023)

References

Android.com. (2025). *Develop for Android.* https://developer.android.com/develop

Apple.com. (2025). *Apple developer documentation.* https://developer.apple.com/documentation/

Flutter. (2025). *Flutter documentation.* https://docs.flutter.dev/

Mahagamage, L. R. (2023). *Ultimate flutter handbook: Learn cross-platform app development with visually stunning UIs and real-world projects.* Orange Education.

Rogers, Y., Sharp, H., & Preece, J. (2023). *Interaction design: Beyond human-computer interaction* (6th ed.). Wiley.

12 Career Ecosystem in Computer Science

12.1 Learning Outcomes

After successful completion of the module, students are able to:

- Understand the structure and dynamics of the European job market.
- Identify key industry sectors and employment trends in Germany.
- Navigate the cultural aspects of working in an European business environment.
- Comprehend the legal framework surrounding employment in Germany.
- Prepare for job interviews in computer science.
- Develop strategies for effective job searching and networking in the German context.
- Analyze the impact of economic development, politics, and technological progress on career opportunities.
- Utilize tools and resources for career planning and development in Europe.
- Appreciate the benefits and handle the challenges of working in a multicultural environment.

12.2 Module Content

This course is designed to help computer science students develop a deep understanding of the German and Central European job market and career ecosystem. It covers various aspects of career development, including identifying suitable roles, building professional networks, and staying updated with emerging trends in the industry. Topics include:

- German and central european job market

  - Emerging technologies and industry trends for employability
  - Regional variations
  - Emerging sectors, clusters, and the profile of top companies

- Legal framework of employment

  - Employment laws and regulations
  - Visa and work permit requirements
  - Contractual and labor relations

- Career paths based on skills, interests, and market demand
- Personal branding, networking and job searching strategies
- Resume writing and cover letter preparation

- Job interview techniques and computer science specific interview questions
- Freelancing, consulting, and entrepreneurship options in computer science
- The role of globalization, outsourcing, and remote work on the job market
- Ethical and cultural considerations and professional responsibilities in the workplace
- Communication and teamwork skills in diverse work environments
- Mindset and continuous learning strategies for long-term career success

Literature

Pyak (2023), Staudacher (2021), Robbers (2023), Xu (2020), McDowell (2015), Orosz (2023), Brikman (2015)

References

Brikman, Y. (2015). *Hello, startup: A programmer's guide to building products, technologies, and teams.* O'Reilly.

McDowell, G. L. (2015). *Cracking the coding interview: 189 programming questions and solutions* (6th ed.). CareerCup.

Orosz, G. (2023). *The software engineer's guidebook: Navigating senior, tech lead, and staff engineer positions at tech companies and startups.* Pragmatic Engineer BV.

Pyak, C. (2023). *How to win jobs and influence germans: The expats' guide to a career in germany.* Immigrant Spirit.

Robbers, G. (Ed.). (2023). *An introduction to german law* (8th ed.). Nomos.

Staudacher, M. (2021). *Playing by the rules: German business culture.* WSPC.

Xu, A. (2020). *System design interview: An insider's guide.* Independently published.

13 Contributions to Computer Science

13.1 Learning Outcomes

After successful completion of the module, students are able to:

- Describe the origins and evolution of computing machines, programming languages, and computational theory including algorithms.
- Structure and discuss the key concepts, technologies, and challenges related to computer science with clarity and precision.
- Prepare independently a semester paper that presents and critically discusses either past advancements or contemporary challenges in computer science, demonstrating strong academic writing skills.

13.2 Module Content

This module explores the historical and contemporary developments and advancements in the field of computer science. It reviews a carefully chosen set of influential contributions that have shaped the discipline. Additionally, the module examines the most recent breakthroughs, providing an overview of the current topics and challenges that computer science is presently facing.

- Origins and early computing machines
- Turing machines, computability theory and algorithms
- The birth of modern computers
- The evolution of programming languages
- Operating systems and system software
- The internet and networking
- Computer graphics and visualization
- Human-computer interaction
- The rise of personal computing and smart devices
- Artificial intelligence and machine learning
- Parallel and distributed computing
- Computer security and cryptography
- Big data and data science
- Quantum computing and future technologies
- Social and ethical implications of computer science

Literature Petzold (2022), Lewis (2021), Springer (2025), Association for Computing Machinery (2025)

References

Association for Computing Machinery. (2025). *Journal of the ACM.* https://dl.acm.org/journal/jacm

Lewis, H. R. (2021). *Ideas that created the future: Classic papers of computer science.* The MIT Press.

Petzold, C. (2022). *Code: The hidden language of computer hardware and software* (2nd ed.). Microsoft Press.

Springer. (2025). *Lecture notes in computer science (LNCS).* https://www.springer.com/gp/computer-science/lncs

14 Data Science and Supervised Learning

14.1 Learning Outcomes

After successful completion of the module, students are able to:

- Describe the importance of data science in businesses and the implications of a data-driven environment, as well as judge the relevance and efficacy of various data science workflows and life cycles.
- Distinguish and discuss empirical research strategies to identify causal mechanisms, causes, and effects
- Describe the scope and definitions of supervised learning and its applications.
- Distinguish between statistical methods for supervised learning and explain their use cases in data analysis.

14.2 Module Content

Data science in practice

- Importance of data science in businesses and trends in a data-driven business environment
- Defining Business Intelligence (BI), Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL)
- Workflows and data science life cycles (e.g., OSEMN, CRISP-DM, Kanban, TDSP)
- Overview of various job roles for business analytics: data engineer, data analyst, machine learning engineer, business intelligence analyst, database administrator, data product manager, market research analyst, fraud analyst, …

Quantitative research design

- Techniques for measuring socio-economic and business phenomena
- Strategies for identifying causes of effects and effects of causes
- The fundamental problem of causal inference
- Techniques to establish causation: matching, natural experiments, field experiments, and laboratory experiments

Introduction to the statistical models of supervised learning

- Definition and scope of supervised learning

- Regression analysis

    - Simple linear regression
    - Multiple linear regression
    - Non-linear regression models
    - Model diagnostics, selection, and validation

- Estimators: Concept and applications

    - Ordinary Least Squares (OLS)
    - Weighted Least Squares (WLS)
    - Generalized Least Squares (GLS)
    - Maximum Likelihood Estimation (MLE)

- Time series analysis

    - Components of time series data (trend, seasonality, cyclicality)
    - Stationarity and unit root tests
    - Autoregressive (AR) and Moving Average (MA) models
    - ARIMA models

- Applications of classification and prediction techniques in economics, finance, and social sciences

Literature

Freeman & Ross (2018), Wickham et al. (2025), Igual & Seguí (2017), Cunningham (2021), Huntington-Klein (2022), Çetinkaya-Rundel & Hardin (2024), Békés & Kézdi (2021), Gelman et al. (2020), Bruce & Bruce (2020)

References

Békés, G., & Kézdi, G. (2021). *Data analysis for business, economics, and policy.* Cambridge University Press.

Bruce, P., & Bruce, A. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media.

Çetinkaya-Rundel, M., & Hardin, J. (2024). *Introduction to modern statistics* (2nd ed.). OpenIntro. https://openintro-ims.netlify.app/

Cunningham, S. (2021). *Causal inference: The mixtape.* Accessed January 30, 2023; Yale University Press. https://mixtape.scunning.com/

Freeman, M., & Ross, J. (2018). *Programming skills for data science.* Addison-Wesley Professional.

Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories.* Cambridge University Press.

Huntington-Klein, N. (2022). *The effect: An introduction to research design and causality.* CRC Press. https://theeffectbook.net

Igual, L., & Seguí, S. (2017). *Introduction to data science: A Python approach to concepts, techniques and applications.* Springer.

Wickham, H., Çetinkaya-Runde, M., & Grolemund, G. (2025). *R for data science (2e): Import, tidy, transform, visualize, and model data.* online. https://r4ds.hadley.nz/

15 Artificial Intelligence and Unsupervised Learning

15.1 Learning Outcomes

After successful completion of the module, students are able to:

- Describe the fundamental concepts and applications of AI.
- Discuss the ethical, legal, and technical implications of AI.
- Explain the concepts of deep learning, including neural network architectures and their applications in various fields.
- Investigate the techniques used in Natural Language Processing (NLP) and evaluate their applications in areas such as chatbots and language translation.
- Describe how machines interpret visual data in computer vision and analyze its applications, including image recognition and autonomous vehicles.
- Demonstrate knowledge of common unsupervised learning algorithms, including clustering, dimensionality reduction, and anomaly detection.
- Explain popular AI frameworks and libraries in Python and/or R to implement machine learning techniques.
- Solve problems related to data mining, including evaluating model performance and addressing challenges such as interpretability and sensitivity to parameters.

15.2 Module Content

Fundamentals

- Overview of Artificial Intelligence (AI) and its key components
- Exploration of various domains of AI (e.g., Natural Language Processing, Computer Vision, Robotics)
- Historical milestones in AI development (e.g., Turing Test, Neural Networks, Deep Learning breakthroughs)
- Review of applications across industries (e.g., healthcare, finance, transportation, entertainment, customer service)
- Ethics in AI
    - Discussion of bias in AI algorithms and data
    - Implications of privacy, accountability, and transparency

- Challenges in AI

  - Building a legal framework
  - Technical challenges in developing robust AI systems
  - Societal concerns such as job displacement and misinformation

- Future trends in AI

Core components of AI

- Machine Learning (ML)

  - Definition and differences between AI and ML
  - Types of machine learning: Supervised, Unsupervised, and Reinforcement Learning

- Deep Learning

  - Overview of neural networks and their architectures
  - Applications of deep learning in various fields

- Natural Language Processing (NLP)

  - Introduction to NLP and techniques used
  - Applications such as chatbots and language translation

- Computer vision

  - Basics of how machines interpret visual data
  - Applications in image recognition, video analysis, and autonomous vehicles

Unsupervised learning

- Overview of common algorithms such as decision trees, support vector machines, and gradient boosting
- Review of popular AI frameworks and libraries in Python and/or R.
- Classification
- Clustering
- Dimensionality reduction
- Association analysis
- Anomaly detection
- Applications of data mining and unsupervised learning
- Challenges in data mining

  - Difficulty in evaluating model performance
  - Interpretability of results
  - Sensitivity to parameter selection and data scaling

Literature James et al. (2021), James et al. (2023), Starmer (2022), Békés & Kézdi (2021), Bruce & Bruce (2020), Burkov (2019)

References

Békés, G., & Kézdi, G. (2021). *Data analysis for business, economics, and policy.* Cambridge University Press.

Bruce, P., & Bruce, A. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media.

Burkov, A. (2019). *The hundred-page machine learning book.* Andriy Burkov.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://www.statlearning.com

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python.* Springer. https://www.statlearning.com

Starmer, J. (2022). *The StatQuest illustrated guide to machine learning.* StatQuest Publications.

16 Applied Machine Learning

16.1 Learning Outcomes

After successful completion of the module, students are able to:

- Contextualize the importance of data preprocessing in machine learning for achieving accurate predictions.
- Differentiate and evaluate techniques for data cleaning, and explain various feature selection and engineering methods.
- Present the implementation of regression analysis for prediction, including model evaluation techniques and the application of regularization methods.
- Explain the process of implementing classification analyses utilizing decision trees and random forests and evaluate their effectiveness.
- Demonstrate the implementation of clustering analysis and the application of dimensionality reduction techniques, including Principal Component Analysis (PCA).
- Evaluate model performance through tuning techniques, including cross-validation methods and hyperparameter optimization strategies.

16.2 Module Content

In this course, students will learn how to implement statistical methods of machine learning using appropriate software packages.

- Python or R libraries for machine learning
- Data Preprocessing

    - Data cleaning: handling missing values, outliers, and duplicates
    - Feature selection and engineering
    - Data normalization and standardization
    - Data splitting: training and testing sets

- Implementation of regression analysis for prediction, including model evaluation techniques and regularization
- Implementation of classification analysis with decision trees and random forests
- Implementation of clustering analysis
- Implementation of dimensionality reduction techniques, including Principal Component Analysis (PCA)

- Model evaluation and tuning, including cross-validation techniques and hyperparameter optimization

Literature

VanderPlas (2022), Irizarry (2024), VanderPlas (2022), Géron (2022), James et al. (2023), James et al. (2021), James et al. (2023), Bruce & Bruce (2020)

References

Bruce, P., & Bruce, A. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media.

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly Media.

Irizarry, R. A. (2024). *Advanced data science: Statistics and prediction algorithms through case studies.* http://rafalab.dfci.harvard.edu/dsbook-part-2/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://www.statlearning.com

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python.* Springer. https://www.statlearning.com

VanderPlas, J. (2022). *Python data science handbook: Essential tools for working with data* (2nd ed.). O'Reilly. https://jakevdp.github.io/PythonDataScienceHandbook/

17

17.1 Learning Outcomes

After successful completion of the module, students are able to:

17.2 Module Content

Literature