UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

Sviluppo e ottimizzazione di algoritmi di visione volti alla diagnosi di difetti di produzione

Development and optimization of vision algorithms for the diagnosis of production faults

Relatore:
Prof. Freddi Alessandro

Tesi di Laurea di:
Angelone Mattia Domenico

Correlatori:
Ing. Grisostomi Massimo

Anno Accademico 2023/2024

Sommario

Introduzione

Questo elaborato ha l'obiettivo di sviluppare e applicare algoritmi di visione per il controllo qualità nella produzione industriale. Nello specifico, è stata svolta in collaborazione con l'azienda META S.r.l. , un'azienda nata nel 2014 come spin-off dell'Università Politecnica delle Marche, durante il periodo di tirocinio e ha avuto come ambito applicativo il controllo qualità di scaldabagni elettrici prodotti da una nota multinazionale italiana (Ariston Group). La richiesta si basava sull'implementazione di un algoritmo per la diagnosi di difetti di produzione al fine di rilevale la presenza di fault durante il processo di smaltatura interna dei boiler, processo fondamentale nella lavorazione degli stessi.

Lo scopo era quello di automatizzare tutto il processo di produzione tramite l'utilizzo di un software di fault detection che, tramite l'analisi di features estratte da foto interne del boiler, vada a diagnosticare la presenza di un difetto di produzione, riducendo al minimo la presenza di un operatore che aveva lo scopo di validare ogni singolo pezzo mediante soglie fisse poste ad ogni feature estrapolata.

La tesi sarà così suddivisa:

- Capitolo 1: panoramica generale sui temi affrontati durante la tesi, analizzando tutto l'ambito riguardante la fault diagnosis e il quality control, introducendo e spiegando il machine learning e deep learning ed analizzando la visione artificiale e gli algoritmi di visione. Piccolo focus finale sugli strumenti utilizzati per portare a termine il lavoro, quali linguaggi di programmazione e ambiente di sviluppo.

- Capitolo 2: descrizione del processo in analisi ed i relativi problemi che si possono riscontrare in controllo qualità, descrivendoli in modo approfondito anche tramite le foto dalle quali sono state estratte le features

- Capitolo 3: introduzione di tutti i dati estrapolati per il quality control e contestualmente analisi dei metodi di lavoro utilizzati al fine di raggiungere l'obiettivo prefissato, descrivendo il pre-processamento dei dati, il ranking features, il bilanciamento del dataset in favore della fault label meno numerosa, la creazione ed addestramento della rete neurale ed il testing con il calcolo delle varie metriche per la valutazione.

- Capitolo 4: esposizione dei risultati del lavoro svolto, partendo dal training del modello di diagnosi fino al testing, mostrando tutte le metriche di valutazione sia quantitative che grafiche.

- Capitolo 5: chiusura della tesi con le conclusioni finali sul lavoro svolto ed i risultati raggiunti. Introduzione ed analisi di sviluppi futuri per il miglioramento dei risultati ottenuti.

Capitolo 1

Stato dell'Arte

1.1 Fault Diagnosis e Quality Assurance

1.1.1 Diagnosi Guasti

La Fault Diagnosis, o diagnosi dei guasti, riveste un ruolo cruciale in vari ambiti tecnologici e industriali. Con l'aumento delle richieste di efficienza e qualità del prodotto, nonché l'integrazione crescente dei sistemi di controllo automatico in processi meccatronici costosi e critici per la sicurezza, la supervisione, la rilevazione dei guasti e la diagnosi dei guasti giocano un ruolo importante nell'ambito della manutenzione preventiva (Figura 1).

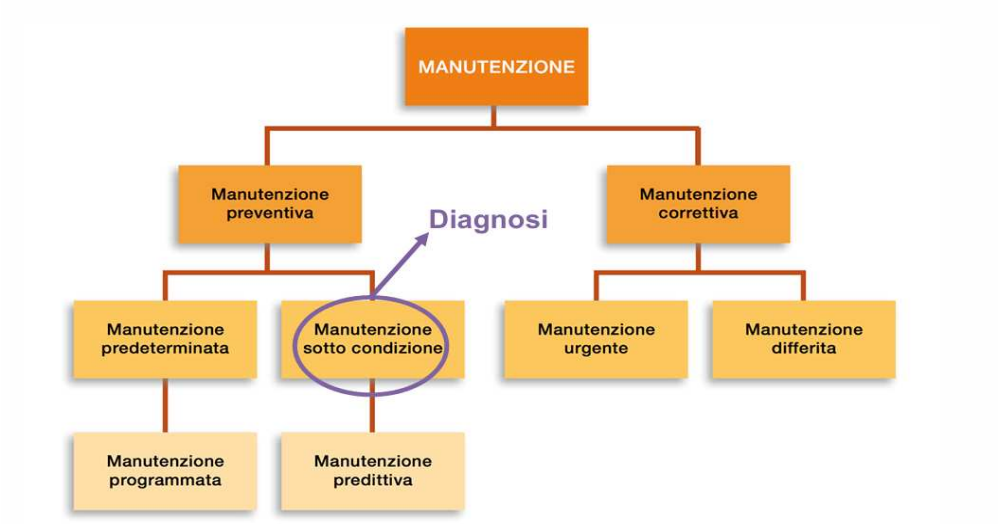


Figura 1: Collocamento della Fault Diagnosis nella famiglia della Manutenzione (e.g. Freddi, 2022)

Poiché l'obiettivo è prevenire un eventuale fault non attraverso interventi di manutenzione programmati ma attraverso analisi di dati, è stata introdotta l'idea di diagnosticare un guasto proprio come la diagnosi di una malattia, partendo dai sintomi (nel nostro caso i dati affetti da variazione) per arrivare alla causa (ovvero il fault). Consiste quindi nell'identificare, localizzare e comprendere i guasti nei sistemi o nelle attrezzature.

Quando si parla di diagnosi di un guasto, in realtà si sta facendo riferimento ad un insieme di tecniche che hanno lo scopo di individuare, isolare e identificare un guasto:

- **Fault Detection**: determinazione dei fault presenti nel sistema e dei momenti in cui essi sono individuati.

- **Fault Isolation**: determinazione della tipologia, posizione e tempo di individuazione del fault.

- **Fault Identification**: determinazione della dimensione e del comportamento tempo variante di un fault.

La Fault Diagnosis è l'unione della Fault Isolation ed Identification ed insieme alla Fault Detection rappresenta tutto il funzionamento della diagnosi di un fault in un sistema automatico (Abid, Khan, Iqbal, 2020).

1.1.2 Controllo Qualità

Il lavoro di tesi si basa sui principi della Fault Diagnosis ma non per identificare guasti al fine di mantenere l'affidabilità e la sicurezza di sistemi automatici, bensì per prevenire errori e concentrarsi sulla conformità agli standard di qualità. Questi principi sono alla base del Controllo Qualità: un processo fondamentale per garantire che i prodotti o i servizi soddisfino gli standard richiesti. Coinvolge l'analisi, il monitoraggio e l'ottimizzazione dei processi per ridurre al minimo gli errori e massimizzare la qualità.
La logica dietro la Quality Assurance è la stessa che ritroviamo nella Fault Diagnosis, ossia si concentra sulla prevenzione piuttosto che sulla correzione con l'obiettivo è evitare che gli errori si verifichino in primo luogo (Mrugalska, Tytyk, 2015).
Queste similitudini hanno portato a trattare un problema di controllo qualità come un problema di diagnosi guasti, nel quale per l'identificazione e la diagnosi esistono due metodi principali:

1) **Inferenza**: metodo che permette di diagnosticare un difetto di produzione sfruttando la conoscenza, ove disponibile, delle relazioni causali che regolano il funzionamento del sistema.

2) **Classificazione**: metodo che permette di diagnosticare un difetto di produzione tramite il Machine Learning, una tecnica avanzata con la quale da un set di dati di addestramento si allena un algoritmo di classificazione.

Nell'ambito della classificazione, oltre ai vari algoritmi di Machine Learning è stato introdotto un nuovo concetto di addestramento automatico: il Deep Learning tramite Reti Neurali Artificiali (Frank, Koppen-Seliger, 1997).

1.2 Machine Learning e Deep Learning

1.2.1 Machine Learning

Il Machine Learning (ML) è lo studio scientifico degli algoritmi e dei modelli statistici che i sistemi informatici utilizzano per eseguire specifiche attività senza essere programmati esplicitamente (Mahesh,2019). Gli algoritmi di apprendimento vengono utilizzati in molte applicazioni che utilizziamo quotidianamente. Ad esempio, ogni volta che utilizziamo un motore di ricerca web come Google per cercare informazioni su Internet, uno dei motivi per cui funziona così bene è l'utilizzo di un algoritmo di apprendimento che ha imparato a classificare le pagine web. Questi algoritmi vengono utilizzati per scopi diversi, come il data mining, l'elaborazione delle immagini, l'analisi predittiva, ecc. Il principale vantaggio dell'utilizzo del machine learning è che, una volta che un algoritmo ha imparato come gestire i dati, può svolgere automaticamente il suo lavoro.

Si basa su diversi algoritmi per risolvere problemi legati ai dati. Non esiste un unico algoritmo adatto a tutti i casi: la scelta dipende dal tipo di problema da risolvere, dal numero di variabili coinvolte e dal modello più adatto (Figura 2).

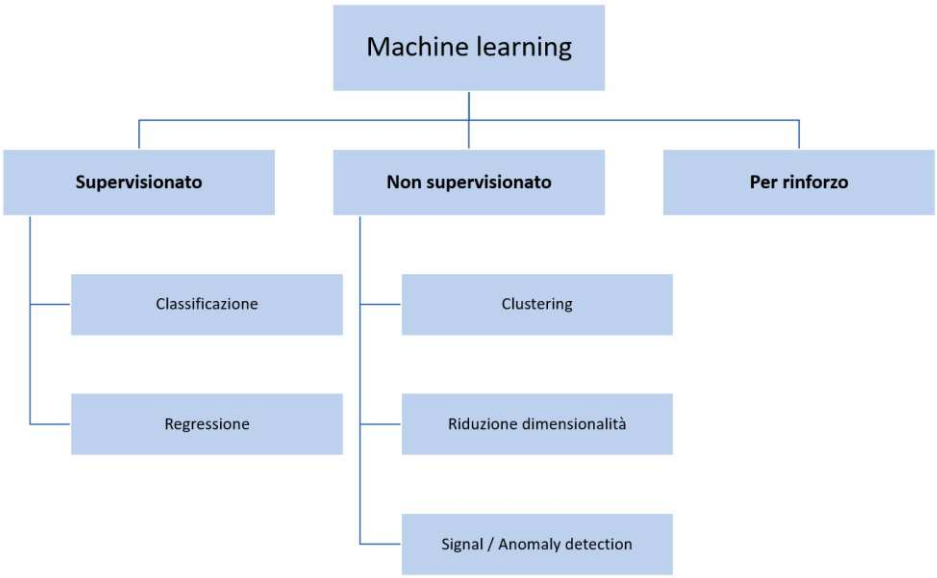


Figura 2: Algoritmi di Machine Learning (e.g. D'Agostino, 2022)

Tra i vari algoritmi di machine learning, possiamo introdurre tre grandi famiglie che rappresentano quasi la totalità dei modelli di addestramento:

- **Apprendimento Supervisionato** (Figura 3): una delle principali metodologie nel campo del machine learning, consiste nell'utilizzo di un dataset etichettato, cioè un insieme di dati in cui ogni esempio di input è associato a un output corretto. L'obiettivo è addestrare un modello che possa fare previsioni accurate su nuovi dati non etichettati basandosi sui pattern appresi dai dati di addestramento.

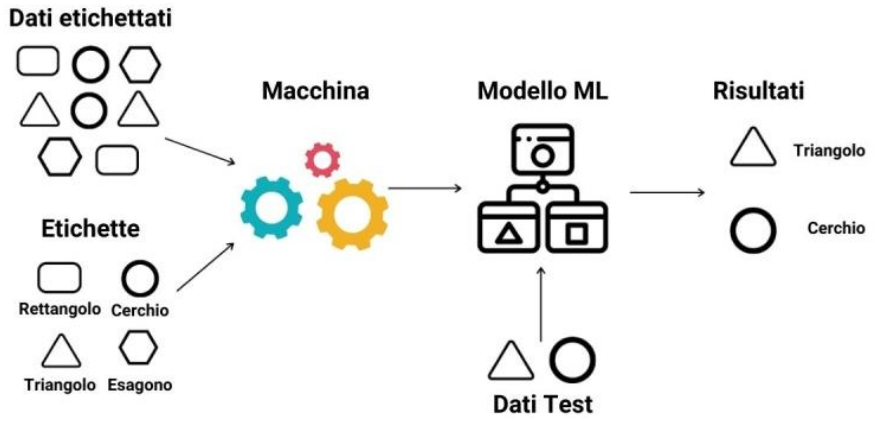


Figura 3: Apprendimento Supervisionato (e.g. Esker, 2023)

- **Apprendimento non supervisionato** (Figura 4): metodologia del machine learning in cui l'algoritmo impara a identificare pattern e strutture nei dati senza l'uso di etichette predefinite. A differenza dell'apprendimento supervisionato, non ci sono output noti associati ai dati di input. L'obiettivo è esplorare i dati e scoprire informazioni nascoste o raggruppamenti naturali.

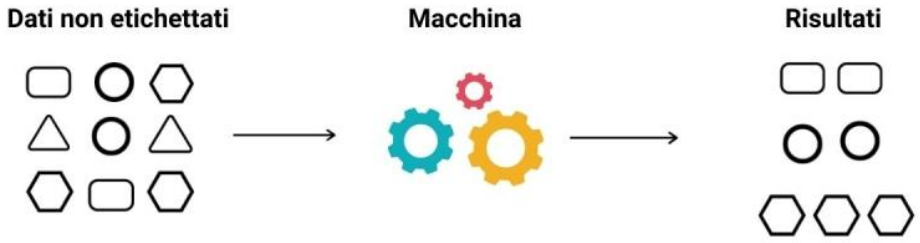


Figura 4: Apprendimento Non Supervisionato (e.g. Esker, 2023)

- **Apprendimento per rinforzo**: area del machine learning che si occupa di come gli algoritmi dovrebbero intraprendere azioni in un ambiente al fine di massimizzare una certa nozione di ricompensa cumulativa.

1.2.2 Deep Learning

Come detto precedentemente, nel mondo del Machine Learning possiamo trovare vari sottogruppi di algoritmi, ma recentemente è stato introdotto un sottoinsieme del Machine Learning che rappresenta l'apice di questa evoluzione, portando l'apprendimento automatico a un livello superiore di complessità e abilità: il Deep Learning (Figura 5).

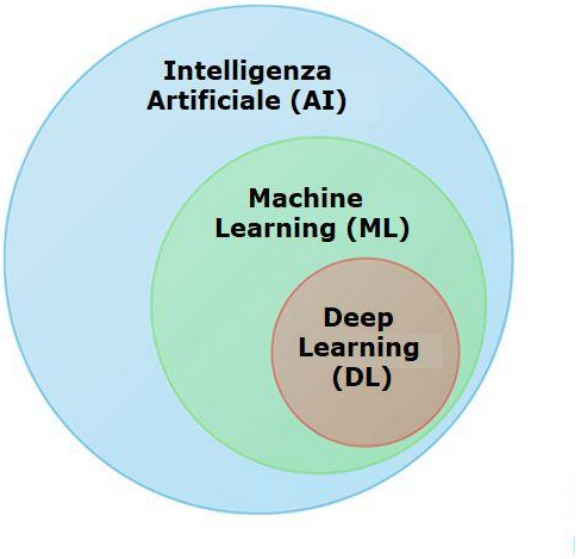


Figura 5: AI, Machine Learning e Deep Learning (e.g. D'Agostini, 2018)

Il deep learning è un concetto di apprendimento automatico basato su reti neurali artificiali (Figura 6). Per molte applicazioni, i modelli di deep learning superano i modelli di apprendimento automatico superficiali e gli approcci tradizionali di analisi dei dati (Janiesch, Zschech, Heinrich, 2021).

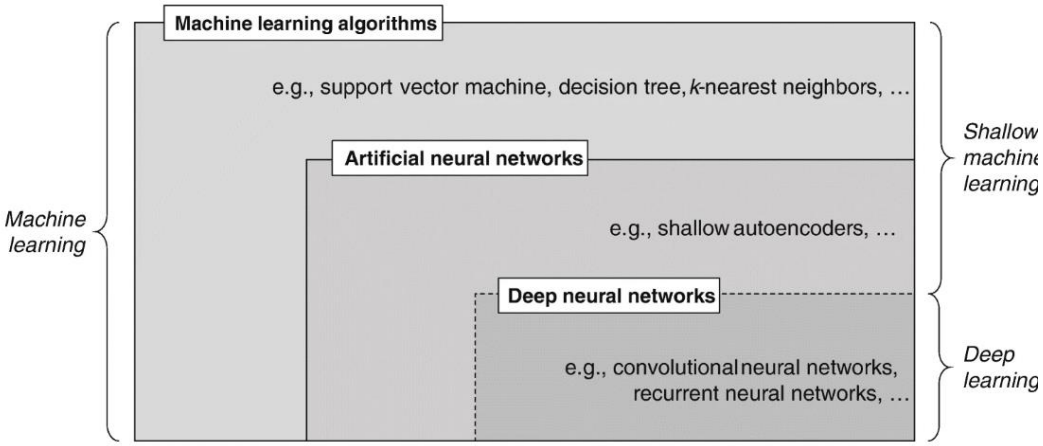


Figura 6: ANN e Deep Learning (e.g. Goodfellow, Bengio, Courville, 2016)

Una delle famiglie di modelli di apprendimento automatico sono le cosiddette Reti Neurali Artificiali (ANN): sono modelli di calcolo ispirati al funzionamento del cervello umano. Sono costituite da unità di elaborazione connesse, chiamate neuroni artificiali, che trasmettono segnali. Le ANN tradizionali hanno una struttura flessibile che può essere modificata per una varietà di contesti e tipicamente includono uno o più strati (hidden layers) nascosti tra lo strato di input (input layer) e quello di output (output layer). La forma più semplice è una rete neurale artificiale che utilizza un solo layer nascosto, molto simile al modello di regressione lineare.

Si inizia ad entrare nell'ambito del Deep Learning quando gli hidden layer iniziano ad aumentare e quindi si parla di reti neurali con molteplici strati nascosti, chiamate reti neurali profonde.

Queste reti sono in grado di apprendere automaticamente rappresentazioni dei dati con livelli di astrazione sempre più elevati. A differenza delle ANN tradizionali, il DL può gestire input grezzi (foto, video, ecc) e scoprire autonomamente le rappresentazioni necessarie per la classificazione o la predizione.

Grazie al loro apprendimento avanzato, le reti neurali profonde sono utilizzate in diversi ambiti (Sharma N, Sharma R, Jindal, 2021):

- Riconoscimento Vocale: traduzione di parole parlate in testo, utile in vari settori come l'assistenza sanitaria e i sistemi automobilistici.

- Analisi Medica: utilizzo di reti neurali per diagnosticare malattie attraverso immagini radiografiche e rilevare variazioni nei parametri di salute.

- Visione: utilizzo di algoritmi per elaborare, analizzare e riconoscere dati visivi, con applicazioni come il riconoscimento facciale e la scansione di documenti scritti a mano.

1.3 Algoritmi di Visione in Ambito Industriale

La visione artificiale è una disciplina che si occupa di fornire alle macchine la capacità di "vedere" e interpretare le immagini. Questa capacità è fondamentale in molte applicazioni industriali, dove le macchine devono essere in grado di riconoscere e manipolare oggetti in modo preciso e affidabile.

Gli algoritmi di visione, che sono i metodi computazionali utilizzati per interpretare le immagini, sono quindi un componente chiave di questi sistemi. In ambito industriale, gli algoritmi di visione sono utilizzati per una serie di applicazioni, tra cui l'ispezione di qualità, la navigazione robotica, e il riconoscimento di oggetti. Queste applicazioni richiedono algoritmi in grado di rilevare e localizzare gli oggetti all'interno di un'immagine, seguire il movimento di questi oggetti nel tempo, e identificare eventuali difetti o anomalie (Figura 7).

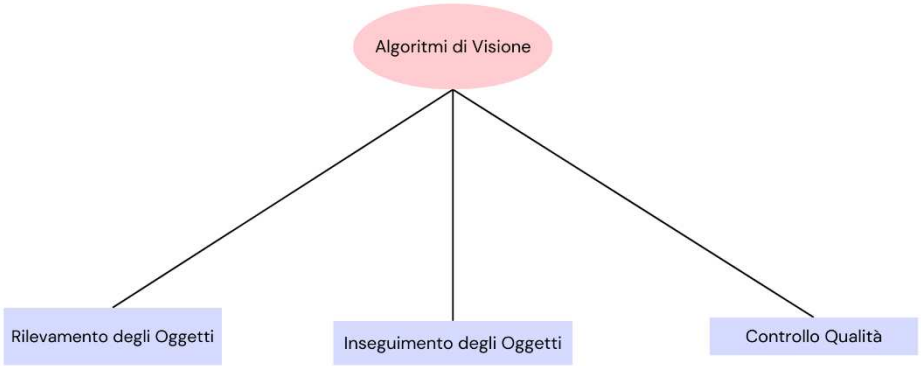


Figura 7: Algoritmi di Visione

1.3.1 Algoritmi di Rilevamento degli Oggetti

Gli algoritmi di rilevamento degli oggetti sono una componente fondamentale della visione artificiale, con applicazioni che vanno dall'automazione industriale alla sicurezza. Questi algoritmi sono progettati per identificare e localizzare gli oggetti all'interno di un'immagine o di una serie di immagini (Al-Kaff, Martín, García, de la Escalera, María Armingol, 2018).

Ci sono molti algoritmi di rilevamento degli oggetti, ma due dei più comuni sono:

- **R-CNN** (Region-based Convolutional Neural Networks): Questo algoritmo utilizza una combinazione di reti neurali convoluzionali e proposte di regioni per rilevare gli oggetti (Potrimba, 2023).

- **YOLO** (You Only Look Once): A differenza di R-CNN, YOLO esegue la rilevazione degli oggetti in un'unica passata, il che lo rende significativamente più veloce (Liu, Tao, Liang, Li, Chen, 2018).

1.3.2 Algoritmi di Inseguimento degli Oggetti

Gli algoritmi di inseguimento degli oggetti sono una categoria di algoritmi di visione artificiale che si concentrano sul tracciamento del movimento di un oggetto nel tempo (Dutta, Mondal, Dey, Sen, Moraru, Hassanien, 2020). Questi algoritmi sono particolarmente utili in applicazioni come la videosorveglianza, la navigazione robotica e l'analisi del movimento.
Due esempi comuni di algoritmi di tracking degli oggetti sono:

- **DeepSort**: Questo algoritmo avanzato di tracking visivo utilizza il filtro di Kalman per il tracciamento degli oggetti e introduce un descrittore di aspetto basato su apprendimento profondo per ridurre gli scambi di identità, rendendo il tracciamento più efficiente (Hou, Wang, Chau, 2019)

- **MeanShift** e **CamShift**: Questi algoritmi sono particolarmente utili per il seguimento degli oggetti in video. MeanShift utilizza la densità di probabilità per localizzare l'oggetto, mentre CamShift estende MeanShift per includere l'adattamento alla scala e alla rotazione (Wen, Cai, 2006).

1.3.3 Algoritmi di Controllo Qualità

Gli algoritmi di controllo qualità sono una componente essenziale dei sistemi di visione artificiale utilizzati nell'industria manifatturiera. Questi algoritmi sono progettati per ispezionare automaticamente i prodotti e identificare eventuali difetti o anomalie (De Ketelaere, Wouters, Kalfas, Van Belleghem, Saeys, 2022).

Ci sono molti tipi di algoritmi di controllo qualità, ma due dei più comuni sono:

- **Algoritmi di Template Matching**: Questi algoritmi confrontano un'immagine di un prodotto con un'immagine di riferimento per identificare eventuali difetti (Brunelli, 2021).

- **Reti neurali convoluzionali** (CNN): Le CNN possono essere addestrate per riconoscere una vasta gamma di difetti, rendendole estremamente versatili per l'ispezione di qualità (Wang, Chen, Qiao, 2018).

Il lavoro di tesi si baserà proprio sulla programmazione di un algoritmo di visione per il controllo qualità basato su una rete neurale con 3 hidden layers con lo scopo di creare un modello di classificazione numerica al fine di diagnosticare eventuali difetti di produzione in uno specifico processo lavorativo durante la realizzazione di scaldabagni elettrici per uso domestico.

1.4 Software e Linguaggi

Di seguito è riportata una panoramica dei software e linguaggi di programmazioni utilizzati durante il lavoro di tesi (Figura 8):

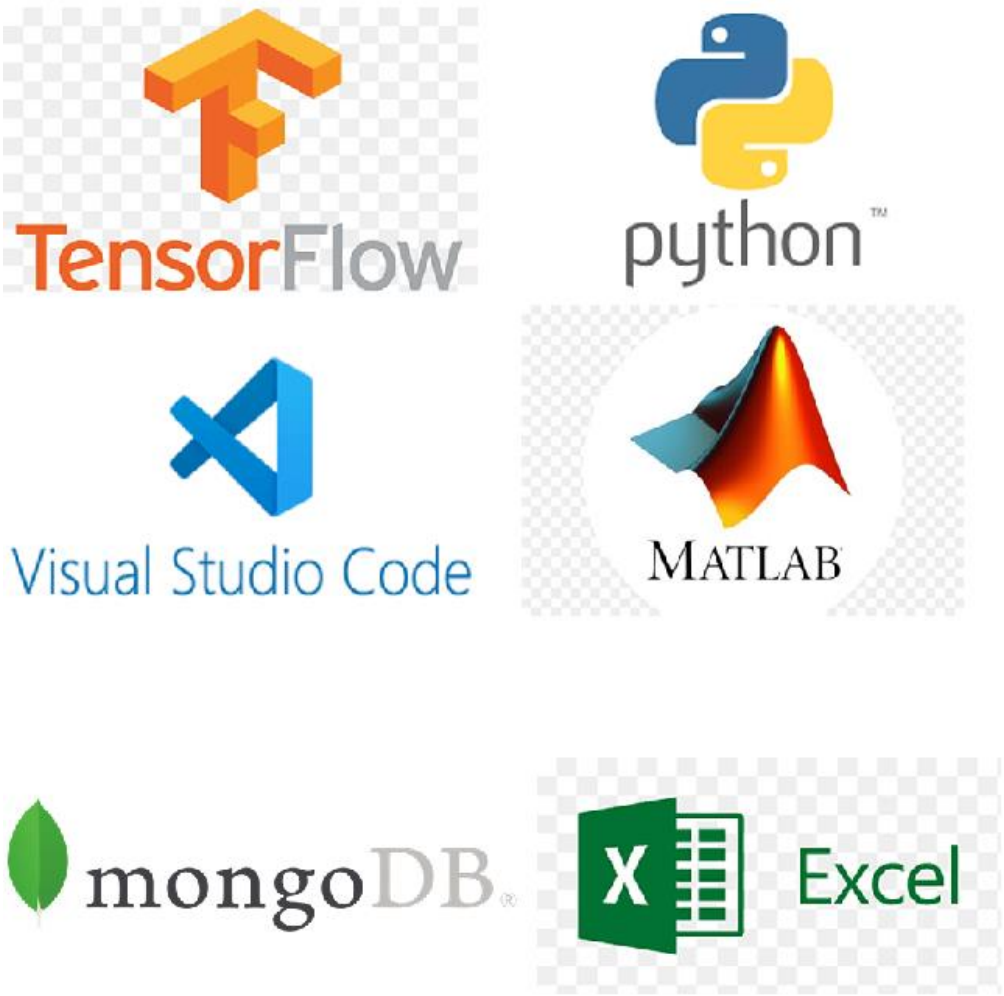


Figura 8: Software e linguaggi usati nel lavoro di tesi

Capitolo 2

Processo di Smaltatura

Il lavoro di tesi si basa sulla richiesta di un sistema per il rilevamento dei difetti di produzione da parte dell'azienda Ariston Group per rilevare eventuali fault progettuali durante una fase cruciale nel processo di fabbricazione di scaldabagni elettrici.

La creazione di boiler elettrici è un processo industriale complesso che coinvolge diverse fasi, dall'assemblaggio delle componenti metalliche fino alla smaltatura del serbatoio interno. La smaltatura è un passaggio cruciale, poiché protegge il metallo dalla corrosione e prolunga la vita utile del boiler. Proprio per questo passaggio è stato richiesto alla società Meta S.r.l., dove è stato portato avanti il lavoro di tesi, un sistema automatico di rilevazione e diagnosi di possibili errori di smaltatura nei vari elementi prodotti quotidianamente.

Durante questa fase cruciale, si può incombere in vari difetti di produzione che possono rendere difettosi i boiler appena prodotti:

- **Assenza di smalto**: la presenza di intere aree sulla superficie interna dei boiler con assenza di smaltatura

- **Caduta di smalto**: piccoli puntini sulla superfice dove non è presente lo smalto applicato internamente

- **Basso spessore**: una smaltatura interna con uno spessore troppo fine da poter assicurare l'integrità dell'elemento analizzato

Sulla linea di produzione è stato introdotto un processo di controllo qualità basato sull'estrazioni di features provenienti da foto interne dei boiler e l'utilizzo di soglie fisse per individuare eventuali difetti di produzione di smaltatura.

2.1 Analisi visiva

Dopo la fase di smaltatura interna del boiler, sulla linea di produzione sono state inserite due sonde visive che, entrando nell'elemento in analisi, irradiano con luce blu la parte interna appena smaltata effettuando delle foto (Figura 9). Le sonde sono due poiché la prima effettua la foto della parte alta e della parte centrale, mentre la seconda effettua la foto della parte inferiore.

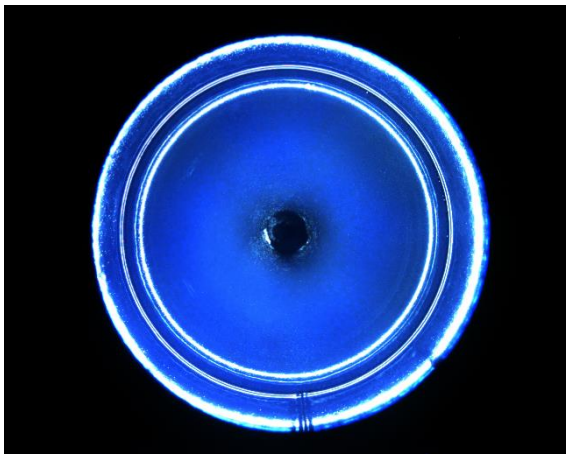


Figura 9: Foto della parte superiore del boiler

2.2 Estrazione Features e Valutazione con Soglie

Raccolte le foto dell'interno del boiler, queste vengono passate in un software, sviluppato interamente nell'ambiente di sviluppo LabView, dove si estrapolano le features numeriche dalla foto analizzata e si settano le soglie fisse scelte precedentemente al fine di verificare se ogni feature estratta rientra nel range di soglia scelto.

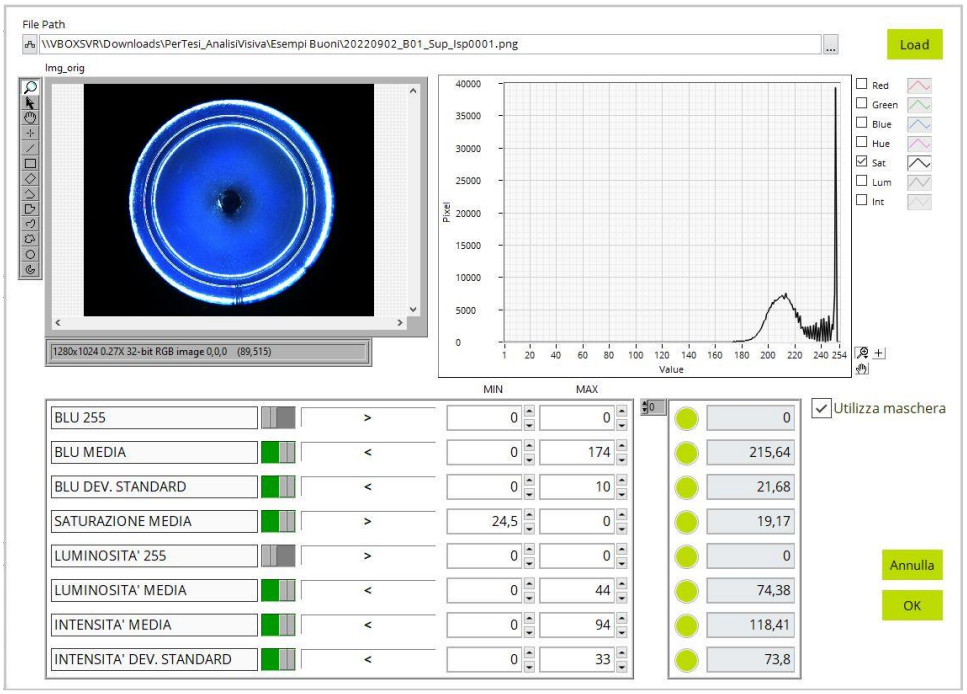


Figura 10: Analisi immagine con assenza di fault

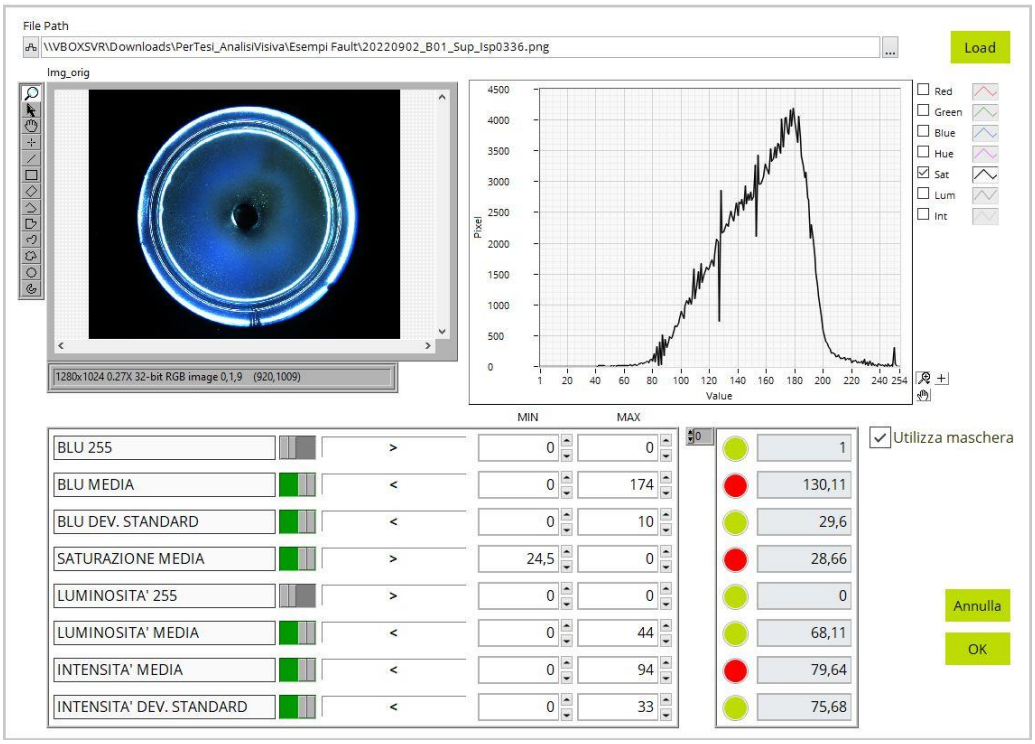


Figura 11: Analisi immagine con presenza di fault

Per ogni feature è stato predisposto un range di soglia con un valore minimo ed un valore massimo ed è stato stabilito se il valore della feature deve essere al di fuori del range scelto (utilizzando il simbolo <) oppure all'interno del range scelto (utilizzando il simbolo >).

Dalla figura 10 si può notare come ogni valore estratto rispetta le soglie prefissate mentre la figura 11 ha dei valori che non rispettano le soglie.

Anche solo una feature fuori soglia decreta il fault nel processo di smaltatura del boiler analizzato.

Questo procedimento viene fatto per ogni foto scattata dalle sonde all'interno dei boiler (Figura 12-15), ma in questo lavoro di tesi ci concentreremo sulla rilevazione di errori di smaltatura nella parte superiore e centrale dei boiler.

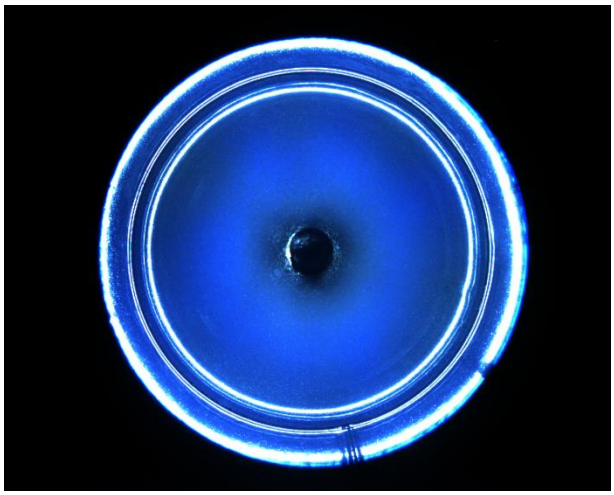


Figura 12: Parte superiore del boiler senza guasto

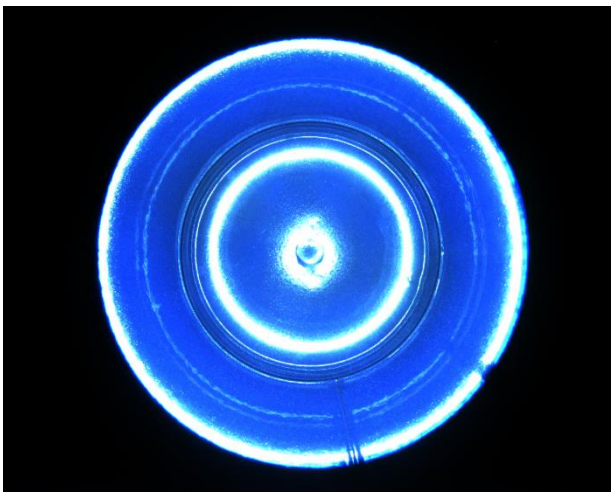


Figura 13: Parte centrale del boiler senza guasto

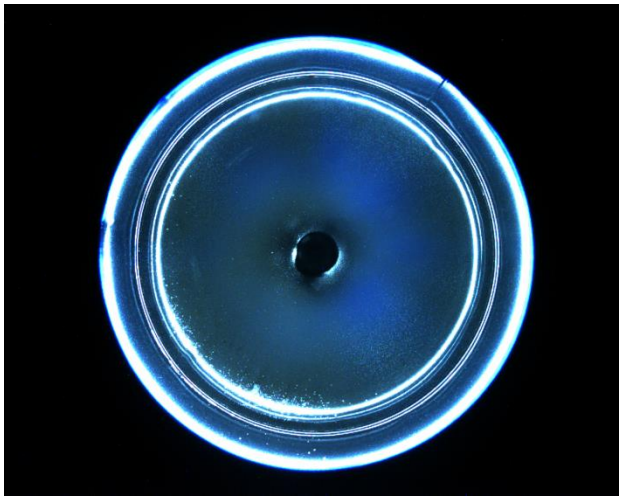


Figura 14: Parte superiore del boiler con guasto

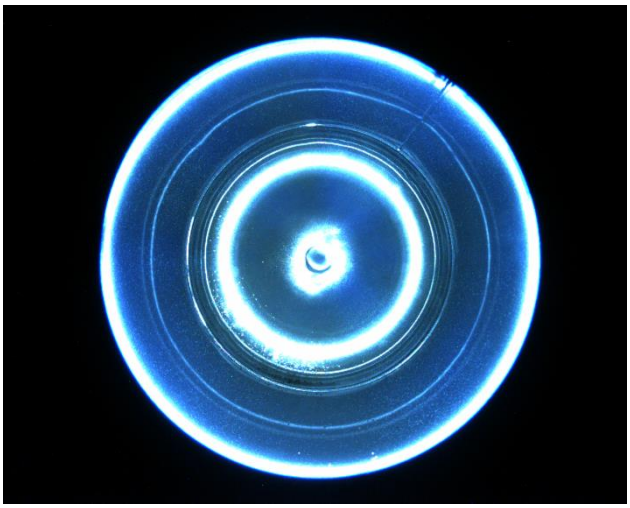


Figura 15: Parte centrale del boiler con guasto

Il lavoro di tesi è basato sulla progettazione di un modello di rilevamento delle difettosità basato su un classificatore che prenda in input le features estratte dalle foto scattate dalla sonda e dia in output l'etichetta di fault o no fault per l'elemento analizzato.

L’obiettivo è implementare un sistema automatico che vada ad eliminare l’utilizzo delle soglie fisse ed a limitare l’intervento umano nella convalida delle stesse e nell’etichettare il boiler come guasto o non guasto.

Capitolo 3

Materiali e Metodi

3.1 Dati di partenza

Lo sviluppo di un modello di classificazione ai fini della tesi parte da una raccolta di 670 file .txt dove sono riportate tutte le informazioni estrapolate dai vari elementi analizzati, tra cui anche le features estratte tramite il software sviluppato in LabView. Ogni file .txt è caratterizzato da un nome con la seguente struttura: "Log_Results_156_St3_Verticale 80 lt GIZEM.txt" in cui “156” sta ad indicare il giorno dell'anno, mentre “St3” indica che è associato alla stazione 3 (ci sono 3 stazioni di collaudo) e “Verticale 80 lt GIZEM” indica il tipo di boiler, specificando il nome, il litraggio (varia tra 50, 80 o 100 litri) ed il tipo di smalto utilizzato. All’interno dei file ci sono molteplici dati, ma ai fini del lavoro i campi di interesse sono:

- **BS_MidSup, BS_Sup**: 0 o 1, indicano la presenza o meno del difetto di Basso spessore e/o bruciatura individuato dal software di analisi rispettivamente per la parte centrale e superiore del boiler (saranno utilizzati con Label Fault durante l’addestramento del classificatore)
- **CS_Red, CS_White, CL_MidSup**: 0 o 1, indicano la presenza di caduta di smalto rispettivamente sulla parte superiore analizzando il colore

rosso o il colore bianco nella foto fatta dalla sonda e la presenza di caduta di smalto nella parte centrale del boiler.

- **ABS_Sup**: 0 o 1, indica la possibile assenza di smalto nella parte superiore del boiler.
- **BLU Media sup, BLU Media midsup:** features che analizzano la quantità media di blu presente nella foto analizzata
- **BLU 255 sup, BLU 255 midsup:** features che analizzano la quantità di blu normalizzata
- **ROSSO Media sup, ROSSO Media midsup:** features che analizzano la quantità media di rosso
- **VERDE Media sup:** feature che analizza la quantità media di verde (unica feature estrapolata solo dalle foto della parte superiore)
- **SATURAZIONE Media sup, SATURAZIONE Media midsup:** features che analizza la saturazione media di colore
- **LUMINOSITÀ Media sup, LUMINOSITÀ Media midsup:** features che analizza la luminosità media di colore
- **LUMINOSITÀ 255 sup, LUMINOSITÀ 255 midsup:** features che analizza la luminosità media di colore normalizzata
- **INTENSITÀ Media sup, INTENSITÀ Media midsup:** features che analizza l'intensità media di colore

Le quantità medie dei vari colori RGB sono state calcolate estraendo per ogni colore il proprio piano colore, applicando una maschera per selezionare solo la zona centrale dell'immagine e calcolando la media sommando i valori dei

pixel all'interno della zona centrale e dividi per il numero totale di pixel in quella zona:

$$Media = \frac{\sum Valori\ dei\ pixel}{Numero\ di\ pixel}$$

Per quanto riguarda la quantità media di blu normalizzata, il calcolo ulteriore fatto è stato di normalizzare la media dividendola per 255 (massimo valore che può avere un pixel):

$$Media\ Normalizzata = \frac{Media\ dei\ valori\ dei\ pixel}{255}$$

Come primo task, è stata creata un'unica query tramite il programma Excel dove sono stati inseriti tutti i dati presenti nei 670 file.txt (Figura 16).

BLU 255.txt	BLU MEDIA.txt	ROSSO MEDIA.txt	SATURAZIONE MEDIA.txt	LUMINOSITA' 255.txt	LUMINOSITA' MEDIA.txt	INTENSITA' MEDIA.txt	VERDE MEDIA.txt	BLU 255 midpoint	BLU MEDIA midpoint	ROSSO MEDIA midpoint	SATURAZIONE MEDIA midpoint	LUMINOSITA 255 midpoint
4,000000	242,8	34,100000	17,200000	0,000000	84,300000	138,200000	79,100000	907,000000	245,700000	27,400000	15,300000	82,000000
8,000000	244,6	26,900000	10,800000	0,000000	75,000000	135,500000	66,600000	381,000000	248,100000	26,400000	11,400000	34,000000
3,000000	248,7	27,600000	9,100000	0,000000	76,700000	137,900000	68,400000	606,000000	249,900000	24,800000	9,700000	43,000000
2,000000	250,7	29,100000	8,200000	0,000000	78,900000	139,700000	70,800000	814,000000	251,000000	27,300000	8,600000	99,000000
4,000000	243,2	31,500000	17,200000	0,000000	81,000000	137,100000	74,700000	723,000000	247,500000	27,500000	13,700000	38,000000
74,000000	245,9	27,300000	14,400000	5,000000	75,600000	136,300000	67,200000	479,000000	248,900000	28,200000	12,000000	7,000000
0,000000	249,2	25,900000	10,500000	0,000000	73,800000	137,300000	64,100000	380,000000	251,000000	26,000000	8,000000	9,000000
0,000000	243,7	27,500000	16,100000	0,000000	75,600000	135,300000	67,600000	327,000000	245,600000	28,200000	15,700000	53,000000
6,000000	213,7	21,500000	22,200000	0,000000	61,300000	117,300000	51,900000	202,000000	188,300000	18,200000	17,600000	10,000000
0,000000	189,1	12,100000	11,600000	0,000000	45,500000	100,300000	34,600000	140,000000	207,100000	14,700000	15,800000	11,000000
2,000000	226,7	17,300000	11,200000	0,000000	58,200000	121,700000	46,400000	315,000000	210,500000	15,700000	14,800000	23,000000
21,000000	212,8	16,300000	19,400000	3,000000	54,800000	114,300000	43,600000	304,000000	208,500000	17,900000	19,400000	14,000000
0,000000	181,8	11,700000	9,600000	0,000000	43,900000	96,500000	33,500000	52,000000	205,400000	14,200000	14,600000	1,000000
4,000000	220,6	18,600000	18,900000	0,000000	58,800000	119,400000	47,900000	185,000000	221,100000	19,700000	18,500000	22,000000
0,000000	223,3	18,500000	18,400000	0,000000	59,000000	120,700000	47,700000	562,000000	205,700000	18,900000	18,800000	47,000000
0,000000	206,3	15,500000	15,200000	0,000000	52,200000	110,600000	41,000000	130,000000	220,200000	16,600000	15,400000	3,000000
0,000000	217,5	22,200000	21,900000	0,000000	62,900000	119,600000	53,600000	134,000000	215,000000	19,900000	18,000000	5,000000
0,000000	186	14,000000	14,800000	0,000000	47,400000	99,800000	37,500000	101,000000	199,800000	18,200000	17,900000	3,000000
108,000000	241,5	23,800000	15,800000	12,000000	69,600000	132,400000	59,600000	275,000000	238,900000	24,400000	18,000000	6,000000
2,000000	247,5	25,800000	10,700000	0,000000	73,900000	136,400000	64,700000	708,000000	248,900000	25,400000	10,200000	49,000000
7,000000	243	25,100000	15,600000	0,000000	69,900000	133,800000	59,000000	259,000000	249,100000	24,300000	10,900000	20,000000
3,000000	246,5	28,700000	11,100000	0,000000	77,600000	137,300000	69,700000	613,000000	249,800000	29,600000	10,500000	19,000000
1,000000	249,8	29,400000	9,000000	0,000000	79,500000	139,400000	72,000000	727,000000	250,600000	27,100000	8,600000	27,000000
6,000000	248,6	29,400000	10,500000	0,000000	79,400000	138,700000	72,000000	405,000000	250,900000	27,800000	8,600000	17,000000
1,000000	251,3	27,900000	7,300000	0,000000	74,700000	139,400000	64,200000	522,000000	249,800000	28,500000	11,400000	37,000000
15,000000	250	38,000000	11,800000	0,000000	88,600000	143,700000	83,000000	1059,000000	250,300000	37,700000	12,100000	23,000000
203,000000	248,3	36,100000	15,500000	7,000000	83,500000	141,900000	75,600000	730,000000	246,800000	38,000000	17,800000	8,000000
44,000000	251,1	32,200000	10,700000	1,000000	80,900000	141,400000	72,700000	264,000000	251,000000	36,200000	9,900000	4,000000
0,000000	252,7	33,000000	5,600000	0,000000	81,100000	142,600000	74,000000	773,000000	252,000000	34,900000	7,100000	37,000000
0,000000	252	31,900000	6,700000	0,000000	80,100000	141,700000	71,300000	380,000000	252,700000	34,700000	4,600000	11,000000
6,000000	252,8	34,500000	2,500000	0,000000	85,200000	143,400000	78,400000	884,000000	252,100000	34,500000	6,300000	79,000000
1,000000	251,7	33,600000	8,100000	0,000000	83,300000	142,400000	76,000000	501,000000	251,000000	35,600000	10,100000	23,000000
40,000000	247,7	38,500000	16,000000	2,000000	88,800000	142,900000	83,600000	628,000000	250,000000	35,000000	11,900000	46,000000
3,000000	251,3	30,800000	7,800000	0,000000	78,800000	140,800000	69,800000	730,000000	251,700000	34,400000	7,400000	28,000000
0,000000	252,5	33,800000	4,000000	0,000000	84,200000	142,900000	77,200000	1188,000000	251,900000	34,800000	7,200000	61,000000
49,000000	251,2	31,700000	10,300000	1,000000	80,800000	141,200000	72,700000	275,000000	250,500000	34,200000	10,800000	7,000000
15,000000	248	37,400000	15,600000	0,000000	88,000000	142,600000	82,500000	1002,000000	249,700000	34,700000	12,300000	90,000000

Figura 16: File Excel con tutti i dati importati

Importando tutti i dati, abbiamo creato un unico file in cui sono presenti tutti le statistiche utili al fine di addestrare un classificatore, sfruttando la comodità di avere tutti in un solo file .xlmx anziché in 670 file .txt migliorando anche la facilità di reperire informazioni qualora non fossero state utilizzate durante il lavoro di creazione e addestramento del modello.

Ai fini dell'addestramento del modello, in questo file troviamo BS_Sup e BS_MidSup usate come label target binarie e tutte le feature introdotte precedentemente utilizzate come predittori di un possibile difetto di produzione.

Prima di iniziare il lavoro di addestramento, è bene considerare che le fault label usate come target di difetti di produzione sono suscettibili di incertezza poiché il dataset è stato etichettato con le precedenti logiche ed un minimo di supporto umano per la validazione dell'etichetta. Ciò implica che tutto il lavoro potrebbe avere dei problemi impliciti dovuti al dataset etichettato.

3.2 Pre-Elaborazione e Bilanciamento

3.2.1 Ranking Features

Passo fondamentale per la creazione del modello di rilevamento delle difettosità è il pre-processamento dei dati grezzi per renderli più puliti possibili ed evitare eventuali rumori nel processo di addestramento.

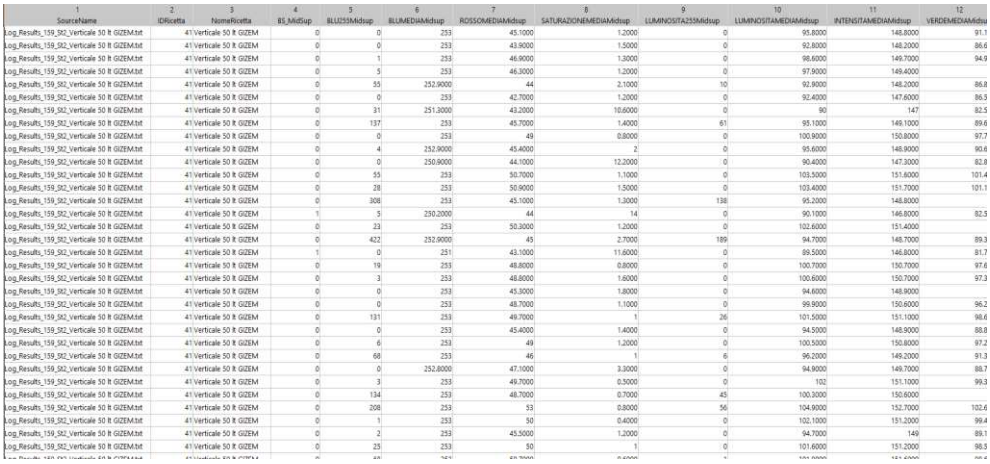
Prima operazione effettuata è stata l'importazione dell'intero dataset nell'ambiente Matlab, dividendolo in due dataset:

1. Primo Dataset: comprende tutti i dati, la fault label e le features inerenti alle foto della parte superiore (Figura 17)

1 SourceName	2 IDRicetta	3 NomeRicetta	4 BS_Sup	5 BLU255Sup	6 BLUMEDIASup	7 ROSSOMEDIASup	8 SATURAZIONEMEDIASup	9 LUMINOSITA255Sup	10 LUMINOSITAMEDIASup	11 INTENSITAMEDIASup	12 VERDEMEDIAS
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	35.5000	0.9000	0	82.8000	144	73.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	40.3000	0.5000	0	89.1000	146.4000	82.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	252.9000	36.9000	2.2000	0	85.2000	144.7000	77.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	252.9000	36	1.9000	0	83.2000	144.2000	74.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	42	0.4000	0	91.6000	147.3000	85.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	44.5000	0.3000	0	94.6000	148.5000	89.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	37.8000	1.1000	0	85.8000	145.1000	77.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	36.8000	1.1000	0	84.3000	144.6000	75.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	2.0	253	37.7000	0.5000	0	85.6000	145.1000	77.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	1.0	253	41.4000	0.4000	0	90.9000	147	84.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	19.0	253	47.6000	1.4000	0	97.9000	150	93.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	1.0	252.5000	33.8000	4.7000	0	80.2000	142.9000	70.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	12.0	253	49.1000	0.3000	0	100.2000	150.8000	96.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	45.2000	0.3000	0	95.4000	148.8000	90.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	1.0	252.7000	34.8000	3.6000	0	81.6000	143.5000	72.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	1.0	253	46.9000	0.2000	0	97.8000	149.7000	93.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	2.0	253	44.2000	0.3000	0	94.7000	148.4000	89.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	1.0	253	38.5000	0.6000	0	86.5000	145.5000	78.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	8.0	252.7000	44.3000	4	0	94.1000	148.2000	88.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	43.8000	0.3000	0	94	148.1000	88.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	4.0	253	44.9000	0.3000	0	95.5000	148.7000	90.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	44.6000	0.4000	0	94.6000	148.5000	89.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	252.9000	37.6000	1.8000	0	85.8000	145	77.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	252.9000	37.7000	2.3000	0	85.1000	145	76.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	3.0	253	45.4000	0.3000	0	96.1000	148.9000	91.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	5.0	253	46.6000	0.3000	0	97.2000	149.6000	92.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	253	41.6000	0.5000	0	90.8000	147	84.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	0.0	252.9000	38.5000	2.2000	0	86.2000	145.4000	
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	5.0	253	50.6000	0.3000	0	102.3000	151.6000	99.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	<undefined>	NaN	NaN	NaN	NaN	NaN	NaN	
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	75.0	253	51.4000	0.4000	0	103.1000	151.9000	100.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	4.0	253	50	0.3000	0	102	151.3000	99.
Log_Results_150_St2_Verticale 80 lt GIZEM.txt	43	Verticale 80 lt GIZEM	0	14.0	253	47.3000	0.3000	0	97.4000	149.9000	92.

Figura 17: Dataset Matlab BS_Sup

2. Secondo Dataset: comprende tutti i dati, la fault label e le features inerenti alle foto della parte centrale (Figura 18)

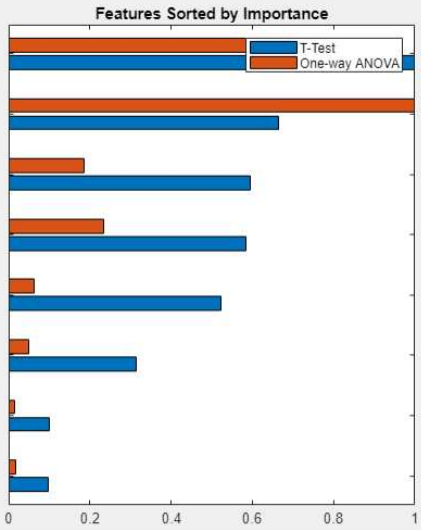


1 SourceName	2 IDRicetta	3 NomeRicetta	4 BS_MidSup	5 BLU255Midsup	6 BLUMEDIAMidsup	7 ROSSOMEDIAMidsup	8 SATURAZIONEMEDIAMidsup	9 LUMINOSITA255Midsup	10 LUMINOSITAMEDIAMidsup	11 INTENSITAMEDIAMidsup	12 VERDEMEDIAMidsu
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	45.1000	1.2000	0	95.8000	148.8000	91.1
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	43.9000	1.5000	0	92.8000	148.2000	86.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	1	253	46.9000	1.3000	0	98.6000	149.7000	94.9
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	5	253	46.3000	1.2000	0	97.9000	149.4000	
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	55	252.9000	44	2.1000	10	92.9000	148.2000	86.8
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	42.7000	1.2000	0	92.4000	147.6000	86.3
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	31	251.3000	43.2000	10.6000	0	90	147	82.5
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	137	253	45.7000	1.4000	61	95.1000	149.1000	89.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	49	0.8000	0	100.9000	150.8000	97.7
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	4	252.9000	45.4000	2	0	95.6000	148.9000	90.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	250.9000	44.1000	12.2000	0	90.4000	147.3000	82.8
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	55	253	50.7000	1.1000	0	103.5000	151.6000	101.4
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	28	253	50.9000	1.5000	0	103.4000	151.7000	101.1
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	308	253	45.1000	1.3000	138	95.2000	148.8000	
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	1	5	250.2000	44	14	0	90.1000	146.8000	82.5
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	23	253	50.3000	1.2000	0	102.6000	151.4000	
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	422	252.9000	45	2.7000	189	94.7000	148.7000	89.3
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	1	0	251	43.1000	11.6000	0	89.5000	146.8000	81.7
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	19	253	48.8000	0.8000	0	100.7000	150.7000	97.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	3	253	48.8000	1.6000	0	100.6000	150.7000	97.3
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	45.3000	1.8000	0	94.6000	148.9000	
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	48.7000	1.1000	0	99.9000	150.6000	96.2
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	131	253	49.7000	1	26	101.5000	151.1000	98.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	253	45.4000	1.4000	0	94.5000	148.9000	88.8
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	6	253	49	1.2000	0	100.5000	150.8000	97.2
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	68	253	46	1	6	96.2000	149.2000	91.3
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	0	252.8000	47.1000	3.3000	0	94.9000	149.7000	88.7
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	3	253	49.7000	0.5000	0	102	151.1000	99.3
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	134	253	48.7000	0.7000	45	100.3000	150.6000	
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	208	253	53	0.8000	56	104.9000	152.7000	102.6
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	1	253	50	0.4000	0	102.1000	151.2000	99.4
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	2	253	45.5000	1.2000	0	94.7000	149	89.1
Log_Results_159_S02_Verticale 50 lt GIZEM.txt	41	Verticale 50 lt GIZEM	0	25	253	50	1	0	101.6000	151.2000	98.3

Figura 18: Dataset Matlab BS_MidSup

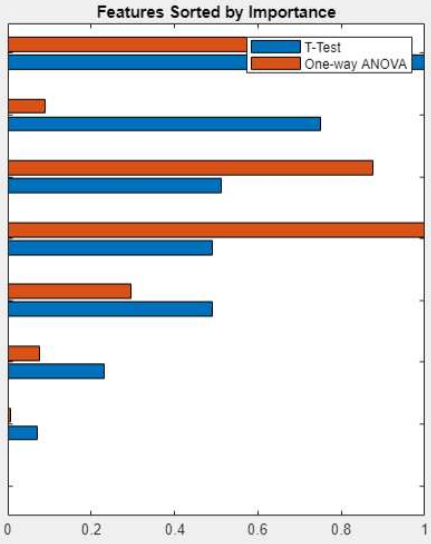
Questi due dataset sono stati importati in un App di Matlab, applicazioni interattive scritte per eseguire task di calcolo, chiamata Diagnostic Features Designer (DFD) nel quale è stato fatto un lavoro di ranking delle features: di norma tutti i dati sono utili ad estrapolare informazioni valide per capire e diagnosticare un guasto, però può capitare che la natura di un'informazione sia molto diversa dalle altre e ciò fa si che quel dato possa essere visto come inutile ai fini diagnostici o addirittura dannoso tanto da comportarsi come rumore. Creare un ranking tra i vari dati serve a determinare l'importanza relativa delle variabili presenti in un dataset rispetto a un obiettivo predittivo (Vettoretti, Di Camillo,2021). Questa tecnica serve a migliorare l'efficacia e l'efficienza dei processi di modellazione riducendo, dove necessario, la dimensione del dataset e migliorando la precisione del modello.

Grazie al DFD è stato possibile classificare le features di ogni dataset grazie a due metodi statistici: T-Test e One-way Anova.



Feature	T-Test	One-way ANOVA
SATURAZIONEME...	92.9334	2.9642e+04
INTENSITAMEDIA...	61.8621	4.1216e+04
VERDEMEDIAMid...	55.3081	7.7125e+03
LUMINOSITAMEDI...	54.2791	9.6009e+03
BLUMEDIAMidsup	48.4954	2.5665e+03
ROSSOMEDIAMid...	29.1539	1.9980e+03
BLU255Midsup	9.2132	614.8813
LUMINOSITA255M...	9.0702	727.8003

Figura 19: Ranking features BS_Sup



Feature	T-Test	One-way ANOVA
BLUMEDIASup	93.0550	1.0869e+05
SATURAZIONE...	69.8687	1.1247e+04
BLU255Sup	47.6851	1.0965e+05
LUMINOSITA25...	45.6666	1.2514e+05
INTENSITAME...	45.6574	3.6941e+04
ROSSOMEDIA...	21.5709	9.6203e+03
LUMINOSITAM...	6.5898	689.1984
VERDEMEDIA...	0.2145	0.6357

Figura 20: Ranking features BS_MidSup

Dalla Figura 19 e Figura 20 si può notare come tra le features della parte superiore sono state scartate le features LUMINOSITÀ MEDIA Sup e VERDE MEDIA Sup mentre tra le features della parte centrale sono state scartate le features BLU 255 MidSup e LUMINOSITÀ 255 MidSup.

3.2.2 Features Balanced

Individuati i dati non utili ai fini della modellazione ed addestramento, è stato utilizzato il linguaggio di programmazione Python nell'ambiente di sviluppo Visual Studio Code per importare il dataset creando due data frame grezzi, il primo con le features ed il label target utili all'addestramento di un classificatore per i difetti di fabbrica della parte superiore ed il secondo con le features ed il label target utili all'addestramento di un classificatore per i difetti di fabbrica della parte centrale :

- un data frame contiene i predittori BLU MEDIA sup, BLU 255 sup, LUMINOSITÀ 255 sup, INTENSITÀ MEDIA sup, SATURAZIONE MEDIA sup, ROSSO MEDIA sup con la label target BS_Sup e l'aggiunta delle fault label CS_Red, CS_White e ABS_Sup

- un data frame contiene i predittori BLU MEDIA midsup, VERDE MEDIA midsup, LUMINOSITÀ MEDIA midsup, INTENSITÀ MEDIA midsup, SATURAZIONE MEDIA midsup, ROSSO MEDIA midsup con la label target BS_MidSup e l'aggiunta delle fault label CL_MidSup e ABS_Sup

```
import numpy as np
df= df[df['CS_Red'] != 1]
df= df[df['ABS_Sup'] != 1]
df= df[df['CS_White'] != 1]
X = df[["BLU MEDIA sup","SATURAZIONE MEDIA sup","BLU 255 sup","LUMINOSITA' 255 sup","INTENSITA' MEDIA sup","ROSSO MEDIA sup"]]
Y = df['BS_Sup']
X=X.replace(['NO','SI'],np.nan)
X=X.fillna(0)
```

Figura 21: Codice ETL BS_Sup

```
import numpy as np
df= df[df['ABS_Sup'] != 1]
df= df[df['CL_MidSup'] != 1]
X = df[["BLU MEDIA midsup", "ROSSO MEDIA midsup", "SATURAZIONE MEDIA midsup",
        "LUMINOSITA MEDIA midsup", "INTENSITA MEDIA midsup",
       "VERDE MEDIA midsup"]]
Y = df['BS_MidSup']
X=X.replace(['NO','SI'],np.nan)
X=X.fillna(0)
```

Figura 22: Codice ETL BS_MidSup

Come prima operazione di ETL (Extract-Transform-Load) è stato effettuato un lavoro di bilanciamento dei dati in base alle label target (Figura 21,22) poiché c'è uno sbilanciamento dei dati etichettati come 0 (senza difetto) rispetto ai dati etichettati come 1 (con difetto).

Ogni dataset è stato modificato nel modo seguente:

- per il primo dataset sono state eliminate tutte le righe dove era presente un valore binario pari ad 1 di ABS_Sup, CS_Red o CS_White poiché i valori dei predittori possono essere influenzati dalla presenza di questi difetti che non sono presi in considerazione nell'addestramento del classificatore, cosi da avere un dataset influenzato solo dalla label target BS_Sup. Inoltre sono stati trasformati tutti gli elementi NaN o alfabetici in valori numerici 0.

- per il secondo dataset sono state effettuate le stesse modifiche, ma in questo caso sono state eliminate tutte le righe dove era presente un valore binario pari ad 1 di ABS_Sup o CL_MidSup.

Processati i due dataset, è stato attuato un bilanciamento (Figura 23,24) in favore della fault label meno numerosa usando una tecnica di over sampling chiamata SMOTE (Chawla, Bowyer, Hall, Kegelmeyer, 2002): i dati sintetici vengono generati a partire da osservazioni reali della classe minoritaria. Per fare questo, si seleziona un'osservazione reale e si calcolano le distanze euclidee tra l'osservazione e i suoi k vicini più vicini. I dati sintetici vengono quindi generati in punti casuali all'interno della regione di spazio che circonda l'osservazione selezionata.

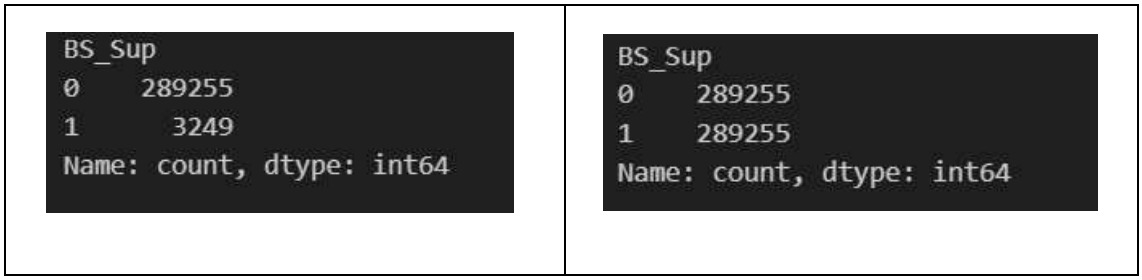


Figura 23: BS_Sup Pre e Post SMOTE Over sampling

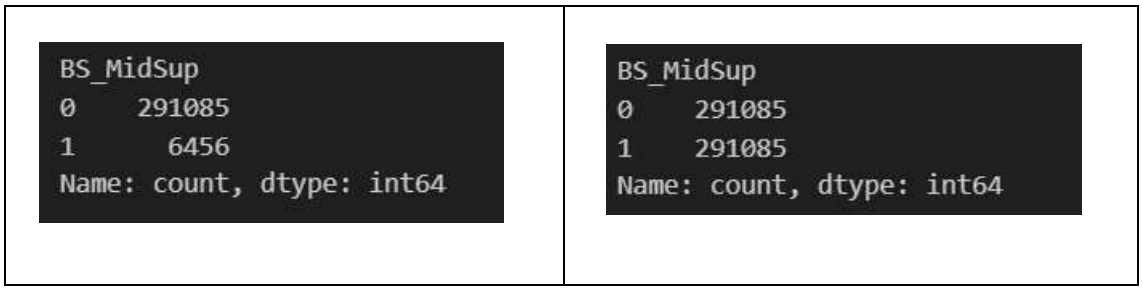


Figura 24: BS_MidSup Pre e Post SMOTE Over sampling

3.3 Creazione ed Addestramento Modello

Salvati i nuovi dataset bilanciati, questi sono stati utilizzati per la modellazione di due classificatori binario basati su rete neurale rispettivamente per diagnosticare i fault BS_Sup e BS_MidSup.

3.3.1 Modello di Rete Neurale

```
# Dividere i dati
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,stratify=y)

# Creare il modello usando l'API funzionale
input_layer = Input(shape=(X_train.shape[1],))
x = Dense(128, activation='relu')(input_layer)
x = BatchNormalization()(x)
x = Dropout(0.4)(x)
x = Dense(64, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.4)(x)
x = Dense(32, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.4)(x)
output_layer = Dense(1, activation='sigmoid')(x)

model_BS = Model(inputs=input_layer, outputs=output_layer)

# Compilare il modello
model_BS.compile(optimizer='adam',
                 loss='binary_crossentropy',
                 metrics=['accuracy'])
```

Figura 25: Codice Modello Rete Neurale BS_Sup

Dapprima i dati ed i rispettivi array dei fault label sono stati divisi in percentuale 80%-20% rispettivamente in dataset di training e di testing, bilanciando ogni dataset grazie ad una stratificazione in base all'array dei fault label (Figura 25).

Dopodiché è stato implementato un classificatore mediante un modello di rete neurale grazie all'utilizzo di Tensorflow, libreria open-source completa

per il machine learning e il deep learning, utilizzata per costruire, addestrare e distribuire modelli di apprendimento automatico.

Tramite questa libreria si può sfruttare l'API di Keras, un'interfaccia di programmazione di alto livello che semplifica la costruzione e l'addestramento di modelli di deep learning. Originariamente sviluppata come libreria indipendente, Keras è stata successivamente integrata in TensorFlow, fornendo una API intuitiva e potente per costruire modelli di machine learning e deep learning.

Keras fornisce due modi principali per costruire modelli di deep learning:

- **Modello Sequenziale** (Sequential): utilizzato per creare modelli layer-wise, dove i layer sono impilati uno sopra l'altro in modo lineare.

- **API Funzionale** (Functional): permette di costruire modelli complessi con grafi computazionali non lineari, inclusi modelli con strati condivisi, ingressi multipli e uscite multiple.

Per questo lavoro è stato scelto di costruire un modello di rete neurale tramite API funzionale definendo:

1. **Input Layer**: primo strato di una rete neurale che accetta i dati di input e definisce la forma dei dati che verranno utilizzati dal modello.

2. **Hidden Layers**: strati nascosti della rete neurale di cui fanno parte i Dense, strati della rete dove ogni neurone è connesso a tutti neuroni

dello strato precedente, i BatchNormalization, strati dove vengono normalizzati gli input di un layer per migliorare la stabilità e la velocità di addestramento, e i Dropout, strati di regolazione dove si riduce l'over fitting nelle reti neurali, disattivando casualmente una frazione di neuroni nel layer corrente (nel nostro caso 0.4 ossia viene disattivato il 40% dei neuroni)

3. **Output Layer**: l'ultimo strato della rete neurale che fornisce le previsioni finali del modello.

Alla fine, il modello viene creato specificando input ed output e viene compilato con ottimizzatore, funzione di perdita e metrica.

3.3.2 Addestramento

Creato il modello di rete neurale, viene addestrato definendo (Figura 26):

```
# Configurare i callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.001)

# Allenare il modello
history = model_BS.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2,
                       callbacks=[early_stopping, reduce_lr])
```

Figura 26: Addestramento Modello BS_Sup

- **Epochs**: numero massimo di epoche, un ciclo completo di addestramento

- **Batch_size**: numero di campioni di dati utilizzati per addestrare il modello in una singola iterazione, utilizzato per aggiornare i pesi del modello.

- **Validation_split**: percentuale dei dati di addestramento che viene utilizzato per la validazione.

Inoltre, sono stati definiti dei callback, funzioni che vengono chiamate automaticamente durante il processo per ottimizzare l'addestramento:

- **EarlyStopping**: interrompe l'addestramento se la perdita di validazione non migliora per 10 epoche consecutive e ripristina i pesi migliori.

- **ReduceLROnPlateau**: riduce il learning rate di un fattore di 0.2 se la perdita di validazione non migliora per 5 epoche consecutive, con un minimo di 0.001

3.4 Valutazione del Modello

Addestrato il modello di rete neurale, è stato sottoposto a testing e sono state calcolate delle metriche per la valutazione delle prestazioni (Zheng, 2015) partendo dalla matrice di confusione, una tabella che mostra i risultati delle previsioni del modello rispetto alle classi effettive, evidenziando i True Positive (tp), True Negative (tn), False Positive (fp), False Negative (fn).
Con questi elementi sono state calcolate le seguenti metriche (Figura 27):

- **Accuracy**: misura la percentuale di previsioni corrette rispetto al numero totale di campioni.

- **Precision**: misura la percentuale di previsioni positive corrette rispetto al numero totale di previsioni positive fatte dal modello.

- **Recall**: misura la percentuale di campioni positivi correttamente identificati dal modello rispetto al numero totale di campioni positivi nella popolazione.

- **F1-score**: è una media armonica tra la precision e la recall. È utile quando si desidera trovare un equilibrio tra precisione e recall.

- **Specificity**: percentuale di veri negativi identificati sul totale delle istanze negative effettive.

- **False_Positive_Rate**: percentuale di falsi positivi sul totale delle istanze negative effettive.

- **False_Negative_Rate**: percentuale di falsi negativi sul totale delle istanze positive effettive.

- **Receiver Operating Characteristic** (curva ROC): esprime il numero dei veri positivi in funzione dei falsi positivi, al variare di un parametro del classificatore. Il valore associato alla curva ROC è **l'Area Under the Receiver Operating Characteristic curve** (AUC-ROC) che rappresenta l'area sottesa alla curva ROC e fornisce una misura complessiva delle prestazioni del modello.

```python
from sklearn.metrics import accuracy_score, log_loss, confusion_matrix, classification_report, roc_curve, auc
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns

# Prevedi i fault sui dati di test
predictions = model_BS.predict(X_test)
predictions = (predictions > 0.5).astype(int)  # Converti le probabilità in etichette binarie

# Calcolare e visualizzare la matrice di confusione
cm = confusion_matrix(y_test, predictions)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap=plt.cm.Blues)
plt.show()

# Estrai i dati dalla matrice di confusione
tn, fp, fn, tp = cm.ravel()

# Calcolo manuale delle metriche
accuracy = (tp + tn) / (tp + tn + fp + fn)
precision = tp / (tp + fp) if (tp + fp) != 0 else 0
recall = tp / (tp + fn) if (tp + fn) != 0 else 0
f1 = 2 * precision * recall / (precision + recall) if (precision + recall) != 0 else 0
specificity = tn / (tn + fp) if (tn + fp) != 0 else 0
false_positive_rate = fp / (fp + tn) if (fp + tn) != 0 else 0
false_negative_rate = fn / (fn + tp) if (fn + tp) != 0 else 0

fpr, tpr, thresholds = roc_curve(y_test, predictions)
roc_auc = auc(fpr, tpr)
```

Figura 27: Metriche di valutazione

3.5 Database NoSQL e Testing

Per completare il lavoro svolto, è stato deciso di integrare un database NoSQL (Figura 28) dove sono stati caricati tutti i dati presenti nei 670 file .txt così da avere un enorme database da cui estrapolare le informazioni utili con più facilità e dove poter inserire rilevazioni future di features così da poter essere analizzate dal classificatore addestrato.

Un database non relazionale (NoSQL) è un tipo di sistema di gestione di database che fornisce un meccanismo per l'archiviazione e il recupero dei dati che viene modellato in modi diversi rispetto ai database relazionali tradizionali (SQL). NoSQL è stato progettato per affrontare le limitazioni dei database relazionali in termini di scalabilità, prestazioni, flessibilità e capacità di gestire grandi volumi di dati non strutturati.

Base

	_id ObjectId	Source Object	Time String	Ispezione Int32	ID Ricetta Int32	Nome Ricetta
1	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230518_144522"	0	41	"Verticale 50
2	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230518_144535"	0	41	"Verticale 50
3	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112803"	12901242	44	"Verticale 100
4	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112803"	12901241	44	"Verticale 100
5	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112840"	12901251	44	"Verticale 100
6	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112840"	12901252	44	"Verticale 100
7	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112916"	12901262	44	"Verticale 100
8	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_112916"	12901261	44	"Verticale 100
9	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113001"	12901272	44	"Verticale 100
10	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113001"	12901271	44	"Verticale 100
11	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113042"	12901282	44	"Verticale 100
12	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113042"	12901281	44	"Verticale 100
13	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113124"	12901292	44	"Verticale 100
14	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113124"	12901291	44	"Verticale 100
15	ObjectId('6654c3f52cca3e3…	{} 1 fields	"20230509_113212"	12901302	44	"Verticale 100

Figura 28: Database NoSQL

Grazie all'utilizzo di MongoDB, un database NoSQL che archivia i dati in formato BSON (una rappresentazione binaria di JSON) consentendo di memorizzare documenti complessi con strutture annidate, è stata creata un'unica collection con tutti i dati inerenti tutti i boiler controllati e dai quali sono state estratti features visive ed eventuali fault di progettazione.

```
# Numero di documenti casuali da ottenere
num_docs_to_sample = 1000

# Selezionare documenti casuali con specifici campi
pipeline = [
    {"$sample": {"size": num_docs_to_sample}},  # Seleziona documenti casuali
    {"$project": {                               # Specifica i campi da includere
        "VERDE MEDIA midsup": 1,
        "BLU MEDIA midsup": 1,
        "ROSSO MEDIA midsup": 1,
        "SATURAZIONE MEDIA midsup": 1,
        "LUMINOSITA MEDIA midsup": 1,
        "INTENSITA MEDIA midsup": 1,
        "BS_MidSup": 1,
        "_id": 0  # Esclude il campo _id
        }
    }
    ]

# Eseguire l'aggregazione
random_documents = list(collection.aggregate(pipeline))

# Convertire i documenti in un DataFrame pandas
df = pd.DataFrame(random_documents)
df=df.replace(['NO','SI','#NUM!'],np.nan)
df=df.fillna(0)
df = df.replace({',': '.'}, regex=True)
df = df.astype(float)

X = df.drop("BS_MidSup", axis=1).values
y = df["BS_MidSup"].values

predictions = loaded_model.predict(X)
predictions = (predictions > 0.5).astype(int)  # Converti le probabilità in etichette binarie
```

Figura 29: Testing Classificatore BS_Sup tramite MongoDB

```python
# Numero di documenti casuali da ottenere
num_docs_to_sample = 1000

# Selezionare documenti casuali con specifici campi
pipeline = [
    {"$sample": {"size": num_docs_to_sample}},  # Seleziona documenti casuali
    {"$project": {                              # Specifica i campi da includere
        "BLU 255 sup": 1,
        "BLU MEDIA sup": 1,
        "ROSSO MEDIA sup": 1,
        "SATURAZIONE MEDIA sup": 1,
        "LUMINOSITA' 255 sup": 1,
        "INTENSITA' MEDIA sup": 1,
        "BS_Sup": 1,
        "_id": 0  # Esclude il campo _id
        }
    }
    ]

# Eseguire l'aggregazione
random_documents = list(collection.aggregate(pipeline))

# Convertire i documenti in un DataFrame pandas
df = pd.DataFrame(random_documents)
df=df.replace(['NO','SI','#NUM!'],np.nan)
df=df.fillna(0)
df = df.replace({',': '.'}, regex=True)
df = df.astype(float)

X = df.drop("BS_Sup", axis=1).values
y = df["BS_Sup"].values
predictions = loaded_model.predict(X)
predictions = (predictions > 0.5).astype(int)  # Converti le probabilità in etichette binarie
```

Figura 30: Testing Classificatore BS_MidSup tramite MondoDB

Sfruttando il database NoSQL appena introdotto, è stato effettuato un secondo lavoro di testing per valutare la bontà del sistema per il rilevamento dei difetti di produzione (Figura 29,30): sono stati estratti casualmente 1000 elementi dai quali sono stati estrapolate rispettivamente le features e la label target per il classificatore della parte superiore e per il classificatore della parte centrale.

Dopo un piccolo lavoro di elaborazione, queste features sono state sottoposte ai relativi modelli per essere classificate ed i risultati sono stati valutati tramate le metriche di valutazione elencate nella sezione precedente.

Capitolo 4

Risultati

In questo capitolo vengono illustrati i risultati dell'addestramento dei due modelli di diagnosi guasti per il basso spessore di smaltatura nella parte superiore e nella parte centrale dei boiler.

Nella prima sezione vengono riportate la matrice di confusione e le metriche calcolate dopo la fase di testing dei due classificatori.

Nella seconda sezione del capitolo vengono riportati i risultati del secondo processo di testing grazie all'utilizzo del database NoSQL, mostrando per i due modelli di diagnosi la matrice di confusione e le metriche False Positive Rate e False Negative Rate di quattro iterazioni diverse di testing.

4.1 Risultati addestramento

- Modello Classificatore BS_Sup

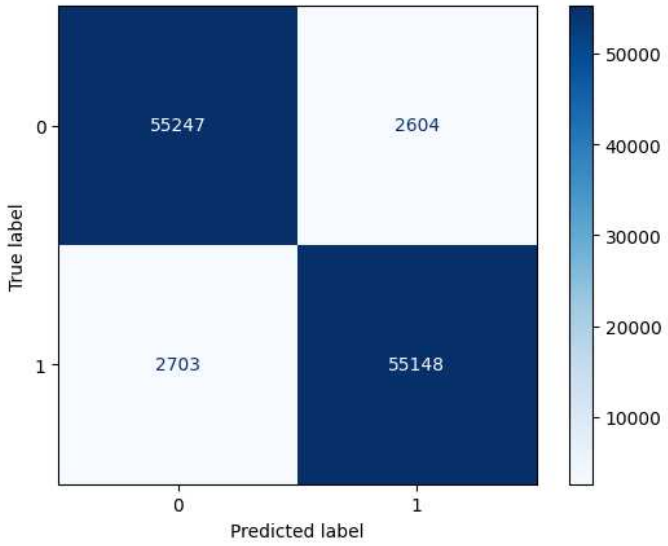


Figura 31: Confusion Matrix modello BS_Sup

```
Accuracy: 0.9541321671189781
Precision: 0.954910652444937
Recall: 0.9532765207170144
F1 Score: 0.9540928868627977
Specificity: 0.9549878135209418
False Positive Rate: 0.04501218647905827
False Negative Rate: 0.0467234792829856
```

Figura 32: Metriche modello BS_Sup

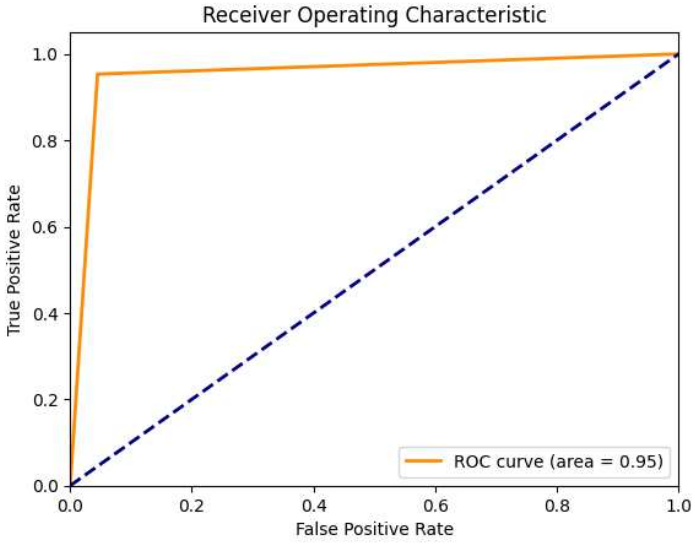


Figura 33: Curva ROC modello BS_Sup

- Modello Classificatore BS_MidSup

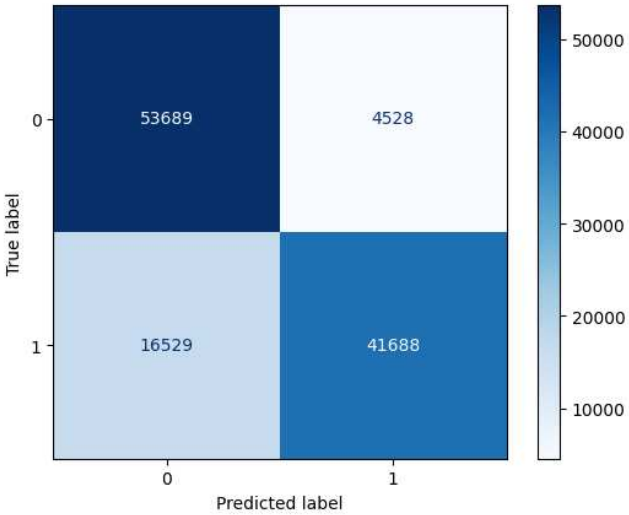


Figura 34: Confusion Matrix modello BS_MidSup

```
Accuracy: 0.8191507635226823
Precision: 0.9020252726328544
Recall: 0.716079495679956
F1 Score: 0.798368331849128
Specificity: 0.9222220313654087
False Positive Rate: 0.07777796863459127
False Negative Rate: 0.28392050432004395
```

Figura 35: Metriche modello BS_MidSup

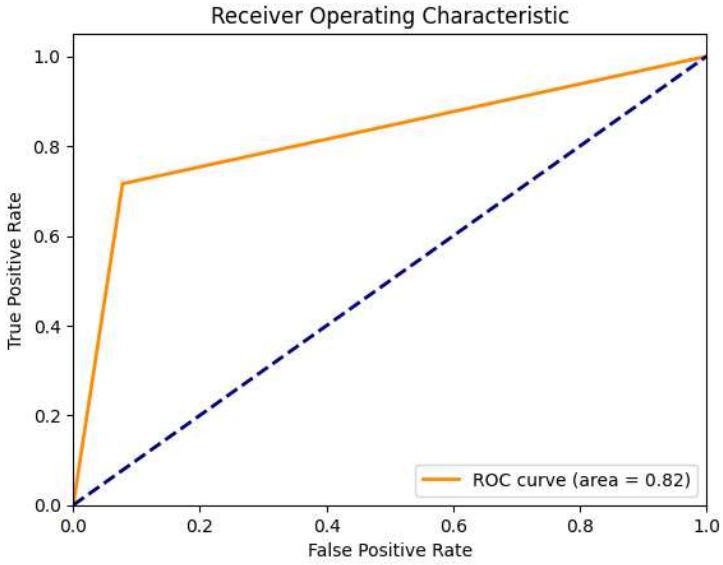


Figura 36: Curva ROC modello BS_MidSup

4.2 Risultati Testing Database NoSQL

- Modello Classificatore BS_Sup

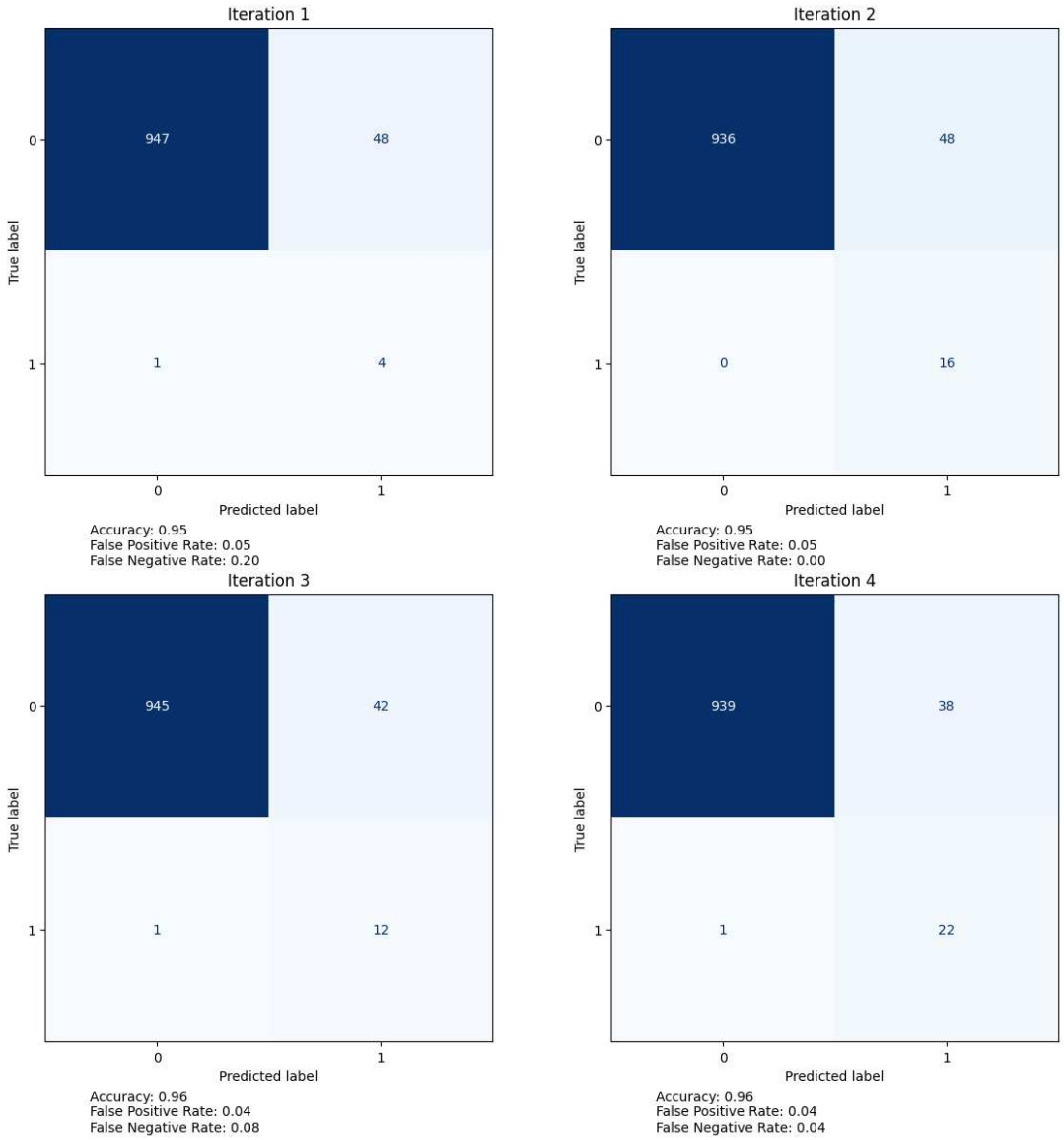


Figura 37: Risultati Testing con Database NoSQL modello BS_Sup

- Modello Classificatore BS_MidSup

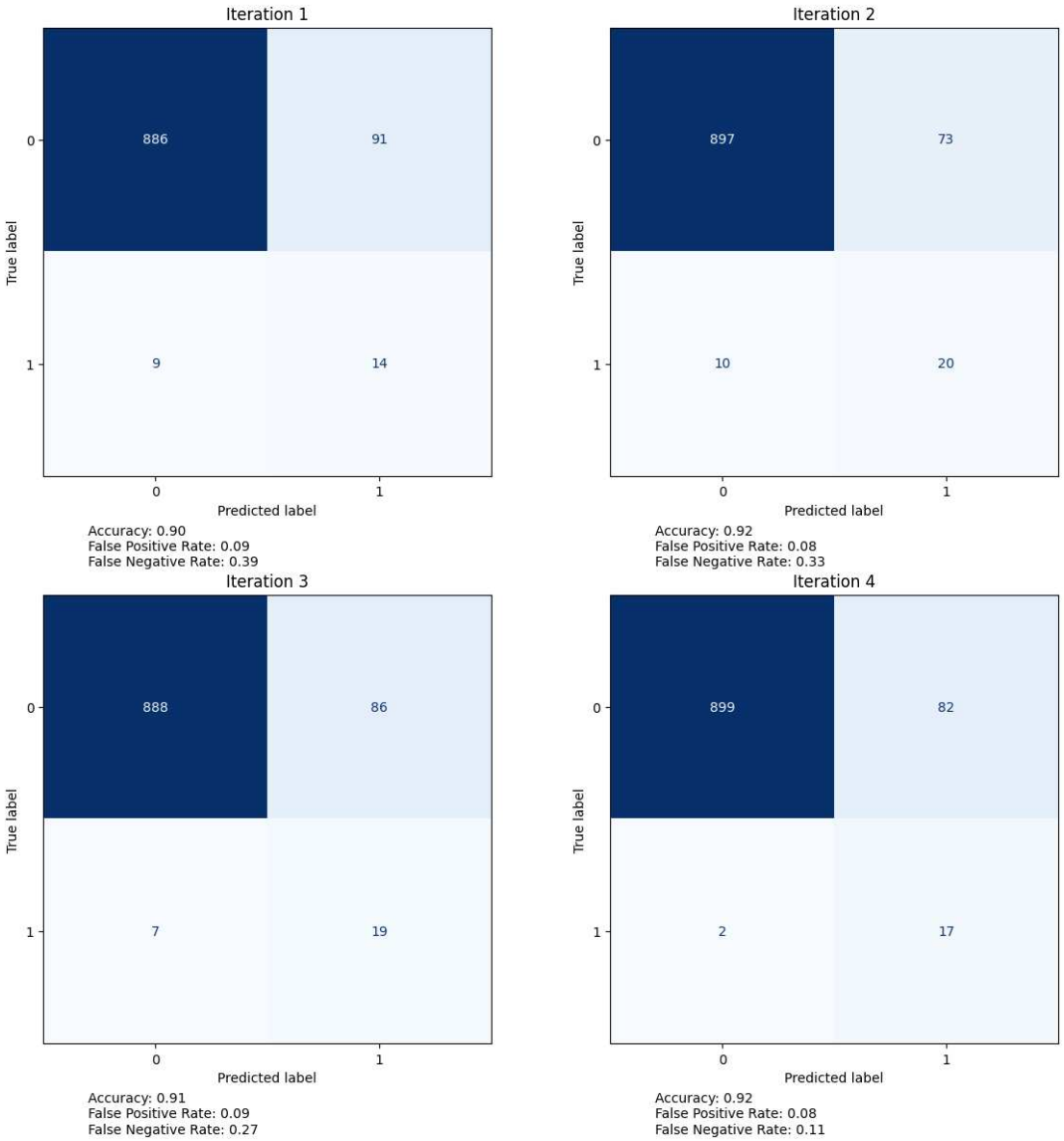


Figura 38: Risultati Testing con Database NoSQL modello BS_MidSup

Capitolo 5

Conclusioni e Sviluppi futuri

Le specifiche tecniche richieste dall'azienda Ariston Group per il modello di diagnosi guasti sono:

- False Positive Rate: 6%
- False Negative Rate: 0.5%

Analizzando i risultati ottenuto dall'addestramento dei classificatori (Figura 31-36) basato su una rete neurale per i due fault di basso spessore presi in considerazione, si possono notare per il modello di classificazione BS_Sup ottime metriche, con tutti i valori al di sopra del 95% compresa la metrica AUC-ROC della curva ROC e valori pari al 4% per il False Positive Rate ed il False Negative Rate.

Per quanto riguarda il modello di classificazione BS_MidSup, si possono notare risultati comunque positivi con valori delle metriche che variano tra il 71% e il 92% compresa la metrica AUC-ROC della curva ROC pari all'82%, però bisogna sottolineare come, nonostante un buon False Positive Rate pari al 7%, si ha un False Negative Rate pari al 28%, un valore troppo alto.

Malgrado i risultati positivi dei due classificatori, soprattutto di quello per il basso spessore della parte superiore dei boiler, le specifiche tecniche richieste dall'Ariston Group non sono state soddisfatte e ciò si nota anche dal testing tramite il database NoSQL (Figura 37,38) dove anche in quel caso le metriche non soddisfano le richieste fatte. Essendo un lavoro di ricerca, è stata intrapresa una strada che poteva non portare ad un risultato positivo, ma anche un non risultato è comunque un punto di partenza per intraprendere nuove strade al fine di arrivare all'obiettivo richiesto.

Come già anticipato, la soluzione adottata per portare a termine il lavoro può presentare dei limiti di progettazione impliciti, dati dalle label target usate: la scelta di etichettare i predittori con una label pari a 0 o 1 è frutto della tecnica usata precedentemente mediante delle soglie fisse validate da un intervento umano, per cui tutto il processo di classificazione supervisionato si basa sia sul modo in cui sono state calcolati i predittori sia sulla bontà delle label target che possono presentare incertezza. Inoltre un ulteriore limite può essere rappresentato dal limitato numero di features utilizzate.

Questo lavoro è stato basato su un primo approccio di automatizzazione del processo di controllo qualità, andando non a rivoluzionare il metodo quotidianamente utilizzato ma ad integrarne un miglioramento senza rompere tutta la catena di controllo già creata (Foto-Estrazioni Features-Valutazione tramite Soglie). Seguendo questa logica, tutto l'addestramento dei modelli di classificazione è stato portato avanti tramite l'uso del dataset formato da dati numerici estratti tramite il processo di controllo qualità gia in uso. Essendo un dataset semplice, formato solo da features numeriche e da target label binarie, è stato scelto di approcciarsi al problema utilizzando

un metodo meno complicato e più facile da implementare, ossia una classica rete neurale con 3 hidden layer e un numero ristretto di neuroni.
Come sviluppi futuri, si possono intraprendere due strade:

1) per ovviare al possibile limite dato dal numero dei predittori, possono essere calcolate nuove features dalle foto, ampliando quelle già utilizzate così da aumentare la quantità di informazione da dare al modello di classificazione in fase di addestramento.
   Nell'ottica di poter intraprendere questa strada, è stato scelto di integrare nel progetto un database NoSQL, in modo tale da poter modificare ogni elemento aggiungendo le nuove features calcolate.

2) nell'ottica di eliminare l'incertezza di progettazione data dalle target label, è possibile utilizzare un modello di classificazione basato su una rete neurale CNN (Convolutional Neural Network), in grado di prendere come input direttamente le foto da dover classificare.
   Questa scelta può eliminare l'incertezza poiché si utilizzano direttamente le foto senza doverle prima processare per estrarre i predittori, eliminando l'eventuale incertezza data dal modo di calcolare le features, e si limita l'intervento umano per la validazione delle stesse.

Bibliografia

1) Abid, Anam & Khan, Muhammad & Iqbal, Javaid. (2021). A review on fault detection and diagnosis techniques: basics and beyond. Artificial Intelligence Review. 54. 1-26. DOI: 10.1007/s10462-020-09934-2.

2) Mrugalska, Tytyk. (2015). Quality control methods for product reliability and safety. Procedia Manufacturing, 3, 2730-2737. https://doi.org/10.1016/j.promfg.2015.07.683

3) Frank, Paul & Köppen-Seliger, Birgit. (1997). New Developments Using AI in Fault Diagnosis. Engineering Applications of Artificial Intelligence. 10. DOI: 10.1016/S0952-1976(96)00072-3.

4) Mahesh, Batta. (2019). Machine Learning Algorithms -A Review. International Journal of Science and Research (IJSR) DOI: 10.21275/ART20203995.

5) Janiesch, Christian & Zschech, Patrick & Heinrich, Kai. (2021). Machine learning and deep learning. Electronic Markets. 31. DOI: 10.1007/s12525-021-00475-2.

6) Sharma, Neha & Sharma, Reecha & Jindal, Neeru. (2021). Machine Learning and Deep Learning Applications-A Vision. Global Transitions Proceedings. 2. DOI: 10.1016/j.gltp.2021.01.004.

7) Al-Kaff, A., Martín, D., García, F., de la Escalera, A., & Armingol, J. M. (2018). Survey of computer vision algorithms and applications for unmanned aerial vehicles. Expert Systems With Applications, 92, 447-463. https://doi.org/10.1016/j.eswa.2017.09.033

8) Petru Potrimba, What is R-CNN?, Roboflow, (Sep 25, 2023). Blog: https://blog.roboflow.com/what-is-r-cnn/

9) Liu, Chengji & Tao, Yufan & Liang, Jiawei & Li, Kai & Chen, Yihang. (2018). Object Detection Based on YOLO Network. 799-803. DOI: 10.1109/ITOEC.2018.8740604.

10) Dutta, A., Mondal, A., Dey, N., Sen, S., Moraru, L., & Hassanien, A. E. (2020). Vision Tracking: A Survey of the State-of-the-Art1. SN Computer Science, 1(57). https://doi.org/10.1007/s42979-019-0059-z2

11) Hou, Xinyu & Wang, Yi & Chau, Lap-Pui. (2019). Vehicle Tracking Using Deep SORT with Low Confidence Track Filtering. 1-6. DOI: 10.1109/AVSS.2019.8909903.

12) Wen, Zhi-Qiang & Cai, Zi-Xing. (2006). Mean Shift Algorithm and its Application in Tracking of Objects. 2006. 4024 - 4028. DOI: 10.1109/ICMLC.2006.258803.

13) Bart De Ketelaere, Niels Wouters, Ioannis Kalfas, Remi Van Belleghem & Wouter Saeys (2022) A fresh look at computer vision for industrial quality control, Quality Engineering, 34:1, 152-158, DOI: 10.1080/08982112.2021.2001828

14) Brunelli, R. (2021). Template Matching Techniques in Computer Vision: Theory and Practice. John Wiley & Sons, Inc.

15) Wang, Tian & Chen, Yang & Qiao, Meina & Snoussi, Hichem. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control. The International Journal of Advanced Manufacturing Technology. 94.
DOI: 10.1007/s00170-017-0882-0.

16) Vettoretti, Martina & Di Camillo, Barbara. (2021). A Variable Ranking Method for Machine Learning Models with Correlated Features: In-Silico Validation and Application for Diabetes Prediction. Applied Sciences. 11. 7740.
DOI: 10.3390/app11167740.

17) Chawla, Nitesh & Bowyer, Kevin & Hall, Lawrence & Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357.
DOI: 10.1613/jair.953.

18) Zheng, A. (2015). Evaluation Metrics. In Zheng, A. (Ed.), Evaluating Machine Learning Models (pp 7-18). O'Reilly Media, Inc.

Fonti immagini

Figura 1: Freddi A., slide del corso di Manutenzione preventiva per l'Automazione e la Robotica, Università Politecnica delle Marche, 2022.

Figura 2: D'Agostino A, "Cos'è il Machine Learning: la definizione di un data scientist", Diario di Un Analista, 31 marzo 2022.
https://www.diariodiunanalista.it/posts/cosa-e-il-machine-learning/

Figura 3: "Machine Learning Cos'è? Il Futuro delle Aziende del Domani", Esker, 08 novembre 2023.
https://www.esker.it/blog/tecnologia/machine-learning-cose-il-futuro-delle-aziende-del-domani/

Figura 4: "Machine Learning Cos'è? Il Futuro delle Aziende del Domani", Esker, 08 novembre 2023.
https://www.esker.it/blog/tecnologia/machine-learning-cose-il-futuro-delle-aziende-del-domani/

Figura 5: Agostini A, "ARTIFICIAL INTELLIGENCE, MACHINE LEARNING O DEEP LEARNING? POSSIBILI APPLICAZIONI ALL'ECOSISTEMA BIM?", Digitalbimitalia, 19 settembre 2018
https://www.digitalbimitalia.it/it/news/artificial-intelligence-machine-learning-o-deep-learning-possibili-applicazioni-allecosistema-bim/

Figura 6: Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. The MIT Press

Elenco delle Figure