POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea

Metodologie data-driven a supporto della manutenzione predittiva. Applicazioni in ambito robotica nell'era Industry 4.0

Relatrice:

Prof.ssa Tania Cerquitelli

Candidato:

Bruno Pinci

Correlatore:

Dott. Francesco Ventura

Anno Accademico 2018/2019

SOMMARIO

1 – Introduzione **3**
2 – Industry 4.0 ..... **6**
2.1 – Significato e tecnologie cardine 6
2.2 – La *smart factory* 10
2.3 – Situazione attuale 12
3 – Contesto e tecnologie **14**
3.1 – *Data analytics* 14
3.2 – *Machine learning* 20
3.3 – Tecniche di classificazione 25
3.4 – Manutenzione predittiva 27
3.4.1 – Stato dell'arte 31
4 – Caso di studio **34**
4.1 – *Belt tensioning* 35
4.2 – Struttura dei dati 37
4.3 – Strumenti utilizzati 44
5 – Realizzazione dei modelli **47**
5.1 – Preprocessing dei dati 48
5.2 – Modello Generale 57
5.2.1 – Modelli con il Random Forest 58
5.2.2 – Modelli con l'Isolation Forest 70
5.3 – Modello Evolutivo 76
5.3.1 – Prima fase 79
5.3.2 – Seconda fase 82
6 – Conclusioni **96**
Bibliografia e sitografia **99**

1 – INTRODUZIONE

Lo sviluppo tecnologico negli ultimi anni ha compiuto passi da gigante, dando forma a nuove invenzioni, nuovi modelli di business e nuovi settori in campo industriale e dei servizi, che fino a poco tempo fa neanche esistevano e che stanno rivoluzionando la vita quotidiana dell'uomo. Ogni giorno vengono pensate nuove idee, e progetti innovativi sono lanciati sul proprio mercato di riferimento, sconvolgendo regole e paradigmi fino a questo momento in uso. Basti pensare ai settori delle telecomunicazioni e della telefonia, passando per l'informatica e tutto ciò che riguarda internet fino ad arrivare alla robotica, guidata da sistemi fondati su una connessione totale delle parti tramite l'intelligenza artificiale. Queste due parole, da tempo sulla bocca di tutti ma spesso utilizzate senza conoscerne il vero significato, sono ormai parte della vita umana di tutti i giorni. Inoltre, l'evoluzione digitale ha fatto sì che chiunque, anche solo utilizzando il proprio smartphone, possa collezionare e trasmettere informazioni di valore che possono essere sfruttate in moltissimi modi diversi. Miliardi di dati sono scambiati ogni giorno e il loro potere cresce minuto dopo minuto: al giorno d'oggi il loro studio sta diventando un pilastro fondamentale per ogni impresa che voglia guadagnare un vantaggio competitivo sulle altre, e sopravvivere in un mercato dove chi si ferma anche solo per un secondo rischia di perdere ingenti somme di denaro. Il motivo dell'importanza dei dati – che provengano dalle persone, dall'uso di internet o dalle macchine di una fabbrica – è molto semplice: essi permettono di capire al meglio il mondo in cui viviamo, consentono di ottimizzare i nostri sforzi dirigendoli in direzioni ben precise e in alcuni casi danno la possibilità di creare modelli di predizione del futuro tramite algoritmi induttivi, che dai casi particolari comprendono il comportamento di sistemi sempre più generali.

Pertanto, questa tesi s'inserisce nell'ambito della *big data analytics*, andando ad approfondire nel dettaglio il contesto che ha permesso a questa tecnologia di divenire così determinante negli ultimi anni. Verranno affrontati temi importanti che fanno da contorno alla scienza dei dati, come l'ormai attuale quarta rivoluzione industriale che sta toccando tutti i settori più evoluti, oltre che le nuove applicazioni dell'apprendimento automatico dei computer. In particolare, grazie alla collaborazione

con un'importante società attiva nei campi di automazione e robotica, ci si occuperà di una casistica reale con l'applicazione di alcune tecniche di *data mining* su dati raccolti in ambito industriale. Il caso di studio affrontato, riguarda un progetto di manutenzione predittiva, effettuato tramite lo studio approfondito del segnale di corrente emesso da un robot appartenente all'impresa appena citata. Questa analisi ha lo scopo primario di cercare di cogliere, prima che sia troppo tardi, i segni di un possibile malfunzionamento del braccio robotico, probabilmente dovuto ad un cattivo tensionamento della sua cinghia di trasmissione, tramite un modello predittivo di classificazione in grado di avvisare nel momento in cui è necessario un intervento di riparazione. Manutenzione predittiva, apprendimento automatico e *data mining* sono tutti termini connessi tra loro che formano in un certo senso il filone dominante di questa tesi e fanno parte della base da cui è partita la già citata Industria 4.0.

L'elaborato è organizzato in sei capitoli, di cui il primo è semplicemente l'introduzione, che sono strutturati nel seguente modo:

- Capitolo 2: in questo capitolo verrà introdotto il concetto di Industry 4.0. Verranno spiegati i motivi per cui si sta sviluppando questo fenomeno tecnologico, i cambiamenti che sta portando nelle fabbriche in Italia e nel mondo e verranno descritte le più importanti innovazioni introdotte per migliorare le condizioni della produzione, spesso definite come "tecnologie abilitanti".

- Capitolo 3: un altro capitolo "teorico" in cui però si entrerà più nel dettaglio nella spiegazione di alcuni metodi e tecniche relative al caso di studio affrontato. In questo capitolo infatti verranno spiegati i concetti di *data analytics* e *machine learning*, con un particolare focus sulle tecniche di classificazione esistenti. Seguirà poi la descrizione di una delle più importanti applicazioni pratiche di queste tecniche in ambito manifatturiero, la già citata manutenzione predittiva, relativamente alla quale verrà anche presentata una panoramica sullo stato dell'arte attuale.

- Capitolo 4: in questo capitolo sarà introdotto il caso di studio affrontato con i dati del robot. Si parlerà della problematica principale che lo riguarda, chiamata "*belt tensioning*", dei dati forniti, della loro struttura e degli strumenti

tecnologici utilizzati per costruire i modelli di classificazione. Inoltre questo capitolo presenterà un'iniziale indagine esplorativa sui dati stessi, effettuata in fase preliminare rispetto alle analisi vere e proprie con lo scopo di poterle indirizzare al meglio.

- Capitolo 5: è il cuore della tesi, in cui verrà descritto dettagliatamente il progetto svolto. Esso è diviso in tre parti, che rappresentano rispettivamente le diverse fasi del lavoro: nella prima si analizzerà il preprocessing dei dati, quel trattamento preliminare utile per predisporre al meglio il dataset ricevuto, la seconda e la terza parte invece verteranno sulla trattazione di due diversi tipi di modelli sviluppati, chiamati modello generale e modello evolutivo, con lo scopo comune di creare un sistema di aiuto nella manutenzione predittiva del braccio robotico.

- Capitolo 6: si tratta di un breve capitolo conclusivo, in cui saranno analizzati i risultati raggiunti nel progetto svolto e si elencheranno alcuni possibili sviluppi futuri.

2 – INDUSTRY 4.0

In questo capitolo, verrà spiegato il concetto di Industry 4.0 [1] [2] [3], che fa parte del titolo della tesi, e ne verranno analizzate le principali caratteristiche oltre alle sue numerose applicazioni, che negli ultimi anni hanno influenzato fortemente il mondo della produzione industriale, delle fabbriche e più in generale hanno radicalmente modificato l'approccio delle aziende nei confronti della tecnologia. Dopo aver introdotto questo importante concetto, verranno descritte le principali tecnologie che lo compongono, le caratteristiche della *smart factory* del futuro e la situazione attuale in Italia e nel mondo di fronte a questa rivoluzione industriale.

2.1 – SIGNIFICATO E TECNOLOGIE CARDINE

Il termine Industry 4.0 è stato utilizzato per la prima volta durante una comunicazione alla fiera di Hannover, in Germania, nel 2011, con lo scopo di descrivere un progetto volto a "costruire l'industria del futuro", mettendo in atto una serie di investimenti per sfruttare al meglio la tecnologia nell'ambito della produzione industriale. Tale progetto è stato concretizzato nel 2013, rilanciando in modo decisivo il sistema produttivo tedesco, ma nel frattempo l'espressione Industry 4.0 ha, negli anni, assunto un'altra forma e un altro utilizzo: essa attualmente va infatti ad identificare quel processo di digitalizzazione, interconnessione e automazione sempre maggiore che sta interessando sempre di più non solo le fabbriche, ma buona parte di tutte le imprese nelle zone più sviluppate, come Stati Uniti, Giappone ed Europa. Infatti, i risultati ottenuti dalla Germania in questo campo, hanno fatto sì che molti altri paesi perseguissero questa stessa politica di forte innovazione a base tecnologica, per mezzo di nuovi paradigmi di produzione e nuovi modelli di business.

Questa enorme ondata di rinnovato interesse per la tecnologia ha portato molti studiosi ad analizzare la situazione e in particolar modo le conseguenze che tali innovazioni avranno sul contesto socioeconomico dei paesi, andando a definire questo momento storico come "quarta rivoluzione industriale".

Se da un lato sono abbastanza chiari i punti salienti e le tecnologie dominanti della cosiddetta quarta rivoluzione industriale, d'altro canto ancora non è possibile stabilire la sua data di inizio, trattandosi di un processo appena iniziato e tuttora in corso, e solamente in futuro si potrà dire quale sia stato il suo atto fondante e in quale momento sia avvenuto. Ciò che per ora si può affermare è che la quarta rivoluzione industriale si fonda su alcuni punti cardine come ad esempio l'automatizzazione e l'interconnessione totale nell'ambito della produzione industriale, mentre il concetto di Industry 4.0 identifica la fabbrica del futuro, ovvero il modo migliore per completare questa rivoluzione tramite un piano di investimenti specifici e mirati, al quale anche l'Italia sta prendendo parte. Così come ogni rivoluzione industriale della storia del mondo occidentale si è sviluppata grazie ad una particolare invenzione o ad un sistema innovativo di produzione – basti pensare all'introduzione della macchina a vapore per la prima rivoluzione industriale nel 18° secolo, alla diffusione dell'elettricità durante la seconda rivoluzione unito alla produzione di massa verso fine Ottocento, o alla nascita dell'era digitale per la terza rivoluzione (a partire dagli anni Settanta) – anche per quanto riguarda l'attuale quarta rivoluzione industriale esistono diverse tecnologie portanti, sia software che hardware. La **figura 2.1** schematizza brevemente la storia delle rivoluzioni industriali.

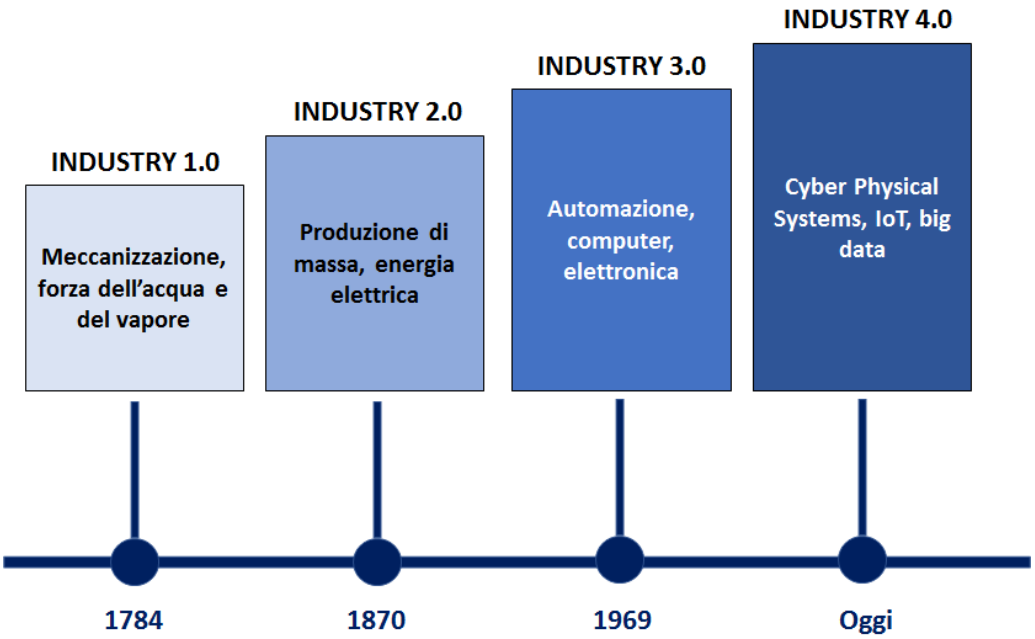

Figura 2.1: schema riassuntivo delle quattro rivoluzioni industriali

La società di consulenza americana McKinsey ha evidenziato la presenza di queste tecnologie cardine nella quarta rivoluzione industriale in un rapporto, dividendole in quattro gruppi principali [4]:

1. Il primo gruppo ha a che fare con l'utilizzo dei dati. Anche se si hanno sempre avute a disposizione grandi quantità di dati riguardanti qualsiasi attività di ogni azienda, solo adesso si ha la possibilità di sfruttarli maggiormente per migliorare le performance delle imprese. Fanno parte di questa suddivisione tutte le tecniche per la gestione di enormi quantità di dati (chiamati appunto "big data") e per la loro conservazione, nonché soluzioni basate su IoT (sigla che significa *"Internet of Things"*, e che va a identificare il collegamento di vari dispositivi sia hardware che software appunto tramite internet) e sistemi M2M (*machine to machine*) con lo scopo di incrementare la potenza di calcolo dei computer e la loro interconnessione. Per quanto concerne la centralizzazione delle informazioni è necessario invece citare i sistemi di *cloud computing* (termine che identifica generalmente un servizio di hosting su internet da usare anziché costruire e mantenere infrastrutture fisiche).

2. Il secondo gruppo è quello che riguarda la branca dell'analisi dei dati (la cosiddetta "*data analytics*"). Una volta immagazzinate grandi quantità di dati, è necessario estrarre il loro valore e utilizzarli per ottenere dei miglioramenti nella propria attività. Al giorno d'oggi vengono prodotti miliardi di dati ogni giorno e solo una minuscola percentuale di essi viene sfruttata adeguatamente. In questo contesto si collocano soluzioni di *data intelligence* e algoritmi di *data mining*, per categorizzare i dati presenti e fare delle previsioni su quelli futuri, e di *machine learning*, che si basano sull'autoapprendimento delle macchine. Di queste tecnologie si parlerà in modo più approfondito nel terzo capitolo di questa tesi.

3. Il terzo gruppo riguarda tutte le forme di interazione tra uomo e macchina. Alcuni esempi concreti, che ormai siamo abituati e vedere tutti i giorni nella vita quotidiana, possono essere le interfacce touch degli schermi di cellulari e computer oppure i sistemi di realtà aumentata. Quest'ultima è una tecnologia che, sfruttando i display degli apparecchi mobili e utilizzando, per esempio, dei

dispositivi indossabili, permette a chi la usa di "visualizzare" informazioni aggiuntive sovrapposte alla realtà. Utilizzata inizialmente soprattutto per lo sviluppo di videogiochi interattivi, oramai la realtà aumentata è diffusa in diversi ambiti applicativi, dal settore della chirurgia robotica a quello della navigazione stradale tramite nuovi navigatori satellitari potenziati, fino all'utilizzo nei musei per mezzo di software dedicati, che permettono di leggere informazioni sulle opere d'arte. Sia la realtà aumentata che gli schermi touch screen hanno in comune la semplicità d'uso che le rende appunto fruibili da buona parte della gente comune, e questo è sicuramente uno dei motivi della loro diffusione ormai sempre più capillare.

4. Il quarto gruppo, infine, comprende tutte quelle tecnologie digitali che hanno applicazioni pratiche nel campo della produzione. Tra esse possono essere ricordate la manifattura additiva, la stampa 3D [5] e la robotica: infatti sempre più spesso vengono utilizzate soluzioni di produzione automatizzata presso le fabbriche. Questi elementi hanno in comune il fatto di puntare ad una forte diminuzione dei costi per le imprese, tramite il miglioramento della qualità nella produzione e l'utilizzo mirato dell'energia, evitando in questo modo sprechi di materiale e di tempo. In realtà, il primo brevetto di stampa 3D risale addirittura agli anni Ottanta, ma soltanto di recente sta iniziando ad essere economicamente sostenibile e quindi utilizzabile nelle industrie oltre che per la prototipazione di oggetti, anche per la produzione vera e propria. La stampa 3D permette di realizzare tramite un unico processo, oggetti che solitamente richiedono l'assemblaggio di più componenti singoli; attualmente tuttavia, al termine del processo di stampa sono spesso necessari interventi di post-produzione e di finitura. Il concetto di manifattura additiva dunque, nasce da un nuovo modo di intendere la produzione industriale, che come input riceve un modello in 3D e procede grazie ad un processo semi-automatico in cui l'oggetto viene scomposto in più strati effettivamente stampabili uno sopra l'altro.

La **figura 2.2** riassume le principali tecnologie del modello Industry 4.0 di cui si è parlato finora.

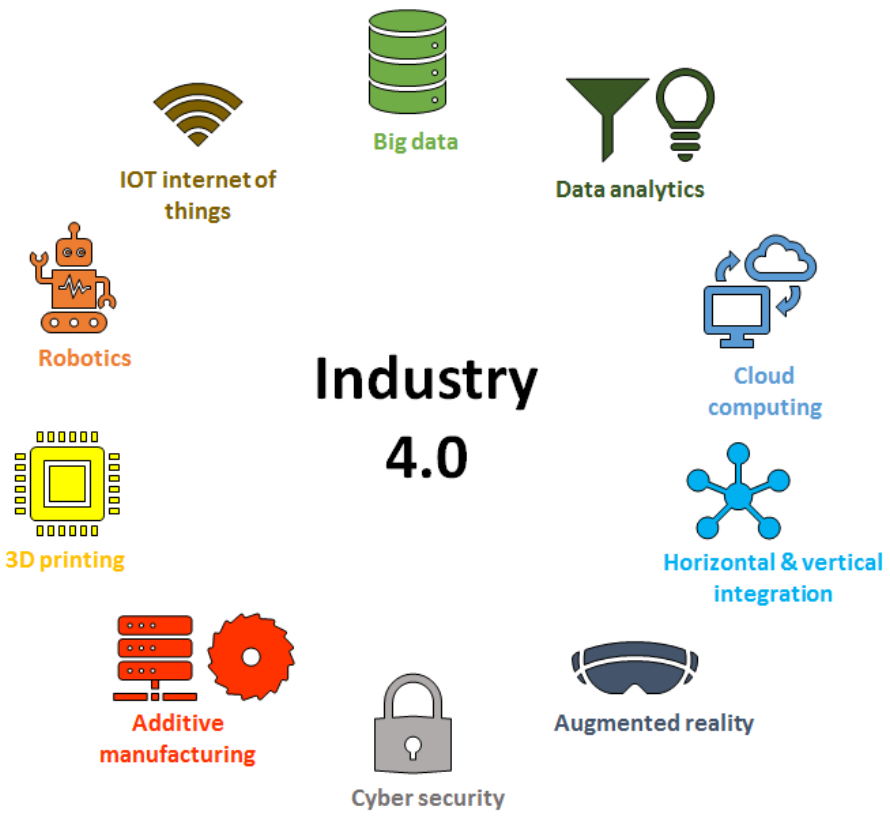

Figura 2.2: le tecnologie cardine del modello Industry 4.0

2.2 – LA *SMART FACTORY*

Per applicare le innovazioni appena descritte è necessario come prima cosa un radicale cambiamento nella fabbrica stessa, siccome essa deve poter ospitare le nuove tecnologie e trarre da esse più vantaggi possibili. L'aggettivo, in lingua inglese, che meglio identifica la Fabbrica 4.0, ovvero l'evoluzione della fabbrica classica in seguito all'arrivo della quarta rivoluzione industriale, è *smart*. Questo termine ha il significato di "intelligente" ma anche di "brillante" ed "elegante", e rappresenta appieno il nuovo modello di industria che vedremo sempre più spesso diffondersi in tutto il mondo. I tratti ricorrenti della *smart factory* sono strettamente connessi al modo in cui le nuove tecnologie sopra descritte vengono adoperate: tutti i macchinari devono essere completamente interconnessi tra loro tramite internet, in modo da poter comunicare, trasmettere dati ed effettuare operazioni di manutenzione predittiva.

L'integrazione all'interno della fabbrica deve essere sia orizzontale che verticale, per coinvolgere in questo costante scambio di informazioni ogni elemento della catena di produzione. Questa interconnessione totale può avvenire grazie alla diffusione dei sistemi cyber-fisici (CPS, sigla di "*Cyber Physical Systems*") [6], una rete di componenti software collegata a parti meccaniche ed elettroniche tramite internet con diverse funzionalità all'interno di una fabbrica, tra cui la capacità di acquisizione di dati in *real time* da sensori appositi, la gestione e il trasferimento di questi ultimi tramite opportuni protocolli di comunicazione e l'abilità di fornire adeguati feedback alla realtà fisica a partire da quella virtuale in modo da applicare azioni correttive e aiutare i manager in fase di *decision making*. Pertanto, questi CPS permettono agli impianti produttivi di vantare una grande flessibilità identificando svariati scenari e utilizzando metodi di simulazione, così da poter garantire il massimo grado di personalizzazione dei prodotti per i clienti finali.

Sempre per quanto riguarda la crescita dell'interconnessione e dell'automatizzazione nelle fabbriche, si stima che entro il 2020 la capacità di manutenzione effettuata dai robot su sé stessi o su altri robot, supererà per qualità e velocità quella praticata dagli esseri umani, specialmente per merito delle tecniche di autoapprendimento dei robot. Questi ultimi saranno in grado di imparare il proprio lavoro velocemente e in modo naturale, di simulare il flusso di lavoro da effettuare e di prevedere in anticipo guasti e fallimenti per poi mettere in opera azioni correttive. Questo nuovo tipo di produzione intelligente sarà inoltre affiancata da adeguate infrastrutture informatiche (ad esempio, con lo scopo di raccogliere dati e immagazzinarli) e da sistemi attenti al risparmio energetico nella fabbrica.

Un altro importante fattore che non può mancare nella struttura della fabbrica 4.0 è la *cyber security*. Con l'aumentare delle connessioni interne ed esterne, e con il moltiplicarsi delle quantità di dati raccolti ogni giorno, è fondamentale un sistema che garantisca l'integrità di tali informazioni e che ne impedisca l'accesso a esterni.

2.3 – SITUAZIONE ATTUALE

Una domanda che può sorgere spontanea, a proposito del nuovo modello di fabbrica, è la seguente: se la maggior parte del lavoro verrà svolto dai robot, che ruolo avrà l'essere umano nell'ambito della fabbrica 4.0? Sembra abbastanza scontato affermare che questa rivoluzione industriale stia già avendo, e avrà ancora di più nei prossimi anni, un grande impatto sul mondo del lavoro. La ricerca "*The future of the jobs*" [7], presentata nel 2016 al World Economic Forum afferma infatti che vi saranno diversi cambiamenti nelle posizioni ricercate dalle aziende e che nel frattempo nasceranno nuovi lavori e ne scompariranno altri. È stato stimato che in tutto il mondo verranno creati 2 milioni di nuovi posti di lavoro e ne spariranno altri 7: le maggiori perdite si avranno nei campi amministrativi e della produzione, mentre saranno sempre più ricercate figure professionali nell'ambito dell'ingegneria, dell'informatica, della finanza e del management.

Per quanto riguarda l'Italia, si stimano 200 mila posti di lavoro persi e altrettanti creati, ma il vero problema che è necessario risolvere per affrontare la quarta rivoluzione industriale è quello di fornire delle adeguate competenze digitali ai lavoratori in modo da renderli pronti per svolgere le occupazioni del futuro. Solo in questo modo il nostro paese sarà in grado di cogliere appieno i benefici che quest'ondata di innovazione potrà portare. Seguendo tale obiettivo i governi che si sono succeduti negli ultimi anni hanno messo in opera un piano di investimenti specifici, offrendo alle grandi imprese e soprattutto alle PMI una serie di agevolazioni in modo da invogliarle a prendere parte al modello Industry 4.0.

In particolare, a partire dal 2016 gli investimenti sono stati sfruttati nell'ambito della ricerca e sviluppo, con un interesse maggiore per tutte le tecnologie dell'Industry 4.0. Il piano proposto dal Ministero dello Sviluppo Economico proponeva alcune interessanti iniziative:

- Iper e super ammortamento: provvedimento che va direttamente a supporto di quelle imprese che investono i propri capitali in beni, sia materiali che non (come ad esempio dei sistemi software) a base tecnologica e utili alla

trasformazione digitale tipica dell'industria 4.0, con la funzione di fornire a tali imprese un'agevolazione fiscale.

- Nuova Sabatini: iniziativa volta a garantire una parziale copertura degli interessi da pagare su finanziamenti bancari forniti per investimenti in nuovi macchinari e impianti innovativi.

- Semplificazione amministrativa e maggiori agevolazioni fiscali per le startup e le PMI innovative. Un esempio è stata la possibilità per le startup di cedere le proprie perdite (relative soltanto ai primi 3 esercizi) ad una società quotata che ne possedesse almeno il 20% del capitale, o anche la garanzia di ottenere una detrazione fiscale per chi decidesse di investire nelle startup.

- Istituzione di alcuni centri di competenza per svolgere attività di formazione e orientamento alle imprese, oltre che supporto nella ricerca e nei progetti a base innovativa.

Attualmente [8], questo piano, dopo aver subito alcune modifiche, è stato depotenziato dalla Legge di Bilancio 2019, mentre alcuni dei suoi punti chiave sono stati attuati, sebbene leggermente a rilento rispetto alle previsioni. L'iperammortamento ad esempio ha subito alcune modifiche importanti (l'agevolazione fiscale per investimenti fino a 2,5 milioni di euro è salito al 270% rispetto al 250% del piano iniziale, per poi diminuire con il crescere dell'importo investito) con lo scopo di avvantaggiare sempre di più le PMI e sempre di meno le grandi imprese, che si suppone abbiano già effettuato gli investimenti necessari negli anni passati. Per i centri di competenza invece è stato istituito un bando di aggiudicazione, anche se inizialmente essi dovevano svilupparsi attorno ad alcune importanti Università d'Italia, e per tale ragione non sono ancora operativi.

3 – CONTESTO E TECNOLOGIE

Lo scopo di questo capitolo è quello di fornire al lettore un'idea generale del contesto in cui nasce e prende forma il progetto a cui si riferisce questa tesi. Pertanto, verranno analizzati e descritti i principali concetti che stanno alla base del lavoro svolto e che quindi si riveleranno utili per comprenderne non soltanto l'ambito di sviluppo ma anche la metodologia e le tecniche utilizzate durante le varie analisi. Si tratta di *data analytics*, *machine learning* e scendendo più nel dettaglio si parlerà delle principali tecniche di classificazione e della manutenzione predittiva. Il capitolo si conclude con una panoramica sullo stato dell'arte attuale per quanto riguarda l'utilizzo di queste tecnologie.

3.1 – *DATA ANALYTICS*

Come è già stato sottolineato in precedenza, l'analisi approfondita dei dati rappresenta un blocco fondamentale per il modello Industry 4.0 ed è ormai materia essenziale per ogni impresa che lavora con grandi moli di dati ed ha interesse nello sfruttare il loro enorme potenziale. Inoltre, nell'ambito di questa tesi, sono state utilizzate diverse tecniche di *data analytics*, motivo per cui è necessario approfondire questo argomento. L'espressione *data analytics* [9] [10] [11] può essere semplicemente definita come il processo di studio di alcuni set di dati con lo scopo di ottenere informazioni circa il loro contenuto. Ovviamente non si tratta di una semplice lettura dei dataset, bensì di un'analisi mirata a scoprire informazioni nascoste all'interno dei dati stessi e non leggibili senza l'aiuto di sistemi software specializzati.

Prima ancora di andare a descrivere più nel dettaglio le applicazioni e i vantaggi della *data analytics*, è però doveroso ricordare il motivo per il quale essa è diventata sempre più importante negli ultimi anni. Infatti, l'analisi di dati deve la sua importanza al fatto che solo di recente è diventato possibile raccogliere un numero consistente di dati su cui effettuare degli studi: questi grandi insiemi di dati vengono definiti "big data", proprio a causa della loro estensione in termini di volume, tale da rendere necessario l'uso di

tecnologie apposite per estrarne il valore. Sebbene ogni giorno vengano prodotte tantissime informazioni classificabili come "dati" sia dalle imprese sia dalle persone comuni (basti pensare all'enorme numero di smartphone attivi in tutto il mondo: ogni operazione effettuata con uno di essi può essere registrata in un database e in seguito estratta e sfruttata come informazione), soltanto di recente sono nati dei metodi per poter immagazzinarle e gestirle nel modo migliore.

Un sistema di big data indica generalmente un dataset che a causa delle sue dimensioni sovrasta le capacità dei sistemi classici di gestione dei dati di immagazzinarli ed effettuare analisi su di essi. Al giorno d'oggi le aziende raccolgono dati da sensori, da telefoni, da immagini, da e-mail, da informazioni GPS e addirittura dai social network. Come può essere facile immaginare, questi dati non possiedono tutti la stessa struttura siccome provengono da fonti eterogenee e per questo motivo sono nate nuove tecnologie adatte ad ospitare questi tipi di dati, basate sulla distribuzione di risorse e servizi. Tra esse è possibile citare i database non relazionali, ottimi per la gestione di dati non strutturati (ossia dati senza alcuno schema, come ad esempio dati testuali o transazionali) che a differenza dei classici database non hanno bisogno che i dati abbiano una struttura fissa, sotto forma di tabella. Questi sistemi hanno migliorato notevolmente le prestazioni in termini di velocità di estrazione ed analisi dei big data rispetto ai database relazionali che si servono del linguaggio SQL per accedere ai dati.

A causa del loro volume, i big data pongono altri due problemi che chiunque si trovi a svolgere un progetto di *data analytics* deve affrontare. Innanzitutto, è molto probabile che ogni analisi di grandi quantità di dati non sia immediata, perciò anche partendo da dati recenti i risultati si otterrebbero dopo un certo lasso di tempo e non in tempo reale, e nel frattempo un'altra grande mole di dati verrebbe generata e non subito inclusa nell'analisi. Il secondo problema riguarda lo storage dei dati: ogni azienda interessata a raccogliere informazioni sul proprio operato necessita di adeguate infrastrutture per la loro archiviazione. Oltre a immagazzinare più dati possibili, è necessario difenderne l'accuratezza e l'affidabilità, e il fatto di non dotarsi di un buon sistema di storage potrebbe determinare la buona o cattiva riuscita di una strategia aziendale. In questo ambito, una buona soluzione potrebbe essere trovata nell'uso di sistemi Cloud, che

possono garantire alcune importanti caratteristiche per i dati come ad esempio flessibilità, portabilità e dinamicità.

Pertanto, la *data analytics* prende come input questi big data e li sottopone al proprio processo che, come verrà spiegato in seguito, è composto da diverse fasi, si basa sull'applicazione di algoritmi e spesso si fonda su princìpi matematici, con lo scopo finale di ottenere informazioni utili. Uno dei principali obiettivi dell'analisi dei dati presso le imprese è sicuramente l'aumento del fatturato (eventualmente per mezzo della riduzione di costi): a volte è sufficiente analizzare attentamente i propri dati per migliorare le performance o per capire quale fetta di mercato è più interessante; altri obiettivi non trascurabili sono sicuramente legati alla previsione della domanda (ad esempio tramite l'osservazione di serie storiche), o al miglioramento dei rapporti con i clienti, proponendo azioni di vendita mirate.

Per ottenere risultati di questo genere, alcuni studi hanno diviso i processi di *data analytics* in tre categorie, separate in base al proprio scopo e alla propria metodologia:

- Analisi descrittiva: utile a descrivere una situazione attuale o passata tramite dati e visualizzarla, anche graficamente, per mezzo di semplici report in maniera sintetica.
- Analisi predittiva: si tratta di un'analisi più complessa che come obiettivo si pone quello di prevedere un comportamento futuro basandosi sulle informazioni del passato. Sfrutta modelli matematici come ad esempio la regressione e algoritmi informatici di *data mining*.
- Analisi prescrittiva: comprende applicazioni avanzate che, dopo aver svolto le proprie analisi, riescono a proporre soluzioni operative rispetto al problema di partenza.

A queste tre categorie ne va aggiunta una quarta, l'analisi automatizzata, che ha come obiettivo quello di implementare autonomamente l'azione ideata dal risultato delle analisi. Quest'ultima tipologia di analisi però è ancora poco conosciuta presso grandi e medie aziende e la sua diffusione è presente quasi esclusivamente a livello di pilota, mentre invece le prime due categorie sono ormai utilizzate dalla maggioranza delle grandi imprese e da una buona percentuale, in crescita, delle PMI.

Nella **figura 3.1** viene mostrato uno schema che descrive la crescita del mercato dell'*analytics* in Italia secondo gli studi di una ricerca condotta dall'Osservatorio Big Data Analytics & Business Intelligence del Politecnico di Milano [12]. Come si può facilmente notare si tratta di un mercato il cui valore complessivo supera il miliardo di euro ed è in costante crescita: infatti questo andamento riflette il sempre maggiore orientamento delle imprese nei confronti dell'estrazione e dell'analisi dei big data.

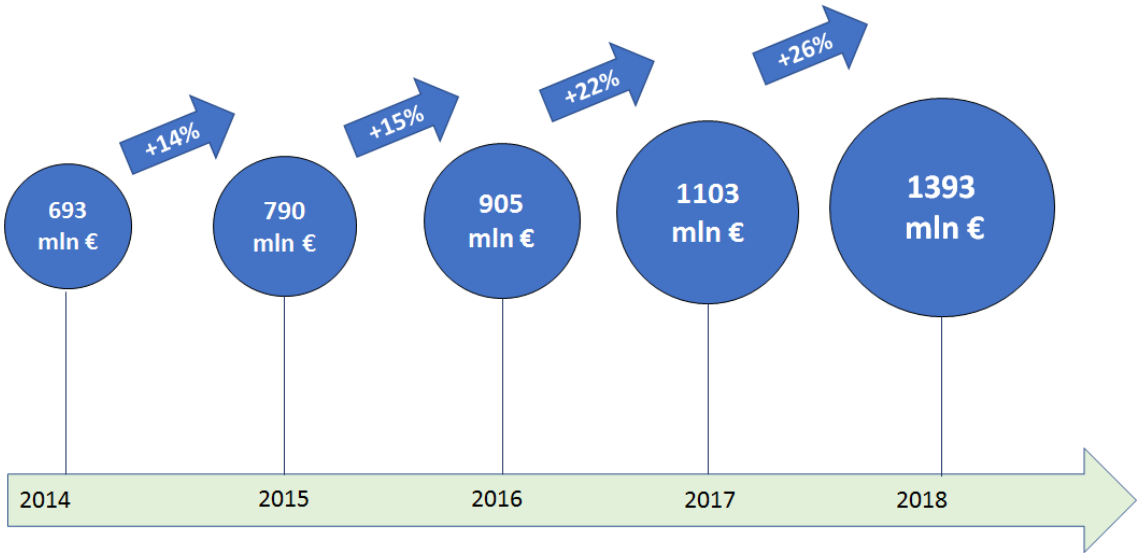

Figura 3.1: la crescita del mercato dell'analytics negli ultimi anni in Italia

Come già accennato precedentemente, al giorno d'oggi la *data analytics* ha moltissimi campi di applicazione per il semplice motivo che le imprese, operative in tutti i settori, possono raccogliere dati e farne l'uso che preferiscono. Pertanto, in seguito verranno descritti alcuni esempi di possibili analisi per chiarire al meglio quali sono gli utilizzi possibili e le potenzialità di questa scienza. Nella **figura 3.2** vengono dunque elencati i settori nei quali l'analisi dei dati è più presente, descrivendone anche il trend di crescita oltre che la quota di mercato. Le banche, e in generale il mondo della finanza, hanno occupato, nel 2017, la fetta più ampia del mercato, effettuando grandi investimenti in questa direzione: uno degli utilizzi più diffusi da parte delle banche è, ad esempio, l'analisi dei prelievi e delle tipologie di spesa da parte dei propri clienti con lo scopo di evitare scambi di identità o eventuali frodi. Anche il settore manifatturiero, spinto dalla

nascita del modello Industry 4.0, ha occupato una posizione di rilievo nel mercato. Uno dei suoi principali scopi dell'utilizzo dei dati potrebbe essere l'elaborazione di modelli di manutenzione predittiva (proprio come nel caso di studio portato avanti in questa tesi) per ottenere una previsione dei guasti dei macchinari prima che essi avvengano.

Mercato Analytics in Italia nel 2017		
Settore	Quota di mercato	Trend di crescita
Assicurazioni	6%	molto elevato
Banche	28%	elevato
Retail	7%	elevato
Manifatturiero	24%	molto elevato
Sanità	7%	stabile
Servizi	8%	molto elevato
Telecomunicazioni	14%	elevato
Utility	6%	stabile

Figura 3.2: i settori maggiormente attivi nell'ambito dell'analisi dei dati

Per quanto riguarda i settori di media e telecomunicazioni, le aziende analizzano i dati dei propri clienti per prevedere il tasso di abbandono o per potenziare la loro relazione con essi tramite offerte personalizzate e nuove campagne di marketing. Non va scordato poi il settore sanitario dove i dati sui pazienti vengono usati per ottenere informazioni sull'efficacia di determinate cure o per prevenire il manifestarsi di una malattia. Le assicurazioni, che nel 2017 occupavano una minore fetta di mercato, mostrano però un trend di crescita molto elevato, siccome tramite lo studio dei dati passati è possibile tracciare i profili di diverse tipologie di clienti a cui, per esempio, proporre offerte ad hoc.

Per impostare nel modo corretto un buon progetto di analisi di big data è necessario avere più o meno chiara la sua finalità nel business dell'impresa per evitare che alla fine risulti inutile e non porti alcun vantaggio a chi può sfruttarne i risultati. Una volta definita la direzione da far prendere al progetto vanno seguiti questi step:

1. Selezione e pulizia dei dati: in questa prima fase si "entra in confidenza" con i dati a disposizione e si applica la pulizia per evitare errori o informazioni inutili o fuorvianti negli stessi (ad esempio, i dati incompleti possono essere sostituiti o eliminati mentre i dati rumorosi, ossia quelli troppo diversi dagli altri, vanno di norma rimossi).
2. Trasformazione dei dati: essi vanno preparati appositamente per avere una struttura utile per l'analisi. Ad esempio, possono essere utilizzate tecniche di normalizzazione su valori numerici oppure di discretizzazione per dividere i valori di un attributo continuo in diversi intervalli. Inoltre, nel caso in cui alcuni attributi dei dati si rivelassero inutili o ridondanti, potrebbero essere eliminati dal dataset tramite un'operazione di *features selection*.
3. Costruzione vera e propria del modello di analisi dei dati e sua conseguente applicazione. Come prima cosa bisogna decidere quale tipo di analisi effettuare, dopodiché si passa alla scelta dell'algoritmo più appropriato. I modelli possono essere inizialmente utilizzati solo su una porzione di dati per essere testati: qualora i risultati ottenuti non fossero soddisfacenti i modelli potrebbero essere revisionati e in seguito testati nuovamente. Una volta ottenuta la versione finale del modello, esso viene utilizzato definitivamente su tutti i dati a disposizione.
4. Valutazione, interpretazione e comunicazione dei risultati ottenuti tramite tecniche di visualizzazione di informazioni di semplice lettura, in modo da poter aiutare nelle operazioni di *decision making.*

Le prime due fasi fanno parte del cosiddetto *preprocessing*, mentre la terza è il vero cuore dell'analisi (infatti solo in questa parte vengono applicati gli algoritmi utili per il raggiungimento dello scopo finale). Le tecniche più utilizzate, come il clustering, i modelli di classificazione e l'estrazione di regole di associazione, fanno parte dell'ambito del *data mining*, metodologia che sfrutta teorie statistiche e sistemi di gestione di database per effettuare un'analisi di dati che può essere descrittiva o predittiva, e che sempre più spesso ormai si serve di strumenti come l'intelligenza artificiale e il *machine learning* per far sì che i modelli generati migliorino le proprie

performance man mano che le analisi procedono e che nuove informazioni vengono generate ed elaborate.

3.2 – *MACHINE LEARNING*

Il *machine learning* [13] [14] [15] può essere considerato come un insieme di tecniche basate sull'idea che i computer siano in grado di apprendere e fare scelte in modo quasi totalmente autonomo, con l'intervento umano ridotto al minimo, semplicemente sottoponendo grandi porzioni di dati a determinati algoritmi in maniera iterativa. Tali metodi permettono alle macchine di individuare nuovi pattern e informazioni a partire dai dati senza che gli si venga detto in modo esplicito dove e come ricavarli, ma la cosa più interessante è sicuramente il fatto che le prestazioni di modelli e algoritmi migliorano se questi ultimi vengono ripetuti più volte. Infatti, le elaborazioni precedenti "insegnano" ai computer quali strade percorrere, facendo sì che essi possano adattarsi in modo sempre migliore ai dati che volta dopo volta vengono processati, producendo come output risultati più precisi e replicabili.

Il *machine learning* basa i suoi concetti su una branca dell'informatica teorica detta "teoria dell'apprendimento". Tale teoria sostiene che l'apprendimento automatico di una macchina avviene nel momento in cui essa è in grado di generalizzare dalla propria esperienza, ovvero quando riesce ad occuparsi di dati o compiti nuovi, mai affrontati, dopo aver imparato a gestire quelli vecchi (anche detti di "train"). Secondo questa teoria la macchina può compiere ragionamenti induttivi, passare dal caso particolare a quello generale, ed elaborare un modello probabilistico che possa essere valido per il più grande numero possibile di casi. Ovviamente, siccome i dati di train rappresentano un insieme finito e limitato rispetto a tutte le casistiche possibili, è logico che la teoria dell'apprendimento automatico non possa garantire sempre ottime prestazioni, siccome limitata da vincoli probabilistici. Pertanto, i computer compiranno sempre alcuni errori in fase di generalizzazione, nonostante la complessità del modello e la vastità dei dataset che esso ha già affrontato: è proprio qui che si fa più interessante l'intervento umano nel processo di apprendimento del computer. È l'uomo infatti che ha il compito di "guidare" la macchina, portarla verso risultati migliori fornendole un buon modo per

"allenarsi" e cercando di massimizzare le sue prestazioni. Oltre a tutto questo l'uomo ha poi sicuramente interesse nel valutare la fattibilità temporale dell'apprendimento della macchina: se esso avvenisse in tempi troppo lunghi non sarebbe probabilmente più utile allo scopo finale.

La maggior parte degli algoritmi di *machine learning* ha come obiettivo quello di identificare una funzione che associ i dati passati come input ad una variabile di output. Questa funzione non è osservabile dall'esterno, e la macchina la impara direttamente dai dati: è quindi chiaro che più dati si hanno a disposizione, più la macchina avrà informazioni disponibili per provare a capire che struttura abbia questa funzione. In ogni applicazione reale però non vale quasi mai la semplice relazione $Y = f(x)$ a causa della presenza di un errore: infatti ci sarà sempre qualche dato che non può essere spiegato da un modello matematico. Lo scopo finale comunque non è quello di conoscere la forma di questa funzione che verrà usata, nella maggior parte dei casi, per fare predizioni, ma è ottenere la massima precisione nei risultati. Per questo motivo, nelle applicazioni che sfruttano il *machine learning* buona parte del tempo viene impiegata in tentativi di miglioramento della stima di questa funzione invisibile che si nasconde tra i dati, affinché le predizioni dei modelli che ne derivano siano sempre migliori. Ogni algoritmo lavora in modo differente (ad esempio facendo diverse assunzioni sulla forma della funzione, sulla sua linearità o non-linearità) e quindi diventa importante provarne diversi in modo da capire quale si possa adattare meglio ai dati.

Esistono diversi metodi utilizzati per il *machine learning*, e la **figura 3.3** li sintetizza tramite uno schema, proponendo anche alcuni esempi di applicazione:

- Apprendimento supervisionato: in questa tipologia la macchina viene addestrata tramite l'uso di serie storiche. Per ogni dato facente parte del dataset di train si ha a disposizione l'informazione di interesse, detta "etichetta", utile per generare il modello e testarlo. Una volta superata questa fase, il modello viene usato su dati nuovi di cui non si conosce ancora l'etichetta, in modo da poterla predire. I modelli predittivi sono infatti una delle applicazioni più diffuse tra quelle che sfruttano l'apprendimento supervisionato, proprio perché in questi casi si ritiene che i dati storici siano in grado di poter predire quelli futuri. Le metodologie più

usate sono, ad esempio, la classificazione, dove ad ogni dato si associa appunto un'etichetta, oppure la regressione, dove invece ai dati si assegna come output un valore numerico coerente con il modello creato.

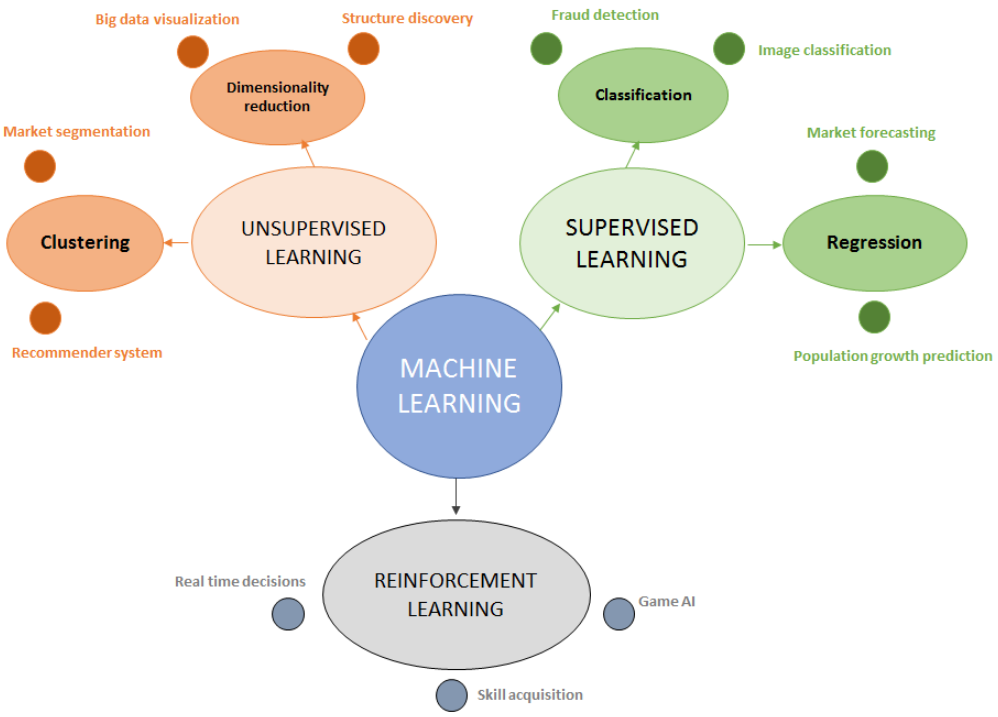

Figura 3.3: Metodi più utilizzati di machine learning

- Apprendimento non supervisionato: in questa seconda casistica non viene fornita alcuna informazione su un'eventuale classificazione interna ai dati, ma è l'algoritmo stesso che ha il compito di trovare, se esistono, strutture interne ai dati tali da segmentarli in gruppi che siano il più coesi possibile tra loro. Un tipico esempio di apprendimento non supervisionato è il clustering, spesso utilizzato come metodo per individuare consumatori con caratteristiche simili tra loro (che però non siano conosciute a priori), oppure, nella fase di *preprocessing*, per identificare ed eliminare valori anomali presenti tra i dati.
- Apprendimento per rinforzo: in questo caso l'apprendimento della macchina avviene tramite un meccanismo basato su ricompense. Dopo aver compiuto le

sue azioni, l'algoritmo scopre, passando per diversi tentativi ed errori, quali tra di esse portano a maggiori premi. Ovviamente il suo obiettivo è quello di massimizzare il valore di questi ultimi e pertanto sarà portato a scegliere quelle azioni considerate giuste, raggiungendo l'obiettivo in un lasso di tempo sempre minore. I principali utilizzi di questo metodo si hanno nei campi della robotica e dei videogiochi.

- Apprendimento semi supervisionato: si trova a metà strada tra l'apprendimento supervisionato e quello non supervisionato siccome nel dataset di train vengono inseriti dati etichettati insieme a dati non etichettati (la cui acquisizione è sicuramente meno onerosa). L'obiettivo finale è sempre e comunque la creazione di un modello predittivo e questo metodo viene usato quando non si ha a disposizione una quantità sufficiente di dati già classificati o quando raccogliere questi dati è troppo complicato o costoso.

Molto spesso il *machine learning* viene utilizzato in concomitanza con il *data mining*, e siccome molte delle tecniche finora citate, come clustering e classificazione, si utilizzano in entrambi i casi, occorre fare chiarezza andando ad analizzare le differenze tra essi. Il *data mining* si focalizza sull'estrazione di informazioni e pattern utili e finora sconosciuti sfruttando la potenza dei dati, e per fare ciò utilizza svariate tecniche tra le quali figurano anche quelle del *machine learning*. Per quanto riguarda quest'ultimo invece, le sue prestazioni sono valutate in base alla capacità di fare previsioni e riprodurre conoscenza già precedentemente acquisita.

Il valore della tecnologia del *machine learning* è attualmente ormai riconosciuto dappertutto, specialmente in quei settori dove ogni giorno vengono raccolte grandi quantità di dati. Utilizzando le tecniche sopra descritte molte società sono riuscite a guadagnare un vantaggio competitivo sulle rivali, andando anche ad aumentare i propri profitti.

Tra i principali utilizzi del *machine learning* al giorno d'oggi si possono citare buona parte delle tecnologie cardine dell'Industry 4.0: ad esempio il riconoscimento vocale e quello facciale (ormai ampiamente usati dai nuovi smartphone: più queste applicazioni

vengono usate, più saranno rapide ed accurate le loro risposte), l'analisi di video, di immagini e di file testuali, sistemi di traduzione e trascrizione automatica (come il traduttore di Google), nonché tutta la famiglia degli assistenti digitali e dei bot intelligenti (come Alexa, Cortana e Siri, sviluppate rispettivamente da Amazon, Microsoft e Apple).

Tutte queste tecnologie, che fino a pochi anni fa, sebbene già esistessero, non funzionavano assolutamente bene come invece adesso fanno, hanno migliorato notevolmente i loro risultati utilizzando le tecniche di apprendimento automatico, sfruttando le informazioni passate per affrontare le sfide future.

Il *machine learning* è anche utilizzato in alcune strategie di marketing, soprattutto online, per esempio da alcuni social network come Facebook o Instagram, dove gli algoritmi, basandosi sui passati comportamenti degli utenti sulla piattaforma, predicono interessi o consigliano articoli e pagine. Circa lo stesso fatto avviene quando siti di vendita online come Amazon, suggeriscono nuovi acquisti da fare in base a quelli effettuati da consumatori con caratteristiche simili a quelle di chi sta acquistando (lo stesso discorso è valido anche per Netflix che calcola un punteggio di compatibilità per gli utenti rispetto alle varie serie tv e film proposti).

Altri settori nei quali gli algoritmi di *machine learning* sono ormai ampiamente diffusi sono quello sanitario (dove, grazie anche alla nascita di tecnologie e sensori indossabili, è possibile verificare lo stato di salute de pazienti in pochissimo tempo o identificare segnali di allarme), quello dei trasporti (ad esempio per cercare nuove rotte e identificare i migliori percorsi per automobili e mezzi pubblici: tipico esempio di tecnologia usata in questo modo è Google Maps, che ormai funge perfettamente da navigatore dopo aver raccolto dati geografici per anni) e quello delle risorse (per trovare giacimenti di gas e petrolio analizzando i dati del suolo, ad esempio).

Sempre restando nell'ambito dell'industria 4.0, la *smart factory* sfrutta algoritmi di *machine learning* per diversi scopi: durante la catena di produzione, sistemi intelligenti sono in grado di verificare la presenza di oggetti e riconoscerne le caratteristiche, come le dimensioni o il colore, e valutare la presenza di eventuali errori. Questi sistemi vengono, pertanto, utilizzati nel controllo della qualità, aiutando le fabbriche ad essere

più efficienti. L'altra importante applicazione di questi algoritmi nel settore manifatturiero è sicuramente la manutenzione predittiva, che utilizza sensori per catturare dati dalle macchine in funzionamento e che verrà spiegata in seguito più approfonditamente, trattandosi di uno degli argomenti chiave di questa tesi.

3.3 – TECNICHE DI CLASSIFICAZIONE

Gli algoritmi di classificazione fanno parte del *data mining* e utilizzano metodi di apprendimento automatico supervisionato con l'obiettivo di fare predizioni sui dati. In particolare, come input viene fornito un set di dati già suddivisi ("etichettati") in due o più classi di appartenenza grazie ai quali viene creato un modello di classificazione, che in seguito verrà utilizzato su dati nuovi (non etichettati) per assegnarli alla classe appropriata.

Il dataset di partenza viene solitamente diviso in due gruppi, ossia dataset di train, che viene usato per creare il modello, e dataset di test, che ha lo scopo di testare il modello. La validazione del modello avviene seguendo delle tecniche particolari di partizionamento, come ad esempio la *Cross Validation*. Quest'ultima funziona suddividendo il dataset in un certo numero (K, scelto a priori) di gruppi: a rotazione su tutti i gruppi, uno di essi fungerà da test set e tutti gli altri da training set. La *Cross Validation* funziona generalmente bene su molti tipi di dataset, ma all'occorrenza esistono altri metodi che svolgono la stessa funzione (come ad esempio un partizionamento fisso, usato per dataset di dimensioni molto grandi).

Un tipico esempio di applicazione delle tecniche di classificazione si ha quando le banche valutano se fornire un prestito ai propri clienti analizzandone le caratteristiche personali. In base ai dati storici infatti è possibile costruire un modello che associ ogni persona ad un'etichetta che indica la possibilità di una frode oppure un'alta probabilità di restituire il debito contratto. Nell'ambito di questa tesi invece, sono stati creati dei classificatori da utilizzare per un progetto di manutenzione predittiva.

Esistono molteplici tecniche di classificazione che presentano notevoli differenze tra di loro. Le più conosciute sono le seguenti:

- Alberi di decisione: algoritmi con struttura ad albero dove ogni nodo rappresenta un determinato test sugli attributi dei dati e ogni ramo è una "strada" che viene percorsa in base all'esito del test. I nodi finali sono l'etichetta a cui ogni dato può essere associato. I suoi punti di forza sono la sua interpretabilità, l'efficienza e una buona accuratezza mentre il principale punto di debolezza riguarda la sensibilità ai dati mancanti.

- Random Forest: si tratta di classificatori che mettono insieme i risultati di più alberi di decisione per ottenere maggiore accuratezza. Il loro punto di debolezza però riguarda una minore scalabilità rispetto alla dimensione dei training set.

- Classificazione bayesiana: si basa sul calcolo della probabilità che un dato appartenga a una certa classe. Si tratta di un classificatore accurato e con discreta interpretabilità ma la generazione del modello è molto lenta in caso di dataset corposi. Per far fronte a questo problema è spesso necessario introdurre l'ipotesi di indipendenza statistica tra gli attributi del dataset (detta ipotesi Naïve, che però rischia di semplificare troppo il modello, riducendone l'accuratezza).

- K-Nearest Neighbors: algoritmo basato sul calcolo della distanza (spesso si utilizza quella euclidea) tra gli elementi del dataset. Ad esempio, un dato viene assegnato ad una determinata classe se sufficientemente vicino agli altri dati della medesima classe. Il parametro K rappresenta il numero di dati vicini presi in considerazione durante l'assegnazione alle classi. Il K-NN rischia di diventare computazionalmente oneroso a causa del calcolo delle distanze tra i dati, soprattutto nei casi in cui siano presenti molti attributi.

- Reti neurali [16]: si tratta di tecniche molto accurate e robuste in caso di dati mancanti o outliers che però presentano una scarsa interpretabilità e un processo di apprendimento lento. Il loro funzionamento ricorda quello del cervello umano: ogni nodo, che rappresenta il neurone, riceve i dati, li elabora e trasmette

ai nodi successivi i dati e le proprie analisi: così facendo i nodi dei livelli successivi ottengono informazioni sempre più dettagliate.

Gli output dei vari algoritmi di classificazione possono essere valutati tramite il calcolo di alcune metriche che ne verificano la qualità, in modo da capire se il modello creato sta funzionando bene o necessita alcuni aggiustamenti:

- L'accuratezza del modello, che si calcola come rapporto tra il numero di dati classificati correttamente e il numero totale di dati presenti nel dataset di test.

- Il richiamo e la precisione, calcolati per ogni diversa classe. Il primo è il rapporto tra i dati classificati correttamente in una determinata classe e i dati totali appartenenti alla stessa classe, mentre la seconda è il rapporto tra i dati classificati correttamente in una determinata classe e il numero di dati assegnati a tale classe.

Richiamo e precisione devono essere calcolati perché la sola accuratezza non basta per poter descrivere l'output del modello, soprattutto in caso di dataset sbilanciati nella distribuzione delle classi.

3.4 – MANUTENZIONE PREDITTIVA

La manutenzione predittiva [17] [18] è un metodo di manutenzione innovativa che, insieme allo sviluppo delle nuove tecnologie nelle fabbriche e all'affermazione della quarta rivoluzione industriale, negli ultimi anni ha avuto una rapida diffusione nell'ambito della produzione industriale, e non solo.

Fin dalla nascita delle prime fabbriche, che possiamo collocare nel periodo della prima rivoluzione industriale, sono sempre state effettuate operazioni di manutenzione sulle macchine impiegate nella produzione, con la differenza che inizialmente, di fronte al possibile guasto di uno dei macchinari, la reazione dei manager era quella attenderlo e in seguito di chiamare qualcuno per aggiustarlo, in modo da riprendere subito dopo il lavoro. L'idea di attendere la rottura delle macchine e solo in seguito porvi rimedio è ciò

che attualmente viene chiamata manutenzione reattiva, e si tratta di una soluzione grezza e poco efficiente.

Per evitare perdite di tempo ed eccessivi costi di riparazione (da quando la macchina si rompe fino a che viene aggiustata si ha infatti un'interruzione del lavoro con una conseguente diminuzione della produttività, e il costo da affrontare ogni volta che un macchinario si guasta non è indifferente), il modello di manutenzione dominante è cambiato, passando a quella che viene definita manutenzione preventiva. Pur non essendo ancora il metodo ideale, quest'ultima, che consiste essenzialmente nell'effettuare controlli periodici sulle macchine e nell'intervenire prima che i guasti si verifichino se si notano delle anomalie, presenta alcuni vantaggi rispetto alla precedente, in quanto ha permesso di allungare la vita degli impianti senza aumentare eccessivamente i costi di gestione della fabbrica, pianificando il meglio possibile le operazioni di controllo. Tuttavia, questo sistema presenta ancora diverse falle: trattandosi di interventi pianificati ad intervalli regolari, è molto frequente trovarsi nella situazione in cui la macchina sia in perfetta salute durante un controllo oppure, al contrario, essa si guasti improvvisamente in un periodo in cui tali verifiche non sono state pianificate, comportando in entrambi i casi un inutile aumento di costi.

Queste problematiche hanno portato alla nascita di diversi studi secondo i quali la soluzione ottimale è quella di monitorare il lavoro dei macchinari e trarre conclusioni sul livello del loro funzionamento, predire dei buoni o cattivi risultati e infine rendersi conto se essi sono in buono stato di salute o stanno per guastarsi. Da questo concetto nasce la manutenzione predittiva, che permette a chi gestisce le fabbriche di conoscere in anticipo i fallimenti dei propri impianti e di effettuare azioni correttive prima che i guasti si verifichino, risparmiando tempo e denaro. Oltre a portare vantaggi in termini economici e di produttività dei macchinari, questo tipo di manutenzione porta sicuramente dei miglioramenti per quanto riguarda la sicurezza nelle fabbriche: gli operai infatti, lavorando a stretto contatto con le macchine, avranno più garanzie sullo stato di funzionamento di queste ultime. Grazie a questi importanti vantaggi buona parte delle grandi imprese manifatturiere sta iniziando ad adottare questo sistema, che entro qualche anno diventerà uno standard operativo. Tuttavia, la sua implementazione, specialmente all'inizio, può risultare complessa, specialmente se paragonata alle

tipologie di manutenzione precedenti dove l'unica difficoltà era quella di programmare gli interventi periodici di riparazione.

Per ottenere vantaggi dall'applicazione della manutenzione predittiva è necessario infatti sviluppare determinate competenze, sia per saper interpretare gli output del modello generato, sia per capire su quali impianti applicarla. Infatti, cercare di monitorare il funzionamento di tutti i macchinari presenti in una fabbrica potrebbe essere molto costoso e addirittura inutile. La manutenzione predittiva andrebbe applicata preferibilmente su quei macchinari critici, il cui blocco impedirebbe completamente l'avanzamento dei lavori provocando danni per tutta la produzione. Un altro motivo per cui è necessario scegliere con cura su quali macchinari effettuare questo tipo di manutenzione è che essa è specifica per ognuna di essi, e non è intercambiabile, proprio perché ogni impianto lavora a suo modo e ci fornisce tipologie di dati differenti: i modelli di manutenzione si adattano alle macchine e siccome per elaborare un modello da capo potrebbero volerci addirittura mesi, ecco spiegato il perché non si debba scegliere male l'impianto da controllare.

Capire quando una macchina sta smettendo di funzionare bene non è così immediato comunque: per mettere in atto un piano di manutenzione predittiva è necessario come prima cosa raccogliere una vasta quantità di dati dalla macchina in fase di lavoro, e tali dati devono essere "lo specchio" della bontà del suo stato di funzionamento, in modo che, leggendo delle anomalie in tali informazioni, il modello sia in grado di segnalarle tempestivamente. La fase di raccolta dei dati può durare intere settimane e addirittura mesi (ciò dipende infatti dalla loro complessità, oltre che dal modo in cui funziona la macchina stessa), ed è molto importante che essi siano raccolti con strumenti adeguati (dati di scarsa qualità infatti sono completamente inutili per un lavoro di questo tipo).

Una volta terminata l'acquisizione dei dati, segue un vero e proprio progetto di *data analytics* in cui si susseguono tutte le fasi che sono state descritte in precedenza: estrazione, pulizia e trasformazione dei dati, scelta degli algoritmi migliori, creazione e test di un modello predittivo. Quest'ultimo dev'essere in grado di definire lo stato di salute dei macchinari e di prevederne i guasti, e chiaramente deve poter "imparare" dai dati forniti per migliorare le proprie performance, sfruttando quindi le tecniche di *machine learning*. L'intervento umano nella manutenzione predittiva si limita alla

creazione del modello e all'interpretazione dei risultati, il resto consiste in un processo automatico dove i nuovi dati generati dalle macchine vengono raccolti ed elaborati dal modello. Esso, oltre ad avvisare nel caso in cui gli impianti necessitino riparazione, può anche informare i responsabili circa il tipo di manutenzione da svolgere e su quale componente sta per rompersi. In **figura 3.4** è schematizzato un riassunto delle 3 principali fasi che compongono il progetto e l'attuazione della manutenzione predittiva.

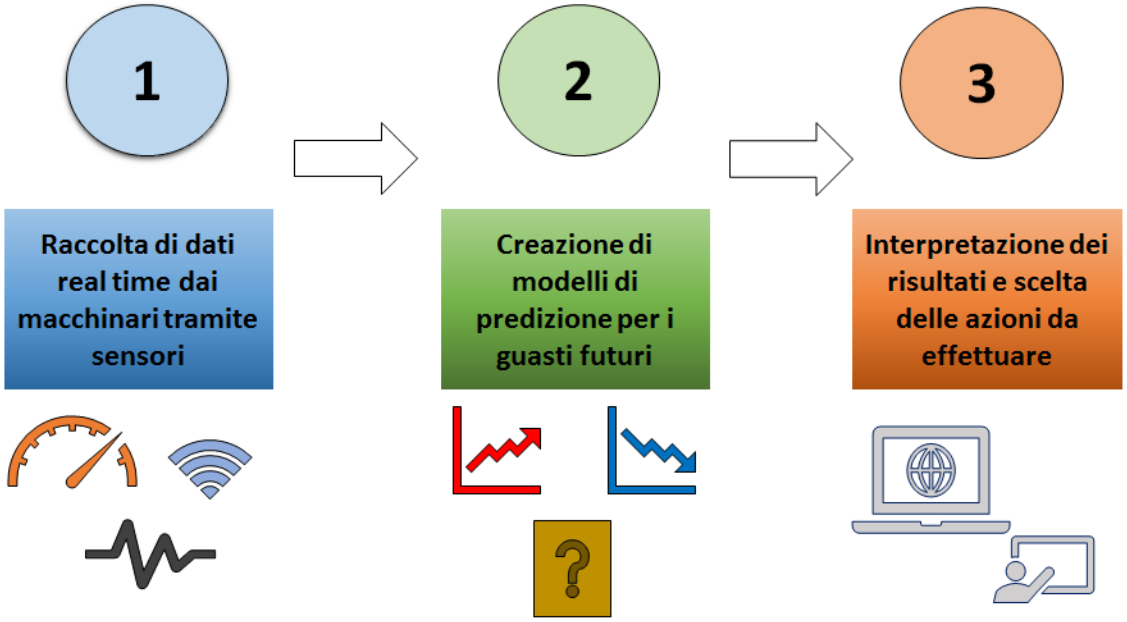

Figura 3.4: Schema riassuntivo della manutenzione predittiva

La maggior parte delle applicazioni di manutenzione predittiva utilizza, come metodo di analisi di dati, le principali tecniche di *data mining*: il clustering viene sfruttato per la sua capacità di segmentazione dei dati e per la possibilità di distinguere in questa maniera il buono e il cattivo funzionamento, mentre i classificatori e le regole di associazione spesso permettono di capire quali relazioni intercorrono tra i dati e quali siano le cause che portano a un eventuale guasto. Inoltre, possono essere adoperate anche tecniche di regressione, ad esempio per evidenziare un trend tra i dati, o metodi di analisi statistica per capire se il processo è in controllo o se sta lentamente variando le sue caratteristiche (queste tipologie di monitoraggio sono ampiamente usate nell'ambito

del controllo della qualità, e prendono il nome di "carte di controllo": il loro principale scopo è proprio quello di identificare rapidamente i punti fuori controllo, ovvero quelli troppo lontani dal valore medio degli altri dati, e capire se si trattano di semplici anomalie o se è necessario intervenire).

Esistono numerose tipologie di dati che possono essere raccolte ed analizzate nell'ambito di un progetto di manutenzione predittiva: nel caso di studio affrontato in questa tesi si è trattato di un segnale di corrente, ma ogni macchina ha il suo personale "linguaggio" per trasmettere informazioni sul suo funzionamento. Ad esempio esistono analisi di tipo acustico che permettono di percepire, tramite ultrasuoni, lo stress applicato sui macchinari durante importanti spostamenti di carichi, oppure analisi per mezzo di raggi infrarossi, le quali vengono usate per controllare le componenti dei macchinari e svelare la presenza di problemi di varia natura, sia meccanici che elettronici. Inoltre, anche le analisi sullo stato dell'olio sono molto diffuse per verificare lo stato di lubrificazione di alcune componenti meccaniche o anche per controllare il loro livello di qualità.

3.4.1 – STATO DELL'ARTE

In quest'ultima sezione del capitolo verranno presentati alcuni esempi di applicazioni reali della manutenzione predittiva, con lo scopo di far comprendere in quali ambiti sia già diffuso il suo utilizzo al giorno d'oggi.

Un tipico caso d'uso della manutenzione predittiva è la rilevazione di vibrazioni anomale da un motore [19]. Tempo prima che si verifichi un guasto infatti un motore può emettere piccole vibrazioni che con il passare del tempo diventano sempre più evidenti: esse sono i primi segnali di un qualche problema che sta iniziando a manifestarsi. Se il motore non viene riparato in tempo può poi presentare altri sintomi di malfunzionamento come l'usura, un calo delle prestazioni o rumori e temperature insolite. Più tempo passa, più è alta la probabilità di un guasto più grave e più l'intervento di un meccanico può essere difficile e costoso: un intervento immediato, ovvero non appena le prime vibrazioni anomale vengono rilevate, può permettere un

risparmio di costi e assicurare che, dopo un piccolo aggiustamento, il motore continuerà a funzionare al solito modo. Tramite alcuni sensori specifici comunque, è possibile percepire le piccole vibrazioni iniziali e permettere un intervento mirato ben prima che il problema diventi grave. Questi sensori infatti trasmettono i dati raccolti sulle vibrazioni al modello informatico che viene adeguatamente interpretato da chi di dovere.

Questo sistema di manutenzione si sta sviluppando, oltre che nell'ambito della produzione industriale, anche in altri settori: per esempio, molte compagnie aeree stanno investendo i propri capitali in questa tecnologia per poter applicare sulle proprie flotte diverse tecniche di manutenzione in modo da evitare guasti ai velivoli e ottenere un forte risparmio di costi. Un'indagine Honeywell [20] ha affermato che tra le compagnie aeree, circa il 69% di esse aumenterà il budget destinato a queste operazioni e che, per esempio, EasyJet ha implementato e sta già utilizzando oltre 50 algoritmi predittivi sui propri aerei. La compagnia aerea low-cost infatti ha cominciato ad effettuare delle operazioni di manutenzione predittiva già durante il volo dell'aereo, monitorando i principali segnali emessi dal velivolo. Così facendo, i dati vengono inviati a terra e analizzati da tecnici e ingegneri, che in caso di guasto e malfunzionamento previsto, possono segnalare i problemi e programmare la sostituzione dei pezzi difettosi ancora prima che l'aereo atterri [21]. Grazie a questa strategia è possibile per EasyJet risparmiare tempo (programmare in anticipo le sostituzioni delle componenti può evitare ritardi nei voli successivi) e, cosa altrettanto importante, monitorare gli aerei durante il loro effettivo funzionamento. Infatti, se la manutenzione venisse effettuata solamente a terra, sarebbe impossibile scovare tutti i problemi a causa delle condizioni completamente diverse: temperatura, pressione e vibrazioni analizzate durante la fase di volo influiscono parecchio sul monitoraggio dei velivoli.

Un altro caso di applicazione della manutenzione predittiva è il monitoraggio dello stato di salute dei riduttori delle turbine eoliche. A tale scopo è stato avviato un progetto apposito, chiamato SIMAP [22] (sigla che sta per "sistema intelligente per la manutenzione predittiva") finalizzato alla progettazione di un calendario di manutenzione dinamico, ottimizzato per le esigenze e la vita operativa delle turbine

eoliche. Esso, oltre ad individuare anomalie nel funzionamento, consente anche la pianificazione anticipata di interventi di riparazione. Vi sono diversi parametri che possono essere studiati e registrati durante il funzionamento di una turbina eolica. Anche in questo caso molto spesso è necessario fare attenzione alle vibrazioni emesse, facendo però attenzione al fatto che spesso esse dipendono anche dalla velocità di rotazione delle turbine stesse: a velocità basse i sensori potrebbero non cogliere i segnali di un possibile guasto. Oltre alle vibrazioni, vengono inoltre monitorate l'emissione acustica, che può essere molto utile per individuare dei problemi ancora in fase iniziale, persino a velocità basse di rotazione, la temperatura e i segnali di corrente e potenza durante la fase operativa. Dopo esser stato testato con le turbine, SIMAP è stato anche impiegato per la diagnosi in tempo reale nei processi industriali.

Infine, un altro esempio di applicazione della manutenzione predittiva riguarda le reti ferroviarie [23]. In particolare, sono stati sviluppati sistemi per rilevare importanti informazioni tramite dei sensori posti sulla rete, come ad esempio dati sulla temperatura, sulla deformazione di parti meccaniche, sull'impatto e sul peso di varie componenti. Lo scopo finale è quello di evitare il più possibile interruzioni di servizio e quindi aumentare la velocità media sul percorso della rete ferroviaria per far fronte alla crescente domanda di servizi e carico sulla rete stessa. Per sviluppare un sistema del genere sono stati sfruttati migliaia di dati storici riguardanti i guasti passati, le tipologie di treni che percorrono la rete e persino dati metereologici: incrociando tutte queste informazioni il modello generato è in grado di avvisare in caso di rischi di guasto e pianificare un intervento di manutenzione. Per quanto riguarda l'Italia, anche Trenitalia sta mettendo in atto un ambizioso piano di investimenti per attuare questo tipo di manutenzione [24]. L'obiettivo è quello di posizionare un elevato numero di sensori (da 500 a 1000) sui treni stessi in modo tale da verificarne lo stato di funzionamento in real time e poter intervenire solamente in caso di vero bisogno. Il piano stima all'incirca 1,3 miliardi di euro l'anno risparmiati per la manutenzione e una riduzione di circa 15 milioni di euro annui per i costi del disservizio (tra i quali figurano i rimborsi ai passeggeri). Attualmente i parametri studiati riguardano principalmente lo stato di funzionamento della batteria e dei sistemi frenanti dei treni.

4 – CASO DI STUDIO

Dopo aver descritto dettagliatamente il contesto che circonda il lavoro effettuato in questa tesi nei capitoli 2 e 3, in questo quarto capitolo si esporranno le caratteristiche del caso di studio affrontato. Esso riguarda, come già accennato in precedenza, un progetto di manutenzione predittiva nell'ambito di una produzione industriale snella ed efficiente, tipica del modello di fabbrica *smart* e dell'industria 4.0. I dati raccolti e processati provengono dal lavoro di alcuni bracci robotici, che sono stati monitorati grazie a diversi sensori in modo tale da poter comprendere lo stato di funzionamento di tali parti meccaniche che solitamente vengono settate da operai e in seguito lasciate ai propri compiti svolti in automatico. L'operato della società a cui appartengono questi bracci robotici si concentra soprattutto nella realizzazione di prodotti e servizi di automazione, come ad esempio macchine e robot per la saldatura da impiegare nella produzione industriale, ed occupa al giorno d'oggi una posizione di rilievo in questo stesso settore.

Come si può facilmente comprendere, il miglior metodo per effettuare dei controlli su questi sistemi ad alta tecnologia, è proprio quello di raccogliere il più possibile informazioni sul loro operato per metterle in relazione al loro stato di funzionamento, con lo scopo di capire se ci sono problemi o possibilità di guasto. I dati raccolti per questo progetto, in particolare, sono collegati ad uno studio sul tensionamento delle cinghie di trasmissione dei bracci robotici: il comportamento di questi sistemi varia in relazione alla tensione applicata sulla cinghia che a sua volta dipende dal numero di rondelle applicate manualmente ai robot. L'analisi dei dati che verrà descritta in questo e nei prossimi capitoli verterà esattamente sulla ricerca di un modello che, adattandosi ai dati forniti, possa spiegare le differenze nei comportamenti delle macchine ed eventualmente segnalare la presenza di anomalie.

In questo capitolo di descrizione del caso di studio perciò si parlerà del problema generale che riguarda da vicino i robot in questione, ossia il *belt tensioning*, che verrà analizzato da un punto di vista più tecnico per poterlo comprendere al meglio. Dopodiché verranno descritti i dati forniti, verranno messi in relazione con lo studio

pratico del problema da affrontare, e infine verrà fornita un breve spiegazione sugli strumenti usati in questa tesi, essenzialmente il linguaggio di programmazione Python con le sue librerie e il database non relazionale MongoDB, per dare il via al progetto di *data analytics*.

4.1 – *BELT TENSIONING*

Una cinghia di trasmissione [25] [26] può essere generalmente definita come un sistema che collega due alberi rotanti (il più delle volte paralleli) tramite l'uso di pulegge, poste tra loro ad una certa distanza che può essere anche abbastanza grande. Le cinghie sono solitamente fatte di cuoio, fibre tessili o nylon e permettono la trasmissione di potenze elevate, garantendo l'assenza di forti rumori, l'assorbimento di urti e la protezione dei motori o dei supporti da improvvise variazioni di carico. Rispetto ai classici sistemi di trasmissione ad ingranaggi, le cinghie sono meno rigide e resistenti, tuttavia il fatto di avere maggiore elasticità le rende meno soggette a danni e per questo stesso motivo necessitano di una minore manutenzione. Al giorno d'oggi esistono diverse tipologie di cinghie, e le più conosciute sono la cinghia piatta, la cinghia trapezoidale, la cinghia tonda e quella dentata. Il loro ambito di utilizzo è molto vario, infatti possono trovare applicazione nelle automobili come cinghie di distribuzione o come cinghie dei servizi (vengono usate soprattutto cinghie dentate in questo caso) o anche in macchine utensili nella produzione, nell'elettronica e nei lettori ottici.

Come ogni sistema meccanico progettato per lavorare in maniera automatica e trasmettere grandi quantità di potenza, anche le cinghie di trasmissione possono essere soggette a malfunzionamenti, specialmente dopo lunghi periodi di utilizzo. I principali effetti dei guasti sono solitamente il disallineamento (ovvero uno spostamento laterale che avviene se la cinghia, o parte di essa, gira al di fuori delle pulegge sfregando contro altri componenti), l'usura della cinghia sui denti o sui fianchi, la presenza di crepe sul dorso o la rumorosità della cinghia stessa durante la sua attività. Molto spesso, alla base di tutti questi possibili malfunzionamenti, è presente un errore nel tensionamento [27] della cinghia, un problema molto serio che bisogna costantemente monitorare se si ha a che fare con questi sistemi di cinghie e pulegge. Purtroppo però, quando vengono allo

scoperto gli effetti di un cattivo tensionamento sul sistema, non si è più in tempo per correggere il problema perché è molto probabile che le varie componenti abbiano ormai subito un danno piuttosto grave. Perciò il miglior modo per gestire questa casistica è quello di monitorare in modo continuo lo stato di salute dei componenti e provare a effettuare azioni di manutenzione predittiva. Inoltre, è essenziale configurare alla perfezione la cinghia e le pulegge al momento dell'installazione, in modo da favorire un corretto svolgimento del lavoro e la massimizzazione della vita di esercizio del sistema. Mantenere il corretto stato di tensionamento, oltre che evitare la maggior parte dei danni e aumentare la vita media della cinghia, permette anche la trasmissione di una maggiore potenza e un notevole risparmio energetico.

Per controllare che la tensione applicata alla cinghia sia quella corretta esistono diverse tecniche ed è stato anche ideato uno strumento, chiamato appunto misuratore di tensione, che svolge in modo automatico l'operazione di controllo sulla cinghia. In ogni caso il problema del *belt tensioning* si verifica quando uno dei due rami della cinghia (detto "ramo condotto") risulta troppo meno teso rispetto all'altro ramo in tensione, e dunque la cinghia stessa non aderisce perfettamente alla puleggia durante il suo movimento. La tensione applicata alla cinghia deve tuttavia essere una via di mezzo: una tensione troppo bassa provoca slittamento, surriscaldamento e usura prematura della cinghia e della puleggia, mentre una tensione troppo alta porta a sforzi eccessivi su cinghie, cuscinetti e alberi. Generalmente viene considerata come tensione corretta la tensione minima a cui la cinghia non scivola né stride sotto l'applicazione del carico massimo. Chiaramente questa affermazione è molto generica e può facilmente essere fraintesa, ma serve a dare un'idea del significato del corretto tensionamento, che è un problema molto studiato in ingegneria meccanica e per il quale sono stati elaborati diversi modelli matematici, che calcolano la tensione corretta a partire da alcuni parametri dati, come la velocità di rotazione del motore, la potenza nominale del sistema, la distanza tra le pulegge, la sezione e la lunghezza delle cinghie.

Nel problema affrontato nel caso di studio relativo a questa tesi, come già accennato in precedenza, la tensione applicata sulla cinghia è dipendente dal numero di rondelle utilizzate: più questo numero cresce, meno la cinghia è tesa. Perciò è stato scelto il

numero di rondelle come informazione base per poter studiare il comportamento dei macchinari e provare a definire quale sia lo stato ideale di lavorazione.

4.2 – STRUTTURA DEI DATI

I dati da utilizzare per le analisi sulla manutenzione predittiva dei bracci robotici, sono stati forniti in formato JSON (acronimo di *Javascript Object Notation*), molto semplice da leggere e interpretare sia per le persone che per i principali linguaggi di programmazione. Il JSON è un formato di trasmissione di dati in realtà indipendente da tali linguaggi, che però utilizza alcune convenzioni tipiche della programmazione, nel senso che raggruppa i propri dati come insiemi di coppie chiave – valore (che ricordano i dizionari di Python o le mappe di Java, per esempio) e come elenchi ordinati, molto simili ai classici array. La sintassi dei file JSON pertanto segue uno schema fisso dove ad ogni chiave (che rappresenta uno specifico campo, o attributo) corrisponde un elemento (che può essere una stringa, un numero, un elenco o un valore booleano che contiene l'informazione di interesse). I dati relativi al progetto di questa tesi dunque sono stati salvati come JSON perché oltre alla facilità di utilizzo, questo formato permette di salvare informazioni non del tutto strutturate (ad esempio, gli elenchi non devono necessariamente avere tutti la stessa lunghezza) e può essere letto da database non relazionali.

Gli elementi contenuti nel file fornito, quindi, presentano la struttura riportata di seguito; tra virgolette ed in grassetto sono scritti i nomi degli attributi e per ognuno di essi è segnalata la tipologia di dati che contengono:

{

*"**id**":* STRING

*"**robotId**":* STRING

*"**StartTime**":* DATETIME

*"**CycleTime**":* FLOAT

*"**RobotProgram**":* INT

*"**sensorData**": { "**Current**": { "**UoM**":* STRING

```
                    "values": ARRAY [FLOAT] },
            "Position": { "UoM": STRING
                    "values": ARRAY [FLOAT] }
            },
"numWashers": INT
"currentFeatures": { "Max": FLOAT
                "Min": FLOAT
                "Mean": FLOAT
                "Peak2Peak": FLOAT
                "Var": FLOAT
                "Std": FLOAT
                "Rms": FLOAT
                "Kurtosis": FLOAT
                "Skewness": FLOAT
                "Rms_LP": FLOAT
                "Rms_HP": FLOAT
                "Rms_LHP": FLOAT
               }
}
```

Ogni elemento descritto in questo formato rappresenta un ciclo di lavoro di un robot: esso viene identificato univocamente dal campo "*id*", che però non fornisce indicazioni sul suo svolgimento, e dal campo "*StartTime*", che indica il momento preciso (data ed ora esatta) in cui il ciclo è iniziato. "*RobotId*" è una serie di caratteri che identificano la macchina in attività e "*CycleTime*" è un numero in secondi che contiene l'informazione sulla durata del ciclo; teoricamente, se non ci sono stati intoppi, ogni ciclo dovrebbe avere la stessa durata, ovvero circa 24 secondi. I dati veri e propri, quelli realmente utilizzati nel progetto, sono comunque i dati raccolti dai sensori, inseriti nel campo "*sensorData*" e divisi tra "*Current*" e "*Position*". I valori di questi due campi sono due elenchi che indicano rispettivamente il segnale di corrente (in Ampere) emesso durante il ciclo e la corrispondente posizione della macchina, e ognuno dei due elenchi (che logicamente hanno la stessa lunghezza) contiene poco meno di 12.000 elementi, il che

significa che le due informazioni di interesse sono state monitorate e registrate dai sensori ogni due millisecondi. Il campo "*numWashers*" invece rappresenta il numero di rondelle applicate alla macchina durante un preciso ciclo. Ogni ciclo può avere un diverso numero di rondelle e nel dataset utilizzato in questo progetto sono compresi valori di "*numWashers*" da 0 a 4. Questo campo è stato determinante nelle analisi effettuate poiché indica l'etichetta dei dati, ossia il valore da predire nei modelli di classificazione generati, che dovrebbe svolgere la funzione di discriminante nel capire se e quando il robot si comporta diversamente dal solito e se sta offrendo performance migliori o peggiori. I campi restanti sono delle metriche che descrivono la distribuzione statistica dei valori della corrente per ogni ciclo: tramite queste ed altre features è stato possibile analizzare dettagliatamente il comportamento delle macchine durante la loro attività.

Nella **tabella 4.1** sono sintetizzate alcune informazioni sui cicli presenti nel dataset. In particolare è importante notare quali giornate di lavoro sono state monitorate, quanti cicli sono stati effettuati e quali numeri di rondelle sono stati usati per ogni data. Ad esempio durante l'11 ottobre 2018 sono stati registrati 1298 cicli, da mezzanotte alle 18 circa, e sono state usate prima 4 e poi 3 rondelle. Come si può facilmente notare, molto spesso le macchine hanno lavorato ininterrottamente per più giornate di fila mentre i sensori continuavano a raccogliere dati.

Data	Ora primo ciclo	Ora ultimo ciclo	Numero di cicli	Rondelle
10/10/2018	16.57.02	23.59.49	531	4
11/10/2018	00.00.37	17.48.26	1298	4, 3
12/10/2018	09.02.58	23.57.11	1122	1
13/10/2018	00.00.23	23.59.26	1804	1
14/10/2018	00.01.02	23.58.30	1803	1
15/10/2018	00.00.05	23.59.23	1764	1, 2
16/10/2018	00.00.59	23.58.25	1731	2, 0
17/10/2018	00.00.48	23.59.54	1730	0, 3
18/10/2018	00.00.42	08.54.22	670	3
22/10/2018	12.55.32	23.59.32	820	3
23/10/2018	00.00.20	23.59.24	1433	3
24/10/2018	00.00.11	23.59.15	1805	3
25/10/2018	00.00.03	23.59.24	1762	3

26/10/2018	00.00.12	23.59.43	1739	3, 2
27/10/2018	00.00.31	11.30.31	868	2
29/10/2018	08.32.57	23.59.29	1150	2
30/10/2018	00.00.16	23.59.26	1803	2

Tabella 4.1: Informazioni sui cicli presenti nel dataset

Nella **tabella 4.2** invece viene riportato il numero totale di cicli per ogni configurazione diversa di rondelle, indipendentemente dal giorno in cui sono stati effettuati: si può notare che nella maggior parte dei casi si tratta di cicli con 1, 2 o 3 rondelle, mentre i casi più "esterni", ovvero 0 e 4 rondelle, compaiono molte meno volte (in tutto meno del 15% del dataset totale).

Rondelle	Numero di cicli	% sul dataset totale
0	2392	10.04
1	5367	22.52
2	6212	26.06
3	8707	36.53
4	1155	4.85
totale:	**23833**	**100**

Tabella 4.2: Numero di cicli per ogni configurazione di rondelle

Prima di passare alle analisi vere e proprie, è stato dunque svolto un lavoro di caratterizzazione del dataset per capire come fossero strutturati i dati forniti. Dopo aver analizzato la distribuzione dei cicli e del numero di rondelle durante i giorni presi come campione, si è cercato di capire qualcosa in più sul comportamento del segnale di corrente e su come poterlo sfruttare per elaborare il modello di manutenzione predittiva. In **figura 4.1** viene mostrato l'andamento dei valori registrati per la corrente rispetto al numero di osservazioni, durante il primo ciclo del giorno 10 ottobre 2018. Si può notare che sono presenti alcuni picchi e valori sia negativi che positivi.

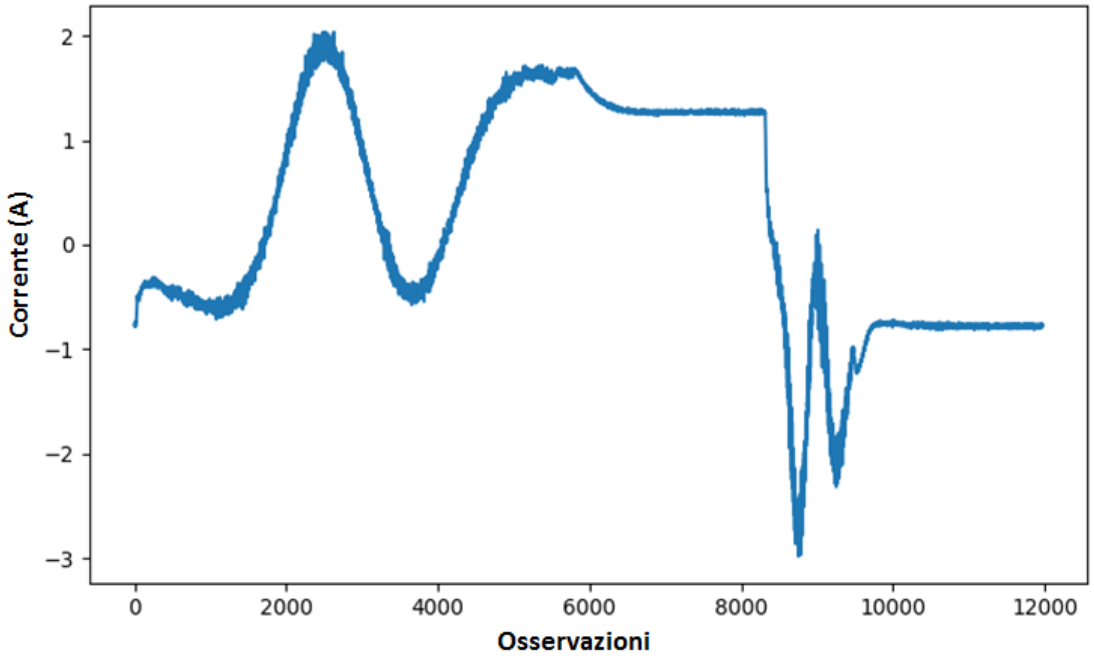

Figura 4.1: Andamento della corrente durante il primo ciclo (ore 16.57.02) del 10/10/18

Invece nella **figura 4.2** i valori della corrente sono stati messi in relazione a quelli della posizione della macchina, andando in questo modo a descrivere il suo ciclo di lavoro. L'andamento della corrente rispetta all'incirca questo stesso andamento in tutti i cicli analizzati, ma presenta alcuni scostamenti di valore, molto piccoli in valore assoluto, che però possono essere significativi nell'indicare un diverso comportamento della macchina. Pertanto, le prime analisi condotte sono state indirizzate nel cercare delle possibili differenze tra i vari cicli, provando soprattutto a capire quando si siano verificati scostamenti tra i valori e a cosa tali scostamenti possano essere dovuti, mettendo poi i risultati raggiunti in relazione alle rondelle in uso.

Da questi primi due grafici è possibile dedurre che, nel ciclo di lavoro dei macchinari, ci sono alcuni momenti in cui la variazione dei valori di corrente è molto più pronunciata, che pertanto potrebbero essere più significativi nella ricerca di differenze tra i vari cicli e tra i diversi giorni, ma per paragonare in maniera più efficace un numero più grande di cicli e visualizzare graficamente i risultati è stato necessario utilizzare le informazioni sulla media della corrente ed effettuare un campionamento sui cicli.

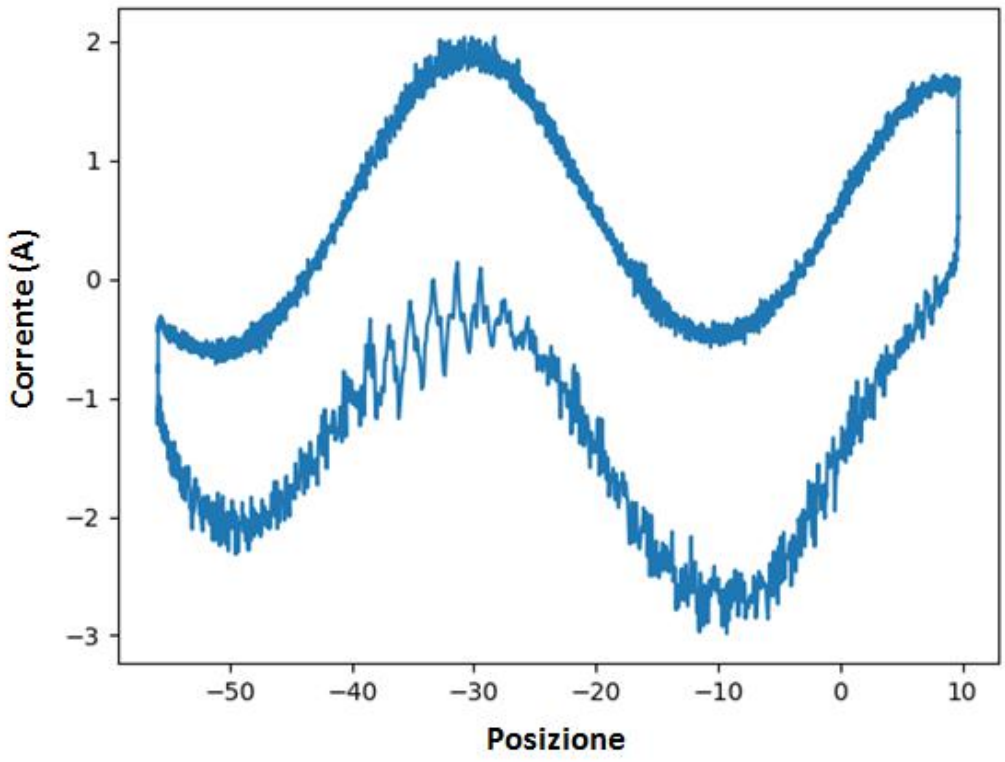

Figura 4.2: Andamento della corrente durante il primo ciclo (ore 16.57.02) del 10/10/2018 rispetto alla posizione

Nella **figura 4.3** viene mostrato pertanto l'andamento della media della corrente nei giorni a disposizione. Ovviamente non sono presenti informazioni su tutti i cicli (che, come visto in precedenza, sono oltre 20.000) ma ne sono stati selezionati alcuni per ogni data.

Oltre a mostrare una serie di punti ordinati per data e ora, questo grafico mette in relazione i valori della corrente con il numero di rondelle utilizzate in ogni ciclo (i punti sono infatti colorati secondo "*NumWashers*"). Analizzando attentamente la **figura 4.3** dunque è possibile trarre le prime importanti conclusioni su questo caso di studio, che forniscono alcuni spunti decisivi per le analisi di cui si parlerà nel prossimo capitolo:

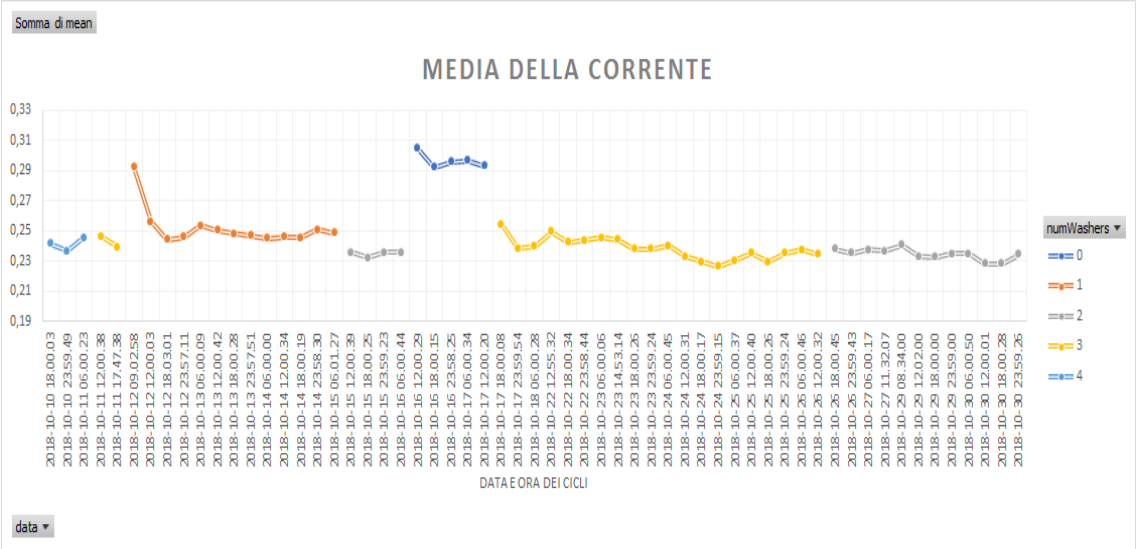

Figura 4.3: Andamento della media della corrente nei giorni studiati

- Molto spesso, all'inizio della giornata lavorativa o dopo che una macchina riprende a lavorare dopo alcune ore di pausa, si hanno maggiori assorbimenti di corrente. Questo comportamento si presuppone sia dovuto al fatto che la macchina debba ancora riscaldarsi e per cominciare a funzionare normalmente ci impieghi qualche minuto. C'è il rischio che andando ad analizzare una così grande quantità di dati, i valori della corrente registrati ad inizio giornata (o anche appena una rondella è stata cambiata, tolta o aggiunta) possano essere confusi con un malfunzionamento. È stato necessario tenere conto di questo fatto nel lavoro affrontato.

- I valori medi della corrente effettivamente cambiano in relazione al numero di rondelle applicate. Questa differenza si nota soprattutto tra il gruppo 0 e tutti gli altri gruppi, mentre i gruppi 2, 3 e 4 presentano valori abbastanza simili. Per quanto riguarda il gruppo 1, tolta la coda iniziale dovuta alla macchina ancora fredda, i valori sono leggermente più elevati rispetto ai gruppi 2, 3 e 4, e decisamente inferiori al gruppo 0. L'analisi del valore medio della corrente in questi cicli comunque non può bastare per trarre conclusioni precise sul

comportamento dei macchinari, e pertanto è stato necessario utilizzare molte altre features.

- Ci sono pochi dati relativi ai gruppi 0 e 4, il che potrebbe portare complicazioni o fraintendimenti sul loro comportamento. Nelle analisi successive, questi due gruppi sono stati considerati casistiche estreme, concentrandosi maggiormente sugli altri 3 gruppi. In ogni caso il gruppo 4 è molto simile come valori al gruppo 2 e al gruppo 3, mentre il gruppo 0 è un gruppo abbastanza coeso internamente, ma molto diverso dagli altri.

4.3 – STRUMENTI UTILIZZATI

Gli strumenti utilizzati per il progetto relativo a questa tesi sono essenzialmente alcuni programmi software adatti alla gestione e alla manipolazione dei big data. Il file JSON contenente tutte le informazioni sui cicli è stato aperto ed esplorato tramite MongoDB [28], un DBMS non relazionale orientato ai documenti in grado di leggere dati non strutturati e quindi classificato come database NoSQL. MongoDB è un software libero e opensource e presenta alcuni evidenti vantaggi nella gestione di grandi volumi di dati, come ad esempio la semplice e rapida integrazione del sistema con applicazioni esterne tramite delle apposite API, la possibilità di effettuare query ad hoc sui dati, permettendo di usare ricerche per campi ed espressioni regolari, o la velocità di queste ultime nel portare i propri risultati anche in presenza di svariati GB di dati. Nel progetto affrontato tuttavia, MongoDB è servito soltanto come grossa base di dati a cui si poteva comodamente accedere tramite il linguaggio di programmazione ad alto livello Python [29], grazie alla libreria PyMongo, che ha l'esatto scopo di connettere queste due parti. Una volta estratti da MongoDB, i dati sono dunque stati rielaborati con Python, e la maggior parte dei risultati intermedi sono stati salvati su file Excel o CSV, oppure in grafici riassuntivi delle analisi effettuate.

Dunque, il vero fulcro del progetto di *data analytics* è stato svolto con Python (e tramite Pycharm, un ambiente di sviluppo integrato utilizzato soprattutto per questo linguaggio) perché esso possiede diverse librerie ottimizzate per l'analisi di dati che si sono rivelate

fondamentali non solo per la formulazione dei modelli di predizione, ma anche proprio per la semplice gestione dei dati grazie ad adeguate strutture interne. Le principali librerie utilizzate sono riportate nel seguente elenco, insieme ad una sintetica descrizione delle operazioni che hanno permesso di svolgere:

- PyMongo [30]: come già sottolineato in precedenza, è stata usata per effettuare query sui dati contenuti in MongoDB tramite Python.

- Pandas [31]: è una libreria che permette la gestione di dati in forma tabellare (tramite i cosiddetti DataFrame, delle strutture dati molto simili alle tabelle di un classico database, che consentono, tra le altre cose, di indicizzare i dati e di manipolarli in maniera efficace) o sequenziale (tramite vettori mono-dimensionali anch'essi indicizzabili chiamati Series, meno utilizzati in questa tesi rispetto ai DataFrame). Tra le sue più importanti caratteristiche vi sono anche la capacità di effettuare operazioni numeriche, statistiche e di visualizzazione di risultati in modo molto rapido ed intuitivo. Oltre che per queste ragioni, Pandas è stata poi utilizzata per i suoi metodi di lettura e scrittura di file esterni in differenti formati, come ad esempio i CSV.

- Matplotlib [32] e Seaborn [33]: si tratta di librerie utili per la realizzazione di grafici. Durante questo lavoro è stato fatto ampio uso di queste librerie, in quanto la visualizzazione grafica dei risultati ottenuti è stata una componente fondamentale del progetto. Essa si è rivelata fondamentale in fase di esplorazione del dataset per indirizzare le analisi verso una certa direzione, e anche durante i test dei modelli di predizione per smentire o confermare ipotesi e scelte effettuate. Sia Matplotlib che Seaborn, in poche righe di codice, permettono di visualizzare grafici in 2D o in 3D: i più utilizzati in questo progetto sono stati gli "*scatterplot*", ossia i grafici a dispersione, per poter vedere ad esempio le differenze tra i punti di gruppi diversi.

- Scikit-learn [34]: è una libreria orientata all'analisi di dati e al *machine learning*, progettata apposta per essere usata insieme ad altre librerie come Numpy (altra libreria di Python che supporta grandi vettori e matrici multidimensionali e aggiunge diverse funzioni matematiche) o Pandas (molte funzionalità di questa

libreria sono infatti pensate per ricevere come input un DataFrame, per esempio). Contiene svariate classi e metodi per poter svolgere qualunque tipologia di analisi, grazie possibilità di implementare tutti i più importanti algoritmi di *data mining*. Nell'ambito di questa tesi è stata utilizzata soprattutto per la fase di preprocessing e per la costruzione, il train e il test dei vari classificatori progettati.

5 – REALIZZAZIONE DEI MODELLI

Come è già stato detto in precedenza, il progetto affrontato per questa tesi riguarda la costruzione di uno o più modelli di manutenzione predittiva tramite tecniche di *data analytics* e *machine learning*, applicati ad un caso concreto dove i dati forniti riguardano l'emissione di corrente da parte di alcuni macchinari durante il loro ciclo di lavoro, messa in relazione al tensionamento della cinghia di trasmissione (definita dal numero di rondelle presenti). In questo capitolo verranno spiegati dettagliatamente tutti i passaggi svolti nel progetto e tutte le analisi effettuate per giungere allo scopo finale.

Per ogni step verranno spiegati la metodologia utilizzata e le motivazioni di tutte le eventuali scelte prese, nonché i risultati ottenuti, anche se parziali o intermedi. Il lavoro svolto non è stato del tutto lineare: molto spesso infatti sono state percorse ed esplorate diverse soluzioni possibili contemporaneamente e solo infine me è stata scelta una, mentre altre volte alcune vie hanno portato ad un nulla di fatto e sono state abbandonate per tornare indietro e optare per altre possibilità. Comunque, l'architettura generale del progetto è sintetizzata nella **figura 5.1**, dove vengono mostrati i principali blocchi caratterizzanti il lavoro svolto, che verranno spiegati durante questo capitolo.

I blocchetti raffigurati in giallo, ossia "Calcolo Features" e "Selezione Features", fanno parte del preprocessing e riguardano la scelta e la gestione degli attributi (detti appunti "features") relativi ai valori della corrente registrati nei vari cicli. I blocchetti di colore blu invece rappresentano gli step per la creazione del modello predittivo generale che presenta, tra le sue funzionalità, l'analisi di serie storiche sia trascurando il vincolo temporale sui dati (considerando dunque solo l'etichetta "*numWashers*"), sia conservando tale informazione (ovvero il campo relativo a data e ora del ciclo), oltre che la ricerca di una quantità minima di dati per effettuare delle predizioni corrette. Infine, i blocchetti verdi fanno parte della generazione di un modello cosiddetto "evolutivo", per cui sono stati utilizzati solo i gruppi principali (quelli con "*numWashers*" pari a 1, 2 o 3) come dataset di train ed il cui scopo è stato quello di simulare una situazione di degrado della macchina evidenziando le differenze tra i 5

insiemi di dati anche tramite il calcolo di alcuni indici di coesione dei gruppi (come MSE e silhouette).

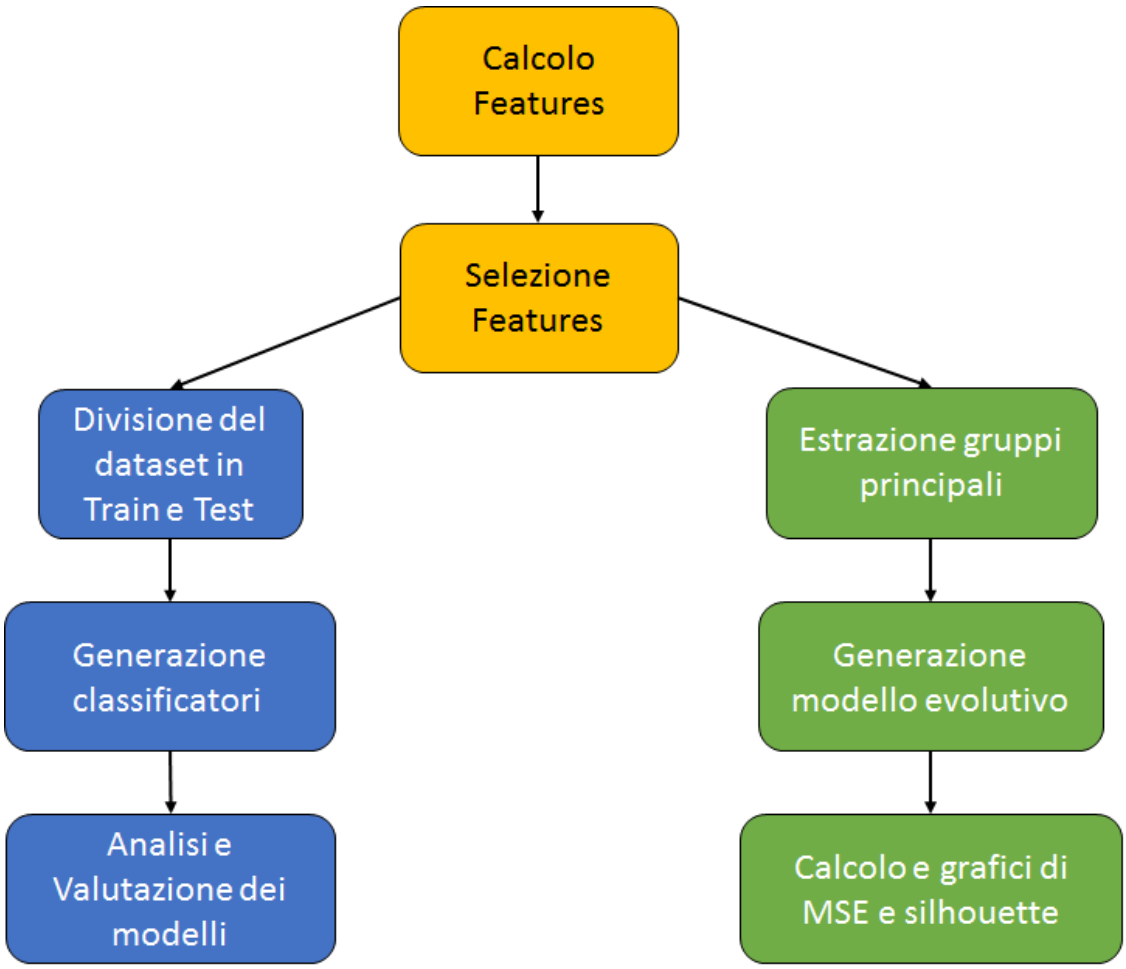

Figura 5.1: blocchi fondamentali del progetto svolto

5.1 – PREPROCESSING DEI DATI

La prima operazione svolta nell'ambito del preprocessing è stata quella di determinare quali features calcolare per ottenere una chiara e precisa descrizione dei cicli. Una volta estratti i dati da MongoDB, le features già presenti nel file JSON sono state ricalcolate velocemente grazie alla funzione `describe()` di Python, usata sui DataFrame nei quali venivano memorizzati tutti i valori del segnale di corrente. Dopodiché sono state

aggiunte altre features in modo che ogni ciclo venisse rappresentato non più dall'insieme dei circa 12.000 valori di corrente, ma da queste metriche di statistica descrittiva in grado di raggruppare l'informazione circa la distribuzione di tali dati. Le features calcolate sono elencate di seguito:

- Media;
- Deviazione standard e varianza;
- Massimo e minimo;
- Primo quartile, mediana, terzo quartile;
- Kurtosi (indice statistico che riflette la concentrazione dei dati attorno alla propria media: se > 0 essi sono molto vicini alla media e la curva della distribuzione apparirà allungata, se < 0 al contrario si avrà una curva appiattita dovuta a dati più lontani dalla media);
- Skewness (indice di simmetria di una distribuzione: se > 0 i dati si concentrano perlopiù a sinistra rispetto alla propria media, se < 0 invece essi sono più concentrati a destra, mentre se è pari a 0 i valori sono equamente distribuiti intorno alla propria media);
- RMS (*Root Mean Square*), ossia la radice quadrata della media dei quadrati dei valori del segnale, anche noto come "valore efficace";
- RMS HP: questa metrica consiste nel calcolo dell'RMS sui valori derivati dall'applicazione di un filtro passa alto;
- RMS BP: questa metrica consiste nel calcolo dell'RMS sui valori derivati dall'applicazione di un filtro passa banda;
- RMS LP: questa metrica consiste nel calcolo dell'RMS sui valori derivati dall'applicazione di un filtro passa basso;

Dopo aver stabilito quali metriche calcolare per ogni ciclo, confrontando gli andamenti della corrente emessa di più cicli diversi anche non consecutivi, è stato notato che, nell'arco dei 24 secondi di lavoro, ci sono alcuni istanti temporali in cui è possibile notare una maggiore differenza tra i valori registrati. Questa osservazione ha portato a pensare che probabilmente esistono delle fasi del ciclo che impattano maggiormente sulle buone o cattive prestazioni della macchina e che pertanto meritavano uno studio più approfondito. Per questa ragione è stato scelto di analizzare i cicli secondo per secondo, dividendoli quindi in 24 segmenti in ordine temporale, ognuno contenente circa 495 valori di corrente: dopo aver diviso ogni ciclo presente nel dataset secondo questa logica, è stata utilizzata la funzione di Python per calcolare tutte le statistiche sopra elencate su ognuno dei segmenti dei cicli. Così facendo, per ogni ciclo di lavoro sono state calcolate altre 336 features (ossia 14x24, un valore piuttosto alto, specie considerando il già elevato numero di cicli nel dataset), oltre alle 14 già presenti. Infine, per concludere questa parte iniziale di preprocessing, i risultati sono stati salvati in un file csv chiamato "*general matrix*", dove affianco alle 350 features, per ogni ciclo sono state riportate anche le informazioni riguardanti il numero di rondelle e il timestamp.

L'operazione successiva è stata la selezione delle features, un'analisi di controllo per decidere quali tra i 350 attributi generati fossero davvero interessanti ai fini del progetto. Infatti, in parte per alleggerire il file da processare, in parte per eliminare quelle features che contenevano informazioni ridondanti e quindi inutili, è stato necessario analizzare attentamente la correlazione presente tra esse. Lo strumento statistico tramite cui si calcola questo dato è la matrice di correlazione, dove per ogni coppia di features viene indicato un valore, detto appunto coefficiente di correlazione, compreso tra -1 e 1. Se tale coefficiente, calcolato per due variabili, è circa pari a 1 allora esse sono da considerare correlate positivamente; al contrario, se assume valori prossimi a -1, le due variabili sono correlate negativamente. Infine, per valori del coefficienti attorno allo 0, due variabili possono essere considerate non correlate, il che significa che l'informazione da esse contenuta non è ridondante e quindi aggiungono valore all'insieme delle features. La matrice di correlazione è una matrice simmetrica i cui elementi sulla diagonale principale valgono sempre 1 (infatti ogni variabile è correlata a sé stessa).

Nel caso in esame, tale matrice ha dimensioni 350x350, siccome non sono stati considerati l'etichetta "*numWashers*" e l'informazione sull'orario dei cicli. In **figura 5.2** è mostrata la parte della matrice di correlazione relativa alle features di base, quelle calcolate per i cicli interi. Come si può vedere, ci sono alcune features molto correlate tra loro, per esempio media e deviazione standard, che presentano una forte correlazione positiva, o mediana e skewness con un'evidente correlazione negativa.

Correlazione	mean	std	var	min	max	1 quartile	mediana	3 quartile	kurtosis	skew	RMS	RMS_LP	RMS_HP	RMS_BP
mean	1	0.949	0.951	-0.218	0.486	0.574	0.912	0.747	0.640	-0.841	0.968	-0.388	0.968	-0.389
std	0.949	1	0.999	-0.265	0.552	0.315	0.819	0.856	0.472	-0.746	0.998	-0.212	0.998	-0.212
var	0.951	0.999	1	-0.264	0.555	0.325	0.822	0.853	0.481	-0.746	0.998	-0.211	0.998	-0.212
min	-0.218	-0.265	-0.264	1	-0.548	-0.152	-0.189	-0.139	-0.284	0.257	-0.257	0.208	-0.256	0.208
max	0.486	0.552	0.555	-0.548	1	0.088	0.277	0.556	0.166	-0.252	0.543	-0.213	0.541	-0.212
1 quartile	0.574	0.315	0.325	-0.152	0.088	1	0.704	-0.023	0.865	-0.716	0.371	-0.615	0.374	-0.616
mediana	0.912	0.819	0.822	-0.189	0.277	0.704	1	0.451	0.849	-0.967	0.845	-0.415	0.847	-0.415
3 quartile	0.747	0.856	0.853	-0.139	0.556	-0.023	0.451	1	-0.005	-0.331	0.840	-0.027	0.839	-0.027
kurtosis	0.640	0.472	0.481	-0.284	0.166	0.865	0.849	-0.005	1	-0.896	0.511	-0.507	0.514	-0.508
skew	-0.841	-0.746	-0.746	0.257	-0.252	-0.716	-0.967	-0.331	-0.896	1	-0.771	0.443	-0.774	0.443
RMS	0.968	0.998	0.998	-0.257	0.543	0.371	0.845	0.840	0.511	-0.771	1	-0.249	1	-0.249
RMS_LP	-0.388	-0.212	-0.211	0.208	-0.213	-0.615	-0.415	-0.027	-0.507	0.443	-0.249	1	-0.250	1
RMS_HP	0.968	0.998	0.998	-0.256	0.541	0.374	0.847	0.839	0.514	-0.774	1	-0.250	1	-0.251
RMS_BP	-0.389	-0.212	-0.212	0.208	-0.212	-0.616	-0.415	-0.027	-0.508	0.443	-0.249	1	-0.251	1

Figura 5.2: Matrice di correlazione relativa alle features di base

Tuttavia la metodologia seguita non è stata quella di andare a controllare per ogni coppia di features quali presentano una certa correlazione e poi eliminarne una delle due dall'elenco, bensì, partendo dalla matrice, si è scelto di calcolare la correlazione media di tutti gli attributi per poi scartare quelli associati ad un valore superiore ad una determinata soglia decisa in partenza.

Il calcolo della correlazione media è stato effettuato considerando tutti i valori di correlazione per ogni feature presi in valore assoluto (infatti, ai fini di questo progetto, non ha importanza se la correlazione presente è negativa o positiva), escludendo per ovvi motivi la diagonale principale. I risultati di questo calcolo sono raffigurati in

figura 5.3, un istogramma in cui si può osservare la distribuzione della correlazione media delle features.

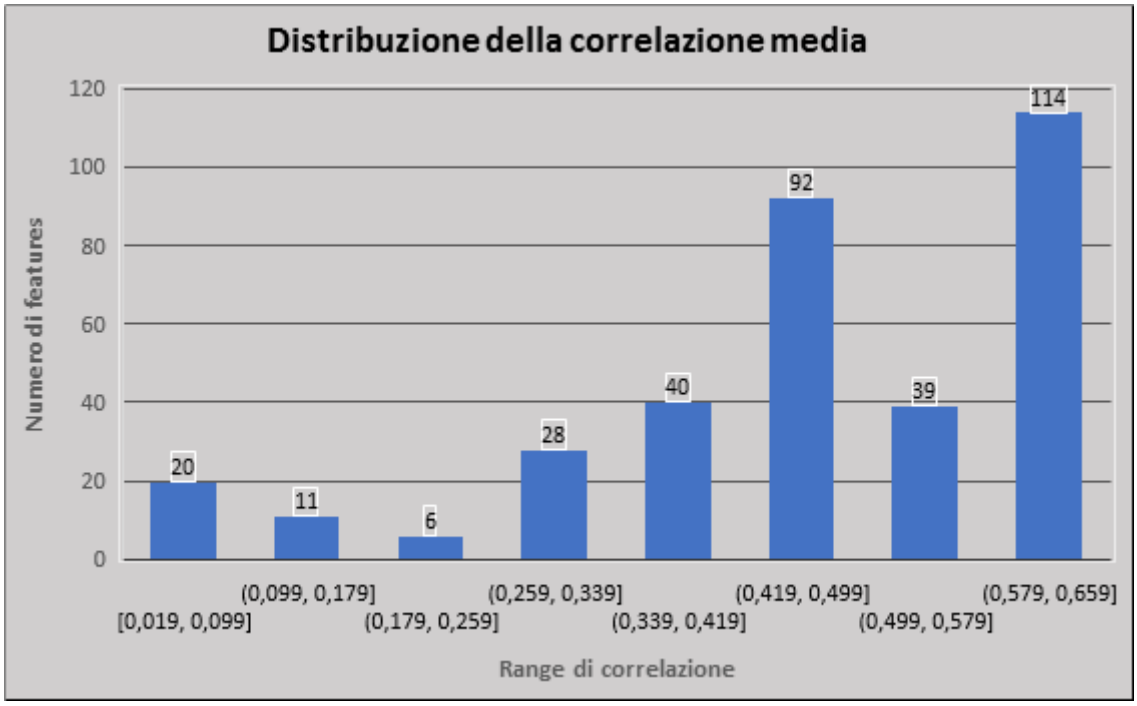

Figura 5.3: Distribuzione della correlazione media

I valori assunti dalla correlazione sono stati raggruppati in 8 classi di uguale ampiezza, che contengono però un numero decisamente diverso di elementi al proprio interno. Si può facilmente notare che la maggior parte delle features presentano una correlazione media abbastanza alta ma che nessuna assume valori oltre 0,659, mentre sono una minoranza quelle mediamente poco correlate.

Questa particolare distribuzione della correlazione media ha pertanto anche influenzato la scelta del valore soglia da utilizzare per decidere quali features tenere per le analisi vere e proprie. Infatti, inizialmente era stato pensato di adottare come numero soglia 0,8, valore che solitamente distingue una correlazione media da una correlazione alta. Però in questo caso non è stato possibile scegliere questa cifra in quanto non sono presenti feature così tanto correlate alle altre. Inoltre, trattandosi di correlazione media e non di correlazione semplice tra due variabili, è raro ottenere valori così alti. Per questo

motivo sono stati adottati altri due numeri soglia per scartare almeno una parte dei 350 attributi: essi sono 0,3 e 0,5.

È stato osservato che, utilizzando come soglia 0,3 il numero di features cala nettamente (soltanto 48 features infatti presentano una correlazione media inferiore a tale valore), mentre con 0,5 si ottiene un risultato intermedio, mantenendo ben 198 attributi per i dati.

Prima di elaborare i modelli di classificazione, è stato effettuato un ulteriore studio per decidere quale di questi tre dataset (con 48, 198 o tutte le features) utilizzare effettivamente tramite l'analisi della silhouette, una metrica che valuta la coesione interna dei cluster (i cluster in questo caso sono già definiti in partenza e sono rappresentati ancora una volta dal numero di rondelle). In particolare, la silhouette indica quanto un oggetto è simile agli altri contenuti nello stesso gruppo in base anche alla distanza da gruppi diversi. Questo indice può avere valori compresi tra 1 (che si ha in caso di un dato perfettamente assegnato al proprio gruppo) e -1 (caso in cui probabilmente il dato in questione appartiene a un gruppo diverso); nel caso di valori intorno allo 0, probabilmente alcuni cluster sono sovrapposti tra loro. La silhouette è stata calcolata tramite una funzione apposita di Scikit-learn per ogni punto dei dataset, utilizzando come metrica la distanza euclidea. Nella **tabella 5.1** vengono evidenziati i valori medi della silhouette di ogni gruppo per ognuna delle diverse configurazioni di features, mentre in **figura 5.4** si hanno delle rappresentazioni grafiche di questo stesso indice (viene mostrato il suo andamento dopo aver effettuato un campionamento di 1000 elementi per ogni gruppo).

Silhouette media	48 features	198 features	350 features
Gruppo 0	0.143	0.574	0.663
Gruppo 1	0.07	0.379	0.445
Gruppo 2	0.067	0.195	0.213
Gruppo 3	-0.005	0.003	-0.006
Gruppo 4	0.059	0.153	0.208

Tabella 5.1: Silhouette media dei gruppi in base alle features tenute nel dataset

Si può notare che mediamente i valori della silhouette crescono con l'aumentare del numero di features che vengono lasciate nel dataset. Il gruppo 0 è il più coeso (presenta sempre valori maggiori rispetto agli altri gruppi), mentre il gruppo 3 è quello che sembrerebbe essere più sparso (ed ha persino dei valori medi negativi).

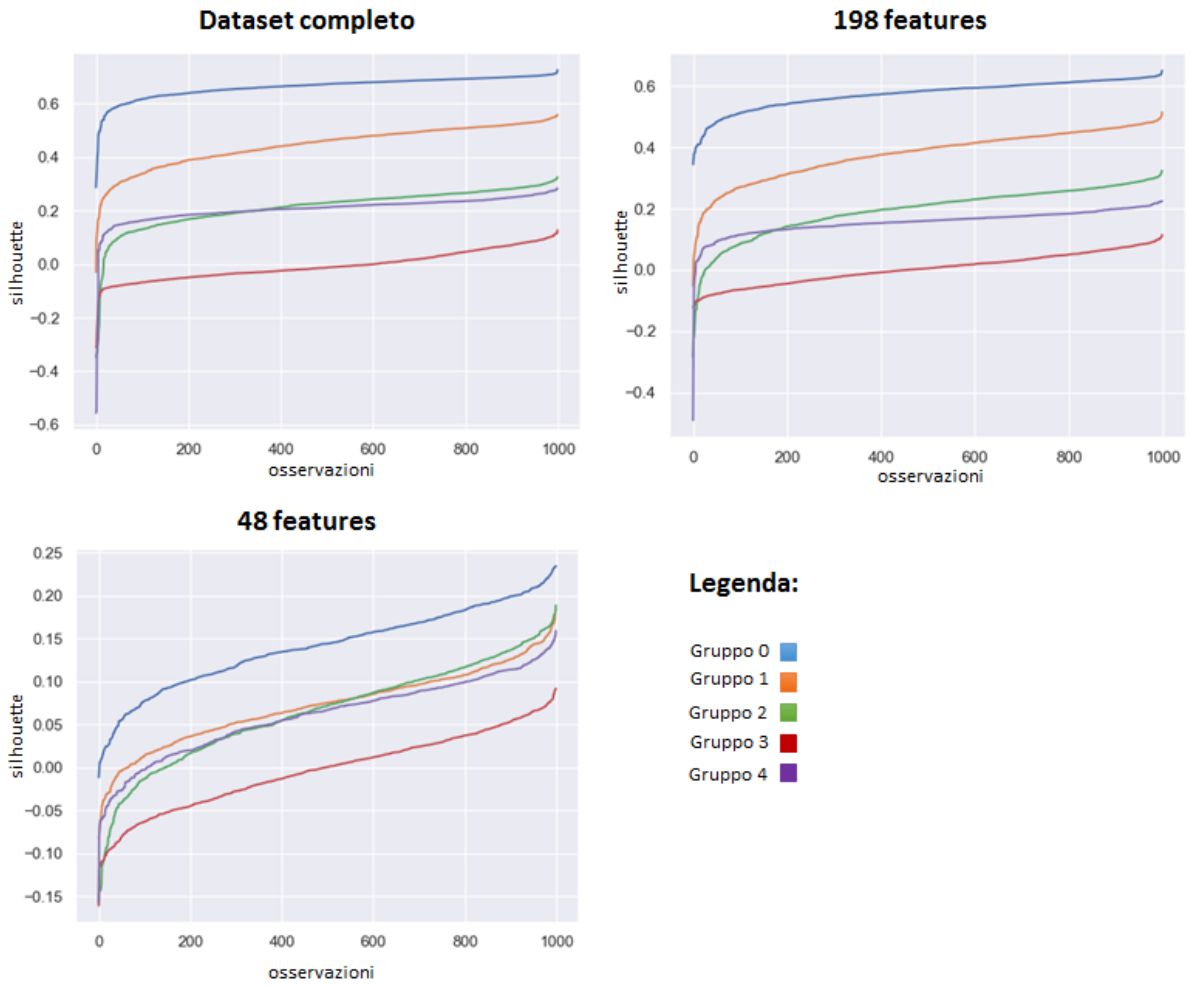

Figura 5.4: Andamento della silhouette per i dataset dopo il preprocessing

Dopo aver analizzato l'andamento della silhouette per i vari gruppi ed i suoi valori medi si è giunti alla conclusione di utilizzare il dataset intermedio (con le features con correlazione media minore di 0,5) per portare avanti tutte le analisi. Il dataset completo con 350 features è stato scartato perché troppo grande e perché la coesione dei gruppi non mostrava significativi miglioramenti pur mantenendo un numero così elevato di

features, mentre il dataset più piccolo (features con correlazione media minore di 0,3) non è stato utilizzato per i suoi valori di silhouette eccessivamente bassi.

L'ultimo step della fase di preprocessing è stato quello di applicare una tecnica di normalizzazione ai valori ottenuti dopo la *features selection*. La normalizzazione è molto utile per far sì che tutte le features abbiano valori contenuti nello stesso ordine di grandezza, ed è fondamentale nei casi in cui sia necessario effettuare dei calcoli di distanza tra i dati. Nelle analisi che verranno descritte in seguito, non sempre i dati hanno subito un processo di normalizzazione tuttavia: anche in questo caso sono state svolte diverse prove e in seguito sono stati scelti i risultati migliori. Chiaramente essa è stata effettuata prima del calcolo della silhouette, che, basandosi su un calcolo di distanza tra punti, richiedeva necessariamente quest'operazione. La formula scelta per la normalizzazione è la stessa che viene ampiamente utilizzata nella standardizzazione statistica, che riconduce le variabili ad una distribuzione con media pari a 0 e deviazione standard pari a 1:

$$Z = \frac{X - \mu}{\sigma}$$

Nel caso in esame, Z rappresenta i valori delle features dopo la normalizzazione, X i valori prima di quest'ultima, mentre μ e σ sono rispettivamente la media e la deviazione standard della distribuzione di una feature.

Oltre all'analisi della matrice di correlazione e della correlazione media delle features, è stato scelto di implementare parallelamente un'altra tecnica di preprocessing, basata anch'essa sulla riduzione della dimensione del dataset, chiamata PCA (sigla di "*Principal Component Analysis*", che in italiano significa semplicemente "analisi delle componenti principali"). Questa tecnica consiste dunque nel trasformare il dataset di partenza dando vita a delle nuove variabili (minori in numero rispetto a quelle originali), limitando il più possibile la perdita di informazioni. La PCA può essere fondamentale nei casi in cui siano presenti una moltitudine di features di cui alcune inutili per alleggerire in dataset e velocizzare le analisi, ed è anche utilizzabile per visualizzare graficamente i dati sfruttando un numero piccolo di attributi. Questo tipo di analisi è stato effettuato in Python tramite una funzione predisposta appositamente da Scikit-

learn, che si adatta molto bene ai DataFrame di Pandas e in cui è sufficiente specificare il numero di nuovi attributi da creare in sostituzione di quelli precedenti. I valori scelti sono stati 3, 10 e 20 e tale funzione è stata applicata direttamente sul dataset intero, ossia senza prima aver eliminato alcuna feature. I risultati ottenuti sono stati utilizzati anch'essi per la creazione dei modelli e in seguito paragonati ai risultati ottenuti con il metodo della correlazione media.

In **figura 5.5** viene mostrato un grafico in 3D dove ognuno degli assi rappresenta una delle componenti della PCA con parametro pari a 3 (pertanto, le 350 features sono state condensate in 3 attributi, che nel grafico sono denominati PC1, PC2 e PC3). Il principale vantaggio di una tecnica come questa è la possibilità di vedere graficamente quali relazioni intercorrono tra i dati, ma sintetizzando così tante informazioni in 3 attributi, essa non permette a chi osserva il grafico di capire realmente cosa significhino tali componenti.

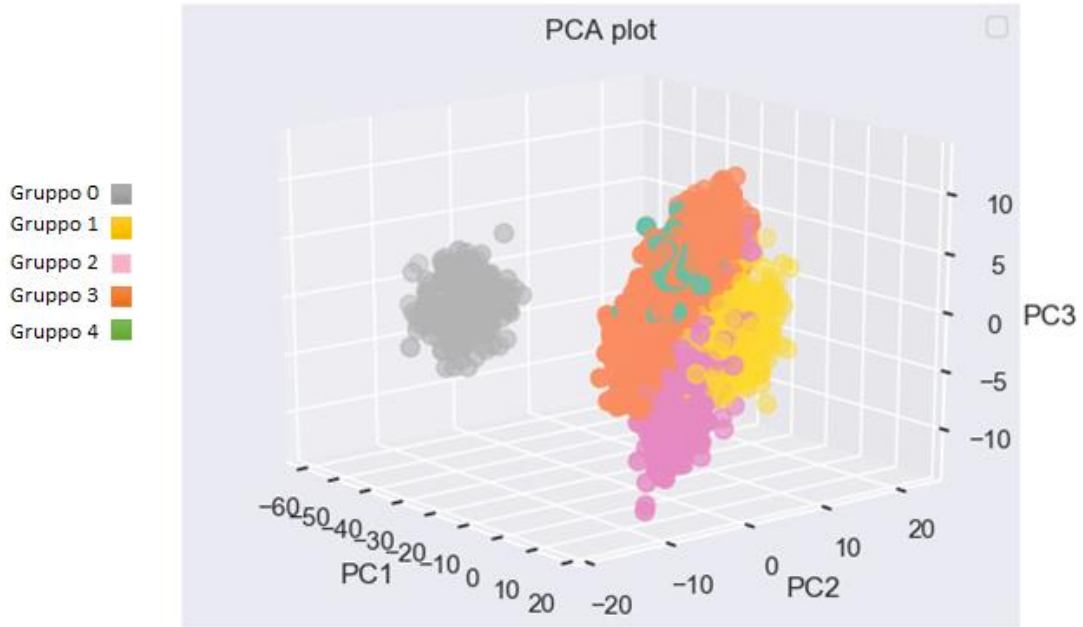

Figura 5.5: Grafico 3D delle componenti della PCA con 3 attributi

Da questo grafico si ha subito la conferma di ciò che già in fase di caratterizzazione del dataset era stato dedotto: i dati con 0 rondelle hanno un comportamento nettamente diverso da tutti gli altri. Per quanto riguarda gli altri gruppi, si può notare che i cluster sono comunque ben visibili e distinti, anche se in certi tratti sono un po' mischiati tra loro. Soprattutto il gruppo 4 pare quasi "immerso" nel gruppo 3, mentre il gruppo 2 sembrerebbe essere poco coeso internamente (infatti compaiono elementi rosa anche vicino al gruppo 3 e al gruppo 1).

5.2 – MODELLO GENERALE

Dopo aver studiato e rielaborato il dataset durante la fase di preprocessing, si è passati alla definizione e alla costruzione dei modelli da usare per la manutenzione predittiva dei robot. La natura stessa dei dati ha fatto sì che, tra le tecniche di analisi di dati esistenti, l'unica che avesse davvero senso utilizzare fosse la classificazione. Infatti, trattandosi di dati già etichettati (si ricorda che la *label* che contraddistingue i gruppi interni al dataset è "*numWashers*") non sarebbe stato possibile effettuare operazioni, per esempio, di clustering, né eseguire una regressione sui valori assunti delle features in quanto il segnale di corrente è sempre registrato dai sensori (e quindi vale in ogni caso come input del problema). Pertanto, l'obiettivo dei classificatori progettati, è quello di utilizzare il numero di rondelle come etichetta che identifica gruppi ben separati di informazioni per poter predire tale valore su nuovi dati. I nuovi dati sui quali verrà effettivamente utilizzato il modello, in realtà, non è detto che non abbiano l'informazione sulle rondelle, ma il vero scopo dei classificatori in questo caso è quello di dire se il macchinario sta funzionando esattamente come dovrebbe (caso in cui l'etichetta predetta corrisponde a quella effettiva) o se sta assumendo un comportamento errato (ovvero quando la predizione si discosta dalla realtà). In quest'ultimo caso sarà necessario intervenire variando il numero di rondelle e cercando di capire quale difetto stia provocando il funzionamento anomalo.

Durante le diverse prove effettuate, è stato definito come modello generale un insieme di classificatori che, preso come input un train set contenente dati su cicli con tutte le configurazioni possibili di "*numWashers*" (ovvero da 0 a 4 per il dataset utilizzato),

effettuano predizioni su altri dati non usati per addestrare il modello. La bontà dei modelli è stata valutata confrontando i risultati delle predizioni con i dati appartenenti ad un dataset di test, di cui è chiaramente nota l'etichetta a priori. Tra gli obiettivi finali del modello generale tuttavia, non vi era solamente quello di creare un modello di predizione del funzionamento dei robot, ma anche quello di cercare una quantità minima di dati da utilizzare per creare tale modello. Infatti, nel caso reale, non è noto a priori quanti dati si possano avere a disposizione né se i dati di train contengano tutte le casistiche possibili (anzi, è considerato generalmente molto raro che un modello conosca a priori tutti i diversi comportamenti dei macchinari: un buon modello si riconosce anche dal fatto di saper gestire casistiche mai affrontate in precedenza).

5.2.1 – MODELLI CON IL RANDOM FOREST

Tra gli algoritmi di classificazione esistenti (di cui è stata fornita una descrizione nel capitolo terzo di questa tesi) è stato scelto di utilizzare il Random Forest per elaborare il modello generale. Il motivo di questa decisione è dato dal fatto che è stato necessario e costruttivo per le analisi sfruttare un modello interpretabile, ad esempio per capire quali fossero effettivamente le features più decisive nella separazione dei gruppi, ma senza dover rinunciare ad un buon livello di accuratezza, infatti alcuni gruppi spesso tendono ad essere troppo "vicini" tra loro, e un buon modello deve essere in grado di cogliere il più possibile le differenze esistenti tra essi. Il Random Forest, trattandosi di un algoritmo che analizza i risultati provenienti dalla creazione di diversi alberi di decisione, si concilia perfettamente con le necessità di questo progetto: è un modello facilmente interpretabile, infatti l'output finale è leggibile grazie alla sua struttura ad albero, ed è anche piuttosto accurato siccome il suo risultato finale è ottenuto scegliendo i migliori risultati tra i vari alberi generati.

Per l'implementazione di questo algoritmo su Python, la libreria Scikit-learn mette a disposizione una classe apposita, chiamata "RandomForestClassifier". Tuttavia, non è sufficiente creare un oggetto di tale classe e chiamare un metodo per adattare il classificatore al dataset di train, poiché esso ha bisogno di alcuni parametri, precedentemente scelti e calcolati, per poter performare al meglio. La scelta dei

parametri del Random Forest è un'operazione piuttosto importante ai fini della costruzione e del successivo utilizzo del classificatore, e per questa ragione sempre Scikit-learn ha ideato un'altra classe, chiamata "GridSearchCV", con lo scopo di facilitare la ricerca dei parametri migliori da inserire nella programmazione del Random Forest.

Pertanto, i modelli con il Random Forest sono stati sviluppati seguendo questi passi:

1. È stata scritta una funzione per validare i parametri del Random Forest tramite un oggetto della classe "GridSearchCV". Questa funzione esamina diverse possibilità e restituisce come output i migliori parametri tra quelli indicati.

2. È stato in seguito creato un classificatore Random Forest che prende come input, oltre che il dataset di train, anche i parametri ottenuti precedentemente.

3. Tramite il classificatore creato, sono state calcolate le predizioni sul dataset di test, l'accuratezza, la precisione e il richiamo del modello (e tutte queste informazioni sono state visualizzate tramite una matrice di confusione).

Adesso verranno spiegati un po' più nel dettaglio i primi due punti, mentre il terzo verrà approfondito in seguito. Il codice tramite cui è stato creato l'oggetto "GridSearchCV" è il seguente:

```
gscv = GridSearchCV (rfc, param_grid = param_grid, cv=10, scoring =
{"accuracy": "accuracy", "f1": "f1_micro", "precision":
"precision_micro", "recall": "recall_micro"},
refit="f1", return_train_score = True, iid =False)
```

come si può notare, anch'esso richiede il settaggio di alcuni parametri: "rfc" è il tipo di classificatore che si deve validare, in questo caso viene passato un oggetto di classe "RandomForestClassifier"; "scoring" è un elenco di metriche che permette di valutare la bontà dei modelli creati; "cv" serve invece per determinare la strategia di divisione del dataset con lo scopo di effettuare una "*Cross-Validation*": in questo caso il parametro è settato pari a 10, che significa che il dataset sarà diviso in 10 sezioni (tramite un campionamento stratificato) di cui a turno ognuna svolgerà il ruolo di test set mentre le altre faranno le veci del train set. L'attributo "param_grid" invece è un dizionario dove

ad ogni chiave, che è il nome di un parametro specifico del classificatore, corrisponde una lista di valori che essa può assumere. Tale attributo permette la ricerca dei migliori valori possibili tra quelli indicati, creando uno alla volta diversi Random Forest e testando tutte le combinazioni possibili. Alla fine viene scelto il modello che massimizza il calcolo delle metriche contenute nel campo "scoring".

Nel caso applicativo in esame, sono stati inseriti nel dizionario "param_grid" tutti i più importanti parametri caratteristici del Random Forest insieme a una lista di possibili valori da testare. Essi sono elencati e descritti brevemente di seguito:

- Criterio di divisione dei nodi: è il metodo per misurare la qualità della divisione dei nodi degli alberi di decisione tramite cui viene costruito il Random Forest. I possibili criteri supportati in questo caso sono l'indice di impurità di Gini e l'*information gain*, che misura la diminuzione di entropia dopo che il dataset viene diviso in base a un attributo.

- Massima profondità degli alberi: è un valore intero che sostanzialmente indica il massimo numero di test sugli attributi che possono essere effettuati prima di giungere alla conclusione dell'albero (ossia al nodo che indica a quale classe sono assegnati i dati).

- Numero di stimatori: è il numero (intero) di alberi utilizzati per creare la "foresta". Chiaramente più alberi vengono creati, più sono alte le probabilità di ottenere un classificatore performante, ma d'altro canto un numero troppo elevato comprometterebbe la fattibilità computazionale di questa operazione.

- Numero massimo di features usate per cercare la miglior divisione possibile sui nodi degli alberi: può essere indicato in questo caso un numero intero o uno decimale, oltre che i valori standard "sqrt" e "log2". In questi ultimi due casi il numero massimo di features usate sarà la radice quadrata delle features totali del dataset oppure il logaritmo in base due di tale valore.

Dopo diverse prove effettuate variando le dimensioni del dataset di train e, chiaramente, i valori da testare per ogni diverso parametro, è stato notato che il miglior metodo in assoluto per il criterio di divisione dei nodi risultava essere sempre l'*information gain*,

mentre per il numero massimo di features la tecnica più efficiente era quella di usare la radice quadrata piuttosto che il logaritmo. Per quanto riguarda la massima profondità e il numero di stimatori, la validazione del GridSearch tendeva a scegliere sempre i valori più alti tra quelli proposti durante i test. Pertanto, sono stati cercati dei numeri sufficientemente elevati in modo da ottenere prestazioni accettabili dall'algoritmo ma che comunque non inficiassero troppo sulle sue performances: dunque la massima profondità degli alberi è stata settata spesso intorno a 20 e il numero di stimatori intorno a 15 (ad esempio, optare per un valore come 1000 per questi parametri probabilmente avrebbe portato a un risultato leggermente migliore ma l'algoritmo ci avrebbe impiegato ore a terminare tutti i calcoli, senza considerare il fatto che un albero di decisione così profondo rischierebbe anche di perdere la sua interpretabilità).

I primi esperimenti condotti con il Random Forest e con i suoi parametri appena settati sono stati effettuati con lo scopo principale di dare vita a dei buoni classificatori con tutti i dati che si avevano a disposizione, anche per iniziare a studiare il loro comportamento di fronte ad essi. Perciò è stata scritta una prima funzione per dividere il dataset in train e test, sfruttando un metodo preesistente in Scikit-learn già predisposto a svolgere questo compito, senza tenere conto dell'informazione temporale dei dati:

```
def split(tabella, testsize):
    x = tabella.drop(["ciclo", "numWashers"], axis=1)
    x = normalizza_classica(x)
    y = tabella["numWashers"]
    X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size = testsize, random_state=0)
    return X_train, X_test, y_train, y_test
```

Questa funzione si occupa di dividere le features (e normalizzarle) dall'etichetta del dataset creando due diverse strutture dati (denominate x e y), per poi passarle come parametri alla funzione di split, a cui viene specificato dall'utente la dimensione, in percentuale, del dataset di test. In questo modo si ottengono due partizioni tramite un campionamento non stratificato: "X_train" e "y_train" verranno impiegate nell'addestramento del modello di classificazione, "X_test" sarà invece usata per effettuare le predizioni sui valori di "*numWashers*", che verranno a loro volta paragonati al vettore "y_test" per quantificare la bontà del modello ottenuto.

I risultati presentati in questo capitolo saranno quelli più significativi ai fini delle indagini condotte e saranno visualizzati tramite le cosiddette matrici di confusione, ovvero delle tabelle dove ogni riga rappresenta i valori reali di una classe e ogni colonna i suoi valori predetti e dove quindi è possibile osservare i dati classificati correttamente sulla diagonale principale. Oltre alle matrici di confusione, per ogni esperimento verranno riportati il valore numerico dell'accuratezza e quelli di richiamo e precisione per ogni classe.

Il primo risultato mostrato è stato ottenuto impostando la dimensione del dataset di test pari al 10% del totale (per cui il dataset di train è automaticamente il restante 90%) e si trova in **tabella 5.2**.

	predetti								
	numWashers	0	1	2	3	4	somma	richiamo	precisione
reali	0	246	0	2	10	0	258	0.953	0.995
	1	0	552	2	3	0	557	0.991	0.985
	2	1	5	629	26	0	661	0.951	0.966
	3	0	2	9	794	2	807	0.983	0.933
	4	0	1	9	18	73	101	0.722	0.973
	somma	247	560	651	851	75	2384	accuratezza	0.962

Tabella 5.2: Matrice di confusione (train = 90% ottenuto con campionamento casuale)

Anche se i valori mostrati a primo impatto potrebbero sembrare ottimi (tutte e tre le metriche sono prossime ad 1, che significa che quasi tutti i dati sono stati classificati correttamente), un risultato di questo genere potrebbe nascondere delle problematiche. Infatti la grande differenza di dimensione tra il set di train e quello di test, unita al campionamento casuale e al consistente numero di features, potrebbe aver portato alla creazione di un modello eccessivamente complesso che si sarebbe adattato bene a qualunque tipologia di dati. Si tratta, molto probabilmente, di una problematica di *overfitting*, motivo per cui questo primo modello è stato scartato.

Un secondo tentativo, sempre nell'ottica di esplorare diverse casistiche per capire il comportamento del classificatore, è stato effettuato provando la situazione opposta alla

precedente: la dimensione del dataset di test è stata impostata pari addirittura al 90% dei dati, lasciando soltanto il 10% per il set di train. Questa prova è stata fatta anche per vedere se, cambiando così drasticamente le percentuali, sarebbero variati anche i risultati, dimostrando in un certo senso la presenza di *overfitting* nel modello descritto in precedenza. Il risultato è mostrato nella **tabella 5.3**.

	predetti								
	numWashers	0	1	2	3	4	somma	richiamo	precisione
reali	0	287	694	372	707	93	2153	0.133	0.087
	1	1106	1448	988	1189	100	4831	0.3	0.23
	2	687	1382	2737	695	90	5591	0.49	0.439
	3	989	2621	1966	1897	364	7837	0.242	0.395
	4	238	158	178	312	154	1040	0.148	0.192
	somma	3307	6303	6241	4800	801	21452	accuratezza	0.304

Tabella 5.3: Matrice di confusione (train = 10% ottenuto con campionamento casuale)

Si può subito notare che i risultati sono decisamente scadenti, specialmente nei casi di richiamo e precisione dei gruppi 0 e 4 (quelli meno numerosi), e che l'accuratezza generale del modello è appena superiore al 30%: molto probabilmente il 10% è una percentuale troppo piccola per addestrare adeguatamente il classificatore.

I successivi esperimenti sono stati effettuati utilizzando degli altri metodi per spezzare il dataset nelle due parti. La prima tipologia di analisi è stata condotta tramite una funzione in grado di mantenere l'ordine temporale dei cicli e quindi di inserire una percentuale fissa di dati ordinati per data ed ora nel dataset di train. L'obiettivo di queste analisi è stato principalmente quello di provare a scoprire se una piccola porzione di dati, raccolti in ordine di tempo, potesse fornire una buona predizione per quelli futuri, appartenenti a una diversa finestra temporale. Come appena detto, le percentuali usate per il train in questa fase sono state sempre abbastanza basse al fine di trovare una quantità minima di dati utilizzabili appunto per la creazione del modello di predizione: i valori usati per tale parametro sono stati fatti variare dal 5% al 30%. La seconda tipologia di analisi, molto simile alla precedente per quanto riguarda i valori della percentuale di dati presenti nel train set, invece è stata portata avanti sfruttando un'altra

funzione per la divisione del dataset, costruita appositamente per inserire nel train set una percentuale fissa di dati per ogni gruppo, raccolta però tramite un campionamento casuale e non in ordine di tempo. Sebbene questi due tipi di esperimenti possano sembrare simili, in realtà hanno portato a risultati incredibilmente diversi. Nelle **tabelle 5.4** e **5.5** sono mostrate rispettivamente due matrici di confusione: nella prima il dataset di train contiene il primo 10% dei cicli di ogni diverso gruppo, mentre nella seconda esso è composto sempre dal 10% dei cicli di tutti i gruppi, raccolti però in maniera casuale. Queste due casistiche sono state riportate di seguito a titolo di esempio, poiché rispecchiano bene i risultati di tutti gli esperimenti della stessa tipologia che sono stati provati, cambiando soltanto il valore della percentuale.

	predetti								
	numWashers	0	1	2	3	4	somma	richiamo	precisione
reali	0	373	757	350	595	78	2153	0.173	0.089
	1	1317	1620	689	1078	127	4831	0.335	0.228
	2	723	1793	2546	368	161	5591	0.455	0.455
	3	1459	2738	1863	1357	420	7837	0.173	0.374
	4	326	204	151	235	124	1040	0.119	0.136
	somma	4198	7112	5599	3633	910	21452	accuratezza	0.281

Tabella 5.4: Matrice di confusione (train = primo 10% di ogni gruppo)

	predetti								
	numWashers	0	1	2	3	4	somma	richiamo	precisione
reali	0	372	890	255	631	5	2153	0.173	0.734
	1	69	3768	248	742	4	4831	0.780	0.695
	2	8	65	5285	233	0	5591	0.945	0.861
	3	7	275	255	7298	2	7837	0.931	0.782
	4	51	425	92	433	39	1040	0.038	0.780
	somma	507	5423	6135	9337	50	21452	accuratezza	0.781

Tabella 5.5: Matrice di confusione (train = estrazione casuale del 10% dei cicli da ogni gruppo)

I risultati mostrati nella prima matrice non sono affatto buoni: gli elementi presenti sulla diagonale principale sono una minoranza rispetto agli altri, e i valori delle metriche calcolate sono tutti quasi sempre inferiori al 50%. In generale tutti gli esperimenti

condotti mantenendo l'ordine temporale dei cicli non sono andati a buon fine, perché evidentemente, per esempio, il primo 10% dei dati di un gruppo non è adatto a predire la parte restante. Può essere infatti che la struttura dei dati e i valori delle features cambino nel tempo, e il modello per funzionare adeguatamente avrebbe bisogno di dati relativi al funzionamento standard della macchina, formati da gruppi il più possibile coesi tra loro.

La seconda tipologia di esperimenti è andata meglio: si può notare che l'accuratezza del modello mostrato arriva al 78% e che gli unici problemi si hanno in corrispondenza del richiamo dei gruppi più esterni (ciò significa che i dati reali dei gruppi 0 e 4 spesso vengono classificati negli altri gruppi). Siccome l'unica vera differenza tra i due modelli stava nel metodo tramite cui era creato il dataset di train, si può presupporre che estrarre casualmente una percentuale fissa di dati da ogni gruppo faccia sì che il modello venga addestrato meglio, avendo la possibilità di "vedere" più casistiche, relative a più finestre temporali, e quindi porti a risultati più accurati. Nonostante la bontà di questa tipologia di prove, esse risolvono le problematiche di partenza solamente in parte: infatti, seguendo questa strada, c'è il rischio che il modello generato, una volta utilizzato con dati nuovi appartenenti a diverse finestre temporali e quindi logicamente almeno un po' distanti dal dataset di train, non si adatti come dovrebbe, portando a risultati falsati.

Esperimenti simili a quelli visti in precedenza sono stati effettuati con un dataset diverso, derivante dall'applicazione della PCA nella fase di preprocessing anziché della matrice di correlazione, e un esempio di matrice di confusione ottenuta è mostrata in **tabella 5.6**.

	predetti								
	numWashers	0	1	2	3	4	somma	richiamo	precisione
reali	0	533	781	489	340	10	2153	0,248	0,457
	1	308	2855	945	706	17	4831	0,591	0,551
	2	135	626	4049	777	4	5591	0,724	0,625
	3	109	502	798	6422	6	7837	0,819	0,753
	4	82	420	197	282	59	1040	0,057	0,615
	somma	1167	5184	6478	8527	96	21452	accuratezza	0,649

Tabella 5.6: Matrice di confusione ottenuta con la PCA con parametro pari a 20 (train = estrazione casuale del 10% dei cicli da ogni gruppo)

Il parametro scelto in questo particolare caso è 20 (pertanto, il dataset usato ha 20 attributi) e la divisione tra test e train è stata fatta nello stesso modo descritto prima, ovvero prendendo percentuali fisse estratte casualmente da ogni gruppo. Come si può notare, mantenendo la dimensione del train pari al 10%, il risultato ottenuto è, in termini di accuratezza, peggiore rispetto al caso precedente che è direttamente paragonabile ad esso. Generalmente ogni esperimento con la PCA, pur variando il numero di attributi, ha portato risultati simili o leggermente peggiori rispetto ai casi base a parità di percentuale del dataset di train, e per questo motivo gli esperimenti effettuati con questa tecnica sono stati interrotti dopo questa serie di prove.

Ciò che fino a questo punto è stato possibile dedurre dai vari tentativi è che una percentuale pari al 10% del dataset potrebbe anche essere adeguata a classificare il restante 90%, ma solo a patto che questo 10% non provenga da dati necessariamente vicini tra loro temporalmente, ma che venga estratto casualmente dall'intero dataset (mantenendo comunque le giuste proporzioni dei gruppi nelle due partizioni). Le successive indagini sono state volte alla ricerca di una nuova possibile soluzione per questa problematica.

Gli esperimenti finali con il Random Forest sono stati strutturati appositamente per cercare una quantità minima di dati in grado di predire correttamente nuovi dati, e allo stesso tempo mantenere l'informazione temporale sulle serie storiche dei cicli. Questa fase di lavoro è stata caratterizzata quindi da 10 esperimenti, numerati da 1 a 10 in modo da poter essere facilmente identificati, tramite cui sono stati creati altrettanti modelli di classificazione. La logica secondo cui essi sono stati condotti è stata quella di non considerare sempre l'intero dataset, bensì soltanto una parte di esso, identificata da una percentuale crescente esperimento dopo esperimento, per analizzare il comportamento dei dati nel tempo. Pertanto, per l'esperimento numero 1 è stato effettuato sul 10% dei cicli, il secondo sul 20%, il terzo sul 30% e così via fino al decimo, che ha sfruttato l'intero dataset. Un'altra caratteristica importante di questo ciclo di esperimenti è che i dati sono stati selezionati in maniera stratificata sulle classi e in ordine temporale: per fare un esempio, dunque, il dataset del primo esperimento contiene il primo 10% dei dati di ogni gruppo, mentre la restante parte dei cicli è stata ignorata. In questo modo sono state mantenute le solite proporzioni tra i vari gruppi (gli

elementi del gruppo 3, ad esempio, saranno sempre in maggioranza) e in ogni esperimento compaiono sempre cicli di ogni classe. Per ogni esperimento inoltre, il dataset utilizzato è stato a sua volta diviso in train e test per poter valutare le metriche di accuratezza, precisione e richiamo del modello utilizzando queste percentuali: 30% per il test e 70% per il train.

In **tabella 5.7** è presente uno schema che riassume le informazioni più importanti per capire questa serie di esperimenti: per ognuno di essi vengono riportati il numero e la percentuale di cicli totali utilizzati, la cardinalità di ogni gruppo e il risultato relativo all'accuratezza ottenuta.

esperimento	#cicli totali	% cicli usati	#g0	#g1	#g2	#g3	#g4	accuratezza
1	2381	10%	239	536	621	870	115	0.96932
2	4765	20%	478	1073	1242	1741	231	0.96927
3	7148	30%	717	1610	1863	2612	346	0.93200
4	9530	40%	956	2146	2484	3482	462	0.90528
5	11915	50%	1196	2683	3106	4353	577	0.88591
6	14299	60%	1435	3220	3727	5224	693	0.86583
7	16680	70%	1674	3756	4348	6094	808	0.87937
8	19064	80%	1913	4293	4969	6965	924	0.87467
9	21447	90%	2152	4830	5590	7836	1039	0.87013
10	23833	100%	2392	5367	6212	8707	1155	0.86943

Tabella 5.7: Riassunto informazioni sui 10 esperimenti finali con il Random Forest

Come ci si poteva aspettare, l'aumentare dei cicli per i dataset usati, che simula in un certo senso il passare del tempo, fa sì che ci sia un trend decrescente nel valore dell'accuratezza (che è possibile osservare in **figura 5.6**): questo può essere un segno del fatto che la macchina è sottoposta ad usura e che nei giorni il comportamento dei gruppi subisca piccole variazioni (e a questo sarebbero dovuti quindi gli errori del classificatore, che rischia sempre di più di confondere le classi tra loro).

I valori ottenuti per richiamo e precisione sono mostrati rispettivamente nelle **tabelle 5.8** e **5.9**, e il loro trend, anch'esso prevalentemente decrescente, nelle **figure 5.7** e **5.8**.

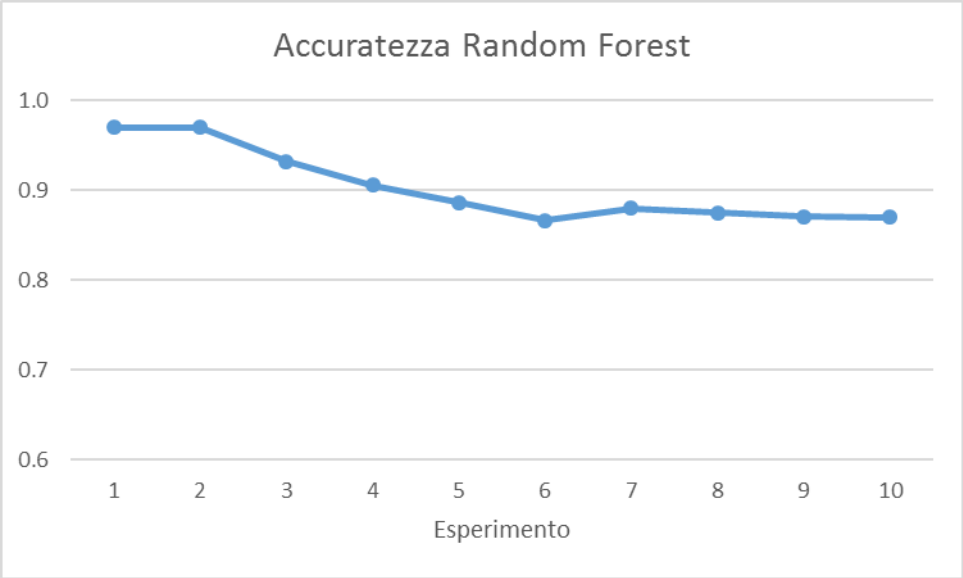

Figura 5.6: Accuratezza dei 10 esperimenti

esperimento	richiamo g0	richiamo g1	richiamo g2	richiamo g3	richiamo g4
1	0.9722	0.9814	0.9893	0.9924	0.6286
2	0.8958	0.9845	0.9893	1.0000	0.7143
3	0.7870	0.9360	0.9714	0.9898	0.5673
4	0.6760	0.9037	0.9745	0.9809	0.4460
5	0.5989	0.8957	0.9700	0.9686	0.3621
6	0.4942	0.9038	0.9669	0.9528	0.2596
7	0.5527	0.9104	0.9801	0.9584	0.2757
8	0.5226	0.9193	0.9765	0.9507	0.2770
9	0.5387	0.9152	0.9684	0.9528	0.1955
10	0.5223	0.9249	0.9630	0.9422	0.2795

Tabella 5.8: Valori del richiamo per i 10 esperimenti finali con il Random Forest

esperimento	precisione g0	precisione g1	precisione g2	precisione g3	precisione g4
1	0.9859	0.9080	0.9737	1	1
2	0.9485	0.9324	0.9840	0.9849	1
3	0.8901	0.8499	0.9766	0.9640	0.9516
4	0.8050	0.8163	0.9491	0.9544	0.9254
5	0.7517	0.7837	0.9607	0.9288	0.9403
6	0.7220	0.7496	0.9442	0.9211	0.8438
7	0.8176	0.7708	0.9495	0.9130	0.9710
8	0.7692	0.7654	0.9541	0.9131	0.9390

9	0.8150	0.7711	0.9448	0.8956	0.8971
10	0.8047	0.7618	0.9462	0.9022	0.9238

Tabella 5.9: Valori della precisione per i 10 esperimenti finali con il Random Forest

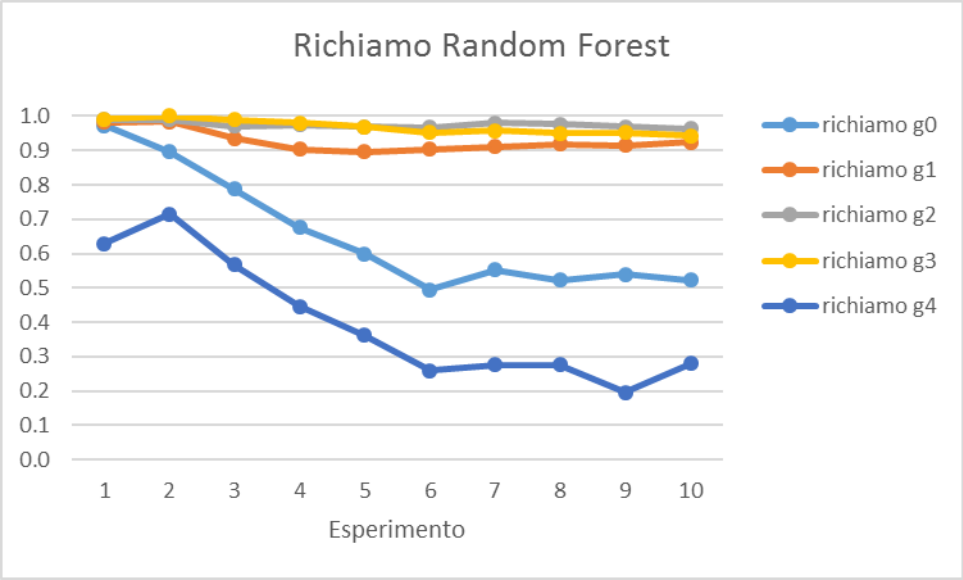

Figura 5.7: Richiamo per ogni gruppo dei 10 esperimenti

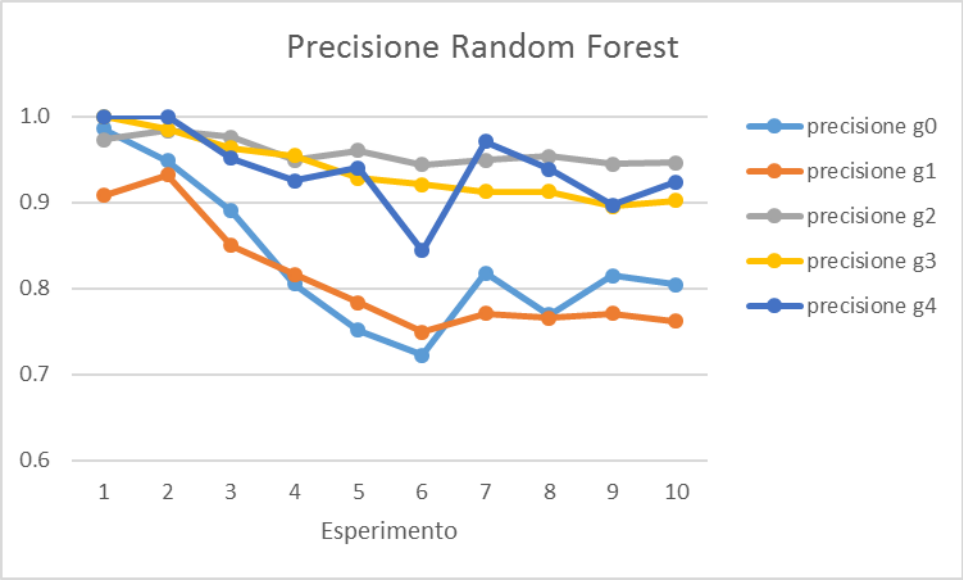

Figura 5.8: Precisione per ogni gruppo dei 10 esperimenti

Per quanto riguarda la precisione, i valori ottenuti sono sempre abbastanza alti, soprattutto per i gruppi 2, 3 e 4, mentre le curve delle classi 1 e 0 evidenziano una decrescita maggiore con l'aumentare dei dati negli esperimenti, rimanendo tuttavia sempre al di sopra del 70%. Il fatto che i gruppi 0 e 1 abbiano ottenuto i valori di precisione più bassi significa che il classificatore tende ad assegnare loro una percentuale considerevole di dati in realtà appartenenti ad altre classi. I valori ottenuti per il richiamo invece sono ottimi per quanto concerne i gruppi "interni" e più numerosi, 1, 2 e 3 (pressoché sempre oltre il 90% in tutti gli esperimenti), mediocri per il gruppo 0 e piuttosto bassi per il gruppo 4. Evidentemente, il classificatore predice male i dati di questi ultimi gruppi, assegnandoli ad altre classi, mentre invece associa quasi sempre in modo corretto i cicli dei gruppi "interni".

5.2.2 – MODELLI CON L'ISOLATION FOREST

Sempre nell'ambito della ricerca di un modello generale per la previsione del comportamento del robot, è stata pensata un'altra possibile soluzione che sfrutta però un diverso algoritmo di classificazione, l'Isolation Forest. Il suo funzionamento è sostanzialmente lo stesso di una tecnica per la ricerca di outliers in un dataset poiché esso viene addestrato come un qualunque algoritmo di classificazione con la differenza che nel dataset di train devono comparire solamente elementi appartenenti alla stessa classe, e in seguito il modello generato sarà in grado di distinguere nuovi elementi appartenenti alla classe di train o che ne sono esterni. I dati di classi diverse vengono riconosciuti appunto come outliers dall'algoritmo che li classifica come anomalie, ed il suo funzionamento è piuttosto simile a quello del già utilizzato Random Forest, nel senso che viene costruito basandosi su un insieme di alberi di isolamento binari.

In particolare, nel caso applicativo di questa tesi, la sua utilità può essere spiegata in questo modo: nel caso in cui si abbiano pochi dati a disposizione, o se ne abbiano soltanto alcuni appartenenti alla stessa classe, ma che comunque possano ben rappresentare il corretto funzionamento della macchina, potrebbe non essere possibile costruire un classificatore standard che per lavorare ha bisogno di almeno due etichette distinte. In questo caso, i dati raccolti dovrebbero fungere da train per un modello in

grado di riconoscere nelle analisi successive se i nuovi dati nascondono un malfunzionamento (caso in cui il classificatore riconosce la differenza rispetto ai dati di partenza, ponendoli come outliers), o se la macchina sta continuando a funzionare nello stesso modo (in questo caso i dati verrebbero messi nella stessa classe dei dati originali). Pertanto l'Isolation Forest si potrebbe rivelare molto utile nel rilevare dei cambiamenti nel tempo del segnale di corrente senza considerare il cambio di numero di rondelle (per il quale resterebbe necessario utilizzare il modello generale precedentemente descritto).

In Python è possibile costruire un modello con l'Isolation Forest per mezzo, ancora una volta, di una classe di Scikit-learn che porta lo stesso nome dell'algoritmo. Anche in questo caso, esiste un parametro molto importante da settare, chiamato "contamination", un numero decimale che va a indicare la percentuale, stimata dall'utente, di outliers presenti nel dataset di train. È pertanto necessario fare alcune prove di verifica per cercare una buona stima di questo parametro prima di utilizzare il modello generato con dati completamente nuovi. La metodologia seguita nel progetto per gli esperimenti con l'Isolation Forest è descritta in questi passi:

1. Sono state definite tre diverse finestre temporali che coincidono all'incirca con le settimane a disposizione nel dataset, sulle quali sono stati costruiti altrettanti modelli. Il motivo di questa operazione è sempre dovuto al fatto che lo scopo principale del modello generale resta quello di trovare dei trend di cambiamento dei dati nel tempo, e quello di dividere le analisi in base alla data dei cicli potrebbe essere un buon modo per cercare queste evidenze. La prima finestra temporale va dal 10 ottobre 2018 al 15 ottobre compreso, per la seconda invece sono stati considerati i cicli fino al 23 ottobre (sempre partendo dal 10) e per l'ultima è stato considerato finalmente l'intero dataset. Nella **tabella 5.10** è presente la ripartizione dei vari gruppi per queste finestre temporali di cui si è parlato finora. È importante notare che dal 10 al 15 ottobre non sono presenti cicli del gruppo 0, che il gruppo 1 e 4 sono concentrati totalmente tra queste stesse date, mentre le cardinalità dei gruppi 2 e 3 crescono nel tempo.

cicli per gruppo	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
gruppo 0	0	2392	2392
gruppo 1	5367	5367	5367
gruppo 2	1126	1733	6212
gruppo 3	674	4059	8707
gruppo 4	1155	1155	1155
totale	8322	14706	23833

Tabella 5.10: Distribuzione dei cicli di ogni gruppo in tre finestre temporali

2. A turno, ogni gruppo è stato utilizzato come dataset di train per i modelli, nel senso che tale set veniva composto solamente da dati appartenenti al gruppo selezionato. È stato creato anche un dataset di test composto da una piccola percentuale (10%) dello stesso gruppo, in modo da verificare se il classificatore fosse in grado di riconoscere dati da considerare "corretti". Per ogni gruppo è stato trovato un possibile valore del parametro di contaminazione sfruttando anche i risultati delle prove su questo test set, e i valori trovati, che riflettono effettivamente gli andamenti delle silhouette dei gruppi calcolati in fase di preprocessing per quanto riguarda la loro coesione interna, sono mostrati nella **tabella 5.11**. Chiaramente, più outliers sono presenti in un gruppo, meno è probabile che il modello sarà buono.

gruppo	outliers stimati
0	1%
1	5%
2	8%
3	15%
4	10%

Tabella 5.11: Outliers stimati per ogni gruppo per l'applicazione dell'Isolation Forest

3. I modelli generati infine sono stati testati su tutti gli altri gruppi non appartenenti al train e sono stati valutati tramite il calcolo dell'accuratezza. Nella successiva descrizione dei risultati ottenuti, tale metrica è stata calcolata seguendo questa logica: per il set di test (appartenente allo stesso gruppo del train) essa è rappresentata dalla percentuale di dati classificati correttamente all'interno del

gruppo di train, mentre per tutti gli altri gruppi (quelli esterni) è intesa come la percentuale di dati classificati come outliers. I modelli generati portano come risultato, a differenza del Random Forest e degli altri algoritmi che restituiscono solitamente un'etichetta per ogni dato, un elenco di 1 e -1, dove 1 indica l'appartenenza al gruppo iniziale e -1 la presenza di un outlier. I dati, prima di essere processati, non sono stati normalizzati in questo caso, perché non era necessario effettuare calcoli di distanza tra essi e perché mantenendo i valori originali delle features le differenze tra i gruppi erano maggiormente evidenziabili.

I risultati ottenuti con l'Isolation Forest vengono mostrati nelle tabelle seguenti, una per ogni gruppo usato come train. Invece, per ogni gruppo su cui è stato testato il modello vengono riportate le percentuali di dati classificati come esterni e come interni al gruppo base, divise per ogni finestra temporale. Il primo gruppo per cui è stato creato un modello di questo tipo è il numero 1 (risultati visibili in **tabella 5.12**) perché è il più coeso tra i gruppi "interni" e potrebbe rappresentare piuttosto bene il corretto funzionamento del robot, dopodiché sono stati creati gli altri modelli supponendo che il comportamento standard del robot fosse assimilabile volta per volta ai gruppi di train.

GRUPPO DI TRAIN: 1

gruppo di test passato all'IF	accuratezza classificazione	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
test set g1	dentro g1	93%	93%	93%
	fuori da g1	7%	7%	7%
gruppo 2	dentro g1	10%	8%	2%
	fuori da g1	90%	92%	98%
gruppo 3	dentro g1	5%	11%	5%
	fuori da g1	95%	89%	95%
gruppo 4	dentro g1	1%	1%	1%
	fuori da g1	99%	99%	99%
gruppo 0	dentro g1	no data	0%	0%
	fuori da g1	no data	100%	100%
tutti i gruppi	dentro g1	5%	6%	3%
	fuori da g1	95%	94%	97%

Tabella 5.12: Risultati dell'Isolation Forest con gruppo di train 1

I risultati sono generalmente buoni per tutti i gruppi: oltre il 90% dei dati di tutti i gruppi viene classificato come esterno al gruppo 1, mentre per quanto riguarda il test set di tale gruppo, è riconosciuto interno ad esso nel 93% dei suoi dati. Nelle **tabelle 5.13, 5.14, 5.15** e **5.16** sono presentati i risultati del modello con tutti gli altri gruppi impostati come dataset di train.

GRUPPO DI TRAIN: 2

gruppo passato all'IF	accuratezza classificazione	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
test set g2	dentro g2	95%	93%	91%
	fuori da g2	5%	7%	9%
gruppo 1	dentro g2	7%	4%	1%
	fuori da g2	93%	96%	99%
gruppo 3	dentro g2	16%	12%	26%
	fuori da g2	84%	88%	74%
gruppo 4	dentro g2	4%	5%	19%
	fuori da g2	96%	95%	81%
gruppo 0	dentro g2	no data	0%	0%
	fuori da g2	no data	100%	100%
tutti i gruppi	dentro g2	8%	6%	21%
	fuori da g2	92%	94%	79%

Tabella 5.13: Risultati dell'Isolation Forest con gruppo di train 2

GRUPPO DI TRAIN: 3

gruppo passato all'IF	accuratezza classificazione	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
test set g3	dentro g3	85%	82%	85%
	fuori da g3	15%	18%	15%
gruppo 1	dentro g3	1%	1%	1%
	fuori da g3	99%	99%	99%
gruppo 2	dentro g3	1%	8%	18%
	fuori da g3	99%	92%	82%
gruppo 4	dentro g3	30%	37%	82%
	fuori da g3	70%	63%	18%

gruppo 0	dentro g3	no data	0%	0%
	fuori da g3	no data	100%	100%
tutti i gruppi	dentro g3	5%	5%	14%
	fuori da g3	95%	95%	86%

Tabella 5.14: Risultati dell'Isolation Forest con gruppo di train 3

GRUPPO DI TRAIN: 4

gruppo passato all'IF	accuratezza classificazione	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
test set g4	dentro g4	86%	86%	86%
	fuori da g4	14%	14%	14%
gruppo 1	dentro g4	0%	0%	0%
	fuori da g4	100%	100%	100%
gruppo 2	dentro g4	1%	1%	3%
	fuori da g4	99%	99%	97%
gruppo 3	dentro g4	55%	29%	34%
	fuori da g4	45%	71%	66%
gruppo 0	dentro g4	no data	0%	0%
	fuori da g4	no data	100%	100%
tutti i gruppi	dentro g4	5%	9%	14%
	fuori da g4	95%	91%	86%

Tabella 5.15: Risultati dell'Isolation Forest con gruppo di train 4

GRUPPO DI TRAIN: 0

gruppo passato all'IF	accuratezza classificazione	dal 10 al 15 ottobre	dal 10 al 23 ottobre	dal 10 al 30 ottobre
test set g0	dentro g0	no data	98%	98%
	fuori da g0	no data	2%	2%
gruppo 1	dentro g0	no data	1%	1%
	fuori da g0	no data	99%	99%
gruppo 2	dentro g0	no data	0%	0%
	fuori da g0	no data	100%	100%
gruppo 3	dentro g0	no data	0%	0%
	fuori da g0	no data	100%	100%

gruppo 4	dentro g0	no data	0%	0%
	fuori da g0	no data	100%	100%
tutti i gruppi	dentro g0	no data	1%	1%
	fuori da g0	no data	99%	99%

Tabella 5.16: Risultati dell'Isolation Forest con gruppo di train 0

Una volta osservati i valori di accuratezza ottenuti è possibile fare alcuni commenti:

- Per quanto riguarda il gruppo 0, esso è talmente diverso da tutti gli altri che viene sempre riconosciuto come esterno quando è passato come test, mentre quando svolge la funzione di train si ottengono valori di accuratezza molto vicini al 100% per tutti i gruppi. Tale modello funziona bene ma è anche necessario riconoscere che difficilmente il gruppo 0, che è un gruppo "esterno" e poco numeroso, potrebbe rappresentare il funzionamento standard del robot.

- I modelli creati con i gruppi 2, 3 e 4 sono generalmente buoni, ma le loro prestazioni tendono a peggiorare nel tempo, ovvero ogni volta che vengono aggiunti nuovi dati. Inoltre i gruppi 3 e 4 spesso vengono confusi tra loro, infatti i valori più bassi di accuratezza si hanno proprio quando nel modello con gruppo di train pari a 3 viene predetto il quarto gruppo, e viceversa.

- Tutti i set di test contenenti lo stesso gruppo di quello di train funzionano adeguatamente: i peggiori sono nuovamente quelli dei gruppi 3 e 4 che comunque presentano un'accuratezza superiore all'80%.

- Il modello migliore, tolto quello del gruppo 0 di cui è stato già parlato, è quello del gruppo 1, che peraltro non mostra segni di peggioramento nel tempo.

5.3 – MODELLO EVOLUTIVO

Tramite i risultati raggiunti dal modello generale è stato possibile ottenere alcuni buoni classificatori, studiarne il comportamento in relazione alle serie storiche dei dati, e dedurre alcuni valori di percentuale e numerosità dei dataset utilizzati affinché i calcoli

delle metriche selezionate risultassero soddisfacenti. Tuttavia, si presuppone che i dati utilizzati fossero tutti relativi ad un funzionamento buono, o quantomeno standard, del robot e che ciò che differenziasse il loro andamento non fosse altro se non il numero di rondelle settato arbitrariamente da chi di dovere prima della raccolta del segnale di corrente. Pertanto, risulta complicato poter effettuare operazioni di manutenzione predittiva finché non si conosce quale sia la differenza tra un buono e un cattivo funzionamento: proprio per portare una possibile soluzione a questa problematica, si è pensato di generare una nuova tipologia di modello, denominato modello evolutivo per il suo obiettivo primario di sapersi adattare a condizioni mutevoli e non ordinarie della macchina.

L'esigenza di questo modello nasce quindi dal tentativo di simulare il meglio possibile, con i dati a disposizione, una casistica possibile nella realtà industriale. Infatti solitamente, dopo aver addestrato un modello di classificazione con i dati di train, questo viene utilizzato con informazioni completamente nuove la cui etichetta reale non è conosciuta: ovviamente chi lavora con il robot sa quante rondelle sono presenti, ma questo valore potrebbe non corrispondere al funzionamento standard della macchina con tale numero poiché quest'ultima, soggetta ad usura, potrebbe aver variato il suo comportamento. In questo caso sarebbe necessario un intervento esterno per ritensionare la cinghia di trasmissione e ripristinare il corretto funzionamento del robot.

La metodologia impiegata nella creazione del modello evolutivo è sintetizzata in questi step, che verranno approfonditi al completo in seguito:

1. Per poter veramente simulare l'arrivo di dati nuovi di cui l'etichetta non è nota a priori, il training del modello è avvenuto diversamente rispetto al modello generale: dal dataset completo sono stati estratti i gruppi principali ed "interni" (i gruppi 1, 2 e 3) per addestrare il modello, mentre i gruppi "esterni" gli sono stati passati successivamente. Sono state effettuate più prove, cambiando le configurazioni di train e test per provare a individuare quella più realistica. Ad ogni modo il modello generato conosce soltanto l'esistenza di alcune classi, ed è obbligato a classificare all'interno di esse i dati esterni.

2. La scelta dell'algoritmo utilizzato per il modello evolutivo è ricaduta sempre sul Random Forest, esattamente come nel modello generale. Allo stesso modo è avvenuta la procedura di test e settaggio dei parametri, già descritta in precedenza.

3. Una volta ottenuti i risultati della classificazione, non essendo possibile calcolare accuratezza, richiamo e precisione, i risultati sono stati valutati tramite il calcolo di alcuni indici di dispersione dei gruppi: la silhouette (già utilizzata tra l'altro in fase di preprocessing, ma con uno scopo diverso), l'MSE ed il MAAPE.

La scelta delle classi "interne" per il dataset di train può essere giustificata grazie ad una serie di ragioni: innanzitutto, in ambito industriale, la situazione con 0 rondelle è generalmente utilizzata come test della macchina e non durante la produzione vera e propria, e anche quella con 4 rondelle è da considerarsi un caso estremo e non standard. Non a caso i gruppi 1, 2 e 3 formano, insieme, circa l'85% del dataset totale. Pertanto i gruppi 0 e 4 potrebbero riprodurre, nel contesto di questo progetto, la situazione in cui il robot smette di funzionare bene e cambia comportamento: si tratta di una buona simulazione della realtà, in cui spesso si ha a disposizione un modello costruito con i dati "buoni" e a poco a poco ne vengono aggiunti di nuovi di cui non si conosce lo stato. Nell'ambito del modello evolutivo, i cicli sono sempre stati ordinati per data ed ora e spesso sono state effettuate diverse analisi dividendo i dati in base al tempo, sempre con l'idea di sfruttare questa informazione per rendere più realistico il caso affrontato dai classificatori.

Mentre per quanto riguarda i vari esperimenti del modello evolutivo sono state sfruttate diverse configurazioni di train e test set, i metodi di valutazione sono stati circa sempre gli stessi, motivo per cui verranno approfonditi in questa sezione del capitolo. Come già detto, la silhouette è un indice con dominio compreso tra 1 e -1 che valuta la coesione di un dato all'interno del proprio cluster. Durante la valutazione del modello evolutivo essa è calcolata per tutti i gruppi di train prima della classificazione e dopo di essa: infatti man mano che dati esterni vengono classificati e quindi aggiunti a tali gruppi, ciò che si dovrebbe osservare è un graduale abbassamento dei valori di tale indice, che

sottintende una diminuzione della coesione dei gruppi e una conseguente interpretazione come malfunzionamento della macchina.

Allo stesso scopo è stato anche calcolato l'MSE (*mean square error*) che rappresenta lo scostamento quadratico medio tra valori osservati e valori stimati. Il suo calcolo è stato effettuato utilizzando il valore medio di ogni record separatamente per ogni gruppo, grazie all'apposita funzione di Scikit-learn. Più si ottengono valori elevati, più i dati dei gruppi sono da considerare distanti tra loro.

Infine il MAAPE (*mean arctangent absolute percentage error*), l'ultimo indice utilizzato, è anch'esso una misura di scostamento percentuale tra valori osservati e stimati che sfrutta la funzione di arcotangente affinché in presenza di valori prossimi allo zero non si ottengano risultati indefiniti o tendenti ad infinito (nel caso in esame infatti sono molto frequenti valori prossimi allo zero specialmente per quanto riguarda le features relative alla parte iniziale del ciclo di lavoro). L'arcotangente permette di risolvere questa problematica passando ai radianti. La formula del MAAPE è la seguente:

$$MAAPE = \frac{1}{N}\sum_{t=1}^{N} arctan\left(\left|\frac{y_y - f_t}{y_t}\right|\right)$$

In questo progetto il MAAPE è stato utilizzato per confrontare le curve di silhouette prima e dopo la classificazione: y_t rappresenta i valori della silhouette base e f_t quelli ottenuti dopo aver aggiunto i nuovi dati predetti. Trattandosi anch'esso di un indicatore di errore, valori bassi indicano alta coesione e valori alti bassa coesione.

5.3.1 – PRIMA FASE

La prima fase di esperimenti condotti per il modello evolutivo riguarda principalmente lo studio dell'andamento della silhouette utilizzando tutti i dati a disposizione: il dataset di train in questo caso contiene tutti e tre i gruppi principali e, dopo aver addestrato il modello, quest'ultimo è stato usato per predire gli altri due gruppi in maniera separata e indipendente tra essi. Una volta ottenuti i risultati, è stato calcolato l'indice e sono stati

predisposti dei grafici per la sua visualizzazione. Bisogna sottolineare che i dati dei gruppi 0 e 4, prima di essere processati dal classificatore, sono stati privati dell'informazione sul numero di rondelle: considerandoli dunque come dati senza etichetta, il modello di predizione ha dovuto associare ad essi una delle classi "interne".

Per quanto riguarda questa prima fase di esperimenti, verranno semplicemente mostrati dei grafici che sintetizzano i risultati ottenuti. Essi contengono delle curve che rappresentano i valori della silhouette calcolata per i cicli dei gruppi 1, 2 e 3 prima e dopo la predizione (si ricorda che dopo tale predizione i dati, in realtà appartenenti a 0 e 4 sono stati inseriti nel DataFrame contenente il gruppo predetto). Ciò che è importante osservare è lo scostamento tra le curve. Al fine di semplificare la visualizzazione dei risultati, è stato applicato un campionamento di 1200 elementi per ogni gruppo di train poiché, dovendo osservare e paragonare due curve, è importante che esse contengano la stessa numerosità di punti (e chiaramente i gruppi ottenuti dopo la predizione sono più numerosi rispetto a prima). In ogni caso il campionamento è stato effettuato solamente dopo il calcolo della silhouette, che invece è stato eseguito per la totalità dei cicli.

Dunque, i risultati di questa prima analisi sono mostrati nelle **figure 5.9** e **5.10**. Nella prima è presente la differenza delle silhouette dopo aver effettuato la predizione sul gruppo 0, mentre nella seconda il gruppo predetto è stato il 4.

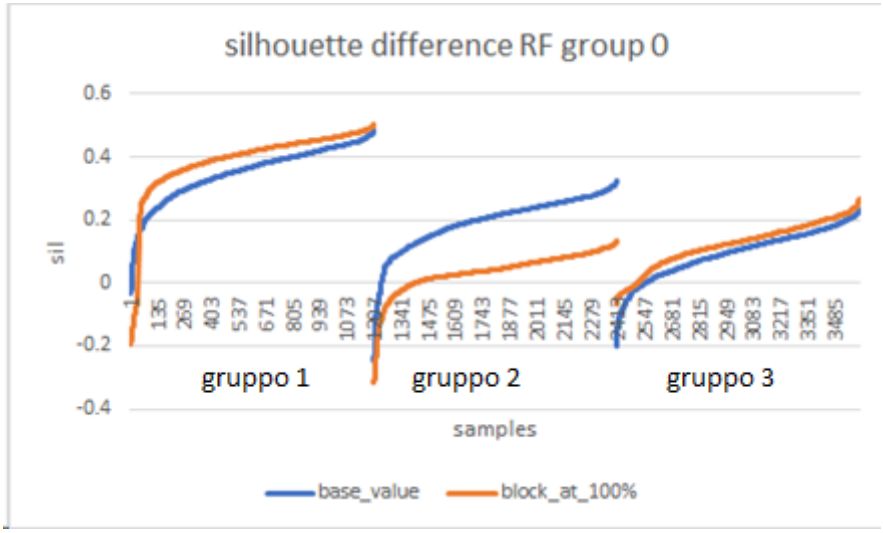

Figura 5.9: Differenza nella silhouette per gruppi 1, 2, 3 dopo la predizione del gruppo 0

Le curve blu rappresentano i valori base di silhouette per i gruppi 1, 2 e 3, e quelle arancioni (identificate come “block at 100%” nelle immagini) indicano la silhouette in seguito alla classificazione, e teoricamente dovrebbero sempre trovarsi al di sotto delle precedenti. La sigla RF nel titolo delle figure è una precisazione per spiegare che l’algoritmo utilizzato è sempre il Random Forest.

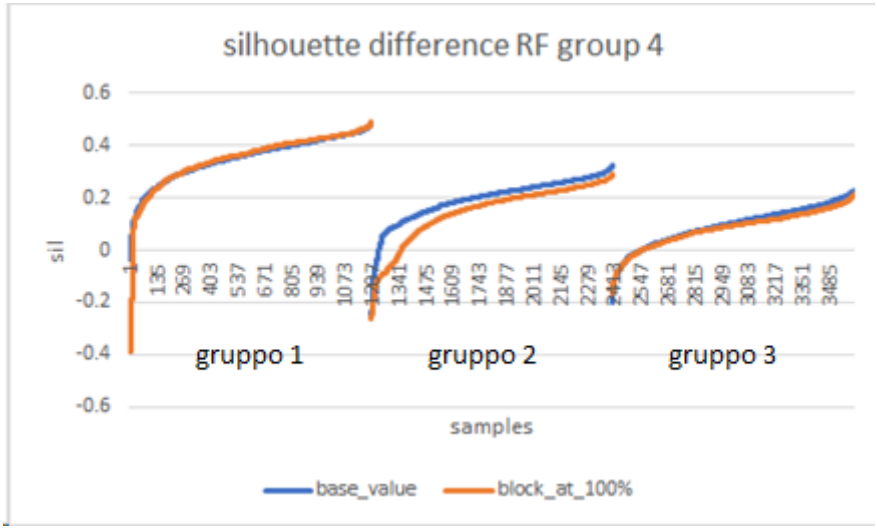

Figura 5.10: Differenza nella silhouette per gruppi 1, 2, 3 dopo la predizione del gruppo 4

Per quanto riguarda il gruppo 0, il degrado della silhouette è visibile soltanto nel gruppo 2, ed anzi, le due curve mostrano una spiccata differenza. La maggior parte degli elementi del gruppo 0 è stata proprio assegnata al gruppo 2, e tale fatto ha portato ad un notevole abbassamento dell’indice. Invece per gli altri due gruppi la silhouette mostra, seppur di poco, un aumento: questo significa una maggior coesione interna in seguito all’unione di alcuni dati esterni.

Per quanto riguarda le predizioni del gruppo 4 invece l’effetto è molto meno visibile: le curve dei gruppi 1 e 3 sono pressoché sovrapposte, mentre per il gruppo 2 c’è un lieve abbassamento; da qui si deduce che unire elementi del gruppo 4 non ha eccessivamente cambiato la coesione dei gruppi.

Questi risultati ottenuti sono abbastanza contrastanti e differenti dalle aspettative: pertanto è stato necessario indagare ulteriormente in questa stessa direzione per ottenere

modelli più efficaci nello scorgere il degrado della coesione dei gruppi, variando ad esempio la configurazione del dataset di train e utilizzando anche gli altri indici di dispersione.

5.3.2 – SECONDA FASE

La seconda fase di analisi del modello evolutivo si è dunque concentrata su una ricerca più approfondita del degrado della macchina nel tempo. Per sfruttare l'informazione temporale presente nel dataset sono stati effettuati alcuni accorgimenti: sono state definite 3 diverse fasce temporali, esattamente come già fatto durante la costruzione del modello generale, dove la prima contiene i cicli che vanno dal 10 al 15 ottobre, la seconda quelli dal 10 al 23 ottobre e la terza infine è formata dall'intero set. La distribuzione dei cicli all'interno di queste finestre temporali è la stessa già mostrata in **tabella 5.10**. In questa seconda fase di analisi tuttavia è stato deciso di eliminare il gruppo 0, poiché, come già accennato in precedenza, non si tratta di una casistica utilizzata realmente in ambito industriale e contiene informazioni troppo distanti dai restanti gruppi.

Dopo aver effettuato questa divisione dei dati in fasce temporali, è stata ideata un'altra tecnica per poter meglio scorgere il progressivo diminuire della silhouette nel tempo: ogni gruppo utilizzato per il test dei modelli generati non è stato passato tutto insieme al classificatore, bensì è stato precedentemente diviso in 5 sottogruppi di uguale dimensione in ordine temporale. Così facendo ogni sottogruppo, che contiene il 20% del gruppo di test, è stato processato dal modello e i suoi dati, dopo aver ottenuto la propria etichetta, sono stati aggiunti ad uno dei gruppi di test. La silhouette non è stata calcolata solamente all'inizio e alla fine del processo, come durante la prima fase, ma prima e dopo la predizione di ognuno dei sottogruppi. In questo modo, in seguito all'aggiunta di ogni blocco, è stato possibile osservare una diminuzione più graduale dell'indice di coesione.

Sono stati condotti, seguendo questa metodologia appena descritta e sfruttando il calcolo di silhouette, MSE e MAAPE, più esperimenti variando la configurazione del

dataset di train e del test del modello. Questi esperimenti sono elencati di seguito e sono identificati grazie ad un numero intero crescente:

1. Esperimento 1: il train comprende, come nella prima fase, i gruppi 1, 2 e 3, mentre il test è semplicemente formato dal gruppo 4

2. Esperimento 2: il train è formato dai due gruppi considerati "ottimali" per il funzionamento del robot, ossia le classi 1 e 2. I due gruppi di test, indipendenti tra loro, sono invece formati dai due gruppi restanti, 3 e 4.

3. Esperimento 3: in questo caso il train consiste nell'unione dei gruppi 2 e 3 poiché a livello industriale il 3 è considerato accettabile sebbene porti a una minore coesione interna. Il gruppo di test è nuovamente formato dalla classe 4.

I risultati di questi esperimenti saranno mostrati, separatamente per finestra temporale, per mezzo di una tabella che riassume le predizioni (ovvero come i dati del gruppo di test sono stati classificati), dal grafico dell'andamento della silhouette di ogni sottogruppo (mostrato sempre dopo un campionamento di 1200 elementi per ogni curva) e dal grafico della silhouette media ad ogni step temporale. Spesso sono stati scelti uno o più grafici (quelli meglio rappresentativi dei risultati raggiunti) per ogni caso di ogni esperimento. Per evitare di fornire informazioni ripetitive, spesso non sono stati riportati i grafici relativi all'andamento del MAAPE (che contengono la stessa informazione di quelli della silhouette) e dell'MSE (che inoltre contengono un risultato poco interpretabile e di difficile lettura).

ESPERIMENTO 1

Nelle **tabelle 5.17a**, **5.17b** e **5.17c** sono presenti i valori delle predizioni del primo esperimento (dunque per il gruppo 4, classificato nelle le classi 1, 2 e 3), rispettivamente per la prima, la seconda e la terza finestra temporale. I vari campi "block at" identificano i 5 sottogruppi del test set.

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	182	198	204	195	216
group 2	48	33	27	36	15
group 3	1	0	0	0	0

Tabella 5.17a: Predizioni esperimento 1 sul gruppo 4, dati dal 10 al 15 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	96	89	86	76	59
group 2	105	112	124	122	152
group 3	30	30	21	33	20

Tabella 5.17b: Predizioni esperimento 1 sul gruppo 4, dati dal 10 al 23 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	24	29	15	24	11
group 2	151	148	162	154	188
group 3	56	54	54	53	32

Tabella 5.17c: Predizioni esperimento 1 sul gruppo 4, dati dal 10 al 30 ottobre

Per la prima fascia temporale, la maggior parte dei dati del gruppo 4 è classificata all'interno del primo gruppo, mentre il gruppo 3 non ha pressoché nessun ciclo predetto al suo interno. Il motivo di questa classificazione è probabilmente dovuto al fatto che nella prima settimana in esame questo gruppo presenta una scarsa quantità di dati, a differenza proprio del gruppo 1. Ed effettivamente i risultati cambiano parecchio nella seconda e soprattutto nella terza tabella: il gruppo che contiene la maggior parte di dati predetti diventa il secondo, a discapito del primo.

Per quanto riguarda l'analisi della silhouette, ci si aspetta che, per ogni fascia temporale, il gruppo con maggior degrado sia quello in cui sono stati aggiunti più dati. In **figura 5.11** è mostrato il grafico dell'andamento della silhouette per la prima finestra temporale, e nelle **figure 5.12** e **5.13** sono presenti gli andamenti della silhouette media per la seconda e terza finestra temporale.

Il degrado maggiore è visibile specialmente per il primo gruppo nel primo caso, ma anche per il secondo gruppo. La silhouette media mostra un progressivo abbassamento per tutte e tre le classi, soprattutto nella seconda finestra temporale per il secondo

gruppo, mentre per il terzo, sebbene non si tratti di un peggioramento netto e plateale, è possibile visualizzare un leggero trend decrescente.

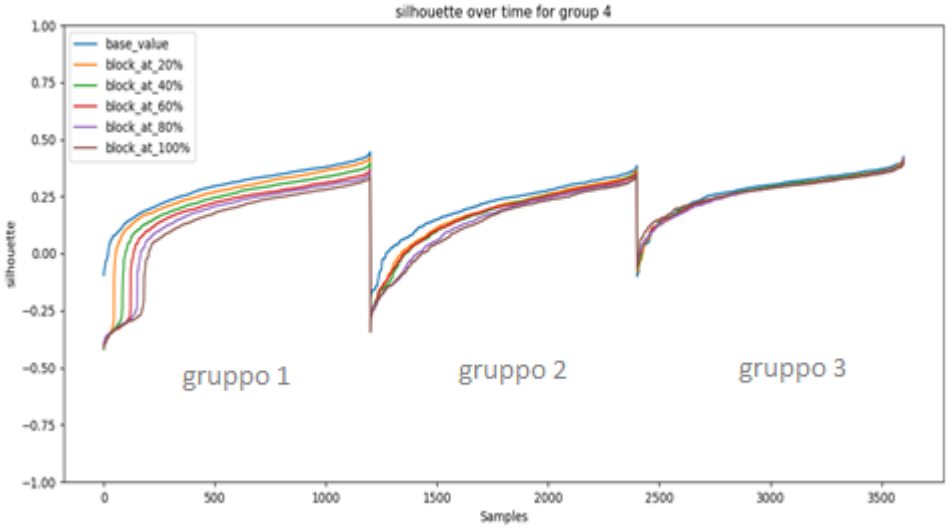

Figura 5.11: Andamento silhouette nell'esperimento 1, dati dal 10 al 15 ottobre

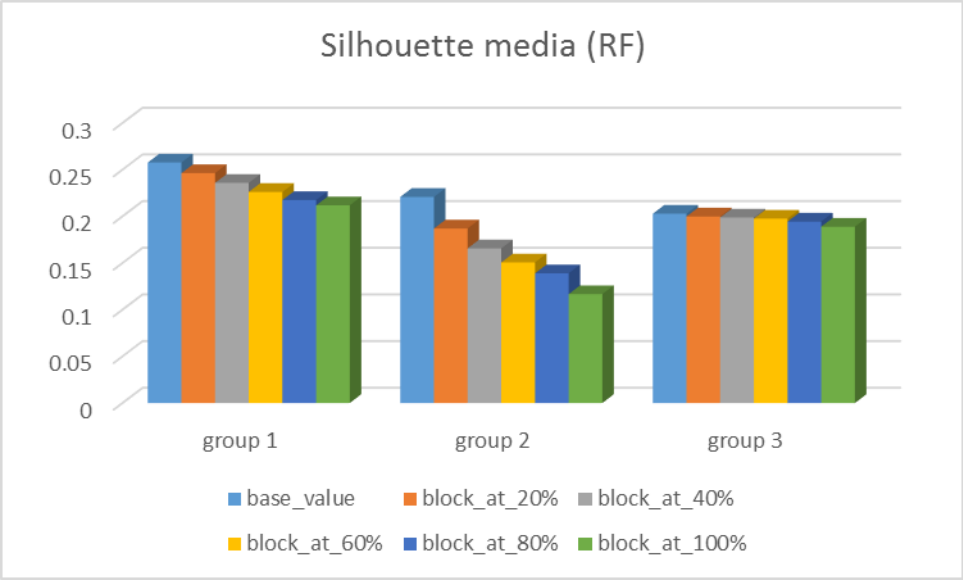

Figura 5.12: Andamento silhouette media nell'esperimento 1, dati dal 10 al 23 ottobre

In ogni caso, non è mai visibile un aumento della silhouette, né per quanto riguarda i valori singoli, né per i calcoli delle medie. Questo è un buon risultato, che permette di

osservare che il modello generato è in grado di riconoscere le differenze tra i gruppi e i diversi comportamenti che essi generano nella macchina.

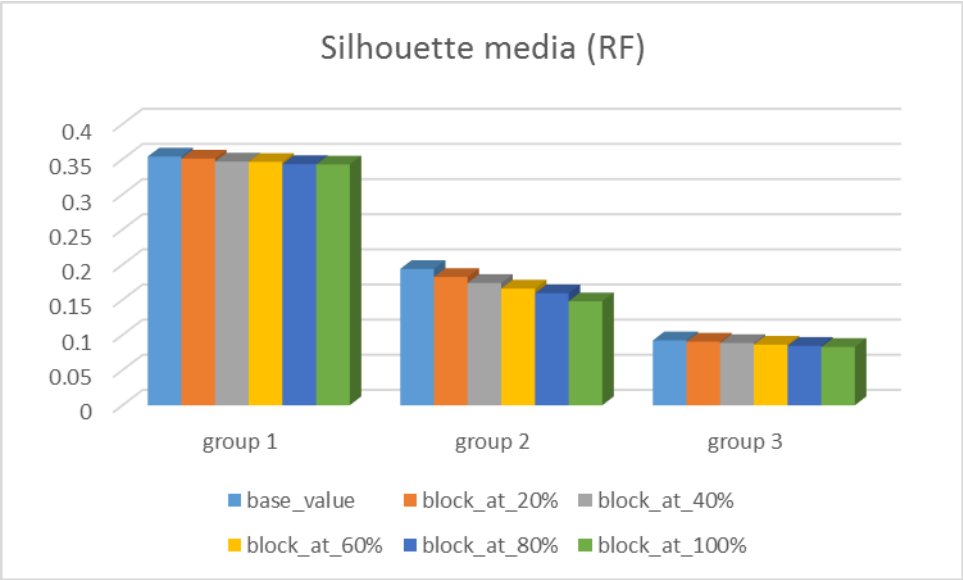

Figura 5.13: Andamento silhouette media nell'esperimento 1, dati dal 10 al 30 ottobre

Oltre alla silhouette, viene di seguito riportato il grafico relativo al calcolo dell'MSE (in **figura 5.14**) per questo esperimento.

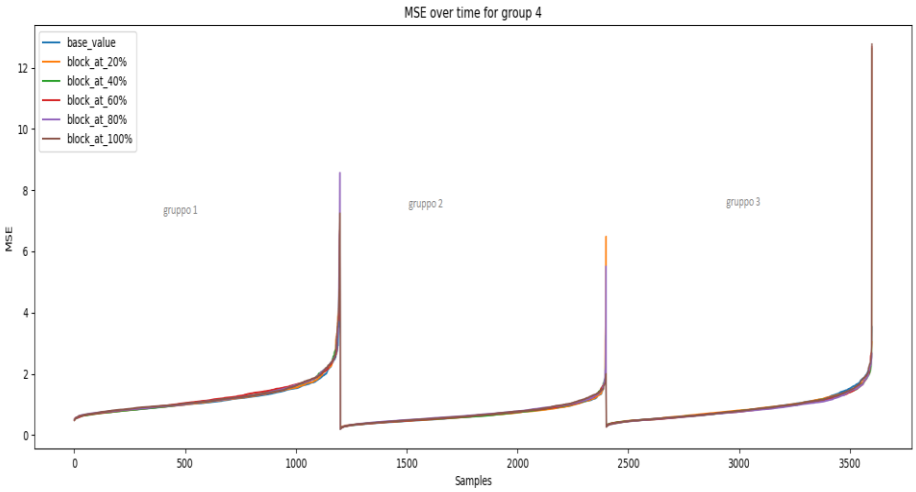

Figura 5.14: Andamento MSE nell'esperimento 1, dati dal 10 al 23 ottobre

Quest'ultimo grafico contiene 6 curve che rappresentano i valori assunti dall'indice calcolato per i gruppi 1, 2 e 3 a partire dal caso base e aggiungendo uno per uno i sottogruppi del gruppo 4. È facile osservare che si tratta di un grafico illeggibile e che le curve sono tutte sovrapposte tra loro. Generalmente per l'MSE sono sempre stati trovati risultati simili che però non verranno riportati in questa tesi, in quanto inutili ai fini della comprensione delle analisi.

ESPERIMENTO 2

Uno dei principali obiettivi di questo secondo esperimento è quello di esplorare l'adattamento dei classificatori rispetto ai due casi migliori in ambito industriale. Verranno presentati prima i risultati ottenuti con il gruppo 4 come test e in seguito quelli con il gruppo 3. Questi ultimi sono particolarmente importanti perché mostrano come i modelli siano in grado di riconoscere il diverso comportamento della macchina anche in condizioni di funzionamento circa standard (il gruppo 3 è anch'esso uno di quelli "interni" e ha al suo interno moltissimi dati). Gli elementi presenti nel gruppo 4 sono costanti in tutte e tre le fasce temporali mentre invece quelli del gruppo 3 aumentano progressivamente: questo è un fatto che senza dubbio influisce sui risultati ottenuti nelle predizioni e sullo scostamento visibile nei grafici della silhouette.

Nelle **tabelle 5.18a**, **5.18b** e **5.18c** ci sono i risultati delle predizioni del secondo esperimento con il gruppo 4 come test (anche in questo caso ogni tabella rappresenta una diversa fascia di tempo). I grafici riassuntivi dell'andamento della silhouette per questa casistica sono invece mostrati nelle **figure 5.15, 5.16** e **5.17**.

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	216	217	215	231	223
group 2	15	14	16	18	8

Tabella 5.18a: Predizioni esperimento 2 sul gruppo 4, dati dal 10 al 15 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	191	189	188	193	211
group 2	40	42	43	38	20

Tabella 5.18b: Predizioni esperimento 2 sul gruppo 4, dati dal 10 al 23 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	98	85	84	84	53
group 2	133	146	147	147	178

Tabella 5.18c: Predizioni esperimento 2 sul gruppo 4, dati dal 10 al 30 ottobre

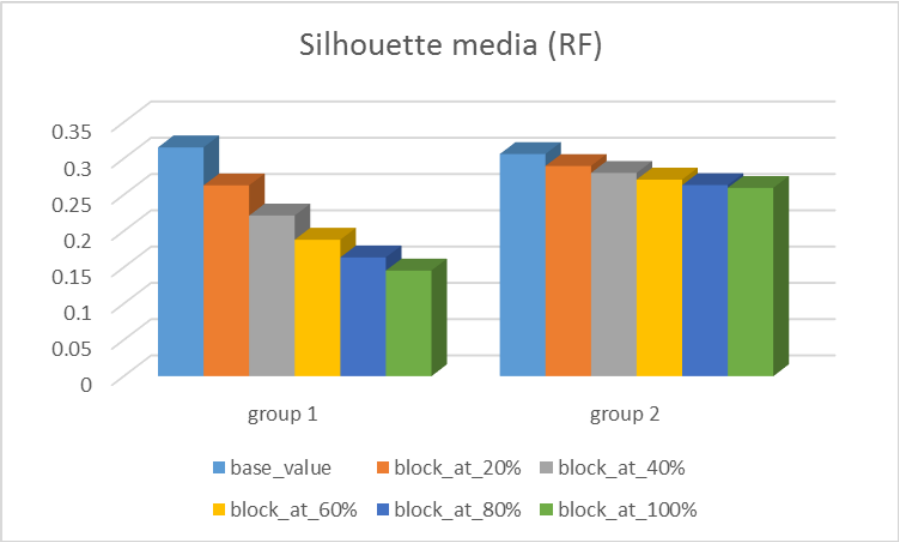

Figura 5.15: Andamento silhouette media nell'esperimento 2 sul gruppo 4, dati dal 10 al 15 ottobre

I risultati di questo secondo esperimento con il gruppo 4 sono molto simili a quelli precedenti, appena visti con l'esperimento 1. I dati vengono classificati in entrambi i gruppi e il degrado della silhouette è sempre abbastanza visibile, specialmente per quanto riguarda il gruppo 1.

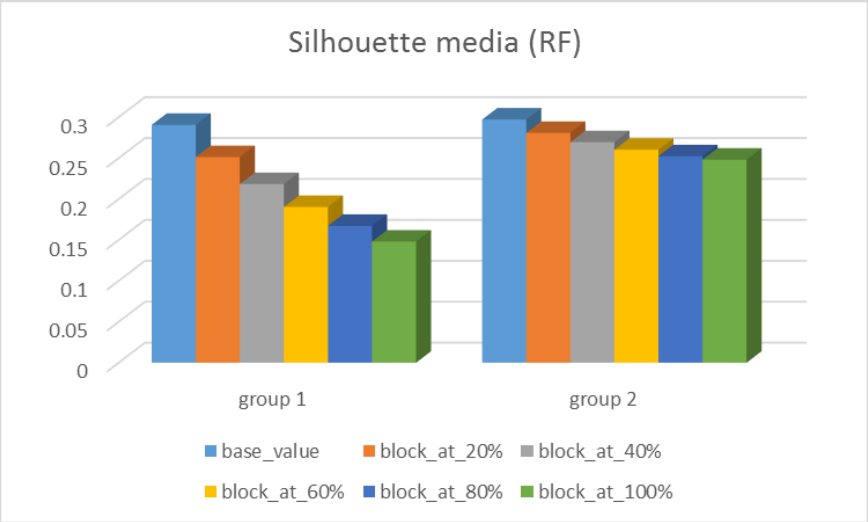

Figura 5.16: Andamento silhouette media nell'esperimento 2 sul gruppo 4, dati dal 10 al 23 ottobre

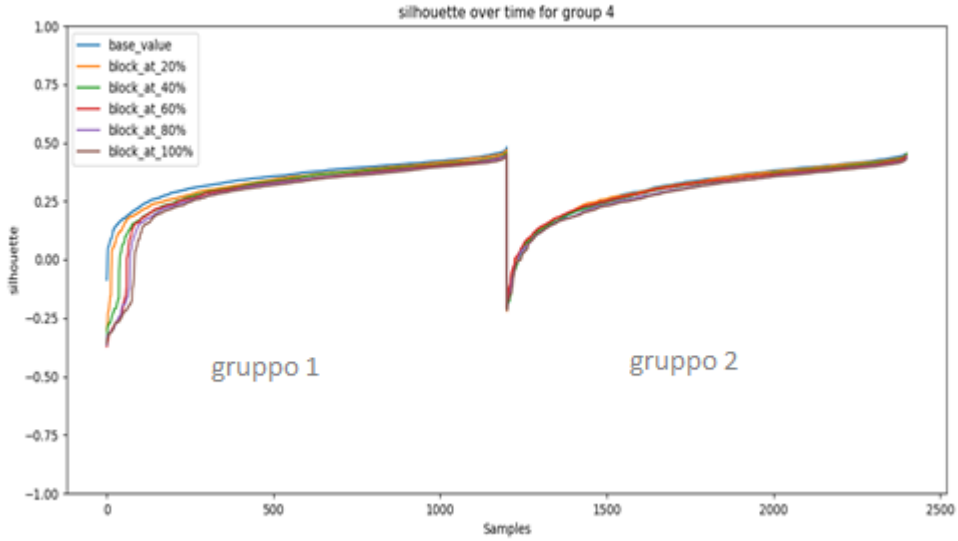

Figura 5.17: Andamento silhouette nell'esperimento 2 sul gruppo 4, dati dal 10 al 30 ottobre

Volendo approfondire una delle casistiche mostrate in questo esperimento 2 (con i dati delle prime due settimane presenti), dove si vede la presenza di un degrado ma non è semplicissimo capirne l'entità e l'importanza, viene di seguito riportato anche il grafico relativo al calcolo del MAAPE tra la curva base e tutte le altre (**figura 5.18**). Essendo il MAAPE il calcolo di scostamento tra due insiemi di numeri, i valori rappresentati sono solo 5 e indicano la distanza tra la curva di silhouette base e tutte quelle con l'aggiunta

dei sottogruppi. Si può notare un MAAPE sempre crescente per il gruppo 1 (sintomo della presenza di degrado, ovvero del fatto che il modello sa riconoscere un peggioramento nella macchina), mentre per il gruppo 2 il risultato è più incerto, forse anche dovuto alla minore numerosità delle predizioni al suo interno in questa finestra temporale.

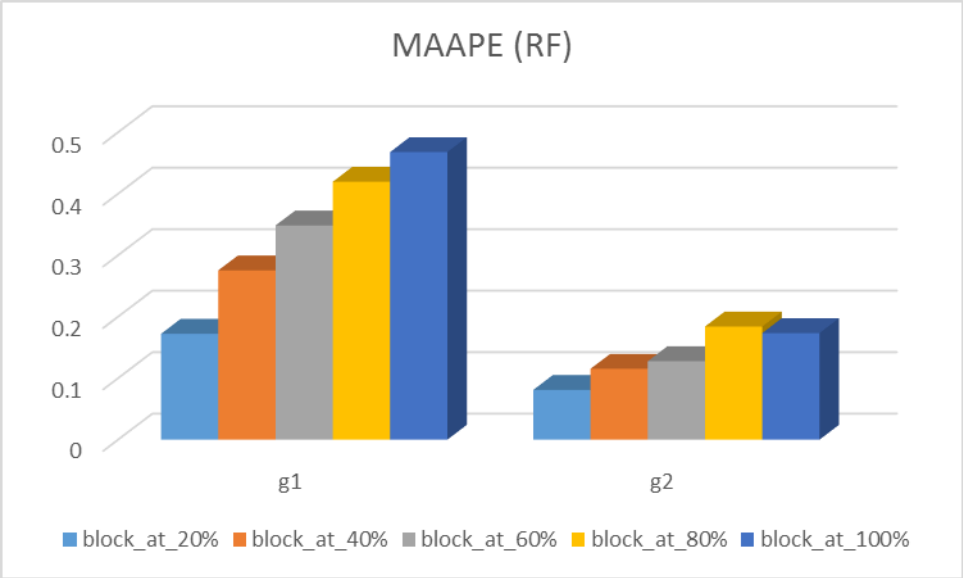

Figura 5.18: Andamento MAAPE nell'esperimento 2 sul gruppo 4, dati dal 10 al 23 ottobre

I risultati ottenuti utilizzando il terzo gruppo come test sono invece mostrati nelle **tabelle 5.19a, 5.19b** e **5.19c** e nelle **figure 5.19, 5.20** e **5.21**.

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	121	126	117	122	121
group 2	13	8	17	12	13

Tabella 5.19a: Predizioni esperimento 2 sul gruppo 3, dati dal 10 al 15 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	732	732	721	714	711
group 2	79	79	90	97	100

Tabella 5.19b: Predizioni esperimento 2 sul gruppo 3, dati dal 10 al 23 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 1	743	763	795	695	733
group 2	998	978	946	1046	1008

Tabella 5.19c: Predizioni esperimento 2 sul gruppo 3, dati dal 10 al 30 ottobre

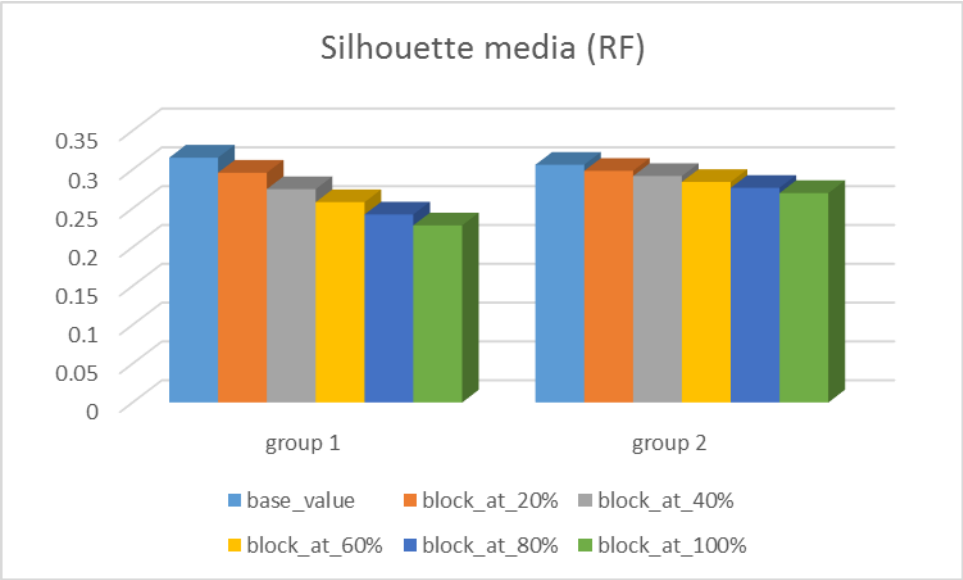

Figura 5.19: Andamento silhouette media nell'esperimento 2 sul gruppo 3, dati dal 10 al 15 ottobre

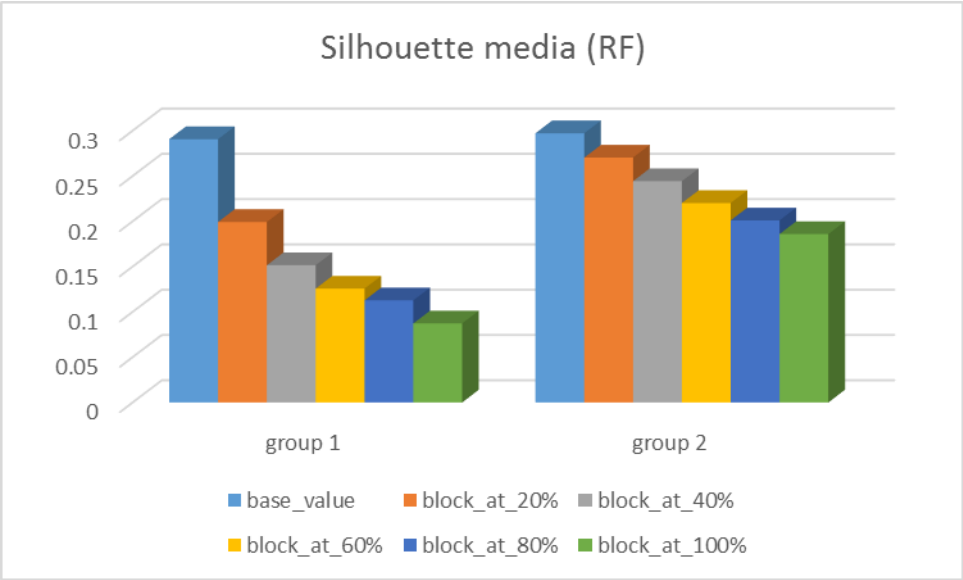

Figura 5.20: Andamento silhouette media nell'esperimento 2 sul gruppo 3, dati dal 10 al 23 ottobre

I dati del gruppo 3 vengono predetti principalmente nel gruppo 1 per i primi due casi, mentre nell'ultimo vengono assegnati in maggioranza al gruppo 2. È presente un forte degrado per quanto riguarda la silhouette e questo è dovuto oltre alla diminuzione di coesione tra i gruppi 1 e 2, soprattutto alla grande numerosità del gruppo 3 usato come test. Infatti tale degrado è visibile soprattutto nella **figura 5.21** in cui sono utilizzati tutti i dati a disposizione: si può notare che i valori raggiunti dalla silhouette sono decisamente bassi, e spesso inferiori allo zero.

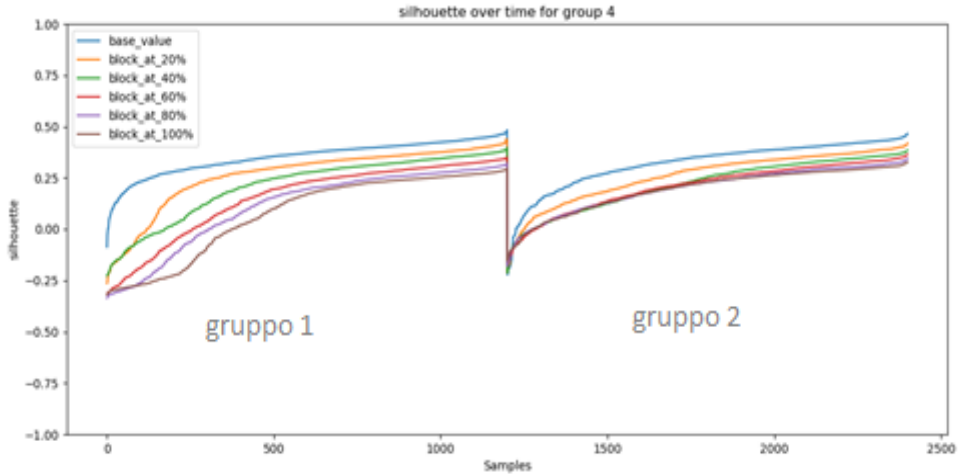

Figura 5.21: Andamento silhouette nell'esperimento 2 sul gruppo 3, dati dal 10 al 30 ottobre

ESPERIMENTO 3

Il terzo ed ultimo esperimento è stato effettuato per poter considerare come dataset di base i cicli del gruppo 2, situazione operativa standard, e quelli del gruppo 3, anch'essi relativi ad una normale situazione sebbene presentino già meno coesione interna e quindi possano rappresentare una sorta di inizio di un degrado. Il gruppo 4, come al solito, è servito come gruppo di test. I risultati ottenuti sono visibili nelle **tabelle 5.20a, 5.20b** e **5.20c,** oltre che nelle figure **5.22, 5.23** e **5.24**.

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 2	176	175	178	173	179
group 3	55	56	53	58	52

Tabella 5.20a: Predizioni esperimento 3, dati dal 10 al 15 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 2	62	66	69	59	59
group 3	169	165	162	172	172

Tabella 5.20b: Predizioni esperimento 3, dati dal 10 al 23 ottobre

RF	block_at_20%	block_at_40%	block_at_60%	block_at_80%	block_at_100%
group 2	93	75	85	84	71
group 3	138	156	146	147	160

Tabella 5.20c: Predizioni esperimento 3, dati dal 10 al 30 ottobre

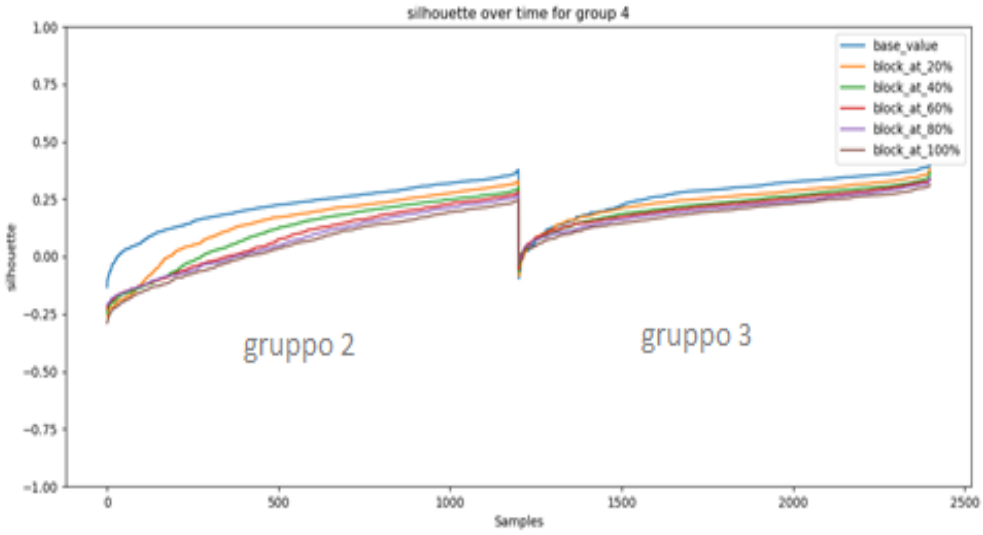

Figura 5.22: Andamento silhouette nell'esperimento 3, dati dal 10 al 15 ottobre

Tolte le predizioni della prima settimana, dove il gruppo 3 ha pochissimi cicli, generalmente il gruppo 4 viene assegnato di meno al gruppo 2 e di più al 3, come era prevedibile.

Per quanto riguarda la silhouette nelle prime due fasce temporali i due gruppi di train mostrano valori simili di silhouette: vi è inoltre per entrambi i gruppi un trend decrescente nella media di tale indice e un abbassamento delle curve a partire da quella base fino a quella con il 100% dei dati che sono stati aggiunti. La situazione cambia nel caso in cui l'intero dataset è utilizzato: il valore medio della silhouette del terzo gruppo subisce un abbassamento notevole già a partire dal valore base. Un piccolo trend decrescente è visibile specialmente per il gruppo 2.

In conclusione è possibile affermare che nel terzo esperimento il modello fa leggermente più fatica a evidenziare i cambiamenti nei segnali del robot forse anche per il fatto che un gruppo poco numeroso come il quarto non può influire più di tanto sul calcolo della silhouette di due gruppi ben più numerosi come il 2 e il 3.

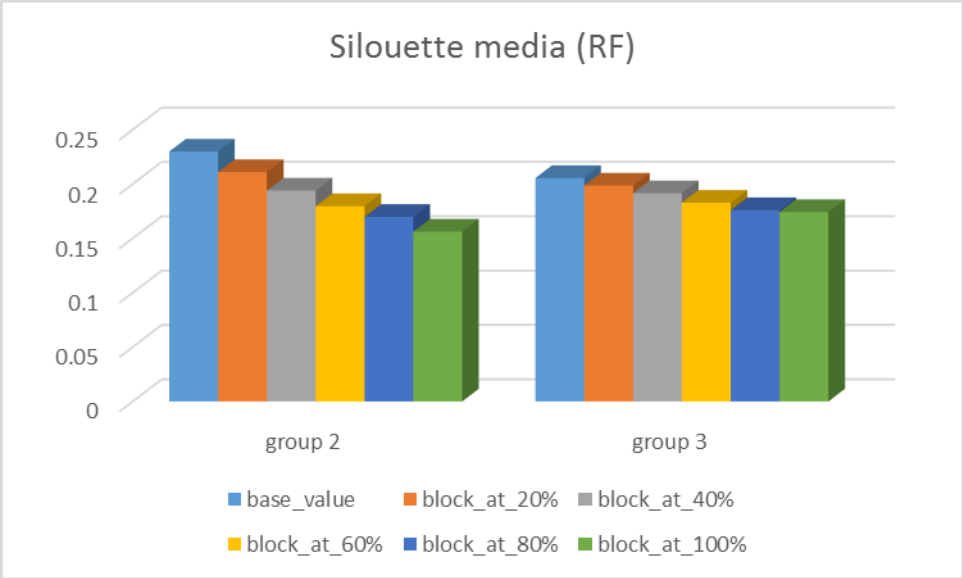

Figura 5.23: Andamento silhouette media nell'esperimento 3, dati dal 10 al 23 ottobre

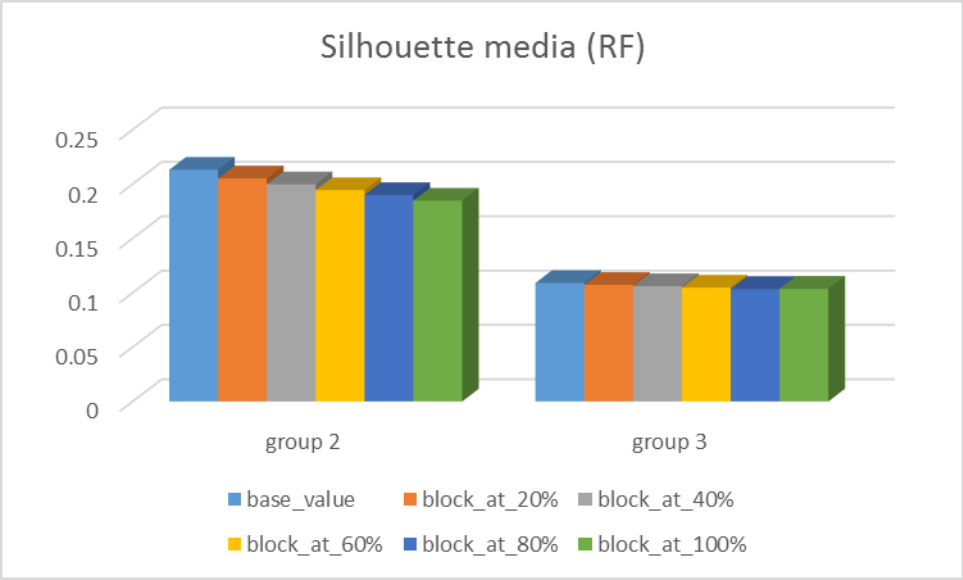

Figura 5.24: Andamento silhouette media nell'esperimento 3, dati dal 10 al 30 ottobre

6 – CONCLUSIONI

Durante i capitoli di questa tesi è stato descritto per intero il processo che ha portato allo sviluppo dei modelli, generale ed evolutivo, per la manutenzione predittiva del braccio robotico grazie all'analisi della corrente emessa durante i suoi cicli lavorativi. In questo capitolo finale saranno invece descritti i risultati raggiunti e i possibili sviluppi futuri di questo progetto che non sono stati ancora affrontati.

Per quanto riguarda il modello generale, si è notato che le prestazioni del classificatore usato (Random Forest) decrescono all'aumentare dei cicli presenti nel dataset: con 2381 record l'accuratezza ottenuta è pari a circa il 97% mentre con 23883 dati tale misura decresce di circa un 10%, a parità di proporzioni tra train e test del modello. Questa decrescita è quasi sicuramente causa del fatto che i dati subiscono nel tempo una piccola deviazione dovuta all'usura della macchina, ma per confermare quest'ipotesi sarebbe necessario poter studiare gli stessi dati in un intervallo di tempo maggiore (si ricorda che i dati a disposizione sono relativi ai giorni che vanno dal 10 al 30 ottobre 2018). Qualora si decidesse di non tenere in considerazione l'informazione temporale i risultati ottenuti sarebbero del tutto diversi: persino una piccola percentuale come il 10% dei dati totali sarebbe buona per effettuare predizioni sul resto dei cicli, a patto però che tale percentuale venga estratta in maniera casuale tra le classi. Ma in questo caso non è detto che il modello generato funzionerebbe ugualmente aggiungendo nuovi casi ancora non affrontati durante il suo addestramento. Siccome uno dei principali obiettivi del modello generale è quello di ottenere un buon classificatore in grado di riconoscere le differenze tra i gruppi prendendo i dati ordinati per tempo, si può affermare che utilizzare il 70% dei dati per il train e il 30% per il test porti a risultati soddisfacenti, se i nuovi dati, su cui verrà utilizzato il modello vengono aggiunti pochi per volta (un po' come è stato fatto con i modelli con l'Isolation Forest, o con gli ultimi 10 esperimenti del Random Forest) in modo da valutare se il modello è ancora adatto, se c'è bisogno di un intervento per aggiustare nella macchina e se il modello è da riaddestrare.

Per ottimizzare un processo del genere potrebbe essere utile integrare al modello generale quello evolutivo, che ha sviluppato esattamente la capacità di riconoscere

quando i dati stanno modificando il loro comportamento. Infatti i risultati raggiunti da questo modello permettono di affermare che il classificatore progettato è generalmente in grado di distinguere le differenze tra i vari gruppi tramite il calcolo periodico dell'indice di coesione (la silhouette infatti va calcolata spesso, e a questo scopo è servita la divisione del gruppo di test in 5 sottogruppi, per osservare pian piano il degrado della macchina). Purtroppo le prestazioni del modello evolutivo non sono state testate con casi reali, nel senso che i dati di test non riguardavano un malfunzionamento vero e proprio ma semplicemente un comportamento diverso da quello a cui il classificatore era abituato. La reazione di quest'ultimo a casistiche mai incontrate precedentemente è stata quella di inserire i nuovi dati nella classe più vicina ad essi, portando però sempre una diminuzione della coesione dei gruppi. Anche in questo caso sarebbe stato preferibile testare i suoi risultati su una finestra temporale maggiore o con dati relativi ad un malfunzionamento realmente avvenuto durante la produzione.

Infatti, i principali possibili miglioramenti e sviluppi futuri di questo progetto, riguardano il train e il test di modelli in cui sono presenti dati relativi a fallimenti della macchina. Per quanto i due modelli costruiti possano essere in grado di segnalare la necessità del ritensionamento della cinghia dopo aver osservato un cambiamento nei dati, le loro prestazioni sarebbero decisamente migliori se a priori già conoscessero quali sono i dati buoni e quali derivanti da un problema del robot. Se si avessero a disposizione queste informazioni sarebbe anche possibile effettuare una *Survival Analysis*, un metodo statistico per stimare la durata temporale della macchina prima che avvenga un determinato evento. Nell'applicazione in esame, questo evento sarebbe rappresentato dal momento in cui la cinghia deve essere sostituita o tensionata nel modo corretto. Un'analisi di questo tipo consente di calcolare anche altre informazioni di interesse come ad esempio la probabilità di "sopravvivenza" del macchinario in relazione alle ore di lavoro. Purtroppo però, come già affermato in precedenza, bisognerebbe raccogliere molti più dati di corrente e attendere che la macchina si guasti per sfruttare queste informazioni.

Altre possibili analisi che non è stato possibile effettuare durante questo lavoro riguardano, ad esempio, la relazione che intercorre tra la corrente emessa e altre variabili esterne o interne alla macchina durante il suo ciclo. Una di esse può

sicuramente essere la temperatura: essa, come già parzialmente osservato, incide particolarmente all'inizio della giornata lavorativa (o nel momento in cui viene acceso il motore, ancora freddo, che aziona i movimenti). Potrebbe essere interessante indagare se questa variabile porti anche altri effetti sul funzionamento della macchina. Anche mettere in relazione la corrente alla posizione del braccio robotico, oltre che alla sua velocità di movimento potrebbe essere uno studio da approfondire.

BIBLIOGRAFIA E SITOGRAFIA

Industry 4.0:

- [1] Almada-Lobo F., *"The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES)",* Journal of Innovation Management JIM 3-4, 16-21, 2015
- [2] Hermann M., Pentek T., Otto B., "*Design principles for industrie 4.0 scenarios",* 49th Hawaii International Conference on System Sciences (HICSS), 2016
- [3] Gilchrist A., "*Industry 4.0: The Industrial Internet of Things",* Apress, 2016
- [4] https://www.internet4things.it/industry-4-0/big-data-e-data-science-nellindustry-4-0-vogliono-infrastrutture-e-storage-focalizzate-sul-dato/
- [5] https://www.internet4things.it/iot-library/che-cose-il-3d-printing-e-come-si-colloca-nellambito-industry-4-0-e-iot/
- [6] http://www.bandwich.it/2017/11/16/cyber-physical-system-industry-4-0/
- [7] https://www.economyup.it/innovazione/cos-e-l-industria-40-e-perche-e-importante-saperla-affrontare/
- [8] https://www.agendadigitale.eu/industry-4-0/industry-40-tutto-quello-che-c-e-da-sapere-su-piano-e-attuazione/

Data analytics & machine learning:

- [9] https://searchdatamanagement.techtarget.com/definition/data-analytics
- [10] https://www.zerounoweb.it/analytics/big-data/come-fare-big-data-analysis-e-ottenere-valore-per-le-aziende/
- [11] O'Neil C., Schutt R., *"Doing Data Science: Straight Talk from the Frontline",* O'Reilly Media Inc, 2013
- [12] https://www.bigdata4innovation.it/big-data/big-data-analytics-in-italia/
- [13] https://www.simplilearn.com/data-science-vs-data-analytics-vs-machine-learning-article

- [14] https://www.hannovermesse.de/en/news/key-topics/artificial-intelligence/
- [15] https://www.sas.com/it_it/insights/analytics/machine-learning.html
- [16] https://www.ai4business.it/intelligenza-artificiale/deep-learning/reti-neurali/

Manutenzione predittiva e stato dell'arte:

- [17] https://www.sysman.it/processi/iot-predictive-maintenance.html
- [18] Mobley R. K., *"An introduction to predictive maintenance",* Butterworth-Heinemann, 2002
- [19] https://www.toolsforsmartminds.com/it/insight/blog/178-manutenzione-predittiva-tipi-di-manutenzioni-a-confronto-e-vantaggi
- [20] https://www.internet4things.it/industry-4-0/verso-la-vera-predictive-maintenance-come-fondere-iot-e-machine-learning/
- [21] http://meccanica-plus.it/volare-in-sicurezza-con-easyjet_73246/
- [22] Garcia M. C., Sanz-Bobi M. A., Del Pico J., *"SIMAP: intelligent system for predictive maintenance: application to the health condition monitoring of a windturbine gearbox"*, Computers in Industry, volume 57, issue 6, 552-568, 2006
- [23] Li H., Parikh D., He Q., Qian B., Li Z., Fang D., Hampapur A., *"Improving rail network velocity: a machine learning approach to predictive maintenance"*, Transportation research part C: emerging technologies, volume 45, 17-26, 2014
- [24] https://www.digital4.biz/supply-chain/visibility-e-collaboration/trenitalia-con-big-data-e-iot-la-manutenzione-diventa-predittiva-un-passo-cruciale-della/

Belt tensioning:

- [25] https://www.ilprogettistaindustriale.it/cinghie-di-trasmissione/
- [26] https://www.gatestechzone.com/it/diagnostica-dei-problemi/sistema-di-tramissione-a-cinghia-sincrona
- [27] https://www.ibtinc.com/how-to-tension-a-v-belt/

Strumenti utilizzati:

- [28] https://www.mongodb.com/it
- [29] https://www.python.org/
- [30] https://api.mongodb.com/python/current/
- [31] https://pandas.pydata.org/
- [32] https://matplotlib.org/
- [33] https://seaborn.pydata.org/
- [34] https://scikit-learn.org/stable/