POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Gestionale

Progettazione e sviluppo di una metodologia data-driven per l'identificazione della deriva dei dati. Caso di studio: outlier detection nell'industria 4.0

Relatore

Prof. Tania CERQUITELLI

Candidato

Angelica Rita GIORDANO

Correlatori

Dott. Riccardo CALLA'

Dott. Paolo BETHAZ

Luglio 2020

Ai miei genitori

"Alle persone che guardano le stelle
ed esprimono desideri.
Alle stelle che ascoltano,
e ai sogni che si avverano"

Indice

Capitolo 1

Introduzione

Il paradigma industriale nel corso del tempo è stato caratterizzato da una serie di trasformazioni che hanno cambiato profondamente le strutture produttive e sociali grazie all'affermazione di nuove tecnologie. Il passaggio da una rivoluzione industriale all'altra è stato graduale e migliorativo interessando sempre più settori manifatturieri. La quarta rivoluzione industriale ha avuto un impatto fortissimo su tutti gli aspetti della vita produttiva. Non è stata necessaria una massiva sostituzione di macchinari e impianti già esistenti, ma questi ultimi sono stati "aggiornati" e dotati di sensori, connettività di rete, spazio di archiviazione e potere computazionale che ha permesso lo sviluppo dell'Internet of Things. La fusione del mondo fisico con quello virtuale è resa possibile grazie alle 9 tecnologie abilitanti che hanno migliorato quelle già esistenti. In questo contesto di "smart manufacturing" assume un ruolo molto importante l'uso dei big data e delle tecniche di machine learning per le attività di manutenzione predittiva. Il machine learning ha trasformato radicalmente il settore manifatturiero con una grande varietà di algoritmi, teorie e metodi. È importante fare delle previsioni sulle condizioni di salute dei macchinari, sulla vita utile rimanente e sui guasti per evitare interruzioni improvvise e, di conseguenza, grosse perdite a livello aziendale. I sensori permettono di monitorare le informazioni in tempo reale sullo stato di salute dei macchinari. L'integrazione tra dati storici e dati generati in tempo reale aiuta a simulare e ottimizzare programmi di manutenzione.

Nell'ambito dell'analisi predittiva e dell'apprendimento automatico, con il termine "concept drift" (letteralmente: deriva del concetto) si indica il fenomeno

in cui le proprietà statistiche dei dati appresi da un modello di machine learning mutano nel tempo in modi imprevisti, creando un problema nelle previsioni che diventano sempre meno precise. Nel contesto della manutenzione predittiva ci si basa spesso su tecniche supervisionate. Tra queste gli algoritmi più diffusi sono quelli di classificazione, spesso difficili da valutare nel tempo. Le metriche e gli indici più usati, come precisione, richiamo e accuratezza richiedono la presenza delle etichette di classe reali per poter valutare i modelli. Con l'ingresso di nuovi dati e il rischio di concept drift non sempre ciò è possibile, dunque per superare questi problemi nascono dei nuovi strumenti di autovalutazione che sono in grado di rilevare automaticamente quando l'adeguatezza di un modello degrada troppo per i dati analizzati. Questi nuovi metodi sfruttano indici di coesione intra-classe e di separazione inter-classe in grado di quantificare il degrado del modello di classificazione all'ingresso di nuovi dati nel sistema.

L'obiettivo di questa tesi è quello di proporre una metodologia data-driven per il rilevamento di tali variazioni dei dati e del degrado del modello adottato. A partire dallo stato dell'arte, si è cercato di riadattare le tecniche già esistenti ad un caso di studio reale, proponendo un'alternativa che potesse superarne i limiti e che potesse sfruttare un processo di outlier detection. Per fare ciò sono stati analizzati i dati provenienti da un'industria specializzata in processi di automazione e, su di essi, sono stati eseguiti gli esperimenti per la metodologia proposta.

Prima di iniziare la fase di progettazione e sviluppo, nel *Capitolo 2* è stata fatta una panoramica sul contesto dell'Industria 4.0 in cui si colloca il caso di studio. Si è parlato in breve della storia delle quattro rivoluzioni industriali analizzandone gli elementi chiave. Relativamente all'Industria 4.0 si è posta l'attenzione sulle tecnologie abilitanti e in particolare sui Big data e il Machine Learning. Questi hanno permesso la crescita e la trasformazione industriale ma anche uno sviluppo nelle attività di manutenzione degli impianti. Nel *Capitolo 3* si è discusso in generale del ruolo del data mining e del machine learning, esponendo poi nel dettaglio una pipeline di base per il processo di estrazione della conoscenza dai dati. È stato poi fatto un excursus sulle principali tecniche di data mining come la classificazione, il clustering e le regole di associazione analizzando poi in breve le varie tipologie di algoritmi esistenti. Nel *Capitolo 4* si è parlato del Concept drift, delle metodologie presenti allo stato dell'arte discutendo dei vantaggi e degli svantaggi che esse presentano nei vari casi di studio. È stata poi proposta una

metodologia alternativa, a partire da un processo di Anomaly detection, molto importante nell'ambito degli stream di dati provenienti dai sensori. Sono state presentate le sfide attuali dell'anomaly detection e le tecniche più usate mostrando più nel dettaglio il processo di rilevamento del drift in un contesto di outlier detection.

Nel *Capitolo 5* sono stati esposti i risultati sperimentali. Dopo una prima fase di analisi del caso di studio e dei dataset utilizzati, è stata eseguita la validazione e l'analisi del modello ed è stata poi applicata la metodologia esposta. Alla fine del capitolo è stato fatto un confronto dei risultati con quelli ottenuti applicando quanto proposto nello stato dell'arte e infine ne sono state discusse le limitazioni.

Nel *Capitolo 6* infine sono stati riassunti e discussi i risultati ottenuti in questo studio, del quale sono stati poi proposti dei possibili sviluppi futuri.

Capitolo 2

Contesto: Industria 4.0

La crisi finanziaria ed economica del 2008 ha accentuato la necessità per le aziende di ridefinire il proprio modello di business con una grossa spinta verso l'innovazione strategica. È in questo contesto che prende piede l'"Industria 4.0". Con questo termine si identifica la quarta rivoluzione industriale, una profonda e irreversibile trasformazione digitale del sistema produttivo di quello socio-economico.

La nozione di "Industria 4.0" ha stimolato un crescente dibattito nella società tedesca. Inizialmente esso era relativo alle nuove opzioni tecnologiche nell'ambito della manifattura ma ben presto ha iniziato a diffondersi in molti altri ambiti della società.

Il termine Industria 4.0 è stato usato per la prima volta nel 2011 alla Fiera di Hannover, in Germania, come ipotesi di progetto da cui è partito un gruppo di lavoro che nel 2012 ha presentato al governo tedesco l'implementazione del Piano Industria 4.0. L'8 aprile del 2013, sempre ad Hannover, è stato diffuso il report finale sugli investimenti necessari per ammodernare il sistema produttivo tedesco e riportare la manifattura tedesca ai vertici mondiali rendendola competitiva a livello globale. Questo modello è poi diventato una fonte di ispirazione per tutti gli altri Paesi. [1]

Le ricerche tedesche si sono focalizzate sulle tecnologie per il settore manifatturiero con sensori intelligenti, reti di sensori wireless e CPSs. La piattaforma cloud creata da Siemens, Sinalytics, è un esempio. Essa fornisce una comunicazione sicura e l'analisi di un'elevata quantità di dati generati dai macchinari per il miglioramento

del monitoraggio e l'ottimizzazione delle capacità delle strutture attraverso i dati e i feedback. È sulla strada della Germania che l'Unione Europea ha lanciato il suo più grande programma di ricerca e innovazione, Horizon 2020 con dei fondi disponibili per gli anni dal 2014 al 2020. Nell'ambito di Horizon 2020, il nuovo partenariato pubblico-privato contrattuale (PPP) sulle fabbriche del futuro (FoF) si baserà sui successi del 7° programma quadro di ricerca e sviluppo tecnologico dell'Unione europea (7PQ 2007-2013), PPP FoF. La tabella di marcia pluriennale FoF per gli anni dal 2014 al 2020 stabilisce una visione e delinea le rotte verso tecnologie di produzione ad alto valore aggiunto per le fabbriche del futuro, che saranno pulite, altamente performanti, rispettose dell'ambiente e socialmente sostenibili. Queste priorità sono state concordate all'interno della vasta comunità di parti interessate in Europa, dopo un'ampia consultazione pubblica. [2] La commissione europea ha lanciato una piattaforma comune per la condivisione delle migliori prassi e, se possibile, per mettere in sinergia grandi progetti. Incrociando le analisi dell'associazione Adapt, Politecnico di Milano ed I-Com, emerge innanzitutto il grado di evoluzione del caso tedesco nell'ambito tecnologico. L'Italia invece è tra i Paesi guida per il sostegno fiscale alle imprese. [3]

2.1 Le rivoluzioni industriali

Per comprendere fino in fondo la potenzialità dell'Industria 4.0 può essere importante capire e fare un confronto con le precedenti rivoluzioni industriali. Un motto dell'Institute for the Future è "looking backward to look forward", cioè guardare al passato traendo insegnamenti per il presente e comprendendo i cambiamenti in atto. Ciascuna delle rivoluzioni industriali che si sono susseguite nel tempo sono state caratterizzate da profonde trasformazioni delle strutture produttive e sociali con l'affermazione di nuove tecnologie. [4] La prima rivoluzione industriale iniziò nel 1760 e interessò principalmente i settori tessile, metallurgico ed estrattivo. Il sistema agricolo-artigiano-commerciale divenne industriale. Una delle più importanti innovazioni fu il motore a vapore di Watt che trainò l'intera rivoluzione industriale. Lo sfruttamento di queste innovazioni fu reso possibile dalla nascita di nuove fabbriche, caratterizzate da un'integrazione verticale delle fasi produttive e un'organizzazione burocratica per aumentare la produttività. L'evoluzione tecnologica graduale degli anni successivi ha poi portato alla Seconda Rivoluzione Industriale iniziata nel

1870. Essa interessò il settore elettrico e chimico-petrolifero. Fondamentali sono stati gli studi nei laboratori universitari e non che portarono all'invenzione del motore a scoppio alimentato a petrolio e dell'elettricità che permise la creazione della prima lampadina da parte di Edison. La creazione di processi standardizzati all'interno della fabbrica ha portato alla creazione della catena di montaggio di Ford che ha contribuito in maniera determinante allo sfruttamento di queste innovazioni tecnologiche. La terza rivoluzione industriale partita nel 1970 ha interessato tutti i settori manifatturieri. La nascita dell'Information & Comunication Technology che riunisce i settori dell'elettronica, informatica e telecomunicazioni ha cambiato radicalmente i sistemi di produzione rendendoli più automatizzati e meno dipendenti dalla manodopera diretta. Il nuovo modello di business è stato caratterizzato da una produzione sempre più flessibile, fornitori e clienti sono stati maggiormente coinvolti nei processi di ricerca e sviluppo per dar vita ad una produzione sempre più personalizzata e differenziata. La quarta rivoluzione industriale ha coinvolto tutti i settori manifatturieri. L'impatto si ha sia sul sistema produttivo che su quello socio-economico. I macchinari e gli impianti di produzione non sono massivamente sostituiti ma saranno "aggiornati" e dotati di maggiori sensori e connettività di rete, spazio di archiviazione e potere computazionale che permette lo sviluppo dell'Internet of Things. La fabbrica è resa più "intelligente" e i sistemi fisici e virtuali sono integrati a livello di value-chain, value-system e value ecosystem, considerando l'intero ciclo di vita del prodotto. [4]

2.2 Industria 4.0: le iniziative in Europa

2.2.1 Piano Nazionale Industria 4.0

In Italia il Piano Nazionale Industria 4.0-2017-2020 è stato presentato il 21 settembre 2016 dal Ministro dello Sviluppo Economico Calenda ed è stato volto a favorire gli investimenti per l'innovazione e la competitività. Gli obiettivi principali del Piano sono legati alle seguenti necessità:

1. Incentivare le imprese che investono in beni strumentali nuovi, in beni materiali e immateriali (software e sistemi IT) funzionali alla trasformazione tecnologica e digitale dei sistemi produttivi;

2. Sostenere le imprese che richiedono finanziamenti bancari per investimenti in nuovi strumenti a uso produttivo o tecnologie digitali;

3. Stimolare la spesa privata in Ricerca e Sviluppo per rinnovare processi e prodotti e garantire la competitività futura delle imprese, incentivare la collocazione in Italia di beni immateriali detenuti all'estero da imprese italiane o estere e al contempo incentivare il mantenimento dei beni immateriali in Italia;

4. Sostenere le imprese innovative in tutte le fasi del loro ciclo di vita e diffondere una nuova cultura imprenditoriale votata alla collaborazione, all'innovazione e all'internalizzazione;

I provvedimenti investono tutti gli aspetti del ciclo di vita delle imprese le quali hanno acquisito la possibilità di aumentare la propria competitività attraverso i supporti messi a disposizione dal piano Industria 4.0. [5]

2.2.2 Politiche europee su Industria 4.0

La quarta rivoluzione industriale ha permesso una grande crescita nell'economia europea. La grande opportunità risiede nella trasformazione delle industrie già esistenti piuttosto che la nascita di nuove. Il tasso di adozione delle nuove tecnologie in Europa è ancora basso, più del 41% delle imprese non hanno ancora investito sull'Industria 4.0, questo mostra che è ancora una sfida per le aziende mettere in atto la trasformazione richiesta. Tuttavia, i dati di un sondaggio mostrano che il 75% delle imprese crede nell'opportunità delle nuove tecnologie e circa il 64% le ha già adottate. Per affrontare la sfida richiesta, molti governi europei hanno adottato delle nuove politiche per accrescere la produttività e la competitività e migliorare le skill necessarie per far fruttare al meglio l'alta tecnologia. Queste politiche, unite dall'obiettivo comune, si differenziano in termini di progetti, approcci e implementazione strategica. Purtroppo, nonostante la consapevolezza dell'importanza della cooperazione, è mancante un sostengo reciproco tra le nazioni. In alcuni Paesi, l'iniziativa politica è un diretto risultato di un quadro nazionale globale. Ad esempio, l'iniziativa tedesca "Industrie 4.0" è iniziata come parte dei 10 progetti sotto il nome di "Action Plan High-Tech Strategy 2020". Nel caso della Spagna la parte digitale dell'Agenda per il rafforzamento del settore industriale

si è gradualmente trasformata in "Industria Conectada 4.0". Invece "High-Value Manufacturing Catapult" nel Regno Unito mostra come il governo abbia agito attraverso delle raccomandazioni per fondare dei centri tecnologici in diversi settori. Ogni politica possiede degli elementi che la rendono unica rispetto alle altre. Le iniziative di Francia e Spagna hanno un approccio basato sul mercato fornendo un prestito alle imprese che partecipano al programma. La Svezia con il P2030 ha fornito enormi finanziamenti garantendo una sostenibilità a lungo termine, invece il Regno Unito ha investito molto sulla ricerca e sull'innovazione. Le politiche nazionali approvate sono relative ai fondi pubblici, ma è altrettanto importante la complementarietà degli investimenti privati. Anche in questo caso le iniziative adottate variano in termini di tipologia di azione. L'"Industrie du Futur" francese e "HVMC" inglese hanno messo in atto misure più complete. La prima ha fornito degli incentivi su Ricerca & Sviluppo, la seconda invece un impegno strategico con partner industriali chiave e supporti dedicati alle piccole e medie imprese. [6]

2.3 Industria 4.0: i concetti chiave

L'industria 4.0 è un'iniziativa tedesca strategica volta a creare fabbriche intelligenti dove le tecnologie nel settore manifatturiero sono migliorate e trasformate attraverso cyber-physical systems, l'Internet of Things e cloud computing. I sistemi manifatturieri sono in grado di monitorare i processi fisici, creare i cosidetti "digital twin" del mondo fisico e prendere decisioni tramite la comunicazione in tempo reale tra umani, macchine e sensori. Rispetto alle rivoluzioni precedenti, l'attuale si caratterizza per la possibilità di ottimizzare l'impiego delle risorse materiali, partendo da un miglior sfruttamento di quelle digitali che rendono intelligenti sia i prodotti che i processi. Molte nuove tecnologie dirompenti come il cloud computing, Internet of Things, big data analytics e intelligenza artificiale hanno iniziato a prendere un posto sempre più rilevante. Queste hanno permesso di rendere l'industria manifatturiera più intelligente e anche in grado di intraprendere nuove sfide come l'aumento delle personalizzazioni, il miglioramento della qualità e la riduzione del time to market (tempo che intercorre tra la creazione di un prodotto e la sua messa sul mercato). [7] La fusione del mondo fisico con quello virtuale è resa possibile dall'avvento delle 9 tecnologie abilitanti, che apportano miglioramenti alle tecnologie già presenti nelle rivoluzioni industriali precedenti.

La trasformazione digitale del manifatturiero, abilitata da queste tecnologie, ha permesso alle industrie di rivedere l'intero ciclo di vita del prodotto.

2.3.1 I nove pilastri fondamentali

I 9 pilastri tecnologici sono emersi da uno studio condotto da Boston Consulting dove si afferma che la quarta rivoluzione industriale si centra sull'adozione di 9 progressi tecnologici fondamentali. Durante questa trasformazione, sensori, macchine, pezzi e sistemi IT sono connessi attraverso la catena del valore. La maggior parte delle tecnologie abilitanti erano già state usate in ambito manifatturiero, ma con l'avvento dell'Industria 4.0 esse hanno dato vita ad un cambiamento radicale nella produzione. I singoli elementi di un'industria vengono integrati, automatizzati e ottimizzati con l'obiettivo di aumentare l'efficienza e cambiare le relazioni tra fornitori, produttori e clienti, oltre a quelle uomo-macchina.[8]

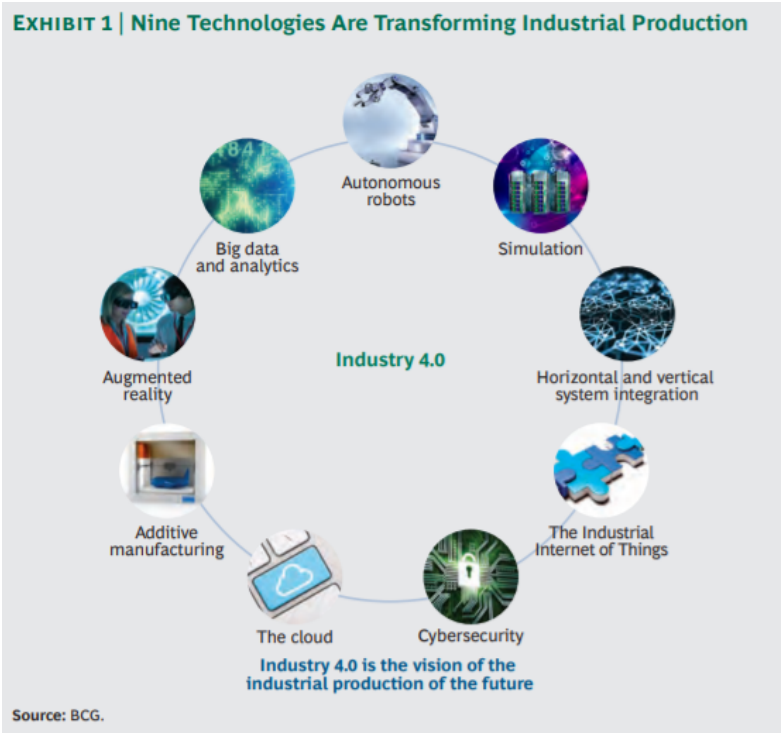


Figura 2.1: I nove pilastri tecnologici [8]

Big data and analytics

Gli studi analitici basati su dataset con volumi elevati sono diventati sempre più presenti nel mondo manifatturiero con l'obiettivo di ottimizzare la qualità della produzione, limitare il consumo energetico e migliorare i macchinari. In un contesto Industria 4.0, la raccolta dei dati da fonti eterogenee e la loro valutazione sono alla base del processo decisionale.

Autonomous Robots

I produttori hanno usato a lungo i robot per poter affrontare i compiti più complessi, ma questi adesso si stanno evolvendo per avere un'utilità maggiore. Stanno diventando più autonomi, flessibili e cooperativi. Essi interagiscono tra loro, lavorano affiancando l'uomo, costano meno e hanno maggiori capacità.

Simulation

In ambito ingegneristico, la simulazione in 3D di prodotti, materiali e processi produttivi era già presente ma adesso essa può evolversi ed essere usata in ogni attività industriale. Queste simulazioni fanno leva sui dati in tempo reale che rispecchiano il mondo fisico in un modello virtuale che include macchine, prodotti e umani. Questo permette di testare e ottimizzare le impostazioni dei macchinari sul mondo virtuale in modo da far diminuire i tempi di setup e aumentare la qualità.

Integrazione sistemica verticale e orizzontale

Molti sistemi IT non sono pienamente integrati. Le aziende, i fornitori e i clienti raramente sono strettamente collegati. L'industria 4.0 richiede una maggiore integrazione dei processi lungo la catena del valore, essa può essere orizzontale o verticale rendendo tutti i reparti e le funzioni aziendali parte di un unico sistema integrato. L'integrazione orizzontale riguarda gli agenti coinvolti nella catena del valore come partner commerciali o clienti. Quella verticale riguarda i sistemi di produzione intelligenti che stanno a supporto dell'integrazione orizzontale.

Internet of Things

I sensori e i macchinari possono far parte della stessa rete, essere integrati con dei dispositivi di elaborazione computazionale e collegati tra loro attraverso tecnologie standard. Le "cose" possono così comunicare e interagire tra loro.

Cybersecurity

L'aumento della condivisione dei dati tra sistemi e dispositivi ha reso necessaria una maggiore protezione della rete informatica da potenziali minacce. Si tratta di una tutela legata all'ambito del cyberspace, sia degli elementi fisici che degli elementi digitali che processano le informazioni raccolte. Le interconnessioni tra computer, macchine e dispositivi hanno fatto sorgere moltissime vulnerabilità, che è necessario combattere proteggendo da forze distruttive (intenzionali e non) e dalle azioni indesiderate di utenti non autorizzati.

Cloud

Con l'industria 4.0 più imprese legate alla produzione hanno aumentato la quantità di dati condivisi attraverso i siti e superando i confini dell'azienda. Allo stesso tempo le performance delle tecnologie cloud sono migliorate raggiungendo tempi di elaborazione molto bassi. La quantità sempre maggiore di dati che le imprese raccolgono non può più essere gestita con i server tradizionali. Il cloud favorisce l'agilità aziendale, fa sfruttare le potenzialità IT e consente di rispondere in tempo reale alle varie esigenze.

Manifattura additiva

I metodi di manifattura additiva, nota anche come stampa 3D, si riferiscono alla produzione di oggetti a partire da modelli virtuali. Si differenzia dalla manifattura tradizionale perché, anziché asportare il materiale da un pezzo solido o modificare la forma di un pezzo senza alternarne il volume, realizza un oggetto sovrapponendo materiale. Questa porta alla digitalizzazione della produzione, dall'idea si arriva alla materializzazione del prodotto. In questo modo è possibile realizzare piccoli lotti di prodotti altamente personalizzati.

Realtà aumentata

Si tratta di una tecnologia che permette di alterare e aumentare la realtà che l'uomo normalmente può percepire. Essa supporta una grande varietà di servizi e può fornire ai lavoratori informazioni in tempo reale migliorando il processo decisionale e le attività lavorative.

2.3.2 Smart Manufacturing

Il livello di innovazione industriale è tale per cui oggi il sinonimo di *Industria 4.0* è *Smart Manufacturing*. Con il termine "smart" si accomunano tutti gli elementi di una gestione integrata delle informazioni, associata all'uso della tecnologia digitale. La "Smart Manufacturing" insieme alla "Smart Supply Chain", diventano nuove declinazioni del paradigma *Internet of Things* che sta rivoluzionando molti settori e ambiti aziendali. [9] La paternità del termine "Smart manufacturing" è degli Stati Uniti. Mentre in Germania nel 2012 veniva definito il programma Industry 4.0, negli Stati Uniti veniva costruita la *Smart Manufacturing Leadership Coalition (SMLC)*, un'associazione no profit nata per favorire la collaborazione tra aziende produttrici, enti di ricerca, università con lo scopo di sviluppare standard, piattaforme e infrastrutture condivise per l'adozione dello Smart manufacturing. [10] La manifattura tradizionale ha perso i propri vantaggi a favore della tecnologia relativa alla produzione intelligente a cui i Paesi Industrializzati hanno dedicato un'attenzione sempre maggiore. Nello "smart manufacturing" tutto è collegato con l'aiuto di sensori e chip RFID. I prodotti, i mezzi di trasporto, gli strumenti e i macchinari comunicano tra loro con l'obiettivo di migliorare la produzione complessiva. I *Cyber-Physical Production System (CPPS)* hanno permesso la connessione tra le attrezzature industriali fisiche, i dispositivi su Internet e l'analisi dei big data diventando la materializzazione dei più generici *Cyber-Physical System (CPS)* nell'ambiente produttivo. I CPS sono costituiti da una componente fisica e una virtuale. La componente fisica permette attraverso sensori, memorie e capacità computazionale di percepire "il mondo reale". La componente digitale è costituita da un digital twin del dispositivo materiale che permette di simularne il comportamento e prevenirne gli errori, determinare le condizioni operative ottimali in termini di costi e rischi e monitorarne la correttezza ed efficienza durante tutto il ciclo di vita.

La smart manufacturing ha suscitato interesse sia da parte del settore industriale che dagli accademici e studiosi. I dati pubblicati dal 2005 al 2016 relativi alla manifattura intelligente sono stati raccolti dallo Scopus database il quale mostra una crescita significativa di documenti pubblicati nel 2015. [7]

Il concetto di Industria 4.0 nel settore manifatturiero copre una vasta di applicazioni, dal design alla logistica. Il ruolo della meccatronica è stato adattato a quello dei CPS. La manutenzione predittiva e le sue applicazioni è tra gli argomenti più importanti trattati dall'Industria 4.0. Anche la gestione dell'energia prevede un sistema di monitoraggio dei consumi autonomo e auto ottimizzato basato su un sistema decisionale. A questo proposito è stato proposto un framework relativo allo *Smart Manufacturing System* che incrocia tra loro i vari aspetti dell'industria 4.0 su più dimensioni e applicazioni.

- *Smart design*: si è sviluppato grazie a nuove tecnologie come la realtà aumentata e la realtà virtuale. I software di design come CAD (Computer Aided Design) o CAM (Computer Aided Manufacturing) permettono di interagire con i prototipi in tempo reale attraverso la stampa 3D integrata ai CPS e AR.

- *Smart machining*: si è sviluppato attraverso gli smart robot e altri dispositivi che possono percepire e interagire reciprocamente in tempo reale. Le macchine CPS sono in grado di raccogliere i dati e inviarli a dei sistemi centrali cloud based.

- *Smart monitoring*: il monitoraggio è un aspetto importante per il funzionamento, mantenimento e schedulazione ottimale per i sistemi produttivi 4.0. Questo è stato reso possibile dalla diffusione di sensori che raccolgono dati come la temperatura, il consumo elettrico, le vibrazioni e le velocità. Questo permette di avere non solo una visione grafica di quello che succede ma anche di avvisare in caso di anomalie alle macchine e alle strumentazioni. CPS e IoT sono gli elementi chiave di questo aspetto.

- *Smart control*: è eseguito soprattutto per la gestione di macchinari e strumentazioni smart attraverso piattaforme cloud.

- *Smart scheduling*: utilizza modelli avanzati e algoritmi per tracciare l'informazione catturata dai sensori. La schedulazione può essere effettuata attraverso tecniche data-driven e architetture decisionali avanzate.

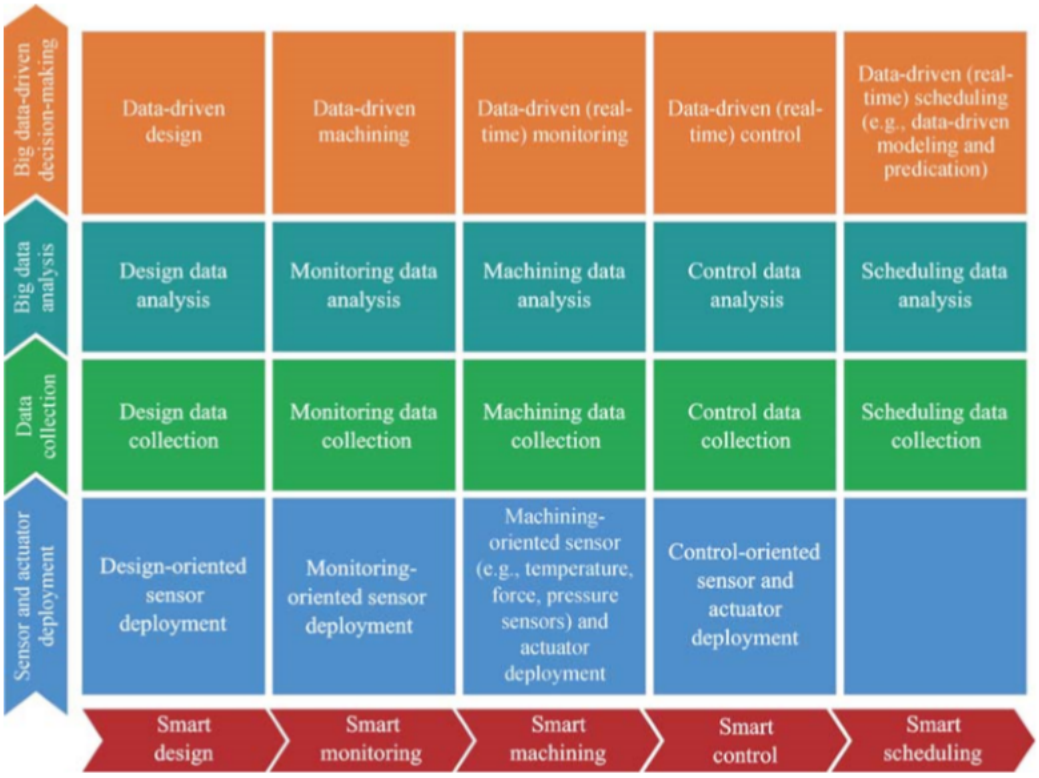


Figura 2.2: Framework concettuale smart manufacturing [7]

Lo smart decision making è al centro dell'Industria 4.0. L'obiettivo finale dell'impiego dei sensori è quello di ottenere un modello decisionale attraverso la raccolta dei dati. Big data e le sue analisi hanno un ruolo cruciale per la realizzazione di modelli data driven e manutenzione predittiva. Grazie all'automazione e all'interazione cyber-fisica è possibile ridurre le tempistiche dei processi decisionali diminuendo l'intervento umano. Ottimizzare i processi portando efficienza e maggiore visibilità in ogni anello della supply chain garantisce una comunicazione multidirezionale nell'ambito di tutti i processi produttivi. I dati offrono alle aziende una nuova capacità di analisi predittiva che aiuta a creare una base informativa per il miglioramento dei prodotti e a supportare al meglio le decisioni. Non c'è Industry 4.0 senza Big Data Management e Business Intelligence che permette di offrire alle aziende misure sempre più personalizzate. In questo modo le imprese possono scegliere la migliore innovazione tecnologica rispettando i sempre più stringenti vincoli di budget, trasformando gli investimenti iniziali in costi ricorrenti.[7] [9]

2.3.3 Tecnologie per lo Smart Manufacturing

Il modello di business dello smart manufacturing è reso possibile dall'adozione di tutte le nove tecnologie abilitanti l'industria 4.0 anche se alcune di esse ricoprono un ruolo particolare: IoT industriale, Big Data & analytics, robot autonomi e integrazione orizzontale e verticale. Il modello di business dello smart manufacturing integra in mondo fisico con quello digitale per automonitorarsi, autoapprendere, autogestirsi e autoadattarsi. É necessario un elevato livello di integrazione sia di tipo verticale tra tutti i livelli di automazione, sia di tipo orizzontale, tra tutti gli attori della catena del valore interna ed esterna. L'integrazione verticale dei sensori avviene in due forme: uomo-macchina e macchina-macchina. Nel primo caso i robot autonomi vengono utilizzati per attività complesse a supporto della risorsa umana, nel secondo viene creato un network che permette uno scambio continuo di informazioni per ottimizzare i processi. L'integrazione orizzontale invece consiste nell'interazione tra le imprese lungo la value chain permettendo la condivisione delle informazioni e la collaborazione tra gli attori che intervengono durante tutto il ciclo di vita del prodotto. [4]

Internet of things industriale

L'internet of things è uno degli elementi chiave dell'industria 4.0 e può essere definito come un network di sistemi fisici che interagiscono tra loro con un protocollo standard di comunicazione per raggiungere un obiettivo comune. È una comunicazione che coinvolge i macchinari impiegati, le componenti e i prodotti, senza l'intervento umano e con l'uso di dispositivi elettronici wireless sfruttando i CPS. L'IoT integra le Information Technology con le Operation Technology portando l'industria sempre più verso la digitalizzazione. [4]

La combinazione delle informazioni e dei dati provenienti dai diversi dispositivi rende possibile il decentramento del monitoraggio dei processi fisici e il controllo da remoto in modo da individuare e risolvere i problemi a distanza. L'Internet of Things utilizza terminali intelligenti con capacità di rilevamento, comunicazione ed elaborazione per acquisire le informazioni dal mondo fisico. Per ottenere un rilevamento completo è necessario che i vari dispositivi coinvolti siano interconnessi tra loro. I dispositivi possono essere di varia natura ad esempio codici a barre,

tecnologia di identificazione a radiofrequenza, reti di sensori wireless, CPS e sistemi machine-to-machine (M2M). L'IoT comprende tre livelli:

- Livello di estensione percettiva, per la raccolta delle informazioni relative al mondo reale;
- Livello di rete, cioè una rete di comunicazione eterogenea;
- Livello di servizio, servizi applicativi per il rilevamento di informazioni per i dispositivi terminali, come telefoni cellulari e PC.

Il concetto di "Internet of Things" è stato presentato per la prima volta nel 1999 dal MIT Auto ID Center e ideato da un imprenditore inglese, Kevin Ashton, il quale ha immaginato un sistema dove tutto il mondo materiale è interconnesso, scambia informazioni raccolte attraverso sensori e prende decisioni sulla base dell'elaborazione di tali informazioni. Internet è una delle quattro tecnologie utilizzate per realizzare l'IoT insieme a RFID, nanotecnologia, tecnologia dei sensori e tecnologia intelligente integrata. Nel 2009, IBM ha proposto "Smart Earth" dando vita al boom dell'Internet of Things in tutto il mondo. [11] L'integrazione tra Information Technology e Operation Technology porta l'impresa verso una digitalizzazione sempre più ampia con un continuo passaggio dal mondo fisico al digitale e viceversa. L'IoT introdotto nel mondo delle imprese offre grandi opportunità, il suo utilizzo infatti a prescindere dalla dimensione della fabbrica e dal tipo di prodotti porta grossi benefici sia all'impresa stessa che al consumatore finale, coinvolto nelle fasi di ingegnerizzazione e progettazione del prodotto e del servizio. La risorsa alla base dell'IoT sono i dati, essi vengono raccolti in tempo reale dai sensori di cui i dispositivi sono dotati e sono utilizzati per migliorare l'efficienza dei processi e la risoluzione dei problemi. Il sistema dei sensori, degli attuatori e dei trasmettitori permette una raccolta più agevole dei dati grazie alla loro economicità, alle piccole dimensioni e alla sofisticatezza. I dispositivi sono interconnessi tramite trasmissioni wireless che permettono lo scambio di informazioni, interagiscono tra loro e monitorano i processi. La presenza di un software di rete permette la comunicazione machine-to-machine, la manutenzione predittiva e il maggior coinvolgimento del cliente nell'uso del prodotto. Le caratteristiche che questi strumenti devono avere sono: l'interoperabilità, assicurata da standard comuni, in quanto spesso dispositivi prodotti da vendor diversi non possono interconnettersi tra loro, il consumo di

energia deve essere parsimonioso in quanto la maggior parte di questi strumenti funziona a batteria, la manutenzione e il costante aggiornamento, che diventano dispendiosi se si pensa all'elevato numero di device presenti. È importante che anche le risorse umane abbiano le competenze adatte alle nuove tecnologie in termini di sviluppo software e di analisi. Le imprese che investono in IoT devono dunque tener conto anche della formazione del proprio personale. Insieme all'IoT nascono gli smart product, essi permettono ai produttori di conoscerne le performance lungo tutto il ciclo di vita e di utilizzarle per migliorarli, soddisfare le aspettative dei clienti e identificare nuovi servizi. L'impatto economico-finanziario della tecnologia IoT si ritrova ad esempio nella manutenzione predittiva, che apporta benefici in termini di riduzione dei costi, permettendo di limitare le ispezioni manuali e i guasti non previsti. La crescita della competitività nel settore ha portato le aziende ad aumentare i propri investimenti e questo è stato compensato dall'aumento dei ricavi, che hanno accresciuto il valore aggiunto della produzione. [4] E' un dato di fatto che le imprese che hanno introdotto nei loro impianti tecnologie abilitanti, tra cui l'IoT, hanno stimato una crescita di efficienza produttiva pari al 30-50[12]

2.3.4 Big data & analytics

Con il termine *Big Data* si intendono quelle tecnologie che supportano il processo di raccolta, organizzazione e analisi di grandi quantità di dati provenienti da una varietà di fonti diverse. Il concetto di "Big data" è legato anche alla capacità computazionale dei modelli per l'elaborazione dei dati in tempo reale. Le tecnologie legate all'elaborazione dei big data devono risolvere problematiche legate a quattro dimensioni: volume, ovvero la dimensione dei dati, la velocità con cui questi dati sono raccolti, la varietà riferita all'eterogeneità dei dati e la veridicità che riguarda l'attendibilità dei dati. In questo contesto riveste un ruolo chiave la Data Science, ovvero l'insieme di principi metodologici basati sul metodo scientifico e delle tecniche multidisciplinari volto a interpretare a estrarre conoscenza dai dati. È possibile trattare dati strutturati e non strutturati combinando una serie di tecniche statistiche, matematiche e di programmazione. I dati, alla base di questa tecnologia, sono utilizzati per migliorare tutti i processi aziendali e lo sviluppo e il funzionamento dei prodotti. Per poter gestire i dati è necessaria la figura di un data scientist che abbia competenze in materia di software e algoritmi oltre dei metodi per analizzare, valutare, progettare e gestire sistemi, strutture e processi complessi.

L'impatto economico è duplice, dal lato dei costi si ha una netta riduzione dei guasti nei processi produttivi e dal lato ricavi in modo diretto con la vendita e in modo indiretto con lo sfruttamento per il miglioramento generale di processi e prodotti.

Industrial big data environment

Le tecniche di data mining hanno dato le radici ai social network. Molte organizzazioni di ricerca e aziende si sono dedicate a questo nuovo argomento di studio concentrandosi per lo più sui social o sull'ambito commerciale. Quest'ultimo in particolare include previsioni di vendita, clustering e apprendimento delle relazioni con gli utenti, opinion mining ecc. Tuttavia queste ricerche si focalizzano su dati generati dall'uomo anziché dati industriali o prodotti dai macchinari, che includono sensori, controller dei macchinari e sistemi produttivi. Come si è già visto, nell'era dell'Industria 4.0 i sistemi analitici intelligenti e i sistemi cyber-fisici sono integrati insieme per realizzare un nuovo concetto di gestione della produzione e di trasformazione industriale. Usando l'installazione di sensori appropriati possono essere estratti dati su vari segnali come vibrazioni, pressione ecc. Possono essere inoltre raccolti dati storici per ulteriori approfondimenti sui dati. L'aggregazione di tutti questi dati raccolti prende il nome di Big Data. Gli agenti di trasformazione sono costituiti da più componenti ad esempio piattaforme integrate, analisi predittive e tool di visualizzazione. La piattaforma di distribuzione è scelta sulla base della velocità computazionale, costi e facilità di distribuzione e aggiornamento. Lo sviluppo del framework dell'IoT e l'emergere della tecnologia di rilevamento con sensori hanno creato un'informazione che unisce umani e sistemi, popolando ulteriormente il contesto dei Big data. Successivamente con l'avvento del cloud computing e dei CPS il futuro dell'industria è stato in grado di raggiungere un sistema di informazioni che aiutano i macchinari ad auto apprendersi e prevenire potenziali anomalie. Per un sistema meccanico l'autoapprendimento consiste nell'essere in grado di valutare le condizioni presenti e passate della macchina e reagire alla valutazione di output. La valutazione dello stato di salute del macchinario può avvenire usando un algoritmo data-driven per analizzare le informazioni raccolte dal macchinario stesso o dall'ambiente in cui esso si trova. Tuttavia gli algoritmi di diagnosi o di prognosi solitamente sono relativi ad uno specifico macchinario, motivo per cui non sono abbastanza flessibili o adattabili. [13] Utilizzare big data

nel mondo dell'industria, quindi, può consentire di ottenere vantaggi su quattro "assi" fondamentali per il business, a partire dalla qualità della produzione, che può essere monitorata e indirizzata ottenendo risultati migliori volta per volta, intervenendo sia sul miglioramento della funzionalità, sia sulle prestazioni sia in chiave estetica. La prospettiva di cui si parla già con insistenza è quella di arrivare a una produzione "personalizzata" con prodotti che si adattano sempre più alle esigenze dei singoli utenti, senza per questo dover sostenere i costi estremamente alti che si dovevano affrontare finora per apportare cambiamenti al processo di produzione. Un'opportunità che può tradursi immediatamente in più competitività sul mercato. Da considerare inoltre che la digitalizzazione, quando si parla di aumentare la qualità dei prodotti, non si limita a interessare soltanto gli ambienti produttivi, ma arriva a coinvolgere anche il rapporto con gli utenti finali, con i feedback dei consumatori che vengono utilizzati per orientare la produzione. Un altro dei vantaggi apportati dai big data è la riduzione del time to market cioè il tempo che intercorre tra la progettazione di un prodotto e la sua messa sul mercato. Grazie al digitale la prototipazione diventa più agile, questo consente di arrivare in tempi più rapidi alla fase di produzione e di gestirla nel modo più "razionale", mantenendo sempre il controllo sull'intera supply chain, riducendo quindi al minimo indispensabile le scorte di magazzino e i costi dovuti a questa parte della filiera. Il terzo effetto tangibile è quello che si riscontra sui costi: utilizzare l'analisi dei big data nella produzione consente una programmazione degli acquisti e degli utilizzi delle materie prime con margini di errore minimi. Un altro ambito in cui gli analytics possono raggiungere risultati importanti è quello dei consumi energetici: avere infatti il quadro completo dell'energia consumata in tempo reale da un impianto produttivo consente di attuare una serie di accorgimenti che possano ridurre i consumi nei momenti di picco e razionalizzare l'utilizzo dell'energia all'interno della fabbrica. Un altro elemento rivoluzionario è la flessibilità. Si ha la possibilità di personalizzare la produzione rendendola agile, controllandone il ritmo e modificandolo in base a ciò che arriva dal mercato e dai consumatori e adattandola in tempo reale ai gusti e alle preferenze, sulla base dei feedback ricevuti. [14]

Nel settore manifatturiero i big data implicano un elevato volume di dati generati dall'intero ciclo di vita dei prodotti. È così possibile trovare i colli di bottiglia all'interno di un processo e rilevarne le cause, gli impatti e trovare delle soluzioni.

Aumentano in questo modo l'efficienza e la competitività. In questo ambito è necessario porre l'attenzione sull'interazione tra mondo fisico e mondo cibernetico. La "digital twin" apre la strada all'integrazione tra i due mondi. Essa crea dei modelli virtuali per gli oggetti fisici attraverso rilevamento dei dati in modo da predire, stimare e analizzare i cambiamenti. In questo modo la digital twin può permettere l'ottimizzazione dell'intero processo manifatturiero.

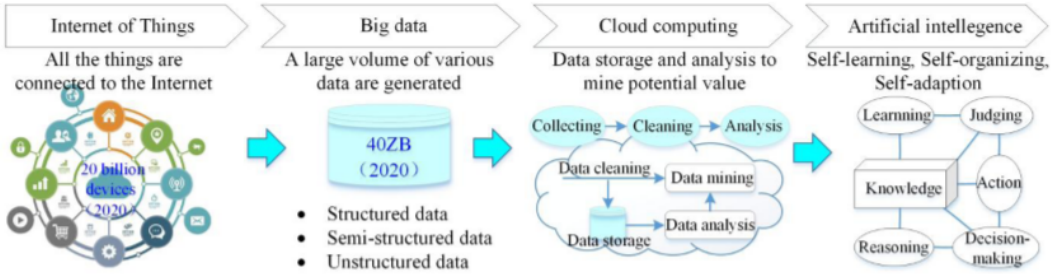


Figura 2.3: Processo di applicazione Big Data [15]

Per gli utenti i dati sono anche interpretati come la capacità di ottenere valore e informazione nascosta da un elevata quantità di dati. I big data possono anche essere definiti con le 4Vs:

- Volume: la scala dei dati è dell'ordine dei Terabyte
- Varietà: la dimensione, il contenuto, il formato e le applicazioni sono diversificati. Per esempio i dati possono essere strutturati (numeri, simboli e tabelle), semi strutturati (alberi, grafi, documenti XML) o non strutturati (loghi, audio, immagini, video).
- Velocità: significa che la generazione dei dati è molto rapida, di conseguenza l'elaborazione dei dati richiede elevate tempestività.
- Valore: la chiave della competitività è come estrarre il valore attraverso algoritmi potenti.

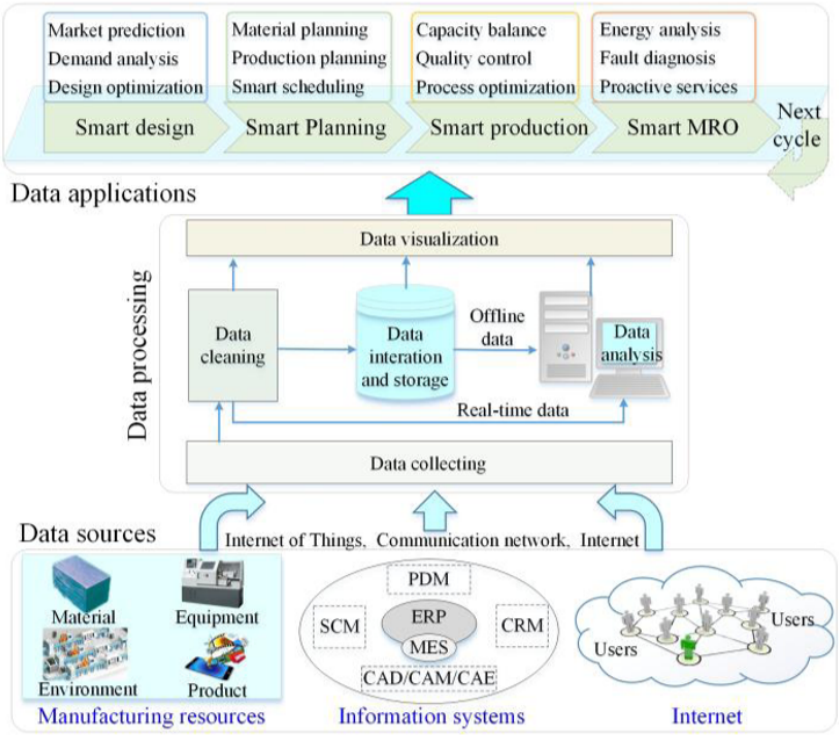


Figura 2.4: Fonti, processi e applicazioni dei big data nella produzione [15]

Le fonti dei dati possono essere diverse. Ad esempio le risorse produttive della smart factory producono dati raccolti attraverso l'uso di tecnologie IoT, dati ambientali, dati dai materiali o dai prodotti stessi. Altri dati provengono dalla gestione dei sistemi informativi manifatturieri (ERP, CRM) oppure dai sistemi computer aided (CAD,CAE,CAM). Un'altra fonte molto diffusa è la rete come per i dati utente dalle piattaforme di e-commerce o dai social network, dati pubblici da siti web open delle istituzioni pubbliche. Poiché i dati grezzi non sono molto utili è necessario processarli attraverso una serie di step per estrarne valore. Prima di tutto i dati sono raccolti attraverso IoT, API, (Application Programming Interface), SDK (Software Developement Kit) ecc. I dati provengono da più sorgenti, sono eterogenei, multiscala e ricchi di rumore pertanto è necessario ripulirli prima di processarli. I dati puliti a questo punto sono integrati e storicizzati per la condivisione su tutti i livelli. Successivamente, con il cloud computing, i dati in tempo reale e quelli raccolti off line sono analizzati e compresi attraverso analisi avanzate come machine learning o modelli di previsione. La preziosa conoscenza estratta

permette ai produttori di approfondire la comprensione dei vari step del ciclo di vita dei prodotti e di prendere delle decisioni in maniera più razionale, informata e responsabile.[15] Infine le applicazioni dei Big Data nel settore produttivo possono riguardare diversi aspetti:

1. Nell'ambito del product design ci si sposta da una progettazione ispirata e basata sull'esperienza ad una progettazione guidata dai dati e dalle analisi. I dati sui comportamenti degli utenti e dei trend del mercato permettono di quantificare la domanda e definire i requisiti di qualità.

2. In relazione ai dati globali, i programmi di pianificazione ottimizzati e globali migliorano la velocità e l'accuratezza.

3. Il controllo e il miglioramento sono integrati in ogni singolo step, dalle materie prime ai prodotti finiti. Ad esempio, il rilevamento anticipato dei difetti di qualità e la rapida diagnosi dei problemi di funzionamento permettono di garantire un elevato livello di qualità in tempo reale.

4. Infine i big data hanno apportati grandi cambiamenti nei modelli MRO (Mantainace, Repair, Overhaul) passivi tradizionali, adesso è possibile monitorare lo stato di salute dei prodotti e dei macchinari realizzando un MRO attiva e preventiva.

Digital twin

Il concetto di Digital twin è stato presentato per la prima volta da Grieves. I modelli virtuali degli oggetti fisici sono creati in un modo digitale per simulare il loro comportamento nel mondo reale. Pertanto il digital twin è composto da tre componenti:

- Entità fisiche in un mondo fisico
- Modelli virtuali in un mondo virtuale
- Dati che connettono i due mondi

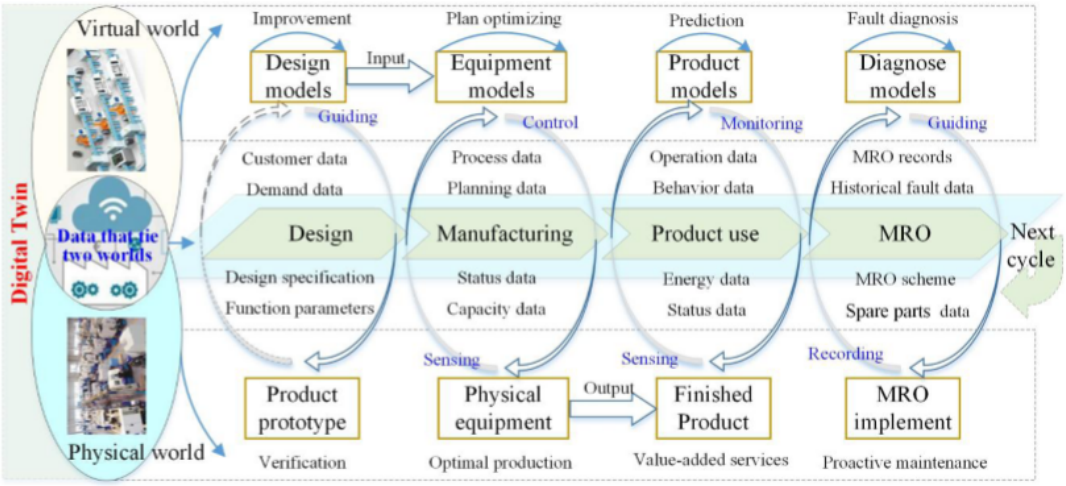


Figura 2.5: Digital twin in ambito manifatturiero [15]

I processi fisici in questo modo sono giudicati, analizzati, predetti e ottimizzati attraverso mezzi virtuali. In seguito alla simulazione e ottimizzazione dei processi di design, manifattura o di manutenzione il digital twin guida i processi fisici verso una soluzione ottimizzata. I dati dal mondo fisico sono trasmessi attraverso i sensori per completare la simulazione, la validazione e la dinamica di aggiustamento. Viceversa i dati della simulazione ritornano al mondo fisico per rispondere ai cambiamenti, migliorare le operazioni e accrescere il valore. Queste analisi incrociate sono possibili solo attraverso la convergenza dei dati. Tra le varie applicazioni, come si vede in figura 2.5, è importante porre particolare attenzione ai modelli di diagnosi. La manutenzione predittiva si basa sulla previsione delle condizioni di salute, vita rimanente e guasti in modo da evitare interruzioni improvvise. Quando avviene un guasto esso può essere diagnosticato e analizzato attraverso i modelli virtuali che permettono ai tecnici di visionare dove l'arresto è avvenuto e altre informazioni importanti. Le strategie di *Maintenance, Repair and Overhaul (MRO)* vengono prima applicate sulla realtà virtuale o sulla aumentata, in questo modo è possibile valutare quali sono quelle più efficaci e ottimali prima di eseguirle sul prodotto fisico. Tutti i dati relativi ad ogni step del ciclo di vita del prodotto sono raccolti per sfruttarli nelle future innovazioni del prodotto stesso. Sebbene esistano molte differenze tra i big data e il digital twin essi svolgono dei ruoli complementari nella produzione. Senza i big data la maggior parte delle funzioni dei digital twin non

esisterebbero. E senza il digital twin l'analisi dei dati non sarebbe in parallelo con la produzione reale. I progettisti creano degli scenari di simulazione per prevedere la qualità e la fattibilità. Se lo schema di progettazione non ha superato il test di simulazione esso deve essere riprogettato in tempo reale. I big data hanno un ruolo cruciale per l'identificazione dei problemi e l'inserimento di migliorie nello schema di progettazione. Nel MRO i modelli virtuali di prodotti fisici sono sincronizzati con lo stato reale del prodotto attraverso i sensori. Lo stato operativo del prodotto e lo stato di integrità vengono acquisiti in tempo reale. Integrando i dati storici con i dati in tempo reale il digital twin può anche rilevare problemi sconosciuti confrontando la risposta reale con quella virtuale, in questo modo è possibile simulare e ottimizzare programmi di manutenzione per questi problemi sconosciuti e facilitare così la manutenzione effettiva. L'analisi dei big data si occupa dell'analisi dei dati necessari per la produzione intelligente, invece il digital twin compensa per gli inconvenienti che i big data non sono in grado di simulare. Questo implica che la convergenza di questo due tecnologie è fondamentale per la produzione intelligente.[15]

2.4 Prognostic and Health Management (PHO)

In letteratura, i sistemi di *Prognosis and Health management (PHM)* sono stati studiati da molti ricercatori in diversi campi per migliorare l'affidabilità, la disponibilità, la sicurezza e per ridurre i costi di manutenzione degli asset. I profitti e la competitività di un'impresa dipendono dalla progettazione e dalla realizzazione di sistemi o prodotti di buona qualità. Tuttavia progettare sistemi sofisticati comporta molte problematicità e preoccupazioni relative ai costi di manutenzione. È necessario dunque sviluppare un sistema di gestione e valutazione dello stato di salute dei macchinari, come PHM. La disciplina dei sistemi di PHM fornisce una visione generale dello stato di salute delle macchine o di sistemi complessi e assiste ai processi decisionali sulla loro manutenzione. I compiti principali delle tecnologie PHM sono: rilevare i guasti o le componenti difettose, eseguire la diagnostica e la prognostica dei guasti e gestire lo stato di salute degli asset produttivi. Tra gli obiettivi principali nella costruzione di un sistema PHM robusto ci sono: la stima dello stato di salute attuale, predizione di stati futuri e il loro impatto sulle performance del sistema. Esistono tre diverse categorie di approcci PHM:

- Model-based prognostics: il processo di degradazione verso i guasti è descritto da un modello matematico o da un'equazione derivata dai sistemi fisici. Questi modelli sono più accurati rispetto agli altri approcci e hanno un orizzonte temporale più lungo per la predizione della *RUL (Remaining useful life)* ma necessitano della conoscenza degli esperti. Inoltre, creare dei modelli a partire dai sistemi fisici reali è una vera e propria sfida per la sua complessità e degradazione stocastica dei comportamenti delle componenti.

- Data-driven prognostic: si prova a costruire un modello di degradazione usando i dati raccolti attraverso i sensori e a predire il futuro stato di salute. Questo approccio può essere facilmente applicato ai problemi di predizione ma richiede un tempo computazionale più elevato rispetto agli approcci model-based. L'accuratezza in questo caso dipende molto dalla numerosità dei dati disponibili.

- Hybrid Prognostic Approach: i modelli precedenti hanno i loro vantaggi e svantaggi, con il tempo si è cercato di creare un approccio che potesse integrare le potenzialità di entrambi minimizzandone i limiti per creare una predizione migliore dello stato di salute e della vita utile rimanente dei sistemi a cui è applicato. Non esistono approcci migliori di altri ma è importante capire come sfruttarli al meglio a seconda del singolo caso di applicazione.

Ciascun approccio può essere valutato attraverso specifiche metriche: il tempo di esecuzione descrive il tempo impiegato per il processo di stima e predizione; l'accuratezza descrive la bontà della stima o della predizione degli strumenti nelle attività di prognostica. La robustezza è l'abilità della prognosi nel gestire il livello di rumore e le situazioni di incertezza del sistema. L'orizzonte di predizione descrive la capacità dello strumento prognostico di prevedere l'evoluzione futura dei guasti con una scala temporale della vita residua degli asset, più è lungo l'orizzonte di previsione maggiore sarà il peso di questa metrica. Il tempo di apprendimento è il tempo impiegato per la prognosi data-driven per allenare il modello di misurazione usando i dati, dipende dalle dimensioni, dal tipo e dalla qualità dei dati stessi.[16]

2.4.1 Tipologie di manutenzione

La manutenzione è alla base della PHM, essa fu definita nel 1963 dall'OCSE con: "S'intende per manutenzione quella funzione aziendale alla quale sono demandati il controllo costante degli impianti e l'insieme dei lavori di riparazione e revisione necessari ad assicurare il funzionamento regolare e il buono stato di conservazione degli impianti produttivi, dei servizi e delle attrezzature di stabilimento." L'obiettivo principale della manutenzione è minimizzare non solo i guasti o le interruzioni dei macchinari ma anche i costi operativi. [17] La manutenzione può essere suddivisa in due tipi di strategie: correttiva e preventiva.

Manutenzione correttiva

La manutenzione correttiva è effettuata dopo il riconoscimento del guasto con l'obiettivo di far ritornare l'oggetto allo stato in cui può svolgere la propria funzione. La manutenzione correttiva è una strategia manutentiva che include tutte le azioni di manutenzione non previste per riportare il sistema ad una specifica condizione. Questo tipo di approccio comporta costi molto elevati legati sia alle perdite di produzione sia ai guasti improvvisi.[18] In questo caso l'azione manutentiva è subordinata all'attesa del manifestarsi del guasto. Solo a guasto avvenuto è possibile eseguire l'intervento che riporta la prestazione del sistema al momento precedente al problema. In questo caso non si tratta di azioni volte a migliorare le prestazioni dell'oggetto in questione ma semplicemente viene ripristinato lo "stato quo ante" [19]

Manutenzione preventiva

L'applicazione della manutenzione preventiva si basa sull'approccio scientifico presentato negli anni '50. Il vantaggio più importante è che tale approccio si basa su decisioni costruite a partire dai fatti reali. In questo caso la manutenzione avviene prima dell'arrivo del guasto ed è effettuata ad intervalli periodici a seconda dello stato di salute del sistema, riducendo la probabilità di guasto o la degradazione del funzionamento. Con la manutenzione preventiva si cerca di conservare lo stato di un oggetto ad una specifica condizione attraverso ispezioni, indagini e prevenzione dei guasti. Tra gli svantaggi di questo tipo di manutenzione c'è il fatto che spesso gli intervalli periodici sono definiti sulla base dell'esperienza e della

conoscenza degli eventi precedenti, pertanto l'impresa è fortemente vincolata alle conoscenze dei propri ingegneri e operatori. Inoltre, ogni macchinario lavora in un ambiente diverso e necessita di una differente manutenzione preventiva e a intervalli diversi. Si può anche correre il rischio che i fornitori dei macchinari possano voler aumentare la vendita dei pezzi di ricambio massimizzando il numero degli interventi preventivi.[18] Esistono diverse categorie di manutenzione preventiva [20]:

- Manutenzione time-based: si occupa della sostituzione o rinnovo di un determinato articolo a intervalli fissi nel tempo indipendentemente dalle sue condizioni. Questa può essere applicata quando le parti soggette a usura hanno un tempo medio tra i guasti conosciuto e si presuppone che il problema sia legato alla vita utile. È chiaro che questo tipo di manutenzione non può coprire la vasta gamma dei guasti in quanto solo una piccola percentuale è legata all'età dell'oggetto in questione.

- Ricerca dei guasti: ha lo scopo di rilevare i guasti nascosti associati a funzioni di protezione in quanto non si saprà se un'apparecchiatura è funzionante finché non arriva il momento in cui deve entrare in funzione.

- Manutenzione risk-based: si utilizza per assegnare le risorse di manutenzione sulla base del rischio associato all'asset in caso di guasto. Le apparecchiature con un rischio maggiore sono soggette a ispezioni più frequenti. È importante minimizzare il rischio dell'intero impianto attraverso un'analisi accurata e una valutazione dei rischi.

- Manutenzione condition-based: la maggior parte delle modalità di guasto non è correlata all'età, questa categoria di manutenzione cerca prove fisiche che si stia verificando un errore. C'è un momento in cui il guasto inizia a manifestarsi e può essere rilevato prima del suo blocco funzionale. Questo intervallo di tempo è importante al fine di rilevare il guasto imminente e intervenire in tempo. Ciò è reso possibile da un processo efficiente ed efficace per la raccolta dei dati, l'analisi e il processo decisionale.

- Manutenzione predittiva: inizialmente essa si riconduceva al termine di Manutenzione condition-based (CBM), oggi con l'avvento dell'intelligenza artificiale, i bassi costi delle apparecchiature e dell'apprendimento automatico è opportuno differenziarle. La manutenzione predittiva può essere considerata come

un'evoluzione della manutenzione CBM in quanto è grazie ai sensori installati che è possibile monitorare le condizioni operative dell'impianto e capire quando ci si sta avvicinando ai guasti. Nel paragrafo successivo questa categoria di manutenzione sarà affrontata più nel dettaglio.

2.4.2 Manutenzione predittiva

Quando si parla di manutenzione predittiva si pensa alla possibilità di predire e prevedere i fermi macchina con l'obiettivo di ridurre al massimo i guasti sulla produzione. In realtà essa rappresenta una vera e propria svolta per la trasformazione delle imprese provocando dei grandi cambiamenti anche a livello di business. Si tratta di un servizio che diventa parte essenziale di un modello di lungo termine, in grado di generare ricavi e creare valore addizionale nella relazione tra un'impresa e i suoi clienti. L'adozione della predictive mantainance porta con sé una serie di benefici. Il primo è sicuramente la creazione di un modello "as a service" che permette di rafforzare le relazioni con i propri clienti. Il secondo beneficio è il miglioramento della qualità dei prodotti. Raccogliendo i dati dai macchinari o dagli strumenti sul campo, ne deriva una più profonda comprensione del loro comportamento e delle loro performance. Il terzo beneficio è la riduzione dei guasti, cercando di minimizzare il più possibile la loro ricorrenza e di conseguenza i costi ad essi associati. Questo è un elemento molto importante se si pensa all'impatto economico di un'interruzione non pianificata. Un altro vantaggio è la pianificazione della predictive maintanance che consente di ottimizzare la logistica delle parti di ricambio riducendone così i costi di gestione.[21] La manutenzione predittiva è tra i temi più caldi dell'industria 4.0. Essa rappresenta una vera e propria evoluzione rispetto ai modelli del passato. Si tratta di perseguire due obiettivi molto interconnessi tra loro: la previsione di quando un guasto o malfunzionamento potrebbe verificarsi e il prevenirli attraverso l'attività di manutenzione. Questo approccio si distacca da quello applicato dalla manutenzione reattiva, che prevede un intervento all'occorrenza del guasto, e dalla manutenzione preventiva o pianificata, nella quale gli interventi avvengono a intervalli di tempo prestabiliti anche senza una effettiva necessità. La predictive mantainance consente di intervenire solo quando necessario ottimizzando in questo modo i tempi e i costi di intervento.

Figura 2.6: Vantaggi predictive maintenance [21]

Manutenzione predittiva in Italia

Uno studio effettuato dall'Osservatorio Internet of Things del Politecnico di Milano ha valutato il mercato italiano dell'IoT a circa 3,7 miliardi di euro. Questo conferma come l'IoT sia una tecnologia abilitante per una varietà molto ampia di soluzioni e servizi. La ricerca dell'Osservatorio ha evidenziato come le applicazioni oggi più significative siano legate a soluzioni per la valorizzazione in real time di data analytics (31% per la gestione dell'avanzamento della produzione, 28% predictive maintenance e 22% per il supporto agli operatori nello svolgimento delle attività sulla linea di produzione. La manutenzione predittiva è un passo avanti verso la macchina come servizio, ed è proprio sui servizi che si sta realizzando un'importante innovazione dall'installazione di oggetti smart all'invio di notifiche in caso di situazioni di emergenza.

Il crescere del volume dei dati porta ad una maggiore necessità di razionalizzare

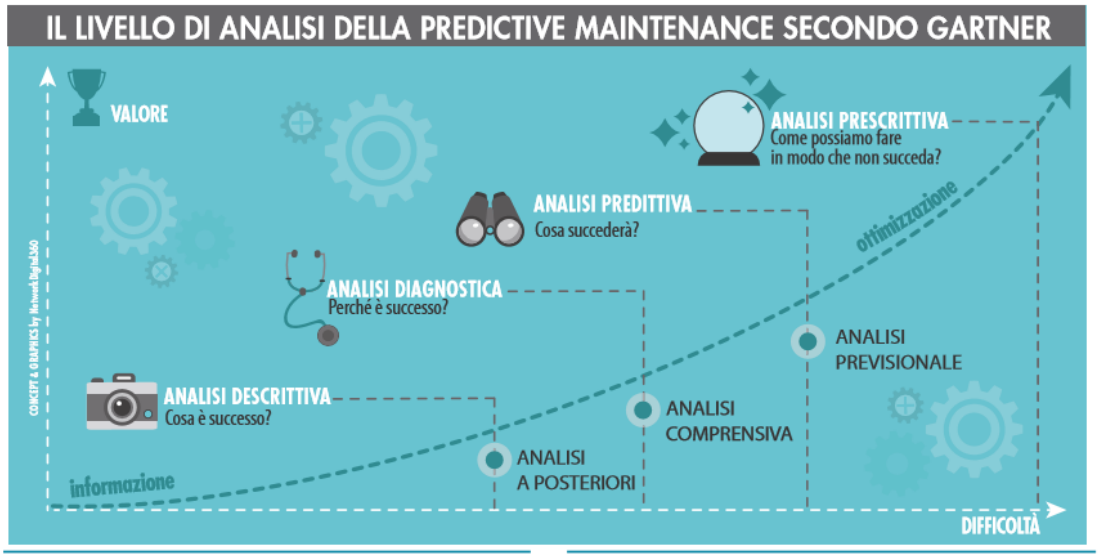


Figura 2.7: Il livello di analisi della predictive maintenance [21]

i flussi, selezionando quelli sui quali lavorare già in locale. Questa esigenza si sposa con l'aumento della capacità computazionale dell'IoT, con la possibilità di dotare gli apparati IoT di un sistema operativo che permette di disaccoppiare hardware e software in vista di una maggiore multifunzionalità in quanto le varie applicazioni possono funzionare su dispositivi diversi. Si sta anche passando a soluzioni di Edge Computing che stanno andando verso la centralizzazione. Dalla continua interrogazione dei sensori, allo scopo di osservare i dati rilevanti, si arriva ad una logica di tipo event-based con i sensori che comunicano nel momento in cui rilevano cambiamenti significativi.

Tecnologie abilitanti la manutenzione predittiva

Alla base di tutto il percorso della predictive maintenance c'è una forte componente tecnologica. Il punto di partenza è rappresentato dalle macchine connesse tra loro e dai sensori dai quali si generano i dati necessari per le successive analisi, dalla temperatura all'umidità, dalle vibrazioni alla conduttività e molte altre. I dati provenienti dai sensori non bastano ma devono essere integrati, prima di passare al livello successivo, con quelli provenienti da altre fonti come come i Programmable Logic Controller (PLC), i Manufacturing Execution Systems (MES)

o ancora gli Enterprise Resource Planning (ERP) aziendali. Questi elementi vanno interconnessi tra loro attraverso tecnologie come Bluetooth o Wi-fi oppure Rfid o LoRa, provenienti dal mondo IoT. Una volta raccolti tutti dati è necessario che siano implementate soluzioni specifiche capaci di gestire dati non strutturati, tecnologie di machine learning e di intelligenza artificiale. È importante avere un sistema di machine learning in grado di lavorare al meglio e con i giusti algoritmi. I segnali vengono così trasformati in dati, i quali vengono successivamente processati, analizzati e visualizzati ed infine è possibile tradurre in azione quanto ottenuto.

Il ruolo dei dati I dati sono l'elemento centrale in un processo di manutenzione predittiva, purché siano pertinenti, sufficienti e di qualità. È importante capire cosa ci si aspetta dal processo di manutenzione predittiva in modo da selezionare i dati che servono allo scopo. Il numero di record di dati per addestrare i modelli predittivi deve essere adeguato, più sono gli errori o failure a disposizione più accurato sarà il risultato.

Il ruolo del cloud In un progetto di predictive maintenance il Cloud gioca un ruolo fondamentale, come in tutti i progetti di smart manufacturing. Il Cloud è la soluzione più importante per creare una fabbrica connessa, esso garantisce, non solo la scalabilità e la potenza di calcolo necessarie, ma anche tutte le funzioni di analisi dei dati e di machine learning alla base della manutenzione predittiva.

Data ingestion Data ingestion è il processo di importazione, trasferimento, caricamento e processamento dei dati da fonti eterogenee per un successivo utilizzo o per l'arricchimento di database preesistenti. I dati possono essere importati in tempo reale oppure con la modalità back che prevede un'importazione a intervalli di tempo prestabiliti. La fase di data ingestion richiede un'infrastruttura adeguata, in termini di banda, in grado di supportare volumi variabili di dati e dunque di scalare in base all'effettiva necessità.

2.5 Machine learning

Il machine learning è sempre più usato in ambito industriale perché facilita la comprensione di dati e processi complessi, dunque si applica bene a quei contesti

ricchi di grandi quantità di dati e di asset. Esistono due modalità di machine learning, supervisionato e non supervisionato.

2.5.1 Come fondere IoT e machine learning

I vantaggi dati dalla possibilità di conoscere in tempo reale e di prevedere il comportamento di reti, macchinari, output, processi in funzione delle diverse variabili che coinvolgono la filieri devono ancora essere colti in pieno ma è già molto evidente quale direzione sta prendendo il mercato. Nel 2016 a livello globale la Predictive Maintenance aveva un giro di affari pari a 1,6 miliardi di dollari, adesso, secondo un rapporto fatto da Data Bridge Market Research, fino al 2024 il settore conoscerà un'impennata, con una crescita annua del 29%. Questa crescita è dovuta al miglioramento sia della parte hardware che della parte software. Le informazioni prodotte e inviate dagli oggetti interconnessi dell'Internet of Things sono importanti per definire uno status quo. Per poter avere maggiori dettagli per identificare i fenomeni e circoscriverli è fondamentale l'elaborazione dei dati con soluzioni Analytics. Conoscere i meccanismi e le varie dinamiche, risalire alle cause anche più remote per cercare di cambiare il risultato da ottenere può avvenire solo attraverso il Machine Learning, tecniche di acquisizione di dati da più fonti, interne ed esterne che alimentano i sistemi di Intelligenza Artificiale. Essi combinano e confrontano tra loro le ricorrenze e le eccezioni e imparano a determinare, tramite calcoli probabilistici, quale causa comporta quale effetto e quindi fare previsioni sul modo in cui si comporteranno macchinari e prodotti in situazioni reali e simulate prevedendo i malfunzionamenti. [22]

2.5.2 Machine learning in manufacturing

Grazie alla rapida evoluzione nel settore degli algoritmi, della capacità di calcolo e una maggiore disponibilità di dati con la diffusione di sensori a basso costo l'apprendimento automatico nell'ambito della produzione crescerà sempre di più. Allo sviluppo del machine learning si affianca quello dei Big Data, entrambi sono degli strumenti molto potenti da usare nei sistemi di produzione e la loro interdisciplinarietà presenta una grande opportunità ma anche un grande rischio perché necessaria una collaborazione tra diverse discipline come l'informatica,

l'ingegneria industriale, la matematica e l'ingegneria elettrica per guidare l'industria verso il progresso. [23]

Capitolo 3

Stato dell'arte: Knowledge Discovery from Data e tecniche di data mining

La gestione e l'elaborazione dei dati consente all'analisi predittiva di scoprire i comportamenti anomali prevedendo guasti imminenti che possono mettere in pericolo la produzione di un impianto. L'analisi predittiva, come si è visto, dà una svolta all'interno del contesto della manutenzione permettendo di trasformare un problema, come un guasto o un'interruzione, in fattore strategico. L'analisi predittiva si serve delle tecniche di data mining e delle architetture dei sistemi cyber-fisici (CPS) per analizzare grandi quantità di dati e ottenere informazioni relative al processo di produzione. Attraverso la fusione tra mondo fisico e mondo virtuale è possibile ottenere maggiore flessibilità nonché un miglioramento generale nella produzione che porta anche dei vantaggi commerciali competitivi. Il processo *"Knowledge Discovery from Data" (KDD)* è un'analisi automatica, esplorativa e di modellizzazione su repository con un'elevata quantità di dati. Si tratta di un processo organizzato per l'identificazione di pattern validi, nuovi, utili e comprensibili da dataset grandi e complessi. Le tecniche di Data Mining (DM) sono alla base del processo KDD perché permettono di applicare gli algoritmi necessari per l'esplorazione dei dati, sviluppare un modello e scoprire pattern sconosciuti. L'elevata disponibilità di dati rende il processo di scoperta della

conoscenza un problema molto importante e necessario. Questo campo negli ultimi anni ha riscontrato grande interesse da parte di ricercatori e professionisti. Sono tanti i metodi e le tecniche disponibili e non c'è un modello migliore di altri, è necessario valutare e analizzare gli approcci sulla base dei casi e degli strumenti che si hanno a disposizione. La proliferazione di elaboratori sempre più performanti ed economici ha contribuito ad aumentare l'uso dei database in molti campi, dalle informazioni sulle vendite, ai registri governativi, dalle informazioni mediche ai dati di pagamento e molti altri. [24] Il KDD sfrutta le metodologie di molti settori come il machine learning, l'intelligenza artificiale, database management, statistica, esperti di sistema e modelli di visualizzazione dei dati. Il campo del KDD e delle tecniche di DM è diventato sempre più necessario per ridurre il divario che si crea nel momento in cui la disponibilità dei dati cresce in modo esponenziale e il livello di elaborazione dell'uomo per tali dati non riesce a seguirne il trend, migliorando in modo costante e quindi più lentamente. Nell'ambito della manutenzione predittiva, la raccolta e l'archiviazione dei dati è fatta attraverso le componenti fisiche, come i sensori o i "data event" ED, che includono le informazioni sulle azioni di manutenzione messe in atto in seguito agli eventi che avvengono sui componenti, ad esempio azione di manutenzione in seguito al guasto di un componente.

3.1 Data Mining

3.1.1 Evoluzione tecnologica del data mining

Dagli anni '60, i database e il settore dell'Information Tecnology (IT) si sono evoluti da sistemi di elaborazione di file primitivi a sistemi di database sofisticati e potenti. La ricerca e lo sviluppo in questo campo, dagli anni '70 in poi, hanno portato grandi progressi evolvendo i sistemi gerarchici e di rete in database relazionali, strutturati in tabelle e con strumenti di modellazione dei dati e metodi di accesso indicizzati. Inoltre, gli utenti sono stati in grado di accedere ai dati in modo agevole e flessibile attraverso linguaggi di query, interfacce utente, ottimizzazione delle query e gestione delle transazioni. Ad esempio, sono stati sviluppati metodi molto efficienti per le transazioni online con query di sola lettura contribuendo a rendere la tecnologia relazionale come uno strumento molto efficiente per la gestione di un elevato ammontare di dati. Dopo l'istituzione di sistemi di gestione dei database,

dagli anni '80 in poi, la tecnologia si è evoluta con sistemi di database sempre più avanzati e complessi, i data warehouse accompagnati dalla nascita del data mining. Questi nuovi sistemi incorporano nuovi e potenti modelli di dati, orientati agli oggetti, relazionali e deduttivi. Anche il progresso nel campo dell'hardware, con apparecchiature potenti ed economiche per la raccolta e l'archiviazione dei dati ha dato un grande impulso al settore dei database. Una delle architetture emergenti nell'ambito dei repository dei dati è il *Data Warehouse*. Questa tecnologia include la pulizia dei dati, l'integrazione e l'elaborazione analitica online (OLAP). Per supportare le analisi e i processi decisionali è necessario integrare questi strumenti con le tecniche di data mining che forniscono classificazione dei dati, clustering, rilevamento delle anomalie e studio delle variazioni dei dati nel tempo. Negli anni '90, con l'avvento del World Wide Web e dei database web based (XML), le diverse tipologie di database interconnessi ed eterogenei hanno acquisito un ruolo cruciale nel settore dell'informazione. Il compito arduo dunque è effettuare un'analisi efficiente ed efficace anche con dati provenienti da fonti eterogenee, integrando le tecnologie per la raccolta dei dati, data mining e la rete. La difficoltà sta nel capire quali sono le informazioni importanti a partire da una grossa quantità di dati. Diventa compito del decisore e degli esperti di dominio avere le giuste intuizioni perché non sempre si hanno a disposizione gli strumenti adatti per estrarre conoscenza. [25]

3.1.2 Cos'è il data mining?

Il data mining è, come si è visto, parte integrante del processo di scoperta della conoscenza noto con il nome di KDD. Il termine data mining viene spesso usato per riferirsi all'intero processo KDD pertanto si può anche definire il data mining come il processo di scoperta di modelli e conoscenza da una grande quantità di dati. In generale, il data mining può essere applicato a qualsiasi tipo di dato significativo per una determinata applicazione. Esistono diversi tipi di dati strutturati:

- **Database data (database management system - DBMS)**: consiste in una collezione di dati interconnessi tra loro (database) e di un set di programmi software per gestirli e accedere ad essi. Un database relazionale è una collezione di tabelle, a ciascuna delle quali è assegnato un nome unico. In ogni tabella ci sono gli attributi (in colonna) e le tuple o record (in riga). Ogni tupla, in una

tabella relazionale, rappresenta un oggetto identificato da una chiave unica e descritto da una serie di attributi. Il modello dei dati in questo caso può essere rappresentato da un diagramma entità-relazione che è costruito sulla base delle relazioni tra i dati.

- **Data warehouse**: è un repository di informazioni raccolte da fonti eterogenee e memorizzate sotto uno schema unificato. I data warehouse sono costruiti attraverso un processo di data cleaning, data integration, data transformation, data loading e aggiornamento periodico. In questo caso i dati vengono archiviati con una prospettiva storica e organizzati attorno a dei soggetti principali. Ad esempio, anziché archiviare una singola transazione di vendita, il data warehouse può archiviare un riepilogo delle transazioni per tipologia di articolo, per ciascun negozio o per ogni area di vendita. Di solito sono modellati con una struttura dati multidimensionale chiamata "data cube" in cui ad ogni dimensione corrisponde un attributo o un set di attributi e ad ogni cella il valore aggregato della misura (somma o conteggio).

- **Database transazionale**: cattura una transazione, ad esempio l'acquisto da parte di un cliente, che ha un codice identificativo unico e una lista di oggetti che danno vita alla transazione. Un database transazionale può avere tabelle aggiuntive, che contengono altre informazioni legate alle transazioni, ad esempio descrizioni o informazioni su venditori o filiali di vendita.

I dati possono assumere forme e strutture versatili e con significati diversi. A questa tipologia appartengono: dati temporali, flussi di dati provenienti da sensori, dati spaziali come quelli delle mappe, dati multimediali come testi, immagini, video e audio, grafici e dati dal web. Queste applicazioni fanno nascere nuove sfide relative a nuovi tipi di conoscenza.

A partire dai vari tipi di dati e repository attraverso le tecniche di data mining possono essere estratti vari modelli in base alle funzionalità. Queste includono: caratterizzazione e discriminazione, regole di associazione, classificazione e regressione, analisi di clustering e analisi degli outliers. In generale queste tecniche possono essere classificate in due categorie: descrittive, che trovano modelli interpretabili dall'uomo per descrivere i dati, o predittive, che usano alcune varibili per predire valori futuri e sconosciuti di altre variabili.

3.1.3 Tecnologie del data mining

La natura interdisciplinare delle ricerche e dello sviluppo del data mining contribuisce in modo significativo al successo del data mining stesso e delle sue ampie applicazioni. Le discipline più importanti che influenzano il data mining sono [25]:

1. La **Statistica**: Un modello statistico è un set di funzioni matematiche che descrivono il comportamento degli oggetti in una classe target in termini di variabili casuali e delle loro distribuzioni di probabilità. I modelli statistici da un lato possono essere il risultato di un'attività di data mining, dall'altro le attività di data mining possono essere costruite sopra i modelli statistici, ad esempio la statistica può essere usata per modellare il rumore dei dati o i valori mancanti. L'inferenza statistica (o statistica predittiva) modella i dati tenendo conto della casualità e dell'incertezza e viene usata per trarre previsioni sul processo che si sta studiando. I test di ipotesi, invece, possono essere sfruttati per valutare i risultati di un modello e capire se questi sono significativi o meno. In generale i metodi statistici hanno un elevata complessità di calcolo e quando vengono applicati su grandi set di dati, distribuiti logicamente e fisicamente, è necessario che siano progettati degli algoritmi in grado di ridurre i tempi e i costi di calcolo.

2. Il **Machine Learning**: Il machine learning indaga su come i computer possano imparare o migliorare le proprie performance attraverso i dati. I problemi classici affrontati dall'apprendimento automatico sono fortemente correlati al data mining. È possibile distinguere due tipologie:

   - *Apprendimento supervisionato*: è sinonimo di classificazione. La supervisione proviene dagli esempi etichettati dei dati di training.
   - *Apprendimento non supervisionato*: è sinonimo di clustering. Il processo di apprendimento in questo caso non è supervisionato, poiché i dati non sono etichettati, ed è usato per scoprire le classi all'interno del set dei dati.
   - *Apprendimento semi-supervisionato*: in questo caso si fa uso di dati etichettati e non etichettati. I dati etichettati sono usati per apprendere le classi del modello invece quelli non etichettati per ridefinire i confini tra le classi.

- *Apprendimento attivo*: è un approccio che consente agli utenti di svolgere un ruolo attivo nel processo di apprendimento. Ad esempio, un esperto di dominio può intervenire per etichettare dei dati senza etichetta. L'obiettivo di questa tipologia di machine learning è ottimizzare il modello attraverso l'uso della conoscenza umana.

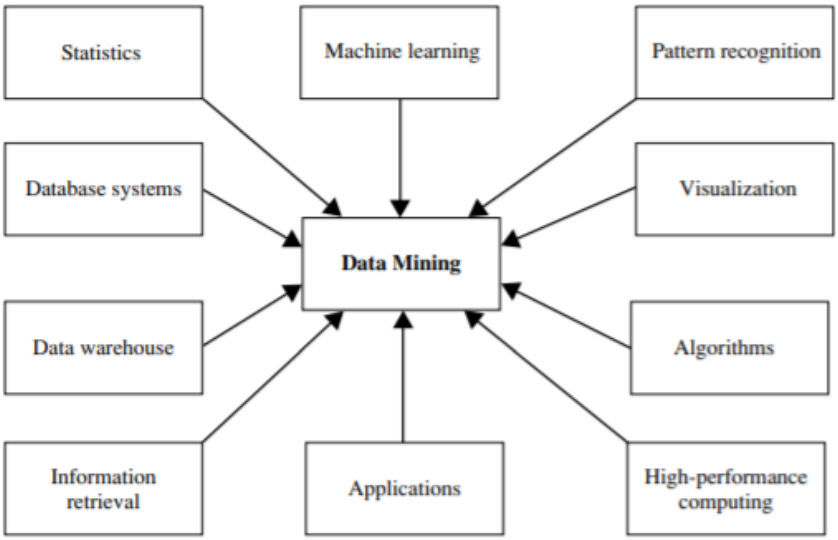


Figura 3.1: Tecnologie per il data mining [25]

Le applicazioni del data mining sono molteplici e la sua diffusione è sempre maggiore. La ricerca sul data mining deve affrontare alcune problematiche legate alla metodologia, all'interazione con l'utente, all'efficienza e alla scalabilità, alla diversità dei dati e alla società. Molti di questi problemi sono stati già affrontati e altri sono ancora in fase di ricerca e continuano a stimolare ulteriori indagini e miglioramenti nell'ambito del data mining.[25]

3.2 Il processo KDD

Il processo di estrazione della conoscenza si basa sul data mining ed è iterativo, interattivo e costituito da più step. Ad ogni step potrebbe essere necessario tornare indietro ai passi precedenti. All'interno del processo non si ha una formula ben definita e non è possibile fare una tassonomia completa di tutte le scelte possibili, pertanto è necessario comprendere quali sono le esigenze e le possibilità in ogni singola fase. L'obiettivo del processo KDD è l'estrazione dai dati di pattern validi, con un certo grado di certezza, nuovi, utili e comprensibili.

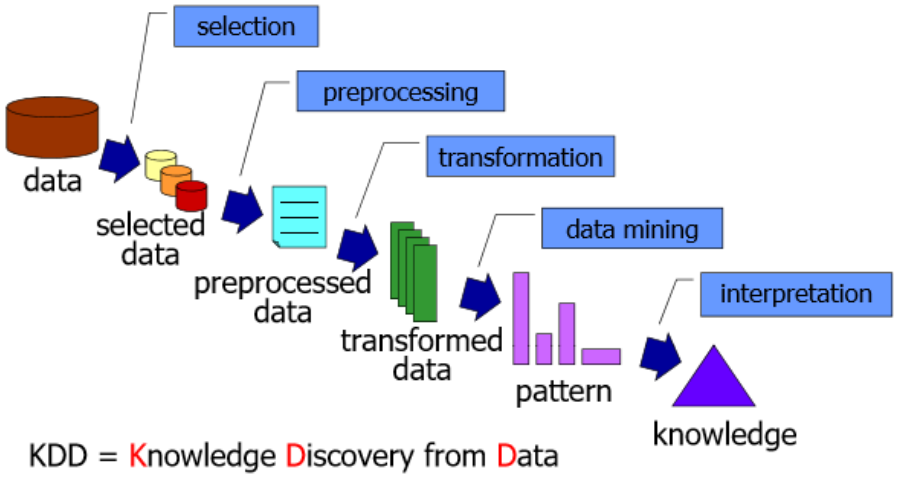


Figura 3.2: Il processo KDD

In una prima fase preliminare è importante, per gli analisti che si occupano del processo, capire qual è il contesto in cui si opera e comprendere e definire gli obiettivi dell'utente finale. Essendo il processo iterativo sarà possibile tornare indietro a questo passaggio per rivedere gli obiettivi iniziali che erano stati prefissati. [26]

3.2.1 Selezione dei dati

Una volta definiti gli obiettivi dell'analisi possono essere selezionati i dati necessari e utili al processo di estrazione della conoscenza. Questa fase include la scoperta dei dati a disposizione e la determinazione di quelli necessari e alla base della costruzione del modello, senza i quali lo studio non potrebbe essere portato a

termine. Da un lato maggiore è il numero di dati a disposizione più dettagliata sarà l'analisi, dall'altro è importante tenere conto della capacità necessaria per poter gestire e organizzare un elevato numero di dati. È fondamentale trovare il giusto compromesso tra i costi di elaborazione e di archiviazione e i benefici relativi ad un'analisi più accurata e dettagliata dei fenomeni. Il processo parte con il miglior set disponibile di dati ma attraverso l'aspetto interattivo e iterativo tutto è in continua espansione ed evoluzione in base ai risultati che si ottengono in ogni fase. [26]

3.2.2 Preprocessing e data cleaning

Questo passo è importante per migliorare l'affidabilità dei dati, vengono cancellati quelli non necessari, gestiti i valori mancanti ed eliminati i valori anomali e il rumore. Spesso la mancanza del dato è dovuta a cause diverse legate al processo di "data entry", ad esempio malfunzionamento degli strumenti utilizzati nel processo di raccolta dati oppure mancato inserimento di un dato perché ritenuto non importante. I dati rumorosi possono invece essere causati da errori di trasmissione di rete o da limitazioni tecnologiche. La fase di preprocessing può comportare l'uso di metodi statistici complessi o di tecniche di data mining, talvolta può essere pure molto dispendiosa e occupare una parte importante dell'intero processo KDD. Se, ad esempio, un determinato attributo presenta molti "missing value" può essere considerato poco affidabile e, di conseguenza, può nascere un nuovo obiettivo per un algoritmo di data mining supervisionato: la previsione dei valori mancanti. Per l'individuazione del rumore possono essere usate soluzioni di data mining come il clustering o la regressione. [26] Alcuni metodi di gestione dei valori mancanti prevedono le seguenti tecniche:

1. Ignorare il record contenente il valore mancante: questo di solito viene fatto quando manca ad esempio l'etichetta di classe e la tecnica di data mining usata è la classificazione. È un metodo poco efficace se la tupla non contiene un numero elevato di valori mancanti in quanto gli altri attributi presenti avrebbero potuto essere utili nel corso delle analisi.

2. Inserire il valore mancante manualmente: questa tecnica è molto dispendiosa in termini di tempo se i valori mancanti sono molti.

3. Sostituire il valore mancante con: una costante, una misura di tendenza centrale come media o mediana di quell'attributo, il valore più probabile che potrebbe avere determinato attraverso la regressione.

Le tecniche per la gestione dei dati rumorosi invece possono essere:

1. Binning: si cerca di uniformare un valore con quelli ad esso circostanti, in particolare i valori dell'attributo vengono ordinati e separati in bin per cui avviene un "livellamento" locale.

2. Regressione: questa tecnica prevede l'adattamento dei dati ad una funzione.

3. Analisi degli outlier: i valori anomali possono essere trovati con il clustering, i valori che non rientrano in nessuno dei gruppi individuati sono considerati outliers.

[25]

3.2.3 Trasformazione dei dati

In questa fase i dati grezzi sono trasformati in modo da essere resi fruibili per le analisi successive con le tecniche di data mining. I metodi usati in questa porzione del processo riguardano ad esempio la riduzione delle dimensioni, in particolare la features selection consente la riduzione del numero attributi selezionando solo le caratteristiche più importanti o l'estrazione di un campione di record. Un'altra tecnica usata è la trasformazione degli attributi: gli attributi numerici possono essere discretizzati. Questo è un passaggio cruciale per l'intero processo di KDD e, solitamente, è fatto su misura per il progetto in esame. Non sempre è facile ottenere una prima trasformazione corretta, pertanto spesso ci si ritrova costretti a dover effettuarla nuovamente sfruttando i risultati ottenuti, che potrebbero suggerire la strada giusta da seguire. [26] In particolare, per quanto riguarda la data reduction essa viene utilizzata per ottenere una rappresentazione ridotta del set di dati che è più piccola dal punto di vista del volume ma ne mantiene comunque l'integrità. La data reduction si distingue in:

- Dimensionality reduction: si riduce il numero delle variabili o degli attributi da tenere in considerazione per le analisi successive. I metodi includono il wavelet transform, elaborazione di un segnale lineare che trasforma un vettore

attraverso dei coefficienti, o la principal component analysis, che riducono i dati originali in uno spazio più piccolo. La selezione degli attributi, invece, permette l'eliminazione di attributi irrilevanti o ridondanti.

- Numerosity reduction: sostituiscono il volume dei dati originali con forme più piccole e rappresentative. Queste tecniche possono essere parametriche o non parametriche. I metodi parametrici, sono utilizzati per stimare i dati per cui è necessario memorizzare solo i parametri anziché i dati veri e propri. I metodi non parametrici invece includono ad esempio i campionamenti. Il campionamento è una delle tecniche più usate per ottenere un set di dati rappresentativo di dimensioni ridotte, può avvenire con o senza sostituzione oppure stratificato. In quest'ultimo caso si trova un attributo su cui stratificare e si crea un campione proporzionale ai valori assunti.

- Data compression: vengono applicate per ottenere una rappresentazione compressa dei dati. Dai dati compressi è poi possibile tornare ai dati originali senza alcuna perdita di informazione oppure con un'approssimazione

Esistono dunque molte tecniche per ridurre la dimensione dei dati ma, in ogni caso, il tempo impiegato per effettuare queste trasformazioni non deve superare il tempo risparmiato sull'applicazione dei data mining sui dati con dimensioni ridotte. Le strategie di trasformazione dei dati includono, oltre quelle sopracitate, anche:

- Costruzione degli attributi: dove sono costruiti nuovi attributi e aggiunti al set di variabili già esistenti con l'obiettivo di aiutare il processo di data mining.

- Aggregazione: ai dati sono applicate operazioni di aggregazione, ad esempio i dati di vendita giornalieri potrebbero essere aggregati per calcolare l'ammontare mensile o annuale.

- Normalizzazione: in cui i dati degli attributi sono ridimensionali in intervalli di valori più piccoli.

- Discretizzazione: i valori degli attributi numerici continui sono sostituiti da intervalli etichettati o etichette concettuali (ad esempio giovane, adulto, anziano).

La **normalizzazione** è una fase molto importante del processo in quanto le unità di misura influiscono sull'analisi dei dati portando anche a risultati molto diversi tra loro se si passa da un'unità di misura ad un'altra. Per evitare la dipendenza dei risultati dalla scelta dell'unità di misura si può optare per la normalizzazione o standardizzazione dei dati. Questi metodi hanno l'obiettivo di dare agli attributi un peso uguale, ad esempio sono molto utili per gli algoritmi di classificazione o di clustering quando questi si basano sulla misura delle distanze. Esistono diverse tecniche di normalizzazione:

- *Normalizzazione min max*: una trasformazione lineare dei dati originali. In questo caso i valori dei dati sono riportati in un intervallo che va da 0 a 1. Questa tipologia di trasformazione è adatta ai casi in cui non si conosce la distribuzione dei dati oppure si tratta di una distribuzione diversa da quella gaussiana.

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

  Dove x_i è il valore da normalizzare e min e max sono i valori di minimo e di massimo assunti dal campione. In questo tipo di normalizzazione vengono conservate le relazioni tra i valori dei dati originali.

- *Normalizzazione z-score*: i valori di un attributo sono normalizzati sulla base della sua media μ e della sua deviazione standard σ . Questo metodo è utile quando i valori di minimo e di massimo di un attributo sono sconosciuti o quando ci sono outlier che influenzano la normalizzazione min-max. In questo modo è facilitata la gestione dei valori anomali e i dati assumeranno la distribuzione di una normale standardizzata.

  $$z_i = \frac{x_i - \mu}{\sigma}$$

- *Normalizzazione con ridimensionamento decimale*: normalizza i valori spostando il punto dei decimali. Il numero di punti decimali spostati dipende dal massimo in valore assoluto assunto dall'attributo.

  $$v_i' = \frac{v_i}{10}$$

La **discretizzazione** è molto utile quando i valori di un attributo sono di tipo continuo, in questo modo le variabili da continue diventano discrete riducendo la cardinalità del dominio dei valori che possono assumere. Le tecniche di discretizzazione che possono essere adottate sono:

- *Suddivisione dei dati in N intervalli di uguale dimensione*: questo è un metodo facile da implementare ma può essere influenzato dalla presenza di outlier.

- *Suddivisione dei dati in N intervalli con la stessa cardinalità*: è un approccio non incrementale ed è adatto ad essere usato quando i dati sono sparsi e con valori anomali.

- *Clustering*: lavora bene con dati sparsi e usa il clustering monodimensionale come il k-means.

[25]

3.2.4 Data mining

Una volta effettuata la trasformazione dei dati è possibile procedere con l'applicazione degli algoritmi di data mining. La scelta dipende dagli obiettivi del KDD e anche dalle decisioni prese ai passi precedenti. Gli obiettivi della fase di data mining sono due: previsione e descrizione. La previsione avviene attraverso le tecniche supervisionate, invece la descrizione include gli algoritmi non supervisionati e la visualizzazione. La maggior parte delle tecniche di data mining si basa sull'apprendimento induttivo in cui il modello è costruito a partire da una serie di esempi di addestramento e può essere applicato a casi futuri. Decisa la strategia da adottare deve essere selezionato il metodo specifico da utilizzare. Per ogni strategia infatti ci sono diverse possibilità per metterla in atto. Il meta-learning si occupa di spiegare quando un algoritmo è performante e quando non lo è in relazione ad un determinato problema. Questo approccio permette di capire quali sono le condizioni che rendono un algoritmo più appropriato rispetto ad un altro. Dopo aver scelto l'algoritmo specifico esso dovrà essere applicato ai dati a disposizione. Talvolta potrebbe essere necessario far girare più volte l'algoritmo settando parametri diversi in modo da ottenere un risultato il più soddisfacente possibile. [26]

3.2.5 Interpretazione dei risultati ed estrazione della conoscenza

In questa fase si valutano e si interpretano i modelli estratti rispetto agli obiettivi fissati al primo passo. Sui risultati ottenuti dopo l'applicazione dell'algoritmo si valutano gli effetti delle elaborazioni fatte nei primi passaggi del processo di KDD e si ha un feedback complessivo di ciò che è stato rilevato dall'uso del Data Mining. Dopo aver avuto un riscontro su quanto effettuato dal processo si è pronti a sfruttare le conoscenze ottenute e apportare eventualmente modifiche al sistema in cui si opera. Questa fase determina il successo dell'intero processo di estrazione KDD ma spesso si va incontro ad alcune sfide e rischi, ad esempio le strutture dei dati e le condizioni in cui si è operato possono variare nel tempo, alcuni attributi potrebbero non essere più disponibili e il dominio dei dati potrebbe non essere più lo stesso. [26]

3.3 Tecniche di data mining

3.3.1 Classificazione

L'uomo ha una capacità innata nel classificare in categorie tutto ciò che lo circonda, la classificazione manuale è adatta a set di dati piccoli e semplici ma quando si ha a che fare con dati più complessi è richiesto l'intervento di soluzioni automatizzate.

I dati per un'attività di classificazione sono una collezione di istanze o record. Ogni istanza è caratterizzata da una serie di attributi e da un'etichetta di classe. Mentre gli attributi possono essere di qualsiasi tipo, le etichette di classe devono essere categoriche. [27] La classificazione estrae modelli dai dati attraverso i classificatori che ne predicono le etichette di classe. I ricercatori nel campo del machine learning e della statistica hanno proposto numerosi algoritmi di classificazione, volgendo il proprio impegno a sviluppare una classificazione scalabile e in grado di gestire grandi quantità di dati. La classificazione ha numerose applicazioni tra cui rilevamento frodi, marketing mirato ad un target, previsione delle prestazioni, produzione e diagnosi medica [25] Un modello di classificazione è una rappresentazione astratta della relazione tra un insieme di attributi e un'etichetta di classe, può essere rappresentato in diversi modi ma da un punto di vista formale e matematico

si tratta di una funzione che prende come input i valori assunti degli attributi e come output l'etichetta predetta.

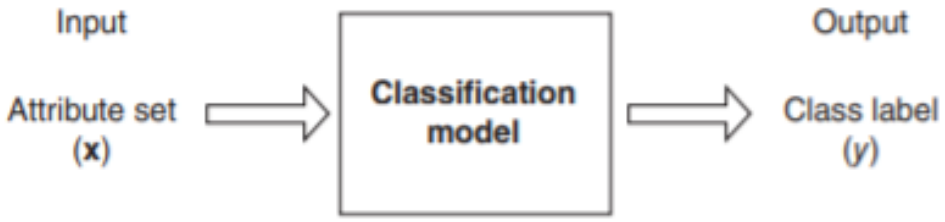


Figura 3.3: Rappresentazione schematica di un processo di classificazione

Come è stato detto in precedenza, i modelli di data mining possono essere usati per scopo predittivo o descrittivo, la classificazione è utile in entrambi i casi. È usata come modello predittivo per classificare istanze non precedentemente etichettate, come modello descrittivo, invece, per identificare le caratteristiche che distinguono le istanze nelle diverse classi. Un classificatore è creato sfruttando un set di istanze conosciuto come training set, contenente attributi valorizzati e dati etichettati. L'approccio sistematico per apprendere un modello di classificazione, dato il set di training, è conosciuto come algoritmo di apprendimento. Il processo in cui l'algoritmo di apprendimento costruisce il modello di classificazione prende il nome di **induzione**. Invece, il processo di predizione delle etichette per istanze appartenenti al dataset di test e non ancora "viste" dal modello è chiamato **deduzione**.

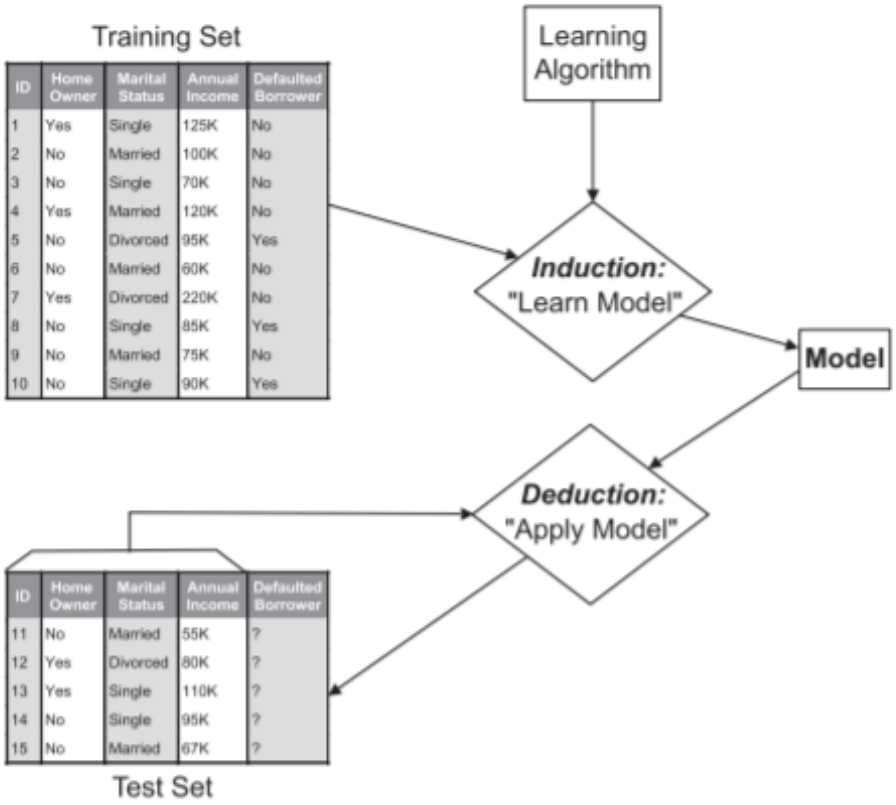


Figura 3.4: Costruzione di un modello di classificazione [27]

Le fasi di induzione e deduzione sono applicate separatamente ai dati. I set di training e di test infatti devono essere indipendenti tra loro in modo che il modello indotto possa prevedere le etichette con precisione senza aver mai "incontrato" precedentemente i dati testati. I modelli di classificazione si prestano molto alla generalizzazione. Le performance possono essere valutate confrontando le etichette vere con quelle predette riassumendole in una matrice di confusione, in forma tabellare. Poiché le etichette dei record di training sono conosciute, si parla di "apprendimento supervisionato". Sebbene la matrice di confusione dia le informazioni necessarie per capire come lavora il modello di classificazione è possibile aggregare il tutto in una metrica con cui poter confrontare tra loro i classificatori. Questa metrica è l'accuratezza che indica la percentuale di record, appartenenti al dataset di test, correttamente assegnati dal classificatore, dunque

numero di predizioni corretto diviso per il totale delle predizioni fatte.

$$Accuratezza = \frac{\text{Numero di predizioni corrette}}{\text{Numero totale di predizioni}}$$

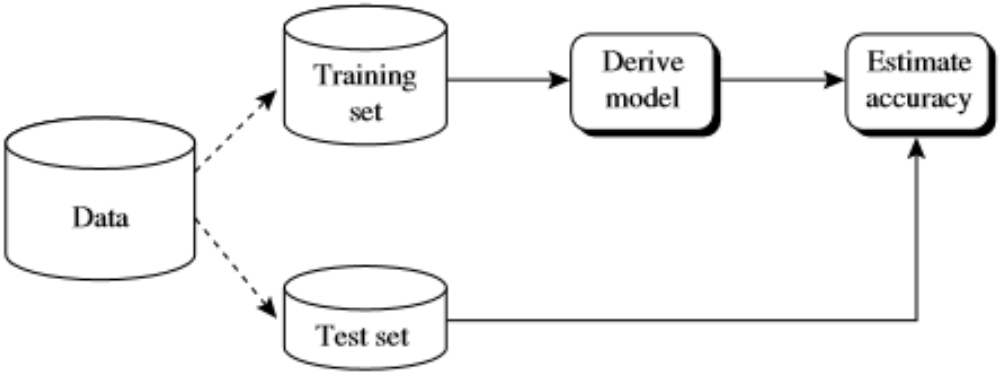


Figura 3.5: Processo di stima del modello e calcolo dell'accuratezza [27]

Altre misure importanti per valutare un classificatore sono quelle della valutazione delle singole classi:

$$Precisione = \frac{\text{Numero di oggetti assegnati correttamente alla classe}}{\text{Numero di oggetti assegnati alla classe}}$$

$$Richiamo = \frac{\text{Numero di oggetti assegnati correttamente alla classe}}{\text{Numero di oggetti appartenenti alla classe}}$$

In aggiunta a queste metriche, i classificatori possono essere confrontati sulla base di alcuni aspetti:

- *Velocità*: si riferisce ai costi computazionali legati alla costruzione e all'uso del classificatore.
- *Robustezza*: l'abilità del classificatore di fare predizioni corrette nonostante valori mancanti o dati rumorosi. Questo aspetto può essere valutato aumentando il grado di rumore e valori mancanti.

- *Scalabilità*: si riferisce alla capacità di costruire il classificatore efficientemente all'aumentare della quantità dei dati.

- *Interpretabilità*: è legata alla comprensione e all'intuitività del classificatore. Si tratta di un aspetto soggettivo che è difficile da valutare.

Come si è visto la classificazione lavora su porzioni diverse di dati, il partizionamento può essere fatto attraverso i seguenti metodi:

- **Holdout**: dato un set di dati viene fatto un partizionamento casuale in due set indipendenti, training set e test set, di solito pari a 2/3 il primo e 1/3 il secondo.

- **Cross validation**: i dati iniziali sono partizionati casualmente in k sottoinsiemi o "fold" di dimensioni simili. Training e test vengono eseguiti k volte. Ad ogni iterazione si ha una diversa combinazione dei sottoinsiemi tra train e test. La differenza rispetto all'holdout è che ogni campione viene usato lo stesso numero di volte per l'allenamento e una volta come test. Inoltre la stima dell'accuratezza è il numero complessivo di classificazioni corrette avvenute durante le iterazioni, diviso per il numero totale di tuple nei dati iniziali, in questo modo si può ottenere una distribuzione dei tassi di errore di test calcolata per ogni partizionamento. Di solito si applica una convalida incrociata stratificata con k=10 per la stima dell'accuratezza, questo metodo serve a garantire che la distribuzione delle classi sia proporzionale a quella dei dati complessivi.

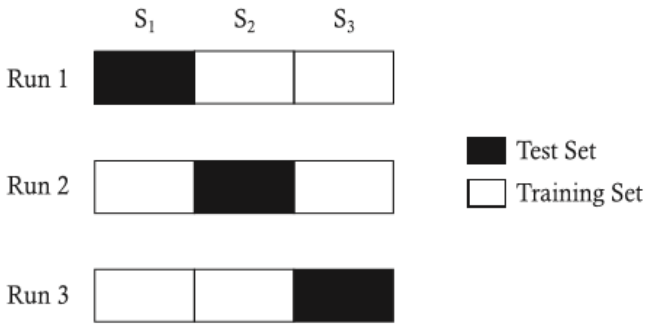


Figura 3.6: Esempio di Cross Validation [27]

Esistono diverse tecniche di classificazione, la scelta deriva dalla strategia che si vuole adottare e dai dati a disposizione.

Decision tree

Un albero decisionale è una struttura simile ad un diagramma di flusso, in cui ciascun nodo interno rappresenta un test su un attributo, ogni ramo è il risultato del test e ogni nodo foglia contiene l'etichetta di classe. Il nodo alla sommità dell'albero è il nodo principale o nodo radice.

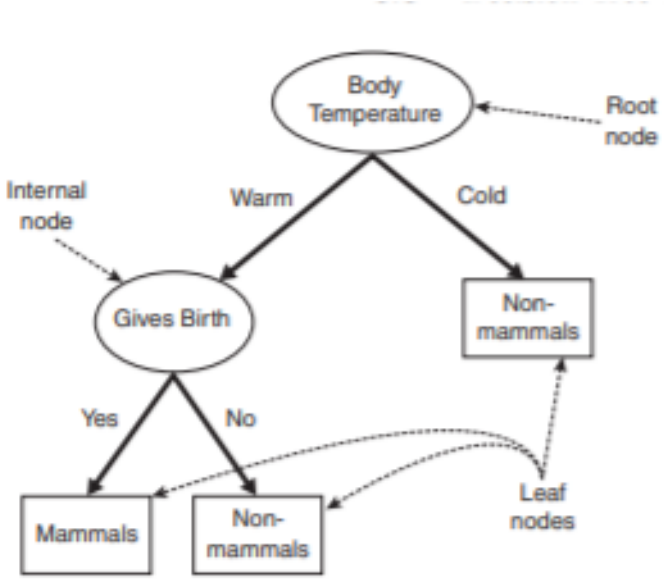
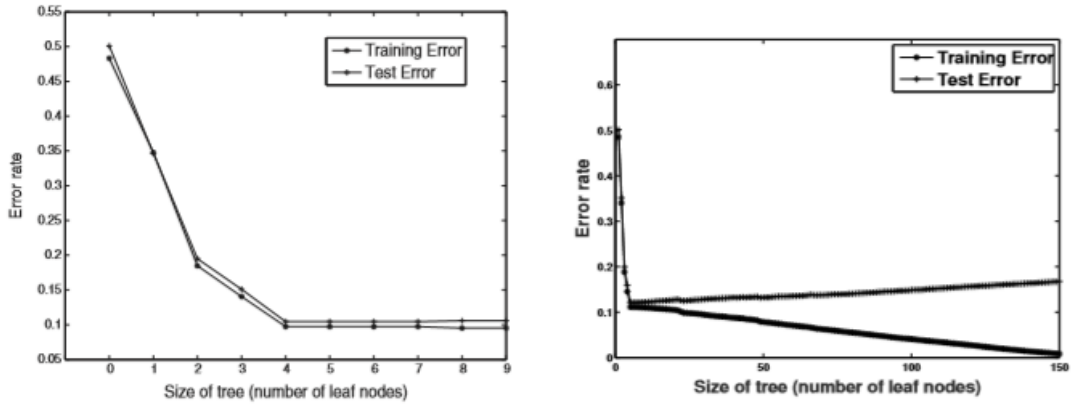


Figura 3.7: Decision tree per un problema di classificazione [27]

Dato un record per il quale non è conosciuta l'etichetta di classe, i valori dei suoi attributi sono testati sul decision tree. In questo modo si traccia un percorso dalla radice al nodo foglia, che contiene la classe prevista. L'albero decisionale può essere facilmente convertito in regole di classificazione.

La costruzione del classificatore del decision tree non richiede conoscenze di dominio o impostazione di parametri, dunque è adatto alla scoperta della conoscenza esplorativa. Gli alberi decisionali possono gestire anche dati multidimensionali e la loro rappresentazione è molto intuitiva e facilmente comprensibile dall'uomo. Le fasi di apprendimento sono semplici e veloci e in generale il modello ha una buona precisione.

È importante la scelta del criterio di split in quanto ci dice quale attributo testare in un determinato nodo con l'obiettivo di separare al meglio i record nelle classi. Il criterio di split ci dice quali rami crescono dal nodo in questione rispetto ai risultati del test scelto. La scelta del criterio dipende da quale delle partizioni risultanti si ottengono rami il più puri possibile. Una partizione è pura se tutte le tuple in essa contenute appartengono alla stessa classe. L'algoritmo di base si ferma solo quando le istanze di training associate al nodo sono della stessa classe e quindi le foglie sono pure. Anche se basta questo per fermare l'algoritmo possono esserci più ragioni per interromperlo prima di arrivare alle foglie pure. Uno dei fenomeni che riguardano un modello che arriva fino in profondità e si adatta perfettamente ai dati di training è l'overfitting, cioè quando il modello è troppo preciso e non è in grado di generalizzare il problema.



(a) Varying tree size from 1 to 8. (b) Varying tree size from 1 to 150.

Figura 3.8: Effetto dell'overfitting all'aumentare della dimensione dell'albero [27]

Gran parte delle performance dipende anche dai dati che si hanno a disposizione, sono tanti gli ambiti applicativi i quali spaziano dalla medicina alla produzione, dall'analisi finanziaria all'astrologia e biologia molecolare. Spesso questa tipologia di classificatori è sfruttata per indurre delle regole commerciali.

Random Forest

La foresta casuale costruisce un insieme di alberi decisionali e li unisce per ottenere una previsione più accurata e stabile. In questo caso si parla di apprendimento di insieme, si uniscono più volte diversi tipi di algoritmo o lo stesso algoritmo per creare un algoritmo di previsione più potente.

Uno dei maggiori vantaggi è che questo algoritmo può essere usato sia per problemi di classificazione che di regressione. Nella fase di training si costruisce una serie di alberi decisionali che dà come output la classe assegnata ovvero la "moda" delle classi degli alberi individuali. In questo modo si corregge il problema dell'overfitting presente negli alberi decisionali. Le random forest sono un modo per calcolare la media di più alberi decisionali profondi, addestrati su diverse sezioni del set di training per ridurre la varianza. Si perde così l'interpretabilità ma si guadagna in termini di prestazioni del modello finale. Mentre le previsioni di un singolo albero sono molto sensibili al rumore, la random forest non lo è purché gli alberi non siano correlati tra loro. Il campionamento bootstrap permette di eliminare la correlazione tra gli alberi, anziché allenare tutti gli alberi sullo stesso set di training, permette di farlo su campioni diversi del set. La scelta del numero di alberi dipende dalle dimensioni e dalla natura dei dati e il numero ottimale può essere trovato con la cross-validation. [28]

Il generale l'algoritmo "random forest" comporta poca distorsione perché esistono più alberi e ogni albero viene addestrato su un sottoinsieme di dati. Si tratta di un algoritmo stabile, se arrivano nuovi dati esso non viene influenzato, si può avere impatto sul singolo albero ma non su tutti gli alberi. Inoltre, funziona bene sia con attributi numerici che categorici e non risente dei valori mancanti. Purtroppo, nonostante sia uno strumento molto valido, la random forest è molto complessa e richiede molte risorse da un punto di vista computazionale e temporale. [29]

Classificazione bayesiana

I classificatori Bayesiani sono classificatori statistici. Essi possono predire la probabilità che una tupla appartenga ad una determinata classe. La classificazione bayesiana si basa sul teorema di Bayes. Uno degli algoritmi di questa tipologia è il classificatore Naive Bayesian, le cui performance sono confrontabili con il Decision Tree e le Reti Neurali. Questa tipologia di classificatori ha elevati valori

di accuratezza e velocità quando sono applicati a database di grandi dimensioni. L'assunzione su cui si basa il classificatore è che l'effetto del valore di un attributo su una determinata classe è indipendente rispetto ai valori degli altri attributi.

Dato un set di dati rappresentati da una serie di attributi e a cui sono associate delle etichette il classificatore cercherà di predire l'appartenenza ad una classe basandosi sulla maggiore probabilità a posteriori, condizionata ai valori degli attributi. Ogni attributo e etichetta di classe si considerano come variabili casuali. Da un record con un set di attributi appartenenti al vettore X, l'obiettivo è predire la classe Y, in particolare si vuole trovare il valore di Y che massimizzi la $P(Y|X_1, X_2, \dots)$. Questa probabilità può essere calcolata assumendo l'indipendenza delle variabili casuali. Nel caso in cui gli attributi siano continui è opportuno effettuare una discretizzazione del dominio, stimare la funzione di densità di probabilità, assumendo che l'attributo segua una distribuzione normale, e usare media e deviazione standard per calcolare la probabilità condizionata. Questo tipo di classificazione è robusta alla presenza di valori anomali isolati e può maneggiare agevolmente i valori mancanti ignorando l'istanza durante i calcoli della stima della probabilità. Il Naive Bayesian è robusto con gli attributi irrilevanti ma la presenza di attributi ridondanti o correlati tra loro viola l'assunzione del teorema di Bayes sull'indipendenza reciproca tra gli attributi. Inoltre è di facile interpretazione in quanto la probabilità da una spiegazione sul perché un dato è assegnato ad una classe. Si tratta infine di un modello incrementale che si aggiorna in automatico con l'arrivo di nuovi dati.

Reti neurali

Gli esseri umani hanno la capacità di identificare modelli all'interno delle informazioni accessibili con un livello elevato di accuratezza. Il sistema nervoso umano è costituito da miliardi di neuroni che elaborano collettivamente l'input ricevuto dagli organi sensoriali, prendono l'informazione e decidono cosa fare in risposta all'input. [30] L'algoritmo delle reti neurali artificiali è ispirato alla struttura del sistema nervoso dell'uomo, i neuroni sono le unità di elaborazione invece le sinapsi sono le connessioni della rete. [27]

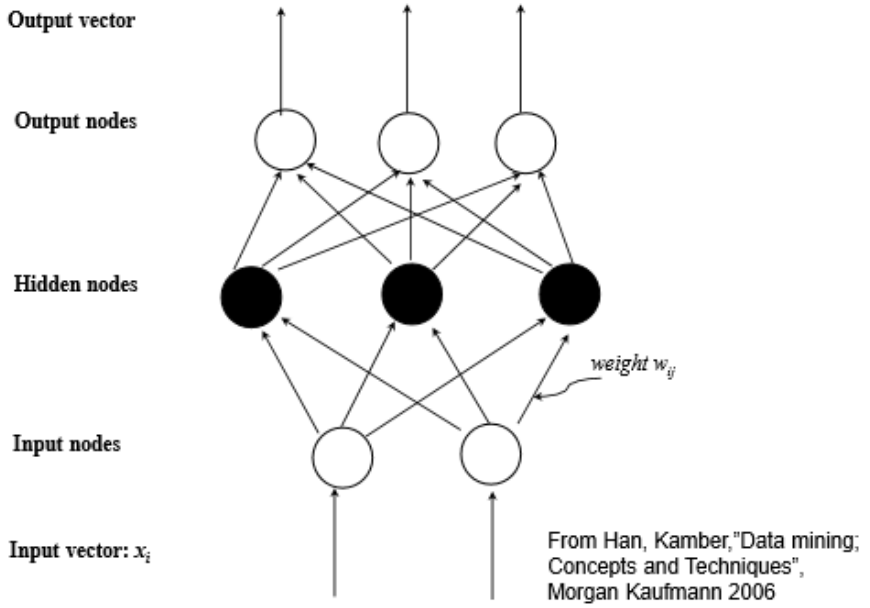


Figura 3.9: Esempio reti neurali [25]

Per ogni nodo sono definiti un set di pesi e un valore di offset che forniscono un'accuratezza elevata ai dati di training su cui viene applicato un approccio iterativo. Inizialmente sono assegnati dei valori casuali ai pesi e agli offset. Per ogni neurone, si calcola il risultato applicando i pesi, gli offset e la funzione di attivazione di un'istanza. Nelle reti neurali artificiali si hanno più nodi collegati in rete: si ha uno strato di input, uno o più livelli nascosti e uno di output. Una rete neurale viene eseguita in due fasi:

- Fase forward: in cui ogni nodo riceve dal nodo di input dei valori che multiplica per i pesi e su cui applica una funzione di attivazione e si va poi avanti fino ad arrivare al nodo di output.

- Fase backward: si confronta l'output ottenuto con quello che ci si aspettava e si valuta l'errore, a questo punto si cerca di minimizzarlo aggiornando i pesi e gli offset per ogni neurone.

Le reti neurali sono un algoritmo molto accurato, robusto al rumore e agli outlier, supporta sia output discreti che continui ed è efficiente durante la classificazione. Tuttavia presenta alcuni punti a sfavore tra cui elevati tempi di elaborazione e

realizzazione di un modello non interpretabile, non è possibile sfruttare la conoscenza degli esperti di dominio.

Support Vector Machine

Il metodo SVM è usato per la classificazione di dati lineari e non lineari. Inizialmente fu introdotto negli anni '60 e in seguito fu perfezionato negli anni '90. Nel caso di dati separabili linearmente in due dimensioni, l'algoritmo tenta di trovare un limite, tra le infinite linee possibili, che divida i dati in modo da ridurre al minimo l'errore di classificazione. A livello tridimensionale invece, si vuole trovare l'iperpiano massimo marginale, cioè che ha il margine massimo dai punti più vicini di ogni classe al margine di decisione. I punti in questione sono chiamati vettori di supporto.[27]

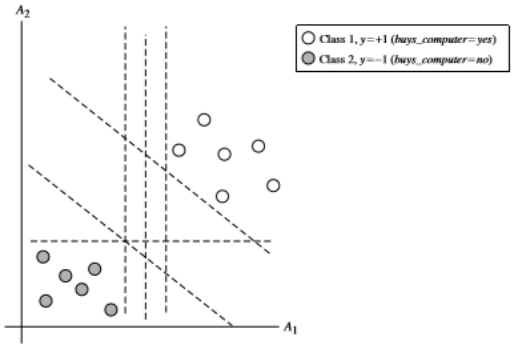


Figura 3.10: SVM dati separabili linearmente [27]

Se i dati non sono separabili linearmente non è possibile trovare una linea retta che separa i dati. A questo punto i dati originali di input sono trasformati in un spazio a più dimensioni usando una mappatura non lineare. Una volta avvenuta la trasformazione il passo successivo prevede di cercare un iperpiano di separazione lineare nel nuovo spazio.

K-Nearest Neighbor

L'algoritmo K-nearest neighbors è una tipologia di algoritmo di apprendimento automatico supervisionato molto semplice da implementare ma che svolge compiti di classificazione piuttosto complessi. Esso calcola la distanza di un nuovo punto rispetto agli altri. Esistono diversi tipi di distanza, come quella Euclidea o Manhattan.

A differenza degli altri algoritmi, il KNN, per ogni record, analizza il dataset di training per vedere quali sono i k elementi simili ad esso. Il parametro da impostare è K, con un K piccolo si ha un breve tempo di classificazione, con K più alto si ottiene un modello più accurato, pertanto è importante trovare il giusto valore di trade off. A partire dal valore di K e dal tipo di distanza, si calcola la distanza tra i record di training, si identificano i K vicini e si usa per il punto in questione l'etichetta di classe del "vicino più vicino".

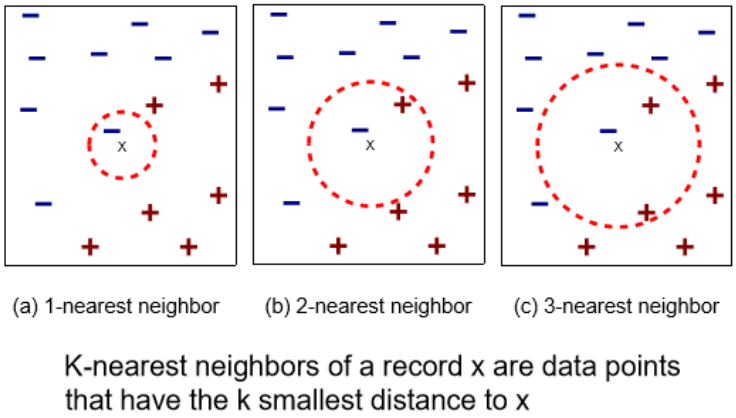


Figura 3.11: Rappresentazione dei vicini "più vicini"

Questa tipologia di algoritmo non funziona bene con un dataset di dimensioni elevate in quanto sarebbe molto dispendioso in termini di tempo calcolare le distanze tra tutte le coppie di record. Inoltre non è adatto a dati con attributi categorici in quanto potrebbe essere difficile calcolarne la distanza.[31]

3.3.2 Regole di associazione

Molti settori di business accumulano grandi quantità di dati dalle loro operazioni quotidiane. Ad esempio, si parla di transazioni quando avviene un qualsiasi acquisto in un negozio. Ad ogni transazione si fa corrispondere una riga della tabella, a cui viene assegnato un codice identificativo univoco. I venditori al dettaglio sono molto interessati a questa tipologia di dati per comprendere il comportamento dei propri clienti e supportare le attività di marketing, gestione dell'inventario e gestione delle relazioni con i clienti. Questo approccio permette dunque di scoprire interessanti relazioni nascoste nei grandi set di dati. Tali relazioni possono essere

rappresentate sotto forma di insiemi di elementi detti anche "itemset frequenti" o regole di associazione. L'analisi delle associazioni può essere applicata anche in altri campi come la bioinformatica, la diagnosi medica, il web mining e l'analisi scientifica dei dati. Questi modelli possono presentare alcune problematiche da un punto di vista computazionale e inoltre alcuni modelli possono emergere a caso, quindi alcuni sono considerati più interessanti rispetto ad altri. Gli elementi coinvolti nelle regole di associazione sono:

- **Itemset**: un insieme di uno o più oggetti
- **Supporto**: la percentuale delle transazioni che contengono un determinato itemset.
$$s(X) = \frac{\sigma(X)}{N}$$
- **Itemset frequente**: un itemest il cui supporto è maggiore o uguale alla soglia di supporto minimo indicata.

La regola di associazione è definita come un'espressione di implicazione nella forma

$$X \Rightarrow Y$$

dove X e Y sono degli itemset. La forza di una regola di associazione può essere misurata in termini di:

- Supporto (s): la frazione delle transazioni sul totale che contengono sia X che Y.
- Confidenza (c): misura quanto spesso un oggetto Y appare nelle transazioni che contengono X.

Dato un set di transazioni T, l'obiettivo delle regole di associazione è trovare tutte le regole che superano le soglie di supporto minimo e confidenza minima. Queste metriche sono importanti perché, ad esempio, da un punto di vista aziendale è improbabile che una regola con supporto basso sia interessante. La confidenza misura l'affidabilità dell'inferenza fatta da una regola e fornisce anche una stima della probabilità condizionale di Y dato X.

I risultati di queste analisi devono essere interpretati con cautela in quanto l'inferenza non sempre è dovuta ad un'effettiva causalità. Tra i punti di forza delle regole di associazione troviamo:

1. Interpretabilità del modello;
2. Un'accuratezza migliore rispetto a quella ottenuta con l'albero decisionale;
3. Una classificazione più efficiente;
4. Buona scalabilità;
5. La mancanza di alcuni dati non influenza il modello;

Tra i punti di debolezza invece:

1. La generazione delle regole potrebbe essere molto lenta, questo fattore dipende dalla scelta del supporto;
2. Poca scalabilità dal punto di vista del numero di attributi;

È interessante notare che spesso le regole di associazione sono usate per effettuare la classificazione. Si parla di classificatori associativi basati sulle regole di associazione. In questo caso si dirà che X implica una classe, cioè le regole avranno nella testa l'etichetta di classe. Si estraggono gli itemset frequenti e si definisce il criterio di ordinamento delle regole sulla base di supporto e confidenza. Ad esempio se si usa l'albero decisionale, le regole si estraggono dopo aver costruito l'albero. Ogni percorso dell'albero può essere rappresentato da una regola ed è indipendente dagli altri. Le performance dei classificatori basati sulle regole di associazione sono buone e confrontabili con quelle degli alberi decisionali.[27]

3.3.3 Clustering

L'analisi dei cluster divide i dati in gruppi (cluster) significativi e utili per rappresentare la struttura naturale dei dati. Queste tecniche di data mining hanno da sempre svolto un ruolo molto importante in vari ambiti: psicologia e altre scienze sociali, biologia, statistica, recupero delle informazioni e apprendimento automatico. [27] Il clustering è il partizionamento di un insieme di dati in sottoinsiemi. Ogni sottoinsieme è un cluster e gli elementi al suo interno devono essere simili tra loro ma dissimili dagli elementi contenuti negli altri cluster. Il partizionamento è eseguito dall'algoritmo stesso e non manualmente, dunque tale tecnica è utile se si vogliono scoprire gruppi di dati che prima non si conoscevano.

Nell'ambito del data mining l'analisi dei cluster può essere sfruttata come strumento autonomo per ottenere informazioni dettagliate sulla distribuzione dei dati, osservare le caratteristiche su ciascun cluster e concentrarsi su un particolare sottoinsieme per ulteriori analisi. Oppure può essere usata come strumento di preelaborazione per altri algoritmi o di rilevamento dei valori anomali.

Poiché un cluster contiene elementi con caratteristiche in comune esso può essere trattato come una classe implicita e dunque il processo di clustering prende il nome di classificazione automatica. L'automazione dell'analisi è un vantaggio rispetto ad altre tecniche di data mining, motivo per cui è diventato un argomento di ricerca molto diffuso. In particolare, gli studi si stanno concentrando sulla scalabilità, sull'efficacia dei metodi e sui tipi di dati. Il clustering è noto come apprendimento non supervisionato poiché non si hanno a disposizione informazioni sull'etichetta di classe.

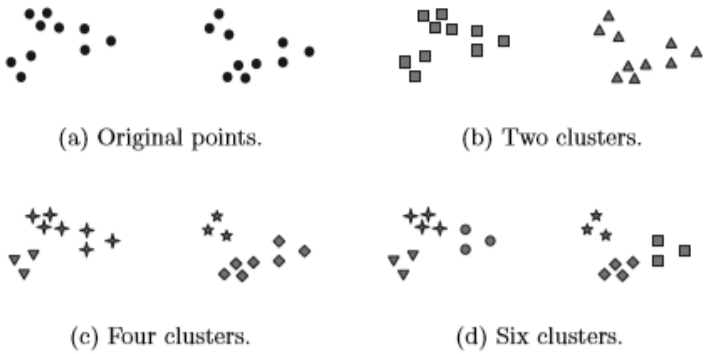


Figura 3.12: Tre modi diversi di clustering per lo stesso set di dati [27]

I requisiti importanti per gli algoritmi di clustering sono: la scalabilità, per gestire dati di dimensioni elevate, capacità di gestire diverse tipologie di attributi, capacità di rilevare cluster con forma arbitraria e diversa da quella sferica poiché i fenomeni naturali non sempre hanno una forma ben definita, evitare l'inserimento di parametri per i quali è difficile conoscere il dominio, capacità di gestire dati rumorosi, clustering incrementali e non sensibili all'ordine di inserimento dei dati, interpretabilità dei risultati e usabilità. Esistono diversi algoritmi di clustering che si distinguono per alcuni aspetti su cui possono essere confrontati:

- **Criteri di partizionamento**: in alcuni metodi gli oggetti sono partizionati

sulla base di una gerarchia tra i cluster, in altri sono tutti allo stesso livello concettuale.

- **Separazione dei cluster**: alcuni metodi prevedono una suddivisione mutualmente esclusiva dei cluster, in altri un oggetto può appartenere a più di un cluster
- **Misura di similarità**: alcuni metodi determinano la similarità tra due oggetti attraverso la distanza reciproca, altri invece si basano sui concetti di densità o contiguità senza considerare la distanza assoluta tra i due elementi
- **Spazio dei cluster**: alcuni metodi cercano i cluster nell'intero spazio a disposizione, ciò è utile con dataset piccoli, altri invece, soprattutto quando i dati sono molti, considerano irrilevanti alcuni attributi trovando dei sottospazi significativi in cui cercare i cluster.

Di seguito sono presentati gli algoritmo di clustering più importanti, ognuno con caratteristiche diverse. [27]

K-means

Il k-means è un tipo di algoritmo partizionale. Questo è il modo più semplice per organizzare i dati separandoli in un set di insiemi esclusivi. Si suppone di avere un set iniziale di dati contenuti nello spazio euclideo. Il metodo distribuisce gli oggetti in k cluster, che rappresenta spesso un numero noto a priori. La funzione obiettivo del modello prevede di valutarne la bontà attraverso la qualità del partizionamento, essa aumenta se aumenta la somiglianza intra cluster e diminuisce quella inter cluster. L'algoritmo del k-means è una tecnica di partizionamento basata sui centroidi, concettualmente i punti centrali all'interno di un cluster. Tali punti possono rappresentare la media o il medoide di quelli assegnati al cluster. Tra ogni punto del gruppo e il suo centroide corrispondente viene calcolata la distanza, da cui si calcola la qualità del cluster attraverso la somma degli errori quadratici (SSE). L'SSE dice quanto i cluster risultano coesi e separati tra loro. L'ottimizzazione della variazione all'interno del cluster è impegnativa dal punto di vista computazionale. L'algoritmo in un primo passo seleziona casualmente k elementi dal dataset considerandoli come centroidi del cluster. Successivamente ogni oggetto rimanente viene assegnato al cluster a cui è più simile. In questo

caso, la somiglianza si basa distanza euclidea tra l'oggetto stesso e i punti scelti al primo passo. L'algoritmo migliora iterativamente, ad ogni nuova iterazione vengono scelti nuovi centroidi e i punti vengono riassegnati. L'algoritmo si ferma quando l'assegnazione è stabile cioè quando i punti vengono tutti assegnati allo stesso cluster dell'iterazione precedente.

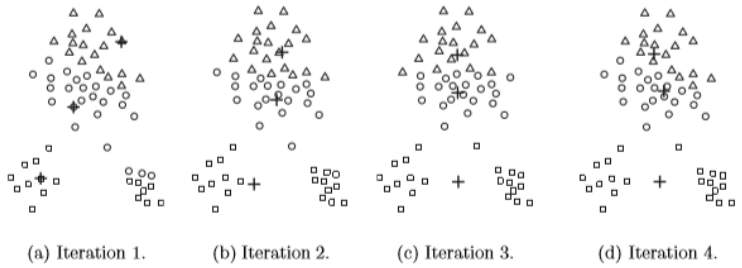


Figura 3.13: Iterazioni algoritmo K-Means [27]

Il k-means è un algoritmo semplice e può essere usato per tantissime tipologie di dati. È abbastanza efficiente anche se spesso sono necessarie più iterazioni. Tuttavia esso non gestisce cluster non globulari o cluster di dimensioni e densità diverse, sebbene possa trovare dei sottocluster se il parametro K specificato è elevato. Un altro dei problemi di questo algoritmo è la difficoltà nella gestione degli outlier, in questi casi è molto utile la fase di preprocessing che ne riduce la presenza.

Clustering gerarchico agglomerativo

Un'altra categoria importante di cluster è quella dei cluster gerarchici. Mentre con i metodi partizionali i cluster sono esclusivi, in alcune situazioni potrebbe essere utile suddividere i dati in diversi livelli gerarchici. La difficoltà che l'algoritmo può incontrare sta nella scelta dei punti di unione e divisione, è una decisione importante perché il processo andrà avanti senza possibilità di annullare quanto fatto in precedenza, una soluzione a questo problema potrebbe essere raggruppare parzialmente i dati usando un'altra tecnica, ad esempio il k-means. Esistono due approcci per la generazione di cluster gerarchici:

- **Agglomerativo (bottom-up)**: inizia considerando i punti come cluster individuali e ad ogni passo unisce le coppie più vicine tra loro. Questo richiede

la definizione di una nozione di prossimità.

- **Divisivo (top-down)**: inizia con un unico cluster contenente tutti i punti e ad ogni step si divide finché non rimangono cluster di punti singoli. Occorre capire quale cluster suddividere e come eseguire la divisione.

Spesso questi cluster vengono rappresentati graficamente con un albero chiamato dendrogramma che mostra sia le relazioni tra cluster e sottocluster sia l'ordine in cui i cluster vengono agglomerati o divisi. La prossimità tra due cluster può essere studiata attraverso una matrice di prossimità, ci sono diversi modi per applicare la strategia di unione tra i cluster:

- Massimo: la prossimità tra due cluster è calcolata come la distanza tra i due punti più distanti appartenenti ai due cluster.
- Media: la prossimità tra i due cluster è la media delle distanze tra tutte le coppie di punti appartenenti ai due cluster.
- Metodo di Ward: la somiglia tra i due cluster si basa sull'aumento dell'errore quadratico medio quando i cluster vengono uniti
- Minimo: la prossimità è la distanza minima calcolata tra le coppie dei punti nei due cluster

I cluster agglomerativi in generale sono di buona qualità, sono usati quando i dati sono strutturati in una tassonomia che può essere rappresentata come una gerarchia e non è necessario avere un particolare numero di cluster a priori, tagliando il dendrogramma in un determinato punto si può ottenere il numero di cluster desiderato. Tuttavia questi algoritmi sono costosi in termini di requisiti computazionali e di archiviazione.

DBSCAN

Il DBSCAN è una tipologia di clustering density-based che distingue le regioni con elevata densità da quelle a bassa densità. Si tratta di un algoritmo semplice ed efficace che sfrutta alcuni concetti importanti legati alla densità dello spazio. Ci sono molti approcci per definire la densità, il più usato è quello center-based per il quale la densità per un particolare punto del dataset è stimata contando il

numero di punti contenuti all'interno di un'area circoscritta da un raggio *Epsilon*, includendo il punto stesso.

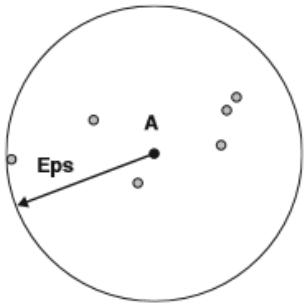


Figura 3.14: Principi di base algoritmo DBSCAN [27]

Il metodo è molto semplice da implementare ma la densità di un punto dipende dal raggio specificato. Ogni punto può essere classificato sulla base della posizione in cui si trova e della densità che la caratterizza:

- Core points: sono denominati così i punti all'interno di un cluster denso, l'area è densa se c'è un numero di punti superiore al parametro *MinPoints* all'interno di un raggio *Eps*, sia *MinPoints* che *Eps* sono parametri specificati dall'utente.
- Border points: un punto che non è core ma si trova nell'area di un core point.
- Noise points: ogni punto che non è né core point né border point.

I punti vicini che si trovano ad una distanza *Eps* sono inseriti nello stesso cluster, i punti noise sono invece scartati. Uno dei problemi legati a questo algoritmo è la determinazione dei parametri *MinPoints* e *Eps*. L'approccio base è vedere il comportamento delle distanze dal k-esimo vicino per ogni punto, con un grafico chiamato *k-dist*.

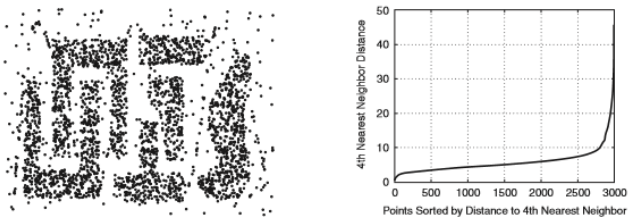


Figura 3.15: Distribuzione dei dati e K-Dist plot [27]

Per i punti che appartengono al cluster il valore di k-dist sarà piccolo, invece per i punti noise k-dist sarà grande. Come si vede in figura 3.15 le distanza per tutti i punti dati rispetto al k-esimo punto sono ordinate in modo crescente e in corrispondenza del "ginocchio" della curva si trova il valore migliore di *Eps*. È importante trovare il giusto valore di K perché se K è troppo piccolo i punti rumorosi potrebbero essere etichettati come cluster, viceversa se k è grande si rischia di considerare rumore anche cluster piccoli. Il DBSCAN si basa sul concetto di densità, è resistente al rumore ed è in grado di gestire cluster di forme e dimensioni variabili, trovando cluster che con il k-means non sarebbe stato possibile trovare. Questo però è un problema quando la densità dei cluster è molto variabile, in particolare quando i dati sono tanti. Inoltre è un algoritmo molto costoso in termini di tempo perché è richiesto il calcolo di tutte le distanze a coppie e ciò diventa complicato nel caso di dataset di grandi dimensioni.

Capitolo 4

Anomaly detection per la deriva dei dati: metodologia e strumenti utilizzati

4.1 Anomaly detection

Dato un processo statistico che genera un set di oggetti, l'outlier è quell'oggetto che devia in modo significativo rispetto agli altri come se fosse stato generato da un meccanismo diverso. Gli outlier sono diversi rispetto ai dati rumorosi. Questi ultimi infatti riguardano errori casuali o la varianza della variabile misurata e solitamente dovrebbero essere rimossi prima dell'outlier detection. Gli outlier sono interessanti da studiare e capire perché potrebbero non essere stati generati dagli stessi meccanismi del resto dei dati. La fase di anomaly o outlier detection è importante per giustificare il motivo per cui i valori anomali sono stati generati da altri meccanismi. Questo spesso si ottiene formulando delle ipotesi sugli altri dati e dimostrando che le anomalie violano in modo significativo tali ipotesi.[25]

4.1.1 Sfide dell'anomaly detection

La qualità dell'anomaly detection dipende fortemente dalla modellizzazione dei dati normali e non anomali. La costruzione di un modello completo per la normalità dei dati è impegnativa in quanto è difficile enumerare tutti i possibili casi di "normalità"

e il confine tra la normalità e l'anomalia spesso non è ben definito. Per questo molte metodologie non assegnano un'etichetta per indicare che il dato sia anomalo o meno, ma un punteggio che quantifica il carattere anomalo di un determinato oggetto.

Un altro problema è la scelta della misura di somiglianza/distanza e quella del modello delle relazioni per descrivere gli oggetti. La specificità del campo applicativo non permette di sviluppare un metodo applicabile in modo universale.

Inoltre, come già detto, i dati rumorosi sono diversi dai dati anomali e la loro presenza abbassa la qualità dei dati creando distorsione e confusione tra i dati normali e quelli anomali. Un punto anomalo può essere confuso come rumoroso e sfuggire all'outlier detection.

In alcuni scenari applicativi potrebbe essere importante capire perché gli oggetti rilevati sono anomali. Un metodo che può essere usato per farlo è quello statistico legato alla probabilità che l'oggetto sia stato generato dallo stesso meccanismo degli altri, minore è questa probabilità, maggiore è quella che l'oggetto sia un valore anomalo.

4.1.2 Anomaly detection nei data stream

Il data streaming, in cui i dati arrivano sotto forma di flusso, è la tipologia di formato usato per i dati raccolti in tempo reale da determinate applicazioni, ad esempio il monitoraggio della produzione tramite sensori. In questi casi tenere sotto controllo i dati anomali o non previsti riveste un ruolo molto importante nel garantire la sicurezza e la stabilità dei sistemi. Molti studi si focalizzano sul rilevamento delle anomalie nei flussi di dati. Lo streaming di dati implica un elevato utilizzo di memoria e gli algoritmi tradizionali che lavorano offline non sono in grado di gestirlo. Inoltre, la maggior parte dei dati rientrano nei valori normali, per cui le anomalie spesso sono rare e difficili da rilevare in quanto necessitano di un modello sofisticato in grado di farlo. Infine, molte delle caratteristiche dei dati possono variare nel tempo dando vita al fenomeno comune del concept drift, da considerare nell'ambito del rilevamento delle anomalie. Una volta avvenuta la deriva del concetto il modello precedente potrebbe non essere più coerente con i nuovi concetti e dunque incapace di gestire la nuova condizione. Comunemente, il modello di anomaly detection degli streaming di dati è visto come la generalizzazione del modello classico di anomaly detection quando i dati aumentano all'infinito.

L'*Isolation Forest* è uno degli algoritmi adatti per il rilevamento delle anomalie nello streaming di dati.[32]

4.1.3 Tecniche per l'anomaly detection

L'anomaly detection o outlier detection è un'attività che ha l'obiettivo di trovare pattern nei dataset non conformi con quelli attesi o che deviano dal comportamento ottimale. È una fase molto importante in tante applicazioni ed esistono vari metodi per effettuarla, ad esempio distance-method based, density-based method, modal-based method e isolation based method. Nel contesto dei data stream alcuni metodi non sono convenienti in quanto hanno una scarsa adattabilità ed estensibilità, elevato costo di aggiornamento e incapacità di rilevare nuove anomalie. Con il rapido sviluppo del settore IT i dati sono diventati di grandi dimensioni. Trovare i punti o i pattern anomali manualmente non è stato più fattibile in molte applicazioni. Come risultato di questo fenomeno sono state proposte nuove metodologie basate sulle tecniche tradizionali e applicate ad alcuni contesti per l'anomaly detection dei data stream. In base alle specifiche applicazioni, le tecniche impiegate variano dalle analisi di semplici serie temporali a quelle complesse e multidimensionali degli streaming di dati.[32] I metodi per il rilevamento degli outliers possono essere classificati in due modi: in base alla presenza di etichette fornite dagli esperti di dominio o in base alle ipotesi relative agli oggetti normali rispetto ai valori anomali. [25] I primi metodi sono:

- **Supervised**: nei metodi supervisionati gli esperti di dominio esaminano ed etichettano i campioni, dunque l'attività di outlier detection si riconduce ad un modello di classificazione: gli esperti possono etichettare i dati normali in modo da considerare tutti gli altri come valori anomali.

- **Semi-supervised**: in alcuni casi è etichettata solo una piccola parte dei dati, i metodi semi supervisionati vengono applicati in queste occasioni. Ad esempio, se sono disponibili alcuni oggetti normali etichettati, il modello si addestra su questi e può essere poi utilizzato per rilevare i valori anomali perché diversi rispetti a quelli che già conosce.

- **Unsupervised**: in alcuni scenari le etichette dei dati non sono disponibili, pertanto si fanno delle ipotesi. I valori normali sono in qualche modo

raggruppati e seguono un modello, invece quelli anomali si trovano in uno spazio lontano rispetto a tutti gli altri. In questi casi potrebbe esserci un alto tasso di falsi positivi quindi i metodi non supervisionati sono meno efficaci dei supervisionati. Spesso questo tipo di outlier detection può essere ricondotta agli algoritmi di clustering.

I secondi invece:

- **Metodi statistici**: si fanno assunzioni sulla normalità dei dati e i dati che non seguono il modello sono considerati outlier.
- **Metodi basati sulla prossimità**: un oggetto è un valore anomalo se i punti ad esso più vicini sono molto lontani nello spazio delle caratteristiche, ovvero se si discosta significativamente rispetto ai suoi vicini.
- **Metodi basati sul clustering**:in questo caso si assume che i valori normali appartengono a cluster grandi e densi mentre gli outlier a cluster piccoli e sparsi o non appartengono a nessun cluster.

Uno dei metodi più usati per i data stream è l'*Isolation Forest* che si basa sull'assunzione che i punti dei dati anomali sono sempre rari e lontani dal centro dei cluster normali. Si tratta di una tecnica efficiente ed efficace che sfrutta la struttura binaria degli alberi e costruisce un insieme di *Isolation Tree* per un determinato dataset attraverso un campionamento randomico. L'idea più importante dell'*Isolation Tree* è quella di trarre vantaggio quando i valori anomali sono pochi e diversi. La costruzione di un singolo albero serve per selezionare casualmente un sottoinsieme dal set di training. L'*Isolation Forest* non utilizza misure di distanza o densità eliminando i costi computazionali in modo significativo ed ha una complessità lineare nel tempo richiedendo poca memoria. Un'altra importante caratteristica è l'*ensemble*, cioè gli alberi vengono raggruppati in una foresta e questo rende l'algoritmo molto più efficiente. [32]

4.1.4 Isolation Forest

Il termine "isolation" indica la separazione di un'istanza rispetto alle altre. Poiché le anomalie sono poche e differenti sono più soggette all'isolamento. [33] L'*Isolation Forest* è un algoritmo di apprendimento non supervisionato usato principalmente

nel campo dell'anomaly detection. Si tratta di un modello che costruisce un insieme di isolation tree multipli creati scegliendo casualmente gli attributi e i valori degli attributi su cui effettuare gli split scegliendoli tra i valori massimi e minimi. In ogni nodo dell'isolation tree le istanze sono divise in due split sulla base degli attributi scelti e dei loro valori. Le istanze anomale sono quegli oggetti i cui attributi hanno valori molto diversi dagli altri e sono più facili da individuare rispetto ai punti normali. Più le anomalie sono vicine al nodo radice più semplice è rilevarle. Per mitigare l'effetto random nel processo di isolation forest si calcola la profondità media delle istanze all'interno degli isolation tree di cui essa è composta e si usa come punteggio di anomalia delle istanze. Più bassa è la profondità, maggiore è la probabilità di avere un'anomalia. [32] In un albero casuale il partizionamento dei dati avviene ripetutamente e in modo ricorsivo fino a quando tutte le istanze non sono state isolate. Il percorso sarà più breve per le anomalie perché i loro valori si distinguono maggiormente dagli altri e si prestano di più ad un partizionamento iniziale. Come insieme di alberi, l'*Isolation Forest* identifica le anomalie come i punti con il percorso più breve e la presenza di più alberi permette di individuare un maggior numero di anomalie. Questo algoritmo funziona meglio con campioni di addestramento piccoli poiché non è necessario isolare i valori normali, anzi dataset di dimensioni elevate riducono la capacità dell'Isolation forest di rilevare le anomalie perché i valori normali possono maggiormente interferire nel processo[33]

Training e test

Il processo di anomaly detection attraverso l'isolation forest è costituito da due fasi. Il primo passo è il training, in cui si costruiscono gli alberi usando i sottocampioni del set di training. Il secondo passaggio consiste nell'applicare gli isolation tree sulle istanze di test e ottenere un punteggio di anomalia per ogni istanza.

- **Fase di training**: gli Isolation Tree sono costruiti in modo ricorsivo partizionando un determinato set di training finché le istanze non sono isolate o l'altezza di un albero non raggiunge un determinato valore. La dimensione del campione controlla quella del set di training. Empiricamente settare questo valore a 256 fornisce un'analisi abbastanza dettagliata per il rilevamento delle anomalie. Un altro valore da specificare in questa fase è il numero di alberi da raggruppare che solitamente è settato a 100.[33]

- **Fase di test**: I test saranno alimentati da una finestra che scorre di dimensione predefinita. Ad ogni test ogni istanza nella finestra è esaminata dall'anomaly detector che determina se il punto è un'anomalia o meno a seconda del punteggio di anomalia ottenuto. Quando sono state completate le istanze all'interno della finestra testata viene acquisito un risultato statistico che può essere sfruttato per il rilevamento della deriva del concetto, se il tasso di anomalie nella finestra è inferiore ad una determinata soglia allora il concept drift non avviene e il modello non sarà cambiato. Viceversa, significa che lo spostamento del concetto è avvenuto e il modello addestrato deve essere modificato e riaddestrato.[33]

4.2 Concept drift

Nell'ambito dell'analisi predittiva e dell'apprendimento automatico, il concept drift (letteralmente: deriva del concetto) indica che le proprietà statistiche della variabile che il modello deve prevedere mutano nel tempo in modi imprevisti e le previsioni diventano sempre meno precise. [34]

Anche un modello ben costruito può subire un peggioramento delle proprie abilità predittive nel corso del tempo. In generale, un modello può decadere in due modi: a causa della deriva dei dati, quando i dati si evolvono nel tempo introducendo potenzialmente una varietà di dati sconosciuta e mai vista prima, oppure a causa della deriva dei concetti, quando cambia l'interpretazione dei dati nel tempo anche se la loro distribuzione non è mutata. [35]

La maggior parte dei sistemi di apprendimento automatico lavora per lotti. Si analizza un set di dati storici e si sviluppa un modello che rispecchia la realtà di quando esso è stato costruito. Tuttavia, il mondo è dinamico e in continua evoluzione e le distribuzioni complesse studiate dal modello stesso variano nel tempo causandone in questo modo un peggioramento delle performance. Questo problema può essere affrontato sviluppando dei meccanismi che sono in grado di rilevare e gestire la deriva dei concetti e dei dati.

Uno dei campi applicativi in cui il concept drift è maggiormente usato è quello dei data stream (o flussi di dati). Un data stream è un set di dati in cui ogni record ha un timestamp, il tempo è un concetto centrale per l'elaborazione del flusso. In quasi tutti i modelli infatti ogni elemento del flusso è associato ad uno

o più timestamp di un determinato dominio temporale che potrebbe indicare, ad esempio, quando l'elemento è stato generato, la validità del suo contenuto, o quando è diventato disponibile per la sua elaborazione.

Il processo che genera il flusso può essere considerato come una variabile casuale χ con gli oggetti o appartenenti al suo dominio. Per le tecniche di classificazione è necessario distinguere una variabile di classe, o etichetta y, appartenete al $dom(Y)$, e x appartenente a $dom(X)$, dove X è una variabile casuale sul vettore dei valori degli attributi. In questo caso χ rappresenta l'unione delle distribuzioni XY e o rappresenta la coppia (x, y). Nell'ambito della classificazione, $P(Y)$ è distribuzione di probabilità a priori sulle etichette di classe e $P(X)$ è quella della probabilità a priori su x. $P(X, Y)$ è la probabilità congiunta sugli oggetti e sulle classi, $P(Y|X)$ è la distribuzione di proababilità condizionata sulle classi e $P(X|Y)$ è la distribuzione di probabilità condizionata su x.

Per fare riferimento alla distribuzione di probabilità in un determinato istante di tempo, si aggiunge la distribuzione di probabilità al tempo t, $P_t(\chi)$

4.2.1 Definizione di Concept Drift

Esistono diverse definizioni di concetto, una di queste lo definisce come un insieme di vettori di valori X, qualsiasi oggetto con un vettore di valori contenuti nell'insieme dei valori di X appartiene al concetto, con una funzione $X \rightarrow Y$. Questo tipo di definizione non consente che oggetti con valori di attributo identici possano appartenere a concetti diversi. Tuttavia, la maggior parte delle applicazioni dell'apprendimento automatico richiede una mappatura molti a molti dai valori X ai valori Y.

Una delle definizioni più usate dà una sfumatura probabilistica alla nozione di concetto. Esso è definito come la combinazione tra la probabilità a priori $P(Y)$ e la probabilità condizionata $P(X|Y)$. Poiché $P(Y)$ e $P(X|Y)$ determinano la distribuzione di probabilità congiunta $P(X, Y)$ allora un concetto può essere definito come:

$$Concetto = P(X, Y)$$

Nel caso in cui non esista un attributo di classe, quando l'apprendimento non è supervisionato allora semplicemente:

$$Concetto = P(\chi)$$

Nel contesto dei data stream i concetti possono cambiare nel tempo, dunque si può definire il concetto in un determinato istante:

$$Concetto = P_t(\chi)$$

Il concept drift si verifica quando cambiano le distribuzioni tra il tempo t e il tempo $t + 1$.

$$P_t(\chi) \neq P_{t+1}(\chi)$$

.

4.2.2 Misure quantitative del drift

Alla base della nozione di concept drift c'è la quantificazione del grado di differenza tra due punti temporali. Questa grandezza prende il nome di *drift magnitude* o *gravità del drift*. Anziché specificare quale misura di distanza dovrebbe essere usata, si può usare una funzione generale che si adatta in base al dominio in cui si sta lavorando:

$$D(t, t + m)$$

Questa funzione ritorna un valore non negativo che indica la differenza tra il tempo t e il tempo $t + m$.

Un'altra misura importante è la *drift duration* cioè la durata della deriva, il tempo trascorso dal momento t quando inizia la deriva e il tempo u quando la deriva termina. [36]

$$Duration_{t-u} = u - t$$

4.2.3 Le fonti del drift

Poiché il concept drift è definito come il cambio della probabilità congiunta $P(X, Y)$ esso può essere scomposto in:

$$P(X, Y) = P_t(X) * P_t(Y|X)$$

A partire da questa formula si possono riconoscere 3 fonti di cambiamento per il drift:

1. **Fonte I**: quando $P_t(X) \neq P_t + 1(X)$ mentre $P_t(y|X) = P_{t+1}(y|X)$, la ricerca si focalizza sul primo termine, poiché il drift su $P_t(X)$ non influenza il limite decisionale ed è considerato come un drift virtuale. Si tratta di una deriva delle caratteristiche dell'istanza.

2. **Fonte II**: quando $P_t(Y|X) \neq P_{t+1}(Y|X)$, questo drift fa modificare il limite decisionale, fa diminuire l'accuratezza dell'apprendimento e prende il nome di actual drift. È la deriva del limite decisionale.

3. **Fonte III**: è un mix tra la Fonte I e la Fonte II, il drift interessa entrambe le probabilità ed entrambi i cambiamenti forniscono informazioni importanti sull'ambiente di apprendimento. Questo è il caso più frequente nel mondo reale.

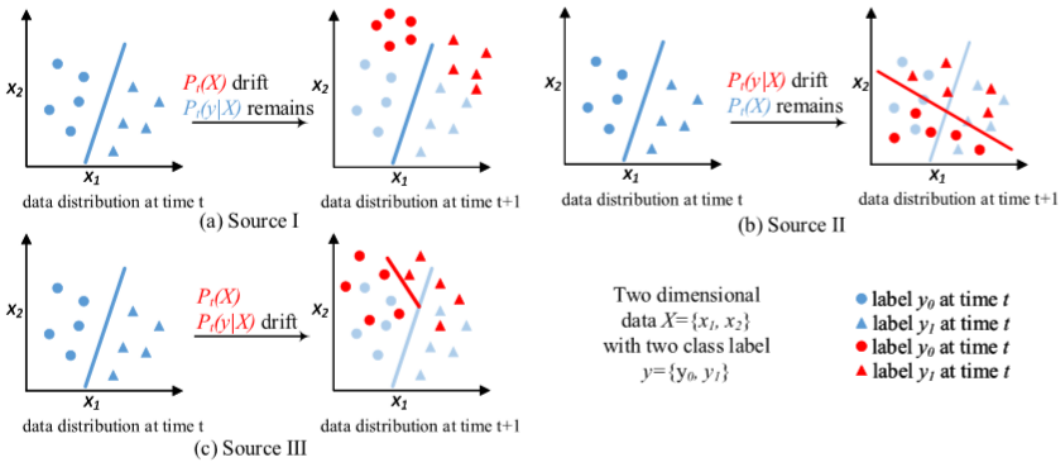


Figura 4.1: Le fonti del concept drift

Dalla figura si nota come queste fonti differiscano nello spazio bidimensionale.[37]

4.2.4 Apprendimento incrementale nel concept drift

In un contesto incrementale ad ogni step t si ha una serie di dati storici etichettati X_1, X_2... Quando arriva una nuova istanza X_{t+1} si vuole predirne l'etichetta Y_{t+1} e per farlo si costruisce un modello che sfrutterà tutti i dati storici o una selezione di essi.

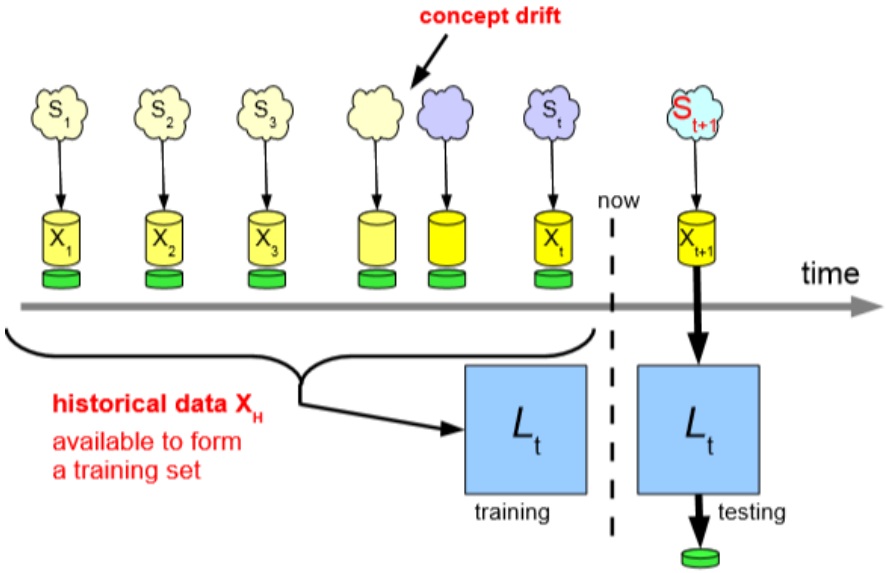


Figura 4.2: Processo incrementale [38]

Allo step successivo, dopo aver effettuato la classificazione o la predizione, l'etichetta Y_{t+1} sarà resa disponibile e l'istanza X_{t+1} farà parte dei dati storici. Ogni istanza X_t è generata da una fonte S_t. Se tutti i dati provengono dalla stessa fonte allora si dice che il concetto è stabile. Se invece per due punti temporali i e j le fonti sono diverse allora si dice che c'è una deriva del concetto. Una deviazione o un rumore casuale non sono "concept drift" perché il quel caso la fonte che genera i dati è la stessa. Il problema della deriva si fonda sull'incertezza del futuro, tutto può essere solamente stimato o previsto ma non c'è alcuna certezza. La stagionalità periodica può essere considerata come un problema di concept drift solo quando essa non si conosce con certezza. [38]

4.2.5 Tipi di cambiamento

Per tipologia di cambiamento si intendono gli schemi di configurazione delle fonti dei dati nel tempo. Il modello più semplice di cambiamento è il *Sudden Drift*, quando al tempo t_0 una fonte S_I è sostituita dalla fonte S_{II}. Un altro tipo di schema è il *Gradual Drift*. Esistono poi due tipologie che possono essere considerate come un mix di queste due. Il primo tipo di *Gradual Drift* si riferisce ad un periodo in cui sia la fonte S_I che la fonte S_{II} sono attive. Con il passare del tempo la probabilità

relativa alla fonte S_I diminuisce e aumenta quella relativa a S_{II}. Un'altra tipologia di drift graduale include più di due fonti. La differenza tra queste fonti è molto piccola e la deriva si nota solo quando si osserva il fenomeno per un periodo di tempo più lungo. Un altro tipo di deriva è chiamato *Reoccurring Concepts*, quando il concetto attivo precedentemente riappare dopo qualche tempo, non si può parlare di stagionalità perché non si sa quando la fonte potrebbe riapparire. Dato un segmento temporale di lunghezza t, le possibili combinazioni delle due fonti S_I e S_{II} e quindi di cambiamenti possibili sono 2^t. Tuttavia si può assumere che il flusso dei dati sia infinito e in questo modo infinito è il numero di configurazioni di cambiamenti possibili. [38]

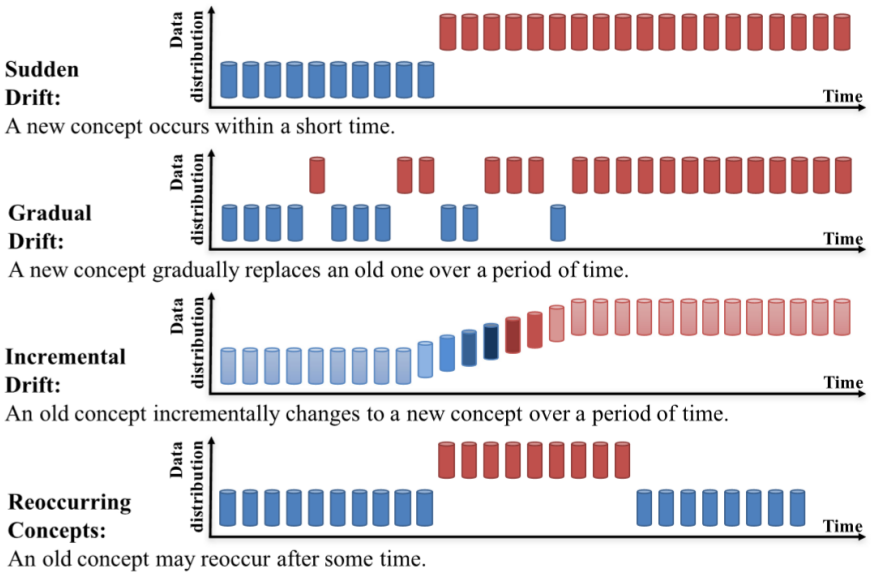


Figura 4.3: Tipologie di concept drift [37]

La deriva, come si è visto, avviene improvvisamente, gradualmente o in modo incrementale. Per dimostrare le differenze tra queste tipologie è stato introdotto il *"concetto intermedio"*. La deriva può non avvenire in corrispondenza di un determinato timestamp ma può durare anche per un lungo periodo, durante il quale avviene la trasformazione da un concetto ad un altro. Il *"concetto intermedio"*, nel caso di deriva incrementale, è un mix tra il concetto iniziale e quello finale. Nel caso di deriva graduale, il *"concetto intermedio"* può essere quello inizio o di fine. [37]

4.3 Framework di ricerca del concept drift

La ricerca convenzionale relativa all'apprendimento automatico è stata rivoluzionata e migliorata con l'introduzione delle tecniche di concept drift nel campo del data science e dell'intelligenza artificiale, in particolare nel data mining. Il concept drift è anche una problematica fortemente legata al contesto dei big data dal momento che in essi è intrinseca l'incertezza dei tipi di dati e delle loro distribuzioni. Il machine learning è costituito da: fase di apprendimento o training e fase di previsione. La ricerca sul concept drift presenta poi tre componenti: il rilevamento della deriva, la sua comprensione e la reazione alla sua esistenza. [37]

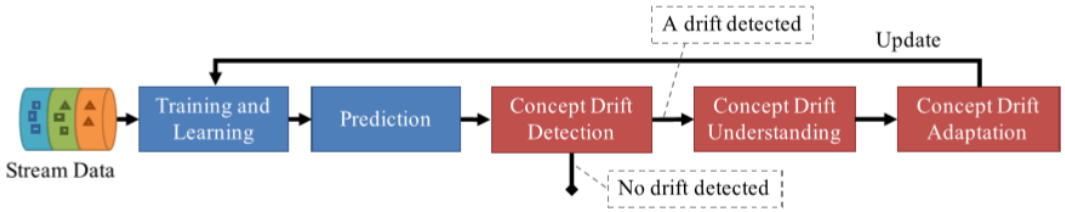


Figura 4.4: Framework concept drift nel machine learning

4.3.1 Processo di rilevamento del concept drift

Ci sono diverse tecniche e meccanismi che consentono di rilevare e quantificare il concept drift, attraverso l'identificazione dei punti di cambiamento o gli intervalli in cui avviene.Una panoramica generale del processo è presentata in figura 4.5.

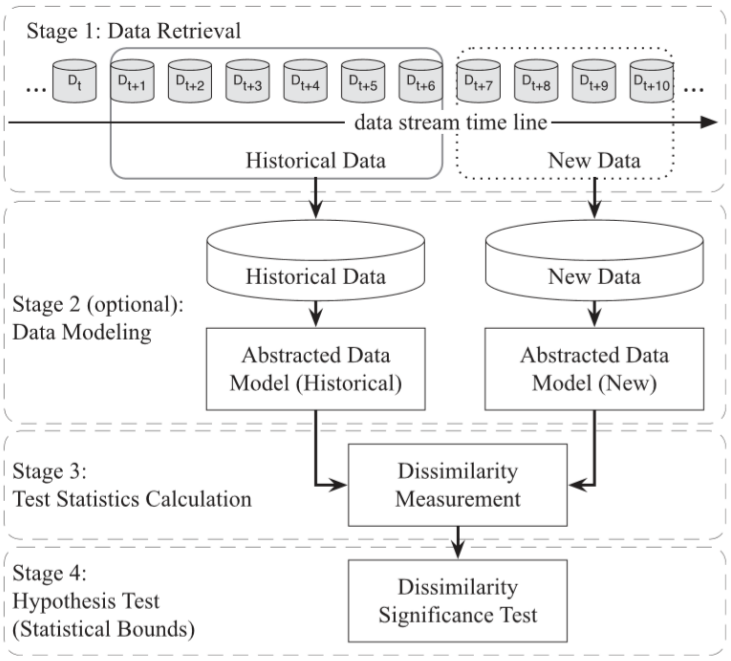


Figura 4.5: Framework generale per il processo di rilevamento del concept drift [37]

Esso è costituito da 4 fasi principali:

1. **Fase di recupero dei dati**: ha l'obiettivo di recuperare blocchi di dati dai data stream che, a differenza delle singole istanze di dato, possono fornire informazioni in più sulla distribuzione complessiva. A partire da questi blocchi è possibile creare degli schemi importanti e significativi per conoscere meglio i dati a disposizione.

2. **Fase di modellazione dei dati**: è una fase opzionale che consente di estrarre le informazioni più sensibili e che incidono maggiormente in caso di deriva. La modellazione riguarda principalmente i processi di riduzione della dimensionalità o la riduzione del campione per facilitare e velocizzare l'archiviazione dei dati.

3. **Fase di calcolo delle statistiche**: è il momento del processo in cui si misura o si stima la differenza tra i nuovi dati e quelli storici. Si quantifica la gravità della deriva e si propongono le statistiche di test per il test di ipotesi. È

una fase molto dispendiosa perché non è semplice definire una misurazione accurata e solida della differenza.

4. **Fase del test di ipotesi**: utilizza un test di ipotesi specifico per valutare la significatività statistica dei risultati ottenuti nella fase 3. Senza la fase 4, le statistiche ottenute nella fase precedente non avrebbero rilevanza. I test di ipotesi permettono di determinare l'intervallo di confidenza della deriva, ovvero, quanto è probabile che il cambiamento sia causato dal concept drift e non dal rumore o dalla distorsione della selezione casuale del campione.

Algoritmi di rilevamento del concept drift

Gli algoritmi di concept drift possono essere classificati in tre categorie in base alle statistiche dei test che applicano. [37]

1. **Error rate-based drift detection**: gli algoritmi di rilevamento della deriva basati sul tasso di errore online sono la categoria più ampia. Se un aumento o una diminuzione del tasso di errore risulta statisticamente significativo, verrà attivato un processo di aggiornamento con l'avviso della deriva. Uno di questi algoritmi è il *Drift Detection Method (DDM)*, che sfrutta una finestra temporale di riferimento. Quando arriva una nuova istanza di dati il DDM rileva se il tasso di errore online complessivo è significativo, in questo caso il DDM inizia a creare un nuovo modello che prenderà il posto del vecchio se si raggiunge il livello di deriva. Il DDM si serve di un classificatore per fare le previsioni. Oltre al DDM esistono altri algoritmi simili.

2. **Data Distribution-based Drift Detection**: la seconda categoria più grande di algoritmi è quella che si basa sulla distribuzione dei dati per rilevare la deriva. Gli algoritmi usano una funzione o una metrica di distanza per quantificare la diversità tra la distribuzione dei dati storici e dei dati nuovi. Se la differenza è significativa allora sarà attivato un processo di aggiornamento del modello di apprendimento. Questo tipo di algoritmi è in grado di identificare accuratamente il tempo di deriva e informazioni sulla sua posizione. Il costo computazionale è elevato ed è necessario predefinire la finestra dei dati storici e quella dei nuovi, solitamente la prima è fissa e la seconda è mobile.

3. **Multiple Hypothesis Test Drift Detection**: le tecniche sono simili a quelle delle due categorie sopracitate. La novità consiste nell'uso di test di ipotesi multiple per rilevare il concept drift in modi diversi. Questa tipologia può essere suddivisa in due gruppi: test di ipotesi multiple parallele e test gerarchici di ipotesi multiple.

4.3.2 Concept drift understanding

L'output della "Drift Detection" è la comprensione del drift che consiste nel trovare informazioni relativamente a quando, come e dove il drift è avvenuto e con quale grado di severità.

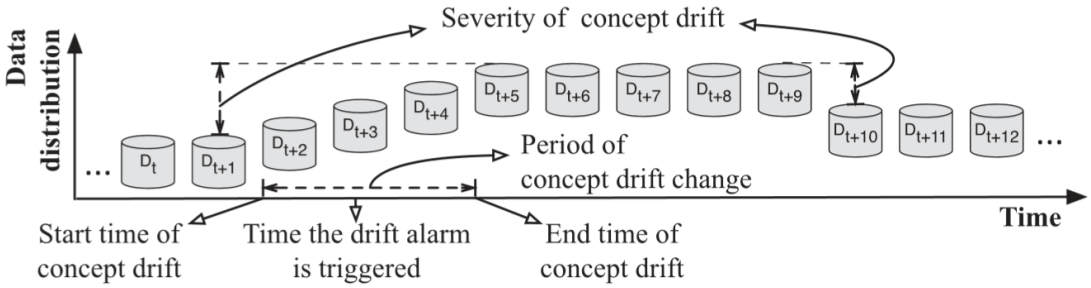


Figura 4.6: Gravità del concept drift [37]

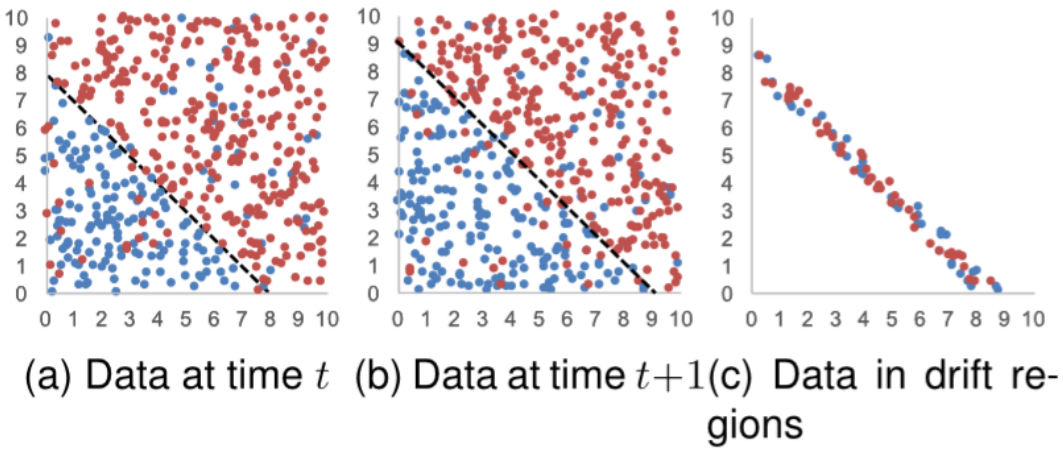


Figura 4.7: Un esempio di regioni di drift [37]

Quando avviene il drift

La funzione di base di rilevamento del drift è identificare il timestamp in cui esso avviene. Riprendendo la definizione di drift: $\exists t : P_t(X, y) \neq P_{t+1}(X, y)$, la variabile t rappresenta l'istante in cui la deriva si presenta. Gli algoritmi e i metodi di rilevamento del concept drift sfruttano un segnale o un allarme che indica se il drift è avvenuto all'istante corrente. Tale segnale è usato anche per iniziare l'adattamento del sistema di apprendimento ad un nuovo concetto. È fondamentale capire con precisione il momento in cui si verifica la deriva perché un ritardo o un falso allarme fanno perdere la tracciabilità dei nuovi concetti. L'allarme di deriva ha una garanzia statistica, cioè si ha una frequenza predefinita per i falsi allarmi. Gli algoritmi di concept drift detection basati sul rilevamento del tasso di errore monitorano il sistema attraverso il controllo statistico dei processi. Gli algoritmi basati sulla distribuzione dei dati invece rilevano il drift quando c'è una differenza statisticamente significativa tra due campionamenti. È importante l'esplorazione dei vari istanti in cui il drift si verifica, quello iniziale, il periodo di cambiamento e il punto finale. Tutte queste informazioni sono poi l'input per la fase successiva di adattamento al concept drift. Il livello di avviso della deriva, in molti algoritmi, è un livello rilassato della soglia di deriva. I dati raccolti tra questi due livelli servono per consentire l'aggiornamento del modello di apprendimento.

Come avviene il drift e la sua severità

La gravità della deriva si riferisce alla quantificazione della somiglianza tra il nuovo concetto e il precedente. Formalmente la gravità di una deriva del concetto può essere rappresentata con la formula:

$$\Delta = \delta(P_t(X, y), P_{t+1}(X, y))$$

Dove δ è una funzione che misura la differenza tra due distribuzioni di dati e t corrisponde al timestamp in cui avviene il drift. Maggiore è il valore di Δ maggiore sarà l'entità della deriva, di solito è un valore non negativo. In generale, gli algoritmi basati sul tasso di errore non sono in grado di misurare direttamente la gravità del drift in quanto studiano e rilevano il cambiamento del modello di apprendimento non del drift stesso, tuttavia la decrescita dell'accuratezza può essere usata indirettamente per misurarne l'entità. I metodi basati sulla distribuzione dei

dati calcolano direttamente tale metrica in quanto il confronto tra due campioni riflette già la differenza. Maggiore è la distanza maggiore sarà il concept drift, una distanza pari a 0 indica che i due concetti sono identici. La severità del concept drift può essere una linea guida per la scelta della strategia di adattamento al drift. Ad esempio, se la gravità della deriva in un task di classificazione è bassa, il limite decisionale potrebbe spostarsi di poco con il nuovo concetto. Viceversa, se la gravità è elevata lo spostamento del limite decisionale potrebbe essere significativo e portare alla costruzione di un nuovo modello di apprendimento anziché aggiornare il vecchio.

Dove avviene il drift

Il drift avviene nelle regioni in cui il nuovo concetto e il precedente entrano in conflitto. Queste regioni si trovano nello spazio di X in cui $P_t(X, y)$ e $P_{t+1}(X, y)$ sono significativamente diversi. Le tecniche per identificare tali regioni dipendono fortemente dal modello usato per il rilevamento del drift. L'individuazione delle regioni di deriva del concetto agevola la fase di adattamento. Anche se l'intero set di dati va alla deriva, nello spazio esistono delle zone in cui le caratteristiche rimangono stabili più a lungo rispetto ad altre regioni, in queste aree i vecchi modelli di apprendimento possono ancora essere usati per prevedere le istanze lì situate. Inoltre ricerche successive hanno permesso di individuare dati obsoleti e in conflitto con i nuovi in modo da distinguere il rumore.

4.3.3 Adattamento al drift

Sono diverse le strategie che permettono al modello di adattarsi e di reagire al drift. Tre categorie di metodi gestiscono le diverse tipologie di drift.

Simple retraining per il global drift

Il modo più semplice di reagire alla deriva del concetto è riaddestrare un nuovo modello con i dati più recenti per sostituire quello più obsoleto. È necessario avere un rilevatore di deriva del concetto esplicito per decidere quando è il momento di aggiornare il modello. A tal proposito viene adottata una finestra per conservare i dati più recenti e aggiornare il modello oppure i dati più vecchi per effettuare i test sulla modifica della distribuzione. Quando si adottano le strategie delle finestre

è importante la scelta delle dimensioni. Una finestra più piccola può rispecchiare meglio la distribuzione dei dati più recente, una più grande, invece, fornisce più dati per l'addestramento di un nuovo modello.

Ensamble retraining per il recurring drift

Quando la deriva riguarda concetti ricorrenti il riutilizzo di vecchi modelli può evitare sforzi significativi per riqualificare un modello. I metodi ensemble comprendono una serie di classificatori parametrici di base. L'output di ogni classificatore è combinato con gli altri in modo da prevedere i nuovi dati.

Model adjusting per il regional drift

Un'alternativa all'aggiornamento dell'intero modello è lo sviluppo di una tipologia che permetta di apprendere i dati in modo adattivo e di aggiornarsi automaticamente e solo parzialmente. Questo approccio è più efficiente se la deriva si verifica solo in alcune regioni. L'algoritmo più usato in questi modelli è l'albero decisionale perché ha la capacità di esaminare e adattarsi separatamente per ciascuna regione. [37]

4.4 Stato dell'arte: metodologia per il concept drift detection

Come si è visto, l'adozione di sensori in ambito industriale ha creato un approccio nuovo nei processi produttivi e i benefici dei dati raccolti si riscontrano anche nell'ambito delle decisioni aziendali, arrivando a dei risultati più accurati e preziosi. In molti contesti e applicazioni industriali reali, il processo di manutenzione sfrutta strategie complesse, come la manutenzione predittiva, basate su algoritmi supervisionati. Gli algoritmi più usati sono quelli di classificazione, che possono essere difficili da valutare nel tempo. Le metriche e gli indici usati di solito, come precisione, richiamo e accuratezza richiedono la presenza delle etichette di classe reali per poter valutare i modelli. Con l'ingresso di nuovi dati e il rischio di concept drift non sempre ciò è possibile e le prestazioni del modello peggiorano nel tempo.

Per superare questi problemi nascono dei nuovi strumenti di autovalutazione che sono in grado di rilevare automaticamente quando l'adeguatezza di un modello degrada troppo per i dati analizzati. Gli elementi chiave di questi nuovi metodi sono

l'uso di indici che descrivano la coesione all'interno della classe e la separazione tra le classi e che siano in grado di quantificare il degrado del modello di classificazione all'ingresso di nuovi dati nel sistema. [39] In alcune applicazioni può essere molto complesso avere le etichette reali dei dati, il processo può richiedere molto tempo e risorse per isolare i concetti e ottenere un dataset consistente per riaddestrare il modello predittivo. A questo punto è necessaria una metodologia scalabile, innovativa, non supervisionata e in grado di autovalutarsi in modo efficace per indirizzare la stima in tempo reale del degrado del modello. Per una gestione in tempo reale del concept drift l'obiettivo è quello di introdurre uno step nella consueta pipeline analitica e predittiva. In [40] viene presentato un processo non supervisionato in grado di rilevare automaticamente il concept drift basato sulle classi e gestire la ricostruzione di un nuovo modello valutando il degrado della predizione fatta sui nuovi dati. L'obiettivo è quello di identificare quando i nuovi dati possono avere, a causa della presenza di nuove o diverse etichette di classe, una distribuzione diversa da quella disponibile al momento del training. Queste nuove classi possono essere scoperte automaticamente e il nuovo modello predittivo, che può riconoscere le nuove distribuzioni, viene allenato. A questo punto si aprono una serie di interrogativi a cui la ricerca vuole rispondere:

1. Come valutare autonomamente il degrado del modello predittivo in tempo reale?

2. Quali metriche possono guidare il modello di valutazione del degrado?

3. Come scoprire automaticamente le nuove classi dei dati?

4. Come costruire un nuovo modello senza una richiesta specifica dall'utente?

La metodologia proposta in [40] ha alcune caratteristiche peculiari:

1. Rilevamento automatico, ad esempio auto valutazione;

2. Assenza delle etichette reali per i nuovi campioni classificati. La soluzione è basata su una stima non supervisionata che attiva in automatico il retraining del modello predittivo con una descrizione immediata delle variazioni nelle distribuzioni delle etichette di classe motivando l'aggiornamento del modello e dando in questo modo un'interpretazione comprensibile all'uomo;

3. Approccio generale, non fatto su misura per un caso d'uso specifico o per un dominio di applicazione o per una specifica tipologia di dato;

4. Scalabilità, l'algoritmo è stato progettato per essere scalabile e applicabile orizzontalmente nei vari contesti del big data.

4.4.1 Metodologia di rilevamento automatico del concept drift

Aggiornare un modello di predizione estendendo il set di training ai nuovi dati può essere costoso da un punto di vista computazionale e, peggio, può richiedere l'intervento degli esperti di dominio per interpretare i cambiamenti dei fenomeni ed effettuare scelte appropriate per le attività di predizione. Per queste ragioni, è impossibile o sub ottimale riaddestrare frequentemente il modello. Nella soluzione proposta in [41] si è introdotta una fase in cui si identifica e si quantifica il degrado della qualità di una predizione nel tempo. Le metriche utilizzate permettono di attivare automaticamente il riaddestramento del modello predittivo, determinando il momento opportuno in cui è necessario adattare meglio i dati, e di descrivere il cambiamento delle distribuzione dei dati motivandolo con l'aggiornamento delle etichette. I metodi di rilevamento della variazione della distribuzione dei dati sono tre:

- Instance selection: con l'obiettivo di estrarre i campioni più importanti del concetto corrente, queste tecniche si basano su finestre che si spostano sulle ultime instanze;

- Instance weighting: si basa su un algoritmo di apprendimento che considera i pesi delle istanze separatamente;

- Ensemble learning: questa tecnica sfrutta gli insiemi per cogliere meglio le sfumature dei dati.

4.4.2 Self-evaluating della degradazione del modello

Dato un modello predittivo addestrato, poiché la sua conoscenza si basa sulle informazioni contenute nei campioni di train, cioè nei dati storici etichettati raccolti dai sensori, è difficile che le sue performance rimangano immutate nel tempo. Le previsioni sui nuovi dati, se questi hanno una distribuzione diversa da quelli visti in precedenza, possono essere errate o fuorvianti. Le tecniche di valutazione per le analisi predittive in questo contesto non sono applicabili perché necessitano delle etichette reali dei dati, mancanti nei nuovi dati in arrivo. Dunque si sfruttano gli indici non supervisionati che permettono di quantificare la coesione intra classe e la separazione interclasse. Il degrado è definito come una variazione negativa del valore dell'indice dei nuovi dati classificati rispetto a quello calcolato sul set di training.

Il modello proposto effettua prima un calcolo delle metriche non supervisionate sul set di training e, successivamente, nella fase di self-evaluation, ricalcola periodicamente le stesse metriche sui nuovi dati confrontando le une con le altre. Poiché le metriche di qualità dipendono dalla numerosità dei nuovi dati e delle loro etichette, il processo di autovalutazione avviene automaticamente quando un'etichetta di classe subisce un aumento di elementi al suo interno che supera una certa soglia di percentuale rispetto al test precedente. La soglia, definita dagli esperti di dominio e dagli utenti finali, se superata attiva la ricostruzione del modello.

La metrica sfruttata è la *Silhouette*, una misura dell'adattamento del campione alla sua classe predetta. Essa misura quanto un campione è simile alla classe di appartenenza rispetto alle altre classi. La Silhouette può essere usata per ogni tipo di dato usando l'appropriata misura di distanza, ad esempio la distanza Euclidea per i dati strutturati e numerici. [41]

All'interno di una pipeline di machine learning è inserita dunque una fase di self-assessment per identificare il possibile degrado dell'attività di previsione causato da cambiamenti dell'ambiente di produzione per cui è necessario aggiornare e riaddestrare il modello predittivo con i nuovi dati.

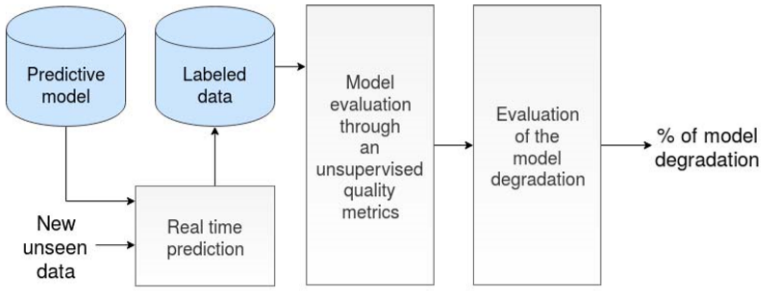


Figura 4.8: Processo di rilevamento automatico del degrado del modello [39]

La Silhoeutte è un indice molto usato per valutare la qualità dei cluster in termini di coesione e separazione. Ha un costo computazionale di $O(N^2)$ dove N è la cardinalità del set di dati, talvolta la dimensione del dataset può essere una limitazione nel contesto dei Big Data in quanto è necessario calcolare l'indice per tutte le distanze a coppie tra i punti del set di dati. [39]

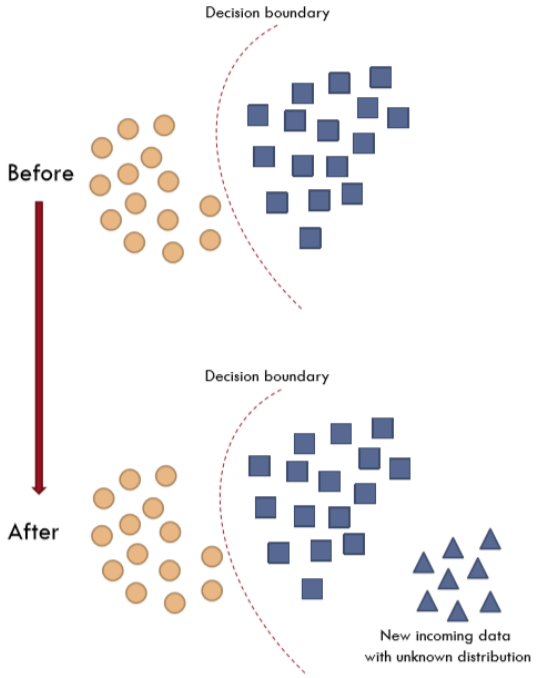


Figura 4.9: Arrivo dei nuovi dati [42]

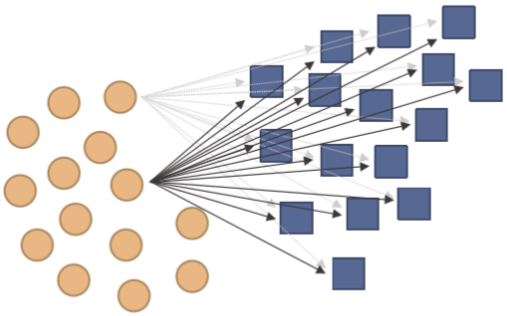


Figura 4.10: Rappresentazione calcolo della Silhouette [42]

Per calcolare la Silhouette per ogni punto si usa il seguente approccio, sia nel calcolo delle distanze che delle somiglianze tra i punti:

1. Dato un oggetto i-esimo, si calcola la sua distanza media rispetto a tutti gli altri elementi appartenenti al suo cluster, questa distanza sarà a_i.

2. Per ogni oggetto i-esimo, si calcola la distanza media rispetto a tutti i punti non appartenenti al suo cluster, il valore minimo è chiamato b_i

3. Per ogni oggetto i-esimo, il coefficiente Silhouette è

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

Il valore della Silhouette può variare tra -1 e 1. Un valore negativo è da evitare perché corrisponde al caso in cui a_i, cioè la media delle distanze tra i punti nello stesso cluster è maggiore di b_i cioè della distanza media minima con i punti di un altro cluster. È desiderabile che il coefficiente di Silhouette sia positivo $a_i < b_i$ con a_i il più vicino possibile a 0 in modo che il coefficiente possa essere uguale a 1. La Silhouette media di un gruppo o cluster può essere calcolata facendo la media delle silhouette di tutti i punti, avendo in questo modo una valutazione generale. [27]

Per superare il costo computazionale della Silhouette [39] ha studiato un nuovo indice, il *"Descriptor silhouette"*. Il *DS* si basa sull'idea che la forma geometrica di un gruppo di punti può essere descritta con un basso numero di descrittori ben

distribuiti nello spazio. Gli algoritmi non supervisionati di clustering si adattano bene nel descrivere gli spazi geometrici. I descrittori sono estratti sfruttando i centroidi, calcolati con il k-means, e usati per descrivere la classe a cui appartengono. In questo modo viene ridotto drasticamente il numero di distanze da calcolare in quanto saranno calcolate solamente quelle tra ciascun punto del set di dati e i descrittori. La complessità computazionale passa da $O(N^2)$ a $O(N * C * D)$, dove N è la cardinalità del dataset, C è il numero di classi conosciute dal classificatore e D è il numero di descrittori per ogni classe. Nel caso di studio discusso nel capitolo successivo non sarà necessario il calcolo del *Descriptor Silhouette* in quanto la dimensione dei dataset usati consente di calcolare in tempi ragionevoli i valori delle distanze tra i punti.

La valutazione si effettua con il confronto di due indici di qualità: quello base, calcolato sul set di training e quello corrente, calcolato sull'unione tra i dati di training e quelli nuovi. La variazione della qualità può essere quantificata separatamente per ogni classe. L'indice viene così calcolato di volta in volta per ogni classe e ne viene plottata la curva, ottenuta ordinando i valori in modo crescente. Per rendere possibile il confronto tra le curve, esse devono avere la stessa cardinalità di punti, se la numerosità è maggiore nei vari passi deve essere effettuato un down-sampling per adattare il numero di valori.

Nel caso della valutazione del modello, le classi del classificatore possono essere considerate come cluster a cui sono assegnati nuovi dati in arrivo senza etichetta. Ogni classe di classificazione è descritta da una curva di Silhouette, ottenuta ordinando i valori calcolati per ogni punto di ogni classe.

La curva ottenuta descrive la coesione intra-classe e la separazione inter-classe per il dataset di training. Con l'arrivo di nuovi dati il classificatore li etichetta assegnando le classi già conosciute. A questo punto inizia la fase di Self-assessment in cui viene ricalcolata la Silhouette includendo i nuovi dati etichettati: uno spostamento della curva verso l'alto rappresenta un miglioramento in termini di coesione intra-classe e separazione inter-classe mentre uno shift verso il basso ne indica un peggioramento. Il degrado della Silhouette indica la presenza di nuovi punti, non presenti al momento di addestramento del modello. Questo scenario può essere tradotto come un degrado del modello stesso di classificazione: il modello non è in grado di riconoscere la nuova distribuzione dei dati in quanto essi non erano ancora disponibili al momento della sua creazione.

Dato un modello di predizione trainato su una o più classi c, al tempo t il degrado della classe c è descritto con la seguente relazione:

$$DEG(c,t) = \alpha * MAAPE(Sil_{t0}, Sil_t) * \frac{N_c}{N}$$

$$\alpha = \begin{cases} 1, & \text{if } Sil_{t0} \geq Sil_t \\ -1, & \text{if } Sil_{t0} < Sil_t \end{cases}$$

Il coefficiente α definisce se il degrado è positivo o negativo. Nel primo caso è possibile che ci sia una riduzione delle prestazioni del classificatore, nel secondo caso i nuovi dati arrivati hanno una distribuzione simile a quella dei dati su cui il modello è stato addestrato facendo aumentare la coesione della classe analizzata. Il *MAAPE* quantifica lo spostamento della curva della Silhouette ed è la media dell'*Arctangent Absolute Percentage Error (AAPE)* che per un dato punto è definito come:

$$AAPE_t = arctan\left(\left|\frac{Sil_t - Sil_{t0}}{Sil_t}\right|\right)$$

$$MAAPE = \frac{1}{N}\sum_{t=1}^{N} AAPE_t$$

Il degrado è calcolato come l'errore tra la curva della Silhouette iniziale, al momento della creazione del modello e quella ottenuta ad un certo intervallo t, dopo che il modello ha ricevuto i nuovi dati. $\frac{N_c}{N}$ dà un peso alla formula del degrado, N_c è il numero di nuovi record assegnati alla classe c e N è il numero totale di nuovi dati. Il degrado dell'intero modello può essere calcolato come la somma del degrado di tutte le classi.

N assume un valore che può essere limitato a quello del set di training se $N > N_{train}$ in modo da effettuare un confronto equo. Il self-evaluation si attiva quando almeno una classe ha visto un aumento di percentuale nel pacchetto di test N_c rispetto al calcolo precedente. Questa euristica ci permette di evitare ritardi nel rilevamento del concept drift. Infine, si può considerare di attivare una totale ricostruzione del modello quando il degrado generale o di almeno una classe supera una determinata soglia. La valutazione di tali soglie dipende dalle euristiche applicate e può essere oggetto di studi futuri. [39]

4.4.3 Concept drift in un contesto di outlier detection

Nello stato dell'arte è stata riportata una metodologia non supervisionata per il rilevamento del concept drift attraverso il calcolo dell'indicatore della Silhouette. Tuttavia l'approccio proposto nell'articolo [39] è studiato e pensato per un contesto multiclasse in cui più classi sono inserite nel dataset di training ed è su di esse che si studia il degrado con il confronto tra la coesione intra cluster e la distanza inter cluster. In un contesto di outlier detection, in cui l'Isolation forest lavora come classificatore binario, il calcolo della Silhouette incontra delle limitazioni. La presenza di due sole classi, di cui una per i valori normali e l'altra per i valori anomali rende poco gestibile il concetto di distanza inter cluster a causa della presenza di outlier. La classe dei valori normali sarà sempre molto coesa e manterrà la sua coesione anche con l'ingresso di nuovi dati nel test, la classe degli outlier invece tenderà ad essere sempre meno coesa. Dunque per poter capire come varia, in questo caso, la distribuzione dei dati all'interno della classe degli outlier è stato necessario modificare la formula della Silhouette sfruttando solo il termine relativo alla distanza intra cluster. Per ogni punto all'interno delle due classi si calcola la media delle distanze rispetto ad ogni altro punto all'interno della stessa classe.

La metodologia qui proposta per il contesto di outlier detection sfrutta la misura della distanza per capire quanto due elementi i e j appartenenti allo stesso gruppo sono coesi o meno tra loro. Più gli oggetti sono lontani maggiore è la loro diversità, minore è la coesione. Per fare ciò si sfrutta la *Matrice di dissimilarità* e il concetto di *Distanza Euclidea*. Prima di effettuare il calcolo delle distanze è importante accertarsi che gli attributi numerici considerati siano tutti riportati con la stessa unità di misura, in caso contrario è opportuno effettuare una normalizzazione.

La *Distanza Euclidea* è la misura più usata per il calcolo delle distanze. Dati due oggetti $i = (x_{i1}, x_{i2}, ..., x_{ip})$ e $j = (x_{j1}, x_{j2}, ..., x_{jp})$ descritti da p attributi, la *Distanza Euclidea* tra loro è definita come:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$$

Per ogni punto si calcolano le distanze a coppie con tutti gli altri punti, di queste distanze viene fatta una media che indica il valore di distanza intra cluster per quel determinato punto.

Nella metodologia proposta, al tempo di training t_0 tali distanze vengono calcolate per ogni punto nel set di training, ordinate e infine plottate in una

curva. Successivamente per ogni tempo di test t_i si effettuano le nuove predizioni delle etichette, si uniscono i dati di test con i dati di training e su questi viene ricalcolata la curva delle distanze medie intra cluster per tutti i punti. Per rendere confrontabili le curve ottenute al tempo t_0 e nei tempi successivi di test t viene fatto un downsample sull'unione di train e test per ottenere una cardinalità pari a quella di training. A questo punto si plotta la curva ordinata delle distanze intra-cluster e si osserva il suo spostamento, più la curva si alza, maggiore sarà la distanza tra i punti e di conseguenza minore sarà la coesione all'interno della classe, implicando così la presenza di una deriva dei dati. A questo punto per poter quantificare la variazione della distribuzione dei dati e quindi il drift si calcola, come visto già in [39], il *Mean Arctangent Absolute Percentage Error (MAAPE)* tra i valori delle distanze nel seguente modo:

$$AAPE_t = arctan\left(\left|\frac{DistanzaIntraCluster_t - DistanzaIntraCluster_{t0}}{DistanzaIntraCluster_t}\right|\right)$$

$$MAAPE = \frac{1}{N}\sum_{t=1}^{N} AAPE_t$$

Il $MAAPE$ indica il valore del degrado totale, se moltiplicato per la percentuale di punti $\frac{N_c}{N}$ si ottiene il valore approssimato del degrado per la singola classe.

Le due metodologie proposte sono delle alternative per il rilevamento del concept drift in modo non supervisionato. Nel capitolo successivo saranno riportati i risultati delle due alternative, sarà fatto un confronto e si determinerà la soluzione migliore per il caso di studio.

4.5 Strumenti di implementazione usati

4.5.1 Python

Per le analisi oggetto di questa tesi il linguaggio di programmazione usato è Python. Si tratta di un linguaggio di programmazione ad alto livello, rilasciato per la prima volta nel 1991 dal suo creatore Guido Van Rossum. Python supporta diversi paradigmi di programmazione, da quella procedurale con l'uso di funzioni a quella ad oggetti. [43] È fornito di una libreria built-in e di una gestione automatica della

memoria e delle eccezioni. Queste peculiarità rendono Python uno dei linguaggi più ricchi e comodi da usare. Adotta un meccanismo garbage collection che si occupa automaticamente dell'allocazione e del rilascio della memoria. Questo consente al programmatore di usare variabili liberamente senza dichiararle e allocarle manualmente. Python è nato per essere un linguaggio facilmente interpretabile, con una sintassi pulita e snella che permette una programmazione molto chiara e senza ambiguità in cui i blocchi logici vengono costruiti allineando le righe. Si tratta di un linguaggio che si definisce pseudocompilato, in cui un interprete si occupa di analizzare il codice sorgente, ovvero file testuali con estensione .py, e se sintatticamente corretto di eseguirlo. Il codice sorgente non viene convertito direttamente in linguaggio macchina, ma passa prima da una fase di pre compilazione in bytecode per essere poi riutilizzato dopo la prima esecuzione del programma ed evitare così di reinterpretare ogni volta il file sorgente migliorando le prestazioni. A differenza del C non esiste una fase di compilazione separata che generi un file eseguibile a partire da quello sorgente. Il file sorgente Python può essere interpretato e eseguito dalla maggior parte delle piattaforme esistenti. È un linguaggio portabile sviluppato in ANSI C ed è possibile usarlo su qualsiasi piattaforma purché abbia installato l'interprete Python. Python è gratuito e può essere liberamente modificato e ridistribuito con licenza open-source. Tutte queste caratteristiche lo rendono uno dei linguaggi più diffusi al mondo in quanto può essere usato in moltissimi campi applicativi.

Libreria scikit-learn

È una libreria open source di apprendimento automatico per il linguaggio di programmazione Python. Si tratta di un modulo integrato ad altri pacchetti usati solitamente per l'apprendimento automatico classico ad esempio numpy, matplotlib e scipy. È uno strumento molto potente e in grado di risolvere la maggior parte dei problemi di machine learning. Fornisce una vasta gamma di algoritmi di apprendimento supervisionato e non supervisionato, oltre alle varie metriche di qualità dei modelli. Le analisi che sono state effettuate hanno sfruttato gli algoritmi di machine learning presenti in questa libreria. [44]

Libreria pandas

Questa libreria ha l'obiettivo di eseguire analisi di dati reali su Python, essa è uno strumento utile per manipolare i dati in modo flessibile e potente. Pandas fornisce delle strutture dati e delle funzioni progettate per far lavorare in modo rapido, facile ed espressivo i dati strutturati. È uno degli elementi chiave che rendono Pyhton potente e produttivo nel contesto del data analysis. Nell'ambito del progetto di tesi è stata usata per gestire i dati in formato Json. Essi sono stati strutturati in un oggetto DataFrame, che è alla base di questa libreria. Il DataFrame è una struttura tabellare bidimensionale e orientata per colonne con etichette di righe e di colonna. Pandas inoltre combina le caratteristiche molto performanti di NumPy con una manipolazione flessibile di fogli di calcolo o database relazionali. Ha una funzionalità molto sofisticata di indicizzazione per semplificare la rimodulazione, la suddivisione e le aggregazioni di sottoinsiemi di dati. [45]

Libreria Matplotlib

Matplotlib è la libreria Python più usata per la produzione di grafici e altre visualizzazioni di dati 2D. Si integra bene con IPython fornendo un ambiente interattivo per la stampa e l'esplorazione dei dati. Matplotlib fornisce una grande selezione di grafici che aiutano a capire i modelli, le tendenze e le correlazioni all'interno dei dati. È importante per fornire una maggiore comprensione delle informazioni quantitative. [45]

4.5.2 Json

JavaScript Object Notation o in breve JSON è un formato di scambio di dati introdotto nel 1999 ed è stato ampiamente adottato a metà degli anni 2000. È il formato standard più usato nell'ambito della comunicazione tra i servizi web e i loro clienti. JSON deriva da JavaScript ma è un formato indipendente dalla piattaforma. Sono tanti i linguaggi di programmazione in grado di leggere e gestire i file di questa estensione. I file JSON sono formati da stringhe, ciascuna delle quali rappresenta un oggetto. In Python i file JSON possono essere letti tramite Pandas con il metodo $read\_json()$ che restituisce un DataFrame che memorizza i dati sotto forma di colonne e righe in modo da renderli pronti per gestirli e usarli per le analisi. [46]

Capitolo 5

Caso di studio e risultati sperimentali

Dopo aver discusso del contesto dell'Industria 4.0 in cui il lavoro di tesi si colloca e aver spiegato la metodologia di lavoro usata, in questo capitolo saranno esposti il caso di studio oggetto di analisi e gli algoritmi testati e implementati per il rilevamento del concept drift.

5.1 Caso di studio

I dataset analizzati contengono i dati raccolti attraverso dei sensori installati su alcuni bracci robotici che operano in un modello di fabbrica intelligente nel contesto dell'industria 4.0. L'azienda a cui appartengono i bracci robotici sviluppa e realizza processi di automazione, soluzioni e servizi di produzione ed è specializzata in robot di saldatura e in macchine per magazzini automatizzati. In particolare, lo studio di questo progetto di tesi è volto ad analizzare il tensionamento delle cinghie di trasmissione dei bracci robotici. Questi sistemi variano il comportamento sulla base della tensione applicata alla cinghia, che dipende dal numero di rondelle applicate manualmente ai robot. Il tensionamento della cinghia può avere dei comportamenti anomali che possono impedire il funzionamento del robot. Ad esempio, una tensione troppo bassa può causare slittamenti, surriscaldamento e usura precoce. Viceversa, il tensionamento elevato può danneggiare cinghie e cuscinetti. Il ciclo di produzione del braccio robotico ha una durata di circa 24 secondi con una rilevazione ogni 2

millisecondi per un totale di 11967 rilevazioni. Le misurazioni sono in Ampere e rappresentano la corrente consumata nel corso delle fasi del ciclo di lavorazione. L'osservazione della corrente consumata permette di rilevare l'effetto della tensione della cinghia. [39]

E' stato osservato il comportamento di due bracci robotici praticamente identici, i cui dataset contenenti le informazioni sui cicli si distinguono in *Gray* e *White*. I dati, in entrambi i casi, sono stati salvati in formato JSON MIMOSA. Ogni rilevazione proveniente dallo stream di dati è stata registrata in un file che contiene le informazioni relative al ciclo corrispondente, ai fini dell'analisi sono stati considerati: il timestamp, che si riferisce all'istante di tempo in cui è stata effettuata la misurazione, gli 11967 valori di corrente raccolti durante il ciclo con relative misure statistiche e l'etichetta assegnata dagli esperti di dominio. Una prima parte dell'analisi si è concentrata sullo studio dei dati a disposizione per conoscerne le caratteristiche statistiche, la presenza di valori anomali e le distribuzioni. Nella seconda parte, per ogni ciclo di produzione, è stata fatta una *data transformation* durante la quale sono stati calcolati gli "smart data" su cui poi sono stati applicati gli algoritmi di outlier detection e di rilevamento del concept drift.

5.1.1 Analisi del segnale

I due dataset corrispondenti ai due bracci robotici hanno una dimensione leggermente diversa. Nel caso di *Gray* si ha un totale di 6019 record, corrispondenti ai relativi timestamp, *White* invece ha un totale di 5729 elementi. In entrambi i casi ogni record rappresenta un ciclo di lavorazione e lo stream di dati appartiene ad un intervallo temporale compreso tra il **24/02/2020** e il **04/03/2020**. A ciascun record è stata assegnata un'etichetta che lo identifica alla classe di appartenenza. Il significato delle tre etichette (0, 10, 15) presenti, assegnate dagli esperti di dominio, è stato generalizzato ai fini delle analisi. La distribuzione dei dati nelle etichette è la seguente:

Gray		
Etichetta	**Numero Cicli**	**% Dataset**
0	1287	21,4 %
10	3419	56,8 %
15	1313	21,8 %

White		
Etichetta	**Numero Cicli**	**% Dataset**
0	1216	21,2 %
10	3275	57,2 %
15	1238	21,6 %

Ogni ciclo di lavorazione, costituito da 11967 rilevazioni, è rappresentato dalla curva del segnale riportata in figura 5.1. Nella curva è possibile distinguere 4 fasi (figura 5.2): la prima è caratterizzata da oscillazioni, che nella seconda si stabilizzano seguendo un rettilineo, nella terza fase si ha un crollo del segnale, con valori negativi e infine una fase con valori quasi costanti. Ad ogni fase corrisponde una diversa posizione del motore: da una posizione iniziale a -500 gradi raggiunge lentamente +90 gradi ad una velocità pari al 20% della sua velocità massima, mantiene la posizione per 5 secondi e poi ritorna a -500 gradi con la velocità massima mantenendo infine la nuova posizione per 5 secondi.

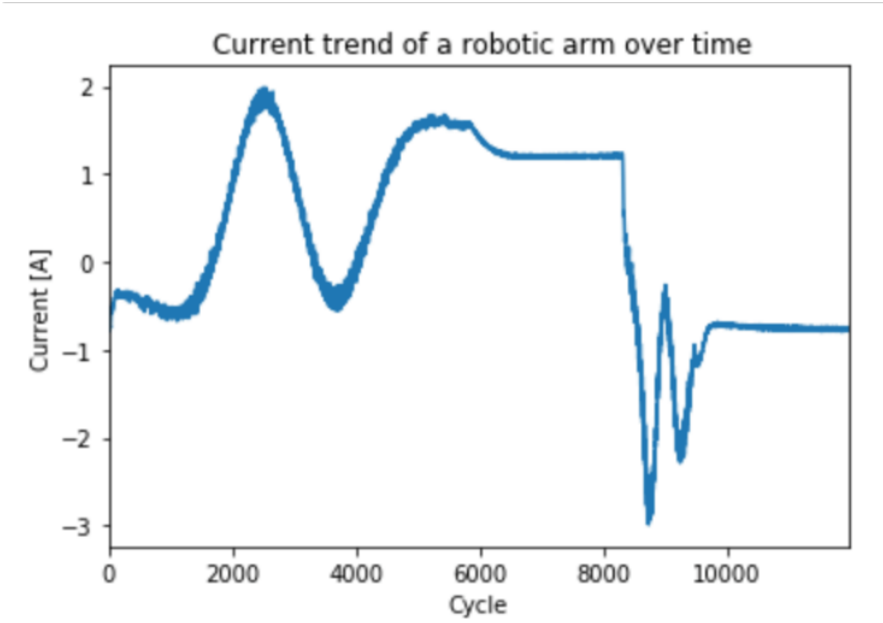


Figura 5.1: Andamento segnale ciclo di produzione

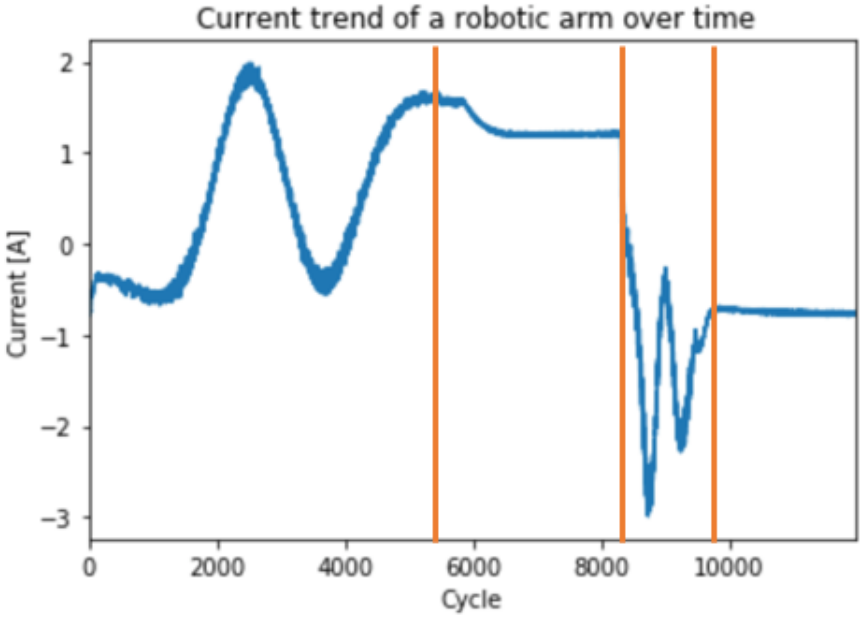


Figura 5.2: Andamento segnale ciclo di produzione con fasi

Per ogni timestamp oltre ai valori della corrente sono riportate alcune feautures che caratterizzano il segnale. Una delle misure più significative è la media, il cui andamento è stato sfruttato per l'individuazione di alcune anomalie.

Gray

E' stata fatta una verifica sui valori della media forniti, dall'analisi sono emerse alcune incongruenze rispetto ai valori calcolati a partire dai valori di corrente:

- Cambia il range dei valori assunti, la media calcolata varia da 0.18 A a 0.31 A (Figura 5.3), la media contenuta già nel dataset invece varia da 0.21 A a 0.27 A (Figura 5.4).

- Dal plot sovrapposto di entrambe le medie si nota la presenza di due valori anomali in corrispondenza del campione 43 (timestamp "24-02-2020 14.28.07") e 84 (timestamp "24-02-2020 15.00.49")

È stato ritenuto opportuno eliminare i due valori anomali per le analisi successive in modo da rendere più coerente lo studio.

Un'informazione che si può cogliere dall'andamento della media è il chiaro passaggio da un'etichetta all'altra, ad esempio si nota perfettamente il sovratensionamento dell'etichetta 10 rispetto alla 0 e alla 15, nella parte centrale della curva tra i campioni 1287 e 4705.

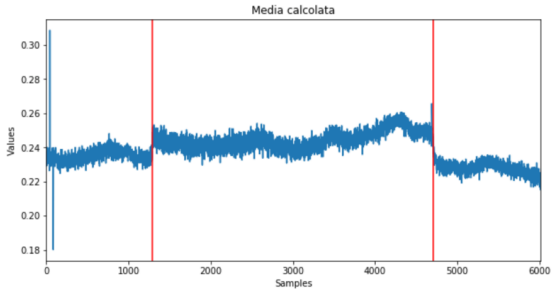


Figura 5.3: Andamento media calcolata

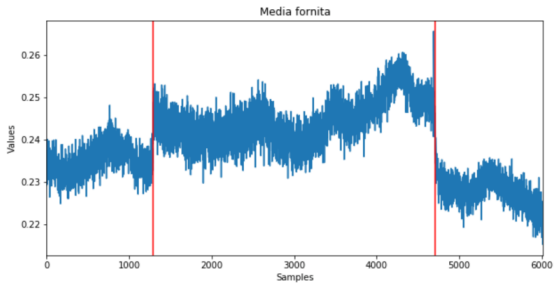


Figura 5.4: Andamento media fornita

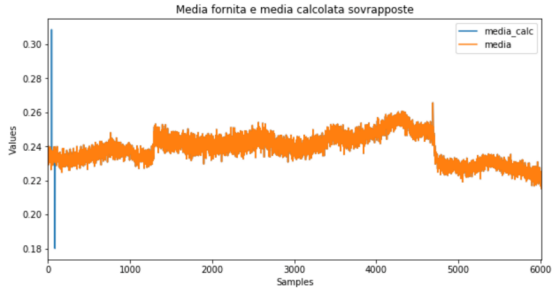


Figura 5.5: Andamento medie sovrapposte

All'interno della classe 10 si può notare una variabilità maggiore nei valori assunti dalla media, in particolare nel tratto precedente al passaggio alla classe 15

è possibile notarne un picco. Considerando nel dettaglio i valori della media in corrispondenza dell'intervallo anomalo si è riusciti a individuare che il ciclo numero 4690, appartenente alla classe 10, assume il valore anomalo (Figura 5.6).

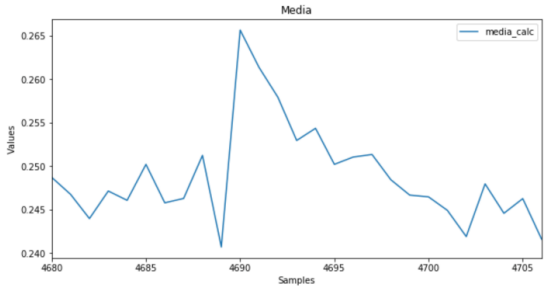


Figura 5.6: Valore anomalo nell'andamento della media

Il picco presente potrebbe essere giustificato dal salto temporale tra i rilevamenti 4689 e 4690. Dalla figura 5.7 si nota un passaggio, in corrispondenza di questi due campioni, dal 27/02/2020 al 03/03/2020.

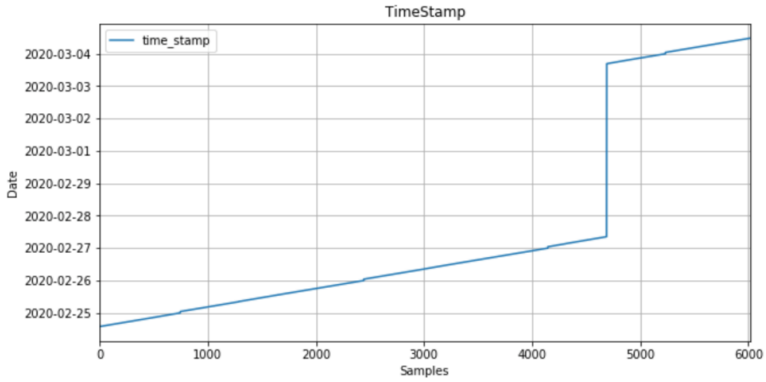


Figura 5.7: Andamento Timestamp dei cicli di lavorazione in Gray

In figura 5.8 sono rappresentati con uno scatter plot i punti della media interpolati da una retta che ne traccia un andamento leggermente negativo. I due punti individuati nella parte sinistra del grafico sono gli outlier già visti al passo precedente e poi rimossi per le analisi successive.

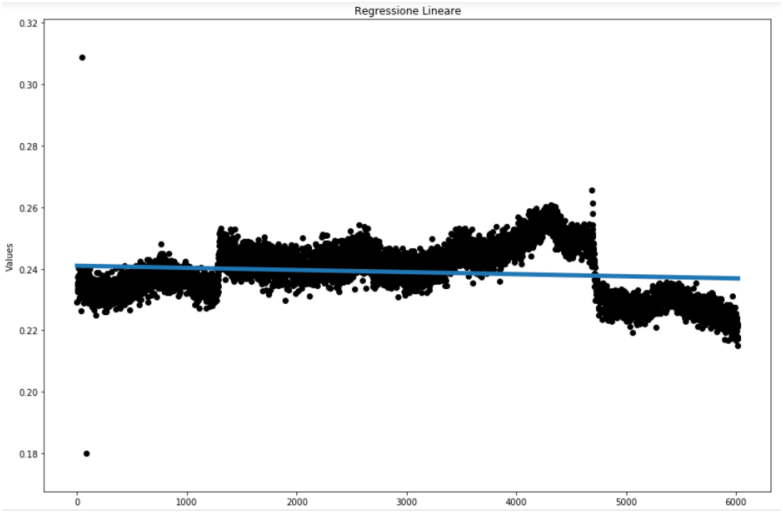


Figura 5.8: Regressione lineare dataset Gray

White

A differenza del caso di *Gray*, in *White* la media calcolata coincide per tutto l'andamento dei valori del dataset a quella già fornita. Il grafico in figura 5.9 mostra la perfetta sovrapposizione delle medie. Le linee verticali indicano il passaggio da una classe all'altra, i primi campioni appartengono all'etichetta 0, nella parte centrale alla 10 e in quella finale alla 15.

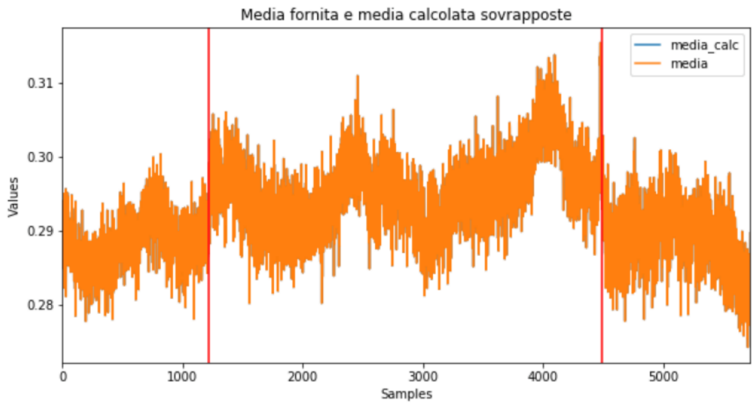


Figura 5.9: Medie sovrapposte white

Lo stacco tra le classi è meno marcato rispetto al dataset *Gray*, si nota un andamento oscillatorio e un picco intorno ai campioni finali dell'etichetta 10 (Figura

5.10). Anche in questo caso i valori anomali sono da imputare al salto temporale tra i cicli di produzione, come si vede in figura 5.11. Si ha il passaggio dal 27/02/2020 al 4/03/2020, il record corrispondente è il 4476.

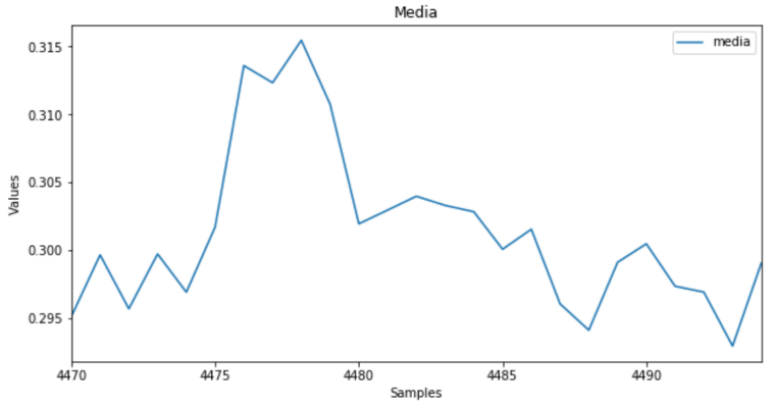


Figura 5.10: Valori anomali nei cicli di produzione di White

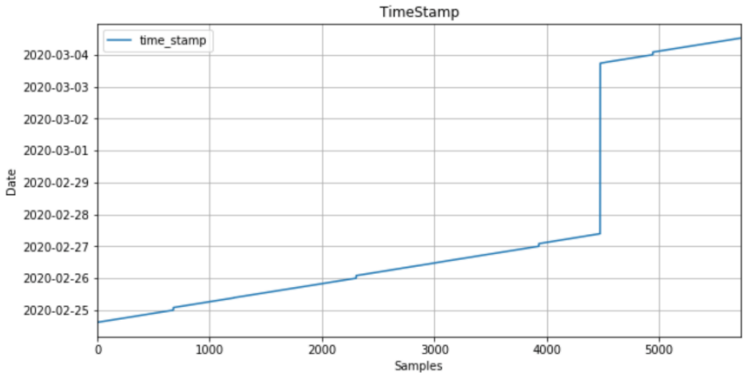


Figura 5.11: Andamento Timestamp cicli di produzione in White

L'andamento della media è stato interpolato con un retta di regressione lineare e si nota che l'inclinazione è leggermente crescente.

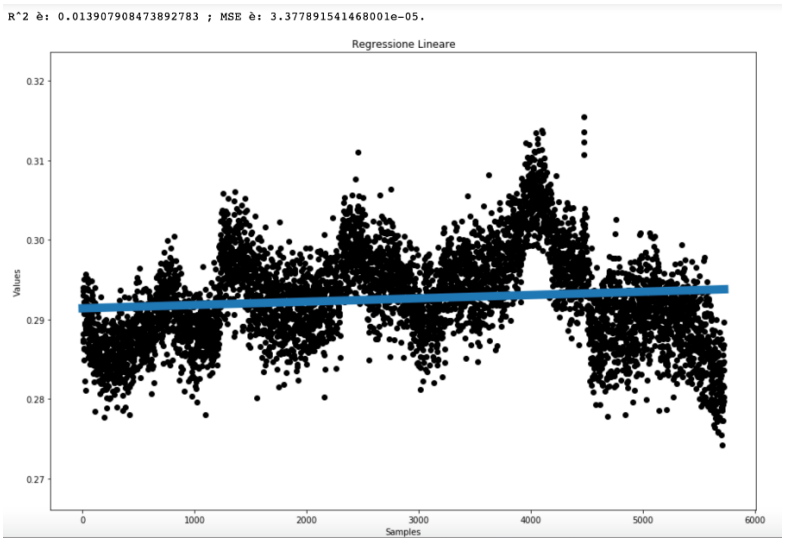


Figura 5.12: Regressione lineare dataset White

5.1.2 Data Transformation: Smart data

I sensori industriali monitorano molti processi produttivi caratterizzati da una ripetizione periodica e da una durata specifica. La fase del processo di *Data Transformation* ha il compito di trasformare ed elaborare i dati grezzi provenienti dai sensori per estrarre le caratteristiche principali che descrivono i segnali. Ogni ciclo monitorato è stato suddiviso in più split temporali con l'obiettivo di estrarre meglio la variabilità locale dei sotto-cicli. Per ogni divisione sono state calcolate alcune caratteristiche statistiche: media, deviazione standard, minimo, massimo, mediana, quartili, curtosi, asimmetria, errore quadrato della media, somma dei valori assoluti, numero di elementi sulla media, energia assoluta e variazione assoluta della media. Queste caratteristiche calcolate prendono il nome di *Smart Data* e sono usate per le analisi successive sui cicli. A causa dell'elevato numero di statistiche calcolate è possibile che in questa fase sia creato un numero elevato di attributi che può influenzare le prestazioni delle analisi. Alcune features possono essere fortemente correlate tra loro portando quindi delle informazioni ridondanti con il rischio di produrre del rumore nella fase di creazione del modello. A questo punto, è utile selezionare solo gli attributi che contengono le informazioni rilevanti. La metodologia che è stata applicata in questo lavoro di tesi è quella proposta nell'articolo [39]. La tecnica considerata include il test di correlazione

di Pearson. Si calcola la correlazione di ciascuna coppia di attributi rimuovendo quelli maggiormente correlati, relativamente a tutti gli altri attributi si possono identificare quelli che possono essere scartati senza perdere la precisione nella costruzione del modello.

Nel caso in esame inizialmente il segnale di corrente è stato suddiviso in 16, 24 e 32 splits e in seguito ad alcune analisi preliminari si è scelto di proseguire lo studio con il modello da 24 splits.

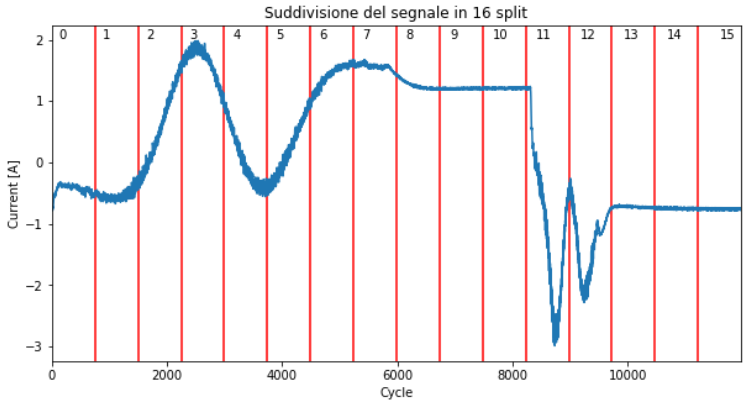


Figura 5.13: Divisione del segnale in 16 split

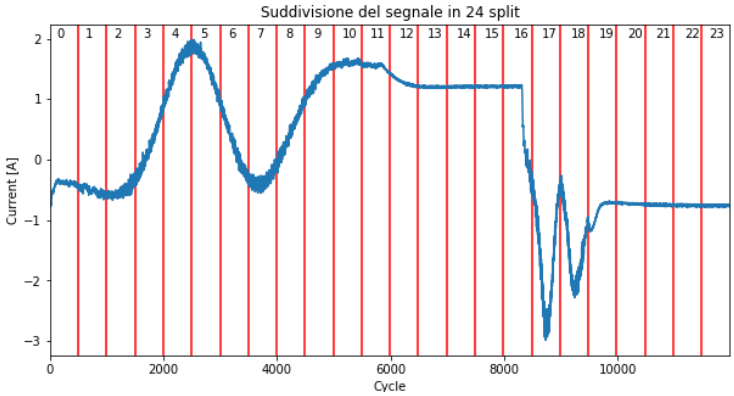


Figura 5.14: Divisione del segnale in 24 split

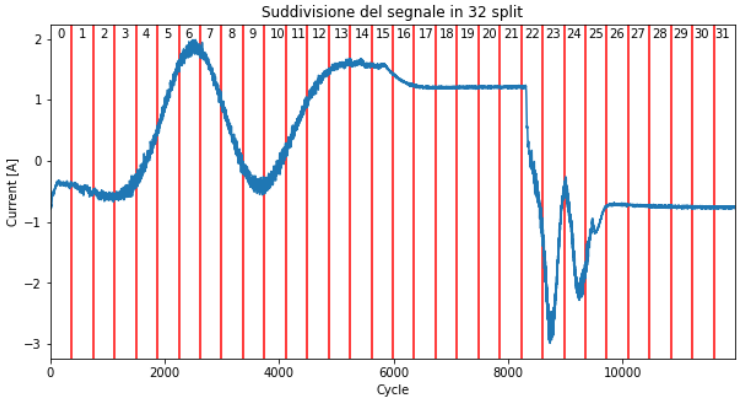


Figura 5.15: Divisione del segnale in 32 split

5.1.3 Features selection

La fase di *Features selection* ha selezionato gli attributi più importanti eliminando quelli con un valore del coefficiente *Mean Absolute Correlation (MAC)* superiore a 0.5. Nelle tabelle 5.1 e 5.2 sono mostrati i numeri degli attributi rimasti al termine della features selection. In generale si nota una maggiore riduzione di attributi nel dataset *White*. Questo può essere dovuto alla presenza in *Gray* di caratteristiche più diversificate e con una bassa correlazione in quanto gli split tra loro presentano una maggiore variabilità.

Gray		
Numero splits	**#Attributi iniziali**	**#Attributi rimasti**
16	224	187
24	336	299
32	448	424

Tabella 5.1: Numero attributi dopo la features selection per dataset Gray

White		
Numero splits	**#Attributi iniziali**	**#Attributi rimasti**
16	224	164
24	336	249
32	448	352

Tabella 5.2: Numero attributi dopo la features selection per dataset White

Considerando i dataset di 24 split dopo la features selection, i valori degli attributi sono stati normalizzati e ne sono state calcolate le componenti tramite la *Principal Component Analysis (PCA)*. La *PCA* è un metodo statistico che crea nuove funzionalità o caratteristiche dei dati utilizzando quelle del set di dati iniziali e combinandole insieme. Le 3 componenti utili per creare lo scatter plot in 3D dei dati sono state calcolate tramite il metodo *PCA* di *Scikit-learn* [47]. Una visione tridimensionale dei dataset è importante per capire come si distribuiscono le etichette nello spazio e comprendere le loro posizioni reciproche. Ad ogni colore corrisponde un'etichetta di classe, nel caso di *Gray* si nota come la classe 15 sia ben separata dalle altre due, nel caso di *White* tutte e tre le etichette hanno forma simile e risultano ben separate e distinguibili. Queste caratteristiche hanno dei risvolti nell'ambito dei risultati sperimentali, come sarà possibile osservare nella successiva sezione.

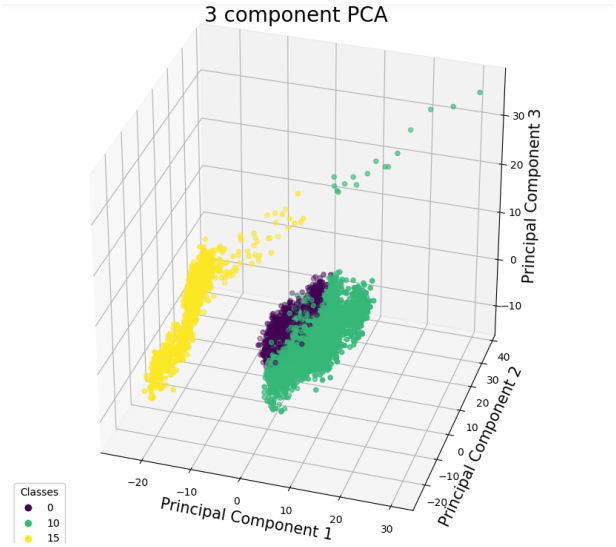


Figura 5.16: Rappresentazione con PCA dataset Gray

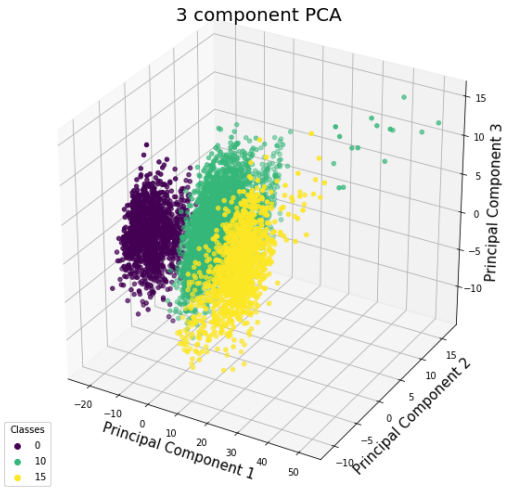


Figura 5.17: Rappresentazione con PCA dataset White

5.2 Risultati sperimentali: Anomaly detection e Concept Drift Detection

Gli esperimenti sono stati eseguiti su un PC con Intel® Core™ i7-8565U CPU @ 1.80 GHz 1.99 GHz e 8 GB di RAM. Il sistema operativo è Windows 10. Gli algoritmi sono stati programmati in Python sulla piattaforma Jupyter. Dalle informazioni sul contesto sperimentale e sul caso di studio, si è dedotto che il classificatore più adatto è l'*Isolation Forest*. La cinghia parte da una condizione di funzionamento normale, a cui si associa la classe 0, e si vuole verificare cosa avviene quando inizia un degrado e un cambio delle prestazioni. Con questo obiettivo si immagina di iniettare nel classificatore un flusso di nuovi dati appartenenti ad una classe diversa da quella di addestramento e si cerca di individuare il momento in cui si verifica la deriva (o drift) dei dati. Per verificare la robustezza e la validità del metodo dell'Isolation Forest sono state fatte delle prove di *Outlier Detection* su entrambi i dataset. Dopo aver provato la robustezza del modello sono state applicate le tecniche di rilevamento del *Concept Drift* sfruttando la metodologia non supervisionata esposta nel capitolo precedente. L'uso di un classificatore binario e non multiclasse ha reso poco fruibile la metrica della *Silhouette* che è stata sostituita da quella della sola *Distanza intra cluster*, indicatore della Coesione all'interno delle classi. Sarà presentato di seguito un confronto tra le due metriche e sarà esposta nel dettaglio la soluzione proposta.

5.2.1 Isolation Forest in Scikit-learn

Il metodo utilizzato è l'*sklearn.ensemble.IsolationForest* della libreria Scikit-learn [47] il quale ritorna il punteggio di anomalia per ogni campione usando l'algoritmo dell'Isolation Forest. Si tratta di un classificatore binario che assegna due possibili etichette. Il metodo *predict*, applicato sul dataset di test, restituisce per ogni elemento l'etichetta predetta, assegna classe -1 per indicare gli outliers, 1 per i valori normali. I parametri settati nel corso delle analisi sono:

- **n_estimators**: è il numero di stimatori "Isolation tree" raggruppati nell'Isolation Forest.
- **contamination**: è la % di outliers nel dataset. E' un parametro che dipende dalla composizione dei dati.

5.2.2 Valutazione del modello con Outlier Detection

In generale negli esperimenti effettuati si è assunto che il comportamento considerato normale è quello della classe 0, mentre la classe 10 rappresenta dei dati sovratensionati che potrebbero implicare un comportamento anomalo del robot o l'arrivo di una nuova classe. A tal proposito si è deciso di addestrare l'Isolation Forest sul 68.8% della classe 0 e di fare il test inserendo nel classificatore il restante 31.2% della classe 0 e il 100% della classe 10. I dati della classe 0 e della classe 10 sono stati ordinati per timestamp in modo da simulare, da un punto di vista temporale, l'ingresso dei dati nel classificatore. Una volta fatto il predict sul dataset di test creato dall'unione della rimanenza della classe 0 e della classe 10 si sono valutate le etichette predette. Per lo svolgimento degli esperimenti si è testata la sensibilità del modello sulla contamination con i seguenti passaggi:

1. L'Isolation Forest è stata addestrata facendo variare i valori della contamination tra 0, 0.01, 0.05, 0.1 e 0.5.

2. Per ogni valore di contamination è stato fatto il predict sul set di test e infine un confronto tra le etichette reali e le etichette predette per verificare la robustezza e la validità del classificatore. In generale per la verifica, i punti che il classificatore ha assegnato alla classe 1 si è assunto siano stati assegnati all'etichetta 0, i punti della classe -1 alla 10. L'accuratezza valuta in generale la bontà dell'intero modello ma poiché le due classi del dataset di test sono sbilanciate numericamente l'accuratezza è affiancata dal richiamo e dalla precisione che valutano le predizioni per le singole classi. Per il calcolo delle metriche è stato usato il modulo *sklearn.metrics* della libreria Scikit-learn [47].

3. Infine, sulle etichette predette è stata creata una *Sliding window* o *Finestra scorrevole* di 50 elementi per verificare la % di elementi appartenenti a ciascuna delle 2 classi e il suo andamento.

Dataset Gray

Il valore migliore di contamination tra quelli testati per il dataset *Gray* è 0.1, come è possibile osservare in tabella 5.3, raggiungendo il giusto trade-off tra i valori di accuratezza, precisione e richiamo. L'accuratezza del modello aumenta all'aumentare del valore di contamination per poi diminuire quando diventa troppo

alto. Questo è giustificabile dalla distribuzione dei dati. L'etichetta 0 è ben separata dalla 10, l'isolation forest trainata sulla classe 0 riesce a riconoscere come outliers gran parte degli elementi della classe 10 assegnandoli alla classe -1. Ma, se il valore di contamination è troppo alto come nel caso di 0.5, il classificatore assegnerà erroneamente alla classe -1 gran parte degli elementi della classe 0, come si nota dai valori del richiamo. Viceversa un valore di contamination troppo basso considererebbe molti elementi dell'etichetta 10 come normali assegnandoli erroneamente alla classe 1 anziché alla -1, come si nota dal valore basso di precisione per la classe 0.

Gray			
Contamination	**Accuratezza**	**Precisione [0,10]**	**Richiamo [0,10]**
0	0.488	[0.1699, 1]	[1, 0.4284]
0.01	0.801	[0.345, 0.999]	[0.9975, 0.7785]
0.05	0.912	[0.547, 0.994]	[0.96, 0.907]
0.1	0.952	[0.728, 0.985]	[0.877, 0.961]
0.5	0.936	[1, 0.933]	[0.3925, 1]

Tabella 5.3: Valutazione del modello al variare della contamination in Gray

Il grafico 5.18 mostra la distribuzione delle etichette reali 0 e 10 all'interno delle classi 1 e -1 predette dall'Isolation Forest, si nota come il classificatore distribuisca bene i dati e separi le due classi. La tabella 5.4 riporta la distribuzione numerica dei dati del grafico in una pivot.

	Classe -1	**Classe 1**
Etichetta 0	49	351
Etichetta 10	3288	131

Tabella 5.4: Tabella pivot sulla distribuzione dei dati in Gray

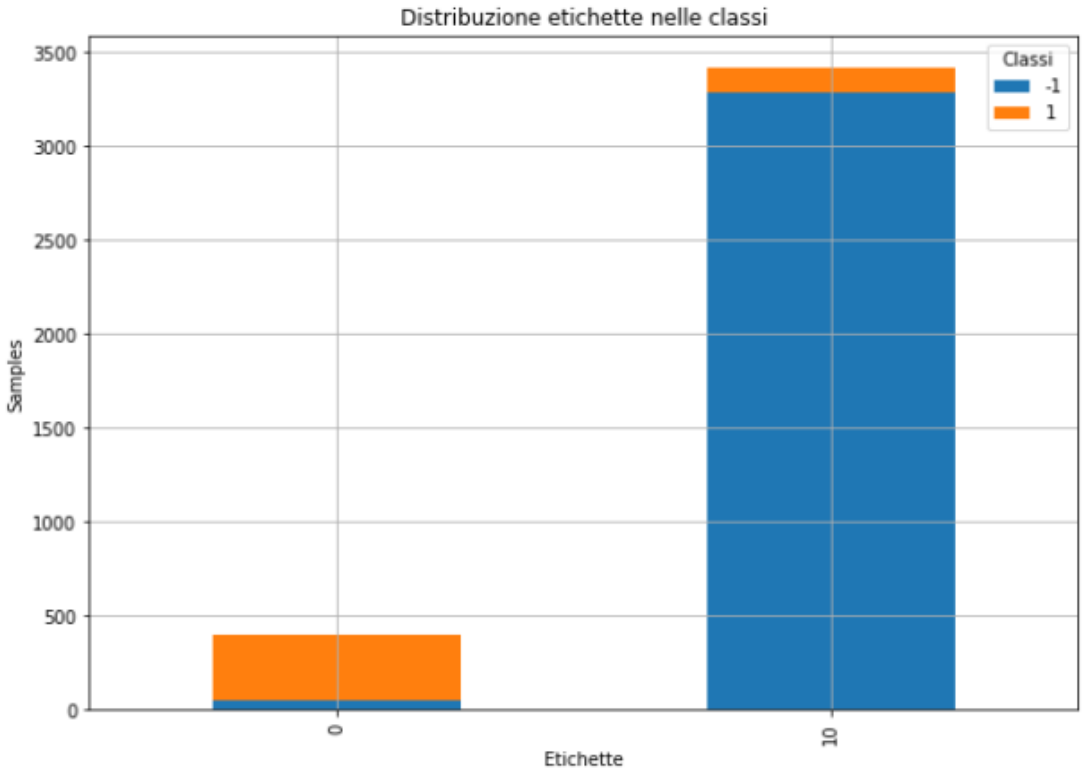


Figura 5.18: Distribuzione delle etichette 0 e 10 nelle classi 1 e -1 in Gray

Nei grafici presenti nelle figure 5.19 e 5.20 si nota l'andamento complementare delle percentuali delle classi nelle finestre di test. Fino alla finestra 400 circa si ha la sola presenza dell'etichetta 0 e di conseguenza una maggiore percentuale di punti assegnati dal classificatore alla classe 1. Si hanno valori attorno al 100% con una leggera variabilità che scende fino al 60% per indicare che qualche punto è stato erroneamente assegnato alla classe -1. Dalla finestra 400 in poi si ha un crollo delle percentuali di 1 all'ingresso nella finestra dell'etichetta 10. Anche in questo caso si nota un aumento della curva fino alla finestra 1500 per poi assestarsi allo 0%. Questa variazione di percentuale può essere dovuta alla variabilità interna della classe 10, come già visto nelle analisi precedenti, per cui il classificatore assegna erroneamente alla classe 1 elementi appartenenti all'etichetta 10. L'andamento delle curve mostra in modo supervisionato quando avviene il drift dei dati. Attraverso l'andamento decrescente della classe 1 (e viceversa crescente della classe -1), si può osservare infatti una deriva dei dati tramite l'aumento del numero di outlier che il

classificatore trova.

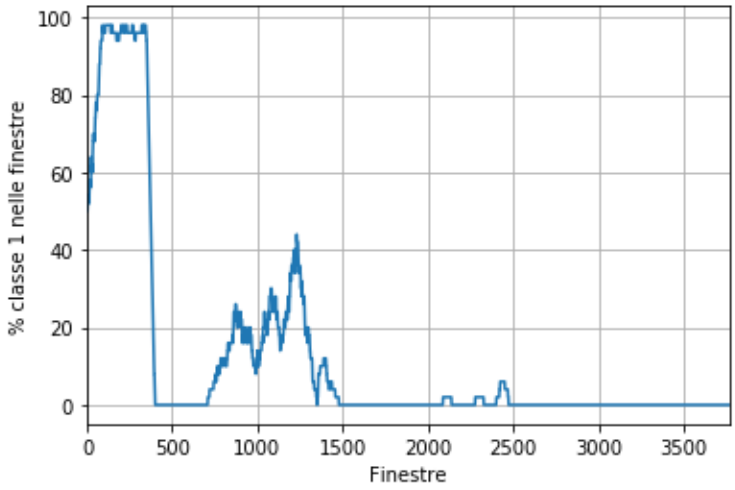


Figura 5.19: Andamento delle percentuali di 1 nelle finestre in Gray

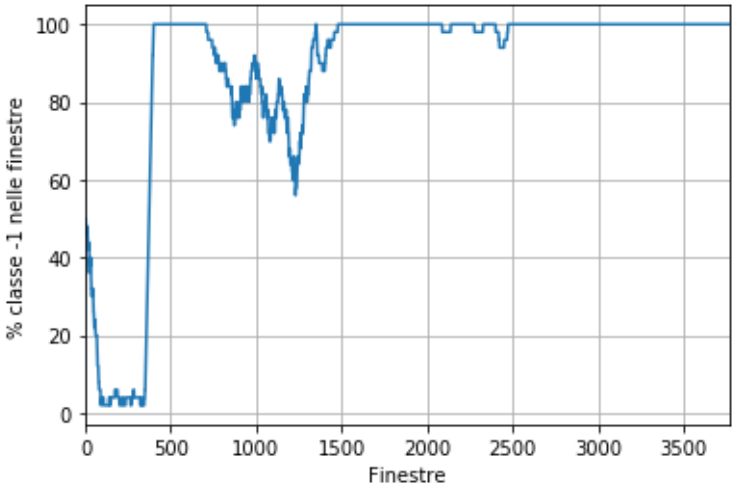


Figura 5.20: Andamento delle percentuali di -1 nelle finestre in Gray

Dataset White

Anche nel caso di *White* si è addestrato il modello variando i valori della contamination. Dalla tabella 5.5 si osservano i valori di accuratezza, precisione e richiamo e si può considerare come valore migliore 0.1. L'andamento dei valori delle metriche coincide con quello delle metriche in *Gray*: una contamination pari a 0 assume

che non ci siano outlier, di conseguenza i dati della classe 10 sono erroneamente assegnati alla classe 1 dei valori normali. Una precisione di 1 per l'etichetta 10 nel caso di contamination pari a 0 implica che i pochi valori assegnati alla classe -1 sono corretti, ma se si osserva il richiamo si nota che solo il 3% dei dati della 10 sono assegnati correttamente alla classe degli outlier. L'etichetta 0 è separata dalla 10 ma meno rispetto al dataset Gray, questo è visibile dallo Stacked Bar in figura 5.21 e dalla pivot 5.6 in cui un numero maggiore di elementi dell'etichetta 10 viene assegnato alla classe 1 erroneamente.

White			
Contamination	**Accuratezza**	**Precisione [0,10]**	**Richiamo [0,10]**
0	0.136	[0.1118, 1]	[1, 0.0305]
0.01	0.525	[0.185, 0.9967]	[0.9875, 0.4696]
0.05	0.830	[0.3855, 0.9903]	[0.935, 0.818]
0.1	0.941	[0.6755, 0.9854]	[0.885, 0.948]
0.5	0.933	[1, 0.9306]	[0.39, 1]

Tabella 5.5: Valutazione del modello al variare della contamination in White

Il grafico 5.21 e la tabella 5.6 si riferiscono alle distribuzione dei dati nelle etichette reali e in quelle predette considerando il valore migliore di contamination pari a 0.1.

	Classe -1	**Classe 1**
Etichetta 0	46	354
Etichetta 10	3105	170

Tabella 5.6: Tabella pivot sulla distribuzione dei dati in White

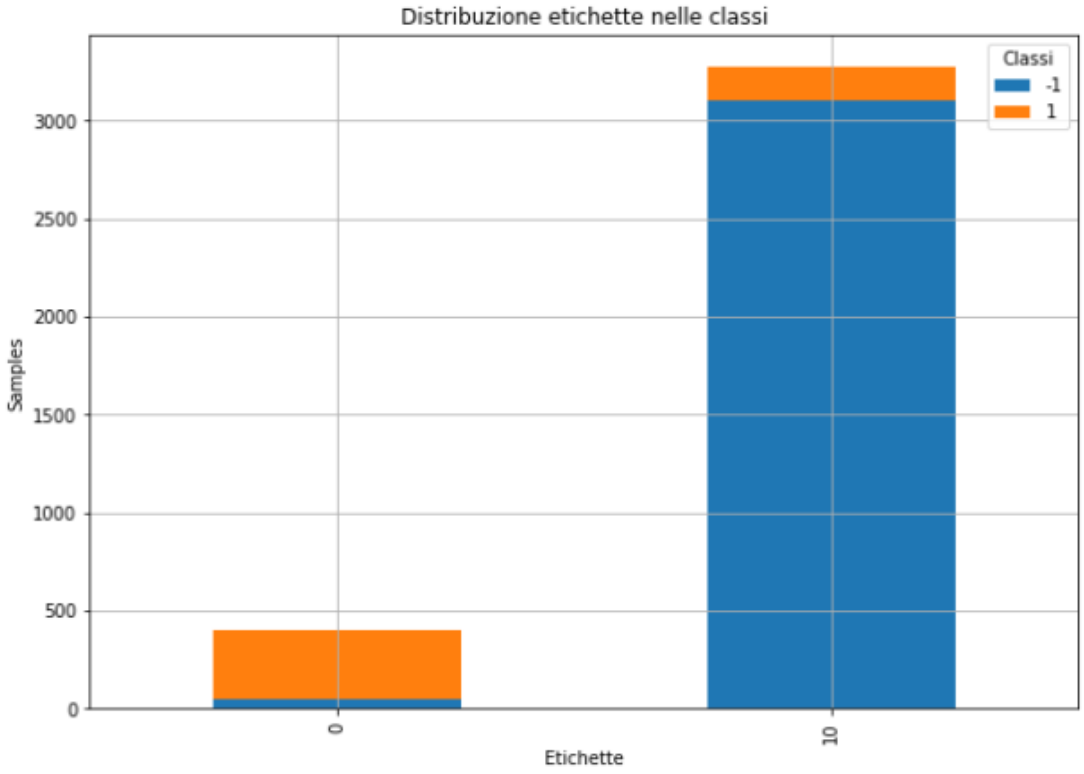


Figura 5.21: Distribuzione delle etichette 0 e 10 nelle classi 1 e -1 in White

Le figure 5.22 e 5.23 mostrano l'andamento delle percentuali delle classi nelle finestre di test per il dataset *White*. Anche in questo caso finché si ha la presenza della sola etichetta 0 la percentuale di classe 1 rimane attorno al 100% con una leggera variabilità fino ad un minimo dell'80%. Dalla finestra 400 in poi, il crollo delle percentuali di 1 con un maggior numero di outlier trovati conferma l'ingresso dell'etichetta 10 e la presenza di un drift nei dati. Dalla finestra 400 alla finestra 2800 circa, nonostante il crollo dei valori di percentuale di classe 1, si notano delle variazioni con picchi del 40% dovute al fatto che il classificatore assegna erroneamente i punti dell'etichetta sconosciuta dal train alla classe 1. Questo può dipendere da una separazione meno marcata tra le classi 0 e 10. L'andamento delle curve mostra in modo chiaro quando avviene il drift dei dati. Si tratta comunque di un'osservazione supervisionata basata sulle etichette assegnate dal classificatore. Sarà importante, nelle fasi successive dell'analisi, una valutazione non supervisionata della classificazione attuata dal modello attraverso lo studio

delle distribuzioni dei dati e della loro variazione, sfruttando gli indici di distanza intra-cluster.

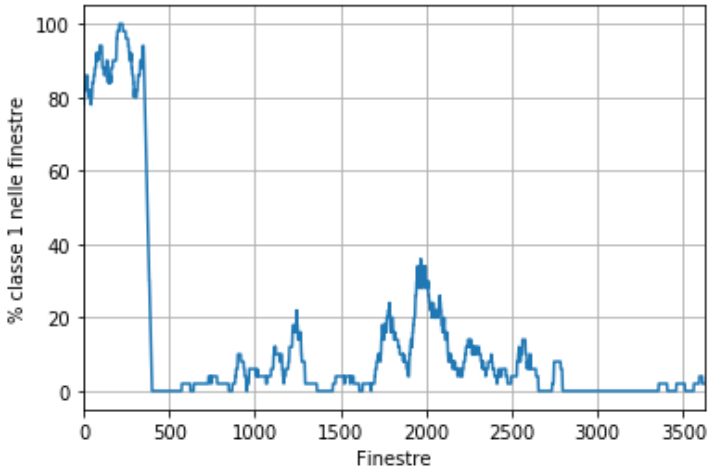


Figura 5.22: Andamento delle percentuali di 1 nelle finestre in White

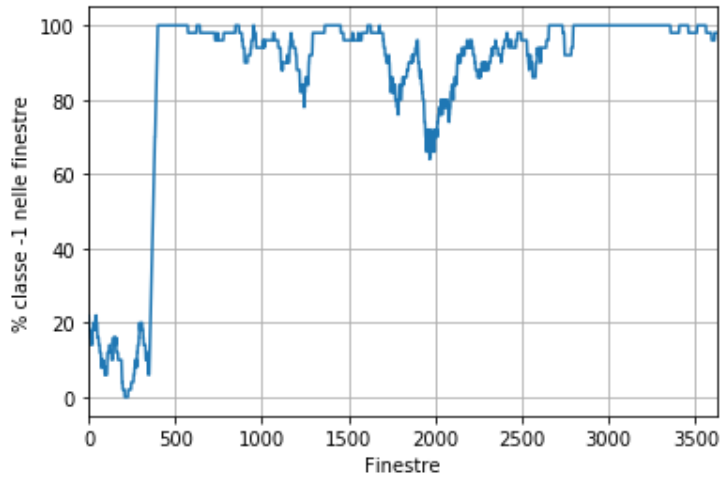


Figura 5.23: Andamento delle percentuali di -1 nelle finestre in White

5.2.3 Self assessment e Concept Drift detection

Una volta confermata la validità del modello per la gestione di questa tipologia di dati, si è deciso di applicare la metodologia esposta nel capitolo precedente sui dati a disposizione attraverso il classificatore dell'Isolation Forest. L'obiettivo di questa fase di analisi è capire attraverso una tecnica unsupervised quando avviene la deriva e di quanto il modello si degrada con l'arrivo di nuovi dati. Si è deciso di considerare come accettabile il funzionamento del robot quando la classe assegnata è la 0. Il modello è trainato su circa metà della classe 0. Per simulare un flusso di nuovi dati in entrata si è deciso di sfruttare la rimanente classe 0 e il 100% di una delle altre classi presenti nel dataset ma non ancora conosciuta dal modello. Il set di dati di test è iniettato nel classificatore a istanti di tempo specifici. Si è deciso di variare la cardinalità del campione di test definendo delle dimensioni di finestra variabile dal 20% all'80% della dimensione del train. Per ogni dimensione sono stati effettuati i test che seguono lo schema riportato in tabella 5.7. Nel test 0 è presente solamente la classe 0 già conosciuta, a partire dal test 1 in poi si ha una percentuale sempre maggiore della classe sconosciuta fino al 100% e una diminuzione complementare della classe 0 fino allo 0%.

Test t	**% Label 0**	**% New Label**
0	100%	0%
1	90%	10%
2	80%	20%
3	70%	30%
4	60%	40%
5	50%	50%
6	40%	60%
7	30%	70%
8	20%	80%
9	10%	90%
10	0%	100%

Tabella 5.7: Composizione pacchetti di test ad ogni tempo t

Al tempo t_0 quando viene addestrato il modello, l'Isolation Forest ha il parametro contamination settato a 0.1, che è il valore ottimale trovato per entrambi i dataset al passo precedente. La metodologia proposta prevede, in questa prima fase, il calcolo

delle distanze medie intra-cluster per ogni punto del train, sfruttando i metodi *sklearn.metrics* della libreria *Scikit-learn* [47]. Le distanze calcolate sono state ordinate, plottate in un curva e successivamente confrontate con quelle calcolate per ogni test. È stato fatto il calcolo dell'errore tra le curve al t_0 di train e ogni t_i trovando il degrado totale. Per ognuna delle classi 1 e -1 è stato poi approssimato il degrado moltiplicando il degrado totale per la percentuale di punti nella finestra di test appartenenti ad una determinata classe. Per gli esperimenti sono state considerate le altre due classi sconosciute:

1. Primo test: train su classe 0, test su classe 0 e 10
2. Secondo test: train su classe 0, test su classe 0 e 15

5.2.4 Dataset Gray

Nelle analisi sul dataset *Gray*, il set di training è costituito da 685 elementi che rappresentano circa il 53% della classe 0. Nella figura 5.24 è stata plottata la curva ordinata delle distanze intra cluster calcolate sui punti del set di training al tempo t_0. La curva arancione rappresenta la distanza euclidea media calcolata per ogni punto del training. Le curve blu e verde invece sono i valori della media +/- la deviazione standard in modo da tracciare una fascia di valori possibili delle distanze.

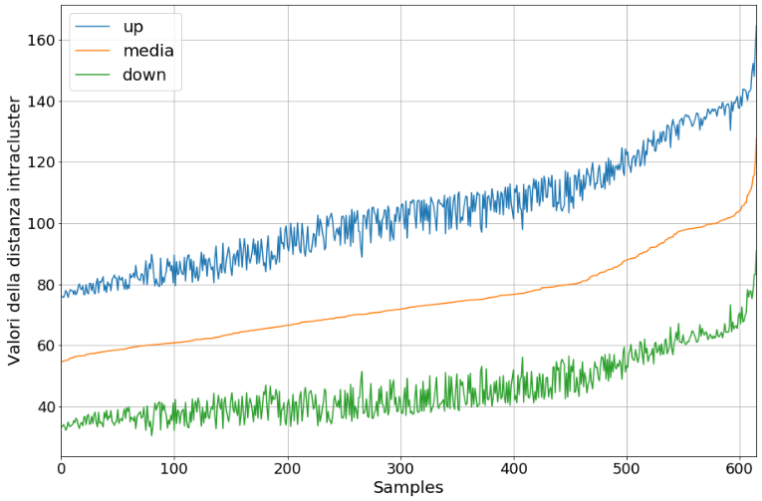


Figura 5.24: Gray: curva delle distanze intra cluster al tempo t_0

Train su classe 0, test su classe 0 e 10

Nel primo test la nuova classe in arrivo sconosciuta dal modello è la 10. Come primo risultato vengono mostrate delle matrici riassuntive sul degrado totale (considerando entrambe le classi 1 e -1) e sul degrado delle singole classi. Ogni cella rappresenta la percentuale di degrado per ogni test e per ogni dimensione della finestra. La matrice va letta come segue:

- Da sinistra verso destra: aumenta la dimensione della finestra di test, aumenta il numero di nuovi elementi inseriti e di conseguenza si nota un degrado crescente.

- Dall'alto verso il basso: aumenta in ogni test la percentuale di dati della classe sconosciuta (in questo caso la classe 10) e diminuisce la percentuale di dati della classe conosciuta (la classe 0)

Dalla legenda in figura 5.25 si nota che un colore chiaro dello sfondo della cella indica un valore più elevato di degrado. Nella prima riga il degrado è molto basso, questo è dovuto al fatto che il primo test viene effettuato su dati appartenenti solamente alla classe 0 su cui è stato fatto il training. I valori rimangono tutti intorno all' 1% e questo conferma il fatto che la classe 0 è molto coesa, con l'ingresso dei nuovi dati della classe conosciuta la distanza media intra-cluster non varia. Ciò implica che il classificatore ha lavorato bene e ha assegnato la maggior parte dei nuovi dati alla classe 1 trovando pochi outlier, come confermato dalla matrice heatmap della classe -1 in cui la prima riga ha valori intorno allo 0.

Dalla seconda riga in poi è introdotta una percentuale sempre maggiore di classe 10 nel pacchetto di test. Come previsto la percentuale di degrado aumenta ma non in modo eccessivo. L'aumento progressivo avviene sia sulle colonne che sulle righe, fino a convergere al valore maggiore sull'angolo in basso a destra. I valori bassi di degrado sono dovuti all'elevata coesione dei dati nelle etichette 0 e 10 e alla loro separazione che permette al classificatore di lavorare bene e assegnare correttamente le classi 1 e -1. Tuttavia il degrado totale rispetto alle distanze al tempo t_0 è evidente. Quando si ha una finestra di test di quantità pari all'80% del totale dei dati usati come train con tutti i dati appartenenti alla classe 10 si ha un aumento di dati etichettati come outliers, le distanze intra-cluster per i punti -1 crescono e di conseguenza il degrado raggiunge il suo picco.

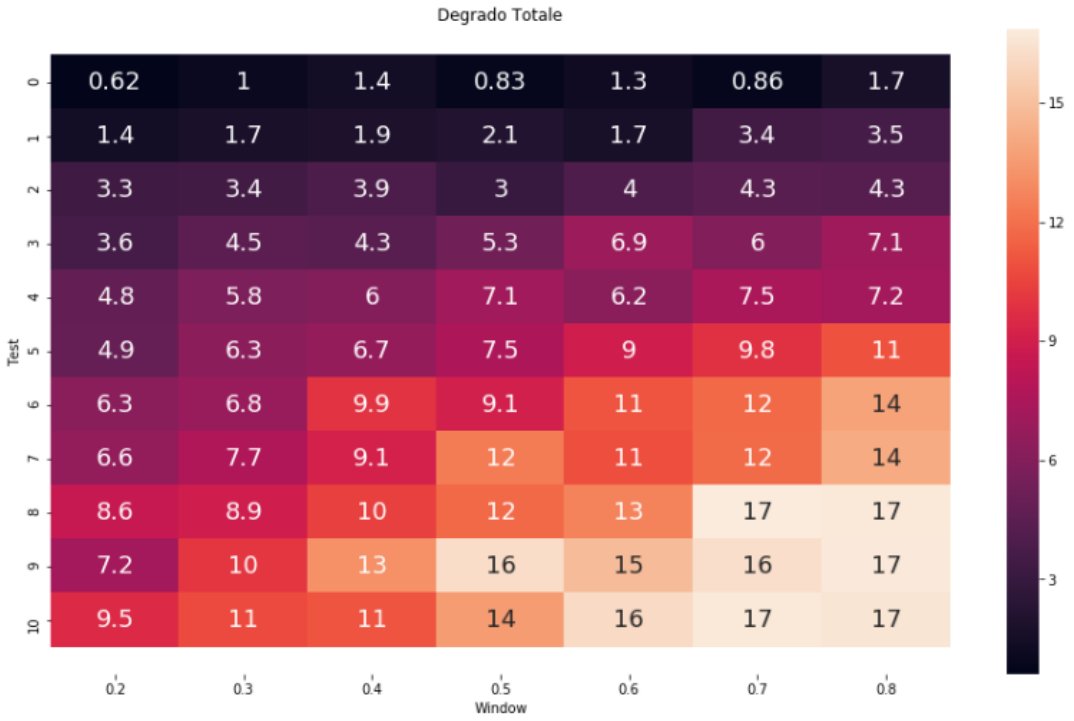


Figura 5.25: Gray: matrice degrado totale

Come è possibile vedere dalle figure 5.26 e 5.27 il contributo maggiore al degrado totale è dato dalla classe -1. Questo conferma l'approccio teorico dell'Isolation Forest: una volta definiti i confini della classe 1 con il parametro contamination, al suo interno entrano tutti quei punti che si trovano nel perimetro delineato dunque la distanza media varia di poco e la classe rimane coesa. Nella classe -1, viceversa, sono assegnati quasi tutti i punti della classe 10 e le distanze medie sono superiori a quelle della classe 1. Il classificatore dell'Isolation forest è robusto: predice correttamente cosa inserire nella classe 1 e inserisce nella classe -1 tutto ciò che non rientra nei confini e che può essere classificato come outlier, causando un forte aumento delle distanze intra cluster della classe -1.

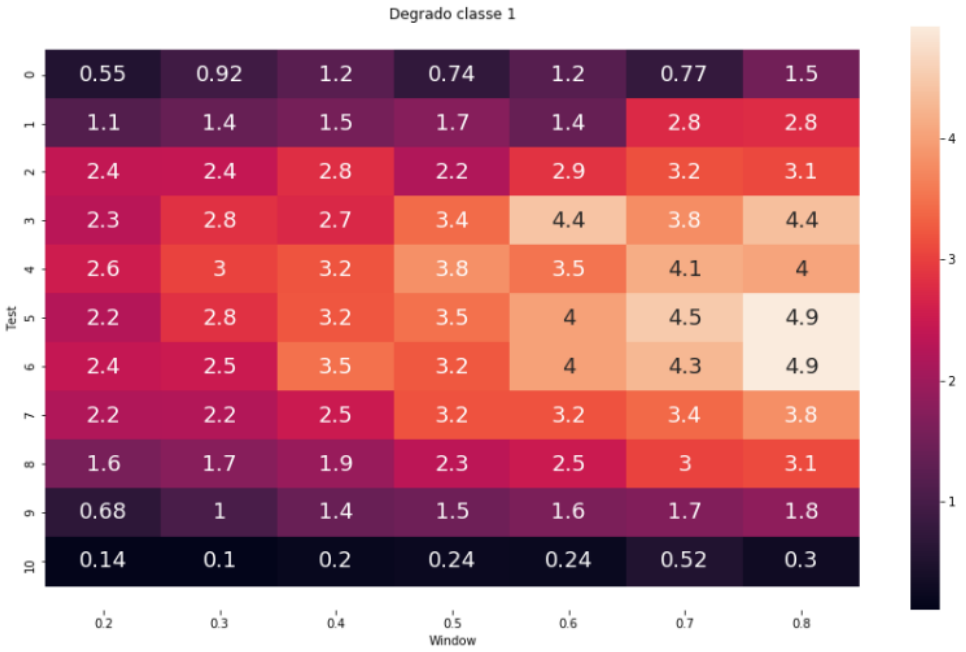


Figura 5.26: Gray: matrice degrado classe 1

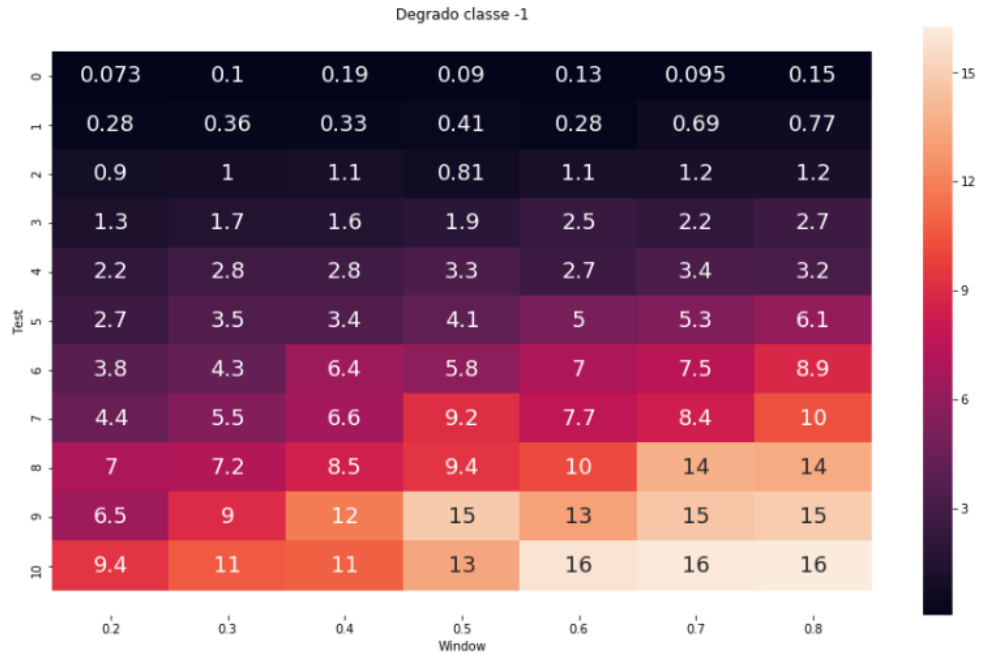


Figura 5.27: Gray: matrice degrado classe -1

Per un'analisi più di dettaglio è stato selezionato l'ultimo timestamp di test nella finestra 0.8 che presenta il valore massimo di degrado. In figura 5.28 è possibile analizzare le curve delle distanze intra-cluster calcolate al momento di train (base) e successivamente con l'arrivo di nuovi dati (profilo degradato). Confrontando le curve si nota che i primi valori sono gli stessi, dal campione 200 circa in poi i valori della curva degradata aumentano perché aumentano sempre più le distanze intra-cluster. Il picco si nota in corrispondenza degli ultimi campioni in cui, mentre la distanza della curva base si arresta a circa 150, la curva degradata supera il valore 700. Presumibilmente gli ultimi punti della curva sono quelli appartenenti alla classe -1, confermando l'ipotesi che il degrado trovato sia da imputare alla classe degli outlier e che il modello abbia riconosciuto l'arrivo di nuovi dati diversi da quelli di train classificandoli tutti come non appartenenti alla classe di train.

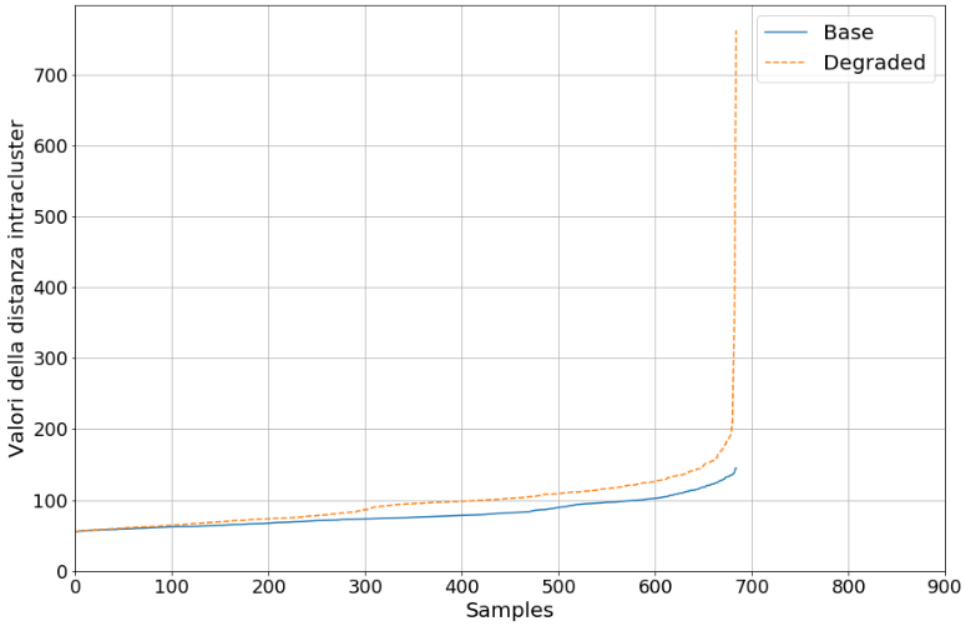


Figura 5.28: Gray: curva distanza intra-cluster degradata (Window 0.8, test 10)

Le figure 5.29 e 5.30 mostrano nel dettaglio l'andamento del degrado per ogni classe e per ogni finestra di test. Per quanto riguarda la classe 1, si mantiene lo stesso andamento in ogni dimensione di finestra di test. In particolare è presente prima un andamento crescente e infine decrescente, questo è dovuto al fatto che nei primi test con l'ingresso dei dati sconosciuti si ha un aumento generale delle

distanze intra-cluster. Negli ultimi test invece la classe 1 è sempre meno presente poiché il classificatore trova più outliers e di conseguenza l'aumento del degrado e delle distanze intra-cluster rispetto alla curva di train si nota maggiormente nella classe -1. Il degrado totale segue l'andamento di quello della classe -1, come era già visibile con le heatmap. Nella finestre di test di dimensione 0.7 e 0.8 si nota come il valore di degrado si assesti e non aumenti più dal test 8 in quanto il drift è già avvenuto e non peggiora più.

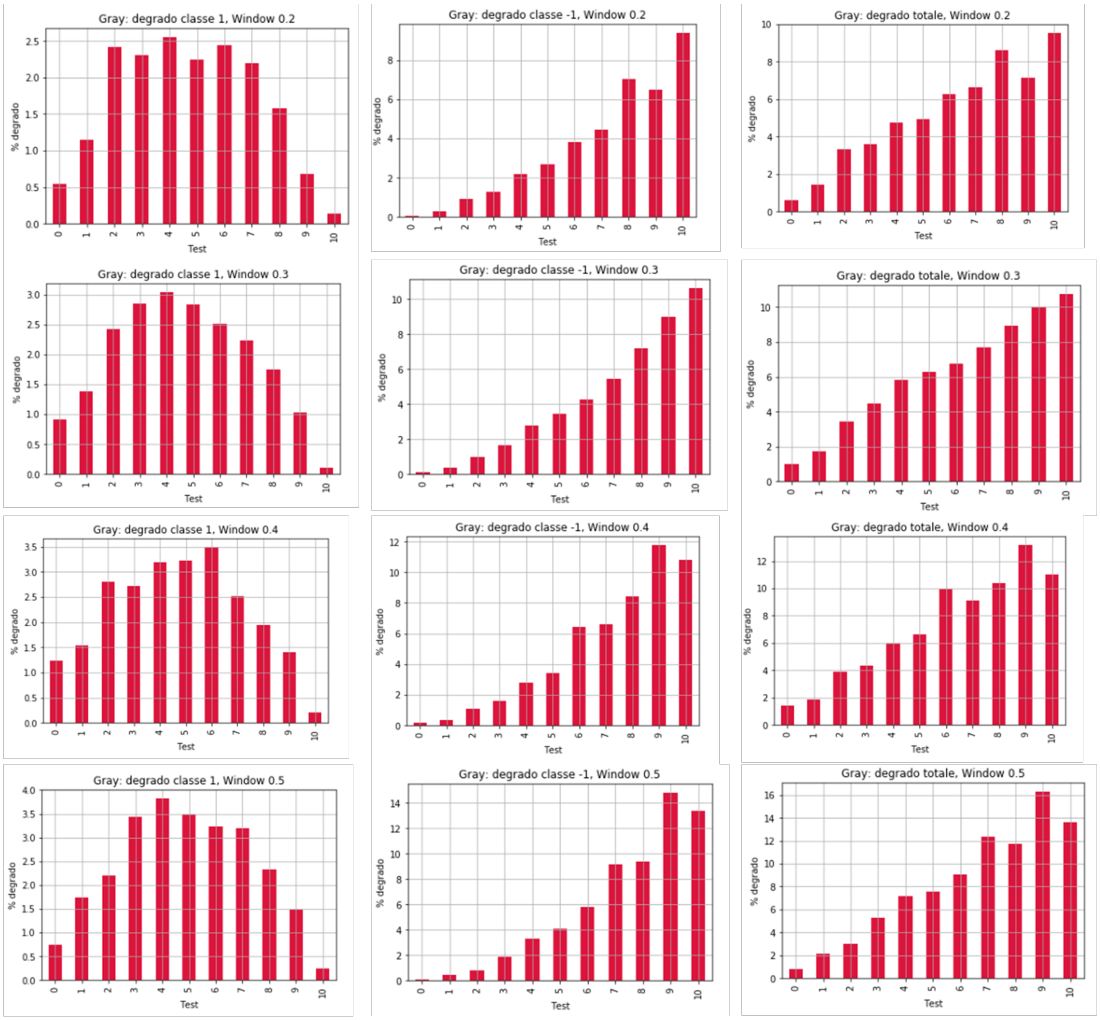


Figura 5.29: Gray: percentuale degradi Window 0.2, 0.3, 0.4, 0.5

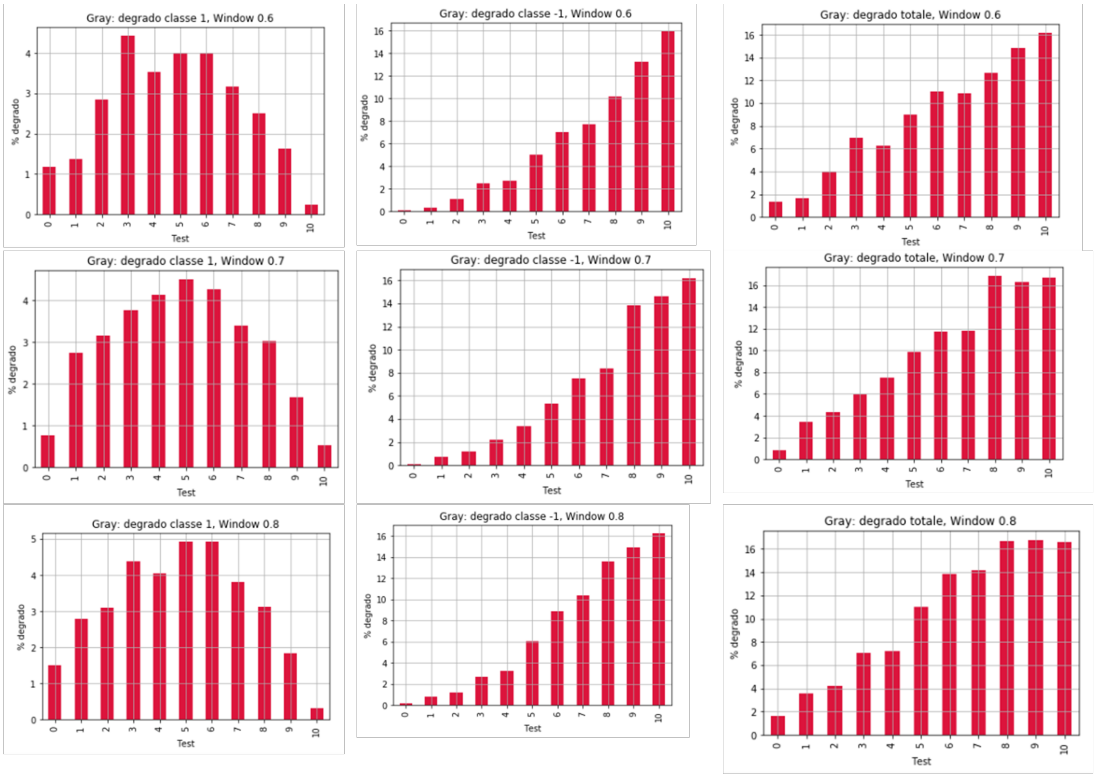


Figura 5.30: Gray: percentuale degradi Window 0.6, 0.7, 0.8

I risultati ottenuti sono più evidenti e trattati nel dettaglio in figura 5.31 in cui sono plottate le curve delle distanze nella classe -1 per ogni test. I valori variano di molto fino a raggiungere picchi oltre 600. Questo giustifica la scelta del modello nel classificarli come outliers. I valori elevati delle distanze spingono l'aumento del degrado rispetto alla distribuzione nel train. Al contrario nella classe 1 in figura 5.32 i valori delle distanze si mantengono tutti intorno allo stesso range compreso tra 50 e poco più di 130, questo perché la coesione della classe rimane la stessa anche se ad essa sono assegnati nuovi punti. Da queste analisi si deduce che è possibile individuare il drift dei dati in modo non supervisionato osservando la variazione della distribuzione nella classe degli outlier -1.

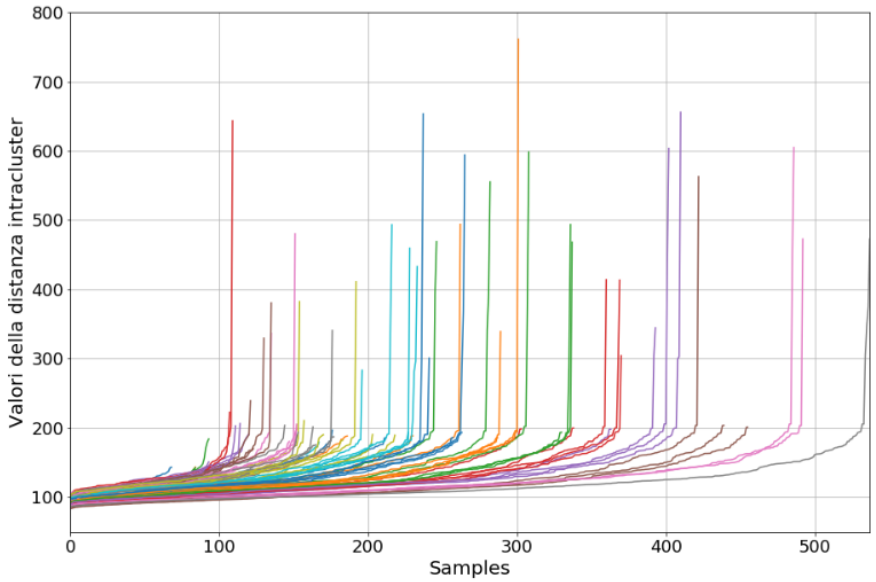


Figura 5.31: Gray: curve delle distanze intra cluster -1 per ogni tempo t

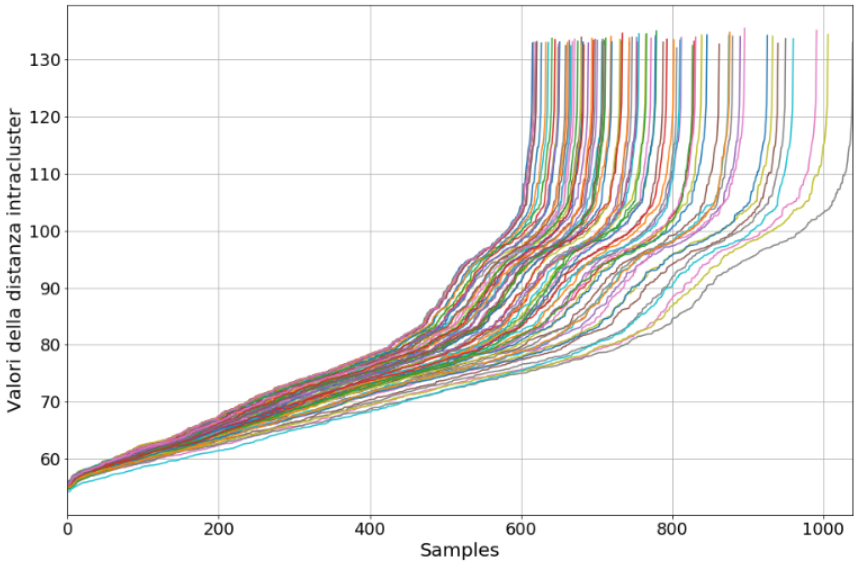


Figura 5.32: Gray: curve delle distanze intra cluster 1 per ogni tempo t

Train su classe 0, test su classe 0 e 15

Nel secondo test la nuova classe in arrivo è la 15. Di seguito saranno riportati sinteticamente i risultati ottenuti. In generale, come visto nel corso delle analisi preliminari, il comportamento della classe 0 è molto simile a quello della classe 15 pertanto i valori di degrado trovati sono più bassi e lo stesso si può dire per le distanze medie tra i punti. Come primo risultato vengono mostrate le matrici riassuntive sul degrado totale (considerando entrambe le classi 1 e -1) e sul degrado delle singole classi. Nella prima riga il degrado è molto basso, come si è visto nel test precedente. Dalla seconda riga in poi è introdotta una percentuale sempre maggiore di classe 15 nel pacchetto di test. Come previsto la percentuale di degrado aumenta ma non in modo eccessivo. L'aumento progressivo avviene solo lungo le colonne, per cui l'aumento della dimensione del test non influisce molto sul degrado probabilmente perché i dati della classe 15 sono molto coesi tra loro.

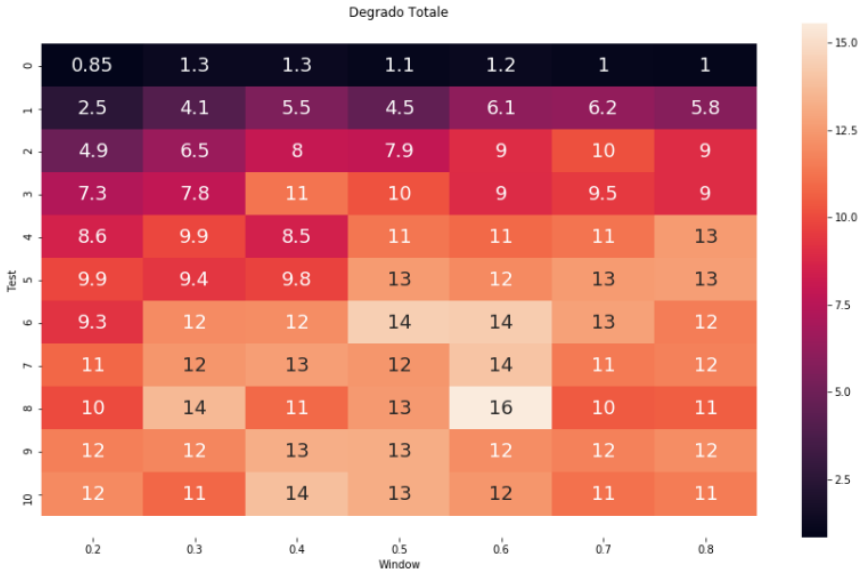


Figura 5.33: Gray: matrice degrado totale con test su classe 15

Come è possibile vedere dalle figure 5.34 e 5.35 il contributo maggiore al degrado totale è ancora una volta dato dalla classe -1. Nella classe 1 si nota un aumento e poi un decremento del degrado. Dai test 0 a 5 si ha in generale l'aumento delle distanze intra-cluster con l'arrivo della classe 15, dal 5 al 10 la diminuzione del degrado è legata all'abbassamento di percentuale di classe 0 nel set di test.

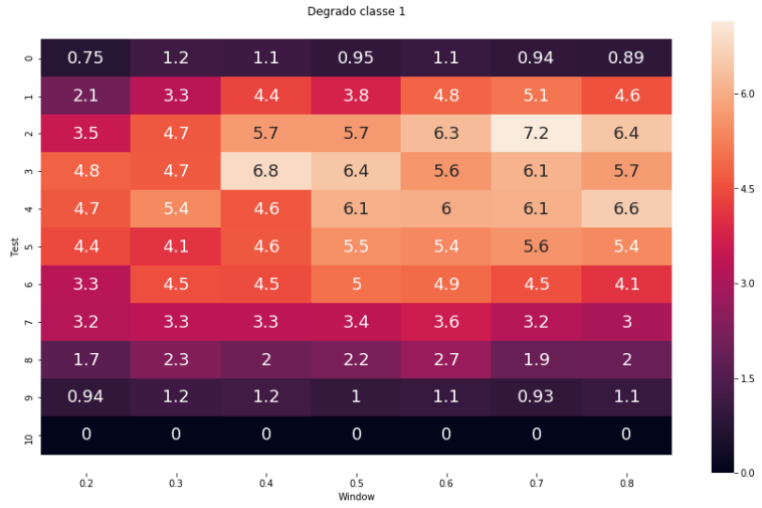


Figura 5.34: Gray: matrice degrado classe 1 con test su classe 15

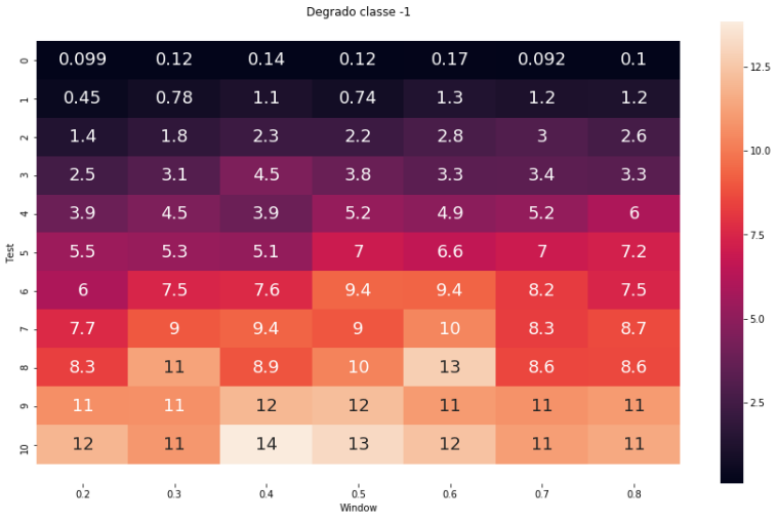


Figura 5.35: Gray: matrice degrado classe -1 con test su classe 15

La coesione della classe 15 è più evidente osservando il comportamento delle curve delle distanze nella classe -1 degli outlier per ogni test (Figura 5.36). Le curve raggiungono anche il valore 300 ma si mantengono più o meno nello stesso range senza raggiungere particolari picchi. Le distanze della classe -1 superano i valori di quelle per la classe 1 (Figura 5.37). Questo giustifica la scelta del modello nel classificarli come outliers e l'aumento del degrado rispetto alla distribuzione nel train. Nella classe 1 in figura 5.37 i valori delle distanze si mantengono tutti intorno allo stesso range compreso tra 50 e poco più di 130, questo perché la coesione della classe rimane la stessa anche se ad essa sono assegnati nuovi punti. L'andamento delle curve della classe 1 conferma quello che la stessa classe ha nel caso del test su classe 0 e 10, visto al punto precedente.

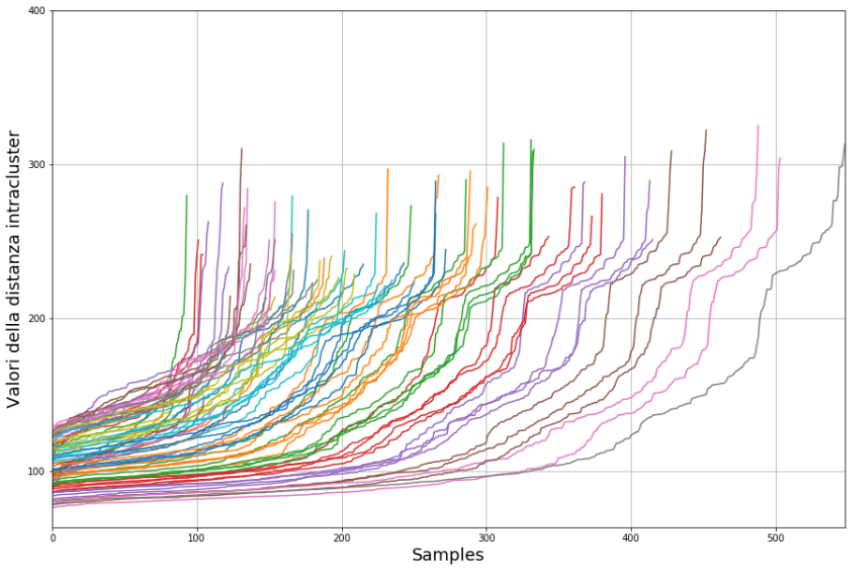


Figura 5.36: Gray (con test su classe 15): curve delle distanze intra cluster -1 per ogni tempo t

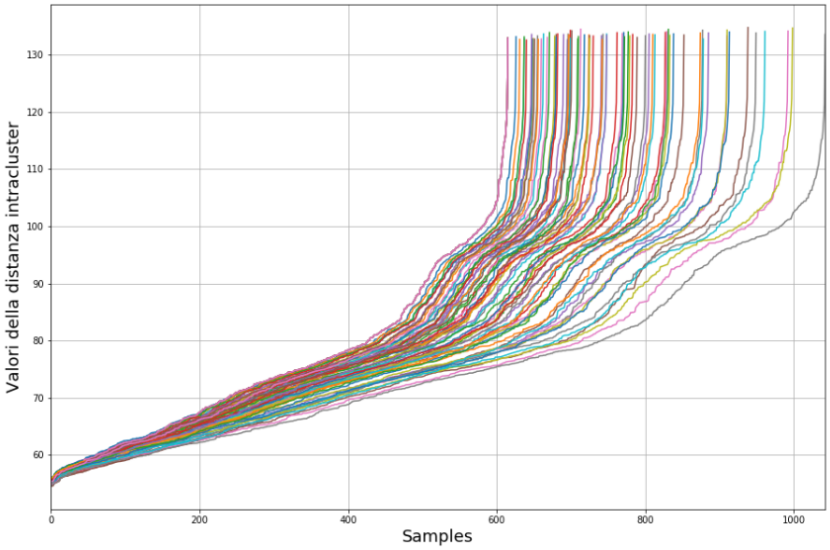


Figura 5.37: Gray (con test su classe 15): curve delle distanze intra cluster 1 per ogni tempo t

5.2.5 Dataset White

Nelle analisi sul dataset *White* il set di training è costituito da 616 elementi che rappresentano circa il 50% della classe 0. Nella figura 5.38 è rappresentata la curva ordinata delle distanze intra cluster calcolate sui punti del set di training al tempo t_0. Anche qui per tracciare i valori possibili di distanze è stata rappresentata la curva della media (al centro) e quella dei valori della media +/- la deviazione standard. Si può notare una variabilità leggermente inferiore rispetto al dataset *Gray*.

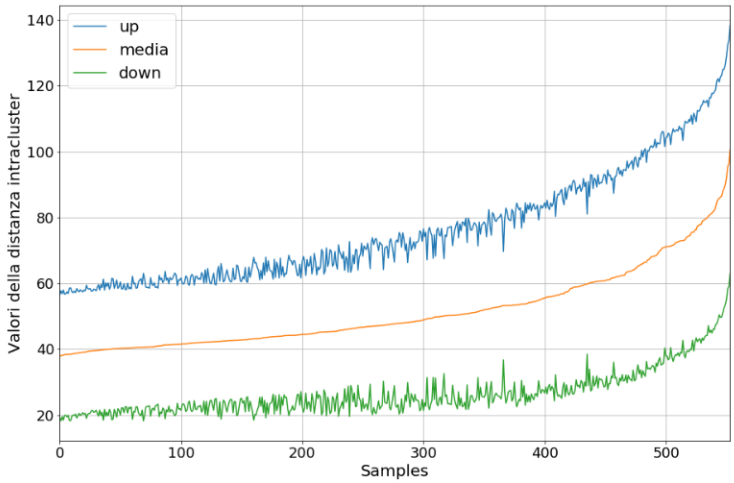


Figura 5.38: White: curva delle distanze intra cluster al tempo t_0

Train su classe 0, test su classe 0 e 10

La nuova classe in arrivo è la 10, i risultati delle matrici riassuntive sul degrado totale e sul degrado delle singole classi 1 e -1 confermano l'andamento già visto nel caso del dataset *Gray*:

- Da sinistra verso destra: aumenta la dimensione della finestra di test, aumenta il numero di nuovi elementi inseriti e di conseguenza si nota un degrado crescente.

- Dall'alto verso il basso: aumenta in ogni test la percentuale di dati della classe sconosciuta (in questo caso la classe 10) e diminuisce la percentuale di dati della classe conosciuta (la classe 0)

Nella prima riga il degrado è molto basso con valori che rimangono intorno a 1, il primo test viene effettuato su dati appartenenti solamente alla classe su cui è stato fatto il training. Questo conferma il fatto che, anche in questo caso, la classe 0 è molto coesa, con l'ingresso dei nuovi dati della classe conosciuta la distanza media intra-cluster non varia. Con una percentuale sempre maggiore di classe 10 sconosciuta nel pacchetto di test il degrado aumenta ma non in modo eccessivo. L'aumento progressivo avviene sia sulle colonne che sulle righe, fino a convergere al valore maggiore sull'angolo in basso a destra. In generale si possono notare dei valori di degrado più bassi rispetto al caso di *Gray* e questo probabilmente è dovuto al fatto che i dati di *White* sono più coesi, le distanze intra-cluster sono inferiori e anche le classi 0 e 10 sono meno separate. L'Isolation forest ancora una volta si dimostra un classificatore valido che riesce a circoscrivere i dati nelle due classi 1 e -1. Il degrado totale rispetto alle distanze al tempo t_0 è evidente quando aumenta la dimensione della finestra di test e i dati al suo interno sono tutti appartenenti alla classe 10 sconosciuta con conseguente aumento delle distanze intra-cluster del valore di degrado. Anche in questo caso il contributo maggiore al degrado totale è dato dalla classe -1 (figura 5.41). Nella classe degli outlier, come avviene per la classe 1, le distanze medie intra-cluster assumono valori inferiori rispetto al dataset *Gray*, questo conferma una maggiore coesione dei dati e di conseguenza un degrado totale inferiore. I risultati della heatmap nella classe 1 non hanno un andamento convergente come nel caso della classe -1, quindi non è possibile, al suo interno, individuare il drift dei dati come accade invece osservando la classe degli outlier. All'interno della classe 1 il classificatore inserisce correttamente i dati appartenenti all'etichetta 0 escludendo quelli della 10.

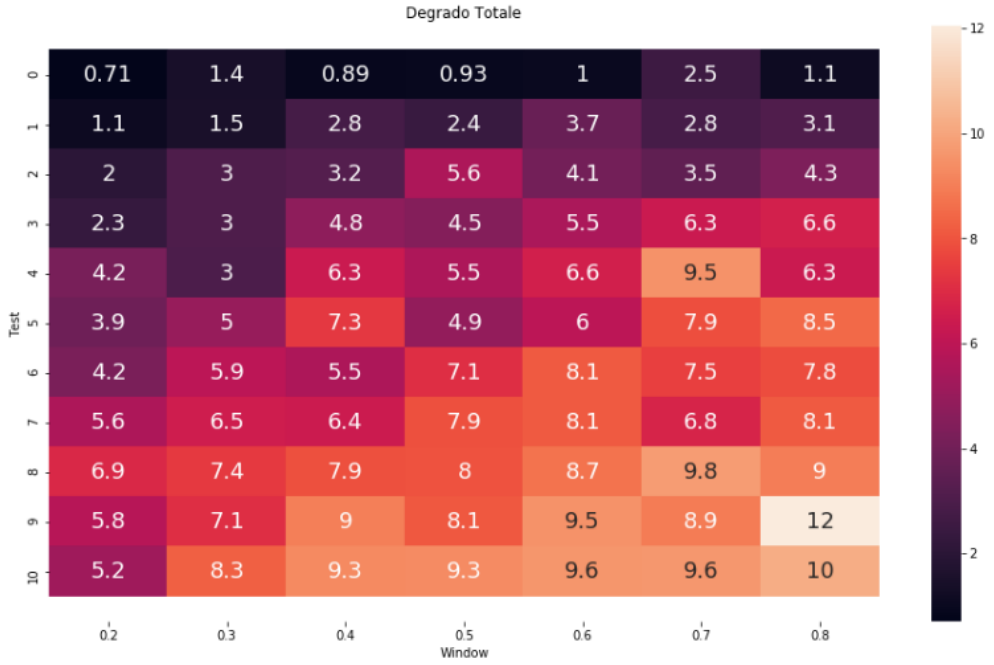


Figura 5.39: White: matrice degrado totale

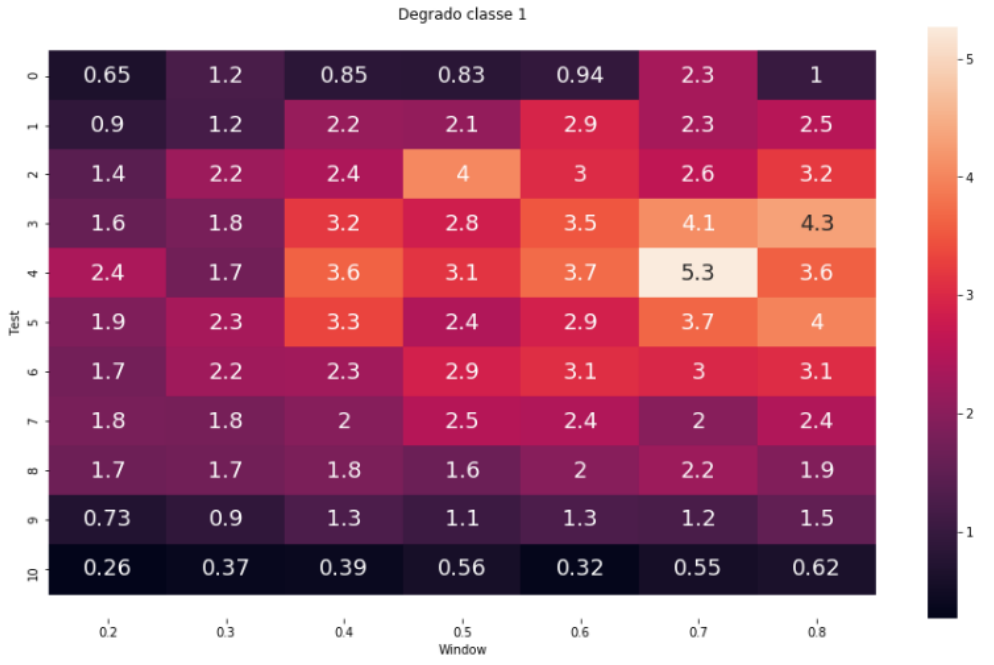


Figura 5.40: White: matrice degrado classe 1

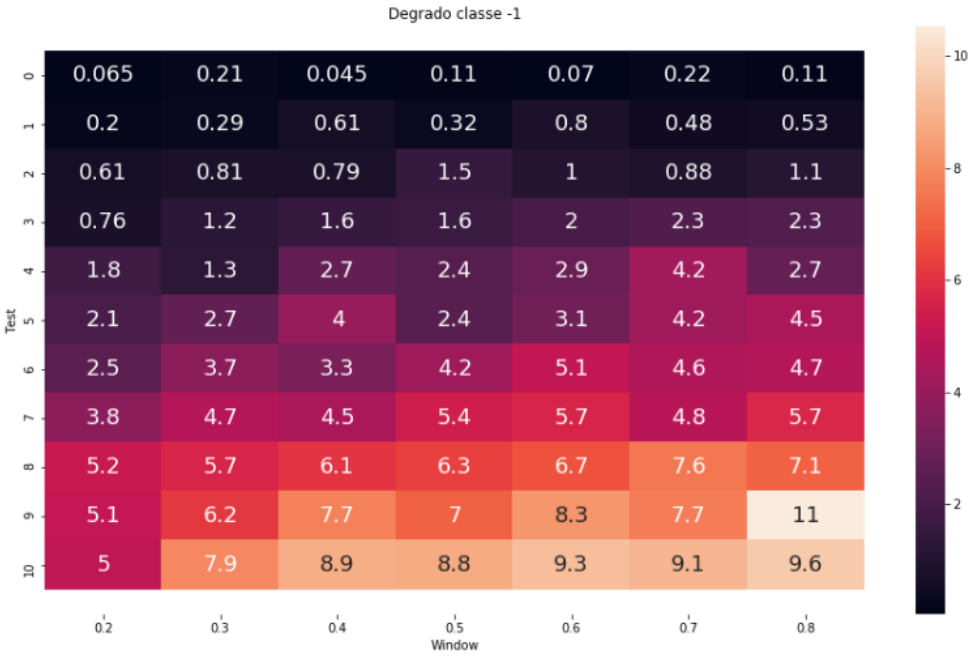


Figura 5.41: White: matrice degrado classe -1

In figura 5.42 è possibile analizzare la curva delle distanze intra-cluster calcolate al momento di train (base) e il suo profilo degradato, corrispondente all'ultimo timestamp di test nella finestra 0.8 con il valore massimo di degrado. Anche in questo caso nei primi campioni le curve si sovrappongono, dal campione 100 circa in poi i valori della curva degradata aumentano per le maggiori distanze intra-cluster. La differenza tra le curve è minore rispetto a quanto visto per *Gray* ma allo stesso tempo è possibile osservare, in corrispondenza degli ultimi campioni, un aumento radicale delle distanze nella curva degradata fino a 350, mentre la distanza della curva base si arresta a circa 150. Gli ultimi punti della curva appartengono alla classe -1, mostrando una maggiore presenza di outliers nel dataset di test, che conferma una variazione nella distribuzione dei dati e di conseguenza la deriva dei dati.

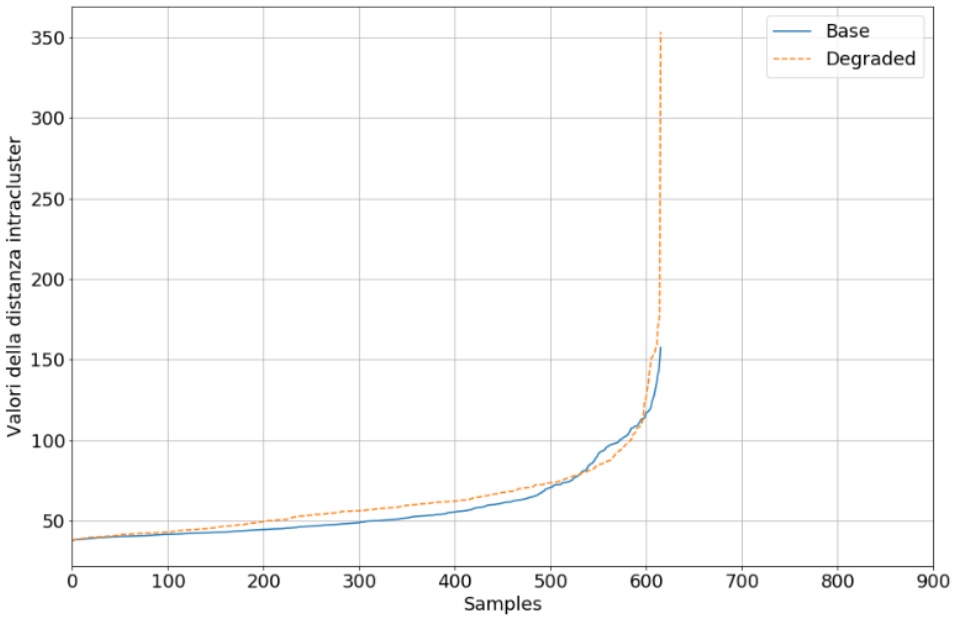


Figura 5.42: White: curva distanza intra-cluster degradata (Window 0.8, test 10)

Dai grafici della degradazione del modello nelle figure 5.43 e 5.44 si nota ancora una volta l'andamento crescente del degrado totale di pari passo con quello del degrado della classe -1. All'aumentare della presenza di classe 10 nel test il classificatore assegna un numero sempre maggiore di elementi alla classe -1, questo fa aumentare le distanze medie intra-cluster e di conseguenza il degrado totale del modello. In generale nella classe 1 si nota un andamento prima crescente e poi decrescente: nei primi test la presenza della classe 1 supera quella della classe -1 e di conseguenza gran parte del degrado totale è assegnato alla classe 1. Il fatto che nei grafici il degrado della classe 1 aumenti non vuol dire che è presente una deriva dei suoi dati e un aumento delle sue distanze intra cluster, perché i risultati sono falsati dalla % di 1 nel test. Anche in questo caso ciò che conta è quello che avviene nella classe -1.

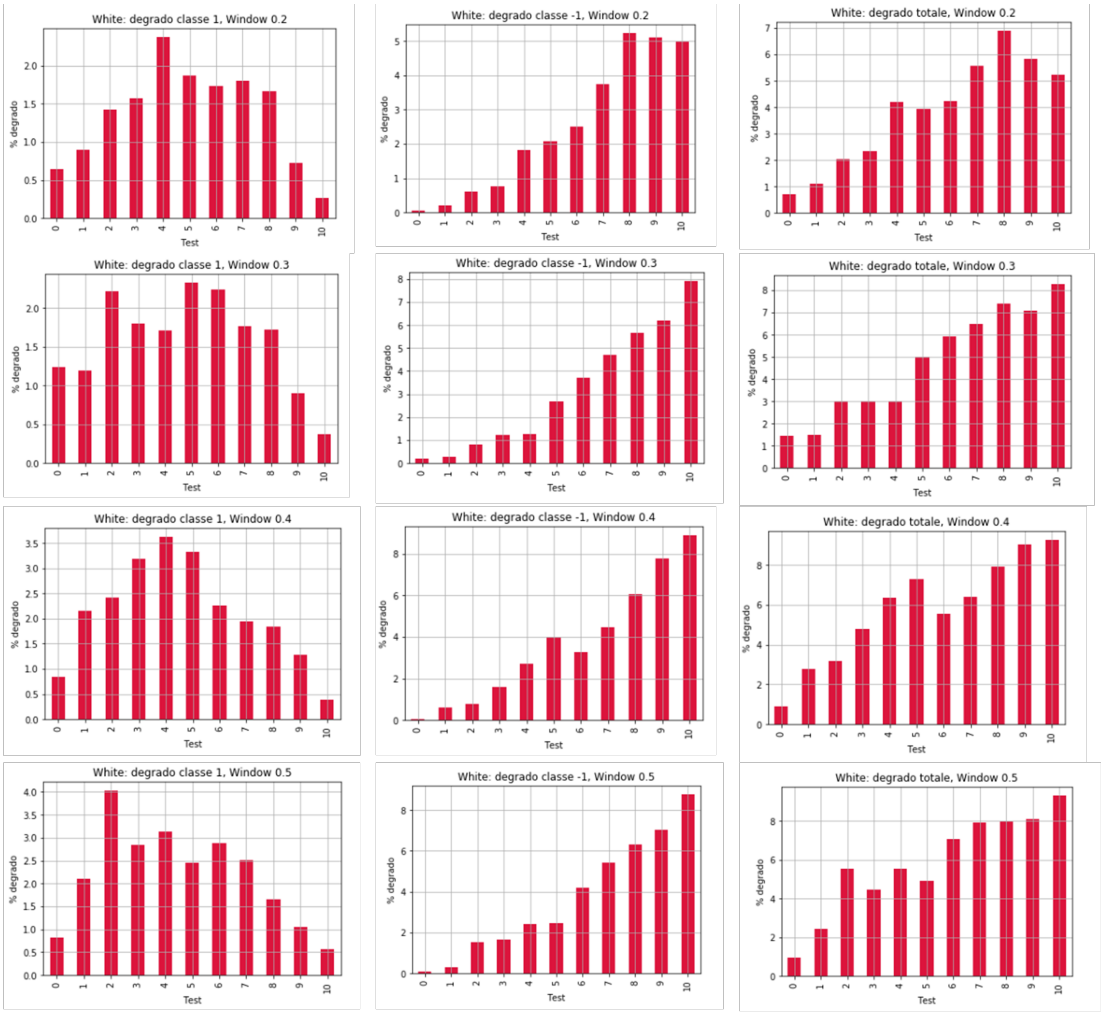


Figura 5.43: White: percentuale degradi Window 0.2, 0.3, 0.4, 0.5

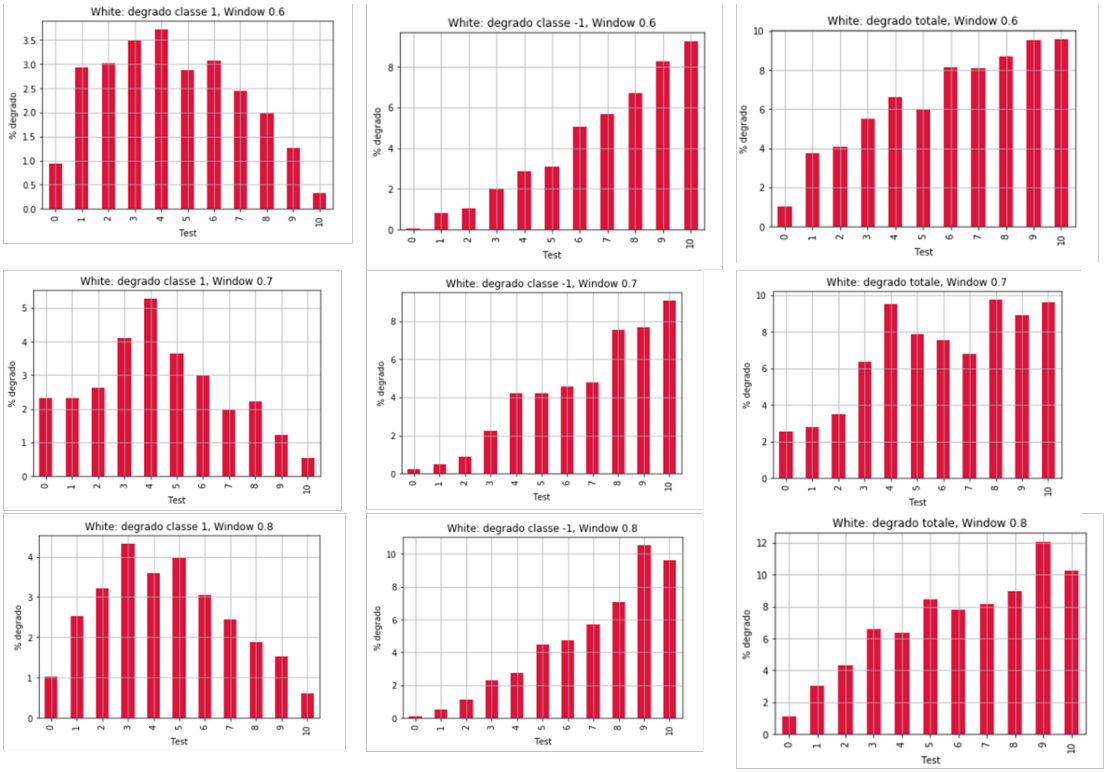


Figura 5.44: White: percentuale degradi Window 0.6, 0.7, 0.8

Anche in questo caso, per ottenere risultati più evidenti, sono stati separati per ogni test i punti della classe 1 e quelli della classe -1, sono state calcolate le distanze intra-gruppo. In figura 5.45 sono plottate le curve delle distanze nella classe -1 per ogni test. In generale le distanze medie tra i punti sono inferiori rispetto a *Gray* e questo giustifica la disposizione dei punti nello spazio tridimensionale. Si nota che i valori delle distanze degli outlier variano di molto raggiungendo picchi di oltre 350, anche se mantengono valori inferiori rispetto a quelle dei punti di *Gray*. Ritroviamo così i risultati già visti nei grafici precedenti che confermano una maggiore coesione per i dati del dataset *White*. Nella classe 1 in figura 5.32 i valori delle distanze si mantengono tutti intorno allo stesso range compreso tra 40 e 110 confermando che le distanze tra i punti dei test non variano e rimangono pressoché costanti all'aumentare del numero dei punti nella classe 1.

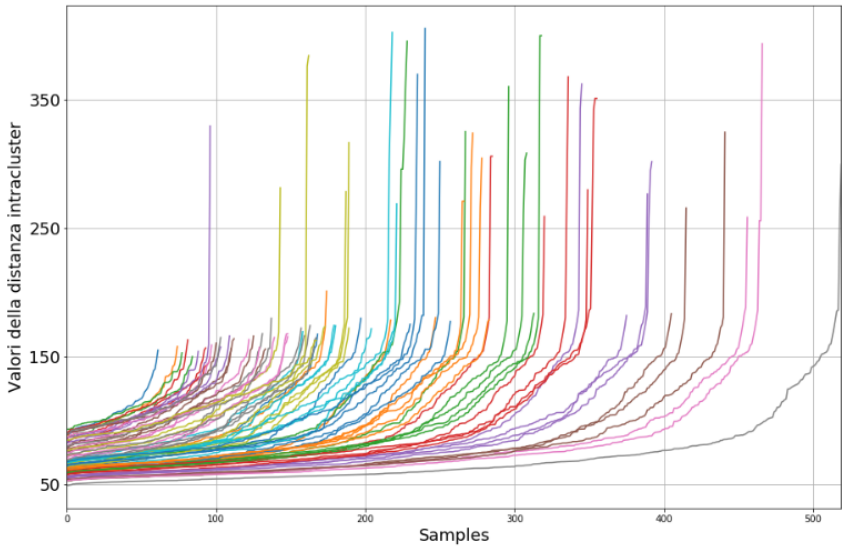


Figura 5.45: White: curve delle distanze intra cluster -1 per ogni tempo t

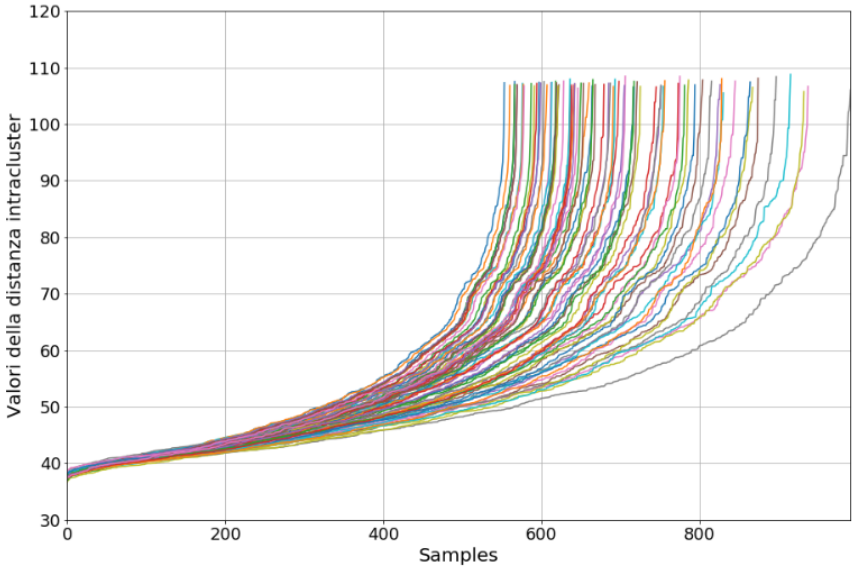


Figura 5.46: White: curve delle distanze intra cluster 1 per ogni tempo t

Train su classe 0, test su classe 0 e 15

Come già fatto per *Gray* si riportano sinteticamente i risultati ottenuti nel secondo test con la nuova classe 15. In questo caso sembrerebbe che la classe 0 sia molto diversa dalla 15 pertanto i valori di degrado trovati sono elevati e lo stesso si può dire per le distanze medie tra i punti. Le matrici riassuntive sul degrado totale e sul degrado delle singole classi mostrano un andamento come quello del test precedente, in cui le percentuali di degrado aumentano sia sulle righe che sulle colonne convergendo sull'angolo in basso a destra. Dalla seconda riga in poi è introdotta una percentuale sempre maggiore di classe 15 nel pacchetto di test. In questo caso l'aumento del degrado è da assegnare sia al progressivo arrivo della nuova classe 15, sia all'aumento della dimensione della finestra di test.

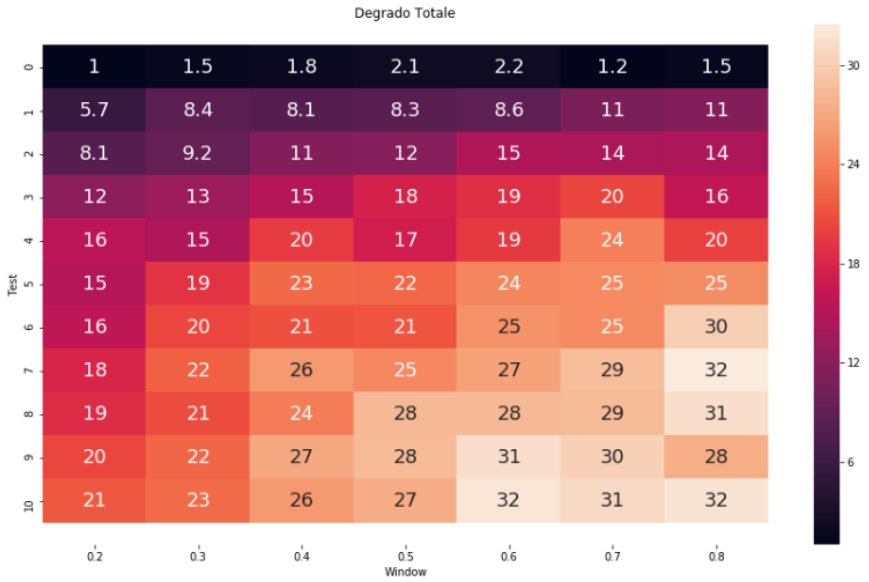


Figura 5.47: White: matrice degrado totale con test su classe 15

Dalle figure 5.48 e 5.49 si nota che il contributo maggiore al degrado totale è ancora una volta dato dalla classe -1. Dai test 0 a 5 si ha l'aumento del degrado dovuto all'aumento delle distanze intra-cluster per l'arrivo della classe 15, dal 5 al 10 la diminuzione del degrado è legata all'abbassamento di percentuale di classe 0 nel set di test, fino ad arrivare allo 0%.

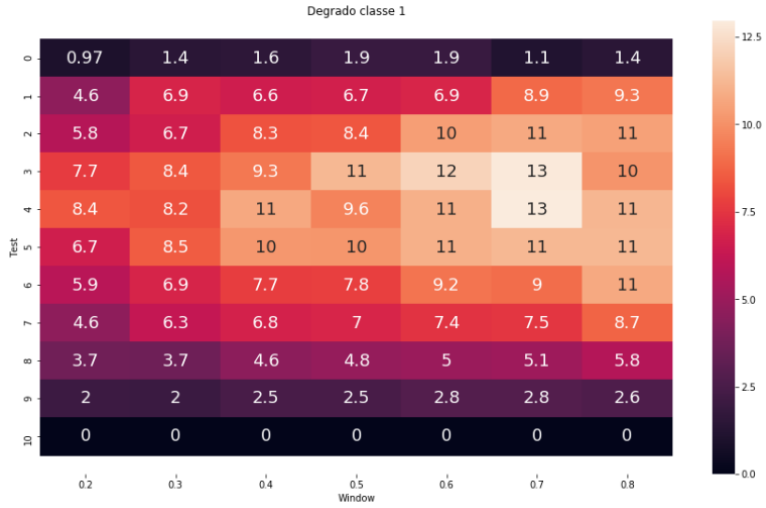


Figura 5.48: White: matrice degrado classe 1 con test su classe 15

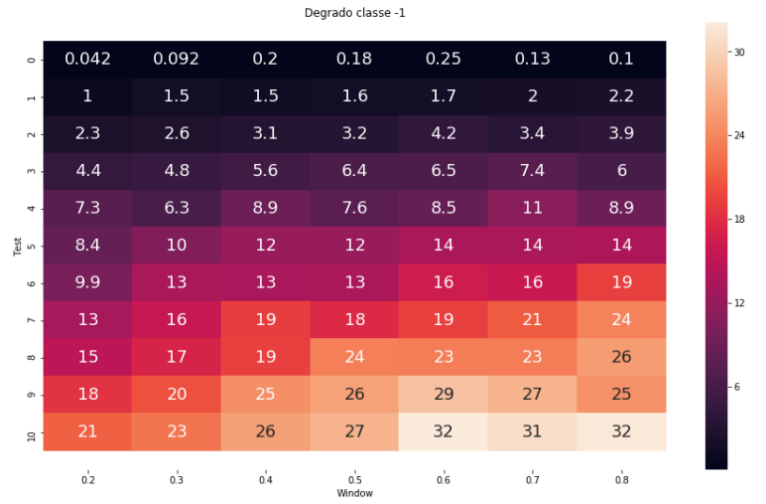


Figura 5.49: White: matrice degrado classe -1 con test su classe 15

Nelle figure 5.50 e 5.51 si riportano le curve delle distanze separate tra le classe -1 e 1. L'omogeneità e la coesione delle distanze nella classe 15 è evidente perché le curve si mantengono più o meno nello stesso range senza raggiungere particolari picchi. Il valore massimo che raggiunge la classe -1 è 350, mentre la classe 1 si ferma a quasi 110. Quello che si nota osservando le distanze più piccole dei punti

considerati come outlier è la maggiore variabilità dei loro valori, questo può essere dovuto alla variabilità interna alla classe 15.

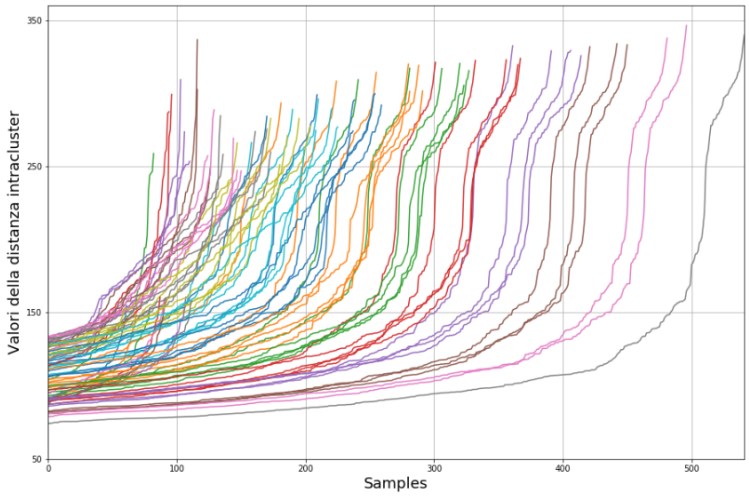


Figura 5.50: White (con test su classe 15): curve delle distanze intra cluster -1 per ogni tempo t

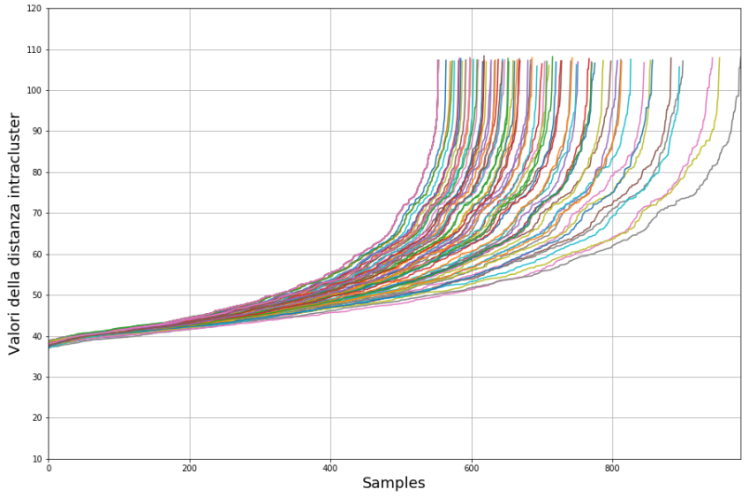


Figura 5.51: White (con test su classe 15): curve delle distanze intra cluster 1 per ogni tempo t

5.2.6 Limitazioni del calcolo della Silhouette nel caso di studio

I risultati fin qui esposti derivano da una modifica della formula usata negli studi esposti in [39]. L'applicazione del calcolo della *Silhouette* era stata effettuata in un contesto multiclasse che si distacca da quello qui analizzato. La formula di questa metrica presenta dei limiti che è possibile notare nei risultati mostrati di seguito. La limitazione principale è che la formula della Silhouette comprende sia i valori della distanza intra cluster che quelli della distanza inter cluster. Il classificatore dell'Isolation Forest è binario per cui il train è fatto su una sola classe e non è possibile per il primo passo valutare la separazione tra più classi. Si è scelto di addestrare nuovamente il modello sulla classe 0 e di iniettare successivamente i dati della classe 10. Come si evince dalle heatmap non si ha un andamento convergente verso un unico punto con massimo valore di degrado e non è possibile rilevare un trend delle percentuali di degrado tra le direzioni della matrice. Le heatmap relative alle due classi hanno un andamento opposto e complementare, questo è dovuto semplicemente all'andamento opposto che hanno le percentuali delle due classi nei vari test. I valori di distanza intra-gruppo e distanza inter-gruppo si compensano a vicenda non permettendo di osservare una deriva dei dati coerente con i dati in ingresso nel test, pertanto come soluzione è stato preferito il primo approccio che elimina la distanza inter-gruppo. Il discorso vale per entrambi i dataset *Gray* e *White*.

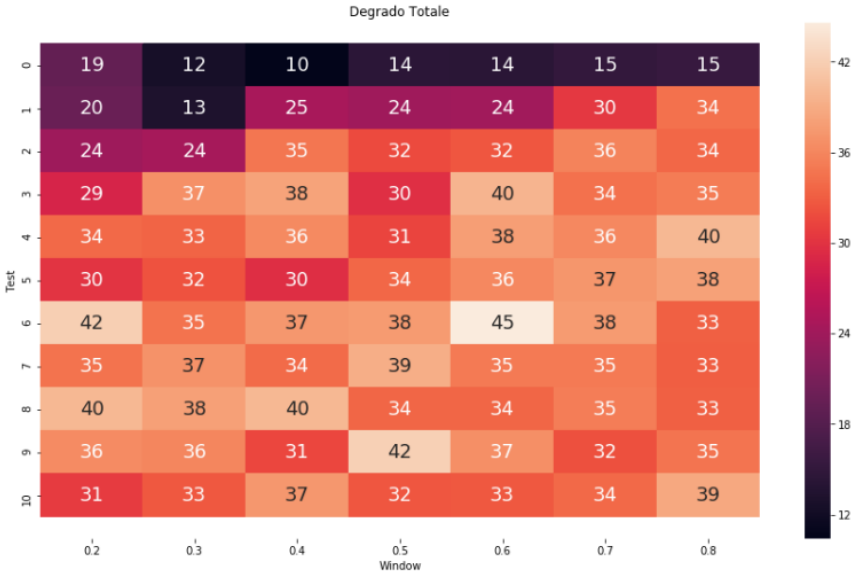


Figura 5.52: Gray: degrado totale con il calcolo della Silhouette

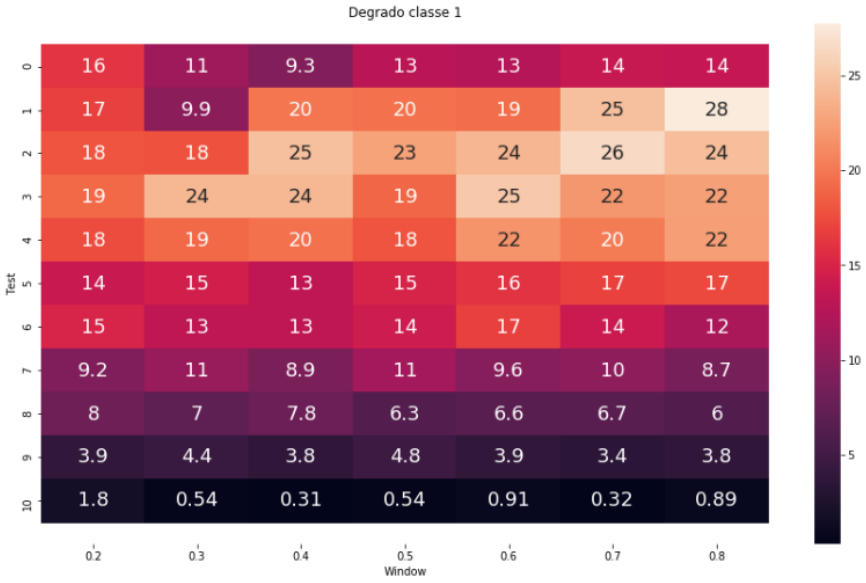


Figura 5.53: Gray: degrado classe 1 con il calcolo della Silhouette

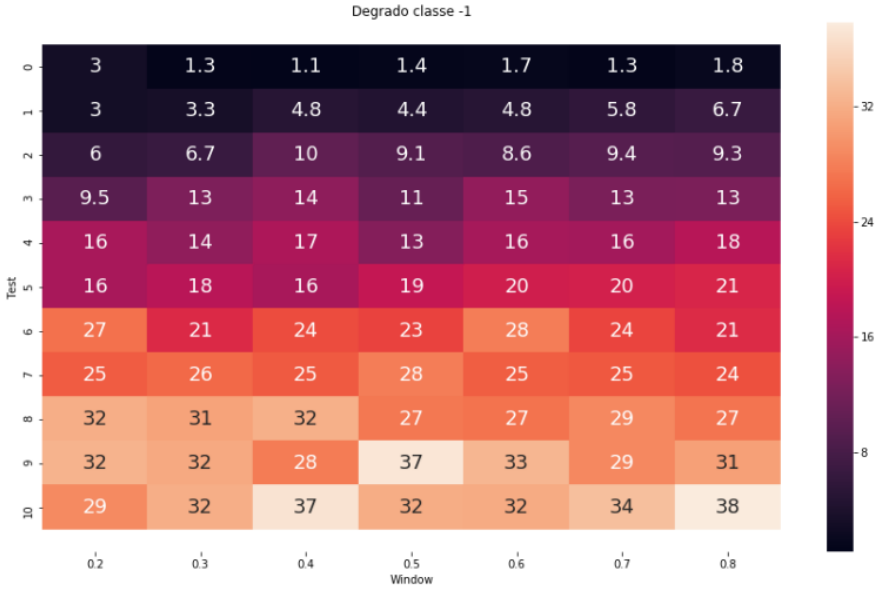


Figura 5.54: Gray: degrado classe -1 con il calcolo della Silhouette

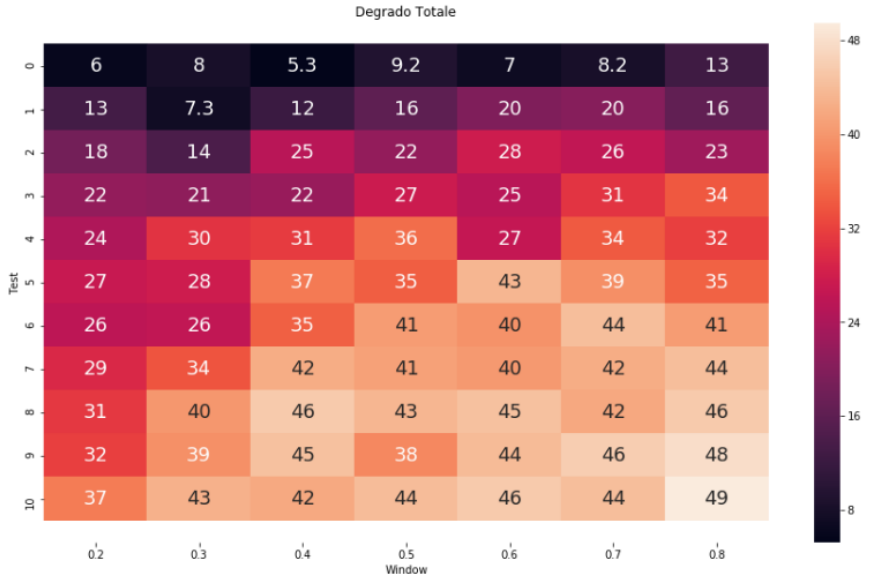


Figura 5.55: White: degrado totale con il calcolo della Silhouette

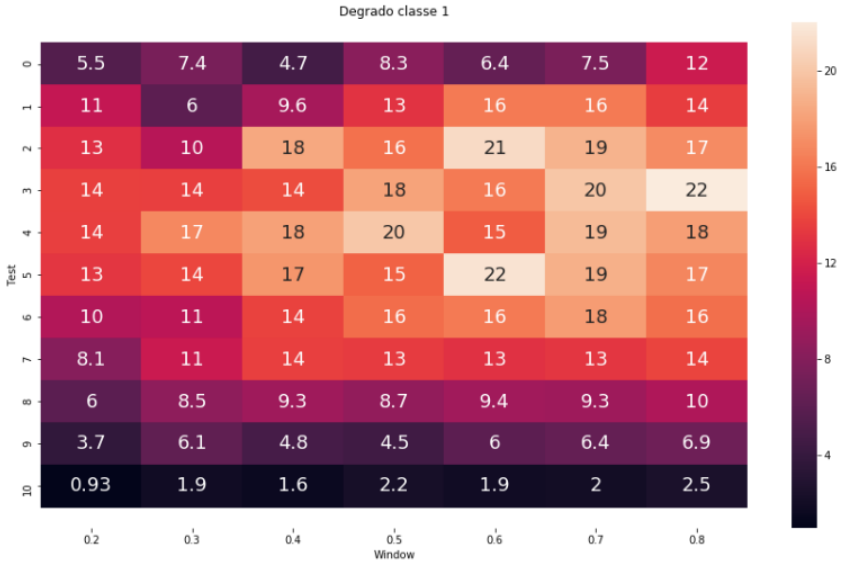


Figura 5.56: White: degrado classe 1 con il calcolo della Silhouette

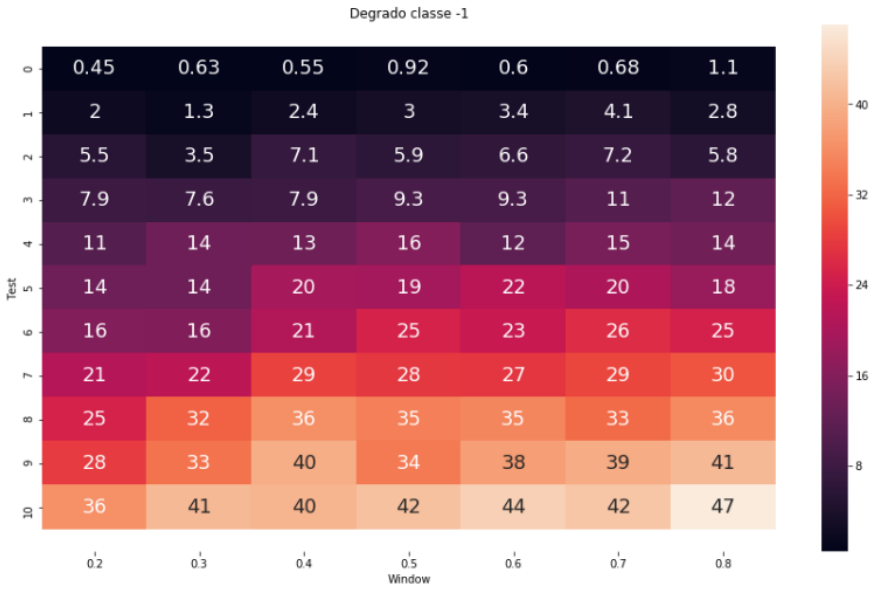


Figura 5.57: White: degrado classe -1 con il calcolo della Silhouette

Capitolo 6

Conclusioni

L'identificazione della deriva dei dati è un problema complesso che non ha ancora trovato una soluzione valida e applicabile in ogni campo. Esistono diverse metodologie e ognuna presenta delle peculiarità che si adattano ai vari casi di studio. In questo elaborato si è posta l'attenzione su quelle metodologie di rilevamento del concept drift basate sulle distribuzioni dei dati e chiamate "data-driven". Si è partiti da un contesto già definito nello stato dell'arte che prevede l'uso dell'indice silhouette per valutare la corretta assegnazione delle etichette da parte di un classificatore multiclasse attraverso il concetto di distanza tra i punti. A partire dai dati a disposizione e dal problema ad essi connesso si è deciso di sfruttare un classificatore binario come l'Isolation Forest per effettuare un'attività di outlier detection in grado di rilevare l'arrivo di una nuova classe e, di conseguenza, la presenza di un comportamento anomalo da parte dei bracci robotici da cui derivano gli stream di dati usati.

Per questa casistica sono state presentate due alternative: la prima riprende quello che è stato riportato come stato dell'arte, la seconda invece presenta una modifica alla formula della silhouette considerando solo la distanza intra-gruppo dei dati. Sono stati effettuati gli esperimenti con entrambe le metodologie e, sulla base dei risultati ottenuti è stato possibile confermare la validità del secondo approccio a discapito del primo.

Lo studio della sola distanza intra-cluster ha permesso di individuare in maniera più immediata il drift dei dati. La particolarità del metodo risiede soprattutto nello studio degli outliers e della loro distribuzione e distanza reciproca. I valori anomali

trovati dal classificatore non sono altro che ciò che arriva sotto forma di novità per il modello. Di conseguenza questa metodologia potrebbe essere una buona soluzione per problematiche di manutenzione e individuazione di guasti e anomalie.

I dataset analizzati sono due ed entrambi provenienti da bracci robotici. I risultati in entrambi i casi sono stati quelli attesi ed è stato possibile rilevare l'arrivo di nuovi dati appartenenti ad una classe diversa da quella di train. Da ciò si deduce che le classi all'interno dei dataset sono ben separate tra loro, permettendo al classificatore di lavorare al meglio. Si tratta di dati che per queste caratteristiche si distaccano da un contesto reale, dove è di gran lunga più complesso distinguere le classi. La metodologia si è mostrata efficace ma possibili sviluppi futuri potrebbero essere l'estensione della validazione sperimentale a nuovi dati in modo da testare l'approccio con dati dalle caratteristiche diverse.

Un altro sviluppo potrebbe essere l'uso di nuove metriche non supervisionate per lo studio della distribuzione dei dati in seguito ad un'attività di classificazione. Ad esempio, l'applicazione di algoritmi di clustering come il k-means o il dbscan potrebbe essere molto utile per riconoscere i sottogruppi presenti all'interno di una classe oppure per separare le aree più dense da quelle meno dense. In particolare, nel caso di studio in esame, poiché la classe 1 è poco soggetta al drift per via della natura del classificatore, potrebbe essere utile studiare meglio i dati etichettati come outliers ed, eventualmente, riconoscere le sottocategorie presenti nella classe -1 e scoprire in questo modo nuove classi.

Sarebbe inoltre interessante la ricerca di nuove metriche in alternativa al calcolo della silhouette in modo da confrontarne l'efficacia e i risultati con quelli ottenuti in questo elaborato.

Ringraziamenti

E' doveroso a questo punto ringraziare tutte le persone che mi hanno sostenuta durante tutti questi anni universitari e, soprattutto, durante i lunghi mesi di lavoro di tesi.

Innanzitutto vorrei ringraziare i miei genitori, colonna portante della mia vita, senza di loro nulla di tutto ciò sarebbe stato possibile. Li ringrazio per avermi sostenuta con amore in ogni mia scelta e per avermi accompagnata in ogni momento importante, vi voglio bene.

Un sentito ringraziamento va alla professoressa Cerquitelli, a Riccardo Callà e Paolo Bethaz per la loro disponibilità, per i consigli utili e il sostegno durante tutto il lavoro di tesi.

Ringrazio le mie compagne di avventura in questo progetto di tesi Rebecca e Ylenia perché con loro ho imparato che l'unione fa la forza, grazie per il vostro supporto e per aver allietato le lunghe giornate di lavoro.

Ringrazio i miei amici torinesi, le colleghe che hanno condiviso con me parte del mio percorso accademico e i miei amici di giù, ognuno di voi nel suo piccolo e con la sua presenza mi ha sostenuta e supportata durante il mio percorso universitario.

Elenco delle tabelle

Elenco delle figure

Bibliografia

[1] Zanotti Laura. *Industria 4.0: storia, significato ed evoluzioni tecnologiche a vantaggio del business*. NetworkDigital360. Set. 2019. URL: https://www.digital4.biz/executive/industria-40-storia-significato-ed-evoluzioni-tecnologiche-a-vantaggio-del-business/ (cit. a p. 5).

[2] Ray Y. Zhong; Xun Xu; Eberhard Klot; Stephen T. Newman. *Intelligent Manufacturing in the Context of Industry 4.0: A Review*. Ott. 2017 (cit. a p. 6).

[3] Fotina Carmine. *Germania e Italia, doppio modello per Industria 4.0*. IlSole24ore. Set. 2017. URL: https://www.ilsole24ore.com/art/germania-e-italia-doppio-modello-industria-40-AEOLA3ZC (cit. a p. 6).

[4] C. Bagnoli; A. Bravin; M. Massaro; A. Vignotto. *I modelli di business vincenti per le imprese italiane nella quarta rivoluzione industriale*. Edizioni Ca Foscari, 2018 (cit. alle pp. 6, 7, 16, 18).

[5] Ministero dello Sviluppo Economico. *Piano Industria 4.0*. www.mise.gov.it. URL: https://www.mise.gov.it/images/stories/documenti/Piano_Industria_40.pdf (cit. a p. 8).

[6] Commissione Europea. *Key lessons from national industry 4.0 policy initiatives in Europe*. https://ec.europa.eu/. Mag. 2017. URL: https://ec.europa.eu/growth/tools-databases/dem/monitor/sites/default/files/DTM_Policy%5C%20initiative%5C%20comparison%5C%20v1.pdf (cit. a p. 9).

[7] Pai ZHENG. «Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives». In: *Springer* 9 (2017) (cit. alle pp. 9, 14, 15).

[8] Markus Lorenz. «Industry 4.0: The future of Productivity and Growth in Manufacturing Industries». In: *Boston Consulting Group* 4 (2015) (cit. a p. 10).

[9] Zanotti Laura. *Industria 4.0: Cos'è, come fare ed esempi concreti di smart manufacturing*. NetworkDigital360. Feb. 2017. URL: https://www.internet4things.it/industry-4-0/industria-4-0-significato-opportunita-ed-esempi-concreti-dello-smart-manufacturing/ (cit. alle pp. 13, 15).

[10] Miragliotta Giovanni. *Smart Manufacturing e Industria 4.0: un po' di storia*. Osservatori.net Digital Innovation. Ott. 2018. URL: https://blog.osservatori.net/it_it/smart-manufacturing-significato (cit. a p. 13).

[11] Li Zhang. «Research on the overall architecture of Internet of Things middleware for intelligent industrial parks». In: *Springer-The International Journal of Advanced Manufacturing Technology* 9 (2019) (cit. a p. 17).

[12] Smactory. *Cos'è l'Industria del futuro e quali vantaggi apporta alle imprese italiane*. Smactory-. URL: https://www.smactory.com/industria4-0-definizione-e-benefici/ (cit. a p. 18).

[13] Jay Lee. «Service innovation and smart analytics for Industry 4.0 and big data environment». In: *Elsevier* (2019) (cit. a p. 19).

[14] Bellini Mauro. *Manifatturiero: cos'è, settori e futuro dell'industria manifatturiera*. Internet4Things. 2020. URL: https://www.internet4things.it/industry-4-0/industria-40-la-nuova-era-del-manifatturiero/ (cit. a p. 20).

[15] QINGLIN QI e FEI TAO. «Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison». In: *IEEEAccess* 2 (2018) (cit. alle pp. 21–25).

[16] Vepa Atamuradov. *Prognostics and Health Management for Maintenance Practitioners - Review, Implementation and Tools Evaluation*. 2017. URL: https://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2016/ijphm_17_060.pdf (cit. a p. 26).

[17] Wikipedia. *Manutenzione*. Wikipedia. URL: https://it.wikipedia.org/wiki/Manutenzione (cit. a p. 27).

[18] VERONICA FORNLÖF. *IMPROVED REMAINING USEFUL LIFE ESTIMATIONS FOR ONCONDITION PARTS IN AIRCRAFT ENGINES*. University of Shodve. 2016. URL: https://www.diva-portal.org/smash/get/diva2:945577/FULLTEXT01.pdf (cit. alle pp. 27, 28).

[19] Wikiversity. *Tipi di manutenzione*. Wikiversity. URL: https://it.wikiversity.org/wiki/Tipi_di_manutenzione (cit. a p. 27).

[20] Hupjé Erik. *9 TYPES OF MAINTENANCE HOW TO CHOOSE THE RIGHT MAINTENANCE STRATEGY*. Roadtoreliability. URL: https://www.roadtoreliability.com/types-of-maintenance (cit. a p. 28).

[21] TECHEDGE. «Cloud e Iot come leve per la predictive maintenance». In: *Network Digital 360* 11 (2018) (cit. alle pp. 29–31).

[22] Aliperto Domenico. *Verso la vera Predictive Maintenance, come fondere IoT e Machine learning*. Internet4things. 2018. URL: https://www.internet4things.it/industry-4-0/verso-la-vera-predictive-maintenance-come-fondere-iot-e-machine-learning/ (cit. a p. 33).

[23] Thorsten Wues. «Machine learning in manufacturing: advantages, challenges, and applications». In: *Taylor & Francis* 5 (2016) (cit. a p. 34).

[24] Susan P. Imberman. «EFFECTIVE USE OF THE KDD PROCESS AND DATA MINING FOR COMPUTER PERFORMANCE PROFESSIONALS». In: *ResearchGate* 1 (2001) (cit. a p. 36).

[25] Jiawei Han. *Data mining concepts and techniques*. Morgan Kaufmann, 2012 (cit. alle pp. 37, 39, 40, 43, 46, 47, 56, 67, 69).

[26] Oded Maimon. «INTRODUCTION TO KNOWLEDGE DISCOVERY IN DATABASES». In: *Tel-Aviv University* () (cit. alle pp. 41–43, 46, 47).

[27] Pang-Ning Tan. *Introduction to Data Mining*. Pearson, 2019 (cit. alle pp. 47, 49–53, 55, 57, 60–63, 65, 89).

[28] Wikipedia. *Random Forest*. Wikipedia. URL: https://en.wikipedia.org/wiki/Random_forest (cit. a p. 54).

[29] Usman Malik. *Random Forest Algorithm with Python and Scikit-Learn*. Stack abuse. URL: https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/ (cit. a p. 54).

[30] Scott Robinson. *Introduction to Neural Networks with Scikit-Learn*. Stack abuse. URL: `https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/` (cit. a p. 55).

[31] Scott Robinson. *K-Nearest Neighbors Algorithm in Python and Scikit-Learn*. Stack abuse. URL: `https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/` (cit. a p. 58).

[32] Zinguo Ding e Minrui Fei. «An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window». In: (2013) (cit. alle pp. 69–71).

[33] Fei Tony Liu. «Isolation Forest». In: *Monash University, Australia* () (cit. alle pp. 70–72).

[34] Wikipedia. *Concept drift*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Concept_drift` (cit. a p. 72).

[35] Ashok Chilakapati. *Concept Drift and Model Decay in Machine Learning*. Towards data science. URL: `https://towardsdatascience.com/concept-drift-and-model-decay-in-machine-learning-a98a809ea8d4` (cit. a p. 72).

[36] Geoffrey I. Webb · Roy Hyde · Hong Cao · Hai Long Nguyen · Francois Petitjean. «Characterizing Concept Drift». In: () (cit. a p. 74).

[37] Jie Lu. «Learning under Concept Drift: A Review». In: *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 12 (2019) (cit. alle pp. 75, 77–81, 84).

[38] Indre Zliobaite. «Learning under Concept Drift: an Overview». In: *Vilnius University* () (cit. alle pp. 76, 77).

[39] Francesco Ventura- Stefano Proto- Daniele Apiletti- Tania Cerquitelli- Simone Panicucci- Elena Baralis- Enrico Macii- Alberto Macii. «A new unsupervised predictive-model self-assessment approach that SCALEs». In: *IEEE International Congress on Big Data (BigData Congress)* (2019) (cit. alle pp. 85, 88, 89, 91–93, 98, 105, 143).

[40] Tania Cerquitelli. «Towards a real-time unsupervised estimation of predictive model degradation». In: (2019) (cit. a p. 85).

[41] Tania Cerquitelli. «Automating concept-drift detection by self-evaluating predictive model degradation». In: (2019) (cit. alle pp. 86, 87).

[42] F.Ventura e al. «A new unsupervised predictive-model self-assessment approach that SCALES». In: *Politecnico di Torino* () (cit. alle pp. 88, 89).

[43] Ezio Melotti. *Perchè usare Python*. HTML.IT. URL: `https://www.html.it/pag/15608/perche-usare-python/` (cit. a p. 93).

[44] Wikipedia. *Scikit-learn*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Scikit-learn` (cit. a p. 94).

[45] Wes McKinney. *Python for Data Analysis*. O'Reilly, 2012 (cit. a p. 95).

[46] Usman Malik. *Lettura e scrittura di file JSON in Python con Panda*. Stackabuse. URL: `https://stackabuse.com/reading-and-writing-json-files-in-python-with-pandas/` (cit. a p. 96).

[47] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. alle pp. 108, 110, 111, 119).