

Consegna 1

Domande parte introduttiva

Lorenzo Dentis, lorenzo.dentis@edu.unito.it

8 marzo 2023

1 Domande

Rispondere alle seguenti domande. Le risposte devono essere motivate citando le letture indicate su moodle. Dovete esprimere la vostra opinione rispetto alle posizioni espresse dagli articoli degli autori: siete d'accordo? se sì/no perchè?

1. E' importante insegnare informatica come materia scolastica?
2. L'informatica è una scienza?
3. Qui sotto trovate una lista di criteri estratti dall'articolo The Science in Computer Science. Sono usati dall'autore per definire la credibilità di un settore come "scienza". Siete d'accordo con la scelta di questi criteri? Motivare le risposte: Organized to understand, exploit, and cope with a pervasive phenomenon.
 - Encompasses natural and artificial processes of the phenomenon.
 - Codified structured body of knowledge.
 - Commitment to experimental methods for discovery and validation.
 - Reproducibility of results.
 - Falsifiability of hypotheses and models.
 - Ability to make reliable predictions, some of which are surprising.

L'informatica soddisfa qualcuno dei criteri sopra elencati? Se sì quali? e perchè?

2 Risposte

2.1 È importante insegnare informatica come materia scolastica?

Secondo me insegnare l'informatica nelle scuole è importante, ma ancor più importante è insegnare le competenze digitali.

Ogni cittadino dovrebbe essere in grado di operare un computer o device digitale, perchè ci stiamo muovendo verso una società che per svolgere anche compiti necessari alla basilare sopravvivenza richiede di utilizzare dispositivi informatici. Saper utilizzare correttamente un computer e navigare su internet sta diventando una competenza basilare tanto quanto leggere e scrivere.

Parlando invece di *Informatica* invece credo che fornire delle basi sia importante.

Trovo che l'Informatica come materia scolastica abbia molto in comune con la Storia, in quanto, anche nel caso in cui non rilevante nella crescita accademica di un individuo, è fondamentale per capire il mondo intorno a noi. Il manifesto per l'umanesimo digitale parla molto di come le tecnologie dovrebbero essere adattate a supporto della libertà e della democrazia, ma secondo me deve esserci anche una modifica del comportamento umano nei confronti della tecnologia.

Non si può avere democrazia senza conoscenza. Anche presupponendo di avere un prodotto perfetto e costruito con le migliori intenzioni, che rispetta la privacy dell'utente e garantisce imparzialità, tale strumento è inutile se il suo utilizzatore non si fida e non ci può essere fiducia se prima lo strumento non viene compreso. Un buon esempio può essere il voto elettronico. Se anche venisse realizzata una piattaforma online perfetta poi bisognerebbe convincere le persone ad usarla. Io stesso, come immagino chiunque altro, mi rifiuterei di votare se questa fosse una "black box" dal funzionamento sconosciuto.

Quindi concordo con l'affermazione **"L'educazione all'informatica e al suo impatto sociale devono iniziare il prima possibile"**, perchè bisogna fornire ai futuri cittadini le competenze per utilizzare e soprattutto comprendere il funzionamento dei nuovi strumenti tecnologici che permeano tutti gli aspetti della nostra vita.

2.2 L'informatica è una scienza?

Sì, come affermato durante le lezioni ciò dipende molto da cosa si considera scienza e cosa no. Io mi rifaccio alla definizione data da Galilei, cioè che una disciplina è scienza quando rispetta il metodo scientifico. Quindi una disciplina che segue le seguenti fasi: osservazione di un fenomeno, misura del fenomeno, formulazione di una ipotesi, preparazione di un esperimento che provi l'ipotesi e che sia ripetibile. Nel caso dell'informatica ovviamente l'oggetto di studio è una informazione.

In particolare io "traccio la linea di demarcazione" tra scienza e non scienza sulla ripetibilità di un esperimento. Ad esempio non considero scienza le scienze sociali o l'economia, perchè gli esperimenti non possono essere effettuati "in ambiente sterile" ed i risultati ottenuti da differenti esperimenti non saranno mai identici, perchè è impossibile partire dalle stesse condizioni di partenza.

In informatica è invece partendo dalle stesse condizioni iniziali (che sono com-

pletamente controllabili) ed eseguendo lo stesso algoritmo si giunge sempre alle stesse conclusioni, l'informatica è deterministica.

2.3 Quali criteri soddisfa l'informatica?

Encompasses natural and artificial processes of the phenomenon. Come dice il documento di Peter J. Denning l'argomentazione "l'informatica studia solo processi artificiali" cade quando si considera il fatto che l'informatica è la disciplina che studia le informazioni, non le tecnologie. Si potrebbe dibattere sul fatto che tutte le informazioni trattate dall'informatica derivano da fenomeni artificiali, anche le previsioni del tempo di un algoritmo meteorologico non sono lo studio dell'evento meteorologico in se, ma lo studio della simulazione dello stesso. Ciò non toglie che "le informazioni" non sono per forza artificiali, anche una reazione chimica porta con se delle informazioni, quindi io credo che l'informatica soddisfi questo criterio, seppur non strettamente.

Codified structured body of knowledge. La conoscenza dell'informatica è basata su principi teorici, modelli matematici e algoritmi che sono stati sviluppati e codificati nel corso degli anni. Basti pensare a tutti gli studi di informatica teorica quali algoritmi, complessità e calcolabilità, ma io includerei anche argomenti come i sistemi operativi, le reti, la sicurezza informatica, l'analisi dei dati, l'intelligenza artificiale e molti altri.

Commitment to experimental methods for discovery and validation. Questo è un altro punto su cui si può dibattere, in quanto l'informatica di uso quotidiano non fa uso di esperimenti. Quando viene scritto un software difficilmente viene prima validato (nonostante ci siano strumenti per farlo). D'altra parte si potrebbe considerare l'uso da parte dell'utente come una sperimentazione, in quanto il software viene costantemente aggiornato, una specie di sperimentazione sul campo.

In ogni caso questa argomentazione decade quando si pensa a software sviluppato per scopo di ricerca. Mi viene in mente l'esempio del Q-learning, algoritmo di apprendimento automatico che si basa proprio sull'idea di effettuare molte azioni differenti e trovare quella che fornisce un risultato migliore in fase di addestramento, cioè esattamente sperimentare differenti strategie finché non si trova quella migliore.

Reproducibility of results. Ho ampiamente discusso di questo punto nella domanda precedente. Partendo da un ambiente controllato, situazione molto semplice da produrre in informatica dato che siamo noi a generare gli input, gli algoritmi utilizzati produrranno output prevedibili.

Falsifiability of hypotheses and models. Come ho accennato in precedenza ci sono metodi formali per la validazione del software, che poi questi non vengano usati

è un altro discorso. Il documento di Peter J. Denning fa lo stesso ragionamento (portando a supporto di questa tesi il fatto che circa il 50% dei modelli e delle ipotesi proposte non erano state rigorosamente testate). Direi che l'informatica soddisfa questo requisito se non fosse per alcuni problemi indecidibili, primo tra tutti l'*halting problem* di Turing. In informatica (come in matematica) ci sono dei problemi indecidibili, di conseguenza non tutte le ipotesi ed i modelli sono verificabili falsi. Quindi a mio parere l'informatica non soddisfa questo criterio.

Ability to make reliable predictions, some of which are surprising. Riguardo al fatto che l'informatica permetta di effettuare predizioni ho già speso molte parole. Mi interrogo sul perchè una scienza debba per forza ottenere risultati sorprendenti, dato che questo è un metro di valutazione soggettivo, ma in ogni caso direi che l'informatica permette di fare scoperte sorprendenti o quantomeno non intuitive. I risultati delle scoperte informatiche hanno radicalmente cambiato il mondo forse più di quanto qualsiasi altra scienza abbia fatto. Un esempio di predizione sorprendente a mio parere è la teoria della complessità. Un problema semplice come le torri di Hanoi quando viene esteso sopra una certa soglia richiede un tempo computazionale sorprendente, cosa che non ci si aspetterebbe.