UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Laurea Triennale in Ingegneria Informatica

Progettazione di un sistema di riconoscimento della qualità di produzione di una tostatrice industriale di caffè mediante intelligenza artificiale spiegabile

Relatori:

Ing: Antonio Luca Alfeo

Prof: Mario G.C.A. Cimino

Candidata:

Martina Fabiani

ANNO ACCADEMICO 2023/2024

Abstract

Lo studio di ricerca affrontato si concentra sull'applicazione dell'Artificial Intelligence nel contesto dell'Industria 4.0, un settore che sta rivoluzionando il modo in cui le aziende producono e gestiscono i loro processi produttivi. In particolare l'attenzione è rivolta ai metodi relativi al Machine Learning, una branca dell'Intelligenza Artificiale che consente alle macchine di apprendere dai dati e di migliorare le proprie prestazioni nel tempo, senza la necessità di essere esplicitamente programmate.

Nel campo dell'industria un'efficace gestione della qualità del prodotto è fondamentale ed il Machine Learning consente di identificare rapidamente eventuali difetti o anomalie nella produzione, contribuendo a garantire che i prodotti soddisfino gli standard di qualità richiesti.

Lo studio condotto è basato su uno studio reale di dati provenienti da una tostatrice di caffè, che prende in considerazione i valori di temperatura misurati all'interno delle cinque camere presenti in essa, il volume della materia prima e la sua umidità associandoli ad uno fra tre stati che definiscono la qualità del prodotto.

Durante l'analisi condotta ho affrontato inizialmente il problema della classificazione. Sono stati impiegati due classificatori di Machine Learning, MLPClassifier e RandomForestClassifier, al fine di individuare l'algoritmo più adatto per prevedere con la massima accuratezza possibile la corretta qualità del prodotto.

Successivamente è stato affrontato il problema dell'interpretabilità dei risultati approfondendo temi relativi all'eXplainable Artificial Intelligence (XAI).
L'eXplainable Artificial Intelligence risulta fondamentale quando vogliamo impiegare un sistema basato su Machine Learning nelle industrie, perché i risultati prodotti dal sistema devono poter essere affidabili e facilmente interpretabili da chi non ha nessuna conoscenza nel settore del Machine Learning.
Per comprendere il percorso decisionale del modello utilizzato è stata condotta un'analisi delle feature importance, una misura che valuta l'influenza o il contributo di ciascuna feature all'interno del modello predittivo impiegato. Questa misura permette di comprendere facilmente quali features sono più significative per la previsione del modello e quali hanno un impatto minore o trascurabile. Per il calcolo delle feature importance sono stati utilizzati tre approcci principali: il metodo nativo di RandomForest, l'algoritmo SHAP ed infine il nuovo metodo BoCSoR. L'analisi condotta ha rivelato che il modello attribuisce particolare importanza alle temperature misurate nella camera numero 3 della tostatrice. Questo significa che le variazioni di temperatura registrate in questa specifica camera hanno un'influenza significativa sulla qualità del caffè tostato, secondo il modello utilizzato.

I risultati ottenuti dall'analisi dell'uso degli algoritmi di Machine Learning e dalle feature importance hanno fornito preziose informazioni sulle feature più influenti per la predizione della qualità del prodotto, consentendo una migliore comprensione dei fattori che possono incidere sulla qualità.

Indice

1. Introduzione

L'industria 4.0 si basa sull'uso di tecnologie, come sistemi basati su IoT, Intelligenza Artificiale (AI), robotica avanzata, realtà aumentata (AR) e computazione cloud, per creare un ambiente di produzione altamente interconnesso e digitalizzato.
Gli elementi chiave dell'Industria 4.0 includono la digitalizzazione dei processi produttivi e la raccolta di grandi quantità di dati attraverso sensori e dispositivi, l'analisi predittiva per prevedere guasti e ottimizzare le prestazioni, la personalizzazione di massa attraverso la produzione flessibile e la comunicazione diretta tra macchine, sistemi e persone.
Questo nuovo approccio non solo sta trasformando la produzione, ma anche il modo in cui le aziende concepiscono i loro prodotti e interagiscono con i clienti. L'industria 4.0 promette di portare benefici significativi, come una maggiore efficienza operativa, una riduzione dei costi di produzione, una maggiore agilità nell'adattarsi alle esigenze del mercato in continua evoluzione.

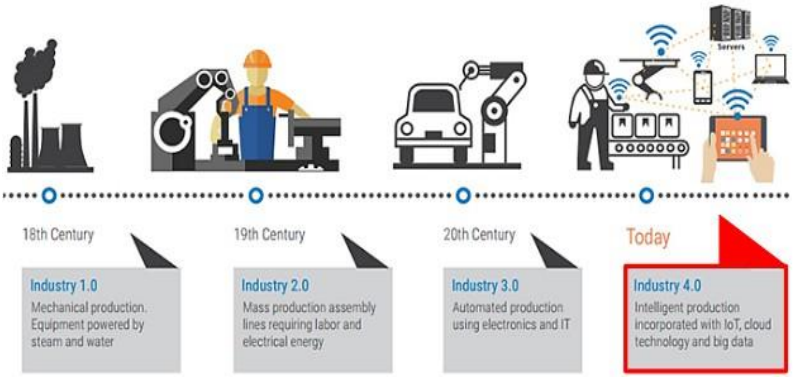


Figura 1: evoluzione dalla prima alla quarta rivoluzione industriale

Un aspetto cruciale di questo processo di trasformazione digitale del settore manifatturiero è il Machine Learning. Esso è una branca dell'Intelligenza artificiale, che consente alle macchine di apprendere dai dati e di migliorare le proprie prestazioni nel tempo senza essere esplicitamente programmate, andando a velocizzare e a rendere più economico un particolare lavoro.

Nel contesto dell'Industria 4.0 il Machine Learning viene utilizzato per svolgere molteplici compiti. Tra i principali utilizzi abbiamo quello della manutenzione predittiva, dell'ottimizzazione dei processi e del controllo di qualità.

1.1 Controllo di qualità

L'utilizzo del Machine Learning per il controllo di qualità offre numerosi vantaggi significativi, migliorando l'efficienza, la precisione e la tempestività delle operazioni di controllo.
Tutto ciò si ottiene riducendo gli errori umani, le distorsioni e la soggettività, automatizzando le attività di ispezione e misurazione, ispezionando e misurando ogni prodotto e fornendo feedback e avvisi immediati.
Questo nuovo approccio rappresenta una svolta fondamentale nell'ottimizzazione dei processi produttivi, consentendo alle aziende di raggiungere livelli di efficienza e qualità senza precedenti, mentre si riducono costi e rischi associati a difetti o prodotti non conformi.

1.2 eXplainable Artificial Intelligence (XAI)

Per poter effettivamente impiegare un sistema basato su Machine Learning nelle industrie, questo modello deve essere in grado di fornire una spiegazione affidabile e facilmente interpretabile. Nelle industrie, spesso mancano esperti specializzati in AI e Machine Learning, dunque l'azienda non può comprendere o fidarsi dei risultati prodotti dai modelli impiegati ed è altamente improbabile che li adotti pienamente, anche se potrebbero offrire vantaggi significativi in termini di efficienza e produttività.
Per evitare questo fenomeno è importante rendere trasparenti e comprensibili i processi decisionali dei modelli di Machine Learning, così da consentire anche ad operatori non addetti al settore di comprendere il funzionamento dei modelli e potersi fidare dei risultati prodotti. Questa analisi è possibile grazie alle nuove tecniche dell'***Intelligenza Artificiale spiegabile (XAI).***

La XAI è un campo di ricerca che mira a rendere i risultati dei sistemi di intelligenza artificiale più comprensibili per gli esseri umani.
Come viene definito in [4] la XAI mira a *"produrre modelli più spiegabili, pur mantenendo un elevato livello di prestazioni di apprendimento (accuratezza della previsione); e consentire agli utenti umani di comprendere, in modo appropriato, fidarsi e gestire in modo efficace la generazione emergente di partner artificialmente intelligenti".*

La spiegabilità e l'interpretabilità sono concetti strettamente correlati nel contesto dell'AI. Un sistema interpretabile è considerato spiegabile se le sue operazioni posso essere comprese dall'uomo.
La XAI si concentra sulla sfida di demistificare i modelli di intelligenza artificiale,

affrontando il problema delle *'black box'.*
Quando ci riferiamo alle *'black box'*, intendiamo i modelli AI complessi che producono previsioni o decisioni senza fornire spiegazioni comprensibili su come sono giunti a tali conclusioni. Questo fenomeno è problematico perché rende difficile per gli utenti comprendere il ragionamento alla base delle previsioni del modello. Senza una spiegazione chiara, gli utenti possono avere difficoltà a fidarsi delle decisioni del modello e può essere difficile identificare e affrontare eventuali errori nei dati o nelle previsioni.

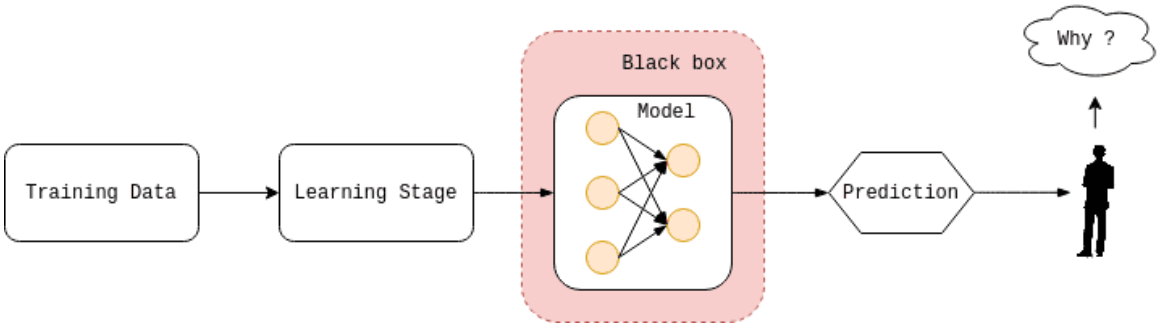


Figura 2: Black Box model

Affrontare la sfida delle "black box" è cruciale per garantire la trasparenza, l'equità e l'etica nell'uso dei modelli di intelligenza artificiale. La XAI si propone di sviluppare metodi e tecniche per rendere i modelli di AI più interpretabili e spiegabili, consentendo agli utenti di comprendere meglio le decisioni del modello e di prendere decisioni informate e consapevoli. Questo è essenziale in una vasta gamma di settori e contesti, come ad esempio:

- *La sanità*: in ambito medico, la trasparenza e l'interpretabilità dei modelli sono fondamentali per aiutare i medici a comprendere le decisioni dei sistemi di supporto decisionale, come la diagnosi assistita da computer o la personalizzazione dei trattamenti. Solo interpretando il modello i medici posso fidarsi dei risultati e trattare il paziente secondo le istruzione fornite.

- *I trasporti:* nell'industria automobilistica, la XAI può essere applicata per spiegare il funzionamento dei modelli di guida autonoma, inclusi quelli legati alla percezione dell'ambiente, alla pianificazione del percorso e alla guida autonoma. Questo è essenziale per garantire la sicurezza dei veicoli autonomi e la fiducia degli utenti.

- *L'industria:* in questo settore la XAI ha diverse applicazioni, contribuendo a migliorare la trasparenza, la fiducia e l'efficienza dei sistemi basati su AI.
    - *Produzione e qualità:* nell'industria manufatturiera la XAI può

essere impiegata per identificare le cause di difetti o anomalie nei processi di produzione. Ciò consente di migliorare la qualità dei prodotti e di ottimizzare i processi.

- *Manutenzione predittiva:* in molti settori dell'industria, come ad esempio il settore dell'energia o dell'aviazione, la XAI può essere impiegata per identificare e prevenire guasti imminenti nei macchinari così da ridurre i tempi di fermo e ottimizzare le operazioni di manutenzione.
- *Automazione industriale:* in questo contesto la XAI può essere utilizzata per consentire agli operatori umani di comprendere al meglio il funzionamento dei sistemi e intervenire in caso di necessità.

La mia tesi tratta diverse metodologie collegate al controllo di qualità attraverso l'utilizzo del Machine Learning per garantire una corretta classificazione e la futura predizione della qualità dei prodotti provenienti dai macchinari industriali.
In particolar modo, in questo studio, è stata posta l'attenzione sulla definizione delle feature importance, che consistono in una misurazione dell'importanza di ciascuna feature per la predizione del risultato finale da parte del modello.
Queste sono state calcolate utilizzando tre approcci diversi. Inizialmente è stato utilizzato il metodo nativo di RandomForest, che valuta quanto una feature riduce l'incertezza nelle predizioni del modello. Successivamente siamo passati all'utilizzo del metodo SHAP, il quale calcola i valori di Shapley e attribuisce a ciascuna feature un valore di importanza. Infine è stato impiegato un metodo chiamato "Boundary Crossing Solo Ratio" (BoCSoR), che valuta l'importanza di ogni singola feature aggregando singole "Counterfactual Explaination".

Tutte le analisi e valutazione condotte mirano ad assicurare che un sistema di classificazione, fondato su Machine Learning, sia in grado di fornire previsioni precise sulla qualità dei prodotti, generando risultati affidabili e comprensibili. Lo scopo principale è semplificare l'adozione di questi modelli nell'industria 4.0.

In un contesto industriale in continua evoluzione, l'intelligenza artificiale spiegabile (XAI) sta guadagnando sempre più importanza. Ciò è particolarmente rilevante nel settore manifatturiero, dove ci si aspetta che, nel prossimo futuro, l'AI possa fornire risultati affidabili e accurati senza possibilità di contestazione. Questo evidenzia la necessità di garantire che i processi decisionali dei modelli siano trasparenti e facilmente interpretabili, al fine di accrescere la fiducia degli operatori e degli stakeholder nell'utilizzare queste tecnologie avanzate.

2. Related Works

Questo capitolo offre una panoramica sulle ricerche correlate al campo di studio, offrendo il contesto a partire dal quale l'analisi si sviluppa. In particolare, esploreremo le problematiche legate alla presenza di forti correlazioni tra le feature dei dataset. Inoltre, evidenzieremo l'importanza dell'utilizzo dei controfattuali nell'ambito dell'Intelligenza artificiale spiegabile (XAI).

2.1 Spiegabilità del modello

Nel contesto sempre più complesso dell'intelligenza artificiale, comprendere e interpretare i modelli di apprendimento automatico è diventato cruciale per garantire la trasparenza, la fiducia e l'efficacia dei sistemi basati sull'AI [1]. Tuttavia, interpretare i modelli non è un compito facile, soprattutto considerando la varietà di algoritmi e tecniche disponibili. In questo contesto, emergono due approcci distinti per l'interpretazione dei modelli [5]:

1. *Tecniche specifiche del modello:* queste tecniche sono progettate per interpretare modelli con caratteristiche e capacità specifiche. Sono sviluppate esclusivamente per una singola classe di algoritmi. Questo significa che ciascuna tecnica di interpretazione è ottimizzata per un particolare tipo di modello, fornendo una comprensione dettagliata del funzionamento interno di quel modello specifico.
2. *Tecniche agnostiche al modello:* queste tecniche di interpretazione, al contrario, possono essere applicate a qualsiasi modello di apprendimento automatico, indipendentemente dal tipo di algoritmo utilizzato. Queste tecniche sono progettate per essere flessibili e generalizzabili, consentendo agli utenti di ottenere spiegazioni comprensibili e affidabili su come i modelli prendono decisioni, indipendentemente dalla loro complessità o struttura.

In particolare in questo studio sono state analizzate tecniche agnostiche al modello, come ad esempio SHAP o BoCSoR, che andremo ad approfondire.

2.1.1 Limitazioni del modello SHAP

Il modello SHapley Additive exPlanations (SHAP) è una tecnica di spiegabilità molto potente e ampiamente utilizzata nel campo dell'Intelligenza Artificiale Spiegabile. Questa tecnica si basa sui concetti della teoria dei giochi e attribuisce un valore di "*Shapley*" a ciascuna feature in una previsione, valutando il contributo marginale

di quella feature al risultato finale della previsione.
I valori di Shapley rappresentano il contributo relativo di ciascuna feature alla differenza tra la previsione effettiva del modello e la previsione media su tutte le possibili combinazioni di feature. Questo processo coinvolge la valutazione di tutte le combinazioni possibili di feature, calcolando il contributo di ciascuna feature a ogni previsione, tenendo conto di tutte le possibili permutazioni di feature.
Tuttavia, nonostante la sua efficacia, questo algoritmo presenta alcune limitazioni significative. Innanzitutto un elevato costo computazionale: la complessità temporale di SHAP aumenta in modo esponenziale con il numero di caratteristiche e in modo lineare con il numero di campione nel dataset. Questo può rendere l'applicazione di SHAP computazionalmente onerosa, specialmente con dataset di grandi dimensioni.
Inoltre, un'altra limitazione critica da considerare riguardo a questo modello è la presenza di un elevato grado di dipendenza tra alcune o tutte le features del dataset. Questo può portare a risultati dell'analisi completamente distorti e interpretazioni inaccurate delle spiegazioni fornite da SHAP. In aggiunta questo fenomeno può causare problemi di interpretazione e rendere le previsioni del modello meno affidabili [2].

2.1.2 L'importanza dei controfattuali

Un controfattuale, nell'ambito dell'Intelligenza Artificiale, è una situazione o uno scenario immaginario che rappresenta una variazione rispetto alla realtà osservata. Esso indica ciò che sarebbe accaduto se le condizioni del contesto fossero state diverse da quelle effettivamente osservate. Nell'ambito dell'XAI, i controfattuali sono strumenti potenti per comprendere e interpretare le decisioni dei modelli di Machine Learning. Consentono agli utenti di esplorare come le previsioni del modello sarebbero cambiate se le condizioni del contesto fossero state diverse, aiutando così a identificare le feature cruciali per le previsioni del modello e a individuare eventuali anomalie. [3]

2.2 Soluzioni proposte

Nell'analisi condotta sono stati confrontati diversi algoritmi di Machine Learning al fine di individuare il modello che fornisce le migliori prestazioni e i risultati più accurati per il compito di classificazione.

È stata esaminata la matrice di correlazione del dataset in esame ed è stata notata la presenza di forti correlazioni tra gruppi di features.
La soluzione proposta, con l'obbiettivo di avere risultati adeguati e accurati

utilizzando l'algoritmo SHAP, è stata selezionare solo due features per ciascun gruppo, al fine di ridurre il dataset eliminando le informazioni ridondati.

Infine, in questa tesi, è stato analizzato il percorso decisionale del modello attraverso l'algoritmo BoCSoR, che punta a ridurre il costo computazionale e si basa sull'aggregazione di diversi risultati ottenuti attraverso la *'counterfactual explanations'*.

3. Design e Implementazione

3.1 Design

In questa sezione viene fornita un'approfondita illustrazione dell'approccio utilizzato durante le varie fasi di analisi, facendo chiarezza in particolar modo sulla tipologia di modelli considerati e i metodi di XAI utilizzati.

3.1.1 Classificazione

La classificazione nel Machine Learning è una tecnica fondamentale che consiste nel costruire modelli in grado di assegnare un'istanza di dati a una o più categorie predefinite. In altre parole, il processo di classificazione consiste nell'attribuire una classe predefinita (nel nostro caso le classi sono LOW, MEDIUM, HIGH in riferimento alla qualità del prodotto) a un input in base alle caratteristiche intrinseche di quell'input. L'obiettivo è quello di insegnare al modello a riconoscere schemi e relazioni nei dati in modo che possa effettuare predizioni accurate su nuovi dati che non ha mai visto in precedenza.
In particolare, sono stati analizzati i risultati dell'algoritmo di classificazione **RandomForestClassifier**.

- *RandomForestClassifier*, è un algoritmo che utilizza sia il bagging che la causalità delle caratteristiche per creare una 'foresta' non correlata di strutture ad albero decisionali. La casualità delle caratteristiche genera un sottoinsieme di caratteristiche, che garantisce una bassa correlazione tra le strutture ad albero decisionali. Questa caratteristica costituisce una differenza fondamentale rispetto agli alberi decisionali singoli, in cui vengono considerate tutte le possibili suddivisioni delle feature. L'algoritmo RandomForest è quindi composto da una serie di alberi decisionali, ognuno addestrato su un sottoinsieme diverso di caratteristiche, e le previsioni finali sono ottenute dall'aggregazione delle previsioni di tutti gli alberi nella foresta. In particolare, nel caso di classificazione, la variabile categoriale più frequente nelle previsioni dei singoli alberi, produrrà la classe prevista. Il vantaggio principale di questo approccio è che combina la potenza della creazione di molteplici alberi decisionali con la casualità introdotta dalla selezione casuale delle feature e dei campioni, riducendo così il rischio di overfitting e migliorando la generalizzazione del modello. Inoltre, è in grado di gestire grandi quantità di dati e di identificare automaticamente le feature più importanti per la

classificazione. Tuttavia, le prestazioni possono essere influenzate dalla qualità e dalla quantità dei dati di addestramento, nonché dalla scelta dei parametri, come il numero di alberi e la dimensione del sottocampione di feature.

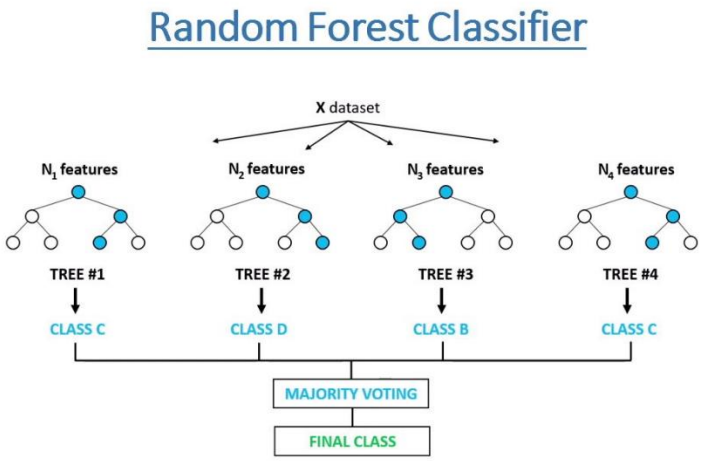


Figura 3: funzionamento algoritmo RandomForestClassifier

3.1.2 Tecniche di pre-processing

Le tecniche di pre-processing sono utilizzate per preparare i dati prima di utilizzarli nell'addestramento di modelli. Queste tecniche sono fondamentali per garantire che i dati siano nella forma più adatta, consentendo di migliorare le prestazioni complessive del modello ottenuto.
Nello specifico, durante l'analisi, sono stante confrontate due tecniche di scaling, che consentono di trasformare le caratteristiche dei dati in un range specifico, al fine di valutare similitudini e differenze.

- *MinMaxScaler*: scaler utile per ridimensionare i dati in modo che siano compresi tra un intervallo specifico, di solito 0 e 1. Questo può essere utile per algoritmi che richiedono che i dati siano in un intervallo specifico. Inoltre è appropriato utilizzarlo quando le variabili hanno una distribuzione non gaussiana ed è necessario scalare i valori in un intervallo specifico senza influenzare la forma della distribuzione.
  La formula utilizzata per il calcolo del valore normalizzato *x'* per ogni valore *x* è la seguente:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- *StandardScaler*: scaler utilizzato per standardizzare i dati in modo che abbiano una media pari a zero e una deviazione standard unitaria, garantendo che i dati

abbiano una distribuzione normale. È utile il suo utilizzo quando le variabili seguono una distribuzione gaussiana oppure quando si utilizzano algoritmi che richiedono che le feature siano distribuite normalmente.
La formula utilizzata da StandardScaler per il calcolo del valore standardizzato *z* per ogni valore *x* è la seguente:

$$z = \frac{x - mean(x)}{std(x)}$$

dove *mean(x)* è la media della feature *x* nell'intero set di dati e *std(x)* è la deviazione standard.

3.1.3 Feature Importance

L'eXplainable Artificial Intelligence (XAI) si dedica a sviluppare algoritmi e strumenti che consentono di comprendere il ragionamento alla base delle decisioni prese da modelli di intelligenza artificiale. L'obiettivo è spiegare il "perché" di un determinato output generato da un modello, consentendo agli utenti di prendere decisioni consapevoli e di interpretare i risultati in modo significativo.

Una delle strategie impiegate per spiegare i risultati dei modelli è quella delle feature importance. Questa misura valuta l'influenza di ciascuna caratteristica del dataset sull'output predetto dal modello.
Nella nostra analisi, sono state utilizzate tre metriche diverse per valutare l'importanza delle caratteristiche: metodo nativo di RandomForestClassifier, SHAP values e BoCSoR.

3.1.3.1 Metodo nativo di RandomForestClassifier

Il metodo nativo di RandomForestClassifier per il calcolo dell'importanza delle features si basa sulla diminuzione dell'indice di impurità di Gini causata da ciascuna variabile quando viene utilizzata per suddividere i nodi dell'albero. L'impurità di Gini misura la disuguaglianza della distribuzione delle classi in un nodo dell'albero. Quando una feature viene utilizzata per suddividere i nodi dell'albero durante la costruzione, l'indice di impurità di Gini diminuisce. Questa riduzione rappresenta la riduzione dell'incertezza nelle predizioni del modello.
In sostanza, il processo valuta quanto una feature riduce l'incertezza nelle predizioni del modello, misurata dall'indice di impurità di Gini. Le feature che causano una maggiore riduzione dell'indice sono considerate più importanti per il modello, poiché forniscono una maggiore separazione delle classi e quindi una

migliore capacità predittiva.

3.1.3.2 SHAP values

Il concetto chiave di SHAP è il valore di Shapley, esso rappresenta il contributo marginale di una feature in una previsione, considerando tutte le possibili combinazioni di variabili e attribuendo a ciascuna variabile una parte della differenza tra le previsioni con e senza la variabile stessa. In altre parole, misura quanto ciascuna variabile contribuisce alla differenza tra la previsione effettiva e la previsione media. Gli SHAP values forniscono una comprensione dettagliata di come ciascuna feature influisce sulle previsioni del modello, consentendo di valutare quanto positivamente o negativamente ciascuna caratteristica influenzi la variabile target.

3.1.3.3 BoCSoR

Gli algoritmi visti fin ora per valutare le features importance presentano alcune limitazioni, tra cui un elevato costo computazionale. Ad esempio, la complessità temporale di SHAP aumenta in modo esponenziale con il numero di caratteristiche e in modo lineare con il numero di campioni nel dataset. Questo non è un problema unico a SHAP, ma si applica a molte misure di feature importance, rendendo il processo computazionalmente oneroso.

Per ridurre i costi computazionali associati alla valutazione delle features importance, si introducono le *"counterfactual explanations*". Queste si basano sul concetto di "***controfattualità***", che riguarda ciò che sarebbe potuto accadere se i dati fossero stati diversi o se una particolare variabile fosse stata modificata. In termini pratici, consideriamo un'istanza di dati *'i'* con la sua classe predetta. Un *controfattuale* è un'istanza *'c'* simile a *'i',* ma classificata in modo diverso. Trovare questa istanza *'c'* simile implica comprendere la minima modifica necessaria per alterare l'esito della classificazione. Questo approccio offre una soluzione più efficiente dal punto di vista computazionale rispetto ad alcuni metodi tradizionali di valutazione dell'importanza delle caratteristiche.

Boundary Crossing Solo Ratio **(BoCSoR)** è un metodo di calcolo delle feature importance che si basa sull'aggregazione di diversi risultati ottenuti attraverso *la "counterfactual explanations"*, in modo da ottenere un quadro informativo globale dei risultati. Il concetto fondamentale di BoCSoR consiste nel valutare l'importanza di una caratteristica osservando quanto spesso i campioni vicino al *confine decisionale* del modello producono una classificazione diversa quando il valore di quella caratteristica viene sostituito con quello del *campione controfattuale* corrispondente. In sostanza, si cerca di capire quanto la variazione di una specifica

caratteristica influisca sul risultato della classificazione, concentrandosi sulla regione in prossimità del confine decisionale del modello.

Per determinare quale campione, all'interno della stessa classe, è più vicino al confine decisionale, si utilizza la distanza Euclidea. Una volta individuato il campione (campione *'s'* nella figura 2), per trovare il controfattuale corrispondente, si considerano *i K-Nearest-Neighbours* della classe controfattuale. Successivamente, vengono generati ***'s'*** punti intermedi tra ogni possibile controfattuale e il campione originale. Attraverso una progressiva esplorazione dei segmenti che collegano il campione con i ***'K'*** "vicini", si individua il controfattuale corrispondente che ha il punto intermedio con la distanza minima e appartiene a una classe diversa. In sostanza, si cerca il punto di transizione più vicino al campione originale che cambia la classe.

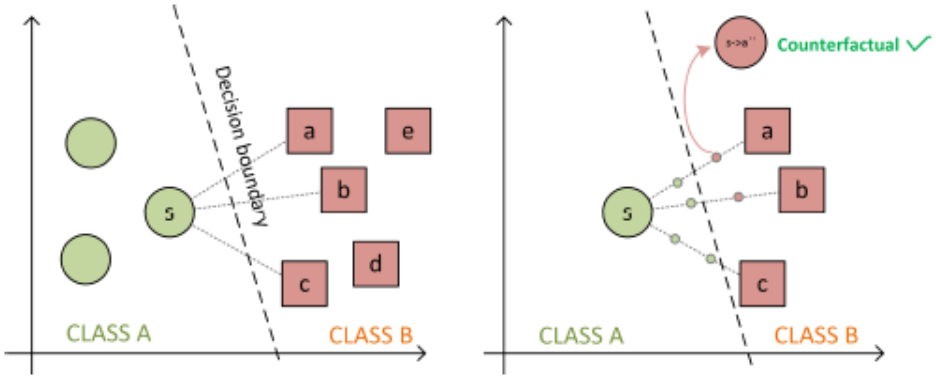


Figura 4: approccio utilizzato per trovare il contraffattuale corrispondente

Ottenuto il contrafattuale corrispondente (contraffattuale *'a'* nell'esempio in figura 2), questo metodo, va a sostituire uno alla volta i valori delle istanze con i contrafattuali associati. Se questa sostituzione corrisponde all'attraversamento del confine decisionale, tale feature è considerata rilevante per il modello.

L'algoritmo appena illustrato ha una complessità temporale che cresce linearmente con il numero di caratteristiche e quadraticamente con il numero di campioni. Questa complessità risulta inferiore rispetto a quella di SHAP, descritta inizialmente. Tuttavia, è importante notare che questo metodo non può garantire con certezza la minima distanza tra l'istanza e il controfattuale ottenuto, né può garantire la migliore approssimazione del confine decisionale.

3.2 Implementazione

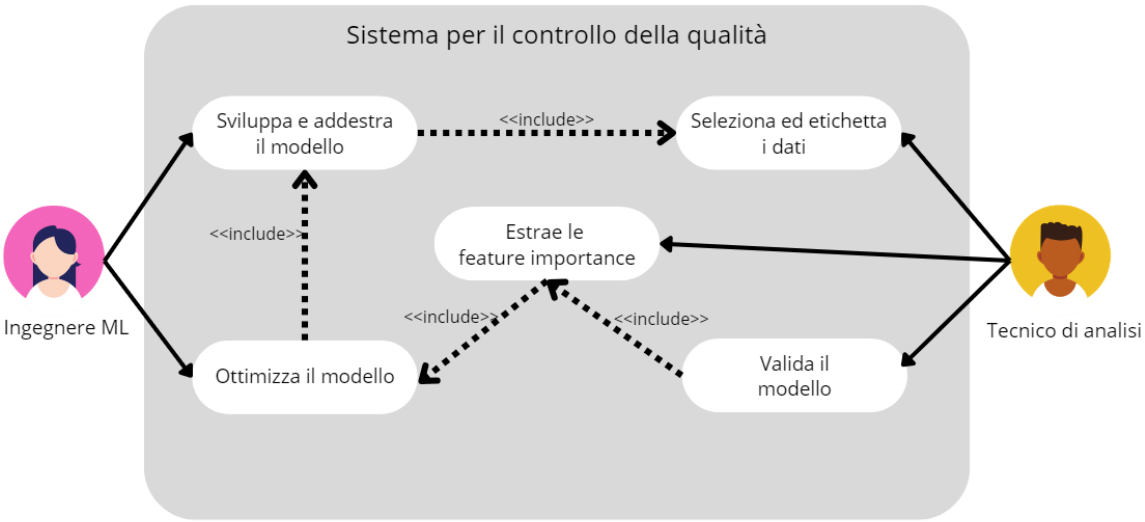


Figura 5: Use case

Un possibile use-case del software realizzato è quello mostrato in figura 3. In questo scenario il software per la valutazione della qualità del prodotto si basa sui dati raccolti dai sensori di temperatura posti nelle cinque camere della macchina per la torrefazione del caffè, sul volume della materia prima e sulla sua umidità.
In questo contesto si hanno due attori principali:

- **Ingegnere ML**: un ingegnere specializzato in Machine Learning, responsabile dello sviluppo e dell'addestramento del sistema. In particolare si occupa di:
  - *Sviluppare e addestrare il modello*: l'ingegnere sviluppa sistemi di Machine Learning, basati su classificatori come MLPClassifier o RandomForestClassifier, per riconoscere la qualità del prodotto. Una volta sviluppato il modello, si occupa di addestrarlo utilizzando algoritmi di Machine Learning per consentire al sistema di prevedere la qualità.
  - *Ottimizzare il modello*: dopo che il modello ha prodotto dei risultati, questi verranno validati dal tecnico di analisi e i feedback ricevuti serviranno all'ingegnere ML per migliorare l'accuratezza e l'affidabilità del modello

- **Tecnico di analisi:** il tecnico di analisi è l'attore principale, colui che usufruisce del prodotto software finale. È una persona in grado di valutare i risultati del nostro modello, ma senza nessun tipo di background riguardante

l'AI. Nel sistema proposto si occupa di:

- *Selezionare ed etichettare i dati:* il tecnico di analisi ha una conoscenza approfondita del prodotto in esame e del processo di produzione associato. Si occupa di analizzare il prodotto in laboratorio ed etichettare i dati in base ai criteri di qualità stabiliti.

- *Estrarre le feature importance:* il tecnico di analisi analizza le feature importance prodotte dal sistema per identificare quali caratteristiche influenzano maggiormente la qualità del prodotto.

- *Validare il modello:* in base ai risultati ottenuti analizzando le feature importance, il tecnico di analisi stabilisce se il modello è in grado di compiere scelte corrette e affidabili. I feedback risultanti verranno utilizzati dall'ingegnere ML per l'ottimizzazione del sistema.

È importante evidenziare come l'utilizzo della XAI, permetta al tecnico di analisi di utilizzare un modello di Intelligenza Artificiale, pur non avendo nessuna conoscenza nel settore.

4. Case Study

4.1 Il Dataset

Il dataset utilizzato è composto da dati reali, provenienti da una macchina per la tostatura del caffè.

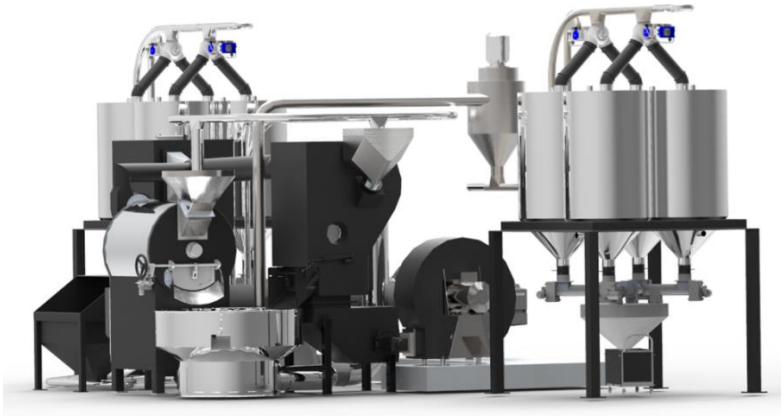


Figura 6: esempio di tostatrice

La ***tostatrice*** è un aggregato composto da 5 camere di uguali dimensioni, ciascuna dotata di 3 sensori di temperatura, situati in punti diversi della camera per garantire la corretta temperatura in ogni parte. Inoltre è presente un sensore per misurare il volume della materia prima e un ulteriore sensore per misurare la sua umidità. Tutte queste caratteristiche, temperatura, volume e umidità, vengono misurate ogni minuto.

Il processo di torrefazione ha durata di un'ora, durante il quale sono state fatte ben 1020 misurazioni (17 al minuto). Al termine di ogni processo sono stati prelevati dei campioni dal lotto tostato per essere analizzati in laboratorio al fine di valutare la qualità del prodotto risultante. Grazie a questa analisi è stato possibile attribuire un livello di qualità, LOW, MEDIUM o HIGH, ai dati misurati dai sensori in quel determinato processo.

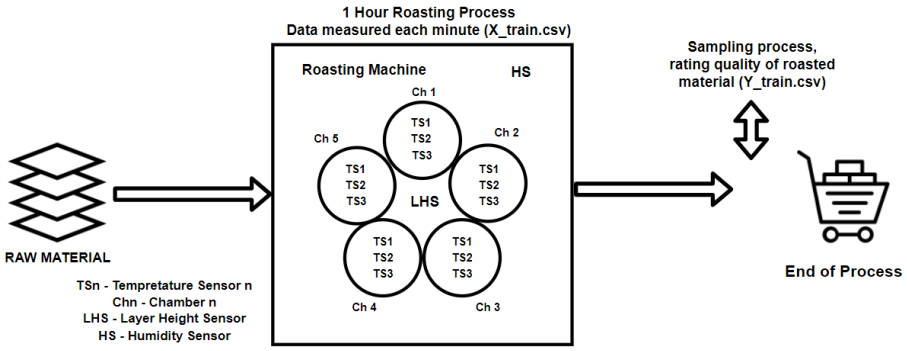


Figura 7: disegno semplificato della tostatrice raffigurante i sensori presenti

4.2 Le Features

Di seguito vengono presentate le features del dataset analizzato.
Le prime due colonne indicano:

- **slot:** il lotto a cui appartengono i campioni analizzati
- **interval:** l'intervallo in cui sono state fatte le misurazioni

Dopodiché abbiamo 56 colonne che rappresentano le misure statistiche dei dati raccolti dai 17 sensori presenti nella tostatrice. I dati delle misurazioni dei tre sensori presenti in ciascuna delle 5 camere sono stati raggruppati, ottenendo così i seguenti gruppi:

- **T1_** → indicano le statistiche delle temperature misurate nella camera 1
- **T2_** → indicano le statistiche delle temperature misurate nella camera 2
- **T3_** → indicano le statistiche delle temperature misurate nella camera 3
- **T4_** → indicano le statistiche delle temperature misurate nella camera 4
- **T5_** → indicano le statistiche delle temperature misurate nella camera 5
- **H_** → indicano le statistiche dell'umidità misurata nella materia prima
- **AH_** → indicano le statistiche del volume della materia prima

Per ogni gruppo sopra indicato, sono state calcolate le seguenti misure:

- la media
- la mediana
- l'asimmetria delle distribuzioni (skewness)
- la deviazione standard
- i valori del 25° percentile
- i valori del 50° percentile
- i valori del 75° percentile
- i valori del 90° percentile

Queste misure statistiche sono utili per comprendere la distribuzione dei dati e per identificare i valori che sono al di sotto o al di sopra della maggior parte delle osservazioni.

Infine abbiamo 2 colonne che rappresentano la qualità risultante dall'analisi del prodotto:

- **quality:** rappresenta un punteggio associato alla qualità del campione
- **quality_cat:** rappresenta la categoria di qualità associata al campione analizzato (LOW, MEDIUM, HIGH)

4.3 Librerie utilizzate

Il codice è stato interamente implementato in Python, sfruttando diverse librerie messe a disposizione.

- ***Scikit-learn**:* è una libreria fondamentale nel campo del Machine Learning. Grazie ai suoi moduli offre un facile accesso a tutte le funzioni e classi necessarie per il pre-processing dei dati, l'implementazione e l'addestramento di algoritmi di classificazione e l'analisi delle prestazione dei modelli generati. Questa libreria mi ha permesso di:
    - Effettuare la divisione dei dati tra set di addestramento e set di test utilizzando la funzione *train_test_split*
    - Utilizzare i classificatori *MLPClassifier* e *RandomForestClassifier*
    - Addestrare i classificatori tramite la funzione *fit*
    - Calcolare l'accuratezza del modello utilizzato, ovvero la percentuale delle previsioni corrette rispetto al totale, tramite la funzione *accuracy_score*
    - Calcolare l'importanza delle caratteristiche all'interno di un modello, grazie all'utilizzo della funzione *feature_importances.*
- ***Pandas:*** libreria essenziale che offre strutture dati flessibili e potenti, come DataFrame, che consentono la gestione efficiente di grandi quantità di dati. Inoltre fornisce funzioni utili per la lettura e la scrittura di file in formato tabulare, come *read_csv* e *to_csv*, semplificando notevolmente il processo di importazione ed esportazione dei dati da e verso diverse fonti.
- ***Matplotlib:*** libreria molto utile per la creazione dei grafici di vario tipo, consentendoci di visualizzare al meglio i risultati ottenuti dalle analisi svolte.
- ***Numpy:*** libreria per lavorare con gli array multidimensionali e matrici. È stata molto importante nell'elaborazione dei dati di input, inclusi quelli utilizzati per le etichette di addestramento e di test.
- ***Seaborn:*** libreria, a sua volta basata su matplotlib, utilizzata per la visualizzazione di alcuni grafici più complessi in modo semplice ed efficiente.
- ***Shap:*** libreria utilizzata per il calcolo degli SHAP values.
- ***BoundaryCrossingSoloRatio:*** libreria utilizzata per lavorare con l'algoritmo BoCSoR.

5. Risultati sperimentali

5.1 Introduzione all'indagine

L'obiettivo principale di questa analisi è fornire una visione d'insieme completa dei risultati ottenuti. Partiremo da una panoramica sull'accuratezza degli algoritmi di classificazione. Successivamente esamineremo in dettaglio le feature importance ottenute attraverso l'algoritmo RandomForest. Passeremo a fare un confronto tra i risultati delle feature importance ottenuti con diversi metodi che utilizzano approcci differenti, come SHAP e BoCSoR.
Infine concluderemo analizzando e confrontando i risultati ottenuti con l'algoritmo BoCSoR al cambiare dei parametri, al fine di trovare la parametrizzazione ottima.

5.2 Risultati della classificazione

Prima di passare all'utilizzo degli algoritmi di classificazione è stata fatta una fase di pre-processing dei dati, durante la quale i dati vengono manipolati prima di essere utilizzati dal classificatore. Nel caso studiato non erano presenti dati mancanti, dunque questa fase si è limitata alla riduzione della dimensionalità del dataset e alla normalizzazione dei dati.
Il numero di features è stato ridotto eliminando le colonne *slot, interval e quality.*
L'analisi del modello è stata fatta 3 volte: inizialmente i dati non sono stati standardizzati con nessun metodo, successivamente è stato applicato lo *Standard Scaler* e infine si è passati ad utilizzare il *MinMax Scaler*.

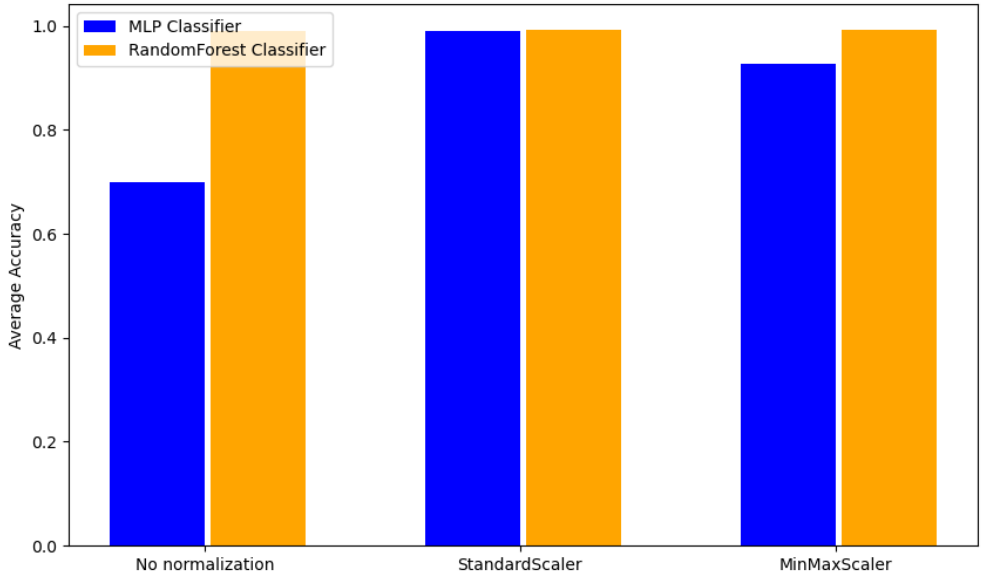


Figura 8: confronto accuratezza media dei due classificatori utilizzati

Infine per avere una stima più precisa dell'accuratezza dei modelli sono stati fatti 10 trial ed è stata calcolata l'accuratezza media.
Dal grafico mostrato in Figura 8, notiamo che in questo specifico caso applicando il RandomForest Classifier, la standardizzazione in fase di pre-processing non porta significativi miglioramenti ai risultati. Difatti abbiamo un'accuratezza media superiore al 95% in tutti e tre i casi analizzati.
Possiamo concludere che questo algoritmo è meno sensibile alla standardizzazione dei dati, e analizzando come lavora possiamo dedurre i motivi per il quale si ha questa caratteristica:

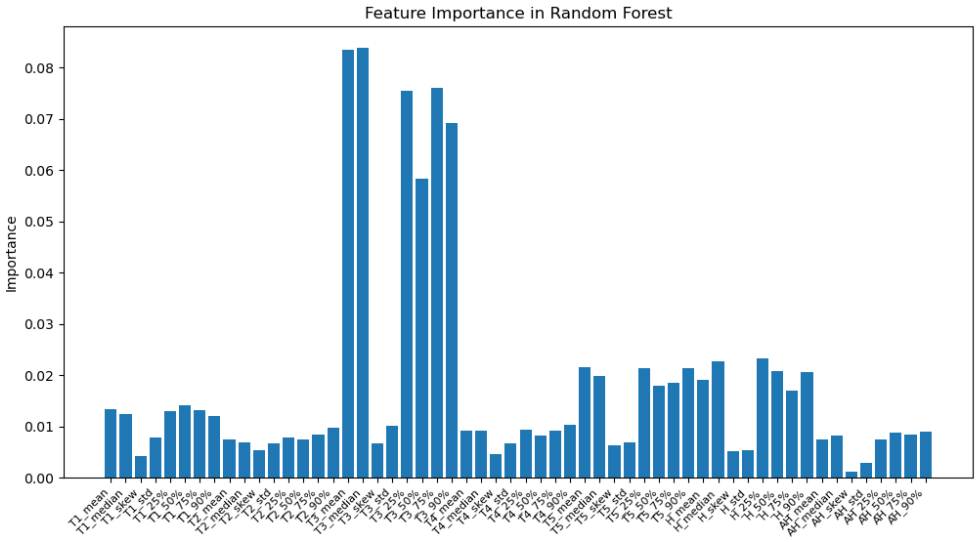
- Esso si basa sulla divisione ricorsiva dei nodi dell'albero decisionale. Queste divisioni vengono fatte su singole caratteristiche e non dipendono dall'unità di misura o dalla scala delle caratteristiche.
- Esso aggrega i risultati di diversi alberi decisionali. Ogni albero può utilizzare caratteristiche a scale diverse, e la combinazione di questi alberi può mitigare gli effetti di eventuali differenze di scala.

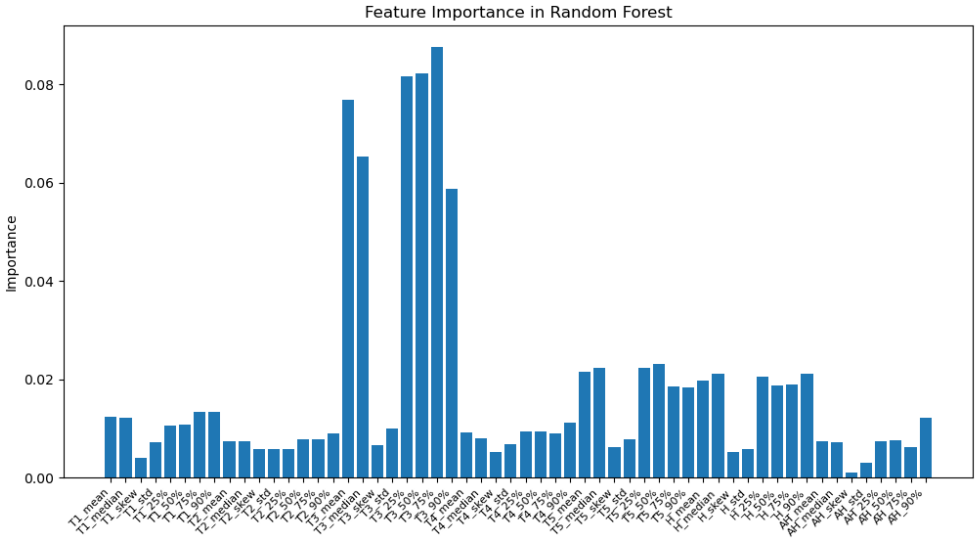
In base a quanto emerso da questo primo studio, le future analisi utilizzeranno come modello il classificatore RandomForest, data la sua elevata accuratezza nel predire la qualità.

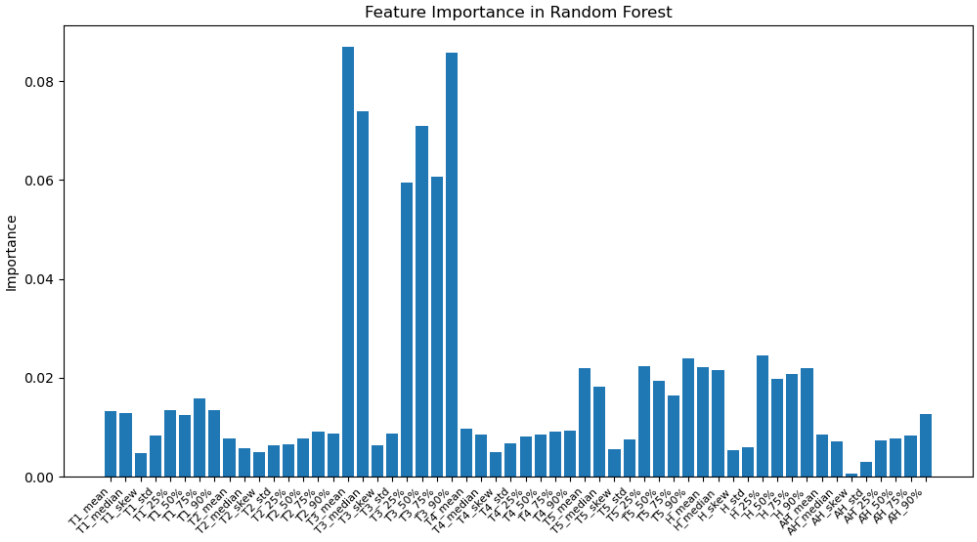
5.3 Analisi delle Feature Importance

5.3.1 Metodo nativo di RandomForestClassifier

Dopo aver accertato l'accuratezza del classificatore RandomForest, siamo passati a calcolare le misure di feature importance che questa tecnica ci offre.
Per poter comprendere al meglio le feature più importanti ho illustrato i dati emersi servendomi di grafici a barre. Inoltre sono stati eseguiti 5 trial per avere una stima più precisa e capire se questo metodo fosse stabile oppure no. I risultati ottenuti sono rappresentati nelle figure seguenti.


Feature Importance in Random Forest
Importance
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08


Feature Importance in Random Forest
Importance
0.00
0.02
0.04
0.06
0.08


Feature Importance in Random Forest
Importance
0.00
0.02
0.04
0.06
0.08

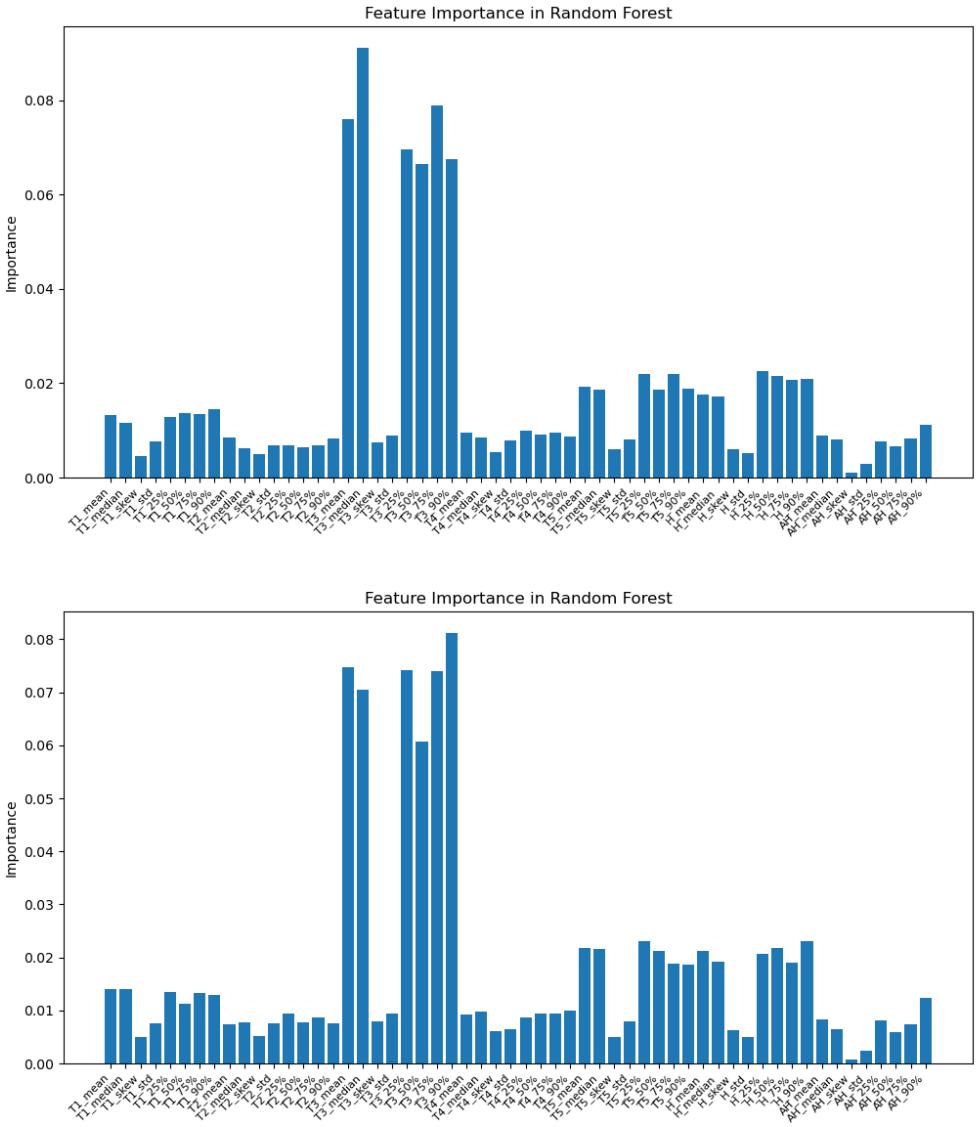


Figura 9: feature importance calcolate tramite il metodo nativo di RandomForest

Analizzando questi 5 grafici ottenuti con il metodo nativo di RandomForest per il calcolo della feature importance, notiamo subito come in tutti e 5 i test le features con maggiore importanza per il modello sono quelle riguardanti la Temperatura rilevati nella camera numero 3 della tostatrice. Inoltre, il fatto che questo approccio abbia prodotto risultati consistenti e stabili tra i diversi test suggerisce che l'importanza delle feature calcolata dal modello è affidabile e coerente.

5.3.2 SHAP values

La seconda analisi sulle feature importance è stata condotta utilizzando il metodo SHAP. Ancora una volta per avere risultati migliori sono stati eseguiti 5 trial e i valori ottenuti sono riportati nei grafici a seguire.

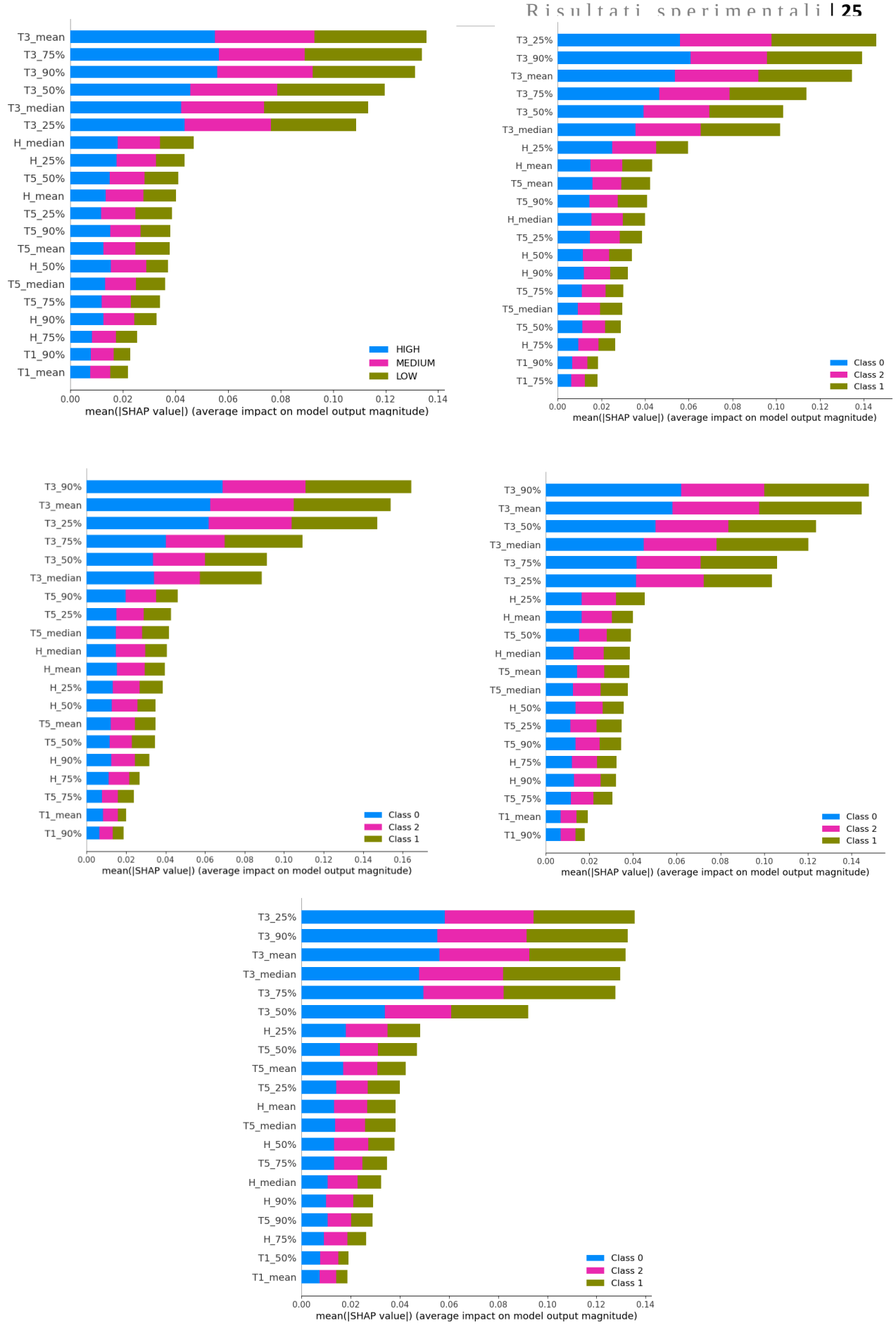


Figura 10: importanza delle features calcolata come valore assoluto medio dei valori di Shapley

Da questi grafici è emerso subito come la temperatura rilevata nella camera 3 del macchinario è un fattore determinante per la qualità del prodotto.
In particolare possiamo anche osservare come tutte e tre le classi utilizzano allo stesso modo le stesse feature, dato che i colori in ogni barra del grafico sono circa di egual misura.
Inoltre, osservando i 5 grafici ottenuti abbiamo notato che i risultati di features importance variano in modo trascurabile, garantendo stabilità.

Infine, attraverso l'analisi dei valori Shapley, ho esaminato come ogni feature impatta positivamente o negativamente sulle predizioni di ciascuna classe, fornendo una visione più dettagliata delle contribuzioni di ogni features.

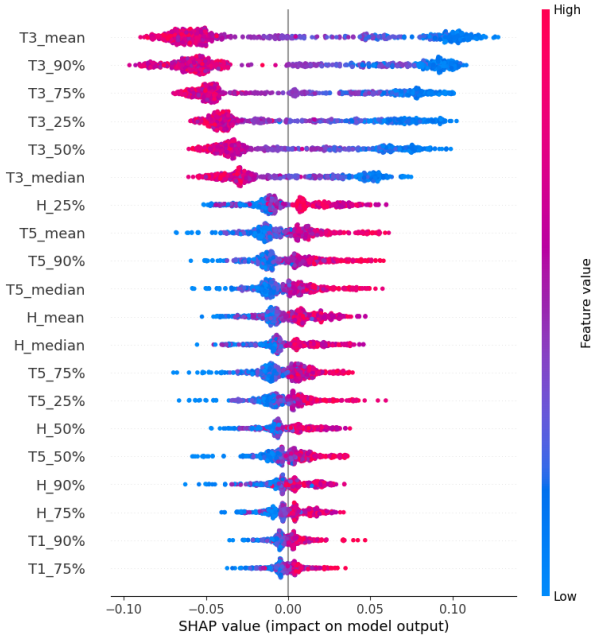


Figura 11: Shap Value classe HIGH

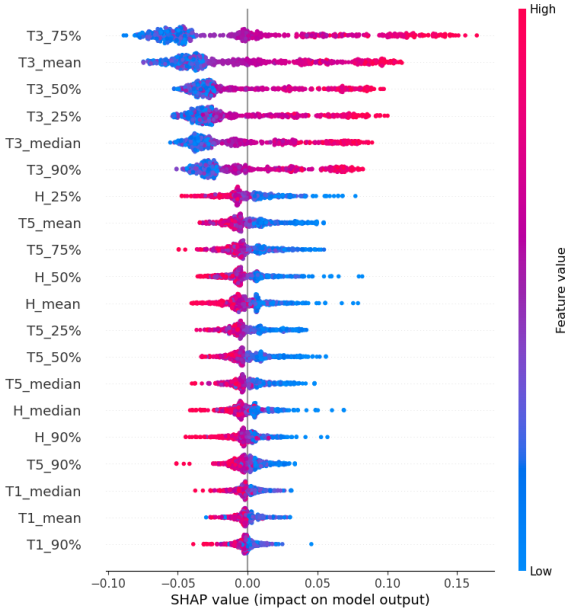
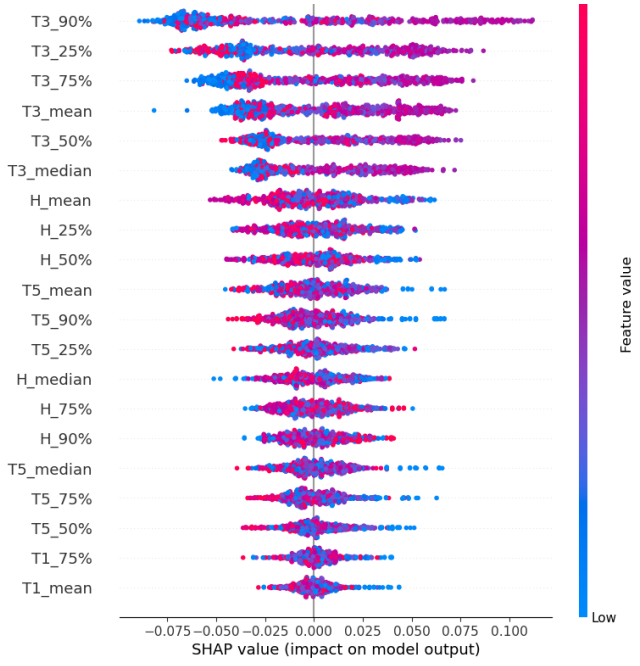


Figura 12: Shap Value classe LOW

Figura 13: Shap Value classe MEDIUM



Analizziamo un grafico alla volta:

- *Grafico 1 (figura 11) :* dal primo grafico rappresentante i Shap Value della classe HIGH notiamo che le features che hanno impattato maggiormente sono sempre le temperature calcolate nella camera 3, ma questa volta osserviamo nel dettaglio che temperature con valori alti lo hanno fatto positivamente mentre temperature con valori bassi negativamente.
- *Grafico 2 (figura 12):* analizziamo ora il grafico della classe LOW. Notiamo subito un risultato opposto al precedente, le temperature alte calcolate nella camera 3 hanno avuto un impatto positivo sul modello, al contrario delle altre features che invece, aumentando il valore, hanno avuto un impatto più negativo.
- *Grafico 3 (figura 13):* per ultimo andiamo ad analizzare il grafico della classe MEDIUM. Emerge subito all'occhio la quantità nettamente superiore di istanze del DataSet avente questa classe, facendo risultare più difficile un'analisi dettagliata di ciascuna features.

5.3.3 BoCSoR

La terza ed ultima analisi sulle feature importance è stata condotta utilizzando il metodo *compute_feature_importance()* e specificando come parametri s=15 e k=10. I risultati ottenuti sono riportati nel grafico sottostante.

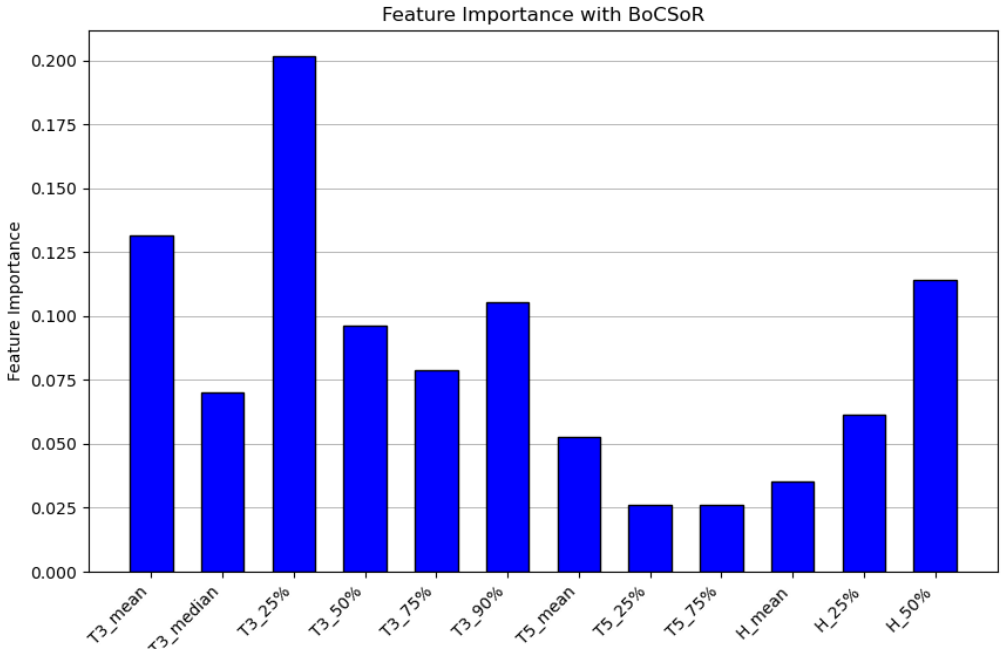


Figura 14: feature importance calcolate tramite BoCSoR

Osservando il grafico in figura 12 notiamo rappresentate solo 12 feature delle 56 contenute nel nostro DataSet. Ciò è dovuto al fatto che queste sono le uniche feature che hanno valori di feature importance diversi da zero. La presenza di molte feature con un valore di feature importance pari a zero potrebbe derivare da una scelta inadeguata dei parametri s e k durante il processo di calcolo.
Inoltre se passiamo ad analizzare quali feature hanno maggiore importanza, notiamo che ancora una volta, la temperatura misurata nella camera numero 3 del macchinario di tostatura risulta la più rilevante per il modello, confermando ciò che era stato analizzato anche attraverso le altre tecniche.
Infine è stato possibile affermare che questo metodo si dimostra estremamente stabile: eseguendo molteplici volte il calcolo delle feature importance ho ottenuto risultati sempre costanti.

5.3.4 Confronto dei risultati

In questa sezione andiamo a confrontare i risultati emersi dal calcolo delle feature importance attraverso i due principali metodi utilizzati fino ad ora: *SHAP*, dove le feature importance sono state valutate utilizzando il classificatore *RandomForest*, e *BoCSoR*, nel quale è stato utilizzato l'algoritmo *CatBoostClassifier*.

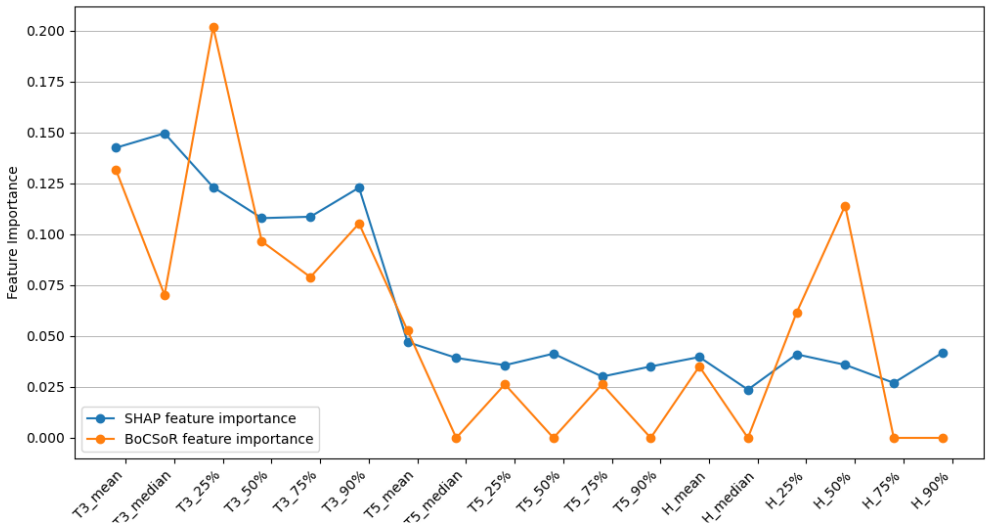


Figura 15: confronto feature importance calcolate con SHAP e BoCSoR

Risalta subito l'importanza che entrambi i metodi afferiscono alla temperatura misurata nella camera 3.
In aggiunta, è importante notare che BoCSoR ha evidenziato risultati che, pur mostrando leggere differenze, sono comunque simili a quelli ottenuti tramite la

rinomata tecnica SHAP. Questa coerenza nei risultati potrebbe essere attribuita alla notevole accuratezza del modello di classificazione CatBoost. La sua capacità di predire le relazioni tra le feature è un indicatore della sua affidabilità nel contesto dell'analisi delle feature importance.

5.4 Matrice di correlazione

Dopo aver messo a confronto i risultati delle features importance ottenuti con i due metodi, SHAP e BoCSoR, è emerso un disaccordo su alcune delle feature analizzate.

L'obiettivo di questo studio è comprendere se questo disaccordo è attribuibile alla correlazione e alla codipendenza tra le feature. Per farlo, è stata analizzata la matrice di correlazione ottenuta dal dataset.

La ***matrice di correlazione*** è una rappresentazione tabellare delle relazioni tra le features di un dataset, esprimendo i valori di correlazione tra di esse su una scala compresa tra -1 e 1. Questa matrice aiuta a comprendere le relazioni tra le diverse caratteristiche e può fornire dettagli sul grado di associazione tra di esse.

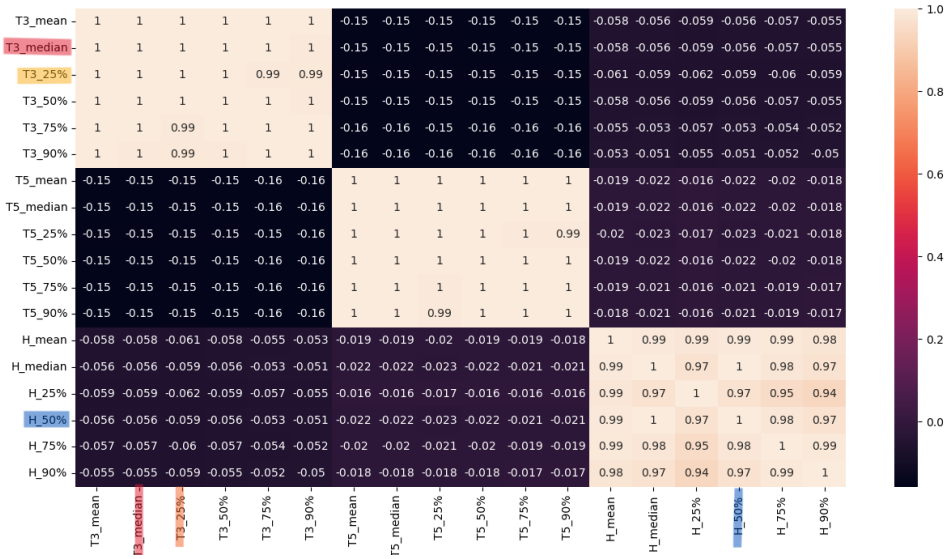


Figura 16: matrice di correlazione

Osservando la matrice di correlazione ottenuta selezionando le 18 features con valore di features importance significativo, notiamo sulla diagonale, come potevamo aspettarci, una forte correlazione tra le caratteristiche che rappresentano le statistiche di un determinato insieme di dati, come ad esempio la temperatura calcolata nelle camere 3 e 5 e l'umidità presente nella materia prima.

Inoltre sono state evidenziate le 3 features che dal grafico di confronto tra BoCSoR e SHAP risultavano in maggiore disaccordo*: T3_median, T3_25% e H_50%.*
Analizzando la correlazione delle 3 features evidenziate con le altre caratteristiche possiamo dire che esse sono fortemente correlate con le features che rappresentano statistiche dello stesso insieme di dati ma, al contrario, con tutte le altre caratteristiche risulta una debole correlazione, talvolta quasi nulla.

5.5 Analisi con Dataset ridotto

Dopo aver analizzato la matrice di correlazione, che ha evidenziato una forte correlazione raggruppata delle features, abbiamo selezionato solo due features per ciascun gruppo. Successivamente, abbiamo ripetuto l'analisi dell'importanza delle features utilizzando i metodi SHAP, BoCSoR e RandomForest, al fine di valutare se questa riduzione delle dimensioni porta a un miglioramento delle prestazioni complessive del modello.

Per questa secondo studio, sono state selezionate le colonne relative alla media e alla mediana dei valori per ogni insieme di dati.

- ***Media****:* la media può essere utile per ottenere una stima del valore centrale dell'insieme di dati. Tuttavia, è importante notare che la media può essere influenzata da valori estremi o outlier presenti nei dati. Se ci sono outlier significativi, la media potrebbe non riflettere accuratamente la tendenza centrale del dataset.
- ***Mediana****:* la mediana, al contrario, è resistente agli outlier e fornisce una misura robusta della tendenza centrale. Essa rappresenta il valore centrale dell'insieme di dati ordinato e non è influenzata dai valori estremi.

Selezionando queste due colonne, è possibile ottenere una rappresentazione più bilanciata della tendenza centrale dei dati, considerando sia la media che la mediana, e valutare l'importanza delle feature in base a queste metriche. Questo approccio può fornire una visione più completa della distribuzione dei dati e delle loro caratteristiche rilevanti per l'analisi.

Prima di procedere con l’analisi delle feature importance è stata calcolata l’accuratezza del classificatore RandomForest per valutare le prestazioni del modello, dato che è stato modificato il Dataset. È molto importante calcolare l'accuratezza perché ci indica quanto il modello è affidabile nel fare predizioni corrette su nuovi dati.

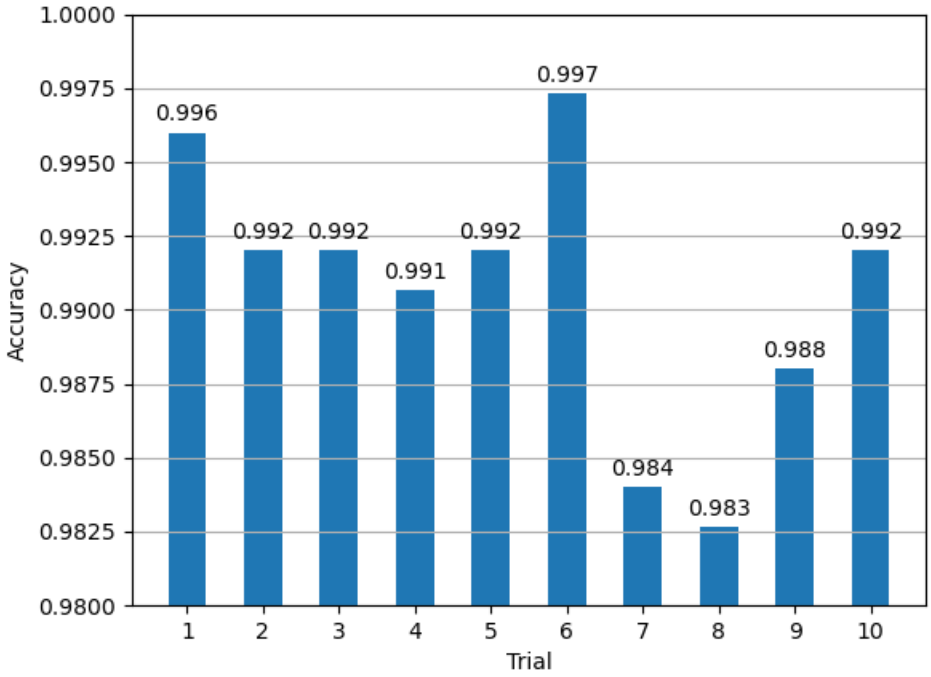


Figura 17: accuratezza RandomForestClassifier con Dataset ridotto

Il grafico soprastante in Figura 17 ci mostra l'accuratezza del modello *RandomForestClassifier* ottenuta svolgendo 10 trial con il nuovo Dataset ridotto. È evidente che le prestazioni del modello rimangono eccellenti, mantenendosi costantemente sopra il 98%. Questo risultato sottolinea la robustezza e l'affidabilità del modello nel fare predizioni accurate su dati non visti, confermando la sua efficacia anche con un set di dati ridotto.

Adesso confrontiamo i risultati emersi dal calcolo delle feature importance con il nuovo DataSet ridotto attraverso i tre principali metodi utilizzati fino ad ora, SHAP , BoCSoR e il metodo nativo di RandomForest.

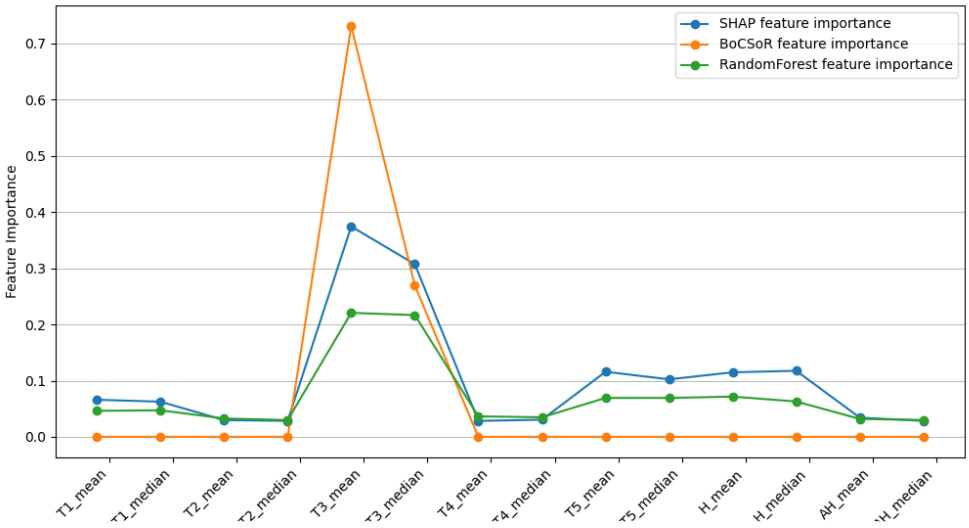


Figura 18: confronto feature importance calcolate con SHAP, BoCSoR e RandomForest

Risalta subito un aumento del valore assegnato alle features, con valori che superano addirittura lo 0.5, a differenza del picco massimo precedentemente osservato, che era solo del 0.2. Questo suggerisce che la selezione delle feature basata sulla media e sulla mediana abbia portato a una maggiore rilevanza delle features nel modello.
Inoltre, si osserva che la feature più importante per tutti e tre i metodi rimane la temperatura misurata nella camera numero 3, confermando la sua forte influenza sui risultati del modello.
È interessante notare che il metodo BoCSoR assegna importanza solo alle features *'T3_mean' e 'T3_median'*, impostando a 0 tutti gli altri valori. Questo suggerisce che, secondo il metodo BoCSoR, solo queste due features sono rilevanti per il modello, mentre le altre non contribuiscono significativamente.
In aggiunta possiamo evidenziare il notevole aumento dei valori di feature importance ottenuti con il metodo nativo di RandomForest. Andandoli a confrontare con quelli ottenuti precedentemente, notiamo che il valore massimo è passato da essere pari a 0.08 ad un valore di 0.2. Questo aumento può indicare che il modello è ora in grado di individuare e dare maggior peso alle features più rilevanti per la predizione dei risultati.
Infine, è rilevante notare che le tecniche mostrano un andamento simile e non evidenziano importanti scostamenti o disaccordi. Questo suggerisce che tutti e tre i metodi convergono su conclusioni simili riguardo all'importanza delle features nel modello, confermando l'affidabilità e la coerenza delle analisi condotte.

5.6 Risultati dello studio della parametrizzazione per BoCSoR

Un fenomeno importante, riguardante lo studio condotto su BoCSoR, per il calcolo delle feature importance, consiste nel numero di queste che vengono poste uguali a zero. Questo dato è significativo perché tipicamente sta a indicare una errata scelta parametrica.
In questa analisi andremo proprio a valutare i cambiamenti dei risultati dovuti a valori diversi dei parametri s e k della funzione *BoundaryCrossingSoloRatio.*

S	k	Features = 0	s	k	Features = 0
10	***10***	12	***22***	***10***	12
10	***14***	11	***22***	***14***	11
10	***18***	11	***22***	***18***	10
10	***22***	11	***22***	***22***	10
10	***26***	11	***22***	***26***	10
10	***30***	10	***22***	***30***	10
14	***10***	12	***26***	***10***	12
14	***14***	11	***26***	***14***	11
14	***18***	10	***26***	***18***	10
14	***22***	10	***26***	***22***	10
14	***26***	10	***26***	***26***	10
14	***30***	10	***26***	***30***	10
18	***10***	12	***30***	***10***	11
18	***14***	11	***30***	***14***	10
18	***18***	10	***30***	***18***	9
18	***22***	10	***30***	***22***	9
18	***26***	10	***30***	***26***	9
18	***30***	10	***30***	***30***	9

Figura 19: Numero di feature uguali a 0 per ogni combinazione dei parametri 's' e 'k'

Da questa analisi possiamo notare come l'aumento dei parametri, ovvero l'aumento dei punti intermedi tra ogni possibile controfattuale e il campione originale e l'aumento dei 'vicini' della classe controfattuale, porta ad ottenere sempre meno features con importanza uguale a zero migliorando le prestazioni.

Individuati i parametri che portano ad avere minori features con valore uguale a 0 (evidenziati in giallo in *figura 19*), abbiamo analizzato le feature importance ottenute impostando tali valori per i parametri 's' e 'k' nella funzione *BoundaryCrossingSoloRatio.*

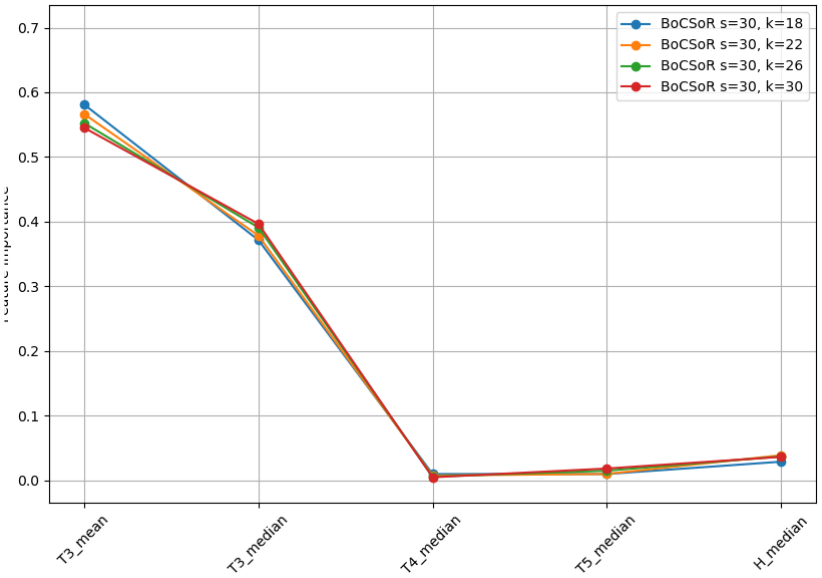


Figura 20: feature importance ottenute con le combinazioni ottime dei parametri 's' e 'k'

Rispetto alla configurazione con i parametri *s=15* e *k=10*, osserviamo che con queste nuove configurazioni abbiamo 5 feature con valori di feature importance diversi da zero, garantendoci una statistica contenente le features *T4_median, T5_median e H_median.*
È interessante notare che tutte e 4 le diverse parametrizzazioni hanno prodotto risultati molto simili, confermando ancora una volta l'importanza della temperatura misurata nella camera 3. Tuttavia, il valore massimo della feature importance non supera più lo 0.7, rimanendo comunque elevato e superiore allo 0.5.
Possiamo quindi dedurre che un aumento dei valori di 's' e 'k' ha portato a un aumento del numero di feature con valori di feature importance maggiori di zero. Tuttavia, però ha ridotto il valore delle feature ritenute maggiormente importanti per il modello, poiché il valore massimo è diminuito rispetto alla configurazione precedente. Questo suggerisce che, anche se più feature vengono considerate rilevanti, ciascuna ha meno peso nel modello rispetto a prima, quando avevamo un valore massimo di feature importance più alto.

In conclusione, possiamo affermare che l'analisi dei nuovi parametri ci conferma ulteriormente l'importanza significativa della temperatura misurata nella camera 3 rispetto alla qualità del prodotto. Questo è evidenziato dal fatto che le feature importance relative a questa temperatura rimangono costantemente elevate nelle diverse configurazioni esaminate. Anche se potrebbe esserci un lieve aumento delle feature importance relative ad altre caratteristiche, la temperatura misurata nella camera 3 continua ad essere un fattore chiave nell'influenzare il modello di ML. Questa consistenza nelle feature importance conferma l'importanza cruciale di monitorare e controllare attentamente la temperatura nella camera 3 per garantire la qualità del prodotto.

6. Conclusioni

Dopo un'attenta analisi dei risultati ottenuti possiamo trarre le seguenti conclusioni con un buon grado di oggettività:

1. La valutazione dell'accuratezza ha mostrato che, nell'ambito del nostro dataset, il modello basato su alberi decisionali si è distinto per ottenere le migliori prestazioni.
2. L'analisi delle feature importance attraverso metodi come SHAP, BoCSoR e l'uso di funzioni come feature_importances ci hanno fornito una comprensione approfondita delle caratteristiche chiave nei dati. L'ampia gamma di metodologie utilizzate ha consentito di delineare un quadro completo e dettagliato delle influenze delle features sul nostro modello, fornendoci preziose informazioni di contesto
3. Durante il confronto tra i vari algoritmi di valutazione, è emerso un interessante fenomeno di convergenza. In particolare, BoCSoR ha dimostrato una notevole coerenza nei risultati ottenuti, con differenze minime rispetto agli altri metodi. Tale coerenza suggerisce la solidità e l'affidabilità del modello BoCSoR nel contesto specifico della nostra analisi.
4. L'analisi delle correlazioni tra i dati si è rivelata cruciale per una migliore comprensione dell'importanza delle caratteristiche per il modello. Questo approccio ci ha permesso di identificare e valutare le relazioni tra le variabili, fornendo così una base solida per la selezione e l'ottimizzazione delle feature nel nostro modello.

Il risultato iniziale evidenzia che l'algoritmo di classificazione ***RandomForestClassifier*** si è distinto come il migliore tra quelli testati. Questo successo è attribuibile all'utilizzo degli alberi decisionali come fondamento del modello. Gli alberi decisionali sono noti per la loro facilità di utilizzo e per la loro capacità di addestrare il modello in modo efficiente in termini di tempo. Inoltre, questi modelli sono in grado di usare dati eterogenei ed essere robusti in caso di mancanza di dati.

Durante il primo calcolo delle feature importance, dove è stato utilizzato l'intero dataset, abbiamo ottenuto come caratteristiche più rilevanti *T3_mean, T3_median, T3_90%* con l'analisi svolta utilizzando SHAP, mentre abbiamo ottenuto *T3_mean, T3_25%, H_50%* con l'analisi svolta da BoCSoR. Tuttavia, è importante evidenziare che i valori ottenuti per l'importanza delle feature sono stati generalmente poco superiori allo zero, suggerendo una scarsa efficacia dei

risultati.
Si è ritenuto necessario effettuare una seconda analisi delle feature importance con il dataset ridotto, focalizzandosi esclusivamente sulle features riguardanti la media e la mediana di ogni insieme di dati. Grazie a questo studio abbiamo ottenuto notevoli miglioramenti nei risultati delle feature importance con i tre metodi utilizzati. Le caratteristiche più rilevanti emerse sono risultate *T3_mean e T3_median* in tutti e tre gli approcci analizzati.
Questi risultati confermano ulteriormente l'importanza significativa della temperatura misurata nella camera 3 rispetto alla qualità del prodotto. Questo è evidenziato dal fatto che le feature importance relative a questa temperatura rimangono costantemente elevate, indipendentemente dal metodo di analisi utilizzato. Anche se potrebbe esserci un lieve aumento delle feature importance relative ad altre caratteristiche, la temperatura misurata nella camera 3 continua ad essere un fattore chiave nell'influenzare il modello di Machine Learning. Questa consistenza nelle feature importance conferma l'importanza cruciale di monitorare e controllare attentamente la temperatura nella camera 3 per garantire un'elevata qualità del prodotto.
Un aspetto fondamentale nell'analisi di BoCSoR è stato scegliere i parametri 's' e 'k'. In particolare, all'aumentare di questi parametri si nota una diminuzione delle feature importance pari a zero, migliorando decisamente le prestazioni del modello.

In questa tesi è stato trattato il fenomeno dell'intelligenza artificiale spiegabile (XAI), applicato al caso dell'industria 4.0, evidenziandone la fondamentale utilità. Non solo ha contribuito a fornire una comprensione approfondita dei risultati ottenuti, ma ha anche reso più accessibile l'analisi dei dati ad un vasto pubblico, come abbiamo potuto vedere con lo use-case.
L'obiettivo principale è stato quello di migliorare l'affidabilità e l'interpretazione dei modelli basati su Machine Learning, ottenendo risultati soddisfacenti. Ciò è stato possibile grazie all'analisi approfondita del dataset e all'utilizzo di metodi per il calcolo delle feature importance, in particolare BoCSoR che attraverso l'utilizzo dei controfattuali ha rivelato avere un costo computazionale minore rispetto ad altri approcci, contribuendo a una maggiore efficacia nell'analisi dei modelli.

In conclusione, questo studio ha rappresentato un contributo significativo all'intelligenza artificiale spiegabile nel contesto dell'industria 4.0. Il lavoro ha fornito una guida per futuri sviluppi nel settore, promuovendo una maggiore trasparenza e comprensione dei modelli di machine learning utilizzati in questo ambito critico.

Bibliografia

[1] Arrieta AB, Rodríguez ND, Ser JD, et al (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fus

[2] A. Aas, Løland, Explaining individual predictions when features are dependent: More accurate approximations to shapley values, arXiv preprint arXiv:1903.10464, (2019).

[3] Byrne RM (2019) Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI, pp 6276–6282

[4] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

[5] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, and Rajiv Ranjan. 2023. Explainable AI (XAI): Core Ideas, Techniques, and Solutions. ACM Comput. Surv. 55, 9, Article 194 (September 2023), 33 pages.