UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Machine Learning for Finance

2627-1-F1602M003

Obiettivi formativi

Conoscenza e capacità di comprensione. Il corso si propone di fornire allo studente una solida competenza nella programmazione in Python e i concetti fondamentali dell'elaborazione dei dati e dell'apprendimento automatico: la rappresentazione dei dati con vettori, matrici e tabelle, i modelli che apprendono dai dati e le reti neurali artificiali.

Conoscenza e capacità di comprensione applicate. Al termine del corso lo studente sarà in grado di scrivere programmi in Python che leggono ed elaborano dati finanziari ed economici, di simulare fenomeni aleatori, di analizzare dati tabellari con le librerie NumPy e pandas, di rappresentarli graficamente e di costruire e addestrare semplici reti neurali con PyTorch.

Autonomia di giudizio. Lo studente imparerà a scegliere in modo autonomo, tra gli strumenti e le tecniche apprese, quelli più adatti a risolvere problemi pratici di moderata complessità legati alla manipolazione e all'analisi di dati in ambito finanziario, e a valutare criticamente la correttezza dei risultati ottenuti.

Abilità comunicative. Lo studente sarà in grado di descrivere in modo chiaro e rigoroso, sia oralmente sia per iscritto, algoritmi, programmi e modelli di apprendimento automatico, utilizzando la terminologia appropriata e scrivendo codice leggibile e ben documentato.

Capacità di apprendere. Lo studente acquisirà le competenze necessarie per un apprendimento autonomo e continuo nel campo della programmazione e dell'analisi dei dati: in particolare, sarà capace di consultare la documentazione ufficiale dei linguaggi e delle librerie per approfondire nuovi strumenti e applicarli a problemi diversi da quelli affrontati nel corso.

Contenuti sintetici

Il corso fornisce una solida competenza nella programmazione in Python, dai fondamenti del linguaggio alle librerie per il calcolo numerico (NumPy), per l'analisi dei dati tabellari (pandas) e per i grafici (matplotlib), e presenta le reti

neurali artificiali, realizzate con PyTorch, come strumento di supporto all'analisi dei dati finanziari.

Programma esteso

1. Algoritmi e programmi: definizione e proprietà di un algoritmo; problemi, istanze, dati e risultati; dall'algoritmo al programma; paradigmi di programmazione.

2. L'ambiente di sviluppo PyCharm: l'interprete Python, progetti e file, esecuzione di un programma, la Python Console, il debugger. Il lessico del linguaggio: parole chiave e identificatori.

3. Tipi di dato e collezioni: numeri interi e in virgola mobile, booleani, stringhe, None, conversioni tra tipi; espressioni aritmetiche, relazionali e logiche; liste, tuple, insiemi e dizionari.

4. Istruzioni composte: la selezione (if, elif, else), le iterazioni for e while, le istruzioni break e continue.

5. Numeri pseudo-casuali e simulazione: il modulo random, distribuzioni discrete e distribuzione normale, simulazione di fenomeni aleatori, processi di Markov.

6. File di testo: apertura e lettura con open e with; file con più campi per riga; righe di intestazione; costruzione di tabelle a partire dai dati letti; parametri letti da file o da tastiera.

7. Funzioni: definizione e chiamata, annotazioni di tipo, parametri e argomenti, valori restituiti, variabili locali e globali, passaggio dei parametri, ricorsione, moduli e import.

8. La libreria NumPy: vettori e matrici, indicizzazione e sottomatrici, operazioni componente per componente e operazioni vettoriali, funzioni di sintesi, maschere booleane, elementi di algebra lineare, lettura di matrici da file, numeri casuali; vettori e matrici nell'apprendimento automatico: la regressione lineare e la discesa del gradiente.

9. La libreria pandas: Series e DataFrame; lettura dei dati con read_csv e read_excel; selezione di righe e colonne con loc, iloc e maschere; etichette comuni (isin, intersection); operazioni aritmetiche e relazionali e allineamento delle etichette; valori di sintesi; aggregazione con groupby e agg; trasposta; ordinamento, aggiunta, ridenominazione ed eliminazione di righe e colonne; unione di tabelle (concat, merge, join); valori mancanti; variabili qualitative; tabelle a doppia entrata; serie storiche; normalizzazione e standardizzazione; covarianza, correlazione e retta di regressione.

10. Grafici con matplotlib: grafici a linee, a dispersione, a barre e istogrammi; più serie, legende e stili; più grafici in una figura; i grafici di pandas.

11. Classi e oggetti: definizione di una classe, metodi, attributi della classe e dell'istanza, metodi speciali, ereditarietà e ridefinizione dei metodi, classi astratte, il metodo **call**.

12. Reti neurali artificiali: il neurone artificiale e il percettrone; le funzioni di attivazione; il percettrone multistrato; l'addestramento di una rete con la discesa del gradiente e la retropropagazione dell'errore. Realizzazione con PyTorch: tensori (torch.tensor), calcolo automatico del gradiente (backward), ottimizzatori (torch.optim, step, zero_grad), strati e classe della rete (torch.nn.Linear, torch.nn.ReLU), funzioni di errore, Dataset e DataLoader, ciclo di addestramento, dropout e normalizzazione dei lotti (torch.nn.Dropout, torch.nn.BatchNorm1d), preparazione dei dati; introduzione alle reti profonde.

Prerequisiti

Le conoscenze matematiche e logiche acquisite nella scuola secondaria superiore; nozioni di base di algebra lineare (vettori, matrici, prodotto scalare) e di statistica (media, varianza, distribuzioni di probabilità). Sono inoltre richieste capacità di base di programmazione in Python: tipi di dato, istruzioni di selezione e di iterazione, funzioni.

Metodi didattici

Lezioni frontali con esercitazioni guidate. Le lezioni si svolgono nei laboratori didattici, per consentire agli studenti di applicare immediatamente al calcolatore i concetti presentati.

Modalità di verifica dell'apprendimento

L'esame prevede una prova scritta, che si svolge nei laboratori didattici di Ateneo e consiste nello sviluppo di programmi in Python per la soluzione di problemi di elaborazione e analisi dei dati. La prova verte su tutti gli argomenti del corso: la programmazione in Python di base, l'elaborazione dei dati con le librerie NumPy e pandas e le reti neurali artificiali. Per superare l'esame è necessario dimostrare una conoscenza sufficiente di ciascuno di questi argomenti; una carenza grave in uno di essi non può essere compensata dal risultato ottenuto negli altri. Nella valutazione si considerano la correttezza dei risultati, la capacità di scegliere gli strumenti adeguati al problema e la chiarezza del codice. Durante la prova è consentita la consultazione della sola documentazione ufficiale di Python e delle librerie utilizzate.

Testi di riferimento

Paul J. Deitel, Harvey M. Deitel, **Introduzione a Python. Per l'informatica e la data science**, Pearson, 2021 (ISBN 978-8891915924).
Appunti del docente, esempi ed esercizi disponibili sulla piattaforma e-learning del corso.
Documentazione ufficiale: docs.python.org, numpy.org/doc, pandas.pydata.org/docs, matplotlib.org, pytorch.org/docs.

Periodo di erogazione dell'insegnamento

Primo semestre

Lingua di insegnamento

Italiano

Sustainable Development Goals

ISTRUZIONE DI QUALITÁ