

L'Intelligenza Artificiale nell'Insegnamento delle Tecnologie Informatiche

Contenuti

Il corso si propone di introdurre all'utilizzo dell'intelligenza artificiale generativa (GenAI) nello sviluppo del software, illustrando come, nella didattica delle discipline informatiche, l'utilizzo di tali strumenti debbano sempre più convivere con metodologie di insegnamento tradizionali.

Finalità

Le finalità principali del corso sono:

- Capire il funzionamento degli strumenti di GenAI
- Comprendere come, ad oggi, gli strumenti di GenAI sono utilizzati (e non) nella pratica dello sviluppo software
- Apprendere i possibili pattern educativi nell'utilizzo di GenAI per lo sviluppo di software
- Conoscere i limiti, problemi etici e legali legati all'uso di GenAI per lo sviluppo di software

Indice

- Introduzione ai modelli transformer e ai Large Language Models (LLM)
- Training, fine-tuning, e prompting di LLM
- Utilizzo di LLM nello sviluppo di software: strumenti e applicazioni
- Utilizzo di LLM nella didattica informatica: risultati di studi empirici
- Pattern di utilizzo di LLM nell'insegnamento dell'informatica, e principali tecniche di prompting
- AI-assisted code review/improvement
- Aspetti etici e legali nell'uso di AI generativa nella didattica e in ambienti di produzione

Lecture consigliate

1. Appunti delle lezioni.
2. Bowen, J. A., & Watson, C. E. (2024). *Teaching with AI: A practical guide to a new era of human learning*. Johns Hopkins University Press
3. Porter, L., & Zingaro, D. (2024). *Learn AI-Assisted Python Programming: With GitHub Copilot and ChatGPT* (2nd ed.). Manning. ISBN 9781633435995

Fondamenti di programmazione

Contenuti

Il corso intende fornire ai partecipanti le basi teoriche e pratiche per l'insegnamento della programmazione nella scuola, con particolare attenzione al pensiero computazionale, all'interdisciplinarietà e alla progettazione di attività didattiche efficaci. I partecipanti svilupperanno una comprensione critica dei linguaggi di programmazione per la didattica, dei modelli cognitivi coinvolti nell'apprendimento della programmazione e delle strategie pedagogiche per introdurre algoritmi e strutture dati in contesti scolastici.

Finalità

1. Comprendere e applicare il pensiero computazionale
2. Integrare il pensiero computazionale nelle discipline scolastiche
3. Conoscere e utilizzare linguaggi di programmazione didattici
4. Comprendere i modelli mentali nella didattica della programmazione
5. Introduzione degli approcci di didattica degli algoritmi

Indice

- Pensiero computazionale e didattica trasversale
- Applicazioni del pensiero computazionale in diverse discipline (arte, letteratura, fisica)
- Didattica della programmazione
- Modelli Mentali nella Didattica della Programmazione
- Linguaggi di Programmazione per la didattica
- Didattica degli algoritmi e delle strutture dati

Lecture consigliate

Dispense e lucidi del docente

1. Maureen D. Neumann, Lisa Dion, Robert Snapp
Insegnare il pensiero computazionale. Un approccio integrato per l'apprendimento nella scuola secondaria di primo e secondo grado Roma: Edizioni Anicia, 2024
2. Shuchi Grover (Editor) *Computer Science in K-12: An A-To-Z Handbook on Teaching Programming* Edfinity, 2020
3. Orit Hazzan, Noa Ragonis, Tami Lapidot
Guide to Teaching Computer Science: An Activity-Based Approach Springer, 2020 (3^a edizione)

Fondamenti di Didattica nell'Informazione

Contenuti

Il corso è incentrato sulle tematiche principali riguardanti la didattica dell'informatica.

Finalità

La finalità principale del corso è porre i corsisti nella condizione di progettare e attuare interventi formativi sui principali argomenti riguardanti l'informatica nelle scuole secondarie superiori **Indice**

- Introduzione, struttura e obiettivi del corso
- La didattica dell'informatica
- La didattica dei linguaggi di programmazione
- Cenni sui paradigmi di apprendimento
- Approccio top-down e bottom-up
- Imparare facendo
- Progettazione didattica
- Didattica della programmazione
- La valutazione

Lecture consigliate

1. Appunti e dispense del docente

Strumenti avanzati per la didattica dell'informazione

Contenuti

Il corso verte sull'insegnamento di strumenti e metodologie avanzate per la didattica dell'informazione prendendo spunto dal "cooperative working" in ambito di sviluppo software. Oltre alle tecniche moderne di gestione del codice e revisione automatica, applicabili sia nella pratica professionale che nel peer programming, il corso integra lo studio della comunicazione dei dati. Verrà mostrato come presentare i numeri in modo efficace e, soprattutto, come riconoscere il "bullshit".

Finalità

Le finalità principali del corso sono:

- Saper organizzare e partecipare a progetti di sviluppo software collaborativo tramite Git e GitHub, applicando flussi di lavoro moderni (branching, pull request, code review).
- Utilizzare strumenti di analisi statica del codice e processi di sviluppo agili con l'ausilio dei GitHub workflows.
- Utilizzare notebook interattivi per la programmazione dal vivo e per la didattica.
- Sviluppare pensiero critico sui dati, imparando a distinguere tra l'uso corretto delle statistiche e la manipolazione informativa.
- Progettare presentazioni efficaci, apprendendo come visualizzare i numeri e i grafici per comunicare informazioni complesse senza distorcere la realtà.

Indice

- Cenni su cooperative working per lo sviluppo del software
- Introduzione al Software Configuration Management (SCM)
- Tracciamento delle modifiche e sistemi di versioning
- Utilizzo Avanzato di Git: fork, rebase, cherry-pick, e strategie di branching, gestione dei conflitti e merge

- Issue report: struttura e qualità
- Processo di code review: buone pratiche e strumenti
- Automatizzazione del processo di code review: tool di analisi statica
- Sviluppo software basato su Pull Request
- Introduzione ai webhook e integrazione con GitHub

- Processi di sviluppo agili
- Introduzione alla Continuous Integration
- GitHub Actions: sintassi e concetti di base
- Creazione e test di workflow personalizzati

- Introduzione ai Jupyter Notebook
- Live programming: vantaggi e rischi
- Pensiero critico sui dati
- Principi di design dell'informazione; progettare slide e grafici che comunicano con chiarezza e integrità
- L'arte di mentire con le statistiche: Analisi di casi reali di manipolazione

Lecture consigliate

- Lucidi del corso
- Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson
- Stephen Few, Show Me the Numbers: Designing Tables and Graphs to Enlighten, Analytics Press
- Darrell Huff, How to Lie with Statistics, W. W. Norton & Company

Metodologie per la didattica dell'informazione

Contenuti

Il corso si concentra sull'insegnamento dei fondamenti teorici, metodologici e operativi per progettare, realizzare e valutare percorsi didattici efficaci in ambito informatico. Particolare attenzione è riservata alla didattica attiva, alla rappresentazione concettuale dei contenuti e alla valutazione dell'apprendimento, con un'integrazione tra informatica, pedagogia e tecnologie didattiche.

Finalità

Le finalità principali del corso:

- Comprendere le cornici teoriche della didattica dell'informatica.
- Saper analizzare e strutturare contenuti concettualmente complessi.
- Progettare percorsi formativi fondati su approcci attivi e cooperativi.
- Scegliere e utilizzare strumenti didattici digitali e ambienti di apprendimento.
- Valutare efficacemente l'apprendimento in contesti di coding e progettazione algoritmica.
- Adottare strategie inclusive per una didattica attenta alla diversità cognitiva.

Indice

- Cornici teoriche della didattica dell'informatica
- Approcci pedagogici (comportamentismo, costruttivismo, costruzionismo, connettivismo)
- Tassonomie educative (Bloom, SOLO) e framework per le competenze digitali (DigCompEdu)
- Analisi del dominio concettuale dell'informatica
- Strumenti per la mappatura concettuale (mappe, graph knowledge)
- Progettazione di percorsi didattici: obiettivi, attività, risultati attesi
- Didattica attiva: coding, debugging, attività unplugged, pair programming
- Ambienti digitali per l'insegnamento del coding (Snap!, Thonny, Replit, Jupyter)
- Strumenti per la gestione e il monitoraggio dell'apprendimento (LMS, CodeRunner, Moodle Quiz)
- Metacognizione e pensiero computazionale (astrazione, decomposizione, generalizzazione, valutazione)
- Valutazione formativa e sommativa in ambito informatico
- Costruzione di rubriche per attività di programmazione
- Strategie inclusive (UDL, attenzione a DSA, ADHD, gifted)
- Prospettive di ricerca nella didattica dell'informatica
- Progetti e iniziative internazionali (Bebras, CSTA, Informatics Europe)

Lecture consigliate

1. Appunti delle lezioni.
2. ACM Curriculum Guidelines for Computer Science Education.
3. Informatics Europe – Informatics for All.
4. Bruner, J. – La cultura dell'educazione
5. Papert, S. – Mindstorms: Children, Computers, and Powerful Ideas.
6. Wing, J. – Computational Thinking.

Metodologie avanzate per la didattica dell'informazione

Contenuti

Il corso verte sull'insegnamento dei principi fondamentali dell'informazione nei suoi elementi e strategie di pensiero essenziali partendo da una introduzione del contesto tecnologico e delle sue premesse. In seguito, il corso elabora sistematicamente alcune fra le fondamentali strategie di apprendimento *attivo* e tramite *serious-gaming*. Il corso include attività di apprendimento pratico e verifica automatica dell'apprendimento.

Finalità

Le finalità principali del corso:

- Confrontare criticamente le principali teorie e modelli di didattica dell'informazione e spiegare come ciascuno supporti diversi risultati di apprendimento cognitivi, affettivi e metacognitivi.
- Selezionare e giustificare metodologie pedagogiche appropriate, come l'apprendimento basato sulla ricerca, i formati di classe capovolta, gli approcci basati su problemi e progetti, per una varietà di contesti di apprendimento ricchi di informazioni.
- Valutare le implicazioni etiche, di accessibilità e inclusività delle tecnologie e delle metodologie didattiche avanzate nell'insegnamento delle competenze informative mediante strumenti di gioco e integranti tecnologie avanzate di tipo playground.
- Selezionare e motivare l'uso di metodologie pedagogiche (inquiry-based, flipped classroom, problem/project-based, ecc.) in diversi contesti ricchi di informazione e valutarne l'impatto cognitivo.
- Valutare implicazioni etiche, di accessibilità e inclusione nell'adozione di tecnologie didattiche avanzate in combinazione con il lavoro di gruppo e i serious gaming.
- Implementare piani di valutazione basati sull'evidenza (rubriche, dashboard, diari di riflessione) per misurare e iterare l'acquisizione di competenze informatiche.

Indice

- Progettazione unità didattiche incentrate sullo studente che integrino strategie avanzate e tecnologicamente avanzate (ad esempio, analisi dell'apprendimento, contenuti adattivi, XR o attività basate su giochi) per promuovere l'alfabetizzazione informativa e la fluidità dei dati.
- Implementazione piani di valutazione basati sull'evidenza che utilizzino strumenti formativi e sommativi (rubriche, dashboard di analisi, diari di riflessione, ecc.) per misurare e iterare l'acquisizione delle competenze informative.
- Progettazione unità didattiche learner-centred che integrino strategie tecnologiche avanzate (learning analytics, contenuti adattivi, XR, gamification) per promuovere information & data literacy.

- Produzione e presentare un prototipo didattico completo (lezione, modulo o microcorso) che dimostri la padronanza di tecniche avanzate di progettazione didattica, implementazione e valutazione.
- Implementazione piani di valutazione evidence-based usando strumenti formativi e sommativi (rubriche, dashboard analitici, diari riflessivi) per misurare e iterare l'acquisizione di competenze informative.

Lettere consigliate

1. Orit Hazzan, Tami Lapidot, and Noa Ragonis. 2011. *Guide to Teaching Computer Science: An Activity-Based Approach* (1st. ed.). Springer Publishing Company, Incorporated.
2. Miller, P., Pane, J., Meter, G. & Vorthmann, S. (1994). Evolution of Novice Programming Environments: The Structure Editors of Carnegie Mellon University. *Interactive Learning Environments*, 4, 140--158. doi: 10.1080/1049482940040202.
3. Fischer, F., Kollar, I., Mandl, H. & Haake, J. M. (eds.) (2007). *Scripting ComputerSupported Collaborative Learning: Cognitive, Computational and Educational Perspectives*.