

## 1 Presentazione corso

Il Master Intelligenza Artificiale e Machine Learning di ELIS nasce per rispondere alla crescente domanda di professionisti in Intelligenza artificiale (AI) e machine learning (ML), ambiti in forte espansione a livello globale.

Negli ultimi anni, l'AI ha rapidamente trasformato numerosi settori, automatizzando processi complessi, migliorando l'analisi dei dati e consentendo lo sviluppo di tecnologie innovative come il riconoscimento vocale e visivo, la guida autonoma, le interazioni automatizzate con i clienti e molto altro. Questo progresso tecnologico sta generando nuove opportunità di carriera, con competenze in AI e ML sempre più richieste nel mercato del lavoro.

Attualmente, il valore globale del mercato dell'AI è stimato a circa 150 miliardi di dollari e si prevede che raggiunga oltre 1.800 miliardi entro il 2030. Questa espansione è trainata dalla crescente integrazione dell'AI in settori come sanità, finanza, produzione e intrattenimento, dove le applicazioni dell'intelligenza artificiale migliorano la precisione, riducono i costi e ottimizzano le operazioni aziendali.

Il ruolo di un AI/ML Engineer è cruciale per progettare, sviluppare e implementare modelli di machine learning e algoritmi di intelligenza artificiale che risolvano problemi complessi. Le competenze fondamentali per questo ruolo includono la programmazione (soprattutto in Python), la conoscenza di framework di ML come TensorFlow e PyTorch, e una solida comprensione della matematica e della statistica. A queste si aggiunge la capacità di manipolare e analizzare grandi quantità di dati, sfruttando le tecniche di data science per addestrare e valutare modelli AI. Il corso Junior AI & ML Engineer è progettato per formare una figura professionale completa e operativa fin dai primi passi, capace di supportare team di sviluppo e gestire progetti di intelligenza artificiale. I partecipanti impareranno le basi della programmazione in AI, le principali tecniche di machine learning, e avranno esperienza pratica nello sviluppo e nell'implementazione di modelli. L'obiettivo è preparare figure pronte ad affrontare le sfide tecniche e a contribuire efficacemente all'innovazione aziendale nel campo dell'intelligenza artificiale e del machine learning.

### 1.1 Cosa fa un AI Engineer?

Un AI Engineer è responsabile della progettazione, sviluppo e implementazione di soluzioni di intelligenza artificiale avanzate, sfruttando competenze in programmazione, data science e machine learning. Questo ruolo prevede l'uso di enormi quantità di dati (Big Data) per creare sistemi intelligenti e automatizzati, capaci di ottimizzare processi aziendali e supportare le decisioni strategiche.

I compiti principali di un Ingegnere dell'Intelligenza Artificiale includono:

- **Progettazione e sviluppo di algoritmi:** Creazione di algoritmi avanzati, come quelli di machine learning e deep learning, che permettono alle macchine di acquisire capacità di apprendimento autonomo per eseguire compiti specifici all'interno dei processi aziendali.
- **Analisi e gestione dei dati:** Raccolta, analisi e gestione di vasti set di dati, necessari per l'addestramento e la valutazione dei modelli di intelligenza artificiale.
- **Implementazione di soluzioni di AI per il business:** Sviluppo di soluzioni AI personalizzate per rispondere a esigenze specifiche delle aziende, identificando le tecnologie più adeguate per ottimizzare processi o risolvere problemi particolari.
- **Integrazione e scalabilità dei modelli AI:** Applicazione pratica dei modelli di intelligenza artificiale in contesti aziendali, garantendo che siano scalabili e adattabili a diverse

dimensioni e complessità operative.

- Manutenzione e miglioramento dei sistemi: Ottimizzazione continua dei sistemi AI esistenti per migliorare le loro prestazioni e affidabilità nel tempo.
- Conformità etica e normativa: Verifica che le soluzioni AI rispettino standard etici e normative di settore, con particolare attenzione alla privacy e alla sicurezza dei dati personali.

Queste attività sono cruciali per garantire che le soluzioni di intelligenza artificiale siano non solo efficienti e sicure, ma anche allineate agli obiettivi strategici e ai valori aziendali.

Programma del corso:

## 1 – Introduzione

## 2 – Python

**Obiettivi del modulo:** Fornire una conoscenza completa e pratica di Python, partendo dalle basi fino ai concetti avanzati della programmazione orientata agli oggetti. Alla fine del corso, gli studenti saranno in grado di sviluppare applicazioni Python, sfruttando le potenzialità del linguaggio per risolvere problemi reali.

### Introduzione a Python

- Storia e filosofia di Python
- Installazione e configurazione dell'ambiente di sviluppo
- Sintassi di base e regole di indentazione

### Variabili e tipi di dati

- Variabili e assegnazioni
- Tipi di dati
- Funzioni di input e output

### Controllo del flusso

- Condizioni: if, elif, else
- Cicli: for, while
- Cicli: for else, while else
- Gestione delle eccezioni con try-except

### Strutture dati

- Liste: creazione, accesso agli elementi, metodi principali
- Tuple: immutabilità e utilizzi
- Dizionari: chiavi, valori e metodi

- Set: insiemi, operazioni matematiche

## **Funzioni e moduli**

- Definizione di funzioni
- Parametri e valori di ritorno
- Lambda functions
- Importazione di moduli e librerie esterne

## **Classi e oggetti**

- Definizione di classi
- Attributi e metodi
- Esempi pratici di utilizzo

## **Ereditarietà**

- Concetto di ereditarietà
- Funzione super()
- Classi base e classi derivate
- Concetto di overriding, dunder methods

## **Incapsulamento**

- Modificatori di accesso (public, private, protected)
- Property e decorators
- Metodi getter e setter

# **3 – Database e Data Management**

**Obiettivi del modulo:** Fornire agli studenti le competenze necessarie per lavorare con database relazionali e non relazionali, ottimizzare le interrogazioni e manipolare dati con Python. Al termine del modulo, gli studenti saranno in grado di progettare, interrogare e gestire database, oltre a trasformare e analizzare dati.

## **Introduzione ai database relazionali**

- Concetti fondamentali di database relazionali
- Modelli E-R e normalizzazione
- Panoramica di SQL e principali RDBMS (MySQL, PostgreSQL, SQLite)

## **CRUD con SQL**

- Creazione di tabelle e definizione degli schemi
- Operazioni di lettura (SELECT, JOIN, GROUP BY)
- Modifica e cancellazione dei dati (INSERT, UPDATE, DELETE)

## **Tecniche di ottimizzazione delle query**

- Strutture di indicizzazione
- Utilizzo di viste e subquery efficienti
- Casi studio di ottimizzazione delle prestazioni

## **Introduzione ai database NoSQL**

- Differenze tra SQL e NoSQL
- Tipologie di database NoSQL (document, key-value, graph, columnar)
- Installazione e configurazione di MongoDB

## **Operazioni con MongoDB**

- Creazione e gestione di collezioni
- Query con MongoDB (find, aggregate)
- Indici e best practices
- Esercizi su dataset reali

## **DataFrame e manipolazione dati: pandas**

- Creazione e caricamento di DataFrame
- Pulizia dei dati (gestione dei null, duplicati, outlier)
- Operazioni di trasformazione (pivot, melt, merge)

## **Aggregazione e analisi avanzata: pandas**

- Raggruppamenti e funzioni di aggregazione
- Visualizzazione dei dati con Matplotlib e Seaborn
- Concetti di EDA (Exploratory Data Analysis)

## 4 – Algebra Lineare e Calcolo Vettoriale per il Machine Learning

**Obiettivo del modulo:** Fornire agli studenti una solida comprensione dei concetti fondamentali di algebra lineare applicati al machine learning, con un approccio pratico basato su Python e la libreria NumPy. Alla fine del modulo, gli studenti saranno in grado di manipolare vettori e matrici, comprendere gli spazi vettoriali e risolvere problemi matematici comuni nell'addestramento dei modelli di machine learning.

### Vettori e operazioni di base

- Definizione di vettore
- Operazioni sui vettori: somma, sottrazione, prodotto scalare
- Normalizzazione e distanza euclidea
- Implementazione con NumPy

### Matrici e trasformazioni lineari

- Creazione e manipolazione di matrici
- Prodotto matrice-vettore e matrice-matrice
- Matrici trasposte e inverse
- Applicazioni pratiche nel machine learning

### Distanze tra vettori

- Distanza euclidea
- Distanza di Manhattan
- Distanza di Minkowski
- Calcolo delle distanze con NumPy

### Similarità e prodotto scalare

- Prodotto scalare e interpretazione geometrica
- Similarità del coseno
- Angolo tra vettori
- Applicazioni della similarità in ambito analitico

### Introduzione ai tensori e manipolazione

- Definizione di tensore
- Tensori di terzo grado e oltre
- Rappresentazione delle immagini come tensori (RGB, canali di colore)
- Manipolazione di immagini con NumPy

## 5 – Machine Learning in Python

**Obiettivo del modulo:** Fornire agli studenti una comprensione approfondita dei principali concetti di machine learning, sia a livello teorico che pratico, utilizzando Python con le librerie scikit-learn e pandas. Alla fine del modulo, gli studenti saranno in grado di costruire, valutare e ottimizzare modelli di machine learning per diverse tipologie di problemi.

### Introduzione al Machine Learning

- Definizione e applicazioni del ML
- Tipi di apprendimento: supervisionato, non supervisionato, per rinforzo
- Workflow tipico di un progetto ML

### Preparazione dei dati

- Caricamento e manipolazione dei dataset con pandas
- Divisione in training set, test set e validation set
- Tecniche di feature scaling e normalizzazione

### Tecniche avanzate di preprocessing

- One-hot encoding e label encoding
- Gestione dei valori mancanti
- Feature engineering

### Panoramica di scikit-learn

- Struttura della libreria
- Pipeline di machine learning
- Esempio pratico di un modello semplice

### Valutazione dei modelli

- Metriche di valutazione: accuratezza, precisione, richiamo, F1-score
- Matrice di confusione
- Metodologie di validazioni dei risultati (k-fold, leave-one-out)

## **Supervised Machine Learning**

- Regressione lineare e logistica: Teoria e implementazione
- Interpretazione dei coefficienti
- Visualizzazione dei risultati
- Decision Tree
- Random Forest e feature importance
- Algoritmi di Boosting (XGBoost ecc.)
- Hyperparameter tuning con Grid Search
- Support Vector Machines

## **Ottimizzazione dei modelli**

- Prevenzione dell'overfitting (regolarizzazione, early stopping)
- Grid Search e Random Search

## **Unsupervised Machine Learning**

- K-means e K-medoids
- Clustering gerarchico
- Valutazione dei cluster (silhouette score, inertia)
- PCA (Principal Component Analysis)

## 6 – Deep Learning con TensorFlow e Pytorch

**Obiettivo del modulo:** Fornire agli studenti una solida comprensione dei fondamenti del deep learning, sia dal punto di vista teorico che pratico, utilizzando TensorFlow e PyTorch. Alla fine del modulo, gli studenti saranno in grado di costruire, addestrare e valutare reti neurali per vari tipi di problemi. La creazione e lo sviluppo pratico dei modelli viene effettuata utilizzando Google Colab per il calcolo accelerato su GPU.

### Introduzione al Deep Learning

- Concetti base delle reti neurali
- Differenze tra machine learning e deep learning
- Panoramica delle librerie TensorFlow e PyTorch

### Funzioni di attivazione

- Sigmoid, ReLU, Tanh, Softmax
- Scelta della funzione di attivazione più adatta
- Implementazione pratica con TensorFlow e PyTorch

### Ottimizzazione e backpropagation

- Discesa del gradiente e varianti (SGD, Adam, RMSprop)
- Algoritmo di backpropagation
- Funzioni di perdita più comuni (MSE, Cross-Entropy)
- Overfitting e underfitting: regolarizzazione e dropout

### Costruzione di una rete neurale

- Layer, pesi, bias e forward pass
- Definizione di modelli sequenziali e funzionali
- Addestramento e validazione dei modelli

### Gestione dei dataset

- Train/test split e validation set
- DataLoader e Dataset in PyTorch
- Dataset predefiniti: utilizzo di MNIST e CIFAR-10

### Visualizzazione e analisi dei risultati

- Uso di TensorBoard per il monitoraggio
- Visualizzazione delle metriche con matplotlib



## **Reti Neurali Convulsive (CNN)**

- Convoluzioni, kernel e feature maps
- Pooling, padding e stride
- Costruzione di una CNN con TensorFlow e PyTorch
- Addestramento su dataset di immagini
- Data augmentation e regularization
- Transfer learning e fine-tuning

## **Reti Neurali Ricorrenti (RNN)**

- Struttura delle RNN
- Problemi di vanishing/exploding gradient
- LSTM e GRU
- Confronto tra le architetture
- Utilizzo delle RNN per l'analisi e la previsione su serie storiche
- Confronto delle prestazioni tra CNN e RNN su serie storiche

## **Introduzione al Natural Language Processing**

- Tokenizzazione e sequenze
- Preparazione e pulizia del testo
- Word embeddings (Word2Vec, GloVe)
- RNN per l'analisi del testo (LSTM e BLSTM)
- Introduzione alle reti Transformer

## 7 – Generative AI

**Obiettivo del modulo:** Fornire agli studenti una comprensione approfondita delle tecniche di Generative AI, esplorando modelli avanzati, dalle architetture Transformer alla generazione di immagini e testo, con un focus pratico sull'uso delle librerie Hugging Face e LangChain.

### Teoria dei Transformer

- Architettura Transformer: Encoder e Decoder
- Meccanismo di self-attention e multi-head attention
- Positional Encoding e gestione delle sequenze

### Utilizzo di modelli preaddestrati

- Introduzione alla libreria Hugging Face
- BERT, GPT, LLaMa: panoramica e differenze
- Caricamento e inferenza con modelli preaddestrati

### Fine-tuning di modelli NLP

- Preparazione del dataset con Hugging Face Datasets
- Addestramento personalizzato su task specifici (es: sentiment analysis, text classification)
- Valutazione e metriche di performance

### Applicazioni pratiche

- Generazione di testo con modelli GPT
- Creazione di chatbot con modelli conversazionali
- Traduzione automatica e text summarization

### Generative Adversarial Networks (GAN)

- Struttura delle GAN: Generator e Discriminator
- Addestramento e loss function
- Esempi pratici di generazione di immagini

## **Autoencoder e VAE**

- Teoria degli Autoencoder
- Variational Autoencoder (VAE)
- Applicazioni alla generazione di dati e compressione

## **Stable Diffusion e generazione avanzata**

- Introduzione ai modelli di diffusione
- Generazione di immagini realistiche con Stable Diffusion
- Prompt engineering per risultati ottimali

## **LangChain e AI nei flussi di lavoro**

- Concetti base di LangChain
- Creazione di catene di chiamate a modelli LLM
- Memoria e gestione del contesto nelle conversazioni

## **Retrieval-Augmented Generation (RAG)**

- Introduzione alla tecnica RAG
- Indicizzazione e recupero delle informazioni
- Implementazione di un sistema RAG con LangChain
- Integrazione di modelli generativi con fonti dati esterne

## 8 – MLOPS, Tecnologie di Deploying e CICD per l'implementazione di sistemi basati su IA

**Obiettivi del modulo** Questo modulo fornisce una panoramica completa delle pratiche di MLOps per la gestione del ciclo di vita di un modello di machine learning in produzione. I partecipanti acquisiranno competenze su architetture di sistema, containerizzazione, pipeline di dati, addestramento e inferenza in tempo reale, nonché sulle tecniche di monitoraggio e ottimizzazione. Verranno esplorate le principali tecnologie utilizzate nel settore per garantire scalabilità, affidabilità e riproducibilità dei sistemi di intelligenza artificiale.

### Fondamenti di MLOps e Architettura ML

- Definizione del problema e inquadramento
- Architettura del sistema ML
- Il paradigma Feature-Training-Inference pipeline
- Costruzione iterativa di sistemi ML modulari
- Strumenti open-source e infrastrutture serverless

### Containerizzazione e Microservizi

- Fondamenti di Docker:
  - Concetti base e containerizzazione
  - Creazione e gestione di immagini Docker
- Sviluppo e containerizzazione di un microservizio in Python e Docker
- Workflow di sviluppo con Makefile
- Automatizzazione di linting e formattazione del codice

### Feature Pipeline

- Ingestione dati in tempo reale
- Trasformazioni di dati stateful in Python
- Lettura e scrittura dati in Apache Kafka
- Scrittura di codice Python modulare per debugging e testing efficiente
- Concetto di Feature Store e importanza nell'architettura ML
- Popolamento dei dati storici (backfilling)
- Rischi di skew offline-online e metodi di mitigazione
- Kafka: gestione di topic, partizioni e gruppi di consumer
- Scalabilità orizzontale delle trasformazioni dati con Docker e Kafka

## **Training Pipeline**

- Sviluppo iterativo di modelli ML
- Addestramento e valutazione di modelli (time-series forecasting, classificazione, regressione)
- Trasformazioni dipendenti dal modello
- Feature engineering avanzato
- Parametrizzazione delle metriche di valutazione
- Tracciamento degli esperimenti con MLflow
- Introduzione ai Model Registry e loro integrazione nella pipeline
- Containerizzazione per garantire riproducibilità

## **Inference Pipeline**

- Predizioni in tempo reale con modelli ML in Python
- Creazione di un microservizio di inferenza in streaming
- Visualizzazione delle predizioni con Streamlit
- Integrazione della pipeline di inferenza con il Model Registry

## **Deployment e Monitoraggio**

- Deploy di microservizi
- Sistemi di monitoraggio
- Validazione della qualità delle feature
- Monitoraggio degli errori del modello
- Pianificazione del ri-addestramento del modello

## **Ottimizzazione e Scalabilità**

- Ottimizzazione dell'ingegneria delle feature
- Ottimizzazione degli iperparametri per migliorare le prestazioni dei modelli
- Scalabilità orizzontale dei servizi di feature engineering