UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea in ingegneria informatica e dell'automazione

Tecniche di apprendimento automatico per la manutenzione predittiva

Studenti:
Riccardo MANCINI
Arment PELIVANI

Docente:
Prof. Domenico Potena

Anno Accademico 2022-2023

Indice

Capitolo 1

Introduzione

Nel contesto industriale, la durata e l'affidabilità dei componenti sono elementi critici che influenzano direttamente l'efficienza e la produttività dei processi di produzione. Nell'ambito di questo progetto, l'obiettivo principale è quello di condurre un confronto dettagliato tra diverse tecniche di apprendimento automatico impiegate a svolgere operazioni di stima e previsione del tempo di vita rimanente e della probabilità di sopravvivenza di componenti fondamentali nel settore, in particolare gli elettromandrini.

Gli elettromandrini, essendo elementi centrali di molte macchine industriali, sono soggetti a sollecitazioni e usure che possono influire sulla loro durata e sulle prestazioni nel tempo. La capacità di stimare in modo accurato il tempo di vita rimanente di tali componenti può consentire alle aziende di pianificare attività di manutenzione preventive, riducendo i costi e minimizzando gli arresti imprevisti della produzione.

Tra gli algoritmi di apprendimento automatico supervisionato scelti, i modelli parametrici si basano sull'ipotesi che i dati seguano una distribuzione statistica specifica, come ad esempio la distribuzione esponenziale o Weibull. Questi modelli richiedono una serie di assunzioni sul comportamento dei dati e sono solitamente efficienti quando queste assunzioni sono valide. D'altra parte, i modelli non parametrici non fanno assunzioni sulla distribuzione dei dati, offrendo così una maggiore flessibilità ma richiedendo un maggior volume di dati per ottenere stime affidabili.

L'obiettivo finale di questo progetto è fornire un quadro chiaro e basato su evidenze scientifiche per prendere decisioni informate sulla manutenzione e la sostituzione degli elettromandrini. Una migliore comprensione dei modelli più appropriati per stimare il tempo di vita rimanente dei componenti può tradursi in un notevole risparmio di tempo, denaro e risorse, migliorando l'efficienza e la produttività complessiva delle operazioni industriali.

1.1 Dataset

In questa sezione, verranno presentati i dataset utilizzati per addestrare i modelli parametrici e non parametrici menzionati in precedenza. I dati svolgono un ruolo fondamentale nello sviluppo di modelli predittivi accurati e affidabili, poiché essi

consentono di cogliere le informazioni rilevanti per stimare il tempo di vita rimanente e la probabilità di sopravvivenza degli elettromandrini discussi.

La creazione dei dataset è stata effettuata attraverso un processo di parsing delle informazioni provenienti dai file di log generati dai sensori che registrano le vibrazioni di lavoro degli elettromandrini, montati nei rispettivi macchinari. Il parsing dei file di log ha coinvolto l'estrazione delle informazioni rilevanti dai dati grezzi. Una volta estratte le informazioni pertinenti, i dati sono stati organizzati in un formato adatto per l'analisi e l'utilizzo nei dataset in questione. Essi sono stati strutturati in colonne o feature corrispondenti alle diverse informazioni temporali, carichi di lavoro e caratteristiche specifiche dei componenti.

Successivamente, verranno forniti brevi approfondimenti sui dataset impiegati nel progetto, offrendo una panoramica dettagliata delle informazioni raccolte e della loro rilevanza per i modelli predittivi utilizzati.

1.1.1 Dataset 1

Il primo dataset utilizzato è stato raccolto e curato per riflettere le reali condizioni operative e le caratteristiche dei componenti elettromeccanici. Tale set di dati è composto da un totale di 62 righe, corrispondenti allo storico di lavorazione di altrettanti elettromandrini. Di queste righe, solo 5 caratterizzano componenti sostituiti per guasto.

Ogni registrazione all'interno del dataset è descritta da un insieme di 64 variabili (feature), tra cui:

- **snMacchina**: Identificativo della macchina che utilizza l'elettromandrino.
- **snEm**: Identificativo dell'elettromandrino.
- **tp**: Tipologia di elettromandrino.
- **classe**: Indica lo stato attuale del componente. La classe 5 indica che l'elettromandrino non è mai stato sostituito ed è ancora in lavorazione, mentre la classe 3 indica che l'elettromandrino è stato sostituito a causa di un guasto.
- **startDate**: Data in cui è stato montato l'elettromandrino.
- **endDate**: Data in cui è stato utilizzato per l'ultima volta l'elettromandrino.
- **deltaDateHour**: Ore totali per cui l'elettromandrino è rimasto montato alla macchina, anche senza lavorare.
- **Ore_lavtotali**: Ore complessive di lavoro effettivo dell'elettromandrino.
- **Perc_ore_lav**: Percentuale di lavoro effettivo dell'elettromandrino rispetto al tempo totale.
- **Lav_mancanti**: Numero di lavorazioni mancanti e non riportate nel dataset.

- **Perc lav manc**: Percentuale di lavorazioni mancanti rispetto al totale.
- **Intervalli di vibrazione**: Numero di ore in cui l'elettromandrino ha trascorso in determinati intervalli di vibrazione.

1.1.2 Dataset 2

Il secondo dataset è stato costruito in modo progressivo, anche in questo caso utilizzando i diversi file di log, ma solo dei relativi componenti guasti. Ognuno dei quali è stato suddiviso in diversi intervalli di 24 ore di lavoro effettivo. In dettaglio, per ciascun componente guasto sono stati generati X intervalli di lavoro effettivo, con X che varia da 24 fino al totale di ore di lavoro effettuate meno 24.

Ogni intervallo di 24 ore ha costituito un elemento del dataset, includendo sia le caratteristiche delle vibrazioni che altre informazioni rilevanti, utilizzate e discusse nel dataset precedente. Infine, tutte queste registrazioni sono state combinate per formare il dataset completo, che risulta composto da un totale di 88 elementi.

A differenza del Dataset 1 però, è stata introdotta una variabile aggiuntiva denominata "Ore di lavoro rimanenti" per ciascuna registrazione. Questa variabile rappresenta una stima delle ore di lavoro rimanenti prima che il componente venga sostituito. Integrando le informazioni sulle vibrazioni con le ore di lavoro rimanenti, il secondo dataset ha fornito un'ulteriore dimensione di previsione, permettendo di ottimizzare la manutenzione e prevenire potenziali guasti futuri in modo più accurato.

Di seguito un esempio concreto di alcune istanze generate a partire da un solo elettromandrino fail che è stato evidenziato in fondo alla Figura 1.1.

delta_date_hour	tempo_lav	ore_lav_rim	[0.0-0.5)	[0.5-1.5)	[1.5-2.5)	...	...	[38.5-39.5)	[39.5-40.5)
1803	24.0	527.82	4199	20539	20980	...	...	0	0
1869	48.0	503.82	11468	39391	56133	...	...	0	0
1991	72.0	479.82	13890	57462	101222	...	...	0	0
2067	96.0	455.82	16104	92306	129929	...	...	0	0
2184	120.0	431.82	17308	130825	163650	...	...	0	0
2311	144.0	407.82	17801	147101	205930	...	...	0	0
2345	168.0	383.82	18922	180396	244936	...	...	0	0
2377	192.0	359.82	22124	219746	276514	...	...	0	0
2462	216.0	335.82	24331	251006	298054	...	...	0	0
2503	240.0	311.82	25043	269212	317580	...	...	0	0
2551	264.0	287.82	27100	289560	334599	...	...	0	0
2599	288.0	263.82	27134	300088	356728	...	...	0	0
2670	312.0	239.82	27194	322479	389149	...	...	0	0
2706	336.0	215.82	27194	333680	412326	...	...	0	0
2811	360.0	191.82	28746	379173	443241	...	...	0	0
2929	384.0	167.82	30723	401058	478665	...	...	0	0
3097	408.0	143.82	36275	419341	507173	...	...	0	0
3309	432.0	119.82	39836	454639	537294	...	...	0	0
3389	456.0	95.82	41596	475101	561857	...	...	0	0
3505	480.0	71.82	42700	500503	595374	...	...	0	0
3584	504.0	47.82	42709	524491	644285	...	...	0	0
3731	528.0	23.82	42709	548316	694531	...	...	0	0
3844	551.82	0.0	43953	572332	735649	...	...	0	0

Figura 1.1: Alcune istanze del Dataset 2 generate dallo stesso componente

Capitolo 2

Analisi e preparazione dati

Nel corso di questo capitolo, saranno esplorate in dettaglio le diverse tecniche di manipolazione e pre-processing dei dati utilizzate per affrontare le diverse problematiche dei campioni di dati e garantire una solida base per le analisi successive al fine di ottenere risultati più affidabili e significativi sia in fase di training che in fase di test.

Tra i diversi approcci implementati è stato necessario operare su entrambi i set di addestramento mediante una operazione di sovracampionamento degli elettromandrini, cercando di ottenere un numero di dati superiore su cui addestrare i vari modelli.

Infine sono state studiate ed applicate tecniche di riduzione delle dimensionalità per affrontare la sfida legata alla complessità e alla sparsità dei dati.

2.1 Analisi della relazione tra componenti guasti e non guasti

Generalmente, quando viene implicato un modello parametrico in un task di Data Mining, è consigliabile utilizzare dati per il training e il test che si distribuiscono in modo simile nello spazio delle feature. Ciò significa che i dati di training e di test dovrebbero coprire una gamma adeguata di valori per ciascuna feature considerata nel modello. Ciò è importante perché un modello parametrico, come in questo caso la distribuzione di Weibull, si basa sull'assunzione che i dati seguano una specifica distribuzione statistica. Se i dati di training e di test hanno una distribuzione simile, ciò può aumentare la validità delle stime dei parametri del modello e delle sue prestazioni predittive. Tuttavia, è anche fondamentale assicurarsi che i dati di test siano rappresentativi di nuovi dati che il modello potrebbe incontrare nel mondo reale. Quindi, sebbene sia auspicabile una distribuzione simile tra i dati di training e di test, è importante garantire che i dati di test siano ancora indipendenti e rappresentativi per valutare le prestazioni del modello su nuovi campioni.

Per far si che questo presupposto teorico venisse rispettato, nel primo dataset è stata necessaria una rielaborazione dell'approccio al task iniziale ed un filtraggio dei dati del campione. Pertanto, si è deciso di addestrare i modelli esclusivamente sui dati dei componenti guasti, nonostante la loro esiguità, e successivamente testarli sui componenti non guasti mantenendo solo parte di questi ultimi, ossia quelli che

avessero operato con un carico di lavoro simile a quelli guasti presenti nel training set. Questa correlazione tra componenti è stata espressa in termini di distanza Euclidea.

Entrando nello specifico dell'algoritmo elaborato, si è costruito una matrice che ha per righe i componenti non guaste e per colonne i componenti guasti. Per ogni componente guasto si è calcolata la distanze con tutte le componenti non guaste ottenendo in questo modo una matrice a struttura rettangolare 46 x 5. Dopo aver normalizzato tale matrice delle distanze, è stata calcolata opportunamente la distanza media per ciascuna di riga, fornendo così la distanza media di ogni componente non guasto rispetto ai componenti guasti. Successivamente, è stata identificata una soglia di filtraggio, determinata come la media delle medie delle distanze. Questa soglia è stata poi utilizzata per eliminare i componenti che presentavano una maggiore diversità, ossia quelli più distanti dai componenti guasti.

2.2 Sovracampionamento dei componenti guasti

In generale, per affrontare la problematica della scarsità di dati esistono varie tecniche a partire dalla più elementare che consiste in un sovracampionamento casuale. Questa tecnica comporta la duplicazione casuale di esempi appartenenti al set di dati di addestramento. Per via della sua semplicità, applicarla in un contesto come questo in cui si hanno pochissimi dati avrebbe voluto dire incombere a degli errori di generalizzazione consistenti (overfitting).

Un'ulteriore tecnica di oversampling è lo SMOTE che funziona selezionando esempi vicini nello spazio delle caratteristiche, tracciando una linea tra gli esempi stessi e individuando un nuovo campione in un punto lungo quella linea. Il problema di questa tecnica deriva dalla distanza degli esempi; infatti, se questa risultasse troppo lontana si genererebbero dei campioni non sufficientemente rappresentativi del dataset di partenza, distorcendo di conseguenza i risultati.

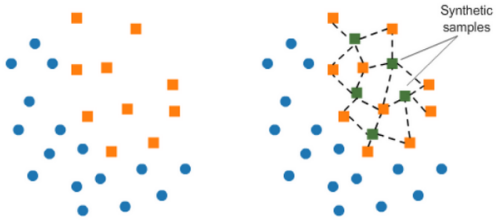

Figura 2.1: Tecnica di oversampling - SMOTE

Preso atto di quanto appena detto, si è deciso di usare una tecnica di oversampling che prende il nome Gaussian Noise Up-Sampling (GNUS). Il rumore gaussiano è distribuito in modo uniforme intorno a un valore medio. Quando viene applicato a un campione, il rumore gaussiano fa sì che il campione si sposti leggermente, creando un nuovo campione che è simile all'originale ma non identico.

Entrando più nel dettaglio, quello che si fa è:

- Estrarre casualmente dei campione tra quelli rappresentanti componenti guasti
- Per ciascuna istanza estratta, si aggiunge del rumore gaussiano ai suoi attributi. Il rumore gaussiano è generato da una distribuzione gaussiana (o normale) che ha media 0 e una deviazione standard specificata, direttamente calcolata dai valori assunti dalle feature del problema
- I nuovi campioni sintetici saranno leggermente diversi dagli esempi originali, ma ancora rappresentativi della classe sottorappresentata. A questo punto vengono aggiunti al dataset originale.

Il risultato di questa operazione, come si può vedere in Figura 2.2, saranno una serie di campioni sintetici definiti nell'intorno dei singoli campioni di minoranza permettendo così di bilanciare il dataset, distorcendo al minimo l'informazione.

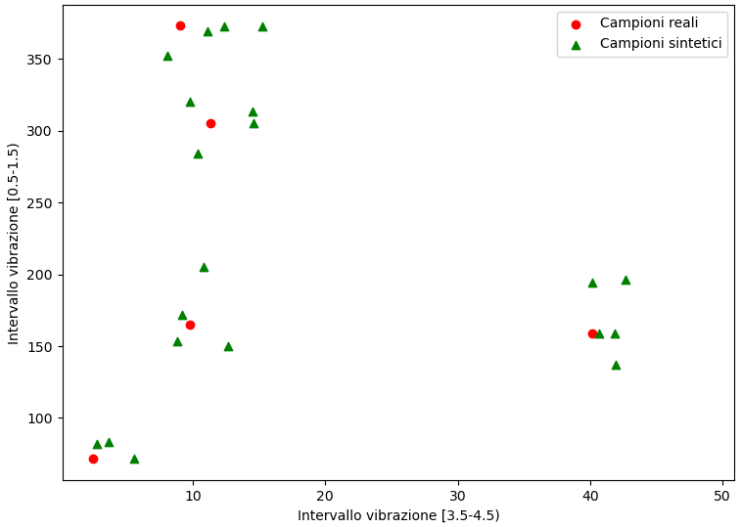

Figura 2.2: Tecnica di oversampling - GNUS

2.3 Tecniche di riduzione delle dimensionalità e loro impiego

I modelli utilizzati richiedono dataset di dimensioni non eccessive per ottimizzare le prestazioni, a causa del fenomeno noto come 'curse of dimensionality'. Questo problema matematico/statistico si verifica quando si lavora con dataset di dimensioni elevate. Secondo una legge empirica, se ad ogni dimensione aggiuntiva non aumenta in modo esponenziale la dimensione del dataset, è molto probabile che le prestazioni dei modelli di classificazione o regressione si degradino. Ciò accade perché l'aumento delle dimensioni comporta un aumento della distanza tra gli elementi nello spazio delle features.

Il curse of dimensionality comporta diversi effetti indesiderati. Da un lato, la sparsità dei dati diventa un problema, poiché con un maggior numero di dimensioni, i punti dati si distribuiscono in modo più irregolare e si riduce la densità di informazioni disponibili. Ciò può influire negativamente sulla capacità dei modelli di identificare relazioni significative e di generalizzare correttamente i dati. Dall'altro lato, l'aumento delle dimensioni aumenta l'incertezza negli algoritmi che cercano di definire il bordo di separazione ottimale tra le classi o un bordo che approssimi meglio possibile la distribuzione di dati, come gli algoritmi SVM. Con un maggior numero di dimensioni, aumenta il numero di gradi di libertà dell'algoritmo, che può comportare la mancata convergenza o l'aumento della probabilità di errori nella definizione del bordo. Riducendo il numero di dimensioni, si cerca di mitigare gli effetti negativi del curse of dimensionality, consentendo ai modelli di operare in modo più efficiente e ottenere risultati migliori su dataset di dimensioni ragionevoli.

Tenendo conto di ciò, la prima operazione applicata è stata quella di rimozione di feature irrilevanti come gli identificativi delle macchine e dei elettromandrini, le percentuali di lavorazione, i periodi di inizio e fine delle lavorazione e il relativo delta correlato a questi ultimi due parametri. Le operazioni di rimozione hanno permesso di ottenere un dataset con solo feature relative alle ore totali di lavorazioni dei singoli elettromandrini, le classi, e le ore di lavorazione trascorse nei singoli range di vibrazione.

Inoltre, per poter ridurre ulteriormente le dimensionalità, si è deciso di applicare ai diversi range di vibrazione una tecnica di feature selection di tipo filtering, ossia la *Correlazione di Pearson*.

La *Correlazione di Pearson* è un metodo statistico che consente di misurare la correlazione lineare tra due variabili, in questo caso i singoli intervalli di vibrazione con la variabile decisionale. Più nello specifico, è espressa come:

$$\rho = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

ossia, il rapporto tra la covarianza tra le due variabili statistiche X,Y con il prodotto tra le singole deviazioni standard delle variabili X e Y.

La *Correlazione di Pearson* riporta risultati espressi in un range di valori compresi tra $+1$ e -1, il quale $+1$ corrisponde alla correlazione lineare positiva, 0 a nessuna correlazione lineare e -1 a correlazione lineare negativa.

Nonostante l'applicazione della *Correlazione di Pearson*, è emerso che molti intervalli di vibrazione presentavano valori pressoché nulli, privi di informazione utile. Pertanto, sono state rimosse le feature relative all'intervallo di vibrazione da 10.5 a 50. Invece, gli intervalli che contenevano informazioni rilevanti, individuati attraverso la *Correlazione di Pearson* con un valore di soglia superiore a 0.5, erano quelli compresi tra 0.0 e 5.5. Successivamente, tali intervalli sono stati combinati con le vibrazioni che possedevano informazioni utili negli intervalli da 5.5 a 10.5.

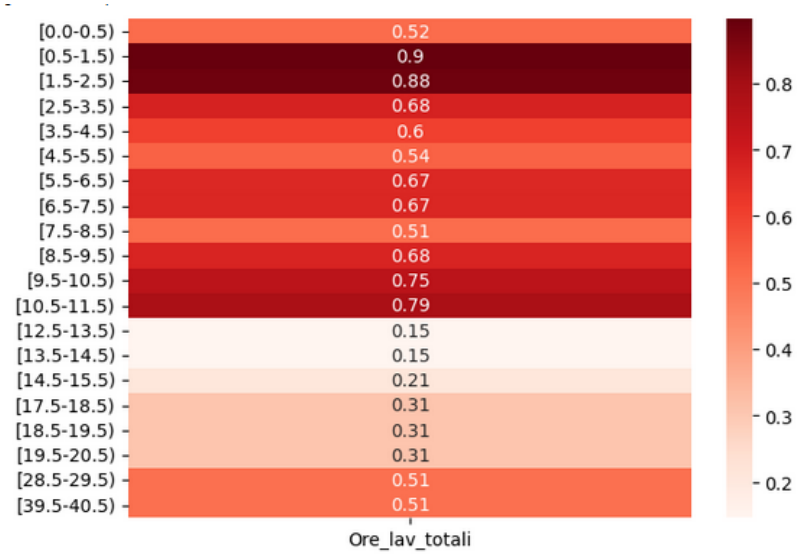

Figura 2.3: Risultati della correlazione di Pearson tra gli intervalli di vibrazione e le ore di lavorazione complessive

Capitolo 3

Tecniche di regressione e addestramento

Concluse le precedenti operazioni di manipolazione dei dataset, si è proseguito con la costruzione dei modelli predittivi. In particolare, il parametro che si è deciso di stimare sono le ore di lavorazione totali per quanto riguarda il primo dataset, e le ore di lavoro rimanenti per il secondo dataset. Per fare questo si è deciso di confrontare le performance di 2 modelli, uno parametrico che prende il nome di Distribuzione di Weibull e uno non parametrico, ovvero l'SVR.

3.1 Modello parametrico: Distribuzione di Weibull

La distribuzione di Weibull è una distribuzione di probabilità continua che viene spesso utilizzata per modellare il tempo di decadimento o di guasto di un oggetto. La distribuzione di Weibull è definita dalla seguente funzione di densità di probabilità (pdf):

$$f(x;k,\gamma) = \frac{k}{\gamma}x^{k-1}e^{-\frac{x^k}{\gamma}} \tag{3.1}$$

dove k e γ sono due parametri reali positivi. Il parametro k controlla la forma della curva pdf, mentre il parametro γ controlla la scala della curva. Con $k > 1$ si ha una distribuzione con tasso di fallimento crescente e $k < 1$ indica una distribuzione con tasso di fallimento decrescente. Quando $k = 1$, la distribuzione di Weibull si riduce a una distribuzione esponenziale. Invece, Un valore maggiore di γ sposta la curva a destra, mentre un valore minore di γ sposta la curva a sinistra.

A partire dalla pdf, per poter modellare la durata di vita dei componenti elettromeccanici, è stata di fondamentale utilizzo la funzione di sopravvivenza. Tale funzione permette di calcolare la probabilità che un oggetto sopravviva almeno fino a un certo tempo x. La funzione di sopravvivenza della distribuzione di Weibull è definita dalla seguente equazione:

$$S(x;k,\gamma) = e^{-\frac{x^k}{\gamma}} \tag{3.2}$$

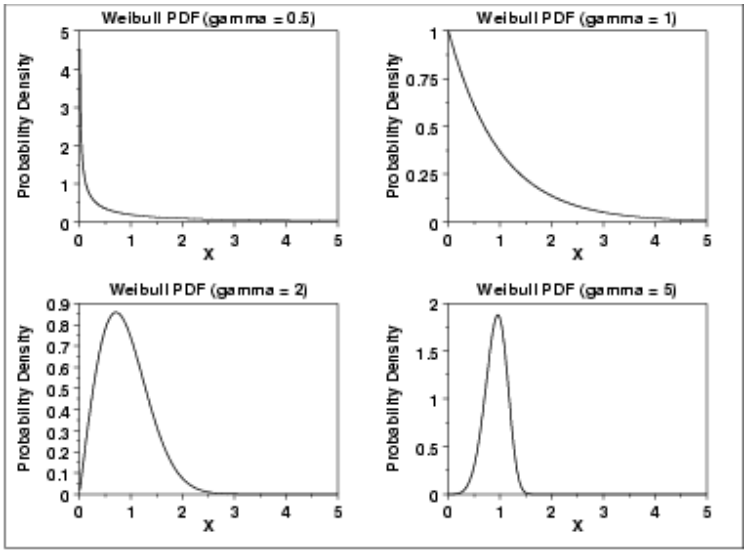

Figura 3.1: PDF al variare di gamma

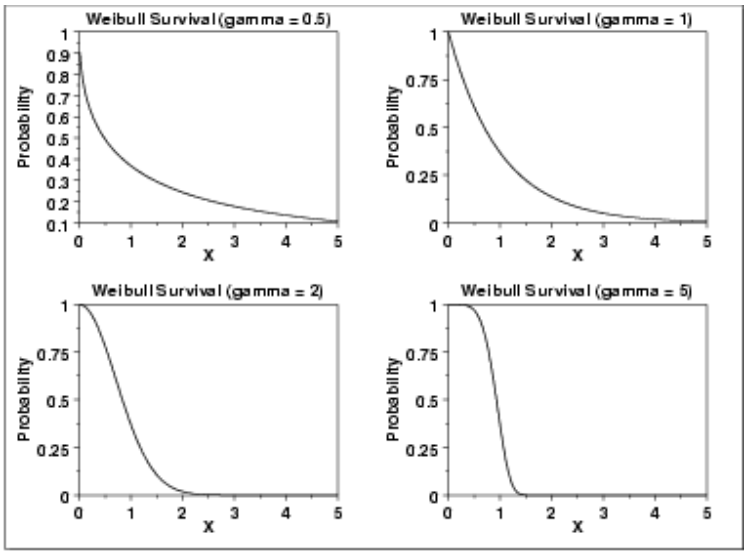

Figura 3.2: Funzione di sopravvivenza al variare di gamma

3.1.1 Configurazione per l'addestramento

Il modello parametrico di Weibull è stato utilizzato sfruttando la libreria *lifelines.WeibullAFTFitter*. Questa libreria prende in input un set di dati di addestramento e una colonna target, che nel primo dataset è l'ora di lavoro totale e nel secondo dataset è l'ora di lavoro rimanente. La libreria quindi calcola direttamente dal campione i parametri, precedentemente discussi, che definiscono la distribuzione.

Inoltre, la libreria permette di utilizzare tutte le funzioni che derivano dal modello Weibull, come la funzione di sopravvivenza impiegata per la probabilità di sopravvivenza dei dati del primo dataset e il metodo *predict expectation* per predire l'aspettativa di vita degli elettromandrini in entrambi i dataset. Questo metodo pren-

de in input un set di dati di test e i parametri del modello Weibull precedentemente stimati dal training set, consentendone di calcolare l'aspettativa di vita tramite la probabilità a posteriori.

Di seguito alcune illustrazioni, sui modelli calcolati in entrambi i dataset, utilizzando tale algoritmo.

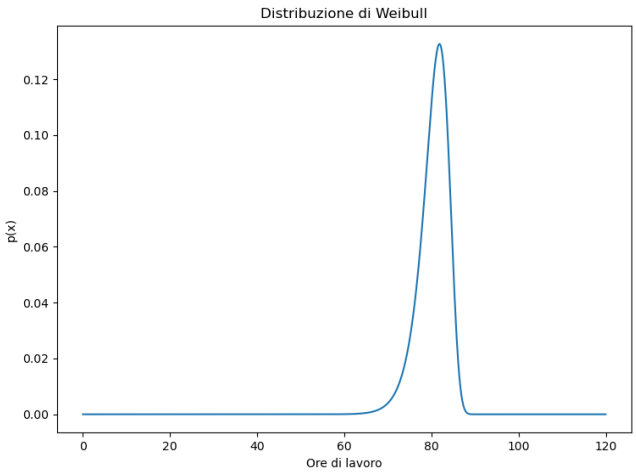

Figura 3.3: PDF (forma=29.5 e scala=81.9) calcolata sui dati di train del Dataset 1

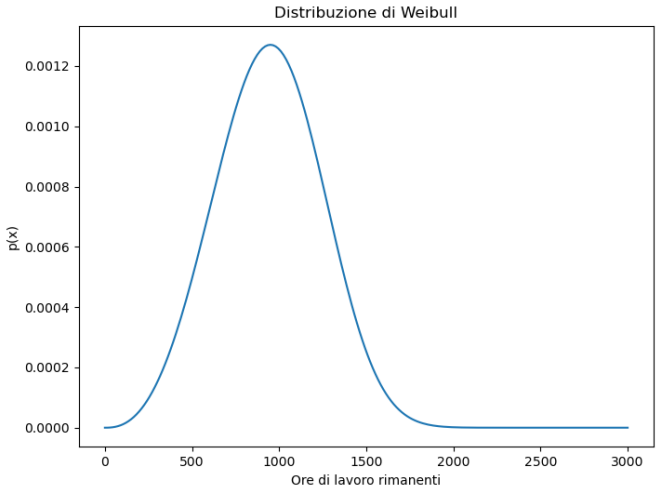

Figura 3.4: PDF (forma=3.4 e scala=1048) calcolata sui dati di train del Dataset 2

3.2 Modello non parametrico: Support Vector Regression - SVR

La regressione vettoriale di supporto (Support Vector Regression - SVR) è un algoritmo di apprendimento automatico non parametrico che viene utilizzato per la regressione. il suo vantaggio principale consiste nel poter modellare oltre alle relazioni lineari anche quelle non lineari. L'SVR ha come scopo quello di trovare un iperpiano, che in generale viene definito dalla kernel function(questa può essere polinomiale, lineare radiale o a base gaussiana), nello spazio delle feature che si adatti meglio ai dati forniti in ingresso. L'idea dietro questo algoritmo è quella di minimizzare la norma $l2$ del vettore dei coefficienti, rappresentata dalla seguente funzione obiettivo:

$$MIN\frac{1}{2}\|w\|^2$$

imponendo come vincoli

$$|y_i - w_i x_i| \leq \varepsilon$$

con y_i target di predizione, x_i che rappressenta il predittore e ε errore ammissibile(tolleranza).

In generale a questo modello, vengono aggiunte delle variabile di slack per misurare l'entità delle violazioni, in modo da permettere al modello di adattarsi ai dati in maniera flessibile.

Il modello completo ha come funzione obiettivo

$$MIN\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}|\zeta|$$

con ζ che rappresenta l'entità della violazione e C l'iperparametro da determinare. I vincoli relativi invece sono indicati come

$$|y_i - w_i x_i| \leq \varepsilon + |\zeta|$$

In Figura 3.5 una rappresentazione del problema appena discusso.

3.2.1 Configurazione per l'addestramento

Il modello SVR per la predizione delle ore di vita dei singoli elettromandrini, realizzato mediante l'utilizzo della libreria *sklearn.svm*, prevede come parametri: il target di predizione, la funzione di kernel che andrà ad approssimare i dati, il parametro di regolarizzazione C, il termine che indica l'intervallo di errore ammissibile ε, il termine indipendente $coef0$, significativo solo per le funzioni polinomiali e sigmoidali, e il coefficiente della funzione di kernel γ nel caso radiale, polinomiale e sigmoidale. Per stimare la configurazione ottimale dei precedenti parametri si deciso

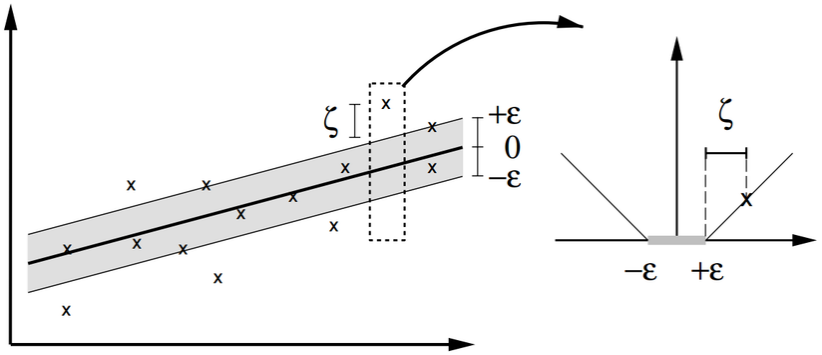

Figura 3.5: SVR

di effettuare delle ricerche esaustive (grid-search) in entrambi i set di addestramento previsti dei 2 dataset.

La ricerca greedy ha fornito risultati notevolmente precisi per tutte le funzioni kernel testate, associate a diverse combinazioni di parametri. Tuttavia, per quanto riguarda il primo dataset, è importante notare che il modello SVR può ottenere una previsione esatta utilizzando la somma delle ore di lavorazione negli specifici intervalli di vibrazione registrare per ogni singolo componente. Questo di fatto porta ad una limitazione nel modello nel riconoscere in modo affidabile quando un elettromandrino no-fail avrà un guasto.

Per ovviare a questo problema, si è introdotta una discretizzazione delle ore di lavorazione, con passo di campionamento pari a 12, 24 e 48 ore. In seguito, per ogni intervallo di campionamento si sono determinate una serie di fasce orarie di lavorazione, per evitare che il modello predicesse i valori mediante la somma delle ore nei singoli intervalli di vibrazione.

Per quanto riguarda il secondo dataset invece, non è stata necessaria alcuna modifica, in quanto le ore trascorse su ogni intervallo non hanno alcuna correlazione con la stima delle ore di lavoro rimanenti.

Di seguito alcune illustrazioni, sui modelli calcolati in entrambi i dataset, utilizzando tale algoritmo.

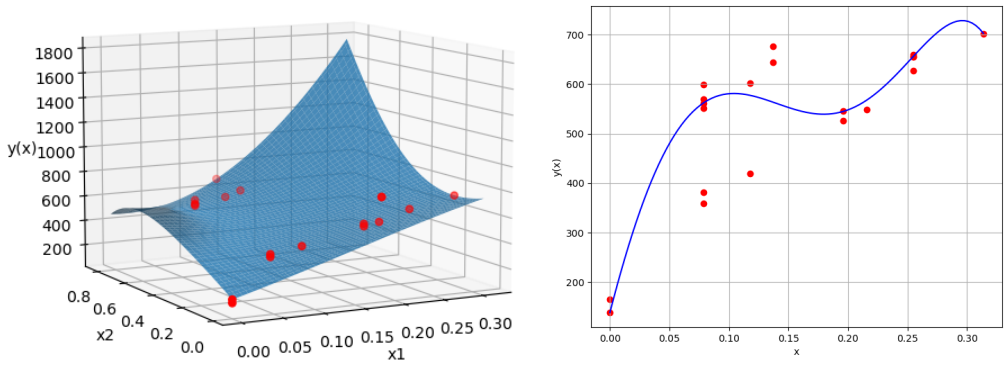

Figura 3.6: Modello 3D e 2D della funzione definita sui dati di train del Dataset 1

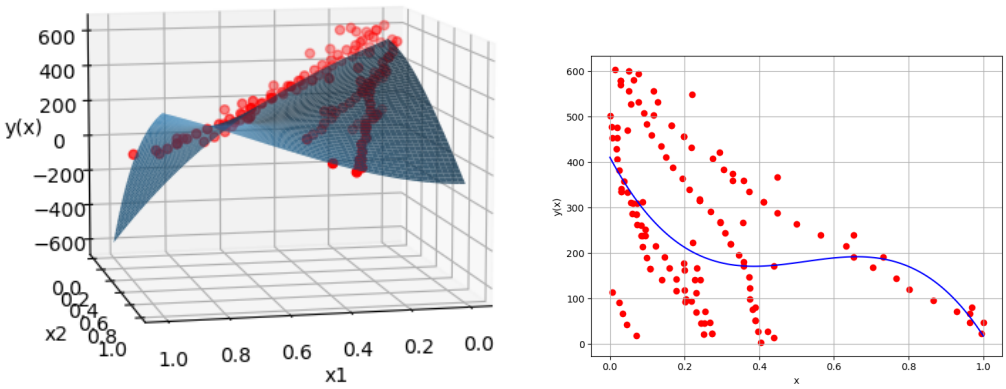

Figura 3.7: Modello 3D e 2D della funzione definita sui dati di train del Dataset 2

Capitolo 4

Test e analisi dei risultati

In questo capitolo, verrà effettuata un'approfondita valutazione delle prestazioni dei modelli implementati, analizzando attentamente i risultati ottenuti durante la fase di test. L'obiettivo principale è determinare l'efficacia e l'adattabilità di ciascun modello di machine learning rispetto all'ambito specifico del problema trattato. Attraverso l'utilizzo di metriche appropriate e analisi approfondite, verranno esplorate le capacità predittive dei modelli, valutando anche le loro potenziali limitazioni.

4.1 Risultati sul Dataset 1

Per quanto riguarda i risultati dei modelli applicati al primo dataset è importante sottolineare che, a causa della mancanza riguardante la tempistica di guasto dei componenti non guasti, non è stato possibile valutare le performance dei modelli in modo oggettivo. Tuttavia, si è effettuata una valutazione approssimativa dei risultati basata sulle ore di lavorazione già registrate.

Di seguito, nelle Figure 4.1 e 4.2, sono state riportate le predizioni ottenute dai due modelli.

Per quanto riguarda le probabilità di sopravvivenza dei vari mandrini, queste sono state estratte direttamente dalle probabilità di sopravvivenza calcolate. Un esempio in Figura 4.3.

Nonostante non sia possibile trarre conclusioni definitive sulla superiorità di un modello rispetto all'altro, si può osservare che i risultati ottenuti dalla distribuzione di Weibull sembrano avvicinarsi maggiormente alle aspettative reali dei dati rispetto all'algoritmo SVR. Questa somiglianza apparente potrebbe far sorgere l'ipotesi che la distribuzione si adatti meglio a tali dati rispetto al modello non parametrico. Tuttavia, è essenziale sottolineare che questa osservazione è solo qualitativa e non costituisce una valutazione quantitativa delle prestazioni dei modelli.

	Predizione	Reale	Prob. sopravvivenza
0	2675,80	1240,24	1,00
1	1166,31	706,78	1,00
2	1460,39	632,49	1,00
3	1401,58	731,70	1,00
4	284,10	223,08	1,00
5	1693,09	753,79	1,00
6	885,43	597,96	1,00
7	936,57	660,33	1,00
8	3212,66	658,31	1,00
9	41790,49	1787,61	1,00
10	255,27	364,30	0,00
11	452,25	357,43	1,00
12	512386,35	1982,66	1,00
13	11511064,20	3429,12	1,00
14	1435,72	711,58	1,00
15	16681,49	1312,52	1,00
16	406,76	408,29	0,25
17	1323,00	808,04	1,00
18	308,36	287,44	1,00
19	218,08	325,94	0,00

Figura 4.1: Risultati distribuzione di Weibull su Dataset 1

	Predizione	Reale
0	28772,22	1240,24
1	1555,47	706,78
2	5204,80	632,49
3	1687,18	731,70
4	407,17	223,08
5	1458,47	753,79
6	873,40	597,96
7	825,26	660,33
8	370,04	658,31
9	13887,20	1787,61
10	400,12	364,30
11	407,64	357,43
12	6017,07	1982,66
13	473801,17	3429,12
14	4782,33	711,58
15	9696,05	1312,52
16	402,82	408,29
17	820,13	808,04
18	405,91	287,44
19	405,91	325,94

Figura 4.2: Risultati SVR su Dataset 1

4.2 Risultati sul Dataset 2

In questo dataset invece, è stato possibile valutare le performance dei modelli in modo oggettivo utilizzando alcune metriche ampiamente impiegate in tasks di regressione. Più basso è il valore di una metrica, migliore è la performance del modello. Le misure impiegate sono:

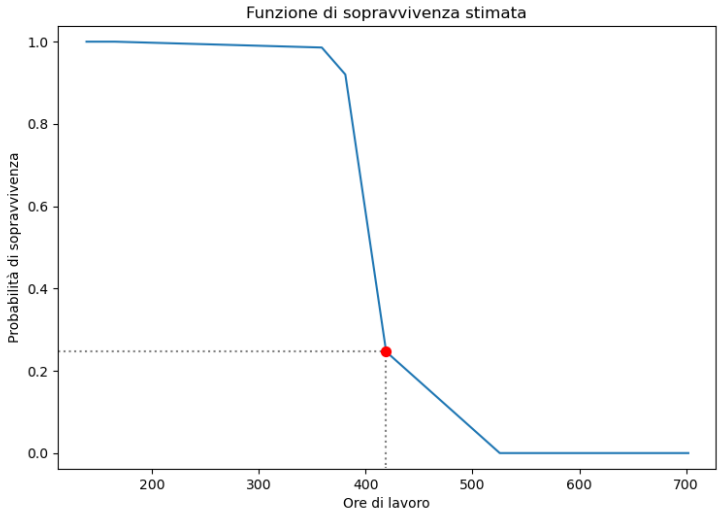

Figura 4.3: Funzione di probabilità di sopravvivenza associata ad un'istanza di test

- MAE (Mean Absolute Error): è la media aritmetica degli errori assoluti. Misura la differenza assoluta tra i valori predetti e i valori reali.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.1}$$

- MSE (Mean Squared Error): è la media aritmetica dei quadrati degli errori. Misura la differenza quadratica tra i valori previsti e i valori effettivi.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.2}$$

- RMSE (Root Mean Squared Error): è la radice quadrata della media aritmetica dei quadrati degli errori, ossia la radice quadrata dell'MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.3}$$

- SSE (Sum of Squared Errors): è la somma dei quadrati degli errori, il che significa che è espressa nelle stesse unità dei valori previsti e dei valori effettivi.

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.4}$$

Nell'ambito della valutazione dei modelli di apprendimento automatico, le metriche MAE, MSE, RMSE e SSE sono state stimate utilizzando la tecnica *LeaveOneOut*. La scelta di questa tecnica deriva dalla dimensione relativamente piccola del dataset,

il che permette una stima più precisa delle varie metriche impiegate. Tale tecnica consiste nel rimuovere un campione dal dataset e nel costruire un modello utilizzando i campioni rimanenti. Il modello viene quindi utilizzato per prevedere il valore del campione rimosso. Questo processo viene ripetuto per tutti i campioni del dataset. Le metriche MAE, MSE, RMSE e SSE vengono quindi calcolate di conseguenza utilizzando i valori previsti e i valori reali. In generale il metodo del *LeaveOneOut* è molto efficace per ridurre il sovra-adattamento del modello e garantire maggiore variabilità, ottenendo di conseguenza delle stime più robuste.

Di seguito i risultati riportati in tabella.

Metrica	**Weibull**	**SVR**
MAE	67.01	45.85
MSE	11079.34	6553.59
RMSE	105.26	80.95
SSE	1551107.64	917503.29

Tabella 4.1: Performance dei modelli Weibull e SVR

Come si puoi vedere, il modello SVR sembra essere migliore per la previsione dei valori rispetto al modello parametrico. Inoltre, è importante considerare che le metriche di valutazione da sole potrebbero non fornire una valutazione completa delle prestazioni del modello. Altri fattori come la robustezza del modello rispetto a dati mancanti o rumorosi e la complessità computazionale possono influenzare la scelta del modello più adatto per uno specifico scenario.

Inoltre, solo per fini illustrativi, il dataset in questione è stato diviso in due sottoinsiemi casuali, uno per l'addestramento e uno per il test. Questo ha permesso di valutare le prestazioni dei due modelli impiegati, vedendo nel dettaglio quali predizioni erano in grado di fare. Tali risultati sono stati riportati in 4.4.

Tuttavia, è importante notare che questi risultati sono basati su un singolo dataset e che i risultati potrebbero essere diversi su altri campioni di dati più rappresentativi. Per testare questa ipotesi, i modelli addestrati su questo campione sono stati applicati al test set di dati no-fail del precedente dataset.

I risultati prodotti dai due modelli, riportati in Figura 4.5, non sono valutabili oggettivamente in quanto non sono disponibili dati di riferimento. Tuttavia, è possibile notare che il modello SVR produce previsioni negative, il che suggerisce che non è in grado di generalizzare su componenti non guasti con caratteristiche differenti da quelli su cui è stato addestrato.

	Prev. Weibull	Prev. SVM	Reale
0	468,88	472,04	479,82
1	84,66	48,26	45,55
2	296,01	380,75	387,55
3	113,82	173,93	178,21
4	99,55	125,97	123,55
5	212,57	271,80	239,82
6	145,21	199,28	166,93
7	622,50	498,65	114,55
8	76,37	41,63	80,00
9	129,57	152,18	165,55
10	141,44	195,83	191,82
11	265,59	306,12	309,55
12	91,71	68,66	69,55
13	87,95	75,47	71,82
14	202,24	265,30	309,55
15	244,52	308,74	238,93
16	231,78	306,10	263,82
17	401,18	440,90	483,55
18	66,41	6,85	51,55
19	142,50	229,42	166,93
20	213,37	267,13	267,55
21	97,87	71,58	98,41
22	184,54	221,45	213,55
23	411,14	436,68	431,82
24	242,22	304,82	285,55
25	437,64	461,57	507,55
26	152,36	143,35	141,55
27	599,54	531,46	579,55

Figura 4.4: Risultati dei modelli su Dataset 2

	Pred. ore rimanenti Weibull	Pred. ore rimanenti SVR	Ore lavoro svolte
0	168963454,16	5675,78	1240,24
1	211,73	-494,96	706,78
2	5,19	151,25	632,49
3	62,65	-341,29	731,70
4	24,32	-651,36	223,08
5	65,91	-116,49	753,79
6	10,35	-844,78	597,96
7	12,25	-866,11	660,33
8	4845,87	-62,28	658,31
9	15301,52	1397,25	1787,61
10	219421,38	-120,22	364,30
11	125,82	-445,53	357,43
12	30841,12	-1929,55	1982,66
13	280184066622242000,00	480,02	3429,12
14	6,46	355,28	711,58
15	60,21	-678,28	1312,52
16	1447,39	199,72	408,29
17	46,50	-424,01	808,04
18	1502,64	378,58	287,44
19	1389,04	664,93	325,94

Figura 4.5: Risultati dei modelli, addestrati su Dataset 2, applicati ai componenti non guasti

Capitolo 5

Conclusione

In conclusione, il presente progetto si proponeva di affrontare la sfida di stimare il tempo di vita rimanente e la probabilità di sopravvivenza degli elettromandrini, componenti fondamentali nel settore industriale. Tale obiettivo è stato realizzato esaminando diverse tecniche di apprendimento automatico.

Nonostante le limitazioni dovute alla scarsità di dati disponibili, l'impiego di modelli parametrici ha dimostrato coerenza e trattabilità dei risultati ottenuti. Questi modelli, basati su precise assunzioni sulla distribuzione dei dati come la Weibull trattata, sono stati in grado di fornire stime attendibili su più fronti, fornendo informazioni utili e di maggior supporto. I risultati ottenuti possono essere utilizzati per migliorare la comprensione dei fenomeni studiati e per sviluppare strategie di intervento più efficaci.