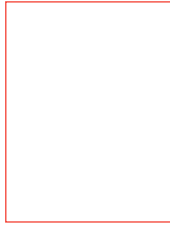


Emanuele Bellini



Neolaureato del corso di Laurea Magistrale in Ingegneria Informatica, curriculum Data Science, presso l'Università degli Studi di Perugia.

Durante il percorso accademico ho sviluppato competenze avanzate in progettazione e sviluppo software, con particolare interesse per cloud computing, analisi dati e machine learning.

Mi contraddistinguono capacità di problem solving, attitudine al lavoro in team e forte motivazione a mettere in pratica le conoscenze acquisite in contesti professionali.

Competenze

- Informatiche:
 - **Linguaggi di programmazione:** Java, Python, JavaScript, R, MATLAB.
 - **Big Data & Database:** SQL, Apache Hadoop, Apache Spark (PySpark), Neo4j.
 - **Sviluppo Web:** HTML, CSS.
 - **Data Science & Engineering:** Machine learning, Deep Learning, Convolutional Neural Networks, Data Analysis & Visualization, Ingegneria del software.
 - **Software & Strumenti:** Microsoft Office.
- Linguistiche:
 - Italiano: Madrelingua
 - Inglese scritto: Buono
 - Inglese parlato: avanzato
- Capacità organizzative
- Team work
- Problem solving
- Rispetto delle tempistiche date

Esperienza

03/23 – 09/23

Borsista di ricerca / Università degli Studi di Perugia

Progetto: "Per una giustizia giusta: innovazione ed efficienza negli uffici giudiziari - GIUSTIZIA AGILE".

Istruzione

2018 - 2022

Laurea triennale in Ingegneria Informatica / Università degli Studi di Perugia

Laurea triennale in Ingegneria Informatica conseguita con la valutazione di 108/110.

Agevolazione per merito.

Inserimento percorso di eccellenza.

2023 - 2026

Laurea magistrale in Ingegneria Informatica (curriculum Data Science) / Università degli Studi di Perugia

Laureato del corso di laurea magistrale in Ingegneria Informatica (curriculum Data Science) con la valutazione di 110L/110.

Agevolazione per merito.

Inserimento percorso di eccellenza.

Progetti Universitari

Progetto e realizzazione di una Interfaccia visuale

Clone di MonacoGP (Videogioco 2D)

- Progettazione e sviluppo di un videogioco arcade 2D ispirato al classico MonacoGP, dotato di interfaccia grafica interattiva personalizzata.
- Applicazione rigorosa dei principi della Programmazione Orientata agli Oggetti (OOP) per modellare le entità di gioco (veicoli, collisioni, tracciato).
- Implementazione di un Game Loop efficiente per gestire separatamente la logica di gioco, il rendering grafico e la cattura degli eventi (input dell'utente) senza bloccare l'interfaccia.

Compressore Dati (Codifica di Huffman)

- Sviluppo di un applicativo per la compressione e decompressione lossless (senza perdita di dati) di file di testo utilizzando l'algoritmo di Huffman.

Risolutore di Problemi Combinatori (Branch and Bound)

- Implementazione di un motore algoritmico basato sulla tecnica Branch and Bound per la risoluzione esatta di problemi di ottimizzazione complessi.
- Sviluppo di funzioni euristiche per il calcolo del limite (bound) e il "pruning" (taglio) dei rami sub-ottimi, riducendo drasticamente i tempi di esecuzione rispetto a un approccio brute-force.

Sistema per la visualizzazione di dati calcistici

- Progettazione e sviluppo di un'interfaccia grafica interattiva in Python per la visualizzazione, manipolazione e analisi di dati calcistici open-source.
- Utilizzo di librerie specifiche come Matplotlib, mplsoccer e PySimpleGUI per generare rappresentazioni grafiche avanzate come heat map, reti di passaggi, mappe di pressione e metriche Expected Goals (xG).
- Implementazione di funzioni dedicate per la manipolazione di complessi data frame contenenti le coordinate e i dettagli degli eventi generati durante i match.

Sistema di Raccomandazione Anime (Big Data)

- Creazione di un sistema di raccomandazione efficiente e scalabile in grado di fornire suggerimenti personalizzati basati sulle preferenze degli utenti e sulle caratteristiche degli anime.
- Integrazione di tecnologie per il calcolo distribuito, impiegando Apache Hadoop e PySpark per l'elaborazione dei dati, e Neo4j per l'archiviazione e la visualizzazione a grafo delle relazioni.
- Implementazione di algoritmi di Machine Learning, nello specifico ALS (Alternating Least Squares) per il collaborative filtering e Cosine Similarity per il content-based filtering.

Analisi Esplorativa sulla Mortalità Infantile (R)

- Conduzione di un'approfondita Analisi Esplorativa dei Dati (EDA) in linguaggio R sulle statistiche mondiali della World Health Organization (WHO) per esaminare le differenze di mortalità per genere, regione e trend temporali.
- Elaborazione del dataset originale tramite pulizia dei valori mancanti e selezione delle feature maggiormente rilevanti per migliorare la qualità delle analisi.
- Applicazione di test statistici rigorosi (test di Shapiro-Wilk, test di Mann-Whitney, test di Kruskal-Wallis) per verificare la significatività statistica delle divergenze riscontrate.

Matrix Completion per Sistemi di Raccomandazione (MATLAB)

- Sviluppo in ambiente MATLAB di diverse tecniche di Matrix Completion per gestire matrici sparse all'interno dei sistemi di raccomandazione (basati sull'Anime Dataset 2023).
- Implementazione e confronto di algoritmi avanzati, tra cui Minimizzazione della Norma Nucleare, Singular Value Thresholding (SVT) e Matrix Factorization tramite Gradient Descent (GD) e Alternating Least Squares (ALS).
- Valutazione delle performance e della qualità di ricostruzione delle matrici per feedback espliciti ed impliciti utilizzando il Normalized Mean Squared Error (NMSE).

Sviluppo di un Classificatore di Machine Learning

- Progettazione, addestramento e validazione di un modello per task di classificazione.
- Implementazione e confronto delle performance di diversi algoritmi predittivi, tra cui Logistic Regression, K-Nearest Neighbors (K-NN) e Support Vector Machines (SVM).
- Sviluppo e fine-tuning di modelli basati su Reti Neurali per ottimizzare l'accuratezza e la generalizzazione della classificazione.

Stima della Profondità (Depth Estimation) con Deep Learning

- Sviluppo di un modello avanzato di Deep Learning dedicato al task di visione artificiale per la stima della profondità (depth estimation).
- Impiego di architetture basate su Reti Neurali Convoluzionali (CNN) per l'estrazione e l'analisi complessa delle feature spaziali.
- Integrazione di ResNet come *backbone* (scheletro) della rete per ottimizzare l'apprendimento profondo e migliorare le prestazioni complessive del modello.