



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Trade-off tra Training e Inferenza di Modelli ML: Uno Studio di Benchmark dei Requisiti Non Funzionali su Smartphone

RELATORE

Prof. Fabio Palomba

CORRELATORE

Dott. Vincenzo De Martino

Università degli Studi di Salerno

CANDIDATO

Samuele Russo

Matricola: 0512113317

Anno Accademico 2024-2025

Questa tesi è stata realizzata nel

sesa^{lab}
SOFTWARE ENGINEERING
SALERNO

"A chi ha avuto il coraggio di lasciare la propria casa e affrontare sacrifici per inseguire i propri sogni."

Abstract

Negli ultimi anni, l'interesse per la Green AI ha stimolato studi sull'efficienza energetica dei modelli di Machine Learning. Tuttavia, la letteratura esistente spesso si limita a un unico dataset e non considera in modo completo le differenze tra training e inferenza. Questa tesi analizza sei algoritmi di ML (SVM, Decision Tree, KNN, Random Forest, AdaBoost e Bagging Classifier) su tre dataset, confrontandone consumo energetico, tempo di inferenza ed utilizzo delle risorse. I risultati ottenuti in fase di training confermano le tendenze di altri studi rilevanti, mentre rivelano differenze significative in inferenza: il Decision Tree emerge come il modello più efficiente, con bassi consumi e tempi di inferenza ridotti, ideale per dispositivi con risorse limitate. AdaBoost e Bagging Classifier offrono un buon equilibrio tra accuratezza ed efficienza, mentre SVM e KNN risultano meno pratici per i loro consumi elevati e tempi lunghi. Il Random Forest, nonostante alte accuratezze, è penalizzato da consumi elevati. Futuri sviluppi potrebbero includere l'estensione dell'analisi a un numero maggiore di dataset provenienti da diverse aree applicative. Un'altra direzione interessante sarebbe l'ottimizzazione dell'efficienza energetica attraverso tecniche o hardware specializzati. Inoltre, sarebbe utile esplorare ulteriori algoritmi e analizzare i compromessi tra accuratezza ed efficienza in scenari di utilizzo reale.

Indice

Elenco delle Figure	iii
Elenco delle Tabelle	iv
1 Introduzione	1
1.1 Motivazioni ed Obbiettivi	1
1.2 Risultati ottenuti	3
1.3 Struttura della tesi	4
2 Stato dell'arte	5
2.1 Related work	5
3 Metodologia	10
3.1 Obbiettivi di ricerca	10
3.2 Selezione dei modelli	11
3.3 Metriche di confronto	12
3.4 Selezione dei dataset	14
3.5 Ambiente di Esecuzione e Metodologia di Test	15
3.6 Training	15
3.7 Inferenza	16
3.8 Analisi dei dati	17

4	Analisi dei risultati	19
4.1	RQ1: Come gli algoritmi di ML differiscono in fase di training su diversi dataset?	19
4.2	RQ2: Come gli algoritmi di ML differiscono in fase di inferenza? . . .	22
4.3	RQ3: Quali sono i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML su dataset eterogenei?	29
5	Conclusioni	32
	Bibliografia	34

Elenco delle figure

1.1	Settori di utilizzo del Machine Learning	2
4.1	Confronto dei consumi energetici dei modelli in fase di training sui tre dataset.	20
4.2	Risultati ottenuti dai modelli sul dataset Compas	23
4.3	Risultati ottenuti dai modelli sul dataset German	25
4.4	Risultati ottenuti dai modelli sul dataset SmsSpam	26
4.5	Consumi energetici su Compas	30
4.6	Consumi energetici su German	30
4.7	Consumi energetici su SmsSpam	31

Elenco delle tabelle

3.1	Caratteristiche dei dataset utilizzati nello studio	14
4.1	Consumo energetico mediano (Joule) durante la fase di training. . . .	22

CAPITOLO 1

Introduzione

Il Machine Learning è un campo dell'intelligenza artificiale che consente ai sistemi di apprendere dai dati ed effettuare previsioni o prendere decisioni [1]. Negli ultimi decenni ha trasformato molteplici settori ed ha mostrato un potenziale significativo nelle innovazioni digitali, con un numero crescente di applicazioni ML che si espandono attraverso un ampio spettro di industrie, dalla sanità all'agricoltura [2]. Tuttavia, questo progresso comporta un costo significativo in termini di risorse computazionali ed energetiche, sollevando importanti questioni sull'efficienza e la sostenibilità [3].

1.1 Motivazioni ed Obbiettivi

Il machine learning sta apportando miglioramenti significativi in diversi settori [4]: nella manifattura e produzione viene utilizzato per la manutenzione predittiva, analizzando i dati dei sensori per prevenire guasti e ridurre i tempi di fermo, ottimizzando anche i processi produttivi in tempo reale per migliorare l'efficienza delle risorse e ridurre gli sprechi. Nel campo della logistica aiuta a prevedere la domanda, ottimizzare i percorsi di consegna e gestire l'inventario, riducendo i costi e migliorando l'affidabilità dei servizi. In ambito sanitario supporta la diagnosi di malattie e

la creazione di piani di trattamento personalizzati. Inoltre, nel settore finanziario, è impiegato nel trading algoritmico, nella rilevazione di frodi e nella valutazione del rischio di credito.

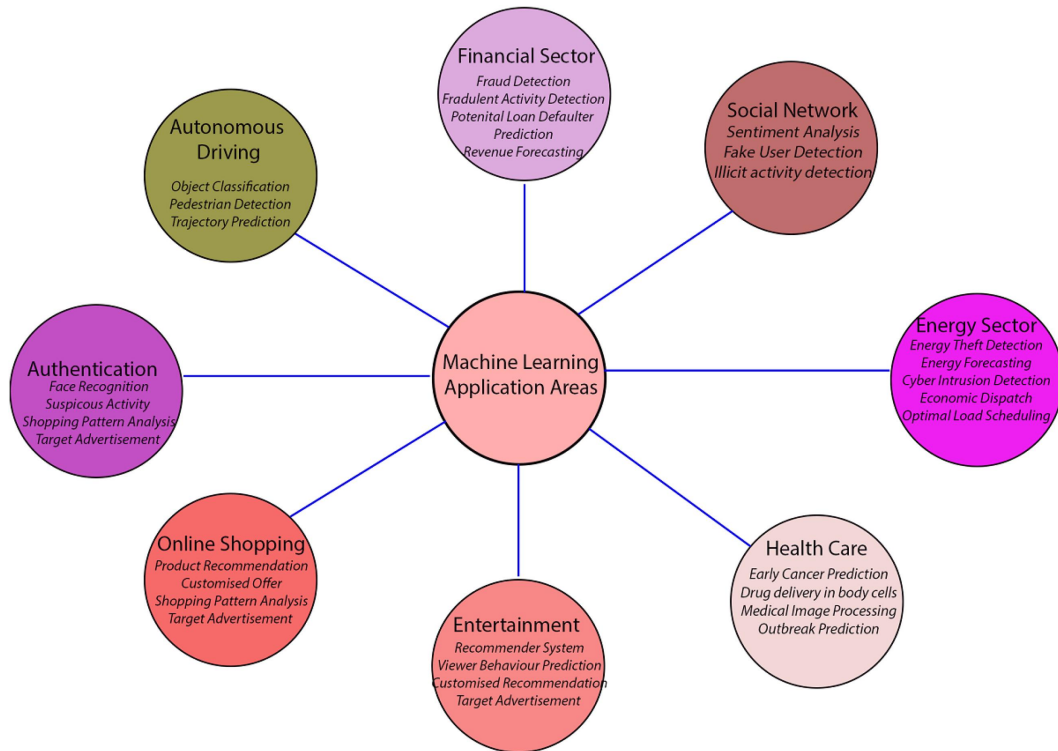


Figura 1.1: Settori di utilizzo del Machine Learning

Il crescente utilizzo del machine learning (ML) nei più svariati settori ha posto l'accento sull'importanza di valutare le implicazioni legate alle risorse computazionali ed energetiche, andando a ridurre le emissioni di carbonio nello sviluppo e nell'esecuzione dei modelli [3]. Tale riflessione ha spinto studi successivi ad orientare la ricerca verso una maggiore attenzione alle implicazioni ambientali; in particolare è stato introdotto il concetto di Green AI, ovvero ricerche innovative sull'IA considerando anche i costi computazionali [5]. Il lavoro di Verdecchia [6] sull'analisi del consumo energetico dei modelli di ML ha contribuito in modo significativo alla comprensione dei costi ambientali associati alla fase di training. Tuttavia, presenta delle limitazioni. In primo luogo, lo studio si concentra esclusivamente sui consumi energetici durante il training, trascurando la fase di inferenza, che può avere un impatto rilevante sulle metriche di consumo complessive. In secondo luogo, l'analisi si limita a un

singolo dataset, il che riduce la generalizzabilità dei risultati. Infine, tale studio non usa come variabili dipendenti il peso del modello, la quantità di memoria usata, la sua accuratezza, il tempo impiegato per l'inferenza e il consumo energetico; tutte metriche utili per valutare la performance di un modello di ML [7]. Questo studio nasce proprio dalla necessità di verificare se e come le prestazioni degli algoritmi di ML cambiano utilizzando diversi dataset, al fine di determinare le correlazioni tra accuratezza, tempo di inferenza, peso del modello, consumo energetico e utilizzo di memoria in contesti differenti. In parallelo, si intende esplorare l'impatto di queste variabili durante la fase di inferenza e confrontare i risultati con quelli ottenuti sulla fase di training. L'utilità di questo studio risiede nella sua capacità di fornire una serie di informazioni fondamentali per progettare modelli che possano massimizzare l'accuratezza mantenendo al contempo un'efficienza ottimale in termini di risorse computazionali ed energetiche.

1.2 Risultati ottenuti

In fase di training, i risultati confermano quanto osservato da Verdecchia et al. [6], evidenziando tendenze simili, con il KNN e il Decision Tree che si sono rivelati i modelli più efficienti dal punto di vista energetico, mentre il Random Forest ha mostrato consumi elevati. In fase di inferenza, emergono differenze significative: il Decision Tree si conferma il modello più efficiente in termini di consumo energetico e tempo di inferenza, risultando particolarmente adatto a contesti con risorse limitate, come i dispositivi IoT. Il KNN offre un buon bilanciamento tra accuratezza ed efficienza, ma i tempi di inferenza più lunghi possono limitarne l'applicazione in ambienti a bassa latenza. Alcuni modelli, pur ottenendo alta accuratezza, non sono i più efficienti energeticamente, difatti il Random Forest, pur garantendo una buona accuratezza, evidenzia consumi energetici relativamente elevati e un maggiore peso del modello, rendendolo più adatto a contesti server, dove le risorse non rappresentano un vincolo stringente. L'SVM, pur garantendo buona accuratezza, ha mostrato una variabilità significativa nei consumi energetici, suggerendo che la sua efficienza dipenda dal dataset utilizzato. AdaBoost e Bagging Classifier si distinguono come soluzioni equilibrate, con consumi contenuti e tempi di inferenza rapidi, risultando

opzioni flessibili per diversi scenari applicativi. I risultati sottolineano, inoltre, l'importanza delle caratteristiche del dataset: le prestazioni dei modelli variano in modo significativo in base ai dati utilizzati.

1.3 Struttura della tesi

La struttura della tesi è organizzata come segue:

- **Capitolo 1:** descrive le motivazioni e gli obiettivi alla base di questa ricerca;
- **Capitolo 2:** analizza brevemente lo stato dell'arte, gli algoritmi di ML e le metriche;
- **Capitolo 3:** presenta in dettaglio la metodologia usata nella ricerca;
- **Capitolo 4:** analizza i risultati ottenuti, tramite le domande di ricerca.
- **Capitolo 5:** sintetizza i risultati e propone sviluppi futuri.

CAPITOLO 2

Stato dell'arte

2.1 Related work

La sempre più ampia adozione dei modelli di ML in svariati ambiti ha portato all'aumento del consumo energetico richiesto e, di conseguenza, delle emissioni di CO₂-equivalente generate. Il concetto di **Green AI** [5] rappresenta un approccio innovativo all'intelligenza artificiale, che punta a ridurre il consumo di risorse computazionali e l'impatto ambientale. A differenza della **Red AI**, che insegue miglioramenti delle prestazioni attraverso l'impiego di modelli sempre più grandi e dispendiosi, Green AI introduce l'efficienza come criterio centrale di valutazione, accanto all'accuratezza. Questo significa considerare aspetti come il numero di operazioni in virgola mobile (FPO), il consumo energetico e il tempo computazionale necessari per sviluppare e utilizzare i modelli. Ciò si riflette nella crescente consapevolezza che l'impatto ambientale dovuto all'elevato consumo di energia e alle emissioni di carbonio associate all'AI è significativo; quindi adottare un approccio più sostenibile non solo riduce i costi energetici, ma diventa una questione ambientale, sociale ed etica. A tal proposito, i ricercatori hanno incrementato l'interesse per il Green AI [8], con lo scopo di analizzare l'impatto ambientale dei modelli di Intelligenza Artificiale. Il documento redatto da Verdecchia et al. [8] fornisce una panoramica dei lavori scien-

tifici esistenti relativi a tale tematica. Gli autori hanno notato che i primi articoli sono stati pubblicati a partire dal 2015, con una crescita più o meno costante nel numero di pubblicazioni negli anni successivi. Inoltre, la maggior parte di questi articoli si concentra principalmente sul tema del consumo energetico, come lo studio di Lacoste et al. [3] che propone un approccio per quantificare le emissioni di carbonio associate all'addestramento dei modelli basandosi su fattori quali: hardware, localizzazione geografica dei data center, durata del training e la fonte di energia utilizzata. Le emissioni di carbonio variano significativamente in base alla fonte energetica locale, ad esempio, data center alimentati principalmente da energie rinnovabili producono emissioni minori rispetto strutture in aree alimentate da combustibili fossili. Anche l'efficienza energetica dell'hardware gioca un ruolo cruciale: le CPU di ultima generazione risultano molto più efficienti delle CPU tradizionali. Nonostante l'importanza dei risultati, lo studio presenta una limitazione significativa: si focalizza esclusivamente sulla fase di addestramento, trascurando l'analisi della fase di inferenza, e utilizza come unità di misura le emissioni di carbonio, fortemente influenzate da fattori esterni quali la localizzazione geografica e l'efficienza della rete elettrica. Al contrario, il mio studio analizza anche la fase di inferenza ed inoltre adotta come metrica il consumo energetico in joule, una misura diretta e indipendente da variabili esterne. In questa direzione, l'articolo di Järvenpää et al [9], che esplora le diverse strategie architetturali per migliorare l'efficienza energetica e ridurre l'impatto ambientale dei sistemi di ML, evidenzia che la fase di training è generalmente quella più energivora all'interno del ciclo di vita dei modelli ML, un aspetto che si riconferma nel mio studio, in cui emerge che, in media, la fase di addestramento consuma più energia rispetto alla fase di inferenza. Inoltre l'articolo sottolinea l'importanza di scegliere algoritmi che siano efficienti dal punto di vista energetico, ma che al contempo riescano a raggiungere gli obiettivi desiderati, senza compromettere le performance. Ad esempio, algoritmi come il K-nearest neighbor (KNN) consumano più energia rispetto al Random Forest (RF), senza però garantire prestazioni superiori. Il mio studio conferma questa osservazione, evidenziando che, nonostante il KNN richieda un maggiore dispendio energetico, il RF offre comunque prestazioni superiori in termini di accuratezza, dimostrando che un algoritmo più efficiente energeticamente può anche ottenere risultati migliori. L'articolo di Joel Castaño et al. [10] analizza la

correlazione tra dimensione dei modelli, dimensione dei dataset e consumo energetico per dei modelli di ML presi da Hugging Face (piattaforma nota principalmente per il suo repository di modelli pre-addestrati e strumenti open source). Attraverso un'analisi dei dati estratti dal repository, viene evidenziato che solo una minima parte dei modelli riporta le emissioni di carbonio, tuttavia viene riportata una lieve diminuzione nel consumo energetico medio riportato negli ultimi anni, suggerendo una graduale adozione di pratiche più sostenibili. Vengono adottati strumenti della letteratura per calcolare il consumo energetico, tra questi CodeCarbon. Inoltre viene evidenziato che la dimensione del dataset ha un impatto significativo sul consumo energetico durante l'addestramento dei modelli ed il mio studio non solo conferma la correlazione tra la dimensione del dataset e consumo energetico ma utilizza proprio CodeCarbon come tracker per i consumi. Un'ulteriore lavoro rilevante è quello di Castanyer et al. [11], che analizza le decisioni progettuali nei modelli di DL integrati in applicazioni mobili, esaminando come il numero di parametri, il tipo di architettura e il dataset influenzino il compromesso tra accuratezza, peso del modello e utilizzo delle risorse computazionali. Nello studio si è osservato che il peso del modello aumenta linearmente con il numero di parametri; tuttavia l'accuratezza non segue una relazione lineare, indicando che un aumento della complessità non garantisce necessariamente migliori prestazioni. Lo studio include molteplici misurazioni, tra queste tempo CPU e peso del modello, ed analizza il comportamento dei modelli in esecuzione su dispositivi mobili. Tale metodologia è rilevante per il mio lavoro, in quanto anche il mio studio è stato condotto su un dispositivo mobile e prevede la misurazione di metriche analoghe; tuttavia si focalizza sui modelli di ML classici. Un ulteriore contributo nell'ambito del machine learning sostenibile è rappresentato dallo studio di Santos et al. [12] che analizza sei algoritmi, tra cui Logistic Regression, Support Vector Machine, Random Forests, Gradient Boosting, Residual Neural Networks e Multilayer Neural Networks. Gli esperimenti vengono eseguiti su tre dataset di classificazione binaria, misurando il consumo energetico avvalendosi della libreria CodeCarbon, e valutando l'accuratezza dei modelli. Tra i vari risultati, oltre che a considerazioni su tecniche di pre-processing, viene evidenziato che il Random Forest è il modello più performante in termini di accuratezza, ed anche nel mio caso tale algoritmo ha ottenuto ottimi risultati in termini di accuratezza in due casi su tre,

confermando il suo potenziale in vari contesti. Il mio studio si distingue dal lavoro di Santos [12] et al. per il maggiore focus sulla fase di inferenza e sull'analisi di variabili computazionali distinte, come l'utilizzo della CPU, della RAM, il peso del modello. Il lavoro di Yuan et al. [7] esplora l'efficacia della Knowledge Distillation come tecnica per migliorare l'efficienza energetica e le prestazioni dei modelli NLP di grandi dimensioni. La Knowledge Distillation è un processo che permette di trasferire le conoscenze da un modello complesso e ingombrante a una versione più compatta e leggera. Lo studio esamina l'impatto di tale tecnica sul consumo energetico e sulla performance dei modelli. I risultati dello studio mostrano che i modelli distillati consumano meno energia rispetto ai modelli originali; inoltre mostrano un tempo di inferenza significativamente più breve ed una riduzione dell'utilizzo della CPU e della memoria. Dunque questo lavoro, a differenza del mio, valuta tecniche per ridurre il consumo energetico, utilizza modelli di deep learning ed infine non svolge gli esperimenti su dispositivi mobili, o che comunque hanno hardware e prestazioni limitate. Nonostante ciò i due studi utilizzano le stesse variabili dipendenti (consumo energetico, utilizzo CPU, utilizzo memoria, tempo inferenza), e mirano seppur in modo diverso a ridurre i consumi energetici ed il conseguente impatto ambientale dell'AI. L'ultimo lavoro analizzato è quello di R. Verdecchia et al. [6], che si concentra sull'analisi dell'efficienza energetica dei modelli di ML, con particolare attenzione al consumo di energia durante il training, esplorando come l'accuratezza sia correlata al consumo energetico. Viene eseguito un benchmark che analizza questa correlazione, con l'obiettivo di comprendere se modelli più accurati consumino necessariamente più energia e quale sia l'efficienza energetica in relazione alla loro performance. I risultati emersi evidenziano che, in generale, i modelli più accurati tendono a consumare più energia durante il training, in quanto richiedono una maggiore capacità computazionale. Tuttavia, la relazione tra accuratezza e consumo energetico non è lineare, poiché diversi modelli con la stessa accuratezza possono presentare consumi energetici differenti. Un altro risultato importante dello studio riguarda l'influenza delle dimensioni del modello sul consumo energetico. È stato osservato che modelli con un numero maggiore di parametri tendono a richiedere più energia per essere addestrati. Tuttavia, Verdecchia nota che non sempre modelli più grandi portano a migliori performance energetiche, e che in alcuni casi modelli più piccoli e meno

complessi possono ottenere risultati comparabili con un consumo inferiore. Nonostante i risultati offrono spunti utili per comprendere la relazione tra accuratezza e consumo energetico, lo studio presenta alcune limitazioni. Un punto critico dello studio è l'utilizzo di un singolo dataset, approccio che limita la generalizzabilità dei risultati, poiché i modelli di ML possono comportarsi in modo diverso su dataset con caratteristiche e complessità differenti. Inoltre, il lavoro si concentra quasi esclusivamente sul consumo energetico durante la fase di training, trascurando altre fasi del ciclo di vita di un modello, come l'inferenza. Il mio studio si distingue per un approccio più ampio e dettagliato, offrendo maggiore generalizzabilità grazie all'analisi di sei modelli di ML su tre dataset distinti; inoltre esamina anche la fase di inferenza, correlando consumo energetico, accuratezza ed altre variabili come peso del modello e l'utilizzo della CPU. Infine conducendo l'analisi su un dispositivo mobile, il mio studio acquisisce una maggiore rilevanza pratica.

CAPITOLO 3

Metodologia

3.1 Obbiettivi di ricerca

Lo studio in questione si concentra sull'analisi del consumo energetico dei modelli di machine learning durante la fase di inferenza, utilizzando un dispositivo mobile. I modelli scelti sono stati valutati su più dataset, analizzando metriche come accuratezza, consumo energetico, utilizzo della CPU, della memoria e tempi di inferenza. Questo approccio mira a comprendere come tali metriche varino in relazione al dataset e a identificare eventuali correlazioni tra ciascuna di esse e l'accuratezza. Inoltre, particolare attenzione è dedicata al confronto tra i consumi energetici durante l'inferenza e quelli rilevati nella fase di training, ponendo lo studio in continuità con quello di Verdecchia et al [6]. Tuttavia, mentre Verdecchia et al [6] si concentra esclusivamente sul consumo energetico nella fase di training e utilizza un unico dataset, il presente lavoro amplia tale analisi includendo più dataset e considerando ulteriori metriche rilevanti, come il tempo di inferenza, il peso del modello e l'utilizzo delle risorse hardware. Per guidare questa ricerca e chiarire al meglio i risultati ottenuti, sono state formulate le seguenti domande di ricerca:

Q RQ₁. *Come gli algoritmi di ML differiscono in fase di training su diversi dataset?*

Questa domanda consente di replicare ed estendere l’approccio di Verdecchia et al. [6], considerando diversi dataset, fornendo una visione più ampia e rappresentativa delle prestazioni energetiche degli algoritmi di ML in questa fase.

Q RQ₂. *Come gli algoritmi di ML differiscono in fase di inferenza?*

Si punta a comprendere come l’inferenza contribuisca all’impatto energetico complessivo dei modelli di ML, affrontando un’importante lacuna nella letteratura esistente. Inoltre, saranno analizzate altre metriche, quali il peso del modello, l’utilizzo della RAM, l’utilizzo della CPU e il tempo di inferenza.

Q RQ₃. *Quali sono i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML su dataset eterogenei?*

L’obiettivo è sintetizzare e confrontare i risultati ottenuti dalle prime due domande di ricerca, identificando i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse per ciascun algoritmo analizzato. Il confronto si basa sulle performance osservate sui tre dataset utilizzati, eterogenei tra loro, con l’intento di descrivere quali algoritmi si comportano meglio in base alla tipologia di dataset.

3.2 Selezione dei modelli

Il machine learning si divide in quattro principali categorie [1], tra queste il Supervised Learning che utilizza dati di training composti da caratteristiche indipendenti e variabili target; l’obiettivo è prevedere etichette (classificazione) o valori (regressione) per nuovi dati, basandosi sui pattern rilevati durante l’addestramento. All’interno del supervised learning, trattato nello studio, ci sono svariati modelli [13], ciascuno con caratteristiche e applicazioni specifiche. Sono stati scelti i modelli descritti di seguito poiché sono stati impiegati anche nello studio di Verdecchia et al [6], che rappresenta un punto di riferimento rilevante per le analisi sul consumo energetico. Ecco una panoramica concisa dei suddetti modelli:

- **Decision Tree:** è un modello interpretabile che suddivide i dati in base a regole gerarchiche, valutando soglie specifiche su una o più caratteristiche. Viene utilizzato sia per classificazione che per regressione;

- **AdaBoost:** è un algoritmo di ensemble (combina le previsioni di più modelli base) che crea una serie di classificatori deboli, come piccoli alberi decisionali, migliorandoli iterativamente;
- **Bagging Classifier:** utilizza campioni casuali con sostituzione dal dataset originale per addestrare più modelli base. Questa tecnica riduce la varianza e aiuta a prevenire l'overfitting (apprendere troppo dai dati di training), specialmente su modelli instabili;
- **K-Nearest Neighbours (KNN):** è un algoritmo non parametrico che assegna un'etichetta a un'istanza basandosi sulla classe predominante tra i k vicini più vicini, utilizzando metriche di distanza come la distanza euclidea. Sebbene semplice e intuitivo, KNN può diventare inefficiente su dataset di grandi dimensioni;
- **Random Forest:** è un algoritmo di ensemble che costruisce una moltitudine di alberi decisionali su diversi sottoinsiemi del dataset. Ogni albero considera un sottoinsieme casuale delle caratteristiche per creare le divisioni. La previsione finale si basa sulla maggioranza delle decisioni degli alberi o sulla media. Questo metodo riduce il rischio di overfitting tipico degli alberi singoli, ma è più costoso in termini di memoria e tempo computazionale;
- **Support Vector Machines (SVM):** sono algoritmi progettati per trovare un iperpiano ottimale che separa le classi con il margine massimo (distanza tra l'iperpiano e i punti più vicini di ciascuna classe, chiamati support vectors).

3.3 Metriche di confronto

Per la valutazione delle prestazioni dei modelli di ML c'è la necessità della scelta di metriche appropriate [14]. Quando vengono sviluppati nuovi modelli di ML, è fondamentale confrontarli con quelli già esistenti per consentire di scartare i modelli che non danno buoni risultati ed ottimizzare quelli promettenti. L'accuratezza di un modello viene valutata attraverso diverse metriche, ognuna delle quali fornisce una prospettiva unica sulla qualità delle predizioni. Queste metriche derivano dalla

matrice di confusione, che riassume i risultati delle predizioni del modello in termini di veri positivi (TP) ovvero istanze che il modello ha classificato correttamente come positive, veri negativi (TN) ovvero istanze classificate correttamente come negative, falsi positivi (FP) ovvero istanze classificate come positive, ma che in realtà sono negative ed infine falsi negativi (FN) ovvero istanze classificate come negative, ma che in realtà sono positive. Tra le principali metriche dell'accuratezza di un modello [14] c'è l'**Accuracy**, che misura la percentuale di predizioni corrette sul totale.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3.1)$$

Oltre alle metriche tradizionali di valutazione, sono fondamentali, per giudicare la qualità delle previsioni di un modello, le metriche relative alle risorse computazionali necessarie per l'esecuzione del modello [6]. Tra queste, trattate nello studio, vi sono:

- **Peso del modello:** si riferisce alla dimensione del file del modello, misurata in kilobyte. Modelli più leggeri possono essere preferiti per ridurre l'uso di memoria e facilitare il caricamento e l'esecuzione su dispositivi con risorse limitate, come smartphone o dispositivi IoT [15].
- **Tempo di inferenza:** rappresenta il tempo impiegato dal modello per generare una previsione dopo aver ricevuto un input. È misurato in millisecondi e calcolato come:

$$T_{\text{inferenza}} = T_{\text{fine}} - T_{\text{inizio}} \quad (3.3.2)$$

- **Utilizzo della memoria:** misura la percentuale della memoria di sistema utilizzata dal modello durante la sua operazione. È calcolato come:

$$\text{Memoria} = \frac{\text{Memoria utilizzata}}{\text{Memoria totale}} \times 100 \quad (3.3.3)$$

- **Utilizzo della CPU:** rappresenta la percentuale di tempo in cui la CPU è attivamente coinvolta nell'esecuzione di istruzioni, rispetto al tempo di inattività calcolato come:

$$\text{CPU} = \frac{\text{Tempo attivo CPU}}{\text{Tempo totale}} \times 100 \quad (3.3.4)$$

- **Consumo energetico:** rappresenta l'energia totale utilizzata durante l'esecuzione di un'operazione, misurata in joule. La potenza consumata è espressa in kWh e successivamente convertita in joule utilizzando la relazione:

$$E = P \cdot 1000 \cdot 3600 \quad (3.3.5)$$

dove P è l'energia consumata in kWh, il fattore 1000 converte i kilowattora in wattora (Wh), e il fattore 3600 converte le ore in secondi. Il risultato, E , è l'energia totale consumata in joule.

3.4 Selezione dei dataset

La selezione dei dataset rappresenta un elemento cruciale per valutare le prestazioni dei modelli di ML, poiché la loro varietà in termini di caratteristiche, dimensioni e complessità consente di analizzare come le metriche di performance variano al variare dei dati. I dataset utilizzati in questo studio sono stati scelti per la loro rilevanza nella letteratura e per la possibilità di confrontare i risultati con studi precedenti. Il dataset **SmsSpam** è stato selezionato per coerenza con lo studio di Verdecchia et al. [6], poiché consente un confronto diretto; mentre i dataset **Compas** e **German** sono stati scelti per la loro diffusione e rilevanza nella letteratura accademica; in particolare, come evidenziato nello studio di Chen et al. [16], questi dataset sono ampiamente utilizzati per analizzare problematiche relative a bias e fairness, rendendoli ideali anche per esplorare altre problematiche. I dettagli sui dataset selezionati sono riportati nella Tabella 3.1.

Tabella 3.1: Caratteristiche dei dataset utilizzati nello studio

Dataset	Numero di entry	Distribuzione delle classi	Descrizione
Compas	7.214	55% recidivo, 45% non recidivo	Utilizzato per studiare la recidività e analizzare bias ed equità nei sistemi ML. Ogni entry rappresenta un individuo con svariate informazioni per valutare la recidività.
German	1.000	70% buon pagatore, 30% cattivo pagatore	Utilizzato per analizzare il rischio di credito nei richiedenti prestiti bancari. Contiene attributi demografici, finanziari e comportamentali per predire se un individuo è un buon o cattivo creditore.
SmsSpam	5.574	85% ham, 15% spam	Contiene messaggi SMS etichettati come spam o non spam. Utilizzato per la classificazione del testo per predire se un messaggio è indesiderato. Le colonne principali sono "Label" (etichetta) e "Message" (testo).

3.5 Ambiente di Esecuzione e Metodologia di Test

Per il presente studio, l'ambiente di esecuzione è stato lo smartphone **Samsung Galaxy A70**, dotato di 6 GB di RAM e 128 GB di memoria interna. Questo modello, rilasciato nel 2022, è un dispositivo di fascia media che offre prestazioni sufficienti per eseguire operazioni di calcolo complesse. Sul dispositivo è stato installato **Kali Linux**¹, una distribuzione di Linux nota per l'ampia gamma di strumenti di sviluppo. Questa configurazione ha permesso di sfruttare la potenza di calcolo dello smartphone in modo più efficiente rispetto a un sistema operativo mobile standard, consentendo l'esecuzione degli esperimenti direttamente sul dispositivo. Il lavoro svolto si articola in due fasi principali: il **training** e l'**inferenza**, entrambe implementate in Python con l'ausilio di librerie quali **scikit-learn** per la costruzione e l'addestramento dei modelli, **psutil** per il monitoraggio delle risorse hardware, **CodeCarbon**² per la stima del consumo energetico, e strumenti di analisi come **pandas**, **NumPy**, **Matplotlib** e **Seaborn** per l'elaborazione e la visualizzazione dei dati.

3.6 Training

Nella fase di training, sei algoritmi di classificazione (Sezione 3.2) sono stati addestrati utilizzando ciascuno dei dataset selezionati (Sezione 3.4). Per garantire una valutazione accurata, le feature e la variabile target sono state suddivise in dati di training e test mediante la funzione `train_test_split` di `scikit-learn`, con una proporzione del 70% per il training e il 30% per il test. Gli algoritmi considerati, implementati utilizzando i moduli specifici di `scikit-learn`, sono stati addestrati con la funzione `fit`. L'accuratezza dei modelli sui dati di test è stata valutata utilizzando la funzione `accuracy_score`. Durante l'addestramento, il consumo energetico è stato monitorato in tempo reale tramite la libreria `CodeCarbon`, utilizzando un oggetto `EmissionsTracker`. Le funzioni `start` e `stop` sono state rispettivamente utilizzate per avviare e concludere il monitoraggio, con il calcolo del consumo totale in kWh, successivamente convertito in Joule. Per ridurre l'in-

¹<https://www.kali.org/>

²<https://codecarbon.io>

fluenza della temperatura iniziale del dispositivo e migliorare l'accuratezza delle misurazioni, è stata eseguita una fase di riscaldamento del dispositivo per 5 minuti. Questa fase ha coinvolto l'esecuzione di un algoritmo di generazione della sequenza di Fibonacci, e solo successivamente è iniziata la fase di training, durante la quale sono state registrate le seguenti informazioni per ciascun algoritmo:

- Consumo energetico totale in Joule;
- Tempo di training in secondi;
- Accuratezza.

I modelli addestrati sono stati salvati in file separati per consentirne l'utilizzo degli stessi per la fase di inferenza senza necessità di riaddestramento.

3.7 Inferenza

La fase di inferenza è stata progettata per valutare il consumo di risorse in scenari di utilizzo reale. I modelli addestrati nella fase precedente sono stati caricati e utilizzati per effettuare predizioni sui dati di test tramite la funzione `predict` di `scikit-learn`. L'obiettivo principale di questa fase è analizzare il comportamento dei modelli in termini di consumo energetico, utilizzo delle risorse hardware, tempo di esecuzione e accuratezza. Durante ogni esecuzione, sono state monitorate le seguenti metriche:

- **Consumo energetico:** anche in questa fase, un oggetto `EmissionsTracker` è stato impiegato per tracciare il consumo energetico in tempo reale;
- **Utilizzo della CPU:** registrato come percentuale di utilizzo medio durante l'inferenza, tramite la funzione `cpu_percent` della libreria `psutil`.
- **Utilizzo della memoria RAM:** calcolato in percentuale rispetto alla memoria disponibile, utilizzando la funzione `virtual_memory` della stessa libreria.
- **Tempo di inferenza:** determinato misurando l'intervallo tra l'inizio e la fine dell'inferenza, con la funzione `time.time`.

- **Peso del modello:** misurato in kilobyte, tramite la funzione `os.path.getsize`, che restituisce la dimensione del file contenente il modello addestrato.
- **Accuratezza:** calcolata utilizzando il metodo `score` di `scikit-learn`, per confrontare le predizioni effettuate con i valori effettivi della variabile target nei dati di test.

Per ogni modello sono stati effettuati 30 run, per un totale di 180 esperimenti per ogni dataset (30 run per ciascuno dei 6 algoritmi distinti). Le predizioni sono state effettuate sui dati di test messi da parte prima dell'addestramento (30% di ogni dataset) tramite `train_test_split`. Le esecuzioni sono state eseguite in ordine randomizzato, ovvero non consecutivo tra i vari modelli, per minimizzare l'impatto di variabili esterne come fluttuazioni temporanee nell'uso delle risorse del dispositivo o picchi di attività del sistema. Inoltre, per ridurre gli effetti di eventuali residui di carico dal run precedente, è stata inserita una pausa di raffreddamento di 60 secondi tra esecuzioni consecutive, migliorando così la precisione delle misurazioni.

3.8 Analisi dei dati

Nel presente studio, per ciascuno dei sei modelli di machine learning analizzati, sono stati eseguiti 30 run distinti per la fase di training e altrettanti per la fase di inferenza. Questo processo è stato ripetuto su ciascuno dei tre dataset considerati. Complessivamente, ciò ha portato alla realizzazione di 540 esperimenti per la fase di training (6 algoritmi * 30 run * 3 dataset), ed altrettanti esperimenti per la fase di inferenza; per un totale di 1080 esperimenti. I valori raccolti sono stati sintetizzati calcolando la media aritmetica, che consente di ottenere un dato rappresentativo riducendo l'influenza di variazioni casuali e garantendo maggiore affidabilità e stabilità nei risultati. I valori medi sono stati successivamente impiegati per generare diversi tipi di grafici, tra cui:

- **Grafici a barre**, utili per confrontare metriche come il consumo energetico tra i vari algoritmi e dataset, offrendo una visione chiara e immediata delle differenze;

- **Grafici a barre con valori di accuratezza sovrapposti**, che permettono di evidenziare i compromessi tra le prestazioni e i costi computazionali;
- **Matrici di correlazione**, impiegate per studiare le relazioni tra le variabili misurate;
- **Heatmap**, utilizzate per rappresentare graficamente i valori medi delle metriche analizzate, evidenziando in modo intuitivo i pattern e le variazioni tra modelli e dataset.

Infine per la conservazione e l'accessibilità dei risultati, tutti i dati raccolti e calcolati sono stati esportati in file CSV, un formato semplice e ampiamente compatibile, rendendo i dati facilmente disponibili per analisi future. Tutto il codice sorgente e i dati sono disponibili nella repository GitHub³.

³https://github.com/russosamuele/SustainablePractises_SmartphoneVersion

CAPITOLO 4

Analisi dei risultati

Nel seguente capitolo sono stati riportati e analizzati i risultati ottenuti, al fine di rispondere alle domande di ricerca discusse nella metodologia.

4.1 RQ1: Come gli algoritmi di ML differiscono in fase di training su diversi dataset?

Lo studio di Verdecchia et al. [6] ha evidenziato significative differenze nei consumi energetici durante il training degli algoritmi di ML: il Random Forest si è rivelato l'algoritmo più energivoro, con un consumo mediano di 1.98 Joule per run, seguito da AdaBoost, che consuma meno della metà (48.9%) dell'energia richiesta dal Random Forest. Al contrario, il KNN è risultato l'algoritmo più efficiente, con un consumo mediano di appena 0.01 Joule, seguito dall'algoritmo Decision Tree, che richiede 0.12 Joule. Per ampliare i risultati dello studio sopra citato, è stato analizzato il consumo energetico di sei algoritmi di ML durante la fase di training su tre dataset differenti: Compas, German e SmsSpam. Di seguito vengono mostrati e analizzati i consumi energetici misurati per ogni dataset.

4.1 – RQ1: Come gli algoritmi di ML differiscono in fase di training su diversi dataset?

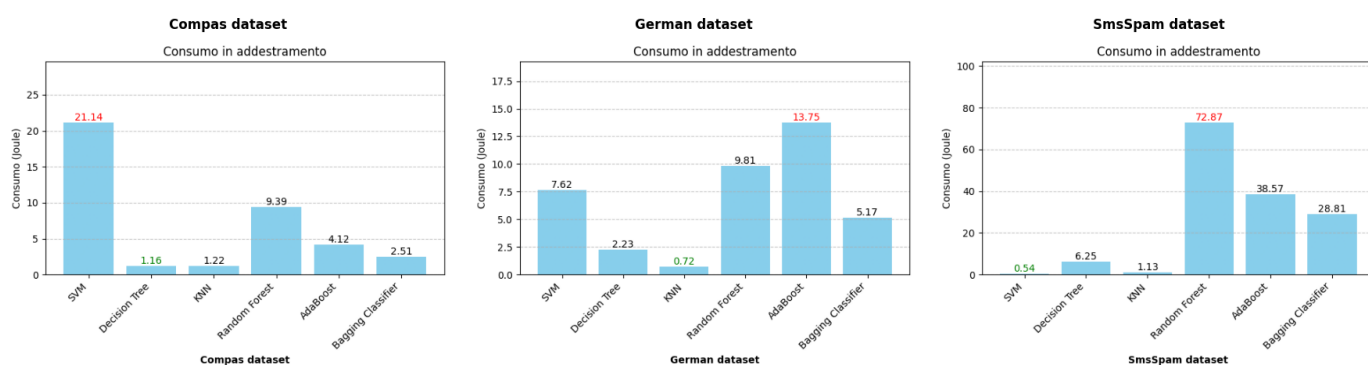


Figura 4.1: Confronto dei consumi energetici dei modelli in fase di training sui tre dataset.

Come mostrato nella Figura 4.1 nel caso del dataset Compas, composto da 7.214 entry con una distribuzione delle classi bilanciata (55% recidivi, 45% non recidivi), si osservano consumi energetici significativamente diversi tra i modelli. Ad esempio, l'SVM emerge come il più dispendioso, con un consumo di 21.14 Joule, un valore attribuibile alla complessità del modello che deve adattarsi alla varietà degli attributi presenti. Anche il Random Forest, con un consumo di 9.39 Joule, riflette i costi legati alla costruzione e gestione di molteplici alberi decisionali. Al contrario, il KNN e il Decision Tree registrano consumi notevolmente inferiori, rispettivamente 1.22 Joule e 1.16 Joule, suggerendo che la loro semplicità strutturale permetta di gestire meglio il dataset senza un eccessivo dispendio energetico. Per quanto riguarda il dataset German, caratterizzato da uno squilibrio delle classi (70% buon pagatore, 30% cattivo pagatore) e da una dimensione ridotta di 1.000 entry, i consumi energetici mostrano un comportamento differente. AdaBoost, che enfatizza gli errori generati durante il training, consuma 13.75 Joule, rendendolo il modello più energivoro per questo dataset. Anche il Random Forest presenta un consumo elevato di 9.81 Joule, simile al Compas, probabilmente per la necessità di adattarsi a uno squilibrio che aumenta il costo computazionale. Tuttavia, il KNN mantiene la sua efficienza, registrando il valore più basso (0.72 Joule), mentre il Decision Tree si attesta su 2.23 Joule, evidenziando come modelli più semplici continuino a essere avvantaggiati in contesti con un numero ridotto di osservazioni. Infine, nel caso del dataset SmsSpam, con 5.574 entry e una distribuzione fortemente sbilanciata (85% ham, 15% spam), emergono

4.1 – RQ1: Come gli algoritmi di ML differiscono in fase di training su diversi dataset?

differenze marcate nei consumi. Il Random Forest risulta il modello più energivoro, con un consumo impressionante di 72.87 Joule, riflettendo l'impatto combinato della complessità del modello e della dimensione elevata del dataset. AdaBoost, con 38.57 Joule, e Bagging Classifier, con 28.81 Joule, evidenziano anch'essi consumi elevati, sebbene inferiori rispetto al Random Forest. Al contrario, modelli più leggeri come il KNN (1.13 Joule) e l'SVM (0.54 Joule) registrano valori significativamente inferiori, mostrando come la natura testuale e la ridotta complessità intrinseca del dataset possano favorire algoritmi meno complessi. Lo studio conferma quanto riportato da Verdecchia et al. [6], secondo cui il KNN è, in fase di training, tra i modelli più efficienti dal punto di vista energetico, difatti su tutti i dataset analizzati, il KNN ha registrato consumi inferiori a 1.5 Joule. Stesso discorso vale per il Decision Tree, che si è dimostrato molto efficiente. Al contrario, il Random Forest non ha mostrato buoni risultati, evidenziando inefficienze energetiche rispetto ad altri modelli, riconfermando lo studio in questione. Invece l'SVM, ha mostrato una grande variabilità; comportamento che sottolinea come il consumo energetico dell'SVM potrebbe essere fortemente influenzato dal dataset. Tuttavia, rispetto ai risultati ottenuti da Verdecchia et al. [6], i consumi energetici registrati nel seguente studio appaiono decisamente superiori. Questa differenza può essere attribuita alle caratteristiche hardware del dispositivo utilizzato, difatti avendo eseguito lo studio su uno smartphone, anziché su un PC, potrebbe aver influito sulle misurazioni, portando a consumi energetici maggiori. Per una sintesi più chiara e immediata dei consumi energetici misurati durante la fase di training, i risultati ottenuti, estratti dai grafici sopra rappresentati, sono riportati nella Tabella 4.1.

✚ **Answer to RQ₁.** Dall'analisi complessiva emerge chiaramente che il consumo energetico degli algoritmi di ML dipende non solo dall'algoritmo, ma anche dalle caratteristiche specifiche del dataset. Dataset con distribuzioni delle classi sbilanciate, come SmsSpam, o con una maggiore dimensionalità, come Compas, richiedono maggiori consumi energetici dei modelli usati. Modelli semplici come KNN e Decision Tree si confermano invece generalmente più efficienti su tutti i dataset analizzati.

Tabella 4.1: Consumo energetico mediano (Joule) durante la fase di training.

Algoritmo	Compas	German	SmsSpam
SVM	21.14	7.62	0.54
Decision Tree	1.16	2.23	6.25
KNN	1.22	0.72	1.13
Random Forest	9.39	9.81	72.87
AdaBoost	4.12	13.75	38.57
Bagging Classifier	2.51	5.17	28.81

4.2 RQ2: Come gli algoritmi di ML differiscono in fase di inferenza?

In questa sezione, esploreremo come i diversi algoritmi di ML si comportano durante la fase di inferenza, analizzando le principali metriche di valutazione su ogni dataset. Utilizzando heatmap che aggregano i risultati per ogni dataset, sarà possibile confrontare simultaneamente il consumo energetico, il peso del modello, il tempo di inferenza, l'utilizzo della CPU e della memoria. Seguono i risultati ottenuti dai modelli sul dataset Compas (Figura 4.2), con le sue 7214 entry e una distribuzione bilanciata tra recidivi (55%) e non recidivi (45%).

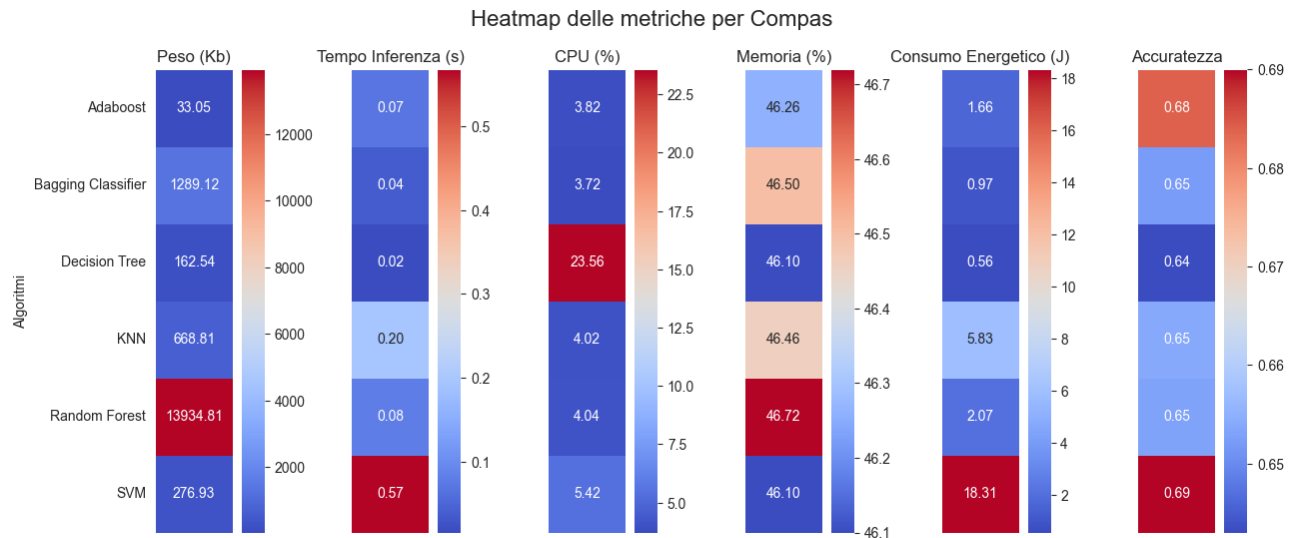


Figura 4.2: Risultati ottenuti dai modelli sul dataset Compas

- SVM ha mostrato un'accuratezza di 0.69, con un consumo energetico relativamente alto (18.31 J), probabilmente a causa della complessità del modello e delle sue operazioni computazionali. L'utilizzo della CPU è stato moderato, ma il tempo di inferenza (0.57 s) indica un'elevata complessità computazionale. Il peso del modello (276.93 KB) è relativamente contenuto, mentre l'utilizzo della memoria è stato superiore rispetto ad altri modelli. Questo suggerisce che SVM potrebbe essere più sensibile alla dimensione del dataset e alla sua distribuzione, evidenziando il compromesso tra accuratezza e efficienza energetica;
- Decision Tree, con un'accuratezza di 0.64, ha mostrato una straordinaria efficienza energetica (0.56 J) e un tempo di inferenza molto rapido (0.02 s), il che lo rende un modello estremamente competitivo in termini di risorse. L'utilizzo della CPU, sebbene alto, non ha compromesso le prestazioni. Il peso del modello è di soli 162.54 KB, un dato che suggerisce una struttura meno complessa rispetto a modelli come l'SVM, contribuendo a una minore necessità di memoria e a un'inferenza più veloce;
- KNN, con un'accuratezza di 0.65, ha consumato 5.83 J e ha richiesto 0.2 s per l'inferenza. Il peso del modello (668.81 KB) è tra i più alti, e ciò potrebbe spiegare il maggiore consumo energetico rispetto a Decision Tree, ma anche il maggior utilizzo della memoria. L'efficienza di KNN potrebbe essere influenzata dalla

dimensione del dataset e dalla sua necessità di memorizzare un grande numero di istanze per calcolare le predizioni. Sebbene sia relativamente poco efficiente in termini di energia, la sua performance accettabile potrebbe essere legata alla sua capacità di adattarsi facilmente a dataset con relazioni meno complesse;

- Random Forest ha dato un'accuratezza del 0.65, con un consumo energetico moderato (2.07 J) e un tempo di inferenza di 0.08 s. Nonostante un peso modello elevato (13,934.81 KB), il modello ha mostrato una buona bilanciatura tra accuratezza e utilizzo delle risorse. L'utilizzo della CPU è stato elevato, ma ciò non ha pregiudicato l'efficienza in termini di tempo di inferenza;
- AdaBoost ha ottenuto un'accuratezza di 0.68, con un consumo energetico contenuto (1.66 J) e un tempo di inferenza rapido (0.07 s). Il peso del modello (33.05 KB) è molto basso, indicando una struttura leggera. AdaBoost si è comportato molto bene in Compas, dove la necessità di un modello robusto ma veloce è importante. L'utilizzo della CPU è stato contenuto, a dimostrazione della sua efficienza in scenari con risorse limitate;
- Bagging Classifier ha dato un'accuratezza di 0.65, con un consumo energetico basso (0.97 J) e tempi di inferenza molto rapidi (0.04 s). Nonostante un peso di 1289.13 KB, che è tra i più alti dopo KNN e Random Forest, il modello ha mostrato una buona efficienza in termini di utilizzo della CPU e memoria. Questo modello è interessante su questo dataset in quanto offre buone prestazioni senza compromettere troppo le risorse, sebbene la sua accuratezza non sia ai livelli di altri modelli.

Seguono i risultati ottenuti dai modelli sul dataset German (Figura 4.3), che comprende 1000 entry con una distribuzione delle classi di 70% per i "buoni pagatori" e 30% per i "cattivi pagatori".

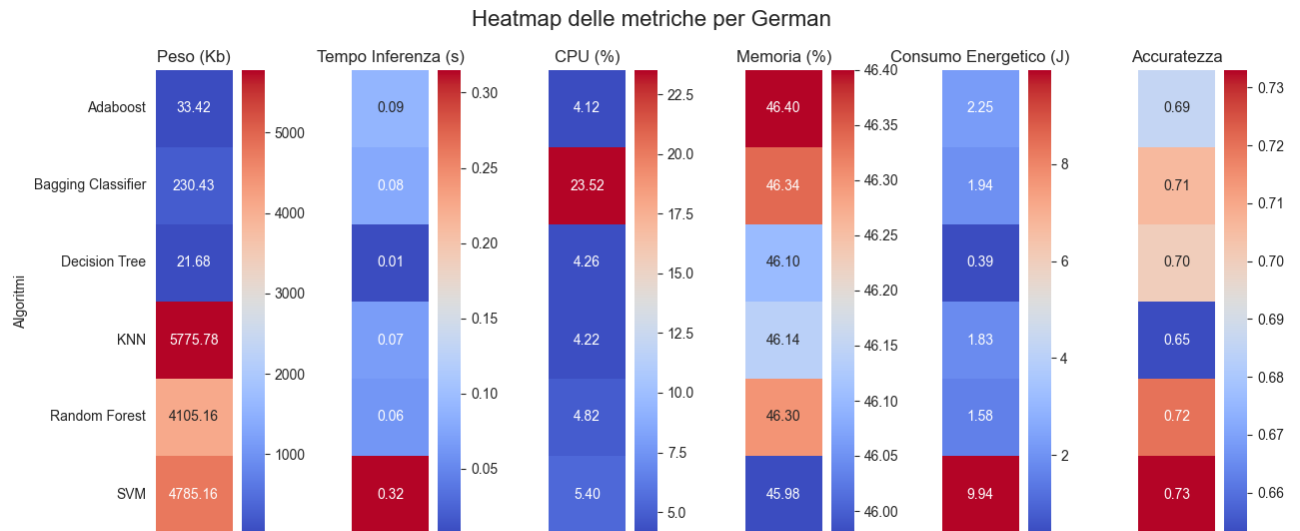


Figura 4.3: Risultati ottenuti dai modelli sul dataset German

La struttura del dataset, con una predominanza di una classe, potrebbe influire sui comportamenti dei modelli, soprattutto in relazione all'accuratezza e all'efficienza.

- SVM, con un'accuratezza di 0.73, ha consumato 9.94 J, il che è abbastanza alto rispetto ad altri modelli nel dataset. L'utilizzo della CPU è stato moderato, ma il tempo di inferenza (0.32 s) è più elevato, suggerendo che SVM non si adatta perfettamente a dataset di dimensioni più contenute come questo. Il peso del modello (4785.16 KB) è piuttosto alto, il che impatta anche sull'utilizzo della memoria;
- Decision Tree, con un'accuratezza di 0.70, ha avuto un consumo energetico molto basso (0.39 J) e un tempo di inferenza rapidissimo (0.007 s), il che lo rende ideale per dataset con un numero limitato di feature come German. Il peso del modello (21.68 KB) e l'utilizzo della CPU sono modesti, rendendolo molto efficiente;
- KNN, con un'accuratezza di 0.65, ha consumato 1.83 J, con un tempo di inferenza di 0.07 s e un peso di 5775.79 KB. Questo suggerisce che KNN, pur essendo efficiente in termini di accuratezza in alcuni casi, non è il più adatto in contesti dove la memoria e il tempo di inferenza devono essere ottimizzati. L'alto consumo energetico potrebbe essere dovuto alla necessità di gestire una grande quantità di dati per il calcolo delle distanze;

- Random Forest, con un'accuratezza di 0.72, ha mostrato un buon equilibrio con un consumo energetico di 1.58 J e un tempo di inferenza di 0.06 s. Nonostante il peso relativamente alto (4105.17 KB), il modello ha dimostrato una buona performance, anche se il suo utilizzo della CPU è stato maggiore rispetto a Decision Tree;
- AdaBoost, con un'accuratezza di 0.69, ha consumato 2.25 J e ha mostrato un buon tempo di inferenza (0.09 s), ma il suo peso contenuto (33.42 KB) e il basso utilizzo della CPU lo rendono una scelta efficiente. L'efficienza energetica e la rapidità nell'inferenza sono i punti di forza di questo modello sul dataset German;
- Bagging Classifier, con un'accuratezza di 0.71, ha consumato 1.94 J e ha avuto un tempo di inferenza di 0.08 s. Il peso del modello (230.43 KB) è piuttosto contenuto, e l'utilizzo della CPU è stato ben bilanciato. Questo modello in questo caso rappresenta una soluzione bilanciata.

Passiamo all'analisi dei risultati ottenuti dai modelli sul dataset SmsSpam (Figura 4.4).

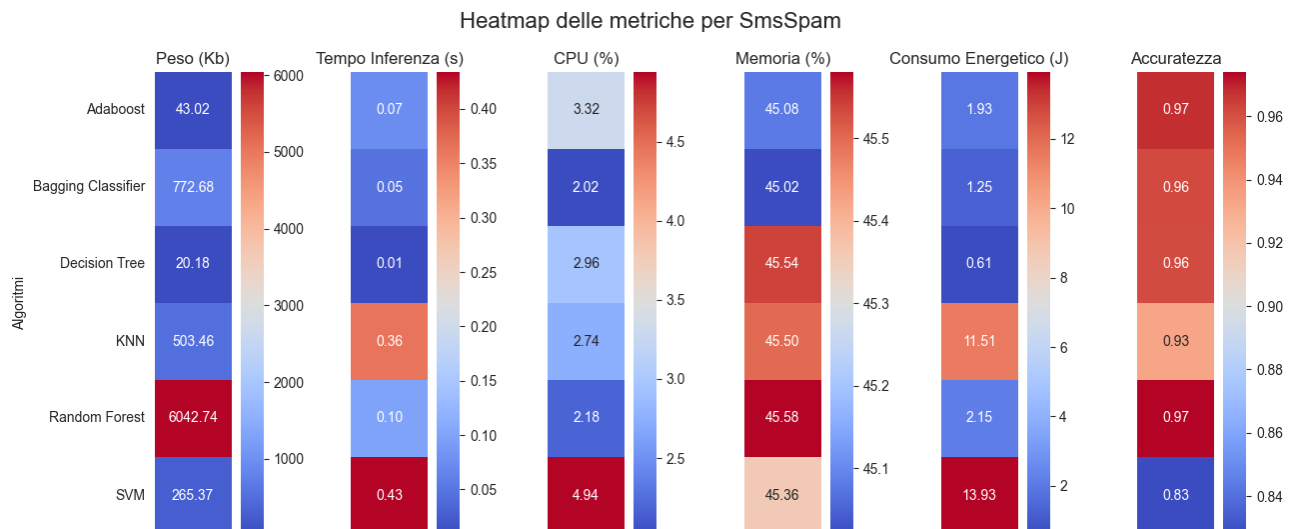


Figura 4.4: Risultati ottenuti dai modelli sul dataset SmsSpam

I risultati ottenuti su questo dataset, con 5.574 entry e una distribuzione delle classi di 85% di messaggi non spam e 15% spam, rivelano diverse dinamiche. C'è

però da considerare che la distribuzione sbilanciata tra le due classi potrebbe aver influenzato la performance dei modelli, in particolare in termini di accuratezza e quindi riguardo la capacità di distinguere correttamente tra ham e spam.

- SVM, con un'accuratezza di 0.83, ha consumato 13.93 J, un dato relativamente alto, con un tempo di inferenza di 0.43 s. Il modello ha anche un peso di 265.37 KB, ma l'utilizzo della CPU è stato moderato, segnalando che SVM potrebbe non essere la scelta migliore per dataset con classi sbilanciate;
- Decision Tree è il modello che mostra la performance più impressionante in termini di efficienza energetica, con un consumo di soli 0,61 Joule. Con un tempo di inferenza rapidissimo (0.001 s) e un basso utilizzo della CPU, il Decision Tree riesce a ottenere un'accuratezza di 0.96, la più alta tra i modelli testati. Il peso del modello (20.18 KB) è estremamente ridotto, contribuendo ulteriormente alla sua efficienza complessiva. È però da considerare che lo sbilanciamento delle classi potrebbe aver creato overfitting;
- KNN mostra un'accuratezza di 0.93, ma un consumo energetico di 11.51 Joule, il lo rende meno efficiente rispetto al Decision Tree. Il tempo di inferenza (0.36 s) è più elevato rispetto ad altri modelli, il che suggerisce che la complessità del calcolo delle distanze tra i punti potrebbe essere la causa di questo maggiore consumo energetico. Il peso del modello (503.46 KB) è anche più alto rispetto agli altri modelli, indicando che il KNN potrebbe richiedere più risorse per gestire il dataset. Nonostante questi svantaggi in termini di efficienza, il modello mantiene una buona accuratezza;
- Random Forest presenta un'accuratezza di 0.97, una delle più alte tra i modelli, ma il suo consumo energetico (2.16 Joule) è più alto rispetto ad altri modelli con performance simili. Il tempo di inferenza (0.1 s) è relativamente contenuto e l'utilizzo della CPU è molto basso, suggerendo che il modello è ben ottimizzato dal punto di vista delle risorse. Il peso del modello (6042.74 KB) è il più elevato, indicando una maggiore complessità. Nonostante l'elevato peso, il Random Forest riesce comunque a bilanciare bene l'accuratezza e l'efficienza energetica;

- AdaBoost, con un'accuratezza di 0.97, consuma 1.93 Joule di energia. Il tempo di inferenza (0.07 s) è relativamente veloce. Nonostante il consumo energetico superiore rispetto ad altri modelli, AdaBoost ottimizza bene l'utilizzo delle risorse, con un peso del modello di 43.02 KB, che lo rende leggero. La sua performance complessiva è buona;
- Bagging Classifier, con un'accuratezza di 0.96, offre un buon compromesso tra prestazioni e utilizzo delle risorse. Con un consumo energetico di 1.25 Joule e un tempo di inferenza di 0.05 s, risulta efficiente. L'utilizzo della CPU è molto contenuto e il peso del modello (772.69 KB) è moderato, suggerendo che il modello è relativamente leggero rispetto ad altri come il Random Forest, pur mantenendo prestazioni competitive.

🔗 **Answer to RQ₂.** Dall'analisi emerge che la scelta del modello dipende fortemente dalle caratteristiche del dataset e dalle esigenze specifiche in termini di efficienza energetica e risorse. L'SVM e il Random Forest offrono un'accuratezza elevata, ma richiedono un notevole consumo energetico e un utilizzo intensivo della CPU e della memoria. Questi modelli si adattano bene a dataset complessi, ma la loro efficienza energetica è limitata, soprattutto quando il dataset cresce in dimensioni. Il KNN si distingue per l'efficienza energetica, in particolare su dataset più semplici, ma la sua performance diminuisce con l'aumento della complessità e delle dimensioni del dataset, mostrando una maggiore richiesta di risorse e un consumo energetico crescente. Il Decision Tree e il Bagging Classifier sono modelli più leggeri e efficienti dal punto di vista energetico. In particolare, il Decision Tree si comporta molto bene in termini di efficienza, ma l'accuratezza tende a diminuire in presenza di dati complessi. Il Bagging Classifier, pur mostrando una buona efficienza, presenta una performance meno consistente rispetto agli altri modelli. AdaBoost emerge come un ottimo compromesso tra efficienza energetica e accuratezza, adattandosi bene a vari tipi di dataset e offrendo prestazioni competitive senza un eccessivo utilizzo delle risorse.

4.3 RQ3: Quali sono i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML su dataset eterogenei?

Tale domanda si propone di analizzare i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di machine learning, confrontando le due fasi operative principali: training e inferenza. Questa domanda nasce dall'osservazione che i modelli si comportano in modo diverso a seconda della fase considerata e delle caratteristiche del dataset. Sul dataset Compas, i modelli mostrano comportamenti distinti per quanto riguarda i compromessi tra accuratezza, consumo energetico e utilizzo delle risorse. Il Decision Tree si distingue per la sua efficienza complessiva, consumando circa 1 Joule in training e 0.56 Joule in inferenza, con tempi di inferenza ridotti a 0.01 secondi e un peso minimo di 21 KB. Tuttavia, la sua accuratezza di 0.64 è leggermente inferiore rispetto agli altri modelli. Modelli più complessi come il Random Forest e l'AdaBoost, pur essendo più energivori in training, riducono significativamente il loro consumo in inferenza (rispettivamente 2.07 e 1.66 Joule), offrendo accuratezze competitive. Il peso del Random Forest (14 MB) rappresenta però un limite per dispositivi con memoria ridotta. L'SVM, pur garantendo l'accuratezza più alta (0.69), richiede oltre 20 Joule in training e 18.3 Joule in inferenza, con tempi di inferenza lunghi (0.56 secondi). Infine, il KNN, con un consumo di poco più di un Joule in training e 5.83 Joule in inferenza, presenta un peso non elevatissimo (0.66 MB) e tempi di inferenza relativamente lunghi (0.19 secondi), limitando la sua scalabilità nonostante l'accuratezza di 0.65. Nel dataset German, i consumi sono generalmente inferiori a quelli osservati su Compas, grazie alla minore complessità e dimensione del dataset. Il Decision Tree si conferma il modello più efficiente, consumando 2.23 Joule in training e 0.39 Joule in inferenza, con tempi di inferenza minimi (0.01 secondi) e un'accuratezza di 0.70. Il Random Forest e l'AdaBoost offrono un compromesso tra accuratezza ed efficienza: pur consumando più di 9 Joule in training, riducono il loro consumo in inferenza (1.58 e 2.25 Joule rispettivamente) e mantengono un'accuratezza di 0.72 e 0.68. L'SVM, con consumi di 7.6 Joule in training e 9.94 Joule in inferenza, ottiene la migliore accuratezza (0.73),

4.3 – RQ3: Quali sono i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML su dataset eterogenei?

ma i suoi requisiti energetici e i tempi di inferenza più elevati (0.30 secondi) lo penalizzano. Il KNN, pur consumando in inferenza 1.83 Joule, richiede 0.07 secondi per ogni inferenza, con un peso di 5.7 MB. Sul dataset SmsSpam, il Decision Tree continua a distinguersi per la sua efficienza: meno di 1 Joule in training, 0.61 Joule in inferenza, e un tempo di inferenza di soli 0.01 secondi, con un'accuratezza elevata di 0.96. I modelli ensemble come l'AdaBoost e il Bagging Classifier, pur consumando oltre 25 Joule in training, ottimizzano il consumo in inferenza (rispettivamente 1.93 e 1.25 Joule) e mantengono accuratezze molto alte (0.96). L'SVM e il KNN, invece, mostrano il compromesso peggiore: l'SVM consuma 0.54 Joule in training e 13.9 Joule in inferenza, con un tempo di inferenza di 0.43 secondi, mentre il KNN consuma 1.13 Joule in training e 11.51 Joule in inferenza, con un tempo di 0.36 secondi. Sebbene offrano accuratezze rispettivamente di 0.83 e 0.93, i loro requisiti in termini di risorse e i tempi d'inferenza lunghi li rendono meno competitivi in contesti con risorse limitate. Per meglio comprendere il compromesso tra consumo energetico in fase di training ed inferenza tra i vari algoritmi sui tre dataset sono stati creati i seguenti grafici:

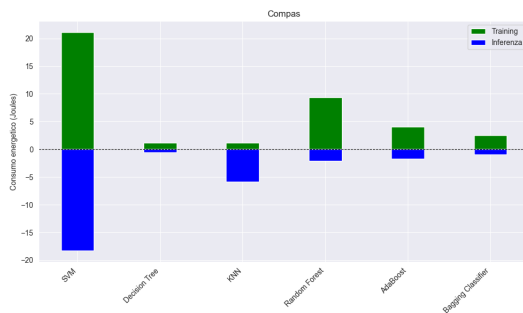


Figura 4.5: Consumi energetici su Compas

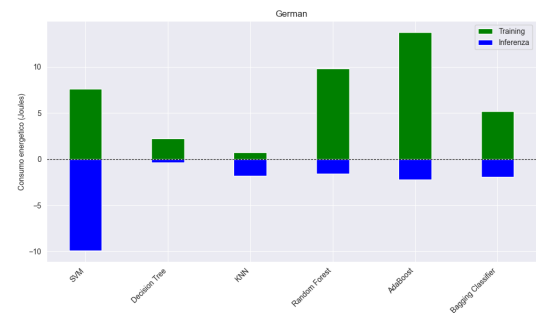


Figura 4.6: Consumi energetici su German

4.3 – RQ3: Quali sono i compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML su dataset eterogenei?

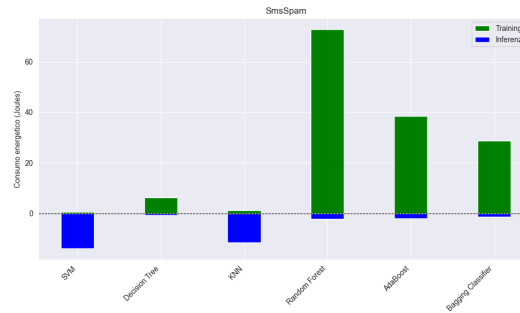


Figura 4.7: Consumi energetici su SmsSpam

🔗 **Answer to RQ₃.** I compromessi tra accuratezza, efficienza energetica e utilizzo delle risorse nei diversi algoritmi di ML variano sensibilmente in base alla fase operativa e al dataset. Il Decision Tree emerge come il modello più efficiente, grazie al suo basso consumo energetico e ai tempi di inferenza minimi, rendendolo ideale per dispositivi con risorse limitate, soprattutto su dataset come German. I modelli ensemble, come l'AdaBoost e il Bagging Classifier, si dimostrano buoni compromessi tra accuratezza ed efficienza. Nonostante i consumi elevati in training, ottimizzano significativamente i consumi in inferenza, mantenendo alte accuratezze e tempi di inferenza contenuti, specialmente su dataset complessi o sbilanciati come SmsSpam. Modelli complessi come l'SVM e il KNN, pur raggiungendo buone accuratezze, sono penalizzati dai loro elevati consumi energetici e dai tempi di inferenza lunghi, rendendoli meno pratici in contesti con vincoli di risorse. La scelta del modello deve quindi bilanciare le necessità di accuratezza e i vincoli operativi, garantendo il miglior compromesso per il contesto applicativo.

CAPITOLO 5

Conclusioni

Questo studio contribuisce a colmare un gap significativo nella letteratura riguardando all'efficienza energetica dei modelli di ML durante le fasi di training e inferenza, un aspetto meno esplorato rispetto ad altri. Lo studio offre una comprensione più approfondita del comportamento dei modelli, evidenziando le differenze nei consumi energetici e nell'utilizzo delle risorse a seconda del tipo di algoritmo e del dataset; è quindi utile nel dimostrare che la scelta di un algoritmo di ML non deve essere motivata solo da considerazioni di accuratezza, ma deve anche tenere conto dei compromessi tra performance e risorse disponibili. Si è quindi analizzata l'efficienza energetica dei principali modelli di ML durante la fase di training, e in particolare modo durante la fase di inferenza dove sono state confrontate anche altre metriche come il tempo di inferenza, l'utilizzo delle risorse e il peso del modello su tre diversi dataset. Per quanto riguarda la fase di training, i risultati ottenuti nel presente studio confermano i risultati di Verdecchia et al. [6], mostrando tendenze simili nella maggior parte dei casi analizzati, rafforzandone la validità. Invece per quanto riguarda la fase di inferenza, i risultati ottenuti hanno evidenziato differenze significative tra i vari algoritmi, in particolare, il Decision Tree si è dimostrato il modello più efficiente in termini di consumo energetico, mentre il Random Forest, pur garantendo una buona accuratezza, ha mostrato un maggiore peso e dei consumi relativamente elevati. Il

KNN ha bilanciato bene l'accuratezza e l'efficienza, anche se ha sofferto di tempi di inferenza relativamente più lunghi. Al contrario, l'AdaBoost e il Bagging Classifier hanno offerto un buon compromesso tra consumi energetici e tempi di inferenza rapidi. I risultati ottenuti suggeriscono che la scelta del modello dipende non solo dall'accuratezza desiderata, ma anche dalle risorse disponibili in particolare in contesti dove le risorse sono limitate. Un ulteriore aspetto importante emerso dalla ricerca è la variabilità dei risultati in base al dataset utilizzato. Per quanto riguarda i possibili sviluppi futuri, un passo importante sarebbe l'estensione dell'analisi a un numero maggiore di dataset provenienti da diverse aree applicative, come l'elaborazione di immagini o dati temporali, includendo anche l'uso di tecniche di pre-processing per ridurre o modificare la dimensione dei dataset, al fine di comprendere meglio come migliorare l'efficienza energetica dei modelli. Questo permetterebbe di ottenere risultati più robusti e generalizzabili, validando ulteriormente le conclusioni raggiunte. Un'altra direzione interessante potrebbe essere l'ottimizzazione dell'efficienza energetica mediante l'uso di tecniche avanzate o l'adozione di hardware specializzato, per ridurre ulteriormente i consumi senza compromettere le performance. Inoltre, sarebbe utile esplorare ulteriori algoritmi di ML, così come analizzare i compromessi tra accuratezza ed efficienza in scenari di utilizzo reale.

Bibliografia

- [1] S. Tufail, H. Riggs, M. Tariq, and A. I. Sarwat, “Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms,” 2023. (Citato alle pagine 1 e 11)
- [2] H. Pallathadka, M. Mustafa, D. T. Sanchez, G. Sekhar Sajja, S. Gour, and M. Naved, “Impact of machine learning on management, healthcare and agriculture,” 2023. (Citato a pagina 1)
- [3] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, “Quantifying the carbon emissions of machine learning,” 2019. (Citato alle pagine 1, 2 e 6)
- [4] V. Ponnusamy, N. V., R. R. S., A. N., and G. P., *Modern Real-World Applications Using Data Analytics and Machine Learning*. Singapore: Springer Nature Singapore, 2024, pp. 215–235. [Online]. Available: https://doi.org/10.1007/978-981-97-0448-4_11 (Citato a pagina 1)
- [5] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green ai,” 2020. (Citato alle pagine 2 e 5)
- [6] R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier, “Data-centric green ai an exploratory empirical study,” in *2022 International Conference on ICT for Sustainability (ICT4S)*, 2022, pp. 35–45. (Citato alle pagine 2, 3, 8, 10, 11, 13, 14, 19, 21 e 32)

-
- [7] Y. Yuan, J. Shi, Z. Zhang, K. Chen, J. Zhang, V. Stoico, and I. Malavolta, “The impact of knowledge distillation on the energy consumption and runtime efficiency of nlp models,” in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, ser. CAIN '24, 2024. (Citato alle pagine 3 e 8)
- [8] R. Verdecchia, J. Sallou, and L. Cruz, “A systematic review of green ai,” *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 4, p. e1507, 2023. (Citato a pagina 5)
- [9] H. Järvenpää, P. Lago, J. Bogner, G. Lewis, H. Muccini, and I. Ozkaya, “A synthesis of green architectural tactics for ml-enabled systems,” in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society*, ser. ICSE-SEIS'24. New York, NY, USA: Association for Computing Machinery, 2024, p. 130–141. (Citato a pagina 6)
- [10] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner, “Exploring the carbon footprint of hugging face’s ml models: A repository mining study,” in *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2023, pp. 1–12. (Citato a pagina 6)
- [11] R. C. Castanyer, S. Martínez-Fernández, and X. Franch, “Which design decisions in ai-enabled mobile applications contribute to greener ai?” vol. 29, no. 1. Springer, 2024, p. 2. (Citato a pagina 7)
- [12] S. O. S. Santos, A. Skiarski, D. García-Núñez, V. Lazzarini, R. De Andrade Moral, E. Galvan, A. L. C. Ottoni, and E. Nepomuceno, “Green machine learning: Analysing the energy efficiency of machine learning models,” in *2024 35th Irish Signals and Systems Conference (ISSC)*, 2024, pp. 1–6. (Citato alle pagine 7 e 8)
- [13] K. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2012. (Citato a pagina 11)
- [14] K. R. Rainio Oona, Teuho Jarmo, “Evaluation metrics and statistical tests for machine learning,” *Scientific Reports*, 2024. (Citato alle pagine 12 e 13)

- [15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2016. (Citato a pagina 13)
- [16] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, "Maat: a novel ensemble approach to addressing fairness and performance bugs for machine learning software," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE. New York, NY, USA: Association for Computing Machinery, 2022. (Citato a pagina 14)