Università degli studi di Milano-Bicocca

Industry Lab

Progetto finale

Bosch VHIT: Caso di studio 1

Analisi dati raccolti dalla linea di montaggio MVP11

Autori:
Matteo Gaverini - 808101 - m.gaverini1@campus.unimib.it
Matteo Licciardello - 799368 - m.licciardello@campus.unimib.it
Paolo Mariani - 800307 - p.mariani20@campus.unimib.it

Luglio 2020

Sommario

I dati ottenuti dai sensori o, in generale, da un qualsiasi apparecchio elettronico in un sistema di produzione possono essere utili, oltre a determinare la validità di un prodotto, a capire se esistono eventuali anomalie nel processo produttivo. La capacità di un'azienda di creare valore dai dati generati può risultare quindi un fattore di competitività molto importante; a tal ragione sempre più imprese manifatturiere e meccaniche, come Bosch, sfruttano ed utilizzano tecnologie volte a generare conoscenza e valore attraverso la *Data Science*. Nel progetto che viene presentato si svolge dapprima un'analisi delle eventuali relazioni tra i dati registrati da Bosch, in un certo intervallo temporale, provenienti da una linea di montaggio di una pompa meccanica; successivamente si risolve un problema di classificazione binario utile a determinare se un prodotto è conforme alle specifiche o meno.

N.B. *per motivi di struttura è stato deciso di inserire le immagini nell'Appendice, si invitano i lettori a sfruttare i riferimenti per visionare le immagini. Alcune di esse saranno fornite in una cartella esterna al report per motivi di impaginazione, saranno citate con il nome corrispondente.*

Introduzione

Bosch è un'azienda multinazionale tedesca il cui business principale è quello di produrre componenti per autovetture, ma oltre a questo possiede quote di mercato anche in altri settori quali *Industrial Technology*, *Consumer Goods* e *Energy and Building Technology* [1]. Fondata nel 1886 da Robert Bosch, oggi l'azienda conta circa 460 società filiali in 60 Paesi del mondo dove lavorano più di 400 mila dipendenti [2]. Bosch possiede diverse quote di mercato e risulta essere *leader* in diversi Paesi, questo è dato, oltre ad un attenta ed equilibrata espansione economica, anche dalla capacità dell'azienda di creare valore dai dati generati nei diversi processi produttivi attraverso la *Data Science*: molti processi di assemblaggio sono stati ottimizzati grazie a tecniche di *Machine Learning* che hanno reso possibile una diminuzione drastica dei costi di produzione ed un aumento consistente dell'efficacia del prodotto realizzato, come è stato illustrato dal Dott. Sesini di Bosch in una presentazione nella quale spiega le principali applicazioni della *Data Science* adottate dall'azienda (link). Il progetto prevede 2 task utilizzando un singolo dataset realizzato da Bosch, riguardante un periodo temporale di 5 anni, in cui sono registrate un insieme di variabili fisiche estratte dalle 6 diverse stazioni che formano la linea di montaggio *MVP11* per la produzione di una pompa del vuoto (componente dell'impianto frenante di veicoli pesanti). Nel primo task viene svolta un'analisi delle relazioni tra le variabili più rilevanti attraverso alcune regressioni, il secondo task invece consiste nel risolvere un problema di classificazione binario. Quest'ultima operazione, a differenza delle prime due che erano implicitamente richieste da Bosch, è stata svolta in seguito ad un interesse maturato da parte del referente aziendale, ovvero il Dott. Federico Astori, di determinare in modo automatico la conformità o meno di una pompa del vuoto senza svolgere concretamente alcun collaudo.

1 Processo Produttivo

Il caso di studio viene applicato da Bosch nello stabilimento *automotive* VHIT di Offanengo (CR), esso rappresenta un intero sistema di produzione composto da apparati differenti (risorse

umane, macchinari e attrezzature combinate in maniera organizzata) che partecipano alla produzione di pompe idrauliche, meccaniche ed elettriche da destinare al mercato dei mezzi pesanti.

La modalità di produzione basa il proprio funzionamento sulla logica *PULL* (l'ingresso in produzione dei prodotti è posteriore alla richiesta del cliente e non anticipato rispetto agli ordini) tipica dell' *Assembly To Order* (ATO), procedura che stabilisce di iniziare la produzione solo nel momento in cui vi è una singola richiesta dell'utente o un numero minimo di richieste da soddisfare. In tale ottica risulta fondamentale la preparazione del sistema di produzione all'attività che dev'essere svolta, considerando la stagionalità delle richieste dei clienti e l'allestimento dell'infrastruttura in modo tale che sia pronta per soddisfare gli ordini. Per la linea di produzione considerata tutti i componenti assemblati provengono da fornitori che devono adeguarsi alla logica *PULL* e il loro assemblaggio avviene con un processo continuo e tecnicamente obbligato.

Basando il funzionamento sui principi di suddivisione ed analisi del lavoro introdotto da Taylor [3], da tale sistema di produzione è possibile catturare delle enormi moli di dati adatti all'ottimizzazione del processo produttivo. In questo contesto l'uomo diventa parte integrante del sistema, in quanto è fondamentale che svolga un ruolo di supervisione del processo di produzione e del risultato che esso genera: ciascun operatore abilita il funzionamento del macchinario designato e valuta il risultato ottenuto grazie a specifiche misurazioni, determinando così se il prodotto sia conforme ai parametri ingegneristici stabiliti in fase di progettazione oppure se sia necessario svolgere un'ulteriore lavorazione o, ancora, lo scarto del prodotto. Nonostante la presenza dell'operatore che garantisce un immediato riscontro sul risultato della produzione, si considera vantaggioso svolgere delle analisi approfondite dei dati generati dal sistema interconnesso IoT [4] per poter valutare quali siano le caratteristiche di ciascun prodotto che ne determinano la conformità o difformità, per ogni processo dell'assemblaggio. Ciò è particolarmente importante perché, trattandosi di un componente critico per la sicurezza stradale dei mezzi pesanti, risulta fondamentale per Bosch evitare ogni possibile tipo di errore in fase di produzione: tali problemi comporterebbero grosse conseguenze ai clienti e quindi l'azienda dovrebbe attivare poi delle campagne di richiamo costosissime in termini contrattuali.

Per quanto riguarda il progetto, il caso preso in esame risulta essere la linea di produzione MVP11: questa viene impiegata per l'assemblaggio di pompe del vuoto (o *vacuum pumps*) utilizzate negli impianti frenanti servo-assistiti per amplificare la forza che viene esercitata sul pedale del freno. La specifica pompa del vuoto prodotta dalla linea di produzione MVP11 è composta dai seguenti componenti (Figura 1) da assemblare tra loro:

- Corpo pompa in alluminio pressofuso a camera calda (in cui la temperatura dello stampo è controllata dalla fornace) che funge da sede per tutti gli altri componenti;
- Filtro e guarnizione sostenute da un sistema di fissaggio composto da lamelle in acciaio forato;
- Valvola di non-ritorno (VNR) assemblata nell'apposito alloggio direttamente nella cavità superiore del corpo principale;
- Gruppo rotore-giunto-lamierino proveniente da una stazione esterna alla linea di produzione analizzata;

- Paletta con corpo in resina (materiale liquido polimerico sottoposto ad elevate temperature e pressioni che ne causano un irreversibile indurimento) e condotti in termoplastica;
- Coperchio di chiusura del corpo principale in acciaio.

L'assemblaggio di tali componenti avviene tramite l'impiego di macchinari specifici allestiti in modo da formare dei tavoli industriali (Figura 2), i quali permettono di svolgere una specifica operazione effettuata singolarmente da un operatore. Di seguito, viene riportato lo schema dettagliato delle operazioni eseguite per ciascuna stazione:

1. *ST10*: Posizionamento, da parte dell'operatore, dei componenti corpo, lamella, controlamella, vite-lamella. Il macchinario verifica la presenza e la posizione dei suddetti componenti e mediante l'automazione installata procede con l'avvitatura della vite e con la cianfrinatura del filtro. Prima di procedere alla stazione successiva viene oliata la sede della VNR.
2. *ST20*: L'operatore inserisce la VNR in un sistema di aspirazione, successivamente l'automazione della stazione posiziona la VNR e la vite di sostegno sul corpo pompa. Viene anche eseguito un test funzionale mediante flussaggio della valvola stessa.
3. *ST30*: Stazione stand-alone che assembla il gruppo rotore-giunto-lamierino; è esterna alla linea di produzione.
4. *ST40*: Un operatore posiziona il gruppo rotore-giunto-lamierino ed assembla la paletta, vengono avvitati l'O-ring e il coperchio al corpo pompa che viene serrato tramite viti. Infine viene verificata la rotazione del sistema all'interno del corpo.
5. *ST50*: Dopo aver terminato l'assemblaggio della pompa si verifica la tenuta del prodotto in un dato intervallo di tempo effettuando un collaudo pneumatico (viene immessa aria e si chiude l'interfaccia della pompa).
6. *ST60*: Stazione atta al collaudo funzionale, la pompa viene messa in rotazione a diverse velocità misurandone la depressione generata, la coppia e la tenuta della VNR.
7. *ST70*: Stazione conclusiva in cui si appone la marcatura DMC (definita lunga la linea aggiungendo delle porzioni di codifica stazione dopo stazione) per tracciare il componente.
8. *ST80*: Stazione in cui avviene l'imballaggio del prodotto per essere spedito successivamente al cliente.

2 Analisi del dataset

Il dataset fornito da Bosch contiene tutti i dati registrati alla linea di produzione MVP11 dal 11/12/2015 al 29/4/2020, riguardo ad ogni singolo item prodotto durante la giornata lavorativa. Le variabili considerate sono relative a misure che vengono valutate ad ogni stazione e rappresentano parametri fisici di interesse che vengono confrontati con degli intervalli di accettazione stabiliti secondo studi ingegneristici. Oltre a queste misure fisiche, sono contenute anche informazioni specifiche per ogni item come la marcatura e l'ultima stazione raggiunta. Si procede ad elencare le variabili fornite in Tabella 1:

Tabella 1: Attributi dataset

Variabile	Descrizione	Intervallo
DMC	Marcatura del singolo prodotto	-
UltimaStazione	Stazione in cui il prodotto esce dalla linea di produzione	-
Data_Ingresso	Data e ora inizio lavorazione (formato YYYY-MM-DD HH:MM:SS)	-
Data_Uscita	Data e ora fine lavorazione (formato YYYY-MM-DD HH:MM:SS)	-
Esito_S*(*10, 20, 40, 50, 60*)	Esito ottenuto dalla verifica delle misurazioni	0 - Stazione non raggiunta 1 - Stazione completata correttamente 2 - Stazione non completata correttamente 3 - Stazione saltata 4 - Rilavorazione del prodotto
S*(*10, 20, 40Vite1, 40Vite2, 40Vite3*)*Angolo	Angolo di avvitatura delle viti nelle varie stazioni	[0, 50]
S10Coppia	Coppia di serraggio della vite nella stazione S10	[4, 4.5]
S20Coppia	Coppia di serraggio della vite nella stazione S20	[6, 8]
S20Portata	Portata in fase di controllo tramite flussaggio della valvola di non ritorno	[13.4, 15.5]
S40F2MomentoTorcMax	Momento torcente generato dal sistema giunto-paletta	[-0.1, 0.6]
S40Vite*(*1, 2, 3*)*Coppia	Coppia di serraggio delle viti 1, 2 e 3 nella stazione S40	[7, 9]
S50PressionePT	Pressione ottenuta in fase di collaudo pneumatico	[900, 1150]
S50TenutaPZ	Tenuta per un dato intervallo di tempo in cui viene eseguito un tappaggio dell'interfaccia della pompa	[-∞, 2.4]
S60F2DepresMin	Depressione ottenuta in fase di collaudo funzionale	[330, ∞]
S60F2Coppia	Coppia ridondante rispetto alla libera rotazione misurata durante il test funzionale	[0.05, 4]
S60F2Velocita	Velocità di rotazione raggiunta dalla pompa durante il test funzionale	[390, 410]
S60F2TenutaVNR	Tenuta della valvola di non ritorno in fase di collaudo funzionale	[$-\infty$, 3]

Il dataset si compone di 1.294.504 osservazioni di cui si elimina una piccola porzione secondo le seguenti condizioni:

- Osservazioni con esiti "3" e "4" registrati nelle variabili *Esito_S**, poiché rappresentano i casi in cui il prodotto non fa registrare alcun valore di misurazione o perché rappresenta un item che viene rilavorato a seguito di una considerazione svolta da un operatore tramite PLC su cui non si dispongono ulteriori informazioni;
- Osservazioni la cui variabile *Esito_S10* risulta assumere valore "0" (stazione non raggiunta);
- Osservazioni la cui variabile *UltimaStazione* assume valore "70" che però possiedono una variabile *Esito_S** con valore "2" (non conforme) in una delle stazioni precedenti.

A seguito dell'eliminazione di queste osservazioni il dataset risulta composto da 1.277.959 istanze diverse. Sono state introdotte 4 ulteriori variabili, riportate in Tabella 2:

Tabella 2: Nuovi attributi

Variabile	Descrizione	Intervallo
Data_Ingresso_day	Giorno in cui si produce l'item (estratto dalla variabile *Data_Uscita*)	[1, 31]
Data_Ingresso_month	Mese in cui si produce l'item (estratto dalla variabile *Data_Uscita*)	[1, 12]
Data_Ingresso_year	Anno in cui si produce l'item (estratto dalla variabile *Data_Uscita*)	[2015, 2020]
Success	Variabile di controllo per gli esiti (non presente nel dataset originale)	0 - il prodotto non raggiunge la stazione 70 o raggiunge la stazione 70 con esiti non compatibili con gli intervalli di accettazione 1 - il prodotto raggiunge la stazione 70 ed è conforme

L'analisi è presentata nel notebook Analisi e Preprocessing.
Una volta introdotte le nuove variabili, si è potuto determinare la percentuale di scarti rispetto agli item validi; come evidenziato dal bar-chart (Figura 3), si osserva una forte disparità dei prodotti conformi (96.8%) rispetto a quelli non conformi (3.2%). Un'ulteriore analisi effettuata, riguarda il conteggio dei fallimenti che coinvolgono ogni stazione, il cui risultato si può visionare con un plot (Figura 4). Dal bar-chart illustrato è evidente come la maggior parte degli item riscontra dei problemi di conformità già dalla stazione *ST10*, con un elevato numero di scarti prodotto anche dalla stazione *ST40*.
Successivamente, vengono analizzati anche i valori assunti dalla variabile *UltimaStazione* rappresentati graficamente da un bar-chart (Figura 5) in cui si mostra il conteggio di item che escono dalla linea di produzione per ciascuna stazione, indipendentemente dall'esito. Come già evidenziato in precedenza, la gran parte dei prodotti è conforme agli standard posti in fase di progettazione per cui la frequenza maggiore si ha per i valori della stazione *ST70*; risulta invece interessante l'analisi della seconda frequenza più alta, ovvero la frequenza del valore "0", questa viene spiegata dal referente aziendale come una non conformità dei componenti da assemblare, tipicamente segnalata dall'operatore presente sul posto, già prima che questi entrino nella linea di produzione alla stazione *ST10*.
La fase di analisi preliminare si conclude con alcune considerazioni relative alle distribuzioni dei prodotti, sia conformi che non, rispetto alle variabili *Data_Ingresso_day*, *Data_Ingresso_month* e *Data_Ingresso_year* che rappresentano rispettivamente giorno, mese e anno relativi alla produzione del singolo item. Per far ciò si producono diversi grafici (presenti nella cartella "*Distribuzione variabili Dataset/Distribuzioni temporali*") che mostrano, in generale, sia l'andamento della produzione che una distribuzione dei relativi esiti (in verde i successi, in blu gli scarti).
Si procede infine a considerare due misure fondamentali nell'analisi dei dati: l'indice di correlazione di Pearson [5] e la covarianza. Queste misure sono calcolate sull'intero dataset originale, considerando anche la variabile *Success*. Osservando la *heatmap* che rappresenta i valori di

correlazione (Figura 6), emergono due aspetti interessanti: il primo è che *S20Portata* risulta essere molto correlata con *S50PressionePT* (0.81) e *S60F2Velocita* (0.73), il secondo aspetto rilevante è che la variabile *Esito_S60* risulta essere altamente correlata con *S50PressionePT* (0.84): ciò significa che potrebbe esserci un'influenza di quest'ultima sull'esito della stazione *ST60* e che verrà successivamente analizzata.

3 Approccio al problema

Si procede ad illustrare l'approccio utilizzato per effettuare i due diversi task finalizzati ai seguenti obiettivi:

- Valutare l'esistenza di particolari relazioni di interesse attraverso analisi di regressione tra le variabili del dataset.
- Determinare per ogni prodotto, la conformità o meno rispetto alle specifiche di produzione alle stazioni *ST50* e *ST60* attraverso tecniche di *Machine Learning*.

Prima di proseguire con la fase di pulizia del dataset, si è voluto svolgere un'analisi approfondita dei dati per incrementare ulteriormente la conoscenza. A tal proposito si è intrapreso un dialogo con il referente aziendale al fine di ottenere risposte precise sull'assegnazione dei valori per gli attributi: da questo è emerso che i valori negativi per alcune specifiche variabili (ad esempio tutte le variabili relative a coppia, angolo, velocità o pressione), non sono plausibili in quanto la grandezza fisica rappresentata da queste feature non può assumere valori minori di 0. Tale comportamento, attribuito ad un errore nella comunicazione del dato da parte del PLC, comporta ulteriori difficoltà nello svolgere un'analisi statistica corretta pertanto il dato viene ritenuto errato e quindi eliminato. Un'ulteriore considerazione, supportata da test statistici (*Shapiro-Wilk* [6], *Kolmogorov-Smirnov* [7], *Skewness Test* [8]), pone in evidenza come tutti gli attributi del dataset non possano essere considerati normalmente distribuiti: in generale, si nota che per tutte le variabili c'è una forte tendenza dei valori nello stare all'interno dell'intervallo di accettazione, tuttavia si evidenzia anche una presenza considerevole di *outlier* che tendono a manifestarsi intorno al valore zero o assumono valori negativi che potrebbero quindi essere causa della non normalità rilevata. Per il trattamento degli *outlier* si è fatto ricorso a tecniche multivariate in quanto quelle basate su analisi univariata, non riescono a catturare l'intera complessità del problema ed effettuano interventi troppo drastici sui dati. Le principali tecniche multivariate che si possono utilizzare per identificare gli *outlier* sono:

- *Local Outlier Factor* [9]: basa il suo funzionamento sulla considerazione di un punto (osservazione) e dei suoi k vicini (*neighbors*) nello spazio; si calcola, per ogni punto, una densità in funzione della distanza rispetto ai suoi k-neighbors, se risulta essere molto inferiore alla densità di altri punti/zone dello spazio si deduce che l'osservazione considerata sia un *outlier*.
- *Elliptic Envolve* [10]: consiste nel calcolo della matrice di covarianza su una piccola parte di dati per comprendere in che modo una o più variabili cambiano come conseguenza del variare di altre variabili; da questo è possibile produrre degli stimatori, robusti agli *outlier*, che vengono applicati sulla restante porzione di dati e classificano come

outlier ciascun dato che risulti eccedere una data soglia di distanza dalla distribuzione secondo una specifica metrica (es. *Mahalanobis Distance* [11]). L'utilizzo di questa tecnica suppone che la distribuzione sia normalmente distribuita una volta che gli *outlier* vengono esclusi da quest'ultima.

Tra le due soluzioni si è deciso di utilizzare la tecnica *Local Outlier Factor* per poter evidenziare gli *outlier* e procedere ad eliminarli dal dataset, accertandosi però che non assumessero valori interni all'intervallo di accettazione per le variabili considerate. La decisione di eliminare tali osservazioni è legata al fatto che risultano essere estremamente influenti per i compiti di regressione e classificazione, per cui si vuole avere un dataset pulito che garantisca esiti non alterati dai valori (estremi) delle relative istanze.

Per poter lavorare con i dati forniti, si è reso necessario eseguire una serie di test preliminari per verificare se l'insieme di dati risultanti dalla rimozione degli *outlier* contenesse features distribuite normalmente. Nonostante la tecnica utilizzata, le operazioni effettuate non risultano essere risolutive ed è quindi necessario applicare delle operazioni più stringenti relative alla selezione iniziale dei dati. Oltre a questo si evidenzia che alcune osservazioni ottengono la conformità alle specifiche nonostante presentino una o più misurazioni dei parametri che non rispettano i limiti stabiliti: queste violazioni sono state identificate e registrate tramite un algoritmo che verifica l'effettiva appartenenza di ciascuna variabile agli intervalli di accettazione introducendo delle nuove variabili categoriche chiamate **-INT*. Così facendo è possibile filtrare tutte le osservazioni erroneamente considerate conformi (per via del giudizio personale di un operatore o per una mancanza di automazione nel processo di rimozione del componente dalla linea). La nuova feature creata per ogni variabile, assume 3 valori così definiti:

- 1 quando il valore del parametro assunto è interno all'intervallo di accettazione;
- 0 quando il valore del parametro assunto è esterno ma è positivo;
- -1 quando il valore del parametro assunto è negativo ed è quindi causato dal PLC.

Filtrando i risultati errati, si ottiene un dataset privo di rumore che verrà utilizzato successivamente per il task di regressione. In seguito, è stata sviluppata una versione alternativa del dataset con l'aggiunta di variabili derivate chiamate **-RIS*: tali attributi rappresentano, per ogni misura, l'appartenenza di singoli valori ad un intervallo nel quale la probabilità che si verifichi uno scarto del prodotto risulta essere superiore. L'identificazione di questo intervallo è stato effettuato sfruttando un procedimento grafico basato sulla proporzione di prodotti scartati rispetto al totale delle osservazioni per ogni singolo valore osservabile. La procedura è stata svolta per le feature considerate fino alla stazione *ST50* e nuovamente ripetuta per le ulteriori variabili misurate prima della stazione *ST60*. Per comprendere meglio il procedimento dell'identificazione dell'intervallo si considera la variabile *S20Portata* e la relativa distribuzione riportata in Figura 7; come si può vedere dal grafico, si evidenzia che i valori della portata inferiori al valore 13.6 e quelli superiori a 15.3 sono più propensi a portare a scarti del prodotto, raggiungendo un indice (rappresentato dalla linea arancione) sempre superiore al 4% e talvolta superiore al 30%. Per le immagini relative alle altre variabili **-RIS* si invita il lettore a consultare la cartella presente al seguente percorso "*Distribuzione variabili Dataset/VariabiliRIS Stazione ST50*" e "*Distribuzione variabili Dataset/VariabiliRIS Stazione ST60*".

Per quanto riguarda il task di regressione, si sono creati sia modelli di regressione lineare che non lineare per spiegare le variabili ritenute più interessanti, ovvero quelle relative alle stazioni *ST50* e *ST60*. Passando a considerare il compito di classificazione, è stato deciso di applicare diversi approcci alla risoluzione del task che hanno coinvolto classificatori classici ed approcci basati su reti neurali. Oltre a questo, si è svolto uno studio di *Anomaly Detection* [12] tramite *Autoencoder* [13], *Isolation Forest* [14] e *Gaussian Distribution* [15]. Infine, per la classificazione che avrà senso approfondire ulteriormente (*ST60*), si utilizzano tecniche di *Interpretable Machine Learning* [16] per agevolare la comprensione del funzionamento del modello.

4 Sviluppo dei modelli

4.1 Regressione

Il compito di regressione è stato svolto considerando due diversi obiettivi: il primo consiste nello spiegare le variabili della stazione *ST50*, il secondo invece per le variabili della stazione *ST60*. I modelli che verranno presentati includono le variabili che il processo produttivo permette di raccogliere fino alla stazione di interesse, considerando come variabile dipendente una sola feature per volta della stazione in analisi. Si presenta nel Notebook Regressione il lavoro svolto per la regressione.
Ogni variabile della stazione *ST50* e *ST60* viene regredita inizialmente tramite un modello lineare, successivamente si cerca di migliorare il *fitting* del modello applicando diverse funzioni matematiche , sia alle variabili dipendenti che ai regressori. Per tutti i modelli lineari realizzati si verificano sei assunzioni molto importanti per decretarne la loro validità:

- Distribuzione attesa dei residui pari a zero, non deve esserci sistematicità.
- Omoschedasticità: la varianza degli errori del modello deve risultare costante, cioè devono essere caratterizzati da sfericità.
- Non autocorrelazione degli errori: l'errore che viene calcolato dal modello rispetto all'osservazione *i-esima* non deve essere correlato rispetto all'errore dell'osservazione precedente/successiva.
- Non collinearità: le variabili esplicative del modello non devono essere altamente correlate tra di loro, tali per cui una sia combinazione lineare delle altre.
- Normalità dei residui: i residui calcolati dal modello devono essere distribuiti normalmente.
- Assenza di osservazioni *outlier*, cioè quelle che si distanziano particolarmente dalla distribuzione e possono influenzare pesantemente il modello.

Considerando l'enorme influenza che alcune osservazioni divergenti rispetto alla distribuzione dei dati possono generare, si stabilisce che molti item del dataset debbano essere eliminati, in quanto causano un rumore che può influenzare negativamente la regressione. Questa procedura di eliminazione viene svolta imponendo alle osservazioni alcune condizioni importanti: le variabili misurate devono assumere valori interni agli intervalli di accettazione

(condizione verificata tramite le variabili **-INT* pari ad 1) ed esiti positivi riguardanti le lavorazioni delle stazioni precedenti, non vincolando a tali condizioni le variabili della stazione considerata. I dataset ottenuti possiedono quindi, rispettivamente, 1.251.948 osservazioni per la stazione *ST50* e 1.243.745 per la stazione *ST60*.

Ogni modello che viene stimato viene validato considerando le assunzioni precedentemente riportate, secondo l'ordine invertito rispetto all'elenco fornito; tutti i tentativi effettuati, anche considerando diverse combinazioni delle variabili esplicative e trasformazioni matematiche delle variabili dipendenti o dei regressori, hanno evidenziato la presenza di valori anomali che causano una non normalità nella distribuzione dei residui. Per correggere il problema, a ciascuna variabile dipendente considerata viene applicata un'analisi univariata degli *outlier* tramite *Z-score* [17] e *Inter-Quartile Range* [18] (IQR) tramite l'utilizzo di boxplot; ciò comporta l'eliminazione di ulteriori osservazioni per entrambi i task di regressione in svolgimento, avendo così un totale di 1.245.311 osservazioni per il dataset della stazione *ST50* e 1.241.450 per la stazione *ST60*. Nonostante le misure preventive intraprese per evitare il problema della non normalità dei residui, questo si è presentato ugualmente per tutti i modelli stimati. Per risolvere tale problema si è applicata una trasformazione matematica al modello, in particolare si fa riferimento al procedimento di *Box-Cox* [19] descritto dalla Formula 1:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log y, & \text{if } \lambda = 0 \end{cases} \tag{1}$$

Da questa formula si evince che la trasformazione dipenda dal valore *lambda*, ovvero il valore che indica la tipologia di trasformazione matematica da attuare, che viene selezionato automaticamente secondo la misura di verosimiglianza (*Log-likelihood*).

4.2 Regressione sulle variabili ST50

Per ogni variabile dipendente che viene spiegata con i modelli di regressione, è stata scelta la combinazione delle variabili esplicative che ha garantito il risultato migliore: in tutti i casi considerati nella stazione *ST50* essa rappresenta l'insieme globale delle misure disponibili prima della stazione considerata. La regressione utilizzata per stimare la relazione tra la variabile *S50TenutaPZ* e i suoi regressori viene sviluppata a partire dalla funzione lineare, successivamente trasformata in logaritmica (applicata alternativamente alla variabile dipendente ed alle variabili esplicative), tuttavia nessuna delle due trasformazioni è in grado di risolvere i problemi relativi alle assunzioni dei modelli. È stata quindi applicata la trasformazione *Box-Cox* che non ha portato ad alcun miglioramento: permane il problema della non normalità dei residui che non rende possibile decretare alcun tipo di relazione che spieghi i rapporti tra le variabili. Si ipotizza che una rimozione di ulteriori valori finora non considerati *outlier* possa aiutare a migliorare l'esito, tuttavia tale operazione eliminerebbe osservazioni conformi alle specifiche di produzione e quindi non viene svolta.
Ugualmente alla variabile appena considerata, le problematiche si ripetono per la variabile *S50PressionePT*: in tutti i modelli stimati (versione lineare, con applicazione del logaritmo alle variabili esplicative, con l'elevamento a potenza fino al terzo grado della variabile dipendente) ed applicando la trasformazione *Box-Cox* non vi è alcun cambiamento. Si deduce che non è possibile spiegare il rapporto tra le variabili considerate, decretando che il problema

risieda nella natura dei dati e, in particolare, che l'effetto esplicativo rispetto alle due variabili considerate sia catturato da feature non presenti nel dataset.

4.3 Regressione sulle variabili ST60

Il procedimento per la stima del modello applicato precedentemente alla stazione *ST50* viene replicato per le variabili della stazione *ST60*. Si osserva il ripetersi delle problematiche considerate alla stazione precedente, questa volta le variabili dipendenti sono: *S60F2Coppia*, *S60F2Velocita* e *S60F2TenutaVNR*. Come per la stazione *ST50*, la combinazione di regressori che ha garantito i risultati migliori è l'insieme complessivo di variabili che si registrano prima della stazione in esame. Il tipo di regressione che ha prodotto risultati migliori per tutte e tre le variabili dipendenti risulta essere quello lineare, nonostante siano state sperimentate diverse versioni non lineari caratterizzate dall'applicazione di logaritmi, radici e potenze: l'applicazione di queste trasformazioni, inclusa quella *Box Cox*, non ha risolto il problema di mancanza della normalità dei residui, quindi nessuna regressione è in grado di spiegare efficacemente la relazione tra le variabili dipendenti e i regressori utilizzati.

Valutando invece i residui della variabile *S60F2DepresMin*, si nota come questi siano distribuiti secondo una forma particolarmente simile a quella di una curva gaussiana (Figura 8), questa ipotesi trova riscontro nei test statistici *Kolmogorov-Smirnov* e *Shapiro-Wilk* che, pur non rifiutando l'ipotesi nulla di non normalità, assumono valori molto vicini a tale soglia. Si decide pertanto di assumere la presenza della normalità e di proseguire cercando di risolvere i successivi problemi. In seguito al test di *Durbin-Watson* [20], applicato per valutare la presenza di autocorrelazione dei residui, si deduce che per risolvere tale problematica è richiesta l'applicazione di un modello *Generalized Least Squares* (GLS [21]) per stimare i parametri sconosciuti. Tuttavia questo approccio richiede la creazione della matrice di *Toeplitz* [22] di dimensioni $n * n$ con n pari al numero di osservazioni del dataset su cui viene applicato il modello per cui, nel caso preso in esame, questa matrice risulta avere una dimensione che non può essere computazionalmente elaborata.

Il compito di regressione non risulta portare alcuna informazione aggiuntiva, la motivazione è dovuta principalmente ad un problema di scarsa capacità delle variabili a disposizione di spiegare le relazioni presenti tra di esse. Si considera quindi la presenza di *Omitted Variable Bias* [23] dovuto all'assenza di una o più variabili esplicative mancanti nel dataset che potrebbero essere incluse tramite delle misurazioni aggiuntive.

4.3.1 Test con matrice di Toeplitz

Con lo scopo di approfondire ulteriormente i risultati ottenuti per la regressione della variabile *S60F2DepresMin*, si propone un approccio basato sul campionamento del dataset in modo tale da poter calcolare la matrice di *Toeplitz*. In primis, è stato necessario campionare il dataset raggiungendo il numero massimo di osservazioni consentite per il calcolo della matrice, pari a 500, tramite il metodo *resample* di *Sci-Kit Learn* [24]. Considerando le stesse variabili esplicative del modello presentato nella Sezione 4.3, si ripropone una nuova stima di questo modello con meno osservazioni, riscontrando dai risultati la presenza di non normalità dei residui. Il problema viene risolto applicando la trasformazione di *Box-Cox* che produce un modello caratterizzato da residui distribuiti in maniera approssimabile ad una distribuzione normale.

Si considera perciò soddisfatta l'ipotesi di normalità, ma si osserva che si verificano problemi di eteroschedasticità e autocorrelazione dei residui. Il modello di regressione che si deve applicare in questo contesto è il modello *Generalized Least Squares* descritto in precedenza, che richiede di calcolare la matrice di Toeplitz. Il dataset ora considerato possiede un numero di osservazioni che rende trattabile tale matrice: si procede quindi al calcolo del parametro *Sigma* che permette di stimare correttamente il modello. Dal risultato emerge che, con un numero estremamente ridotto di osservazioni, l'esito è in linea con quanto ottenuto sull'intero dataset: il *p-value* ottenuto dal test di normalità di *Jarque-Bera* non risulta influenzato dalla numerosità delle osservazioni, pertanto si considera che l'ipotesi di normalità non sia soddisfatta. Tuttavia, come già considerato in precedenza, si procede a svolgere ugualmente l'analisi dei residui relativamente ad eteroschedasticità ed autocorrelazione: nonostante l'applicazione del modello *GLS* dovesse risolvere i problemi elencati, non si ottiene il risultato sperato in quanto il test di *Durbin-Watson* non evidenzia alcun miglioramento.
Avendo già assunto verificata l'ipotesi di normalità, quando in realtà il test dedicato evidenzia il contrario, considerando i due problemi appena citati legati alla sfericità degli errori, si conclude l'analisi stabilendo che anche il modello *GLS* appena stimato non è in grado di spiegare la variabile *S60F2DepresMin* rispetto alle variabili esplicative considerate.

4.4 Classificazione

Il compito di classificazione è stato svolto per valutare se fosse fattibile identificare la conformità di un prodotto, senza svolgere alcun test pneumatico o funzionale ma sfruttando unicamente un approccio basato su *Machine Learning*. Questo compito è stato concordato con il referente Astori che ha indicato come possibili stazioni *target ST50* e *ST60*, che comprendono i macchinari atti allo svolgimento dei collaudi. Questi strumenti sono estremamente dispendiosi in termini economici, tali per cui potrebbe essere remunerativo sostituirli con metodi più economici ma che non inficino sull'affidabilità del prodotto finale. Si procede a svolgere la classificazione per l'esito delle due stazioni considerate. Nel Notebook Classificazione si presenta il lavoro svolto per risolvere i task descritti in questa Sezione.

Il dataset utilizzato per svolgere i due task di classificazione comprende gli attributi " **-INT*" e " **-RIS*", come già introdotto precedentemente nella Sezione 3. Si stabilisce che le osservazioni considerate per il task di classificazione alla stazione *ST50* sono quelle che rispettano i vincoli ingegneristici e sono conformi al momento del raggiungimento della stazione, ugualmente per il task di classificazione della stazione *ST60* ma comprensivi anche delle variabili della stazione precedente. Infine, per entrambi i compiti affrontati, è stata applicata la standardizzazione (Formula 2) delle variabili numeriche del dataset.

$$X = \frac{X - \mu}{\sigma} \tag{2}$$

Prima di procedere con la descrizione dei modelli utilizzati, è necessario specificare che le variabili *target* delle due classificazioni vengono create sulla base dell'appartenenza di tutte le misure rispetto ai relativi intervalli di accettazione: se tutti gli intervalli sono rispettati viene assegnata la conformità, altrimenti l'istanza verrà identificata come non conforme. Come è stato notato in fase di analisi del dataset, la bontà del processo produttivo comporta un forte sbilanciamento a favore delle osservazioni conformi. A tal proposito si è deciso di adottare un

doppio approccio: il primo prevede di usare il dataset nella sua forma sbilanciata, il secondo invece prevede di applicare una tecnica *Random Undersampling* stratificata rispetto alla variabile *target* in modo da risolvere il problema dello sbilanciamento dei dati.
Per risolvere i problemi di classificazione descritti sono stati utilizzati, in entrambi i casi, 4 modelli di Machine Learning che sono: *Random Forest* [25], *XGBoost* [26], *K-Nearest Neighbors* [27] e *Neural Network* [28].

4.4.1 Random Forest

Random Forest è un algoritmo di apprendimento basato su un insieme di *decision tree*. Si tratta, dunque, di un modello *ensemble*: ciò significa che vengono combinati più modelli dove ciascun classificatore dà un voto rispetto all'assegnamento di una *label* per ciascuna istanza e, in base alla modalità di *voting* definita, viene decretata l'etichetta finale. Nel caso del Random Forest l'esito della classificazione viene stabilito tramite la modalità *Max Voting*, cioè si attribuisce all'osservazione la classe con il maggior numero di voti.

4.4.2 XGBoost

XGBoost è un modello di classificazione che implementa un *Gradient Boosted Decision Tree* [29]. Questo classificatore prevede di realizzare un certo numero di alberi decisionali in sequenza, in modo tale che ogni *decision tree* complementi quelli precedentemente costruiti. Questo modello è stato scelto in quanto è conosciuto per la sua efficacia in dataset altamente sbilanciati come quello in esame ed è stato implementato tramite la libreria *XGBoost*.

4.4.3 K-Nearest Neighbor

K-Nearest Neighbor è un modello di apprendimento supervisionato che, basandosi sul concetto di *neighbor*, assegna l'etichetta finale ad un'istanza. L'idea generale è quella di rappresentare le osservazioni in forma di punti in uno spazio m-dimensionale (considerando m features), e, fornita una *distance function*, si procede a calcolare la distanza con le k osservazioni circostanti per ogni osservazione. Si procede quindi ad assegnare ad ogni istanza i la classe di appartenenza in funzione della classe più rappresentata nell'area nello spazio comprendente i k vicini, considerando la distanza da ciascuno di essi.

4.4.4 Neural Network

Il quarto classificatore scelto è Neural Network, ovvero un modello composto da neuroni artificiali, interconnessi tra loro, i quali elaborano i segnali ricevuti e trasmettono il risultato ai nodi successivi: l'attivazione di ciascun nodo è controllata dalla funzione di attivazione che definisce come i nodi corrispondenti reagiscono ai segnali. Le reti neurali richiedono un complesso meccanismo di addestramento per poter rendere al meglio: esso è composto da epoche, durante le quali i nodi coinvolti assumono dei pesi per trovare (al termine di ogni epoca) la migliore interpretazione possibile del segnale ricevuto. La stima della bontà dei pesi per ogni epoca è ottenuta valutando una funzione obiettivo, da minimizzare, chiamata *loss function*. L'interpretazione viene riveduta e corretta, grazie al valore di *loss* ottenuto, al passare di ogni epoca, in modo da convergere verso un risultato finale che rappresenti la massima capacità di interpretazione del modello.

4.5 Classificazione ST50

L'obiettivo stabilito inizialmente prevede di effettuare la classificazione dell'esito di produzione conseguito alla stazione *ST50*. Potendo assumere solamente due valori, questo compito viene identificato come task di classificazione binario, affrontato tramite l'utilizzo di classificatori più comuni e modelli basati su reti neurali applicati ai dati.

Ogni modello viene valutato con una procedura di *train-test split* in cui i dati vengono separati nelle proporzioni 80%-20% rispettivamente in *train set* e *test set*, in modo da poter allenare gli algoritmi su un insieme di dati diverso da quello su cui verranno successivamente testati per la valutazione delle performance di classificazione. Si osservano i risultati ottenuti per entrambi i dataset, sbilanciato e non, sulle relative porzioni di *test set*. Nel primo, tutti i classificatori ricadono nella problematica della *Zero Rule* [30]: uno specifico caso in cui l'apprendimento soffre di un *bias* che porta a classificare ogni singola osservazione con la classe maggioritaria (commettendo un numero di errori sostanzioso). Considerando il caso in cui il dataset è bilanciato, è possibile osservare che le prestazioni dei classificatori risultano ugualmente deludenti, garantendo il raggiungimento di un'*accuracy* (metrica utilizzata per la valutazione dei modelli di questa stazione) pari al 57% per *Random Forest* ed *XGBoost*, considerando gli altri modelli invece le performance si abbassano di un punto percentuale. Le performance raggiunte non risultano essere in alcun modo soddisfacenti, motivo per il quale si è deciso di introdurre un approccio basato su *Anomaly Detection*.

Le tecniche di *Anomaly Detection* provenienti dal mondo *Machine Learning* permettono di individuare anomalie nei dati (nel caso specifico rappresentate dai prodotti scartati) le quali si differenziano per alcune caratteristiche che altri approcci *Machine Learning* non sono in grado di identificare.

Gli *Autoencoder* sono adatti a tale scopo. Si tratta di modelli basati sulla struttura di una rete neurale, la quale decrementa la dimensione della rappresentazione dei dati con una porzione di rete *Encoder*; questi dati possono essere poi decodificati per generare una rappresentazione sintetica attraverso la porzione di rete chiamata *Decoder*. L'utilità che questi modelli forniscono in ambito *Anomaly Detection* viene catturata da entrambi *Encoder* e *Decoder*: l'encoder apprende (in fase di training del modello) quale sia la normale distribuzione delle variabili che rappresentano un'osservazione conforme e ne crea una versione sintetica; il decoder estrae da questa versione una rappresentazione il più simile possibile a quella originale. Nel momento in cui, in fase di test, l'osservazione è caratterizzata da anomalie o da pattern specifici, tramite la codifica e la successiva decodifica dell'osservazione si potrà osservare che il relativo esito è differente dall'osservazione fornita in ingresso. Utilizzando la misura *Mean Squared Error* (*MSE* [31]) e selezionando un valore di *MSE* soglia secondo il percentile che rappresenta la proporzione di valori anomali, è possibile determinare che, se l'osservazione possiede una distanza superiore rispetto alla soglia, allora rappresenta un'anomalia e quindi uno scarto di produzione. Tale tecnica viene applicata sul dataset in cui non sono incluse le variabili di rischio, riconoscendo solo 744 delle oltre 34.000 osservazioni non conformi: un risultato che non permette di considerare ulteriori sviluppi in tale contesto.

La seconda tecnica applicata per implementare un sistema di *Anomaly Detection* è costituita dalle *Isolation Forest*, ovvero un algoritmo simile a Random Forest adattato per eseguire l'*Anomaly Detection*: inizialmente i dati vengono rappresentati su un iperpiano che viene suddiviso in settori secondo la densità degli elementi, successivamente la tecnica prevede

che ciascun elemento venga separato dagli altri. La rilevazione di un valore anomalo avviene sulla base del numero di iterazioni necessarie per effettuare l'isolamento: se un elemento richiede molte iterazioni significa che questo si troverà in una zona ad alta densità, pertanto sarà molto simile ad altri elementi; se invece un elemento viene isolato facilmente significa che questo si trova in una zona a bassa densità di elementi, ovvero una condizione tipica delle anomalie. La tecnica in questione non ha portato i risultati sperati in quanto, nonostante si siano provate diverse configurazioni della foresta decisionale, nessuna di queste ha ottenuto un'*accuracy* soddisfacente: l'algoritmo cade nella *Zero Rule* ed identifica tutte le osservazioni come conformi.

L'ultimo approccio che viene presentato si basa sul *Multivariate Gaussian Distribution for Anomaly Detection*. Per utilizzare tale metodo viene considerato il dataset originale, già filtrato dalle osservazioni che causano scarti prima della stazione *ST50*, senza variabili "**-INT*" o "**-RIS*". Nell'utilizzo di questa tecnica il dataset viene diviso in tre porzioni: *train*, *validation* e *test* rispettivamente di dimensione *60%*, *20%* e *20%*. Ottenute le porzioni di dataset, si utilizza l'insieme di *train* per poter calcolare la media e la deviazione standard delle variabili e si procede ad utilizzarle per rappresentare i dati con una distribuzione gaussiana multivariata tale per cui ad ogni punto della distribuzione si associa una probabilità: la probabilità è più alta nella porzione centrale della curva e più bassa agli estremi. Al di sotto di una specifica soglia di probabilità determinata dall'algoritmo sulla porzione di dati *validation*, un'osservazione verrà considerata *outlier* e quindi non conforme. Si applica l'algoritmo alla porzione di dati di *test* con la specifica soglia stabilita al passo precedente, generando dei risultati che sono poi confrontati con il reale esito dell'osservazione: a causa dell'elevatissimo valore che la soglia assume, per via del dataset composto da osservazioni *outlier* considerate ugualmente conformi, tutte le osservazioni della porzione di *test* sono considerate anomale. Il metodo, a seguito del pessimo risultato e del lunghissimo procedimento di selezione della soglia durato 4 ore, viene scartato in quanto assolutamente inadatto a svolgere il suo scopo.

Nonostante le ottime premesse, né l'utilizzo di classificatori appositi né le svariate tecniche di *Anomaly Detection* applicate hanno portato a risultati soddisfacenti, non è quindi possibile considerare di implementare un sistema di sostituzione della stazione *ST50* a livello pratico a meno che non vengano forniti dati diversi e più esplicativi nei confronti della conformità finale del prodotto. Si deduce che, a fronte dei risultati ottenuti, non sia conveniente svolgere ulteriori approfondimenti sul task in esame per la stazione *ST50*.

4.6 Classificazione ST60

Il secondo task di classificazione binario prevede di decretare se un prodotto sarà conforme o meno, in base a tutti i valori registrati al momento del raggiungimento della stazione *ST60*. La bontà della classificazione dei modelli implementati viene valutata attraverso una funzione di guadagno, da massimizzare, implementata appositamente: a seguito del confronto con il referente Astori, è emerso che la mancata segnalazione di conformità di un prodotto porta a delle conseguenze importanti riguardanti i contratti con gli acquirenti; in caso di componente dal dubbio funzionamento, lo scarto del prodotto rappresenta una soluzione molto meno costosa. Pertanto, si presenta la funzione di guadagno che si è deciso di ipotizzare empiricamente, in base a delle considerazioni progettuali. Tale formula è così definita:

$$Gain = 2 * TP + (-20) * FP + (-1) * FN + 2 * TN \tag{3}$$

dove *TP* indica i prodotti correttamente classificati come conformi, *FP* i prodotti erroneamente classificati come conformi (componente di costo più grave), *FN* i prodotti conformi classificati come non conformi ed infine *TN* come i prodotti correttamente scartati.
Complessivamente, il processo implementato per la risoluzione del problema di classificazione è simile a quello svolto per la stazione *ST50*, ma presenta alcune differenze. La prima è che l'approccio implementato attraverso i modelli classici di *Machine Learning* ha garantito l'ottenimento di risultati migliori nella versione con il dataset bilanciato (unico dataset considerato) comprensivo delle variabili "**-RIS*": ciò ha permesso di applicare tecniche di ottimizzazione degli iperparametri al fine di migliorare le performance ottenute partendo da un modello già soddisfacente. Il procedimento di *train* e *test* dei modelli ha previsto l'utilizzo dell'80% del dataset per allenare il modello tramite *5-Fold Cross Validation* [32] ed ottimizzazione degli iperparametri tramite *AutoML*, ovvero un processo bayesiano [33] che determina la combinazione ottima degli iperparametri di un modello. Il rimanente 20% dei dati viene utilizzato per poter valutare le performance del modello nelle sue due combinazioni di iperparametri migliori: la versione che massimizza la funzione di guadagno ed un'altra in cui viene massimizzata la misura di *precision* [34]. Questa seconda misura di performance è stata preferita all'*accuracy* perché si è preferito minimizzare la mancata identificazione dei falsi positivi a causa dell'elevato costo che questi comportano all'azienda; inoltre, si vuole avere una misura a supporto della funzione di guadagno definita. I risultati ottenuti in fase di test dai modelli vengono presentati tramite la seguente Tabella 3:

Tabella 3: Risultati Test set

Metrica	**Random Forest**	**XGBoost**	**Nearest Neighbors**	**Neural Network**
Gain	-17229.4	-18008.4	-18246.6	-12537.2
Precision	0.733468	0.726846	0.727473	0.750731

I parametri dei modelli migliori appena presentati sono disponibili nella sezione dedicata "Caricamento dei modelli migliori" del Notebook Classificazione. Al fine di migliorare l'interpretabilità dei risultati si è utilizzato *Skater*, una libreria pensata per aiutare gli sviluppatori a comprendere meglio quale sia il comportamento di un modello quando deve prendere una decisione in base a delle feature. Fornita in input una porzione ridotta del dataset (nel caso in esame solo 15.000 osservazioni), si procede ad effettuare una perturbazione dei dati e restituisce come output l'importanza che ciascun classificatore fornisce ad ogni feature, che viene poi visualizzata tramite grafici disponibili nella cartella "*Distribuzione variabili Dataset/Grafici misure dai test*".

Come per il task di classificazione precedente, anche per la stazione *ST60* si è provato ad implementare un sistema basato su *Anomaly Detection* coinvolgendo *Autoencoder* e *Isolation Forest* sul dataset non bilanciato e privo delle variabili "**-RIS*": in questo caso, non viene svolto l'approccio *Gaussian Distribution for Anomaly Detection* in quanto non viene garantito il raggiungimento del risultato entro i tempi e le capacità di calcolo concessi dal servizio di macchine virtuali di *Google Colab* [35]. Entrambi i metodi applicati non garantiscono performance adeguate a svolgere tale compito: l'*AutoEncoder* identifica come negativi solamente il 4.3% del totale dei veri negativi. Ugualmente l'*Isolation Forest* non garantisce risultati affidabili, caratterizzati da *Zero Rule*.

5 Conclusioni

Il progetto si è incentrato dapprima su un'analisi complessiva dei dati e delle relazioni che esistono tra le variabili, successivamente si è proposto un approccio allo svolgimento dei test pneumatici e funzionali alternativo a quello attuale. Dalla prima sezione del progetto è emerso che, dai dati disponibili, non è possibile determinare alcuna relazione utile con le tecniche di regressione applicate: si considera che i dati a disposizione non risultano essere sufficientemente esplicativi per le relazioni d'interesse e che sia necessario provvedere a considerare ulteriori misure o processi di misurazione per poter completare le informazioni mancanti.

Per quanto riguarda il secondo task si possono trarre alcune conclusioni: il task di classificazione dell'esito della produzione alla stazione *ST50* non può essere svolto con i soli dati disponibili in quanto non sufficientemente informativi riguardanti l'adeguatezza del componente; inoltre, questi assumono valori troppo simili tra loro (e interni all'intervallo di accettazione) relegando il fallimento ad un evento che pare casuale. Invece, per il secondo caso di classificazione riguardante l'esito della stazione *ST60*, è necessario definire quale sia la funzione di costo idonea per valutare correttamente i modelli implementati, in quanto si è proposta una funzione fittizia basata su considerazioni progettuali e non indicativa del contesto reale di applicazione.

Alcuni possibili suggerimenti possono riguardare l'utilizzo di tecniche basate sul riconoscimento delle immagini per valutare la qualità dell'assemblaggio svolto o per valutare la corretta dimensione di ciascun componente. Infine, si propone all'azienda di effettuare misurazioni riguardanti la durata delle singole operazioni svolte durante l'assemblaggio con un duplice fine: fornire più conoscenza per il processo di classificazione ed, in secondo luogo, introdurre un'analisi di stagionalità per osservare eventuali pattern ricorrenti.

Riferimenti bibliografici

[1] BoschGroup, "Cosa facciamo," Bosch.

[2] Wikipedia. Robert bosch (azienda). Bosch Wiki.

[3] C. R. Littler, "Understanding taylorism," *British Journal of Sociology*, pp. 185–202, 1978.

[4] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[5] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[6] P. Royston, "Approximating the shapiro-wilk w-test for non-normality," *Statistics and computing*, vol. 2, no. 3, pp. 117–119, 1992.

[7] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[8] K. D. Hopkins and D. L. Weeks, "Tests for normality and measures of skewness and kurtosis: Their place in research reporting," *Educational and Psychological Measurement*, vol. 50, no. 4, pp. 717–729, 1990.

[9] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1649–1652.

[10] C. Chio and D. Freeman, *Machine learning and security: Protecting systems with data and algorithms*. " O'Reilly Media, Inc.", 2018.

[11] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[13] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016.

[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[15] A. Ng. Anomaly detection using the multivariate gaussian distribution. coursera.org/lecture/anomaly_detection.

[16] A. Kramer and P. Choudhary. Skater xai. github.com/oracle/Skater.

[17] R. E. Shiffler, "Maximum z scores and outliers," *The American Statistician*, vol. 42, no. 1, pp. 79–80, 1988.

[18] N. C. Schwertman, M. A. Owens, and R. Adnan, "A simple more general boxplot method for identifying outliers," *Computational statistics & data analysis*, vol. 47, no. 1, pp. 165–174, 2004.

[19] R. M. Sakia, "The box-cox transformation technique: a review," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, no. 2, pp. 169–178, 1992.

[20] J. A. Tillman, "The power of the durbin-watson test," *Econometrica: Journal of the Econometric Society*, pp. 959–974, 1975.

[21] F. Schmittroth, "Generalized least-squares for data analysis," Hanford Engineering Development Lab., Richland, Wash.(USA), Tech. Rep., 1978.

[22] G. Strang, "A proposal for toeplitz matrix calculations," *Studies in Applied Mathematics*, vol. 74, no. 2, pp. 171–176, 1986.

[23] K. A. Clarke, "The phantom menace: Omitted variable bias in econometric research," *Conflict management and peace science*, vol. 22, no. 4, pp. 341–352, 2005.

[24] "scikit-learn machine learning in python," Sklearn home.

[25] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93–104, 2012.

[26] M. Luckner, B. Topolski, and M. Mazurek, "Application of xgboost algorithm in fingerprinting localisation task," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2017, pp. 661–671.

[27] tutorialspoint, "Knn algorithm - finding nearest neighbors," tutorialspoint*link*.

[28] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[29] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[30] M. L. Catalogue. Zero rule. Zero Rule - MLC.

[31] D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, 1971.

[32] S. R. Sain, K. A. Baggerly, and D. W. Scott, "Cross-validation of multivariate densities," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 807–817, 1994.

[33] B. M. Colosimo and E. Del Castillo, *Bayesian process monitoring, control and optimization*. CRC Press, 2006.

[34] S. A. Alvarez, "An exact analytical relation among recall, precision, and classification accuracy in information retrieval," *Boston College, Boston, Technical Report BCCS-02-01*, pp. 1–22, 2002.

[35] Google, "Google colab resources," Colab FAQ.

6 Appendice

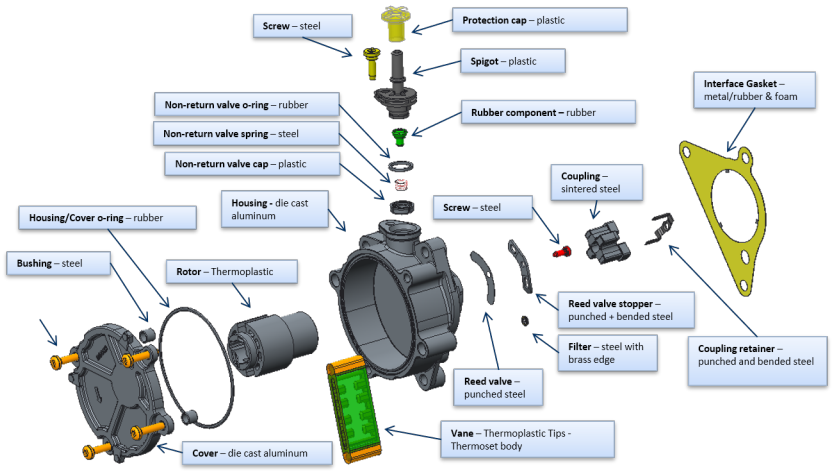

Figura 1: Esploso della vacuum pump prodotta alla linea MVP11

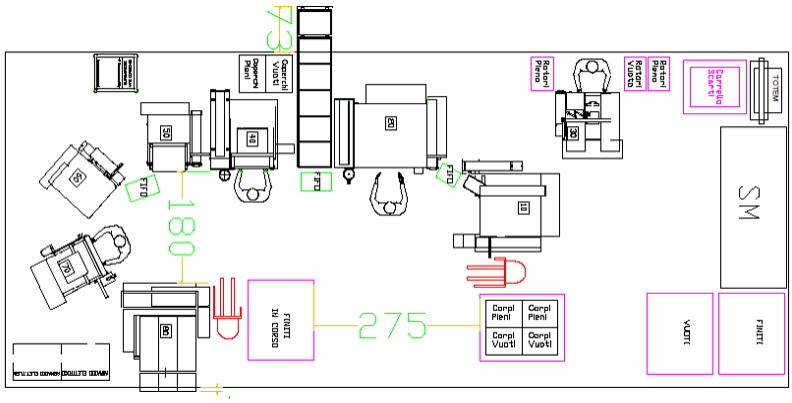

Figura 2: Schema linea di produzione MVP11

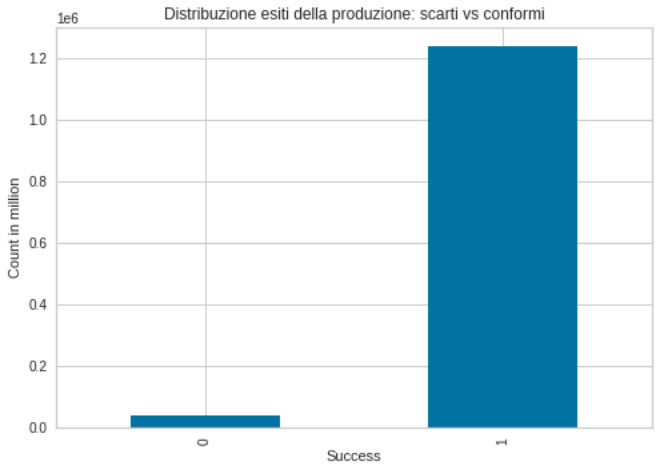

Figura 3: Proporzione conformi-scarti

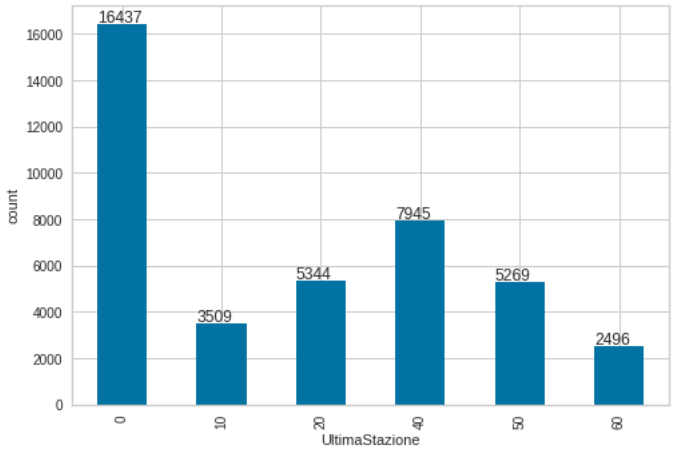

Figura 4: Scarti per stazione

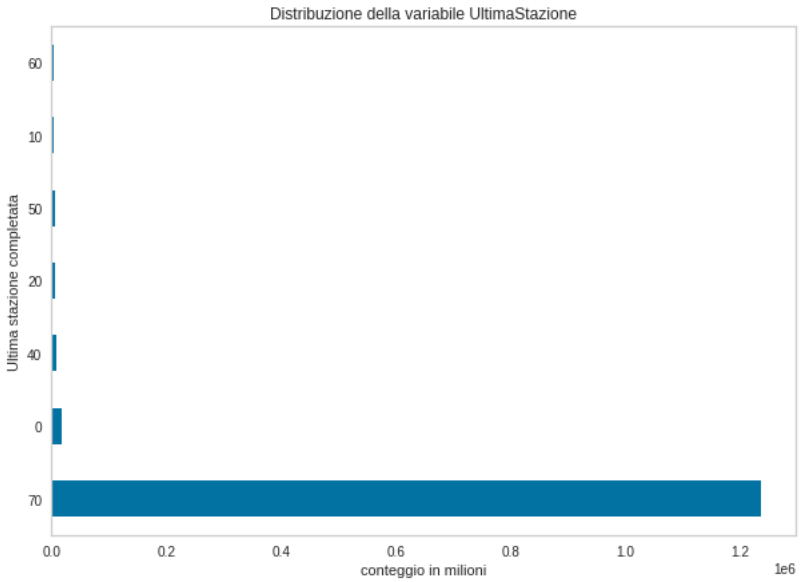

Figura 5: Distribuzione valori ultima stazione

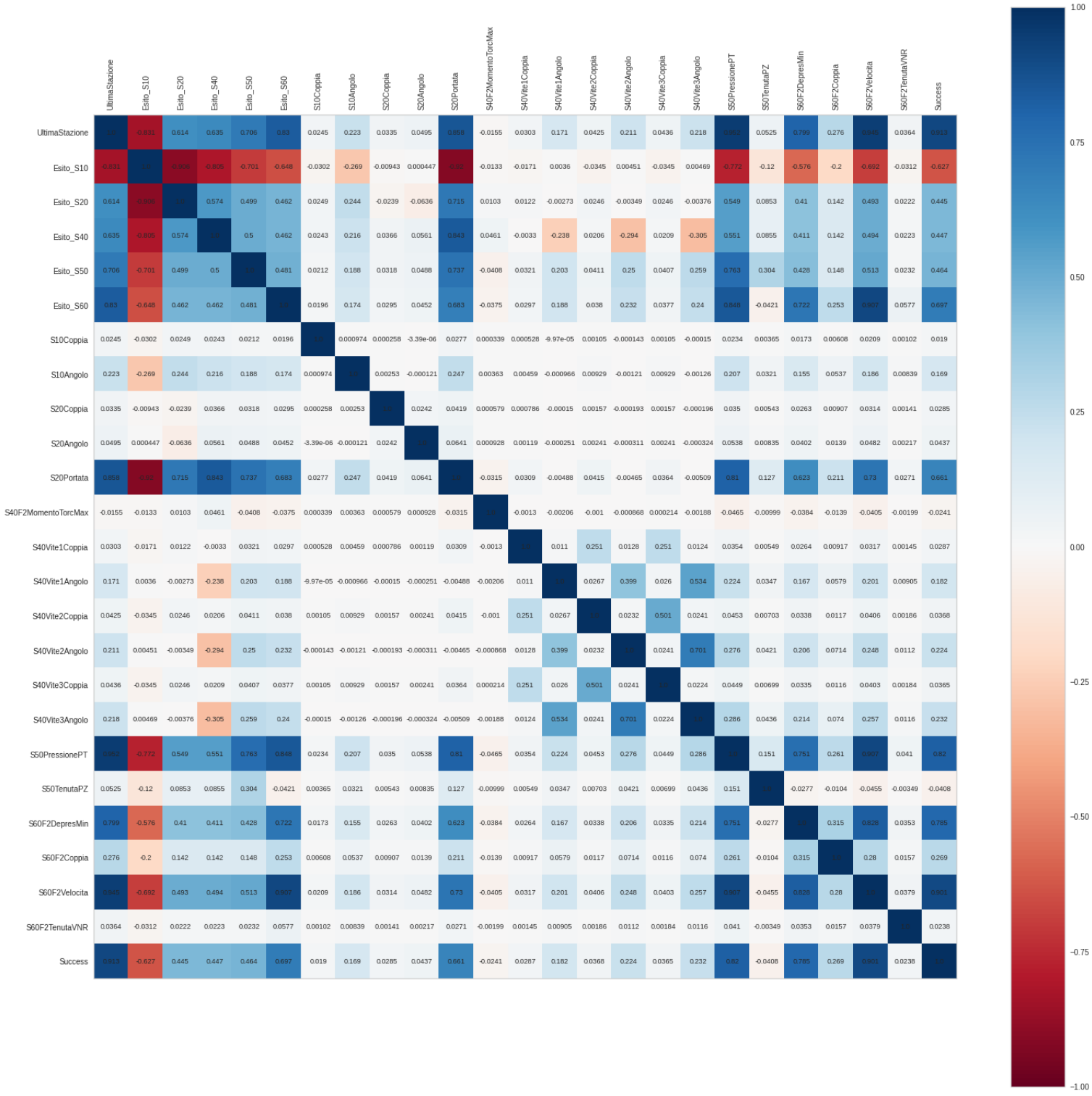

Figura 6: Correlazione

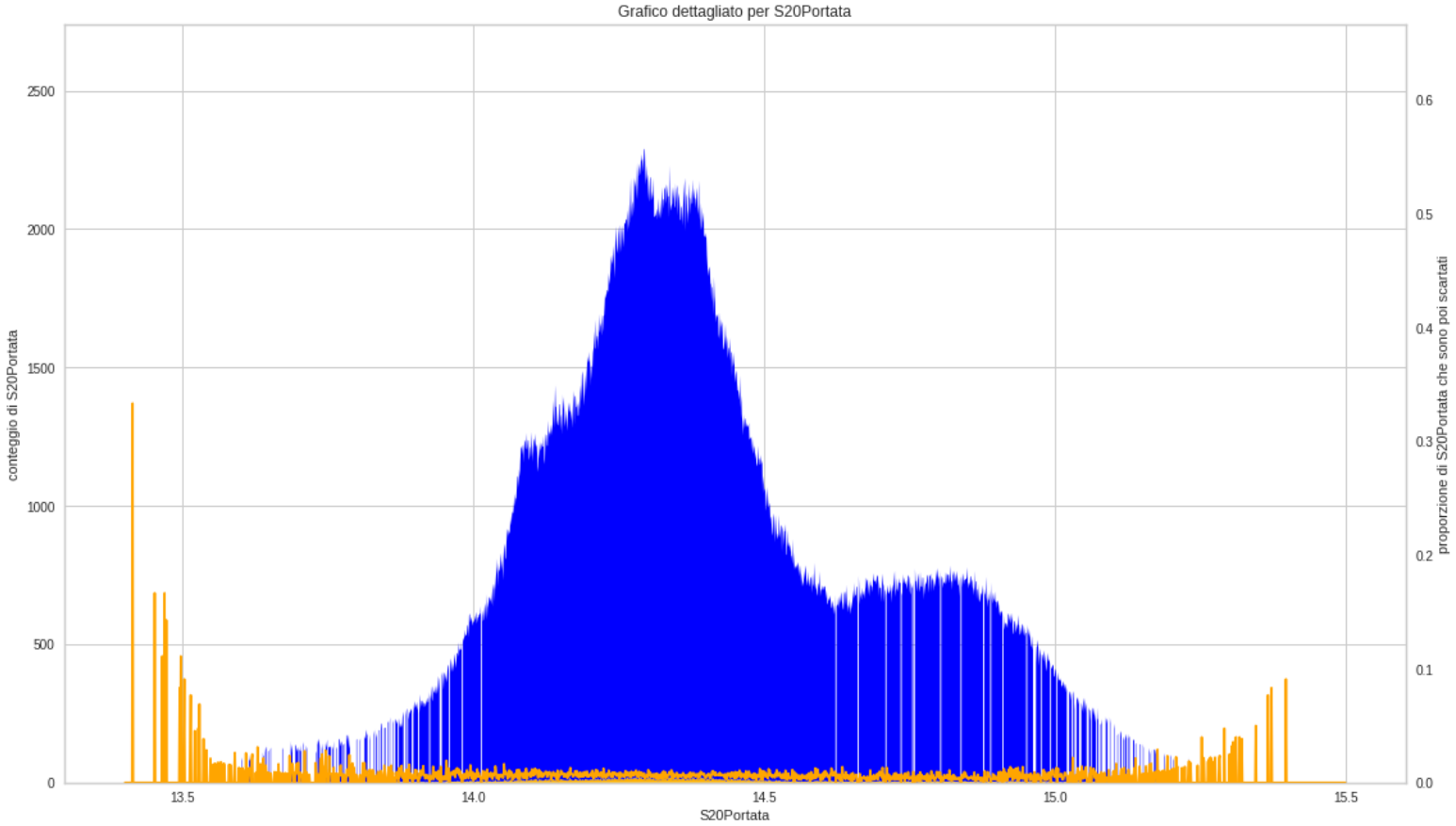

Figura 7: Distribuzione successi e fallimenti per ciascun valore assunto

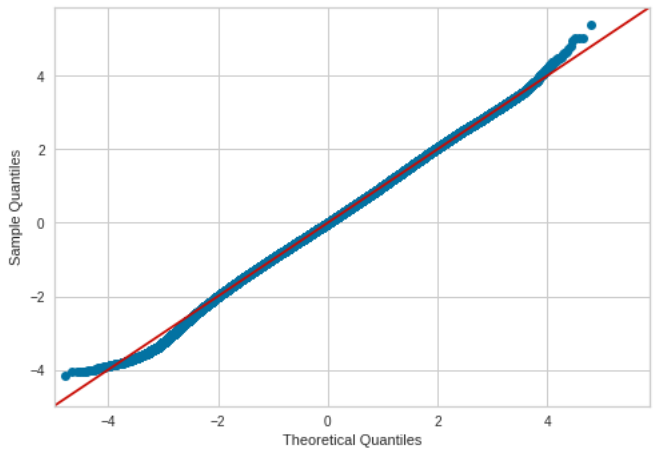

Figura 8: Distribuzione dei residui variabile S60F2DepresMin