UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Foundations of Computer Science

2627-1-FDS02Q009

Obiettivi

Al termine del corso lo studente sarà in grado di interrogare e di inferire la struttura implicita della base di dati a partire dalle tabelle che la compongono.

Inoltre lo studente saprà scrivere e correggere gli errori di semplici programmi in Python per la gestione e l'analisi di dataset formati da poche tabelle interconnesse, come quelle disponibili a kaggle.com (formati CSV, TSV, JSON). In particolare, le analisi verranno fatte utilizzando il Jupyter Notebook.

Infine, lo studente potrà interrogare database relazionali usando SQL, sia come linguaggio separato, sia come chiamata in un programma Python.

Conoscenza e capacità di comprensione

Questo insegnamento fornisce le conoscenze basilari e capacità di comprensione relativamente a:

- Database.
- SQL
- Programmazione in Python
- Notebook Jupyter
- Gestione di dati tabellari con Pandas

Conoscenza e capacità di comprensione applicate

Alla fine dell'insegnamento gli studenti saranno in grado di:

- Scrivere interrogazioni in SQL
- Scrivere notebook in Python per gestire dataset
- Scrivere notebook in Python per analizzare e pulire dataset

Autonomia di giudizio

Gli studenti saranno in grado di comprendere autonomamente la qualità dei dati da gestire ed effettuare un'analisi esplorativa usano un notebook Jupyter

Abilità comunicative

Gli studenti saranno in grado di descrivere i processi decisionali che li hanno portati a gestire, pulire, analizzare un dataset.

Capacità di apprendere

Gli studenti saranno in grado di apprendere nuovi linguaggi di programmazioni dedicati all'analisi statistica, in particolare R.

Contenuti sintetici

Organizzazione di dati grezzi: file system, file con delimitatori.

Introduzione ai database. Il modello relazionale dei dati e linguaggio SQL. Select ... From … Where.

Introduzione alla programmazione in Python. Programmazione esplorativa. Gestione dati in formato tabellare.

Cenni di test e debugging.

Programma esteso

1. Organizzazione di dati grezzi
2. file, directory, tipi di file
3. principali comandi da terminale
4. file con delimitatori.
5. Introduzione ai database.
6. Il modello relazionale dei dati
7. Il linguaggio SQL: Select ... From … Where su una tabella.
8. Interrogazioni su più tabelle.
9. Introduzione alla programmazione in Python.
10. Array, liste, dizionari
11. Cicli
12. Organizzazione di un programma: funzioni
13. Librerie
14. Programmazione esplorativa. Gestione dati in formato tabellare.
15. Jupyter Notebook
16. Pandas
17. Cenni di test e debugging.

Prerequisiti

Conoscenza di un linguaggio di programmazione

Modalità didattica

Lezioni e esercizi al computer.
Tutte le lezioni prevedono una prima parte (circa il 20% del tempo) dove vengono spiegati i contenuti in modalità erogativa, seguita da una parte interattiva basata sullo svolgimento di esercizi.
Le lezioni non vengono registrate nè trasmesse in streaming.

Materiale didattico

- Downey Think Python 2e: How To Think Like a Computer Scientist
- Downey Think Stats 2e
- Downey Elements of Data Science. This book is a bit too basic.
- VanderPlas, Python Data Science Handbook
- Elmasri, Navathe, Fundamentals of Database Systems

Periodo di erogazione dell'insegnamento

Primo semestre (prime 6 settimane)

Modalità di verifica del profitto e valutazione

L'esame consiste di tre parti: uno scritto, un progetto da sviluppare in piccoli gruppi (da 1 a 3 persone) e una prova orale. Tutte le parti devono essere sostenute in lingua inglese.

Esame scritto

Prevede quattro domande a risposta aperta, ciascuna consistente in una query SQL. Vengono valutate la completezza e la correttezza delle risposte. Se lo scritto risulta insufficiente, non è possibile presentare il progetto.

Progetto (in Python)
Il progetto integra tutte le competenze acquisite nell'insegnamento. Il testo della prova è unico per tutti gli appelli fino a settembre 2027.

Modalità di consegna: Successivamente al superamento dello scritto, e comunque entro un mese dalla data dello stesso (e almeno 3 giorni prima della prova orale, se sostenuta), il progetto deve essere caricato pubblicamente su GitHub sotto forma di un singolo notebook Jupyter. Il link al repository va comunicato tramite un apposito form online.

Utilizzo di LLM/Generative AI: È ammesso, ma il notebook deve contenere una sezione dedicata che descriva come tali strumenti siano stati utilizzati. Il progetto viene valutato per correttezza, completezza e originalità (specialmente rispetto alle soluzioni standard generate dalle AI).

Se la discussione evidenzierà utilizzi di AI non dichiarati, verrà assegnato un nuovo progetto e la prova scritta sarà annullata.

Prova orale

Deve essere sostenuta entro un mese dallo scritto e consiste in alcune domande sul progetto e su tutti gli argomenti trattati a lezione, volte a verificare la comprensione del programma.
E' possibile scegliere di sostenere la prova orale solo con domande sul progetto (quindi senza domande sugli argomenti trattati a lezione): chi sceglie questa modalità può ottenere al massimo 24/30.
Un esito insufficiente all'orale rende insufficiente l'intera prova d'esame.

Valutazione finale

Il voto di partenza è calcolato sulla media geometrica (arrotondata per difetto) tra il voto dello scritto e quelli delle tre componenti del progetto (ciascuno espresso da 0 a 30). La prova orale modifica questo voto di un massimo di 3 punti, sia in aumento sia in diminuzione.

È obbligatorio iscriversi allo scritto tramite la segreteria online. Non saranno ammesse eccezioni.

Orario di ricevimento

Il ricevimento è online e su appuntamento a https://www.unimib.it/gianluca-della-vedova

Sustainable Development Goals

ISTRUZIONE DI QUALITÁ